



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μέτρηση αιωρούμενων μικρο-σωματιδίων με Arduino.

Νικόλαος Δημόπουλος

A.M. : HN08256

Επιβλέπων Καθηγητής : Ιωάννης Βανδίκας

Κοζάνη . Οκτώβριος 2022

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και ρητά, ότι η παρούσα Διπλωματική Εργασία με τίτλο «Μέτρηση αιωρούμενων μικροσωματιδίων με Arduino» γνωρίζοντας τις κυρώσεις το Ν.2121/1993 περί Πνευματικής Ιδιοκτησίας, τα αρχεία που έχουν αναπτυχθεί ή τροποποιηθεί στα πλαίσια αυτής της εργασίας και δεν είναι προϊόν ολικής ή μερικής αντιγραφής. Οτιδήποτε έχει χρησιμοποιηθεί για την ολοκλήρωση της συγγραφής της εργασίας περιλαμβάνονται στη βιβλιογραφία.

Δημόπουλος Νικόλαος

Υπογραφή

Ευχαριστίες

Θα ήθελα σε αυτό το σημείο να ευχαριστήσω τον καθηγητή μου κ. Ιωάννη Βανδίκη για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα, όσο και για την αμέριστη βοήθεια του καθόλη τη διάρκεια συγγραφής της παρούσας πτυχιακή εργασίας, βοήθεια χωρίς την οποία δεν θα ήταν δυνατή η ολοκλήρωση της.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για την απολύτιμη στήριξη τους, τους γονείς μου, συμφοιτητές και τους φίλους μου για την υποστήριξη τους όλα αυτά τα χρόνια.

Δημόπουλος Νικόλαος

Περίληψη

Σκοπός της παρούσας εργασίας είναι η μελέτη , σχεδίαση και υλοποίηση ενός συστήματος επιτήρησης των μικροσωματιδίων του χώρου ,με ενεργοποίηση ενός ανεμιστήρα σε περίπτωση ανίχνευσης υψηλού αριθμού σωματιδίων στο χώρο. Το συγκεκριμένο σύστημα που προαναφέραμε υλοποιείται με την χρησιμοποίηση μιας πλακέτας Arduino, ενός αισθητήρα μέτρησης μικρο-σωματιδίων PMS7003, ενός ηλεκτρονόμου (Relay), μιας οθόνης TFT 1.8 που μας δείχνει σε πραγματικό χρόνο τις μετρήσεις, NeoPixel Ring RGB, και ενός ανεμιστήρα.

Η πτυχιακή εργασία αποτελείται από τέσσερα κεφάλαια : στο πρώτο κεφάλαιο γίνεται μια εισαγωγή και ο λόγος ώθησης για τη δημιουργία αυτής της εργασίας, στο δεύτερο αναφερόμαστε στην πλακέτα του ARDUINO και στα ηλεκτρονικά εξαρτήματα και λογισμικά που χρησιμοποιήθηκαν για την υλοποίηση της, στο τρίτο αναφερόμαστε στη λειτουργία και σχεδίαση του προτεινόμενου συστήματος με την ανάλυση των επιμέρους εξαρτημάτων που χρησιμοποιήθηκαν και στο τέταρτο κεφάλαιο παρατίθεται ο κώδικας που γράφτηκε για την ομαλή λειτουργία του. Ακολουθούν τα συμπεράσματα , βελτιώσεις και οι βιβλιογραφικές πηγές.

Λέξεις Κλειδιά: Arduino, ηλεκτρονόμος (relay), μικροελεγκτής , PMS7003, TFT 1.8, Neopixel Ring,

Abstract

The purpose of this work is the study, design and implementation of a system for monitoring microparticles in the air by activating a fan in case of detecting a high number of particles in the air. The specific system we mentioned above is implemented using an Arduino board, PMS7003 micro-particle measurement sensors, relay, a 1.8 TFT screen which shows us the measurements in real time, NeoPixel Ring RGB, and a fan.

The thesis consists of four chapters: in the first chapter there is an introduction and the motivation for creating this work, in the second we refer to the ARDUINO board and the electronic components and software used for its implementation, in the third part we refer to the function ,design of the proposed system with the analysis of the individual components used, the fourth chapter lists the code written for its smooth operation. The conclusions, improvements and bibliographic sources follow

Keywords: Arduino, relay, microcontroller, PMS7003, TFT 1.8, Neopixel Ring RGB

Πίνακας Περιεχομένων

<u>Ευχαριστίες.....</u>	<u>I</u>
<u>Περίληψη</u>	<u>II</u>
<u>Abstract</u>	<u>III</u>
<u>ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ.....</u>	<u>VI</u>
<u>Κεφάλαιο 1 – Εισαγωγή.....</u>	<u>1</u>
<u>1.1 Αντικείμενο Διπλωματικής Εργασίας.....</u>	<u>1</u>
<u>1.2 Κίνητρα και στόχοι υλοποίησης.....</u>	<u>2</u>
<u>Κεφάλαιο 2 – Θεωρητικό υπόβαθρο.....</u>	<u>3</u>
<u>2.1 PMS 7003 Air Quality Sensor.....</u>	<u>3</u>
<u>2.2 Arduino Uno</u>	<u>10</u>
<u>2.3 DS1307.....</u>	<u>15</u>
<u>2.4 LED NeoPixel 24.....</u>	<u>18</u>
<u>2.4.1 HW-482 5v Relay.....</u>	<u>21</u>
<u>2.4.1 BC547</u>	<u>23</u>
<u>2.4.1 Shields</u>	<u>25</u>
<u>2.5 Πλατφόρμες Ανάπτυξης Λογισμικού</u>	<u>26</u>
<u>2.5.1 Miniblog :.....</u>	<u>26</u>
<u>2.5.2 ModKit :</u>	<u>27</u>
<u>2.5.3 Code::Blocks :</u>	<u>28</u>
<u>2.5.4 Arduino IDE (Integrated Development Environment)</u>	<u>29</u>

<u>2.6 Σύνοψη Κεφαλαίου</u>	<u>33</u>
<u>Κεφάλαιο 3 – Συναρμολόγηση του συστήματος μέτρησης</u>	<u>33</u>
<u>3.1 Το ηλεκτρικό κύκλωμα του συστήματος</u>	<u>33</u>
<u>Κεφάλαιο 4 - Κώδικας Λειτουργίας</u>	<u>39</u>
<u>Συμπεράσματα</u>	<u>50</u>
<u>Κόστος Εργασίας</u>	<u>50</u>
<u>Κεφάλαιο Βελτιώσεων</u>	<u>51</u>
<u>Βιβλιογραφία</u>	<u>52</u>

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1 « Αισθητήρας PMS7003»	4
Εικόνα 2 « Αρχή λειτουργίας PMS7003»	5
Εικόνα 3 « Ακίδες PMS7003»	7
Εικόνα 4 « Μπλοκ διάγραμμα σύνδεσης PM2.5»	8
Εικόνα 5 « Πλακέτα Arduino Uno»	13
Εικόνα 6 « Arduino UnoPin Connection»	14
Εικόνα 7 « Πλακέτα DS1307»	15
Εικόνα 8 « Pin Connection DS1307»	17
Εικόνα 9 « I2C Protocol»	18
Εικόνα 10 « RGB Led Connection»	19
Εικόνα 11 « RGB Led Pin»	19
Εικόνα 12 « RGB Led Τιμές φωτεινότητας και έντασης»	20
Εικόνα 13 « HW- 482 5v Relay»	21
Εικόνα 14 « HW- 482 5v Relay Normally Open»	22
Εικόνα 15 « HW- 482 5v Relay Normally Closed »	23
Εικόνα 16 « BC547 NPN Transistor»	23
Εικόνα 17 « TFT Shield »	25
Εικόνα 18 « Minibloq »	27
Εικόνα 19 ModKit »	28
Εικόνα 20 « Code Blocks »	29
Εικόνα 21 « Arduino IDE »	30
Εικόνα 22 « Σχεδιάγραμμα εργασίας»	34
Εικόνα 23 « Απεικόνιση των μετρήσεων»	35
Εικόνα 24 « Ένδειξη Ημερομηνίας και ώρας σε πραγματικό χρόνο »	36
Εικόνα 25 « Συνδεσμολογία ρολόι DS1307»	37
Εικόνα 26 « NeoPixel RGB Ring φωτεινές ενδείξεις»	37
Εικόνα 27 « Συνδεσμολογία NeoPixel RGB Ring»	38

Κεφάλαιο 1 – Εισαγωγή

Στις μέρες μας ο αυτοματισμός παίζει σημαντικό ρόλο στην καθημερινότητά μας, για αυτό το λόγο οι επιστήμονες, προσπαθούν να αυτοματοποιήσουν όσο το δυνατόν περισσότερες συσκευές για την ολοκλήρωση μιας εργασίας, όσο δύσκολη και αν είναι. Έτσι, στις συσκευές και στα συστήματα καθημερινής χρήσης ενσωματώνονται μικροελεγκτές, αισθητήρες για να επιτευχθεί αυτονομία. Για πολλά χρόνια απασχολεί την ανθρωπότητα η μόλυνση της ατμόσφαιρας. Ακόμη, περισσότερο η ποιότητα του αέρα έχει αλλάξει λόγω της ρύπανσης από επιβλαβή αέρια της ατμόσφαιρας, για αυτό και στα σπίτια που υπάρχουν παιδιά και ηλικιωμένοι με αναπνευστικά προβλήματα, επιλέγουν καθαριστές αέρα που υπάρχουν στην αγορά. Λαμβάνοντας αυτό υπόψη καταλήγουμε ότι θα ήταν ιδιαίτερα χρήσιμο, η κατασκευή ενός συστήματος που θα παρακολουθεί την ποιότητα του αέρα, τόσο σε εσωτερικούς όσο και σε εξωτερικούς χώρους. Αυτό το σύστημα θα αποτελείται από το arduino και λίγους αισθητήρες για τη λήψη των δεδομένων.

1.1 Αντικείμενο Διπλωματικής Εργασίας

Η παρούσα διπλωματική εργασία αφορά την ανάπτυξη ενός μετρητή μικροσωματιδίων μικρού μεγέθους, ώστε να μπορεί ο χρήστης να παρακολουθεί την ποιότητα του αέρα σε περιοχές με συνωστισμό ή βιομηχανικές περιοχές. Εφόσον ανεβεί ο αριθμός των μικροσωματιδίων θα δίνεται εντολή σε ένα ανεμιστήρα για την ανανέωση του καθαρού αέρα.

Το σύστημα αποτελείται από ένα μετρητή μικροσωματιδίων PMS7003, μέσα από τον οποίο περνάει ο αέρας, αναλύεται και επιστρέφει τα αποτελέσματα σε μορφή δεδομένων σε ένα μικροελεγκτή Arduino Uno R3. Το Arduino Uno αποτελεί τον εγκέφαλο του όλου συστήματος, αφού όλες οι υπόλοιπες συσκευές συνδέονται πάνω του.

Ο κώδικας, ο οποίος είναι υπεύθυνος για την ορθή βαθμονόμηση των μετρήσεων χρησιμοποιεί τα δεδομένα που λαμβάνει από τον PMS και έπειτα από την ανάλυση τα αντικατοπτρίζει σε μια οθόνη, όχι μόνο σαν αριθμούς που μεταβάλλονται αλλά και ως γράφημα.

1.2 Κίνητρα και στόχοι υλοποίησης

Ο Παγκόσμιος Οργανισμός Υγείας (Π.Ο.Υ) το 2014 ανακοίνωσε περίπου 7 εκατομμύρια θανάτους ανθρώπων παγκοσμίως λόγω της ατμοσφαιρικής ρύπανσης. Μια παρόμοια προσέγγιση ισοδυναμεί χονδρικά από τον Διεθνή Οργανισμό Ενέργειας (IEA) επίσης. Αυτοί οι ρύποι ή χημικές ουσίες είναι επίσης υπεύθυνοι σε μεγάλο βαθμό για διάφορες περιβαλλοντικές καταστροφές, όπως η όξινη βροχή αλλά και η καταστροφή του στρώματος του όζοντος. Με αφορμή αρκετών ανθρωπογενών δράσεων, η ρύπανση του αέρα αναπτύσσεται συνεχώς και ο έλεγχός - περιορισμός αυτής της κλιμάκωσης είναι σημαντικής σημασίας για να την ανακούφιση συγκεκριμένων ενεργειών . Στο παρελθόν, οι αισθητήρες μέτρησης της ποιότητας του αέρα ήταν πολύ μεγάλοι, μη φορητοί και ακριβοί. Στην εποχή μας η ανάπτυξη των περισσότερων αισθητήρων ρύπανσης έχουν επικεντρωθεί στους πέντε πιο συνηθισμένους ρύπους δηλαδή. οξείδιο του αζώτου, μονοξείδιο του άνθρακα, όζον, διοξείδιο του θείου και σωματίδια. Στον σημερινό κόσμο, η ρύπανση του αέρα και η παρακολούθηση της ποιότητας είναι πραγματικά ζωτικής σημασίας ,καθώς έχει μεγάλη επίδραση στην ανθρώπινη υγεία. Ο αισθητήρας μέτρησης της ποιότητας του αέρα μπορεί να αναγνωρίσει και να παρατηρήσει την επίπτωση της ατμοσφαιρικής ρύπανσης στις παρακείμενες περιοχές. Μπορεί να χρησιμοποιηθεί τόσο για εσωτερικούς όσο και για εξωτερικούς χώρους. Στο μέλλον και με τις αλματώδεις τεχνολογικές βελτιώσεις που σημειώνονται, αυτοί οι αισθητήρες θα γίνουν φθηνότεροι και φορητοί , οι αισθητήρες μετρήσεων πιθανόν να μπορούν να φορεθούν από τους ανθρώπους ανάλογα με την τοπική ποιότητα αέρα.

Κεφάλαιο 2 – Θεωρητικό υπόβαθρο

Σε αυτήν την ενότητα θα παρουσιάσουμε αναλυτικά τα υλικά τα οποία χρησιμοποιήθηκαν για υλοποιηθεί το σύστημα , όπως και το λογισμικό που χρησιμοποιήθηκε για τη δημιουργία του προγράμματος, ώστε να έχει ομαλή λειτουργία. Συγκεκριμένα στο παρόν κεφάλαιο θα δούμε τις αλληλεπιδράσεις των ηλεκτρονικών και μηχανικών εξαρτημάτων,τη χρήση τους,τις βασικές λειτουργίες και τις προδιαγραφές τους. Σαν μηχανικό ενσωματώνονται ο ανεμιστήρας , ενώ το ηλεκτρονικό τμήμα χωρίζεται σε δυο κύριες μονάδες σε αυτές που επεξεργάζονται τα δεδομένα και εκτελούν το πρόγραμμα, και σε αυτές που συλλέγουν τα δεδομένα.

