



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Σχεδιασμός και ανάπτυξη συστήματος μέτρησης
καρδιακών παλμών και κορεσμού οξυγόνου με τη
χρήση Raspberry»



Φοιτητής
Γεωργιάδης Χαράλαμπος
Αρ. Μητρώου: 154438

Επιβλέπουσα
Παπαδοπούλου Μαρία
Επίκουρη Καθηγήτρια

10 Φεβρουαρίου 2024

Τίτλος Π.Ε:

«Σχεδιασμός και ανάπτυξη συστήματος μέτρησης καρδιακών παλμών και κορεσμού οξυγόνου με τη χρήση Raspberry»

Κωδικός Π.Ε. 23173

Γεωργιάδης Χαράλαμπος

Εισηγήτρια: Παπαδοπούλου Μαρία

Ημερομηνία ανάληψης Π.Ε: 26-03-2023

Ημερομηνία περάτωσης Δ.Ε. 10-02-2024

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Γεωργιάδη Χαράλαμπου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιοδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Στην οικογένειά μου»

Πρόλογος

Κατά την διάρκεια της πανδημίας του Covid 19, ολόκληρος ο πλανήτης βρέθηκε σε μια κρίσιμη κατάσταση, όσον αφορά τον τομέα της υγείας. Σε μια κατάσταση που ο κάθε άνθρωπος είχε τον δικό του ρόλο να συνεισφέρει με τον καλύτερο δυνατό τρόπο, εμείς, οι άνθρωποι της τεχνολογίας, έπρεπε να βρούμε έξυπνες λύσεις ώστε να βοηθήσουμε με τον τρόπο και τις γνώσεις μας την κοινωνία να αντιμετωπίσει αποτελεσματικότερα την εξάπλωση του ιού και να είναι πιο προετοιμασμένη για τυχόν παρόμοιες συνθήκες στο μέλλον. Είδαμε ότι αναπτύχθηκαν πολλά και διαφορετικά πληροφοριακά συστήματα ώστε να διευκολύνουν τους πολίτες προκειμένου να μειωθούν οι μετακινήσεις τους και να ελαχιστοποιήσουν το ρίσκο διασποράς ή και νόσησης από τον ιό.

Ωστόσο, η τεχνολογία έχει εξελιχθεί σε σπουδαίο βαθμό για αυτό και δεν πρέπει να επαναπαυόμαστε σε λύσεις που ήδη υπάρχουν, αλλά να προσπαθούμε συνεχώς να βελτιώσουμε και να αναβαθμίσουμε τα συστήματα μας ώστε να έχουμε αποτελεσματικότερη πρόληψη και θεραπεία σε ανάλογες συνθήκες. Η πανδημία του Covid 19 ανάγκασε τους ανθρώπους της τεχνολογίας να σκεφτούν και να υλοποιήσουν συστήματα που βελτιώνουν τη καθημερινότητα των πολιτών. Έτσι λοιπόν και εμένα, μου έδωσε την ιδέα να δημιουργήσω ένα σύστημα παρακολούθησης των καρδιακών παλμών και του κορεσμού του οξυγόνου στο αίμα.

Περίληψη

Ο τομέας της υγείας αναζητεί συνεχώς νέες και αποδοτικότερες λύσεις, ώστε να διευκολύνεται το υγειονομικό προσωπικό αλλά και να βελτιώνεται η υγεία και η ποιότητα ζωής των ασθενών. Ειδικότερα στην περίοδο της πανδημίας του COVID-19 είδαμε πολλά κενά στις παροχές υγείας από κράτη που όλοι θεωρούμε πλήρως ανεπτυγμένα.

Αυτό ίσως και να ήταν το καμπανάκι που έπρεπε να χτυπήσει ώστε όλοι να συνειδητοποιήσουμε πως ειδικά στο θέμα της υγείας θα πρέπει συνεχώς να βελτιωνόμαστε και να σκεφτόμαστε περισσότερους, αποτελεσματικότερους και πιο ευφυείς τρόπους αντιμετώπισης των πραγμάτων.

Έτσι λοιπόν, στην παρούσα πτυχιακή θα αναπτύξουμε πως μπορεί να υλοποιηθεί μια συσκευή μέτρησης των καρδιακών παλμών και του κορεσμού οξυγόνου του αίματος, ώστε αυτές οι μετρήσεις να παρακολουθούνται από τον γιατρό χρησιμοποιώντας απλά το κινητό του τηλέφωνο, μέσω του πρωτοκόλλου Bluetooth, ώστε να είναι σε θέση να αξιολογεί την κατάσταση της υγείας του ασθενή και να επεμβαίνει όποτε το θεωρεί απαραίτητο.

«Design and development of a heart rate and oxygen saturation measurement system using Raspberry»

«Charalampos Georgiadis»

Abstract

The health sector is constantly looking for new and more efficient solutions, in order to facilitate the health staff and also to improve the health and quality of life of the patients. In particular, during the period of the COVID-19 pandemic we have seen many gaps in health services from states that we all consider fully developed.

This might have been the bell that had to ring so that we all realize that especially in the matter of health, we should constantly improve and think of more, more efficient and more intelligent ways.

So, in this thesis we will develop how a device can be implemented to measure heart rates and blood oxygen saturation, so that these measurements can be monitored by the doctor simply using his mobile phone, through the Bluetooth protocol, so that able to assess the patient's health status and intervene whenever he deems it necessary.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά όλους του καθηγητές μου και τους συμφοιτητές μου που σε αυτά τα φοιτητικά χρόνια διαμόρφωσαν σε σημαντικό βαθμό την πορεία μου ως επαγγελματίας αλλά κυρίως και ως άνθρωπο. Ειδικότερα θα ήθελα να πω και ένα μεγάλο ευχαριστώ στην οικογένεια μου, που όλα αυτά τα χρόνια μου έδινε δύναμη και αντοχές ώστε να μπορέσω να ανοίξω τον δρόμο μου για την επαγγελματική μου σταδιοδρομία. Τέλος, θα ήθελα να ευχαριστήσω από καρδιάς και την επιβλέπουσα καθηγήτρια μου, που ήταν πάντα διαθέσιμη και πρόθυμη να μου λύσει απορίες και να με καθοδηγήσει με συνέπεια και υπομονή.

Περιεχόμενα

Πρόλογος.....	iv
Περίληψη	v
Abstract	vi
Ευχαριστίες	vii
Περιεχόμενα	viii
Κατάλογος Εικόνων.....	x
Συντομογραφίες.....	xii
1. Εισαγωγή.....	13
1.1 Εισαγωγή	13
1.2 Δομή της εργασίας.....	13
2. Raspberry Pi	14
2.1 Εισαγωγή	14
2.2 Τι είναι το Raspberry Pi.....	14
2.2.1 Λεπτομερής περιγραφή των Raspberry Pi συσκευών.....	14
2.2.2 Ιστορική αναδρομή	15
2.2.3 Οι δημιουργοί του Raspberry Pi.....	16
2.2.4 Το χρονοδιάγραμμα των Raspberry Pi model A και B.....	17
2.3 Raspberry pi 3 Model B+.....	17
3. Ο αισθητήρας MAX30102	19
3.1 MAX30102 Module Hardware Overview.	19
3.2 Απαιτήσεις ισχύος	19
3.3 On-Chip Temperature Sensor.....	20
3.4 I2C Interface	20
3.5 FIFO Buffer.....	21
3.6 Interrupts	21
3.7 Technical Specifications	21
3.8 Λειτουργία αισθητήρα MAX30102.....	22
3.8.1 Hear Rate Measurement.....	22

3.8.2	Pulse Oximetry	23
4.	Software Details.....	24
4.1	Εισαγωγή	24
4.2	Η γλώσσα προγραμματισμού Python	24
4.3	Ιστορική αναδρομή της Python.....	24
4.4	Χαρακτηριστικά και φιλοσοφία της Python	25
4.5	Πλεονεκτήματα της Python	26
4.6	Βέλτιστες πρακτικές και συμβουλές.....	27
5.	Raspberry Pi 3 Model B & MAX30102.....	29
5.1	Συνδεσιμότητα	29
5.2	Ενεργοποίηση Bluetooth κατά την εκκίνηση.....	30
5.3	Λειτουργία κουμπιού.....	30
5.4	Εκτύπωση μετρήσεων.....	31
5.4.1	Δυνατότητα εκτύπωσης σε παραπάνω από μια συσκευή.....	34
6.	Αποτελέσματα μετρήσεων και συμπεράσματα.....	37
6.1	MAX30102 συμβουλές και πρακτικές	37
6.2	Σύγκριση με Xiaomi 13 Lite	38
6.3	Σύγκριση με Smartwatch BlackView X1	39
6.4	Σύγκριση με οξύμετρο YM101	40
6.5	Σύγκριση με οξύμετρο Esperanza	41
6.6	Συμπεράσματα	42
7.	BIBΛΙΟΓΡΑΦΙΑ	43
8.	ΠΑΡΑΡΤΗΜΑ Α: BaseHeartRateSPO2.py	45
9.	ΠΑΡΑΡΤΗΜΑ Β: gain_heartbeat_SPO2.py.....	48
10.	ΠΑΡΑΡΤΗΜΑ Γ: Button.py	51

Κατάλογος Εικόνων

Εικόνα 0.1: Raspberry Pi 3

Εικόνα 0.2: Led αισθητήρα MAX30102

Εικόνα 0.3: LDO Regulators

Εικόνα 3.2.2 Logic Level Selection Jumper

Εικόνα 3.7: Τεχνικά Χαρακτηριστικά

Εικόνα 3.8: Λειτουργία Photodetector και LED

Εικόνα 3.8.1: Μέτρηση καρδιακών παλμών

Εικόνα 3.8.2: Απορρόφηση HbO₂ και Hb

Εικόνα 5.1: Συνδεσιμότητα RPI με MAX30102

Εικόνα 5.1.1: Ενεργοποίηση I2C Interface

Εικόνα 5.2: Systemd service: rfcomm watch hci0

Εικόνα 5.3: Momentary button

Εικόνα 5.3.1: button.desktop file

Εικόνα 5.4: Serial Bluetooth Terminal by Kai Morich

Εικόνα 5.4.1: Εκτύπωση μετρήσεων στο κινητό

Εικόνα 5.4.2: Εκτύπωση μετρήσεων measures.txt

Εικόνα 5.4.3: Εικονική αναπαράσταση 1^{ου} βήματος

Εικόνα 5.4.4: Εικονική αναπαράσταση 2^{ου} βήματος

Εικόνα 5.4.5: Εικονική αναπαράσταση 3^{ου} βήματος

Εικόνα 5.4.6: Εικονική αναπαράσταση 4^{ου} βήματος

Εικόνα 5.4.7: Εικονική αναπαράσταση 5^{ου} βήματος

Εικόνα 6.2: Xiaomi 13 Lite vs MAX30102

Εικόνα 6.3: BlackView X1 vs MAX30102

Εικόνα 6.4: YM101 vs MAX30102

Εικόνα 6.5: Esperanza vs MAX30102

Εικόνα 6.5.1: Συγκριτικός πίνακας μετρήσεων

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
RPI	Raspberry Pi
PSF	Python Software Foundation

1. Εισαγωγή

1.1 Εισαγωγή

Η παρούσα πτυχιακή εργασία, όπως αναφέραμε και στον πρόλογο, προήλθε από μια σειρά ζητημάτων που βγήκαν στην επιφάνεια μετά την πανδημία του Covid-19. Το συμπέρασμα, από την περιπέτεια που ζήσαμε το 2020, είναι ότι σε μια εποχή που η τεχνολογία συνεχώς αναπτύσσεται, δυστυχώς στον τομέα της υγείας, για την επίβλεψη και πρόληψη της υγείας των πολιτών είμαστε σε μια θέση που θα πρέπει να κάνουμε όχι απλά βήματα αλλά άλματα προόδου, ώστε να συμβαδίσουμε με την εποχή.

Προφανώς, δεν είμαστε οι ειδικοί και οι κατάλληλοι να επιρρίψουμε ευθύνες αλλά ίσως είμαστε αυτοί που μπορούμε να βοηθήσουμε με τις ιδέες μας ώστε η τεχνολογία να εναρμονιστεί άψογα με την υγεία. Ξεκινώντας λοιπόν από απλές αλλά εξίσου σημαντικές λύσεις, φτάνουμε στην παρούσα πτυχιακή όπου αναλύουμε πώς με τη χρήση του υπολογιστή Raspberry σε συνδυασμό με συμβατούς αισθητήρες μπορούμε πολύ εύκολα να δημιουργήσουμε διαφορετικών λειτουργιών συσκευές ώστε να διευκολύνουμε σε μεγάλο βαθμό τους πολίτες αλλά και τους υγειονομικούς μας, οι οποίοι, όπως όλοι είδαμε στην περίοδο της πανδημίας, κατέβαλλαν ηρωικές προσπάθειες να αντιμετωπίσουν τον μεγάλο αριθμό ανθρώπων που χρειαζόταν τη βοήθεια τους.

Στη συγκεκριμένη πτυχιακή εργασία λοιπόν, δημιουργήσαμε μια συσκευή μέτρησης καρδιακών παλμών και κορεσμού οξυγόνου του αίματος χρησιμοποιώντας τον υπολογιστή Raspberry Pi 3 Model B, ενσωματώνοντας πάνω σε αυτόν τον αισθητήρα MAX30102.

Συγκεκριμένα, όχι απλώς μετράμε τους καρδιακούς παλμούς και τον κορεσμό του οξυγόνου του αίματος, αλλά εκμεταλλευόμαστε πλήρως μια από τις πολλές σημαντικές λειτουργίες του Raspberry Pi 3 Model B και φυσικά μιλάμε για το Bluetooth μέσω του οποίου ο γιατρός από το κινητό του μπορεί οποιαδήποτε στιγμή να ελέγχει τις μετρήσεις του αισθητήρα και να επεμβαίνει όποτε αυτό κρίνει πως είναι απαραίτητο.

1.2 Δομή της εργασίας

Στο πρώτο κεφάλαιο, κάνουμε μια απλή αναφορά στην υλικά που χρησιμοποιήσαμε όπως επίσης και μια μικρή εισαγωγή αναλύοντας λίγο περισσότερο την σκέψη πίσω από το συγκεκριμένο θέμα της πτυχιακής εργασίας. Τέλος, στο πρώτο κεφάλαιο μπορούμε να διακρίνουμε και την δομή που θα έχει η συγκεκριμένη πτυχιακή εργασία.

Στο δεύτερο κεφάλαιο, αναλύουμε εκτεταμένα όλο το υλικό που χρησιμοποιήθηκε. Από τον υπολογιστή Raspberry και όλες τις απαραίτητες πληροφορίες που πρέπει να ξέρουμε για αυτόν. Στο τρίτο κεφάλαιο θα περιγράψουμε τι είναι ο αισθητήρας MAX30102, που χρησιμοποιήθηκε και πώς λειτουργεί.

