



ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΔΥΤΙΚΗΣ ΕΛΛΑΔΟΣ (ΠΑΤΡΑ)
ΣΧΟΛΗ : ΣΤΕ
ΤΜΗΜΑ : ΗΛΕΚΤΡΟΛΟΓΟΙ ΜΗΧΑΝΙΚΟΙ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
ΑΡΙΘΜΟΣ. ΠΤΥΧΙΑΚΗΣ 1351

**ΣΤΑΘΜΟ ΣΥΛΛΟΓΗΣ ΔΕΔΟΜΕΝΩΝ
ΜΕ ΤΗΝ ΠΛΑΚΕΤΑ ARDUINO
ΚΑΙ ΤΗΝ ΔΥΝΑΤΟΤΗΤΑ ΑΝΑΡΤΗΣΗΣ ΣΤΟ ΔΙΑΔΙΚΤΥΟ
ΜΕΣΩ ΠΛΑΤΦΟΡΜΑΣ INTERNET OF THINGS**

ΕΠΙΜΕΛΕΙΑ: ΚΟΛΙΟΠΟΥΛΟΣ ΑΘΑΝΑΣΙΟΣ
ΜΑΣΟΥΡΗΣ ΠΑΡΑΣΚΕΥΑΣ

ΕΙΣΗΓΗΤΗΣ: ΦΩΤΟΠΟΥΛΟΣ ΒΑΣΙΛΕΙΟΣ
ΒΛΑΧΟΠΟΥΛΟΣ ΠΕΤΡΟΣ

ΠΑΤΡΑ 2014

Θα θέλαμε να ευχαριστήσουμε,

Τον επιβλέποντα της πτυχιακής μας εργασίας κύριο Βασίλη Φωτόπουλο τόσο για την αμέριστη βοήθεια του για την εκπόνηση της πτυχιακής όσο και για την πίστη του για την επιτυχή ολοκλήρωση της. Είχαμε την τύχη να τον έχουμε καθηγητή στο εργαστήριο των Ψηφιακών Συστημάτων και όπως και τότε έτσι και τώρα ήταν πρόθυμος και συνεργάσιμος μαζί μας και κυρίως με ανθρώπινη αντιμετώπιση ειδικά όταν τα χρονικά περιθώρια είχαν στενέψει ασφυκτικά.

Τέλος θα θέλαμε να ευχαριστήσουμε τις οικογένειες μας για την συμπαράσταση τους και την πολύτιμη ψυχολογική ενθάρρυνση καθώς και για την τεράστια υπομονή τους.

*Σε αυτούς που πίστεψαν σε μας,
Κολιόπουλος Θάνος
Μασούρης Παρασκευάς
Πάτρα, Ιούνιος, 2014*

ΠΕΡΙΕΧΟΜΕΝΑ

| | |
|--|----------|
| Πρόλογος | 1 |
| ΚΕΦΑΛΑΙΟ 1 | 2 |
| 1.1. Αρχιτεκτονική Arduino | 2 |
| 1.1.1. Περιγραφή του Arduino | 3 |
| 1.1.2. Ρεύμα | 4 |
| 1.1.3. Μνήμη | 5 |
| 1.1.4. Είσοδος και Έξοδος | 5 |
| 1.1.5. Επικοινωνία | 6 |
| 1.1.6. USB Προστασία Διαρροής Ρεύματος | 7 |
| 1.1.7. Φυσικά χαρακτηριστικά της πλατφόρμας | 7 |
| 1.1.8. Πλεονεκτήματα της πλατφόρμας Arduino | 7 |
| 1.2. Προγραμματισμός του Arduino | 8 |
| 1.2.1 Ολοκληρωμένο Περιβάλλον Ανάπτυξης του Arduino | 9 |
| 1.2.2 Μενού Edit (Επεξεργασία) | 11 |
| 1.2.3. Μενού Sketch (Σχέδιο) | 11 |
| 1.2.4. Μενού Tools (Εργαλεία) | 11 |
| 1.2.5. Sketchbook | 12 |
| 1.2.6 Tabs and Multiple Files (Καρτέλες και Πολλαπλά αρχεία) | 12 |
| 1.2.7 Uploading (Ανέβασμα) | 12 |
| 1.2.8 Libraries (Βιβλιοθήκες) | 13 |
| 1.2.9 Serial Monitor (Σειριακή Οθόνη) | 14 |
| 1.2.10. setup() και loop() | 15 |
| 1.2.11. Digital Pins (Ψηφιακές Ακίδες) | 15 |
| 1.2.12. Analog Input Pins (Αναλογικές Ακίδες Εισόδου) | 16 |
| 1.2.13. Pin mapping (Χαρτογράφηση ακίδων) | 17 |
| 1.2.14 Pullup Αντιστάσεις | 17 |
| 1.2.15 PWM (Pulse Width Modulation) | 18 |

| | |
|--|-----------|
| 1.2.16. Μνήμη | 19 |
| 1.2.17. Σειριακή Θύρα | 20 |
| ΚΕΦΑΛΑΙΟ 2 ΛΑΜΒΑΝΟΝΤΑΣ ΕΙΣΟΔΟ ΑΠΟ ΑΙΣΘΗΤΗΡΕΣ | 22 |
| 2.1.Εισαγωγή | 22 |
| 2.1.1. Ψηφιακή on / off | 23 |
| 2.1.2. Αναλογική | 23 |
| 2.1.3 Εύρος παλμού | 23 |
| 2.1.4 Σειριακή | 23 |
| 2.1.5 Σύγχρονα πρωτόκολλα: I2C και SPI | 24 |
| 2.2.Ανίχνευση Φωτός | 25 |
| 2.2.1. Πρόβλημα | 25 |
| 2.2.2. Λύση | 26 |
| 2.2.3 Συζήτηση | 26 |
| 2.3. Ανίχνευση κίνησης (Ενσωμάτωση Ανιχνευτών Παθητικών Υπερύθρων) | 27 |
| 2.3.1 Πρόβλημα | 27 |
| 2.3.2 Λύση | 27 |
| 2.3.3 Συζήτηση | 29 |
| 2.4. Μέτρηση Απόστασης | 30 |
| 2.4.1 Πρόβλημα | 30 |
| 2.4.2 Λύση | 30 |
| 2.4.3 Συζήτηση | 32 |
| 2.4.4 Δείτε επίσης | 35 |
| 2.5 Μέτρηση της θερμοκρασίας | 36 |
| 2.5.1 Πρόβλημα | 36 |
| 2.5.2 Λύση | 36 |
| 2.5.3 Συζήτηση | 37 |
| 2.6 Ανάγνωση ετικετών RFID | 39 |
| 2.6.1 Πρόβλημα | 39 |

| | |
|---|-----------|
| 2.6.2 Λύση | 39 |
| 2.6.3 Συζήτηση | 41 |
| ΚΕΦΑΛΑΙΟ 3 ΕΙΣΑΓΩΓΗ ΣΤΟ INTERNET ΤΩΝ ΠΡΑΓΜΑΤΩΝ | 44 |
| 3.1. Το Internetτων πραγμάτων | 45 |
| 3.1.2 Οι βασικές έννοιες | 47 |
| 3.1.3 Αλληλεπίδραση με το Internet | 49 |
| 3.1.4 Βασικά Εξαρτήματα από IoT συσκευές | 50 |
| 3.1.5 Μονάδες Ελέγχου | 50 |
| 3.1.6 Αισθητήρες | 52 |
| 3.1.7 Ενότητες Επικοινωνίας | 53 |
| 3.1.8 Πηγές ενέργειας | 56 |
| 3.1.9 Τεχνολογίες Επικοινωνίας | 57 |
| 3.1.10 RFID | 57 |
| 3.1.11 Bluetooth | 59 |
| 3.1.12 ZigBee | 60 |
| 3.1.13 WiFi | 61 |
| 3.1.14 Συνδέσεις RF | 62 |
| 3.1.15 Κυψελοειδήδίκτυα :Το Mobile Internet | 63 |
| 3.1.16 Καλωδιακή επικοινωνία | 64 |
| 3.1.17 Ποιο είναι το καλύτερο? | 65 |
| 3.1.18 Τρέχουσα Κατάσταση και Κοντινό Μέλλον | 66 |
| 3.1.19 Περίληψη | 68 |
| ΚΕΦΑΛΑΙΟ4 ΥΛΟΠΟΙΗΣΗ | 69 |
| ΒΙΒΛΙΟΓΡΑΦΕΙΑ | 74 |

ΠΡΟΛΟΓΟΣ

Στην πτυχιακή εργασία αυτή θα παρουσιάσουμε ένα σταθμό συλλογής δεδομένων με την πλακέτα Arduino και την δυνατότητα ανάρτησης στο διαδίκτυο μέσω πλατφόρμας InternetofThings. Σκοπός μας είναι να παρουσιάσουμε τις δυνατότητες που έχει η πλακέτα Arduino καθώς και την χρήση μιας πλατφόρμας IoT με την επίδειξη της σύνδεσης δύο αισθητήρων συνδεδεμένων πάνω στην πλακέτα.

ΚΕΦΑΛΑΙΟ 1

1.1. Αρχιτεκτονική Arduino

Το Arduino είναι μία φυσική υπολογιστική πλατφόρμα ανοιχτού λογισμικού ηλεκτρονικής προτυποποίησης η οποία βασίζεται σε ένα ευέλικτο και εύκολο στη χρήση υλικό και λογισμικό. Το Arduino με τους κατάλληλους αισθητήρες μπορεί να ανιχνεύει το περιβάλλον του και μπορεί να αντιδράσει ανάλογα, για παράδειγμα με τη χρήση λαμπτήρων, μοτέρ και άλλα. Ο μικροελεγκτής του κυκλώματος προγραμματίζεται χρησιμοποιώντας την γλώσσα προγραμματισμού Wiring και το περιβάλλον προγραμματισμού του Arduino (το οποίο είναι βασισμένο στο περιβάλλον του Processing).

Τα διάφορα έργα που μπορούν να υλοποιηθούν στο Arduino μπορούν να είναι αυτόνομα είτε να επικοινωνούν με άλλες εφαρμογές του υπολογιστή όπως το Flash, Processing και MaxMSP. Οι πλακέτες αυτές μπορούν να κατασκευαστούν στο χέρι ή μπορούν να αγοραστούν συναρμολογημένες. Το λογισμικό είναι δωρεάν και μπορεί να το κατεβάσει ο καθένας από το Διαδίκτυο, από την ιστοσελίδα www.arduino.cc. Ακόμη, τα σχέδια του κυκλώματος προσφέρονται και αυτά δωρεάν και μπορεί να τα προσαρμόσει ο καθένας στις δικές του ανάγκες.



Εικόνα 1. ArduinoMega 1280 (μπροστά όψη)

1.1.1 Περιγραφή του Arduino

Υπάρχουν πολλές διαφορετικές υλοποιήσεις της πλακέτας Arduino (ArduinoUno, ArduinoMega 2560, ArduinoPro κτλ), οι οποίες έχουν και διαφορετικά χαρακτηριστικά, και μπορεί οποιοσδήποτε να τις βρει στο διαδίκτυο, μέσω της ιστοσελίδας του [Arduino](http://www.arduino.cc/en/Main/Buy) www.arduino.cc/en/Main/Buy ή από άλλες. Εδώ θα γίνει περιγραφή της έκδοσης η οποία χρησιμοποιήθηκε στην εργασία αυτή. Το μοντέλο φέρει την ονομασία ArduinoMega 1280 (εικόνα 1). Το μοντέλο αυτό είναι πλακέτα η οποία βασίζεται στον μικροελεγκτή 8-bitAtmega1280 της Atmel. Η πλακέτα περιλαμβάνει τα εξής:

- § 54 ψηφιακές εισόδους / εξόδους (από τις οποίες οι 14 μπορούν να χρησιμοποιηθούν σαν έξοδοι διαμόρφωσης πλάτους-παλμών (PWMoutput))
- § 16 αναλογικές εισόδους
- § 4 UARTS (σειριακές θύρες)
- § 16 MHz κρυσταλλικό ταλαντωτή (παράγει παλμούς που χρονίζουν τον μικροελεγκτή σε μια δεδομένη συχνότητα)
- § μία σύνδεση USB

§ μία έξοδος ρεύματος

§ μια κεφαλίδα σειριακού προγραμματισμού κυκλώματος (ICSP - InCircuitSerialProgramming), σε περίπτωση που θέλουμε να χρησιμοποιήσουμε εξωτερική συσκευή προγραμματισμού AVR μικροελεγκτών

§ ένα κουμπί επαναφοράς (Reset)

Το μόνο που χρειάζεται για να ξεκινήσει κανείς είναι είτε να το συνδέσει στο ρεύμα είτε να το συνδέσει στον υπολογιστή με το USB.

1.1.2. Ρεύμα

Η πλακέτα μπορεί να πάρει ρεύμα είτε από το USB είτε από εξωτερική πηγή ρεύματος. Αν χρησιμοποιηθεί εξωτερική τροφοδοσία τότε αυτή μπορεί να είναι είτε από μετασχηματιστή με βύσμα 2.1 χιλιοστών με κέντρο θετικής τάσης είτε από μπαταρία η οποία μπορεί να συνδεθεί στις αντίστοιχες εξόδους Gnd και Vin.

Η πλακέτα μπορεί να λειτουργήσει με εξωτερική τροφοδοσία μεταξύ 6V και 20V. Σε περίπτωση που εφαρμοστεί τάση μικρότερη από 7V τότε η έξοδος των 5V θα παρέχει λιγότερα από 5V και η πλακέτα θα είναι ασταθής. Σε αντίθετη περίπτωση που εφαρμοστούν περισσότερα από 12V, ο ρυθμιστής τάσης ενδέχεται να υπερθερμανθεί και να προξενήσει βλάβη στην πλακέτα. Οπότε το προτεινόμενο εύρος τάσης για την ομαλή λειτουργία της πλακέτας είναι 7V με 12V.

Οι ακίδες ρεύματος είναι οι παρακάτω:

§ VIN: Η τάση εισόδου στην πλακέτα Arduino όταν χρησιμοποιεί μια εξωτερική πηγή ισχύος (και όχι με 5V από τη σύνδεση USB ή από άλλη ρυθμισμένη πηγή ενέργειας). Μπορεί να τροφοδοτηθεί μέσω αυτής της ακίδας ή αν η τάση παρέχεται με το powerjack, δίνεται πρόσβαση μέσω αυτής της ακίδας.

§ 5V: Η ρυθμιζόμενη παροχή ηλεκτρικού ρεύματος που χρησιμοποιείται για την τροφοδοσία του μικροελεγκτή και άλλα στοιχεία στην πλακέτα. Αυτό μπορεί να προέρχεται είτε από VIN με ενσωματωμένο ρυθμιστή, ή να παρέχεται από USB ή άλλη ρυθμισμένη παροχή 5V.

§ 3V: Μία παροχή 3.3V που παράγονται από το ενσωματωμένο chipFTDI. Μέγιστη κατανάλωση ρεύματος είναι 50 mA.

§ GND: Γειωμένες ακίδες (η πλατφόρμα διαθέτει τρεις).

1.1.3. Μνήμη

Ο μικροελεγκτής ATmega1280 έχει 128 KB μνήμη flash για την αποθήκευση κώδικα (εκ των οποίων 4 KB χρησιμοποιούνται για τη φόρτωση της εκκίνησης (bootloader), 8 KB μνήμη SRAM και 4 KB μνήμη EEPROM (που μπορεί να διαβαστεί και να γραφτεί με τη βιβλιοθήκη EEPROM).

1.1.4. Είσοδος και Έξοδος

Καθεμία από τις 54 ψηφιακές ακίδες στο Mega μπορούν να χρησιμοποιηθούν ως είσοδος ή έξοδος, χρησιμοποιώντας τις λειτουργίες pinMode(), digitalWrite(), και digitalRead(). Λειτουργούν στα 5V. Κάθε ακίδα μπορεί να προσφέρει ή να λάβει το μέγιστο των 40 mA και έχει μια εσωτερική pull-up αντίσταση (αποσυνδεδεμένη από προεπιλογή) των 20-50 kOhms. Επιπλέον, μερικές ακίδες έχουν εξειδικευμένες λειτουργίες:

§ Ακίδες για σειριακή επικοινωνία: 0, 15, 17, 19 (RX) για λήψη και 1, 14, 16, 18 (TX) για αποστολή TTL σειριακών δεδομένων. Οι ακίδες 0 και 1 συνδέονται επίσης με τους αντίστοιχους ακροδέκτες του σειριακού FTDI τσιπ μετατροπής USB-σε-TTL.

§ Εξωτερικά σήματα διακοπής: Οι ακίδες 2, 3, 18, 19, 20, και 21 μπορούν να ρυθμιστούν για να προκαλέσουν διακοπή όταν η τιμή της ακίδας αλλάξει ή είναι χαμηλή (low), ή και όταν η κατάσταση της ακίδας πάει από low σε high ή από high σε low.

§ PWM: Είναι οι ακίδες από 0 έως 13. Παρέχει 8-bit εξόδου PWM με την συνάρτηση `analogWrite()`.

§ SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Αυτές οι ακίδες υποστηρίζουν την επικοινωνία SPI.

§ LED: 13. Υπάρχει ένα ενσωματωμένο LED που συνδέεται με την ψηφιακή ακίδα 13. Όταν η ακίδα είναι σε υψηλά βολτ, το LED ανάβει, όταν η ακίδα είναι σε χαμηλά, το LED δεν ανάβει.

§ I2C: 20 (SDA) και 21 (SCL). Υποστηρίζει την επικοινωνία I2C (TWI) με τη βιβλιοθήκη `Wire` (τεκμηρίωση στην ιστοσελίδα της Wiringwww.wiring.org.co).

§ Το Mega έχει 16 αναλογικές εισόδους, καθεμία από τις οποίες παρέχει ανάλυση 10 bits (δηλαδή 1024 διαφορετικές τιμές).

§ AREF: Τάση αναφοράς για τις αναλογικές εισόδους (0 έως 5 V). Χρησιμοποιείται με την συνάρτηση `analogReference()`.

§ Reset: Για την επαναφορά του μικροελεγκτή της πλατφόρμας, έτσι ώστε να ξεκινήσει η εκτέλεση του προγράμματος από την αρχή.

1.1.5 Επικοινωνία

Το `ArduinoMega` έχει μια σειρά από λειτουργίες για την επικοινωνία με έναν υπολογιστή, ένα άλλο `Arduino`, ή άλλους μικροελεγκτές. Ο `ATmega1280` παρέχει τέσσερα UARTs υλικού για TTL (5V) σειριακή επικοινωνία. Ένα `FTDI232RL` πάνω στην πλακέτα διοχετεύει ένα από αυτά μέσω USB και οι FTDI οδηγοί (περιλαμβάνονται με το λογισμικό του `Arduino`) παρέχουν μια εικονική θύρα COM για το λογισμικό στον υπολογιστή. Το λογισμικό `Arduino` περιλαμβάνει μια

σειριακή οθόνη που επιτρέπει την απλή μορφή κειμένου να αποστέλλεται από και προς την πλακέτα του Arduino. Τα RX και TXLEDs της πλακέτας αναβοσβήνουν όταν γίνεται μετάδοση δεδομένων μέσω του τσιπ FTDI και της USB σύνδεσης με τον υπολογιστή (όχι όμως για σειριακή επικοινωνία στις ακίδες 0 και 1).

Ο ATmega1280 υποστηρίζει επίσης την I2C (TWI) και SPI επικοινωνία.

1.1.6 USB Προστασία Διαρροής Ρεύματος

Το ArduinoMega έχει μία πολυμερή ασφάλεια επαναφοράς που προστατεύει τις θύρες USB του υπολογιστή σας από βραχυκύκλωμα και διαρροή ρεύματος. Αν και οι περισσότεροι υπολογιστές παρέχουν την δικιά τους εσωτερική προστασία, η ασφάλεια αυτή παρέχει ένα επιπλέον επίπεδο προστασίας. Εάν περισσότερα από 500 mA εφαρμοστούν στη θύρα USB, η ασφάλεια θα διακόψει αυτόματα τη σύνδεση μέχρι το βραχυκύκλωμα ή η υπερφόρτωση να αφαιρεθεί.

1.1.7 Φυσικά χαρακτηριστικά της πλατφόρμας

Το μέγιστο μήκος και το πλάτος του MegaPCB (PrintedCircuitBoard) είναι 4 και 2,1 ίντσες αντίστοιχα, με την υποδοχή USB και την υποδοχή ρεύματος να επεκτείνονται πέραν της προηγούμενης διάστασης. Τρεις τρύπες για βίδες επιτρέπουν στην πλακέτα να προσκολλάται σε μια επιφάνεια ή θήκη. Η απόσταση μεταξύ των ψηφιακών ακίδων 7 και 8 είναι 160 χιλ. (0,16"), ενώ των άλλων ακίδων είναι 100 χιλ. μεταξύ τους.