2.1 PMS 7003 Air Quality Sensor

Τα μικροσωματίδια είναι ένας ατμοσφαιρικός ρύπος που ανησυχεί για την υγεία των ανθρώπων και η μεγάλη συγκέντρωσή τους κάνει τον αέρα να φαίνεται μουντός. Επομένως, ο αριθμός των αιωρούμενων σωματιδίων στον αέρα είναι μια πολύ σημαντική περιβαλλοντική τιμή. Αυτό το προτεινόμενο σύστημα χρησιμοποιεί έναν αισθητήρα PMS7003 για να ανιχνεύσει τη συγκέντρωση μικροσωματιδίων στον αέρα. Το PMS7003 είναι ένα είδος ψηφιακού αισθητήρα μέτρησης συγκέντρωσης σωματιδίων, που μπορεί να χρησιμοποιηθεί για να ληφθεί ο αριθμός των αιωρούμενων σωματιδίων στον αέρα, δηλαδή τη συγκέντρωση των σωματιδίων και την έξοδο τους με τη μορφή ψηφιακής διεπαφής. Αυτός ο αισθητήρας μπορεί να εισαχθεί σε μεταβλητά όργανα που σχετίζονται με η συγκέντρωση αιωρούμενων σωματιδίων στον αέρα ή σε άλλο περιβάλλον εξοπλισμού βελτίωσης για την έγκαιρη παροχή σωστών δεδομένων συγκέντρωσης.

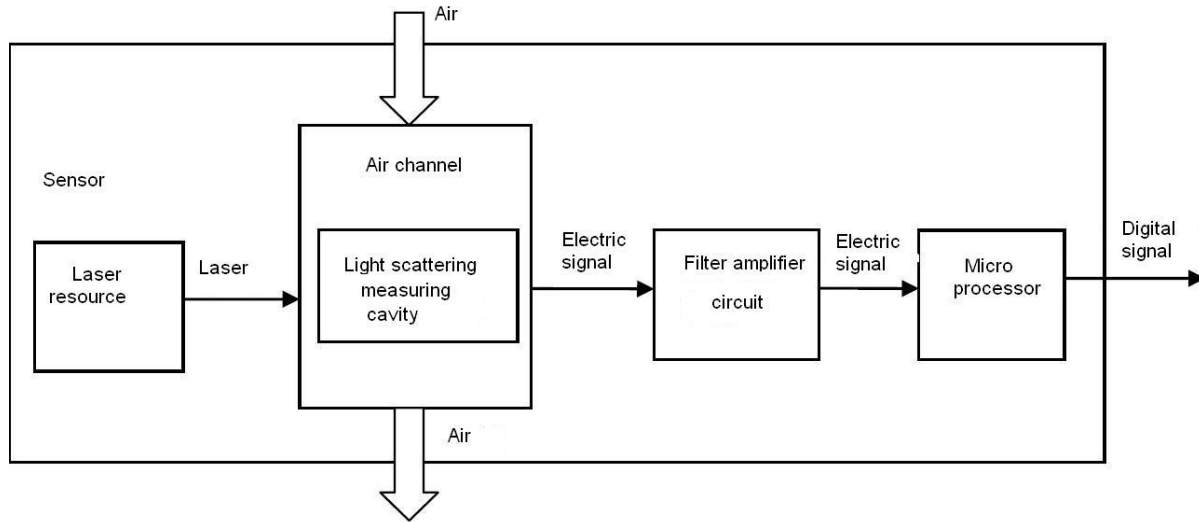


Εικόνα 1 « Αισθητήρας PMS7003» .

Όταν στον όρο σκέδαση ή σκέδαση του φωτός ονομάζουμε τον διασκορπισμό των φωτεινών ακτίνων ,όταν προσπέσουν σε μικροσκοπικά σωματίδια, έτσι ώστε να διαχέονται στο χώρο χωρίς να φαίνονται. Φαινόμενο σκέδασης ονομάζουμε την οπτική αντίληψη του γεγονότος αυτού. Η σκέδαση είναι κάτι που δημιουργείται απο πολλές ανακλασης και όχι απο μια μόνο. Ουσιαστικά, αναφερόμαστε στην ανάκλαση και απορρόφηση του φωτός από ένα σωματίδιο και την άμεση επανεκπομπή του από τα μικροσκοπικά σωματίδια πάνω στα οποία προσπίπτει, ενώ οι συχνότητες από τις οποίες αποτελείται το φως, παραμένουν αμετάβλητες.^[10]

Αρχή λειτουργίας:

Η αρχή σκέδασης με λέιζερ χρησιμοποιείται για έναν τέτοιο αισθητήρα, δηλαδή παράγει σκέδαση χρησιμοποιώντας λέιζερ για την ακτινοβολία των αιωρούμενων σωματιδίων στον αέρα και στη συνέχεια, συλλέγει την σκεδασμένη δέσμη φωτός σε συνάρτηση με το χρόνο. Στο τέλος, ισοδύναμη διάμετρος σωματιδίων και ο αριθμός των σωματιδίων με διαφορετική διάμετρο ανά μονάδα όγκου μπορεί να υπολογιστεί με μικροεπεξεργαστή με βάση τη θεωρία ΜΙΕ. Βρείτε το λειτουργικό διάγραμμα κάθε τμήματος του αισθητήρα από το σχήμα 1 ως εξής.



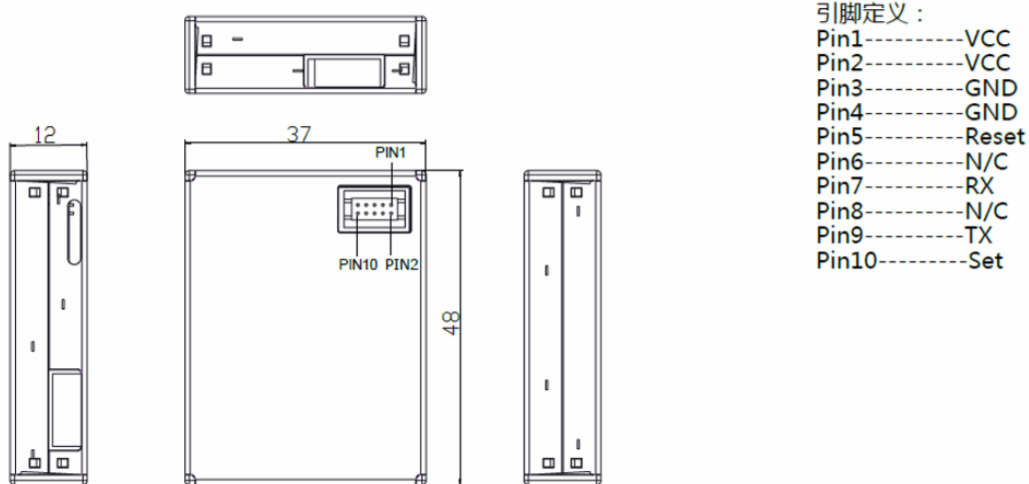
Εικόνα 2 « Αρχή λειτουργίας PMS7003»

Όπως φαίνεται στο σχήμα 3.5, η σκέδαση φωτός μπορεί να προκληθεί, όταν τα σωματίδια περνούν από την περιοχή ανίχνευσης. Το σκεδαζόμενο φως μετατρέπεται σε ηλεκτρικά σήματα και αυτά τα σήματα θα ενισχυθούν και θα υποβληθούν σε επεξεργασία. Ο αριθμός και η διάμετρος των σωματιδίων μπορούν να ληφθούν με ανάλυση, επειδή η κυματομορφή του σήματος έχει κάποια σχέση με τη διάμετρο των σωματιδίων. Για να λειτουργεί το σύστημα με ακρίβεια, ο ανιχνευόμενος αέρας που διατρέχει τον αισθητήρα πρέπει να είναι αεριζόμενος, έτσι ένας ανεμιστήρας είναι ενσωματωμένος στον αισθητήρα.

Parameter	index	unit
Range of measurement	0.3~1.0 , 1.0~2.5 , 2.5~10	Micrometer (μ m)
Minimum distinguishable particle diameter	0.3μm	
Counting Efficiency	50% @ 0.3 - 98% @>= 0.5	Micrometer (μ m)
Operational voltage:	DC 5	Volt (V)
Effective Range (PM2.5 standard)	0 ~ 500	μ g/m ³
Maximum Range (PM2.5 standard)	≥ 1000	μ g/m ³
Resolution	1	μ g/m ³
Standard Volume	0.1	Litre (L)
Total Response Time:	≤10	Second (s)
Single Response Time	≤1	Second (s)
DC Power Supply	Typ:5.0 Min:4.5 Max: 5.5	Volt (V)
Activev Current	≤100	Microampere (μ A)
Standby Current	≤200	Microampere (μ A)
Interface Level	L <0.8 @3.3 > 2.7 @3.3	Volt (V)
Working Temperature Range	-10 ~ +60	°C
Working Humidity Range	0 ~ 99%	
Storage Temperature Range	-40 ~ +80	°C
MTTF	≥3	Year (Y)
Dimension	48×37×12mm	Millimeter (mm)

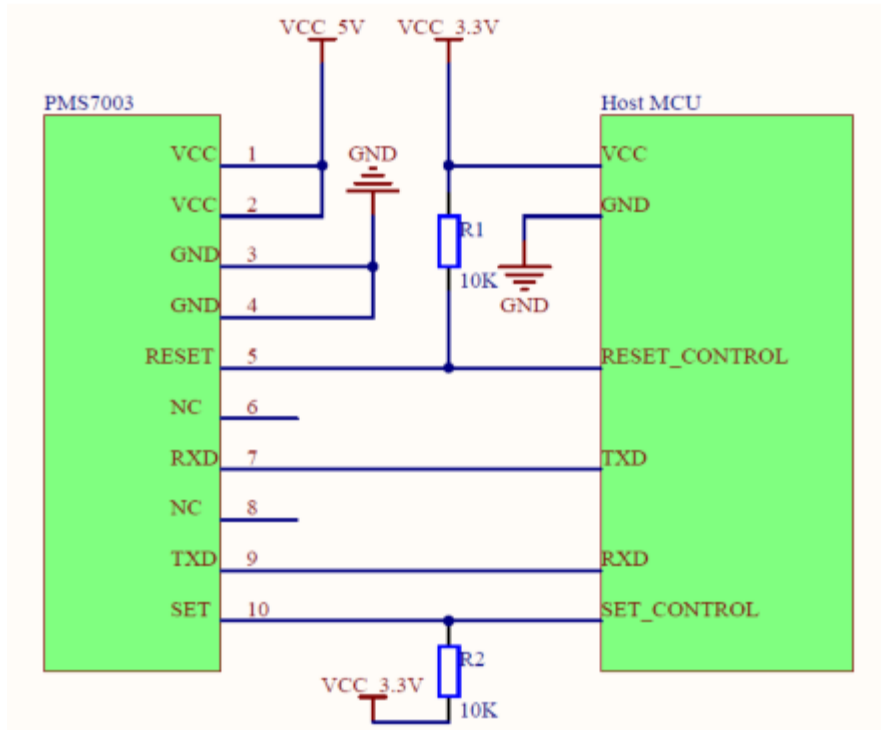
Πίνακας 1 PMS7003 Χαρακτηριστικά αισθητήρα

Σημείωση 1 : Ως μέγιστο μήκος θεωρείται ότι η έξοδος απο το PM 2.5 standard data δεν είναι λιγότερη απο 1000



Εικόνα 3 « Ακίδες PMS7003»

Αυτή η μονάδα αισθητήρα έχει δέκα ακίδες: 2xGND, 2xVCC, SET, RX, TX, RESET, PIN6 και PIN8. Σύμφωνα με το εγχειρίδιο, τα PIN6 και PIN8 δεν πρέπει ποτέ να συνδέονται. Ο ακροδέκτης VCC πρέπει να συνδεθεί στο τροφοδοτικό 5V. Η ακίδα γείωσης πρέπει να συνδέεται με το έδαφος. Ο αισθητήρας βρίσκεται σε κατάσταση λειτουργίας, όταν ο ακροδέκτης SET παίρνει τιμή 1 (HIGH). Όταν η ακίδα SET είναι σε χαμηλή στάθμη πάρει τιμή 0 (LOW), ο αισθητήρας τίθεται σε κατάσταση αδράνειας. Όταν ο ακροδέκτης RESET πέσει χαμηλά πάρει τιμή 0 (LOW), ο αισθητήρας θα επαναφέρει το σήμα, οπότε θέτουμε αυτόν τον ακροδέκτη ψηλά για να διασφαλίσουμε ότι διατηρείται η κατάσταση λειτουργίας. Οι ακίδες RX και TX συνδέονται με τη θύρα UART του Arduino για τη μετάδοση των δεδομένων.



Εικόνα 4 « Μπλοκ διάγραμμα σύνδεσης PM2.5»

Το πακέτο δεδομένων PM2.5 έχει 32 byte. τα δύο πρώτα byte δεδομένων, 0x42 και 0x4D, είναι δύο σταθερές ενδείξεις έναρξης επικοινωνίας. Το 13ο και το 14ο byte δεδομένων καθορίζουν την τιμή του PM2,5 (το 13ο είναι υψηλό 8 bit και το 14ο είναι χαμηλό 8 bit). Για να βελτιώσουμε την ακρίβεια, ρυθμίσαμε τον αισθητήρα να λαμβάνει 10 διαδοχικές μετρήσεις, εξαιρουμένων των υψηλότερων και χαμηλότερων μετρήσεων και, στη συνέχεια, παίρνουμε τον μέσο όρο των υπόλοιπων 8 μετρήσεων.

