ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Σχεδίαση και ανάπτυξη εικονικής επισκόπησης δεδομένων σε Ίντερνετ των Πραγμάτων για την βιομηχανία.

Κοσμάς Σιδηρόπουλος

Επιβλέπων

Σταμούλης Γεώργιος

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις (1), που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί χωρίς να τα περικλείω σε εισαγωγικά και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.

2. Δέχομαι ότι η αυτολεξεί παράθεση χωρίς εισαγωγικά, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

Περίληψη

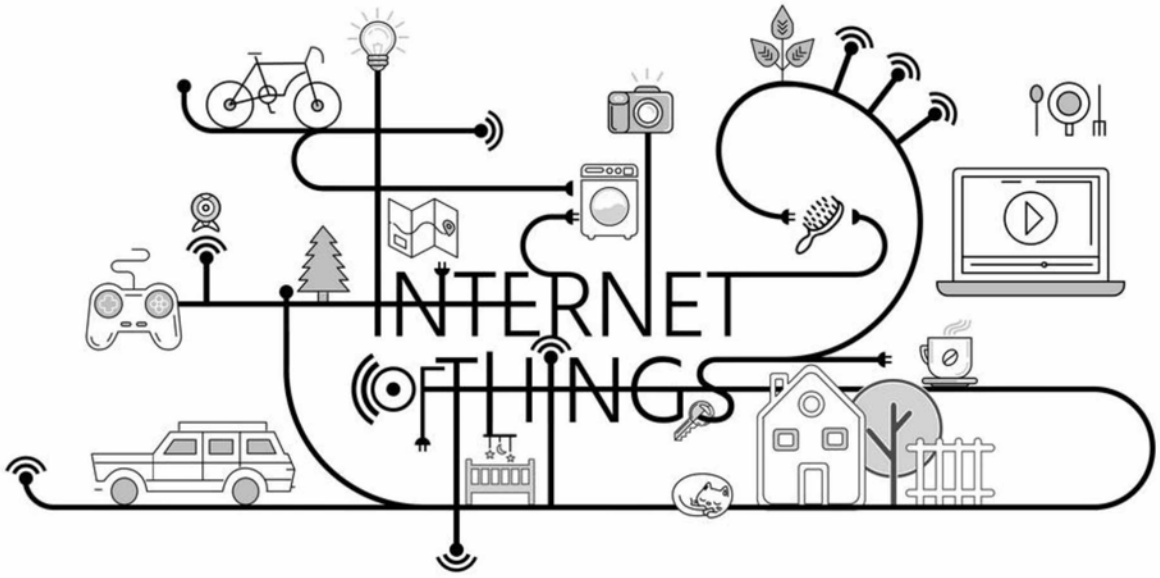
Διανύουμε μια εποχή όπου οι έξυπνες συσκευές βρίσκονται παντού στην καθημερινότητα μας. Από τα έξυπνα ρολόγια, που μετράνε βήματα και αξιολογούν τις επιδόσεις μας μέχρι τιςεργοστασιακές συσκευές, που ελέγχουν πυρηνικούς αντιδραστήρες. Οι συγκεκριμένες συσκευές αποτελούν αυτό που σήμερα ονομάζουμε Ίντερνετ των Πραγμάτων. Όλες οι πληροφορίες αυτές καταλήγουν σε μια βάση δεδομένων και εκμεταλλεύονται από μια διαδικτυακή εφαρμογή σε έναν υπολογιστή ή ένα κινητό ή μια οποιαδήποτε άλλη συσκευή, που επικοινωνεί με το διαδίκτυο. Σε αυτή την εργασία θα παρουσιαστεί ηυλοποίηση ενός πρωτοτύπου μιας έξυπνης συσκευής, που θα μπορούσε να χρησιμοποιηθεί, σαν ένα Βιομηχανικό Ίντερνετ των Πραγμάτων προϊόν. Ακόμα θα εστιάσουμε στα βήματα σχεδίασης και ανάπτυξης μιας διαδικτυακής εφαρμογής, η οποία θα εμφανίζει βασικές πληροφορίες στον πραγματικό χρόνο της συσκευής στην εικονική και επαυξημένη πραγματικότητα.

Περιεχόμενα

1. Το Διαδίκτυο των πραγμάτων

1.1 Ορισμός

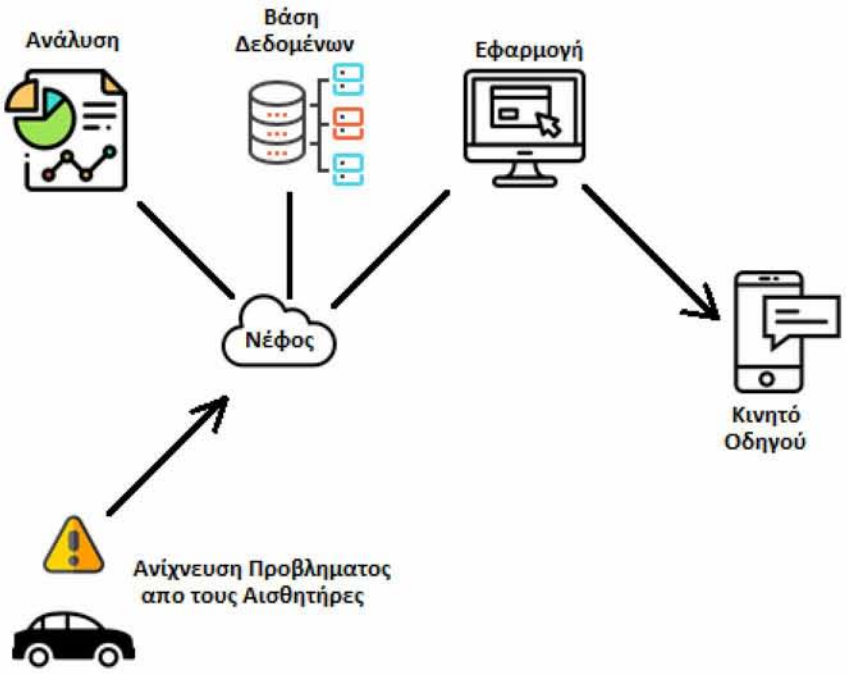
Το **Ίντερνετ των πραγμάτων (Internet of Things)** αποτελεί το δίκτυο επικοινωνίας ανάμεσα στις ηλεκτρονικές συσκευές, που συνδέουν αισθητήρες (sensors) ή ενεργοποιητές (actuators) μέσω κάποιου λογισμικό με το διαδίκτυο, ώστε να επιτρέπει τον έλεγχο, την ανταλλαγή και την επεξεργασία δεδομένων.



1.2 Πως δουλεύει

Οι συσκευές συλλέγουν πληροφορίες από τον φυσικό κόσμο μέσα από έναν αριθμό αισθητήρων. Έπειτα, οι πληροφορίες αυτές μεταφέρονται στο διαδίκτυο μέσω διάφορων μοντέλων επικοινωνίας όπου εκεί θα αναλυθούν από κάποιο υπολογιστικό σύστημα και θα παρθούν κάποιες αποφάσεις ανάλογα με την λογική και τους κανόνες, που έχει ορίσει ο κατασκευαστής στο σύστημα. Από εκεί και πέρα οι πληροφορίες αυτές μπορούν να χρησιμοποιηθούν σε διάφορες άλλες εφαρμογές ή και να σταλθεί κάποια απάντηση με βάση αυτών των πληροφοριών πίσω στις συσκευές. Για παράδειγμα,το αυτοκίνητο ενός ανθρώπου που οδηγεί, έχει επάνω του ενσωματωμένους αισθητήρες,οι οποίοι ανιχνεύουν χαμηλή πίεση στα φρένα του αυτοκινήτου. Η πληροφορία αυτή στέλνεται μέσω του δορυφόρου σε κάποιο υπολογιστικό σύστημα της εταιρίας,που ανήκει το αμάξι, οπού εκεί τις επεξεργάζεται. Στο σύστημα αυτό υπάρχει ένας κανόνας, που ορίζει ότι, αν ένα αμάξι έχει χαμηλή πίεση στα

φρένα τότε πρέπει να σταλθεί ένα μήνυμα στο κινητό του πελάτη το οποίο θα αναφέρει το πρόβλημα. Παράλληλα αυτή η πληροφορία μαζί με την τρέχουσα τοποθεσία του αυτοκινήτου,χρησιμοποιείται από μια εφαρμογή στο σύστημα,η οποία εμφανίζει συνεργαζόμενα συνεργεία, που μπορούν να λύσουν το πρόβλημα του αυτοκινήτου. Έτσι, λοιπόν, πέρα από το μήνυμα που ενημερώνει τον οδηγό για το πρόβλημα, η εφαρμογή στέλνει και το πιο κοντινό διαθέσιμο συνεργείο, για να αναλάβει την επισκευή του αυτοκινήτου. Με αυτόν τον τρόπο, ο οδηγός ξέρει πώς να λύσει το πρόβλημα του αυτοκινήτου, απλά κοιτάζοντας το κινητό του.



Εικόνα 1.1 – Ένα σύστημα IoT

1.3 Πεδία εφαρμογής

Εξαιτίας της ραγδαίας εξέλιξης της τεχνολογίας, είναι σχεδόν απίθανο να προβλέψει κανείς όλες τις πιθανές εφαρμογές του IoT. Ο όρος ‘‘Ίντερνετ των Πραγμάτων’’ χρησιμοποιήθηκε για πρώτη φορά, μόλις, πριν σχεδόν 20 χρόνια. Πρόκειται για μια νέα τεχνολογία, που μέσα σε αυτά τα χρόνια κατάφερε να εισχωρήσει στις ζωές μας με τον εξωφρενικό αριθμό των 25 δισεκατομμυρίων συσκευών. Οι έρευνες έχουν υποδείξει πως μέχρι το 2020 θα έχουν συνδεθεί 50 δισεκατομμύρια συσκευές στο Ίντερνετ. Σήμερα οι εφαρμογές του IoT χωρίζονται σε διάφορες κατηγορίες και στησυγκεκριμένη εργασία θα αναλυθεί το Βιομηχανικό Ίντερνετ των Πραγμάτων (IndustrialInternetofThings).

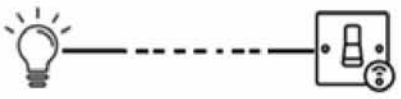
1.4 Συνδεσιμότητα

Τον Μάρτιο του 2015 το Συμβούλιο Αρχιτεκτονικής του Διαδικτύου (InternetArchitectureBoard) δημοσίευσε ένα έγγραφο για την δικτύωση των έξυπνων συσκευών, που περιγράφει τέσσερα διαφορετικά μοντέλα επικοινωνίας και χρησιμοποιούνται από συσκευές IoT.

Εικόνα 1.2 – Μοντέλα επικοινωνίας

Το Μοντέλο Επικοινωνίας : Συσκευή-προς-Συσκευή (Device-to-Device)

Το μοντέλο σύνδεσης Συσκευή-με-Συσκευή επιτρέπει σε μια ή περισσότερες συσκευές να επικοινωνούν μεταξύ τους μέσω του διαδικτύου χωρίς την μεσολάβηση κάποιου εξυπηρετητή (server). Αυτές οι συσκευές έχουν συμφωνήσει να επικοινωνούν με ένα συγκεκριμένο πρωτόκολλο επικοινωνίας όπως Bluetooth, Z-Wave, ZigBeeκτλ. Συνήθως, χρησιμοποιείται σε εφαρμογές που δεν επικοινωνούν με μεγάλο όγκο δεδομένων και δεν επηρεάζονται από τον χρόνο αποστολής.



Εικόνα 1.2 Μοντέλο επικοινωνίας Συσκευή-προς-Συσκευή

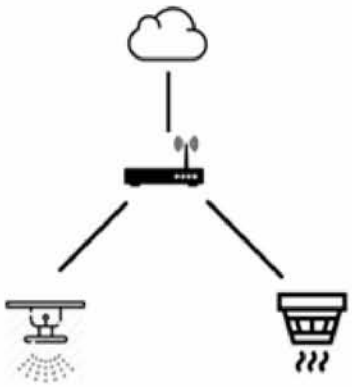
Το Μοντέλο Επικοινωνίας : Συσκευή-προς-Νέφος (Device-to-Cloud)

Το μοντέλο αυτό συνδέει την συσκευή IoT απευθείας μεέναν πάροχομιας διαδικτυακής υπηρεσίας νέφους. Σε αυτό το μοντέλο δίνεται η ικανότητα τηςεξ' αποστάσεως επικοινωνίας με την συσκευή. Συνήθως χρησιμοποιούνται επικοινωνίες όπως το Ethernetκαι το Wi-Fiγια ενσύρματες ή ασύρματες αντίστοιχα συνδέσεις ανάμεσα στην συσκευή και τον πάροχο.

Εικόνα 1.3 - Μοντέλο επικοινωνίας Συσκευή-προς-Νέφος

Το Μοντέλο Επικοινωνίας : Συσκευή-προς-Πύλη-Δικτύου (Device-to-Gateway)

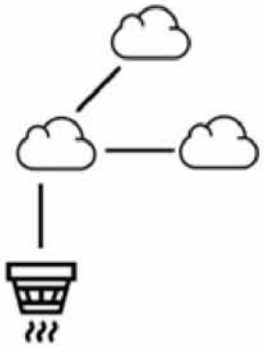
Σε αυτή την σύνδεση οι πληροφορίες πάνε στον πάροχο αποκλειστικά μέσω κάποιου δίαυλου επικοινωνίας, συνήθως μια πύλη δικτύου. Από εκεί οι πληροφορίες επεξεργάζονται και ο πάροχος έχει την δυνατότητα να στείλει δεδομένα πίσω στην πύλη, η οποία θα τα δρομολογήσει, ώστε να φτάσουν στην κατάλληλη συσκευή. Ιδανικό μοντέλο για να συνδέει έξυπνες συσκευές με το smartphone,το οποίο στην συνέχεια μέσω μιας εφαρμογής επικοινωνεί με τον πάροχο.



Εικόνα 1.4 - Μοντέλο επικοινωνίας Συσκευή-προς-Πύλη-Δικτύου

Το Μοντέλο Επικοινωνίας : BACK-ENDΑνταλλαγής-Δεδομένων (BACK-ENDData-SharingModel)

Είναι μια επέκταση του μοντέλου Συσκευή-προς-Νέφος που δίνει την ικανότητα στον χρήστη να εξάγει και να αναλύει πληροφορίες των έξυπνων συσκευών από έναν πάροχομαζί με δεδομένα άλλων πηγών.



Εικόνα 2 - Μοντέλο Επικοινωνίας BACK-END Ανταλλαγής-Δεδομένων

2. Ίντερνετ των Πραγμάτων για την Βιομηχανία

2.1 Ορισμός

Το Ίντερνετ των Πραγμάτων για την Βιομηχανία (IndustrialInternetofThings) έχει λάβει πολλέςονομασίες από εταιρίες όπως ‘‘IndustrialInternet’’, ‘‘InternetofEverything’’ ή και ‘‘Internet 4.0’’. Καθώς δημιουργούνται περισσότερες Machine-to-Machine (M2M) εφαρμογές, η συνδεσιμότητα στο Διαδίκτυο θα γίνει απαραίτητη σε πολλές βαριές βιομηχανίες σε όλο τον κόσμο και αυτό, γιατί ο όγκος των δεδομένων που πρέπει να αναλυθεί αυξάνεται συνεχώς. Πρέπει να γίνει αντιληπτό ότι το IIoT έχει διαφορετικά στοχευμένα ακροατήρια, τεχνικές απαιτήσεις και στρατηγικές. Για παράδειγμα, η καταναλωτική αγορά του IoT, ενδιαφέρεται περισσότερο για έξυπνα σπίτια, wearables, που συλλέγουν και επεξεργάζονται δεδομένα φυσικής κατάστασης (συνολικά βήματα, χρόνος τρεξίματος, κ.τ.λ.) και διάφορες άλλες συσκευές gadgets για διασκέδαση. Παρομοίως, και η εμπορική αγορά ενδιαφέρεται για υπηρεσίες, που συμπεριλαμβάνουν οικονομικά και επενδυτικά προϊόντα όπως τραπεζικές εργασίες, ασφάλεια, οικονομικές υπηρεσίες και το ηλεκτρονικό εμπόριο, που εστιάζουν στο ιστορικό, την απόδοση και την αξία του καταναλωτή.

Εικόνα 2.1 – Επιστημονικοί κλάδοι IIoT

Το Ίντερνετ των Πραγμάτων για την Βιομηχανία περιλαμβάνει ένα τεράστιο αριθμό επιστημονικών κλάδων όπως βιομηχανοποίηση, γεωργία, υγειονομική περίθαλψη, λιανική, μεταφορές, αεροπορία, διαστημικά ταξίδια και πολλά άλλα. Πολλοί βιομηχανικοί ηγέτες

προβλέπουν ότι το βιομηχανικό Ίντερνετ θα προσφέρει πρωτοφανή επίπεδα ανάπτυξης και παραγωγικότητας κατά την επόμενη δεκαετία. Οι ηγέτες των επιχειρήσεων, οι κυβερνήσεις, οι ακαδημαϊκοί και οι πωλητές τεχνολογίας εργάζονται μαζί, για να προσπαθήσουν να αξιοποιήσουν και να εκμεταλλευτούν αυτές τις τεράστιες δυνατότητες.

2.2 Πλεονεκτήματα του IIoT

Ένα από τα πιο σημαντικά πλεονεκτήματα του Βιομηχανικού Ίντερνετ είναι η δυνατότητα πρόβλεψηςβλάβης κάποιου μηχανήματος. Οι αισθητήρες, που έχουν ενσωματωθεί πάνω στις μηχανές μεταφέρουν πληροφορίες σε κάποιο σύστημα IIoT, το οποίο μπορεί να προβλέψει, αν η μηχανή χρειάζεται κάποια συντήρηση, πριν παύσει πια να λειτουργεί, γλιτώνοντας πολλά έξοδα στην εταιρία.

Ένα άλλο πλεονέκτημα είναι η βελτίωση της εξυπηρέτησης πελατών για κάποιο προϊόν. Οι τεχνολογίες του IIoT ενημερώνουν τους τεχνικούς μέσω τουπροσδιορισμού πιθανών προβλημάτων, που μπορεί να αντιμετωπίζουν τα προϊόντα τους. Έτσι, οι τεχνικοί μπορούν να επιλύσουν το πρόβλημα ταχύτερα και ο πελάτης δεν θα ενοχληθεί.

Τα IIoT μπορούν να χρησιμοποιηθούν, για να λύσουν προβλήματα, που υπάρχουν και στην συγκοινωνία. Χιλιάδες συσκευές, που παρακολουθούν τους δρόμους συνδεμένες σε κάποιο κέντρο, μπορούν να ανιχνεύουν την κίνηση, που υπάρχει στον δρόμο και να προειδοποιούν τον οδηγό με εναλλακτικές διαδρομές για τον προορισμό του. Επίσης, όταν συμβαίνουν ατυχήματα στον δρόμο οι έξυπνες συσκευές θα το εντοπίζουν και ανάλογα με το είδος του ατυχήματος θα ειδοποιούν την οδική ασφάλεια ή ενδεχομένως κάποιο ασθενοφόρο. Πολλά από τα προβλήματα αυτά θα επιλύονται σε τόσο μικρό χρονικό διάστημα, που οι συγκοινωνίες θα βελτιωθούν σε τεράστιο βαθμό.

Παρόλο που οι βιομηχανικές επιχειρήσεις έχουν αισθητήρες και συσκευές, που επεξεργάζουν δεδομένα (M2M) εδώ και πολλά χρόνια το IIoT είναι ακόμα σε πρώιμο στάδιο. Αυτό γίνεται, επειδή πολλές βιομηχανίες διστάζουν να υιοθετήσουν το IIoT και δεν γνωρίζουν πως θα επηρεάσει τις ήδη υπάρχουσες βιομηχανίες, τις επιχειρησιακές στρατηγικές και το πιο σημαντικό την παραγωγικότητα και τα προϊόντα. Είναι μια λογική αντίδραση, αφού το Βιομηχανικό Ίντερνετ συνδυάζει ένα υψηλό και πολύπλοκο επίπεδο τεχνολογιών, που δεν πρέπει να εφαρμοστεί, αν δεν γίνει απόλυτα κατανοητό.