1.1.8 Πλεονεκτήματα της πλατφόρμας Arduino

§ Οικονομική: Σε σύγκριση με άλλες πλατφόρμες μικροελεγκτή που

κυκλοφορούν στο εμπόριο είναι σχετικά φθηνή. Και στην περίπτωση που αγοραστεί μη συναρμολογημένη είναι ακόμα πιο οικονομική.

§ Μεταφέρσιμη: Το λογισμικό Arduino τρέχει σε Windows, MacintoshOSX και Linux λειτουργικά συστήματα. Τα περισσότερα συστήματα μικροελεγκτή περιορίζονται στα Windows.

§ Απλό, σαφές περιβάλλον προγραμματισμού: Το περιβάλλον προγραμματισμού του Arduino είναι εύκολο στη χρήση για αρχάριους, αλλά και αρκετά ευέλικτο για προχωρημένους χρήστες. Για τους εκπαιδευτικούς, επειδή είναι βασισμένο πάνω στο περιβάλλον προγραμματισμού του Processing, οι μαθητές που μαθαίνουν να προγραμματίζουν σε αυτό το περιβάλλον θα είναι εξοικειωμένοι με την εμφάνιση και την αίσθηση του Arduino.

§ Επεκτάσιμο υλικό και λογισμικό: Τόσο το λογισμικό όσο και το υλικό είναι ανοικτού κώδικα και μπορούν να επεκταθούν και να βελτιωθούν.

1.2. Προγραμματισμός του Arduino

Ο μικροελεγκτής ATmega1280 στο ArduinoMega έρχεται έτοιμος φορτωμένος με έναν bootloader που επιτρέπει να φορτωθεί νέος κώδικας σε αυτόν χωρίς τη χρήση εξωτερικού προγραμματιστή υλικού. Επικοινωνεί χρησιμοποιώντας το αρχικό πρωτόκολλο STK500 (http://www.atmel.com/dyn/resources/prod_documents/doc2525.pdf). Επίσης μπορεί κανείς να παρακάμψει τον bootloader και το πρόγραμμα του μικροελεγκτή μέσω του ICSP (In-CircuitSerialProgramming).

1.2.1 Ολοκληρωμένο Περιβάλλον Ανάπτυξης του Arduino










Εικόνα 2. Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino

Το περιβάλλον ανάπτυξης Arduino περιέχει ένα πρόγραμμα επεξεργασίας κειμένου για τη συγγραφή κώδικα, μια περιοχή μηνυμάτων, μια κονσόλα κειμένου, μια γραμμή εργαλείων με κουμπιά για κοινές λειτουργίες, καθώς και μια σειρά από μενού. Συνδέεται με το υλικό Arduino για τη φόρτωση προγραμμάτων και για να επικοινωνούν μαζί τους.

Ένα ολοκληρωμένο πρόγραμμα συνήθως ονομάζεται sketch (σχέδιο). Αυτό το sketch είναι γραμμένο με το πρόγραμμα επεξεργασίας κειμένου. Έχει δυνατότητες για την αντιγραφή/επικόλληση και για την αναζήτηση/αντικατάσταση κειμένου. Η περιοχή μηνυμάτων

ανατροφοδοτεί αποθηκεύοντας παράλληλα και εξάγοντας την εμφάνιση των σφαλμάτων. Η κονσόλα απεικονίζει την έξοδο του κειμένου από το περιβάλλον Arduino συμπεριλαμβάνοντας πλήρη μηνύματα λάθους και άλλες πληροφορίες. Τα κουμπιά της γραμμής εργαλείων επιτρέπουν τον έλεγχο και το ανέβασμα των προγραμμάτων, τη δημιουργία, το άνοιγμα και την αποθήκευση των σχεδίων και ανοίγουν τη σειριακή οθόνη:

-  Verify/Compile (Ελέγχο/Μεταγλωττίζω): Ελέγχει για λάθη στον κώδικα.
-  Stop (Σταμάτα): Σταματάει τη σειριακή οθόνη, ή υποτονίζει κάποια κουμπιά.
-  New (Νέο): Δημιουργεί ένα νέο sketch (σχέδιο).
-  Open (Άνοιγμα): Παρουσιάζει ένα μενού με όλα τα sketches (σχέδια) στο sketchbook. Κάνοντας κλικ σε ένα από αυτά θα ανοίξει μέσα στο τρέχον παράθυρο.
-  Save (Αποθήκευση): Αποθηκεύει το sketch (σχέδιο).
-  Upload to I/O Board (Ανέβασμα στην πλακέτα εισόδου/εξόδου): Μεταγλωττίζει τον κώδικα και τον ανεβάζει στην πλακέτα εισόδου/εξόδου του Arduino.
-  SerialMonitor (Σειριακή οθόνη): Ανοίγει την σειριακή οθόνη.

Πρόσθετες εντολές βρίσκονται μέσα στα πέντε μενού: File, Edit, Sketch, Tools, Help. Τα μενού είναι ευαίσθητα, που σημαίνει ότι μόνο εκείνα τα στοιχεία που αφορούν το έργο που διενεργεί εκείνη τη στιγμή είναι διαθέσιμα.

1.2.2 Μενού Edit (Επεξεργασία)

- CopyforForum (Αντιγραφή για το Forum): Αντιγράφει τον κώδικα του sketch (σχεδίου) στο πρόχειρο έτσι ώστε να επικολληθεί μετά στο φόρουμ, με το πλήρες χρωματισμό της σύνταξης.
- CopyasHTML (Αντιγραφή σαν HTML): Αντιγράφει τον κώδικα του sketch (σχεδίου) στο πρόχειρο σαν HTML, έτσι ώστε να είναι κατάλληλο για να ενσωματωθεί σε ιστοσελίδες.

1.2.3. Μενού Sketch(Σχέδιο)

- ImportLibrary (Εισαγωγή Βιβλιοθήκης): Προσθέτει μια βιβλιοθήκη στο sketch (σχέδιο), εισάγοντας #include δηλώσεις στον κώδικα.
- ShowSketchFolder (Εμφάνισε τον φάκελο των σχεδίων): Ανοίγει τον φάκελο των sketch (σχεδίων) στην επιφάνεια εργασίας.

1.2.4. Μενού Tools (Εργαλεία)

- AutoFormat (Αυτόματη Μορφοποίηση): Αυτή η λειτουργία μορφοποιεί τον κώδικα όμορφα, έτσι ώστε να είναι αναγνώσιμος και σωστά δομημένος.
- Board (Πλακέτα): Επιλογή της πλακέτα που χρησιμοποιεί ο καθένας, από μια λίστα που έχει όλες τις πλατφόρμες Arduino του εμπορίου.
- SerialPort (Σειριακή Θύρα): Αυτό το μενού περιέχει όλες τις σειριακές συσκευές (πραγματικές ή εικονικές) του υπολογιστή. Οπότε από εδώ γίνεται η επιλογή της σειριακής θύρας όπου είναι συνδεδεμένη η πλατφόρμα Arduino.
- BurnBootloader (Τοποθέτηση Φορτωτή Εκκίνησης): Η επιλογή

αυτή επιτρέπει την τοποθέτηση ενός φορτωτή εκκίνησης (bootloader) πάνω στον μικροελεγκτή μιας πλακέτας Arduino. Αυτό δεν είναι απαραίτητο για την κανονική χρήση μιας πλακέτας Arduino, αλλά είναι χρήσιμο εάν υπάρχει ένα νέο ATmega (τα οποία κανονικά έρχονται χωρίς φορτωτή εκκίνησης). Πρέπει να είναι βεβαιωμένο ότι έχει επιλεγεί η σωστή πλακέτα από το μενού Board (Πλακέτα) πριν την τοποθέτηση του φορτωτή εκκίνησης.

1.2.5. Sketchbook

Το περιβάλλον Arduino περιλαμβάνει την έννοια ενός sketchbook: ένα πρότυπο χώρο για την αποθήκευση των προγραμμάτων (ή των σχεδίων). Τα sketches (σχέδια) στο sketchbook μπορούν να ανοίξουν από το μενού File>Sketchbook ή από το κουμπί Open (Άνοιγμα) στη γραμμή εργαλείων. Την πρώτη φορά που θα εκτελεστεί το λογισμικό Arduino, θα δημιουργήσει αυτόματα έναν φάκελο για το sketchbook. Η αλλαγή της τοποθεσίας του sketchbook μπορεί να γίνει μέσα από το πλαίσιο διαλόγου Preferences (Προτιμήσεις).

1.2.6 Tabs and Multiple Files (Καρτέλες και Πολλαπλά αρχεία)

Επιτρέπει τη διαχείριση των sketches (σχεδίων) με περισσότερα από ένα αρχεία (καθένα από τα οποία εμφανίζεται σε δική του καρτέλα). Αυτά μπορεί να είναι φυσιολογικά αρχεία κώδικα Arduino (χωρίς επέκταση), C αρχεία (επέκταση .c), C ++ αρχεία (επέκταση .cpp), ή αρχεία κεφαλίδας (επέκταση .h).

1.2.7 Uploading(Ανέβασμα)

Πριν το ανέβασμα του σχεδίου, πρέπει να επιλεγούν τα σωστά στοιχεία από τα μενού Tools>Board και Tools>SerialPort. Οι πλακέτες

περιγράφονται παρακάτω: Σε Mac, η σειριακή θύρα είναι μάλλον κάτι σαν `/dev/tty.usbserial-1B1` (για μία πλακέτα USB), ή `/dev/tty.USA19QW1b1P1.1` (για μία σειριακή πλακέτα). Στα Windows, είναι πιθανώς COM1 ή COM2 (για μία σειριακή πλακέτα) ή COM4, COM5, COM7, ή μεγαλύτερη (για USB πλακέτα). Στο Linux, θα πρέπει να είναι `/dev/ttyUSB0`, `/dev/ttyUSB1` ή κάτι παρόμοιο.

Μόλις επιλεγεί η σωστή σειριακή θύρα και η πλακέτα, μετά μένει να πατηθεί το κουμπί Upload στη γραμμή εργαλείων ή να επιλεγεί το Upload to I/O Board από το μενού File. Οι σύγχρονες πλακέτες Arduino κάνουν επαναφορά αυτόματα και μετά αρχίζει το «ανέβασμα» (uploading). Οι παλιότερες πλακέτες δε διαθέτουν αυτόματη επαναφορά και χρειάζεται να πατηθεί το κουμπί επαναφοράς στην πλακέτα λίγο πριν ξεκινήσει το «ανέβασμα». Στις περισσότερες πλακέτες, τα RX και TX LEDs αναβοσβήνουν, καθώς το sketch (σχέδιο) φορτώνεται. Το περιβάλλον Arduino θα εμφανίσει ένα μήνυμα όταν ολοκληρωθεί το «ανέβασμα», ή θα δείξει ένα σφάλμα (error).

Όταν φορτώνεται ένα sketch (σχέδιο) χρησιμοποιείται ο bootloader (φορτωτής εκκίνησης) του Arduino, ένα μικρό πρόγραμμα που έχει φορτωθεί στον μικροελεγκτή της πλακέτας. Επιτρέπει να φορτωθεί ο κώδικας, χωρίς να χρησιμοποιεί οποιοδήποτε πρόσθετο υλικό (π.χ. μια εξωτερική συσκευή προγραμματισμού). Ο bootloader είναι ενεργός για μερικά δευτερόλεπτα, όταν η πλακέτα κάνει επαναφορά. Τότε ξεκινάει όποιο sketch (σχέδιο) ήταν πιο πρόσφατα φορτωμένο στον μικροελεγκτή. Ο bootloader θα αναβοσβήσει το LED (pin 13) της πλακέτας, όταν ξεκινήσει (δηλαδή όταν η πλακέτα κάνει επαναφορά).

1.2.8 Libraries (Βιβλιοθήκες)

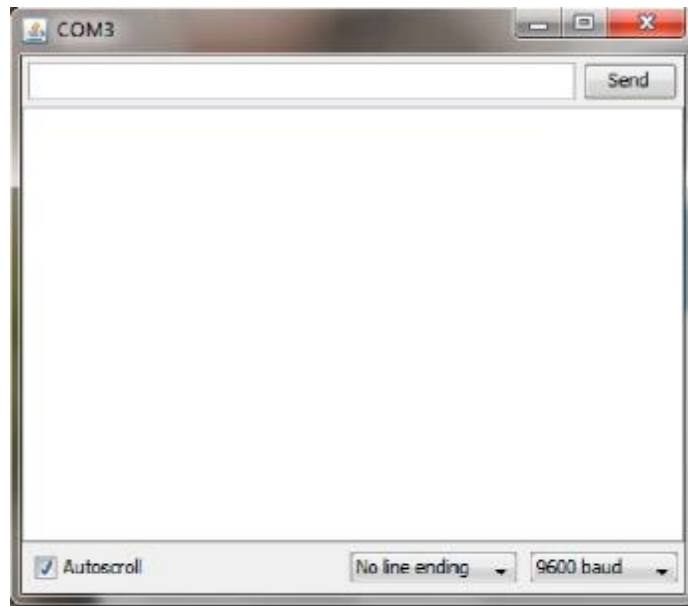
Οι βιβλιοθήκες προσφέρουν επιπλέον λειτουργικότητα στα sketches (σχέδια), π.χ. να εργάζονται με το υλικό ή να χειρίζονται τα δεδομένα. Για

να χρησιμοποιηθεί μια βιβλιοθήκη σε ένα sketch (σχέδιο), μπορεί να επιλεγεί από το μενού Sketch>ImportLibrary και μετά επιλέγετε και το sketch. Αυτό θα εισάγει μία ή περισσότερες #include δηλώσεις στην κορυφή του sketch και θα γίνει μεταγλώττιση της βιβλιοθήκης με το sketch. Επειδή οι βιβλιοθήκες φορτώνονται στην πλακέτα με το sketch, αυξάνουν το μέγεθος του χώρου που καταλαμβάνεται. Εάν ένα sketch δε χρειάζεται πλέον μια βιβλιοθήκη, απλά μπορούν να διαγραφούν οι #include δηλώσεις από την κορυφή του κώδικα.

Στην ιστοσελίδα του Arduino <http://arduino.cc/en/Reference/Libraries> υπάρχει μια λίστα των βιβλιοθηκών (π.χ. EEPROM, Ethernet, Firmata κ.α.). Ορισμένες βιβλιοθήκες συμπεριλαμβάνονται στο λογισμικό του Arduino. Για την εγκατάσταση των βιβλιοθηκών που δεν υπάρχουν ήδη στο λογισμικό, μπορεί να δημιουργηθεί ένας κατάλογος με την ονομασία libraries (βιβλιοθήκες), μέσα στον κατάλογο του sketchbook. Στη συνέχεια αποσυμπιέζεται η βιβλιοθήκη εκεί. Για παράδειγμα, για να εγκατασταθεί η βιβλιοθήκη DateTime, τα αρχεία της θα πρέπει να είναι στο/libraries/DateTime, υποφάκελο του φακέλου sketchbook.

1.2.9 SerialMonitor(Σειριακή Οθόνη)

Εμφανίζει τα σειριακά δεδομένα που αποστέλλονται από την πλακέτα Arduino (USB ή σειριακή πλακέτα). Η αποστολή δεδομένων στην πλακέτα γίνεται, εισάγοντας κείμενο και πατώντας το κουμπί Send (Αποστολή) ή πατώντας το Enter. Επίσης δεξιά στο κάτω μέρος της σειριακής οθόνης μπορεί να γίνει η επιλογή της κατάλληλης ταχύτητας (baud) από την λίστα που εμφανίζεται. Να σημειωθεί ότι σε Mac ή στα Linux, η πλακέτα Arduino θα κάνει επαναφορά (reset) του sketch μόλις συνδεθεί με την σειριακή οθόνη.



Εικόνα 3. Η σειριακή οθόνη του IDE του Arduino

1.2.10. setup() και loop()

Υπάρχουν δύο ειδικές συναρτήσεις που είναι μέρος του κάθε sketch του Arduino: η setup() και η loop(). Η setup() καλείται μια φορά, όταν το sketch ξεκινά ή όποτε κάνει επαναφορά (reset) η πλατφόρμα Arduino. Σε αυτήν γίνονται οι αρχικοποιήσεις των μεταβλητών, η ρύθμιση της κατάστασης των ακίδων (pins), η προετοιμασία των βιβλιοθηκών κ.τ.λ. Η συνάρτηση loop() κάνει αυτό ακριβώς που υποδηλώνει το όνομά της, δηλαδή καλείται ξανά και ξανά, επιτρέποντας έτσι στο πρόγραμμα να αλλάξει και να ανταποκριθεί. Και οι δύο συναρτήσεις πρέπει να περιλαμβάνονται στο sketch, ακόμα και αν δεν περιέχουν κάτι και είναι κενές.

1.2.11. DigitalPins (Ψηφιακές Ακίδες)

Οι ακίδες αυτές στο Arduino μπορούν να ρυθμιστούν είτε ως είσοδοι είτε ως έξοδοι, όμως από προεπιλογή είναι ρυθμισμένες ως είσοδοι. Οι ακίδες (pins) που έχουν ρυθμιστεί ως είσοδοι λέγεται ότι είναι σε μια κατάσταση υψηλής αντίστασης ενώ οι ακίδες που έχουν ρυθμιστεί

ως έξοδοι είναι σε μια κατάσταση χαμηλής αντίστασης. Επίσης αξίζει να σημειωθεί ότι η συντριπτική πλειοψηφία των αναλογικών ακίδων του Arduino (Atmega), μπορεί να ρυθμιστεί, και να χρησιμοποιηθεί, με τον ίδιο ακριβώς τρόπο όπως οι ψηφιακές ακίδες. Οι συναρτήσεις ψηφιακής εισόδου και εξόδου είναι οι παρακάτω:

- `pinMode()`: Ρυθμίζει τη συγκεκριμένη ακίδα να συμπεριφέρεται είτε ως είσοδος είτε ως έξοδος.

Σύνταξη: `pinMode(pin, mode)` Παράμετροι:

`pin`: ο αριθμός της ακίδας της οποίας η λειτουργία είναι επιθυμητό να αλλάξει.

`mode`: είτε INPUT είτε OUTPUT Επιστρέφει: Τίποτα

`value`: HIGH ή LOW Επιστρέφει: Τίποτα

- `digitalRead()`: Διαβάζει την τιμή από μια συγκεκριμένη ψηφιακή ακίδα, που είναι είτε HIGH είτε LOW.

Σύνταξη: `digitalRead(pin)` Παράμετροι:

`pin`: ο αριθμός της ακίδας που είναι επιθυμητό να διαβαστεί (int)

Επιστρέφει: HIGH ή LOW

1.2.12. AnalogInputPins (Αναλογικές Ακίδες Εισόδου)

Οι ελεγκτές Atmega που χρησιμοποιούνται για την πλατφόρμα Arduino περιέχουν έναν ενσωματωμένο αναλογικό-σε-ψηφιακό (A/D) μετατροπέα 6 καναλιών. Ο μετατροπέας διαθέτει ανάλυση 10 bit, επιστρέφοντας ακέραιους από 0 έως 1023. Ενώ η κύρια λειτουργία της αναλογικής ακίδας για τους περισσότερους χρήστες Arduino είναι να διαβάζει αναλογικούς αισθητήρες, οι αναλογικές ακίδες έχουν επίσης όλες τις λειτουργίες των γενικών ακίδων εισόδου/εξόδου (GPIO) (το ίδιο με τις ψηφιακές ακίδες 0-13). Οι συναρτήσεις αναλογικής εισόδου και εξόδου είναι οι παρακάτω:

ακίδες 3, 5, 6, 9, 10 και 11. Στο ArduinoMega, λειτουργεί στις ακίδες 2

μέχρι 13. Παλαιότερες πλακέτες Arduino με ATmega8 υποστήριζαν την `analogWrite()` μόνο στις ακίδες 9, 10 και 11. Σύνταξη: `analogWrite(pin, value)` Παράμετροι:

`pin`: η ακίδα που επάνω θα γράψει.

`value`: ο κύκλος λειτουργίας: μεταξύ 0 (πάντα off) και 255 (πάντα on).

Επιστρέφει: Τίποτα

- `analogRead()`: Διαβάζει την τιμή από την καθορισμένη αναλογική ακίδα.

