



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ
Η/Υ

Διπλωματική Εργασία

"Έξυπνο σύστημα οικιακού αυτοματισμού βασισμένο σε Arduino, απομακρυσμένη παρακολούθηση και διαχείριση για την εξοικονόμηση ενέργειας"

"Smart home automation system based on arduino, remote monitoring and management for energy saving "

Ελευθέριος Σμυρναίος

Επιβλέπων Καθηγητής :

Τσουκαλάς Ελευθέριος, Καθηγητής ΠΘ

Συνεπιβλέπουσα Καθηγήτρια:

Δασκαλοπούλου Ασπασία, Επίκουρη Καθηγήτρια Π.Θ.

Βόλος, Οκτώβριος 2017

(κενή σελίδα)



ΕΥΧΑΡΙΣΤΙΕΣ

Αρχικά νιώθω ευγνωμοσύνη για την συνεργασία μου με τον κ.Τσουκαλά, ο οποίος με καθοδηγούσε και με συμβούλευε καθ'όλη την διάρκεια της περιόδου που εκτελούσα την διπλωματική μου εργασία.Επίσης θα ήθελα να εκφράσω την αγάπη μου στους ανθρώπους που γνώρισα κατά την διάρκεια των φοιτητικών μου σπουδών και να τους ευχαριστήσω για τις όμορφες αλλά και δύσκολες στιγμές που μοιραστήκαμε.Τέλος, θέλω ολόψυχα, μέσα από την καρδιά μου να πω ένα τεράστιο ευχαριστώ στην οικογένεια μου και κυρίως στους γονείς μου, οι οποίοι πάντα μου έδιναν τη δύναμη, πάρα τις δυσκολίες που περνούσαν.Αποτέλεσαν τον κύριο παράγοντα ώστε να με κάνουν να μην τα βάλω ποτέ κάτω και να προσπαθώ συνεχώς να πετύχω τους στόχους μου, με όποιο εμπόδιο και δυσκολία βρέθηκαν στον δρόμο μου.



(κενή σελίδα)



ΠΕΡΙΛΗΨΗ

Η παρούσα διπλωματική εργασία, θέλει να αναδείξει την ευκολία, την απλότητα, την ευελιξία, την χρησιμότητα που έχει ένα «έξυπνο σπίτι», αλλά και την καλύτερη διαχείριση της ενέργειας που καταναλώνει ένα σπίτι με τη χρήση ενός μικροελεγκτή. Θα αναδείξει ουσιαστικά την καρδιά του «έξυπνου σπιτιού», ποιες είναι οι βασικές του μονάδες, πώς λειτουργούν και πώς αλληλεπιδρούν μεταξύ τους.

Ένα "έξυπνο σπίτι" είναι συνήθως ένα οικιακό περιβάλλον που έχει μερικώς αυτοματοποιηθεί. Ο αυτοματισμός στο σπίτι περιλαμβάνει κεντρικό έλεγχο φωτισμού, HVAC (θέρμανση, εξαερισμό και κλιματισμό), διαχείριση συσκευών και άλλα. Ο αυτοματισμός στο σπίτι στοχεύει στην ενίσχυση της άνεσης, της αποδοτικότητας της κατανάλωσης ενέργειας και της ασφάλειας. Γενικά, τα σπίτια είναι εξοπλισμένα με ανεξάρτητους πίνακες ελέγχου για τον έλεγχο όλων των συστημάτων και συσκευών που υπάρχουν στο σπίτι. Επιπλέον, οι εν λόγω πίνακες ελέγχου συχνά δεν συνδέονται μεταξύ τους. Ο κύριος σκοπός ενός έξυπνου σπιτιού είναι να συγκεντρώσει τον έλεγχο όλων των συσκευών σε μια ενιαία μονάδα ελέγχου που μπορεί να προγραμματιστεί για να κάνει συγκεκριμένες εργασίες κατάλληλες για τον ιδιοκτήτη και το σπίτι. Ο στόχος ενός έξυπνου σπιτιού δεν είναι μόνο η ευκολία, αλλά και η μείωση της κατανάλωσης πόρων όπως η ηλεκτρική ενέργεια, το φυσικό αέριο κλπ. Λόγω της τρέχουσας τιμολόγησης της ενέργειας, η διατήρηση των πόρων έχει γίνει μέρος μιας καθημερινής ζωής ατόμων. Εάν ένα άτομο έχει τη δυνατότητα να ελέγχει εξ αποστάσεως τον οικιακό αυτοματισμό του, μπορεί να



μειώσει την κατανάλωση ενέργειας και έτσι να μειώσει τα έξοδα. Επιπλέον, η περιβαλλοντική βιωσιμότητα έχει αποκτήσει ενδιαφέρον κατά τα τελευταία έτη. Αν κάποιος είναι μακριά από το σπίτι, δεν χρειάζεται να λειτουργεί το κλιματιστικό ή ο εξαερισμός. Η ίδια αρχή ισχύει για φωτισμό, θέρμανση και άλλες συσκευές. Ορισμένα συστήματα έξυπνων κατοικιών διακόπτουν τη λειτουργία των εξαρτημάτων μέχρι να χρειαστούν ξανά.

Επιπλέον, υπάρχουν πολλές διαφορετικές τεχνολογίες για την εφαρμογή τέτοιων έξυπνων σπιτιών. Ορισμένα πρότυπα χρησιμοποιούν πολύπλοκα πρωτόκολλα επικοινωνίας και καλωδίωση ελέγχου, ενώ άλλα βασίζονται σε ενσωματωμένα σήματα στο υπάρχον κύκλωμα ισχύος του σπιτιού.

Λέξεις κλειδιά: Έξυπνο σπίτι, arduino, αυτοματισμός, εξοικονόμηση ενέργεια, μικροελεγκτής, εξ' αποστάσεως αυτοματισμός, κατανάλωση ενέργειας



ABSTRACT

This diploma thesis wants to highlight the ease, simplicity, flexibility, usefulness of a "smart home", and better management of the energy consumed by a home using a microcontroller. It will bring out the heart of the "smart home", what are its core units, how they work and how they interact with each other.

A "smart home" typically is a domestic environment that has been partially automated. Home automation includes centralized control for lighting, HVAC (heating, ventilation and air conditioning), appliance management, and others. Home automation aims to enhance the comfort, energy consumption efficiency and security in domestic scenarios. Generally, houses are equipped with independent control panels to control all of the systems and appliances present in the house. Moreover, those control panels are often not related each other. The main purpose of a smart home is to centralize the control of all the devices into a single control unit which can be programmed to do specific tasks suitable for the owner and the home in question. The goal of a smart home is not only convenience but also to reduce the consumption of resources such as power, gas, etc. Due to the current pricing on energy, resource conservation has become a part of a person's day-to-day life. If a person has the possibility to control his home automation remotely he can reduce the consumption of energy and thus cutting down on expenses. Furthermore, environmental sustainability has gained relevance in the latest years. If a person is away from home



there is no need for the air conditioner or ventilation to operate. The same principle applies to illuminations, heating and other appliances. Some smart homes systems pause the operation of appliances until they are needed again.

Furthermore, there are several different technologies for implementing such smart homes. Some standards utilize complex communication protocols and control wiring, others rely on embedded signals in the existing power circuit of the house.

Key words: Smart home, arduino, microcontroller, automation, energy saving, automation remotely, power consumption



ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	4
ABSTRACT.....	6
ΠΡΟΛΟΓΟΣ.....	10
1. ΕΙΣΑΓΩΓΗ.....	13
1.1 Μικροελεγκτής Arduino.....	13
. 1.1.1 Τι είναι ο μικροελεγκτής και που χρησιμοποιείται.....	16
. 1.1.2 Ιστορική αναδρομή.....	18
. 1.1.3 Είδη διάφορων μοντέλων	20
. 1.1.4 Χαρακτηριστικά του arduino Mega 2560.....	21
2. ΠΕΡΙΦΕΡΕΙΑΚΑ ΥΛΙΚΑ ΚΑΙ SHIELDS.....	24
. 2.1 Arduino Shields.....	25
. 2.2 Περιφερειακά υλικά.....	27
2.2.1 Wires.....	27
2.2.2 Resistors.....	28
2.2.3 Buzzer.....	30
2.2.4 Leds.....	31
2.2.5 Relay 8 channels.....	32
2.2.6 Sensors.....	33



2.2.7 Breadboard.....	37
2.2.8 Bluetooth HC-05.....	38
3. ΥΛΟΠΟΙΗΣΗ ΚΩΔΙΚΑ.....	39
3.1 Πλατφόρμα ανάπτυξης λογισμικού Arduino IDE.....	39
3.1.1 Συνάρτηση θερμοκρασίας/υγρασίας	41
3.1.2 Συνάρτηση κίνησης(φως).....	43
3.1.3 Συνάρτηση αισθητήρα φωτός.....	44
3.1.4 Συνάρτηση κίνησης(συναγερμός).....	45
3.2 Εφαρμογή Blynk.....	46
4. Έξυπνο σπίτι.....	49
4.1 Η εφαρμογή ασύρματων συστημάτων οικιακού αυτοματισμού..	49
4.2 Κόστος διάφορων συστημάτων αυτοματισμού.....	51
4.3 Τα οφέλη ενός “έξυπνου σπιτιού”.....	52
4.4 Σενάριο.....	55
5. ΣΥΜΠΕΡΑΣΜΑΤΑ	57
ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ.....	60
ΠΑΡΑΡΤΗΜΑ 1.....	61
ΠΑΡΑΡΤΗΜΑ 2.....	63
ΠΑΡΑΡΤΗΜΑ 3.....	67
DATASHEETS.....	72
ΒΙΒΛΙΟΓΡΑΦΙΑ	73



ΠΡΟΛΟΓΟΣ

Στη σημερινή πραγματικότητα, η έλλειψη ηλεκτρισμού είναι αυτή του μεγάλου προβλήματος που αντιμετωπίζουν οι άνθρωποι καθημερινά στη ζωή. Οι πόροι πρέπει να χρησιμοποιούνται οικονομικά για τη διατήρησή τους για μελλοντική χρήση, δεδομένου ότι είναι περιορισμένοι. Πρέπει λοιπόν να εξοικονομούμε ενέργεια καθημερινά. Για να ξεπεραστεί η κατάσταση αυτή πρέπει να προταθεί ένα σύστημα έτσι ώστε οι ηλεκτρικές συσκευές να σβήνουν όταν δεν χρησιμοποιούνται γιατί ο άνθρωπος πολλές φορές ξεχνάει να τις απενεργοποιήσει. Επομένως χρειάζεται ένα «έξυπνο» σύστημα για τη λειτουργία, αλλά και τη διαχείριση των ηλεκτρικών συσκευών από απόσταση. Πολλές μέθοδοι έχουν προταθεί για τη στρατηγική διατήρησης της ενέργειας και αυτός είναι ο στόχος της συγκεκριμένης διπλωματικής εργασίας, δηλαδή ένα σύστημα με χαμηλό κόστος για την αποσύνδεση ή τη σύνδεση των οικιακών συσκευών, αλλά και τη διαχείρισή τους, για την όσο το δυνατόν χαμηλότερη κατανάλωση ενέργειας. Πριν περάσουμε όμως, στα επιμέρους ηλεκτρολογικά του μέρη, θα δούμε πρώτα τι κάνει αυτό το «έξυπνο σπίτι», γιατί είναι έξυπνο και πόσο και, καταληκτικά, να αποφανθούμε για το αν και κατά πόσο είναι οικονομικά συμφέρον προς εμάς να μένουμε σε ένα τέτοιο σπίτι και γιατί;

Τα «έξυπνα σπίτια» μπορούν να αναλάβουν πρωτοβουλίες, όπως να ρυθμίσουν την εσωτερική θερμοκρασία τους, να κλείσουν τα καλοριφέρ, να προσομοιώσουν κίνηση ανοιγοκλείνοντας τα φώτα και τα ρολά, αποθαρρύνοντας τους διαρρήκτες κατά την απουσία μας, ή



απλώς να ενημερώσουν τον ιδιοκτήτη για την κατάσταση της οικίας του, μέσω διαδικτύου - θερμοκρασία, ηλιοφάνεια, βροχή, αέρας, αποθέματα νερού, πετρελαίου, κατάσταση ρολών, εξόδων και οτιδήποτε άλλο για το οποίο θα έπρεπε να είμαστε ενήμεροι.

Τα σενάρια που μπορούμε να εφαρμόσουμε είναι ουσιαστικά άπειρα και αφορούν την εξοικείωση του χρήστη με το πρόγραμμα. Γι' αυτό πολλές φορές το σύστημα προγραμματίζεται με κάποια βασικά σενάρια και στην πορεία, ανάλογα με τις επιθυμίες του εκάστοτε χρήστη της οικίας, προσαρμόζεται σε πιο σύνθετα σενάρια. Μελλοντικές επεκτάσεις ή διαφοροποιήσεις πραγματοποιούνται με μικρό κόστος, εφόσον η αλλαγή ενυπάρχει του προγραμματισμού. Μερικά από τα πιθανά σενάρια που μπορούμε να εφαρμόσουμε σε μια οικία είναι:

- * Φεύγω / Έρχομαι (όταν αποχωρώ από το σπίτι το σύστημα κλείνει όλες τις ηλεκτρικές καταναλώσεις.
- * Σενάρια φωτισμού κατοικίας (party mode, home cinema κ.α.)
- * Κλείσιμο, άνοιγμα όλων των ρολών ταυτόχρονα / ασφάλιση της κατοικίας (το βράδυ ή όταν ξυπνάμε το πρωί).
- * Δυνατότητα προγραμματισμού αυτ όματης εκτέλεσης λειτουργιών (ανάβουν σταδιακά τα φώτα όσο δύει ο ήλιος, ανοίγουν αυτόματα τα ρολά όταν έχουμε alarm φωτιάς, κ.α.).



* Επιστρέφουμε από τη δουλειά - με την χρήση του τηλεφώνου ανάβουμε το θερμοσίφωνο πριν φτάσουμε ή κλείνουμε την παροχή ρεύματος σε κάποια συσκευή που έχουμε ξεχάσει ανοιχτή.

* Χρονοπρογράμματα για το αυτόματο πότισμα.

* Έλεγχος θέρμανσης ή κλιματισμού.

* Αναφορές κατάστασης για εσωτερική, εξωτερική θερμοκρασία, βροχή, υγρασία κτλ.

Αξίζει βέβαια να αναφερθεί ότι τα έξυπνα σπίτια δεν προάγουν μόνο την άνεση, την ευκολία και την εξοικονόμηση πολύτιμου χρόνου, αλλά μπορούν να επιτελέσουν και σημαντικό ρόλο στην υγεία και στη φροντίδα ανθρώπων που το έχουν ανάγκη. Στη Βρετανία μελετώνται έξυπνα σπίτια που φροντίζουν τους ηλικιωμένους, καταγράφοντας τις κινήσεις τους και παρακολουθώντας την υγεία τους. Νέες τεχνολογίες, όπως αισθητήρες, δορυφορικά συστήματα, GPS, που «κρύβονται» στα «έξυπνα σπίτια», θα επιτρέπουν στους ηλικιωμένους να ζουν ανεξάρτητοι, με ασφάλεια στο δικό τους χώρο για μεγαλύτερο διάστημα, χωρίς να απαιτείται η παρουσία τρίτων.



