

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ



Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Σύστημα Επικοινωνίας Μικροελεγκτή - Κινητής Συσκευής Δια
Μέσου Εξυπηρετητή Για Απομακρυσμένο Έλεγχο

Microcontroller - Mobile Device Communication System Through
Server For Remote Control

Διπλωματική Εργασία

Αθανάσιος Α. Λιάπης

Επιβλέποντες Καθηγητές

Αλκιβιάδης Ακρίτας

Καθηγητής ΠΘ

Γεώργιος Σταμούλης

Καθηγητής ΠΘ

Βόλος, Μάρτιος 2015

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

Σύστημα Επικοινωνίας Μικροελεγκτή - Κινητής Συσκευής Δια
Μέσου Εξυπηρετητή Για Απομακρυσμένο Έλεγχο

Microcontroller - Mobile Device Communication System Through
Server For Remote Control

Διπλωματική Εργασία
Αθανάσιος Α. Λιάπης

Επιβλέποντες Καθηγητές
Αλκιβιάδης Ακρίτας
Καθηγητής ΠΘ

Γεώργιος Σταμούλης
Καθηγητής ΠΘ

Εγκρίθηκε από την διμελή εξεταστική επιτροπή την 6^η Μαρτίου 2015

.....

.....

Αλκιβιάδης Ακρίτας

Γεώργιος Σταμούλης

Καθηγητής ΠΘ

Καθηγητής ΠΘ

(Υπογραφή)

.....

Αθανάσιος Α. Λιάπης

Διπλωματούχος Μηχανικός Ηλεκτρονικών Υπολογιστών, Τηλεπικοινωνιών και Δικτύων του
Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστημίου
Θεσσαλίας

Ευχαριστίες

Στο σημείο αυτό της ολοκλήρωσης των προπτυχιακών μου σπουδών, θα ήθελα να ευχαριστήσω την οικογένειά μου για την ηθική και οικονομική υποστήριξή τους, απαραίτητες προϋποθέσεις για την έως τώρα πορεία μου.

Τους διδάσκοντες του τμήματος για την μικρή ή μεγάλη συμβολή στα εφόδια που απαιτήθηκαν ώστε να ολοκληρώσω τις σπουδές μου, αλλά κυρίως να νοιώσω την απαραίτητη αυτοπεποίθηση στον εργασιακό χώρο επαγγελματικά. Κυρίως όσο αφορά τη συγκεκριμένη εργασία, τον κ. Ακρίτα για την εμπιστοσύνη του στο πρόσωπό μου και την συνεργασία του.

Για δεύτερη φορά τον αδερφό μου, ο οποίος είχε μεγάλη συμβολή στο ηλεκτρολογικό κομμάτι της εφαρμογής αυτής και στις ιδέες που την εμπλούτισαν σε μεγάλο βαθμό. Επίσης αισθάνομαι την υποχρέωση να ευχαριστήσω τον Βασίλη για την αμέριστη συμβολή του.

Περιεχόμενα

Ευχαριστίες.....	6
Περίληψη.....	9
Η Πλευρά του Web Application.....	10
Το World Wide Web.....	10
Uniform Resource Identifiers.....	10
Το πρωτόκολλο Hypertext Transfer Protocol.....	11
Χειρισμός Συνδέσεων HTTP.....	13
Κατατμημένη κωδικοποίηση μεταφοράς (chunked transfer encoding) μέσω HTTP.....	13
Μορφές Web.....	13
Hypertext Markup Language.....	14
Web Applications.....	14
Web Services.....	14
Γλώσσες – Τεχνολογίες του Web.....	15
Η γλώσσα προγραμματισμού PHP.....	15
MySQL.....	15
Apache HTTP.....	16
LAMP.....	17
Η Πλευρά του Μικροελεγκτή.....	18
Μικροελεγκτές Γενικά.....	18
Arduino Uno – AVR ATmega328p.....	19
Βασικά χαρακτηριστικά του Arduino Uno.....	20
Ιδιότητες των pins που ορίζονται σαν είσοδοι (INPUT).....	21
Ιδιότητες των pins που ορίζονται σαν έξοδοι (OUTPUT).....	21
Αναλογικά pins (Analog Input Pins).....	21
Διαμόρφωση Πλάτους Παλμών (PWM).....	21
Γλώσσα.....	22
Γιατί επέλεξα το Arduino.....	22
Διαδικασία ανάπτυξης.....	24
Υλικό μέρος.....	24
Απαραίτητα στοιχεία.....	24
Προγραμματιστικό μέρος Arduino – Web Application.....	29
Web Application.....	30
Προγραμματισμός Arduino.....	34

Διαδικασία Εγκατάστασης.....	39
Προτάσεις Βελτίωσης.....	41
Συμπεράσματα	43
Πηγές.....	44

Περίληψη

Η παρούσα διπλωματική εργασία, έχει σκοπό να περιγράψει την υλοποίηση μια εφαρμογής αυτοματισμού χειριζόμενης από οποιαδήποτε συσκευή διαθέτει σύνδεση στο διαδίκτυο και κάποιον φυλλομετρητή. Να εντοπίσει επίσης τις δυσκολίες που προκύπτουν και να δώσει λύση σε αυτές. Κάποιες από τις εγγενείς λειτουργίες μια τέτοιας εφαρμογής σε οικιακό περιβάλλον, είναι που μου ενέπνευσαν και τον τίτλο του ίδιου το πονήματος. Το πώς θα διαμεσολαβήσει ένας απομακρυσμένος εξυπηρετητής, ανάμεσα σε μια συσκευή ελέγχου και την συσκευή προς έλεγχο, έτσι ώστε να καταστήσει δυνατή την επικοινωνία μεταξύ τους.

Αναλυτικά, χωρίζω το κείμενο αυτό σε τμήμα υλοποίησης της εφαρμογής, από άποψη υλικού και από άποψη λογισμικού, στο τμήμα της εγκατάστασής του και στο τμήμα των προτάσεων μου για μελλοντική ανάπτυξη του κατασκευάσματος αυτού. Οι προτάσεις αφορούν είτε βελτίωση του υπάρχοντος είτε προσθήκη επιπλέον λειτουργιών που θα δώσουν άλλη διάσταση σε κάτι φαινομενικά μόνο αντίστοιχο.

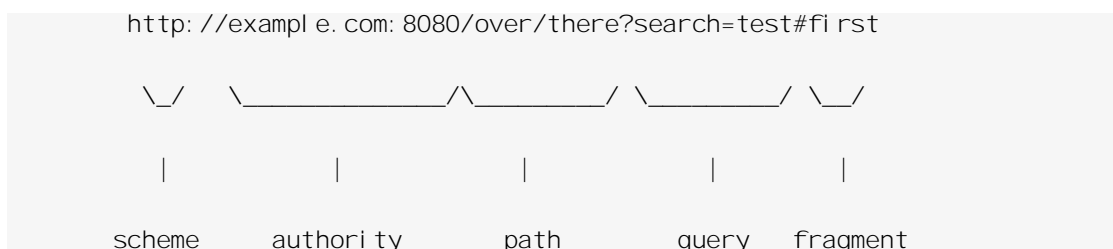
Η Πλευρά του Web Application

To World Wide Web

Το Διαδίκτυο και το WWW έχει γίνει η ραχοκοκαλιά της σύγχρονης κοινωνίας και της οικονομίας μας. Το WWW είναι ένα κατακευματισμένο σύστημα αλληλένδετων πόρων στο διαδίκτυο. Βασίζεται σε ένα σύνολο διαφορετικών, συσχετιζόμενων τεχνολογιών. Θα ρίξουμε μια σύντομη ματιά στα πιο σημαντικά πρωτόκολλα και μορφές σε αυτήν την ενότητα, συμπεριλαμβανομένων των Uniform Resource Identifiers (URI), HTTP και HTML.

Uniform Resource Identifiers

Τα URI, που καθορίζονται επί του παρόντος στο RFC 3986 [BL05], είναι strings που χρησιμοποιούνται για την κλήση κάποιων πόρων. Σε όρους κατακευματισμένων συστημάτων, ένα URI έχει τρεις διακριτούς ρόλους, την ονοματοδοσία, την διευθυνσιοδότηση και τον εντοπισμό των πόρων. Θα επικεντρωθούμε στην ταυτοποίηση των πόρων από αυτά στο WWW, αν και μπορούν να χρησιμοποιηθούν για άλλες αφηρημένες ή φυσικές οντότητες επίσης. Σύμφωνα με τις προδιαγραφές, ένα URI αποτελείται από πέντε μέρη: scheme, authority, path, query και fragment. Ωστόσο, μόνο τα scheme και path είναι υποχρεωτικά, ενώ τα υπόλοιπα τμήματα είναι προαιρετικά. Το scheme δηλώνει το είδος του URI και έτσι καθορίζει την έννοια των άλλων τμημάτων του. Αν χρησιμοποιηθεί, το authority καταδεικνύει την αρμόδια αρχή για τον αναφερόμενο πόρο. Στην περίπτωση του HTTP ως scheme, το τμήμα αυτό καθίσταται υποχρεωτικό και περιέχει τον host του web server που φιλοξενεί τον πόρο. Μπορεί προαιρετικά να περιέχει έναν αριθμό θύρας (η 80 υπονοείται για http) και τα στοιχεία ταυτότητα. Ο υποχρεωτικός τομέας path χρησιμοποιείται για την διευθυνσιοδότηση τον προσιτό από το scheme πόρο. Το προαιρετικό query παρέχει μη-ιεραρχημένα δεδομένα ως μέρος του αναγνωριστικού πόρου. Τα fragments μπορούν να χρησιμοποιηθούν για να υποδείξουν ένα συγκεκριμένο τμήμα εντός του πόρου. Το ακόλουθο παράδειγμα έχει προσαρμοστεί από το RFC 3986 [BL05] και κάνει χρήση των πέντε μερών:



Προσδιορίζει έναν πόρο του ιστοχώρου (scheme: http), που φιλοξενείται στο example.com (στη θύρα 8080). Η διαδρομή των πόρων είναι η /over/there και το query περιέχει το search=test. Επιπλέον, το fragment περιέχει την τιμή first.

Το πρωτόκολλο Hypertext Transfer Protocol

Το HTTP είναι ένα πρωτόκολλο επιπέδου εφαρμογής που αντιπροσωπεύει το θεμέλιο της επικοινωνίας για το WWW πάνω από το TCP / IP. Το HTTP, όπως ορίζεται στο RFC 2616, είναι ένα πρωτόκολλο σύμφωνο με την αρχιτεκτονική client/server και ένα μοντέλο επικοινωνίας αίτησης/απάντησης. Οι διακομιστές φιλοξενούν πόρους που προσδιορίζονται από κάποιο URI και προσπελαύνονται έτσι από πελάτες. Ο πελάτης εκδίδει μια αίτηση HTTP στον server που σε αντάλλαγμα παρέχει μια απόκριση HTTP. Το μοντέλο επικοινωνίας περιορίζει τις πιθανές μορφές μηνυμάτων σε αλληλουχίες αιτήσεων/απαντήσεων που ξεκινούν πάντα από τον πελάτη. Εκτός από τους πελάτες και τους διακομιστές, το HTTP περιγράφει επίσης μεσάζοντες, όπως οι proxy. Τα στοιχεία αυτά παρέχουν πρόσθετες δυνατότητες, όπως προσωρινή αποθήκευση ή φιλτράρισμα. Οι proxies συνδυάζουν τα χαρακτηριστικά πελάτων και εξυπηρετητών και επομένως είναι αφανείς για τους πελάτες και τους διακομιστές από άποψη επικοινωνίας.

Οι αιτήσεις και οι απαντήσεις HTTP έχουν μια κοινή δομή. Και οι δύο ξεκινούν με μια γραμμή αιτήματος. Το επόμενο μέρος περιέχει ένα σύνολο γραμμών κεφαλίδας που περιλαμβάνουν πληροφορίες σχετικά με την αίτηση (και αντίστοιχα την απάντηση) και για την οντότητα/περιεχόμενο.