Σύνταξη: `analogRead(pin)` Παράμετροι:

`pin`: ο αριθμός της αναλογικής ακίδας εισόδου, από όπου θα διαβάσει (0 έως 5 στις περισσότερες πλακέτες, 0 έως 7 για το Mini και το Nano, 0 έως 15 για το Mega) Επιστρέφει: ακέραιο `int` (0 έως 1023)

1.2.13. Pinmapping (Χαρτογράφηση ακίδων)

Οι αναλογικές ακίδες μπορούν να χρησιμοποιηθούν για τον ίδιο σκοπό με τις ψηφιακές ακίδες, χρησιμοποιώντας τα ψευδώνυμα A0 (για αναλογική είσοδο 0), A1, κλπ. Για παράδειγμα, για να ρυθμιστεί η αναλογική ακίδα 0 ως έξοδος, και να τεθεί σε υψηλή (HIGH) τιμή, ο κώδικας θα είναι κάπως έτσι:

```
pinMode(A0, OUTPUT);  
digitalWrite(A0, HIGH);
```

1.2.14 Pullup Αντιστάσεις

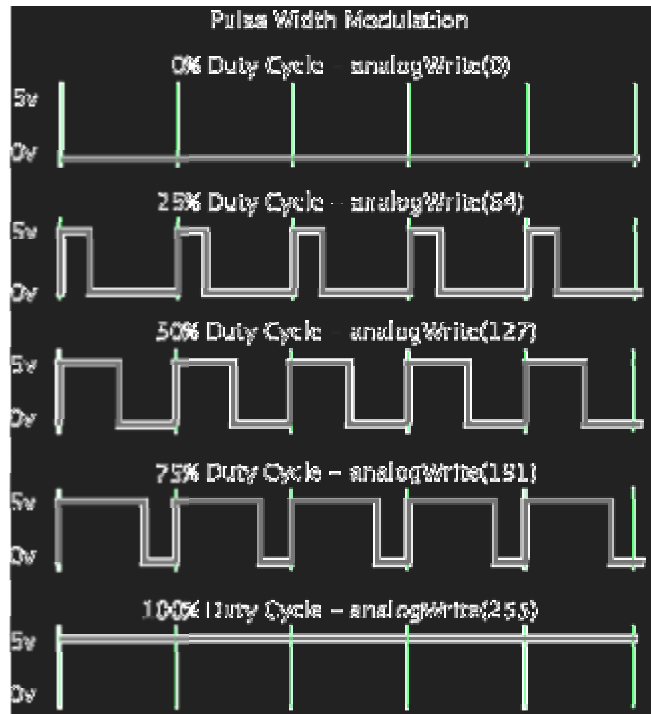
Οι αναλογικές ακίδες έχουν επίσης pullup αντιστάσεις, που λειτουργούν παρόμοια με τις pullup αντιστάσεις στις ψηφιακές ακίδες. Ενεργοποιούνται με εντολές όπως:
`digitalWrite(A0, HIGH);` // θέτει pullup στην αναλογική ακίδα 0, ενώ η ακίδα είναι είσοδος.

1.2.15 PWM (PulseWidthModulation)

Η διαμόρφωση πλάτους παλμών ή PWM (PulseWidthModulation), είναι μια τεχνική για να πάρουμε αναλογικά αποτελέσματα με ψηφιακά μέσα. Χρησιμοποιείται ψηφιακός έλεγχος για να δημιουργηθεί ένα τετραγωνικό κύμα, ένα σήμα που αλλάζει ανάμεσα σε on και off. Αυτό το on-off πρότυπο μπορεί να προσομοιώσει τάσεις μεταξύ 5 Volts (on) και 0 Volt (off), αλλάζοντας το ποσοστό του χρόνου που το σήμα είναι on έναντι του χρόνου που είναι off . Η διάρκεια του «χρόνου» ονομάζεται πλάτος παλμού. Για να πάρουμε διαφορετικές αναλογικές τιμές, μπορούμε να αλλάξουμε ή να ρυθμίσουμε το πλάτος του παλμού. Εάν επαναλάβουμε αυτό το on-off πρότυπο αρκετά γρήγορα, με ένα LED για παράδειγμα, το αποτέλεσμα θα είναι όπως ένα σήμα το οποίο έχει σταθερή τάση μεταξύ 0 και 5V και ελέγχει τη φωτεινότητα του LED.

Το παράδειγμα Fading (Ξεθωριάσμα) δείχνει τη χρήση της αναλογικής εξόδου (PWM) να ξεθωιάζει ένα LED. Είναι διαθέσιμο στο μενού File- >Examples->Analog->Fading του λογισμικού του Arduino, ή στην ιστοσελίδα του Arduino <http://www.arduino.cc/en/Tutorial/Fading>.

Στο παρακάτω γράφημα, οι πράσινες γραμμές αντιπροσωπεύουν μια κανονική χρονική περίοδο. Μια κλήση της analogWrite() είναι σε μια κλίμακα από 0 έως 255, έτσι ώστε για παράδειγμα, η κλήση analogWrite(255) ζητά το 100% του κύκλου λειτουργίας (πάντα on), και η analogWrite(127) το 50% του κύκλου λειτουργίας (στο μισό χρόνο). Οι πληροφορίες αυτές λήφθηκαν από την ιστοσελίδα: <http://arduino.cc/en/Tutorial/PWM>



Εικόνα 4. Διαμόρφωση πλάτους παλμού

1.2.16. Μνήμη

Υπάρχουν τρία κομμάτια της μνήμης του μικροελεγκτή που χρησιμοποιούνται στις πλακέτες Arduino (ATmega1280):

- Μνήμη Flash (ο χώρος του προγράμματος), όπου είναι αποθηκευμένο το sketch (σχέδιο) του Arduino.
- SRAM (στατική μνήμη τυχαίας προσπέλασης), όπου το sketch δημιουργεί και χειρίζεται τις μεταβλητές, καθώς εκτελείται.
- EEPROM είναι χώρος μνήμης, τον οποίο οι προγραμματιστές μπορούν να χρησιμοποιήσουν για την αποθήκευση μακροπρόθεσμων πληροφοριών.

Η μνήμη Flash και η μνήμη EEPROM είναι σταθερές (οι πληροφορίες παραμένουν μετά την απενεργοποίηση του ρεύματος). Η SRAM είναι ασταθής και οι πληροφορίες χάνονται όταν εναλλάσσεται το ρεύμα.

Επειδή δεν υπάρχει πολλή διαθέσιμη SRAM, αν τελειώσει, το

πρόγραμμα μπορεί να αποτύχει με απροσδόκητους τρόπους. Μπορεί να φαίνεται ότι φορτώνει με επιτυχία, αλλά δεν τρέχει, ή τρέχει παράξενα. Για να ελεγχθεί εάν αυτό συμβαίνει, μπορούν να μειωθούν τα σχόλια ή οι σειρές ή άλλες δομές δεδομένων στο sketch (χωρίς να αλλάξει ο κώδικας). Εάν λειτουργεί με επιτυχία στη συνέχεια, κατά πάσα πιθανότητα έχει εξαντληθεί η SRAM. Ένας τρόπος για να αντιμετωπιστεί αυτό το πρόβλημα είναι:

Αν υπάρχουν πίνακες αναζήτησης ή άλλοι μεγάλοι πίνακες, τότε μπορεί να χρησιμοποιηθεί ο μικρότερος τύπος δεδομένων που είναι αναγκαίος για να αποθηκευτούν οι τιμές που χρειάζονται. Για παράδειγμα, ένας ακέραιος (int) καταλαμβάνει δύο bytes, καθώς το ένα byte χρησιμοποιεί μόνο έναν ακέραιο (αλλά μπορεί να αποθηκεύσει μικρότερο εύρος τιμών).

1.2.17. Σειριακή Θύρα

Χρησιμοποιείται για επικοινωνία μεταξύ της πλακέτας Arduino και ενός υπολογιστή ή με άλλες συσκευές. λες οι πλακέτες Arduino έχουν τουλάχιστον μία σειριακή θύρα (επίσης γνωστή ως UART ή USART): Serial. Επικοινωνεί με τις ψηφιακές ακίδες 0 (RX) και 1 (TX), καθώς και με τον υπολογιστή μέσω USB. Έτσι, εάν χρησιμοποιούνται αυτές οι λειτουργίες, δεν μπορούν ταυτόχρονα να χρησιμοποιηθούν οι ακίδες 0 και 1 για ψηφιακή είσοδο ή έξοδο.

Η ενσωματωμένη σειριακή οθόνη στο περιβάλλον του Arduino μπορεί να χρησιμοποιηθεί για να επικοινωνεί με την πλακέτα Arduino. Κάνοντας κλικ στο κουμπί της σειριακής οθόνης στη γραμμή εργαλείων και επιλέγοντας την ίδια ταχύτητα που χρησιμοποιείται στην κλήση της begin().

Το Arduino Mega έχει τρεις επιπλέον σειριακές θύρες: Serial1 στις ακίδες 19 (RX) και 18 (TX), Serial2 στις ακίδες 17 (RX) και 16 (TX),

Serial3 στις ακίδες 15 (RX) και 14 (TX). Για να χρησιμοποιηθούν αυτές οι ακίδες, ώστε να επικοινωνούν με τον προσωπικό υπολογιστή, θα χρειαστεί ένα επιπλέον USB που να βγάζει σε σειριακό προσαρμογέα, δεδομένου ότι δεν είναι συνδεδεμένες με το USB-σε-σειριακό προσαρμογέα του Mega. Για τη χρήση αυτών των ακίδων, ώστε να επικοινωνούν με εξωτερική TTL σειριακή συσκευή, συνδέουμε την ακίδα TX στην ακίδα RX της συσκευής, την RX στην TX ακίδα της συσκευής, και την γείωση του Mega με την γείωση της συσκευής. (Δε συνδέουμε αυτές τις ακίδες κατευθείαν σε μία RS232 σειριακή θύρα, γιατί λειτουργούν σε +/- 12V και μπορούν να βλάψουν την πλακέτα του Arduino). Αναφορικά οι βασικές συναρτήσεις της σειριακής θύρα 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 τα 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

Για την συγγραφή αυτών των πληροφοριών πολύτιμες φάνηκαν οι ιστοσελίδες:

- <http://www.arduino.cc/>
- <http://www.wikipedia.org/>

ΚΕΦΑΛΑΙΟ 2

ΛΑΜΒΑΝΟΝΤΑΣ ΕΙΣΟΔΟ ΑΠΟ ΑΙΣΘΗΤΗΡΕΣ

2.1 Εισαγωγή

Λαμβάνοντας και χρησιμοποιώντας δεδομένα από αισθητήρες επιτρέπει στο Arduino να απαντήσει ή να υποβάλει αναφορά σχετικά με τον κόσμο γύρω του. Αυτή είναι μία από τις πιο συνηθισμένες εργασίες που θα συναντήσετε. Αυτό το κεφάλαιο παρέχει απλά και πρακτικά παραδείγματα για το πώς να χρησιμοποιούν οι πιο δημοφιλείς συσκευές και αισθητήρες εισόδου. Διαγράμματα καλωδίωσης δείχνουν τον τρόπο για να συνδέσετε και να τροφοδοτήσετε τις συσκευές, και προβάλλουν παραδείγματα που δείχνουν πώς να χρησιμοποιείτε τα δεδομένα που προέρχονται από τους αισθητήρες.

Οι αισθητήρες ανταποκρίνονται στην εισαγωγή από τον φυσικό κόσμο και τη μετατρέπουν σε ένα ηλεκτρικό σήμα που το Arduino μπορεί να διαβάσει σε ένα pin εισόδου. Η φύση του ηλεκτρικού σήματος που παρέχεται από έναν αισθητήρα εξαρτάται από το είδος του αισθητήρα και πόσες πληροφορίες χρειάζεται να μεταδώσει. Μερικοί αισθητήρες (όπως οι φωτοαντιστάσεις και οι αισθητήρες knockPiezo) είναι κατασκευασμένοι από μια ουσία που μεταβάλλει τις ηλεκτρικές τους ιδιότητες ανάλογα με κάποια φυσική αλλαγή. Άλλοι είναι εξελιγμένα ηλεκτρονικά modules, που χρησιμοποιούν το δικό τους μικροελεγκτή για την επεξεργασία πληροφοριών πριν περάσουν ένα σήμα για το Arduino.

Οι αισθητήρες χρησιμοποιούν τις ακόλουθες μεθόδους για την παροχή πληροφοριών:

2.1.1 Ψηφιακή on / off

Ορισμένες συσκευές , όπως ο αισθητήρας κλίσης στην Οδηγία 6.1 και ο αισθητήρας κίνησης στην Οδηγία 6.3, απλά ανάβουν μια τάση και τη σβήνουν.

2.1.2 Αναλογική

Άλλοι αισθητήρες παρέχουν ένα αναλογικό σήμα (μία τάση που είναι ανάλογη με ό,τι ανιχνεύεται, όπως η θερμοκρασία ή το επίπεδο φωτός). Οι οδηγίες για την ανίχνευση του φωτός (Οδηγία 6.2), κίνησης (Οδηγίες 6.1 και 6.3), κραδασμών (Οδηγία 6.6), ήχου (Οδηγία 6.7) , και επιτάχυνσης (Οδηγία 6.18) δείχνουν πώς οι αναλογικοί αισθητήρες μπορούν να χρησιμοποιηθούν. Όλοι τους χρησιμοποιούν την εντολή analogRead.

2.1.3 Εύρος παλμού

Οι αισθητήρες αποστάσεως , όπως οι PING στην Οδηγία 6.4, παρέχουν δεδομένα χρησιμοποιώντας διάρκεια παλμού ανάλογη προς την τιμή αποστάσεως. Αυτοί οι αισθητήρες μετρούν τη διάρκεια ενός παλμού χρησιμοποιώντας την εντολή pulseIn.

2.1.4 Σειριακή

Μερικοί αισθητήρες παρέχουν τιμές χρησιμοποιώντας ένα σειριακό πρωτόκολλο. Για παράδειγμα, ο αναγνώστης RFID στην Οδηγία 6.9 και το GPS στο Συνταγή 6.14 επικοινωνούν μέσω της σειριακής θύρας του Arduino. Οι περισσότεροι πίνακες Arduino έχουν μόνο μία σειριακή θύρα hardware, οπότε διαβάστε την Οδηγία 6.14 για ένα παράδειγμα για το πώς μπορείτε να προσθέσετε επιπλέον σειριακές θύρες λογισμικού, αν έχετε πολλαπλούς σειριακούς αισθητήρες ή η σειριακή θύρα του hardware καταλαμβάνεται για κάποιο άλλο έργο.

2.1.5 Σύγχρονα πρωτόκολλα: I2C και SPI

Τα I2C και SPI ψηφιακά πρότυπα δημιουργήθηκαν για μικροελεγκτές όπως το Arduino για να μιλάνε με εξωτερικούς αισθητήρες και μονάδες. Η Οδηγία 6.16 δείχνει πώς μια μονάδα πυξίδας συνδέεται με τη χρήση σύγχρονης ψηφιακής σήμανσης. Αυτά τα πρωτόκολλα χρησιμοποιούνται εκτενώς για αισθητήρες, ενεργοποιητές και περιφερειακά.

Υπάρχει και μια άλλη γενική κατηγορία των συσκευών ανίχνευσης που μπορείς να κάνεις χρήση. Αυτές είναι καταναλωτικές συσκευές που περιέχουν αισθητήρες, αλλά πωλούνται ως συσκευές με δικά τους δικαιώματα και όχι από ό,τι ως αισθητήρες. Παραδείγματα αυτών στο κεφάλαιο αυτό περιλαμβάνουν ένα ποντίκι PS2 και έναν ελεγκτή παιχνιδιών PlayStation. Αυτές οι συσκευές μπορεί να είναι πολύ χρήσιμες, παρέχουν αισθητήρες που έχουν ήδη ενσωματωθεί σε ισχυρές και εργονομικές συσκευές. Επίσης, είναι φθηνές (συνήθως λιγότερο ακριβές από την αγορά των αισθητήρων που περιέχουν), δεδομένου ότι είναι μαζικής παραγωγής.

Μπορεί να έχετε μερικές από αυτές γύρω σας. Εάν χρησιμοποιείτε μια συσκευή που δεν καλύπτεται ειδικά σε μια οδηγία, μπορεί να είστε σε θέση να προσαρμόσετε μια συνταγή για μια συσκευή που παράγει ένα παρόμοιο είδος παροχής. Πληροφορίες σχετικά με το σήμα εξόδου ενός αισθητήρα είναι συνήθως διαθέσιμες από την εταιρεία από την οποία αγοράσατε τη συσκευή ή από ένα φύλλο δεδομένων για τη συσκευή σας (το οποίο μπορείτε να βρείτε μέσω αναζήτησης στο Google με τον κωδικό ανταλλακτικού ή την περιγραφή).

Τα φύλλα δεδομένων απευθύνονται σε μηχανικούς σχεδιασμού προϊόντων που πρόκειται να κατασκευαστούν και συνήθως παρέχουν περισσότερες λεπτομέρειες από όσες χρειάζεστε για να “στήσετε” το προϊόν και να λειτουργήσει. Οι πληροφορίες σχετικά με το σήμα εξόδου

θα είναι συνήθως σε ένα κεφάλαιο που αναφέρεται στην μορφή των δεδομένων, διασύνδεση, σήμα εξόδου ή κάτι παρόμοιο. Μην ξεχάσετε να ελέγξετε τη μέγιστη τάση (συνήθως σε μια ενότητα με τίτλο "AbsoluteMaximumRatings"), για να εξασφαλίσετε ότι δε θα προκαλέσετε στο εξάρτημα.

Οι αισθητήρες που έχουν σχεδιαστεί για ένα μέγιστο 3,3 V μπορεί να καταστραφούν από τη σύνδεσή τους σε 5 V. Ελέγξτε την απόλυτη μέγιστη τιμή για τη συσκευή σας, πριν τη σύνδεση.

Οι αισθητήρες ανάγνωσης από τον ακατάστατο αναλογικό κόσμο είναι ένα μείγμα επιστήμης, τέχνης και επιμονής.

Μπορεί να χρειαστεί να χρησιμοποιήσετε εφευρετικότητα και τη δοκιμή και το λάθος για να πάρετε ένα επιτυχές αποτέλεσμα. Ένα κοινό πρόβλημα είναι ότι ο αισθητήρας σας λέει μόνο ότι μια φυσική κατάσταση παρουσιάστηκε, όχι αυτό που το προκάλεσε. Βάζοντας τον αισθητήρα στο σωστό πλαίσιο (τοποθεσία, εύρος, προσανατολισμό) και περιορίζοντας την έκθεσή του σε πράγματα που δεν θέλετε να το δραστηριοποιήσετε, είναι δεξιότητες που θα αποκτήσετε με την εμπειρία.

Ένα άλλο ζήτημα αφορά τον διαχωρισμό του επιθυμητού σήματος από το θόρυβο του περιβάλλοντος. Η Οδηγία 6.6 δείχνει πώς μπορείτε να χρησιμοποιήσετε ένα κατώτατο όριο για να ανιχνεύσετε πότε ένα σήμα είναι πάνω από ένα ορισμένο επίπεδο και η Οδηγία 6.7 δείχνει πώς μπορείτε να πάρετε το μέσο όρο του αριθμού αναγνώσεων για να εξομαλύνετε τις αιχμές του θορύβου .

2.2. Ανίχνευση Φωτός

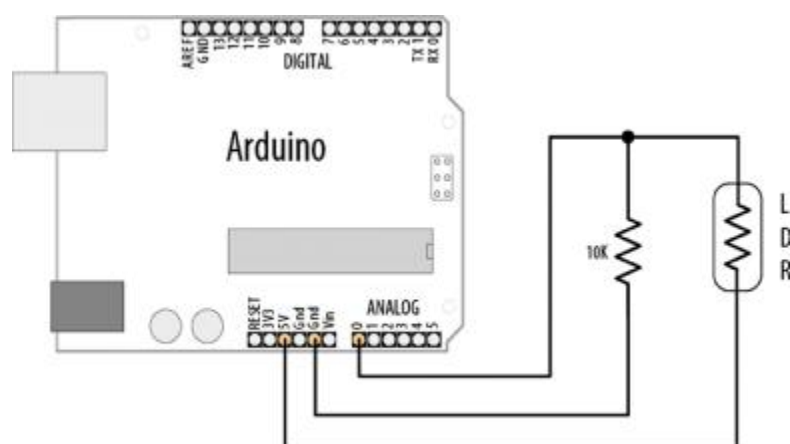
2.2.1 Πρόβλημα

Θέλετε να ανιχνεύσετε τις αλλαγές στα επίπεδα του φωτός. Μπορεί να θέλετε να ανιχνεύσετε μια αλλαγή, όταν κάτι περνάει μπροστά από

έναν ανιχνευτή φωτός ή να μετρήσετε το επίπεδο φωτός - για παράδειγμα, ανιχνεύοντας τότε ένα δωμάτιο είναι πάρα πολύ σκοτεινό.

2.2.2 Λύση

Ο ευκολότερος τρόπος για την ανίχνευση των επιπέδων φωτός είναι να χρησιμοποιήσετε μία ελαφριά αντίσταση (LDR). Αυτό αλλάζει την αντίσταση με την αλλαγή των επιπέδων φωτός και όταν συνδέεται στο κύκλωμα που φαίνεται στο Σχήμα 6-2 παράγει μια αλλαγή στην τάση, που οι ακίδες αναλογικής εισόδου του μπορούν να αισθανθούν.