1. ΕΙΣΑΓΩΓΗ

1.1 Μικροελεγκτής Arduino

Το Arduino είναι ένας μικροελεγκτής μονής πλακέτας, δηλαδή μια απλή μητρική πλακέτα ανοιχτού κώδικα με ενσωματωμένο έναν μικροεπεξεργαστή και εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure data, SuperCollider. Επίσης, μπορεί να χρησιμοποιηθεί για τη δημιουργία διαδραστικών αντικειμένων, λαμβάνοντας δεδομένα από μια ποικιλία συλλογών των διακοπών ή αισθητήρων και τον έλεγχο μιας σειράς φώτων, κινητήρων και άλλων φυσικών εξόδων. Εστιάζοντας γύρω από το περιβάλλον προγραμματισμού μέσω επεξεργασίας, η γλώσσα προγραμματισμού του Arduino είναι μια εκτέλεση της καλωδίωσης, μιας πλατφόρμας φυσικής επεξεργασίας υπολογιστών.



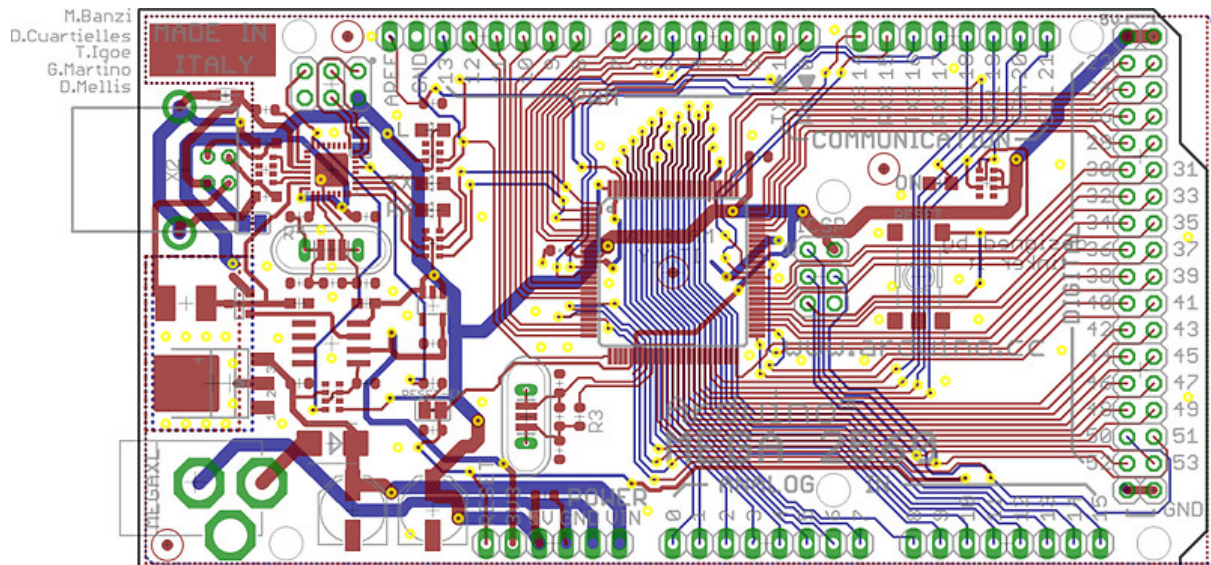
1.Εικόνα

Arduino Mega 2560

Το Arduino είναι φυσική επεξεργασία ανοικτού κώδικα, η οποία βασίζεται σε πίνακα μικροελεγκτών και ένα ενσωματωμένο περιβάλλον ανάπτυξης για την προγραμματισμένη πλακέτα. Το πρόγραμμα Arduino μπορεί να λειτουργήσει με λειτουργίες Windows, Macintosh και Linux(OS).Είναι ένα εργαλείο που χρησιμοποιείται για την κατασκευή μιας καλύτερης έκδοσης ενός υπολογιστή που μπορεί να ελέγχει,να αλληλεπιδρά και να “αισθάνεται” περισσότερο από έναν κανονικό επιτραπέζιο υπολογιστή. Είναι μια φυσική επεξεργασία ανοικτού κώδικα γύρω από μια κάρτα μικροελεγκτή.

Επιπλέον είναι συμβατά boards που χρησιμοποιούν την τεχνολογία των shields, τυπωμένων boards επεκτάσιμων κυκλωμάτων που συνδέονται στα κανονικά παρεχόμενα του Arduino pin-headers.Αυτά

τα shields μπορούν να παρέχουν έλεγχο στα motors, GPS, Ethernet, LCD εικόνας.



Arduino Mega 2560 Reference Design

Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, AND IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any feature or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

2.Εικόνα Arduino Mega 2560 Reference Design

1.1.1 Τι είναι ο μικροελεγκτής και που χρησιμοποιείται

Φανταστείτε το μικροελεγκτή σαν ένα τσιπάκι που περιέχει μέσα του ένα ολόκληρο μικρό υπολογιστή. Είναι ένας υπολογιστής ειδικού σκοπού: δεν θα χρησιμοποιηθεί για να γραφτεί ένα κείμενο αλλά μπορεί να προγραμματιστεί να ελέγχει άλλες συσκευές: να αναβοσβήνει φωτάκια (LED), να περιστρέφει κινητήρες κλπ.

Χρησιμοποιείται παντού γύρω μας, χωρίς συχνά να δίνεται σημασία. Σχεδόν όλες οι ηλεκτρονικές συσκευές που διαθέτουν οι άνθρωποι έχουν τουλάχιστον έναν μέσα τους. Το touchpad στο laptop που γράφτηκε αυτή η διπλωματική αυτή τη στιγμή ελέγχεται από έναν μικροελεγκτή: διαβάζει την επιφάνεια αφής και στέλνει τα αντίστοιχα δεδομένα στην κεντρική πλακέτα του υπολογιστή. Το ψυγείο, ο φούρνος μικροκυμάτων, το κινητό, η τηλεόραση, το αυτοκίνητο, ο διεθνής διαστημικός σταθμός και τα ρομπότ που έχουν σταλεί στον Άρη είναι γεμάτα μικροελεγκτές.

Στους σύγχρονους μικροεπεξεργαστές για μη ενσωματωμένα συστήματα (πχ τους μικροεπεξεργαστές των προσωπικών υπολογιστών), δίνεται έμφαση στην υπολογιστική ισχύ. Η ευελιξία ανάπτυξης διαφορετικών εφαρμογών είναι μεγάλη, καθώς η λειτουργικότητα του τελικού συστήματος καθορίζεται από τα εξωτερικά περιφερειακά τα οποία διασυνδέονται με την κεντρική μονάδα (μικροεπεξεργαστή), η οποία δεν είναι εξειδικευμένη. Αντίθετα, στους μικροεπεξεργαστές για ενσωματωμένα συστήματα (μικροελεγκτές), οι οποίοι έχουν μικρότερες ή και μηδαμινές δυνατότητες συνεργασίας με εξωτερικά περιφερειακά, αυτού του



είδους, η ευελιξία είναι περιορισμένη, καθώς και η υπολογιστική ισχύς. Οι μικροελεγκτές δίνουν έμφαση στο μικρό αριθμό ολοκληρωμένων κυκλωμάτων που απαιτείται για τη λειτουργία μιας συσκευής, το χαμηλό κόστος και την εξειδίκευση.

Αναλυτικά, τα πλεονεκτήματα των μικροελεγκτών είναι:

1. Αυτονομία, μέσω της ενσωμάτωσης σύνθετων περιφερειακών υποσυστημάτων όπως μνήμες και θύρες επικοινωνίας. Έτσι πολλοί μικροελεγκτές δεν χρειάζονται κανένα άλλο ολοκληρωμένο κύκλωμα για να λειτουργήσουν.
2. Η ενσωμάτωση περιφερειακών σημαίνει ευκολότερη υλοποίηση εφαρμογών λόγω των απλούστερων διασυνδέσεων. Επίσης, οδηγεί σε χαμηλότερη κατανάλωση ισχύος, μεγιστοποιώντας τη φορητότητα και ελαχιστοποιεί το κόστος της συσκευής στην οποία ενσωματώνεται ο μικροελεγκτής.
3. Χαμηλό κόστος.
4. Μεγαλύτερη αξιοπιστία, και πάλι λόγω των λιγότερων διασυνδέσεων.
5. Μειωμένες εκπομπές ηλεκτρομαγνητικών παρεμβολών και μειωμένη ευαισθησία σε αντίστοιχες παρεμβολές από άλλες ηλεκτρικές και ηλεκτρονικές συσκευές. Το πλεονέκτημα αυτό προκύπτει από το μικρότερο αριθμό και μήκος εξωτερικών διασυνδέσεων καθώς και τις χαμηλότερες ταχύτητες λειτουργίας.
6. Περισσότεροι διαθέσιμοι ακροδέκτες για ψηφιακές εισόδους-εξόδους (για δεδομένο μέγεθος ολοκληρωμένου κυκλώματος),



λόγω της μη δέσμευσής τους για τη σύνδεση εξωτερικών περιφερειακών.

7. Μικρό μέγεθος συνολικού υπολογιστικού συστήματος.

1.1.2 Ιστορική αναδρομή

Το 2005 ένα σχέδιο ξεκίνησε προκειμένου να φτιαχτεί μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο φθηνή από άλλα πρωτότυπα συστήματα διαθέσιμα εκείνη την περίοδο. Οι ιδρυτές Massimo Banzi και David Cueartielles ονόμασαν το σχέδιο από τον Arduino της Ivrea και ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρέα, κομόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας.

Το Arduino ξεκίνησε στο Ινστιτούτο Σχεδίασης Αλληλεπίδρασης Ivrea (IDII) στην Ivrea της Ιταλίας. Εκείνη την εποχή, οι μαθητές χρησιμοποίησαν έναν μικροελεγκτή BASIC Stamp με κόστος \$100, ένα σημαντικό κόστος για πολλούς φοιτητές. Το 2003, ο Hernando Barragán δημιούργησε την πλατφόρμα ανάπτυξης ως σχέδιο διπλωματικής εργασίας στο IDII, υπό την επίβλεψη των Massimo Banzi και Casey Reas, οι οποίοι είναι γνωστοί για τη δουλειά τους στη γλώσσα επεξεργασίας. Ο στόχος του έργου ήταν να δημιουργηθούν απλά, χαμηλού κόστους εργαλεία για τη δημιουργία ψηφιακών έργων



από μη μηχανικούς. Η πλατφόρμα καλωδίωσης αποτελείται από έναν πίνακα τυπωμένου κυκλώματος (PCB) με έναν μικροελεγκτή ATmega168, έναν IDE βασισμένο σε λειτουργίες επεξεργασίας και βιβλιοθήκης για τον εύκολο προ γραμματισμό του μικροελεγκτή. Το 2003, ο Massimo Banzi, με τον David Mellis, έναν άλλο φοιτητή IDII και τον David Cuartielles, πρόσθεσε την υποστήριξη του φτηνότερου μικροελεγκτή ATmega8 στην καλωδίωση. Αλλά αντί να συνεχίσουν τις εργασίες σχετικά με την καλωδίωση, διέκοψαν το έργο και μετονομάστηκαν σε Arduino.

Η αρχική ομάδα του Arduino αποτελείται από τον Massimo Banzi, τον David Cuartielles, τον Tom Igoe, τον Gianluca Martino και τον David Mellis, χωρίς όμως τον Barragán που δεν κλήθηκε να συμμετάσχει. Μετά την ολοκλήρωση της πλατφόρμας καλωδίωσης, διανεμήθηκαν ελαφρύτερες και λιγότερο ακριβές εκδόσεις στην κοινότητα ανοιχτού κώδικα.

Η Adafruit Industries, ένας προμηθευτής πινάκων, εξαρτημάτων και Arduino της Νέας Υόρκης, εκτιμάται στα μέσα του 2011 ότι πάνω από 300.000 Arduino είχαν παραχθεί εμπορικά και το 2013 700.000 πλακέτες Arduino ήταν στα χέρια των χρηστών.

Τον Μάρτιο του 2009 ανακοινώθηκε το Arduino Mega, με το οποίο και έγινε η έρευνα και τα πειράματα για την συγκεκριμένη διπλωματική εργασία, είναι βασισμένο στο Atmel ATmega1280.

1.1.3 Μοντέλα μικροελεγκτών Arduino

Στην εικόνα 3 και 4 παρουσιάζονται τα διάφορα μοντέλα μικροελεγκτών arduino. Οι διαφορές τους συνήθως είναι στους ακροδέκτες (pins) που έχουν, στην τάση εισόδου και εξόδου, καθώς και στα χαρακτηριστικά των συστημάτων που υλοποιούν (arduino Mega συνήθως χρησιμοποιείται για βιομηχανικά συστήματα ενώ το arduino Uno συνήθως για συστήματα έρευνας ή ερασιτεχνικά). Για την ανάπτυξη του συστήματος 'Smart Green' Home χρησιμοποιήθηκε το arduino MEGA 2560.

Μοντέλα μικροελεγκτών Arduino



Εικόνα 4: Arduino UNO

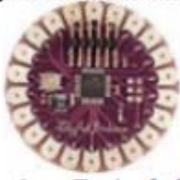





Εικόνα 5: Arduino



Εικόνα 6: Arduino Mega 2560

3. Εικόνα
Μοντέλα μικροελεγκτών

Leonardo		
 Εικόνα 7: Arduino LilyPad	 Εικόνα 8: Arduino Mega ADK	 Εικόνα 9: Arduino Fio
 Εικόνα 10: Arduino Pro	 Εικόνα 11: Arduino BT	 Εικόνα 12: Arduino Nano

4.Εικόνα Μοντέλα μικροελεγκτών

1.1.4 Χαρακτηριστικά του Arduino Mega 2560

Το Arduino Mega 2560 είναι ένας πίνακας μικροελεγκτών που βασίζεται στο ATmega2560. Όπως φαίνεται στις εικόνες 5 και 6 διαθέτει 54 ψηφιακές ακίδες εισόδου / εξόδου (από τις οποίες 15 μπορούν να χρησιμοποιηθούν ως έξοδο PWM), 16 αναλογικές εισοδοι, 4 UART (σειριακές θύρες υλικού), ταλαντωτής κρυστάλλου 16 MHz, σύνδεση USB, υποδοχή τροφοδοσίας, και ένα κουμπί επαναφοράς. Περιέχει όλα τα απαραίτητα για την υποστήριξη του μικροελεγκτή. Απλά συνδέεται με έναν υπολογιστή με καλώδιο USB ή ενεργοποιείται με προσαρμογέα εναλλασσόμενου ρεύματος ή μπαταρία για να ξεκινήσει. Ο πίνακας Mega 2560 είναι συμβατός με τις



περισσότερες ασπίδες που έχουν σχεδιαστεί για το Uno και τις πρώην κάρτες Duemilanove ή Diecimila.

