ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΔΥΤΙΚΗΣ ΕΛΛΑΔΟΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ 1499

Android εφαρμογή μέσω Arduino για απομακρυσμένο έλεγχο έξυπνου σπιτιού

ΣΠΟΥΔΑΣΤΗΣ: Πουλής Μάριος-Αντώνιος (5880)

ΣΠΟΥΔΑΣΤΗΣ: Κανελλόπουλος Παναγιώτης (5878)

ΕΠΟΠΤΕΥΟΝ ΚΑΘΗΓΗΤΗΣ

Χριστόπουλος Κωνσταντίνος

ΠΑΤΡΑ 2015

Πρόλογος

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι η υλοποίηση ενός συστήματος ελέγχου και πρόσβασης ενός έξυπνου σπιτιού. Το ενσωματωμένο σύστημα κατασκευάστηκε με την αρχιτεκτονική Arduino. Το Arduino συνδέεται μέσω διαδικτύου, με την βοήθεια της Ethernet Shield, για να μπορούμε να κάνουμε τον έλεγχο από οποιοδήποτε σημείο εμείς επιθυμήσουμε.

Περίληψη

Η παρούσα πτυχιακή εργασία έχει σκοπό τον έλεγχο και την διαχείριση μιας ηλεκτρικής εγκατάστασης από απόσταση. Στην προκειμένη περίπτωση, θα ελέγχουμε διάφορους αισθητήρες και συσκευές που έχουν εγκατασταθεί μέσα σε μια ηλεκτρολογική εγκατάσταση μιας μονοκατοικίας. Οι αισθητήρες και οι συσκευές συνδέονται σε μια πλακέτα η οποία ονομάζεται Arduino και αποτελείται από έναν μικροελεγκτή που θα είναι κατάλληλα προγραμματισμένος. Ο προγραμματισμός επιτυγχάνεται με την χρήση ενός ηλεκτρονικού υπολογιστή με την προϋπόθεση, ότι η πλακέτα είναι συνδεδεμένη με αυτόν και έχοντας εγκαταστήσει το αντίστοιχο λογισμικό. Επίσης, όσον αφορά την πρόσβαση στο διαδίκτυο, έχει προστεθεί μια επιπλέον πλακέτα που ονομάζεται Ethernet Shield η οποία μας παρέχει την ικανότητα να συνδεόμαστε στην εφαρμογή που θα υλοποιήσουμε μέσω διαδικτύου, με αποτέλεσμα να έχουμε τον έλεγχο από όποιο σημείο επιθυμούμε. Λόγω του ότι η τεχνολογία έχει εξελιχθεί σε διάφορους τομείς της καθημερινότητάς μας, καθώς και στον δικό μας, δημιουργήσαμε μια Android εφαρμογή για σύγχρονη, άμεση και εύχρηστη διαχείριση της εγκατάστασής μας.

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 3^{Ο} - Πειραματικό μέρος

ΚΕΦΑΛΑΙΟ 1Ο

1.1 Arduino

Εικόνα 1: Arduino

1.1.1 Τι είναι το Arduino.

Το Arduino είναι ένας single-board μικροελεκτής, δηλαδή μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή, εισόδους/εξόδους, η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring (ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, SuperCollider. Οι περισσότερες εκδόσεις του Arduino μπορούν να αγοραστούν προ-συναρμολογημένες, το διάγραμμα και πληροφορίες για το υλικό είναι ελεύθερα διαθέσιμα για αυτούς που θέλουν να συναρμολογήσουν το Arduino μόνοι τους.

1.1.2 Λόγοι επιλογής Arduino.

Υπάρχουν πολλοί άλλοι μικροελεγκτές που διατίθενται για την εξυπηρέτηση των αναγκών μας, όπως είναι οι Parallax Basic Stamp, Netmedia του BX-24, Phidgets, Handyboard του MIT και πολλοί άλλοι προσφέρουν παρόμοια λειτουργικότητα. Όλα αυτά τα εργαλεία καλύπτονται κατά ένα πολύ μεγάλο μέρος τους, από τον προγραμματιζόμενο μικροελεγκτή Atmega το οποίο είναι ένα πακέτο εύκολο στη χρήση. Το Arduino απλοποιεί τη διαδικασία της εργασίας με μικροελεγκτές αλλά προσφέρει ακόμα και κάποιο πλεονέκτημα για τους εκπαιδευτικούς, τους μαθητές και τους ενδιαφερόμενους ερασιτέχνες σε σχέση με άλλα συστήματα. Αυτά είναι:

-Λίγα έξοδα –

Οι Arduino πλακέτες είναι σχετικά φθηνές σε σύγκριση με άλλες πλακέτες μικροελεγκτών. Οι πιο φθηνές εκδόσεις της ενότητας Arduino μπορούν να συναρμολογηθούν με το χέρι, ακόμη και οι προ-συναρμολογημένες ενότητες Arduino κοστίζουν λιγότερο από 25 €.

-Cross-platform –

Το Arduino λογισμικό τρέχει σε Windows, Macintosh OSX και Linux λειτουργικά συστήματα. Τα περισσότερα συστήματα μικροελεγκτή περιορίζονται στα Windows.

-Απλό και ξεκάθαρο περιβάλλον προγραμματισμού –

Το περιβάλλον προγραμματισμού Arduino είναι εύκολο στη χρήση για αρχάριους, αλλά είναι και αρκετά ευέλικτο για προχωρημένους χρήστες να το εκμεταλλευτούν. Επίσης για τους εκπαιδευτικούς, είναι μια χρήσιμη βάση το περιβάλλον προγραμματισμού , έτσι ώστε οι μαθητές να μάθουν και μέσα από αυτό το περιβάλλον θα είναι πιο εξοικειωμένοι με την εμφάνιση και την αίσθηση του Arduino.

-Open source και επεκτάσιμο λογισμικό –

Το λογισμικό Arduino δημοσιεύεται ως εργαλείο ανοικτού κώδικα, που διατίθεται για την επέκταση από έμπειρους προγραμματιστές. Η γλώσσα μπορεί να επεκταθεί σε C ++ βιβλιοθήκες και οι άνθρωποι που ξέρουν και κατανοούν τις τεχνικές λεπτομέρειες μπορούν να ασχοληθούν στο Arduino με τη γλώσσα προγραμματισμού AVR-C επί των οποίων βασίζεται. Ομοίως, μπορείτε να προσθέσετε τον κωδικό AVR-C απευθείας στο Arduino τα προγράμματά σας.

-Το Arduino βασίζεται σε ATmega μικροελεγκτές της Atmel. –

Τα Sketch δημοσιεύονται με την άδεια Creative Commons, κάποιοι έμπειροι προγραμματιστές μπορούν να κάνουν το δικό τους Sketch, επεκτείνοντας και βελτιώνοντάς το. Ακόμα, σχετικά άπειροι χρήστες μπορούν να δημιουργήσουν οποιοδήποτε κύκλωμα πάνω στη breadboard, προκειμένου να κατανοήσουνε πώς λειτουργεί και να εξοικονομήσουνε χρήματα.

1.1.3 Μικροελεγκτής ATmega328.

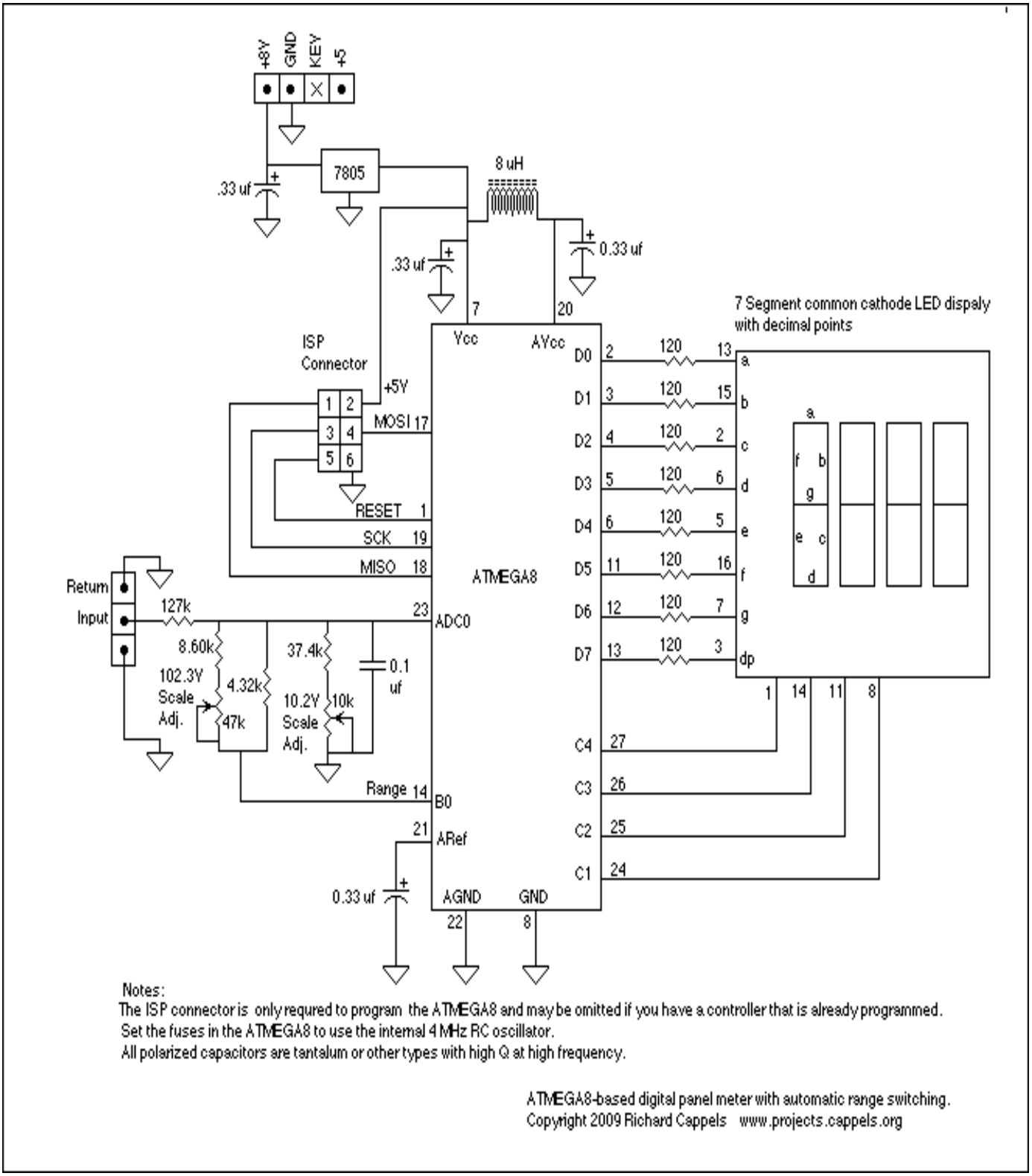
Εικόνα 2: ATmega328

Η καρδιά του Arduino Uno είναι ο ATmega328, ένας 8-bit μικροελεγκτής, ο οποίος λειτουργεί στα 16MHz. Ο ATmega328 έχει δικιά του μνήμη τριών διαφορετικών τύπων:

-Η πρώτη έχει 2Kb μνήμης SRAM, η οποία είναι η μνήμη που μπορούν να χρησιμοποιήσουν τα Sketch μας για να κάνουν Save τα διάφορα στοιχεία εν ώρα λειτουργίας. Όπως και σε έναν υπολογιστή, αυτή η μνήμη χάνει τα DATA της όταν γίνει Reset ή το Arduino δεν τροφοδοτείται πια.

-Η δεύτερη έχει 1Kb μνήμης EEPROM, η οποία μπορεί να χρησιμοποιηθεί για εγγραφή/ανάγνωση DATA (χωρίς Datatype) ανά Byte από τα Sketch μας, εν ώρα λειτουργίας. Σε αντίθεση με την SRAM, η EEPROM δεν χάνει τα περιεχόμενά της όταν για παράδειγμα πάψει να τροφοδοτείται το Arduino ή γίνει Reset, οπότε είναι το ανάλογο του σκληρού δίσκου.

-Η Τρίτη έχει 32Kb μνήμης Flash, από τα οποία τα 2Kb χρησιμοποιούνται από το Firmware του Arduino που έχει εγκαταστήσει ήδη ο κατασκευαστής του. Το Firmware, αυτό που στην ορολογία του Arduino ονομάζεται Bootloader είναι αναγκαίο για την εγκατάσταση των δικών μας Sketch στον μικροελεγκτή μέσω της θύρας USB, χωρίς δηλαδή να χρειάζεται εξωτερικός Hardware Programmer. Τα υπόλοιπα 30Kb της μνήμης Flash χρησιμοποιούνται για την αποθήκευση αυτών ακριβώς των προγραμμάτων. Η μνήμη Flash, όπως και η EEPROM δεν χάνει τα περιεχόμενά της με απώλεια τροφοδοσίας ή Reset.



Εικόνα 3: Διάγραμμα ATmega

1.1.4 Περιγραφή πλακέτας Arduino UNO.

Το Arduino έχει σειριακό Interface. Ο ATmega "σηκώνει" σειριακή επικοινωνία, την οποία το Arduino την στέλνει μέσα από ένα στοιχείο που ελέγχει (Serial-Over-USB) ώστε να μπορεί να συνδεθεί μέσω USB. Η σύνδεση αυτή χρειάζεται για την φόρτωση των Sketch από τον υπολογιστή στο Arduino αλλά και για την επικοινωνία του Arduino με τον υπολογιστή εν ώρα λειτουργίας. Επιπλέον, στην πάνω μεριά του Arduino υπάρχουν 14 Pin, που μπορούν να χρησιμοποιηθούν ως ψηφιακές είσοδοι/έξοδοι. Δουλεύουν στα 5V.

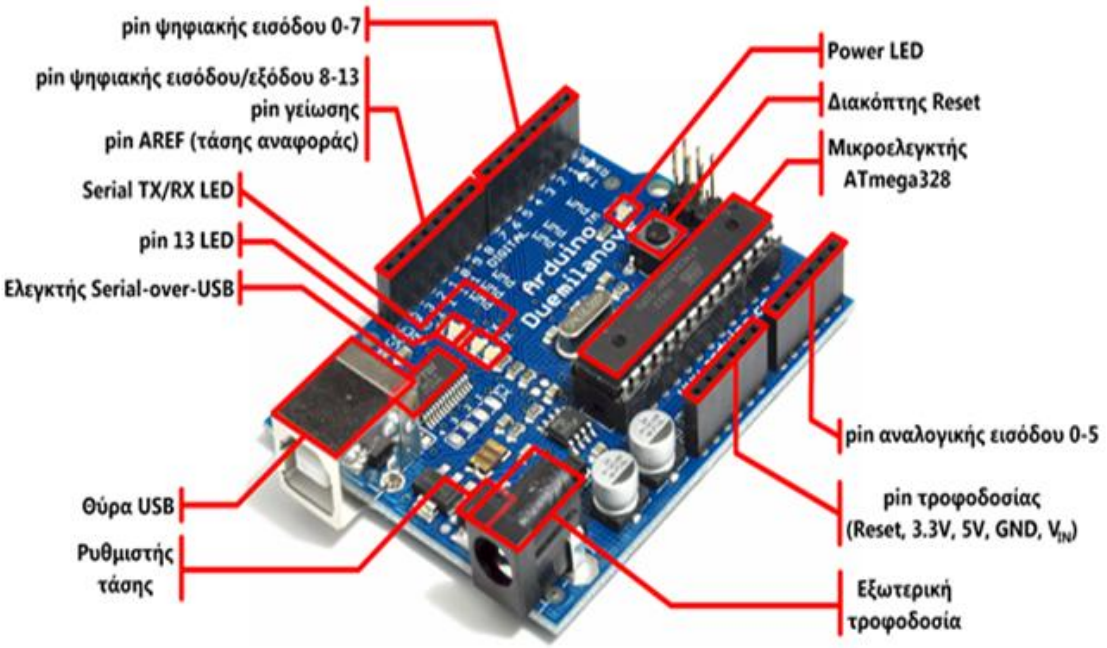
Ως ψηφιακή έξοδος, μπορεί να τεθεί ένα από αυτά τα Pin από το Sketch μας σε κατάσταση High ή Low, οπότε το Arduino θα ξέρει αν πρέπει να δώσει ή όχι ρεύμα σε αυτό το Pin. Με αυτόν τον τρόπο μπορούμε για παράδειγμα να λειτουργήσουμε ένα Led που έχουμε συνδέσει σε αυτό το Pin. Αν πάλι ρυθμίσουμε ένα από αυτά τα Pin ως ψηφιακή είσοδο μέσα από το Sketch μας, μπορούμε με την κατάλληλη εντολή να διαβάσουμε την κατάστασή του, ανάλογα με το αν η συσκευή που έχουμε συνδέσει πάνω σε αυτό το Pin δίνει ρεύμα ή όχι. Κάποια από τα Pin, έχουν την δυνατότητα και για δεύτερη λειτουργία δηλαδή:

- Τα αρχικά Pin (0 - 1) δουλεύουν και ως RX και TX όταν το Sketch μας δίνει λειτουργία στην σειριακή θύρα. Έτσι, για παράδειγμα όταν το Sketch μας στέλνει δεδομένα στην σειριακή, αυτά στέλνονται και στην θύρα USB μέσω ενός στοιχείου που ελέγχει (Serial-Over-USB) αλλά και στο Pin 0 για να τα διαβάσει επίσης μια άλλη συσκευή. Αυτό φυσικά σημαίνει ότι αν στο Sketch μας ενεργοποιήσουμε το σειριακό Interface, χάνουμε 2 Digital εισόδους/εξόδους.

- Τα Pin 2 - 3 μπορούμε να τα χειριστούμε και ως εξωτερικά Interrupt (Interrupt : έχουμε την δυνατότητα να παρεμβάλουμε κάποιες άλλες εντολές να τρέχουν σχεδόν ταυτόχρονα με το βασικό σκέλος του Sketch μας). Για παράδειγμα, μπορούμε να τα ρυθμίσουμε μέσα από το Sketch μας να είναι ψηφιακές είσοδοι στις οποίες όταν γίνονται κάποιες τροποποιήσεις, η σειρά του Sketch να διακόπτετε αμέσως και να τίθεται σε λειτουργία μια άλλη συνάρτηση. Τα εξωτερικά Interrupt είναι ιδιαίτερα χρήσιμα σε εφαρμογές που απαιτούν συγχρονισμό μεγάλης ακρίβειας.

- Τα Pin 3, 5, 6, 9, 10 και 11 έχουν την δυνατότητα να δουλέψουν και ως εικονικοί αναλογικοί έξοδοι με το PWM (Pulse Width Modulation), δηλαδή το ίδιο σύστημα που χρησιμοποιούνται από τις μητρικές των υπολογιστών για να έχουν τον έλεγχο περιστροφής των ανεμιστήρων. Έτσι, μπορούμε να συνδέσουμε για παράδειγμα ένα Led σε κάποιο από αυτά τα Pin και να ελέγξουμε πλήρως το πόσο φωτεινό είναι με την ανάλυση 8Bit, αντί να έχουμε απλά την δυνατότητα ON/OFF που παρέχουν οι υπόλοιπές ψηφιακές έξοδοι.

Είναι σημαντικό να καταλάβουμε ότι το PWM δεν είναι στην πραγματικότητα αναλογικό και ότι θέτοντας στην έξοδο την τιμή 118, δεν σημαίνει ότι θα δίνει 2,5V αντί των 5V αλλά ότι θα δίνει ένα παλμό που θα εναλλάσσεται με υψηλή συχνότητα και για ίδιο χρονικό διάστημα μεταξύ Low και High .

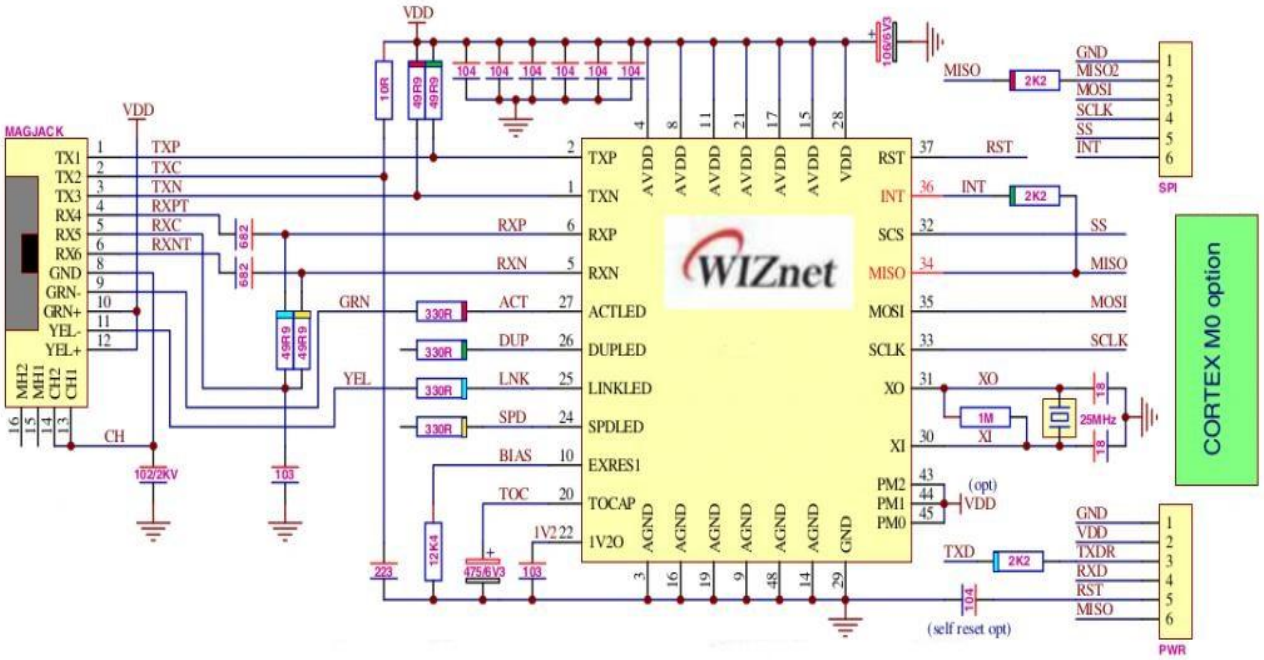


Εικόνα 4: Arduino UNO R3

Στην κάτω μεριά της πλακέτας, υπάρχει άλλη μια σειρά από 6 Pin. Αυτά λειτουργούν ως αναλογικοί είσοδοι χρησιμοποιώντας το ADC (Analog Digital Converter) που είναι πάνω στον μικροελεγκτή. Για παράδειγμα, μπορούμε να τροφοδοτήσουμε ένα από αυτά με μια τάση την οποία μπορούμε να πειράζουμε με ένα ποτενσιόμετρο, ας υποθέσουμε από 0V έως 5V Τότε από το Sketch μας μπορούμε να διαβάσουμε την τιμή του Pin ως ένα ακέραιο αριθμό. Η τάση αναφοράς μπορεί να ρυθμιστεί σε όποια τάση επιθυμούμε έως 5V. Τέλος, το καθένα από αυτά τα 6 Pin, μέσα από το Sketch μπορούν να μετατραπούν σε ψηφιακά Pin εισόδων/εξόδων όπως και τα 14 που είναι από την άλλη μεριά.

1.2 Ethernet Shield

1.2.1 Τι είναι η Ethernet Shield/Απαιτήσεις λειτουργίας.



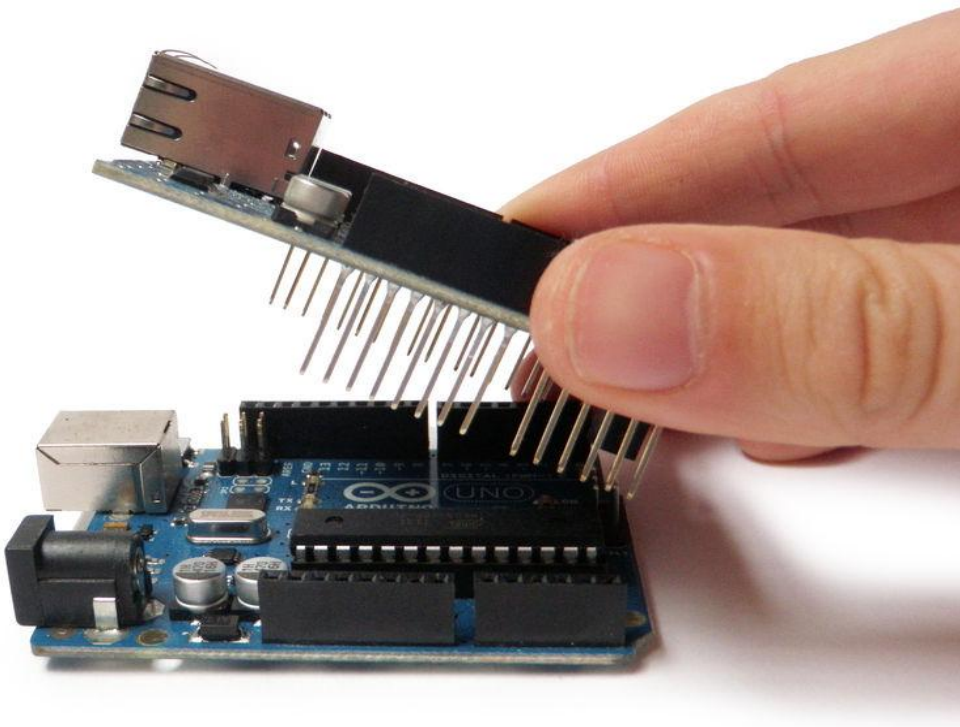
Εικόνα 5: Διάγραμμα W5100

Η Ethernet Shield συνδέει το Arduino μας στο διαδίκτυο σε λίγα λεπτά. Εφαρμόζουμε την Ethernet Shield πλακέτα, πάνω στην πλακέτα Arduino μας, έπειτα συνδέουμε με ένα καλώδιο RJ45 και ακολουθούμε μερικές απλές οδηγίες για να ξεκινήσουμε τον έλεγχο μέσω του διαδικτύου. Τα πάντα στο Arduino, λογισμικό και Sketch είναι ελεύθερα και διαθέσιμα (ανοικτοί κώδικες). Αυτό σημαίνει ότι μπορούμε να μάθουμε από ήδη έτοιμα προγράμματα πώς ακριβώς μπορούμε να δημιουργήσουμε ένα Webserver και να τα αναπτύξουμε για το σχεδιασμό ενός δικού μας κυκλώματος. Πάρα πολλά Sketch Arduino ήδη τροφοδοτούν τη δημιουργικότητα των ανθρώπων σε όλο τον κόσμο.

Για να τεθεί σε λειτουργία :

- Απαιτεί μια πλακέτα Arduino
- 5V Τάση λειτουργίας
- Ταχύτητα σύνδεσης: 10 / 100Mb
- Σύνδεση με Arduino στη θύρα SPI

Η Wiznet W5100 παρέχει ένα δίκτυο (IP). Υποστηρίζει έως τέσσερις συνδέσεις μαζί. Χρησιμοποιούμε τη βιβλιοθήκη Ethernet για να συνδεθούμε στο Internet χρησιμοποιώντας την πλακέτα Ethernet. Η Ethernet Shield συνδέεται με την πλακέτα Arduino εφαρμόζοντας όλα της τα Pin στα Pin του Arduino. Αυτό κρατά τη διάταξη τους άθικτη και επιτρέπει η μια στην άλλη άμεση επικοινωνία. Η Ethernet Shield έχει ένα πρότυπο RJ-45 σύνδεσης, με ενσωματωμένο μετασχηματιστή γραμμής.



