INTERNATIONAL HELLENIC UNIVERSITY

DEPARTMENT OF INDUSTRIAL ENGINEERING AND MANAGEMENT

Σύστημα Καταγραφής Θερμοκρασίας Με Τον Nrf52840-DK
Temperature Data Logger With Nrf52840-DK

Φώτιος Καργάκης

Επιβλέπων καθηγητής: Μιχάλης Κιζήρογλου

Θεσσαλονίκη 2021

ABSTRACT

The contents of the literature refer to temperature detection using TMP36 analog sensors and the Nordic controller NRF52840-DK. The purpose of this analysis is to capture temperatures, convert and display the measurements in digital form, for a specific period of time, on the Computer (PC). To achieve the goal, it is necessary to combine two basic pieces. One part refers to the components that will be used and how they are connected to each other, while developing all the features and capabilities of the board and the sensors. The second part includes the programming of the controller in a special environment, which is discussed at the same time about its functions and capabilities. This is where the code is developed, explained and temperatures and volts are displayed. In the process, the calibration is achieved for a wide range of temperatures, while in order to avoid possible errors, a special thermometer is used that measures with absolute accuracy, so that the values of the thermometer are the same as those of the sensors. Next, the procedure for presenting the developer's experiment to other users on github is listed. All the above are analyzed in the text with enough detail and awareness, as they are the necessary steps for the implementation of the experiment. Thus, the researcher will be able to implement his work, while at the same time he will be able to be sure of his results.

ΠΕΡΙΛΗΨΗ

Το περιεχόμενο της βιβλιογραφίας αναφέρεται στην ανίχνευση θερμοκρασιών με την χρήση αναλογικών αισθητήρων TMP36 και του ελεγκτή Nordic NRF52840-DK. Σκοπός της ανάλυσης αυτής είναι η σύλληψη θερμοκρασιών , η μετατροπή και η εμφάνιση των μετρήσεων σε ψηφιακή μορφή , για συγκεκριμένο χρονικό διάστημα , στον Ηλεκτρονικό Υπολογιστή (Η/Υ). Για την επίτευξη του στόχου , απαραίτητος είναι ο συνδυασμός δύο βασικών κομματιών. Το ένα κομμάτι αναφέρεται στα εξαρτήματα που θα χρησιμοποιηθούν και πως συνδέονται αυτά μεταξύ τους , ενώ αναπτύσσονται όλα τα χαρακτηριστικά και οι δυνατότητες της πλακέτας και των αισθητήρων. Το δεύτερο κομμάτι περιλαμβάνει τον προγραμματισμό του ελεγκτή σε ειδικό περιβάλλον , για το οποίο γίνεται ταυτόχρονα λόγος για τις λειτουργίες και της δυνατότητες που διαθέτει. Εκεί πραγματοποιείται η ανάπτυξη του κώδικα , η επεξήγηση του και η εμφάνιση των θερμοκρασιών και των τάσεων. Στη πορεία επιτυγχάνεται η βαθμονόμηση για ένα μεγάλο εύρος θερμοκρασιών , ενώ για την αποφυγή πιθανών σφαλμάτων χρησιμοποιείται ένα ειδικό θερμόμετρο που μετράει με απόλυτη ακρίβεια , έτσι ώστε οι τιμές του θερμομέτρου μ' αυτές των αισθητήρων να είναι οι ίδιες. Έπειτα , αναγράφεται η διαδικασία για τη παρουσίαση του πειράματος του προγραμματιστή ως προς τους υπολοίπους χρήστες , στο github. Όλα τα παραπάνω , αναλύονται στο κείμενο με αρκετή λεπτομέρεια και συνείδηση , καθώς αποτελούν τα απαραίτητα βήματα για την υλοποίηση του πειράματος. Έτσι λοιπόν , ο μελετητής θα είναι σε θέση να υλοποιήσει το έργο του , ενώ να παράλληλα θα μπορέσει να σιγουρευτεί για τα αποτελέσματά του.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες σε όλους όσους συνέβαλλαν για την εκπόνηση της διπλωματικής μου εργασίας.

Αρχικά , ευχαριστώ θερμά τον επιβλέπων καθηγητή μου , κύριο Μιχάλη Κιζήρογλου , για την επιστημονική καθοδήγηση του , τη διαρκή συμπαράστασή του και το αμείωτο ενδιαφέρον που έδειξε , για το θέμα της εργασίας που με ανέθεσε.

Επίσης , ευχαριστώ την εταιρία ηλεκτρονικών εξαρτημάτων , GRobotronics , για την άμεση ανταπόκριση και αποστολή των υλικών , απαραιτήτων για την υλοποίηση του πειραματικού μέρους , του έργου μου.

Επιπλέον , προσωπικές ευχαριστίες οφείλω να δώσω στους γιατρούς , του τμήματος της Νευρολογικής μονάδας του Στρατιωτικού Νοσοκομείου 424 Θεσσαλονίκης , που με την επιμονή τους , κατάφεραν για ένα ολόκληρο μήνα , να με απαλλάξουν από το ξαφνικό πρόβλημα υγείας που με προέκυψε , ενώ δεν επέτρεψαν ούτε στιγμή , να κατρακυλήσει η ψυχολογία μου.

Τέλος , θα ήθελα να εκφράσω την ευγνωμοσύνη μου στην οικογένεια μου , για τη πλήρη συμπαράσταση που με πρόσφεραν όλα αυτά τα χρόνια κατά τη διάρκεια της φοίτησης μου.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΠΕΞΗΓΗΣΗ ΑΚΡΩΝΥΜΙΩΝ

- μA : μικροΑμπέρ του Διεθνούς Συστήματος Μονάδων Μέτρησης (SI)

- Η/Υ : Ηλεκτρονικός Υπολογιστής

- ADC: Analog to Digital Converter (Μετατροπή Αναλογικού Σε Ψηφιακό)

- AREF: Analog Reference (Αναλογική Αναφορά)

- BIT: Binary Digit (Δυαδικό Ψηφίο)

- BLE: Bluetooth Low Energy (Bluetooth Χαμηλής Ενέργειας)

- BPS: Bits Per Second (Bits Ανά Δευτερόλεπτο)

- COMP: Comparator (Συγκριτής)

- CPU: Central Processing Unit (Κεντρική Μονάδα Επεξεργασίας)

- DC: Direct Current (Συνεχές Ρεύμα)

- FTP: File Transfer Protocol (Πρωτόκολλο Μεταφοράς Αρχείων)

- GND: Ground (Γείωση)

- GHz : Giga Hertz του Διεθνούς Συστήματος Μονάδων Μέτρησης (SI)

- I/O: Input / Output (Είσοδος / Έξοδος)

- J-LINK : Αλλιώς το συναντάμε και ως J-TAG και πρόκειται για το καλώδιο που συνδέει, τον ελεγκτή NRF με τον Η/Υ. Η μία έξοδος καταλήγει στην πλακέτα μας με βύσμα τύπου J2, ενώ η άλλη σε κάποια από τις θύρες USB του Η/Υ.

- KB : Kilo Byte του Διεθνούς Συστήματος Μονάδων Μέτρησης (SI)

- LED: Light Emitting Diode (Δίοδος Εκπομπής Φωτός)

- MB : Mega Byte του Διεθνούς Συστήματος Μονάδων Μέτρησης (SI)

- MCU: Microcontroller Unit (Μονάδα Μικροελεγκτή)

- MHz : Mega Hertz του Διεθνούς Συστήματος Μονάδων Μέτρησης (SI)

-MIT: Massachusetts Institute of Technology (Ινστιτούτο Τεχνολογίας της Μασαχουσέτης)

- PC: Computer (Ηλεκτρονικός Υπολογιστής)

- RAM: Random Access Memory (Μνήμη Τυχαίας Προσπέλασης)

- RF: Radio Frequency (Ραδιοφωνική Συχνότητα)

- SW: Switch (Διακόπτης)

- USB: Universal Serial Bus (Γενικό Σειριακό Λεωφορείο)

- V : Voltage ή Volt (Τάση) του Διεθνούς Συστήματος Μονάδων Μέτρησης (SI)

- VDIV: Voltage Divider (Διαιρέτης Τάσης)
- VS CODE: Visual Studio Code (Πρόγραμμα για Ανάπτυξη Κώδικα)
- Wi-fi : Ασύρματο Δίκτυο Τεχνολογίας

ΕΙΣΑΓΩΓΗ

Το σύγγραμμα δίνει την ευκαιρία στον αναγνώστη να καταφέρει να μελετήσει και να υλοποιήσει το πείραμα που περιγράφεται. Σκοπός του πειράματος είναι ο μελετητής μέσα από το έργο να κατορθώσει να το κατασκευάσει , καθώς και να κατανοήσει πως θα λάβει μετρήσεις σε ψηφιακή μορφή χρησιμοποιώντας κάποια αναλογικά υλικά για κάποιο χρονικό διάστημα που αυτός θα ορίσει. Αυτά τα υλικά αποτελούν οι αισθητήρες θερμοκρασίας TMP36 , οι οποίοι συνδέονται σε τέσσερα αναλογικά κανάλια του ελεγκτή Nordic NRF52840-DK. Στην ουσία θα δούμε τον τρόπο με τον οποίο συνδέονται οι αισθητήρες με την πλακέτα , τον προγραμματισμό που απαιτείται έτσι ώστε να πάρουμε ξεκάθαρα και ορατά αποτελέσματα , την αποφυγή και εκμηδένιση πιθανών αποκλίσεων των αποτελεσμάτων , καθώς και την κοινοποίηση του έργου ως προς άλλους χρήστες που ενδιαφέρονται να υλοποιήσουν το ίδιο ή παρόμοιο πείραμα.

Αναλυτικότερα , στο πρώτο κεφάλαιο περιγράφονται τα χαρακτηριστικά της πλακέτας Nordic NRF52840-DK , καθώς δίνονται πληροφορίες για την εταιρία που τη παρασκευάζει. Ακόμα , μεγάλη έμφαση δίνεται στα αναλογικά κανάλια του ελεγκτή , αφού διαθέτει έξι κανάλια από τα οποία θα επιλεχθούν τα τέσσερα και ταυτόχρονα αναγράφονται οι λόγοι της προτίμησής τους. Επιπλέον , αναπτύσσονται δεδομένα για τα bits καθώς και για τη τάση αναφοράς του NRF. Το δεύτερο κεφάλαιο μιλάει για τον αισθητήρα TMP36 , δίνοντας βάση στα χαρακτηριστικά του , ενώ παράλληλα περιγράφονται όλα τα βήματα που απαιτούνται για την σύνδεσή και των τεσσάρων αισθητήρων , με τον ελεγκτή που μελετάμε. Το τρίτο κεφάλαιο που αποτελεί και το μεγαλύτερο της βιβλιογραφίας , είναι αρκετά σημαντικό ώστε να αποφευχθούν πιθανές δυσκολίες πριν μπούμε στη διαδικασία του προγραμματισμού. Σ’ αυτό λοιπόν αναγράφεται το πρόγραμμα που θα χρησιμοποιηθεί για την υλοποίηση του κώδικα του έργου. Το πρόγραμμα αυτό είναι το Visual Studio Code (VS Code) , το οποίο περιέχει και άλλα προγράμματα , εκ’ των οποίων το ένα από αυτά ονομαζόμενο Platform IO αποτελεί το βασικό πυλώνα για το πείραμα. Στη πορεία , επιτυγχάνεται η ανάλυση του μενού του VS Code , αλλά και η εγκατάσταση του Platform IO μαζί με την επεξήγηση του περιβάλλοντός του. Έπειτα , λίγο πριν μπούμε στο προγραμματιστικό κομμάτι , αξίζει να σιγουρευτούμε ότι η πλακέτα που διαθέτουμε έχει αναγνωριστεί από τη συσκευή με την οποία δουλεύουμε. Επομένως, αναπτύσσονται όλα τα βήματα , για τη εγκατάσταση του NRF με τον ηλεκτρονικό υπολογιστή (Η/Υ). Στο τέταρτο κεφάλαιο πραγματοποιείται η γραφή και η επεξήγηση του κώδικα. Το κεφάλαιο όμως δεν ξεκινάει αμέσως με το

προγραμματισμό. Προαναφέρονται ενότητες , οι οποίες περιέχουν αρκετά σημαντικές πληροφορίες και έννοιες , χρήσιμες για το κώδικα. Αυτές αφορούν το ρυθμό μετάδοσης της ταχύτητας (baud rate) καθώς και κάποιες βασικές καρτέλες , στις οποίες πραγματοποιούνται απαραίτητες ρυθμίσεις σύμφωνα με τα χαρακτηριστικά που διαθέτει η πλακέτα εργασίας. Αμέσως μετά , σειρά παίρνει η εκτέλεση του κώδικα , αλλά και η παρουσίαση των αποτελεσμάτων που θα πρέπει να φαίνονται στις οθόνες του χρήστη. Στο πέμπτο κεφάλαιο , εξακριβώνονται τα αποτελέσματα με την βοήθεια ενός ειδικού θερμομέτρου της βρετανικής εταιρίας Lascar , καθώς για συγκεκριμένες θερμοκρασίες να λαμβάνονται οι σωστές αντίστοιχες τάσεις , σύμφωνα με τα χαρακτηριστικά του TMP36. Στην ουσία δημιουργείται ένα διάγραμμα μεταξύ θερμοκρασιών και χρονικής διάρκειας με αποτέλεσμα να επιτυγχάνεται η βαθμονόμηση , για ένα μεγάλο εύρος τιμών. Το έκτος κεφάλαιο αποτελεί και το τελευταίο της συγγραφής και αναφέρεται στη διεπαφή του χρήστη με άλλους προγραμματιστές. Στην ουσία , παρουσιάζονται όλες οι απαραίτητες διαδικασίες , χρήσιμες για την κοινοποίηση του έργου σε μια ιστοσελίδα, ονομαζόμενη Github , ενώ παράλληλα αναπτύσσεται η έννοιά του. Στη συνέχεια , δίνεται το link από το οποίο μπορεί κανείς να αντλήσει πληροφορίες για το έργο και να βρει έτοιμο το κώδικα και τη βαθμονόμηση που πραγματοποιήθηκε. Τέλος , ακολουθούν δύο παραρτήματα , που αφορούν τον επίλογο και τις πηγές του συγγράμματος.

ΚΕΦΑΛΑΙΟ 1: ΠΛΗΡΟΦΟΡΙΕΣ ΓΙΑ ΤΟΝ NRF52840-DK ΚΑΙ ΑΝΑΛΥΣΗ ΤΩΝ ΑΝΑΛΟΓΙΚΩΝ ΚΑΝΑΛΙΩΝ

1.1 ΠΡΩΤΗ ΓΝΩΡΙΜΙΑ ΜΕ ΤΟΝ NORDIC NRF52840-DK

Η NORDIC NRF52840-DK (Σχήμα 1.1) είναι μια μικροπλακέτα προγραμματισμού [1], η οποία μπορεί να χρησιμοποιηθεί για τις απαιτήσεις της σύγχρονης τεχνολογίας. Η γλώσσα για το προγραμματισμό που επιλέγουμε κατά τη διάρκεια εκτέλεσης του έργου , είναι η C++ , μέσω του λογισμικού Platform IO του Visual Studio Code , συμβατό με Windows , Linux και OSX. Διαθέτει Bluetooth Low Energy (BLE) , γεγονός που το καθιστά να παρέχει μεγάλη ασύρματη εμβέλεια με τη βοήθεια της συσκευής Zigbee , καθώς και να καταναλώνει ελάχιστη ενέργεια. Η τάση με την οποία λειτουργεί κυμαίνεται από 1.7 έως 5.5 Volts (V) και τροφοδοτείται είτε με τη χρήση επαναφορτιζόμενων μπαταριών ή μέσω ενός καλωδίου , που στη μία έξοδο περιέχει βύσμα τύπου J2 , ενώ η άλλη έξοδος του μπορεί να συνδεθεί με κάποια από τις USB θύρες του Η/Υ. Επίσης , αξίζει να σημειωθεί , ότι περιέχει 48 ψηφιακές εισόδους/εξόδους (I/O) , αλλά παράλληλα οι 6 αναλογικοί είσοδοι (ADC) που διαθέτει είναι αυτοί , οι οποίοι θα μας απασχολήσουν για την υλοποίηση του πειράματος. Ακόμα , μεγάλες μπορούν να θεωρηθούν οι μνήμες του φλας και της τυχαίας προσπέλασης (RAM) , με 1 Mega Byte (MB) και 256 Kilo Bytes (KB) αντιστοίχως , ενώ η ταχύτητα του ρολογιού του , είναι στα 64 Mega Hertz (MHz). Τέλος, περιλαμβάνει κρυπτογραφικό υλικό ARM Cryptocell 310. Όλα τα παραπάνω, αναλύονται στις επόμενες ενότητες και κεφάλαια.

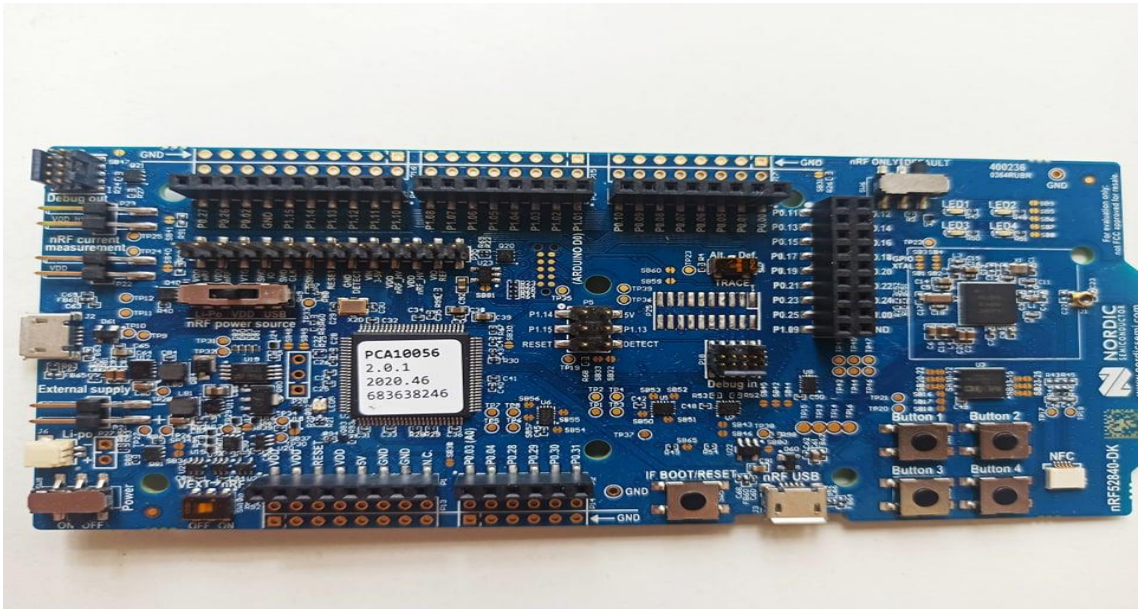
1.1.1 Η ΕΤΑΙΡΕΙΑ NORDIC

Η εταιρεία NORDIC SEMICONDUCTOR , εδρεύει στο Trondheim , σε μια περιοχή της σκανδιναβικής Νορβηγίας [2]. Πρόκειται για μια εταιρία η οποία παρασκευάζει ημιαγωγούς. Επίσης, είναι ειδική για την κατασκευή των τσιπ SoC , για τα οποία αποδίδει με ασύρματο τρόπο χαμηλή ισχύ. Ακόμα , ασχολείται με συσκευές οι οποίες συνδέονται με τη ραδιοφωνική ζώνη RF 2,4 GHz που χρησιμεύουν σε βιομηχανικούς κυρίως σκοπούς. Ακόμα , η εταιρεία ενσωμάτωσε στις συσκευές της BLE με τις σειρές NRF52 να αποτελούν τις πρώτες , οι οποίες αποδείχθηκαν από τις πιο πετυχημένες σειρές της εταιρείας. Αξίζει

όμως παρακάτω να μελετήσουμε το ξεκίνημα της και να δούμε την σημερινή απασχόληση της.

