

Εκπαιδευτική Ρομποτική και μετεωρολογία με Raspberry Pi και Python



Φίλιππος Κουτσάκας
Εκπαιδευτικός Πληροφορικής

Πίνακας Περιεχομένων

1.	Τίτλος Σεναρίου.	4
2.	Εκτιμώμενη διάρκεια.	4
3.	Ένταξη του Διδακτικού Σεναρίου στο πρόγραμμα σπουδών και Προαπαιτούμενες Γνώσεις.	4
4.	Σκοποί και στόχοι του διδακτικού σεναρίου	4
4.1	Στο πεδίο των γνώσεων	4
4.2	Στο πεδίο των ικανοτήτων	4
4.3	Στο πεδίο των στάσεων	5
5.	Συνοπτική περιγραφή του διδακτικού σεναρίου	5
6.	Επιστημολογική προσέγγιση και εννοιολογική ανάλυση – Θέματα θεωρίας του διδακτικού σεναρίου	5
7.	Επεκτάσεις/διασυνδέσεις των εννοιών	6
8.	Πολλαπλές αναπαραστάσεις – πολλαπλές προσεγγίσεις	7
9.	Πρόβλεψη δυσκολιών στο διδακτικό σενάριο	8
10.	Γιατί να χρησιμοποιηθεί ο υπολογιστής;	8
11.	Διδακτικός θόρυβος	8
12.	Χρήση εξωτερικών πηγών	8
13.	Υποκείμενη θεωρία μάθησης	9
14.	Επισήμανση μικρομεταβολών στην οργάνωση του μαθήματος και στο νόημα των εννοιών	9
15.	Διδακτικό συμβόλαιο	9
16.	Οργάνωση τάξης – εφικτότητα σχεδίασης	10
17.	Περιγραφή και ανάλυση των φύλλων εργασίας (ή και άλλου διδακτικού υλικού) – Αξιολόγηση	10
17.1	Φύλλο Εργασίας 1	10
17.2	Φύλλο Εργασίας 2	10
17.3	Φύλλο Εργασίας 3	10
17.4	Φύλλο Εργασίας 4	11
17.5	Φύλλο Εργασίας 5	11
17.6	Φύλλο Εργασίας 6	11
17.7	Φύλλο Εργασίας 7	11

18.	Φύλλα εργασίας	12
18.1	Φύλλο Εργασίας 1: Γνωριμία με τον υπολογιστή μικρού μεγέθους Raspberry Pi	12
18.2	Φύλλο Εργασίας 2: Αναγνώριση των εξαρτημάτων του μετεωρολογικού σταθμού	13
18.3	Φύλλο Εργασίας 3: Συναρμολόγηση του Μετεωρολογικού σταθμού	16
18.4	Φύλλο Εργασίας 4: Εγκατάσταση Λειτουργικού Συστήματος στο Raspberry Pi	21
18.5	Φύλλο Εργασίας 5: Γνωριμία με τις βιβλιοθήκες της Python	24
18.6	Φύλλο Εργασίας 6: Εγκατάσταση των απαραίτητων βιβλιοθηκών για την λειτουργία και διαχείριση των αισθητήρων του μετεωρολογικού σταθμού.	25
18.7	Φύλλο Εργασίας 7: Βήμα-βήμα μελέτη του κώδικα σε Python.	29
18.7.1	Βήμα 1: Δημιουργία του βασικού πλαισίου του προγράμματος μας	30
18.7.2	Βήμα 2: Ανάγνωση της θερμοκρασίας από τον αισθητήρα DS18B20	31
18.7.3	Βήμα 3: Ανάγνωση του αισθητήρα BME280	32
18.7.4	Βήμα 4: Ανάγνωση αισθητήρα ADC (κατεύθυνση ανέμου)	34
18.7.5	Βήμα 5: Ανάγνωση Αισθητήρα Ταχύτητας Ανέμου	37
18.7.6	Βήμα 6: Ανάγνωση αισθητήρα βροχόπτωσης	42
18.7.7	Βήμα 7: Όλα έτοιμα!!!	47
19.	Βιβλιογραφικές Αναφορές	48

1. Τίτλος Σεναρίου.

Εκπαιδευτική Ρομποτική και μετεωρολογία με Raspberry Pi και Python.

2. Εκτιμώμενη διάρκεια.

8-10 διδακτικές ώρες.

3. Ένταξη του Διδακτικού Σεναρίου στο πρόγραμμα σπουδών και Προαπαιτούμενες Γνώσεις.

Το σενάριο προορίζεται για μαθητές των Β' και Γ' τάξεων Επαγγελματικών Λυκείων.

Το σενάριο κατ' αρχήν προορίζεται για τους μαθητές του Τομέα Πληροφορικής για υλοποιηθεί και να επεκτείνει τις γνώσεις των μαθητών στα πλαίσια του μαθήματος «Αρχές Προγραμματισμού Υπολογιστών»

<http://ebooks.edu.gr/new/books-pdf.php?course=DSEPAL-B133> της Β' τάξης καθώς και του μαθήματος «Προγραμματισμός Υπολογιστών»

<http://ebooks.edu.gr/new/books-pdf.php?course=DSEPAL-C219>

Επιπλέον το σενάριο θα μπορούσε να αξιοποιηθεί στα πλαίσια του μαθήματος γενικής παιδείας «Εισαγωγή στις Αρχές της Επιστήμης των ΗΥ» που διδάσκεται στη Β' τάξη όλων των τομέων των Επαγγελματικών Λυκείων.

Τέλος το σενάριο θα μπορούσε να αξιοποιηθεί στα πλαίσια projects και δράσεων προγραμματισμού στον φυσικό κόσμο (physical computing).

4. Σκοποί και στόχοι του διδακτικού σεναρίου

Σκοπός του σεναρίου είναι οι μαθητές να εξοικειωθούν με τον προγραμματισμό στον φυσικό κόσμο (physical computing) με Raspberry Pi και Python και να μπορούν να δημιουργούν τα δικά τους έργα.

Οι ειδικότερα οι μαθητές μετά το τέλος της διδασκαλίας θα πρέπει να είναι σε θέση:

4.1 Στο πεδίο των γνώσεων

- Να διακρίνουν τα βασικά εξαρτήματα του μικροϋπολογιστή Raspberry Pi
- Να αναγνωρίζουν αισθητήρες μέτρησης καιρικών φαινομένων (θερμοκρασία, υγρασία, ατμοσφαιρική πίεση, ταχύτητα και κατεύθυνση ανέμου, ύψος βροχόπτωσης, κλπ)
- Να αναγνωρίζουν έννοιες όπως θερμοκρασία, υγρασία, ατμοσφαιρική πίεση, ταχύτητα και κατεύθυνση ανέμου, ύψος βροχόπτωσης, κλπ)
- Να αναλύουν ένα σύνθετο πρόβλημα σε απλούστερα.
- Να συνθέτουν τις απλές λύσεις για να δημιουργήσουν ένα έργο.
- Να προγραμματίζουν σε Python.
- Να διορθώνουν λογικά λάθη κατά τη δημιουργία των προγραμμάτων τους.

4.2 Στο πεδίο των ικανοτήτων

- Να δημιουργούν κυκλώματα και κατάλληλες συνδέσεις ηλεκτρονικών εξαρτημάτων και αισθητήρων
- Να εγκαθιστούν ΕΛ/ΛΑΚ λειτουργικά συστήματα και σε ανοιχτό υλικό
- Να δημιουργούν τα δικά τους project
- Να εγκαθιστούν βιβλιοθήκες της γλώσσας Python για τη διαχείριση αισθητήρων

4.3 Στο πεδίο των στάσεων

- Να εκτιμούν τη χρήση Η/Υ για τη λύση προβλημάτων της καθημερινότητας.
- Να υιοθετούν καλές πρακτικές για τον προγραμματισμό.
- Να συνεργάζονται σε ομάδες.

5. Συνοπτική περιγραφή του διδακτικού σεναρίου

Αρχικά ο εκπαιδευτικός υλοποιεί δραστηριότητες ψυχολογικής και γνωστικής προετοιμασίας. Γίνεται διάλογος για το Physical Computing, το Raspberry Pi την Python και τι μπορούν οι μαθητές να δημιουργήσουν. Ο εκπαιδευτικός μπορεί να δείξει στους μαθητές κάποια εκπαιδευτικά ενημερωτικά βίντεο σχετικά με το Raspberry Pi καθώς και αντιπροσωπευτικά projects που έχουν υλοποιηθεί στην πλατφόρμα αυτή.

Ενδεικτικά:

- Ξεκινώντας με το Raspberry Pi: <https://projects.raspberrypi.org/el-GR/projects/raspberry-pi-getting-started>
- Χρήση το Raspberry Pi: <https://projects.raspberrypi.org/el-GR/projects/raspberry-pi-using>
- What is a Raspberry Pi: <https://www.youtube.com/watch?v=uXUjwk2-qx4>
- Get started with Raspberry Pi 4: <https://www.youtube.com/watch?v=BpJCAafw2qE>
- Raspberry Pi Projects 2020: <https://www.youtube.com/watch?v=ZDfhcA0SCiM>
- Top 10 RPi Projects for beginners 2020: <https://www.youtube.com/watch?v=etJom0S5g8k>

Στη συνέχεια γίνεται μια αναφορά στην γλώσσα προγραμματισμού Python με απλά παραδείγματα. Οι μαθητές είναι εξοικειωμένοι με την γλώσσα προγραμματισμού δεδομένοι ότι την διδάσκονται τόσο ως μάθημα γενικής παιδείας (Β' ΕΠΑΛ όσο και ως μάθημα ειδικότητας). Οι μαθητές συμμετέχουν αναφέροντας δικά τους παραδείγματα προγραμμάτων σε Python.

Στις επόμενες διδακτικές ώρες υλοποιούνται οι δραστηριότητες των φύλλων εργασίας.

6. Επιστημολογική προσέγγιση και εννοιολογική ανάλυση – Θέματα θεωρίας του διδακτικού σεναρίου

Είναι γενικά αποδεκτό ότι στις σύγχρονες κοινωνίες της δημιουργικότητας η ενσωμάτωση των ΤΠΕ σε όλους, σχεδόν, τους τομείς της καθημερινής πραγματικότητας γίνεται με γοργούς ρυθμούς. Όμοια και στον χώρο της εκπαίδευσης εισάγονται καινοτομίες που μεταβάλλουν το παλαιό διδακτικό μοντέλο (Resnick, 2002).

Τα τελευταία χρόνια η εκπαιδευτική ρομποτική ως εργαλείο μάθησης αποτελεί επιλογή όλο και περισσότερων εκπαιδευτικών για την βιωματική προσέγγιση εννοιών σε διαφορετικά

γνωστικά αντικείμενα προσφέροντας εξαιρετικά πλεονεκτήματα και οφέλη στην μαθησιακή πορεία για την οικοδόμηση της γνώσης (Alimis, 2009). Ο Sullivan (2008) και οι Afari & Khine (2017) υποστηρίζουν ότι διδακτικά σενάρια στα οποία εμπλέκεται η ρομποτική δίνει την δυνατότητα για την ανάπτυξη γνωστικών και μαθησιακών δεξιοτήτων (μεταγνωστικές διεργασίες). Έρευνες από τους William et al.(2007) δείχνουν ότι οι ρομποτικές πλατφόρμες μέσα σε από ένα καθοδηγούμενο περιβάλλον μάθησης στη τάξη στο πλαίσιο της φιλοσοφίας της επίλυσης προβλημάτων προσφέρει την ευκαιρία στους μαθητές/τριες να συνδέσουν τις εμπειρίες τους με επιστημονικές έννοιες.

Εκτός, όμως, από την οικοδόμηση της γνώσης στις φυσικές επιστήμες και την τεχνολογία στη βάση μιας STEM φιλοσοφίας, η χρήση της εκπαιδευτικής ρομποτικής αναπτύσσει ένα ευρύ φάσμα δεξιοτήτων σε διάφορα γνωστικά αντικείμενα. Σε έρευνες των McDonald et al (2012) επισημαίνεται η ανάπτυξη γλωσσικών δεξιοτήτων εκτός από την υπολογιστική σκέψη με χρήση ρομποτικών εφαρμογών, ενώ σε αρκετές έρευνες στη βιβλιογραφία η εφαρμογή της ρομποτικής στα σχολεία διευρύνεται συνεχώς συμπεριλαμβάνοντας και τα λεγόμενα non-STEM γνωστικά αντικείμενα στο πλαίσιο των κοινωνικών και ανθρωπιστικών επιστημών.

Η θεωρητική βάση όλων των προσεγγίσεων με τη χρήση του εργαλείου της ρομποτικής αποτελεί η ιδέα του Κονστρουκτιβισμού (Constructionism) του S. Papert στηριζόμενος στις θεωρίες μάθησης του Κονστρουκτιβισμού (Constructivism) του J. Piaget (Papert, 1980, Kafai et al 1996). Αν και ο όρος Κονστρουκτιβισμός περιέχει τη έννοια της κατασκευής δεν αναφέρεται σε κατασκευάσματα αλλά στην ενεργητική διαδικασία δόμησης της γνώσης. Ο S. Papert και οι συνεργάτες του εισήγαγαν την αντίληψη της χρήσης κατασκευών μέσα από Hands-on δραστηριότητες προς μια ενεργητική μάθηση με νόημα (meaningful).

Η χρήση ρομποτικών διατάξεων εξυπηρετεί ακριβώς αυτό το σκοπό καθιστώντας στα μάτια των μικρών μαθητών την μαθησιακή διαδικασία διασκεδαστική (EDUTAINMENT, Johnson 2003) μέσα σε ένα εργαστήριο αυτενέργειας όπου η γνώση κατακτάται μέσα από την δράση σύμφωνα και με το J. Dewey. Η εκπαιδευτική ρομποτική, όμως, δεν αποτελεί ως μαθησιακό εργαλείο από μόνο του το κλειδί της καινοτομίας. Σύμφωνα με τον Alimisis (2012) δεν είναι τόσο σημαντικό ποια ακριβώς ρομποτική πλατφόρμα θα χρησιμοποιηθεί σε μια τάξη αλλά η προσοχή πρέπει να εστιάζεται στη διδακτική μέθοδο που θα ακολουθηθεί. Η ιδέα του Papert για τη μάθηση μέσα από τη προγράμματα της LOGO είχαν ανακαλυπτικό χαρακτήρα. Η ανακαλυπτική μάθηση αργότερα μετεξελίχθηκε σε καθοδηγούμενη ανακάλυψη καθώς και στο μοντέλο της διερευνητικής μάθησης η οποία στις φυσικές επιστήμες αποτελεί μια μοντέρνα στρατηγική που βελτιώνει τις ερευνητικές δεξιότητες των μαθητών σύμφωνα με έρευνες των Alti and Pedaste (2013). Από την άλλη η μέθοδος της επίλυσης προβλημάτων που και αυτή αποτελεί επαγωγική διδακτική μέθοδο και το μοντέλο της ανταγωνιστικής (competition-based learning) μάθησης αποτελούν στο τομέα της εκπαιδευτικής ρομποτικής τις πιο διαδεδομένες και επαρκείς μεθόδους χρήσης ρομποτικών κατασκευών στα μαθηματικά, φυσική και άλλα γνωστικά αντικείμενα (Altin et al 2013).

Υπάρχουν πολλές πλατφόρμες που μπορούν να χρησιμοποιηθούν όπως Arduino (<https://www.arduino.cc/>), Lego Mindstorms (<https://www.lego.com/en-us/mindstorms/>), Raspberry Pi (<https://www.raspberrypi.org/>), BBC Microbit (<https://microbit.org/>) και άλλες.