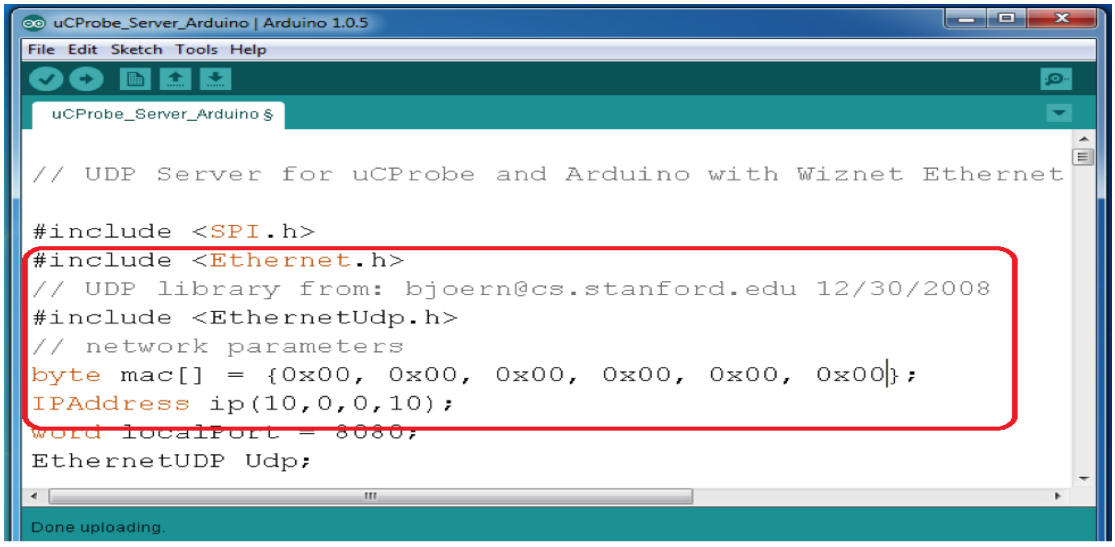
Εικόνα 6: Τρόπος σύνδεσης Ethernet Shield με Arduino

Υπάρχει μια ενσωματωμένη υποδοχή κάρτας Micro-SD, το οποίο μπορεί να χρησιμοποιηθεί για την αποθήκευση αρχείων. Για παράδειγμα θα μας εξυπηρετούσε για την χρήση κάποιας ιστοσελίδας. Είναι συμβατό με το Arduino Uno και Mega (με τη χρήση της βιβλιοθήκης Ethernet). Η κάρτα Micro-SD γίνετε προσβάσιμη μέσω της βιβλιοθήκης SD. Η Ethernet Shield έχει Power Over Ethernet (PoE) ενότητα σχεδιασμένη για να εξάγει ενέργεια από το καλώδιο συνεστραμμένου ζεύγους κατηγορίας 5.

1.2.2 Ρυθμίσεις δικτύου.

Για την Ethernet Shield θα πρέπει να καθοριστεί μέσα στο Sketch για την κάθε μία ξεχωριστά, μια διεύθυνση Mac (αναγράφετε πιθανόν σε αυτοκόλλητο στο κάτω μέρος της) και μια σταθερή διεύθυνση IP (συγκεκριμένη εσωτερική διεύθυνση του Router) χρησιμοποιώντας την εντολή Ethernet.begin (). Η διεύθυνση Mac είναι ένα μοναδικό αναγνωριστικό για μια συγκεκριμένη συσκευή, ωστόσο για τις πιο παλιές Ethernet Shield που δεν γνωρίζουμε την ειδική διεύθυνση Mac, θα πρέπει να υπάρξουν κάποιες δοκιμές για να την εφεύρουμε. Έπειτα όσον αφορά την διεύθυνση IP είμαστε υποχρεωμένοι να την έχουμε στα υπόψιν μας και να είναι έγκυρη, αυτό εξαρτάτε από τη διαμόρφωση του δικτύου μας.

Είναι δυνατόν να χρησιμοποιήσουμε και το πρωτόκολλο DHCP για να βάλουμε δυναμική IP. Μία ακόμα σημαντική πληροφορία για την Ethernet Shield είναι ότι μπορούμε να προσδιορίσουμε μονό μία IP . Για παράδειγμα αν επιθυμούμε να δημιουργήσουμε ένα πρόγραμμα το οποίο στον ιστότοπο θα περιέχει πολλαπλά interface και ρυθμίσεις, τότε θα αναγκαστούμε να δουλέψουμε σε μία IP και πάνω σε αυτήν να ξεχωρίσουμε σελίδες (όπως 192.168.178.1/otidhpote).

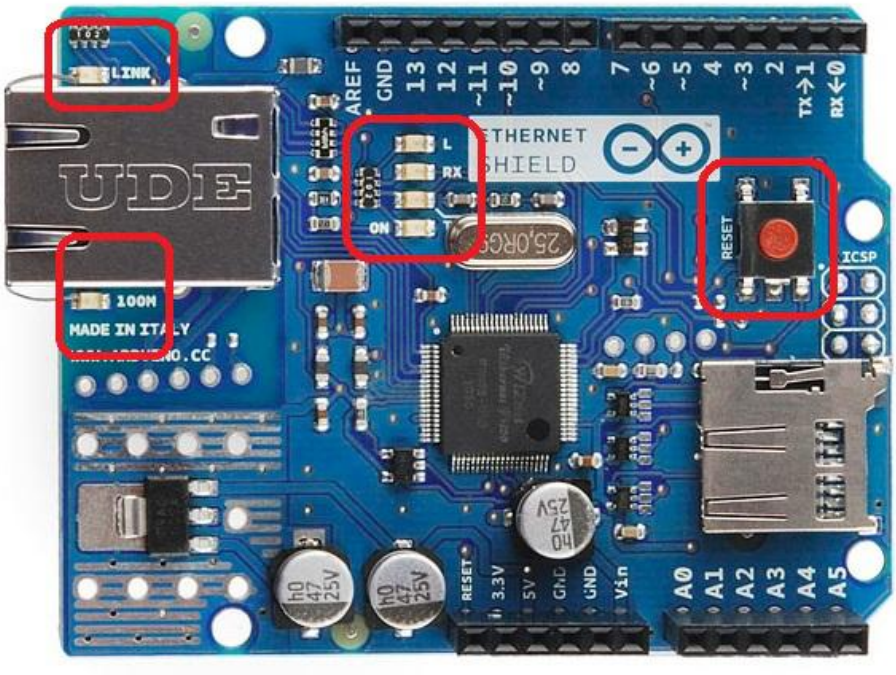


Εικόνα 7: Τροποποιήσεις για την Ethernet Shield

1.2.3 Περιγραφή πλακέτας Ethernet Shield.

Το Arduino επικοινωνεί με την W5100 και την κάρτα SD, χρησιμοποιώντας το SPI . Αυτό είναι για τα ψηφιακά Pin 10, 11, 12, και 13 για το Uno και για το Mega 50, 51, και 52. Μάλιστα και στα δύο Arduino τα Pin 10 χρησιμοποιούνται για την επιλογή του W5100 και το Pin 4 για την κάρτα SD. Αυτά τα Pin δεν μπορούν να χρησιμοποιηθούν για οποιαδήποτε ανάγκη μας. Η Ethernet Shield μας παρέχει ένα πρότυπο, το RJ45 Ethernet Jack.. Η Ethernet Shield περιλαμβάνει:

- Το κουμπί Reset πάνω στην Ethernet Shield επαναφέρει και τον W5100 αλλά και την πλακέτα Arduino.
- PWR: υποδεικνύει ότι η πλακέτα τροφοδοτείτε.
- LINK: δείχνει την παρουσία ενός δικτύου και αναβοσβήνει όταν η ασπίδα μεταδίδει ή λαμβάνει δεδομένα.
- FULLD: υποδεικνύει ότι η σύνδεση δικτύου είναι Full Duplex.
- 100M: δείχνει την ταχύτητα σύνδεσης με το δίκτυο αν είναι 100 Mb / s .
- RX: αναβοσβήνει όταν η πλακέτα λαμβάνει δεδομένα.
- TX: αναβοσβήνει όταν η πλακέτα στέλνει δεδομένα.
- COLL: αναβοσβήνει όταν υπάρχουν δυσλειτουργίες δίκτυο.



Εικόνα 8: Ethernet Shield

1.3 Αισθητήρες και εξαρτήματα

1.3.1 Γενικά.

Αισθητήρας είναι ένας μετατροπέας που το είδος του εξαρτάτε για ποιόν ακριβώς σκοπό έχει δημιουργηθεί. Ανιχνεύει γεγονότα ή αλλαγές στο περιβάλλον και δίνει μια αντίστοιχη έξοδο, γενικά ως ένα ηλεκτρικό ή οπτικό σήμα. Για παράδειγμα, ένα θερμοστοιχείο μετατρέπει την θερμοκρασία σε μια τάση. Οι αισθητήρες θα πρέπει να σχεδιάζονται έτσι ώστε να έχουν μικρή ευαισθησία σε ότι μετράνε, καθιστώντας τον αισθητήρα βελτιωμένο και πολύ εξυπηρετικό. Η τεχνολογική πρόοδος επιτρέπει όλο και περισσότερο στους αισθητήρες να κατασκευάζονται σε μικροσκοπική κλίμακα, όπως οι Microsensors χρησιμοποιώντας την τεχνολογία MEMS. Στις περισσότερες περιπτώσεις, ένας Microsensors φθάνει σε μία πολύ μεγαλύτερη ταχύτητα και ευαισθησία σε σχέση με τους υπόλοιπους. Ωστόσο, όλοι οι αισθητήρες έχουν τις δικές τους μεθόδους διασύνδεσης οπότε το άτομο που θα τους εγκαταστήσει θα πρέπει να είναι σχετικό με τον κλάδο.

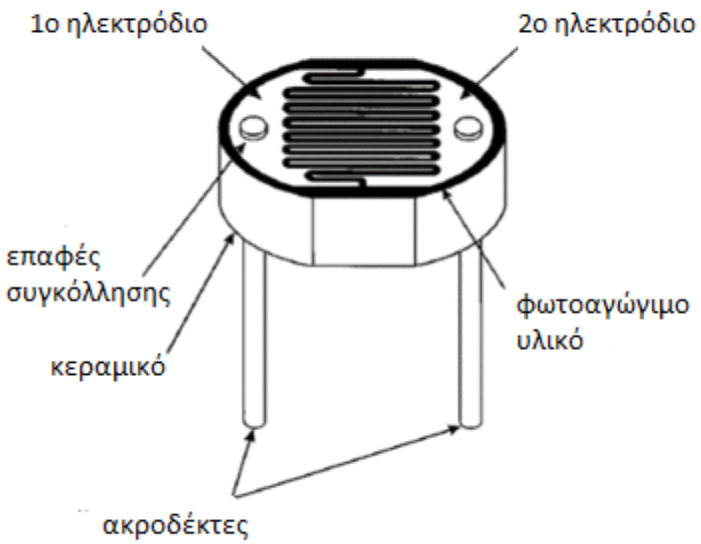
1.3.2 Φωτοκύτταρα.

Τα φωτοκύτταρα είναι οι αισθητήρες που ανιχνεύουν το φως. Είναι μικρά, φθηνά, χαμηλής ισχύος, εύκολο στη χρήση και δεν φθείρονται. Για το λόγο αυτό εμφανίζονται συχνά στα παιχνίδια, Gadgets και συσκευές. Αναφέρονται συχνά και ως CdS κύτταρα (που είναι κατασκευασμένα από κάδμιο και σουλφίδιο) και (LDR) φωτοαντιστάσεις.

Εικόνα 9: Φωτοκύτταρο

Το φωτοκύτταρο είναι βασικά μια αντίσταση που μεταβάλλεται η τιμή της ανάλογα με το πόσο φως πέφτει πάνω στην επιφάνειά της. Είναι πολύ χαμηλό το κόστος, προσαρμόζεται σχεδόν σε οποιαδήποτε ανάγκη μας εφόσον παρασκευάζεται σε πολλά μεγέθη και προδιαγραφές, μα είναι και αρκετά ανακριβής. Κάθε φωτοκύτταρο θα λειτουργεί λίγο διαφορετικά από το άλλο ακόμα και αν είναι του ίδιου τύπου. Επίσης για την εγκατάσταση του, θα πρέπει το μέρος να είναι κατάλληλο και να μην υπάρχει κάποιο εμπόδιο που να μην επιτρέπει την διέλευση της μετρούμενης δέσμης. Τρόπος δοκιμής:

Ο ευκολότερος τρόπος για να διαπιστώσουμε πώς και αν λειτουργεί το φωτοκύτταρο μας είναι να συνδέσουμε ένα πολύμετρο (ρυθμισμένο στα Ωμ) στην αντίσταση μέτρησης, στους δύο αγωγούς και να δούμε πώς η αντίσταση αλλάζει όταν πέσει πάνω της σκιά είτε σβήνοντας τα φώτα είτε με το χέρι μας. Επειδή η αντίσταση αλλάζει πολύ, μια αυτόματη κλίμακα στο πολύμετρο θα μας εξυπηρετούσε. Διαφορετικά, αν δεν διαθέτουμε τέτοιο πολύμετρο απλά φροντίζουμε να δοκιμάσουμε διαφορετικές κλίμακες, μεταξύ 1ΜΩ - 1ΚΩ .



Εικόνα 10: Ανάλυση φωτοκύτταρου

1.3.3 Αισθητήρας θερμοκρασίας.

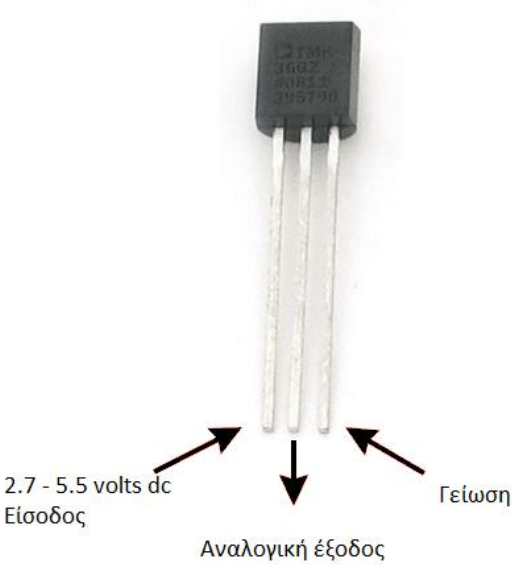
Ένας αναλογικός αισθητήρας θερμοκρασίας είναι αρκετά εύκολο να εξηγηθεί, είναι ένα εξάρτημα που μας λέει ποιά είναι η θερμοκρασία του περιβάλλοντος την κάθε στιγμή. Οι αισθητήρες αυτοί χρησιμοποιούν μια τεχνική στερεάς κατάστασης για τον προσδιορισμό της θερμοκρασίας. Δηλαδή, δεν χρησιμοποιούν υδράργυρο (όπως και τα παλιά θερμόμετρα). Στην θέση τους, χρησιμοποιούν το γεγονός ότι όσο η θερμοκρασία μεταβάλλεται, ταυτόχρονα τόσο μεταβάλλεται και η τάση της διόδου κατά ένα γνωστό ρυθμό (Βασικά αυτό είναι η πτώση τάσης μεταξύ της βάσης και εκπομπού - το Vbe -ενός τρανζίστορ). Με μια ακριβής ενίσχυση στην τάση καθώς μεταβάλλεται, είναι εύκολο να παράγει ένα αναλογικό σήμα που είναι ευθέως ανάλογη προς τη θερμοκρασία. Υπάρχουν βέβαια και ορισμένες βελτιώσεις όσον αφορά το τεχνικό και το κατασκευαστικό μέρος του αισθητήρα. Επειδή αυτοί οι αισθητήρες δεν έχουν κινούμενα μέρη, δεν έχουν μεγάλα σφάλματα στο αποτέλεσμα, δεν φθείρονται, δεν χρειάζονται βαθμονόμηση και λειτουργούν άψογα υπό πολλές περιβαλλοντικές συνθήκες. Επιπλέον είναι πολύ φθηνοί και είναι αρκετά εύκολοι στη χρήση.

Περίπτωση προβλήματος.

Αν για παράδειγμα θελήσουμε να προσθέσουμε περισσότερους αισθητήρες και διαπιστώσουμε ότι η θερμοκρασία είναι ασυνεπής, τότε αυτό δείχνει ότι οι αισθητήρες παρεμβάλλονται μεταξύ τους κατά την ενεργοποίηση του κυκλώματος.

Τρόπος δοκιμής.

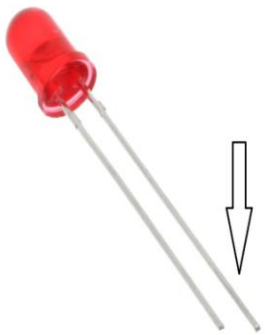
Ο έλεγχος αυτών των αισθητήρων είναι αρκετά εύκολος, αλλά θα χρειαστεί μια μπαταρία ή ένα τροφοδοτικό. Συνδέουμε μια παροχή 2.7-5.5V , η γείωση να συνδέεται με τον ακροδέκτη 3 (δεξιός ακροδέκτης) και δίνουμε τροφοδοσία στον ακροδέκτη 1 (αριστερός ακροδέκτης). Στην συνέχεια, συνδέουμε το πολύμετρο μας σε κλίμακα συνεχούς τάσης προς τη γείωση και τον ακροδέκτη 2 (μεσαίος ακροδέκτης). Εάν έχουμε έναν TMP36 και η θερμοκρασία είναι περίπου δωματίου (25 ° C), τότε η τάση θα πρέπει να είναι περίπου 0.75V. Να σημειώσουμε ότι αν χρησιμοποιούμε έναν LM35, τότε η τάση θα είναι 0,25V. Επίσης μπορούμε να μεταβάλλουμε λίγο την θερμοκρασία τρίβοντας σιγά το πλαστικό περίβλημα του αισθητήρα με το χέρι μας και τότε θα δούμε την αύξηση της θερμοκρασίας / τάσης. Στην παρακάτω εικόνα βλέπουμε το ρόλο του κάθε ακροδέκτη.



Εικόνα 11: Αισθητήρας θερμοκρασίας

1.3.4 LED.

Ένα Led είναι μια δίοδος εκπομπής φωτός το οποίο αποτελείται από μόλυβδο. Όταν μια κατάλληλη τάση εφαρμόζεται στα ηλεκτρόδια, τα ηλεκτρόνια είναι σε θέση να απελευθερώνουνε ενέργεια με τη μορφή φωτονίων. Αυτό το φαινόμενο ονομάζεται ηλεκτροφωταύγεια και το χρώμα του φωτός καθορίζεται από το ενεργειακό χάσμα ζώνης του ημιαγωγού. Τα Led έχουν πολλά πλεονεκτήματα σε σχέση με το φως πυρακτώσεως, όπου κάποια από αυτά είναι η χαμηλότερη κατανάλωση ενέργειας, μεγαλύτερη διάρκεια ζωής, μικρότερο μέγεθος, και ταχύτερη εναλλαγή.

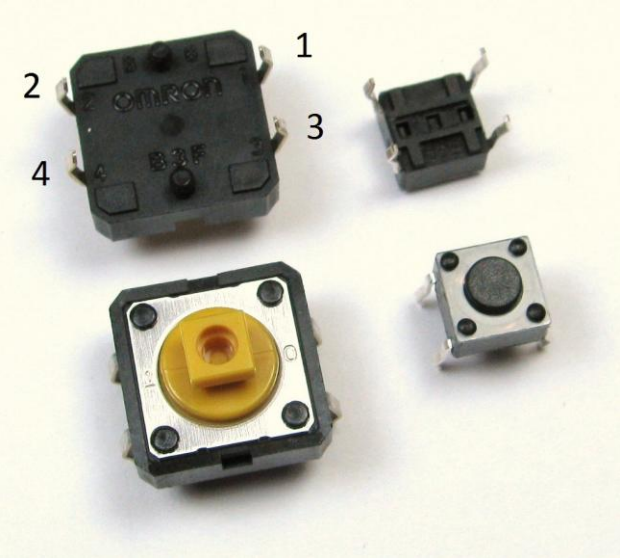


Εικόνα 12: Led

Ο τρόπος σύνδεσης ενός Led είναι συγκεκριμένος. Βλέποντας την παραπάνω εικόνα, παρατηρούμε ότι ο ένας ακροδέκτης από τους δύο επισημαίνεται και αυτός είναι ο μακρύτερος. Λοιπόν, αυτός ο ακροδέκτης είναι που πρέπει να τοποθετηθεί στην τροφοδοσία (+) και ο άλλος στη γείωση (-) . Για τον λόγο ότι το Led είναι μια δίοδος, σε μια διαφορετική συνδεσμολογία δεν θα μπορούσε να ανάψει.

1.3.5 Button Arduino.

Το μπουτόν με τους 4 ακροδέκτες θα μπορούσαμε να το θεωρήσουμε εύκολα ίσο με δύο μπουτόν. Αποτελείται από δύο διακόπτες που κλείνουν με το πάτημα ενός κουμπιού. Για να το χρησιμοποιήσουμε θα πρέπει να γνωρίζουμε το διάγραμμά του.



Εικόνα 13: Button

Αν κοιτάξουμε το μπουτόν θα παρατηρήσουμε ότι έχει δύο ζεύγη ακροδεκτών σε αντίθετες πλευρές. Το ζεύγος των ακροδεκτών το οποίο είναι σε κάθε πλευρά, είναι συνδεδεμένο με ένα διακόπτη. Σε αυτήν την περίπτωση άλλο διακόπτη έχουν οι ακροδέκτες 2 - 4 και άλλο οι 1 - 3. Το μπουτόν κλείνει και τους δύο διακόπτες ταυτόχρονα όταν πιέζεται, ενώ όταν το αφήσουμε επανέρχεται στην αρχική του κατάσταση

Εικόνα 14: Διάγραμμα Button

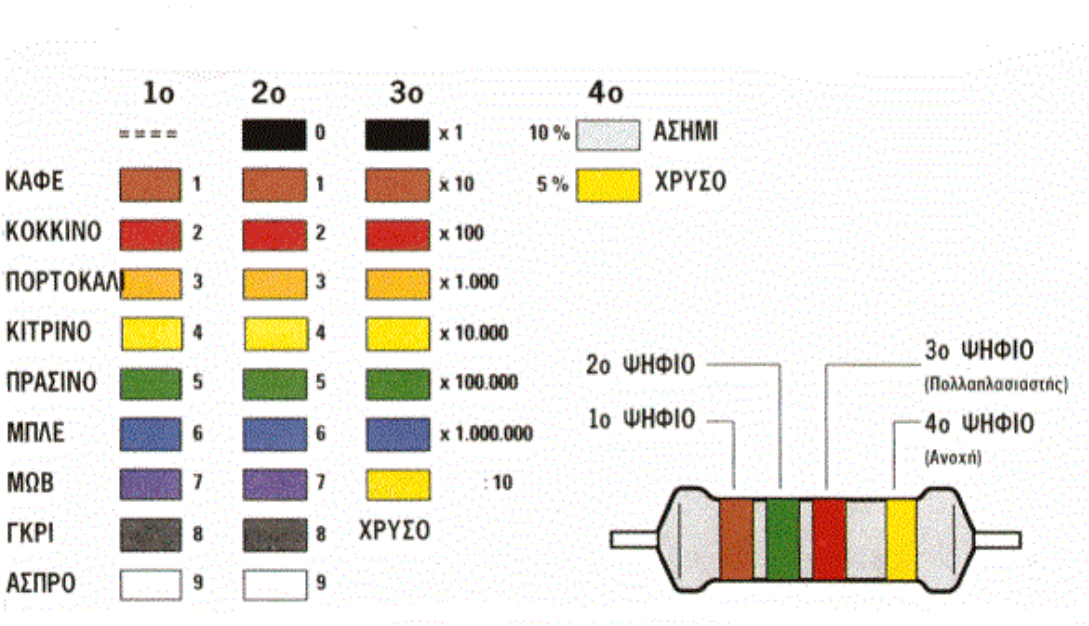
Από το διάγραμμα μπορούμε να συμπεράνουμε ότι, οι ακροδέκτες 1 – 2 είναι βραχυκυκλωμένοι το ίδιο και οι 3 – 4. Οπότε θα πρέπει να είμαστε αρκετά προσεκτικοί για την συνδεσμολογία και πλέον όταν θα έχουμε ολοκληρώσει το κύκλωμά μας, να είμαστε σίγουροι πριν το τροφοδοτήσουμε. Σε περίπτωση λάθος συνδεσμολογίας του μπουτόν, υπάρχει μεγάλος κίνδυνος να αχρηστέψουμε κάποια από τα εξαρτήματά μας ή ακόμα και να μας πάθει βλάβη η πλακέτα Arduino, εφόσον πλέον υπάρχει βραχυκύκλωμα.

1.3.6 Αντιστάσεις (Resistors).

Εικόνα 15: Αντιστάσεις

Η αντίσταση είναι ένα εξάρτημα, που ο ρόλος της σε ένα κύκλωμα είναι να μειώνει τη ροή της ηλεκτρικής ενέργειας. Επιτυγχάνεται αυτό, από το υλικό το οποίο είναι παρασκευασμένη η αντίσταση, μετατρέποντας ένα μέρος της ηλεκτρικής ενέργειας σε θερμότητα και άλλα. Για παράδειγμα αν βάλουμε μια αντίσταση σε σειρά με ένα εξάρτημα, όπως ένα Led, η αντίσταση θα δυσκολέψει την διέλευση της ηλεκτρικής ενέργειας η οποία θα περνούσε και ως αποτέλεσμα είναι το Led να δεχθεί λιγότερη ενέργεια. Με αυτό τον τρόπο παρατείνουμε τον χρόνο ζωής του διότι αν το είχαμε συνδέσει απευθείας χωρίς αντίσταση, θα ήταν πιο φωτεινό και γρήγορα θα καιγότανε. Η ικανότητα αυτή της αντίστασης, μας δίνει τη δυνατότητα να προμηθεύσουμε τα εξαρτήματά μας με το ποσοστό ενέργειας που χρειάζονται.

Πώς διαβάζονται οι κωδικοί των χρωμάτων στις αντιστάσεις.



Εικόνα 16: Πίνακας χρωμάτων

Η τιμή της αντίστασης σημειώνεται πάνω της χρησιμοποιώντας χρώματα. Κάθε χρώμα είναι και ένας διαφορετικός αριθμός. Μπορούμε να θυμόμαστε αυτούς τους αριθμούς ή μπορούμε να κοιτάμε μέσα από το ενημερωτικό φυλλάδιο για την επιλογή της κάθε μίας αντίστασης. Παραπάνω είναι ο πίνακας με τα χρώματα και τους αριθμούς. Όπως μπορούμε να δούμε είναι:

ΜΑΥΡΟ: 0

ΚΑΦΕ: 1

ΚΟΚΚΙΝΟ: 2

ΠΟΡΤΟΚΑΛΙ: 3

ΚΙΤΡΙΝΟ: 4

ΜΠΛΕ: 6

VIOLET: 7

ΓΚΡΙ: 8

ΛΕΥΚΟ: 9

ΠΡΑΣΙΝΟ: 5

Από δεξιά προς τα αριστερά, το δεύτερο χρώμα είναι πολλαπλασιαστής. Ο αριθμός από τον πρώτο χρωματισμό που βρίσκεται πάνω στην αντίσταση πρέπει να πολλαπλασιάζεται με τον αριθμό αυτού του χρώματος:

ΜΑΥΡΟ: 1

ΚΑΦΕ: 10

ΚΟΚΚΙΝΟ: 100

ΠΟΡΤΟΚΑΛΙ: 1000

ΚΙΤΡΙΝΟ: 10000

ΠΡΑΣΙΝΟ: 100000

ΜΠΛΕ: 1000000

ΧΡΥΣΟ: 0,1

ΑΣΗΜΙ: 0,01

Το επόμενο χρώμα είναι η ανοχή. Η ανοχή είναι η ακρίβεια του αντιστάτη και δίνεται ως ποσοστό. Για παράδειγμα, μια αντίσταση 390 Ω. με ανοχή ± 10 % θα έχει μια τιμή, μεταξύ 390 - 39 = 351 και 390 + 39 = 429 (39 είναι το 10% των 390 Ω).

BROWN: 1%

ΚΟΚΚΙΝΟ: 2%

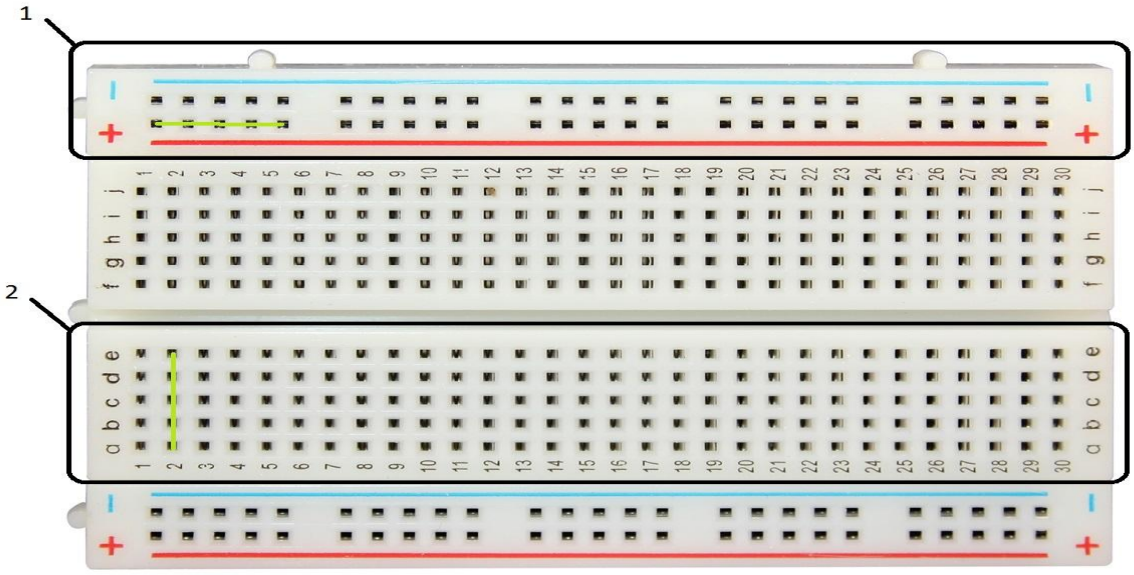
GOLD: 5%

SILVER: 10%

ΤΙΠΟΤΑ: 20%

1.3.7 Breadboard.

Το Breadboard είναι μια κατασκευή το οποίο μας επιτρέπει να συνδέουμε ηλεκτρονικά εξαρτήματα χωρίς κάποια συγκόλληση. Το υλικό του είναι ένα επεξεργασμένο κομμάτι ξύλου το οποίο κατά δεκαετία του 1970 το χρησιμοποιούσαν για τον τεμαχισμό του ψωμιού. Από εκεί προέρχεται και ο όρος Breadboard.



