



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**  
**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΪΑΤΡΙΚΗΣ**

**Π.Μ.Σ ΠΡΟΗΓΜΕΝΑ ΣΥΣΤΗΜΑΤΑ & ΜΕΘΟΔΟΙ**  
**ΣΤΗ ΒΙΟΪΑΤΡΙΚΗ ΤΕΧΝΟΛΟΓΙΑ**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**«Real-Time παρακολούθηση ζωτικών παραμέτρων & συνθηκών  
χώρου μέσω Wi-Fi με τη χρήση του ESP8266»**

**Τσάλτας Θεόδωρος**  
**(Α.Μ 1916)**

Επιβλέπων  
**ΑΣΒΕΣΤΑΣ ΠΑΝΤΕΛΕΗΜΩΝ**  
**(Αναπληρωτής Καθηγητής)**

Σεπτέμβριος, Αθήνα 2022

## Η Τριμελής Εξεταστική Επιτροπή

Ο Επιβλέπων Καθηγητής

Όνομα Επώνυμο

Όνομα Επώνυμο

Όνομα Επώνυμο

[ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ]

[ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ]

[ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ]

## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο υπογράφων **ΤΣΑΛΤΑΣ ΘΕΟΔΩΡΟΣ** του **ΓΕΩΡΓΙΟΥ**, με αριθμό μητρώου **1916** φοιτητής του Τμήματος **ΜΗΧΑΝΙΚΩΝ ΒΙΟΪΑΤΡΙΚΗΣ** της Σχολής **ΜΗΧΑΝΙΚΩΝ** του Πανεπιστημίου Δυτικής Αττικής, δηλώνω υπεύθυνα ότι:


«Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του διπλώματός μου».

Ημερομηνία

23/09/2022

Ο Δηλών



Τσάλτας Θεόδωρος

## Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω θερμά τον Καθηγητή κ. Παντελή Ασβεστά, τόσο για την ανάθεση της συγκεκριμένης διπλωματικής εργασίας, όσο και για τις πολύτιμες συμβουλές και διορθώσεις του. Η καθοδήγησή του, σε κρίσιμα σημεία της εργασίας ήταν καθοριστική. Είναι τιμή μου που συνεργάστηκα μαζί του αυτόν τον ενάμιση χρόνο στα πλαίσια της εκπόνησης της εργασίας, καθώς και κατά τη διάρκεια της φοίτησής μου στη σχολή.

Επίσης θα ήθελα να ευχαριστήσω την μητέρα μου Μαρία, που μου έδινε κουράγιο τις δύσκολες στιγμές και για την συμπαράστασή της για την ολοκλήρωση αυτής της διπλωματικής.

Τέλος θα ήθελα να ευχαριστήσω τον κ. Χάρη Χοϊδά για την πολύτιμη καθοδήγησή του στην γλώσσα προγραμματισμού PHP. Με τη συνεισφορά του κατάφερα να κατανοήσω τις βασικές αρχές της PHP και να ολοκληρώσω ένα κομβικό κομμάτι της εργασίας μου.



## Πίνακας περιεχομένων

ΠΕΡΙΛΗΨΗ .....	8
ABSTRACT .....	9
ΠΡΟΛΟΓΟΣ .....	10
<b>1. ΕΙΣΑΓΩΓΗ .....</b>	<b>11</b>
1.1 ΜΟΝΑΔΕΣ ΕΝΤΑΤΙΚΗΣ ΘΕΡΑΠΕΙΑΣ (ΜΕΘ) .....	11
1.2 ΚΑΤΗΓΟΡΙΕΣ ΜΕΘ .....	11
1.2.1 Πολυδύναμη .....	11
1.2.2 Νεογνική .....	12
1.2.3 Παιδιατρική.....	13
1.2.4 Μετεγχειρητική.....	13
1.3 Ο ΡΟΛΟΣ ΤΗΣ ΜΕΘ ΣΤΗΝ ΕΠΟΧΗ ΤΟΥ COVID .....	13
1.4 ΕΞΟΠΛΙΣΜΟΣ ΣΤΗ ΜΕΘ .....	14
1.4.1 Αναπνευστήρες.....	14
1.4.2 Κρεβάτια Νοσηλείας.....	17
1.4.3 Θερμοκοιτίδες.....	18
1.4.4 Υγραντήρες.....	19
<b>2. ΓΕΝΙΚΕΣ ΑΡΧΕΣ ΣΥΣΤΗΜΑΤΩΝ ΕΠΙΤΗΡΗΣΗΣ ΑΣΘΕΝΟΥΣ .....</b>	<b>22</b>
2.1 ΑΝΑΙΜΑΚΤΗ ΠΙΕΣΗ (NIBP) .....	23
2.2 SpO2 .....	25
2.3 ECG.....	30
2.4 ΡΥΘΜΟΣ ΑΝΑΠΝΟΗΣ .....	35
2.5 ΚΑΠΝΟΓΡΑΦΙΑ .....	36
2.6 ΘΕΡΜΟΚΡΑΣΙΑ .....	38
2.7 ΚΕΝΤΡΙΚΟ ΣΥΣΤΗΜΑ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ (CENTRAL MONITORING SYSTEM).....	39
<b>3. ΥΛΙΣΜΙΚΟ.....</b>	<b>42</b>
3.1 ΜΙΚΡΟΕΛΕΓΚΤΗΣ ESP8266 NODEMCU .....	43
3.1.1 Τμήμα ESP-12E.....	43
3.1.2 Απαιτήσεις ισχύος.....	44
3.1.3 Περιφερειακά & Είσοδοι / Έξοδοι (I/O).....	44
3.1.4 Ενδείξεις LED & Κουμπιά στην Πλακέτα .....	45
3.1.5 Σειριακή Επικοινωνία.....	46
3.1.6 Ακροδέκτες του ESP8266 NodeMCU.....	46
3.1.7 Σύγκριση ESP8266 NodeMCU με το Arduino .....	51
3.2 ΑΙΣΘΗΤΗΡΑΣ DS18B20 .....	52
3.2.1 Γενικές πληροφορίες.....	52
3.2.2 Ακροδέκτες του DS18B20 και σύνδεση με τον μικροελεγκτή.....	54
3.2.3 Εσωτερική Δομή του DS18B20.....	55
3.2.4 Αρχή λειτουργίας DS18B20.....	55
3.2.5 One-Wire Bus .....	57
3.3 ΑΙΣΘΗΤΗΡΑΣ ΘΕΡΜΟΚΡΑΣΙΑΣ & ΥΓΡΑΣΙΑΣ DHT22 .....	58
3.3.1 Γενικές πληροφορίες.....	58
3.3.2 Σύγκριση DHT11 με DHT22 .....	59
3.3.3 Επισκόπηση υλισμικού .....	59
3.3.4 Μορφολογία Ακροδεκτών .....	60
3.3.5 Συνδεσμολογία με τον μικροελεγκτή.....	61

