UNIVERSITY OF THESSALY

DOCTORAL THESIS

Design, Development and Performance Evaluation of Ultra Low-Power Wireless Sensor Network Devices

Author:
Giannis KAZDARIDIS

Supervisor:
Assoc. Prof. Thanasis KORAKIS

A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

in the

Department of Electrical and Computer Engineering

Dissertation Committee:
Assoc. Prof. Athanasios Korakis
Prof. Leandros Tassiulas
Assoc. Prof.Antonios Argyriou

April 2021

Declaration of Authorship

I, Giannis KAZDARIDIS, declare that this thesis titled, "Design, Development and Performance Evaluation of Ultra Low-Power Wireless Sensor Network Devices" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ____

Date: ____

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Περίληψη

Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Σχεδιασμός, Ανάπτυξη και Αξιολόγηση Απόδοσης Συσκευών Ασυρμάτων Δικτύων Αισθητήρων Εξαιρετικά Χαμηλής Κατανάλωσης Ενέργειας

Γιάννης Καζδαρίδης

Η ενεργειακή απόδοση στον τομέα του **Internet-of-Things (IoT)** κυριαρχεί στο ενδιαφέρον της ερευνητικής κοινότητας. Στις περισσότερες πραγματικές εφαρμογές, οι αισθητήρες λειτουργούν με μπαταρία, αντιμετωπίζοντας τον εγγενή περιορισμό της διάρκειας ζωής που εξαρτάται αποκλειστικά από το υπόλοιπο φορτίο της μπαταρίας και το ενεργειακό προφίλ του κόμβου. Μια κοινή προσέγγιση για εξοικονόμηση ενέργειας στα δίκτυα αισθητήρων είναι η μέθοδος του **duty-cycle**, καθώς ο μεγάλος βαθμός αδρανούς ακρόαση **(idle listening)** είναι μια σημαντική πηγή σπατάλης ενέργειας. Στην πραγματικότητα, η κατανάλωση ρεύματος σε κατάσταση αδράνειας είναι περίπου ίση με την ενέργεια που απαιτείται για τη λήψη ενός πακέτου. Για τον λόγο αυτόν, οι κόμβοι αισθητήρων έχουν σχεδιαστεί έτσι, ώστε να εισέρχονται σε λειτουργία χαμηλής κατανάλωσης, την αποκαλούμενη κατάσταση ύπνου **(sleep state)**, προκειμένου να εξοικονομήσουν όσο το δυνατόν περισσότερη ενέργεια κατά τη διάρκεια των ανενεργών περιόδων τους. Η κατάσταση ύπνου διακόπτεται από πολύ σύντομα συμβάντα, όπου οι αισθητήρες αισθάνονται, επεξεργάζονται και μεταδίδουν δεδομένα. Μπορεί να φαίνεται λογικό η κατανάλωση ενέργειας στην κατάσταση ύπνου να θεωρηθεί αμελητέα, εφόσον περισσότερες από δύο με τρείς τάξεις μεγέθους διαχωρίζουν την κατανάλωση ρεύματος στην κατάσταση ύπνου και την ενεργή. Ωστόσο, δεδομένου του γεγονότος ότι οι τυπικές εφαρμογές αισθητήρων λειτουργούν σε αρκετά χαμηλά **duty-cycles** που κυμαίνονται από **0.01 %** έως **1 %**, αναμένεται ότι και οι δύο καταστάσεις επιδρούν σημαντικά στη συνολική κατανάλωση ενέργειας του συστήματος. Συγκεκριμένα, το ρεύμα κατά την διάρκεια του ύπνου είναι συνήθως της τάξης μερικών **μA**, πράγμα που υποδηλώνει ότι μπορεί να επιτευχθεί σημαντική εξοικονόμηση ενέργειας. Αξίζει να σημειωθεί ότι οι προαναφερθείσες συσκευές **IoT** διαθέτουν πολύ μεγάλο δυναμικό εύρος στο προφίλ κατανάλωσης ενέργειάς τους, το οποίο καθιστά αρκετά δύσκολη την αξιολόγηση της συμπεριφοράς τους, προκειμένου να αναπτυχθούν ενεργειακά αποδοτικά σχήματα και αλγόριθμοι.

Ένας βασικός παράγοντας για τη μελέτη της συμπεριφοράς των τεχνολογιών **IoT** είναι η δημιουργία μιας πειραματικής υποδομής μεγάλης κλίμακας που αποτελείται από διάφορους μεμονωμένους κόμβους που έχουν παραταχθεί σε ρεαλιστικές συνθήκες, προκειμένου να εξαχθούν όσο το δυνατόν πιο ακριβείς πληροφορίες. Ιδανικά, η προαναφερθείσα εγκατάσταση πρέπει να μπορεί να ρυθμιστεί εξ’ αποστάσεως προκειμένου να υποστηρίζει την εκτέλεση διαφορετικών σεναρίων, βοηθώντας στην αξιολόγηση των τεχνολογιών που βρίσκονται υπό εξέταση. Τα τελευταία χρόνια έχουν παρουσιαστεί νέες **Low-Power WAN (LPWAN) IoT** τεχνολογίες, με την τεχνολογία **LoRa** να ξεχωρίζει, η οποία υπόσχεται εξαιρετική εμβέλεια επικοινωνίας λόγω των υψηλών ορίων ευαισθησίας που υποστηρίζει.

Οι θεμελιώδεις ερωτήσεις που προσπαθούμε να απαντήσουμε είναι οι εξής: **1)** Μπορούμε να εφαρμόσουμε νέες μεθόδους για να μειώσουμε την κατανάλωση ενέργειας των σύγχρονων αισθητήρων για την παράταση της διάρκειας ζωής τους σε μπαταρία μίας φόρτισης; **2)** Μπορούμε να αναπτύξουμε ασύγχρονα δίκτυα για την ασύρματη αφύπνιση συγκεκριμένων κόμβων του δικτύου, προκειμένου να ξεπεράσουμε τα εμπόδια που δημιουργούνται από την υιοθέτηση της μεθόδου λειτουργίας με **duty-cycle**; **3)** Επιπλέον, πώς είναι δυνατόν να μετρήσουμε και να χαρακτηρίσουμε με ακρίβεια το προφίλ κατανάλωσης ενέργειας των κόμβων **IoT** που παρουσιάζουν μεγάλο δυναμικό εύρος που κυμαίνεται από λίγα **nA** όταν είναι ανενεργά σε αρκετά **mA** όταν είναι πλήρως ενεργά; ή/και να δημιουργήσουμε τη σχετική υποδομή που θα επιτρέψει την αξιολόγηση της κατανάλωσης ενέργειας των σχετικών συστημάτων δικτύωσης; **4)** Τέλος, μπορούμε να αξιολογήσουμε την απόδοση των πρόσφατων τεχνολογιών **IoT Low-Power WAN** όπως το πρότυπο **LoRa**; αξιοποιώντας παράλληλα τις δυνατότητές του, ώστε να βοηθήσουν στην ανάπτυξη σεναρίων εφαρμογών μεγάλης εμβέλειας;

Αρχικά, ξεκινάμε παρουσιάζοντας μια καινοτόμα αρχιτεκτονική διαχείρισης ενέργειας προς την εξάλειψη της κατανάλωσης ενέργειας των **duty-cycled IoT** συσκευών κατά τη διάρκεια των ανενεργών περιόδων. Το σχέδιό μας προτείνει τη χρήση ενός **off-chip Real-Time-Clock (RTC)** που χρησιμοποιείτε για τον έλεγχο της τροφοδοσίας του υπό εξέταση κόμβου-αισθητήρα, ενεργοποιώντας ή απενεργοποιώντας την παροχή ενέργειας του με τη μέθοδο του **power-gating.** Το επιλεγμένο **RTC** διαθέτει εξαιρετικά χαμηλής κατανάλωση ενέργειας και είναι η μόνη μονάδα που παραμένει ενεργοποιημένη κατά τη διάρκεια του ύπνου, επομένως η συνολική κατανάλωση του κόμβου μειώνεται σημαντικά. Η προτεινόμενη αρχή μπορεί να υιοθετηθεί από οποιονδήποτε εμπορικό ή πρωτότυπο **IoT** κόμβο, προκειμένου να επεκταθεί το προσδόκιμο ζωής των εφαρμογών που χρειάζονται μπαταρία, ωθώντας τα ρεύματα κατά τον ύπνο μία τάξη μεγέθους χαμηλότερα. Επισημαίνουμε, ότι το κόστος του προτεινόμενου συστήματος είναι μικρότερο από **3** ευρώ, ενώ η πολυπλοκότητα του κυκλώματος είναι ιδιαιτέρως μικρή. Η άμεση σύγκριση της κατανάλωσης ενέργειας σε κατάσταση **sleep state** με άλλους **state-of-the-art** αισθητήρες δείχνει βελτιώσεις περίπου κατά **98 % - 99.8 %**, ενώ αποδεικνύουμε ότι το προσδόκιμο ζωής των ίδιων κόμβων μπορεί να παραταθεί από **2.7** έτη σε **19** έτη σε συγκεκριμένα **duty-cycles.**

Στη συνέχεια, εστιάζουμε στις ασύγχρονες αφυπνίσεις στα δίκτυα αισθητήρων για την υπέρβαση των περιορισμών που παρουσιάζονται στις **duty-cycled** εφαρμογές. Για το σκοπό αυτό, προτείνουμε ένα νέο σχήμα για **semi-passive Wake-Up Receiver** που εμφανίζει αξιοσημείωτη ευαισθησία, κάτω των **-70 dBm**, ενώ οι σύγχρονοι δέκτες απεικονίζουν ευαισθησία έως και **-55 dBm.** Ο προτεινόμενος δέκτης χρησιμοποιεί την τυπική αρχή ενός **envelope detector** που συλλέγει ενέργεια **RF** από την κεραία του, ενώ χρησιμοποιεί έναν ενισχυτή λειτουργίας νάνο-ισχύος για να εντείνει το ληφθέν σήμα πριν από την τελική αποκωδικοποίηση που πραγματοποιείται με τη βοήθεια ενός κυκλώματος συγκριτή **(comparator circuit).** Λειτουργεί στη ζώνη **868 MHz ISM** χρησιμοποιώντας **OOK** σήματα που διαδίδονται μέσω των πομποδεκτών τεχνολογίας **LoRa**, ενώ υποστηρίζει επίσης δυνατότητα αντιστοίχισης διευθύνσεων **(address matching)** για να ξυπνήσει μόνο καθορισμένους κόμβους του δικτύου. Η κατανάλωση ενέργειας του ανεπτυγμένου δέκτη είναι πολύ χαμηλή, έως και τα **580 nA**, παραμένοντας στα ίδια επίπεδα κατανάλωσης ενέργειας με τις σύγχρονες υλοποιήσεις.

Το επόμενο βήμα μας είναι η μέτρηση της κατανάλωσης ενέργειας **IoT** συσκευών που παρουσιάζουν ένα ευρύ φάσμα κατανάλωσης. Για τον λόγο αυτό παρουσιάζουμε τον **eProfiler**, ένα μετρητή που διαθέτει έναν αυτόματο διακόπτη αντιστάσεων **(auto-ranging shunt-resistor switch)** που υποστηρίζει εξαιρετικά γρήγορες εναλλαγές των **10 ns**, που διαμορφώνονται με τη βοήθεια συγκριτών υψηλής ταχύτητας, που μπορούν να προσαρμοστούν επιτυχώς σε οποιαδήποτε απότομη μετάβαση. Επιπλέον, χρησιμοποιείται ένας **Analog-to-Digital Converter (ADC)** με πολλαπλές εισόδους για την ταυτόχρονη παρακολούθηση των αντιστάσεων, παρέχοντας ταχύτητα **150 kSamples/s** σε ανάλυση **16 bit.** Ο προτεινόμενος μετρητής μπορεί να αξιοποιηθεί για την παρακολούθηση σε πραγματικό χρόνο, αλλά και για πειράματα μακράς διάρκειας ενώ το κόστος του δεν ξεπερνά τα **90** ευρώ. Επιπλέον, το προτεινόμενο σύστημα μπορεί να ενεργοποιήσει ή να εντοπίσει εναλλαγές κατάστασης σε **I/O pins** με μέγιστη καθυστέρηση **2 μs**, προκειμένου συσχετιστούν διάφορα περιστατικά **(events)** με τις μετρήσεις κατανάλωσης ενέργειας για την εξαγωγή ακριβών συμπερασμάτων. Ο **eProfiler** διαθέτει ένα ευρύ εύρος μετρήσεων **1.000.000:1**, ενώ παράλληλα μπορεί να μετρήσει ρεύματα μερικών **nA.** Το μέσο σφάλμα του μετρητή είναι **0.45 %**, με το μέγιστο σφάλμα να μην ξεπερνά το **1.6 %.**

Στη συνέχεια παρουσιάζουμε μια πειραματική υποδομή **(testbed)** σε κλίμακα πόλης, που βασίζεται στην τεχνολογία **LoRa** και χρησιμοποιεί πολλές συσκευές ανίχνευσης διασκορπισμένες σε όλη την αστική περιοχή για την αξιολόγηση της ποιότητας του αέρα και των καιρικών συνθηκών σε πραγματικό χρόνο. Στο **testbed** αυτό έχουμε αναπτύξει επιπλέον ένα **framework** αξιολόγησης της ποιότητας της κάθε σύνδεσης καταγράφοντας το **Packet Delivery Ratio (PDR)** σε σχέση με το **RSSI** για να χαρακτηρίσουμε την απόδοση του προτύπου **LoRa** υπό ρεαλιστικές συνθήκες. Τα πειραματικά αποτελέσματα, που συλλέχθηκαν σε περίοδο **2** μηνών, αναλύουν αποτελεσματικά την απόδοση του **LoRa** σε ένα ευρύ φάσμα του πρωτοκόλλου. Τέλος, παρουσιάζουμε πειράματα στο εργαστήριο που χαρακτηρίζουν την αποδοτικότητα της τεχνολογίας **LoRa** όσον αφορά την απόδοση ενέργειας ανά **bit (energy efficiency per bit)**, μαζί με πολύτιμες πληροφορίες που εξήχθησαν από τα πειράματά μας και στοχεύουν στην ανάπτυξη ενεργειακά αποδοτικών πρωτοκόλλων.

Ακολούθως, παρουσιάζουμε την πλατφόρμα **NITOS BikesNet**, μια υποδομή για υποστήριξη κινητών πειραμάτων σε κλίμακα πόλης που βασίζεται σε ποδήλατα εθελοντών χρηστών. Το **BikesNet** χρησιμοποιεί έναν προσαρμοσμένο ενσωματωμένο κόμβο που μπορεί να εξοπλιστεί με διαφορετικού τύπους αισθητήρων και να τοποθετηθεί εύκολα σε ένα ποδήλατο, προκειμένου να συλλέξει ευκαιριακά περιβαλλοντικές και **WiFi** μετρήσεις σε διάφορα μέρη της πόλης. Οι χρήστες μπορούν απομακρυσμένα να δεσμεύσουν και να παραγοντοποιήσουν τους κόμβους αισθητήρων στα ποδήλατα, καθώς και να συλλέξουν/οπτικοποιήσουν τις μετρήσεις τους μέσω του **OMF/OML framework**, το οποίο επεκτάθηκε προκειμένου να διαχειριστεί τη διαλείπουσα συνδεσιμότητα των κινητών κόμβων. Παρέχουμε επίσης ανάλυση απόδοσης του πρωτότυπου κόμβου μας όσον αφορά το **sensing latency**, την ικανότητα μετάδοσης δεδομένων από άκρο σε άκρο και την κατανάλωση ενέργειας, και δείχνουμε ένα πρώτο πείραμα που πραγματοποιήθηκε χρησιμοποιώντας το **NITOS BikesNet** στην πόλη του Βόλου.

Το επόμενο μέρος της εργασίας μας επικεντρώνεται στον τομέα των υποβρύχιων δικτύων αισθητήρων, στα οποία η ενεργειακά αποδοτική λειτουργία είναι ζωτικής σημασίας. Πιο συγκεκριμένα, παρουσιάζουμε το σύστημα παρακολούθησης ενέργειας **EVERUN,** που αποτελείται από στοιχεία υλικού και λογισμικού. Ενσωματώνουμε το ανεπτυγμένο σύστημα στα Αυτόνομα Υποβρύχια Οχήματα **(AUVs)** του υποβρύχιου **testbed** του **SUNRISE**, για την αξιολόγηση της λειτουργίας τους όσον αφορά την ενεργειακή απόδοση. Μέσω της εκτέλεσης ενός ευρέος συνόλου πειραμάτων υπό ρεαλιστικές συνθήκες, επισημαίνουμε τους περιορισμούς των εργαλείων αξιολόγησης ενέργειας βάσει μοντέλου και χαρακτηρίζουμε την ενεργειακή απόδοση βασικών πρωτοκόλλων και μηχανισμών. Η ακρίβεια των συλλεγόμενων δεδομένων ενέργειας, μαζί με τα ενδιαφέροντα συμπεράσματα, επιβεβαιώνουν τη δυνατότητα εφαρμογής της προσέγγισής μας στην αξιολόγηση της ενεργειακής απόδοσης των προτεινόμενων λύσεων.

Τέλος, παρουσιάζουμε μια πρωτότυπη αρχιτεκτονική για τη δυνατότητα **on-line** επικοινωνίας με εγκαταστάσεις παρακολούθησης θαλάσσιου περιβάλλοντος. Για το σκοπό αυτό, βασιζόμαστε σε ένα σύνολο τεχνολογιών επικοινωνίας που κυμαίνονται από **IoT** πρότυπα επικοινωνίας χαμηλού ρυθμού δεδομένων έως τα ευρέως υιοθετημένα πρωτόκολλα **WiFi** και **LTE** που είναι σε θέση να υποστηρίξουν εφαρμογές με μεγάλες απαιτήσεις. Για να επιτύχουμε ενεργειακά αποδοτική λειτουργία, απενεργοποιούμε όλες τις διεπαφές και τα περιφερειακά που καταναλώνουν ενέργεια εκμεταλλευόμενοι τη μέθοδο του **power-gating,** ενώ διατηρούμε ενεργή μια διεπαφή μεγάλης εμβέλειας και χαμηλής ισχύος **(LoRa standard),** αποκλειστικά για τον έλεγχο/αφύπνιση των υπόλοιπων εξαρτημάτων.

UNIVERSITY OF THESSALY

Abstract

Department of Electrical and Computer Engineering

Design, Development and Performance Evaluation of Ultra Low-Power Wireless Sensor Network Devices

by Giannis KAZDARIDIS

Energy-efficiency in the domain of *Internet-of-Things* (*IoT*) dominates the interest of the research community. In most real-world applications, sensors are battery operated, facing the inherent constraint of life duration that is solely dependent on the battery's remaining charge and the node's power profile. A common approach for saving energy in sensor networks is the *duty-cycle* concept, since overhearing and idle listening are major sources of energy wastage. As a matter of fact, current consumption in idle state is roughly equal to the energy required for receiving a packet through the radio. To this end, sensor nodes are configured to enter a low-power mode, the so-called sleep state, in order to save as much energy as possible during their inactive periods. The sleep state is interrupted by short, burst events, where sensors sense, process and propagate data. It might seem reasonable to neglect energy consumption in the sleep state, since more than three orders of magnitude separate current consumption in sleep and active states. However, given the fact that typical sensor applications operate at quite low *duty-cycles* ranging from *0.01 %* to *1 %*, it is expected that both states account for the systems' power budget expenditure. Notably, sleep current is usually in the order of a few μA, which suggests that substantial energy savings can be attained.

Apparently, the aforementioned *IoT* devices feature extreme dynamic range in their power consumption profile, which is rather challenging when it comes to evaluate their behavior, in order to develop energy-efficient schemes and algorithms.

A key enabler for studying the behavior of *IoT* technologies is the creation of a large-scale setup that consist of several individual nodes deployed under realistic conditions, in order to extract as accurate insights as possible. Ideally, the aforementioned setup must be remotely configurable in order to support the execution of different scenarios to aid in the assessment of the under consideration technologies. Over the last years new *IoT Low-Power WAN* (*LPWAN*) technologies, such as the *LoRa* standard have been introduced that promise exceptional communication ranges due to their extremely high sensitivity thresholds.

The fundamental questions that we try to answer are the following: **1)** Can we apply novel methods in order to drop the power consumption of the state-of-the-art sensors towards extending their life duration on a single charge battery? **2)** Can we develop asynchronous networks for awaking specific network's nodes wirelessly in order to surpass the barriers created by the adoption of the duty-cycle concept? **3)** Moreover, how is it possible to accurately measure and characterize the power consumption profile of *IoT* nodes that present wide dynamic range drawing from a few nA when inactive to several mA when being fully activated? and/or to create relevant infrastructure for allowing the power consumption evaluation of relative networking systems? **4)** Lastly, how can we evaluate the performance of the recently introduced *IoT Low-Power WAN* technologies such as the *LoRa* standard, while also exploiting its capabilities so as to aid in the development of application scenarios that require long-range communication standards?

Initially, we begin by illustrating a novel power management architecture towards eliminating the power draw of *duty-cycled IoT* sensing devices during inactive periods. Our principle suggests the employment of an off-chip *Real-Time-Clock* (*RTC*) configured to control the power supply of the under consideration mote, by enabling or disabling its power in a *power-gating* fashion. The selected *RTC* features an ultra-low power profile and it is the only module that remains powered during sleep, hence the overall mote's consumption is substantially diminished. The proposed principle can be adopted by any commercial or prototyping *IoT* mote, in order to extend the life expectancy of battery-powered applications, by pushing sleep currents an order of magnitude lower. We remark that the cost of the proposed system is less than *3 euros*, while the complexity of the suggested circuitry is trivial. Direct comparison of power draw in *sleep state* with *state-of-the-art* sensors illustrates improvements of roughly *98 % - 99.8 %*, while we demonstrate that the life expectancy of the same motes can be prolonged from *2.7 years* to *19 years* under specific *duty-cycles*.

Following this, we focus on asynchronous awakenings in sensor networks towards surpassing the limitations presented in *duty-cycled* applications. To this end, we propose a new scheme for semi-passive *Wake-Up Receiver* circuits that attains remarkable sensitivity beyond *-70 dBm*, while the state-of-the-art receivers illustrate sensitivity of up to *-55 dBm*. The receiver employs the typical principle of an envelope detector that harvests RF energy from its antenna, while it employs a nano-power operation amplifier to intensify the obtained signal prior to the final decoding that is realized with the aid of a comparator circuit. It operates at the *868 MHz ISM* band using *OOK signals* propagated through *LoRa* transceivers, while also supporting addressing capabilities in order to awake only the specified network's nodes. The power expenditure of the developed receiver is as low as *580 nA*, remaining at the same power consumption levels as the state-of-the-art implementations.

As our next step, measuring the power expenditure of sensing devices that illustrate a wide current range. Our meter features an auto-ranging shunt-resistor switch that supports ultra-fast alternations of *10 ns*, formed with the aid of high-speed comparators, that can successfully adapt to any sharp transition. Moreover, a high-speed *Analog-to-Digital Converter* (*ADC*) with multiple inputs is employed to simultaneously monitor the shunt-resistors, delivering a speed of *150 kSamples/s* at *16 bit resolution*. The proposed meter can be leveraged for the real-time, as well as the long-term monitoring of *IoT* devices with its cost being less than *90 euros*. Furthermore, the proposed system can actuate or trace state alternations on *I/O pins* with a maximum delay of *2 μs*, in order to provide correlation capabilities with the obtained power measurements. The eProfiler features a wide dynamic range of 1.000.000:1 while also being able to monitor currents of a few *nA*. The average obtained error of the meter is *0.45 %*, with a maximum error of *1.6 %*.

Next we present a *LoRa* based city-scale testbed that employs several sensing devices scattered across the urban area to characterize air quality and weather conditions in real-time. The installation is augmented through a custom link quality evaluation framework that continuously monitors the *Packet Delivery Ratio* versus *RSSI* relation to characterize the performance of *LoRa* standard under realistic conditions. Experimental results, collected over a period of 2 months, efficiently analyze *LoRa's* performance across a wide range of protocol configurations. Finally, we also present in-lab experiments that characterize the efficiency of *LoRa* modules in terms of power and *energy efficiency per bit*, along with valuable insights aimed at the development of energy efficient protocol improvements.

We follow by presenting the *NITOS BikesNet platform*, a city-scale mobile sensing infrastructure that relies on bicycles of volunteer users. The *BikesNet* employs a custom-built embedded node that can be equipped with different types of sensors, and which can be easily mounted on a bicycle in order to opportunistically collect environmental and WiFi measurements in different parts of the city. Experimenters can remotely reserve and control the sensor nodes on bicycles as well as collect/visualize their measurements via the *OMF/OML* framework, which was extended in order to handle the intermittent connectivity and disconnected operation of the mobile nodes. We also provide a performance analysis of our node prototype in terms of sensing latency, end-to-end data transmission capability and power consumption, and report on a first experiment that was performed using the *NITOS BikesNet* in the city of Volos, Greece.

The next part of our work is focused on the domain of the underwater sensor networks, in which energy efficient operation is of crucial importance. More specifically, we introduce the *EVERUN* power monitoring framework, consisting of hardware and software components. We integrate the developed system into the *Autonomous Underwater Vehicles* (*AUVs*) of the SUNRISE underwater testbed, towards evaluating their operation in terms of energy efficiency. Through the execution of a wide set of experiments under realistic conditions, we highlighted the limitations of model-based energy evaluation tools and characterized the energy efficiency performance of key protocols and mechanisms. The accuracy of the collected power data, along with the interesting derived findings, verified the applicability of our approach in evaluating the energy efficiency performance of proposed solutions.

Finally, we present a novel architecture for enabling on-line communication with marine environment monitoring deployments. To this aim, we rely on a set of communication technologies that range from *IoT* related low data rate communication standards to the widely adopted *WiFi* and *LTE* protocols that are able to support bandwidth demanding applications. To achieve energy-efficient operation we turn off all the power-hungry interfaces and peripherals exploiting the power-gating method, while we retain active a low-power long-range interface (*LoRa* standard), dedicated to control the rest of the components.

List of Publications

The results, ideas and figures of this thesis have been included in the following publications:

Journals

[J1] G. Kazdaridis, I. Zografopoulos, N. Sidiropoulos, P. Symeonidis and T. Korakis. **eProfiler 2: A High-Precision Power Monitoring Tool Supporting Extreme Dynamic Consumption Range**, *under submission*.

[J2] S. Keranidis, G. Kazdaridis, V. Passas, T. Korakis, I. Koutsopoulos and L. Tassiulas. **NITOS Energy Monitoring Framework: Real time Power Monitoring in Experimental Wireless Network Deployments**, *ACM Mobile Computing and Communications Review (MC2R), Special issue, 2014*, ACM.

Conferences

[C1] G. Kazdaridis, N. Sidiropoulos, I. Zografopoulos, P. Tzimotoudis and T. Korakis. **eWake: A Novel Architecture for Semi-Active Wake-Up Radios Attaining Ultra-High Sensitivity at Extremely-Low Consumption**, *under submission*.

[C2] G. Kazdaridis, I. Zografopoulos, N. Sidiropoulos, P. Symeonidis and T. Korakis. **eProfiler: High-Precision Power Monitoring System for IoT Devices Featuring Extreme Dynamic Range of Operation**, in *the proceedings of the ACM International Workshop on Energy Neutral Sensing Systems (ENSsys) 2020*, ACM.

[C3] G. Kazdaridis, N. Sidiropoulos, I. Zografopoulos, P. Symeonidis and T. Korakis. **Nano-Things: Pushing Sleep Current Consumption to the Limits in IoT Platforms**, in *the proceedings of the ACM International Conference on the Internet of Things (IoT), 2020*, ACM.

[C4] G. Kazdaridis, S. Keranidis, P. Symeonidis, P. Tzimotoudis and T. Korakis. **Evaluation of LoRa Performance in a City-wide Testbed: Experimentation Insights and Findings**, in *the proceedings of the ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH), MobiCom, 2019*, ACM.

[C5] G. Kazdaridis, S. Keranidis, P. Symeonidis, P. Dias, P. Gonçalves, B. Loureiro, P. Gjanci and C. Petrioli. **EVERUN - Enabling Power Consumption Monitoring in Underwater Networking Platforms**, in *the proceedings of the ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH), Mobicom, 2017*, ACM.

[C6] G. Kazdaridis, P. Symeonidis, I. Zographopoulos, T. Korakis, K. Klun and N. Kovac. **On the Development of Energy-Efficient Communications for Marine Monitoring Deployments**, in *the proceedings of the IEEE International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS), 2017*, IEEE.

[C7] G. Kazdaridis, D. Stavropoulos, V. Maglogiannis, T. Korakis, S. Lalis, L. Tassiulas. **NITOS BikesNet: Enabling Mobile Sensing Experiments through the OMF Framework in a city-wide environment**, in *the proceedings of the IEEE International Conference on Mobile Data Management (MDM), 2014*, IEEE.

[C8] S. Keranidis, G. Kazdaridis, V. Passas, T. Korakis, I. Koutsopoulos, L. Tassiulas. **Online Energy Consumption Monitoring of Wireless Testbed Infrastructure through the NITOS EMF Framework**, in *the proceedings of the ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH), Mobicom, 2013*, ACM. **(Best Paper Award)**

Demonstrations

[D1] G. Kazdaridis, N. Sidiropoulos, I. Zografopoulos, and T. Korakis. **A Novel Architecture for Semi-Active Wake-Up Radios Attaining Sensitivity Beyond -70 dBm**, in *the proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN) 2021*, ACM/IEEE.

[D2] G. Kazdaridis, P. Tzimotoudis, N. Sidiropoulos, P. Symeonidis and T. Korakis. **The NITOS Wireless Sensor Network Testbed for Experimenting with Long-Range Technologies**, in *the proceedings of the ACM International Conference on the Internet of Things (IoT), 2020*, ACM.

[D3] G. Kazdaridis, N. Sidiropoulos, I. Zografopoulos, P. Symeonidis and T. Korakis. **The ICARUS Mote: Employing Off-Chip RTC to Attain 22 nA Sleep Current in Duty-Cycled IoT Devices**, in *the proceedings of the ACM International Conference on the Internet of Things (IoT), 2020*, ACM.

[D4] G. Kazdaridis, N. Sidiropoulos, I. Zographopoulos, P. Symeonidis and T. Korakis. **Nano Power Draw in Duty-Cycled Wireless Sensor Networks**, in *the proceedings of the ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH), MobiCom, 2019*, ACM.

[D5] G. Kazdaridis, I. Zographopoulos, P. Symeonidis, P. Skrimponis, T. Korakis and L. Tassiulas. **In-situ Power Consumption Meter for Sensor Networks supporting Extreme Dynamic Range**, in *the proceedings of the ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH), Mobicom, 2017*, ACM.

[D6] G. Kazdaridis, P. Skrimponis, I. Zographopoulos, P. Symeonidis, T. Korakis and L. Tassiulas. **Enabling Asynchronous Awakenings in Wireless Sensor Networks Towards Removing Duty-Cycle Barriers**, in *the proceedings of the ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH), Mobicom, 2017*, ACM.

[D7] G. Kazdaridis, D. Stavropoulos, S. Ioannidis, T. Korakis, S. Lalis, L. Tassiulas. **A Demonstration of the NITOS BikesNet Framework**, in *the proceedings of the IEEE International Conference on Mobile Data Management (MDM), 2014*, IEEE.

[D8] V. Maglogiannis, G. Kazdaridis, D. Stavropoulos, T. Korakis and L. Tassiulas. **Enabling Mobile Sensing through a DTN Framework**, in *the proceedings of the ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH), Mobicom, 2013*, ACM.

[D9] S. Keranidis, G. Kazdaridis, V. Passas, T. Korakis,I. Koutsopoulos, L. Tassiulas. **Online Energy Consumption Monitoring of Wireless Testbed Infrastructure through the NITOS EMF Framework**, in *the proceedings of the ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH), Mobicom, 2013*, ACM.

[D10] G. Kazdaridis, S. Keranidis, H. Niavis, T. Korakis, I. Koutsopoulos and L. Tassiulas. **An Integrated Chassis Manager Card Platform featuring multiple sensor modules**, in *the proceedings of the EAI International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom), 2012*, EAI. **(Best Demo Award)**

In addition, our research efforts within the same period led to the following publications that are not directly related to this thesis:

Conferences

[C1] D. Stavropoulos, G. Kazdaridis, N. Makris, H. Niavis, I. Igoumenos, T. Korakis and L. Tassiulas. **Enabling experimentation in mobile sensing scenarios through 4G networks: the NITOS approach**, in *the proceedings of the European Conference on Networks and Communications (EuCNC), 2015*, IEEE.

[C2] S. Keranidis, G. Kazdaridis, N. Makris, T. Korakis, I. Koutsopoulos, L. Tassiulas. **Experimental Evaluation and Comparative Study on Energy Efficiency of the Evolving IEEE 802.11 Standards**, in *the proceedings of the ACM International Conference on Future Energy Systems (e-Energy), 2014*, ACM.

[C3] R. Gonzales, C. Guerrero, E. Ego, F. Matera, S. Keranidis, G. Kazdaridis and T. Korakis. **The TREND Experimental Activities on “green” Communication Networks**, in *in the proceedings of the IEEE Tyrrhenian International Workshop on Digital Communications (TIWDC), 2013*, IEEE.

[C4] G. Kazdaridis, S. Keranidis, A. Fiamegkos, T. Korakis, I. Koutsopoulos and L. Tassiulas. **Novel Metrics and Experimentation Insights for Dynamic Frequency Selection in Wireless LANs**, in *the proceedings of the ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH), MobiCom, 2011*, ACM.

Demonstrations

[D1] P. Tzimotoudis, S. Keranidis, G. Kazdaridis, P. Symeonidis and T. Korakis. **LoRa Mesh network experimentation in a city-wide testbed**, in *the proceedings of the ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WiNTECH), MobiCom, 2019*, ACM.

[D2] S. Keranidis, G. Kazdaridis, V. Passas, G. Igoumenos, T. Korakis, I. Koutsopoulos and L. Tassiulas. **NITOS Mobile Monitoring Solution: Realistic Energy Consumption Profiling of Mobile Devices**, in *the proceedings of the ACM International Conference on Future Energy Systems (e-Energy), 2014*, ACM.

[D3] V. Maglogiannis, D. Giatsios, G. Kazdaridis, T. Korakis, I. Koutsopoulos and L. Tassiulas. **Integrating sensor measurements through CM cards as an OMF service**, in *the proceedings of the EAI International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom), 2012*, EAI.

[D4] A. Apostolaras, K. Choumas, I. Syrigos, G. Kazdaridis, T. Korakis, I. Koutsopoulos, A. Argyriou and L. Tassiulas. **A Demonstration of a Relaying Selection Scheme for Maximizing A Diamond Network's Throughput**, in *the proceedings of the EAI International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom), 2012*, EAI.

[D5] D. Stavropoulos, G. Kazdaridis, T. Korakis, D. Katsaros and L. Tassiulas. **Demonstration of a Vehicle-to-Infrastructure (V2I) Communication Network featuring Heterogeneous Sensors and Delay Tolerant Network Capabilities**, in *the proceedings of the EAI International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom), 2012*, EAI.

[D6] H. Niavis, G. Kazdaridis, T. Korakis and L. Tassiulas. **Enabling sensing and mobility**, in *the proceedings of the EAI International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom), 2012*, EAI.

[D7] G. Kazdaridis, S. Keranidis, A. Fiamegkos, T. Korakis, I. Koutsopoulos and L. Tassiulas. **Dynamic Frequency Selection through Collaborative Reporting in**

WLANs, in *the proceedings of MobiCom 2011, Las Vegas, Nevada, USA, September 2011*, ACM.

Acknowledgements

As this long endeavor has reached its end, I would like to express my deep gratitude to a group of people who provided me with guidance, assistance, support, and encouragement. It is certain that without their help, I would not have been able to complete my PhD studies.

First and foremost, I would like to deeply thank my supervisor, Assoc. Prof. Thanasis Korakis, as well as Prof. Leandros Tassiulas, who have provided me with guidance, motivation and support throughout the years of my studies in University of Thessaly, and gave me the opportunity to work in such an inspiring research team. Thanks to their continuous support and my exposure to different research ideas and projects, I was able to broaden my knowledge in different aspects. I am also grateful to the members of my dissertation committee, Assoc. Prof. Dimitrios Katsaros, Assist. Prof. Antonios Argyriou, Assoc. Prof. Gerasimos Potamianos, Assoc. Prof. Spyridon Lalis, Prof. Symeon Papavassiliou and Assist. Prof. Anna Tzanakaki for accepting to serve in the examination committe of my thesis. Moreover, I would like to thanks the Assoc. Director of the WINLAB, Ivan Seskar, for sharing his fruitful throughts and ideas with me. To Assist. Prof. Navid Nikaein I want to express my sincere thanks for guiding me through the early implementations of the thesis during the first year.

I would also like to thank all my friends and lab-mates in NITLab for their support and collaboration everyday. Especially, I would like to thank Stratos Keranidis with whom I have been working and collaborating since the very beginning in the lab, and has provided me with guidance and motivation throughout all the years of my studies. Also I would like to thanks the guys that we were working closely together, Polychronis Symeonidis, Nikos Sidiropoulos, Ioannis Zografopoulos, Panagiotis Skrimponis, Panagiotis Tzimotoudis, Vasilis Maglogiannis, and Stavros Ioannidis. Of cource, I would like to thanks Apostolis Apostolaras, Kostas Choumas and Nikos Giallelis for their support and guidance in the first years of my studies. Also the rest of the guys, Harris Niavis, Pavlos Basaras, Donatos Stavropoulos, Nikos Makris, Virgilios Passas, Ilias Syrigos, Kostas Chounos, Dimitris Giatsios, Antonis Kalkanof, Giannis Moutsinas, Ellie Eustathiou, Ioannis Igoumenos as well as Nicolas Barati and Fraida Fund.

My sincere thanks to my closest friends, Nikos, Christina, Thanasis, Dimitris, Adamantios and Kostis for their encouragement all these years. They have been at my side for more than twenty years now, always encouraging and supporting each other in every aspect.

Lastly, I want to thank my family, my mother Eleni, my father Kyriakos, and my brothers Fanis and Giorgos. I have no words to express my gratitude for them, their support, patience and unconditional love for all my decisions.

Dedicated to my aunt Georgia Kazdaridou.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation & Problem Statement

Energy efficiency is a major topic of research in the community of *Wireless Sensor Networks* (*WSNs*). In most real-world applications, sensors are battery operated, facing the inherent constraint of life duration that is solely dependent on the battery's remaining charge and the node's power profile. Fortunately, some deployment scenarios allow for battery replacement, which is a demanding procedure nonetheless and increases the maintenance overhead which evidently is a demanding procedure and raises the attendance cost of the network. On the contrary, it is not feasible to replace the batteries of nodes that are buried under the asphalt [211] to monitor available parking slots, or ones built into houses during construction [41] to allow for smart-home monitoring. The above highlight the requirement for further improvements in the sensors' power consumption profile to operate on a single battery charge.

Of course, considerable research work have been carrying out in several energy-efficiency directions regarding *WSNs*. For instance, in the energy harvesting domain several works suggest significant lifeduration improvements in specific *WSNs* applications. Despite the fact that latest products [70] illustrate significant advancements in the aforementioned domain, it is not yet possible to drive power-hungry sensor entities, towards achieving services of *10-20 years*. This stems from the fact that small-size transducers, in the size of a few cm^2, generate only a few μWatts or even nWatts [190], unable to power typical sensing devices. Notably, such types of transducers do only drive ultra-low power sensors with quite limited capabilities (e.g. sporadic transmission of data packets, limited number of peripherals, etc.). Several other directions endeavor to reduce the energy consumption of sensor devices by applying dynamic voltage scaling techniques [162, 160, 126] and/or by varying the operating frequency [6] of the host sensor's microcontroller, in order to save as energy as possible, but still energy savings potential in the *WSNs* remains huge.

A common approach for saving energy in sensor networks is the *duty-cycle* concept [125], since overhearing and idle listening are major sources of energy wastage [261]. As a matter of fact, current consumption in idle state is roughly equal to the energy required for receiving a packet through the radio. To this end, sensor nodes are configured to enter a low-power mode, the so-called sleep state, in order to save as much energy as possible during their inactive periods. The sleep state is interrupted by short, burst events, where sensors sense, process and propagate data. It might seem reasonable to neglect energy consumption in the sleep state, since more than three orders of magnitude separate current consumption in sleep and active states [94, 189]. However, given the fact that typical sensor applications operate at quite low *duty-cycles* ranging from *0.01 %* to *1 %* [94], it is expected that both states account for the systems' power budget expenditure [142]. Notably, sleep current is usually in the order of a few *μA*, which suggests that substantial energy savings can be attained. For example, consider a sensor node that draws *15 mA* on average when in active state and *5 μA* when in sleep. Assume also that the node features an available battery capacity of *200 mAh* and it is configured to operate in a duty-cycle fashion of *0.02 %*. The described sensor application will last for roughly *2.4 years* [177]. Consider now that a new power mechanism is employed that drops the drawn sleep current to *50 nA* (two orders of magnitude lower). In this case the resulting lifetime will reach *6.4 years*, which is undoubtedly a tremendous improvement.

Of course, the recent advancements presented in the modern *MCUs* has led the power consumption of the *standby* state to quite low levels. However, most commercial or prototype sensor devices feature sleep currents of several μA, which suggests huge potential for further improvements. **So, a natural question, is how the architecture design of the existing sensing devices can be revised, towards achieving lower power dissipation as well as longer life duration of *WSNs*.**

Apparently, most sensing applications operate in duty-cycle fashion since this is a reasonable/ the predefined/ordinary way to cope with the requirements of specific application scenarios. However, this is not always the best practice. When in ducycycle, a sensor network must strictly follow a given interval (even if it is modified over the time), which suggests that all sensor nodes must switch to their active state at specific timestamps, in order to check for any pending request and/or to perform an extra sensing cycle (even when this is not required). Apparently, the duty-cycle concept significantly diminishes the idle-listening, however, does not completely overcome the energy wastage issue, while it comes at the price of high latencies (especially when low duty-cycles are employed), which is undesirible in many application scenarios. A smart method to overcome the duty-cycle concept is the adoption of an auxillary receiver the so-called *Wake-Up Receiver* (*WuR*), dedicated to notify the host sensor only when required, establishing an asynchronous wireless communication. This principle is presented in several research works [82, 140, 229, 147, 14, 76, 105, 171]. This receiver is actually an auxiliary circuit usually attached to the main sensing device in order to notify the latter to switch from it's sleep to it's active phase when required so. The *Wake-Up Receiver* is continously listening for any upcoming wake-up packet, the so-called *Wake-up Beacon* (*WuB*), while consuming ultra-low power, usually in the range of a few μA or even a few hundred nA. The sole capability of waking up neighbors, however, would not provide the major improvements required by new critical applications, as nodes not interested in communications will be up wasting energy in overhearing and contentions. Thus, wake-up architectures allow the awakening of specific nodes (selective awakenings), which brings new requirements and challenges as well.

Despite the fact that the *WuRs* seems as the ideal solution for eliminating the idle-listening problem, the developed radios are characterized by very low sensitivity thresholds, therefore they can just support shorth communication ranges. The aforementioned receivers can be fully passive, including only passive circuits which are powered by the harvested RF energy, semi-active, combining passive and active circuits, or active, integrating actively powered components. When reviewing the passive *WuRs* schemes we observe that the proposed receivers can support a sensitivity around -25 dBm, unable to support communication ranges over a few meters. The semi-active implementations illustrate a fair trade-off between the supported sensitivity, typically around -40 dBm to -55 dBm, and the power dissipation which is in best case scenario is roughly 600 nA. On the other hand, active implementations may support high sensitivities, around -80 dBm, however, they are characterized by high current consumption which is inpractical for the creation of long-lived sensor networks. Therefore, among the three categories we distinguish the semi-active case. **The question that is raised, is how to develop a WuR with even higher sensitivity threshold, surpassing the obtained -55 dBm [140], while retaining the power consumption of the proposed receiver at the same power levels (few hundred nA).**

Evidently, energy-efficiency in the domain of Internet-of-Things (IoT) dominates the interest of the research community. Further improvements in the sensors' power profile will increase the life duration of the network, which requires realistic feedback from in-situ power meters. To this end, several works [94, 64, 133, 268, 205, 57, 187, 154, 186, 121, 119, 108, 110, 218, 89, 124, 188, 27, 226, 244], offer real-time monitoring to assist in the development of energy-efficient algorithms.

Recent IoT systems, integrate state-of-the-art elements such as RF radios, power regulators, sensing modules and micro-controllers, that feature improved power characteristics, as well as various operating and sleep modes. As a result, modern devices feature extremely low currents when switching to their sleep state conserving as much energy as possible. For instance, the Waspmote [1] and the eZ430-RF2500 [68] motes consume 860 nA and 690 nA respectively in their sleep state, while they draw several mA when active. Similarly, the ICARUS [112] mote exhibits a wide power profile dissipating only 22 nA in it's quiescent state. Another prime example with significant power consumption deviations between it's active and sleep state is the double-dip energy-harvesting system [148]. Although double-dip's consumption when active is typically in the range of few mA, in it's quiescent state power consumption diminishes to only 700 nA. Identical behavior is observed by other energy-harvesting devices such as the [262, 40]. Apart from the aforementioned IoT systems as whole, their power supply sub-circuits are equally intriguing featuring wide dynamic currents that may vary from a few nA to a several mA. Especially, in the case of RF energy-accumulators the harvested currents may be in excess of a few nA [45], even in close transmission distances. Similarly, the wake-up radio in [230] features ultra-low consumption dissipating roughly 600 nA when inactive and again severla mA in the active state.

All the aforementioned scenarios suggest that the power profile of modern sensor nodes exceeds at least three orders of magnitude. Therefore, sophisticated tools with wide dynamic range spanning the entire spectrum of possible current draws are required to capture and characterize the power profile of *IoT* devices. **So the question in this field, is how can we develop power consumption monitoring tools to cope with the dynamic range illustrated by the modern IoT devices in order to obtain accurate consumption measurements. Moreover, how it is possible to measure ultra-low currents in the order of a few nA with high accuracy?** Notably, it might be trivial to monitor mA loads, but when it comes to acquire nA measurements the electronics that must be employed for, must be carefully chosen so as not to distort the obtained signal.

A key enabler for studying the behavior of *IoT* technologies is the creation of a large-scale setup that consist of several individual nodes deployed under realistic conditions, in order to extract as accurate insights as possible. Ideally, the aforementioned setup must be remotely configurable in order to support the execution of different scenarios to aid in the assesment of the under consideration technologies. Over the last years new IoT Low-Power WAN (LPWAN) technologies, such as the NB-IoT, the LoRa and the LoRaWAN have been introduced that promise exceptional communication ranges due to their extremely high sensitivity thresholds. All these technologies use a physical layer that trades throughput performance, supporting from hundreds of bits to a few kilobits per second, to provide long range communication in the order of tens of kilometers. Moreover, they are characterized by low-power consumption which makes them ideal for a city-scale deployments.

LoRa [8] is one prominent technology of this type, employing a specific radio layer based on the *Chrip Spread Spectrum* (*CSS*) modulation. This standard allows the configuration of four critical parameters, *Bandwidth* (*BW*), *Spreading Factor* (*SF*), *Coding Rate* (*CR*) and *Transmit Power* (TX_P). Despite the huge potential that the LoRa standard exhibits, it has been insufficiently researched and studied. **How can we evaluate the behavior of the LoRa technology under realistic conditions, while also to characterize the power consumption profile of the LoRa chipsets?**

Driven by recent technological advances, mobile devices become increasingly sophisticated and miniaturized. Tablets, smartphones and various wearable devices, with embedded cameras, microphones, accelerometers, GPS and other types of sensors, are now part of everyday life. Carried by people everywhere they go, such devices can record a wealth of data, as well as give rise to new applications. For instance, the glasses launched by Google allow users to take photos and videos on the go, just by issuing voice commands. Along the same lines, smart wristwatches or clothes can infer the user's physical activity or detect accidents.

But apart from benefiting the individual, mobile and wearable sensors and information devices can also be useful for the society as a whole. In the spirit of so-called participatory sensing [36], people may contribute to a common goal by recording and publishing information on a voluntary basis, without even knowing each other; for instance, overflowed garbage bins or street potholes can be reported as part of daily commuting activities. More generally, one can view people with their smartphones and wearable devices as a mobile ad-hoc sensing infrastructure, which can be employed, in an opportunistic or targeted way, to perform large-scale sensing tasks. Note that the coverage and/or density that can be achieved this way could very well surpass that of any planned/fixed sensing infrastructure; in particular, since many governments and local authorities cannot afford to setup, operate and maintain nation-wide or even city-scale sensor networks.

At the same time, there is a rising need for the research community but also companies (in particular *SMEs*) to conduct experiments and pilot deployments in the real world, as opposed to a controlled lab setting. This holds even more for *Internet-of-Things* and crowdsensing scenarios, where real people with real behaviors and real mobility patterns are required to evaluate different approaches under realistic conditions at a large scale. However, while a lot of work has been done on testbeds that can be used remotely by researchers to run experiments on fixed wireless sensor networks, much less has been done to support experimentation with mobile sensor networks. **How can we exploit the mobility patterns already provided by the individuals in order to aid in the field of participatory sensing under realistic conditions?**

Another rapidly evolving research thrust is the Underwater Networking that aims at enabling the exploitation and monitoring of the vast natural resources existing in the undersea environment that covers more than 70% of the earth's surface. The key communication technology that enables these applications in the extreme underwater environment, is based on the propagation of Wireless Acoustic Signals. The recent advances in acoustic modem and sensor technologies have motivated the deployment of an increasing number of underwater sensors. Parallel to these technological advances, research has moved into the area of networking, resulting in the deployment of realistic experimentation platforms.

In this context, the *SUNRISE* [234] is the first project offering open experimentation facilities, integrating physical systems with software development into the *"Internet* *of Underwater Things"*. SUNRISE provides a wide set of different underwater hardware device types that enable experimentation in sea environments. The SUNSET framework [182], developed by SENSES Lab [220], provides all core functionalities for implementing *Underwater Sensor Network* (*UWSN*) protocols over multiple layers, considering also various underwater hardware devices. SUNRISE also employs *Autonomous Unmanned vehicles* (*AUVs*) [149] to perform underwater survey missions such as detecting submerged wrecks or mapping the structure of seafloor. Of course, *AUVs* also embed a vast number of environmental sensors to measure the concentration of different elements at various points, while they support wireless communication with the shore equipment over acoustic modems. Despite the recent advances in the field of realistic UWSN testbed evaluation, experimental assessment of proposed schemes in terms of energy efficiency still lags behind. Currently, evaluation of energy consumption is only supported in the Simulation framework of *SUNSET*, thus failing to address the complexity of real scenarios and to support the high level of heterogeneity that characterizes *UWSN* hardware. **So the question raised in this field is how can we augment the developed AUVs towards providing power consumption measurements under realistic conditions.**

The unleashed potential of the ocean ecosystem has raised the interest of the research community, towards understanding and exploiting the vast resources existing in the aquatic environment Towards this direction scientists all over the globe deploy permanent observatory systems (buoys) in marine waters in order to acquire long-term environmental data in real-time. This is also highlighted by the fact that several initiatives [53, 79] aim at establishing Earth observation services, freely accessible and at a large scale. Despite the recent technological advances and the need for real-time, in-situ monitoring the aforementioned observatory systems still present some inefficiencies. One key parameter of such systems is their backbone communication link, which is used for measurements offloading as well as for configuring the sensing setup. Typically, observatory systems exploit either WiFi or GSM technologies. The first is widely used when a system is deployed in a coastal zone, where it is likely to experience line-of-sight with a shore gateway, while the second one is mostly used in far-located systems. Both technologies are targeted at specific applications, while the latest protocols present improved features in terms of achievable throughput and power consumption. However, the recently introduced *IoT Low-Power WAN* (*LPWAN*) technologies, such as the *NB-IoT*, the *LoRa* and the *LoRaWAN* have not yet been integrated in such installation, while they seem to be ver attractive solutions. **The question is how the aforementioned low-power technologies can be exploited in such installations for aiding in the monitoring of the aquatic environment.**

1.2 Thesis Synopsis

In this thesis, we study modern principles adopted by sensing devices towards achieving energy-efficient operation. We revise the proposed principles, proposing novel strategies for further reducing the power draw in sleep state in an effort to prolong the life duration of sensor networks. Moreover, we develop power consumption monitoring tools in order to evaluate the developed sensing devices in terms of energy-efficiency. Lastly, we illustrate real-word deployments and experimentation scenarios to evaluate the performance of IoT devices in realistic scenarios. In the following paragraphs, we provide a summary of the works included in this thesis, towards addressing the questions presented previously.

Initially, in **Chapter 2** we illustrate a novel power management architecture towards eliminating the power draw of *duty-cycled IoT* sensing devices during inactive periods. Our principle suggests the employment of an off-chip *Real-Time-Clock* (*RTC*) configured to control the power supply of the under consideration mote, by enabling or disabling it's power in a *power-gating* fashion. The selected *RTC* features an ultra-low power profile and it is the only module that remains powered during sleep, hence the overall mote's consumption is substantially diminished. Additionally, we introduce an alternative topology in which the host *MCU* remains powered in sleep state while the *power-gating* method is applied only in the rest of the peripherals of the *IoT* node, in an effort to exploit the *MCUs* benefits such as *RAM* retention and ultra-fast wake-ups. The proposed principle can be adopted by any commercial or prototyping *IoT* mote, in order to extend the life expectancy of battery-powered applications, by pushing sleep currents an order of magnitude lower. We remark that the cost of the proposed system is less than 3 euros, while the complexity of the suggested circuitry is trivial. Moreover, we demonstrate the *ICARUS* mote, the first sensor that draws a sleep current of only *22 nA* on a *3 V* supply. Direct comparison of power draw in *sleep state* with *state-of-the-art* sensors illustrates improvements of roughly *98 % - 99.8 %*, while we demonstrate that the life expectancy of the same motes can be prolonged from *2.7 years* to *19 years* under specific *duty-cycles*.

Following this, in **Chapter 3** we focus on asynchronous awakenings in sensor networks towards surpassing the limitations presented in *duty-cycled* applications. To this end, we propose a new scheme for semi-passive *Wake-Up Receiver* circuits that attains remarkable sensitivity beyond *-70 dBm*, while the state-of-the-art receivers illustrate sensitivity of up to *-55 dBm*. The receiver employs the typical principle of an envelope detector that harvests RF energy from it's antenna, while it employs a nano-power operation amplifier to intensify the obtained signal prior to the final decoding that is realized with the aid of a comparator circuit. It operates at the 868 MHz ISM band using *OOK signals* propagated through LoRa transceivers, while also supporting addressing capabilities in order to awake only the specified network's nodes. The power expenditure of the developed receiver is as low as *580* *nA*, remaining at the same power consumption levels as the state-of-the-art implementations.

The above implementations present wide dynamic range in their power consumption profile, thus accurate power meters able to measure as low as a few nA are required for their evaluation in terms of energy-efficiency. To this end, in **Chapter 4** we introduce the eProfiler, a novel in-situ system for measuring the power expenditure of sensing devices that illustrate a wide current range. Our meter features an auto-ranging shunt-resistor switch that supports ultra-fast alternations of *10 ns*, formed with the aid of high-speed comparators, that can successfully adapt to any sharp transition. Moreover, a high-speed *Analog-to-Digital Converter* (*ADC*) with multiple inputs is employed to simultaneously monitor the shunt-resistors, delivering a speed of *150 kSamples/s* at *16 bit resolution*. The proposed meter can be leveraged for the real-time, as well as the long-term monitoring of IoT devices with it's cost being less than *90 euros*. Furthermore, the proposed system can actuate or trace state alternations on *I/O pins* with a maximum delay of *2 μs*, in order to provide correlation capabilities with the obtained power measurements. The eProfiler features a wide dynamic range of *1.000.000:1* while also being able to monitor currents of a few *nA*. The average obtained error of the meter is *0.45 %*, with a maximum error of *1.6 %*.