Εικόνα 17: Breadboard

Μέσα στο Breadboard τα Pin έχουν μία συγκεκριμένη διάταξη. Στην παραπάνω εικόνα, στο πλαίσιο «1», όλα τα Pin δεξιά και αριστερά είναι βραχυκυκλωμένα όπως είναι η πράσινη σημείωση. Ενώ, στο πλαίσιο «2», όλα τα Pin στη μέση είναι βραχυκυκλωμένα με διαφορετικό τρόπο και τον τρόπο αυτόν μπορούμε να τον διακρίνουμε επίσης από την πράσινη σημείωση που βρίσκεται εντός πλαισίου «2».

1.4 Περιβάλλον IDE Arduino

1.4.1 Sketch.

Στο ξεκίνημα θα δούμε ένα Sketch που αναβοσβήνει ένα Led. Σε αυτό το πρόγραμμα, θα μάθουμε πώς κάθε τμήμα του Sketch λειτουργεί. Ένα Sketch είναι το όνομα που χρησιμοποιεί το Arduino για ένα πρόγραμμα. Είναι η μονάδα του κώδικα που φορτώνεται και εκτελείται σε μια πλακέτα Arduino.

Οι πρώτες γραμμές του sketch Blink είναι ένα σχόλιο:

*

* Blink

*

Το βασικό παράδειγμα Arduino είναι να ανάβει η ενδεικτική λυχνία Led για ένα δευτερόλεπτο, Στη συνέχεια σβήνει για ένα δευτερόλεπτο, και ούτω καθεξής. Τα πάντα μεταξύ του / * - * / αγνοείται από το Arduino όταν εκτελείται το sketch (το * κατά την έναρξη της κάθε γραμμής είναι μόνο εκεί για να κάνει το σχόλιο, δεν απαιτείται). Αυτό βοηθά τους ανθρώπους να μάθουν ή να τροποποιούν τον κώδικά τους. Υπάρχει ένα άλλο στυλ για τα σχόλια μιας γραμμής. Αυτά ξεκινούν με // και συνεχίζουν μέχρι το τέλος της γραμμής. Για παράδειγμα, στη γραμμή:

int ledPin = 13; // LED που συνδέονται με ψηφιακά pin 13

το μήνυμα "LED που συνδέονται με ψηφιακά pin 13" είναι ένα σχόλιο.

1.4.2 Οι μεταβλητές.

Μια μεταβλητή είναι μια θέση για την αποθήκευση ενός κομματιού των δεδομένων. Έχει ένα όνομα, έναν τύπο, και μια τιμή. Για παράδειγμα, η γραμμή από το sketch Blink παραπάνω δηλώνει μια μεταβλητή, με την “ledPin” το όνομα, τον τύπο με το “int” και μια αρχική τιμή τον “13”. Το 13 είναι για να δηλώσουμε σε ποιο Pin Arduino είναι το Led συνδεδεμένο. Κάθε φορά που το όνομα “ledPin” εμφανίζεται στον κώδικα, εφόσον το έχουμε δηλώσει στην αρχή του Sketch, το καλούμε. Σε αυτή την περίπτωση αφού είναι δηλωμένο, το πρόσωπο που γράφει το πρόγραμμα θα μπορούσε να έχει επιλέξει να μην γράφει το όνομα της μεταβλητής ledPin και αντί αυτού να γράφει απλά 13 και αν χρειάζεται να ορίσετε έναν αριθμό Pin.

Το πλεονέκτημα της χρήσης μιας μεταβλητής είναι ότι είναι πιο εύκολο να αλλάξουμε την συνδεσμολογία του Led και να το οδηγήσουμε σε μια διαφορετική Pin, το μόνο που χρειάζεται είναι να επεξεργαστούμε την γραμμή που αντιστοιχεί στην αρχικοποίηση της μεταβλητής. Ωστόσο συχνά, η τιμή μιας μεταβλητής θα αλλάξει, ενώ το Sketch τρέχει. Για παράδειγμα, μπορείτε να αποθηκεύσετε την τιμή που διαβάζεται από μια είσοδο σε μια μεταβλητή.

1.4.3 Λειτουργίες.

Εδώ είναι ο ορισμός της λειτουργίας ρύθμισης από το παράδειγμα με το Led:

```
void setup ()

{

pinMode (ledPin, OUTPUT); // Θέτει το ψηφιακό Pin ως έξοδος

}
```

Η πρώτη γραμμή παρέχει πληροφορίες σχετικά με τη λειτουργία. Ο κώδικας μεταξύ των { , } ονομάζεται το Body της συνάρτησης, δηλαδή τι λειτουργία έχει. Μπορούμε να καλέσουμε μια συνάρτηση που έχει ήδη οριστεί είτε στο Sketch μας ή ως μέρος της γλώσσας Arduino. Για παράδειγμα, η γραμμή "pinMode (ledPin, OUTPUT);" καλεί τη συνάρτηση pinMode (). Αυτή η συνάρτηση χρησιμοποιείται από τη λειτουργία pinMode () για να αποφασίσει ποια Pin και πως θα λειτουργήσουν.

digitalWrite (), και η delay ()

Οι λειτουργίες digitalWrite () εξάγει μια τιμή . Για παράδειγμα, η γραμμή:

```
digitalWrite (ledPin, HIGH);
```

Ρυθμίσαμε το ledPin (Pin 13) σε High ή αλλιώς 5V. Γράφοντας ένα Pin σε Low συνδέεται με τη γείωση ή αλλιώς 0V.

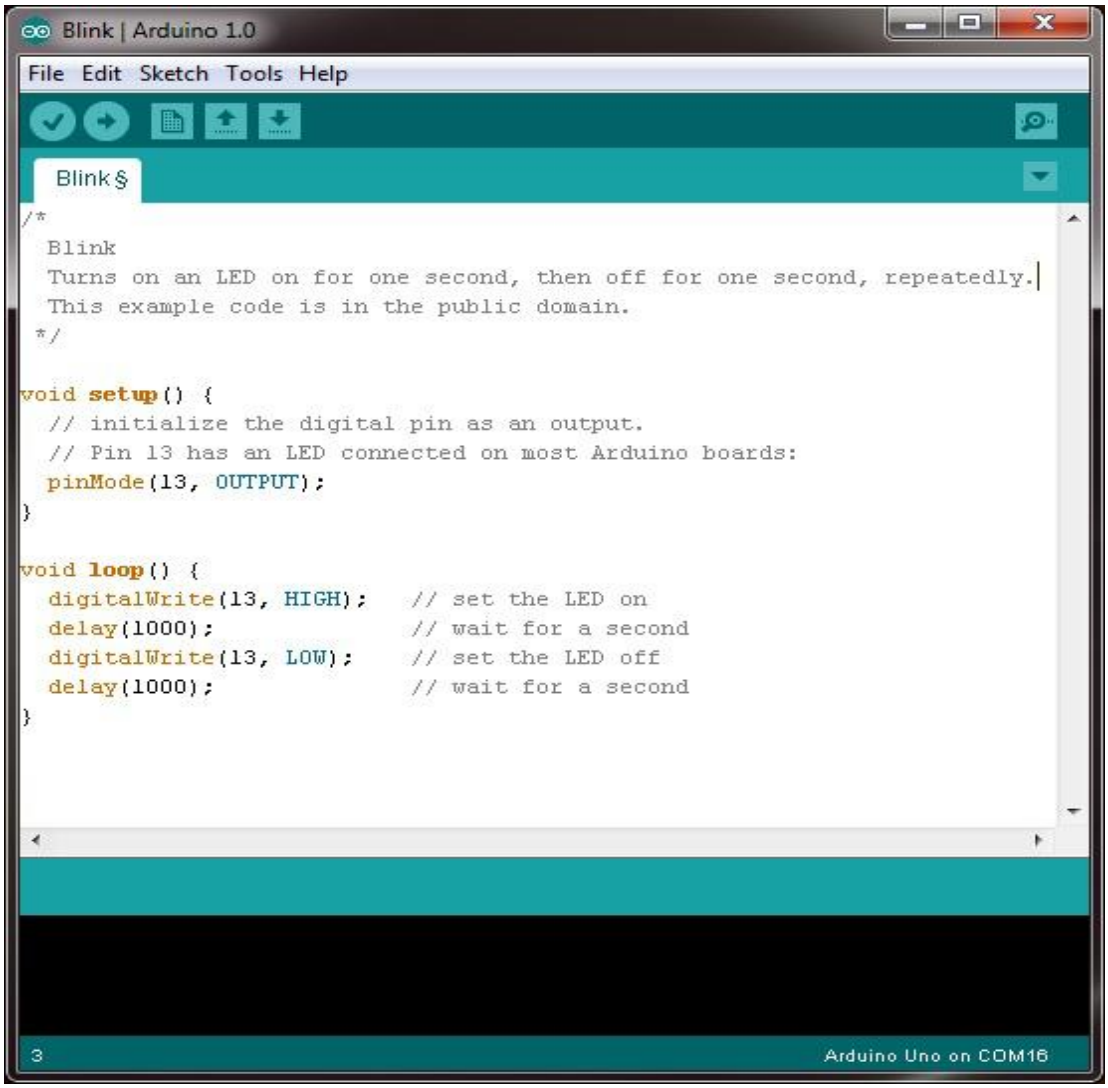
Η Delay δίνει εντολή στο Arduino να περιμένει για ένα καθορισμένο χρονικό διάστημα πριν συνεχίσουμε στην επόμενη γραμμή. Υπάρχουν 1000 χιλιοστά του δευτερολέπτου, έτσι ώστε η γραμμή:

```
delay (1000);
```

Να δημιουργεί μια καθυστέρηση ενός δευτερολέπτου.

setup () και loop ()

Υπάρχουν δύο ειδικές λειτουργίες που είναι ένα μέρος και μπορούμε να τις καλέσουμε μονό μια φορά σε κάθε Sketch Arduino , αυτές είναι οι void setup () και void loop (). Η void setup () καλείται μία φορά, όταν ξεκινά το Sketch. Είναι ένα καλό μέρος για να κάνει εργασίες εγκατάστασης, όπως τη ρύθμιση εισόδων/εξόδων ή την προετοιμασία των βιβλιοθηκών. Η λειτουργία void loop () καλείται ξανά και ξανά και είναι η καρδιά του Sketch. Θα πρέπει να περιλαμβάνουν και τις δύο λειτουργίες στο Sketch μας, ακόμα κι αν δεν τα χρειαζόμαστε για τίποτα.



Εικόνα 18: Παράδειγμα σε περιβάλλον arduino

ΚΕΦΑΛΑΙΟ 2Ο

2.1 Android

2.1.1 Τι είναι το Android.

Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα. Χρησιμοποιείται περισσότερο για φορητές συσκευές όπως Smartphones και Tablets. Οι εταιρίες που το ανέπτυξαν είναι η Google και ποιό μετά η Open Handset Alliance. Η Google, ανέπτυξε ένα λογισμικό που έχει ως βάση τα Linux, το οποίο το προώθησε σαν ευέλικτο και αναβαθμίσιμο σύστημα. Το λειτουργικό Android εμφανίστηκε πρώτη φορά, τέλει του έτους 2007 και έγινε διαθέσιμο το 2010.

2.1.2 Εφαρμογές Android.

Οι εφαρμογές για Android δημιουργούνται με στόχο την λειτουργικότητα και την καλύτερη εξυπηρέτηση του χρήστη. Η ηλεκτρονική γλώσσα το οποίο συντάσσεται μια Android εφαρμογή είναι της Java. Εφόσον διαθέτουμε μια αντίστοιχη συσκευή Android μπορούμε να περιηγηθούμε στο διαδίκτυο και να κατεβάσουμε οποιαδήποτε εφαρμογή επιθυμούμε, είτε δωρεάν είτε χρειαστεί να πληρώσουμε ορισμένα λεφτά. Η επιλογή θα γίνει ανάμεσα σε πάρα πολλές, για τον λόγο ότι υπάρχουν πάρα πολλές διαθέσιμες εφαρμογές στο διαδίκτυο. Τέλος, αφού μια εφαρμογή ληφθεί με επιτυχία και πλέον έχει εγκατασταθεί στην συσκευή μας, μπορούμε να την τρέξουμε.

2.1.3 Android SDK.

Το Android SDK (Software Developers Kit) είναι πρόγραμμα που μας παρέχει ένα σύνολο βιβλιοθηκών, αρχείων και εργαλείων το οποίο χρειαζόμαστε για την διαμόρφωση και την ολοκλήρωση μιας Android εφαρμογής. Οπότε είναι απαραίτητο να το εγκαταστήσουμε στον ηλεκτρονικό υπολογιστή με το οποίο πρόκειται να μελετήσουμε την Android εφαρμογή. Κάποια από αυτά που μας παρέχει το Android SDK είναι:

- Εξομοιωτής συσκευών (Android Virtual Machines)
- Τεκμηρίωση
- Κομμάτια από κώδικες

2.1.4 Εξέλιξη του Android.

Γενικά στις μέρες μας η εξέλιξη της τεχνολογίας αυξάνεται με αρκετά μεγάλους ρυθμούς. Παρόμοιους ρυθμούς συντηρεί και το Android. Η συσκευές Android έχουν μπει στην καθημερινότητα μας και διευκολύνουν την ζωή μας. Έτσι η ζήτηση από τους χρήστες γίνεται όλο ένα και ποιο πολύ. Εκτός από την απλή χρήση του, μπορεί να μας προσφέρει και άλλες υπηρεσίες όπως είναι η ψυχαγωγία, η εκπαίδευση, η αγορές, η επικοινωνία και άλλα. Επίσης στην Ελλάδα έχουν δημιουργηθεί και εφαρμογές android η οποίες παρέχουν ασφάλεια στους μικρότερους σε ηλικία χρήστες, με το πάτημα μονό ενός κουμπιού.

ΚΕΦΑΛΑΙΟ 3Ο

3.1 Πειραματικό μέρος

Στο πειραματικό μέρος θα ασχοληθούμε με το έξυπνο σπίτι. Ποιο συγκεκριμένα θα έχουμε την δυνατότητα να ανοίγουμε μια Android εφαρμογή, η οποία θα έχει ένα μενού που θα μπορούμε να ελέγχουμε της συσκευές και τους αισθητήρες που έχουμε τοποθετήσει στο σπίτι μας. Στην δική μας περίπτωση δεν θα χρησιμοποιήσουμε ένα κανονικό σπίτι, αλλά θα το υλοποιήσουμε σε μια μακέτα με μια κάτοψη σπιτιού στην οποία θα τοποθετήσουμε πάνω αισθητήρες, Led, ανεμιστήρα και άλλα. Για να το πραγματοποιήσουμε, θα χρησιμοποιήσουμε ένα Arduino Uno και την Ethernet Shield, η οποία τοποθετείτε πάνω στο Arduino Uno. Τους αισθητήρες και τα Led θα τα συνδέσουμε πάνω στα pin που έχει η Ethernet Shield. Έπειτα θα περάσουμε τον κώδικα στο Arduino με το αντίστοιχο λογισμικό που διαθέτει. Στην συνέχεια θα δημιουργήσουμε και την εφαρμογή Android η οποία θα επικοινωνεί με την Ethernet Shield και θα περιέχει το μενού της διαχείρισης μας. Τέλος για να έχουμε και απομακρυσμένο έλεγχο θα χρησιμοποιήσουμε ένα Dynamic Domain Name System (DDNS), για την ευκολία του χρήστη.

3.1.1 Εγκατάσταση προγράμματος Arduino.

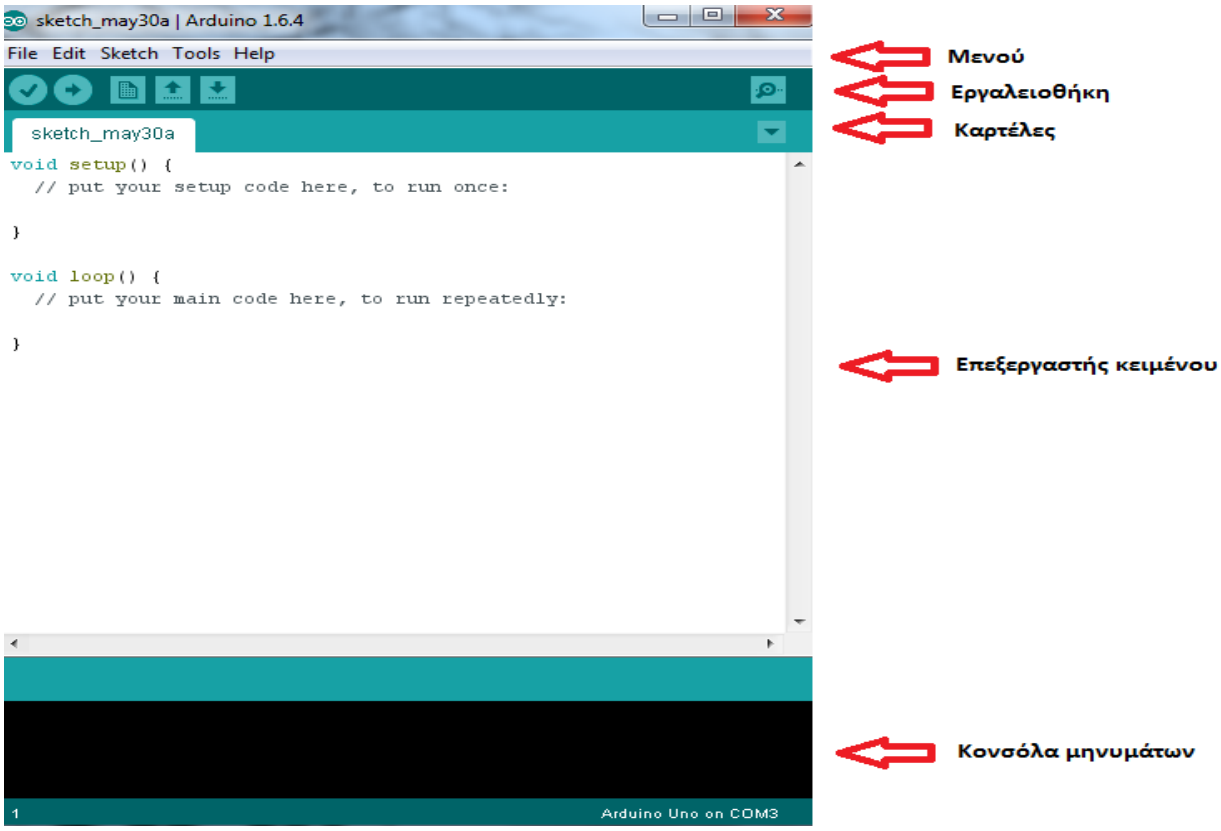
Το πρόγραμμα του Arduino είναι διαθέσιμο για λειτουργικά Windows, Mac, και Linux 32bits ή 64bits. Εμείς θα χρησιμοποιήσουμε Windows λειτουργικό. Θα κατεβάσουμε την τρέχουσα έκδοση του προγράμματος Arduino 1.6.4 από το επίσημο Site του Arduino στο ακόλουθο URL "http://www.arduino.cc/en/Main/Software" και το συμπιεσμένο αρχείο "arduino-1.6.4-windows". Που φαίνετε στην ακόλουθη εικόνα 19.

Εικόνα 19: Συμπιεσμένο αρχείο

Εφόσον κατεβάσουμε το συμπιεσμένο αρχείο που είναι χωρητικότητας 139 MB, έπειτα πρέπει να το κάνουμε αποσυμπίεση (Extract).Μόλις τελειώσει το Extract, δεν χρειάζεται κάποια άλλη ιδιαίτερη εγκατάσταση απλός είμαστε έτοιμοι να χρησιμοποιήσουμε το πρόγραμμα μας. Σε παλαιότερες εκδόσεις χρειαζόταν και άλλα βήματα για να κάνεις την εγκατάσταση. Όταν ανοίγουμε το φάκελο που κατεβάσαμε, μέσα υπάρχει ένα εικονίδιο Arduino Application που πατώντας επάνω του διπλό αριστερό κλικ μας ανοίγει το πρόγραμμα μας. Μέσα στο φάκελο υπάρχουν και άλλη φάκελοι όπως Libraries, Drivers, Examples, Java και άλλα τα οποία δεν πρέπει να τα πειράξουμε για την σωστή λειτουργία του προγράμματος.

3.1.2 Περιβάλλον ανάπτυξης του Arduino.

Το περιβάλλον ανάπτυξης είναι μια εφαρμογή γραμμένη σε Java. Έχει σχεδιαστεί για να εισάγει τον προγραμματισμό και στους νέους, που δεν είναι εξοικειωμένοι με την αναπτύξει προγραμματισμού. Περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα με χαρακτηριστικά όπως είναι η επισήμανση σύνταξης, ο συνδυασμός αγκυλών και είναι σε θέση να μεταγλωττίζει και να φορτώνει προγράμματα στην πλακέτα μόνο με ένα κλικ. Τα προγράμματα είναι γραμμένα σε γλώσσα C ή C++. Το περιβάλλον ανάπτυξης του Arduino αποτελείται από το μενού, την εργαλειοθήκη στην οποία τρέχουμε το πρόγραμμα στο Arduino, της καρτέλες που έχουμε ανοικτές, τον επεξεργαστή κειμένου στον οποίο γράφουμε το πρόγραμμα και την κονσόλα μηνυμάτων στην οποία βλέπουμε αν έχει περάσει το πρόγραμμα επιτυχώς ή αν έχει λάθη.



Εικόνα 20: Περιβάλλον ανάπτυξης

Κουμπιά εργαλειοθήκης:

Verify/Compile (Έλεγχος/ Μεταγλώττιση): Έλεγχος για λάθη στον κώδικα

Upload: Ανέβασμα του κώδικα στον μικροελεγκτή.

New(Νέο): Δημιουργεί ένα νέο Sketch

Open(Άνοιγμα): Παρουσιάζει ένα μενού με όλα τα Sketch.

Save(Αποθήκευση): Αποθηκεύει το Sketch

Serial Monitor(Σειριακή οθόνη): Ανοίγει την σειριακή οθόνη ώστε να μπορούμε να δώσουμε δεδομένα από το πληκτρολόγιο

3.1.3 Η δομή του προγράμματος.

Ένα πρόγραμμα Arduino πρέπει να έχει την παρακάτω δομή:

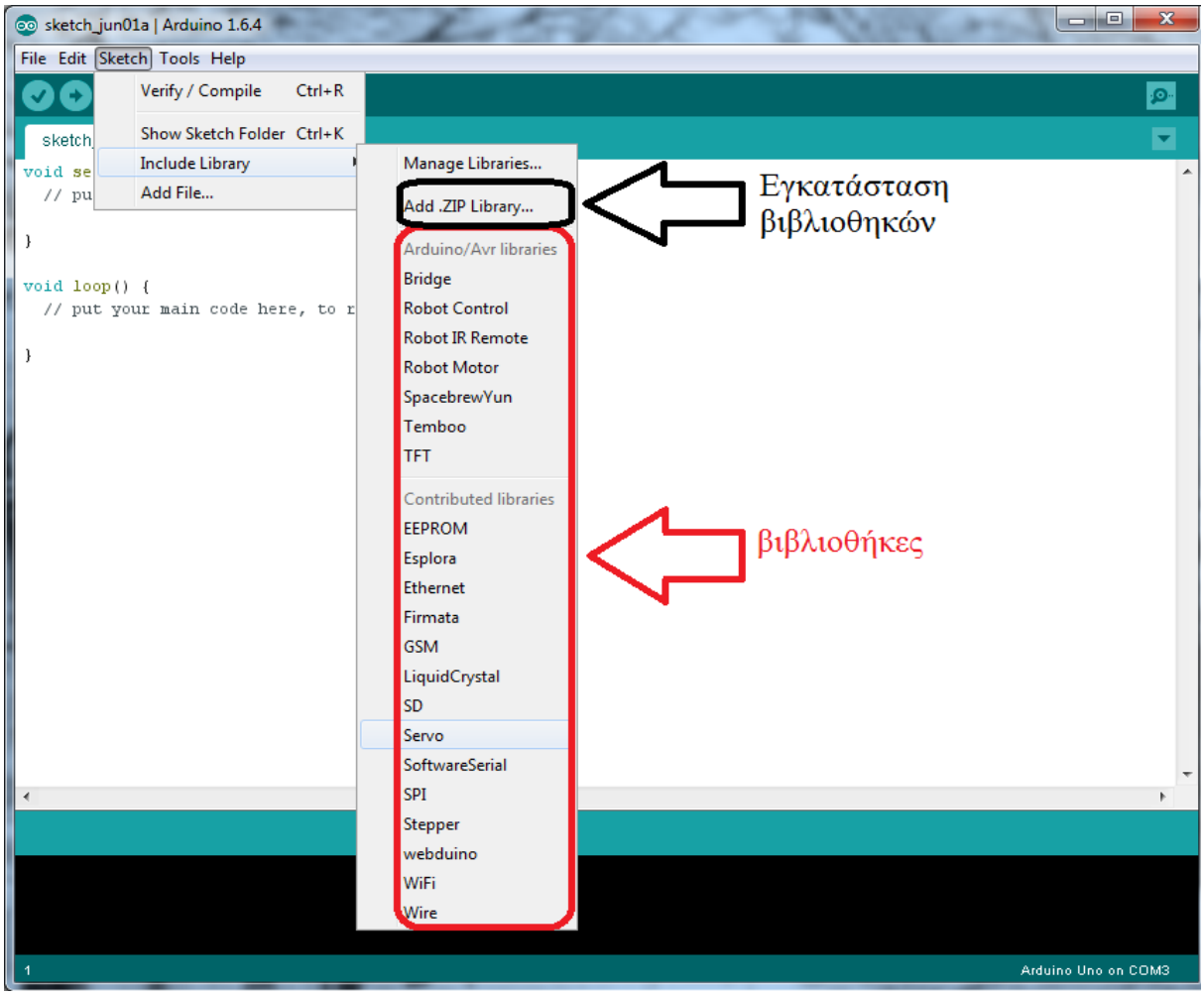
Δήλωση μεταβλητών

```
void setup() {
  // Αρχικοποίησης
}
void loop() {
  // Κώδικας
}
```

Πρέπει να αποτελείται από ένα void setup και ένα void loop. Επίσης δεν έχουμε την δυνατότητα να τρέχουμε πολλά Sketch ταυτόχρονα. Αξίζει να αναφέρουμε ότι όταν φορτώσουμε ένα Sketch στο arduino και μετά επιθυμούμε να φορτώσουμε ένα άλλο, το πρώτο Sketch δεν έχει την δυνατότητα να αποθηκευτεί στην μνήμη του και διαγράφετε αυτόματα με την εισαγωγή του νέου. Αν επιθυμήσουμε να το επαναφέρουμε πρέπει να το ξανατρέξουμε από την αρχή.

3.1.4 Εγκατάσταση βιβλιοθηκών στο πρόγραμμα.

