



**Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης**  
**Σχολή Τεχνολογικών Εφαρμογών**  
**Τμήμα Μηχανικών Πληροφορικής**

**Πτυχιακή Εργασία**  
Επιβλέπων : Παπαδάκης Νικόλαος  
Επίκουρος Καθηγητής

---

---

**Έλεγχος τάσης φορτίου μέσω ιστοσελίδας  
αναρτημένη από πλατφόρμα Arduino DUE**

---

---

**Λασηθιωτάκης Μιχαήλ-Άγγελος**  
**A.M.: 3402 E.E.:2010 ΧΕΙΜ**

Ηράκλειο, Ιούνιος 2015



## Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον κ. Παπαδάκη Νικόλαο για την εμπιστοσύνη που μου έδειξε αναθέτοντας το θέμα της πτυχιακής καθώς και την πολύτιμη καθοδήγησή του για την διεκπεραίωση της, μέσα στα χρονικά πλαίσια.

Θα θελα επίσης να απευθύνω τις ευχαριστίες μου στους γονείς μου Γιώργο και Κατερίνα καθώς και στον αδερφό μου Νίκο, οι οποίοι στήριξαν τις σπουδές μου και τις επιλογές μου, φροντίζοντας για την καλύτερη δυνατή μόρφωση μου.



## Περιεχόμενα

Ευχαριστίες .....	1
Abstract .....	5
Σύνοψη .....	6
Κεφάλαιο 1 <sup>ο</sup> .....	7
1.1 Εισαγωγή .....	7
1.2 Πρωτόκολλο IEEE 802.11 .....	8
1.3 Arduino IDE .....	10
1.3.1 Εγκατάσταση Arduino IDE .....	10
1.3.2 Βασικές λειτουργίες .....	12
1.4 Arduino Due .....	15
1.3.1 Βασικά χαρακτηριστικά .....	15
1.3.2 Θύρες Programming και Native .....	16
Κεφάλαιο 2 <sup>ο</sup> .....	18
2.1 Θεωρητική Ανάλυση .....	18
2.2 Υλοποίηση Συστήματος .....	19
2.2.1 Βιβλιοθήκες .....	19
2.2.2 Σύνδεση στο Δίκτυο .....	21
2.2.3 Διαθεσιμότητα της κάρτας SD .....	21
2.2.4 Έλεγχος κατάστασης Relay .....	22
2.2.5 Έλεγχος ασφαλείας ιστοσελίδας .....	23
2.2.6 Ιστοσελίδα .....	25
2.2.7 Αλλαγή κατάστασης με χρήση κουμπιών .....	28
2.2.8 Συνάρτηση StrClear( ) .....	29
2.2.9 Συνάρτηση GetRelayState( ) .....	30
2.2.10 Συνάρτηση GetSDState( ) .....	31
2.2.11 Σειριακή οθόνη .....	32
2.2.12 Συνδεσμολογία Συστήματος .....	33
Κεφάλαιο 3 <sup>ο</sup> .....	34
3.1 Αποτελέσματα .....	34
3.2 Συμπεράσματα .....	35
3.3 Μελλοντική Εργασία και Επεκτάσεις .....	38
3.4 Μικροελεγκτές .....	40
Raspberry Pi .....	40



Intel 8051 .....	40
LPC2000 .....	41
PicoBlaze .....	41
Βιβλιογραφία.....	42



## Πίνακας Εικόνων

Εικόνα 1 - Αποσυμπίεση Αρχείου.....	10
Εικόνα 2 - Διαχείριση Συσκευών .....	11
Εικόνα 3 - Κεντρική Κονσόλα Arduino IDE .....	11
Εικόνα 4 - Μεταγλώττιση .....	14
Εικόνα 5 - Μεταφόρτωση .....	14
Εικόνα 6 - Arduino Due Ports .....	16
Εικόνα 7 - Programming and Native Ports .....	17
Εικόνα 8 - Διάγραμμα Συστήματος.....	18
Εικόνα 9 - Arduino Due και Ethernet Shield .....	18
Εικόνα 10 - Relay 4 καναλιών .....	19
Εικόνα 11 - Περιεχόμενα κάρτας SD.....	22
Εικόνα 12 - Κρυπτογράφηση Base64 .....	24
Εικόνα 13 - Παράθυρο Πιστοποίησης Χρήστη.....	25
Εικόνα 14 - Μη πιστοποιημένη πρόσβαση .....	25
Εικόνα 15 - Ιστοσελίδα: κάρτα SD Συνδεδεμένη .....	26
Εικόνα 16 - Ιστοσελίδα: κάρτα SD Αποσυνδεδεμένη .....	27
Εικόνα 17 - Σειριακή Οθόνη Arduino.....	32
Εικόνα 18 - Σχηματικό Διάγραμμα Συνδεσμολογίας.....	33

## Λίστα Πινάκων

Πίνακας 1 - Βασικά Χαρακτηριστικά Arduino Due.....	15
Πίνακας 2 – Συνδεσμολογία.....	33



## Abstract

In the current dissertation, our concern is the programming the prototype platform of Arduino. It's a low-cost project which in conjunction with Arduino Ethernet Shield and the four channels Relay and SD card utilities using the necessary libraries.

Our aim is to control four electrical appliances via a Web Interface design by us posted on Local Network. It is required the preparation of programming code in C++, HTML and JavaScript on Arduino Integrated Development Environment (IDE) along with practical implementation. The development work is reliable and is applicable both in industrial and domestic level (Smart Houses). One of the main reasons for the implementation is the features that can provide the user from their computer or from their mobile. User can turn any device provided it is connected to our system, for example, a heater, a television, a computer or even the lights in our room.

It has its own security system giving access only to authorized persons to access the control panel. It features a safety valve in case of power failure by storing the last state of each device on the SD micro available, so when the power is restored to restore the devices to their correct state. It has the ability to be mounted on any network is necessary as it has extensibility, adding a Virtual Point Network (VPN), allowing access to the control panel from all sides.



## Σύνοψη

Η παρούσα πτυχιακή εργασία αφορά τον προγραμματισμό χαμηλού κόστους μικροεπεξεργαστών- πλατφόρμων , όπως η σειρά *Arduino* σε συνδυασμό με το *Arduino Ethernet Shield*, το *4 Channel Relay* και το *SD Shield* με την χρήση των απαραίτητων βιβλιοθηκών .

Σκοπός μας, είναι ο έλεγχος τεσσάρων ηλεκτρικών συσκευών μέσω διεπαφής ιστοσελίδας (Web Interface) σχεδιασμένη από εμάς αναρτημένη στο παρών τοπικό δίκτυο . Απαιτείται η σύνταξη προγραμματιστικού κώδικα (σε γλώσσα C++ ,C, HTML και Javascript) στο περιβάλλον *Arduino Integrated Development Environment (IDE)* σε συνδυασμό με την πρακτική υλοποίηση.

Η ανάπτυξη της εργασίας είναι αξιόπιστη και έχει εφαρμογή τόσο σε βιομηχανικό όσο και σε οικιακό επίπεδο (Smart Houses). Ένας από τους σημαντικότερους λόγους υλοποίησης της είναι οι δυνατότητες που μπορεί να παρέχει στον χρήστη από τον υπολογιστή του ή από το κινητό του. Μπορεί να ενεργοποιήσει οποιαδήποτε συσκευή αρκεί να είναι συνδεδεμένη στο σύστημα μας για παράδειγμα ένα θερμοσίφωνο, μία τηλεόραση, έναν υπολογιστή ή ακόμη τα φώτα στο χώρο μας.

Διαθέτει σύστημα ασφαλείας δίνοντας έτσι την πρόσβαση μόνο σε εξουσιοδοτημένα άτομα την πρόσβαση στον πίνακα ελέγχου. Διαθέτει δικλίδα ασφαλείας, σε περίπτωση διακοπής ρεύματος αποθηκεύοντας την τελευταία κατάσταση της κάθε συσκευής στην *micro SD* που διαθέτει, έτσι όταν επανέλθει το ρεύμα να επαναφέρει τις συσκευές στην σωστή τους κατάσταση. Έχει την δυνατότητα να στηθεί σε οποιοδήποτε δίκτυο είναι απαραίτητο καθώς έχει και την δυνατότητα επέκτασης, προσθέτοντας το σε ένα *Virtual Point Network (VPN)* δίνοντας την πρόσβαση στον πίνακα ελέγχου από παντού .



## Κεφάλαιο 1<sup>ο</sup>

### 1.1 Εισαγωγή

Οι πλατφόρμες Arduino κατασκευάζονται κυρίως από την εταιρία Smart Project. Ωστόσο, το Arduino ξεκίνησε ως έργο προς ανάπτυξη το 2005 στην Ιταλία, στο Ινστιτούτο Αλληλεπίδρασης Σχεδίασης Invea ώστε οι φοιτητές του Ινστιτούτου να αναπτύσσουν ενσωματωμένα συστήματα οικονομικά και αποδοτικά αξιοποιώντας τις δυνατότητες και τις ευκαιρίες που μπορεί να προσφέρει το ελεύθερο λογισμικό.

Γενικότερα, το Arduino θα λέγαμε ότι είναι ένα εργαλείο που μπορούμε να κατασκευάσουμε ένα υπολογιστικό σύστημα με την έννοια ότι αυτό θα ελέγχει συσκευές του φυσικού κόσμου, σε αντίθεση με τον κοινό Ηλεκτρονικό Υπολογιστή. Βασίζεται σε ευέλικτο, εύκολο στη χρήση υλικό και λογισμικό, σε μια αναπτυξιακή πλακέτα που ενσωματώνει επάνω έναν μικροελεγκτή και συνδέεται με τον Η/Υ για να προγραμματιστεί μέσα από ένα απλό περιβάλλον ανάπτυξης Arduino IDE. Με το Arduino δημιουργούνται συσκευές οι οποίες εξυπηρετούν διάφορους σκοπούς έχοντας την δυνατότητα να δέχονται ερεθίσματα από το περιβάλλον του και να αντιδρούν ανάλογα με το πως έχουν προγραμματιστεί. Τα παραπάνω δεν ακούγονται πρωτότυπα. Υπάρχουν και άλλες πλατφόρμες και υλοποιήσεις που μπορούν να κάνουν τα ίδια πράγματα. Ποια είναι η ειδοποιός διαφορά; Το Arduino βασίζεται σε τεχνολογίες ανοιχτού κώδικα. Μπορεί να κατασκευαστεί από τον καθένα, μπορεί να ενσωματωθεί σε συσκευές ακόμα και για εμπορικούς σκοπούς και το σημαντικότερο είναι ότι υπάρχει μια ολόκληρη κοινότητα που χρησιμοποιεί το Arduino σε κατασκευές άρα υπάρχει μεγάλος όγκος ελεύθερης πληροφορίας. Γενικά, τα Projects στον εν λόγω Μικροελεγκτή μπορούν να είναι αυτόνομα (σε επίπεδο hardware) ή να επικοινωνούν με κάποιο software στον Η/Υ του προγραμματιστή. Το Arduino χρησιμοποιεί τώρα ένα ειδικά προγραμματιζόμενο SAM3X8E ώστε αυτό να επιτρέπει τόσο την πιο γρήγορη ταχύτητα μεταφοράς όσο και τη γρήγορη σειριακή επικοινωνία.

Ο μικροεπεξεργαστής ενός Arduino συνήθως προγραμματίζεται εκ των προτέρων ώστε να παρέχει κάποιο φορτωτή εκκίνησης (BootLoader). Ο φορτωτής εκκίνησης υπάρχει ώστε να απλοποιεί την διαδικασία της αποθήκευσης των προγραμμάτων στην Flash Memory του Arduino μέσω σειριακής USB θύρας.

Επιπλέον, η γλώσσα προγραμματισμού, οι διάφορες βιβλιοθήκες και το ολοκληρωμένο περιβάλλον ανάπτυξης που υπάρχουν για τον προγραμματισμό της πλατφόρμας Arduino αποτελούν ανοιχτό λογισμικό προσφέροντας έτσι γνώση σε όλους.

Βασικά Πλεονεκτήματα πλατφόρμας Arduino:

- Οικονομική: Η πλατφόρμα Arduino αποτελεί οικονομική λύση διότι είναι φθηνότερη. Επιπλέον, είναι αρχιτεκτονικά ανοιχτή και μπορεί ο οποιοσδήποτε να την αναπτύξει από μόνος του.
- Μεταφέρσιμη: Σε σχέση με τις υπάρχουσες πλατφόρμας στο εμπόριο η πλατφόρμα Arduino παρέχει πλήρη μεταφερσιμότητα με αποτέλεσμα να μπορεί να προγραμματιστεί στα περισσότερα λειτουργικά συστήματα.





- Επεκτάσιμη: Το υλικό και το λογισμικό της πλατφόρμας Arduino είναι ανοιχτά και ελεύθερα για όλους. Καθημερινά, χιλιάδες υποστηρικτές του ελεύθερου λογισμικού αναπτύσσουν διάφορες βιβλιοθήκες για την υποστήριξη της πλατφόρμας. Παράλληλα, τόσο η αρχιτεκτονική όσο και το υλικό της πλατφόρμας εξελίσσονται συνεχώς.

### 1.2 Πρωτόκολλο IEEE 802.11

Η οικογένεια ασύρματων προτύπων IEEE 802.11 έχει γνωρίσει χαρακτηριστική αποδοχή και ανάπτυξη τα τελευταία χρόνια. Συνήθως αποκαλείται WiFi ή πιο απλά WLAN. Τα πλεονεκτήματα της είναι αρκετά. Βασικότερο αυτών είναι ικανοποίηση της ανάγκης που υπήρχε και υπάρχει για αντικατάσταση της παραδοσιακής καλωδιακής δικτύωσης των σπιτιών, των γραφείων και άλλων χώρων όπως οι καφετέριες κ.τ.λ. Ένα άλλο πλεονέκτημα είναι το χαμηλό κόστος των προϊόντων που είναι συμβατά με τα πρότυπα 802.11, όπως τα σημεία πρόσβασης (Access Points) και οι επαναλήπτες σήματος (Repeaters). Επίσης, τα πιο πρόσφατα πρότυπα της οικογένειας IEEE802.11 παρέχουν αρκετά μεγάλες ταχύτητες μετάδοσης δεδομένων. Για παράδειγμα το πρότυπο 802.11n μπορεί να επιτύχει ρυθμό μετάδοσης δεδομένων πάνω από 500 Mbps.

Η διάδοση αυτής της τεχνολογίας έφερε στο χρήστη εύκολη και πολλές φορές δωρεάν πρόσβαση στο διαδίκτυο σε διάφορους δημόσιους χώρους. Τέλος, τα πρότυπα IEEE 802.11 λειτουργούν και αυτά στο μη δανειοδοτημένο φάσμα ISM (Industrial Scientific Medical) στη συχνότητα των 2.4 GHz. Η τοπολογία δικτύου που χρησιμοποιείται είναι «κεντροποιημένη». Αυτό πρακτικά σημαίνει ότι όλοι οι πελάτες (clients) ενός δικτύου συνδέονται σε μια κεντρική οντότητα, που συχνά αναφέρεται ως σημείο πρόσβασης.

Αυτή η οντότητα λαμβάνει δεδομένα από τους πελάτες και κατόπιν τα δρομολογεί. Σήμερα γίνεται προσπάθεια παροχής δικτύων «mesh». Με ένα τέτοιο δίκτυο οι συσκευές δε θα χρειάζονταν μια κεντρική οντότητα. Αντιθέτως, θα δρουν αυτόνομα προωθώντας τα αντίστοιχα δεδομένα μεταξύ τους. Πρέπει να σημειωθεί εδώ ότι οι ανάγκες σε ενέργεια για τις διάφορες συσκευές με κάρτες που υποστηρίζουν τα πρότυπα IEEE 802.11 κυμαίνονται μεταξύ 2.06W και 0.14W όταν αυτές είναι σε αναμονή. Το τελευταίο αυτό χαρακτηριστικό αποτελεί και το βασικότερο μειονέκτημα της συγκεκριμένης τεχνολογίας όσον αφορά την ευρεία χρησιμοποίησή της στα έξυπνα σπίτια. Και αυτό γιατί οι αισθητήρες που αποτελούν βασικό συστατικό ενός έξυπνου σπιτιού δεν πρέπει να καταναλώνουν μεγάλα ποσά ενέργειας.

Άλλο ζήτημα είναι η ασφάλεια των δεδομένων. Τα πρωτόκολλα 802.11 περιλαμβάνουν έναν προαιρετικό μηχανισμό πιστοποίησης κόμβων (μόνο για δομημένα δίκτυα) και κρυπτογράφησης δεδομένων, ονόματι WEP, ο οποίος λειτουργεί ως εξής: μετά τη συσχέτιση ενός σταθμού σε ένα BSS ο κόμβος στέλνει στο AP ένα αίτημα πιστοποίησης ταυτότητας. Το AP απαντά στέλνοντας του ένα τυχαίο κείμενο το οποίο ο κόμβος κρυπτογραφεί, χρησιμοποιώντας ένα κλειδί που έχει ρυθμιστεί από τους χρήστες του τοπικού δικτύου και είναι κοινό σε όλους τους κόμβους, και στέλνει πίσω στο σημείο πρόσβασης. Το AP επιβεβαιώνει ότι το κείμενο που έλαβε είναι η ορθά κρυπτογραφημένη, σύμφωνα με το σωστό κλειδί, εκδοχή αυτού που έστειλε και απαντά με μια επιβεβαίωση πιστοποίησης η οποία αναθέτει μία ταυτότητα στον κόμβο (η οποία γίνεται γνωστή και στους άλλους κόμβους). Από εκείνη τη στιγμή κι έπειτα ο σταθμός μπορεί να ανταλλάσσει δεδομένα στο WLAN με την ταυτότητα αυτή.