Σχήμα 6-2 . Συνδέοντας μία ελαφριά αντίσταση

2.2.3 Συζήτηση

Το κύκλωμα για αυτή την οδηγία είναι ο συνηθής τρόπος για να χρησιμοποιήσετε τον αισθητήρα που αλλάζει την αντίστασή του βασισμένος σε κάποιο φυσικό φαινόμενο (βλ. κεφάλαιο 5 για τις βασικές πληροφορίες στην ανταπόκριση σε αναλογικά σήματα). Το κύκλωμα στην Εικόνα 6-2 θα αλλάξει την τάση στο αναλογικό pin 0 , όταν η αντίσταση των LDR αλλάξει με διαφορετικά επίπεδα φωτός.

Ένα κύκλωμα όπως αυτό δεν θα δώσει το πλήρες εύρος των πιθανών τιμών από την αναλογική είσοδο - 0 έως 1,023 - καθώς η τάση δεν θα πρέπει να αιωρείται από 0 V έως 5 V. Αυτό είναι γιατί πάντα θα υπάρχει

μια πτώση τάσης στα άκρα κάθε αντίστασης , έτσι ώστε η τάση , όπου θα συναντηθούν, ποτέ δεν θα φτάσει τα όρια του τροφοδοτικού. Κατά τη χρήση αισθητήρων, όπως αυτοί, είναι σημαντικό να ελέγχετε τις πραγματικές τιμές που η συσκευή επιστρέφει στην κατάσταση που θα τη χρησιμοποιήσετε. Στη συνέχεια, θα πρέπει να καθορίσετε τον τρόπο για τη μετατροπή τους σε τιμές που πρέπει να ελέγχουν ό, τι πρόκειται να ελεγχθεί. Δείτε την Οδηγία 5.7 για περισσότερες λεπτομέρειες σχετικά με την αλλαγή του εύρους τιμών .

Ο LDR είναι ένα απλό είδος αισθητήρα που ονομάζεται αντίσταση του αισθητήρα. Μια σειρά από αισθητήρες αντίστασης ανταποκρίνονται στις αλλαγές των διαφορετικών φυσικών χαρακτηριστικών. Το ίδιο κύκλωμα θα λειτουργήσει για κάθε είδους απλό αισθητήρα αντίστασης .

2.3 Ανίχνευση κίνησης (Ενσωμάτωση Ανιχνευτών Παθητικών Υπερύθρων)

2.3.1 Πρόβλημα

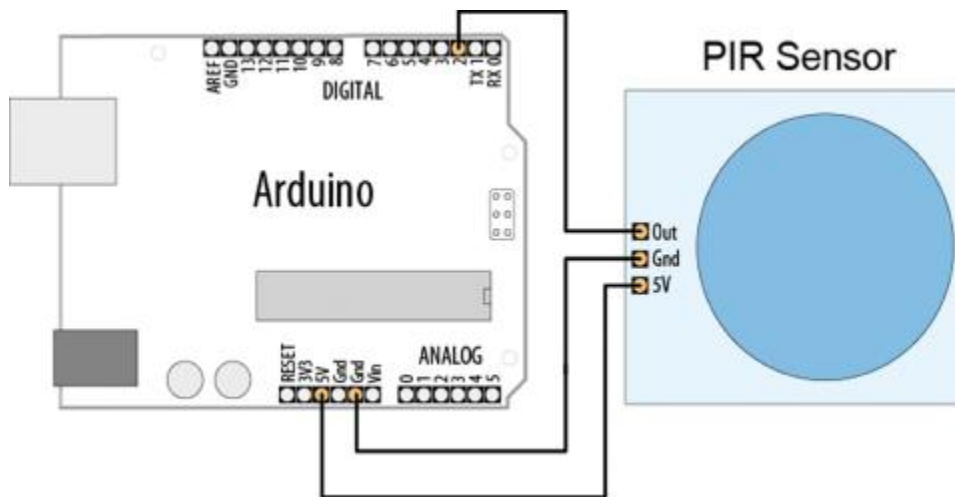
Θέλετε να ανιχνεύσει όταν οι άνθρωποι κινούνται κοντά σε έναν αισθητήρα.

2.3.2 Λύση

Χρησιμοποιήστε ένα αισθητήρα κίνησης , όπως ένα αισθητήρα Παθητικών Υπερύθρων (PIR), για να αλλάξετε τις τιμές σε μια ψηφιακή ακίδα, όταν κάποιος κινείται στο χώρο.

Αισθητήρες, όπως οι Motion Sensor SparkFun PIR (SEN - 08630) και οι Parallax PIR

Αισθητήρες (555-28027), μπορούν εύκολα να συνδεθούν με ακίδες Arduino, όπως φαίνεται στο Σχήμα 6-3.



Σχήμα 6-3 . Σύνδεση με αισθητήρα κίνησης PIR

Ελέγξτε το φύλλο δεδομένων για τον αισθητήρα για να προσδιορίσετε τις σωστές ακίδες. Ο αισθητήρας Parallax έχει ακίδες με την ένδειξη " OUT" , " - " και " + " (για την Έξοδο, Gnd και +5 V). Ο SparkFun αισθητήρας, με την ένδειξη " Alarm" , " GND " , και "DC" (για την Έξοδο, Gnd και +5 V).

Το ακόλουθο σχεδιάγραμμα θα ανάψει το LED στο ακίδα 13 του Arduino, όταν ο αισθητήρας ανιχνεύσει κίνηση:

```

/*
PIR sketch
a Passive Infrared motion sensor connected to pin 2
lights the LED on pin 13
*/
constintledPin = 13;           // choose the pin for the LED
constintinputPin = 2;         // choose the input pin (for the PIR
sensor)
void setup() {
pinMode(ledPin, OUTPUT);     // declare LED as output
pinMode(inputPin, INPUT);    // declare pushbutton as

```

```

input
  }
  void loop(){
    int val = digitalRead(inputPin); // read input value
    if (val == HIGH) // check if the input is HIGH
    {
      digitalWrite(ledPin, HIGH); // turn LED on if motion
detected
      delay(500);
      digitalWrite(ledPin, LOW); // turn LED off

```

2.3.3 Συζήτηση

Αυτός ο κωδικός είναι παρόμοιος με τα παραδείγματα μπουτόν που φαίνονται στο Κεφάλαιο 5. Αυτό συμβαίνει γιατί ο αισθητήρας λειτουργεί σαν διακόπτης, όταν ανιχνεύεται κίνηση. Διαφορετικά είδη αισθητήρων PIR είναι διαθέσιμα και θα πρέπει να ελέγξετε τις πληροφορίες για αυτό που έχετε συνδέσει.

Μερικοί αισθητήρες, όπως ο Parallax, έχουν μια ταμπέλα που καθορίζει τον τρόπο που συμπεριφέρεται η έξοδος, όταν ανιχνεύεται κίνηση. Κατά ένα τρόπο, η έξοδος παραμένει HIGH ενώ ανιχνεύεται κίνηση ή μπορεί να ρυθμιστεί έτσι ώστε η έξοδος να πηγαίνει στο HIGH για λίγο και στη συνέχεια στο LOW, όταν ενεργοποιείται. Το παράδειγμα με το σκίτσο στη λύση της οδηγίας θα λειτουργήσει και στις δυο περιπτώσεις.

Άλλοι αισθητήρες μπορεί να πάνε στο LOW ανιχνεύοντας κίνηση. Αν η ακίδα εξόδου του αισθητήρα σας πηγαίνει στο LOW όταν ανιχνευθεί κίνηση, αλλάξτε τη γραμμή που ελέγχει την τιμή εισόδου, έτσι ώστε το LED να ανάβει όταν είναι στο LOW :

```

if ( val == LOW ) // κίνησης όταν η είσοδος είναι στο

```

LOW

Οι αισθητήρες PIR βρίσκονται σε μια ποικιλία από στυλ και είναι ευαίσθητοι σε διαφορετικές αποστάσεις και γωνίες. Η προσεκτική επιλογή και τοποθέτηση μπορεί να τους κάνει να ανταποκριθούν σε κίνηση σε μέρος του δωματίου, και όχι σε όλο.

2.4 Μέτρηση Απόστασης

2.4.1 Πρόβλημα

Θέλετε να μετρήσετε την απόσταση σε κάτι , όπως ένας τοίχος ή κάποιος που κατευθύνεται προς το Arduino .

2.4.2 Λύση

Αυτή η οδηγία χρησιμοποιεί το δημοφιλή αισθητήρα υπερήχων απόστασης Parallax PING για να μετρήσει την απόσταση ενός αντικειμένου που κυμαίνεται από 2 cm έως περίπου 3 m. Εμφανίζει την απόσταση σε σειριακή οθόνη και αναβοσβήνει ένα LED γρηγορότερα καθώς τα αντικείμενα πλησιάζουν. (Το Σχήμα 6-4 δείχνει τις συνδέσεις):

```
/* Ping))) Sensor
 * prints distance and changes LED flash rate
 * depending on distance from the Ping))) sensor
 */
const int pingPin = 5;
const int ledPin = 13; // pin connected to LED

void setup()
{
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}
```

```

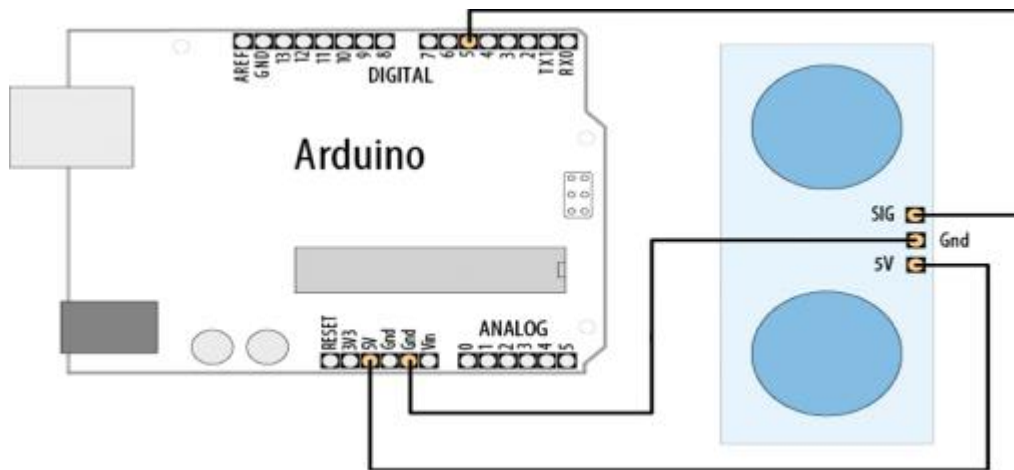
void loop()
{
int cm = ping(pingPin) ;
Serial.println(cm);
digitalWrite(ledPin, HIGH);
delay(cm * 10 ); // each centimeter adds 10 milliseconds delay
digitalWrite(ledPin, LOW);
delay( cm * 10);
}
// following code based on http://www.arduino.cc/en/Tutorial/Ping
// returns the distance in cm
int ping(intpingPin)
{
// establish variables for duration of the ping,
// and the distance result in inches and centimeters:
long duration, cm;
// The PING))) is triggered by a HIGH pulse of 2 or more
microseconds.
// Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
pinMode(pingPin, OUTPUT);
digitalWrite(pingPin, LOW);
delayMicroseconds(2);
digitalWrite(pingPin, HIGH);
delayMicroseconds(5);
digitalWrite(pingPin, LOW);
pinMode(pingPin, INPUT);
duration = pulseIn(pingPin, HIGH);
// convert the time into a distance
cm = microsecondsToCentimeters(duration);

```

```

return cm ;
}
long microsecondsToCentimeters(long microseconds)
{
// The speed of sound is 340 m/s or 29 microseconds per centimeter.
// The ping travels out and back, so to find the distance of the
// object we take half of the distance travelled.
return microseconds / 29 / 2;
}

```



Σχήμα 6-4 . Ping)) συνδέσεις αισθητήρων

2.4.3 Συζήτηση

Οι αισθητήρες υπερήχων παρέχουν μια μέτρηση του χρόνου που χρειάζεται ο ήχος για να αναπηδήσει ένα αντικείμενο και να επιστρέψει στον αισθητήρα.

Ο παλμικός ήχος “σφύριγμα” παράγεται όταν το επίπεδο ringPin πηγαίνει στο HIGH για δύο μικροδευτερόλεπτα.

Στη συνέχεια, ο αισθητήρας θα δημιουργήσει έναν παλμό που τερματίζει όταν ο ήχος επιστρέψει.

Το εύρος του παλμού είναι ανάλογο της απόστασης που διήνυσε ο ήχος και το σκίτσο στη συνέχεια, χρησιμοποιεί τη συνάρτηση pulseIn για

τη μέτρηση αυτής της διάρκειας. Η ταχύτητα του ήχου είναι 340 μέτρα ανά δευτερόλεπτο, το οποίο είναι 29 μικροδευτερόλεπτα ανά εκατοστό. Ο τύπος για την απόσταση μετ' επιστροφής είναι: Μετ' επιστροφής = μικροδευτερόλεπτα / 29

Έτσι, ο τύπος για την απόσταση μίας κατεύθυνσης σε εκατοστά είναι : μικροδευτερόλεπτα / 29/2.

Ο MaxBotix EZ1 είναι ένας άλλος αισθητήρας υπερήχων που μπορεί να χρησιμοποιηθεί για τη μέτρηση της απόστασης.

Είναι ευκολότερο να ενσωματωθεί σε σχέση με το Ping))) , διότι δε χρειάζεται να "pinged". Μπορεί να παρέχει συνεχή ενημέρωση απόστασης, είτε ως αναλογική τάση ή ανάλογη με το πλάτος παλμού. Το Σχήμα 6-5 δείχνει τις συνδέσεις.

Σχήμα 6-5 . Συνδέοντας έξοδο EZ1 PW σε ψηφιακή ακίδα εισόδου

Το σκίτσο που ακολουθεί χρησιμοποιεί τον EZ1 παλμό εξόδου για την παραγωγή εξόδου παρόμοια με του προηγούμενου σκίτσου:

```
/*  
 * EZ1Rangefinder Distance Sensor  
 * prints distance and changes LED flash rate  
 * depending on distance from the Ping))) sensor  
 */  
const int sensorPin = 5;  
const int ledPin = 13; // pin connected to LED  
long value = 0;  
int cm = 0;  
int inches = 0;
```

```

void setup()
{
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);
}
void loop()
{
  value = pulseIn(sensorPin, HIGH) ;
  cm = value / 58; // pulse width is 58 microseconds per cm
  inches = value / 147; // which is 147 microseconds per inch
  Serial.print(cm);
  Serial.print(',');
  Serial.println(inches);
  digitalWrite(ledPin, HIGH);
  delay(cm * 10 ); // each centimeter adds 10 milliseconds delay
  digitalWrite(ledPin, LOW);
  delay( cm * 10);
  delay(20);
}

```

Ο EZ1 τροφοδοτείται από +5 V και ακίδες εδάφους και αυτά είναι συνδεδεμένα με τις αντίστοιχες ακίδες του Arduino . Συνδέστε την ακίδα EZ1 PW (PW είναι η έξοδος Πλάτος Παλμού) με την ψηφιακή ακίδα 5 του Arduino. Το σκίτσο μετρά το πλάτος του παλμού με την εντολή pulseIn. Το πλάτος του παλμού είναι 58 μικροδευτερόλεπτα ανά εκατοστό ή 147 μικροδευτερόλεπτα ανά ίντσα.

Μπορείτε επίσης να λάβετε μια ένδειξη της απόστασης από το EZ1 μέσω της αναλογικής του εξόδου - συνδέστε την ακίδα AN σε μια αναλογική είσοδο και διαβάστε την τιμή με analogRead.

Ο παρακάτω κωδικός τυπώνει την αναλογική είσοδο μετατρεπόμενη

σε ίντσες:

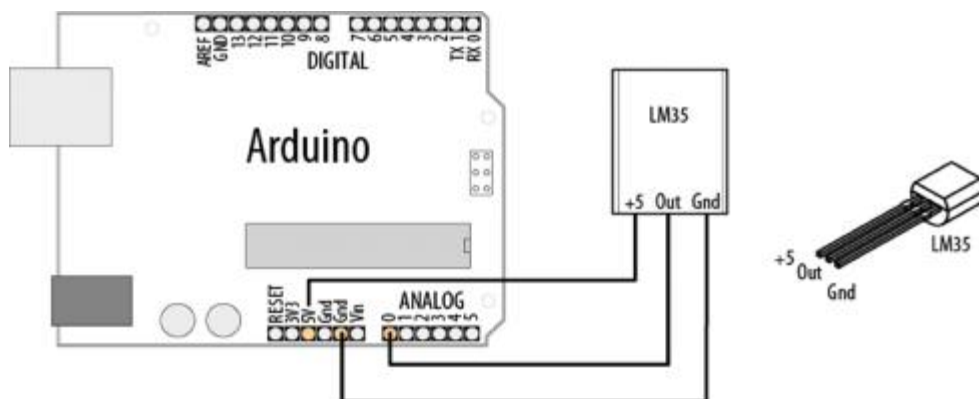
```
value = analogRead(0);  
inches = value / 2; // each digit of analog read is around 5mv  
Serial.println(inches);
```

Η αναλογική έξοδος είναι περίπου 9,8 mV ανά ίντσα. Η τιμή από analogRead είναι περίπου 4,8 mV ανά μονάδα (βλέπε Οδηγία 5.6 για περισσότερα σχετικά με analogRead) και ο προηγούμενος κωδικός γύρω από αυτά, έτσι ώστε κάθε ομάδα των δύο μονάδων είναι μία ίντσα. Το σφάλμα στρογγυλοποίησης είναι μικρό σε σύγκριση με την ακρίβεια της συσκευής, αλλά αν θέλετε πιο ακριβή υπολογισμό, μπορείτε να χρησιμοποιήσετε κινητή υποδιαστολή, ως εξής:

```
value = analogRead(0);  
float mv = (value /1024.0) * 5000 ;  
float Inches = mv / 9.8; // 9.8mv per inch  
Serial.println(Inches) ;
```

2.4.4 Δείτε επίσης

Η Οδηγία 5.6 εξηγεί πώς να μετατρέψετε μετρήσεις από analogInput σε τιμές τάσης. Η αναφορά Arduino για pulseIn: <http://www.arduino.cc/en/Reference/PulseIn>.



Σχήμα 6-10 . Το ηχητικό σήμα που δείχνει DC offset (μέσω σήμα)

2.5 Μέτρηση της θερμοκρασίας

2.5.1 Πρόβλημα

Θέλετε να εμφανίσετε τη θερμοκρασία ή να χρησιμοποιήσετε την τιμή για να ελέγξετε μια συσκευή. Για παράδειγμα, να ανάψετε κάτι, όταν η θερμοκρασία φτάσει σε ένα όριο.

2.5.2 Λύση

Αυτή η οδηγία δείχνει τη θερμοκρασία σε βαθμούς Φαρενάιτ και Κελσίου (Κελσίου) χρησιμοποιώντας το δημοφιλή LM35 αισθητήρα ανίχνευσης θερμότητας. Ο αισθητήρας μοιάζει με ένα τρανζίστορ και είναι συνδεδεμένος, όπως φαίνεται στο Σχήμα 6-11:

```
/*  
lm35 sketch  
prints the temperature to the Serial Monitor  
*/  
const inPin = 0; // analog pin  
void setup()  
{  
  Serial.begin(9600);  
}  
void loop()  
{  
  int value = analogRead(inPin);  
  Serial.print(value); Serial.print(" > ");  
  float millivolts = (value / 1024.0) * 5000;  
  float celsius = millivolts / 10; // sensor output is 10mV per degree  
  Celsius  
  Serial.print(celsius);
```


output pin

```
void setup()
{
  Serial.begin(9600);
  pinMode(outPin, OUTPUT);
}
void loop()
{
  int value = analogRead(inPin);
  longcelsius = (value * 500L) / 1024; // 10 mV per degree c, see text
  Serial.print(celsius);
  Serial.print(" degrees Celsius: ");
  if(celsius > threshold)
  {
    digitalWrite(outPin, HIGH);
    Serial.println("pin is on");
  }
  else
  {
    digitalWrite(outPin, LOW);
    Serial.println("pin is off");
  }
  delay(1000); // wait for one second
}
```

Το σκίτσο χρησιμοποιεί μεγάλους (32 - bit) ακέραιους αριθμούς για τον υπολογισμό της αξίας. Το γράμμα L μετά τον αριθμό κάνει τον υπολογισμό να πραγματοποιείται με τη χρήση ακέραιων μαθηματικών αριθμών, έτσι ώστε ο πολλαπλασιασμός της μέγιστης θερμοκρασίας (500 σε ένα 5V Arduino) και η τιμή που διαβάζεται από την αναλογική είσοδο

δεν “ξεχειλίσει”. Δείτε τις οδηγίες στο κεφάλαιο 5 για περισσότερα σχετικά με τη μετατροπή αναλογικών επιπέδων σε τιμές τάσης.