Μέτρηση αιωρούμενων μικρο-σωματιδίων με Arduino

AQI	Επίπεδα Ατμοσφαιρικής Ρύπανσης	Επιπτώσεις στην Υγεία	Δήλωση Προειδοποίησης (for PM2.5)
0 - 50	Καλό	Η ποιότητα του αέρα θεωρείται ικανοποιητική και η ατμοσφαιρική ρύπανση ενέχει ελάχιστο ή καθόλου κίνδυνο	Καθόλου
51 -100	Μέτριο	Η ποιότητα του αέρα θεωρείται αποδεκτή . Ωστόσο ορισμένοι ρύποι μπορεί να δημιουργήσουν ανησυχία για την υγεία για ένα πολυ μικρό αριθμό ατόμων , οι οποίοι είναι ασυνήθιστα ευαίσθητοι στην ατμοσφαιρική ρύπανση.	Παιδιά και ενήλικες , όπως και άτομα με αναπνευστικές παθήσεις, (π.χ.άσθμα) θα πρέπει να περιορίσουν την παρατεταμένη άσκηση σε εξωτερικούς χώρους.
101-150	Ανθυγιεινό για ευαίσθητες ομάδες	Άτομα από ευαίσθητες ομάδες μπορεί να αντιμετωπίσουν προβλήματα υγείας. Ο γενικός πληθυσμός δεν διατρέχει κάποια παρενέργεια.	Παιδιά και ενήλικες , όπως και άτομα με αναπνευστικές παθήσεις, (π.χ.άσθμα) θα πρέπει να περιορίσουν την παρατεταμένη άσκηση σε εξωτερικούς χώρους
151-200	Ανθυγιεινό	Όλος ο πληθυσμός μπορεί να διατρέξει κάποια παρενέργεια. Άτομα από ευαίσθητες ομάδες μπορεί να αντιμετωπίσουν σοβαρά προβλήματα υγείας.	Παιδιά και ενήλικες , όπως και άτομα με αναπνευστικές παθήσεις, (π.χ.άσθμα) θα πρέπει να περιορίσουν την παρατεταμένη άσκηση σε εξωτερικούς χώρους. Όλα τα άτομα και ειδικότερα τα παιδιά πρέπει να περιορίσουν την παρατεταμένη άσκηση σε εξωτερικούς χώρους.
201-300	Αρκετά Ανθυγιεινό	Προειδοποιήσεις υγείας για καταστάσεις έκτακτης ανάγκης. Ολόκληρος ο πληθυσμός είναι πιο πιθανό να επηρεαστεί.	Παιδιά και ενήλικες , όπως και άτομα με αναπνευστικές παθήσεις, (π.χ.άσθμα) θα πρέπει να περιορίσουν κάθε άσκηση σε εξωτερικούς χώρους. Όλα τα άτομα και ειδικότερα τα παιδιά πρέπει να περιορίσουν την παρατεταμένη άσκηση σε εξωτερικούς χώρους.
300+	Επικίνδυνα	Προειδοποίηση για την υγεία: ο καθένας μπορεί να αντιμετωπίσει πιο σοβαρές επιπτώσεις στην υγεία	Όλοι πρέπει να αποφεύγουν κάθε άσκηση σε εξωτερικούς χώρους

Πίνακας 2 « Πίνακας Τιμών AQI PM2.5»

2.2 Arduino Uno

Όσον αφορά την αρχιτεκτονική του το Arduino Uno είναι μια πλατφόρμα βασισμένη στον ATMEGA328. Περιέχει 14 ψηφιακούς ακροδέκτες I/O εισόδου-εξόδου, 6 αναλογικές εισόδους και 6 ψηφιακές εξόδους έναν ταλαντωτή των 16 MHz, υποδοχή σύνδεσης με USB καλώδιο, υποδοχή σύνδεσης με ρεύμα, δυνατότητα εντός του κυκλώματος σειριακού προγραμματισμού-ICSP (In-Circuit Serial Programming) και ένα κουμπί επανεκκίνησης (reset) σε περίπτωση που βραχυκυκλώσει χωρίς την θέλησή μας . Εμπεριέχει όλα όσα χρειάζονται για την υποστήριξη ενός μικροελεγκτή. Το μόνο που απαιτείται είναι η σύνδεσή του με έναν ηλεκτρονικό υπολογιστή μέσω ενός καλωδίου USB ή τροφοδοσία του μέσω ενός μετασχηματιστή τάσης για τη μετατροπή μεταξύ εναλλασσόμενου και ασυνεχούς ρεύματος (adaptor AC-to-DC-Alternating Current to Direct Current) ή μιας μπαταρίας. Για τον προγραμματισμό του Arduino χρησιμοποιείται η γλώσσα Wiring, η οποία αφομοιώνει στοιχεία από την C και την C++. Η δομή της είναι ίδια με αυτή της C και υποστηρίζει τις περισσότερες βιβλιοθήκες της. Για την δημιουργία των προγραμμάτων στη γλώσσα wiring, έχει αναπτυχθεί από την κατασκευάστρια εταιρία του Arduino, το λογισμικό Arduino IDE, το οποίο θα αναλυθεί περαιτέρω παρακάτω.

Οι πλατφόρμες Arduino διαθέτουν τρεις βασικές μνήμες:

Flash memory (32 Kbytes) στην οποία τοποθετείται κάθε φορά το πρόγραμμα που πρόκειται να εκτελεστεί καθώς και ο φορτωτής εκκίνησης που διευκολύνει την διαδικασία του προγραμματισμού της πλατφόρμας.

SRAM memory (στατική μνήμη τυχαίας προσπέλασης των 2 Kbytes) η οποία χρησιμοποιείται για την προσωρινή αποθήκευση των στατικών και των μεταβλητών δεδομένων του προγράμματος που εκτελείται.

EEPROM memory (1 Kbytes) στην οποία αποθηκεύονται οι τιμές των μεταβλητών, όταν η πλατφόρμα σβήσει(OFF). Χρησιμοποιείται για την αποθήκευση ρυθμίσεων και άλλων παραμέτρων ανάμεσα στα Reset του Arduino.

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

Επειδή δεν υπάρχει πολύ 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.

Ακροδέκτες ρεύματος

Οι Ακροδέκτες ρεύματος αποτελούνται από

3.3V: 3.3Volts έξοδο (νούμερο 12 στην παρακάτω εικόνα)

5V: 5 Volts έξοδο (νούμερο 13 στην παρακάτω εικόνα)

GND: 2 γειώσεις (νούμερα 3 και 14 στην παρακάτω εικόνα)

VIN: Η τάση εισόδου στην πλακέτα Arduino ,όταν είναι με μια εξωτερική πηγή ενέργειας (σε αντιδιαστολή με 5 volt από τη σύνδεση USB ή άλλη ρυθμιζόμενη πηγή ενέργειας (νούμερο 11 στην παρακάτω εικόνα)) .

Αναλογικοί ακροδέκτες

Αναλογικοί ακροδέκτες (νούμερο 15 στην παρακάτω εικόνα) είναι οι συντελεστές όπου θα συνδέσετε τα αναλογικά συστατικά, όπως τα ποτενσιόμετρα και άλλους αισθητήρες. Αν και οι ψηφιακές εισοδοί / έξοδοι λειτουργούν μόνο με 0 και 1 τιμές, οι αναλογικές inputs λειτουργούν με τιμές 0 έως 1023.

Ψηφιακοί ακροδέκτες

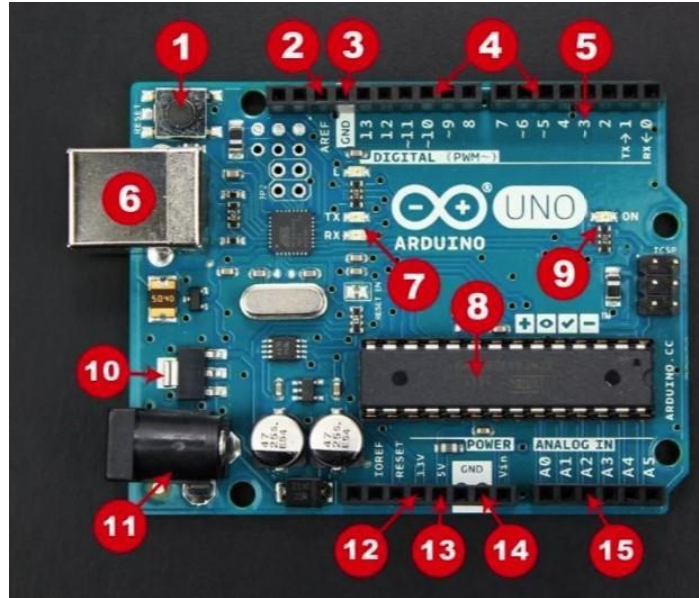
Ψηφιακοί ακροδέκτες (νούμερο 4 στην παρακάτω εικόνα) μπορεί να λειτουργήσουν ως εισροές ή εκροές και να ορίσετε πώς θα λειτουργήσουν με την pinMode λειτουργία (). Οι πείροι που έχουν το "~" μπροστά από τα νούμερα είναι PWM (Pulse Width Modulation) έξοδοι, και μπορούν να μιμηθούν αναλογικής εξόδου με λειτουργία analogWrite (). Όταν χρησιμοποιείτε αυτή τη λειτουργία, διαμορφώνετε το πλάτος του παλμού, μεταβάλλοντας τη συχνότητα max και χαμηλών παλμών (0 ή 1), μιμούμενοι ένα αναλογικό παλμό.

Συνοπτική περιγραφή Arduino

Το Arduino Uno R3 παρουσιάζει τα εξής χαρακτηριστικά:

- Μικροελεγκτής ATMEGA328
- Τάση λειτουργίας 5V
- Τάση εισόδου (συνίσταται) 7-12V
- Όρια τάσης εισόδου 6-20V
- Ψηφιακοί ακροδέκτες I/O 14(6 εκ των οποίων PWM έξοδο)
- Αναλογικοί ακροδέκτες εισόδου 6
- Ισχύς συνεχόμενου ρεύματος ανά ακροδέκτη 40mA
- Ισχύς συνεχόμενου ρεύματος για ακροδέκτη τάσης 3.3V 50mA
- Μνήμη flash 32KB(ATMEGA328)
- Μνήμη SRAM 2KB(ATMEGA328)
- Μνήμη EEPROM 1KB(ATMEGA328)
- Ταχύτητα ρολογιού 16MHz

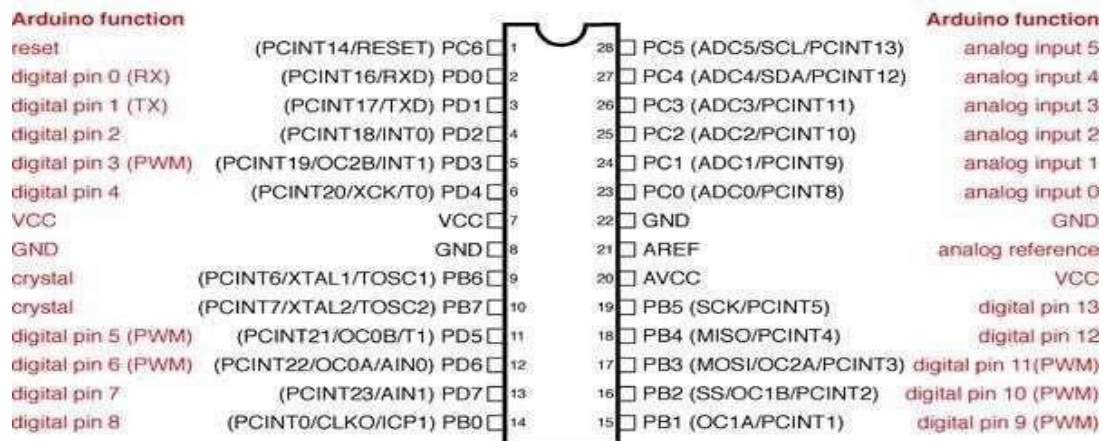
Ανάλυση πλακέτας Arduino UNO



Εικόνα 5 « Πλακέτα Arduino Uno»

1. Κουμπί Reset: Κάνει επανεκκίνηση του κώδικα που έχει φορτωθεί στην πλακέτα Arduino.
2. AREF: Υποστηρίζει "Αναλογική Αναφορά" και χρησιμοποιείται για τη ρύθμιση μιας εξωτερικής τάσης αναφοράς.
3. Ακροδέκτης γείωσης.
4. Ψηφιακή είσοδος / έξοδος: Οι ακροδέκτες 0-13 μπορούν να χρησιμοποιηθούν για ψηφιακή είσοδο ή έξοδο.
5. PWM: Οι ακροδέκτες που επισημαίνονται με το σύμβολο (~) μπορούν να προσομοιάσουν την αναλογική έξοδο.
6. Σύνδεση USB: Χρησιμοποιείται για τον προγραμματισμό ή τροφοδοσία..
7. TX / RX: Ένδειξη LED εκπομπής και λήψης δεδομένων.
8. ATMEGA 328P Μικροελεγκτής: Είναι ο εγκέφαλος που εκτελούνται τα προγράμματα.

9. Ενδεικτική λυχνία ισχύος: Η λυχνία LED ανάβει κάθε φορά που η πλακέτα είναι συνδεδεμένη σε μια πηγή τροφοδοσίας.
10. Σταθεροποιητής τάσης.
11. Υποδοχή Jack DC: Χρησιμοποιείται για την τροφοδοσία του Arduino από τροφοδοτικό.
12. 3.3V ακροδέκτης: Παρέχει 3.3V.
13. 5V ακροδέκτης: Παρέχει 5V.
14. Ακροδέκτης γείωσης
15. Αναλογικοί ακροδέκτες: Αυτοί οι ακροδέκτες μπορούν να διαβάσουν αναλογικό σήμα .

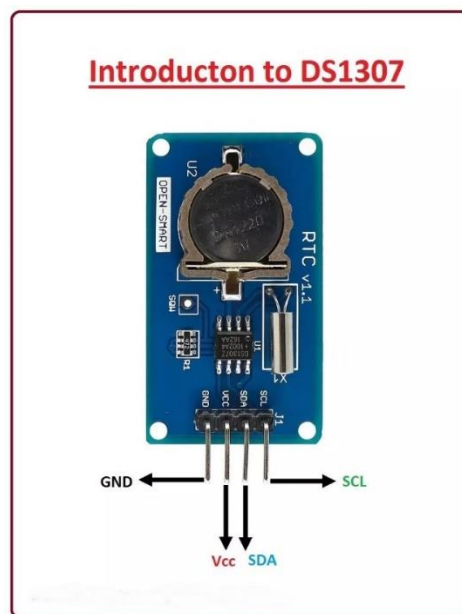


Εικόνα 6 « Arduino UnoPin Connection»

Το Arduino Uno χρειάζεται πηγή τάσης για να λειτουργήσει και μπορεί να τροφοδοτηθεί με διάφορους τρόπους. Τροφοδοτείται απευθείας μέσω υπολογιστή (με USB καλώδιο) ή ακόμα και από μπαταρία 12V. Η κατανάλωση ισχύος είναι 50 mA και το κόστος αγοράς ανέρχεται περίπου στα 12,00 €

2.3 DS1307

Το DS1307 είναι ένα ρολόι που εφόσον ρυθμιστεί μας δείχνει τον πραγματικό χρόνο. Είναι μια συσκευή χαμηλής ισχύος και έχει υποδοχή για μια μπαταρία τύπου 3032, η οποία τροφοδοτεί με ισχύ, όταν η εξωτερική τροφοδοσία δεν λειτουργεί ή είναι απενεργοποιημένη. Λειτουργεί με το πρωτόκολλο I2C. Είναι μια αμφίδρομη συσκευή και μπορεί να στέλνει και να λαμβάνει δεδομένα και από τις δύο πλευρές. Το DS1307 χρησιμοποιείται σε οποιοδήποτε σύστημα το οποίο απαιτεί σταθερή ώρα και ημερομηνία ορισμένων έργων ή εργασιών.



Εικόνα 7 « Πλακέτα DS1307»

- Το DS1307 είναι ένα ρολόι πραγματικού χρόνου (RTC). Στο DS1307, τα δεδομένα μεταφέρονται σε δυαδικό δεκαδικό κωδικό, σε bits. Ο ρυθμός μεταφοράς των δεδομένων ανέρχεται σε 56 byte.
- Η μνήμη που χρησιμοποιείται στο DS1307 είναι NV SRAM. Το NV SRAM είναι βασικά μια μη πτητική μνήμη τυχαίας πρόσβασης. Κατά τη λειτουργία, το NV SRAM είναι αρκετά παρόμοιο με τη στατική μνήμη τυχαίας πρόσβασης (SRAM).
- Το DS1307 είναι μια ηλεκτρονική συσκευή που παίζει ένα ισχυρό ρόλο όταν ενσωματώνεται σε συστήματα που θέλουν να χρησιμοποιήσουν πραγματικό χρόνο. Στα

συστήματα που έχει ενσωματωθεί έχουμε τα πλεονεκτήματα της ώρας και της παρακολούθησης της ημερομηνίας που μπορούν να χρησιμοποιηθούν ώστε να γίνουν εργασίες σε συγκεκριμένο χρόνο .