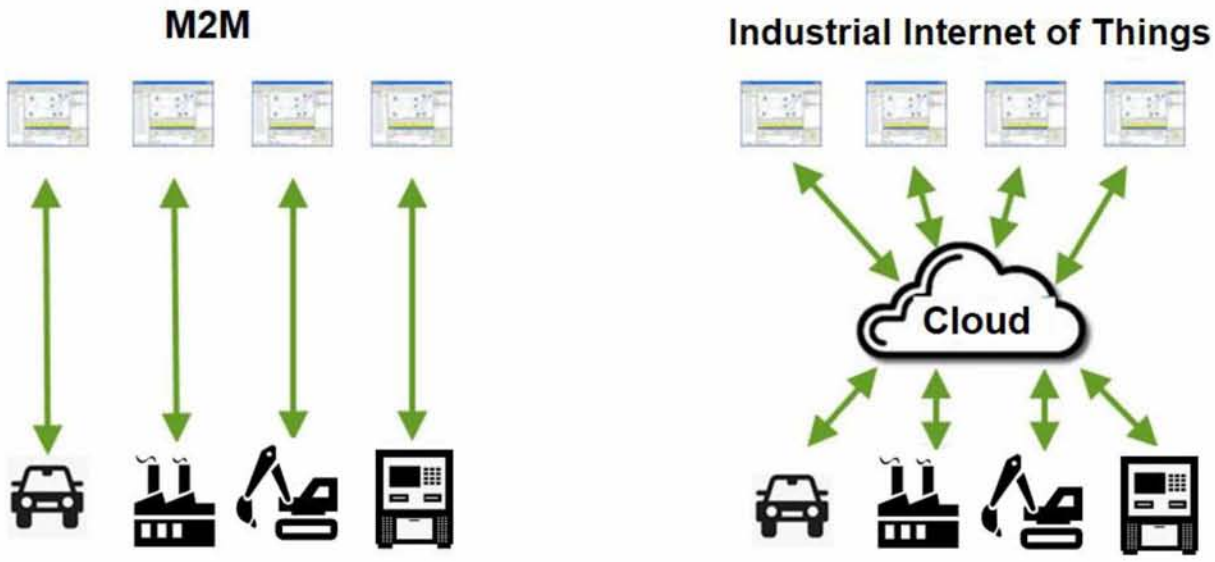
2.3 Διαφορές IIoT με IoT

Το IoT με το IIoT έχουν πολλές κοινές τεχνολογίες όπως συστήματα νέφους, αισθητήρες, μοντέλα επικοινωνίας, ανάλυση δεδομένων και επικοινωνία μηχανή με μηχανή (M2M). Όμως, όπως ήδη αναφέραμε, όλες αυτές οι τεχνολογίες χρησιμοποιούνται για διαφορετικούς σκοπούς.

Φυσικά υπάρχουν κάποιες παράμετροι, οι οποίες διαφέρουν από IoT σε IIoT. Πρώτη παράμετρος είναι η ασφάλεια η οποία είναι κρίσιμη για όλων των ειδών IoT, όμως στο βιομηχανικό τομέα απαιτεί πιο ισχυρά μέτρα λόγω του μεγέθους και του μεγαλύτερου αντίκτυπου σε περιπτώσεις παραβίασης. Μια άλλη παράμετρος είναι το μέγεθος του δικτύου, που στην βιομηχανία είναι απαραίτητο να υποστηρίζεται η επικοινωνία χιλιάδων αισθητήρων, μηχανών και άλλου είδους συσκευές. Τέλος, το δίκτυο πρέπει να έχει πολύ μικρότερη καθυστέρηση στηνανίχνευση, στην απόφαση ή στην εκτέλεση,διαφορετικάμπορεί να προκαλέσει πολλά προβλήματα στην ποιότητα των προϊόντων, την ασφάλεια των εργατών κτλ. Επομένως, το IIoT απαιτεί πολύ μεγαλύτερη ταχύτητα αποστολής της πληροφορίας.

3. Αρχιτεκτονική του IIoT

Πριν την δημιουργία των IoT,η επικοινωνία ανάμεσα στις συσκευές γινόταναποκλειστικά μέσω κάποιας άλλης συσκευής. Αυτή η τεχνολογία είναι γνωστή ως M2M (MachinetoMachine). Σήμερα το IoTείναι μια φυσική εξέλιξη αυτής της τεχνολογίας.



Εικόνα 3.3 - M2M και IIoT αρχιτεκτονικές

Η εικόνα 3.1 συγκρίνει την αρχιτεκτονική M2M με την αρχιτεκτονική του IIoT. Αμέσως μπορούμε να παρατηρήσουμε ότι η μόνη διαφορά σε αυτό τουψηλό επίπεδο είναι η προσθήκη του Internet.

3.1 Three-Tier Topology

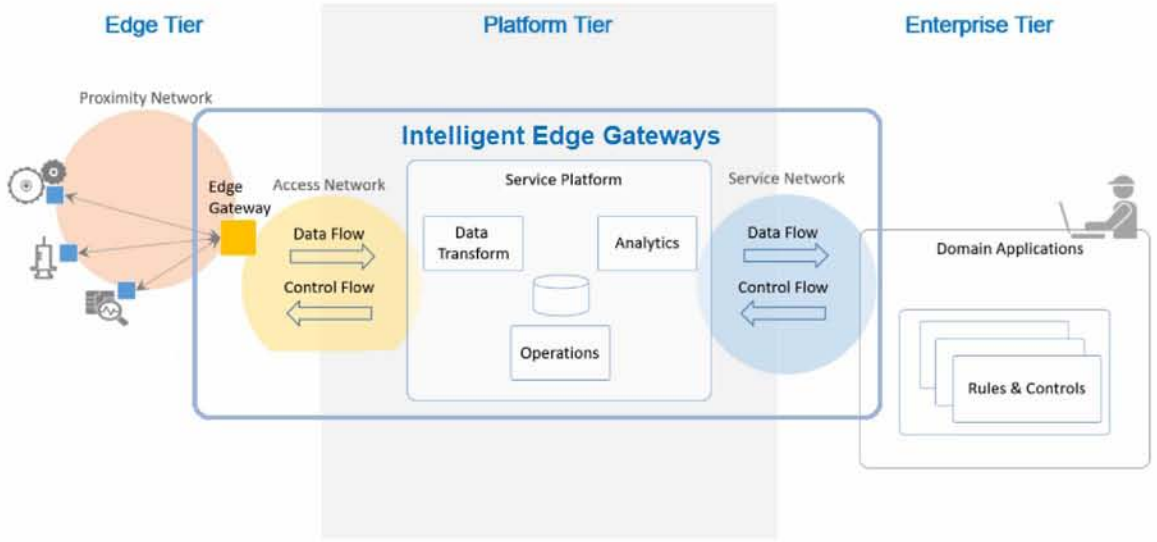
ΗαρχιτεκτονικήThree-TierTopology(εικόνα 3.2) είναι ο βέλτιστος τρόπος, για να περιγράψουμε ένα σύστημα IIoT.Είναι ένας πολύ απλός τρόπος, για να περιγράψουμε την αρχιτεκτονική ενός συστήματος βιομηχανικού Ίντερνετ (IndustrialInternetSystem). Όπως δηλώνει και το όνομά της, η αρχιτεκτονική χωρίζεται σε τρία μέρη.

Η βαθμίδα του άκρου (EdgeTier) συλλέγει όλες τις πληροφορίες, τις ταξινομεί και τις στέλνει μέσω ενός δίαυλου επικοινωνίας σε μια πύλη του δικτύου (gateway). Ανάλογα με τα πρωτόκολλα επικοινωνίας και την τεχνολογία, που χρησιμοποιούνται στην βαθμίδα του άκρου μπορεί να υπάρξουν κόμβοι συλλογής πληροφορίας (όταν συνήθως υπάρχουν πολλές πληροφορίες από διαφορετικά σημεία/αισθητήρες), πριν φτάσουν στην πύλη του

δικτύου(gateway-mediatededge). Η δημιουργία των κόμβων γίνεται με ποικίλους τρόπους όπως με την ύπαρξη διάφορων μικροελεγκτών που συλλέγουν πληροφορίες από περισσότερους αισθητήρες.

Η βαθμίδα της πλατφόρμας (PlatformTier) δέχεται πληροφορίες από την βαθμίδα του άκρου μέσω του διαδικτύου και είναι υπεύθυνη για την μετατροπή και επεξεργασία αυτών των πληροφοριών. Ακόμα είναι υπεύθυνη,διαχείρισης δεδομένων, που έρχονται από την τρίτη και τελευταία βαθμίδα συνήθως, για να τα στέλνει πίσω στην βαθμίδα του άκρου. Τις περισσότερες φορές αυτές οι πληροφορίες, που μεταφέρονται από την τρίτη στην πρώτη βαθμίδα είναι εντολές χειρισμού (π.χ. έλεγχος κάποιου servοκινητήρα).

Τέλος, η βαθμίδα της επιχείρησης (EnterpriseTier) υλοποίει την εφαρμογή ή τους κανόνεςπου επιβάλλουν οι λειτουργικές προδιαγραφές για την εφαρμογή.



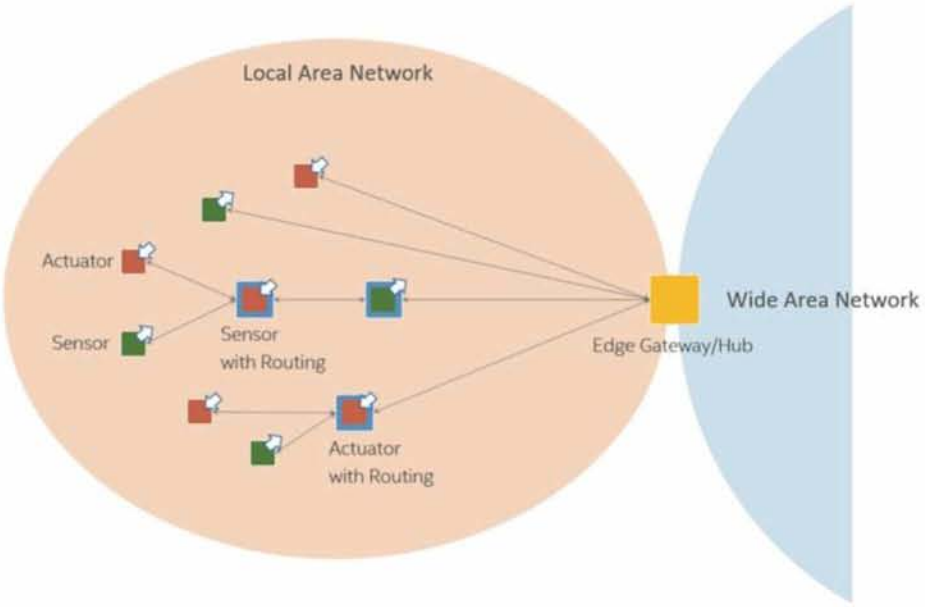
Εικόνα 3.2 - ΑρχιτεκτονικήThree-TierTopology

3.2 Άκρη διαμεσολαβημένης πύλης δικτύου

Όπως αναφέραμε και παραπάνω, όταν συνήθως έχουμε πολλάδεδομένα από διαφορετικά σημεία μπορεί να σχηματιστούν κόμβοι, που συλλέγουν ή ενώνουν τις πληροφορίες, πριν τις στείλουν στην πύλη δικτύου ή σε κάποιον άλλον κόμβο. Όλοι αυτοί οι κόμβοι ενώνονται σε μια συσκευή, που ονομάζουμε άκρη διαμεσολαβημένης πύλης δικτύου

(gateway-mediatededge).Ο σκοπός τηςδιαμεσολαβημένης πύλης δικτύου είναι να συγκεντρώσει όλους τους κόμβους πληροφοριών και να παρέχει μια άκρο προς άκρο (pointtopoint) σύνδεση. Ένα από τα πολλά πλεονεκτήματα από την χρήση αυτής της συσκευής, είναι ηαπλοποίηση τηςπολυπλοκότηταςτου δικτύου. Έτσι, παρέχεται ένα μοναδικό σημείο στην είσοδο/έξοδο της βαθμίδας του άκρου, που είναι ιδανικό για την διαχείριση, επεξεργασία και ανάλυση δεδομένων.

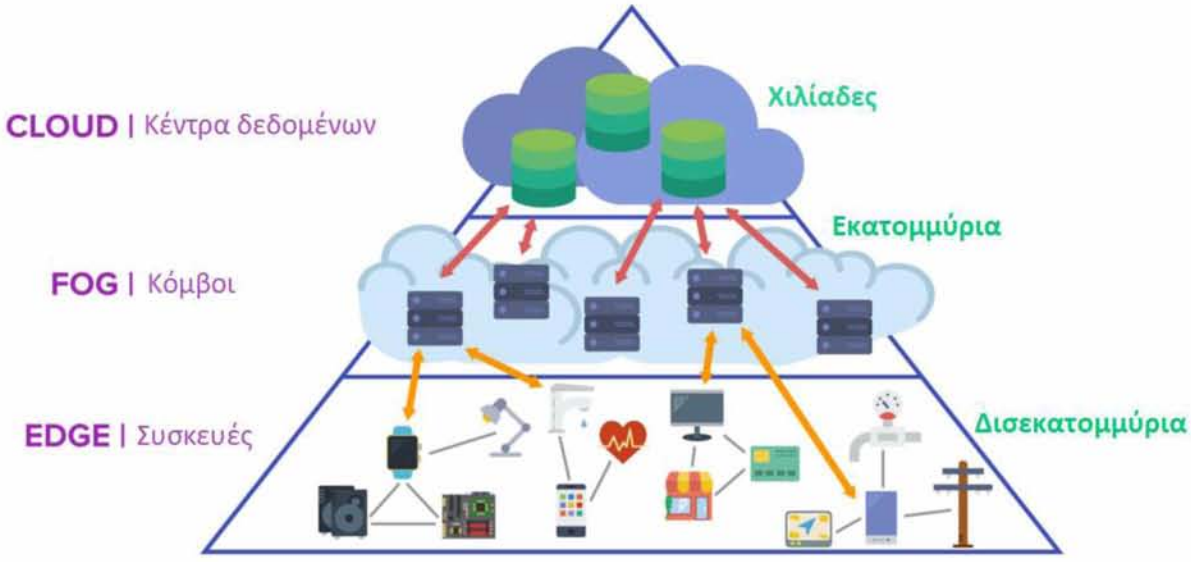
Η έννοια του να υπάρχει ένα ενδιάμεσο σημείο, το οποίο συλλέγει τις πληροφορίες και προαιρετικά κάνει κάποια ανάλυση ή επεξεργασία, πριν τις στείλει στον τελικό προορισμό είναι αυτό, που ονομάζουμε υπολογισμός σε ομίχλη (fogcomputing). Με αυτόν τον τρόπο μειώνουμε την καθυστέρηση της μεταφοράς δεδομένων και δεν επιβαρύνουμε το νέφος με όλες τις πληροφορίες που στέλνουν οι συσκευές. Η καθυστέρηση της μεταφοράς των δεδομένων μειώνεται, επειδή το ενδιάμεσο σημείο επιλέγει να στείλει χρήσιμες μόνο προς το νέφος πληροφορίες. Το αποτέλεσμα της ανάλυσης και επεξεργασίας των δεδομένων που γίνεται εκεί,ενδεχομένως να είναι μικρότερο από όλες τις πληροφορίες που διαφορετικά θα στέλναμε.



Εικόνα3.3 – Gateway-mediatedEdge

3.3 Υπολογισμός σε ομίχλη

Ο υπολογισμός σε ομίχλη (fogcomputing) είναι ένας όρος, που έδωσε η Cisco.Οσυγκεκριμένος αναφέρεται στην επέκταση και πολύ πιθανόν στο κοντινό μέλλον να είναι η επόμενη γενιά του υπολογισμού σε νέφος (cloud computing). Στον υπολογισμό σε ομίχλη γίνεται η βέλτιστη κατανομή των δεδομένων, που είναι για υπολογισμό ή για αποθήκευση σε δεδομένα, που θα μείνουν κάπου τοπικά και σε αυτά, που θα μεταφερθούν στο νέφος. Αυτό γίνεται για να μειωθεί ο τεράστιος όγκος δεδομένων και να μειωθεί η καθυστέρηση της μεταφοράς των πληροφοριών που παράγεται από το IoT στο νέφος. Μερικές πληροφορίες πρέπει να αναλυθούν σε πολύ περιορισμένο χρόνο, κατά τον οποίο η διάρκεια της μεταφοράς τους στο νέφος δεν το επιτρέπει. Ο υπολογισμός σε ομίχλη βοηθάει στην αντιμετώπιση αυτού του προβλήματος, αναλύοντας τις πληροφορίες, που απαιτούν άμεση επεξεργασία στην πύλη του δικτύου αντί να στέλνει μεγάλο και ευαίσθητο ως προς τον χρόνο μεταφοράς όγκο δεδομένων στο νέφος.



Εικόνα 3.4 – Πυραμίδα συνδεσμολογίας IoT

3.4Ευφυήςάκρη δικτύου

Εφαρμόζοντας την ιδέα του υπολογισμού σε ομίχλη στην πύλη δικτύου (gateway) έχουμε αυτό, που ονομάζουμε υπολογισμό άκρης (edgecomputing) με αποτέλεσμα η πύλη δικτύου εκτός από το να δρομολογεί πληροφορίες από τους κόμβους συσκευών στο νέφος ή το αντίστροφο μπορεί πλέον να αναλύει και να επεξεργάζεται τοπικά, κάποια δεδομένα. Με

αυτόν τον τρόπο η πλέον ευφυής πύλη δικτύου (IntelligentEdgeGateway) έχει την δυνατότητα να εφαρμόζει και λειτουργίες της βαθμίδας της πλατφόρμας που είδαμε στην παράγραφο 3.1.

Εικόνα 3.5 – Μια ευφυής πύλη δικτύου της Dell

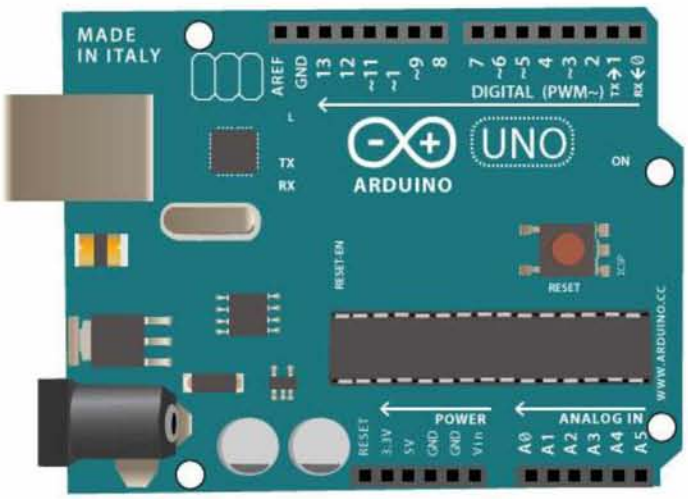
4. Δημιουργώντας ένα IIoT

4.1 Περιγραφή του προϊόντος

Στο χώρο της βιομηχανίας στις παραγωγικές διαδικασίες μπορεί να χρησιμοποιηθούν λέβητες παραγωγής ατμού (ή ατμολέβητες), που είναι ειδικά κατασκευασμένοι για την ατμοποίηση του νερού σε κάποια συγκεκριμένη θερμοκρασία και πίεση. Το ενδεχόμενο να ξεφύγουν αυτές οι μέτρησης έξω από το όριο, είναι κάτι το οποίο πρέπει να αποφεύγεται. Μια λύση, για να ελέγξουμε αυτόν τον κίνδυνο είναι να έχουμε κάποιον μηχανικό, ο οποίος να παρακολουθεί τους λέβητες και στην περίπτωση που συμβεί κάτι να επέμβει. Ένα IIoT λοιπόν θα μπορούσε να είναι ο λέβητας. Αυτό θα ξεκινήσει με την σύνδεση των αισθητήρων του στο διαδίκτυο και μετά από συγκεκριμένα βήματα ο μηχανικός με κάποιον ειδικό εξοπλισμό στα μάτια του θα μπορεί να βλέπει όλες τις χρήσιμες πληροφορίες, που αφορούν τον λέβητα σε εικονική και επαυξημένη πραγματικότητα. Στην συνέχεια των επόμενων κεφαλαίων θα εφαρμόσουμε την αρχιτεκτονική Three-Tier Topology σε έναν βραστήρα νερού, που θα αντιπροσωπεύει τον λέβητα ενός εργοστασίου.