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

5.Εικόνα

Τεχνικά Χαρακτηριστικά Arduino Mega 2560

Οι ακίδες ρεύματος είναι ως εξής:

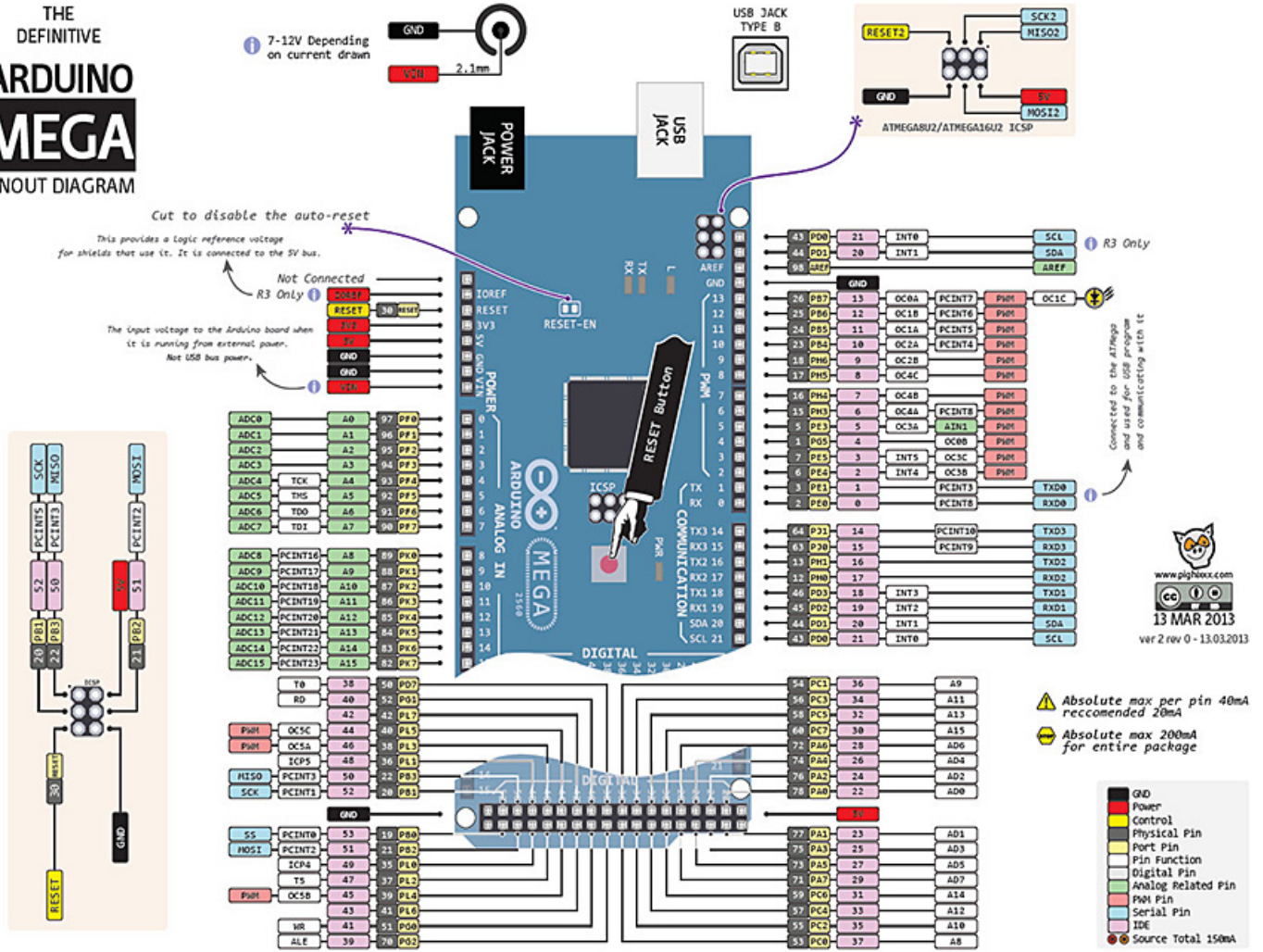
- VIN -> Η τάση εισόδου στην πλακέτα Arduino όταν χρησιμοποιείται μια εξωτερική πηγή ισχύος (σε αντιδιαστολή με 5 βολτ από τη



σύνδεση USB ή άλλες οργανωμένες πηγή ενέργειας). Μπορείτε να παρέχετε τάση μέσω αυτής της υποδοχής.

- 5V -> Η ρυθμιζόμενη παροχή ηλεκτρικού ρεύματος που χρησιμοποιείται για να τροφοδοτήσει το μικροελεγκτή και άλλα συστατικά στην πλακέτα. Αυτό μπορεί να προέλθει είτε από VIN μέσω ενός ρυθμιστή επί της πλακέτας, ή να τροφοδοτείται από USB ή άλλη ρυθμιζόμενη παροχή 5V.
- 3V3 -> Η παροχή 3,3 βολτ που παράγεται από το chip επί της πλακέτας FTDI. Η μέγιστη κατανάλωση ρεύματος είναι 50 mA.
- GND -> Καρφίτσα γείωσης.

THE DEFINITIVE
ARDUINO MEGA
PINOUT DIAGRAM







6. Εικόνα
Arduino Mega 2560 (pin mapping)

2. ΠΕΡΙΦΕΡΕΙΑΚΑ ΥΛΙΚΑ ΚΑΙ SHIELDS

2.1 Arduino Shields

Τα Arduino Shields είναι πλακέτες οι οποίες μπορούν να συνδεθούν πάνω από την προγραμματιζόμενη πλακέτα του Arduino, επεκτείνοντας τις δυνατότητες του. Η έννοια της Shield στην αρχιτεκτονική του Arduino, είναι η ενσωμάτωση επιπλέον υλικού στον μικροελεγκτή που του δίνει καινούριες δυνατότητες, κυρίως σε θέματα επικοινωνίας. Η επικοινωνία από σειριακή μετατρέπεται στην αντίστοιχη που διακρίνει η κάθε Shield, όπως για παράδειγμα μέσω Internet (Ethernet Shield).

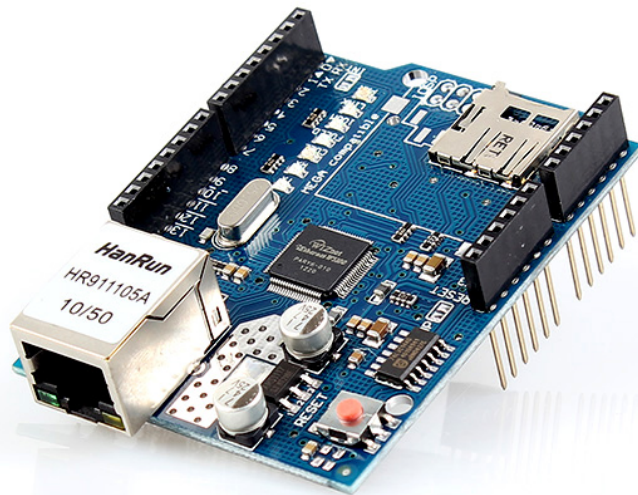
Μοντέλα Arduino Shields	
Μοντέλο:	Περιγραφή:
 Εικόνα 13: Arduino Wifi shield	Επιτυγχάνει ασύρματη επικοινωνία μέσω wifi με το διαδίκτυο. Περιλαμβάνεται θύρα υποδοχής SD.
 Εικόνα 14: Arduino Ethernet shield	Επιτυγχάνει ενσύρματη επικοινωνία με το διαδίκτυο με τη χρήση καλωδίου Ethernet. Απαιτείται η χρήση router (δρομολογητή) για την εισαγωγή του καλωδίου. Περιλαμβάνεται θύρα υποδοχής SD.
 Εικόνα 15: Wireless SD shield	Επιτυγχάνει ασύρματη επικοινωνία με μία προκαθορισμένη ασύρματη μονάδα. Η εμβέλεια της επικοινωνίας ορίζεται έως και 100 πόδια σε εσωτερικούς χώρους ή έως και 300 πόδια σε εξωτερικού. Περιλαμβάνεται θύρα υποδοχής SD.
 Εικόνα 16: Arduino GPRS shield	Επιτυγχάνει επικοινωνία με τη χρήση GSM δικτύου. Απαιτείται η χρήση GSM κάρτας από κάποιον τηλεφωνικό πάροχο κινητής τηλεφωνίας.[8]

7.Εικόνα

Μοντέλα Arduino Shields

Οι διαφορετικές Shields ακολουθούν την ίδια φιλοσοφία με αυτή της αρχικής εργαλειοθήκης, οι οποίες είναι φθηνές να παραχθούν. Λαμβάνουν όλη την πολυπλοκότητα του υλικού και την μειώνουν σε μια απλή διεπαφή. Αυτό επιτρέπει στο χρήστη να υλοποιήσει γρήγορα την ιδέα του.

Υπάρχουν Shields για πολλά διαφορετικά είδη λειτουργιών όπως βλέπουμε και στην εικόνα 7 και 8 :WIFI Shields, Ethernet Shields, Wireless SD Shields,Bluetooth Shields,GPRS Shields, Motor Shields, Protection Shields.



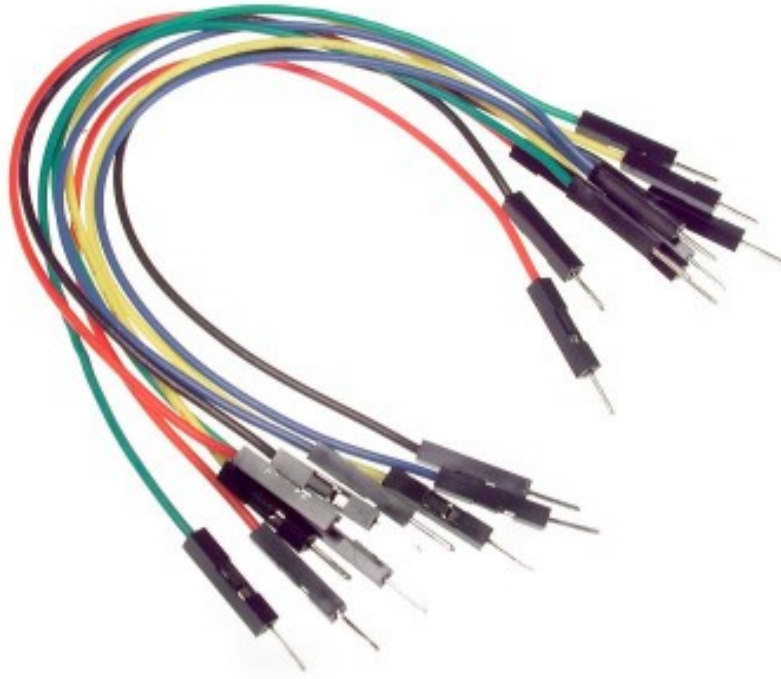
8. Εικόνα
Arduino Ethernet Shield

2.2 Περιφερειακά υλικά

Τα περιφερειακά υλικά είναι απαραίτητη προϋπόθεση για την σύνδεση του μικροελεγκτή Arduino και του Breadboard για την ομαλή λειτουργία και την περάτωση κάποιου project. Για την προσομοίωση και μελέτη της συγκεκριμένης διπλωματικής εργασίας με στόχο τους αυτοματισμούς κάποιων λειτουργιών μέσα και έξω από το σπίτι αλλά και η διαχείριση τους ασύρματα με τη δημιουργία ενός application, με στόχο την δημιουργία ενός έξυπνου σπιτιού αναλύονται παρακάτω :

2.2.1 Wires

Τα καλώδια όπως φαίνονται στην εικόνα 9 χρησιμοποιούνται για τη δημιουργία συνδέσεων μεταξύ των αντικειμένων του Breadboard και των head pins του Arduino. Χρησιμοποιούνται για να τη δημιουργία όλων των κυκλωμάτων της εργασίας.

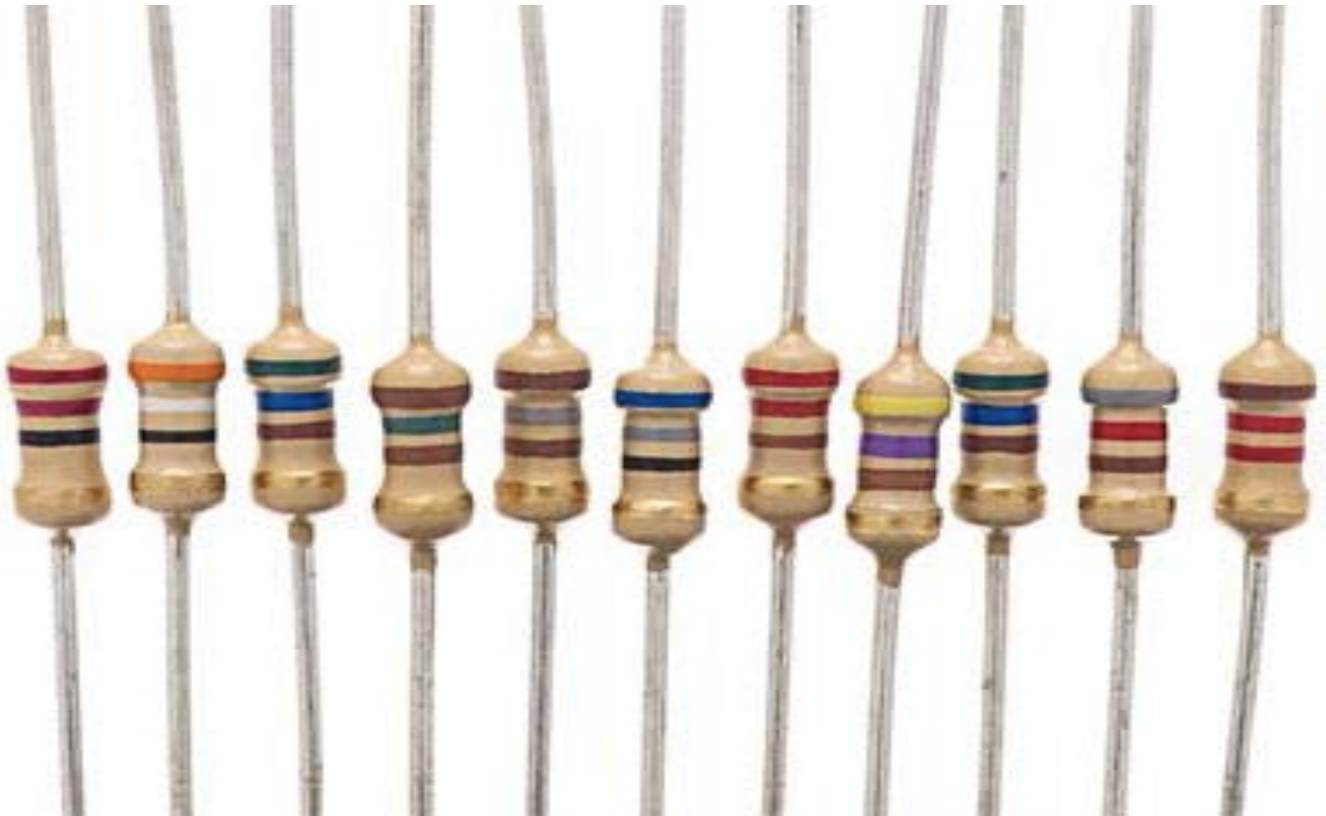


9.Εικόνα
Jumper Wires

2.2.2 Resistors

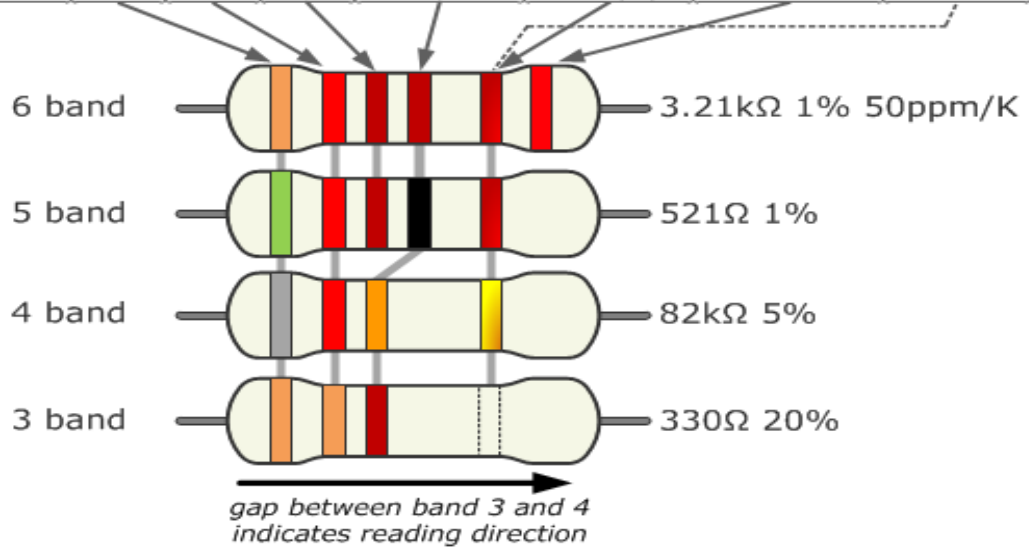
Όπως υποδηλώνει το όνομα(resist), οι αντιστάσεις αντιστέκονται στη ροή ηλεκτρισμού και όσο υψηλότερη είναι η τιμή της αντίστασης, τόσο περισσότερο αντιστέκεται και τόσο λιγότερο ηλεκτρικό ρεύμα θα ρέει μέσα από αυτό. Ο κώδικας χρώματος αντίστασης διαχωρίζει τι είδος αντίστασης είναι.Παρακάτω υπάρχουν τα είδη των χρωμάτων που αντικατοπτρίζουν έναν αριθμό για μια αντίσταση όπως παρατηρούμε και στις εικόνες 10 και 11 :