Στο τέταρτο κεφάλαιο, θα μιλήσουμε για το software που χρησιμοποιήθηκε στην παρούσα υλοποίηση. Θα γίνει αναλυτική περιγραφή της γλώσσας προγραμματισμού python, ενώ στο πέμπτο κεφάλαιο πλέον, παρουσιάζεται η συσκευή μέτρησης καρδιακών παλμών και κορεσμού οξυγόνου του αίματος παραθέτοντας επίσης φωτογραφίες αλλά και τον python κώδικα που χρησιμοποιήσαμε.

Στο έκτο και τελευταίο κεφάλαιο, θα παρατηρήσουμε την σύγκριση μετρήσεων του αισθητήρα με άλλες συσκευές, λεπτομέρειες που ίσως μας βοηθήσουν στην καλύτερη εξαγωγή μετρήσεων από τον αισθητήρα αλλά και τα συμπεράσματα που βγήκαν μέσα από το project μας σχετικά με την αποδοτικότητα και την ορθότητα των μετρήσεων του αισθητήρα.

2. Raspberry Pi

2.1 Εισαγωγή

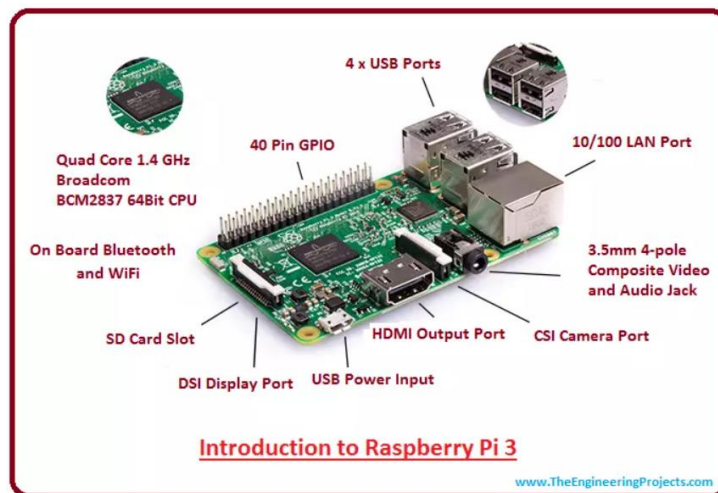
Στο συγκεκριμένο κεφάλαιο θα κάνουμε μια αναλυτική περιγραφή – παρουσίαση του hardware που χρησιμοποιήθηκε για την εκπόνηση αυτής της πτυχιακής εργασίας μελετώντας το κάθε κομμάτι ξεχωριστά παραθέτοντας βασικά χαρακτηριστικά και πληροφορίες απαραίτητες ώστε να τα μάθουμε και να κατανοήσουμε σε μεγάλο βαθμό.

2.2 Τι είναι το Raspberry Pi

Το Raspberry Pi είναι ένας αρκετά φθηνός μικρός υπολογιστής, σε μέγεθος λίγο μεγαλύτερος από μια πιστωτική κάρτα, όπως φαίνεται και στην παρακάτω εικόνα. Από το 2012, που βγήκε στην αγορά, έχει γίνει περιζήτητος, όχι μόνο για το μικρό κόστος και μέγεθος, αλλά κυρίως για τις απίστευτα πολλές δυνατότητες που προσφέρει. [1]

Αρχικά, η ιδέα ήταν να βγουν λίγα κομμάτια για παραγωγή και να επικεντρωθεί η προώθηση τους σε σχολεία, ώστε να κεντρίσουν το ενδιαφέρον των μαθητών και να ασχοληθούν όλο και περισσότερο με την επιστήμη των υπολογιστών και του προγραμματισμού. [1] Σαφώς, και η μεγάλη ζήτηση του συγκεκριμένου μικρό υπολογιστή ξάφνιασε ακόμα και τους δημιουργούς του και δημιούργησε την ανάγκη για αρκετά μεγαλύτερη παραγωγή. [2]

Αρκεί μόνο να σκεφτούμε πως η αρχική ιδέα ήταν να κατασκευαστούν γύρω στα 10.000 κομμάτια, ενώ στην πραγματικότητα σε λίγο περισσότερο από έναν χρόνο είχαν πουληθεί σχεδόν 1.000.000 R Pis. [1]



Εικόνα 2.1: Raspberry Pi 3 [6]

2.2.1 Λεπτομερής περιγραφή των Raspberry Pi συσκευών.

Τα R Pis ουσιαστικά μετά την εκτόξευση της ζήτησης τους κατηγοριοποιήθηκαν σε τρία μοντέλα, το Raspberry Pi A, B και Zero. Υπάρχει και ένα τέταρτο το Raspberry Compute Module το οποίο χρησιμοποιείται κυρίως σε βιομηχανικές εφαρμογές. [1] [2]

Οι βασικές αρχές των παραπάνω τριών μοντέλων RPIs είναι αρκετά κοινές. Όλα διαθέτουν ένα system-on-a-chip με ενσωματωμένη κεντρική μονάδα επεξεργασίας CPU (central process unit), ενσωματωμένη μονάδα επεξεργασίας γραφικών GPU (graphics processing unit), ενσωματωμένη μνήμη και είσοδο ρεύματος ισχύος 5V. Όλα τα μοντέλα διαθέτουν επίσης θύρα για σύνδεση αποκλειστικής κάμερας όπως επίσης και μια σειρά ακροδεκτών εισόδου/εξόδου γενικής χρήσης (GPIO) όπου μπορούν να χρησιμοποιηθούν για ένα ευρύ φάσμα ηλεκτρονικών περιφερειακών όπως φώτα led, κουμπιά, κάμερες, μικροκινητήρες καθώς και μια τεράστια γκάμα από αισθητήρες. [1] [3] [4]

Τα περισσότερα μοντέλα διαθέτουν θύρα ethernet αλλά και WiFi, Bluetooth. Σε κάθε μοντέλο μπορεί να εφαρμοστεί πάνω στα GPIO pins και μια ειδική πλακέτα επέκτασης η οποία ονομάζεται HAT (hardware attached on top) προσφέροντας ένα μέσο να χρησιμοποιηθούν λειτουργίες όπως αναγνώριση ραδιοσυχνότητων, διαχείριση ενέργειας, διαχείριση κινητήρων και εγγραφή ήχου υψηλής ποιότητας προσφέροντας στους Raspberry Pi υπολογιστές τεράστια ευελιξία. [1] [3] [4]

Όπως αναφέραμε το Raspberry Pi είναι ένας μικρός υπολογιστής. Συνεπώς, προσφέρει ότι και ένας κανονικός υπολογιστής όπως σύνδεση πληκτρολογίου, ποντικιού και οθόνης χωρίς καμία παραμετροποίηση ώστε να μπορεί ο χρήστης να έχει πρόσβαση σε ένα φιλικό περιβάλλον λειτουργικού συστήματος είτε πρόκειται για Linux είτε για Windows 10 IoT είτε για Android. [4] Φυσικά, το Raspberry Pi μπορεί να χρησιμοποιηθεί και ως «ακέφαλη» μονάδα όπου μπορεί απομακρυσμένα να ελεγχθεί και να προγραμματιστεί από μια μεγάλη γκάμα υποστηριζόμενων γλωσσών προγραμματισμού εκτελώντας με αυτόν τον τρόπο αυτοματοποιημένες λειτουργίες μέσω διαφορετικών scripts. [1] [3]

2.2.2 Ιστορική αναδρομή

Όπως ήδη αναφέρθηκε, το πρώτο RPI βγήκε στην αγορά το 2012, η ιδέα όμως ήρθε αρκετά χρόνια νωρίτερα, το 2006, από το πανεπιστήμιο του Cambridge. Οι καθηγητές πληροφορικής που εργαζόνταν στο συγκεκριμένο project ήταν οι: Rob Mullins, Jack Lang, Alan Mycroft και ο Eben Upton, η σημαντικότερη μορφή για την δημιουργία του Raspberry Pi. [3] [4]

Με επικεφαλής λοιπόν τον Eben Upton όλη η ομάδα επικεντρώθηκε στον εξής στόχο: Όλοι οι νέοι μαθητές θα πρέπει να είναι εξοικειωμένοι με τις τεχνικές πτυχές της πληροφορικής. [3] Σημαντική ανησυχία για ολόκληρη την ομάδα ήταν πως έβλεπαν ότι όλα τα προγράμματα της πληροφορικής, είχαν αρχίσει να δίνουν έμφαση στην χρησιμοποίηση των υπολογιστών και όχι στην κατανόηση των μηχανισμών πίσω από αυτούς. [3] [4]

Εκείνη τη περίοδο η ομάδα επικεντρωνόταν σε φοιτητές πανεπιστημιακού επιπέδου, σύντομα όμως, διεύρυναν το πεδίο εφαρμογής τους για να συμπεριλάβουν και παιδιά σχολικής ηλικίας. Παρακινούμενοι λοιπόν από τον στόχο τους, ώστε να το κάνουν όσο πιο εύκολο γίνεται να αποκτηθούν οι γνώσεις γύρω από τους υπολογιστές, δημιούργησαν το ίδρυμα Raspberry Pi. [3] Εργάστηκαν έξι έτη μέχρι που κατάφεραν να δημιουργήσουν μια φθηνή και προσβάσιμη, σε επίπεδο κόστους συσκευή, που θα μπορούσε να χρησιμοποιηθεί σε σχολεία για τη διδασκαλία προγραμματισμού. Ο αιώτερος στόχος τους ήταν να κατασκευάσουν έναν υπολογιστή που θα μπορούσε να χρησιμοποιηθεί για να διδάξει του μαθητές πως πραγματικά λειτουργεί η πληροφορική.

Την περίοδο που αναπτύχθηκε ο υπολογιστής Raspberry Pi υπήρχε μια τάση στους κατασκευαστές υπολογιστών να ονομάζουν τα προϊόντα τους και τις συσκευές τους με ονόματα από φρούτα. Αδιαμφισβήτητα, το πιο διάσημο παράδειγμα είναι αυτό της Apple. Την ίδια πορεία έπειτα ακολούθησε και η Blackberry. [3]

Η ομάδα του Eben Upton ακολούθησε αυτή τη τάση και ονόμασαν τον υπολογιστή τους από το ταπεινό αλλά γλυκό μούρο (Raspberry) όπως είχε ο ίδιος δηλώσει.

Και πώς ακολουθεί το Pi μετά το Raspberry; Αυτό ουσιαστικά είναι απλό. Το Pi αναφέρεται στην γλώσσα προγραμματισμού του RPi, την Python. Φυσικά, είναι και ένα νεύμα στην μαθηματική έννοια του π (pi). [3]

2.2.3 Οι δημιουργοί του Raspberry Pi

Στην παιδική του ηλικία, ο Eben Urton ήταν μεγάλος θαυμαστής του BBC Micro, ενός από τους δύο υπολογιστές που είχαν κατασκευαστεί από το Cambridge την δεκαετία του 1980. Ο δεύτερος ήταν ο Sinclair Spectrum. Αυτοί οι υπολογιστές έπαιξαν μεγάλο ρόλο στην απόφαση των επόμενων γενεών να γίνουν προγραμματιστές. Ο BBC Micro εκτός από το έναυσμα που έδωσε στον Eben Urton να αγαπήσει τον προγραμματισμό τον έκανε επίσης ενθουσιώδη για τους υπολογιστές μικρού κόστους στους οποίους θα μπορούσαν να έχουν πρόσβαση οι νέοι. [3] [4]

Μάλιστα, ο Eben Urton είχε δηλώσει πως εκείνη τη περίοδο οι συζητήσεις για να φτιαχτεί και πάλι ο BBC Micro υπολογιστής με σκοπό να ενδιαφερθούν όλο και περισσότεροι νέοι να γίνουν προγραμματιστές, συνεχώς αυξάνονταν. [5]

Ωστόσο, ο σχεδιασμός ενός ισχυρού αλλά ταυτόχρονα χαμηλού κόστους υπολογιστή απαιτούσε μια ολόκληρη ομάδα να εργαστεί πάνω σε αυτό η οποία και βρέθηκε με τους Robert Mullins, Alan Mycroft και τον επιχειρηματία Jack Lang. Η ομάδα συνεργάστηκε και δημιούργησε τέσσερις βασικούς κανόνες που θα έπρεπε να ακολουθήσουν για να πετύχουν τον στόχο τους. [3] [4]

1. Να είναι ένα προγραμματιζόμενο κομμάτι υλικού. [3] [4]
2. Να είναι διασκεδαστικό. Η ομάδα ήθελε να υιοθετήσει το πνεύμα της δεκαετίας του '80, όταν είχαν έρθει για πρώτη φορά σε επαφή με τους υπολογιστές μέσα από παιχνίδια και προγραμματισμό. Ήθελαν το βασικό στοιχείο του RPi να είναι η διασκέδαση που θα πρόσφερε μέσω της εκμάθησης των υπολογιστών και του προγραμματισμού. [3] [4]
3. Να έχει προσιτή τιμή. Σε αυτόν τον κανόνα ήταν ανένδοτοι. Ήθελαν όλοι να μπορούν να έχουν πρόσβαση στον υπολογιστή, για αυτό τον λόγο είχαν συμφωνήσει και στην τιμή των 25\$, καθιστώντας το κόστος προσιτό για τα παιδιά της σχολικής ηλικίας αλλά και εύκολο για τα σχολεία να το επιδοτούν σε οικογένειες που δεν είχαν την οικονομική δυνατότητα. [3] [4]
4. Να είναι ανθεκτικό. Καθώς η συσκευή απευθυνόταν σε παιδιά και νέους έπρεπε να μπορεί να αντέχει στις μετακινήσεις και την καθημερινότητα. «Θέλαμε πολύ τα παιδιά να έχουν το δικό τους Raspberry Pi, και αυτό σημαίνει πως θα πρέπει να αντέχει να το βγάλουν και να το βάλουν στην τσάντα τους χίλιες φορές» είχε πει ο Eben Urton. [3] [4]

Όταν η ομάδα κατάφερε και πέτυχε και τους τέσσερις κανόνες στην συσκευή τους ήταν το 2012 όπου κυκλοφόρησε και το πρώτο Raspberry Pi. Και ήταν το μοναδικό στο είδος του. Μέχρι τότε δεν υπήρχε τίποτα αντίστοιχο σε αυτή τη τιμή και με αυτές τις δυνατότητες. Μια συσκευή που μπορούσε άνετα να υποστηρίξει ένα λειτουργικό σύστημα όπως το Linux. Οι στόχοι και οι προθέσεις τους είχαν γίνει πλέον πιο ξεκάθαροι από ποτέ. Μια συσκευή προσβάσιμη από κάθε άνθρωπο, κάθε ηλικίας και υπόβαθρου που θα έδινε την ευκαιρία να ασχοληθεί με την επιστήμη των υπολογιστών και τον προγραμματισμό. Αν μη τι άλλο μια πολύ πρωτοποριακή και δημοκρατική πρωτοβουλία για εκείνη την εποχή. Το 2013, μόλις ένα χρόνο μετά την κυκλοφορία του, κέρδισε το INDEX Design Award. Στη συνέχεια, το 2017, κέρδισε το βραβείο MacRobert της Royal Academy of Engineering. [4]

