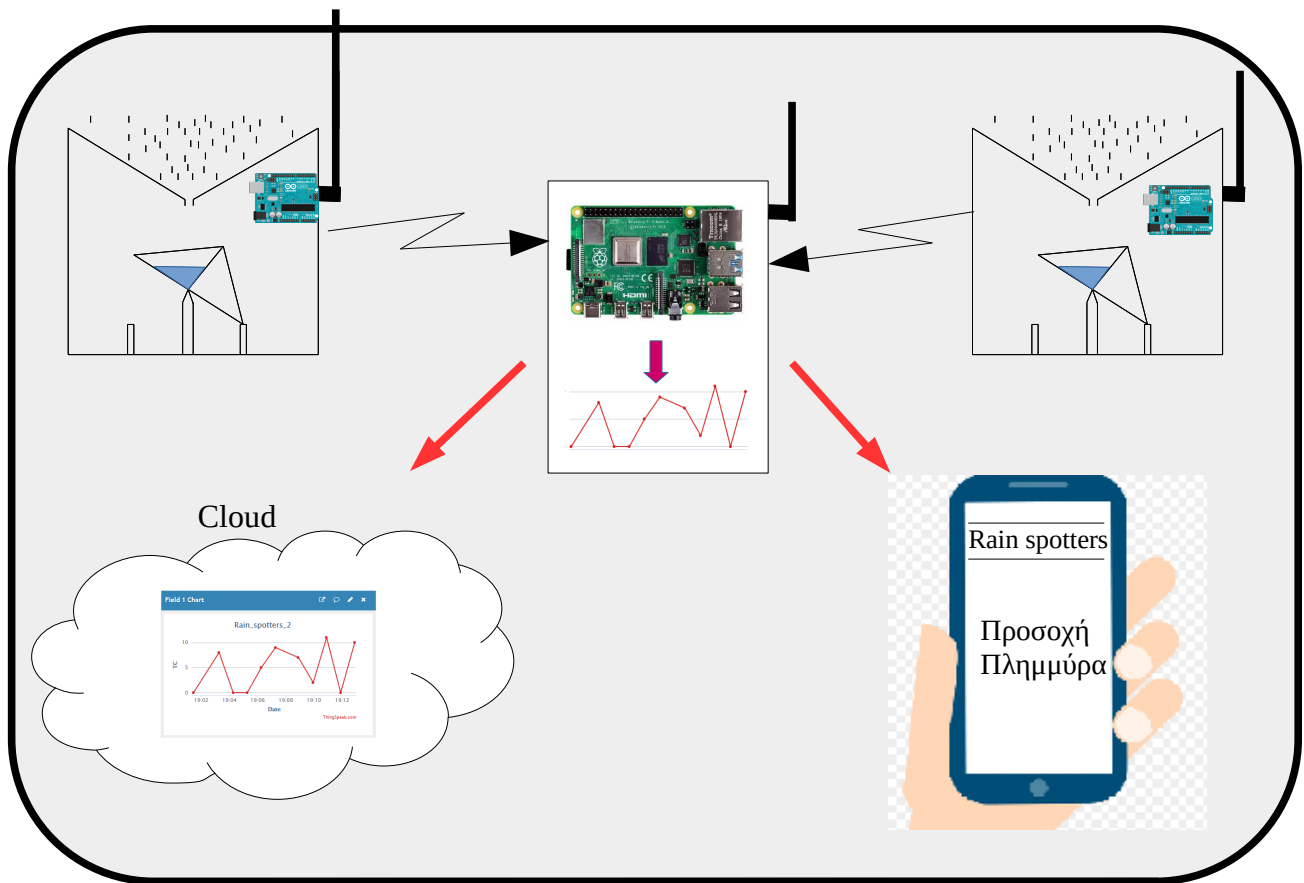


RAIN SPOTTERS



Η ομάδα μας

Μέντορες

Γιάννης Λαγογιάννης (Φυσικός),

Γιώργος Λαγογιάννης (Μηχανικός Η/Υ και Πληροφορικής)

Μαθητές Γυμνασίων Ξυλοκάστρου

Λευτέρης Ηλίας

Γιάννης Παπαχρηστόπουλος

Θοδωρής Λαγογιάννης

Εύδοξος Χασούρας

Δημήτρης Τεάζης

Αργύρης Μουρίκης

Ηλίας Σκούρας

Παναγιώτης Μπονάτσος

Πρόλογος

Η ιδέα μας ήταν να δημιουργήσουμε ένα σύστημα προειδοποίησης πλημμυρών. Ασύρματοι αισθητήρες (τερματικοί κόμβοι) βασισμένοι σε arduino μετράνε το ύψος της βροχής που πέφτει και στέλνουν ανά τακτά χρονικά διαστήματα τις μετρήσεις στους σε ένα κεντρικό σταθμό, βασισμένο σε ένα raspberry pi. Ο κεντρικός σταθμός συγκεντρώνει τα μηνύματα που στέλνουν οι αισθητήρες και σημαίνει συναγερμό αν το ύψος βροχής ξεπεράσει μία ορισμένη τιμή. Στην πορεία είδαμε ότι είναι πολύ εύκολο να στέλνει ο κεντρικός σταθμός τις μετρήσεις σε μία πλατφόρμα στο “σύννεφο” όπου μπορούν όλοι να τις δουν και ενδεχομένως να τις επεξεργαστούν. Αφού ήταν εύκολο, το κάναμε.

Για την υλοποίηση του project μας χρησιμοποιήσαμε ένα πρόσφατο project με τίτλο “*Low-cost LoRa IoT framework*” το οποίο αναπτύχθηκε στα πλαίσια των *EU H2020 WAZIUP/WAZIHUB projects*. Ο κώδικας του project αυτού υπάρχει στο github (μπορείτε να το δείτε κάνοντας CTRL+κλικ [εδώ](#)) και δίνεται ελεύθερος για χρήση από όποιον το επιθυμεί, σύμφωνα με τους όρους της [Γενικής Άδειας Δημόσιας Χρήσης GNU](#). Το project είναι ένα χαμηλού κόστους LoRa Gateway που εγκαθίσταται σε ένα raspberry, ενώ περιέχει και όλα όσα χρειάζεται κανείς για να κατασκευάσει τους τερματικούς κόμβους οι οποίοι επικοινωνούν ασύρματα με τον κεντρικό σταθμό. Ακολουθώντας απλά τις οδηγίες μπορεί κανείς να δημιουργήσει εύκολα τη δική του εφαρμογή. Στη σελίδα του project περιγράφονται πολύ αναλυτικά οι ενέργειες που πρέπει κανείς για να χρησιμοποιήσει τον κώδικα, ενώ δίνονται links για βίντεο που δείχνουν όλα τα βήματα που πρέπει να κάνει κανείς. Για την επίτευξη του στόχου μας έπρεπε κι εμείς να προσθέσουμε δικό μας κώδικα.

Η παρουσίαση του δικού μας project θα γίνει με τη λογική ότι το κείμενο αυτό θα πρέπει να χρησιμεύσει σε κάποιον που θέλει να κάνει και αυτός το project μας, χωρίς να έχει ιδιαίτερες γνώσεις προγραμματισμού. Έτσι, το τελικό αποτέλεσμα παρουσιάζεται μέσα από μικρά βήματα που οδηγούν στον τελικό στόχο. Στο κείμενο παρεμβάλλονται (σκιασμένα) σχόλια των μεντόρων της ομάδας προς τους μαθητές που θα προσπαθήσουν να κάνουν το project.

Το project μας λειτουργεί στο εργαστήριο και όταν ο αισθητήρας μας δεν απέχει μεγάλη απόσταση από τον κεντρικό σταθμό. Για να μπορέσει κανείς να επιτύχει ασύρματη επικοινωνία σε μεγάλες αποστάσεις, χρειάζεται να εγκαταστήσει και στον αισθητήρα και στον κεντρικό κόμβο εξωτερικές κεραίες. Παρόλ’ αυτά, ακόμα και στη μορφή που το υλοποιήσαμε εμείς, μπορεί κανείς να εγκαταστήσει τον αισθητήρα στην ταράτσα ή στην αυλή του σπιτιού του και να λαμβάνει τις μετρήσεις στον κεντρικό σταθμό που θα βρίσκεται στο σπίτι του. Ένα άλλο σημείο στο οποίο θα πρέπει να σταθούμε είναι ότι (λόγω κόστους) υλοποιήσαμε μόνο ένα αισθητήρα. Αυτό δε σημαίνει ότι είναι δύσκολο να προσθέσει κανείς κι άλλους.

Ελπίζουμε να βρείτε το project μας ενδιαφέρον και αν αποφασίσετε να το υλοποιήσετε κι εσείς, ελπίζουμε η παρουσίαση αυτή και όλο το υπόλοιπο υλικό που έχουμε αναρτήσει στη σελίδα του project μας στο github να σας φανούν χρήσιμα και κατατοπιστικά.