- Μαύρο 0
- Καφέ 1
- Κόκκινο 2
- Πορτοκαλί 3
- Κίτρινο 4
- Πράσινο 5
- Μπλε 6
- Μωβ 7
- Γκρι 8
- Λευκό 9



10. Εικόνα
Resistors

	Color	Significant figures			Multiply	Tolerance (%)	Temp. Coeff. (ppm/K)	Fail Rate (%)
		1st	2nd	3rd				
Bad	black	0	0	0	x 1		250 (U)	
Beer	brown	1	1	1	x 10	1 (F)	100 (S)	1
Rots	red	2	2	2	x 100	2 (G)	50 (R)	0.1
Our	orange	3	3	3	x 1K		15 (P)	0.01
Young	yellow	4	4	4	x 10K		25 (Q)	0.001
Guts	green	5	5	5	x 100K	0.5 (D)	20 (Z)	
But	blue	6	6	6	x 1M	0.25 (C)	10 (Z)	
Vodka	violet	7	7	7	x 10M	0.1 (B)	5 (M)	
Goes	grey	8	8	8	x 100M	0.05 (A)	1(K)	
Well	white	9	9	9	x 1G			
Get	gold			3th digit only for 5 and 6 bands	x 0.1	5 (J)		
Some	silver				x 0.01	10 (K)		
Now!	none					20 (M)		



11. Εικόνα

Colours of resistor

2.2.3 Buzzer

Ο βομβητής (Εικόνα 12) συνδέεται με τη γείωση και με ένα Pin ελέγχου του Arduino. Όταν ο διαχειριστής επιθυμεί να ενημερωθεί, ο βομβητής ενεργοποιείται μέσω του κώδικα και αρχίζει να εκπέμπει ήχο.



12. Εικόνα

Buzzer

2.2.4 Leds

Για την προσομοίωση της εργασίας οι δοκιμές έγιναν σε led λαμπτήρες οι οποίοι αποτελούνται από 2 άκρα, το μικρό πόδι είναι το αρνητικό ενώ το πιο μακρύ το θετικό, όπως φαίνεται στην εικόνα 13.Θα τους χρησιμοποιηθούν για ορισμένους αυτοματισμούς μέσα και έξω από το σπίτι.



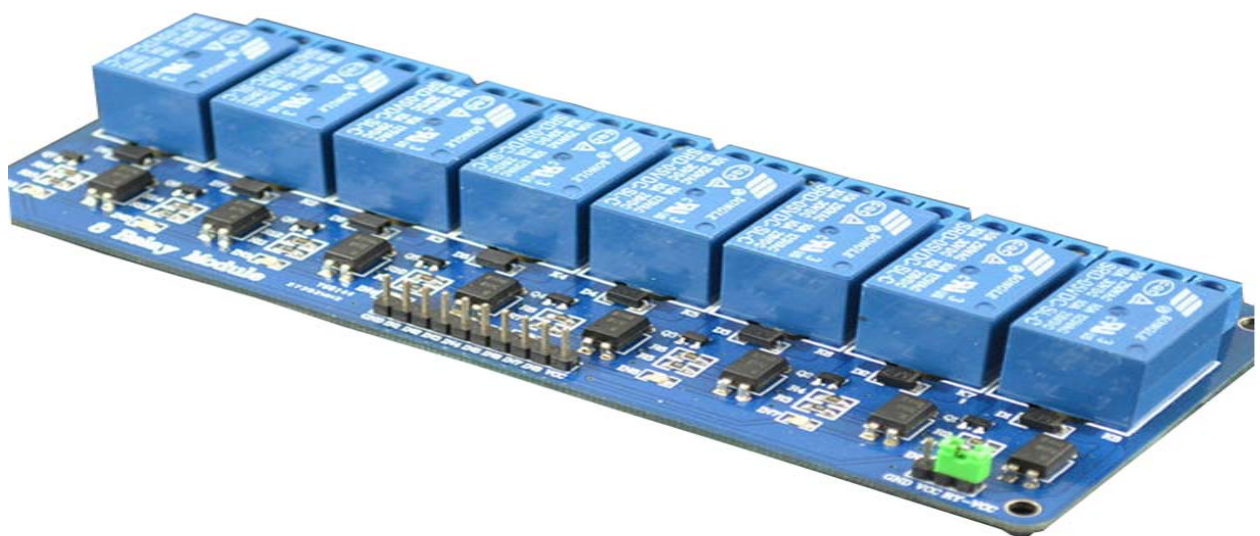
13. Εικόνα

Leds

2.2.5 Relay 8 channel

Με το Relay(Εικόνα 14) μπορούμε να ελέγξουμε τις ηλεκτρονικές συσκευές υψηλής τάσης χρησιμοποιώντας ρελέ. Ένα ρελέ είναι στην πραγματικότητα ένας διακόπτης που λειτουργεί ηλεκτρικά από έναν ηλεκτρομαγνήτη. Ο ηλεκτρομαγνήτης ενεργοποιείται με χαμηλή τάση, για παράδειγμα 5 βολτ από έναν μικροελεγκτή και τραβάει μια επαφή για να κάνει ή να σπάσει ένα κύκλωμα υψηλής τάσης.

Στην συγκεκριμένη διπλωματική χρειάζεται ένας ηλεκτρονόμος (relay) 8 καναλιών με τα οποία θα είναι συνδεδεμένα οι λαμπτήρες, το buzzer, fan. στα οποία θα στέλνονται εντολές από τον προγραμματισμένο μικροελεγκτή για να ενεργοποιούνται ή να απενεργοποιούνται οι αντίστοιχες συσκευές.



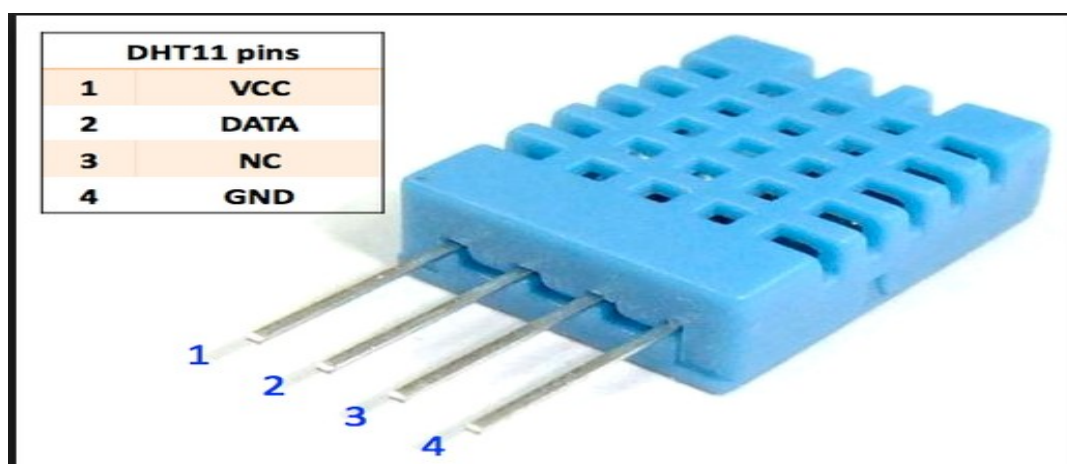
14. Εικόνα

Relay 8 channels

2.2.6 Sensors

Αισθητήρας θερμοκρασίας και υγρασίας

- Ο αισθητήρας θερμοκρασίας(Εικόνα 15) και υγρασίας DFRobot DHT11 διαθέτει αισθητήρα θερμοκρασίας και υγρασίας με βαθμονομημένο ψηφιακό σήμα έξοδο. Χρησιμοποιώντας μια τέτοια τεχνική ψηφιακού σήματος με τεχνολογία ανίχνευσης θερμοκρασίας & υγρασίας, εξασφαλίζεται υψηλή αξιοπιστία και εξαιρετική μακροπρόθεσμη σταθερότητα. Αυτός ο αισθητήρας περιλαμβάνει μια μέτρηση υγρασίας αντιστατικού τύπου και ένα στοιχείο μέτρησης θερμοκρασίας NTC η οποία προσφέρει άριστη ποιότητα και γρήγορη απόκριση.

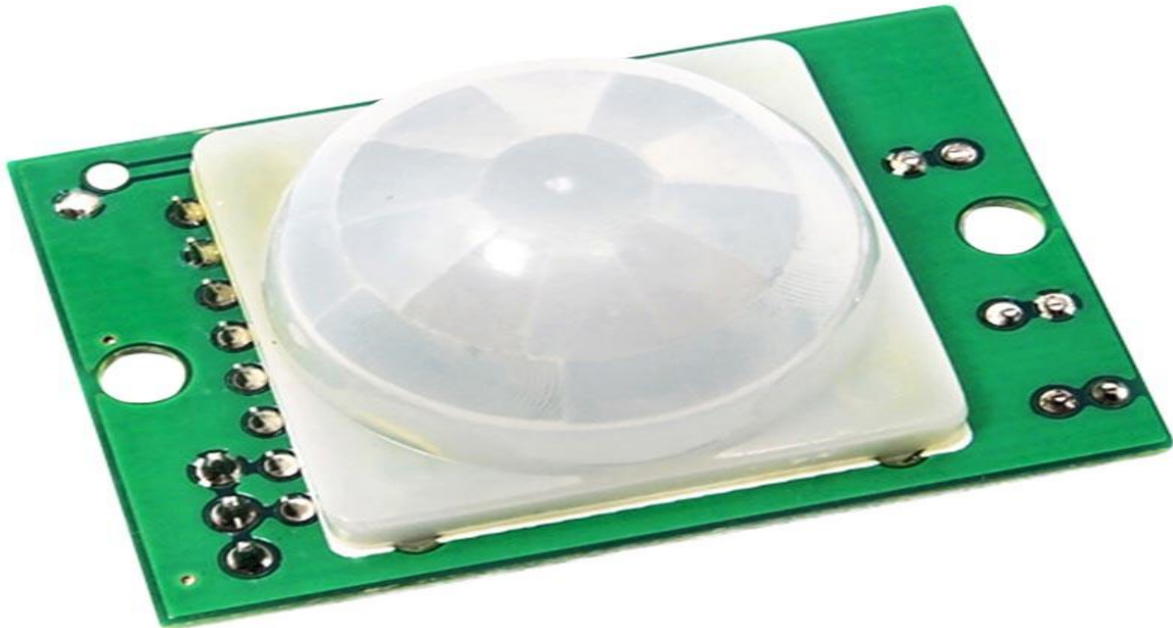


15.Εικόνα
DFRobot DHT11

Αισθητήρας κίνησης

Ο αισθητήρας κίνησης PIR Motion Sensor (Εικόνα 16) είναι ένας απλός αισθητήρας. Ενεργοποιείται και περιμένει 1-2 δευτερόλεπτα ώστε να πάρει ένα στιγμιότυπο του δωματίου. Αν κάτι μετακινηθεί μετά από αυτή την περίοδο, ο ακροδέκτης «συναγερμού» θα μειωθεί. Αυτή η μονάδα λειτουργεί πολύ από 5 έως 12V (το φύλλο δεδομένων δείχνει 12V).

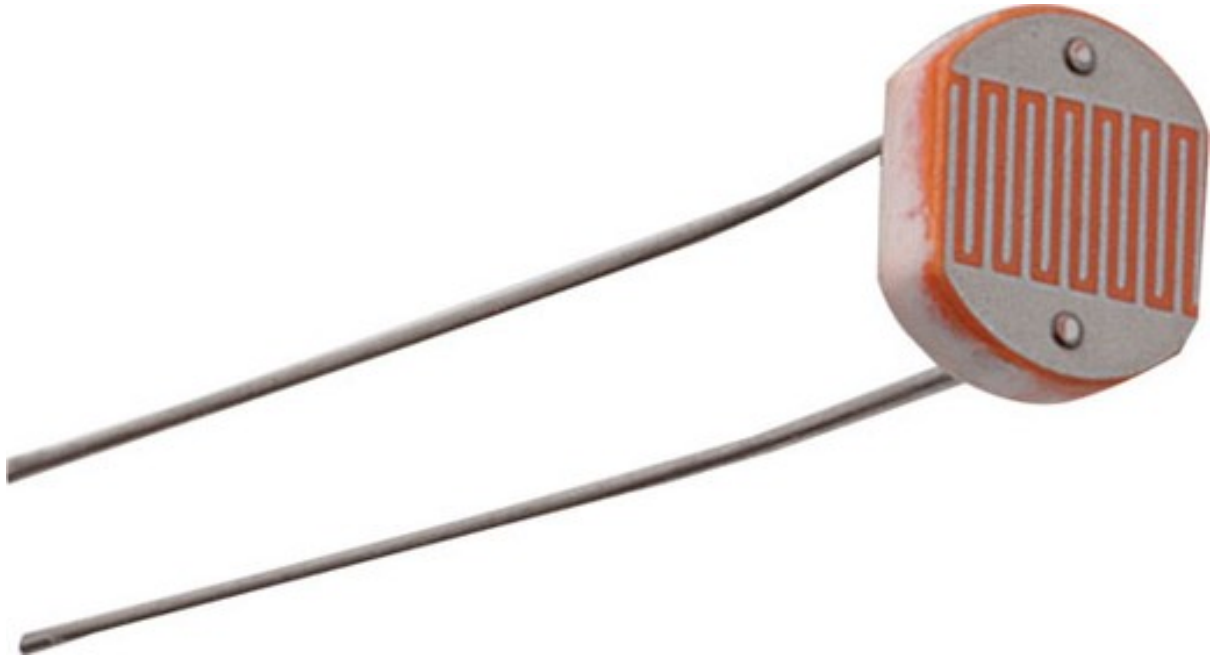
Ο ακροδέκτης συναγερμού είναι ένας ανοιχτός συλλέκτης που σημαίνει ότι θα χρειαστεί μια αντίσταση pull up (Με αντιστάτη στην οποία ο ακροδέκτης εισόδου θα διαβάσει μια υψηλή κατάσταση όταν ενεργοποιηθεί. Με άλλα λόγια, μια μικρή ποσότητα ρεύματος ρέει ανάμεσα στο VCC και τον ακροδέκτη εισόδου (όχι στη γείωση), οπότε το pin εισόδου διαβάζει κοντά στο VCC. Όταν ενεργοποιηθεί ο αισθητήρας, συνδέει τον ακροδέκτη εισόδου απευθείας στη γείωση.) προς τον ακροδέκτη συναγερμού. Η εγκατάσταση του αισθητήρα κίνησης επιτρέπει την σύνδεση πολλαπλών αισθητήρων κίνησης σε έναν μοναδικό pin εισόδου.



16. Εικόνα
PIR Motion Sensor

Αισθητήρας φωτός

Ο αισθητήρας φωτός η αλλιώς photoresistor ή light-dependent resistor(LDR) ή photocell(Εικόνα 17) είναι μία μεταβλητή αντίσταση, ελεγχόμενη από το φως. Η αντίσταση του αισθητήρα μειώνετε με αυξανόμενη ένταση του φωτός, με άλλα λόγια είναι αγωγιμο στο φως.