Η οντότητα είναι προαιρετικά το σώμα ενός μηνύματος HTTP που περιέχει ωφέλιμη πληροφορία, όπως μια αναπαράσταση του πόρου. Ενώ τα δύο πρώτα μέρη ενός μηνύματος HTTP βασίζονται σε κείμενο, η οντότητα μπορεί να είναι οποιοδήποτε σύνολο από bytes. Η γραμμές αιτημάτων HTTP περιέχουν ένα URI και μια μέθοδο. Υπάρχουν διάφορες μέθοδοι HTTP που παρέχουν ερμηνεύονται διαφορετικά όταν μεταβιβάζονται σε έναν πόρο, όπως φαίνεται στον πίνακα:

Method	Usage	Safe	Idempotent	Cachable
GET	This is the most common method of the WWW. It is used for fetching resource representations.	✓	✓	(✓)
HEAD	Essentially, this method is the same as GET, however the entity is omitted in the response.	✓	✓	(✓)
PUT	This method is used for creating or updating existing resources with new representations.		✓	
DELETE	Existing resources can be removed with DELETE		✓	
POST	POST is used to create new resources. Due to its lack of idempotence and safety, it also often used to trigger arbitrary actions.			
OPTIONS	Method provides meta data about a resource and available representations.	✓	✓	

Μια απλή συναλλαγή αίτησης/απάντησης:

```
GET /html/rfc1945 HTTP/1.1
Host: tools.ietf.org

User-Agent: Mozilla/5.0 (Ubuntu; X11; Linux x86_64; rv:9.0.1) Gecko/20100101
Firefox/9.0.1

Accept: text/html, application/xhtml+xml, application/xml; q=0.9, */*; q=0.8

Accept-Language: de-de, de; q=0.8, en-us; q=0.5, en; q=0.3

Accept-Encoding: gzip, deflate

Accept-Charset: ISO-8859-1, utf-8; q=0.7, *; q=0.7

Connection: keep-alive

If-Modified-Since: Sun, 13 Nov 2011 21:13:51 GMT

If-None-Match: "182a7f0-2aac0-4b1a43b17a1c0; 4b6bc4bba3192"

Cache-Control: max-age=0

HTTP/1.1 304 Not Modified

Date: Tue, 17 Jan 2012 17:02:44 GMT

Server: Apache/2.2.21 (Debian)

Connection: Keep-Alive

Keep-Alive: timeout=5, max=99

Etag: "182a7f0-2aac0-4b1a43b17a1c0; 4b6bc4bba3192"

Content-Location: rfc1945.html

Vary: negotiate, Accept-Encoding
```

Χειρισμός Συνδέσεων HTTP

Όπως έχει ήδη αναφερθεί, το HTTP χρησιμοποιεί το πρωτόκολλο TCP / IP ως πρωτόκολλο μεταφοράς. Θα εξετάσουμε τώρα την ακριβή χρήση των TCP sockets για αιτήσεις HTTP. Οι προηγούμενες προδιαγραφές του HTTP είχαν προτείνει μια ξεχωριστή υποδοχή σύνδεσης για κάθε κύκλο αίτησης / απάντησης. Προσθέτοντας την επιβάρυνση της δημιουργίας μιας σύνδεσης TCP για κάθε αίτημα οδηγούμαστε σε κακή απόδοση και απουσία επαναχρησιμοποίησης των υφιστάμενων συνδέσεων. Η μη-τυπική κεφαλίδα Connection: Keep-Alive ήταν μια προσωρινή λύση, αλλά η τρέχουσα προδιαγραφή HTTP 1.1 ασχολήθηκε με αυτό το θέμα λεπτομερέστερα. Το HTTP 1.1 εισήγαγε τις μόνιμες συνδέσεις ως προεπιλογή. Δηλαδή, η βασική σύνδεση TCP ενός αιτήματος HTTP επαναχρησιμοποιείται για τις επόμενες αιτήσεις HTTP. Η τοποθέτηση αιτήσεων σε ουρά βελτιώνει περαιτέρω την απόδοση των μόνιμων συνδέσεων, επιτρέποντας την αποστολή πολλαπλών αιτήσεων, χωρίς να αναμένονται οι απαντήσεις των προηγούμενων. Ο διακομιστής ανταποκρίνεται στη συνέχεια σε όλα τα εισερχόμενα αιτήματα με τη ίδια σειρά. Και οι δύο μηχανισμοί έχουν βελτιώσει τις επιδόσεις και μείωσαν τα προβλήματα καθυστέρησης των web applications. Αλλά η διαχείριση πολλαπλών ανοιχτών συνδέσεων και η επεξεργασία των αιτήσεων διοχέτευσης αποκάλυψε νέες προκλήσεις για τους web servers.

Κατατμημένη κωδικοποίηση μεταφοράς (chunked transfer encoding) μέσω HTTP

Ένα μήνυμα HTTP πρέπει να περιέχει το μήκος του περιεχομένου του, αν υπάρχει αυτό. Στο HTTP 1.1, αυτό είναι απαραίτητο για τον καθορισμό του συνολικού μήκους του μηνύματος και τον διαχωρισμό του επόμενου μηνύματος της σύνδεσης. Μερικές φορές, το ακριβές μήκος ενός φορέα δεν μπορεί να προσδιοριστεί εκ των προτέρων. Αυτό είναι ιδιαίτερα σημαντικό για το περιεχόμενο που δημιουργείται δυναμικά ή για τις οντότητες που έχουν συμπεστεί on-the-fly. Ως εκ τούτου, το HTTP παρέχει εναλλακτικές κωδικοποιήσεις μεταφοράς. Όταν χρησιμοποιείται κατατμημένη κωδικοποίηση μεταφοράς, ο πελάτης ή ο διακομιστής στέλνει κομμάτια αυτής διαδοχικά. Το μήκος του επόμενου αναμενόμενου κομματιού προφτάνει του πραγματικού κομματιού. Ένα μηδενικό μήκος κομματιού προδίδει το τέλος του περιεχομένου. Ο μηχανισμός αυτός επιτρέπει τη μεταφορά των παραγόμενων περιεχομένων με αυθαίρετο μήκος.

Μορφές Web

Το HTTP δεν περιορίζει τις μορφές εγγράφων που χρησιμοποιούνται ως περιεχόμενο. Ωστόσο, η βασική ιδέα του WWW βασίζεται στα υπερμέσα, έτσι οι περισσότερες από τις μορφές είναι συμβατές με αυτά. Το πιο σημαντικό είναι η HTML και οι απόγονοί της.

Hypertext Markup Language

Η HTML είναι μια γλώσσα σήμανσης που προέρχεται από την SGML [ISO86] και επηρεάζεται από την XML. Η HTML παρέχει ένα σύνολο στοιχείων, ιδιοτήτων και κανόνων για την περιγραφή των ιστοσελίδων με λέξεις. Ένα πρόγραμμα περιήγησης αναλύει το HTML έγγραφο, με τη χρήση της διαρθρωτικής σημασιολογίας του για να αποδώσει μια οπτική αναπαράσταση για τους ανθρώπους. Η HTML υποστηρίζει υπερμέσα μέσω δεσμών (hyperlinks) και διαδραστικών μορφών. Επίσης, αντικείμενα πολυμέσων όπως εικόνες μπορούν να χρησιμοποιηθούν σε ένα έγγραφο HTML. Η εμφάνιση και το στυλ ενός εγγράφου HTML μπορεί να προσαρμοστεί με τη χρήση CSS. Για περισσότερες δυναμικές διεπαφές χρήστη και διαδραστική συμπεριφορά, τα έγγραφα HTML μπορούν να εμπλουτιστούν με ενσωματωμένο κώδικα γλωσσών scripting, όπως η JavaScript. Για παράδειγμα, μπορεί να χρησιμοποιηθεί για να φορτώσει προγραμματισμένα περιεχόμενα στο παρασκήνιο, χωρίς πλήρη επαναφόρτωση της σελίδας. Η τεχνική αυτή, γνωστή επίσης και ως AJAX, υπήρξε ένα από τα κλειδιά για πιο responsive διεπαφές χρήστη. Κατά συνέπεια, επιτρέπει σε web applications να μοιάζουν με διεπαφές παραδοσιακών desktop εφαρμογών.

Web Applications

Υπάρχει ένας τεράστιος αριθμός από διαφορετικές εφαρμογές διαθέσιμες στο διαδίκτυο. Στη συνέχεια, διακρίνουμε δύο διαφορετικούς τύπους των web Applications – τα web sites (ιστοσελίδες) και τα web services. Οι ιστοσελίδες είναι web-based εφαρμογές που έχουν σχεδιαστεί για τον άνθρωπο και την πρόσβαση μέσω browser. Αντίθετα, οι υπηρεσίες Ιστού είναι διεπαφές web-based για επικοινωνία μηχανής με μηχανή. Αν και αυτή η διάκριση είναι μάλλον αυθαίρετη, αυτό θα μας βοηθήσει να εντοπίσουμε τα διαφορετικά χαρακτηριστικά αλλά και τις απαιτήσεις τους.

Web Services

Τα Web Services παρέχουν πρόσβαση σε υπηρεσίες εφαρμογών μέσω HTTP. Έτσι, οι υπηρεσίες Ιστού, συχνά μοιάζουν με παραδοσιακούς μηχανισμούς για τα καταναμημένα υπολογιστικά συστήματα όπως το RPC ή τη μεταβίβαση μηνυμάτων, αν και βασίζονται σε τεχνολογίες web.

Αντίθετα με τις ιστοσελίδες, τα Web Services δεν στοχεύουν στην άμεση (ανθρώπινη) πρόσβαση των χρηστών ως κύριο μέλημα. Αντ' αυτού, τα Web Services επιτρέπουν τη μηχανής-προς-μηχανή επικοινωνία και παρέχουν λειτουργίες της εφαρμογής μέσω μιας διασύνδεσης και δομημένων μηνύματων. Αρκετά Web Services, προσφέρουν τόσο μια ιστοσελίδα όσο και μια διεπαφή Web Service (API). Αν και η ιστοσελίδα χρησιμοποιείται για την πρόσβαση μέσω browser, το Web Service μπορεί να χρησιμοποιηθεί για εξειδικευμένες εφαρμογές, όπως mobile εφαρμογές πελάτη ή κρυπτογραφημένες αλληλεπιδράσεις των services.

Γλώσσες – Τεχνολογίες του Web

Η γλώσσα προγραμματισμού PHP

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

Ένα αρχείο με κώδικα PHP θα πρέπει να έχει την κατάλληλη επέκταση (π.χ. *.php, *.php4, *.html κ.ά.). Η ενσωμάτωση κώδικα σε ένα αρχείο επέκτασης .html δεν θα λειτουργήσει και θα εμφανίσει στον browser τον κώδικα χωρίς καμία επεξεργασία, εκτός αν έχει γίνει η κατάλληλη ρύθμιση στα MIME types του server. Επίσης ακόμη κι όταν ένα αρχείο έχει την επέκταση .php, θα πρέπει ο server να είναι ρυθμισμένος για να επεξεργάζεται και να μεταγλωτίζει τον κώδικα PHP σε HTML που καταλαβαίνει το πρόγραμμα πελάτη. Ο διακομιστής Apache, που χρησιμοποιείται σήμερα ευρέως σε συστήματα με τα λειτουργικά συστήματα GNU/Linux, Microsoft Windows, Mac OS X υποστηρίζει εξ ορισμού την εκτέλεση κώδικα PHP, είτε με την χρήση ενός πρόσθετου (mod_php) ή με την αποστολή του κώδικα προς εκτέλεση σε εξωτερική διεργασία CGI ή FCGI ή με την έλευση της php5.4 υποστηρίζονται η εκτέλεση σε πολυάσχολους ιστοχώρους, FastCGI Process Manager (FPM).

MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Έλαβε το όνομά της από την κόρη του Μόντυ Βιντένιους, τη Μάι. Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων.

Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιόκτητων συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle.

Η MySQL είναι δημοφιλής βάση δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες. Χρησιμοποιείται σε κάποιες από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες, όπως το Flickr, το YouTube, η Wikipedia, το Google, το Facebook και το Twitter.

Με τον όρο βάση δεδομένων (database) εννοείται μία συλλογή από συστηματικά μορφοποιημένα σχετιζόμενα δεδομένα στα οποία είναι δυνατή η ανάκτηση δεδομένων μέσω αναζήτησης κατ' απαίτηση. Ο Αμερικανός επιστήμονας υπολογιστών Τζιμ Γκρέϊ (Jim Gray) έχει γράψει για τις βάσεις δεδομένων: «Όταν οι άνθρωποι χρησιμοποιούν τις λέξεις βάση δεδομένων, διατυπώνουν στην ουσία ότι τα δεδομένα πρέπει να αυτοπροσδιορίζονται και να έχουν μια σχηματική δομή. Αυτό ακριβώς περιγράφουν οι λέξεις βάση δεδομένων».

Ειδικότερα, στην επιστήμη της πληροφορικής και στην καθημερινή χρήση των ηλεκτρονικών υπολογιστών, με τον όρο βάσεις δεδομένων αναφερόμαστε σε οργανωμένες, διακριτές συλλογές σχετιζόμενων δεδομένων ηλεκτρονικά και ψηφιακά αποθηκευμένων, στο λογισμικό που χειρίζεται τέτοιες συλλογές (Σύστημα Διαχείρισης Βάσεων Δεδομένων, ή DBMS) και στο γνωστικό πεδίο που το μελετά. Πέρα από την εγγενή της ικανότητα να αποθηκεύει δεδομένα, η βάση δεδομένων παρέχει μέσω του σχεδιασμού και του τρόπου ιεράρχησης των δεδομένων, τα αποκαλούμενα συστήματα διαχείρισης περιεχομένου, δηλαδή τη δυνατότητα γρήγορης άντλησης και ανανέωσης των δεδομένων.

Apache HTTP

Ο Apache HTTP γνωστός και απλά σαν Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού (web). Όταν ένας χρήστης επισκέπτεται ένα ιστότοπο το πρόγραμμα πλοήγησης (browser) επικοινωνεί με έναν διακομιστή (server) μέσω του πρωτοκόλλου HTTP, ο οποίος παράγει τις ιστοσελίδες και τις αποστέλλει στο πρόγραμμα πλοήγησης. Ο Apache είναι ένας από τους δημοφιλέστερους εξυπηρετητές ιστού, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες όπως τα Windows, το Linux, το Unix και το Mac OS X. Κυκλοφόρησε υπό την άδεια λογισμικού Apache και είναι λογισμικό ανοιχτού κώδικα. Συντηρείται από μια κοινότητα ανοικτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation).

Ο Apache χρησιμοποιείται και σε τοπικά δίκτυα σαν διακομιστής συνεργαζόμενος με συστήματα διαχείρισης Βάσης Δεδομένων π.χ. Oracle, MySQL.