- Το DS1307 αποτελείται από ένα ενσωματωμένο κύκλωμα εφεδρικής τροφοδοσίας. Ο σκοπός του ενσωματωμένου κυκλώματος εφεδρικής τροφοδοσίας είναι ότι, εάν διακοπεί η τροφοδοσία, τότε θα αλλάξει αυτόματα σε εφεδρική παροχή ρεύματος. Με αυτόν τον τρόπο το κύκλωμά μας παραμένει σε λειτουργική κατάσταση.

- Το πρωτόκολλο στο οποίο λειτουργεί το DS1307 είναι το I2C. Το I2C είναι ένα πρωτόκολλο μιας γραμμής στο οποίο τα δεδομένα μεταφέρονται κομμάτι προς κομμάτι κατά μήκος ενός μόνο καλωδίου.

PIN	Operation
Vcc	+5 v
GND	GND
SDA , SCL	RTC and EEPROM I2C communication
DS	Temperature sensor – 1 wire protocol
BAT	Battery voltage - ADC
SQ	Square wave output (1,4,8,32 KHz)

Πίνακας 3 « Σημεία Σύνδεσης DS1307»

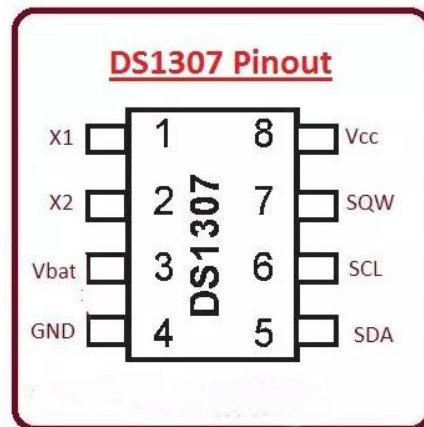
Σημεία σύνδεσης DS130

Το DS1307 έχει συνολικά 8 pinouts, τα οποία περιγράφονται παρακάτω:

- PIN 1,2: Αυτές οι ακίδες είναι για την σύνδεση των κρυστάλλων χαλαζία 32.768. Και οι δύο ακίδες μπορούν να χρησιμοποιηθούν ως είσοδος και έξοδος για εσωτερικό ταλαντωτή. Εάν το X1 είναι είσοδος, τότε το X2 χρησιμοποιείται ως έξοδος.

- PIN 3: Αυτή η ακίδα χρησιμοποιείται για σύνδεση μπαταρίας στο DS1307

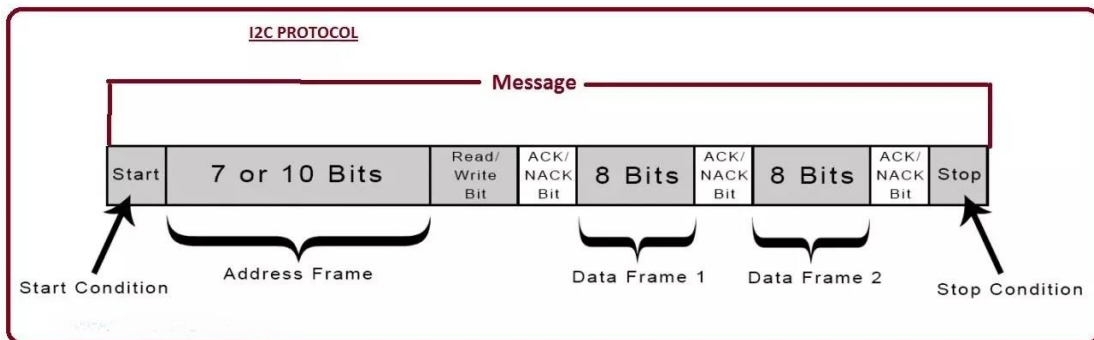
- PIN 4: Η συγκεκριμένη ακίδα είναι για την σύνδεσης του GRD σε αυτό το pin.
- PIN 5: Με αυτή την ακίδα γίνεται η μεταφορά δεδομένο μέσω της ετικέτας SDA, η οποία είναι συντομογραφία της γραμμής δεδομένων σειράς.
- PIN 6: Χρησιμοποιείται για είσοδο σειριακού ρολογιού (SCL) και συγχρονισμό δεδομένων.
- PIN 7: Αυτή η ακίδα χρησιμοποιείται για λήψη τετραγωνικών κυμάτων εξόδου (SQW).
- PIN 8: Σε αυτήν την ακίδα, τροφοδοτείται το ρολόι (Vcc).



Εικόνα 8 « Pin Connection DS1307»

- Το I2C είναι ένα σειριακό πρωτόκολλο στο οποίο τα δεδομένα μεταφέρονται κομμάτι προς bit.
- Το I2C συνδυάζει το καλύτερο συνδυασμό του SPI και του UART. Χρησιμοποιώντας το με έναν μικροελεγκτή μπορούμε να ελέγξουμε πολλές slave συσκευές.

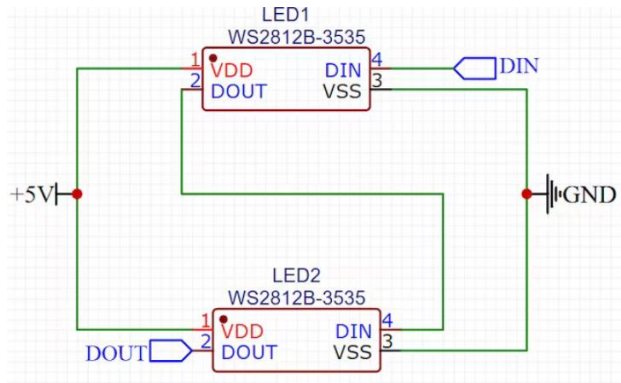
- Στο I2C τα δεδομένα μεταφέρονται με τη μορφή μηνυμάτων, στη συνέχεια μετατρέπουμε τα μηνύματα σε μορφή δεδομένων. Κάθε μήνυμα έχει ένα πλαίσιο διευθύνσεων που περιέχει μια δυαδική διεύθυνση συσκευών που είναι υπό έλεγχο.
- Το πρωτόκολλο I2C είναι φθηνότερο σε εφαρμογή από το πρωτόκολλο SPI. Το SPI ελέγχει μία εξαρτημένη συσκευή ενώ το I2C ελέγχει περισσότερες από μία συσκευές.
- Για καλύτερη κατανόηση, ας ρίξουμε μια ματιά στην εικόνα του πρωτοκόλλου I2C.



Εικόνα 9 « I2C Protocol»

2.4 LED NeoPixel 24

Το LED NeoPixel 24 που διαλέξαμε αποτελείται από 24 έξυπνα LED τα οποία είναι τοποθετημένα σε κύκλο με διάμετρο 66 mm. Οι δακτύλιοι είναι συνδεδεμένοι ως εξής η έξοδος του ενός αποτελεί τον ακροδέκτη εισόδου του επόμενου. Με ένα ακροδέκτη ενός μικροελεγκτή για να ελέγξουμε όσα περισσότερα μπορούμε μεταξύ τους.



Εικόνα 10 « RGB Led Connection»

Κάθε LED είναι διευθυνσιοδοτούμενο καθώς το τσιπ του προγράμματος οδήγησης βρίσκεται μέσα στο LED. Κάθε ένα έχει ~18mA σταθερό ρεύμα κίνησης, οπότε το χρώμα θα είναι πολύ σταθερό, ακόμη και αν η τάση ποικίλλει και δεν απαιτούνται εξωτερικές αντιστάσεις τσοκ, καθιστώντας τη σχεδίαση λεπτή. Τροφοδοσία με 5VDC. Υπάρχει μια ενιαία γραμμή δεδομένων με πρωτόκολλο πολύ συγκεκριμένου χρονισμού. Δεδομένου ότι το πρωτόκολλο είναι πολύ ευαίσθητο στο χρονοδιάγραμμα, απαιτεί έναν μικροελεγκτή πραγματικού χρόνου όπως ένα, Arduino, Raspberry Pico, Microbit. Έτσι δεν μπορεί να χρησιμοποιηθεί με μικροϋπολογιστή που βασίζεται σε Linux όπως το Raspberry Pi ή ερμηνεύσιμο μικροελεγκτή.



Εικόνα 11 « RGB Led Pin»

Τα LED που χρησιμοποιούμε είναι διευθυνσιοδοτούμενα 5mm*5mm WS2812 RGB. Αυτά τα led έχουν ένα ειδικό ενσωματωμένο τσιπ που τους δίνει διεύθυνση για να λάμψουν. Χρησιμοποιώντας το MCU μπορούμε να δημιουργήσουμε μια αναλογική τιμή μεταξύ 0-255 η οποία ορίζεται επίσης στον κώδικα ως φωτεινότητα 0. Στον κώδικα βάζουμε 3 διαφορετικές τιμές όπως (0, 125, 255) , σαν πρώτη τιμή ορίζεται η ένταση του κόκκινου χρώματος , η δεύτερη την ένταση του πράσινου χρώματος και η τρίτη του μπλε αντίστοιχα. Ανάλογα με το διαφορετικό συνδυασμο που θα αναπαραστήσουμε θα παρουμε και το αντίστοιχο χρώμα.

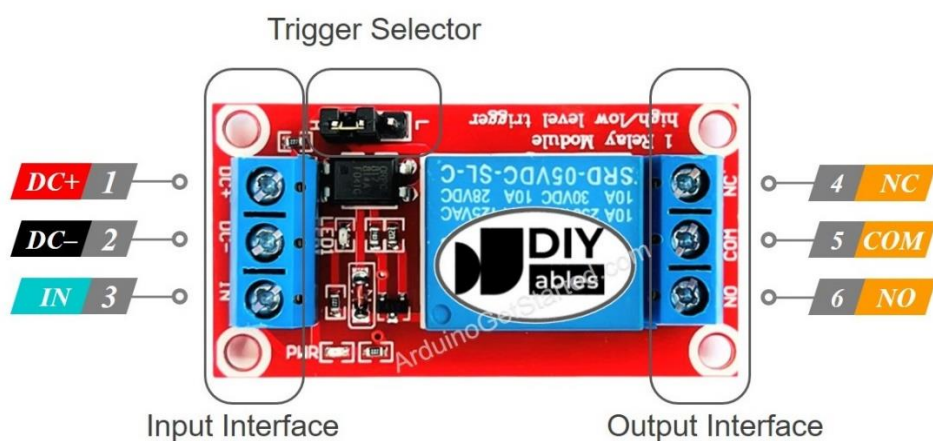
LED Characteristics

	Ref. Value
Quiescent Current	0.7mA
RGB Channel Constant Current	16mA
RED Brightness (Central Value)	600mcd
GREEN Brightness (Central Value)	1200mcd
BLUE Brightness (Central Value)	300mcd
WHITE Brightness (Central Value)	2100mcd
RED Wavelength	620-630nm
GREEN Wavelength	520-530nm
BLUE Wavelength	465-475nm

Εικόνα 12 « RGB Led Τιμές φωτεινότητας και έντασης»

2.4.1 HW-482 5v Relay

Η μονάδα HW-482 είναι κατάλληλα φτιαγμένη για την εφαρμογή της με το Arduino. Διαθέτει τρεις ακίδες, το VCC, το GND και το Signal. Μπορεί να λειτουργήσει ως διακόπτης, εάν το κύκλωμα και το κύκλωμα φορτίου έχουν διαφορετική τάση τροφοδοσίας. Το χρησιμοποιούμε συνήθως εφόσον το κύκλωμα φορτίου είναι AC. Είναι ένας διακόπτης που χρησιμοποιείται για τη σύνδεση απομονωμένης σύνδεσης από το κύκλωμα χρησιμοποιώντας σήμα κυκλώματος. Διαθέτει κόκκινο LED που ανάβει κάθε φορά που βρίσκεται το πηνίο, ενεργοποιείται ή η ακίδα σήματος έχει υψηλή είσοδο.

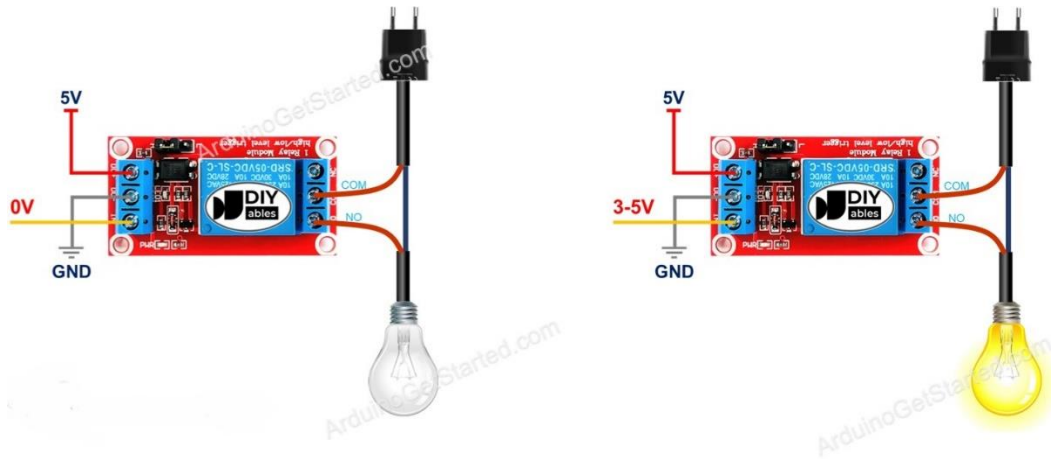


Εικόνα 13 « HW- 482 5v Relay»

Όπως παρατηρούμε και στην εικόνα 13 υπάρχουν δυο ομάδες ακίδων , ομάδα εισόδου (χαμηλή τάση) και η ομάδα εξόδου (υψηλη τάση).

