

ΑΥΤ.
636

**ΟΙΚΙΑΚΟΣ ΑΥΤΟΜΑΤΙΣΜΟΣ ΜΕ ΧΡΗΣΗ ΥΠΟΛΟΓΙΣΤΗ
RASPBERRY PI**



ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΙΔΡΥΜΑ ΠΕΙΡΑΙΑ
Τμήμα Αυτοματισμού

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
Σπύρος Φιτιλής



Επιβλέπων: Γρηγόριος Νικολάου

ΕΙΣΑΓΩΓΗ

Η ραγδαία τεχνολογική εξέλιξη που σημειώνεται τις τελευταίες δεκαετίες στο χώρο του ηλεκτρικού οικιακού εξοπλισμού αλλά και των κεντρικών εγκαταστάσεων κλιματισμού, ασφάλειας και κατ'οίκον διασκέδασης έχει οδηγήσει σε αύξηση των ηλεκτρικών συσκευών και συστημάτων που εγκαθίστανται στα σύγχρονα κτίρια. Κατά συνέπεια, γίνεται όλο και πιο έντονη η ανάγκη για σωστή διαχείριση των συσκευών αυτών καθώς και των πηγών ενέργειας από τις οποίες οι συσκευές αυτές τροφοδοτούνται. Οι προσπάθειες για την ικανοποίηση αυτής της ανάγκης οδήγησαν στο σχεδιασμό διαφόρων συστημάτων οικιακού αυτοματισμού, τεχνολογίες οι οποίες στη συνέχεια ενοποιήθηκαν στην τεχνολογία του «έξυπνου σπιτιού».

Παρά το κόστος το οποίο συμπορεύεται με την εγκατάσταση των συστημάτων οικιακού αυτοματισμού –το οποίο, όμως, με την πάροδο του χρόνου παρουσιάζει πτωτικές τάσεις, κοινό χαρακτηριστικό των τεχνολογικών εξελίξεων- υπάρχουν σημαντικοί παράγοντες οι οποίοι κρίνουν την εγκατάσταση τέτοιων συστημάτων διαχείρισης ηλεκτρικών συσκευών και συστημάτων ως ιδιαίτερα ελκυστικοί. Κυριότερος είναι ο οικονομικός, αφού κύριος στόχος ενός αυτόματου συστήματος διαχείρισης ηλεκτρικών συσκευών είναι ο περιορισμός της κατανάλωσης ηλεκτρικής ενέργειας στα επίπεδα της ελάχιστης απαραίτητης. Από τον οικονομικό απορρέει ένας εξίσου σημαντικός παράγοντας ο οποίος είναι ο περιβαλλοντικός αφού η ορθολογική χρήση ηλεκτρικής ενέργειας έχει θετικό αντίκτυπο στο περιβάλλον. Επίσης, σε μια κοινωνία ευημερίας και ανοδικού βιωτικού επιπέδου, είναι σημαντικός παράγοντας άνεσης και ποιοτικών συνθηκών διαβίωσης και εργασίας καθώς και ένας καλαίσθητος οικιακός περιβάλλοντος. Τέλος, είναι πολύ σημαντικός παράγοντας τόνωσης του αισθήματος ασφάλειας του κατοίκου ο οποίος πλέον μπορεί να χρησιμοποιεί πλατφόρμες συνολικής διαχείρισης των συστημάτων ασφαλείας της περιουσίας του τόσο όταν βρίσκεται εντός της οικίας του όσο και όταν απουσιάζει από αυτή.

Η έννοια του οικιακού αυτοματισμού εμφανίστηκε πρώτη φορά περί τη δεκαετία του 1970 σε κτίρια υπηρεσιακού χαρακτήρα. Οι εφαρμογές της εποχής αποσκοπούσαν σε κεντρικό έλεγχο του φωτισμού, των κλιματικών συνθηκών και της ασφάλειας των κτιρίων. Μέχρι σήμερα, η κεντρική ιδέα πίσω από τον οικιακό αυτοματισμό παραμένει η ίδια. Ωστόσο, έχουν εξελιχθεί οι τομείς στους οποίους αυτός εφαρμόζεται. Έτσι, υπάρχουν πλέον ειδικά συστήματα αυτοματισμού που προορίζονται για οικίες ατόμων με ειδικές ανάγκες ή για ηλικιωμένους. Επίσης, νέες εφαρμογές επιτρέπουν τη ρύθμιση των κλιματικών συνθηκών κλειστών χώρων τόσο μέσω συστημάτων κλιματισμού, όσο και μέσω συστημάτων σκίασης και αποθήκευσης συγκεκριμένων κλιματικών συνθηκών αντιστοιχίζοντάς τα σε συγκεκριμένους χρήστες προσωποποιώντας εξολοκλήρου τις ανάγκες των κατοίκων. Σημαντικές είναι και οι νέες εφαρμογές στο χώρο της ασφάλειας. Έτσι, γίνεται επιλεκτική ενεργοποίηση συσκευών και φωτισμού κατά την απουσία των ατόμων από το σπίτι και ενεργοποίηση διαφόρων επιπέδων συστημάτων ελέγχου και συναγερμών. Είναι δυνατή, για παράδειγμα, η ενεργοποίηση συστήματος το οποίο να μπορεί να ελέγχει ανά

συγκεκριμένα χρονικά διαστήματα την ύπαρξη ή μη ποσοτήτων φυσικού αερίου στην οικιακή ατμόσφαιρα ή τα επίπεδα της σχετικής υγρασίας εντοπίζοντας κατ'αυτόν τον τρόπο διαρροές φυσικού αερίου ή πλημμύρες με δυνατότητα διακοπής της παροχής φυσικού αερίου ή νερού. Στη συγκεκριμένη περίπτωση, η ανάπτυξη στις εφαρμογές ακολουθεί αυτή της τεχνολογίας των αισθητήρων που αποτελούν κύριο συστατικό των συστημάτων ελέγχου. Πλήθος εφαρμογών θα αναφερθούν και θα σχολιαστούν στο αντίστοιχο κεφάλαιο της παρούσας εργασίας.

Σκοπός της εργασίας είναι η δημιουργία ενός αυτοσχέδιου συστήματος αυτόματου ελέγχου ηλεκτρικών συσκευών με τη χρήση του υπολογιστή Raspberry Pi. Θεωρούμε ιδιαίτερα ενδιαφέρουσα την προσπάθεια επίτευξης ενός τέτοιου συστήματος του οποίου η κατασκευή είναι ιδιαίτερα οικονομική από πλευράς υλικού αλλά και ιδιαίτερα εύκολη από πλευράς τεχνικών γνώσεων. Έτσι, κατά τη δόμηση της εργασίας, η εκκίνηση δίνεται με το Κεφάλαιο 1 όπου θα γίνει αναφορά στις επικοινωνίες M2M (Machine-To-Machine Communications). Οι επικοινωνίες M2M είναι ο θεμέλιος λίθος της λειτουργίας οποιουδήποτε συστήματος αυτόματου ελέγχου και διαχείρισης συσκευών για ευνόητους λόγους οπότε σκόπιμα αποτελεί και το θεμέλιο λίθο της εργασίας μας.

Στο Κεφάλαιο 2, ο αναγνώστης έχει την ευκαιρία να ενημερωθεί για τον υπολογιστή Raspberry Pi. Ο Raspberry Pi είναι ένας mini προσωπικός υπολογιστής ο οποίος έκανε το εμπορικό του ντεμπούτο το Φεβρουάριο του 2012 από το βρετανικό κοινωφελές ίδρυμα "Raspberry Pi Foundation" αποσκοπώντας στην παροχή σύγχρονης τεχνολογίας σε όσο το δυνατόν περισσότερους ανθρώπους αφού το ελάχιστο κόστος του σε σύγκριση με τις παρεχόμενες δυνατότητες τον καθιστούν ελκυστικό για μεγάλο εύρος εφαρμογών.

Στο Κεφάλαιο 3, καταπιανόμαστε με την πλατφόρμα Arduino. Το Αρδθινο είναι ένας microcontroller, ο οποίος ουσιαστικά θα αναλάβει τη διαχείριση του υλισμικού του συστήματός μας. Έτσι, κρίνεται σκόπιμη μια μικρή εισαγωγή στις βασικές απαιτήσεις και προδιαγραφές του.

Στο Κεφάλαιο 4 αναφερόμαστε ενδελεχώς στην έννοια του οικιακού αυτοματισμού καθώς και στο σύνολο των υπαρχουσών εφαρμογών που εντάσσονται στα πλαίσιά του. Επίσης, αναφερόμαστε στη σημαντικότητα ύπαρξης συστημάτων οικιακού αυτοματισμού καθώς και σε προσωπικές προβλέψεις αναφορικά με τη μελλοντική εξέλιξή τους.

Στο Κεφάλαιο 5 καταπιανόμαστε με το πρακτικό μέρος της εργασίας το οποίο δεν είναι άλλο από το στήσιμο του Raspberry Pi mini-PC. Διαδοχικά, θα γίνει περιγραφή του υλικού όπως το παραλαμβάνει ο αγοραστής από την παραγωγό εταιρεία και των προδιαγραφών αυτού, τη μορφοποίηση του σκληρού του δίσκου ο οποίος δεν είναι άλλος από μια Secure Digital κάρτα (SD card) και την εγκατάσταση του λειτουργικού συστήματος σε αυτή.

Στο τελευταίο κεφάλαιο, το Κεφάλαιο 6, γίνεται μια προσπάθεια να χρησιμοποιηθεί το μηχάνημα που κατασκευάσαμε σε στοιχειώδεις εφαρμογές οικιακού αυτοματισμού. Αυτές θα περιλαμβάνουν την κατασκευή ενός θερμομέτρου, ενός θερμοστάτη και ενός συστήματος αυτόματου χειρισμού των οικιακών κουρτίνων.

Κλείνοντας, καταθέτουμε τον επίλογό μας όπου ανακεφαλαιώνουμε όσα συζητήθηκαν στα πέντε αυτά κεφάλαια παραθέτοντας προσωπικές απόψεις αναφορικά με την εξέλιξη των συστημάτων αυτοματισμού για κατ'οίκον χρήση.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Σπύρου Φιτιλή, ο οποίος και διατηρεί κάθε δικαίωμα χρήσης ή αναπαραγωγής.

Αθήνα, Γενάρης 2013

Σπύρος Φιτιλής

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1: ΕΠΙΚΟΙΝΩΝΙΕΣ MACHINE-TO-MACHINE (M2M)

- 1.1 Εισαγωγή
- 1.2 Τα στάδια της M2M επικοινωνίας
- 1.3 Αρχιτεκτονική της M2M επικοινωνίας

ΚΕΦΑΛΑΙΟ 2: Ο ΥΠΟΛΟΓΙΣΤΗΣ RASPBERRY PI

- 2.1 Εισαγωγή
- 2.2 Το Hardware του Raspberry Pi
- 2.3 Το Software του Raspberry Pi
- 2.4 Εξαρτήματα για τον υπολογιστή Raspberry Pi

ΚΕΦΑΛΑΙΟ 3: Η ΠΛΑΤΦΟΡΜΑ ARDUINO

- 3.1 Εισαγωγή
- 3.2 Το Hardware του Arduino
- 3.3 Το Software του Arduino
- 3.4 Σύνδεση του Arduino με υπολογιστή
 - 3.4.1 Σύνδεση Arduino με Raspberry Pi
 - 3.4.2 Δημιουργία λογισμικού για το Arduino

ΚΕΦΑΛΑΙΟ 4: ΟΙΚΙΑΚΟΣ ΑΥΤΟΜΑΤΙΣΜΟΣ

- 4.1 Εισαγωγή
- 4.2 Ωφέλη οικιακού αυτοματισμού
- 4.3 Στοιχεία Συστήματος Οικιακού Αυτοματισμού
 - 4.3.1 Το Πρωτόκολλο X10
- 4.4 Κόστος Οικιακού Αυτοματισμού

ΚΕΦΑΛΑΙΟ 5: ΚΑΤΑΣΚΕΥΗ ΕΝΟΣ ΣΥΣΤΗΜΑΤΟΣ RASPBERRY PI - ARDUINO

- 5.1 Εισαγωγή
- 5.2 Δημιουργία του υπολογιστή Raspberry Pi

- 5.2.1 Προετοιμασία SD κάρτας
- 5.2.2 Εγκατάσταση Λειτουργικού Συστήματος
- 5.3 Εγκατάσταση της ασπίδας Arduino
- 5.3.1 Ασπίδα γέφυρας Raspberry Pi σε Arduino

ΚΕΦΑΛΑΙΟ 6: ΕΦΑΡΜΟΓΕΣ ΟΙΚΙΑΚΟΥ ΑΥΤΟΜΑΤΙΣΜΟΥ

- 6.1 Εισαγωγή
- 6.2 Κατασκευή ενός βασικού θερμόμετρου
 - 6.2.1 Ορισμός Θερμομέτρου
 - 6.2.2 Υλισμικό του θερμομέτρου
 - 6.2.3 Λογισμικό του θερμομέτρου
 - 6.2.4 Χρήση του θερμομέτρου σε M2M επικοινωνίες
- 6.3 Κατασκευή ενός θερμοστάτη
 - 6.3.1 Ορισμός του θερμοστάτη
 - 6.3.1 Υλισμικό του θερμοστάτη
 - 6.3.2 Λογισμικό του θερμοστάτη
- 6.4 Κατασκευή μιας βάσης δεδομένων για αποθήκευση των αποτελεσμάτων μετρησης της θερμοκρασίας
 - 6.4.1 Δημιουργία Της Βάσης Δεδομένων
 - 6.4.2 Δημιουργία Του Web Server
 - 6.4.3 Συγγραφή Του Κώδικα Της Βάσης Δεδομένων

ΕΠΙΛΟΓΟΣ

ΚΕΦΑΛΑΙΟ 1

ΕΠΙΚΟΙΝΩΝΙΕΣ MACHINE-TO-MACHINE (M2M)

1.1 Εισαγωγή

Με τον όρο «M2M Επικοινωνίες» ή «Machine-To-Machine Communications» αναφερόμαστε στην χαμηλής ισχύος και χαμηλού κόστους αυτόματη ανταλλαγή πληροφοριών μεταξύ ασύρματων ή ενσύρματων μηχανών και συσκευών του ίδιου τύπου με ελάχιστη ανθρώπινη παρεμβάση ή ακόμη και με πλήρη απουσία αυτής. Κατά τη διαδικασία αυτή, εξασφαλίζονται οι κατάλληλες συνθήκες ώστε μηχανήματα και συσκευές να επικοινωνούν μεταξύ τους ανταλλάσσοντας πληροφορίες οι οποίες επεξεργαζόμενες και αξιολογούμενες προκαλούν αυτόματες λειτουργίες ωφέλιμες για την ανθρώπινη δραστηριότητα. Η επικοινωνία αυτή γίνεται μέσω της χρήσης ενός δικτύου επικοινωνίας όπως το Bluetooth, το Wi-Fi και το Zigbee καθώς και ενός συνόλου από υλισμικής σκοπιάς όπως ένας υπολογιστής με ενσωματωμένο μικροεπεξεργαστή, αισθητήρες, διακόπτες, οθόνη χειρισμού κ.α..

Η κύρια ιδέα γύρω από την οποία έχουν δομηθεί οι M2M επικοινωνίες είναι αυτή του δικτύου και πιο συγκεκριμένα η αντίληψη πως οι συσκευές λειτουργούν βέλτιστα όταν είναι συνδεδεμένες σε δίκτυο ενώ ταυτόχρονα το ίδιο το δίκτυο επωφελείται από την προσθήκη όλο και περισσότερων συσκευών σε αυτό. Έτσι, υπάρχει η τάση για όλο και περισσότερες συσκευές να συνδέονται σε δίκτυα επικοινωνίας με την M2M επικοινωνία να τείνει να τυποποιηθεί στα πλαίσια των ασύρματων επικοινωνιών 5^{ης} γενιάς. Έτσι, μέσα στην επόμενη χρονιά (2014) προβλέπεται ότι θα υπάρχουν 1.5 δισεκατομμύρια συνδεδεμένες ασύρματες συσκευές (εξαιρουμένων των κινητών τηλεφώνων), σπάζοντας κάθε προηγούμενο ρεκόρ αναφορικά με τον όγκο των M2M δεδομένων. Προς το παρόν, οι διάφορες εφαρμογές των M2M επικοινωνιών έχουν κάνει την εμφάνισή τους σε τομείς όπως η υγειονομική περίθαλψη, τα κυβερνοσυστήματα μεταφοράς (CTS) και οι αυτόματες οικιακές τεχνολογίες, που αποτελούν και το αντικείμενο που θα μελετήσουμε.

Αρχικά, η M2M επικοινωνία εφαρμόστηκε στη βιομηχανική οργανολογία όπου ένας αισθητήρας ή μετρητής καλούταν να ανιχνεύσει ένα συγκεκριμένο γεγονός (το επίπεδο πληρότητας μιας αποθήκης τροφίμων είναι μια τέτοια ανιχνεύσιμη ποσότητα), σήμα το οποίο μεταβιβαζόταν μέσω ενός δικτύου (ενσύρματου, ασύρματου ή υβριδικού) σε εφαρμογή (επίπεδο λογισμικού) η οποία θα το μετέφραζε σε χρήσιμη πληροφορία αποσκοπώντας σε κάποια λήψη αποφάσεων (η αποθήκη, για παράδειγμα, θα έπρεπε ίσως να ενισχυθεί από άποψη προμηθειών). Σχηματικά, λοιπόν, η επικοινωνία αυτή επιτυγχανόταν μέσω ενός δικτύου συσκευών οι οποίες μεταβίβαζαν δεδομένα προς ανάλυση σε έναν κεντρικό κόμβο και στη συνέχεια η παραγόμενη από τον κόμβο πληροφορία θα επανακατευθυνόταν σε ένα σύστημα τύπου προσωπικού υπολογιστή.

Ωστόσο, οι σύγχρονες M2M επικοινωνίες έχουν ξεφύγει από τα στενά πλαίσια της σύνδεσης συσκευών μια-προς-μια και έχουν αποκτήσει τη μορφή του συστήματος δικτύων που εκπέμπουν δεδομένα σε προσωπικές ηλεκτρικές συσκευές. Αυτή η εξέλιξη ωφείλεται σε πολύ μεγάλο βαθμό στην ανάπτυξη των IP δικτύων παγκοσμίως, και ανάπτυξη η οποία καθιστά την M2M επικοινωνία ευκολότερη, γρηγορότερη και με μικρότερες ενεργειακές ανάγκες. Έτσι, ενώ αρχικά οι M2M επικοινωνίες ξεκίνησαν ως βιομηχανικές εφαρμογές, στις μέρες μας έχουν πλέον υπεισέρθει στο τεχνολογικό κλάδο της Τηλεματικής.

Θεωρώντας την εξέλιξη τους με το πέρασμα του χρόνου, οι M2M επικοινωνίες έχουν υπάρξει σε διάφορες μορφές από την έλευση του δικτυακού αυτοματισμού μέχρι σήμερα και έχουν χρησιμοποιηθεί σε εφαρμογές της Τηλεμετρίας, της Βιομηχανίας, του Αυτοματισμού καθώς και σε εφαρμογές SCADA (Supervisory Control & Data Acquisition).

Παρά το γεγονός ότι τα κυψελωτά δίκτυα γίνονται συνεχώς όλο και πιο συνηθισμένα, ένας σημαντικός όγκος συσκευών ακολουθεί να χρησιμοποιεί επίγειες γραμμές (POTS, DSL, καλώδια) για τη σύνδεση σε δίκτυα IP. Η M2M επικοινωνία αναδύθηκε το 1995 όταν η Siemens κατασκεύασε ειδικό τμήμα στην μονάδα της κινητής της τηλεφωνίας ώστε να κατασκευαστεί και να δοθεί στην αγορά ένα δίκτυο τύπου GSM με την ονομασία «M1» που θα επιτρέπει την επικοινωνία συσκευών μέσω ασύρματων δικτύων. Τον Οκτώβριο του 2000 δημιουργήθηκε ειδικό τμήμα εντός της εταιρείας με την ονομασία «Μονάδα Ασύρματων» το οποίο τον Ιούνιο του 2008 μετατράπηκε σε αυτόνομη εταιρεία με την ονομασία «Cinterion». Οι πρώτες μονάδες M1 χρησιμοποιήθηκαν σε κινούμενα οχήματα, την παρακολούθηση εξ'αποστάσεως και σε εφαρμογές παρακολούθησης και εντοπισμού. Οι πρώτες εμπορικές βιομηχανίες οι οποίες αναγνώρισαν τα ωφέλη της νεοαφιχθείσας M2M επικοινωνίας ήταν η GM και η εταιρεία ηλεκτρονικών Hughes. Σήμερα, η M2M επικοινωνία έχει πλέον ραφιναρισμένα χαρακτηριστικά και δυνατότητες όπως ο παγκόσμιος εντοπισμός στίγματος και ενσωματωμένη Java, σημαντική δυνατότητα για την επιτάχυνση της τεχνολογίας Ίντερνετ των Πραγμάτων (Internet of Things - IoT). Από άποψης υλικού, τα δίκτυα επικοινωνίας M2M κατασκευάζονται από πολύ συγκεκριμένες εταιρείες. Το 1998, η Quake Global κατασκεύασε τα πρώτα M2M δορυφορικά και επίγεια μόντεμ και ενώ μέχρι εκείνη την περίοδο εξαρτώταν πλήρως από το δίκτυο της ORBCOMM για τις δορυφορικές του υπηρεσίες, επεκτίνοντας τις δυνατότητές της στις δορυφορικές επικοινωνίες απέκτησε σαφές προβάδισμα στην αγορά δικτυακών προϊόντων. Το 2004, η Digi Int. ξεκίνησε την παραγωγή ασύρματων δρομολογητών ενώ το 2006 εξαγόρασε την Max Stream, κατασκευάστρια εταιρεία των ασύρματων XBee. Έτσι, είχε πλέον τη δυνατότητα να συνδεει συσκευές, όσο απομακρυσμένες κι αν ήταν αυτές. Μέχρι σήμερα, η Digi έχει συνεργαστεί με αρκετές εταιρείες με αποτέλεσμα να έχει επιτύχει τη σύνδεση εκατοντάδων χιλιάδων συσκευών παγκοσμίως. Το 2009, οι AT&T και Jasper ήρθαν σε συμφωνία ώστε να κατασκευάσουν από κοινού συσκευές για M2M επικοινωνία. Κατά την υπογραφή της συμφωνίας αυτής, οι εταιρείες δήλωσαν πως θα προσπαθήσουν να προωθήσουν τη συνδεσιμότητα μεταξύ ηλεκτρονικών συσκευών που αφορούν εξατομικευμένους καταναλωτές (και όχι πλέον σε βιομηχανικό ή εταιρικό επίπεδο), αυξάνοντας την ταχύτητα λειτουργίας αλλά και την αποδοτικότητα των συσκευών αυτών. Το 2009, επίσης, έλαβε χώρα το λανσάρισμα της

πλατφόρμας PRiSMPro™ από την KORE Telematics με σκοπό τη διαχείριση πολλαπλών δικτύων αλλά και της Telenor Connexion της νορβηγικής εταιρείας Telenor, η οποία είναι ο ηγέτης στην ευρωπαϊκή αγορά υπηρεσιών εφοδιασμού, διαχείρισης στόλου, ασφάλειας οχημάτων, υγειονομικής περίθαλψης και έξυπνων μετρητικών συστημάτων ηλεκτρικής κατανάλωσης. Το 2009 είναι η χρονιά που και η ισπανική Telefonica ανακοινώνει τη δημιουργία μονάδας M2M επικοινωνιών στη Μαδρίτη. Στις αρχές του 2010, οι AT&T, KPN, Rogers, Telcel και Jasper ξεκίνησαν μια συνεργασία για την κατασκευή ενός κεντρικού κόμβου εξυπηρέτησης προγραμματιστών που εργάζονται στις M2M επικοινωνίες ενώ οι Vodafone, Verizon και nPhase δημιούργησαν μια στρατηγική συμμαχία που αποσκοπούσε στη δημιουργία προϊόντων M2M με προορισμό την ευρωπαϊκή και αμερικανική αγορά. Το 2011, η Aeris Communications ανακοίνωσε τη συνεργασία της με την αυτοκινητοβιομηχανία Hyundai και η KORE συνεργάστηκε με τη Vodafone και την Iridium Communications για την προώθηση των προϊόντων και των υπηρεσιών της σε πάνω από 180 χώρες παγκοσμίως. Ωστόσο, ήταν η συνεργασία της με την Mach Communications η οποία εισήγαγε την KORE στις αγορές Ασίας-Ειρηνικού. Μελλοντικά, θεωρείται βέβαιο πως η συνδεσιμότητα σε νέφος θα ωφελήσει σημαντικά τις M2M επικοινωνίες αναφορικά με τις ταχύτητες σύνδεσης. Επίσης, μέσα στα επόμενα 5 χρόνια αναμένεται σημαντική ανάπτυξη στο χώρο των εφαρμογών ασφάλειας. Συγκεκριμένα, η ABI Research προβλέπει μέχρι το 2018 έσοδα ύψους \$198 εκατομμυρίων από εφαρμογές ασφάλειας.

Οι εφαρμογές των δικτύων επικοινωνίας M2M είναι πολυάριθμες. Στη βιομηχανία τα δίκτυα επικοινωνίας M2M προσφέρουν τη δυνατότητα ύπαρξης ταχέων γραμμών παραγωγής υψηλής παραγωγικότητας προϊόντων που θα καταλήξουν στο εμπόριο. Σπουδαίο παράδειγμα αποτελούν οι γραμμές παραγωγής των αυτοκινητοβιομηχανιών όπου τα δίκτυα M2M προσφέρουν πολύτιμη ανάδραση αναφορικά με την ανάκλιση συγκεκριμένων μερών της γραμμής και την αντικατάστασή τους. Μια, επίσης, σπουδαία εμπορική εφαρμογή είναι η τοποθέτηση μετρητών κατανάλωσης ως σύστημα ελέγχου μέσω ασύρματου δικτύου για την παρακολούθηση της πιθανής παραποίησης προϊόντων, εφαρμογή που αποτελεί σπουδαίο ποιοτικό έλεγχο κατά της οποιασδήποτε προσπάθειας εξαπάτησης από τη πλευρά κάποιου υπαλλήλου προς τη διοίκηση. Στον τομέα της διαφήμισης, τα M2M δίκτυα προσφέρουν τη δυνατότητα αυτόματης ενημέρωσης των ηλεκτρονικών πινακίδων με διαφορετικά μηνύματα ανάλογα με την ημέρα ή την ώρα της ημέρας και επίσης αυτόματη ενημέρωση αναφορικά με σημαντικές αλλαγές όπως για παράδειγμα στην παγκόσμια τιμή του πετρελαίου. Στον τομέα της αυτοκίνησης, μια ενδιαφέρουσα εφαρμογή είναι αυτή του εκ των προτέρων προγραμματισμού ενός ταξιδιού όπου ο οδηγός μπορεί να εποπτεύει και να ελέγχει τη διαθεσιμότητα των καυσίμων για τα συμβατικά αυτοκίνητα ή και της φόρτισης για τα υβριδικά, να προγραμματίζει τις στάσεις που θα πραγματοποιηθούν τόσο αριθμητικά όσο και τη θέση τους γεωγραφικά, να προθερμάνει ή να ψύξει το όχημα κτλ. Τέλος, ιδιαίτερης σημασίας είναι οι εφαρμογές της M2M επικοινωνίας στην Ιατρική όπου είναι δυνατή η παροχή φαρμακευτικής αγωγής σε συγκεκριμένο χρόνο και δόση σε ασθενείς που βρίσκονται σε καταστολή ή κώμα αλλά και η λειτουργία του εμφυτεύσιμου καρδιακού απινιδωτή-ανατακτη όπου λαμβάνει σήμα από την καρδιά αναφορικά με τον καρδιακό ρυθμό και

αντιδρά είτε λειτουργώντας ως βηματοδότης είτε δίνοντας ηλεκτροσοκ στον ασθενή για να αποφευχθεί η μαρμαρυγή και το ενδεχόμενο θανάτου ενώ ταυτόχρονα μπορεί να αποθηκεύει τα δεδομένα αυτά και να τα επικοινωνήσει στον ηλεκτρονικό υπολογιστή του προσωπικού γιατρού του ασθενούς.



Εικόνα 1: Τομείς όπου υφίστανται εφαρμογές της M2M επικοινωνίας.