2.2.4 Το χρονοδιάγραμμα των Raspberry Pi model A και B

Το 2006 ξεκίνησαν οι πρώτες εργασίες για το Raspberry Pi στο πανεπιστήμιο του Cambridge από τους Eben Upton, Rob Mullins, Jack Lang και Alan Mycroft. Έξι χρόνια αργότερα και συγκεκριμένα τον Φεβρουάριο του 2012 πραγματοποιήθηκε η πρώτη εμπορική κυκλοφορία του Raspberry Pi. [4]

Λίγους μόνο μήνες αργότερα, τον Σεπτέμβριο του ίδιου έτους, ανακοινώθηκε πως η διαδικασία κατασκευής των υπολογιστών Raspberry Pi θα λάμβανε χώρα στο Ηνωμένο Βασίλειο, στις εγκαταστάσεις παραγωγής της Sony στην Ουαλία. έναντι της έως πρότινος χώρας κατασκευής που ήταν η Κίνα. [4]

Χωρίς να περάσουν πολλά χρόνια, τον Ιούλιο του 2014, η εταιρεία ανακοίνωσε την περαιτέρω εξέλιξη του προϊόντος με την επερχόμενη κυκλοφορία του Raspberry Pi Model B+. Εντός λίγων μηνών, τον Νοέμβριο του ίδιου έτους, ανακοινώθηκε επίσης το Raspberry Pi Model A+. [4]

Το επόμενο έτος, το 2015, ανακοινώθηκαν δύο ακόμα προϊόντα. Τον Φεβρουάριο, η εταιρεία ανακοίνωσε το Raspberry Pi 2, ενώ τον Νοέμβριο κυκλοφόρησε το μικρότερο και οικονομικότερο έως τώρα, το Raspberry Pi Zero, με διαστάσεις 65 mm × 30 mm και τιμή πώλησης 5\$. Το 2016 συνεχίστηκε με αμείωτο ρυθμό η εξέλιξη της εταιρείας με την κυκλοφορία του Raspberry Pi 3 τον Φεβρουάριο και την ανακοίνωση του Raspberry Pi Camera v2.1 τον Απρίλιο του ίδιου έτους. [4]

Το 2018, η εταιρεία αποφάσισε να προχωρήσει στην αναβάθμιση ήδη υπάρχοντος προϊόντος, του Raspberry Pi 3, με την κυκλοφορία του Raspberry Pi 3 Model B+ τον Μάρτιο 2018. Στην συνέχεια, τον Νοέμβριο του 2018 κυκλοφόρησε μια ακόμα αναβάθμιση του προϊόντος, το Raspberry Pi 3 Model A+. Τον Ιούνιο του 2019 ανακοινώθηκε η κυκλοφορία του Raspberry Pi 4 Model B. Το επόμενο χρόνο, το 2020, η εταιρεία προχώρησε σε μια ανακοίνωση και δύο νέες κυκλοφορίες. Τον Μάιο του 2020 ανακοινώθηκε μια έκδοση 8 GB του Raspberry Pi 4 για τιμή 75\$ ενώ τον Οκτώβριο και τον Νοέμβριο κυκλοφόρησαν τα Compute Module 4 και Raspberry Pi 400 αντίστοιχα. [4]

Τον Ιανουάριο του 2021 η εταιρεία παρουσίασε για πρώτη φορά το Raspberry Pi Pico και προς το τέλος του ίδιου έτους, τον Οκτώβριο, κυκλοφόρησε ένα ενημερωμένο Raspberry Pi OS, με πηγουμένη ονομασία Raspbian. Σημαντική παρατήρηση αποτελεί το γεγονός πως τον Φεβρουάριο του 2022, όποτε και συμπληρώθηκαν δέκα χρόνια από την κυκλοφορία του πρώτου RPi, η εταιρεία κατάφερε να σημειώσει συνολικές πωλήσεις που αγγίζουν το ύψος των 46 εκατομμυρίων τεμαχίων. [4]

2.3 Raspberry pi 3 Model B+

Καθώς το Raspberry Pi 3 Model B+ είναι αυτό το οποίο χρησιμοποιήθηκε στη παρούσα πτυχιακή εργασία θα μιλήσουμε αναλυτικά για τα πλήρη χαρακτηριστικά του. [6]

Το Raspberry Pi 3 Model B+ είναι το πιο πρόσφατο προϊόν της σειράς Pi 3. Διαθέτει τετραπύρηνο επεξεργαστή 64bit που τρέχει στα 1,4 GHz, dual band Wi-Fi στα 2.4 και 5 GHz, Bluetooth 4.2/BLE, ταχύτερο ethernet 1Gbps και δυνατότητα PoE μέσω ξεχωριστού PoE HAT. [6]

Το Raspberry Pi 3 Model B+ διατηρεί το ίδιο μηχανικό αποτύπωμα με τα δύο το Raspberry Pi 2 Model B και το Raspberry Pi 3 Model B. [6]

Παρακάτω παρουσιάζονται αναλυτικά τα χαρακτηριστικά του:

Processor:	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @1.4GHz
Memory:	1GB LPDDR2 SDRAM
Connectivity:	2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE

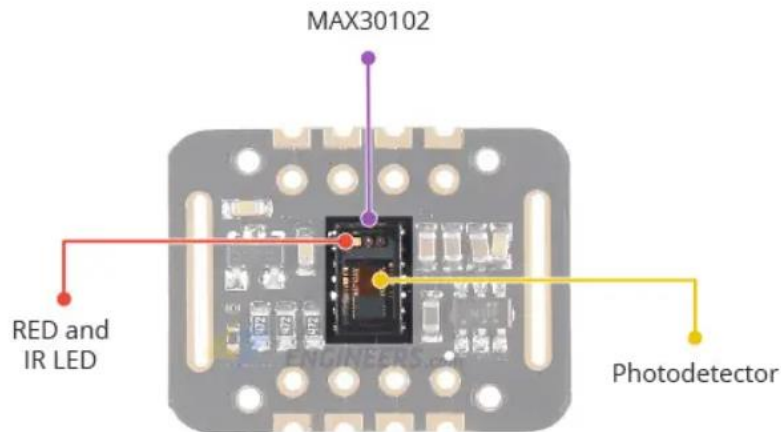
	Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps)
	4 × USB 2.0 ports"
Access:	Extended 40-pin GPIO header
Video & Sound:	1 × full size HDMI MIPI DSI display port MIPI CSI camera port 4 pole stereo output and composite video port
Multimedia:	H.264, MPEG-4 decode (1080p30), H.264 encode (1080p30), OpenGL ES 1.1, 2.0 graphics
SD card support:	Micro SD format for loading operating system and data storage
Input power:	5V/2.5A DC via micro-USB connector 5V DC via GPIO header Power over Ethernet (PoE) – enabled (requires separate PoE HAT)
Environment:	Operating temperature, 0–50°C
Product Lifetime:	To Raspberry Pi 3 Model B+ αναμένεται να παραμείνει σε παραγωγή μέχρι τον Ιανουάριο του 2023 [6]

3. Ο αισθητήρας MAX30102

Το παλμικό οξύμετρο και ο αισθητήρας καρδιακού ρυθμού MAX30102 είναι ένας βιομετρικός αισθητήρας plug-and-play χαμηλής κατανάλωσης ρεύματος και βασισμένος στο I2C πρωτόκολλο. Μπορεί να χρησιμοποιηθεί από φοιτητές, χομπίστες, μηχανικούς, κατασκευαστές και προγραμματιστές που θέλουν να ενσωματώσουν ζωντανά δεδομένα καρδιακού παλμού στα έργα τους. [7] [8]

3.1 MAX30102 Module Hardware Overview.

Η μονάδα του MAX30102 διαθέτει ένα ενσωματωμένο παλμικό οξύμετρο και αισθητήρα καρδιακού παλμού. Συνδυάζει δύο LED φώτα, φωτοανιχνευτή και βελτιστοποιημένη οπτική επεξεργασία αναλογικού σήματος χαμηλού θορύβου για την ανίχνευση σημάτων παλμικής οξυμετρίας (SpO2) και καρδιακού ρυθμού (HR). [7]

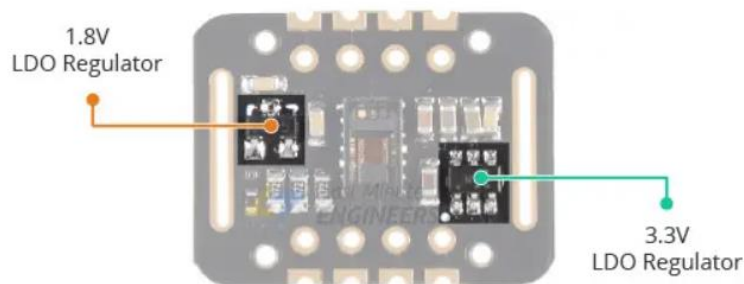


Εικόνα 3.1: Led αισθητήρα MAX30102 [13]

Όπως φαίνεται και στην παραπάνω φωτογραφία, στη μία πλευρά ο MAX30102 έχει δύο LED. Ένα κόκκινο και ένα IR Led. [13] Στην άλλη πλευρά υπάρχει ένας πολύ ευαίσθητος φωτοανιχνευτής. Η βασική ιδέα είναι ότι ανάβει ένα μόνο LED κάθε φορά, ανιχνεύεται η ποσότητα του φωτός που ανάβει πίσω στον ανιχνευτή κάθε φορά και με βάση την υπογραφή που δημιουργεί μπορούμε να μετρήσουμε τα επίπεδα οξυγόνου στο αίμα και τον καρδιακό παλμό.

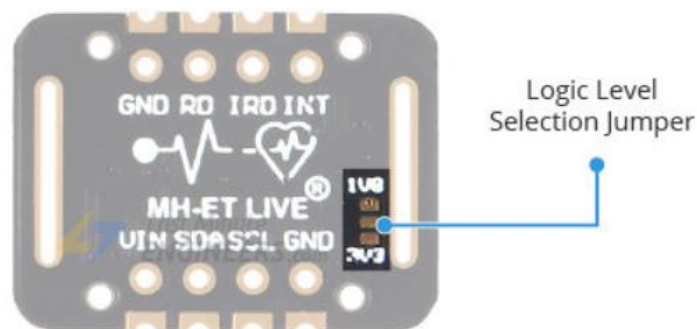
3.2 Απαιτήσεις ισχύος

Το microchip MAX30102 απαιτεί δύο διαφορετικές τάσεις τροφοδοσίας. Μία 1,8V για το ολοκληρωμένο κύκλωμα IC (Integrated Circuit) και μία 3,3V για τα δύο LED που χρησιμοποιεί. Συνεπώς η πλακέτα κατασκευάζεται με 2 ρυθμιστές των 1,8V και 3,3V. [7]



Εικόνα 3.2: LDO Regulators [13]

Στο πίσω μέρος της πλακέτας βρίσκεται ένα jumper ώστε να δίνει την δυνατότητα βραχυκύκλωσης και μπορεί να χρησιμοποιηθεί για να επιλεγεί μεταξύ του 3,3V και 1,8V λογικού επιπέδου (logic level). Από προεπιλογή επιλέγεται το logic level 3,3V. Φυσικά, δίνεται η δυνατότητα να επιλεγεί το 1,8V logic level σύμφωνα με τις εκάστοτε απαιτήσεις. Αυτή η λειτουργία μας επιτρέπει να συνδέσουμε τον MAX30102 σε οποιονδήποτε μικροελεγκτή με I/O (Input/Output) στάθμης 5V, 3,3V ακόμα και 1,8V.



Εικόνα 3.2.2 Logic Level Selection Jumper [13]

Ένα από τα πλέον σημαντικότερα χαρακτηριστικά του MAX30102 είναι η πολύ χαμηλή κατανάλωση ενέργειας. Ο MAX30102 καταναλώνει λιγότερα από 600mA κατά τη διάρκεια της μέτρησης. Επίσης, παρέχεται η δυνατότητα ο MAX30102 να τεθεί σε κατάσταση αναμονής όπου καταναλώνει μόνο 0,7mA. Αυτή η πολύ χαμηλή κατανάλωση δίνει τη δυνατότητα να χρησιμοποιηθεί σε συσκευές που τροφοδοτούνται από μπαταρία. [7]

3.3 On-Chip Temperature Sensor

Ο MAX30102 διαθέτει αισθητήρα θερμοκρασίας στο chip που μπορεί να χρησιμοποιηθεί για να αντισταθμίσει τις αλλαγές στο περιβάλλον και να βαθμονομήσει τις μετρήσεις. Αυτός είναι ένας πολύ ακριβής αισθητήρας που μπορεί να μετρήσει σε συνθήκες θερμοκρασίας από -40°C έως $+85^{\circ}\text{C}$ με ακρίβεια $\pm 1^{\circ}\text{C}$. [8] [13]

3.4 I2C Interface

Ο αισθητήρας χρησιμοποιεί μια απλή I2C διεπαφή δύο καλωδίων για επικοινωνία με τον μικροελεγκτή. Έχει μια σταθερή διεύθυνση I2C 0xA6HEX (για λειτουργία εγγραφής) και 0xAFHEX (για λειτουργία ανάγνωσης). [13]

3.5 FIFO Buffer

Ο MAX30102 ενσωματώνει ένα FIFO buffer για την αποθήκευση των δεδομένων. Το FIFO διαθέτει τράπεζα μνήμης 32 δειγμάτων που σημαίνει ότι μπορεί να αποθηκεύσει έως και 32 μετρήσεις καρδιακών παλμών και οξυγόνου του αίματος. Η προσωρινή μνήμη του FIFO buffer μπορεί να αποφορτίσει τον μικροελεγκτή από την ανάγνωση κάθε νέου δείγματος από τον αισθητήρα, εξοικονομώντας με αυτόν τον τρόπο την ισχύ του συστήματος. [13]

3.6 Interrupts

Ο MAX30102 μπορεί να προγραμματιστεί ώστε να δημιουργεί μια διακοπή, επιτρέποντας στον κεντρικό μικροελεγκτή, να εκτελεί άλλες εργασίες ενώ τα δεδομένα συλλέγονται από τον αισθητήρα. Η διακοπή μπορεί να ενεργοποιηθεί για πέντε διαφορετικές πηγές: [13]

- Power Ready: Ενεργοποιείται κατά την ενεργοποίηση ή μετά από κατάσταση διακοπής λειτουργίας.
- New Data Ready: Ενεργοποιείται μετά τη συλλογή κάθε δείγματος δεδομένων SpO2 και HR.
- Ambient Light Cancellation: Ενεργοποιείται όταν η λειτουργία ακύρωσης φωτός περιβάλλοντος της φωτοδιόδου SpO2/HR φτάσει στο μέγιστο όριο, επηρεάζοντας την έξοδο του ADC.
- Temperature Ready: Ενεργοποιείται όταν ολοκληρωθεί η μετατροπή θερμοκρασίας εσωτερικής μήτρας.
- FIFO Almost Full: Ενεργοποιείται όταν το FIFO buffer γεμίσει και μελλοντικές μετρήσεις πρόκειται να χαθούν. [8] [13]

3.7 Technical Specifications

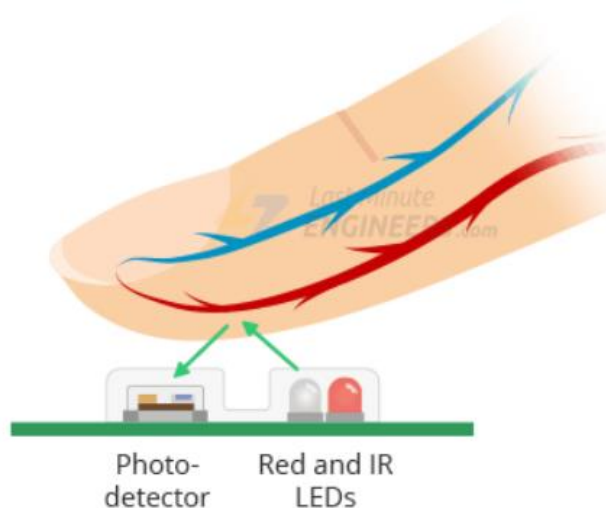
Ορισμένα βασικά τεχνικά χαρακτηριστικά φαίνονται στην παρακάτω εικόνα. Όπως φαίνεται στον παρακάτω πίνακα, η τάση του ρεύματος που μπορεί να δεχθεί κυμαίνεται από 3.3V έως 5.5V, ενώ η συσκευή είναι λειτουργική σε θερμοκρασίες από -40°C έως +85°C.