1.1.2 Η ΙΣΤΟΡΙΑ ΤΗΣ ΕΤΑΙΡΕΙΑΣ

Η εταιρεία Nordic Semiconductor, ιδρύθηκε το έτος 1983 [3], όπου το αρχικό της όνομα ήταν NORDIC NVLSI. Στην αρχή η εταιρεία ασχολήθηκε με τον σχεδιασμό ολοκληρωμένων κυκλωμάτων. Το όνομα της άλλαξε στις 14 Μαϊου 2004 της οποίας διατηρείται μέχρι και σήμερα , καθώς εκείνη την χρονιά παρουσίασε στο χώρο της αγοράς τη πρώτη της ασύρματη συσκευή στα 2,4GHz. Τα τελευταία χρόνια η εταιρεία ασχολείται αποκλειστικά με την παράδοση των ημιαγωγών που κατασκευάζει ανεβάζοντας τη πελατεία της , σε αρκετά υψηλά επίπεδα , σύμφωνα με τον διευθύνων σύμβουλό της , το κύριο Svenn Tore Lasen.



Σχήμα 1.1 : Ο μικροελεγκτής NORDIC NRF52840-DK.

1.2 ΤΑ ΑΝΑΛΟΓΙΚΑ ΚΑΝΑΛΙΑ ΤΟΥ NRF

Όπως αναφέραμε και παραπάνω , υπάρχουν έξι αναλογικά κανάλια εισόδου (ADC) , στον μικροελεγκτή. Εκεί , θα συνδεθούν οι τέσσερεις αναλογικοί αισθητήρες θερμοκρασίας TMP36 , όπου περισσότερες πληροφορίες γι‘ αυτούς αναγράφονται στο επόμενο κεφάλαιο. Συγκεκριμένα , οι έξι αυτοί αναλογικοί είσοδοι είναι οι εξής :

1)P0.03

2)P0.04

3)P0.28

4)P0.29

5)P0.30

6)P0.31

Ουσιαστικά , θα γίνει η επιλογή τεσσάρων καναλιών (ADC) , των οποίων η δουλειά τους είναι να καθορίσουν το σημείο , στο οποίο θα συλλεχθούν τα δεδομένα από τους αισθητήρες θερμοκρασίας και θα μεταφερθούν στο Cloud του Η/Υ. Έστω , ότι επιλέγουμε τα κανάλια P0.03 , P0.04 , P0.28 και P0.30. Για τη συλλογή αυτή , θα πρέπει να γνωρίζουμε σε πόσα δυαδικά ψηφία (bits) , λειτουργεί ο NRF. Έτσι , η υποενότητα 1.2.2 του κεφαλαίου αναφέρεται στα bits των (ADC).

1.2.1 P0.29 (VDIV) ΚΑΙ P0.31 (AREF)

Υπάρχει λόγος , για τον οποίο δεν θα γίνει χρήση των ADC P0.29 (VDIV) και P0.31 (AREF). Στη πρώτη περίπτωση , αν ο ADC χρησιμοποιηθεί ως είσοδος η σύνδεση πρέπει να γίνει μέσω ενός ειδικού πείρου , ο οποίος μεν θα προσφέρει στο κύκλωμα μεγαλύτερη προστασία ως προς την μπαταρία της συσκευής , αλλά από την άλλη θα χαθεί η δυνατότητα ανάγνωσης της μπαταρίας στο πρόγραμμα. Γι' αυτό το λόγο , ο A6/P0.29 χρησιμοποιείται ως έξοδος για την ανάγνωση της μπαταρίας , αφού πρώτα έχει γίνει αλλαγή του πείρου σε αναλογική είσοδο.

Στην δεύτερη περίπτωση , έχουμε μια αναλογική έξοδο η οποία θα μπορεί να συνδεθεί με τον περιφερειακό συγκριτή COMP. Θα ήταν εφικτό βέβαια να το συνδέσουμε με έναν εξωτερικό AREF και να το προγραμματίσουμε . Ωστόσο , θα πρέπει να δώσουμε ιδιαίτερη προσοχή στην τροφοδοσία του καθώς η τάση του θα πρέπει να είναι μικρότερη ή τουλάχιστον ίση με το VDD.

1.2.2 ΤΑ BITS ΚΑΙ ΟΙ ΤΙΜΕΣ ΤΩΝ ADC

Σε μια μικροπλακέτα σημαντικό ρόλο παίζει να γνωρίζουμε την ανάλυση των bits των καναλιών που θα χρησιμοποιηθούν [4]. Κάθε αναλογικό κανάλι (ADC) στον NRF52840-DK , διαθέτει ανάλυση 12bit και επομένως , ο μικροελεγκτής μπορεί να στείλει 2^12= 4.096 δηλαδή (0…4095) , διαφορετικές τιμές στον Η/Υ. Επειδή , το εύρος της τάσης στον NRF είναι 3,0 V , αυτό σημαίνει ότι για κάθε ψηφίο θα επιστέφονται 3,0/4.096= 0,0007324 V. Άρα , κάθε ADC περιλαμβάνει εσωτερική τάση 3V/6= 0,5 V και χάρη σ' αυτή ,

μπορούν να ρυθμιστούν και να δημιουργήσουν δεδομένα των 12bit , έτσι ώστε τα αποτελέσματα να είναι ακριβή και ορατά. Στα επόμενα κεφάλαια , αναπτύσσεται ο τρόπος με τον οποίο δηλώνουμε τον αριθμό των bits , μέσα στο πρόγραμμα που δημιουργείται ο κώδικάς.

1.2.3 ΤΑΣΗ ΑΝΑΦΟΡΑΣ (VOLTAGE REFERENCE)

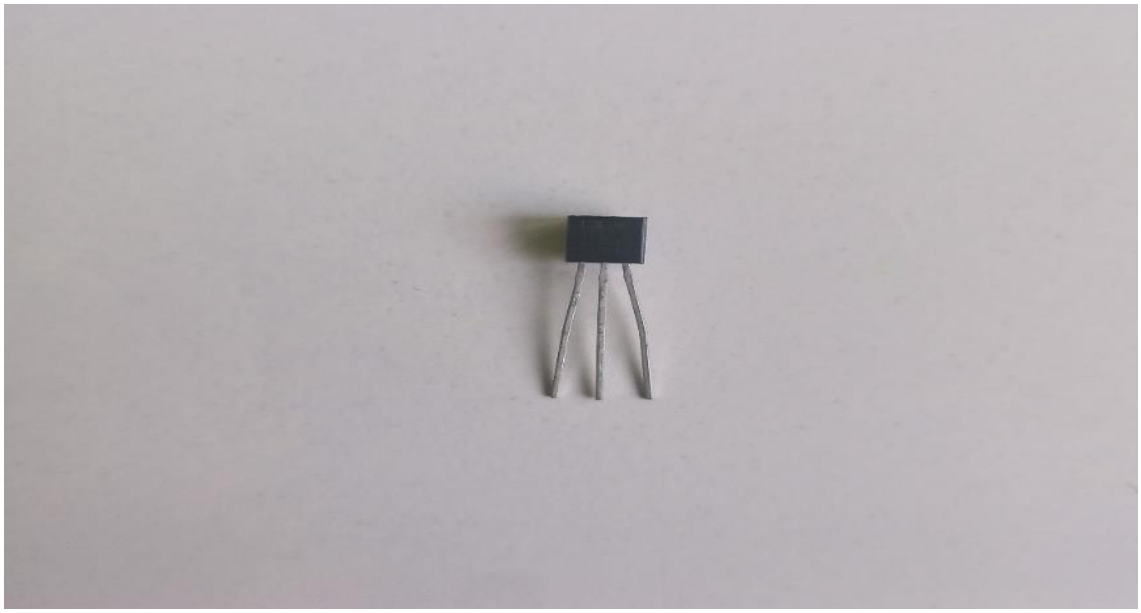
Για την ρύθμιση των καναλιών , ώστε να μπορούν να αναγνωστούν σωστά και με απόλυτη ακρίβεια οι δειγματοληψίες που θα πάρουμε , απαραίτητη προϋπόθεση είναι να γνωρίζουμε την τάση αναφοράς του μικροελεγκτή που πρόκειται να χρησιμοποιήσουμε. Στη προκειμένη περίπτωση , η τάση αναφοράς βρίσκεται στα 3.0V και η χρήση της βοηθά στην μετατροπή του αναλογικού σήματος σε ψηφιακή μορφή. Σε αντίθεση με άλλους κοινούς μικροελεγκτές , όπου ο λειτουργούν στα 3.3 V ή ακόμα και στα 3.6 V, ο NRF είναι μία μικροπλακέτα που κατασκευάστηκε με σκοπό να καταναλώνει χαμηλή ισχύ. Γι' αυτό το λόγο , η τάση του πρέπει να οριστεί στα 3.0 V , κατά τη διάρκεια της ανάπτυξης του κώδικα. Είναι η τάση η οποία χρησιμοποιείται για την μετατροπή της αναλογικής τιμής που διαβάζει ο αισθητήρας TMP36 , σε ψηφιακή μορφή. Αυτή η επεξήγηση αποτελεί απαραίτητο κομμάτι κατά τη δημιουργία του κώδικα , γιατί μια λάθος εκτίμηση θα οδηγήσει σε εσφαλμένες τιμές.

ΚΕΦΑΛΑΙΟ 2: Ο ΑΝΑΛΟΓΙΚΟΣ ΑΙΣΘΗΤΗΡΑΣ TMP36

2.1 ΠΛΗΡΟΦΟΡΙΕΣ ΚΑΙ ΚΟΣΤΟΣ

Ο αισθητήρας TMP36 είναι ένας αισθητήρας θερμοκρασίας [5] , ο οποίος έχει αναλογική έξοδο. Έχει την δυνατότητα να μετράει θερμοκρασίες από -40 έως και +150 βαθμούς Κελσίου. Η τροφοδοσία του κυμαίνεται από 2.7V έως και 5.5V συνεχούς ρεύματος (DC). Επίσης , θερμαίνεται πολύ δύσκολα , λιγότερο από 0.1 βαθμούς Κελσίου , καθώς το ρεύμα που το διαπερνάει είναι αρκετά πιο μικρό από 50μA. Ακόμα, για κάθε αύξηση της μονάδας θερμοκρασίας , η τιμή της τάσης αυξάνεται κατά 0.01 V. Άλλο ένα από τα ιδιαίτερα του χαρακτηριστικά είναι ότι παρέχει έξοδο 750mV στους 25 βαθμούς Κελσίου.

Ότι αφορά το κόστος για την αγορά του , η τιμή του μπορεί να θεωρηθεί χαμηλή καθώς το υλικό κατασκευής αποτελείται από μόλυβδο. Έτσι , για το πείραμα μας θα χρειαστούμε τέσσερεις αισθητήρες των οποίων το ποσό τους ανέρχεται σε λιγότερο από αυτό των 10 ευρώ.



Σχήμα 2.1 : Ο αισθητήρας TMP36

2.2 ΟΙ ΑΚΡΟΔΕΚΤΕΣ ΤΟΥ TMP36

Ο TMP36 αποτελείται από τρεις ακροδέκτες , όπως φαίνεται στο παραπάνω σχήμα (Σχήμα 2.1). Ο πρώτος ακροδέκτης , από τα αριστερά συνδέεται με την τροφοδοσία του ράστερ , ενώ ο τρίτος συνδέεται με την γείωση (GND) του. Από την άλλη , ο μεσαίος ακροδέκτης , συνδέεται με κάποιο από τα αναλογικά κανάλια του μικροελεγκτή που διαθέτουμε. Ο τρόπος με τον οποίο επιτυγχάνεται η σύνδεση , αναγράφεται αμέσως μετά. Ωστόσο πρέπει

να δοθεί ιδιαίτερη προσοχή στο τρόπο τοποθέτησης του πάνω στο ράστερ ώστε οι ακροδέκτες του , μέσω μικρών καλωδίων, να καταλήξουν στα σημεία που αναφέρθηκαν πιο πάνω. Μια λάθος εκτίμηση θα οδηγήσει σε πιθανή καταστροφή του αισθητήρα.

2.2.1 ΣΥΝΔΕΣΗ ΤΩΝ ΑΙΣΘΗΤΗΡΩΝ TMP36 ΜΕ ΤΟΝ NRF

Για την συναρμολόγηση του πειράματος μας , θα χρειαστούμε τα παρακάτω εξής εξαρτήματα:

1) Ένα breadboard (ράστερ) , του οποίου η προμήθεια γίνεται από κάποιο κατάστημα που πουλάει ηλεκτρονικά είδη.

2) Τέσσερεις αναλογικούς αισθητήρες TMP36 , των οποίων η παραλαβή μπορεί να πραγματοποιηθεί διαδικτυακά από την εταιρία GRobotronics.

3) Ένα μικροελεγκτή NORDIC NRF52840-DK , της εταιρείας Digi Key. Εδώ πρέπει να υπάρξει ιδιαίτερη προσοχή , γιατί μπορεί να προκύψει επιπρόσθετη χρέωση από το τελωνείο.

4) Αρκετά μικρά καλώδια σύνδεσης τύπου αρσενικό-αρσενικό , διαφόρων χρωμάτων.

5) Ένα καλώδιο ονόματι J-link το οποίο διαθέτει στην μία άκρη βύσμα τύπου J2 και στην άλλη τύπου USB ώστε να συνδέει τον μικροελεγκτή με τον Η/Υ.

6) Αλλά και ένα ειδικό θερμόμετρο της εταιρίας Lascar , με το οποίο μετά την εκτέλεση του πειράματος , αποτελεί χρήσιμο εργαλείο για την εξακρίβωση των αποτελεσμάτων. Γι' αυτό , παραπάνω πληροφορίες δίνονται στο πέμπτο κεφάλαιο.

Αφού λοιπόν , έχουμε στην διάθεση μας όλα τα παραπάνω υλικά είμαστε έτοιμοι να ξεκινήσουμε τη σύνδεση ακολουθώντας τα παρακάτω βήματα:

1) Τοποθετούμε σε διαφορετικά σημεία του ράστερ τους τέσσερεις αισθητήρες TMP36.

2) Στη συνέχεια , συνδέουμε τα 5V του NRF στο «+» του ράστερ και το GND στο «-» του board , με την βοήθεια των καλωδίων που διαθέτουμε.

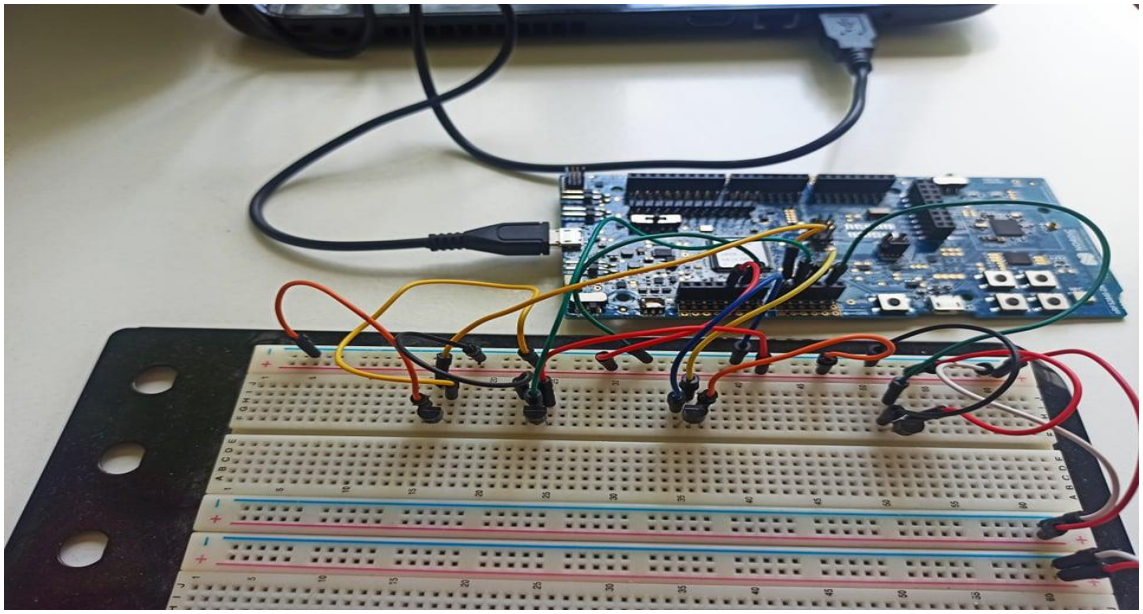
3) Έπειτα , συνδέουμε τους αισθητήρες στην τροφοδοσία και στην γείωση με τον τρόπο που αναφέραμε παραπάνω , στην ενότητα 2.2.

4) Ενώνουμε τον μεσαίο ακροδέκτη του πρώτου αισθητήρα με το αναλογικό κανάλι P0.03 του μικροελεγκτή , του δεύτερου με το P0.04 ,του τρίτου με το P0.28 και του τέταρτου με το κανάλι P0.30.

5) Τέλος , το μόνο που απομένει είναι να συνδέσουμε τον NRF με τον υπολογιστή μέσω του καλωδίου J-link. Ωστόσο , θα χρειαστεί να πραγματοποιηθούν κάποια βήματα που αναφέρονται πιο κάτω , για την πλήρη εγκατάσταση του τόσο στον Η/Υ αλλά και στο πρόγραμμα εργασίας.

Άρα , η συνδεσμολογία μας θα πρέπει να είναι όμοια με αυτή που απεικονίζεται στο Σχήμα2.2. Επομένως , σημαντική έμφαση πρέπει να αποδοθεί στη σύνδεση μεταξύ των εξαρτημάτων , γιατί μία λάθος πρωτοβουλία θα προκαλέσει το κάψιμο των αισθητήρων ή ακόμα χειρότερα της μικροπλακέτας.

Πλέον , βρισκόμαστε σε ετοιμότητα για να περάσουμε στο προγραμματιστικό κομμάτι. Πριν από αυτό όμως , θα δούμε στο επόμενο κεφάλαιο το τρόπο εγκατάστασης του προγράμματος VS Code , τα χαρακτηριστικά του , τις βασικές του λειτουργείες , το μενού του καθώς και το Platform IO μαζί την ανάπτυξη της επεξήγησης του περιβάλλοντος του.



Σχήμα 2.2: Η συνδεσμολογία μεταξύ των αισθητήρων TMP36 με τον NRF52840-DK.

ΚΕΦΑΛΑΙΟ 3: ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ VS CODE ΚΑΙ ΕΠΕΞΗΓΗΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ PLATFORMIO

3.1 ΓΕΝΙΚΕΣ ΠΛΗΡΟΦΟΡΙΕΣ ΓΙΑ ΤΟ VISUAL STUDIO CODE

Το Visual Studio Code πρόκειται για ένα σύγχρονο πρόγραμμα [6] , το οποίο δημιουργήθηκε από την Microsoft για να διευκολύνει την ζωή των προγραμματιστών. Μ΄ αυτό , οι προγραμματιστές μπορούν να αναπτύξουν και να επεξηγήσουν το πηγαίο τους κώδικα με διάφορες λειτουργίες. Μπορεί να χρησιμοποιηθεί στα γνωστά λογισμικά των Windows , Linux και macOS.