Αν χρειάζεστε τις τιμές σε βαθμούς Φαρενάιτ, μπορείτε να χρησιμοποιήσετε τον αισθητήρα LM34 , καθώς αυτός παράγει μια έξοδο σε βαθμούς Φαρενάιτ ή μπορείτε να μετατρέψετε τις τιμές αυτής της οδηγίας, χρησιμοποιώντας τον παρακάτω τύπο:

$$\text{float } f = (\text{celsius} * 9) / 5 + 32);$$

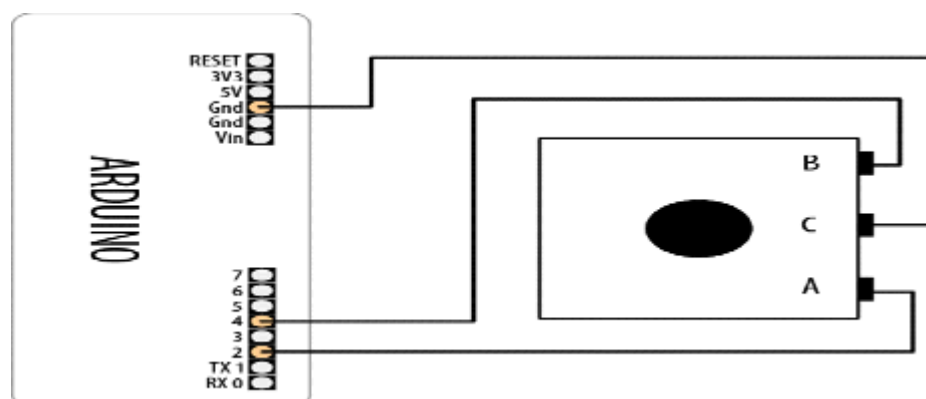
2.6 Ανάγνωση ετικετών RFID

2.6.1 Πρόβλημα

Θέλετε να διαβάσετε μια ετικέτα RFID και να απαντήσετε σε συγκεκριμένες ταυτότητες.

2.6.2 Λύση

Το Σχήμα 6-12 δείχνει ένα αναγνώστη Parallax RFID (αναγνώριση μέσω ραδιοσυχνότητας) που συνδέεται με τη σειριακή θύρα Arduino. (Ίσως χρειαστεί να αποσυνδέσετε τη συσκευή ανάγνωσης από τη σειριακή θύρα, όταν φορτώσετε το σκίτσο).



Σχήμα 6-12 . Σειριακός αναγνώστης RFID συνδεδεμένος με τον Arduino.

Το σκίτσο διαβάζει και εμφανίζει την τιμή της ετικέτας RFID:

```
/*
```

```
RFID sketch
```

```
Displays the value read from an RFID tag
```

```
*/
```

```
constintstartByte = 10; // ASCII line feed precedes each tag
```

```
constintendByte = 13; // ASCII carriage return terminates each tag
```

```
constinttagLength = 10; // the number of digits in tag
```

```
constinttotalLength = tagLength + 2; //tag length + start and end bytes
```

```
char tag[tagLength + 1]; // holds the tag and a terminating null
```

```
intbytesread = 0;
```

```
void setup()
```

```
{
```

```
Serial.begin(2400); // set this to the baud rate of your RFID reader
```

```
pinMode(2,OUTPUT); // connected to the RFID ENABLE pin
```

```
digitalWrite(2, LOW); // enable the RFID reader
```

```
}
```

```
void loop()
```

```
{
```

```
if(Serial.available() >= totalLength) // check if there's enough data
```

```
{
```

```
if(Serial.read() == startByte)
```

```
{
```

```
bytesread = 0; // start of tag so reset count to 0
```

```
while(bytesread<tagLength) // read 10 digit code
```

```
{
```

```
intval = Serial.read();
```

```
if((val == startByte)||val == endByte) // check for end of code
```

```
break;
```



```

tag[bytesread] = val;
bytesread = bytesread + 1; // ready to read next digit
}
if(Serial.read() == endByte) // check for the correct end character
{
tag[bytesread] = 0; // terminate the string
Serial.print("RFID tag is: ");
Serial.println(tag);
}
}
}
}
}
}
}

```

2.6.3 Συζήτηση

Μια ετικέτα αποτελείται από ένα χαρακτήρα έναρξης ακολουθούμενο από μία 10-ψήφια ετικέτα και τερματίζεται με ένα τελικό χαρακτήρα. Το σκίτσο περιμένει ένα πλήρες μήνυμα ετικέτας να είναι διαθέσιμο και προβάλλει την ετικέτα, αν είναι έγκυρη. Η ετικέτα λαμβάνεται ως ψηφία ASCII (βλέπε Οδηγία 4.4 για περισσότερες πληροφορίες σχετικά με τη λήψη ψηφίων ASCII). Μπορεί να θέλετε να τη μετατρέψετε σε έναν αριθμό, αν θέλετε να αποθηκεύσετε ή να συγκρίνετε τις ληφθείσες τιμές. Για να το κάνετε αυτό, αλλάξτε τις τελευταίες γραμμές ως εξής:

```

if(Serial.read() == endByte) // checkforthecorrectendcharacter
{
tag[bytesread] = 0; // terminate the string
longtagValue = atol(tag); // convert the ASCII tag to a long integer
Serial.print("RFID tag is: ");
Serial.println(tagValue);
}
}
}
}
}
}
}

```

```
}
```

RFID σημαίνει αναγνώριση μέσω ραδιοσυχνοτήτων και όπως υποδηλώνει το όνομα, είναι ευαίσθητη σε ραδιοσυχνότητες και μπορεί να είναι επιρρεπής σε παρεμβολές. Ο κωδικός στη λύση αυτής της οδηγίας θα χρησιμοποιήσει μόνο τον κωδικό του σωστού μήκους που περιέχει τις σωστές τιμές έναρξης και λήξης, που θα πρέπει να εξαλείψουν τα περισσότερα σφάλματα. Αλλά μπορείτε να κάνετε τον κώδικα πιο ανθεκτικό διαβάζοντας την ετικέτα πάνω από μία φορά και μόνο με τη χρήση των δεδομένων αν είναι κάθε φορά τα ίδια. (RFID αναγνώστες, όπως οι Parallax θα επαναλάβουν τον κωδικό, ενώ μια έγκυρη κάρτα είναι κοντά στον αναγνώστη).

Για να το κάνετε αυτό , προσθέστε τις ακόλουθες γραμμές στις τελευταίες γραμμές στο προηγούμενο απόσπασμα κώδικα:

```
if(Serial.read() == endByte) // check for the correct end character
{
  tag[bytesread] = 0; // terminate the string
  longtagValue = atol(tag); // convert the ASCII tag to a long integer
  if (tagValue == lastTagValue)
  {
    Serial.print("RFID tag is: ");
    Serial.println(tagValue);
    lasTagValue = tagValue;
  }
}
```

Θα χρειαστεί να προσθέσετε τη δήλωση για lastTagValue στην κορυφή του σκίτσου :

```
Long lastTagValue=0;
```

Η προσέγγιση αυτή είναι παρόμοια με το κώδικα από την Οδηγία 5.3. Αυτό σημαίνει ότι θα πάρετε επιβεβαίωση της κάρτας, μόνο αν

παρουσιάζεται για αρκετά μεγάλο διάστημα για δύο αναγνώσεις που είναι να ληφθούν, αλλά ψευδείς αναγνώσεις θα είναι λιγότερο πιθανές. Μπορείτε να αποφύγετε τυχαία πυροδότηση καθιστώντας αναγκαίο για την κάρτα να είναι παρούσα για ένα ορισμένο χρονικό διάστημα, πριν ο αριθμός αναφερθεί.

ΚΕΦΑΛΑΙΟ 3

ΕΙΣΑΓΩΓΗ ΣΤΟ INTERNET ΤΩΝ ΠΡΑΓΜΑΤΩΝ

«Όταν μιλάμε για ένα Internet πραγμάτων, δεν είναι βάζοντας απλά ετικέτες RFID σε κάποια χαζά πράγματα, γι 'αυτό εμείς οι έξυπνοι άνθρωποι γνωρίζουμε τι είναι αυτό το χαζό πράγμα. Είναι για την ενσωμάτωση νοημοσύνης, έτσι τα πράγματα να γίνονται πιο έξυπνα και να κάνουν περισσότερα από ό, τι ήταν σχεδιασμένο να κάνουν. »

NicholasNegroponte

3.1 ΤΟ ΙΝΤΕΡΝΕΤ ΤΩΝ ΠΡΑΓΜΑΤΩΝ

Το Διαδίκτυο έχει αρχικά ξεκινήσει ως το «Διαδίκτυο των Υπολογιστών», ένα παγκόσμιο δίκτυο που επιτρέπει υπηρεσίες που περιλαμβάνουν τώρα το WorldWide Web (WWW), το FileTransferProtocol (FTP) και άλλους χρήστες να επικοινωνούν μεταξύ τους και να ανταλλάσσουν πληροφορίες. Τα τελευταία χρόνια, η ισχύς επεξεργασίας της συσκευής και η ικανότητα αποθήκευσης αυξάνονται ενώ την ίδια ώρα, η τεχνολογία κάνει τις συσκευές διαβρωτικές, κινητές και φθαρτές. Επιπλέον, οι τεχνολογίες δικτύωσης εξελίσσονται και τα συστήματα ηλεκτρονικής επικοινωνίας γίνονται όλο μικρότερα και φθηνότερα. Οι συσκευές είναι όλο και περισσότερο εξοπλισμένες με αισθητήρες και ενεργοποιητές, δημιουργώντας περιβάλλοντα όπου συνδέονται με διάφορα δίκτυα. Οι συσκευές μπορούν να αισθάνονται, υπολογίζουν, ενεργούν και ως εκ τούτου να γίνουν έξυπνα τμήματα του λεγόμενου «Internet των πραγμάτων».

Υπάρχουν διάφοροι ορισμοί για το Διαδίκτυο των πραγμάτων (IoT), που επίσης εξηγεί ποιες είναι οι κύριες λειτουργίες του και τι πρέπει να περιμένουμε όταν συνδέουμε «Πράγματα», μεταξύ τους και με το Διαδίκτυο. Μερικοί άνθρωποι προτείνουν ότι το «διαδίκτυο των πραγμάτων μπορεί να θεωρηθεί ως ένα δυνητικά ολοκληρωμένο μέρος του «Μελλοντικού Ίντερνετ». Η Wikipedia ορίζει το IoT, ως εξής:

"Ένα μέρος από την δυναμική παγκόσμια υποδομή του δικτύου με την αυτο – ρύθμιση των δυνατοτήτων που βασίζονται σε ανοικτά και δυσλειτουργικά πρωτοκόλλα επικοινωνίας, όπου τα φυσικά και τα εικονικά «Πράγματα», αλληλεπιδρούν μεταξύ τους. Αυτά τα «Πράγματα» έχουν συγκεκριμένες ταυτότητες, φυσικά χαρακτηριστικά, εικονικές «προσωπικότητες» και χρησιμοποιούν ευφυείς διεπαφές. Είναι σε θέση να αλληλεπιδρούν και να επικοινωνούν μεταξύ τους και με το περιβάλλον, με την ανταλλαγή δεδομένων και πληροφοριών

«ευαίσθητοποιημένα » για το περιβάλλον , ενώ αντιδρούν αυτόνομα στις εκδηλώσεις του « πραγματικού / φυσικού κόσμου και τους επηρεάζουν με διεργασίες που εκτελούνται, που ενεργοποιούν δράσεις και τη δημιουργία υπηρεσιών με ή χωρίς άμεση παρέμβασή μας. Διασυνδέσεις στην μορφή υπηρεσιών διευκολύνουν αλληλεπιδράσεις με αυτά τα «έξυπνα πράγματα», μέσω του Διαδικτύου , ερωτούν και αλλάζουν την κατάσταση τους και οποιαδήποτε πληροφορία που σχετίζεται με αυτά."

Για ανθρώπους σαν χομπίστες, λάτρεις των ηλεκτρονικών ή ερευνητές του αισθητήρα, το Ίντερνετ των πραγμάτων είναι νέα ευκαιρία και ταυτόχρονα μια νέα πρόκληση για τη διαχείριση των δεδομένων που έχουμε αποκτήσει από ενσωματωμένα ηλεκτρονικά μας έργα και τον έλεγχο των εξόδων τους.

Φανταστείτε να έχετε μια μικρή συσκευή στο μέγεθος ενός σπιρτόκουτου , που μπορεί να ανιχνεύει τη θερμοκρασία, υγρασία και τις συνθήκες φωτισμού του δωματίου σας , και να τα αναφέρει απευθείας σε μία web-based υπηρεσία .

Οι ενδείξεις από τους αισθητήρες μπορούν να προσεγγιστούν μόνο από εσάς μέσω browser της αρεσκείας σας, από το κινητό σας τηλέφωνο και από άλλες συσκευές στο χώρο σας , όπως το κεντρικό / σύστημα κλιματισμού θέρμανσης ή τα εσωτερικά φώτα συστήματος ελέγχου. Το τελευταίο μπορεί να προσαρμόσει τη θερμότητα και το φωτισμό στο χώρο σας αυτόματα , βεβαιώνοντάς σας ότι έχετε πάντα τις ιδανικές συνθήκες , όπως έχουν οριστεί από εσάς στην web-based υπηρεσία .

Τώρα φανταστείτε ότι δεν πρέπει να κτίσετε τα πάντα από το μηδέν, προκειμένου να αναπτύξετε ένα τέτοιο σύστημα. Φανταστείτε , επίσης, ότι δεν έχετε να ανησυχείτε για τη διαχείριση των δεδομένων. Πώς είναι τα δεδομένα αποθηκευμένα στο διαδίκτυο , τι είδους webserver και τεχνολογία web εφαρμογής πρέπει να χρησιμοποιήσετε για την υπηρεσία σας, πώς να ασφαλίσετε τα δεδομένα σας και να εφαρμόσετε διάφορους

μηχανισμούς αυθεντικότητας και κρυπτογράφησης! Δεν χρειάζεται καν να ανησυχείτε για την εκμάθηση πώς να αναπτύξετε κινητές εφαρμογές που μιλούν στην υπηρεσία σας! Οι web-based υπηρεσίες που αποκτούν και διαχειρίζονται τα δεδομένα σας, οι διεπαφές επικοινωνίας και τα πρωτόκολλα ανταλλαγής πληροφοριών είναι ήδη στη διάθεσή σας για να τα εξερευνήσετε ! Τι είναι αυτά; Είναι εφαρμογές και χαρακτηριστικά υφιστάμενων πλατφορμών που παρέχονται στους χρήστες με τη λειτουργικότητα του Ίντερνετ των πραγμάτων. Πού είναι; Θα εξερευνήσουμε μερικά από αυτά και τους τρόπους για να τα χρησιμοποιήσουμε στα ακόλουθα κεφάλαια αυτού του βιβλίου. Κατ'αρχάς , ας ρίξουμε μια πιο προσεκτική ματιά στις έννοιες του IoT .

3.1.2 Οι βασικές έννοιες

Ας συζητήσουμε εν συντομία τις βασικές έννοιες και τα βασικά συστατικά που χρησιμοποιούνται συνήθως για να περιγράψουν τον κόσμο του Ίντερνετ των πραγμάτων .

Όταν αναφερόμαστε σε « πράγματα », μιλάμε για συσκευές και αντικείμενα καθημερινής χρήσης , από τα μικρά (όπως ρολόγια χειρός και ιατρικούς αισθητήρες) έως τα πραγματικά μεγάλα (σαν ρομπότ , αυτοκίνητα και κτίρια).

Όλες αυτά περιέχουν συσκευές που αλληλεπιδρούν με τους χρήστες με τη δημιουργία και ανάκτηση πληροφοριών σχετικά με και από το περιβάλλον τους (βλ. Σχήμα 1-1). Περιέχουν επίσης υλικό που τους επιτρέπει να ελέγχουν τις εξόδους, (όπως διακόπτες ρελέ ή ψηφιακές θύρες).

Δεν έχει σημασία τι ορισμό του Ίντερνετ των πραγμάτων μπορείτε να βρείτε , η κύρια ιδέα πίσω από κάθε τεχνολογία Ίντερνετ των πραγμάτων και εφαρμογή είναι η ίδια: οι συσκευές έχουν ενσωματωθεί με τον εικονικό κόσμο του Διαδικτύου και αλληλεπιδρούν με αυτό με τον

εντοπισμό , την ανίχνευση και την παρακολούθηση των αντικειμένων και του περιβάλλοντος αυτών. Οι χρήστες και οι προγραμματιστές προσθέτουν στοιχεία, τους δίνουν αίσθηση και ικανότητες δικτύωσης ,τα προγραμματίζουν για να εκτελέσουν τις παραπάνω ενέργειες και να δημιουργήσουν εφαρμογές Web που αλληλεπιδρούν με τις συσκευές.

Τα χαρακτηριστικά μιας συσκευής που μπορεί να δράσει ως μέλος ενός δικτύου ΙΟΤ μπορούν να συνοψιστούν στα ακόλουθα:

- Συλλογή και διαβίβαση δεδομένων: Η συσκευή μπορεί να αισθανθεί το περιβάλλον (π.χ. , το σπίτι σας ή το σώμα σας) και να συλλέξει πληροφορίες σχετικά με αυτό (π.χ. , θερμοκρασίας και συνθήκες φωτισμού) και να το διαβιβάσει σε άλλη συσκευή (μπορεί να είναι το κινητό σας τηλέφωνο ή το laptop) ή στο Internet .

- Ενεργοποίηση συσκευών που βασίζονται στις σκανδάλες: Μπορεί να προγραμματιστεί ώστε να ενεργοποιεί άλλες συσκευές π.χ., να ανάψει τα φώτα ή να απενεργοποιήσει τη θέρμανση), με βάση τους όρους που καθορίζονται από εσάς. Για παράδειγμα, μπορείτε να προγραμματίσετε τη συσκευή για να ανάψει τα φώτα , όταν πέφτει το σκοτάδι στο δωμάτιό σας .

- Λήψη πληροφοριών : Ένα μοναδικό χαρακτηριστικό για συσκευές ΙοΤ είναι ότι μπορούν επίσης να λαμβάνουν πληροφορίες από το δίκτυο στο οποίο ανήκουν (δηλ. άλλες συσκευές) ή μέσω του Internet (π.χ., πληροφορίες από εσάς, όπως νέες ωθήσεις, νέο καθεστώς λειτουργίας και σε ορισμένες περιπτώσεις, νέες λειτουργίες) .

- Βοήθεια Επικοινωνίας : συσκευές ΙοΤ που είναι μέλη ενός μίας δικτυακής συσκευής μπορούν επίσης να βοηθήσουν στην επικοινωνία (π.χ. διαβίβαση δεδομένων) μεταξύ άλλων κόμβων του ίδιου δικτύου .

Σκεφτείτε τους ως αγγελιαφόρους για συσκευές (κόμβους) που δεν είναι πολύ κοντά σε ένα τελικό σημείο (π.χ. ο router σας), προκειμένου να πάρει άμεση πληροφόρηση.

Σχήμα 1-1 . Μια απεικόνιση του «Ίντερνετ των πραγμάτων» . «Τα πράγματα », που αποτελούνται από διάφορους αισθητήρες και ενεργοποιητές αλληλεπιδρούν με το περιβάλλον και το Internet επιτρέπει στους χρήστες να διαχειρίζονται τα δεδομένα τους πάνω από διάφορες διεπαφές .

3.1.3 Αλληλεπίδραση με το Internet

Αυτό που κάνει τις IoT συσκευές διαφορετικές από τις συνηθισμένες συσκευές αισθητήρων είναι ουσιαστικά η δυνατότητα να επικοινωνούν (πιο συχνά), άμεσα ή έμμεσα, με το Internet. Λοιπόν, ποιοι είναι οι κύριοι λόγοι που μία συσκευή θα πρέπει να επικοινωνεί με μία υπηρεσία Internet ; Τι είδους υπηρεσία θα ήταν και τι είδους χαρακτηριστικά θα είχε; Πρώτον , οι αισθητήρες παράγουν πολλά δεδομένα που χρειάζεται με κάποιο τρόπο να διαχειρίζονται. Συνήθως , η ενσωματωμένη μνήμη είναι αρκετά περιορισμένη , οπότε οι άνθρωποι χρησιμοποιούν εναλλακτικές λύσεις, όπως η αποθήκευση δεδομένων σε κάρτες μνήμης , ή σε ηλεκτρονικούς υπολογιστές σε περιπτώσεις που οι αισθητήρες συνδέονται με αυτούς. Δεδομένου ότι οι αισθητήρες μπορούν να ενσωματωθούν σε συσκευές με δυνατότητες περαιτέρω δικτύωσης, γιατί να μην αποθηκεύονται οι πληροφορίες σε απευθείας σύνδεση; Μέσα από αυτόν τον τρόπο μπορούμε να λύσουμε το θέμα της περιορισμένης αποθήκευσης και την ίδια στιγμή, μπορούμε να έχουμε πρόσβαση στα δεδομένα οπουδήποτε , οποτεδήποτε με τη χρήση κατάλληλων εφαρμογών web. Η Εικόνα 1-2 απεικονίζει τα κύρια χαρακτηριστικά των « πραγμάτων» και της αλληλεπίδρασής τους με Υπηρεσίες Internet .