Next in, **Chapter 5**, we present a *LoRa* based city-scale testbed that employs several sensing devices scattered across the urban area to characterize air quality and weather conditions in real-time. The installation is augmented through a custom link quality evaluation framework that continuously monitors the *Packet Delivery Ratio* versus *RSSI* relation to characterize the performance of the *LoRa* standard under realistic conditions. Experimental results, collected over a period of 2 months, efficiently analyze *LoRa's* performance across a wide range of protocol configurations. Finally, we also present in-lab experiments that characterize the efficiency of *LoRa* modules in terms of power and *energy efficiency per bit*, along with valuable insights aimed at the development of energy efficient protocol improvements.

In **Chapter 6**, we present the *NITOS BikesNet* platform, a city-scale mobile sensing infrastructure that relies on bicycles of volunteer users. The *BikesNet* employs a custom-built embedded node that can be equipped with different types of sensors, and which can be easily mounted on a bicycle in order to opportunistically collect environmental and *WiFi* measurements in different parts of the city. Experimenters can remotely reserve and control the sensor nodes on bicycles as well as collect/visualize their measurements via the *OMF/OML* framework, which was extended in order to handle the intermittent connectivity and disconnected operation of the mobile nodes. We also provide a performance analysis of our node prototype in terms of sensing latency, end-to-end data transmission capability and power consumption, and report on a first experiment that was performed using the *NITOS BikesNet* in the city of Volos, Greece.

Chapter 7 focuses on the domain of the underwater sensor networks, in which energy efficient operation is of crucial importance. To this end, researchers and equipment vendors require in-depth understanding of the power consumption characteristics of underwater hardware when deployed in-field. In this work, we introduce the *EVERUN* power monitoring framework, consisting of hardware and software components. More specifically, we integrated the developed system into the *Autonomous Underwater Vehicles* (*AUVs*) of the *SUNRISE underwater testbed* towards evaluating their operation in terms of energy efficiency. Through the execution of a wide set of experiments under realistic conditions, we highlighted the limitations of model-based energy evaluation tools and characterized the energy efficiency performance of key protocols and mechanisms. The accuracy of the collected power data, along with the interesting derived findings, verified the applicability of our approach in evaluating the energy efficiency performance of proposed solutions.

Finally, in **Chapter 8** we present a novel architecture for enabling on-line communication with marine environment monitoring deployments. To this end, we rely on a set of communication technologies that range from *IoT* related low data rate communication standards to the widely adopted *WiFi* and *LTE* protocols that are able to support bandwidth demanding applications. To achieve energy-efficient operation we turn off all the power-hungry interfaces and peripherals exploiting the power-gating method, while we retain active a low-power long-range interface, dedicated to control the rest of the components. We present the installation of the developed system in the Vida oceanographic buoy, in Slovenia and evaluate our device in terms of power consumption.

Chapter 2

Nano-Things: Pushing Sleep Current Consumption to the Limits in IoT Platforms

Contents

2.1 Introduction

Energy efficiency is a major topic of research in the community of *Wireless Sensor Networks* (*WSNs*). In most real-world applications, sensors are battery operated, facing the inherent constraint of life duration that is solely dependent on the battery's remaining charge and the node's power profile. Fortunately, some deployment scenarios allow for battery replacement, which is a demanding procedure nonetheless and increases the maintenance overhead which evidently is a demanding procedure and raises the attendance cost of the network. On the contrary, it is not feasible to replace the batteries of nodes that are buried under the asphalt [211] to monitor available parking slots, or ones built into houses during construction [41] to allow for smart-home monitoring. The above highlight the requirement for further improvements in the sensors' power consumption profile to operate on a single battery charge.

Of course, considerable research work have been carrying out in several energy-efficiency directions regarding *WSNs*. For instance, in the energy harvesting domain several works suggest significant lifeduration improvements in specific *WSNs* applications. Despite the fact that latest products [70] illustrate significant advancements in the aforementioned domain, it is not yet possible to drive power-hungry sensor entities, towards achieving services of *10-20 years*. This stems from the fact that small-size transducers, in the size of a few cm^2, generate only a few μWatts or even nWatts [190], unable to power typical sensing devices. Notably, such types of transducers do only drive ultra-low power sensors with quite limited capabilities (e.g. sporadic transmission of data packets, limited number of peripherals, etc.). Several other directions endeavor to reduce the energy consumption of sensor devices by applying dynamic voltage scaling techniques [162, 160, 126] and/or by varying the operating frequency [6] of the host sensor's microcontroller, in order to save as energy as possible, but still energy savings potential in the *WSNs* remains huge.

A common approach for saving energy in sensor networks is the *duty-cycle* concept [125], since overhearing and idle listening is a major source of energy wastage [261]. As a matter of fact, current consumption in idle state is roughly equal to the energy required for receiving a packet through the radio. To this end, sensor nodes are configured to enter a low-power mode, the so-called sleep state, in order to save as much energy as possible during their inactive periods. The sleep state is interrupted by short, burst events, where sensors sense, process and propagate data. It might seem reasonable to neglect energy consumption in the sleep state, since more than three orders of magnitude separate current consumption in sleep and active states [94, 189]. However, given the fact that typical sensor applications operate at quite low *duty-cycles* ranging from *0.01 %* to *1 %* [94], it is expected that both states account for the systems' power budget expenditure [142]. Notably, sleep current is usually in the order of a few μA, which suggests that substantial energy savings can be attained. For example, consider a sensor node that draws *15 mA* on average when in active state and *5 μA* when in sleep. Assume also that the node features an available battery capacity of *200 mAh* and it is configured to operate in a duty-cycle fashion of *0.02 %*. The described sensor application will last for roughly *2.4 years* [177]. Consider now that a new power mechanism is employed that drops the drawn sleep current to *50 nA* (two orders of magnitude lower). In this case the resulting lifetime will reach *6.4 years*, which is undoubtedly a tremendous improvement.

Undoubtedly, the recent advancements presented in the modern *MCUs* has led the power consumption of the *standby* state to quite low levels. However, most commercial or prototype sensor devices feature sleep currents of several μA, which suggests the potential for further improvements. So, a natural question, is how the sleep state consumption of the existing devices can be revised, towards achieving lower power draw and apparently longer life duration of *WSNs*.

In this chapter we introduce a novel mechanism to eliminate power draw in sleep state, while we demonstrate it's performance and it's applicability. The key contributions are outlined:

- we present an innovative energy management architecture that eliminates the current consumption in sleep state in *duty-cycling* applications
- we present the *ICARUS* mote, the first device that features an outstanding current draw of *22 nA* in sleep state
- we evaluate the proposed system in terms of power consumption and wake-up performance
- we compare the life duration of indicative motes when adopting our principle versus when using standard features, noting substantial lifetime extensions

  Notably, the presented principle can be applied in conjunction with other systems for intermittent operation [**hester**] or with IoT operating systems [**riot**], to provide energy efficient operation.

The remainder of the chapter is organized as follows. Section 2.2 discusses preliminary notions and power saving strategies regarding the motes' standard operation. Section 2.3 reviews the related work. System components and implementation are described in section 2.4, while the system's performance and evaluation in section 2.5. Moreover, section 2.6 compares the performance of indicative platforms when the proposed principle is applied. In section 2.7 we present a similar principle employing an ultr low power timer circuit fo and section 2.8 concludes the chapter.

2.2 Preliminaries

In this section we introduce key notions and principles used in low power systems, in order to ease the presentation of the concept.

Duty-cycling in sensor systems is realized by disabling as many parts as possible during inactive periods to conserve energy. Commonly, on-board modules and peripherals feature a low-power state where they consume as little power as possible, while the host *MCU* is responsible for managing the whole process. Of course, the host *MCU* also enters a sleep state to save energy. Actually, modern *MCUs* feature a number of low-power modes, ranging from *light-sleep* or *standby* mode to complete *shut-off* [142].

MCU blocks, as well as their interconnected peripherals, are sourced by different clocks which operate at variable frequencies. These clocks are progressively disabled, according to each low power mode's constraints, in favor of better energy consumption. For instance, *TI's MSP430* family supports several lower power modes, the so-called *LPMs*. The clock that each module (e.g. *ADC, Comparators, etc.*) will be sourced from can be set, making it usable in more energy efficient *LPMs* albeit with a lowered performance. Evidently, the wake-up time is also affected by the *LPM* being used. However, the time required for an *MCU* to boot when power is initially supplied is referred to as *cold-start* time, which is always longer than any other wake-up time.

The lowest possible state in terms of power draw is the *shut-off* / *deep-sleep state*, in which the *MCU* turns off completely, while only the minimum functionality required to restore the *MCU* back to active state from an external signal is preserved. This mode reduces the power consumption to an absolute minimum, in some microcontrollers as low as *20 nA*. Apparently, this mode requires an external stimulus, thus is it not often opted for in *duty-cycled* systems. When considering *duty-cycled* schemes, the so-called *standby* state is employed. In this state *MCUs* may preserve only a time-keeping circuit active, to provide the required interrupts in the given intervals.

The typical time-keeping circuits integrated into *MCUs* are the *Watchdog timer* (*WDT*) and the *Real-Time Clock* (*RTC*). The *WDT* is a specific guard timer used to detect and recover the *MCU* after a malfunction occurs. Commonly, it draws more power than the *RTC* while it has more limitations in the supported intervals and the supported time accuracy, hence the *RTC* is usually preferred to provide the interrupt stimulation. However, several *MCUs*, such as the *ATmega* family, do not incorporate an *RTC* circuit to allow for wake-ups, hence *WDT* remains the only option. The *RTC* is a time-keeping circuit used in a wide range of systems. It is running over either a crystal oscillator or a relaxation oscillator, usually at low speed clocks, resulting in low power draw. Additionally, it offers advanced time accuracy, which is crucial when considering synchronized wake-up schemes for sensor networks. The power draw of a *MCU* in *standby state*, when the *RTC* remains active can be anywhere from a few hundred *nA* to a few *μA*, substantially lower compared to a *WDT* timer.

Another consideration in *duty-cycled* sensor systems is the memory retention. Most *MCUs* support low-power modes that retain volatile memory contents by constantly supplying power to the module. Following the latter principle, *MCU* manufacturers provide memory retention capabilities which preserve the state of the executed program in an effort to resume the execution from the point it was left off. Apparently, this process is crucial in intermittent computational systems [82, 7], such as sensor networks in *duty-cycle* schemes, providing fast recover times. On the other hand, there are state retention schemes that utilize non-volatile memories that do not require power to retain their content, such as *Flash* or *EEPROM* [41]. Non-volatile memory is, however, significantly slower and more energy hungry than *RAM*. A modern type of non-volatile memory, *FRAM*, grants much higher access speeds than other types of non-volatile memory. Although it consumes marginally more power than *RAM*, it still remains a better alternative, consumption-wise, to *Flash* or *EEPROM*. Notably, *TI* integrates *FRAM* technology in it's *MSP430FR MCU* family [74].

Undoubtedly, the power profile of modern *MCUs* in the *standby* state has been been remarkably improved over the past years, however, the total consumption of sensor nodes is often marginally higher, since most sensor devices integrate a a vast number of external devices, e.g., sensing modules, an *RF* chip and all the requisite power electronic circuitry. All the aforementioned individual elements may also feature their own standby mode, in which they draw insignificant amount of power, however, the aggregated consumption is never negligible. In the context of low-power sensor design, all power expenditures should be accounted for, and to tackle this challenge, the *power-gating* [227, 62] technique is enforced. *Power-gating* suggests that peripherals that are not in use may be entirely disconnected from the power source, by employing a *load-switch* that is usually controlled by the *MCU*. The only sub-system that must always remain powered is the *MCU* that retains the *time-keeping* circuit. Apparently, this strategy is not often applied in the majority of sensor nodes. Even when adopted, the *standby* current will be roughly of a few *μA*, since the power regulator's quiescent current must be taken into consideration as well, which is a major source of energy wastage as also noted in [265].

2.3 Related Work

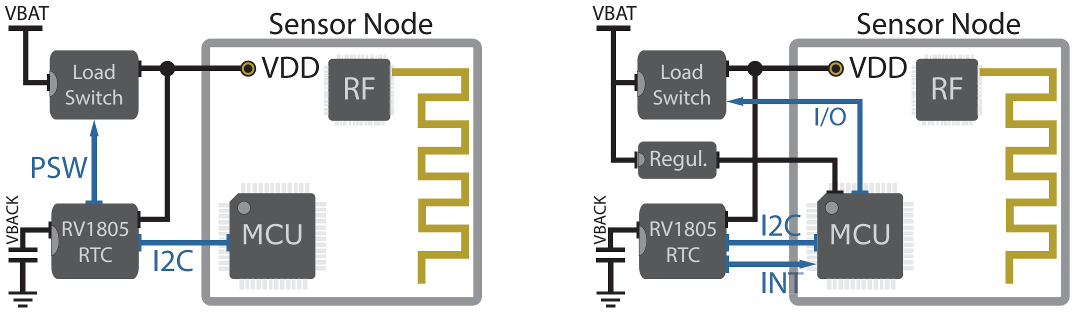
In this section we present the most widely adopted *IoT* devices along with their power characteristics, in order to provide a thorough comparison with our proposed system.

The *MicaZ* [155] and the *TelosB* [184] are considered to be [13] two of the most energy efficient platforms, that draw *15 μA* and *8.8 μA* respectively on a *3.3 V* supply, when in sleep mode. Both boards were designed more than a decade ago, and therefore cannot compete with the latest sensor developments anymore. The *Opal* [97] is a prototyping platform based on an *32-bit ARM Cortex-M3 MCU* that consumes roughly *8.9 μA* in sleep state. Apparently, it is a good example of a board that embeds a vast number of electronic peripherals as well as an extra wireless module, without adopting any *power-gating* scheme, thus it features such high power draw in spite of utilizing modern *ICs*. The *Storm* [13] is also a recently developed prototyping board that utilizes a powerful *32-bit ARM Cortex-M4 MCU*. Despite the fact that the authors claim the selected *MCU* can compete with the best-in-class IoT platforms in terms of energy efficiency, the resulted draw in sleep phase is *13 μA* and *2.3 μA* on *3.3 V* and *1.8 V* voltage rails respectively, which is not considered low. Notably, all the aforementioned sensor systems rely on the internal time-keeping functions of their host *MCUs*, in order to trigger wake-up signals when in sleep mode.

An alternative principle to provide wake-up interrupts is presented by the *XYZ* [136] and the *EcoBT* [254] sensor devices. These platforms suggest the integration of an off-chip *RTC* circuit, replacing the *MCU's* power consuming time-keeping functions, in order to provide external interrupts to wake-up the host node. This method suggests that the *MCU* will enter a *deep-sleep* state consuming much less power. However, the attained power draw is roughly *30 μA* and *2 μA* respectively, since the selected *ICs* do not feature power efficient *deep-sleep* modes.

A remarkable IoT platform in terms of power efficiency is the *TI's eZ430-RF2500* [68], which draws roughly *1 μA* in sleep. It's ultra-low power profile is attributed to the on-board *MCU* which draws just *600 nA* in it's *LPM3*, where it's internal *WDT* remains active to re-trigger the node when required. Notably, apart from the *MCU* the board features only the *CC2500* RF chipset, without integrating any other sensing module nor a voltage regulator, hence it is able to attain that compelling power draw. On the other hand, there is the *Waspmote* sensor [1] (version 1.2), which can be fitted with a vast variety of sensors featuring only *860 nA* in it's lowest sleep mode with wake-up capabilities. To achieve this remarkable performance all the on-board or attached modules are *power-gated* with the aid of *load-switches* that are controlled by the host *MCU*. Moreover, the *Waspmote* is outfitted with an off-chip *RTC*, the *DS3231*, that, along with a *load-switch*, cuts the power of the entire node. Since, the *DS3231* can only provide short interrupt signals, a latch circuit, formed by a monostatic multivibrator and a few logic gates, is employed to retain the state in order to smoothly drive the *load-switch*. It is worth noting that the *Waspmote* is a commercial platform that doesn't reveal the principle behind the wake-up implementation, but we reverse engineered the aforementioned circuit block to determine the components used and the architecture followed.

Apparently, the majority of the devices exceed *1 μA* in sleep state with just the *Waspmote* and the *eZ430* achieving better performance. In our work we follow a principle similar to the *Waspmote's* by utilizing an extremely low-power off-chip *RTC* module and configuring it to manage the power of the entire sensor node, achieving a current draw of as low as *22 nA* at *3 V* operation. A direct comparison between the *Waspmote* and our system, illustrates impressive reduction in power consumption, when, at the same time, our architecture does not require the complicated array of components used by the *Waspmote* to maintain the state of the alarm interrupt since our selected *RTC* features such capabilities. Lastly, we refer to our previous work [111] presenting a small set of the proposed principle, in which we employ the *TI's TPL5111*, an ultra-low power timer, instead of the proposed *RTC*. The *TPL5111* is used to provide the external stimulus featuring *33 nA* power draw. However, the time accuracy of the selected timer is utterly poor featuring *100 ppm* drift, while the selected *RTC* supports *2 ppm* accuracy, that allows the formation of synchronized wake-up events in distributed networks. Moreover, the proposed *RTC* can be easily re-synchronized to correct any time drift, while the aforementioned timer does not



(A) Proposed PSW Architecture Diagram

(B) Proposed Interrupt Architecture Diagram

FIGURE 2.1: Proposed Architecture Diagrams

support such capabilities.

It is worth noting that several other works aim at developing energy-efficient sensing devices in order to last as long as possible on a single battery charge. For instance, the ZebraNetwork [265, 135], the Camazotz [96], the [63] and several sensing devices such as the [256, 69, 128, 183, 264, 246, 166, 267, 217, 201, 200]. Lastly, we note several usefull articles with technical details regarding the design and development of low-power sensing devices [92, 255, 185, 37, 38, 12, 56, 98, 81, 127].

2.4 Nano Power System Implementation

In this section we describe the proposed architecture and we detail the technicalities of our system.

2.4.1 Niche Low Power Methodology

Typically, sensor nodes draw significant amounts of power when in sleep state, due to the poor power efficiency of their time-keeping circuits (regardless if they are incorporated in the *MCU* or not), power leakage issues and the fact that the aggregated consumption of the peripheral modules (sensors, *ICs*, etc.) is not negligible. In this work, we propose the employment of an on-board, off-chip *RTC* module with an ultra-low power profile, to manage the go-to-sleep and wake-up phases of battery powered *IoT* nodes. In essence, we replace the existing time-keeping circuit that consumes substantially more power, and we *shut-off* the entire node in order to alleviate any power expenditure. We propose two different topologies for utilizing an off-chip *RTC*, each one with different trade-offs.

PSW Topology: In the *PSW* (*Power SWitch*) case we employ a *load-switch* along with the *RTC*, to completely power-off the under-consideration sensor node adopting the *power-gating* [227] method, as illustrated in Fig. 2.1a. Practically, the *IoT* node is powered via the *load-switch* which is controlled through the *RTC*. This approach results in as low consumption as possible, since it eliminates any power draw induced by the sensor node, as only the *off-chip RTC* and the *load-switch* remain active. Evidently, this configuration induces noteworthy delays when considering the cold-start boot

Manufact.	Model	Consumption	Accuracy	Operating Volt.	Output
Renesas	*ISL12022*	*1 μA*	*3 ppm*	*2.7-5.5V*	*Interrupt*
Maxim Int.	*DS3231*	*840 nA*	*2 ppm*	*2.3-5.5V*	*Interrupt*
Microchip	*MCP79412*	*700 nA*	*10 ppm*	*1.8-5.5V*	*Interrupt*
Maxim Int.	*DS1307*	*500 nA*	*23 ppm*	*4.5-5.5V*	*Interrupt*
ST	*M41T62*	*350 nA*	*2 ppm*	*1.3-4.4V*	*Interrupt*
NXP	*PCF8563*	*225 nA*	*29 ppm*	*1-5V*	*Interrupt*
MicroCrystal	*RV3028*	*45 nA*	*1 ppm*	*1.1-5.5V*	*Interrupt*
MicroCrystal	*RV1805*	*17/22/60 nA*	*2 ppm*	*1.1-3.6V*	*Interrupt / PSW*
Abracon	*AM0805*	*15/22/55 nA*	*2 ppm*	*1.5-3.6V*	*Interrupt*

TABLE 2.1: Compelling RTCs and their Specifications

time, which is relatively longer compared to the wake-up time in standard operation. Given the above trade-off, the impact of the proposed principle depends on the characteristics of each node and the *duty-cycle* of the application, as described in section 2.6. It is worth noting that the proposed topology can benefit from the *FRAM* technology, to instantly return to the desired program state after a sleep period, substantially reducing the comprehensive overhead.

Interrupt Topology: On the other hand, in the *Interrupt* (*INT*) configuration the *MCU* remains constantly powered even in sleep state, while the off-chip *RTC* is employed to provide waking signals. This strategy suggests that the host *MCU* exploits the *deep-sleep* mode, in which it consumes only a few *nA*, in contrast to the typical power-consuming *standby* mode. Moreover, we apply the *power-gating* principle to the remaining electronics and peripherals of the *IoT* node with the aid of a *load-switch* in an effort to eliminate their power draw when asleep. It is worth noting that in this topology we also employ an external voltage regulator featuring extremely low quiescent current in order to power the *MCU*. The proposed architecture is illustrated in Fig. 2.1b. Despite the fact that in this topology the overall power draw is apparently higher compared to the *PSW* case, it still remains in the order of a few *nA* when considering modern *MCUs*. The crucial advantage of this approach is the fast wake-up time, comparable or even the same as in standby mode. Another asset of this scheme is that the host *MCU* can be configured at any low-power mode during sleep phase, in order to take advantage of any supported feature, such as the RAM retention capability that allows for incredibly fast wake-ups.

2.4.2 Components Selection

In this subsection we present the selected components detailing their characteristics and the exact wiring of the proposed system.

RTC: We reviewed several off-the-shelf *RTCs* and listed the more compelling ones along with their characteristics in Table 2.1. The most widely used in *IoT* devices is the *DS1307*, which features poor accuracy, while in some delicate applications we meet the *DS3231*. The latter provides sufficient accuracy of *2 ppm*, but consumes *840 nA*, while other candidates exhibit better performance, consuming a few hundred nA, but are barely seen in any *IoT* devices. In our system we opted for the

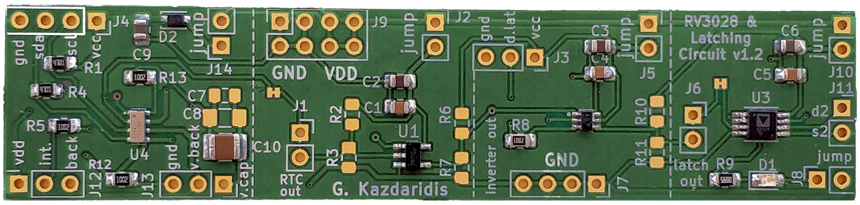


FIGURE 2.2: RV3028 RTC w/ Latching Circuit

Micro Crystal RV1805-C3 [158], which achieves similar performance with the *Abracon AM0805 RTC*, but also features a specially designed output able to directly drive external loads.

The *RV1805* supports two modes of operation, where different oscillators are activated each time. In the *XTAL* mode, a *32.768 kHz* clock is running featuring *60 nA* power draw, while the *RC* mode achieves worse accuracy but draws only *17 nA*. To improve the *RC* mode performance, *RV1805* enforces an auto-calibration mechanism, exploiting the *XTAL* crystal oscillator, in which case the average power draw is roughly *22 nA*. In our application scenarios we configure the *RV1805* in *RC* mode with auto-calibration every 512 seconds. The factory calibrated clock achieves a time accuracy of typically ± *2.0 ppm* at *25 °C*, while communication with the host *MCU* is attained over *I2C*.

The selected *RTC* features two types of output signal, an Interrupt pin (*nINT*) providing a short pulse to trigger external devices (lasting for *200 ms*) and a *stable-state* output pin (*PSW*) that changes state when triggered and remains in this second state until otherwise instructed. When advised so, the *PSW* output automatically returns back to it's original low-state awaiting for a second trigger pulse. In our implementation we utilize both the *nINT* and *PSW* pins to form the *Interrupt* and *PSW* schemes respectively. Notably, a *stable-state* signal is required to drive the *load-switch* in order to support the *PSW* topology.

The *RV1805* is essentially the only available *RTC* supporting a *stable-state* output (the *PSW* pin). In an effort to implement the proposed *PSW* configuration with alternative *RTC ICs* that do not support *PSW* output, we implemented a proof-of-concept prototype board based on the *RV3028 RTC*. This circuit, illustrated in Fig. 2.2, employs a *latch IC* to retain the state after triggered by a short interrupt signal. The prototype features the *NC7SU04 Inverter* and the *74AUP1G373 D-type latch IC*, which are ultra-low power and draw roughly *3nA* in total, substantially lower compared to the *Waspmote's state-retention* circuit that uses the *SN74LVC1G123 monostable multivibrator*.

Load Switch: Regarding the *load-switch* which we employ to implement the *power-gating* principle, there are several available off-the-shelf models. We distinguish the *TPS22860* and the *ADG821* featuring a quiescent current of roughly 0.3 nA, while supporting ultra-fast response times. Both switches drive loads of up to *200 mA*,

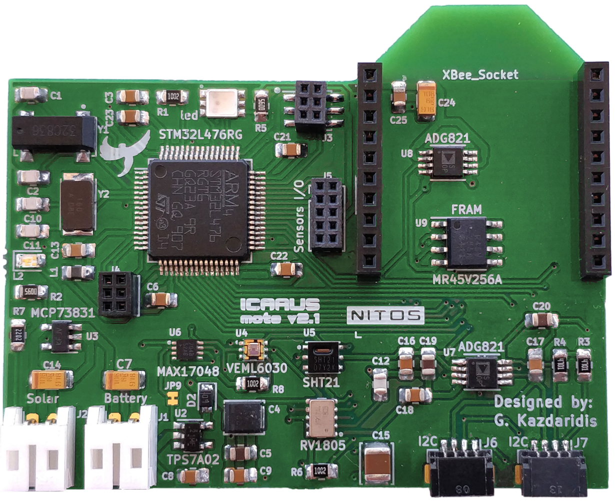


FIGURE 2.3: ICARUS Prototype Mote

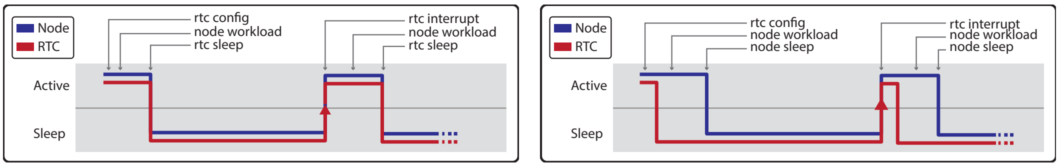
whereas higher load output switches can be employed when considering power-hungry *IoT* nodes. Alternatively, *MOSFET ICs* can also play the role of the load-switch.

It is worth noting that all the selected components are of low-cost (<5 euros in total), while the proposed circuit is characterized by low complexity, hence it can be easily adopted by any commercial or prototype *IoT* node. Especially when referring to the *PSW* topology, the required perturbation to an existing device is minimum.

2.4.3 Implementation Setup & Design

In this subsection we demonstrate the prototype *ICARUS* mote that encompasses the proposed system architecture and we explain the configuration of the system in each topology.

ICARUS Prototype Mote: The *ICARUS* mote, illustrated in Fig. 2.3, features the *STM32L476RG* which is an ultra-low power *ARM Cortex-M4 32-bit RISC core MCU* operating at a frequency of up to *80 MHz*. It embeds high-speed *Flash* memory of *1 MB* and an *SRAM* of *128 KB*. The mote integrates an *XBee-footprint* socket for plugging-in wireless interfaces, such as *LoRa*, *ZigBee*, *BLE*, etc. Moreover, it embeds the *SHT21 temperature & humidity*, the *VEML6030 light intensity* and the *MAX17048G+ battery gauge* sensors, while extra sensing modules can be interfaced through the available *I2C* and *I/O* ports. In addition, an off-chip *FRAM* memory, the MR45V256A, is assembled on the board that can be used for memory retention schemes even when the host *MCU* is not powered. Of course, the *ICARUS* also integrates the *RV1805 RTC*, which, along with the *ADG821 load-switch*, is used to switch the power of the mote entirely on or off, or to alternatively serve as an interrupt source for waking-up the host *MCU*. An extra *ADG821* is employed to control the power rails of the wireless interface and of the attached sensors and peripherals. Notably, each *ADG821* features two internal switches. The *ICARUS* exhibits roughly *22 nA* power draw in sleep state, exploiting the proposed mechanism.



(A) Activity in *PSW configuration*

(B) Activity in *Interrupt configuration*

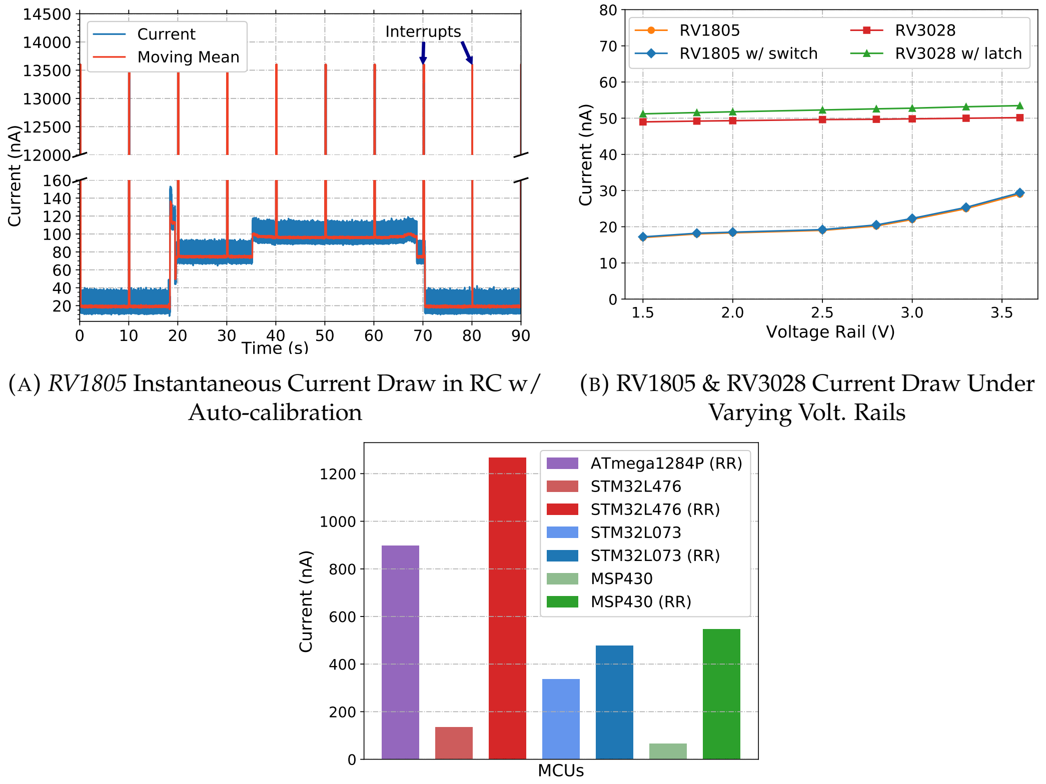
FIGURE 2.4: Proposed Architecture Diagrams

PSW Topology Configuration: In the case of the *PSW topology*, when the mote receives power for first time, it initially configures the *RV1805* with the appropriate settings and the desired time interval, as illustrated in Fig. 2.4a. The mote then performs it's typical workload and when it has completed it's tasks it issues a *sleep command* to the *RV1805*. Automatically, the *RTC* asserts it's output signal low (the *PSW* pin is employed) which, as a result, cuts the power from the entire node via the *load-switch*. Notably, the *RV1805* features a sleep state as well, in which it turns off the I2C interface and enters into a low-power state until the next cycle. After the specified by the application interval, the *RV1805's* timer fires up restoring power to the host sensor through the *load-switch*. Apparently, in this topology, *I2C* communication with the host *MCU* is required during every active period for the instruction of the *sleep command* to the *RTC* after the completion of the workload.

Interrupt Topology Configuration: In the second scheme, the *RTC* is again configured by the host *MCU* during the initialization of the sensor, as illustrated in Fig. 2.4b. However, in this approach the *RV1805* is set up to provide interrupt signals directly to the host *MCU* without requiring the reception of a *sleep command* by the host node as in the first approach. Therefore, after the initial configuration, the *I2C* channel may be re-established only for the re-synchronization of the *RV1805*, so as to support synchronized wake-up schemes in mesh networks. In this topology, the *load-switch* which provides power to the remaining electronics is controlled by the host *MCU* via an *I/O* pin.

RV3028's Equivalent PSW Topology Configuration: Lastly, we refer to the prototype developed based on the *RV3028*. The configuration of the *RV3028* is realized similarly to the *Interrupt topology*, providing short interrupt signals. An unbuffered inverter is used to convert the generated *active-low* interrupt signal to an *active-high* which is required by the utilized latch in order to change it's state from low to high and successfully drive the *load-switch*. The D-type latch features two input pins, the *clock* and the *data*. The *clock* input is connected to the inverter's output, while the *data* input is asserted high, so as to switch to high-state when triggered. When the node's workload is completed, the host *MCU* asserts the *data* pin low and instantly provides an interrupt pulse to the *clock* input in order to modify the state to low and cut the power off.

In an effort to test the proposed system with different *MCU* architectures we used



(A) *RV1805* Instantaneous Current Draw in RC w/ Auto-calibration

(B) RV1805 & RV3028 Current Draw Under Varying Volt. Rails

(C) Current Draw of Various MCUs in Interrupt Topology

FIGURE 2.5: RV1805 Instantaneous Current Draw & Power Expenditure of the Proposed PSW and Interrupt Schemes

the *ATMega1284p*, the*MSP430FR5969* and the *STM32L476 MCUs*. To enable communication between the selected *MCUs* and the proposed RTCs, we utilized a slightly modified version of *SparkFun's* library for the *ATMega* case, while we developed the corresponding libraries for the *MSP430* and *STM32L476*.

Regarding the power supply of the *RV1805*, it can be powered directly from a *3 V* lithium-manganese dioxide battery. However, when considering typical *IoT* applications, sensor nodes often feature *3.7 V* cells that output roughly *3.7* to *4.2 V*. In this case, a voltage regulator is required to provide lower voltage within the accepted range. However, this is not the best practice, since regulators typically feature quiescent current of a few μA. Only the *TI TPS62740* and the *TI TPS7A02* feature an outstanding quiescent draw of *360 nA* and *25 nA* respectively, which is certainly an exception not used by any known mote. In fact, most *IoT* nodes feature quite high sleep currents, for the sake of powering their *MCUs* and other peripherals with the appropriate voltage rail, commonly at *3.3 V*. Another way to power the *RV1805* without employing a voltage regulator is to exploit the back-up capacitor that it supports for such purposes. The *RV1805* features an internal circuit able to instantly charge it's back-up capacitor when power is applied, and automatically switch to this power source when power is disconnected. Practically, in our system the *RV1805* charges the capacitor in every active-cycle by the sensor's regulated rail, while it consumes zero power in sleep.

Lastly, we note that we employ the *TPS7A02* regulator to power the *MCU* in the sleep state when the Interrupt topology is applied.

2.5 Performance Evaluation

In this section we analyze the performance of the proposed system, evaluating both the derived power draw as well as the wake-up performance of a wide set of *MCUs*.

2.5.1 Power Consumption Evaluation

In this subsection we evaluate the power consumption profiles of the *RV1805* and the *RV3028* under the different modes and varying voltage rails; we also discuss the instantaneous power consumption of a sensor system when our principle is applied.

In our implementation we opted for the *RC mode* with auto-calibration, since in *XTAL mode* the *RV1805* draws significantly more energy (*60 nA*), while in *RC mode* (*17 nA*) without calibration the supported accuracy is fairly poor. The auto-calibration is performed either every *1024* or *512 seconds* and lasts for roughly *50 seconds*. Therefore, it is not trivial to calculate the average power draw of the *RV1805*. In our evaluation experiments we exploited the *uCurrent* meter [187], which is designed to measure ultra-low currents with a resolution of up to *1 pA*. However, *uCurrent* focuses on monitoring currents that remain stable for substantial duration, thus, we used it only to measure the level of each phase with high accuracy. For capturing transient phases we rely on our high-fidelity monitoring tools [108, 110]. Specifically, [110] features a dynamic shunt resistor switch that alternates depending on the flowing current, which eases the process of measuring the power profile of the *RV1805*.

Fig. 2.5a illustrates the obtained power results at *3.4 V* power supply. Notably, the red line represents the average power draw calculated as a moving mean over a window of *100* values so as to provide a more representative indication. In the illustrated experiment we configured the *RV1805* to provide alarm signals with an interval of *10 seconds*, while we measure it's power draw. We observe that when the *RV1805* is in idle state, the instantaneous power draw is roughly *19.2 nA*. At *18.4 seconds* the process of auto-calibration is initiated. At first the *XTAL crystal* powers up, stabilizes and at *20 seconds* the calibration function begins, lasting for *50 seconds*. The sudden power draw oscillations at *13.6 μA* that occurred every *10 seconds* are attributed to the interrupt signals generated as configured in this experiment. To determine the power draw of the *RV1805* we calculate the average value over a single auto-calibration period, without of course considering interrupt events in this time frame. The obtained results of the *RV1805*, plus the *RV1805* together with the

MCU	LPM w/ full RAM retention			LPM w/o retention			Cold-start
	cons. w/ RTC	w/o RTC	wake-up	cons. w/ RTC	w/o RTC	wake-up	
ATmega1284p	4.5 μA (WDT)	850 nA	0.1 - 65 ms	n/a	n/a	n/a	0.37 - 65 ms
STM32G473RE	81.5 μA	80.5 μA	9.5 μs	672.5 nA	65 nA	267.9 μs	-
STM32L476RG	1.72 μA	1.22 μA	7 μs	503 nA	88.5 nA	256 μs	3.9 ms
STM32L432KC	1.63 μA	1.15 μA	8.2 μs	482.5 nA	128.15 nA	261.5 μs	1.1 ms
STM32L073RZ	860 nA	430 nA	3.5 μs	590 nA	290 nA	60 μs	7.52 ms
MSP430FR5969	700 nA	500 nA	7 μs	350 nA	20 nA	1 ms	1 ms
MSP430F2274	600 nA (WDT)	100 nA	1 μs	n/a	n/a	n/a	2 ms
MSP432P401R	860 nA	700 nA	9 μs	630 nA	25 nA	1 ms	1 ms
ATSAM4LC8C	3.4 μA	2.3 μA	1.5 μs	1.5 μA	900 nA	1.5 ms	4.2 ms

TABLE 2.2: Compelling MCUs Characteristics in Different Low-Power States

ADG821 under varying voltage are presented in Fig. 2.5b along with the corresponding values of the *RV3028* with and without the proposed latching circuit.

The resulted draw of a sensor system in sleep state when employing the *PSW* topology is presented in Fig. 2.5b. Even when assuming the utilization of *FRAM* retention schemes, supported by the *MSP430 MCU* family, the power draw will remain the same, since *FRAM* does not require power to retain data. On the contrary, when considering the power consumption of a mote in *Interrupt topology* we must calculate the aggregate draw of the selected *RTC* and *load-switch* plus the power draw of the regulator and the *MCU* in the selected sleep state. Table 2.2 consolidates detailed characteristics of a wide-range of *MCUs* in various Low-Power Modes (*LPMs*), noting the draw of each state with and without a time-keeping circuit activated. Notably, the Table is separated into two main *LPM* categories, the obtained consumption with *RAM* retention enabled and without. For instance, the *STM32L476* draws *1.72 μA* with *RTC* enabled and *1.22 μA* without any time-keeping circuit, while supporting RAM retention in both cases. The same *MCU* draws *503 nA* with *RTC* and *88.5 nA* without, when not supporting RAM retention. Now considering a sensor node that features the above *MCU* while configured in Interrupt topology, it will consume roughly *1.27 μA* with RAM retention and *136 nA* without. In the same scenario, a sensor node featuring the *MSP430FR5969* will consume *547 nA*, with RAM retention and only *69 nA* with no retention capabilities. Fig. 2.5c summarizes the power draw of indicative *MCUs* in sleep state with and without *RAM* Retention (*RR*), when the *Interrupt* topology is applied.

It is worth noting that some *MCUs* feature quite high power draw when their internal time-keeping circuit is activated but achieve extremely low consumption when it is not. For example, the *STM32G473* if no retention is supported, consumes *672 nA* with the *RTC* enabled and only *65 nA* with the *RTC* disabled. Similarly the *MSP430F2274*, attains *600 nA* with the *RTC* and *100 nA* without, when *RAM* retention is supported. Apparently, in the latter scenarios it is deemed efficient to adopt the Interrupt topology, while for other *MCUs*, such as the *ATmega1284p*, that present extreme power draw (*850 nA*), the *PSW scheme* seems a better option.

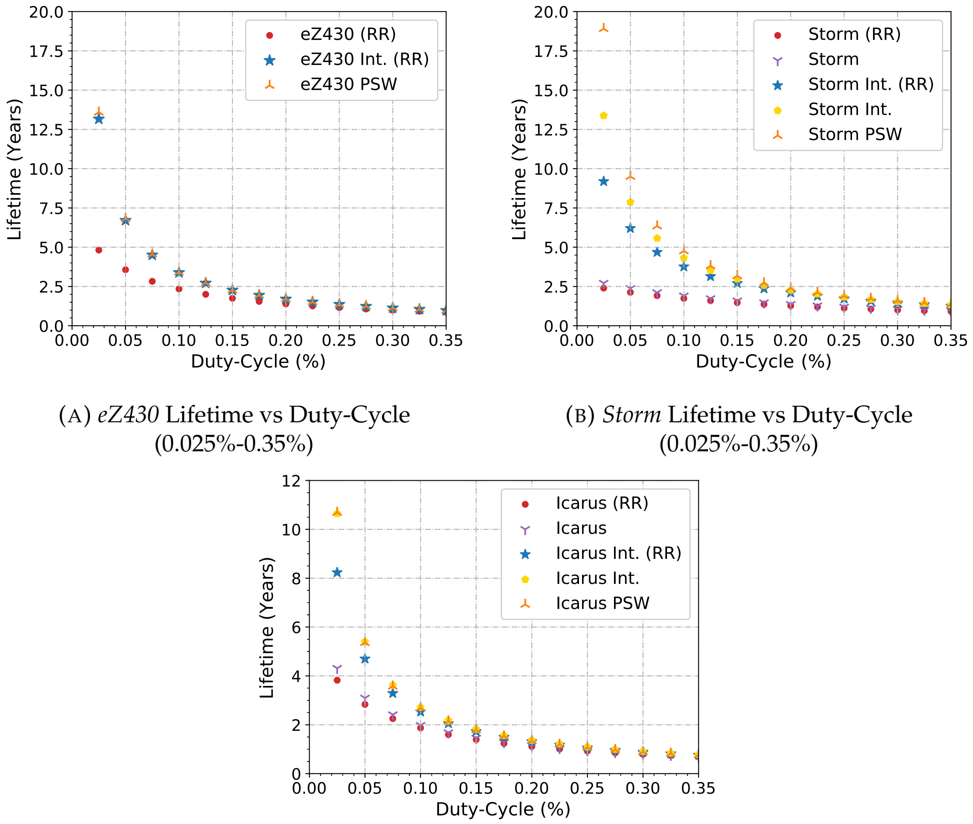
Lastly, we demonstrate the percentage reduction of the sleep current, when comparing existing sensor systems with our principle. When considering the *eZ430* we illustrate a reduction of *97.8 %*, while in the case of *Storm*, we highlight a substantial reduction of *99.83 %*, assuming that the *PSW* topology is applied in both cases.

2.5.2 Wake-Up Time Evaluation

In this subsection we characterize the wake-up time required for different *MCUs* to recover to an active state from the various low-power modes, and how this affects the performance of the proposed system. Apparently, the application of the proposed *PSW* topology comes at the cost of additional time required for the sensor to enter an active state, since in this topology the node disconnects completely from power when asleep. Table 2.2 presents the wake-up times of indicative *MCUs* under various low-power modes. Evidently, the cold-start time is always slower compared to the *standby* mode, while the modes that support *RAM* retention capabilities are exceptionally fast. For example, the *STM32L476* features *7 μs* when waking-up from a *RAM* retention state and *256 μs* when no *RAM* retention is supported, while it requires *3.9 ms* for cold-start. The wake-up times in different modes must be taken into consideration when selecting between the *PSW* and the *Interrupt* topology. For instance, a *MCU* with fast cold-start, such as the *MSP430FR5969*, can exploit the *PSW* topology while for slower *MCUs*, such as the *STM32L073*, it is preferable that they are configured in *Interrupt* topology.

Undoubtedly, wake-up times add to the overall active time of a node operating in *duty-cycle*. While potentially a drawback for our implementations, it should be noted that this is only the case when these times comprise a substantial portion of the cycle's duration in a *duty-cycled* scheme. It is worth noting that a typical node requires roughly *200 ms* [94] in active period when required to propagate a frame, while much longer active cycles are required when considering more sophisticated applications or processes, such as the re-establishment of a mesh network. As a result, the overhead of *1 ms* cold-start in the case of the *MSP430*, might be an acceptable energy cost to pay in order to minimize the power draw in sleep state. However, the *7.52 ms* cold-start of the *STM32L073* or an even longer cold-start duration might add a notable energy loss in the sensor's power budget. Consequently, *MCUs* featuring fast wake-up times are less likely to affect the overall active energy, but when considering slower families, it is preferable to take the overall energy trade-off into account before adopting the proposed principle. Of course, the above is strictly dependent on the *IoT* application's *duty-cycle*, as illustrated in the next section. We would like to note that the cold-start times are not provided by the manufacturers but measured for this work, and they can substantially change if some of the boot parameters are modified.

As already mentioned, *FRAM* functionality can be exploited in *PSW* toplogy to allow for state retention. However, there is a minor overhead time required to store all data



(A) *eZ430* Lifetime vs Duty-Cycle (0.025%-0.35%)

(B) *Storm* Lifetime vs Duty-Cycle (0.025%-0.35%)

(C) ICARUS Lifetime vs Duty-Cycle (0.025%-0.35%)

FIGURE 2.6: Lifetime Expectancy vs Duty-Cycle of Indicative Platforms

before the shut down and a similar overhead to recover the data from the *FRAM*. We measured this time exploiting *TI's* libraries for the *MSP430FR5969* and obtained an overhead of roughly 1 ms when restoring the application context and *100 B* of data from *FRAM* to *RAM*. This overhead must also be taken under consideration when assuming state retention via *TI's FRAM* support.

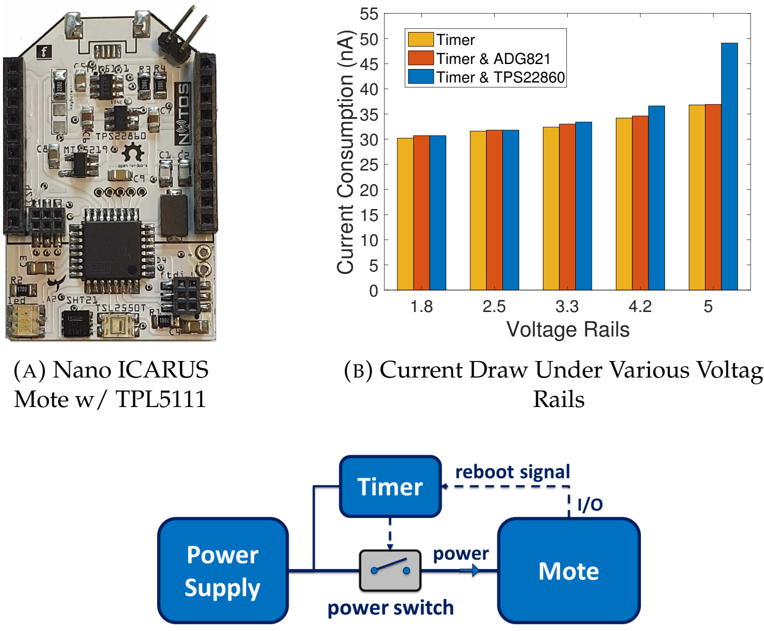
2.6 Comparison

In this experiment we evaluate the lifetime of two *state-of-the-art* platforms under varying *duty-cycles* comparing their performance when adopting our principle versus the standard features, in order to understand the improvement induced by the proposed system. Indicatively, we consider the platforms *eZ430-RF2500* equipped with the *MSP430F2274 MCU* and the *Storm* which encompasses the *ATSAM4LC8C*. Their power draw when asleep is *5.53 μA* and *13 μA* respectively, while the resulting power draw when adopting our technique is discussed in section 2.5.1. Moreover, Table 2.2 details the wake-up times for each *MCU*. Notably, the *MSP430F2274* supports only a *WDT* timer which is unable to provide alarm intervals greater of 8 s, thus we configure the mote to perform intermediate wake-ups in order to extend the maximum supported interval, which results in an average draw of 5.53 μA. Regarding their consumption in active state, we consider the average values of *12 mA* for the *eZ430* and of *8.6 mA* for the *Storm*, calculated by their instantaneous power profiles under different tasks when active. In addition, we assume an active period of *200 ms* [94] and that both motes feature batteries with an available capacity of *360 mAh* while the battery's self discharge is ignored.

Fig. 2.6a and 2.6b depict the calculated life expectancy of the aforementioned motes under varying *duty-cycles*. In each case, we also plot configurations that feature *RAM* retention (*RR*) where available. Apparently, when considering a *0.025 % duty-cycle*, we observe a substantial increase of *183 %* and *603 %* for the *eZ430* and the *Storm* respectively, considering the *PSW* configuration for both motes. Specifically, we note that the *eZ430* extends it's life from *4.8 years* to *13.6* and the *Storm* to *19 years* all the way from just *2.7 years*. Notably, under this *duty-cycle*, the *Storm* presents a worse lifetime with the standard features compared to the *eZ430*, while it presents higher life duration increase when our method is applied. This comes as a result of it's poor performance in sleep (*13 μA*) compared to the *eZ430* (*5.53 μA*), while featuring lower consumption in it's active state. The *Interrupt* configuration yields less pronounced but similar results. In the case of the *Storm* there is a differentiation among the *PSW*, *Interrupt* and *Interrupt RR* topologies in low *duty-cycles*, which stems from the significant variation of the power draw obtained between the corresponding states, *22 nA*, *944 nA* and *2.3 μA* respectively. On the contrary, the *eZ430* features nearly equal consumption on these states, hence the resulting lifetime is roughly similar. Higher *duty-cycles* present lower lifetime increase for both devices. However, even on *0.2 %* and *0.3 % duty-cycles* the Storm presents an increase of *75 %* and *50 %* respectively, when using the *PSW topology*. At the same time, the effectiveness of our approach for the *eZ430* tapers off (less than 10 % increase) for *duty-cycles* over *0.45 %*, while for the *Storm*, for *duty-cycles* over *1.5 %*.

Similarly, we plot the calculated life expectancy of the *ICARUS* mote, in different topologies in Fig. 2.6c. To this aim, we assume a consumption of *7 μA* in sleep state when in standard operation and an average consumption of *15 mA* in the active phase. We observe that the *PSW* and *Interrupt* topologies achieve nearly equal behavior, since the obtained power draw in these states are quite similar, *22 nA* and *132 nA* respectively. On the other hand, when *RAM* retention (*1.26 μA* draw) is enabled the overall lifetime is diminished, but still substantially improved over the standard operation.

Concluding, our principle greatly benefits IoT platforms with low energy-efficiency in sleep state when in low *duty-cycles*. We also note that the duration of the active period, does not affect the life expectancy of an *IoT* platform, unless it is extremely low, comparable with the wake-up times of the various *MCUs*.



(A) Nano ICARUS Mote w/ TPL5111

(B) Current Draw Under Various Voltage Rails

(C) Proposed Architecture Diagram

FIGURE 2.7: Developed Nano ICARUS Mote & Current Draw Measurements & Proposed Architecture

2.7 NanoPower Timer

Similarly to the implementation presented above using the RV1805 RTC IC we also present a nano-power system that is based on a timer IC. More specifically, we employ an on-board timer circuit along with a power switch to switch entirely off the sensor device and wake it again when the timer fires up. By employing this timer we are able to achieve the outstanding draw of 33 nA in the sleep state.

2.7.1 NanoPower Timer System Implementation

Our implementation is composed of a timer circuit and a power switch. The aforementioned elements can be integrated into the PCB of an available mote or even to be interfaced as an external circuit. We configure the timer to provide a one-shot high output signal after a specified interval through which we drive the power switch which in turn powers the under consideration mote that automatically enters in the active state. When all tasks in the active state are completed and the data are stored safely, the mote signals the timer circuit to reboot, thus the timer instantly cuts the power from the mote and countdowns again until the next interval. Below we present the details of the components used.

NanoPower Timer: Texas Instruments (TI) recently announced a series of nano-power timers ideal for power gating in duty-cycled applications. The timer we selected to use is the TPL5111 [243] which provides selectable timing intervals from 100 ms to 7200 s. In addition, the TPL5111 has a unique one-shot feature where the timer asserts it's output pulse for one cycle. We exploit this feature to provide power to the host mote, while we reset the function of the timer when the mote completes it's tasks by signaling it's *DONE* pin, as illustrated in Fig. 2.7c through an I/O pin of the host mote. The selected timer features wide voltage supply range from 1.8 V to 5.5 V, thus the proposed architecture can be easily integrated with all prototypes or commercial motes, just by utilizing one I/O pin for the provision of the *reset* signal. Notably, the time drift of the proposed timer varies between 100 - 400 ppm, which implies that we can't rely on stable intervals to form a synchronized mesh network, but we can only consider star network topologies. Despite the fact that the proposed timer is a quite inexpensive solution and the proposed circuit topology is simple, the timer lacks in terms of timing accuracy, thus the method presented above with the RTC (roughly 2 ppm accuracy) is the ideal solution when considering synchronized networks.

Load Switch: In our implementation we evaluated two different power switches that feature ultra-low current draw. The TI TPS22860 and the Analog Devices ADG821. Both can drive loads up to 200 mA, while they feature ultra-fast response times. The switches implement the same functionality, however the TPS22860 features roughly 2-3 nA power consumption, while the ADG821 only 0.2 nA.

In Fig. 2.7a we illustrate the nano ICARUS prototype mote, which implements the proposed circuitry. The mote relies on the ATmega328p, while it features an XBee-footprint radio socket, a light intensity and a temperature & humidity sensor.

2.7.2 NanoPower Timer Evaluation

To demonstrate the performance of our system we measure the power draw under various voltage levels, and plot the results in Fig. 2.7b. We clearly observe that the lower the voltage supply is the lower the consumption of our system is. We also observe that the TPS22860 switch presents higher draw in higher voltage rails than the ADG821. By comparing the achieved current draw of our implementation at 3.3 V with the eZ430 (draws roughly 1 μA on average) we observe a reduction of 96.7 %, which can clearly impact on the network's lifetime. To measure the power consumption profile of the proposed system we employed our high-fidelity monitoring tools presented in Chapter 4 and 7.