Χάρη στον επεξεργαστή του έχει σαν αποτέλεσμα να χρησιμοποιείται από πολλούς χρήστες , ενώ διατίθεται δωρεάν στο διαδίκτυο. Το γεγονός αυτό το καθιστά ως το πιο δημοφιλές εργαλείο των προγραμματιστών. Η πρώτη κυκλοφορία έγινε στις 29 Απριλίου 2015 , μετά από το ετήσιο συνέδριο της Microsoft Build που πραγματοποιήθηκε , χωρίς την άδεια MIT. Η άδεια εγκρίθηκε στις 18 Νοεμβρίου της ίδιας χρονιάς και έτσι πλέον το λογισμικό θεωρήθηκε και επίσημα ελεύθερο , δηλαδή ο χρήστης θα μπορούσε πλέον να έχει όλες τις διαθέσιμες λειτουργίες που θα τον βοηθήσουν για ανάπτυξη του πηγαίου του κώδικα , για το οποίο πληροφορίες βρίσκονται στο GitHub. Μερικές από αυτές τις λειτουργείες , θα αναλύσουμε αμέσως παρακάτω , οι οποίες είναι απαραίτητες για την διευκόλυνση και την καλύτερη κατανόηση του κώδικα για το πείραμά μας.

3.1.1 ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ

Για μεγαλύτερη ευκολία της υλοποίησης του προγράμματος μας , μπορούμε να δούμε τις λειτουργίες που παρέχει το Visual Code Studio. Το πρόγραμμα διαθέτει πλήθος δυνατοτήτων, όπου οι πιο βασικές από αυτές, είναι να επισημάνει την σύνταξη του κειμένου, να το ολοκληρώνει και να εντοπίζει τα σφάλματα του κώδικα.

Η λειτουργία που αφορά την επισήμανση της σύνταξης είναι μία στρατηγική για την διευκόλυνση ανάγνωσης αλλά και κατανόησης του κειμένου κατά , την επεξεργασία του. Αυτό επιτυγχάνεται , με την δυνατότητα που έχει ο χρήστης να γράφει το κώδικα με διάφορα χρώματα που αυτός έχει ορίσει , έτσι ώστε να αποφύγει κάποιο πιθανό σφάλμα ή ακόμα και να το εντοπίσει με περισσότερη ευκολία , λόγω της διχρωμίας που χρησιμοποιεί , κατά την σύνταξη του. Επίσης , παρουσιάζει σημαντικό ρόλο για την ορθή

δομή του κειμένου , ειδικά όταν έχουμε να κάνουμε με προγραμματισμό , όπου εκεί το πρόγραμμα , ίσως , να είναι αρκετά μεγάλο.

Η διαδικασία της ολοκλήρωσης του κώδικα , σημάνει μια έξυπνη λειτουργία , η οποία μπορεί το συμπληρώσει με αυτοματοποιημένο τρόπο το κώδικα. Η χρησιμότητα της είναι σημαντική σε μεγάλο βαθμό , αφού βοηθάει στην αποφυγή τυπογραφικών λαθών. Επιπλέον , ο προγραμματιστής κερδίζει σε χρόνο για την υλοποίηση του έργου του , ενώ η δυνατότητα της απομνημόνευσης των μεταβλητών που δήλωσε σε προηγούμενες σειρές του κώδικα , περιορίζουν τη χρήση του πληκτρολογίου.

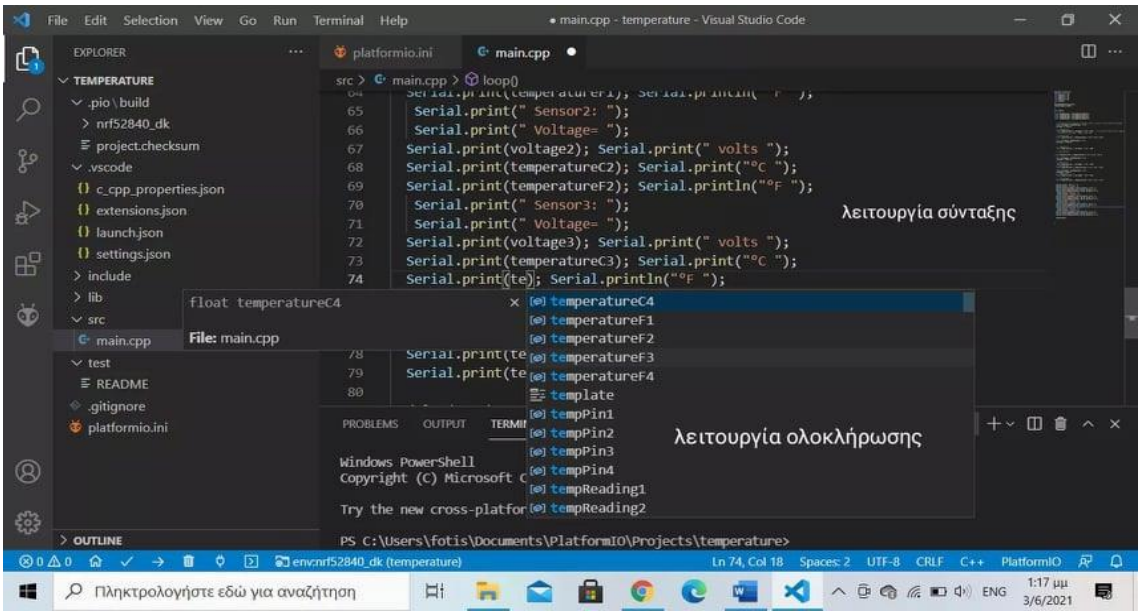
Μία από τις σημαντικότερες λειτουργίες , είναι η διαδικασία εντοπισμού των σφαλμάτων που προκύπτουν. Το πρόγραμμα ψάχνει , εντοπίζει και επισημαίνει τα λάθη στο χρήστη που αποτρέπουν στη σωστή λειτουργία του προγράμματος του. Όταν ο προγραμματιστής χρειαστεί να ελέγξει το κώδικα που έχει γράψει , μπορεί να πάει στο μενού του Visual Studio Code και να επιλέξει το 'build'. Εκεί το πρόγραμμα εμφανίζει το κατάλληλο μήνυμα , αν ο κώδικας είναι ορθός ή εσφαλμένος. Εάν είναι σωστός τότε ο χρήστης είναι έτοιμος να τρέξει το πρόγραμμα. Στην αντίθετη περίπτωση , όπου το πρόγραμμα αποτυγχάνει να κάνει 'build' , βγαίνει στην οθόνη το σχετικό μήνυμα 'error' και γίνεται αναφορά στο σφάλμα που ανίχνευσε.

Όλες οι παραπάνω , αποτελούν από τις πιο χρήσιμες λειτουργίες του προγράμματος και είναι άκρως απαραίτητες για την υλοποίηση του πειράματος που θα πραγματοποιηθεί. Επομένως , στο παρακάτω (Σχήμα 3.1) παρουσιάζονται δύο από τις λειτουργίες για τις οποίες έγινε νωρίτερα αναφορά. Στη σειρά 74 παρατηρεί κανείς την διαδικασία της ολοκλήρωσης , ενώ στις υπόλοιπες σειρές διακρίνεται χαρακτηριστικά αυτή της σύνταξης , των διαφόρων χρωμάτων. Επομένως , όλες αυτές οι λειτουργίες χρήζουν απαραίτητες για να διευκολύνουν τη ζωή των προγραμματιστών.

3.1.2 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ

Ένα από τα κύρια χαρακτηριστικά του Visual Studio Code είναι ότι δίνει στο προγραμματιστή την δυνατότητα να επιλέξει ο ίδιος σε τι γλώσσα προγραμματισμού θα γράψει το κώδικα του. Υποστηρίζει πλήθος γλωσσών , μερικές από αυτές αρκετά γνωστές, όπως την Python , Java , JavaScript , C , C++ κ.α. Πρόκειται για ένα χρήσιμο πρόγραμμα που δίνει το δικαίωμα στο χρήστη να χρησιμοποιήσει το κώδικα του , για μελλοντικούς σκοπούς , λόγω των πολλαπλών καρτελών που διαθέτει και είναι εύκολο να αποθηκευτούν

σε φακέλους. Συγκεκριμένα , για το έργο μας θα χρησιμοποιηθεί η C++ , όπου θα χρειαστεί η εγκατάσταση της μέσα από τα εργαλεία του ίδιου του προγράμματος.



Σχήμα 3.1 Κάποιες από τις λειτουργίες του Visual Studio Code.

Ακόμα , όπως θα δούμε και παρακάτω , το πρόγραμμα αυτό είναι εύκολο ως προς την λήψη του και την εγκατάσταση του , καθώς μπορεί εύκολα κάποιος να το κατεβάσει εντελώς δωρεάν , χάρη στο πρωτόκολλο μεταφοράς αρχείων (FTP). Αποτελεί ένα πρόγραμμα τελείως αυτόνομο , καθώς δεν χρειάζεται την βοήθεια κάποιου άλλου λογισμικού , για το συγχρονισμό ανάμεσα στον διακομιστή και του προγράμματος που επεξεργάζεται.

Επιπλέον , ακόμη ένα χαρακτηριστικό είναι ότι ο χρήστης έχει στη διάθεση του να ορίσει το χρωματικό θέμα στο περιβάλλον το οποίο δουλεύει. Αυτό μπορεί να το πραγματοποιήσει μέσα από το μενού του προγράμματος , ανοίγοντας το file επιλέγοντας το preferences και από εκεί το color theme. Συγκεκριμένα στο πείραμα μας θα χρησιμοποιηθεί ως θέμα το dark+ για ευνόητους λόγους , όπως στην ευκολία της σύνταξης του κώδικα που γράφουμε , έτσι ώστε να μπορέσουμε να αντιληφθούμε με περισσότερη ευκολία κάποιο πιθανό σφάλμα που θα δημιουργηθεί.

Επομένως , βλέποντας όλα τα παραπάνω χαρακτηριστικά καθώς και τις λειτουργείες του προγράμματος , έφτασε η ώρα της εγκατάστασης και της ανάλυσης των δυνατοτήτων του.

3.1.3 Ο ΤΡΟΠΟΣ ΕΓΚΑΤΑΣΤΑΣΗΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

Σ΄ αυτή την υπό ενότητα , δίνονται οι κατάλληλες πληροφορίες και οδηγίες για την λήψη του Visual Studio Code για τον τρόπο εγκατάστασης του. Είναι σημαντικό γεγονός η

ακολουθία της περιγραφής αυτής , γιατί μια εσφαλμένη εγκατάσταση μπορεί να οδηγήσει σε προβλήματα που θα δυσκολέψουν το έργο μας.

Επίσης απαραίτητη προϋπόθεση , αποτελεί ο χρήστης να γνωρίζει την έκδοση που πρέπει να κατεβάσει από το διαδίκτυο , όπου συνήθως είναι και η πιο πρόσφατη. Μια αρκετά παλιότερη έκδοση μπορεί να κάνει το πρόγραμμα να μην λειτουργεί σωστά. Άρα , ο προγραμματιστής πριν προβεί σε κάποια περαιτέρω ενέργεια , οφείλει να ξέρει το λογισμικό σύστημα του Η/Υ που χρησιμοποιεί , καθώς και την έκδοση του. Ωστόσο , στα θετικά του προγράμματος αυτού είναι ότι ο διακομιστής μπορεί να το κατεβάσει , δίχως να χρειαστεί να επιλέξει κάποια έκδοση , καθώς το πρόγραμμα διαθέτει από μόνο του την πιο σύγχρονη ενημέρωση. Σε περίπτωση που θελήσουμε να αλλάξουμε την έκδοση του προγράμματος μας αυτό μπορούμε να το πραγματοποιήσουμε μέσα από το ίδιο το πρόγραμμα.

Έτσι , τηρώντας όλα τα παραπάνω και αφού έχουμε συνδέσει τον Η/Υ μας με κάποιο ασύρματο (Wi-fi) ή ενσύρματο (Ethernet) δίκτυο , ακολουθούμε προσεκτικά τα παρακάτω βήματα :

1) Αρχικά , θα πρέπει να έχουμε στην διάθεση μας κάποια μηχανή αναζήτησης στον υπολογιστή η οποία θα συνδέεται με το διαδίκτυο , όπως είναι το Google Chrome, το Firefox κτλ. Σε περίπτωση που δεν έχετε στην διάθεση σας τα προγράμματα αυτά μπορείτε να μπείτε από το Internet Explorer να τα κατεβάσετε και να τα εγκαταστήσετε στην επιφάνια εργασίας του υπολογιστή σας.

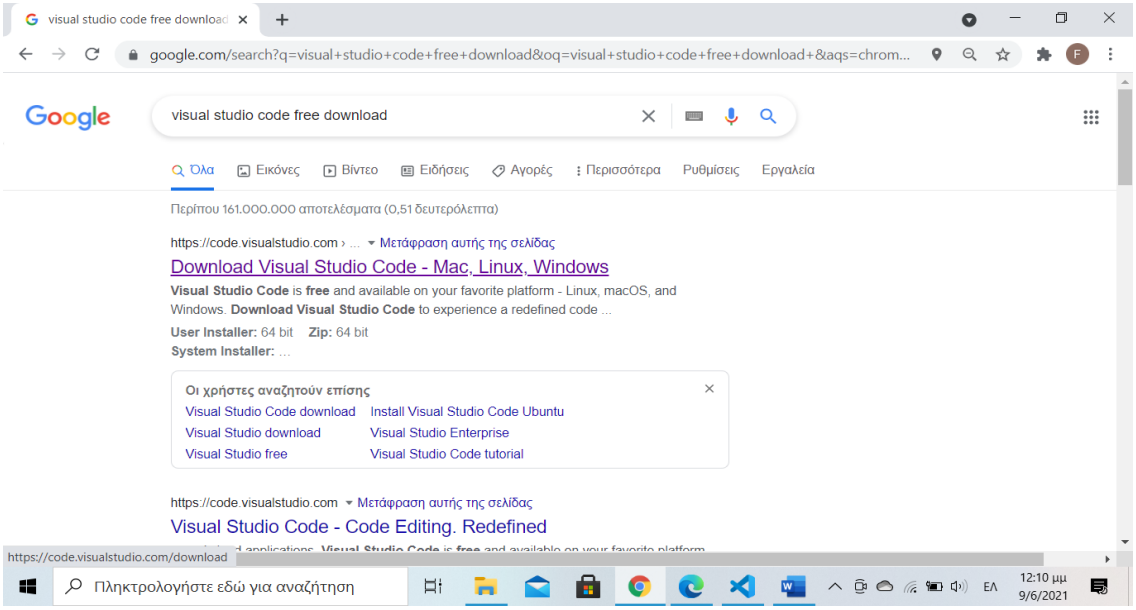
2) Ανοίγετε το πρόγραμμα σας και πάτε στο μενού της αναζήτησης , όπου εκεί θα πληκτρολογήσετε “ visual studio code free download” και πατάτε “Enter”.

3) Αφού πραγματοποιήσετε το βήμα 2 , θα πρέπει να εμφανίζεται στην οθόνη σας το Σχήμα 3.2.

4) Εκεί διαλέγουμε την πρώτη επιλογή (Σχήμα 3.2), η οποία αποτελεί την επίσημη ιστοσελίδα του Visual Studio Code.

5) Στη πορεία θα πρέπει να μας εμφανίσει στην οθόνη το Σχήμα 3.3. Εκεί ανάλογα με το λογισμικό του συστήματος μας επιλέγουμε να κατεβάσουμε το κατάλληλο αρχείο. Στη προκειμένη περίπτωση το λογισμικό σύστημα με το οποίο θα υλοποιηθεί το πείραμα μας είναι τα Windows , εκδόσεως 10. Επομένως , σε αυτή τη περίπτωση οφείλουμε να διαλέξουμε το πρώτο από τα τρία μπλε αρχεία που εμφανίζονται. Κάποιος που διαθέτει

Linux επιλέγει το δεύτερο αρχείο , ενώ για τους χρήστες του macOS συμβατό θεωρείται το τρίτο. Αμέσως μετά θα σας μεταφέρει στη αρχική σελίδα , όπου μπορεί κανείς να αντλήσει πληροφορίες για το πρόγραμμα.



Σχήμα 3.2 : Αναζήτηση και επιλογή της επίσημης ιστοσελίδας του Visual Studio Code από την οποία θα κατεβάσουμε το πρόγραμμα μας.

6) Αφού έχουμε επιλέξει ποιο από τα τρία αρχεία πρέπει να κατεβάσουμε , κάτω αριστερά για τους χρήστες των Windows , βλέπουμε ότι το αρχείο αρχίζει να κατεβαίνει αυτόματα , ενώ διακρίνεται και η έκδοση που κατεβάζουμε συγκεκριμένα την 1.56.2 (Σχήμα 3.3).

7) Όταν ολοκληρωθεί η λήψη του αρχείου , επιλέγετε να το ανοίξετε. Εκεί θα εμφανιστεί μια οθόνη με τις επιλογές “εκτέλεση” και “ακύρωση”. Πατάτε στο σημείο όπου γράφει “εκτέλεση”.

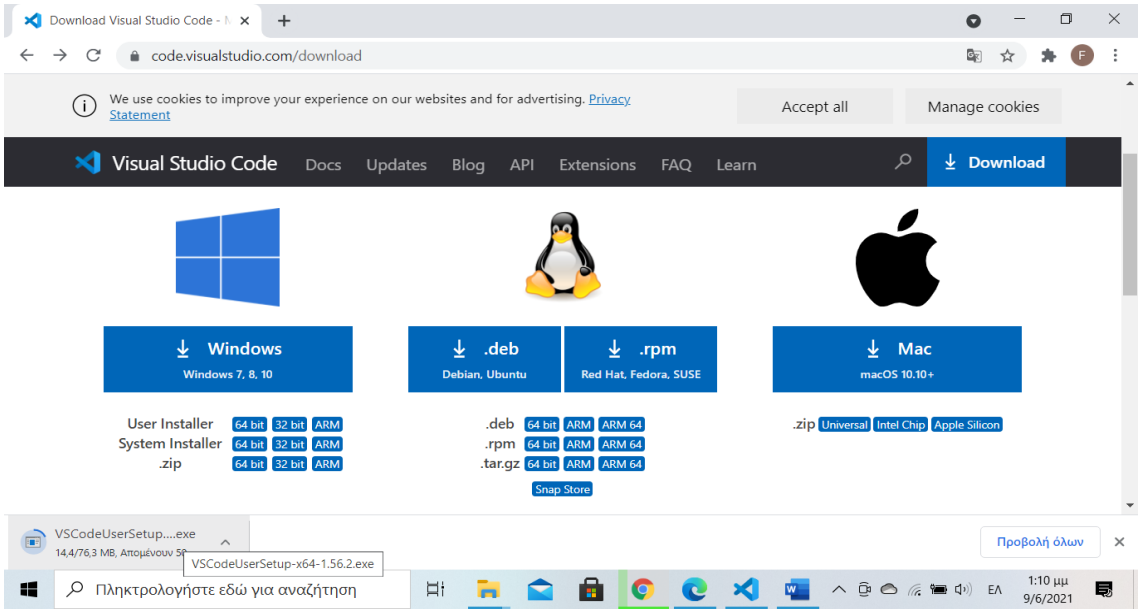
8) Έπειτα μεταφερόσαστε αυτόματα σε ένα νέο πινακάκι το οποίο αφορά τους όρους για την χρήση του προγράμματος μας. Αφού τους διαβάσετε με προσοχή πατάτε πάνω στο “I accept the agreement” και στη πορεία πάνω στο “Next” για να συνεχίσετε την εγκατάστασή σας.