7. Επεκτάσεις/διασυνδέσεις των εννοιών

Μετά την ολοκλήρωση του σεναρίου ο διδάσκων θα μπορούσε να συνεχίσει προετοιμάζοντας την τοποθέτηση του μετεωρολογικού σταθμού σε εξωτερικό χώρο (παραμετροποίηση του λειτουργικού συστήματος ώστε να εκκινεί αυτόματα μετά από διακοπές ρεύματος και να εκκινεί την εφαρμογή του μετεωρολογικού σταθμού αυτόματα, κατασκευή στεγανού κουτιού, κατασκευή ιστού για τοποθέτηση, προετοιμασία παροχής ρεύματος, κλπ). Περισσότερες πληροφορίες σχετικά με αυτή την επέκταση του σεναρίου θα βρείτε εδώ: <https://www.bc-robotics.com/tutorials/raspberry-pi-weather-station-part-4/>

Επιπλέον ο διδάσκων θα μπορούσε να επεκτείνει το σενάριο συνδέοντας το πρόγραμμα διαχείρισης του μετεωρολογικού σταθμού με την ιστοσελίδα ThinkSpeak (<https://thingspeak.com/>) η οποία δίνει τη δυνατότητα προβολής των δεδομένων αξιοποιώντας έτοιμα πρότυπα γραφικών παραστάσεων (δες την εικόνα που ακολουθεί).

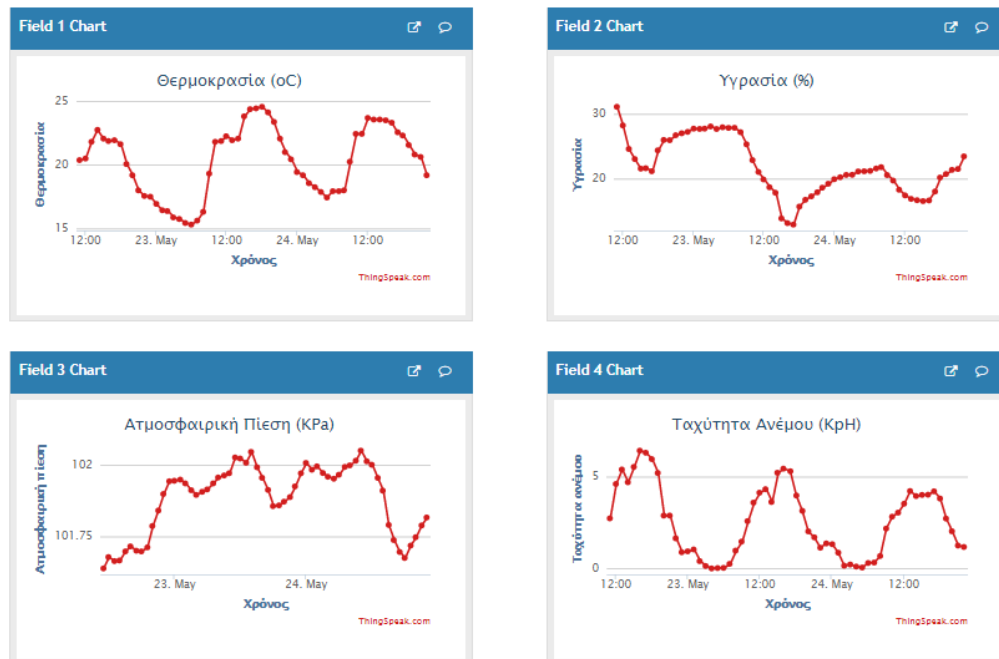


Figure 1. Γραφική αναπαράσταση τιμών αξιοποιώντας το εργαλείο ThinkSpeak

Πληροφορίες σχετικά με την υλοποίηση αυτής της δράσης μπορείτε να βρείτε στον σύνδεσμο <https://www.bc-robotics.com/tutorials/raspberry-pi-weather-station-part-3/>

Τέλος ο διδάσκων θα μπορούσε να συνεχίσει το παρόν σενάριο επιλέγοντας από μία τεράστια γκάμα διαθέσιμων projects της κοινότητας του Raspberry Pi με χρήση άλλων αισθητήρων ή κινητήρων (<https://projects.raspberrypi.org/>)

8. Πολλαπλές αναπαραστάσεις – πολλαπλές προσεγγίσεις

Το σενάριο προτείνεται να υλοποιηθεί αξιοποιώντας το Raspberry Pi 3A+ ή το 4 και τους αισθητήρες θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας και κατεύθυνσης ανέμου. Ο παραπάνω εξοπλισμός θα ενοποιηθεί αξιοποιώντας το Raspberry Pi Weather

Board της BC Robotics (<https://www.bc-robotics.com/shop/raspberry-pi-weather-board/>) και τη διαχείριση των μετρήσεων θα την έχει μία εφαρμογή που θα υλοποιηθεί σε Python.

Εναλλακτικά ο μετεωρολογικός σταθμός θα μπορούσε να υλοποιηθεί αξιοποιώντας την πλατφόρμα BBC Microbit. Ενδεικτικές υλοποιήσεις μπορείτε να βρείτε στους παρακάτω συνδέσμους:

- <https://www.instructables.com/id/MicroBit-Weather-Station/>
- <https://www.newscientist.com/article/mg24332450-900-make-a-weather-station-with-bbc-microbits-and-a-sensor/>
- <https://www.bbc.co.uk/programmes/articles/4fgQITdij5g8lth67xRgZl/storm-watching>

9. Πρόβλεψη δυσκολιών στο διδακτικό σενάριο

Οι μαθητές για πρώτη φορά θα υλοποιήσουν συγκολλήσεις οπότε θα χρειαστούν τη βοήθεια των διδασκόντων.

Οι δυσκολίες στον προγραμματισμό με Python εξαρτώνται από την εμπειρία που έχουν οι μαθητές στον προγραμματισμό. Δεδομένου όμως ότι οι μαθητές του Τομέα Πληροφορικής των ΕΠΑΛ διδάσκονται Python στην Β και Γ λυκείου σε συνδυασμό με το ότι ο κώδικας παρουσιάζεται βαθμιαία και πλήθος επεξηγήσεων, δεν αναμένεται να αντιμετωπίσουν ιδιαίτερες δυσκολίες στην κατανόηση του κώδικα.

Γενικότερα δεν αναμένονται ιδιαίτερες δυσκολίες κατά την εκτέλεση του σεναρίου.

10. Γιατί να χρησιμοποιηθεί ο υπολογιστής;

Το Physical Computing είναι άμεσα συνδεδεμένο με τους υπολογιστές αφού η βασική ιδέα είναι ένα πρόβλημα του φυσικού κόσμου (ερέθισμα) να λυθεί μέσω υπολογιστή (αντίδραση). Η δυνατότητα να βλέπουν οι μαθητές άμεσα το έργο που δημιουργήσαν, να μπορούν να το τροποποιήσουν και να πειραματιστούν, καθώς και η ποικιλία αντικειμένων που μπορούν να χρησιμοποιήσουν δίνουν διδακτικά πλεονεκτήματα. Η δημιουργικότητα και η φαντασία των μαθητών απελευθερώνονται και μπορούν να συμμετέχουν ενεργά στην οικοδόμηση της γνώσης.

11. Διδακτικός θόρυβος

Δεν αναμένεται διδακτικός θόρυβος πέρα από τον συνήθη όταν οι μαθητές συνεργάζονται.

12. Χρήση εξωτερικών πηγών

Ξεκινώντας με το Raspberry Pi:

<https://projects.raspberrypi.org/el-GR/projects/raspberry-pi-getting-started>

Χρήση το Raspberry Pi: <https://projects.raspberrypi.org/el-GR/projects/raspberry-pi-using>

What is a Raspberry Pi: <https://www.youtube.com/watch?v=uXUjwk2-qx4>

Get started with Raspberry Pi 4: <https://www.youtube.com/watch?v=BpJCAafw2qE>

Raspberry Pi Projects 2020: <https://www.youtube.com/watch?v=ZDfhcA0SCiM>

Top 10 RPi Projects for beginners 2020: <https://www.youtube.com/watch?v=etJom0S5g8k>

Build a Raspberry Pi Weather Station (Parts 1 – 4), BC Robotics Tutorials

- <https://www.bc-robotics.com/tutorials/raspberry-pi-weather-station-part-1/>
- <https://www.bc-robotics.com/tutorials/raspberry-pi-weather-station-part-2/>
- <https://www.bc-robotics.com/tutorials/raspberry-pi-weather-station-part-3/>
- <https://www.bc-robotics.com/tutorials/raspberry-pi-weather-station-part-4/>

Soldering is Easy. Here's how to do it.

https://mightyohm.com/files/soldercomic/FullSolderComic_EN.pdf

Σενάριο Physician Computing με Raspberry Pi και Python., Εκπαίδευση Επιμορφωτών Β' Επιπέδου ΤΠΕ, Ενότητα 7, Ειδικά θέματα Ρομποτικής, Επιμορφωτικό Υλικό: Ειδικός Μέρος, Ιούλιος 2018.

13. Υποκείμενη θεωρία μάθησης

Το σενάριο βασίζεται στο εποικοδομιστικό μοντέλο, το οποίο έχει ως βασική διδακτική παραδοχή ότι οι γνώσεις δεν μεταδίδονται απλά με τη διδασκαλία, αλλά "οικοδομούνται" και αναδομούνται από τον μαθητή ο οποίος είναι υπεύθυνος για τη μάθησή του. Οι μαθητές ωθούνται να πειραματιστούν και να οικοδομήσουν τη γνώση τους ανιχνεύοντας, διερευνώντας και αλληλεπιδρώντας μεταξύ τους, με το υλικό και με το λογισμικό.

Συνοπτικά θα πούμε ότι οι μαθησιακές δραστηριότητες του παρόντος σεναρίου σχεδιάστηκαν ακολουθώντας τη φιλοσοφία της διερευνητικής μάθησης (inquiry learning) και της συνεργατικής μάθησης (collaborative learning).

Ένα μέρος των φύλλων εργασίας είναι σε μορφή tutorial, γιατί σκοπός τους είναι, οι μαθητές να εξοικειωθούν με το Raspberry Pi, το λειτουργικό του σύστημα και τις βασικές τους λειτουργίες. Οι προτεινόμενες δραστηριότητες είναι κλιμακωτής δυσκολίας φύλλα εργασίας - διερευνητικού τύπου μέσα από τις οποίες στήνεται βήμα-βήμα ένας μετεωρολογικός σταθμός βασισμένος στο υπολογιστή μικρού μεγέθους Raspberry Pi. Η διδακτική προσέγγιση στηρίζεται κατά κύριο λόγο στην ενεργητική συμμετοχή των μαθητών και στον πειραματισμό.

Ο ρόλος του εκπαιδευτικού είναι καθοδηγητικός, διαμεσολαβητικός, διευκολυντικός. Αξιοποιεί τις δυνατότητες των ΤΠΕ για να σχεδιάσει, να οργανώσει, να επεξεργαστεί και να αναλύσει τα δεδομένα του εξεταζόμενου θέματος μαζί με τους μαθητές του, στο πλαίσιο μιας ενεργητικής και διδακτικής διαδικασίας με στόχο την κατανόηση της εξεταζόμενης κατάστασης. Η επικοινωνία εκπαιδευτικού - μαθητών δεν εξυπηρετεί τη «μεταφορά» γνώσεων μέσω της τεχνολογίας από τον εκπαιδευτικό στον μαθητή, αλλά επιτρέπει τη γνωστική επεξεργασία μέσω των δυνατοτήτων που η ίδια η τεχνολογία παρέχει

Οι μαθητές ενθαρρύνονται να συζητούν τα προβλήματα που αντιμετωπίζουν και να συνεργάζονται με τους συμμαθητές τους.

14. Επισήμανση μικρομεταβολών στην οργάνωση του μαθήματος και στο νόημα των εννοιών

Δεν αναφέρονται.

15. Διδακτικό συμβόλαιο

Οι μαθητές πρέπει να τηρούν τους όρους του διδακτικού συμβολαίου που συντάχθηκε στην αρχή της χρονιάς και που θα περιλαμβάνει κανόνες για τη σωστή χρήση του σχολικού εργαστηρίου.

Οι μαθητές μπορούν να συζητούν στο πλαίσιο της συνεργασίας τους σε λογικά όρια, χωρίς να παρενοχλούν τους συμμαθητές τους και την εκπαιδευτική διαδικασία.

Ο/Η εκπαιδευτικός παρακολουθεί και συντονίζει τις ομάδες στη διάρκεια εκτέλεσης των δραστηριοτήτων και παρέχει όλες τις οδηγίες που θα διευκολύνουν τους μαθητές να ολοκληρώσουν την εργασία τους. Τα φύλλα εργασίας είναι απλά και μέσα στις δυνατότητες των μαθητών, οπότε το διδακτικό συμβόλαιο δεν αναμένεται να ανατραπεί.

16. Οργάνωση τάξης – εφικτότητα σχεδίασης

Το διδακτικό σενάριο προβλέπεται να υλοποιηθεί στο εργαστήριο Πληροφορικής όπου προτείνεται να υπάρχει projector ή κατάλληλο λογισμικό επίδειξης. Η κάθε ομάδα θα έχει στη διάθεσή της σε ηλεκτρονική μορφή το φύλλο εργασίας που πρόκειται να επεξεργαστεί.

Προτείνεται οι μαθητές να οργανωθούν σε ομάδες των 3-4 ατόμων.

Οι δραστηριότητες ψυχολογικής και γνωστικής προετοιμασίας, οι δραστηριότητες διδασκαλίας και εμπέδωσης του αντικειμένου και η αξιολόγηση γίνονται στο σχολικό εργαστήριο, με τον κάθε μαθητή να δουλεύει συνεργατικά με τους συμμαθητές του.

Οι δραστηριότητες που αφορούν στην κατασκευή/συναρμολόγηση του μετεωρολογικού σταθμού προτείνεται αρχικά να παρουσιαστούν/συζητηθούν στην ολομέλεια. Στη συνέχεια κάθε ομάδα θα αναλάβει την προετοιμασία και υλοποίηση τουλάχιστον μίας εξ αυτών την οποία θα παρουσιάσει (κατά τη διάρκεια της υλοποίησης της) στους υπόλοιπους μαθητές. Ο διδάσκων θα έχει υποστηρικτικό ρόλο σε αυτή τη φάση.

Οι δραστηριότητες προγραμματισμού σε Python θα υλοποιηθούν από όλες τις ομάδες ταυτόχρονα. Μετά την ολοκλήρωση του η κάθε ομάδα θα δοκιμάσει την ορθότητα του κώδικα της στο μετεωρολογικό σταθμό.

Η οργάνωση των μαθητών σε ομάδες ευνοεί την αλληλεπίδραση και τη συνεργασία μεταξύ των μαθητών οι οποίοι αντιμετωπίζουν την επίλυση των προτεινομένων προβλημάτων ως ένα κοινό project, ως μια κοινή προσπάθεια.

17. Περιγραφή και ανάλυση των φύλλων εργασίας (ή και άλλου διδακτικού υλικού) – Αξιολόγηση

17.1 Φύλλο Εργασίας 1

Το φύλλο εργασίας 1 είναι εισαγωγικό και έχει ως στόχο την εξοικείωση των μαθητών με τον υπολογιστή μικρού μεγέθους Raspberry Pi. Οι μαθητές καλούνται να απαντήσουν σε ερωτήσεις σχετικές με το Raspberry Pi, τη χρησιμότητά του, τις λειτουργίες του, τα χαρακτηριστικά του, κλπ.

17.2 Φύλλο Εργασίας 2

Το φύλλο εργασίας 2 είναι επίσης εισαγωγικό και έχει ως στόχο την εξοικείωση των μαθητών με τα εξαρτήματα που θα συνθέσουν τον μετεωρολογικό σταθμό και με τους αισθητήρες που θα χρησιμοποιηθούν για την μέτρηση της θερμοκρασίας, της υγρασίας, της ατμοσφαιρικής πίεσης, της βροχόπτωσης, της κατεύθυνσης και ταχύτητας του ανέμου.