3.4 ΑΙΣΘΗΤΗΡΑΣ SpO2 & HEART RATE MAX30102 .....	63
3.4.1 Γενική επισκόπηση αισθητήρα .....	63
3.4.2 Τεχνικά χαρακτηριστικά.....	64
3.4.3 Τρόπος λειτουργίας MAX30102.....	65
3.4.4 Ακροδέκτες & συνδεσμολογία.....	66
<b>4. ΛΟΓΙΣΜΙΚΟ.....</b>	<b>68</b>
4.1 ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΤΟ ARDUINO IDE .....	68
4.1.1 Εγκατάσταση του Arduino IDE .....	68
4.1.2 Εγκατάσταση βιβλιοθήκης DHT22 .....	71
4.1.3 Εγκατάσταση της βιβλιοθήκης για τον αισθητήρα DS18B20.....	72
4.1.4 Εγκατάστασης της βιβλιοθήκης για τον αισθητήρα MAX30102.....	73
4.1.5 Εγκατάσταση βιβλιοθήκης για την εφαρμογή Blynk .....	74
4.2 ΕΠΕΞΗΓΗΣΗ ΚΩΔΙΚΑ ARDUINO IDE .....	74
4.3 ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΡΥΘΜΙΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ BLYNK.....	79
4.3.1 Εφαρμογή στο Smartphone .....	79
4.3.2 Ρύθμιση του Blynk στην Web εφαρμογή.....	81
4.4 ΔΗΜΙΟΥΡΓΙΑ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ .....	83
4.5 ΕΠΕΞΗΓΗΣΗ ΚΩΔΙΚΑ PHP .....	85
4.5.1 Σύνδεση της βάσης δεδομένων με τον μικροελεγκτή μέσω PHP .....	85
4.5.2 Απεικόνιση και φιλτράρισμα των αποτελεσμάτων της βάσης δεδομένων σε ιστοσελίδα .....	87
<b>5. ΑΠΟΤΕΛΕΣΜΑΤΑ .....</b>	<b>90</b>
5.1 ΟΛΟΚΛΗΡΩΜΕΝΟ ΚΑΤΑΣΚΕΥΑΣΤΙΚΟ ΣΥΣΤΗΜΑ.....	90
5.2 ΠΕΙΡΑΜΑΤΑ - ΈΛΕΓΧΟΙ .....	94
5.2.1 Έλεγχος για την αξιοπιστία των μετρήσεων SpO2 & Heart Rate .....	94
5.2.2 Έλεγχοι και αποτελέσματα πειραματικής διάταξης SpO2 & Heart Rate .....	97
5.2.3 Έλεγχοι και αποτελέσματα πειραματικής διάταξης για την μέτρηση θερμοκρασίας σώματος.....	102
5.2.4 Έλεγχοι και αποτελέσματα πειραματικής διάταξης για την μέτρηση θερμοκρασίας και υγρασίας χώρου .....	103
<b>6. ΣΥΜΠΕΡΑΣΜΑΤΑ &amp; ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ .....</b>	<b>105</b>
6.1 ΣΥΜΠΕΡΑΣΜΑΤΑ.....	105
6.2 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ .....	106
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ .....</b>	<b>107</b>
<b>ΚΩΔΙΚΑΣ .....</b>	<b>114</b>
ΚΩΔΙΚΑΣ ARDUINO.....	114
ΚΩΔΙΚΑΣ PHP (SENSOR-DATA.PHP) .....	124
ΚΩΔΙΚΑΣ PHP (POST-SENSOR-DATA.PHP).....	130