- Οι ακίδες της ομάδας εισόδου συνδέονται με το Arduino και περιέχουν τρεις ακίδες :
- Ακίδα DC – : πρέπει να συνδεθεί με το **GRD (0V)**
- Ακίδα DC + : πρέπει να συνδεθεί με το **VCC (5V)**
- Ακίδα Signal – IN : συνδέεται με το αρθθινο και λαμβάνει σήμα

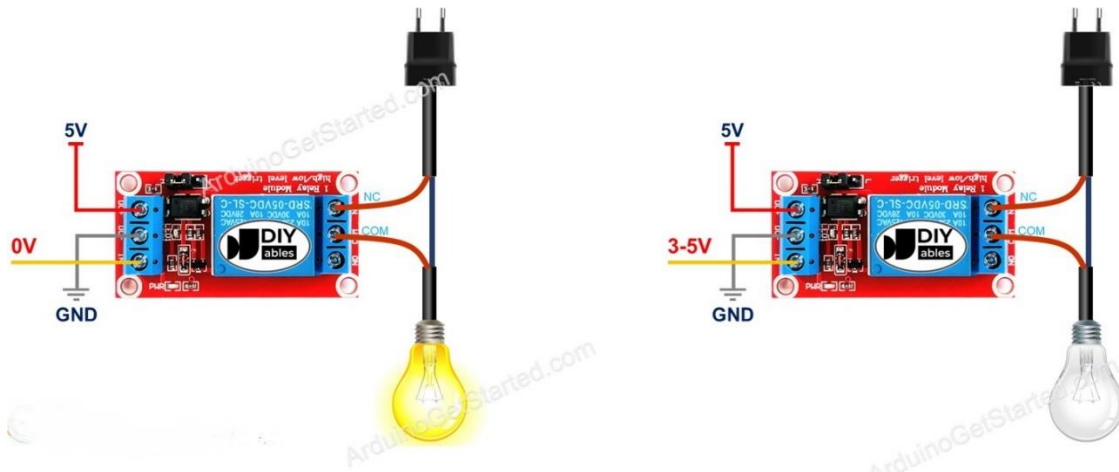
- Οι ακίδες της ομάδας εξόδου συνδέονται με συσκευή υψηλής τάσης και περιέχουν τρεις ακίδες (συνήθως με βιδωτό ακροδέκτη) :



HIGH Level Trigger - Normally Open

Εικόνα 14 « HW- 482 5v Relay Normally Open»

- COM : είναι η κοινή ακίδα , χρησιμοποιείται για να λειτουργήσει σαν κανονικά ανοιχτή ή κανονικά κλειστή.
- Ακίδα NO : Κανονικά ανοιχτή ,χρησιμοποιείται για λειτουργία κανονικά ανοιχτή
- Ακίδα NC : Κανονικά κλειστή ,χρησιμοποιείται για λειτουργία κανονικά κλειστή

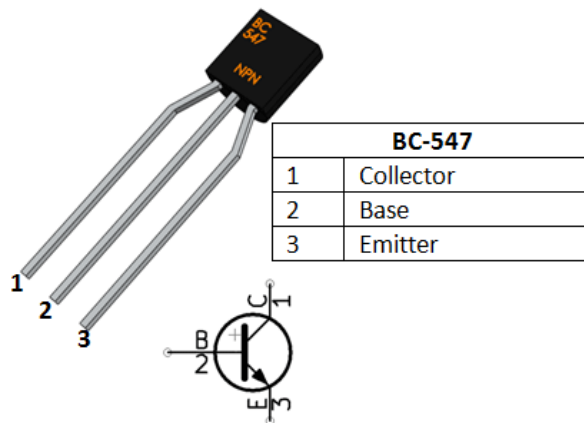


HIGH Level Trigger - Normally Closed

Εικόνα 15 « HW- 482 5v Relay Normally Closed »

Όταν γίνεται εφαρμογή στην πράξη γίνεται χρήση μιας απο τις δυο λειτουργίες

2.4.1 BC547



Εικόνα 16 « BC547 NPN Transistor»

Το BC547 είναι ένα τρανζίστορ NPN, επομένως ο συλλέκτης και ο πομπός θα παραμείνουν ανοιχτοί (Προώθηση αντίστροφης) ,όταν ο ακροδέκτης βάσης συγκρατείται στη γείωση και θα είναι κλειστός (Προωθούμενος προς τα εμπρός) όταν παρέχεται σήμα στον ακροδέκτη βάσης.

1	Collector	Το ρεύμα ρέει μέσω του συλλέκτη
2	Base	Ελέγχει την πόλωση του τρανζίστορ
3	Emitter	Το ρεύμα αποστραγγίζεται μέσω του πομπού

Πίνακας 4 επεξήγησης ακίδων BC547

Η μέγιστη ποσότητα ρεύματος που θα μπορούσε να διαρρέει την ακίδα του συλλέκτη είναι 100 mA, επομένως δεν μπορούμε να συνδέσουμε φορτία που καταναλώνουν περισσότερα από 100 mA χρησιμοποιώντας αυτό το τρανζίστορ. Για να πολώσουμε ένα τρανζίστορ πρέπει να τροφοδοτήσουμε ρεύμα στον ακροδέκτη βάσης, αυτό το ρεύμα (I_B) θα πρέπει να περιορίζεται στα 5 mA.

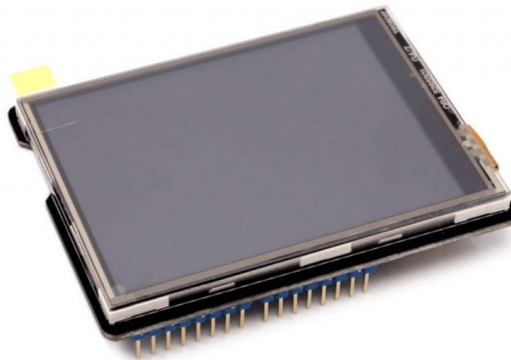
Όταν αυτό το τρανζίστορ είναι πλήρως πολωμένο, τότε μπορεί να επιτρέψει το μέγιστο 100 mA να ρέει κατά μήκος του συλλέκτη και του πομπού. Αυτό το στάδιο ονομάζεται Περιοχή Κορεσμού και η τυπική τάση που επιτρέπεται στον εκπομπό συλλέκτη (VCE) ή τον εκπομπό βάσης (VBE) μπορεί να είναι 200 και 900 mV αντίστοιχα. Όταν αφαιρεθεί το ρεύμα βάσης, το τρανζίστορ απενεργοποιείται πλήρως, αυτό το στάδιο ονομάζεται Περιοχή αποκοπής και η τάση του εκπομπού βάσης θα μπορούσε να είναι περίπου 660 mV. Οπότε κόβοντας το ρεύμα στην βάση ή δίνοντας δουλεύει σαν διακόπτης που δεν επιτρέπει να επιστρέψει μεγάλο ρεύμα πίσω και κάψει τον μικροεπεξεργαστή.

2.4.1 Shields

Οι Shields είναι τυποποιημένες πλακέτες επεκτάσεων κυκλωμάτων που συνδέονται στα κανονικά παρεχόμενα Arduino pin-headers, φυσικά μπορεί κανείς να στοιβάξει αρκετές Shields την μια πάνω από την άλλη για να επιτύχει το Project που έχει σχεδιάσει, αρκεί να υπολογίσει τα Pin που θα χρησιμοποιηθούν να είναι ίδια για όλες τις επεκτάσεις και να μην ζητάμε να αλλάζει από επέκταση σε επέκταση.

Arduino Display Shield

1.8 TFT Shield



Εικόνα 17 « TFT Shield »

Αυτή είναι μια πολυλειτουργική οθόνη αφής συμβατή με Arduino / Seeeduno / Arduino Mega. Μπορεί να χρησιμοποιηθεί ως συσκευή προβολής ή σκίτσο. Σε σύγκριση με την προηγούμενη έκδοση, 2.8" TFT Touch Shield V1.0, αντικαταστήσαμε το πρόγραμμα οδήγησης οθόνης με ένα πιο επαγγελματικό πρόγραμμα οδήγησης, ILI9341, παρέχοντας διαφορετική επικοινωνία SPI εξοικονόμησης pin χωρίς να θυσιάσουμε την ταχύτητα μετάδοσης δεδομένων. Λόγω της αλλαγής της μεθόδου επικοινωνίας, τα προγράμματα που αναπτύχθηκαν για την αρχική έκδοση

χρειάζονται τροποποιήσεις πριν μεταμοσχευτούν στη νέα έκδοση. Με μια μονάδα κάρτας SD ενσωματωμένη επίσης σε αυτήν την ασπίδα, αυτή η ασπίδα διατηρεί μεγάλο χώρο για άλλες επεκτάσεις στο έργο σας.

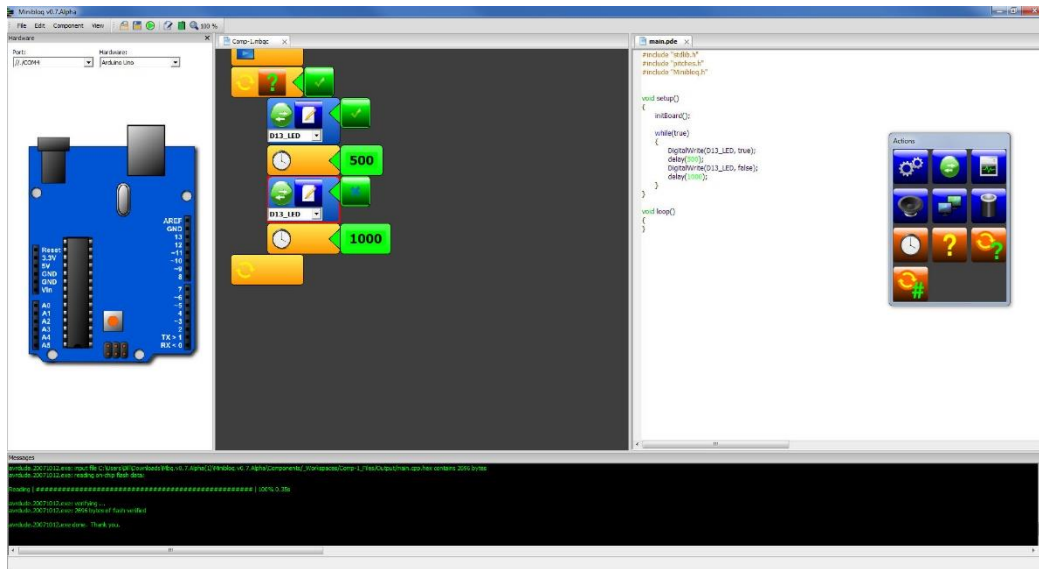
2.5 Πλατφόρμες Ανάπτυξης Λογισμικού

Στην ενότητα αυτή θα περιγράψουμε κάποιες από τις πλατφόρμες που μπορούν να χρησιμοποιηθούν για τον προγραμματισμό των ηλεκτρονικών μονάδων του συστήματος και την σχεδίαση του πλάνου και των υλικών που απαιτήθηκαν για την ολοκλήρωση του έργου. Πρόκειται για γνώριμες πλατφόρμες δωρεάν διανομής ή επιλογή εγγραφής μέσω ιδρύματος πανεπιστημίου και εύκολες στην χρήση και κατανόηση.

2.5.1 Minibloq :

- Γραφικό περιβάλλον
- Windows/Linux/OLPC/Mac OS

Το Minibloq είναι ένα περιβάλλον ανάπτυξης γραφικών για το Arduino^[12] και άλλες πλατφόρμες. Ο κύριος στόχος του είναι να βοηθήσει στη διδασκαλία του προγραμματισμού. Χρησιμοποιείται ειδικά στη ρομποτική σε δημοτικά, γυμνάσια και λύκεια. Χρησιμοποιείται ευρέως στην Αργεντινή, όπου ακριβώς στην επαρχία San Luis, περισσότερα από 60000 παιδιά έχουν εκπαιδευτεί με αυτό το λογισμικό σε δημόσια σχολεία.

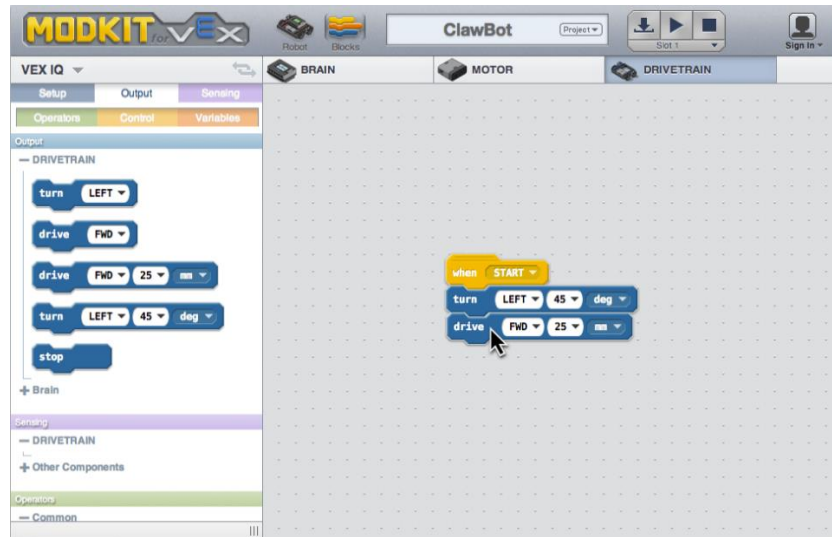


Εικόνα 18 « Miniblog »

2.5.2 ModKit :

- Θυμίζει scratch
- Windows/Linux/Mac OS
- Free online – pay to download

Το Modkit for VEX είναι ένα γραφικό περιβάλλον προγραμματισμού που αναπτύχθηκε ειδικά για το VEX IQ. Βασισμένο στη διεπαφή Scratch του MIT, το Modkit for VEX είναι χτισμένο γύρω από μπλοκ snap-together που κάνουν τον προγραμματισμό πραγματικά προσβάσιμο και διασκεδαστικό για όλες τις ηλικίες και τα επίπεδα δεξιοτήτων. Σχεδιασμένο για να απλοποιεί τον έλεγχο του μηχανισμού σε μορφή διαισθητική για μαθητές, το Modkit for VEX διαθέτει τη δυνατότητα εικονικής διαμόρφωσης του ρομπότ σας μέσα στο λογισμικό. Όταν καθορίσετε ποιους αισθητήρες είναι συνδεδεμένοι και παραμέτρους, όπως οι διαστάσεις της βάσης μετάδοσης κίνησης ή οι τύποι τροχών, το Modkit for VEX θα δημιουργήσει προσαρμοσμένα μπλοκ για το ρομπότ σας (όπως "Στρίψτε 90° Αριστερά") και θα φροντίσει τους περίπλοκους υπολογισμούς (όπως η είσοδος γυροσκοπίου) πίσω από τις σκηνές.

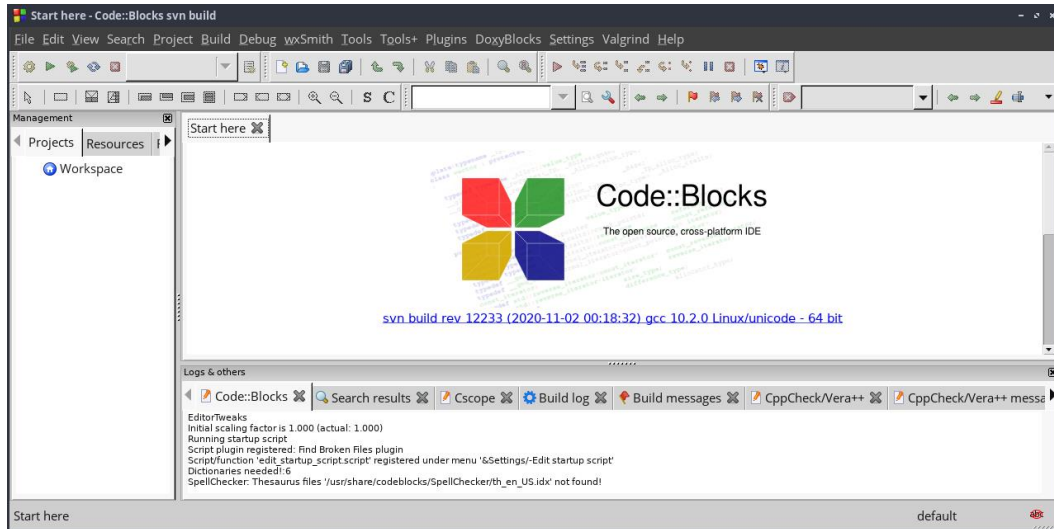


Εικόνα 19 ModKit »

2.5.3 Code::Blocks :

- Θυμίζει scratch
- Windows/Linux/Mac OS
- Free to download

Το Code::Blocks είναι ένα δωρεάν λογισμικό προγραμματισμού ανοιχτού κώδικα στις γλώσσες C/C++ και Fortran IDE. Παρέχει πληθώρα βασικών συναρτήσεων και μεταγλωττιστές με αποτέλεσμα την εύκολη και γρήγορη σύνθεση και εκτέλεση κώδικα. Έχει σχεδιαστεί για να είναι επεκτάσιμο και πλήρως διαμορφώσιμο. Οποιοδήποτε είδος λειτουργικότητας μπορεί να προστεθεί εγκαθιστώντας ή κωδικοποιώντας ένα πρόσθετο. Για παράδειγμα, η λειτουργία μεταγλώττισης και εντοπισμού σφαλμάτων παρέχεται από πρόσθετα.