Σχήμα 1-2 . Εικονογράφηση από τα κύρια χαρακτηριστικά των «πραγμάτων» στα δίκτυα IoT και οι αντίστοιχες Internet υπηρεσίες.

Επιπλέον, οι περισσότερες από αυτές τις εφαρμογές web παρέχουν διεπαφές για την ανταλλαγή δεδομένων μεταξύ άλλων εφαρμογών και πιο

χρήσιμα, με κινητές εφαρμογές. Έτσι, είναι πιθανό ότι το iPhone σας ή Android κινητό μπορεί να μιλήσει απευθείας στη συσκευή σας IoT!

Μπορούμε επίσης να χρησιμοποιήσουμε τις διαδικτυακές πλατφόρμες για να στείλουμε τα στοιχεία πίσω στις συσκευές . Τα δεδομένα μπορεί να είναι το έναυσμα, οδηγίες για την ενεργοποίηση των ενεργοποιητών και διακόπτες , ή απλά πληροφορίες από το Διαδίκτυο, σαν ένα κανάλι καιρού ή ένα λογαριασμό Twitter που μπορεί να εμφανιστεί μέσω μιας διεπαφής , όπως μια οθόνη LCD.

Οι web-based πλατφόρμες που επιτρέπουν την προαναφερθείσα λειτουργικότητα για τη διαχείριση δεδομένων και την ανταλλαγή πληροφοριών βασίζονται σε Cloudcomputing. Περισσότερες πληροφορίες σχετικά με Cloudcomputing πλατφόρμες θα παρουσιαστούν στο Κεφάλαιο 3 αυτού του βιβλίου .

Προκειμένου τα πράγματα να μιλούν μεταξύ τους και με το Διαδίκτυο , θα πρέπει να έχουν ειδικές ικανότητες και να εξυπηρετούν πιο συγκεκριμένες λειτουργίες. Η επόμενη ενότητα συζητά ποια είναι τα στοιχεία που περιλαμβάνονται στις συσκευές IoT, που τους επιτρέπουν να αντιλαμβάνονται το περιβάλλον , να δρουν και να επικοινωνούν .

3.1.4 Βασικά Εξαρτήματα από IoT συσκευές

Ας συζητήσουμε ποια είναι τα κύρια συστατικά της συσκευής IoT . Δηλαδή, τέτοια συστατικά είναι οι κύριες μονάδες ελέγχου μ(οι «εγκέφαλοι » των συσκευών), οι αισθητήρες που συλλέγουν πληροφορίες – σήματα από το περιβάλλον , οι ενότητες επικοινωνίας και οι πηγές ενέργειας.

3.1.5 Μονάδες Ελέγχου

Συνήθως στον κόσμο του Ίντερνετ των πραγμάτων , οι συσκευές χρησιμοποιούν έναν μικροελεγκτή ως κύρια μονάδα ελέγχου που είναι

αρμόδιο για τις προαναφερθείσες λειτουργίες. Ένας μικροελεγκτής μπορεί να θεωρηθεί ως ένας μικρός υπολογιστής σε ένα ενιαίο ολοκληρωμένο κύκλωμα (συντομογραφία συχνά ως IC) που περιέχει ένα πυρήνα επεξεργαστή, τη μνήμη και προγραμματιζόμενα περιφερειακά εισόδου και εξόδου . Η εικόνα 1-3 παρουσιάζει ένα πολύ δημοφιλές μικροελεγκτή chip . Τα περιφερειακά ελέγχονται μέσω διαφόρων ακροδεκτών εισόδου / εξόδου γενικού σκοπού , που είναι γνωστά ως καρφίτσες GPIO . Καρφίτσες GPIO διαμορφώνονται είτε με είσοδο (ανάγνωση) δεδομένων ή δεδομένα εξόδου. Όταν είναι διαμορφωμένα σε μια κατάσταση εισόδου , χρησιμοποιούνται για την ανάγνωση πληροφοριών από αισθητήρες και δέχονται εξωτερικά σήματα (π.χ. εκδηλώσεις κουμπι) . Ρυθμισμένες για την κατάσταση της εξόδου , οι καρφίτσες GPIO μπορούν να ελέγξουν εξωτερικές συσκευές, όπως τα LED , κινητήρες , διακόπτες ρελέ , κλπ. Επιπλέον, αυτές οι πινέζες χρησιμοποιούνται επίσης από τον υπεύθυνο για την επικοινωνία με συσκευές , όπως μόντεμ και άλλου είδους ενότητες επικοινωνίας.

Άλλα σημαντικά τμήματα των μικροελεγκτών είναι οι μετατροπείς αναλογικού σε ψηφιακό (ADC) που χρησιμοποιούνται για τη μετατροπή εισερχόμενων αναλογικών δεδομένων (σήμα) σε μια μορφή που ο επεξεργαστής μπορεί να αναγνωρίσει.

Εκτός από τους μετατροπείς , πολλές τέτοιες μονάδες ελέγχου περιλαμβάνουν επίσης μια ποικιλία χρονομέτρων.

Θα συζητήσουμε περισσότερα για μικροελεγκτές και τα συστατικά τους στα επόμενα κεφάλαια .

Πρόσθετες ενότητες όπως PulseWidthModulation (PWM) ενότητες περιλαμβάνονται επίσης στο μικροελεγκτή και το ωθούν να μπορέσει να επεξεργαστεί και να ελέγχει πιο προηγμένες συσκευές όπως τα ηλεκτρικοί μετατροπείς και κινητήρες χωρίς τη χρήση πολλών πόρων στην παραγωγή παλμικών σημάτων προγραμματισμού .

Σχήμα 1-3 . Το τσιπ ATmega328 από την Atmel . Ένας από τους πιο ευρέως χρησιμοποιούμενους μικροελεγκτές σε ενσωματωμένα προγράμματα . Κάθε άξονας έχει μια συγκεκριμένη λειτουργία. Λαμβάνει αναλογικά σήματα , επικοινωνεί μέσω σειριακής διασύνδεσης με άλλα συστατικά , λαμβάνει τάση εισόδου, παράγει παλμούς ή ψηφιακή έξοδο, κλπ. (εικόνα του Sparkfun).

3.1.6 Αισθητήρες

Οι αισθητήρες είναι συσκευές που μπορούν να μετρήσουν μια φυσική ποσότητα (όπως θερμοκρασία, υγρασία, κ.λπ.) και να τη μετατρέψουν σε ένα σήμα , το οποίο μπορεί να διαβαστεί και να ερμηνευτεί από τη μονάδα μικροελεγκτή. Είναι οι συσκευές που είναι πιο πιθανό να συνδέονται με τις ακίδες εισόδου του μικροελεγκτή σε ενσωματωμένα μαζέργα. Γενικά, οι περισσότεροι αισθητήρες εμπίπτουν σε δύο κατηγορίες: αναλογικών και ψηφιακών αισθητήρων .

Ένα αναλογικό αισθητήρα , όπως ένα θερμίστορ (το οποίο στην πραγματικότητα είναι μια αντίσταση που αλλάζει την αντίστασή της με βάση τη θερμοκρασία που χρησιμοποιείται σε ψηφιακά θερμόμετρα) , ή έναν αισθητήρα υγρασίας, συνδέεται μέσα σε κυκλώματα , έτσι ώστε να παράγει ένα συγκεκριμένο εύρος τάσης , συνήθως μεταξύ 0 Volt έως 5 βολτ. Αυτή η έξοδος πηγαίνει στον αναλογικό πείρο εισόδου της μονάδας μικροελεγκτή (βλ. Σχήμα 1-4). Το επόμενο χρησιμοποιεί το κατάλληλο αναλογικό προς ψηφιακό (A / D) κύκλωμα για τη μετατροπή της τάσης σε μια αριθμητική τιμή που μπορούμε αργότερα να διαβάσουμε, να επεξεργαστούμε και να αποστείλουμε στο Internet . Το σχήμα 1-4 δείχνει πώς ένας αναλογικός αισθητήρας μπορεί να είναι συνδεδεμένος με μια συσκευή μικροελεγκτή. Όλοι οι αναλογικοί αισθητήρες συνήθως έρχονται

με 3 ακίδες σύνδεσης, μία για τη λήψη τάσης (Vcc), μία για σύνδεση γείωσης (GND) και τον πείρο τάσης εξόδου (OUT).

Σχήμα 1-4 . Αυτό είναι το πώς ένας αισθητήρας υγρασίας συνδέεται με το μικροελεγκτή ATmega168 . Το Vcc λαμβάνει τάση λειτουργίας και το GND συνδέεται με το έδαφος της πηγής ισχύος . Το OUT είναι η ακίδα που εξάγει το ληφθέν σήμα και συνδέεται με ακίδα μικροελεγκτή 23 (αναλογική θύρα) .

Οι ψηφιακοί αισθητήρες δημιουργούν αυτό που ονομάζεται ένα χαρακτηριστικό σήμα. Ένα τέτοιο σήμα έχει διαφορετικά (συνήθως μόνο δυαδική) states, όπως τα states λογικής πύλης 0 και 1, τα οποία αντιπροσωπεύονται από υψηλό και χαμηλό ρεύμα. Για παράδειγμα, σκεφτείτε ένα διακόπτη μπουτόν (που , επί τη ευκαιρία, είναι ένα από τις απλούστερες μορφές ψηφιακών αισθητήρων): έχει δύο διακριτές τιμές . Είναι ενεργοποιημένο ή είναι απενεργοποιημένο. Κάποιοι πιο περίπλοκοι αισθητήρες έρχονται επίσης με μια ψηφιακή διασύνδεση. Αυτό σημαίνει ότι αντί να παράγουν ένα σήμα που αντιπροσωπεύει τη μεταβολή τάσης μεταξύ 0-5 με βάση τη μέτρηση τους, στην πραγματικότητα παράγουν ένα ρεύμα δυαδικών ψηφίων (0 και 1 states).

Περισσότερες πληροφορίες σχετικά με το πώς οι αισθητήρες δουλεύουν, παρέχονται στο κεφάλαιο 2 του παρόντος βιβλίου .

3.1.7 Ενότητες Επικοινωνίας

Οι ενότητες επικοινωνίας είναι μέρη της συσκευής , τα οποία είναι υπεύθυνα για την επικοινωνία με το υπόλοιπο της πλατφόρμας IoT . Παρέχουν σύνδεση , σύμφωνα με το ασύρματο ή ενσύρματο πρωτόκολλο επικοινωνίας, για το οποίο έχουν σχεδιαστεί (τέτοια πρωτόκολλα παρουσιάζονται στο υπόλοιπο του παρόντος κεφαλαίου) . Συνήθως αποτελούνται από ενσωματωμένες ηλεκτρονικές μονάδες που εφαρμόζουν την επικοινωνία (δηλαδή μετατρέπουν την πληροφορία που

λαμβάνεται σε bits και bytes, σε ραδιοκύματα ή σήματα που μεταφέρονται από καλώδιο, αντίστοιχα). Οι ασύρματες μονάδες συνήθως αποτελούνται επίσης από μία κεραία για την επίτευξη της μέγιστης κλίμακας, είτε εξωτερικά είτε εσωτερικά για τη βελτιστοποίηση του μεγέθους της συσκευής.

Πιο πιθανό, η επικοινωνία μεταξύ των συσκευών IoT και το Διαδίκτυο γίνεται με δύο τρόπους : α) υπάρχει ένας ενδιάμεσος κόμβος μέσω Internet , που λειτουργεί ως πύλη, β) η συσκευή IoT έχει άμεση επικοινωνία με το Διαδίκτυο. Στην πρώτη περίπτωση, η συσκευή είναι συνήθως συνδεδεμένη με τον υπολογιστή και στέλνει δεδομένα σε αυτό χρησιμοποιώντας , π.χ. , μια θύρα USB. Ο υπολογιστής λαμβάνει τα δεδομένα και χρησιμοποιώντας το κατάλληλο λογισμικό το προωθεί στο διαδίκτυο. Στη δεύτερη περίπτωση , τα πράγματα είναι πολύ πιο απλά και οι συσκευές μπορούν να λειτουργούν και να επικοινωνούν πιο αυτόνομα.

Σε κάθε περίπτωση, η επικοινωνία μπορεί να γίνει σύμφωνα με την ασύρματη τεχνολογία που χρησιμοποιείται. Οι περισσότερες συσκευές που υποστηρίζουν ασύρματες (όπως WiFi , Bluetooth και ZigBee) και ενσύρματες τεχνολογίες (όπως η Ethernet) υποστηρίζουν επίσης το πρωτόκολλο TCP / IP . Το πρωτόκολλο TCP / IP (για όσους δεν είναι εξοικειωμένοι με αυτό) είναι ο τρόπος που υπολογιστές, κινητά τηλέφωνα και συσκευές μέσω Internet επικοινωνούν μεταξύ τους και με το Διαδίκτυο. Ορίζει όλα τα απαραίτητα της επικοινωνίας (π.χ., πως οι συσκευές προσδιορίζονται η μία από την άλλη χρησιμοποιώντας διευθύνσεις IP, πώς οι πληροφορίες είναι χωρισμένες σε μικρά πακέτα και μεταδίδονται, κ.λπ.).

Λαμβάνει επίσης μέριμνα όλων των ζητημάτων ρύθμισης της σύνδεσης και εξασφαλίζει ότι οι πληροφορίες παραδίδονται σωστά από την αναμετάδοση δεδομένων κάθε φορά που αυτό κρίνεται απαραίτητο.

Τι γίνεται με την επικοινωνία μεταξύ των συσκευών; Η επικοινωνία μεταξύ των κύριων μονάδων ελέγχου και τις διάφορες μονάδες επικοινωνίας εκτελείται στις περισσότερες περιπτώσεις (και στην περίπτωση της μελέτης μας) χρησιμοποιώντας το σειριακό πρωτόκολλο. Σειριακό είναι ένα πρωτόκολλο επικοινωνίας, που είναι στάνταρ σχεδόν σε κάθε υπολογιστή και ηλεκτρονική συσκευή . Η έννοια του σειριακού πρωτοκόλλου είναι απλή . Κάθε συσκευή έχει δύο θύρες (ή συνδετήρες), μία για τη μετάδοση δεδομένων , που συνήθως περιγράφεται ως Tx , και μία για λήψη δεδομένων , που συνήθως περιγράφεται ως Rx. Επίσης, απαιτούνται γείωση και σύνδεση που παρέχονται από την πηγή ενέργειας. Κάθε σειριακή θύρα στέλνει και λαμβάνει bytes πληροφοριών, ένα bit τη φορά. Έτσι , προκειμένου να επικοινωνήσουν , και οι δύο μονάδες ελέγχου (π.χ. μικροελεγκτές) και οι ενότητες επικοινωνίας μοιράζονται ένα ζεύγος θυρών Rx / Tx , όπως φαίνεται στο Σχήμα 1-5.

Σχήμα 1-5 . Αυτό είναι το πώς ένα σύστημα επικοινωνίας Bluetooth συνδέεται με το ATmega168 μικροελεγκτή.

Το Vcc λαμβάνει τάση λειτουργίας και το GND συνδέεται με το έδαφος της πηγής ισχύος.

Tx και Rx καρφίτσες είναι οι καρφίτσες που στέλνουν και λαμβάνουν σειριακά δεδομένα και να συνδέονται σε ακίδες μικροελεγκτή 2 και 3(Rx και Tx αντίστοιχα).

Αν και αυτό είναι πιο αργή από την παράλληλη επικοινωνία, η οποία επιτρέπει τη μετάδοση ενός ολόκληρου byte τη φορά, είναι πιο απλό και μπορείτε να το χρησιμοποιήσετε σε μεγαλύτερες αποστάσεις. Επειδή η σειριακή είναι ασύγχρονη , η θύρα μπορεί να μεταδίδει δεδομένα σε μια γραμμή , ενώ λαμβάνει δεδομένα σε άλλη.

Μια σύντομη παρουσίαση της επικοινωνίας τεχνολογιών γενικής εφαρμογής για συσκευές IoT ακολουθεί στην Ενότητα "Τεχνολογίες Επικοινωνίας"

3.1.8 Πηγές ενέργειας

Κάθε ηλεκτρονική συσκευή χρειάζεται ηλεκτρική ενέργεια για να λειτουργήσει σωστά . Η ηλεκτρική ενέργεια είναι το αποτέλεσμα της διαφοράς δυναμικού μεταξύ δύο σημείων, αλλιώς γνωστή ως τάση, και του ηλεκτρικού ρεύματος που διαρρέει μέσω ενός κυκλώματος. Στις περισσότερες ηλεκτρονικές συσκευές , καθώς και στα έργα που ασχολούμαστε σε αυτό το βιβλίο , τα δύο σημεία αναφέρονται ως σημείο εισόδου τάσης (συνήθως χαρακτηρίζονται ως V_{in} ή V_{cc}) και της γείωσης (που συνήθως χαρακτηρίζονται ως GND).

Σε μικρές συσκευές το ρεύμα παράγεται συνήθως από πηγές όπως μπαταρίες , θερμοηλεκτρικά ζεύγη και ηλιακά κύτταρα . Οι μπαταρίες είναι ηλεκτροχημικά στοιχεία που μετατρέπουν τη χημική δύναμη σε ηλεκτρική παράγοντας συνεχές ρεύμα (DC).

Φθαρτές και κινητές συσκευές τροφοδοτούνται κυρίως από μικρές ελαφριές μπαταρίες που μπορούν επίσης να επαναφορτίζονται (όπως στο Σχήμα 1-6) για μεγαλύτερη διάρκεια ζωής. Με τη χρήση απλών κυκλωμάτων και κατάλληλου προγραμματισμού μικροελεγκτών , οι συσκευές μπορούν να έχουν γνώση της κατάστασης της μπαταρίας και ειδοποιούν τους χρήστες όταν πρέπει να επαναφορτιστούν ή να αντικατασταθούν.

Σχήμα 1-6 . Πιθανώς η μικρότερη επαναφορτιζόμενη μπαταρία κατανάλωσης. Ζυγίζει μόνο 330 mg και λειτουργεί σε 3.6V (εικόνα του Powerstream).

Όταν επιλέγετε την κατάλληλη πηγή της μπαταρίας για τα δικά σας έργα, μπορείτε να αναζητήσετε τη λειτουργική τάση και χωρητικότητα της

μπαταρίας . Ο επόμενος συνδέεται με τη συνολική ενέργεια που αποθηκεύεται εντός της μπαταρίας . Η χωρητικότητα σε γενικές γραμμές μετράται σε βατ ανά ώρα (Wh), κιλοβάτ ανά ώρα(kWh) ή αμπέρ ανά ώρα (Ahr). Για τους αισθητήρες και άλλα εξαρτήματα της IoT μελέτης που θα «κτίσετε» , το πιο κοινό μέτρο χωρητικότητας της μπαταρίας είναι Ah.

Για να σας δώσω ένα παράδειγμα σχετικά με την ικανότητα και την πραγματική σημασία της, αν η συσκευή σας IoT χρειάζεται 50mA για να λειτουργήσει και η μπαταρία σας παρέχει 50mAh , θα τροφοδοτήσει τη συσκευή σας κατάλληλα για περίπου 1 ώρα πριν χρειαστεί επαναφόρτιση.

3.1.9 Τεχνολογίες Επικοινωνίας

Τα « πράγματα » έχουν ανάγκη να μιλήσουν το ένα στο άλλο και να μιλάνε στο Διαδίκτυο, προκειμένου να ανταλλάσσουν εξόδους αισθητήρα , ωθήσεις , μηνύματα κατάστασης, κλπ. Για να γίνει αυτό, οι συσκευές πρέπει να ενσωματώσουν ένα ασύρματο (κατά προτίμηση) ή ένα ενσύρματο σύστημα επικοινωνίας. Οι κύριες τεχνολογίες της επικοινωνίας που μπορεί να χρησιμοποιηθούν από τέτοιες συσκευές συνοψίζονται σε αυτό το τμήμα. Εκτός από την σύντομη περιγραφή των τεχνολογιών, παρουσιάζονται επίσης δείγματα των ηλεκτρονικών μονάδων που επιτρέπουν την αντίστοιχη επικοινωνία.