1.2 Τα στάδια της M2M επικοινωνίας

Σκοπός της M2M επικοινωνίας είναι η δημιουργία μιας ροής δεδομένων μεταξύ μηχανών και συσκευών του ίδιου τύπου είτε με ενσύρματα ή με ασύρματα δίκτυα αποσκοπώντας τελικά αυτή η ροή να καταλήξει σε σύστημα διαχειριζόμενο και εποπτευόμενο από τον άνθρωπο. Ανεξάρτητα από το είδος, τον κατασκευαστή ή τη λειτουργία των μηχανημάτων και των συσκευών, αυτή είναι και η κεντρική ιδέα πίσω από την M2M επικοινωνία. Ωστόσο, το είδος της συνδεσμολογίας των συσκευών, το πρωτόκολλο επικοινωνίας καθώς και ο τρόπος επεξεργασίας των πρωτογενών δεδομένων μπορεί κάθε φορά να διαφέρει. Παράμετρος ρύθμισης των παραπάνω μεταβλητών που δημιουργούν την τελική μορφή της M2M επικοινωνίας είναι το είδος της εφαρμογής που αναπτύσσεται κάθε φορά, ο τρόπος και ο τόπος που αυτή θα εγκατασταθεί και οι γνώσεις που θα πρέπει να διαθέτει ο χρήστης της.

Υπάρχουν 4 πολύ βασικά στάδια επικοινωνίας που είναι κοινά στην πλειοψηφία των M2M εφαρμογών. Αυτά είναι τα εξής:

1. **Συλλογή δεδομένων:** Είναι μια πολύ απλή διαδικασία η οποία επιτελείται, για παράδειγμα, από έναν απλό αισθητήρα (π.χ. θερμοκρασίας), το

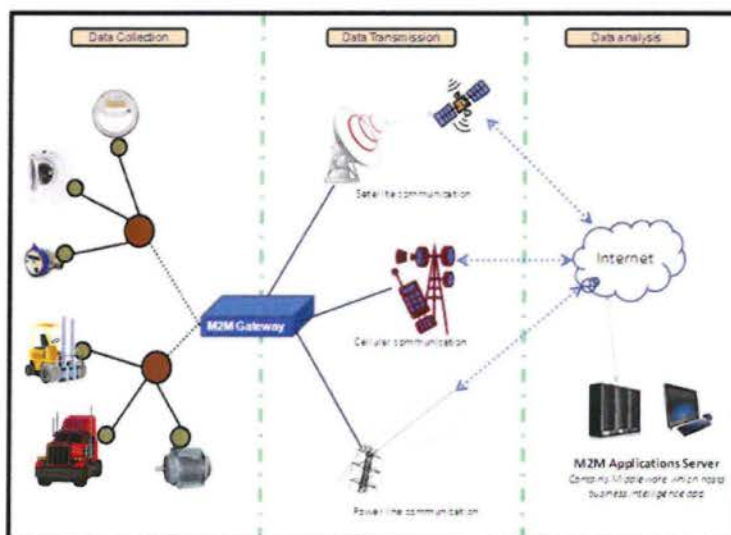
κλείσιμο μιας επαφής ή ενός διακόπτη ή ένα σύστημα με θύρες επικοινωνίας. Η μηχανή, λοιπόν, λαμβάνει τα πρωτογενή δεδομένα τα οποία θα αποσταλούν σε ένα δίκτυο όπου και θα αναλυθούν. Το στοιχείο αυτό που λαμβάνει τα δεδομένα μπορεί είτε να είναι συνδεδεμένο σε δίκτυο για άμεση ασοτολή τους, είτε να διαθέτει σειριακή θύρα όπου μια οποιαδήποτε έξυπνη συσκευή μπορεί να συνδεθεί για να «αιτηθεί» τα δεδομένα. Το συγκεκριμένο στάδιο της επικοινωνίας, εκτός από τη συλλογή των δεδομένων, αφορά και τη «γεφύρωση» του μηχανήματος με το δίκτυο επικοινωνιών. Ανάλογα με τον τρόπο που θα γίνει η λήψη και η μετάδοση των δεδομένων (εξαρτάται από το πρωτόκολλο που τίθεται σε ισχύ) ενσωματώνεται στο μηχανήμα και το κατάλληλο ασύρματο μοντέλο δεδομένων. Αν το ελεγχόμενο μηχανήμα είναι η master συσκευή του συστήματος, τότε διαχειρίζεται την M2M συσκευή ως μόντεμ φορτώνοντάς του τα δεδομένα ώστε αυτό να τα μεταφέρει στο δίκτυο. Αν η master συσκευή είναι η M2M, τότε αυτή κάνει περιοδικούς ελέγχους στη συσκευή ελέγχου/τους αισθητήρες/τους διακόπτες/κτλ και λαμβάνει τα δεδομένα. Στις εφαρμογές υψηλής επικινδυνότητας, η ροή δεδομένων από τη συσκευή ελέγχου προς την M2M πρέπει να είναι σταθερή και σε πραγματικό χρόνο. Ωστόσο, στις περισσότερες εφαρμογές αυτό δεν είναι απαραίτητο και γι'αυτό αποφεύγεται αφού ανεβάζει το κόστος λειτουργίας του συστήματος. Σε αυτές τις περιπτώσεις, λοιπόν, καθορίζονται κάποια συγκεκριμένα όρια για το εύρος τιμών που μπορεί να λαμβάνει η τιμή της μεταβλητής που μετράει η συσκευή ελέγχου και η συσκευή M2M εκπέμπει πληροφορία σε πραγματικό χρόνο μόνο αν η τιμή αυτή σε κάποιον από τους αναγνωστικούς ελέγχους αποδειχθεί ότι ξεπέρασε κάποιο από αυτά τα όρια. Παρ'όλα αυτά, η αποστολή στοιχείων είναι πάντα δυνατή σε οποιαδήποτε χρονική στιγμή ως απάντηση σε αίτημα που προέρχεται από τον server.

2. Μετάδοση δεδομένων: Στο στάδιο αυτό λαμβάνει χώρα η μεταφορά των δεδομένων από έναν εξοπλισμό –συνήθως βρίσκεται αρκετά απομακρυσμένος- στο κέντρο λειτουργίας του δικτύου μέσω ενός δικτύου επικοινωνίας. Οι πιο κοινές λύσεις αναφορικά με το δίκτυο επικοινωνίας είναι αυτό της κινητής τηλεφωνίας, οι τηλεφωνικές γραμμές και οι δορυφόροι επικοινωνίας. Η πιο οικονομική από αυτές είναι αυτή της σταθερής τηλεφωνίας γιατί υπάρχουν εγκατεστημένες γραμμές σχεδόν σε όλα τα κτίρια πλέον και το κόστος της χρήσης της μπορεί να διαμοιραστεί και σε άλλες χρήσεις. Ωστόσο, μπορεί η πιο δαπανηρή από αυτές τις επιλογές να είναι η δορυφορική επικοινωνία αλλά είναι η πιο αποτελεσματική –πολλές φορές και η μόνη- λύση για πολύ απομακρυσμένες περιοχές. Τέλος, το μεγάλο πλεονέκτημα της χρήσης του δικτύου κινητής τηλεφωνίας είναι η ευρεία γεωγραφική κάλυψη με την παράλληλη ολοένα συνεχιζόμενη πτώση των τιμολογίων των εταιρειών που παρέχουν τις υπηρεσίες. Στη Βόρεια Αμερική χρησιμοποιούνται ευρύτατα το CDMA και το GPRS λόγω της αυξημένης ικανότητάς τους να αποστέλουν μεγάλο όγκο δεδομένων. για τη σύνδεση στα δίκτυα κινητής είναι υποχρεωτική η ύπαρξη πύλης που θα λάβει τα δεδομένα από το ασύρματο δίκτυο επικοινωνίας, θα τα μετατρέψει σε μορφή που να μπορούν να αποσταλούν προς το κέντρο λειτουργίας του

δικτύου μέσω μιας σύνδεσης στο Internet ή μέσω μιας frame relay σύνδεσης. Σωστή επιλογή της πύλης και του χρησιμοποιούμενου software εγγυάται την ασφαλή μετάδοση των δεδομένων. Επίσης, σωστή επιλογή της πύλης εγγυάται και τη λειτουργία της ως μετατροπέας πρωτοκόλλου, αυτή τη φορά από υψηλού εύρους ζώνης Internet πρωτόκολλα σε χαμηλού εύρους ζώνης ασύρματα πρωτόκολλα όταν αντιστρέφεται η ροή των δεδομένων.

3. Αξιολόγηση των δεδομένων: Τα δεδομένα τα οποία μεταδόθηκαν στο προηγούμενο στάδιο εμφανίζονται είτε στο software της εφαρμογής είτε σε ένα αυτόνομο σύστημα ειδικά σχεδιασμένο για την εξυπηρέτηση της M2M επικοινωνίας. Ωστόσο, στις μέρες μας έχει επικρατήσει η τάση να προτιμούνται τα αυτόνομα συστήματα αφού αποφεύγεται έτσι το κόστος ενσωμάτωσης ενός M2M συστήματος σε ένα ήδη υπάρχον σύστημα.
4. Ανταπόκριση στη διαθέσιμη πληροφορία: Είναι το στάδιο της επικοινωνίας όπου ουσιαστικά αυτοματοποιείται η οποιαδήποτε ελεγχόμενη διαδικασία ανάλογα με τις εντολές που έχει λάβει το σύστημα κατά τον προγραμματισμό του.

Για τη βέλτιστη λειτουργία της παραπάνω διαδοχής των σταδίων επικοινωνίας, τα M2M δίκτυα θα πρέπει να είναι ασφαλή, εύχρηστα και αξιόπιστα. Αυτές οι απαιτήσεις θα πρέπει κάθε φορά να εξυπηρετούν συγκεκριμένους στόχους. Έτσι, από τη στιγμή που ένα M2M δίκτυο θα πρέπει να μπορεί να υποστηρίξει όσο το δυνατόν περισσότερες συσκευές, το δίκτυο θα πρέπει να υποστεί τις κατάλληλες βελτιστοποιήσεις ώστε να αποφεύγεται η υπερφόρτωση του συστήματος. Επίσης, οι M2M συσκευές θα πρέπει να είναι σε θέση να είναι - σχεδόν- μόνιμα συνδεδεμένες στο δίκτυο ώστε να μπορούν να πεταδίζουν ακόμη και τον ελάχιστο όγκο πληροφορίας όταν αυτό θεωρείται σημαντικό, να εκτελούν διαφορετικές εφαρμογές M2M με διαφορετικά χαρακτηριστικά και απαιτήσεις, να διαχειρίζονται ορθολογικά τους διαθέσιμους πόρους καθώς και να μπορούν να υποστηρίξουν προηγμένους μηχανισμούς ελέγχου και ασφάλειας αφού θα λειτουργούν χωρίς ανθρώπινη επίβλεψη. Τέλος, θα πρέπει οι M2M συσκευές να διατίθενται σε χαμηλές τιμές, όπως χαμηλές θα πρέπει να είναι και οι χρεώσεις των υπηρεσιών δικτύου.



Εικόνα 2: Τα στάδια της M2M επικοινωνίας σχηματικά.

1.3 Αρχιτεκτονική της M2M επικοινωνίας

Από όσα συζητήθηκαν στην προηγούμενη παράγραφο προκύπτει εύκολα το συμπέρασμα πως ένα δίκτυο M2M επικοινωνίας είναι ουσιαστικά ένα κράμα από διάφορες ετερογενείς τεχνολογίες. Το σύστημα αυτό περιλαμβάνει συσκευές, δίκτυα, πύλες επικοινωνίας και λειτουργικό σύστημα καθώς και το τελικό προϊόν της εφαρμογής. Τα δομικά αυτά στοιχεία περιγράφονται ακολούθως:

- **Συσκευές και μοντέλα επικοινωνίας:** Οι συσκευές που χρησιμοποιούνται στα M2M δίκτυα επικοινωνίας είναι εξοπλισμένες κάποιο μηχάνημα, κάποια συσκευή ή κάποια εφαρμογή που ενσωματώνει ένα μοντέλο επικοινωνίας όπως για παράδειγμα οι ενεργοποιητές, οι αισθητήρες, οι ετικέτες RFID, τα PLCs, τα μοντέλα I/O κτλ. Μέσω των αισθητήρων και των ενεργοποιητών, για παράδειγμα, μπορεί να γίνει η λήψη πληροφορίας και το σύστημα να εκτελέσει εντολές σύμφωνα με την επιθυμία του εκάστοτε ανθρώπινου παράγοντα ή ακόμα και εξαιτίας απουσίας κάποιας εντολής. Οι συσκευές των M2M συστημάτων είναι προγραμματισμένες να αντιδρούν σε μεταβολές των τιμών συγκεκριμένων μεγεθών, όπως για παράδειγμα η θερμοκρασία ή η πίεση, η ανίχνευση φωτός ή μη κτλ. Επίσης, με τη βοήθεια αισθητήρων κατασκευάζονται PLCs και GSM μηχανήματα για ασύρματη λειτουργία συσκευών. Τα ασύρματα, αυτά, μοντέλα επικοινωνίας μπορούν είτε να ενσωματωθούν στο σύστημα είτε να προστεθούν ως ξεχωριστά εξαρτήματα.
- **Δίκτυα:** Τα M2M συστήματα περιλαμβάνουν δύο ειδών δίκτυα. Το πρώτο είναι ένα δίκτυο επικοινωνίας το οποίο αποτελεί την κεντρική σύνδεση μεταξύ της συσκευής και του απομακρυσμένου χρήστη. Τυπικά δίκτυα επικοινωνιών είναι το xDSL, το IEEE802.11, τα τοπικά δίκτυα (LAN), τα W-LAN, το WiMAX, το Gern, το UTRAN, το GSM, το GPRS, το LTE κτλ. Υποχρέωση του δικτύου επικοινωνίας είναι η παραλαβή της πληροφορίας από την M2M πύλη (θα συζητηθεί παρακάτω) και η αποστολή αυτής σε έναν back-end server ο οποίος επεξεργάζεται τα δεδομένα και τα αποστέλει μέσω Internet στην κεντρική εγκατάσταση παρακολούθησης και ελέγχου του M2M συστήματος. Το δεύτερο είδος δικτύου που περιλαμβάνεται σε ένα σύστημα M2M είναι το M2M δίκτυο το οποίο είναι αυτό που συνδέει τη συσκευή με την M2M πύλη. Τυπικά M2M δίκτυα είναι το Wireless, το Zigbee, το Bluetooth, τα δίκτυα αισθητήρων κτλ. Η συνδεσιμότητα των συσκευών μπορεί να επωφελείται από όλα τα υπάρχοντα πρωτόκολλα (όπως, για παράδειγμα, το TCP/IP). Ωστόσο, στο εύρος των εφαρμογών του οικιακού αυτοματισμού που πραγματεύεται η εργασία μας, το πρωτόκολλο που χρησιμοποιείται είναι το X10 (θα συζητηθεί παρακάτω).
- **Πύλη M2M:** Η πύλη M2M παραλαμβάνει τα πρωτογενή δεδομένα από τη συσκευή και τα προετοιμάζει για αποστολή μέσω του δικτύου επικοινωνίας. Στην πύλη βρίσκεται εγκατεστημένο και το πρωτόκολλο το οποίο θα μεταφράσει τα δεδομένα σε μορφή ικανή για αποστολή και

αναγνώριση από τη συσκευή που θα τα παραλάβει ή ακόμη και κατανοητά από τον άνθρωπο.

- **Λογισμικό:** Το M2M σύστημα διαθέτει ένα πλήθος εφαρμογών το οποίο αποσκοπεί κατά κύριο λόγο στην επικοινωνία της οποιασδήποτε πληροφορίας με τους χρήστες. Η επικοινωνία αυτή μπορεί να γίνεται είτε αυτόματα είτε μετά από σχετικό αίτημα του χρήστη σε οποιοδήποτε επίπεδο λειτουργίας του συστήματος. Σημαντικό ρόλο σε αυτό το στοιχείο του συστήματος παίζει η εξέλιξη στις γλώσσες προγραμματισμού και ιδιαίτερα η ανάπτυξη στις γλώσσες web προγραμματισμού καθώς και των διαφόρων υπηρεσιών web αφού με αυτόν τον τρόπο είναι δυνατή η ανταλλαγή δεδομένων μέσω του διαδικτύου καθιστώντας ευκολότερη τη διαχείριση των διαφόρων δεδομένων από τις εταιρείες που παρέχουν τα M2M προϊόντα. Σε αυτή την περίπτωση προτιμούνται οι ανεξάρτητες γλώσσες προγραμματισμού οι οποίες μπορούν να χρησιμοποιηθούν τόσο σε ένα μεγάλο εύρος εφαρμογών, όσο και σε ένα ευρύ φάσμα λειτουργικών προγραμμάτων. Ένα τέτοιο παράδειγμα ανεξάρτητης γλώσσας προγραμματισμού είναι η Java η οποία μπορεί να χρησιμοποιηθεί για το «χτίσιμο» σχεδόν κάθε εφαρμογής ενώ είναι συμβατή τόσο με Unix όσο και με MacOS και Windows. Ένα ξεχωριστό τμήμα του τρέχοντος λογισμικού του συστήματος είναι το Middleware το οποίο παρέχει δρομολόγηση και buffering στα δεδομένα μεταξύ των απομακρυσμένων μηχανών και των κεντρικών συστημάτων πληροφορικής. Ουσιαστικά, το Middleware πρέπει να φροντίζει ώστε τα πολυάριθμα μηχανήματα από τα οποία συνίσταται ένα M2M σύστημα να μπορούν να επικοινωνούν αποδοτικά μεταξύ τους αφού γεφυρώνει τα διάφορα πρωτόκολλα μεταξύ των διαφορετικών υποσυστημάτων που περιλαμβάνονται. Τέλος, στο λογισμικό των M2M συστημάτων περιλαμβάνονται συγκεκριμένα πακέτα τα οποία ουσιαστικά κλειδώνουν τις εφαρμογές προς όφελος της εταιρίας που της ανέπτυξε. Παραδείγματα τέτοιων πακέτων είναι το EPR, το CRM και το SCM.
- **Εφαρμογές για απομακρυσμένους χρήστες:** Έχει τη μορφή είτε λογισμικού είτε συγκεκριμένου υλικού που παραλαμβάνει την πληροφορία από το δίκτυο επικοινωνίας και τη μεταφέρει στους απομακρυσμένους χρήστες. Ως παραλήπτες μπορούν να θεωρηθούν τα κινητά τηλέφωνα, οι web browsers, το ηλεκτρονικό ταχυδρομείο καθώς και τα μηνύματα ειδοποιήσεων τύπου SMS. Αυτοί οι χρήστες χρησιμοποιούν μέρος του λογισμικού του συστήματος για να λάβουν τα δεδομένα, να αναλυθούν, να αναφερθούν και να δράσουν.

ΚΕΦΑΛΑΙΟ 2

Ο ΥΠΟΛΟΓΙΣΤΗΣ RASPBERRY PI

2.1 Εισαγωγή

Μέχρι τη δεκαετία του 1970, οι υπολογιστές αποτελούσαν αποκλειστικό προνόμιο ενδοπαανεπιστημιακών ερευνητικών εργαστηρίων καθώς και συγκεκριμένων μεγάλων εταιριών. Από το 1970, όμως, και μετά η κατάσταση αυτή άλλαξε άρδην. Η αλλαγή αυτή υποκινήθηκε κυρίως από την κυκλοφορία της TVT (TV Typewriter) και του Apple I και στη συνέχεια του Apple II. Έτσι, η δεκαετία του 1980 έφερε χαμηλού κόστους προσωπικούς υπολογιστές για οικιακή χρήση όπως ο ZX Spectrum και ο Commodore 64 και ταυτόχρονη ανάπτυξη στον κλάδο των προγραμματιστών. Μέχρι το 1990, ο προγραμματισμός είχε ήδη γίνει αναπόσπαστο κομμάτι της επιστήμης των υπολογιστών τόσο σε ακαδημαϊκό όσο και σε βιομηχανικό επίπεδο, γεγονός που οδήγησε και στη διαδικτυακή επανάσταση που συντελέστηκε την εποχή εκείνη.

Εμπνευσμένη από αυτή την πορεία την επιστήμης του προγραμματισμού και θέλοντας να παρέχει έναν οικονομικό μικρο-υπολογιστή με δυνατότητα να υποστηρίξει ελαφριά λειτουργικά συστήματα στους ερασιτέχνες προγραμματιστές, τους φοιτητές και τα παιδιά, μια ομάδα επιστημόνων των υπολογιστών υπό την καθοδήγηση του Eben Upton στο Εργαστήριο Υπολογιστών του Πανεπιστημίου του Cambridge ξεκίνησε το 2006 την προσπάθεια κατασκευής του Raspberry Pi. Το 2008, το εγχείρημα αυτό άρχισε να παίρνει σάρκα και οστά με αποτέλεσμα το λανσάρισμα του Raspberry Pi την 29^η Φλεβάρη του 2012. Σημαντική βοήθεια στην επιτυχία αυτής της προσπάθειας προσέφερε η φοβερή ανάπτυξη που είχε την εποχή εκείνη η κινητή τηλεφωνία, αφού χρησιμοποιήθηκαν μονάδες CPU της βρετανικής εταιρείας ARM που αρχικά προορίζονταν για κινητά τηλέφωνα. Αυτό είχε σημαντική επίπτωση στη μείωση της τιμής του Raspberry Pi αφού μπορούσε να χρησιμοποιήσει ήδη υπάρχουσα τεχνογνωσία. Έτσι, ιδρύθηκε και επίσημα το «Ίδρυμα Raspberry Pi» το οποίο έχει σαν κύριο στόχο την κατασκευή και βελτίωση των Raspberry Pi υπολογιστών.

Το πρώτο μοντέλο Raspberry Pi, το Alpha model, κατασκευάστηκε το 2011 και δόθηκε στο κοινό για να γνωρίσει αυτό το νέο είδος μικροϋπολογιστή και να τροφοδοτήσει την κατασκευάστρια εταιρεία με πολύτιμη ανάδραση. Οι δοκιμές αυτές κατέληξαν στο συμπέρασμα πως επρόκειτο για έναν μικροσκοπικό υπολογιστή ο οποίος μπορούσε να προσφέρει πάρα πολλά σε σύγκριση με το πολύ μικρό του κόστος. Τελικά, το 2012 ο Raspberry Pi ήταν έτοιμος για την πρώτη έξοδό του στο εμπόριο για ευρεία κατανάλωση με τη μορφή δύο μοντέλων, το A model και το B model. Τα δύο μοντέλα διέφεραν μεταξύ τους μόνο στο ότι το A δε διέθετε θύρα Ethernet, διαφοροποίηση που μεταφράστηκε σε μια διαφορά στην τιμή του ύψους των \$10. Η λατασκευή τους ξεκίνησε στην Κίνα ενώ συνεχίστηκε επί βρετανικού εδάφους όταν τη διαδικασία ανέλαβε η

Sony. Μετά από αρκετές βελτιώσεις και συμμορφώσεις σε κανονισμούς, ο Raspberry Pi είναι πλέον στη διάθεση του καταναλωτικού κοινού και απολαμβάνει σημαντικής αποδοχής.

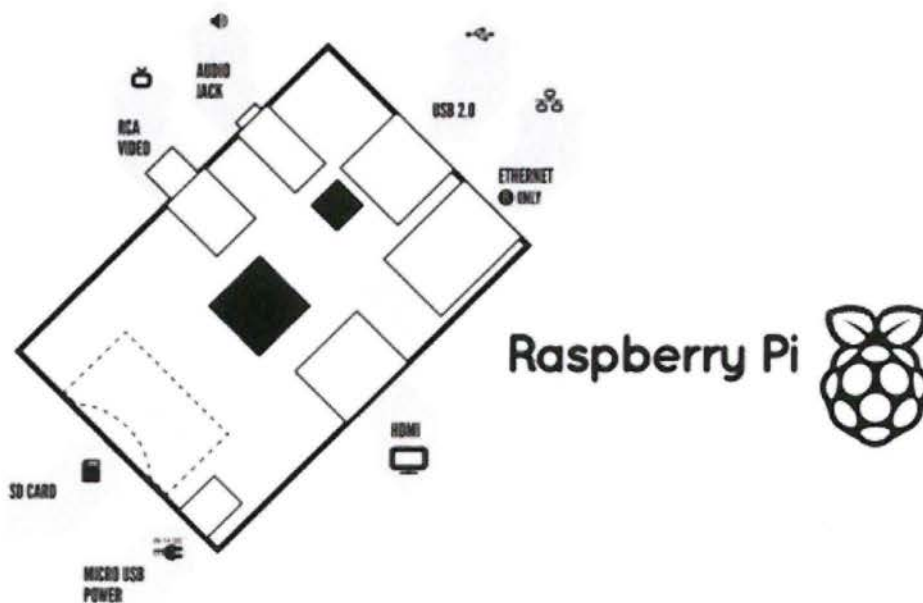


Εικόνα 3: Το B model του Raspberry Pi.

2.2 Το Hardware του Raspberry Pi

Η κατασκευή του Raspberry Pi έχει βασιστεί στον Broadcom BCM2835, έναν επεξεργαστή για εφαρμογές πολυμέσων που προορίζεται για κινητές συσκευές. Πάνω σε αυτόν έχουν προστεθεί διάφορα άλλα στοιχεία ώστε να στηριχθούν τεχνολογίες όπως η τοποθέτηση κάρτας SD, η ύπαρξη θύρας USB και θύρας RCA.

Ο υπολογιστής Raspberry Pi έχει, σχηματικά, την παρακάτω μορφή:



Εικόνα 4: Σχηματική αναπαράσταση του υπολογιστή Raspberry Pi.

Οι διαστάσεις του είναι αρκετά μικρές, $85.60mm \times 53.98mm \times 17mm$ με βάρος μόλις 45gr. Οι διαστάσεις του αυτές είναι που τον καθιστούν ιδανικό στοιχείο συστημάτων αυτοματισμού μικρών εγκαταστάσεων (όπως στην περίπτωση μας που μας αφορούν εφαρμογές οικιακού αυτοματισμού) αφού στις συγκεκριμένες εφαρμογές χρειαζόμαστε πολύ μικρές διατάξεις οι οποίες θα τοποθετηθούν εντός ήδη υπάρχουσας συσκευής ή διάταξης (για παράδειγμα η αντικατάσταση ενός θερμοστάτη).

Ο υπολογιστής Raspberry Pi διαθέτει θύρες εισόδου/εξόδου για διαφόρων τύπων επικοινωνίες του συστήματος έτσι ώστε να είναι δυνατή η υλοποίηση διαφορετικών ειδών εφαρμογών. Διαθέτει αναλογική θύρα ήχου 3.5 mm για τη χρησιμοποίηση ακουστικών και μικροφώνων, ιδιαίτερα χρήσιμη για τις εφαρμογές πολυμέσων. Επίσης, διαθέτει μια θύρα RCA έτσι ώστε να μπορεί να χρησιμοποιηθεί η τηλεόραση ως μόνιτορ. Αναφορικά με τη USB συνδεσιμότητά του, ο Raspberry Pi διαθέτει δύο θύρες USB 2.0 και μια micro USB οι οποίες επιτρέπουν τη σύνδεση πληκτρολογίου και ποντικιού και σύνδεση με πηγή ηλεκτρικού ρεύματος αντίστοιχα. Υπάρχει ακόμα μια θύρα HDMI (High Definition Multimedia Interface) η οποία καθιστά δυνατή τη σύνδεση του συστήματος με τηλεοράσεις και μόνιτορ υψηλής ευκρίνειας. Αυτή η θύρα χρησιμοποιείται όταν επιθυμείται η αναπαραγωγή ολυμέσων με ήχο και εικόνα ταυτόχρονα. Η επικοινωνία του υπολογιστή με άλλες συσκευές καθώς και η σύνδεσή του στο διαδύκτιο γίνονται μέσω μιας θύρας Ethernet. Μέσω αυτής ο Raspberry Pi συνδέεται με τον κοινό οικιακό δρομολογητή (router) που χρησιμοποιείται για τη σύνδεση οποιουδήποτε προσωπικού υπολογιστή στο διαδίκτυο. Ωστόσο, θύρα Ethernet διαθέτει μόνο το model B, το model A συνδέεται στο δίκτυο μέσω εξωτερικής θύρας Ethernet συνδεδεμένης σε μια από τις USB θύρες είτε με προσαρμογέα Wi-Fi πάλι συνδεδεμένο σε κάποια από τις θύρες USB. Ο κύριος τρόπος αποθήκευσης δεδομένων του υπολογιστή Raspberry Pi είναι μέσω χρήσης κάρτας SD η οποία τοποθετείται στην κατάλληλη SD θύρα. Αυτός ακριβώς ο αποθηκευτικός χώρος είναι το σημείο όπου θα εγκατασταθεί το λειτουργικό σύστημα του υπολογιστή και κατ'επέκταση θα λειτουργεί ως ο σκληρός του δίσκος. Επειδή, όμως, μια κάρτα SD προσφέρει περιορισμένο αποθηκευτικό χώρο, υπάρχει πάντα η δυνατότητα επέκτασης αυτού μέσω εξωτερικού σκληρού δίσκου συνδεδεμένου με το σύστημα μέσω των θυρών USB. Τα προηγούμενα μοντέλα Raspberry Pi είναι εφοδιασμένα με προεγκατεστημένη μνήμη SDRAM 256 MB ενώ τα πιό σύγχρονα με SDRAM 512 MB. Αυτή η προεγκατεστημένη μνήμη συγκρινόμενη με τη μνήμη που διαθέτει ένας σύγχρονος προσωπικός υπολογιστής είναι ελάχιστη. Ωστόσο, για το εύρος των εφαρμογών για τις οποίες προορίζεται ένας Raspberry Pi υπολογιστής τόσο η μνήμη των 256 MB όσο και η μνήμη των 512 MB θεωρούνται κάτι παραπάνω από αρκετές. Αξίζει να σημειωθεί, βέβαια, πως στις 15 Οκτώβρη 2012, το Ίδρυμα Raspberry Pi ανακοίνωσε πως όλα τα μοντέλα του θα κατασκευάζονται πια με μνήμη 512 MB.