17.3 Φύλλο Εργασίας 3

Το φύλλο εργασίας 3 παρουσιάζει βήμα-βήμα τη διαδικασία συναρμολόγησης και συγκόλλησης του

- του [Raspberry Pi GPIO Header](#) πάνω στο [Raspberry Pi Weather Board](#)
- του [Adafruit BME280 Temperature / Pressure / Humidity Sensor](#)
- του [DS18B20 Temperature Sensor](#) αισθητήρα θερμοκρασίας.

Τέλος στο φύλλο εργασίας 3 ολοκληρώνεται η σύνδεση των αισθητήρων πάνω στον μετεωρολογικό σταθμό.

17.4 Φύλλο Εργασίας 4

Το φύλλο εργασίας 4 παρουσιάζει μέσα από καθοδηγούμενες δραστηριότητες την διαδικασία εγκατάστασης του λειτουργικού συστήματος (Rasbian) για τον υπολογιστή μικρού μεγέθους Raspberry Pi και τις απαραίτητες παραμετροποιήσεις για τη λειτουργία του ως μετεωρολογικό σταθμό.

17.5 Φύλλο Εργασίας 5

Το φύλλο εργασίας 5 έχει στόχο την εξοικείωση των μαθητών με τις βιβλιοθήκες των γλωσσών προγραμματισμού γενικότερα και ειδικότερα με τις βιβλιοθήκες της γλώσσας προγραμματισμού Python.

17.6 Φύλλο Εργασίας 6

Το φύλλο εργασίας 6 παρουσιάζει μέσα από καθοδηγούμενες δραστηριότητες την εγκατάσταση των απαραίτητων βιβλιοθηκών για τη λειτουργία και διαχείριση των αισθητήρων του μετεωρολογικού σταθμού.

17.7 Φύλλο Εργασίας 7

Το φύλλο εργασίας 6 αναλύει βήμα-βήμα τον κώδικα Python ο οποίος είναι υπεύθυνος για τη λειτουργία, συλλογή και παρουσίαση των μετρήσεων θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, βροχόπτωσης, ταχύτητας και κατεύθυνσης ανέμου.

18. Φύλλα εργασίας

18.1 Φύλλο Εργασίας 1: Γνωριμία με τον υπολογιστή μικρού μεγέθους Raspberry Pi

Αναζητήστε στο διαδίκτυο πληροφορίες σχετικά με τον υπολογιστή μικρού μεγέθους Raspberry Pi και απαντήστε στις παρακάτω ερωτήσεις:

- Τι είναι το Raspberry Pi και τι μπορούμε να κάνουμε με αυτό;
- Εκτός της πλακέτας του Raspberry Pi ποια επιπλέον εξαρτήματα χρειαζόμαστε για να καταστήσουμε πλήρως λειτουργικό;
- Ποιο είναι ένα ενδεικτικό κόστος αγοράς του τελευταίου μοντέλου Raspberry Pi μαζί με τα απαραίτητα παρελκόμενα για να είναι πλήρως λειτουργικό;
- Ποια είναι τα κύρια τεχνικά χαρακτηριστικά του; (επεξεργαστής, κύρια μνήμη, κάρτες επέκτασης, θύρες επέκτασης και εισόδου/εξόδου, κλπ)
- Τι είναι οι ακίδες GPIO που βρίσκονται πάνω στην πλακέτα;
- Τι λειτουργικό σύστημα μπορεί να υποστηρίξει;
- Βρείτε τρία (3) ενδιαφέροντα projects που έχουν υλοποιηθεί αξιοποιώντας το Raspberry Pi.
- Στην εικόνα που ακολουθεί συμπληρώστε τα βασικά συστατικά εξαρτήματα του Raspberry Pi 4.

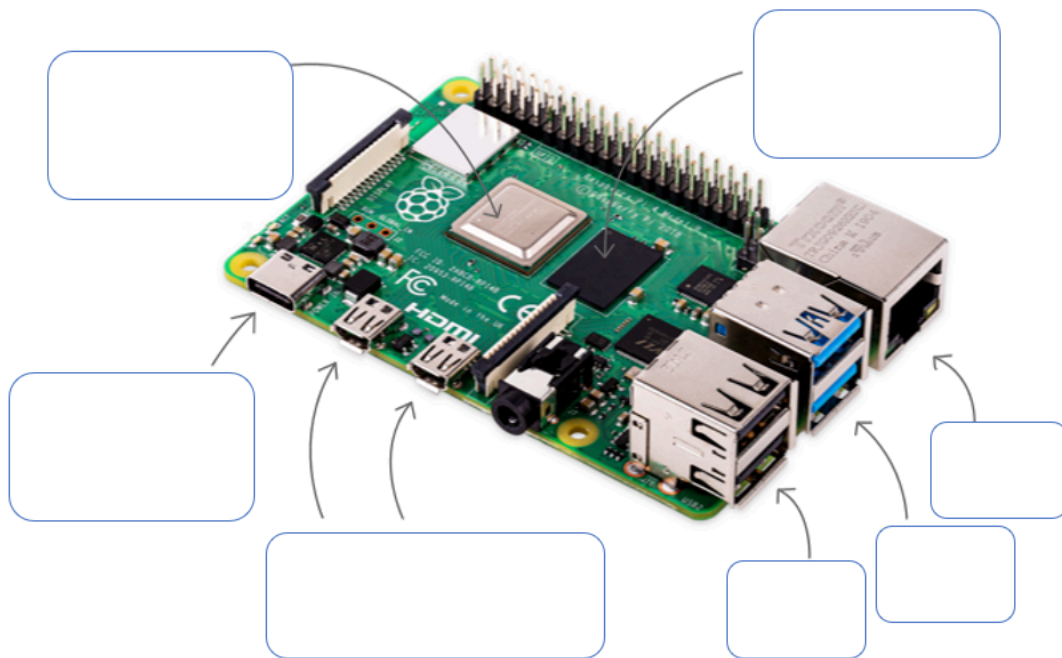


Figure 2. Ο υπολογιστής μικρού μεγέθους Raspberry Pi

18.2 Φύλλο Εργασίας 2: Αναγνώριση των εξαρτημάτων του μετεωρολογικού σταθμού

Το 2016 η εταιρία Oracle έστειλε περίπου 1000 kits μετεωρολογικών σταθμών που ήταν βασισμένοι στον υπολογιστή μικρού μεγέθους Raspberry Pi σε σχολεία σε όλο τον κόσμο. Το αρχικό Kit Oracle Weather Station περιελάμβανε εκτός από το Raspberry Pi ένα HAT που προσαρμόζεται πάνω στο Raspberry Pi και το οποίο του επιτρέπει να συλλέγει δεδομένα καιρού χρησιμοποιώντας μια ποικιλία αισθητήρων. Έχει σχεδιαστεί για χρήση σε σχολεία για να επιτρέπει στους μαθητές να δημιουργήσουν το δικό τους μετεωρολογικό σταθμό και διαθέτει μια σειρά πόρων για να υποστηρίξει τη χρήση του.



Figure 3. Μετεωρολογικός σταθμός βασισμένος σε Raspberry Pi

Στους παρακάτω συνδέσμους περιγράφεται αναλυτικά το project υλοποίησης ενός μετεωρολογικού σταθμού βασισμένου στο Raspberry Pi.

- <https://projects.raspberrypi.org/en/projects/build-your-own-weather-station>
- <https://www.bc-robotics.com/tutorials/raspberry-pi-weather-station-part-1/>

Για την υλοποίηση του μετεωρολογικού μας σταθμού θα επιλέξουμε την υλοποίηση προτείνεται στον δεύτερο σύνδεσμο από την BC Robotics (<https://www.bc-robotics.com/>).

Μεταβείτε στον δεύτερο σύνδεσμο και μελετήστε τον ενδεικτικό εξοπλισμό που θα χρειαστείτε για να υλοποιήσετε τον μετεωρολογικό σας σταθμό.

Στη συνέχεια συμπληρώστε τον πίνακα που ακολουθεί δίνοντας μια συνοπτική περιγραφή της λειτουργίας και του ρόλου του κάθε εξαρτήματος στον μετεωρολογικό σταθμό.

Ο εξοπλισμός που θα χρειαστεί είναι ο εξής:

A/A	Περιγραφή Εξοπλισμού	Ποσότητα	Συνοπτική περιγραφή της λειτουργίας και του ρόλου του εξοπλισμού
1.	Raspberry Pi (3A+ ή 4)	1	
2.	microSD card (> 16 GBytes)	1	
3.	Pi 3 or Pi 4 power supply	1	
4.	Raspberry Pi GPIO Header	1	
5.	Raspberry Pi HAT Hardware	1	
6.	3 Pin 2.54mm Screw Terminal	1	
7.	DS18B20 Temperature Sensor	1	
8.	Raspberry Pi Weather Board	1	
9.	SparkFun Weather Meters a. wind speed b. wind direction c. rain meter	1	



Figure 5. Αισθητήρες μετεωρολογικού σταθμού

18.3 Φύλλο Εργασίας 3: Συναρμολόγηση του Μετεωρολογικού σταθμού

Στόχος της δραστηριότητας αυτής είναι η συναρμολόγηση βήμα-βήμα του Weather Station HAT. Στο Weather Station HAT θα συνδεθούν όλοι οι αισθητήρες καιρού του σταθμού μας.

Το Weather Station HAT μετά την ολοκλήρωση της συναρμολόγησης θα έχει την εξής μορφή:

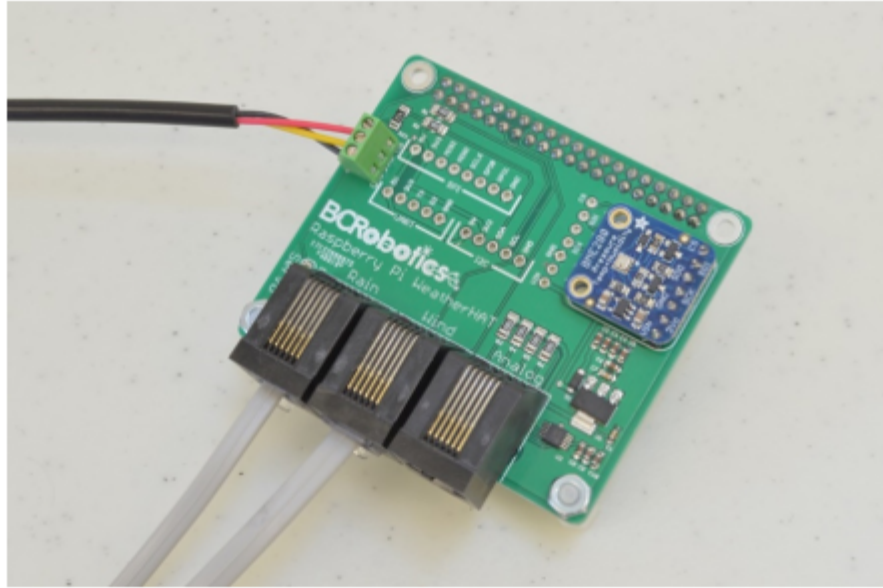


Figure 6. Weather Station Board

Μετά τη συναρμολόγηση του το Weather Station HAT θα τοποθετηθεί πάνω στο Raspberry Pi.

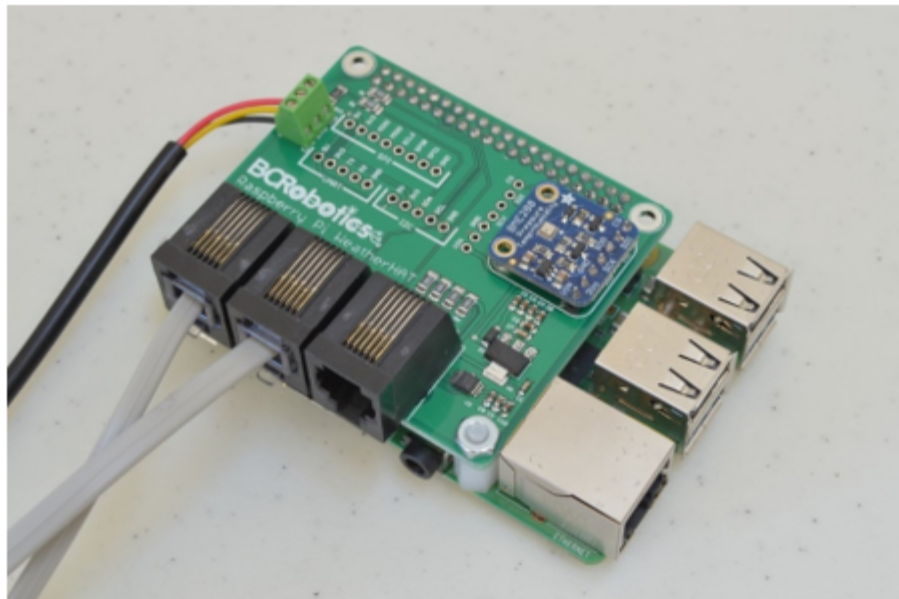


Figure 7. Raspberry Pi 3A+ με ενσωματωμένο (τοποθετημένο πάνω του) το Weather Station Board

Η συναρμολόγηση / συγκόλληση των εξαρτημάτων στο HAT θα γίνει με την εξής σειρά:

- Συγκόλληση του [Raspberry Pi GPIO Header](#) πάνω στο [Raspberry Pi Weather Board](#)
- Συγκόλληση του [Adafruit BME280 Temperature / Pressure / Humidity Sensor](#)
- Προσθήκη του [DS18B20 Temperature Sensor](#) αισθητήρα θερμοκρασίας.

Figure 8. Το τελικό αποτέλεσμα

Βήμα 1: Συγκόλληση του [Raspberry Pi GPIO Header](#) πάνω στο [Raspberry Pi Weather Board](#)

Αρχικά θα πρέπει να γίνει η συγκόλληση του [Raspberry Pi GPIO Header](#) πάνω στο [Raspberry Pi Weather Board](#) ώστε να μπορέσουμε να τοποθετήσουμε/συνδέσουμε το στο [Raspberry Pi Weather Board](#) στο Raspberry Pi. Συνιστάται η χρήση αυτής της κεφαλίδας, αλλά οποιαδήποτε συμβατή κεφαλίδα Raspberry Pi GPIO θα λειτουργήσει. Εάν δεν έχετε κάνει συγκόλληση στο παρελθόν ρίξτε μια ματιά στον «[To Soldering Is Easy!](#)»

Ξεκινήστε πιέζοντας τις δύο αντίθετες γωνίες του συνδέσμου ελέγχοντας προσεκτικά την ευθυγράμμιση των ακίδων και των υποδοχών, ώστε να διασφαλίσετε ότι ο σύνδεσμος θα τοποθετηθεί σωστά πριν κολλήσουμε και τους 40 ακροδέκτες.

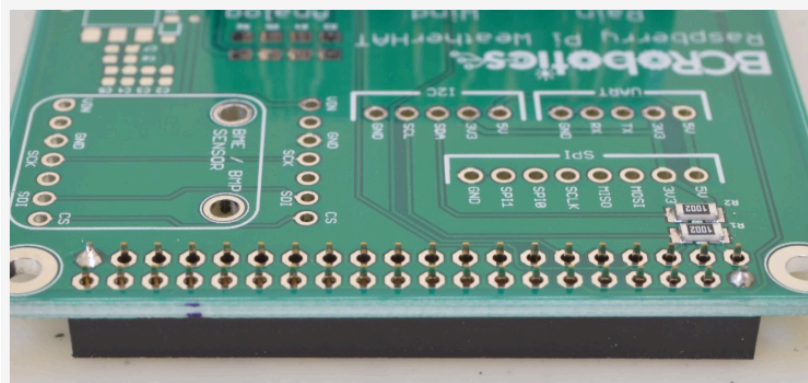


Figure 9. Τοποθέτηση του [Raspberry Pi GPIO Header](#) – μέρος 1

Μόλις ο σύνδεσμος ευθυγραμμιστεί με την πλακέτα και είστε ικανοποιημένοι με την ευθυγράμμιση, κολλήστε τους υπόλοιπους ακροδέκτες. Το αποτέλεσμα θα πρέπει να μοιάζει την φωτογραφία που ακολουθεί.