2.8 Conclusions

In this work we employed an external *RTC* module to control the *go-to-sleep* and wake-up functions of any IoT mote. The suggested principle is set-up with low-cost *off-the-shelf* components, while it remains minimally invasive to the host node. By adopting our strategy the power expenditure of an *IoT* mote in sleep state can drop as low as *22 nA*, which is at worst a reduction of *98 %*, compared to the most power efficient platforms available. Remarkable lifetime extensions can be achieved in low *duty-cycled* scenarios, while in higher *duty-cycles* the benefit is minimal. Specifically, we noted a lifetime extension for a state-of-the-art sensor node to *20 years* on a *0.025 % duty-cycle* scenario attaining an incredible increase of *538 %*. Moreover, we presented an analogous system employing an ultra low power timer circuit that draws only *33 nA* in sleep state. This system also drops remarkably the power consumption of any *IoT* mote, however, it does not support accurate time intervals, thus not proper for developing synchronized networks. Finally, we foresee that *IC* manufacturers will add similar functionalities to the ones proposed in this work into their *MCU's* within the next years in order to reduce the power profile of their products.

Our future plans include the development of a smart sub-circuit that will power any IoT node during its sleep phase with the aid of a super-capacitor in order to avoid the employment of a regulator IC in this phase. Notably, regulators feature high quiescent current draw, especially in low supply currents their efficiency is extremely low. In order to eliminate their power draw we propose the exploitation of a super-capacitor that will be charged during active phase (few mseconds are enough for this action), while the power supply of the under consideration IoT node will be switched to the super-capacitor in the sleep phase.

Chapter 3

eWake: A Novel Architecture for Semi-Active Wake-Up Radios Attaining Ultra-High Sensitivity at Extremely-Low Consumption

Contents

3.1 Introduction

Energy efficiency is a leading topic of research in the domain of *Wireless Sensor Networks* (*WSNs*). In most real-world applications, sensor nodes are battery operated, while their life duration is solely dependent on the battery's remaining charge and the node's power profile. A common strategy for saving energy in sensor networks is the *duty-cycle* practice, which suggests that sensor nodes enter a low-power mode, the so-called sleep state, in order to save as much energy as possible during their inactive periods. The sleep state is interrupted by short, burst events, where sensors sense, process and propagate data. The above principle is usually realized using internal or external time keeping circuits that provide fixed interrupt signals to awake the devices from their sleep state. Moreover, it is common that the interval of the wake-up signals is fixed and predefined depending on the application scenario of the network. Despite the fact that the aforementioned principle significantly reduces the overhearing and idle listening problem, which is a major source of energy wastage [261], it is not considered to be the best practice, especially in application scenarios that do not require fixed time intervals.

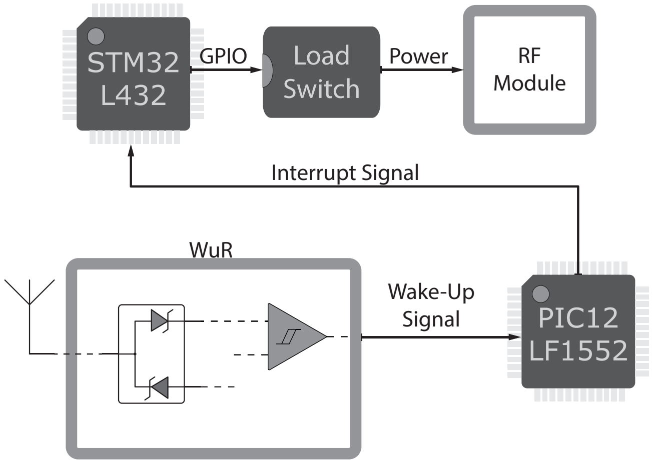


FIGURE 3.1: Proposed *WuR* Architecture

Apparently, most sensing applications operate in duty-cycle fashion since this is a reasonable way to cope with the requirements of specific application scenarios. However, this is not always the best practice. When in ducy-cycle, a sensor network must strictly follow a given interval (even if it is modified over the time), which suggests that all sensor nodes must switch to their active state at specific timestamps in order to perform a new sensing cycle. As expected, all network's nodes expend significant amounts of energy to perform an extra sensing, since they usually draw several mA when awake.

To this end, significant research attention has been given to the development of *Wake-Up Receivers* (*WuRs*) presented in a few research works [82, 140, 229, 147, 14, 76, 105, 171, 99]. This receiver is actually an auxiliary circuit usually attached to the main sensing device in order to notify the latter to switch from its sleep to its active phase when instructed so. This circuit typically draws less than *1 μA* in order to remain as energy efficient as possible. Usually, a semi-active *WuR* combines an envelope detector that is a passive circuit along with a low-power comparator circuit that is an active component. It is worth noting that the best obtained performance in semi-active *WuR* systems in terms of sensitivity is observed in [140] with a reported sensitivity of *-55 dBm*, while the receiver circuit drains *600 nA* when operating in its quiescent state. The aforementioned circuits have also gain significant attention over the last years with new principles presented in [54, 257, 235, 39]. Lastly, other notable works that touch the same concept are presented in [174, 266, 178, 239, 238, 93, 122].

In this work we leverage the existing semi-active principle and we introduce *eWake*, a novel enhancement that offers substantially increased sensitivity, that goes beyond *-70 dBm*, while the power consumption of our *WuR* circuit remains at the same levels.

3.2 Wake-Up System Implementation

In this section we present the mini-ICARUS mote that integrates a novel WuR circuit able to detect wake-up signals at up to -70 dBm, while we discuss the technical

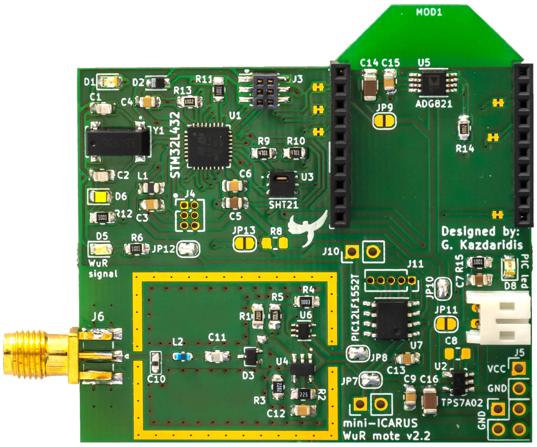


FIGURE 3.2: The mini ICARUS mote w/ *WuR* Receiver

details of the mote and the developed WuR.

3.2.1 System Architecture

The architecture of the mini-ICARUS mote is illustrated in the Fig. 3.1, while the mote is presented in Fig. 3.2. The device integrates a novel *WuR* circuit, an ultra low-power microcontroller (MCU) for supporting addressing capabilities, a host *MCU* and an *XBee footprint* for plugging in wireless interfaces.

The WuR circuit is discussed in detail in next, while the low-power MCU is the *8-bit PIC12LF1552T*. The PIC MCU is configured in its lowest sleep state, while it is awaken through a raising-edge Interrupt signal upon the reception of a wake-up packet. The *PIC MCU* process the received signal and identifies whether it should trigger the host *MCU* to wake-up, in case of receiving the correct wake-up packet, or to return back to the sleep phase if not. The host *MCU* is the *STM32L432* which is an ultra low-power *ARM Cortex-M4 32-bit RISC MCU* operating at a frequency of up to *80 MHz*. The *STM32L432* remains in sleep state, consuming only *20 nA*, awaiting from the PIC *MCU* an interrupt signal when a correct wake-up packet is received (containing the proper wake-up address). Upon the reception of such a packet the *STM32L432* switches to its active state in order to measure, process and transmit data as described by the application scenario. The mini-ICARUS mote integrates an *XBee footprint* socket where the user can attach the desired wireless interface, such as *IEEE 802.15.4* modules, *LoRa* transceivers or other wireless modems. Notably, wireless interfaces usually draw significant amount of power even when in sleep [112], thus we employed a load-switch to entirely turn off the power supply of the wireless module when not in use. The aforementioned method is known as power-gating technique [227].

3.2.2 Wake-up Receiver Implementation

The developed prototype receiver consists of low-cost off-the-shelf electronics and a low-power micro-controller. The schematic diagram of the receiver is illustrated in Fig. 3.3, which is integrated into the mini-ICARUS mote. For the wake-up receiver

Comparators	**Drain**	V_{OS}	I_B	**Sensitivity**
TLV3691	110 nA	3 mV	80 pA	-32 dBm [140]
TLV7031/41	335 nA	100 μV	2 pA	-
TLV3701	560 nA	200 μV	80 pA	-
LPV7215	580 nA	300 μV	-40 fA	-55 dBm [140]
LTC1540	300 nA	n/a	10 fA	-51 dBm [147]
MAX919	350 nA	1 mV	150 fA	-
TS881	260 nA	500 μV	1 pA	-
ADCMP380	92 nA	n/a	4 nA	-

TABLE 3.1: Compelling Comparators and their Specifications

a matching network, a passive rectifier (*HSMS-285c*), an Operation Amplifier and a Comparator *IC* (Integrated Circuit) were used. Typically, the matching network consists of a capacitor and an inductor element that together form an L-C network used to match the impedance of the antenna with the rest circuit. Then, a passive rectifier in the topology of an envelope detector is formed with the aid of two Schottky diodes, used to discard the high frequency signals and to deliver the modulated *OOK* signal. In our setup we used the HSMS-285c rectifier by Avago Technologies. Next, the harvested signal is processed by a low-power operation amplifier and a comparator circuit in order to fed it in binary format to the PIC12LF1552T MCU. The PIC12LF1552T MCU is responsible for processing the received signal and identifying the acquired address to verify whether it should wake-up the host node or not. To awake the network's nodes we utilize *868 MHz LoRa* radio transceivers, by modulating the propagated information using *On-Off Keying (OOK)* modulation.

Our finding lies in the observation that the power harvested by the envelope detector (the output signal of the HSMS-285c rectifier - *TestPoint A*, in Fig. 3.3) is not adequate to trigger the next stage when receiving high attenuated wake-up packets. Notably, in all the previously published works the harvested signal by the envelope detector (*HSMS-285c*) in *TestPoint A* is fed directly to the comparator *IC*, thus the overall sensitivity of the *WuR* is solely depended on the characteristics of the comparator being used. When using a comparator with low Input Offset Voltage (V_{OS}), as for example the *LPV7215* (300 μV V_{OS}), the *WuR* circuit achieves high sensitivity of *-55 dBm*, while when employing the *TLV3691* (*3 mV* V_{OS}) the obtained sensitivity is only at *-32 dBm*. Of course, the power consumption of these comparators is proportional to their performance, meaning that the *TLV3691* draws only *110 nA* while the *LPV7215* consumes roughly *580 nA*. Table 3.1 summarizes the compelling analog comparators along with their characteristics. Notably, another crucial parameter that plays significant role in the performance of the *WuR* circuit, is the Input Bias Current (I_B), which is drained by the comparator itself to bias the internal circuitry of the IC. In our work we propose a new architecture, by first amplifying the harvested RF signal (*TestPoint A*) and then feeding the intensified signal (*TestPoint B*) to a low-power comparator. Leveraging this strategy we are able to detect RF signals even when receiving highly attenuated packets, thus we achieve sensitivity far beyond the state-of-the-art. Our early experiments have illustrated sensitivity beyond

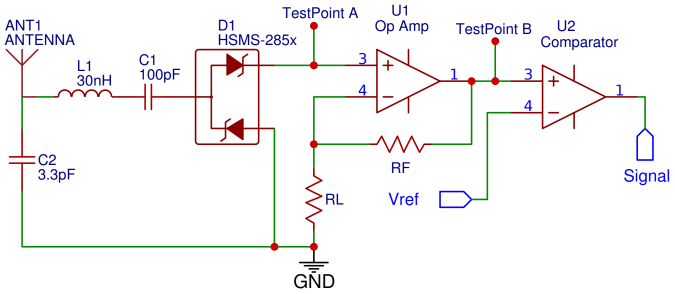


FIGURE 3.3: Proposed *WuR* Schematic Circuit

-70 dBm.

Fig. 3.4 illustrates the obtained signal level amplified by the operation amplifier as measured in *TestPoint B* with a yellow line, and the output of the comparator *IC* (labeled as *Signal*) with a green line, when receiving a wake-up packet at *-60 dBm*.

Components Selection: The Radio Frequency (RF) passive rectifier we used is the HSMS-285c which is the best option for harvesting RF signals, optimized for frequencies below 1.5 GHz. As regards the Operation Amplifier we have employed is the *LPV811* that features *450 nA* power draw, configured to amplify the obtained signal a hundredfold. Next the signal is fed to the comparator *IC*, which in our prototype is the *TLV3691* that consumes roughly *120 nA*. We opted for the *TLV3691* in spite of featuring high V_{OS}, since the proposed circuit is no longer dependent on the comparator's V_{OS}. The overall consumption of *525 nA* can be further optimized by reducing the supply voltage from *3.3V* to *1.6V*.

Regarding the Operation Amplifier and the comparator *ICs*, there are also several other options that could be used instead. Table 3.2 summarizes some recently launched *ICs* by *Texas Instruments* that feature extremely low-power consumption, in the order of a few hundred nA. Notably, the majority of the general purpose operation amplifiers consume at least a few μA, thus the usage of the proposed amplifiers is deemed necessary. Apparently, another very attractive option is the *LPV801* amplifier that draws only *320 nA*, while also featuring very low I_B of *100 fA*. In future designs we are looking into using the aforementioned amplifier to reduce the overall power consumption. By doing so, the overall consumption of the *WuR* circuit will be as low as *450 nA*. Regarding the employed comparator, is already one of the best options (Table 3.2), since only the *ADCMP380 IC* drains lower power but it comes with a fixed voltage reference of *500 mV* or *1 V*, thus not that applicable in our system. Notalby, currently we use a voltage reference threshold of toughly *27 mV*.

It is worth noting that all the selected components are off-the-shelf and substantially inexpensive, thus the proposed WuR can be easily implemented.

Address Matching: In order to reduce the overall false positive wake-ups as a result also the power consumption of the entire system, we employed an ultra low-power *MCU* that process the received signal prior triggering the host *MCU*. This *MCU* is

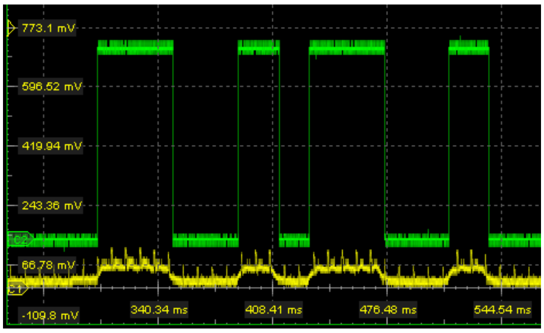


FIGURE 3.4: Amplified Signal upon Packet Reception at *-60 dBm* (Yellow line) & Comparator's Output (Green line)

the *8-bit PIC12LF1552T*. The *PIC12LF1552T* is configured in sleep state, consuming only 20 nA, while it is awaken upon the reception of a wake-up frame. Notably, the *PIC12LF1552T* requires a few μs to wake-up in order to process the signal, thus our wake-up packet contains a preamble with a similar time duration in order to be sure that the *MCU* is ready to read and process the required information. We selected the PIC *MCU* to provide address matching capabilities, because it drains only 32 μA per MHz thus considered to be an extremely inexpensive solution in terms of power consumption. In our application it is configured to operate at *2 MHz*, as also in most related works [140].

Matching Network: In order to tune the matching network of a *WuR* circuit, a network analyzer is essential since it calculates the exact values required for the *L-C* components employed, to transfer as much power as possible of the received radio signal available on the antenna to the rest of the circuit. This step is essential, since the only way to evaluate the sensitivity of the receiver is to execute realistic experiments with RF transmitters and of course by employing variable attenuators to precisely diminish the transmitted radio signal. However, such an apparatus is extremely expensive, therefore not easy to find and use in every laboratory. In our setup we followed a different approach to tune the matching network, by employing the low-cost *ADALM-PLUTO* [161] *Software Defined Radio (SDR)*. We used the *SDR* platform to transmit signals in different center frequencies in order to detect the frequency of the utilized *L-C* network that provides less attenuation. In essence, we plot the performance of the *L-C* network around the given frequency band. After repeating the same experiment by using different *L-C* values we identified the elements that provide the highest received signal strength in the 868 MHz band. To characterize the strength of the received signal we measured the harvested voltage output of the *HSMS-285c* rectifier (*TestPoint A* in Fig. 3.3) of the proposed *WuR* circuit. Notably, we easily approached the required *L-C* values by starting our trials with values similar to the ones presented in other works [172].

Transmitter Device: To awake the network's nodes we use the *LoRa* [8] technology, which is a low-power ultra-long range *IoT* technology. The wake-up packets are modulated using *OOK* modulation, which in essence means that carrier signal is

Op. Amp.	**Consumption**	V_{OS}	I_B
LPV521	350 nA	100 μV	40 fA
LPV801	320 nA	550 μV	100 fA
LPV811	450 nA	55 μV	100 fA
LPV821	650 nA	1.5 μV	7 pA
TLV8541	480 nA	300 μV	100 fA
TLV8801	450 nA	550 μV	100 fA
TLV8811	450 nA	75 μV	100 fA

TABLE 3.2: Compelling Operation Amplifiers

transmitted to represent the binary one, or suspended to represent the binary zero. Of course, the signal is modulated at a fixed transmission rate, so as the receiver to be able to extract the information by the received signal. Notably, the modulated packet contains two chunks of information, the *network id* and the *address* of the targeted node. This way, the wake-up circuit, ensures firstly that the packet belongs to its wake-up network and secondly whether the address refers to it. Of course, sensor nodes, can support more than one wake-up addresses, so that we can awaken a set of nodes at the same time. It is worth noting that the wake-up signal begins with an artificial delay to allow the activation of the *PIC12LF1552T MCU*, prior to the reception of the modulated information.

3.3 Conclusions

In this chapter we showcase a new scheme for semi-passive *WuR* circuits that remarkably increases the sensitivity of the existing state-of-the-art implementations, reaching beyond *-70 dBm*. Our finding lies in the employment of a nano-power amplifier that intensifies the signal prior to the decoding process. The overall consumption of our proposed circuit is roughly *580 nA*, while it can be substantially reduced by using alternative *ICs*. Lastly, our future plans include the measurement and characterization of the proposed *WuR's* power consumption profile using our power monitoring tools [106, 110].

This WuR implementation is part of the manuscripts presented in [103, 109, 105].

Chapter 4

eProfiler: High-Precision Power Monitoring System for IoT Devices Featuring Extreme Dynamic Range of Operation

Contents

4.1 Introduction

Energy-efficiency in the domain of Internet-of-Things (IoT) dominates the interest of the research community since real-world applications mainly employ battery operated sensors. Therefore, the life expectancy of a sensor network is solely dependent on the sensing node's power profile and the capacity of the battery in use. Evidently, further improvements in the sensors' power profile will increase the life duration of the network, which requires realistic feedback from in-situ power meters. To this end, several works [94, 64, 133, 268, 205, 57, 187, 154, 186, 121, 119, 108, 110, 218, 89, 124, 188, 27, 226, 244, 78, 5, 146, 206, 75, 152, 120], offer real-time monitoring to assist in the development of energy-efficient algorithms.

Recent IoT systems, integrate state-of-the-art elements such as RF radios, power regulators, sensing modules and micro-controllers, that feature improved power characteristics, as well as various operating and sleep modes. As a result, modern devices feature extremely low currents when switching to their sleep state conserving as much energy as possible. For instance, the Waspmote [1] and the eZ430-RF2500 [68] motes consume 860 nA and 690 nA respectively in their sleep state, while they draw several mA when active. Similarly, the ICARUS [112] mote exhibits a wide power profile dissipating only 22 nA in its quiescent state. Another prime example with significant power consumption deviations between its active and sleep state is the double-dip energy-harvesting system [148]. Although double-dip's consumption when active is typically in the range of few mA, in its quiescent state power consumption diminishes to only 700 nA. Identical behavior is observed by other energy-harvesting devices such as the [262, 40]. Apart from the aforementioned IoT systems as whole, their power supply sub-circuits are equally intriguing featuring wide dynamic currents that may vary from a few nA to a several mA. Especially, in the case of RF energy-accumulators the harvested currents may be in excess of a few nA [45, 20], even in close transmission distances. All the aforementioned scenarios suggest that the power profile of modern sensor nodes exceeds at least three orders of magnitude. Therefore, sophisticated tools with wide dynamic range spanning the entire spectrum of possible current draws are required to capture and characterize the power profile of IoT devices.

Commonly, power meters utilize a precise, low-impedance resistor, the so-called shunt resistor, which is placed in series, between the power supply and the Device Under Test (DUT). The voltage developed across the shunt is proportional to the current draw of the DUT according to Ohm's law. The selected shunt plays significant role in the accuracy of the entire system, thus, it must be carefully chosen. The shunt resistor role is twofold, *(i)* it should not affect the operation of the DUT, and *(ii)* provide sufficient level of detail to the next stage. Typically, a maximum voltage drop of 100 mV in a 3.3 V rail is tolerated by the DUT, therefore, the shunt must be calculated according to this limitation. For instance, for a maximum current of 100 mA, the selected resistor may not be greater than 1 Ω. This signal (1 mV-100 mV) can be processed by a pre-amplification circuit that magnifies the observed voltage drop to a full-scale analog signal to ensure accurate sampling by the ADC Integrated Circuit (IC). However, when measuring currents ranging from 1 μA or 1 nA the selected resistor (1 Ω) will introduce a voltage drop as low as 1 μV or 1 nA respectively, which is infeasible to prevail over noise and parasitic thermo-voltages [203]. Therefore, the gain must be generated from the shunt resistor itself. In essence, it is necessary to adjust the shunt resistor during the measurement procedure, without interrupting the operation of the DUT, while constantly measuring even within the transition phases.

In this chapter we introduce eProfiler, a high-precision power consumption meter that integrates a resistor switch able to adapt among the ranges of mA, μA and nA. The device supports 150 kSamples/s with 16 bit resolution. The key contributions are outlined:

- we present an innovative dynamic shunt switch able to adapt to different consumption ranges within 6 ns
- eProfiler is the first in-situ power meter able to measure currents of a few nA
- eProfiler integrates GPIO tracing and actuation to allow the correlation of the obtained data with specific DUT events
- we evaluate the performance of our meter in terms of error accuracy noting a maximum observed error of 1.6 %

The remainder of the chapter is organized as follows. Section 4.2 reviews the related work. System architecture and implementation are described in section 4.3, while the system's evaluation is presented in section 4.4. Finally, section 4.6 concludes the chapter.

4.2 Related Work

In this section we distinguish and present the state-of-the-art power meters. The SPOT [94] is an in-situ power meter that provides a dynamic range of 45.000:1, with an average error of 3 %. It uses a differential amplifier to boost the voltage drop across the shunt resistor and a Voltage-to-Frequency Converter (VFC) for the digitization process, to convert the amplified signal into frequency. The resulted frequency signal is accumulated and stored into two on-board counter ICs. Notably, this meter mainly focuses on long-term energy consumption monitoring, rather than acquiring detailed power measurements as we intend to do in this work. Moreover, the VFC IC cannot offer the desired resolution for measuring currents lower than 1 μA. Another in-situ power meter is the iCount [64], that provides a dynamic range of 100.000:1. The iCount measures the switching cycles of the regulator powering a sensing node to infer the power draw. This method enables power consumption monitoring with minimum additional hardware but suffers from high error rates reaching up to 20 % inhibiting low power measurement resolution. Two other power meters are the EMPIOT [57] and the ECO [205, 207]. They both measure the voltage drop across a shunt resistor to determine the power draw employing the INA219B and the INA226 current sense amplifiers respectively. These amplifiers integrate 12 bit and 16 bit ADC units, respectively. Apparently, the selected amplifiers feature high input offset voltage which results in significant output errors. However, the biggest pitfall of the mentioned ICs is their high input bias current which affects the measurement accuracy, discussed in detail in Section 3.3. Consequently, these amplifiers are not suitable for measuring currents in the range of μA and lower, but only in the mA range.

The authors in [133] present the FlockLab testbed, an observer board employed to remotely configure, measure power draw, and trace GPIO events of several attached sensor devices. For acquiring power consumption measurements, the observer features the MAX9923 current sense amplifier and a 24 bit ADC, while the Gumstix embedded device is used as the host computer. The MAX9923 is an excellent choice able to measure very low currents, since it features only 1 pA input bias current. However, the authors utilize only one shunt resistor, thus it is not feasible to accurately monitor currents ranging from a few nA to a few mA. Even with the aid of the selected 24 bit ADC, it is not possible to attain such dynamic range. The main drawback of FlockLab is the fact that it employs a Linux board, i.e., the Gumstix device, to communicate with the ADC unit and obtain the measurements. Linux being a process scheduling OS, and not a real-time system, cannot guarantee fixed intervals between the measurements, which is a significant requirement when designing power meters. This issue is also noted by the authors since they implement the entire communication with the ADC in kernel modules to run leveraging the highest OS priority. However, this method is merely a workaround, since even with the highest priority predictable measuring intervals cannot be guaranteed. In our work we also use a Linux-based device since this is the only way to support long-term monitoring capabilities along with an immediate user-interface. However, we overcome the aforementioned issue by employing a Programmable Real time Unit (PRU) integrated in the embedded system we selected to use which guarantees precise timing during the measurement acquisition, which actually shares its memory with the main system and guarantees precise timing in the acquisition of measurements.

Another distinguished work is the Nemo [268] in-situ power meter, which employs a dynamic shunt resistor switch of 4 different resistors, enabling power measurements in the range of 0.8 μA to 202 mA (250.000:1), maintaining the average error of 1.34 %. The Nemo is the first in-situ meter to adopt a dynamic shunt switch, which uses MOSFET elements to enable or disable each shunt resistor. In addition, a single Operational Amplifier (OpAmp) is used to amplify the voltage drop, applied to the entire resistor array. This is not an appropriate strategy since the impedance presented by each MOSFET affects the overall impedance of the array and leads to imprecise results. In our implementation we use a similar dynamic shunt switch formed by 3 resistors but we employed 3 different amplifiers, each applied directly to the corresponding resistor, without any other external elements impacting on the measurement accuracy. Moreover, the Nemo employs a voltage comparator to generate an interrupt to the host MCU upon a sudden current increase or decrease, which in turn adjusts the resistance of the shunt resistor switch accordingly. Despite using the modern TI MSP430F2618 MCU, the obtained delay when switching between different ranges is roughly 7 μs, which is significant when measuring transient phenomena. In fact, it is common that upon these transitions (i.e. wake-up from sleep state) sensor nodes may present extreme spikes in their consumption due to the fact that some electronics are initially powered (such as power capacitors, MCU, RF Radio, etc.). Notably, the Nemo discards any measurements polled within this period, since the validity of the data obtained during this period is questionable. Lastly, another limitation of Nemo, is that uses its EEPROM memory to store the obtained measurements, when configured for in-situ monitoring, which substantially limits the supported monitoring duration. thus not sufficient for long-term observations. Another meter that follows a similar approach is the CurrentRanger [154], which is a commercial tool for measuring power that relies on its host MCU, an STM32, to decide whether to switch to another current range, introducing notable transition delays in the order of ms. In our implementation we tackle this challenge using high-speed comparators directly controlling the switches which bridge the shunt resistors, achieving roughly 10 ns switching speed.

Apparently, a similar architecture is implemented by the RocketLogger [226], which also uses a comparator circuit to control the activation of the proper shunt resistors depending on the flowing current. The RocketLogger meter employs two different shunts, the first for measuring loads from 2 mA to 500 mA, while also a second resistor for monitoring low-currents in the range of 20 nA to 2 mA. However, the employed TLV1702 comparator features low propagation delay of 560 ns, while the overall delay of the system when alternating from the one range to the other is measured to be at 1.4 us, which is still a notable delay. Moreover, the RocketLogger features the 24-bit ADS131E08 ADC IC that supports only 64 kSampes/s, while our meter can reach up to 150 kSamples/s. Notably, the same power measurement system is exploited by the FlockLab 2 [244] observer board, hence this meter also features the same specifications in terms of power consumption monitoring.

Lastly, we note the Shepherd observer [78], a power monitoring device that focuses on the characterization of energy harvesting systems, featuring 381 nA resolution for measuring current loads, unsuitable for measuring nA currents.

4.3 System Implementation

In this section we highlight the challenges for designing a novel power meter, while we describe the proposed architecture and we detail the characteristics of our system.

4.3.1 Design Challenges

Wide Current Range: Any power monitoring system aiming to accurately measure the power consumption profiles of IoT sensing devices should feature a dynamic range spanning the entire spectrum of possible current draws. Hence, it must support readings from several mA all the way down to a few nA, which results in a range of at least 1.000.000:1 (considering 100 nA to 100 mA).

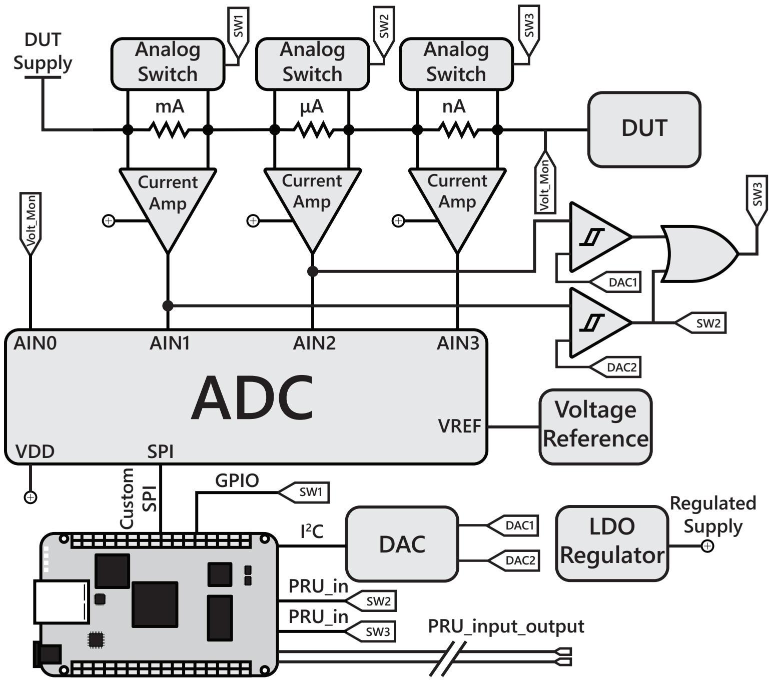


FIGURE 4.1: Architecture Diagram of the eProfiler

High-speed & high-resolution: The power meter should capture even the shortest in duration events to effectively record the power profile of a sensor system. Considering that the latest WiFi-enabled IoT nodes implement the IEEE 802.11n, that supports T_X rates up to 72 Mbps, and that a typical short packet has a size of 300 Bytes, the propagation of that frame would require roughly 33 μs. Therefore, a sampling rate of at least 100 kHz is required.

Long-term monitoring: The power monitor must be able to capture power data that last for several hours or even days.

Non-invasive operation: The power consumption monitoring procedure should not interfere with the normal operation or affect the actual energy consumption of the DUT.

GPIO tracing: We expect that a high-fidelity power meter must synchronize with the DUT to allow the correlation of the obtained measurements. An efficient way to implement this, is to trace GPIO events artificially realized in the DUT. The readings of these events must be as prompt as possible.

4.3.2 System Architecture

The architecture we followed to develop the proposed meter is illustrated in Fig. 4.1, while the developed Printed Circuit Board (PCB) is shown in Fig. 4.2. Notably, this board attaches on top of the selected embedded PC. The device consists of 6 main blocks, the embedded device, the ADC unit, the current sense amplifiers, the shunt

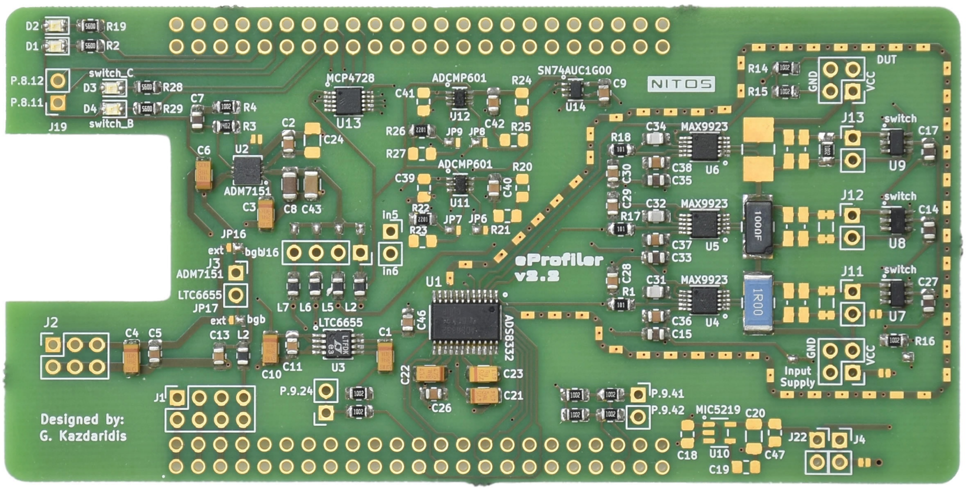


FIGURE 4.2: The eProfiler Power Meter

resistors switch, the comparators block and the power supply circuits. Our meter is based on the high-side current [**105**] sensing topology, which exhibits significant advantages over the low-side configuration featuring undesirable ground path-induced disturbances. We develop a dynamic shunt resistor switch that supports 3 different ranges (i.e., mA, μA, nA) in order to adapt to a wide range of currents. High-speed comparators are used to control the shunt resistor array, without the intervention of a logic unit that usually induces significant delays as denoted in [268, 154]. An important distinction of our architecture is the employment of a dedicated current amplifier for each shunt resistor and the fact that the selected ADC supports multiple inputs enabling the continuous polling of the amplifier channels. Apart from the amplifiers, the eProfiler is constantly monitoring the outputs of the comparators (labeled as SW2, SW3) on each measurement cycle. Upon completion of each experiment, a post-process script combines all the data to a final measurements file, accounting for the state of each comparator indicating which resistor/range was active at every discrete sample. The other advantage of our meter is the PRU unit integrated by the embedded device we selected which enables real-time communication with the employed ADC, supporting fixed intervals between measurement cycles. Moreover, we use the same PRU unit to trace I/O events of the DUT with ultra-low latency, to support the correlation of the obtained power measurements with specific activities of the DUT. The eProfiler also monitors the derived voltage rail that powers the DUT that can be used to infer accurate energy measurements, especially when significant power rail fluctuations occur as noted in [268]. Lastly, eProfiler features an on-board regulator which features an ultra-low noise output to power the sensitive electronics, while a reference circuit provides an ultra-stable reference voltage for converting the analog signals to digital.

4.3.3 Selected Components & Characteristics

In this section we present the components we opt for detailing their characteristics and the reasons for their selection.

	Type	G. Error	V_{OS}	I_B	CMRR	PSRR	Used in
INA139	CurS	0.10%	200 μV	10 μA	115 dB	-	-
INA219	CurS	-	40 μV	20 μA	120 dB	100 dB	EMPIoT [57]
INA225	CurS	0.05%	75 μV	72 μA	105 dB	140 dB	-
INA226	CurS	0.02%	2.5 μV	10 μA	140 dB	112 dB	ECO [205, 207]
INA282	CurS	0.01%	20 μV	25 μA	140 dB	110 dB	-
INA301	CurS	0.10%	25 μV	120 μA	110 dB	140 dB	-
AD8219	CurS	0.10%	200 μV	130 μA	110 dB	110 dB	-
LTC6102	CurS	-	3.5 μV	60 pA	-	150 dB	in [186]
MAX9923	CurS	0.12%	0.1 μV	1 pA	140 dB	99 dB	FlockLab [133]
OPA2333	OpAm	-	2 μV	150 pA	130 dB	120 dB	Nemo [268]
MAX4239	OpAm	-	0.1 μV	1 pA	140 dB	140 dB	uCurrent [187]
ADA4522	OpAm	-	1 μV	50 pA	160 dB	160 dB	Rock. [226], Flock.2 [244]
AD8422	OpAm	-	-	200 pA	150 dB	-	Shepherd [78]

TABLE 4.1: Compelling Amplifiers and their Specifications

BeagleBone Embedded PC: The embedded PC we opted for, is the Linux-based BeagleBone Black Rev. C [28], or the BeagleBone Wireless that supports wireless connectivity. The BeagleBone is a low-cost, embedded platform characterized by sufficient processing power capabilities (1GHz CPU with 512MB RAM), low-power consumption and several communication interfaces. The platform is responsible for controlling the peripheral units, and implements the software framework for the energy monitoring system. Furthermore, the device features an external microSD, used to locally cache the acquired measurements, prior to the offloading process. Apparently, the only limitation in the measurement duration is the size of the card employed. We avoid using a micro-controller device, which is common in the other power meters, but opt for a Linux-based system in order to support fast communication with the user and the ability to transfer large files. It is worth noting, that the BeagleBone is the only embedded PC that incorporates a PRU which is imperative for our meter.

ADC unit: The ADC we employ is the Texas Instruments (TI) ADS8332 [59] supporting the SAR architecture [253], which features 16 bits of resolution, high SNR of 91 dB and high Power Supply Rejection Ratio (PSRR) of 74 dB. The ADS8332 performs conversions at 500 kSamples/s (kSps), while it integrates 8 input pins allowing for parallel sampling with the aid of it's internal multiplexer. Therefore, the supported sampling rate is split among the configured channels. Apparently, when using all the supported current ranges (mA, μA, nA) 3 inputs are employed, thus the maximum attained speed is 166 kSps. On the other hand, when the user decides to also monitor the voltage rail of the DUT, 4 inputs are engaged resulting in a maximum speed of 125 kSps. Of course, users can also select to monitor only 1 or 2 channels (different ranges) in order to attain higher sampling speeds (500 and 250 kSps respectively). An alternative ADC that also features excellent characteristics is the ADS1278T. This is a 24-bit ADC featuring 8 inputs as well, but it supports simultaneous monitoring among all inputs at the maximum speed of 144 kSps.

Current Sense Amplifier: The current amplifier is of vital importance since it amplifies the obtained voltage drop across the shunt resistors and feeds it to the utilized ADC. Hence, it must be carefully chosen to distort the observed signal as little as possible. Table 4.1 summarizes the compelling and most used current amplifiers.

We select the MAX9923 [10] since it presents excellent characteristics in all parameters. The MAX9923 presents ultra-low Input Offset Voltage (V_{OS}). V_{OS} defines the output voltage deviation from the ideal value. Notably, the rest of the other amplifiers feature a few or in some cases several μA, thus they may induce significant variations from the ideal value, while the MAX9923 features just a few nA. PSRR and Common Mode Rejection Ratio (CMRR) also affect the performance of the amplifier. They define the ability of the device to reject common-mode and power supply noise signals respectively. High CMRR and PSRR is required when a differential signal is amplified in the presence of strong or even some electromagnetic interference. Of course, the selected IC also features low gain error, which is also critical for our application.

Another crucial parameter, overlooked by other works, is the Input Bias Current (I_B), which is drained by the amplifier itself to bias the internal circuitry of the IC. All listed amplifiers present good performance when measuring currents in the range of mA, but when it comes to amplifying signals in the range of nA, they drain comparable current to the monitored signals, which results in false readings. For instance, consider measuring a 1 μA load current using the INA139 amplifier which features 10 μA I_B. The INA139 amplifies the obtained signal across the resistor, but also drains 10 μA, which get amplified and summed up with the load current, resulting in a totally inaccurate reading. Consequently, amplifiers that feature I_B of a few μA should only be used to measure currents in the range of mA and above, and should be avoided for lower current ranges. Even when considering the OPA2333, that features 150 pA I_B, the obtained error for measuring a 1 nA current will be 15 %. The MAX9923 and the MAX4239 are the only ICs that feature such low I_B values, specifically 1 pA. However, the MAX4239 being an OpAmp implies that it's gain is controlled by external resistors, which can greatly drift due to temperature fluctuations adversely affecting the accuracy of the amplifier. Hence, it is recommended to utilize a current amplifier instead of a generic OpAmp as in [268, 94, 226, 244]. Lastly, we note that despite the fact that the LTC6102 seems to be a good candidate, it features a current output which requires an external resistor to assist the current-to-voltage conversion, before the signal can be sampled by the ADC. As a result, the accuracy of the system relies on the employed external resistor, which is a bad practice.

Another limitation of the current sense amplifiers, is the fact that they present high error rates at low differential input voltages (usually below 1 mV). We evaluated the INA139, INA282, AD8219 and LTC6102 obtaining an error of 5 - 8 % at the input voltage of 500 μV. On the other hand, the INA255, and the MAX9923 exhibit minimum errors at the same input voltage, roughly 2 %. This finding is not mentioned in the datasheets of the respective ICs, but is highly important when monitoring differential signals in the range of μV.

The MAX9923 is available at 3 different amplification levels (gain), 25, 100 and 250. We select to use the 25 gain, in order to effectively cover the whole range from 100 nA to 100 mA, using 3 current amplifiers, in combination with the selected shunt resistors. Lastly, we note that the MAX9923 supports lower Bandwidth signal (BW) compared to the rest of the amplifiers, thus an alternative amplifier, such as the INA301 or the OPA2333, can be used to monitor the mA range if required so, while the MAX9923 can be employed for the μA and nA ranges, where ultra-low I_B is deemed imperative.

Dynamic Shunt Resistor Switch: The shunt resistor switch features 3 resistors targeting 3 different current ranges. The values of the resistors have been selected at 1 Ω, 100 Ω and 10 kΩ aiming for the currents ranges of 100 mA to 1 mA, 1 mA to 10 μA and 10 μA to 100nA respectively. We refer to these bands as mA, μA and nA current ranges respectively for the rest of the chapter. We also note that each current amplifier is set to obtain only 1 to 100 mV input voltage signal in order to sustain high accuracy. Of course, the shunt resistors should be as precise as possible while featuring ultra-low temperature coefficients to avoid drift due to temperate deviations. Special resistor series are available for those kind of applications, usually in a 4 terminal format [225].

The tripping between different measurement ranges is realized by two ultra-fast comparators, the ADCMP601 featuring 3.5 ns delay. The ADCMP601 [51] requires a reference voltage, based on which it determines whether the obtained signal is below or above the desired threshold. In order to allow remote configuration we integrate the MCP4728 [52], which is a 12 bit DAC IC featuring 4 discrete outputs. The mentioned DAC, provides a resolution of roughly 800 μV in the scale of 0 – 3.3 V, offering wide flexibility in the provided reference threshold. In our application, we configure the transition thresholds at 1 mA and at 10 μA. An extra OR gate is employed in order to drive the third switch (nA range) that induces an extra delay of 2.5 ns. The ADCMP601 features programmable hysteresis (from 2 - 160 mV) that is once again controlled through the MCP4728 DAC. Thus, we avoid fluctuations between two current ranges when the measured load is close to the given threshold. Lastly, an extra switch is connected across the first shunt resistor – that is controlled by the Beaglebone and used to bypass the input pins of the first amplifier – in order to measure it's V_{OS} for calibration purposes as in [94]. We also note that the utilized load switches, used to bypass the employed shunt resistors, feature 4.5 ns turn-on delay, thus the overall time required to alternate from one current range to another is roughly 10 ns.

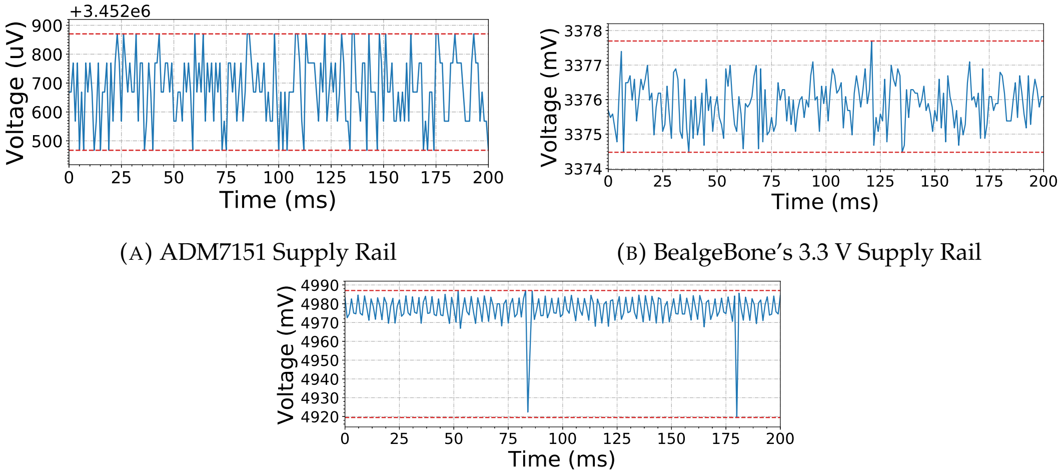
Power Supply & Voltage Reference Circuits: Despite the fact that the selected ADC and the current amplifier feature high PSRR values in order to reject any potential interference, it is recommended to employ low noise power sources [3]. ADCs are very sensitive ICs prone to external noise, thus must be treated very carefully in the design process. Especially when engaging embedded systems, such as the BeagleBone, that incorporate high-frequency clocks and step-down regulators that commonly feature high voltage ripple, the process of powering the ADC and current amplifier is even more challenging [94]. To this end, eProfiler embeds the Analog Devices (AD) ADM7151 [196] ultra-low noise, high PSRR linear regulator. The regulator features the extremely low noise of 1 μV and 94 dB PSRR. The voltage of the regulator is configured at 3.45 V through externals resistors. The ADC of eProfiler operates using a 3.3V reference value, but we selected to provide an extra 150 mV headroom to the on-board electronics to avoid potential saturation issues. It is worth noting that other power supply circuits such as the TPS7A49, which is recommended for powering precision analog applications [3, 197], features 12.7 μV noise, while a typical regulator such as the LP2985 presents 30 μV.

Typically, ADCs require a reference voltage to convert the obtained signals to the corresponding digital value. This voltage can significantly affect the performance of the system, thus an accurate voltage level, with extremely low noise, and high stability is required. For this purpose we employ the LTC6655 that features 625 nV peak-to-peak noise, high accuracy of 0.025 % and ultra-low drift of 2 ppm/°C. Notably, the LTC6655 outperforms the best in class reference circuits [85] (such as the REF3433 or the MAX6126).

PCB Design Considerations: We follow several principles for the design of the PCB board in order to attain high performance measurements. More specifically, we place a guard ring around the analog front-ends to prevent high-frequency noise from the digital parts of the system to distort the sensitive analog signals [94]. Also, we placed two separate ground planes [86, 216] for the digital and the analog parts that are connected through a pair of back-to-back Schottky [215] diodes and we place bypass capacitors [84] to all used ICs as close to their supply pins as possible. Lastly, a combination of ferrite beads and decoupling capacitors is employed in every power rail, used to attenuate the high frequency noise [156, 204].

4.3.4 Software Implementation

To reliably get measurements at specific time intervals, a real-time system is required. The Beaglebone helps greatly in this regard as it features two on-board PRUs [163]. The main processor, used by the Linux host, communicates with the PRUs via a section of shared memory. One more section of shared memory exists between the two PRUs available for communication. The latter of the two memory sections achieves faster communication, requiring only 1 cycle at a 200 MHz clock frequency, while the first memory requires 3 cycles. To fully control the ADS8332



(A) ADM7151 Supply Rail

(B) BealgeBone's 3.3 V Supply Rail

(C) BealgeBone's 5 V Supply Rail

FIGURE 4.3: Various Supply Rails Employed for the Evaluation Performance of ADS8332

we employ both PRUs programmed in assembly language for timing consistency. The first is used to control the ADC (ADC-PRU) and to obtain the measurements, while the second serves as a clock generator (Clock-PRU), allowing the user to precisely control the sampling rate. Actually, the ADC-PRU reads the Clock-PRU state through their shared memory pool avoiding toggling and reading of GPIO states. The selected ADC communicates through the SPI protocol, thus a custom SPI implementation was developed on the ADC-PRU. The process of acquiring one measurement of 16 bits lasts roughly 835 ns. When sampling at 500 kSps, the time interval between the measurements is 2000 ns, thus the remaining available time is 1165 ns. Within this period the acquired data are saved into the shared memory. Moreover, within the available time the desired GPIOs are monitored by the PRU, such as the outputs of the comparators (SW2, SW3), as well as the GPIOs used to trace events on the DUT. Similarly, these GPIOs are used as actuation pins. These actions require only one cycle by the PRU, while 3 additional cycles are required to write the obtained values into the memory (at 20 MHz speed). Notably, we perform GPIO measurements / actuation at every sampling cycle, thus the maximum attained delay is 2 μs; faster than the delay presented in FlockLab, which varies from 90 to 280 μs. Lastly, we note that an intuitive, network accessible User-Interface (UI) is developed, using the *Angular* framework for the front-end, and the *Node.js* platform for the back-end aiding the measurement process.

4.4 Performance Evaluation

In this section we evaluate the performance of the eProfiler.

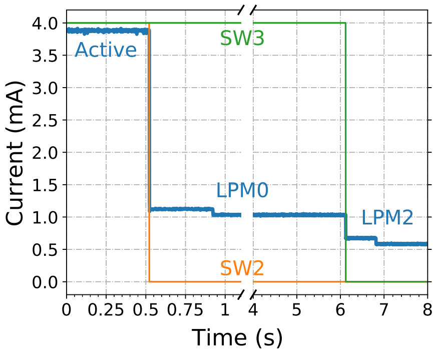


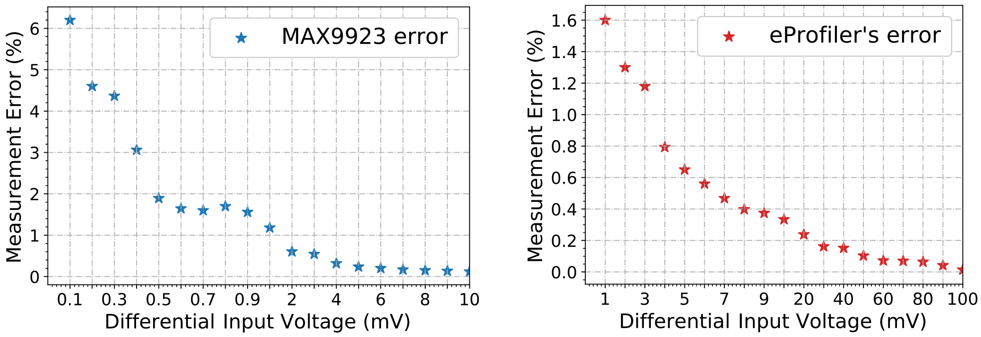
FIGURE 4.4: Comparators Evaluation

4.4.1 Power Supply Evaluation

In this section we highlight how different power sources can affect the performance of our system. We consider 3 different power rails, the output rail of the ADM7151 which is at 3.45 V, and the 3.3 V and 5 V rails of the BeagleBone. Figure 4.3, illustrates the measured rails with our tool. Apparently, the voltage rail supplied by the ADM7151 is quite stable featuring only 450 μV ripple voltage, while the 3.3 V and the 5V rails present 3.3 mV and 67.5 mV, respectively. Next we power the ADS8332 by employing the different rails while measuring a fixed reference voltage. The obtained results when powering the ADC with the ADM7151 introduces a STDEV of 0.1 (the values used in calculations are in mV), while the STDEV when the 3.3 V and 5V rails are employed are 6 and 80 respectively. Notably, the deviation of the obtained reference voltage is extremely high when powering the ADC with the Beaglebone's 3.3 V and the 5 V rails. This clearly highlights the impact of a power source on the performance of measurement acquisition systems [3, 192]. Similar but less pronounced results are obtained when powering the MAX9923 current amplifier with the various supplies. Apparently, the amplifier is not affected that greatly, but it still presents worse behavior when powered by the 3.3 V and 5 V rails. These findings imply that the noise of the power source plays a significant role in power meter systems.

4.4.2 Comparators' Evaluation

In this experiment we evaluate the response speed of the comparators in our system. To this end, we artificially bypass the μA and nA ranges through external jumpers, to disable the auto-ranging feature, performing only mA range measurements. However, the two comparators remain operational, both configured to monitor the output of the first current amplifier (mA range). Moreover, the BeagleBone samples their output values through the PRU unit. As external load, we engage the eZ430-RF2500 mote [68], and configure it to switch from it's active state to it's Low-Power Mode 0 (LPM0) state, and then to it's Low-Power Mode 2 (LPM2) state. The current consumption of the mote is roughly 3.8 mA, 1 mA and 510 μA in the given states, respectively. Thus, we set the thresholds in the comparators accordingly, to match



(A) MAX9923 Measurement Error

(B) eProfiler Measurement Error

FIGURE 4.5: Measurement Error of MAX9923 and eProfiler

the given currents. Notably, a 20 Ω shunt resistor was employed to better adapt to the given currents. Next we monitor this setup by sampling only the output of the first current amplifier (mA range) with a speed of 500 kSps. The Fig. 4.4 illustrates the draw of the node with the blue line, while the orange, and the green lines plot the outputs of the two comparators (signals SW2 and SW3). The comparator detects the current state of the eZ430 promptly, alternating it's output accordingly. We repeat the same experiment using a high-fidelity oscilloscope and observe that the response of each comparator is below 5 μs. Apparently, it is not possible to obtain a higher resolution than that.

4.4.3 Obtained Accuracy

In this set of experiments we evaluate the accuracy of the acquisition system. We first measure the accuracy of the MAX9923 by varying it's input voltage and recording it's output with a high-end voltage-meter (Keysight 34470A). The Fig. 4.5a illustrates the obtained error in the input voltage range from 0.1 mV to 10 mV. The MAX9923 presents notable error at 0.1 mV input (6.2 %), thus we decided to avoid applying such low signals. Similarly, it attains roughly 1.89 % error at 0.5 mV, while it's error at 1 mV is only 1.08 %. When measuring, higher input voltages it's accuracy improves further, introducing only 0.12 % error at 10 mV input and 0.013 % at 100 mV input. By properly selecting the shunt resistors, we can always maintain the input voltage polled by the MAX9923 in the 1 mV – 100 mV range; the obtained error will never exceed the aforementioned 1.08 % regardless if measuring nA, μA or mA currents.

Next we proceed by repeating the same experiment while measuring the obtained results with the on-board ADC in order to evaluate the accuracy of the entire system. In this experiment we vary the input voltage from 1 mV to 100 mV as actually happens when measuring the different ranges. Fig. 4.5b plots the obtained error. Apparently, the proposed device features a maximum error of 1.6 %, obtained at 1 mV input voltage, while the error at 10 mV and 100 mV input voltages is 0.33 % and 0.018 % respectively. It is worth noting that the average error is only 0.45 %, which illustrates an outstanding performance. Notably, the eProfiler can be used to measure even lower currents, in the range of 10 nA to 50 nA, however the obtained error at these inputs is 7.8 % and 4.4 %, respectively. This happens because the ADC unit fails to monitor such low voltage values. However, the whole setup can be re-adjusted with different shunt resistor values to support different current ranges. For instance, the values of 10 Ω, 1 kΩ and 100 kΩ can be employed to support the current range from 1 nA to 1 mA, maintaining the same accuracy, as highlighted in the next experiment.