Όταν ο κόμβος επιθυμεί να αποχωρήσει από το δίκτυο στέλνει ένα αίτημα πιστοποίησης πριν την συσχέτιση. Για την αποφυγή υποκλοπών τα δεδομένα που διακινούνται στο δίκτυο κρυπτογραφούνται με ένα κρυφό κλειδί 40-bit (το ίδιο με αυτό της πιστοποίησης), το οποίο συνδυάζεται με μια γεννήτρια bit και το αποτέλεσμα συνδυάζεται με τα δεδομένα μέσω λογικού XOR (αποκλειστικό Ή) για να επιτευχθεί η κρυπτογράφηση. Στον παραλήπτη ακολουθείται η αντίστροφη διαδικασία.

Εκτός αυτών των εκδόσεων έχουν προταθεί και κάποιες επεκτάσεις τους, οι οποίες όμως δεν έχουν υλοποιηθεί σε εμπορικά προϊόντα και έχουν περισσότερο ακαδημαϊκό ενδιαφέρον. Οι σπουδαιότερες είναι:

- 802.11f ή **IAPP**, το οποίο επιτρέπει άμεση επικοινωνία μεταξύ διαφορετικών AP ώστε να εξαλειφθεί η απώλεια πλαισίων κατά τη μεταγωγή. Ο σχετικός μηχανισμός ενεργοποιείται από ένα αίτημα επανασυσχέτισης.
- 802.11e ή **QoS** το οποίο προσπαθεί να διασφαλίσει ποιότητα υπηρεσιών για εφαρμογές πραγματικού χρόνου που εκτελούνται πάνω σε ένα WLAN ελαχιστοποιώντας ή μεγιστοποιώντας ένα από τα παρακάτω κριτήρια: μέση καθυστέρηση από άκρο σε άκρο, μέση μεταβολή της καθυστέρηση ή μέσο ποσοστό επιτυχούς παράδοσης πλαισίων. Αυτό το επιτυγχάνει βελτιώνοντας τους μηχανισμούς DCF και PCF με τους μηχανισμούς EDCAF, ο οποίος αναθέτει προτεραιότητες στα πλαίσια δεδομένων ανάλογα με το πόσο χρονικά κρίσιμη είναι η παράδοση τους και με τα μεγαλύτερης προτεραιότητας πλαίσια να έχουν περισσότερες πιθανότητες να κερδίσουν στον ανταγωνισμό για την πρόσβαση στο κοινό μέσο, και HCF, ο οποίος περιορίζει το μέγιστο χρόνο δέσμευσης του καναλιού από ένα τερματικό, αντίστοιχα.
- 802.11n, το οποίο με χρήση πολλαπλών κεραιών (μέθοδος γνωστή ως **MIMO**, εκ του Multiple Inputs Multiple Outputs) αναμένεται να παρέχει ονομαστικό ρυθμό μετάδοσης τουλάχιστον 108 Mbps. Σε αντίθεση με τα δύο προηγούμενα πρόκειται να τυποποιηθεί σύντομα και να κυκλοφορήσουν εμπορικά προϊόντα βασισμένα σε αυτό. Μάλιστα κάρτες ασύρματης δικτύωσης συμβατές με το 802.11n έχουν ήδη βγει στην αγορά από ορισμένους προμηθευτές, χωρίς να έχει οριστικοποιηθεί ακόμα το επίσημο πρότυπο (αναμένεται στα τέλη του 2009).



## 1.3 Arduino IDE

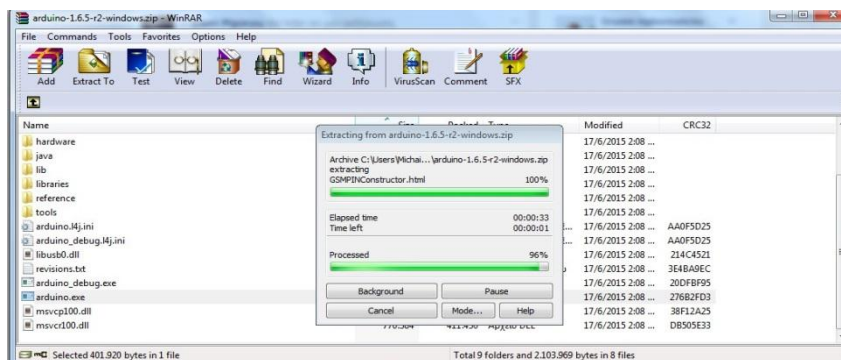
Το περιβάλλον ανάπτυξης του Arduino (IDE) έχει συγγραφεί με την γλώσσα προγραμματισμού JAVA και αυτό το καθιστά μεταφέρσιμο στα περισσότερα λειτουργικά συστήματα. Το IDE του Arduino περιέχει έναν έξυπνο συντάκτη, μεταγλωττιστή της C, C++, τερματικό για σειριακή επικοινωνία με το Arduino.

Πιο συγκεκριμένα, η γλώσσα προγραμματισμού που χρησιμοποιείται για την συγγραφή προγραμμάτων στο Arduino είναι η Wiring (C, C++). Το IDE του Arduino χρησιμοποιεί εργαλεία GNU Toolchain και AVR Libc για να παρέχει την μεταγλώττιση προγραμμάτων από C, C++ σε κατάλληλες AVR εντολές γλώσσας μηχανής, καθώς και το εργαλείο avrdude για την αποστολή του εκτελέσιμου προγράμματος στην Flash memory του Arduino.

Η ψηφιακή σχεδίαση του υλικού μέρους του Arduino είναι ανοιχτή και προσβάσιμη από όλους μια και είναι δημοσιευμένη υπό την άδεια Creative Commons Attribution Share - Alike 2.5. Επίσης, το περιβάλλον ανάπτυξης (IDE) του Arduino είναι ελεύθερο λογισμικό και είναι δημοσιευμένο υπό την άδεια GNU General Public License Version 2.

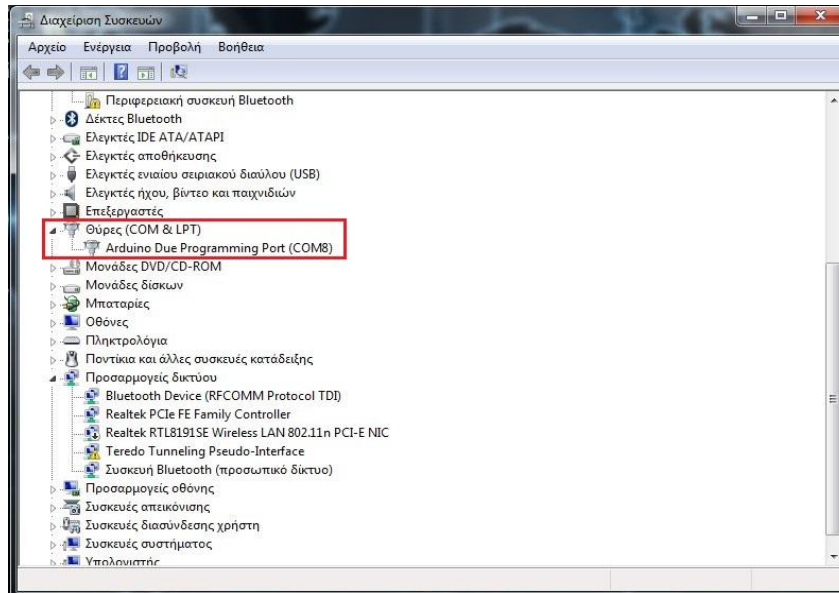
### 1.3.1 Εγκατάσταση Arduino IDE

Μπορούμε να εγκαταστήσουμε το Arduino IDE από την επίσημη ιστοσελίδα της εταιρίας ([Arduino.cc](http://Arduino.cc)). Η έκδοση που θα χρησιμοποιήσουμε στη παρούσα φάση είναι η 1.6.5 για λειτουργικό σύστημα των Windows 7, ενώ είναι διαθέσιμο και για Mac OS X αλλά και για Linux. Η εγκατάσταση είναι απλή πηγαίνουμε στις λήψεις και επιλέγουμε την κατάλληλη έκδοση για το σύστημα μας. Στην συνέχεια αποσυμπιέζουμε το αρχείο (\*.zip) σε φάκελο της επιλογής μας.



Εικόνα 1 - Αποσυμπίεση Αρχείου

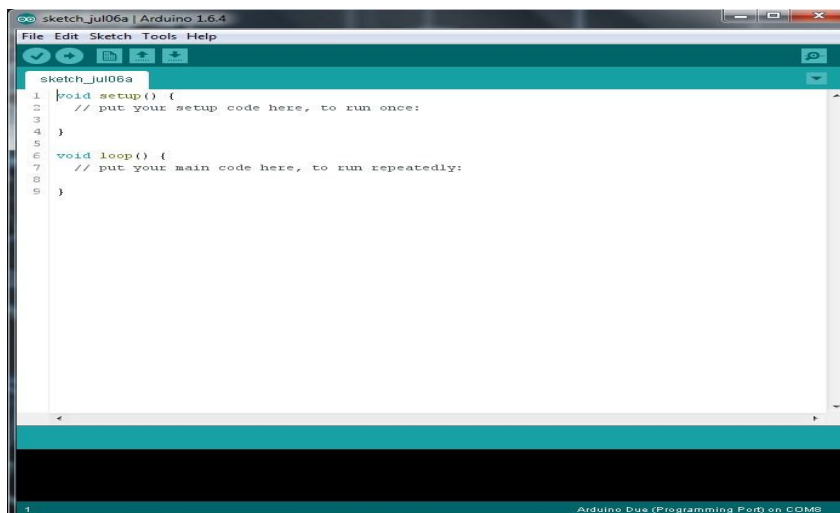
Όταν η διαδικασία ολοκληρωθεί ανοίγουμε τον φάκελο και δημιουργούμε ένα εικονίδιο συντόμευσης του Arduino IDE στην επιφάνεια εργασίας (Εικόνα 1). Στη συνέχεια ανοίγουμε την διαχείριση συσκευών του υπολογιστή και συνδέουμε την πλακέτα που διαθέτουμε (π.χ. Arduino Due) σε μια σειριακή θύρα, USB (Εικόνα 2).



Εικόνα 2 - Διαχείριση Συσκευών

Το Arduino θα εμφανιστεί σαν άγνωστη συσκευή. Στην συνέχεια από τις επιλογές, επιλέγουμε ενημέρωση προγράμματος οδήγησης επιλέγουμε να βρούμε τους drivers χειροκίνητα. Στο φάκελο που αποσυμπιέσαμε, επιλέγουμε τον υποφάκελο drivers και συνεχίζουμε την διαδικασία εγκατάστασης.

Μόλις ολοκληρωθεί βλέπουμε πως ο υπολογιστής αναγνωρίζει πλέον το Arduino και το έχει προσθέσει στις θήρες COM και LPT δίνοντας του την δική του COM (π.χ. COM 8). Μπορούμε τώρα να ανοίξουμε το πρόγραμμα του Arduino.



Εικόνα 3 - Κεντρική Κονσόλα Arduino IDE

Από το menu του προγράμματος επιλέγουμε την καρτέλα Tools > Board και διαλέγουμε την πλακέτα με την οποία εργαζόμαστε. Εμείς θα χρησιμοποιήσουμε την πλακέτα Arduino Due (Programming Port). Προς αποφυγή σφαλμάτων και λαθών, ελέγχουμε αν η θύρα, COM, που μας έχει δοθεί από τον υπολογιστή αναγράφεται στο Tools > Port.

Το πρόγραμμα έρχεται με μία σειρά παραδειγμάτων που μπορούν να βοηθήσουν το κάθε χρήστη στο να φτιάξει το δικό του εκτελέσιμο πρόγραμμα. Τα παραδείγματα συνδέονται



άμεσα με τις βιβλιοθήκες τις οποίες καλούν. Ένα από τα μεγαλύτερα πλεονεκτήματα είναι ότι μπορούμε να χρησιμοποιήσουμε και βιβλιοθήκες άλλων χρηστών, οι οποίες είναι κατάσκευασμένες για κάποιο σκοπό. Έτσι, μπορούμε να επεκτείνουμε τα project μας με την χρήση custom libraries.

### 1.3.2 Βασικές λειτουργίες

Μετά την εγκατάσταση του Arduino IDE μπορούμε να γράψουμε τα πρώτα μας τμήματα κώδικα. Η λογική του Arduino είναι πολύ απλή - στην ουσία υπάρχουν δύο βασικές συναρτήσεις, η *setup()* και η *loop()* οι οποίες δουλεύουν ως εξής:

- *setup()* - εδώ βάζουμε όλες τις εντολές που πρέπει να τρέξουν μία φορά, όταν ενεργοποιείται η μονάδα μας (όταν δηλαδή δίνουμε ρεύμα ή όταν πατηθεί το πλήκτρο reset που υπάρχει). Συνήθως μπαίνουν αρχικοποιήσεις τιμών μεταβλητών και οπωσδήποτε ο χαρακτηρισμός των εισόδων/εξόδων που θα χρησιμοποιήσουμε (αν δηλαδή ένα συγκεκριμένο Pin θα είναι είσοδος ή έξοδος).

- *loop()* - εδώ γράφουμε το πρόγραμμά μας. Οι εντολές που υπάρχουν θα τρέξουν κι όταν φτάσει στο τέλος θα ενεργοποιηθεί ξανά η *loop()*, συνεχίζοντας από την αρχή της, και ξανά. Αυτό θα συμβαίνει συνεχώς, όσο έχει ρεύμα το Arduino ή μέχρι να πατηθεί το πλήκτρο reset.

Έτσι, η βασική λειτουργία του Arduino είναι ότι τρέχει η συνάρτηση *setup()* μία φορά στην αρχή και ακολούθως η *loop()* ξανά και ξανά μέχρι να το κλείσουμε (να μην τροφοδοτείται με ρεύμα) ή να πατήσουμε το πλήκτρο reset. Στην περίπτωση του Reset ξανατρέχει η συνάρτηση *setup()* μία φορά και ακολούθως η *loop()* ξανά και ξανά, όπως δηλαδή ακριβώς και όταν αρχικά ενεργοποιείται με ρεύμα ο μικροελεγκτής. Στην περίπτωση που έχουμε κάνει αλλαγές στο πρόγραμμά μας και το φορτώσουμε στον μικροελεγκτή αρκεί να πατήσουμε το πλήκτρο Reset ώστε να φορτώσει το πρόγραμμά μας από την αρχή με τον τρόπο που περιγράφηκε.

### Μεταβλητές

Όπως σε όλες τις γλώσσες προγραμματισμού, μπορούμε να δηλώσουμε ονόματα μεταβλητών. Οι τύποι μεταβλητών που υποστηρίζονται στο Arduino είναι αρκετοί. Για έναν χρήστη οι παρακάτω τύποι θα είναι αρκετοί:

- *boolean*, με τιμές το 0 και 1 (ή True – False)
- *byte*, με τιμές από 0 έως και 255
- *int*, ακέραιος με δυνατές τιμές από -32768 έως και 32767
- *long*, ακέραιος με δυνατές τιμές από -2147483648 έως και 2147483647
- *float*, δεκαδικοί αριθμοί
- *char*, ένας χαρακτήρας (μέγεθος ένα Byte)
- *string*, πίνακας χαρακτήρων



## **Συναρτήσεις διαχείρισης θυρών εισόδου – εξόδου**

Όπως αναφέρθηκε, η κύρια λειτουργία του μικροελεγκτή βασίζεται στο να ελέγχει τις θύρες που διαθέτει και είτε να δίνει ρεύμα είτε να παίρνει ρεύμα από αυτές. Στην αρχικοποίηση κάθε προγράμματος (μέσα στη συνάρτηση setup) θα χρειαστεί να χαρακτηρίσουμε τα Pins που χρησιμοποιούμε ως είσοδο ή ως έξοδο.

Η συνάρτηση pinMode (Pin, Mode) χρησιμοποιείται με το όνομά της και ορίσματα α) τον αριθμό Pin και β) την κατάσταση λειτουργίας που χαρακτηρίζεται με τη λέξη INPUT (είσοδος) ή OUTPUT (έξοδος).