Power supply	3.3V to 5.5V
Current draw	~600μA (during measurements)
	~0.7μA (during standby mode)
Red LED Wavelength	660nm
IR LED Wavelength	880nm
Temperature Range	-40°C to +85°C
Temperature Accuracy	±1°C

Εικόνα 3.7: Τεχνικά Χαρακτηριστικά [13]

3.8 Λειτουργία αισθητήρα MAX30102

Ο MAX30102 και οποιοσδήποτε αισθητήρας μέτρησης καρδιακών παλμών και οξυγόνου του αίματος αποτελείται από ένα ζεύγος LED υψηλής έντασης (RED και IR Led), τα οποία λειτουργούν σε διαφορετικά μήκη κύματος, και έναν φωτοανιχνευτή. Τα μήκη κύματος αυτών των LED είναι 660nm (RED) και 880nm(IR LED). [13]

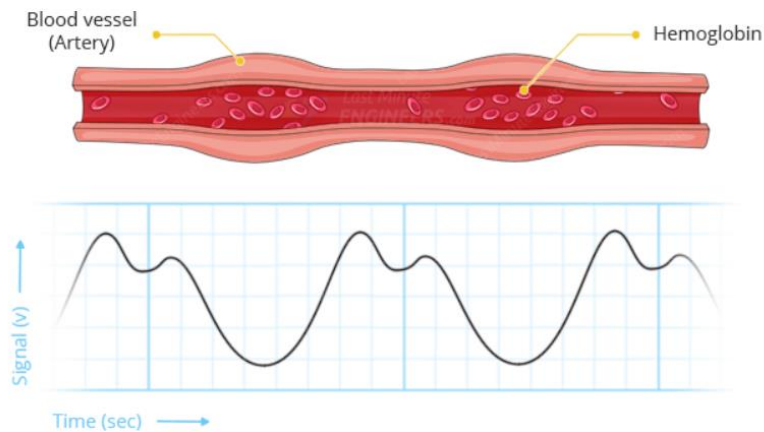


Εικόνα 3.8: Λειτουργία Photodetector και LED [13]

Ο MAX30102 λειτουργεί εκπέμποντας και τα δύο φώτα στο δάχτυλο ή τον λοβό του αυτιού (ή ουσιαστικά οπουδήποτε στο δέρμα όπου δεν είναι πολύ παχύ, έτσι ώστε να μπορούν τα δύο φώτα να διαπεράσουν εύκολα τον ιστό) και μετρώντας την ποσότητα του ανακλώμενου φωτός από τον φωτοανιχνευτή. Αυτή η μέθοδος ανίχνευσης παλμών μέσω του φωτός ονομάζεται φωτοπληθυσμογράφημα. [13]

3.8.1 Hear Rate Measurement

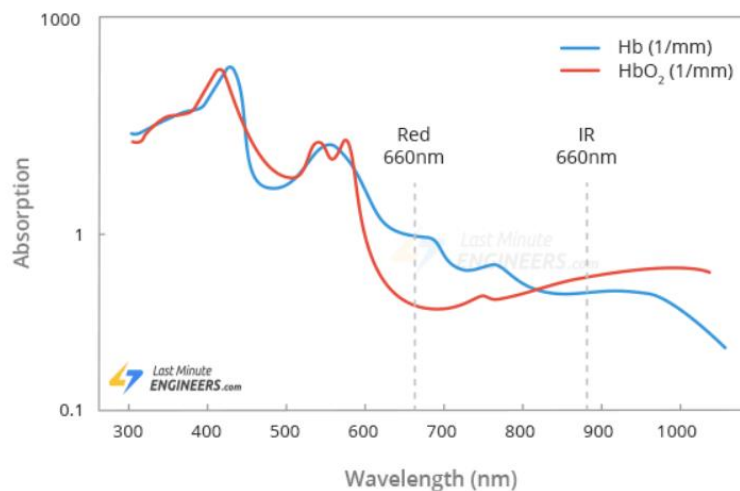
Η οξυγονωμένη αιμοσφαιρίνη (HbO₂) στο αρτηριακό αίμα έχει το χαρακτηριστικό της απορρόφησης IR φωτός. Όσο πιο κόκκινο είναι το αίμα (όσο υψηλότερη συγκέντρωση αιμοσφαιρίνης), τόσο περισσότερο φως IR απορροφάτε. Καθώς το αίμα διοχετεύεται μέσω του δακτύλου, με κάθε καρδιακό παλμό η ποσότητα του ανακλώμενου φωτός αλλάζει, δημιουργώντας μια μεταβαλλόμενη κυματομορφή στην έξοδο του φωτοανιχνευτή. Καθώς συνεχίζει να στέλνει φως και να λαμβάνει μετρήσεις με φωτοανιχνευτή, αρχίζει να λαμβάνει τη μέτρηση του καρδιακού παλμού (HR). [13]



Εικόνα 3.8.1: Μέτρηση καρδιακών παλμών [13]

3.8.2 Pulse Oximetry

Η παλμική οξυμετρία βασίζεται στην αρχή ότι η ποσότητα του κόκκινου και υπέρυθρου φωτός που απορροφάτε ποικίλλει ανάλογα με την ποσότητα οξυγόνου στο αίμα. Το παρακάτω γράφημα είναι το φάσμα απορρόφησης της οξυγονωμένης αιμοσφαιρίνης (HbO_2) και της αποξυγονωμένης αιμοσφαιρίνης (Hb). [13]



Εικόνα 3.8.2: Απορρόφηση HbO_2 και Hb [13]

Όπως φαίνεται από το γράφημα, το αποξυγονωμένο αίμα απορροφά περισσότερο κόκκινο φως (RED 660nm), ενώ το οξυγονωμένο αίμα απορροφά περισσότερο φως υπέρυθρων (IR LED 880nm). Με τη μέτρηση της αναλογίας IR και κόκκινου φωτός που λαμβάνει ο φωτοανιχνευτής, υπολογίζεται το επίπεδο οξυγόνου (SpO_2) στο αίμα. [13]

4. Software Details

4.1 Εισαγωγή

Έχοντας πλέον περιγράψει λεπτομερώς το RPi και τον αισθητήρα MAX30102 θα εμβαθύνουμε ακόμα περισσότερο μιλώντας για την γλώσσα προγραμματισμού πίσω από όλη αυτή τη διαδικασία.

4.2 Η γλώσσα προγραμματισμού Python

Η Python είναι μια ερμηνευμένη, διαδραστική, αντικειμενοστραφής γλώσσα προγραμματισμού.. Η σύνταξη της Python επιτρέπει στους προγραμματιστές να εφαρμόζουν συγκεκριμένα πρότυπα σε λιγότερες γραμμές κώδικα από αυτά που θα μπορούσαν να είναι βιώσιμα σε γλώσσες όπως η C για παράδειγμα. [9] Ενσωματώνει λειτουργικές μονάδες, εξαιρέσεις, δυναμική πληκτρολόγηση, πολύ υψηλού επιπέδου δυναμικούς τύπους δεδομένων και κλάσεις, προσφέροντας έτσι δομές που προορίζονται να ενεργοποιήσουν καθαρά προγράμματα τόσο σε μικρή όσο και σε τεράστια κλίμακα. [12]

Η Python υποστηρίζει μερικά πρότυπα προγραμματισμού, μαζί με μοτίβα αντικειμενοστραφών διαδικαστικών και λειτουργικών στυλ. Διαθέτει σύστημα πλήρους δυναμικού τύπου και αυτόματη διαχείριση μνήμης, παρόμοια με αυτή των Scheme, Ruby, Perl και Tclm ενώ ταυτόχρονα διαθέτει μια μεγάλη και πλήρη, τυπική βιβλιοθήκη. Μπορεί επίσης να χρησιμοποιηθεί ως γλώσσα επέκτασης για εφαρμογές που χρειάζονται προγραμματιζόμενη διεπαφή. [9] [12]

Όπως και άλλες δυναμικές γλώσσες, η Python χρησιμοποιείται συχνά ως γλώσσα δέσμης ενεργειών, αλλά χρησιμοποιείται επίσης και σε μια τεράστια ποικιλία πλαισίων χωρίς σενάριο. Χρησιμοποιώντας εργαλεία τρίτων, ο Python κώδικας μπορεί να συσκευαστεί σε αυτόνομα εκτελέσιμα πακέτα. Βασικό στοιχείο της είναι οι Python interpreters οι οποίοι σαφώς είναι διαθέσιμοι σε πάρα πολλά λειτουργικά συστήματα συμπεριλαμβανομένων των Windows, Linux, MacOS. [12]

Η CPython, η πιο ευρέως χρησιμοποιούμενη εφαρμογή της γλώσσας Python, γραμμένη στη γλώσσα προγραμματισμού C, είναι λογισμικό ελεύθερου και ανοιχτού κώδικα και έχει ένα μοντέλο ανάπτυξης που βασίζεται στην κοινότητα (community based), όπως και σχεδόν όλες οι εναλλακτικές υλοποιήσεις της. Την Python διαχειρίζεται το μη κερδοσκοπικό σωματείο Python Software Foundation. [9] [12]

Το Python Software Foundation είναι ένας ανεξάρτητος μη κερδοσκοπικός οργανισμός που κατέχει τα πνευματικά δικαιώματα στις εκδόσεις Python 2.1 και νεότερες. Η αποστολή του PSF είναι να προωθήσει την τεχνολογία ανοιχτού κώδικα που σχετίζεται με τη γλώσσα προγραμματισμού Python και να δημοσιοποιήσει τη χρήση της Python. [9] [12]

4.3 Ιστορική αναδρομή της Python

Η Python επινοήθηκε στα τέλη της δεκαετίας του 1980 από τον Guido van Rossum στο ερευνητικό κέντρο Centrum Wiskunde & Informatica (CWI) της Ολλανδίας ως διάδοχος της γλώσσας προγραμματισμού ABC μιας και από αυτή εμπνεύστηκε τη δημιουργία της ο Guido Van Rossum. [9] [10]

Η εφαρμογή της ξεκίνησε τον Δεκέμβριο του 1989. Ο Guido Van Rossum ανέλαβε την αποκλειστική ευθύνη για το έργο, ως κύριος προγραμματιστής, μέχρι τις 12 Ιουλίου 2018, όταν ανακοίνωσε την

απόσυρση του από τις ευθύνες του έργου ως «Benevolent Dictator for life» της Python, έναν τίτλο που του απένειμε η κοινότητα της Python ως αναγνώριση για την μακροπρόθεσμη δέσμευση του. [10]

Τον Ιανουάριο του 2019, ενεργοί προγραμματιστές της Python εξέλεξαν ένα πενταμελές Διοικητικό Συμβούλιο για να ηγηθεί του έργου. Η Python 2.0 κυκλοφόρησε στις 16 Οκτωβρίου 2000, με πολλά σημαντικά νέα χαρακτηριστικά, όπως list comprehensions, cycle-detecting garbage collection, reference counting και υποστήριξη Unicode. [9] [12]

Στις 3 Δεκεμβρίου 2008 κυκλοφόρησε η Python 3.0, με πολλά από τα κύρια χαρακτηριστικά της να αναφέρονται στις εκδόσεις 2.6.x και 2.7.x. Οι εκδόσεις της Python 3 περιλαμβάνουν το βοηθητικό πρόγραμμα 2to3, το οποίο αυτοματοποιεί τη μετάφραση του κώδικα Python 2 σε Python 3. Το τέλος ζωής της Python 2.7 είχε αρχικά οριστεί για το 2015, στη συνέχεια αναβλήθηκε για το 2020 λόγω ανησυχίας ότι ένας μεγάλος όγκος υπάρχοντος κώδικα δεν θα μπορούσε εύκολα να μεταφερθεί στην Python 3. Δεν θα κυκλοφορήσουν περαιτέρω ενημερώσεις κώδικα ασφαλείας ή άλλες βελτιώσεις για αυτή. Προς το παρόν υποστηρίζονται μόνο 3.7 και νεότερες εκδόσεις. [9] [12]

Το 2021, η Python 3.9.2 και 3.8.8 επιταχύνθηκαν καθώς όλες οι εκδόσεις της Python (συμπεριλαμβανομένης της 2.7) είχαν ζητήματα ασφαλείας που οδηγούσαν σε πιθανή απομακρυσμένη εκτέλεση κώδικα και δηλητηρίαση της προσωρινής μνήμης ιστού. Συνεχίζοντας την πορεία εξέλιξης, το 2022 επιταχύνθηκαν οι Python 3.10.4 και 3.9.12 καθώς στις εκδόσεις 3.8.13 και 3.7.13 εμφανίστηκαν πολλά επιπλέον ζητήματα ασφαλείας. [12]

Όταν κυκλοφόρησε η Python 3.9.13 τον Μάιο του 2022, ανακοινώθηκε ότι η σειρά 3.9 (που θα ενταχθεί στις παλαιότερες σειρές 3.8 και 3.7) θα λάμβανε μόνο επιδιορθώσεις ασφαλείας στο μέλλον. Στις 7 Σεπτεμβρίου 2022, πραγματοποιήθηκαν τέσσερις νέες εκδόσεις λόγω πιθανής επίθεσης άρνησης υπηρεσίας (DDoS attacks) οι οποίες ήταν οι: 3.10.7, 3.9.14, 3.8.14 και 3.7.14. [9]

Από τον Νοέμβριο του 2022, η έκδοση Python 3.11 είναι η σταθερή έκδοση. Οι αξιοσημείωτες αλλαγές από την 3.10 περιλαμβάνουν αυξημένη ταχύτητα εκτέλεσης προγράμματος και βελτιωμένη αναφορά σφαλμάτων. [9] [12]