In this experiment we characterize the power consumption of an ultra-low power Real-Time-Clock (RTC), the RV1805 [158]; the corresponding results from our measurement device are illustrated in Fig. 2.5a. The RV1805 dissipates only 17 nA in it's time keeping mode, however to retain the stated accuracy an auto-calibrating function – lasting for several seconds – is enacted on a periodic basis. In Fig. 2.5a, we demonstrate this auto-calibrating feature, lasting from second 20 to 70, alongside the corresponding power draw increase. Notably, we have configured the RV1805 to provide interrupt signals every 10 seconds, resulting in burst currents of 13.6 μA. Our obtained results are validated by the consumption values provided in the manufacturer's datasheet, as well as by the acquired measurements leveraging the uCurrent meter [187] observing minute discrepancies (less than 0.11 % obtained error). Our setup highlights the prominence of a high-sampling tool featuring extreme dynamic range to obtain high fidelity power consumption measurements even in nano-Ampere current ranges.

Lastly, we refer to the resolution of the system which is 199 pA in the nA range, 20 nA in the μA range and 2 μA in the mA range, providing impressive detail in the supported bands.

Notably, the proposed meter has been successfully employed to evaluate the power draw of several devices as well as discrete modules in our previous works [112, 105, 115, 107, 117, 116, 111, 113, 80, 102, 247]. Whlist, we the proposed system can be used to characterize the energy efficiency of many systems, such as of the [143, 144, 145, 129, 16, 237, 46, 169, 49, 50, 90, 212, 77, 19, 251, 35, 214, 17, 236, 48, 47, 170, 231, 15, 247, 114, 233, 104].

4.5 Development of the ePfofiler v2.0

In an effort to enhance some technical aspects of the presented power meter, we have designed a second version with some modifications both in hardware and software domains. The main modification is that the eProfiler v2.0 features the ADS1278T ADC IC, while also employs 6 current sense amplifiers to support wider dynamic range than the eProfiler v1. An architecture diagram of the eProfiler v2.0 is presented in Fig. 4.6. Apparently, the eProfiler v2.0 features 3 current ranges as also the previous version, the mA, the uA and the nA. The main enhancement is that

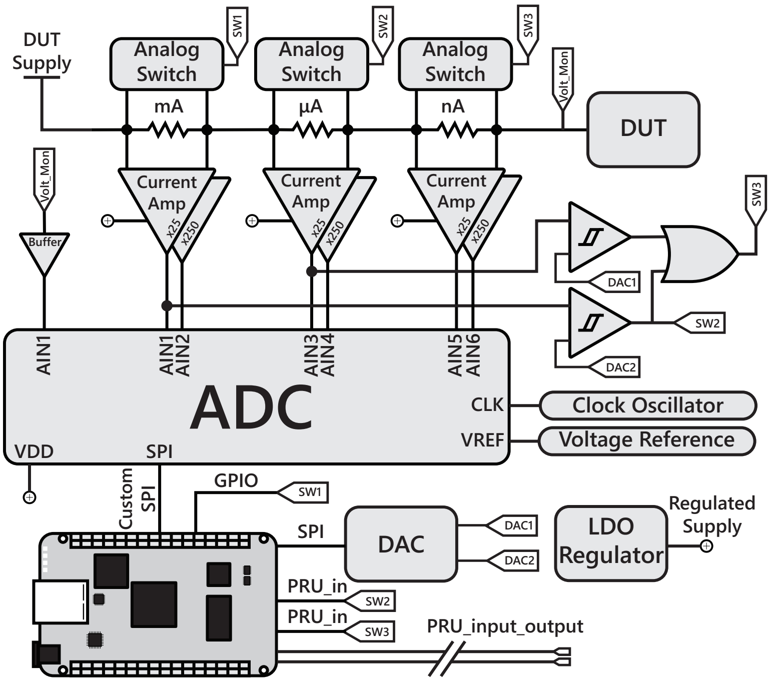


FIGURE 4.6: Architecture Diagram of the eProfiler v2.0

for each different range the eProfiler v2.0 features 2 current shunt monitors, as a result 6 monitors in total. We noticed that by using only one current shunt monitor (eProfiler v1) is not possible to cover a span of 1000:1 (i.e. to cover an entire band, such as mA or uA or nA). For example, the eProfiler v1.0 employs one MAX9923 to measure the range from 1 mA to 100 mA which is a current range of 100:1. The same amplifier could not be used for currents below 1 mA, since the delivered signal would feature immense noise, while also could not be used for currents above 100 mA due to the fact that the amplified signal would significantly exceed the supply rail, thus it would lead to saturation problems. However, one shunt resistor is able to accurately deliver clean signal within one band (mA or uA or nA), hence is not required to switch the resistor when measuring loads in the same band, but only when moving from one band to another. Upon this observation we decided to equip every shunt resistor with two current amplifiers featuring different gains. For instance, the first amplifier features a gain of 25 (MAX9923T) while the second one of 250 (MAX9923F). Both amplifiers sense the same resistor, while the first will focus on the current range from 1 mA to 100 mA and the second from 100 mA to 1000 mA. Using the same logic as in eProfiler v1, both inputs are sensed by the ADC, while the developed firmware running on the PRU unit is responsible to read the two values and decide which one is within the correct range so as to keep the respective input and skip the other one. Along with the measured value, we save information regarding the selected input so as to know the amplification gain used for each reading (i.e. 25 or 250).

Below we briefly describe some new components used in eProfiler v2:

ADC IC: We opt for the TI ADS1278T, which is a 24-bit ADC with parallel interface able to support sampling speeds of 144 kSamples/s.

Clock Oscillator: In this version of eProfiler, we selected to use an external clock oscillator, the DSC1001CL5, with a predefined clock speed at 25.6 MHz towards making available the second PRU unit that in this version is used to read GPIO pins as the first PRU is dedicated to communicate with the selected ADC IC.

Comparator IC: We select to integrate the LTC6752,which is a replacement of the comparator we used in the eProfiler v1.0, featuring a propagation delay of only 2.9 ns.

DAC IC: We select to integrate the TI DAC8564, which is a 16-bit IC providing better resolution in our system for the selection of the transition steps among the different current ranges.

MOSFETs: In order to bypass the shunt resistors used in the eProfiler v2.0 we integrated the Si7615ADN P-Channel MOSFETS, characterized by ultra-fast response time of 41 ns.

More details on eProfiler v2, are illustrated in Appendix A.

4.6 Conclusions

In this work we present eProfiler, a high-fidelity power consumption meter with long-term monitoring capabilities. The meter supports wide current ranges spanning from 100 nA to 100 mA, and features an average error of only of 0.45 % throughout it's measuring range. It is the first meter to support monitoring of nA currents which are becoming increasingly popular in IoT ecosystems. The eProfiler integrates GPIO state tracing to allow correlation of the obtained results with specific events realized on the DUT. All the aforementioned features aid in the power profile characterization of IoT devices. The eProfiler provides an in-situ, economically reasonable, and precise measuring framework, especially, for energy-harvesting applications. Leveraging eProfiler the consumption profile assessment of sensor nodes, and the efficiency of energy-harvesters can be effectively evaluated, enabling realistic estimations for the lifetime of deployed IoT sensor networks. Lastly, we note that the accuracy of the eProfiler can be further improved by applying the calibration method presented in the Excalibur [130]. Future plans include the assessment of eProfiler in effectively measuring currents in the pA range, as well as generating a power consumption taxonomy of popular and diverse IoT platforms.

Chapter 5

Evaluation of LoRa Performance in a City-wide Testbed: Experimentation Insights and Findings

Contents

5.1 Introduction

Air pollution is a crucial environmental situation that drastically impacts on public health, global environment, and worldwide economy. The World Health Organization confirms that air pollution is now the world's largest single environmental health risk, responsible for numerous premature deaths worldwide [175]. Long-term exposition to atmospheric pollutants cause health disorders such as respiratory, cardiovascular and cancerous diseases [138]. All the above highlight the urgent need for air pollution monitoring facilities, towards providing meaningful data to environmental protection agencies to better understand the effects and protect the public.

Latest advancements in embedded system technologies, facilitate the development of low-cost and small-size devices, capable of addressing the challenge for large-scale, dense deployments to monitor the pattern and presence of harmful pollutants. This direction is a key enabler as a step beyond the old-fashioned, atmospheric monitoring installations, usually deployed in a single location, incapable of delivering the requisite spatial resolution. To date, several research institutes have deployed pilot city-scale facilities to effectively provide real-time measurements of the atmospheric pollutants [87], [211]. In the city of Zurich [88], trams are equipped with environmental sensors forming a high spatio-temporal infrastructure, while in [211] the city of Santander is augmented with a vast number of sensing nodes providing real-time data.

Evidently, a major concern in such installations is the backbone connectivity among the sensing devices, as it has to cope with the large volume and the distribution of the deployed sensors. The recently introduced IoT Low-Power WAN (LPWAN) technologies, such as NB-IoT, LoRa, LoRaWAN and SigFox appear to be attractive candidates to cope with the nature of the aforementioned scenario. All these technologies use a physical layer that trades throughput performance, supporting from hundreds of bits to a few kilobits per second, to provide long range communication in the order of tens of kilometers.

LoRa [8] is one prominent technology of this type, employing a specific radio layer based on the *Chrip Spread Spectrum* (*CSS*) modulation. This standard allows the configuration of four critical parameters, *Bandwidth* (*BW*), *Spreading Factor* (*SF*), *Coding Rate* (*CR*) and *Transmit Power* (TX_P). Despite the huge potential that the LoRa standard exhibits, it has been insufficiently researched and studied. Several works [180, 131, 61, 263], study the performance of LoRa and LoRaWAN technologies in urban and rural environments, providing some first results in the behavior of this technology.

In this chapter we present a custom city-scale monitoring installation, reporting air quality and weather conditions in real-time. The framework consists of monitoring stations that assess various important parameters, such as temperature, humidity, dust particles and concentration of various gases, including *NO2, SO2, CO, NH3, O3*. The sensing devices are distributed across the city area of Volos, Greece, while they employ the LoRa communication standard to propagate their measurements. The key contributions of our work follow:

- development of a real-time air quality and weather conditions evaluation testbed setup in city-scale, employing custom sensor nodes and gateway devices
- thorough analysis of protocol PDR versus RSSI performance exploiting measurements over a period of 2 months
- detailed in-lab measurements characterizing the energy efficiency performance of the LoRa chipset, across a wide range of protocol configurations along with

meaningful insights and findings

The remainder of the chapter is organized as follows. Section 5.2 reviews related work. System Implementation and developed components are described in Section 5.3, along with Experimentation Parameters configuration in Section 5.4. Section 5.5 characterizes the obtained power consumption results, while experimentation evaluation is demonstrated in Section 5.6. Finally, Section 5.7 concludes the chapter and outlines future directions.

5.2 Related Work

In this section we discuss the related work divided into two fields. Firstly, we present works that investigate the performance of LoRa as a communication protocol, while in next we remark works that evaluate energy efficiency of LoRa standard.

In the first domain, several existing works [33, 26, 21, 173, 179] provide measurements regarding the coverage the capacity, the delay and the throughput performance of LoRa protocol. In [180] authors measure the performance of LoRa protocol in an urban environment, highlighting it's variable behavior delivering a packet error rate between *3* and *90* %, even when using high transmission power settings. Moreover, they present experimental results illustrating that is not straightforward how to correlate RSSI and packet error rate parameters when using LoRa technology, noting that even when the RSSI mean is high there are scenarios that packet error rate can be low. [32] illustrates an experimental study of the impact of the LoRa transmission parameters selection on the communication performance. An indoor setup of two nodes is used along with a developed link probing scheme to evaluate the optimal parameters for the packet transmissions. However, the setup used can not lead to safe conclusions since it is based only on two nodes located in a specific environment.

Considering testbed platforms, [61] and [60] provide framework tools to evaluate LoRa technology. Specifically, *OpenChirp* [60], demonstrates custom LoRa-enabled sensing nodes and Gateways that utilize the Semtech's *SX1276* chipset. The firmware is built on top of the open-source *TI-RTOS* and users can register and define the parameters they wish to follow. Similarly, *WiSH-WalT* [61], is a framework for controllable and reproducible experiments. A custom interface allows the parameterization of LoRa settings, such as *TX*, *SF*, *BW* and *Error Coding Rate*. The sensor nodes feature again the *SX1276* and the chapter also includes findings regarding the correlation of PDR versus the spreading factor used as well as *PDR* versus TX_P.

We remark the work in [131] that provides several useful findings based on realistic experimentation. Authors proved that LoRa is capable of communicating over 10 km links in *LOS* conditions, while in *NLOS* is severely affected by obstructions and vegetation. Moreover, authors quantify the importance of the *SF* parameter over the rest ones and urges the selection of the lowest possible *SF* whenever possible.

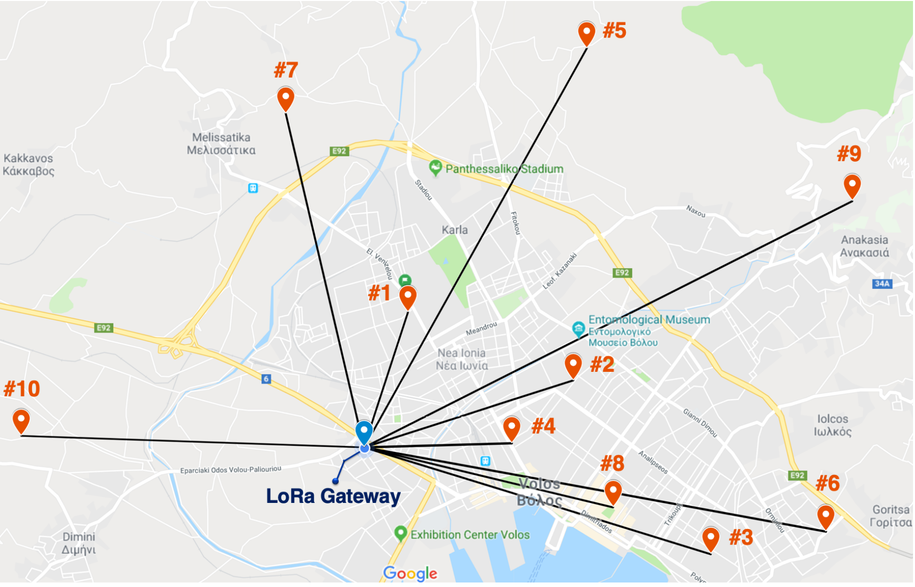


FIGURE 5.1: LoRa Testbed Topology

Authors in [44] suggest that when considering nodes at the edge of communication range it's preferable to employ faster *PHY* settings at high TX_P than lower *rate* configurations, by comparing the effective bit rate in each case. In our work, we employ a city-scale experimental setup to obtain meaningful data in order to characterize the performance of LoRa links. More specifically, we analyze the obtained results and correlate the obtained *PDR* with the *RSSI*, while we also suggest the optimal rate configurations to be used for each different link scenario. Other notable works are the [176, 30, 134, 242, 71, 100, 31, 165, 22].

In the field of energy consumption characterization of LoRa protocol, there are several works [32, 141] that are based on data-sheet power data, while some other [60, 131, 43] employ measurement tools to obtain power consumption. All works aim at modeling the energy performance of LoRa to characterize the lifetime of the under consideration setups. We distinguish the work in [131], in which authors correlate the obtained power consumption data versus the selected transmission power setting and estimate the lifetime of four different nodes that utilize the Semtech's *SX1276* and the *RFM96* LoRa chipsets. In our work we focus on detailing the instantaneous current draw for each possible combination, providing also meaningful findings, while we also calculate the energy efficiency per bit in both transmission and reception phase, across different payload sizes, which data can aid the development of energy efficient rate adaptation algorithms. Other notable works are the [157, 260].

5.3 System Architecture & Implementation

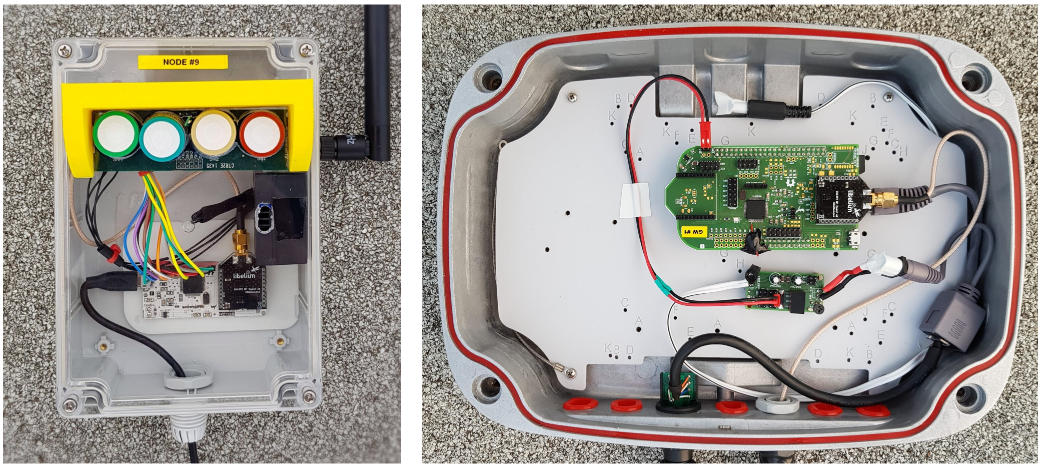
In this section, we present the components and the architecture of our testbed that consists of one LoRa Gateway (*GW*) that is deployed on the rooftop of our University's premises and ten Sensing nodes that are scattered across the city of Volos,

ID	GW Dist. (km)	Altitude (m)	LOS/NLOS	Sensors	Bytes
GW	-	23	-	-	-
1	1.3	43	NLOS	3	24
2	1.98	30	NLOS	3	24
3	3.25	32	NLOS	1	14
4	1.33	25	NLOS	1	14
5	4.12	179	LOS	3	24
6	4.22	55	NLOS	3	24
7	3.11	128	LOS	7	44
8	2.32	35	NLOS	7	44
9	4.92	170	LOS	7	44
10	3.10	8	NLOS	3	24

TABLE 5.1: Testbed Node Characteristics

Greece. The end nodes feature various air quality probes, characterizing gas concentrations, dust particle concentration, along with outdoor air temperature and humidity modules. Fig. 5.1 illustrates the testbed topology, while the characteristics of each individual sensor device are presented in Table 5.1. We clearly observe that the testbed offers a wide range of wireless links that differentiate in terms of communication distance, elevation difference and *LOS/NLOS* conditions. For each node, we also note the amount of payload bytes transmitted when transferring the respective number of sensed parameters, which includes *5 Bytes* per environmental parameter plus *9 Bytes* of own protocol overhead, while it does not include the LoRa *preamble*, *header* and *CRC*, as specified in [8]. Below we present the custom developed *GW* and sensing devices, as well as the software toolkit developed for obtaining LoRa performance measurements.

LoRa Gateway: The *GW* is based on the BeagleBone Black Rev. C board [28] which is a low-cost, embedded platform characterized by sufficient processing power capabilities (*ARM Cortex-A8 Processor* at *1 GHz* with *512 MB RAM*) and several I/O pins, while it also embeds an Ethernet port for communicating with the backbone network. On top of BeagleBone we attach a custom-designed PCB board (cape) that integrates our electronic components and features a slot for connecting a LoRa transceiver. More specifically, the micro-processor we employed is the *ARM Cortex-M4 32-bit MK20DX128VLH5* which is interfaced with the LoRa transceiver responsible to run all the requisite libraries for it's operation. The micro-processor software is also responsible for implementing a polling mechanism to aquire measurements from the sensor nodes and also initiate the link quality experiments. Moreover, the micro-processor communicates with the BeagleBone via a UART Bus. The micro-processor passes all the relevant information from the received packets, through the UART, to the Beaglebone. The Beaglebone receives and handles the data through software that has been developed using the NodeJS framework. The software receives the packets from the serial port in raw-byte format from the LoRa interface, using an own defined communication protocol, and then processes and decodes the data from the packet. The measurements are then published to a PubSub server using a light-weight messaging protocol (MQTT) and are stored to a time-series



(A) Sensor device

(B) Gateway device

FIGURE 5.2: Testbed Devices

database (InfluxDB). For the LoRa transceiver we employ the SX1272 [202] chipset manufactured by Semtech, which is paired with a 4.5 dBi antenna. Fig. 5.2b illustrates the developed gateway device deployed inside a waterproof aluminum case. The custom PCB featuring the microcontroller and the LoRa transceiver is the green board, which is placed on top of a BeagleBone device, while a small step-down converter is placed alongside employed to provide regulated power of 5 Volts via the Power-over-Ethernet connection.

LoRa Sensing Devices: The same micro-processor is used for developing our custom sensing device. The MK20DX128VLH5 [159] is a *32-bit ARM Cortex-M4* CPU, clocking up to *96 MHz* and featuring *128 KBytes* of flash memory. It also supports ultra-low power sleeping modes to allow for power efficient operations. Furthermore, the node employs the same LoRa interface (*SX1272*) as the gateway device, attached to a 4.5 dBi omni-directional antenna. Each node is equipped with a varying set of sensors, as illustrated in Table 5.1, featuring 1, 3 or 7 sensors in total. The sensors utilized are gas concentrations probes (*NO2, SO2, CO, NH3, O3*), dust particle concentration *PM2.5*, *PM10* and a temperature & humidity module. The air pollution probes are manufactured by Alphasense [9] and utilize a custom analog front-end to properly amplify the obtained signal which is fed to the micro-controller's analog pins. The dust particle concentration module is the HPMA115S0 by Honeywell [221] while the temperature & humidity sensor is the SHTC3 by Sensirion [224]. The micro-processor hosts a program that implements the sensor reading and collection of measurements, and also the communication between itself and the SX1272 module. The software is designed to periodically read data from the sensors, transmit the sensor data and also perform LoRa-link performance metrics on specified time-slots, as requested by the *GW*. Fig.5.2a presents one sensor device attached with four gas concentration probes at the top and a dust particle concentration module attached inside the box in the right side, while the host micro-controller board is illustrated in the middle of the enclosure interfaced with the LoRa SX1272 transceiver.

TX_M	BW	SF	Data Rate (bps)	Sensitivity (dBm)	Airtime (msec)
1	125	12	293	-137	1155.07
2	250	12	586	-135	577.53
3	125	10	977	-133	329.72
4	500	12	1172	-129	288.76
5	250	10	1954	-130	164.86
6	500	11	2149	-128	164.86
7	250	9	3516	-128	82.43
8	500	9	7032	-122	41.21
9	500	8	12500	-119	23.16
10	500	7	21875	-116	11.58

TABLE 5.2: LoRa Protocol specifications

Software Toolkit: Through the deployed list of devices, we continuously collect detailed air quality measurements, while also evaluate the performance of the employed LoRa communication links. The overall procedure is orchestrated at the *GW* level, which periodically polls the different sensing nodes that continuously remain in receive mode. Sensor polling is executed in a Round-robin fashion, instructing sensors to transmit collected air quality measurements (with an interval of 10 minutes) or to participate in a link quality evaluation experiment (executed twice per day). The two procedures are executed independently of each other, taking advantage of the employed polling approach, which enables us to avoid the impact of overlapping transmission interference, when performing link quality experiments. Upon receiving a request for experiment, each sensing node transmits 10 consecutive packets at the *data rate*, *payload length* and *transmission power* that has been determined by the *GW*. The gateway sends the experiment settings embedded to the experiment request packet, and the node replies with an acknowledgement packet in order to start the procedure. The per-node link quality of the network is determined through the *Packet Delivery Ratio* (*PDR*) of the aforementioned experiments. The experiment results are also forwarded to the *MQTT* broker through the *GW* and then stored on a time-series database, resulting in a distributed and scalable architecture.

5.4 Experimentation Parameters

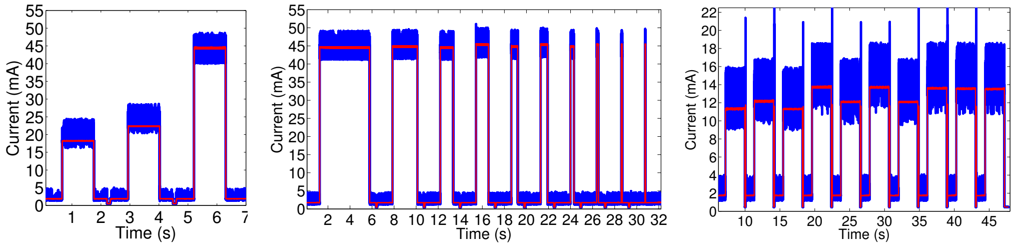
In this section we detail the procedure we are following when using the developed software toolkit in order to evaluate the performance of LoRa links. In essence, we are testing, adopting a probing fashion, all possible LoRa configurations to generate a rich set of measurements. More specifically, we vary the *Transmission Power* (TX_P), the *PHY* layer *data rate* and the frame *payload* size at each transmitter node. Starting with the (TX_P), we configure three different levels (*0, +7, +14 dBm*) . Through this configuration, we result in 30 different link combinations, as illustrated in terms of uplink RSSI in Fig. 5.8a. Considering the parameters affecting the resulting *PHY* layer *data rate*, we configure the full list of available settings of (*BW*, *SF* and *CR*), by using the list of combinations presented in Table 5.2. For simplicity, we refer to these combinations as *Transmission Modes* (TX_M) from 1 to 10. We specifically use the aforementioned list of combinations, considering Semtech's recommendation of using *SF* settings of *SF7* to *SF12* and *BW 125*, *250*, and *500 kHz* [202], so as to ensure acceptable transmission distance and data rate trade-off, since both *SF* and *BW* affect propagation duration and data rate. Notably, in all combinations the *CR* option is set to *4/5*. Additionally, Table 5.2 lists the resulting sensitivity and indicative airtime for each TX_M, using a reference payload size of *10 Bytes*. The specified *data rate* combinations in Table 5.2 are also followed by the Libelium [132] as well as by the [44] if consider only *4/5 CR*. Finally, regarding the varying payload sizes, we configure the values of *14, 24 and 44 Bytes*, which correspond to the actual number of payload bytes transmitted when collecting air quality measurements from *1*, *3* and *7* air sensors accordingly. In addition, we also use the payload size of *255 Bytes* for stress testing the LoRa protocol, as this is the maximum supported value.

5.5 LoRa Power Consumption Evaluation

In this section, we characterize the power consumption profile of the LoRa SX1272 chipset, under various protocol configurations. The experiments are conducted in the lab, employing a set of a transmitter and a receiver. To measure power draw, we fabricated an intermediate circuit board that is placed between the host sensor/microcontroller board and the LoRa interface. This board features a small shunt resistor (*1* Ω) that is placed between the sensor's regulated power supply (*3.3 Volts*) and the SX1272's input supply pin. The voltage drop across this resistor is proportional to the current draw of the LoRa interface, which we measure with our high-end power consumption monitoring tool, presented in Chapters 4 and 7. Our monitoring device reports the instantaneous power consumption, providing high-fidelity measurements at ultra fast sampling rate. In this set of measurements focusing on LoRa power consumption evaluation, we employed *100 MHz* sampling speed at resolution of *16 Bits* that provides the required level of detail. The measurements are processed offline to calculate the energy draw. To characterize specific events, such as packet transmission or reception, we isolate the respective part of the measurements and calculate the average value. Notably, the radio interface used for the power consumption measurements is the Libelium's SX1272 module [132] that integrates only the RF chipset along with some integral electronics requisite for the modules operation, which do not affect the modules total consumption.

5.5.1 Instantaneous current consumption profile

In the first experiment, we characterize the instantaneous current consumption profile of the SX1272 LoRa chipset, during frame transmission and reception under various TX_P and TX_M settings. More specifically, we configure the LoRa chipset in all available configurations, firstly in transmission mode and then in reception mode and we isolate the respective events in order to measure their instantaneous current draw.



(A) Varying TX_P, in TX_M 1

(B) Current cons. in T_X, across different TX_M, employing +14 dBm TX_P

(C) Current cons. in R_X, across different TX_M

FIGURE 5.3: Obtained Current Consumption Measurements

The results of the first experiment are illustrated in Fig. 5.3a where the TX_P increases from *0 dBm* to *+7 dBm* and *+14 dBm*, while using the fixed TX_M setting of the first rate configuration and a payload size of *10 Bytes*. Similarly, in Fig. 5.3b we set the TX_P to *+14 dBm* and vary the TX_M to all the available rate configurations from 1 to 10. In this plot we use a fixed payload size of 100 Bytes and we clearly observe that the relation between the duration of each frame propagated and the transmission speed of each TX_M is inversely proportional as expected. We repeat the same experiment but this time we configure the LoRa chipset as a receiver and we measure the instantaneous power draw, as illustrated in Fig. 5.3c. We can observe the variation in instantaneous power consumption in each different TX_M. Notably, the red line illustrated in Fig. 5.3a, Fig. 5.3b and Fig. 5.3c represents the average power draw calculated as a moving mean over a window of 100 values in an effort to provide a more representative indication to derive safe results, since the obtained measurements present high variations. Finally, the sudden drops from idle state to zero between each packet transmission or other spikes illustrated are related to the re-configuration of the LoRa chipset parameters that we execute prior to each transmission or reception.

The analytic results across all tested parameters are listed in Table 5.3. We clearly observe that switching from the *+14 dBm* TX_P (*44.64 mA*) to the *+7 dBm* TX_P (*22.13 mA*), we can reduce current consumption by *50.43* %, while marginal decrease of *14.33* % is obtained when switching from *+7 dBm* to *0 dBm* TX_P. Both observations are made for the first TX_M configuration. Considering the whole set of TX_M configurations the reduction varies at the same levels, specifically in the case of switching from the *+14 dBm* to the *+7 dBm* TX_P it varies from *50.43* % (TX_M: *1*) to *49.42* % (TX_M: *4*), while in the case of *+7 dBm* to *0 dBm* TX_P we obtain a variation from *14.45* % (TX_M: *7*) to *13.78* % (TX_M: *6*). It is worth noting that the differentiation among the different TX_P levels, stems from the fact that a Power Amplifier (PA) circuit block is engaged in order to amplify the propagated signal, which in some cases drains the most of the total power [123]. The latter, suggests that it is preferable to use *+7 dBm* TX_P instead of *0 dBm*, to achieve sufficiently higher PDR at almost the same energy cost in scenarios where higher link budget is required.

TX_M	TX_P **(mA) +0 dBm**	TX_P **(mA) +7 dBm**	TX_P **(mA) +14 dBm**	RX_P **(mA)**
1	18.96	22.13	44.64	11.24
2	19.27	22.41	44.93	12.26
3	18.96	22.13	44.64	11.24
4	19.84	23.02	45.51	13.86
5	19.23	22.46	44.97	12.26
6	19.84	23.01	45.54	13.86
7	19.20	22.45	44.93	12.26
8	19.78	22.99	45.51	13.86
9	19.72	22.90	45.46	13.86
10	19.72	22.90	45.47	13.86

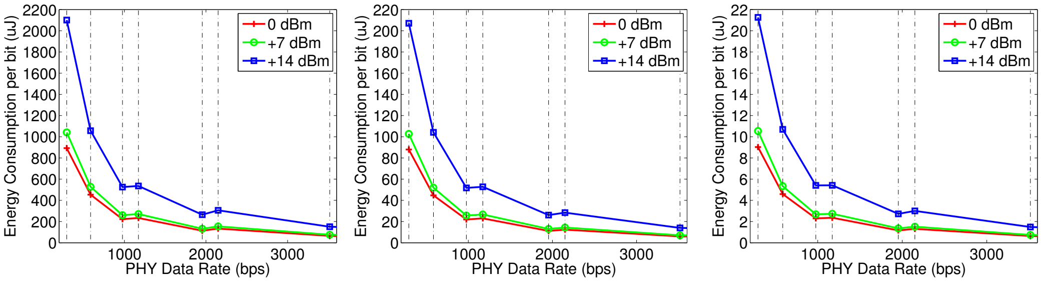
TABLE 5.3: Instantaneous Power Draw of *SX1272* chipset

The obtained results also highlight the impact of TX_M parameter in power consumption under the same TX_P. Relating consumption data with *BW* and *SF* parameters of Table 5.2, it is made clear that the *BW* parameter plays significant role in the resulting draw, while *SF* configurations merely impact the obtained current consumption. We observe that TX_M configurations with fixed *BW* present roughly stable power draw. For instance when employing *BW* of 125 KHz (which is realized in TX_M *1* and *3*) the power consumption remains exactly the same under the same TX_P setting. Whilst, in the rest *BW* configurations of 250 KHz and 500 KHz the power consumption variation under the same *BW* and TX_P settings does never exceed *1* %, which further highlights that the *BW* parameter totally affects the resulted power draw, while *SF* setting plays minor role. Notably, the highest obtained draw variation under the same TX_P is *4.47* % for the *0 dBm* TX_P, *4.01* % for the *+7 dBm* and *2.03* % for the *+14 dBm*. It is worth noting that the adjacent TX_M 5 and 6 present significant draw variation despite the fact they present marginal difference in their attained data rates, *1.954 kbps* and *2.149 kbps* respectively. Specifically, they present 3.08 % current increase.

Identical findings can be drawn when using reception mode (RX_P), where the impact of *BW* is even more highlighted. More specifically, during reception, we note a decrease of 11.54 % when switching from *500 KHz BW* (TX_M: *4*) to *250 KHz* (TX_M: *2*) and the remarkable variation of *18.9* % when switching from *500 KHz* (TX_M: *4*) to *125 KHz* (TX_M: *1*). Notably, the same variations in TX_M during transmission in the max TX_P provides marginal power consumption decrease of up to *2.03* %. This clearly shows that the power draw of the reception mode, is even more affected by the employed *BW* configuration, which is also noticeable by Fig. 5.3b.

5.5.2 Energy Efficiency per Bit

As frame transmission duration is monotonically related to the configured data rate, it is important to quantify energy efficiency in terms of energy consumption per transmitted bit of information (E_B), as analyzed in our previous works [121, 118], which focus on analyzing the efficiency of the WiFi protocol. We calculate E_B, expressed in *Joules/bit*, as the division of the power consumption values collected for



(A) E/bit in T_X, employing 14 Bytes Payload

(B) E/bit in T_X, employing 44 Bytes Payload

(C) E/bit in T_X, employing 255 Bytes Payload

FIGURE 5.4: Energy Efficiency per bit Measurements in T_X Mode Across Different Payload Sizes

each different TX_M and TX_P by the exact *data rate* value expressed in *bits/sec*.

In Fig. 5.4, we plot the obtained transmission E_B across the available *data rate* configurations and zoom in on the first *7* TX_M settings. We plot the E_B for the payload sizes of *14, 44 and 255 Bytes*, while we omit the case of *24 Bytes* for space reasons. Notably, the dashed lines on the plots represent the respective *data rates* of each TX_M. We notice that higher *data rate* settings always result in lower E_B, which is mainly due to the decreased duration of frame transmissions. The only two cases that present different behavior are the adjacent pairs *3, 4* and *5, 6*. We observe that the calculated E_B in TX_M *4* is marginally higher than that of TX_M *3*, which is also true for TX_M *6* and TX_M *5*. This happens since, TX_M *4* and *6* present higher draw than *3* and *5* respectively, while the achieved data rates are nearly equal. This finding suggests skipping TX_M *3* and *5*, and proceeding to TX_M *4* and *6* respectively, when considering energy efficient rate adaptation algorithms. Moreover, the highest observed variation is up to *15.12* % when comparing the pair of TX_M *5* and *6*, realized in the *+14 dBm* setting when transmitting packets of *14 Bytes* payload and *9.98* % and *11* % for the cases of *44* and *255 Bytes* respectively. Regarding the comparison of the calculated E_B among the illustrated payload sizes we observe an extreme variation. More specifically, when comparing the E_B of *14 Bytes* payload with the one of *44 Bytes* under the TX_M *1* the reduction is *90.14* %. Whilst, the difference is even bigger when comparing the *14 Bytes* with the *255 Bytes* under the TX_M *1*, with reduction reaching up to *98.99* %.

Similarly, we plot the E_B under varying payload sizes in the reception phase, in Fig. 5.5, by zooming in on the first *7* TX_M. The obtained measurements present identical behavior for TX_M *3, 4* and *5, 6* as in the transmission phase. To note that the variation of the E_B for the pairs TX_M *3, 4* and *5, 6* is even higher than the ones in the transmission phase. Namely, it is *27.33* %, *19.17* % and *11.35* % for the cases of *14, 44* and *255 Bytes* respectively, when considering the pair TX_M *5* and *6*. Thus, we further highlight the urge to omit the TX_M *3* and *5*, and select the TX_M *4* and *6* when designing power efficient rate adaptation algorithms. Direct comparison of the E_B among payloads of *14, 24* and *44 Bytes*, during reception, presents remarkable energy

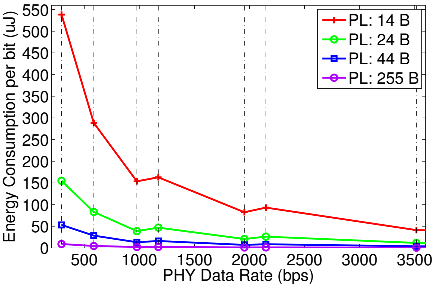


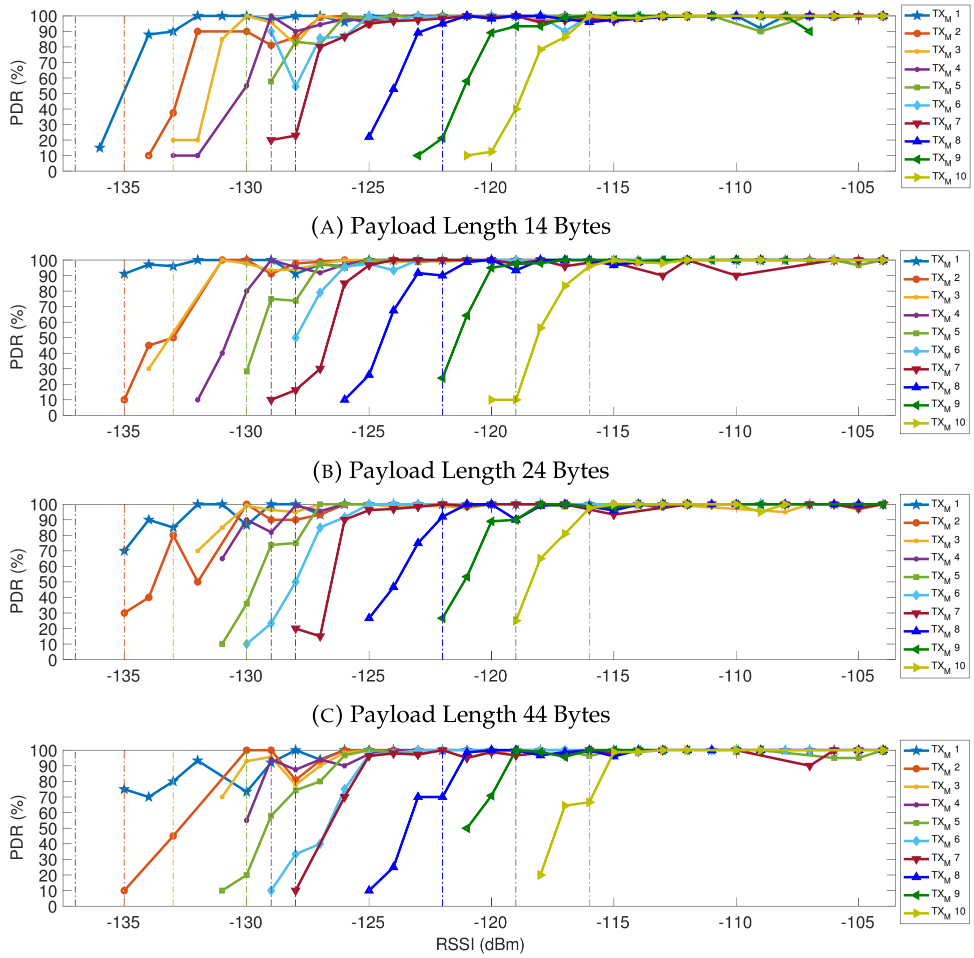
FIGURE 5.5: Energy Efficiency per bit Measurements in R_X Mode Across Different Payload Sizes

efficiency improvement of *71* % and *90* % respectively. The variation is even higher when comparing payloads of *14* and *255 Bytes*, reaching up to *99.78* %. This finding, both during transmission and reception mode, highlights the fact that measurement aggregation techniques can achieve extremely improved energy efficiency in LoRa setups.

Assuming an energy experiment based on our deployed network, in which the sensor nodes propagate their measurements once per hour, we can estimate the lifetime of the network on typical batteries and compare it with the same experiment when employing aggregation mechanisms. In a scenario where a node is propagating *14 Bytes* of information using the maximum TX_P setting, we can calculate that the LoRa chipset will consume roughly *140 mAh* of the total power budget in a year, if we assume ideal conditions in sleep mode and wake-up operation. Considering a power budget of *400 mAh* dedicated only for LoRa transmissions the module's lifetime will be approximately *2.87* years. When employing an aggregation mechanism, grouping several individual measurement frames into a single *100 Bytes* packet the consumed power will be *62 mAh* in a year, which suggests lifetime of roughly *6.38* years. The latter implies an increase of *122* % in lifetime, while aggregation using *255 Bytes* frame will lead to an increase of *160* % in lifetime.

5.6 Experimental LoRa Performance Evaluation

This section presents a thorough analysis of LoRa performance across a wide range of conducted testbed experiments, under uplink communication initiated from by each one of the *10* nodes and destined to the *GW*. The list of varying parameters (Table 5.2) corresponds to a total of *1200* experimental combinations that are conducted every hour, corresponding to *24K* frames transmitted per day. The resulting large dataset includes measurements collected over a period of 2 months, In order to efficiently visualize the wide set of collected measurements, we decided to group the node and TX_P combinations, in terms of the resulting *RSSI* per link. Considering the above, LoRa performance is evaluated in terms of *PDR* and spans the *RSSI* range between *-102* and *-137 dBm*.



(A) Payload Length 14 Bytes

(B) Payload Length 24 Bytes

(C) Payload Length 44 Bytes

(D) Payload Length 255 Bytes

FIGURE 5.6: LoRa PDR Performance Versus Uplink RSSI and TX_M

Figures 5.6a, 5.6b, 5.6c and 5.6d illustrate the *PDR* performance versus uplink *RSSI* and TX_M, across the *4* different tested payloads. The solid lines correspond to the different TX_M (Table 5.2), while the dashed lines refer to the corresponding sensitivity thresholds, as calculated employing payload of *10 bytes* [202] compared to the ones calculated through our realistic experiments (*14 Bytes*). Considering the results across increasing payload sizes, we clearly see that the obtained sensitivity thresholds tend to increase (at maximum by *1 dB* per step), which also verifies the validity of the experimental results. In the extreme cases employing the TX_M *1* and *10*, we observe the remarkable performance of LoRa achieve approximately *90% PDR* at -*134 dBm* and *-118 dBm RSSI* for transmissions at TX_M *1* and *10* respectively, under realistic city-scale experiments. Focusing on the maximum payload case, we can observe that even *-115 dBm* LoRa links are able to transfer frames of *255 Bytes* at TX_M *10* with approximately *100% PDR* that is able to achieve *21.89 Kbps* and can support the transferring of low-quality images or even video streams in real-time. Relating this performance insight with Fig. 5.8a, we clearly notice that *8* out of the *10* considered links are able to support such demanding applications in the maximum TX_P setting.

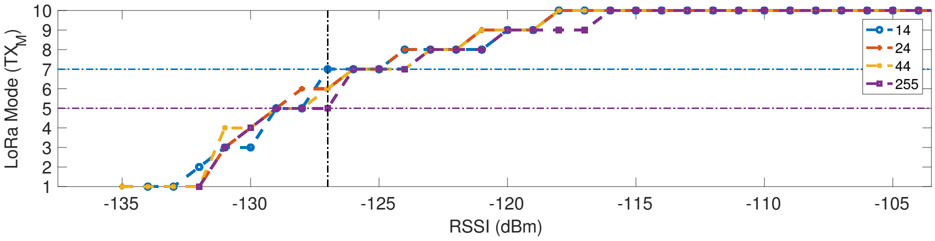
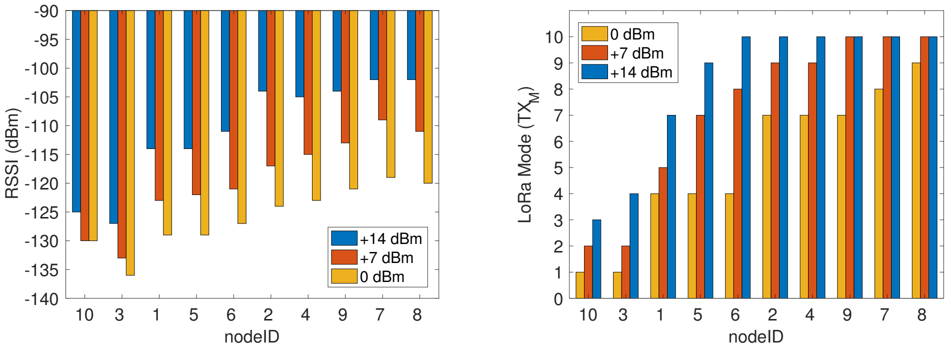


FIGURE 5.7: LoRa Mode Versus Uplink RSSI and Payload Length



(A) Uplink RSSI per LoRa link across different TX_P levels (*0*, *+7*, *+14 dBm*)

(B) Minimum calculated LoRa mode per link, for achieving at least *90* % PDR

FIGURE 5.8: Testbed Devices

In the rest of this section, we attempt to define the optimal mapping of *RSSI* to TX_M, when considering the *30* different link scenarios (*10 nodes - 3* TX_P *levels*), as illustrated in Fig. 5.8a. Following a heuristic approach commonly employed in rate adaptation mechanisms, we identify the best *PHY* layer setting, as the maximum setting able to maintain higher than *90*% of *PDR*. Fig. 5.8b illustrates the mapping of TX_M to the *30* link types, considering the aforementioned rate selection principle. We clearly see that most of the tested links for the *0 dBm* TX_P level, are able to maintain higher than *90*% of *PDR* for TX_M higher than *4* for *RSSI* levels higher than *-129 dBm*. On the other hand, considering the maximum TX_P setting (*+14 dBm*), we notice that most links can maintain the TX_M of *7* for *RSSI* levels higher than *-114 dBm*. Only *6* out of *30* tested links, generated from nodes *3* and *10*, correspond to link quality lower than *-127 dBm* and require the configuration of lower TX_M settings. The sum of collected results highlight the potential of LoRa to achieve stable communication performance for data rates much higher than the basic one, across a wide range of low quality links.

Finally, in Fig. 5.7, we illustrate the detailed mapping of *RSSI* to TX_M, considering the varying frame payload parameter as well. It is made clear that the transmission of larger frames, requires the increase of TX_M, by up to two levels, as in the case of *-127 dBm*, which corresponds to results collected from nodes *3 and 6* under TX_P accordingly. Concluding, we remark that the wide range of collected results highlights the importance of employing automated rate adaptation for LoRa communication that jointly considers the varying *RSSI* and payload parameters.

5.7 Conclusions & Future Directions

In this chapter we presented a city-scale monitoring LoRa-based infrastructure that provides real-time air and link quality indications. Furthermore, we characterize the energy efficiency of the utilized LoRa chipset by conducting in-depth lab experiments. We conclude that the LoRa standard is a very attractive option for deploying energy and cost efficient *IoT* applications in city-scale environments. Specifically, our findings indicate that rate adaptation algorithms and packet aggregation techniques can drastically benefit the energy efficiency of the LoRa protocol, when considering battery powered applications. Moreover, our city-scale experiments suggest that LoRa can support demanding applications achieving impressively high PDR at extremely low RSSI conditions, even when using high data rates. Our future directions include the expansion and the rearrangement of our testbed, so as to extend the range of collected data, specifically of low *RSSI* links and to characterize the performance of the setup across more advanced metrics, such as throughput and latency. Lastly, we plan to conduct additional experiments to characterize the performance of our LoRa network across more advanced metrics, such as throughput and latency.

Chapter 6

NITOS BikesNet: Enabling Mobile Sensing Experiments in a city-wide environment

Contents

6.1 Introduction

Driven by recent technological advances, mobile devices become increasingly sophisticated and miniaturized. Tablets, smartphones and various wearable devices, with embedded cameras, microphones, accelerometers, GPS and other types of sensors, are now part of everyday life. Carried by people everywhere they go, such devices can record a wealth of data, as well as give rise to new applications. For instance, the glasses launched by Google allow users to take photos and videos on the go, just by issuing voice commands. Along the same lines, smart wristwatches or clothes can infer the user's physical activity or detect accidents.

But apart from benefiting the individual, mobile and wearable sensors and information devices can also be useful for the society as a whole. In the spirit of so-called participatory sensing [36], people may contribute to a common goal by recording and publishing information on a voluntary basis, without even knowing each other; for instance, overflowed garbage bins or street potholes can be reported as part of daily commuting activities. More generally, one can view people with their smartphones and wearable devices as a mobile ad-hoc sensing infrastructure, which can be employed, in an opportunistic or targeted way, to perform large-scale sensing tasks. Note that the coverage and/or density that can be achieved this way could very well surpass that of any planned/fixed sensing infrastructure; in particular, since many governments and local authorities cannot afford to setup, operate and maintain nation-wide or even city-scale sensor networks.

At the same time, there is a rising need for the research community but also companies (in particular SMEs) to conduct experiments and pilot deployments in the real world, as opposed to a controlled lab setting. This holds even more for Internet-of-Things and crowdsensing scenarios, where real people with real behaviors and real mobility patterns are required to evaluate different approaches under realistic conditions at a large scale. However, while a lot of work has been done on testbeds that can be used remotely by researchers to run experiments on fixed wireless sensor networks, much less has been done to support experimentation with mobile sensor networks.

In this chapter, we present the NITOS BikesNet framework, which can be used to realize such a testbed at the scale of an entire city, using sensor nodes mounted on bicycles. Our platform is based on a custom wireless embedded node that can host different types of sensors. The sensor nodes are mounted on the bicycles of ordinary people who go about their daily routine as usual. NITOS BikesNet allows external researchers to plan experiments, remotely control the operation of the mobile nodes (e.g., turn on/off individual sensors, or set the sensor sampling period), and retrieve, process and visualize their measurements. The management of the sensor nodes and the data produced by them is done via the cOntrol and Management Framework (OMF) [191] and the OMF Measurement Library (OML) [151], respectively. Leveraging our previous experience in delay tolerant networking (DTN) [232, 139], we extend OMF to handle intermittent and disconnected operation due to mobility, in a largely transparent fashion: on the one hand, each node saves its measurements on local stable storage, and incrementally uploads data to the NITOS server when it has Internet connectivity; on the other hand, the commands issued by the experimenter are queued on the NITOS server, and are forwarded to the sensor nodes when they contact the server.

The main contributions of this chapter are: (i) we identify the most important requirements for a testbed that allows experimentation with mobile sensor nodes; (ii) we describe the extensions made to the OMF framework in order to deal with mobility issues; (iii) we describe the implementation of our sensor node prototype; (iv) we analyze the performance of the node in terms of sensing latency, end-to-end data transfer capability and power consumption; and (v) we present indicative results from a first experiment that was performed using our platform. We believe that at least some aspects of our work have wider applicability, and can inspire the design or enhancement of other testbeds.

The rest of the chapter is organized as follows. Related work is presented in Section 6.2. Section 6.3 lists the most important requirements for a testbed that allows experimentation with mobile sensor nodes, along with the choices we made for the NITOS BikesNet platform. Section 6.4 presents the NITOS BikesNet architecture, the extensions made to OMF to deal with the mobility of the sensor nodes, and the implementation of our sensor node prototype. Section 6.5 discusses the performance of the sensor node, while Section 6.6 presents an experiment that was realized using NITOS BikesNet in the city of Volos. Finally, Section 6.7 concludes the chapter and outlines directions for future work.

6.2 Related Work

Bicycles Related Implementations: To the best of our knowledge, the only system that is similar to our work is [65]. It is based on the Moteiv Tmote Invent platform, and uses a variety of sensors to measure environmental parameters as well as to record data about the performance/fitness of the cyclists themselves. Unlike NITOS BikesNet, the system was not designed as an open testbed that allows external researchers to perform experiments of their own. Also, the sensing devices mounted on the bicycles are not easily customizable, whereas the NITOS mobile sensor node can be equipped with different sensors according to the application's requirements. Another work that focuses on bicycles is presented in [195]. In this case, cyclists use their mobile phone to record and share the paths traveled and ride statistics, which in turn can be used to infer the difficulty and noisiness of the various routes. However, this is a dedicated application for the bicycle owners themselves rather than a testbed infrastructure for running experiments that exploit the sensing capabilities of other people's bicycles. In addition, for all practical purposes, one is limited to whichever sensors are available on smartphones.

Sensing Infrastructures/Frameworks: An exemplary case of a city-scale wireless sensor network testbed is SmartSantander [210], which provides access to numerous fixed and some mobile sensor nodes deployed in the city of Santander. The mobile nodes are based on commercial sensing devices installed on public transport buses and municipality vehicles, and are mainly used to perform air quality measurements throughout the city. Moreover, the node control plane is implemented over GPRS and assumes practically constant connectivity, as opposed to NITOS BikesNet that supports an asynchronous transfer of both control commands and sensor measurements. The testbed management functionality of SmartSantander is based on an adapted version of WISEBED [73], which also supports the reprogramming of sensor nodes over the air. Still, in [210], the creators of SmartSantander acknowledge the importance of adopting a widely used experiment management framework such as OMF, which is embraced by the majority of the FIRE initiative [72] projects. As another example of a city-scale mobile sensor network, [87] describes a system developed for the city of Zurich, Switzerland, where trams carry devices that sense air pollution. However, this is an application-specific system, which uses custom node management software that does not offer any experimentation support to third parties. Another difference with NITOS BikesNet is that the routes and time schedules of the mobile sensor nodes are known in advance. Previous work on the integration of OMF with sensor resources to implement structural health monitoring of bridges [245, 34] extended the framework to support the control of fixed nodes. Unfortunately, it seems that this activity is not continued in the latest release of OMF. Also, this work did not aim to support an asynchronous and disconnected operation of OMF with mobile sensor nodes, as this is the case with NITOS BikesNet. Other notable works that touch a similar direction are the [137, 2, 4, 252].

Smartphone-based Sensing Platforms: A lot of work has been done on mobile sensing platforms based on smartphones. Code-In-The-Air (CITA) [193] allows the user to write sensing tasks for his smartphone in the form of high-level scripts. These are compiled into a server-side and a mobile part, with the latter being shipped and executed on the user's phone. The interaction between the server and the phone is based on an asynchronous messaging service. The Mobile Sensor Data Engine (MOSDEN) [91] is a remote sensing framework for smartphones. It uses the concept of plug-ins to enable a flexible integration of on-board and external sensors without recompiling and/or redeploying the system. MOSDEN implements generic sensing and data storage functions directly on the smartphone, letting each device act as a server that can be used by several applications. With EasyHarvest [101], the user uploads sensing tasks written in Java on a server, which distributes them on smartphones and collects the data produced, in a transparent way. Smartphone owners control the execution of sensing tasks on their devices through a single interface, without having to repeatedly download, install and configure individual sensing applications. Like MODSEN and EasyHarvest, NITOS BikesNet is designed to engage a large number of mobile devices and can thus support city-scale sensing scenarios. While our platform does not support the shipment of application code to the mobile devices, as CITA and EasyHarvest, it lets the user remotely configure their operation. Also, the fact that NITOS BikesNet is integrated with OMF/OML simplifies the organization, controlled execution and monitoring of experiments that involve selected or all mobile sensor nodes.