Η μετατροπή του βραστήρα σε ένα προϊόν IIoT θα χρειαστεί εκτός από το κατάλληλο λογισμικό (software), το υλικό (hardware) κομμάτι.

4.2 Arduino και Internet of Things



Εικόνα 5- ArduinoUnoR3

Το ArduinoUNO(εικόνα 2) είναι ένας ανοιχτού κώδικα μικροελεγκτής, βασισμένος στο μικροεπεξεργαστή MicrochipATmega328P και είναι σχεδιασμένος από την Arduino.cc. Η πλακέτα είναι ενσωματωμένη με 14 ψηφιακές και 6 αναλογικές εισόδους/εξόδους, που αλληλοεπιδρούν με διάφορα άλλα κυκλώματα, αισθητήρες, επεκτάσεις κ.τ.λ.. Προγραμματίζεται με το ArduinoIDE (IntegratedDevelopmentEnvironment) με την χρήση ενός Β τύπου USB καλωδίου, το οποίο λειτουργεί και σαν τροφοδοσία. Επιπλέον, τροφοδοσία μπορεί να δεχτεί και από μια εξωτερική μπαταρία των 9 volt. Παρόμοιες πλακέτες είναι το ArduinoNano και Leonardo. Το τσιπάκι (ATmega328) πάνω στο ArduinoUno είναι ήδη προγραμματισμένο με έναν bootloader, που του επιτρέπει να φορτώνει νέο κώδικα χωρίς την χρήση κάποιου άλλου υλικού (hardware). Η επικοινωνία γίνεται μέσω του STK500 πρωτοκόλλου. Το ArduinoUNO υπάρχει στην αγορά με αξία λιγότερη από 25 ευρώ και με κλώνους λιγότερο από 5 ευρώ. ΤοIoTσυνδέει συσκευές με το Internet. Μπορούμε να συνδέσουμε τον βραστήρα με το Arduino,όμως,για να επικοινωνήσουμε με το διαδίκτυο το Arduinoαπό μόνο του δεν αρκεί, αφού όπως είδαμε το Arduinoείναι απλά ένας μικροελεγκτής. Η επικοινωνία με το διαδίκτυο θα επιτευχθεί, αν συνδέσουμε στο Arduinoένα τσιπάκι, που ονομάζεται ESP8266-01.

Εικόνα 6- ToArduinoIDE

4.3 ESP8266-01

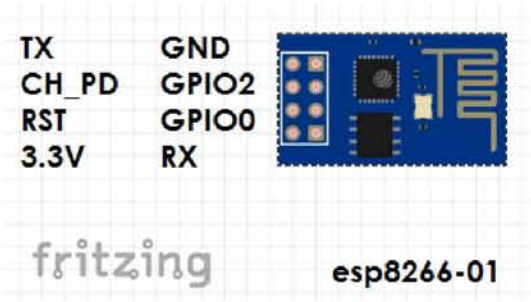
Εικόνα 7 - ESP8266-01

ToESP8266-01 (εικόνα 4) είναιέναχαμηλούκόστουςchipμεολοκληρωμένηTCP/IPστοίβακαι μικροελεγκτή. Κατασκευάστηκε από την Ai-Thinkerκαι επιτρέπει στους μικροελεγκτές να συνδέονται στο Wi-Fiδίκτυο και να εκτελεί απλές TCP/IPσυνδέσεις με εντολές Hayes-style. Το ESP8266-01 προσφέρει μια ολοκληρωμένη και αυτοδύναμη λύσηστην σύνδεση του βραστηρα στο δίκτυο μέσωWi-Fi.

Στην εικόνα 5 παρατηρούμε μια λίστα με τις εντολές, που μας επιτρέπουν να προγραμματίσουμε το τσιπάκι.

Function	***AT Command***	***Response***
Working	AT	OK
Restart	AT+RST	OK [System Ready, Vendor:www.ai-thinker.com]
Firmware version	AT+GMR	AT+GMR 0018000902 OK
List Access Points	AT+CWLAP	AT+CWLAP +CWLAP:(4,"RochefortSurLac",-38,"70:62:b8:6f:6d:58",1) +CWLAP:(4,"LiliPad2.4",-83,"f8:7b:8c:1e:7c:6d",1) OK
Join Access Point	AT+CWJAP? AT+CWJAP="SSID","Password"	Query AT+CWJAP? +CWJAP:"RochefortSurLac" OK
Quit Access Point	AT+CWQAP=? AT+CWQAP	Query OK
Get IP Address	AT+CIFSR	AT+CIFSR 192.168.0.105 OK
WiFi Mode	AT+CWMODE? AT+CWMODE=1 AT+CWMODE=2 AT+CWMODE=3	Query STA AP BOTH
Set up TCP or UDP connection	AT+CIPSTART=? (CIPMUX=0) AT+CIPSTART = <type>,<addr>,<port> (CIPMUX=1) AT+CIPSTART= <id><type>,<addr>, <port>	Query id = 0-4, type = TCP/UDP, addr = IP address, port= port
TCP/IP Connection Status	AT+CIPSTATUS	AT+CIPSTATUS? no this fun
Send TCP/IP data	(CIPMUX=0) AT+CIPSEND=<length>; (CIPMUX=1) AT+CIPSEND= <id>,<length>	
Close TCP / UDP connection	AT+CIPCLOSE=<id> or AT+CIPCLOSE	

Εικόνα 8 - Εντολές ESP8266

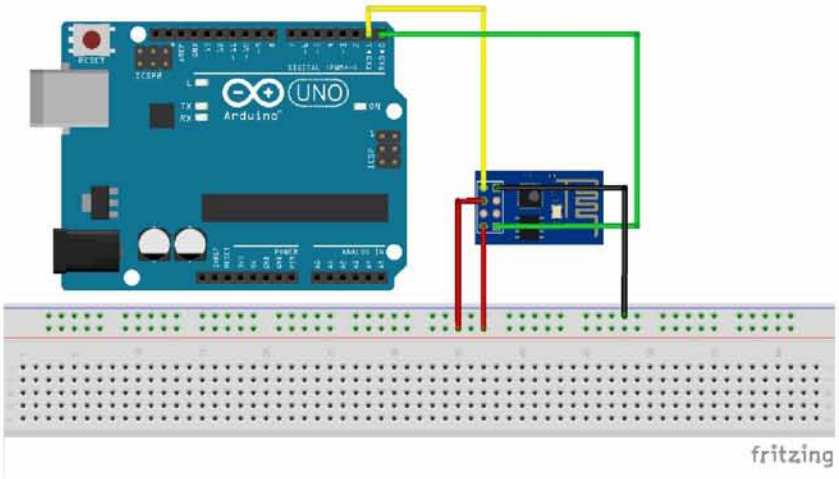


Εικόνα 9 - Είσοδοι/Εξοδοι του ESP8266-01

Για να προγραμματίσουμε λοιπόν το ESP8266-01 πρέπει να το συνδέσουμε με το Arduino. Στην εικόνα 7 βλέπουμε πως γίνεται η συνδεσμολογία του ArduinoUNO με το ESP8266-01. Η εκτέλεση των εντολών θα γίνει μέσω του ArduinoIDE, αφού πρώτα φορτώσουμε τον παρακάτω άδειο (sketch) κώδικα.

```
1
2void setup(){
3// put your setup code here, to run once:
4}
5
6void loop(){
7// put your main code here, to run repeatedly:
8}
```

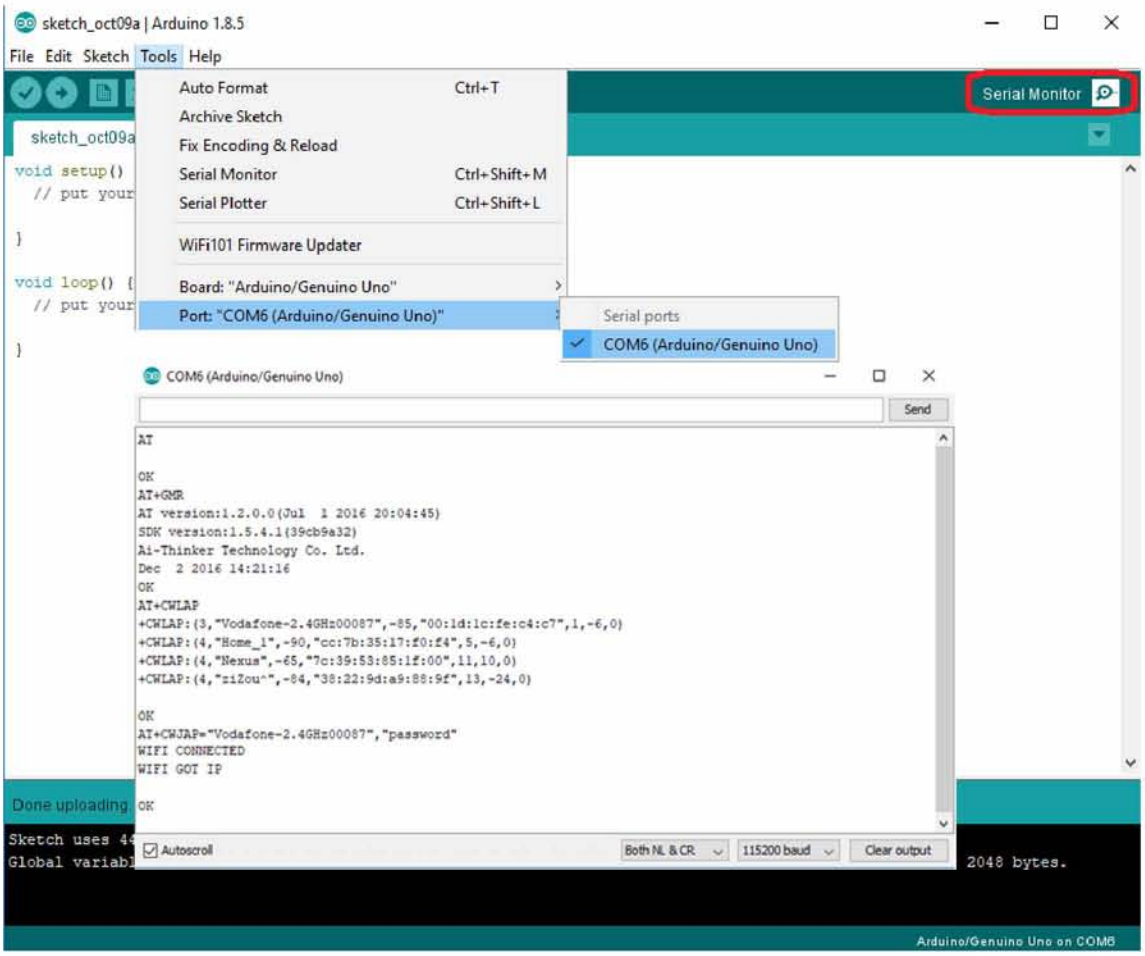
Μόλις ολοκληρωθεί η φόρτωση μπορούμε να ανοίξουμε ένα τερματικό και να συνδεθούμε σειριακά στην θύρα COM του Arduino. ΤοESP8266 επικοινωνεί με baudrate 115200.



Εικόνα 10 - Συνδεσμολογία Arduinoμε ESP8266-01

Το ArduinoIDE διαθέτει δικό του τερματικό, για να εκμεταλλευθούμε, όπως παρατηρούμε στην εικόνα 8. Στο πλαίσιο εισαγωγής κειμένου τώρα πια μπορούμε να εκτελέσουμε τις εντολές (εικόνα 5) του ESP8266-01 και από κάτω θα εμφανίζεται η απάντηση της εντολής.

Στο επόμενο κεφάλαιο θα δούμε πως θα χρησιμοποιηθεί το ESP8266-01, για να στέλνουμε πληροφορίες, που θα συλλέγουμε από τον βραστήρα στο Ίντερνετ.



Εικόνα 11-Τοτερματικό του ArduinoIDE

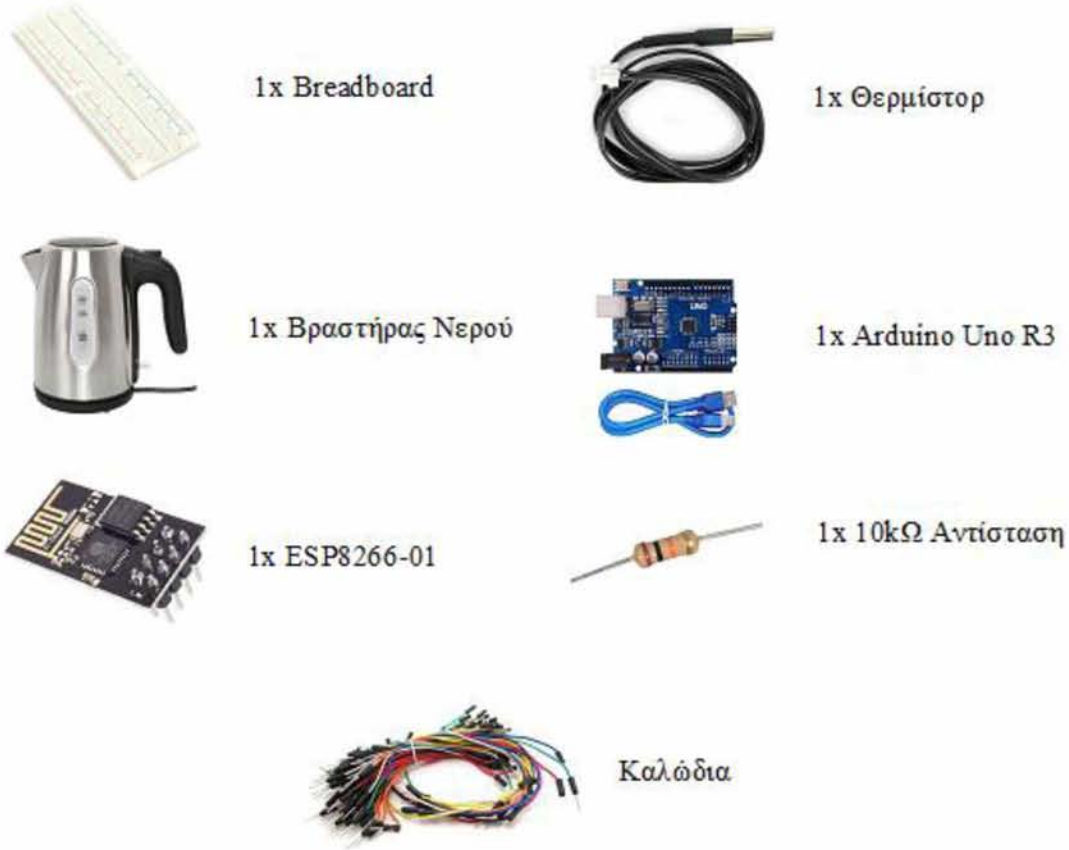
Το γεγονός:Ένα Arduinoαπό μόνο του δεν έχει καμιά σχέση με το IoT. ΤοArduinoείναιέναμικροελεγκτής. Βοηθάειστηνεπικοινωνίαμετονφυσικόκόσμο. Αλλάμπορούμενα τροποποιήσουμε τοArduino,ώστενα προκύψει έναIoTπροϊόνπροσθέτοντας ένα *ESP8266*.

[Απαντήσεις Quora]

5. Έξυπνος Βραστήρας

5.1 Υλικά

Στην εικόνα 9 παρατηρούμε τα υλικά που θα χρειαστούμε, για να υλοποιήσουμε τον έξυπνο βραστήρα. Όλα τα υλικά είναι συνηθισμένα και γνωστά στην αγορά. Ο βραστήρας λοιπόν, που θα φτιάξουμε θα μοιράζεται την πληροφορία της θερμοκρασίας του νερού στο Ίντερνετ. Πρέπει να τονίσουμε εδώ ότι ο θερμίστορ επιβάλλεται να είναι αδιάβροχος, αφού θα τον βάλουμε στο εσωτερικό του βραστήρα μέσα στο νερό.



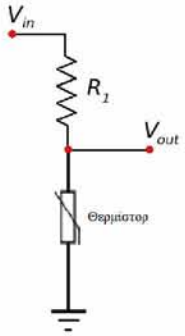
Εικόνα 12- Υλικά του Έξυπνου Βραστήρα

5.2 Ηsυναρμολόγηση του βραστήρα.

5.2.1 Ο θερμίστορ

Ο θερμίστορ έχει μια αντίσταση, που αλλάζει ανάλογα με την θερμοκρασία. Υπάρχουν δύο ειδών, αυτοί, που η αντίσταση τους αυξάνεται με την άνοδο της θερμοκρασίας (PTC) και αυτοί, που η αντίσταση τους μειώνεται με την αύξηση της θερμοκρασία (NTC). Οι PTCθερμίστορ συνήθως συνδέονται σε σειρά στα κυκλώματα, σαν αυτοεπαναφερόμενες ασφάλειες και οι NTCχρησιμοποιούνται συχνά, σαν αισθητήρες θερμοκρασίας. Ο θερμίστορ, που θα χρησιμοποιούμε είναι τύπου NTC. Για να λάβουμε την τιμή της αντίστασης λοιπόν, αρκεί να τον συνδέσουμε μαζί με μια αντίσταση των 10kΩ σε μορφή κυκλώματος διαιρέτη τάσης. Η τιμή, που παράγεται από τον διαιρέτη τάσης λαμβάνεται από το Arduinoστην πρώτη αναλογική είσοδο (A0) (εικόνα 12).

Εικόνα 13 - Διάγραμμα NTC θερμίστορ

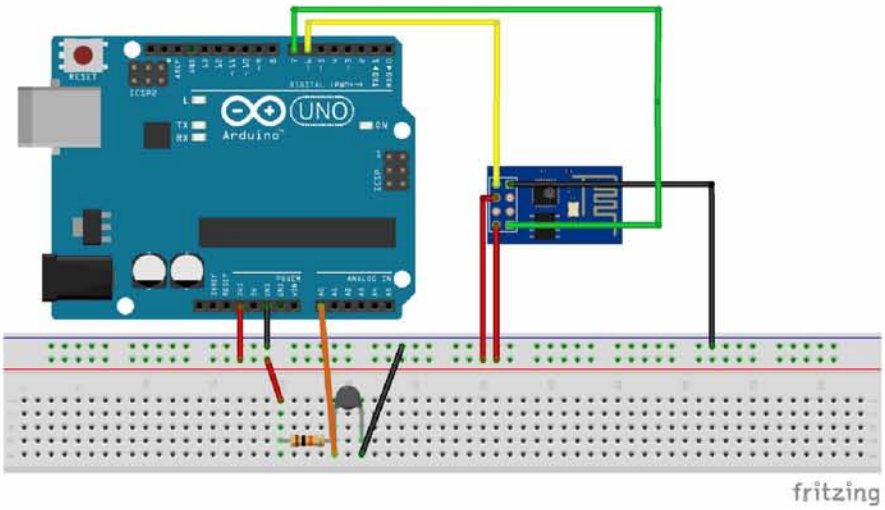


Εικόνα 14- Διαιρέτης Τάσης

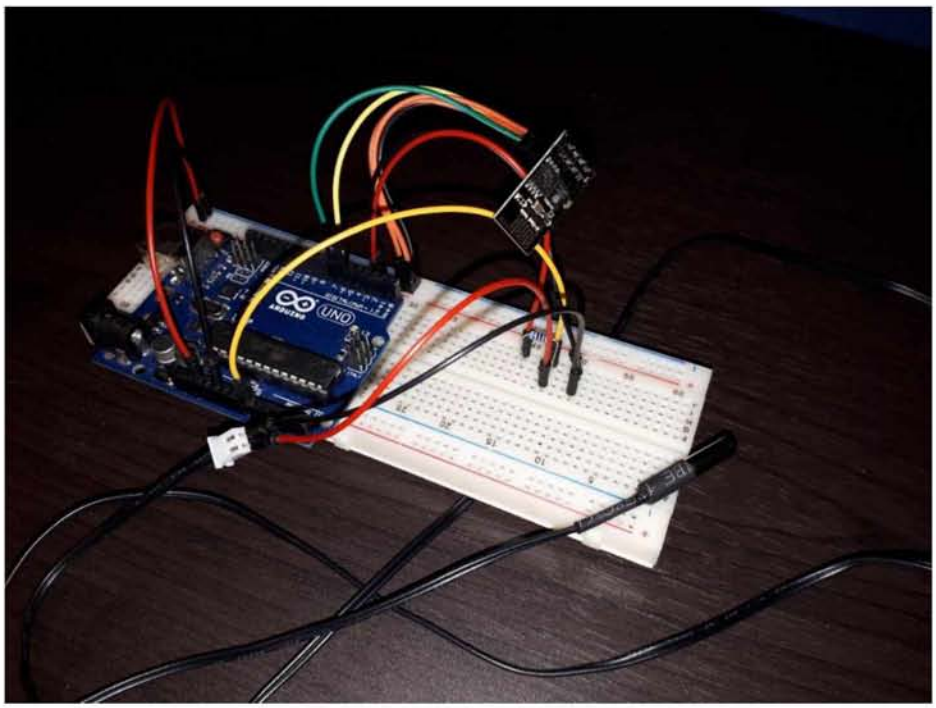
5.2.2 Το ESP8266-01

Στο προηγούμενο κεφάλαιο είδαμε πως μπορούμε να εκτελέσουμε από το τερματικό εντολές στο ESP8266-01 μέσο του Arduino. Αυτό έγινε μέσο της σειριακής επικοινωνίας. Αν παρατηρήσουμε την εικόνα 7 θα δούμε ότι έχουμε συνδέσει τα RXκαι TXτου ESP8266-01 στις ψηφιακές θύρες εισόδου/εξόδου (IO pins) 0 και 1του Arduino. Αυτές οι θύρες είναι ειδικού σκοπού για σειριακή επικοινωνία απευθείας συνδεδεμένες με το FTDI USB-to-TTL, όπου το Arduino λαμβάνει τον νέο κώδικα αλλά πραγματοποιείκαι την σύνδεση/επικοινωνία

με τον ηλεκτρονικό υπολογιστή. Συνδέοντας το ESP8266-01 σε αυτές τις θύρες μας επέτρεπε να επικοινωνούμε από τον υπολογιστή κατευθείαν με αυτό. Τώρα όμως επιθυμούμε το Arduinova αναλάβει την επικοινωνία με το ESP8266-01.Έτσι, λοιπόν, συνδέουμε το RXκαι TX στις θύρες 7 και 6 αντίστοιχα όπως βλέπουμε στην εικόνα 12.