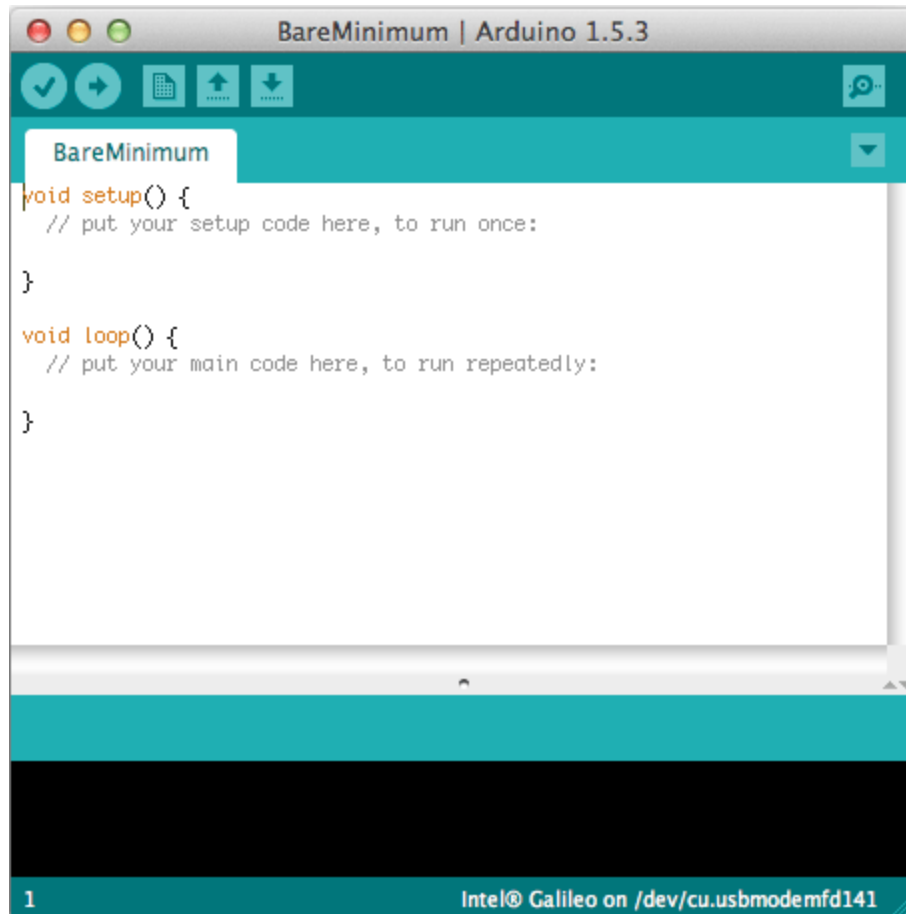
Εικόνα 20 « Code Blocks »

Εικόνα

2.5.4 Arduino IDE (Integrated Development Environment)

- Βασίζεται στην C
- Windows/Linux/Mac OS
- Ελεύθερο – Ανοικτού κώδικα

Το Arduino IDE είναι το λογισμικό της αναπτυξιακής πλατφόρμας Arduino. Πρόκειται για ένα ολοκληρωμένο προγραμματιστικό περιβάλλον, μέσα στο οποίο μπορούμε να συντάξουμε τα προγράμματά μας σε γλώσσα Wiring C και στη συνέχεια να τα φορτώσουμε στην εκάστοτε πλακέτα.



Εικόνα 21 « Arduino IDE »

Είναι και το λογισμικό που χρησιμοποιήθηκε για την υλοποίηση αυτής της εργασίας . Παρακάτω θα εξηγήσουμε λεπτομερέστερα την πλατφόρμα.

Τα κουμπιά της γραμμής εργαλείων:



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



Upload: Ανέβασμα του κώδικα στον μικροελεγκτή



New (Νέο): Δημιουργεί ένα νέο sketch



Open (Άνοιγμα): Παρουσιάζει ένα μενού με όλα τα sketch, κάνοντας κλικ σε ένα από αυτά θα ανοίξει μέσα στο τρέχον παράθυρο



Save (Αποθήκευση): Αποθηκεύει το sketch



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

Η Δομή του προγράμματος

Ένα τυπικό πρόγραμμα Arduino έχει την παρακάτω δομή:

```
void setup () //δήλωση μεταβλητών
{
  //αρχικοποιήσεις
}
void loop ()
{
  //Κώδικας
}
```

Υπάρχουν δυο ειδικές συναρτήσεις που είναι μέρος του κάθε sketch του Arduino οι οποίες είναι η `setup()` και η `loop()`. Η `setup()` καλείται μια φορά, όταν το sketch ξεκινά ή όποτε κάνει επαναφορά (reset) η πλατφόρμα Arduino. Κυρίως, σε αυτήν γίνονται οι αρχικοποιήσεις των μεταβλητών, η ρύθμιση της κατάστασης των ακίδων (pins) και η προετοιμασία των βιβλιοθηκών. Αντιθέτως, η συνάρτηση `loop()` καλείται ξανά και ξανά επιτρέποντας έτσι στο πρόγραμμα να ανταποκριθεί σε εξωτερικά ερεθίσματα. Και οι δυο συναρτήσεις πρέπει να περιλαμβάνονται στο sketch, ακόμα και αν δεν περιέχουν κάτι και να είναι κενές.

Βασικές δομές και λειτουργίες προγραμματισμού:

Παρακάτω, ακολουθούν μερικές από τις πιο βασικές δομές και λειτουργίες που μπορεί να αξιοποιηθούν ως εργαλεία κατά την συγγραφή ενός προγράμματος Arduino :

Δομές ελέγχου ροής:

if	(δομή ελέγχου μίας συνθήκης)
if ... else	(δομή ελέγχου πολλαπλών συνθηκών)
for	(δομή επαναληπτικού ελέγχου συνθήκης)
while	(δομή επαναληπτικού ελέγχου συνθήκης)
do ... while	(δομή επαναληπτικού ελέγχου συνθήκης)
switch ... case	(δομή ελέγχου περιπτώσεων)
break	(εντολή διακοπής μιας επαναληπτικής δομής)
continue	(εντολή παράλειψης της τρέχουσας επανάληψης)
return	(εντολή επιστροφής από μία συνάρτηση)
goto	(εντολή μετάβασης σε κάποιο σημείο του κώδικα)

Όπως παρατηρούμε από τις δομές ελέγχου ροής ότι η βασικές εντολές μοιάζουν πολύ με τις εντολές της C++.

Αφού γράψουμε τον κώδικα που θέλουμε και συνδέσουμε το Arduino με την θύρα Usb μπορούμε να ‘ανεβάσουμε’ το πρόγραμμα και μας μένει μόνο να συνδέσουμε σωστά τα καλώδια στα Pin του Arduino και στο Breadboard ή στα Shields που θα χρειαστεί το πείραμα μας, ώστε να κάνουμε τις δοκιμές μας.

2.6 Σύνοψη Κεφαλαίου

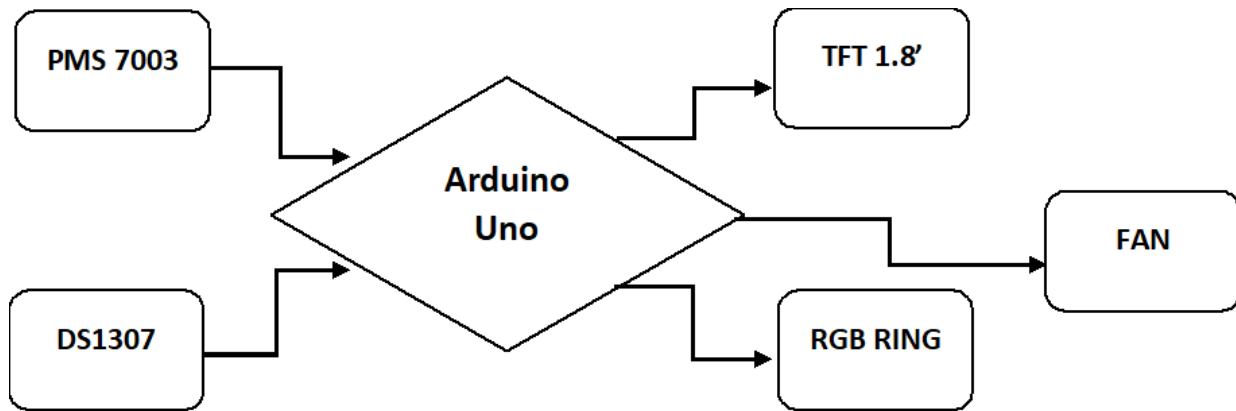
Η θεωρία αποτελεί το θεμέλιο σε μία κατασκευή συστήματος και η κατανόηση της για την εφαρμογή της μεθοδολογίας αναπόσπαστο κομμάτι. Την ανάλυση της θεωρίας πίσω από την υλοποίηση διαδέχεται το κεφάλαιο 3, στο οποίο αναφέρονται και περιγράφονται τα χαρακτηριστικά των φυσικών υλικών και ηλεκτρονικών μονάδων που χρησιμοποιήθηκαν στην παρούσα εργασία. Το κεφάλαιο 3 χωρίζεται σε υποενότητες.

Κεφάλαιο 3 – Συναρμολόγηση του συστήματος μέτρησης

Ως σύστημα μέτρησης αναφερόμαστε στα φυσικά εξαρτήματα που απαρτίζουν το σύστημά μας. Σε αυτή την ενότητα γίνεται αναφορά στη διαδικασία που ακολούθησα για την συναρμολόγηση και κατασκευή του συστήματος ανίχνευσης μικροσωματιδίων, στο λόγο που επιλέχθηκαν τα συγκεκριμένα υλικά καθώς και το λόγο ύπαρξής τους στο τελικό αποτέλεσμα.

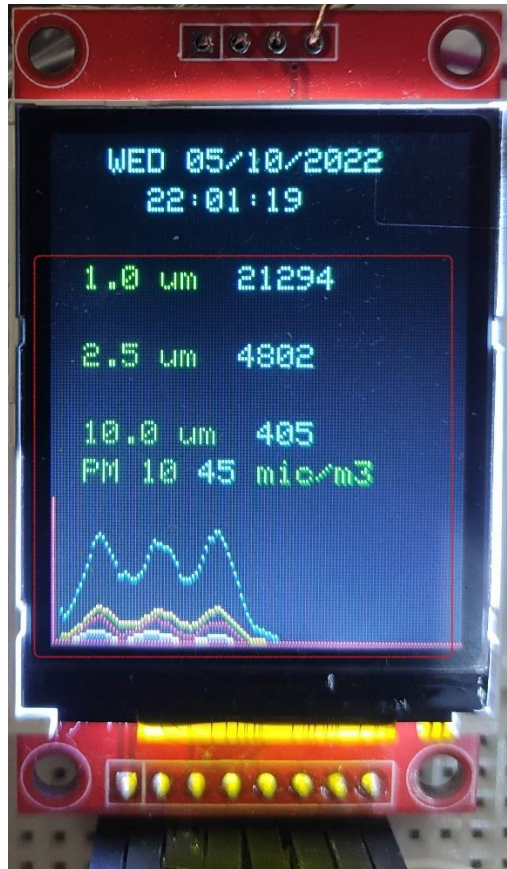
3.1 Το ηλεκτρικό κύκλωμα του συστήματος

Στο παρακάτω διάγραμμα απεικονίζεται η συνδεσμολογία του συστήματος που θα την επεξηγήσουμε στην συνέχεια.



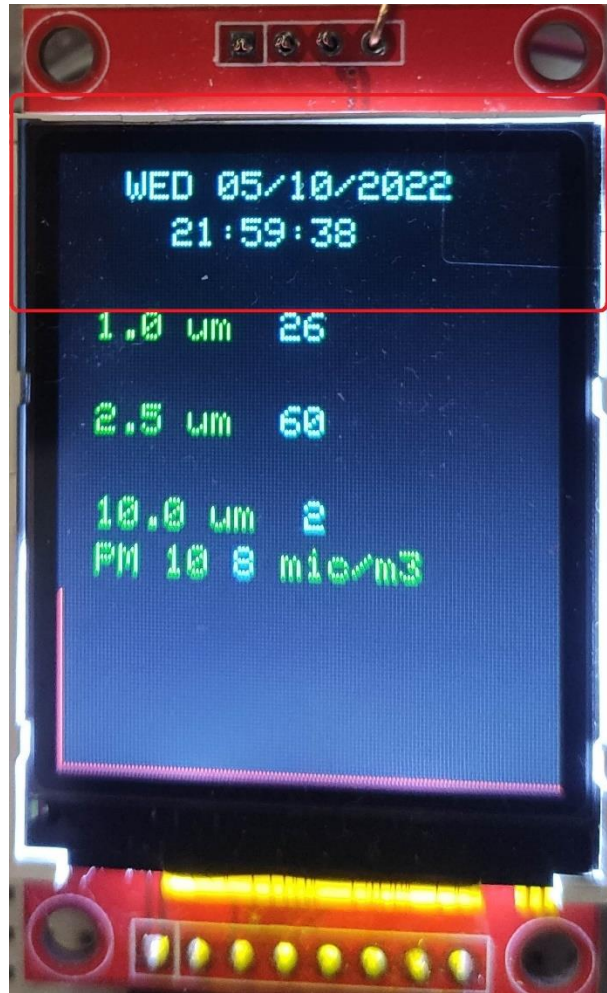
Εικόνα 22 « Σχεδιάγραμμα εργασίας»

Το Arduino που είναι ο εγκέφαλος του συστήματος παρατηρούμε ότι απο τις ψηφιακές θύρες 3 (TX) - 4 (RX) που είναι οι θύρες με τον αισθητήρα μέτρησης μικροσωματιδίων PMS7003 γίνεται η αμφίδρομη επικοινωνία. Με τις μετρήσεις να αποθηκεύονται στην SD Card η οποία είναι ενσωματωμένη πάνω στην πλακέτα της οθόνης. Τα δεδομένα «διαβάζονται» από το αρχείο που είναι στην SD Card και απεικονίζονται στην αντίστοιχη γραμμή που έχουμε ορίσει για την κάθε τιμή.