## **Ψηφιακή είσοδος – έξοδος**

Τα 14 ψηφιακά pins του Arduino μπορούν δουλεύουν ως ψηφιακές εισοδοι, δηλαδή να “διαβάσουν” ως είσοδο τάση με τιμή είτε 0 είτε 5V. Αυτό γίνεται με χρήση της συνάρτησης digitalRead(Pin), όπου το όρισμα Pin αναφέρεται στο νούμερο της θύρας για την οποία θα πάρουμε είσοδο, ενώ η συνάρτηση επιστρέφει με το όνομά της την τιμή εισόδου. Η τάση εισόδου μπορεί να είναι 0V ή 5V, οι οποίες αναπαρίστανται με προκαθορισμένες τιμές στην τιμή που διαβάζουμε:

- LOW : όταν λάβει τάση 0 V στην είσοδο (pin)
- HIGH : όταν λάβει τάση 5 V στην είσοδο (pin)

Και τα 14 pins του Arduino μπορούν δουλεύουν ως ψηφιακές εξοδοι, δηλαδή δίνουν έξοδο 0 ή 5V. Αυτό γίνεται με χρήση της συνάρτησης digitalWrite(Pin, Value), όπου το όρισμα Pin αναφέρεται στο νούμερο της θύρας για την οποία θα δώσουμε τάση εξόδου, ενώ η τάση εξόδου μπορεί να είναι 0 V ή 5 V, οι οποίες αναπαρίστανται με προκαθορισμένες τιμές στην παράμετρο value

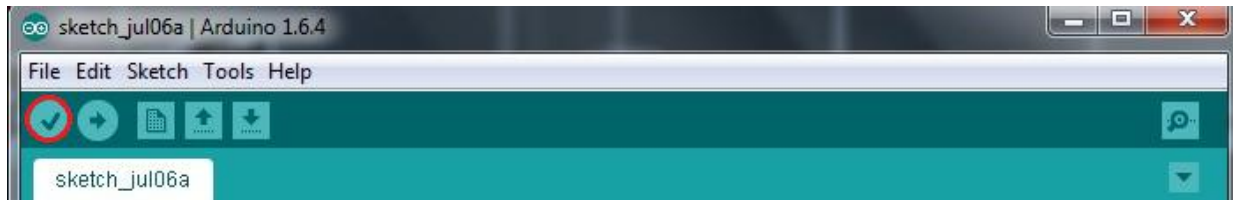
- LOW : θα δώσει 0 V στην έξοδο (pin)
- HIGH : θα δώσει 5 V στην έξοδο (pin)

## **Μεταφόρτωση προγράμματος**

Για να μεταφορτώσουμε το πρόγραμμά μας στη μονάδα θα πρέπει να τη συνδέσουμε με ένα USB καλώδιο στον υπολογιστή. Ο υπολογιστής μας θα αναγνωρίσει τη μονάδα Arduino μας ως σειριακή θύρα, κάτι που μπορείτε να επιβεβαιώσετε και από τον πίνακα ελέγχου του Η/Υ σας. Από το μενού Εργαλεία του Arduino επιλέγουμε δύο πράγματα:

- Πλακέτα - διαλέγουμε τον τύπο της μονάδας μας. π.χ. Arduino Due (Programming Port).
- Σειριακή θύρα - είναι η σειριακή θύρα που έχει αντιστοιχίσει το λειτουργικό σας στην πλακέτα Arduino που συνδέεται μέσω του USB καλωδίου. Αν χρησιμοποιείτε Windows αυτή θα είναι της μορφής COMX ενώ στο Linux η θύρα θα εμφανιστεί ως /dev/ttyXXX.





Εικόνα 4 - Μεταγλώττιση

Ακολούθως, πατάμε το πλήκτρο της μεταγλώττισης το οποίο θα ελέγξει το πρόγραμμά μας για λάθη και θα το προετοιμάσει για τη μεταφόρτωση στην πλακέτα. Αν τυχόν υπάρξουν λάθη, αυτά εμφανίζονται με μορφή μηνυμάτων με κόκκινο χρώμα στο κάτω μέρος της οθόνης (Εικόνα 4).



Εικόνα 5 - Μεταφόρτωση

Τέλος, εφόσον έχουμε επιτυχώς εκτελέσει όλα τα παραπάνω, δηλαδή έχουμε συνδέσει τη μονάδα μας, έχουμε επιλέξει τον τύπο της και τη θύρα που είναι συνδεδεμένη, έχουμε γράψει κάποιο πρόγραμμα και το έχουμε μεταγλωττίσει χωρίς λάθη, μπορούμε πατώντας το πλήκτρο της φόρτωσης να μεταφορτώσουμε το πρόγραμμα πλέον στη μονάδα και αυτό να αρχίσει να τρέχει πλέον σε πραγματικό περιβάλλον. Η διαδικασία της μεταγλώττισης επαναλαμβάνεται αυτόματα στο βήμα αυτό (Εικόνα 5).



## 1.4 Arduino Due

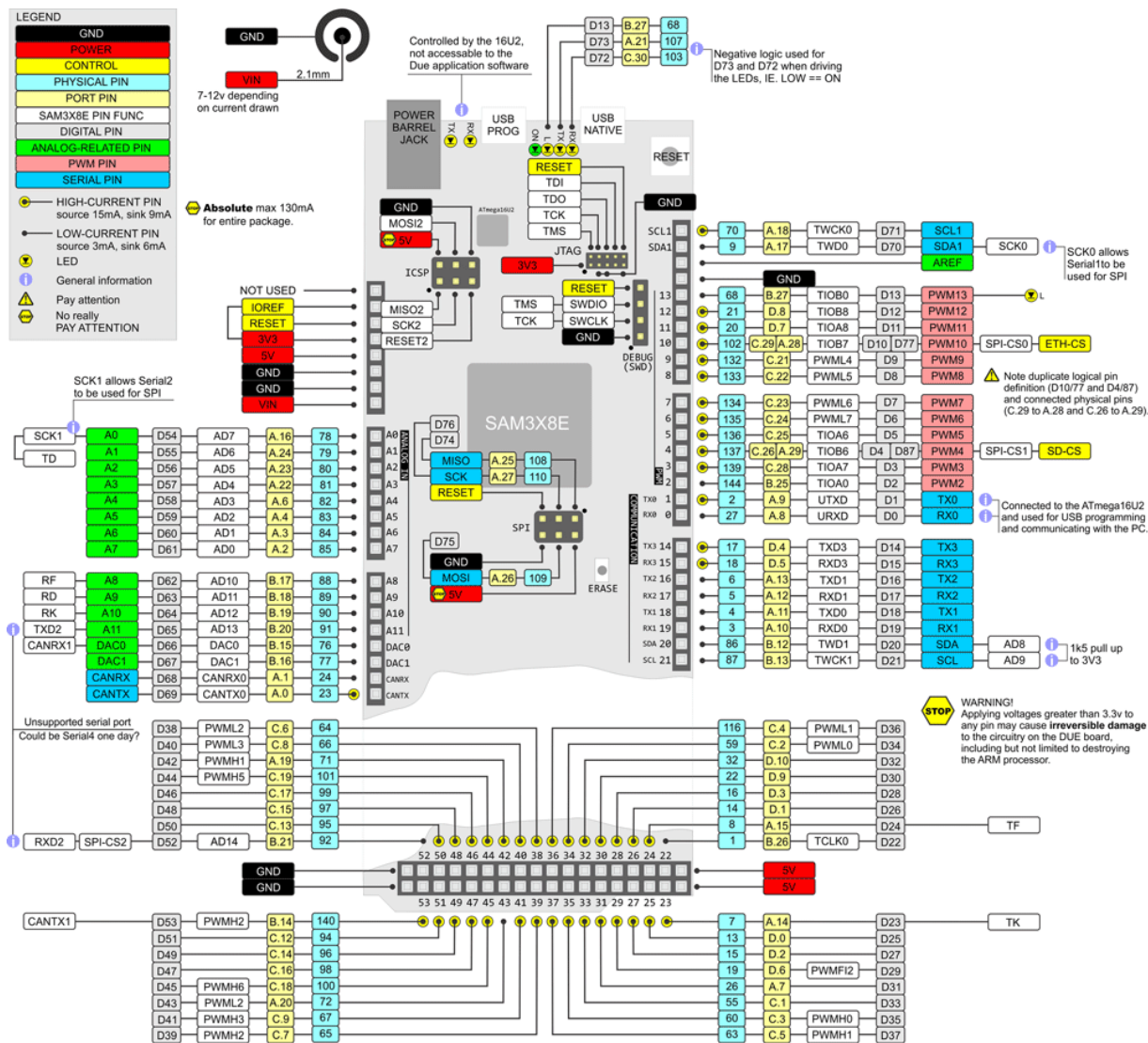
Τον Οκτώβριο του 2012 ανακοινώθηκε το Arduino Due. Είναι βασισμένο στο Atmel SAM3X8E, που είχε πυρήνα ARM Cortex-M3. Η ARM είναι μια αρχιτεκτονική συνόλου εντολών RISC των 32-bit, η οποία έχει αναπτυχθεί από την ARM Holdings. Τα αρχικά σημαίνουν Προχωρημένη Μηχανή RISC (Advanced RISC Machine), ενώ παλαιότερα σήμαιναν Μηχανή RISC Acorn (Acorn RISC Machine). Η αρχιτεκτονική ARM είναι η πιο συχνά χρησιμοποιούμενη αρχιτεκτονική συνόλου εντολών 32-bit όσον αφορά τους επεξεργαστές που παράγονται. Η αρχιτεκτονική δημιουργήθηκε αρχικά από την Acorn Computers για χρήση στους προσωπικούς υπολογιστές της, με τα πρώτα προϊόντα που βασίζονταν στην αρχιτεκτονική ARM να είναι οι σειρά Acorn Archimedes που εμφανίστηκε το 1987. Οι επεξεργαστές ARM είναι σχετικά απλοί, κάτι που τους κάνει κατάλληλους για εφαρμογές χαμηλής ισχύος. Αυτό έχει ως αποτέλεσμα να έχουν υπερισχύσει στις αγορές των κινητών και των ενσωματωμένων συστημάτων, σαν μικροί και σχετικά χαμηλού κόστους μικροεπεξεργαστές και μικροελεγκτές (Εικόνα 6).

### 1.3.1 Βασικά χαρακτηριστικά

Μικροελεγκτής	AT91SAM3X8E
Τάση λειτουργίας	3.3V
Τάση εισόδου (συνιστάται)	7-12V
Τάση εισόδου (όρια)	6-16V
Ψηφιακές I/O Pins	54 (εκ των οποίων 12 παρέχουν PWM έξοδο)
Αναλογικές Είσοδοι Pins	12
Αναλογικές Έξοδοι Pins	2 (DAC)
Συνολικό ρεύμα εξόδου (DC) σε όλες τις γραμμές I/O	130 mA
DC ρεύμα για 3.3V Pin	800 mA
DC ρεύμα για 5V Pin	800 mA
Flash Memory	512 KB διαθέσιμα για τις εφαρμογές του χρήστη
SRAM	96 KB (δύο τράπεζες : 64KB and 32KB)
Ταχύτητα Ρολογιού	84 MHz
Μήκος	101.52 mm
Πλάτος	53.3 mm
Βάρος	36 g

Πίνακας 1 - Βασικά Χαρακτηριστικά Arduino Due





Εικόνα 6 - Arduino Due Ports

### 1.3.2 Θύρες Programming και Native

Και οι δύο θύρες μπορούν να χρησιμοποιηθούν για τον προγραμματισμό της πλακέτας αν και συνιστάται να χρησιμοποιήσετε τη θύρα προγραμματισμού λόγω του τρόπου χειρισμού διαγραφής του chip (Εικόνα 7):

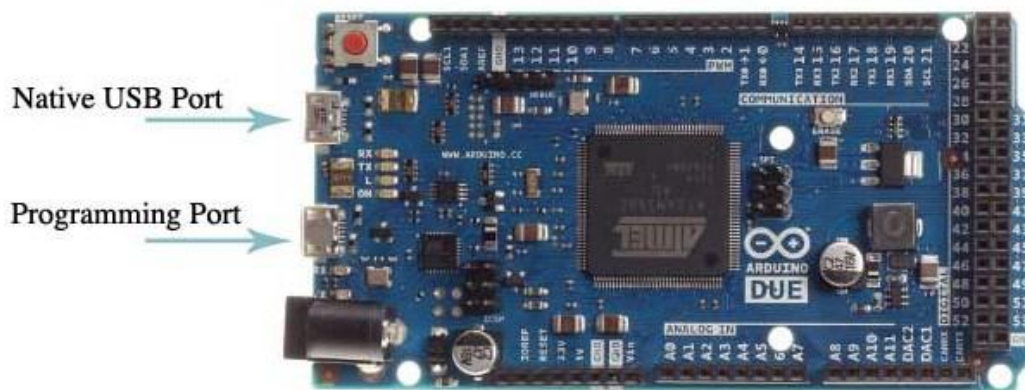
**Θύρα Programming:** Για να χρησιμοποιήσουμε αυτή την θύρα, επιλέγουμε "Arduino Due (Programming Port)" ως την πλακέτα μας στο Arduino IDE (Tools > Board). Συνδέουμε την θύρα τις πλακέτας (είναι η πλησιέστερη στην τροφοδοσία DC ) με τον υπολογιστή. Η θύρα προγραμματισμού χρησιμοποιεί το 16U2 ως USB σε σειριακή συνδέοντας το με το πρώτο UART του SAM3X (RX0 και TX0). Η 16U2 έχει δύο ακροδέκτες που συνδέονται με το Reset και το Delete του SAM3X. Ανοίγοντας και κλείνοντας την θύρα προγραμματισμού συνδέεται στα 1.200 hrs πυροδοτώντας μια «οριστική διαγραφή» στο chip του SAM3X, την ενεργοποιώντας την διαγραφή και την επαναφορά στους ακροδέκτες του SAM3X πριν από την επικοινωνία με το UART. Αυτή είναι η συνιστώμενη θύρα για τον προγραμματισμό του Due.



Είναι πιο αξιόπιστο από την «ομαλή διαγραφή» που εμφανίζεται στη θύρα Native και θα πρέπει να λειτουργεί ακόμη και αν η κύρια MCU έχει καταρρεύσει.

**Θύρα Native:** Για να χρησιμοποιήσουμε αυτή την θύρα, επιλέγουμε "Arduino Due (Native Port)" ως την πλακέτα μας στο Arduino IDE. Η θύρα Native συνδέεται απευθείας με τον SAM3X. Συνδέουμε την θύρα τις πλακέτας (είναι η πλησιέστερη προς το κουμπί *reset*) με τον υπολογιστή. Ανοίγοντας και κλείνοντας την θύρα Native συνδέεται στα 1.200 bps πυροδοτώντας μια «ομαλή διαγραφή» στη flash memory της πλακέτας, ενώ η πλακέτα εκτελεί επανεκκίνηση με τον bootloader. Ένα η MCU καταρρεύσει για κάποιο λόγο είναι πιθανό ότι η διαδικασία διαγραφής δεν θα λειτουργήσει μιας και η εκτέλεση έγινε εξ ολοκλήρου σε λογισμικό πάνω στον SAM3X. Ανοίγοντας και κλείνοντας την θύρα Native σε διαφορετικό ρυθμό baud δεν θα επαναφέρει τον SAM3X.

Σε αντίθεση με άλλες πλακέτες Arduino που χρησιμοποιούν avrdude για το φόρτωμα, τα παραπάνω βασίζονται σε bossac.



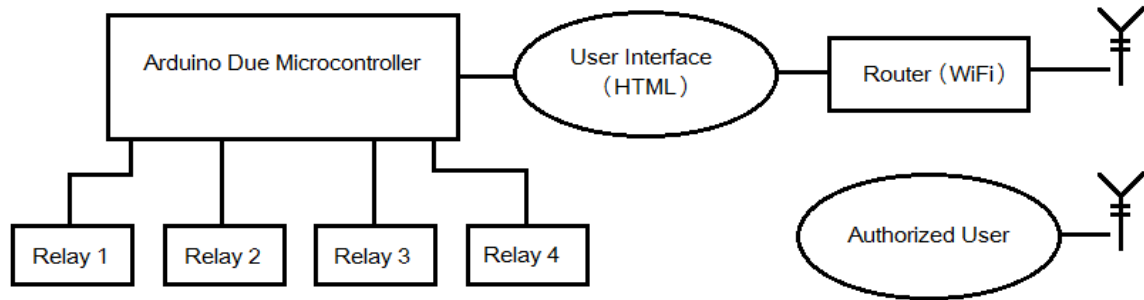
Εικόνα 7 - Programming and Native Ports



## Κεφάλαιο 2<sup>ο</sup>

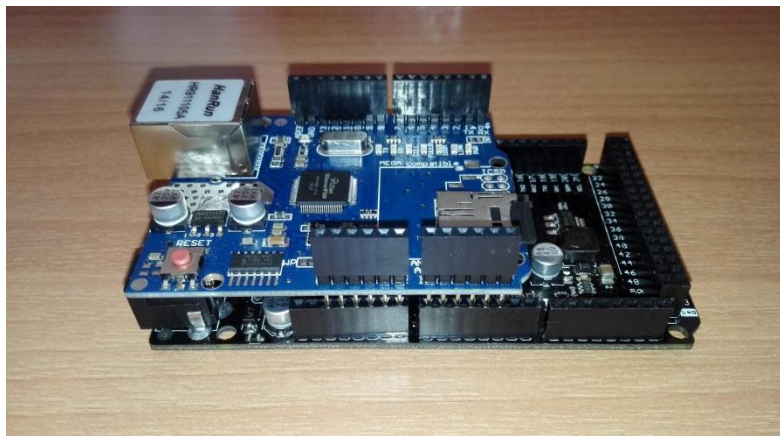
### 2.1 Θεωρητική Ανάλυση

Μία προσέγγιση για την παρούσα πτυχιακή εργασία είναι η υλοποίηση ενός αυτοματισμού για τον απομακρυσμένο έλεγχο συσκευών σε ένα οικιακό περιβάλλον ή μία επιχείρηση με απαραίτητη προϋπόθεση την πρόσβαση στο internet. Το διάγραμμα που θα μπορούσε να περιγράψει ένα τέτοιο σύστημα είναι το εξής:



Εικόνα 8 - Διάγραμμα Συστήματος

Κύριος σκοπός μας είναι η σύνταξη προγραμματιστικού κώδικα με σκοπό την δημιουργία κατάλληλης διεπαφής χρήστη-υπολογιστή. Αρχικά, στήνουμε τον μικροελεγκτή σε ένα τοπικό δίκτυο, αυτό πρακτικά σημαίνει ότι το Arduino μας έχει πρόσβαση στο δίκτυο μας. Για να το πετύχουμε αυτό προσθέτουμε την αντίστοιχη Ethernet Shield πάνω στον κεντρικό μικροελεγκτή Arduino Due (Εικόνα 9).