6.3 Requirements and Implementation Choices

The goal of our work is to provide a platform for easy experimentation and prototyping with mobile sensor nodes mounted on bicycles. We envision a testbed that is open for third parties to test mobile crowdsensing ideas, methods and protocols, under real-world conditions and at city-scale. We also place great emphasis in using low-cost and energy-efficient nodes that can be easily adapted to support different sensing scenarios. Below, we list the most important requirements which we believe such a platform should satisfy, along with the choices we made for NITOS BikesNet.

Experimentation Capability: The user should be able to plan, execute, monitor and control an experiment, from a remote location. This includes the reservation of (perhaps specific) mobile sensor nodes for a timespan, the parameterization of the sensing task, as well as the ability to record, retrieve, analyze and visualize the measurements taken. NITOS BikesNet builds on top of the OMF/OML framework, which is widely used to manage experiments on top of different networking testbeds all over the world. It also provides easy-to-use data visualization tools.

Remote Configuration: The sensor node should work right out of the box, without the bicycle owner having to configure it in any way. The setting of the configuration parameters that drive the operation of the mobile sensor node should be done in a transparent fashion, without requiring physical access. NITOS BikesNet lets the user configure the mobile nodes according to the needs of the experiment, via OMF commands, at any point in time (also during the experiment).

Disconnected Operation: The sensor node mounted on the bicycle is unlikely to have constant or reliable network connectivity. On the contrary, its connection to the Internet will most likely be sporadic and short-lived, e.g. via the open access points encountered in the city. NITOS BikesNet employs delay-tolerant communication techniques to let the transfer of commands (to nodes) and measurements (from nodes) occur in an incremental and asynchronous way, whenever nodes connect to the server.

Support for Heterogeneous Networking Technologies: WiFi is currently the most popular choice for ad-hoc wireless networking in a city, due to the abundance of open access points. Still, it is desirable to let researchers experiment with different networking technologies, either for the communication between the sensor node with the server, or between nodes themselves in the spirit of vehicle-to-vehicle communications. The NITOS BikesNet sensor node is designed to enable simultaneous usage of two separate network interfaces. Currently these can be either WiFi or ZigBee (the respective hardware modules can be plugged/unplugged, at will); we also provide a custom ZigBee access point through which nodes can connect to the Internet. Other technologies, such as Bluetooth and Cellular, could be supported in a similar way; this is future work.

Low-Cost: The cost of the sensor node is crucial, especially if one targets large-scale deployments. Also, having low-cost nodes significantly simplifies their replacement, e.g., in case they brake (due to abrupt shocks/falls of the bicycle), or get stolen (if people forget to remove them when leaving their bicycle unattended). NITOS BikesNet employs a custom-build sensor node that costs less than 100 euros in total. The cost could drop further, if one decides to go for a mass production. Also, the installation of nodes on bicycles does not require any technical skills and has zero cost.

Low-Power: The mobile sensor node should be able to operate for a longer period of time on batteries. Asking the bicycle owner to recharge the node once a day or change batteries every week, could already be quite annoying. This is even more so if people are supposed to be part of the platform for a long period (e.g., a year). The NITOS BikesNet node is lightweight, using components with low-power characteristics and sleep modes; it is also possible to completely power-off individual components. Currently, without any fancy optimizations, and under reasonable assumptions, our sensor node can operate autonomously for about a month.

Small Size: The mobile sensor node should be small in order to be easily mounted on the bicycle without blocking the cyclists' moves. The NITOS BikesNet node is designed to fit underneath the bike saddle, where it is also well-protected against falls and weather conditions (rain). It can be attached to and detached from a bicycle in a few seconds.

Extensibility: It is impossible for a single device to incorporate all the sensors that could be required by different experiments. Ideally, the sensor node should be extensible, making it possible to add new sensors that are not available in the standard package. The NITOS BikesNet node is built in a modular way, so that existing sensors can be replaced with new ones, almost in a plug-and-play fashion. The same applies for the device firmware, which can be augmented with the appropriate device drivers. However, the respective effort and financial costs would have to be covered by the experimenter or some source of funding.

In the next section, we discuss the NITOS BikesNet platform in more detail, focusing on our mobile sensor node prototype and the extensions made to the OMF/OML framework in order to support disconnected operation and opportunistic connectivity of the sensor nodes via WiFi and ZigBee.

6.4 NITOS BikesNet Platform

The NITOS Future Internet (FI) experimental facility [168] is one of the FIRE infrastructures, which is continuously evolving through major extensions reflecting the latest technologies and trends in the FI ecosystem. Currently, NITOS offers several testbed facilities to the research community, featuring technologies in several different areas, such as wireless networks (WiFi, WiMAX and LTE), wired networks, opportunistic networks, software-defined radios and networks, Internet of Things (IoT) and smart city infrastructures. NITOS is open and remotely accessible to any researcher who wishes to deploy and experimentally evaluate networking protocols and applications in real world settings. Users can reserve and control the respective testbed resources through the NITOS scheduler [11], which works in conjunction with the OMF management framework.

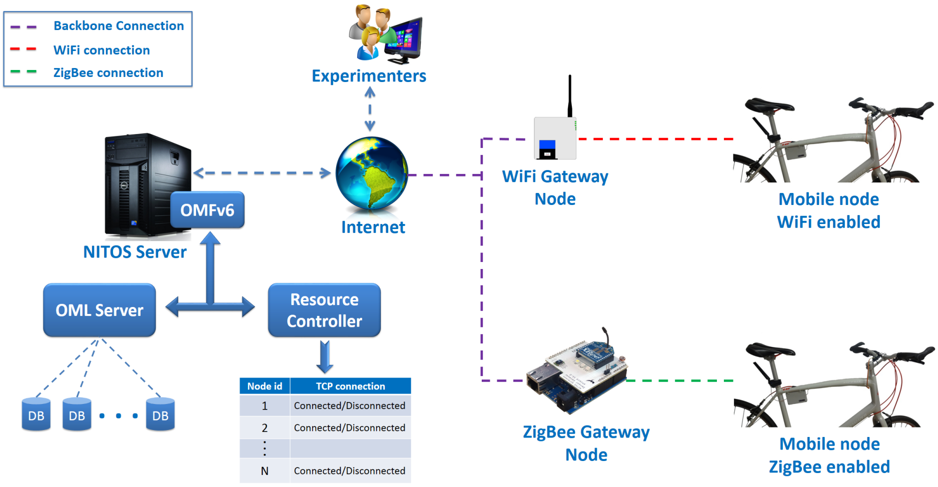
6.4.1 Overview of the OMF/OML Framework

The control and management of the NITOS facility is done using the OMF open-source software. OMF was originally created in the Orbit [194] testbed, and soon became the most widely used tool for experiment control among the majority of the testbeds worldwide. Notably, OMF is the primary experiment control tool in the FIRE initiative [72] as well as in GENI [25].

In particular, OMF enables the experimenter to automate an experiment instead of setting up everything manually by logging into each node to configure/control its operation. The concept is similar to network simulators where the user describes a topology along with the applications that run during the simulation. The difference is that the topology consists of physical nodes on which OMF runs applications like a traffic generator. Also, the measurements are automatically collected with the help of the OML. The configuration and control of node operation occurs through specific properties, which are part of "formal" resource descriptions, and can be done not only at experiment setup but also during experiment runtime.

The basic components of the OMF framework are the Experiment Controller (EC) and the Resource Controllers (RCs). The role of the EC is to orchestrate the execution of the experiments, written in the OMF Experiment Description Language (OEDL). The EC interprets OEDL and sends appropriate messages to the corresponding RCs. In turn, each RC is responsible for abstracting and controlling one or more underlying physical or logical resources. It basically converts the messages received from the EC into resource-specific commands, and relays the response back to the EC. It is important to note that the message exchange between the EC and the RCs is performed using a publish-subscribe mechanism, assuming a stable and reliable communication. Thus, in case of network problems, the messages published by the EC and/or the RC are dropped.

Finally, the measurements produced by an experiment are stored by the respective RCs directly on an OML server, without any involvement of the EC. Given that several experiments can run at the same time, a separate database is maintained for each experiment. The user can inspect and retrieve the results of his experiment at any point in time.



Node id	TCP connection
1	Connected/Disconnected
2	Connected/Disconnected
⋮	
N	Connected/Disconnected

FIGURE 6.1: NITOS BikesNet Architecture

6.4.2 NITOS BikesNet Architecture

The NITOS BikesNet platform consists of three distinct physical entities: the server, gateways and mobile sensor nodes. The NITOS server runs an instance of the OMF/OML framework, appropriately extended to handle node mobility (see next). Gateways provide wireless Internet connectivity to the sensor nodes. They are placed/-fixed at different areas of the city and have a backbone connection to the public Internet. There are currently two gateway types: WiFi access points / routers, and custom-made NITOS ZigBee access points. WiFi is a widely adopted standard with numerous APs available in every city, while ZigBee has a lower power consumption and thus could be an interesting alternative to explore for nodes that run on batteries; also, ZigBee operates at several different frequencies and can avoid the interference in the 2.4GHz unlicensed band induced by WiFi networks. Finally, NITOS mobile sensor nodes are mounted on the bicycles of volunteer participants. They are implemented using a self-designed embedded device and custom firmware that controls the onboard sensors, logs measurements on stable storage and manages the communication with the server, via the gateways. Fig. 6.1 illustrates the system architecture.

To deal with the intermittent connectivity and disconnected operation of mobile sensor nodes, NITOS BikesNet employs a custom-developed RC component, which performs the actual communication with the nodes while hiding any disconnections from the rest of the NITOS platform; as before, the RC itself has a stable connection to the EC and OML components. More specifically, in addition to conventional (synchronous) commands, the RC supports so-called *asynchronous* commands. A conventional command succeeds only if the node to which it is addressed is available (currently connected to the server through a gateway), else it fails; this corresponds to the usual operation of the RC. In contrast, an asynchronous command is deferred if the corresponding node is not available; it is queued within the RC, and is forwarded to the node as soon as it becomes available. Asynchronous commands can

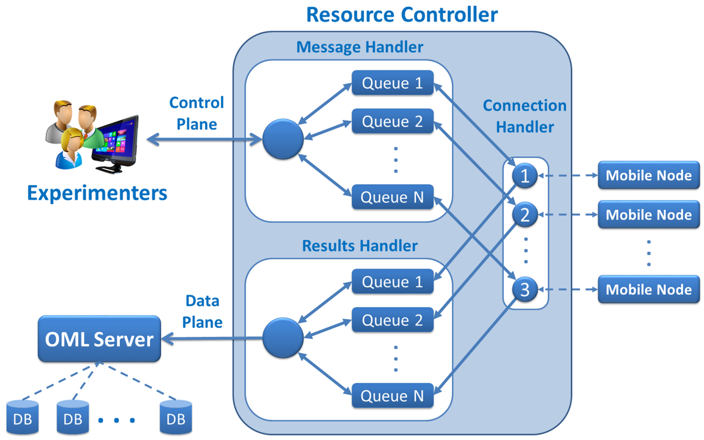


FIGURE 6.2: Resource Controller Architecture

be associated with an expiration time, after which they are dropped from the queue; they are also removed from the queue if in the meantime the user issues a new, competing command that effectively cancels them. Along the same lines, when a mobile sensor node becomes available, the RC retrieves its measurements and forwards them to an OML collection point.

While the user can address/control each mobile node separately, for practical reasons, we interface all nodes via a single RC, which is responsible for keeping track of their status. This way, a single endpoint address needs to be reserved at the NITOS server, and all nodes can be pre-configured with it. Also, the EC can control several nodes via a single command to the RC, which can fan-out this command to several or all nodes. Furthermore, having only one RC conserves resources on the server without incurring significant service delays; provided that the nodes contact the server at different intervals and stay connected for a relatively short duration (which is a reasonable assumption in our case). This said, our implementation can be modified in a straightforward way to let nodes connect to different RCs for better scalability.

The internals of the RC and its interaction with the other components of the NITOS platform are shown in Fig. 6.2. The RC includes handlers for managing the connections with the mobile sensor nodes, processing the control messages coming from the EC, and processing the measurements coming from the nodes. The connection handler is responsible for keeping track of the online status of the nodes, and for forwarding messages and results between the nodes and the other two handlers. The message handler basically keeps a separate queue for each node, where incoming commands are stored when the node is disconnected. Similarly, the result handler maintains a separate queue for each experiment, where it places the respective measurements received from each node, in order for them to be forwarded to the proper OML collection point.

Direction :: Message	Comments
N → RC :: <node id>:HELLO	Node announces its availability. The RC should update the node status internally. Initiate transmission of pending commands from the RC to the node, as well as transmission of pending results from the node to the RC.
RC → N :: <node id>:CMD:<seq. #>:<payload> N → RC :: <node id>:CMDACK:<seq. #>	RC sends the next pending command, and waits for an acknowledgment. Can be repeated several times.
N → RC :: <node id>:RSLT:<seq. #>:<payload> RC → N :: <node id>:RSLTACK:<seq. #>	Node sends next batch of results to the RC, and waits for acknowledgment. Can be repeated several times.
N → RC :: <node id>:BYE	Node informs that it will sign off and become unavailable. The RC should update the node status internally.

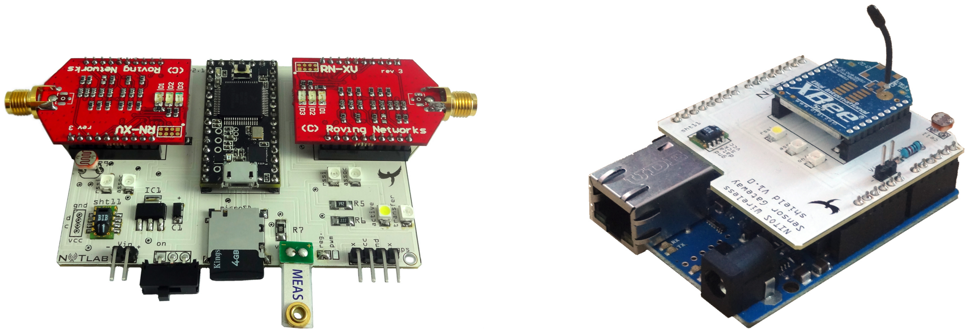
TABLE 6.1: Messaging Protocol for the Communication Between the Resource Controller (RC) and the Node (N)

6.4.3 Sensor Node Protocol

The high-level protocol between a mobile sensor node and the RC is outlined in Table 6.1. The conversation is initiated by the node, which announces itself to the RC via a "HELLO" message. Conversely, it sends a"BYE" message when it intends to end the conversation; there is no negotiation here, as the node unilaterally decides to sign-off. But note that the conversation may also end abruptly, without the node sending a "BYE" message. The transfer of commands to the node and measurements/results to the RC is done via the "CMD" and "RSLT" transaction, respectively. In both cases, the other side must confirm the receipt and successful handling of the message via an acknowledgment ("CMDACK" or "RSLTACK"), else the transaction will be considered as failed and the message will be eventually retransmitted. All protocol messages carry the node identifier in order for the RC to associate them with the corresponding internal data structures. In addition, command/result messages and the respective acknowledgments carry sequence numbers for association and duplicate detection purposes.

Our current implementation uses TCP/IP as the underlying transport service, because of its support for reliable communication, flow-control and full-duplex bidirectional data transfers. Mobile nodes are pre-configured with the IP address and TCP port number of the RC (which runs on a publicly accessible machine with a static address). Nodes connect to the RC when they encounter a gateway and have not communicated with the RC for some time. The connection may remain open for a longer period, in which case the node receives the new commands from the RC as soon as these are issued from the user, and sends new measurements to the RC as soon as these are performed. A node will close the connection when all measurements have been successfully transferred to the RC, and it has not received a new command from the RC for some time. Of course, the connection can brake at any point in time during the above transfers, as nodes may shut down, reboot, or go out of range of the gateway.

Note that the protocol does not make any strong assumptions about the underlying transport. Thus it can be applied, virtually unchanged, on top of different transports besides TCP/IP. In fact, we exploit the "portable" nature of the protocol in how we



(A) Mobile Sensor Node

(B) ZigBee Gateway

FIGURE 6.3: Custom-built Hardware of the NITOS BikesNet Platform

interface the mobile sensor nodes to the RC via ZigBee, as will be discussed in the sequel.

6.4.4 Developed Hardware and Firmware

The NITOS BikesNet sensor node, depicted in Fig. 6.3a, is built by combining different hardware components to create a unified solution that meets the requirements stated in Section 6.3. Our prototype is based on an Arduino-like board and several Arduino compatible modules. We chose the Arduino platform [18] because of its open approach, great flexibility and the large number of publicly available hardware modules and software libraries. In the following, we briefly describe the components of the node and explain how they contribute to the overall node functionality. A high-level component diagram of the sensor node is shown in Fig. 6.4. We also discuss the node firmware and the custom-built ZigBee gateway.

Teensy Microcontroller Board: The main module of the node is a Teensy 3.0 development board [**teensy**], which features a 32-bit ARM Cortex-M4 microprocessor with 128KB of flash (program space) and 2KB of EEPROM (long-term storage) memory. The board also has 34 digital I/O pins (14 of those can be configured as analog ones) and 3 UART ports. It operates at 3.3V, while the microprocessor can be configured to run at 96MHz, 48MHz, 24MHz or 2MHz. The latter frequency can be used during idle periods where high-processing capabilities are not required, to save energy. Furthermore, Teensy can be put in sleep mode to minimize power consumption. Notably, Teensy is fully compatible with Arduino software, and it can be programmed using the same tools.

Wireless Interfaces: In order to give the node sufficient flexibility in terms of networking, it has two sockets for plugging-in communication modules. These are connected to the Teensy via two different UART ports. Each interface has separate hardware connections that are used to control the power state (put the modules in sleep mode) and to monitor the association status of the respective network interface.

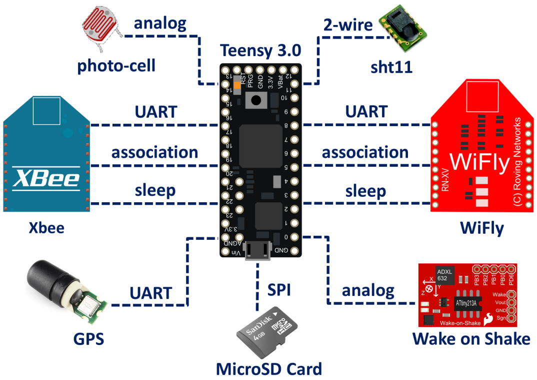


FIGURE 6.4: Sensing's Device Architecture Diagram

We currently employ two modules that support different popular wireless standards: the XBee Series 2 module [**Xbee**], and the WiFly RN-171 module [208]. The XBee module implements the ZigBee protocol on top of 802.15.4 in the ISM 2.4GHz band. The WiFly module implements the 802.11 b/g radio protocol and is compatible with commercial WiFi access points. Furthermore, it incorporates a full TCP/IP stack that can be invoked by the microcontroller of the Teensy. Both modules operate at 3.3V, thus allowing communication with Teensy without requiring an extra hardware component for logical level conversion. They also support a "command" mode, thereby allowing the Teensy to configure and control their operation in a straightforward way.

In our current implementation, the XBee is used to associate with ZigBee gateways through which the node can communicate with the RC. Of course, the WiFly also serves this purpose. In addition, it can be used as a sensor, to scan available WiFi networks. Different node configurations are possible. For instance, one may have a WiFly module acting as a network sensor, and an XBee module or a second WiFly module for the communication with the RC. Alternatively, the node may feature just a WiFly module that can provide both functions, driven by the Teensy as needed.

Sensors: The node is equipped with a GPS unit and temperature, humidity and light-intensity sensors. All modules operate at 3.3V, so they can be directly connected to Teensy. In addition, all modules are powered through an individual I/O pin of Teensy, so that Teensy can turn them off for power saving.

The GPS is a D2523T receiver [248] that provides geo-positioning as well as ground speed and altitude. It also provides exact time/date information, which in turn can be used to timestamp measurements in a globally consistent way. The GPS is interfaced with Teensy through a UART port, and is driven via the tinyGPS Arduino library. Teensy can dynamically turn the GPS on/off to save power.

The air temperature and humidity is measured via a sht11 digital sensor [223], connected to Teensy over a 2-wire interface. The sensor is driven using the sensirion

Arduino library. Finally, a photocell sensor that consists of a light-dependent resistor is interfaced with one of Teensy's analog I/O pins, along with an electrical resistor forming a voltage divider circuit. To acquire luminosity measurements Teensy performs an analog to digital conversion, estimates the voltage drop across the photocell and translates the obtained value into a light intensity value.

microSD Card: The node also features a microSD slot connected to Teensy via the SPI bus. The attached card is powered through an I/O pin of the Teensy, thus it can be turned off as the aforementioned sensor modules. The microSD is used to persistently log sensor measurements, until they are uploaded to the RC. This way the node is able to collect a large amount of data without having network connectivity to the RC. Moreover, the node can be put in sleep mode or be completely turned-off without data loss, which is of crucial importance in order to achieve good autonomous operation on batteries. It also ensures that measurements will not be lost in case the node resets/reboots or runs out of batteries.

Wake-on-Shake: The Wake-on-Shake (WoS) [228] combines a low-power microcontroller with the ADXL362 accelerometer and has extremely low power consumption. It provides an analog pin that is activated in case of an abrupt shake (strong acceleration). We use WoS to awake Teensy as soon as the mobile node starts moving, in order to initiate the sensor measurement process. To achieve this, the analog pin of WoS is connected to an interrupt pin of the Teensy that is pre-configured to awake Teensy in case of sensing a "HIGH" signal. Note that the Teensy cannot acquire vibration measurements by itself when in sleep mode.

Firmware: The firmware of the NITOS BikesNet sensor node controls the entire operation of the device, e.g., it configures and controls the peripheral modules, performs sensing tasks, logs the respective measurements, implements power management, etc. The software is custom-developed but also relies on several open-source Arduino libraries, mainly for controlling the peripherals.

In a nutshell, as long as the node is moving, the Teensy periodically executes a full measurement cycle. If the node stays immobile for a longer period of time, Teensy puts itself and the wireless interfaces in sleep mode while it completely turns off the rest peripherals apart from the WoS. Normal operation is resumed when Teensy is woken up by WoS. Note that this approach is quite meaningful, as it is typically not useful to take many samples at exactly the same location. More importantly, the node remains asleep when it is not moving, which we expect to be the case most of the time (a bicycle is typically used only for short periods during the day). Of course, it is possible to implement more refined/clever policies, which we plan to do in the future.

To use the WiFly as a sensor (for detecting the WiFi networks in range), it is put in command mode and is instructed to perform a scan operation. When the scan is completed, the WiFly responds with the list of available networks. When using a network module (WiFly or ZigBee) as a communication channel, the firmware monitors the association pin to detect that a connection to a gateway has been established. Then, it initiates a conversation with the RC (as described previously). To save power, this is done only if the node has some measurements to upload and it has not already contacted the RC recently, else the module used for communication is put in sleep mode.

As said, the XBee module is used exclusively for the purpose of communicating with the RC (via a NITOS ZigBee gateway). But unlike WiFly, XBee does not come with built-in support for TCP/IP. Fortunately, the firmware can talk the same high-level protocol on top of the XBee application-level transport service; in turn, the ZigBee gateway acts as a proxy for the node over a TCP connection to the RC. On the other hand, an alternative option would have been to use an IP stack for resource-constrained devices, such as 6LoWPAN [164], which allows IPv6 packets to be sent and received over IEEE 802.15.4 networks. However, this would require additional development effort and would substantially increase the complexity of the firmware without providing any extra functionality in our framework.

The association with NITOS ZigBee gateways is done using a pre-configured PAN id, stored in the node's EEPROM. This can be done because all ZigBee gateways are part of our testbed. For WiFi, we obviously want the node to exploit third-party access points that are available all over the city. To this end, the firmware maintains in EEPROM a list of known APs along with the respective passwords. Like other node configuration parameters, this information can be remotely updated by issuing corresponding control commands via the EC/RC.

ZigBee Gateway: As already mentioned, we want a mobile node to be able to communicate with the RC not only via WiFi but also via ZigBee. The respective gateway, illustrated in Fig. 6.3b, is custom built for this purpose. It is based on an Arduino Ethernet board, on top of which we mount a custom shield with an XBee module. The Ethernet is connected to a backbone network through which the RC can be reached. The XBee is configured as a coordinator for a given PAN id. We do not currently use the security support of XBee, but this could be easily done, if required, in the future.

When a node associates with the ZigBee network of the gateway, the gateway initiates a TCP connection to the RC on behalf of the node. From that point onwards, the gateway receives (over the XBee transport) the protocol messages sent by the node, and forwards them (over TCP/IP) to the RC. The same is done for the messages flowing in the other direction. When the association is lost, the gateway sends a “BYE” message on behalf of the node to the RC, and closes the TCP/IP connection. This proxying is transparent for the RC, which handles all TCP connections (whether through WiFly or the ZigBee gateway) in the same way.

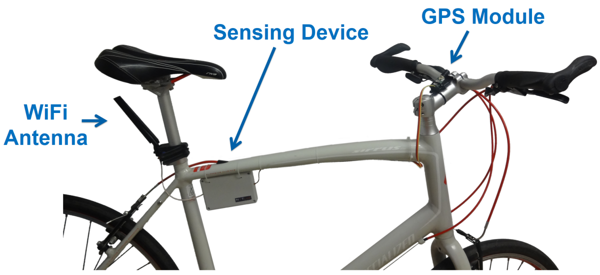


FIGURE 6.5: NITOS Mobile Sensor Node Mounted on Bicycle

Note that in principle the NITOS ZigBee gateway could use a single TCP/IP connection for all nodes; de-multiplexing is possible since each protocol message carries the node id. However, this would be meaningful only if a large number of nodes were likely to use a gateway simultaneously, which is currently not the case in our deployment (we have deployed the NITOS mobile sensor node on just a few bicycles).

6.4.5 Installation of the NITOS Sensor Node

The sensor node, enclosed in a waterproof case, is attached to the bicycle using tire-ups. The antenna for the WiFi radio is fixed beneath the saddle, and is connected via a pigtail to the WiFly interface of the node. Finally, the GPS unit is placed on the handlebar of the bicycle in order to have good signal reception. Fig. 6.5 illustrates the complete setup on one of our bicycles. Note that the most expensive parts, i.e., the node and the GPS, can be effortlessly mounted and unmounted, in a few seconds. This is important since the process may have to be repeated several times during the day, whenever leaving the bicycle unattended.

6.4.6 Visualization Tools

Researchers typically need to visualize and analyze large data sets collected from their experiments, in an easy and flexible way. To this end, we have developed tools that can be used to display measurements as well as to perform different filtering and aggregation operations. This functionality is provided through the NITOS portal using the Google Maps [83] API. Behind the scenes, we provide a mechanism that retrieves raw measurements from the corresponding experiment OML database, and performs the necessary pre-processing before feeding data into the map.

Specifically, the user can select for display specific data sets from those that were produced by the experiment, or view all collected measurements at once. He can also filter measurements based on their values and attributes (WiFi networks, environmental parameters, GPS coordinates, etc.), for instance to consider only the measurements taken in a specific region, or to omit measurements that do not satisfy a property of interest. Moreover, in case of large data sets where it is likely to have several measurements at adjacent locations, one can merge measurements that lie within the same area into a single data point by supplying the aggregation radius.

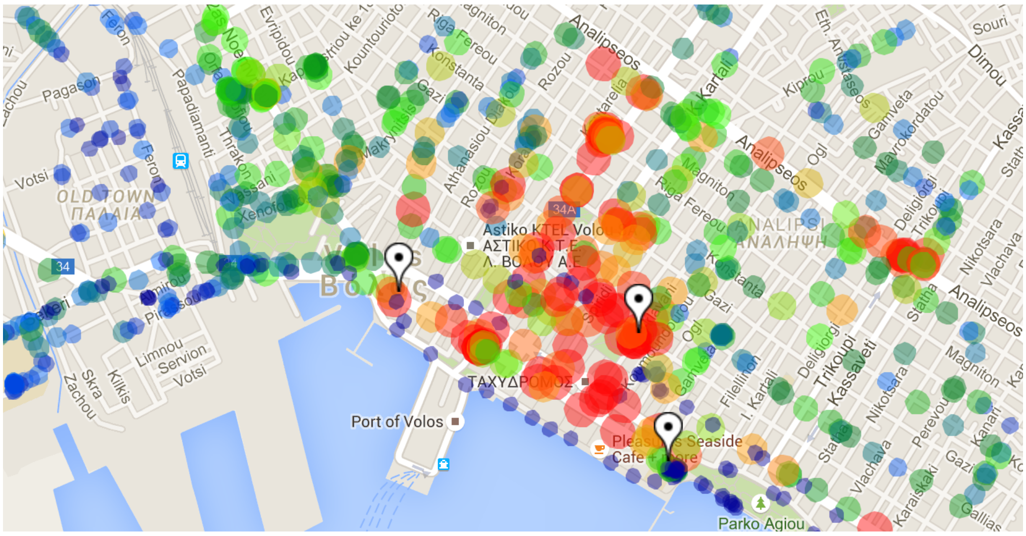


FIGURE 6.6: Measurement Points Loaded on Google Maps API

The maps are interactive, allowing the user to retrieve all the details of a specific data point just by clicking on it. Fig. 6.6 illustrates the developed visualization tool. Of course, the user always has the option to retrieve the raw measurements from the OML database in order to perform his own data processing and visualization.

6.5 Performance of the Mobile Sensor Node

This section analyzes the performance of the NITOS mobile sensor node in terms of sensing latency, communication capability and power consumption. These performance figures give an idea about the potential as well as the limitations of our prototype, and can be used to guide the development of enhanced versions.

6.5.1 Sensing Latency

We have performed several tests in order to measure the latency of each sensor used in our node prototype. The results presented next were derived after repeating each measurement a number of times to eliminate random effects and short-term fluctuations.

Data acquisition from the sht11 sensor requires 319 ms, while 230 μs are needed to perform five consecutive readings of the photo cell (the effective luminosity level is computed as the average of these values). The time for receiving GPS data over UART is 2 ms. Also, it takes about 3180 ms to invoke the WiFi scan procedure and collect the results; this time can vary depending on the number of the discovered networks due to the limited bitrate of the UART port to WiFly. The time required to store all the above measurements (with 10 WiFi access points being detected) in the microSD card is approximately 13 ms. In total, the time required to perform a full sense-and-log cycle is roughly 3514 ms. Hence the minimum sensing period, when using all onboard sensors, is about 4 seconds (the node can perform up to 15 full sensing cycles a minute).

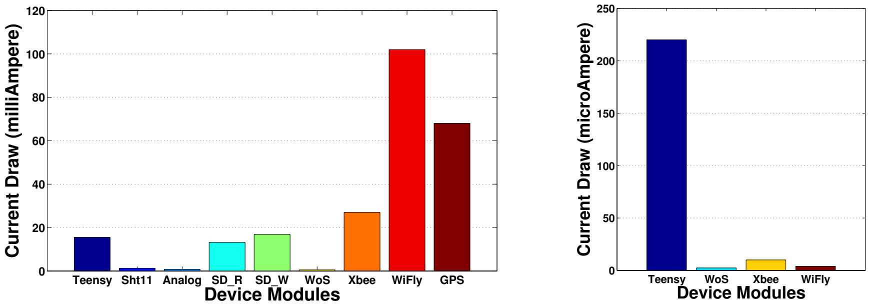
6.5.2 Connectivity & Transmission Latency

An important performance factor is the time required to connect to a gateway, and the data transfer rate that can be achieved, using a given networking technology. For XBee, we measured a network association delay of roughly 7 seconds, whereas for WiFly the delay typically varies between 2 and 4 seconds. The rate at which data can be uploaded to server (RC) depends not only on the nominal bandwidth of the networking technology but also on the bitrate supported by the UART ports and the networking modules of the sensor node. More specifically, while the XBee module has a physical rate of 250 Kbps, it only supports a maximum UART baud rate of 57600, which results in an effective throughput of about 46 Kbps. The WiFly module, which has a nominal physical rate of 464 Kbps and supports a much higher baud rate of 460800 over the UART, achieves a throughput of 358 Kbps.

To get a practical feeling of these numbers, let us assume the mobile sensor node has collected 100 measurements from all onboard sensors; with continuous operation and a sensing period of 10 seconds, the node would take roughly 16-17 minutes to collect these measurements. The total data size is about 91 KBytes. The time needed to upload this data on the NITOS server is close to 16 seconds over XBee, and only about 2 seconds over WiFly. Also, if one takes into account the respective network association delays, the total amount of time for completing the data transfer is roughly 23 seconds and 6 seconds for XBee and WiFly, respectively. Note that the delay for WiFly is quite acceptable, even for performing data upload on the move, since bicycles do not move very fast in a city and thus are likely to remain in range of an access point for some time. Clearly, this is not the case for XBee. However, the NITOS ZigBee gateways are installed only at our office building and a few homes, where bicycles typically stay for hours, giving nodes more than enough time to upload the data collected. We are currently considering different options to achieve better upload performance, e.g., by employing the modern Digi S6B WiFi module [58], which nominally supports a rate of 1Mbps over UART and up to 6 Mbps over SPI.

6.5.3 Power Consumption Evaluation

To measure the instantaneous power consumption of the components used in our prototype device, we follow a widely adopted measurement methodology whereby a high-precision, low impedance current-shunt resistor is placed in series with the power source and the power supply pin of the component to be measured. We use custom-built hardware developed in previous work [121], which enables online and accurate monitoring of the voltage drop across the resistor at the high sampling rate of 63 KHz via the prototype NITOS ACM card.



(A) Active State (B) Sleep State

FIGURE 6.7: Power Consumption of the NITOS Mobile Sensor Node

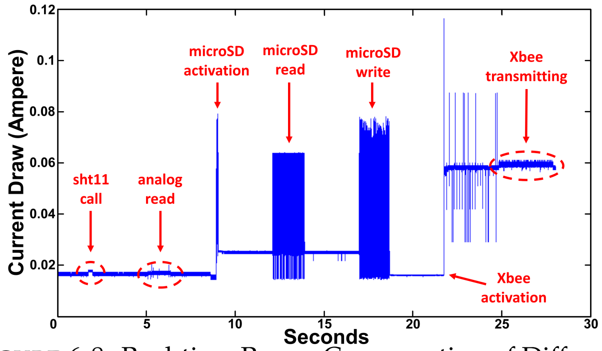
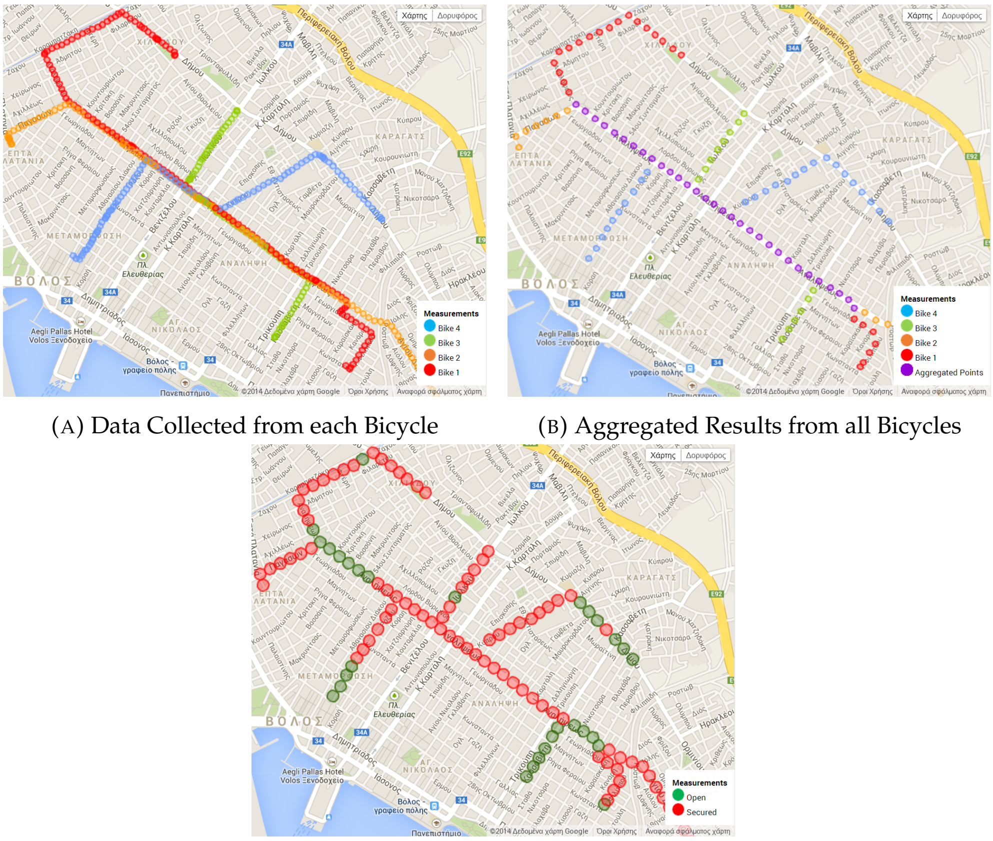


FIGURE 6.8: Real-time Power Consumption of Different Components

The current draw of the components used in the NITOS mobile sensor node, when in active mode, is shown in Fig. 6.7a. As expected, the most expensive ones are the GPS and the two wireless modules. We note that the acquisition of temperature, humidity and luminosity measurements as well as the microSD operations are performed once per sensing period and last only for a short period of time. This implies that their power consumption does not highly affect the node's average power consumption expenditure. Fig. 6.7b shows the current draw of the components that can be put in sleep mode, when in this state: roughly 220 μA for Teensy, 2.35 μA for WoS, 10 μA for XBee and 4 μA for WiFly. The difference compared to the active mode is huge (μA vs. mA). This clearly indicates that significant energy savings can be achieved by putting these components in sleep mode as often as possible.

Finally, Fig. 6.8 plots the current draw of the sensor node while various components are activated to perform typical sensing, storage and data transmission tasks (in this case, the node communicates with the RC via XBee). We note that (for visualization purposes) the footprints shown here correspond to an artificially prolonged operation of the respective components, not to a typical sensing cycle. Based on repeated measurements in active mode (when the bicycle is moving), involving all the onboard sensors and with a sensing/logging period of 10 seconds, we estimate that the device drains 256 mA on average. When put to sleep (the bicycle does not move) the node requires negligible energy. We have equipped the node with 3 type AAA batteries giving a total capacity of 6600 mAh in order to achieve more than 25 hours



(A) Data Collected from each Bicycle

(B) Aggregated Results from all Bicycles

(C) Open/Secured WiFi Networks Discovered

FIGURE 6.9: Visualization of the WiFi Discovery Experiment Results

of continuous operation. Assuming a bicycle is on the move for less than 40 minutes a day, this results in a lifetime of one month; after that, the bicycle owner has to change the batteries of the node.

6.6 A Use Case / Experimentation Scenario

For the time being, we have deployed NITOS mobile sensor nodes on a few bicycles that belong to members of NITlab [167], who volunteered to try out the NITOS BikesNet platform. We have performed a number of small experiments to test and debug the system. Here, we briefly report on one of those experiments.

The objective of this particular use case is to discover and report the available WiFi networks in the city of Volos. In this case, we employed four bicycles. Each cyclist was instructed to follow a different route (the routes where partly overlapping). The sensor nodes on the bicycles were equipped with a WiFly module used to sense WiFi networks, and an XBee module for the communication with the RC. Via OMF, the nodes were configured to perform a sensing operation every 5 seconds (as long as they were moving). The collected measurements were uploaded to the NITOS server at the end of the route, via a ZigBee gateway.

Bike id	Samples	Duration	Distance	Avg. speed	Data Size
1	136	11′:15″	3.7 Km	19.7 Km/h	106 kB
2	82	6′:46″	1.7 Km	15.1 Km/h	67,7 kB
3	127	10′:28″	3 Km	17.2 Km/h	97,9 kB
4	102	8′:25″	2.2 Km	15.7 Km/h	84,7 kB

TABLE 6.2: Statistics of the WiFi-Scan Experiment

Fig. 6.9a depicts the individual measurements collected by each bicycle. All data points on the map are clickable and a pop-up box appears containing information concerning the nearby WiFi networks. An aggregated view of this data is shown in Fig. 6.9b, where adjacent measurements from different bicycles are merged into a single data point in the map. Finally, Fig. 6.9c shows the result of filtering this data in order to distinguish between open and secured networks.

Table 6.2 lists, for each bicycle, the number of samples collected, the duration of the course, the distance covered, the average speed and the total amount of data generated (all sensors were active during the experiment, not just the WiFi sensor). Based on these numbers, one can estimate that the mobile nodes took a measurement every 20-27 meters, and generated data at rates between 1,28 and 1,36 Kbps. By extrapolating these results, one can also estimate that after one hour of biking (with the same sensing frequency as before) each node would produce about 612 kB of data. Uploading this data on the server via a ZigBee gateway would then take a bit less than 2 minutes. This shows that our platform can support high-frequency and long-term sensing tasks, without the respective data uploads being prohibitively expensive.

6.7 Conclusions and Future Work

We presented the NITOS BikesNet platform, which can be used to realize a city-scale mobile sensor testbed, based on nodes that are mounted on bicycles. The experimenter can control these nodes via the OMF/OML framework, in a convenient way. Such a testbed could stimulate research and service provisioning in the areas of mobile ad-hoc sensing, participatory and crowdsensing applications. We also believe that our work can provide valuable insights to researchers working in similar testbed platforms.

Our future plans include the extension of the NITOS sensor node to support additional types of sensors, e.g., for measuring noise, air pollution, and performing spectral scans of the wireless bands. We also wish to test alternative wireless access technologies for the communication between the mobile nodes and the NITOS server, such as Cellular or Bluetooth, as well as to investigate combined usage scenarios. Another item of high importance is the implementation and evaluation of smarter power management policies at the firmware level, to further increase the autonomy of the mobile nodes.

We are also interested in exploring the co-existence of the NITOS mobile sensor node with the smartphones of bicycle owners [209]. The latter could play the role of a gateway to be used for urgent interactions with the NITOS server, e.g., high-priority commands or crucial event notifications. It would also be possible to exploit the smartphone's GPS and accelerometers, as well as its extremely powerful CPU to offload heavyweight processing tasks.

Finally, we are considering more application scenarios that could be supported using our platform, like environmental monitoring, the detection of potholes on roads, or even inferring traffic jams to propose alternative routes. We also intend to make the NITOS BikesNet platform known to the wider community of Volos, in order to attract volunteer cyclists outside the University microcosm but also to inspire people to come up with their own application ideas.

Chapter 7

EVERUN: Enabling Power Consumption Monitoring in Underwater Networking Platforms

Contents

7.1 Introduction

Underwater Networking is a rapidly evolving research thrust, enabling the exploitation and monitoring of the vast natural resources existing in the undersea environment that covers more than 70% of the earth's surface. The key communication technology that enables these applications in the extreme underwater environment, is based on the propagation of Wireless Acoustic Signals. The recent advances in acoustic modem and sensor technologies have motivated the deployment of an increasing number of underwater sensors. Parallel to these technological advances, research has moved into the area of networking, resulting in the deployment of realistic experimentation platforms.

In this context, SUNRISE [234] is the first project offering open experimentation facilities, integrating physical systems with software development into the *"Internet of Underwater Things"*. SUNRISE provides a wide set of different underwater hardware device types that enable experimentation in sea environments. The SUNSET framework [182], developed by SENSES Lab [220], provides all core functionalities for implementing underwater sensor network (UWSN) protocols over multiple layers, considering also various underwater hardware devices. Despite the recent advances in the field of realistic UWSN testbed evaluation, experimental assessment of proposed schemes in terms of energy efficiency still lags behind. Currently, evaluation of energy consumption is only supported in the Simulation framework of SUNSET, thus failing to address the complexity of real scenarios and to support the high level of heterogeneity that characterizes UWSN hardware.

In this work, we develop the EVERUN power monitoring framework and integrate it with the SUNRISE facilities and SUNSET framework, for enabling researchers and underwater equipment vendors to characterize the performance of proposed protocols from an energy efficient point of view. The key contributions of our work are listed below:

- the developed system was integrated with autonomous underwater vehicles (AUVs) and gateway (GW) devices developed by UPorto - LSTS [249], which feature a vast variety of sensing, communication and navigation components.
- novel applications and services were developed on top of SUNSET to support experimental in-field validation, as a fully automated procedure.
- a wide set of experiments were executed under realistic conditions to characterize the power behavior of key UWSN devices at the component level.
- the energy efficiency of important protocol parameters and mechanisms was characterized, resulting in interesting insights.
- specific experiments were conducted both in the testbed and simulated through SUNSET to highlight modeling inefficiencies
- accurate collected power data were integrated with the Simulation framework to improve the energy modeling accuracy.

The rest of the chapter is organized as follows. Section 7.2 lists important requirements and considerations for the development of online energy monitoring tools. Related work is discussed in Section 7.3, while system implementation and integration with SUNRISE appear in Sections 7.4 and 7.5 accordingly. Finally, Section 7.6

presents our detailed experimental evaluation and Section 7.7 concludes the chapter. For simplicity, for the rest of the chapter we refer to the selected assets, AUV and GW nodes, as the devices under test or DUT.

7.2 Key Requirements

In this section, we list the key requirements for enabling online power metering in UWSN testbeds, along with the implementation options that we pursued.

Noninvasive operation: Energy monitoring procedure should not interfere with the normal operation or affect the actual energy consumption of the DUT. This suggests that the monitoring device should be powered from an independent power source, or alternatively the point of measurement should be confined to the component of the DUT to be measured. Moreover, it implies that the monitoring device should not rely on DUT resources for computation tasks, but instead perform the monitoring procedure in an autonomous fashion, so that the consumption of the DUT remain unaffected. In our setup, the point of measurement is properly placed on each module of the DUT and the developed device features an embedded system for the required assignments.

Online monitoring: Online monitoring needs to be applied, in order to enable energy efficiency evaluation under composite configurations and topologies. The proposed framework allows power monitoring in parallel with the execution of real UWSN experiments.

Adaptation to different device types: DUT feature various components that exhibit unique energy consumption profiles, implying that individual components should be independently monitored. Due to this fact, the developed monitoring device follows a rather generic measurement procedure applicable to any type of device, also featuring several individual input channels to enable parallel monitoring of multiple components.

Dynamic range: The components of the DUT may exhibit a perplex array of energy profiles, depending on the application scenario. The energy monitoring system must feature a wide dynamic range spanning the entire spectrum of possible current draws. The proposed device features high resolution ensuring sufficient capturing of dynamic transitions.

High sampling rate: To adequately monitor the consumption of the DUT, even the shortest in duration events of each discrete component should be considered to determine the proper sampling rate. The developed device features a high sampling rate unit to enable precise monitoring of all possible events, over several input channels, which last only a few μs.

Long-term monitoring: DUT are designed to execute missions lasting several hours. To this end, the proposed system should feature sufficient capabilities in terms of monitoring duration. Considering the expected duration along with the defined sampling rate and resolution, we equipped our system with enough storage space to locally cache the acquired measurements, prior to the offloading.

Small-size & low-cost hardware: A small-sized device eases the installation phase, requiring the least possible space for placement, while low-cost hardware allows large-scale deployments. Our monitor is a credit-card sized device, comprised of low-cost components.

7.3 Related Work

Most relevant works in the fields of in-situ power meters have been already discussed in detail in Section 4.2. Therefore in this section we focus on the presentation of UWSN simulation frameworks.

In the field of energy efficiency characterization of UWSN protocols, currently only the DESERT [150] and SUNSET [182] simulation frameworks support power modeling of relevant equipment. Both frameworks extend the open source NS2 simulation engine to support various underwater hardware devices and monitor energy consumption at the device level, when running in simulation mode. Between the two, SUNSET provides a more accurate energy model that also supports the configuration of different transmission levels for the acoustic hardware, while not currently supporting modeling of energy consumption behavior for AUV devices. Nonetheless, both frameworks do not offer support for energy efficiency evaluation during realistic sea experimentation. In this work, we focus on providing realistic energy consumption data for UWSN sea experimentation.

7.4 Framework Considerations

In this section we present the UPorto - LSTS experimental facility, investigating the specifications of the assets under consideration.

7.4.1 UPorto - LSTS testbed

The UPorto - LSTS testbed [149] is targeted at supporting collaborative evaluation and testing of network behavior and control of surface and underwater vehicles. The testbed is remotely accessible, enabling experimenters to evaluate their algorithms and protocols, and supports operations such as environmental monitoring and seabed mapping. It is deployed in the Leixoes marina located 5km south of Porto harbour, in Portugal.

The key components of the testbed are, the Light Autonomous Underwater Vehicles (LAUVs) and the Manta gateway. The first is a torpedo shaped vehicle with one propeller and 4 control fins. It's maximum speed is 2m/s, while it can reach up to 100m depth. The LAUVs integrate a wide set of sensors and systems to deliver the supported functionalities in UWSN experimentation, such as sidescan/imaging sonars,

Component	Type	Supply	Max Draw	Pow. Save	Rate	Range	Resistor
Evologics S2CR Ac. Modem	LAUV & GW	Bat.	80 W	2.5 mW	30 kHz	32000:1	0.02 Ω
Ubiquiti Picostation M2	LAUV & GW	12 V	8 W	-	120 kHz	-	0.1 Ω
Huawei 3G USB Dongle	GW	5 V	2.5 W	-	30 kHz	-	0.2 Ω
GSM & Iridium	LAUV	5V	8 W	-	30 kHz	-	0.2 Ω
Motor & Servomechanisms	LAUV	Bat.	145 W	50 mW	20 kHz	2900:1	0.01 Ω
Embedded PC	LAUV	5 V	10 W	-	120 kHz	-	0.02 Ω

TABLE 7.1: Considered Components and Characteristics per Device

environmental sensors, video cameras, and several other modules. Moreover, they are equipped with a vast number of network devices, like acoustic modems, WiFi, GSM and Iridium interfaces. The second component is the Manta gateway, which is a portable communication hub, supporting several types of wireless and acoustic networks, aiming at providing backbone network to the LAUVs.

7.4.2 Specific Requirements & Considerations

In our work, we identified a specific subset of components that exhibit varying consumption patterns affected by the executed mission's characteristics, and remarkably affect the duration of UWSN missions. Namely, we consider the power hungry LAUV motor and acoustic hardware, along with the wireless networking interfaces and the embedded PC. In Table 7.1, we present the list of considered components per device type. Notably, there are some components that are coupled together in the same power rail, thus we can only measure the aggregated power consumption. More specifically, those components are the LAUV motor along with the servomechanisms that actuate the fins and the GSM interface that is integrated in the same board with the Iridium module.

Each LAUV is equipped with a 7S Li-ion 25.9 V battery pack, while the GW device features a similar pack with lower capacity. As shown in Table 7.1, some of the components are powered by the regulated rails of 5 V and 12 V, while some others directly by the batteries. To this end, for the LAUVs, we require 5 channels for monitoring the current draw of the selected components and 3 additional channels to monitor the different voltage rails, in order to ensure accurate power results. Notably, it is very critical to also monitor the voltage rails, especially in the case of battery powered components since it presents high variations depending on the load and the remaining charge.

To properly determine the required sampling rate for each component we should identify the shortest possible events in terms of duration that actually affect it's power consumption and estimate their length. In the case of the WiFi interface it is the propagation of a small size packet at the highest Physical Transmission Rate (TX_R). For the under consideration WiFi device the maximum supported TX_R is 150 Mbps, when using the MCS7 modulation & coding scheme, while employing a 40 MHz Bandwidth channel. Moreover, the shortest feasible packet is of 300 Bytes, which implies that the time required for it's transmission is 16 μs. Thus, we decided

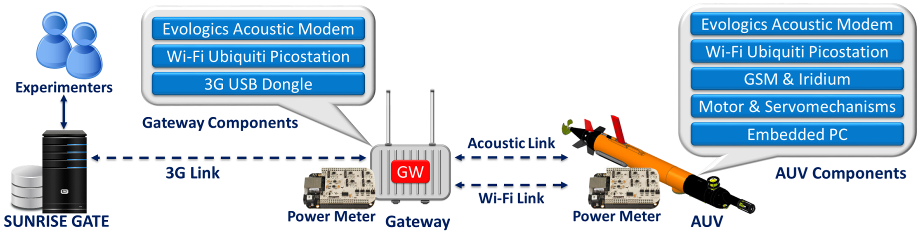


FIGURE 7.1: EVERUN Power Consumption Monitoring Framework, Integrated with the SUNRISE Testbed

to sample the WiFi module at 120 KHz, which is twice the required rate for capturing such events. Concerning the CPU, which varies based on the running process, it is particularly important in special scenarios involving local processing of measurements at the host PC, we decided to use the same high speed sampling, to ensure accurate monitoring. As for the acoustic hardware [67] the shortest event is the propagation of the preamble signal which requires a sampling rate of 30 kHz. Similarly, GSM and Iridium modules require a sampling rate of 30 kHz and the motor along with the servomechanisms a rate of 20 kHz. Notably, we performed extensive measurements with the maximum supported sampling rate in order to derive safe results. Table 7.1 summarizes the extracted sampling rates.

To determine the required resolution we consider the components that feature power saving mode that typically exhibit wide range in their consumption pattern. As shown in Table 7.1 only acoustic modems and motors support power saving state. Acoustic modems may drain minimal power amounts of 2.5 mW when in idle state, while reaching up to 80 W when transmitting at the highest power configuration. This implies that their consumption range can reach up to 32.000:1, which is the highest range of the components under consideration. Finally, in order to achieve long-term monitoring we need to ensure adequate storage space, taking into consideration the sampling rate and the resolution.

7.5 System Implementation & Integration

In this section, we present the architecture of the EVERUN framework, highlighting it's key features. and moreover we discuss the development of the monitoring device along with our implementation choices. Lastly, we present the integration steps towards the implementation of a functional monitoring system.

7.5.1 Energy Monitoring Framework Architecture

The architecture illustrated in Fig. 7.1 comprises of three tiers: the sensing, the intermediate that relays messages from the DUT to the outside world and the user framework. The first tier consists of the power metering device deployed inside the DUT (Fig. 7.3a) and properly connected to measure the components under consideration. The device is connected to the LAN network of the DUT through it's Ethernet port, taking advantage of the existing Ethernet switch that is used to support the

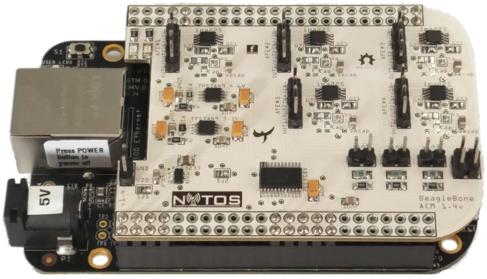


FIGURE 7.2: Developed Power Consumption Monitoring Device

connection of several on-board devices. In addition, the monitoring device hosts a web-service that integrates a set of commands using a RESTful API to remotely configure it's parameters and control the activation/operation of the monitoring procedure.

The second layer is responsible for relaying messages between the user framework and the monitoring devices. To this end, we engaged the embedded PC of the DUT to issue the appropriate HTTP commands, for controlling the monitoring process and allowing the offloading of measurements. When a mission is performed by the LAUVs, the monitoring device is automatically instructed by the host CPU of the DUT to initiate the sampling procedure of the requested components at the specified sampling rate. When the mission has been completed, the DUT commands the monitoring device to halt the sampling operation. In turn, the acquired measurements are stored locally on the on-board storage, ready to be offloaded after the completion of each experiment with the aid of the SCP protocol.

Finally, the third entity is the user interface through which users access the testbed resources as well as design and execute their experiments. In our case, this is the SUNRISE GATE [181], which is a complete tool enabling users to specify the experiment parameters and configure the energy monitoring process. Through the SUNRISE GATE the user can load a completed experiment and replay the missions by using a simple web interface. Upon the completion of the experiment, the data retrieved from the device, processed and uploaded to the SUNRISE GATE server and the user is provided with the energy consumption data that correspond to the assets involved and components selected. When the user replays a mission the energy data illustrated in the form of a live line chart that is populated as the mission progresses.