Για να μπορούμε να ανιχνεύουμε την κίνηση θα βασιστούμε σε ένα arduino uno και σε ένα reed switch. Το reed switch είναι ένας διακόπτης που κλείνει όταν βρίσκεται κοντά του ένα αρκούντως ισχυρό μαγνητικό πεδίο. Θα τον ονομάζουμε από εδώ και πέρα, *μαγνητικό διακόπτη*. Ο μηχανισμός λειτουργεί ως εξής: Τοποθετούμε ένα μαγνήτη στο σημείο X. Καθώς κινούνται τα δοχεία, κινείται προφανώς και ο μαγνήτης, ο οποίος σε κάποιο σημείο της κίνησής του περνάει πολύ κοντά από το

μαγνητικό διακόπτη. Εκείνη τη στιγμή ο διακόπτης κλείνει και το συμβάν αυτό μπορούμε να το ανιχνεύσουμε προγραμματίζοντας κατάλληλα ένα Arduino. Προφανώς το Arduino πρέπει να τροφοδοτείται με ρεύμα. Για το σκοπό αυτό χρησιμοποιούμε μία μπαταρία 18650, η οποία φορτίζεται όσο υπάρχει ήλιος με τη βοήθεια ενός μικρού φωτοβολταϊκού.

1.3 Η ασύρματη επικοινωνία.

Το arduino εκπέμπει ασύρματα το πλήθος των κινήσεων των δοχείων, μέσω της τεχνολογίας μετάδοσης LoRa [2]. Μέσω της τεχνολογίας αυτής επιτυγχάνεται ασύρματη επικοινωνία σε μεγάλες αποστάσεις (αρκετά χιλιόμετρα) με πολύ μικρή κατανάλωση ενέργειας. Η ταχύτητα επικοινωνίας είναι από 250 bps μέχρι 50 kbps και για την εφαρμογή που σχεδιάζουμε, λίγα kbps είναι υπερ-αρκετά. Τα δεδομένα εκπέμπονται προς πύλες (gateways) οι οποίες στη συνέχεια τα προωθούν συνήθως σε ένα LoRa Server. Στη δική μας περίπτωση γίνεται επεξεργασία των δεδομένων στην πύλη, ενώ ταυτόχρονα τα δεδομένα αποστέλλονται και σε μία cloud πλατφόρμα (thingspeak.com) όπου μπορεί ο καθένας να τα δει.

1.4 Ο “κεντρικός σταθμός”

Το arduino στέλνει ανά τακτά χρονικά διαστήματα το πλήθος των κινήσεων που έχει ανιχνεύσει, σε ένα “κεντρικό σταθμό” (ένα raspberry pi) ο οποίος συγκεντρώνει τις αντίστοιχες τιμές από όλους τους αισθητήρες. Στον κεντρικό σταθμό οι τιμές αποθηκεύονται σε κάποιο αρχείο και μπορούμε να απεικονίσουμε τις τιμές που έχουμε λάβει. Επίσης, ο κεντρικός σταθμός μπορεί να στέλνει τα δεδομένα που λαμβάνει σε cloud πλατφόρμες, όπου τα δεδομένα θα είναι ορατά από όλους.

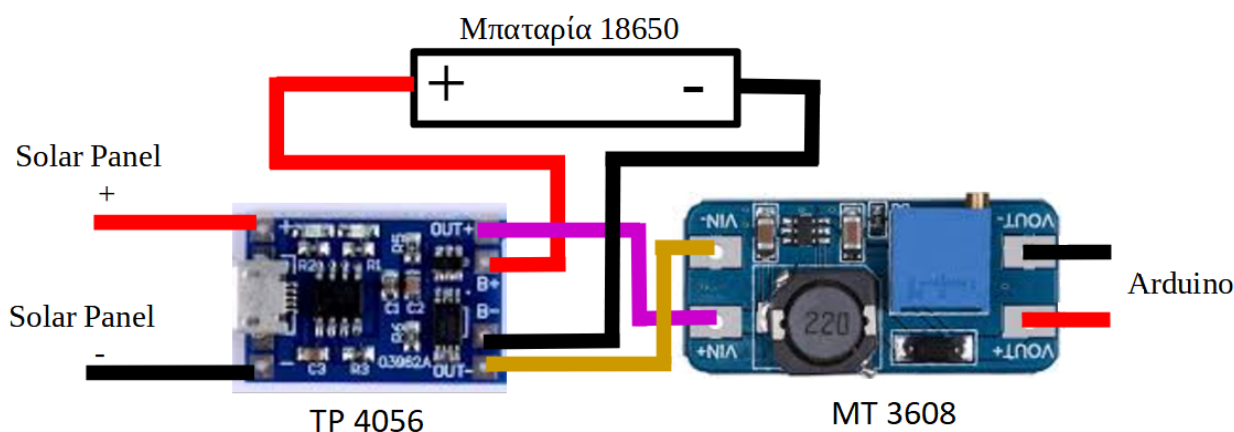
2. Υλοποίηση του αισθητήρα

2.1 Εξασφαλίζοντας την ενέργεια του αισθητήρα.

Ο αισθητήρας μας θα μπορεί να τοποθετηθεί οπουδήποτε, χωρίς την απαίτηση να υπάρχει παροχή ρεύματος. Για το σκοπό αυτό ο αισθητήρας μας έχει μία μπαταρία 18650, καθώς και ένα φωτοβολταϊκό που φορτίζει την μπαταρία όταν υπάρχει ηλιοφάνεια έτσι ώστε να επιτύχουμε τη μεγαλύτερη δυνατή αυτονομία. Για να υλοποιηθεί αυτή η δυνατότητα χρησιμοποιήσαμε:

- Ένα MT3608 Step Up Converter Booster Power Supply ο οποίος μετατρέπει την τάση που δίνει η μπαταρία από περίπου 4 volts σε κάτι παραπάνω από 5 volts (που είναι η ελάχιστη τάση για να λειτουργεί σωστά το arduino).
- Ένα TP 4056 Micro Usb 1a 18650 Lithium Battery Charger. Το κύκλωμα αυτό συνδέεται με το φωτοβολταϊκό, με την μπαταρία και με το MT3608.

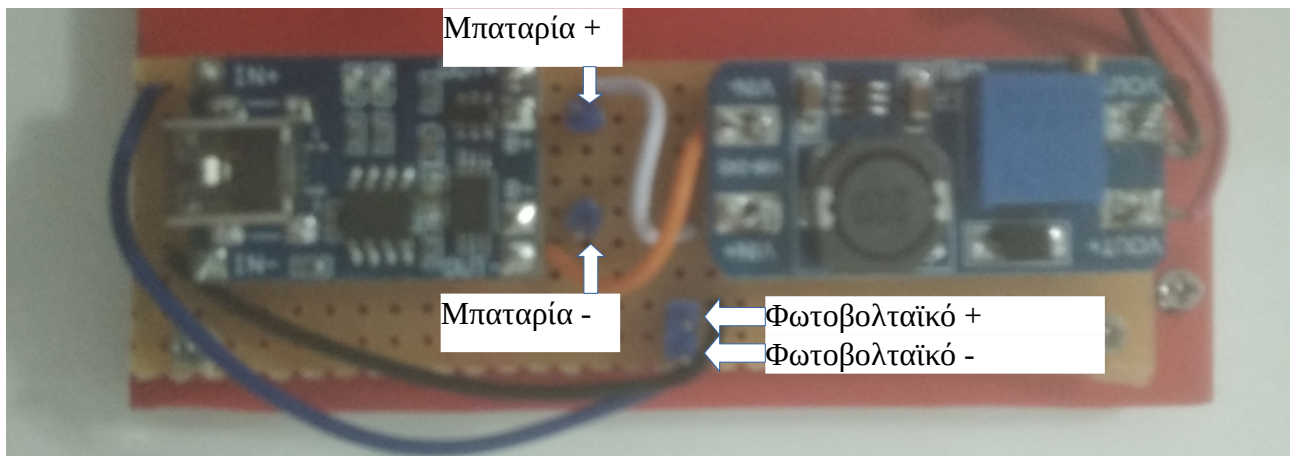
Το σχεδιάγραμμα του “συγκροτήματος παροχής ενέργειας” φαίνεται στην παρακάτω εικόνα



Εικόνα 2: Το "Συγκρότημα παροχής ενέργειας"

Το MT3608 μπορεί να μπερδέψει όσους προσπαθήσουν να το χρησιμοποιήσουν. Η τάση εξόδου θα πρέπει να ρυθμιστεί με ένα κατσαβίδι (απλά γυρίζουμε κάτι που μοιάζει με βίδα προς τη μία ή προς την άλλη κατεύθυνση). Όμως είναι δυνατό να έχει κάνει κανείς πολλές στροφές και η τάση εξόδου να παραμένει αμετάβλητη. Δοκιμάστε μέχρι να φτάσετε σε κάποιο σημείο που η τάση εξόδου έχει αλλάξει, και μετά με μικρές κινήσεις της βίδας προσπαθήστε να πετύχετε τάση εξόδου λίγο πάνω από 5 volts.