Η πρώτη του έκδοση, γνωστή ως NCSA HTTPd, δημιουργήθηκε από τον Robert McCool και κυκλοφόρησε το 1993. Θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του παγκόσμιου ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε

απέναντι στον εξυπηρετητή http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται άλλους εξυπηρετητές βασισμένους στο Unix σε λειτουργικότητα και απόδοση. Από το 1996 ήταν από τους πιο δημοφιλείς όμως από τον Μάρτιο του 2006 έχει μειωθεί το ποσοστό της εγκατάστασής του κυρίως από τον Microsoft Internet Information Services και την πλατφόρμα .NET. Τον Οκτώβριο του 2007 το μερίδιο του ήταν 47.73% από όλους τους ιστοτόπους.

LAMP

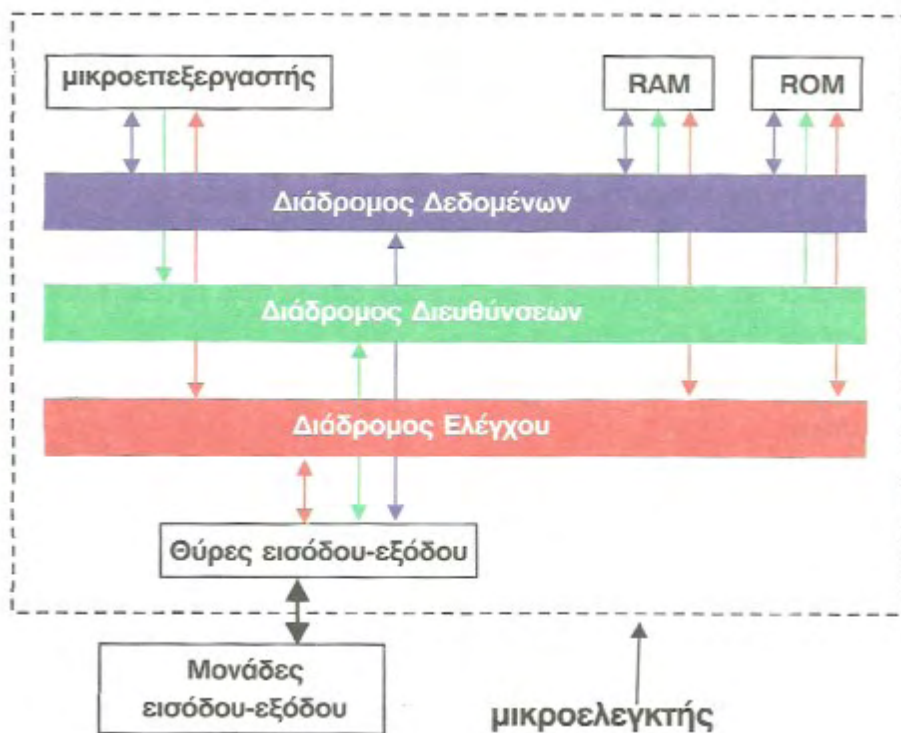
Ο συνδυασμός Linux/Apache/MySQL/PHP, που είναι η πιο δημοφιλής πλατφόρμα εκτέλεσης ιστοσελίδων είναι γνωστός και με το ακρωνύμιο LAMP. Παρόμοια, ο συνδυασμός */Apache/MySQL/PHP ονομάζεται *AMP, όπου το πρώτο αρχικό αντιστοιχεί στην πλατφόρμα, στην οποία εγκαθίστανται ο Apache, η MySQL και η PHP (π.χ. Windows, Mac OS X).

Ο LAMP συνήθως εγκαθίσταται και ρυθμίζεται στο Linux με τη βοήθεια του διαχειριστή πακέτων της εκάστοτε διανομής. Στην περίπτωση άλλων λειτουργικών συστημάτων, επειδή το κατέβασμα και η ρύθμιση των ξεχωριστών προγραμμάτων μπορεί να είναι πολύπλοκη, υπάρχουν έτοιμα πακέτα προς εγκατάσταση, όπως το XAMPP και το WAMP για τα Windows και το MAMP για το Mac OS X.

Η Πλευρά του Μικροελεγκτή

Μικροελεγκτές Γενικά

Η σχεδίαση ενός μικροϋπολογιστικού συστήματος, ενός συστήματος βασισμένο σε μικροεπεξεργαστή, ο οποίος ελέγχει μια πληθώρα από περιφερειακές συσκευές, περιλαμβάνει, το ολοκληρωμένο κύκλωμα του μικροεπεξεργαστή και τις περιφερειακές συσκευές όπως μνήμες, συσκευές θύρες κ.τ.λ. Η σύνδεση και ο συγχρονισμός των συσκευών με τον μικροεπεξεργαστή καθιστά πολύπλοκη τη σχεδίαση του. Μια λύση στο πρόβλημα θα ήταν να υπήρχε ένα ολοκληρωμένο το οποίο να ενσωματώνει κάποιες απαραίτητες περιφερειακές συσκευές. Ένα τέτοιο κύκλωμα είναι ο μικροελεγκτής (microcontroller).



Ένας μικροελεγκτής έχει ενσωματωμένη μνήμη και θύρες Εισόδου/Εξόδου (I/O), ενώ αντίθετα σε ένα μικροεπεξεργαστή χρειάζεται να συνδέσουμε τα παραπάνω σαν πρόσθετα περιφερειακά κυκλώματα. Οι μικροελεγκτές είναι το κύριο στοιχείο σε πολλά είδη ηλεκτρονικού εξοπλισμού. Ένα τυπικό σπίτι στο δυτικό κόσμο είναι πιθανό να περιλαμβάνει μόνο έναν ή δύο γενικού σκοπού μικροεπεξεργαστές, αλλά περισσότερους από είκοσι μικροελεγκτές. Μικροελεγκτές απαντώνται σε οποιονδήποτε τύπο ηλεκτρικής συσκευής, πλυντήρια ρούχων, φούρνους μικροκυμάτων, τηλέφωνα κ.λπ.

Οι περισσότεροι μικροελεγκτές βασίζονται στην αρχιτεκτονική Von Neumann, η οποία καθόρισε σαφώς τα τέσσερα βασικά συστατικά που απαιτούνται για ένα ψηφιακό σύστημα. Αυτά περιλαμβάνουν έναν επεξεργαστικό πυρήνα (CPU), τη μνήμη για το πρόγραμμα και τα δεδομένα (RAM), χώρο μόνιμης αποθήκευσης (FLASH σε έναν

Μικροελεγκτή), καθώς επίσης και τις θύρες I/O για επικοινωνία με εξωτερικές περιφερειακές μονάδες - όλο αυτά σε ένα ενιαίο ολοκληρωμένο κύκλωμα.

Οι μικροελεγκτές συνήθως περιλαμβάνουν θύρες I/O για σύνδεση με ποικίλες συσκευές εισόδου/εξόδου, όπως αναλογικές (με ενσωματωμένους μετατροπείς από αναλογικό σε ψηφιακό), UARTs ή ειδικευμένα σειριακά interface επικοινωνιών όπως το I²C, το SPI και το Controller Area Network (CAN). Μερικοί σύγχρονοι μικροελεγκτές έχουν ενσωματωμένη και μία υψηλού επιπέδου γλώσσα προγραμματισμού όπως η BASIC.

Arduino Uno – AVR ATmega328p

Στο σύστημα που υλοποιήθηκε χρησιμοποιήθηκε ο μικροελεγκτής AVR της ATMEL, και ειδικότερα ο ATmega328p, λόγω των πολλών δυνατοτήτων του, αλλά και της χαμηλής τιμής του

Τα βασικά χαρακτηριστικά του συγκεκριμένου μικροελεγκτή είναι:

- Αναβαθμισμένη RISC αρχιτεκτονική
- 131 πολύ ισχυρές εντολές – οι περισσότερες απαιτούν μόνο ένα κύκλο ρολογιού για την εκτέλεση τους
- 32 καταχωρητές μεγέθους 8-bit γενικής χρήσης
- Μέχρι 20 MIPS στα 16MHz
- On chip πολλαπλασιαστής 2 κύκλων
- Μνήμη προγράμματος και δεδομένων
- 32 KB Αυτοπρογραμματιζόμενη μνήμη flash
- 1 KB EEPROM μνήμη
- 2 KB εσωτερική SRAM
- Πλήθος περιφερειακών
- 2 timers 8-bit με ξεχωριστά ρολόγια
- 1 timer 16-bit με ξεχωριστό ρολόι
- Μετρητής πραγματικού χρόνου με εξωτερικό κρύσταλλο
- 14 γραμμές για ψηφιακή είσοδο/έξοδο (6 εκ των οποίων υποστηρίζουν PWM)
- 6 γραμμές για αναλογική είσοδο (με 10-bit αναλογο-ψηφιακό μετατροπέα ADC)
- Master/Slave SPI λειτουργία
- Σειριακή θύρα με δυνατότητες σύγχρονης και ασύγχρονης λειτουργίας
- Watchdog timer με ξεχωριστό κρύσταλλο
- On-chip αναλογικό συγκριτή

Ειδικά χαρακτηριστικά – Reset αυτόματα με την τροφοδότηση

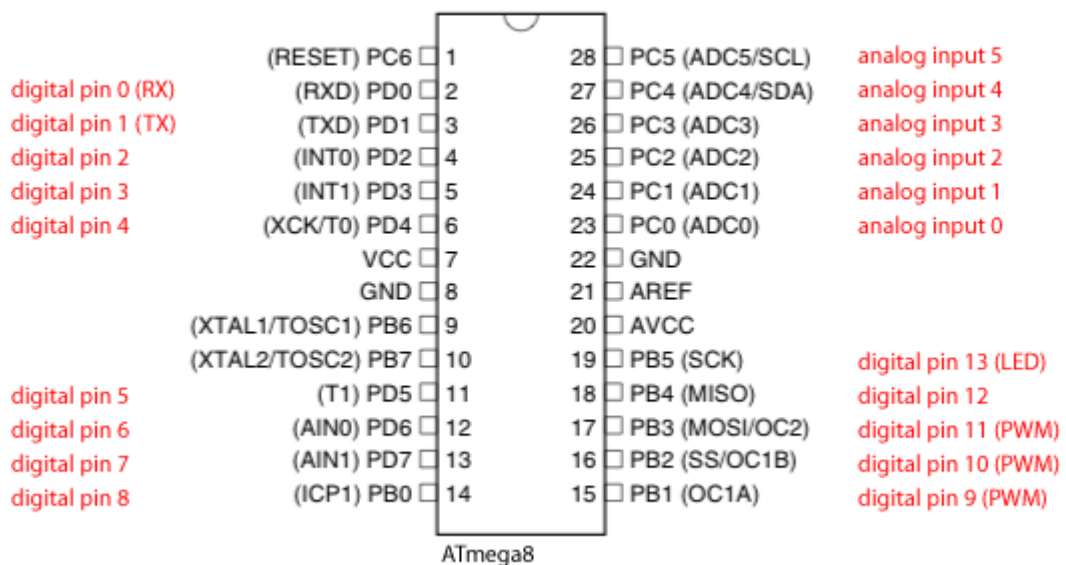
- Εσωτερικό ρολόι
- Εσωτερικές και εξωτερικές διακοπές
- 6 sleep modes για εξοικονόμηση ενέργειας
- Ταχύτητα έως 16MHz

Το αναπτυξιακό που επιλέχθηκε για την υλοποίηση του συστήματος και ενσωματώνει τον ATmega328p είναι το Arduino Uno.

Πρόκειται για μια ηλεκτρονική πλατφόρμα ανοιχτού λογισμικού. Μπορεί να χρησιμοποιηθεί από οποιονδήποτε που ενδιαφέρεται να δημιουργήσει διαδραστικές εφαρμογές. Μπορεί να δεχθεί σαν είσοδο μια ποικιλία από αναλογικά ή ψηφιακά σήματα και να ελέγχει σύμφωνα με τα σήματα αυτά κάποιες περιφερειακές συσκευές που θα είναι συνδεδεμένες σε αυτόν, όπως LEDs, διακόπτες, κινητήρες κ.τ.λ. Ο μικροελεγκτής του προγραμματίζεται χρησιμοποιώντας την γλώσσα προγραμματισμού Arduino. Στην παρακάτω εικόνα φαίνεται η αντιστοίχιση των ακροδεκτών (pins) του ολοκληρωμένου κυκλώματος του μικροελεγκτή με τους ακροδέκτες του αναπτυξιακού -Arduino Uno.

Arduino Pin Mapping

www.arduino.cc



Βασικά χαρακτηριστικά του Arduino Uno

- Ψηφιακά pins (Digital Pins)

Το Arduino Uno έχει 14 ψηφιακά pins τα οποία μπορούν να οριστούν είτε σαν είσοδοι, είτε σαν έξοδοι. Από τα 14 αυτά ψηφιακά pins κάποια εκτός από τη λειτουργία I/O έχουν και άλλες επιπρόσθετες λειτουργίες :

- Digital pins 1, 2 : για σειριακή επικοινωνία UART
- Digital pins 3, 5, 6, 9, 10, 11 : PWM (Pulse Width Modulation), για να έχουμε ψηφιακούς παλμούς μεταβλητού πλάτους.

Είναι σημαντικό επίσης, να αναφερθεί ότι και οι αναλογικές είσοδοι (analog inputs) μπορούν να χρησιμοποιηθούν σαν ψηφιακά pins.