Όπως προαναφέρθηκε, ο κεντρικός επεξεργαστής (CPU) ενός υπολογιστή Raspberry Pi είναι βρετανικής κατασκευής και πιο συγκεκριμένα είναι το μοντέλο ARM1176JZF-S-core στα 700 MHz της σειράς 32-bit πολυεπεξεργαστών ARM 11 της εταιρείας ARM. Ο CPU είναι υπεύθυνος για την εκτέλεση των εντολών κάθε προγράμματος που φέρει ο Raspberry Pi μέσω μαθηματικών και λογικών εντολών. Τέλος, έχει ενσωματωθεί η καρτα γραφικών Broadcom

VideoCore IV, η οποία είναι ειδικά σχεδιασμένη ώστε να επιταχύνεται η διαχείριση εικόνων. Η δυνατότητα αυτή είναι ιδιαίτερα σημαντική για τις εφαρμογές 3D γραφικών, ή για χρησιμοποίηση του υπολογιστή ως κονσόλα παιχνιδιών.

Ο υπολογιστής Raspberry Pi, όμως, διαθέτει και θέσεις σύνδεσης ηλεκτρονικού υλικού, όπως το Arduino που θα χρησιμοποιηθεί και στη δικιά μας κατασκευή. Τα GPIO pins (General Purpose Input/Output pins) προγραμματίζονται στον Raspberry Pi με εντολές εισόδου κι εξόδου. Οι ασπίδες του Arduino συνδέονται στα GPIO pins μέσω γεφυρών επιτρέποντα τη μεταφορά δεδομένων από τους αισθητήρες που βρίσκονται επί της συσκευής που χρησιμοποιούμε προς τον Raspberry Pi. Αυτή η δυνατότητα είναι ιδιαίτερα σημαντική στις εφαρμογές οικιακού αυτοματισμού όπου επιθυμούμε είτε την αποθήκευση δεδομένων είτε το χειρισμό μηχανημάτων βάσει προγραμμάτων που είναι εγκατεστημένα στο σκληρό δίσκο του Raspberry Pi.

Συγκεντρωτικά, για τις προδιαγραφές του υπολογιστή Raspberry Pi μπορούμε να δούμε τον παρακάτω πίνακα:

13	Model A	Model B
CPU	700 MHz ARM1176JZF-S-core	
GPU	250 MHz Broadcom VideoCore IV OpenGL ES 2.0 MPEG-2 VC-1	
SDRAM	256 MB	512 MB
SoC	Broadcom BCM2835	
Θύρες USB	1	2
Είσοδος Εικόνας	CSI	
Έξοδος Εικόνας	Composite RCA HDMI	
Έξοδος Ήχου	3.5 mm jack HDMI I ² S	
Μνήμη	SD	
Ethernet	Όχι	10/100 Ethernet (8P8C)
Διαστάσεις	85.60mm×53.98mm×17mm	
Βάρος	45 γρ	
Λειτουργικό Σύστημα	Arch Linux ARM, Debian GNU/Linux, Fedora, FreeBSD, NetBSD, Plan 9, Raspbian OS, RISC OS,[30] Slackware Linux	
Τιμή	\$ 25.00	\$35.00

Πίνακας 1: Προδιαγραφές Υπολογιστή Raspberry Pi.

2.3 Το Software του Raspberry Pi

Κατά το σχεδιασμό του υπολογιστή Raspberry Pi στοχεύθηκε η δημιουργία ενός συστήματος το οποίο θα δύναται να λειτουργεί με πολύ ελαφριά λειτουργικά συστήματα. Έτσι, ο υπολογιστής αυτός χρησιμοποιεί κατά κύριο λόγο λειτουργικά συστήματα βασισμένα στον πυρήνα του Linux.

Το Ίδρυμα Raspberry Pi δημιούργησε έναν ιδιαίτερα εύχρηστο installer προγραμμάτων, τον NOOBS (New Out Of Box System), ο οποίος διευκολύνει ιδιαίτερα την εγκατάσταση λειτουργικών συστημάτων και από τον Ιούλιο του 2013 συνιστά τη χρήση αυτού για την εγκατάσταση Raspbian στα υπολογιστικά του συστήματα. Με αυτή τη διαδικασία, το Ίδρυμα προσπαθεί να ξεκινήσει την ανάπτυξη ενός καταστήματος εφαρμογών όπου οι χρήστες θα μπορούν να ανταλλάσσουν προγράμματα.

Το Raspbian είναι ένα λειτουργικό σύστημα βασισμένο σε βελτιστοποίηση του Debian ώστε να μπορεί να υποστηριχθεί από έναν υπολογιστή Raspberry Pi. Είναι το λειτουργικό σύστημα το οποίο -προς το παρόν- συνιστά το Ίδρυμα και λανσαρίστηκε επίσημα τον Ιούλιο του 2012 παρά το γεγονός ότι ήταν ακόμα υπό ανάπτυξη. Είναι ελεύθερο λογισμικό του οποίου η ανάπτυξη γίνεται εξ'ολοκλήρου από το Ίδρυμα και βασίζεται στην αρχιτεκτονική του Debian 7 σε περιβάλλον LXDE. Το Raspbian διαθέτει μερικά deb πακέτα προγραμμάτων και απαιτεί μια ελάχιστη μνήμη SD κάρτας των 2 GB. Ωστόσο, συνιστάται η χρήση SD κάρτας από 4 GB και πάνω.

Εκτός, όμως, από το Raspbian, ο NOOBS εγκαθιστά μερικά ακόμα λειτουργικά συστήματα, ανάλογα με την προτίμηση του χρήστη. Αυτά είναι το Archlinux ARM, το OpenELEC, το Pidora (λειτουργικό σύστημα βασισμένο στο Fedora), το Raspbmc και το RISC OS. Ωστόσο, το Ίδρυμα συνιστά τη χρήση Raspbian.

Επίσης, υπάρχει διαθέσιμο λογισμικό και από άλλες εταιρείες που έχουν ασχοληθεί με τα συστήματα Raspberry Pi. Έτσι, διατίθεται το Raspbian Server Edition (RSEv2.4) το οποίο είναι μια έκδοση του Raspbian με ενσωματωμένα πακέτα λογισμικού. Το Slackware ARM (έκδοση 13.37 ή νεότερη) είναι συμβατό με τον Raspberry Pi χωρίς καμμία τροποποίηση. Το λογισμικό αυτό χρειάζεται μνήμη 64 MB σε έναν ARM επεξεργαστή όπως αυτός που χρησιμοποιείται στον Raspberry Pi με αποτέλεσμα τα 128-496 MB που είναι διαθέσιμα στη μνήμη του συγκεκριμένου υπολογιστή να είναι παραπάνω από αρκετά. Αυτή η εξοικονόμηση μνήμης προέρχεται από το γεγονός ότι, ενώ τα υπόλοιπα λειτουργικά συστήματα που βασίζονται σε πυρήνα Linux φορτώνονται μέσω διεπαφής με το χρήστη με απαιτήσεις από την κάρτα γραφικών, το Slackware ARM λειτουργεί στο περιβάλλον του ίδιου του φλοιού και σε παράθυρο εντολών. Επιπροσθέτως, αρκετές ομάδες εργάζονται πάνω σε ελαφριές εκδόσεις Linux όπως το IPFire, το OpenELEC, το Raspbmc καθώς και το ελεύθερου κώδικα κέντρο πολυμέσων XBMC.

Ιδιαίτερης σημασίας ήταν η ανακοίνωση που εξέδωσε το Ίδρυμα στις 24 Οκτώβρη 2012 όπου το κοινό ενημερώθηκε πως εφεξής το συμβατό με ARM VideoCore λανσάρεται ως ελεύθερο λογισμικό με άδεια τύπου BSD. Έτσι, πλέον, το καταναλωτικό κοινό έχει στα χέρια του το πρώτο SoC πολυμέσων βασισμένο σε επεξεργαστή ARM με πλήρως λειτουργικό σύστημα ελεύθερου κώδικα.

Υπάρχουν, βέβαια, ενστάσεις αναφορικά με την εγκυρότητα ή μη της δήλωσης αυτής. Ωστόσο, το γεγονός της ύπαρξης λογισμικού ελεύθερου κώδικα καθεαυτό είναι πολύ σημαντικό αναφορικά με τις εφαρμογές οικιακού αυτοματισμού.

Τέλος, μια ακόμη προσθήκη στα υποστηριζόμενα λογισμικά έγινε στις 21 Νοέμβρη 2013 όταν και ενσωματώθηκε στο Raspbian πλήρης έκδοση του μαθηματικού πακέτου Mathematica της Wolfram.

2.4 Εξαρτήματα για τον υπολογιστή Raspberry Pi

Το Μάη του 2012, το Ίδρυμα Raspberry Pi ανακοίνωσε πως έχει ήδη δοκιμαστεί μια πρωτότυπη κάμερα για σύνδεση με τον υπολογιστή. Η κάμερα αυτή είχε ανάλυση 14 megarixels, ωστόσο η κάμερα που θα λανσαριστεί αρχικά αναμένεται να είναι ανάλυσης 5 megarixels και είναι κατασκευής των εταιρειών RS Components και Premier Farnell/Element 14. Συνδέεται μέσω επίπεδου καλωδίου στη θύρα CSI ανάμεσα στις Ethernet και HDMI, υποστηρίζει βίντεο 1080p, 720p και 640×480p ενώ η τιμή της στην Ευρώπη είναι στα 20 Euro.

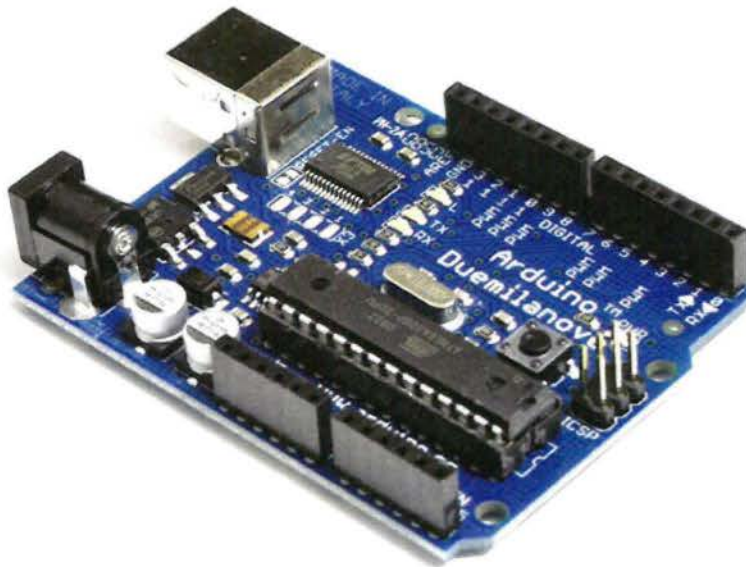
Επίσης, υπάρχει μεγάλη γκάμα περιφερειακών συσκευών και θηκών. Σε αυτές περιλαμβάνεται το Gertboard κατασκευής του Ιδρύματος Raspberry Pi το οποίο έχει αναπτυχθεί για εκπαιδευτικούς σκοπούς και επεκτείνει τις δυνατότητες των GPIO pins του υπολογιστή έτσι ώστε να επιτρέπεται η διασύνδεση με LEDs, διακόπτες, αισθητήρες και λοιπές συσκευές και τον πλήρη έλεγχο αυτών.

ΚΕΦΑΛΑΙΟ 3

Η ΠΛΑΤΦΟΡΜΑ ARDUINO

3.1 Εισαγωγή

Το Arduino είναι μια ανοιχτού κώδικα υπολογιστική πλατφόρμα πρωτοτυποποίησης ηλεκτρονικών βασισμένη σε μια απλή μητρική ρλακέτα με ενσωματωμένο μικροελεγκτή και GPIO pins προγραμματιζόμενη σε C++ ή ακριβέστερα σε Wiring (τροποποιημένη C++). Είναι ιταλικής προέλευσης και κατασκευάζεται από την εταιρεία Smart Projects. Χρησιμοποιείται για την ανάπτυξη διαδραστικών αντικειμένων ενώ έχει δυνατότητα σύνδεσης με υπολογιστή. Οι περισσότερες εκδόσεις του υπάρχουν προσυναρμολογημένες. ωστόσο, υπάρχουν ελεύθερα διαθέσιμα τόσο τα υλικά όσο και το διάγραμμα για τον καταναλωτή που επιθυμεί να συναρμολογήσει ο ίδιος τη δική του πλακέτα. Τα σχέδια αναφοράς του υλισμικού του Arduino διανέμονται υπό την Creative Commons Attribution Share-Alike 2.5 άδεια και είναι διαθέσιμα στην ιστοσελίδα του Arduino. Αφού κατασκευαστεί, μπορεί να συμπεριφερθεί σαν ένας μικροσκοπικός υπολογιστής, αφού ο χρήστης μπορεί να συνδέσει επάνω του πολλαπλές μονάδες εισόδου/εξόδου και να προγραμματίσει τον μικροελεγκτή να δέχεται δεδομένα από τις μονάδες εισόδου, να τα επεξεργάζεται και να στέλνει κατάλληλες εντολές στις μονάδες εξόδου.



Εικόνα 1: Η πλακέτα Arduino Duemilanove.

Το Arduino βέβαια, δεν είναι ούτε ο μοναδικός, και κατά πολλούς ούτε και ο καλύτερος δυνατός τρόπος για την δημιουργία μιας οποιασδήποτε διαδραστικής ηλεκτρονικής συσκευής. Όμως το κύριο πλεονέκτημά του είναι η τεράστια κοινότητα που το υποστηρίζει και η οποία έχει δημιουργήσει, συντηρεί και επεκτείνει μια ανάλογου μεγέθους online γνωσιακή βάση. Έτσι, παρότι ένας έμπειρος ηλεκτρονικός μπορεί να προτιμήσει διαφορετική πλατφόρμα ή

εξαρτήματα ανάλογα με την εφαρμογή την οποία επιθυμεί να αναπτύξει, το Arduino, με το εκτενές documentation με το οποίο υποστηρίζει τους καταναλωτές του, καταφέρνει να κερδίσει όλους αυτούς των οποίων οι γνώσεις στα ηλεκτρονικά περιορίζονται σε ερασιτεχνικά επίπεδα.

Η ιστορία του Arduino ξεκινά το 2005 όταν οι Massimo Banzi και David Cueartielles προσπάθησαν να κατασκευάσουν μια οικονομική συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές. Η κατασκευή της συσκευής ξεκίνησε με το όνομα Arduin στην Ιταλία. Η υλοποίηση της πλατφόρμας βασίστηκε στην πλατφόρμα Wiring και έχει προγραμματιστεί χρησιμοποιώντας την ομώνυμη γλώσσα προγραμματισμού. Με το Arduin ξεκίνησε ένα ντόμινο εκδόσεων. Έτσι, το 2006 έγινε το λανσάρισμα του Arduino Mini ενώ το 2008 του Arduino Duemilanove. Το 2009 ήταν η χρονιά του Arduino Mega ενώ ακολούθησαν το Arduino Leonardo, το Arduino Due και το Arduino Micro το 2012. Πιο πρόσφατες από όλες τις εκδόσεις είναι το Arduino Robot και το Arduino Yun που ανακοινώθηκαν το 2013.

3.2 Το Hardware του Arduino

Κεντρικό συστατικό μια πλακέτας Arduino είναι ο μικροελεγκτής Atmel AVR – μοντέλο ATmega328, ένας μικροελεγκτής 8-bit RISC που χρονίζει στα 16 MHz. Τον μικροελεγκτή συνοδεύουν συμπληρωματικά εξαρτήματα για τη διευκόλυνση του χρήστη κατά τον προγραμματισμό αλλά και την ενσωμάτωσή του σε άλλα συστήματα, όπως για παράδειγμα σε έναν υπολογιστή Raspberry Pi. Ο ATmega328 διαθέτει ενσωματωμένη μνήμη τριών ειδών. Διαθέτει μνήμη SRAM των 2 KB που αποτελεί την ωφέλιμη μνήμη που χρησιμοποιούν τα προγράμματα ώστε να αποθηκεύουν μεταβλητές, πίνακες κτλ όταν τρέχουν. Η μνήμη αυτή καθαρίζει όταν διακοπεί η παροχή ρεύματος στο Arduino ή όταν γίνει επανεκκίνηση του συστήματος. Διαθέτει, επίσης, μνήμη EEPROM μεγέθους 1 KB, η οποία μπορεί να χρησιμοποιηθεί για «ωμή» ανάγνωση και εγγραφή δεδομένων από τα προγράμματα του συστήματος ενώ δεν καθαρίζει με τρόπο ίδιο με την SRAM. Τέλος, διαθέτουν μνήμη Flash των 32 KB εκ των οποίων τα 2 KB χρησιμοποιούνται από τον κατασκευαστή για την εγκατάσταση του firmware του Arduino. Το firmware αυτό είναι το bootloader του Arduino στο οποίο θα αναφερθούμε ακολούθως. Τα υπόλοιπα 30 KB χρησιμοποιούνται για την αποθήκευση των προγραμμάτων του εκάστοτε χρήστη στο σύστημα, αφού βέβαια πρώτα μεταγλωττιστούν από αυτό. Όπως και η μνήμη EEPROM, δεν καθαρίζει από τα δεδομένα της με την απώλεια τροφοδοσίας ή με το reset του συστήματος. Είναι σημαντικό, όμως, να τονιστεί ότι ενώ η μνήμη Flash δε χρησιμοποιείται για την αποθήκευση δεδομένων κατά το «τρέξιμο» των προγραμμάτων έχει κατασκευαστεί μια βιβλιοθήκη ώστε να είναι δυνατή η χρήση όσου χώρου εξ'αυτής περισσεύει.

Ο μικροελεγκτής αυτός είναι προγραμματισμένος με ένα bootloader έτσι ώστε να μη χρειάζεται εξωτερικός hardware programmer. Έτσι είναι δυνατή η εγκατάσταση προσωπικών προγραμμάτων στο Arduino μέσω της θύρας USB. Όλες οι πλακέτες είναι προγραμματισμένες με μια RS-232 σειριακή σύνδεση, ωστόσο ο τρόπος με τον οποίο έχει γίνει ο προγραμματισμός αυτός διαφέρει

ανάλογα με την έκδοση. Οι πιο πρόσφατες συσκευές προγραμματίζονται μέσω USB και με τη βοήθεια της εφαρμογής προσαρμοστικών chip USB-to-Serial ενώ υπάρχουν και συσκευές, όπως το Arduino Mini, που χρησιμοποιούν είτε ένα αφαιρούμενο καλώδιο USB-to-Serial είτε Bluetooth είτε κάποια άλλη μέθοδο.

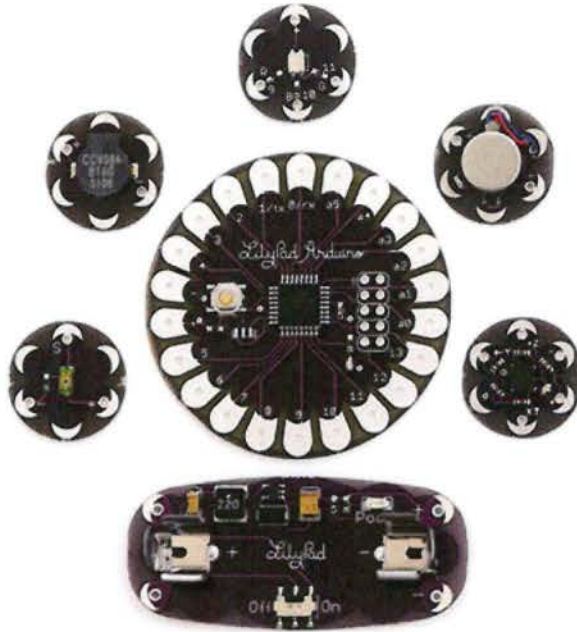
Ένα ενδιαφέρον στοιχείο της πλακέτας Arduino είναι ότι διαθέτει τα περισσότερα GPIO pins από κάθε άλλο κύκλωμα. Αυτά, βρίσκονται στην κορυφή της πλακέτας μέσω female headers 2.2 mm ενώ τα Arduino Nano, Arduino-Compatible Bare Bones Board και Boarduino Board ενδέχεται να έχουν και male header pins στο κάτω μέρος της πλακέτας ώστε να συνδέονται σε Breadboards. Λειτουργούν στα 5V και καθένα μπορεί να παρέχει ή να δεχτεί το πολύ 40mA. Το στοιχείο αυτό είναι αρκετά σημαντικό αφού προσφέρει πολύ μεγάλη συνδεσιμότητα της πλακέτας του Arduino με άλλα κυκλώματα ακι boards. Υπάρχουν πολλά boards συμβατά με το Arduino ενώ υπάρχουν και αρκετά τα οποία προέρχονται από Arduino boards. Από αυτά, κάποια είναι λειτουργικά ισάξια μεταξύ τους, άλλα είναι ηλεκτρικά ισάξια ενώ υπάρχουν και συσκευές με τελείως διαφορετικούς επεξεργαστές και κατ'επέκταση διαφορετικά επίπεδα συμβατότητας.

Πλακέτα	Στοιχεία
Serial Arduino	Προγραμματισμένο με ATmega8 & σειριακή σύνδεση
Arduino Extreme	Προγραμματισμένο με ATmega8 & USB interface
Arduino Mini	Προγραμματισμένο με surface mounted ATmega168
Arduino Nano	Προγραμματισμένο με surface mounted ATmega168
LilyPad Arduino	Προγραμματισμένο με surface mounted ATmega328
Arduino NG	Προγραμματισμένο με ATmega8 & USB interface
Arduino NG Plus	Προγραμματισμένο με surface mounted ATmega168
Arduino Bluetooth	Προγραμματισμένο με surface mounted ATmega168
Arduino Diecimilla	Προγραμματισμένο με ATmega168 & USB interface
Arduino Duemillanuoove	Προγραμματισμένο με ATmega168 & USB interface
Arduino Mega	Προγραμματισμένο με surface mounted ATmega1280
Arduino Uno	Προγραμματισμένο με ATmega168 & USB interface
Arduino Mega 2560	Προγραμματισμένο με surface mounted ATmega2560
Arduino Leonardo	Προγραμματισμένο με ATmega32U4 chip
Arduino Esplora	Προγραμματισμένο με ATmega168 & USB interface
Arduino Due	Προγραμματισμένο με ATmega168 & USB interface

Πίνακας 1: Οι 16 επίσημες πλακέτες Arduino.

Αναφορικά με τις θύρες εισόδου και εξόδου του Arduino γενικότερα, θα πρέπει να τονιστεί ότι υπακούουν σε σειριακό interface αφού ο μικροελεγκτής ATmega328 υποστηρίζει σειριακή επικοινωνία, την οποία το Arduino προωθεί μέσα από έναν ελεγκτή Serial-over-USB ώστε να συνδέεται με τον υπολογιστή μέσω USB. Η σύνδεση αυτή χρησιμοποιείται για την μεταφορά των

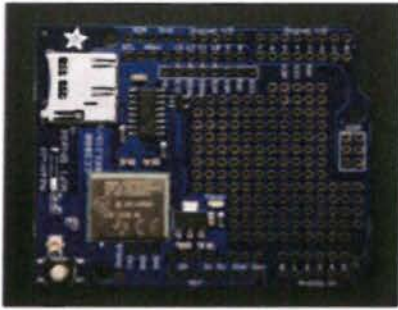
προγραμμάτων που σχεδιάζονται από τον υπολογιστή στο Arduino αλλά και για αμφίδρομη επικοινωνία του Arduino με τον υπολογιστή μέσα από το πρόγραμμα την ώρα που εκτελείται.



Εικόνα 2: Το kit του LilyPad Arduino, μια πολύ έξυμνη συσκευή για εφαρμογές ένδυσης και E-Textiles. Το εύρος των εφαρμογών Arduino αποδεικνύεται ιδιαίτερα ευρύ.

Οι πλακέτες Arduino καθώς και όλα τα συμβατά boards χρησιμοποιούν την τεχνολογία των ασπίδων, οι οποίες είναι τυπωμένες πλακέτες επεκτάσεων κυκλωμάτων που συνδέονται στα κανονικά παρεχόμενα Arduino pin-headers. Μερικές από τις πιο δημοφιλείς ασπίδες που υπάρχουν στο εμπόριο είναι η ασπίδα Ethernet (δίνει στο Arduino τη δυνατότητα να δικτυωθεί σε ένα δίκτυο LAN ή στο Internet μέσω ενός τυπικού καλωδίου Ethernet), η ασπίδα Wi-Fi (ουσιαστικά λειτουργεί όπως η Ethernet χωρίς να χρειάζεται χρήση καλωδίου), η ασπίδα GPS (δίνει τη δυνατότητα εντοπισμού στίγματος) και ασπίδες οθόνης (δίνουν τη δυνατότητα ενσωμάτωσης οθόνης στο σύστημα που κατασκευάζεται με το Arduino, η ασπίδα εξαρτάται από την οθόνη την οποία επιθυμεί ο χρήστης να συνδέσει και υπάρχει ένα εύρος αυτών από ασπίδες για απλού τύπου οθόνες όπως αυτή του υπολογιστή τσέπης τύπου calculator μέχρι ασπίδες για οθόνες αφής υψηλής ανάλυσης). Οι ασπίδες είναι ακτασκευασμένες με τέτοιο τρόπο ώστε αφού συνδεθούν με το Arduino να παρέχουν νέες θέσεις σύνδεσης ώστε να μπορούν να συνδεθούν επιπλέον εξαρτήματα ή ακόμα και επιπρόσθετες ασπίδες. Ωστόσο, αυτή η δυνατότητα συνδεσιμότητας δεν είναι απεριόριστη αφού κάθε ασπίδα χρησιμοποιεί ορισμένους από τους πόρους του Arduino. Επίσης, κάποιες από τις ασπίδες ίσως είναι ασύμβατες μεταξύ τους γιατί χρειάζονται να συνδεθούν σε ίδια pins του Arduino για να επιτευχθεί επικοινωνία με αυτό. Τέλος, υπάρχουν κάποιες ασπίδες που δεν υπάρχει νόημα να συνδεθούν με επιπλέον ασπίδα από πάνω. Ένα παράδειγμα είναι οι ασπίδες

οθονών αφού τότε η νέα επιπρόσθετη ασπίδα θα κάλυπτε την οθόνη. Γι'αυτές τις περιπτώσεις υπάρχουν ειδικές επεκτάσεις των ασπίδων που δίνουν τη δυνατότητα σε δύο ασπίδες (και όχι μια) να κουμπώσουν πάνω τους.



Εικόνα 2: Ασπίδες για Arduino. Η πρώτη (δεξιά) είναι μια ασπίδα Wi-Fi ενώ η δεύτερη (αριστερά) είναι μια ασπίδα για σύνδεση με 320×240 οθόνη αφής.

Το Arduino μπορεί να τροφοδοτηθεί με ρεύμα με δύο εναλλακτικούς τρόπους, είτε με ρεύμα μέσω τροφοδοτικού που συνδέεται σε υποδοχή τύπου φικς 2.1 mm (ένα κοινό τροφοδοτικό του εμπορίου είναι απόλυτα συμβατό με το Arduino) είτε μέσω της σύνδεσης USB από τον υπολογιστή με τον οποίο είναι συνδεδεμένο. Η εξωτερική τροφοδοσία πρέπει να είναι από 7 έως 12 V.