Εικόνα 15 - Συνδεσμολογία Arduino



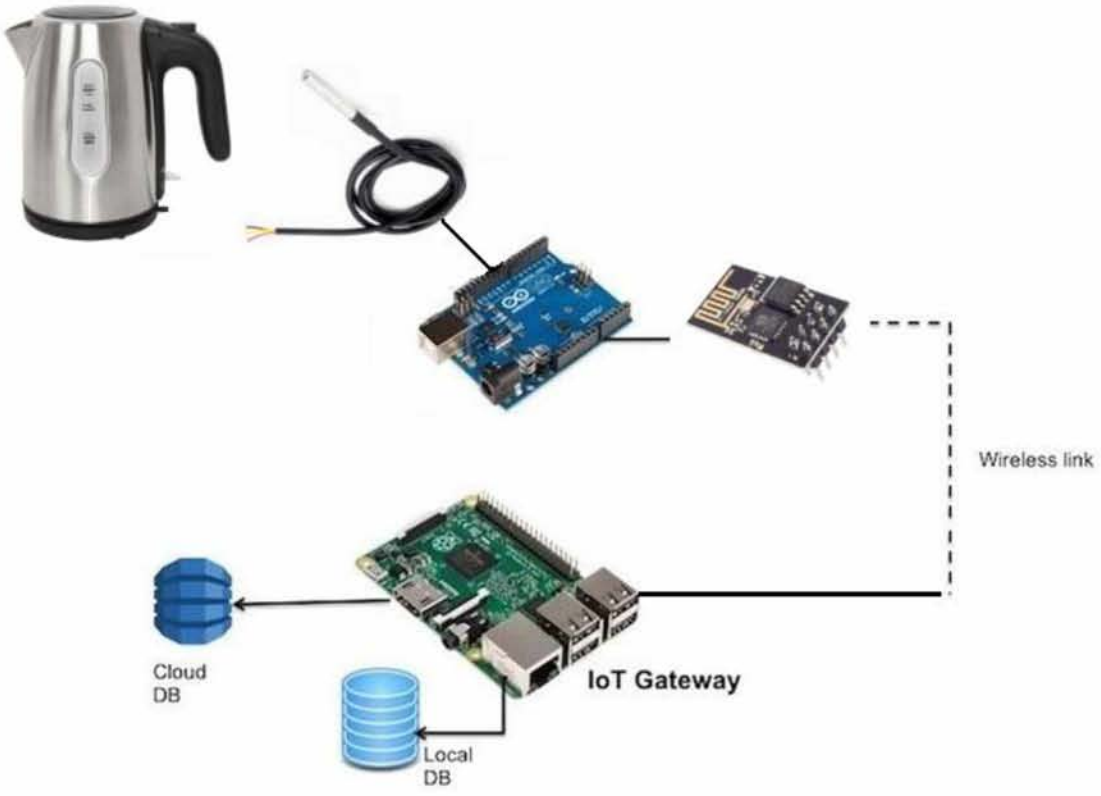
Εικόνα16 - ΣυνδεσμολογίαArduino

6. Η πύλη Δικτύου και η Επικοινωνία με το Νέφος

6.1 Πύλη Δικτύου (gateway)

6.1.1 Ορισμός

Μια IoT πύλη δικτύου (gateway) είναι μια φυσική συσκευή ή ένα λογισμικό πρόγραμμα,που συνδέει αισθητήρες, ελεγκτές και συσκευές με το νέφος (cloud). Όπως αναφέραμε και στο τρίτο κεφάλαιο αυτής της εργασίας η πύλη δικτύου μπορεί να επεξεργάζεται και να αποθηκεύει δεδομένα τοπικά. Για τον σκοπό αυτής της εργασίας θα χρησιμοποιήσουμε ένα RaspberryPiτο οποίο θα υποδύεται τον ρόλο της πύλης δικτύου.

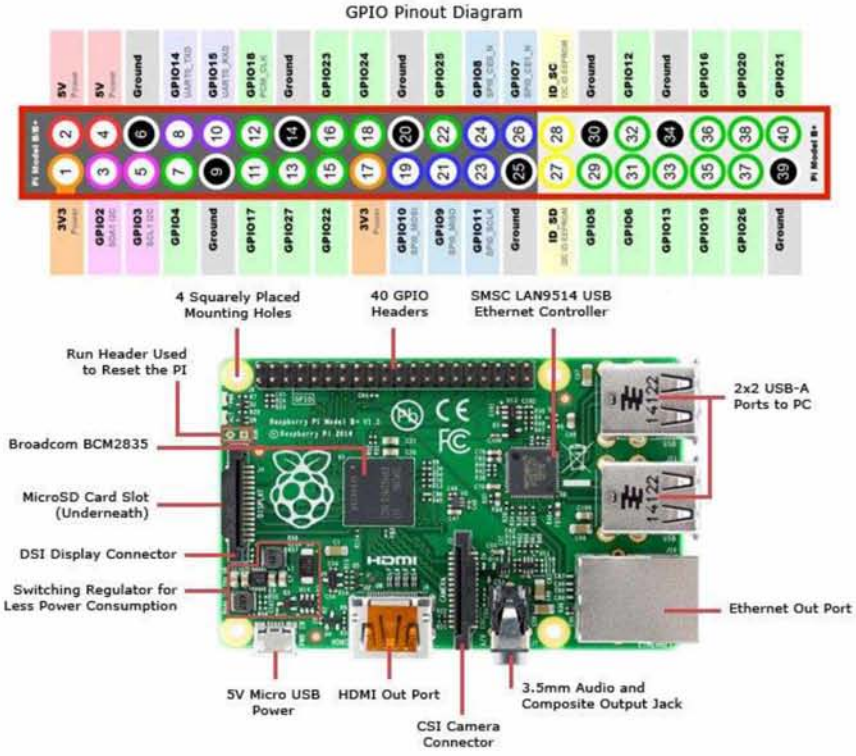


Εικόνα 6.16 - Προσθήκη του RaspberryPi σαν gateway

6.1.2 Raspberry Pi

ToRaspberryPiείναι ένας χαμηλού κόστους ηλεκτρονικός υπολογιστής με πολύ μικρή κατανάλωση ρεύματος στο μέγεθος μιας πιστωτικής κάρτας. Έχει πολύ λιγότερες δυνατότητες από έναν σύγχρονο υπολογιστή, όμως τρέχει ένα ολοκληρωμένο σύστημα

Linux. Διαθέτει γενικού σκοπού θύρες εισόδου/εξόδου (GPIO),που επιτρέπουν την επικοινωνία του RaspberryPiμε κάποιο άλλο κύκλωμα ή συσκευή.Δημιουργήθηκε αρχικά για εκπαιδευτικούς σκοπούς, αλλά γρήγορα αποδείχτηκε ένα ιδανικό εργαλείο για την σχεδίαση και υλοποίηση IoT προϊόντων.



Εικόνα 6.17 – Χαρακτηριστικά του RaspberryPi 3

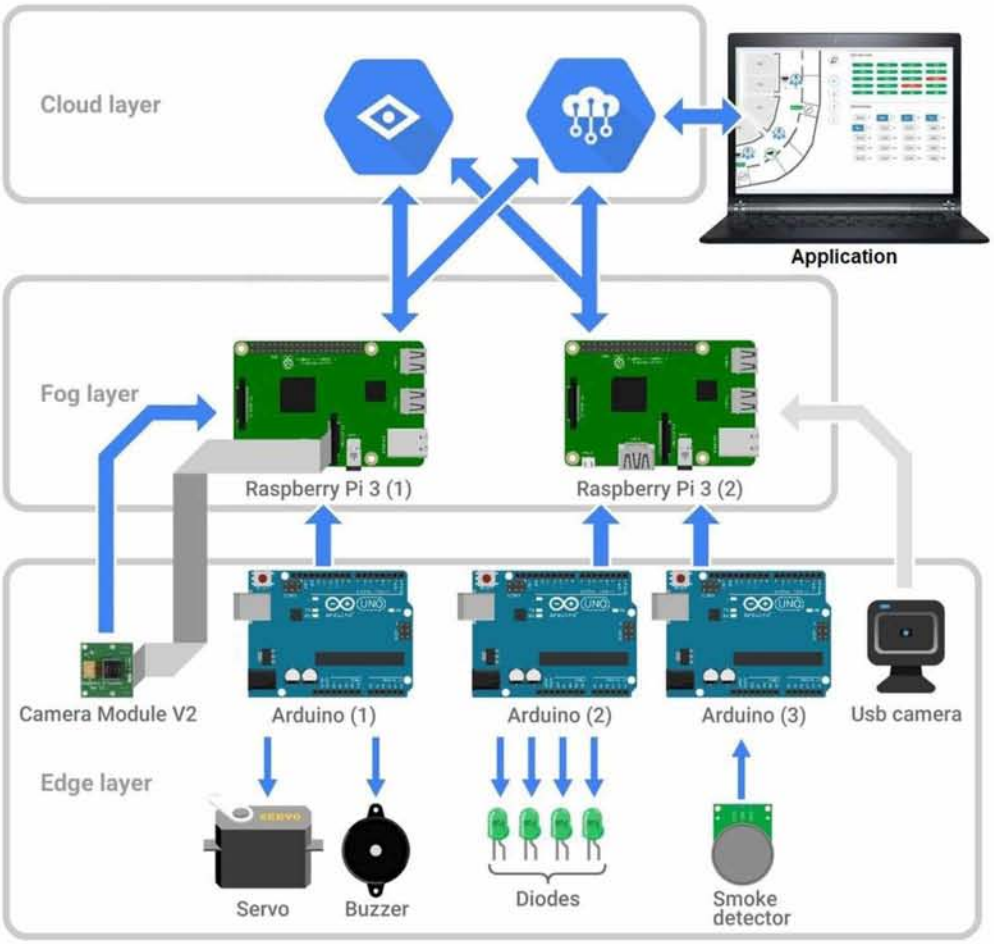
6.1.3 Σχεδιάζοντας την πύλη δικτύου του βραστήρα.

Η πύλη δικτύου, που σχεδιάσαμε για τον βραστήρα διαθέτει έναν LAMPεξυπηρετητή(Linux, Apache, MySQL, PHP), ο οποίος θα ικανοποιεί της αιτήσεις από το Arduinoκαι θα διαθέτει μια βάση δεδομένων όπου εκεί συλλέγονται τα δεδομένα, που εισέρχονται. Όπως αναφέραμε και στα προηγούμενα κεφάλαια μια πύλη δικτύου μπορεί να ξεχωρίζει τα σημαντικά δεδομένα από τα ασήμαντα, για να μην επιβαρύνει το νέφος με τεράστιο όγκο δεδομένων. Ακόμη, αναφέραμε πως υπάρχουν ευφυής πύλες δικτύου, που μπορούν να επεξεργάζονται κάποια δεδομένα τοπικά, δηλαδή στο RaspberryPiκαι στην συνέχεια να στέλνουν το αποτέλεσμα. Έτσι, λοιπόν το Arduinoστέλνει συνεχώς στο RaspberryPiτην τρέχουσα θερμοκρασία του βραστήρα. Το RaspberryPiαποθηκεύει στην βάση δεδομένων του, την θερμοκρασία, μόνο, όταν η θερμοκρασία, που μόλις δέχτηκε είναι

διαφορετική από την τελευταία, που αποθήκευσε γλιτώνοντας μεγάλο αχρείαστο όγκο δεδομένων.

Μια ακόμα λειτουργία, που θα εκτελεί το RaspberryPiείναι να υπολογίζει την μέση θερμοκρασία των τελευταίων τριών ωρών του βραστήρα και μαζί με την τρέχουσα θερμοκρασία να τις στέλνει στο νέφος. Αυτό θα γίνει με την εκτέλεση προγράμματος σε περιβάλλον python, που θα εκτελείται σε ατέρμοντα βρόχο. Οι πληροφορίες αυτές θα στέλνονται, μόνο όταν θα έχουμε νέα θερμοκρασία από το βραστήρα.

Στην εικόνα 6.3 βλέπουμε έναν αριθμό από ArduinoUNOκαι RaspberryPiοργανωμένα όπως περιγράψαμε στο κεφάλαιο 3 με τα Arduinova έχουν τον ρόλο των κόμβων συλλογής πληροφορίας και τα RaspberryPiτον ρόλο της πύλης δικτύου.



Εικόνα 6.3 – Παράδειγμα αρχιτεκτονικής IoT με πολλαπλά Arduinoκαι RaspberryPi

6.2 Επικοινωνία με το νέφος.

6.2.1 Cloud Computing

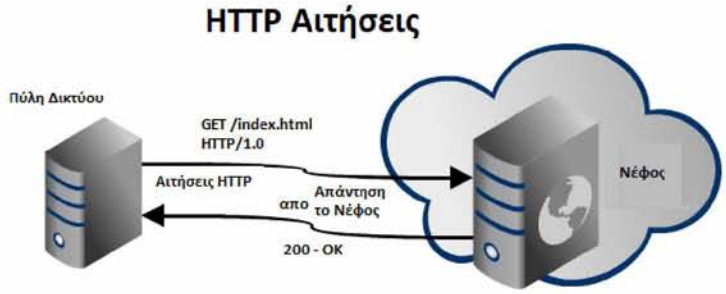
Η υπολογιστική νέφους (cloudcomputing) είναι μια τεχνολογία υποδομής, που παρέχει υπηρεσίες μέσω του διαδικτύου. Οι υπηρεσίες που προσφέρονται στην υπολογιστική νέφους επιτρέπουν σε ιδιώτες και επιχειρήσεις να χρησιμοποιήσουν υλικό και λογισμικό, που βρίσκονται σε απομακρυσμένες περιοχές.

Στην εργασία για τον ρόλο του νέφους χρησιμοποιήσαμε έναν εξυπηρετητή apache,ο οποίος με κατάλληλα phpαρχεία επεξεργάζεται τις πληροφορίες που του στέλνουμε από την πύλη δικτύου και στην συνέχεια τις αποθηκεύει σε μια MySQLβάση δεδομένων.

6.2.2 Το Πρωτόκολλο Μεταφοράς Υπερκειμένου

Το Πρωτόκολλο Μεταφοράς Υπερκειμένου (HyperText Transfer Protocol) ή αλλιώς HTTPείναι το πρωτόκολλο επικοινωνίας που χρησιμοποιήσαμε για την επικοινωνία της πύλης δικτύου με το νέφος. Το HTTPείναι επίσης το κύριο πρωτόκολλο που χρησιμοποιείται στους φυλλομετρητές του Παγκοσμίου Ιστού, για να μεταφέρει δεδομένα ανάμεσα σε έναν διακομιστή (server) και έναν πελάτη (client) μέσω κάποιον αιτήσεων που ονομάζονται μέθοδοι. Οι πιο βασικέςμέθοδοι αίτησης του πρωτοκόλλου HTTP είναι οι POSTκαι GET.

Η μέθοδος GETχρησιμοποιείται για την αίτηση αποστολής κάποιου αρχείου από την εξυπηρετητή πίσω στον πελάτη. Η μέθοδος POST από την άλλη χρησιμοποιείται για την αποστολή δεδομένων στον εξυπηρετητή όπως για παράδειγμα ένα όνομα και ο κωδικός πρόσβασης ενός χρήστη που αποστέλλονται στον εξυπηρετητή μέσο μια φόρμας.



7. Ανάπτυξη της Εφαρμογής

7.1 Εργαλεία

7.1.1 Unity

Η ανάπτυξη της εφαρμογής μας θα γίνει με την χρήση της διάσημης μηχανής Unity, η οποία έχει κατασκευαστεί κυρίως για την ανάπτυξη ηλεκτρονικών παιχνιδιών (videogames), αλλά οι δυνατότητες της δεν σταματάνε εκεί. Για παράδειγμα, υποστηρίζει και την ανάπτυξη εφαρμογών εικονικής πραγματικότητας και πλέον από την έκδοση 2017.2 η Unity έχει ενσωματώσει τοκιτ ανάπτυξης λογισμικούVuforia,το οποία μας επιτρέπει να αναπτύξουμε εφαρμογές σε επαυξημένη πραγματικότητα (AugmentedReality).

7.1.2 Vuforia

ToVuforia είναι ένα κιτ ανάπτυξης λογισμικού (SDK), το οποία μας δίνει την δυνατότητα να δημιουργούμε εφαρμογές που χρησιμοποιούν επαυξημένη πραγματικότητα. Χρησιμοποιεί την τεχνολογία της μηχανικής όρασης (ComputerVision), η οποία με κατάλληλους αλγορίθμους αναπαράγει την αίσθηση της όρασης,αναγνωρίζοντας εικόνες ή τρισδιάστατα αντικείμενα σε πραγματικό χρόνο. Η εμπορική χρήση τουVuforiaαπαιτεί την αγορά άδειας παρόλα αυτά είναι δωρεάν για την ανάπτυξη πρωτότυπα εφαρμογών. Ακόμα ένα χαρακτηριστικό της είναι ότιπροσφέρει διεπαφές στις γλώσσες προγραμματισμού C++ και Java και συνεργάζεται με την μηχανή Unity.