Ιδιότητες των pins που ορίζονται σαν είσοδοι (INPUT)

Όλα τα pins του μικροελεγκτή λειτουργούν εξ' ορισμού σαν είσοδοι, οπότε δεν χρειάζεται να δηλώνεται η συγκεκριμένη λειτουργία στον κώδικα του μικροελεγκτή. Αυτό σημαίνει ότι η κατάσταση μιας εισόδου μπορεί να αλλάξει και από κάποια τυχαία αιτία. Κάτι τέτοιο βέβαια εγκυμονεί κινδύνους. Για παράδειγμα, αν ένα pin είναι στον αέρα ή είναι συνδεδεμένο με ένα μη οδηγούμενο καλώδιο, μπορεί ανά πάσα στιγμή, λόγω εξωτερικού θορύβου, στατικού ηλεκτρισμού, επιρροής πεδίου κ.τ.λ να αλλάξει η κατάστασή του. Όπως είναι φυσικό αυτό αποτελεί μια ανεπιθύμητη λειτουργία. Γι' αυτό λοιπόν ένα pin εισόδου τοποθετείται σε μια γνωστή κατάσταση αν δεν οδηγείται από κάποιο σήμα εισόδου. Αυτό μπορεί να επιτευχθεί τοποθετώντας μια αντίσταση (pullup) μεταξύ του pin και της γραμμής τροφοδοσίας του μικροελεγκτή. Συγκεκριμένα, τοποθετούνται εσωτερικά σε κάθε pin αντιστάσεις 20KΩ, οι οποίες ενεργοποιούνται ή απενεργοποιούνται προγραμματιστικά κλείνοντας ή ανοίγοντας ένα διακόπτη. Με αυτόν τον τρόπο αποφεύγουμε ανεπιθύμητες εισόδους.

Ιδιότητες των pins που ορίζονται σαν έξοδοι (OUTPUT)

Τα pins που ορίζονται ως έξοδοι μπορούν να οδηγήσουν ένα κύκλωμα που θα συνδεθεί σε αυτά. Τα pins του Atmega μπορούν να δώσουν μέχρι 40mA σε κάποιο κύκλωμα. Αυτό είναι αρκετό να ανάψει ένα LED ή να ενεργοποιηθούν κάποιοι αισθητήρες, αλλά δεν επαρκεί να ενεργοποιηθούν ρελέ ισχύος ή κινητήρες. Γι' αυτό το λόγο συνιστάται να συνδέεται σε κάθε έξοδο μια αντίσταση 470Ω ή 1KΩ για προστασία του μικροελεγκτή. Σε αντίθετη περίπτωση μπορεί να έχουμε καταστροφή κάποιου pin του μικροελεγκτή.

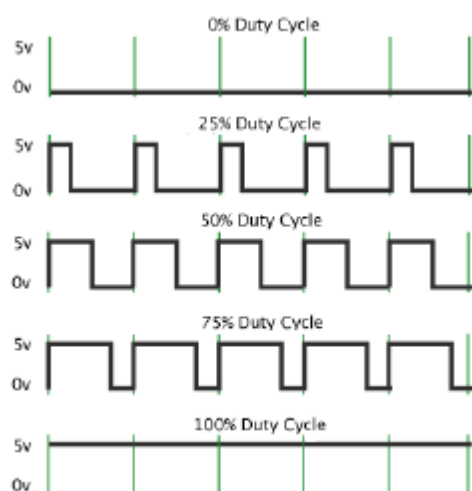
Αναλογικά pins (Analog Input Pins)

A/D Converter

Το Arduino board έχει 6 γραμμές για αναλογική είσοδο, σε καθεμία από τις οποίες συνδέεται ένας μετατροπέας από αναλογικό σε ψηφιακό (Analog to Digital Converter - ADC). Ο κάθε μετατροπέας έχει 10 bit ανάλυσης, με αποτέλεσμα να επιστρέφει ακεραίους από 0 έως 1023, ενώ η βασική λειτουργία των αναλογικών pins είναι να διαβάζουν αναλογικές εισόδους π.χ. αισθητήρες. Μπορούν επίσης να χρησιμοποιηθούν σαν ψηφιακές γραμμές εισόδου/εξόδου γενικού σκοπού, όπως ακριβώς και τα pins 0-13.

Διαμόρφωση Πλάτους Παλμών (PWM)

Η διαμόρφωση Πλάτους Παλμών, PWM, μας δίνει τη δυνατότητα να παράγουμε ψηφιακούς παλμούς μεταβλητού πλάτους. Η διάρκεια που ο παλμός είναι στο λογικό «1» (5V) ονομάζεται πλάτος του παλμού (pulse width), ενώ το ποσοστό του πλάτους του παλμού στο διάστημα μιας περιόδου ονομάζεται duty cycle. Ο ATmega που φιλοξενεί το Arduino board μας δίνει τη δυνατότητα να μεταβάλλουμε το duty cycle των παλμών εξόδου στις PWM ψηφιακές γραμμές. Η χρησιμότητα των γραμμών με δυνατότητα παραγωγής παλμών μεταβλητού πλάτους θα γίνει αντιληπτή από το ακόλουθο παράδειγμα: δίνοντας παλμούς με μεταβαλλόμενο duty cycle σε ένα LED, το αποτέλεσμα θα είναι να αλλάξει η φωτεινότητα του LED σαν να οδηγούνταν από μια αναλογική τάση μεταξύ 0-5 Volt. Χρησιμοποιώντας δηλαδή PWM παλμούς μπορούμε να πάρουμε αναλογικό αποτέλεσμα (έξοδο) από ψηφιακούς παλμούς (τους PWM).



Γλώσσα

Η γλώσσα του Arduino βασίζεται στη γλώσσα Wiring μια παραλλαγή C/C++ για μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega, και υποστηρίζει όλες τις βασικές δομές της C καθώς και μερικά χαρακτηριστικά της C++. Για compiler χρησιμοποιείται ο AVR gcc και ως βασική βιβλιοθήκη C χρησιμοποιείται η AVR libc. Λόγω της καταγωγής της από τη C, στη γλώσσα του Arduino, μπορούν να χρησιμοποιηθούν ουσιαστικά οι ίδιες βασικές εντολές και συναρτήσεις, με την ίδια σύνταξη, τους ίδιους τύπων δεδομένων και τους ίδιους τελεστές όπως και στη C. Πέρα από αυτές όμως, υπάρχουν κάποιες ειδικές εντολές, συναρτήσεις και σταθερές που βοηθούν για τη διαχείριση του ειδικού hardware του Arduino. Τα προγράμματα του Arduino διαίρονται σε τρία μέρη: δομή (structure), τιμές (values) και συναρτήσεις (functions). Οι βασικές και απαραίτητες συναρτήσεις ενός κώδικα για Arduino είναι οι void setup() και void loop(). Η πρώτη χρησιμοποιείται για την αρχικοποίηση των απαραίτητων δεδομένων και εκτελείται μια φορά κατά την εκκίνηση. Η δεύτερη, όπως προδίδει και το όνομά της εκτελείται επαναληπτικά για όσο χρονικό διάστημα είναι ενεργοποιημένη η συσκευή.

```
void setup(){
}

void loop(){
}
```

Γιατί επέλεξα το Arduino

Υπάρχουν πολλές πλατφόρμες υλικού που διατίθενται, αλλά το Arduino είναι μια καλή επιλογή, διότι:

Πρόκειται για ένα ανοιχτού κώδικα software και hardware, έτσι τόσο το λογισμικό όσο και το υλικό είναι εξαιρετικά προσιτά και πολύ ευέλικτα και μπορούν εύκολα να προσαρμοστούν και να επεκταθούν. Επίσης σε μια ενδεχόμενη μετατροπή της κατασκευής σε προϊόν, η σχεδίαση μπορεί να κρατήσει τα απαραίτητα στοιχεία μόνο, να προσθέσει άλλα επιπλέον και να προσαρμοστεί κατά βούληση.

Είναι ευέλικτο, προσφέρει διάφορες ψηφιακές και αναλογικές εισόδους, SPI, I2C, σειριακή διασύνδεση και ψηφιακές και PWM εξόδους.

Είναι εύκολο στη χρήση, συνδέεται με έναν υπολογιστή μέσω USB και επικοινωνεί χρησιμοποιώντας το πρότυπο σειριακό πρωτόκολλο, τρέχει σε αυτόνομη λειτουργία και ως διεπαφή συνδεδεμένο με υπολογιστές PC / Macintosh.

Είναι οικονομικό, κοστίζει λιγότερο από 15 ευρώ η πλακέτα και έρχεται με ελεύθερο περιβάλλον ανάπτυξης.

Υποστηρίζεται από μια ήδη ευρεία on-line κοινότητα, πολύς πηγαίος κώδικας είναι ήδη διαθέσιμος και έτοιμος να χρησιμοποιηθεί. Η αξιοπιστία του έχει δοκιμαστεί σε βάθος χρόνου.

Έχει χαμηλή κατανάλωση ρεύματος, της τάξης των 250 mW χωρίς να εφαρμοστεί κάποια διαδικασία ύπνωσης ή εξοικονόμησης ενέργειας.

Διαδικασία ανάπτυξης

Υλικό μέρος

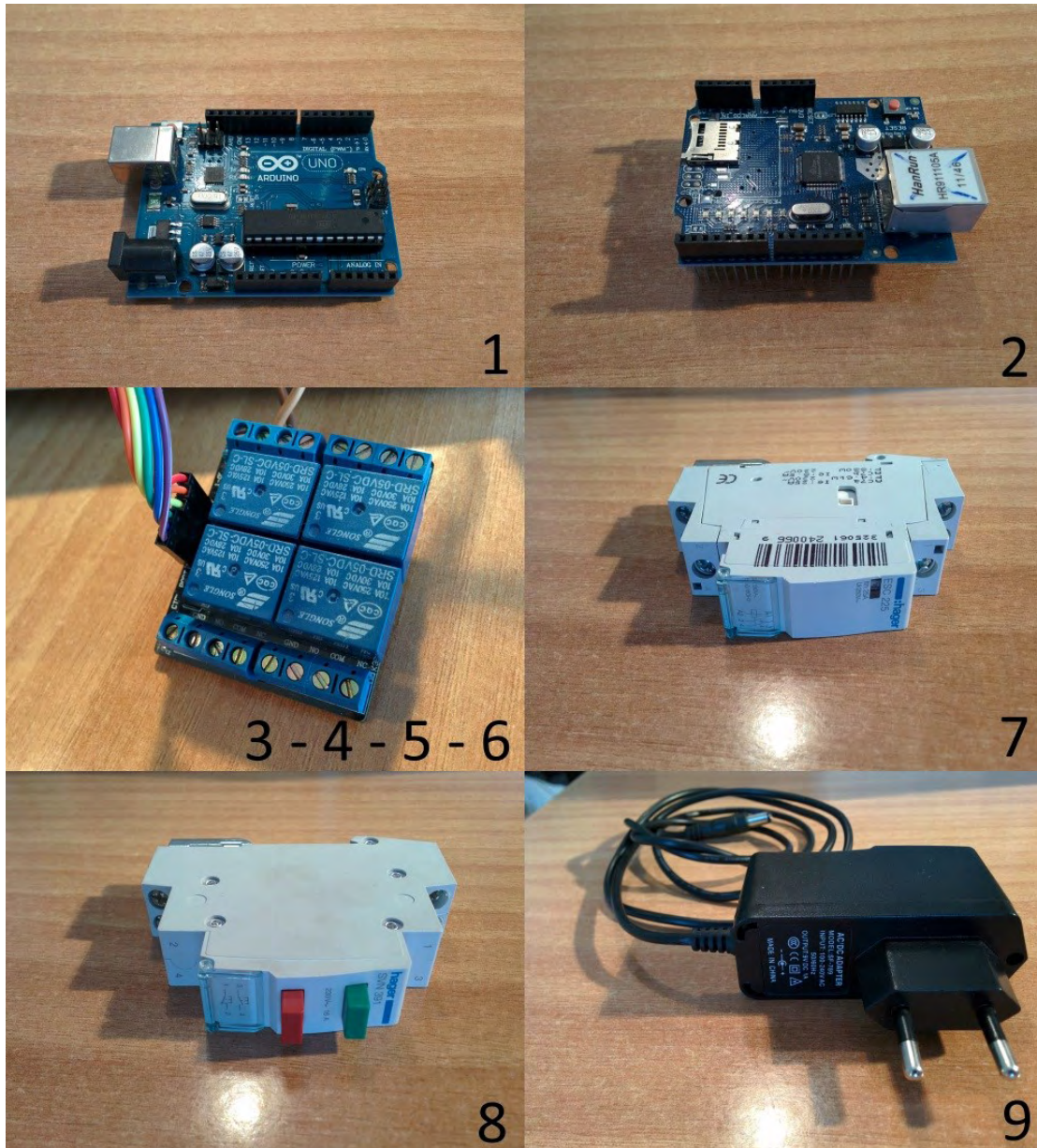
Απαραίτητα στοιχεία

Ο σκοπός του πονήματος αυτού, είναι η επίδειξη υλοποίησης ενός τηλεχειριζόμενου μέσω οποιουδήποτε Browser διακόπτη. Για το σκοπό αυτό επέλεξα μια συσκευή ηλεκτρικού θερμοσίφωνα. Αντίστοιχα μπορεί να χρησιμοποιηθεί για οποιαδήποτε ηλεκτρική συσκευή, όμως για τη συγκεκριμένη θα απαιτηθούν κάποια επιπλέον στάδια, λόγω υψηλής ροής ρεύματος.