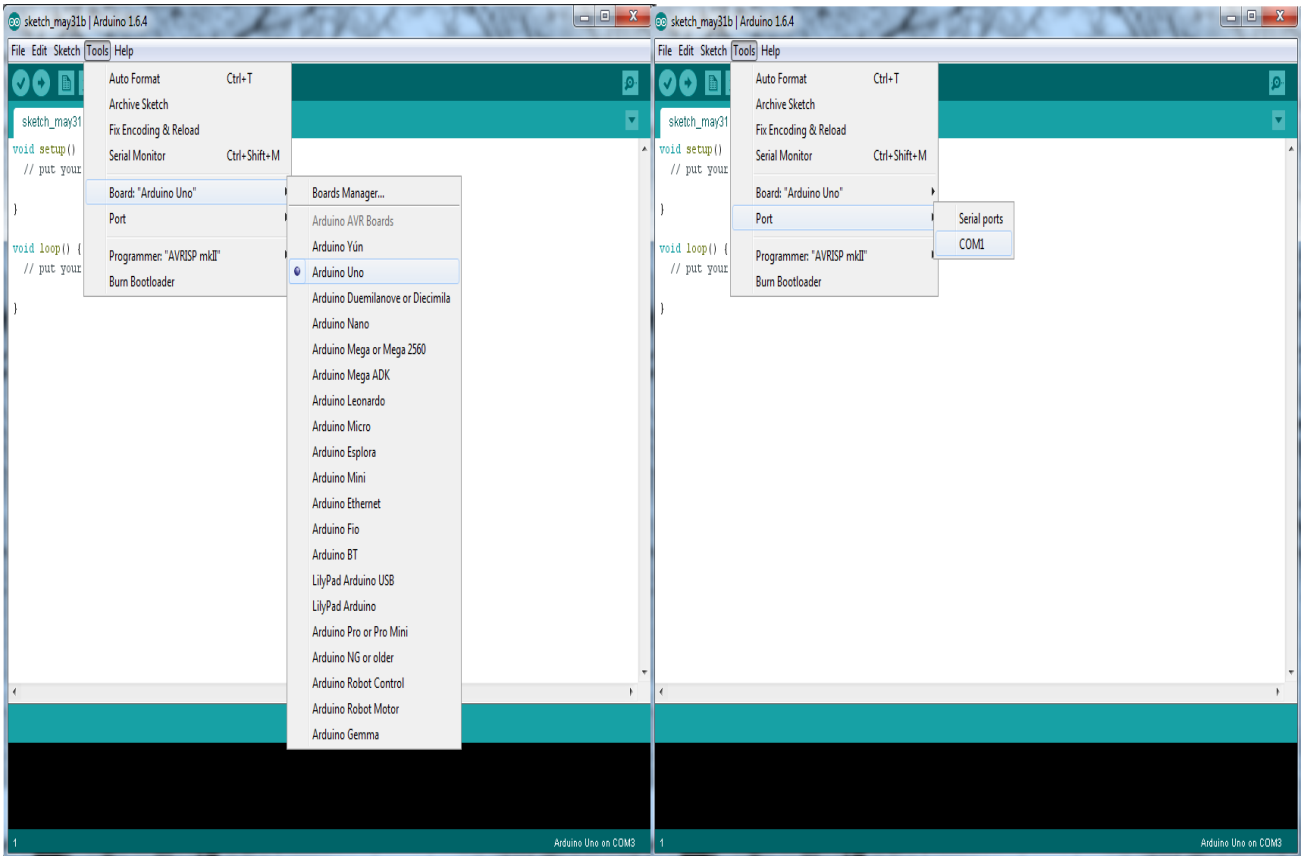
Το περιβάλλον Arduino μπορεί να επεκταθεί με την χρήση βιβλιοθηκών όπως και οι περισσότερες γλώσσες προγραμματισμού. Οι βιβλιοθήκες παρέχουν επιπλέον λειτουργικότητα στην χρήση των Sketch και αποτελούνται από έναν κώδικα που μας καθιστά εύκολο να συνδέσουμε μια οθόνη έναν αισθητήρα κ.τ.λ. Για να τρέξουμε τον κώδικα μιας βιβλιοθήκης πρέπει πρώτα να την καλέσουμε στο πρόσραμμα μας, αυτό φαίνετε στην εικόνα 21. Πατώντας στο μενού Sketch, Include Library και επιλέγοντας την βιβλιοθήκη που θέλουμε να χρησιμοποιήσουμε. Πολλές βασικές βιβλιοθήκες έχουν εγκατασταθεί ήδη με την εγκατάσταση του προγράμματος, αλλά εμείς έχουμε την δυνατότητα να επεκτείνουμε της βιβλιοθήκες, πατώντας στο μενού, Sketch, Include Library και επιλεγούμε Add Zip Library, έπειτα πρέπει να βρούμε την βιβλιοθήκη την οποία έχουμε κατεβάσει στον υπολογιστή μας από την επίσημη ιστοσελίδα του Arduino στο ακόλουθο URL " http://www.arduino.cc/en/Reference/Libraries" να την επιλέξουμε και να κάνουμε open.



Εικόνα 21: Βιβλιοθήκες

3.1.5 Επικοινωνία Arduino με περιβάλλον ανάπτυξης.

Για την σωστή επικοινωνία του Arduino με το περιβάλλον ανάπτυξης πρέπει να γίνουν κάποιες διαδικασίες. Στην δικιά μας περίπτωση θα χρησιμοποιήσουμε την πλακέτα Arduino Uno, την οποία πάνω της έχει εφαρμοστεί η πλακέτα Ethernet Shield. Έπειτα συνδέουμε το καλώδιο USB στον υπολογιστή και στην υποδοχή του Arduino. Για να αναγνωρίσει το πρόγραμμα ποιο Arduino χρησιμοποιούμε και σε ποια θύρα USB το έχουμε συνδέσει, πρέπει να το ρυθμίσουμε εμείς. Για την ρύθμιση του Arduino Uno πάμε στο μενού, Tools και επιλέγουμε στο Board το Arduino Uno. Για την σειριακή θύρα πάμε πάλι στο μενού, Tool και επιλέγουμε το Port για τον δικό μας υπολογιστή είναι το COM1. Αυτά φαίνονται και στην εικόνα 22 που ακολουθεί.



Εικόνα 22: Επιλογή Board και Port(σειριακής θύρας)

Σε κάποια windows χρειάζεται να κάνετε εγκατάσταση τους Drivers ώστε να αναγνωρίσει η σειριακή θύρα το Arduino, όμως στην δικιά μας περίπτωση δεν χρειάστηκε. Αν χρειαζόταν έπρεπε να κάνουμε τα συγκεκριμένα βήματα που βρίσκονται στο ακόλουθο URL το οποίο εξηγούνται αναλυτικά. "http://www.arduino.cc/en/Guide/UnoDriversWindowsXP".

3.1.6 Επικοινωνία Ethernet Shield με Router.

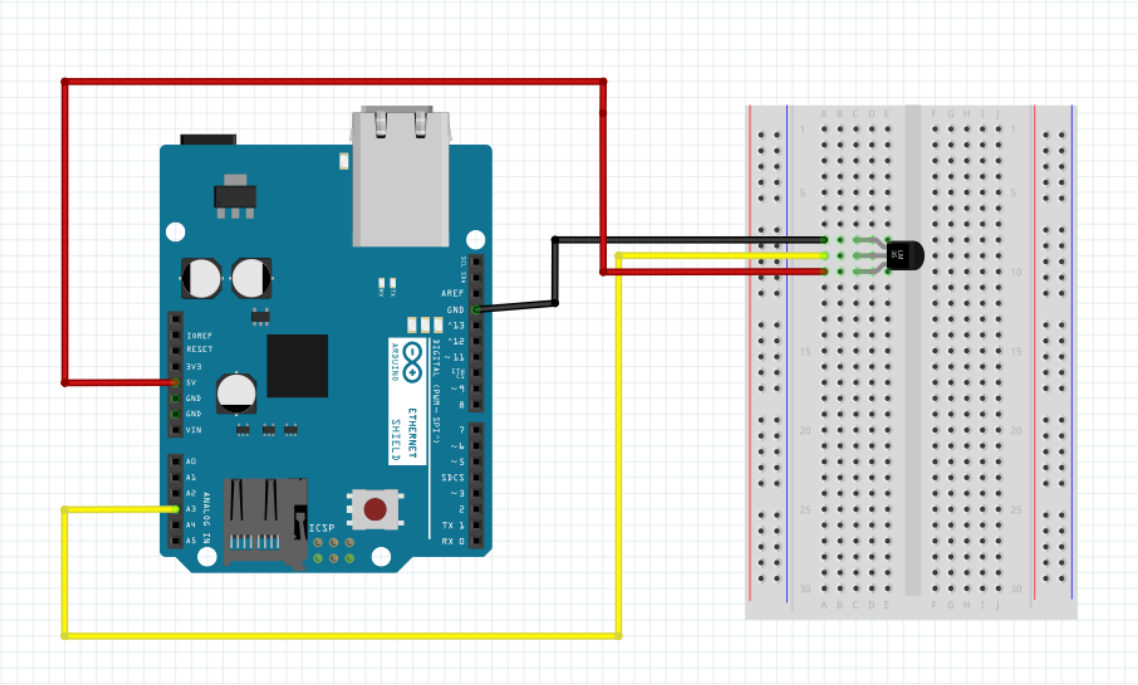
Για να συνδεθεί το Arduino Uno με το διαδίκτυο γίνεται με την βοήθεια της Ethernet Shield (εφόσον είναι τοποθετημένη πάνω στο Arduino Uno). Για να γίνει αυτό απλά συνδέουμε το καλώδιο RJ45 στην υποδοχή που έχει η πλακέτα Ethernet Shield με την Ethernet Port του Router. Επίσης πάνω στην Ethernet Shield έχουμε τοποθετήσει μια SD-Card στην υποδοχή που διαθέτει, για να αποθηκεύσουμε κάποια αρχεία που θα χρειαστούμε στο πείραμα μας.

3.1.7 Συνδεσμολογία.

Το επόμενο βήμα είναι να κάνουμε την συνδεσμολογία. Για την συνδεσμολογία θα χρειαστούμε:

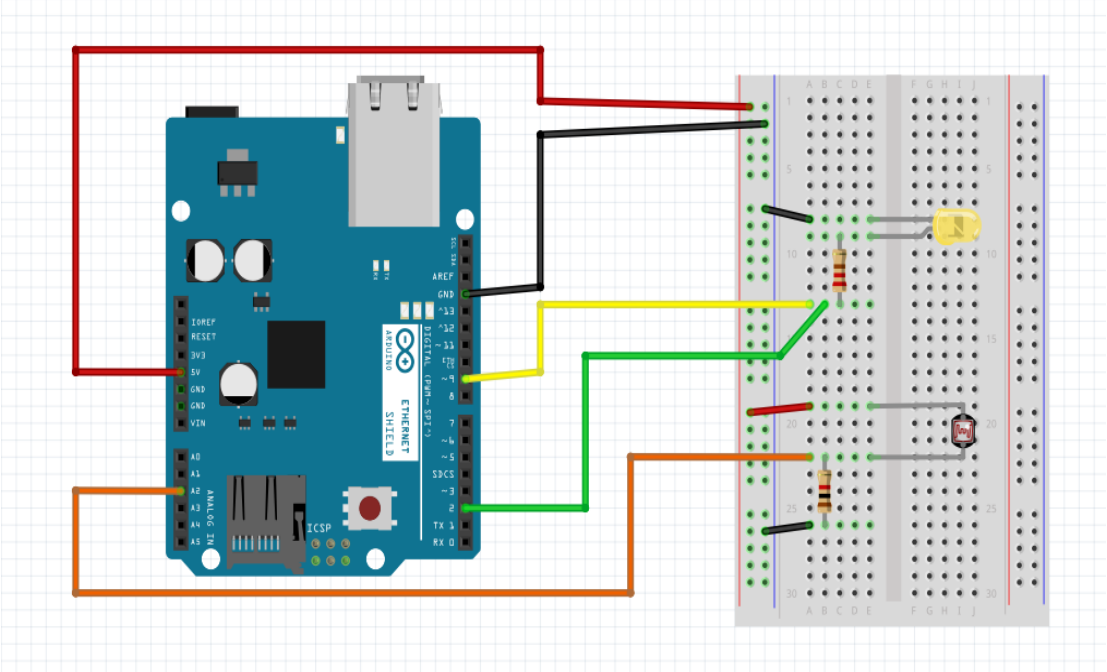
1. Ένα Arduino Uno.
2. Μια Ethernet Shield.
3. Μια Breadboard.
4. Τέσσερα Led (άσπρο, κίτρινο, γαλάζιο, και κόκκινο).
5. Μια φωτοδίοδο.
6. Έναν αισθητήρα θερμοκρασίας.
7. Ένα ανεμιστήρα (5V).
8. Δυο διακόπτες-κουμπιά.
9. Πέντε αντιστάσεις(4 των 220Ω και 1 των 1ΚΩ).
10. Καλώδια.

Στην αρχή θα πάμε να συνδέσουμε τον αισθητήρα θερμοκρασίας. Με την συνδεσμολογία αυτή θα καταφέρουμε να παίρνουμε την ένδειξη της θερμοκρασίας. Στο αριστερό ποδαράκι του αισθητήρα θα βάλουμε τα 5V, στο δεξιά ποδαράκι θα το βάλουμε την γείωση και το μεσαίο ποδαράκι θα το βάλουμε στο Pin A3. Το μεσαίο είναι αυτό που θα μας δίνει την ένδειξη της θερμοκρασίας. Στην παρακάτω εικόνα 23 φαίνεται η συνδεσμολογία που αναφέραμε.



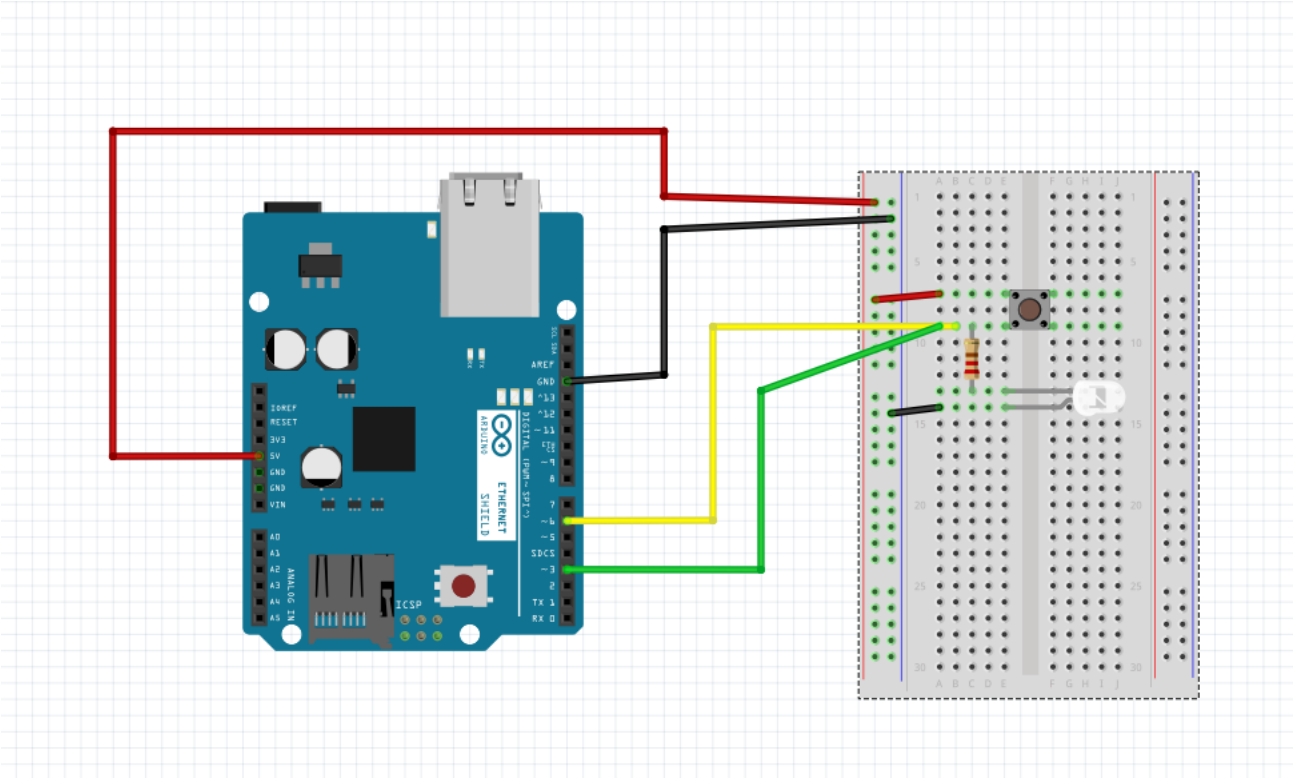
Εικόνα 23: Συνδεσμολογία αισθητήρα θερμοκρασίας

Επόμενο βήμα είναι να συνδέσουμε την φωτοδίοδο και το κίτρινο Led. Με αυτή την συνδεσμολογία, θα καταφέρουμε όταν δείξει η φωτοδίοδος κάτω από κάποια Lux που θα ορίσουμε αργότερα στον κώδικα μας να ανάβει το κίτρινο Led. Επίσης θα βλέπουμε και την κατάσταση του Led, αν είναι ON ή OFF. Γειώνουμε το ένα ποδαράκι του Led, στο άλλο ποδαράκι τοποθετούμε μια αντίσταση 220Ω και από την αντίσταση φεύγουν δυο καλώδια, τα οποία το ένα πάει στο Pin 9 και το άλλο στο Pin 2. Στην φωτοδίοδο το ένα ποδαράκι το συνδέουμε στην φάση, από το άλλο φεύγουμε πάμε σε μια αντίσταση 1ΚΩ και μετά στην γείωση. Από το δεύτερο ποδαράκι της φωτοδιόδου περνούμε ένα καλώδιο και το συνδέουμε στο Pin A2. Στην παρακάτω εικόνα 24 φαίνεται η συνδεσμολογία που αναφέραμε πιο πάνω.



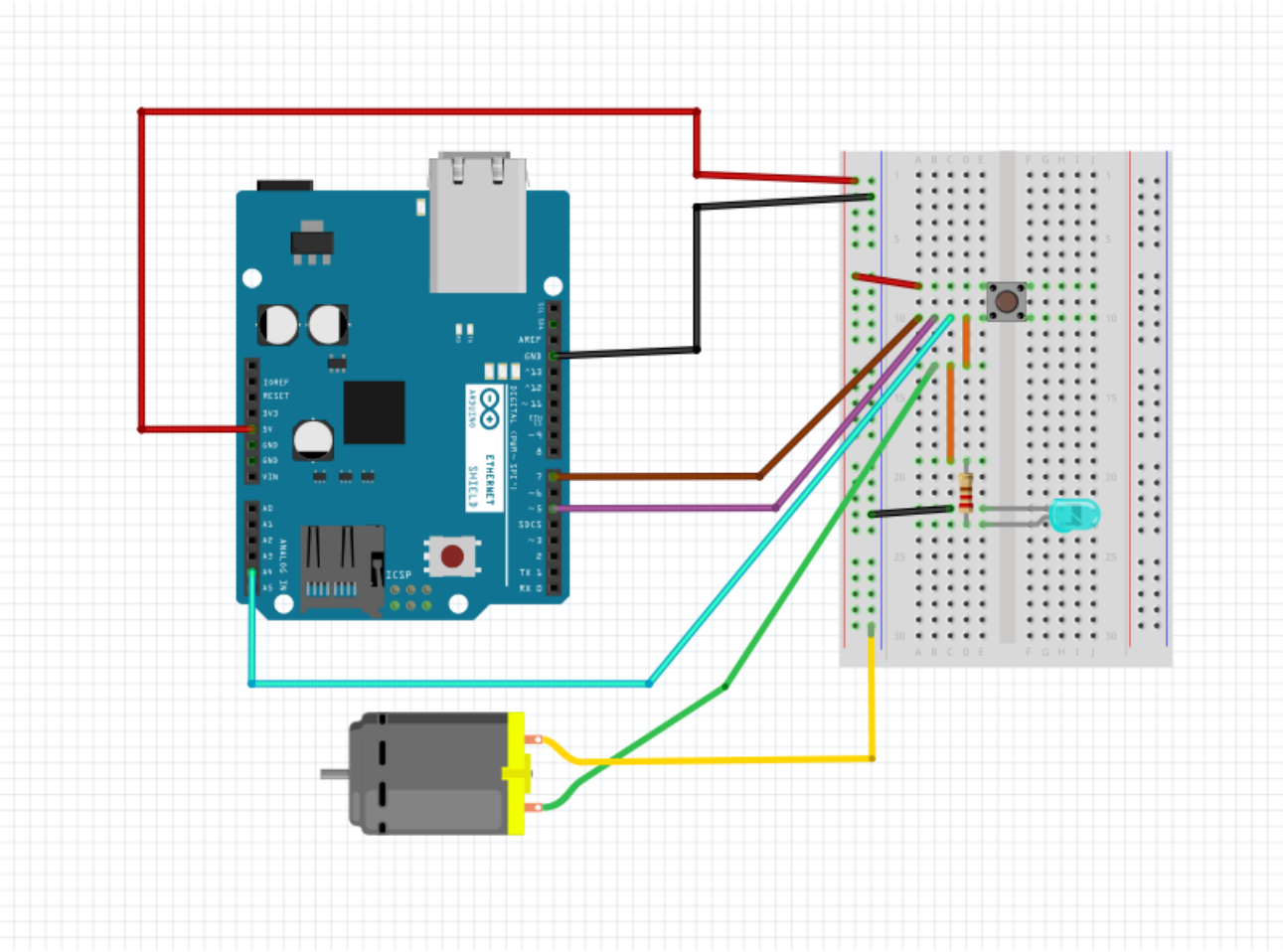
Εικόνα 24: Συνδεσμολογία φωτοδιόδου με κίτρινο Led

Επόμενη κίνηση είναι να τοποθετήσουμε το κουμπί με το άσπρο Led. Στην συνδεσμολογία αυτή θα καταφέρουμε να έχουμε χειροκίνητο και εξ'αποστάσεως έλεγχο του Led. Επίσης θα βλέπουμε την κατάσταση του Led αν είναι ON ή OFF. Στο πρώτο άκρο του κουμπιού δίνουμε τάση 5V, από το άλλο άκρο του κουμπιού φεύγουν 2 καλώδια, το οποίο το ένα πάει στο Pin 3 και το άλλο στο Pin 6. Επίσης στο δεύτερο άκρο του κουμπιού τοποθετούμε μια αντίσταση 220Ω, η οποία καταλήγει σε ένα ποδαράκι του άσπρου Led και το άλλο ποδαράκι το γειώνουμε. Στην ακόλουθη εικόνα 25 φαίνεται η συνδεσμολογία που αναφέραμε.



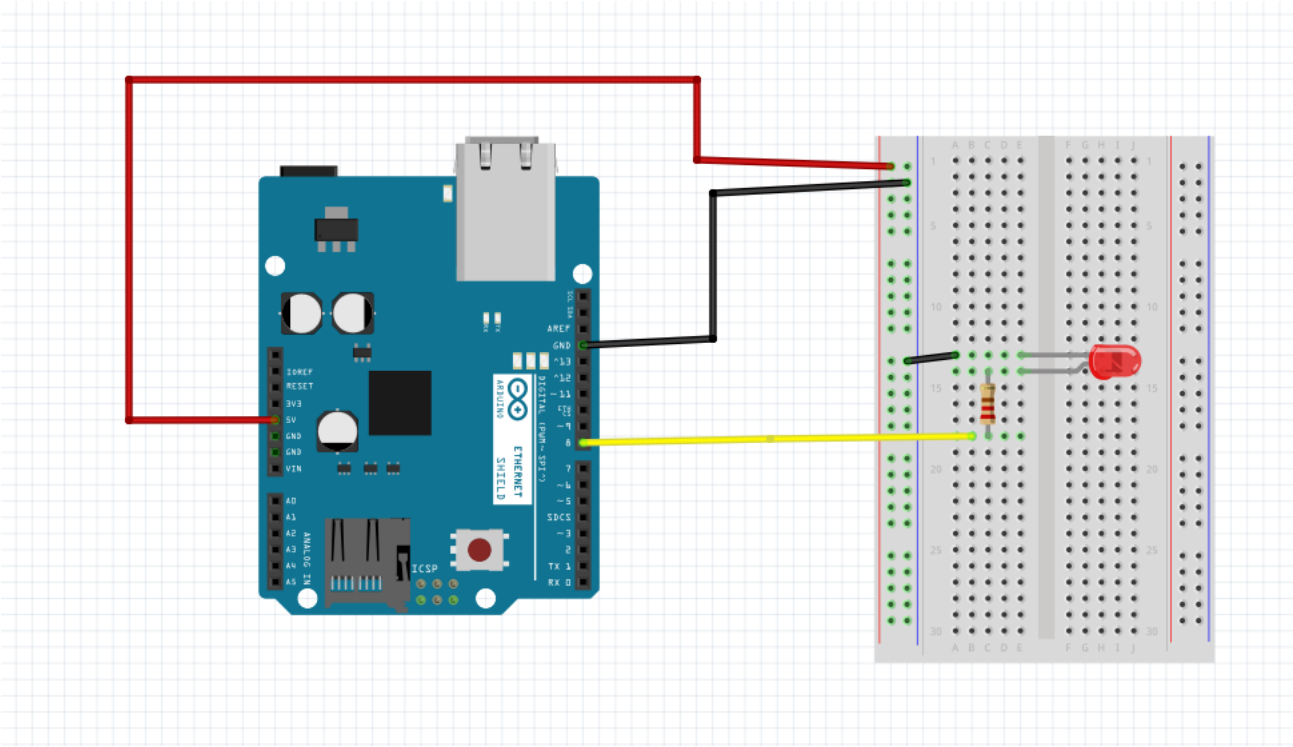
Εικόνα 25: Συνδεσμολογία άσπρου Led με κουμπί

Επόμενη συνδεσμολογία είναι να τοποθετήσουμε ένα κουμπί, ένα γαλάζιο Led και ένα ανεμιστήρα. Στην συνδεσμολογία αυτή θα καταφέρουμε να έχουμε τον χειροκίνητο και εξ'αποστάσεως έλεγχο, του ανεμιστήρα και του Led. Επίσης θα διαβάζει την κατάσταση και των δυο στοιχείων και θα την εμφανίζει στην εφαρμογή android που θα φτιάξουμε αργότερα. Στο πρώτο άκρο του κουμπιού δίνουμε 5V. Από το άλλο άκρο περνούμε 3 καλώδια, το ένα το πάμε στο Pin A4, το άλλο στο Pin 7 και το άλλο στο Pin 5. Επίσης από το δεύτερο άκρο του κουμπιού κάνουμε μια διακλάδωση και παίρνουμε ένα καλώδιο που το τοποθετούμε σε μια αντίσταση 220Ω. Από την αντίσταση φεύγουμε και πάμε στο ποδαράκι του Led και από το άλλο ποδαράκι πάμε στην γείωση. Από την διακλάδωση που κάναμε πριν παίρνουμε και ένα άλλο καλώδιο, το οποίο το πάμε στο θετικό άκρο του ανεμιστήρα και το άλλο άκρο του το βάζουμε στην γείωση. Στην ακόλουθη εικόνα 26 φαίνεται η συνδεσμολογία που αναφέραμε.



Εικόνα 26: Συνδεσμολογία γαλάζιου led με κουμπί και ανεμιστήρα

Τέλος, θα βάλουμε και ένα κόκκινο Led το οποίο θα έχουμε την ικανότητα να το ελέγχουμε μόνο από απόσταση. Θα κάνουμε την απλή συνδεσμολογία ενός Led, δηλαδή το αρνητικό ποδαράκι στην γείωση, το θετικό ποδαράκι στην αντίσταση και από την αντίσταση στο Pin 8. Ακολουθεί στην εικόνα 27 η συνδεσμολογία του κόκκινου Led.



Εικόνα 27: Συνδεσμολογία κόκκινου Led

3.1.8 Κώδικας arduino.

Όταν τελειώσουμε με την συνδεσμολογία είμαστε έτοιμοι να πάμε να γράψουμε τον κώδικα στο πρόγραμμα, το οποίο θα περάσουμε στο Arduino Uno. Θα δούμε μερικά βασικά μέρη του κώδικα, για να κατανοήσουμε την λειτουργία του. Αρχικά πρέπει να καλέσουμε της βιβλιοθήκες που θα χρησιμοποιήσουμε, οι οποίες είναι:

```
#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>
```

Με το #include και δίπλα την αντίστοιχη βιβλιοθήκη, την καλούμε στο πρόγραμμα μας. Η SPI.h (Serial Peripheral Interface) είναι ένα σύγχρονο σειριακό πρωτόκολλο δεδομένων, που χρησιμοποιούνται από μικροελεγκτές για την επικοινωνία με μια ή περισσότερες περιφερικές συσκευές. Στην δικιά μας περίπτωση συνδέει το Arduino Uno με την Ethernet Shield. Η Ethernet.h βιβλιοθήκη είναι για την σύνδεση με το Internet και η SD.h είναι για να αναγνωρίζει και να διαβάζει την SD-Card που έχουμε τοποθετήσει στην Ethernet Shield.

```
byte mac[] = { 0x90, 0xA2, 0xDA, 0x0D, 0xC3, 0x61 };
IPAddress ip(192, 168, 2, 10);
EthernetServer server(8080);
```

Η byte mac (Media Access Control) είναι ένας μοναδικός σειριακός αριθμός για κάθε δικτυακή συσκευή, η οποία ελέγχει την πρόσβαση στο διαδίκτυο. Για την δικιά μας περίπτωση ο αριθμός είναι αυτός 0x90, 0xA2, 0xDA, 0x0D, 0xC3, 0x61 που τον βρήκαμε πάνω στην Ethernet Shield.