4.4 Χαρακτηριστικά και φιλοσοφία της Python

Η Python είναι μια γλώσσα προγραμματισμού πολλαπλών παραδειγμάτων: Ο αντικειμενοστραφής προγραμματισμός και ο δομημένος προγραμματισμός υποστηρίζονται πλήρως, καθώς υπάρχει και ένας αριθμός γλωσσικών χαρακτηριστικών που υποστηρίζουν τον λειτουργικό προγραμματισμό και τον προγραμματισμό προσανατολισμένο στις πτυχές. [9] Πολλά άλλα παραδείγματα υποστηρίζονται χρησιμοποιώντας επεκτάσεις, συμπεριλαμβανομένου του σχεδιασμού με σύμβαση και του λογικού προγραμματισμού. Η Python χρησιμοποιεί δυναμική πληκτρολόγηση, συνδυασμό μέτρησης αναφορών και συλλέκτη σκουπιδιών που ανιχνεύει κύκλους για τη διαχείριση της μνήμης. Επιπλέον, σημαντικό χαρακτηριστικό της είναι η δυναμική ανάλυση ονόματος (late binding), η οποία δεσμεύει ονόματα μεθόδων και μεταβλητών κατά την εκτέλεση του προγράμματος. [12]

Ο σχεδιασμός της Python προσφέρει μόνο περιορισμένη υποστήριξη για λειτουργικό προγραμματισμό. Η γλώσσα διαθέτει λειτουργίες χαρτογράφησης, μείωσης και φιλτραρίσματος του κώδικα, κατανοήσιμες για λίστες, λεξικά και σύνολα, καθώς και γεννήτρια εκφράσεων. Η τυπική βιβλιοθήκη έχει δύο ενότητες (itertools και functools) που υλοποιούν λειτουργικά εργαλεία δανεισμένα από τις γλώσσες Haskell και Standard ML. [11]

Η βασική φιλοσοφία της γλώσσας συνοψίζεται με έναν πολύ έξυπνο τρόπο από την συλλογή «PEP 20 (The Zen of Python)», η οποία περιλαμβάνει περίτεχνους ορισμούς για να την περιγράψει. Μερικοί από αυτούς είναι:

- Το όμορφο είναι καλύτερο από το άσχημο.
- Το ρητό είναι καλύτερο από το άρρητο.
- Το απλό είναι καλύτερο από το σύνθετο.
- Το σύνθετο είναι καλύτερο από το περίπλοκο.
- Η αναγνωσιμότητα μετράει. [11]

Εκτός από την απαίτηση όλης της επιθυμητής λειτουργικότητας για να ενσωματωθεί στον πυρήνα της γλώσσας, η Python σχεδιάστηκε για να είναι εξαιρετικά επεκτάσιμη. Ενώ προσφέρει επιλογές στη μεθοδολογία κωδικοποίησης, η φιλοσοφία της Python απορρίπτει την πληθωρική σύνταξη, όπως στη Perl, υπέρ μιας πιο αραιής, λιγότερο ακατάστατης γραμματικής. Όπως το έθεσε ο Alex Martelli: «Το να περιγράψεις κάτι ως έξυπνο ΔΕΝ θεωρείται κομπλιμέντο στην κουλτούρα της Python. Θα πρέπει να υπάρχει ένας - και κατά προτίμηση μόνο ένας - προφανής τρόπος για να γίνει αυτό». [11]

Η CPython είναι επίσης διαθέσιμη η οποία μεταφράζει ένα σενάριο Python σε C και πραγματοποιεί άμεσες κλήσεις API επιπέδου C στον διερμηνέα Python. Οι προγραμματιστές της Python προσπαθούν να αποφύγουν την βιαστική βελτιστοποίηση, απορρίπτουν ενημερώσεις κώδικα σε μη κρίσιμα μέρη της CPython που θα πρόσφεραν μια μικρή αύξηση της ταχύτητας με το κόστος της σαφήνειας. Όταν η ταχύτητα είναι σημαντική, χρησιμοποιείται το PyPy, ένας just in time compiler, δηλαδή ένας μεταγλωττιστής που τρέχει κατά την εκτέλεση του προγράμματος. Διαφορετικά, μετακινούν λειτουργίες κρίσιμες για το χρόνο σε μονάδες επέκτασης γραμμένες σε γλώσσες "πιο κοντά στο μέταλλο", όπως η C. [9] [12]

Μια κοινή ορολογία στην κοινότητα Python είναι ο ορισμός pythonic, ο οποίος έχει ένα μεγάλο φάσμα σημασιών που σχετίζονται με τον κώδικα του προγράμματος. Το να πούμε ότι ο κώδικας είναι pythonic σημαίνει ότι χρησιμοποιεί καλά ιδιωτισμούς Python, ότι είναι φυσικός ή δείχνει ευχέρεια στη γλώσσα, ότι συμμορφώνεται με τη μινιμαλιστική φιλοσοφία της Python και την έμφαση στην αναγνωσιμότητα. Αντίθετα, ο κώδικας που είναι δύσκολο να κατανοηθεί ή διαβάζεται σαν μια πρόχειρη μεταγραφή από άλλη γλώσσα προγραμματισμού ονομάζεται unpythonic. Οι προγραμματιστές Python - ιδιαίτερα όσοι θεωρούνται γνώστες ή έμπειροι - αναφέρονται συχνά ως: Pythonists, Pythonistas, and Pythoners. [9] [11] [12]

4.5 Πλεονεκτήματα της Python

Η Python έχει κερδίσει πολύ μεγάλη δημοτικότητα τα τελευταία χρόνια και αυτό οφείλεται στα πολλά πλεονεκτήματα που έχει συγκριτικά με άλλες γλώσσες προγραμματισμού.

Αρχικά η Python είναι εύκολη στη μάθηση και την αναγνωσιμότητα. Το συντακτικό της είναι ακριβές και απλό, καθιστώντας με αυτό τον τρόπο ιδανική επιλογή για αρχάριους προγραμματιστές και γιατί όχι και για εμπειρότερους. Χαρακτηρίζεται επίσης από ευελιξία. Όπως αναφέραμε η Python υποστηρίζει και διαδικαστικό και αντικειμενοστραφή και λειτουργικό προγραμματισμό δίνοντας την δυνατότητα στον προγραμματιστή να επιλέξει να γράψει με την προσέγγιση που ταιριάζει καλύτερα στις ανάγκες του. [16]

Στα πλεονεκτήματά της εντάσσεται η μεγάλη πρότυπη βιβλιοθήκη. Η Python αποτελείται από μια τεράστια βιβλιοθήκη που παρέχει ένα μεγάλο φάσμα ενοτήτων, καθιστώντας την ανάπτυξη ταχύτερη και πιο αποτελεσματική. Διαθέτει ακόμα βιβλιοθήκες κοινότητας και τρίτων. Ταυτόχρονα με την τυπική

βιβλιοθήκη της, η Python έχει μεγάλη και ενεργή κοινότητα προγραμματιστών οι οποίοι έχουν αναπτύξει πάρα πολλές ακόμα βιβλιοθήκες τρίτων για διάφορους σκοπούς όπως η μηχανική μάθηση, ανάπτυξη ιστού, ανάλυσης δεδομένων κλπ. [16]

Επιπροσθέτως διακρίνεται για την ανεξαρτησία πλατφόρμας. Η Python μπορεί να εκτελεστεί σε πάρα πολλά λειτουργικά συστήματα χωρίς να χρειάζεται καμία αλλαγή ο κώδικας της από λειτουργικό σε λειτουργικό. Ένα ακόμα πλεονέκτημά της είναι η επεκτασιμότητα. Η Python μπορεί ανά πάσα στιγμή να ενσωματώσει κώδικα C και C++ δίνοντας την δυνατότητα στους προγραμματιστές να βελτιστοποιούν την απόδοση σε κομμάτια κρίσιμα των εφαρμογών τους. [16]

Συμπληρωματικά, η Python έχει άψογη υποστήριξη για την ανάπτυξη ιστού, με frameworks όπως το Django και το Flask να αποτελούν τις δημοφιλέστερες επιλογές για τη δημιουργία εφαρμογών ιστού. Επιπλέον πλεονέκτημα αποτελεί η επιστήμη δεδομένων και μηχανική μάθηση. Η Python έχει καθιερωθεί πλέον ως κυρίαρχη γλώσσα στους τομείς της επιστήμης δεδομένων και της μηχανικής μάθησης, χάρη σε βιβλιοθήκες της όπως οι NumPy, Pandas και TensorFlow. [16]

Τέλος, η τεράστια υποστηρικτική κοινότητα της Python δίνει την δυνατότητα στους προγραμματιστές να έχουν πρόσβαση σε πολλούς πόρους όπως, documentation, βιβλιοθήκες, troubleshooting. [16]

Η ευελιξία, η ευκολία χρήσης και η μεγάλη κοινότητα που υποστηρίζει την Python την έχει κάνει περιζήτητη στους προγραμματιστές που είτε είναι αρχάριοι είτε πιο έμπειροι, την χρησιμοποιούν υιοθετώντας την σε πολλούς και διαφορετικούς κλάδους

4.6 Βέλτιστες πρακτικές και συμβουλές

Κατά τη σύνταξη κώδικα Python, η τήρηση βέλτιστων πρακτικών και συμβάσεων κωδικοποίησης μπορεί να βελτιώσει την αναγνωσιμότητα, τη δυνατότητα συντήρησης και τη συνολική ποιότητα του κώδικα. Ακολουθούν μερικές βασικές συμβουλές:

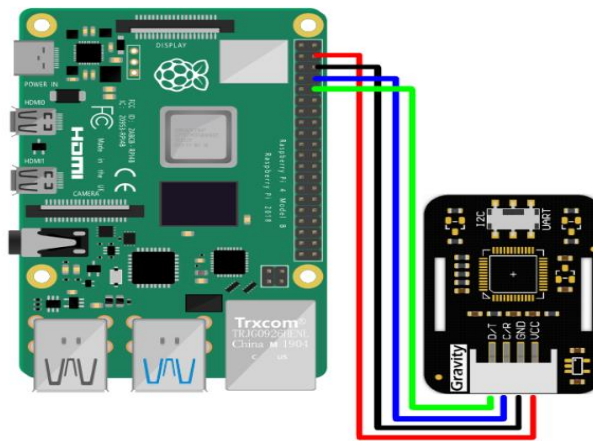
- Χρήση περιγραφικών ονομάτων μεταβλητών και συναρτήσεων: Επιλέξτε ονόματα με νόημα που αντικατοπτρίζουν τον σκοπό των μεταβλητών και των συναρτήσεων, καθιστώντας τον κώδικα αυτονόητο. [9]
- Ακολουθήστε το PEP 8: Το PEP 8 είναι ο επίσημος οδηγός στυλ για τον κώδικα Python. Η τήρηση των οδηγιών του, όπως η σταθερή εσοχή, η σωστή απόσταση και οι συμβάσεις ονομασίας, κάνουν τον κώδικά σας πιο ευανάγνωστο. [17]
- Αποφύγετε τους μαγικούς αριθμούς: Αντί να χρησιμοποιείτε αυθαίρετους αριθμούς απευθείας στον κώδικα, χρησιμοποιήστε επώνυμες σταθερές ή μεταβλητές για να βελτιώσετε τη διατήρηση του κώδικα. [9]
- Τεκμηριώστε τον κωδικό σας: Προσθέστε σχόλια και συμβολοσειρές εγγράφων για να εξηγήσετε τη λογική και τον σκοπό του κώδικά σας. Αυτό βοηθά άλλους προγραμματιστές να κατανοήσουν τη λειτουργικότητα του κώδικα. [17]
- Χρήση κατανοήσεων λίστας: Όταν δημιουργείτε λίστες με βάση υπάρχουσες λίστες, χρησιμοποιήστε κατανοήσεις λιστών για να γράψετε συνοπτικό και ευανάγνωστο κώδικα.
- Αποφύγετε καθολικές μεταβλητές: Ελαχιστοποιήστε τη χρήση καθολικών μεταβλητών, καθώς μπορεί να οδηγήσουν σε ανεπιθύμητα αποτελέσματα και να κάνουν τον κώδικα πιο αργό και δυσανάγνωστο. [17]
- Χρήση εικονικών περιβαλλόντων: Δημιουργήστε απομονωμένα εικονικά περιβάλλοντα για τα έργα σας για τη διαχείριση των εξαρτήσεων και την αποφυγή διενέξεων εκδόσεων.
- Δοκιμάστε τον κώδικά σας: Γράψτε δοκιμές μονάδας για να βεβαιωθείτε ότι ο κώδικάς σας λειτουργεί όπως αναμένεται και εντοπίζει νωρίς πιθανά σφάλματα.

- Έλεγχος έκδοσης: Χρησιμοποιήστε συστήματα ελέγχου έκδοσης όπως το Git για να παρακολουθείτε τις αλλαγές στον κώδικά σας και να συνεργάζεστε αποτελεσματικά με άλλους.
- Βελτιστοποίηση απόδοσης: Δημιουργήστε προφίλ του κώδικά σας για να εντοπίσετε σημεία συμφόρησης και να βελτιστοποιήσετε κρίσιμα τμήματα εάν χρειάζεται. [9]

5. Raspberry Pi 3 Model B & MAX30102

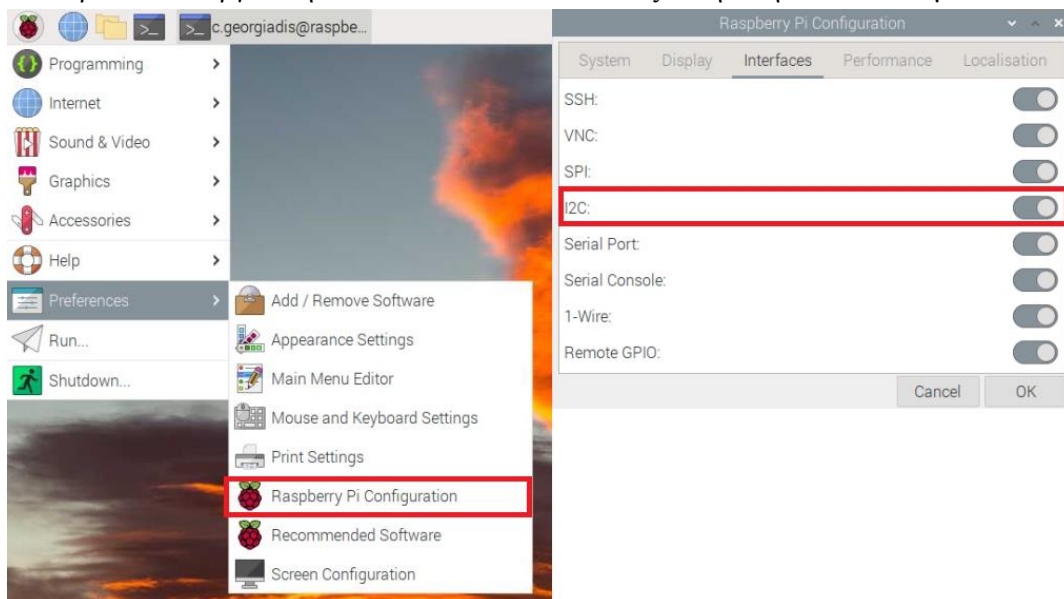
Σε αυτό το κεφάλαιο θα παρουσιαστεί η λειτουργικότητα του αισθητήρα πάνω στο RPI και θα αναλύσουμε επιπλέον πως έγινε η συνδεσμολογία, η επικοινωνία, αλλά και η εμφάνιση των μετρήσεων των καρδιακών παλμών και το οξυγόνο του αίματος στην οθόνη του Android κινητού μέσω Bluetooth.