## Περίληψη

Μια από τις πιο σημαντικές ιατρικές συσκευές σε ένα νοσοκομείο είναι τα μόνιτορ παρακολούθησης. Στόχος των μόνιτορ είναι η συνεχής καταγραφή και παρακολούθηση πολλαπλών ζωτικών παραμέτρων ταυτόχρονα. Σκοπός της παρούσης εργασίας είναι η κατασκευή ενός συστήματος απομακρυσμένης επιτήρησης ζωτικών λειτουργιών και συνθηκών χώρου σε πραγματικό χρόνο. Στο πρώτο σκέλος της εργασίας χρησιμοποιείται η αναπτυξιακή πλατφόρμα ESP8266, η οποία συνδέεται ασύρματα στο WiFi, μέσω της οποίας αρχικοποιούνται αισθητήρες για την καταγραφή των Παλμών ανά λεπτό (BPM), θερμοκρασία σώματος και το Ποσοστό Οξυγόνου στο αίμα (SpO2) ενός ασθενούς καθώς και η θερμοκρασία και υγρασία του χώρου όπου βρίσκεται ο ασθενής. Στη συνέχεια οι καταγραφές των αισθητήρων μπορούν να απεικονιστούν σε πραγματικό χρόνο σε οποιοδήποτε Browser Η/Υ καθώς και στην ειδικά σχεδιασμένη εφαρμογή Blynk η οποία είναι διαθέσιμη σε Smartphone. Στο δεύτερο σκέλος της εργασίας κατασκευάζεται μια βάση δεδομένων MySQL στην οποία αποθηκεύονται όλες οι τιμές των αισθητήρων. Παράλληλα τα αποτελέσματα μπορούν να απεικονιστούν σε ξεχωριστή ιστοσελίδα με σκοπό την μεταγενέστερη ανάλυσή τους και μέσω φίλτρων και κριτηρίων αναζήτησης να εμφανιστούν συγκεκριμένα αποτελέσματα. Τέλος έγιναν στατιστικοί έλεγχοι ώστε να αξιολογηθεί η αξιοπιστία και η ακρίβεια των μετρήσεων των αισθητήρων. Διαπιστώθηκε ότι το σύστημα είναι σε θέση να καταγράψει με πιστότητα και ακρίβεια τόσο τις συνθήκες χώρου όσο και τις ζωτικές παραμέτρους.

**Λέξεις Κλειδιά:** ESP8266, Βάση Δεδομένων, PHP, DHT22, DS18B20, MAX30102

## **Abstract**

One of the most crucial medical devices in a hospital are patient monitors. The main goal of medical monitors is the continuous measurement and recording of multiple vital signs at the same time. The purpose of this thesis is the construction of a real-time, remote system that has the ability to monitor the vital signs of a patient, as well as the room conditions the patient is in. In the first part of the project the ESP8266 Developer Board is used, which is connected to the existing WiFi Network. Through the ESP8266, sensors responsible for the recording of Pulses per Minute (BPM), Body Temperature and Oxygen Saturation in Blood of a patient are initialized as well as sensors responsible for the monitoring of room conditions. Next, all the measurements taken from the sensors can be displayed in real-time to both any PC Browser and an especially designed Smartphone app named Blynk. Blynk is available to any Smartphone App Store. In the second part of the project a MySQL Database is constructed which is able to store all the recordings from the sensors. At the same time, those recordings can be displayed to a separate web page and via filters and search criteria, specific results can be shown. Lastly, statistical tests were conducted in order to evaluate the reliability and precision of the measurements. It was concluded that the system is in position to measure properly the room conditions as well as the vital signs.

**Keywords:** ESP8266, Database, PHP, DHT22, DS18B20, MAX30102

## Πρόλογος

Μια από τις πιο χρήσιμες ιατροτεχνολογικές συσκευές ενός νοσοκομείου είναι τα μόνιτορ παρακολούθησης ζωτικών παραμέτρων. Ο λόγος που χρησιμοποιούνται ευρέως είναι γιατί παρέχουν πολύτιμες πληροφορίες στο νοσηλευτικό και ιατρικό προσωπικό σχετικά με την κλινική εικόνα του ασθενή. Υπάρχουν διαφορετικά είδη μόνιτορ αλλά όλα έχουν κάποιες κοινές λειτουργίες όπως την μέτρηση καρδιακών παλμών (BPM), Οξυμετρίας (SpO<sub>2</sub>), Μη αιματηρής πίεσης (NIBP), Θερμοκρασίας Σώματος (T<sub>1</sub>) και Ηλεκτροκαρδιογραφήματος (ΗΚΓ).