Η IPAddress (Internet Protocol Address) είναι ένας μοναδικός αριθμός, που χρησιμοποιείται από συσκευές για τη μεταξύ τους αναγνώριση και συνεννόηση σε ένα δίκτυο . Δεν είναι σε όλους η ίδια, αλλάζει ανάλογα με το δίκτυο.

Η EthernetServer server(8080) δημιουργεί ένα Web Server στην πόρτα 8080 για την μεταφορά δεδομένων(στην δικιά μας περίπτωση το HTTP αρχείο που θα είναι μέσα στην SD-Card) ανάμεσα σε ένα διακοσμητή και έναν χρήστη.

```
void setup()

//Είσοδοι
   pinMode(2, INPUT);
   pinMode(3, INPUT);
   pinMode(5, INPUT);
//Εξοδοι
   pinMode(6, OUTPUT);
   pinMode(7, OUTPUT);
   pinMode(8, OUTPUT);
```

Στις πρώτες 3 εντολές αρχικοποιούμε ότι τα Pin 2,3,5 θα είναι είσοδοι, αυτά τα Pin θα μας διαβάζουν την κατάσταση των Led και για αυτό τα χρησιμοποιούμε σαν εισόδους.

Στις άλλες 3 εντολές αρχικοποιούμε ότι τα Pin 6,7,8 θα είναι έξοδοι, σε αυτά τα Pin είναι συνδεδεμένα τα Led τα οποία θα τα ανάβουμε και θα τα σβήνουμε.

```
void loop()

// LED1  (pin 6)
   if (StrContains(HTTP_req, "LED1=1")) {
```


```
    LED_state[0] = 1;  // save LED state
    digitalWrite(6, HIGH);


  }
  else if (StrContains(HTTP_req, "LED1=0")) {
    LED_state[0] = 0;  // save LED state
    digitalWrite(6, LOW);
```

Τώρα έχουμε μπει στο μέρος του προγράμματος void loop (επαναλαμβανόμενη διαδικασία), θα δούμε το pin 6 που έχουμε αρχικοποίηση σαν έξοδο πως λειτουργεί. Ίδια διαδικασία είναι και για τα υπόλοιπα pin 7 - 8. Για αρχή διαβάζει από το HTTP αρχείο που είναι τοποθετημένο στην SD-Card, εάν το led1=1 δηλαδή εάν έχουμε πατήσει να ανάψει το Led από το Web περιβάλλον που θα δούμε αργότερα. Εάν είναι 1 τότε κάνει αποθήκευση την κατάσταση του Led και έπειτα του λέμε να κάνει την έξοδο Pin 6 High(ενεργοποίηση).

Εάν δεν ισχύει τίποτα από τα παραπάνω ξαναδιαβάζει το HTTP αρχείο και αν είναι το led1=0 δηλαδή δεν έχουμε πατήσει το κουμπί τότε κάνει αποθήκευση την κατάσταση του Led, ότι είναι 0 και του λέμε να κάνει την έξοδο Pin 6 Low(απενεργοποίηση).

```
for (count = 2; count <= 2; count++) {
      analog_val = analogRead(count);
      cl.print("<analog>");
      cl.print(analog_val);
      cl.println("</analog>");
    if (analog_val > 400){
  digitalWrite(ledpin, LOW); // Led σβηστό
 }
 else{
  digitalWrite(ledpin, HIGH); // Led αναμένο
```

Στον παραπάνω κώδικα βλέπουμε πως λειτουργεί η φωτοδίοδος με το Led. Στο αναλογικό Pin A2 έχουμε συνδέσει τη φωτοδίοδο, η οποία διαβάζει τα Lux που έχει εκείνη την στιγμή το σπίτι. Έπειτα μας τα δείχνει στο Web περιβάλλον. Επίσης έχουμε προγραμματίσει, όταν τα Lux του σπιτιού μας πέσουν κάτω από 400 τότε να ανάψει το Led αλλιώς να παραμένει κλειστό.

```
for (count = 3; count <= 3; count++) {
      analog_val = analogRead(count);
```


```
        cl.print("<analog>");
        cl.print(analog_val * 0.15 );
        cl.println("</analog>");
      }
```

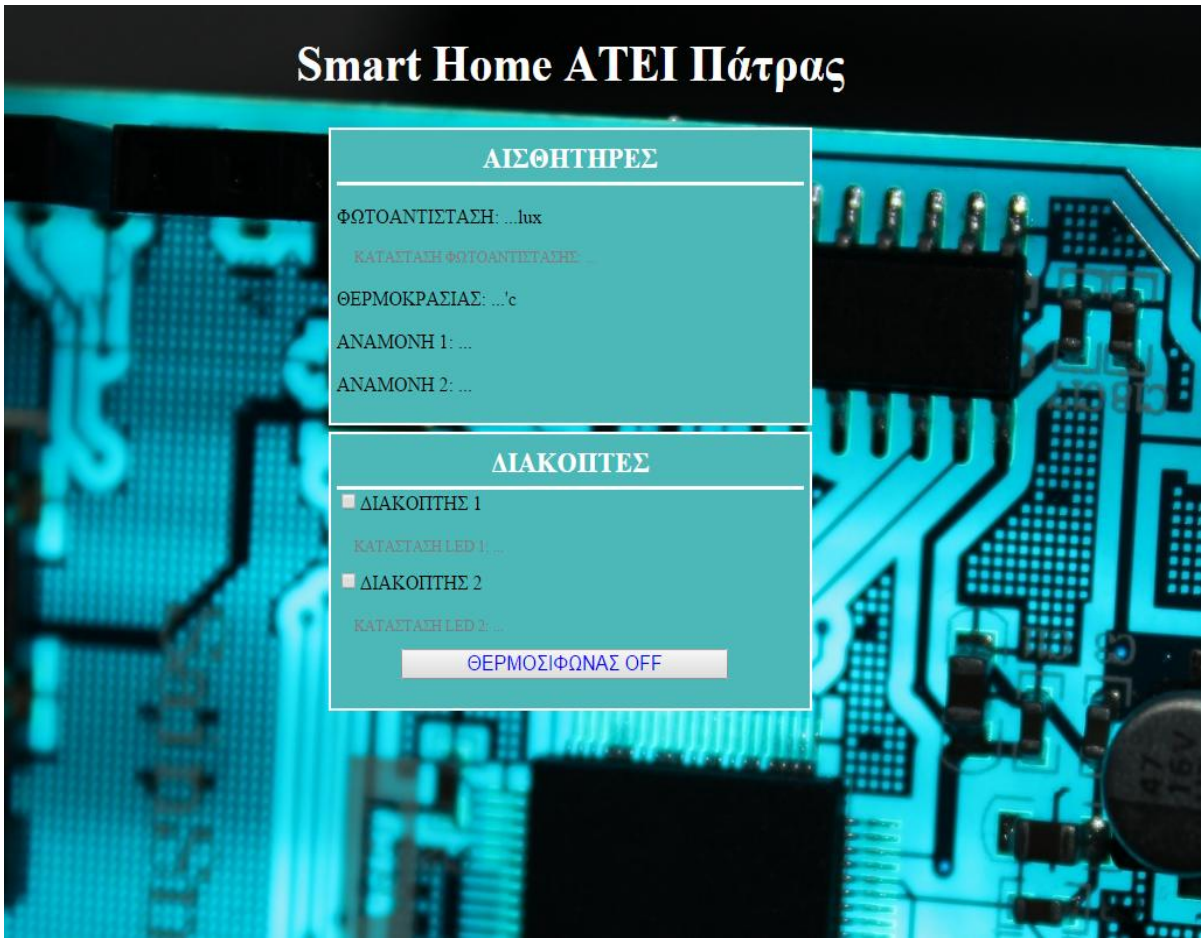
Εδώ βλέπουμε τον κώδικα της θερμοκρασίας. Στο αναλογικό Pin A3 έχουμε συνδέσει τον αισθητήρα, το οποίο διαβάζει την θερμοκρασία. Την μετρούμενη τιμή του αισθητήρα, την πολλαπλασιάζουμε με το 0.15 για να δείχνει σωστά την θερμοκρασία και να μας την τυπώνει στο Web περιβάλλον.

```
for (count = 0; count < 3; count++) {
        cl.print("<switch>");
        if (digitalRead(sw_arr[count])) {
           cl.print("ON");
        }
        else {
           cl.print("OFF");
        }
        cl.println("</switch>");
    }
```

Τέλος βλέπουμε τον κώδικα που μας εμφανίζει τις καταστάσεις που θα έχουν τα Led και ο ανεμιστήρας. Στον κώδικα διαβάζονται και οι 3 καταστάσεις ταυτόχρονα. Για παράδειγμα, εάν το Led είναι ανοιχτό τότε τυπώνει στο Web περιβάλλον ON Αλλιώς εάν το Led είναι κλειστό τότε τυπώνει στο Web περιβάλλον OFF.

3.1.9 Αρχείο HTML SD-Card.

Το HTML αρχείο που βρίσκεται στην SD-Card δημιουργεί το περιβάλλον που θα βλέπουμε στο Web Browser αλλά και στην εφαρμογή Android. Αυτό το περιβάλλον θα αποτελείται από κουμπιά, εικόνες, ενδείξεις, λέξεις και αλλά τα οποία θα επικοινωνούν με το Arduino. Στην ακόλουθη εικόνα 28 βλέπουμε πως θα είναι το περιβάλλον που θα χειριζόμαστε της συσκευές μας.



Εικόνα 28: Περιβάλλον HTML αρχείου.

Στο πρώτο κουτάκι βλέπουμε τους αισθητήρες που θα χρησιμοποιήσουμε. Στην φωτοαντίσταση θα βλέπουμε τα Lux του σπιτιού μας και ακριβώς από κάτω θα βλέπουμε την κατάσταση του Led, δηλαδή αν τα Lux πέσουν κάτω από 400 που έχουμε ορίσει στον κώδικα να ανάψει το Led και η κατάσταση θα γίνει ON. Εάν είναι πάνω από 400 θα γράφει OFF η κατάσταση της φωτοαντιστασης.

Ακριβώς από κάτω βλέπουμε την θερμοκρασία από τον αισθητήρα θερμοκρασίας που έχουμε τοποθετήσει στην συνδεσμολογία. Επίσης υπάρχουν και δυο αναμονές για να τοποθετήσουμε όποιον άλλο αισθητήρα θέλουμε στο κύκλωμα μας, για να παίρνουμε της ενδείξεις του και να εμφανίζονται εκεί.

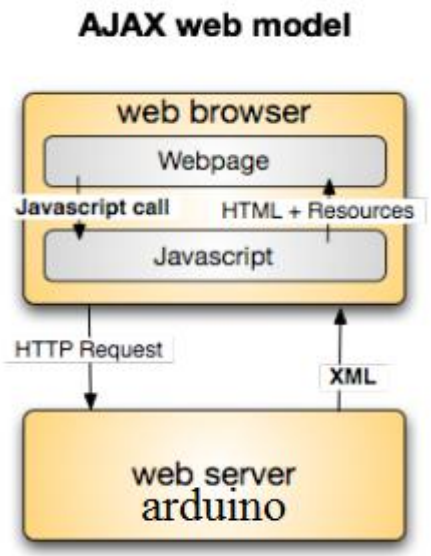
Στο δεύτερο κουτάκι βλέπουμε τους διακόπτες από τα Led. Στους δυο πρώτους διακόπτες έχουμε την ικανότητα να βλέπουμε και την κατάσταση του Led. Αν είναι κλειστός θα γράφει κατάσταση Led 1 OFF, ενώ αν είναι ανοιχτός θα γράφει κατάσταση Led 2 ON. Τέλος έχουμε και ένα τρίτο διακόπτη από τον οποίο θα ανοίγουμε και θα κλείνουμε τον θερμοσίφωνα (Led). Αφού είδαμε πως θα είναι το περιβάλλον και πως λειτουργεί, πάμε να δούμε τι είναι το HTML αρχείο και πως δημιουργήθηκε.

Η HTML (Hyper Text Markup Language) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχειά της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες (Tags), οι οποίες περικλείονται μέσα σε σύμβολα. Ανάμεσα στις ετικέτες μπορούμε να τοποθετήσουμε κείμενο πίνακες εικόνες κλπ. Ο σκοπός του Wed Browser είναι να διαβάζει τα έγραφα HTML και να τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει και να ακούσει. Ο Browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να αποκρυπτογραφήσει το περιεχόμενο της σελίδας. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων, καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παρενθέσεις και άλλα. Στον κώδικά μας έχουμε ενσωματώσει και σενάρια εντολών σε γλώσσα JavaScript τα οποία επηρεάζουν την συμπεριφορά τον ιστοσελίδων HTML

Η JavaScript είναι μια γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές. Η σύνταξή της είναι επηρεασμένη από την C. Η JavaScript αντιγράφει ονόματα και αρκετά άλλα από τη Java. Αρχικά αποτέλεσαι μέρος της υλοποίησης των Wed Browser, ώστε τα σενάρια από την πλευρά του πελάτη να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσουν δεδομένα ασύγχρονα και να αλλάζουν το περιεχόμενο του εγγράφου που εμφανίζεται. Άρα στην δικιά μας περίπτωση θα επικοινωνεί ο Wed Browser που περιέχει μέσα το JavaScript με το Arduino.

Επίσης έχουμε χρησιμοποιήσει και μια γλώσσα, την CSS. Την χρησιμοποιούμε για έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML δηλαδή για τον έλεγχο της εμφάνισης της ιστοσελίδας μας και γενικότερα του ιστότοπού μας. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει το στυλ σε μια ιστοσελίδα, να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την HTML.

Για την επικοινωνία μεταξύ όλων αυτόν που αναφέραμε πιο πάνω αλλά και του Arduino με την Ethernet Shield, θα χρησιμοποιήσουμε το AJAX (Asynchronous JavaScript And XML) Web Model. Το πως συνδέει όλες τις τεχνολογίες μεταξύ τους φαίνεται στην εικόνα 29.



Εικόνα 29: AJAX Web Model.

Ας δούμε μια απλή ανάλυση της εικόνας 29.

1.Μεσα από το Interface της ιστοσελίδας μας (Webpage) κάνουμε κάποιο Request, στο δικό μας παράδειγμα θα πατήσουμε τον διακόπτη για να ανοίξει το Led.

2.Με το που πατηθεί ο διακόπτης δημιουργείται από το JavaScript το HTTP Request. Στο HTTP συνδεόμαστε με κάποιον Browser για να δούμε κάποιο έγγραφο σε ένα Server. Αφού δέχεται τα δεδομένα από το χρήστη επικοινωνεί με τον Server του Arduino που βρίσκεται στη βάση δεδομένων.

3.Αν είναι αληθής η διαδικασία τότε το Led ανάβει

4.Ο Web Server του Arduino επιστέφει τα δεδομένα στο JavaScript

5.Οταν τα δεδομένα επιστρέψουν στο Webpage από το HTML, τότε αλλάζει και η κατάσταση στην σελίδα. Στην δικιά μας περίπτωση η κατάσταση του Led θα γίνει ON και ο διακόπτης θα έχει ένα τικ.

3.1.10 Κώδικας αρχείου HTML στην SD-Card.

Εφόσον εξηγήσαμε μερικές βασικές έννοιες, πάμε να δούμε κάποια μέρη του κώδικα. Στις παρακάτω εντολές βλέπουμε την δομή που έχει το HTML.

```
<!DOCTYPE html>
<html>

          <head>
           <title>Arduino ATEI Patras</title>
           <script>
           "κώδικας"
           </script>
           <style>
           "κώδικας"
           </style>
           </head>
           <body>
           "κώδικας"
           </body>

</html>
```


Στο πρώτο Head έχουμε γράψει ένα τίτλο (Arduino ATEI Patras), όπου αυτός ο τίτλος θα εμφανίζετε στην καρτέλα του Web Browser που θα χρησιμοποιήσουμε. Ανάμεσα στα Script θα μπει ο κώδικας, ο οποίος θα επικοινωνεί με τον κώδικα του Arduino. Έπειτα ανάμεσα στο Style θα δημιουργήσουμε τα κουτιά, τα χρώματα, τα μεγέθη, τις εικόνες και άλλα. Τέλος ανάμεσα στο Body θα βάλουμε το κείμενο, το είδος των διακοπτών, και άλλα. Πιο κάτω θα δούμε ένα παράδειγμα για το κάθε ένα που αναφέραμε πιο πάνω, το οποίο χρησιμοποιούμαι στο HTML.

Ανάμεσα στα Script:

```
   if
(this.responseXML.getElementsByTagName('LED')[2].childNodes[0].node
Value === "on") {
   document.getElementById("LED3").innerHTML = "ΘΕΡΜΟΣΙΦΩΝΑΣ ON";
   LED3_state = 1;
}
  else {
   document.getElementById("LED3").innerHTML = "ΘΕΡΜΟΣΙΦΩΝΑΣ OFF";
   LED3_state = 0;
```

Στο παραπάνω Script βλέπουμε πως λειτουργεί το κουμπί του θερμοσίφωνα. Όταν πατήσουμε το κουμπί ο θερμοσίφωνας θα γίνει ON τότε αποθηκεύει την κατάσταση (led_state=1) και επικοινωνεί με το πρόγραμμα του Arduino κάνοντας την έξοδο High (ενεργοποιεί το Led). Την εντολή led_state=1 την έχουμε βάλει αντίστοιχα και στο κώδικα του Arduino, για να καταλάβει την κατάσταση. Το ίδιο γίνεται και όταν γίνει OFF.

Ανάμεσα στα Style:

```
.button_style
{
        float: center;
        text-align: center;
        text-color: black;
        margin: 0px 10px 0px 55px;
        border-bottom-style: black;
        color: blue;
        }
```

Εδώ δημιουργούμε το κουμπί του θερμοσίφωνα σε ποιο σημείο θα βρίσκεται στην σελίδα, εμείς το έχουμε βάλει στο κέντρο. Έχουμε ορίσει ακόμα τις διαστάσεις του και τα χρώματα του.

Ανάμεσα στα Body:

```
<h1>Smart Home ATEI Πάτρας</h1>

<button class="button_style" type="button" id="LED3" onclick="GetButton1()">ΘΕΡΜΟΣΙΦΩΝΑΣ OFF</button><br /><br />
```

Τέλος βλέπουμε το κείμενο που έχουμε βάλει στην σελίδα (Smart Home ATEI Πάτρας) και ακριβώς από κάτω έχουμε το είδος του κουμπιού του θερμοσίφωνα και την ονομασία την οποία θα εμφανίζει.

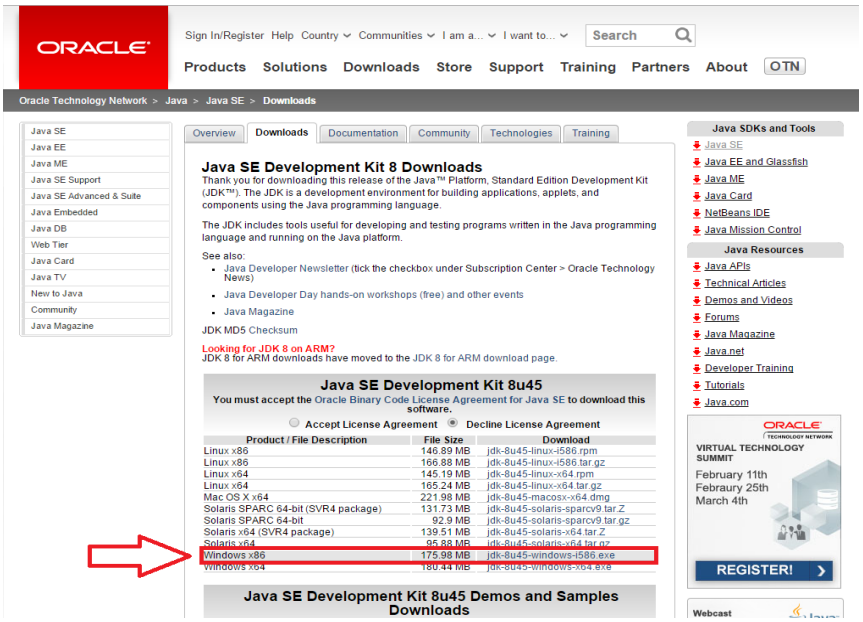
Εφόσον έχουμε περάσει το αρχείο HTML στην SD-Card και την έχουμε τοποθετήσει στην Ethernet Shield, είμαστε έτοιμοι να περάσουμε τον κώδικα στον μικροελεγκτή, από το περιβάλλον του Arduino. Αφού το περάσουμε και όλα πλέον επικοινωνούν σωστά μεταξύ τους, πάμε σε ένα Web Browser πατάμε την τοπική IP που έχουμε ορίσει στο πρόγραμμα και την πόρτα που έχουμε ανοίξει (π.χ. 192.168.2.10/8080). Μόλις φορτώσει η σελίδα είμαστε σε θέση να διαχειριστούμε το Smart Home που φτιάξαμε. Μπορούμε επίσης αντί για Led να χρησιμοποιήσουμε και κανονικούς λαμπτήρες ή ακόμα και να συνδέσουμε την καφετιέρα μας, την τηλεόραση και ότι άλλο επιθυμούμε. Οι ενώσεις αυτές δεν θα γίνουν απευθείας αλλά με κατάλληλη χρήση ρελέ.

3.1.11 Android εφαρμογή.

Ήρθε η στιγμή να δημιουργήσουμε και την Android εφαρμογή μας. Θα φτιάξουμε μια απλή εφαρμογή που θα έχουμε την ικανότητα να την εγκαταστήσουμε σε οποιαδήποτε Android συσκευή. Μόλις τρέξουμε την εφαρμογή θα έχουμε και από εκεί τον απόλυτο έλεγχο του Smart Home που έχουμε φτιάξει έως τώρα. Δηλαδή η εφαρμογή θα διαβάζει τα δεδομένα από τον HTTP Server και θα τα προβάλει. Για να είμαστε σε θέση να φτιάξουμε την εφαρμογή πρέπει να έχουμε εγκαταστήσει τα απαραίτητα προγράμματα στον υπολογιστή μας. Την εγκατάσταση των προγραμμάτων, θα την δούμε παρακάτω βήμα προς βήμα.

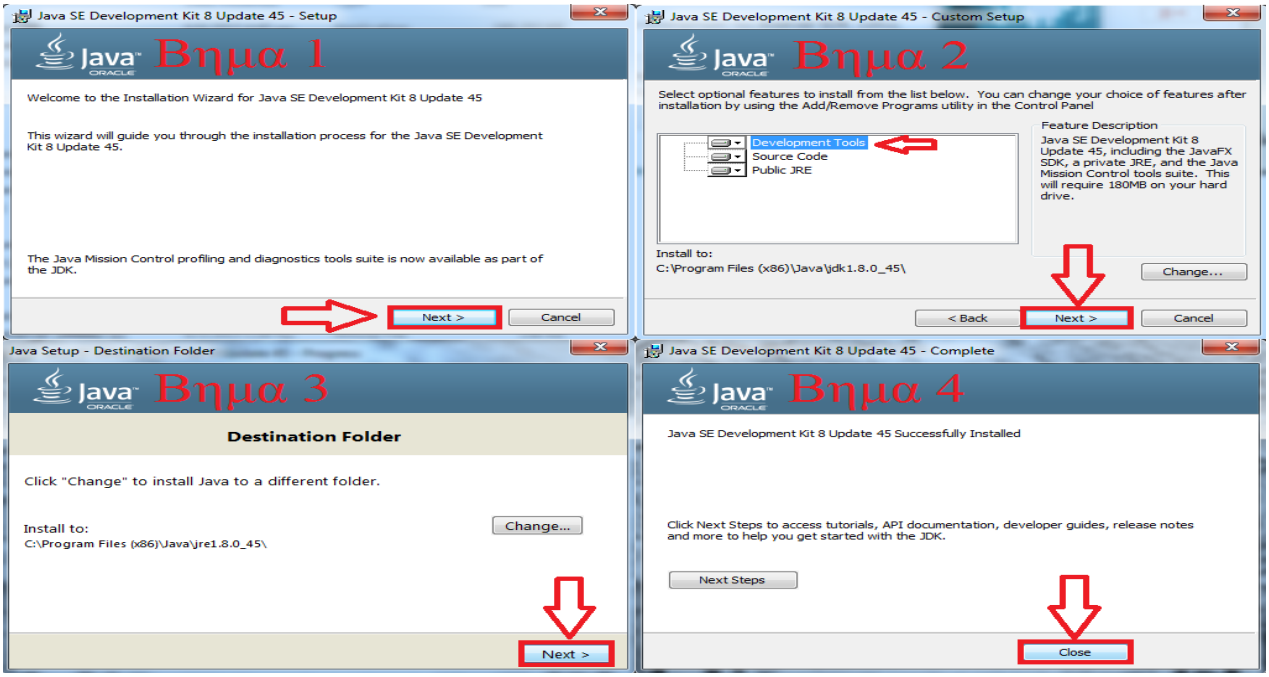
3.1.12 Εγκατάσταση Java Se Development Kit (JDK).

Για αρχή πρέπει να κατεβάσουμε το Java Se Development Kit από το ακόλουθο URL "http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html" και το αρχείο "jdk-8u45-windows-i586.exe". Το αρχείο αυτό είναι για Windows λειτουργικό και ανάλογα την έκδοση των windows(64 Bit ή 32 Bit) που έχουμε, πρέπει να κατεβάσουμε το αντίστοιχο αρχείο. Επίσης αναφέρουμε ότι αυτές οι εκδόσεις που χρησιμοποιούμε είναι οι πιο πρόσφατες και με τον καιρό μπορούν να αναβαθμιστούν. Στην ακόλουθη εικόνα 30 φαίνεται το αρχείο που θα κατεβάσουμε.



Εικόνα 30: JDK αρχείο.

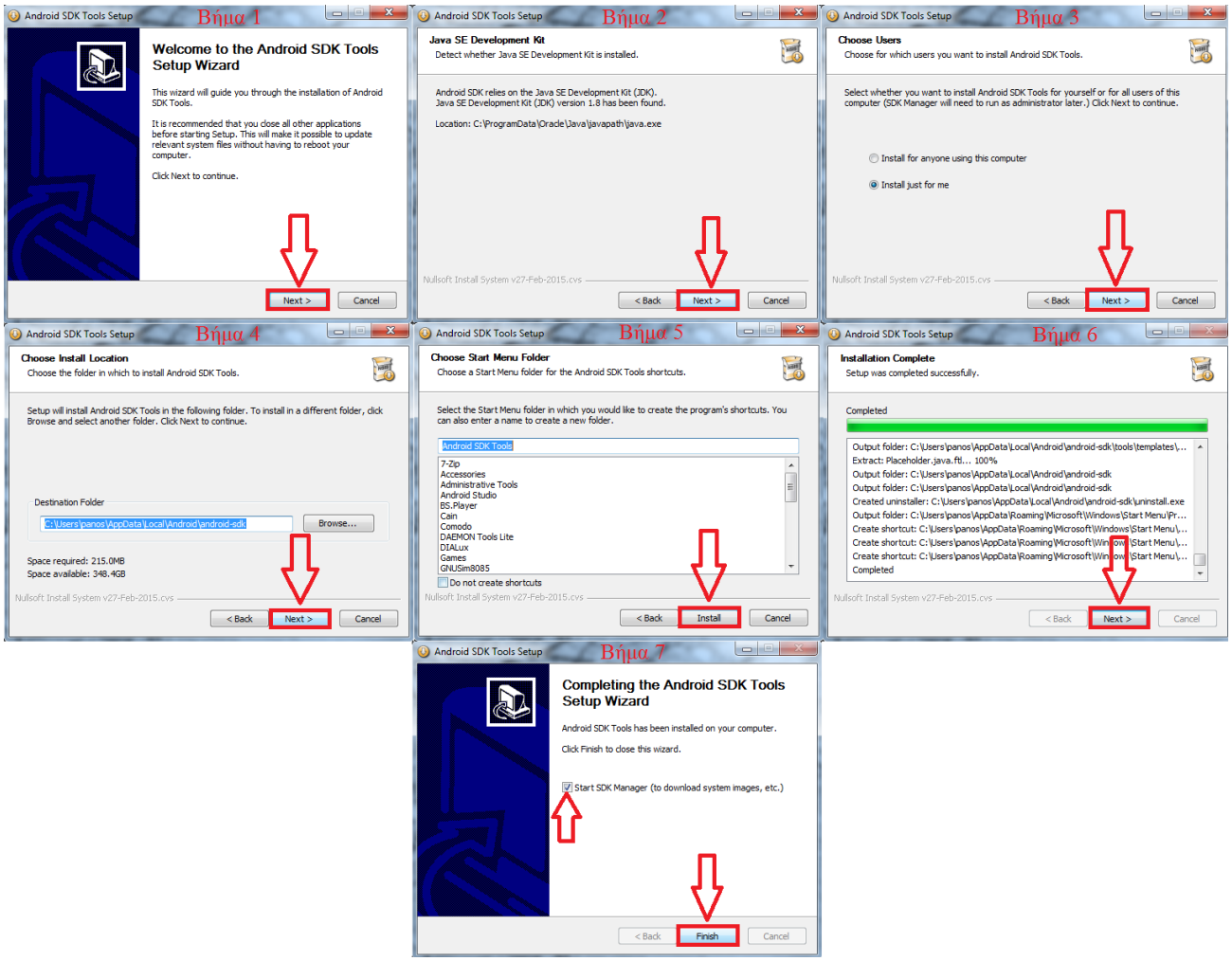
Αφού κατεβάσουμε το αρχείο, το ανοίγουμε και ακολουθούμε τα βήματα της εγκατάστασης που φαίνονται στην εικόνα 31 που ακολουθεί.