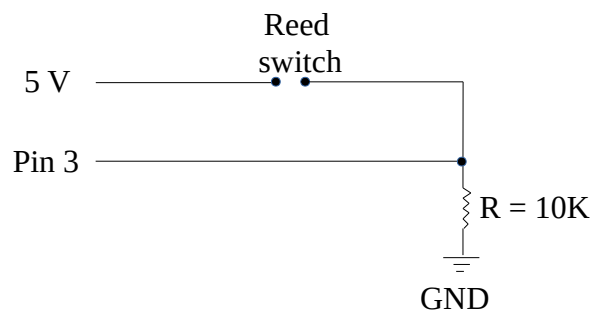
Μπορεί κανείς να υλοποιήσει όλο το “συγκρότημα παροχής ενέργειας” πάνω σε ένα breadboard. Το δοκιμάσαμε κι εμείς αλλά παρατηρήσαμε ότι οι συνδέσεις δεν ήταν τόσο σταθερές και μικρές κινήσεις των καλωδίων δημιουργούσαν πρόβλημα (ίσως να ήταν ελαττωματικό το δικό μας breadboard). Για το λόγο αυτό υλοποιήσαμε το συγκρότημα παροχής ενέργειας σε ένα StripBoard, έπρεπε όμως να κάνουμε κάποιες κολλήσεις με το κολλητήρι μας. Στην εικόνα 3 βλέπετε το StripBoard στο οποίο έχουμε κολλήσει το MT3608 και το TP4056. Στην κάτω πλευρά, που έχει επίστρωση μετάλλου, έχουμε κόψει κάποιες γραμμές αλλά αυτό δε φαίνεται στη φωτογραφία. Το StripBoard κόβεται σχετικά εύκολα με ένα χαρτοκόπτη και γενικά αποδείχτηκε μία καλή επιλογή για το συγκρότημα παροχής ενέργειας.



Εικόνα 3: Το συγκρότημα παροχής ενέργειας;

2.2 Μετρώντας τον αριθμό των μεταβάσεων

Όσα αναφέρουμε σε αυτή την υποενότητα βασίζονται πάνω στο βίντεο [1] που βρήκαμε στο youtube (που όπως είπαμε, το ονομάζουμε **Πηγή 1**)

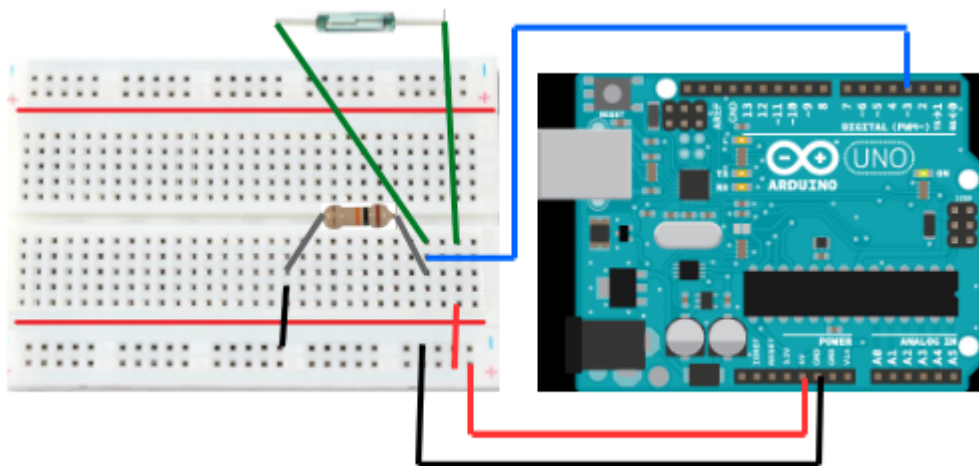


Εικόνα 4: Το βασικό κύκλωμα

Στην Πηγή 1 υπάρχει το κύκλωμα της Εικόνας 4, το οποίο επιτυγχάνει το ζητούμενο. Χρησιμοποιώντας ένα ψηφιακό pin του arduino μπορούμε να ανιχνεύσουμε την κατάσταση LOW, (όταν το δυναμικό του pin είναι ίδιο με το δυναμικό της γείωσης - GND) ή HIGH όταν το δυναμικό του pin είναι ίδιο με το δυναμικό της τροφοδοσίας (5Volts)

Ειδικότερα για τους μαθητές της Α Γυμνασίου, στο σημείο αυτό ενδεχομένως να εισάγονται έννοιες που χρειάζονται αποσαφήνιση από τον καθηγητή φυσικής του σχολείου σας. Είναι μία καλή ευκαιρία να εξοικειωθείτε με τις έννοιες αυτές.

Όταν ο μαγνητικός διακόπτης είναι ανοιχτός, το Pin 3 του Arduino είναι στην κατάσταση LOW, αφού συνδέεται με τη γείωση (GND). Στην περίπτωση αυτή, είτε υπάρχει είτε δεν υπάρχει η αντίσταση R, είναι το ίδιο. Όταν όμως κλείσει ο διακόπτης, ηλεκτρικό ρεύμα αρχίζει να περνάει. Μιας και δεν υπάρχει αντίσταση ανάμεσα στο Pin 3 και στα 5 Volts, το Pin 3 περνάει στην κατάσταση HIGH. Όταν κλείνει ο διακόπτης, η αντίσταση R έχει ένα κρίσιμο ρόλο να παίζει. Αν δεν υπήρχε, το pin 3 θα ήταν συνδεδεμένο ταυτόχρονα και στα 5Volts και στη γείωση. Ένα τέτοιο pin ονομάζεται **floating pin** και η τιμή του είναι απρόβλεπτη (άλλοτε HIGH, άλλοτε LOW).



Εικόνα 5: Το βασικό κύκλωμα, με arduino και breadboard

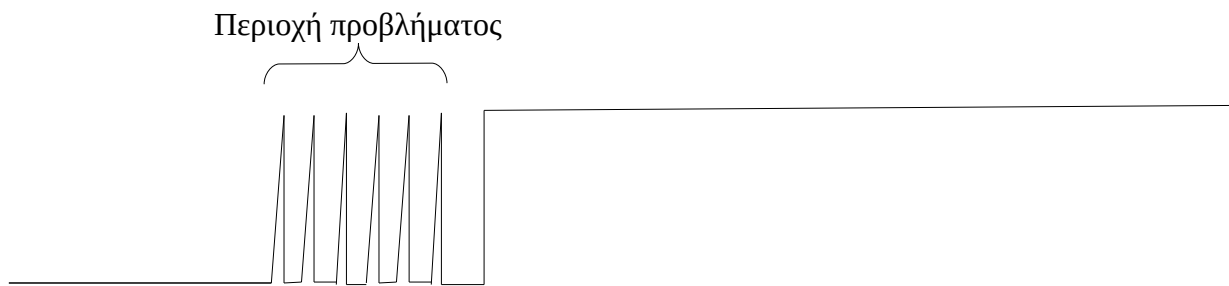
Μπορείτε να υλοποιήσετε στο εργαστήριο το παραπάνω κύκλωμα με τη βοήθεια ενός breadboard, μίας αντίστασης 10K, ενός μαγνητικού διακόπτη και μερικών καλωδίων. Σε περίπτωση μάλιστα που δεν έχετε ξαναφτιάξει κύκλωμα σε breadboard, το παραπάνω κύκλωμα είναι ιδανικό για να ξεκινήσετε αφού είναι ιδιαίτερα απλό. Μπορείτε να το δείτε στην Εικόνα 5. Τροφοδοσία και γείωση μπορείτε να πάρετε από το arduino.

Το πρόγραμμα που πρέπει να φορτώσουμε στο arduino έχοντας υλοποιήσει το κύκλωμα της Εικόνας 5, είναι το `rain_gauge_0` το οποίο υπάρχει στο αποθετήριο μας στο Github (εναλλακτικά κάνετε CTRL + κλικ [εδώ](#)). Είναι ιδιαίτερα απλοϊκό. Κάθε φορά που ξεκινάει η επανάληψη διαβάζουμε την κατάσταση του pin 3. Αν η κατάσταση είναι 1 (high) ενώ η προηγούμενη κατάσταση (αυτή δηλαδή που διαβάσαμε στην προηγούμενη επανάληψη ήταν 0 (low), τότε έχουμε ανιχνεύσει μία μετάβαση και άρα αυξάνουμε το μετρητή μας κατά 1. Στο τέλος της επανάληψης αποθηκεύουμε την κατάσταση του pin 3 που μόλις διαβάσαμε, σαν “προηγούμενη κατάσταση”, μιας και την επόμενη φορά που θα μπούμε στην επανάληψη θα διαβάσουμε την καινούργια τρέχουσα κατάσταση. Πλησιάζοντας ένα μαγνήτη κοντά στο μαγνητικό διακόπτη, παρατηρούμε ότι το κύκλωμα λειτουργεί, αλλά κάποιες φορές ο μετρητής αυξάνεται περισσότερες από μία φορές για κάθε πλησίασμα του μαγνήτη. Είναι κάτι που σίγουρα δε μας αρέσει γιατί θέλουμε ο αισθητήρας μας να είναι ακριβής.