Στην **Εισαγωγή** παρουσιάζονται οι διάφορες κατηγορίες ΜΕΘ, ο ρόλος της ΜΕΘ στην περίοδο του Covid-19 και ο κυριότερος ιατροτεχνολογικός εξοπλισμός που βρίσκεται σε αυτές.

Στις **Γενικές Αρχές Συστημάτων Επιτήρησης Ασθενούς** παρουσιάζονται τα χαρακτηριστικά των μόνιτορ παρακολούθησης, οι διάφορες λειτουργίες που έχουν, ο τρόπος απαγωγής των σημάτων καθώς και η χρησιμότητά τους σε κρίσιμες μονάδες ενός νοσοκομείου.

Στο **Υλισμικό** παρουσιάζονται τα εξαρτήματα τα οποία χρησιμοποιήθηκαν. Συγκεκριμένα αναλύονται οι έννοιες των αισθητήρων και της πλατφόρμας ESP8266.

Στο **Λογισμικό** παρουσιάζεται ο κώδικας που χρησιμοποιήθηκε. Αναλύονται τα προγράμματα που χρησιμοποιήθηκαν, ποιες βιβλιοθήκες εγκαταστάθηκαν καθώς και μια εκτενής ανάλυση του κώδικα.

Στα **Αποτελέσματα** παρουσιάζονται τα αποτελέσματα της εργασίας μαζί με στατιστική ανάλυσή τους για να εξακριβωθεί η ορθότητά τους

Στα **Συμπεράσματα & Μελλοντική Εργασία** παρουσιάζονται τα συμπεράσματα που προέκυψαν από την εργασία καθώς και μελλοντικές επεκτάσεις που θα μπορούσαν να εφαρμοστούν ώστε να εξελιχθεί το σύστημα.

# 1. ΕΙΣΑΓΩΓΗ

## 1.1 ΜΟΝΑΔΕΣ ΕΝΤΑΤΙΚΗΣ ΘΕΡΑΠΕΙΑΣ (ΜΕΘ)

Από τότε που εμφανίστηκαν για πρώτη φορά περίπου μισό αιώνα πριν, οι μονάδες εντατικής θεραπείας (ΜΕΘ) έχουν γίνει ένα καίριας σημασίας κομμάτι του συστήματος ιατροφαρμακευτικής περίθαλψης. Η ΜΕΘ είναι ένα οργανωμένο σύστημα το οποίο διαθέτει, σε άρρωστους οι οποίοι βρίσκονται σε κρίσιμη κατάσταση, εξειδικευμένη ιατρική και νοσηλευτική φροντίδα, βελτιωμένη δυνατότητα παρακολούθησης ζωτικών παραμέτρων (monitoring) και πολλαπλά μέσα υποστήριξης ζωτικών οργάνων. Η ΜΕΘ βρίσκεται σε 24ωρη ετοιμότητα για αντιμετώπιση επειγόντων περιστατικών από κάθε τμήμα του νοσοκομείου όπως το Τμήμα Επειγόντων Περιστατικών, τα Χειρουργεία, το Μαιευτήριο και τις Μονάδες Νοσηλείας.

Η ΜΕΘ είναι ένα ξεχωριστό και αυτόνομο τμήμα εντός ενός νοσοκομείου, εξοπλισμένο με παροχές υψηλής τεχνολογίας, που έχουν σχεδιαστεί για την στενή παρακολούθηση, ταχεία παρέμβαση και συχνά εκτεταμένη θεραπεία ασθενών με σοβαρή δυσλειτουργία οργάνων. Ο σκοπός της Εντατικής Φροντίδας είναι να διατηρήσει τις ζωτικές λειτουργίες με σκοπό να αποτραπεί η περαιτέρω ζωτική επιδείνωση, να μειώσει τη θνητότητα και να αποτρέψει τη θνητότητα σε σοβαρά ασθενείς.[1],[2]

## 1.2 Κατηγορίες ΜΕΘ

Μια ΜΕΘ μπορεί να κατηγοριοποιηθεί ανάλογα με τις ειδικότητες που καλύπτει σε Πολυδύναμη ή Ειδική. Οι συνηθέστερες ΜΕΘ είναι οι εξής:

### 1.2.1 Πολυδύναμη

Η Πολυδύναμη μονάδα εντατικής θεραπείας προορίζεται για τη φροντίδα ενήλικων ασθενών οι οποίοι πάσχουν από παθήσεις που απαιτούν τακτική παρακολούθηση, εξειδικευμένη παρακολούθηση και θεραπευτική αγωγή. Σε αυτές τις παθήσεις περιλαμβάνεται η διαβητική κετοξέωση, η γαστρεντερική αιμορραγία, η υπερβολική

κατανάλωση ναρκωτικών ουσιών, η αναπνευστική ανεπάρκεια, η σήψη, το εγκεφαλικό και ο καρκίνος.[2]



Εικόνα 1. ΜΕΘ, Ιδιωτική Κλινική ΜΗΤΕΡΑ [3]

### 1.2.2 Νεογνική

Η νεογνική ΜΕΘ (MENN) είναι υπεύθυνη για την ιατρική παρακολούθηση των πρόωρων, υψηλού ρίσκου και σε κρίσιμη κατάσταση νεογνών.