7.2 Περιγραφή της εφαρμογής

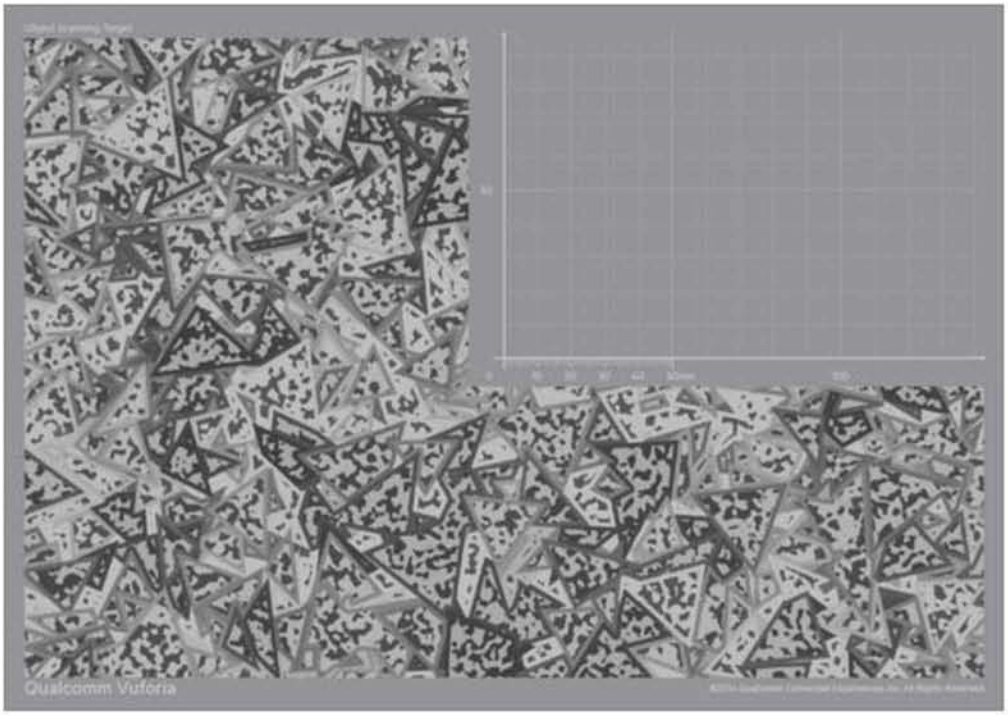
Η λογικής της εφαρμογής μας είναι σχετικά απλή. Θέλουμε να αναπτύξουμε μια εφαρμογή για κινητό Android, το οποία στην συνέχεια θα εφαρμόζεταιπάνω σε ένα VRheadset. Μέσω αυτού του headset θα χρήστης θα βλέπει μέσα από την κάμερα του κινητού τον πραγματικό κόσμο με την χρήση της εικονικής πραγματικότητας.Στην συνέχεια, όταν θα κοιτάει τον βραστήρα, θα εμφανίζεται σε επαυξημένη πραγματικότητα ένα πάνελ, που θα περιέχει ένα εικονικό θερμόμετρο με την τρέχουσα θερμοκρασία του βραστήρα και δίπλα του ένα γράφημα, που θα δείχνει την μέση θερμοκρασία που επικρατούσε μέσα στον

βραστήρα τις τελευταίες τρεις ημέρες. Η μέση θερμοκρασία θα ανανεώνεται κάθε τρεις ώρες.

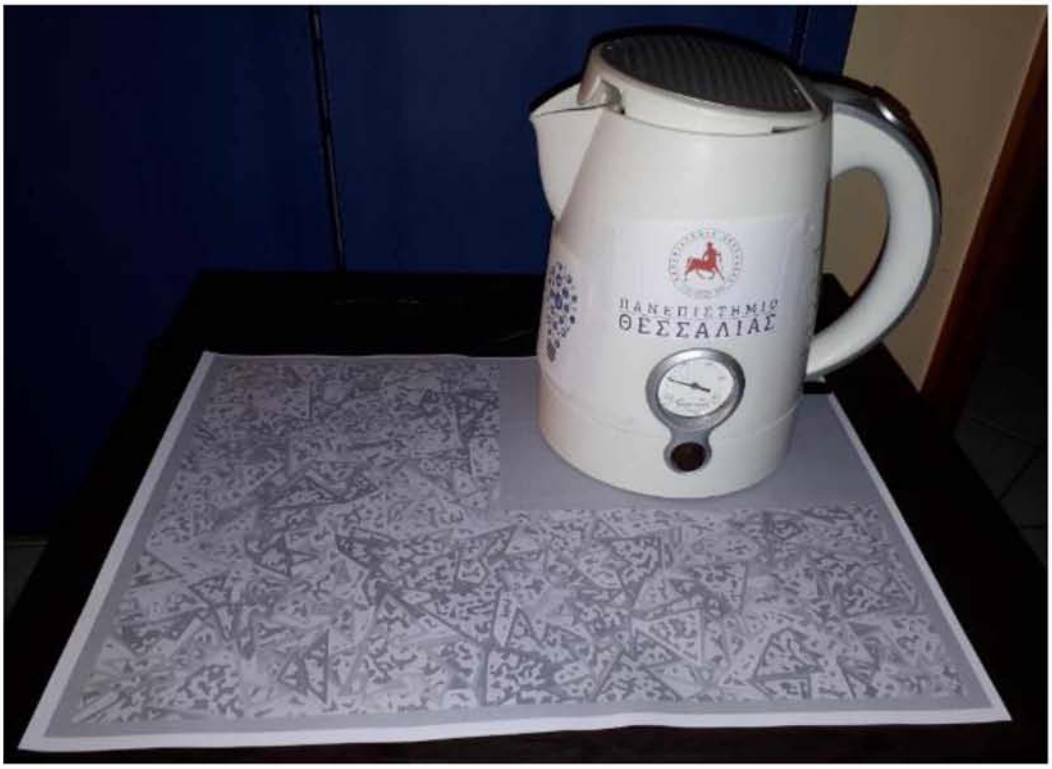
7.3 Περιγραφή ανάπτυξης της εφαρμογής

7.3.1 Vuforia Developer Portal

Όπως αναφέραμε προηγουμένως χρησιμοποιούμε το κιτ ανάπτυξης λογισμικού Vuforia, για να εμφανίζουμε γραφικά στοιχεία σε επαυξημένη πραγματικότητα. Εκείνο που δεν αναφέρθηκε, είναι ότι το Vuforia μας δίνει την δυνατότητα να ανιχνεύουμε εικόνες και αντικείμενα. Στην περίπτωση μας θέλουμε να ανιχνεύουμε τον βραστήρα. Χρησιμοποιώντας την εφαρμογή Android με όνομα VuforiaObjectScanner, που μπορούμε να την βρούμε από την ιστοσελίδα του Vuforia σαρώνουμε με την κάμερα του κινητού το αντικείμενο. Πρώτα όμως πρέπει να εκτυπώσουμε το ObjectScanningTarget που βλέπουμε στην εικόνα 8.1 σε χαρτί A3. Οποιοδήποτε μεγαλύτερο μέγεθος εκτύπωσης μπορεί να δημιουργήσει πρόβλημα στην σάρωση του αντικειμένου. Τοποθετούμε τον βραστήρα στο πάνω δεξιά μέρος του ObjectScanningTarget όπως βλέπουμε στην εικόνα 8.2

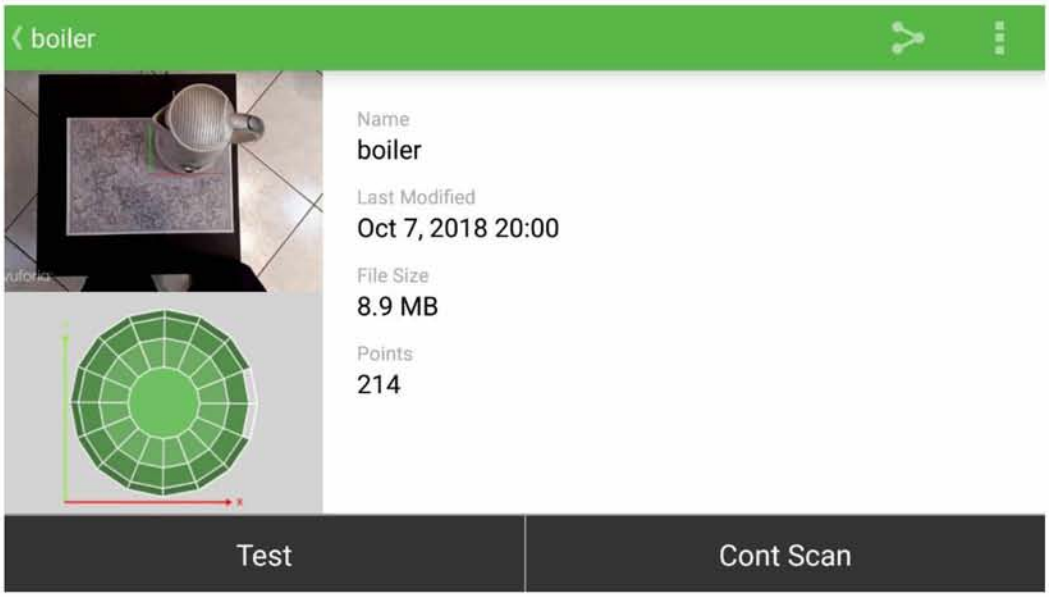


Εικόνα 8.1 - Object Scanning Target



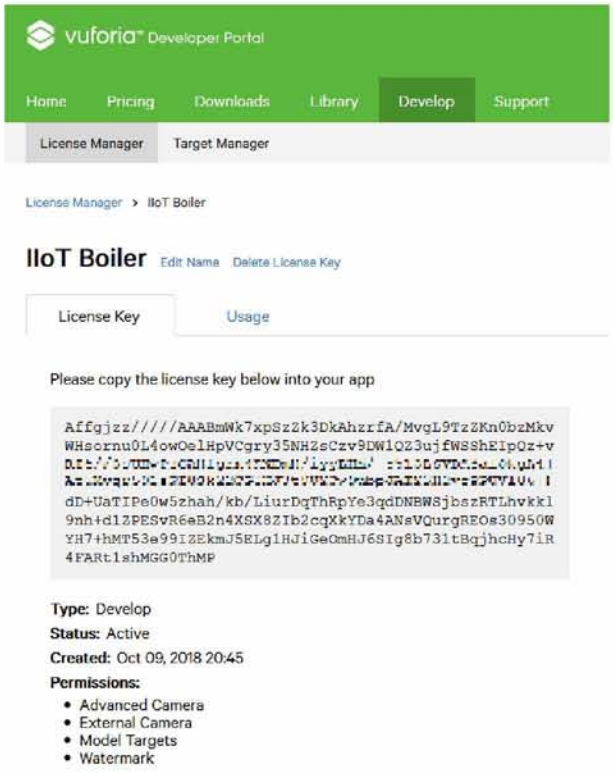
Εικόνα 8.2 – Τοποθέτηση βραστήρα πάνω στο ObjectScanningTarget

Ξεκινώντας την Androidεφαρμογή Scannerακολουθούμε τα βήματα δημιουργία ενός .odaρχείου που ονομάσαμε boiler. Αυτό το αρχείο πρέπει να ανέβει στην ιστοσελίδα του Vuforia,ώστε να παραχθούν τα απαραίτητα αρχεία που θα χρησιμοποιήσουμε στην μηχανή Unity.



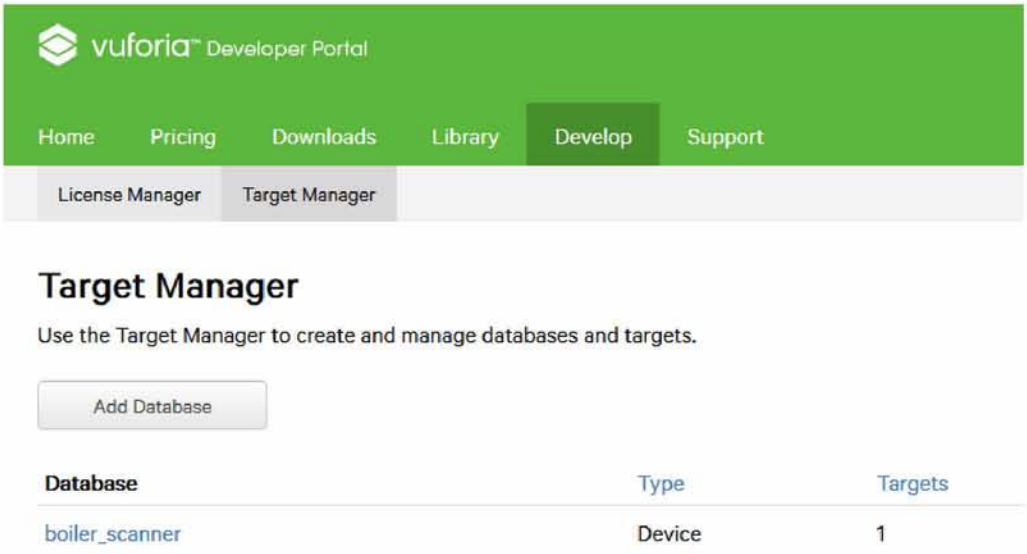
Εικόνα 8.3 – Η σάρωση του βραστήρα

Ανοίγοντας έναν περιηγητή ιστού επισκεπτόμαστε την ιστοσελίδα τουVuforiaκαι επιλέγουμε να μεταβούμε στο DeveloperPortal. Πριν ανεβάσουμε το αρχείο boiler.odπρέπει να δημιουργήσουμε μια δωρεάν άδεια/κλειδί, αφού πρώτα δώσουμε ένα όνομα για την εφαρμογή μας. Στην εικόνα 8.4βλέπουμε την άδεια για την εφαρμογή μας με όνομα IIoT Boiler.

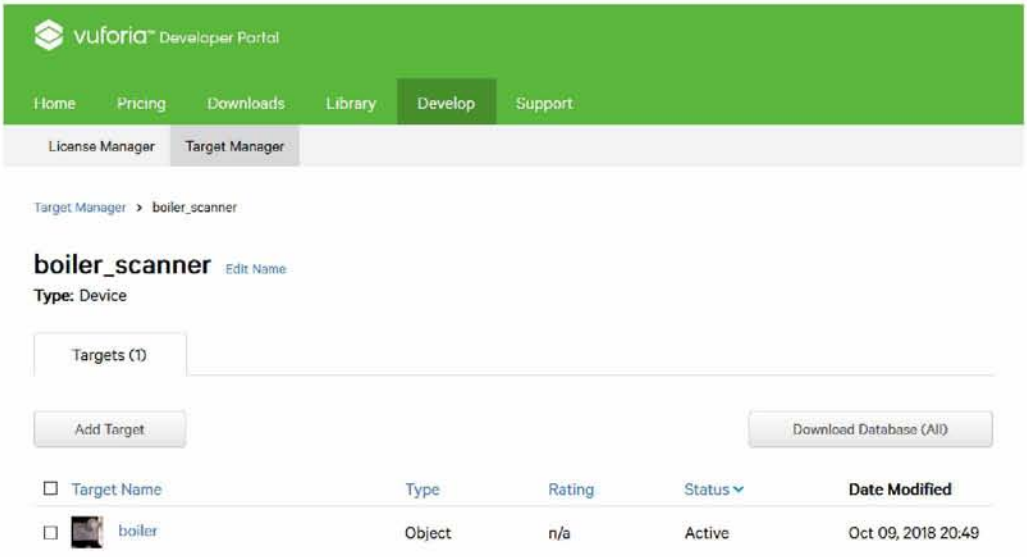


Εικόνα 8.4 Το κλειδί για την εφαρμογή IIoT Boiler

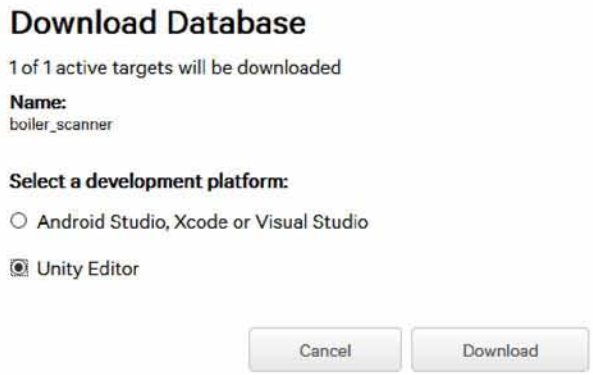
Μεταβαίνοντας στην σελίδα TargetManagerπροσθέτουμε μια βάση δεδομένων όπου θα προσθέτουμε τις εικόνες και αντικείμενα που πρέπει να εντοπιστούν.Δίνουμε το όνομα **boiler_scanner**για την βάση δεδομένων και πατώντας **AddTarget**επιλέγουμε **3DObject**δίνοντας την τοποθεσία του αρχείου boiler.od,που σαρώσαμε με την εφαρμογή Scanner. Μόλις ολοκληρωθεί η μεταφόρτωση του αρχείου κατεβάζουμε την βάση δεδομένων πατώντας το **DownloadDatabase (All)** επιλέγοντας για πλατφόρματο **UnityEditor**.



Εικόνα 8.5 – Η βάση δεδομένων boiler_scanner



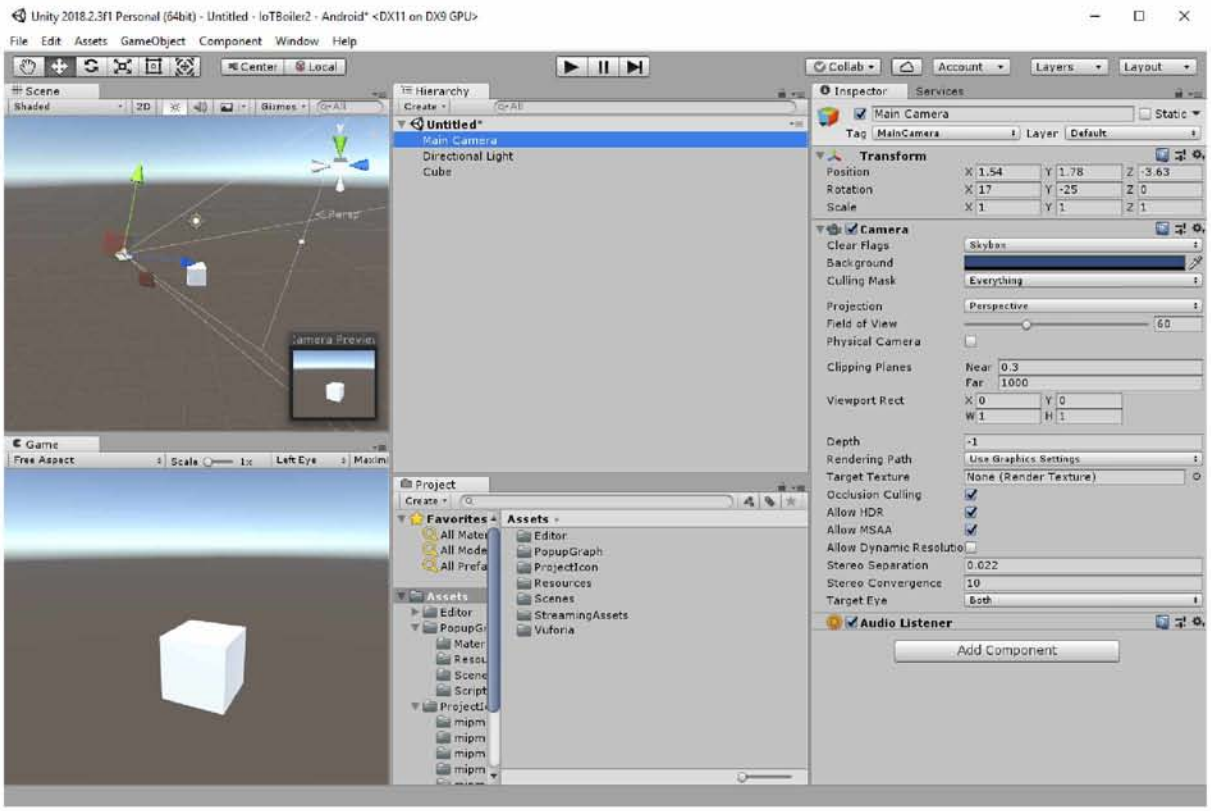
Εικόνα 8.6 – Toboiler.odαρχείο που μεταφορτώσαμε



Εικόνα 8.7 – Κατεβάζουμε την βάση δεδομένων επιλέγοντας UnityEditor

7.3.2 Unity Editor

Έχοντας κατεβάσει την βάση δεδομένων μας μπορούμε τώρα να συνεχίσουμε στην μηχανή Unity.OEditor της μηχανής Unity(εικόνα 8.8) αποτελείται από panels,που το καθένα εξυπηρετεί κάποιο σκοπό στο πρότζεκτ.