9) Έτσι , προχωράτε στον επόμενο πίνακα στον οποίο διαλέγετε σε ποιο μέρος του υπολογιστή σας θέλετε να κάνετε την εγκατάστασή σας. Μάλιστα στο κάτω μέρος αναγράφεται και το μέγεθος για τον ελεύθερο χώρο , που απαιτείται για την αποθήκευση του προγράμματος. Για να συνεχίσετε παρακάτω , πατάτε και πάλι πάνω στο “Next”.

10) Το επόμενο παράθυρο αναφέρεται στις συντομεύσεις του προγράμματος. Για ακόμη μια φορά επιλέγετε το “Next”.

11) Τέλος , εμφανίζεται ξανά η επιλογή αυτή και μετά την εκτέλεση της ξεκινάει η εγκατάσταση του προγράμματος. Όταν ολοκληρωθεί η εγκατάσταση κάνετε επιλογή πάνω στο “Finish” και είσαστε έτοιμοι για να τρέξετε το πρόγραμμα.

Στην επόμενη ενότητα , θα αναλύσουμε το μενού που διαθέτει το πρόγραμμα και θα δώσουμε βάση στις ιδιότητες των επιλογών και λειτουργειών του περιβάλλοντός του. Επομένως , για την καλύτερη κατανόηση του VS Code βασικό κριτήριο είναι να γνωρίζουμε τις δυνατότητες που έχει , για να μπορέσουμε να τις εφαρμόσουμε κατά την υλοποίηση του έργου. Ακόμα , αναλύεται η εγκατάσταση του Platform IO , καθώς και εκεί θα γίνεται αναφορά για τις λειτουργείες που αυτό διαθέτει. Τέλος , αναπτύσσονται όλα τα απαραίτητα βήματα έτσι ώστε η μικροπλακέτα NRF52840-DK να αναγνωριστεί από αυτό.



Σχήμα 3.3 : Η αυτόματη λήψη του προγράμματος μας , ύστερα από την επιλογή του κατάλληλου αρχείου.

3.2 ΕΠΕΞΗΓΗΣΗ ΤΩΝ ΛΕΙΤΟΥΡΓΕΙΩΝ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ ΚΑΙ ΕΓΚΑΤΑΣΤΑΣΗ ΤΟΥ PLATFORM IO

Σύμφωνα με τα παραπάνω , μετά την επιτυχή εγκατάσταση του VS Code , έφτασε η ώρα της εκτέλεσης του. Με το άνοιγμα το προγράμματος μεταφερόμαστε μέσα στο περιβάλλον του , όπου εκεί διακρίνεται το μενού του (Σχήμα 3.4). Στα αριστερά του, απεικονίζονται κάποια σύμβολα τα οποία αφορούν τις λειτουργείες του προγράμματος , καθώς συμβάλουν και στη διευκόλυνση του χρήστη να αποθηκεύσει, να δημιουργήσει και να τρέξει το έργο του.

Το πρώτο σύμβολο που έχει σαν λογότυπό δύο σελίδες αφορά την αποθήκευση και το άνοιγμα των φακέλων από έργα που ο χρήστης ήδη έχει ή σκοπεύει να υλοποιήσει.

Αμέσως από κάτω , παρατηρεί κανείς το σύμβολο του μεγεθυντικού φακού. Πρόκειται για ένα βοηθητικό εργαλείο , αφού εκεί μπορεί εύκολα κάποιος να αναζητήσει κάποια έκφραση που χρησιμοποίησε στο κώδικα του. Για παράδειγμα , έστω στο κώδικα μας χρησιμοποιούμε την εντολή "print". Αν πάμε στο φακό και πληκτρολογήσουμε την εντολή αυτή , τότε θα μας μεταφέρει αυτόματα στη γραμμή του κώδικα όπου υπάρχει αυτή η εντολή. Επομένως, αποτελεί ένα χρήσιμο εργαλείο, ειδικά όταν έχουμε να κάνουμε με μεγάλα και δυσανάγνωστα προγράμματα.

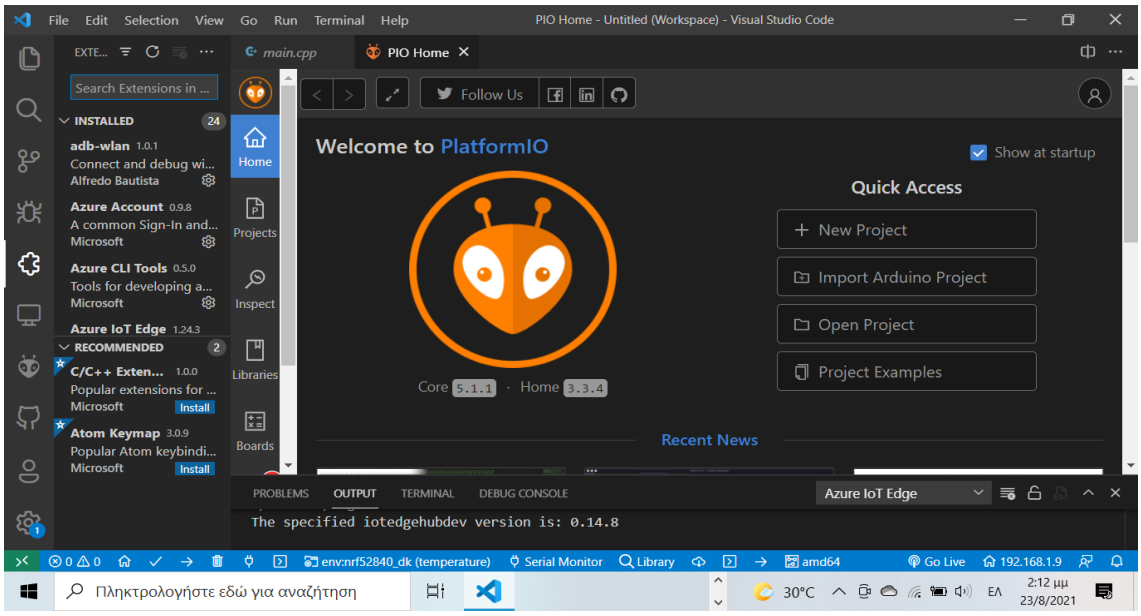
Το επόμενο σύμβολο δηλώνει την διαμεσολάβηση του VS Code με το git , ενώ κάτω από αυτό υπάρχει το σύμβολο του debug. Η λειτουργία αυτή βοηθάει τον χρήστη ώστε να του εντοπίζει τα σφάλματα και να τον ενημερώνει με τα κατάλληλα μηνύματα.

Μία από τις πιο βασικές λειτουργίες αποτελεί αυτή , η οποία έχει για σύμβολό της το εικονίδιο του παζλ. Αυτή αφορά την εγκατάσταση των προγραμμάτων και των γλωσσών που θέλουμε να χρησιμοποιήσουμε για κάποιο έργο. Περιλαμβάνει όλες τις γνωστές γλώσσες προγραμματισμού , όπως είναι η C , C++, Java , Python κ.α. καθώς και σημαντικά προγράμματα όπως είναι το Platform IO στην περίπτωσή μας. Για αυτό το λόγο , πριν καν ξεκινήσουμε την ανάπτυξη του κώδικά μας , πρέπει να γίνει η εγκατάσταση της C++ και του Platform IO. Όταν πραγματοποιηθεί η εγκατάσταση το Platform IO , στην οθόνη του Υ/Η θα εμφανιστεί το παρακάτω Σχήμα 3.4 , όπου αποτελεί την αρχική σελίδα το προγράμματος , ενώ παρατηρείται η δημιουργία ενός νέου συμβόλου με το λογότυπο του εξωγήινου. Το γεγονός αυτό αποδεικνύει ότι το πρόγραμμα εγκαταστάθηκε με επιτυχία. Περισσότερη ανάλυση για το μενού του γίνεται παρακάτω.

Στο Σχήμα 3.4 φαίνονται κι άλλες λειτουργείες οι οποίες είναι κυρίως προγράμματα που έχουν εγκατασταθεί παλιότερα. Για κάποιον που ανοίγει για πρώτη φορά το VS Code μετά το παζλ διακρίνονται ακόμα δύο σύμβολα. Στο πρώτο , διακρίνεται ένα ανθρωπάκι το οποίο αφορά την δημιουργία λογαριασμού. Από την άλλη το γρανάζι είναι κάποιες ρυθμίσεις οι οποίες αφορούν κυρίως τις νέες αναβαθμίσεις του προγράμματος.

Παρόλα αυτά , σχετικές λειτουργείες υπάρχουν και στο πάνω μέρος του προγράμματος , οι οποίες είναι σημαντικές για ανάλυση και αρκετά βοηθητικές ως προς το χρήστη. Κάποιες από αυτές είναι όμοιες με τις προηγούμενες που αναφέρονται παραπάνω. Επομένως , αξίζει να τις δούμε μία προς μία , ως προς το περιεχόμενο τους:

1) File. Όπως και σε άλλα προγράμματα , έτσι και στο VS Code , ο χρήστης μπορεί να επιλέξει κάποιο αρχείο , να το αποθηκεύσει καθώς και να ανοίξει νέο παράθυρο για την δημιουργία ενός καινούριου έργου. Επιπλέον , είναι εφικτή η αλλαγή του θέματος, σύμφωνα με τις προτιμήσεις του προγραμματιστή.
2) Edit. Αφορά λειτουργείες οι οποίες είναι εφικτές από το πληκτρολόγιο όπως αντιγραφή , αποκοπή , επικόλληση κ.α.
3) Selection. Είναι χρήσιμη για την διευκόλυνση του χρήστη , καθώς μπορεί να μετακινήσει τις εντολές του προγράμματος του κάποιες σειρές παραπάνω ή παρακάτω , ανάλογα με τις επιθυμίες του.



Σχήμα 3.4: Το μενού του προγράμματος και η αρχική σελίδα του Platform IO.

4) View. Με την επιλογή αυτή , έχει κανείς τη δυνατότητα να επιλέξει ποιες από τις λειτουργείες επιθυμεί να διακρίνονται στο μενού του VS Code.
5) Go. Σε πολλές περιπτώσεις η επιλογή αυτή είναι αρκετά σημαντική , κυρίως για θέματα όπου εντοπίζεται κάποιο λάθος στο κώδικα. Στην ουσία , επιδεικνύει στο χρήστη σε ποια ακριβώς σειρά δημιουργήθηκε το σφάλμα και του το υπογραμμίζει.
6) Run. Εκεί μπορεί ο προγραμματιστής να κάνει debug το κώδικά του , να τον τρέξει και ενεργοποιήσει το breakpoint στη σειρά όπου αυτός επιθυμεί.
7) Terminal. Από εδώ ενεργοποιούνται τα αποτελέσματα που θα πάρει κανείς στην οθόνη του Η/Υ του , αφού έχει τρέξει το πρόγραμμά του έστω και μία φορά. Επίσης , εμφανίζεται ένα πινακάκι στο κάτω μέρος του μενού , στο οποίο γίνεται λόγος για τα προβλήματα που προέκυψαν , ενώ παράλληλα αναγράφεται και η ποσότητα τους.

8) Help. Αποτελεί την τελευταία επιλογή του μενού , με την οποία μπορεί κανείς να αντλήσει πληροφορίες , με την βοήθεια του διαδικτύου , καθώς τον μεταφέρει στη κεντρική ιστοσελίδα του VS Code.

Ύστερα από την επεξήγηση του προγράμματος , θα πρέπει να γνωρίζουμε και λίγα πράγματα για το Platform IO το βασικό πυλώνα για την ανάπτυξη του κώδικα. Έτσι λοιπόν , μετά την ανάλυση του VS Code , στην επόμενη υπό ενότητα , θα δούμε το μενού του Platform IO , πριν περάσουμε στο προγραμματιστικό κομμάτι του κεφαλαίου 4.

3.2.1 ΤΟ ΜΕΝΟΥ ΤΟΥ PLATFORM IO

Σύμφωνα με το Σχήμα 3.4 που μελετήσαμε προηγουμένως , μπορεί κανείς να διακρίνει το μενού του Platform IO. Το μενού του αποτελεί εξαιρετικά χρήσιμο εργαλείο , καθώς σ΄ αυτό περιέχονται απαραίτητες βιβλιοθήκες για την εκπόνηση του προγράμματος μας. Ακόμα , ορίζεται ως υπεύθυνο για τις μικροπλακέτες , γιατί εκεί ο χρήστης έχει την δυνατότητα να επιλέξει την πλακέτα με την οποία δουλεύει , δημιουργώντας μια συμβατότητα μεταξύ αυτής και του VS Code. Επίσης , ο προγραμματιστής μπορεί με εύκολο τρόπο να ανοίξει ένα από τα έργα του, να δημιουργήσει ένα νέο , αλλά και να αντλήσει πληροφορίες από διάφορα παραδείγματα , για την διευκόλυνση του. Όλα αυτά θα τα μελετήσουμε αμέσως μετά για το τρόπο που χρησιμοποιούνται και ποιες βιβλιοθήκες θα χρειαστεί να εγκαταστήσουμε , για να περάσουμε στο κομμάτι του προγραμματισμού, αποφεύγοντας κάποιο πιθανό σφάλμα που μπορεί να προκύψει.

3.2.2 ΤΑ ΒΗΜΑΤΑ ΤΟΥ ΠΕΙΡΑΜΑΤΟΣ ΣΤΟ PLATFORM IO

Αρχικά , θα πρέπει να σιγουρευτούμε ότι όντως η πλακέτα μας έχει συνδεθεί με τον Η/Υ και έχει αναγνωριστεί από αυτόν. Επομένως , θα πρέπει να συνδέσουμε την πλακέτα με κάποια από τις διαθέσιμες θύρες του υπολογιστή. Αμέσως μετά ενεργοποιούμε τον NRF σύμφωνα με στο Σχήμα 3.5. Μετακινούμε τον SW8 στην θέση ON. Ο SW8 είναι ο διακόπτης ενεργοποίησης του NORDIC NRF 52840-DK. Έπειτα , σέρνουμε τον διακόπτη SW9 στη θέση VDD. Ο SW9 είναι ο διακόπτης που φανερώνει πως τροφοδοτείται η πλακέτα μας. Εάν το τροφοδοτούσαμε με κάποια μπαταρία ο διακόπτης θα ήταν σε θέση Li-Po. Αφού πραγματοποιήσαμε τα παραπάνω , θα ανάψει το LED5 σε χρώμα ανοικτού πράσινου. Στη περίπτωση που το LED5 δεν ανάψει , ο λόγος του προβλήματος οφείλεται στο καλώδιο που συνδέει την πλακέτα μας με τον υπολογιστή. Για το λόγο αυτό , προτιμήστε ένα καλώδιο το οποίο είναι σχετικά πιο παχύ από άλλα όμοια του.

Το λειτουργικό σύστημα όπου παρουσιάζεται το πείραμά μας , πραγματοποιείται με τα Windows 10. Τα παρακάτω βήματα ίσως να διαφέρουν σε διαφορετικά λογισμικά ή εκδόσεις των Windows. Η επιβεβαίωση της σύνδεσης γίνεται με τον εξής τρόπο :

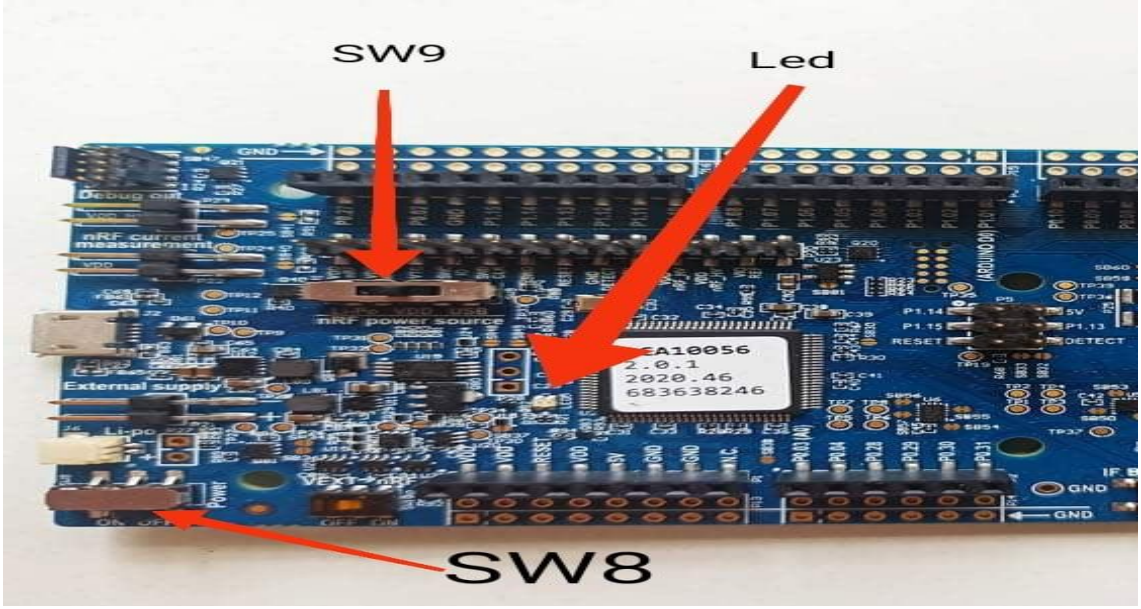
1) Πάμε στην αναζήτηση , δίπλα από την έναρξη.
2) Πληκτρολογούμε «Διαχείριση υπολογιστή» και πατάμε enter .
3) Αμέσως ανοίγει ο πίνακας Διαχείριση υπολογιστή.
4) Εκεί επιλέγουμε την Διαχείριση Συσκευών.
5) Πατάμε πάνω στις θύρες , όπου φαίνονται ποιες συσκευές είναι συνδεδεμένες με τον υπολογιστή μας. Εκεί εμφανίζεται το καλώδιο του Jlink και αναγράφεται σε ποια θύρα είναι συνδεδεμένο (Σχήμα 3.6).

Μετά από αυτά τα βήματα ο μικροελεγκτής έχει συνδεθεί με τον Η/Υ. Παρόλα αυτά , για τον προγραμματισμό του , πρέπει να υπάρξει σύμβαση μεταξύ αυτού και του Visual Studio Code. Στο γεγονός αυτό θα συμβάλει το Platform IO , μέσω του οποίου θα πραγματοποιηθεί η εγκατάσταση του NRF.