Μπορείτε κι εσείς να φτιάξετε το κύκλωμα και να φορτώσετε το πρόγραμμα `rain_gauge_0` στο arduino σας, και να δείτε τι συμβαίνει

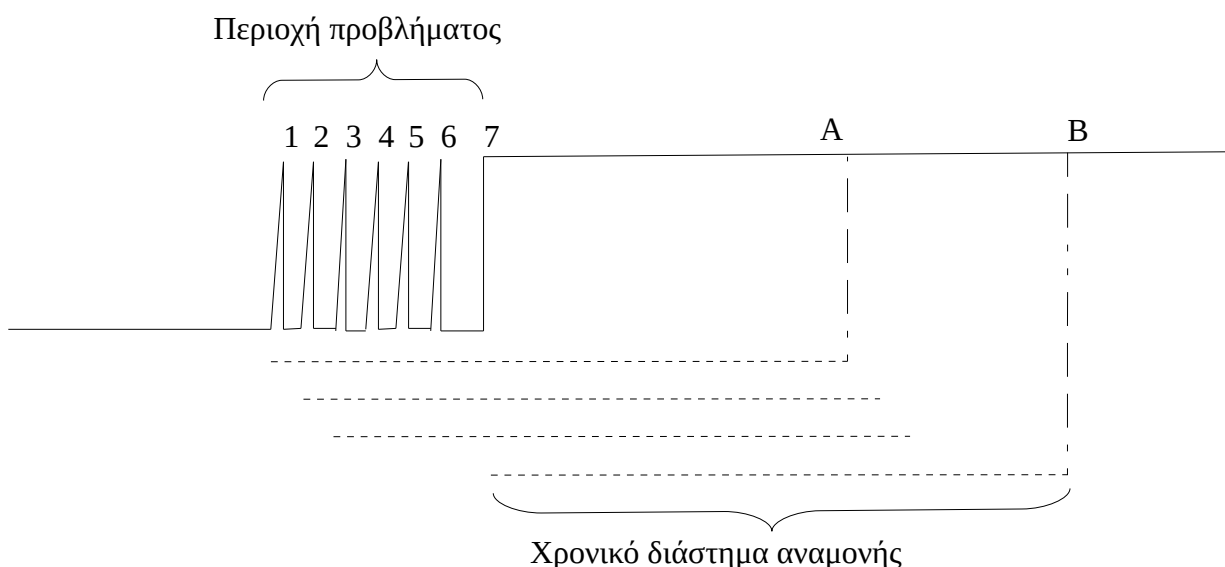
Το πρόβλημα, όπως περιγράφεται και στην Πηγή 1, είναι ο μαγνητικός διακόπτης είναι κάπως ελαττωματικός (όχι μόνο ο δικός μας, αλλά όλοι). Πιο συγκεκριμένα όπως βλέπουμε και στην Εικόνα 6, την ώρα που πλησιάζει ο μαγνήτης, ο μαγνητικός διακόπτης “τρεμοπαίζει” ανοιγοκλείνοντας, στιγμιαία και για ένα πολύ μικρό χρονικό διάστημα, μέχρι να κλείσει κάποια στιγμή σταθερά. Προφανώς δεν πρέπει να αυξάνουμε το μετρητή μας κάθε φορά που βρίσκουμε το pin 3 σε κατάσταση HIGH, αλλά κάθε φορά που ανιχνεύουμε μετάβαση από LOW σε HIGH. Αυτό όμως σημαίνει ότι στην Εικόνα 6, είναι δυνατό να ανιχνεύσουμε 7 μεταβάσεις, και συνεπώς να αυξήσουμε το μετρητή μας κατά 7, ενώ στην πραγματικότητα θα έπρεπε να τον αυξήσουμε κατά 1.

Μπορούμε θεωρητικά να λύσουμε το παραπάνω πρόβλημα, με το πρόγραμμα [rain_gauge1.ino](#). Η ιδέα είναι να περιμένουμε ένα χρονικό διάστημα από τη στιγμή που ανιχνεύτηκε ότι το pin 3 έγινε HIGH, για να αυξήσουμε το μετρητή μας. Το χρονικό αυτό διάστημα πρέπει να είναι τέτοιο ώστε με δεδομένη την ταχύτητα προσέγγισης του μαγνήτη, να προσπεράσουμε το “τρέμουλο” του μαγνητικού διακόπτη.



Εικόνα 6: Ο πραγματικός τρόπος με τον οποίο λειτουργεί ο μαγνητικός διακόπτης.

Ας δούμε όμως, με τη βοήθεια της Εικόνας 7, το πρόγραμμα αυτό λίγο πιο αναλυτικά, για να εξηγήσουμε τη λειτουργία του. Στη συνάρτηση `loop()` αρχικά διαβάζουμε την κατάσταση του `pin 3`. Θέλουμε να μπορούμε να αναγνωρίζουμε το συμβάν “κινήθηκαν τα δοχεία” χωρίς να υπάρχει η πιθανότητα να μετρήσουμε περισσότερες από μία φορές την ίδια κίνηση. Για να λύσουμε το πρόβλημα αυτό, δεν ανιχνεύουμε απλές μεταβάσεις από LOW σε HIGH, αλλά μεταβάσεις από LOW σε HIGH όπου η κατάσταση παραμένει HIGH μετά τη μετάβαση για ένα “χρονικό διάστημα αναμονής”, που το επιλέγουμε έτσι ώστε να μας οδηγήσει στην περιοχή όπου ο μαγνητικός διακόπτης έχει σταθεροποιηθεί. Το πόσο είναι αυτό το χρονικό διάστημα αναμονής μπορούμε να το βρούμε με πειράματα. Στη θέση μνήμης `previous_time` αποθηκεύουμε την τελευταία χρονική στιγμή που ανιχνεύσαμε LOW. Έτσι κάθε φορά που ανιχνεύουμε LOW, αποθηκεύουμε την τιμή του χρόνου (που τη δίνει η συνάρτηση `millis()`) στη θέση μνήμης `previous_time`. Κάθε φορά που ανιχνεύουμε HIGH ελέγχουμε πόσος χρόνος πέρασε από την τελευταία χρονική στιγμή που ανιχνεύσαμε LOW (η οποία όπως είπαμε είναι αποθηκευμένη στη θέση μνήμης `previous_time`). Αν ο χρόνος που πέρασε είναι μικρότερος από το χρονικό διάστημα αναμονής, τότε δεν κάνουμε τίποτα αφού δεν έχουμε ακόμη ανιχνεύσει το συμβάν που θέλουμε. Για παράδειγμα, μπορεί η θέση μνήμης `previous_time` να περιέχει μία χρονική στιγμή λίγο πριν από το σημείο 1, και να βρισκόμαστε τώρα σε κάποιο από τα σημεία 2, 3, 4, 5 ή 6. προφανώς στην περίπτωση αυτή δε θέλουμε να αυξήσουμε το μετρητή αφού αυτός θα αυξηθεί ξανά όταν γίνει η τελική μετάβαση (όταν δηλαδή σταθεροποιηθεί ο μαγνητικός διακόπτης). Αν όμως ο χρόνος που πέρασε είναι τουλάχιστον ίσος με το χρονικό διάστημα αναμονής, τότε ξέρουμε ότι το switch έχει σταθεροποιηθεί, και τότε ξέρουμε ότι έχουμε ανιχνεύσει μία μετάβαση. Στην Εικόνα 7, αυτό θα



Εικόνα 7: Ορίζοντας κατάλληλα το χρονικό διάστημα αναμονής, μπορούμε να προσπεράσουμε την περιοχή του προβλήματος και να αυξήσουμε το μετρητή μας μία φορά, κάπου ανάμεσα στα σημεία A και B

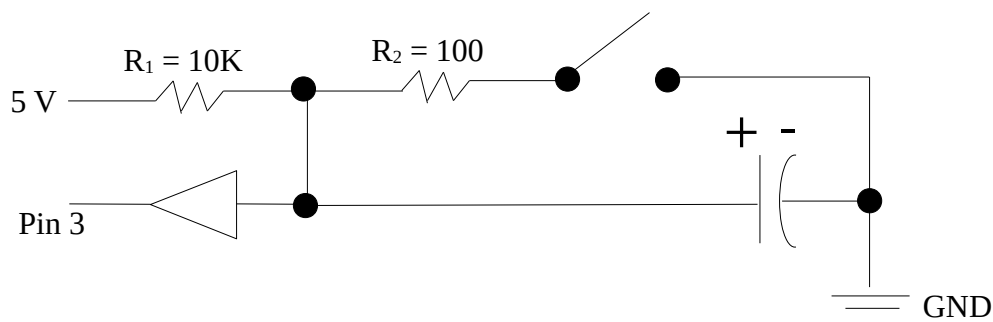
συμβεί κάπου μέσα στην περιοχή ανάμεσα στα σημεία A (αν ξεκινήσαμε να μετράμε από το σημείο 1 και B (αν ξεκινήσαμε να μετράμε από το σημείο 7).

Αφήνουμε για το τέλος, τη θέση μνήμης flag. Δοκιμάστε να εκτελέσετε το πρόγραμμα χωρίς αυτή. Τι παρατηρείτε; μπορείτε τώρα να καταλάβετε τι ρόλο παίζει στο πρόγραμμά μας; Θα μπορούσατε να ρωτήσετε και τον καθηγητή πληροφορικής του σχολείου σας.

Οι δοκιμές έδειξαν ότι το πρόβλημα δε λύθηκε απολύτως. Ανάλογα με την ταχύτητα κίνησης του μαγνήτη, μπορούσε να αυξάνεται 2 φορές. Βελτιστοποιώντας την κίνηση του μαγνήτη έτσι ώστε αυτή να προσομοιάζει όσο γίνεται περισσότερο με την κίνηση που έχει μέσα στον αισθητήρα, παρατηρήσαμε ότι θέτοντας το χρονικό διάστημα αναμονής σε 350 msecs πετύχαμε να αυξάνεται ο μετρητής μία μόνο φορά για κάθε μετάβαση του μαγνήτη. Δε μας φάνηκε όμως η λύση αυτή τόσο αξιόπιστη. Δεν επιμείναμε, και προχωρήσαμε με ένα άλλο κύκλωμα που αναφέρεται στην Πηγή 1.