Η μονάδα εντατικής νοσηλείας νεογνών (MENN) έχει εξειδικευμένο ιατρικό προσωπικό και εξοπλισμό για την φροντίδα πρόωρων και άρρωστων νεογέννητων παιδιών. Στον εξοπλισμό που στελεχώνει την MENN και έχει αναπόσπαστο κομμάτι περιλαμβάνονται οι θερμοκοιτίδες, οι αναπνευστήρες νεογνών, φορητά μόνιτορ & οξύμετρα κ.ά. Αυτό το τμήμα του νοσοκομείου αναφέρεται μερικές φορές και ως μονάδα εντατικής θεραπείας νεογέννητων. Τα μωρά μπορεί να χρειαστεί να εισαχθούν στην MENN εάν:

- Έχουν γεννηθεί νωρίτερα από τις 37 εβδομάδες κύησης
- Υπήρξαν επιπλοκές κατά τη διάρκεια του τοκετού
- Υπάρχουν επιπλοκές όπως αναπνευστικά προβλήματα, μολύνσεις, γενετικές ανωμαλίες ή αν το μωρό χρειάζεται εγχείρηση
- Έχουν χαμηλό βάρος κατά τη γέννηση (χαμηλότερο από 2.5 Kg)
- Είναι δίδυμα, τρίδυμα [4],[5]



Εικόνα 2. ΜΕΝΝ, Ιδιωτική Κλινική ΜΗΤΕΡΑ [6]



### 1.2.3 Παιδιατρική

Η παιδιατρική μονάδα εντατικής θεραπείας είναι μια ειδικευμένη μονάδα του νοσοκομείου όπου νοσηλεύονται τα πιο άρρωστα παιδιά. Στην παιδιατρική ΜΕΘ εισάγονται ασθενείς από νεογνά (εάν έχουν πάει σπίτι μετά το νοσοκομείο, αλλιώς εισάγονται στην νεογνική ΜΕΘ) μέχρι ανήλικα παιδιά σε νεαρή ηλικία. Οι ασθενείς στην παιδιατρική ΜΕΘ μπορεί να νοσηλεύονται, λόγω αναπνευστικών δυσκολιών, μολύνσεων, γενετικών και μεταβολικών διαταραχών, τραυμάτων και ατυχημάτων, παιδιά που μόλις υπεβλήθησαν σε χειρουργείο και βρίσκονται υπό το φόβο επιδείνωσης και πολλά άλλα. [2],[7]



Εικόνα 3. ΜΕΘ ΠΚΧ, Ιδιωτική Κλινική ΜΗΤΕΡΑ [8]

### 1.2.4 Μετεγχειρητική

Η μετεγχειρητική ΜΕΘ απευθύνεται, όπως προδίδει και το όνομά της, στην ιατρική παρακολούθηση ασθενών μετά το πέρας της χειρουργικής πράξης. Σε αυτή την κατηγορία περιλαμβάνονται ασθενείς που έχουν υποβληθεί σε σοβαρές χειρουργικές επεμβάσεις (π.χ κοιλίας, κρανιοτομής, θωρακοτομής κ.ά), βρίσκονται σε κρίσιμη κατάσταση και χρήζουν συνολικής φροντίδας, συνεχούς παρακολούθησης (monitoring) και πιθανών χειρουργικών παρεμβάσεων. [2],[9]

## 1.3 Ο Ρόλος της ΜΕΘ στην εποχή του COVID

Όπως αναφέρθηκε και προηγουμένως η ΜΕΘ παρέχει φροντίδα σε ασθενείς οι οποίοι βρίσκονται σε κρίσιμη κατάσταση και χρειάζονται ιατρικά μηχανήματα για τους διατηρήσουν στη ζωή. Εκτός από τους ασθενείς που πάσχουν από σοβαρά τραύματα, καρδιακή ανεπάρκεια και εγκαύματα, συμπεριλαμβάνονται πλέον και ασθενείς που πάσχουν από Covid-19.

Οι ασθενείς που είναι σοβαρά άρρωστοι με Covid-19 τείνουν να χρειάζονται υψηλότερα επίπεδα φροντίδας από τους τυπικούς ασθενείς ΜΕΘ. Συνήθως πρέπει να είναι διασωληνωμένοι με αναπνευστήρα ή άλλο παραπλήσιο εξοπλισμό ώστε να

τους διατηρήσει στη ζωή και χρειάζονται ιατρική παρακολούθηση από πολλαπλά μέλη του ιατρικού και νοσηλευτικού προσωπικού. Επίσης έχει διαπιστωθεί ότι οι ασθενείς με κορονοϊό τείνουν να νοσηλεύονται για περισσότερο χρονικό διάστημα στη ΜΕΘ. Ενδεικτικά ο μέσος ασθενής που θα διασωληνωθεί με αναπνευστήρα θα χρειαστεί μερικές μέρες, ενώ οι ασθενείς με COVID τουλάχιστον 7-10 μέρες. [10],[11]