Για το πείραμά μας , θα χρειαστεί να κάνουμε κάποιες απαραίτητες ενέργειες , πριν ξεκινήσουμε με την δημιουργία του έργου μας. Επομένως , θα πρέπει να εκπληρώσουμε τα παρακάτω εξής βήματα:

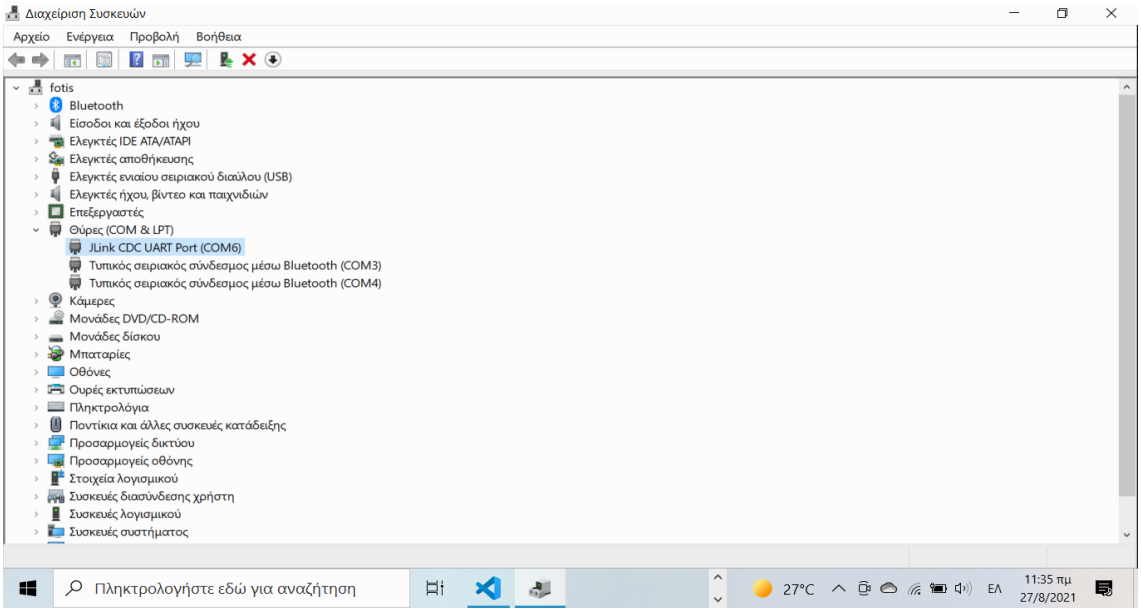
1) Μεταφερόμαστε στο αρχικό μενού του Platform IO.
2) Επιλέγουμε το boards. Εκεί αναζητούμε την πλακέτα με την οποία δουλεύουμε. Επομένως , πληκτρολογούμε Nrf52840-DK.
3) Σαν αποτελέσματα ίσως βγουν παραπάνω από μία πλακέτες. Η πλακέτα του πειράματός μας είναι βασισμένη στην δομή του Arduino, άρα επιλέγουμε πάνω σ΄ αυτή όπου της οποίας η δομή της είναι ο Arduino , ενώ θα πρέπει τα χαρακτηριστικά της να είναι τα ίδια με αυτά της πλακέτας μας. Ο Nrf που δουλεύουμε όπως αναφέραμε και στο κεφάλαιο 1 , διαθέτει για μνήμες 1MB ROM και 256 KB RAM. Αυτό αποτελεί και το κριτήριο για την σωστή επιλογή της πλακέτας.
4) Αφού επιλέξαμε την πλακέτα όπου θέλουμε να δουλέψουμε , ανοίγει μια νέα καρτέλα, η platform.ini. Στην ουσία αυτή η καρτέλα αποτελεί το χώρο στον οποίο αποθηκεύονται όλες οι βιβλιοθήκες που έχουμε εγκαταστήσει , ενώ παράλληλα σ' αυτή , έχουμε τη δυνατότητα μέσω προγραμματισμού να κάνουμε και κάποιες απαραίτητες ρυθμίσεις , έτσι ώστε το VS Code να λειτουργεί με βάση τα θέλω της

πλακέτας μας. Οι πρώτες βιβλιοθήκες έχουν εγκατασταθεί ήδη αυτόματα , ύστερα με όσα αναφέραμε στο βήμα 3.



Σχήμα 3.5 : Τροφοδοσία του Nordic NRF52840-DK.

Σύμφωνα με τα παραπάνω , εκτελώντας με προσοχή όλα τα βήματα , η πλακέτα μας έχει εγκατασταθεί με επιτυχία στο πρόγραμμα που θα δουλέψουμε , δηλαδή με το VS Code. Στο κεφάλαιο που ακολουθεί , αναπτύσσεται και αναλύεται με πλήρη ακρίβεια κάθε σειρά που αφορά τον κώδικα μας , καθώς υπάρχουν και κάποιες απαραίτητες πληροφορίες για την σωστή λειτουργεία του προγράμματος μέσω κάποιων σημαντικών ρυθμίσεων που θα χρειαστεί να πραγματοποιηθούν στη καρτέλα platformio.ini.



Σχήμα 3.6 : Επιβεβαίωση της σύνδεσης του Nordic NRF52840-DK με τον ηλεκτρονικό υπολογιστή.

ΚΕΦΑΛΑΙΟ 4: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΗΣ ΠΛΑΚΕΤΑΣ ΚΑΙ ΑΝΑΛΥΣΗ ΤΟΥ ΚΩΔΙΚΑ

4.1 ΠΑΡΑΤΗΡΗΣΕΙΣ ΠΡΙΝ ΤΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

Πριν μπούμε στο προγραμματιστικό κομμάτι , αξίζει να σημειώσουμε και να αναπτύξουμε κάποιες πληροφορίες για τις ρυθμίσεις που θα πρέπει να πραγματοποιήσουμε , έτσι ώστε ο κώδικάς μας να τρέχει με σωστό τρόπο , καθώς και να αποφευχθούν προβλήματα κατά την λήψη των αποτελεσμάτων. Στο πρώτο κεφάλαιο αναφερθήκαμε στα bits του ελεγκτή μας. Έγινε αναφορά για τα αναλογικά κανάλια που διαθέτει καθώς και για την ανάλυση του. Σ ΄ αυτό το κεφάλαιο θα δούμε πως θα δηλώσουμε τα bits μέσα στο πρόγραμμα μας , γιατί τα προκαθορισμένα bits του VS Code , είναι στα 10. Επομένως , στην καρτέλα που έχει ανοίξει αυτόματα main.cpp , λόγω της εγκατάστασής της γλώσσας C++ θα χρειαστεί να γράψουμε μία εντολή συγκεκριμένη , η οποία θα μετατρέπει τα bits σε 12 , όσα είναι δηλαδή και τα bits του NRF που έχουμε στην διάθεσή μας.

Ακόμα μια καινούρια ερμηνεία που θα δούμε , αφορά αυτή του baud rate. Είναι μια σημαντική ρύθμιση που πρέπει να πραγματοποιηθεί , ειδάλλως δεν θα μπορέσουμε να πάρουμε ξεκάθαρα αποτελέσματα , ενώ θα εμφανίζονται και μηνύματα λάθους κατά την εκτέλεση του κώδικά μας. Όπως και με τα bits , έτσι και εδώ υπάρχει προκαθορισμένος αριθμός baud rate. Επειδή κάθε πλακέτα έχει και το δικό του προκαθορισμένο ρυθμό μεταβολής , θα πρέπει να δούμε ότι ο ρυθμός του προγράμματός μας είναι συμβατός με τον αντίστοιχο της πλακέτας που έχουμε στην διάθεσή μας. Ο ρυθμός και σ' αυτή την περίπτωση , δεν είναι συμβατός με αυτόν του προγράμματος που δουλεύουμε. Επομένως και εδώ θα χρειαστεί να κάνουμε κάποιες ρυθμίσεις και στις 2 καρτέλες , τις platformio.ini και main.cpp. Έτσι στην επόμενη υπό ενότητα αναπτύσσεται η έννοια του baud rate , για τη σημαντικότητα που επιβάλει με την χρήση του.

4.1.1 Η ΕΝΝΟΙΑ ΤΟΥ ΡΥΘΜΟΥ ΜΕΤΑΒΟΛΗΣ (BAUD RATE)

Ο ρυθμός μεταβολής ή αλλιώς baud rate, αφορά τη ταχύτητα και τη συχνότητα με την οποία μια σειριακή γραμμή θα λάβει δεδομένα [7]. Μονάδα μέτρησης αποτελούν τα bps , δηλαδή bits ανά δευτερόλεπτο. Ο συνήθης αριθμός των περισσότερων συσκευών είναι στα 9.600 bps , ενώ οι υπόλοιποι αριθμοί είναι τα 1.200 , 2.400 , 4.800 , 19.200 , 38.400 , 57.600 και τα 115.200 bps. Για την σωστή μεταφορά των δεδομένων θα πρέπει οι συσκευές μας να διαθέτουν τον ίδιο ρυθμό μεταβολής. Όσο πιο μεγάλος είναι αυτός ο

αριθμός τόσο περισσότερος είναι και ο ρυθμός με τον οποίο θα σταλούν δεδομένα από την μια συσκευή στην άλλη. Η μικροπλακέτα μας διαθέτει ρυθμό μεταβολής 115.200 bps , ενώ το πρόγραμμα στο οποίο δουλεύουμε ο προκαθορισμένος ρυθμός μεταβολής βρίσκεται στα 9.600 bps. Επομένως για την σωστή επικοινωνία μεταξύ των δύο αυτών συσκευών , ώστε να λάβουμε ορατά αποτελέσματα , θα πρέπει να ρυθμίσουμε το πρόγραμμά μας να λειτουργεί στα 115.200 bps. Η ρύθμιση αυτή πραγματοποιείται κατά την ανάπτυξη του κώδικα , που ακολουθεί αμέσως μετά.

4.2 ΑΝΑΠΤΥΞΗ ΚΑΙ ΕΠΕΞΗΓΗΣΗ ΤΟΥ ΚΩΔΙΚΑ

Ακολουθεί η ανάπτυξη του κώδικα μας και στις δύο καρτέλες που αναφέραμε , δηλαδή την platformio.ini και την main.cpp , ενώ παράλληλα αναλύεται η κάθε εντολή και γραμμή ξεχωριστά.

Στο Σχήμα 4.1 διακρίνονται οι ρυθμίσεις που πρέπει να γίνουν πριν περάσουμε στο κυρίως κώδικά μας. Στο προηγούμενο κεφάλαιο μελετήσαμε την καρτέλα platform.ini και είπαμε κάποια λόγια γι' αυτή. Όταν κάναμε την εγκατάσταση της πλακέτας μας , άνοιξε η καρτέλα αυτή. Πέρα από τις γραμμές 15 , 16 , 18 και 19 , οι υπόλοιπες εντολές συμπληρώθηκαν αυτόματα κατά την εγκατάσταση , σύμφωνα με τα χαρακτηριστικά της μικροπλακέτας. Τα πράσινα γράμματα δηλώνουν απλά κάποια σχόλια που μπορεί ο χρήστης να δημιουργήσει κατά το γράψιμο του κώδικά του. Σ΄ αυτή τη περίπτωση , έχουμε κάποια σχόλια που εμφανίστηκαν κατά το άνοιγμα της καρτέλας. Ας δούμε όμως τι κάνει κάθε εντολή ξεχωριστά :

Σειρά 12 : Με την εντολή αυτή αναγράφεται η πλατφόρμα με την οποία δουλεύουμε καθώς και σε ποια σειρά της εταιρίας ανήκει. Η πλακέτα μας είναι της σειράς Nrf52 και είναι απαραίτητη η δήλωσή της , άσχετα που τις περισσότερες φορές δημιουργείται κατά την εγκατάσταση , όπως συμβαίνει και στη δική μας περίπτωση.

Σειρά 13 : Η εντολή με την οποία δηλώνεται η πλακέτα όπου δουλεύουμε.

Σειρά 14 : Δηλώνει την μονάδα μικροελεγκτή (MCU) του NRF.

Σειρά 15 : Πρόκειται για την εντολή με την οποία ορίζουμε με ποιο πρωτόκολλο δουλεύουμε. Στο πείραμά μας χρησιμοποιούμε το πρωτόκολλο J-link. Αν κάποιος είχε στη διάθεσή του μία συσκευή Zigbee η οποία είναι συμβατή για να συνδεθεί με τον NRF , θα μπορούσε να δηλώσει αντί για τη θέση του J-link , την IP διεύθυνση του.

Σειρά 16 : Το εργαλείο με το οποίο επιτυγχάνεται ο εντοπισμός σφαλμάτων είναι το J-link.

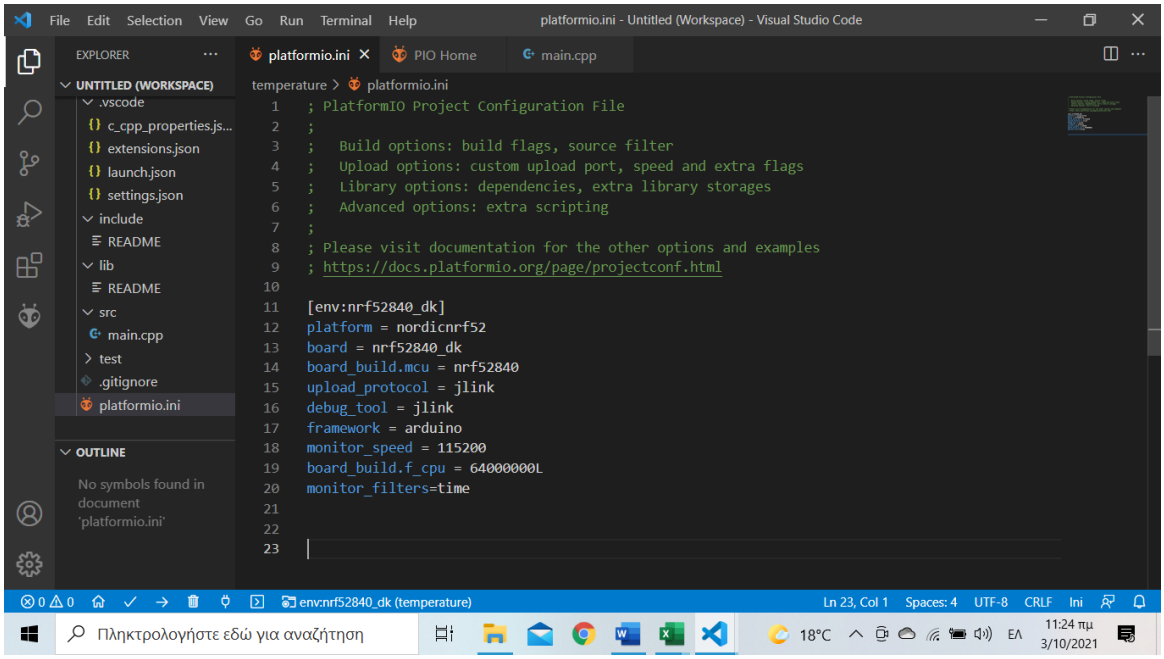
Σειρά 17 : Η εντολή αυτή μας δείχνει ότι η πλακέτα μας έχει ως δομή την πλακέτα του Arduino.

Σειρά 18 : Παραπάνω κάναμε αναφορά για το baud rate και ότι πρέπει να το ρυθμίσουμε με το κατάλληλο τρόπο , έτσι ώστε να μπορέσουμε να πάρουμε ορατά αποτελέσματα. Από τη στιγμή που στη πλακέτα μας το baud rate είναι ίσο με 115.200 bps , θα πρέπει να ορίσουμε τα ίδια bps και στο πρόγραμμά μας.

Σειρά 19 : Δηλώνει την κεντρική μονάδα επεξεργασίας (CPU) του NRF.

Σειρά 20 : Εδώ δηλώνονται η ώρα , τα λεπτά και τα δευτερόλεπτα που θα εμφανίζονται κάθε φορά αυτόματα , κατά την εκτύπωση αποτελεσμάτων στο κυρίως μέρος του κώδικα.

Αφού πραγματοποιήθηκαν όλες οι απαραίτητες ρυθμίσεις , είμαστε έτοιμοι να περάσουμε στο κυρίως κομμάτι του κώδικά μας. Σκοπός του προγράμματος μας είναι να μπορέσουμε από τα τέσσερα αναλογικά κανάλια , να πάρουμε τιμές σε ψηφιακή μορφή , καθώς ανάλογα με τις τιμές που λαμβάνει ο αισθητήρας μας , θα αποτυπώνεται και πόση τάση καταναλώνει σε κάθε μέτρηση. Επομένως , μεταφερόμαστε στη καρτέλα main.cpp , όπου εκεί σύμφωνα με τα παρακάτω σχήματα (Σχήμα 4.2 , Σχήμα 4.3, Σχήμα 4.4 , Σχήμα 4.5) , αναπτύσσεται ο κώδικάς μας , ενώ παράλληλα διατυπώνεται η σαφής επεξήγηση των εντολών.



Σχήμα 4.1 : Οι ρυθμίσεις που πρέπει να πραγματοποιήσουμε στη καρτέλα platformio.ini.

Ακολουθεί η επεξήγηση του κώδικα του σχήματος (Σχήμα 4.2) :

Σειρά 1 : Πρόκειται για μια βιβλιοθήκη που πρέπει να δηλώσουμε , καθώς η δομή της πλακέτας μας είναι βασισμένη στο Arduino , υλοποιημένη γλώσσα της C++. Αποτελεί την αρχή του προγράμματός μας.

Σειρά 2 : Η βιβλιοθήκη αυτή μπορεί να παραληφθεί στην περίπτωσή μας. Η ύπαρξή της στο σύγγραμμα είναι για κάποιον που έχει προχωρήσει με τον τρόπο της IP.

Σειρά 3 : Είναι μια σημαντική εντολή που δε πρέπει να παραληφθεί στη γλώσσα του Arduino. Αν δεν δηλωθεί θα υπάρξει πρόβλημα για τη συνέχεια του κώδικά μας. Όπως και στο Arduino , έτσι και στο στον NRF , υπάρχει ένα led λειτουργείας το οποίο είναι ενσωματωμένο στη πλακέτα και είναι συνδεδεμένο με το pin 13. Από εκεί προκύπτει και ο αριθμός αυτός.

Σειρά 4 : Για να δηλώσουμε το αναλογικό κανάλι P0.03 (A0) , ορίζουμε μια μεταβλητή, με όνομα tempPin1.

Σειρά 5 : Για να δηλώσουμε το αναλογικό κανάλι P0.04 (A1) , ορίζουμε μια μεταβλητή, με όνομα tempPin2.

Σειρά 6 : Για να δηλώσουμε το αναλογικό κανάλι P0.28 (A2) , ορίζουμε μια μεταβλητή, με όνομα tempPin3.

Σειρά 7 : Για να δηλώσουμε το αναλογικό κανάλι P0.30 (A4) , ορίζουμε μια μεταβλητή, με όνομα tempPin4.

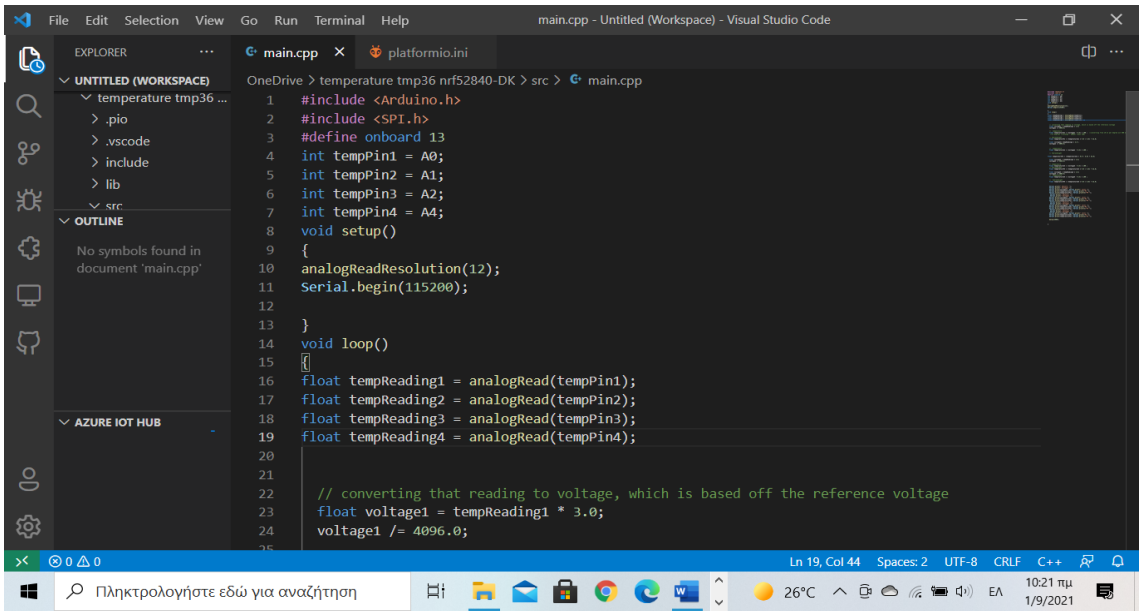
Σειρά 8 : Την εντολή αυτή τη συναντάμε σχεδόν σε όλα τα προγράμματα και αποτελεί μία συνάρτηση μέσα στην οποία μπορούμε να ορίσουμε διάφορα πράγματα.