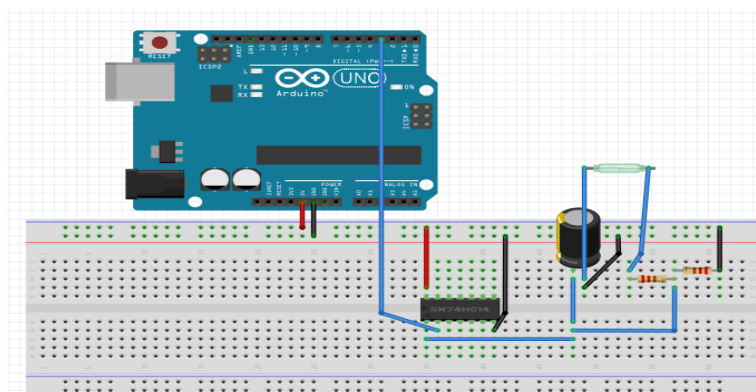
Το τελικό μας κύκλωμα, που κι αυτό όπως είπαμε το πήραμε από την Πηγή 1, φαίνεται στην Εικόνα 8. Σε σχέση με το κύκλωμα της Εικόνας 4, περιέχει επιπλέον έναν αντιστροφέα (το τρίγωνο) και ένα πυκνωτή (οι δύο γραμμές – μία ίσια και μία καμπύλη). Ο αντιστροφέας μετατρέπει το HIGH σε LOW και το LOW σε HIGH. Ο πυκνωτής “απορροφάει” το “τρέμουλο” του μαγνητικού διακόπτη, επιτρέποντας να εμφανίζεται στην έξοδο μία μόνο μετάβαση, από HIGH σε LOW (θυμηθείτε ότι υπάρχει ο αντιστροφέας).

Ζητήστε από τον καθηγητή φυσικής του σχολείου σας να σας εξηγήσει τη λειτουργία του τελικού κυκλώματος, και το ρόλο που παίζει στη λειτουργία αυτή ο πυκνωτής.



Εικόνα 8: Το τελικό κύκλωμα

Στην Εικόνα 9 βλέπετε το τελικό κύκλωμα σχηματικά, πάνω σε breadbord. Το τσιπάκι περιέχει 6 αντιστροφείς ενώ χρειάζεται τροφοδοσία και γείωση (μπορείτε να ψάξετε για τις απαιτούμενες πληροφορίες στον παγκόσμιο ιστό, χρησιμοποιώντας το όνομα του τσιπ. Οι αντιστάσεις που θα χρησιμοποιήσετε δεν έχουν τα ίδια χρώματα με αυτές του σχήματος. Χρησιμοποιήσαμε ένα πυκνωτή χωρητικότητας 22 μ F.



Εικόνα 9: το τελικό κύκλωμα πάνω σε breadboard (η εικόνα δημιουργήθηκε με τη βοήθεια του προγράμματος fritzing). Στον πυκνωτή που θα χρησιμοποιήσετε, η θετική πλάκα αντιστοιχεί στο πιο μακρύ ποδαράκι

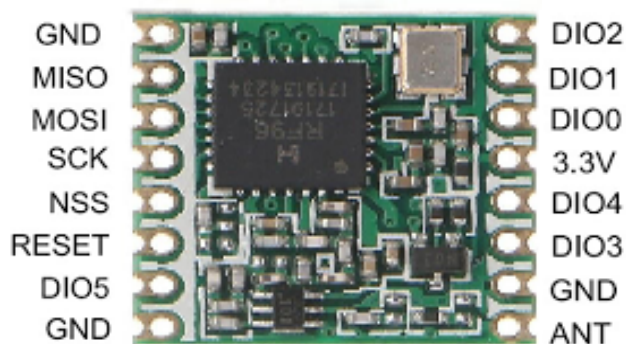
Το κύκλωμα της Εικόνας 8 μπορεί να γίνει και με λιγότερα καλώδια. Μπορείτε να φτιάξετε μία πιο οικονομική (σε καλώδια) έκδοσή του; ρωτήστε τον καθηγητή του σχολείου σας τι ρόλο παίζει η χωρητικότητα του πυκνωτή. Μπορεί το κύκλωμά μας να μη λειτουργεί λόγω της πολύ μικρής (ή μεγάλης) χωρητικότητας του πυκνωτή;

Εφόσον με το κύκλωμα αυτό έχουμε εξαφανίσει το “τρέμουλο” του μαγνητικού διακόπτη, το πρόγραμμα `rain_gauge_0`) το πρώτο που είδαμε θα πρέπει τώρα να μετράει με ακρίβεια τις μεταβάσεις. Οι δοκιμές που κάναμε έδειξαν ότι αυτό ισχύει. Άρα καταλήγουμε στο κύκλωμα της Εικόνας 8 και στο πρόγραμμα `rain_gauge_0`.

Μπορεί κανείς να κάνει το κύκλωμα σε breadboard, αλλά η επιλογή αυτή δε μας φάνηκε και τόσο ασφαλής, Αποφασίσαμε έτσι να δημιουργήσουμε το δικό μας `arduino shield`, που περιέχει το παραπάνω κύκλωμα και το `Lora module`. Όμως όταν το αποφασίσαμε αυτό, είχαμε ήδη δημιουργήσει μία μικρή πλακέτα με το `Lora module` γι, αυτό θα συνεχίσουμε περιγράφοντας πως δημιουργήσαμε αυτό την πλακέτα.

2.3 Μονάδα LORA (ασύρματη επικοινωνία)

Για να μπορεί το `arduino` να μεταδώσει ασύρματα την τιμή του μετρητή χρειάζεται ένα `LoRa module`. Εμείς χρησιμοποιήσαμε το `RFM96w` το οποίο χρησιμοποιεί τη συχνότητα 433 MHz. Όταν το αγοράσαμε κόστιζε περίπου 10 ευρώ. Το τίμημα που πληρώνει κανείς γι, αυτή τη χαμηλή τιμή, είναι ότι πρέπει να φτιάξει μόνος του το pcb έτσι ώστε να “βισματώνει” καλώδια.



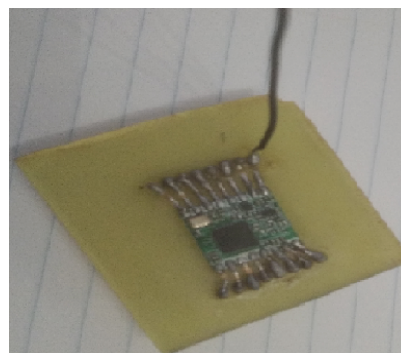
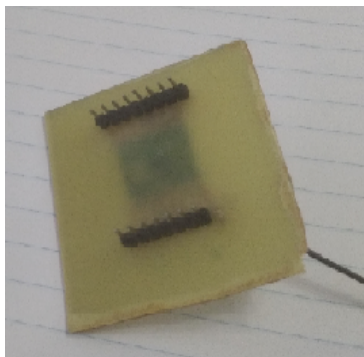
Εικόνα 10: Το module `RFM96w`. Αν και στην Εικόνα το βλέπετε μεγάλο, οι διαστάσεις του είναι 15mm x 17mm

Στη μορφή λοιπόν που το αγοράζουμε δεν είναι εύκολα αξιοποιήσιμο. Η μόνη λύση που έχουμε είναι να δημιουργήσουμε μία δική μας πλακέτα και να το ενσωματώσουμε στην πλακέτα αυτή. Υπάρχει στο εμπόριο μία τέτοια πλακέτα. Η διαδικασία φαίνεται [εδώ](#), [εδώ](#) και [εδώ](#). Εμείς το κάναμε λίγο πιο απλά, αλλά και πάλι δε μπορεί κανείς αποφύγει τη χρήση υλικών και εργαλείων **που θα μπορούσαν να χαρακτηριστούν επικίνδυνα**. Θα περιγράψουμε εδώ τη διαδικασία, θα πρέπει όμως να έχετε στο νου σας ότι δε μπορείτε να κάνετε όλα τα βήματα. Κάποια από αυτά θα πρέπει να γίνουν από έναν ενήλικα με σχετικές γνώσεις.

Ξεκινάμε τυπώνοντας το pdf αρχείο που βλέπετε [εδώ](#). Το αρχείο περιέχει την “εικόνα” του κυκλώματος. Από ένα pcb μία όψης, κόβουμε ένα κομμάτι 4cm x 4cm. Κόβουμε με ένα ψαλίδι το κομμάτι χαρτί που μόλις τυπώσαμε, έτσι ώστε να μας μείνει ένα κομμάτι χαρτιού 4cm x 4cm στο κέντρο του οποίου βρίσκεται η εικόνα του κυκλώματος. Κολλάμε το χαρτί αυτό πάνω στο pcb που κόψαμε. Υπάρχουν πολλοί τρόποι για να μεταφέρουμε την εικόνα του κυκλώματος πάνω στο pcb.