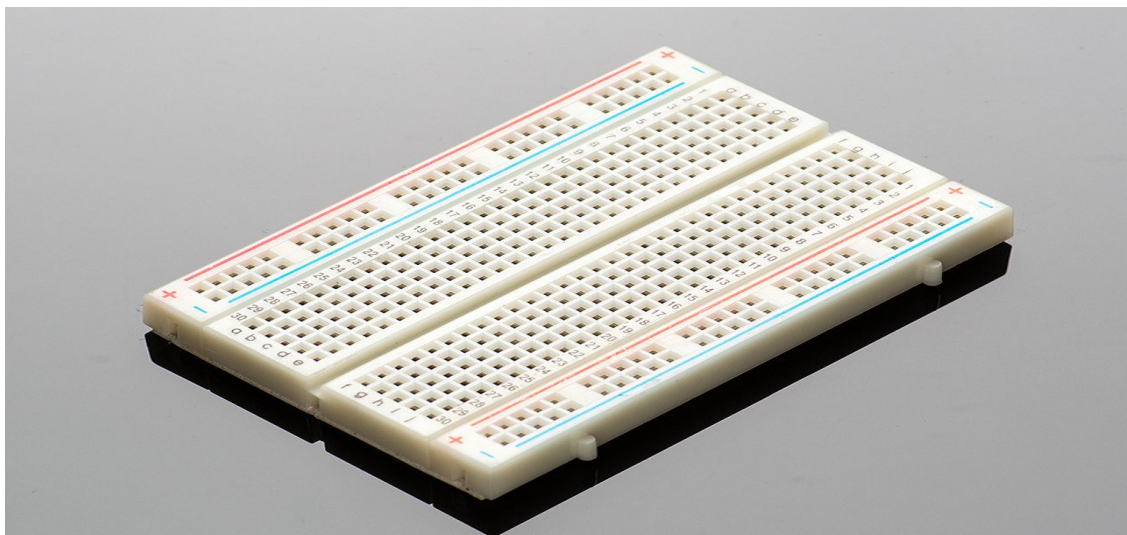
17.Εικόνα

light-dependent resistor(LDR)

Ο αισθητήρας φωτός είναι φτιαγμένος από υψηλής αντίστασης ημιαγωγό. Στο σκοτάδι ο αισθητήρας μπορεί να έχει αντίσταση της τάξης των megaohms ($M\Omega$), σε αντίθεση με συνθήκες φωτεινότητας όπου ο αισθητήρας έχει αντίσταση της τάξης των λίγων εκατοντάδων ohms. Γενικά ο αισθητήρας είναι πολύ μικρός σε μέγεθος, έχει γρήγορη απόκριση και μεγάλη ευαισθησία.

2.2.7 Breadboard

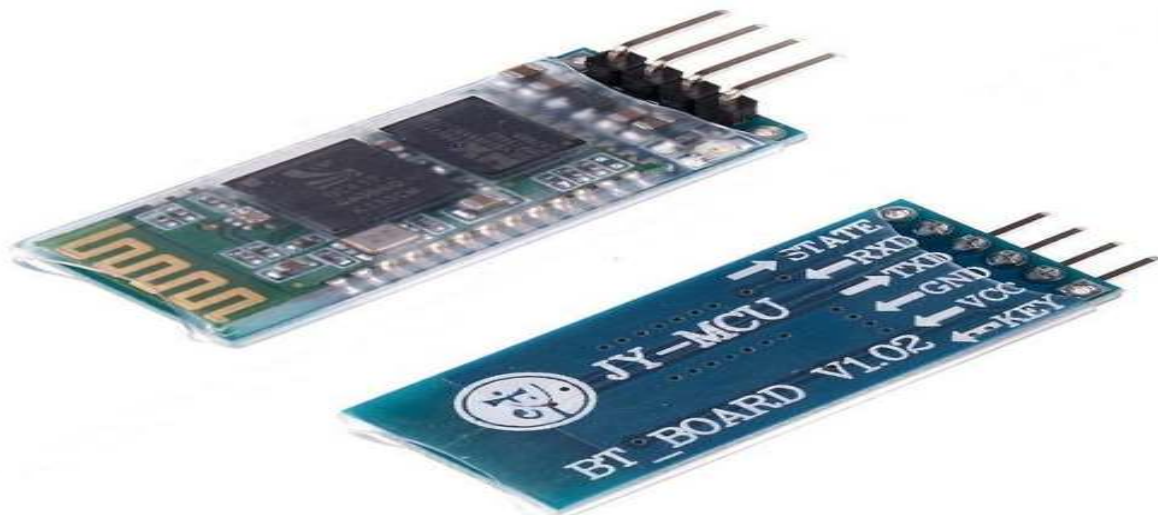
Ένα breadboard(Εικόνα 18) είναι μια βάση κατασκευής για την παραγωγή πρωτότυπων ηλεκτρονικών. Στη δεκαετία του '70 έγινε διαθέσιμη η συγκολλητική πλάκα (plugboard, μια πλακέτα πίνακα τερματικών) και σήμερα ο όρος "breadboard" χρησιμοποιείται συνήθως για να αναφέρεται σε αυτά. Επειδή το συγκολλητικό πανί δεν απαιτεί συγκόλληση, είναι επαναχρησιμοποιήσιμο. Αυτό το καθιστά εύκολο στη χρήση για τη δημιουργία προσωρινών πρωτοτύπων για την πειραματική σχεδίαση κυκλωμάτων. Για το λόγο αυτό, είναι εξαιρετικά δημοφιλείς στους μαθητές και στην τεχνολογική εκπαίδευση. Οι παλαιότεροι τύποι breadboard δεν είχαν αυτήν την ιδιότητα.



**18. Εικόνα
Breadboard**

2.2.8 Bluetooth HC-05

Η μονάδα HC-05 (Εικόνα 19) είναι μια εύκολη στη χρήση μονάδα SPP Bluetooth (Πρωτόκολλο σειριακής θύρας), σχεδιασμένη για διαφανή ρύθμιση ασύρματης σειριακής σύνδεσης. Το Bluetooth της σειριακής θύρας είναι πλήρως πιστοποιημένο με τεχνολογία Bluetooth V2.0 + EDR (Ενισχυμένη ταχύτητα δεδομένων) 3Mbps. Διαμορφωμένο με πλήρη ασύρματο πομποδέκτη 2.4GHz και ζώνη βάσης. Χρησιμοποιεί το CSR Bluecore 04. Διαθέτει εξωτερικό σύστημα Bluetooth με τεχνολογία CMOS και με AFH (Frequency Hopping Feature). Έχει το αποτύπωμα τόσο μικρό όσο 12,7mm x 27mm.



19. Εικόνα
Bluetooth HC-05

3. ΥΛΟΠΟΙΗΣΗ ΚΩΔΙΚΑ

3.1 Πλατφόρμα ανάπτυξης λογισμικού Arduino IDE

Στην ενότητα αυτή θα αναπτυχθεί ο προγραμματισμός του Arduino ο οποίος χωρίζεται σε τέσσερα στάδια: η δημιουργία των συναρτήσεων για την αυτοματοποίηση κάθε συσκευής, η δημιουργία συναρτήσεων για τη σύνδεση του μικροελεγκτή με την εφαρμογή στο κινητό μέσω Bluetooth HC-05, η αποστολή των δεδομένων στην εφαρμογή και η εμφάνιση των δεδομένων της θερμοκρασίας και της υγρασίας του εσωτερικού του “έξυπνου σπιτιού” στην εφαρμογή.

Το σύστημα τίθεται σε δύο καταστάσεις: στην αδρανή και στην ενεργοποίηση των περιφερειακών. Αν υπάρχει διαθέσιμο δεδομένο για τη συγκεκριμένη συσκευή αυτό εμφανίζεται στην οθόνη του κινητού ενώ παράλληλα, μπορεί να ενεργοποιείται το Led και το Buzzer αν ο ένας αισθητήρας κίνησης ενεργοποιηθεί (Alarm), ενώ ο δεύτερος αισθητήρας κίνησης αν ενεργοποιηθεί ανάβει ένα άλλο Led που εμείς έχουμε ορίσει.

Στην αδρανή κατάσταση τίθεται το Arduino όταν δεν υπάρχει είτε κάποια εντολή από τον χρήστη που κατέχει το κινητό είτε αν κανένας αισθητήρας δεν τεθεί σε λειτουργία.



Τα Arduino προγράμματα είναι γραμμένα σε C ή C++. Το Arduino IDE έρχεται με μια βιβλιοθήκη λογισμικού που ονομάζεται "Wiring", από το πρωτότυπο σχέδιο Wiring, γεγονός που καθιστά πολλές κοινές λειτουργίες εισόδου/εξόδου πολύ πιο εύκολες. Οι χρήστες πρέπει μόνο να ορίσουν δύο λειτουργίες για να κάνουν ένα πρόγραμμα κυκλικής εκτέλεσης:

-setup():μία συνάρτηση που τρέχει μία φορά στην αρχή του προγράμματος η οποία αρχικοποιεί τις ρυθμίσεις

-loop():μία συνάρτηση που καλείται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί

Ένα τυπικό πρώτο πρόγραμμα για έναν μικροελεγκτή αναβοσβήνει απλά ένα LED. Στο περιβάλλον του Arduino, ο χρήστης μπορεί να γράψει ένα πρόγραμμα σαν αυτό:

```
#define LED_PIN 13  
void setup () {  
  pinMode (LED_PIN, OUTPUT); // enable pin 13 for digital output  
}  
void loop () {  
  digitalWrite (LED_PIN, HIGH); // turn on the LED  
  delay (1000); // wait one second (1000 milliseconds)  
  digitalWrite (LED_PIN, LOW); // turn off the LED  
  delay (1000); // wait one second  
}
```

3.1.1 Συνάρτηση θερμοκρασίας/υγρασίας

Στη συνέχεια θα αναλυθεί το πρώτο στάδιο του προγραμματισμού του μικροελεγκτή Arduino, το οποίο είναι η δημιουργία των συναρτήσεων για την αυτοματοποίηση της κάθε συσκευής.

Στο πρώτο κομμάτι του προγράμματος υλοποιείται ο κώδικας για τον έλεγχο της θερμοκρασίας και της υγρασίας μέσα στο σπίτι με έναν ειδικό αισθητήρα(DHT11) όπως φαίνεται στην εικόνα 20.

```
/*-----TEMPERATURE_HUMIDITY-----*/  
  
void temperatureNhumidity(){  
  
    float h = dht.readHumidity();  
    float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit  
    analogRead(h);  
    analogRead(t);  
  
    //Blynk.virtualWrite(V5, h);  
    //Blynk.virtualWrite(V6, t);  
  
    Serial.print("Temperature = ");  
    Serial.print(t);  
    Serial.println(" C");  
    Serial.print("Humidity = ");  
    Serial.print(h);  
    Serial.print(" %");  
    Serial.println("");  
    delay(400);  
  
}
```

20. Εικόνα



Για να λειτουργήσει ο συγκεκριμένος αισθητήρας σωστά χρησιμοποιήθηκε η βιβλιοθήκη DHT.h όπως φαίνεται στην εικόνα 21.

```
/*-----INCLUDE FOR TEMP AND HUMIDITY-----*/  
  
#include <DHT.h>  
#define DHTPIN 5 // What digital pin we're connected to (D5)  
#define DHTTYPE DHT11 // DHT 11  
  
DHT dht(DHTPIN, DHTTYPE);  
  
/*-----*/
```

21. Εικόνα

Με τη βοήθεια αυτού του αισθητήρα, ένα σπίτι μπορεί να εξοικονομεί καθημερινά μεγάλο ποσοστό ηλεκτρικής ενέργειας και αυτό συμβαίνει για τον εξής λόγο. Γενικά, η κατανάλωση της ενέργειας που θα έχει μια κλιματιστική μονάδα για να ψύξει ή να θερμάνει μία οικία μία ή δύο φορές τη μέρα θα είναι πολύ μεγαλύτερη από την κατανάλωση που θα έχει αν λειτουργούσε κάθε φορά που θα χρειαζόταν, ώστε να κρατάει μια σταθερή θερμοκρασία στον εσωτερικό χώρο του σπιτιού. Αυτό συμβαίνει γιατί για παράδειγμα για να αυξηθεί η θερμοκρασία από τους 10C στους 22C χρειάζεται να καταναλωθεί πολύ μεγαλύτερο ποσοστό ενέργειας από το να θερμανθεί από τους 20C στους 22C. Αυτό οπωσδήποτε προϋποθέτει η οικία να είναι πολύ καλά μονωμένη ώστε να μην χρειάζεται να λειτουργεί συνεχώς η κλιματιστική μονάδα.

Με τον αισθητήρα θερμοκρασίας και με έναν ειδικό προγραμματισμό για την αυτοματοποίηση της κλιματιστικής μονάδας όταν αυτή χρειάζεται να ενεργοποιηθεί παρατηρείται ότι ανοίγει ένας νέος δρόμος προς την εξοικονόμηση ηλεκτρικής ενέργειας με απλούς τρόπους.

3.1.2 Συνάρτηση κίνησης(φως)

Στη συνέχεια υλοποιήθηκε μία συνάρτηση για ένα αυτοματοποιημένο σύστημα φωτισμού (Εικόνα 22), το οποίο όπως θα αναλύσουμε και στη συνέχεια προσφέρει με πολύ απλό τρόπο μικρότερη κατανάλωση ηλεκτρικής ενέργειας.

```
/*-----MOTION_LIGHT-----*/  
void MotionLight(){  
  
    digitalWrite (ledMotion,LOW);  
    delay(1000); //this delay is to let the sensor settle down before taking a reading  
    int sensor_1 = digitalRead(motionLight);  
  
    if (sensor_1 == LOW){  
  
        digitalWrite(ledMotion,LOW);  
  
    }  
    else{  
  
        digitalWrite(ledMotion,HIGH);  
        delay(2000);  
  
    }  
}
```

22. Εικόνα

3.1.3 Συνάρτηση αισθητήρα φωτός

Εκτός από την εξοικονόμηση ενέργειας σε ένα σπίτι αναγκαία είναι και η ασφάλεια των ανθρώπων η οποία στις εποχές αυτές παρατηρείται ότι εγκληματικότητα και η διαρρήξεις είναι από τα κύρια προβλήματα και ανησυχίες ενός ανθρώπου που διαμένει σε μια οικία. Για τον λόγο αυτό υλοποιήθηκαν 2 συναρτήσεις, οι οποίες όπως φαίνονται παρακάτω έχουν την δυνατότητα να δημιουργήσουν ένα αίσθημα ασφάλειας στον ένοικο.