Σειρά 9 : Είναι ένα άγκιστρο το οποίο σηματοδοτεί την έναρξη για μια ομάδα εντολών που επρόκειτο να ακολουθήσουν.

Σειρά 10 : Με την εντολή αυτή δηλώνουμε ότι το πρόγραμμα μας θέλουμε να δουλεύει με την ανάλυση των 12 bit , όσα είναι και τα bit του NRF.

Σειρά 11 : Στη γραμμή αυτή γίνεται η δήλωση του ρυθμού μεταβολής , ώστε το πρόγραμμα μας να δουλεύει στα 115.200 bps. Ο σκοπός της εντολής αυτής αναφέρθηκε προηγουμένως.

Σειρά 12 : Όπως θα παρατηρήσει κανείς υπάρχουν κι άλλες κενές σειρές μέσα στο κώδικα. Οι σειρές που θα περιέχουν κενά αλλά και σχόλια από εδώ και πέρα , θα παραλείπονται.



Σχήμα 4.2 : Ανάπτυξη του κώδικα για τις σειρές 1 έως και 24.

Σειρά 13 : Είναι ένα άγκιστρο το οποίο σηματοδοτεί το τερματισμό για μια ομάδα εντολών που προηγήθηκαν. Σχετίζεται με το άγκιστρο της σειράς 9 , καθώς για κάθε αρχή θα πρέπει να υπάρχει και ένα τέρμα , ειδάλλως θα εμφανιστεί μήνυμα λάθους.

Σειρά 14 : Πρόκειται για ένα βρόχο μέσα στον οποίο αναπτύσσεται το κυρίως μέρος του κώδικα και μπορεί να τρέξει άπειρες φορές.

Σειρά 15 : Ότι ακριβώς ισχύει και στη σειρά 9.

Σειρά 16 : Δήλωση της μεταβλητής tempReading1 και διάβασμα της αναλογικής τιμής που λαμβάνει από το κανάλι A0.

Σειρά 17 : Δήλωση της μεταβλητής tempReading2 και διάβασμα της αναλογικής τιμής που λαμβάνει από το κανάλι A1.

Σειρά 18 : Δήλωση της μεταβλητής tempReading3 και διάβασμα της αναλογικής τιμής που λαμβάνει από το κανάλι A2.

Σειρά 19 : Δήλωση της μεταβλητής tempReading4 και διάβασμα της αναλογικής τιμής που λαμβάνει από το κανάλι A4.

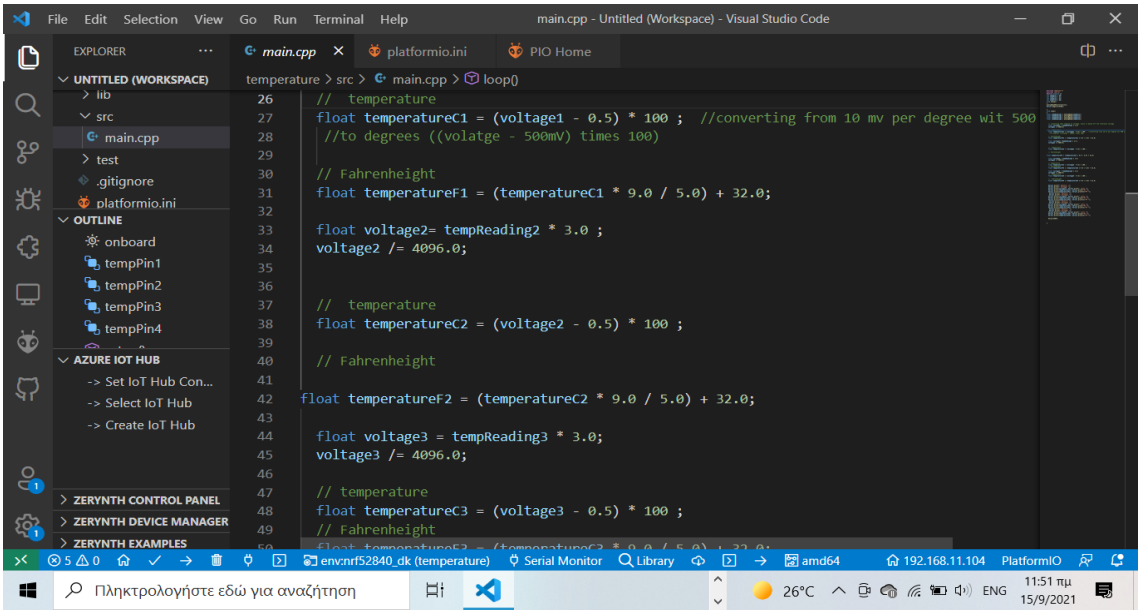
Σειρά 23 : Δήλωση της τάσης ως μεταβλητή voltage1 και ότι αυτή είναι ίση με τη τιμή του αναλογικού καναλιού A0 , πολλαπλασιασμένη με την τάση αναφοράς , που αναλύσαμε στο κεφάλαιο 1.

Σειρά 24 : Χρήσιμη εντολή για τη μετατροπή της αναλογικής τιμής της τάσης σε ψηφιακή. Επειδή ο ελεγκτής μας είναι 12 bit , άρα θα χρειαστεί να διαιρεθεί με τον αριθμό 4.096 και όπως είχαμε αναφέρει , προκύπτει από τη μετατροπή 2 ^ 12 = 4.096.

Ακολουθεί η επεξήγηση του κώδικα του σχήματος (Σχήμα 4.3) :

Σειρά 27 : Δήλωση της μεταβλητής temperatureC1. Πρόκειται για τη μεταβλητή της θερμοκρασίας του πρώτου αισθητήρα , η οποία μετριέται σε βαθμούς κελσίου και είναι ίση με την διαφορά της ψηφιακής τάσης voltage1 , μειωμένη κατά 0.5 (τιμή σφάλματος) και πολλαπλασιασμένη με το 100. Η τιμή 0.5 προέκυψε μέσα από πηγές που προαναφέρθηκαν ότι ο αισθητήρας για τάση 0.75 volt δείχνει 25 βαθμούς κελσίου. Σε επόμενο κεφάλαιο , θα αποδείξουμε ότι λαμβάνουμε 10mV ανά μονάδα θερμοκρασίας , εξίσου και η τιμή 100 στην εντολή μας.

Σειρά 31 : Δήλωση της μεταβλητής temperatureF1 , για τη μέτρηση της θερμοκρασίας του πρώτου αισθητήρα , σε βαθμούς Φαρενάϊτ. Η εντολή αυτή μετατρέπει τους βαθμούς Κελσίου σε Φαρενάϊτ.



Σχήμα 4.3 : Ανάπτυξη του κώδικα για τις σειρές 25 έως και 49.

Σειρά 33 : Δήλωση της τάσης ως μεταβλητή voltage2 και ότι αυτή είναι ίση με τη τιμή του αναλογικού καναλιού A1 , πολλαπλασιασμένη με την τάση αναφοράς .

Σειρά 34 : Χρήσιμη εντολή για τη μετατροπή της αναλογικής τιμής της τάσης σε ψηφιακή. Επειδή ο ελεγκτής μας είναι 12 bit , άρα θα χρειαστεί να διαιρεθεί με τον αριθμό 4.096 και όπως είχαμε αναφέρει , προκύπτει από τη μετατροπή 2 ^ 12 = 4.096.

Σειρά 38 : Δήλωση της μεταβλητής temperatureC2. Πρόκειται για τη μεταβλητή της θερμοκρασίας του δεύτερου αισθητήρα , η οποία μετριέται σε βαθμούς κελσίου και είναι ίση με την διαφορά της ψηφιακής τάσης voltage2 , μειωμένη κατά 0.5 (τιμή σφάλματος) και πολλαπλασιασμένη με το 100. Η τιμή 0.5 προέκυψε μέσα από πηγές που προαναφέρθηκαν ότι ο αισθητήρας για τάση 0.75 volt δείχνει 25 βαθμούς κελσίου. Ακόμα , λαμβάνουμε 10mV ανά μονάδα θερμοκρασίας , εξίσου και η τιμή 100 στην εντολή μας.

Σειρά 42 : Δήλωση της μεταβλητής temperatureF2 , για τη μέτρηση της θερμοκρασίας του δεύτερου αισθητήρα , σε βαθμούς Φαρενάϊτ. Η εντολή αυτή μετατρέπει τους βαθμούς Κελσίου σε Φαρενάϊτ.

Σειρά 44 : Δήλωση της τάσης ως μεταβλητή voltage3 και ότι αυτή είναι ίση με τη τιμή του αναλογικού καναλιού A2 , πολλαπλασιασμένη με την τάση αναφοράς .

Σειρά 45 : Χρήσιμη εντολή για τη μετατροπή της αναλογικής τιμής της τάσης σε ψηφιακή. Επειδή ο ελεγκτής μας είναι 12 bit , άρα θα χρειαστεί να διαιρεθεί με τον αριθμό 4.096 και όπως είχαμε αναφέρει , προκύπτει από τη μετατροπή 2 ^ 12 = 4.096.

Σειρά 48 : Δήλωση της μεταβλητής temperatureC3. Πρόκειται για τη μεταβλητή της θερμοκρασίας του τρίτου αισθητήρα , η οποία μετριέται σε βαθμούς κελσίου και είναι ίση με την διαφορά της ψηφιακής τάσης voltage3 , μειωμένη κατά 0.5 (τιμή σφάλματος) και πολλαπλασιασμένη με το 100. Η τιμή 0.5 προέκυψε μέσα από πηγές που προαναφέρθηκαν ότι ο αισθητήρας για τάση 0.75 volt δείχνει 25 βαθμούς κελσίου. Λαμβάνουμε 10mV ανά μονάδα θερμοκρασίας , εξίσου και η τιμή 100 στην εντολή μας.

Ακολουθεί η επεξήγηση του κώδικα του σχήματος (Σχήμα 4.4) :

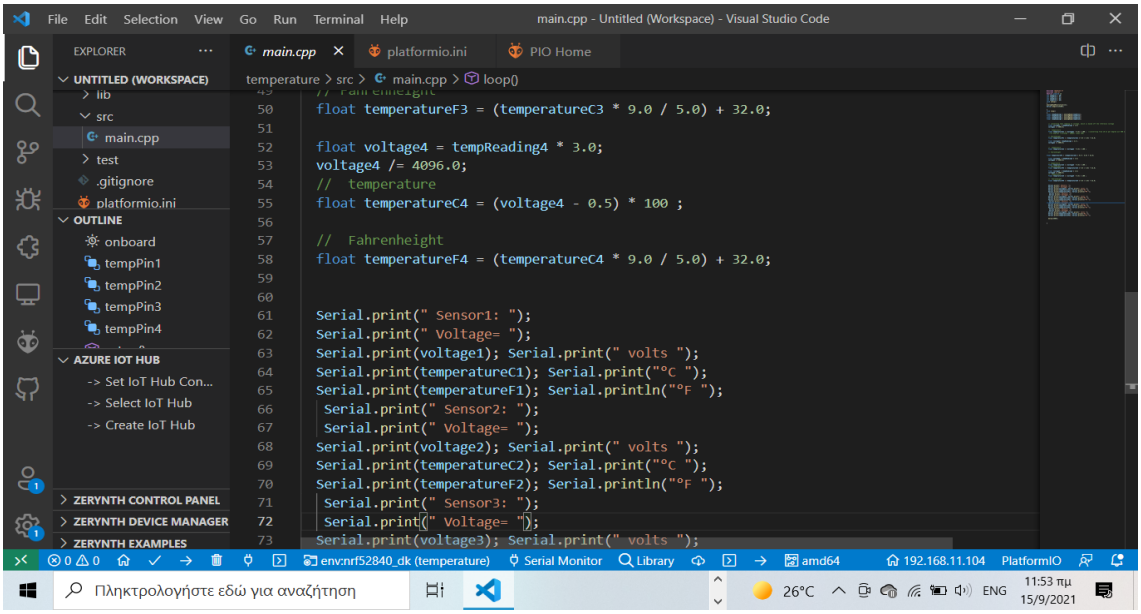
Σειρά 50 : Δήλωση της μεταβλητής temperatureF3 , για τη μέτρηση της θερμοκρασίας του τρίτου αισθητήρα , σε βαθμούς Φαρενάϊτ. Η εντολή αυτή μετατρέπει τους βαθμούς Κελσίου σε Φαρενάϊτ.

Σειρά 52 : Δήλωση της τάσης ως μεταβλητή voltage4 και ότι αυτή είναι ίση με τη τιμή του αναλογικού καναλιού A4 , πολλαπλασιασμένη με την τάση αναφοράς .

Σειρά 53 : Χρήσιμη εντολή για τη μετατροπή της αναλογικής τιμής της τάσης σε ψηφιακή. Επειδή ο ελεγκτής μας είναι 12 bit , άρα θα χρειαστεί να διαιρεθεί με τον αριθμό 4.096 και όπως είχαμε αναφέρει , προκύπτει από τη μετατροπή 2 ^ 12 = 4.096.

Σειρά 55 : Δήλωση της μεταβλητής temperatureC4. Πρόκειται για τη μεταβλητή της θερμοκρασίας του τέταρτου αισθητήρα , η οποία μετριέται σε βαθμούς κελσίου και είναι ίση με την διαφορά της ψηφιακής τάσης voltage3 , μειωμένη κατά 0.5 (τιμή σφάλματος) και πολλαπλασιασμένη με το 100. Η τιμή 0.5 προέκυψε μέσα από πηγές που προαναφέρθηκαν ότι ο αισθητήρας για τάση 0.75 volt δείχνει 25 βαθμούς κελσίου. Λαμβάνουμε 10mV ανά μονάδα θερμοκρασίας , εξίσου και η τιμή 100 στην εντολή μας.

Σειρά 58 : Δήλωση της μεταβλητής temperatureF4 , για τη μέτρηση της θερμοκρασίας του τέταρτου αισθητήρα , σε βαθμούς Φαρενάϊτ. Η εντολή αυτή μετατρέπει τους βαθμούς Κελσίου σε Φαρενάϊτ.



Σχήμα 4.4 : Ανάπτυξη του κώδικα για τις σειρές 50 έως και 72.

Σειρές 61 έως και 65 : Η διαφορά της εντολής Serial.print με την Serial.println είναι ότι στη πρώτη περίπτωση θα εμφανίσει το αποτέλεσμα στην ίδια γραμμή , ενώ με την δεύτερη εντολή δηλώνουμε ότι αλλάζουμε γραμμή και τα επόμενα αποτελέσματα θα εμφανιστούν σε ξεχωριστή γραμμή. Επομένως , θα εκτυπώνει στην ίδια γραμμή όλα τα παραπάνω που αναγράφονται μέχρι να συναντήσει την εντολή Serial.println. Όσα αναγράφονται μέσα σε “ ” αποτελούν χαρακτήρες , ενώ τα υπόλοιπα είναι οι μεταβλητές.

Σειρές 66 έως και 72 : Εκτύπωση χαρακτήρων και μεταβλητών.

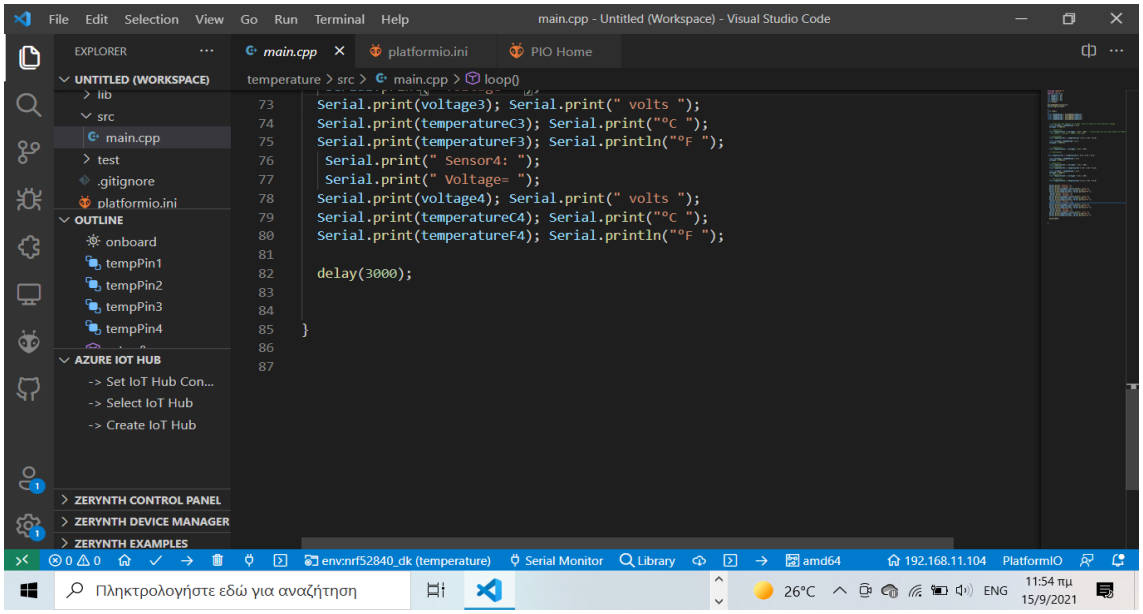
Ακολουθεί η επεξήγηση του κώδικα του σχήματος (Σχήμα 4.5) :

Σειρές 73 έως και 80 : Εκτύπωση χαρακτήρων και μεταβλητών.

Σειρά 82 : Η εντολή αυτή , δηλώνει ότι κάθε 3 δευτερόλεπτα θα εμφανίζονται νέες μετρήσεις.

Σειρά 85 : Ότι ισχύει ακριβώς και στη σειρά 13.

Κάπου εδώ ολοκληρώνεται η ανάπτυξη του κώδικα και η επεξήγησή του και στην επόμενη υποενότητα θα περιγράψουμε το τρόπο με τον οποίο θα ελέγξουμε το πρόγραμμα και θα το τρέξουμε.