Όμως επειδή το κύκλωμα είναι απλοϊκό, και θέλουμε να λιγοστεύσουμε όσο γίνεται τα υλικά που χρειαζόμαστε, θα δουλέψουμε λίγο “αντισυμβατικά”. Βλέποντας το χαρτί που έχουμε κολλήσει στο pcb, βλέπουμε 16 γραμμές που ξεκινούν από τα “ποδαράκια” του RFm96w και ανοίγονται προς τα έξω, έτσι ώστε να δημιουργηθούν οι σωστές αποστάσεις ανάμεσα στα ποδαράκια. Τα 16 εξωτερικά σημεία (εκεί που τελειώνουν οι γραμμές) θα πρέπει να τρυπηθούν. Προς το παρόν με ένα σφυράκι και ένα καρφί, χτυπάτε το pcb στα άκρα κάθε γραμμής, έτσι ώστε το σημείο που χτυπήσατε, να είναι ορατό αφού αφαιρέσετε το χαρτί από το pcb. Αφαιρείτε το χαρτί και ήρθε η ώρα να τρυπήσετε τα 16 εξωτερικά σημεία. Τη δουλειά αυτή θα πρέπει ή να την αναθέσετε σε κάποιον ενήλικα, ή να χρησιμοποιήσετε κάποιο “αθώο” εργαλείο όπως αυτό που βλέπετε στο βίντεο. Αφού γίνουν οι τρύπες, θα ζωγραφίσετε το κύκλωμα (όπως το βλέπετε στην εικόνα που τυπώσατε) πάνω στο pcb με τη βοήθεια ενός permanent marker. Μπορείτε να χρησιμοποιήσετε και ένα χάρακα για να ζωγραφίσετε τις γραμμές. Φροντίστε ώστε οι γραμμές να είναι παχιές (όσο παχιές τις βλέπετε και στην εικόνα που τυπώσατε ή και πιο παχιές), αρκεί να μην πλησιάζουν πολύ οι γραμμές μεταξύ τους καθώς πλησιάζουν στο RFM96w. Όταν τελειώστε με τη ζωγραφική, πάρτε ένα πλαστικό μπουλ και βάλτε μέσα κάτι έτσι ώστε να μπορείτε να τοποθετήσετε πάνω του το pcb (δε θέλουμε το pcb να ακουμπάει στον πάτο του μπουλ. Προσέξτε γιατί το μελάνι δεν έχει στερεοποιηθεί καλά και αν το ακουμπήσετε μπορεί να καταστρέψετε τη ζωγραφιά σας. Στη συνέχεια ρίξτε στο μπουλ μία μικρή ποσότητα ασετόν (η ποσότητα που χρειάζεται είναι μικρή, περίπου 1 καπάκι από μπουκάλι νερού). Το ασετόν μπορείτε να το βρείτε εύκολα και δεν θεωρείται επικίνδυνο . Μπορεί να υπάρχει και στο σπίτι σας, ενώ σε ένα κατάστημα καλλυντικών μπορείτε να αγοράσετε με 1 ευρώ ποσότητα για να φτιάξετε πάρα πολλές πλακέτες. Αφού βάλετε το ασετόν στο μπουλ, κλείστε το μπουλ από πάνω και περιμένετε μερικά λεπτά. Στη συνέχεια ανοίξτε το μπουλ και βγάλτε το pcb. Το μελάνι στο pcb έχει τώρα σκληρύνει. Από εδώ και πέρα οι ενέργειες θα πρέπει να γίνουν από έναν ενήλικα, ή υπό την επίβλεψη ενός ενήλικα.

Το επόμενο βήμα είναι βάλουμε το pcb σε οξύ. Σε ένα πλαστικό μπουλ ρίχνουμε το pcb και λίγο νερό, έτσι ώστε να καλυφθεί το pcb. Στη συνέχεια ρίχνουμε στο νερό μία κουταλιά του γλυκού τριχλωριούχο σίδηρο, και ανακατεύουμε προσεκτικά με ένα πλαστικό κουταλάκι, με προσοχή να μην ακουμπάμε το pcb (μπορούμε να ρίξουμε το pcb μέσα αφού διαλυθεί ο τριχλωριούχος σίδηρος στο νερό). Είναι απαραίτητο για την προστασία σας να φοράτε πλαστικά γάντια που υπάρχουν γι, αυτό το σκοπό. Αφήνουμε τον τριχλωριούχο σίδηρο να κάνει τη δουλειά του, να φάει δηλαδή το χαλκό, εκεί που έρχεται σε επαφή μαζί του (δε μπορεί όμως να έρθει σε επαφή με το χαλκό εκεί που ο χαλκός προστατεύεται από ένα στρώμα μελανιού συνεπώς ο χαλκός κάτω από τις γραμμές που ζωγραφίσατε θα επιβιώσει). Καλό είναι με κάποιο τρόπο το μπουλ να κουνιέται λίγο, γιατί τότε η δουλειά γίνεται πολύ πιο γρήγορα. Ανά τακτά χρονικά διαστήματα (ας πούμε ανά δεκάλεπτο), με το πλαστικό κουταλάκι πιάνετε προσεκτικά το pcb και το σηκώνετε για να δείτε αν έχει φαγωθεί ο χαλκός. Όταν δείτε ότι φαγώθηκε ο χαλκός, παίρνετε το pcb και του ρίχνετε άφθονο νερό.



Εικόνα 11: Το LoRa module μετά τις εργασίες που περιγράφηκαν παραπάνω

Επόμενο βήμα είναι να κολλήσουμε το RFM96w πάνω στην πλακέτα μας. Για τη δουλειά αυτή χρειάζεται κολλητήρι. Αν και η δουλειά είναι απλή, οι υψηλές θερμοκρασίες δημιουργούν **κίνδυνο για έγκαιμα** ή ακόμη **και πυρκαγιά**, αν συμβεί απροσεξία. Τέλος, πρέπει να κολλήσουμε τα pin headers, και την κεραία. Και αυτή η δουλειά γίνεται με το κολλητήρι. Φτιάξαμε και κολλήσαμε μία

απλή κεραία σύμφωνα με τις οδηγίες που βρήκαμε [εδώ](#) χρησιμοποιώντας μονόκλωνο καλώδιο διατομής 0.5 mm. Είναι ό,τι απλούστερο μπορεί να κάνει κανείς, ίσως όμως όχι ό,τι καλύτερο. Στην Εικόνα 11 βλέπετε το module μετά την παραπάνω διαδικασία.

Στη συνέχεια πρέπει να συνδέσουμε το Lora module με το arduino μας, κάτι που περιγράφεται σε [αυτό το βίντεο](#) (μέχρι το 7:24). Αν και στο βίντεο δε χρησιμοποιείται arduino uno (το οποίο χρησιμοποιήσαμε εμείς, τίποτα δεν αλλάζει αν χρησιμοποιήσει κανείς arduino uno (όπως κάναμε εμείς) Οι συνδέσεις έχουν ως εξής:

Arduino	Radio module
GND-----	GND (ground in)
3V3-----	3.3V (3.3V in)
SS pin D10-----	NSS(CS chip select in)
SCK pin D13-----	SCK (SPI clock in)
MOSI pin D11-----	MOSI (SPI Data in)
MISO pin D12-----	MISO (SPI Data out)

Οι ανάγκη να γίνουν οι παραπάνω συνδέσεις ήταν αυτό που μας ανάγκασε να φτιάξουμε τη δική μας πλακέτα για το LoRa module. Τώρα, μπορούμε πολύ εύκολα να κάνουμε τις συνδέσεις αυτές χρησιμοποιώντας καλώδια, όπως ακριβώς είδατε και στο βίντεο. Όταν γίνουν αυτές οι συνδέσεις, το hardware κομμάτι του αισθητήρα έχει τελειώσει..

Ασύρματη επικοινωνία αισθητήρα

Σειρά έχει το software κομμάτι, που περιγράφεται από το σημείο 7:24 και μετά. Το ολοκληρωμένο hardware φαίνεται στην ακόλουθη εικόνα.

Όπως αναφέρεται και στο βίντεο, αρχικά πρέπει να πάμε σε [αυτή τη διεύθυνση](#) (εκεί δηλαδή που περιέχεται όλο το project) και να το κατεβάσουμε στην υπολογιστή μας σε συμπιεσμένη μορφή. Στη συνέχεια το βίντεο αναφέρει ότι πρέπει να κατεβάσουμε σε συμπιεσμένη μορφή και το project που βρίσκεται [εδώ](#). Αυτό το δεύτερο “κατέβασμα” χρειάζεται αν θέλουμε να βάζουμε το arduino μας σε να λειτουργεί μόνο με την ελάχιστη απαιτούμενη ενέργεια όσο χρόνο δεν κάνει τίποτα. Εμείς δεν ασχοληθήκαμε με αυτό, γιατί κρίναμε ότι δεν ταιριάζει στην εφαρμογή μας. Κι’ αυτό γιατί το arduino μας πρέπει να πάντα ελέγχει για τυχόν κίνηση των δοχείων, δεν έχουμε δηλαδή την πολυτέλεια να σταματάμε την εκτέλεση του προγράμματος που είναι φορτωμένο στο arduino.