Η πρώτη συνάρτηση (Εικόνα 23) απευθύνεται στην αυτόματη ενεργοποίηση ή απενεργοποίηση των λαμπτήρων στους εξωτερικούς χώρους του “έξυπνου σπιτιού” όταν νυχτώνει (ανάβουν) ή όταν ξημερώνει (σβήνουν). Αυτό πραγματοποιήθηκε με έναν αισθητήρα φωτός, ο οποίος είναι φτιαγμένος από υψηλής αντίστασης ημιαγωγό.

```
/*-----LEDOUT TURN OFF ON DAY AND TURN ON AT NIGHT-----*/  
void dayNnight(){  
  
    int ldrStatusOut = analogRead(sensorLight); //read the status of the LDR value  
  
    if (ldrStatusOut <=300) {  
        analogWrite(ledLight, HIGH);  
    }  
    else{  
        analogWrite(ledLight, LOW);  
    }  
}
```

23. Εικόνα

3.1.4 Συνάρτηση κίνησης(συναγερμός)

Η δεύτερη συνάρτηση(Εικόνα 24) είναι προγραμματισμένη να ενεργοποιεί ή να απενεργοποιεί αυτόματα τον συναγερμό μια κατοικίας μαζί με κάποια φώτα.Αυτό πραγματοποιείται με τον προγραμματισμο του συγκεκριμένου αισθητήρα κίνησης ο οποίος όταν “καταλάβει” κάποια κίνηση στον χώρο όπου είναι τοποθετημένος τότε ενεργοποιεί τον συναγερμό.

```
/*-----MOTION_ALARM-----*/  
void MotionAlarm(){  
  
    digitalWrite (beep,LOW);  
    delay(1000); //this delay is to let the sensor settle down before taking a reading  
    int sensor_2 = digitalRead(motionAlarm);  
  
    if (sensor_2 == LOW){  
  
        noTone(beep);  
        digitalWrite(ledAlarm,LOW);  
        // Serial.println("ALARM DEACTIVATED");  
  
    }  
    else{  
  
        tone(beep,HIGH);  
        digitalWrite(ledAlarm,HIGH);  
        Serial.println("-----ALARM ACTIVATED-----");  
        delay(2000);  
  
    }  
  
}
```

24. Εικόνα

3.2 Εφαρμογή Blynk

Για την εξοικονόμηση ενέργειας σε ένα “έξυπνο” σπίτι δεν αρκούν μόνο κάποιοι αυτοματισμοί, αλλά είναι αναγκαία και η διαχείριση των διάφορων ηλεκτρικών συσκευών απομακρυσμένα, ώστε να μην υπάρχει άσκοπη σπατάλη ενέργειας. Για τον λόγο αυτόν χρησιμοποιήθηκε η εφαρμογή Blynk.

Το Blynk είναι μια πλατφόρμα με εφαρμογές iOS και Android για τον έλεγχο του Arduino ή του Raspberry Pi μέσω του Διαδικτύου. Είναι ένας ψηφιακός πίνακας εργαλείων όπου μπορεί ο



χρήστης να δημιουργήσει μια γραφική διεπαφή για το έργο του, μεταφέροντας απλώς widgets.

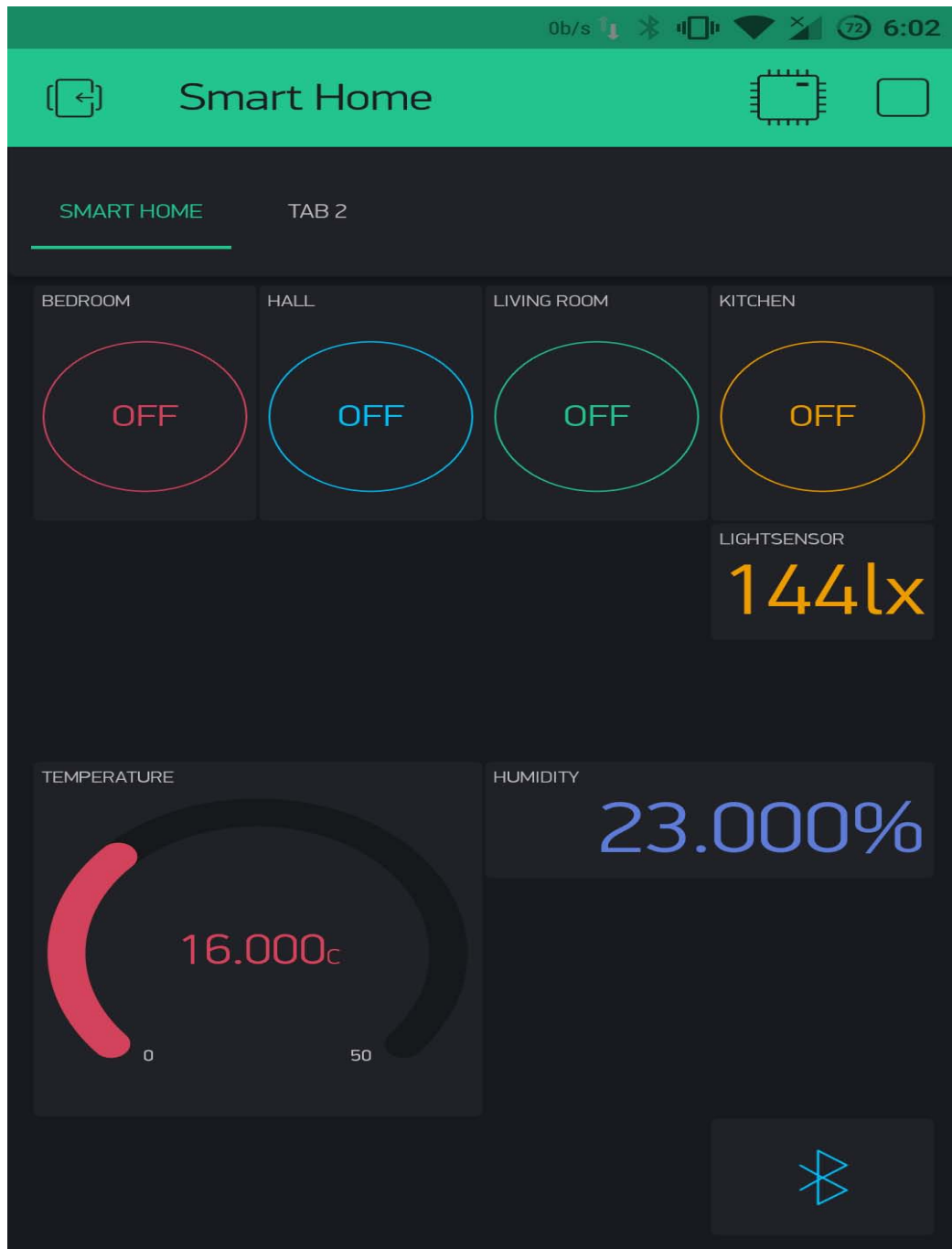
Το Blynk δεν είναι συνδεδεμένο με κάποιο ειδικό breadboard ή shield. Αντ 'αυτού, υποστηρίζει το υλικό της επιλογής του κάθε προγραμματιστή. Είτε το Arduino είτε το Raspberry Pi συνδέονται με το Διαδίκτυο μέσω Bluetooth, Wi-Fi, Ethernet ή ενός νέου τσιπ ESP8266. Στην εικόνα 25 απεικονίζεται η διεπαφή της εφαρμογής Blynk.



25. Εικόνα Blynk

Στην εικόνα 26 υπάρχει η εφαρμογή και απεικονίζονται τα δεδομένα από τα πειράματα που εξελίχθηκαν για την συγκεκριμένη διπλωματική εργασία. Εμφανίζεται η θερμοκρασία, η υγρασία του εσωτερικού του σπιτιού, ο αισθητήρας φωτός ο οποίος μετράει τα την φωτεινότητα έξω από το σπίτι και τέλος

κάποια κουμπιά με τα οποία διαχειριζόμαστε την ενεργοποίηση ή την απενεργοποίηση κάποιων λαμπτήρων μέσα στο σπίτι.



26. Εικόνα

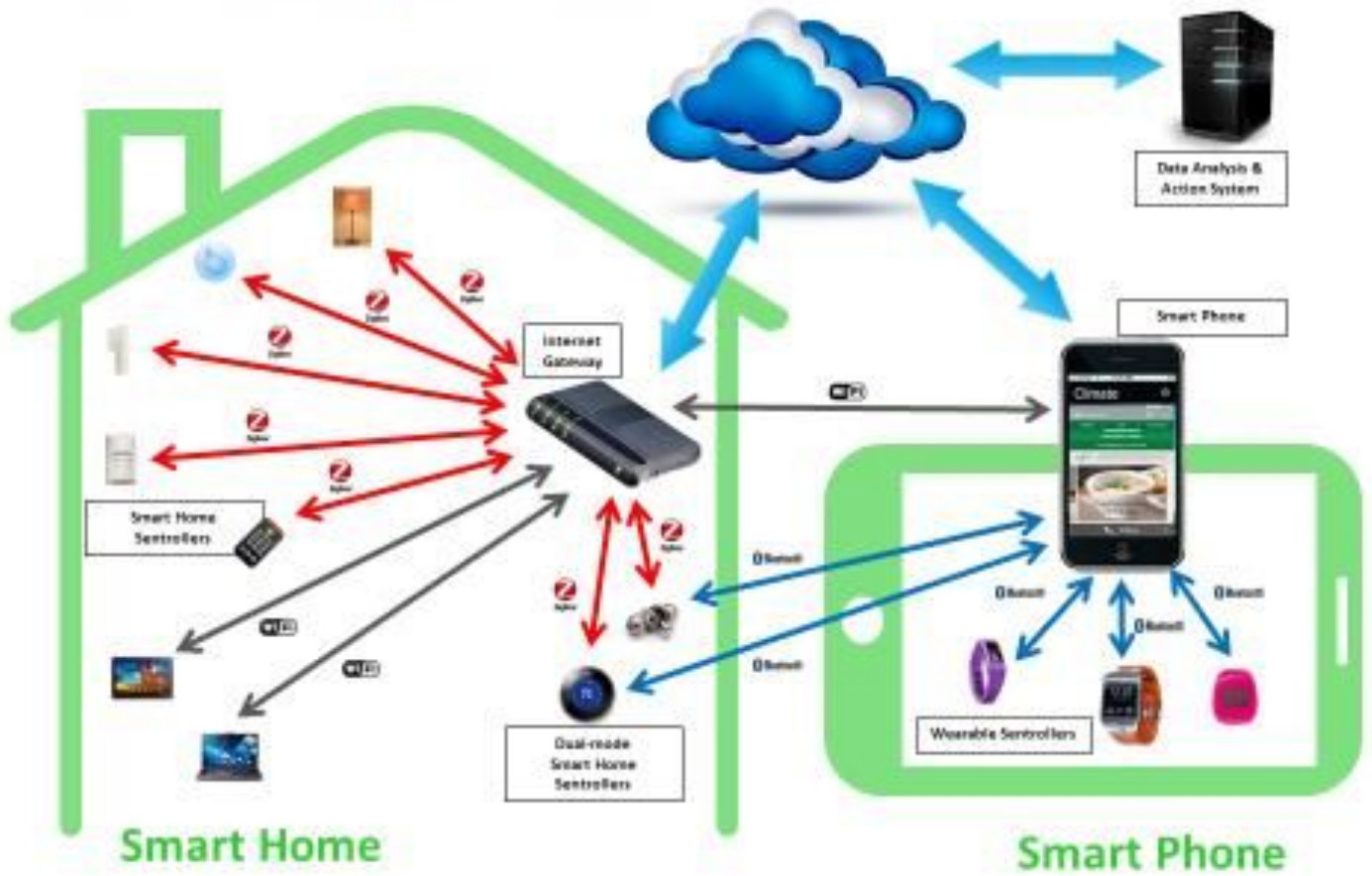
4. ΈΞΥΠΝΟ ΣΠΙΤΙ

4.1 Η εφαρμογή ασύρματων συστημάτων οικιακού αυτοματισμού

Οι εφαρμογές του IoT εξακολουθούν να αποτελούν πρόκληση σε πολλές διαφορετικές βιομηχανίες. Η κατανάλωση ενέργειας παραμένει ένα σημαντικό ζήτημα για τη διαχείριση των συσκευών στα σπίτια, καθώς μπορεί να χρησιμοποιηθεί ανεπιθύμητη πρόσθετη ισχύς σε ανεπιθύμητους χρόνους. Ως αποτέλεσμα, οι πόροι που χρησιμοποιούνται στην παραγωγή ενέργειας δεν χρησιμοποιούνται αποτελεσματικά. Αυτοί οι πόροι μπορούν να χρησιμοποιηθούν σε άλλες παραγωγικές εφαρμογές.

Για την καταπολέμηση την άσκοπης κατανάλωσης ενέργειας είναι σημαντική η διαχείριση των διάφορων ηλεκτρικών συσκευών με τη χρήση της τεχνολογίας και συγκεκριμένα με τον συνδυασμό των προγραμματισμένων μικροελεγκτών και την απεικόνιση των δεδομένων σε μια έξυπνη συσκευή(πχ κινητό). Με το τρόπο αυτό ο χρήστης μπορεί ανά πάσα στιγμή να ελέγχει ακόμα και απομακρυσμένα αν οι συσκευές του σπιτιού του βρίσκονται σε ενεργή ή ανενεργή κατάσταση.Όπως παρατηρείται και στην εικόνα 25, όπου περιγράφεται ένα γενικό σύστημα αυτοματοποιημένου “έξυπνου σπιτιού”.

Small Data System Architecture



27. Εικόνα

Γενικό σύστημα αυτοματοποιημένου “έξυπνου σπιτιού”

4.2 Κόστος διάφορων συστημάτων αυτοματισμού

Το κόστος ενός συστήματος αυτοματισμού στο σπίτι είναι επίσης σημαντικό για τους πελάτες. Κάθε κιτ εκκίνησης περιλαμβάνει το βασικό υλικό και λογισμικό εγκατάστασης που χρειάζεται για την αυτοματοποίηση των συσκευών φωτισμού.

Όπως παρατηρείται, το φτηνότερο είναι το σύστημα που είναι πιο δύσκολο να εγκατασταθεί. Το DomoticHome δεν διαθέτει κιτ εκκίνησης. Για την εγκατάσταση χρειάζεται μια πλακέτα Arduino που συμπληρώνεται με μια μονάδα ethernet η οποία είναι περίπου 120 ευρώ. Ορισμένοι άλλοι πόροι που μπορεί να χρειαστούν (συμπεριλαμβανομένης της καλωδίωσης, των ρελέ κ.λπ.) είναι μάλλον φθηνοί. Επιπλέον, οι τιμές για το Android @ Home δεν είναι ακόμα διαθέσιμες και τα κιτ εκκίνησης Insteon πωλούνται από 75 ευρώ και τα modules από 17,99 ευρώ, αλλά αυτή τη στιγμή η τεχνολογία δεν είναι συμβατή με τις ευρωπαϊκές τάσεις και πρίζες. Οι κατασκευαστές της Insteon ανακοίνωσαν ότι θα ξεκινήσουν τη διανομή συσκευών συμβατών με την EE το καλοκαίρι του 2012. Από την άλλη πλευρά, το X10 είναι το φτηνότερο από όλα, με τιμές για το κιτ εκκίνησης από 57,99 ευρώ και ενότητες από 4,99 ευρώ. Αυτό οφείλεται στο γεγονός ότι η τεχνολογία ήταν περίπου 35 χρόνια και είναι ξεπερασμένη. Ωστόσο, η εγκατάσταση είναι δύσκολη, γι 'αυτό απαιτείται η πρόσληψη τεχνικών - αυτό συνεπάγεται επιπλέον κόστος



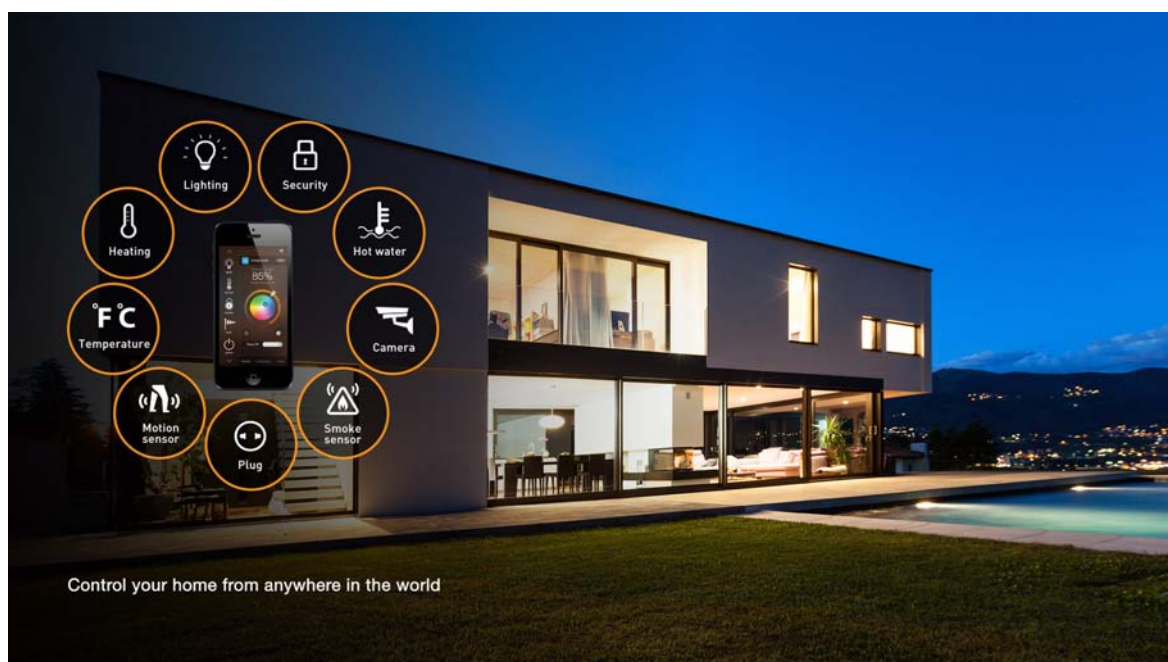
υποστήριξης και διαβούλευσης. Το ίδιο ισχύει και για τα ZigBee και Z-Wave. Ένα κιτ εκκίνησης είναι περίπου 200 ευρώ και οι ενότητες είναι από 21,50 ευρώ. Αυτό αποτελεί σημαντικό μειονέκτημα, δεδομένου ότι ήδη δημιουργεί κάποιες δυσκολίες εγκατάστασης και το κόστος είναι υψηλότερο σε σύγκριση με τους ανταγωνιστές του.