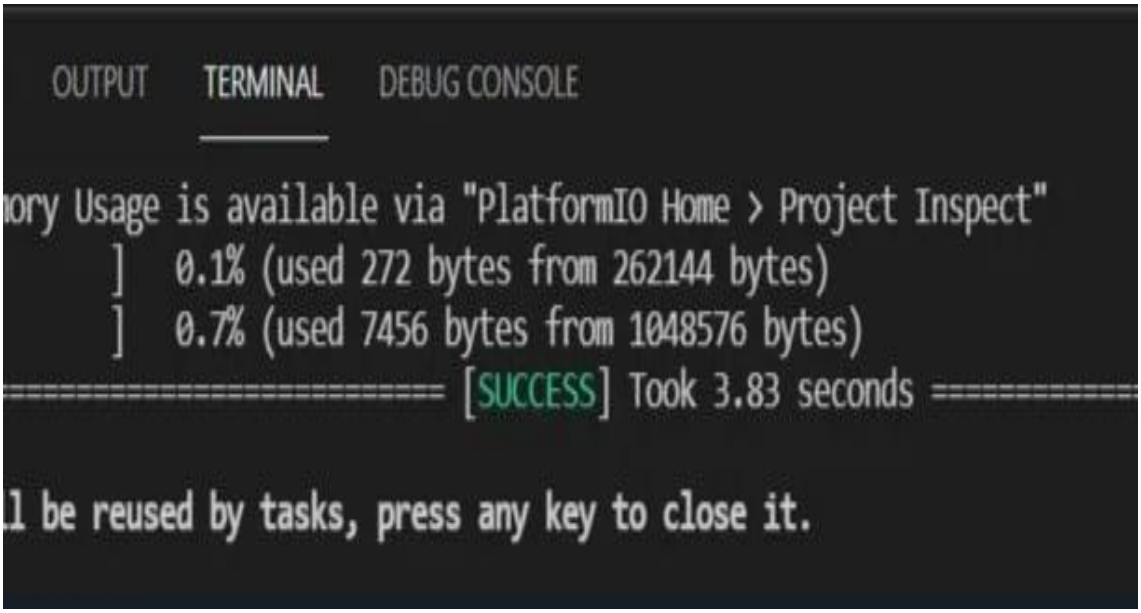


Σχήμα 4.5 : Ανάπτυξη του κώδικα για τις σειρές 73 έως και 85.

4.2.1 ΕΚΤΕΛΕΣΗ ΤΟΥ ΚΩΔΙΚΑ

Αφού λοιπόν γράψαμε και επεξηγήσαμε το κώδικά μας ήρθε η ώρα να το τρέξουμε. Έτσι , πάμε στο μενού και επιλέγουμε το run , start debugging ή πολύ απλά πατάμε το κουμπί F5 , για να ξεκινήσουμε τον εντοπισμό σφαλμάτων. Διαφορετικά , πάμε στο μενού που έχουμε στα αριστερά , πάνω στο σύμβολο του platform io και επιλέγουμε build. Στο κάτω μέρος της οθόνης και συγκεκριμένα στο terminal θα πρέπει και στις δύο περιπτώσεις , να εμφανιστεί η λέξη success με πράσινα γράμματα (Σχήμα 4.6). Αυτό σημαίνει , ότι δε παρουσιάστηκε κάποιο σφάλμα και μπορούμε να ανεβάσουμε το κώδικά μας. Στην αντίθετη περίπτωση , όπου θα εντοπιστεί κάποιο λάθος , θα παρατηρήσει κανείς τη λέξη failed με κόκκινα γράμματα , επομένως θα πρέπει να πάει να στο κώδικά του και να το εντοπίσει , έτσι ώστε να μπορέσει να το διορθώσει και να τρέξει επιτυχώς το πρόγραμμά

του. Καθώς ο κώδικάς δεν περιέχει κανένα σφάλμα , επιλέγουμε το εικονίδιο του platform io και από εκεί upload and monitor. Πάλι στο terminal , θα εμφανιστεί η λέξη success και μετά από λίγο θα πρέπει να εμφανίζονται τα αποτελέσματα του κώδικά μας.

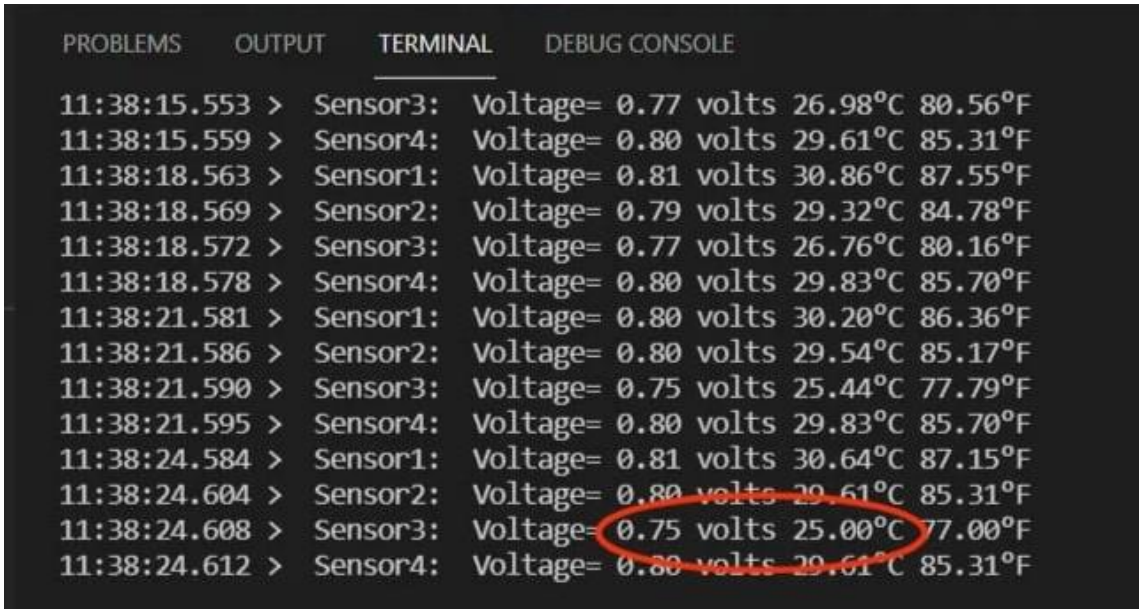


Σχήμα 4.6 : Το πρόγραμμα αναπτύχθηκε και ανέβηκε με επιτυχία.

ΚΕΦΑΛΑΙΟ 5: ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΒΑΘΜΟΝΟΜΗΣΗ

5.1 ΕΜΦΑΝΙΣΗ ΤΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Όπως αναφέραμε και παραπάνω , μετά την επιλογή upload and monitor , θα πρέπει να εμφανιστούν τα αποτελέσματα από τις τιμές που λαμβάνουμε , από το κώδικα που αναπτύξαμε. Επομένως τα αποτελέσματα θα πρέπει να παρουσιάζονται σύμφωνα με τη μορφή του Σχήματος 5.1. Άρα για κάθε τρία δευτερόλεπτα , καταφέραμε να μετρήσουμε τη τάση που υπάρχει σε καθένα από τα τέσσερα αναλογικά κανάλια του ελεγκτή μας , ανάλογα με τη θερμοκρασία που έχουμε κάθε φορά. Είμαστε όμως σίγουροι ότι οι αισθητήρες μας λειτουργούν σωστά και αποτελεσματικά ; Στο φύλλο δεδομένων του TMP36 , αναγράφεται ότι για κάθε μία μονάδα αύξησης θερμοκρασίας σε βαθμούς κελσίου , η τάση ανεβαίνει κατά 0,01V. Επομένως , η επόμενη υποενότητα αναφέρεται στις μεθόδους που χρησιμοποιούνται για την ακριβή μέτρηση των θερμοκρασιών , ενώ πιο κάτω πραγματοποιείται βαθμονόμηση μεταξύ βαθμών κελσίου και τάσης , για ένα εύρος τιμών που εμείς ορίζουμε.



Σχήμα 5.1 : Ο τρόπος εμφάνισης των αποτελεσμάτων.

5.1.1 ΜΕΘΟΔΟΣ ΕΞΑΚΡΙΒΩΣΗΣ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Πριν προχωρήσουμε στη βαθμονόμηση των θερμοκρασιών , θα πρέπει να είμαστε σίγουροι ότι οι αισθητήρες μας μετράνε με απόλυτη ακρίβεια. Για να πετύχουμε κάτι τέτοιο , θα μπορούσαμε να συγκρίνουμε τις τιμές που παίρνουμε , για κάποια χρονική στιγμή , με τις τιμές που λαμβάνουμε από ένα άλλο θερμόμετρο ότι είναι ίδιες. Το θερμόμετρο που

χρησιμοποιείται για το πείραμά μας δεν είναι όπως τα υπόλοιπα κοινά. Πρόκειται για ένα θερμόμετρο , το οποίο έχει τη δυνατότητα να συνδεθεί με τον Η/Υ σε μία θύρα USB , ενώ παράλληλα ένα καλώδιο που συνδέεται μ΄ αυτό , διαθέτει στην έξοδο του μία ακίδα , η οποία ακουμπάει τον αισθητήρα TMP36. Τα αποτελέσματα των μετρήσεων παρουσιάζονται σε ένα πινακάκι το οποίο προκύπτει, ύστερα από την εγκατάσταση του στο λειτουργικό σύστημα όπου δουλεύουμε.

Το θερμόμετρο με το οποίο γίνεται η μέτρηση των θερμοκρασιών στο πείραμα μας είναι αυτό της βρετανικής εταιρίας Lascar. Για να το προμηθευτείτε , αρκεί να επισκεφτείτε τη κεντρική ιστοσελίδα της εταιρίας www.lascarelectronics.com. Το πακέτο αυτό περιλαμβάνει ένα Compact Disc (CD) , το οποίο είναι απαραίτητο για την εγκατάσταση του προγράμματος που θα χρησιμοποιηθεί καθώς εκεί καταγράφονται οι τιμές του θερμομέτρου. Αφού λοιπόν η εγκατάσταση ολοκληρωθεί με επιτυχία , στην επιφάνεια εργασίας στου Η/Υ θα εμφανιστεί ένα εικονίδιο , με ονομασία EasyLog USB. Πριν όμως ανοίξουμε το πρόγραμμα που εγκαταστήσαμε , πρέπει να συνδέσουμε το καλώδιο που θα βρούμε μέσα σ' αυτό το πακέτο με το θερμόμετρο. Είναι ένα καλώδιο που διαθέτει στη μια πλευρά του 2 ακροδέκτες «+» και «-» οι οποίοι καταλήγουν στη θετική και αρνητική πλευρά του θερμομέτρου αντίστοιχα , ενώ στην άλλη άκρη του υπάρχει μία ακίδα η οποία καταλήγει πάνω στον αισθητήρα μας. Η ένωση της ακίδας με τον αισθητήρα μπορεί να γίνει με τη χρήση από μανταλάκι. Άρα μετά την εγκατάσταση και τη σύνδεση ανοίγουμε το πρόγραμμα όπου εκεί υπάρχουν τρείς επιλογές.

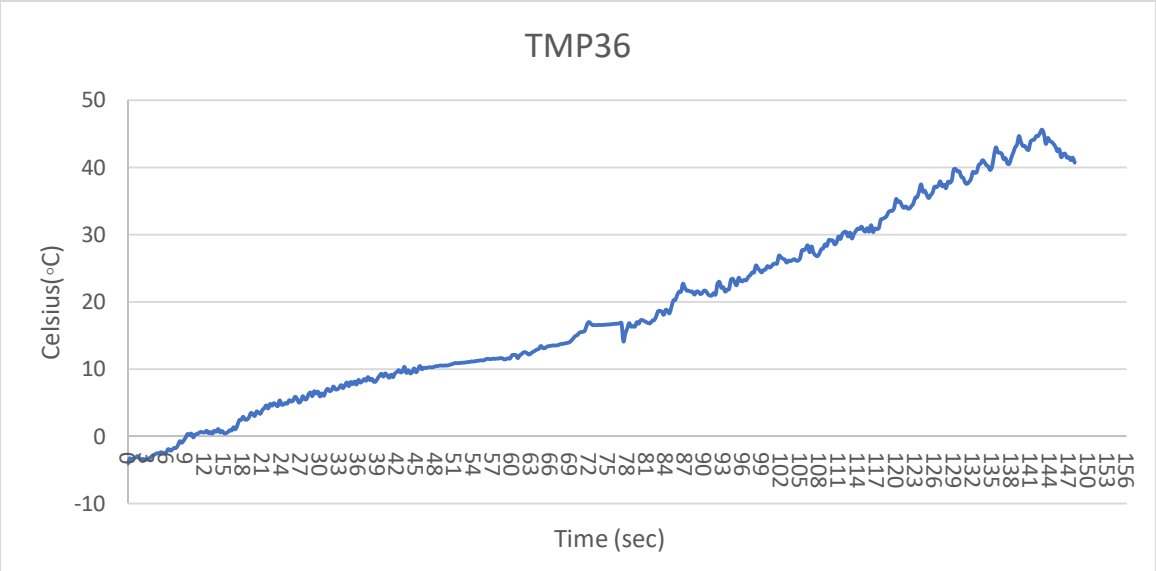
Η πρώτη επιλογή αφορά σε τι μονάδα μέτρησης θερμοκρασίας θέλουμε να εμφανίσει τα αποτελέσματα , ενώ παράλληλα μπορούμε να ορίσουμε και τη χρονική στιγμή για τη λήψη των θερμοκρασιών. Η δεύτερη επιλογή είναι να για να σταματήσουμε να λαμβάνουμε θερμοκρασίες και να δούμε τις τιμές που καταγράφηκαν για τη συγκεκριμένη τιμή που ορίσαμε. Η τρίτη επιλογή , εμφανίζει τα αποτελέσματα της τελευταίας μέτρησης.

Αν ακολουθήσει κανείς από την αρχή μέχρι και εδώ όλες τις απαραίτητες ενέργειες και βήματα που αναφέρθηκαν έως τώρα , θα μπορέσει να υλοποιήσει το πείραμα χωρίς να αντιμετωπίσει κάποιο ιδιαίτερο πρόβλημα. Επομένως θα είναι σε θέση να λαμβάνει τις τιμές του αναλογικού αισθητήρα σε ψηφιακή μορφή , ενώ ταυτόχρονα μπορεί να εξακριβώσει ότι τα αποτελέσματά του είναι σωστά. Αφού λοιπόν γνωρίζουμε ότι ο αισθητήρας εμφανίζει ορθές μετρήσεις και με βάση το φύλλο δεδομένων του , πρέπει να σιγουρευτούμε ότι η τάση του είναι κι αυτή ορθή. Όταν ο αισθητήρας μας μετρήσει 25

βαθμούς κελσίου , η τάση που πρέπει να έχουμε κατά την έξοδο , βρίσκεται στα 0,75 V , γεγονός το οποίο επιβεβαιώνεται και στο Σχήμα 5.1 το οποίο αποτελεί τις μετρήσεις του πειράματος. Η επόμενη ενότητα αναφέρεται τη βαθμονόμηση μεταξύ θερμοκρασιών και αντίστοιχων τάσεων , για ένα εύρος 50 τιμών , όπου εκεί αποδεικνύεται ότι η τάση που λαμβάνουμε είναι σωστή , ενώ παρουσιάζεται και η ανάλυση των αποτελεσμάτων που πάρθηκαν από τα δύο συστήματα , περί ίδιας χρονικής στιγμής.

5.2 ΒΑΘΜΟΝΟΜΗΣΗ ΘΕΡΜΟΚΡΑΣΙΩΝ ΚΑΙ ΤΑΣΗΣ

Ο αισθητήρας TMP36 όπως αναγράφεται και στο κεφάλαιο 2 , μετράει για ένα εύρος θερμοκρασιών από -40 έως και 150 βαθμούς κελσίου. Για την βαθμονόμηση του πειράματος χρησιμοποιήθηκε ένα μικρότερο εύρος τιμών , καθώς οι πολύ χαμηλές και υψηλές θερμοκρασίες είναι δύσκολο να αναπτυχθούν σε ένα κλειστό εργαστηριακό χώρο, ένα χώρο δωματίου. Επομένως στο πείραμά μας λάβαμε θερμοκρασίες αρκετά πιο εφικτές, οι οποίες κυμαίνονται από -5 έως και 44 βαθμούς κελσίου. Για τις τιμές από -5 έως και 15 έγινε χρήση της κατάψυξης , για 16 έως και 30 οι μετρήσεις πάρθηκαν στον περιβάλλοντα χώρο του δωματίου , και για θερμοκρασίες από 31 έως και 44 απαιτήθηκε χρήση πολύφωτου με λάμπα φθορίου. Στο Σχήμα 5.2 διακρίνονται τα αποτελέσματα των μετρήσεων που έχουν ληφθεί για κάθε 3 δευτερόλεπτα.

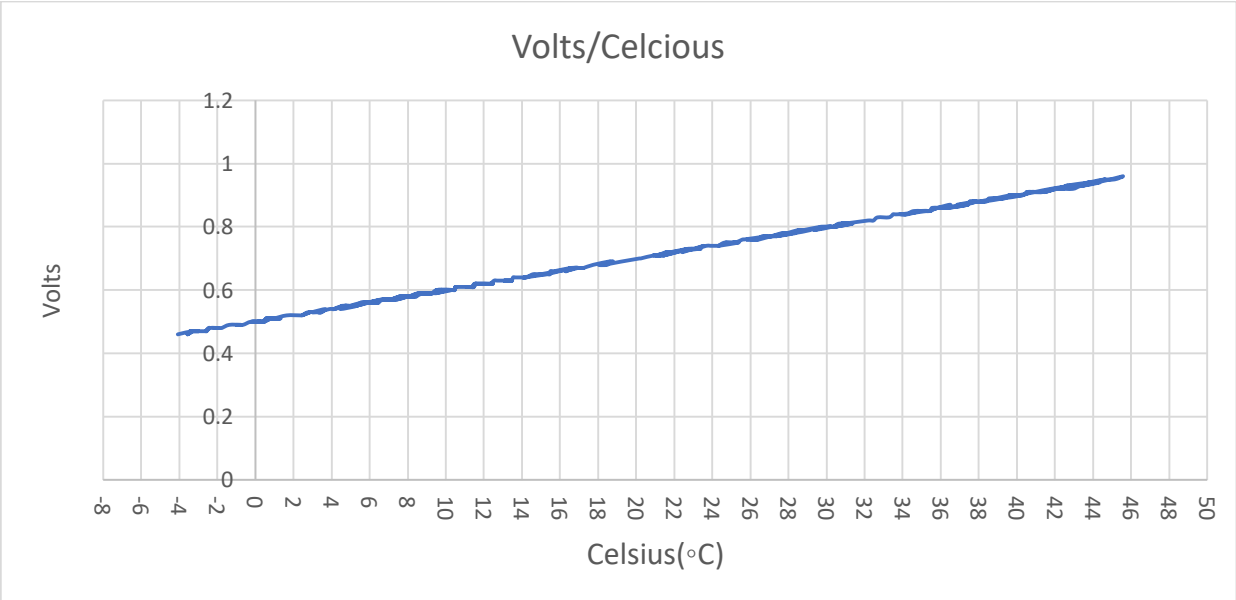


Σχήμα 5.2 : Λήψη θερμοκρασιών για κάθε 3 δευτερόλεπτα , που πραγματοποιήθηκε από τους αισθητήρες TMP36.

Ωστόσο , σημαντικό αποτελεί το γεγονός να επαληθευτεί η θεωρεία που ισχύει για το πλήθος μεταξύ θερμοκρασιών και τάσεων με το εύρος της βαθμονόμησης που επιτεύχθηκε. Επομένως , πραγματοποιείται μια σύγκριση μεταξύ του Πίνακα 5.3 και του Σχήματος 5.4.