Στη συνέχεια πρέπει να αποσυμπίεσουμε τα αρχεία που κατεβάσαμε. Αν δεν έχετε το κατάλληλο πρόγραμμα για να το κάνετε, μπορείτε να εγκαταστήσετε στον υπολογιστή σας το 7.zip, το οποίο είναι δωρεάν (γκουγκλάρετέ το, θα το βρείτε αμέσως). Αποσυμπίεστε το αρχείο που κατεβάσατε (θα δημιουργήσετε ένα νέο φάκελο, στο σημείο του δίσκου που εσείς θα επιλέξετε). Μπείτε στο φάκελο αυτό και στη συνέχεια μπείτε στο φάκελο “Arduino” που περιέχεται εκεί. Στη συνέχεια μπείτε στο φάκελο Libraries και αντιγράψτε το φακέλο “SX1272”. Στη συνέχεια πηγαίνετε στο φάκελο “documents” του δικού σας υπολογιστή, και στη συνέχεια στο φάκελο “Arduino” (Ο φάκελος αυτός υπάρχει αν έχετε εγκαταστήσει το Arduino IDE στον υπολογιστή σας). Μέσα στο φάκελο “Arduino”, υπάρχει ο φάκελος “Libraries”. Επικολλήστε μέσα στο φάκελο αυτό το φάκελο “SX1272” που είχατε αντιγράψει.

Ο αποσυμπίεσμένος φάκελος περιέχει πολλά παραδείγματα για διαφορετικές περιπτώσεις αισθητήρων. Ο πιο απλός τρόπος για να συνεχίσετε είναι να δημιουργήσετε ένα νέο φάκελο και μέσα σε αυτόν να επικολλήσετε το αρχείο .ino που βλέπετε εδώ. Στο αρχείο αυτό έχουμε κρατήσει μόνο τον κώδικα που είναι απαραίτητος για τη δική μας περίπτωση, όπου

- ξέρουμε το board που θα χρησιμοποιήσουμε (arduino uno)
- αποφασίσαμε να μη χρησιμοποιήσουμε τις βιβλιοθήκες για την εξοικονόμηση ενέργειας

Τα βασικά κομμάτια του κώδικα και το τι αυτά κάνουν, μπορείτε να τα βρείτε και [εδώ](#) (σας συστήνουμε να το διαβάσετε για να καταλάβετε πως λειτουργεί ο κώδικας). Σε σχέση με αυτά που θα διαβάσετε, έχουμε καταργήσει τα αρχεία "my_demo_sensor_code.h" και "my_demo_sensor_code.cpp". Δε χρειαζόμαστε επίσης τα PIN_HEAD και PIN_POWER καθώς και τις `Sensor_Init()` και `sensor_getValue()`

Δώσαμε στον κόμβο (αισθητήρα) μας τον αριθμό 5, και επιλέξαμε να δουλέψουμε με `MODE = 1`. Αν θέλετε να ορίσετε `MODE = 11` θα πρέπει να βάλετε ακόμη κάποιες γραμμές κώδικα που εμείς έχουμε σβήσει.

Επιλέξαμε να μηδενίζουμε την τιμή του μετρητή μετά από κάθε ασύρματη εκπομπή. Έτσι, κάθε φορά στέλνεται ο αριθμός των μεταβάσεων που έγιναν μέσα στο τελευταίο λεπτό. Ο κώδικας στη συνάρτηση `loop()` είναι στην ουσία ο κώδικας που βρίσκεται στο αρχείο [rain_gauge_0](#), με την προσθήκη μίας ασύρματης εκπομπής της τιμής του μετρητή, κάθε 60 δευτερόλεπτα.

Πηγαίνοντας στο serial monitor, βλέπουμε τα ακόλουθα

```
SX1276 detected, starting
SX1276 LF/HF calibration
...
Setting Mode: state 0
Setting Channel: state 0
Setting Power: state 0
Setting node addr: state 0
SX1272 successfully configured
```

που σημαίνει ότι το LoRa module λειτουργεί σωστά (σε άλλη περίπτωση θα πρέπει να κοιτάξετε τη συνδεσμολογία για κάποιο λάθος). Μετά από 1 λεπτό, στέλνεται το πρώτο μήνυμα και στο serial monitor βλέπουμε τα ακόλουθα

```
Sending \!TC/0.00
Real payload size is 9
--> CS1
--> CAD 549
OK1
--> RSSI -123
Packet sent, state 0
```

που σημαίνει ότι στάλθηκε η τιμή 0, με την ετικέτα "TC". Αντί για "TC" θα μπορούσαμε να έχουμε οποιοδήποτε όνομα σας αρέσει αλλάζοντας την τιμή του `nomenclature_str` (στην εντολή `char nomenclature_str[4]="TC";`). Μπορείτε να χρησιμοποιήσετε το μαγνήτη σας για να αυξήσετε την τιμή του μετρητή. Κάθε φορά που η τιμή αυξάνεται, η νέα τιμή της εμφανίζεται στο serial monitor και αυτό το κάναμε για να μπορούμε να βλέπουμε αν λειτουργεί καλά ο μαγνητικός διακόπτης. Στην τελική έκδοση του προγράμματος που θα φορτώσετε στο arduino, Μπορείτε να σβήσετε την εντολή που εμφανίζει τη νέα τιμή του μετρητή.

Στο σημείο αυτό τελειώνουν οι ενέργειές μας στην πλευρά του αισθητήρα, αφού ολοκληρώσαμε την περιγραφή του υλικού και του λογισμικού. Το arduino μας στέλνει επιτυχώς τα μηνύματά του, αλλά δεν υπάρχει ο κεντρικός σταθμός για να τα διαβάσει. Είναι ώρα λοιπόν να ασχοληθούμε με τον κεντρικό σταθμό.

Κεντρικός σταθμός

Εγκατάσταση και βασικές ρυθμίσεις

Μπείτε στη σελίδα του project LowCostLoRaGw (Low Cost LoRa Gateway) κάνοντας κλικ [εδώ](#). Θα χρειαστείτε ένα micro SD card (8 GB το λιγότερο) και ένα micro SD USB card reader. Στη συνέχεια πρέπει να κατεβάσετε το συμπιεσμένο κώδικα για τον κεντρικό σταθμό. Στην ιστοσελίδα του LowCostLoRaGw project δίνεται link για να κατεβάσετε το συμπιεσμένο κώδικα στον υπολογιστή σας. Όταν κατεβάσετε το συμπιεσμένο αρχείο, θα πρέπει να “φλασάρετε” τον κώδικα στο raspberry. Για το σκοπό αυτό εμείς χρησιμοποιήσαμε το πρόγραμμα Balena Etcher (είναι δωρεάν). Όταν το “φλασάρισμα” τελειώσει βγάξετε το micro SD card από το card reader και το τοποθετείτε στο raspberry. Τροφοδοτείτε με ρεύμα το raspberry και και το συνδέετε στο Διαδίκτυο με ένα καλώδιο δικτύου. Όταν εκκινήσει το λογισμικό που “φορτώσατε” στο micro SD card, δημιουργείται ένα νέο ασύρματο δίκτυο με όνομα WAZIUP_PI_GW_XXXXXXXXXX (τα “X” θα είναι αριθμοί). Το password για αυτό το WiFi είναι “loragateway”. Αφού μπείτε σε αυτό το WiFi, ανοίξετε το φυλλομετρητή του Ιστού που έχετε στον υπολογιστή σας (firefox, chrome, κλπ) και πληκτρολογήστε τη διεύθυνση 192.168.200.1

Βλέπετε μία σελίδα που έχει τίτλο “access to the data from MongoDB αλλά δε βλέπετε τίποτα χρήσιμο στη σελίδα αυτή. Η λέξη MongoDB ίσως σας είναι άγνωστη. Η MongoDB είναι μία βάση δεδομένων

Ρωτήστε τον καθηγητή πληροφορικής του σχολείου σας να σας εξηγήσει τι είναι μία Βάση Δεδομένων

Στο κάτω μέρος της σελίδας υπάρχουν επιλογές που καθορίζουν το είδος των δεδομένων που θα εμφανιστούν καθώς και την πηγή (αισθητήρα) από την οποία έρχονται τα δεδομένα. Θα δείτε να αναφέρεται μόνο η πηγή 36, ενώ ο δικός μας κόμβος έχει την τιμή 5. Στα δεδομένα θα δείτε την ετικέτα “TC” που αντιστοιχεί στην τιμή του μετρητή καθώς και τους όρους RSSI και SNR (γκουγκλάρετε: LoRa RSSI SNR για να δείτε τι αντιπροσωπεύουν.

Ζητήστε από τον καθηγητή φυσικής του σχολείου σας να σας εξηγήσει όσα αναφέρονται σε [αυτό το βίντεο](#) για τους όρους RSSI και SNR

Δε βλέπουμε κάτι άλλο (προς το παρόν) στην ιστοσελίδα που βρισκόμαστε. Επιλέγουμε “Web admin interface”. Βάζουμε για username, “admin” και για password, “loragateway” (χωρίς τα “αυτάκια”) και εμφανίζεται η σελίδα με τίτλο “gateway configuration”. Το πρώτο πράγμα που έχουμε να κάνουμε να κάνουμε είναι να επιλέξουμε “Gateway Update” στα αριστερά της οθόνης μας και στη συνέχεια να επιλέξουμε “full update”. Θα περιμένουμε μέχρι να εμφανιστεί το μήνυμα ότι η ενημέρωση έχει ολοκληρωθεί. Στη συνέχεια, θα επιλέξουμε “Gateway configuration” στην αριστερή πλευρά της οθόνης και θα ασχοληθούμε με τη ρύθμιση του gateway.