Για την υλοποίηση συνολικά του παραπάνω σχήματος, θα χρειαστούν κατ' ελάχιστο τα παρακάτω:

1. Arduino Uno (ή οποιαδήποτε άλλη έκδοση)
2. [Ethernet Shield](#)
3. [Διακόπτης 5V 10A](#) *
4. Τρανζίστορ 2N2222 *
5. Δίοδος 1N4007 *
6. Αντίσταση 1KΩ *
7. [Relay 220V 25A](#)
8. Διακόπτης 2 κομβίων 220V 16A
9. Μετασχηματιστής 9V DC 1A για την τροφοδοσία του Arduino

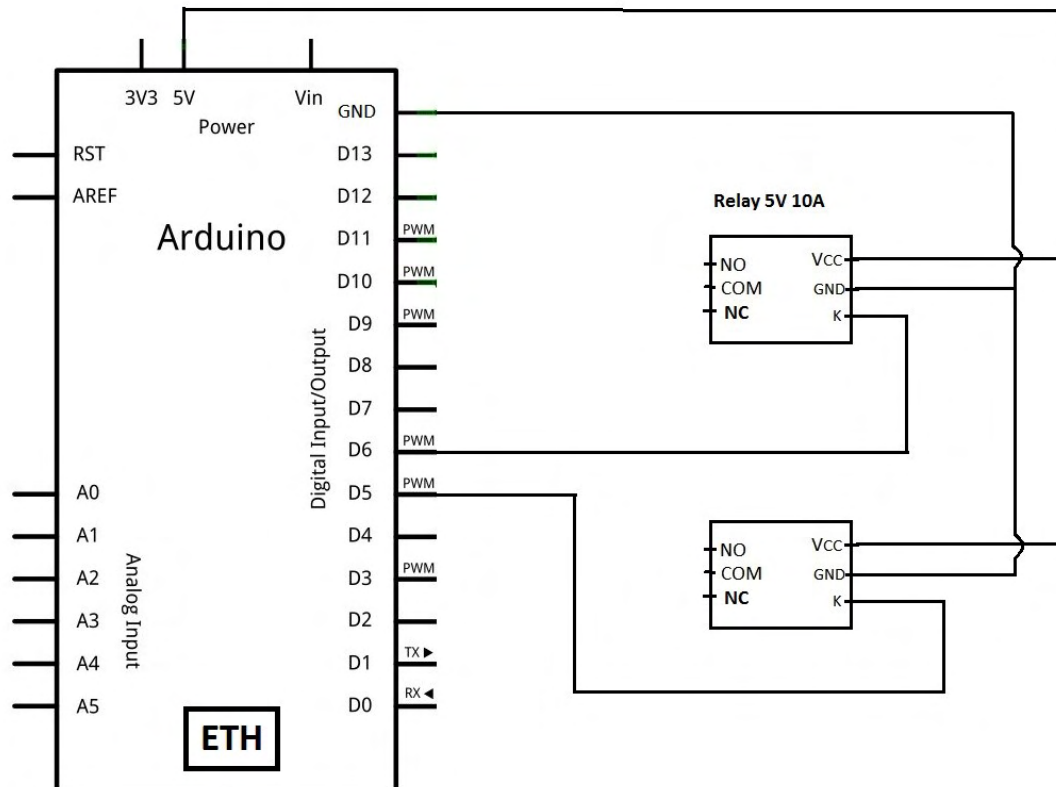
*Τα συγκεκριμένα στοιχεία τα προμηθεύτηκα σε ένα πακέτο [αυτού](#) του τύπου.



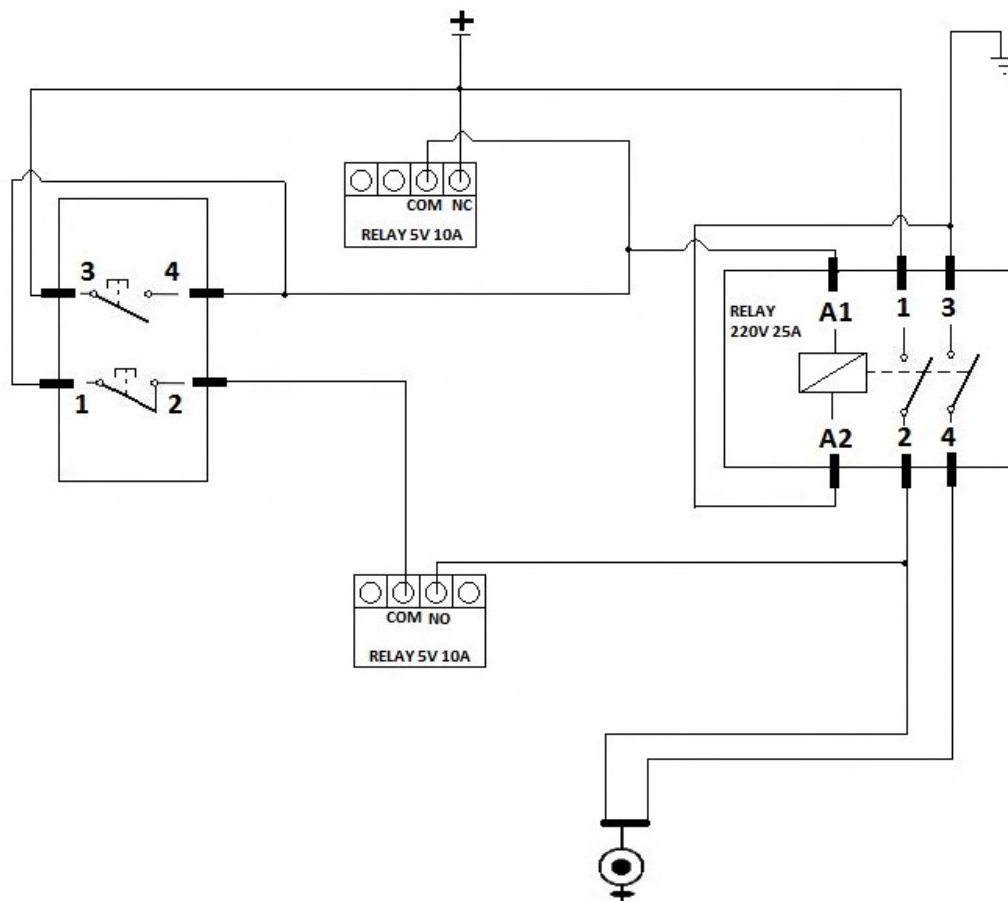
Διαδικασία επιλογής-διασύνδεσης

Η συσκευή του Arduino, μαζί με όλα τα απαιτούμενα shields, απαιτούν ρεύμα θεωρητικά από 3 έως 20 Volts. Αυτό μπορεί να διχορευθεί με τρεις διαφορετικούς τρόπους. Είτε από τη σύνδεση μέσω USB μέσω ενός Η/Υ, είτε από εξωτερική τροφοδοσία (δίκτυο ηλεκτροδότησης ή μπαταρίες), είτε μέσω του καλωδίου Ethernet. Η σταθερότερη λειτουργία και αυτονομία επιτυγχάνεται μέσω τροφοδοτικού των 9 Volt, 1 Ampere. Αυτό δεν μπορεί να τεκμηριωθεί ή να υπολογιστεί λαμβάνοντας υπ' όψη τον αριθμό των χρησιμοποιούμενων περιφερειακών, αλλά είναι ένα ασφαλές νούμερο που έχει επαληθευτεί μέσω της πράξης. Η επιλογή μου λοιπόν είναι αυτή, μέσω του μετασχηματιστή DC της εικόνας 9.

Οι ψηφιακές εξόδους του Arduino όπως έχω αναλύσει προηγουμένως είναι συνολικά 14, από 0 έως 13. Από αυτές, οι 4, δηλαδή από 10 έως 13 χρησιμοποιούνται από τη διασύνδεση του Ethernet shield. Οι διαθέσιμες δηλαδή είναι από 0 έως την 9. Σε αυτές συνδέω τα τέσσερα διαθέσιμα ρελέ (των 10 Ampere), καθώς το πακέτο που έχω προμηθευτεί περιέχει τετράδα. Από αυτά θα χρησιμοποιήσω τα 2, ένα για την ενεργοποίηση του ισχυρότερου ρελέ (των 25 Ampere) κι ένα για την απενεργοποίησή του. Επίσης η απαιτούμενη για τη λειτουργία τους τάση θα δοθεί από τις εξόδους του Arduino, +5Volt και Gnd.



Πάνω στα ρελέ αυτά (της εικόνας, σχήμα 3-4-5-6), στις εξόδους NO, COM και NC, COM θα συνδεθεί το ισχυρό ρελέ (στην εικόνα, σχήμα 7). Λόγω της φύσης της εφαρμογής, η οποία είναι πάνω σε ηλεκτρικό θερμοσίφωνο, δεν μπορεί ένα ρελέ που ενεργοποιείται με τα 5 Volts της εξόδους του Arduino να ελέγξει τα 20 περίπου Ampere τα οποία διαρρέουν το κύκλωμα του θερμοσίφωνα. Οπότε η λύση που αποφάσισα να δώσω είναι η χρήση αυτών των δύο ρελέ «σταδιακά» σε σειρά.



Επίσης, κρίθηκε απαραίτητη η χρήση ενός φυσικού διακόπτη στη θέση του συνηθισμένου, στον ηλεκτρολογικό πίνακα, ο οποίος να συνεργάζεται όμως με την όλη ιδέα του εξ αποστάσεως χειρισμού. Η πιο απλή λύση στο πνεύμα του μικροελεγκτή είναι ένας διακόπτης ο οποίος δίνει στον ίδιο το μικροελεγκτή το σήμα για την κλείσιμο ή το άνοιγμα του κυκλώματος και ταυτόχρονα την γνώση σε αυτόν της κατάστασης του, αφού όλες οι κινήσεις θα περνάνε μέσα από αυτόν. Το μειονέκτημα της υλοποίησης αυτής έγκειται στην πλήρη εξάρτηση της χρήσης της συσκευής μας από τον μικροελεγκτή, κάνοντας αδύνατο να λειτουργήσει η συσκευή σε περίπτωση βλάβης σε αυτόν. Αυτός είναι ο λόγος που κατέληξα στην χρήση του διακόπτη διπλού κομβίου (εικόνα 8). Έχει συνδεθεί παράλληλα με τα δύο αυτόματα ρελέ που χρησιμοποιούνται. Το μεγάλο αυτόματο ρελέ πυροδοτείται (είτε για να ανοίξει το κύκλωμα είτε για να το κλείσει), όχι με συνεχόμενο ρεύμα (AC 220V), αλλά με έναν παλμό. Από πλευράς αυτόματων ρελέ υλοποιείται με στιγμιαία τάση των 300 ms (χωρίς να υπάρχει συγκεκριμένο όριο για αυτό). Ο κώδικας που τον παράγει έχει αυτή τη μορφή:

```

void switchON(int K){
    digitalWrite(K, LOW);
    delay(300);
    digitalWrite(K, HIGH);
}

```

```

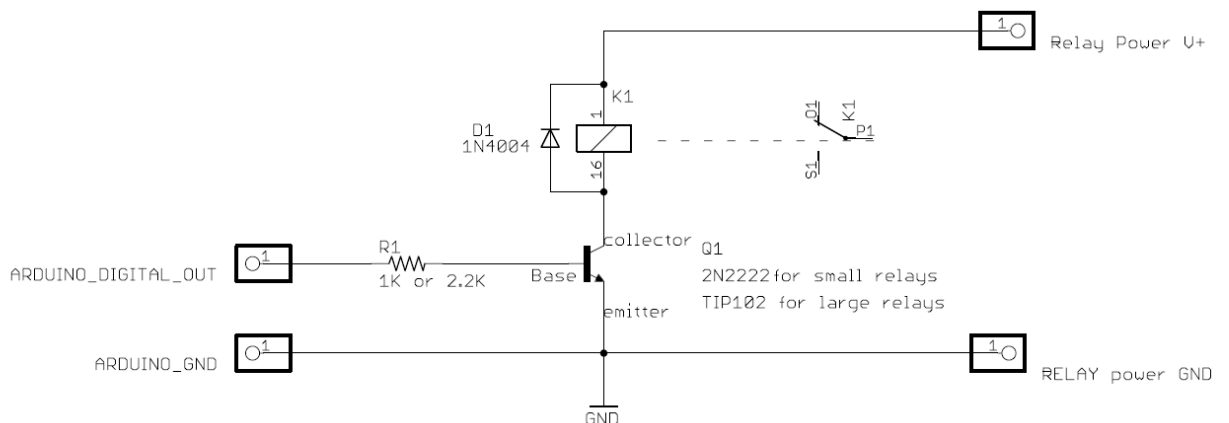
void switchOFF(int K){
    int i = K+1;
    digitalWrite(i, LOW);
    delay(300);
    digitalWrite(i, HIGH);
}

```

Από την πλευρά του χειροκίνητου διακόπτη, ο παλμός αυτός παράγεται με το στιγμιαίο πάτημα του κουμπιού. Δηλαδή για όση ώρα είναι πατημένο (είτε το ON είτε το OFF) περνάει ρεύμα (AC 220V) προς το ισχυρό ρελέ και το πυροδοτεί/αποπυροδοτεί μόνιμα.