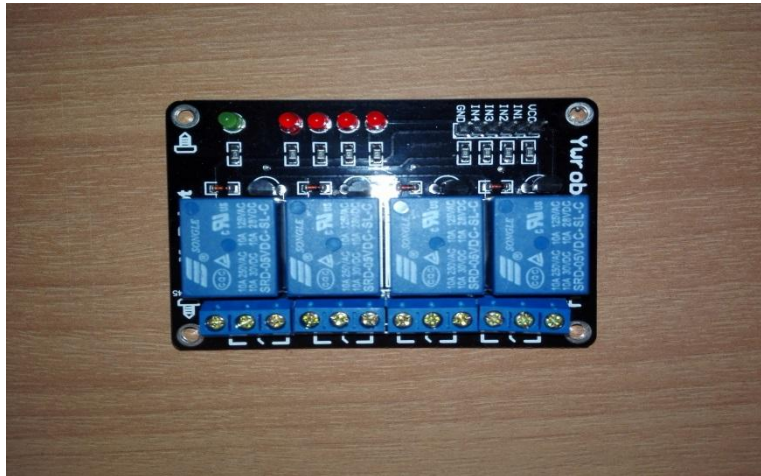


Εικόνα 9 - Arduino Due και Ethernet Shield

Έχουμε πλέον επεκτείνει την πλακέτα μας και μπορούμε εύκολα με την χρήση ενός καλωδίου Ethernet να το συνδέσουμε στο router μας. Μέσα από τον κώδικα μας θα αναρτήσουμε μία ιστοσελίδα, έναν πίνακα ελέγχου, που θα χρησιμοποιήσουμε για τον έλεγχο των συσκευών που θα συνδέσουμε τα relay.



Το board με τα relay που θα χρησιμοποιήσουμε κατασκευάζεται από την Saint Smart είναι το μοντέλο Ywrobot 4 channel relay (Εικόνα 10):



Εικόνα 10 - Relay 4 καναλιών

Το relay είναι ένας ηλεκτρικός διακόπτης που ανοίγει και κλείνει ένα ηλεκτρικό κύκλωμα κάτω από τον έλεγχο ενός άλλου ηλεκτρικού/ηλεκτρονικού κυκλώματος. Αυτό το 'άλλο' κύκλωμα για εμάς θα είναι το Arduino.

Όταν ηλεκτρικό ρεύμα διαρρέει το πηνίο του ηλεκτρονόμου, το παραγόμενο μαγνητικό πεδίο έλκει έναν οπλισμό που είναι μηχανικά συνδεδεμένος σε μια κινούμενη επαφή. Έτσι, η κινούμενη επαφή είτε συνδέεται με μια σταθερή επαφή είτε αποσυνδέεται από τη σταθερή επαφή. Μόλις το ηλεκτρικό ρεύμα στο πηνίο διακοπεί, ο οπλισμός επιστρέφει στη θέση ηρεμίας του εξαιτίας μιας δύναμης επαναφοράς, που είναι ίση με το ήμισυ της μαγνητικής. Κάθε επαφή ενός relay μπορεί να είναι Κανονικά-Ανοικτή (Normally Open, NO), Κανονικά-Κλειστή' (Normally Closed, NC) ή μεταγωγικός (change-over), ανάλογα με τον τύπο της.

- Μια επαφή Κανονικά-Ανοικτή συνδέει το κύκλωμα όταν το relay ενεργοποιείται
- Μια επαφή Κανονικά-Κλειστή αποσυνδέει το κύκλωμα όταν το relay ενεργοποιείται.
- Μια επαφή Μεταγωγική μπορεί να ελέγχει δύο κυκλώματα. Ισοδυναμεί με μια επαφή κανονικά-ανοικτή και μια επαφή κανονικά-κλειστή που έχουν ένα κοινό ακροδέκτη και το αποτέλεσμα της αλλάζει όταν το relay ενεργοποιείται.

## 2.2 Υλοποίηση Συστήματος

### 2.2.1 Βιβλιοθήκες

Ξεκινώντας με το Arduino IDE εισάγουμε τις βιβλιοθήκες που θα χρειαστούμε :

```
//Included Libraries:  
#include <SPI.h>  
#include <Ethernet.h>  
#include <SD.h>
```





**SPI.h:** Serial Peripheral Interface είναι ένα σύγχρονο πρωτόκολλο δεδομένων που χρησιμοποιείται από μικροελεγκτές για την επικοινωνία με μία ή περισσότερες περιφερειακές συσκευές γρήγορα σε μικρές αποστάσεις. Μπορεί επίσης να χρησιμοποιηθεί για την επικοινωνία μεταξύ δύο μικροελεγκτών. Με μία σύνδεση SPI υπάρχει πάντα μια συσκευή πρωτεύων (συνήθως ένας μικροελεγκτής) η οποία ελέγχει τις περιφερειακές. Συνήθως υπάρχουν τρεις γραμμές κώδικα κοινές για όλες τις συσκευές.

- MISO (Master In Slave Out) – Η δευτερεύων (slave) γραμμή για την αποστολή δεδομένων προς την κύρια (master).
- MOSI (Master Out Slave In) – Κύρια γραμμή για την αποστολή δεδομένων με τα περιφερειακά.
- SCK (Serial Clock) – Οι παλμοί του ρολογιού που συγχρονίζουν την διαβίβαση των δεδομένων που δημιουργούνται από την πρωτεύων.
- SS (Slave Select) – Είναι ο ακροδέκτης σε κάθε συσκευή που η κύρια γραμμή μπορεί να χρησιμοποιήσει για να ενεργοποιήσει ή να απενεργοποιήσει συγκεκριμένες συσκευές.

**Ethernet.h:** Με το Arduino Ethernet Shield, αυτή η βιβλιοθήκη επιτρέπει στο Arduino να συνδεθεί με το internet. Μπορεί να χρησιμοποιηθεί ως διακομιστής (server) και να αποδέχεται εισερχόμενες συνδέσεις ή σαν πελάτης (client) και να διαχειρίζεται εξερχόμενες συνδέσεις. Η βιβλιοθήκη αυτή υποστηρίζει έως και τέσσερις ταυτόχρονες συνδέσεις (εισερχόμενη ή εξερχόμενη ή ένα συνδυασμό αυτών). Το Arduino επικοινωνεί με την ασπίδα χρησιμοποιώντας το πρωτόκολλο SPI. Αυτό υποστηρίζεται στους ψηφιακούς ακροδέκτες 11, 12 και 13 για το Uno και στους 50,51,52 και 53 για το Mega και το Due. Στις δύο τελευταίες ο ακροδέκτης SS (53) δεν χρησιμοποιείται για την επιλογή της W5100 αλλά πρέπει να διατηρηθεί ως έξοδος ειδάλλως η διεπαφή του SPI δεν θα λειτουργήσει.

**SD.h:** Η βιβλιοθήκη SD.h επιτρέπει την αναγνώριση και την εγγραφή σε κάρτες SD, πχ σχετικά με το Arduino Ethernet Shield. Είναι βασισμένο σε sdfatlib από τον William Greiman. Η βιβλιοθήκη υποστηρίζει FAT16 και FAT32 συστήματα σχετικά με τις τυποποιημένες κάρτες SD και SDHC. Χρησιμοποιεί συντομώσεις 8.3 για τις ονομασίες των αρχείων. Τα ονόματα των αρχείων που προωθούνται από τις συναρτήσεις της βιβλιοθήκης SD.h μπορεί να περιλαμβάνουν διαδρομές που διαχωρίζονται από τα εμπρός-καθέτους, /, π.χ. "Κατάλογος / filename.txt". Επειδή ο κατάλογος εργασίας είναι πάντα η ρίζα της κάρτας SD, ένα όνομα που παραπέμπει στο ίδιο αρχείο ακόμα και αν δεν συμπεριλαμβάνεται η κύρια κάθετος (leading slash). Η επικοινωνία μεταξύ του μικροελεγκτή και της κάρτας SD χρησιμοποιεί την διάυλο SPI, η οποία λαμβάνει χώρα στους ψηφιακούς ακροδέκτες 11, 12, και 13 (για τα περισσότερα διοικητικά συμβούλια Arduino) ή 50, 51, και 52 (Mega Arduino και Arduino Due). Επιπλέον, ένα άλλος ένας ακροδέκτης πρέπει να χρησιμοποιηθεί για να επιλέξετε την κάρτα SD. Αυτό μπορεί να είναι ο ακροδέκτης SS - pin 10 (στις περισσότερες πλακέτες Arduino) ή pin 53 (για το Mega/Due) - ή κάποιο άλλο ακροδέκτη που καθορίζει την κλήση της *SD.begin()*. Να σημειωθεί ότι ακόμα και αν δεν χρησιμοποιείτε τον πείρο SS, θα πρέπει να μείνει ως έξοδος ή ειδάλλως η βιβλιοθήκη SD δεν θα λειτουργήσει.



## 2.2.2 Σύνδεση στο Δίκτυο

```
// The MAC address of Arduino-Controller:
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
// The IP address of Local Network:
byte ip[] = { 192, 168, 1, 231 };
// Gateway and SubnetMask for Internet access.
byte gateway[] = { 192, 168, 1, 1 };
byte subnet[] = { 255, 255, 255, 0 };
// Default port for HTTP: 80.
EthernetServer server(80);
```

Με τις παραπάνω εντολές στήνουμε το Arduino Due στο δίκτυο μας. Ορίζουμε ως MAC Address της πλακέτας μας την DE:AD:BE:EF:ED. Δηλώνουμε ως IP Address, την διεύθυνση που θα σηκώσουμε την ιστοσελίδα, την 10.0.1.231 με Gateway του δικτύου την 10.0.1.1 και μάσκα υποδικτύου (Subnet Mask) την 255.255.255.0 Η προεπιλεγμένη πύλη για το HTTP είναι η 80 την αφήνουμε αμετάβλητη.

**EthernetServer:** Η δήλωση αυτή καλείται για να δημιουργήσει ένα διακομιστή (server) ο οποίος θα δέχεται εισερχόμενες συνδέσεις από μία συγκεκριμένη πύλη (HTTP: 80). Παίρνει σαν όρισμα την πύλη που θα ελέγχει ενώ δεν έχει τιμή επιστροφής.

```
Ethernet.begin(mac, ip, gateway, subnet);
server.begin();
```

Με την κλήση της συνάρτησης *Ethernet.begin()* ξεκινάμε τον διακομιστή (server) στις ρυθμίσεις που δώσαμε με τις παραπάνω εντολές.

## 2.2.3 Διαθεσιμότητα της κάρτας SD

```
Serial.print("Initializing SD Card: ");
if (!SD.begin(4)){
  Serial.print("Failed!\n");
  Serial.println("Relays Turned OFF for your Safety...");
  sdCheck=false;}
else{Serial.print("Complete!");
  sdCheck=true;
```

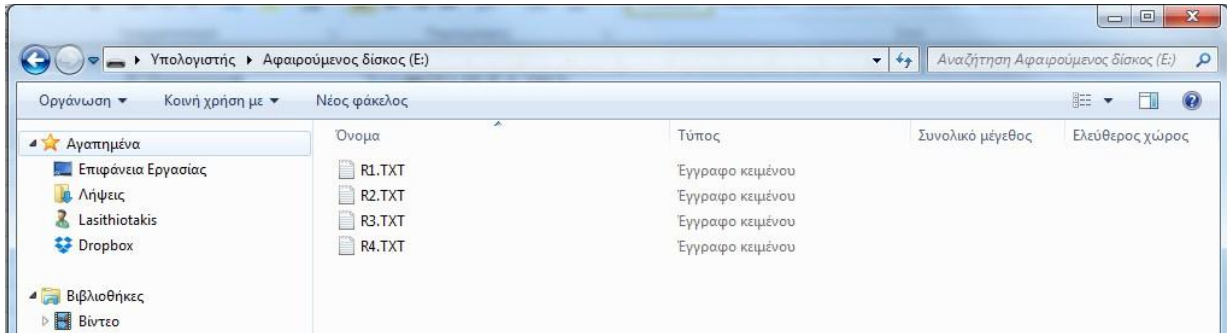
Με τις παραπάνω γραμμές ελέγχεται η διαθεσιμότητα της κάρτας SD. Αυτό κατά βάση γίνεται με την κλήση της συνάρτησης *SD.begin(4)*.

**SD.begin():** Προετοιμάζει την βιβλιοθήκη SD.h και την κάρτα. Αυτό αρχίζει χρήση της διόδου SPI (ψηφιακοί ακροδέκτες 11, 12, και 13 για τα περισσότερα διοικητικά συμβούλια Arduino 50, 51, και 52 στο Mega και Due) Να σημειωθεί ότι ακόμα και αν χρησιμοποιείτε ένα διαφορετικό τσιπ επιλογής, ο ακροδέκτης SS πρέπει να διατηρηθεί ως έξοδο ειδιάλλως οι συναρτήσεις της βιβλιοθήκης SD δεν θα λειτουργήσουν.



Η χρήση της SD κάρτας γίνεται για καταγράφονται οι αλλαγές στις καταστάσεις των relay. Έτσι φτιάχνουμε τέσσερα αρχεία τύπου κειμένου (\*.txt) τα οποία περιέχουν ένα αριθμό (Εικόνα 11).

- **0** αν το relay ήταν κλειστό (OFF)
- **1** αν ήταν ανοικτό (ON)



Εικόνα 11 - Περιεχόμενα κάρτας SD

Αυτό βοηθάει τον αυτοματισμό μας να κρατάει της καταστάσεις στα relay μας. Είναι πολύ σημαντικό σε περίπτωση διακοπής της τροφοδοσίας να γνωρίζουμε ποιες συσκευές ήτα εξαρχής κλειστές και ποιες ανοικτές. Έτσι με την επαναφορά της τροφοδοσίας να μην έχουμε δυσάρεστες επιπτώσεις. Σε περίπτωση που η κάρτα SD απουσιάζει από το Ethernet Shield τότε εμφανίζεται το σχετικό μήνυμα στον χρήστη ενώ παράλληλα τα relay απενεργοποιούνται για λόγους ασφαλείας διότι δεν γνωρίζουμε σε πια κατάσταση θα έπρεπε να ήταν.

### 2.2.4 Έλεγχος κατάστασης Relay

Έχουμε δημιουργήσει τέσσερα αρχεία (R<sub>x</sub>LogData, όπου x αριθμοί από το 1 έως το 4) που περιέχουν την τελευταία γνωστή κατάσταση των relay. Παίρνουμε για παράδειγμα το relay 1:

```
R1LogData = SD.open("R1.TXT");
if (R1LogData){
  while (R1LogData.available()){
    checkR1=R1LogData.read();
    if (checkR1 == '1'){
      st1=true;
      digitalWrite(r1, HIGH);
      Serial.println("Relay 1, last state was: ON");}
    else if(checkR1 == '0'){
      st1=false;
      digitalWrite(r1,LOW);
      Serial.println("Relay 1, last state was: OFF");}}
R1LogData.close();}
```



Με την κλήση της συνάρτησης *SD.open()* καλούμε την SD να ανοίξει το αρχείο R1.txt και κάνουμε το R1LogData ίσο με το R1.txt. Εάν το αρχείο R1 υπάρχει στην κάρτα και όσο το αρχείο είναι διαθέσιμο αποθηκεύουμε τα περιεχόμενα του R1LogData σε μία μεταβλητή τύπου χαρακτήρα την checkR1. Έτσι μπορούμε να ελέγξουμε ποια είναι η τελευταία τιμή του R1. Ακολουθεί ο έλεγχος της checkR1 αν είναι ίση με άσσο τότε με την συνάρτηση *digitalWrite(r1,HIGH)* δίνουμε στον r1 (ακροδέκτης 5) τάση 5V, το relay ανάβει. Στην περίπτωση που η τιμή της μεταβλητής είναι ίση με το μηδέν τότε χρησιμοποιούμε την *digitalWrite(r1,LOW)* που μας δίνει τάση 0V στο ακροδέκτη, το relay κλείνει. Τέλος πριν προχωρήσουμε παρακάτω πρέπει να καλέσουμε την συνάρτηση *R1LogData.close()* με την οποία κλείνουμε το αρχείο που ανοίξαμε για να μην προκληθούν λογικά λάθη στην συνέχεια. Την διαδικασία αυτή την επαναλαμβάνουμε και για τα άλλα τρία relay με τον ίδιο σχετικό κώδικα.