Εικόνα 31: Εγκατάσταση JDK.

3.1.13 Εγκατάσταση Android Software Developers Kit (SDK).

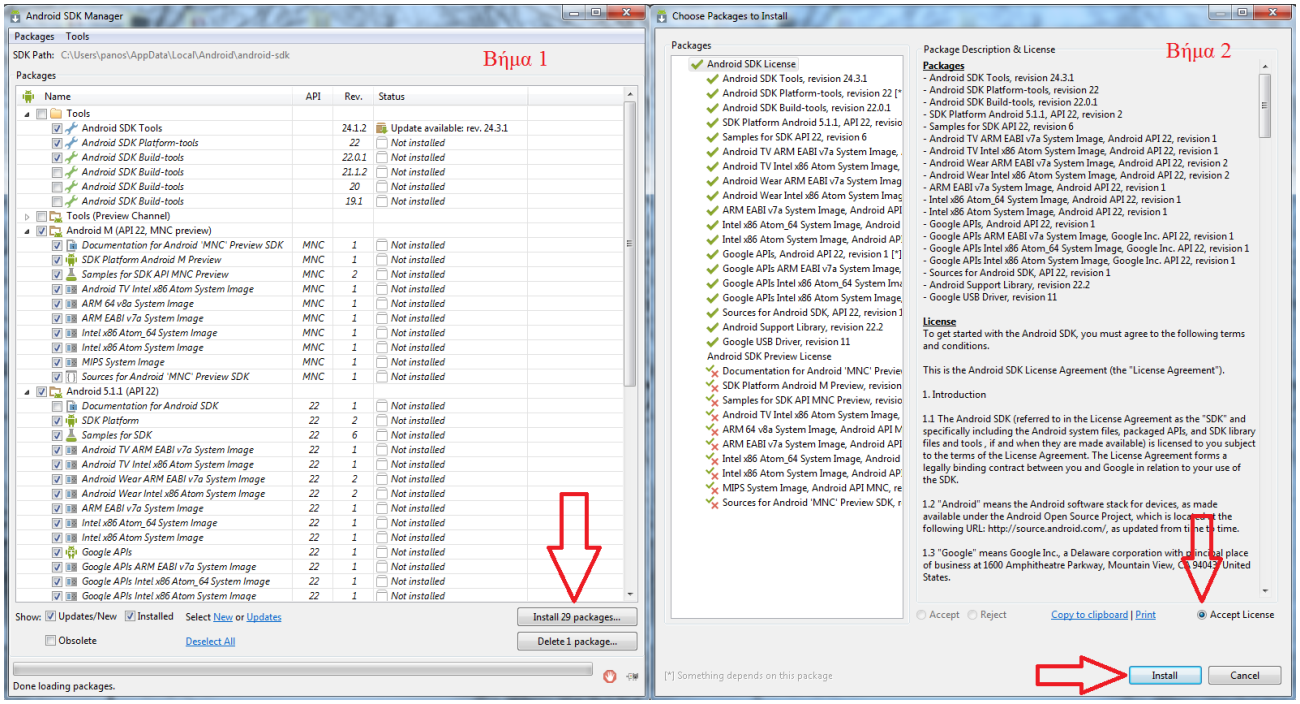
Επόμενη κίνηση είναι να κατεβάσουμε το Android SDK, από το ακόλουθο URL "https://developer.android.com/sdk/index.html".Όταν κατεβεί το αρχείο στον υπολογιστή μας, το ανοίγουμε και το τρέχουμε για να κάνουμε την εγκατάστασή του. Στο τέλος της εγκατάστασης θα μας ζητηθεί να τρέξουμε εκείνη την στιγμή το SDK Manager, εμείς το αποδεχόμαστε για να κατεβάσουμε τα απαραίτητα πακέτα που χρειάζονται. Στην εικόνα 32 που ακολουθεί φαίνεται η εγκατάστασή του.



Εικόνα 32: Εγκατάσταση Android SDK.

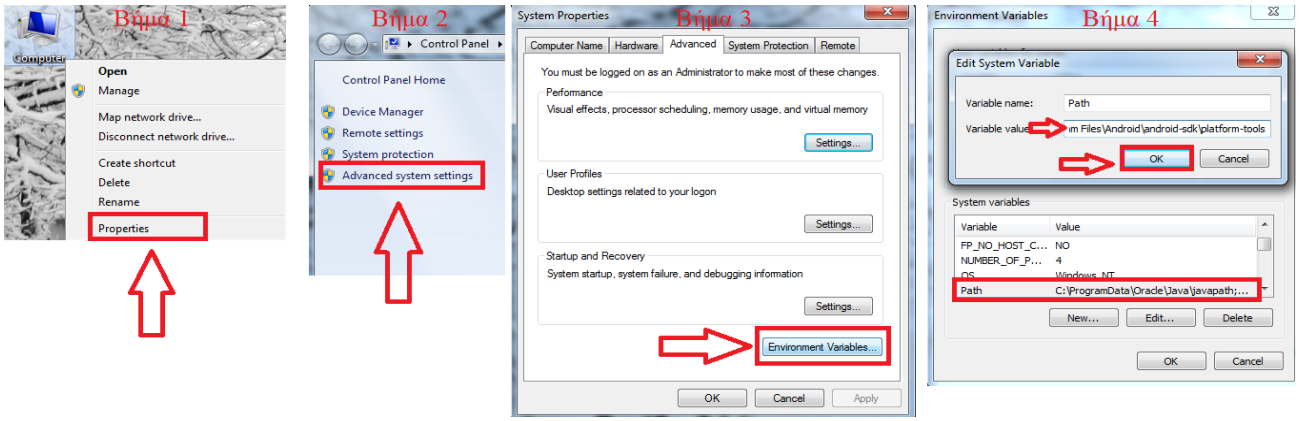
3.1.14 Εγκατάσταση Android SDK Manager.

Μόλις ανοίξει το SDK Manager θα πρέπει να επιλέξουμε κάποια πακέτα για να κατέβουν στον υπολογιστή μας. Στην εικόνας 33 που ακολουθεί βλέπουμε τις επιλογές των πακέτων που κάναμε. Εμείς κάναμε την πλήρη σχεδόν εγκατάσταση όλον των πακέτων, για την τρέχουσα έκδοση android 5.1.1 (API 22).Θα χρειαστεί αρκετός χρόνος μέχρι να ολοκληρωθεί η εγκατάσταση.



Εικόνα 33: Εγκατάσταση Android SDK Manager.

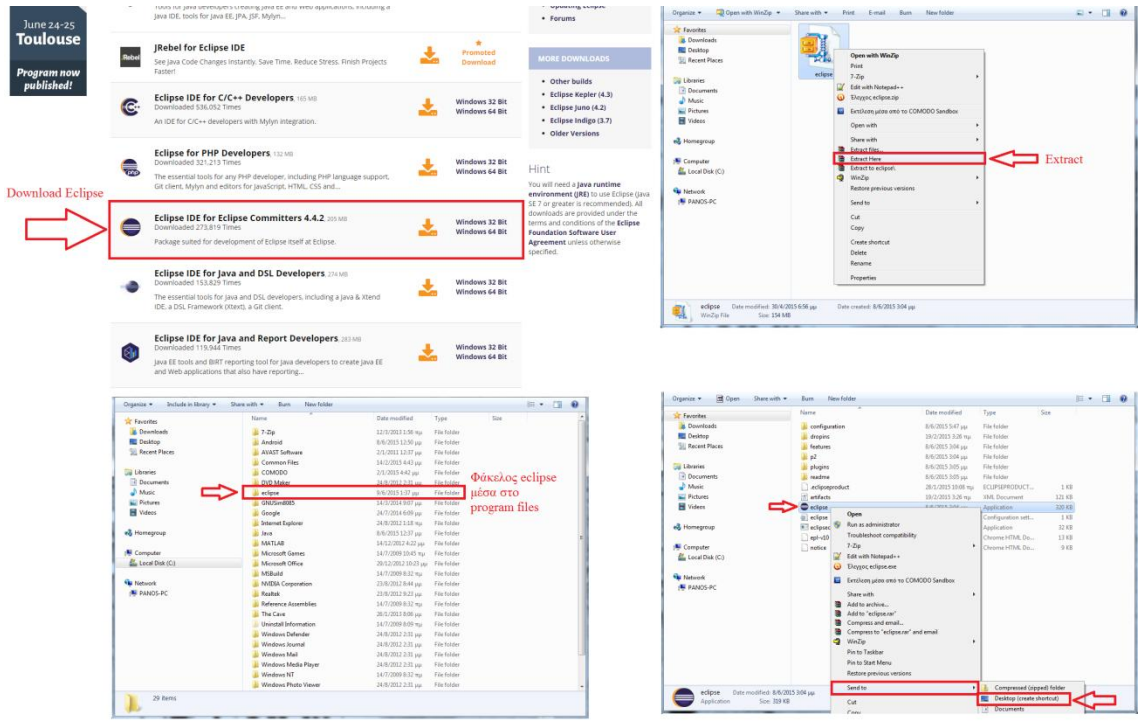
Όταν τελειώσει η εγκατάσταση, για να έχουμε πιο εύκολη πρόσβαση στα Command-Line εργαλεία, θα δημιουργήσουμε ένα Path στα Directories που οδηγούν στους φακέλους Tools και Platform-Tools. Για να το κάνουμε αυτό, πάμε στο Computer κάνουμε δεξί κλικ, Properties, Advanced System Setting, Environment Variables, βρίσκουμε το Path το επιλέγουμε και το κάνουμε Edit. Έπειτα στο Variable Value προσθέτουμε αυτό το Path(;C:\Program Files\Android\android-sdk\tools;C:\Program Files\Android\android-sdk\platform-tools). Η παραπάνω διαδικασία φαίνεται στην εικόνα 34.



Εικόνα 34: Δημιουργία Path.

3.1.15 Εγκατάσταση Eclipse.

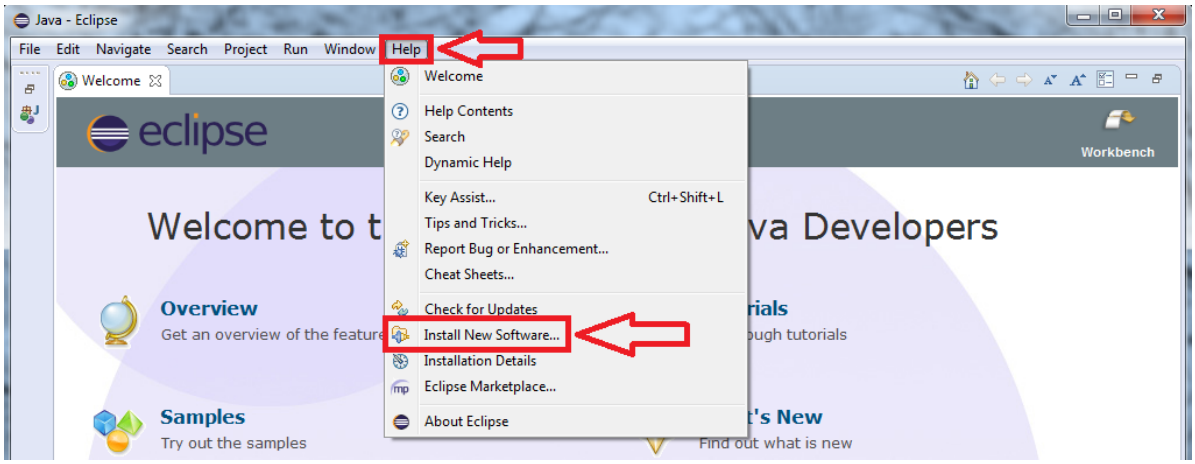
Κατεβάζουμε το Eclipse από το ακόλουθο URL "http://www.eclipse.org/downloads/". Όταν κατέβει, βρίσκουμε τον φάκελο του κάνουμε Extract , παίρνουμε όλο τον φάκελο του Eclipse και τον τοποθετούμε μέσα στο Program Files. Στην συνέχεια δημιουργούμε μια συντόμευση (Shortcut) πατώντας δεξί κλικ πάνω στο εικονίδιο του Eclipse(Application), Send To και επιλέγοντας Desktop.Το εικονίδιο βρίσκεται μέσα στον φάκελο που κατεβάσαμε. Όλες αυτές η ενέργειες φαίνονται στην ακόλουθη εικόνα 35.



Εικόνα 35: Eclipse.

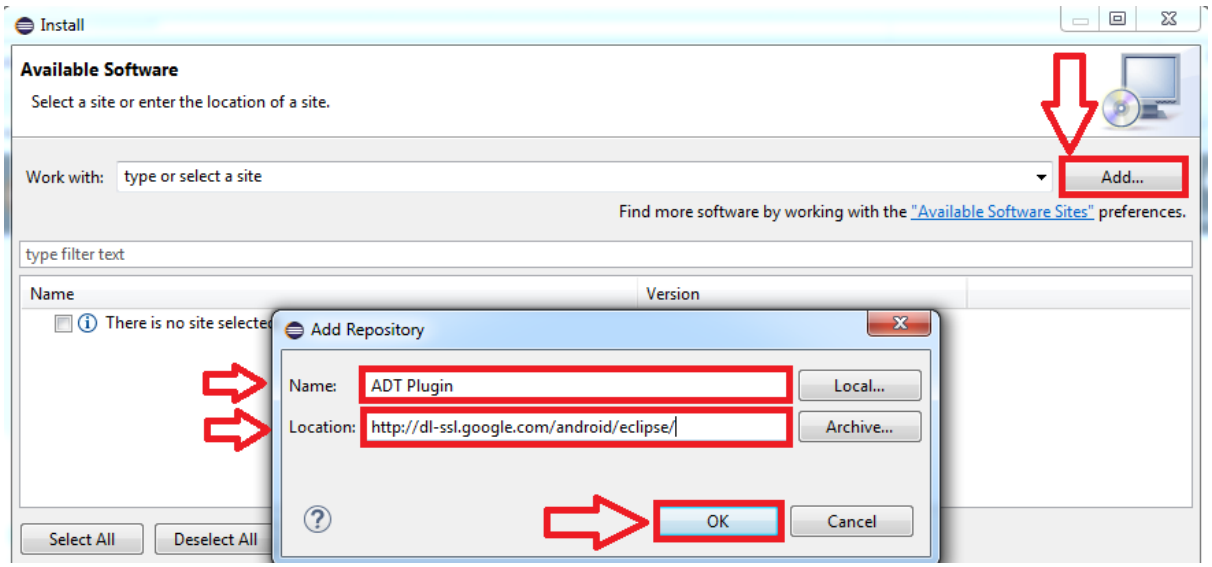
3.1.16 Εγκατάσταση ADT Plugin στο Eclipse.

Αφού τρέξουμε το Eclipse, θα μας εμφανίσει ένα παράθυρο (Workspace Launcher), στο οποίο πρέπει να επιλέξουμε μια διαδρομή που θα αποθηκεύονται οι εργασίες που θα κάνουμε. Θα επιλέξουμε μια εύκολη διαδρομή για να μπορούμε να βρίσκουμε τις εργασίες μας. Στην συνέχεια θα πάμε να εγκαταστήσουμε στο Eclipse το ADT Plugin. Στο περιβάλλον του Eclipse πάμε στο Help και επιλέγουμε Install New Software, εικόνα 36.



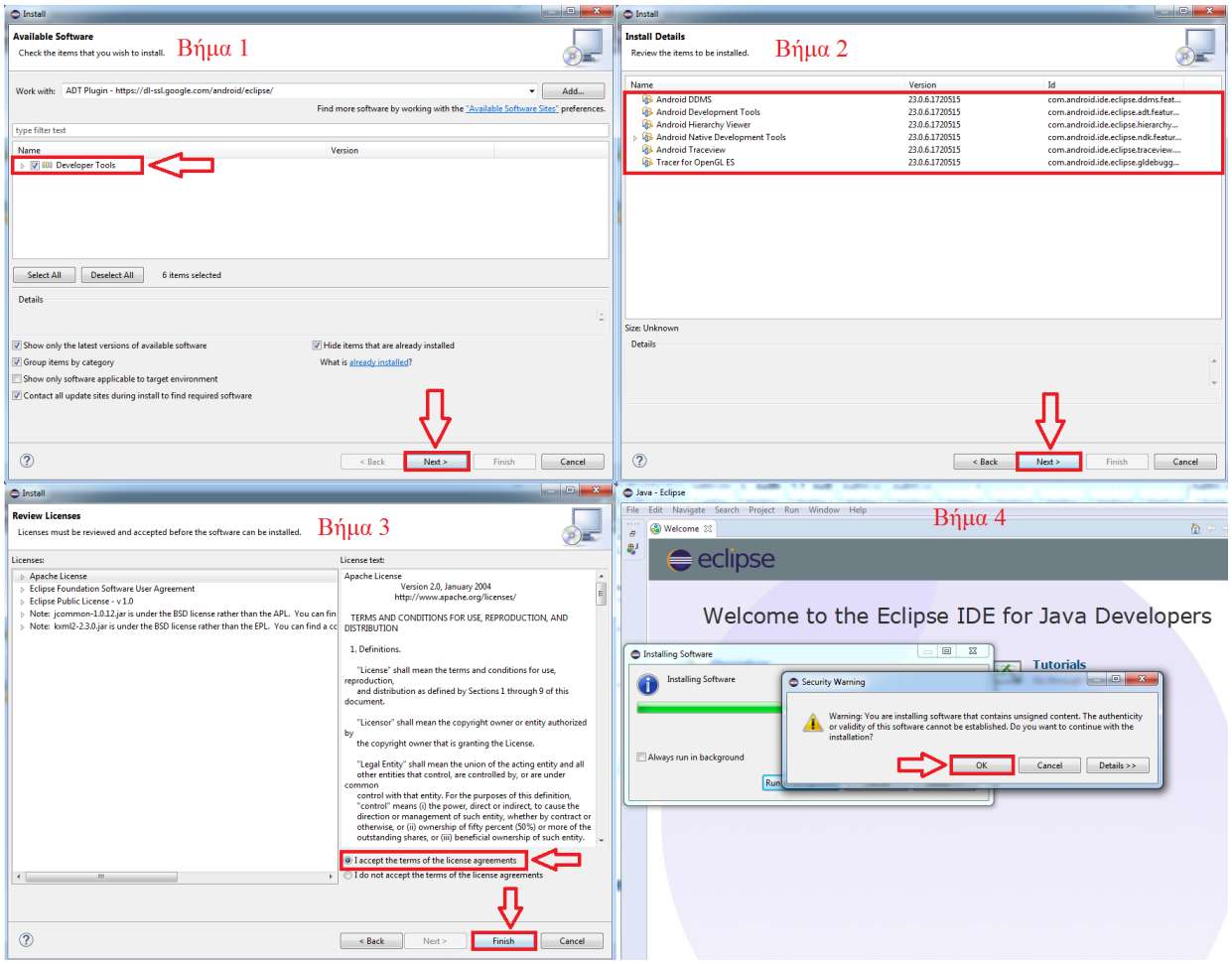
Εικόνα 36: Eclipse ADT Plugin.

Έπειτα θα μας εμφανίσει ένα παράθυρο, πάνω δεξιά πατάμε Add και συμπληρώνουμε για Name: "ADT Plugin" και για Location: "https://dl-ssl.google.com/android/eclipse/" μετά πατάμε OK, εικόνα 37.



Εικόνα 37: Eclipse ADT Plugin.

Μετά θα εμφανιστεί ένας φάκελος, τον επιλέγουμε και πατάμε Next. Μετά θα εμφανιστούν τα πακέτα που θα γίνουν εγκατάσταση και πατάμε ξανά Next. Τέλος πρέπει να αποδεχτούμε τους όρους και πατάμε Finish. Αν εμφανιστεί κατά την διάρκεια της εγκατάστασης κάποιο Warning το αποδεχόμαστε πατώντας OK. Αυτά τα βήματα φαίνονται στην ακόλουθη εικόνα 38.

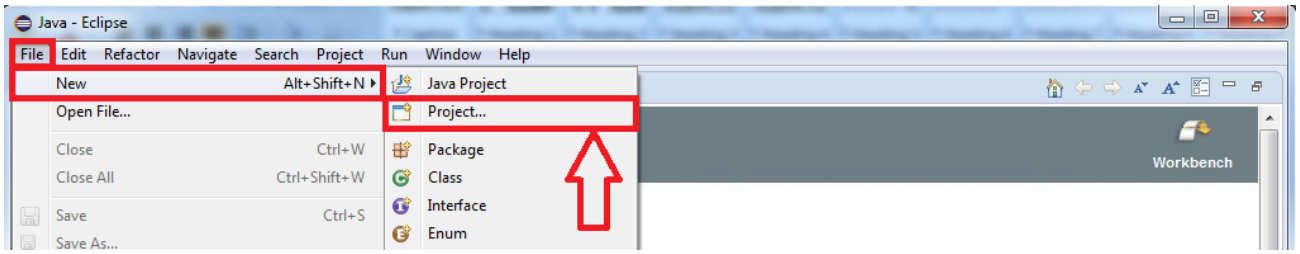


Εικόνα 38: Eclipse ADT Plugin.

Μόλις τελειώσει η εγκατάσταση κάνουμε επανεκκίνηση στο Eclipse και τώρα ο υπολογιστής μας είναι έτοιμος να δημιουργήσει ότι εφαρμογή Android επιθυμούμε. Στα παρακάτω βήματα θα δούμε πως φτιάξαμε την εφαρμογή Android που θα χρησιμοποιήσουμε στο Smart Home.

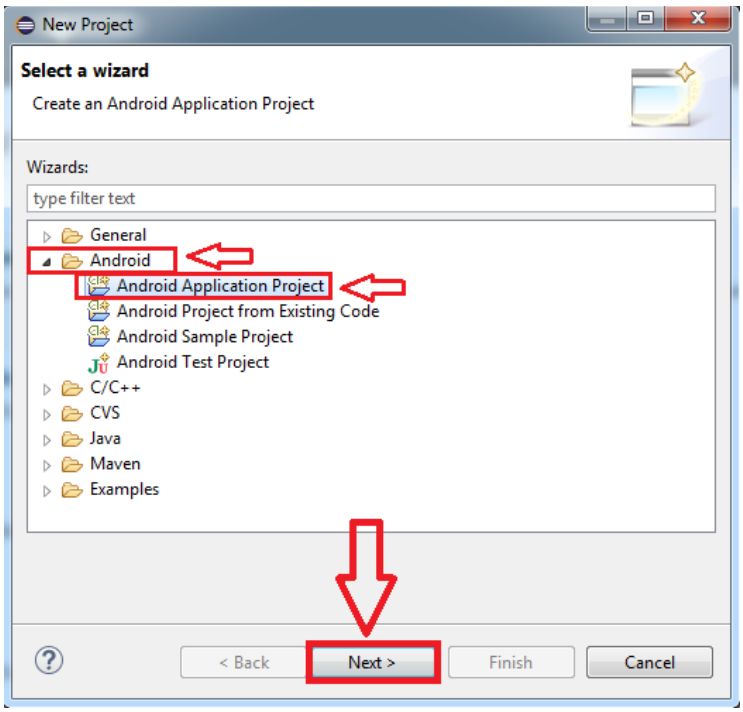
3.1.17 Δημιουργία Android εφαρμογής.

Για να δημιουργήσουμε την εφαρμογή ανοίγουμε το πρόγραμμα του Eclipse και πάνω αριστερά επιλέγουμε File, New, Project, εικόνα 39.



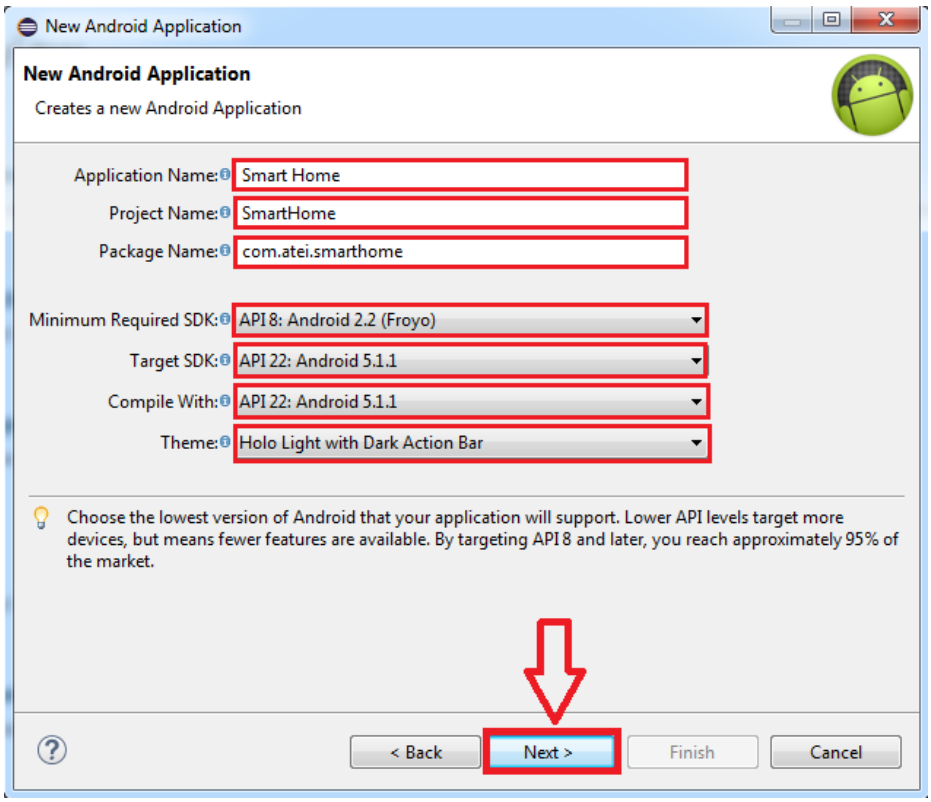
Εικόνα 39: Δημιουργία εφαρμογής Android

Μας εμφανίζει ένα καινούργιο παράθυρο στο οποίο επιλέγουμε τον φάκελο Android, Android Application Project και πατάμε Next, εικόνα 40.