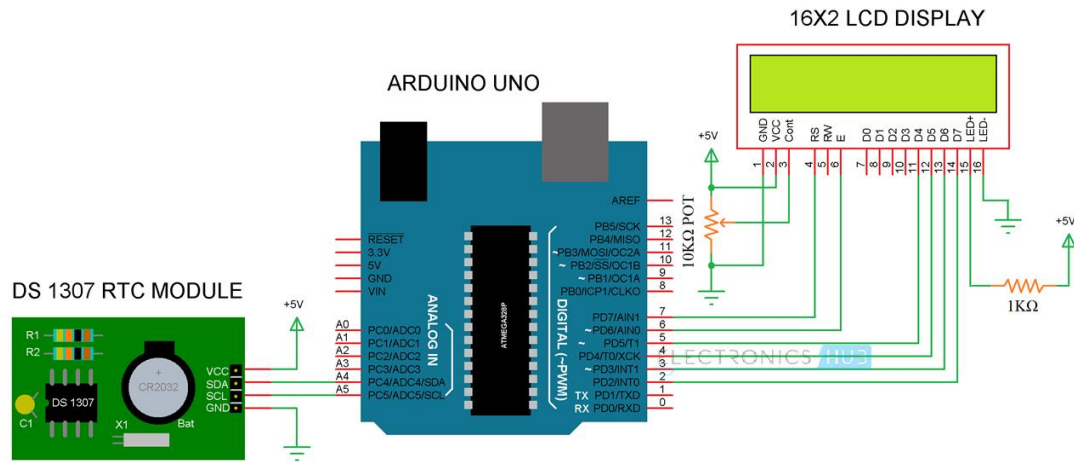
Εικόνα 23 « Απεικόνιση των μετρήσεων»

Ανάλογα με την μεταβολή των τιμών δημιουργείται και το γράφημα το οποίο έχουμε ορίσει να εμφανίζεται κάτω απο το σύνολο των μετρήσεων, με διαφορετικό χρώμα για κάθε τιμή.



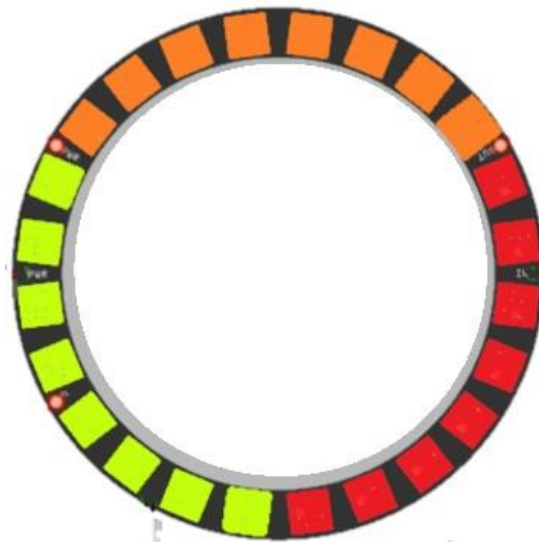
Εικόνα 24 «Ενδειξη Ημερομηνίας και ώρας σε πραγματικό χρόνο »

Πάνω απο τις ενδείξεις των μετρήσεων παρατηρούμε ότι υπάρχει ένα ρολόι με την υφιστάμενη ώρα και την τρέχουσα ημερομηνία που τη διαβάζει από τον αντάπτορα DS1307 που αναφέραμε παραπάνω ,του οποίου η συνδεσμολογία απεικονίζεται στις παρακάτω φωτογραφίες και είναι η εξής:



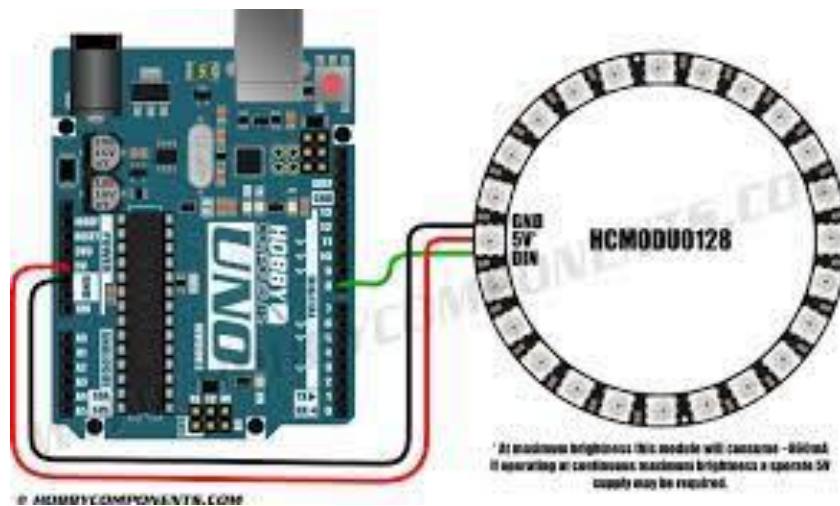
Εικόνα 25 « Συνδεσμολογία ρολόι DS1307»

Στο σύστημα έχουμε προσθέσει ένα RGB Ring 24 Led στο οποίο έχουμε χωρίσει να αλλάζει το χρώμα των led σε ομάδες των 8, π.χ. αν είναι οι μετρήσεις εντός των υποφερτών ορίων ανάβουν τα 8 πρώτα led (1-21850), αν οι μετρήσεις είναι μεταξύ των μετρήσεων (21850-43700) ο χρωματισμός αλλάζει σε πορτοκαλί, και αν ξεπεράσει τις μετρήσεις των (43701) τότε αλλάζουν σε κόκκινο. Στην φωτογραφία παρακάτω έχουμε ένα δείγμα:



Εικόνα 26 « NeoPixel RGB Ring φωτεινές ενδείξεις»

Το RGB Ring έχει 3 καλώδια τι +5v το GND και ένα που μεταφέρει τα δεδομένα και πηγαίνει στο DIN των Led. Παρακάτω βλέπουμε την συνδεσμολογία.



Εικόνα 27 « Συνδεσμολογία NeoPixel RGB Ring»