Τέλος, πάνω στην πλακέτα του Arduino, εκτός από το LED και την αντίσταση φορτίου που είναι συνδεδεμένα στο pin νούμερο 13, το Arduino διαθέτει ενσωματωμένα στην πλακέτα του και έναν διακόπτη και ακόμα 3 LED επιφανειακής στήριξης. Ο διακόπτης έχει αναλάβει τη λειτουργία reset ενώ το ένα από τα LED έχει την ένδειξη "power" οπότε η λειτουργία του είναι η αυτονόητη. Τα εναπομείναντα δύο LED χρησιμοποιούνται ως ένδειξη λειτουργίας του δειριακού interface καθώς ανάβουν όταν το Arduino στέλνει ή λαμβάνει δεδομένα μέσω USB.

3.3 Το Software του Arduino

Το ολοκληρωμένο περιβάλλον ανάπτυξης του Arduino (IDE) είναι προγραμματισμένο σε Java και είναι συμβατό με πολλές πλατφόρμες. Σκοπός της ανάπτυξης του Arduino ήταν αρχικά η κατασκευή ενός συστήματος το οποίο θα χρησιμοποιηθεί από μη εξειδικευμένους στην ανάπτυξη λογισμικού χρήστες, όπως οι μαθητές. Περιλαμβάνει έναν τυπικό συντάκτη κώδικα με δυνατότητα μεταγλώττισης και μπορεί να φορτώνει προγράμματα με ένα απλο πάτημα κουμπιού. Η διεπαφή με τον χρήστη είναι αρκετά φιλική και δεν απαιτείται η χρήση παραθύρου εντολών.

Κάθε πρόγραμμα γραμμένο για το Arduino ονομάζεται σκίτσο (sketch). Η γραφή του γίνεται σε C ή C++ ενώ χρησιμοποιείται η βιβλιοθήκη λογισμικού Wiring από το ομώνυμο πρωτότυπο σχέδιο της πλακέτας που διευκολύνει αρκετά πολλές κοινές λειτουργίες εισόδου/εξόδου. Το μόνο που απομένει να πραγματοποιηθεί από τη μεριά των χρηστών είναι η ενεργοποίηση ενός προγράμματος κυκλικής εκτέλεσης με τις εντολές `setup()` και `loop()`. Οι περισσότερες πλακέτες Arduino διαθέτουν ένα LED και μια αντίσταση φορτίου συνδεδεμένα μεταξύ του pin 13 και του εδάφους ώστε να μπορούν να χρησιμοποιηθούν για απλά τεστ των προγραμμάτων. Το IDE του Arduino χρησιμοποιεί το GNU toolchain και το AVR Libc για να μεταγλωττίζει προγράμματα και το avrdude για να φορτώνει προγράμματα στην πλακέτα. Δεδομένου ότι η πλατφόρμα Arduino χρησιμοποιεί Atmel μικροελεγκτές, το περιβάλλον ανάπτυξης της Atmel, το AVR Studio ή το νεότερη έκδοση του Atmel Studio, μπορεί επίσης να χρησιμοποιηθεί για την ανάπτυξη λογισμικού για το Arduino.

3.4 Σύνδεση του Arduino με υπολογιστή

Όπως προαναφέρθηκε, το Arduino διαθέτει πλήρες περιβάλλον ανάπτυξης το IDE. Εξ'αιτίας του IDE, το οποίο διατίθεται διαδικτυακά για τα τρία δημοφιλέστερα λειτουργικά συστήματα (Mac OS X, Linux & Windows) είναι δυνατή η πλήρης διαχείριση του Arduino από τον ηλεκτρονικό υπολογιστή του χρήστη. Το IDE είναι ένα περιβάλλον Java που προσφέρει στο χρήστη έναν πλήρη συντάκτη προγραμμάτων, με πολλά παραδείγματα και έτοιμες βιβλιοθήκες, προεγκατεστημένο μεταγλωττιστή των sketch και ένα σειριακό μόνιτορ. Επίσης, το IDE προσφέρει στο χρήστη τη δυνατότητα να εγκαταστήσει το μεταγλωττισμένο πρόγραμμά του στο Arduino. Το μόνο που απαιτείται για την εγκατάσταση αυτή είναι ένα καλώδιο USB από type A σε type B. Επίσης, από άποψης λογισμικού, χρειάζεται η εγκατάσταση του οδηγού του FTDI chip (ο ελεγκτής του Serial-over-USB).

Η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring, μια παραλλαγή των C και C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega328 και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++. Ως μεταγλωττιστής χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc. Λόγω της καταγωγής της από την C, στην γλώσσα του Arduino γίνεται ουσιαστικά χρήση των ίδιων βασικών εντολών και συναρτήσεων με την ίδια σύνταξη, τους ίδιους τύπων δεδομένων και τους ίδιους τελεστές όπως και στην C. Πέρα από αυτές όμως, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για την διαχείριση του ειδικού hardware του Arduino.

3.4.1 Σύνδεση Arduino με Raspberry Pi

Όπως συμβαίνει και με κάθε άλλη σύνδεση, το Arduino συνδέεται με τον Raspberry Pi χρησιμοποιώντας τα GPIO pins ώστε να γίνει η τοποθέτηση της ανάλογης ασπίδας. Ωστόσο, στη δική μας εργασία θα ακολουθηθεί συγκεκριμένη

μεθοδολογία όπου δε θα χρησιμοποιηθεί καθεαυτός ο μικροελεγκτής Arduino αλλά θα χρησιμοποιηθεί ο Raspberry Pi με τέτοιο τρόπο ώστε να εκπληρώνει το σκοπό του Arduino. Παρόλα αυτά, επειδή δεν είναι εφικτό ο Raspberry Pi να υποκαταστήσει πλήρως το Arduino, θα χρησιμοποιηθεί η ασπίδα σύνδεσης Arduino σε Raspberry Pi έτσι ώστε να μπορούμε από εκεί και πέρα να συνδέσουμε οποιαδήποτε άλλη ασπίδα χρειαζόμαστε και να τις ελέγχουμε όλες μέσω της γλώσσας προγραμματισμού του Arduino.

Στην παρούσα εργασία χρησιμοποιήθηκε η ασπίδα σύνδεσης Arduino σε Raspberry Pi υπολογιστή που κατασκευάζεται και διατίθεται στο εμπόριο από την εταιρεία Cooking Hacks με έδρα την Ισπανία. Για την τοποθέτηση της ασπίδας αυτής απαιτείται η εγκατάσταση του λογισμικού arduinoPi. Έτσι, ο εκάστοτε χρήστης δύναται να επικοινωνήσει με τις ηλεκτρονικές του συσκευές, το λειτουργικό σύστημα του Raspberry Pi καθώς και με διαδικτυακές εφαρμογές.

Η ασπίδα σύνδεσης Arduino σε Raspberry Pi είναι μια πλακέτα σε μέγεθος πιστωτικής κάρτας η οποία ουσιαστικά μιμείται τον μικροελεγκτή του Arduino και περιέχει όλα τα pins και τις θύρες σύνδεσης που περιέχει μια τυπική πλατφόρμα αυτού του τύπου. Η ασπίδα υποστηρίζει δύο υποδοχές Xbee για τη σύνδεση ασύρματων μονάδων. Βέβαια, όπως έχει προαναφερθεί, ο Raspberry Pi διαθέτει θύρα Ethernet οπότε στις δικές μας εφαρμογές οικιακού αυτοματισμού δε θα χρειαστεί να χρησιμοποιηθούν οι Xbee συνδέσεις. Ωστόσο, είναι πάντα διαθέσιμες σε περίπτωση που για τον οποιονδήποτε λόγο δεν είναι επιθυμητή ή εφικτή η σύνδεση μέσω Ethernet. Η ασπίδα διαθέτει ακόμα έναν διακόπτη που ελέγχει την παροχή ρεύματος σε αυτή καθώς και μια σειριακή πύλη UART (Universal Asynchronous Receiver Transmitter) για τη γέφυρα της ασπίδας με τις ενδείξεις Rx και Tx. Η πύλη αυτή είναι πολύ σημαντική γιατί χρησιμεύει στη μετάδοση σειριακών δεδομένων όπως ένα τμήμα κειμένου, για παράδειγμα κατά τη γραφή ενός προγράμματος. Ακόμα, υπάρχουν GPIO pins για τη σύνδεση έταιρων ηλεκτρονικών στοιχείων όπως ένας αισθητήρας ή ένας διακόπτης. Εκτός, όμως, από τα GPIO υπάρχουν και SPI (Serial Peripheral Interface) pins για τη σύνδεση περιφερειακών συσκευών στην ασπίδα αυτή. Υπάρχει μια σύνδεση ICSP (In Circuit Serial Programmer) η οποία επιτρέπει τον προγραμματισμό του μικροελεγκτή του Arduino και power pins με τα οποία πρέπει να συνδέσουμε οποιαδήποτε συσκευή χρειάζεται να τροφοδοτηθεί ενεργειακά από την ασπίδα. Εκτός, όμως, από τις σειριακές θύρες υπάρχουν και αναλογικές εισοδοί για τη σύνδεση συσκευών που στέλνουν αναλογικό σήμα προς την ασπίδα. Τέλος, υπάρχει θύρα για τη σύνδεσή της με έναν Raspberry Pi. Η θύρα αυτή βρίσκεται στο κάτω μέρος της ασπίδας ενώ η σύνδεση γίνεται με τα GPIO pins του Raspberry Pi.

3.4.2 Δημιουργία λογισμικού για το Arduino

Το σύστημα Raspberry Pi-Ασπίδας Arduino δε μπορεί να φανεί χρήσιμο σε καμία περίπτωση παρά μόνο αν υπάρχει το κατάλληλο λογισμικό για το χειρισμό αυτού και των δομικών στοιχείων του, όπως αισθητήρες, διακόπτες κτλ.

Συνιθέστερα, η κατασκευή λογισμικού γίνεται μέσω της γλώσσας

προγραμματισμού του Arduino η οποία είναι μια παραλλαγή της C++. Η χρήση της συγκεκριμένης γλώσσας πλεονεκτεί στο ότι υπάρχει σημαντικός αριθμός ελεύθερων προγραμμάτων και βιβλιοθηκών στο διαδίκτυο τα οποία μπορούν να χρησιμοποιηθούν ως δομικά στοιχεία νεότερων προγραμμάτων.

Επίσης, μια συχνή επιλογή κατά τον προγραμματισμό του Arduino είναι η Python. Η Python είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου που επιτρέπει την κατασκευή εφαρμογών διαδικτύου και βάσεων δεδομένων που μπορούν να χρησιμοποιηθούν στη διαχείριση των εξόδων των προγραμμάτων του Arduino.

Στη συγκεκριμένη εργασία θα χρησιμοποιήσουμε τόσο τη γλώσσα προγραμματισμού του Arduino όσο και την Python ενώ για τη κατασκευή και διαχείριση βάσεων δεδομένων θα χρησιμοποιηθούν και οι SQL και HTSQL.

ΚΕΦΑΛΑΙΟ 4

ΟΙΚΙΑΚΟΣ ΑΥΤΟΜΑΤΙΣΜΟΣ

4.1 Εισαγωγή

Ο οικιακός αυτοματισμός είναι η επέκταση της τεχνολογίας του αυτοματισμού κτιρίων σε κατοικίες και περιλαμβάνει τον αυτοματισμό της οικίας καθευατής, των εργασιών που επιτελούνται εντός αυτής καθώς και των ηλεκτρικών συσκευών που περιλαμβάνονται εντός της. Ο οικιακός αυτοματισμός περιλαμβάνει εφαρμογές όπως ο κεντρικός έλεγχος του φωτισμού, η εφαρμογή HVAC (Heating, Ventilation, Air Conditioning) που ελέγχει ουσιαστικά τις κλιματικές συνθήκες εντός της οικίας μέσω ρύθμισης της θέρμανσης, της ψύξης και του εξαερισμού, το κλείδωμα των θυρών και των πυλών καθώς και πολυάριθμες άλλες οι οποίες παρέχουν ασφάλεια, άνεση, εξοικονόμηση ενεργειακών πόρων καθώς και κατ'οίκον διασκέδαση. Επίσης, στην περίπτωση ηλικιωμένων ατόμων καθώς και ατόμων με αναπηρίες, ο οικιακός αυτοματισμός παρέχει υψηλού επιπέδου ποιότητα ζωής καθιστώντας τα άτομα όσο το δυνατόν πιο αυτόνομα, χωρίς να έχουν πάντα την ανάγκη επαγγελματικής φροντίδας κατ'οίκον.

Η δημοτικότητα του οικιακού αυτοματισμού έχει αυξηθεί τα τελευταία χρόνια εξαιτίας των προσιτών –πλέον- τιμών των ανάλογων εγκαταστάσεων αλλά και της μεγάλης συνδεσιμότητας των απλοποιημένων αυτών εφαρμογών με smartphones και tablets. Βέβαια, αυτή η αύξηση της δημοτικότητάς του είναι άμεση συνέπεια της ανάπτυξης του τεχνολογικού τομέα του «Ίντερνετ των Πραγμάτων» (Internet of Things – IoT).

Ουσιαστικά, ο οικιακός αυτοματισμός συνδέει τις οικιακές ηλεκτρικές συσκευές μεταξύ τους χρησιμοποιώντας τεχνικές που χρησιμοποιήθηκαν αρχικά στον αυτοματισμό βιομηχανικών και υπηρεσιακών κτιρίων ενώ περιλαμβάνει και συστήματα ελέγχου με μετατροπές για τις αναλογίες των οικιακών εφαρμογών όπως για παράδειγμα στην περίπτωση των οικιακών συστημάτων διαχείρισης πολυμέσων, στην περίπτωση των κηπευτικών εργασιών, την τροφοδοσία των κατοικιδίων κτλ. Οι συσκευές αυτές δύνανται να συνδέονται μέσω ενός δικτύου υπολογιστών ενώ ο έλεγχός τους να γίνεται μέσω του προσωπικού ηλεκτρονικού υπολογιστή του χρήστη ενώ είναι δυνατός ο έλεγχός τους εξ'αποστάσεως μέσω του διαδικτύου.

Ιστορικά, ο οικιακός αυτοματισμός έκανε την εμφάνισή του στις αρχές του 20^{ου} αιώνα ως επακόλουθο της ραγδαίας εισαγωγής του ηλεκτρισμού στο σπίτι και της ανάπτυξης της τεχνολογίας των υπολογιστών. Από τα τέλη του 1800 είχαν ήδη κάνει την εμφάνισή τους οι πρώτες συσκευές τηλεχειρισμού ενώ οι πρώτες ηλεκτρικές οικιακές συσκευές κυκλοφόρησαν στο εμπόριο μεταξύ του 1915 και του 1920. Ωστόσο, εκείνη την εποχή, η ηλεκτροδότηση των σπιτιών ήταν ακόμα στα πρώτα στάδια της ανάπτυξής της με αποτέλεσμα η χρήση ηλεκτρικών οικιακών συσκευών να αποτελεί πολυτέλεια των λίγων και πιο εύπορων.

Εφαρμογές παρόμοιες με αυτές του σύγχρονου οικιακού αυτοματισμού άρχισαν να κανουν την εμφάνισή τους στα τέλη της δεκαετίας του 1930 ενώ μέχρι τη δεκαετία του 1960 ερασιτέχνες αυτοματιστές κατασκεύαζαν τα πρώτα αυτοματοποιημένα σπιτια με ενσύρματο δίκτυο. Το πρώτο σύστημα οικιακού αυτοματισμού ονομάστηκε ECHO IV αλλά δεν κυκλοφόρησε ποτέ στο εμπόριο. Ο όρος «έξυπνο σπίτι» χρησιμοποιήθηκε πρώτη φορά από την Αμερικανική Ένωση Οικοδόμων το 1984. Ωστόσο, ανακάλυψη-σταθμός για την πορεία του οικιακού αυτοματισμού αποτέλεσε αυτή των μικροχειριστηρίων αφού επέφερε σημαντική μείωση στο κόστος των συστημάτων αυτοματισμού. Οι κατασκευάστριες εταιρίες ενσωμάτωσαν στα συστήματά τους τηλεχειριστήρια καθώς και συσκευές έξυπνου χειρισμού ενώ ταυτόχρονα οι κατασκευαστές ηλεκτρονικών συσκευών ενσωμάτωσαν την τεχνολογία αυτή στις συσκευές τους. Έτσι, οι ηλεκτρικές συσκευές έγιναν περισσότερο προσβάσιμες ενώ ο χρήστης είχε τη δυνατότητα πλήρους ελέγχου αυτών. Μέχρι το 1990, ο οικιακός αυτοματισμός είχε εγκατασταθεί στη συνείδηση των κατασκευαστριών εταιρειών και έτσι δημιουργήθηκε ο κλάδος της δομοτικής, μια τεχνολογική έννοια που ενσωματώνει στην οικοδόμηση κτιρίων τις έννοιες της πληροφορικής και της τηλεματικής.

Ωστόσο, μέχρι το 1990 ο οικιακός αυτοματισμός ήταν ενασχόληση των εύπορων και των ερασιτεχνών αυτοματιστών καθώς τα συστήματα αυτά θεωρούνταν τόσο αρκετά ακριβά όσο και δύσχρηστα λόγω έλλειψης ενός συγκεκριμένου εννιαίου πρωτόκολλου λειτουργίας τους. Έτσι, υπήρχε πρόσφορο έδαφος για σημαντική αύξηση των εγκαταστάσεων οικιακού αυτοματισμού, οι οποίες στις Ηνωμένες Πολιτείες Αμερικής το 2012 έφτασαν τα 1,5 εκατομμύρια συστημάτων ενώ μέχρι το 2017 αναμένεται ο αριθμός αυτός να αγγίξει τα 8 εκατομμύρια.

4.2 Ωφέλη οικιακού αυτοματισμού

Τα συστήματα οικιακού αυτοματισμού έχουν ένα πολύ μεγάλο εύρος εφαρμογών. Μπορούμε να συναντήσουμε συστήματα που ξεκινούν από μια πολύ απλή λειτουργία όπως ο χειρισμός του φωτισμού μέσω τηλεχειριστηρίου μέχρι και πολύπλοκα δίκτυα που βασίζονται σε μικροϋπολογιστές με διάφορα επίπεδα ευφυίας και αυτοματισμού. Αυτός είναι και ο κυριότερος λόγος που προτιμάται η τεχνολογία του οικιακού αυτοματισμού, προσφέρει ένα εύχρηστο σύστημα διαχείρισης ενός μεγάλου εύρους εργασιών με ασφαλή τρόπο που οδηγεί -εκτός των άλλων- σε σημαντική εξοικονόμηση ενέργειας.

Στον ανεπτυγμένο και αναπτυσσόμενο κόσμο, όλα τα σπίτια διαθέτουν παροχή ηλεκτρικού ρεύματος, τηλεφωνικό δίκτυο και παροχή υπηρεσιών τηλεόρασης (δορυφορική, ψηφιακή ή καλωδιακή). Ο οικιακός αυτοματισμός βασίστηκε πάνω στην ύπαρξη αυτών των στοιχείων αλλά αναπτύχθηκε μέσω της εφεύρεσης εξειδικευμένων συσκευών, όπως για παράδειγμα τα αυτόματα πλυντήρια ρούχων και οι ηλεκτρικοί θερμοσίφωνες που εκμεταλλεύτηκαν την ηλεκτρική ενέργεια για να μειώσουν τη χειρωνακτική εργασία του πλυσίματος των ρούχων και της θέρμανσης του νερού που προοριζόταν για το μπάνιο του κατοίκου αντίστοιχα. Επίσης, διευκολύνθηκαν κι άλλες σημαντικές οικιακές

δραστηριότητες όπως η διατήριση των ευπαθών τροφίμων με τη βοήθεια του ψυγείου, η προετοιμασία των γευμάτων με τη βοήθεια των ηλεκτρικών εστιών, του ηλεκτρικού φούρνου, του φούρνου μικροκυμάτων και άλλων παρόμοιων συσκευών. Σε πολλές από αυτές τις διεργασίες, για παράδειγμα στη διατήριση των ευπαθών τροφίμων, βοήθησε και η βιομηχανική εξέλιξη μέσω εξελιγμένων συσκευασιών κτλ. Ο λόγος που θεωρείται σημαντική αυτή η συνεισφορά είναι πως σε συστήματα οικιακού αυτοματισμού κάποιες εφαρμογές θεωρούνται ιδιαίτερα ακριβές για να επιτευχθούν εξολοκλήρου κατ'οίκον οπότε η μαζική προετοιμασία τους σε βιομηχανικές εγκαταστάσεις είναι απαραίτητη.

Εξίσου σημαντική υπήρξε και η χρησιμοποίηση καυσίμων σε υγρή και αέρια μορφή ως πηγές ενέργειας εναλλακτικές της ηλεκτρικής. Έτσι, έγινε εφικτό ένα υψηλό επίπεδο αυτοματισμού των οικίων από ενεργειακής σκοπιάς ενώ η ανάπτυξη των θερμοστατών οδήγησε στον έλεγχο αυτών των ήδη αυτοματοποιημένων διεργασιών, όπως για παράδειγμα ο αυτόματος έλεγχος της ψύξης και της θέρμανσης.

Οι παραπάνω εξελίξεις αφορούσαν τα πλεονήκτα των ηλεκτρικών συσκευών καθ'αυτών. Ωστόσο, ο οικιακός αυτοματισμός δε θα μπορούσε να επιτευχθεί χωρίς την ύπαρξη δικτύων επικοινωνίας και διασύνδεσης των παραπάνω συσκευών. Τα δίκτυα αυτά είναι απαραίτητα για την επικοινωνία των συσκευών μεταξύ τους (βλ. Κεφάλαιο 1 – Επικοινωνίες Machine-To-Machine), την αποστολή δεδομένων, την ενεργοποίηση συναγερμών (οπτικών και ηχητικών ειδοποιήσεων) και την επικοινωνία με τον χρήστη του συστήματος όταν αυτός δε βρίσκεται εντός της οικίας. Βασική τεχνολογία σε αυτή την επικοινωνία είναι και οι αισθητήρες οι οποίοι λαμβάνουν τα δεδομένα από το περιβάλλον και ουσιαστικά παρέχουν τα δεδομένα εισόδου ώστε αν λειτουργήσει το σύστημα στο σύνολό του.



Εικόνα 1: Οι πιο διαδεδομένες εφαρμογές οικιακού αυτοματισμού.

Όπως έχει προαναφερθεί, οι εφαρμογές του οικιακού αυτοματισμού ποικίλλουν.

Στην εικόνα 1 μπορούμε να δούμε τις πιο διαδεδομένες εφαρμογές, οι οποίες πολλές φορές εγκαθίστανται σε οικίες και ανεξάρτητες χωρίς την ύπαρξη ενός συστήματος συνολικής διαχείρισης των οικιακών εργασιών. Σε απλές εγκαταστάσεις, συνεπώς, οι εφαρμογές μπορεί να είναι πολύ απλές, όπως η ενεργοποίηση του φωτισμού όταν κάποιο άτομο εισέρχεται εντός συγκεκριμένου δωματίου. Σε πιο σύνθετες εγκαταστάσεις, το σύστημα δεν αντιλαμβάνεται μόνο την είσοδο κάποιου ατόμου αλλά και ποιό άτομο είναι αυτό και έτσι έχει τη δυνατότητα -αν έχει προγραμματιστεί κατάλληλα- να ενεργοποιήσει το φωτισμό, τον κλιματισμό, τα επίπεδα έντασης του ήχου κατά την αναπαραγωγή πολυμέσων καθώς και τα προτιμώμενα κανάλια στην τηλεόραση για το άτομο αυτό συγκεκριμένα. Άλλες παράμετροι στην ενεργοποίηση του σωστού προφίλ λειτουργίας είναι για παράδειγμα η ημέρα, η ώρα της ημέρας, η εποχή του χρόνου και άλλες.

Από όλα τα παραπάνω μπορούν πολύ εύκολα να προκύψουν τα οφέλη που συνοδεύουν την εγκατάσταση ενός συστήματος αυτοματισμού του σπιτιού. Ανάλογα, βέβαια, με τα σημεία των καιρών κάποια από αυτά τα πλεονεκτήματα γίνονται πιο σημαντικά από άλλα. Για παράδειγμα, ζούμε σε μια εποχή όπου το ενεργειακό ζήτημα απασχολεί όλο και περισσότερο μέρος του καταναλωτικού κοινού και κατά συνέπεια η εξοικονόμηση ενεργειακών πόρων είναι πιο σημαντική γι'αυτούς. Με ανάλογο τρόπο σκέψης, για έναν ιδιοκτήτη μιας εξοχικής οικίας ιδιαίτερα απομακρυσμένης είναι σημαντικότερο, ίσως, το γεγονός πως οι εφαρμογές οικιακού αυτοματισμού προσφέρουν υψηλά επίπεδα ασφάλειας και προστασίας τόσο της οικίας όσο και των κατοίκων. Εποπτικά, με την εγκατάσταση ενός συστήματος οικιακού αυτοματισμού μπορούμε να επωφεληθούμε με τους εξής τρόπους:

1. Αυξάνει τα επίπεδα ασφάλειας της οικίας.

Πολλές από τις εφαρμογές οικιακού αυτοματισμού στοχεύουν στη δημιουργία ενός περιβάλλοντος που θα παρέχει ασφάλεια στον κάτοικο τόσο αναφορικά με τον εαυτό του και τους οικίους του όσο και με το οίκημα. Κυριότερη εφαρμογή είναι αυτή του αυτοματισμού του κλειδώματος των θυρών και των κεντρικών πυλών του σπιτιού. Μπορούμε, έτσι, να έχουμε πλήρη έλεγχο του κλειδώματος του σπιτιού μας και να μπορούμε κάθε στιγμή να ελέγξουμε αν οι θύρες παραμένουν κλειδωμένες κατά την απουσία μας. Ταυτόχρονα, είναι δυνατή η ενεργοποίηση συστήματος ειδοποίησης το οποίο να ενημερώνει το χρήστη πως οι θύρες παραμένουν ξεκλειδωτες σε χρονικά διαστήματα που δεν έχει διαπιστωθεί ανθρώπινη δραστηριότητα στο χώρο. Επίσης, μέσω ενός απλού συστήματος ελέγχου μπορεί να διαπιστωθεί αν έχουν απενεργοποιηθεί όλες οι ηλεκτρικές συσκευές για την αποφυγή πυρκαγιάς ή ακόμη και αν είναι κλειστές όλες οι βρύσες για την αποφυγή πλυμμήρας. Επίσης, μπορούμε να ελέγξουμε τις ηλεκτρικές συσκευές και το φωτισμό όταν δε βρισκόμαστε στο σπίτι ώστε να δημιουργείται η αίσθηση σε όσους βρίσκονται εκτός αυτού ότι κάποιος βρίσκεται στο σπίτι. Αυτές οι εφαρμογές ουσιαστικά προσομοιώνουν την ανθρώπινη δραστηριότητα και αποτελούν σημαντικό εργαλείο για μακρόχρονη απουσία των κατοίκων από το σπίτι, για παράδειγμα κατά την περίοδο των θερινών διακοπών. Τέλος, μέσω συστημάτων παρακολούθησης (κάμερες) μπορούμε πάντα να ελέγξουμε ποιός βρίσκεται τόσο εντός της

οικίας όσο και στον περιβάλλοντα χώρο. Επίσης, μπορούμε να επιβλέπουμε τα μικρά παιδιά όταν είναι στο δωμάτιό τους ή τους ηλικιωμένους και τα άτομα με κινητικές δυσκολίες.

2. Διευκολύνει τη διεκπεραίωση των οικιακών εργασιών

Είναι γεγονός ότι τα σύγχρονα σπίτια αποτελούν κάτι παραπάνω από ένα χώρο για να κατοικεί μια οικογένεια. Μαζί, όμως, με όλες τις δυνατότητες που παρέχουν στους κατοίκους τους έρχονται και πολυάριθμες διεργασίες που πρέπει να γίνουν σε αυτές. Τα συστήματα αυτοματισμού παρέχουν εφαρμογές μέσω των οποίων γίνεται ο προγραμματισμός των οικιακών εργασιών. Έτσι, το σύστημα από μόνο του μπορεί να διευθετήσει καθημερινές εργασίες όπως για παράδειγμα να ενεργοποιήσει τα πλυντήρια, να κρατάει το πρόγραμμα των διαφόρων επισκευών που πρέπει να γίνουν, να κρατάει αρχείο αναφορικά με τις διεργασίες αυτές κτλ. Μπορεί, επίσης, να κρατάει αρχείο με τα προϊόντα που προμηθεύονται οι κάτοικοι ανάλογα με το barcode τους και να προετοιμάζει τη λίστα για τα ψώνια.