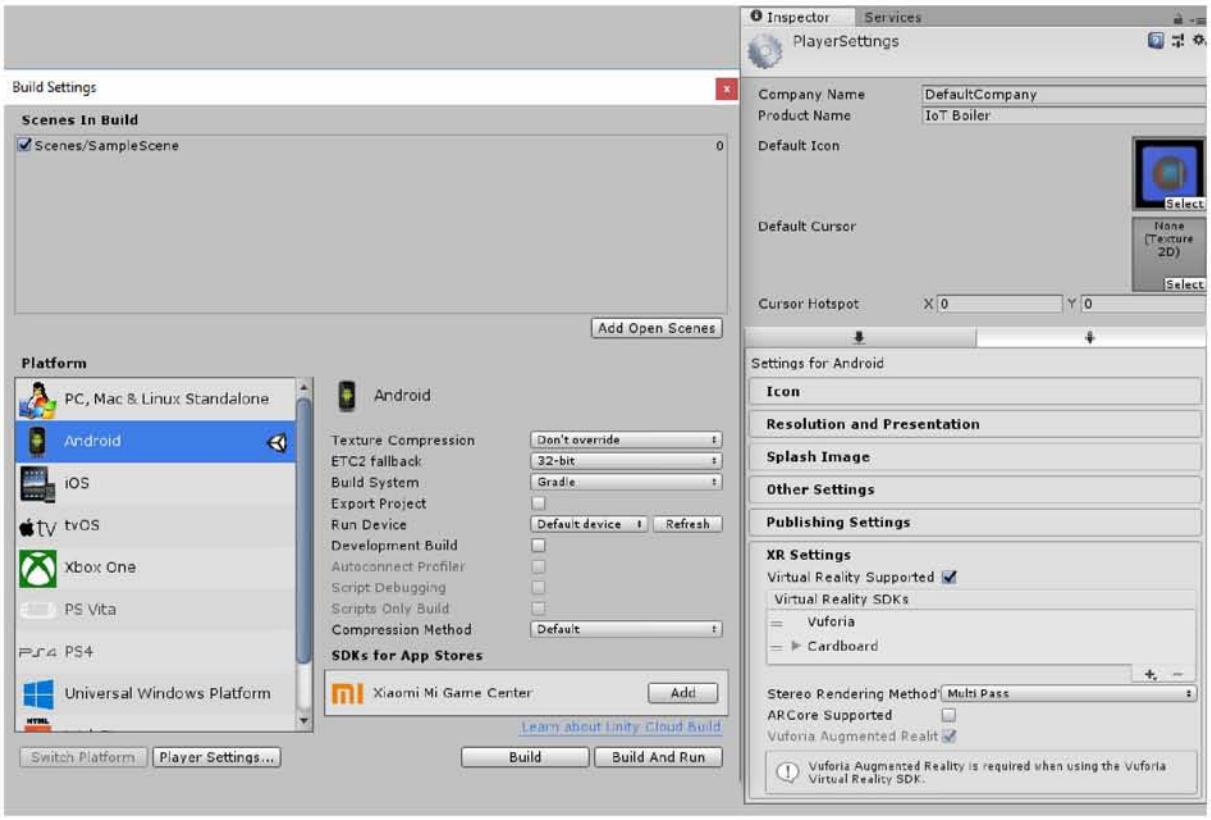
Εικόνα 8.8 – Ο Editorτης Unityμε πέντε ανοιχτά Panel

Τα πάνελ που θα χρησιμοποιήσουμε είναι:

- Το HierarchyPanel: είναι ένα παράθυρο όπου περιέχονται όλα τα αντικείμενα που βρίσκονται στην σκηνή.
- ToScenePanel: είναι ο χώρος κατασκευής της εφαρμογής όπου ο χρήστης μπορεί να πλοηγηθεί και να επεξεργαστεί.
- ToProjectPanel: είναι ένα παράθυρο το οποίο περιέχει όλα τα αρχεία που θα χρειαστούμε για την εφαρμογή.
- ToInspectorPanel: περιέχει όλες τις ρυθμίσεις και τα χαρακτηρίστηκα κάποιου αντικειμένου.
- ToGamePanel:αναπαριστά την εφαρμογή στην τελική μορφή της.

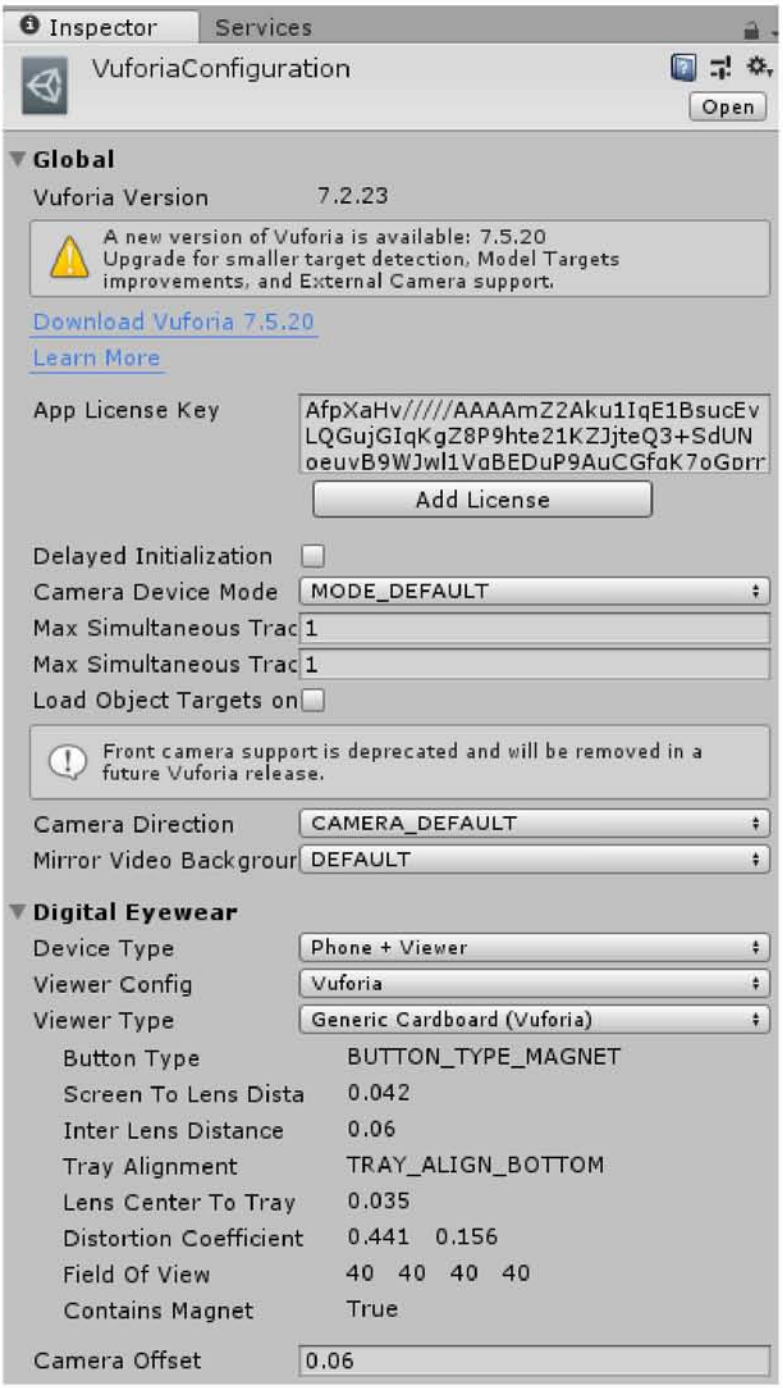
7.3.3 Ρυθμίσεις υποστήριξης Vuforia και εικονικής πραγματικότητας

Προτού ξεκινήσουμε να διαμορφώνουμε την σκηνή, θα πρέπει να αλλάξουμε κάποιες ρυθμίσεις, ώστε η εφαρμογή μας να είναι προορισμένη για android, να υποστηρίζει εικονική πραγματικότητα και τέλος να ενεργοποιήσουμε την υποστήριξη τουVuforia.Αυτό θα γίνει με το να μεταβούμε File ->BuildSettings… και στο νέο παράθυρο, που εμφανίζεταιαριστεράεπιλέγουμεAndroidκαι πατάμε SwitchPlatform. Στο ίδιο παράθυρο επιλέγουμε PlayerSettings…το οποίο εμφανίζει μια σειρά από ρυθμίσεις σε ένα νέο πανελ. Στο τέλος αυτών των ρυθμίσεων επιλέγουμε XRSettingsκαι ενεργοποιούμε την εικονική και επαυξημένη πραγματικότητα επιλέγοντας VirtualRealitySupported και VuforiaAugmentedReality. Εφόσον έχουμε επιλέξει αυτές τις δύο επιλογές κάτω από το VirtualRealitySupported, εμφανίζεται μια λίστα από επιλεγμένα SDK. Πατώντας το ‘+’ προσθέτουμε στην λίστα Vuforiακαι Cardboard.



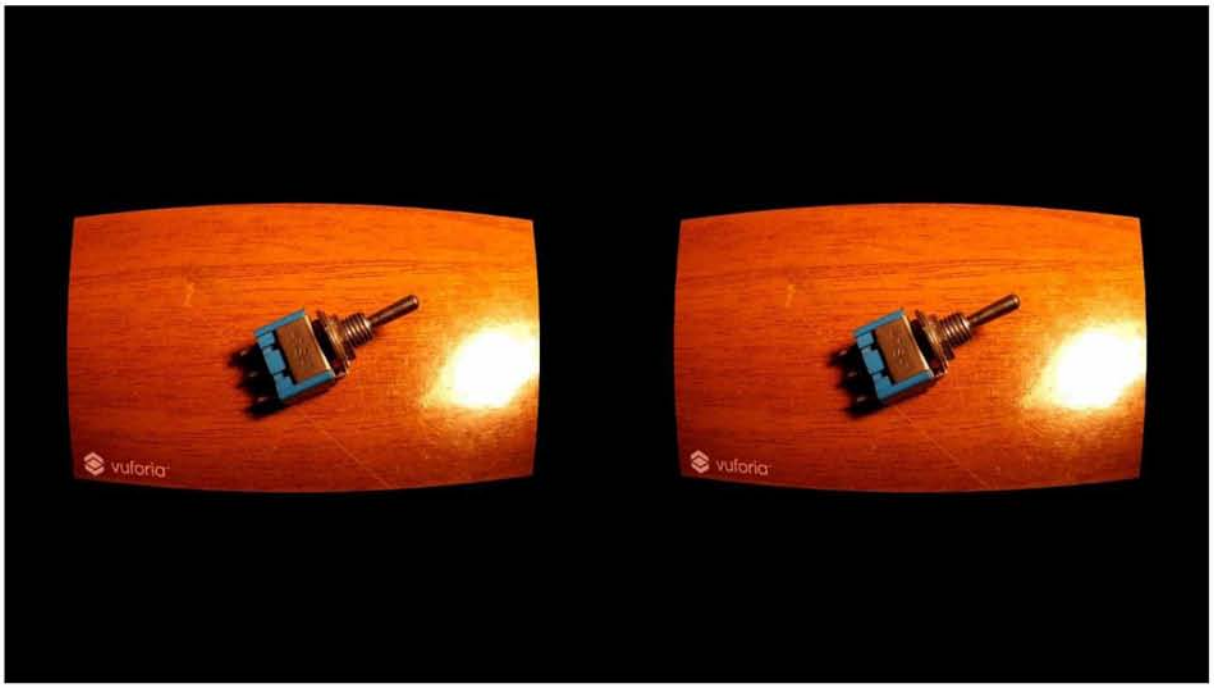
Εικόνα 8.9 – Αλλαγή πλατφόρμας σε Android και ενεργοποίηση Vuforiaκαι Εικονικής πραγματικότητας

Τώρα είμαστε έτοιμοι να ξεκινήσουμε να διαμορφώνουμε την σκηνή μας. Αρχικά διαγράφουμε το αντικείμενο MainCamera από το panelhierarchy. Πηγαίνοντας από το μενούAssets->Vuforia->ARCameraπροσθέτουμε στην σκηνή την κάμερα τουVuforia. Στο Inspectorpanel,αφού επιλέξουμε το αντικείμενο ARCameraεντοπίζουμε το componentVuforiaBehaviour(Script) και επιλέγουμε OpenVuforiaconfiguration. Εκεί, θα πρέπει να συμπληρώσουμε τοκλειδίτης άδειας (εικόνα 8.10) και να επιλέξουμε τον VRεξοπλισμό, που θα χρησιμοποιήσουμε.



Εικόνα 8.10 – Κλειδί άδειας Vuforiaκαι ρυθμίσεις DigitalEyewear

Για να δούμε ότι όλα έγιναν σωστά επιλέγουμε από το μενούτου Editor File ->BuildAndRunκαι περιμένουμε να εγκατασταθεί η εφαρμογή στο Android. Αν όλα έγιναν σωστά, τότε θα ξεκινήσει να εκτελείται μια εφαρμογή, που θα δείχνει την κάμερα σε εικονική πραγματικότητα (παράδειγμα στην εικόνα 8.11) μαζί με το λογότυπο του Vuforiaκάτω αριστερά.

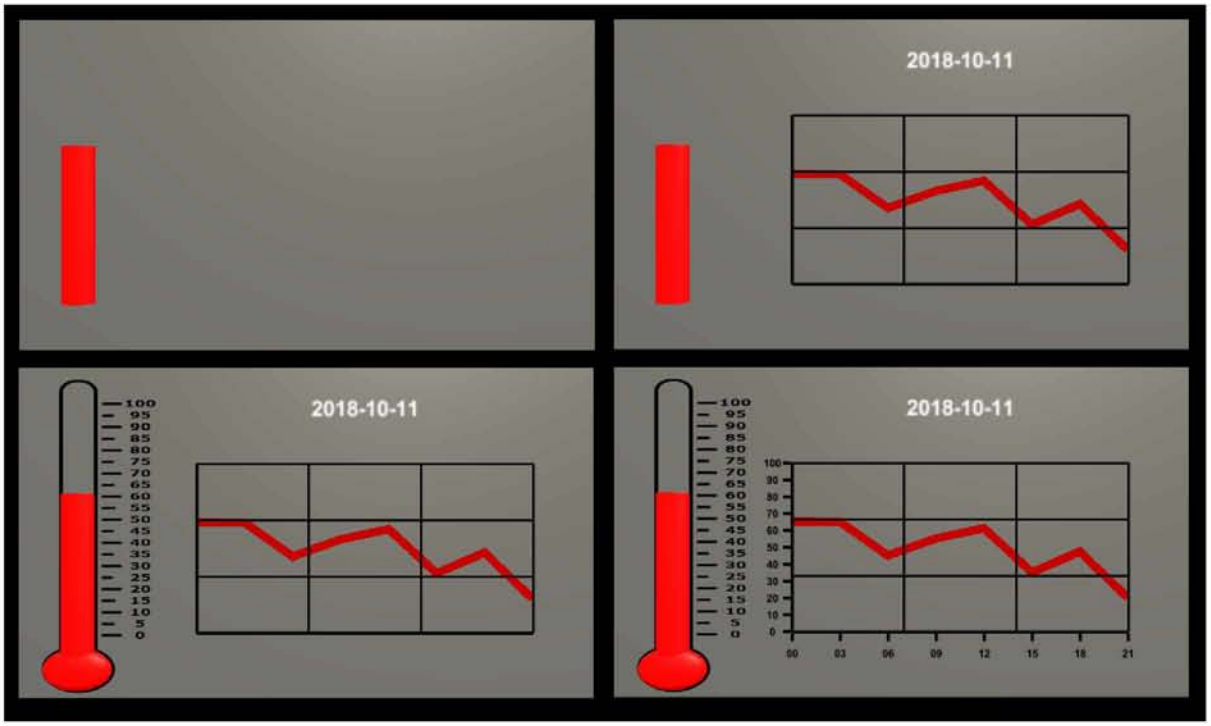


Εικόνα8.11 - ARCamerασε εικονική πραγματικότητα

7.3.4 Πάνελ με πληροφορίες του βραστήρα

Τοπάνελ, που σχεδιάσαμε για την εφαρμογή εκτελεί ένα HTTPαίτημα στο νέφος, το οποίο στέλνει σαν απάντηση ένα JSON αντικείμενο, που περιέχει την τρέχουσα θερμοκρασία του βραστήρα αλλά και μια σειρά μετρήσεων της μέσης θερμοκρασίας ανά τρεις ώρες των τελευταίων τριών ημερών.Τα δεδομένα αυτά, αφού παρθούν από το νέφος επεξεργάζονται και αποθηκεύονται σε ένα GameObjectαντικείμενο, που βρίσκεται καθ' όλη την διάρκεια εκτέλεσης του προγράμματος στην τρέχουσα σκηνή. Το πάνελ περιέχει δύο βασικά μέρη. Το πρώτο είναι ένα θερμόμετρο, που στην ουσία είναι ένα αντικείμενο σε σχήμα κυλίνδρου, που διαθέτει η Unityμηχανή (GameObject-> 3DObject->Cylinder). Έχουμε προσθέσει σε αυτό το αντικείμενο ένα script αρχείο, που ανάλογα με την θερμοκρασία, που έχει ο βραστήρας,μεταβάλλεται το ύψος του κυλίνδρου. Εισάγοντας κάποια γραφικά στοιχεία ο κύλινδρος μοιάζει με ένα κανονικό θερμόμετρο. Το δεύτερο μέρος είναι ένα γράφημα το οποίο θα δείχνει την μέση θερμοκρασία του βραστήρα ανά ημέρα. Το γράφημα παράγεται αυτόματα μέσω ενός scriptαρχείου με την έναρξη της εφαρμογής και δημιουργεί μια κινητή γραμμή, που αλλάζει κάθε δευτερόλεπτο. Ξεκινάει με την πιο παλιά ημερά και όταν δείξει όλο το είκοσι-τετράωρο προχωράει στην επόμενη μέρα ξεκινώνταστην γραμμή από την αρχή.

Ο άξονας Χ περιέχει ανά τρεις τις ώρες της ημέρας και ο άξονας Υ δείχνει την θερμοκρασία από 0 έως 100 βαθμούς.



Εικόνα 8.12 - Βήματα ανάπτυξης του πάνελ

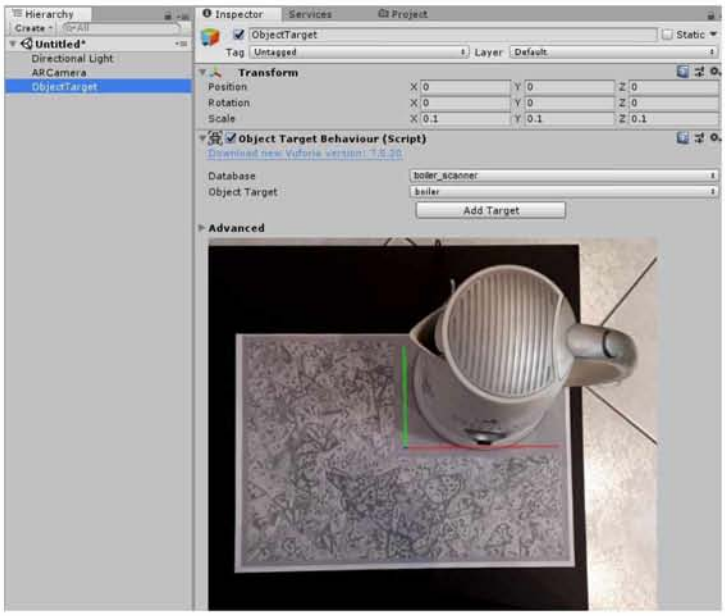
7.3.5 Object Tracking

Τελευταίο βήμα για την ολοκλήρωση της εφαρμογής είναι να προσθέσουμε τον βραστήρα, σαν στόχο, δηλαδή όταν η κάμερα εντοπίζει τον βραστήρα να εμφανίζεται το πάνελ, που είδαμε στην προηγούμενη ενότητα.