7.5.2 Energy Consumption Monitoring Device Development

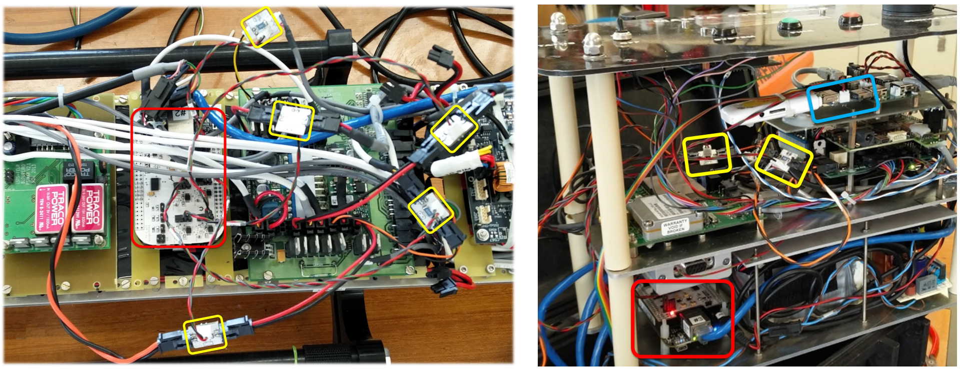
The principle followed to measure the power consumption is the placement of a precise, low-impedance shunt resistor, in series with the component to be measured. The voltage drop across this resistor is proportional to the current drain, according to Ohm's law, enabling the extraction of power data. More details can be found in our previous work [121].

Embedded PC: The monitoring device is composed of different hardware and electronic components, constituting a unified solution able to address the requirements specified in Section 7.2. The core module is the BeagleBone Black Rev. C [28], which is a low-cost, embedded platform characterized by sufficient processing power capabilities (1GHz with 512MB RAM), low-power consumption and several communication interfaces. BeagleBone also supports several operating systems; in our prototype we used the Linux-based Debian OS [55] optimized for the BeagleBone hardware. All of the electronic components used for the power consumption sampling are hosted on a custom-made printed circuit board (PCB) daughterboard mounted on top of the embedded host device, as illustrated in Fig. 7.2. The embedded platform is responsible for controlling the peripheral units and implements the software architecture for the energy monitoring system. Furthermore, the device features an external microSD, used to locally cache the acquired measurements, prior to the offloading process. The BeagleBone along with the on-board electronics is of small size enabling ease of installation and inducing the overall low cost of 80 euros.

Analog to Digital Converter: To convert the analog signal to digital we employed the Texas Instruments (TI) ADS8332 ADC IC [59]. The selected ADC features 8 input pins connected to it's internal multiplexer that enables switching among the available signals in order to allow for parallel sampling of multiple components. Moreover, it features 16-bits resolution and high SNR of 91 dB, while performing conversions at 500 kSamples per second (kSps). The overall sampling rate is split among the configured channels. It supports 8 discrete channels, 5 for current and 3 for voltage measurements. Moreover, it achieves twice the maximum required range (64.000:1), while also exceeding the required aggregated sampling rate. We opted for the ADS8332, because it complies with our defined requirements described in section 7.4 and clearly outperforms the BealgeBone's embedded ADC unit. Finally, given the specified sampling rate and speed and taking into consideration that a typical mission lasts up to 7 hours we calculated that our device will generate roughly 16 GByte of data in case of continuous monitoring. Thus, we attached a 32 GByte memory card to ensure long-term monitoring capabilities.

Analog Front-End: The analog front-end is a circuit that scales the observed voltage drop across the shunt resistor to a full-scale analog signal to ensure optimum signal-to-noise ratio (SNR) and enhances the accuracy of the sampled data. Notably, the observed voltage drop signal varies from a few mVolts to a hundred mVolts, while typical ADCs are sampling signals of a few Volts. Thus, it is imperative to use an amplification circuit, to deliver the required signal scale. In our implementation, we decided to utilize the TI INA225 [240] current-sense amplifier, which is based on a low-offset, zero-drift architecture supporting wide Bandwidth range of 250 kHz, thus making it ideal for our system and the defined requirements.

Supply & Reference Voltage: In our application we power the electronics by LAUV's 5V power rail that is used to power several on-board modules, hence it presents high



(A) LAUV Vehicle Installation (B) Manta Gateway Installation

FIGURE 7.3: Integration with SUNRISE Assets

variations in the voltage and features high-noise, induced by high-frequency components. The selected ICs are highly sensitive, thus we integrated two TI TPS7A49 [197] ultra-low noise, high PSRR (Power Supply Rejection Ratio), regulators. One for powering the ADC IC and another one for the Analog Front-ends. This ensures a clean voltage rail and noise filtering. Moreover, the ADC converter requires a highly precise voltage reference, in order to accurately perform the conversion to a digital value. To this end, we employed the TI REF3233 [241] IC that delivers highly constant output, irrespective of the load on the device, power supply variations and temperature changes.

Custom SPI Implementation: The communication of the ADC IC with the host embedded device is realized through the SPI protocol. The native SPI interface speed on embedded devices, such as the Beaglebone board, imposes certain limitations in terms of the maximum throughput and the jitter between measurements. Notably, when using the native SPI interface we achieve sampling rates of up to a few kSps, thus bottlenecking the ADC. To this end, we utilized a unit integrated in the processor of the Beaglebone, called PRU (Programmable Real-time Unit) [24]. A PRU is a fast (200MHz, 32-bit) processor, programmed in ARM Assembly, ideal for applications that require high speed and precision. We implemented the SPI protocol on the PRU enabling fast (500 kSps), jitter-free and continuous communication between the embedded device and the ADC IC.

Notably, the developed power meter successfully addresses the requirements defined in section 7.4.

7.5.3 Power Monitoring Device Installation

In order to proceed with the integration, we placed individual shunt resistors per considered component, as shown in yellow color in Fig. 7.3a and 7.3b in both LAUVs and the Manta gateway. Notably, the power meter is highlighted with the red frame, the shunt resistors with the yellow and a USB extension board with an on-board shunt resistor to measure 3G Dongle's power with the blue one. The power monitoring device is connected with each resistor to monitor the differential signal.

Configuration	Level	Power Consumption
Tx SPL	Max	29.95 W
Tx SPL	Max -6 dB	9.34 W
Tx SPL	Max -12 dB	4.34 W
Tx SPL	Max -20 dB	2.76 W
Rx Gain	Max	2.26 W
Rx Gain	Max -20 dB	1.98 W

TABLE 7.2: Power Consumption of the Evologics S2CR 18/34 Modem Under Various SPL and Gain Configurations

Moreover, the device is connected with the three different voltage rails to acquire voltage measurements. In order to select appropriate resistor value per component, we conducted a thorough in-lab investigation of the energy profile under high loads, to determine the maximum achievable drain per device component. Based on the above and considering the maximum supported input voltage of the INA225 monitor in our setup (132 mV), we calculated the appropriate resistance value, as shown in Table 7.1. To note that the installation is as less invasive as possible, while the application of the framework does not interfere with the operation of the assets.

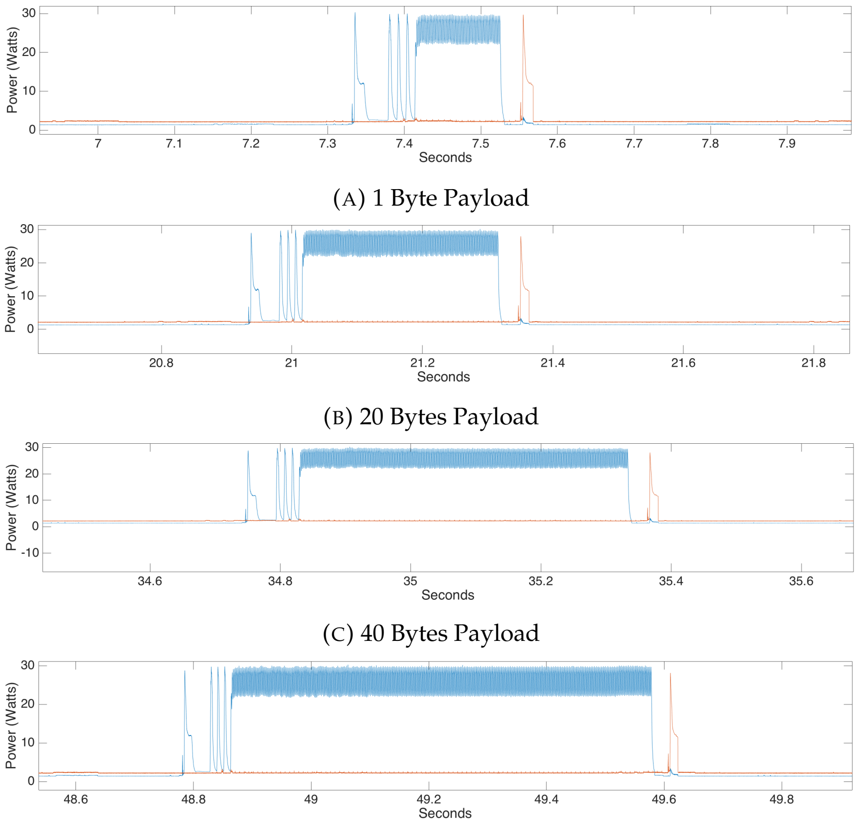
7.6 Experiments

Extensive UWSN experiments were conducted within the Leixões marina located in Porto, Portugal with the support of the UPorto - LSTS team. The employed devices include two Noptilus LAUVs and one Manta GW, which are all equipped with the developed power monitoring equipment. In the following experiments, we highlight how the consumption of each different device component varies when executing a real UWSN mission in the water.

7.6.1 Acoustic modems

6.1.1 Static channel conditions

In this first set of experiments, we focus on highlighting the device settings able to significantly impact the power consumption of acoustic hardware. For this purpose, we setup an acoustic link between two Evologics S2CR 18/34 modems under static channel conditions and simultaneously measure power consumption at both the transmitter and receiver side. Having experimented under various configurable parameters on Evologics equipment, we identified that the two core settings able to impact consumption are the Source Level affecting the sound pressure level (SPL) during transmission and the Gain Level that adjusts the input amplifier gain during reception. To experiment with these two parameters, we transmit saturated traffic between the two modems under the fixed PHY rate of 976 bps, by employing the "Send Instant Message" mode of operation. Under this setup, we vary the SPL and Gain configurations among the supported levels for both link sides and we illustrate the collected power measurements in Table 7.2.



(A) 1 Byte Payload

(B) 20 Bytes Payload

(C) 40 Bytes Payload

(D) 64 Bytes Payload

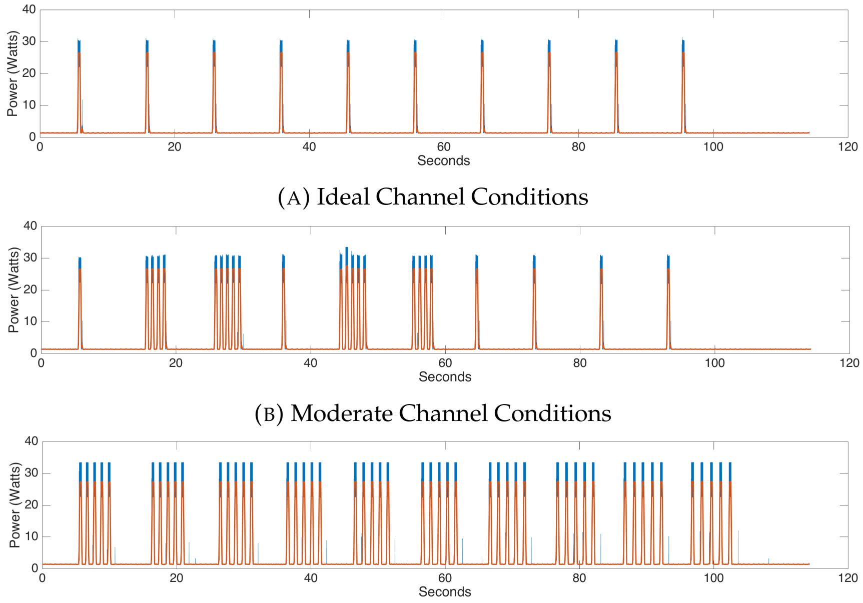
FIGURE 7.4: Power Consumption of the Evologics S2CR 18/34 Modem Under Various Frame Length Transmissions

Payload (Bytes)	Tx Energy (J)	Rx Energy (J)	Tx Energy Effic. (mJ/bit)	Rx Energy Effic. (mJ/bit)
1	3.56	0.44	445.4	15.4
20	8.5	0.85	53.17	12.6
40	13.78	1.32	43.08	11.93
64	19.29	1.78	40.19	11.57

TABLE 7.3: Energy Consumption across Varying Payloads

Apart from considering the acoustic hardware settings directly impacting power drain, we also focus on parameters affecting overall energy consumption. As energy consumption is directly related to the duration of the frame exchange, we focus on the two key related parameters, namely the employed transmission PHY rate and the total length of exchanged frames. Evologics modems implement dynamic PHY rate adaptation through proprietary algorithms by default and do not offer configuration of PHY rates on demand, apart from the option to employ the "Send Instant Message" mode. In an effort to investigate the impact of frame exchange duration on energy consumption, we use the aforementioned setup to vary the length of injected frames, up to the maximum supported length of 64 bytes.

In Fig. 7.4, we plot the instantaneous power consumption of the two modems under 4 indicative payload sizes of 1, 20, 40 and 64 bytes, where the blue and red lines illustrate the power consumption at the TX and RX side respectively. We clearly



(A) Ideal Channel Conditions

(B) Moderate Channel Conditions

(C) Lossy Channel Conditions

FIGURE 7.5: Power Consumption of the Evologics S2CR 18/34 Modem Under Various Channel Conditions

observe that each transmission consists of a fixed first part related with the frame preamble lasting approximately 75 ms and a second part that carries the actual data, whose duration depends on the payload length. In Table 7.3, we characterize the amount of energy that is consumed at each side, in terms of total consumed energy (J) and energy efficiency (mJ/bit). The presented results quantify the improved energy efficiency (91 % between the 64 and 1 byte cases for the TX part) that higher payload transmissions are able to offer, by amortizing protocol overheads, such as frame preamble, headers, etc. Our analysis can be further applied to study the energy impact of vendor specific techniques, such as frame aggregation, forward error correction and data compression.

6.1.2 Varying channel conditions

The second set of experiments on acoustic hardware focuses on highlighting the energy efficiency performance of different protocol configurations under varying channel conditions. For this purpose, we design an appropriate experimental scenario that relates the impact of retransmission attempts with energy consumption. Evologics modems support the configuration of the maximum number of retransmissions the device will perform to deliver an instant message after a failure. We exploit this feature and fix the number of maximum retries to 4, to experiment in a scenario including the transmission of 10 instant messages carrying a payload of 20 bytes at an interval of 10 seconds, while setting the maximum SPL-Gain settings. We repeat the experiment under three types of channel conditions, namely ideal (no retries), intermediate and lossy (max retries attempted per frame), by varying the distance between the acoustic modems.

Motor RPM	0	600	800	900	1000	1100	1200	1300
Power (Watts)	3.05	14.52	23.39	30.36	38.12	46.94	59.54	71.12

TABLE 7.4: LAUV Motor Power Consumption across RPM Levels

In Fig. 7.5, we plot the instantaneous power consumption of the transmitting modem, across the three different scenarios. Considering that the maximum number of retries is fixed to 4, we observe that the number of attempted transmissions varies between 10 and 50, depending on the channel conditions. Focusing on the moderate channel conditions scenario, we see that a total number of 24 transmissions were attempted. In terms of energy consumption, the framework reported the values of 245.05 J, 364.06 J and 595.94 J, under ideal, moderate and lossy channel conditions accordingly. Given the fact that the Evologics software does not report the actual number of performed retries per failed delivery, we understand the inability to build an accurate energy evaluation tool, even when fed with highly accurate power values per device state, as the ones reported in Table 7.2. The inefficiency of such a static power model would induce modeling error of 48.5 %, when considering the ideal channel model instead of the moderate one. This finding highlights the disadvantage of modeling approaches followed by simulator and emulation based systems in deriving the power consumption under realistic experimentation conditions. Through the EVERUN, we are able to capture events related to specific protocol implementations, such as retransmission policies, transmission power adaptation, etc. and quantify their energy impact towards aiding in the development of energy efficient protocols.

7.6.2 LAUVs

6.2.1 Real experiments

We start our experimentation with the Noptilus LAUVs, by investigating the vehicle parameters remarkably affecting power consumption. To this aim, we use different components, such as the motor, servos, under various settings e.g. depth or water current and evaluate the resulting power impact. Through this investigation, we identified that the key setting able to affect consumption is the motor RPM (revolutions per minute), while the other parameters can only pose minimal impact. We characterize the impact of motor RPM, by executing an experiment with the Noptilus-1 LAUV, configured to move at the surface level when no water current is evident and vary the motor RPM across the values of 600 and 1300 that enable the device to start moving and 0 RPM to measure idle power consumption. Table 7.4 summarizes the collected power data, showing the wide range of power drain across different RPM settings. The power data appear to far exceed the results collected under identical RPM settings when applied in the lab with the LAUV not being in contact with water and the drift it results in.

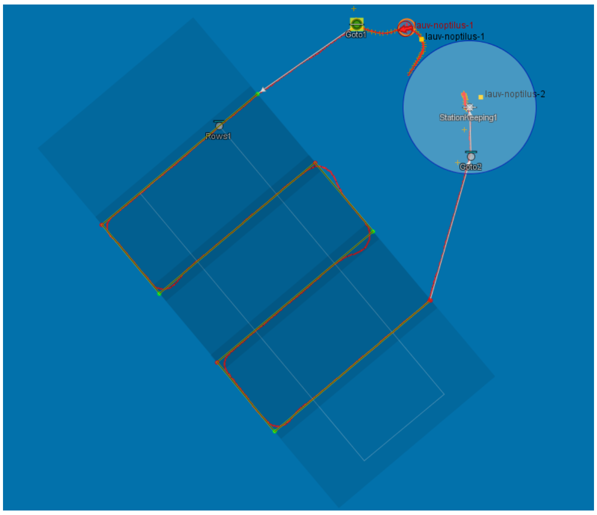


FIGURE 7.6: LAUV Route

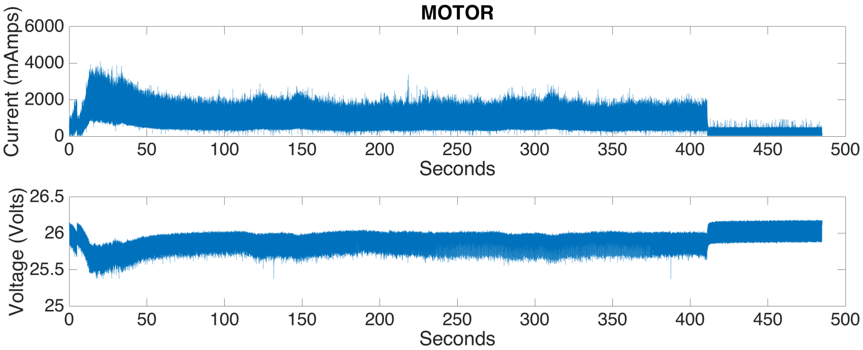


FIGURE 7.7: LAUV Motor Power Consumption

We proceed with our second LAUV experiment, by instructing the Noptilus-1 LAUV to use the fixed speed of 1 m/s, while following the specific route illustrated in Fig. 7.6 and consisting of 4 individual maneuvers. In the experimental logs, we observe that the mission is completed in 485 seconds and that the motor's RPM varies between 800 and 1100 in an effort to maintain the desired speed. Having processed the collected power data through the developed Matlab script, we generate the plot presented in Fig. 7.7 and clearly observe that the monitoring duration matches the experiment duration, verifying the achieved time synchronization exactly.

Focusing on the voltage subplot of Fig. 7.7, we observe high variation between 25.4 V and 26.2 V (voltage difference of 0.6 V), as a consequence of the voltage drop occurring on the internal resistor between the power supply and the monitored component. This finding highlights the importance of monitoring the delivered voltage rail in parallel with the current draw, in order to increase the overall power monitoring accuracy. Considering a fixed voltage of 26.2 V, we would estimate the max instantaneous power consumption of ∼104.8 W instead of 101.6 W inducing a measurement error of ∼3.15%.

Upon observation of the power subplot, we understand that the LAUV is moving for approximately 411 seconds, while it remains idle for the rest 74 seconds, as it

Phase	Real Duration (s)	Simulated Duration (s)	Power (Watts)
Initial	0-50	0-50	44.69
Constant	50-411	50-378	23.89
Idle	411-485	378-485	1.95

TABLE 7.5: Power Consumption Measurements of LAUV Motor Experiment Per Phase

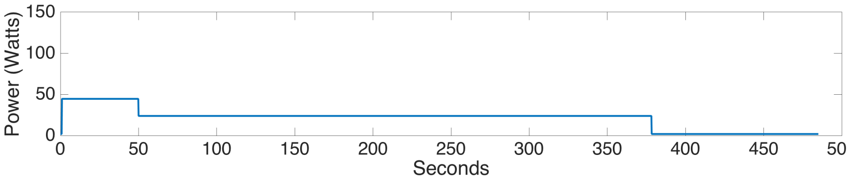


FIGURE 7.8: Simulated LAUV Motor Power Consumption

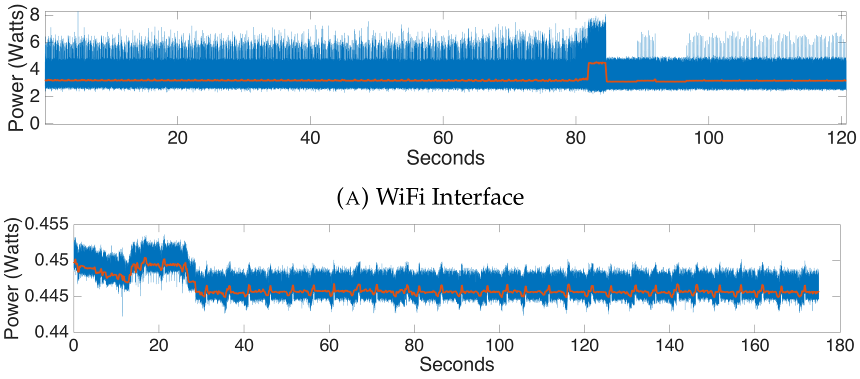
has already reached it's final waypoint. The red line is used to point out the average power value calculated as a moving over a window of 1000 samples. During the moving phase, we see that the LAUV motor power consumption is quite stable apart from the initial phase, where higher power drain is observed during the speed increase until the target speed of 1m/s is achieved.

6.2.2 SUNSET Simulated experiments

In an effort to compare the real power data, we proceed with the simulation of the last mission in the SUNSET simulator framework. SUNSET currently supports a simple mobility model for a generic LAUV. It allows to specify the way-points that the LAUV has to follow with the relative time in which the LAUV can stay fixed at each specific way-point (named in the following as station-keeping). Based on the LAUV's speed (specified in m/s) it is possible to track it's position over time, given the restrictions of modeling movement as a linear line and not considering any other effect, such as current drifting. The real mission can be divided in three phases, as presented in Table 7.5 along with the individual duration and average power consumption, as derived through our calculations.

In an effort to increase the energy modeling accuracy of UWSN equipment in the SUNSET simulator, we incorporated a wide list of power data collected during real experiments (presented in Tables 7.2, 7.4), through the EVERUN framework. Moreover, we extended the simple energy model of SUNSET to consider the three individual states, namely the initial (starting from station-keeping), the constant during which the LAUV moves at a fixed speed and the idle while the LAUV remains in the station-keeping mode.

To demonstrate the extended energy model for the case of the LAUVs, we configured the in-field measurements presented in Table 7.5. The LAUV motor power consumption, as derived through SUNSET is presented in Fig. 7.8, clearly showing that the LAUV reaches it's final destination 33 seconds earlier than the one in the real experiment. This is due to the simplified path followed in the simulation,



(A) WiFi Interface

(B) 3G Interface

FIGURE 7.9: Power Consumption of LAUV's Wireless Communication Interfaces

which slightly differs from the real one followed by the Noptilus-1 LAUV (see Fig. 7.6). However, the overall simulated consumption for the total experiment duration is 10285 J, while the amount of the energy measured through the power monitor is 10789 J. This minor difference of ∼4.9% verifies the ability of a simple energy model to estimate consumption quite accurately when fed with real measurements collected in-field.

7.6.3 Wireless communication interfaces

In this last set of experiments, we start by monitoring the exchange of data over WiFi, between the Manta GW and the Noptilus-2 LAUV. This type of communication is feasible only when the LAUV is moving at the surface level and is used to transfer protocol messages related to scheduled missions or to offload collected sensor measurements. The experimental logs show that the mission starts with the LAUV moving at the surface level up to approximately the 85th second, where the LAUV submerges and continues the mission underwater. In Fig. 7.9a, we plot the power consumption of the WiFi interface. It is evident that frames are frequently exchanged until the LAUV submerges, while later WiFi activity decreases. The WiFi power consumption occurring while underwater is related to periodic transmissions of WiFi probe frames, in an effort to re-establish the WiFi link. We note that the consumption of the WiFi link ranges to a few Watt, since we refer to an entire WiFi embedded system and not to a NIC.

Next, we focus on the consumption of the 3G interface of the Manta GW, which is used to provide remote access to the LAUVs over the Internet. Fig. 7.9b illustrates the collected power measurements for the 3G interface, clearly showing intervals of varying consumption related to the varying amounts of traffic transmitted over 3G. Power monitoring of wireless interfaces can be employed to characterize the energy overhead that the offloading of collected data can pose on the overall energy consumption of underwater missions. Moreover, it can aid in developing mechanisms able to dynamically select the employed technology for data offloading, towards minimizing energy expenditure.

7.7 Conclusions

In this work, we introduced the EVERUN framework that enables researchers to characterize the performance of proposed UWSN protocols from an energy efficient point of view. The developed system was integrated with real LAUV and GW components of the SUNRISE testbed-site in Porto and the SUNSET experimentation framework. Through the execution of a wide set of remote experiments, under realistic conditions, we characterized the power behavior of key UWSN hardware and collected valuable insights about the energy efficiency of important protocol parameters and mechanisms. Inefficiencies of simulation-based energy evaluation tools were highlighted, while the delivered modeling accuracy was improved through the incorporation of accurate power data collected through EVERUN. Notably, there are several works that can employ our developed tool to characterize the energy expenditure of their devices such as the [199, 198, 23, 42].

Chapter 8

On the Development of Energy-Efficient Communications for Marine Monitoring Deployments

Contents

8.1 Introduction

The unleashed potential of the ocean ecosystem has raised the interest of the research community, towards understanding and exploiting the vast resources existing in the aquatic environment. Notably, oceans cover 71% of the globe, while most of it's scale remains unexploited in several fields. Undeniably, it is urgent to understand and interpret the aquatic ecosystem, since it plays a significant role in several sectors, such us fisheries, oil, gas, seabed minerals, etc., and impacts every facet of our society in general.

To address this challenge, it is imperative to keep track of oceanic and coastal areas and assess how these regions are evolving. Towards this direction, scientists all over the globe deploy permanent observatory systems (buoys) in marine waters in order to acquire long-term environmental data in real-time. Along the same lines, some institutions [149] employ Autonomous Unmanned vehicles (AUVs) to perform underwater survey missions such as detecting submerged wrecks or mapping the structure of seafloor. Of course, AUVs also embed a vast number of environmental sensors to measure the concentration of different elements at various points. This need, is also highlighted by the fact that several initiatives [53, 79] aim at establishing Earth observation services, freely accessible and at a large scale. Other notable works present similar implementations [95, 222, 213].

Despite the recent technological advances and the need for real-time, in-situ monitoring the aforementioned observatory systems still present some inefficiencies. One key parameter of such systems is their backbone communication link, which is used for measurements offloading as well as for configuring the sensing setup. Typically, observatory systems exploit either WiFi or GSM technologies. The first is widely used when a system is deployed in a coastal zone, where it is likely to experience line-of-sight with a shore gateway, while the second one is mostly used in far-located systems. Both technologies are targeted at specific applications, while the latest protocols present improved features in terms of achievable throughput and power consumption.

To address this challenge we propose a novel architecture, employing several communication interfaces, divided into a set of low-power ones and a group of high-throughput ones. Our principle is to deactivate all power-hungry interfaces and peripherals when not required. For example, high-throughput interfaces are only activated for transferring large files, while the low-power interfaces remain active to wait for incoming requests dedicated to performing only lightweight tasks. Relying on this, we have developed an energy-efficient communication device, that supports both low and high rate technologies. To evaluate our system we installed a stripped-down version in the Vida oceanographic buoy [250], in Piran, Slovenia.

The rest of the chapter is organized as follows. Section 8.2 lists important requirements and considerations for the development of such system. Related work is discussed in Section 8.3, while system implementation and integration appear in 8.4. Finally, section 8.5 presents our evaluation measurements and Section 8.6 concludes the chapter.

8.2 Key Parameters & Considerations

The nature of ocean observatory systems is characterized by intermittent connectivity, limited energy budget, and varying network conditions. To this end, we should thoroughly consider all the above when designing a communication system for marine monitoring.

Energy-Efficient Operation: Observatory systems typically harvest energy from ambient sources, since they operate unattended for long periods of time, without fixed power connection. Moreover, it is impractical or even unfeasible to visit and replace their batteries.

Even if possible, a power efficient system, would require less frequent battery substitutions, which would remarkably reduce the attendance cost of the system. This implies that the communication system must be power-efficient and drain as less power as possible. In our implementation we adopt some low-power protocols to ensure on-line communication with a power-efficient profile and we employ high data rate ones to transfer large amounts of data only when truly required.

Reliable Connectivity: Marine monitoring systems are unlikely to have constant or reliable network connectivity, due to the unpredictable and sometimes harsh weather conditions or even because they might be deployed at a long distance from the shore. To this end, we exploit more than one wireless technologies, able to switch from one to another to ensure reliable connection with the observatory system in different conditions. For example, in windy conditions, WiFi signal presents a significant decrease when directional antennas are used. But in the same conditions the performance of cellular protocols is not affected greatly, thus, a sophisticated selection is required. Moreover, we employ delay-tolerant techniques to ensure that the acquired measurements are successfully transmitted to the shore even if they do in an asynchronous fashion.

8.3 Related Work

There are numerous devices that aim at enabling wireless communication between marine sensing devices and the shore. In this work, we discuss some indicative examples since typically, off-the-shelf systems are used in such applications. The Manta Gateway [149] is an heterogeneous platform, that embeds a wide variety of communication technologies, such as 3G, WiFi and underwater acoustics. It is mainly used to provide backbone network and remote access to moving AUVs. Along the same lines, the Meshlium gateway [153], integrates several wireless protocols, such as, 4G, WiFi, ZigBee, LoRa and BLE. It can be used in several Wireless Sensor Networks (WSNs) application scenarios as the backbone communication link. Both devices feature several wireless standards, but despite the fact that are powered over batteries they do not implement any energy-efficient mechanisms. Moreover, [259, 234] propose an acoustic backbone network, which undeniably presents several limitations both in terms of throughput and power consumption, while, [219] illustrates a new radio communication system that achieves long distances, but supports limited data rate.

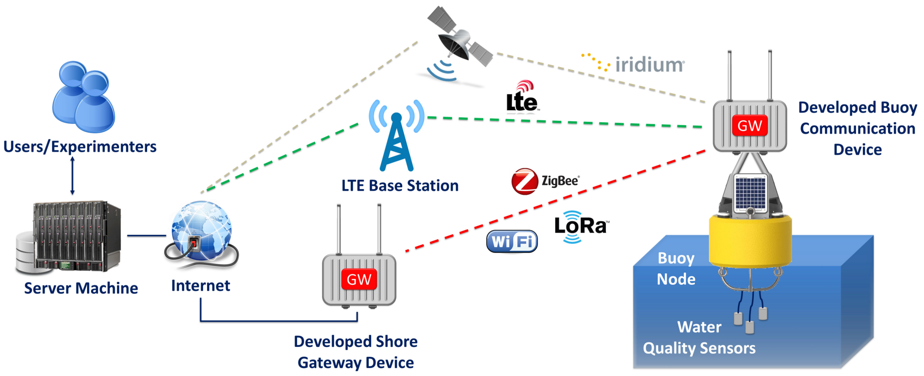


FIGURE 8.1: Proposed Marine Communication System Architecture

8.4 System Implementation & Installation

In this section, we present the architecture of the proposed framework, and moreover, we discuss the development of the communication device. Lastly, we present the installation at the VIDA Buoy.

8.4.1 System Architecture

The overall system architecture is divided in two tiers, the shore-side components and the sea-side ones. The shore components are used to control and manage the sea deployed nodes as well as provide remote access to experimenters. The shore gateway node is a device deployed on the shore and having line-of-sight access with the deployed monitoring system, whilst the sea-side communication device is installed into the Buoy, interfaced with the targeted sensors.

The principle of our system lies in the utilization of several communication protocols and the sophisticated employment of the appropriate one, for each different task. The supported technologies range from IoT-related communication standards, such as LoRa and ZigBee that support low data rates, to the widely adopted WiFi and LTE protocols able to support bandwidth demanding applications. Also, the Iridium interface is utilized to provide remote access to far-located deployments. Notably, IoT-like interfaces present low-power profiles but attain throughput of only a few kbps. However, their speed is adequate to support the needs of such deployments, since environmental sensors operate at low sampling frequencies, generating only a few Bytes per sample. On the contrary, WiFi and LTE are used to enable remote access and large file transfers. Fig. 8.1 illustrates the system architecture.

8.4.2 System Implementation

The communication device is composed of different hardware and electronic components, constituting a unified communication system. Fig. 8.2 illustrates the developed device.

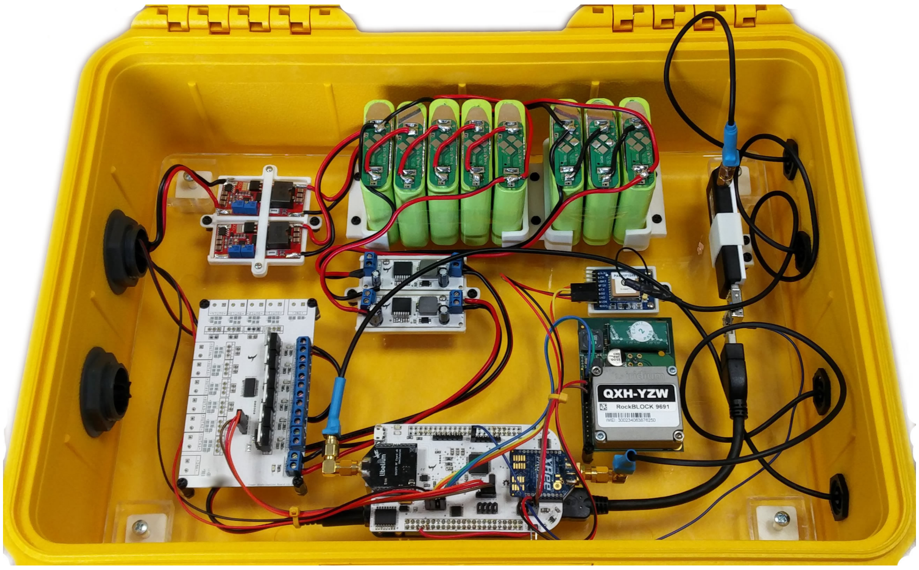


FIGURE 8.2: Developed Communication Device

Embedded Device: The core module is again the BeagleBone Black Rev. C [28], which is a low-cost, embedded platform characterized by sufficient processing capabilities (1GHz with 512MB RAM), low-power consumption and several communication interfaces. All of the electronic components used for are hosted on a custom-made printed circuit board (PCB) daughterboard mounted on top of the embedded host device. The embedded platform is responsible for the communication with the on-board environmental sensors and implements the software architecture for the communication system. Furthermore, the device locally caches the acquired measurements, prior to the offloading process.

Low-power microprocessor: The developed communication device is also equipped with a low-power ARM cortex-M4 microprocessor, the MK20DX128 [159], responsible for the power management of all the on-board components. In brief, the MK20DX128 remains always active and communicates directly with the IoT-like interface (LoRa or XBee) expecting incoming requests, in order to activate the required components. When it is requested to perform a sensing cycle, it wakes the BeagleBone and then acts as a bridge between the IoT interface and the BeagleBone, by forwarding all the incoming data. Of course, it can also activate the WiFi, the LTE or the Iridium interfaces when requested. After the completion of the sensing cycle, it instructs the BeagleBone and any other active component to enter sleep state. Moreover, the microprocessor periodically senses the battery voltage and reports the acquired value via the IoT interface. Notably, the MK20DX128 is configured to operate at 24MHz, draining roughly 12 mA, which is a reasonable consumption for such systems. Finally, the MK20DX128 can be set in a duty-cycle fashion, draining even less. In essence, it can be configured to enter sleep state, consuming just a few μA and use it's internal timers to generate wake-up interrupts, in order to enter it's active state and check for any pending request(s).

XBee 868LP: The XBee S8 [258] is an IoT interface that can reach communication distances of up to 8.4 km. It operates in the 868 MHz ISM band and achieves up to 80 kbps transmission rate. It supports the establishment of ad-hoc networks towards creating a single network of several monitoring systems in a mesh topology. The communication with the MK20DX128 is realized through the UART protocol.

LoRa: LoRa [8] is a recent ultra long-range technology that can reach distances of up to 21 km, by exploiting spread spectrum modulation. It attains very low data rates from 0.3 kbps to 50 kbps, while consuming only a few mA when propagating and receiving frames. Notably, LoRa does not support ad-hoc topologies but only star networks. In our implementation we used the Semtech SX1272 chipset which communicates with the MK20DX128 via SPI.

WiFi: WiFi connectivity is enabled through the commercial Atheros 9271 chipset offering a wide range of supported channel bandwidths from 40 MHz down to 5 MHz, providing a compromise between achievable data rates and transmission range. It communicates with the BeagleBone through a USB port and supports the connection of an external antenna to achieve higher signal. It offers high speed features allowing transfers of large files and remote access.

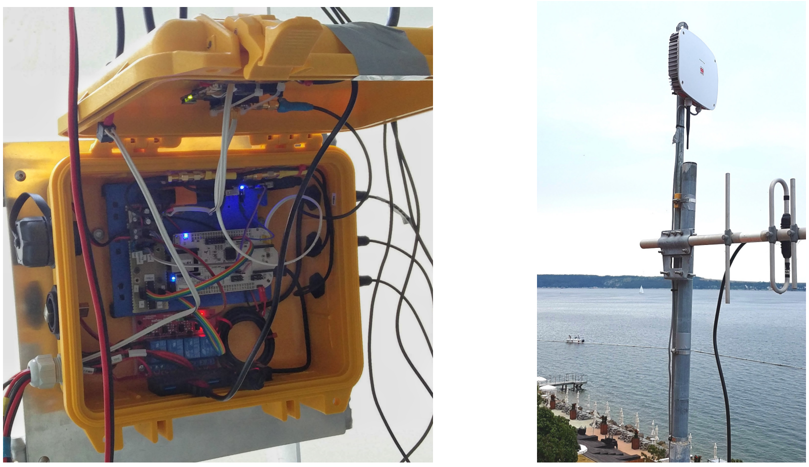
LTE: LTE provides an alternative high bandwidth option, even in far locations where WiFi connectivity can no longer be maintained. We used the Huawei E392 USB Dongle that supports 2x2 MIMO. It features ports for connecting external antennas and reaches up to 100 Mbps download and 50 Mbps upload speeds.

Iridium: We also equipped our device with the Iridium RockBLOCK Mk2 interface that is only used in extreme conditions when no other backbone networks are available.

Finally, communication with the environmental sensors is realized via a serial-to-USB module, connected to the BeagleBone and a GPS interface is used to provide measurement time-stamping and the geo-location of the system.

8.4.3 Power Switches

Some of the aforementioned modules feature a sleep state which they can enter in order to save energy, while some others do not. For instance, WiFi, LTE, Iridium and the on-board environmental sensors do not integrate such feature. To this end, we implemented custom circuits to entirely cut their power off when required. More specifically, for the USB dongles, we developed a USB extension board that features a power switch IC that intercepts the voltage rail. This IC is controlled by the MK20DX128 for turning the targeted modules on and off. Along the same lines, we employed power MOSFETs, used in a topology that forms a power switch, in order to control the operational state of the on-board sensors and peripherals. Notably, the quiescent current consumption of the power switch and mosfet ICs is negligible and does not affect the overall power expenditure of our system.



(A) Installed Communication Device

(B) Shore Device

FIGURE 8.3: Installation Communication Device & Shore Gateway

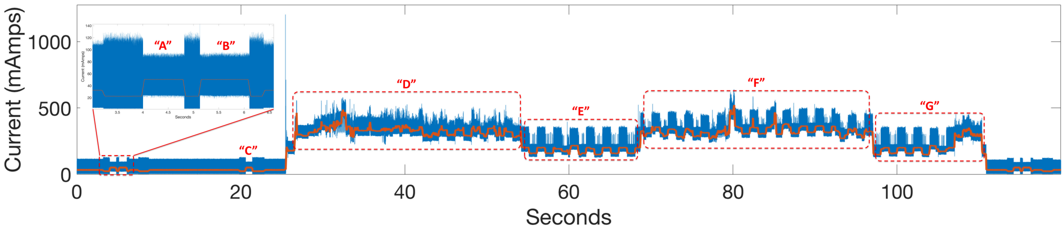


FIGURE 8.4: Power Consumption Profiling of the Developed Buoy Communication Device

8.4.4 Installation

A stripped-down version of the developed communication system was successfully installed in the Vida oceanographic buoy [250], in Piran, Slovenia, which is operated by National Institution of Biology (NIB). Vida is equipped with various physical and chemical sensors constantly measuring relevant environmental parameters, serving as a proper laboratory in the ocean. The installed communication device features a LoRa interface as the low-rate link and an LTE one for high-throughput demanding applications. Moreover, the device interfaces with the WIZ probe [29], which is an in-situ nutrient probe. Also, it features several power mosfets to control the power of the WIZ probe and it's filtration system (pumps). On the shore-side we have deployed a gateway communication device that acts as the LoRa backbone network. Notably, the shore-side gateway is located 3.7 km away from the Vida buoy. Fig. 8.3a and 8.3b illustrate the communication device deployed into the Buoy, and the gateway one, deployed at the NIB premises. The environmental measurements acquired by our system can be accessed at our web-based GUI [66].

8.5 Evaluation

To characterize the power consumption profile of the developed communication system, we performed in-lab investigation, taking advantage of our high-end power

Component	Sleep Draw	Idle Draw	Active Draw
BeagleBone	-	160 mA	328 mA
MK20DX128	150 μA	-	14 mA (at 24 MHz)
XBee 868LP	1.7 μA	22 mA	45 mA (TX_P), 33 mA (RX_P)
LoRa	-	1.65 mA	41 mA (TX_P), 11 mA (RX_P)
WiFi	-	51 mA	101 mA (upload)
LTE	-	120 mA	196 mA (upload)
Iridium	20 μA	-	102 mA

TABLE 8.1: Power consumption measurements of Discrete Components

consumption monitoring tool, developed in [**acm**]. First, we measure the instantaneous power consumption per different component in the supported states, as illustrated in Table 8.1. We can observe a huge difference between the high throughput interfaces and the low-power ones, both in idle and active states, which further highlights the urgency to turn them off when not in use. In Fig. 8.4 we plot the total power consumption draw of the installed device. At first, only the LoRa link is active, while at roughly the 26^{th} second the BeagleBone is instructed to also enter an active state. Notably, this is an artificial experiment aiming to illustrate the different states and power consumption levels of our device. Also, we plot, with red line, the average power consumption which gives us more clear observations.

We have marked the acknowledgement packet transmitted by the LoRa with "A", after the reception of a battery voltage request, which is then followed by the data transmission, marked with "B". Roughly at the 21^{st} second LoRa receives a message instrucing the activation of the BeagleBone, marked with "C", which is realized after a short artificial delay. Frame "D" illustrates the drain of the BeagleBone in boot phase, while frame "E" shows it's idle consumption. In turn, the device is instructed to turn on the LTE interface highlighted with "F", while in "G" the BeagleBone is idle and then powers off completely.

Moreover, we calculated the energy consumption of our system required to perform one full analysis cycle. We estimated that for one cycle, our device must be in active state for roughly 76 seconds. In essence, it takes 38 seconds to initiate the WIZ probe and control the related pump systems and 38 more seconds after the completion of the cycle to acquire the measurements. This implies that the required energy per measurement cycle is roughly 101.4 J, which is quite a reasonable energy expenditure for such systems. Notably, our device only drains 22 mA when in idle state, in which the BeagleBone is in sleep state and only the MK20DX128 and LoRa are activated. Finally, it is worth noting that the LoRa and the LTE links, presented stable network performance during the entire testing phase, not affected by the weather conditions.

8.6 Conclusions

In this work, we introduced a novel communication architecture tailored to marine monitoring deployments. It features ultra low power consumption profile, while also providing high throughput capabilities. Our work highlights the urgency to turn off the high power interfaces and only maintain a low power one to act as the primary communication link. After thorough experiments, we measured that the developed device only requires 102 J to perform one measurement cycle, while it consumes roughly 22 mA in idle state.

Chapter 9

Conclusions

9.1 Conclusions and Future Work

In this thesis, we presented new methods and principles for prologning the life expectancy of modern sensor networks as well as for evaluating the power draw of IoT devices. In Chapter 2 we employed an external *RTC* module to control the *go-to-sleep* and wake-up functions of any IoT mote. The suggested principle is set-up with low-cost *off-the-shelf* components, while it remains minimally invasive to the host node. By adopting our strategy the power expenditure of an *IoT* mote in sleep state can drop as low as *22 nA*, which is at worst a reduction of *98 %*, compared to the most power efficient platforms available. Remarkable lifetime extensions can be achieved in low *duty-cycled* scenarios, while in higher *duty-cycles* the benefit is minimal. Specifically, we noted a lifetime extension for a state-of-the-art sensor node to *20 years* on a *0.025 % duty-cycle* scenario attaining an incredible increase of *538 %*.

In Chapter 3 we showcase a new scheme for semi-passive *WuR* circuits that remarkably increases the sensitivity of the existing state-of-the-art implementations, reaching beyond *-70 dBm*. Our finding lies in the employment of a nano-power amplifier that intensifies the signal prior to the decoding process. The overall consumption of our proposed circuit is roughly *580 nA*, while it can be substantially reduced by using alternative *ICs*.

Subsequently, in Chapter 4 we present the eProfiler, a high-fidelity power consumption meter with long-term monitoring capabilities. The meter supports wide current ranges spanning from 100 nA to 100 mA, and features an average error of only of 0.45 % throughout it's measuring range. It is the first meter to support monitoring of nA currents which are becoming increasingly popular in IoT ecosystems. All the aforementioned features aid in the power profile characterization of IoT devices. The eProfiler provides an in-situ, economically reasonable, and precise measuring framework. Leveraging eProfiler the consumption profile assessment of sensor nodes, and the efficiency of energy-harvesters can be effectively evaluated, enabling realistic estimations for the lifetime of deployed IoT sensor networks.

Following this, Chapter 5 illustrates a city-scale monitoring LoRa-based infrastructure that provides real-time air and link quality indications. Furthermore, we characterize the energy efficiency of the utilized LoRa chipset by conducting in-depth lab experiments. We conclude that the LoRa standard is a very attractive option for deploying energy and cost efficient *IoT* applications in city-scale environments. Specifically, our findings indicate that rate adaptation algorithms and packet aggregation techniques can drastically benefit the energy efficiency of the LoRa protocol, when considering battery powered applications. Moreover, our city-scale experiments suggest that LoRa can support demanding applications achieving impressively high PDR at extremely low RSSI conditions, even when using high data rates.

In Chapter 6 we introduce the NITOS BikesNet platform, which can be used to realize a city-scale mobile sensor testbed, based on nodes that are mounted on bicycles. The experimenter can control these nodes via the OMF/OML framework, in a convenient way. Such a testbed could stimulate research and service provisioning in the areas of mobile ad-hoc sensing, participatory and crowdsensing applications. We also believe that our work can provide valuable insights to researchers working in similar testbed platforms.

Moreover, in Chapter 7 we present the EVERUN framework that enables researchers to characterize the performance of proposed UWSN protocols from an energy efficient point of view. The developed system was integrated with real LAUV and GW components of the SUNRISE testbed-site in Porto and the SUNSET experimentation framework. Through the execution of a wide set of remote experiments, under realistic conditions, we characterized the power behavior of key UWSN hardware and collected valuable insights about the energy efficiency of important protocol parameters and mechanisms. Inefficiencies of simulation-based energy evaluation tools were highlighted, while the delivered modeling accuracy was improved through the incorporation of accurate power data collected through EVERUN.

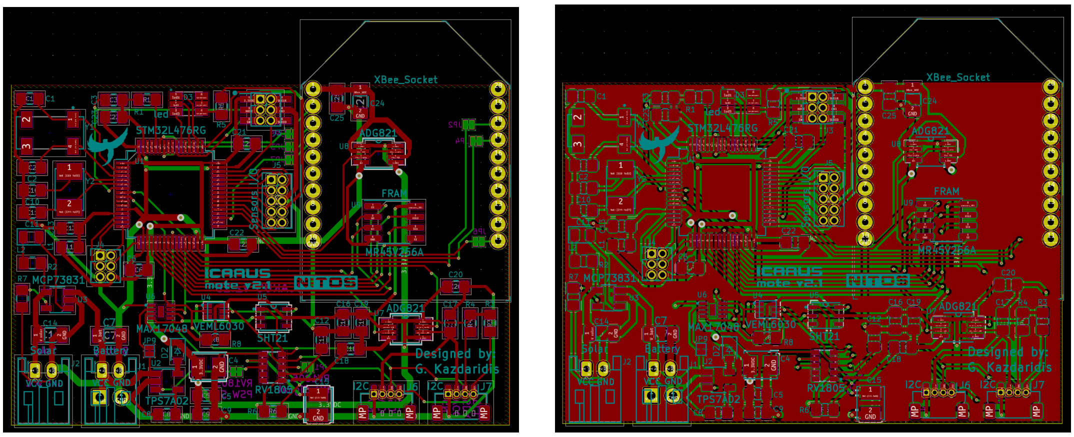
Finally, in Chapter 8, we introduce a novel communication architecture tailored to marine monitoring deployments. It features ultra low power consumption profile, while also providing high throughput capabilities. Our work highlights the urgency to turn off the high power interfaces while only maintain a low power one to act as the primary communication link, with the LoRa technology being an attractive solution.

Appendix A

Additional Information on the Prototypes

In this chapter we provide additional information on the prototypes presented in the main thesis demonstrating the electronic circuits designed and in some cases images from the deployments.