### 2.2.5 Έλεγχος ασφαλείας ιστοσελίδας

```
if (c == '\n' && currentLineIsBlank) {
    //YWRtaW46cGFzcw== is 'admin:pass' [user:password] Base64 Encode.
    // Send a standard HTTP response header.
    if(strstr(header, "YWRtaW46cGFzcw==") != NULL) {
        client.println("HTTP/1.1 200 OK");
        client.println("Content-Type: text/html");
        client.println("Connection: keep-alive");
        .
        .
        .
    } else {
        client.println("HTTP/1.0 401 Authorization Required");
        client.println("WWW-Authenticate: Basic realm=\"Administrator Rights\"");
        client.println("Content-Type: text/html");
        client.println();
        client.println("<HTML>");
        client.println("<H1>Authorization Required.</H1>");
        client.println("This page could not verify that you are authorized to access the control <br/> panel.
            Either you supplied the wrong credentials(e.g. wrong password), or check <br/> your
            browser's settings.");
        client.println("</HTML>");
    }
}
```

Το παραπάνω απόσπασμα από τον κώδικα, μας προσφέρει ασφάλεια στην ιστοσελίδα μας. Ο μηχανισμός πιστοποίησης (Basic Authentication) δεν περιέχει καμία προστασία για τα μεταδιδόμενα διαπιστευτήρια. Διαθέτουν απλώς μία κρυπτογράφηση Base64 κατά την μεταφορά αλλά όχι με την έννοια της κρυπτογράφησης ή του κατακερματισμού.

Επειδή η επικεφαλίδα πιστοποίησης αποστέλλεται με κάθε αίτημα HTTP ο φυλλομετρητής (web browser) θα πρέπει να αποθηκεύει τα διαπιστευτήρια για ένα εύλογο χρονικό διάστημα αποτρέποντας τον χρήστη να εισάγει κάθε φορά το username και το password. Ο χρόνος αυτός διαφέρει από φυλλομετρητή σε φυλλομετρητή.

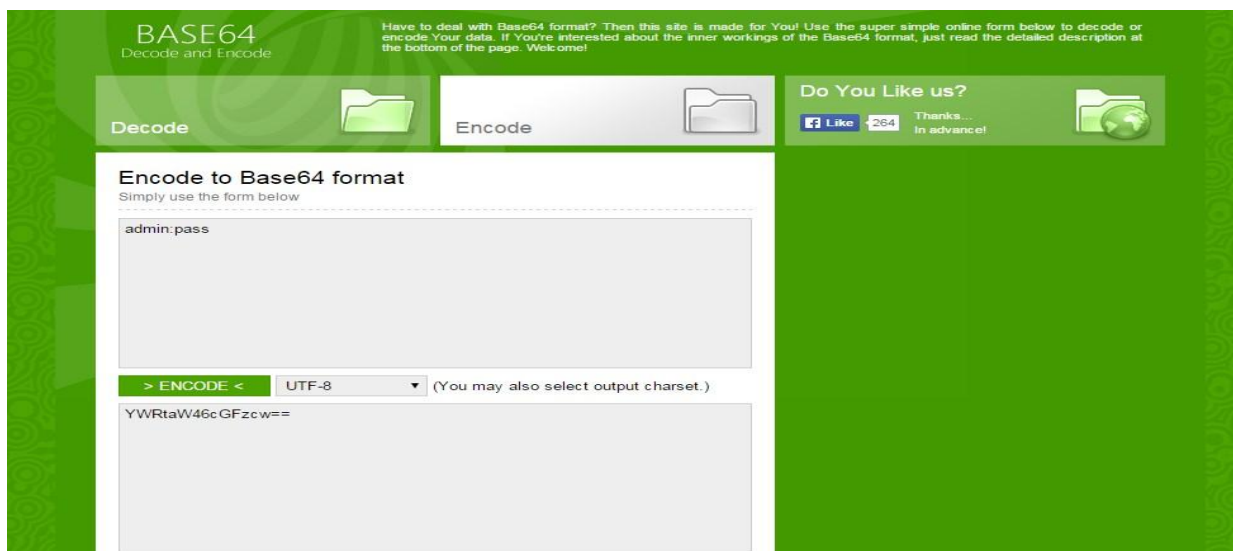




Το HTTP δεν διαθέτει κάποια μέθοδο για τον web server για να διαθέτει στον χρήστη την έξοδο από το σύστημα (log out). Ωστόσο υπάρχουν μία σειρά από μεθόδους για να καταργήσουμε τα προσωρινά αποθηκευμένα διαπιστευτήρια σε ορισμένα προγράμματα περιήγησης του ιστού. Ένας από αυτούς είναι η ανακατεύθυνση του χρήστη σε μία διεύθυνση URL στον ίδιο τομέα (domain) όπου τα διαπιστευτήρια είναι σκόπιμα λανθασμένα .

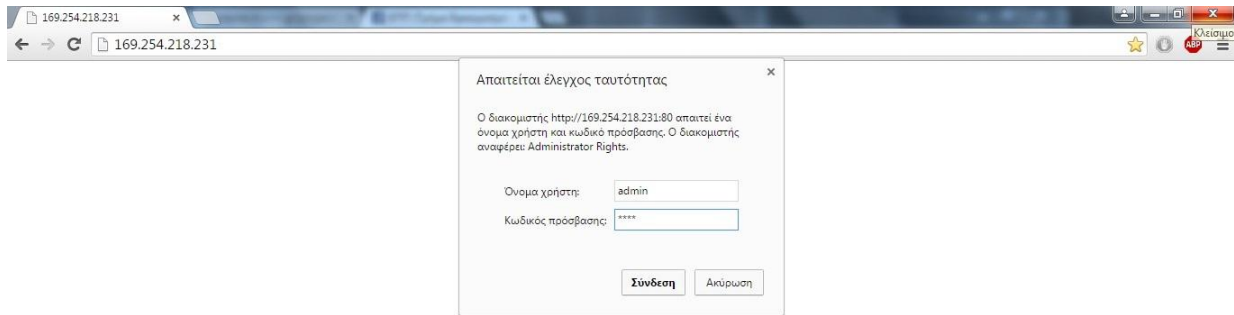
Όταν ο διακομιστής θέλει τον χρήστη να πιστοποιήσει την ταυτότητα του πρέπει να ανταποκριθεί στα σχετικά αιτήματα πιστοποίησης. Χωρίς την πιστοποίηση πρέπει να επιστρέφεται ένα HTTP 401. Μη πιστοποιημένες συνδέσεις περιέχουν μία επικεφαλίδα HTTP www-Authenticate η οποία περιέχει τον βασικό έλεγχο ταυτότητας του χρήστη. Όταν ο χρήστης θέλει να στείλει τα διαπιστευτήρια ελέγχου ταυτότητας του στον διακομιστή μπορεί να χρησιμοποιήσει την κεφαλίδα Εξουσιοδότησης (Authorization Header). Η κεφαλίδα Εξουσιοδότησης κατασκευάζεται ως εξής:

- Το όνομα χρήστη και τον κωδικό πρόσβασης σε μία μεταβλητή τύπου συμβολοσειράς (string) με τη μορφή : "username:password"
- Η συμβολοσειρά που προκύπτει στη συνέχεια κωδικοποιείται χρησιμοποιώντας την παραλλαγή RFC2045-MIME του Base64, αυτό έγινε με την βοήθεια της ιστοσελίδας [www.base64encode.org](http://www.base64encode.org) (Εικόνα 12).



Εικόνα 12 - Κρυπτογράφηση Base64

Έχουμε ορίσει για την ιστοσελίδα μας ως όνομα χρήστη την λέξη admin και ως κωδικό πρόσβασης την λέξη pass (από την αγγλική password). Με την σωστή κρυπτογράφηση της συμβολοσειράς «admin:pass» κατά Base64 προκύπτει η συμβολοσειρά «YWRtaW46cGFzcw== » την οποία και χρησιμοποιούμε μέσα στο κώδικα για τον έλεγχο ενός πιστοποιημένου χρήστη για το σύστημα μας.



Εικόνα 13 - Παράθυρο Πιστοποίησης Χρήστη

Αξίζει να σημειωθεί ότι αν ο χρήστης εισάγει λάθος διαπιστευτήρια η ιστοσελίδα θα φορτώσει εκ νέου το παράθυρο για την εισαγωγή στοιχείων. Αυτό θα επαναλαμβάνεται έως ότου ο χρήστης εισάγει τα σωστά στοιχεία ή πατήσει το κουμπί ακύρωσης ή το κουμπί κλεισίματος παραθύρου.



## Authorization Required.

This page could not verify that you are authorized to access the control panel. Either you supplied the wrong credentials(e.g. wrong password), or check your browser's settings.

Εικόνα 14 - Μη πιστοποιημένη πρόσβαση

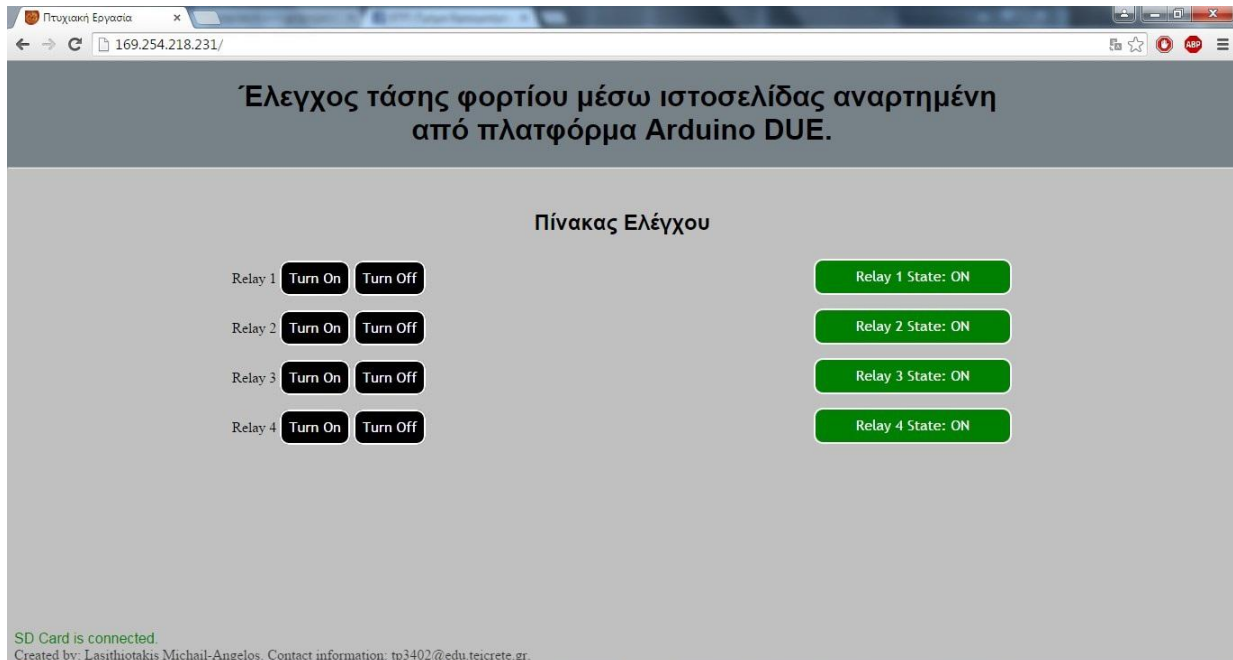
Στην περίπτωση αυτή εμφανίζεται το σχετικό μήνυμα : «This page could not verify that you are authorized to access the control panel. Either you supplied the wrong credentials(e.g. wrong password), or check your browser's settings». Σε ελεύθερη μετάφραση σημαίνει ότι «Αυτή η ιστοσελίδα δεν μπορεί να πιστοποιήσει ότι είστε ένας εξουσιοδοτημένος χρήστης για να αποκτήσετε πρόσβαση στον πίνακα ελέγχου. Είτε εισαγάγατε λάθος διαπιστευτήρια (π.χ. λάθος κωδικό), είτε ελέγξτε τις ρυθμίσεις του φυλλομετρητή σας». Ο χρήστης από αυτό το σημείο μπορεί να κάνει ανακατεύθυνση σε κάποια άλλη ιστοσελίδα ή να κλείσει το πρόγραμμα περιήγησης.

## 2.2.6 Ιστοσελίδα

Πρόκειται για μια ιστοσελίδα τύπου HTML γραμμένη με την χρήση τριών γλωσσών προγραμματισμού. Γνωρίζουμε ότι το Arduino υποστηρίζει C++ για τον λόγο αυτό χρησιμοποιούμε την εντολή `client.println()`, όπου `client` είναι ένα αντικείμενο τύπου `EthernetClient`. Από το αντικείμενο αυτό θα χρησιμοποιήσουμε την μέθοδο `println()`, δηλαδή την ικανότητα να εκτυπώνει στον πελάτη, που στην συγκεκριμένη περίπτωση είναι ο φυλλομετρητής του χρήστη. Με την χρήση της παραπάνω εντολής μπορούμε να γράψουμε το κώδικα της ιστοσελίδας με HTML. Να σημειωθεί ότι σε δύο σημεία έχει γίνει η χρήση του AJAX είναι μία τεχνική για την άμεση αλλαγή των περιεχομένων της ιστοσελίδας χωρίς να



είναι αναγκαία η εκ νέου φόρτωση της σελίδας. Η τεχνική αυτή είναι γραμμένη σε γλώσσα προγραμματισμού JavaScript.

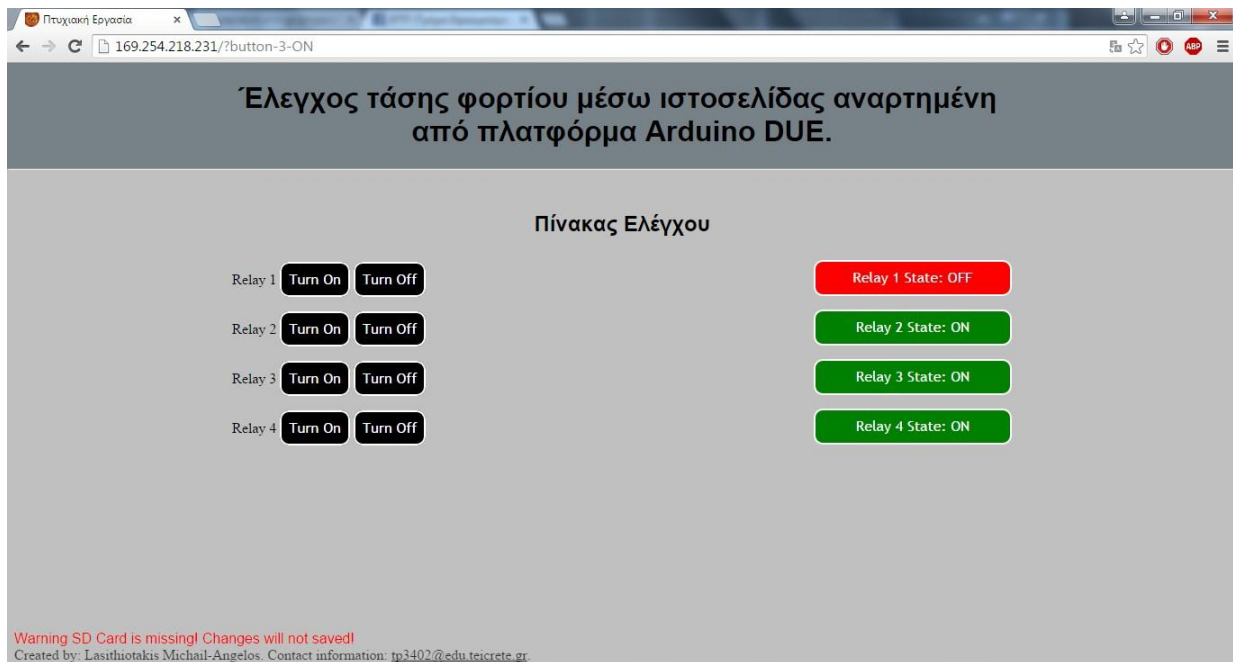


Εικόνα 15 - Ιστοσελίδα: κάρτα SD Συνδεδεμένη

Αυτή είναι μορφή της ιστοσελίδας μας, μπορούμε να διακρίνουμε τα δύο κύρια σημεία του πίνακα ελέγχου. Στα αριστερά έχουν τα κουμπιά μας ON και OFF στην πραγματικότητα δεν πρόκειται για κουμπιά αλλά για συνδέσμους (links) τα οποία μας ουσιαστικά ανακατευθύνουν τις επιλογές μας. Από αυτά τα κουμπιά μπορούμε να ελέγχουμε την κατάσταση των relay που διαθέτουμε. Στα δεξιά έχουμε την τελευταία κατάσταση των relay. Εδώ είναι και το πρώτο σημείο της χρήσης της τεχνικής AJAX μιας και μόλις πατηθεί ένα κουμπί (ON/OFF) αυτόματα αλλάζει και η κατάσταση (state) του relay μας. Έτσι ο χρήστης που έχει πρόσβαση στον πίνακα ελέγχου μας μπορεί να αναγνωρίσει ποιες συσκευές είναι ενεργές και ποιες όχι μέσα στο δίκτυο μας και να πράξει ανάλογα. Είναι πολύ σημαντικό διότι ο χρήστης δεν χρειάζεται να έχει οπτική επαφή με το αντικείμενο που διαχειρίζεται για να ξέρει την κατάσταση του. Πράγμα που καθιστά ικανό τον απομακρυσμένο έλεγχο οποιασδήποτε συσκευής μέσα στο δίκτυο μας ακόμα και από ένα VPN δίκτυο.