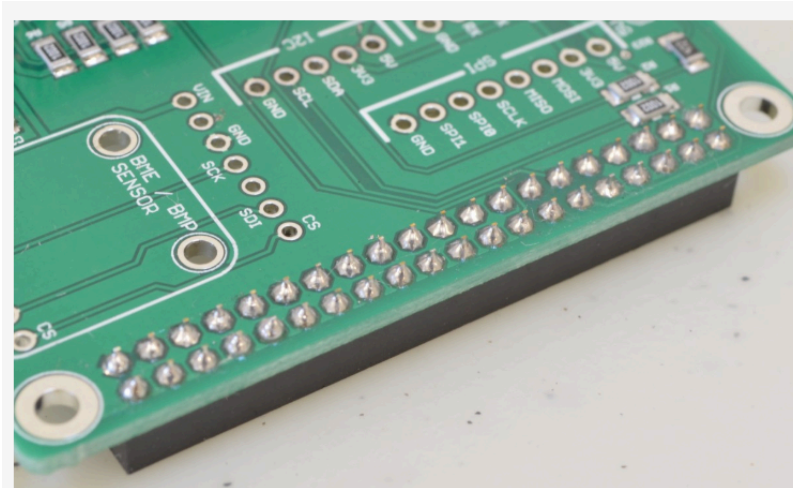


Figure 10. Τοποθέτηση του Raspberry Pi GPIO Header – μέρος 2

Βήμα 2: Τοποθέτηση του [BME280 Temperature / Pressure / Humidity Sensor](#)

Ο BME280 είναι ένας εκπληκτικός αισθητήρας θερμοκρασίας / υγρασίας / πίεσης και λειτουργεί εξαιρετικά με το Pi! Εάν χρησιμοποιείτε το Raspberry Pi 3+ υπάρχουν μερικά πράγματα που πρέπει να λάβετε υπόψη σας. Το Pi3 + διαθέτει επιπλέον κεφαλίδα 4 ακίδων για δυνατότητες Power over Ethernet (POE). Αυτός ο σύνδεσμος πιθανότατα θα έρθει σε επαφή με τις ακίδες στο κάτω μέρος του BME280. Υπάρχουν δύο τρόποι με τους οποίους αυτό μπορεί να αποφευχθεί: Αφαιρέστε την κεφαλίδα POE (αφού μπορεί εύκολα να επανεγκατασταθεί αργότερα) ή εγκαταστήστε το BME280 στην εναλλακτική θέση ακίδων (δείτε την εικόνα που ακολουθεί).

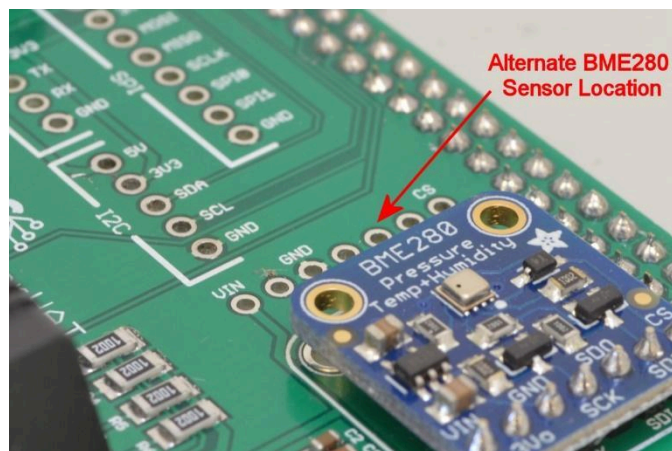


Figure 11. Τοποθέτηση του BME280 Temperature / Pressure / Humidity Sensor.

Βήμα 3: Προσθήκη του [DS18B20 Temperature Sensor](#) αισθητήρα θερμοκρασίας.

Στη συνέχεια θα τοποθετήσουμε τον ακροδέκτη βίδας για να συνδέσουμε τον αισθητήρα θερμοκρασίας DS18B20. Ο DS18B20 είναι ανθεκτικός στις καιρικές συνθήκες, οπότε μπορεί να τοποθετηθεί έξω από το κουτί του μετεωρολογικού σταθμού και να συνδεθεί με καλώδιο στην πλακέτα. Εάν προτιμάτε, μπορείτε να κολλήσετε τα καλώδια του αισθητήρα θερμοκρασίας DS18B20 απευθείας στην πλακέτα αλλά είναι πιο πρακτικό αυτό να γίνει μέσω

ενός ακροδέκτης βίδας. Ακριβώς όπως με όλους τους άλλους συνδέσμους, κολλήστε έναν ακροδέκτη στη θέση του, ελέγξτε την ευθυγράμμιση και στη συνέχεια κολλήστε και τους υπόλοιπους.

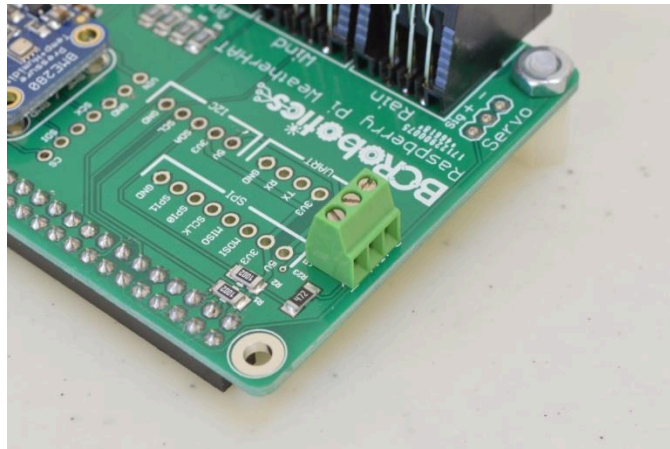


Figure 12. Τοποθέτηση ακροδέκτη βίδας

Ο αισθητήρας DS18B20 βγαίνει σε δύο διαφορετικά σετ χρωμάτων: κόκκινο, μαύρο και άσπρο ή κόκκινο, μαύρο και κίτρινο. Συνήθως από το κόκκινο καλώδιο θα περνάει τάση, το μαύρο θα είναι η γείωση και το λευκό / κίτρινο θα είναι το καλώδιο των δεδομένων. Φροντίστε να ελέγξετε το σωστά τα χρώματα του αισθητήρα DS18B20.

Συνδέστε το DS18B20 σας σύμφωνα με τον κωδικό χρώματος του αισθητήρα σας.

Βήμα 4 – Βροχόπτωση και Ταχύτητα του ανέμου

Το Ανεμόμετρο και ο μετρητής βροχόπτωσης είναι και οι δύο βασικοί ψηφιακοί αισθητήρες. Το ανεμόμετρο μετράει την ταχύτητα ανέμου μετρώντας πόσο γρήγορα περιστρέφεται σε ένα δεδομένο χρονικό διάστημα, ενώ ο μετρητής βροχής μετρά πόσες φορές το μανόμετρο γεμίζει με μια ακριβή ποσότητα νερού κατά τη διάρκεια μίας δεδομένης χρονικής περιόδου. Οι δύο αυτοί αισθητήρες δεν θα χρειαστεί να συγκολληθούν, απλά θα συνδεθούν μαζί με τον αισθητήρα κατεύθυνσης του ανέμου.

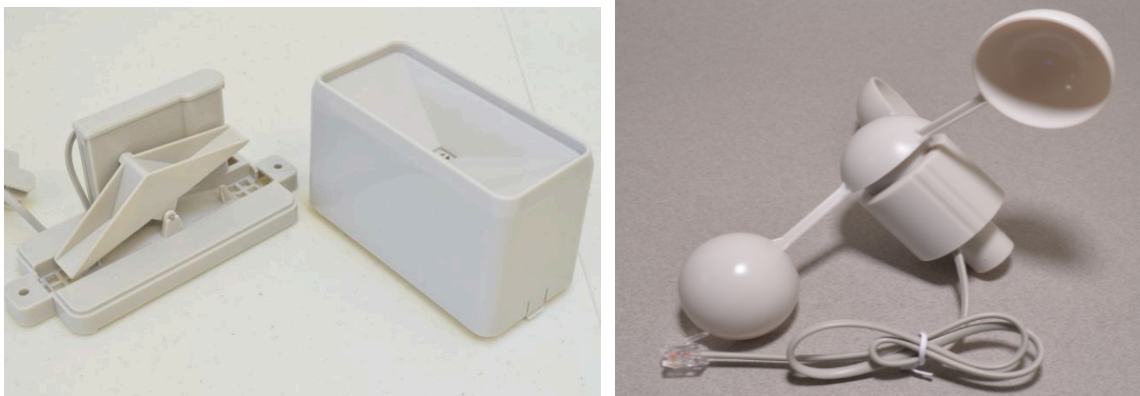


Figure 13. Μετρητής βροχής και Μετρητής ταχύτητας αέρα

Επίσης και ο αναλογικός αισθητήρας που υπολογίζει την κατεύθυνση του ανέμου χρειάζεται απλώς να συνδεθεί στο Weather Board HAT.



Figure 14. Αισθητήρας κατεύθυνσης ανέμου

Βήμα 5: Σύνδεση των αισθητήρων ανέμου και βροχής

Η σύνδεση αυτών των αισθητήρων είναι αρκετά απλή - αν κοιτάξετε προσεκτικά, ένα από τα δύο βύσματα RJ11 στο τέλος των καλωδίων τους έχει 4 καλώδια σε αυτό, ενώ το άλλο έχει δύο. Συνδέστε τη γραμμή 4 καλωδίων στην υποδοχή "Wind" στην πλακέτα καιρού, και συνδέστε τη γραμμή 2 καλωδίων στην υποδοχή "Rain".

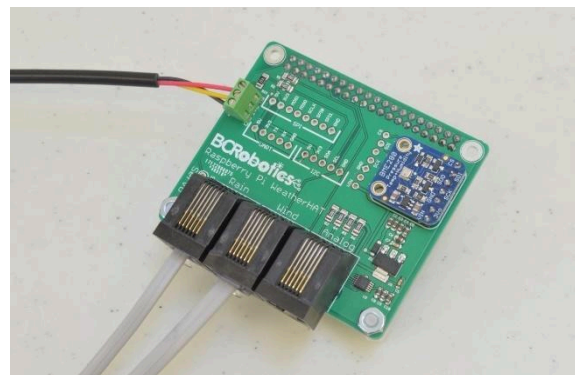
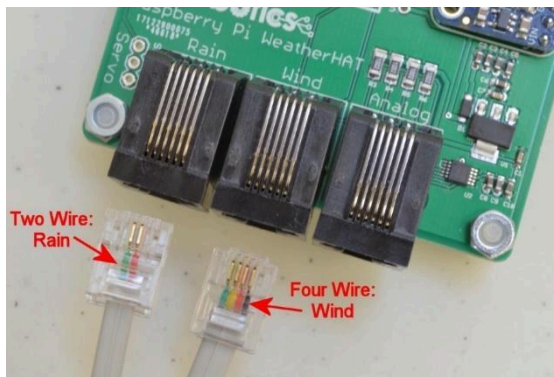


Figure 15. Σύνδεση των αισθητήρων βροχής και αέρα στο

Ο μετεωρολογικός μας σταθμός είναι έτοιμος :-)

18.4 Φύλλο Εργασίας 4: Εγκατάσταση Λειτουργικού Συστήματος στο Raspberry Pi

Στο προηγούμενο φύλλο εργασίας συναρμολογήσαμε τον μετεωρολογικό μας σταθμό. Έχοντας ολοκληρώσει με επιτυχία το κομμάτι του Υλικού (Hardware) μπορούμε να εργαστούμε από την πλευρά του λογισμικού / κωδικοποίησης του σταθμού.

Σε αυτή τη δραστηριότητα λοιπόν:

1. θα εκκινήσουμε το Raspberry Pi
2. θα εγκαταστήσουμε το λειτουργικό σύστημα Raspbian
3. θα κάνουμε κάποια αρχική διαμόρφωση του λειτουργικού συστήματος Raspbian
4. θα εγκαταστήσουμε βιβλιοθήκες Python για μερικούς από τους αισθητήρες

Για να λειτουργήσει ο μικροϋπολογιστής Raspberry Pi θα πρέπει να εγκατασταθεί το λειτουργικό σύστημα Raspbian (βασισμένο σε Debian) και στη συνέχεια να γίνουν οι απαραίτητες παραμετροποιήσεις.

Τι θα χρειαστούμε;

- Τον συναρμολογημένο μετεωρολογικό σταθμό που δημιουργήσαμε στις προηγούμενες δραστηριότητες
- MicroSD Card
- Κατάλληλο τροφοδοτικό Raspberry Pi 3, 3+, 4
- USB Πληκτρολόγιο & Ποντίκι
- HDMI monitor
- Πρόσβαση στο διαδίκτυο

Το λειτουργικό σύστημα είναι διαθέσιμο εδώ:

<https://www.raspberrypi.org/downloads/raspbian/>

Η εγκατάσταση θα ολοκληρωθεί στα επόμενα πέντε βήματα

1. Διαμόρφωση της κάρτας SD που θα τοποθετήσουμε στον υπολογιστή. Καλό θα είναι η χωρητικότητα της κάρτας να είναι τουλάχιστον 32GB.
2. Αντιγραφή των αρχείων του λειτουργικού συστήματος στην κάρτα SD.
3. Αφαιρούμε την κάρτα SD από τον υπολογιστή και τοποθέτηση της στο Raspberry.
4. Εκκίνηση του Raspberry Pi και εκκίνηση της εγκατάστασης του Raspbian. Αναζητήστε στο διαδίκτυο οδηγίες για την εγκατάσταση του Λειτουργικού Συστήματος στο Raspberry Pi και ολοκληρώστε την εγκατάσταση/παραμετροποίηση.
5. Παραμετροποίηση του περιβάλλοντος
 - a. Εγκατάσταση Ελληνικού πληκτρολογίου
 - b. Σύνδεση στο διαδίκτυο για λήψη ενημερώσεων
 - c. Ρύθμιση TimeZone και Area
 - d. Ενεργοποίηση των αισθητήρων I2C και 1-Wire Interfaces

Μόλις ολοκληρωθούν τα παραπάνω, μπορούμε να προχωρήσουμε και να παραμετροποιήσουμε τα μερικά ακόμη πράγματα.

Κάντε κλικ στο Logo Raspberry που βρίσκεται στην επάνω αριστερή γωνία, μετακινηθείτε προς τα κάτω στις Προτιμήσεις και επιλέξτε "Raspberry Pi Configuration".