5.1 Συνδεσιμότητα



Εικόνα 5.1 Συνδεσιμότητα RPI με MAX30102 [14]

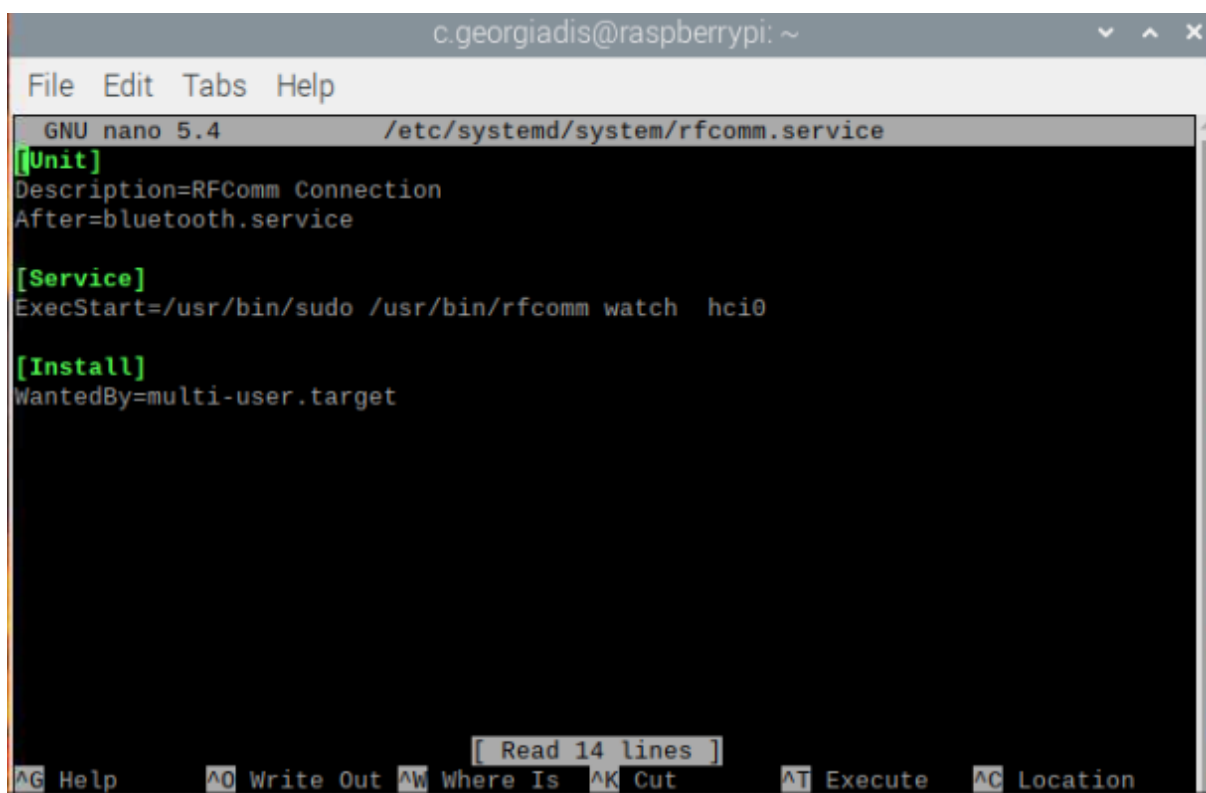
Για να επιτευχθεί η συνδεσιμότητα φυσικά δεν αρκεί μόνο να συνδεθούν τα καλώδια. Στο Raspberry Pi 3 Model B πρέπει να ενεργοποιηθεί το I2C Interface ώστε να ξεκινήσει η επικοινωνία με τον αισθητήρα.



Εικόνα 5.1.1 Ενεργοποίηση I2C Interface

5.2 Ενεργοποίηση Bluetooth κατά την εκκίνηση

Προκειμένου να μπορεί το smartphone να συνδέεται στο Raspberry δημιουργήθηκε ένα script το οποίο κατά την εκκίνηση του λειτουργικού συστήματος αρχίζει και εκτελείται. Με αυτό τον τρόπο το Raspberry ακούει στην εικονική θύρα rfcomm0 και περιμένει ζεύξη με το κινητό τηλέφωνο. Για να επιτευχθεί το παραπάνω δημιουργήθηκε ένα νέο service ώστε να εκτελεί αποκλειστικά την συγκεκριμένη διεργασία.



```
c.georgiadis@raspberrypi: ~
File Edit Tabs Help
GNU nano 5.4 /etc/systemd/system/rfcomm.service
[Unit]
Description=RFCOMM Connection
After=bluetooth.service

[Service]
ExecStart=/usr/bin/sudo /usr/bin/rfcomm watch hci0

[Install]
WantedBy=multi-user.target

[ Read 14 lines ]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
```

Εικόνα 5.1: Systemd service: rfcomm watch hci0

Μπορούμε να διακρίνουμε στην παραπάνω εικόνα ότι έχει δημιουργηθεί ένα bash script με όνομα rfcomm.service το οποίο κατά την εκκίνηση του λειτουργικού εκτελεί την εντολή «rfcomm watch hci0». Μόλις δημιουργηθεί αυτή η νέα διεργασία, είναι απαραίτητη η εντολή «systemctl enable rfcomm.service» ώστε να ενεργοποιηθεί. Προκειμένου να μπορεί το smartphone να συνδεθεί στην εικονική θύρα rfcomm0 θα πρέπει να έχει γίνει ζεύξη μεταξύ αυτού και του RPi.

5.3 Λειτουργία κουμπιού

Η κύρια ιδέα είναι πως θέλουμε οι μετρήσεις να ξεκινούν και να σταματούν με το πάτημα ενός κουμπιού. Αυτή τη λειτουργία την εξυπηρετεί ένα momentary button που συνδέθηκε πάνω στο Rpi, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 5.3 Momentary button

Το κουμπί έχει τοποθετηθεί πάνω σε breadboard για λόγους διευκόλυνσης. Βλέπουμε ότι τα pins που χρησιμοποιήσαμε για να συνδεθεί πάνω στο Raspberry είναι το 10 για data και 17 για 3.3V. Προκειμένου να είμαστε σε θέση να ελέγξουμε πώς ανταποκρίνεται το κουμπί, δημιουργήσαμε ένα python αρχείο με το όνομα button.py. Το αρχείο αυτό εκτελείται κατά την εκκίνηση του Raspberry σε background και είναι σε θέση να ξεκινήσει ή να σταματήσει τις μετρήσεις. Για να μπορεί να τρέξει κατά την εκκίνηση, χρησιμοποιήθηκε η autostart μέθοδος. Πρώτα με την εντολή: `mkdir /home/pi/.config/autostart` φτιάχνουμε ένα φάκελο μέσα στον οποίο θα περιέχει το button.desktop file με την εντολή: `nano /home/pi/.config/autostart/clock.desktop`. Μέσα στο button.desktop file ουσιαστικά θέλουμε να κάνει execute το button.py αρχείο που περιέχει την λειτουργικότητα του κουμπιού.

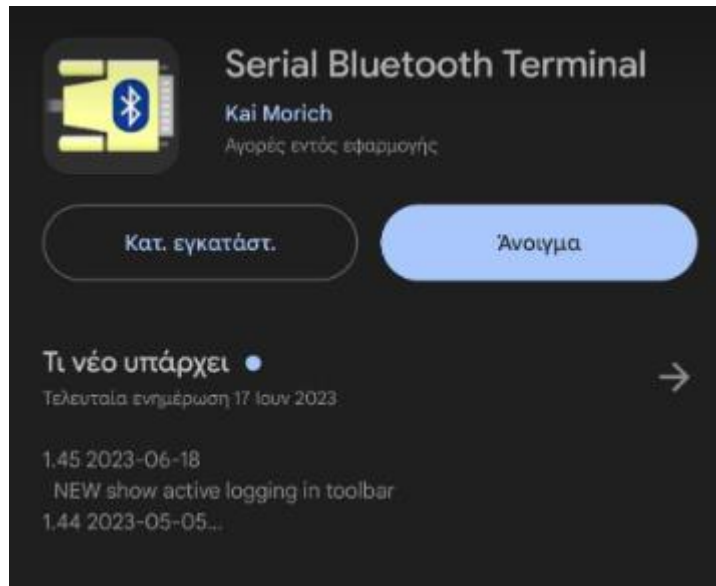
```

c.georgiadis@raspberrypi: ~
File Edit Tabs Help
GNU nano 5.4 /home/c.georgiadis/.config/autostart/button.desktop *
[Desktop Entry]
Type=Application
Name=Button
Exec=/usr/bin/python3 /home/c.georgiadis/Desktop/ptuxiaki/button.py
  
```

Εικόνα 5.3.1 button.desktop file

5.4 Εκτύπωση μετρήσεων

Προκειμένου οι μετρήσεις του αισθητήρα να εμφανίζονται στην οθόνη του κινητού θα πρέπει αρχικά να επιλεγεί η κατάλληλη εφαρμογή. Στην συγκεκριμένη πτυχιακή χρησιμοποιήθηκε στο κινητό η εφαρμογή «Serial Bluetooth Terminal» από τον Kai Morich.



Εικόνα 5.4 Serial Bluetooth Terminal by Kai Morich

Ταυτόχρονα, στο αρχείο `gain_heartbeat_SPO2.py` με την εντολή `«f_device1=open('/dev/rfcomm0','w',1)»` λέμε στο raspberry να στείλει τις μετρήσεις στο smartphone. Προκειμένου όμως να τυπωθούν στην οθόνη χρησιμοποιούμε την εντολή `«f_device1.write("\n SPO2 is : "+str(max30102.SPO2)+"% \n Heart Rate is : "+str(max30102.heartbeat)+ " Times/min \n")»` Επίσης, σημαντικό είναι οι μετρήσεις να μπορούν να αποθηκεύονται σε ένα txt αρχείο ώστε να υπάρχουν αποθηκευμένες για τυχόν συγκρίσεις. Για να μπορεί να συμβεί αυτό, στο αρχείο `gain_heartbeat_SPO2` το οποίο είναι η main κλάση για την λειτουργία του αισθητήρα προστέθηκε η εντολή: `f_measures = open('/home/c.georgiadis/Desktop/ptuxiaki/measures','w')`. Η συγκεκριμένη εντολή δημιουργεί το αρχείο `measures` το οποίο κάθε φορά που ξεκινάνε νέες μετρήσεις γίνεται overwrite με τις καινούργιες ώστε μελλοντικά να αποφευχθούν προβλήματα χώρου στην μνήμη του Raspberry. Με την εντολή: `f_measures.write("\n SPO2 is : "+str(max30102.SPO2)+"% \n Heart Rate is : "+str(max30102.heartbeat)+ " Times/min \n")` τα αποτελέσματα που τυπώνονται στην οθόνη του κινητού τυπώνονται επίσης και στο αρχείο `measures`.

```
22:47:30.427
22:47:30.428 SPO2 is : 99%
22:47:30.428 Heart Rate is :87 Times/min
22:47:31.401
22:47:31.401 SPO2 is : 99%
22:47:31.404 Heart Rate is :87 Times/min
22:47:32.413
22:47:32.413 SPO2 is : 99%
22:47:32.415 Heart Rate is :88 Times/min
22:47:33.421
22:47:33.468 SPO2 is : 99%
22:47:33.469 Heart Rate is :88 Times/min
22:47:34.403
22:47:34.403 SPO2 is : 99%
22:47:34.404 Heart Rate is :88 Times/min
22:47:35.531
22:47:35.632 SPO2 is : 99%
22:47:35.634 Heart Rate is :88 Times/min
22:47:36.436
22:47:36.436 SPO2 is : 99%
22:47:36.438 Heart Rate is :88 Times/min
22:47:37.433
22:47:37.434 SPO2 is : 99%
22:47:37.439 Heart Rate is :88 Times/min
22:47:38.496
22:47:38.496 SPO2 is : 99%
22:47:38.498 Heart Rate is :88 Times/min
22:47:39.465
22:47:39.465 SPO2 is : 99%
22:47:39.467 Heart Rate is :88 Times/min
22:47:40.435
22:47:40.436 SPO2 is : 99%
22:47:40.437 Heart Rate is :89 Times/min
22:47:41.525
22:47:41.525 SPO2 is : 99%
22:47:41.527 Heart Rate is :89 Times/min
22:47:42.451
22:47:42.452 SPO2 is : 99%
22:47:42.455 Heart Rate is :89 Times/min
22:47:43.477
22:47:43.477 SPO2 is : 99%
22:47:43.478 Heart Rate is :89 Times/min
22:47:44.427
22:47:44.431 SPO2 is : 99%
22:47:44.431 Heart Rate is :88 Times/min
```

Εικόνα 5.4.1 Εκτύπωση μετρήσεων στο κινητό

```
383 SPO2 is : 99%
384 Heart Rate is :88 Times/min
385
386 SPO2 is : 99%
387 Heart Rate is :88 Times/min
388
389 SPO2 is : 99%
390 Heart Rate is :89 Times/min
391
392 SPO2 is : 99%
393 Heart Rate is :89 Times/min
394
395 SPO2 is : 99%
396 Heart Rate is :89 Times/min
397
398 SPO2 is : 99%
399 Heart Rate is :89 Times/min
400
401 SPO2 is : 99%
402 Heart Rate is :88 Times/min
403
404 SPO2 is : 99%
405 Heart Rate is :88 Times/min
406
407 SPO2 is : 99%
408 Heart Rate is :88 Times/min
409
410 SPO2 is : 99%
```

Εικόνα 5.4.2 Εκτύπωση μετρήσεων measures.txt

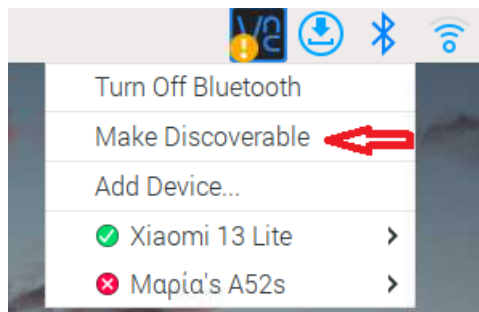
Όπως μπορούμε να παρατηρήσουμε στις παραπάνω δύο εικόνες, μόλις ξεκινήσαμε τις μετρήσεις τυπώθηκαν και στην οθόνη του κινητού και στο αρχείο measures.txt. Με αυτόν τον τρόπο μπορούμε ανά πάσα στιγμή να ελέγχουμε τις μετρήσεις των καρδιακών παλμών και του κορεσμού του οξυγόνου στο αίμα στο κινητό μας αλλά και στο αρχείο measures.txt.