Οι ενότητες που επιλέγονται χαρακτηρίζονται από τις ειδικές διεπαφές που έχουν για απευθείας σύνδεση σε μικροελεγκτές και πλατφόρμες όπως οι ηλεκτρονικές πρωτότυπες open-source πλατφόρμες του «Arduino».

3.1.10 RFID

Η τεχνολογία RadioFrequencyIdentification (RFID) έχει αρχικά εισαχθεί για εντοπισμό και παρακολούθηση αντικειμένων με τη βοήθεια μικρών ηλεκτρονικών τσιπ, που ονομάζονται ετικέτες . Είναι η πιο κοινή

τεχνολογία πίσω από τον εντοπισμό στοιχείων (που σας λέει πού είναι το ταχυδρομικό σας δέμα πριν την άφιξη στον προορισμό του) και τον εντοπισμό αντικειμένων (π.χ., στην αυτόματη είσπραξη διοδίων).

Οι ετικέτες RFID κατηγοριοποιούνται σε παθητικές, ενεργητικές και παθητικές από μπαταρία υποβοηθούμενες (BAP). Οι παθητικές ετικέτες δεν έχουν μια πηγή ισχύος (μπαταρία) και ως εκ τούτου δεν μπορούν να μεταδώσουν πληροφορίες μόνες τους. Τροφοδοτούνται - ενεργοποιούνται από τον αναγνώστη RFID και μεταδίδουν μόνο ένα μικρό αριθμό πληροφοριών (συνήθως έναν αριθμό αναγνώρισης (ID) της ετικέτας). Οι ενεργές ετικέτες αντίθετα έχουν δική τους μπαταρία και μπορούν να μεταδίδουν συνεχώς τα δεδομένα. Μία BAP ετικέτα μπορεί να θεωρηθεί ως ένα υβρίδιο : φέρει μπαταρία, αλλά μεταδίδει πληροφορίες μόνο με την παρουσία ενός αναγνώστη RFID. Η μπαταρία τους βοηθά να μεταδίδουν το σήμα τους σε μεγαλύτερη απόσταση από τις παθητικές ετικέτες (που περιορίζονται σε λίγα εκατοστά).

Σχήμα 1-7 . Η μονάδα RFID και η κεραία του . Μπορεί να επικοινωνεί απευθείας με το Arduino , χρησιμοποιώντας το Σειριακό πρωτόκολλο (εικόνα του Seeedstudio).

Οι περισσότερες ετικέτες RFID αποτελούνται από ένα ολοκληρωμένο κύκλωμα για την αποθήκευση και την επεξεργασία των πληροφοριών , των εξαρτημάτων βασικών ραδιοσυχνοτήτων (RF) για τη διαβίβαση των πληροφοριών ασύρματα και με κεραία.

Το RFID έχει αρχικά χαρακτηριστεί ως η δύναμη επικοινωνίας για το Διαδίκτυο των πραγμάτων , λόγω του χαμηλού κόστους του , η υψηλή κινητικότητα και αποτελεσματικότητα στον εντοπισμό συσκευών και αντικειμένων.

Το σχήμα 1-7 απεικονίζει μία μονάδα RFID και την κεραία που μπορούν να επικοινωνούν απευθείας με μικροελεγκτή.

Παρά το ότι το RFID είναι πολύ δημοφιλές για την ταυτοποίηση της συσκευής και την ανταλλαγή κάποιων πληροφοριών (π.χ. , βάσει RFID αισθητήρες θερμοκρασίας), δεν μπορεί από μόνη της να υποστηρίξει τη δημιουργία δικτύων IoT, δεδομένου ότι δεν μπορεί να παράσχει οποιαδήποτε άμεση ή έμμεση (π.χ. , μέσω μιας πύλης) επικοινωνία με το Internet. Η εγγύτητα της συσκευής είναι επίσης ένα άλλο μειονέκτημα.

3.1.11 Bluetooth

Χρησιμοποιείτε την τεχνολογία Bluetooth για να συνδέσετε ασύρματα τα ακουστικά σας με το κινητό σας τηλέφωνο , ή να μεταφέρετε φωτογραφίες από το τελευταίο στον υπολογιστή σας . Το Bluetooth είναι μια τεχνολογία πρότυπο για την ανταλλαγή δεδομένων σε μικρές αποστάσεις (με τη χρήση μικρού μήκους κύματος ραδιοφωνικές μεταδόσεις στην μπάντα ISM from 2400-2480 MHz) από σταθερές και κινητές συσκευές , δημιουργώντας δίκτυα προσωπικής περιοχής. Το Bluetooth υπήρξε ένα από τα πρώτα ασύρματα πρωτόκολλα επικοινωνίας που έχουν σχεδιαστεί με χαμηλότερη κατανάλωση ενέργειας για την αντικατάσταση των ενσύρματων επικοινωνιών μικρής εμβέλειας (στα περιφερειακά υπολογιστών, αξεσουάρ κινητής τηλεφωνίας , κλπ.) . Ένα σημαντικό χαρακτηριστικό του Bluetooth είναι ότι οι συσκευές μπορούν να ανακαλύψουν και να επικοινωνούν μεταξύ τους χωρίς την ανάγκη να είναι σε οπτική γραμμή της όρασης (όπωςσε επικοινωνία με υπέρυθρες), το οποίο είναι πολύ σημαντικό κατά τη χρήση Bluetooth , ως δίκτυο τεχνολογίας για την ανάπτυξη συστημάτων αισθητήρων.

Το Bluetooth είναι το όνομά του από τον του 10ο αιώνα Δανικό βασιλιά HaraldBlatand, που μεταφράζεται ως HaroldBluetooth στα αγγλικά!

Συνήθως χρησιμοποιείται για τη σύνδεση μικρών συσκευών μεταξύ τους, λόγω του γεγονότος ότι μπορεί να υποστηρίξει αυτόματα τη

δημιουργία δικτύων peer (δηλαδή δίκτυα των συσκευών που ανταλλάσσουν και διαβιβάζουν πληροφορίες) και παρέχει λειτουργικότητα επικοινωνίας με χαμηλή κατανάλωση ενέργειας.

Το τελευταίο είναι πολύ σημαντικό για την περίπτωση της IoT, δεδομένου ότι πολλές από τις συσκευές που κάποιος θα ήθελε να διασυνδέονται με το Ίντερνετ των πραγμάτων (αισθητήρες , ενεργοποιητές , κλπ.) έχουν περιορισμένους πόρους ισχύος . Ωστόσο, το σημαντικό μειονέκτημα του Bluetooth είναι ότι δεν μπορεί να παρέχει άμεση σύνδεση με το Διαδίκτυο. Κάποιος πρέπει να παρέχει ένα ενδιάμεσο κόμβο , π.χ. , έναν υπολογιστή που θα λειτουργεί ως πύλη προς τον έξω κόσμο . Εσείς μπορείτε να δείτε πώς μοιάζει ένα σύστημα επικοινωνίας Bluetooth, στο σχήμα 1-8.

Σχήμα 1-8 . Μονάδα επικοινωνίας Bluetooth για την εύκολη διασύνδεση με συσκευές, όπως το Arduino . Η μονάδα είναι τόσο μικρή που χωράει στο κινητό σας τηλέφωνο , τα χέρια σας είναι ελεύθερα (εικόνα του Sparkfun).

3.1.12 ZigBee

ZigBee είναι μία από τις τελευταίες και πιο προηγμένες ασύρματες τεχνολογίες που έχουν ευρέως ενσωματωθεί στον οικιακό αυτοματισμό και σε έξυπνες συσκευές σε όλο τον κόσμο. Έχει αναπτυχθεί ειδικά ως ένα ανοικτό παγκόσμιο πρότυπο για να αντιμετωπίσει τις μοναδικές ανάγκες των χαμηλού κόστους , χαμηλής ισχύος ασύρματων δικτύων για επικοινωνία μεταξύ συσκευών (επίσης γνωστή ως μηχανή σε μηχανή ή M2M δίκτυα) . Το πρότυπο ZigBee λειτουργεί σε μη αδειοδοτημένες ζώνες , συμπεριλαμβανομένων των 2.4 GHz , 900 MHz και 868 MHz.

Μια μονάδα ZigBee (βλ. Σχήμα 1-9) μπορεί να έχει χαμηλή κατανάλωση 50mA .Για παράδειγμα, μία επαναφορτιζόμενη μπαταρία

850mAh μπορεί να παρέχει περίπου 17 ώρες συνεχούς λειτουργίας για την εν λόγω ενότητα . Ο μέγιστος ρυθμός μετάδοσης δεδομένων είναι περίπου 250kbps και το εύρος επικοινωνίας μπορεί να κυμαίνεται από 100 έως 1 χιλιόμετρο (μέγιστο) ανάλογα με την ισχύ εξόδου (σε μικρά έργα χρησιμοποιούμε συνήθως ενότητες 1mW, που παρέχουν μέγιστο εύρος 100 m).

Σε σύγκριση με το Bluetooth , ZigBee παρέχει καλύτερη απόδοση ισχύος , και μεγαλύτερο εύρος , γεγονός που το καθιστά μια καλύτερη ασύρματη τεχνολογία για το δίκτυό σας IoT . Επίσης, επιτρέπει την αυτόματη δημιουργία δικτύων ομοτίμων και στην πραγματικότητα, θεωρείται ότι είναι πιο επεκτάσιμο σε αυτό το πλαίσιο.

Ωστόσο, εξακολουθεί να απαιτεί μια πύλη με σύνδεση στο Internet (π.χ. , ένα φορητό υπολογιστή), η οποία θα διαβιβάσει πληροφορίες από το Internet στο δίκτυο ZigBee και το αντίστροφο.

Σχήμα 1-9 . Μονάδα επικοινωνίας ZigBee σε αποσπώμενη μορφή, γνωστή ως μονάδες της σειράς XBee (εικόνα του Sparkfun).

3.1.13 WiFi

WiFi , επίσης γνωστό ως το πρότυπο IEEE 802.11x , είναι ο πιο συνηθισμένος τρόπος για να συνδέσετε συσκευές ασύρματα στο Internet. Το laptop σας, smartphone και Tablet PC είναι εξοπλισμένα με WiFiδιεπαφές και μιλάνε με τον ασύρματο router και σας παρέχει με αυτόν τον τρόπο την πρόσβαση στο Internet.

Οι εμπορικώς διαθέσιμες μονάδες WiFi (όπως εκείνη στο Σχήμα 1-10) μπορούν άμεσα να ενσωματωθούν σε μια συσκευή IoT και να παρέχουν άμεση συνδεσιμότητα. Το μείζον πλεονέκτημα έναντι των άλλων ασύρματων τεχνολογιών είναι το γεγονός ότι τα δίκτυα WiFi είναι πολύ εύκολο να εγκατασταθούν και έτσι οι συσκευές IoT με μονάδες

WiFi μπορούν να έχουν άμεση σύνδεση με το Internet . Ένα μειονέκτημα είναι το γεγονός ότι η τεχνολογία αυτή (που δεν ήταν σε καμία περίπτωση σχεδιασμένη για δίκτυα IoT) απαιτεί περισσότερη ενέργεια από τις άλλες.

Σχήμα 1-10 . Μονάδα επικοινωνίας WiFi σε αποσπώμενη μορφή γνωστή ως μονάδες της σειράς XBee (εικόνα του Seeedstudio).

3.1.14 Συνδέσεις RF

Μια άλλη επιλογή για να συνδέσετε συσκευές και να τις κάνετε να μιλήσουν είναι να χρησιμοποιήσετε απλές διασυνδέσεις με ραδιοσυχνότητες (RF).

Το τελευταίο είναι αρκετά φθηνό και μικρό (ιδανικό όταν το μέγεθος μετράει) και μπορεί να παρέχει εμβέλεια επικοινωνίας μεταξύ 100m και 1χλμ (ανάλογα με την ισχύ μετάδοσης και την κεραία που χρησιμοποιείται).

Οι ενότητες επικοινωνίας RF (βλ. Σχήμα 1-11) συνδέονται με μικροελεγκτή και συσκευές μέσω σειριακών θυρών, όπως και οι υπόλοιπες ενότητες που έχουμε εξετάσει . Ωστόσο, δεν παρέχουν καμία εφαρμογή του πρωτοκόλλου επικοινωνίας TCP / IP (ή οποιουδήποτε άλλου πρωτόκολλου).

Αυτό σημαίνει ότι αν θέλετε τις συσκευές σας να επικοινωνούν, πρέπει να δημιουργήσετε το δικό σας πρωτόκολλο για την εγκατάσταση επικοινωνίας, την αναγνώριση συσκευών μεταξύ τους και να βεβαιωθείτε ότι όλες οι πληροφορίες που έχετε διαβιβάσει, έχουν παραδοθεί .Οι ρυθμοί δεδομένων είναι αρκετά χαμηλοί (έως 1Mrbs) και επίσης, χρειάζεται μία Internet-enabled πύλη που θα παρέχει πρόσβαση στις συσκευές σας για την πραγματοποίηση πλήρους δικτύου IoT.

Σχήμα 1-11. Μια υπομονάδα πομποδέκτη RF (μπορεί να χρησιμοποιηθεί τόσο ως πομπός και δέκτης) . Έχει χαμηλή κατανάλωση

ενέργειας , μπορεί να υποστηρίξει ταχύτητες δεδομένων έως 1Mbps και μπορεί εύκολα να διασυνδεθεί με το Arduino σας (εικόνα του Sparkfun).

3.1.15 Κυψελοειδήδίκτυα : ΤοMobileInternet

Το «Mobile Internet » αναφέρεται συνήθως στην πρόσβαση στο Διαδίκτυο από κινητή συσκευή , όπως ένα smartphone ή φορητό υπολογιστή μέσω κινητών ευρυζωνικών δικτύων . Το ευρυζωνικό δίκτυο κινητής τηλεφωνίας είναι βασισμένο στην κυτταρική επικοινωνία , την ίδια τεχνολογία που χρησιμοποιείται στα κινητά μας τηλέφωνα για την εξυπηρέτηση των κλήσεων μας και των μηνυμάτων κειμένου. Μπορεί να προσφέρει άμεση σύνδεση στο Internet σε μια ποικιλία ρυθμού δεδομένων.

Διάφορα πρότυπα του δικτύου υπήρχαν για την εξυπηρέτηση του κινητού Internet . GPRS , 3G , WiMax , και LTE (μία από τις τεχνολογίες 4G) είναι μερικά. Ανάλογα με το πρότυπο και τη διαθέσιμη κάλυψη από το δίκτυο , οι ταχύτητες σύνδεσης μπορεί να πάνε από 80Kbps (GPRS) σε μερικά Mbps (3G και 4G).

Λόγω της πολυπλοκότητας του πρωτοκόλλου επικοινωνίας και κωδικοποίησης των πληροφοριών , εκτός από την υψηλή σε απαιτήσεις ισχύ σε περιπτώσεις όπου το σήμα υποδοχής είναι χαμηλό, η κατανάλωση μπαταρίας των συσκευών Mobile Internet – enabled, είναι ένα ζήτημα . Σκεφτείτε πόσο γρήγορα η μπαταρία του κινητού σας πέφτει, όταν περιηγείστε στο Internet . Εξακολουθεί να είναι μια καλή επιλογή για τη σύνδεση συσκευών απευθείας στο Internet , δεδομένου ότι μικρές GPRS ενότητες για το Arduino είναι διαθέσιμες (βλέπε Σχήμα 1-12) και η συνδεσιμότητα δεν απαιτεί επιπλέον υποδομή (π.χ., φορητό υπολογιστή συνδεδεμένο στο διαδίκτυο, όπως στην περίπτωση του ZigBee ή του Bluetooth).

Σχήμα 1-12. Ασπίδα GPRS για το Arduino σας στα αριστερά. Μονάδα GPRS στα δεξιά . Το μικρό μέγεθος του επιτρέπει να χωράει εύκολα σε κινητά τηλέφωνα, Tablet PC και συσκευές αισθητήρων (εικόνα του Sparkfun).

3.1.16 Καλωδιακή επικοινωνία

Οι συσκευές μπορούν σίγουρα να μιλήσουν μεταξύ τους και με το Internet μέσω ενσύρματων υποδομών , δεδομένης της κατάλληλης διασύνδεση δικτύου . Ποιος είναι ο καλύτερος τρόπος; Από την αρχή της εποχής του Διαδικτύου έως σήμερα , στην επιφάνεια εργασίας του υπολογιστή σας έχει τουλάχιστον μία θύρα Ethernet.

Το πρωτόκολλο Ethernet είναι πολύ καλά εδραιωμένο στις επικοινωνίες υπολογιστών . Δεν απαιτεί τόσο πολλή δύναμη όπως οι ασύρματες επικοινωνίες , μπορεί να επιτύχει πολύ υψηλούς ρυθμούς δεδομένων και το πιο σημαντικό είναι ότι είναι τόσο κοινό σε επικοινωνία υπολογιστών, που το μόνο που χρειάζεται να κάνετε είναι να συνδέσετε ένα καλώδιο Ethernet στη συσκευή και να συνδεθείτε απευθείας. Δεν υπάρχουν ανησυχίες σχετικά με τη διαθεσιμότητα του σήματος . Δεν παρέχει καμία κινητικότητα , αλλά τα πλεονεκτήματά του ήταν ο κύριος λόγος που ήταν μία από τα πρώτες τεχνολογίες δικτύωσης που χρησιμοποιούνται σε πλατφόρμες όπως το μικροελεγκτή Arduino. Το σχήμα 1-13 παρουσιάζει μια μονάδα Ethernet που μπορεί να συνδεθεί απευθείας στο Arduino και σε κάρτα Arduino Ethernet-enabled.

Σχήμα 1-13 . Στα αριστερά : Μια μονάδα Ethernet . Μπορεί να συνδεθεί απευθείας σε οποιαδήποτε συσκευή αισθητήρα και να παρέχει λειτουργικότητα σε ενσύρματο δίκτυο. Στα δεξιά: Μια κάρτα Arduino με ενσωματωμένο interface Ethernet (εικόνα του Arduino).

3.1.17 Ποιο είναι το καλύτερο?

Όταν ερχόμαστε στο σημείο όπου πρέπει να αποφασίσουμε ποια είναι η πιο κατάλληλη τεχνολογία επικοινωνίας για το δίκτυό σας IoT , υπάρχουν πολλά πράγματα που πρέπει να εξετάσουμε: κινητικότητα , εύρος δικτύου, κατανάλωση ενέργειας , μέγεθος και κόστος είναι τα πιο σημαντικά να αναφέρουμε. Ο πίνακας 1 - 1 παρέχει μια επισκόπηση των τεχνολογιών επικοινωνίας που παρουσιάστηκαν συγκρίνοντας τα χαρακτηριστικά τους.

Σε περίπτωση που οι συσκευές σας θα είναι σε μία σταθερή θέση , μια μονάδα Ethernet είναι μία από τις καλύτερες επιλογές για την παροχή επικοινωνίας με μεγάλη ταχύτητα και με χαμηλή κατανάλωση ενέργειας . Σε μια διαφορετική περίπτωση όπου η κινητικότητα είναι υποχρεωτική (ή δεν υπάρχει ενσύρματη υποδομή) , μπορούμε να επιλέξουμε ανάμεσα σε WiFi , κυτταρική και ZigBee . Το RF απαιτεί πολύ μεγαλύτερη προσπάθεια για το «κτίσιμο» ενός πρωτοκόλλου επικοινωνίας. Το ZigBee απαιτεί μια πύλη για την παροχή σύνδεσης στο Internet , αλλά κατά τα άλλα είναι μια πολύ καλή λύση , προσφέροντας χαμηλή κατανάλωση ενέργειας και καλό εύρος κάλυψης. Το WiFi μπορεί να παρέχει άμεση πρόσβαση σε περίπτωση που μια υποδομή WiFi είναι διαθέσιμη, αλλά καταναλώνει πολύ περισσότερη ενέργεια, δεδομένου ότι είχε σχεδιαστεί αρχικά για συσκευές όπως οι φορητοί υπολογιστές με πόρους ενέργειας, όχι για μικρές συσκευές IoT. Όταν η μόνη υποδομή δικτύου που προσφέρεται είναι ένα δίκτυο κινητής τηλεφωνίας , η κυτταρική επικοινωνία δεδομένων είναι η μόνη επιλογή .