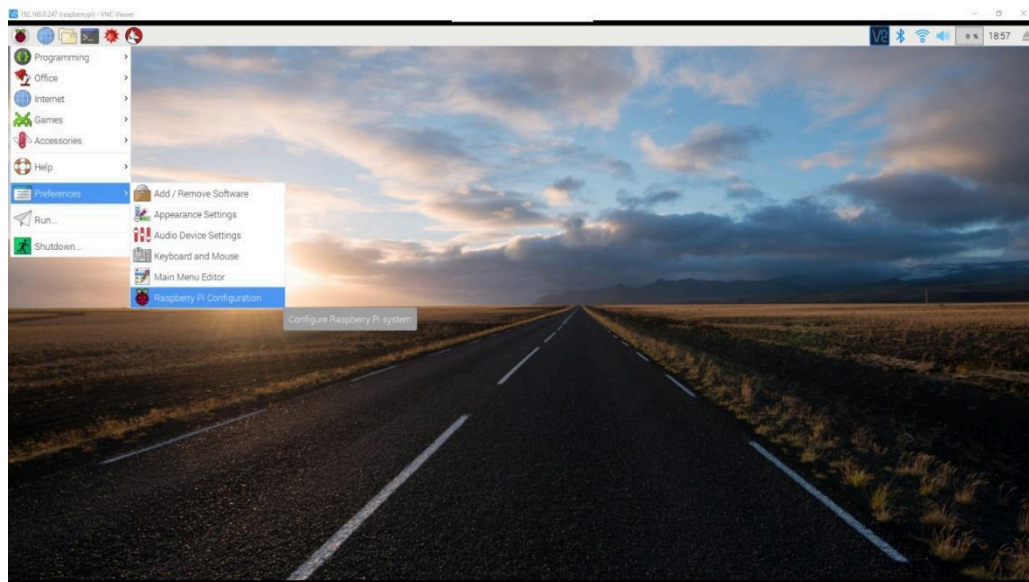


Figure 16. Το περιβάλλον του λειτουργικού συστήματος Raspbian

Στην πρώτη καρτέλα μπορείτε να αλλάξετε τις ρυθμίσεις Overscan ώστε να εξαφανιστούν τυχόν κενές μαύρες περιοχές που μπορεί να εμφανίζονται αριστερά και δεξιά στην οθόνη σας.

Στη δεύτερη καρτέλα θα πρέπει να ενεργοποιήσουμε τις διεπαφές I2C και 1-Wire. Αυτές είναι οι δύο διεπαφές που χρησιμοποιούν οι αισθητήρες μας.

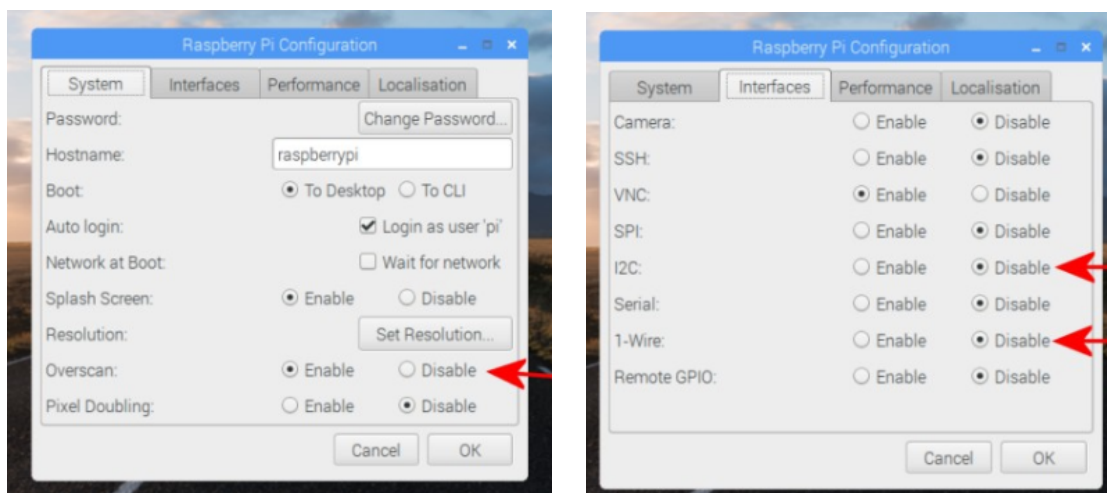


Figure 17. Παραμετροποίηση του Raspbian ώστε να μπορεί να επικοινωνεί με τους εξωτερικούς αισθητήρες

Κάντε επανεκκίνηση εάν σας ζητηθεί. Μόλις το Pi εκκινήσει, είμαστε έτοιμοι να προχωρήσουμε.

Προτού ξεκινήσουμε την εγκατάσταση των βιβλιοθηκών που θα χρειαστεί για να αποκαταστήσουμε την επικοινωνία και να έχουν τη δυνατότητα διαχείρισης των αισθητήρων μας θα πρέπει να συνδέσουμε το Raspberry Pi στο διαδίκτυο. Εάν χρησιμοποιείτε WiFi, αυτό πιθανότατα είχε ήδη ρυθμιστεί κατά την αρχική διαμόρφωση. Εάν δεν λειτουργεί ή παραλείψατε αυτό το βήμα, μπορείτε να αποκτήσετε πρόσβαση κάνοντας κλικ στο λογότυπο συνδεσιμότητας (που βρίσκεται δίπλα από τον έλεγχο έντασης στην επάνω δεξιά γωνία της οθόνης).

Εναλλακτικά μπορείτε απλώς να συνδέσετε το Raspberry Pi ενσύρματα στο διαδίκτυο με ένα καλώδιο Ethernet στην αντίστοιχη θύρα.

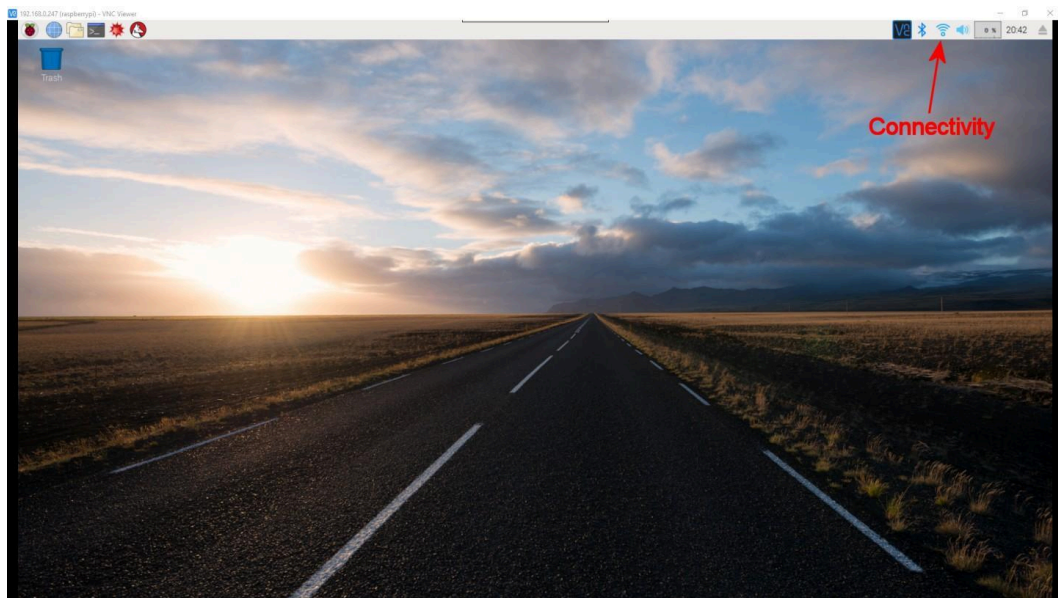


Figure 18. Ρυθμίσεις σύνδεσης στο διαδίκτυο

Αφού ολοκληρώσατε τη σύνδεση σας στο διαδίκτυο μπορείτε να ανοίξετε το τερματικό του λειτουργικού συστήματος ώστε να προχωρήσουμε στην εγκατάσταση (είτε από το μενού είτε με τον συνδυασμό Ctrl+Alt+T).

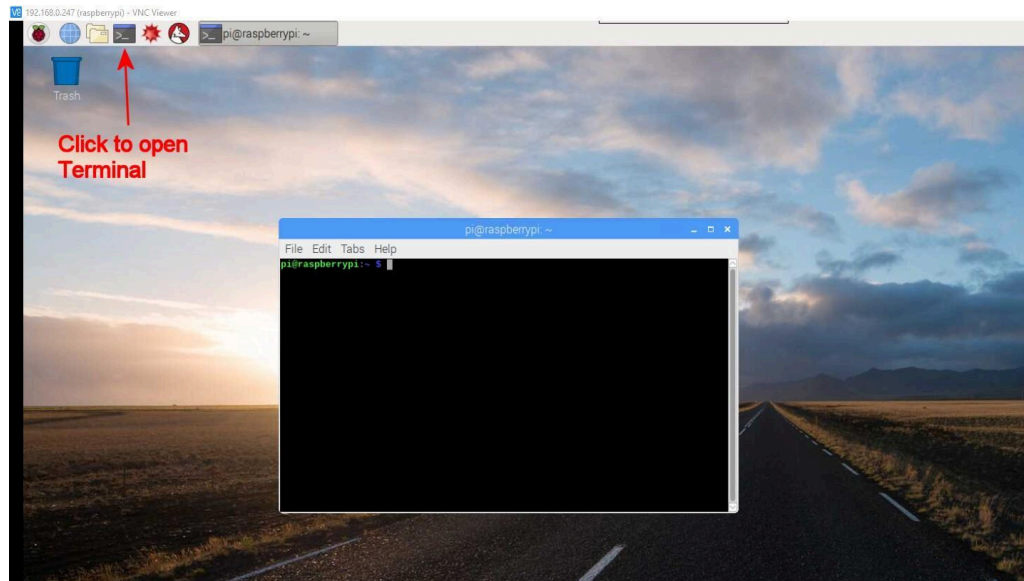


Figure 19. Το τερματικό του Raspbian

18.5 Φύλλο Εργασίας 5: Γνωριμία με τις βιβλιοθήκες της Python

Η Python είναι μια γλώσσα προγραμματισμού γενικού σκοπού που δίνει έμφαση στο να είναι ο κώδικας πιο ευανάγνωστος και να βοηθήσει στην παραγωγικότητα του προγραμματιστή.

Η Python περιγράφεται συχνά ως μια εύκολη στην χρήση γλώσσα. Έχει σαφή σύνταξη, μεγάλη κοινότητα και είναι κατάλληλη για αρχάριους γιατί έχει μικρότερη καμπύλη μάθησης.

Έχοντας μια πολύ απλή και ευανάγνωστη σύνταξη, μπορούμε να γράψουμε γρήγορα και αποτελεσματικά τον κώδικά μας, και να δαπανήσουμε λιγότερο χρόνο για την επιδιόρθωση σφαλμάτων και την αντιμετώπιση προβλημάτων. Η Python, περισσότερο από οποιαδήποτε άλλη γλώσσα, μας επιτρέπει να μεγιστοποιήσουμε το χρόνο που αφιερώνουμε για την λύση του πραγματικού προβλήματος, την διερεύνηση τάσεων και ιδεών και την εξαγωγή αποτελεσμάτων.

Η μεγάλη δύναμη της Python είναι το πλήθος των ελεύθερων third-party βιβλιοθηκών. Στις αρχές του 2019, το επίσημο Python repository περιείχε γύρω στις 165 χιλιάδες projects. Σχεδόν για κάθε περίπτωση, υπάρχει διαθέσιμη μια υψηλής ποιότητας βιβλιοθήκη.

Αναζητήστε πληροφορίες σχετικά με τις βιβλιοθήκες γλωσσών προγραμματισμού γενικότερα και ειδικότερα σχετικά με τις βιβλιοθήκες της Python.

Ενδεικτικά μπορείτε να ψάξετε πληροφορίες στους παρακάτω συνδέσμους. Μην περιοριστείτε μόνο σε αυτούς

https://el.wikipedia.org/wiki/%CE%A0%CF%81%CF%8C%CF%84%CF%85%CF%80%CE%B7_%CE%B2%CE%B9%CE%B2%CE%BB%CE%B9%CE%BF%CE%B8%CE%AE%CE%BA%CE%B7

<https://el.wikipedia.org/wiki/Pygame>

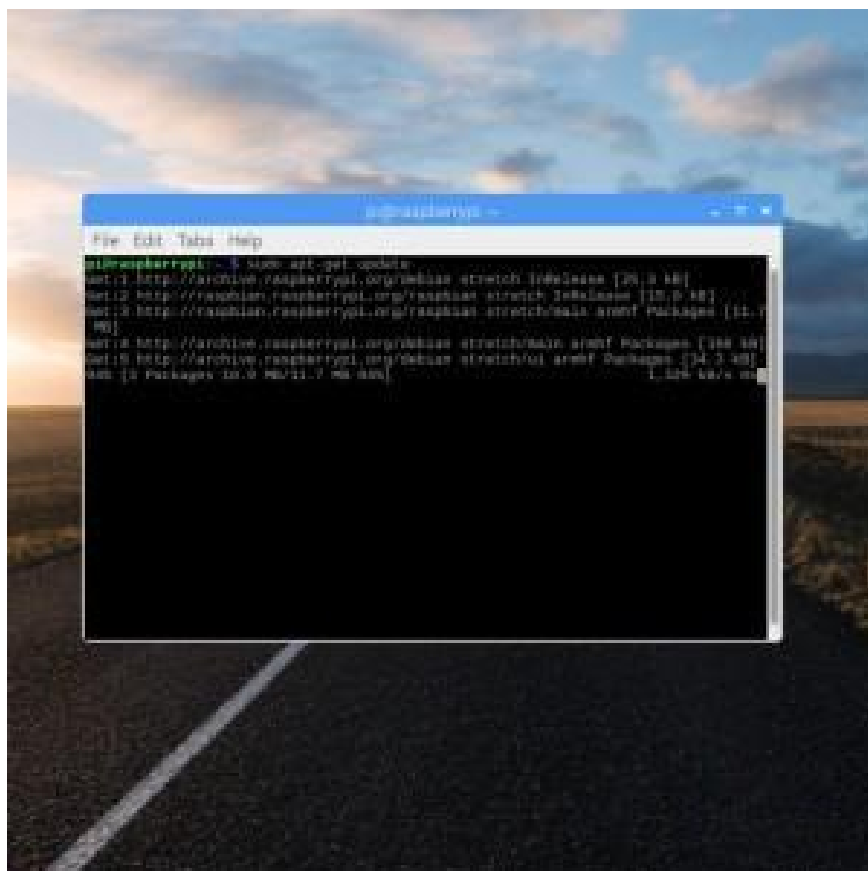
<https://www.pygame.org/wiki/about>

<https://www.secnews.gr/149557/top-%CE%B5%CF%81%CE%B3%CE%B1%CE%BB%CE%B5%CE%AF%CE%B1-python-%CF%80%CE%BF%CF%85-%CE%B4%CE%B5%CE%BD-%CE%BC%CF%80%CE%BF%CF%81%CE%B5%CE%AF%CF%84%CE%B5-%CE%B1%CE%B3%CE%BD%CE%BF%CE%AE%CF%83%CE%B5%CF%84/>

Αφού μελετήσετε πληροφορίες και πηγές στο διαδίκτυο απαντήστε παρακάτω ερωτήσεις:

- Τι είναι οι βιβλιοθήκες γλωσσών προγραμματισμού;
- Σε τι χρησιμεύουν οι βιβλιοθήκες γλωσσών προγραμματισμού; Καταγράψετε μερικά παραδείγματα.
- Αναφέρετε πέντε σημαντικές βιβλιοθήκες της Python καθώς και μία σύντομη περιγραφή των σκοπών που εξυπηρετεί η κάθεμιά τους

Για την αξιοποίηση των αισθητήρων του μετεωρολογικού μας σταθμού θα πρέπει να εγκαταστήσουμε μία σειρά βιβλιοθηκών της γλώσσας Python. Ο απλούστερος τρόπος για να γίνει η εγκατάσταση είναι το τερματικό (Terminal).



Εκκινήστε το τερματικό είτε από το μενού του Raspberry Pi είτε πατώντας τον συνδυασμό πλήκτρων Ctrl+Alt+T.

Εκτελέστε καθεμία από αυτές τις παρακάτω εντολές με τη σειρά που εμφανίζονται. Πληκτρολογήστε την κάθε εντολή στο τερματικό (ή αντιγράψτε την και επικολλήστε την στο παράθυρο του τερματικού) και πατήστε Enter.

```
sudo apt-get update

sudo pip3 install --upgrade setuptools
pip3 install RPI.GPIO
```



```
pip3 install adafruit-blinka
```

Στη συνέχεια θα εγκαταστήσουμε τη βιβλιοθήκη Adafruit BME280 για το Python. Αυτή η βιβλιοθήκη δημιουργήθηκε επίσης από την Adafruit και έχει σχεδιαστεί για να λειτουργεί με τον αισθητήρα θερμοκρασίας / υγρασίας / πίεσης BME280.