3. Εξοικονομεί ενέργεια και κατά συνέπεια και χρήματα

Μια από τις πολύ σημαντικές ιδιότητες των συστημάτων αυτοματισμού είναι η εξοικονόμηση ενεργειακών πόρων. Είναι κάτι παραπάνω από ευνόητο πως όταν είναι δυνατός ο πλήρης έλεγχος των ηλεκτρικών συσκευών, του φωτισμού, της θέρμανσης/ψύξης κτλ, τότε γίνεται μόνο η απόλυτα απαραίτητη κατανάλωση ενέργειας. Επίσης, σε περιπτώσεις όπου διατίθεται οικονομικό τιμολόγιο από την ηλεκτροδότητρια εταιρεία για συγκεκριμένες ώρες της ημέρας (ένα παράδειγμα είναι το νυχτερινό τιμολόγιο της ΔΕΗ) το σύστημα ρυθμίζεται με τρόπο τέτοιο ώστε να εκμεταλλεύεται αυτή τη δυνατότητα. Έτσι, δε γίνεται ενεργειακή σπατάλη, γεγονός το οποίο έχει σημαντικό αντίκτυπο στην τσέπη του καταναλωτή αφού αυτή η οικονομία στην ενέργεια άμεσα μεταφράζεται σε οικονομία χρηματική. Η εξοικονόμηση χρημάτων, όμως, δεν αφορά μόνο την κατανάλωση ενέργειας αλλά και τις συσκευές τις ίδιες αφού μέσω του συστήματος είναι δυνατή η ενεργοποίηση και απενεργοποίηση αυτών και κατ'επέκταση η αποφυγή βλαβών και ανάγκης αντικατάστασης λόγω αλόγιστης χρήσης των συσκευών.

4. Εξοικονομεί χρόνο

Μέσω του προγραμματισμού των διαφόρων εργασιών καθώς και την υπενθύμιση αυτών, τα συστήματα αυτοματισμού δεν αναλαμβάνουν απλά την εκτέλεση εργασιών αντί των κατοίκων εξοικονομώντας τον κόπο που αυτοί θα κατέβαλαν αλλά εξοικονομώντας και το χρόνο που θα απαιτούσαν για τον προγραμματισμό και την εκτέλεση αυτών. Επίσης, τα άτομα δε χρειάζεται να ανησυχούν για να μην ξεχάσουν τις υποχρεώσεις τους, γεγονός το οποίο θα επέφερε κάποιο άγχος στην καθημερινότητά τους, αφού το σύστημα μπορεί και διατηρεί ένα καλά καθορισμένο πρόγραμμα γι'αυτούς.

5. Προσφέρει υψηλού επιπέδου συνθήκες διαβίωσης

Οι εφαρμογές των συστημάτων αυτοματισμού δεν στοχεύουν μόνο στην εξοικονόμηση ενεργειακών πόρων και τη διαχείριση των καθημερινών «υποχρεώσεων» αλλά και στην παροχή ενός περιβάλλοντος που να ανταποκρίνεται στις υψηλές απαιτήσεις ενός συνόλου αρκετών -πολλές φορές- ατόμων. Ένα παράδειγμα είναι η ρύθμιση της θερμοκρασίας εντός

του σπιτιού. Η επιθυμητή θερμοκρασία δεν είναι πάντα η ίδια για όλα τα μέλη μια οικογένειας. Έτσι, μέσω συγκεκριμένων εφαρμογών ορίζεται διαφορετική θερμοκρασία για κάθε δωμάτιο του σπιτιού και ανάλογα με τα ωράρια των μελών της οικογένειας. Αυτή η ρύθμιση μπορεί να γίνεται και εξ'αποστάσεως έτσι ώστε, για παράδειγμα, να μπορεί ο χρήστης να θερμάνει την οικία του μέχρι να φτάσει σε αυτή.

6. Προσφέρει ένα υψηλό επίπεδο κατ'οίκον διασκέδασης

Μέσω συγκεκριμένων εφαρμογών, τα συστήματα αυτοματισμού μπορούν και διαχειρίζονται όλες τις συσκευές αναπαραγωγής πολυμέσων με τρόπο που να ικανοποιεί κάθε φορά τον εκάστοτε χρήστη. Έτσι, είναι δυνατή η αναπαραγωγή ήχου και εικόνας ανάλογα με το ποιός είναι ο δέκτης τους, την ώρα της ημέρας, τις προτιμήσεις του κτλ. Επίσης, είναι δυνατή η επικοινωνία των μέσων αναπαραγωγής με το διαδίκτυο για live streaming εφαρμογές κτλ.

4.3 Στοιχεία Συστήματος Οικιακού Αυτοματισμού

Ένα σύστημα οικιακού αυτοματισμού αποτελείται από ένα πλήθος δομικών στοιχείων ανάμεσα στα οποία υπάρχουν αισθητήρες, χειριστήρια, διακόπτες, συσκευές και άλλα. Επίσης, πρέπει να περιλαμβάνεται τουλάχιστον μια συσκευή για την επικοινωνία του χρήστη με το σύστημα, η οποία έχει τη μορφή οθόνης συνήθως. Τα τελευταία χρόνια είναι πολύ συνηθισμένο κάποιο smartphone να παίζει το ρόλο της συσκευής αυτής. Οι συσκευές μπορούν να επικοινωνούν μεταξύ τους μέσω ενσύρματου ή ασύρματου δικτύου χρησιμοποιώντας γνωστά πρωτόκολλα οικιακού αυτοματισμού. Στα στοιχεία αυτά αναφερθήκαμε και στα προηγούμενα κεφάλαια οπότε κάποια επανάληψη των πληροφοριών δε θεωρείται σκόπιμη. Ωστόσο, πρέπει να αναφερθούμε στο πρωτόκολλο X10, το πιο διαδεδομένο πρωτόκολλο επικοινωνίας συσκευών σε εφαρμογές οικιακού αυτοματισμού.

4.3.1 Το Πρωτόκολλο X10

Το πρωτόκολλο X10 είναι ένα τυποποιημένο πρωτόκολλο οικιακού αυτοματισμού το οποίο χρησιμοποιεί το υφιστάμενο δίκτυο ηλεκτρικής ενέργειας σε ένα σπίτι ή γραφείο για τη διαβίβαση των εντολών. Το χαμηλό κόστος, η ευκολία στη χρήση και την ποικιλία του εξοπλισμού έχουν κάνει το X10 το πιο γνωστό πρωτόκολλο οικιακού αυτοματισμού στον κόσμο. Η τεχνολογία X10 αναπτύχθηκε γύρω στο 1976-1978 από την Pico Electronics Ltd στη Σκωτία. Ο κύριος σκοπός του ήταν η μετάδοση δεδομένων μέσω των γραμμών του ηλεκτρικού ρεύματος σε χαμηλή ταχύτητα (50 bps στην Ευρώπη και 60 bps στις Ηνωμένες Πολιτείες). Το όνομά του το πήρε από το γεγονός ότι αυτό ήταν το δέκατο project της εταιρείας. Στη συνέχεια το δίπλωμα ευρεσιτεχνίας εξαγοράστηκε από την εταιρία X10 Ltd και διατηρήθηκε μέχρι το 1997 που έληξε. Το X10 είναι σήμερα ένα ανοιχτό πρωτόκολλο και υπάρχουν πάρα πολλές εταιρίες που αναπτύσσουν προϊόντα βασισμένα σε αυτή την τεχνολογία.

Οι βασικοί λόγοι για την μεγάλη επιτυχία του X10 στην ΑΜερική και την Ευρώπη είναι η τεράστια ποικιλία συσκευών και διασυνδέσεων X10, η χρήση του υφιστάμενου δικτύου ηλεκτρικού ρεύματος ως μέσο μετάδοσης, οι χαμηλές τιμές των συσκευών X10 και η ευκολία εγκατάστασης (οι περισσότερες συσκευές είναι τύπου Plug'n Play). Όπως αναφέρθηκε, η τεχνολογία X10 χρησιμοποιεί το δίκτυο διανομής της ηλεκτρικής ενέργειας ως το κύριο μέσο επικοινωνίας μεταξύ των διαφόρων συσκευών. Αυτό αποτελεί βασική πτυχή αυτής της τεχνολογίας και το σημαντικό πλεονέκτημά του σε σχέση με άλλα πρωτόκολλα οικιακού αυτοματισμού. Η τεχνολογία αυτή χρησιμοποιεί αποκεντρωμένη αρχιτεκτονική, που δεν απαιτεί κεντρικό ελεγκτή για να λειτουργήσει. Ένα X10 σύστημα μπορεί απλά να αποτελείται από μια σειρά διατάξεων που ελέγχονται απ'ευθείας από το χρήστη. Για παράδειγμα, μέσω τηλεχειρισμού RF (ραδιοσυχνότητες), μπορεί κανείς να στείλει μια εντολή προς το δέκτη X10/RF, ο οποίος μεταδίδεται μέσω του δικτύου ηλεκτρικής ενέργειας σε ένα X-10 ενεργοποιητή, ο οποίος, με τη σειρά του, ανοίγει ή κλείνει τη συσκευή.



Εικόνα 2: Συσκευές X10.

Το πρωτόκολλο X10 βασίζεται σε μια αποκεντρωμένη αρχιτεκτονική, που σημαίνει ότι δεν υπάρχει κάποιος κεντρικός ελεγκτής. Η αρχιτεκτονική αυτή δίνει μεγάλη ευελιξία στο σύστημα και το καθιστά πολύ λιγότερο ευάλωτο σε βλάβες, καθώς σε περίπτωση βλάβης μιας συσκευής X10 τίθεται εκτός λειτουργίας μόνο η συγκεκριμένη συσκευή και όχι όλο το σύστημα X10. Η μετάδοση των σημάτων X10 για τον έλεγχο των συσκευών γίνεται μέσω του δικτύου ηλεκτρικής ενέργειας που υπάρχει στο σπίτι/γραφείο. Αυτός ο τρόπος μετάδοσης είναι πολύ οικονομικός καθώς δεν απαιτούνται επιπλέον καλωδιώσεις και καθιστά τα συστήματα X10 πολύ εύκολα στην εγκατάσταση. Η μετάδοση των σημάτων X10 γίνεται μέσω υψηλών συχνοτήτων που εισάγονται στο δίκτυο ηλεκτρικής ενέργειας και οι οποίες είναι τέτοιες ώστε να μην παρεμβαίνουν στις συνηθισμένες ηλεκτρικές συσκευές. Επίσης στα συστήματα X10 είναι διαδεδομένη η χρήση ραδιοσυχνοτήτων (RF) η οποία είναι ευέλικτη καθώς διαπερνά τοίχους και άλλα εμπόδια. Η χρήση ραδιοσυχνοτήτων

χρησιμοποιείται κυρίως σε τηλεχειριστήρια που απαιτούν μεγάλη κινητικότητα.

Δεδομένου ότι κάθε συσκευή X10 μπορεί να λαμβάνει όλα τα σήματα X10 που μεταδίδονται μέσω του δικτύου ηλεκτρικής ενέργειας, το σύστημα θα πρέπει να είναι σε θέση να διαχειριστεί τις εντολές X10. Για την επίλυση αυτού του προβλήματος, το πρωτόκολλο X10 εφαρμόζει ένα απλό σύστημα που χρησιμοποιεί 16 κωδικούς σπιτιών (από το γράμμα A έως το P) και 16 κωδικούς συσκευών (1 έως 16), επιτρέποντας έτσι 256 (16 x 16) μοναδικές διευθύνσεις. Ο ορισμός της διεύθυνσης μιας συσκευής γίνεται από τον χρήστη. Εάν η ίδια διεύθυνση αποδοθεί σε περισσότερες από μια συσκευές, τότε όλες οι συσκευές με την ίδια διεύθυνση θα ανταποκρίνονται στις εντολές X10 (αυτό πολλές φορές είναι επιθυμητό).

4.4 Κόστος Οικιακού Αυτοματισμού

Ένα ερώτημα που πολύ εύλογα εγείρει το θέμα του οικιακού αυτοματισμού είναι το κόστος που συνεπάγεται στον καταναλωτή η εγκατάστασή του.

Ένα σύστημα οικιακού αυτοματισμού μπορεί να έχει την πολύ απλή μορφή του συνόλου χειριστηρίων ελέγχου ή ακόμα και μια πολύ σύνθετη μορφή συσκευών που συνδέονται στο ηλεκτρικό ρεύμα ακι η λειτουργία τους ελέγχεται εξ'αποστάσεως. Τα κόστη των εγκαταστάσεων αυτών ποικίλλουν και εξαρτώνται κάθε φορά από τις συσκευές, την επίπλωση και τη γενικότερη υποδομή που απαιτείται αλλά και από το κόστος της εγκατάστασης στο χώρο της κατοικίας.

Στα τρέχοντα έξοδα μιας εγκατάστασης οικιακού αυτοματισμού συμπεριλαμβάνονται η κατανάλωση ηλεκτρικού ρεύματος για τη λειτουργία αυτού, κόστη συντήρησης καθώς και το κόστος αναβάθμισης όταν αυτή κριθεί επιθυμητή ή ακόμη και αναγκαία. Τέλος, αν το σύστημα είναι ιδιαίτερα περίπλοκο μπορεί να θεωρηθεί αναγκαία η παρακολούθηση επιμορφωτικών μαθημάτων που πιθανότατα να συνοδεύονται από κάποιο κόστος.

ΚΕΦΑΛΑΙΟ 5

ΚΑΤΑΣΚΕΥΗ ΕΝΟΣ ΣΥΣΤΗΜΑΤΟΣ

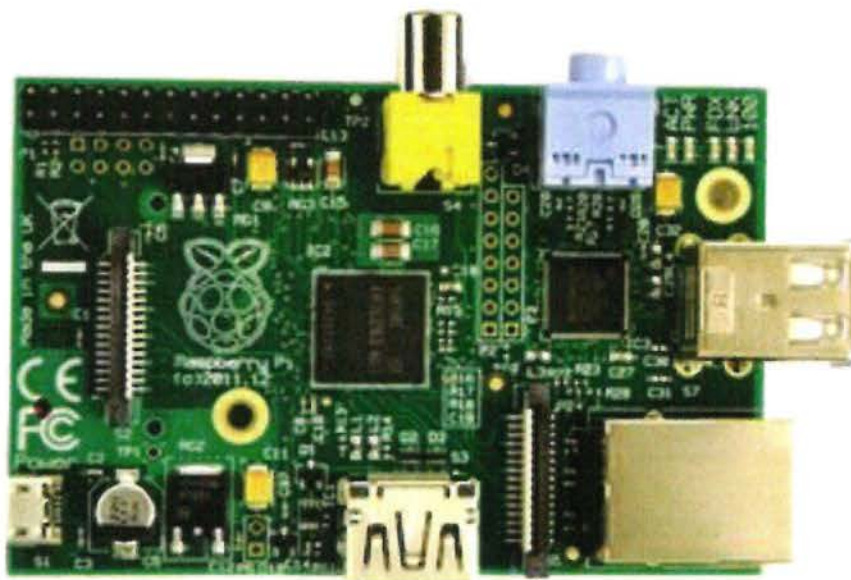
RASPBERRY PI - ARDUINO

5.1 Εισαγωγή

Στο Κεφάλαιο αυτό θα ασχοληθούμε με το κατασκευαστικό τμήμα της εργασίας μας το οποίο δεν είναι άλλο από τη δημιουργία ενός συστήματος Raspberry Pi-Arduino Shield το οποίο στη λογική της επικοινωνίας machine-to-machine θα μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών οικιακού αυτοματισμού. Αρχικά θα ασχοληθούμε με την προετοιμασία του υπολογιστή Raspberry Pi ενώ στη συνέχεια θα ενσωματώσουμε στο σύστημα την ασπίδα του Arduino.

5.2 Δημιουργία του υπολογιστή Raspberry Pi

Έχουμε προμηθευτεί ένα Raspberry Pi μικροϋπολογιστή Model B με επεξεργαστή στα 700 MHz και ενσωματωμένη μνήμη RAM 512 MB.



Εικόνα 1: Υπολογιστής Raspberry Pi Mode B.

Αυτός δεν είναι έτοιμος για χρήση αλλά εκκρεμούν οι εξής διεργασίες:

1. Προετοιμασία της SD κάρτας για εγκατάσταση του λειτουργικού συστήματος.
2. Επιλογή του κατάλληλου λειτουργικού συστήματος και εγκατάσταση αυτού.

Αυτές οι διεργασίες θα περιγραφούν στις επόμενες παραγράφους. Μετά το πέρας τους, ο υπολογιστής θα είναι έτοιμος να αποτελέσει τμήμα ενός συστήματος οικιακού αυτοματισμού.

5.2.1 Προετοιμασία SD κάρτας

Μια κάρτα SD (Secure Digital) είναι ένα μέσο αποθήκευσης υψηλών επιδόσεων διαθέσιμη για ένα μεγάλο εύρος ηλεκτρονικών συσκευών, από φωτογραφικές μηχανές και κινητά τηλέφωνα μέχρι και ηλεκτρονικούς υπολογιστές. Ο Raspberry Pi διαθέτει θύρα κάρτας SD, γεγονός το οποίο μας επιτρέπει να εισάγουμε μια SD κάρτα σε αυτόν και να τη χρησιμοποιήσουμε ως τον κύριο αποθηκευτικό μηχανισμό του υπολογιστή μας, δηλαδή να την καταστήσουμε το σκληρό του δίσκο. Βέβαια, υπάρχει η δυνατότητα να χρησιμοποιηθεί και οποιοδήποτε άλλο αποθηκευτικό μέσο συνδέεται μέσω των θυρών USB. Ωστόσο, εμείς προτιμήσαμε την SD κάρτα αρχικά λόγω του μικρού της μεγέθους, δευτερευόντως λόγω της δυνατότητας ενσωμάτωσης στο σύστημα (δε φέρεται ως μια επιπλέον συσκευή αλλά είναι ενσωματωμένη πλέον μέσα στον Raspberry Pi) και τέλος για την χαμηλή τιμή της. Ωστόσο, κατά την επιλογή της κάρτας SD δε μας απασχόλησε ιδιαίτερα ούτε το μέγεθός της ούτε η κατασκευάστρια εταιρεία αλλά το αν θα διαλέγαμε μια κάρτα με προεγκατεστημένο το λειτουργικό σύστημα ή μια τελείως κενή κάρτα την οποία θα έπρεπε να διαμορφώσουμε εμείς. Οι κάρτες με προεγκατεστημένο το λειτουργικό σύστημα υπάρχουν στο εμπόριο σχεδόν από την κυκλοφορία του Raspberry Pi και αποτελούν μια καλή –και ασφαλή– επιλογή για ερασιτέχνες οι οποίοι θέλουν να ξεκινήσουν την ενασχόλησή τους με τους υπολογιστές Raspberry Pi αλλά δεν επιθυμούν να ακολουθήσουν όλη τη διαδικασία κατασκευής του, είτε γιατί δεν έχουν τις απαραίτητες γνώσεις, είτε γιατί ενδιαφέρονται απλά για τις εφαρμογές που μπορούν να αναπτυχθούν και όχι για το σύστημα καθ'αυτό.



Εικόνα 2: Κάρτα SD με προεγκατεστημένο λειτουργικό σύστημα για χρήση σε υπολογιστή Raspberry Pi, του οποίου και το χαρακτηριστικό σύμβολο φέρει.

Ωστόσο, εμείς προτιμήσαμε να ξεκινήσουμε με μια άδεια κάρτα SD για δύο λόγους: Πρώτον, «στήνοντάς» το σύστημά μας από το μηδέν καταφέραμε να το γνωρίσουμε όσο το δυνατόν καλύτερα και δεύτερον, επειδή κατασκευάζοντας αυτό το σύστημα επιθυμούσαμε να καταφέρουμε να αποκτήσουμε γνώσεις για τη δημιουργία οποιουδήποτε άλλου. Έτσι, επιλέξαμε μια κοινή SD κάρτα του

εμπορίου μνήμης 8 GB. Για τον ίδιο, επίσης, λόγο επιλέξαμε η μορφοποίηση της κάρτας να γίνει μέσω της εφαρμογής BerryBoot, η οποία μας επιτρέπει να επιλέξουμε το κατάλληλο λειτουργικό σύστημα για την εκάστοτε εφαρμογή, την εγκατάσταση πολλαπλών λειτουργικών συστημάτων ή την απεγκατάσταση του ήδη υπάρχοντος και την εγκατάσταση ενός νέου. Με την εφαρμογή αυτή, όμως, θα ασχοληθούμε στη συνέχεια.

Αρχικά, θα πρέπει να στηθεί η SD κάρτα. Εκκίνηση αυτής της διεργασίας είναι η μορφοποίησή της στο κατάλληλο σύστημα αρχείων FAT. Το σύστημα αρχείων FAT (File Allocation Table) είναι μια μέθοδος καταγραφής των τομέων του δίσκου στους οποίους είναι αποθηκευμένα τα αρχεία και ποιοί τομείς είναι ελεύθεροι προς εγγραφή. Εξαιτίας των δυνατοτήτων του αλλά και της απλότητάς του, το σύστημα FAT είναι διαθέσιμο ακόμη και στις κάρτες SD και είναι αυτό που θα χρησιμοποιήσουμε για να τρέξουμε την εφαρμογή για την επιλογή του λειτουργικού μας συστήματος. Αφού λοιπόν διαμορφώσουμε την κάρτα, εγκαθιστούμε το BerryBoot το οποίο ουσιαστικά θα εγκαταστήσει το λειτουργικό μας σύστημα στην κάρτα.

Εισάγουμε, λοιπόν, την SD κάρτα στην ανάλογη θύρα στον Raspberry Pi.



Εικόνα 3: Η θύρα της SD κάρτας στον Raspberry Pi.

Με τη βοήθεια του προσωπικού μας ηλεκτρονικού υπολογιστή (λειτουργικό σύστημα Linux). Βέβαια, πολλές κατασκευάστριες εταιρείες δίνουν στο εμπόριο ήδη διαμορφωμένες κάρτες και δε χρειάζεται πάντα να τις μορφοποιήσουμε. Αυτή η πληροφορία αναγράφεται πάντα στη συσκευασία της κάρτας. Σε παράθυρο εντολών χρησιμοποιούμε το πρόγραμμα `mkdosfs` το οποίο μορφοποιεί το δίσκο ώστε να χρησιμοποιήσει ένα σύστημα FAT. Στο παράθυρο εντολών, λοιπόν, και αφού εκκινήσουμε εργαία ας `root`, με την εντολή `df -h` μπορούμε να βρούμε τόσο το όνομα του συστήματος αρχείων της SD κάρτας όσο και την τοποθεσία της. Αρχικά, με την εντολή `umount` αφαιρούμε την κάρτα χρησιμοποιώντας το όνομα που βρήκαμε προηγουμένως. Με την εντολή `mkdosfs /dev/(SD_card_name) -F32` μορφοποιούμε την κάρτα σε FAT32. Στη συνέχεια, επανατοποθετούμε την κάρτα με την εντολή `mount/dev/(SD_card_name)/media/SDcard` και η κάρτα είναι έτοιμη για την εγκατάσταση του BerryBoot.

5.2.2 Εγκατάσταση Λειτουργικού Συστήματος

Το BerryBoot αποτελεί τον ευκολότερο τρόπο με τον οποίο μπορεί κάποιος να εγκαταστήσει το λειτουργικό σύστημα που επιθυμεί στην κάρτα SD. Εκτός από την ευκολία της, πλεονεκτεί και στο γεγονός ότι είναι συμβατή τόσο με Windows όσο και με Mac και Linux. Η εφαρμογή είναι πολύ απλή στη χρήση της, απλά την ανοίγουμε σε μια μορφοποιημένη SD κάρτα και εκκινεί όταν ενεργοποιήσουμε τον υπολογιστή. Μόλις γίνει αυτό, ο χρήστης μπορεί να επιλέξει το λειτουργικό σύστημα που θέλει να εγκαταστήσει και η εφαρμογή τον καθοδηγεί για να το πράξει. Όπως προαναφέραμε, η εφαρμογή BerryBoot εξυπηρετεί και όταν επιθυμούμε να εγκαταστήσουμε περισσότερα του ενός λειτουργικά συστήματα σε μια μοναδική κάρτα SD.

Αρχικά, προμηθευόμαστε το BerryBoot σε συμπιεσμένη μορφή (21.3 MB) από τον ιστότοπο <http://www.berryterminal.com/doku.php/berryboot>. Στο σύστημά μας υπάρχει ήδη εγκατεστημένο το πρόγραμμα unzip για την αποσυμπίεση του αρχείου οπότε μπορούμε να εξάγουμε τα περιεχόμενα του συμπιεσμένου BerryBoot αρχείου στην κάρτα SD. Μεταξύ αυτών βρίσκονται και τα αρχεία που θα χρησιμοποιηθούν στην πρώτη λειτουργία του Raspberry Pi. Αφού, λοιπόν, αποσυμπιέσουμε το BerryBoot στην κάρτα SD, μπορούμε να συνδέσουμε τον Raspberry Pi με τα λοιπά περιφερειακά και να ξεκινήσουμε την εγκατάσταση του λειτουργικού συστήματος.

Ωστόσο, όπως προαναφέρθηκε, πριν ενεργοποιήσουμε τον Raspberry Pi θα πρέπει να συνδέσουμε τις περιφερειακές του συσκευές. Τοποθετούμε, αρχικά, την κάρτα SD στον Raspberry Pi και συνδέουμε την οθόνη σε αυτόν. Συνδέουμε το πληκτρολόγιο και το ποντίκι στις θύρες USB του ενώ χρησιμοποιώντας ένα καλώδιο Ethernet συνδέουμε το modem μας με τη θύρα Ethernet του υπολογιστή. Τέλος, συνδέουμε το τροφοδοτικό και ενεργοποιούμε τον Raspberry Pi. Στην οθόνη εμφανίζεται το «καλωσόρισμα» του BerryBoot και αυτή είναι η ένδειξη που περιμένουμε σε περίπτωση που το έχουμε εγκαταστήσει σωστά στην κάρτα SD οπότε όλα είναι έτοιμα για την επιλογή και την εγκατάσταση του λειτουργικού συστήματος.

Επόμενο βήμα είναι η επιλογή του κατάλληλου λειτουργικού συστήματος. Σε προηγούμενο Κεφάλαιο αναφερθήκαμε στο Raspbian το οποίο είναι το αυτό που θα χρησιμοποιήσουμε. Η επιλογή αυτού για τις εφαρμογές οικιακού αυτοματισμού γίνεται για πολλούς λόγους. Αρχικά, είναι πολύ σημαντικό το γεγονός ότι το Raspbian έχει βασιστεί πάνω στο λειτουργικό σύστημα Debian Wheezy Linux και έχει βελτιστοποιηθεί για χρήση σε υπολογιστή Raspberry Pi. Παρά το γεγονός, λοιπόν, ότι δεν αποτελεί επίσημο προϊόν του Ιδρύματος Raspberry Pi, είναι αυτό που το Ίδρυμα προτείνει στους αρχάριους χρήστες. Επίσης, το Raspbian λειτουργεί σε εικονικό περιβάλλον παρόμοιο με του Mac και των Windows, το LXDE, με αποτέλεσμα να διευκολύνει τους χρήστες που δεν είναι εξοικειωμένοι με τα παράθυρα εντολών. Ακόμα, σε αυτό έχει προεγκατασταθεί λογισμικό το οποίο θα φανεί χρήσιμο κατά τη δημιουργία των εφαρμογών μας, όπως το Python αλλά και λογισμικό εκπαιδευτικού προσανατολισμού. Τέλος, υπάρχει μεγάλη προγραμματιστική κοινότητα που ασχολείται με το Raspbian, γεγονός που προσφέρει πλήθος πηγών κώδικα και βιβλιοθήκες.

Αφού, λοιπόν, έχουμε «στήσει» τον Raspberry Pi και τον έχουμε ενεργοποιήσει, συναντούμε το παράθυρο που μας καλωσορίζει στο BerryBoot.



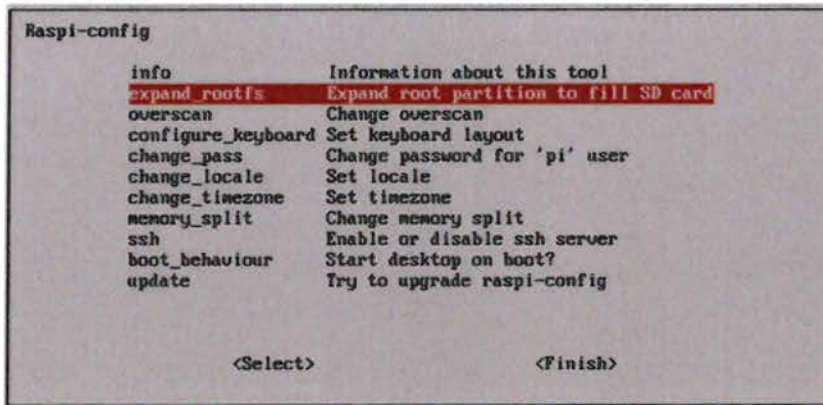
Εικόνα 4: Η οθόνη που μας καλωσορίζει στο BerryBoot.