A.1 ICARUS Prototype



(A) ICARUS Prototype PCB Design (B) ICARUS Prototype PCB Design w/ Planes

FIGURE A.1: ICARUS Prototype Design

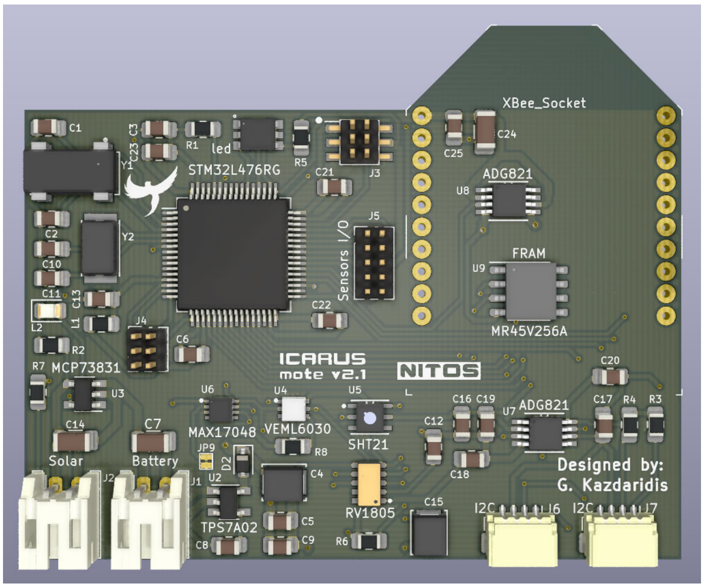


FIGURE A.2: ICARUS Prototype Photorealistic Model

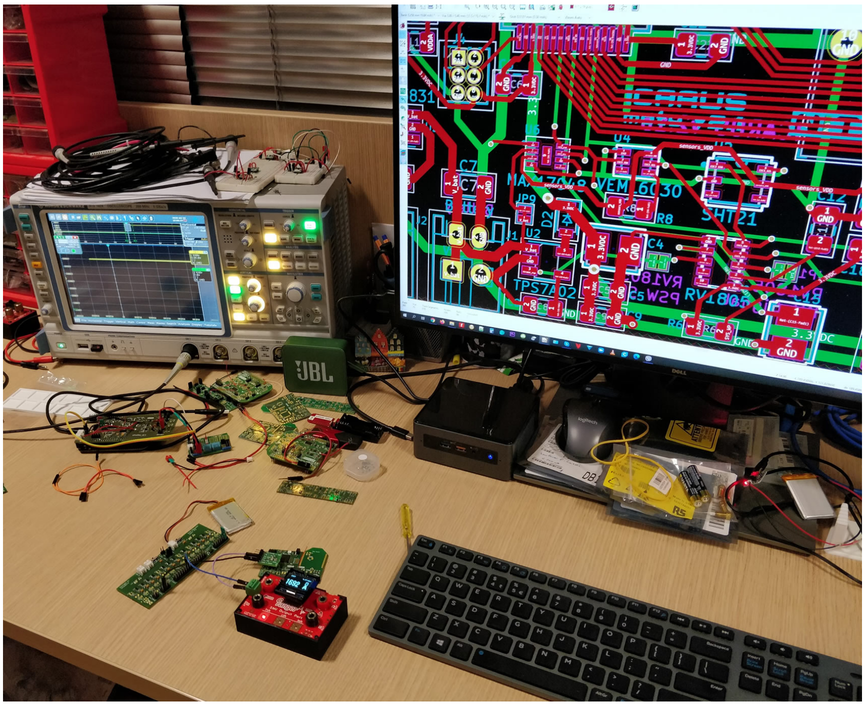


FIGURE A.3: ICARUS Prototype - Measuring RV1805's Interrupt Signal w/ an Oscilloscope

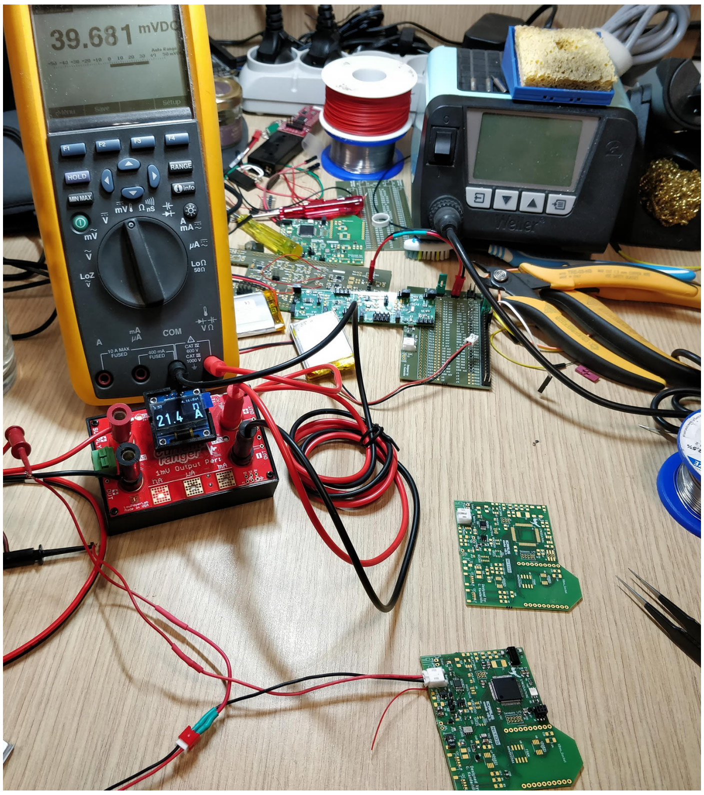
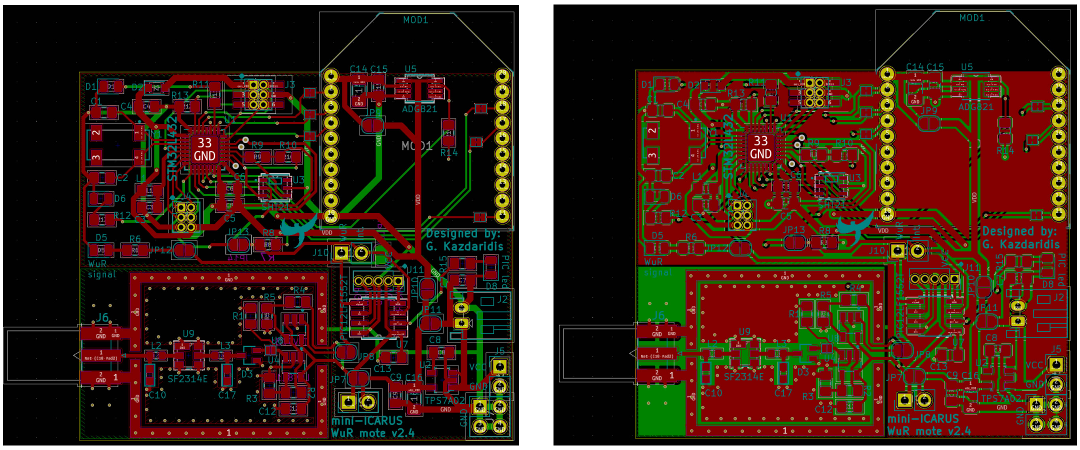


FIGURE A.4: ICARUS Prototype - Power Cons. Monitoring - ICARUS draws only a few nA in Sleep State

A.2 eWake Prototype



(A) eWake Prototype PCB Design

(B) eWake Prototype Design w/ Planes

FIGURE A.5: eWake Prototype Design

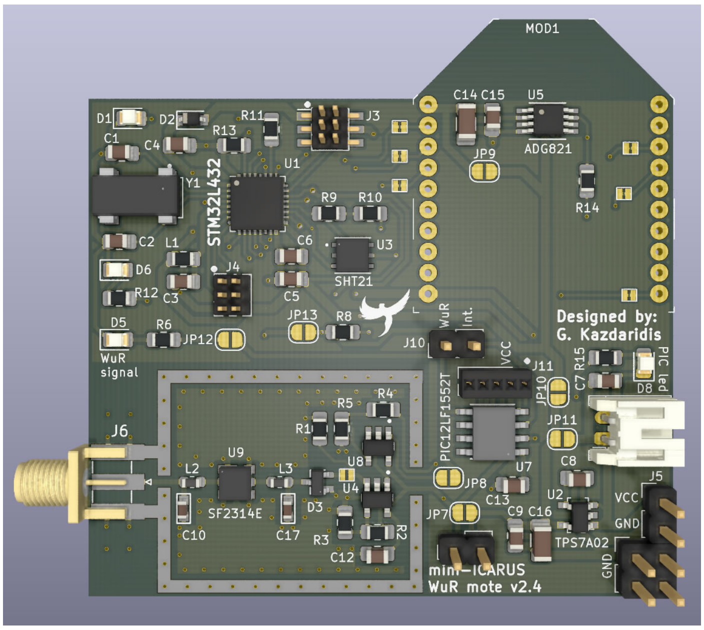


FIGURE A.6: eWake Prototype Photorealistic Model

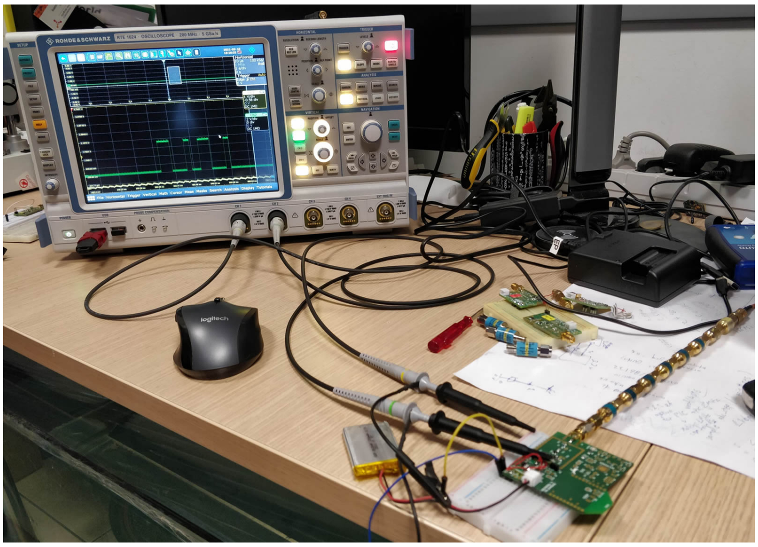
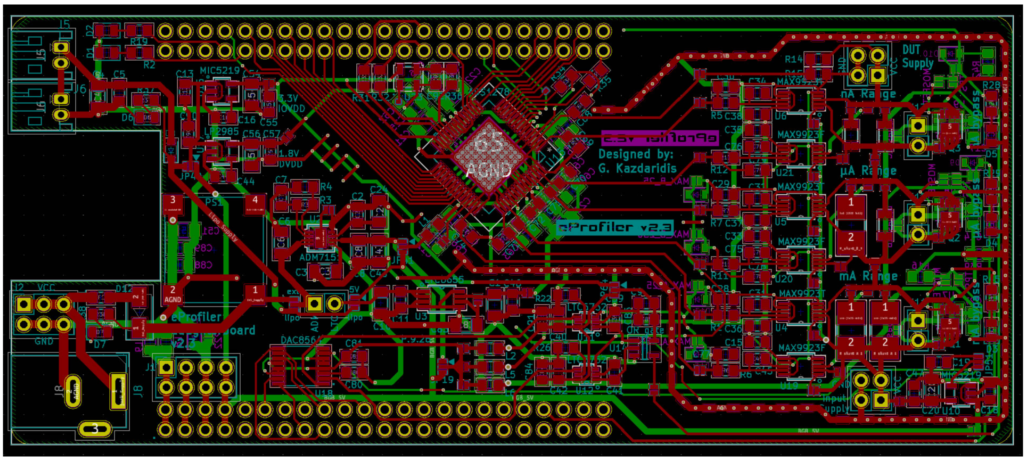
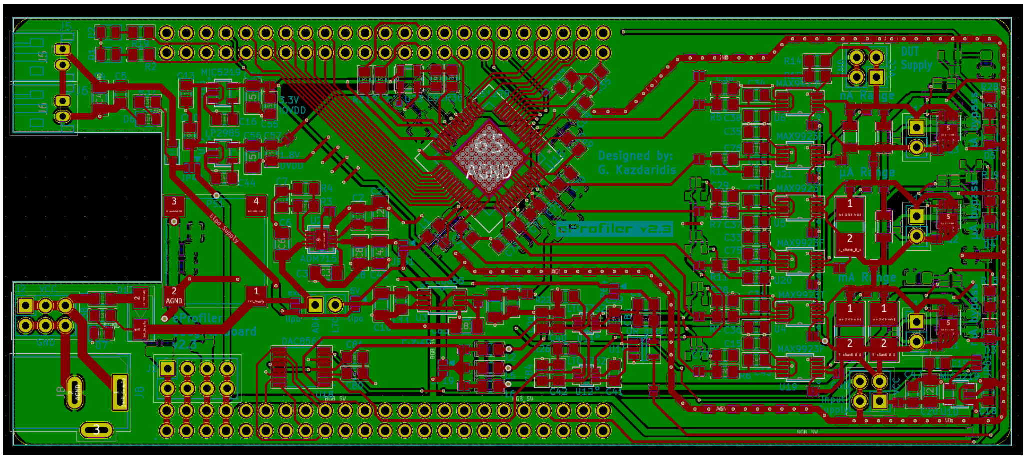


FIGURE A.7: eWake Prototype - Correctly Receiving an Attenuated OOK Signal at -70 dBm

A.3 eProfiler Prototype

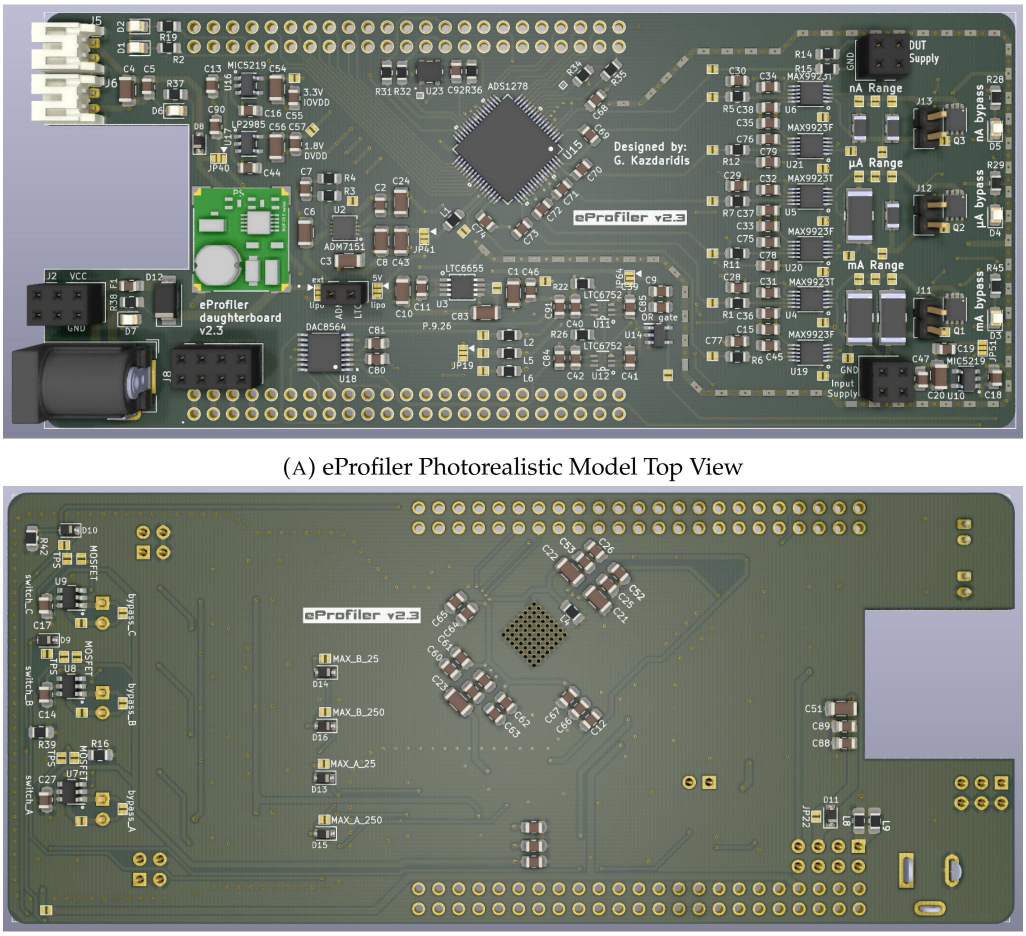


(A) eProfiler Prototype PCB Design



(B) eProfiler Prototype Design w/ Planes

FIGURE A.8: eProfiler Prototype



(A) eProfiler Photorealistic Model Top View

(B) eProfiler Photorealistic Model Bottom View

FIGURE A.9: eProfiler v2 Photorealistic Model

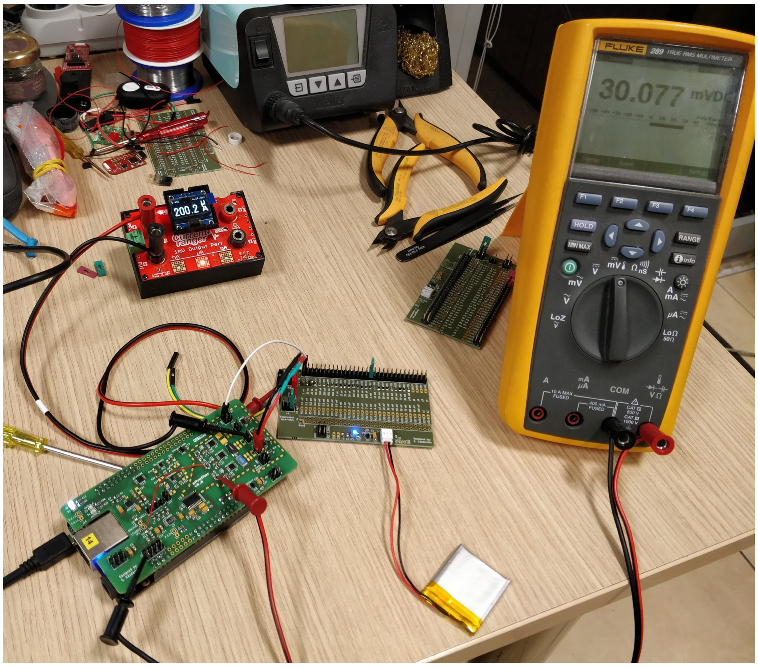


FIGURE A.10: eProfiler - Experimenting w/ eProfiler v1

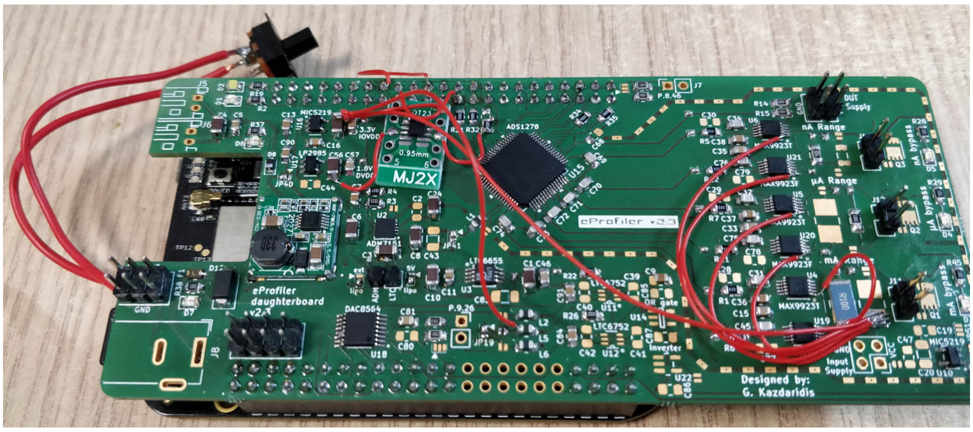
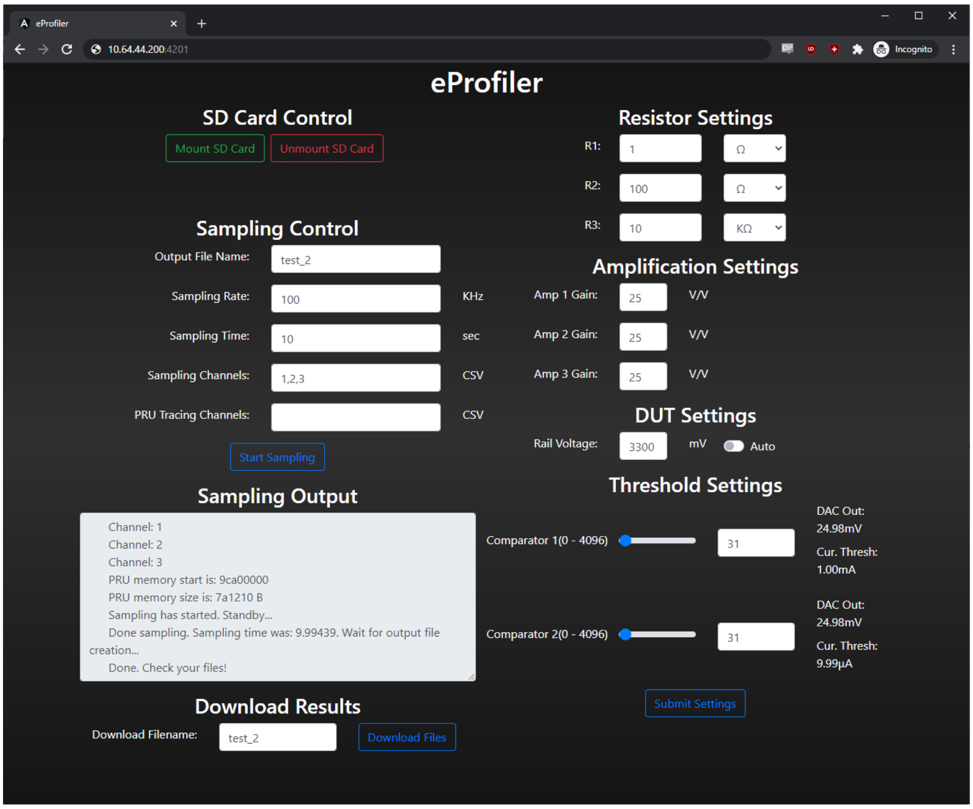


FIGURE A.11: eProfiler v2 During Testing/Development Phase



eProfiler

SD Card Control

Mount SD Card | Unmount SD Card

Sampling Control

Output File Name: test_2

Sampling Rate: 100 KHz

Sampling Time: 10 sec

Sampling Channels: 1,2,3 CSV

PRU Tracing Channels: CSV

Start Sampling

Sampling Output

```
    Channel: 1
    Channel: 2
    Channel: 3
    PRU memory start is: 9ca00000
    PRU memory size is: 7a1210 B
    Sampling has started. Standby...
    Done sampling. Sampling time was: 9.99439. Wait for output file
creation...
    Done. Check your files!
```

Download Results

Download Filename: test_2 Download Files

Resistor Settings

R1: 1 Ω

R2: 100 Ω

R3: 10 KΩ

Amplification Settings

Amp 1 Gain: 25 V/V

Amp 2 Gain: 25 V/V

Amp 3 Gain: 25 V/V

DUT Settings

Rail Voltage: 3300 mV Auto

Threshold Settings

Comparator 1(0 - 4096) 31 DAC Out: 24.98mV Cur. Thresh: 1.00mA

Comparator 2(0 - 4096) 31 DAC Out: 24.98mV Cur. Thresh: 9.99µA

Submit Settings

FIGURE A.12: eProfiler - Developed GUI for Interacting w/ eProfiler v1

A.4 Bikesnet Prototype

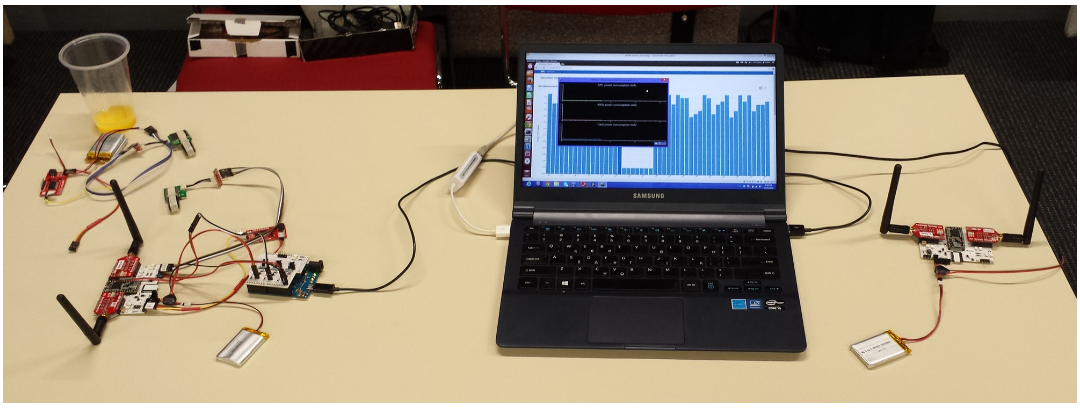


FIGURE A.13: Bikesnet Prototype - Demonstrating a Setup of Two Bikesnet Nodes at IEEE MDM 2014 in Brisbane, Australia

A.5 EVERUN

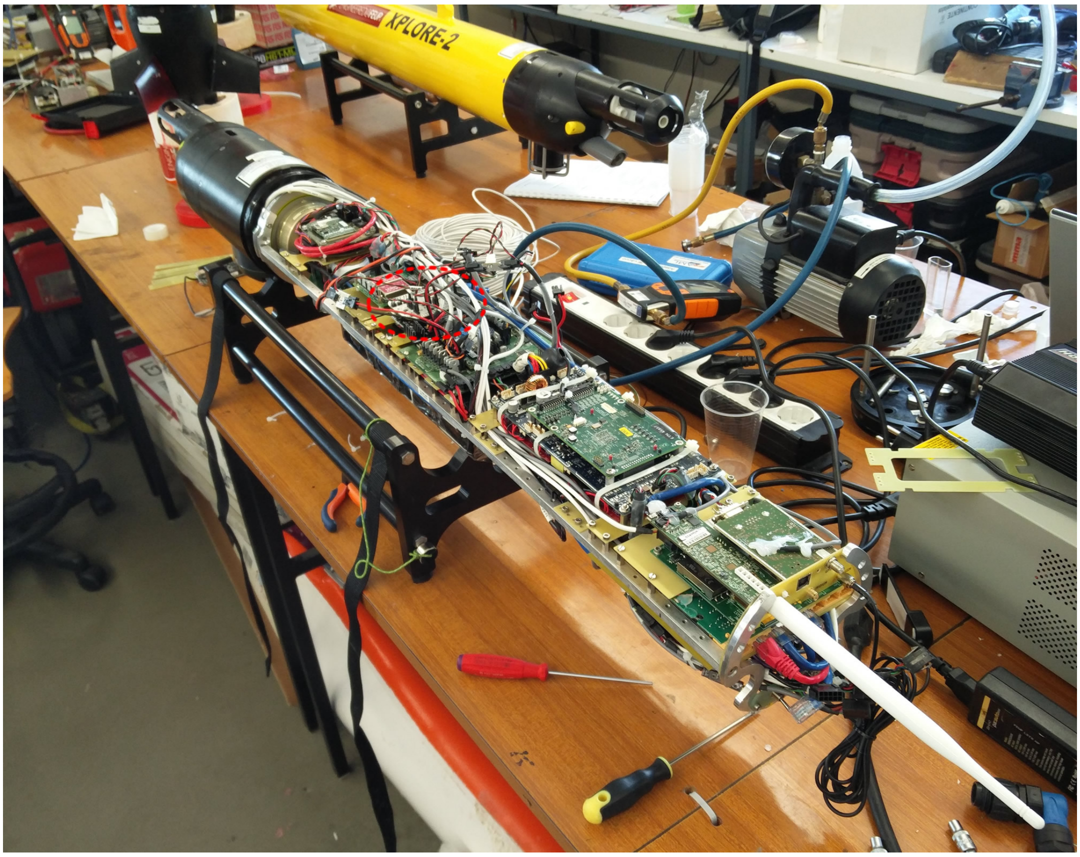


FIGURE A.14: EVERUN - Installed Power Consuption Meter Into Univ. of Porto (LSTS lab.) Underwater Vehicle (Noptulis-1)

FIGURE A.15: EVERUN - Univ. of Porto - LSTS Laboratory

FIGURE A.16: EVERUN - Working at Univ. of Porto (LSTS Labotatory) During our Visit

A.6 Development of Marine Communications

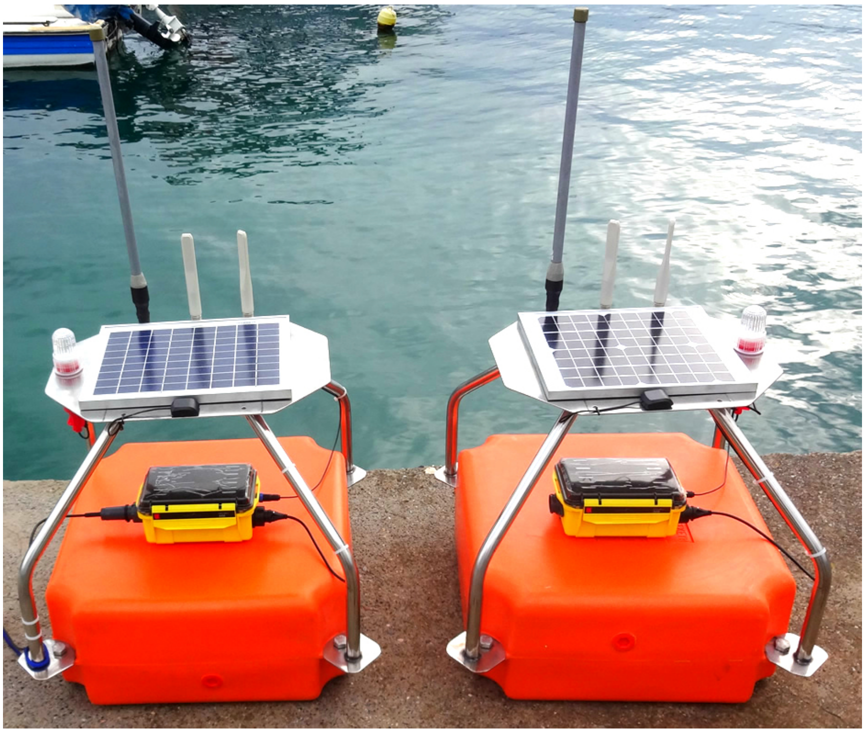


FIGURE A.17: Developed Buoys

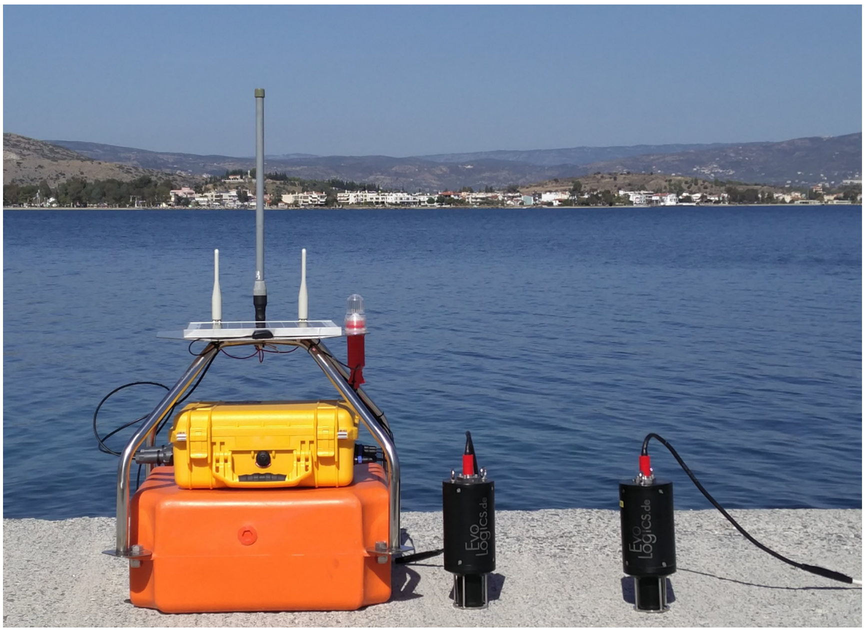


FIGURE A.18: Developed Buoy Alongside a pair of Acoustic Modems by Evologics

Appendix B

Other Projects

In this chapter I provide some additional information/images on projects I had the chance to work during my studies (which are not included in the main thesis).

B.1 NITOS Wireless Experimentation Testbed

During my studies I was a key member for the development of the NITOS wireless testbed, located in Univ. of Thessaly premises, in Volos, Greece.

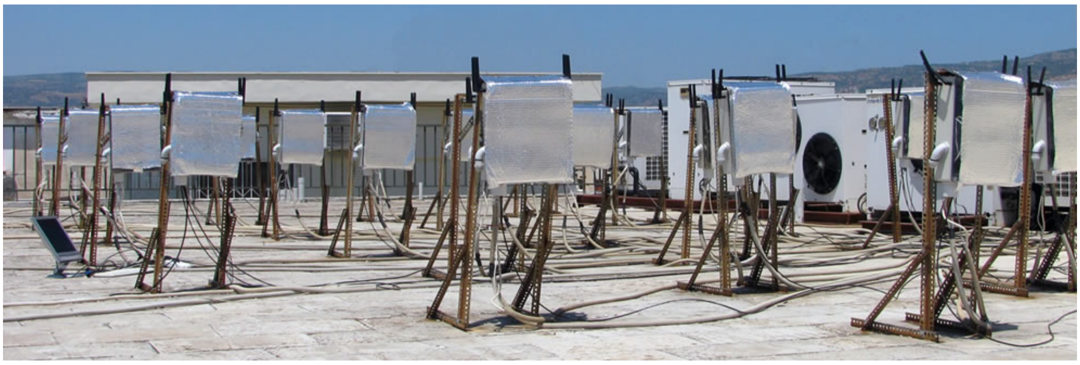


FIGURE B.1: NITOS Outdoor Deployment

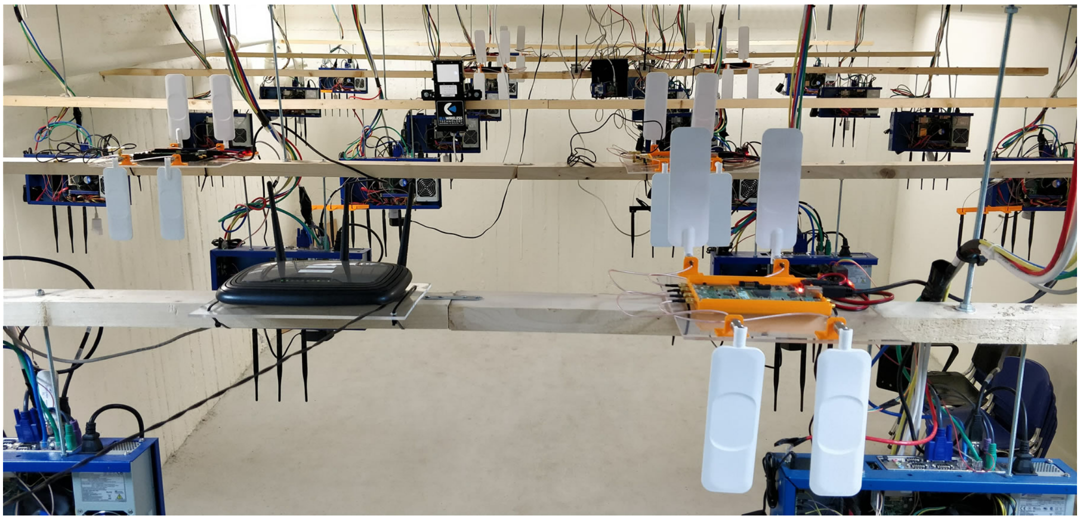


FIGURE B.2: NITOS Indoor Deployment

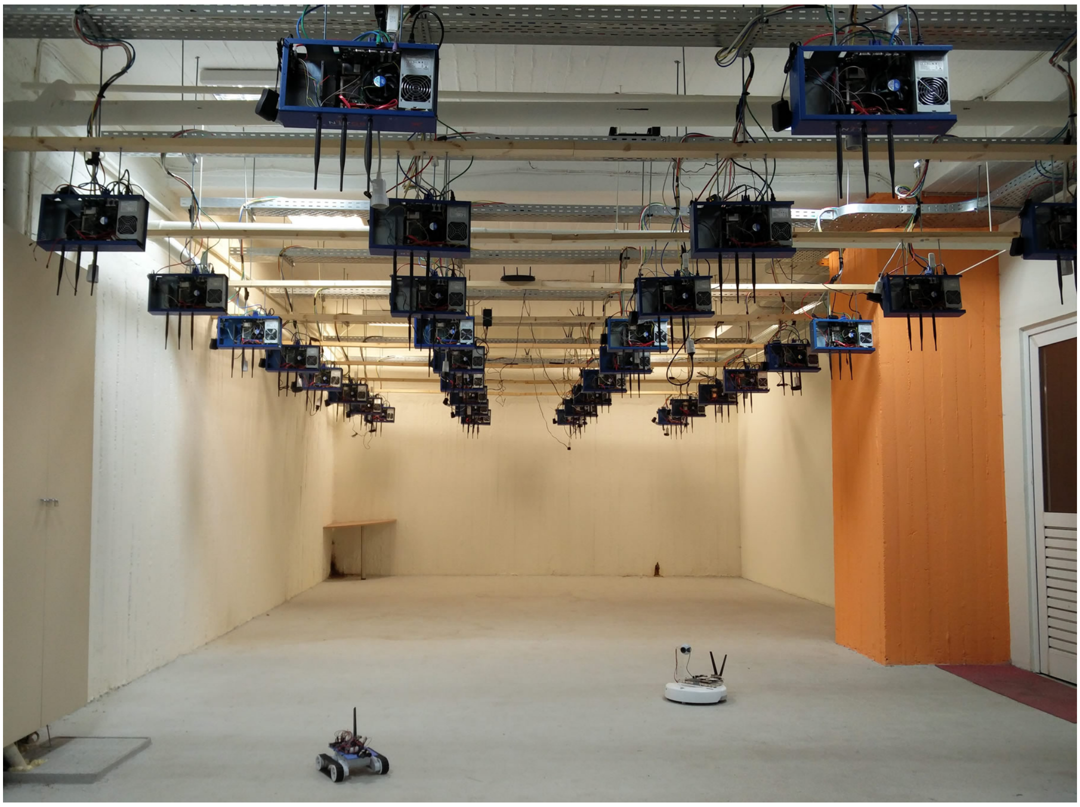


FIGURE B.3: NITOS Indoor Deployment

B.2 Development of Wireless Sensors and Gateways

This is the first version of the ICARUS mote that integrates the ATMega1284p MCU and several sensor modules.

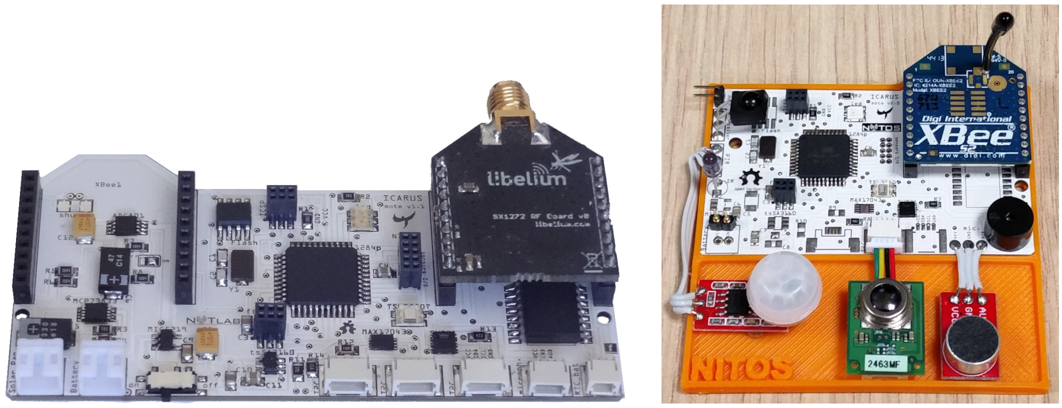


FIGURE B.4: The first version of the ICARUS mote

The following device is an IoT Gateway implemented using a BeagleBone board and a custom PCB on top that features the MK20DX128 ARM-M4 MCU as well as two XBee-like IOT Interfaces, a LoRa and an XBee Pro.

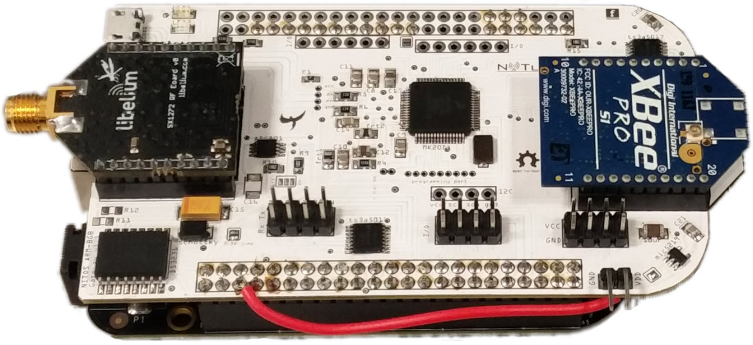


FIGURE B.5: Gateway device LoRa / XBee

Next, we demonstrate the same Gateway device equipped with the CC2500 RF Chipset and a Wireless M-Bus Radio Module.

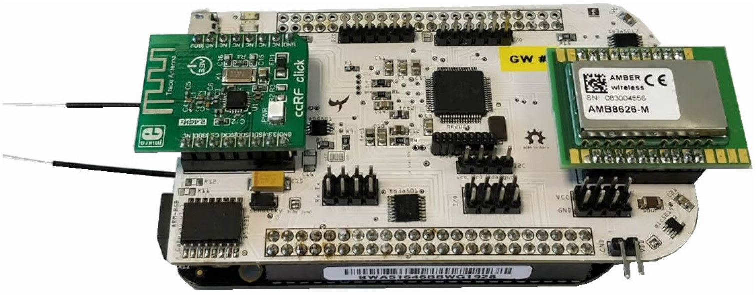


FIGURE B.6: Gateway device CC2500 / M-Bus

Subsequently, we illustrate the LoRa-WAN Gateway Device that is developed using a Raspberry Pi board and the IMST iC880a LoRa-WAN module that acts as concentrator.

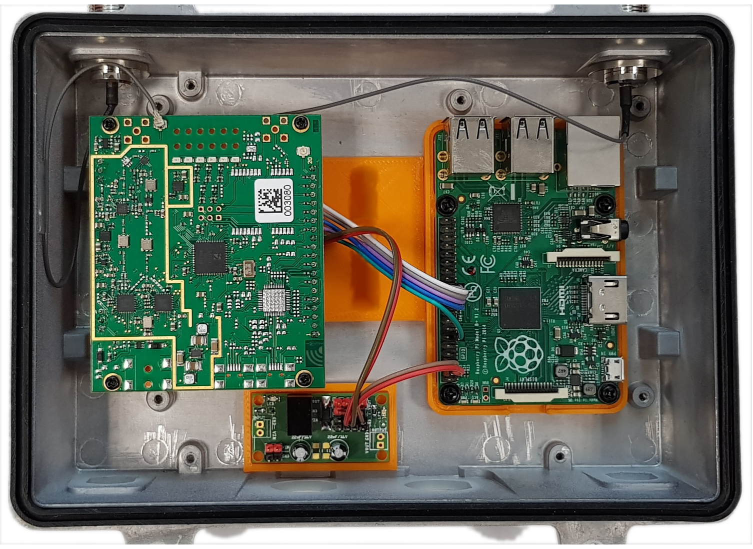


FIGURE B.7: Development of a LoRa-WAN Gateway Device based on the IMST iC880a board

In this image we demonstrate an ICARUS mote that is remotely configurable (also supporting remote firmware uploading) through a BeagleBone PC being part of a City-Scale Wireless Testbed.

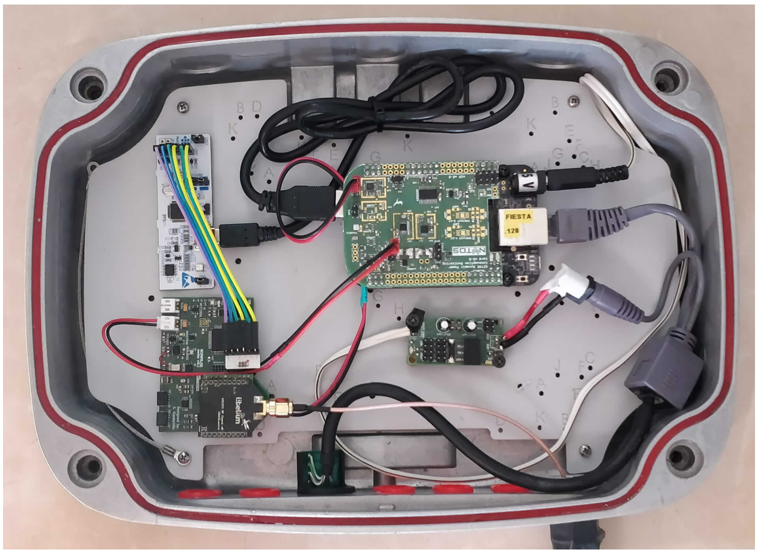


FIGURE B.8: City-Scale Wireless Testbed

B.3 Roboticss

The following robot employs a set of aluminum profiles as a rail, moving explicitly on this mechanism, while featuring a long-distance LiRAR sensor to estimate the distance from a fixed plane placed in the end of the rail in order to estimate its position real-time. Moreover, the robot features a precise short-distance ultrasound sensor to allow the vehicle to park exactly on top of wireless coil so as to be able to charge wirelessly when not in use.

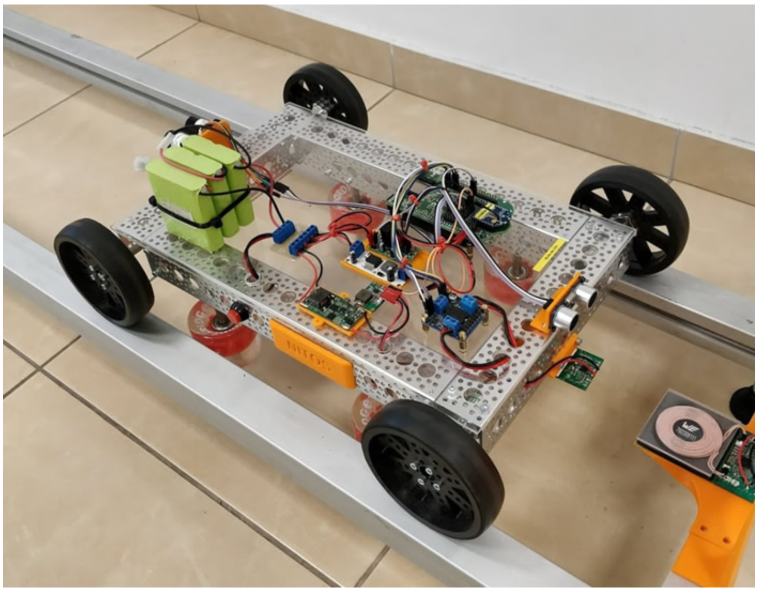
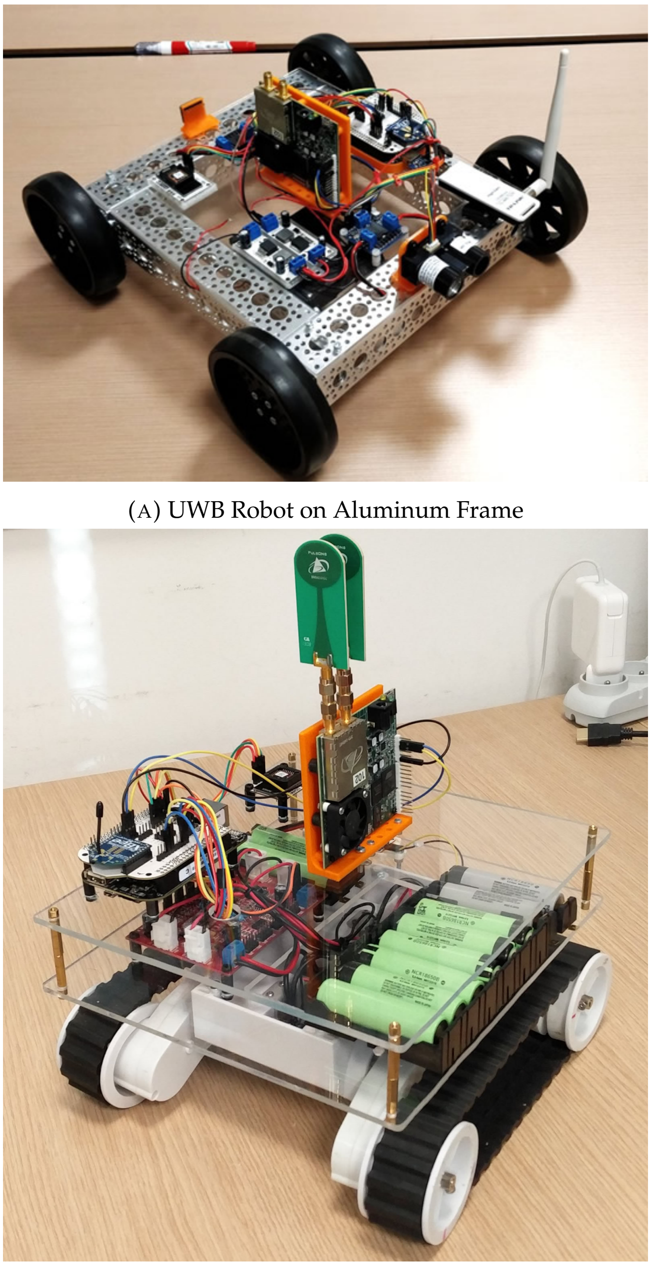


FIGURE B.9: Rail Robot

The following robot can move freely on every direction, while it employs an Ultra-Wideband (UWB) positioning system so as to localize itself as long as it is moving which is realized with the aid of 3 or more fixed UWB nodes playing the role of anchors. During our experiments the robot was following a predefined path, while robot's position accuracy was roughly 2 cm. Notably, the robot is also using an IMU unit that provides compass measurements.



(A) UWB Robot on Aluminum Frame

(B) UWB Robot on Caterpillars

FIGURE B.10: Ultra-WideBand Robot

The next robot is a similar version but for its positioning it uses a 360 degrees LiDAR system which is used to capture the space around the robot providing all the information in a ROS environment to create space's map with the aid of SLAM algorithms running on ArduPilot unit (Pixhawk).

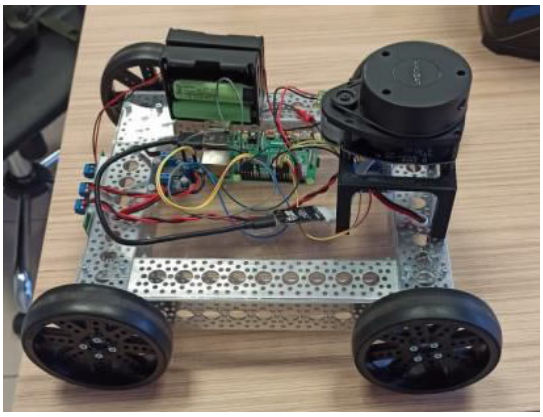


FIGURE B.11: LiDAR Robot

The last ground vehicle employs two cameras for its navigation, the 360 degrees Ricoh Theta and the Intel RealSense depth camera.

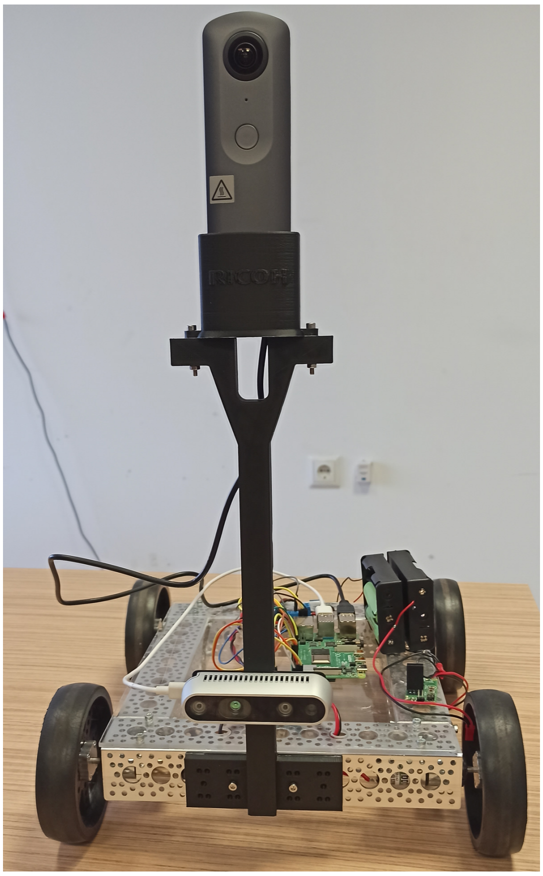


FIGURE B.12: Camera assisted Robot

Lastly, the DJI S1000 multicopter was used for the evaluation of various IoT protocols under aerial mobility. The drone is equipped with a BeagleBone PC that features various IoT interfaces used to exchange wireless frames with stationary devices in order to assess the PDR and other wireless oriented parameters of the established communication.

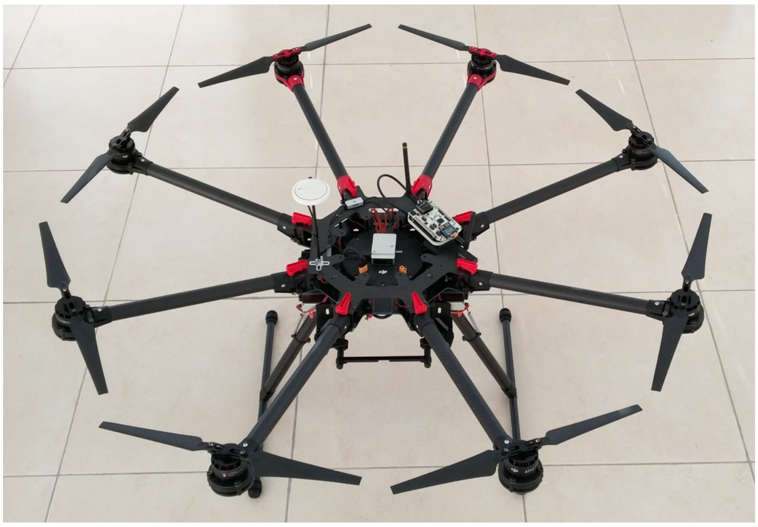
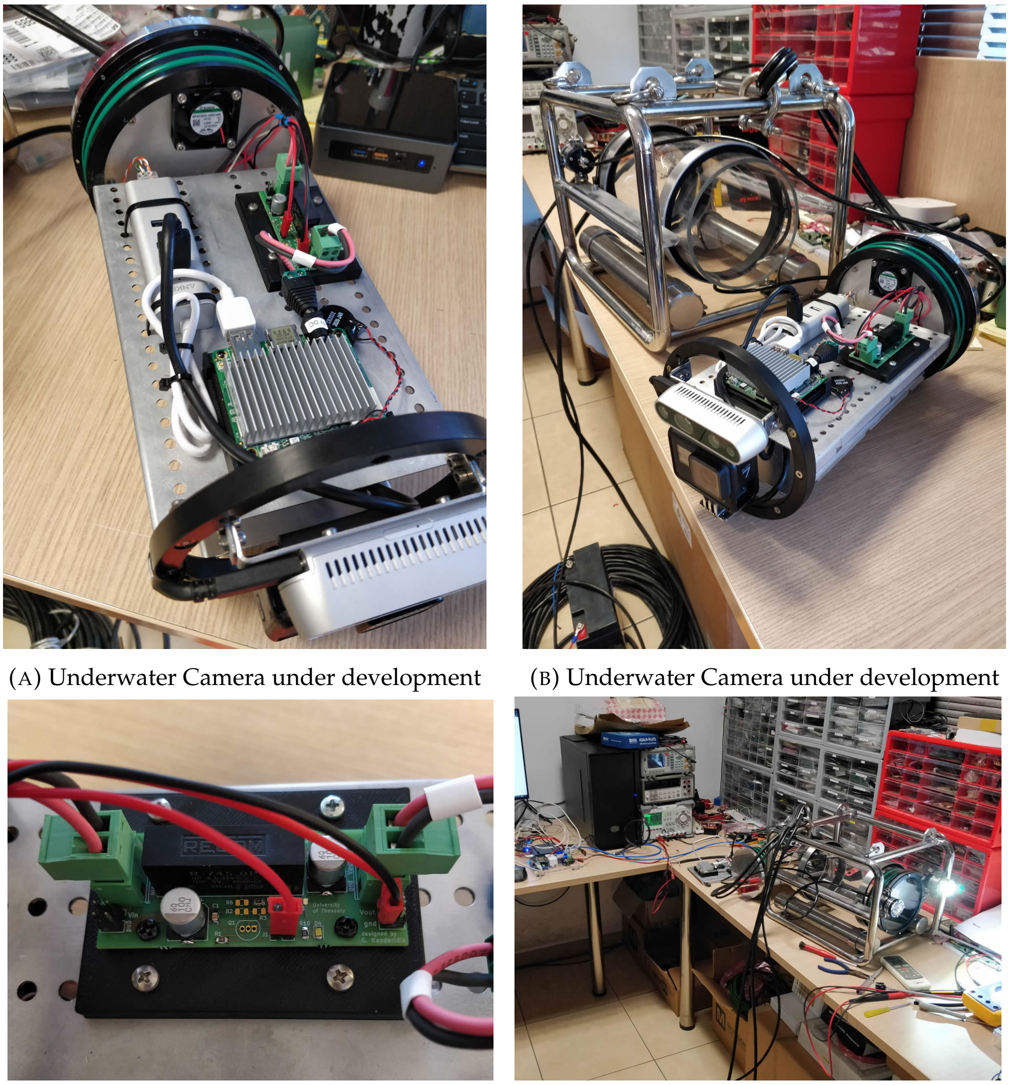


FIGURE B.13: DJI S1000 aircraft equipped with an IoT platform for the evaluation of various IoT protocols

B.4 Underwater Camera for Fish Biomass Estimation

In this project we aim at characterizing the biomass of aquaculture farns by capturing the size of individual fishes inside the cages employing a machine learning mechanism to automatically detect fishes and measure their size using the Intel RealSense depth camera.

In the following photos we demonstrate the developed camera system, deployed at Kefish aquaculture farm in Kefalonia in June 2021.