4.3 Τα οφέλη ενός “έξυπνου σπιτιού”

Διανύεται μια εποχή, αντίθεση σε οποιαδήποτε άλλη. Με τόσες πολλές τεχνολογικές εξελίξεις στα χέρια των ανθρώπων, οι ζωές τους γίνονται ευκολότερες, γεγονός που τους χαρακτηρίζει τυχερούς, λαμβάνοντας υπόψη τις ταχείς συνθήκες ζωής. Τα τελευταία χρόνια, έγιναν σπουδαία βήματα για να καταστούν τα σπίτια λίγο περισσότερο "tech savvy". Αυτή η τεχνολογία, εν ολίγοις, επιτρέπει στον ιδιοκτήτη του σπιτιού να ελέγχει το σπίτι του ενώ βρίσκεται μακριά από αυτό, χρησιμοποιώντας απλά ένα Smartphone ή iPad. Στην πραγματικότητα, αυτά τα σπίτια έχουν χαρακτηριστεί ως "έξυπνα", λόγω του γεγονότος ότι μπορούν φαινομενικά να σκέφτονται μόνα τους. Αυτή η δυνατότητα «σκέψης», επίσης γνωστή ως αυτοματοποιημένο σπίτι, βοηθά τους ιδιοκτήτες με διάφορους τρόπους. Ποιοί είναι οι τρόποι με τους οποίους μπορεί να ωφελήσει ένα σύστημα αυτοματισμού σπιτιού;

1) Προσθέτει ασφάλεια μέσω συσκευής και ελέγχου φωτισμού.

Υπάρχει η δυνατότητα να ελέγχονται οι μικρές συσκευές και ο φωτισμός, με την απλή χρήση του δακτύλου στην αγαπημένη τεχνολογική συσκευή. Το σπίτι και η οικογένειά απολαμβάνουν επίσης ένα πρόσθετο μέτρο ασφάλειας μέσω της ικανότητάς τους να ελέγχουν τα φώτα στο σπίτι τους. Όχι μόνο αυτό, επιτρέπει να βεβαιωθεί ο ιδιοκτήτης ότι τα φώτα είναι σβηστά για να εξοικονομήσει ηλεκτρικό ρεύμα, επιτρέπει επίσης την ενεργοποίησή τους σε συγκεκριμένες ώρες, αν θέλει να μοιάζει ότι κάποιος βρίσκεται σπίτι. Αυτό βοηθά επίσης στην αύξηση της ασφάλειας του σπιτιού.



28. Εικόνα



Έλεγχος σπιτιού

2) Αυξάνει την ευκολία μέσω της προσαρμογής της θερμοκρασίας και μειώνει την άσκοπη κατανάλωση ενέργειας.

Συχνά, ο ιδιοκτήτης φεύγει για δουλειά νωρίς το πρωί και ξεχνάει να ρυθμίσει τον θερμοστάτη. Ως αποτέλεσμα, επιστρέφει σε ένα σπίτι που είναι πολύ ζεστό ή πολύ κρύο. Αυτό είναι ενοχλητικό, καθώς συνήθως χρειάζεται αρκετός χρόνος για να αυξηθεί ή να μειωθεί η θερμοκρασία του σπιτιού μετά την προσαρμογή, αλλά και πολύ δαπανηρό. Ωστόσο, με ένα σύστημα αυτοματισμού στο σπίτι, μπορεί απλά να ρυθμιστεί θερμοστάτης απομακρυσμένα λίγες ώρες πριν από την κατεύθυνση του χρήστη προς το σπίτι. Αυτό είναι οικονομικά αποδοτικό και εξοικονομεί ενέργεια.

3) Εξοικονομεί χρήματα και αυξάνει την ευκολία

Όπως αναφέρθηκε προηγουμένως, ένα σύστημα αυτοματισμού ενός σπιτιού εξοικονομεί χρήματα. Η πιο ευεργετική επίδραση που θα έχει το σύστημα είναι στο μηνιαίο λογαριασμό της οικίας. Δεν θα σπαταλούνται πλέον χρήματα για οικιακές συσκευές που ξεχνιούνται ενεργοποιημένες. Θα εξοικονομείται επίσης το κόστος του φυσικού αερίου, καθώς ποτέ δεν θα παραμείνει ανοιχτός ο θερμοστάτης λόγω της αυτοματοποίησης. Αυτό είναι σίγουρα



βολικό. Ο ιδιοκτήτης μπορεί να έχει πλήρη έλεγχο και να βεβαιώνεται ότι το κόστος είναι χαμηλό χωρίς να ασκεί οποιαδήποτε πρόσθετη προσπάθεια.

4.4 Σενάριο

Το σενάριο περιγράφει ένα άτομο που έχει εγκαταστήσει ένα σύστημα αυτοματισμού στο σπίτι του. Κάθε φως στο διαμέρισμα είναι συνδεδεμένο με υλικό που βασίζεται στο Arduino. Αν αφήνει ένα δωμάτιο, μπορεί εύκολα να ενεργοποιήσει / απενεργοποιήσει τα φώτα στην αίθουσα που φεύγει μέσω του έξυπνου τηλεφώνου του.

Ξεκινά μια εφαρμογή Android που είναι συνδεδεμένη στον πίνακα Arduino χρησιμοποιώντας είτε Bluetooth είτε Wi-Fi. Αργότερα πιέζει το κουμπί με τον τίτλο 'Υπνοδωμάτιο' και πατάει το κουμπί 'ON' στην οθόνη, στέλνοντας την εντολή 'Light On' στα φώτα που είναι εγκατεστημένα στο δωμάτιο με την ετικέτα 'Bedroom'. Αργότερα, η εντολή λαμβάνεται από το Arduino board, το οποίο εκτελεί την εντολή ενεργοποιώντας τα φώτα. Επιπλέον, επιλέγει το "Living room" από την οθόνη του κινητού τηλεφώνου και πατάει "OFF" για να απενεργοποιήσει τα φώτα και η εντολή εκτελείται με παρόμοιο τρόπο. Όταν πηγαίνετε στην αίθουσα, μπορεί να θέλετε να ρυθμίσετε τη θερμοκρασία του σπιτιού



πιέζοντας το πλήκτρο 'Θερμοκρασία' στο τηλέφωνό του. Το τηλέφωνο εμφανίζει την τρέχουσα θερμοκρασία που παίρνει από έναν αισθητήρα θερμοκρασίας, ο οποίος είναι συνδεδεμένος στο Arduino.

5. ΣΥΜΠΕΡΑΣΜΑΤΑ

Ο αυτοματισμός στο σπίτι γίνεται όλο και πιο δημοφιλής λόγω των τελευταίων εξελίξεων στον τομέα του υλικού, οι οποίες έχουν μειώσει σημαντικά το κόστος και βελτίωσαν τις δυνατότητες. Αυτό οφείλεται στο γεγονός ότι η τεχνολογία εξελίσσεται και η πρόσβαση στις απαραίτητες πληροφορίες είναι ευκολότερη από ποτέ. Κατά συνέπεια, η ζήτηση για τα συστήματα αυτά αυξάνεται και διάφοροι κατασκευαστές το συνειδητοποίησαν. Αυτό αποτέλεσε το κίνητρο για πολλές έξυπνες προσεγγίσεις στο σπίτι, όπως το DomotiChome της Mattia Lipreris και το εκλεπτυσμένο Insteon. Επιπλέον, τα εργαλεία όπως το Amarino δεν προορίζονται για οικιακή αυτοματοποίηση, αλλά μπορούν εύκολα να προσαρμοστούν ώστε να ταιριάζουν στις απαιτήσεις του έξυπνου περιβάλλοντος.

Η αυτοματοποίηση του σπιτιού είναι εφικτή αυτές τις μέρες. Παρόλο που υπήρχε εδώ και λίγο καιρό, δεν ήταν μια πιθανή επιλογή για πολλούς ανθρώπους λόγω του τεράστιου κόστους της. Ωστόσο, χάρη στην ανάπτυξη τεχνολογιών Android και Arduino, σχεδόν οποιοσδήποτε μπορεί να εφαρμόσει κάποιο είδος αυτοματισμού στο σπίτι του. Μετά την ανάλυση των τελευταίων εξελίξεων, όπως το Zigbee, το Z-Wave, το Android @ Home, το Domotichome, το X10, το Insteon, έχουν επισημανθεί διαφορετικά



κριτήρια απόφασης και έχουν εξάγει τα πλεονεκτήματα και τα μειονεκτήματα κάθε συστήματος.

Παρόλο που οι μικροελεγκτές βρίσκονταν εδώ και πολύ καιρό σε λύσεις οικιακού αυτοματισμού, κανένας από αυτούς δεν ήταν ανοιχτό υλικό και ανοικτού κώδικα. Η εμφάνιση μικροελεγκτών όπως το Arduino ενθαρρύνει την ανάπτυξη λύσεων έξυπνων κατοικιών. Με τις πρόσθετες μονάδες Arduino δίνονται απεριόριστες ευκαιρίες σύνδεσης και διαμόρφωσης διαφορετικών συσκευών στο εκάστοτε σπίτι. Με την εφαρμογή των ασύρματων συστημάτων, κατανοήθηκε πώς μπορεί να προσαρμοστεί το Arduino σε ένα έξυπνο οικιακό περιβάλλον.

Επιπλέον, υπάρχουν πολλές λύσεις στην αγορά. Η συμβολή της διπλωματικής εργασίας περιλαμβάνει την ανάλυση διαφόρων λύσεων για την επισήμανση της αδυναμίας και των δυνατοτήτων, ώστε να μπορεί να επιλεγεί μια κατάλληλη λύση όσον αφορά τη δυσκολία εγκατάστασης, την αξιοπιστία, τον τύπο επικοινωνίας, την επεκτασιμότητα και το κόστος.

Τέλος, θεωρείται ότι η επιλογή του συστήματος αυτοματισμού εξαρτάται από το σενάριο και τις απαιτήσεις. Σε μακροπρόθεσμες λύσεις οικιακού αυτοματισμού μπορεί να συμβάλει στη μείωση του κόστους, να ενισχύσει τη συγκέντρωση των συσκευών σε μια ενιαία μονάδα ελέγχου και να βοηθήσει στη μείωση του αποτυπώματος άνθρακα με την έξυπνη χρήση των πόρων.



“ USE LESS SAVE MORE ”



ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

1.Έξυπνο σπίτι : Ένα σπίτι με νοημοσύνη που σκέπτεται και ενεργεί για εσάς βάση των καθημερινών σας αναγκών και συνηθειών.

2.GPS : Global Positioning System

3.Wiring : a system of wires providing electric circuits for a device or building

4.IDII : Interaction Design Institute Ivrea

5.PCB : Printed circuit board

6.PWM : Pulse Width Modulation

7.UART : Universal asynchronous receiver-transmitter

8.NTC : Negative Temperature Coefficient

9.VCC : VCC is a positive-voltage supply and the collector terminal of bipolar transistors is connected to the VCC supply or to a load which connects to VCC. VSS connects to the source terminal of a FET

10.LDR : Light Dependent Resistor

11.AFH : Frequency Hopping Feature

12. tech-savvy : Proficient in the use of technology, especially computers.



ΠΑΡΑΡΤΗΜΑ 1

Βιβλιοθήκη DHT-11

```
/* DHT library

MIT license
written by Adafruit Industries
*/
#ifndef DHT_H
#define DHT_H

#if ARDUINO >= 100
  #include "Arduino.h"
#else
  #include "WProgram.h"
#endif

// Uncomment to enable printing out nice debug messages.
// #define DHT_DEBUG

// Define where debug output will be printed.
#define DEBUG_PRINTER Serial

// Setup debug printing macros.
#ifdef DHT_DEBUG
  #define DEBUG_PRINT(...) { DEBUG_PRINTER.print(__VA_ARGS__); }
  #define DEBUG_PRINTLN(...) {
DEBUG_PRINTER.println(__VA_ARGS__); }
#else
  #define DEBUG_PRINT(...) {}
  #define DEBUG_PRINTLN(...) {}
#endif

// Define types of sensors.
#define DHT11 11
#define DHT22 22
#define DHT21 21
#define AM2301 21

class DHT {
public:
  DHT(uint8_t pin, uint8_t type, uint8_t count=6);
  void begin(void);
  float readTemperature(bool S=false, bool force=false);
  float convertCtoF(float);
  float convertFtoC(float);
};
```



```
float computeHeatIndex(float temperature, float
percentHumidity, bool isFahrenheit=true);
float readHumidity(bool force=false);
boolean read(bool force=false);

private:
uint8_t data[5];
uint8_t _pin, _type;
#ifdef __AVR
// Use direct GPIO access on an 8-bit AVR so keep track of
the port and bitmask
// for the digital pin connected to the DHT. Other platforms
will use digitalRead.
uint8_t _bit, _port;
#endif
uint32_t _lastreadtime, _maxcycles;
bool _lastresult;

uint32_t expectPulse(bool level);
};

class InterruptLock {
public:
InterruptLock() {
noInterrupts();
}
~InterruptLock() {
interrupts();
}
};

#endif
```