Στην ενότητα 8.3.1 κατεβάσαμε από το DeveloperPortaltης Vuforiaένα αρχείο με κατάληξη .unitypackage το οποίο περιέχει πληροφορίες για τον εντοπισμό του βραστήρα. Στο μενού του Editor**Assets->ImportPackage->CustomPackage…**Επιλέγουμε το αρχείο με κατάληξη **.unitypackage**και αφού ολοκληρωθεί η εισαγωγή του πακέτου προσθέτουμε στην σκηνή ένα αντικείμενο 3DScanπηγαίνοντας ξανά στο μενούτου Editor

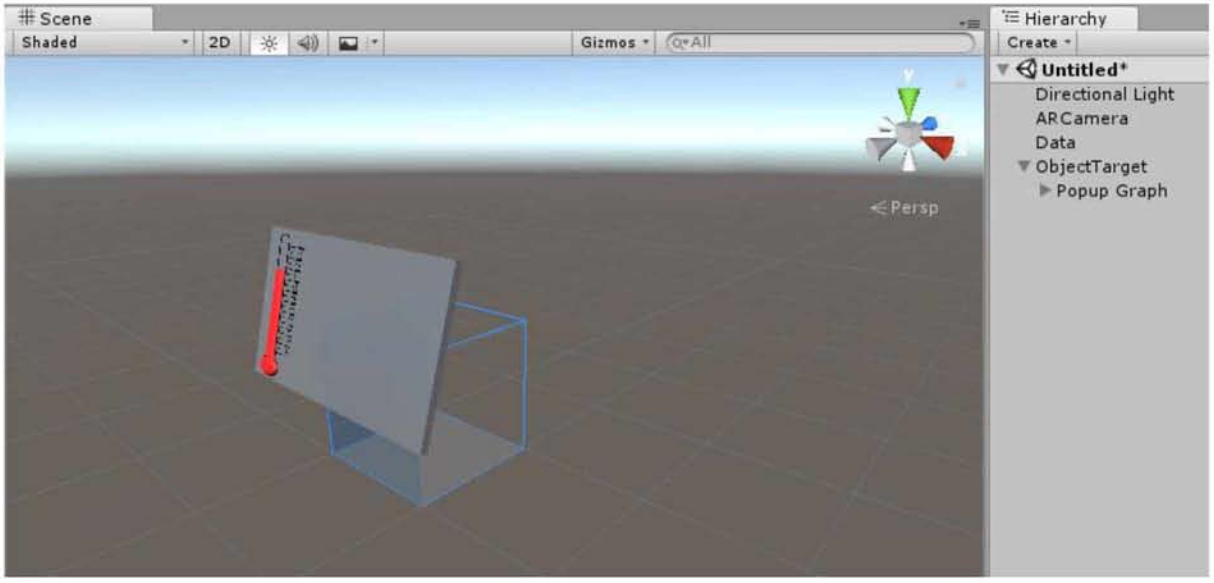
GameObject->Vuforia->3DScan

Αν προσέξουμε στον Inspectorθα δούμε ότι ο editortης Unityέχει ήδη επιλέξει την βάση δεδομένων και το αντικείμενο boiler,που είχαμε δημιουργήσει στην πρώτη ενότητα (εικόνα 8.13).



Εικόνα 8.13 – Επιλογή αντικειμένου για ανίχνευση

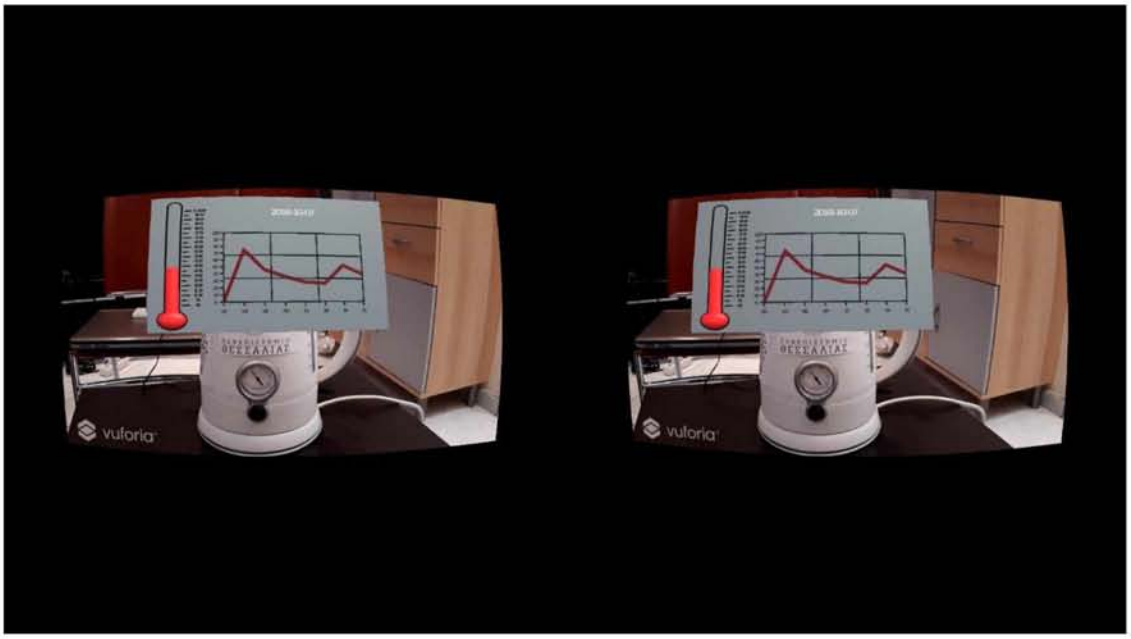
Τώρα πια η κάμερα τουVuforiaμπορεί να εντοπίσει τον βραστήρα. Για να εμφανίσουμε το πάνελ, όταν εντοπιστεί πρέπει να το προσθέσουμε στο hierarchypanel, σαν παιδί του αντικειμένου ObjectTarget.Ο κύβος που φαίνεται στο scenepanelαντιπροσωπεύει το μέγεθος του αντικειμένου, δηλαδή του βραστήρα. Έχοντας αυτό στο νου τοποθετούμε το πάνελ στο σημείο που θέλουμε να εμφανίζεται.



Εικόνα 8.14

7.4 Τελικό αποτέλεσμα

Εικόνα 8.15 – Ο βραστήρας με το Arduino(αριστερά) και το RaspberryPi(δεξιά)



Εικόνα 8.16 – Στιγμιότυπο της εφαρμογής

Συμπεράσματα - Προτάσεις

Με την χρήση της εφαρμογήςσε αυτήν την εργασία,αν κρίνουμε από τα υλικά που χρησιμοποιήσαμε,παρατηρούμε μια όμορφη και καθαρή εικόνα του πάνελ με τις πληροφορίες. Ακόμα, παρατηρήσαμε ότι η μέγιστη απόσταση της κάμερας με τον βραστήρα είναι περίπου ένα με τρία μέτρα γεγονός το οποίο δεν είναι ιδιαίτερα βολικό. Επίσης, η εφαρμογή σάρωσης αντικειμένου του κιτ ανάπτυξης λογισμικού Vuforia, δεν μας δίνει την δυνατότητα να σαρώσουμε μεγαλύτερες συσκευές. Έτσι, οδηγούμαστε στο συμπέρασμα ότι πρέπει να βρεθεί διαφορετική μέθοδος ανίχνευσης για μεγαλύτερα αντικείμενα. Το πρωτόκολλο HTTP που χρησιμοποιήσαμε στην εργασία για την επικοινωνία μεταξύ των συσκευών είναι αρκετά ικανοποιητικό για την συγκεκριμένη εφαρμογή, όμως υπάρχουν πιο γρήγορα πρωτόκολλα όπως το MQTT ή το XMPP για πιο απαιτητικές ταχύτητες που θα μπορούσαν να υλοποιηθούν σε συνδυασμό με κάποιον μεγάλο πάροχο νέφους όπως το AmazonWebServicesή το MicrosoftAzure.Η ασφάλεια στην επικοινωνία είναι ένα άλλο κομμάτι που λείπει από την εργασία, το οποίο πρέπει να αναλυθεί μόνο του, διότι η ασφάλεια στα συστήματα IoT είναι ένα τεράστιο κομμάτι. Μια ιδέα που βασίζεται πάνω σε αυτήν την εργασία είναι η ανάπτυξη ασφαλούς επικοινωνίας ανάμεσα στις συσκευές και το νέφος.

Παράρτημα Α: Κώδικας της Εργασίας

Αρχείο : updater.ino

```
001 #include"SoftwareSerial.h"
002
003Stringssid="SSID";
004String password="PASSWORD";
005/* ARDUINO PIN 6 <-> ESP RX */
006/* ARDUINO PIN 7 <-> ESP TX */
007 SoftwareSerial esp(6,7);
008 String server ="localserver.address";//Raspberry Pi LocalDB
009 String uri ="/update.php";
010
011/**
012     Συνάρτηση που εκτελείται
013     στην αρχή του προγράμματος.
014 */
015void setup()
016{
017//To serial που επικοινωνεί το esp8266
018     esp.begin(115200);
019//To serial που θα χρησιμοποιήσουμε για debugging
020     Serial.begin(115200);
021     esp_check();
022     connectWifi();
023}
024
025/**
026     Έλεγχος λειτουργίας του ESP8266
027     Στέλνουμε για είσοδο το μήνυμα
028     "AT" αν το όλαδουλεύουν σωστά
029     τυπώνει το μήνυμα "OK".
030 */
031void esp_check()
032{
033     esp.println("AT");
034     delay(1000);//1000ms
035if(esp.find("OK"))
036         Serial.println("Module Tested");
037}
038
039/**
040     Σύνδεσητου ESP8266 στοδίκτυο
041 */
042void connectWifi()
043{
044     String cmd ="AT+CWJAP=\""+ssid+"\",\""+ password +"\"";
045     esp.println(cmd);
046     delay(4000);
047if(esp.find("OK"))
048         Serial.println("Connected!");
049else{
050         connectWifi();
051         Serial.println("Cannot connect to wifi");
052}
```

```
053}
054
055/**
056     Συνάρτηση που εκτελείται
057     συνέχεια κατά την λειτουργία
058     του Arduino.
059 */
060void loop ()
061{
062     httppost();
063     delay(500);
064}
065
066/**
067     HTTP POST αίτηση (request) απότο Arduino
068     στοντοπικό (local) servertou
069     δικτύου.
070 */
071void httppost ()
072{
073/* Υπολογισμός της θερμοκρασίας απο το θερμιστορ. */
074float t = calculate_temp();
075     Serial.println(t);
076/* Μαζί με την θερμοκρασία πρέπει να στείλουμε
077     ενα σωστό username και ένα password για να δεχτεί
078     ο server τηνπληροφορία. */
079     String data ="user=admin&pass=admin&temp=";
080     data += t;
081
082/* Δημιουργεί μια TCP σύνδεση */
083     esp.println("AT+CIPSTART=\"TCP\",\""+ server +"\",80");
084if( esp.find("OK"))
085          Serial.println("TCP connection ready");
086
087     delay(500);
088
089/* Ολόκληρο το POST αίτηση (request) */
090     String postRequest =
091"POST "+ uri +" HTTP/1.0\r\n"+
092"Host: "+ server +"\r\n"+
093"Accept: *"+"/"+"*\r\n"+
094"Content-Length: "+ data.length()+"\r\n"+
095"Content-Type: application/x-www-form-urlencoded\r\n"+
096"\r\n"+ data;
097     String sendCmd ="AT+CIPSEND=";
098esp.print(sendCmd);
099/* Μαζί με την αίτηση πρέπει να στείλουμε
100     και το μέγεθος ολόκληρου του μηνύματος. */
101     esp.println(postRequest.length());
102
103     delay(500);
104
105if(esp.find(">"))
106{
107          Serial.println("Sending..");
108          esp.print(postRequest);
109if(esp.find("SEND OK"))
110{
111/*Τομήμυμαστάλθηκε */
112               Serial.println("Packet sent");
113while(esp.available()){
```

```
114                 String tmpResp = esp.readString();
115Serial.println(tmpResp);
116}
117/* Τερματισμόςτηςσύνδεσης */
118             esp.println("AT+CIPCLOSE");
119}
120}
121}
122
123
```

Αρχείο : thermistor.ino

```
01 /* τιμή της αντίστασης που
02 χρησιμοποιήσαμε στην διαίρεση
03 τασης (10KOhm) */
04constintR1 =9940;
05/* Β παράμετρος που δίνεται από τον θερμίστορ. */
06constfloat Bt =3950.0;
07
08/**
09  Υπολογισμός της θερμοκρασίας
10  σε βαθμούς Κελσίου από την τιμή
11  της αντίστασης του θερμίστορ.
12
13  return: float
14  Η θερμοκρασία σε βαθμούς Κελσίου.
15 */
16float calculate_temp()
17{
18/* Θερμοκρασία σε βαθμούς Κελσίου */
19float Tc =0;
20/* Διαβάζουμε την τιμή του θερμίστορ από το Arduino */
21int x = analogRead(A0);
22/* Μετατροπή σε αντίσταση */
23float Vout = x *(5/1023.0);
24/* Η αντίσταση του θερμίστορ από τον διαιρέτη τάσης */
25float Rt = R1 * Vout /(5.0- Vout);
26/* 298.155 αντιπροσωπεύει 25 βαθμούς Κελβιν */
27float a1 =1/298.15;
28float b1 =1/ Bt;
29float c1 = log(Rt /10000.0);
30float y1 = a1 + b1 * c1;
31/* Θερμοκρασία σε Κελβιν */
32float T =1/ y1;
33/* Θερμοκρασία σε Κελσίου */
34  Tc = T -273.15;
35return(Tc);
36}
```

Αρχείο : app.php

```php
<?php
/* Σύνδεσηστον MySQL server */
$mysqli=new mysqli("localhost","root","","cloud_db");
$user=$_GET["user"];
$pass=$_GET["pass"];
$device_id=$_GET["device_id"];

/* Έλεγχος των username και password */
$result=$mysqli->query("SELECT * FROM account
  WHERE username = \"$user\" AND password = \"$pass\"");
if($result->num_rows ==0)
{
http_response_code(404);
die();
}

/* Ζητάμε την τελευταία θερμοκρασία που
 υπάρχειστονserver */
$result=$mysqli->query("SELECT temperature
  FROM thing WHERE id=$device_id");
$row=mysqli_fetch_row($result);
$temperature=$row[0];
/* Κρατάμε την θερμοκρασία στο json αντικείμενο */
$json->temperature[]=$temperature;

/* Επίσης ζητάμε την ημερομηνία, ώρα και μέση θερμοκρασία
 τωντελευταίωντριώνημερών. */
$result=$mysqli->query("SELECT m_date, m_time, avg_temperature
  FROM day_measurements
  WHERE m_date >= DATE_ADD(CURDATE(), INTERVAL -2 DAY)
  AND thing_id = $device_id ORDER BY m_date, m_time");
while($row=mysqli_fetch_row($result))
{
$hour=explode(":",$row[1]);
//Join date, time and temperature into one string.
//e.g "2018-08-20>03>85.6"
/* Κρατάμε τα αποτελέσματα στο json αντικείμενο */
$json->days[]=$row[0].">".$hour[0].">".$row[2];
}

//Echo αναπαράσταση JSON
echojson_encode($json);
?>
```

Αρχείο : collect.php

```
01 <?php
02/* Σύνδεσηστον MySQL server */
03$mysqli=new mysqli("localhost","root","","cloud_db");
04$user=$_POST["user"];
05$pass=$_POST["pass"];
06$device_id=$_POST["device_id"];
07$timestamp=$_POST["timestamp"];
08$current_temp=$_POST["cur_temp"];
09$avg_temp=$_POST["avg_temp"];
10
11/* Έλεγχος των username και password */
12$result=$mysqli->query("SELECT * FROM account
13  WHERE username = \"$user\" AND password = \"$pass\"");
14if($result->num_rows ==0)
15{
16http_response_code(404);
17die();
18}
19
20/* Ανανέωση της βάσης δεδομένων με την νέα θερμοκρασία. */
21$result=$mysqli->query("UPDATE thing SET
22  temperature=$current_temp WHERE id=$device_id")ordie();
23
24/* Κράταμετηντωρινήημερομηνία */
25$datetime=explode(" ",$timestamp);
26$date=$datetime[0];
27$time=explode(":",$datetime[1]);
28
29/* Ανανεώνουμε την μέση θερμοκρασία
30 ανά τρεις ώρες. */
31$t=intval($time[0]);
32if($t>=0and$t<3)$time[0]="00";
33elseif($t>=3and$t<6)$time[0]="03";
34elseif($t>=6and$t<9)$time[0]="06";
35elseif($t>=9and$t<12)$time[0]="09";
36elseif($t>=12and$t<15)$time[0]="12";
37elseif($t>=15and$t<18)$time[0]="15";
38elseif($t>=18and$t<21)$time[0]="18";
39elseif($t>=21and$t<0)$time[0]="21";
40elsedie();
41
42/* Ζητάμε την ημερομηνία και ώρα από τον πίνακα με τις
43 μετρήσειςτωντριώνωρών */
44$result=$mysqli->query("SELECT m_date, m_time FROM day_measurements
45  WHERE m_date=\"$date\" AND m_time = \"$time[0]:00:00\"");
46
47/* Αν υπάρξει αποτέλεσμα τότε ανανεώνουμε την θερμοκρασία,
48 διαφορετικά εισάγουμε νέα εγγραφή με την νέα ώρα και θερμοκρασία. */
49if($result->num_rows ==0)
50{
51$result=$mysqli->query("INSERT INTO day_measurements\
52 (thing_id, m_date, m_time, avg_temperature)
53      VALUES($device_id, \"$date\", \"$time[0]:00:00\", $avg_temp)");
54}
55else
56{
57$result=$mysqli->query("UPDATE day_measurements SET
58      avg_temperature=$avg_temp WHERE thing_id=$device_id AND
59      date(m_date)=\"$date\" AND HOUR(m_time) = $time[0]");
```

```
60}
61?>
```

Αρχείο : gateway_requests.php

```
01 <?php
02/* Σύνδεσηστον MySQL server */
03$mysqli=new mysqli("localhost","root","","gateway_localdb");
04$request=$_POST["request"];
05
06if($request=='1')
07{
08/* Ζητάμε την τελευταία θερμοκρασία από τον server. */
09$result=$mysqli->query("SELECT id, temperature FROM boiler ORDER BY id
DESC LIMIT 1");
10$row=mysqli_fetch_row($result);
11$msg->id =$row[0];
12$msg->temp =$row[1];
13$myJSON=json_encode($msg);
14echo$myJSON;
15}
16else
17{
18$current_date=date('Y-m-d');
19$current_hour=date('H');
20$f="00";
21$s="00";
22$t="00";
23
24$t=intval($current_hour);
25if($t>=0and$t<3){$f="00";$s="01";$t="02";}
26elseif($t>=3and$t<6){$f="03";$s="04";$t="05";}
27elseif($t>=6and$t<9){$f="06";$s="07";$t="08";}
28elseif($t>=9and$t<12){$f="09";$s="10";$t="11";}
29elseif($t>=12and$t<15){$f="12";$s="13";$t="14";}
30elseif($t>=15and$t<18){$f="15";$s="16";$t="17";}
31elseif($t>=18and$t<21){$f="18";$s="19";$t="20";}
32elseif($t>=21and$t<0){$f="21";$s="22";$t="23";}
33
34/* Ζητάμε την μέση τιμή της θερμοκρασίας των τελευταίων τριών ωρών. */
35$result=$mysqli->query("SELECT AVG(temperature) FROM boiler
36      WHERE DATE(time) = '$current_date'
37      AND (HOUR(time) = $f OR HOUR(time) = $s OR HOUR(time) = $t)");
38$row=mysqli_fetch_row($result);
39$msg->id =$row[0];
40$myJSON=json_encode($msg);
41echo$myJSON;
42}
43?>
```

Αρχείο : update.php

```php
<?php
/* Σύνδεσηστον MySQL server */
$mysqli=new mysqli("localhost","root","","gateway_localdb");
$user=$_POST["user"];
$pass=$_POST["pass"];
$temp=$_POST["temp"];

/* Έλεγχοςτων username και password αποτο Arduino */
$result=$mysqli->query("SELECT * FROM account
    WHERE username = \"$user\" AND password = \"$pass\"");
if($result->num_rows ==0)
{
$write="invalid username or password";
file_put_contents('index.html',$write);//Debug
die();
}

/* Ζητάμε την τωρινή ημερομηνία και ώρα. */
date_default_timezone_set('Europe/Athens');
$current_time=date('Y-m-d H:i:s');

/* Ζητάμε την τελευταία εγγραφή και αν είναι διαφορετική
τότε εισάγουμε την νέα θερμοκρασία στον τοπικό server. */
$result=$mysqli->query("SELECT * FROM boiler ORDER BY id DESC LIMIT 1");
$row=mysqli_fetch_row($result);
if($row[2]!=$temp)
{
$sql="INSERT INTO boiler (time, temperature)
        VALUES ('$current_time', '$temp')";
}
$mysqli->close();
?>
```