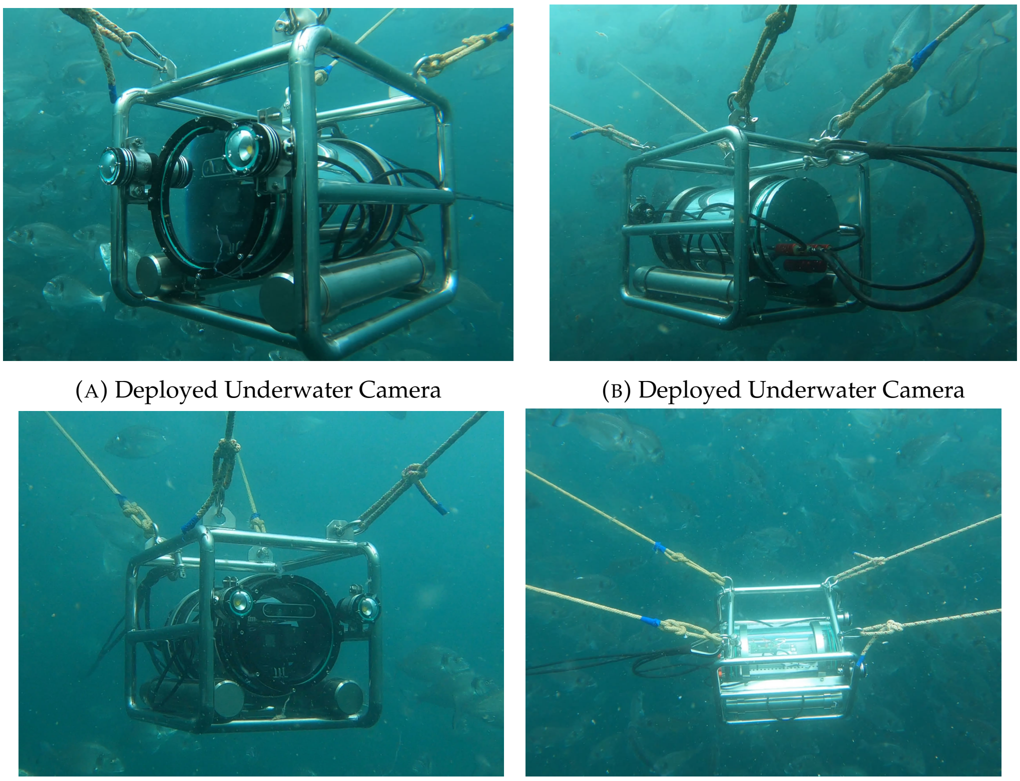
(A) Underwater Camera under development

(B) Underwater Camera under development

(C) Power Regulation System

(D) Underwater Camera under testing

FIGURE B.14: Underwater Camera under development phase



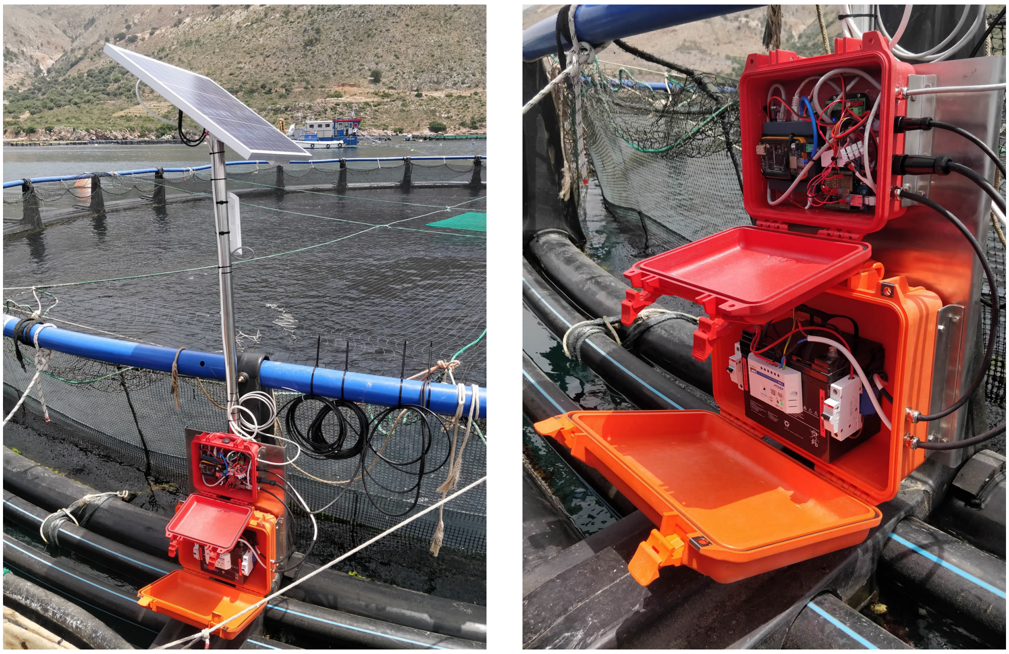
(A) Deployed Underwater Camera

(B) Deployed Underwater Camera

(C) Deployed Underwater Camera

(D) Deployed Underwater Camera

FIGURE B.15: Deployed Underwater Camera



(A) Overwater System w/ Solar Panel

(B) Utilized Electronics and Battery

FIGURE B.16: Deployed Overwater Equipment

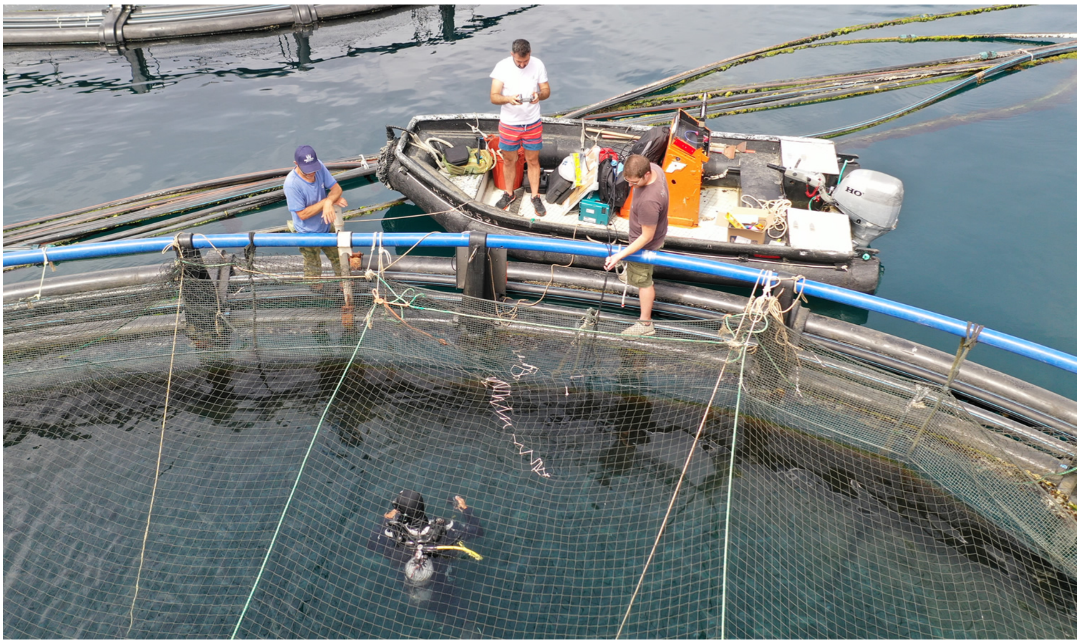
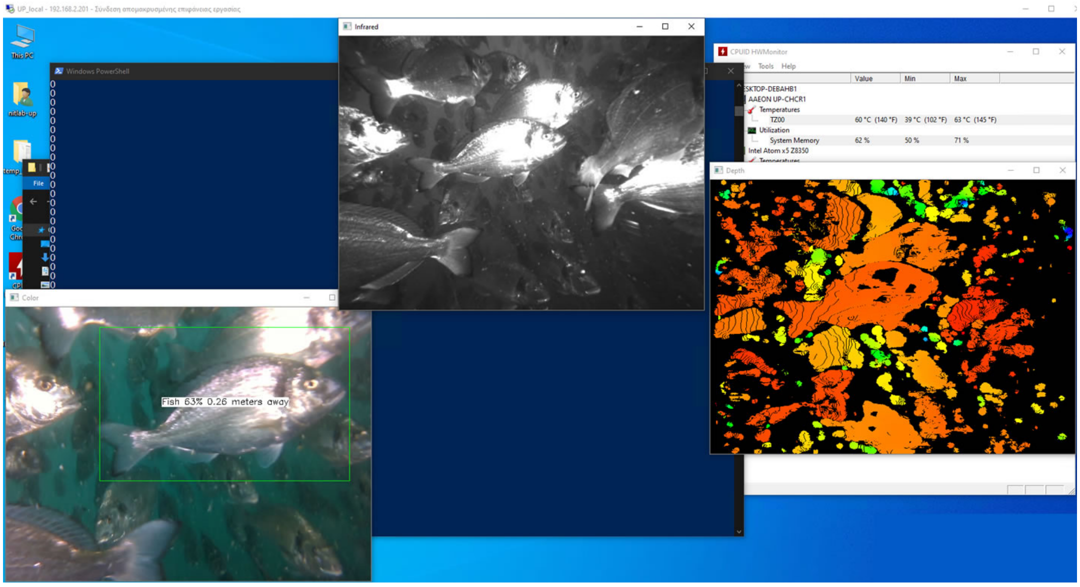


FIGURE B.17: Aerial Image During the Undewater Camera Installation



```
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
```

	Value	Min	Max
SKTOP-DEBAHB1			
AAEON UP-CHCR1			
Temperatures			
TZ00	60 °C (140 °F)	39 °C (102 °F)	63 °C (145 °F)
Utilization			
System Memory	62 %	50 %	71 %
Intel Atom x5 Z8350			
Temperatures			

Fish 63% 0.26 meters away

FIGURE B.18: Footage from the Undewater Camera System

Bibliography

[1] *"Libelium Waspmote"*, http://www.libelium.com/products/waspmote/.

[2] R. Abu-Aisheh et al. *Exploration and Mapping using a Sparse Robot Swarm: Simulation Results*. Tech. rep. Inria, June 2020.

[3] Four ways to improve your ADC system power supply rejection Texas Instruments. https://tinyurl.com/y24x82ox.

[4] C. Adjih et al. "FIT IoT-LAB: A Large Scale Open Experimental IoT Testbed". In: *IEEE World Forum on Internet of Things (IEEE WF-IoT)*. Milan, Italy, 2015.

[5] S. Ahmed et al. "Demystifying Energy Consumption Dynamics in Transiently Powered Computers". In: *ACM Trans. Embed. Comput. Syst.* (2020).

[6] S. Ahmed et al. "Intermittent Computing with Dynamic Voltage and Frequency Scaling". In: *Proceedings of the 2020 International Conference on Embedded Wireless Systems and Networks on Proceedings of the 2020 International Conference on Embedded Wireless Systems and Networks*. EWSN '20. Lyon, France: Junction Publishing, 2020.

[7] Saad Ahmed, Junaid Haroon Siddiqui, and Muhammad Hamad Alizai. "Intermittent Computing with Dynamic Voltage and Frequency Scaling". In: *Proc. of EWSN '20*. 2020.

[8] Lora Alliance. https://goo.gl/DOROCS.

[9] Alphasense Sensor Technology. http://www.alphasense.com/.

[10] Maxim Integrated MAX9923 Ultra-Precision High-Side Current-Sense Amplifier.

[11] Angelos-Christos Anadiotis et al. “A new slicing scheme for efficient use of wireless testbeds”. In: *Proc. 4th ACM Intl Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH)*. 2009.

[12] Dilin Anand and Anagha P. *Golden Rules for Designing Wearables*. June 2015.

[13] M. P. Andersen, G. Fierro, and D. E. Culler. “System Design for a Synergistic, Low Power Mote/BLE Embedded Platform”. In: *In Proc. of IPSN ’16*. 2016.

[14] F. A. Aoudia et al. “Long-short range communication network leveraging LoRa™ and wake-up receiver”. In: *Microprocessors and Microsystems* (2018).

[15] A. Apostolaras et al. “A Demonstration of a Relaying Selection Scheme for Maximizing a Diamond Network’s Throughput”. In: 2012.

[16] A. Apostolaras et al. “C2M: Mobile data offloading to mesh networks”. In: *2014 IEEE Global Communications Conference*. 2014.

[17] A. Apostolaras et al. “On the implementation of relay selection strategies for a cooperative diamond network”. In: *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2013.

[18] *Arduino platform*. URL: http://www.arduino.cc/.

[19] M. Asadpour et al. “Micro aerial vehicle networks: an experimental analysis of challenges and opportunities”. In: *IEEE Communications Magazine* (2014).

[20] S.. Assimonis, S. Daskalakis, and A. Bletsas. “Sensitive and Efficient RF Harvesting Supply for Batteryless Backscatter Sensor Networks”. In: *IEEE Transactions on Microwave Theory and Techniques* (2016).

[21] A. Augustin et al. “A study of LoRa: Long range & low power networks for the internet of things”. In: *Sensors* (2016).

[22] A. Bansal et al. “OwLL: Accurate LoRa Localization Using the TV Whitespaces”. In: *Proceedings of the 20th International Conference on Information Processing in Sensor Networks (Co-Located with CPS-IoT Week 2021)*. 2021.

[23] A. Bartolini et al. "OptoCOMM: Development and experimentation of a new optical wireless underwater modem". In: *OCEANS 2016 MTS/IEEE Monterey* (2016).

[24] *Beaglebone PRU-ICSS Resources*. http://beagleboard.org/pru.

[25] Mark Berman et al. "GENI: A federated testbed for innovative network experiments". In: *Computer Networks, Elsevier* 63 (2014).

[26] Norbert Blenn and Fernando A. Kuipers. "LoRaWAN in the Wild: Measurements from The Things Network". In: *CoRR* (2017). URL: http://arxiv.org/abs/1706.03086.

[27] C. A. Boano et al. "IoTBench: Towards a Benchmark for Low-Power Wireless Networking". In: *2018 IEEE Workshop on Benchmarking Cyber-Physical Networks and Systems (CPSBench)*. 2018.

[28] BeagleBone Black Rev. C Embedded board.

[29] S. Bodini et al. "Automated micro Loop Flow Reactor technology to measure nutrients in coastal water: State of the art and field application". In: *Proc. of OCEANS '15*.

[30] G. Boquet et al. "LoRa-E: Overview and Performance Analysis". In: *IEEE Communications Magazine* (2020).

[31] G. Boquet et al. "LR-FHSS: Overview and Performance Analysis". In: *IEEE Communications Magazine* (2021).

[32] M. Bor and U. Roedig. "LoRa transmission parameter selection". In: *2017 13th Int. Conf. on Distributed Computing in Sensor Systems (DCOSS)*. 2017.

[33] M. Bor, JE. Vidler, and U. Roedig. "LoRa for the Internet of Things". In: (2016).

[34] Athanassios Boulis et al. "A wireless sensor network test-bed for structural health monitoring of bridges". In: *Proc. 36th IEEE Conf on Local Computer Networks (LCN)*. 2011.

[35] O. Brunecker and M. Magno. "TinyBird: An Energy Neutral Acoustic Bluetooth-Low-Energy Sensor Node with RF Energy Harvesting". In: *Proceedings of the*

7th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems. ENSsys'19. New York, NY, USA: Association for Computing Machinery, 2019.

[36] Jeffrey A Burke et al. "Participatory sensing". In: *Proc. 1st ACM Workshop on World-Sensor-Web: Mobile Device Centric Sensory Networks and Applications (WSW), 4th ACM Conf on Embedded Networked Sensor Systems (SenSys)*. 2006.

[37] Dave Bursky. *Understand the Power-Saving Options on Embedded Controllers to Improve System Power Efficiency*. Digi-Key Electronics, January 2012.

[38] Dave Bursky. *Understanding MCU sleep states and energy savings*. March 2012.

[39] F. Cabarcas, J. Aranda, and D. Mendez. "OpenWuR - An Open WSN Platform for WuR-Based Application Prototyping". In: *Proceedings of the 2020 International Conference on Embedded Wireless Systems and Networks on Proceedings of the 2020 International Conference on Embedded Wireless Systems and Networks*. EWSN '20. Lyon, France: Junction Publishing, 2020.

[40] B. Campbell, B. Ghena, and P. Dutta. "Energy-Harvesting Thermoelectric Sensing for Unobtrusive Water and Appliance Metering". In: *Proceedings of the 2nd International Workshop on Energy Neutral Sensing Systems (ENSsys)*. ENSsys '14. Memphis, Tennessee, 2014.

[41] B. Campbell, B.n Ghena, and P. Dutta. "Energy-harvesting Thermoelectric Sensing for Unobtrusive Water and Appliance Metering". In: *in Proc. of ENSsys '14*. 2014.

[42] G. Cario et al. "SeaModem: A low-cost underwater acoustic modem for shallow water communication". In: *OCEANS 2015 - Genova* (2015).

[43] L. Casals et al. "Modeling the energy performance of LoRaWAN". In: *Sensors* (2017).

[44] M. Cattani, C. Boano, and K. Römer. "An experimental evaluation of the reliability of lora long-range low-power wireless communication". In: *Journal of Sensor and Actuator Networks* (2017).

[45] G. Charalampidis, A. Papadakis, and M. Samarakou. "Power estimation of RF energy harvesters". In: *Energy Procedia* (2019). Technologies and Materials for Renewable Energy, Environment and Sustainability (TMREES).

[46] K. Choumas et al. “SDN Controller Placement and Switch Assignment for Low Power IoT”. In: *Electronics* 9 (2020).

[47] K. Chounos, N. Makris, and T. Korakis. “Enabling Distributed Spectral Awareness for Disaggregated 5G Ultra-Dense HetNets”. In: *2019 IEEE 2nd 5G World Forum (5GWF)*. 2019.

[48] K. Chounos et al. “Fast Spectral Assessment for Handover Decisions in 5G Networks”. In: *2019 16th IEEE Annual Consumer Communications Networking Conference (CCNC)*. 2019.

[49] A. Colin, E. Ruppel, and B. Lucia. “A Reconfigurable Energy Storage Architecture for Energy-Harvesting Devices”. In: *SIGPLAN Not.* (2018).

[50] A. Colin, E. Ruppel, and B. Lucia. “A Reconfigurable Energy Storage Architecture for Energy-Harvesting Devices”. In: *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*. ASPLOS ’18. Williamsburg, VA, USA: Association for Computing Machinery, 2018.

[51] Analog Devices ADCMP601 Rail-to-Rail TTL/CMOS Comparator.

[52] Microchip MCP4728 12-Bit Quad Digital-to-Analog Converter.

[53] *Copernicus Marine Monitoring Service*. https://goo.gl/gJ5EKZ.

[54] M. D’Addato et al. “Nanowatt Clock and Data Recovery for Ultra-Low Power Wake-Up Based Receivers”. In: *Proceedings of the 2020 International Conference on Embedded Wireless Systems and Networks on Proceedings of the 2020 International Conference on Embedded Wireless Systems and Networks*. EWSN ’20. Lyon, France: Junction Publishing, 2020.

[55] *Debian Operating System*. https://www.debian.org/intro/about.

[56] Vivien Delport. *Low-Power Wireless Sensor Networks*. May 2015.

[57] Behnam Dezfouli, Immanuel Amirtharaj, and Chia-Chi (Chelsey) Li. “EMPIOT: An energy measurement platform for wireless IoT devices”. In: *Journal of Network and Computer Applications* (2018).

[58] *Digi S6B WiFi Interface*. URL: http://goo.gl/YuT5Ta.

[59] Texas Instruments ADS8332 Analog to Digital Converter.

[60] A. Dongare et al. “OpenChirp: A low-power wide-area networking architecture”. In: *2017 IEEE Int. Conf. on Pervasive Computing and Communications Workshops (PerCom Workshops)*. 2017.

[61] A. Duda et al. “WiSH-WalT: A Framework for Controllable and Reproducible LoRa Testbeds”. In: *IEEE PIMRC*. 2018.

[62] P. Dutta et al. “Putting the Software Radio on a Low-Calorie Diet”. In: *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. Hotnets-IX. Monterey, California: Association for Computing Machinery, 2010.

[63] Prabal Dutta et al. “A Building Block Approach to Sensornet Systems”. In: *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*. SenSys ’08. Raleigh, NC, USA: Association for Computing Machinery, 2008.

[64] Prabal Dutta et al. “Energy Metering for Free: Augmenting Switching Regulators for Real-Time Monitoring”. In: *Proceedings of the 7th International Conference on Information Processing in Sensor Networks*. IPSN ’08. St. Louis, MO, USA, 2008.

[65] Shane B Eisenman et al. “BikeNet: A mobile sensing system for cyclist experience mapping”. In: *ACM Transactions on Sensor Networks (TOSN)* 6.1 (2009).

[66] *Environmental Measurements, SMS research project*. goo.gl/HSd3iJ.

[67] *Evologics S2CR 18/34 Underwater Acoustic Modem*. https://www.evologics.de/en/products/acoustics/s2cr-18-34.html.

[68] eZ430-RF2500. *https://tinyurl.com/7z7zav9*.

[69] Xenofon Fafoutis et al. “Designing Wearable Sensing Platforms for Healthcare in a Residential Environment”. In: *PHAT* 3.12 (Sept. 2017).

[70] Jeff Falin. *Measuring Efficiency of the BQ25504 Energy Harvesting Battery Charger*. Texas Instruments, July 2013.

[71] A. Ferreira et al. "A study of the LoRa signal propagation in forest, urban, and suburban environments". In: *Annals of Telecommunications - annales des télécommunications* (July 2020). URL: https://hal.inria.fr/hal-02907283.

[72] *FIRE: Future Internet Research and Experimentation initiative official Website*. URL: http://www.ict-fire.eu/.

[73] FP7-ICT-2007.1.6-224460. Project WISEBED. http://www.wisebed.eu.

[74] FRAM New Generation of Non-Volatile Memory. *https://tinyurl.com/y8rgdz68*.

[75] D. Friesel, L. Kaiser, and O. Spinczyk. "Automatic Energy Model Generation with MSP430 EnergyTrace". In: CPS-IoTBench '21. New York, NY, USA: Association for Computing Machinery, 2021. ISBN: 9781450384391.

[76] G. U. Gamm et al. "Low power wake-up receiver for wireless sensor nodes". In: *Proc. of ISSNIP '10*. 2010.

[77] K. Geissdoerfer et al. "Getting more out of energy-harvesting systems: energy management under time-varying utility with PreAct". In: *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*. IPSN '19. Montreal, Quebec, Canada: Association for Computing Machinery, 2019.

[78] Kai Geissdoerfer, Mikołaj Chwalisz, and Marco Zimmerling. "Shepherd: A Portable Testbed for the Batteryless IoT". In: *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. SenSys '19. New York, New York: Association for Computing Machinery, 2019.

[79] *GEOSS Earth Observations*. https://goo.gl/FhtEwc.

[80] P. Giaccone et al. "Design and implementation of a belief-propagation scheduler for multicast traffic in input-queued switches". In: *Computer Communications* 103 (2017).

[81] Chris Glaser. *Three Keys to Ultra-Low-Power in Wearable Designs*. September 2014.

[82] A. Gomez et al. “Dynamic energy burst scaling for transiently powered systems”. In: *Design, Automation Test in Europe Conf. Exhibition (DATE) ’16*. Mar. 2016.

[83] *Google Maps API*. URL: http://goo.gl/aqsBzH.

[84] Guy Hoover. *Consequences of Not Following the Data Sheet Bypass Recommendation*. [2010].

[85] Hubert Halloin, Pierre Prat, and Julien Brossard. *Long term characterization of voltage references*. 2013.

[86] Hank Zumbahlen. *Staying Well Grounded*. [2012].

[87] D. Hasenfratz et al. “Pushing the Spatio-Temporal Resolution Limit of Urban Air Pollution Maps”. In: *Proc. 12th Intl Conf on Pervasive Computing and Communications (PerCom)*. 2014.

[88] D. Hasenfratz et al. “Pushing the spatio-temporal resolution limit of urban air pollution maps”. In: *2014 IEEE Int. Conf. on Pervasive Computing and Communications (PerCom)*. 2014.

[89] A. Hergenroeder, J. Wilke, and D. Meier. “Distributed Energy Measurements in WSN Testbeds with a Sensor Node Management Device (SNMD)”. In: *Proceedings of the 23th International Conference on Architecture of Computing Systems 2010*. 2010.

[90] J. Hester, L. Sitanayah, and J. Sorber. “Tragedy of the Coulombs: Federating Energy Storage for Tiny, Intermittently-Powered Sensors”. In: *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. SenSys ’15. Seoul, South Korea: Association for Computing Machinery, 2015.

[91] P. P. Jayaraman et al. “Efficient Opportunistic Sensing using Mobile Collaborative Platform MOSDEN”. In: *Proc. 9th IEEE Intl Conf on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*. 2013.

[92] V. Jelicic. “Power Management in Wireless Sensor Networks with High-Consuming Sensors”. In: 2011.

[93] V. Jelicic et al. “Analytic Comparison of Wake-up Receivers for WSNs and Benefits over the Wake-on Radio Scheme”. In: *Proceedings of the 7th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*. PM2HW2N ’12. Paphos, Cyprus: Association for Computing Machinery, 2012.

[94] Xiaofan Jiang et al. “Micro Power Meter for Energy Monitoring of Wireless Sensor Networks at Scale”. In: *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*. IPSN ’07. 2007.

[95] N. Jovalekic et al. “Experimental Study of LoRa Transmission over Seawater”. In: *Sensors* 18 (2018).

[96] R. Jurdak et al. “Camazotz: Multimodal activity-based GPS sampling”. In: *2013 ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. 2013.

[97] R. Jurdak et al. “Opal: A Multiradio Platform for High Throughput Wireless Sensor Networks”. In: *IEEE Embedded Systems Letters* (2011).

[98] Aleksandras Kaknevicius. *When to use load switches in place of discrete MOSFETs*. February 2016.

[99] E. Kampianakis et al. “A Remotely Programmable Modular Testbed for Backscatter Sensor Network Research”. In: *Real-World Wireless Sensor Networks*. Ed. by K. Langendoen et al. Springer International Publishing, 2014.

[100] Z. Kapetanovic et al. *No Size Fits All: Automated Radio Configuration for LPWANs*. 2021. arXiv: 2109.05103 [cs.NI].

[101] M. Katsomallos and S. Lalis. “EasyHarvest: Supporting the Deployment and Management of Sensing Applications on Smartphones”. In: *Proc. 1st IEEE Workshop on Crowdsensing Methods, Techniques, and Applications (Crowdsensing), 12th IEEE Conf on Pervasive Computing and Communications (PerCom)*. 2014.

[102] G. Kazdaridis et al. “A Demonstration of the NITOS BikesNet Framework”. In: *2014 IEEE 15th International Conference on Mobile Data Management*. Vol. 1. 2014.

[103] G. Kazdaridis et al. “A Novel Architecture for Semi-Active Wake-Up Radios Attaining Sensitivity Beyond -70 DBm: Demo Abstract”. In: *Proceedings of the 20th International Conference on Information Processing in Sensor Networks (Co-Located with CPS-IoT Week 2021)*. New York, NY, USA: Association for Computing Machinery, 2021.

[104] G. Kazdaridis et al. “An Integrated Chassis Manager Card Platform Featuring Multiple Sensor Modules”. In: *TRIDENTCOM*. 2012.

[105] G. Kazdaridis et al. “Enabling Asynchronous Awakenings in Wireless Sensor Networks Towards Removing Duty-Cycle Barriers”. In: *Proceedings of the 11th Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization*. WiNTECH ’17. Snowbird, Utah, USA, 2017.

[106] G. Kazdaridis et al. “EProfiler: High-Precision Power Monitoring System for IoT Devices Featuring Extreme Dynamic Range of Operation”. In: *Proc. of the ENSsys ’20*. New York, NY, USA, 2020.

[107] G. Kazdaridis et al. “Evaluation of LoRa Performance in a City-Wide Testbed: Experimentation Insights and Findings”. In: *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*. WiNTECH ’19. Los Cabos, Mexico, 2019.

[108] G. Kazdaridis et al. “EVERUN: Enabling Power Consumption Monitoring in Underwater Networking Platforms”. In: *Proceedings of the 11th Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization*. WiNTECH ’17. Snowbird, Utah, USA, 2017.

[109] G. Kazdaridis et al. *eWake: A Novel Architecture for Semi-Active Wake-Up Radios Attaining Ultra-High Sensitivity at Extremely-Low Consumption*. 2021. arXiv: 2103.15969 [cs.AR].

[110] G. Kazdaridis et al. “In-situ Power Consumption Meter for Sensor Networks supporting Extreme Dynamic Range”. In: *Proceedings of the 11th Workshop on Wireless Network Testbeds, Experimental Evaluation & CHaracterization*. WiNTECH ’17. Snowbird, Utah, USA, 2017.

[111] G. Kazdaridis et al. “Nano Power Draw in Duty-Cycled Wireless Sensor Networks”. In: *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*. WiNTECH ’19. Los Cabos, Mexico, 2019.

[112] G. Kazdaridis et al. “Nano-Things: Pushing Sleep Current Consumption to the Limits in IoT Platforms”. In: *Proceedings of the 10th International Conference on the Internet of Things*. IoT ’20. Malmö, Sweden, 2020.

[113] G. Kazdaridis et al. “NITOS BikesNet: Enabling Mobile Sensing Experiments through the OMF Framework in a City-Wide Environment”. In: *Proceedings of the 15th International Conference on Mobile Data Management*. IEEE MDM ’14. 2014.

[114] G. Kazdaridis et al. “Novel Metrics and Experimentation Insights for Dynamic Frequency Selection in Wireless LANs”. In: *Proceedings of the 6th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*. WiNTECH ’11. Las Vegas, Nevada, USA: Association for Computing Machinery, 2011. ISBN: 9781450308670.

[115] G. Kazdaridis et al. “On the development of energy-efficient communications for marine monitoring deployments”. In: *2017 13th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*. 2017.

[116] G. Kazdaridis et al. “The ICARUS Mote: Employing Off-Chip RTC to Attain 22 nA Sleep Current in Duty-Cycled IoT Devices”. In: *Proceedings of the 10th International Conference on the Internet of Things*. IoT ’20 Companion. Malmö, Sweden, 2020.

[117] G. Kazdaridis et al. “The NITOS Wireless Sensor Network Testbed for Experimenting with Long-Range Technologies”. In: *10th International Conference on the Internet of Things Companion*. IoT ’20 Companion. Malmö, Sweden, 2020.

[118] S. Keranidis et al. “Experimental Evaluation and Comparative Study on Energy Efficiency of the Evolving IEEE 802.11 Standards”. In: *Proceedings of the 5th Int. Conf. on Future Energy Systems*. e-Energy ’14. 2014.

[119] S. Keranidis et al. “NITOS Energy Monitoring Framework: Real Time Power Monitoring in Experimental Wireless Network Deployments”. In: *SIGMOBILE Mob. Comput. Commun. Rev.* (2014).

[120] S. Keranidis et al. “NITOS mobile monitoring solution: realistic energy consumption profiling of mobile devices”. In: *Proceedings of the 5th international conference on Future energy systems* (2014).

[121] S. Keranidis et al. “Online energy consumption monitoring of wireless testbed infrastructure through the NITOS EMF framework”. In: *Proceedings of the 8th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*. WiNTECH ’13. Miami, Florida, USA, 2013.

[122] A. Keshavarzian, H. Lee, and L. Venkatraman. “Wakeup Scheduling in Wireless Sensor Networks”. In: *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. MobiHoc ’06. Florence, Italy: Association for Computing Machinery, 2006.

[123] R. Khereddine et al. “Adaptive logical control of RF LNA performances for efficient energy consumption”. In: *IFIP/IEEE Int. Conf. on Very Large Scale Integration-System on a Chip*. Springer. 2010.

[124] Nordic Power Profiler Kit.

[125] A. Kozłowski and J. Sosnowski. “Energy Efficiency Trade-Off Between Duty-Cycling and Wake-Up Radio Techniques in IoT Networks”. In: *Journal on Wireless Personal Communications*. 2019.

[126] U. Kulau, F. Büsching, and L. Wolf. “A Node’s Life: Increasing WSN Lifetime by Dynamic Voltage Scaling”. In: *2013 IEEE International Conference on Distributed Computing in Sensor Systems*. 2013.

[127] P. La Rosa R.and Livreri et al. “Strategies and Techniques for Powering Wireless Sensor Nodes through Energy Harvesting and Wireless Power Transfer”. In: *Sensors* 19 (2019).

[128] M. T. Lazarescu. “Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications”. In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 3 (2013).

[129] A. Ledeczi et al. “Multiple simultaneous acoustic source localization in urban terrain”. In: *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.* 2005.

[130] C. Li and B. Dezfouli. “Excalibur: An Accurate, Scalable, and Low-Cost Calibration Tool for Sensing Devices”. In: *IEEE Sensors Journal* (2019).

[131] JC. Liando et al. “Known and Unknown Facts of LoRa: Experiences from a Large-scale Measurement Study”. In: *ACM Transactions on Sensor Networks (TOSN)* (2019).

[132] LoRa SX1272 Networking Guide Libelium’s. `http://tinyurl.com/y6nrb877`.

[133] R. Lim et al. “FlockLab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems”. In: *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*. IPSN ’13. Philadelphia, Pennsylvania, USA, 2013.

[134] Q. Liu et al. “Characterizing Packet Loss in City-Scale LoRaWAN Deployment: Analysis and Implications”. In: *2020 IFIP Networking Conference (Networking)*. 2020.

[135] Ting Liu et al. “Implementing Software on Resource-Constrained Mobile Sensors: Experiences with Impala and ZebraNet”. In: *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services*. MobiSys ’04. Boston, MA, USA: Association for Computing Machinery, 2004.

[136] D. Lymberopoulos and A. Savvides. “XYZ: A motion-enabled, power aware sensor node platform for distributed sensor network applications”. In: *In Proc. of IPSN ’05*. 2005.

[137] B. Maag, Z. Zhou, and L. Thiele. “W-Air: Enabling Personal Air Pollution Monitoring on Wearables”. In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* (2018).

[138] David Mage et al. “Urban air pollution in megacities of the world”. In: *Atmospheric Environment* (). Supercities: Environment Quality and Sustainable Development.

[139] V. Maglogiannis et al. “Enabling mobile sensing through a DTN framework”. In: *Proc. 8th ACM Intl workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH)*. 2013.

[140] M. Magno et al. “Design, Implementation, and Performance Evaluation of a Flexible Low-Latency Nanowatt Wake-Up Radio Receiver”. In: *IEEE Transactions on Industrial Informatics* (2016).

[141] MS. Mahmoud and AAH. Mohamad. “A study of efficient power consumption wireless communication techniques/modules for internet of things (IoT) applications”. In: (2016).

[142] Make The Most Of Your MCU Sleep Modes. https://tinyurl.com/yxvxh2lu.

[143] N. Makris et al. “Cloud-based convergence of heterogeneous RANs in 5G disaggregated architectures”. In: *2018 IEEE International Conference on Communications (ICC)*. IEEE. 2018.

[144] N. Makris et al. “Employing MEC in the Cloud-RAN: An Experimental Analysis”. In: *Proceedings of the 2018 on Technologies for the Wireless Edge Workshop*. 2018.

[145] N. Makris et al. “On minimizing service access latency: Employing MEC on the fronthaul of heterogeneous 5G architectures”. In: *2019 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*. IEEE. 2019.

[146] S. Marathe et al. “CurrentSense: A novel approach for fault and drift detection in environmental IoT sensors”. In: *ACM/IEEE International Conference on Internet of Things Design and Implementation (IoTDI)*. ACM. 2021.

[147] S. J. Marinkovic and E. M. Popovici. “Nano-Power Wireless Wake-Up Receiver With Serial Peripheral Interface”. In: *IEEE Journal on Selected Areas in Communications* (2011).

[148] P. Martin, Z. Charbiwala, and M. Srivastava. “DoubleDip: Leveraging Thermoelectric Harvesting for Low Power Monitoring of Sporadic Water Use”. In: *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. SenSys ’12. Toronto, Ontario, Canada, 2012.

[149] R. Martins et al. “SUNRISE project: Porto university testbed”. In: *Proc of UComms ’14*.

[150] Riccardo Masiero et al. “DESERT Underwater: An NS-Miracle-based framework to design, simulate, emulate and realize test-beds for underwater network protocols”. In: *Proc. of Italian Networking Workshop ’13*.

[151] Olivier Mehani et al. “An instrumentation framework for the critical task of measurement collection in the future internet”. In: *Computer Networks, Elsevier* 63 (2014).

[152] M. Meo et al. “The TREND experimental activities on “green” communication networks”. In: *2013 24th Tyrrhenian International Workshop on Digital Communications - Green ICT (TIWDC)*. 2013, pp. 1–6.

[153] *Meshlium WSN device*. goo.gl/tyD8d5.

[154] Current Ranger Power Meter. *https://lowpowerlab.com/guide/currentranger/*.

[155] MICAz Wireless System. *http://tiny.cc/7jr69y*.

[156] Michael Cobb. *Powering High Speed Analog-to-Digital Converters with Switching Power Supplies*. [2009].

[157] F. Michelinakis et al. *Dissecting Energy Consumption of NB-IoT Devices Empirically*. 2020. arXiv: 2004.07127 [cs.NI].

[158] Micro Crystal RV1805-C3 Real-Time-Clock. *https://tinyurl.com/y5v9fklx*.

[159] Freescale MK20DX128VLH5 micro-processor. *https://goo.gl/eoT8xx*.

[160] R. Min, T. Furrer, and A. Chandrakasan. “Dynamic Voltage Scaling Techniques for Distributed Microsensor Networks”. In: *Proceedings of the IEEE Computer Society Annual Workshop on VLSI (WVLSI’00)*. WVLSI ’00. USA: IEEE Computer Society, 2000.

[161] ADALM-PLUTO Software-Defined Radio Active Learning Module. *https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html*.

[162] P. Moinzadeh et al. “Dynamic Voltage Scaling Techniques for Energy Efficient Synchronized Sensor Network Design”. In: 2011.

[163] Molloy Derek. *Exploring BeagleBone: tools and techniques for building with embedded Linux*. [2019].

[164] Geoff Mulligan. “The 6LoWPAN architecture”. In: *Proc. 4th ACM workshop on Embedded networked sensors*. 2007.

[165] J. Munoz et al. “OpenTestBed: Poor Man’s IoT Testbed”. In: *IEEE INFOCOM - CNERT : Workshop on Computer and Networking Experimental Research using Testbeds*. Paris, France, 2019.

[166] M. Nardello et al. “Camaroptera: A Batteryless Long-Range Remote Visual Sensing System”. In: *Proceedings of the 7th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*. ENSsys’19. New York, NY, USA: Association for Computing Machinery, 2019.

[167] *Network Implementation Testbed Laboratory*. URL: http://nitlab.inf.uth.gr.

[168] *Network Implementation Testbed using Open Source platforms*. URL: http://goo.gl/j67I6k.

[169] H. Niavis, N. Papadis, and L. Tassiulas. *A Blockchain-based Decentralized Data Sharing Infrastructure for Off-grid Networking*. 2020. arXiv: 2006.07521 [cs.DC].

[170] H. Niavis et al. “Enabling Sensing and Mobility on Wireless Testbeds”. In: *Testbeds and Research Infrastructure. Development of Networks and Communities*. Springer Berlin Heidelberg, 2012.

[171] P. Nintanavongsa et al. “Design Optimization and Implementation for RF Energy Harvesting Circuits”. In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 2.1 (2012).

[172] Akos Pasztor. WuLoRa: Energy-Efficient IoT Sensor Node. *https://akospasztor.com/projects/wulora/report.pdf*.

[173] KE. Nolan, W. Guibene, and MY. Kelly. “An evaluation of low power wide area network technologies for the Internet of Things”. In: *2016 Int. wireless communications and mobile computing conference (IWCMC)*. 2016.

[174] P. Oppermann and C. Renner. “Low-Power Ultrasonic Wake-Up and Communication through Structural Elements”. In: *Proceedings of the 7th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*. ENSsys’19. New York, NY, USA: Association for Computing Machinery, 2019.

[175] World Health Organization. *”7 million premature deaths annually linked to air pollution”*. Geneva, 2014.

[176] Roger P. Centelles et al. "LoRaMoto: A communication system to provide safety awareness among civilians after an earthquake". In: *Future Generation Computer Systems* (2021).

[177] M. Pawar, JA Manore, and M. Kuber. "Life Time Prediction of Battery Operated Node for Energy Efficient WSN Applications". In.

[178] Y. Peng et al. "PLoRa: A Passive Long-Range Data Network from Ambient LoRa Transmissions". In: *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. SIGCOMM '18. Budapest, Hungary: Association for Computing Machinery, 2018.

[179] J. Petäjäjärvi et al. "Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage". In: *Int. Journal of Distributed Sensor Networks* (2017).

[180] T. Petrić et al. "Measurements, performance and analysis of LoRa FABIAN, a real-world implementation of LPWAN". In: *2016 IEEE 27th Annual Int. Symp. on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2016.

[181] C. Petrioli et al. "The SUNRISE GATE: Accessing the SUNRISE federation of facilities to test solutions for the Internet of Underwater Things". In: *Proc. of UComms '14*. 2014.

[182] Chiara Petrioli et al. "The SUNSET framework for simulation, emulation and at-sea testing of underwater wireless sensor networks". In: *Proc. of Ad Hoc Networks '15*. 2015.

[183] Congduc Pham. "Low-Cost, Low-Power and Long-Range Image Sensor for Visual Surveillance". In: *Proceedings of the 2nd Workshop on Experiences in the Design and Implementation of Smart Objects*. SmartObjects '16. New York City, New York: Association for Computing Machinery, 2016.

[184] J. Polastre, R. Szewczyk, and D. Culler. "Telos: Enabling Ultra-low Power Wireless Research". In: *Proc. of IPSN '05*. 2005.

[185] E. Popovici, M. Magno, and S. Marinkovic. "Power management techniques for Wireless Sensor Networks: A review". In: *5th IEEE International Workshop on Advances in Sensors and Interfaces IWASI*. 2013.

[186] A. Pötsch et al. “A power measurement system for accurate energy profiling of embedded wireless systems”. In: *Proceedings of IEEE Emerging Technology and Factory Automation (ETFA)*. 2014.

[187] uCurrent Power Meter. http://tiny.cc/dz069y.

[188] R. Pozza, A. Gluhak, and M. Nati. “SmartEye: An Energy-Efficient Observer Platform for Internet of Things Testbeds”. In: *Proceedings of the Seventh ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization*. WiNTECH ’12. Istanbul, Turkey: Association for Computing Machinery, 2012.

[189] Prabal Dutta et al. “Design of a wireless sensor network platform for detecting rare, random, and ephemeral events”. In: *In Proc. of IPSN ’05*. 2005.

[190] Murugavel Raju and Mark Grazier. *ULP meets energy harvesting: A game-changing combination for design engineers*. Texas Instruments, April 2010.

[191] Thierry Rakotoarivelo, Guillaume Jourjon, and Max Ott. “Designing and Orchestrating Reproducible Experiments on Federated Networking Testbeds”. In: *Computer Networks, Elsevier* 63 (2014).

[192] Xavier Ramus. *Minimum power specifications for high-performance ADC power-supply designs*. April 2016.

[193] L. Ravindranath et al. “Code In The Air: Simplifying Sensing and Coordination Tasks on Smartphones”. In: *Proc. 12th ACM Workshop on Mobile Computing Systems & Applications (HotMobile)*. 2012.

[194] Dipankar Raychaudhuri et al. “Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols”. In: *Proc. 3rd IEEE Conf on Wireless Communications and Networking (WCNC)*. 2005.

[195] Sasank Reddy et al. “Biketastic: sensing and mapping for better biking”. In: *Proc. 28th ACM SIGCHI Conf on Human Factors in Computing Systems*. 2010.

[196] Analog Devices ADM7151 Ultra Low Noise High PSRR RF Linear Regulator.

[197] Texas Instruments TPS7A49 Low-noise High-PSRR Low-dropout Voltage Regulator.

[198] B. Renner, J. Heitmann, and F. Steinmetz. “Ahoi: Inexpensive, Low-Power Communication and Localization for Underwater Sensor Networks and uAUVs”. In: *ACM Trans. Sen. Netw.* (2020).

[199] C. Renner and A. J. Golkowski. “Acoustic Modem for Micro AUVs: Design and Practical Evaluation”. In: *Proceedings of the 11th ACM International Conference on Underwater Networks Systems*. WUWNet ’16. Shanghai, China, 2016.

[200] C. Renner and V. Turau. “CapLibrate: Self-Calibration of an Energy Harvesting Power Supply with Supercapacitors”. In: *ARCS Workshops*. 2010.

[201] C. Renner and V. Turau. “State-of-charge assessment for supercap-powered sensor nodes: Keep it simple stupid!” In: *2012 Ninth International Conference on Networked Sensing (INSS)*. 2012.

[202] SX1272/73 - 860 MHz to 1020 MHz Low Power Long Range Transceiver. Rev. 3.1 - March 2017. *http://tiny.cc/tw8w8y*.

[203] M. Riedel and M. Schwarzbach. *Current measurement with auto-ranging yields 180dB range dynamics*. http://www.imc-berlin.com/.

[204] Rob Reeder. *Designing Power Supplies for High Speed ADC*. [2012].

[205] M. Rottleuthner, T. C. Schmidt, and M. Wählisch. “Eco: A Hardware-Software Co-Design for InSitu Power Measurement on Low-End IoT Systems”. In: *Proceedings of the 7th International Workshop on Energy Harvesting & Energy-Neutral Sensing Systems*. ENSsys’19. 2019.

[206] M. Rottleuthner, T. C. Schmidt, and M. Wählisch. “Sense Your Power: The ECO Approach to Energy Awareness for IoT Devices”. In: *ACM Trans. Embed. Comput. Syst.* (2021).

[207] Michel Rottleuthner, Thomas C. Schmidt, and Matthias Wählisch. “Sense Your Power: The ECO Approach to Energy Awareness for IoT Devices”. In: *ACM Trans. Embed. Comput. Syst.* (Mar. 2021).

[208] *Roving Networks WiFly - RN-171XV 802.11 b/g*.

[209] Zheng Ruan, EC-H Ngai, and Jiangchuan Liu. "Wireless sensor network deployment in mobile phones assisted environment". In: *Proc. 18th Intl Workshop on Quality of Service (IWQoS)*. 2010.

[210] Luis Sanchez et al. "SmartSantander: Experimentation and service provision in the smart city". In: *Proc. 16th IEEE Intl Symposium on Wireless Personal Multimedia Communications (WPMC)*. 2013.

[211] Luis Sanchez et al. "SmartSantander: IoT experimentation over a smart city testbed". In: *Computer Networks* 61 (2014). Special issue on Future Internet Testbeds - Part I.

[212] M. Sandhu et al. *Towards Optimal Kinetic Energy Harvesting for the Batteryless IoT*. 2020. arXiv: `2002.08887 [eess.SP]`.

[213] Michiel Sandra, Sara Gunnarsson, and Anders J Johansson. *Internet of Buoys: An Internet of Things Implementation at Sea*. 2020. arXiv: `2012.05653 [eess.SY]`.

[214] Y. Sangar and B. Krishnaswamy. "WiChronos: Energy-Efficient Modulation for Long-Range, Large-Scale Wireless Networks". In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. MobiCom '20. London, United Kingdom: Association for Computing Machinery, 2020.

[215] Sanjay Pithadia and Shridhar More. *Grounding in mixed-signal systems demystified, Part 1*. [2013].

[216] Sanjay Pithadia and Shridhar More. *Grounding in mixed-signal systems demystified, Part 2*. [2013].

[217] N. Saoda and B. Campbell. "No Batteries Needed: Providing Physical Context with Energy-Harvesting Beacons". In: *Proceedings of the 7th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*. ENSsys'19. New York, NY, USA: Association for Computing Machinery, 2019.

[218] A. Schulman et al. "BattOr: Plug-and-debug energy debugging for applications on smartphones and laptops Stanford CS Tech Report". In: 2016.

[219] Norihisa Segawa et al. *A Marine Experiment of a Long Distance Communication Sensor Network -MAD-SS-*.

[220] *SENSES Lab*. http://senseslab.di.uniroma1.it/.

[221] Honeywell HPMA115S0 PM2.5 Particle sensor.

[222] I. B. Shirokov, A. S. Mironov, and A. N. Grekov. "Ocean Surface State Monitoring with Drifters Array". In: *2020 Zooming Innovation in Consumer Technologies Conference (ZINC)*. 2020.

[223] *SHT11 Temp & Hum Sensor*. URL: http://goo.gl/lnzf6N.

[224] Sensirion Temperature & Humidity SHTC3. *http://tiny.cc/s43u8y*.

[225] VISHAY Shunt - Current Shunts and Current-Sensing Resistors. *https://tinyurl.com/yxg2mtv4*.

[226] L. Sigrist et al. "Measurement and validation of energy harvesting IoT devices". In: *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*. 2017.

[227] A. Silva, M. Liu, and M. Moghaddam. "Power-Management Techniques for Wireless Sensor Networks and Similar Low-Power Communication Devices Based on Nonrechargeable Batteries". In: *Journal of Comp. Networks and Com.* (2012).

[228] *Sparkfun Wake-on-shake Board*.

[229] D. Spenza et al. "Beyond duty cycling: Wake-up radio with selective awakenings for long-lived wireless sensing systems". In: *In Proc. of INFOCOM '15*. 2015.

[230] D. Spenza et al. "Beyond duty cycling: Wake-up radio with selective awakenings for long-lived wireless sensing systems". In: *In Proc. of INFOCOM '15*. 2015.

[231] D. Stavropoulos et al. "Demonstration of a Vehicle-to-Infrastructure (V2I) Communication Network Featuring Heterogeneous Sensors and Delay Tolerant Network Capabilities". In: 2012.

[232] D. Stavropoulos et al. "Demonstration of a Vehicle-to-Infrastructure (V2I) Communication Network Featuring Heterogeneous Sensors and Delay Tolerant Network Capabilities." In: *Proc. 8th Int. ICST Conf on Testbeds and Research Infrastructure. Development of Networks and Communities (TRIDENTCOM)*. 2012.

[233] D. Stavropoulos et al. "Enabling experimentation in mobile sensing scenarios through 4G networks: The NITOS approach". In: *2015 European Conference on Networks and Communications, EuCNC 2015, Paris, France, June 29 - July 2, 2015*. IEEE, 2015.

[234] *SUNRISE FP7 research project*. `http://fp7-sunrise.eu`.

[235] Felix Sutton et al. "Zippy: On-Demand Network Flooding". In: *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. SenSys '15. Seoul, South Korea: Association for Computing Machinery, 2015. ISBN: 9781450336314.

[236] I. Syrigos et al. "Enabling Wireless LAN Troubleshooting". In: *Passive and Active Measurement - 16th International Conference, PAM 2015, New York, NY, USA, March 19-20, 2015, Proceedings*. Ed. by J. Mirkovic and Y. Liu. Lecture Notes in Computer Science. Springer, 2015.

[237] D. Syrivelis et al. "Bits and coins: Supporting collaborative consumption of mobile internet". In: *2015 IEEE Conference on Computer Communications (INFOCOM)*. 2015.

[238] K. Takahagi et al. "Low-Power Wake-up Receiver with Subthreshold CMOS Circuits for Wireless Sensor Networks". In: *Analog Integr. Circuits Signal Process.* (2013).

[239] V. Talla et al. "LoRa Backscatter: Enabling The Vision of Ubiquitous Connectivity". In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* (2017).

[240] *Texas Instruments INA225 Current Monitor*.

[241] *Texas Instruments REF3233 reference circuit*.

[242] P. Tian et al. "ChirpBox: An Infrastructure-Less LoRa Testbed". In: EWSN '21. Delft, The Netherlands: Junction Publishing, 2021.

[243] Texas Instruments TPL5111 Nano-Power System Timer.

[244] Roman Trüb et al. "FlockLab 2: Multi-Modal Testing and Validation for Wireless IoT". en. In: 3rd Workshop on Benchmarking Cyber-Physical Systems and Internet of Things (CPS-IoTBench 2020); September 21-25, 2020; Zurich: ETH Zurich, Computer Engineering and Networks Laboratory (TIK), 2020-09.

[245] Yu Tselishchev and Athanassios Boulis. "Wireless sensor network tesbed for structural health monitoring of bridges". In: *Proc. 8th IEEE Conf Sensors*. 2009.

[246] Pere Tuset-Peiró, Xavier Vilajosana, and Thomas Watteyne. "OpenMote+: A Range-Agile Multi-Radio Mote". In: *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*. EWSN '16. Graz, Austria: Junction Publishing, 2016.

[247] Panagiotis Tzimotoudis et al. "Demo: LoRa Mesh Network Experimentation in a City-Wide Testbed". In: WiNTECH '19. Los Cabos, Mexico: Association for Computing Machinery, 2019.

[248] *uBlox 50 Channel D2523T Helical GPS Receiver*.

[249] *UPorto - LSTS (Underwater Systems and Technology Laboratory)*. http://lsts.fe.up.pt.

[250] *Vida oceanographic buoy*. goo.gl/7ymJvX.

[251] P. Volgyesi et al. "Shooter Localization and Weapon Classification with Soldier-Wearable Networked Sensors". In: *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*. MobiSys '07. San Juan, Puerto Rico: Association for Computing Machinery, 2007.

[252] P. Völgyesi et al. "Air Quality Monitoring with SensorMap". In: *Proceedings of the 7th International Conference on Information Processing in Sensor Networks*. IPSN '08. USA: IEEE Computer Society, 2008.

[253] Walt Kester. *Which ADC Architecture Is Right for Your Application?* [2015].

[254] A. Wang et al. "EcoBT: Miniature, Versatile Mote Platform Based on Bluetooth Low Energy Technology". In: *In Proc. of iThings '14*. Sept. 2014.

[255] L. Wang and Y. Xiao. “Energy saving mechanisms in sensor networks”. In: *2nd International Conference on Broadband Networks, 2005.* 2005.

[256] M. Wijesundara et al. “Design of a Kinetic Energy Harvester for Elephant Mounted Wireless Sensor Nodes of JumboNet”. In: *2016 IEEE Global Communications Conference (GLOBECOM)*. 2016.

[257] J. M. Williams et al. “Proof-of-Concept for an IoT Sensor Platform with 2.4GHz Wake-up Radio”. In: *2020 IEEE Radio and Wireless Symposium (RWS)*. 2020.

[258] *XBee Wireless Module*. URL: http://www.digi.com/xbee/.

[259] Chuanwei Xu, Zhongyi Zheng, and Changqing Ji. “Practical Deployments of SEMAT on Wireless Sensor Networks in the Marine Environment”. In: *Proc. of Mobile Ad-hoc and Sensor Networks ’13*.

[260] D. Yang et al. “Understanding Power Consumption of NB-IoT in the Wild: Tool and Large-Scale Measurement”. In: *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. MobiCom ’20. London, United Kingdom: Association for Computing Machinery, 2020. ISBN: 9781450370851.

[261] W. Ye and J. Heidemann. “Wireless Sensor Networks”. In: ed. by C. S. Raghavendra, Krishna M. Sivalingam, and Taieb Znati. Norwell, MA, USA: Kluwer Academic Publishers, 2004. Chap. Medium Access Control in Wireless Sensor Networks.

[262] L. Yerva et al. “Grafting Energy-Harvesting Leaves onto the Sensornet Tree”. In: *Proceedings of the 11th International Conference on Information Processing in Sensor Networks (IPSN)*. IPSN ’12. Beijing, China, 2012.

[263] AM. Yousuf et al. “Throughput, Coverage and Scalability of LoRa LPWAN for Internet of Things”. In: *2018 IEEE/ACM 26th Int. Symp. on Quality of Service (IWQoS)*. 2018.

[264] P. Zhang et al. “EkhoNet: High Speed Ultra Low-Power Backscatter for next Generation Sensors”. In: *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*. MobiCom ’14. Maui, Hawaii, USA: Association for Computing Machinery, 2014.

[265] P. Zhang et al. "Hardware Design Experiences in ZebraNet". In: *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*. SenSys '04. Baltimore, MD, USA: Association for Computing Machinery, 2004.

[266] P. Zhang et al. "HitchHike: Practical Backscatter Using Commodity WiFi". In: *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. SenSys '16. Stanford, CA, USA: Association for Computing Machinery, 2016. ISBN: 9781450342636.

[267] Z. Zhang et al. "Long-term Monitoring of the Sierra Nevada Snowpack Using Wireless Sensor Networks". In: *IEEE Internet of Things Journal* (2020).

[268] R. Zhou and G. Xing. "Nemo: A High-fidelity Noninvasive Power Meter System for Wireless Sensor Networks". In: *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*. IPSN '13. Philadelphia, Pennsylvania, USA, 2013.