## **1.4 Εξοπλισμός στη ΜΕΘ**

Η ΜΕΘ είναι ένας χώρος όπου νοσηλεύονται τα πιο σοβαρά περιστατικά ενός νοσοκομείου. Για το λόγο αυτό είναι επιτακτική ανάγκη να είναι εξοπλισμένη με τελευταίας τεχνολογίας ιατροτεχνολογικό εξοπλισμό. Τα πιο κοινά ιατροτεχνολογικά μηχανήματα μεταξύ των ΜΕΘ για την υποστήριξη αλλά και παρακολούθηση των ζωτικών παραμέτρων των ασθενών είναι τα παρακάτω:

### **1.4.1 Αναπνευστήρες**

Ένας αναπνευστήρας είναι ένα ιατροτεχνολογικό μηχάνημα το οποίο παρέχει μηχανικό αερισμό μεταφέροντας χάρη σε έναν μείκτη το επιθυμητό μείγμα O<sub>2</sub> - αέρα εντός και εκτός των πνευμόνων μέσω των άνω αεροφόρων οδών, διανέμει αναπνοές σε έναν ασθενή που είναι σωματικά ανήμπορος να αναπνεύσει μόνος του ή αναπνέει ανεπαρκώς. Οι αναπνευστήρες είναι αυτοματοποιημένα, ελεγχόμενα μέσω μικροεπεξεργαστών μηχανήματα, αλλά οι ασθενείς μπορούν επίσης να λάβουν αέρα μέσω χειροκίνητου ασκού. Για να επιτευχθεί αυτό, οι ασθενείς διασωληνώνονται με το μηχάνημα είτε μέσω ενδοτραχειακού σωλήνα ή μη επεμβατικά με μια μάσκα.

Ο μηχανικός αερισμός χρησιμοποιεί συσκευές θετικής πίεσης ώστε να βελτιώσει την ανταλλαγή οξυγόνου (O<sub>2</sub>) και διοξειδίου του άνθρακα (CO<sub>2</sub>). Οι δυο κύριοι στόχοι του μηχανικού αερισμού είναι οι εξής:

1. Να διατηρήσει κατάλληλα επίπεδα O<sub>2</sub> και CO<sub>2</sub> στις αρτηρίες,
2. Να μειώσει το έργο αναπνοής του ασθενούς.

Ο μηχανικός αερισμός χρησιμοποιείται ώστε να διασφαλιστεί μια ελεγχόμενη αναπνευστική οδός για ασθενείς που χρειάζονται νάρκωση, όπως για παράδειγμα κατά τη διάρκεια χειρουργικών επεμβάσεων, ή για να μπορέσει να επιτευχθεί τεχνητή αναπνοή και υποστήριξη ζωής. Άλλοι στόχοι του περιλαμβάνουν την οξυγόνωση, την συνολική αναπνοή και την μείωση του έργου αναπνοής. Ο μηχανικός αερισμός είναι μια παρέμβαση με υποστηρικτικό χαρακτήρα και δεν θεραπεύει τις υποκείμενες παθήσεις.

Οι αναπνευστήρες αποτελούνται από τα παρακάτω στοιχειώδη στοιχεία:

- Τη μονάδα ρυθμιστικής βαλβίδας για το αέριο εισόδου, η οποία αποτελείται από ρυθμιστικές βαλβίδες οι οποίες είναι υπεύθυνες για την διατήρηση της

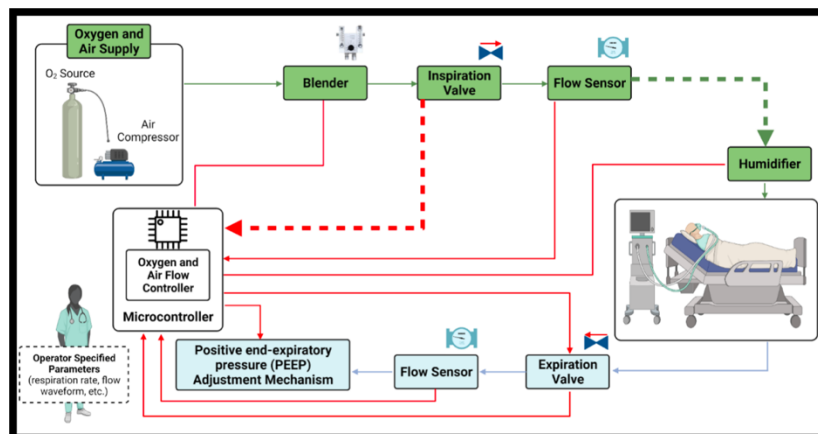
πίεσης των αερίων όπως οξυγόνο και πεπιεσμένο αέρα εντός των λειτουργικών ορίων του εξοπλισμού.