Επιλέγουμε ενσύρματη σύνδεση στο διαδίκτυο και προσδιορίζουμε την ώρα (ο Raspberry Pi δε διαθέτει ενσωματωμένο ρολόι οπότε αυτή η διαδικασία πρέπει να γίνεται με κάθε εκκίνησή του) καθώς και τη μορφή του πληκτρολογίου. Στη συνέχεια κατευθυνόμαστε στην οθόνη επιλογής δίσκου όπου είναι το σημείο που πρέπει να επιλέξουμε την κάρτα SD ως το μέρος όπου θέλουμε να εγκατασταθεί το λειτουργικό μας σύστημα. Επιλέγουμε, λοιπόν, την κάρτα SD ενώ φροντίζουμε ώστε το κουτί επιλογής File System να δείχνει ext 4 ώστε να λειτουργήσουμε μέσω Linux. Επιλέγουμε το κουμπί Format και περιμένουμε μέχρι το τέλος της μορφοποίησης. Ακολουθεί η εμφάνιση της οθόνης Install Operating System όπου διαλέγουμε το Raspbian.



Εικόνα 5: Επιλέγοντας Debian Wheezy Raspbian στο BerryBoot.

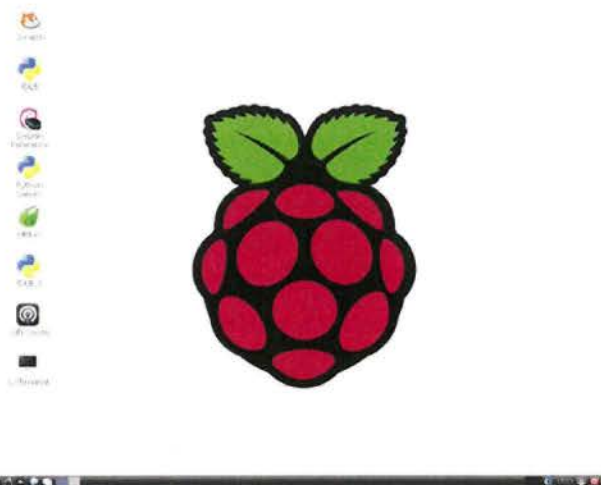
Ενεργοποιούμε την επιλογή Debian Wheezy Raspbian και αναμένουμε για μερικά λεπτά να ολοκληρωθεί η λήψη του αρχείου (430 MB). Στη συνέχεια εμφανίζεται η οθόνη BerryBoot Menu Editor που περιλαμβάνει λενά μενού με τα λειτουργικά συστήματα που αυτή τη στιγμή υπάρχουν εγκατεστημένα στην κάρτα SD, δηλαδή μόνο το Debian Wheezy Raspbian. Επιλέγουμε το λειτουργικό σύστημα αυτό και πατάμε το κουμπί Make Default στο πάνω μέρος της οθόνης έτσι ώστε το Raspbian να είναι το προεπιλεγμένο λειτουργικό σύστημα. Ακολούθως, οδηγούμαστε στην οθόνη διαμόρφωσης του Raspbian (Raspi-config) όπου υπάρχει κατάλληλο μενού για την εκχώρηση τιμών στις παραμέτρους που χρησιμοποιεί το Raspbian.



Εικόνα 6: Η οθόνη του Raspi-config.

Σε αυτό το σημείο θα αλλάξουμε το password του συστήματος από το προεπιλεγμένο που είναι «raspberrypi» σε αυτό που επιθυμούμε εμείς. Επίσης, θα ενεργοποιήσουμε τη δυνατότητα SSH, δηλαδή θα καταστήσουμε δυνατή τη σύνδεση του χρήστη στον Raspberry Pi εξ'αποστάσεως, μέσω κάποιου τρίτου ηλεκτρονικού υπολογιστή. Τερματίζουμε τη διαδικασία αυτή μέσω του πλήκτρου Finish. Ο Raspberry Pi υπολογιστής μας είναι έτοιμος για χρήση.

Αφού ολοκληρώθηκε η παραπάνω διαδικασία, βρισκόμαστε πλέον στην επιφάνεια εργασίας του Raspbian.



Εικόνα 7: Η επιφάνεια εργασίας του Raspbian.

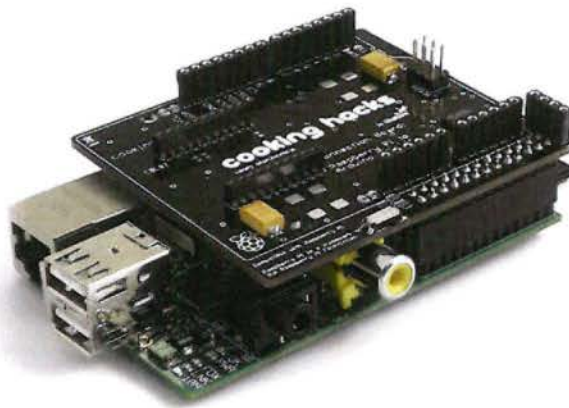
Για μελλοντική χρήση μας ενδιαφέρει να γνωρίζουμε τη διεύθυνση IP του υπολογιστή μας. Οπότε, ανοίγουμε ένα παράθυρο εντολών και με την εντολή `ip addr show eth0` βρίσκουμε τη διεύθυνση IP μας μετά τη λέξη `inet`. Δοκιμάζουμε να συνδεθούμε στον υπολογιστή Raspberry Pi από έναν άλλο προσωπικό υπολογιστή μέσω SSH για να βεβαιωθούμε πως όλες οι ρυθμίσεις μας (ενεργοποίηση SSH, αλλαγή password κτλ) έχουν γίνει σωστά.

5.3 Εγκατάσταση της ασπίδας Arduino

Αφού έχουμε διαμορφώσει τον υπολογιστή Raspberry Pi απομένει να συνδέσουμε σε αυτόν μια ασπίδα σύνδεσης Arduino σε Raspberry Pi. Σε αυτή την περίπτωση, σημαντική είναι η επιλογή της σωστής βιβλιοθήκης λογισμικού. Αφού ολοκληρωθεί και αυτή η διαδικασία, είμαστε έτοιμη να κατασκευάσουμε τις εφαρμογές του οικιακού αυτοματισμού που αφορούν στην παρούσα εργασία.

5.3.1 Ασπίδα γέφυρας Raspberry Pi σε Arduino

Για την κατασκευή μας χρησιμοποιήσαμε την ασπίδα γέφυρας Raspberry Pi σε Arduino την οποία προμηθευτήκαμε έτοιμη προς σύνδεση από το Cooking Hacks.



Εικόνα 8: Ασπίδα γέφυρας Raspberry Pi σε Arduino.

Όπως και με κάθε άλλη συσκευή που χρησιμοποιήσαμε μέχρι σήμερα, έτσι και με την ασπίδα του Arduino προηγείται το στήσιμό της. Αρχικό στάδιο αυτής της διαδικασίας είναι η σύνδεσή της με την πλακέτα του Raspberry Pi υπολογιστή μας ενώ στη συνέχεια θα πρέπει να εγκαταστήσουμε το κατάλληλο λογισμικό.

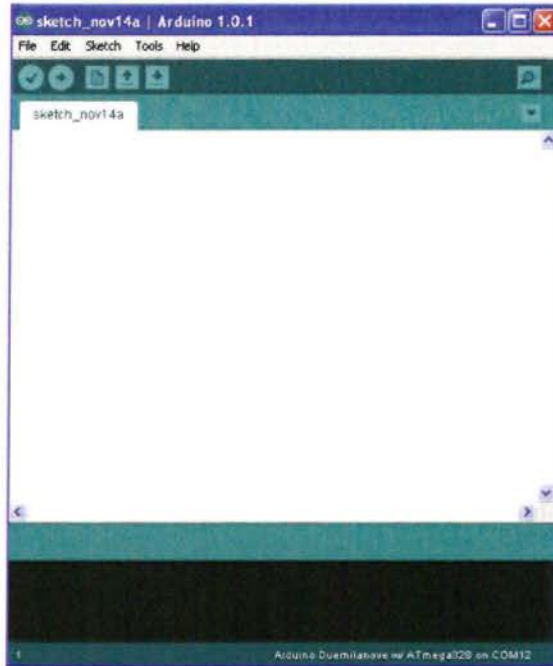
Επιθυμώντας την εγκατάσταση του σωστού λογισμικού για την ασπίδα μας πρέπει να προσδιορίσουμε την ακριβή έκδοση του υπολογιστή Raspberry Pi που έχουμε προμηθευτεί. Ο πιο εύκολος τρόπος για να επιτευχθεί αυτό είναι να κοιτάξουμε πάνω στην πλακέτα του Raspberry Pi. Αν πάνω σε αυτήν υπάρχει τυπωμένη η ένδειξη «Made in the UK» έχουμε στα χέρια μας έναν υπολογιστή δεύτερης έκδοσης. Αν για κάποιο λόγο δεν έχουμε τη δυνατότητα να κοιτάξουμε αυτή τη στιγμή πάνω στην πλακέτα, μπορούμε να πληκτρολογήσουμε σε ένα παράθυρο εντολών την εντολή `cat/proc/cpuinfo` για να πάρουμε τις πληροφορίες που αφορούν στον υπολογιστή Raspberry Pi. Ανάλογα με την ένδειξη που υπάρχει στη θέση Revision προσδιορίζουμε και την έκδοση του υπολογιστή που έχουμε προμηθευτεί.

Στη συνέχεια, συνδέουμε τη γέφυρα στον υπολογιστή. Παίρνουμε τη γέφυρα στα χέρια μας και αναγνωρίζουμε την τοποθεσία του σημείου σύνδεσης (μαύρου χρώματος) στο κάτω μέρος της. Μέσω αυτής της θέσης σύνδεσης, προσδένουμε την ασπίδα στα GPIO pins του υπολογιστή. Η σύνδεση θα πρέπει να είναι σταθερή και όχι χαλαρή.

Μετά την ολοκλήρωση της διαδικασίας σύνδεσης της ασπίδας με τον υπολογιστή, σειρά έχει η εγκατάσταση του λογισμικού το οποίο επιτρέπει τον έλεγχο της μέσω του Raspberry Pi. Για να επιτευχθεί η επικοινωνία μεταξύ του υπολογιστή και της ασπίδας του Arduino θα πρέπει πρώτα να εγκαταστήσουμε τη βιβλιοθήκη arduPi. Το arduPi ουσιαστικά αποτελεί ένα σύνολο από συγκεκριμένα αρχεία C++ τα οποία έχουν γραφτεί από την ομάδα των Cooking Hacks και σκοπός τους είναι να επιτρέπουν στις όποιες εφαρμογές έχουν δημιουργηθεί για τον Raspberry Pi να αλληλεπιδρούν με την ασπίδα μέσω των συνήθων συναρτήσεων που μπορεί κανείς να συναντήσει στις εφαρμογές για Arduino.

Το Arduino IDE δε χρειάζεται στις εφαρμογές που θα αναπτύξουμε στην παρούσα εργασία. Ωστόσο, επιθυμώντας να εκμεταλλευτούμε το σημαντικό όγκο πληροφοριών και ήδη υπαρχόντων προγραμμάτων του, το εγκαταστήσαμε. Η διαδικασία ήταν πολύ εύκολη: σε ένα παράθυρο εντολών πληκτρολογούμε τις εντολές `is up-to-date`: και `sudo apt-get update` για να εμφανιστούν πρώτα όσες ενημερώσεις υπάρχουν διαθέσιμες για τον υπολογιστή μας. Στη συνέχεια εγκαθιστούμε το IDE μέσω της εντολής `sudo apt-get install arduino`. Αυτή η εγκατάσταση θα εμφανίζει το Arduino IDE πλέον και στη μπάρα εκκίνησης του Raspbian, στην κατηγορία των Ηλεκτρονικών. Εκκινούμε το IDE μέσω του Raspberry Pi και συναντούμε ένα κενό παράθυρο με ελεύθερο χώρο για να γράψουμε τον κώδικά μας καθώς και για να φορτώσουμε ήδη υπάρχοντα προγράμματα. Το παράθυρο αυτό φαίνεται στην εικόνα 9. Τα προγράμματα που γράφουμε σε αυτό το παράθυρο, όπως έχουμε προαναφέρει, ονομάζονται πλέον σκίτσα (sketches). Στο μενού του παραθύρου υπάρχει ένα κουμπί με ένα μικρό τρίγωνο στο εσωτερικό του. Αυτό το κουμπί χρησιμοποιείται για τη μεταγλώττιση του κώδικα και την μετακίνησή του στο Arduino. Ωστόσο, όπως επισημάνθηκε στην αρχή του κεφαλαίου, εμείς δε χρησιμοποιήσαμε μικροπλακέτα Arduino αλλά έναν Raspberry Pi και μια γέφυρα Arduino. Εξαιτίας αυτής της διαφορετικής προσέγγισης του συστήματος, αντί για τον μεταγλωττιστή που φέρει ενσωματωμένο το IDE, εμείς χρησιμοποιήσαμε ένα μεταγλωττιστή C++ μέσω του παραθύρου εντολών. Βέβαια, οι έμπειροι

προγραμματιστές μπορούν να διαμορφώνουν οι ίδιοι τον τρόπο με τον οποίο θα χρησιμοποιούν τις διάφορες ιδιότητες και τα διάφορα εργαλεία του Arduino (είναι πολύ εύκολο να βρεθούν στο διαδίκτυο πάρα πολλά εργαλεία για κάθε είδους εφαρμογή).



Εικόνα 9: Το παράθυρο συγγραφής κώδικα και προβολής προγραμμάτων του Arduino.

Τέλος, η δημιουργία προγραμμάτων είναι εφικτή με κάθε γνωστό τρόπο. Έτσι, εμείς προτιμήσαμε να χρησιμοποιήσουμε έναν γνωστό σε εμάς editor για τη δική μας διευκόλυνση και στη συνέχεια να μεταγλωττίσουμε τον κώδικά μας μέσω της γραμμής εντολών. Ο editor αυτός είναι το Leafpad ο οποίος είναι ελεύθερο λογισμικό και εγκαθίσταται σε Linux συστήματα όπως το δικό μας. Είναι ελαφρύς και τον προτιμήσαμε γιατί είναι ο προεπιλεγμένος editor στο περιβάλλον LXDE του Raspbian.

Αναφορά έγινε, επίσης, στην ανάγκη εγκατάστασης της βιβλιοθήκης arduPi των Cooking Hacks αφού αυτή είναι που θα μας επιτρέψει να γράψουμε εφαρμογές για Arduino χωρίς να χρειαστούμε την πλακέτα του αλλά χρησιμοποιώντας τον Raspberry Pi. Αφού, λοιπόν, έχουμε ήδη πιστοποιήσει την έκδοση του υπολογιστή μας (ο δικός μας είναι 2^{ης} έκδοσης), θα προμηθευτούμε την ανάλογο αρχείο από το διαδίκτυο. Κατασκευάζουμε μια τοποθεσία για να εγκαταστήσουμε τη βιβλιοθήκη με την εντολή `mkdir arduPi` `cd arduPi` και κατεβάζουμε το ανάλογο αρχείο με την εντολή `wget http://www.cooking-hacks.com/skin/frontend/default/cooking/images/catalog/documentation/raspberry_arduino_shield/ arduPi_rev2. tar.gz`. Με αυτόν τον τρόπο αποθηκεύσαμε στην τοποθεσία που μόλις δημιουργήσαμε το ανάλογο αρχείο σε μορφή `tar.gz`. Στο παράθυρο εντολών πληκτρολογούμε την εντολή `tar xzf arduPi_rev2. tar.gz` και ξεκινάμε την εξαγωγή των περιεχομένων του φακέλου. Μόλις ολοκληρωθεί η διαδικασία αυτή, συναντούμε 3 νά αρχεία στην τοποθεσία

που δημιουργήσαμε, τα `arduPi.h`, `arduPi.cpp` και `arduPi_template.cpp`. Τα δύο πρώτα είναι αυτά τα οποία καθιστούν ικανή την επικοινωνία μεταξύ Raspberry Pi και ασπίδας Arduino ενώ το τρίτο αποτελεί ένα αρχείο-υπόδειγμα για τη συγγραφή νέων εφαρμογών. Το μόνο που χρειάζεται να κάνουμε είναι να μεταγλωττίσουμε το αρχείο `arduPi.cpp` και γι'αυτή τη διεργασία χρησιμοποιούμε τον C++ μεταγλωττιστή με την εντολή `g++ -c arduPi.cpp -o arduPi.o` και έχουμε ένα έτοιμο πρόγραμμα σύστημα Raspberry Pi-Arduino shield με έτοιμο λειτουργικό σύστημα και το απαραίτητο λογισμικό. Μπορούμε, λοιπόν, να ασχοληθούμε με την κατασκευή των εφαρμογών μας.

ΚΕΦΑΛΑΙΟ 6

ΕΦΑΡΜΟΓΕΣ ΟΙΚΙΑΚΟΥ ΑΥΤΟΜΑΤΙΣΜΟΥ

6.1 Εισαγωγή

Στο τελευταίο αυτό Κεφάλαιο της παρούσας εργασίας, αυτό θα ασχοληθούμε με τη δημιουργία μερικών στοιχειωδών εφαρμογών οικιακού αυτοματισμού οι οποίες είναι θεμελιώδεις από την άποψη ότι πάνω σε αυτές μπορεί να οικοδομηθεί ένα ακόμη πιο σύνθετο σύστημα οικιακού αυτοματισμού.

Αρχικά, λοιπόν, και έχοντας έτοιμο το hardware όπως το συνδέσαμε στο προηγούμενο κεφάλαιο, θα κατασκευάσουμε ένα θερμόμετρο. Στη συνέχεια, θα χτίσουμε πάνω στην εφαρμογή του θερμόμετρου κατασκευάζοντας έναν θερμοστάτη. Τέλος, συνοδεύουμε τις δύο αυτές εφαρμογές με μια προσπάθεια δημιουργίας μιας βάσης δεδομένων, αφού οι βάσεις δεδομένων είναι βασικό χαρακτηριστικό των M2M επικοινωνιών.

6.2 Κατασκευή ενός βασικού θερμόμετρου

Για την κατασκευή του βασικού μας θερμόμετρου χρειαζόμαστε το σύστημα του υπολογιστή Raspberry Pi που είναι συνδεδεμένος με την ασπίδα του Arduino. Χρειαζόμαστε, επίσης, μια πλακέτα breadboard, μια αντίσταση 10k καθώς και έναν θερμίστορ. Ο θερμίστορ είναι μια αντίσταση ειδικού τύπου η οποία μεταβάλλεται έντονα με τις μεταβολές της θερμοκρασίας. Ένας θερμίστορ είναι αυτός που φαίνεται στην εικόνα που ακολουθεί.



Εικόνα 1: Ένας τυπικός θερμίστορ που διατίθεται στο εμπόριο.

Ο θερμίστορ, μεταξύ των άλλων λειτουργιών του, μπορεί να χρησιμοποιηθεί και σαν αισθητήρας θερμοκρασίας αφού είναι γεγονός πως σχεδόν όλες οι συσκευές ρύθμισης θερμοκρασίας περιλαμβάνουν αυτό το ηλεκτρονικό στοιχείο. Ωστόσο, ένας θερμίστορ δεν πρέπει να συγχέεται με τους τυπικούς αισθητήρες θερμοκρασίας αφού η κατασκευή του γίνεται με εντελώς διαφορετικά υλικά. Έτσι, ενώ οι αισθητήρες θερμοκρασίας αποτελούνται κύριως από μέταλλα, ο θερμίστορ είναι κατασκευασμένος από κεραμικό ή πολυμερές υλικό. Επίσης,

υπάρχει διαφοροποίηση σχετικά με την απόκριση του ηλεκτρονικού στοιχείου στη θερμοκρασία με τους αισθητήρες θερμοκρασίας να μπορούν να ανταποκριθούν σε ένα μεγάλο εύρος θερμοκρασιών αλλά με το θερμίστορ να μπορεί να έχει βέλτιστη απόκριση σε ένα μικρό εύρος θερμοκρασιών από -90°C ως 130°C που είναι ένα εύρος θερμοκρασιών που αφορά τον οικιακό αυτοματισμό. Τέλος, από άποψης λογισμικού θα χρειαστούμε επιπροσθέτως το Geany IDE ενώ μεγάλη σημασία στις εφαρμογές μας παίζει και η εντολή `make` του UNIX περιβάλλοντος. Με χρήση των παραπάνω εργαλείων, λοιπόν, θα προσπαθήσουμε να συγγράψουμε μια εφαρμογή η οποία θα μετατρέπει την αντίσταση που θα προκύπτει από το κύκλωμα σε θερμοκρασίες μετρημένες σε βαθμούς Κελσίου, βαθμούς Κέλβιν και βαθμούς Φάρεναϊτ.

6.2.1 Ορισμός Θερμομέτρου

Το θερμόμετρο είναι ένα απλό σύστημα το οποίο ανιχνεύει και καταγράφει μεταβολές στη θερμοκρασία του περιβάλλοντος ή ενός συγκεκριμένου σώματος. Έχει τις καταβολές του στα αρχαία χρόνια αφού το θέμα της μέτρησης της θερμοκρασίας απασχολούσε τον άνθρωπο σε κάθε πολιτισμικό πλαίσιο ανά τους αιώνες. Η εξέλιξή του από τις πρώτες προσπάθειες δημιουργίας του υπήρξε ραγδαία. Έτσι, τα πρώτα θερμοσκόπια των Ελλήνων Ήρωνα και Φίλωνα έχουν πλέον πάρει τη μορφή σύγχρονων συσκευών πολλαπλών χρήσεων. Παραδοσιακά, όμως, το θερμόμετρο έχει τη μορφή γυάλινης ράβδου πληρωμένης σε κάποιο ποσοστό με θερμοευαίσθητη ουσία όπως ο υδράργυρος και βαθμονομημένης σύμφωνα με την κλίμακα του Κελσίου ή του Φάρεναϊτ.

Με την ανάπτυξη των μικροηλεκτρονικών έγινε εφικτή η κατασκευή των ψηφιακών θερμομέτρων. Αυτή η τεχνολογική εξέλιξη ήταν ιδιαίτερα σημαντική στο χώρο των οικιακών εφαρμογών, αφού πλέον μπορούμε να έχουμε εποπτεία των θερμοκρασιών σε όλα τα δωμάτια ενός σπιτιού καθώς και σε χώρους αποθήκευσης ευαίσθητων προϊόντων, όπως είναι για παράδειγμα μαι αποθήκη τροφίμων.

Το δικό μας θερμόμετρο, λοιπόν, θα είναι ένα τέτοιο ψηφιακό θερμόμετρο το οποίο θα αποστέλλει τις καταγραφές του στον υπολογιστή Raspberry Pi και αυτός με τη σειρά του θα τις εμφανίζει στην οθόνη μέσω ενός παραθύρου εντολών.

6.2.2 Υλισμικό του Θερμομέτρου

Στην εφαρμογή αυτή χρησιμοποιούμε διάφορα και διαφορετικά μεταξύ τους κομμάτια. Αυτά τα κομμάτια μπορούν είτε να συγκολληθούν πάνω στην ασπίδα του Arduino, είτε να συνδεθούν με το σύστημα μέσω μιας breadboard πλακέτας. Εμείς επιλέξαμε τη δεύτερη μέθοδο γιατί μέρος των υλικών μας θέλαμε να τα χρησιμοποιήσουμε για την κατασκευή και των υπολοίπων εφαρμογών.

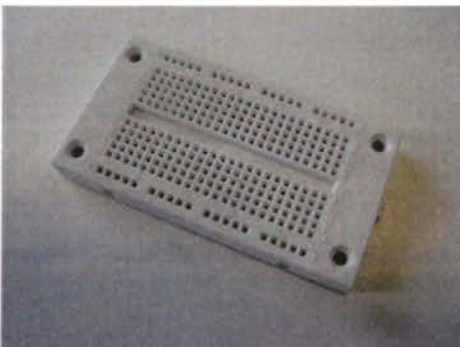
Αρχικά, κατά τη διαδικασία σύνδεσης των μερών του θερμομέτρου μας, βεβαιωνόμαστε πως ο Raspberry Pi είναι σβηστός. Σε αντίθετη περίπτωση,

μπορεί να προκληθούν σημαντικές ζημιές στο σύστημα αν κατά τη διάρκεια της συγκόλλησης των μερών μας «λερωθεί» κάποιο τμήμα της πλακέτας.

Πριν, όμως, εξετάσουμε τον τρόπο με τον οποίο συνδέσαμε τη διάταξη μας, θεωρούμε σκόπιμο να δούμε περιληπτικά τα επιμέρους στοιχεία του.

Αρχικά, στο σύστημα χρησιμοποιήθηκαν δύο διαφορετικά είδη αντιστάτων. Ο αντιστάτης είναι ένα ηλεκτρικό δίπολο το οποίο χρησιμοποιείται για τον περιορισμό της διερχόμενης ηλεκτρικής ενέργειας. Αυτός ο περιορισμός ονομάζεται ηλεκτρική αντίσταση και μετριέται σε Ωμ. Οι αντιστάτες χωρίζονται σε δύο μεγάλες κατηγορίες, τους σταθερούς και τους μεταβλητούς. Ο μεταβλητός αντιστάτης του συστήματός μας είναι ο θερμίστορ. Ο θερμίστορ, όπως προαναφέρθηκε, έχει την ιδιότητα να μεταβάλλει την αντίστασή του ανάλογα με τη θερμοκρασία που επικρατεί στο περιβάλλον του. Υπάρχουν δύο είδη αυτού του ηλεκτρονικού στοιχείου, ο θερμίστορ θετικής σταθεράς (PTC) και ο αντιστοιχος αρνητικής σταθεράς (NTC). Αυτοί οι δύο διαφοροποιούνται από το γεγονός ότι στον πρώτο θερμοκρασία και αντίσταση περιβαλλοντος είναι ευθέως ανάλογα ποσά ενώ στο δεύτερο αύξηση της θερμοκρασίας περιβάλλοντος συνεπάγεται μείωση της αντίστασης του θερμίστορ. Αναφορικά με τη χρήση του, υπάρχουν δύο μεγέθη που μας είναι σημαντικά. Το πρώτο είναι η αντίσταση του στοιχείου σε θερμοκρασία περιβάλλοντος (ορίζουμε τη θερμοκρασία περιβάλλοντος στους 25 βαθμούς Κελσίου) και το δεύτερο είναι η σταθερά του που αντιστοιχεί στη μεταβολή της αντίστασης για κάθε μεταβολή της θερμοκρασίας. Αυτά τα δύο μεγέθη μας δόθηκαν με την αγορά του θερμίστορ. Έτσι, για το δικό μας στοιχείο που είναι ένας θερμίστορ 10K έχουμε μια σταθερά του ύψους 4050K και μια αντίσταση του ύψους των 10K Ω. Ο σταθερός αντιστάτης του συστήματός μας είναι ένας αντιστάτης σταθερής αντίστασης 10K Ω. Η χρήση αυτού είναι να μετατρέπει τη μεταβλητή αντίσταση σε τάση, η οποία με τη σειρά της θα μπορεί να αναγνωστεί από το αντίστοιχο pin του συστήματος Raspberry Pi-Arduino.

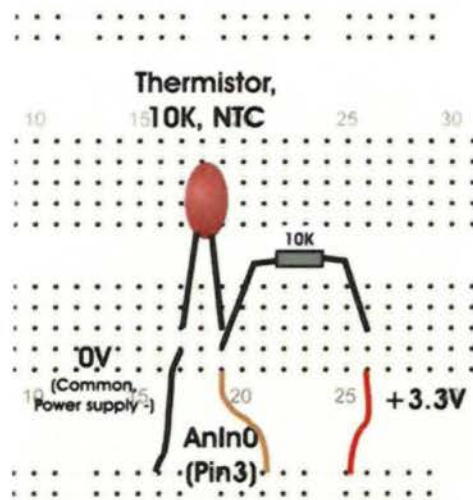
Στη συνέχεια, όπως είναι εύκολα αντιληπτό, χρησιμοποιήσαμε καλώδια για τη σύνδεση συγκεκριμένων στοιχείων. Συγκεκριμένα, χρειαστήκαμε τρία στον αριθμό καλώδια: το πρώτο συνδέθηκε με το 5V pin της ασπίδας του Arduino, το δεύτερο με τη γη και το τρίτο με το pin 0. Σύμφωνα με τις οδηγίες σήμανσης των καλωδίων, αυτά τα τρία καλώδια ήταν αντίστοιχα κόκκινο, μαύρο και κίτρινο.