Δείξαμε πόσο σημαντική είναι η χρήση της κάρτας SD για το σύστημα μας. Για τον λόγο αυτό γίνεται και η χρήση του δεύτερου κομματιού AJAX για τον έλεγχο της κατάστασης της κάρτας μας. Η κάρτα μας διαθέτει τις τελευταίες καταστάσεις των relay, όμως η κάρτα μας απουσιάζει κατά την εκκίνηση της πλακέτας εμφανίζουμε το σχετικό μήνυμα, στο κάτω-δεξιά μέρος της οθόνης, στον χρήστη και αυτομάτως όλες οι συσκευές απενεργοποιούνται για λόγους ασφαλείας. (Εικόνα 16) Αυτό δεν σημαίνει ότι δεν μπορούν να γίνουν αλλαγές στο σύστημα μας απλά δεν θα αποθηκευτούν. Να σημειωθεί ότι αν επιστραφεί η κάρτα θα τεθούν οι ρυθμίσεις που είναι αποθηκευμένες στα αρχεία της SD.



Εικόνα 16 - Ιστοσελίδα: κάρτα SD Αποσυνδεδεμένη

Στην ιστοσελίδα μας έχουμε χρησιμοποιήσει και ένα αρχείο CSS που καθορίζει την εμφάνιση όλου το γραφικού περιβάλλοντος της διεπαφής μας. Τα χρώματα και η στοίχιση των αντικειμένων στον φυλλομετρητή του χρήστη είναι καθορισμένα από αυτό το αρχείο.

```
client.println("<!DOCTYPE html>");
client.println("<HTML>");
client.println("<HEAD>");
client.println("<meta name='apple-mobile-web-app-capable' content='yes' charset=utf-8 />");
client.println("<meta name='apple-mobile-web-app-status-bar-style' content='width=device-width, height=device-height, initial-scale=1.0'>");
client.println("<link rel='shortcut icon' href='https://dl.dropboxusercontent.com/u/106805737/test.ico' />");
client.println("<link rel='stylesheet' type='text/css' href='https://dl.dropboxusercontent.com/u/106805737/Style.css'/>");
```



Οι τρεις χρήσιμες αυτές εντολές, τοποθετούνται στην επικεφαλίδα (head) της σελίδας, δίνοντας οπτικά ωραίες επισημάνσεις στην ιστοσελίδα μας.

- `client.println("<meta name='apple-mobile-web-app-capable' content='yes' charset=utf-8 />");`
- `client.println("<meta name='apple-mobile-web-app-status-bar-style' content='width=device-width, height=device-height, initial-scale=1.0'>");`

Προσαρμόζει την ιστοσελίδα στα μέτρα των κινητών τηλεφώνων καθιστώντας δυνατή σύνδεση των και τον έλεγχο από τα έξυπνα τηλέφωνα (smartphones). Το αποτέλεσμα είναι να φαίνεται η ιστοσελίδα μας όπως παρουσιάζεται στον φυλλομετρητή του υπολογιστή και να ανταποκρίνεται εξίσου επάξια.

- `client.println("<link rel='shortcut icon' href='https://dl.dropboxusercontent.com/u/106805737/test.ico' />");`

Η εντολή αυτή χρησιμοποιείται για την φόρτωση του αρχείου Style.css από τον ιστό. Συγκεκριμένα, για λόγους ευχρηστίας είναι αναρτημένο σε δημόσιο φάκελο της υπηρεσίας Dropbox ([www.dropbox.com/Style.css](https://www.dropbox.com/Style.css)), για να μπορούμε να στήσουμε όσα συστήματα θέλουμε χωρίς να επιβαρύνουμε την μνήμη της πλακέτας μας.

- `client.println("<link rel='stylesheet' type='text/css' href='https://dl.dropboxusercontent.com/u/106805737/Style.css'/>");`

Μία μικρή λεπτομέρεια που προσθέτει κύρος στην ιστοσελίδα μας είναι το λογότυπο που παρίσταται δίπλα στον τίτλο της σελίδας. Πρόκειται για το λογότυπο του TEI Κρήτης, με την ίδια τεχνική χρησιμοποιώντας την υπηρεσία του Dropbox. ([www.dropbox.com/Logo.ico](https://www.dropbox.com/Logo.ico))

### 2.2.7 Αλλαγή κατάστασης με χρήση κουμπιών

Ο κώδικας των κουμπιών είναι σχετικά απλός. Πρόκειται για μία σειρά επαναλαμβανόμενων `else if` συνθηκών μιας και βρισκόμαστε μέσα σε μία συνεχή επανάληψη, `void loop()`. Έτσι σε κάθε επανάληψη ελέγχεται ο σύνδεσμος (url) και ανάλογα με την ανακατεύθυνση που έχει γίνει από τον χρήστη και προστίθεται ένα αντίστοιχο string (`?button-x-ON` ή `OFF`) που ξεχωρίζει τα κουμπιά- συνδέσμους. Έτσι για το κουμπί υπ' αριθμό ένα (1) έχουμε:

```
//Control Buttons:
if (readString.indexOf("?button-1-ON") >0){
  st1=true;
  btnCheck=false;
  digitalWrite(r1, HIGH);
  Serial.println("Relay 1 is ON.");
  if(SD.exists("R1.TXT")){
    SD.remove("R1.TXT");
    R1LogData = SD.open("R1.TXT", FILE_WRITE);
    if (R1LogData) {
      R1LogData.write("1");
      R1LogData.close();}
    else{Serial.println("Error opening R1.txt");}}
else if (readString.indexOf("?button-1-OFF") >0){
  st1=false;
  btnCheck=false;
  digitalWrite(r1, LOW);
```



```
Serial.println("Relay 1 is OFF.");  
if(SD.exists("R1.TXT")){  
SD.remove("R1.TXT");  
R1LogData = SD.open("R1.TXT", FILE_WRITE);  
if (R1LogData) {  
R1LogData.write("0");  
R1LogData.close();}  
else{Serial.println("Error opening R1.txt");}}
```

Ουσιαστικά αυτό που γίνεται είναι ο έλεγχος για το τι υπάρχει μετά το 169.254.218.231/. Τον διαχωρισμό τον καθορίζει η κάθετος (/). Έτσι χρησιμοποιώντας το αντικείμενο `readString` και καλώντας την μέθοδο `indexOf()` μπορούμε να διαβάσουμε το τι αναγράφεται στο `url` συγκρίνοντας το με το ήδη υπάρχων.

Στην περίπτωση που αυτό που διαβάσαμε είναι το «`?button-1-ON`» τότε δίνουμε στο relay 1 τάση 5Volt. Έπειτα, δείχνουμε το σχετικό μήνυμα στην σειριακή οθόνη. Τέλος, ελέγχουμε αν είναι διαθέσιμη η κάρτα SD, αν ναι διαγράφουμε το αρχείο που αντιστοιχεί στο συγκεκριμένο relay και χρησιμοποιώντας την μέθοδο της εγγραφής δημιουργούμε ένα νέο έγγραφο με δεδομένα έναν άσσο (1) και κλείνουμε το αρχείο. Αν δεν υπάρχει διαθέσιμη κάρτα εμφανίζεται κατάλληλο μήνυμα στην σειριακή οθόνη.

Στην περίπτωση που αυτό που διαβάσαμε είναι το «`?button-1-OFF`» τότε δίνουμε στο relay 1 τάση 0Volt. Έπειτα, δείχνουμε το σχετικό μήνυμα στην σειριακή οθόνη. Τέλος, ελέγχουμε αν είναι διαθέσιμη η κάρτα SD, αν ναι διαγράφουμε το αρχείο που αντιστοιχεί στο συγκεκριμένο relay και χρησιμοποιώντας την μέθοδο της εγγραφής δημιουργούμε ένα νέο έγγραφο με δεδομένα ένα μηδέν (0) και κλείνουμε το αρχείο. Αν δεν υπάρχει διαθέσιμη κάρτα εμφανίζεται κατάλληλο μήνυμα στην σειριακή οθόνη.

Η ίδια διαδικασία επαναλαμβάνεται και για τα τέσσερα relay και για τις δύο περιπτώσεις (ON – OFF).

### 2.2.8 Συνάρτηση `StrClear()`

Μία από της συναρτήσεις που χρειάστηκε να δημιουργήσουμε ήταν `StrClear()`

```
void StrClear(char *str, char length){  
for (int i = 0; i < length; i++){str[i] = 0; } }
```

Δέχεται δύο ορίσματα το πρώτο είναι δείκτης σε χαρακτήρα στην περίπτωση μας είναι το `header` και το δεύτερο είναι το μήκος του χαρακτήρα που είναι 500bytes. Με μία επανάληψη μηδενίζουμε το `header`. Αυτό είναι απαραίτητο μιας και βρισκόμαστε μέσα στην συνάρτηση, `void loop()`, και πρέπει να καθαρίσουμε το `header` για την επόμενη ανάγνωση.





## 2.2.9 Συνάρτηση GetRelayState( )

Μέσα στον κώδικα της ιστοσελίδας χρησιμοποιούμε το παρακάτω script για να ελέγξουμε την κατάσταση των Relay:

```
client.println("<script>");
client.println("function GetRelayState() {");
client.println("nocache = \"&nocache=\" + Math.random() * 1000000;");
client.println("var request = new XMLHttpRequest();");
client.println("request.onreadystatechange = function() {");
client.println("if (this.readyState == 4) {");
client.println("if (this.status == 200) {");
client.println("if (this.responseText != null) {");
client.println("document.getElementById(\"relay_txt\").innerHTML = this.responseText;");
client.println("}}}}");
client.println("request.open(\"GET\", \"ajax_relay_switch\" + nocache, true);");
client.println("request.send(null);");
client.println("setTimeout('GetRelayState()', 1000);");
client.println("}");
client.println("</script>");
```

Από αυτό το script, γίνεται η κλήση της συνάρτησης *GetRelayState()* :

```
// send the state of the switch to the web browser
void GetRelayState(EthernetClient client){
    if (st1){
        client.println("<div class='ON'>");
        client.println("Relay 1 State: ON");
        client.println("</div>");
    }
    else{
        client.println("<div class='OFF'>");
        client.println("Relay 1 State: OFF");
        client.println("</div>");
    }
    if (st2){
        client.println("<div class='ON'>");
        client.println("Relay 2 State: ON");
        client.println("</div>");
    }
    else{
        client.println("<div class='OFF'>");
        client.println("Relay 2 State: OFF");
        client.println("</div>");
    }
    if (st3){
        client.println("<div class='ON'>");
        client.println("Relay 3 State: ON");
        client.println("</div>");
    }
    else{
        client.println("<div class='OFF'>");
        client.println("Relay 3 State: OFF");
        client.println("</div>");
    }
}
```



```
if (st4){
    client.println("<div class='ON'>");
    client.println("Relay 4 State: ON");
    client.println("</div>");}
else{
    client.println("<div class='OFF'>");
    client.println("Relay 4 State: OFF");
    client.println("</div>");}
```

Με την κλήση της συνάρτησης ελέγχεται η τιμή της κάθε μεταβλητής st1, st2, st3 και st4 αν είναι αληθείς ή ψευδείς και ανάλογα να εμφανίσει τον σχετικό κομμάτι κώδικα. Ενώ ταυτόχρονα καλείτε το css μορφοποίησης του κώδικα με τις κλάσεις που του έχουμε ορίσει.

### 2.2.10 Συνάρτηση GetSDState( )

Μέσα στον κώδικα της ιστοσελίδας χρησιμοποιούμε το παρακάτω script για να ελέγξουμε την κατάσταση της κάρτας SD:

```
client.println("<script>");
client.println("function GetSDState() {");
client.println("nocache = \"&nocache=\" + Math.random() * 1000000;");
client.println("var request = new XMLHttpRequest();");
client.println("request.onreadystatechange = function() {");
client.println("if (this.readyState == 4) {");
client.println("if (this.status == 200) {");
client.println("if (this.responseText != null) {");
client.println("document.getElementById(\"sd_txt\").innerHTML = this.responseText;");
client.println("}}}}");
client.println("request.open(\"GET\", \"ajax_sd_switch\" + nocache, true);");
client.println("request.send(null);");
client.println("setTimeout('GetSDState()', 1000);");
client.println("}");
client.println("</script>");
```

Από αυτό το script, γίνεται η κλήση της συνάρτησης *GetSDState()* :

```
void GetSDState(EthernetClient client){
    if(sdCheck){
        client.println("<div class='SDON'>");
        client.println("SD Card is connected.");
        client.println("</div>");}
    else{
        client.println("<div class='SDOFF'>");
        client.println("Warning SD Card is missing! Changes will not save!");
        client.println("</div>");}
```

Με την κλήση της συνάρτησης ελέγχεται η τιμή της κάθε μεταβλητής sdCheck αν είναι αληθείς ή ψευδείς και ανάλογα να εμφανίσει τον σχετικό κομμάτι κώδικα.





### 2.2.11 Σειριακή οθόνη

Ένα σημαντικό κομμάτι κυρίως για τους προγραμματιστές είναι η σειριακή οθόνη του Arduino (Ctrl + Shift + M). Να σημειωθεί ότι αφορά αποκλειστικά προγραμματιστές και ότι ο χρήστης δεν έχει κάποια πρόσβαση στο τερματικό.

Η επικοινωνία μέσω του H/Y γίνεται μέσα απ' την σειριακή οθόνη (Serial Monitor) του Arduino IDE. Για να επιτευχθεί η σωστή επικοινωνία θα πρέπει και οι δύο πλευρές, H/Y και Arduino, να έχουν τον ίδιο ρυθμό μετάδοσης δεδομένων (baud rate). Για εμάς πάντα ο ρυθμός μετάδοσης θα είναι τα 9600kbps. Σε προγραμματιστικό επίπεδο, ώστε να μπορέσουμε να πούμε στο Arduino πως θα επικοινωνήσουμε με αυτό, η εντολή που χρησιμοποιούμε είναι η **Serial.begin(9600)**; Η τιμή 9600 αναφέρεται στον ρυθμό μετάδοσης. Η εντολή αυτή αποτελεί πάντα κομμάτι της *void setup()*.

Οι βασικές συναρτήσεις της σειριακής θύρας είναι:

- *begin()* - αρχικοποίηση της σειριακής
- *end()* - κλείσιμο της σειριακής
- *available()* - έλεγχος αν υπάρχουν δεδομένα να διαβαστούν
- *read()* - ανάγνωση των εισερχόμενων σειριακών δεδομένων
- *peek()* - επιστρέφει το επόμενο byte από την σειριακή
- *flush()* - άδειασμα του buffer της σειριακής από δεδομένα που έχει
- *print()* - γράψιμο δεδομένων στη σειριακή
- *println()* - το ίδιο με την *Print()*, αλλά με αλλαγή γραμμής στο τέλος
- *write()* - γράφει δυαδικά δεδομένα στη σειριακή

```
COM8 (Arduino Due (Programming Port))
Initializing SD Card: Complete!
Server is at: 192.168.1.231

Relays Status:
Relay 1, last state was: ON
Relay 2, last state was: ON
Relay 3, last state was: ON
Relay 4, last state was: ON

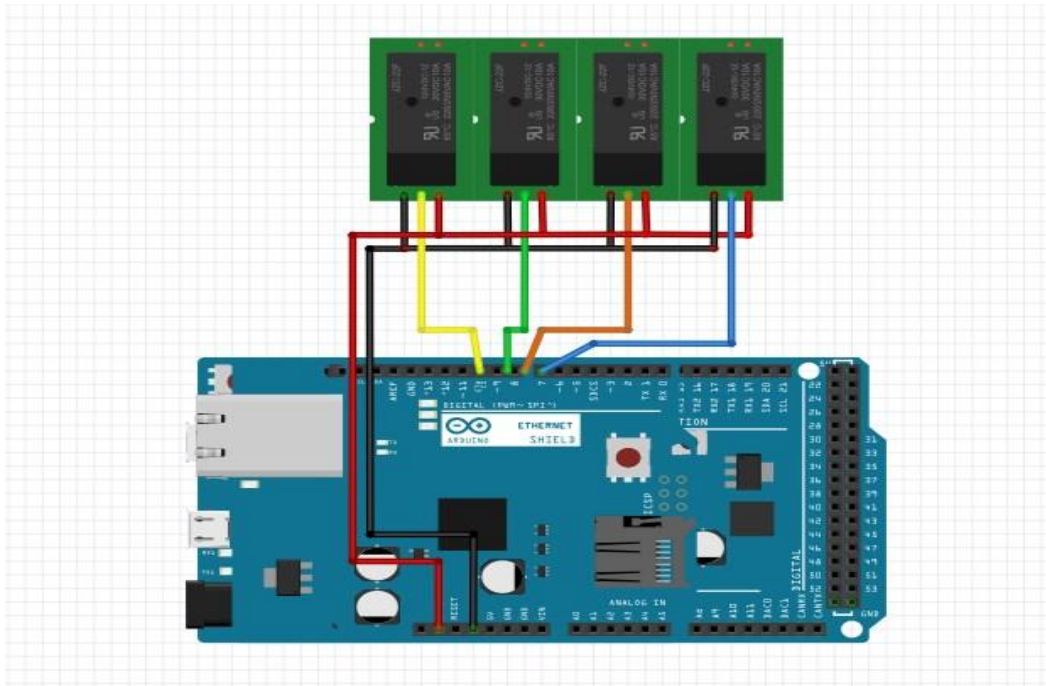
Relay 1 is OFF.
Relay 2 is OFF.
Relay 3 is OFF.
Relay 4 is OFF.
Relay 2 is ON.
Relay 3 is ON.
Relay 4 is ON.
```

Εικόνα 17 - Σειριακή Οθόνη Arduino



### 2.2.12 Συνδεσμολογία Συστήματος

Χρησιμοποιώντας το πρόγραμμα της εταιρίας Fritzing που μπορείτε να κατεβάσουμε [εδώ](#) . Μπορέσαμε να φτιάξουμε το σχηματικό του project μας.