Η ρύθμιση είχε κάποια προβλήματα σε εμάς. Στην καρτέλα Radio επιλέξαμε Mode = 1, Frequence = 433.3 και PA_BOOST = true. Έβγαине όμως ένα κόκκινο μήνυμα rx not found μέχρι που βάλαμε Spreading Factor = 9. Για να λειτουργήσουν οι ρυθμίσεις που κάνατε πρέπει να επανεκκινήσετε το gateway επιλέγοντας reboot στην πάνω δεξιά δεξιά πλευρά της οθόνης σας. Όταν γίνεται reboot το WiFi εξαφανίζεται και πρέπει να περιμένετε να εμφανιστεί ξανά για να πληκτρολογήσετε 192.168.200.1 και στη συνέχεια να πάτε στο “Web admin interface” για να δείτε αν όλα λειτουργούν σωστά, να δείτε δηλαδή αν βγάζει κάποιο μήνυμα λάθους (αυτό θα είναι με κόκκινα γράμματα) και να ο κεντρικός σταθμός λαμβάνει τα “πακέτα” (έτσι λέγονται τα μηνύματα) που στέλνει ο αισθητήρας. Για να δείτε αν λαμβάνει πακέτα, ο ευκολότερος τρόπος είναι να πατηήσετε “pkt logger” πάνω δεξιά στην οθόνη σας. Εκεί εμφανίζονται όλα τα πακέτα που έχουν ληφθεί. Βέβαια για να λάβει πακέτα, πρέπει να τροφοδοτήσετε με ρεύμα τον αισθητήρα σας (απλά συνδέστε το arduino στη θύρα USB του υπολογιστή σας). Μη βιαστείτε να απογοητευτείτε, αφού πρέπει να περιμένετε 1 λεπτό για να σταλεί το πρώτο πακέτο. Αν δείτε να καταγράφονται τα πακέτα στον pkt logger, το μεγάλο βήμα για την επίτευξη του τελικού στόχου, έχει γίνει.

Θα πρέπει να πούμε ότι στα πακέτα που εμείς λαμβάνουμε, ο spreading factor (SF) είναι 12. Αυτό ταιριάζει με ότι έχουμε κάνει στον αισθητήρα, αφού εκεί έχουμε όντως θέσει SF = 12. Όμως,

στον κεντρικό σταθμό έχουμε ορίσει $SF = 9$. Μας παραξένεψε, αλλά δεν επιμείναμε. Καλό θα ήταν, αν αποφασίσετε να κάνετε κι εσείς το project, να δοκιμάσετε με διαφορετικά spreading factors για να δείτε τι γίνεται.

MongoDB

Από τη στιγμή που δείτε τα πακέτα στον pkt logger, ήρθε η ώρα να κάνουμε μερικές επιπλέον ρυθμίσεις, που όμως απαιτούν πρόσβαση στο raspberry. Ένας τρόπος για να πετύχουμε την πρόσβαση αυτή είναι συνδέσουμε στο raspberry ένα πληκτρολόγιο και μία οθόνη. Ο άλλος τρόπος είναι να χρησιμοποιήσουμε την οθόνη και το πληκτρολόγιο του φορητού υπολογιστή μας μέσω του λογισμικού putty (δείτε [εδώ](#) πως, ή ψάξτε άλλες πηγές στον Παγκόσμιο Ιστό). Όπως και να το κάνετε, θα δώσετε login: pi και password: l0ragat0way

Η πρώτη ρύθμιση αφορά την ενεργοποίηση της MongoDB. Τα πακέτα που έρχονται αποθηκεύονται σε μία βάση δεδομένων που δημιουργείται με τη βοήθεια της mongodb

Δίνοντας login και password εκτελείται ένα πρόγραμμα το οποίο σας δίνει ένα μενού εντολών. Πληκτρολογήστε Q και πατήστε enter για να τερματίσετε το πρόγραμμα αυτό. Πληκτρολογήστε mongo και πατήστε enter. Στη συνέχεια δείτε [αυτή τη σελίδα](#) και πληκτρολογήστε κατά σειρά τις εντολές:

```
> show dbs
> use messages
> show collections
> db.ReceivedData.remove({})
> exit
```

Τώρα το αρχείο που περιέχει τα πακέτα είναι άδειο, ενώ πριν περιείχε κάποια δεδομένα από ένα κόμβο με αριθμό 36 (γι, αυτό βλέπατε και τον αριθμό 36 στο κάτω μέρος της σελίδας με διεύθυνση 196.168.200.1). επόμενο βήμα είναι να ενεργοποιήσουμε την αποθήκευση των εισερχόμενων πακέτων στο αρχείο αυτό, από εδώ και πέρα. Για να το κάνουμε αυτό πληκτρολογούμε nano clouds.json (εκτελούμε τον επεξεργαστή κειμένου nano και ανοίγουμε για επεξεργασία το αρχείο clouds.json). Το αρχείο αποτελείται από κομμάτια που ανοίγουν με “{” και κλείνουν με “}”. το πρώτο κομμάτι σχετίζεται με την αποθήκευση των εισερχόμενων πακέτων μέσω της τοπικής βάσης δεδομένων. Στη γραμμή “enabled” αλλάζουμε το false σε true, πατάμε ctrl+x και μετά y. Το αρχείο clouds.json έχει τώρα αλλάξει και από εδώ και πέρα τα εισερχόμενα πακέτα θα αποθηκεύονται και θα εμφανίζονται στη σελίδα 192.168.299.1

Επανεκκινήστε το λογισμικό του raspberry και πηγαίνετε στη σελίδα 192.168.200.1. κάνετε refresh μετά από 1-2 λεπτά (περιμένετε πρώτα να φτάσει το πρώτο πακέτο από τον αισθητήρα σας) και θα δείτε στο κάτω μέρος να σας δίνει τον κόμβο 5. Επιλέξτε τον κόμβο 5 και “TC” κάτω αριστερά στην οθόνη σας. Μπορείτε τώρα να δείτε τη γραφική παράσταση από τις τιμές TC που έστειλε ο κόμβος 5. Επιλέξτε “current day” για να είναι πιο ευδιάκριτες οι τιμές.

Cloud πλατφόρμες

Το λογισμικό μας δίνει τη δυνατότητα χωρίς εμπλακούμε με τον κώδικα, να στέλνουν τις τιμές που λαμβάνονται σε κάποια πλατφόρμα που υπάρχει κάπου στο Διαδίκτυο (χρησιμοποιούμε γι, αυτές τις πλατφόρμες τον όρο “cloud”

Ρωτήστε τον καθηγητή πληροφορικής του σχολείου σας να σας εξηγήσει τι σημαίνει ο όρος “cloud”. Πρόκειται για όρο πολύ δημοφιλή τα τελευταία χρόνια, που πολλοί χρησιμοποιούν χωρίς όμως να καταλαβαίνουν τι σημαίνει.

Από όλες τις cloud πλατφόρμες που μπορούμε να αξιοποιήσουμε επιλέξαμε την “[thingspeak](#)”. Αφού ανοίξετε ένα λογαριασμό στην πλατφόρμα (χρειάζεστε e-mail γι, αυτό), μπορείτε να δημιουργήσετε ένα ή περισσότερα κανάλια. Αρκεί να δώσουμε στο κανάλι μας ένα όνομα. Στην

περίπτωσή μας χρησιμοποιούμε μόνο ένα πεδίο (μπορούμε μέχρι και 8). Ονομάζουμε το πρώτο πεδίο “TC”. Αφού αποθηκεύσουμε όσα κάναμε, πηγαίνουμε στην καρτέλα API keys και αντιγράφουμε κάπου το “write API key”. Στη συνέχεια κάνουμε login στο raspberry (είτε συνδέοντας σε αυτό οθόνη και πληκτρολόγιο είτε μέσω του λογισμικού Putty). Όπως είπαμε είναι login: pi και password: loragateway. Πληκτρολογούμε Q και enter για να τερματίσουμε το εκτελούμενο πρόγραμμα (προσοχή, δεν τερματίζεται έτσι και το λογισμικό gateway). Μετά πληκτρολογούμε

```
nano key_ThingSpeak.py
```

Στη δεύτερη γραμμή του αρχείου, defthingspeak_channel_key='XXXXXXXXXXXXXXXXXX' αντικαθιστούμε τους 16 χαρακτήρες (που αναπαριστούμε εδώ με τα “X”) με το write API key του καναλιού που δημιουργήσαμε. Αποθηκεύουμε την αλλαγή που κάναμε (CTRL+X και μετά y) και επανεκκινούμε το gateway. Μπορούμε τώρα να δούμε το πεδίο 1 του καναλιού μας να εμφανίζει τις τιμές που ο αισθητήρας μας έστειλε στο gateway.

[1]. <https://www.youtube.com/watch?v=RruQhoXrVoE>

[2]. <https://en.wikipedia.org/wiki/LoRa>