Ακόμα χρησιμοποιήσαμε μια πλακέτα breadboard, στην οποία πάνω έγινε η συνδεσμολογία όπως θα την περιγράψουμε ευθύς αμέσως, έτσι ώστε να μην κατασκευαστεί μια μόνιμη διάταξη αλλά να μπορούμε να επαναχρησιμοποιήσουμε όσα από τα ηλεκτρονικά στοιχεία χρειαζόμαστε στις επόμενες εφαρμογές.

Εικόνα 2: Η πλακέτα breadboard.

Η σύνδεση των παραπάνω στοιχείων, την οποία θα περιγράψουμε βήμα-βήμα, υπήρξε εύκολη διαδικασία. Αρχικά, συνδέσαμε το κόκκινο καλώδιο με το pin των 5V της ασπίδας στο ένα του άκρο και την παροχή του ρεύματος της πλακέτας breadboard στο άλλο. Στη συνέχεια, με το μαύρο καλώδιο συνδέσαμε τη γείωση της πλακέτας με το pin της γείωσης. Σειρά είχαν οι αντιστάτες, με τον 10K αντιστάτη να συνδέεται στην περιοχή της παροχής του ρεύματος στην οποία συνδέθηκε και το κόκκινο καλώδιο με το ένα του άκρο ενώ το άλλο συνδέθηκε με ακροδέκτη στο μέσο της πλακέτας breadboard και το θερμίστορ να συνδέεται με το ένα του άκρο στη γείωση και το δεύτερο στην ίδια σειρά ακροδεκτών στο μέσο της πλακέτας που συνδέθηκε και ο σταθερός αντιστάτης.



Εικόνα 3: Εικονική αναπαράσταση της συνδεσμολογίας των αντιστατών.

Το μόνο που έχει απομείνει είναι να συνδέσουμε το κίτρινο καλώδιο στο pin 0 της ασπίδας στο ένα άκρο του ενώ το άλλο άκρο συνδέθηκε στην ίδια σειρά ακροδεκτών που συνδέθηκαν τα άκρα των αντιστατών. Η συνδεσμολογία που περιγράφηκε φαίνεται στην εικόνα 3.

6.2.3 Λογισμικό του Θερμομέτρου

Σειρά έχει η συγγραφή του κώδικα που θα επιτρέπει τη μετατροπή των δεδομένων τα οποία «διαβάζει» ο θερμίστορ σε μορφή που να είναι ευανάγνωστη από το χρήστη ανάλογα με τη μονάδα μέτρησης της θερμοκρασίας την οποία επιθυμεί να χρησιμοποιεί. Γι'αυτή την εργασία θα χρησιμοποιήσουμε το Geany IDE.

Το Geany IDE είναι ένα ελαφρύ περιβάλλον ανάπτυξης εφαρμογών συμβατό με Linux, το οποίο μπορεί να εγκατασταθεί στο Raspbian του υπολογιστή μας και να χρησιμοποιηθεί για τη συγγραφή κώδικα σε γλώσσα προγραμματισμού Arduino/C++. Ο λόγος για τον οποίο προτιμήσαμε το συγκεκριμένο προγραμματιστικό περιβάλλον είναι πως έχει ακριβώς τις ίδιες προγραμματιστικές δυνατότητες με το Arduino IDE ενώ παράλληλα είναι πιο εύχρηστο και πιο γρήγορο από αυτό αφού δίνει τη δυνατότητα αυτοποιημένης συγγραφής, ονοματοδοσίας και μεταγλώττισης του κώδικα.



Η εγκατάσταση του Geany IDE στο σύστημά μας γίνεται με τρόπο που έχει περιγραφεί και παραπάνω για λοιπά λογισμικά. Έτσι, χρησιμοποιώντας την εντολή `sudo apt-get install geany` σε ένα παράθυρο εντολών και χρησιμοποιώντας το πλήκτρο Υ για να επιβεβαιώσουμε την εντολή μας, εγκαταστήσαμε το Geany το οποίο και πρωτοεμφανίστηκε στη συνέχεια στο μενού Programming.

Εικόνα 4: Το χαρακτηριστικό εικονίδιο του Geany.

Βρίσκουμε το εικονίδιο του Geany στο μενού αυτό, το επιλέγουμε και εκκινούμε το λογισμικό. Εμφανίζεται, τότε, ένα παράθυρο συγγραφής κώδικα. Παρά το γεγονός ότι το Geany παρέχει μεγάλη γκάμα δυνατοτήτων και εργαλείων, εμείς χρησιμοποιήσαμε αυτές από το Build μενού οι οποίες μας παρείχαν τη δυνατότητα να γράψουμε κώδικα, να τον αποθηκεύσουμε και να μεταγλωττίσουμε το πρόγραμμά μας.

Μεγάλης σημασίας για τη μεταγλώττιση του κώδικά μας υπήρξε η δυνατότητα κατασκευής ενός Makefile. Η εντολή Makefile εκτελείται μέσω της εντολής `make` του Linux. Έτσι, μπορούμε να μεταγλωττίζουμε εκτελέσιμα αρχεία αποθηκεύοντας τις παραμέτρους σε ένα Makefile και να τα καλούμε όποτε θέλουμε. Αυτή η μέθοδος μας επιτρέπει ουσιαστικά να αυτοματοποιούμε τη διαδικασία της μεταγλώττισης και να την επαναχρησιμοποιούμε κατά το δοκούν χωρίς να χρειάζεται να γράφουμε κάθε φορά τις ίδιες εντολές.

Αναφορικά με τον κώδικα της εφαρμογής καθαυτό, θα χρησιμοποιήσουμε τη βιβλιοθήκη `arduino` καθώς και γλώσσα προγραμματισμού αλλά και σύνταξη `Arduino/C++`. Όπως προείπαμε, σκοπός του κώδικα είναι να μπορεί να συλλέγει τιμές από το κύκλωμα και να τις μετατρέπει σε αναγνώσιμες θερμοκρασίες. Γι'αυτό το λόγο θα χρησιμοποιήσουμε την εξίσωση Steinhart-Hart

$$\frac{1}{T} = A + b \ln(R) + c [\ln(R)]^3$$

η οποία μοντελοποιεί την αντίσταση του θερμίστορ σε διαφορετικές θερμοκρασίες και μπορεί να κωδικοποιηθεί σε εφαρμογή η οποία θα μας δίνει τη θερμοκρασία σε βαθμούς Κελσίου, Κέλβιν και Φάρεναϊτ. Για την εφαρμογή μας διαλέγουμε μια ιδιαίτερη μορφή της παραπάνω εξίσωσης, αυτή όπου

$$\alpha = \frac{1}{T_0} - \frac{1}{B} \ln(R_0), \quad b = \frac{1}{B}, \quad c = 0$$

με αποτέλεσμα η εξίσωση να λύνεται για τη θερμοκρασία ως εξής

$$T = \frac{B}{\ln(R/r_\infty)}$$

Χρησιμοποιώντας τον πίνακα τιμών που μας προμήθευσε η κατασκευάστρια εταιρία του θερμίστορα θα μπορέσουμε να έχουμε τις τιμές των σταθερών που απαιτούνται για τους υπολογισμούς μας. Έτσι, τα μόνα δεδομένα που χρειαζόμαστε είναι η θερμοκρασία του δωματίου σε βαθμούς Κέλβιν, τη σταθερά του θερμίστορα και την αντίστασή του σε θερμοκρασία δωματίου.

Για να γράψουμε τον κώδικα, ανοίξαμε ένα παράθυρο εντολών και αποθηκεύσαμε τον -ανύπαρκτο προς το παρόν- κώδικα με το όνομα thermometer.cpp. Στη βιβλιοθήκη του arduPi υπάρχει έτοιμο δείγμα κώδικα για θερμόμετρο, το arduPi_template.cpp τον οποίο και θα εμπλουτίσουμε.

Αρχίζουμε με την εισαγωγή των απαραίτητων βιβλιοθηκών καθώς και τον ορισμό των απαραίτητων σταθερών. Αυτό γίνεται με τον παρακάτω κώδικα:

```
//Eisagei ti vivliothiki ArduPi
#include "arduPi.h"

//Eisagei ti vivliothiki Math
#include <math.h>

#define TENKRESISTOR 10000 //our 10K resistor

#define BETA 4050 // This is the Beta Coefficient of
your thermistor

#define THERMISTOR 10000 // I antistasi tou
//thermistor se thermokrasia domatiou

#define ROOMTEMPK 298.15 //thermokrasia domatiou se
//kelvin
//(25 Celsius).

#define READINGS 7
```

Μέρος αυτού του κώδικα προέρχεται από το δείγμα της βιβλιοθήκης. Επίσης, υπάρχουν αναφορές στη βιβλιοθήκη Math η οποία περιλαμβάνει κάποιες πολύπλοκες μαθηματικές συναρτήσεις τις οποίες μπορούμε να τις χρησιμοποιούμε απλά καλώντας τις χωρίς να πρέπει να τις πληκτρολογούμε κάθε φορά. Μια από αυτές τις συναρτήσεις είναι η $\log()$ που όπως φαίνεται από τον ορισμό της εξίσωσης Steinhart-Hart θα μας χρειαστεί στους υπολογισμούς μας.

Αρχικοποιούμε τις σταθερές μέσω της εντολής #define ενώ οι σταθερές μας είναι οι εξής:

- **TENKRESISTOR:** Η σταθερή αντίσταση της διάταξης στην οποία έχουμε εκχωρήσει την τιμή 10.000.
- **THERMISTOR:** Η αντίσταση του θερμομέτρου σε θερμοκρασία δωματίου.
- **BETA:** Η σταθερά βήτα της εξίσωσης.
- **ROOMTEMPK:** Η θερμοκρασία σε βαθμούς Κέλβιν που αντιστοιχεί στους 25 βαθμούς Κελσίου.
- **READINGS:** Θα λάβουμε 7 μετρήσεις και θα βρούμε το μέσο όρο αυτών ώστε να έχουμε πιο ακριβή δεδομένα.

Στη συνέχεια γράφουμε το κυρίως μέρος του κώδικα:

```
/******  
*TO MEROS OPOU PREPEI NA ORISTOUN OLES OI METAVLITES*  
*POU PERIEXONTAI STO PROGRAMMA MAS *  
*****/  
  
/******  
* KODIKAS ARDUINO *  
*****/  
  
int main () {  
    setup();  
    while(1) {  
        loop();  
    }  
    return (0);  
}  
  
void setup(void) {  
    printf("Starting up thermometer \n");  
    Wire.begin();  
}  
  
void loop(void) {  
    float avResistance;  
    float resistance;  
    int combinedReadings[READINGS];  
    byte val0;  
    byte val1;  
  
    //Oi metavlites tis thermokrasias  
    float kelvin;  
    float fahrenheit;  
    float celsius;  
    int channelReading;  
    float analogReadingArduino;
```

Στο μέρος αυτό του κώδικα ορίσαμε μια σειρά μεταβλητών οι οποίες κατηγοριοποιούνται ως εξής:

- Ανάγνωση αντιστάσεων: Είναι οι τύπου float μεταβλητές avResistance και resistance και η τύπου byte val0 και val1. Οι πρώτες δύο χρησιμοποιούνται κατά την εκτέλεση του προγράμματος για να εκχωρούνται τιμές που αφορούν στους υπολογισμούς αντιστάσεων ενώ οι δύο τελευταίες χρησιμοποιούνται για την αποθήκευση των μετρήσεων του αναλογικού pin 0 της ασπίδας.
- Υπολογισμοί θερμοκρασιών: Είναι οι τύπου float kelvin, Fahrenheit και celsius και χρησιμοποιούνται για την εκχώριση της τιμής της θερμοκρασίας στην κλίμακα που υποδικνύει το όνομα της κάθε μιας.

Αφού, λοιπόν, ορίσουμε τις μεταβλητές μας, ξεκινάει η ανάγνωση των τιμών του αναλογικού pin 0, αφού αυτό αρχικοποιηθεί. Αυτό επιτυγχάνεται με τον παρακάτω

κώδικα:

```
ADC mappings
```

```
Pin Address
```

```
0 0xDC
1 0x9C
2 0xCC
3 0x8C
4 0xAC
5 0xEC
6 0xBC
7 0xFC
```

```
Wire.beginTransmission(8);
Wire.write(byte(0xDC));
Wire.endTransmission();
```

Τώρα μπορούμε πλέον να λαμβάνουμε και μετρήσεις απο το pin 0. Ωστόσο, για να διασφαλίσουμε ην αλήθεια των δεδομένων μας θα παίρνουμε δύο μετρήσεις, γεγονός που επιτυγχάνεται με τον εξής κώδικα:

```
//Pairnουμε ta dyo byte poy epistrefei to
//analogiko pin 0, ta syndyazoume kai grafoume
//tin timi sto combinedReadings

for(int r=0; r<READINGS; r++){
    Wire.requestFrom(8, 2);
    val0 = Wire.read();
    val1 = Wire.read();
    channelReading = int(val0)*16 +
int(val1>>4);
    analogReadingArduino = channelReading * 1023
/4095;
    combinedReadings[r] = analogReadingArduino;
    delay(100); }
```

Όπως μπορούμε να παρατηρήσουμε, έχουμε εφαρμόσει κάποιου τύπου υπολογισμό μεταξύ των δύο συνδυασμένων bytes. Αυτό συμβαίνει γιατί πρέπει τα αποτελέσματα των υπολογισμών να μετατραπούν σε 10-bit Arduino ανάλυση. Με αυτό τον τρόπο ξεπερνάμε το σκόπελο της μη ύπαρξης της εντολής analogRead() στο Geany IDE, εντολή η οποία περιλαμβάνεται όμως στο Arduino IDE.

Στη συνέχεια, θα πρέπει να υπολογίσουμε τη μέση αντίσταση ως μέσο όο των ληφθέντων δεδομένων μέσω του κώδικα:

```
//Υπολογισμος του M0 ton 7 metriseon gia na
//paroume pio eggyri timi tis antistasis

avResistance = 0;
for (int r=0; r<READINGS; r++) {
    avResistance += combinedReadings[r];
}
avResistance /= READINGS;
```

Σειρά έχουν οι υπολογισμοί οι οποίοι θα μας δώσουν την αντίσταση.
Χρησιμοποιούμε τον κώδικα:

```
avResistance = (1023 / avResistance) - 1;  
avResistance = TENKRESISTOR / avResistance;  
resistance = avResistance / THERMISTOR;
```

Αφού έχουμε την αντίσταση μπορούμε να υπολογίσουμε τη θερμοκρασία, εισάγοντας τη σχέση Steinhart-Hart που συζητήσαμε παραπάνω. το αποτέλεσμα της σχέσης αυτής κατά την επίλυσή της ως προς τη θερμοκρασία δίνεται σε βαθμούς Κέλβιν. Αυτό το αποτέλεσμα θα το εμφανίσουμε στην οθόνη μας και στη συνέχεια θα το μετατρέψουμε σε θερμοκρασίες της κλίμακας Κελσίου και Φάρεναϊτ, τις οποίες επίσης θα τυπώσουμε στην οθόνη μας. Στο τέλος, θα εισάγουμε μια καθυστέρηση μέχρι την έναρξη της επόμενης διαδικασίας μετρήσεων και υπολογισμών:

```
kelvin = log(resistance);  
kelvin /= BETA;  
kelvin += 1.0 / ROOMTEMPK;  
kelvin = 1.0 / kelvin;  
printf("Temperature in K ");  
printf("%f \n", kelvin);  
  
//Metatropi apo Kelvin se Celsius  
celsius = kelvin - 273.15;  
printf("Temperature in C ");  
printf("%f \n", celsius);  
  
//Metatropi apo Celsius se Fahrenheit  
fahrenheit = (celsius * 1.8) + 32;  
printf("Temperature in F ");  
printf("%f \n", fahrenheit);  
  
//Kathysterisi 3 sec mexri tin epomeni metrisi  
delay(3000);  
}
```

Τώρα θα δημιουργήσουμε το Makefile της εφαρμογής με την εντολή

```
Thermo: arduPi.o g++ -lrt -lpthread thermometer.cpp arduPi.o -o  
thermometer
```

και σώζουμε με το όνομα Makefile.

Έχουμε πλέον το εκτελέσιμο αρχείο thermometer, το οποίο το εντοπίζουμε στη βιβλιοθήκη arduPi και το εκκινούμε μέσω της εντολής

```
sudo ./thermometer
```

Η εικόνα την οποία βλέπουμε στην οθόνη μας είναι η παρακάτω:

```
pi@raspberrypi ~/arduPi $ sudo ./thermometer
Starting up thermometer
Temperature in K 302.940094
Temperature in C 29.790094
Temperature in F 85.622169
```

Εικόνα 5: Οι έξοδοι του λογισμικού του θερμομέτρου.

6.2.4 Χρήση του θερμομέτρου σε M2M επικοινωνίες

Έστω το παράδειγμα ενός θερμομέτρου το οποίο θα μετράει τη θερμοκρασία ενός δωματίου. Ο προμηθευτής του συστήματος θα πρέπει να δημοσιεύσει το αντίστοιχο πρωτόκολλο SOAP (Simple Object Access Protocol) σε βάση δεδομένων η οποία είναι συνήθως της μορφής UDDI. Έτσι, κάθε προγραμματιστής ο οποίος επιθυμεί να χρησιμοποιήσει το θερμομέτρο αυτό σε κάποια εφαρμογή του θα μπορεί να έχει πρόσβαση στην περιγραφή της διεπαφής, η οποία είναι συνήθως ένα αρχείο WSDL (Web Service Description Language). Μεταγλωττίζοντας τον παρεχόμενο αυτό κώδικα ανεξαρτητοποιεί την εφαρμογή από τον τρόπο με τον οποίο αυτή δημιουργήθηκε. Έτσι, διαφορετικές εφαρμογές που βασίζονται σε διαφορετικές τεχνολογίες μπορούν να ενοποιηθούν για την κατασκευή ενός πιο σύνθετου εμπορικού προϊόντος. Ωστόσο, ο προγραμματιστής-καταναλωτής της εφαρμογής του θερμομέτρου, στην προσπάθειά του να την ενσωματώσει σε ένα πιο σύνθετο προϊόν πρέπει να καθορίσει τον προτιμώμενο τύπο MIME (Multipurpose Internet Mail Extensions), καθώς και την κωδικοποίησή του και ο εξυπηρετητής M2M μετατρέπει τα παρεχόμενα δεδομένα στην επιθυμητή μορφή

6.3 Κατασκευή ενός θερμοστάτη

Θα συνεχίσουμε με την κατασκευή ενός θερμοστάτη χρησιμοποιώντας την εφαρμογή της υλοποίησης του θερμομέτρου ως δομικό υλικό. Ουσιαστικά, θα δημιουργήσουμε μια εφαρμογή η οποία θα χρησιμοποιεί τα δεδομένα των θερμοκρασιών για να ανοίγει και να κλείνει τα ρελέ του συστήματος. Τα ρελέ είναι τα βασικά ηλεκτρονικά στοιχεία μέσω των οποίων μπορεί να επικοινωνήσει ο Raspberry Pi υπολογιστής μας με συσκευές υψηλής τάσης. Εδώ, ουσιαστικά, έχουμε μια εφαρμογή M2M αφού δε λαμβάνουμε μόνο τα δεδομένα του περιβάλλοντος αλλά τα χρησιμοποιούμε κιόλας για να χειριστούμε κάποια συσκευή. Έτσι, θα δείξουμε με πιο τρόπο μπορούμε να ενεργοποιήσουμε τον αερισμό ενός χώρου (ανεμιστήρας, κλιματιστικό) όταν η θερμοκρασία περιβάλλοντος ξεπερνά μια ανώτατη θερμοκρασία. Στη συνέχεια, όταν φτάσουμε την επιθυμητή θερμοκρασία περιβάλλοντος, το σύστημα θα πρέπει

να κλείνει τον ανεμιστήρα ή το κλιματιστικό (εμείς χρησιμοποιήσαμε ανεμιστήρα γιατί ήταν η ηλεκτρική αυτή συσκευή η οποία υπήρχε στη διάθεσή μας) ενώ τις διακυμάνσεις της θερμοκρασίας του περιβάλλοντος τις τεστάρουμε χρησιμοποιώντας παροχή ζεστού αέρα μέσω ενός σεσουάρ μαλλιών ή με ψύξη μέσω μιας παγοκύστης.

Στο Κεφάλαιο αυτό θα χρειαστούμε τον υπολογιστή Raspberry Pi και την ασπίδα του Arduino, τη συσκευή του θερμομέτρου την οποία κατασκευάσαμε στην προηγούμενη παράγραφο, ρελέ που να είναι συμβατά με Arduino και να μπορούν να συνδεθούν με αυτό, έναν ανεμιστήρα οικιακής χρήσης ο οποίος μας καλύπτει από την άποψη ότι δε λειτουργεί σε μη ασφαλείς τάσεις καθώς και το σεσουάρ και την παγοκύστη που προαναφέραμε.

Σημαντικό είναι το σχόλιο που έγινε για τον ανεμιστήρα, ότι δηλαδή κύριο μέλημά μας είναι να διατηρήσουμε ένα επίπεδο ασφάλειας στην εργασία μας. Έτσι, χρησιμοποιούμε τα σωστά για την τάση του κυκλώματός μας ρελέ.

6.3.1 Ορισμός του θερμοστάτη

Ο θερμοστάτης είναι μια συσκευή ελέγχου η οποία χειρίζεται άλλες συσκευές που λειτουργούν με παράμετρο τη θερμοκρασία του περιβάλλοντος. Αυτή η θερμοκρασία είναι η επιθυμητή τιμή γνωστή ως setpoint, τιμή όπου η υπέρβαση της ή μη καθορίζει τη λειτουργία ή μη των συσκευών. Για παράδειγμα, μια ηλεκτρική θερμάστρα έχει ενσωματωμένο θερμοστάτη τον οποίο ρυθμίζουμε στους 18 βαθμούς Κελσίου. Αυτό σημαίνει ότι για θερμοκρασίες περιβάλλοντος μεγαλύτερες από 18 βαθμούς Κελσίου η θερμάστρα παραμένει σβηστή. Αντίθετα, για θερμοκρασίες μικρότερες από τους 18 βαθμούς Κελσίου, η θερμάστρα ανάβει θερμαίνοντας το χώρο μέχρι η θερμοκρασία περιβάλλοντος να φτάσει τους 18 βαθμούς Κελσίου οπότε και θα ξανασβήσει.

Ο θερμοστάτης είναι εξοπλισμένος με θερμοευαίσθητο ηλεκτρονικό στοιχείο, όπως για παράδειγμα ο θερμίστορ που χρησιμοποιήσαμε στην προηγούμενη εφαρμογή. Με την ανάπτυξη ελεύθερων λογισμικών όπως το Arduino και ο υπολογιστής Raspberry Pi που έχουμε στη διάθεσή μας, έγινε πλέον δυνατή η κατασκευή θερμοστατών για εξατομικευμένες χρήσεις.

6.3.1 Υλισμικό του θερμοστάτη

Το θερμόμετρο της προηγούμενης εφαρμογής είναι ο θεμέλιος λίθος για την ανάπτυξη του θερμοστάτη αφού με αυτό θα ελέγχουμε τη θερμοκρασία του περιβάλλοντος. Ωστόσο, μεγάλη σημασία έχει και η χρήση των ρελέ. Το ρελέ είναι ένας τύπος διακόπτη που ελέγχεται από έναν ηλεκτρομαγνήτη. Μας επιτρέπει να ελέγχουμε μεγάλες ποσότητες ηλεκτρικής ενέργειας καταναλώνοντας μια ελάχιστη ποσότητα από αυτή. Για παράδειγμα, στα οικιακά δίκτυα της χώρας μας ένα τυπικό ρελέ για το ρεύμα των 220V είναι των 9V. Ένα ρελέ έχει τρία σημεία, το Κανονικά Ανοιχτό, τη Μεταγωγική και το Κανονικά Κλειστό. Δύο από αυτά τα σημεία συνδέθηκαν με τον ανεμιστήρα μας

ενώ τα ρελέ που συνδέονται με συστήματα που περιλαμβάνουν Arduino έχουν και ένα επιπρόσθετο pin για να συνδεθεί με τη γη.

Εμείς χρησιμοποιήσαμε ένα ρελέ για σύνδεση στην 5V πηγή ρεύματος της ασπίδας. Για αρχή, λοιπόν, συνδέουμε το pin της παροχής ρεύματος με την παροχή ρεύματος της ασπίδας του Arduino. Βέβαια, επειδή εμείς προτιμούμε να χρησιμοποιούμε την πλακέτα breadboard η σύνδεση αυτή θα γίνει μέσω της παροχής του ρεύματος που υπάρχει πάνω σε αυτή την πλακέτα. Με τον ίδιο τρόπο συνδέουμε τη γείωση του ρελέ με τη γη της breadboard. Στη συνέχεια καλωδιώνουμε το ρελέ με ένα από τα data pins. Εμείς δεν είχαμε αποσυνδέσει το θερμομέτρο οπότε υπάρχει στο σύστημα και όλη η προηγούμενη διάταξη. Τέλος, αν η κατασκευή του ρελέ απαιτεί τροφοδοσία ηλεκτρικού ρεύματος, κάνουμε την ανάλογη σύνδεση. Προς το παρόν, η διάταξη του θερμοστάτη είναι πλήρης

6.3.2 Λογισμικό του θερμοστάτη

Η εργασία μας ξεκίνησε με τη συγγραφή ενός απλού κώδικα ο οποίος θα ανοίγει και θα κλείνει ένα ρελέ συνδεδεμένο με έναν Raspberry Pi. Στη συνέχεια τεστάρουμε ότι ο κώδικας αυτός τρέχει και έπειτα τον διαμορφώσαμε για να ενσωματωθεί με την προηγούμενη εφαρμογή.

Στο Geany IDE, λοιπόν, φτιάχνουμε ένα πρόγραμμα με το όνομα Relay.cpp στην ίδια τοποθεσία με το thermometer στην arduPi. Ο κώδικας έχει ως εξής:

```
#include "arduPi.h"

SerialPi Serial;

WirePi Wire;

/*****
 *ΤΟ ΜΕΡΟΣ ΟΠΟΥ ΠΡΕΠΕΙ ΝΑ ΟΡΙΣΤΟΥΝ ΟΛΕΣ ΟΙ          *
 *ΜΕΤΑΒΛΙΤΕΣ ΠΟΥ ΠΕΡΙΕΧΟΝΤΑΙ ΣΤΟ ΠΡΟΓΡΑΜΜΑ ΜΑΣ   *
 *****/

/*****
 *ΚΟΔΙΚΑΣ ARDUINO          *
 * *****/

int main () {
    setup();
    while(1) {
        loop();
    }
    return (0);
}

void setup() {
    pinMode(4, OUTPUT);
}

void loop()
```

```

{
    digitalWrite(4, HIGH);
    delay(1000);
    digitalWrite(4, LOW);
    delay(1000);
}

```

Μέσα στο πρόγραμμα έχουμε ορίσει το pin το οποίο έχουμε ορίσει ως data pin, δηλαδή το 4.

Στο Makefile αρχείο προσθέτουμε την εντολή

```
Relay: arduPi.o g++ -lrt -lpthread Relay.cpp arduPi.o -o Relay
```

και σώζουμε ως Makefile. Εκτελούμε το πρόγραμμα με την εντολή ./Relay και ακούμε το ρελέ να ενεργοποιείται, γεγονός που δείχνει ότι έχουμε ολοκληρώσει με επιτυχία τη σωστή σύνδεση του ρελέ.

Στη συνέχεια, εγκαθιστούμε στο Raspbian μια εφαρμογή με το όνομα Screen, εφαρμογή η οποία θα μας δώσει τη δυνατότητα να τρέχουμε πολλαπλά παραθύρα σε μια μόνο εφαρμογή του παραθύρου εντολών, τα οποία όμως θα μπορούν να παραμένουν ανοιχτά ακόμη και όταν κλείνουμε το παράθυρο εντολών. Αυτό είναι σημαντικό γιατί μας εγγυάται ότι μια εφαρμογή δε θα τερματιστεί αν εμείς κλείσουμε το παράθυρο εντολών.

Το Screen, όπως και όλες οι εφαρμογές που εγκαθιστούμε στο Raspbian, εγκαθιστάται μέσω της εντολής

```
sudo apt-get install screen
```

Εμείς, καθ'υπόδειξη του tutorial του screen, απλά αλλάξαμε κάποιες από τις ρυθμίσεις του ώστε να γίνει πιο εύχρηστο. Στη συνέχεια το αποθηκεύουμε με το όνομα .screenrc και εκελώντας την εντολή screen στο παράθυρο εντολών εμφανίζεται το παράθυρο της εφαρμογής.