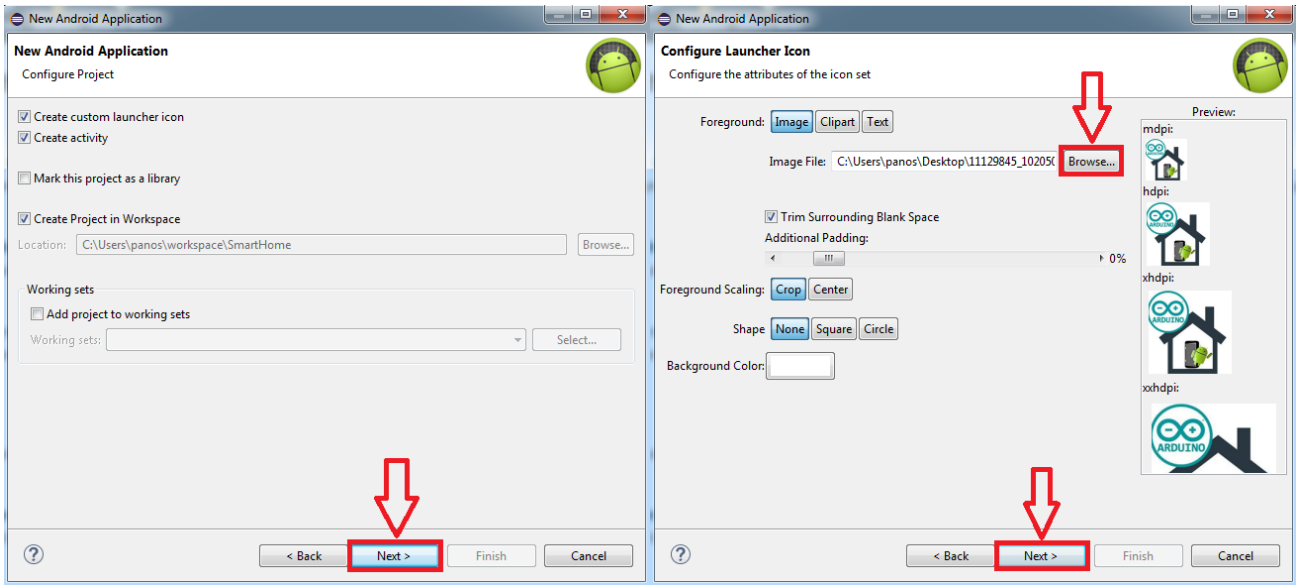
Εικόνα 40: Δημιουργία εφαρμογής Android

Μετά μας εμφανίζει ένα ακόμα παράθυρο στο οποίο θα πληκτρολογήσουμε το όνομα της εφαρμογής μας. Στην δικιά μας περίπτωση το όνομα είναι "Smart Home". Πιο αναλυτικά στο Application Name πληκτρολογούμε "Smart Home" ομοίως και στο Project Name. Στο Package Name θα πληκτρολογήσουμε "com.atei.smarthome", πρέπει να είναι σε τέτοια μορφή. Στις επιλογές που έχει ακριβώς από κάτω στο Minimum Required SDK, επιλέγουμε την παλαιότερη έκδοση που θα υποστηρίζει η εφαρμογή μας ενώ στο Target SDK επιλέγουμε την πιο πρόσφατη έκδοση που θα υποστηρίζει. Επίσης έχει και κάποιες άλλες τυπικές επιλογές όπως το θέμα που θα έχει η εφαρμογή, επιλέγουμε ένα αυτά που έχει και πατάμε Next. Η παραπάνω διαδικασία φαίνεται στην ακόλουθη εικόνα 41.



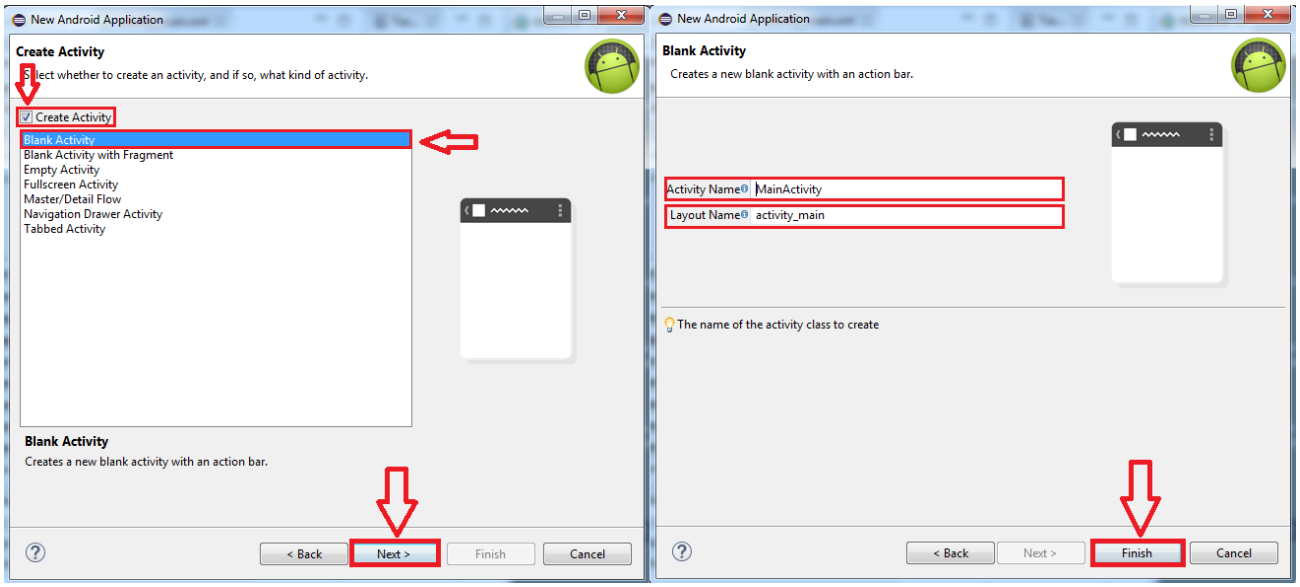
Εικόνα 41: Δημιουργία εφαρμογής Android

Στο επόμενο βήμα μας εμφανίζει ένα άλλο περιβάλλον, στο οποίο δεν χρειάζεται να κάνουμε κάποια επιλογή απλός πατάμε Next. Μετά θα μας εμφανίσει επιλογές για την εικόνα που θα έχει η εφαρμογή μας όταν την ανοίγουμε. Έχουμε την δυνατότητα να επιλέξουμε μια έτοιμη εικόνα από το πρόγραμμα του Eclipse και να την φτιάξουμε ή να δημιουργήσουμε μια δικιά μας. Στην δικιά μας περίπτωση δημιουργήσαμε μια εικόνα και πατώντας Browse επιλέγουμε την εικόνα και πατάμε Open. Όταν την φορτώσει πατάμε Next. Τα παραπάνω βήματα φαίνονται στην ακόλουθη εικόνα 42.



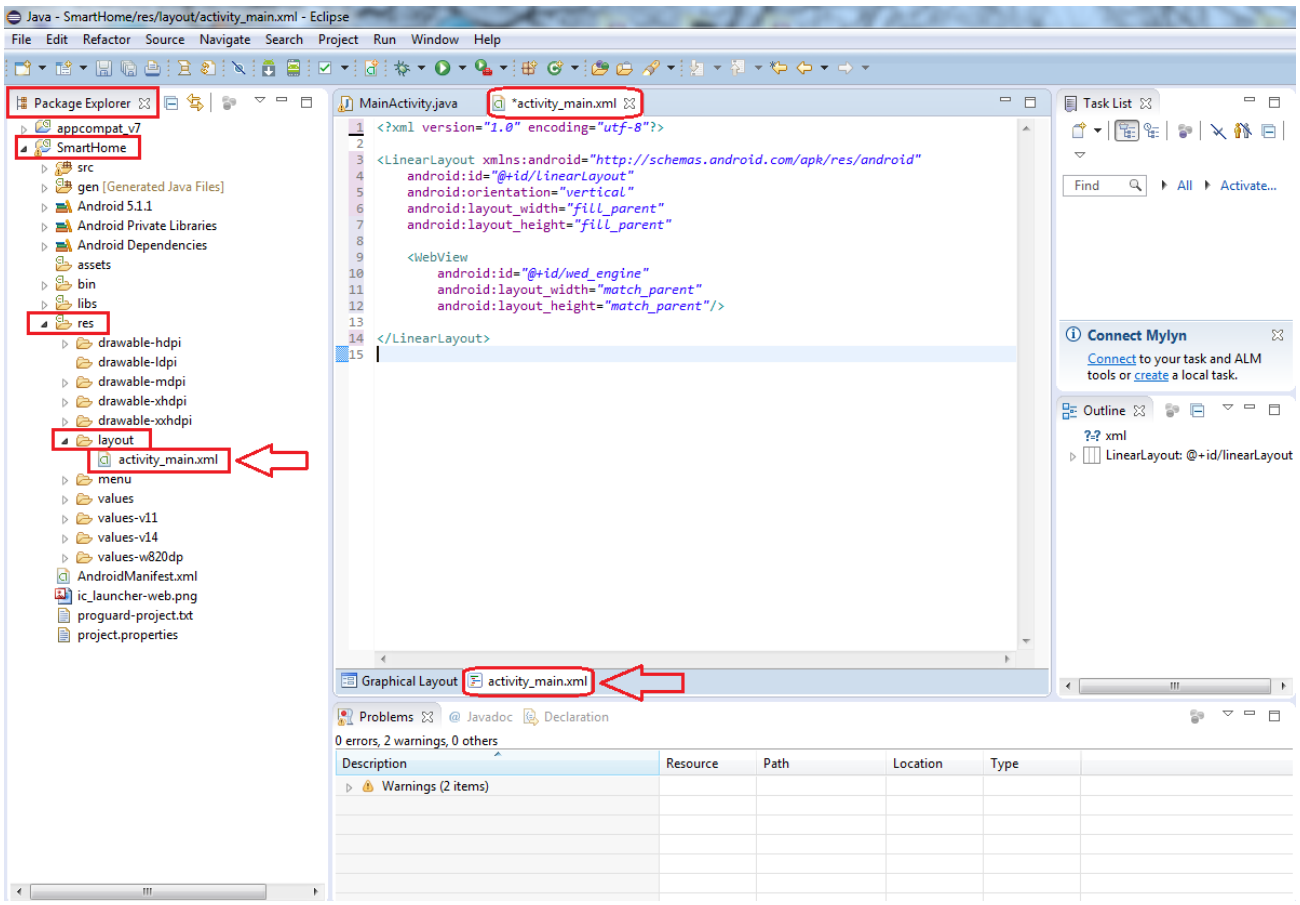
Εικόνα 42: Δημιουργία εφαρμογής Android

Στην συνέχεια επιβεβαιώνουμε ότι είναι επιλεγμένο το Create Activity ,επιλέγουμε το Blank Activity και πατάμε Next. Τέλος θα μας εμφανίσει το Activity Name και το Layout Name, όπου τα αφήνουμε όπως τα έχει δημιουργήσει το Blank Activity και πατάμε Finish. Στην ακόλουθη εικόνα 43 φαίνεται η παραπάνω διαδικασία.



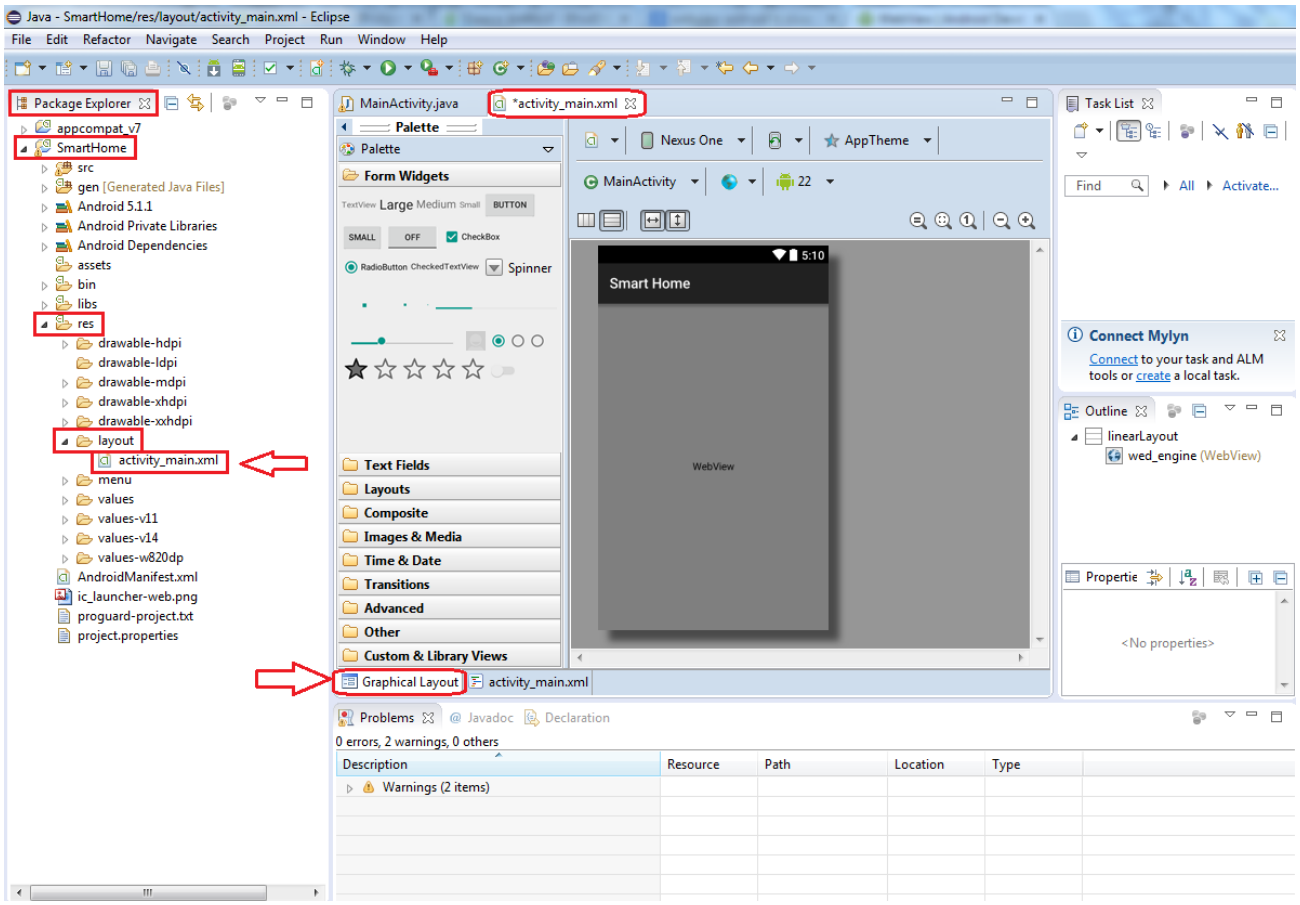
Εικόνα 43: Δημιουργία εφαρμογής Android

Βλέπουμε πάνω αριστερά στο Package Explorer ότι έχουν δημιουργηθεί κάποιοι φάκελοι που χρειάζονται στην εφαρμογή μας. Από το Package Explorer ανοίγουμε το δικό μας Project "Smart Home" πάμε στο φάκελο res, layout και επιλεγούμε το activity_main.xml. Θα μας εμφανίσει δυο παράθυρα, στο πρώτο παράθυρο θα μας εμφανίσει το activity_main.xml, στο οποίο παράθυρο θα έρθουμε να γράψουμε τον κώδικα για το Web View που θα μας εμφανίζει η εφαρμογή. Ο κώδικας του XML και η παραπάνω διαδικασία φαίνεται στην ακόλουθη εικόνα 44.



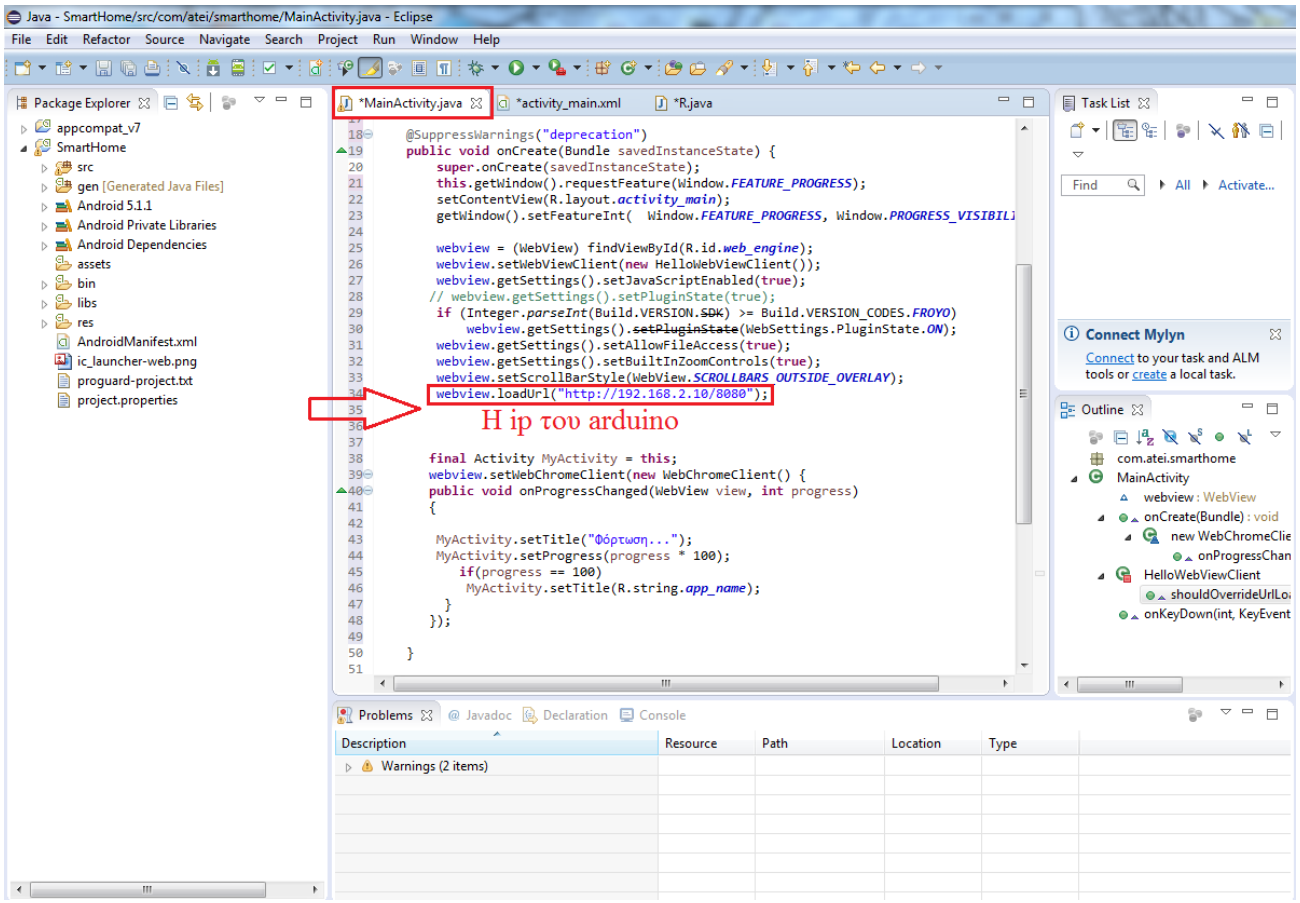
Εικόνα 44: activity_main.xml

Το δεύτερο παράθυρο που θα μας εμφανίσει καθώς έχουμε γράψει τον κώδικα του XML, είναι το Graphical Layout στο οποίο θα βλέπουμε το Web View της εφαρμογής. Το Web View φαίνεται στην εικόνα 45.



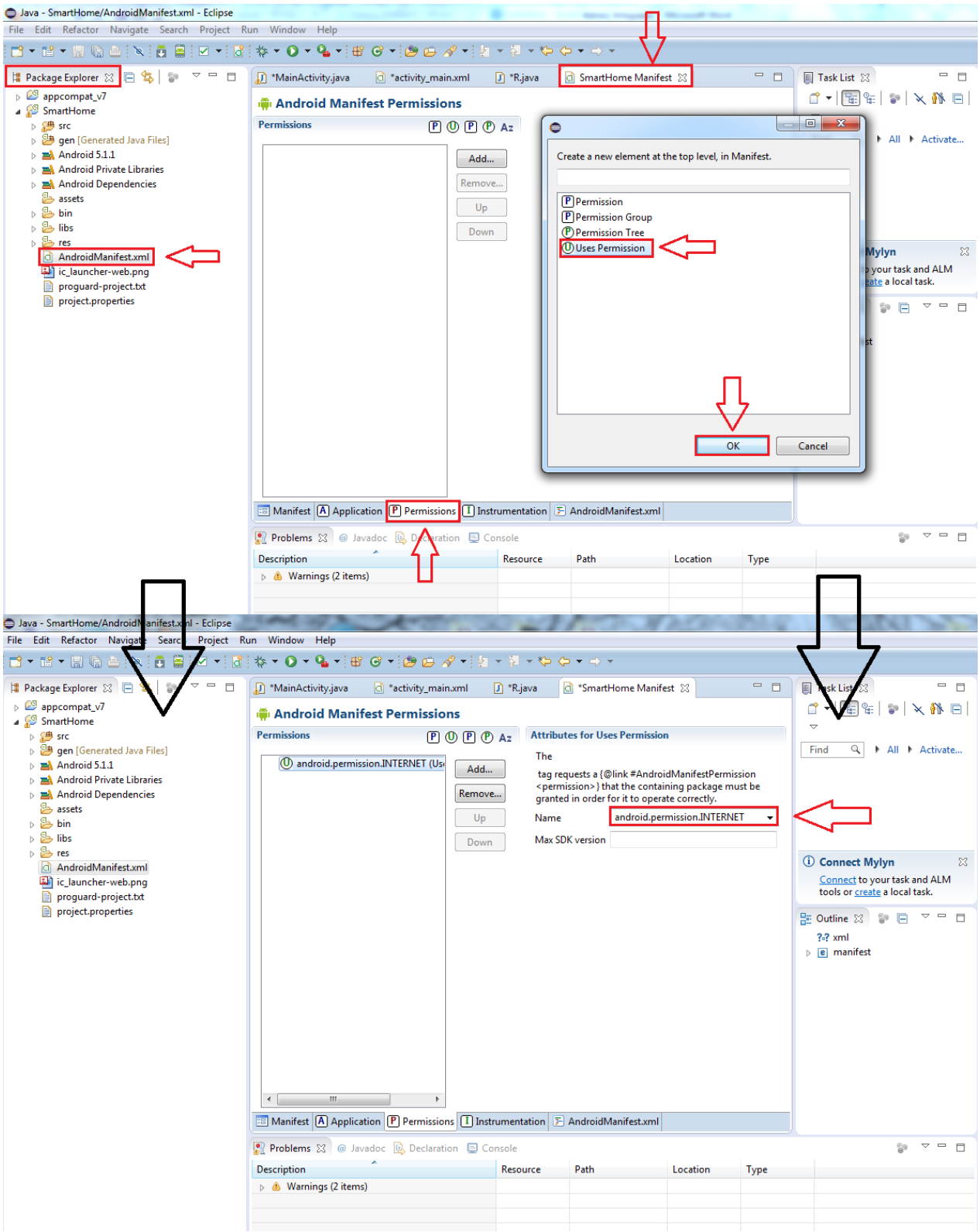
Εικόνα 45: Graphical Layout(Web View)

Στην συνέχεια θα πάμε στο MainActivity.java, για να γράψουμε τον κώδικα Java που θα χρησιμοποιήσουμε. Θα δούμε ένα μέρος του κώδικα αυτού. Πρέπει να προσέξουμε να βάλουμε την δικιά τοπική IP που βάλαμε και στον κώδικα του Arduino, ειδάλλως δεν θα γίνεται η επικοινωνία μεταξύ τους. Να προσθέσουμε επίσης ότι αργότερα που θα φτιάξουμε το DDNS, θα έρθουμε να αντικαταστήσουμε στην θέση τις IP, την διεύθυνση που δημιουργήσαμε στη DDNS, για να έχουμε τον απομακρυσμένο έλεγχο. Μέχρι στιγμής μπορούμε να μπαίνουμε μόνο τοπικά και να κάνουμε έλεγχο. Ο κώδικας Java που χρησιμοποιήσαμε φαίνεται στην ακόλουθη εικόνα 46.



Εικόνα 46: Κώδικας Java

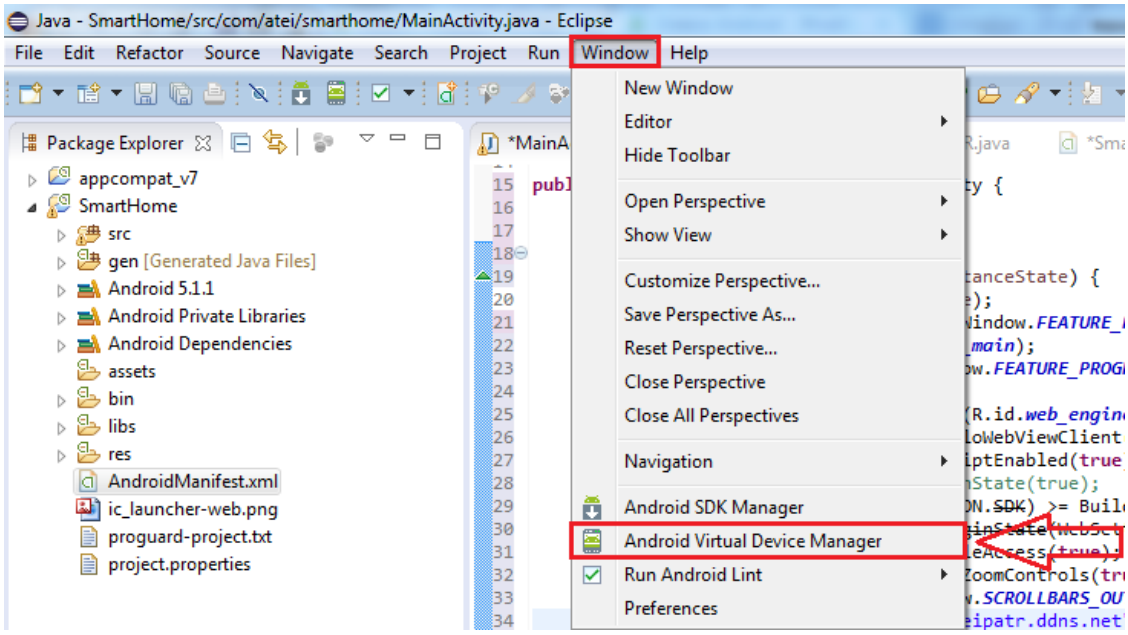
Επόμενο βήμα είναι να πάμε να δηλώσουμε στην εφαρμογή μας, ότι θα μπορεί να χρησιμοποιεί το Internet. Δηλαδή όταν θα πάμε να την εγκαταστήσουμε στην Android συσκευή, θα μας ζητήσει να έχει άδεια για να μπει στο Internet και πρέπει στην συνέχεια να το αποδεχτούμε. Για να δώσουμε αυτήν την άδεια, πάμε στο Package Explorer βρίσκουμε και επιλέγουμε το AndroidManifest.xml, Permissions, Add, κάνουμε κλικ στο Uses Permissions και πατάμε OK. Έπειτα στο καινούργιο παράθυρο που θα μας εμφανίσει, στο Name πληκτρολογούμε "android.permission.INTERNET". Η παραπάνω διαδικασία φαίνεται στην ακόλουθη εικόνα 47.



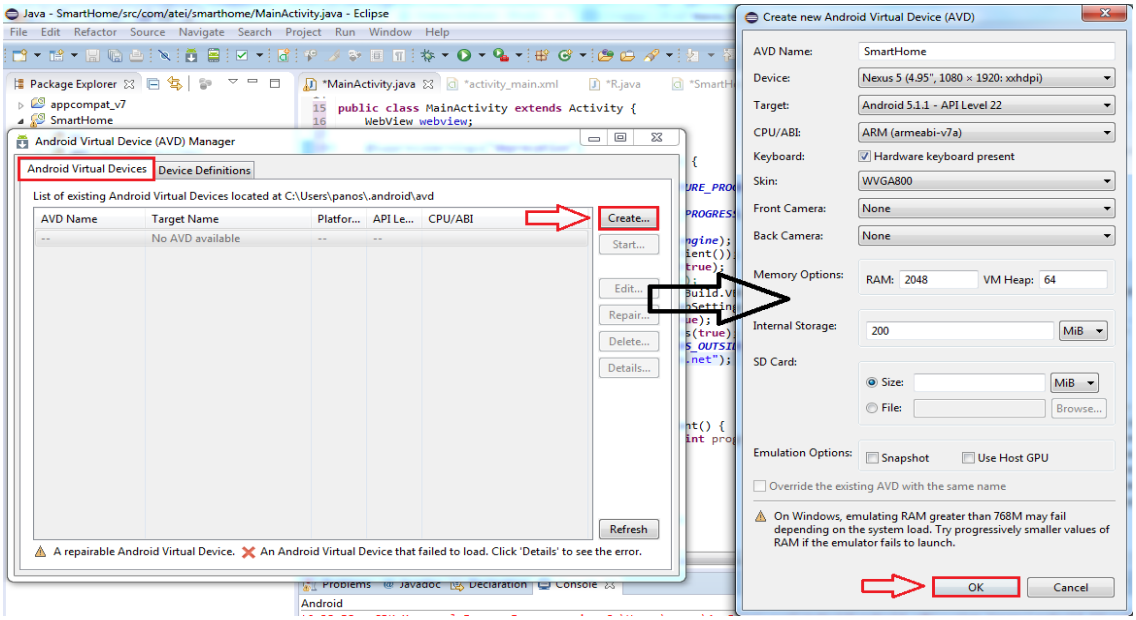
Εικόνα 47: Manifest Permission Internet

Τέλος πρέπει να τρέξουμε την εφαρμογή που έχουμε φτιάξει. Υπάρχουν πολύ τρόποι να το κάνουμε αυτό, ένας είναι από την εργαλειοθήκη να επιλέξουμε Windows, Android Virtual Device Manager, Creative και εκεί να φτιάξουμε με τις επιλογές που έχει, έναν Android Emulator και πατάμε OK. Αφού το φτιάξουμε πατάμε Start, μετά Launch και τρέχει ο Emulator. Μέχρι να ολοκληρωθεί η διαδικασία του Emulator θα περάσουν αρκετά λεπτά. Η διαδικασία αυτή φαίνεται στα ακόλουθα βήματα.