Με τη σειρά του το ισχυρό ρελέ ανοίγει ή κλείνει το κύκλωμα τροφοδοσίας του θερμοσίφωνα. Καλό είναι επίσης, να χρησιμοποιηθεί μια ενδεικτική λυχνία λειτουργίας του θερμοσίφωνα, όπως απαιτείται σε μια κλασική εγκατάσταση, αλλά και να μην αντικατασταθεί επίσης και ο διακόπτης της κλασικής εγκατάστασης, αλλά να συνδεθεί το νέο κύκλωμα σε σειρά και κατόπιν αυτού, καθώς αφαίρεσή του αντιτίθεται στις οδηγίες της ΔΕΗ.

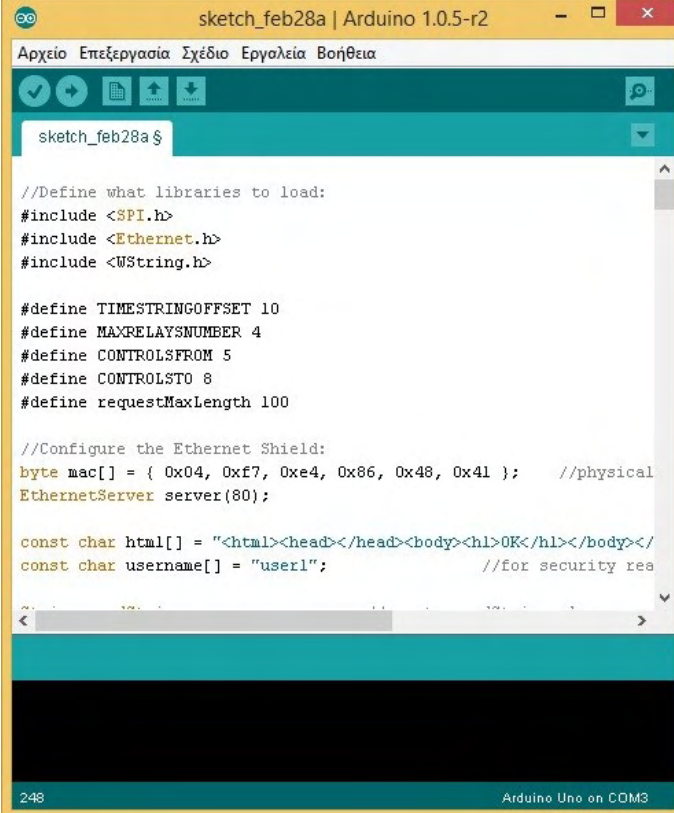
Ένα τυπικό ρελέ που μπορεί να συνδεθεί με Arduino μπορεί να σχεδιαστεί με τον παρακάτω τρόπο:



Προγραμματιστικό μέρος Arduino – Web Application

Η συσκευή του Arduino, προγραμματίστηκε με 4 βασικούς σκοπούς:

1. Να δέχεται μηνύματα από τον διαμεσολαβητή (web application).
2. Να αναλύει το μήνυμα και να εκτελεί αναλόγως.
3. Να στέλνει περιοδικά την κατάσταση των συνδεδεμένων σε αυτό συσκευών, ώστε να ανανεώνεται και η IP του στη βάση δεδομένων.
4. Να ελέγχει περιοδικά για τυχόν ενεργούς χρονοδιακόπτες.



```
sketch_feb28a | Arduino 1.0.5-r2
Αρχείο Επεξεργασία Σχέδιο Εργαλεία Βοήθεια
sketch_feb28a $
//Define what libraries to load:
#include <SPI.h>
#include <Ethernet.h>
#include <WString.h>

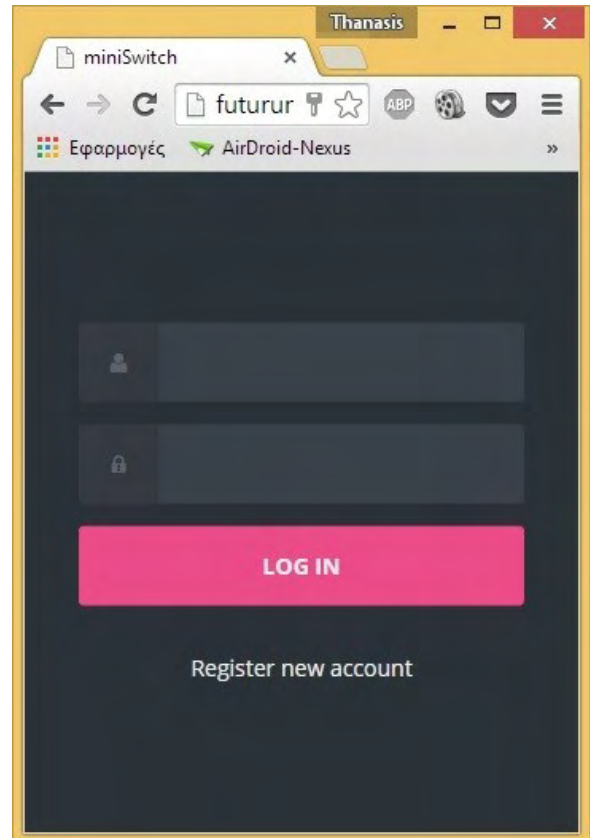
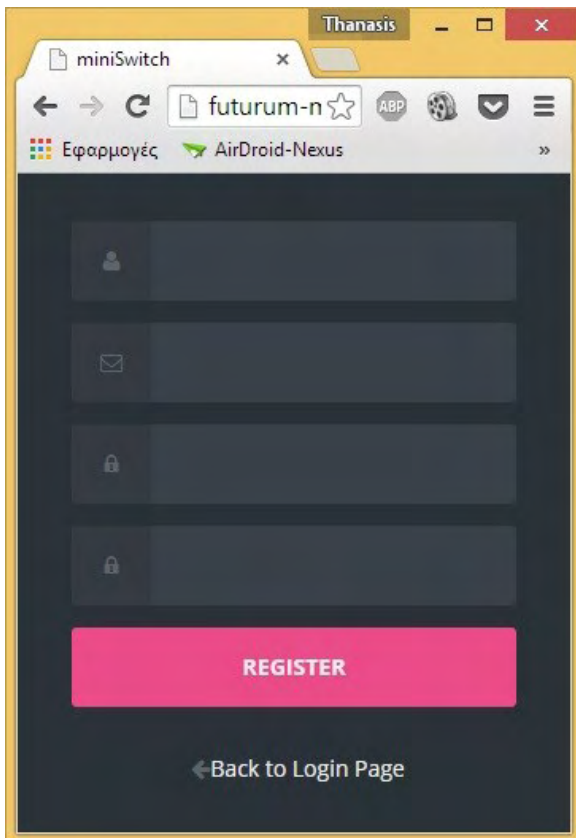
#define TIMESTRINGOFFSET 10
#define MAXRELAYSNUMBER 4
#define CONTROLFROM 5
#define CONTROLSTO 8
#define requestMaxLength 100

//Configure the Ethernet Shield:
byte mac[] = { 0x04, 0xf7, 0xe4, 0x86, 0x48, 0x41 }; //physical
EthernetServer server(80);

const char html[] = "<html><head></head><body><h1>OK</h1></body></html>";
const char username[] = "user1"; //for security rea
248 Arduino Uno on COM3
```

Σε αντιπαράβολή το web application έχει 4 βασικές λειτουργίες:

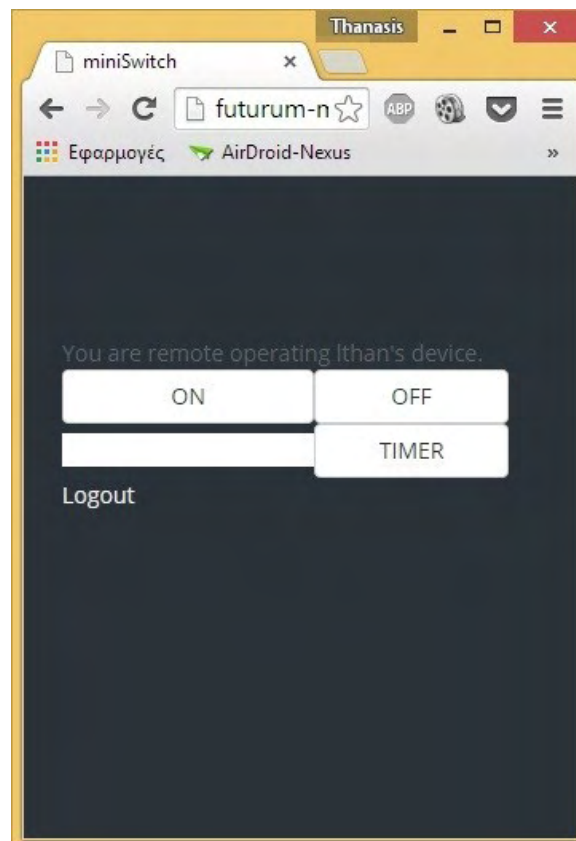
1. Επιτρέπει την εγγραφή κάθε χρήστη με username, password και e-mail διεύθυνση.
2. Συνδέει τον χρήστη στο περιβάλλον της εφαρμογής με τα παραπάνω αυτά στοιχεία.
3. Έχει επιλογή για άνοιγμα του διακόπτη, σβήσιμό του και ενεργοποίηση χρονοδιακόπτη για τη συσκευή.
4. Παράλληλα με αυτά, λαμβάνει μηνύματα από όλες τις συσκευές που είναι εγγεγραμμένες/εξυπηρετεί, ώστε να διατηρεί την κατάστασή τους και να ανανεώνει πιθανώς την διεύθυνση IP τους.



Web Application

Ξεκινώντας του διαμεσολαβητή κι επειδή είναι μια διαδικτυο εφαρμογή, διαλέξω έναν τρόπο όλων των στοιχείων. Η πιο μέθοδος είναι η βάση βασισμένη σε MySQL. στοιχεία που πρέπει για τον κάθε χρήστη

- User
- User
- User



από την πλευρά στον τηλεχειρισμό εκτεθειμένη στο έπρεπε να αποθήκευσης απαιτούμενων διαδομένη δεδομένων, Τα απαραίτητα να διατηρηθούν είναι τα εξής:

name
password
e-mail address

- Arduino device IP
- Arduino device port

Το πρώτο βήμα δηλαδή ήταν να υλοποιήσω την βάση δεδομένων και έγινε έτσι:

```
CREATE DATABASE IF NOT EXISTS `DB_NAME`;
```

```
CREATE TABLE IF NOT EXISTS `DB_NAME`.`tableName` (  
  `user_id` int(11) NOT NULL AUTO_INCREMENT COMMENT 'auto incrementing user_id of each  
user, unique index',  
  `user_name` varchar(64) COLLATE utf8_unicode_ci NOT NULL COMMENT 'user''s name,  
unique',  
  `user_password_hash` varchar(255) COLLATE utf8_unicode_ci NOT NULL COMMENT 'user''s  
password in salted and hashed format',  
  `user_email` varchar(64) COLLATE utf8_unicode_ci NOT NULL COMMENT 'user''s email,  
unique',  
  `device_ip` varchar(20) COLLATE utf8_unicode_ci COMMENT 'device''s IP, unique',  
  `device_port` varchar(10) COLLATE utf8_unicode_ci COMMENT 'device''s port, unique',  
  PRIMARY KEY (`user_id`),  
  UNIQUE KEY `user_name` (`user_name`),  
  UNIQUE KEY `user_email` (`user_email`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci COMMENT='user data';
```

Είναι φανερό ότι οι κωδικοί των χρηστών δεν αποθηκεύονται αυτούσιοι, αλλά hashed με μια ενσωματωμένη στην PHP 5.5 συνάρτηση, την `password_hash()`. Σε περίπτωση παλαιότερης έκδοσης της PHP, χρησιμοποιείται εξωτερικά η ίδια συνάρτηση από τη βιβλιοθήκη `password hashing compatibility`. Οι κωδικοί των χρηστών δεν είναι ανακτήσιμοι. Δηλαδή η αντίστροφη του συγκεκριμένου hashing διαδικασία δεν μπορεί να ακολουθηθεί. Για να γίνει η σύγκριση του αποθηκευμένου και του εισαγόμενου κωδικού, ο εισαγόμενος κωδικοποιείται με τον ίδιο τρόπο και συγκρίνεται το μήκους 40 χαρακτήρων αποτέλεσμά του με το αποθηκευμένο.

Κατά την εγγραφή του χρήστη, για να είναι δυνατή η άμεση λειτουργία της συσκευής, προτείνεται η χρήση του ίδιου δικτύου από το οποίο παίρνει πρόσβαση στο διαδίκτυο και ο μικροελεγκτής. Αυτό γιατί έτσι θα αναγνωριστεί η IP, όντας κοινή, και η επικοινωνία θα είναι σε θέση να γίνει άμεσα. Σε αντίθετη περίπτωση, η IP του μικροελεγκτή θα γίνει γνωστή με την πρώτη επικοινωνία με πρωτοβουλία αυτού με το web application. Αυτό θα διαρκέσει ανάλογα με το χρονικό διάστημα περιοδικής επικοινωνίας που έχουμε ορίσει. Στην περίπτωσή μου 5 λεπτά.