Κεφάλαιο 4 – Κώδικας Λειτουργίας

Παράθεση του κώδικα που χρησιμοποιήθηκε για την υλοποίηση της πτυχιακής εργασίας και φ. ορτώθηκε στη εργασία για να παρατηθεί.

```
#include <RTClib.h> //Βιβλιοθήκη για το ρολόι DS1037
#include <TFT.h> //Βιβλιοθήκη για την οθόνη
#include <SoftwareSerial.h> //Βιβλιοθήκη σειριακής επικοινωνίας
#include <Wire.h> // Βιβλιοθήκη για I2C

RTC_DS1307 RTC;

#include <Adafruit_NeoPixel.h> //Βιβλιοθήκη για το LED NeoPixel Ring

// Which pin on the Arduino is connected to the NeoPixels?
#define PIN.....6 //Προσδιορισμός Ακίδας επικοινωνίας για το LED NeoPixel Ring
// On Trinket or Gemma, suggest changing this to 1

// How many NeoPixels are attached to the Arduino?
#define NUMPIXELS 24 // Προσδιορισμός αριθμού LED Που έχει το LED NeoPixel Ring
Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
/*Εντός της παρένθεσης. ο πρώτος χαρακτηρισμός NUMPIXELS = δίνουμε το αριθμό των Led που έχει η ταινία
ή το Δακτυλίδι
// Ο δεύτερος χαρακτηρισμός PIN = δίνουμε την ακίδα στην οποία θα επικοινωνεί το με το Arduino το LED
Ring
// Ο τρίτος χαρακτηρισμός ακολουθεί ως εξής:
// NEO_KHZ800 800 KHz η ταχύτητα που θα έχει η ροή δεδομένων επικοινωνίας (συνήθως για τα προϊόντα
NeoPixel w/WS2812 LEDs)
// NEO_KHZ400 400 KHz η ταχύτητα που θα έχει η ροή δεδομένων επικοινωνίας (classic 'v1' για τα προϊόντα,
WS2811 drivers)
// NEO_GRB... Η συνδεσμολογία των LED είναι για GRB bitstream (για προϊόντα NeoPixel )
// NEO_RGB... Η συνδεσμολογία των LED είναι για RGB bitstream (v1 για προϊόντα FLORA)
// NEO_RGBW... Η συνδεσμολογία των LED είναι για RGBW bitstream (για προϊόντα NeoPixel RGBW) */

uint32_t green = pixels.Color(0, 250, 0); //Προσδιορισμός του πράσινου χρώματος
uint32_t orange = pixels.Color(250, 250, 0); //Προσδιορισμός του πορτοκαλί χρώματος
uint32_t red = pixels.Color(255, 0, 0); //Προσδιορισμός του κόκκινου χρώματος

SoftwareSerial pmsSerial(2, 3); // Θύρες σειριακής επικοινωνίας
```

```
#define cs 10 // Προσδιορισμός του CS
#define dc 8 // Προσδιορισμός του DC
#define rst 9 // Προσδιορισμός του reset ή μπορεί να συνδεθεί το Reset του Arduino

#include <SPI.h> // Βιβλιοθήκη για SPI
#include <SD.h> // Βιβλιοθήκη για την SD card
const int cs_sd=4;
int temps; // Χρόνος απόκρισης σε ακέραιο αριθμό
double tempsInit; // Να αρχίσει το μέτρηση του χρόνου μετά τον βρόγχο loop()

#ifdef __SAM3X8E__
... #undef __FlashStringHelper::F(string_literal)
... #define F(string_literal) string_literal
#endif

// Option 1: use any pins but a little slower
// Adafruit_ST7735 tft = Adafruit_ST7735(cs, dc, mosi, sclk, rst);

// Option 2: must use the hardware SPI pins
// (for UNO that's sclk = 13 and sid = 11) and pin 10 must be
// an output. This is much faster - also required if you want
// to use the microSD card (see the image drawing example)
Adafruit_ST7735 tft = Adafruit_ST7735(cs, dc, rst); // Ορίζοντας το tft ότι είναι το Adafruit_ST7735 και η
συνδεσμολογία όπως είναι προσδιορισμένη στην παρένθεση.
float number_leds=0; // Δεκαδικός αριθμός που ορίζει τα δεδομένα number_leds
int x=2; // Προσδιορισμός του ακέραιου αριθμού που ορίζει το x
void setup() // Αρχή του βρόγχου setup
{
.. pinMode(5, OUTPUT); // Ορίζει την ψηφιακή ακίδα 5 σαν έξοδο
Serial.begin(9600); // Προσδιορίζει την εκκίνηση για εύρος επικοινωνίας του σειριακού πρωτόκολου
.. Wire.begin(); // Προσδιορίζει την εκκίνηση του I2C πρωτόκολου
.. RTC.begin(); // εκκίνηση της μονάδας του RTC

Serial.print("init SD"); // στείλε σειριακά ότι είναι στην παρένθεση
.. delay(100); // αναμονή 100ms
.. if(!SD.begin(cs_sd)) // Έλεγχος εάν υπάρχει η κάρτα SD στη συσκευή
```

```
..{
..Serial.print("Defaut SD");// Επιστροφή σε περίπτωση που δεν βρεί κάρτα SD
..return;
..}
..Serial.print("Carte SD.OK");// Επιστροφή OK σε περίπτωση που βρεί κάρτα SD

..File data = SD.open("donnees.txt",FILE_WRITE);.....// Δημιουργεί ένα αρχείο "data.txt" στην κάρτα SD
και το ανοίγει
..data.println(""); data.println("Dmarrage acquisition");// Γράφει στο αρχείο
..data.close();

..tft.initR(INITR_BLACKTAB);//Εκκίνηση του ST7735S chip με μύυρο φόντο
..Serial.println("init");// εξαγωγή εφόσον δεν υπάρχει κάποια πρόβλημα

tft.fillScreen(ST7735_BLACK);...// Να γεμίσει όλη η οθόνη με μύυρο
..pmsSerial.begin(9600);...// Η επικοινωνία με τον αισθητήρα ορίζεται στα 9600

..pixels.begin();//Εκκίνηση του αντικειμένου NeoPixel (Απαραίτητο)
..pixels.setBrightness(2);// Ορισμός της φωτεινότητας των LED του NeoPixel

}

struct pms7003data //Ομαδοποίηση των διαφορετικών μεταβλητών που απαρτίζουν τα δεδομένα του
αισθητήρα pms7003
{
..uint16_t framelen;// Πόσο είναι το μήκος δεδομένων.
..uint16_t pm10_standard, pm25_standard, pm100_standard;//Ορισμός των μεταβλητών
..uint16_t pm10_env, pm25_env, pm100_env;//Ορισμός των μεταβλητών
..uint16_t particles_03um, particles_05um, particles_10um, particles_25um, particles_100um;
..uint16_t unused;
..uint16_t checksum;
};

struct pms7003data data; //Προσδιορισμός της ομαδοποίησης που κάναμε παραπάνω σε ομάδα data

void loop() //Αρχή του βρόγχου loop
```

```
{

..pixels.clear(); // "Καθάρισε" όλα τα pixels, δηλ σβήσε ότι είναι στην οθόνη.

..DateTime now = RTC.now(); // Διάβασε την ώρα που δείχνει αυτήν την στιγμή το ρολόι

..poster_date_time(now);

.. temps = ((millis() - tempsInit))/1000; // Έναρξη του χρονομέτρου

..if (readPMSdata(&pmsSerial)) //Εφόσον διαβάσεις τα δεδομένα του αισθητήρα
δειξε στην οθόνη αυτά που ζητάει κάθε εντολή
{

...tft.setCursor(10, 17); // Στην θέση (10, 17) της οθόνης
..tft.setTextColor(ST7735_GREEN,ST7735_BLACK); //Τα γράμματα με πράσινο χρώμα ( GREEN,ST7735) και
ότι δεν γράφεται να είναι μαύρο ( ST7735_BLACK)
..tft.setTextSize(1); //Μέγεθος Γραμματοσειράς

..tft.setCursor(10, 29);
..tft.print("0.3 um.");tft.setTextColor(ST7735_CYAN, ST7735_BLACK);tft.print(data.particles_03um);
// Εκτύπωσε στην οθόνη με χαρακτηρισμό ("0.3 um.") και κυανό χρώμα τα δεδομένα για της μετρήσεις των
0.3 μικροχωματισίων.
..curves(); //Απο εδώ θα διαβάζει για να βγούνε οι καμπύλες του γραφήματος

tft.setCursor(10, 41);
..tft.setTextColor(ST7735_GREEN,ST7735_BLACK);
..tft.setTextSize(1);
..tft.print("0.5 um.");tft.setTextColor(ST7735_CYAN, ST7735_BLACK);tft.print(data.particles_05um);

tft.setCursor(10, 53);
..tft.setTextColor(ST7735_GREEN,ST7735_BLACK);
..tft.setTextSize(1);
..tft.print("1.0 um.");tft.setTextColor(ST7735_CYAN, ST7735_BLACK);tft.print(data.particles_10um);

tft.setCursor(10, 65);
```

```
..tft.setTextColor(ST7735_GREEN,ST7735_BLACK);
..tft.setTextSize(1);
..tft.print("2.5 um.");tft.setTextColor(ST7735_CYAN, ST7735_BLACK);tft.print(data.particles_25um);

/*tft.setCursor(10, 77);
..tft.setTextColor(ST7735_GREEN,ST7735_BLACK);
..tft.setTextSize(1);
..tft.print("5.0 um.");tft.setTextColor(ST7735_CYAN, ST7735_BLACK);tft.print(data.particles_50um);*/

tft.setCursor(10, 89);
..tft.setTextColor(ST7735_GREEN,ST7735_BLACK);
..tft.setTextSize(1);
..tft.print("10.0 um.");tft.setTextColor(ST7735_CYAN, ST7735_BLACK);tft.print(data.particles_100um);

..tft.setCursor(10, 101);
..tft.setTextColor(ST7735_GREEN,ST7735_BLACK);
..tft.setTextSize(1);
..tft.print("PM 10.");tft.setTextColor(ST7735_CYAN,
ST7735_BLACK);tft.print(data.pm100_standard);tft.setTextColor(ST7735_GREEN, ST7735_BLACK);tft.print("
mic/m3.");

..tft.drawLine(0,113,0,158,ST7735_RED);// Δημιουργεί μια γραμμή με αρχή το (0,113) και τέλος (0,158) και
χρώματος κόκκινου
..tft.drawLine(0,158,128,158,ST7735_RED);// Δημιουργεί μια γραμμή με αρχή το (0,158) και τέλος
(128,158) και χρώματος κόκκινου

number_leds =int (((float (data.particles_03um)/65535)*24)); //Ορισμός του number_leds που θα είναι ένας
ακέραιος αριθμός αλλά θα βγαίνει από τον δεκαδικό αριθμό των δεδομένων του αισθητήρα για τα 0.3 Micro
δια το 65535 (διότι το έχουμε ορίσει σαν 16 bit) και το αποτέλεσμα τα πολλαπλασιάζεται με το 24 που είναι
ο αριθμός των LED του NeoPixel Ring.
//number_leds =(8);
Serial.println (number_leds);
```

```
...if ((number_leds<=8) and (number_leds>=1))// Αν το αποτέλεσμα του number_leds είναι μικρότερο ή ίσο
με 8
{
...pixels.fill(green, 0, number_leds); // Γέμισε όλα τα Leds με το χρώμα πράσινο από το πρώτο που ορίζεται
σαν 0 μέχρι τον αριθμό που έχει βγει από τα δεδομένα.
}

...else if ((number_leds<=16) and (number_leds>=8)) // Αν το αποτέλεσμα του number_leds είναι
μικρότερο ή ίσο με 16 και είναι μεγαλύτερο ή ίσο με 8 τότε:

{
....pixels.fill(green, 0, 8); // Γέμισε τα Leds από το 0 έως το όγδοο με το χρώμα πράσινο
....pixels.fill(orange, 8, ((number_leds)-8)); // Γέμισε με το χρώμα πορτοκαλί από το ένατο led αφού
ξεκινάμε να μετράμε από το 0 μέχρι το αποτέλεσμα του number_leds μείον 8

...}
...else if (number_leds>16) // Αλλιώς αν το number_leds είναι μεγαλύτερο του 16
{

....pixels.fill(green, 0, 8); //Γέμισε απο το πρώτο έως το όγδοο με πράσινο χρώμα
....pixels.fill(orange, 8, 8); //Γέμισε απο το ένατο + οχτώ ακόμα με πορτοκαλί χρώμα
....pixels.fill(red, 16, ((number_leds)-16)); // Γέμισε με το χρώμα κόκκινο από το δέκατο έβδομο led αφού
ξεκινάμε να μετράμε από το 0 μέχρι το αποτέλεσμα του number_leds μείον 16 και ανάβουμε τα αντίστοιχα
...}
...else if (number_leds<=1)//Αλλιώς αν το number_leds είναι μικρότερο ή ίσο του 1
{
....pixels.fill(green, 0, 1); //Γέμισε απο το πρώτο πράσινο χρώμα
...}
...pixels.show(); // Στείλε τα δεδομένα για να εμφανιστούν στο NeoPixel Ring.

// Ορισμός Δεδομένων σε string το οποίο αποθηκεύεται σαν κείμενο για κάθε μέτρηση του αισθητήρα.
String PM03=String(data.particles_03um);
String PM05=String(data.particles_05um);
String PM10=String(data.particles_10um);
```

```
..String PM25=String(data.particles_25um);
..String PM100=String(data.particles_100um);
..String PMS10=String(data.pm10_standard);
..String PMS25=String(data.pm25_standard);
..String PMS100=String(data.pm100_standard);

..String Temps=String(temps);// Ορίζουμε το String Temps σαν κείμενο το οποίο παίρνει τα δεδομένα
"διαβάζοντας" το {temps}

..File data=SD.open("donnees.txt",FILE_WRITE); //Γράφουμε τα δεδομένα στην SD κάρτα στο αρχείο που
ορίζουμε
..data.println( Temps + " " + PM03+ " " + PM05 + " " + PM10+ " " + PM25+ " " + PM100+ " " + PMS10+ " " + PMS25+ "
"+PMS100+ " ");
..data.close(); // Κλείνουμε το αρχείο

..}

}

boolean readPMSdata(Stream *s) // Ελέγχει αν "διαβάζοντας" τα δεδομένα του αισθητήρα που τα παίρνει
μέσος από την κάρτα Sd υπάρχουν
{
..if (!s->available()) // Αν δεν βρεί δεδομένα επιστρέφει false
{
...return false;
..}

..if (s->peek() != 0x42) // Διαβάζει ένα byte την φορά μέχρι να φτάσει στην διεύθυνση '0x42' που είναι η
εκκίνηση των δεδομένων
{
...s->read();
...return false;
}
```

```
..}

..if (s->available() < 32) // Διαβάζει όλα τα 32 bytes
{
...return false;
..}

..uint8_t buffer[32];
..uint16_t sum = 0;
..s->readBytes(buffer, 32);

..// get checksum ready
..for (uint8_t i=0; i<30; i++) {
...sum += buffer[i];
..}

...uint16_t buffer_u16[15]; //Ο κώδικας εδώ έχει κάποιο τρόπο endian'd που το καθιστά να δουλεύει σε όλες
τις πλατφόρμες.
..for (uint8_t i=0; i<15; i++) {
...buffer_u16[i] = buffer[2 + i*2 + 1];
...buffer_u16[i] += (buffer[2 + i*2] << 8);
..}

..memcpy((void *)&data, (void *)buffer_u16, 30); // Δημιουργούμε μια ομαδοποίηση των δεδομένων του
data και του buffer.

..if (sum != data.checksum) // Αν το αποτέλεσμα του data.checksum είναι διαφορετικό αυτού που έχουμε
ορίζει γυρνάει σφάλμα
{
...Serial.println("Checksum failure"); // Σε περίπτωση σφάλματος εκτυπώνει "Checksum failure"
...return false;
..}
```



```
..return true; // Όλα καλά συνεχίζει το πρόγραμμα χωρίς σφάλμα

}

String give_day_week(uint8_t j) //Μετατρέπουμε την ημέρα από αριθμό σε μέρα της εβδομάδας αχρίζοντας
από την Κυριακή
{
..switch(j){
..case 0: return "SUN";
..case 1: return "MON";
..case 2: return "TUE";
..case 3: return "WED";
..case 4: return "THU";
..case 5: return "FRI";
..case 6: return "SAT";
..default: return " ";
..}
}

void poster_date_time(DateTime datetime) //Εκτυπώνει την ημερομηνία και την ώρα στην οθόνη

{

..// Ημερομηνία
..String day = give_day_week(datetime.dayOfTheWeek()) + " " +
.....Vers2Chiffres(datetime.day()) + "/" +
.....Vers2Chiffres(datetime.month()) + "/" +
.....String(datetime.year(), DEC);

..// Ώρα
..String hour = "";
..hour = Vers2Chiffres(datetime.hour()) + ":" +
.....Vers2Chiffres(datetime.minute()) + ":" +
.....Vers2Chiffres(datetime.second());

..File data=SD.open("donnees.txt", FILE_WRITE); // Γράφει την ώρα και την ημερομηνία στην SD κάρτα
```

```
..data.print(day + "." + hour + ".");
..data.close();

..tft.setCursor(18,5);
..tft.setTextColor(ST7735_GREEN,ST7735_BLACK);
..tft.setTextSize(1);
..tft.setTextColor(ST7735_CYAN,ST7735_BLACK);
..tft.print(day);tft.setCursor(30,17);tft.setTextColor(ST7735_CYAN,ST7735_BLACK);tft.print(hour);

}

String Vers2Chiffres(byte number) //Εκτυπώνει διψήφιο αριθμό
{
..String result = "";
..if(number < 10)
...result = "0";
..return result += String(number,DEC);
}

void curves()// Δεδομένα για δημιουργία καμπύλων γραφήματος
{

..int newValue03;
..int newValue05;
..int newValue10;
..int newValue25;
..int newValue100;

..newValue03 = map((data.particles_03um),0,65535,158,113);
..newValue05 = map((data.particles_05um),0,65535,158,113);
..newValue10 = map((data.particles_10um),0,65535,158,113);
..newValue25 = map((data.particles_25um),0,65535,158,113);
..newValue100 = map((data.particles_100um),0,65535,158,113);
```

```
...x++;

...tft.drawPixel(x,newValue03,ST7735_CYAN);
...tft.drawPixel(x,newValue05,ST7735_YELLOW);
...tft.drawPixel(x,newValue10,ST7735_RED);
...tft.drawPixel(x,newValue25,ST7735_WHITE);
...tft.drawPixel(x,newValue100,ST7735_BLUE);

...if (x>123) {
...x=2;
...tft.fillRect(2,113,126,45,ST7735_BLACK);

...}
...if (number_leds > 16.) // Αν ανάψουν περισσότερα από 16 led
{
...digitalWrite(5, HIGH); // Άνοιξε την ψηφιακή θύρα 5 οπότε ξεκινάει και ο ανεμιστήρας
...}
...if (number_leds < 16 ) Αν ανάψουν λιγότερα από 16 led
{
...digitalWrite(5, LOW); // Κλείσε την ψηφιακή θύρα 5 οπότε σταματάει και ο ανεμιστήρας
...}
}
```

Συμπεράσματα

Όπως προαναφέραμε στην εργασία μας, ο μικροελεγκτής Arduino είναι μια πλακέτα μικρού μεγέθους ,που μπορεί να μην φαίνεται κάτι το ιδιαίτερο αλλά με τα κατάλληλα πρόσθετα εξαρτήματα (Shields), κάποιες βασικές γνώσεις και τη θέληση μπορεί ο καθένας να δημιουργήσει ένα ολοκληρωμένο κύκλωμα. Έτσι ,το κύκλωμα θα είναι προσαρμοσμένο στις δικές του προσωπικές απαιτήσεις και ρυθμίσεις οι οποίες θα αλλάζουν ανάλογα ,αν θέλει να προσθέσει ή να αφαιρέσει κάτι.Ακόμη, από μόνος του ο χρήστης κάνει τις αλλαγές που επιθυμεί, λαμβάνοντας υπόψη του πληροφορίες από κάποιον παραπλήσιο κώδικα που είναι ελεύθερα στην κοινότητα του Arduino .

Κόστος Εργασίας

1	Arduino Uno	21.50 €
2	PMS7003	32.00 €
3	DS1307	2.25 €
4	BreadBord	5.40 €
5	TFT 1.8'	12.55 €
6	Ανεμιστήρας	5.60 €
7	Ηλεκτρολογικό Κουτί	10.00 €
8	Διαφορα Καλώδια	6.00 €

9	BC547 NPN	1.50 €
10	HW-842 5v Relay	5.20 €
Σύνολικο Κόστος		102 €

Πίνακας 5 Κόστους Εργασίας

Κεφάλαιο Βελτιώσεων

Το σύστημα μπορεί να μην έφτασε τα επίπεδα επιτυχίας άλλων κατασκευών αλλά κατάφερε να πετύχει τον αρχικό του στόχο. Το συνολικό κόστος της κατασκευής υπολογίζεται στα 102€

Όλες οι ηλεκτρονικές μονάδες που χρησιμοποιήθηκαν ωστόσο επιδέχονται αλλαγή. Για παράδειγμα μπορεί η πλακέτα Arduino Uno να αντικατασταθεί με μια Arduino Mega που έχει μεγαλύτερη μνήμη και υποδοχές ώστε να συνδεθούν περισσότεροι αισθητήρες. Επιπλέον μπορούν να μπουν αισθητήρες θερμοκρασίας και υγρασίας για να έχουμε μια ευρύτερη εικόνα για να συγκρίνουμε πόσο αλλάζουν οι μετρήσεις όταν αλλάζει η πυκνότητα του αέρα πως αλλάζουν η υγρασία και η θερμοκρασία.

Βιβλιογραφία

- [1] http://www.multimedialab.be/doc/erg/2017-2018/Arduino/Arduino_Development_Cookbook_2015.pdf
- [2] <https://www.seeedstudio.com/blog/2019/09/25/best-20-arduino-shields-you-should-try/>
- [3] <https://www.arduino.cc/>
- [4] <http://www.modk.it>
- [5] <http://blog.miniblog.org/>
- [6] <https://www.electronicshub.org/arduino-based-digital-thermometer/>
- [7] <http://el.wikipedia.org/wiki>
- [8] <http://creativec0d1ng.blogspot.gr/2012/09/introduction-to-arduino-part-1.html>
- [9] <http://arduino.cc/en/Main/Products?from=Main.Hardware>
- [10] <https://aqicn.org/air/sensor/spec/pms7003-english-v2.5.pdf>
- [11] <https://web.archive.org/web/20161223050831/http://ebooks.edu.gr/modules/ebook/show.php/DSGL-C107/144/1031,3723/>
- [12] <http://www.redusers.com/noticias/miniblog-el-entorno-argentino-para-programar-robots-que-es-usado-en-escuelas-de-todo-el-mundo/>
- [13] <https://aqicn.org/sensor/pms5003-7003/>
- [14] <https://www.theengineeringprojects.com/2019/04/introduction-to-ds1307.html>
- [15] https://create.arduino.cc/projecthub/sainisagar7294/rgb-audi-ring-smt-service-from-jlpcb-d0df61?ref=part&ref_id=11332&offset=15
- [16] <https://cdn-shop.adafruit.com/datasheets/WS2812.pdf>
- [17] <https://arduinogetstarted.com/tutorials/arduino-relay>
- [18] <https://www.circuitbasics.com/setting-up-a-5v-relay-on-the-arduino/>

[19] <https://roboticafacil.es/datasheets/ky-019.pdf>

[20] <https://components101.com/transistors/bc547-transistor-pinout-datasheet>