Εκτελέστε την ακόλουθη εντολή στο τερματικό:

```
sudo pip3 install adafruit-circuitpython-bme280
```

Στη συνέχεια, πρόκειται να ρυθμίσουμε το ADS1015 Analog to Digital chip το οποίο θα χρησιμοποιηθεί για να διαβάσουμε την κατεύθυνση του ανέμου. Για να εγκαταστήσετε τη βιβλιοθήκη, πληκτρολογήστε την ακόλουθη εντολή στο τερματικό:

```
sudo pip3 install adafruit-circuitpython-ads1x15
```

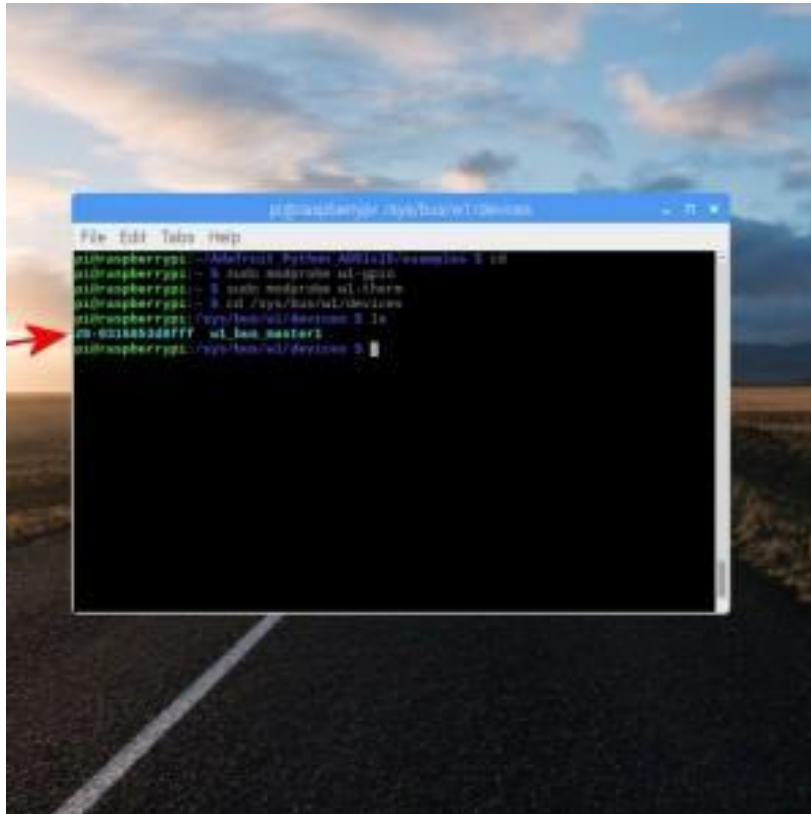


Figure 21. Εγκατάσταση απαιτούμενων βιβλιοθηκών από το περιβάλλον του τερματικού

Στη συνέχεια θα ρυθμίσουμε τον αισθητήρα θερμοκρασίας DS18B20. Η ρύθμιση για αυτόν τον αισθητήρα είναι λίγο διαφορετική. Οι δύο τελευταίοι αισθητήρες χρησιμοποίησαν και το interface I2C, το DS18B20 χρησιμοποιεί το interface 1-Wire. Εισαγάγετε τις ακόλουθες εντολές στο τερματικό:

```
cd
```

```
sudo modprobe w1-gpio  
sudo modprobe w1-therm  
cd /sys/bus/w1/devices  
ls
```

Δεδομένου ότι ο αισθητήρας θερμοκρασίας θα αναφέρεται στο φάκελο συσκευών, θα πρέπει να μπορούμε να βλέπουμε τα περιεχόμενα αυτού του φακέλου.

Η τελευταία εντολή "ls" θα εμφανίσει τα περιεχόμενα του φακέλου στο παράθυρο.

Ο αισθητήρας DS18B20 που χρησιμοποιούμε στο παράδειγμα μας έχει διεύθυνση 28-0316853d8fff. Κάθε αισθητήρας έχει μία μοναδική διεύθυνση και συνεπώς ο αριθμός του δικού σας αισθητήρα δεν θα είναι ίδιος με αυτόν του παραδείγματος μας (που εμφανίζεται στην εικόνα).

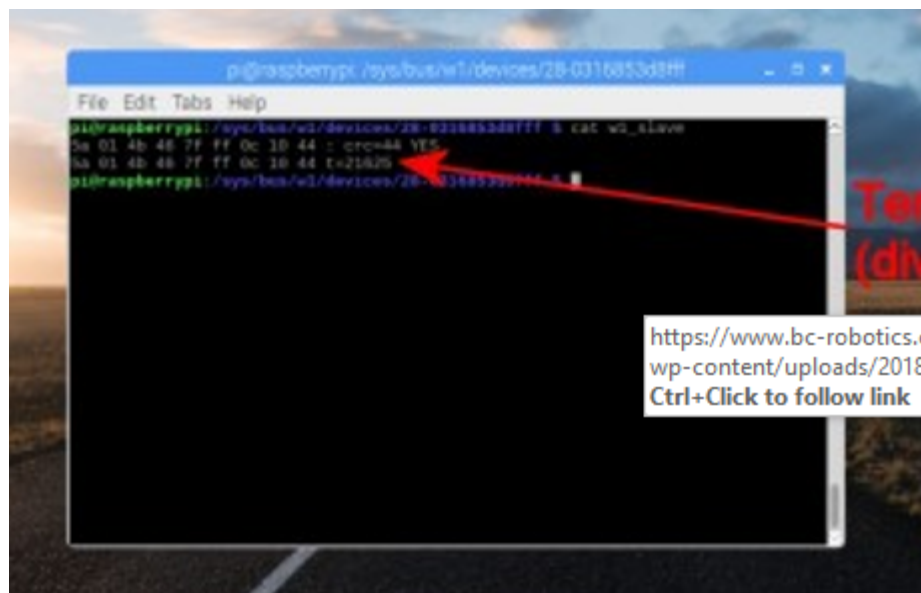


Figure 22. Εμφάνιση τιμών θερμοκρασίας από τον αισθητήρα DS18B20

Ήρθε η ώρα να «διαβάσουμε» την θερμοκρασία από τον αισθητήρα DS18B20!

Εισαγάγετε την ακόλουθη εντολή με το αναγνωριστικό αισθητήρα

```
cd *sensor ID*
```

Τώρα μπορούμε να πάρουμε τη θερμοκρασία από τον αισθητήρα εισάγοντας την ακόλουθη εντολή:

```
cat w1_slave
```

Η θερμοκρασία βρίσκεται στη δεύτερη γραμμή αλλά η κατανόηση της καταγεγραμμένης θερμοκρασίας δεν είναι και τόσο εύκολη γιατί ο αισθητήρας BME280 δεν εμφανίζει την τιμή κατευθείαν σε βαθμούς Κελσίου. Για να το κάνουμε αυτό θα πρέπει να πάρουμε τον αριθμό αυτόν και να τον διαιρέσουμε με 1000.

Επειδή ο DS18B20 δεν είναι ο ευκολότερος αισθητήρας για άμεση ανάγνωση στο Python, θα εγκαταστήσουμε μια τελευταία βιβλιοθήκη για να την κάνουμε την όλη διαδικασία ευκολότερη.

Στο τερματικό πληκτρολογήστε τις ακόλουθες εντολές:

```
cd  
sudo apt-get install python3-w1thermsensor
```

18.7 Φύλλο Εργασίας 7: Βήμα-βήμα μελέτη του κώδικα σε Python.

Όλοι οι σύνθετοι αισθητήρες που έχουν ενσωματωθεί στον υπολογιστή μικρού μεγέθους Raspberry Pi συλλέγουν και έχουν την δυνατότητα να επιστρέψουν μετεωρολογικά δεδομένα. Για να γίνει αυτό όμως θα πρέπει να συντάξουμε ένα πρόγραμμα σε Python το οποίο θα ελέγχει κάθε έναν από τους αισθητήρες, θα συλλέγει και θα εμφανίζει κάθε 15 δευτερόλεπτα:

- τη θερμοκρασία
- την υγρασία
- την ατμοσφαιρική πίεση
- την ταχύτητα του ανέμου
- την κατεύθυνση του ανέμου
- το ύψος της βροχόπτωσης

Ήρθε η ώρα λοιπόν να «βουτήξουμε» στον κώδικα της Python.

Εκκινήστε το περιβάλλον προγραμματισμού που χρησιμοποιείτε. Εμείς για το παράδειγμα μας θα χρησιμοποιήσουμε το περιβάλλον IDLE.

Για να κάνετε κι εσείς το ίδιο εκτελέστε την παρακάτω εντολή στο τερματικό

```
idle3
```

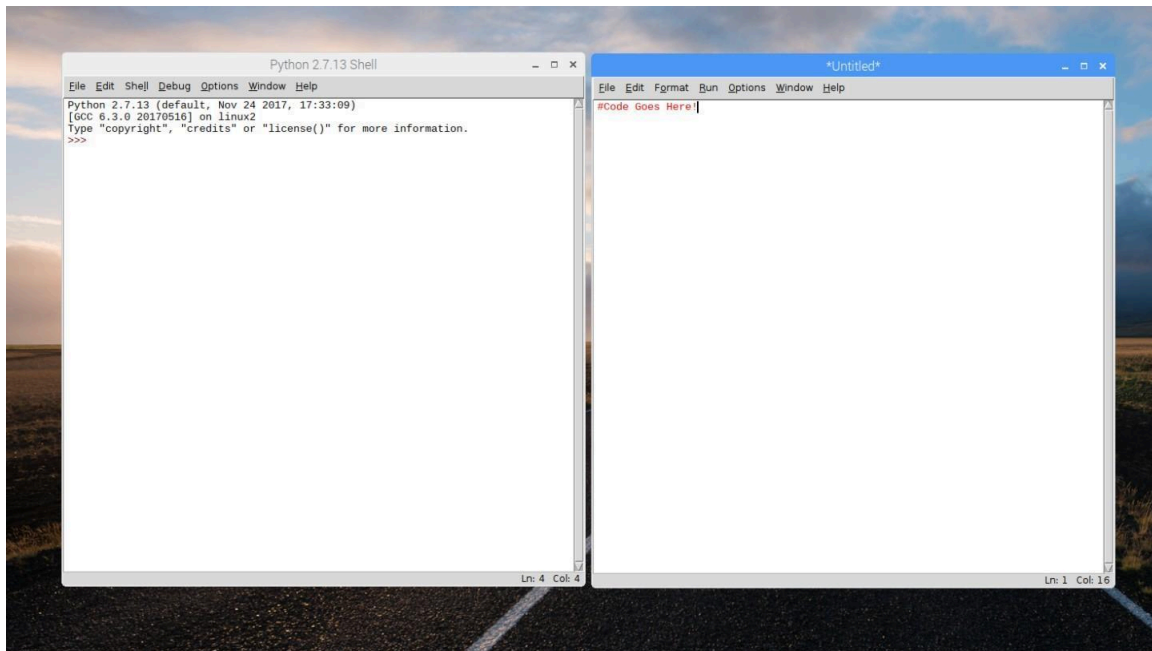


Figure 23. Ο Editor IDLE της Python

Στο παράθυρο Python Shell που μόλις ανοίξατε, κάντε κλικ στο μενού "Αρχείο" και επιλέξτε "Νέο αρχείο" για να ξεκινήσετε ένα νέο αρχείο Python. Το νέο, κενό αρχείο θα ανοίξει σε ένα νέο παράθυρο στο οποίο θα γράψουμε τον κώδικα του προγράμματος μας.

Όπως θε γνωρίζετε στον προγραμματισμό υπάρχουν πολλοί εναλλακτικοί τρόποι υλοποίησης ενός προγράμματος που οδηγούν στο ίδιο αποτέλεσμα που στην περίπτωση μας είναι η συλλογή δεδομένων από αισθητήρες και η εμφάνιση τους κάθε 15 δευτερόλεπτα.

Για γίνει αυτό θα πρέπει να δομήσουμε το πρόγραμμα γύρω από έναν απλό βρόχο 15 δευτερολέπτων. Προφανώς το χρονικό διάστημα μεταξύ των μετρήσεων είναι κάτι που μπορείτε να το αλλάξετε και να το διαμορφώσετε ανάλογα με τις ανάγκες σας.

Επειδή ο κώδικας Python που θα πρέπει να δημιουργήσουμε είναι αρκετά μεγάλος τον έχουμε χωρίσει σε μικρότερα βήματα.

Για όσους δεν είναι εξοικειωμένοι με την Python, να σημειώσουμε ότι η στοίχιση σε αυτή τη γλώσσα προγραμματισμού παίζει πολύ σημαντικό ρόλο.

Συνεπώς φροντίστε να μορφοποιήσετε τον κώδικα σας ακριβώς όπως φαίνεται στο παράδειγμα.

Η προτεινόμενη προσέγγιση για τα επόμενα βήματα είναι η εξής:

- μελετήστε προσεκτικά κάθε τμήμα κώδικα που ακολουθεί αξιοποιώντας τόσο τα σχόλια που θα βρείτε μέσα στον κώδικα όσο και μετά το τέλος του (σχόλια με αναφορά στις γραμμές του προγράμματος)
- σε κάθε νέο τμήμα κώδικα που μελετήστε τις προσθήκες (πράσινες επισημάνσεις) κώδικα προσπαθώντας να καταλάβετε τις νέες λειτουργίες που προσθέτει

Με αυτό τον τρόπο βήμα-βήμα θα έχετε κατανοήσει την λειτουργία του προγράμματος συλλογής και παρουσίασης μετεωρολογικών δεδομένων.

18.7.1 Βήμα 1: Δημιουργία του βασικού πλαισίου του προγράμματος μας

Προτού μπορέσουμε να διαβάσουμε κάποιον από τους αισθητήρες, θα χρειαστεί να δημιουργήσουμε το βασικό πλαίσιο του προγράμματος. Αυτό σημαίνει ότι πρέπει να εισαγάγουμε τη βιβλιοθήκη ώρας, να δημιουργήσουμε τον βρόχο μας (των 15 δευτερολέπτων) και να ορίσουμε το διάστημα που θέλουμε να θέσουμε σε παύση τον κώδικα κάθε φορά που ο βρόχος έχει ολοκληρωθεί.

```
1  import time
2
3  interval = 15  #How long we want to wait between loops (seconds)
4
5
6  while True:
7
8      time.sleep(interval)
```

Οι βιβλιοθήκες προστίθενται στο έργο χρησιμοποιώντας την εντολή `Import`. Στην πρώτη γραμμή εισάγουμε τη βιβλιοθήκη χρόνου.

Στη γραμμή 3 ορίζουμε μια μεταβλητή για την αποθήκευση του χρονικού διαστήματος σε δευτερόλεπτα που θέλουμε να θέσουμε σε αναστολή λειτουργίας το πρόγραμμα σε κάθε βρόχο του κώδικα.

Στη συνέχεια, θέλουμε να δημιουργήσουμε τον βρόχο - στην Python αυτό μπορεί να γίνει με διάφορους τρόπους. Για την περίπτωση μας η εντολή `While True` είναι μια καλή επιλογή.

Σημειώστε ότι όταν πατήσετε "Enter" στο περιβάλλον προγραμματισμού η επόμενη γραμμή γίνεται εσοχή.

Τέλος, στη γραμμή η εντολή 8 "`sleep`" σταματάει το πρόγραμμά μας χρησιμοποιώντας την τιμή της μεταβλητής που θέσαμε παραπάνω.