5.4.1 Δυνατότητα εκτύπωσης σε παραπάνω από μια συσκευή

Έχουμε τη δυνατότητα να τυπώνουμε τα αποτελέσματα των μετρήσεων σε πάνω από μια συσκευή android ή ios. Το μειονέκτημα στη συγκεκριμένη περίπτωση είναι πως η διαδικασία δεν είναι αυτοματοποιημένη και θα πρέπει να επέμβουμε οι ίδιοι στο Raspberry ώστε να μπορεί να κάνει τη σύνδεση με το νέο smartphone.

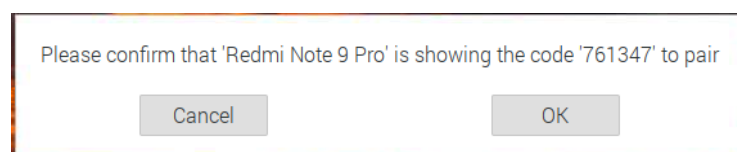
Θα περιγράψουμε τον τρόπο που πρέπει να γίνει η διαδικασία αυτή με συγκεκριμένα βήματα:

- Αρχικά, πρέπει πρώτα να κάνουμε το Raspberry discoverable ώστε να μπορούν οι άλλες συσκευές να κάνουν αίτηση σύνδεσης μέσω Bluetooth.



Εικόνα 5.4.3: Εικονική αναπαράσταση 1^{ου} βήματος

- Είναι κανόνας και αναγκαίο πως όταν δύο συσκευές συνδέονται με Bluetooth πρέπει να γίνουν pair την πρώτη φορά. Συνεπώς, εφόσον θέλουμε να συνδέσουμε ένα νέο smartphone στο raspberry θα πρέπει να δημιουργήσουμε τη ζεύξη.



Εικόνα 5.4.3: Εικονική αναπαράσταση 2^{ου} βήματος

- Έπειτα, αφού η ζεύξη των δύο συσκευών έχει πραγματοποιηθεί πρέπει να ανοίξουμε μια νέα εικονική θύρα rfcomm ώστε να μπορεί να λάβει σειριακά τα αποτελέσματα των μετρήσεων η νέα συσκευή που συνδέσαμε με το Raspberry. Για να επιτευχθεί αυτό χρησιμοποιούμε την εντολή «rfcomm watch rfcomm1» με την οποία ανοίγουμε την εικονική θύρα rfcomm1 στην οποία θα ακούει η συσκευή μας.

```

c.georgiadis@raspberrypi: ~
File Edit Tabs Help
c.georgiadis@raspberrypi:~ $ rfcomm watch rfcomm1
Waiting for connection on channel 1
Connection from 4C:02:20:5D:44:66 to /dev/rfcomm1
Press CTRL-C for hangup

```

Εικόνα 5.4.4: Εικονική αναπαράσταση 3^{ου} βήματος

- Με τη θύρα rfcomm1 πλέον διαθέσιμη και προκειμένου να μπορεί να χρησιμοποιηθεί πρέπει να αλλάξουμε τα δικαιώματα της ώστε να μπορούμε να την αξιοποιήσουμε. Αυτό επιτυγχάνεται με την εντολή «sudo chmod 666 rfcomm1» αφού πρώτα είμαστε στο σωστό path που περιέχει την εικονική θύρα rfcomm1.

```

c.georgiadis@raspberrypi: /dev
File Edit Tabs Help
console      loop-control ram8      tty20  tty47  v4l     vhci
cuse         mapper       ram9      tty21  tty48  vchiq   video10
disk         media0       random    tty22  tty49  vcio    video11
dma_heap     media1       rfcomm0   tty23  tty5   vc-mem  video12
dri          media2       rfcomm1   tty24  tty50  vcs     video13
fd           mem          rfkill    tty25  tty51  vcs1    video14
full         mmcblk0     serial0   tty26  tty52  vcs2    video15
fuse         mmcblk0p1   serial1   tty27  tty53  vcs3    video16
gpiochip0    mmcblk0p2   Smm       tty28  tty54  vcs4    video18
gpiochip1    queue       snd        tty29  tty55  vcs5    video20
gpiochip2    net         spidev0.0 tty3    tty56  vcs6    video21
gpiomem     null        spidev0.1 tty30   tty57  vcs7    video22
hidraw0     ppp         stderr     tty31  tty58  vcsa    video23
hidraw1     ptmx        stdin      tty32  tty59  vcsa1   video31
hwrng       pts         stdout     tty33  tty6   vcsa2   watchdog
i2c-1       ram0        tty        tty34  tty60  vcsa3   watchdog
i2c-2       ram1        tty0       tty35  tty61  vcsa4   zero
initctl     ram10       tty1       tty36  tty62  vcsa5
input       ram11       tty10      tty37  tty63  vcsa6
kmsg        ram12       tty11      tty38  tty7   vcsa7
log         ram13       tty12      tty39  tty8   vcsm-cma
loop0       ram14       tty13      tty4   tty9   vcsu
c.georgiadis@raspberrypi:/dev $ sudo chmod 666 rfcomm1
c.georgiadis@raspberrypi:/dev $

```

Εικόνα 5.4.5: Εικονική αναπαράσταση 4^{ου} βήματος

- Έχοντα πλέον ολοκληρώσει τα παραπάνω βήματα μπορούμε να τροποποιήσουμε την κλάση gain_heartbeat_SPO2.py και κάτω από τις εντολές «f_device1= open('/dev/rfcomm0','w',1)» και f_device1.write("\n SPO2 is : "+str(max30102.SPO2)+"% \n Heart Rate is : "+str(max30102.heartbeat)+ " Times/min \n")) να προσθέσουμε την νέα μας συσκευή η οποία ακούει στην rfcomm1 με τις εντολές εντολές «f_device2= open('/dev/rfcomm1','w',1)» και

```
f_device2.write("\n SPO2 is : "+str(max30102.SPO2)+"% \n Heart Rate is :"  
+str(max30102.heartbeat)+ " Times/min \n")»
```

```
f_device1 = open('/dev/rfcomm0','w',1) #send to android via bluettoth rfcomm0 interface  
f_device2 = open('/dev/rfcomm1','w',1) #send to android via bluettoth rfcomm0 interface  
f_device1.write("\n SPO2 is : "+str(max30102.SPO2)+"% \n Heart Rate is :"+str(max30102.heartbeat)+ " Times/min \n")  
f_device2.write("\n SPO2 is : "+str(max30102.SPO2)+"% \n Heart Rate is :"+str(max30102.heartbeat)+ " Times/min \n")
```

Εικόνα 5.4.6: Εικονική αναπαράσταση 5^{ου} βήματος

Με την παραπάνω διαδικασία έχουμε την δυνατότητα να προσθέσουμε και τρίτη και τέταρτη συσκευή παρακολούθησης των μετρήσεων του αισθητήρα MAX30102 αρκεί για κάθε συσκευή να δημιουργούμε μια νέα εικονική θύρα rfcomm με τα κατάλληλα δικαιώματα και να τροποποιούμε τον κώδικα μας ώστε να μπορεί να στείλει εκεί τις μετρήσεις.

6. Αποτελέσματα μετρήσεων και συμπεράσματα

Στο συγκεκριμένο κεφάλαιο θα συγκρίνουμε τις μετρήσεις από διαφορετικές συσκευές με τις μετρήσεις του αισθητήρα MAX30102, αλλά και θα δούμε ποιος είναι ο βέλτιστος τρόπος να λαμβάνουμε τον κορεσμό του οξυγόνου του αίματος και ένας καρδιακός παλμούς μέσω του αισθητήρα.

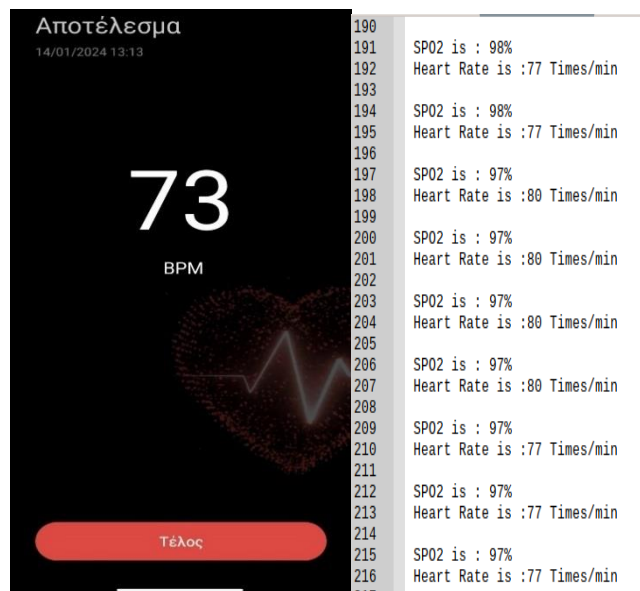
6.1 MAX30102 συμβουλές και πρακτικές

Σε αυτό το σημείο πρέπει να υπογραμμίσουμε ότι ο MAX30102 είναι ένας αρκετά καλός και προσιτός σε τιμή αισθητήρας, κατάλληλος για χομπίστες και προγραμματιστές, όχι όμως για ιατρική χρήση. Μετά από πολλές δοκιμές και συγκρίσεις καταλήγουμε σε ορισμένες συμβουλές σχετικά με τη σωστή χρήση του αισθητήρα, ώστε να μπορούμε να λάβουμε όσο πιο ακριβής μετρήσεις γίνεται.

- Η πίεση του δακτύλου πάνω στον αισθητήρα πρέπει να είναι ισορροπημένη. Ούτε χαλαρή, ώστε να μπορεί να γίνει σωστή επαφή με το δάκτυλο, ούτε πολύ δυνατή ώστε να μην εμποδίσει την κυκλοφορία του αίματος του δακτύλου.
- Προτιμότερο δάκτυλο για να γίνονται μετρήσεις, είναι ο αντίχειρας ώστε να είναι μεγαλύτερη η επιφάνεια που ακουμπάει τον αισθητήρα.
- Το δάκτυλο πρέπει να είναι στεγνό. Ακόμα και ένα ιδρωμένο δάκτυλο θα μπορούσε να προκαλέσει στον αισθητήρα μια σειρά από λανθασμένες μετρήσεις.
- Το δάκτυλο πάνω στον αισθητήρα πρέπει να είναι ακίνητο. Ακόμα και μικρές κινήσεις του δακτύλου πάνω στον αισθητήρα δείχνουν ότι είναι πολύ ευαίσθητος και επηρεάζει τις μετρήσεις.
- Οι σωστές μετρήσεις παίρνουν ώρα. Παρατηρήθηκε πως μετά από 1 λεπτό μέτρησης, οι ενδείξεις ήταν ίδιες ή αρκετά κοντά με μετρήσεις άλλων συσκευών.

6.2 Σύγκριση με Xiaomi 13 Lite

Η πρώτη σύγκριση μετρήσεων θα γίνει ανάμεσα στον αισθητήρα μας MAX30102 και το smartphone Xiaomi 13 Lite. Καθώς το smartphone δεν μπορεί να μετρήσει κορεσμό του οξυγόνου του αίματος, οι μετρήσεις που θα παραθέσουμε αφορούν αποκλειστικά τον καρδιακό παλμό.

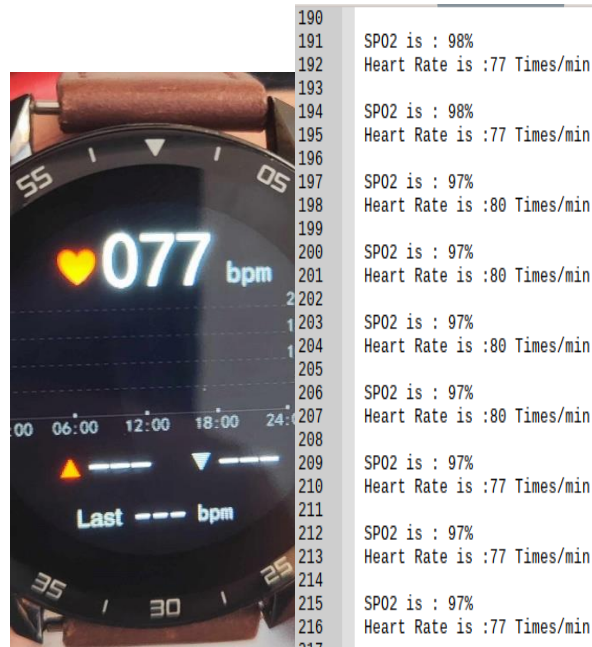


Εικόνα 6.2 Xiaomi 13 Lite vs MAX30102

Μπορούμε να παρατηρήσουμε πως η μέτρηση του καρδιακού παλμού μέσω του smartphone είναι σαφώς μικρότερη κατά τέσσερις έως δέκα χτύπους το λεπτό. Δεν είναι μεγάλη απόκλιση και φυσικά δεν μπορούμε να εμπιστευτούμε δύο συσκευές που δεν είναι κατάλληλες για ιατρική χρήση.

6.3 Σύγκριση με Smartwatch BlackView X1

Η δεύτερη σύγκριση μετρήσεων θα γίνει ανάμεσα στον αισθητήρα μας MAX30102 και το smartwatch BlackView X1. Καθώς το smartwatch δεν μπορεί να μετρήσει κορεσμό του οξυγόνου του αίματος οι μετρήσεις που θα παραθέσουμε αφορούν αποκλειστικά τον καρδιακό παλμό.

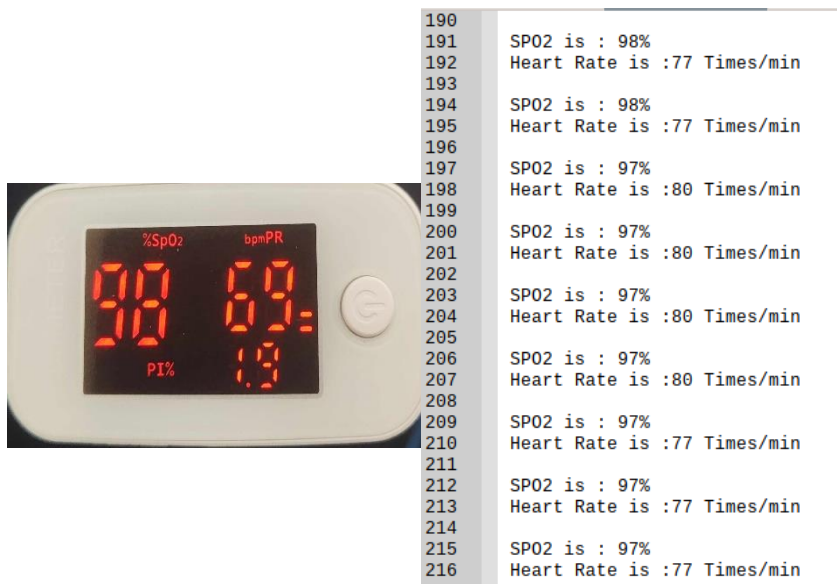


Εικόνα 6.5 BlackView X1 vs MAX30102

Μπορούμε να παρατηρήσουμε πως οι δύο συσκευές μας δίνουν πολύ κοντινά αποτελέσματα καθώς σε live μέτρηση έχουν τους ίδιους καρδιακούς παλμούς ή μια απόκλιση της τάξης τριών με πέντε μονάδων.