- Τον μείκτη ο οποίος είναι υπεύθυνος για την μείξη των αερίων εντός του εύρους της συγκέντρωσης αναφοράς του οξυγόνου το οποίο κυμαίνεται από 21%-100%.
- Τον μικροελεγκτή (ή αλλιώς μονάδα ελέγχου) ο οποίος διαμορφώνει τον εξοπλισμό στις επιθυμητές λειτουργίες, ελέγχει όλες τις υπόλοιπες μονάδες και διαχειρίζεται τους συναγερμούς και τα δεδομένα της συσκευής. Οι βαλβίδες ελέγχου, εκτός από τους αισθητήρες ροής και πίεσης, είναι κομβικής σημασίας στοιχεία τα οποία είναι επίσης άμεσα συνδεδεμένα με τη μονάδα ελέγχου.
- Η μονάδα τροφοδοσίας η οποία είναι υπεύθυνη για την παροχή ενέργειας στο σύστημα από το δίκτυο ή από μπαταρίες.
- Η μονάδα χειρισμού η οποία περιλαμβάνει οθόνη, οθόνη αφής, κομβία και πληκτρολόγιο. Αυτή η μονάδα είναι υπεύθυνη για την επικοινωνία ανάμεσα στον αναπνευστήρα και τον χειριστή.
- Εξωτερικό κύκλωμα ασθενούς που είναι υπεύθυνο για τη “σύνδεση” του ασθενή με το μηχάνημα μέσω μάσκας και την δημιουργία κύκλων εισπνοής - εκπνοής.



Εικόνα 4. Κοινά Εμπορικοί Αναπνευστήρες [12]

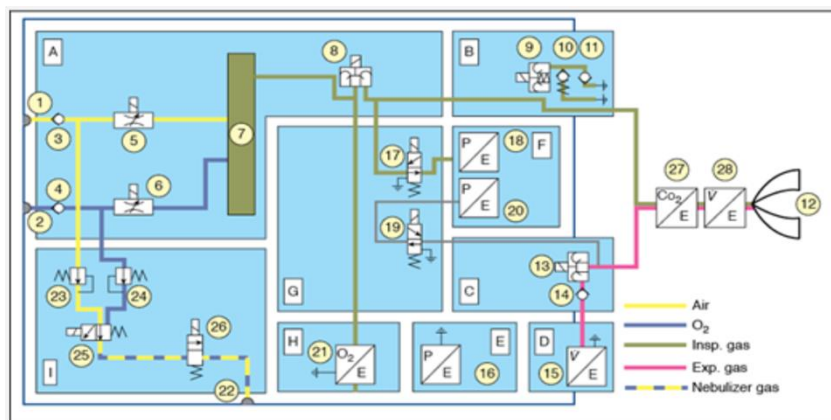
Στην εικόνα 5 φαίνεται το μπλοκ διάγραμμα ενός βασικού αναπνευστήρα. Οι παροχές αέρα και οξυγόνου είναι συνδεδεμένες με βαλβίδες και αισθητήρες ροής. Ο μικροελεγκτής επιτηρεί την έξοδο των αισθητήρων ροής και ελέγχει τις βαλβίδες



Εικόνα 5. Μπλοκ Διάγραμμα Αναπνευστήρα με τα κύρια μέρη του και τη βασική αρχή λειτουργίας του [15]

κατάλληλα. Με αυτόν τον τρόπο ο αναπνευστήρας παράγει αέρα, εμπλουτισμένο με συγκεκριμένη συγκέντρωση σε οξυγόνο. Στη φάση της εισπνοής, το μηχάνημα ενεργοποιεί τη βαλβίδα εισπνοής ενώ απενεργοποιεί αυτή της εκπνοής. Η λογική πίσω από τον τρόπο λειτουργίας της βαλβίδας εισπνοής είναι να ενεργοποιείται και να απενεργοποιείται με τέτοιο τρόπο έτσι ώστε να παρέχονται στον ασθενή αναπνοές με συγκεκριμένη διάρκεια, όγκο ή πίεση.

**Στην εικόνα 6** παρουσιάζεται το πνευματικό διάγραμμα ενός αναπνευστήρα.



Εικόνα 6. Πνευματικό Διάγραμμα Αναπνευστήρα MEΘ Draeger[14]

Στο **A** τμήμα βρίσκεται η μονάδα μείξης και μέτρησης αερίων. Οι επιτοίχιες παροχές αερίων τροφοδοτούν τον αναπνευστήρα με αέρα και οξυγόνο (1,2) μέσω των συνδετικών που διαθέτει. Επίσης για να αποτραπεί η μίξη του ενός αερίου με το άλλο χρησιμοποιούνται δυο αντεπίστροφες βαλβίδες (3,4). Τέλος δυο ρυθμιστικές βαλβίδες (5,6) ελέγχουν την μείξη των αερίων η οποία λαμβάνει μέρος στο δοχείο (7). Τον έλεγχο της εισπνευστικής ροής τον αναλαμβάνει μια επιπλέον βαλβίδα (8).