Εικόνα 18 - Σχηματικό Διάγραμμα Συνδεσμολογίας

Ακροδέκτης Arduino	Καλώδιο	Ακροδέκτης Relay
7	Μπλε	State 1
8	Πορτοκαλί	State 2
9	Πράσινο	State 3
10	Κίτρινο	State 4
VCC	Κόκκινο	VCC
GND	Μαύρο	GND

Πίνακας 2 - Συνδεσμολογία



## Κεφάλαιο 3<sup>ο</sup>

### 3.1 Αποτελέσματα

Το τελικό στάδιο της υλοποίησης του συστήματος είχε σαν στόχο να γίνει προσιτή προς τον τελικό χρήστη η επικοινωνία με τον μικροελεγκτή. Ο χρήστης, δηλαδή, πρέπει να χρησιμοποιεί μία διεπαφή, είτε από τον υπολογιστή του, είτε από το κινητό του τηλέφωνο και να ελέγχει τις συσκευές του σπιτιού του.

Ο αρχικός στόχος που τέθηκε κατά την υλοποίηση του συστήματος επιτεύχθηκε. Υλοποιήθηκε, δηλαδή, ένα σύστημα οικιακού αυτοματισμού πλήρως λειτουργικό. Ο τελικός χρήστης, έχοντας πρόσβαση στο διαδίκτυο, έχει τη δυνατότητα να ελέγχει οικιακές συσκευές. Επιπλέον, το σύστημα είναι πλήρως επεκτάσιμο. Για τη συγκεκριμένη εφαρμογή χρησιμοποιήθηκε σαν συσκευή προς έλεγχο ένα φωτιστικό χώρου. Με τον ίδιο τρόπο θα μπορούσε να ελεγχθεί οποιαδήποτε οικιακή συσκευή που συνδέεται στα 220V του δικτύου της Δ.Ε.Η, δηλαδή, σε μια οικιακή πρίζα. Οι δυνατότητες του συστήματος βέβαια δεν περιορίζονται στον έλεγχο μόνο μίας συσκευής, αφού ο μικροελεγκτής διαθέτει πλήθος γραμμών εισόδου/εξόδου, τόσο ψηφιακών όσο και αναλογικών. Οι αναλογικές εισοδοί μπορούν να χρησιμοποιηθούν είτε σαν ψηφιακές I/O, οπότε έτσι μπορούν να ελεγχθούν και επιπλέον συσκευές, είτε σαν εισοδοί αισθητήρων αναλογικών μεγεθών που θα επιτρέπουν στο χρήστη να ελέγχει αναλογικά μεγέθη όπως πίεση, υγρασία, φωτεινότητα κ.τ.λ. Το σύστημα, συνεπώς, μπορεί να τροποποιηθεί και να ικανοποιήσει διαφορετικές ανάγκες χρηστών.

Η τροποποίηση αυτή περιλαμβάνει τρία στάδια. Το στάδιο της αλλαγής της συνδεσμολογίας του μικροελεγκτή με τις συσκευές προς έλεγχο. Το στάδιο της τροποποίησης του σχεδίου (sketch) που θα φορτωθεί στον μικροελεγκτή, καθώς και το στάδιο της τροποποίησης των τελικών εφαρμογών του χρήστη. Στο σημείο αυτό θα πρέπει να επισημανθεί ότι οι κώδικες που έχουν γραφεί σίγουρα δεν είναι βέλτιστοι οπότε κάποιος μπορεί να τους τροποποιήσει και να έχει καλύτερα αποτελέσματα.

Επιπλέον, θα πρέπει να επισημανθεί και να εξεταστεί το θέμα της ασφάλειας. Η ασφάλεια της επικοινωνίας δεν αποτέλεσε αντικείμενο της παρούσας υλοποίησης. Σίγουρα ένας ειδικός στην ασφάλεια των επικοινωνιών θα βρει τρωτά σημεία στον τομέα της επικοινωνίας, όπως για παράδειγμα το ότι ο κωδικός ασφαλείας ταξιδεύει «φανερά» στη σύνδεση με πιθανό κίνδυνο υποκλοπής.

Θα πρέπει παρόλα αυτά να τονιστεί ότι οι προαναφερθείσες βελτιώσεις και επεκτάσεις δεν αυξάνουν (παρά μόνο ελάχιστα ίσως) το κόστος του συστήματος, το οποίο δεν ξεπέρασε τα 30€, χρησιμοποιώντας παντού καινούρια υλικά. Αυτός ήταν και ένας επιπλέον στόχος της παρούσας διπλωματικής, η ανάπτυξη δηλαδή ενός πλήρως λειτουργικού συστήματος οικιακού αυτοματισμού με το μικρότερο δυνατό κόστος.



### 3.2 Συμπεράσματα

Στις μέρες μας τα πάντα τείνουν να αυτοματοποιηθούν. Σε ένα τέτοιο πλαίσιο αυτοματοποίησης δε θα μπορούσε να μην πάρει μέρος η καθημερινότητα μας, το ίδιο μας το σπίτι δηλαδή. Ο οικιακός αυτοματισμός διευκολύνει τη ζωή μας. Κάποτε, για να είχε κάποιος φως σε ένα δωμάτιο θα έπρεπε να σκαρφαλώνει σε μια σκάλα και να ρυθμίζει μια λάμπα πετρελαίου. Στη συνέχεια τα πράγματα έγιναν ευκολότερα, καθώς αρκούσε μόνο το πάτημα ενός διακόπτη και το φως άναβε. Σήμερα έχουμε φτάσει σε ένα σημείο που πλέον δεν χρειάζεται καν να πατάμε αυτόν το διακόπτη, μιας και ένας υπολογιστής το αναλαμβάνει για εμάς. Κάποιος θα μπορούσε αυτό να το αποκαλέσει οκνηρία, στην τεχνολογική γλώσσα, ωστόσο, ονομάζεται πρόοδος.

Από το αυτόματο πότισμα του κήπου έως την εναλλαγή καναλιών και σημάτων ήχου και video εντός της οικίας, αυτό που ονομάζουμε «έξυπνο» σπίτι μπορεί να κάνει τη ζωή μας ευκολότερη και ποιοτικότερη. Με μια μικρή δαπάνη σε χρόνο και σε χρήματα σχεδόν τα πάντα πλέον μπορούν να αυτοματοποιηθούν σε ένα σπίτι. Ένα έξυπνο σπίτι μπορεί να είναι όσο σύνθετο ή όσο απλό θελήσουμε. Η βασική ιδέα πίσω από το έξυπνο σπίτι και τον οικιακό αυτοματισμό γενικότερα είναι να εξοπλίσουμε μια κατοικία με αισθητήρες και συστήματα ελέγχου και με τη βοήθεια τους να ελέγχουμε τη θερμοκρασία, τον κλιματισμό, τον φωτισμό και πολλές άλλες υπηρεσίες. Προσαρμόζοντας περαιτέρω τους μηχανισμούς του σπιτιού στις ανάγκες του κατόχου του, το έξυπνο σπίτι μπορεί να αποτελέσει ένα πιο ασφαλές, πιο άνετο και πιο οικονομικό κατάλυμα. Παραδείγματος χάριν, ο ηλεκτρονικός ελεγκτής σε ένα αυτοματοποιημένο σπίτι μπορεί να αποφασίσει τότε οι ένοικοι έχουν αποσυρθεί από ένα δωμάτιο και να κλείσει τα φώτα ή να χαμηλώσει τον θερμοστάτη. Μπορεί ακόμη να ενεργοποιήσει συναγερμούς φωτιάς ή κλοπής, καθώς και να βελτιστοποιήσει τη διαχείριση πόρων, όπως το νερό και το ρεύμα.

Το πρώτο οργανωμένο project για το «έξυπνο» σπίτι ξεκίνησε στις αρχές του 1980 ως ένα εγχείρημα του Κέντρου Έρευνας της Εθνικής Ένωσης Κατασκευαστών Σπιτιών των ΗΠΑ (National Research Center of the National Association of Home Builders - NAHB). Αρκετές εταιρείες κυρίως του ιδιωτικού τομέα έλαβαν μέρος και συγχρηματοδότησαν το συγκεκριμένο project.

Επισήμως η λέξη «έξυπνο» χρησιμοποιήθηκε πρώτη φορά κατά τη δεκαετία του '70 ως προσδιορισμός τεχνολογικών επιτευγμάτων. Συγκεκριμένα, αναφερόταν σε στρατιωτικά προϊόντα, όπως βόμβες ή πυραύλους που καθοδηγούσαν τον εαυτό τους προς το στόχο («έξυπνες βόμβες»).

Κατά την τεχνολογική άνθηση της δεκαετίας του '80 η λέξη «έξυπνο» απέκτησε άλλες προεκτάσεις: αναφερόταν σε συσκευές που εμπεριείχαν ολοκληρωμένα κυκλώματα (τσιπ), όπως οι υπολογιστές και οι προηγμένες οικιακές συσκευές. Βέβαια αυτό άλλαξε με την πάροδο του χρόνου και πλέον σήμερα δεν αποκαλούμε έναν σύγχρονο υπολογιστή «έξυπνο», παρόλο που οι σημερινοί υπολογιστές είναι εκθετικά ισχυρότεροι από εκείνους της δεκαετίας του '80. Ο όρος «έξυπνο σπίτι» καθιερώθηκε από την αμερικανική ομοσπονδία κατασκευαστών σπιτιών το 1984. Ένας απλός ορισμός για την έννοια του έξυπνου σπιτιού είναι ο εξής:

*«Έξυπνο σπίτι είναι μια κατοικία που ενσωματώνει ένα δίκτυο επικοινωνίας, το οποίο συνδέει ηλεκτρικές συσκευές και υπηρεσίες και επιτρέπει απομακρυσμένο έλεγχο, παρακολούθηση και πρόσβαση σε αυτές.»*



Όταν μιλάμε για απομακρυσμένο έλεγχο εννοούμε ότι οι συσκευές και οι υπηρεσίες θα μπορούν να ελεγχθούν μέσα ή έξω από την κατοικία. Ο ορισμός αυτός συμφωνεί με τις περισσότερες περιπτώσεις υλοποίησης έξυπνων σπιτιών, αφού σχεδόν πάντα έχουμε να κάνουμε με δικτύωση και αλληλεπίδραση συσκευών. Συνεπώς, ο όρος «έξυπνο» δεν δηλώνει την καθαυτό υποδομή του συστήματος (δικτύωση, αισθητήρες, διακόπτες κ.τ.λ.), αλλά αναφέρεται στο συνδυασμένο περιβάλλον ανθρώπου και υποδομής ώστε οι δράσεις και οι αλληλεπιδράσεις του να έχουν σαφή «έξυπνα» αποτελέσματα στον τελικό χρήστη.

Η βασική τεχνολογία που χρησιμοποιείται στο έξυπνο σπίτι είναι αναμφίβολα η ασύρματη, όπως το Bluetooth, το RFID και το WiFi. Παρόλο που πολλές από τις τεχνολογίες αυτές χρειάζονται μπαταρίες για την τροφοδοσία τους και κάποιες φορές κρίνονται αναποτελεσματικές, το πλεονέκτημα της έλλειψης καλωδίωσης είναι πολύ μεγάλο για τον τελικό χρήστη. Μάλιστα, υπάρχουν ερευνητές που ισχυρίζονται ότι στο μέλλον οι ασύρματες τεχνολογίες θα αποτελούν αυστηρό προαπαιτούμενο για οποιοδήποτε έξυπνο σπίτι. Όταν έχουμε να κάνουμε με δικτύωση σε προϋπάρχοντα σπίτια και όχι σε καινούρια, οι ασύρματες επιλογές μοιάζουν ακόμα καλύτερες. Ένα ξεκάθαρο σημείο υπεροχής είναι ότι η εγκατάσταση καλωδίωσης μπορεί να σημαίνει αρκετά υψηλό κόστος, ειδικά σε ένα σπίτι που δεν είναι υπό κατασκευή και χρειάζονται παρεμβάσεις. Υπολογίζεται ότι το κόστος εγκατάστασης ενός αισθητήρα κυμαίνεται από 50% έως 90% της αξίας του. Ένα ακόμη πλεονέκτημα της ασύρματης τεχνολογίας είναι η κινητικότητα συσκευών και χρηστών που παρέχει το έξυπνο σπίτι. Πιο συγκεκριμένα, ο χρήστης μπορεί να μεταφέρει εύκολα μια συσκευή χωρίς να ανησυχεί για καλώδια, αρκεί να μείνει εντός εμβέλειας. Επίσης, είναι εύκολο να εγκατασταθεί και να είναι λειτουργική μια νέα συσκευή στο ήδη υπάρχον ασύρματο δίκτυο. Αντίθετα, κάτι τέτοιο δεν ισχύει σε ένα ενσύρματο δίκτυο.

Παρά τα αρκετά πλεονεκτήματα των ασύρματων λύσεων η παραδοσιακή καλωδίωση παραμένει μια καλή επιλογή. Μπορεί να κοστίζει περισσότερο και να είναι δύσκολο να αντικατασταθεί ή να τροποποιηθεί μελλοντικά, αλλά η ενσύρματη δικτύωση προσφέρει ξεχωριστά πλεονεκτήματα όπως ασφάλεια και σταθερότητα. Σε αυτούς ακριβώς τους τομείς υστερεί η ασύρματη τεχνολογία. Στη συνέχεια γίνεται αναφορά σε κάποιες τεχνολογίες επικοινωνίας που χρησιμοποιούνται στα έξυπνα σπίτια. Κάθε μια από αυτές έχει τα πλεονεκτήματα και τα μειονεκτήματά της. Έτσι λοιπόν δεν υπάρχει η βέλτιστη επιλογή, αλλά η καλύτερη δυνατή που καλύπτει τις εκάστοτε ανάγκες του τελικού χρήστη.

Όπως το περιγράφει ο δημιουργός του, το Arduino είναι μια «ανοικτού κώδικα» πλατφόρμα «προτυποποίησης» ηλεκτρονικών βασισμένη σε ευέλικτο και εύκολο στη χρήση hardware και software που προορίζεται για οποιονδήποτε έχει λίγη προγραμματιστική εμπειρία, στοιχειώδεις γνώσεις ηλεκτρονικών και ενδιαφέρεται να δημιουργήσει διαδραστικά αντικείμενα ή περιβάλλοντα. Στην ουσία, πρόκειται για ένα ηλεκτρονικό κύκλωμα που βασίζεται στον μικροελεγκτή ATmega της Atmel και του οποίου όλα τα σχέδια, καθώς και το software που χρειάζεται για την λειτουργία του, διανέμονται ελεύθερα και δωρεάν ώστε να μπορεί να κατασκευαστεί από τον καθένα (απ' όπου και ο περίεργος -για hardware- χαρακτηρισμός «ανοικτού κώδικα»). 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. Μάλιστα κάποιος θα μπορούσε να ισχυριστεί - και θα ήταν ένας αρκετά πετυχημένος παραλληλισμός - ότι λειτουργικά το Arduino μοιάζει πολύ με το NXT Brick των Lego



Mindstorms NXT. Άλλωστε η ρομποτική είναι μια από τις πολλές εφαρμογές στις οποίες το Arduino διαπρέπει.

Το Arduino βέβαια, δεν είναι ούτε ο μοναδικός, 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 που έχει στον νου του, το Arduino, με το εκτενές documentation, καταφέρνει να κερδίσει όλους αυτούς των οποίων οι γνώσεις στα ηλεκτρονικά περιορίζονται στα όσα λίγα έμαθαν στο σχολείο. Ακριβώς επειδή απευθύνεται κυρίως σε αρχάριους των ηλεκτρονικών και επειδή, παρά τις αναλυτικότερες οδηγίες που υπάρχουν, δεν έχουν όλοι τις γνώσεις και τα μέσα να κατασκευάσουν μια ηλεκτρονική πλακέτα, κυκλοφορούν έτοιμες, προκατασκευασμένες πλακέτες Arduino. Με λίγα χρήματα παραπάνω μάλιστα, οι περισσότεροι προμηθευτές διαθέτουν Arduino Starter Kit, τα οποία, εκτός από το ίδιο το Arduino, περιέχουν διάφορα άλλα εξαρτήματα και εργαλεία που μπορεί να χρειαστείτε για τις πρώτες σας εφαρμογές (όπως το απαραίτητο καλώδιο USB για την σύνδεση με τον υπολογιστή, ράστερ, καλώδια, LED, διακόπτες, ποτενσιόμετρα, αντιστάσεις, διόδους, τρανζίστορ κ.λπ.).