Πολύ σημαντική σημείωση: Επειδή πρόκειται να προσθέσουμε πολύ κώδικα στα επόμενα βήματα, για τη δική σας ευκολία ανάγνωσης, κάθε νέα γραμμή που προσθέτουμε θα επισημαίνεται με πράσινο χρώμα ενώ ταυτόχρονα στα αριστερά οι γραμμές κώδικα θα είναι αριθμημένες. Τα σχόλια που παραθέτουμε στο τέλος του προγράμματος αντιστοιχούν σε αυτούς τους αριθμούς γραμμών.

18.7.2 Βήμα 2: Ανάγνωση της θερμοκρασίας από τον αισθητήρα DS18B20

Κατά τη διάρκεια της παραμετροποίησης του μετεωρολογικού σταθμού εγκαταστήσαμε μια δέσμη βιβλιοθηκών (Python libraries) με σκοπό να απλοποιήσουμε τη διαδικασία συγγραφής του προγράμματος. Ήρθε η ώρα να αξιοποιήσουμε μία από αυτές τις βιβλιοθήκες για να διαβάσουμε τον αισθητήρα θερμοκρασίας DS18B20.

Στα επόμενα κομμάτια κώδικα πρόκειται (α) να εισαγάγουμε τη βιβλιοθήκη DS18B20, (β) να ρυθμίσουμε τον αισθητήρα, (γ) να διαβάσουμε τον αισθητήρα και (δ) να εκτυπώσουμε το αποτέλεσμα.

```
1  import time
2
3
4
5
6  interval = 15  #How long we want to wait between loops (seconds)
7
8  while True:
9
10     time.sleep(interval)
11
12     #Pull Temperature from DS18B20
```



```

13 temperature = w1thermsensor.get_temperature()
14
15 #Print the results
16 print('Temperature: ', temperature)

```

- (2) Όπως ακριβώς κάναμε με τη βιβλιοθήκη ώρας της Python (time), εδώ εισάγουμε τη βιβλιοθήκη w1Thermsensor
- (4) Στη συνέχεια, δημιουργούμε το αντικείμενο temperature και το δίνουμε όνομα
- (13) Παίρνουμε τη θερμοκρασία από τον αισθητήρα και την αποθηκεύουμε σε μια μεταβλητή που ονομάζεται temperature
- (16) Και τέλος εκτυπώνουμε το αποτέλεσμα χρησιμοποιώντας την εντολή εκτύπωσης

Δοκιμάστε να εκτελέσετε το πρόγραμμα πατώντας "F5". Αρχικά το περιβάλλον IDLE θα σας ζητήσει να αποθηκεύσετε. Μετά την αποθήκευση το πρόγραμμα θα πρέπει να εκτελεστεί και στο παράθυρο Python Shell θα πρέπει να δείτε τη θερμοκρασία να εμφανίζεται μετά από αναμονή 15 δευτερολέπτων.

Το πρόγραμμα θα συνεχίσει να δημοσιεύει ενημερωμένες τιμές θερμοκρασίας κάθε 15 δευτερόλεπτα.

Αν θέλετε να διακόψετε τη λειτουργία του προγράμματος θα πρέπει να πατήσετε "Ctrl + C".

18.7.3 Βήμα 3: Ανάγνωση του αισθητήρα BME280

Στη συνέχεια θα επιχειρήσουμε να διαβάσουμε τις τιμές που συλλέγει ο αισθητήρας BME280, ο οποίος μας παρέχει τρεις μετρήσεις: Θερμοκρασία, Ατμοσφαιρική Πίεση και Υγρασία.

Η βιβλιοθήκη που έχουμε εγκαταστήσει απλοποιεί σε μεγάλο βαθμό τον κώδικα που θα πρέπει να δημιουργήσουμε ώστε να πάρουμε τις τιμές που θέλουμε στην κατάλληλη μορφή.

Όπως ακριβώς κάναμε με τον προηγούμενο αισθητήρα, θα εισαγάγουμε τη βιβλιοθήκη του αισθητήρα BME280, θα ρυθμίσουμε τον αισθητήρα, θα διαβάσουμε τον αισθητήρα και θα εκτυπώσουμε τα αποτελέσματα.

Δεδομένου ότι τώρα χρησιμοποιούμε Circuit Python Compatible libraries αυτή τη φορά θα χρειαστεί να εισάγουμε μερικά επιπλέον κομμάτια κώδικα.

```

1 import time
2 from w1thermsensor import W1ThermSensor
3
4 import board
5 import busio
6 import adafruit_bme280
7 I2C = busio.I2C(board.SCL, board.SDA)

```

```

8
9 time = adafruit_bme280.Adafruit_BME280_I2C(i2c)
10
11
12 ds18b20 = W1ThermSensor()
13
14 interval = 15 #How long we want to wait between loops (seconds)
15
16 while True:
17
18     time.sleep(interval)
19
20     #Pull Temperature from DS18B20
21     temperature = ds18b20.get_temperature()
22
23     #Pull temperature from BME280
24     temp_c = time.temperature
25
26     #Pull pressure from BME280 Sensor & convert to kPa
27     pressure_pa = time.pressure
28     pressure = pressure_pa / 100
29
30     #Pull humidity from BME280
31     humidity = time.humidity
32
33     #Print the results
34     print( 'Temperature: ' , temperature)
35     print( 'Humidity: ' , humidity, '%')
36     print( 'Pressure: ' , pressure, 'kPa')
37     print( ' ')

```

- (4) Εισαγωγή της βιβλιοθήκης board από το Adafruit Blinka
- (5) Εισαγωγή της βιβλιοθήκης busio από το Adafruit Blinka
- (6) Εισαγωγή της βιβλιοθήκης Adafruit BME280
- (7) Διαμόρφωση του I2C για να καταστεί δυνατή η χρήση του I2C interface του Raspberry Pi
- (9) Δημιουργία του αντικειμένου αισθητήρα BME280 και απόδοση σε αυτό ενός ονόματος.

- Αν και δεν χρησιμοποιούμε τις τιμές του αισθητήρα θερμοκρασίας του κυκλώματος BME (αλλά τον αισθητήρα θερμοκρασίας DS18B20), θα πρέπει να διαβάσουμε τις τιμές θερμοκρασίας του. Αυτό θα πρέπει να γίνει λόγω του τρόπου με τον οποίο έχει γραφτεί η βιβλιοθήκη του αισθητήρα η οποία απαιτεί ανάγνωση όλων των τιμών. Σε διαφορετική περίπτωση θα υπάρξει διακοπή λειτουργίας του προγράμματος. Επομένως, διαβάζουμε τη θερμοκρασία και την ονομάσαμε "case temp". Η θερμοκρασία αυτή είναι ένδειξη της θερμοκρασίας του αέρα μέσα στο κουτί στο οποίο έχει τοποθετηθεί ο αισθητήρας BME.
- Μόλις ο κώδικας σας έχει ολοκληρωθεί, πατήστε ξανά το "F5" για να τον εκτελέσετε. Αυτή τη φορά θα πρέπει να υπάρχουν πολλές περισσότερες πληροφορίες!

Η κατεύθυνση του ανέμου είναι η επόμενη τιμή που θα διαβάσουμε. Όπως ακριβώς κάναμε και με τους προηγούμενους αισθητήρες, (α) εισαγάγουμε τη βιβλιοθήκη, (β) ρυθμίζουμε τον αισθητήρα μας και (γ) διαβάζουμε τα αποτελέσματα.

```
1 import time
2 from w1thermsensor import W1ThermSensor
3
4 import board
5 import busio
6 import adafruit_bme280
7 i2c = busio.I2C(board.SCL, board.SDA)
8
9 import adafruit_adxl345 as ADXL
```

```

10 from adafruit_ads1x15_analog_in import AnalogIn
11
12 bme = adafruit_bme280.Adafruit_BME280_I2C(i2c)
13 ads = ADS.ADS1015(i2c)
14 ads.gain =
15
16
17 ds18b20 = W1ThermSensor()
18
19 interval = 15 #How long we want to wait between loops (seconds)
20
21 while True:
22
23     time.sleep(interval)
24
25     #Pull Temperature from DS18B20
26     temperature = ds18b20.get_temperature()
27
28     #Pull temperature from BME280
29     case_temp = bme.temperature
30
31     #Pull pressure from BME280 Sensor & convert to kPa
32     pressure_pa = bme.pressure
33     pressure = pressure_pa / 10
34
35     #Pull humidity from BME280
36     humidity = bme.humidity
37
38     #Calculate wind direction based on ADC reading
39     chan = AnalogIn(ads, ADS.P0)
40     val = chan.value
41     windDir = "Not Connected"
42     windDeg = 99
43
44     if 20000 <= val <= 20000:
45         windDir = "N"

```

```
46     windDeg = 0
47
48     LE 1000 <= val <= 1050:
49         windDir = "NNE"
50         windDeg = 22.5
51
52     LE 1150 <= val <= 1200:
53         windDir = "NE"
54         windDeg = 45
55
56     LE 2000 <= val <= 2250:
57         windDir = "ENE"
58         windDeg = 67.5
59
60     LE 2300 <= val <= 2500:
61         windDir = "E"
62         windDeg = 90
63
64     LE 1500 <= val <= 1950:
65         windDir = "ESE"
66         windDeg = 112.5
67
68     LE 4500 <= val <= 4900:
69         windDir = "SE"
70         windDeg = 135
71
72     LE 3000 <= val <= 3500:
73         windDir = "SSE"
74         windDeg = 157.5
75
76     LE 7000 <= val <= 7500:
77         windDir = "S"
78         windDeg = 180
79
80     LE 6000 <= val <= 6500:
81         windDir = "SSW"
```

```

82     windDeg = 202.4
83
84     IF 16000 <= val <= 16500
85         windDir = "SW"
86         windDeg = 224
87
88     IF 15000 <= val <= 15500
89         windDir = "WSW"
90         windDeg = 247.4
91
92     IF 24000 <= val <= 24500
93         windDir = "W"
94         windDeg = 270
95
96     IF 21000 <= val <= 21500
97         windDir = "WNW"
98         windDeg = 292.4
99
100    IF 22500 <= val <= 23000
101        windDir = "NW"
102        windDeg = 314
103
104    IF 17500 <= val <= 18000
105        windDir = "NNW"
106        windDeg = 337.4
107
108    #Print the results
109    print( 'Temperature: ' , temperature)
110    print( 'Humidity:      ' , humidity, '%')
111    print( 'Pressure:      ' , pressure, 'kPa')
112    print( 'Wind Dir:      ' , windDir, ' °', windDeg, '°')
113    print( ' ')

```

- (9) Όπως ακριβώς κάναμε και στις άλλες βιβλιοθήκες, εισάγουμε τη βιβλιοθήκη Adafruit_ADS1x15
- (10) Φόρτωση της βιβλιοθήκης AnalogIn για το ADS1x15
- (13) Στη συνέχεια δημιουργούμε το αντικείμενο ADC και του δίνουμε ένα όνομα.

- (14) Ρυθμίστε τον προγραμματιζόμενο ενισχυτή κέρδους σε 1 στον ADC
- (39) Διαβάστε το κανάλι ADC 0 και αποθηκεύστε τη στη μεταβλητή "chan"
- (40) Χρειάζεστε μόνο την τιμή adc, οπότε «τραβήξτε» το από το chan και αντιστοιχίστε το σε μια νέα μεταβλητή "val" αφού τον πολλαπλασιάσετε με το 16. Αυτό απαιτείται λόγω αλλαγών στη βιβλιοθήκη Adafruit.
- (41-42) Ορίστε μια ένδειξη επιπέδου βάσης εάν ο αισθητήρας δεν είναι συνδεδεμένος
- (44-106) Ελέγξτε για να δείτε αν το val είναι μεταξύ δύο τιμών, εάν είναι, αντιστοιχίζουμε την κατεύθυνση, αλλιώς συνεχίζει κάτω από τη λίστα μέχρι να βρεθεί η τιμή που ταιριάζει
- (112) Εκτυπώστε τα αποτελέσματα

18.7.5 Βήμα 5: Ανάγνωση Αισθητήρα Ταχύτητας Ανέμου

Οι δύο τελευταίοι αισθητήρες είναι απλές ψηφιακές είσοδοι στο Raspberry Pi. Ο αισθητήρας ταχύτητας ανέμου είναι ουσιαστικά ένας μαγνήτης και ένας μικρός μαγνητικός διακόπτης. Καθώς ο αισθητήρας περιστρέφεται, ο μαγνήτης περνάει μπροστά από τον διακόπτη αναγκάζοντάς τον να «κλείσει». Άνεμος ταχύτητας 1,2km / h προκαλεί το άνοιγμα ή το κλείσιμο αυτού του διακόπτη μία φορά σε διάστημα ενός δευτερολέπτου. Αρκεί λοιπόν να μετρήσουμε πόσες φορές ο διακόπτης ανοίγει και κλείνει σε δευτερόλεπτο για να υπολογίσουμε την ταχύτητα τους ανέμου.

Το Raspberry Pi έχει έναν εύκολο τρόπο παρακολούθησης αυτού του είδους των δεδομένων. Ακόμα κι αν θέσαμε σε παύση ολόκληρο το πρόγραμμα για μια περίοδο 15 δευτερολέπτων, μπορούμε να δημιουργήσουμε μια διαδικασία παρασκηνίου η οποία θα εξακολουθεί να παρακολουθεί τις αλλαγές στον ακροδέκτη που είναι συνδεδεμένος ο αισθητήρας που μας ενδιαφέρει και θα συνεχίζει να μετράει ενώ ο κώδικας του προγράμματος μας βρίσκεται σε παύση.