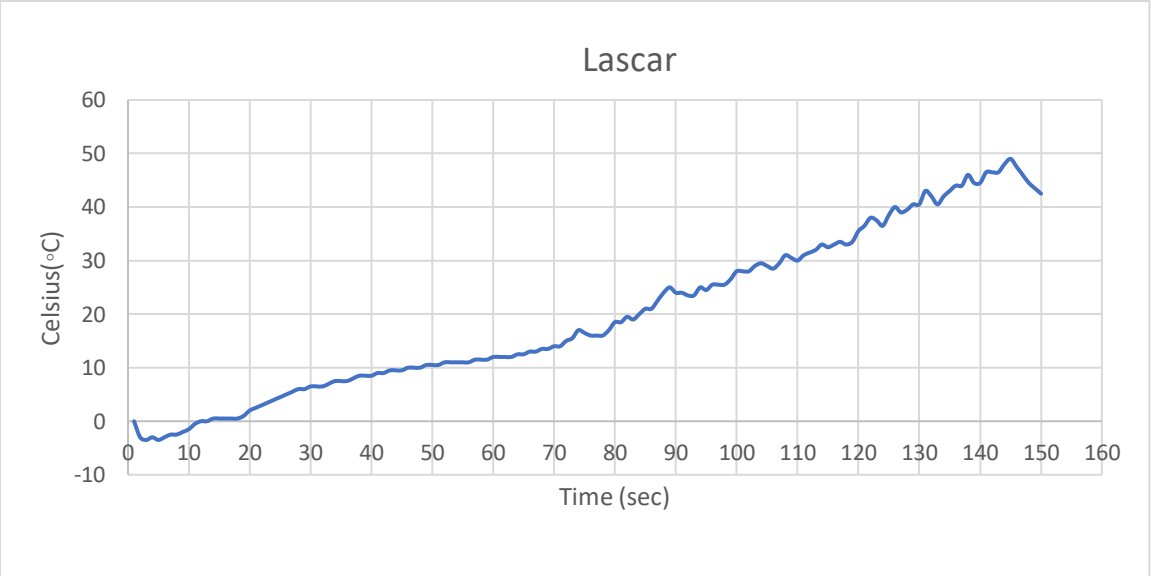
Βαθμοί Κελσίου	Τάσεις	Βαθμοί Κελσίου	Τάσεις
-5	0,45	20	0,70
-4	0,46	21	0,71
-3	0,47	22	0,72
-2	0,48	23	0,73
-1	0,49	24	0,74
0	0,50	25	0,75
1	0,51	26	0,76
2	0,52	27	0,77
3	0,53	28	0,78
4	0,54	29	0,79
5	0,55	30	0,80
6	0,56	31	0,81
7	0,57	32	0,82
8	0,58	33	0,83
9	0,59	34	0,84
10	0,60	35	0,85
11	0,61	36	0,86
12	0,62	37	0,87
13	0,63	38	0,88
14	0,64	39	0,89
15	0,65	40	0,90
16	0,66	41	0,91
17	0,67	42	0,92
18	0,68	43	0,93
19	0,69	44	0,94

Πίνακας 5.3 : Σχέση μεταξύ θερμοκρασιών και τάσεων , για το εύρος της βαθμονόμησης του πειράματος.



Σχήμα 5.4: Διάγραμμα μεταξύ βαθμών θερμοκρασίας και τάσεων κατά τη βαθμονόμηση.

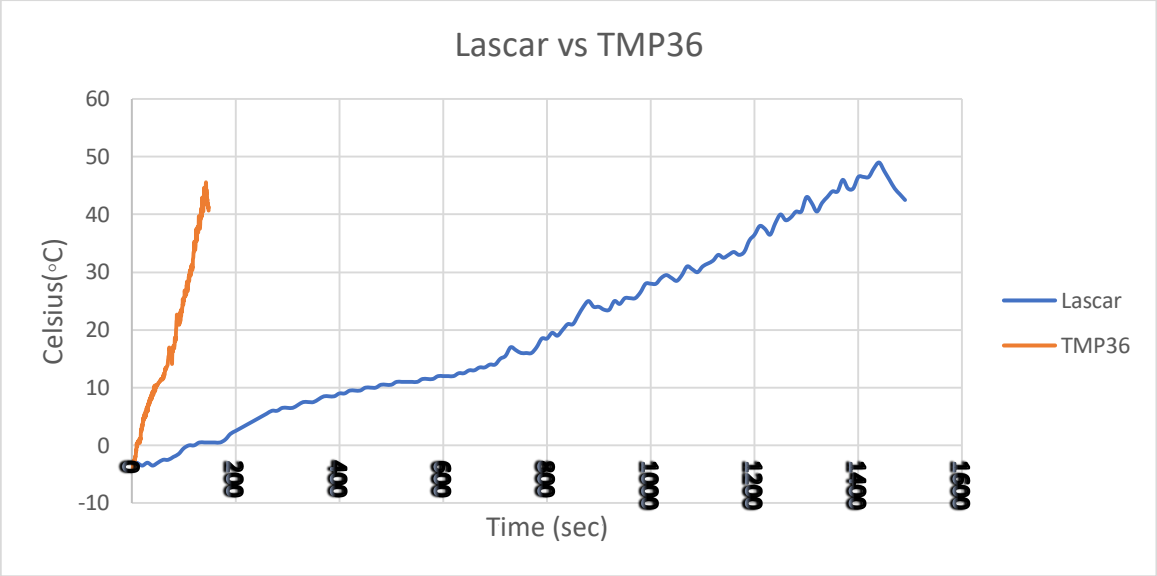
Συγκρίνοντας τη θεωρητική σχέση μεταξύ θερμοκρασιών και τάσεων με τα αποτελέσματα που πετύχαμε κατά τη λήψη 494^{ων} τιμών για κάθε τρία δευτερόλεπτα , παρατηρούμε ότι η σύγκριση αυτή εξακριβώνεται , δίχως κάποιες αποκλίσεις. Αυτό σημαίνει ότι οι αισθητήρες TMP36 λειτουργούν σωστά , ως προς το προγραμματισμό τους. Παρόλα αυτά, πρέπει να σιγουρευτούμε ότι οι τιμές αυτές είναι ορθές και ως προς το πρακτικό μέρος λειτουργείας τους και για το λόγο αυτό , παρουσιάζεται ένα νέο διάγραμμα το οποίο αποκαλύπτει τις μετρήσεις που πάρθηκαν από το θερμόμετρο της Lascar , για κάθε 10 δευτερόλεπτα κατά την ίδια χρονική στιγμή εκκίνησης με αυτή των αισθητήρων. Δυστυχώς, στο συγκεκριμένο πείραμα υπήρξε μια μικρή σχετική απόκλιση κατά τη ταυτόχρονη εκκίνηση των δύο αυτών συστημάτων , μια διαφορά 5 δευτερολέπτων. Το Σχήμα 5.5, αποτελεί ένα διάγραμμα θερμοκρασιών συναρτήση του χρόνου εκτέλεσης του κώδικα, για το σύστημα Lascar.



Σχήμα 5.5 : Διάγραμμα λήψης θερμοκρασιών για κάθε 10 δευτερόλεπτα , που πάρθηκαν από το σύστημα Lascar.

Αυτό που παρατηρεί κανείς , είναι ότι τα σχήματα των δύο συστημάτων μοιάζουν αρκετά μεταξύ τους. Αναλυτικότερα , για καλύτερη κατανόηση θα μπορούσαμε να ενώσουμε τα διαγράμματα τους και να συγκρίνουμε τα αποτελέσματα που προκύπτουν. Παρατηρώντας κανείς το Σχήμα 5.6 , αρχικά θα διαπιστώσει ότι τα συστήματα αυτά δεν έχουν κάποια ομοιότητα. Με μια δεύτερη ματιά όμως , ο μελετητής ίσως αλλάξει γνώμη. Αυτό που συμβαίνει είναι ότι στο σύστημα των αισθητήρων πάρθηκαν αρκετά περισσότερες τιμές από αυτό του συστήματος του Lascar. Στο πρώτο σύστημα , πραγματοποιήθηκε δειγματοληψία για 494 τιμές , ενώ στο δεύτερο οι τιμές που έχουν ληφθεί είναι μόλις 149. Αυτό βεβαίως οφείλεται στο γεγονός , ότι το πρώτο σύστημα έχει λάβει τιμές ανά 3

δευτερόλεπτα , ενώ το άλλο ανά 10. Επομένως , τα δύο αυτά διαγράμματα μπορούν θεωρηθούν μεταξύ τους ίδια , με το δεύτερο να παρουσιάζει πιο απότομες αυξομειώσεις , εξαιτίας των λιγότερων τιμών που διαθέτει.



Σχήμα 5.6 : Σύγκριση των δύο συστημάτων.

Σύμφωνα με τα παραπάνω , καταφέραμε να υλοποιήσουμε και να τρέξουμε το έργο μας αποδεικνύοντας ότι ο μετρήσεις μας είναι ακριβείς , χωρίς κάποιο περιθώριο σφάλματος. Το επόμενο κεφάλαιο , επικεντρώνεται κυρίως στο τρόπο με τον οποίο μπορείτε να ανεβάσετε και να μοιραστείτε το έργο σας , με διάφορους άλλους χρήστες. Ακόμα , σ' αυτό θα δείτε από που μπορείτε να παραλάβετε πλήρως το κώδικα , καθώς και άλλες πληροφορίες που αφορούν το έργο του συγγράμματός.

ΚΕΦΑΛΑΙΟ 6: ΔΙΕΠΑΦΗ ΜΕ ΑΛΛΟΥΣ ΧΡΗΣΤΕΣ ΜΕ ΤΗ ΒΟΗΘΕΙΑ ΤΟΥ GITHUB

6.1 Η ΕΝΝΟΙΑ ΤΟΥ GITHUB

Το κεφάλαιο αυτό που θα μελετήσουμε , αποτελεί και το τελευταίο του συγγράμματος. Πρόκειται για ένα εξίσου σημαντικό κεφάλαιο , καθώς θα δούμε πως μπορεί κάποιος να τοποθετήσει το έργο του σε μια ιστοσελίδα στο διαδίκτυο. Η ιστοσελίδα αυτή είναι αρκετά φημισμένη στο χώρο των προγραμματιστών και όχι μόνο. Πρόκειται για την Github για την οποία καλό θα ήταν να αναφέρουμε κάποια πράματα.

Όπως αναφέραμε προηγουμένως το Github πρόκειται για μια ιστοσελίδα , της οποίας το όνομα προκύπτει από την ένωση δύο λέξεων , Git και Hub [8]. Αυτές αποτελούν δύο διαφορετικά κομμάτια , με τη καθεμία να έχει το δικό της ρόλο. Επομένως , αξίζει να αναλύσουμε αυτές τις λέξεις που συγκροτούν την ευρύτερη έννοια του Github. Στη πρώτη περίπτωση , αν κάποιος προγραμματιστής θελήσει να αναθεωρήσει το κώδικά του , μπορεί να το πραγματοποιήσει όσες φορές εκείνος θέλει. Πως μπορεί να γίνει όμως κάτι τέτοιο ; Πολύ απλά η έννοια του Git προέκυψε από τον κατασκευαστή των Linux , με σκοπό την αναβάθμιση των εκδόσεων για κάποιο πρόγραμμα ανοικτού κώδικα , κάθε φορά που ο χρήστης χρειάζεται να αλλάξει κάτι. Ωστόσο το έργο του χρήστη μπορεί να το πάρει και οποιοσδήποτε άλλος ο προγραμματιστής με ευκολία να το επεξεργαστεί και να το ανεβάσει πάλι στη καινούρια του μορφή. Εκεί κολλάει παράλληλα και η έννοια του Hub. Καθώς οι χρήστες ανεβάζουν ένα έργο πρέπει να το αποθηκεύσουν , όπου τη δουλειά αυτή έρχεται να την πραγματοποιήσει η λειτουργεία Hub. Άρα , με τη βοήθεια του Github επιτυγχάνεται η διεπαφή πολλών χρηστών που μπορούν να λάβουν , να επεξεργαστούν και να αναθεωρήσουν το ίδιο έργο κάθε φορά που αυτοί κρίνουν αναγκαίο.

6.2 ΕΓΓΡΑΦΗ ΚΑΙ ΑΝΕΒΑΣΜΑ ΚΩΔΙΚΑ ΣΤΟ GITHUB

Στη προηγούμενη ενότητα αναλύσαμε την έννοια του Github και αναφερθήκαμε στη χρησιμότητα του. Εδώ θα δούμε πως μπορεί κανείς να δημιουργήσει δικό του λογαριασμό, έτσι ώστε να μπορέσει να ανεβάσει το έργο του επιτυχώς και να είναι ορατό στους άλλους χρήστες. Αρχικά το VS Code δίνει τη δυνατότητα στο προγραμματιστή να αποθηκεύσει το κώδικά του στο προσωπικό του One Drive , ανοίγοντας το File και επιλέγοντας το Save As. Μετά από αυτή τη διαδικασία μέσα στο φάκελο του One Drive , δημιουργήθηκαν διάφοροι φάκελοι που περιλαμβάνουν όλες τις καρτέλες , τους κώδικες και τις βιβλιοθήκες

του έργου. Επομένως , ακολουθούμε προσεκτικά όλα τα απαραίτητα βήματα για τη συμμετοχή και τη προώθηση του έργου στον ιστότοπο :

1) Αρχικά , ανοίγουμε ένα browser μια μηχανή αναζήτησης στο διαδίκτυο.
2) Πληκτρολογούμε www.github.com.
3) Μας μεταφέρει στην αρχική του Github και εκεί πάνω δεξιά αναγράφεται η λέξη sign up , όπου πατάμε πάνω σε αυτό.
4) Στη πορεία ζητάει να βάλουμε κάποιο email και να τοποθετήσουμε διάφορα άλλα προσωπικά μας στοιχεία.
5) Μετά από το βήμα 4 έχουμε φτιάξει ένα λογαριασμό και μπορούμε να πάμε πάνω δεξιά πάλι, εκεί όπου αυτή τη φορά θα επιλέξουμε το sign in και θα τοποθετήσουμε το email και το κωδικό όπου έχουμε ορίσει.
6) Η καινούρια σελίδα που μας μεταφέρει στα δεξιά διαθέτει τρία σύμβολα , αυτά της πρόσθεσης , ένα καμπανάκι καθώς και ένα εικονίδιο το οποίο είναι ο λογαριασμός που δημιουργήσαμε και μπορούμε να κάνουμε διάφορες ενέργειες. Εμείς επιλέγουμε από το εικονίδιο του λογαριασμού μας τη δεύτερη επιλογή που αναγράφει "your repositories".
7) Έπειτα , μεταφερόμαστε σε μια νέα σελίδα στην οποία πατάμε πάνω στο κουμπί "new".
8) Ο νέος ιστότοπος αφορά το έργο μας καθώς εκεί ορίζουμε τίτλο , περιγραφή , αν επιθυμούμε να ορατό ή όχι ως προς τους άλλους χρήστες. Ακόμα μπορούμε να προσθέσουμε τη γλώσσα του προγραμματισμού καθώς και την άδεια που διαθέτει η πλακέτα μας. Να υπενθυμίσουμε ότι στο πείραμα που υλοποιήθηκε με τον Nrf η πλακέτα αυτή διαθέτει άδεια MIT , όπως ειπώθηκε στο πρώτο κεφάλαιο.
9) Αφού συμπληρώσαμε όλα τα κενά , πατάμε πάνω στο "create repository".
10) Η νέα σελίδα διαθέτει τρεις επιλογές όπου αυτές είναι "Go to file" , "Add file" και "Code". Εμείς επιλέγουμε τη δεύτερη και πατάμε στο "Upload files".
11) Στην ιστοσελίδα αυτή που προκύπτει μπορούμε να σύρουμε όλα τα αρχεία που έχουμε αποθηκευμένα στο One Drive του Η/Υ.
12) Τέλος πατάμε στο "commit changes" , και έχουμε καταφέρει να δημοσιεύσουμε το έργο μας ως προς τους υπόλοιπους χρήστες.

Υπάρχουν και άλλοι τρόποι για να μπορέσει κάποιος να μοιραστεί το έργο του με τους υπολοίπους χρήστες , ωστόσο στη συγγραφή προτιμήθηκε ο πιο απλός και κατανοητός , σύμφωνα με την άποψη του συγγραφέα.

Η διεπαφή με άλλους χρήστες διευκολύνει τη ζωή των προγραμματιστών που έχουν την ευκαιρία να πάρουν πληροφορίες για ένα έργο που πρόκειται να υλοποιήσουν ή να το βρουν διαθέσιμο έτσι όπως ακριβώς αυτοί το αναζητούν. Για το λόγο αυτό , προτιμήθηκε η κοινοποίηση του πειράματός στο Github , για οποιονδήποτε χρειαστεί να αντλήσει δεδομένα για το πως θα το πραγματοποιήσει. Όλες αυτές οι πληροφορίες του έργου είναι διαθέσιμες στην ιστοσελίδα https://github.com/fotiskargakis/Nrf52840-DK-TMP36-sensors και μπορείτε να βρείτε όλα τα δεδομένα που έχουν αναφερθεί κατά την επέκταση της συγγραφής , όπως είναι οι κώδικες , οι βιβλιοθήκες μέχρι και τα αρχεία , που περιλαμβάνουν τη βαθμονόμηση των θερμοκρασιών.

ΠΑΡΑΡΤΗΜΑ Α’

ΕΠΙΛΟΓΟΣ

Κάπου εδώ πλησιάζουμε στο τέλος της συγγραφής. Ο αναγνώστης ακολουθώντας όλες τις απαραίτητες οδηγίες και βήματα , είναι σε θέση να υλοποιήσει το πείραμα , να ελέγξει τα αποτελέσματα που έλαβε από αυτό καθώς και να το μοιραστεί με τους υπολοίπους χρήστες. Ουσιαστικά είδαμε πως μπορούμε να πάρουμε από τους αναλογικούς αισθητήρες TMP36 μετρήσεις σε ψηφιακή μορφή , να παρατηρήσουμε τη τάση για κάθε αναλογικό κανάλι που εμείς διαλέξαμε από τον ελεγκτή NRF , για συγκεκριμένες θερμοκρασίες , καθώς και την απαραίτητη συνδεσμολογία για την υλοποίηση του κυκλώματος. Ακόμα , πετύχαμε για ένα μεγάλο εύρος τιμών να εξακριβώσουμε ότι τα αποτελέσματα μας είναι σωστά , ενώ περιγράφεται και η κοινοποίηση του έργου , ως προς τρίτους που ενδιαφέρονται για το συγκεκριμένο έργο ή χρειάζεται να αντλήσουν πληροφορίες από αυτό. Σημειώνεται ότι η βαθμονόμηση μεταξύ θερμοκρασιών και τάσεων , επιτεύχθηκε για ένα εύρος που κατασκευάστηκε σύμφωνα με θερμοκρασίες οι οποίες είναι πιο εφικτές σε ένα περιβάλλον δωματίου , ενώ κάποιος άλλος θα μπορούσε να ορίσει ένα εύρος τιμών για περισσότερο ακραίες θερμοκρασίες. Ακολουθούν οι πηγές στις οποίες βασίστηκε η γραφή του συγγράμματος.

ΠΑΡΑΡΤΗΜΑ Β’

ΠΗΓΕΣ

[1] Nrf52840-DK Datasheet, *www.nordicsemi.com/Products/Development-hardware/nrf52840-dk*

[2] Nordic Semiconductor, *www.craft.co/nordic-semiconductor*

[3] Nordic NVLSI, *www.design-reuse.com/news/7860/nordic-vlsi-asa-becomes-nordic-semiconductor-asa.html*

[4] Nrf52 ADC, *www.learn.adafruit.com/introducing-theadafruit-nrf52840-feather/nrf52-adc*

[5] TMP36 Datasheet, *www.analog.com/media/en/technical-documetation/data-sheets/TMP35_36_37.pdf*

[6] Visual Studio Code, *www.code.visualstudio.com/learn*

[7] Baud Rate Meaning, *www.learn.sparkfun.com/tutorials/serial-communication/rules-of-serial*

[8] GitHub Meaning, *www.el.omatomeloanhikaku.com/htg-explains-what-is-github-and-what-do-geeks-use-it-for-9447*