### 3.3 Μελλοντική Εργασία και Επεκτάσεις

Η εξέλιξη της τεχνολογίας έχει συμβάλει πολύ σημαντικά στην βελτίωση της ζωής του ανθρώπου. Στις μέρες μας οι περισσότεροι από εμάς καταναλώνουμε το μεγαλύτερο μέρος της ημέρας μας με την τεχνολογία. Τα κινητά τηλέφωνα έχουν εξελιχθεί σε πολυεργαλεία της καθημερινότητας. Μπορούμε να επικοινωνήσουμε με τον όλο κόσμο, να ενημερωθούμε, να ακούσουμε την μουσική που μας αρέσει. Τα τελευταία χρόνια τα κινητά τηλέφωνα μπορούν να επικοινωνήσουν και με άλλες συσκευές. Εδώ παίρνουν μέρος οι μικροελεγκτές, που αναλαμβάνουν την επεξεργασία των ερεθισμάτων που παίρνουν από τις συσκευές.

Λόγω του ισχυρότατου ανταγωνισμού αλλά και της τάσης ενσωμάτωσης των μικροελεγκτών σε κάθε ηλεκτρική και ηλεκτρονική συσκευή, η βιομηχανία μικροελεγκτών έχει καταλήξει στην παραγωγή ανταγωνιστικών μοντέλων μαζικής παραγωγής καθώς και μικροελεγκτών για πιο εξειδικευμένες εφαρμογές. Έτσι διακρίνονται οι εξής κυρίως κατηγορίες:

- Μικροελεγκτές (καμιά φορά 4-bit αλλά συνήθως 8-bit) πολύ χαμηλού κόστους, γενικής χρήσης, με πολύ μικρό αριθμό ακροδεκτών (ακόμη και λιγότερους από 8). Σχεδιάζονται με έμφαση στη χαμηλή κατανάλωση ισχύος και την αυτάρκεια, ώστε να χρειάζονται ελάχιστα ή και καθόλου εξωτερικά εξαρτήματα και να μη μπορεί να αντιγραφεί εύκολα το εσωτερικό λογισμικό τους. Απουσιάζει η δυνατότητα επέκτασης της μνήμης τους. Μερικά μοντέλα είναι ευρέως γνωστά στους ερασιτέχνες ηλεκτρονικούς, όπως πχ οι περισσότεροι μικροελεγκτές των σειρών PIC (Microchip), AVR (Atmel) και 8051 (Intel, Atmel, Dallas κα).
- Μικροελεγκτές (συνήθως 8-bit αλλά και 16 ή 32-bit) χαμηλού κόστους, γενικής χρήσης, με μέτριο έως σχετικά μεγάλο αριθμό ακροδεκτών. Διαθέτουν μεγάλο αριθμό κοινών περιφερειακών, όπως θύρες UART, I<sup>2</sup>C, SPI ή CAN, μετατροπείς αναλογικού σε ψηφιακό και ψηφιακού σε αναλογικό. Στους κατασκευαστές της Άπω Ανατολής (Ιαπωνία, Κορέα), συνηθίζεται η ενσωμάτωση ελεγκτών οθόνης υγρών κρυστάλλων και πληκτρολογίου. Μερικές φορές παρέχουν δυνατότητα εξωτερικής επέκτασης της μνήμης τους.
- Μικροελεγκτές (κυρίως 32-bit) μέσου κόστους, γενικής χρήσης, με μεγάλο αριθμό ακροδεκτών. Χαρακτηρίζονται από έμφαση στην ταχύτητα εκτέλεσης εντολών, υψηλή αυτάρκεια περιφερειακών και μεγάλες δυνατότητες εσωτερικής ή εξωτερικής μνήμης προγράμματος (FLASH) και RAM. Στο χώρο αυτό έχουν ισχυρή παρουσία οι αρχιτεκτονικές με υψηλή μεταφερσιμότητα λογισμικού (portability) από τον ένα στον άλλο κατασκευαστή. Πχ μεταξύ των μικροελεγκτών τύπου ARM ή MIPS, το σύνολο των βασικών εντολών που αναγνωρίζει η ALU είναι ακριβώς το ίδιο, μειώνοντας έτσι τις μεγάλες αλλαγές στο λογισμικό, όταν στο μέλλον ο πελάτης υιοθετήσει ένα μικροελεγκτή άλλου κατασκευαστή (αρκεί, φυσικά, να υποστηρίξει κι αυτός το σύνολο εντολών ARM ή MIPS, αντίστοιχα).
- Μικροελεγκτές εξειδικευμένων εφαρμογών, οι οποίοι ενσωματώνουν συνήθως κάποιο εξειδικευμένο πρωτόκολλο επικοινωνίας το οποίο υλοποιείται πάντοτε σε hardware. Τέτοιοι μικροελεγκτές χρησιμοποιούνται σε τηλεπικοινωνιακές συσκευές όπως τα μόντεμ.



Η μεγάλη μερίδα πωλήσεων των μικροελεγκτών εξακολουθεί να αφορά αυτούς των 8-bit, καθώς είναι η κατηγορία με το χαμηλότερο κόστος και το μικρότερο μέγεθος λογισμικού για το ίδιο αποτέλεσμα, ιδίως επειδή οι σύγχρονες οικογένειες μικροελεγκτών 8-bit έχουν πολύ βελτιωμένες επιδόσεις σε σχέση με το παρελθόν

Στον μικροεπεξεργαστή, το ολοκληρωμένο κύκλωμα που τον αποτελεί περιέχει μόνο την *Λογική και Αριθμητική Μονάδα (ALU)*, στοιχειώδεις καταχωρητές (registers), προσωρινή μνήμη RAM πολύ υψηλής ταχύτητας (cache memory) και, κάποιες φορές, τον ελεγκτή μνήμης (memory controller). Όμως, για τη λειτουργία ενός πλήρους ενσωματωμένου υπολογιστικού συστήματος, απαιτούνται πολλά εξωτερικά υποσυστήματα και περιφερειακά. Τέτοια είναι:

- Κύκλωμα συνδετικής λογικής (glue logic) για τη σύνδεση των εξωτερικών μνημών και άλλων περιφερειακών παράλληλης σύνδεσης στην αρτηρία δεδομένων (bus) του επεξεργαστή.
- Μνήμη προγράμματος (τύπου ROM, FLASH, EPROM κλπ) η οποία περιέχει το λογισμικό του συστήματος. Σε κάποια μοντέλα, είναι δυνατό το κλείδωμα αυτής της μνήμης, μετά την εγγραφή της, ώστε να προστατευτεί το περιεχόμενό της από αντιγραφή.
- Μεγάλο μέγεθος μνήμης RAM.
- Μόνιμη μνήμη αποθήκευσης παραμέτρων λειτουργίας (τύπου EEPROM ή NVRAM) η οποία να μπορεί να γράφεται τον πυρήνα του μικροελεγκτή. Αυτή η μνήμη έχει, έναντι της FLASH, το πλεονέκτημα της δυνατότητας διαγραφής και εγγραφής οποιουδήποτε μεμονωμένου byte.
- Κύκλωμα αρχικοποίησης (reset).
- Διαχειριστή αιτήσεων διακοπής (interrupt request controller) από τα περιφερειακά.
- Κύκλωμα επιτήρησης τροφοδοσίας (brown-out detection) το οποίο παρακολουθεί την τροφοδοσία και αρχικοποιεί ολόκληρο το σύστημα όταν αυτή πέσει κάτω από τα ανεκτά όρια, προλαμβάνοντας έτσι την αλλοίωση των δεδομένων.
- Κύκλωμα επιτήρησης λειτουργίας (watchdog timer) το οποίο αρχικοποιεί το σύστημα, αν αυτό εμφανίσει σημάδια δυσλειτουργίας λόγω κολλήματος (hang).
- Τοπικό ταλαντωτή για την παροχή παλμών χρονισμού (clock).
- Έναν ή περισσότερους χρονιστές-απαριθμητές υψηλής ταχύτητας (hardware timer-counter) για τη δημιουργία καθυστερήσεων, μέτρηση διάρκειας γεγονότων, απαρίθμηση γεγονότων και άλλων λειτουργιών ακριβούς χρονισμού.
- Ρολόι πραγματικού χρόνου (Real Time Clock, RTC) το οποίο τροφοδοτείται από ανεξάρτητη μπαταρία και γι αυτό πρέπει να έχει πολύ χαμηλή κατανάλωση ρεύματος.
- Σειρά ανεξάρτητων ψηφιακών εισόδων και εξόδων (Parallel Input-Output, PIO).

Γενικά, όλες οι οικογένειες μικροελεγκτών ενσωματώνουν τα περισσότερα από τα παραπάνω περιφερειακά, με διαφοροποιήσεις κυρίως στην ύπαρξη ή μη εσωτερικής μνήμης προγράμματος και στο είδος της. Έτσι, υπάρχουν:

- Μικροελεγκτές χωρίς μνήμη προγράμματος, οι οποίοι χαρακτηρίζονται ως *ROM-less*. Αυτοί παρέχουν πάντοτε μια παράλληλη αρτηρία (bus) δεδομένων, πάνω στην οποία συνδέονται εξωτερικές μνήμες προγράμματος και RAM. Τέτοιοι τύποι μικροελεγκτών προορίζονται για πιο ισχυρά υπολογιστικά συστήματα ελέγχου, με μεγαλύτερες απαιτήσεις μνήμης.





- Μικροελεγκτές με μνήμη ROM, η οποία κατασκευάζεται με το λογισμικό της (Mask ROM) ή γράφεται μόνο μια φορά (One Time Programmable, OTP). Παρέχουν τη δυνατότητα πολύ χαμηλού κόστους, όταν αγοράζονται σε πολύ μεγάλες ποσότητες.
- Μικροελεγκτές με μνήμη FLASH, οι οποίοι μπορούν συνήθως να προγραμματιστεί πολλές φορές. Αυτή είναι η πιο διαδεδομένη κατηγορία. Συχνά ο προγραμματισμός της μνήμης μπορεί να γίνει ακόμη και πάνω στο κύκλωμα της ίδιας της ενσωματωμένης (embedded) εφαρμογής (δυνατότητα In Circuit Programming, ISP). Αυτοί οι μικροελεγκτές έχουν ουσιαστικά αντικαταστήσει τους παλαιότερους τύπους EPROM που έσβηναν με υπεριώδη ακτινοβολία

### **3.4 Μικροελεγκτές**

Μικροελεγκτές που μπορούν να εκτελέσουν το ίδιο project με την ίδια επιτυχία:

#### **Raspberry Pi**

Το Raspbberriy Pi είναι ένας πλήρης υπολογιστής με μέγεθος πιστωτικής κάρτας.

Η τελευταία του έκδοση, Raspberry Pi 2, με τετραπύρηνο επεξεργαστή τύπου ARM και 1GB RAM κοστίζει γύρω στα 45 ευρώ στην Ελλάδα, ενώ ξεχωριστά θα βρούμε το πλήρες kit με όσα χρειάζονται για να το αξιοποιήσουμε.

Παρά τον ελάχιστον όγκο του, το Raspberry Pi διαθέτει τετραπύρηνο επεξεργαστή 900MHz, διπύρηνη κάρτα γραφικών, 1GB RAM, τέσσερις θύρες USB, έξοδο HDMI, τροφοδοτείται μέσω Micro USB, και 40 pins γενικής χρήσης για σύνδεση με άλλα ηλεκτρονικά και περιφερειακά.

Η σημαντικότερη καινοτομία της νέας έκδοσης του Raspberry Pi είναι πως θα μπορεί να τρέξει τα επερχόμενα Windows 10, τα οποία μάλιστα θα διατίθενται δωρεάν για τη συσκευή μέσω του Windows Developer Program for IoT.

Τα μοντέλα αυτά διαθέτουν επεξεργαστή ARMv6k στα 700 MHz, 256MB RAM, κάρτα γραφικών Broadcom VideoCore IV, και κατανάλωση από 1 έως 3.5 watt, ενώ η αποθήκευση των δεδομένων γινόταν σε κάρτες SD, SDHC και MicroSD.

Μόνο το Raspberry Pi Model B, μέσα σε δύο χρόνια από όταν βγήκε στη μαζική παραγωγή, πούλησε πάνω από δύο εκατομμύρια κομμάτια.

#### **Intel 8051**

Ο επεξεργαστής της Intel 8051 είναι στην ουσία ένας μικροελεγκτής (μC) ο οποίος δημιουργήθηκε από την Intel το 1980 για τη χρήση του σε ενσωματωμένα συστήματα. Ήταν πολύ δημοφιλής από το 1980 μέχρι τις αρχές του 1990, αλλά σήμερα έχει κατά πολύ ξεπεραστεί εξαιτίας της μεγάλης ανάπτυξης καινούργιων επεξεργαστών από εταιρίες όπως η Atmel, Infineon Technologies, Maxim IC (από τη θυγατρική της Dallas Semiconductor subsidiary), NXP (πρώην Philips Semiconductor), Winbond, ST Microelectronics, Silicon Laboratories (παλιότερα Cygnal), Texas Instruments και Cypress Semiconductor, οι οποίοι έχουν παρόμοιες αλλά και περισσότερες δυνατότητες από τον 8051.



Η οικογένεια της Intel 8051 σχεδιάστηκε αρχικά χρησιμοποιώντας την τεχνολογία NMOS, αργότερα πέρασε στη τεχνολογία CMOS, η οποία αναγνωρίζεται στα τσιπάκια με το γράμμα "C" στο όνομά τους π.χ. 80C51. Η CMOS χρησιμοποιεί λιγότερη ενέργεια από την NMOS - κάτι που έκανε τους επεξεργαστές αυτούς να είναι συμβατοί με συσκευές που χρησιμοποιούν μπαταρία.

### **LPC2000**

Η LPC2000 είναι μια σειρά μικροελεγκτών από Ημιαγωγούς της Philips. Η οικογένεια 16/32-κομματιών LPC2000 είναι βασισμένη σε έναν πυρήνα 1.8V που λειτουργεί σε μέχρι 60 MHz μαζί με ένα ευρύ φάσμα των περιφερειακών μονάδων συμπεριλαμβανομένων των πολλαπλάσιων τμηματικών διεπαφών 10-κομμάτι PAX και εξωτερικών διαύλων. Αυτοί οι ελεγκτές σχεδιάζονται για τη χρήση σε μια σειρά των εφαρμογών που απαιτούν τη μεγάλη απόδοση και τη μικρή κατανάλωση ισχύος σε μια οικονομικώς αποδοτική συσκευασία.

### **PicoBlaze**

Ο PicoBlaze είναι ένας ελεύθερος μαλακός πυρήνας επεξεργαστών από Xilinx για τη χρήση σε FPGAs. Είναι βασισμένο RISC η αρχιτεκτονική 8 μπιτ και μπορεί να φθάσει στην ταχύτητα μέχρι 100 MIPS Virtex 4 FPGA "οικογένεια του s. Αυτό επεξεργαστής έχει έναν οκτάμπιτο λιμένα διευθύνσεων και στοιχείων για την πρόσβαση σε ένα ευρύ φάσμα των περιφερειακών μονάδων. Ο πυρήνας είναι δωρεάν λογισμικό και έρχεται με εργαλεία ανάπτυξης. Τα εργαλεία τρίτου είναι διαθέσιμα από Mediatronix και άλλα.



## Βιβλιογραφία

- [1] Ανάπτυξη Εφαρμογών με το Arduino, Παπάζογλου Παναγιώτης, Λιώνης Σπυρίδων Πολυχρόνης, Οργανισμός Εκδόσεων Τεχνικές Εκδόσεις.
- [2] Δομή και Λειτουργία Μικροϋπολογιστών, Κ. Πεκμεστζή, Ι.Βογιατζής, Γ. Λιβιεράτος, Π. Μπουγάς, Οργανισμός Εκδόσεων Διδακτικών Βιβλίων Αθήνας.
- [3] 30 Arduino Projects for the Evil Genius, Simon Monk
- [4] Programming Your Home – Automate with Arduino, Android and your Computer, Mike Riley, Edited by Jacquelyn Carter
- [5] Building Wireless Sensor Networks , Robert Faludi ,O'REILLY
- [6] Beginning Arduino Programming – Writing Code for the Most Popular Microcontroller Board in the World, by Brian Evans , Technology in Action

## Webography

(Όπως εμφανίστηκαν στις 30/06/2015)

- [7] <http://www.arduino.cc> και <http://www.forum.arduino.cc>
- [8] <https://el.wikipedia.org/wiki/Arduino>
- [9] <https://el.wikipedia.org/wiki/Μικροελεγκτής>