ΠΑΡΑΡΤΗΜΑ 2

Βιβλιοθήκη Adafruit_Sensor

```
/*
 * Copyright (C) 2008 The Android Open Source Project
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the
 * License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing,
 * software distributed under the License is distributed on an "AS IS"
 * BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
 * implied.
 * See the License for the specific language governing permissions
 * and
 * limitations under the License.
 */

/* Update by K. Townsend (Adafruit Industries) for lighter
 * typedefs, and
 * extended sensor support to include color, voltage and current
 */

#ifndef _ADAFRUIT_SENSOR_H
#define _ADAFRUIT_SENSOR_H

#if ARDUINO >= 100
#include "Arduino.h"
#include "Print.h"
#else
#include "WProgram.h"
#endif

/* Intentionally modeled after sensors.h in the Android API:
 *
 * https://github.com/android/platform\_hardware\_libhardware/blob/master/include/hardware/sensors.h */

/* Constants */
#define SENSORS_GRAVITY_EARTH (9.80665F)
/**< Earth's gravity in m/s^2 */
#define SENSORS_GRAVITY_MOON (1.6F)
```



```
/**< The moon's gravity in m/s^2 */
#define SENSORS_GRAVITY_SUN (275.0F)
/**< The sun's gravity in m/s^2 */
#define SENSORS_GRAVITY_STANDARD (SENSORS_GRAVITY_EARTH)
#define SENSORS_MAGFIELD_EARTH_MAX (60.0F)
/**< Maximum magnetic field on Earth's surface */
#define SENSORS_MAGFIELD_EARTH_MIN (30.0F)
/**< Minimum magnetic field on Earth's surface */
#define SENSORS_PRESSURE_SEALEVELHPA (1013.25F)
/**< Average sea level pressure is 1013.25 hPa */
#define SENSORS_DPS_TO_RADS (0.017453293F)
/**< Degrees/s to rad/s multiplier */
#define SENSORS_GAUSS_TO_MICROTESLA (100)
/**< Gauss to micro-Tesla multiplier */

/** Sensor types */
typedef enum
{
    SENSOR_TYPE_ACCELEROMETER = (1), /**< Gravity +
linear acceleration */
    SENSOR_TYPE_MAGNETIC_FIELD = (2),
    SENSOR_TYPE_ORIENTATION = (3),
    SENSOR_TYPE_GYROSCOPE = (4),
    SENSOR_TYPE_LIGHT = (5),
    SENSOR_TYPE_PRESSURE = (6),
    SENSOR_TYPE_PROXIMITY = (8),
    SENSOR_TYPE_GRAVITY = (9),
    SENSOR_TYPE_LINEAR_ACCELERATION = (10), /**< Acceleration
not including gravity */
    SENSOR_TYPE_ROTATION_VECTOR = (11),
    SENSOR_TYPE_RELATIVE_HUMIDITY = (12),
    SENSOR_TYPE_AMBIENT_TEMPERATURE = (13),
    SENSOR_TYPE_VOLTAGE = (15),
    SENSOR_TYPE_CURRENT = (16),
    SENSOR_TYPE_COLOR = (17)
} sensors_type_t;

/** struct sensors_vec_s is used to return a vector in a common
format. */
typedef struct {
    union {
        float v[3];
        struct {
            float x;
            float y;
            float z;
        };
    };
    /** Orientation sensors */
    struct {
        float roll; /**< Rotation around the longitudinal
axis (the plane body, 'X axis'). Roll is positive and increasing
when moving downward. -90°≤roll≤90° */
        float pitch; /**< Rotation around the lateral axis
(the wing span, 'Y axis'). Pitch is positive and increasing when
moving upwards. -180°≤pitch≤180° */
    };
};
```



```
        float heading; /**< Angle between the longitudinal
axis (the plane body) and magnetic north, measured clockwise when
viewing from the top of the device. 0-359° */
    };
};
int8_t status;
uint8_t reserved[3];
} sensors_vec_t;

/** struct sensors_color_s is used to return color data in a
common format. */
typedef struct {
    union {
        float c[3];
        /* RGB color space */
        struct {
            float r;      /**< Red component */
            float g;      /**< Green component */
            float b;      /**< Blue component */
        };
    };
    uint32_t rgba;      /**< 24-bit RGBA value */
} sensors_color_t;

/** Sensor event (36 bytes) */
/** struct sensor_event_s is used to provide a single sensor
event in a common format. */
typedef struct
{
    int32_t version;      /**< must be
sizeof(struct sensors_event_t) */
    int32_t sensor_id;      /**< unique sensor
identifier */
    int32_t type;      /**< sensor type */
    int32_t reserved0;      /**< reserved */
    int32_t timestamp;      /**< time is in
milliseconds */
    union
    {
        float data[4];
        sensors_vec_t acceleration;      /**< acceleration
values are in meter per second per second (m/s^2) */
        sensors_vec_t magnetic;      /**< magnetic
vector values are in micro-Tesla (uT) */
        sensors_vec_t orientation;      /**< orientation
values are in degrees */
        sensors_vec_t gyro;      /**< gyroscope
values are in rad/s */
        float temperature;      /**< temperature is
in degrees centigrade (Celsius) */
        float distance;      /**< distance in
centimeters */
        float light;      /**< light in SI
lux units */
        float pressure;      /**< pressure in
```



```
hectopascal (hPa) */
    float          relative_humidity;    /**< relative
humidity in percent */
    float          current;              /**< current in
milliamps (mA) */
    float          voltage;              /**< voltage in
volts (V) */
    sensors_color_t color;               /**< color in RGB
component values */
};
} sensors_event_t;

/* Sensor details (40 bytes) */
/** struct sensor_s is used to describe basic information about a
specific sensor. */
typedef struct
{
    char          name[12];              /**< sensor name */
    int32_t       version;               /**< version of the
hardware + driver */
    int32_t       sensor_id;            /**< unique sensor
identifier */
    int32_t       type;                 /**< this sensor's
type (ex. SENSOR_TYPE_LIGHT) */
    float         max_value;            /**< maximum value
of this sensor's value in SI units */
    float         min_value;           /**< minimum value
of this sensor's value in SI units */
    float         resolution;          /**< smallest
difference between two values reported by this sensor */
    int32_t       min_delay;            /**< min delay in
microseconds between events. zero = not a constant rate */
} sensor_t;

class Adafruit_Sensor {
public:
    // Constructor(s)
    Adafruit_Sensor() {}
    virtual ~Adafruit_Sensor() {}

    // These must be defined by the subclass
    virtual void enableAutoRange(bool enabled) {};
    virtual bool getEvent(sensors_event_t*) = 0;
    virtual void getSensor(sensor_t*) = 0;

private:
    bool _autoRange;
};

#endif
```



ΠΑΡΑΡΤΗΜΑ 3

Βιβλιοθήκη Blynk

```
/**
 * @file      BlynkApiArduino.h
 * @author    Volodymyr Shymanskyy
 * @license   This project is released under the MIT License
 (MIT)
 * @copyright Copyright (c) 2015 Volodymyr Shymanskyy
 * @date      Mar 2015
 * @brief
 *
 */

#ifndef BlynkApiArduino_h
#define BlynkApiArduino_h

#include <Blynk/BlynkApi.h>
#include <Arduino.h>

template<class Proto>
void BlynkApi<Proto>::Init()
{
}

template<class Proto>
BLYNK_FORCE_INLINE
millis_time_t BlynkApi<Proto>::getMillis()
{
// TODO: Remove workaround for Intel Curie
// https://forum.arduino.cc/index.php?topic=391836.0
#ifdef ARDUINO_ARCH_ARC32
  noInterrupts();
  uint64_t t = millis();
  interrupts();
  return t;
#else
  return millis();
#endif
}

#ifdef BLYNK_NO_INFO

template<class Proto>
BLYNK_FORCE_INLINE
void BlynkApi<Proto>::sendInfo() {}

#else
```



```
template<class Proto>
BLYNK_FORCE_INLINE
void BlynkApi<Proto>::sendInfo()
{
    static const char profile[] BLYNK_PROGMEM =
        BLYNK_PARAM_KV("ver"      , BLYNK_VERSION)
        BLYNK_PARAM_KV("h-beat"  ,
BLYNK_TOSTRING(BLYNK_HEARTBEAT))
        BLYNK_PARAM_KV("buff-in",
BLYNK_TOSTRING(BLYNK_MAX_READBYTES))
#ifdef BLYNK_INFO_DEVICE
        BLYNK_PARAM_KV("dev"      , BLYNK_INFO_DEVICE)
#endif
#ifdef BLYNK_INFO_CPU
        BLYNK_PARAM_KV("cpu"      , BLYNK_INFO_CPU)
#endif
#ifdef BLYNK_INFO_CONNECTION
        BLYNK_PARAM_KV("con"      , BLYNK_INFO_CONNECTION)
#endif
        BLYNK_PARAM_KV("build"    , __DATE__ " " __TIME__)
    ;
    const size_t profile_len = sizeof(profile)-1;

#ifdef BLYNK_HAS_PROGMEM
    char mem[profile_len];
    memcpy_P(mem, profile, profile_len);
    static_cast<Proto*>(this)->sendCmd(BLYNK_CMD_INTERNAL, 0,
mem, profile_len);
#else
    static_cast<Proto*>(this)->sendCmd(BLYNK_CMD_INTERNAL, 0,
profile, profile_len);
#endif
    return;
}

#endif

template<class Proto>
BLYNK_FORCE_INLINE
void BlynkApi<Proto>::processCmd(const void* buff, size_t len)
{
    BlynkParam param((void*)buff, len);
    BlynkParam::iterator it = param.begin();
    if (it >= param.end())
        return;
    const char* cmd = it.asStr();
    uint16_t cmd16;
    memcpy(&cmd16, cmd, sizeof(cmd16));
    if (++it >= param.end())
        return;

#ifdef defined(analogInputToDigitalPin)
    // Good! Analog pins can be referenced on this device by
name.
    const uint8_t pin = (it.asStr()[0] == 'A') ?
```



```
analogInputToDigitalPin(atoi(it.asStr()+1)) :
    it.asInt();
#else
    #if defined(BLYNK_DEBUG_ALL)
        #pragma message "analogInputToDigitalPin not defined"
    #endif
    const uint8_t pin = it.asInt();
#endif

    switch(cmd16) {

#ifdef BLYNK_NO_BUILTIN

    case BLYNK_HW_PM: {
        while (it < param.end()) {
            ++it;
            if (!strcmp(it.asStr(), "in")) {
                pinMode(pin, INPUT);
            } else if (!strcmp(it.asStr(), "out") ||
!strcmp(it.asStr(), "pwm")) {
                pinMode(pin, OUTPUT);
#ifdef INPUT_PULLUP
            } else if (!strcmp(it.asStr(), "pu")) {
                pinMode(pin, INPUT_PULLUP);
#endif
#ifdef INPUT_PULLDOWN
            } else if (!strcmp(it.asStr(), "pd")) {
                pinMode(pin, INPUT_PULLDOWN);
#endif
            } else {
#ifdef BLYNK_DEBUG
                BLYNK_LOG4(BLYNK_F("Invalid pin "), pin,
BLYNK_F(" mode "), it.asStr());
#endif
            }
            ++it;
        }
    } break;
    case BLYNK_HW_DR: {
        char mem[16];
        BlynkParam rsp(mem, 0, sizeof(mem));
        rsp.add("dw");
        rsp.add(pin);
        rsp.add(digitalRead(pin));
        static_cast<Proto*>(this)->sendCmd(BLYNK_CMD_HARDWARE, 0,
rsp.getBuffer(), rsp.getLength()-1);
    } break;
    case BLYNK_HW_DW: {
        // Should be 1 parameter (value)
        if (++it >= param.end())
            return;
    }

#endif

#ifdef ESP8266
    // Disable PWM...
```



```
        analogWrite(pin, 0);
#endif
#ifndef BLYNK_MINIMIZE_PINMODE_USAGE
        pinMode(pin, OUTPUT);
#endif
        digitalWrite(pin, it.asInt() ? HIGH : LOW);
    } break;
    case BLYNK_HW_AR: {
        char mem[16];
        BlynkParam rsp(mem, 0, sizeof(mem));
        rsp.add("aw");
        rsp.add(pin);
        rsp.add(analogRead(pin));
        static_cast<Proto*>(this)->sendCmd(BLYNK_CMD_HARDWARE, 0,
rsp.getBuffer(), rsp.getLength()-1);
    } break;

// TODO: Remove workaround for ESP32
#if !defined(ESP32)

    case BLYNK_HW_AW: {
        // Should be 1 parameter (value)
        if (++it >= param.end())
            return;

#ifndef BLYNK_MINIMIZE_PINMODE_USAGE
        pinMode(pin, OUTPUT);
#endif
        analogWrite(pin, it.asInt());
    } break;

#endif // TODO: Remove workaround for ESP32

#endif

    case BLYNK_HW_VR: {
        BlynkReq req = { pin };
        WidgetReadHandler handler = GetReadHandler(pin);
        if (handler && (handler != BlynkWidgetRead)) {
            handler(req);
        } else {
            BlynkWidgetReadDefault(req);
        }
    } break;
    case BLYNK_HW_VW: {
        ++it;
        char* start = (char*)it.asStr();
        BlynkParam param2(start, len - (start - (char*)buff));
        BlynkReq req = { pin };
        WidgetWriteHandler handler = GetWriteHandler(pin);
        if (handler && (handler != BlynkWidgetWrite)) {
            handler(req, param2);
        } else {
            BlynkWidgetWriteDefault(req, param2);
        }
    }
}
```




```
    } break;
    default:
        BLYNK_LOG2(BLYNK_F("Invalid HW cmd: "), cmd);
        static_cast<Proto*>(this)->sendCmd(BLYNK_CMD_RESPONSE,
static_cast<Proto*>(this)->msgIdOutOverride, NULL,
BLYNK_ILLEGAL_COMMAND);
    }
}

#endif
```



DATASHEETS

- Atmel ATmega 2560 datasheet is available in the below link.
http://www.atmel.com/images/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf
- Bluetooth HC-05 datasheet is available in the link below.
- <http://www.electronicaestudio.com/docs/istd016A.pdf>
- PIR (motion detection) sensor datasheet is available in the below link.
<https://learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf>
- Temperature and Humidity sensor (DHT11) datasheet is available in the below link.
<http://files.amperka.ru/datasheets/dht11.pdf>
- Photocell (light sensor) datasheet is available in the below link.
<https://learn.adafruit.com/downloads/pdf/photocells.pdf>
- Multi-channel relay board datasheet is available in the link below.
<http://www.ti.com/lit/ds/symlink/tp19201.pdf>



Βιβλιογραφία:

[1] K. Wacks, Home systems standards: achievements and challenges, Communications Magazine, IEEE 40 (4) (2002) 152 – 159. doi:10.1109/35.995865. 10, 15

[2] Arduino Mega 2560. (2017) Retrieved from <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>
Copyright by Creative Commons Attribution ShareAlike 3.0.

[3] SunfounderArduinoLearning(2017)Retrieved from http://www.sunfounder.com/index.php?c=case_incs&a=typelist#Arduino

[4]Arduino IDE,(2017) Retrieved from <http://www.arduino.cc/en/Main/software>

[5] Atmel ATmega 2560. (2017) Retrieved from <http://www.atmel.com/devices/atmega2560.aspx>

[6]Photocell,(2017) Retrieved from <http://download.instructables.com/EIU/5ET1/FVS81JJ6/Photocell-tutorial.pdf>

[7] Xbee,(2017) Retrieved from <http://en.wikipedia.org/wiki/XBee>

[8] Home automation,(2017) Retrieved from https://en.wikipedia.org/wiki/Home_automation

[9]Smart Home,(2017) Retrieved from <http://www.cedia.org/blog/what-is-a-smart-home-the-basics-of->



[home-automation](#)

[10]Bluetooth HC-05,(2017) Retrieved from

http://wiki.pinguino.cc/index.php/SPP_Bluetooth_Modules

[11] Motion detector,(2017) Retrieved from

https://en.wikipedia.org/wiki/Motion_detector

[12] Arduino History,(2017)Retrieved from

<https://en.wikipedia.org/wiki/Arduino>

[13] C. Douligeris, J. Khawand, C. Khawand, Communications and control for a home automation system, in: Southeastcon '91., IEEE Proceedings of, 1991, pp. 171 –175 vol.1. doi:10.1109/SECON.1991.147729. 10, 11, 12

[14] J.-S. Lee, Y.-W. Su, C.-C. Shen, A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi, in: Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE, 2007, pp. 46 –51. doi:10.1109/IECON.2007.4460126. 8

[15] E. Ferro, F. Potorti, Bluetooth and wi-fi wireless protocols: a survey and a comparison, Wireless Communications, IEEE 12 (1) (2005) 12 – 26. doi:10.1109/MWC.2005. 1404569. 8