Ας ξεκινήσουμε λοιπόν: Πρώτα πρέπει να εισαγάγουμε τη βιβλιοθήκη Raspberry Pi GPIO, να ρυθμίσουμε τον ακροδέκτη που χρησιμοποιούμε και να δημιουργήσουμε τη διαδικασία παρασκηνίου που παρακολουθεί τον ακροδέκτη. Στη συνέχεια θα υπολογίσουμε την ταχύτητα του ανέμου και, τέλος, θα εκτυπώσουμε το αποτέλεσμα.

```

1  import time
2  from w1thermsensor import W1ThermSensor
3
4  import board
5  import busio
6  import adafruit_bme280
7  i2c = busio.I2C(board.SCL, board.SDA)
8
9  import adafruit_ads1x15.ads1015 as ADS
10 from adafruit_ads1x15.analog_in import AnalogIn
11
12 import adafruit_bme280 as bme
```

```

13
14 bme = adafruit_bme280.Adafruit_BME280_I2C(i2c)
15 ads = ADS.ADS1015(i2c)
16 ads.gain = 1
17
18
19 ds18b20 = W1ThermSensor()
20
21 interval = 15 #How long we want to wait between loops (seconds)
22 windTick = 0 #Used to count the number of times the wind speed input
23 is triggered
24 #Set GPIO pins to use BCM pin numbers
25 GPIO.setmode(GPIO_BCM)
26
27 #Set digital pin 17 to an input and enable the pullup
28 GPIO.setup(17, GPIO.IN, pull_up_down=GPIO.PUD_UP)
29
30 #Event to detect wind (4 ticks per revolution)
31 GPIO.add_event_detect(17, GPIO.BOTH)
32 def windtrigger(self):
33     global windTick
34     windTick += 1
35
36 GPIO.add_event_callback(17, windtrigger)
37
38
39 while True:
40
41     time.sleep(interval)
42
43     #Pull Temperature from DS18B20
44     temperature = ds18b20.get_temperature()
45
46     #Pull temperature from BME280
47     case_temp = bme.temperature
48

```

```
49     #Pull pressure from BME280 Sensor & convert to kPa
50     pressure_pa = bme.pressure
51     pressure = pressure_pa / 10
52
53     #Pull humidity from BME280
54     humidity = bme.humidity
55
56     #Calculate wind direction based on ADC reading
57     chan = AnalogIn(ads, ADS.P0)
58     val = chan.value
59     windDir = "Not Connected"
60     windDeg = 999
61
62     if 20000 <= val <= 20500:
63         windDir = "N"
64         windDeg = 0
65
66     if 10000 <= val <= 10500:
67         windDir = "NNE"
68         windDeg = 22.5
69
70     if 11500 <= val <= 12000:
71         windDir = "NE"
72         windDeg = 45
73
74     if 2000 <= val <= 2250:
75         windDir = "ENE"
76         windDeg = 67.5
77
78     if 2300 <= val <= 2500:
79         windDir = "E"
80         windDeg = 90
81
82     if 1500 <= val <= 1950:
83         windDir = "ESE"
84         windDeg = 112.5
```

```
85
86     if 4500 <= val <= 4900:
87         windDir = "SE"
88         windDeg = 135
89
90     if 3000 <= val <= 3500:
91         windDir = "SSE"
92         windDeg = 157.5
93
94     if 7000 <= val <= 7500:
95         windDir = "S"
96         windDeg = 180
97
98     if 6000 <= val <= 6500:
99         windDir = "SSW"
100        windDeg = 202.5
101
102    if 16000 <= val <= 16500:
103        windDir = "SW"
104        windDeg = 225
105
106    if 15000 <= val <= 15500:
107        windDir = "WSW"
108        windDeg = 247.5
109
110    if 24000 <= val <= 24500:
111        windDir = "W"
112        windDeg = 270
113
114    if 21000 <= val <= 21500:
115        windDir = "WNW"
116        windDeg = 292.5
117
118    if 22500 <= val <= 23000:
119        windDir = "NW"
120        windDeg = 315
```

```

121
122     if 17500 <= val <= 18500:
123         windDir = "NNW"
124         windDeg = 337.5
125
126         #Calculate average windspeed over the last 15 seconds
127         windSpeed = (windTick * 1.2) / interval
128         windTick = 0
129
130         #Print the results
131         print( 'Temperature: ' , temperature)
132         print( 'Humidity:      ' , humidity, '%')
133         print( 'Pressure:      ' , pressure, 'kPa')
134         print( 'Wind Dir:      ' , windDir, ' (', windDeg, ')')
135         print( 'Wind Speed:    ' , windSpeed, 'KPH')
136         print( ' ')

```

- (12) Όπως και με τις υπόλοιπες βιβλιοθήκες, εισάγουμε τη βιβλιοθήκη GPIO.
- (22) Δημιουργούμε μια μεταβλητή για να αποθηκεύσουμε πόσες φορές ενεργοποιήθηκε ο αισθητήρας ανέμου
- (25) Ρυθμίζουμε τους ακροδέκτες GPIO
- (28) Ρυθμίζουμε ακροδέκτη 17 ως είσοδο και ενεργοποιήστε το pullup
- (31-36) Δημιουργούμε μια διαδικασία παρασκηνίου για να παρακολουθούμε πόσες φορές ο αισθητήρας 17 αλλάζει κατάσταση από HIGH σε LOW ή LOW σε HIGH, κάθε φορά που συμβαίνει αυτό αυξάνουμε κατά 1 την τιμή της μεταβλητής WindTick
- (127) Υπολογίζουμε την ταχύτητα του ανέμου - 1,2km / h ανά tick ανά δευτερόλεπτο. Αυτό θα έχει ως αποτέλεσμα την μέση ταχύτητα του ανέμου τα τελευταία 15 δευτερόλεπτα
- (128) Μηδενίζουμε τον μετρητή μας
- (135) Εκτυπώνουμε του αποτέλεσμα

Αφού ολοκληρώσετε την πληκτρολόγηση του κώδικα πατήστε το πλήκτρο “F5” για να δείτε το πρόγραμμα σας να εκτελείται. Το πρόγραμμα σας θα εμφανίσει την μέση ταχύτητα του ανέμου τα τελευταία 15 δευτερόλεπτα.

18.7.6 Βήμα 6: Ανάγνωση αισθητήρα βροχόπτωσης

Παρόμοια με τον αισθητήρα ταχύτητας ανέμου, ο αισθητήρας μέτρησης βροχής λειτουργεί με έναν μαγνήτη και έναν διακόπτη. Μέσα σε αυτόν τον αισθητήρα, μια μικρή συσκευή ανατροπής αλλάζει θέση (μπρος-πίσω) κάθε 0.2794 mm βροχής που συγκεντρώνει. Επομένως, θα πρέπει να μετρήσουμε τον αριθμό των αλλαγών θέσης αυτού του αισθητήρα και να πολλαπλασιάσουμε τον αριθμό των εναλλαγών για να υπολογίσουμε το ύψος της βροχόπτωσης.

Θα υλοποιήσουμε μία διαδικασία παρασκηνίου παρόμοια με αυτήν που υλοποιήσαμε για τον αισθητήρα ταχύτητας ανέμου. Σε αυτή την περίπτωση βέβαια αντί να μετράμε μεταβάσεις από υψηλό σε χαμηλό και χαμηλό σε υψηλό, θα μετράμε τον αριθμό των μεταβολών της θέσης του πείρου του αισθητήρα βροχόπτωσης και θα πολλαπλασιάσουμε με τον αντίστοιχο αριθμό για να υπολογίσουμε το ύψος της βροχόπτωσης στο χρονικό διάστημα των 15 δευτερολέπτων.

```
1  import time
2  from wlthermsensor import WlThermSensor
3
4  import board
5  import busio
6  import adafruit_bme280
7  i2c = busio.I2C(board.SCL, board.SDA)
8
9  import adafruit_ads1x15.ads1015 as ADS
10 from adafruit_ads1x15.analog_in import AnalogIn
11
12 import RPi.GPIO as GPIO
13
14 bme = adafruit_bme280.Adafruit_BME280_I2C(i2c)
15 ads = ADS.ADS1015(i2c)
16 ads.gain = 1
17
18
19 ds18b20 = WlThermSensor()
20
21 interval = 15 #How long we want to wait between loops (seconds)
22 windTick = 0 #Counts the times the wind speed input is triggered
23 rainTick = 0 #Counts the times the rain input is triggered
24
25 #Set GPIO pins to use BCM pin numbers
26 GPIO.setmode(GPIO.BCM)
27
28 #Set digital pin 17 to an input and enable the pullup
29 GPIO.setup(17, GPIO.IN, pull_up_down=GPIO.PUD_UP)
30
```

```

31 #Set digital pin 23 to an input and enable the pullup
32 GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_UP)
33
34 #Event to detect wind (4 ticks per revolution)
35 GPIO.add_event_detect(17, GPIO.BOTH)
36 def windtrig(self):
37     global windTick
38     windTick += 1
39
40 GPIO.add_event_callback(17, windtrig)
41
42 #Event to detect rainfall tick
43 GPIO.add_event_detect(23, GPIO.FALLING)
44 def raintrig(self):
45     global rainTick
46     rainTick += 1
47
48 GPIO.add_event_callback(23, raintrig)
49
50
51 while True:
52
53     time.sleep(interval)
54
55     #Pull Temperature from DS18B20
56     temperature = ds18b20.get_temperature()
57
58     #Pull temperature from BME280
59     case_temp = bme.temperature
60
61     #Pull pressure from BME280 Sensor & convert to kPa
62     pressure_pa = bme.pressure
63     pressure = pressure_pa / 10
64
65     #Pull humidity from BME280
66     humidity = bme.humidity

```



```
67
68     #Calculate wind direction based on ADC reading
69     chan = AnalogIn(ads, ADS.P0)
70     val = chan.value
71     windDir = "Not Connected"
72     windDeg = 999
73
74     if 20000 <= val <= 20500:
75         windDir = "N"
76         windDeg = 0
77
78     if 10000 <= val <= 10500:
79         windDir = "NNE"
80         windDeg = 22.5
81
82     if 11500 <= val <= 12000:
83         windDir = "NE"
84         windDeg = 45
85
86     if 2000 <= val <= 2250:
87         windDir = "ENE"
88         windDeg = 67.5
89
90     if 2300 <= val <= 2500:
91         windDir = "E"
92         windDeg = 90
93
94     if 1500 <= val <= 1950:
95         windDir = "ESE"
96         windDeg = 112.5
97
98     if 4500 <= val <= 4900:
99         windDir = "SE"
100         windDeg = 135
101
102     if 3000 <= val <= 3500:
```

```
103         windDir = "SSE"
104         windDeg = 157.5
105
106     if 7000 <= val <= 7500:
107         windDir = "S"
108         windDeg = 180
109
110     if 6000 <= val <= 6500:
111         windDir = "SSW"
112         windDeg = 202.5
113
114     if 16000 <= val <= 16500:
115         windDir = "SW"
116         windDeg = 225
117
118     if 15000 <= val <= 15500:
119         windDir = "WSW"
120         windDeg = 247.5
121
122     if 24000 <= val <= 24500:
123         windDir = "W"
124         windDeg = 270
125
126     if 21000 <= val <= 21500:
127         windDir = "WNW"
128         windDeg = 292.5
129
130     if 22500 <= val <= 23000:
131         windDir = "NW"
132         windDeg = 315
133
134     if 17500 <= val <= 18500:
135         windDir = "NNW"
136         windDeg = 337.5
137
138     #Calculate average windspeed over the last 15 seconds
```

```

139     windSpeed = (windTick * 1.2) / interval
140     windTick = 0
141
142     #Calculate accumulated rainfall over the last 15 seconds
143     rainfall = rainTick * 0.2794
144     rainTick = 0
145
146     #Print the results
147     print( 'Temperature: ' , temperature)
148     print( 'Humidity:      ' , humidity, '%')
149     print( 'Pressure:      ' , pressure, 'kPa')
150     print( 'Wind Dir:      ' , windDir, ' (', windDeg, ')')
151     print( 'Wind Speed:    ' , windSpeed, 'KPH')
152     print( 'Rainfall:      ' , rainfall, 'mm')
153     print( ' ')

```

- (23) Δημιουργήστε μια μεταβλητή για να αποθηκεύσετε τον αριθμό των φορών που ενεργοποιείται ο πείρος του αισθητήρα βροχής
- (32) Ρυθμίστε τον ακροδέκτη 23 ως είσοδο και ενεργοποιήστε το pullup
- (42-48) Δημιουργήστε μια διαδικασία παρασκηνίου για να παρακολουθείτε πόσες φορές ο ακροδέκτης 23 αλλάζει κατάσταση από ΥΨΗΛΗ σε ΧΑΜΗΛΗ, Κάθε φορά που συμβαίνει αυτό αυξάνουμε κατά 1 την μεταβλητή rainTick
- (143) Υπολογίζουμε το ύψος της βροχόπτωσης - 0,2794 mm για κάθε tick. Το αποτέλεσμα θα αφορά στην συνολική βροχόπτωση για το διάστημα των τελευταίων 15 δευτερολέπτων.
- (144) Μηδενίζουμε τον μετρητή βροχόπτωσης
- (152) Εκτυπώνουμε το αποτέλεσμα

18.7.7 Βήμα 7: Όλα έτοιμα!!!

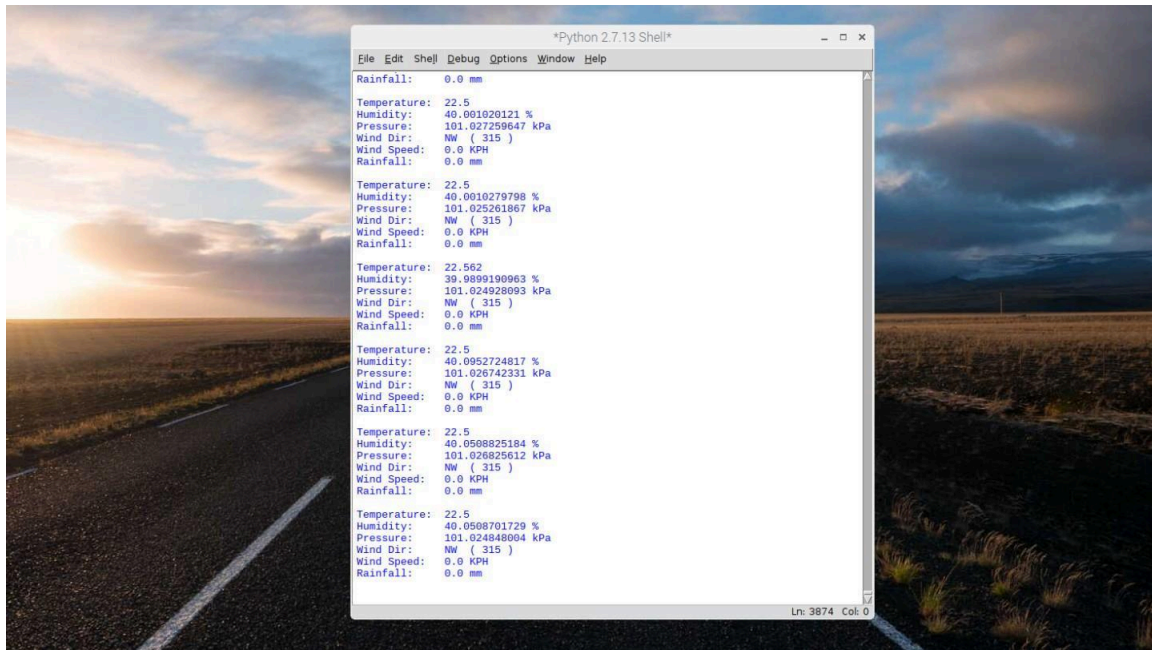


Figure 24. Οι τιμές θερμοκρασίας, υγρασίας, ατμοσφαιρικής πίεσης, ταχύτητας και κατεύθυνσης ανέμου, ύψους βροχής

Αυτός είναι ο κώδικας μας για την ώρα! Εκτελέστε το πρόγραμμα που έχουμε δημιουργήσει μαζί για να δείτε τις τιμές των αισθητήρων κάθε 15 δευτερόλεπτα.

Τα αποτελέσματα θα εμφανίζονται όπως στην παραπάνω εικόνα

19. Βιβλιογραφικές Αναφορές

Afari, E., & Khine, M.S. (2017). *Robotics as an educational tool: Impact of Lego Mindstorms*, International Journal of Information and Education Technology, 7(6), 437-442.

Alimisis, D. (2012). *Robotics in Education & education in robotics: Shifting focus from technology to pedagogy*. Paper presented at the robotics in Education, Praha.

Altin, H., Pedaste, M. (2013). *Learning Approaches to applying robotics in science education*, Journal of Baltic Science Education vol 12, (3) 365-377.

Johnson, J. (2003). *Children, robotics and education*. Artificial Life and Robotics, 7(1), 16-21.

Kafai, Y., Resnick, M. (1996). *Constructionism in practice : Designing, Thinking and Learning in a Digital World LE*, Lawrence Erlbaum Associates, New Jersey.

MacDonald, S., Howwell, J. (2012). *Watching, creating and achieving: Creative technologies as a conduit for learning in the early years*. British journal of educational technology, 43(4), 641-651.

Papert, S. (1980). *Children, Computers and powerful ideas*. Basic Books Inc., New York

Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. New York: Basic books.

Resnick, M. (2002). *Rethinking Learning in the Digital Age*. In G. Kirkman (Ed.), The Global Information Technology Report: Readiness for the Networked World (pp. 32-37). Oxford: Oxford University Cross.

Sullivan, F.R. (2008). *Robotics and science Literacy : thinking skills, science process skills and systems understanding*. Journal of Research in Science Teaching, 45(3), 373-394.

Williams, D., et al (2007). *Acquisition of physics content knowledge and scientific inquiry skills in a robotics summer camp*. Journal of Research on Technology in Education, 40(2), 201-216.