Η πόρτα μέσω της οποίας θα γίνεται η επικοινωνία, ορίζεται από προεπιλογή ως η 8081. Υπάρχει συγκεκριμένος λόγος που έχει να κάνει με το port forwarding του router μας και θα αναλυθεί περαιτέρω αργότερα. Αλλαγή όμως της αποθηκευμένης για επικοινωνία θύρας μπορεί να γίνει με την τακτική επικοινωνία ανανέωσης στοιχείων της βάσης δεδομένων οποιαδήποτε στιγμή θελήσουμε να προγραμματίσουμε διαφορετικά το Arduino.

Κάτι άλλο δεν αξίζει να προστεθεί για τη διαδικασία εγγραφής/εισόδου του χρήστη, εκτός του ότι αυτή μπορεί να γίνει χρησιμοποιώντας είτε το username είτε το password του.

```
SqlCommand = "SELECT user_name, user_email, user_password_hash FROM users  
WHERE user_name = '" . $user_name . "' OR user_email = '" . $user_email . "'";
```

Σε ρηρ το συγκεκριμένο απόσπασμα κώδικα MySQL ενδεικτικά:

Κατόπιν της εισόδου του χρήστη στην κύρια σελίδα χειρισμού του μικροελεγκτή είναι εμφανείς 3 απλές λειτουργίες. Για να λειτουργήσουν αυτές, λαμβάνουν χώρα τα εξής γεγονότα. Ανακτώνται από τη βάση δεδομένων οι ενημερωμένες IP και θύρα της απομακρυσμένης συσκευής και κατόπιν βάσει αυτών χτίζεται μία POST αίτηση. Αυτή έχει δημιουργηθεί ως μια εσωτερική σύμβαση που έγινε, ή αλλιώς ένα απλοποιημένο πρωτόκολλο επικοινωνίας μεταξύ του web application-διαμεσολαβητή και της απομακρυσμένης συσκευής. Οι αιτήσεις που είναι δυνατό να σταλούν έχουν στο σώμα τους μια από τις παρακάτω μορφές:

```
usr=user_name&Kx=state
```

όπου user_name χρησιμοποιείται το αποθηκευμένο, ώστε να γίνει μια τυπική ταυτοποίηση και αποσφαλμάτωση. Δεν τίθεται θέμα με αυτό τον τρόπο ασφάλειας η οποία σε ελάχιστο βαθμό έχει απασχολήσει αυτό το πόνημα και στις προτεινόμενες βελτιώσεις παρακάτω γίνεται μια αναφορά για τα βασικότερα σημεία που χρήζουν «κλειδώματος».

Στη θέση του 'x' μπαίνει ο αριθμός του διακόπτη που θα χρησιμοποιηθεί, σε περίπτωση που περισσότεροι από ένας είναι διαθέσιμοι. Όπου state, μπορούν να λειτουργήσουν τρεις τύποι εντολών:

- 'ON' που κλείνει το εν λόγω κύκλωμα

- 'OF' που ανοίγει το κύκλωμα

- 'OT' που χρησιμοποιείται για να κλείσει το κύκλωμα και να θέσει παράλληλα έναν χρονοδιακόπτη. Αυτό ακολουθείται υποχρεωτικά από '&T=timer', συμβολίζοντας με 'timer' το χρόνο που θα μείνει κλειστό το κύκλωμα σε λεπτά.

Μία συνάρτηση του web application έχει σκοπό τη συλλογή των μηνυμάτων από την πλευρά του μικροελεγκτή. Αυτή εξυπηρετεί έναν διπλό σκοπό. Όπως έχει προαναφερθεί, να συλλέγει πληροφορίες για την κατάσταση του Arduino και να ανανεώνει τη βάση δεδομένων με την πιθανή νέα IP που έχει λάβει η συσκευή λόγω αλλαγής από πλευράς ISP. Η μεν πρώτη λειτουργία εξυπηρετεί καθαρά μελλοντικούς σκοπούς, που θα αναλυθούν σε επόμενο κεφάλαιο, η δεύτερη όμως είναι κρίσιμης σημασίας για τη διατήρηση της επικοινωνίας.

Προγραμματισμός Arduino

Η λειτουργίες που επιτελεί η συσκευή με τα αρθρώματα που έχω χρησιμοποιήσει, είναι αυτές των web server, web client και του μικροελεγκτή. Για αρχή έπρεπε να δώσω στη διασυνδεδεμένη αυτή συσκευή, μια διεύθυνση MAC αλλά και μια προκαθορισμένη διεύθυνση IP και αριθμό θύρας. Η διεύθυνση MAC δεν έχει μεγάλη σημασία αρκεί να είναι μοναδική στο δίκτυο για να μπορέσει να γίνει η διευθυνσιοδότηση της. Ο θύρα που χρησιμοποιήθηκε είναι η 80 όπως είναι το συνηθισμένο για έναν web server. Να τονίσω ότι δεν συμπίπτει με τη θύρα που απευθύνεται η web application (8081), καθώς αυτή γίνεται port forward στην 80 τη στιγμή που παραλαμβάνεται το μήνυμα από το router. Η IP, δίνεται παίρνοντας υπ' όψη τις ρυθμίσεις του router και το εύρος των διευθύνσεων που αυτό χρησιμοποιεί. Επίσης πρέπει να αποφεύγεται η χρήση DHCP, καθώς με τον τρόπο αυτό θα είχαμε κάποιες πιθανότητες αλλαγής διευθύνσεων σε κάθε σύνδεση – αποσύνδεση. Για τη χρησιμοποίηση συναρτήσεων σχετικά με τη σύνδεση Ethernet, χρησιμοποιήθηκε η βιβλιοθήκη Ethernet.h που είναι συμβατή με το chip WizNet5100.

//Configure the Ethernet Shield:

```
byte mac[] = { 0x04, 0xf7, 0xe4, 0x86, 0x48, 0x41 }; //physical mac address
EthernetServer server( 80 ); //Server port
byte ip[] = { 192, 168, 2, 177 }; //Manual setup only
```

Η πρώτη συνάρτηση που εκτελείται μέσα στην loop() είναι η webserver(). Σε αυτή ο μικροελεγκτής περιμένει αιτήσεις POST από την πλευρά του web application, που περιέχουν την εντολή που πρέπει αυτός να εκτελέσει. Το σώμα της POST αποθηκεύεται σε ένα string το οποίο πρέπει ακολούθως να αναγνωριστεί. Η μορφή που έχει μια τέτοια εντολή/μήνυμα είναι προκαθορισμένη και αναλύθηκε στην παρουσίαση της ανάπτυξης του web application.

Στο παρακάτω απόσπασμα φαίνεται ο διαχωρισμός των εντολών σε τύπου 'ON', 'OF' και 'OT'. Φαίνεται ότι είναι απλοϊκός και δεν λαμβάνει υπόψη περιπτώσεις σφαλμάτων, αλλά παραλαμβάνει μόνο μηνύματα από ένα συγκεκριμένο web app το οποίο είναι συμβιβασμένο να γράφει σε αυτή τη μορφή και μόνο.

```

if (input.indexOf("=OT") > -1) { //case timer exists
    switchON(KrelayControl[switchNum]);
    KrelayState[switchNum] = 1;
    KtimerState[switchNum] = 1;
    equalsPlace = input.indexOf('=', equalsPlace + 1);
    tempString = input.substring(equalsPlace + 1);
    tempString.toCharArray(buf, tempString.length());
    KiTime[switchNum] = atol(buf); //mins to millis
    KiTime[switchNum] *= 1000.0;
    KiTime[switchNum] *= 60.0;
}
else if(input.indexOf("=ON") > -1){ //case ON
    switchON(KrelayControl[switchNum]);
    KrelayState[switchNum] = 1;
}
else if(input.indexOf("=OF") > -1){ //case OFF
    switchOFF(KrelayControl[switchNum]);
    KrelayState[switchNum] = 0;
    KtimerState[switchNum] = 0;
}
else{ //case of wrong args
    Serial.println("Wrong incoming arguments");
}

```

Αναλόγως με τους διακόπτες που ενεργοποιούνται/απενεργοποιούνται καθολικές μεταβλητές ενημερώνονται ώστε να κρατείται αρχείο με την κατάστασή όλων. Ιδιαίτερη περίπτωση είναι αυτή του χρονοδιακόπτη. Η τιμή του χρονικού διαστήματος που θα παραμείνει ανοιχτός φυλάσσεται σε ξεχωριστή μεταβλητή (`KiTime[]`) και μετατρέπεται από λεπτά σε χιλιοστά του δευτερολέπτου. Κατόπιν, με μια ρουτίνα ελέγχεται σε κάθε επανάληψη της `loop()` αν ο χρόνος έχει εξαντληθεί, οπότε ο εκάστοτε διακόπτης θα πρέπει να απενεργοποιηθεί.

```
currentMillis = millis();

for(i=0; i<(MAXRELAYSNUMBER); i++){
    if(KtimerState[i] && (currentMillis - previousMillisK > KiTime[i])){
        previousMillisK = currentMillis;
        Serial.print(i);
        Serial.println("closed");
        switchOFF(KrelayControl[i]);
        KrelayState[i] = 0;
        KtimerState[i] = 0;
    }
}
```

Ένας τρόπος να μετρηθεί ο χρόνος είναι αυτός που χρησιμοποιήθηκε με την `millis()`. Αυτή επιστρέφει τι χιλιοστά του δευτερολέπτου από τη στιγμή που έχει ενεργοποιηθεί ο μικροελεγκτής. Υπάρχει ένας τρόπος ακόμη, χρησιμοποιώντας την εντολή `delay()`. Αυτή δεν ταιριάζει με την περίπτωση μας, που είναι συνδυαστική, καθώς κατά την δεδομένη σε αυτήν διάρκεια, παύει οποιαδήποτε λειτουργία και χάνονται οποιεσδήποτε εισερχόμενες αιτήσεις.

Επίσης μέτρηση χρόνου χρησιμοποιήθηκε για την περιοδική αποστολή της κατάστασης των συνδεδεμένων διακοπών. Αυτά τα μηνύματα εξυπηρετούν επίσης, όπως προαναφέρθηκε, το σκοπό της ενημέρωσης για τυχόν αλλαγές της `dynamic IP`.

```
currentMillis = millis();

if(currentMillis - previousMillis > intervalUpdateIP) {
    previousMillis = currentMillis;
    updateIP();
}
```

Με τον παραπάνω τρόπο ελέγχεται η ανάγκη κλήσης της `updateIP()`. Ο χρόνος κατά τον οποίο πρέπει αυτή να κληθεί προκαθορίζεται με την μεταβλητή `intervalUpdateIP` και

στην περίπτωση αυτή επέλεξα να είναι τα 5 λεπτά. Μεγαλώνοντας αυτό το χρονικό διάστημα, μεγαλώνουν και τα νεκρά σημεία στην επικοινωνία του μικροελεγκτή (μόνο τα εισερχόμενα μηνύματα) κατά τη στιγμή της αλλαγής dynamic IP και μετά. Μικραίνοντας το χρονικό διάστημα ανάμεσα στις ενημερώσεις, αυξάνεται ο φόρτος του δικτύου και κυρίως του web server. Μια ακόμη στιγμή που εκτελείται η updateIP() είναι κατά τη διαδικασία αρχικοποίησης. Αν διακρίνουμε τις στιγμές που γίνεται αλλαγή της IP από τον πάροχό μας, αυτές έχουν απόσταση αρκετών ημερών μεταξύ τους, εκτός κι αν επανεκκινηθεί το router μας. Οπότε με τον τρόπο αυτό προλαμβάνω την αδυναμία επικοινωνίας μετά από πτώση ρεύματος. Η αρχικοποίηση θα γίνει μόλις επανέλθει το ρεύμα και μαζί και η ενημέρωση του web server.

```
void updateIP(){
    String data;
    int i;
    data = "usr=" + usernameReg + "&p=" + passReg;

    for(i=0; i<(MAXRELAYSNUMBER); i++){
        data += "&K";
        data += i;
        data += "=";
        if(KrelayState[i]){
            data += "ON";
        }
        else{
            data += "OF";
        }
    }

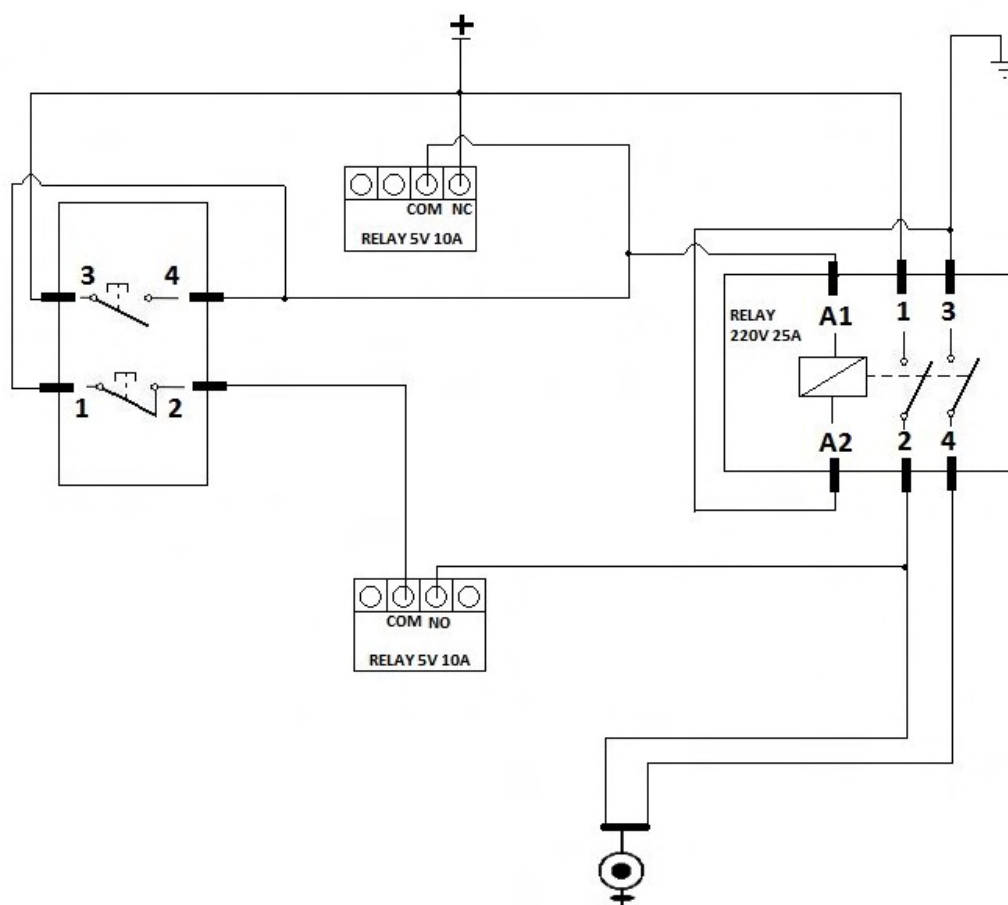
    for(i=0; i<(MAXRELAYSNUMBER); i++){
        data += "&T";
        data += i;
        data += "=";
        if(KtimerState[i]){
            data += KiTime[i];
        }
        else{
            data += "OF";
        }
    }
}
```

```
if (client.connect("webserverAdress",80)) {
    client.println("POST /updateip.php HTTP/1.1");
    client.println("Host: hostAddress");
    client.println("Content-Type: application/x-www-form-urlencoded");
    client.print("Content-Length: ");
    client.println(data.length());
    client.println();
    client.print(data);
}
else{
    Serial.println("No connection to update IP.");
}

if (client.connected()) {
    client.stop();
}
}
```