Εικόνα 6: Παράθυρο του Screen.

Ονομάζουμε το παράθυρο Thermostat και συνεχίζουμε την εργασία μας.

Συνεχίζοντας, λοιπόν, εγκαταστήσαμε στο σύστημά μας τη βιβλιοθήκη ανάπτυξης cURL (client for URLs) έτσι ώστε να μπορούμε να συμπεριλαμβανούμε URLs στον κώδικά μας και να τα εκτελούμε. Αυτή είναι μια

σημαντική ιδιότητα αφού θα μπορούμε, για παράδειγμα, να συνδεόμαστε σε μια Python εφαρμογή από το script μας και να εκχωρούμε τιμές που είναι το αποτέλεσμα των μετρήσεων θερμοκρασίας που έχει εκτελέσει το σύστημά μας. Αυτή τη δυνατότητα θα τη χρειαστούμε πολύ στην επόμενη εφαρμογή οπότε δε θα επεκταθούμε παραπάνω σε αυτό το μέρος της εργασίας. Η εντολή με την οποία εγκαθιστούμε τη βιβλιοθήκη είναι η

```
sudo apt-get install libcurl4-openssl-dev
```

Αρχίζουμε να γράφουμε τον κώδικα του θερμοστάτη. Ονομάζουμε το πρόγραμμά μας thermostat.cpp και θυμόμαστε ότι σκοπός του είναι να ανοίγει και να κλείνει το ρελέ για συγκεκριμένες μεταβολές της θερμοκρασίας καθώς και η δημιουργία ενός URL για την αποθήκευση δεδομένων. Ο κώδικας έχει ως εξής:

```
#include "arduPi.h"

#include <math.h>

#include <stdio.h>

#include <curl/curl.h>

SerialPi Serial;

WirePi Wire;

SPIPi SPI;

//Οι tmes pou xreiazomai stin eksisosi Steinhart-Hart
//kais ton ypologismo tis antistasis
#define TENKRESISTOR 10000
#define BETA 4050
#define THERMISTOR 10000
#define ROOMTEMPK 298.15

//Ο αριθμος ton metriseon pout ha lifthoun oi oποιes
//tha xrisimopoiithoun gia na ypologistei enas MO
//etsi oste ta dedomena mas na einai akrivi
#define READINGS 7

// Relay Pin
#define RPIN 4

// Setpoint
#define SETPOINT 15.0

/*****ΤΟ
MEROS OPOU PREPEI NA ORISTOUN OLES OI *
*METAVLITES POU PERIEXONTAI STO PROGRAMMA MAS *
*****/

/*****
*KODIKAS ARDUINO *
* *****/
```



```

boolean running = false; // Apotelei deikti
leitourgias tou thermostatati

int main (){
    setup();
    while(1){
        loop();
    }
return (0);

void setup(void) {
    printf("Starting up thermostat \n");
    Wire.begin();
    pinMode(RPIN, OUTPUT);
}

void loop(void) {
    float avResistance;
    float resistance;
    int combinedReadings[READINGS];
    byte val0;
    byte val1;

    //Oi metavlites tis thermokrasias
    float kelvin;
    float fahrenheit;
    float celsius;
    int channelReading;
    float analogReadingArduino;

    //Oi metavlites tou cURL
    CURL *curlInst;
    CURLcode result;

/*****

    ADC mappings

Pin Address

0 0xDC
1 0x9C
2 0xCC
3 0x8C
4 0xAC
5 0xEC
6 0xBC
7 0xFC
*****/

Wire.beginTransmission(8);
Wire.write(byte(0xFC));
Wire.endTransmission();

for(int r=0; r<READINGS; r++){
    Wire.requestFrom(8, 2);
    val0 = Wire.read();

```

```

        val1 = Wire.read();
        channelReading = int(val0)*16 + int(val1>>4);
        analogReadingArduino = channelReading * 1023
/4095;
        combinedReadings[r] = analogReadingArduino;
    delay(100);
}

for (int r=0; r<READINGS; r++) {
    avResistance += combinedReadings[r];
}
avResistance /= READINGS;

avResistance = (1023 / avResistance) - 1;
avResistance = TENKRESISTOR / avResistance;
resistance = avResistance / THERMISTOR;

kelvin = log(resistance);
kelvin /= BETA;
kelvin += 1.0 / ROOMTEMPK;
kelvin = 1.0 / kelvin;
printf("\nTemperature in K ");
printf("%f \n",kelvin);

celsius = kelvin -= 273.15;
printf("Temperature in C ");
printf("%f \n",celsius);

fahrenheit = (celsius * 1.8) + 32;
printf("Temperature in F ");
printf("%f \n",fahrenheit);

if(celsius > SETPOINT && running == false) {
    printf("Switching fan on ");
    digitalWrite(RPIN,HIGH);
    running = true;
}
else {
    if(celsius < SETPOINT && running == true) {
        printf("Switching fan off ");
        digitalWrite(RPIN,LOW);
        running = false;
    }
}

//Klisi tis vasis dedomenon thermokrasion
curlInst = curl_easy_init();
if(curlInst) {
    char url[40];
    //I IP diefthinsi tou kathe Raspberry Pi
    sprintf(url,
"http://192.168.1.72/addtemperature?temperature= %f&r
oom=1", celsius);

    curl_easy_setopt(curlInst, CURLOPT_URL, url);

    result = curl_easy_perform(curlInst);

    //Se periptosi apotyxiastha prepei na mas

```

```

//emfanistoun ta lathi.
    if(result != CURLE_OK)
        fprintf(stderr, "curl_easy_perform()
failed: %s\n", curl_easy_strerror(result));
        curl_easy_cleanup(curlInst);
    }

    delay(3000);
}

```

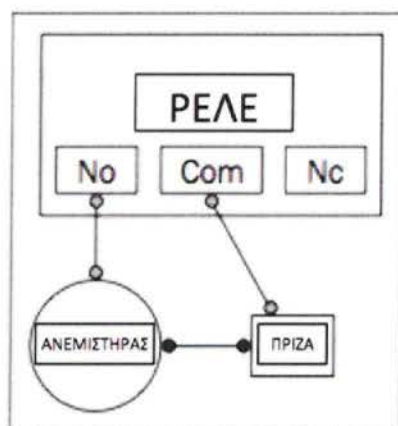
Στον κώδικα αυτό επιθυμούμε να επισημάνουμε τις μετατροπές που κάναμε ως προς την προηγούμενη εφαρμογή. Αρχικά, προσθέσαμε δύο μεταβλητές ακόμα, την RPIN και την SETPOINT, με την πρώτη να παίρνει την τιμή του rin στο οποίο είναι συνδεδεμένο το ρελέ και τη δεύτερη να αντιστοιχεί στη θερμοκρασία η οποία θα προκαλεί άνοιγμα και σβήσιμο του ανεμιστήρα. Στη συνέχεια, προσθέσαμε τις δύο μεταβλητές που αφορούν στο cURL, την curlInst και τη result. Η πρώτη από αυτές είναι αυτή η οποία αρχικοποιεί τα αντικείμενα του cURL ενώ η δεύτερη είναι ο χώρος που θα αποθηκεύονται τα αποτελέσματα όταν καλείται η cURL. Επίσης, τονίσαμε και παραπάνω ότι πλέον δουλεύουμε με το αναλογικό rin 7 -και όχι το μηδέν όπως στην προηγούμενη εφαρμογή. Ο κώδικας όπως έχουμε επανειλημμένα τονίσει αποσκοπεί στο άνοιγμα ή κλείσιμο του θερμοστάτη όταν η θερμοκρασία είναι αντίστοιχα μεγαλύτερη/ίση ή μικρότερη της θερμοκρασίας που έχει εκχωρηθεί στη μεταβλητή SETPOINT. Επιπροσθέτως, μπορεί ακόμα να μην έχουμε συνδέσει τον ανεμιστήρα αλλά ο κώδικας είναι ρυθμισμένος ώστε να ελέγχει αν ο ανεμιστήρας είναι σβηστός όταν η θερμοκρασία έχει ανέβει και ο θερμοστάτης έχει ανοίξει. Σε αυτή την περίπτωση ορίζουμε την RPIN (4) ως HIGH και τη running ως true. Με αυτό το τρόπο αλλάζουμε το διακόπτη του ρελέ και κατά συνέπεια ανάβει ο ανεμιστήρας. Στην περίπτωση που ο ανεμιστήρας λειτουργεί και η θερμοκρασία πέσει κάτω από αυτή που έχει εκχωρηθεί στο SETPOINT, το RPIN (4) παίρνει την τιμή LOW και το ρελέ κλείνει ανοίγοντας το κύκλωμα ενώ ταυτόχρονα η running θα πάρει την τιμή false. Τελικά, δημιουργούμε ένα URL που θα περιέχει όλα τα δεδομένα που λάβαμε κατά τις μετρήσεις. Έτσι, χρειαστήκαμε μια νέα μεταβλητή, την url στην οποία εκχωρείται η τιμή της HTTP διεύθυνσης που θα καλέσουμε. Αν εκτελώντας την URL προκύψει κάποιο λάθος εμφανίζεται ανάλογο μήνυμα. Ο κώδικας είναι πλήρης.

Μέσα από το Geany IDE τρέχουμε την εντολή

```
Thermo: arduPi.o g++ -lrt -lpthread -lcurl thermostat.cpp arduPi.o -o thermostat
```

και σώζουμε ως Makefile. Τώρα το μόνο που απομένει είναι να τεστάρουμε τον κώδικα συνδέοντας τον με μια M2M επικοινωνία με τον ανεμιστήρα.

Με σβηστό τον υπολογιστή Raspberry Pi και τον ανεμιστήρα αποσυνδεδεμένο από το ηλεκτρικό κύκλωμα, κόβουμε το καλώδιο του ανεμιστήρα και συνδέουμε το άκρο της πρίζας με το COM άκρο του ρελέ και το άκρο του ανεμιστήρα με το άκρο NO του ρελέ. Φροντίζουμε όλες οι συνδέσεις να είναι ασφαλείς ακι όπου έχουμε γυμνό καλώδιο μονώνουμε. Η σύνδεση είναι όπως φαίνεται στο σχήμα.



Εικόνα 7: Συνδεσμολογία ανεμιστήρα.

Αφού εξασφαλίσουμε την ασφαλεία των συνδέσεών μας ανοίγουμε την παροχή του ηλεκτρικού ρεύματος στον υπολογιστή, το ρελέ και τον ανεμιστήρα. Αφού φορτώσει το σύστημα του Raspberry Pi, ανοίγουμε ένα παράθυρο εντολών και τρέχουμε την εφαρμογή του θερμοστάτη με την εντολή `./thermostat`. Επειδή έχουμε εγκαταστήσει το λογισμικό Screen δεν έχουμε το φόβο να τερματιστεί η εφαρμογή μας κλείνοντας κάποιο από τα παράθυρα.

Έχουμε, λοιπόν, κατασκευάσει ένα αυτόματο σύστημα χειρισμού του ανεμιστήρα μέσω ελέγχου της θερμοκρασίας του περιβάλλοντος με μεταβολές που επικοινωνούνται μέσω M2M διαδικασιών. Αυτό που έμενε να κάνουμε ήταν να τεστάρουμε αν λειτουργεί σωστά. Για αρχή προσπαθούμε να μεταβάλουμε τη θερμοκρασία του αισθητήρα μας που είναι ο θερμίστορας. Αυτό το κάναμε με μια παγοκύστη την οποία φέραμε σχεδόν -αλλά όχι απόλυτα- σε επαφή με το θερμίστορα είτε με ένα σεσουάρ μαλλιών. Βλέποντας πως ο θερμοστάτης ανοίγει και κλείνει με τη ψύξη και τη θέρμανση επιβεβαιώνουμε πως το σύστημα λειτουργεί απόλυτα σωστά. Έχουμε, λοιπόν, στη διάθεσή μας έναν αυτόματο ανεμιστήρα οικιακής χρήσης.

6.4 ΚΑΤΑΣΚΕΥΗ ΜΙΑΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΑΠΟΘΗΚΕΥΣΗ ΤΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΜΕΤΡΗΣΗΣ ΤΗΣ ΘΕΡΜΟΚΡΑΣΙΑΣ

Κλείνοντας το Κεφάλαιο των εφαρμογών θα προσπαθήσουμε τη δημιουργία μιας βάσης δεδομένων στον υπολογιστή Raspberry Pi χρησιμοποιώντας SQLite. Αυτή η βάση δεδομένων θα χρησιμοποιηθεί για την αποθήκευση των δεδομένων που λαμβάνουμε κατά τις μετρήσεις των προηγούμενων εφαρμογών. Ταυτόχρονα, θα στήσουμε έναν Apache διαδικτυακό εξυπηρετητή (web server) που θα τρέχει Python μέσω WSGI που θα χρησιμοποιήσουμε για να τρέξουμε τις αναζητήσεις στη βάση δεδομένων μας.

6.4.1 ΔΗΜΙΟΥΡΓΙΑ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Το SQLite είναι μια τεχνολογία βάσεων δεδομένων γραμμένη σε C που ουσιαστικά διαχειρίζεται βάσεις δεδομένων υποστηρίζοντας όλες τις πλήρεις εκδόσεις της SQL. Έχει πολλές χρήσεις όπως είναι η δημιουργία βάσεων δεδομένων με σκοπό να ενσωματωθούν σε εφαρμογές όπως αυτές του διαδικτύου ή όπως αυτές που χρησιμοποιούνται σε ελαφριά συστήματα υπολογιστών όπως ο Raspberry Pi. Η HSQLite είναι πρακτική για μικρές εφαρμογές καθώς και για χρήστες που δεν επιθυμούν να επιβαρυνθούν από το κόστος κάποιας πιο πολύπλοκης βάσης δεδομένων.

Στο δικό μας σύστημα εγκαταστάθηκε η SQLite 3.x με την πλέον γνωστή μας εντολή

```
sudo apt-get install sqlite3
```

και με το πέρας της εγκατάστασης με την εντολή

```
cd /home/pi/ mkdir database cd database
```

δημιουργούμε ένα φάκελο στο οποίο θα αποθηκεύσουμε τη βάση δεδομένων μας κατά τη δημιουργία του. Όταν ολοκληρώσουμε τη δημιουργία της βάσης δεδομένων αυτή θα την αντιγράψουμε στον web server μας για να είναι προσπελάσιμη από το Apache.

Τρέχουμε την SQLite με την εντολή `sqlite3` ακολουθούμενη από το όνομα της βάσης δεδομένων με την κατάληξη `.db`. Στη δική μας περίπτωση η βάση δεδομένων θα ονομαστεί `temperature` οπότε θα εκτελούμε με την εντολή `sqlite3 temperature.db`. Με την εκτέλεση της εντολής αυτής ανοίγει ένα παράθυρο της SQLite. Στο παράθυρο αυτό γράφουμε εντολές οι οποίες αποσκοπούν στη δημιουργία πινάκων για την αποθήκευση δεδομένων. Οπότε, πριν δημιουργήσουμε τη βάση δεδομένων αναλογιζόμαστε τις στήλες και τις γραμμές που χρειαζόμαστε. Εμείς έχουμε ανάγκη από δύο απλούς πίνακες, οι οποίοι περιγράφονται παρακάτω.

Αρχικά, θα χρειαστούμε έναν πίνακα για την αποθήκευση των θερμοκρασιών που μετράμε μέσω της ασπίδας του Arduino και τον οποίο ονομάζουμε `temperature`. Χρειαζόμαστε τέσσερις στήλες, οι οποίες είναι οι εξής:

- Στήλη `Id`: Θα περιλαμβάνει τη μοναδική ταυτότητα για κάθε θερμοκρασία που θα εισάγεται στη βάση δεδομένων. Με κάθε νέα τιμή που θα εισάγεται στη στήλη αυτή θα αυξάνει αυτόματα και η διάστασή της.
- Στήλη `Roomid`: Αυτή είναι η στήλη που θα συνδέει κάθε μετρούμενη θερμοκρασία του πίνακα με το δωμάτιο στο οποίο έγινε η μέτρηση. Θα μπορούσαμε, δηλαδή, εδώ να αποθηκεύουμε για παράδειγμα το όνομα του δωματίου που έγινε η μέτρηση.
- Στήλη `Temperaturec`: Αποθηκεύονται οι θερμοκρασίες σε βαθμούς Κελσίου, όπως αυτή η τιμή υπολογίζεται από την ασπίδα του Arduino και αποστέλεται αυτόματα στη βάση δεδομένων.
- Στήλη `Datetime`: Θα υπολογίζουμε μια σφραγίδα της ώρας της ημέρας που

λαμβάνεται η μέτρηση. Έτσι, θα μπορούμε να υπολογίζουμε τις ώρες τις ημέρας για παράδειγμα που είναι πιο κρύες και υπάρχει μεγαλύτερη ανάγκη από θέρμανση.

Ο δεύτερος πίνακας θα είναι για την καταγραφή των ονομάτων των δεδομένων. Και αυτός ο πίνακας μπορεί να επεκταθεί σε περίπτωση που θα πρέπει να συμπεριληφθούν περισσότερα δωμάτια. Εδώ χρειαζόμαστε μόνο δύο στήλες οι οποίες θα είναι οι εξής:

- Στήλη Id: Όπως και πριν, θα περιλαμβάνει μοναδική ταυτότητα του κάθε δωματίου που καταχωρείται. Μπορούμε να μετονομάζουμε τα δωμάτια απλά αλλάζοντας τις τιμές των εισόδων της στήλης αυτής αντί να αλλάζουμε το όνομα του δωματίου δίπλα σε κάθε τιμή του πίνακα temperature.
- Στήλη Roomname: Εδώ απλά καταχωρούμε το όνομα του δωματίου.

Έχοντας ορίσει τους δύο αυτούς πίνακες προχωράμε στη γραφή της SQL. Στο παράθυρό της γράφουμε την εντολή

```
CREATE TABLE roomdetails (id INTEGER PRIMARY KEY AUTOINCREMENT, room VARCHAR(25));
```

η οποία αυτόματα δημιουργεί τον πίνακα roomdetails και του προσθέτει μια στήλη ID που παίρνει ακέραιες τιμές. Στη συνέχεια γράφουμε την εντολή

```
CREATE TABLE temperature (id INTEGER PRIMARY KEY AUTOINCREMENT, roomid INTEGER, FOREIGN KEY(roomid) REFERENCES roomdetails(id));
```

η οποία δημιουργεί τον πίνακα temperature που περιγράψαμε παραπάνω και παίρνει την πρώτη του στήλη από τον πίνακα roomdetails. Η δεύτερη στήλη, επίσης, παίρνει τιμές από τον roomdetails ενώ μπορούμε να προσθέσουμε τις άλλες δύο στήλες μέσω της εντολής

```
ALTER TABLE temperature ADD COLUMN temperaturec FLOAT(8);
```

Η στήλη αυτή περιλαμβάνει τιμές τύπου float μέχρι 8 χαρακτήρων. Τέλος, με την εντολή

```
ALTER TABLE temperature ADD COLUMN datetime DATETIME;
```

προσθέτουμε τη στήλη για τις σφραγίδες των ημερομηνιών, οι οποίες είναι της μορφής YYYY-MM-DD HH:MM:SS.

Σειρά είχε να προσθέσουμε στον roomdetails το δωμάτιο στο οποίο έχουμε εγκαταστήσει το σύστημα Raspberry Pi-θερμοστάτη. Αυτό το πραγματοποιούμε με την εντολή

```
INSERT INTO roomdetails (room) VALUES ('Office');
```

Σε αυτό το δωμάτιο θα αποθηκεύσουμε τις μετρήσεις που θα πάρουμε μέσω της ασπίδας του Arduino. Στον κώδικα είχαμε ορίσει ως παράμετρο ταυτότητας του

δωματίου τον αριθμό 1 οπότε όλες οι μετρήσεις θα καταγράφονται σε αυτό το δωμάτιο.

6.4.2 ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ WEB SERVER

Ο web server Apache είναι ελεύθερου κώδικα και μια από τις καλύτερες εναλλακτικές των επί χρήμασι εμπορεύσιμων προϊόντων της αγοράς. Εμείς χρησιμοποιήσαμε τον Apache 2.x για να φιλοξενήσουμε εφαρμογές στο οικιακό μας δίκτυο. Αυτές οι εφαρμογές μπορούν να χρησιμοποιηθούν για την καταγραφή δεδομένων στη βάση δεδομένων αλλά και για την παροχή μιας διεπαφής διαδικτύου με τα δεδομένα που καταγράψανε οι αισθητήρες μας.

Στο παράθυρο του SQLite εκτελούμε την εντολή

```
sudo apt-get install apache2
```

εγκαθιστώντας το Apache 2.x. Η διεύθυνση IP που εκχωρείται αυτόματα στον server μας είναι η 127.0.1.1. Η χρήση του Apache δεν είναι αντικείμενο αυτής της εργασίας οπότε δε θα συζητηθεί. Ωστόσο, ο Apache τρέχει μέσω του www-data το οποίο εγκαθιστούμε με την εντολή

```
sudo addgroup www-dataandsudo usermod -a -G www-data www-data
```

Επανεκκινούμε τον server μας και μπορούμε να περιηγηθούμε είτε χρησιμοποιώντας την IP του Raspberry Pi ή την 127.0.1.1. Ωστόσο, εκτός από περιήγηση χρειαζόμαστε να συνδεόμαστε στη βάση δεδομένων μας και να διαβάζουμε και να γράφουμε δεδομένα. Αυτό γίνεται αν συμπεριλάβουμε τη WSGI.

Η WSGI είναι μια βασισμένη στην Python διεπαφή η οποία επιτρέπει την επικοινωνία χρήστη με εφαρμογή Python. Την εγκαθιστούμε με την εντολή

```
sudo apt-get install libapache2-mod-wsgi
```

με την εκτέλεση της οποίας έχουμε τον πλήρη απαραίτητο κώδικα που χρειάζεται για να υποστηριχθούν εφαρμογές διαδικτύου της Python. Εκτελούμε διαδοχικά τις εντολές

```
cd /var/www/
```

```
sudo mkdir wsgi-scripts
```

```
/etc/apache2/sites-available
```

έτσι ώστε διαδοχικά να κατευθυνθούμε στο φάκελο όπου βρίσκεται το WSGI και προσθέτουμε το φάκελο wsgi-scripts για να αποθηκεύουμε τα scripts του WSGI. Τέλος κατευθυνόμαστε στον φάκελο sites-available. Εκεί ανοίγουμε το φάκελο default. Εκεί περιλαμβάνονται τα αρχεία του διαδικτυακού τύπου μας. Στο κάτω μέρος της σελίδας βρίσκουμε τη γραμμή

```
WSGIScriptAlias /addtemperature /var/www/wsgi-scripts/addtemperature.wsgi
```

η οποία καθιστά διαθέσιμο το script `addtemperature.wsgi` μέσω του περιηγητή με την URL `/addtemperature`. Εκεί προσθέτουμε τον κώδικα

```
<Directory /var/www/wsgi-scripts>
    Order allow,deny
    Allow from all </Directory>
```

Ο οποίος προσδιορίζει μια τοποθεσία `/var/www/wsgi-scripts` και δίνει την εντολή στο Apache να δέχεται την κίνηση σε αυτό. Αποθηκεύουμε και οδηγούμαστε και πάλι στο φάκελο του `apache2` ως εξής

```
cd /etc/apache2
```

Επανεκκινούμε τον Apache. Πριν γράψουμε τον κώδικα αντιγράφουμε την – άδεια ακόμη- βάση δεδομένων στο διαδικτυακό φάκελο. Εκτελούμε διαδοχικά τις εντολές

```
cd /var/www
```

```
sudo mkdir database cd database
```

```
cp /home/pi/database/temperature.db
```

και έχουμε το Apache έτοιμο να καταχωρηθούν τα δεδομένα μας, εργασία η οποία θα γίνει ευθύς αμέσως.

6.4.3 ΣΥΓΓΡΑΦΗ ΤΟΥ ΚΩΔΙΚΑ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Για τη δουλειά αυτή χρησιμοποιούμε τη γλώσσα προγραμματισμού Python και τη WSGI. Μπαίνουμε στον κατάλληλο φάκελο με την εντολή

```
cd /var/www/wsgi-scripts
```

και δημιουργούμε το αρχείο `addtemperature.wsgi`.

Ο κώδικάς μας υπάρχει έτοιμος στο διαδίκτυο και είναι ο εξής (έχουμε κάνει μετατροπές που αφορούν ονοματοδοσίες κτλ):

```
import sqlite3
from cgi import escape, parse_qs

def application(environ, start_response):
    connection = None
    my_response = ""
    params = parse_qs(environ['QUERY_STRING'])
    room = escape(params.get('room', [''])[0])
    temperature =
    escape(params.get('temperature', [''])[0])
```



```

        my_query = 'INSERT INTO
temperature(roomid, temperaturef, datetime)
VALUES(%s,%s,CURRENT_TIMESTAMP);' %(room, temperature
)
        try:
            connection =
sqlite3.connect('/var/www/database/temperature.db' ,
isolation_level=None)
            cursor = connection.cursor()
            cursor.execute(my_query)
            query_results = cursor.fetchone()
            my_response = 'Inserted %s for
room %s' % (temperature, room) except sqlite3.Error,
e:
            my_response = "There is an error %s:" %
(e)
        finally:
            connection.close()

            status = '200 OK'
            response_headers = [('Content-Type',
'text/plain'),
('Content-Length', str(len(my_response)))]
            start_response(status,
response_headers)
            return [my_response]

```

Αυτός ο κώδικας είναι αρκετός. Έχουμε πλέον στα χέρια μας ένα αυτόματο σύστημα χειρισμού του ανεμιστήρα μας με τα δεδομένα να αποθηκεύονται σε βάση δεδομένων.

ΕΠΙΛΟΓΟΣ

Στην εργασία αυτή συζητήσαμε το ζήτημα των M2M επικοινωνιών υπό το πρίσμα του οικιακού αυτοματισμού. Οι M2M επικοινωνίες είναι ένα σπουδαίο τεχνολογικό ζήτημα της εποχής μας αφού προσφέρουν ένα άλλο επίπεδο αυτοματοποιημένων διαδικασιών που διευκολύνουν κάθε τομέα της ανθρώπινης δραστηριότητας. Ένα από απιο εμπορικά προϊόντα αυτοματοποιημένων M2M επικοινωνιών είναι αυτό του αυτοματισμού οικιών και γι'αυτό κάναμε μια προσπάθεια να εργαστούμε σε εφαρμογές τέτοιου τύπου.

Για αρχή κατασκευάσαμε ένα απλό θερμόμετρο για να μπορέσουμε να γνωρίσουμε καλά τα δομικά μέρη της συσκευής μας. Γράψαμε τον κώδικα λειτουργίας του σε περιβάλλον Geany IDE και γλώσσα Arduino/C++. Αφού ελέγξαμε τη λειτουργία του προχωρήσαμε ένα σκαλί παραπάνω χρησιμοποιώντας το θερμόμετρο αυτό ως δομικό μέρος για την κατασκευή ενός θερμοστάτη, τόσο υλισμικά όσο και λογισμικά. Τέλος, κατασκευάστηκε μια στοιχειώδης βάση δεδομένων για την αποθήκευση των δεδομένων που λαμβάνει ο αισθητήρας μας.

Το σύστημα αυτό κατασκευάστηκε με χρήση του καινοτομικού και οικονομικού υπολογιστή Raspberry Pi και μιας ασπίδας Arduino.

Τα αποτελέσματα της εργασίας αποδικνύουν την ευχρηστη φύση του Raspberry Pi και του Arduino καθώς και την ευκολία με την οποία νέες επιχειρήσεις μικρού μεγέθους αλλά και ερασιτέχνες αυτοματιστές μπορούν να αυτοματοποιήσουν τις οικιακές λειτουργίες.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Boswarth D. et al., M2M Communications - A System Approach, Wiley 2012
2. Hersent O., Boswarthick D., Elloumi O., The Internet of Things- Applications to the Smart Grid and Building Automation, Wiley 2012
3. M2M and Smart Systems, Harbor Research Inc.
4. Gertz E. & Di Justo P., Environmental Monitoring With Arduino, O' Reilly 2012
5. Warner T, Hacking Raspberry Pi, Que 2014
6. Wilcher D., Learn Electronics With Arduino, APress 2012
7. Riley M., Programming Your Home, The Pragmatic Programmers 2012
8. Upton E. & Halfacree, Raspberry Pi User Guide, Wiley 2012