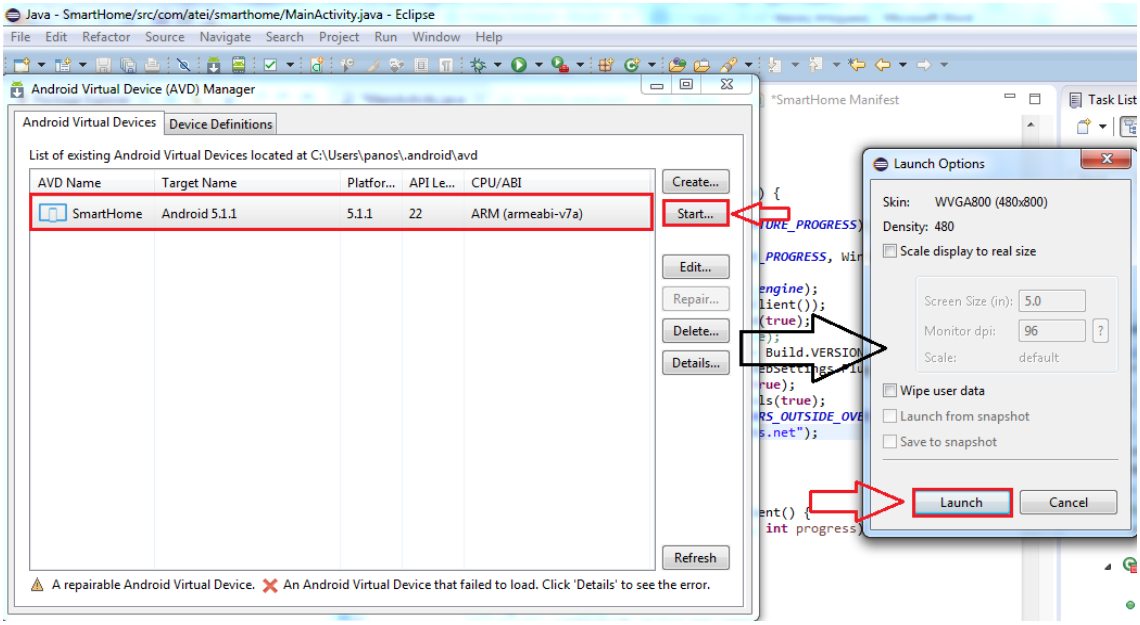
Βήμα 1.



Βήμα 2.



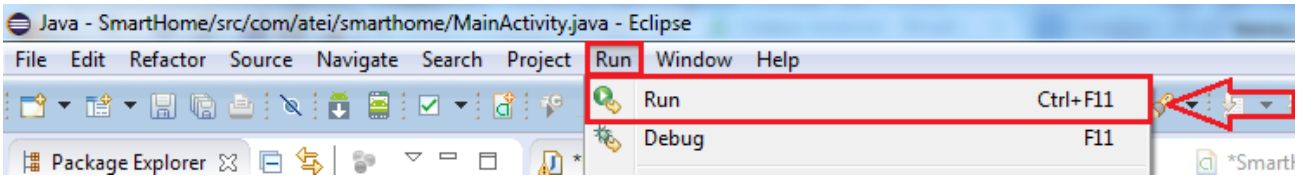
Βήμα 3.



Εικόνα 48: Launch

Όταν τρέξει το Launch πάμε στην εργαλειοθήκη και πατάμε Run, Run,εικόνα 49.

.



Εικόνα 49: Run

Για να πάρουμε την εφαρμογή και να την εγκαταστήσουμε στην Android συσκευή μας, πρέπει να βρούμε το APK αρχείο. Το αρχείο αυτό βρίσκεται μέσα στο Work Space. Άρα για να το βρούμε πάμε Work Space, Smart Home (που είναι το Project μας), Bin και μέσα σε αυτόν τον φάκελο υπάρχει το αρχείο "smarthome.apk". Πατάμε δεξί κλικ πάνω στο αρχείο και του κάνουμε αντιγραφή. Εφόσον έχουμε συνδέσει και την Android συσκευή μας με τον υπολογιστή, πάμε μέσα στα αρχεία της συσκευής και κάνουμε επικόλληση το "smarthome.apk" σε έναν φάκελο. Έπειτα αποσυνδέουμε την Android συσκευή με τον υπολογιστή, πάμε να βρούμε το αρχείο που κάναμε επικόλληση και του κάνουμε εγκατάσταση. Κατά την εγκατάσταση θα πρέπει να ρυθμίσουμε την Android συσκευή να δέχεται άγνωστες πηγές. Αυτό πρέπει να γίνει γιατί η εφαρμογή δεν έχει ληφθεί από το κατάστημα Play και για λόγους ασφαλείας την αποκλείει. Τώρα είμαστε σε θέση να ανοίξουμε την εφαρμογή και να έχουμε τον έλεγχο του Smart Home.

3.1.18 Δημιουργία DDNS.

Για να έχουμε και απομακρυσμένο έλεγχο του Smart Home πρέπει να φτιάξουμε έναν DDNS λογαριασμό. Για να τον φτιάξουμε, πάμε στο ακόλουθο URL"http://www.noip.com/" δημιουργούμε έναν Account και κάνουμε Sing In. Επόμενο βήμα είναι να πάμε να επιλέξουμε το Add Host που βρίσκεται πάνω αριστερά στην ιστοσελίδα και να κάνουμε τις εξής ρυθμίσεις:

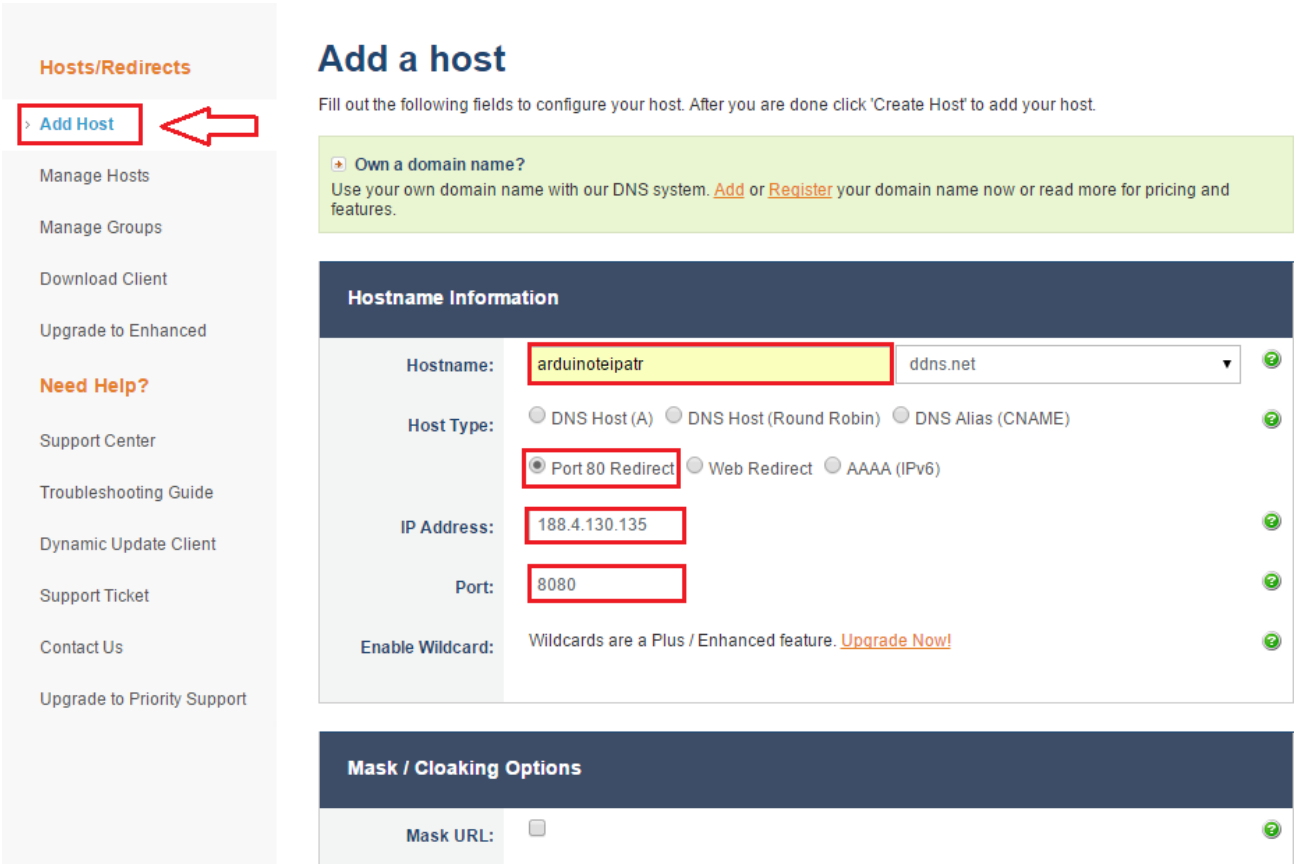
1. Στο Hostname θα γράψουμε το όνομα του URL που θα πληκτρολογούμε στο Web Browser. Στην δικιά μας περίπτωση είναι το "arduinoteipatr.ddns.net".

2. Στο Host Type πάμε και επιλέγουμε το "Port 80 Redirect".

3. Στην IP Address θα πληκτρολογούμε την εξωτερική IP του Router. Αλλάζει ανάλογα με το δίκτυο που χρησιμοποιούμε.

4. Στο Port θα βάλουμε την πόρτα την οποία ανοίξαμε και βάλαμε στο Arduino. Στην δικιά μας περίπτωση είναι η "8080".

Εφόσον έχουμε κάνει τις ρυθμίσεις αυτές πατάμε OK. Στην ακόλουθη εικόνα 50 φαίνονται τα παραπάνω βήματα.



Εικόνα 50: Δημιουργία DDNS

3.1.19 Ρυθμίσεις στο Router για το DDNS.

Για να λειτουργήσει ο λογαριασμός DDNS που φτιάξαμε πρέπει να κάνουμε Port Forward στο Router. Το Port Forward είναι η διαδικασία που πρέπει να γίνει, για να είναι ένας υπολογιστής που βρίσκεται πίσω από ένα Router, προσβάσιμος από άλλους υπολογιστές στο διαδίκτυο. Για να γίνουν οι ρυθμίσεις αυτές, πρέπει αρχικά να μπούμε στο Router. Για να μπούμε, πληκτρολογούμε στον Web Browser την τοπική IP του Router και εισάγουμε τους κωδικούς μας. Όταν μπούμε μέσα πάμε και βρίσκουμε την επιλογή NAT, Virtual Server και κάνουμε τις εξής ρυθμίσεις:

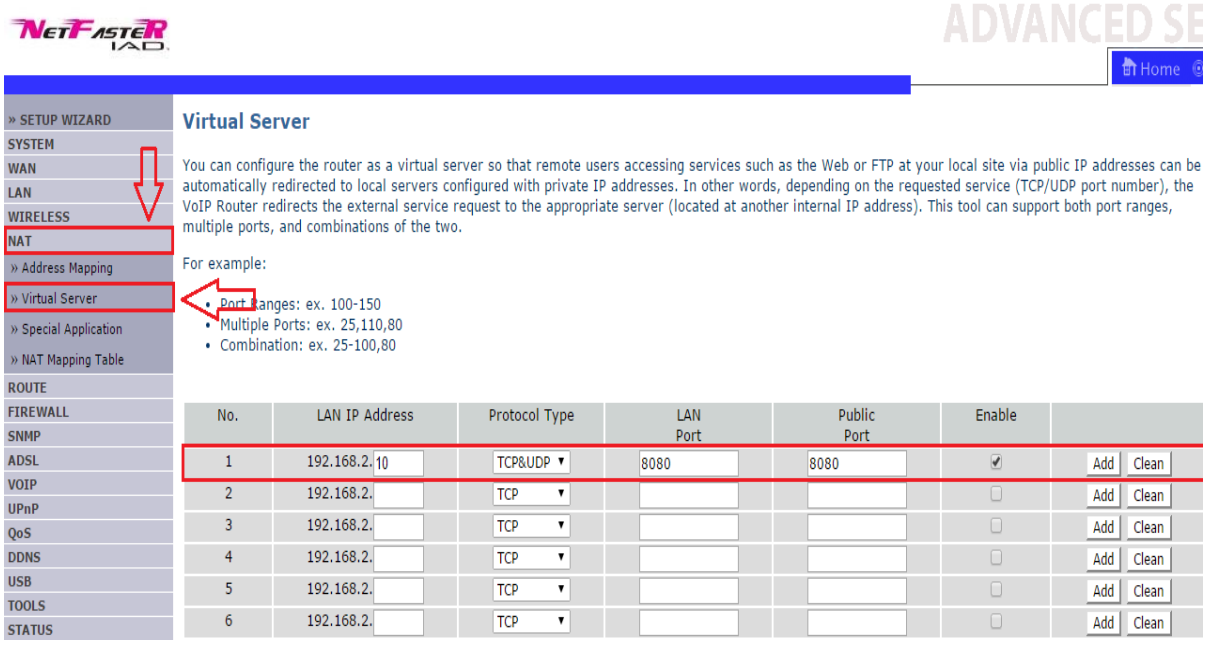
1. Στο LAN IP Address πληκτρολογούμε την IP που έχουμε βάλει στο πρόγραμμα του Arduino"192.168 .2.10".

2. Στο Protocol Type επιλέγουμε το TCP&UDP.

3. Στο LAN Port πληκτρολογούμε την πόρτα που ανοίξαμε και βάλαμε στο Arduino "8080".

4. Στο Public Port πληκτρολογούμε επίσης την πόρτα 8080.

Εφόσον έχουμε κάνει τις ρυθμίσεις αυτές πατάμε SAVE SETTINGS. Στην ακόλουθη εικόνα 51 φαίνονται τα παραπάνω βήματα.

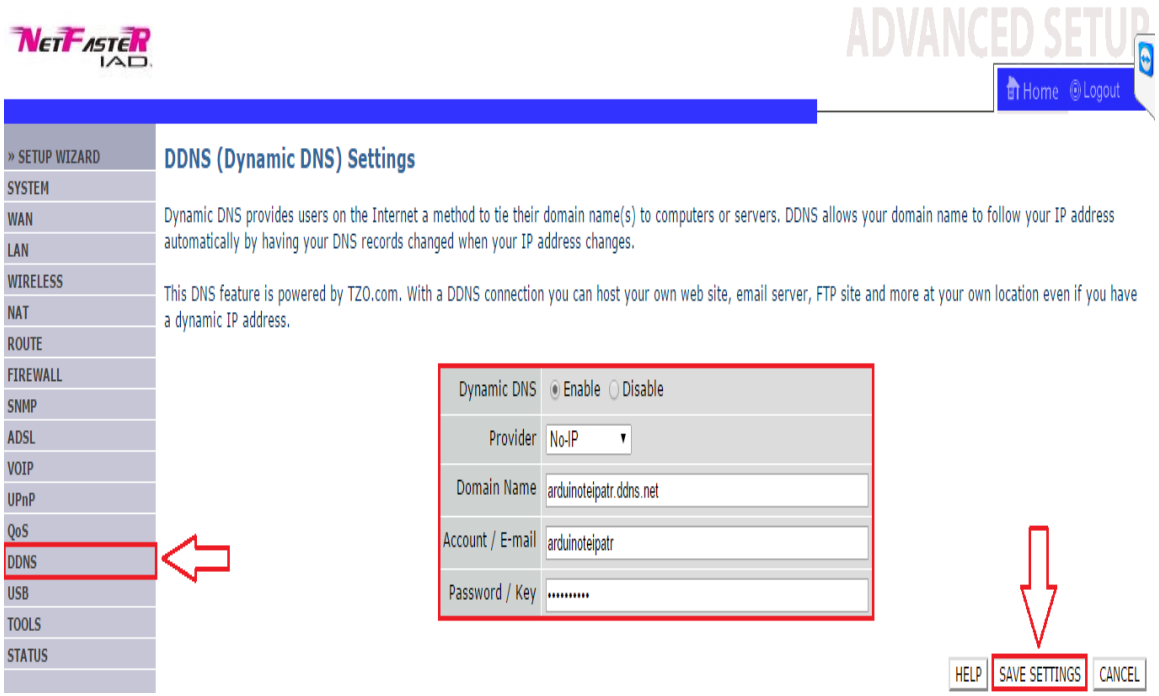


Εικόνα 51: Ρυθμίσεις στο Virtual Server του Router

Στην συνέχεια πρέπει να γίνουν ρυθμίσεις και στο DDNS μέσα στο Router. Οπότε βρίσκουμε το DDNS και εκεί κάνουμε τις εξής ρυθμίσεις:

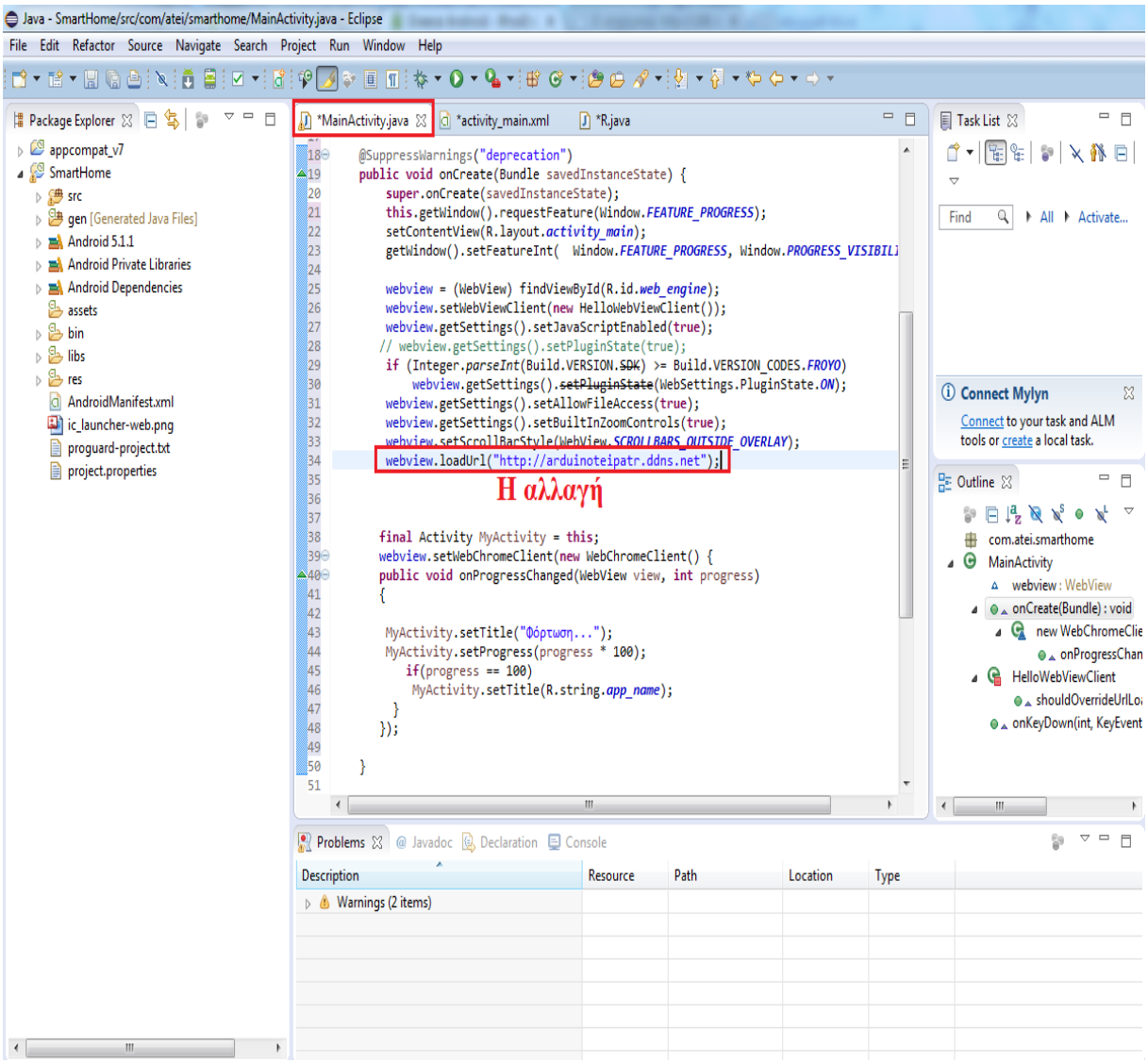
1. Στο Dynamic DNS επιλέγουμε Enable.
2. Στο Provider επιλέγουμε No-IP.
3. Στο Domain Name πληκτρολογούμε την διεύθυνση DDNS «arduinoteipart .ddns.net».
4. Στο Account και το Password βάζουμε τα στοιχεία μας από την No-IP ιστοσελίδα που φτιάξαμε.

Εφόσον έχουμε κάνει τις ρυθμίσεις αυτές πατάμε SAVE SETTINGS. Στην ακόλουθη εικόνα 52 φαίνονται τα παραπάνω βήματα.



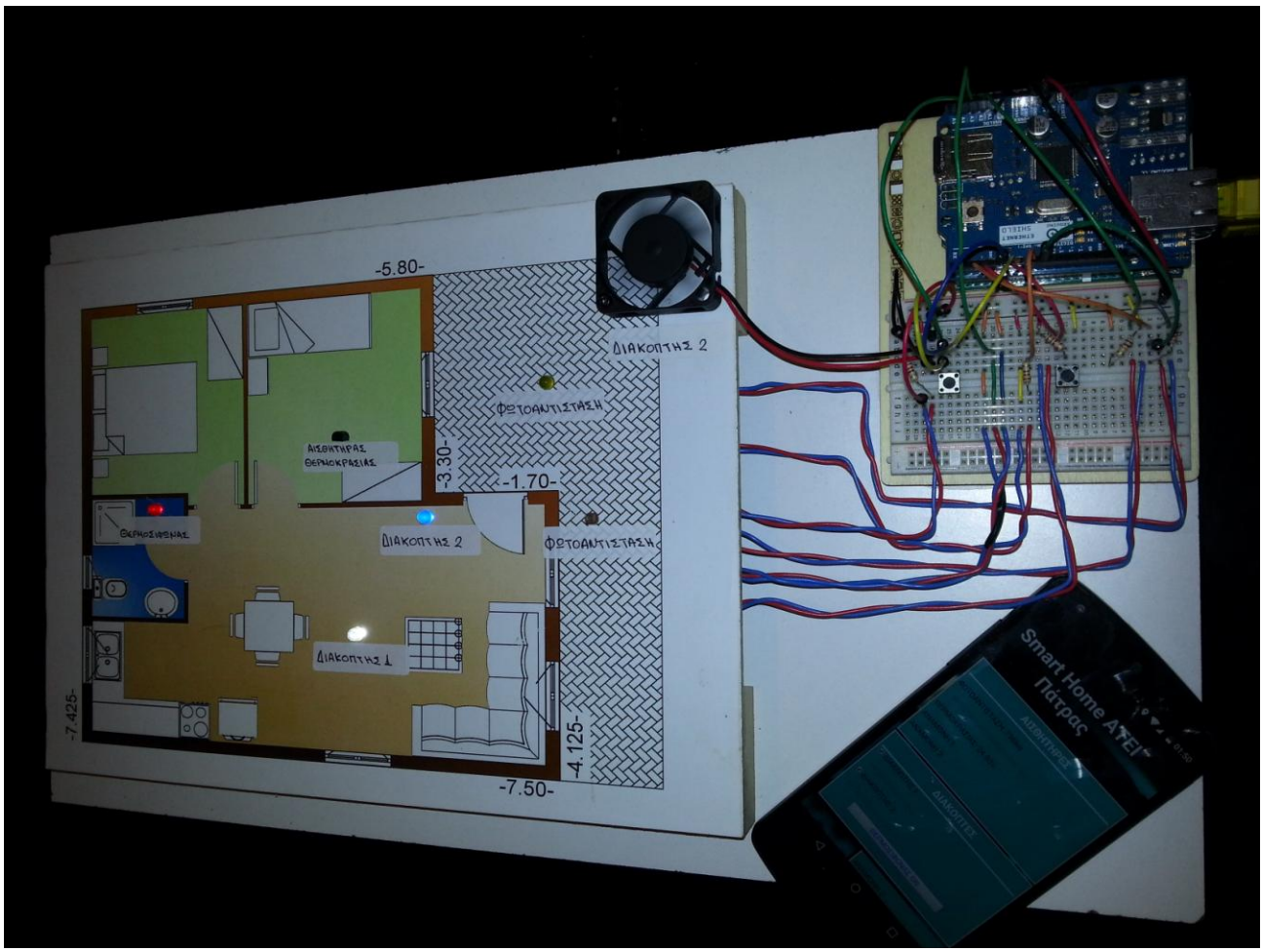
Εικόνα 52: Ρυθμίσεις στο DDNS του Router

Μέχρι στιγμής έχουμε φτιάξει τον απομακρυσμένο έλεγχο που επιτυγχάνετε μόνο μέσω του οποιοδήποτε Wed Browser όχι όμως και από την εφαρμογή. Για να έχουμε και από την εφαρμογή τον απομακρυσμένο έλεγχο πρέπει να κάνουμε μόνο μια αλλαγή. Η αλλαγή αυτή είναι να ανοίξουμε την εφαρμογή που φτιάξαμε στα προηγούμενα βήματα, να πάμε στο MainActivity.java και εκεί που είχαμε βάλει την τοπική IP του Router να βάλουμε πλέον το DDNS "arduinoteipart.ddns.net". Μετά να την αποθηκεύσουμε και να την εγκαταστήσουμε ξανά στην συσκευή android. Η αλλαγή αυτή φαίνεται στην ακόλουθη εικόνα 53. Το βήμα αυτό ήταν το τελευταίο για το πειραματικό μας μέρος.



Εικόνα 53: Αντικατάσταση DDNS στην Android εφαρμογή

Το τελικό αποτέλεσμα Android – Arduino.



Εικόνα 54: Μακέτα πειραματικού μέρους

Βιβλιογραφία.

[1] Επίσημη ιστοσελίδα Arduino, http://www.arduino.cc/ , (Ανάκτηση την 14/4/15).

[2] Ιστοσελίδα αναπτυγμένης τεχνολογίας, http://www.learn.adafruit.com/ , (Ανάκτηση την 3/5/15).

[3] Επίσημη ιστοσελίδα Android, http://developer.android.com/index.html , (Ανάκτηση την 26/5/15).

[4] Ελληνικό Forum Android , http://www.greeceandroid.gr/ , (Ανάκτηση την 2/6/15).

[5] Ιστοσελίδα Wiki, https://en.wikipedia.org/wiki/Wiki , (Ανάκτηση την 6/4/15).

[6] Πρόγραμμα Arduino http://www.arduino.cc/en/Main/Software , (Ανάκτηση την 10/3/15).

[7] Βιβλιοθήκες Arduino , http://www.arduino.cc/en/Reference/Libraries , (Ανάκτηση την 10/3/15).

[8] Drivers , http://www.arduino.cc/en/Guide/UnoDriversWindowsXP , (Ανάκτηση την 10/3/15).

[9] Πρόγραμμα HTML , https://notepad-plus-plus.org/ , (Ανάκτηση την 12/3/15).

[10] JDK , http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html , (Ανάκτηση την 17/2/15).

[11] SDK , https://developer.android.com/sdk/index.html , (Ανάκτηση την 20/03/15).

[12] Eclipse , http://www.eclipse.org/downloads , (Ανάκτηση την 15/04/15).

[13] DDNS , http://www.noip.com , (Ανάκτηση την 2/06/15).