Διαδικασία Εγκατάστασης

Έχοντας ασχοληθεί με την συνδεσμολογία των εξαρτημάτων και τον προγραμματισμό του Arduino, μένει η εγκατάσταση της όλης κατασκευής στο επιθυμητό σημείο του κτιρίου. Η επιλογή των εξαρτημάτων έγινε με την παραδοχή εγκατάστασής τους στον Ηλεκτρολογικό Πίνακα, οπότε οι επιλογές που έχουμε δεν είναι πολλές. Ανατρέχοντας στο σχέδιο που έχω ήδη περιγράψει, διακρίνεται και η συνδεσμολογία με το δίκτυο ηλεκτρισμού σε γενικές γραμμές.



Το σύμβολο στο κάτω μέρος του σχεδίου είναι αυτό του θερμοσίφωνα στο ηλεκτρολογικό σχέδιο. Οι διευκρινίσεις που οφείλουν να γίνουν είναι οι εξής:

Το + στην κορυφή του σχεδίου και η γείωση πρέπει να συνδεθούν στην φάση και στο ουδέτερο του οικιακού ρεύματος. Τονίζω ότι αυτό πρέπει να γίνει κατόπιν των απαιτούμενων μέτρων προστασίας, την ασφάλεια και τον διακόπτη.

Η ηλεκτρολογική εγκατάσταση δεν είναι ο στόχος της εργασίας αυτής. Οπότε συνεχίζοντας, μας μένουν μόνο κάποιες λεπτομέρειες για να έχουμε την κατασκευή μας λειτουργική στο σύνολό της. Αυτές έχουν να κάνουν με την πρόσβαση στο μικροελεγκτή στο διαδίκτυο. Καθώς επιτελεί, μεταξύ άλλων, τη λειτουργία του web server, θα πρέπει να είναι ορατός και από τον έξω κόσμο, εκτός από αυτόν του τοπικού δικτύου. Κάτι τέτοιο δεν γίνεται επιτρεπτό από το router, έτσι χρειάζονται κάποιες ρυθμίσεις αυτό. Για την επεξήγηση κάποιων ορολογιών, γίνεται αναφορά στο Παράρτημα.

Αρχικά πρέπει να γίνει bind η στατική εσωτερική IP που έχουμε επιλέξει με την διεύθυνση MAC που εισάγαμε χειροκίνητα. Αυτό θα επιβάλει την διατήρηση της συγκεκριμένης IP για τη συγκεκριμένη και μόνο συσκευή και θα την αφαιρέσει από τη δεξαμενή στην οποία έχει πρόσβαση ο DHCP.

Κάτι που πρέπει να ρυμιστεί έτσι ώστε τα εισερχόμενα μηνύματα να βρισκουν τον παραλήπτη τους είναι το Portforwarding. Συγκεκριμένα, πρέπει τα μηνύματα που έρχονται στην πόρτα 8081 (αυτή που ορίστηκε στο web application) να κατευθύνονται αυτόματα στην πόρτα 80 και την τοπική IP του Arduino. Αυτή είναι όπως ορίστηκε στο προηγούμενο βήμα.

Το τελευταίο βήμα αφορά τα εξερχόμενα μηνύματα. Η τοπική IP της συσκευής πρέπει να προστεθεί στο 'τύποις' δίκτυο DMZ.

Προτάσεις Βελτίωσης

Η περιγραφόμενη μέχρι αυτή τη στιγμή κατασκευή, έχει λύσει αρκετά από τα προκύπτοντα προβλήματα, απέχει όμως πολύ από το να παράξει αυτό που αποκαλούμε ένα προϊόν. Τουλάχιστον επιδέχεται πολλών βελτιώσεων και επεκτάσεων. Κάποιες από τις εύκολα εφαρμόσιμες βελτιώσεις είναι:

Η μείωση της κατανάλωσης του μικροελεγκτή

- Απενεργοποίηση των ενσωματωμένων LED.
- Χρησιμοποίηση της sleep time ανά περιόδους που δεν χρειάζεται επεξεργαστική ικανότητα. (πχ μέσω των βιβλιοθήκη JeeLib.h ή lowpower.h)
- Βελτιστοποιημένη χρονικά κατανομή των εξερχόμενων προς τον web server σημάτων.

Η βελτίωση της συνδεσιμότητας

- Χρησιμοποίηση RFID relays για την επικοινωνία της συσκευής με απομακρυσμένους διακόπτες.
- Χρήση κάποιου module που υποστηρίζει σύνδεση wifi ανάμεσα στον μικροελεγκτή και τη συσκευή router.
- Η δυνατότητα απομακρυσμένου επαναπρογραμματισμού του μικροελεγκτή.

Η κρυπτογράφηση των επικοινωνιών

Με τη συγκεκριμένη υλοποίηση εξασφαλίζεται σε κάποιο βαθμό η ασφάλεια της βάσης δεδομένων. Όμως είναι διάτρητη η επικοινωνία ανάμεσα στις κινητές συσκευές και του web server και κυρίως ανάμεσα στον web server και τη συσκευή Arduino. Προτείνονται οι παρακάτω βασικές βελτιώσεις:

- Προστασία της βάσης δεδομένων από SQL injection.
- Χρήση πρωτοκόλλου https για την επικοινωνία ανάμεσα στον web server και στον μικροελεγκτή.
- Εφαρμογή κάποιου είδους κρυπτογράφησης για την end to end επικοινωνία web server - μικροελεγκτή. Με αυτό τον τρόπο θα είναι πιο αξιόπιστη η εξακρίβωση της αυθεντικότητας του χρήστη.
- Σε περίπτωση ασύρματης σύνδεσης του μικροελεγκτή στο δίκτυο, μεγάλης σημασίας είναι η ασφάλεια του ασύρματου δικτύου. (πχ κρυπτασφάλιση wpa2 σε συνδιασμό με επιλεκτική εξυπηρέτηση mac διευθύνσεων)

Περεταίρω βελτιστοποιήσεις ασφάλειας είναι εφικτές αλλά δεν αποτελούν αντικείμενο της παρούσας ανάλυσης.

Υποστήριξη περισσότερων εφαρμογών

Η συγκεκριμένη υλοποίηση είναι ενδεικτική μόνο των εφικτών εφαρμογών στα πλαίσια ενός συστήματος αυτοματισμού ή ενός έξυπνου σπιτιού. Κάποιες από τις τετριμμένες μόνο εφαρμογές είναι:

- Ο έλεγχος φωτισμού.
- Ο έλεγχος θέρμανσης/ψύξης μέσω θερμοστάτη.
- Ο έλεγχος διάφορων οικιακών συσκευών.
- Σύστημα καμερών με ζωντανή μετάδοση.
- Σύστημα ανίχνευσης κίνησης/βροχής/καπνού.

Για την εξυπηρέτηση περισσότερων από μία εφαρμογών, ή συνδυασμού αυτών, από τον web application είναι απαραίτητος ο σχεδιασμός του με έναν δυναμικό τρόπο. Απαιτείται ο χρήστης να μπορεί να προσαρμόσει το περιβάλλον ώστε να προσθέτει/αφαιρεί τις εκάστοτε δυνατότητες και τον αριθμό των συσκευών που ενσωματώνει το σύστημά του. Από την πλευρά του μικροελεγκτή, είναι πολύ χρήσιμη η χρήση wifi shield που επιτρέπει τον απομακρυσμένο επαναπρογραμματισμό του χωρίς τη διαμεσολάβηση του χρήστη και καλωδίου USB. Δηλαδή, η ιδανική βελτιστοποίηση φαντάζει η δυνατότητα επιλογής μέσα από το περιβάλλον του web app, των επιθυμητών λειτουργιών ή εφαρμογών, με ταυτόχρονη διαμόρφωση χωρίς διαμεσολάβηση τόσο του προγραμματισμού του μικροελεγκτή, όσο και του ίδιου του περιβάλλοντος.

Συμπεράσματα

Η ιδέα του αυτοματισμού, ως εργαλείο παραγωγικότητας έχει εκτιμηθεί από την ανθρωπότητα εδώ και πολλά χρόνια. Η υστέρηση όμως της εισχώρησης τέτοιων τεχνολογιών στην καθημερινή - οικιακή ζωή, δείχνει κάποιες από τις αδυναμίες των υπαρχόντων τεχνολογιών. Η βασικότερη, κατά τη γνώμη μου, είναι το κόστος απόκτησης τέτοιων τεχνολογιών ή η δυσκολία απόκτησης της τεχνογνωσίας για την κατασκευή τους. Με το πέρασμα των χρόνων όμως έρχεται συνεχής πτώση των τιμών και παρουσίαση πλειάδας νέων προϊόντων αυτοματισμού και οι δύο αποτρεπτικοί παράγοντες χάνουν την ιδιότητά τους αυτή. Μια άλλη και σοβαρότερη πιθανή αιτία είναι αν όντως έχουν κάποια πραγματική αξία ή χρησιμότητα τέτοιες εφαρμογές. Σίγουρα μπορούν να συμβάλλουν στο ευ ζην, είναι όμως αυτό αρκετό;

Πλειάδα πληροφοριών μπορούν να συλλέγονται από έναν μικροελεγκτή και να αποθηκεύονται σε πίνακες βάσεων δεδομένων. Παραδείγματα αυτών είναι οι θερμοκρασίες του χώρου, η συχνότητα χρήσης κάποιων συσκευών, η κατανάλωση ρεύματος. Η χρησιμότητα αυτών είναι σίγουρα μεγαλύτερη από το προφανές. Η αξιοποίησή τους είναι που θα κάνει το "Έξυπνο Σπίτι" πραγματικά έξυπνο και τις ζωές των ενοίκων του, όχι μόνο ευκολότερες, αλλά οικονομικότερες και πιο οικολογικές και υγιεινές. Η επιτυχία κατά τη γνώμη μου τέτοιων προϊόντων έχει να κάνει με το βήμα παραπάνω που πρέπει να κάνουν οι κατασκευαστές του. Όχι δηλαδή να σου προβάλλουν την κατανάλωση του ρεύματος, αλλά να βελτιστοποιούν τις συσκευές και τις ώρες λειτουργίας τους για να μειώσουν την κατανάλωση αυτή. Επίσης όχι να μπορείς απλά να τηλεχειριστείς το σύστημα θέρμανσής σου, αλλά αυτό να μαθαίνει πότε το κάνεις και να το πράττει για σένα κ.ο.κ.

Πηγές

http://en.wikipedia.org/wiki/Web_server

http://en.wikipedia.org/wiki/Application_server

<http://arduino.cc/en/Reference/HomePage>

<http://forum.arduino.cc/>

<https://github.com/panique/php-login-minimal>

Δομή και Λειτουργία Μικροϋπολογιστών, Κ. Πεκμεστζή, Ι.Βογιατζής, Γ. Λιβιεράτος, Π. Μπουγάς, Οργανισμός Εκδόσεων Διδακτικών Βιβλίων Αθήνας, Έκδοση Δ΄ 2005

Τεκμηρίωση ATmega328p Microcontroller with 4/8/16/32K Bytes In-System

Programmable Flash, Έκδοση 8161D – AVR – 10/2009

Arduino Cookbook, Michael Margolis, O'REILLY, 2nd Edition