6.4 Σύγκριση με οξύμετρο YM101

Η τρίτη σύγκριση μετρήσεων θα γίνει ανάμεσα στον αισθητήρα μας MAX30102 και το οξύμετρο YM101. Εδώ είμαστε σε θέση να πάρουμε τις μετρήσεις και για τον κορεσμό του οξυγόνου του αίματος και των καρδιακών παλμών.



Εικόνα 6.4 YM101 vs MAX30102

Παρατηρούμε πως το οξύμετρο YM101 μας έδωσε κορεσμό του οξυγόνου 98% και καρδιακούς παλμούς 68bpmPR. Σε σύγκριση με τον αισθητήρα μας MAX30102, ο οποίος μας έδωσε κορεσμό του οξυγόνου 97-98% και καρδιακούς παλμούς 77-80bpmPR. Παρατηρούμε πως οι μετρήσεις για τον κορεσμό του οξυγόνου του αίματος μας έδωσαν πολύ καλά αποτελέσματα αφού οι τιμές είναι ίδιες ή με απόκλιση μιας μονάδας, σε σύγκριση με τους καρδιακούς παλμούς που η διαφορά κοντεύει τους δέκα με ένδεκα χτύπους ανά λεπτό.

6.5 Σύγκριση με οξύμετρο Esperanza

Η τέταρτη σύγκριση μετρήσεων θα γίνει ανάμεσα στον αισθητήρα μας MAX30102 και το οξύμετρο Esperanza. Πάλι είμαστε σε θέση να πάρουμε τις μετρήσεις και για τον κορεσμό του οξυγόνου του αίματος και των καρδιακών παλμών.



Εικόνα 6.5 Esperanza vs MAX30102

Βλέπουμε πως το οξύμετρο Esperanza μας έδωσε κορεσμό του οξυγόνου 98% και καρδιακούς παλμούς 75bpmPR. Σε σύγκριση με τον αισθητήρα μας MAX30102 ο οποίος μας έδωσε κορεσμό του οξυγόνου 97-98% και καρδιακούς παλμούς 77-80bpmPR. Παρατηρούμε πως οι μετρήσεις για τον κορεσμό του οξυγόνου του αίματος μας έδωσαν πολύ καλά αποτελέσματα αφού οι τιμές είναι ίδιες ή με απόκλιση μιας μονάδας. Όπως επίσης και οι μετρήσεις καρδιακών παλμών που σε σχέση με τις υπόλοιπες συσκευές που έγιναν οι συγκρίσεις εκτός του smartwatch εδώ βλέπουμε πως η μεγαλύτερη διαφορά είναι της τάξης πέντε καρδιακών παλμών ανά λεπτό.

	Xiaomi 13 Lite	BlackView X1	YM101	Esperanza	MAX30102
Spo2			~98%	~98%	~97-98%
bpmBR	~73 bpm	~77 bpm	~69 bpm	~75 bpm	~77-80bpm

Εικόνα 6.5.1 Συγκριτικός πίνακας μετρήσεων

Στην παραπάνω εικόνα μπορούμε να δούμε τα συγκριτικά αποτελέσματα στις μετρήσεις που διενεργήθηκαν. Όσον αφορά τις μετρήσεις του οξυγόνου αίματος, παρατηρούμε πως οι αποκλίσεις ανάμεσα στις τρεις συσκευές είναι από ελάχιστες έως μηδενικές. Αντιθέτως, όσον αφορά τους καρδιακούς παλμούς παρατηρούμε πως οι αποκλίσεις είναι μεν μεγαλύτερες, δεν προκαλούν ωστόσο ανησυχία για πιθανά προβλήματα και ενδεχόμενο παραπληροφόρησης.

6.6 Συμπεράσματα

Σε γενικές γραμμές το συμπέρασμα που αντλούμε από τον αισθητήρα MAX30102 συνδεδεμένο στο Raspberry pi 3 Model B+ είναι θετικό. Οι μετρήσεις είναι κοντά με όμοιες συσκευές εφόσον χρησιμοποιηθεί σωστά. Για απλή συσκευή μέτρησης καρδιακών παλμών και κορεσμού οξυγόνου του αίματος είναι μια τίμια επιλογή ωστόσο δεν συνιστάται σε καμία περίπτωση σε κρίσιμες συνθήκες παρακολούθησης ασθενή. Όπως προαναφέρθηκε, δεν απευθύνεται σε ιατρική χρήση, καθώς είναι πολύ ευαίσθητος και μπορεί να οδηγήσει σε εσφαλμένα συμπεράσματα. Για φοιτητές, χομπίστες, μηχανικούς, κατασκευαστές και προγραμματιστές, που θέλουν να ενσωματώσουν ζωντανά δεδομένα καρδιακού παλμού είναι μια πολύ καλή επιλογή.

Ειδικότερα, καθώς ο αισθητήρας είναι συνδεδεμένος αυτόματα στο Raspberry, μπορούμε να αναπτύξουμε το project περαιτέρω προσθέτοντας στην υπάρχουσα κατασκευή περισσότερους αισθητήρες, όπως αισθητήρα θερμοκρασίας σώματος, ποιότητας αέρα κ.λπ., φτιάχνοντας έτσι ένα πιο ολοκληρωμένο σύστημα βιοϊατρικών και περιβαλλοντολογικών μετρήσεων.

7. ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Getting to Know the Raspberry Pi (Code Power: A Teen Programmer's Guide). Rosen Young Adult July 30, 2014
- [2] Jolles, J. W. (2021). Broad-scale applications of the Raspberry Pi: A review and guide for biologists. *Methods in Ecology and Evolution*, 12, 1562–1579.
- [3] Raspberry Pi History. Awais Khan. January 5, 2023
- [4] The History of Raspberry Pi. DevicePlus Editorial Team. January 13, 2023
- [5] C. Severance, "Eben Upton: Raspberry Pi," in *Computer*, vol. 46, no. 10, pp. 14-16, October 2013, doi: 10.1109/MC.2013.349.
- [6] Introduction to Raspberry Pi 3, Syed Zain Nasir, 19 April 2018
(<https://www.theengineeringprojects.com/2018/04/introduction-to-raspberry-pi-3.html>)
- [7] P. Shruthi and R. Resmi, "Heart Rate Monitoring using Pulse Oximetry and development of Fitness Application," 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur, India, 2019, pp. 1568-1570, doi: 10.1109/ICICICT46008.2019.8993398.
- [8] <https://datasheets.maximintegrated.com/en/ds/MAX30102.pdf>
- [9] "Why was Python created in the first place?". General Python FAQ. Python Software Foundation. <http://docs.python.org/faq/general.html#why-was-python-created-in-the-first-place>. Retrieve 22 March 2007.
- [10] Kuchling, Andrew M. (22 December 2006). "Interview with Guido van Rossum (July 1998)". amk.ca. <http://www.amk.ca/python/writing/gvr-interview>. Retrieved 12 March 2012.
- [11] Peters, Tim (19 August 2004). "PEP 20 – The Zen of Python". Python Enhancement Proposals. Python Software Foundation. <http://www.python.org/dev/peps/pep-0020/>. Retrieved 24 November 2008.
- [12] Van Rossum, Guido (1993). "An Introduction to Python for UNIX/C Programmers". Proceedings of the NLUUG najaarsconferentie (Dutch UNIX users group).
(<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.2023>) "even though the design of C is far from ideal, its influence on Python is considerable."
- [13] Interfacing MAX30100 Pulse Oximeter and Heart Rate Sensor with Arduino
(<https://lastminuteengineers.com/max30100-pulse-oximeter-heart-rate-sensor-arduino-tutorial>)
- [14] SKU_SEN0518_Gravity_MAX30102_Heart_Rate_and_Oximeter_Sensor

(https://wiki.dfrobot.com/SKU_SEN0518_Gravity_MAX30102_Heart_Rate_and_Oximeter_Sensor)

[15] Serial Bluetooth Terminal» by Kai Morich, Play Store Application.

[16] Ajay Rawat. A Review on Python Programming, International Journal of Research in Engineering, Science and Management Volume-3, Issue-12, December-2020.

[17] 8 Python Coding Best Practices You Need to Know. App Academy, August 2023

8. ΠΑΡΑΡΤΗΜΑ Α: BaseHeartRateSPO2.py

```
# -*- coding: utf-8 -*-

import serial

import time

import smbus

import os

import math

import RPi.GPIO as GPIO

DEV_ADDRESS = 0x0020

### klash

class BaseHeartRateSpo2():

    SPO2 = 0

    heartbeat = 0

    def __init__(self ,bus):

        self.i2cbus = smbus.SMBus(bus)

    def begin(self):

        global DEV_ADDRESS ### dilwnw oti mporei na xrisimopoihsei thn metavlth DEV_ADDRESS
        poy exei dilwthei pio panw

        rbuf = self.read_reg(0x04,2) ### kalw thn methodo read_reg pou exei dilwthei pio kat

        if (rbuf[0] & 0xff << 8 | rbuf[1]) == DEV_ADDRESS: ### eksakrivonw oti to device address me thn
        timh pou diavazw apton aisthtira einai idia wste na ksekinisw na pernw metriseis

            return True

        else:
```

```

    return False

def sensor_start_collect(self):
    w_buf = [0x00,0x02]
    self.write_reg(0x1C, w_buf)

def get_heartbeat_SPO2(self):
    rbuf = self.read_reg(0x0C,8) ## diavazw plirofories apo ton register: Led Pulse Amplitude sthn
address 0x0C

    SPO2Valid = rbuf[1]
    HeartbeatValid = rbuf[7]
    self.SPO2 = rbuf[0]
    if self.SPO2 == 0:
        self.SPO2 = -1
    self.heartbeat = int(rbuf[2]) << 24 | int(rbuf[3]) << 16 | int(rbuf[4]) << 8 | int(rbuf[5])
    if self.heartbeat == 0:
        self.heartbeat = -1

def sensor_start_collect(self):
    wbuf = [0x00,0x01]
    self.write_reg(0x20, wbuf)

def sensor_end_collect(self):
    wbuf = [0x00,0x02]
    self.write_reg(0x20, wbuf)

class I2Cinterface(BaseHeartRateSpo2):
    def __init__(self ,bus ,addr):

```

```

self.__addr = addr

super(I2Cinterface, self).__init__(bus)

def write_reg(self, reg_addr, data_buf):

    while 1:

        try:

            self.i2cbus.write_i2c_block_data(self.__addr ,reg_addr ,data_buf)

            return

        except:

            print("please check connection!")

            time.sleep(1)

def read_reg(self, reg_addr ,length):#pernaw to 0x0c kai to 8 ws parametrous

    try:

        rslt = self.i2cbus.read_i2c_block_data(self.__addr ,reg_addr , length)

    except:

        rslt = -1

    return rslt

```

9. ΠΑΡΑΡΤΗΜΑ Β: gain_heartbeat_SPO2.py

```
# -*- coding: utf-8 -*-

import sys

import os

import time

import RPi.GPIO as GPIO

from BaseHeartRateSpo2 import *

import atexit

### ανοιγουμε thn porta mesw tis opoia tha staloun ta data ston xrhsth pou syndethike me bluetooth

try:

    f_device1 = open('/dev/rfcomm0','w',1) #send to android via bluettoth rfcomm0 interface

    #f_device2 = open('/dev/rfcomm1','w',1) #send to android via bluettoth rfcomm0 interface

    #f_device3 = open('/dev/rfcomm2','w',1) #send to android via bluettoth rfcomm0 interface

    f_measures = open('/home/c.georgiadis/Desktop/ptuxiaki/measures','w')

except Exception as e:

    print("Error opening file:", e)

    sys.exit(1)

#sys.path.append(os.path.dirname(os.path.dirname(os.path.dirname(os.path.realpath(__file__))))))

### ginete arxikopoihsh twv 2 metavlitwn kai

I2C_1 = 0x01

I2C_ADDRESS = 0x57

### meta ginete h xrhsh ths classhs I2Cinterface pername tis 2 metavlites ws parametrous kai etsi

dimiourgoume ena antikeimeno typou I2Cinterface,

### apothikevoume to antikeimeno pou dimiourgisame sthn metavlhth max30102

max30102 = I2Cinterface(I2C_1 ,I2C_ADDRESS)
```

```

### h methodos setup

def setup():

    ### kalei thn methodo begin() ths klashs I2Cinterface mesw tou object pou dimiourgisame

    ### kai oso afth (h methodos) epistrefei False tote kanei print oti apetyxe vasei to thread na koimithei
gia 1 second kai ksanaprosathei

    while (False == max30102.begin()):

        print("init fail!")

        time.sleep(1)

    print("Start measuring...")

    max30102.sensor_start_collect()

    time.sleep(1)

### h methodos loop

def loop():

    ### pairnei to oksigono kai to heartbeat apo ton aisthitira mesw ths methodou get_heartbeat_SPO2()

    ### kai to stelnei sto connection pou anoikse panw

    max30102.get_heartbeat_SPO2()

    f_device1.write("\n SPO2 is : "+str(max30102.SPO2)+"% \n Heart Rate is :"+
+str(max30102.heartbeat)+ " Times/min \n")

    #f_device2.write("\n SPO2 is : "+str(max30102.SPO2)+"% \n Heart Rate is :"+
+str(max30102.heartbeat)+ " Times/min \n")

    #f_device3.write("\n SPO2 is : "+str(max30102.SPO2)+"% \n Heart Rate is :"+
+str(max30102.heartbeat)+ " Times/min \n")

    f_measures.write("\n SPO2 is : "+str(max30102.SPO2)+"% \n Heart Rate is :"+
+str(max30102.heartbeat)+ " Times/min \n")

    f_measures.flush()

```

```

print("\n SPO2 is : "+str(max30102.SPO2)+"% \n Heart Rate is :"+str(max30102.heartbeat)+ "
Times/min \n")

time.sleep(1)

### Ksekinaei h efarmogh

if __name__ == "__main__":

    try:

        setup() ### Kaleitai h methodos setup (apo poio panw)

        while True: ### Kai kaleitai h methodos loop (apo poio panw) gia olh thn diarkeia

            loop()

    except KeyboardInterrupt:

        pass

```

10. ΠΑΡΑΡΤΗΜΑ Γ: Button.py

```
import subprocess

import RPi.GPIO as GPIO

import time

import psutil

import os

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BOARD)

GPIO.setup(10, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

button_pin = 10

running = False

process = None

def start_program():

    global running, process

    if not running:

        process = subprocess.Popen(["lxterminal", "-e", "python3",

"/home/c.georgiadis/Desktop/ptuxiaki/gain_heartbeat_SPO2.py"], preexec_fn=os.setsid)

        running = True

        print("Program started")

def stop_program():

    global running, process

    if running:

        running = False

        print("Program stopped")
```

```

for proc in psutil.process_iter(['cmdline']):
    if 'python3' in proc.info['cmdline']:
        proc.kill()
        break

def toggle_program(channel):
    global running
    if not running:
        start_program()
    else:
        stop_program()

try:
    GPIO.add_event_detect(button_pin, GPIO.FALLING, callback=toggle_program, bouncetime=200)

    while True:
        time.sleep(1) # Do other things while waiting for button events

except KeyboardInterrupt:
    GPIO.cleanup()

```