Αρχείο : Boiler.cs

```
01using UnityEngine;
02
03/// <summary>
04/// Αποθηκεύει τις πληροφορίες από
05/// τον (cloud) serversεμορφή
06/// YYYY-MM-DD>HH>TEMPERATURE
07/// </summary>
08publicclassBoiler
09{
10/* Τελευταίαμέτρησηθερμοκρασίας. */
11publicstring[] temperature =newstring[1];
12/* Η μέση θερμοκρασία των τελευταίων τριών ημερών. */
13publicstring[] days =newstring[20];
14
15public Boiler()
16{
17          temperature[0]="0.00";
18for(int i =0; i < days.Length; i++)
19              days[i]="";
20}
```

```
21
22///<summary>
23/// Μετάφραση του JSON (string) σε αντικείμενο.
24///</summary>
25publicstatic Boiler CreateFromJSON(string jsonString)
26{
27return JsonUtility.FromJson<Boiler>(jsonString);
28}
29}
```

Αρχείο : Day.cs

```
01/// <summary>
02/// Η πληρόφορίες (θερμοκρασίες ανα τρεις ώρες)
03/// τηςημέρας.
04/// </summary>
05publicclass Day
06{
07publicfloat[] avg_temperatures;
08publicstring day;
09
10public Day()
11{
12          avg_temperatures =newfloat[9];
13}
14}
```

Αρχείο : Http.cs

```
01using UnityEngine;
02using UnityEngine.Networking;
03using System.Collections;
04
05/// <summary>
06/// ΕκτελείHTTPPOSTαιτήσειςκαιστην
07/// συνέχεια αποθηκεύει την απάντηση σε
08/// ένα αντικείμενο Boiler.
09/// </summary>
10publicclass Http : MonoBehaviour
11{
12public Boiler b;
13
14/// <summary>
15/// Συνάρτηση που εκτελείται με την
16/// κατασκευή του αντικειμένου
17/// </summary>
18void Start()
19{
20          b =new Boiler();
21/* Εκτέλεση μιας HTTP αίτησης (request)
22       κάθε 1 δευτερόλεπτο. */
23       StartCoroutine(FetchDataFromWeb());
24}
25
26/// <summary>
```

```
27/// Ρουτίνα που ζητάει από το (cloud) server
28/// την τελευταία μέτρηση θερμοκρασίας και τα δεδομένα
29/// των τελευταίων τριών ημερών.
30/// </summary>
31  IEnumerator FetchDataFromWeb()
32{
33while(true)
34{
35string setURL ="http://cloud.address/app.php\
36 ?user=admin&pass=admin&device_id=1";
37              WWW www =new WWW(setURL);
38yieldreturn www;
39string json;
40              json = www.text;
41
42/*Boiler αντικείμενο απο το JSON*/
43              b = Boiler.CreateFromJSON(json);
44
45yieldreturnnew WaitForSeconds(1f);
46}
47}
48}
49
```

Αρχείο : Temperature.cs

```
01using UnityEngine;
02
03/// <summary>
04/// Ανανεώνειτοεικονικόθερμόμετρο.
05/// </summary>
06publicclass Temperature : MonoBehaviour {
07
08/* Το GameObject με τις
09  πληροφορίες απο τον (cloud) server */
10public GameObject data;
11/* Οιπληροφορίεςαποτον (cloud) server */
12private Http online_data;
13
14/* Συνάρτηση που εκτελείται
15  στην αρχή του προγράμματος.  */
16void Start (){
17          online_data = data.GetComponent<Http>();
18}
19
20/* Ανανεώνει το εικονικό θερμόμετρο
21  ανάλογα με την τιμή που δίνει το Boiler
22  αντικείμενο. */
23void Update (){
24
25float t =0.0f;
26if(online_data.b.temperature !=null)
27              t =float.Parse(online_data.b.temperature[0]);
28
29float x =0.08f+((t *0.06f)/5);
30if(t >=0.0)
31              transform.localScale =new
32               Vector3(transform.localScale.x, x, transform.localScale.z);
```

```
}
}
```

Αρχείο : gateway_localdb.sql

```
--
-- Table structure for table `account`
--
CREATETABLEIFNOTEXISTS `account` (
 `id` int(11)NOTNULL,
   `username` varchar(32)NOTNULL,
   `password` varchar(32)NOTNULL
) ENGINE=MyISAM AUTO_INCREMENT=2DEFAULT CHARSET=latin1;

INSERTINTO `account` (`id`, `username`, `password`)VALUES
(1,'admin','admin');
--
-- Table structure for table `boiler`
--
CREATETABLEIFNOTEXISTS `boiler` (
 `id` int(11)NOTNULL,
   `time` datetimeNOTNULL,
   `temperature` floatNOTNULL
) ENGINE=InnoDB AUTO_INCREMENT=6509DEFAULT CHARSET=latin1;
```

Αρχείο : cloud_db.sql

```
--
-- Table structure for table `account`
--
DROPTABLEIFEXISTS `account`;
CREATETABLEIFNOTEXISTS `account` (
   `id` int(10) UNSIGNED NOTNULL AUTO_INCREMENT,
   `username` varchar(255)NOTNULL,
   `password` varchar(255)NOTNULL,
PRIMARYKEY(`id`)
) ENGINE=MyISAM AUTO_INCREMENT=2DEFAULT CHARSET=latin1;
INSERTINTO `account` (`id`, `username`, `password`)VALUES
(1,'admin','admin');
--
-- Table structure for table `day_measurements`
--
DROPTABLEIFEXISTS `day_measurements`;
CREATETABLEIFNOTEXISTS `day_measurements` (
   `id` int(11)NOTNULL AUTO_INCREMENT,
   `thing_id` int(11)NOTNULL,
   `m_date` dateNOTNULL,
   `m_time` time(6)NOTNULL,
   `avg_temperature` floatNOTNULL,
PRIMARYKEY(`id`)
) ENGINE=MyISAM AUTO_INCREMENT=90DEFAULT CHARSET=latin1;
```

Αρχείο : gateway.py

```
01import requests, json
02import time
03import datetime
04try:
05#Το id της τελευταίας μέτρησης της θερμοκρασίας.
06      latest_id =-1
07#Η πιο πρόσφατη μέτρηση θερμοκρασίας.
08      temperature =-1
09#Η μέση θερμοκρασία των τελευταίων τριών ωρών.
10      avg_temperature =-1
11whileTrue:
12#HTTP POST αίτημα για την πιο πρόσφατη μέτρηση θερμοκρασίας
13      response = requests.post(
14'http://127.0.0.1/gateway_requests.php',
15          data ={'request':1}
16)
17if response.ok:
18          r = response.json()
19          temperature = r['temp']
20          r = int(r['id'])
21else:
22print("Request Failed")
23          quit()
24
25#Αν έχουμε νέα τιμή θερμοκρασίας
26if(r > latest_id):
27          latest_id = r
28else:
29          time.sleep(1)
30continue
31
32      now = datetime.datetime.now()
33      timestamp = now.strftime("%Y-%m-%d %H:%M:%S")
34
35#HTTP POST αίτημα για τον νέο υπολογισμό μέσης
36#θερμοκρασίας των τελευταίων τριών ωρών.
37      response = requests.post(
38'http://127.0.0.1/gateway_requests.php',
39              data={'request':2}
40)
41if response.ok:
42              r = response.json()
43              avg_temperature = round(float(r['id']),2)
44#print(avg_temperature)
45
46      r = requests.post("http://192.168.1.62/thesis/collect.php",
47          data ={'user':'admin','pass':'admin','device_id':1,
48'timestamp':timestamp,'cur_temp':temperature,
49'avg_temp':avg_temperature
50}
51)
52except KeyboardInterrupt:
53print("KeyboardInterrupt")
54      quit()
```

Αρχείο : StreamingLineChart.cs

```
001using System.Collections;
002using System.Collections.Generic;
003using UnityEngine;
004
005/// <summary>
006/// Παράγει την γραμμή με την μεταβολή
007/// τηςμέσηςθερμοκρασίας.
008/// </summary>
009publicclass StreamingLineChart : StreamingChart
010{
011     LineRenderer lineRenderer;
012int currentDataBatches =-1;
013
014/// <summary>
015/// Συνάρτηση που εκτελείται με
016/// την δημιουργία του αντικειμένου.
017/// </summary>
018void Start()
019{
020         lineRenderer =new LineRenderer();
021base.Init();
022}
023
024/// <summary>
025/// Προσθέτει τις τιμές με την μέση θερμοκρασία
026/// απο το struct attributeParams στην μεταβλητή
027/// που παράγει την γραμμή του γραφήματος.
028/// </summary>
029publicoverridevoid AddMark(AttributeParams attributeParams)
030{
031         lineRenderer = Instantiate(Resources.
032             Load("LineChart",typeof(LineRenderer)),
033             gameObject.transform)as LineRenderer;
034         lineRenderer.widthMultiplier = lineWidth;
035         lineRenderer.alignment = LineAlignment.TransformZ;
036}
037
038/// <summary>
039/// Κάθε φορά που καλείται προσθέτει την μέση
040/// θερμοκρασία των επόμενων τριών ωρών.
041/// </summary>
042publicoverridevoid UpdateData(float newDataValues)
043{
044
045if(attributes.dataValues.Count > maxDataBatches)
046             attributes.dataValues.Dequeue();
047
048if(newDataValues >=0.0)
049             attributes.dataValues.Enqueue(newDataValues);
050
051         RenderLines();
052}
053
054/// <summary>
055/// Σχεδιάζει ευθείες οι οποίες αποτελούν
056/// την γραμμή του γραφήματος με την μεταβολή
057/// τηςθερμοκρασίας
058/// </summary>
```

```
059void RenderLines()
060{
061
062int i =0;
063         AttributeParams attParams = attributes;
064         lineRenderer.positionCount = attParams.dataValues.Count;
065foreach(floatvaluein attParams.dataValues)
066{
067             lineRenderer.SetPosition(i,new Vector3(
068                 ScaleLinear((float)i,0,(float)maxDataBatches +1,
0690, chartWidth +.28f),
070                 ScaleLinear(value, attParams.minValue,
071                 attParams.maxValue,0, chartHeight),0));
072             i++;
073}
074         currentDataBatches++;
075
076if(currentDataBatches ==(maxDataBatches -1))
077{
078             StartCoroutine(ClearLine("temperature",0,100));
079currentDataBatches=0;
080}
081}
082
083/// <summary>
084/// Ρουτίνα που εκτελείται όταν το γράφημα
085/// αναπαραστήσει τις θερμοκρασίες μια ημέρας.
086/// περιμένει 0.25 δευτερόλεπτα πριν διαγράψει
087/// την γραμμή του γραφήματος για να φανεί
088/// η μεταβολή της θερμοκρασίας του τελευταίου
089/// τρίωρου.
090/// </summary>
091public IEnumerator ClearLine(
092string name,float minValue,float maxValue)
093{
094yieldreturnnew WaitForSeconds(0.25f);
095
096         attributes.name = name;
097         attributes.minValue = minValue;
098         attributes.maxValue = maxValue;
099         attributes.dataValues =new Queue<float>();
100
101         lineRenderer = gameObject.transform.
102             Find("LineChart(Clone)").GetComponent<LineRenderer>();
103for(int i =0; i <= maxDataBatches; i++){
104             Debug.Log(i);
105             lineRenderer.SetPosition(i, Vector3.zero);
106}
107         lineRenderer.widthMultiplier = lineWidth;
108         lineRenderer.alignment = LineAlignment.TransformZ;
109
110}
111}
```

Αρχείο : StreamingChart.cs

```
001using System.Collections;
002using System.Collections.Generic;
003using UnityEngine;
004
005publicclass StreamingChart : MonoBehaviour {
006
007// Το πλάτοστουγραφήματος.
008publicfloat chartWidth =2f;
009// Το ύψοστουγραφήματος.
010publicfloatchartHeight=0.5f;
011// Το πλάτος της γραμμής του γραφήματος.
012publicfloat lineWidth =0.01f;
013// Ο τίτλος που θα φαίνεται στην αρχή πάνω από γράφημα
014publicstring chartTitle ="Connecting to server...";
015// Ο μέγιστος αριθμός τιμών που θα φαίνεται στο
016// γράφημα. (24 ώρες / 3 = 8)
017publicint maxDataBatches =8;
018// Το GameObject αντικείμενο με τις πληροφορίες
019// απότον server.
020public GameObject data;
021
022private Http online_data;
023private Day[] days;
024privateint hour_c =0;
025privateintday_c=0;
026
027//Παράμετροι και τιμές για το γράφημα
028publicstruct AttributeParams
029{
030//Όνομαγραμμής
031publicstring name;
032//Μέγιστητιμήτου dataValues
033publicfloat minValue;
034//Ελάχιστητιμήτου dataValues
035publicfloat maxValue;
036//Οι τιμές τις μέσης θερμοκρασίας.
037public Queue<float> dataValues;
038}
039public AttributeParams attributes;
040
041/// <summary>
042/// Από αυτή την συνάρτηση ξεκινά
043/// η δημιουργία του γραφήματος.
044/// </summary>
045publicvoid Init()
046{
047         online_data = data.GetComponent<Http>();
048         attributes =new AttributeParams();
049
050// Construct axes and (empty) labels.
051         ConstructChart();
052// Sets data parameters.
053         AddDataParam("temperature",0,100);
054
055// Adds new simulated data every half second.
056         InvokeRepeating("SimulateNewData",0.0f,0.5f);
057}
```

```
058
059/// <summary>
060/// Κατασκευάζει τους άξονες και
061/// τον τίτλο του γραφήματος.
062/// </summary>
063void ConstructChart()
064{
065         ConstructAxes();
066         ConstructTitle();
067}
068/// <summary>
069/// Κατασκευάζει τους άξονες γραφήματος.
070/// </summary>
071privatevoid ConstructAxes()
072{
073// Οάξονας X
074         DrawLine(new Vector3(0,0,0),
075new Vector3(chartWidth,0,0),0.01f*2.0f);
076
077// ΟάξοναςY
078         DrawLine(new Vector3(0,0,0),
079new Vector3(0, chartHeight,0),0.01f*2.0f);
080
081// Βοηθητικέςγραμμές
082int numGuideLines =3;
083float zpos =0.00f;
084for(int g =1; g <= numGuideLines; g++)
085{
086float ypos = ScaleLinear((float)g,0,
087(float)(numGuideLines),0, chartHeight);
088             DrawLine(new Vector3(0, ypos, zpos),
089new Vector3(chartWidth, ypos, zpos),0.01f*1.5f);
090
091float xpos = ScaleLinear((float)g,0,
092(float)(numGuideLines),0, chartWidth);
093             DrawLine(new Vector3(xpos,0, zpos),
094new Vector3(xpos, chartHeight, zpos),0.01f*1.5f);
095}
096}
097
098/// <summary>
099/// Κατασκευάζει τον τίτλο του γραφήματος.
100/// </summary>
101privatevoid ConstructTitle()
102{
103         GameObject title = Instantiate(Resources.
104             Load("Text"), gameObject.transform)as GameObject;
105         title.transform.localPosition =new Vector3(
106             chartWidth /2.0f,1.4f,0);
107         title.GetComponent<TextMesh>().text = chartTitle;
108}
109
110/// <summary>
111/// Θέτει τις παραμέτρους όνομα, μέγιστη
112/// και ελάχιστη τιμή του γραφήματος.
113/// </summary>
114publicvoid AddDataParam(
115string name,float minValue,float maxValue)
116{
117         attributes.name = name;
118         attributes.minValue = minValue;
```

```
119         attributes.maxValue = maxValue;
120         attributes.dataValues =new Queue<float>();
121         AddMark(attributes);
122}
123
124/// <summary>
125/// Επεξεργάζει τις πληροφορίες από
126/// το αντικείμενο Boiler του GameObject data
127/// δημιουργώντας τρία αντικείμενα Day.
128/// </summary>
129void SimulateNewData()
130{
131         days =new Day[3];
132int day_counter =-1;
133string first_day = online_data.b.days[0].Split('>')[0];
134for(int i =0; i < online_data.b.days.Length; i++)
135{
136string[] fields = online_data.b.days[i].Split('>');
137if(fields.Length ==3)
138{
139if(fields[1]=="00"|| first_day != fields[0])
140{
141                     first_day = fields[0];
142                     days[++day_counter]=new Day();
143                     days[day_counter].day = fields[0];
144}
145int index =int.Parse(fields[1])/3;
146                 days[day_counter].
147                     avg_temperatures[index]=float.Parse(fields[2]);
148}
149}
150
151         UpdateData(CreateSimulatedData());
152}
153
154publicvirtualvoid UpdateData(float newDataValues)
155{
156
157}
158
159/// <summary>
160/// Επιστρέφει τις θερμοκρασίες μια-μια
161/// κάθε φορά που εκτελείται. Όταν σταλθούνε
162/// όλες ξεκινά από την αρχή. Αν αλλάξει η ημερομηνία
163/// τότε την τοποθετεί στον τίτλο του γραφήματος.
164/// </summary>
165float CreateSimulatedData()
166{
167float newData =-1.0f;
168
169if(days[day_c]!=null)
170             gameObject.transform.Find
171("Text(Clone)").GetComponent<TextMesh>
172().text = days[day_c].day;
173
174
175if(days[day_c]!=null)
176{
177             newData = days[day_c].avg_temperatures[hour_c];
178             hour_c++;
179}
```

```
180
181if(hour_c ==8)
182{
183             hour_c =0;
184             day_c++;
185}
186
187if(day_c ==3)
188             day_c =0;
189
190return newData;
191}
192
193publicvirtualvoid AddMark(AttributeParams attributes){}
194
195/// <summary>
196/// Σχεδιάζει μια γραμμή από το διάνυσμα start
197/// μέχρι το διάνυσμα end.
198/// </summary>
199void DrawLine(Vector3 start, Vector3 end,float lineWidth)
200{
201         LineRenderer l = Instantiate(Resources
202.Load("AxisLine",typeof(LineRenderer)),
203             gameObject.transform)as LineRenderer;
204         l.SetPosition(0, start);
205         l.SetPosition(1, end);
206         l.widthMultiplier = lineWidth;
207}
208
209/// <summary>
210/// Υπολογίζειτηνκλίμακαμιαςγραμμής.
211/// </summary>
212publicfloat ScaleLinear(floatvalue,float domainMin,
213float domainMax,float rangeMin,float rangeMax)
214{
215
216return rangeMin +
217(((value- domainMin)/
218(domainMax - domainMin))*
219(rangeMax - rangeMin));
220}
221
222}
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Industry 4.0 The Industrial Internet of Things - Alasdair Gilchrist

[2] Internet of Things A to Z Technologies and Applications – Qusay F. Hassan

[3] The Industrial Internet of Things Volume G1: Reference Architecture - Industrial Internet Consortium

ΔΙΑΔΙΚΤΥΑΚΟΙ ΤΟΠΟΙ

[4] https://en.wikipedia.org/wiki/Internet_of_things

[5] https://www.sas.com/el_gr/insights/big-data/internet-of-things.html

[6] https://www.youtube.com/watch?v=QSIPNhOiMoE

[7] https://www.arduino.cc/

[8] https://unity3d.com/

[9] https://developer.vuforia.com/

[10] https://en.wikipedia.org/wiki/Fog_computing

[11] https://i.dell.com/sites/doccontent/shared-content/data-sheets/en/Documents/value-of-bringing-analytics-at-edge-final.pdf

[12] https://www.elprocus.com/building-the-internet-of-things-using-raspberry-pi/

[13] https://www.tutorialspoint.com/http/http_requests.htm

[14] https://searchmicroservices.techtarget.com/definition/RESTful-API

[15] http://www.iebmedia.com/index.php?id=11008&parentid=63&themeid=255&hft=89&showdetail=true&bb=1

[16] https://www.rfc-editor.org/rfc/rfc7452.txt

[17] https://internetofthingsagenda.techtarget.com/definition/Industrial-Internet-of-Things-IIoT

[18] https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

[19] https://www.networkworld.com/article/3224893/internet-of-things/what-is-edge-computing-and-how-it-s-changing-the-network.html

[20] https://medium.com/mqtt-buddy/mqtt-vs-http-which-one-is-the-best-for-iot-c868169b3105

[21] https://www.raspberrypi.org/help/

[22] https://www.hackster.io/suai-ihpcnt/salutis-project-bf49b4

[23] https://www.pubnub.com/blog/moving-the-cloud-to-the-edge-computing/