Πίνακας 1-1. Επισκόπηση των τεχνολογιών ασύρματης επικοινωνίας.

| Τεχνολογία | Ταχύτητα Δεδομένων | Κλίμακα | Συχνότητα |
|-----------------------|----------------------------|----------|--------------------------------|
| WiFi | 54 Mbps | 150m | 5Ghz |
| Bluetooth | 721 Kbps | 10-150m | 2.4GHz ISM |
| RF-Links | 1 Mbps | 50-100m | 2.4GHz ISM |
| IEEE 802.15.4, ZigBee | 250 kbps, 20 kbps, 40 kbps | 100-300m | 2.4GHz ISM, 868MHz, 915MHz ISM |
| Cellular 3G | 14,4 Mbps / 5.8 Mbps | m-km | 800MHz, 1900MHz |
| Wired (Ethernet) | 100 Mbps-1 Gbps | m-km | - |

3.1.18 Τρέχουσα Κατάσταση και Κοντινό Μέλλον

Ποια ήταν η εξέλιξη του Ίντερνετ των πραγμάτων από την αρχική περιγραφή της έννοιας πίσω στο 1999; Η εξέλιξη του Ίντερνετ των πραγμάτων στην καθημερινή μας ζωή εξαρτάται κυρίως από δύο πράγματα: την τεχνική πρόοδο των μικροελεγκτών και των ενσωματωμένων συσκευών και την εξέλιξη των ασύρματων διεπαφών στα πλαίσια της επικοινωνίας και της ενεργειακής αποδοτικότητας.

Όσον αφορά τις υπηρεσίες για τα δίκτυα IoT, η τρέχουσα κατάσταση περιλαμβάνει διάφορες πλατφόρμες που ήδη παρέχουν λειτουργίες όπως η φιλοξενία των αποθετηρίων δεδομένων, οπτικοποίηση των δεδομένων, επικοινωνία με τις συσκευές μέσω των διασυνδέσεων web, απομακρυσμένη εκκίνηση των γεγονότων, κλπ. Όλες αυτές οι πλατφόρμες και οι υπηρεσίες τους, έχουν ειδικά σχεδιαστεί και υλοποιηθεί για τη διευκόλυνση της επικοινωνίας με συσκευές IoT. Αυτό σημαίνει ότι οι λειτουργίες τους έχουν υλοποιηθεί χρησιμοποιώντας ανοικτά και ελαφρά πρωτόκολλα του Διαδικτύου web, που επιτρέπουν την εύκολη και

άμεση επικοινωνία , προσφέροντας την ίδια στιγμή υλοποιήσεις σε πολλές γλώσσες και περιβάλλοντα προγραμματισμού.

Παραδείγματα υπηρεσιών IoT αποτελούν αυτά που θα παρουσιαστούν σε αυτό το βιβλίο , όπως η υπηρεσία δεδομένων Pachube και ο καταγραφέας δεδομένων Nimbits και πολλά άλλα , όπως το ThingSpeak και η iDigiDeviceCloud . Οι υπηρεσίες αυτές επιτρέπουν στους χρήστες να διαχειρίζονται και να απεικονίζουν τα δεδομένα των αισθητήρων τους, είτε δωρεάν ή με χαμηλό κόστος . Πρόσθετες υπηρεσίες , όπως η πρόσφατα διακοπείσα GoogleHealth ή το Microsoft HealthVault συνεργάζονται με τους προμηθευτές της συσκευής για τη διαχείριση των πληροφοριών άμεσα στο δίκτυο . Internet-enabled οθόνες για την πίεση του αίματος, μετρητές σακχάρου , βηματόμετρα , και παρόμοιες συσκευές παρακολούθησης της υγείας είναι διαθέσιμες στην αγορά και επιτρέπουν στο χρήστη να διαχειριστεί τα στοιχεία της υγείας του από τους λογαριασμούς του στο διαδίκτυο.

Το άμεσο όραμα του μέλλοντος : Είναι ήδη εφικτό να συνδέσετε πλατφόρμες μικροελεγκτή , όπως η Arduino , απευθείας στις υπηρεσίες του IoT και να σας αναφέρουν δεδομένα από αισθητήρες και να απαντούν σε ερεθίσματα ρυθμισμένα από τους χρήστες, εξ' αποστάσεως, μέσω web εφαρμογών . Ενσωματωμένες συσκευές είναι ήδη αρκετά μικρές για να είναι ασύρματες και η πρόοδος στον τομέα των επικοινωνιών και των πόρων ενέργειας επιτρέπει ασύρματη σύνδεση στο Internet και συνδεσιμότητα με όλα τα είδη των συσκευών που χρησιμοποιούμε σήμερα . Φανταστείτε να μπορείτε να ελέγχετε τον αυτοματισμό του σπιτιού ή του γραφείου σας με τον τρόπο που βλέπετε τα e-mails σας σήμερα: είτε από το κινητό σας, το laptop σας ή οποιοδήποτε μέρος στο κόσμο με σύνδεση στο Internet . Πιο σημαντικό και πολύ πιο ενδιαφέρον , φανταστείτε ότι την ίδια στιγμή οι συσκευές αυτοματισμού σας θα είναι σε θέση να ανακτήσουν σημαντικές πληροφορίες για τη λειτουργία τους (π.χ.,

ενημέρωση για καιρικές συνθήκες) απευθείας από το Internet, να παρέχουν αυτοματισμούς και να απαιτούν λιγότερες εισροές και την αλληλεπίδραση από εσάς .

Εξερευνήστε τα παραδείγματα αυτού του βιβλίου και να είστε έτοιμοι να «Internetize» πλήρως τους αισθητήρες σας και τους ενεργοποιητές χτίζοντας το δικό σας δίκτυο Ίντερνετ πραγμάτων με τις συσκευές σας !

3.1.19 Περίληψη

Αυτό το κεφάλαιο σας εισάγει στις βασικές έννοιες του Διαδικτύου των πραγμάτων . Πήρατε παραδείγματα των συσκευών IoT , μαζί με πληροφορίες σχετικά με τα βασικά συστατικά (όπως μικροελεγκτές και αισθητήρες) . Μια σύντομη παρουσίαση και σύγκριση μεταξύ των κύριων τεχνολογιών επικοινωνίας (όπως WiFi, Bluetooth, RF, Cellular, ZigBee και Ethernet) μπορεί να σας δώσει ιδέες για το πώς οι συσκευές IoT μπορεί να αλληλεπιδράσουν με το Διαδίκτυο . Το τελευταίο τμήμα του παρόντος κεφαλαίου σας έχει παράσχει πληροφορίες σχετικά με την τρέχουσα κατάσταση στην εξέλιξη του Ίντερνετ των πραγμάτων και σας έχει δώσει ιδέες για το εγγύς μέλλον.

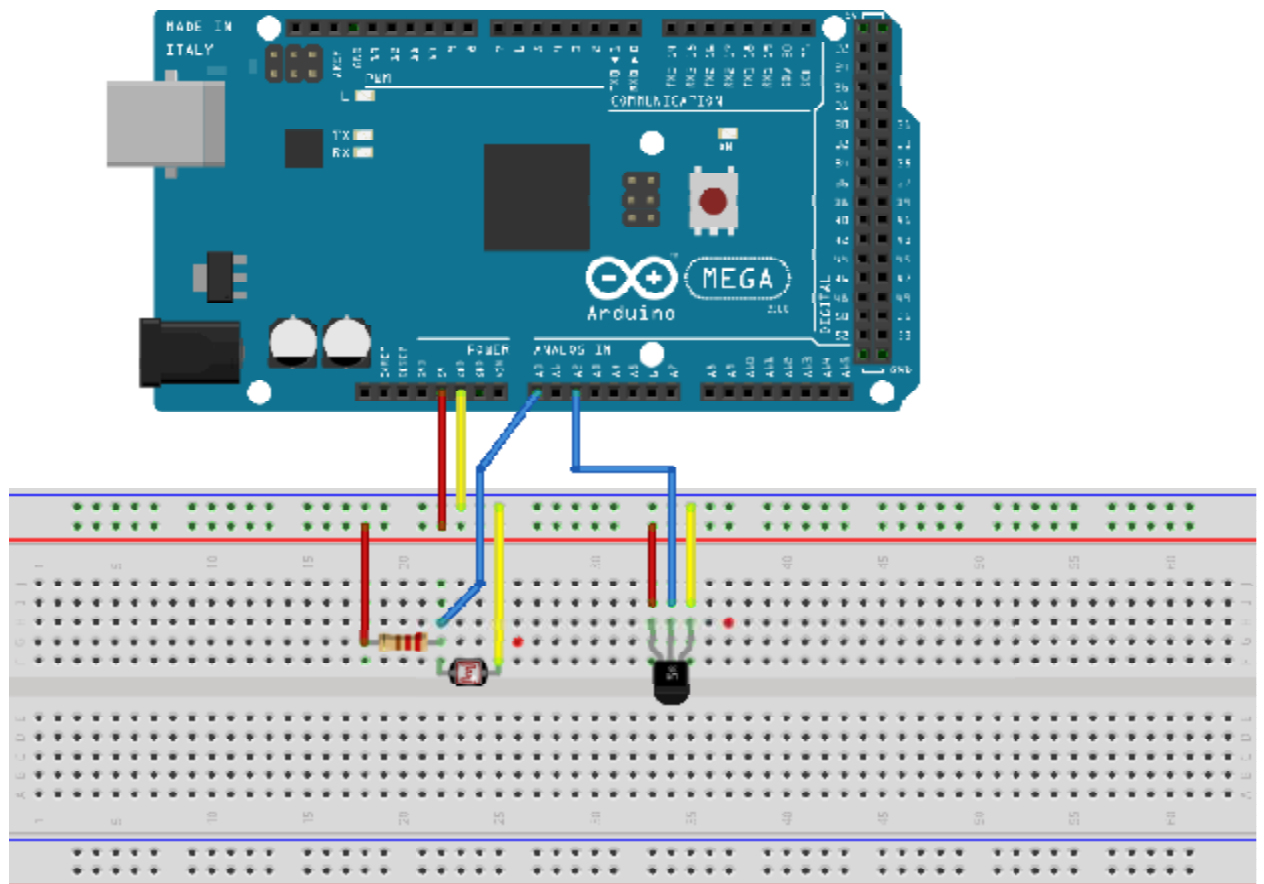
Το επόμενο κεφάλαιο θα σας εξηγήσει πώς χειρίζονται τους αισθητήρες και το είδος των αισθητήρων που μπορείτε να χρησιμοποιήσετε σε μελέτες σας για να «κτίσετε» τη δική σας πλατφόρμα IoT .

ΚΕΦΑΛΑΙΟ 4

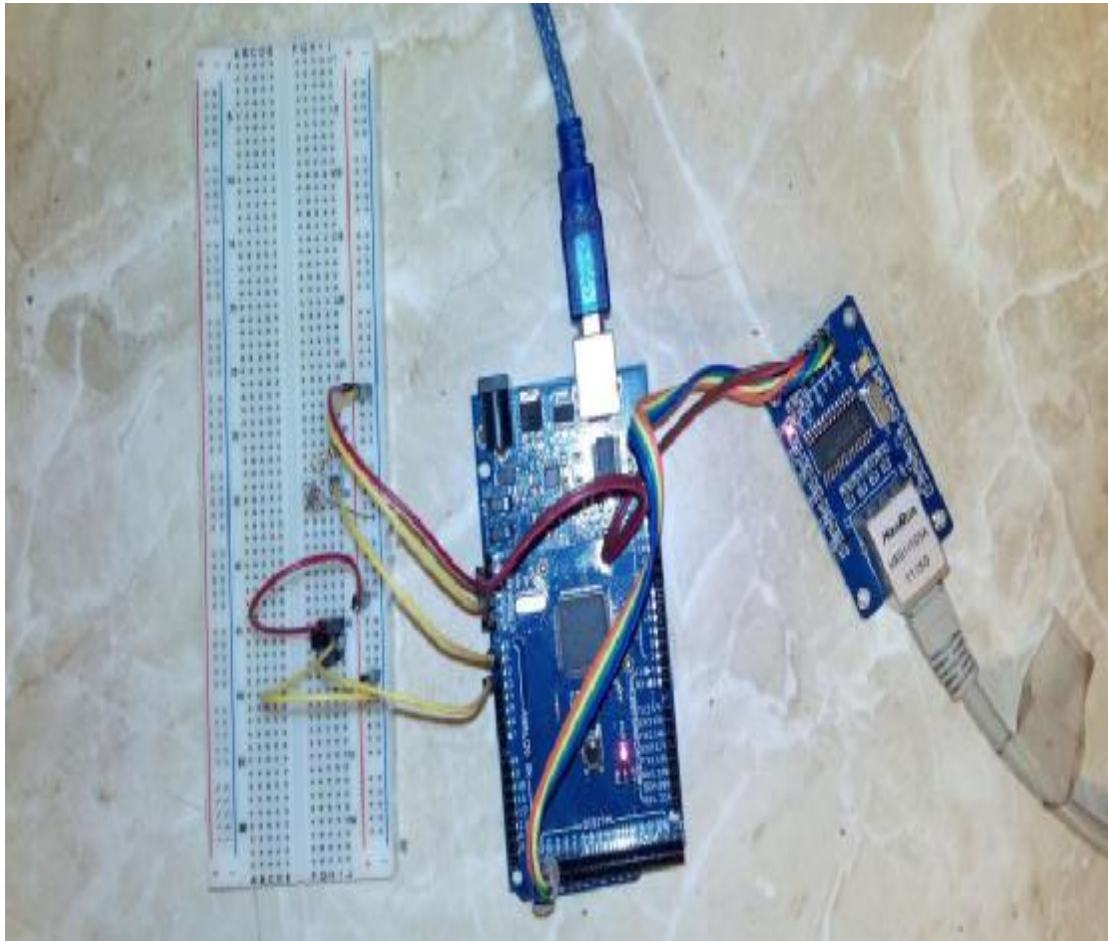
ΥΛΟΠΟΙΗΣΗ

Για την υλοποίηση του κυκλώματος θα χρειάστηκαν τα εξής αντικείμενα:

1. Μια πλακέτα ArduinoMega 1280
2. Μία breadboard
3. Ένας αισθητήρας LM35
4. Ένας αισθητήρας LDR
5. Μία αντίσταση
6. Καλώδια
7. Ένα Ethernet shield



fritzing



```

#include <EtherCard.h>

#define FEED "1278821877"
#define APIKEY "iburHecqDxcEodZkUsPJsfMobqnyJuqb5Q96D1hzlNCjdGtw"

float an0,an1,an2,an3,an4;
inttempPin = 0;

#define STATIC 1 // set to 1 to disable DHCP (adjust myip/gwip values
below)

#if STATIC
// ethernet interface ip address
static byte myip[] = { 194,24,226,111 };
// gateway ip address
static byte gwip[] = { 194,24,226,65 };
// dns server
static byte dnsip[] = { 194,24,226,35 };
#endif

// ethernet interface mac address, must be unique on the LAN
byte mymac[] = { 0x74,0x69,0x69,0x2D,0x30,0x31 };

const char website[] PROGMEM = "api.cosm.com";

byte Ethernet::buffer[700];
uint32_t timer;
Stash stash;

static char statusstr0[10];
static char statusstr1[10];
static char statusstr2[10];
static char statusstr3[10];
static char statusstr4[10];

void setup () {
Serial.begin(57600);
Serial.println("\n[webClient]");

if (ether.begin(sizeof Ethernet::buffer, mymac, 53) == 0)
{
Serial.println("Failed to access Ethernet controller");
}
else
{
Serial.println("Ethernet is fine and up");
}
#if STATIC
ether.staticSetup(myip, gwip, dnsip);
#else
if (!ether.dhcpSetup())
Serial.println("DHCP failed");
#endif

ether.printIp("IP: ", ether.myip);
ether.printIp("GW: ", ether.gwip);
ether.printIp("DNS: ", ether.dnsip);
if (!ether.dnsLookup(website))
Serial.println("DNS failed");

ether.printIp("SRV: ", ether.hisip);
delay(1000);
}

void loop () {
word len = ether.packetReceive();
word pos = ether.packetLoop(len);

if (millis() > timer) {
timer = millis() + 30000;
an0=analogRead(0); //convert the analog data to temperature
delay(250);
an0=analogRead(0);
delay(250);
Serial.println(an0);
an1=analogRead(1); //convert the analog data to temperature

```

```

delay(250);
an1=analogRead(1);
delay(250);
Serial.println(an1);
an2=analogRead(2); //convert the analog data to temperature
delay(250);
an2=analogRead(2);
delay(250);
Serial.println(an2);
an3=analogRead(3); //convert the analog data to temperature
delay(250);
an3=analogRead(3);
delay(250);
Serial.println(an3);
an4=analogRead(4); //convert the analog data to temperature
delay(250);
an4=analogRead(4);
delay(250);
Serial.println(an4);

dtostrf(an0,3,0,statusstr0);
dtostrf(an1,3,0,statusstr1);
dtostrf(an2,3,0,statusstr2);
dtostrf(an3,3,0,statusstr3);
dtostrf(an4,3,0,statusstr4);
// we can determine the size of the generated message ahead of time
byte sd = stash.create();
stash.print("an0,");
stash.println(statusstr0);
stash.print("an1,");
stash.println(statusstr1);
stash.print("an2,");
stash.println(statusstr2);
stash.print("an3,");
stash.println(statusstr3);
stash.print("an4,");
stash.println(statusstr4);
stash.save();

// generate the header with payload - note that the stash size is used,
// and that a "stash descriptor" is passed in as argument using "$H"
Stash::prepare(PSTR("PUT http://\$F/v2/feeds/\$F.csv HTTP/1.0" "\r\n"
"Host: $F" "\r\n"
"X-PachubeApiKey: $F" "\r\n"
"Content-Length: $D" "\r\n"
"\x\n"
"$H"),
website, PSTR(FEED), website, PSTR(APIKEY), stash.size(), sd);

// send the packet - this also releases all stash buffers once done
ether.tcpSend();
}
}

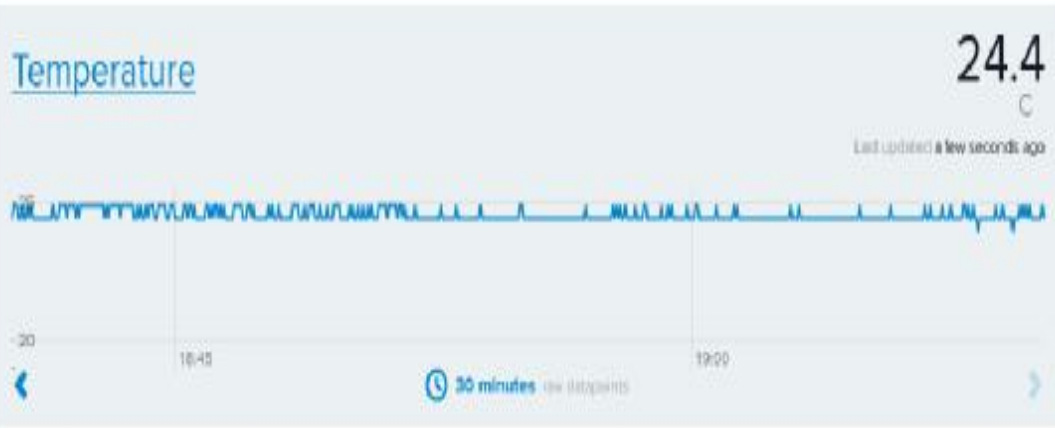
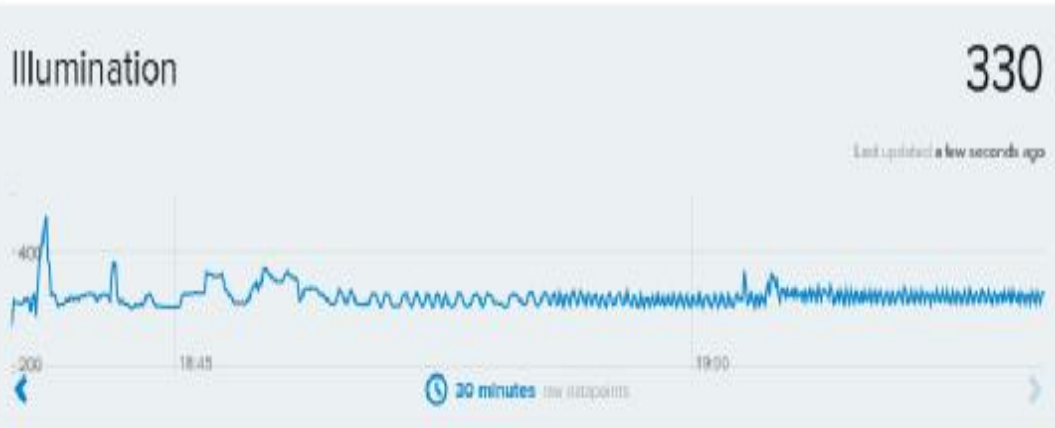
```


Arduino Mega 1280

Feed ID 301642943
API Endpoint <https://api.xively.com/v2/feeds/301642943>

Channels Last updated a few seconds ago

[N](#) Graphs



ΒΙΒΛΙΟΓΡΑΦΕΙΑ

WWW.Arduinocc.com