**B.** Η μονάδα εισπνοής απαρτίζεται από τρεις βαλβίδες: μια ασφαλείας (9) και δυο αντεπίστροφες (10,11). Σε κανονικές συνθήκες η βαλβίδα ασφαλείας είναι απενεργοποιημένη έτσι ώστε να φτάνει στους πνεύμονες του ασθενούς η εισπνευστική ροή (12). Η βαλβίδα ασφαλείας είναι ενεργοποιημένη κατά τη διάρκεια που ο αναπνευστήρας βρίσκεται σε κατάσταση αναμονής έτσι ώστε να είναι εφικτή η αυθόρμητη εισπνοή μέσω της βαλβίδας έκτακτης ανάγκης (11). Σαν εναλλακτική λύση σε περίπτωση που μπλοκάρει η βαλβίδα εκπνοής (13) λειτουργεί η βαλβίδα εκπνοής έκτακτης ανάγκης (10).

**C.** Η μονάδα που είναι υπεύθυνη για την εκπνοή αποτελείται από την ομώνυμη βαλβίδα (13), η οποία χρησιμεύει στην διαμόρφωση της πίεσης στο αναπνευστικό κύκλωμα του ασθενή, καθώς και μια αντεπίστροφη βαλβίδα. Σε συνδυασμό με την βαλβίδα έκτακτης παροχής αέρα (10), η αντεπίστροφη βαλβίδα (14) εμποδίζει την παλινδρομική αναπνοή κατά τη διάρκεια της αυθόρμητης αναπνοής.

**D.** Αισθητήρας εκπνευστικής ροής.

**E.** Αισθητήρας βαρομετρικής πίεσης

**F.** Μονάδα μέτρησης της πίεσης. Η πίεση στο κύκλωμα ασθενούς μετράται με δυο ανεξάρτητους αισθητήρες πίεσης (18,20)

**G.** Μονάδα καλιμπραρίσματος. Οι αισθητήρες πίεσης συχνά καλιμπράρονται στο μηδέν μέσω δυο βαλβίδων καλιμπραρίσματος (17,19)

**H.** Αισθητήρας οξυγόνου

**I.** Νεφελοποιητής [13],[14],[15],[16]

### **1.4.2 Κρεβάτια Νοσηλείας**

Από τεχνικής άποψης τα κρεβάτια νοσηλείας σε μια ΜΕΘ εντάσσονται στον βιοϊατρικό εξοπλισμό ενός νοσοκομείου. Είναι κρίσιμος εξοπλισμός για τον χώρο της ΜΕΘ καθώς οι ασθενείς περνούν μεγάλο χρονικό διάστημα της νοσηλείας τους, είτε επειδή είναι υπό την επήρεια νάρκωσης είτε επειδή δεν έχουν τις αισθήσεις τους, ξαπλωμένοι σε αυτά τα κρεβάτια. Αυτό σημαίνει ότι οι συγκεκριμένες κλίνες πρέπει να διαθέτουν κάποιες βασικές προδιαγραφές ώστε να υποστηρίζουν τις ανάγκες των συγκεκριμένων ασθενών οι οποίες διαφέρουν από τις ανάγκες των ασθενών που βρίσκονται εκτός ΜΕΘ. Οι προδιαγραφές αυτές είναι οι εξής:

- **Τα κρεβάτια πρέπει να είναι ηλεκτρικά**

Αυτό σημαίνει ότι πρέπει να διαθέτουν χειριστήριο ή πάνελ ελέγχου ώστε να μπορεί το νοσηλευτικό και ιατρικό προσωπικό να προσαρμόζει τη θέση του αντί για χειροκίνητα.

- **Πρέπει να διαθέτει την ικανότητα να τοποθετείται σε θέση Τρεντέλεμποργκ και σε θέση καρδιακής καρέκλας.**

Αυτές οι θέσεις διευκολύνουν την αναπνοή και την κυκλοφορία του αίματος για την καλύτερη λειτουργία του σώματος. Πιο συγκεκριμένα η θέση καρδιακής καρέκλας παρέχει ανακούφιση στους πνεύμονες και την κυκλοφορία του αίματος και τους βοηθάει (τους πνεύμονες) να έρχονται από μια επίπεδη θέση σε μια κάθετη (καθιστική) χωρίς να προκαλείται τραυματισμός ή υπερβολική καταπόνηση. Από την άλλη μεριά η θέση Τρεντέλεμποργκ είναι μια μέθοδος κατά την οποία ο ασθενής τοποθετείται υπό κλίση 15 - 30 μοιρών, τοποθετώντας τα πόδια υψηλότερα από το κεφάλι. Χρησιμοποιείται κυρίως για χειρουργικές εμβολές αέρα, για την βοήθεια μείωσης κοιλιακών κηλών και την διευκόλυνση τοποθέτησης καθετήρων στην εσωτερική σφαγιτιδική φλέβα.

- **Να μπορεί να προσαρμοστεί σε αυτά στρώμα ανακούφισης πίεσης**

Λόγω της παρατεταμένης ακινησίας των ασθενών, εμφανίζονται συχνές περιπτώσεις όπου το σώμα παρουσιάζει πληγές. Αυτά τα στρώματα διαθέτουν κυψέλες αέρα οι οποίες φουσκώνουν και ξεφουσκώνουν διαδοχικά ώστε να ασκείτε μια ομοιόμορφη πίεση στο σώμα και να υπάρχει σωστή κυκλοφορία αίματος.

- **Μοχλό CPR**

Αυτή η λειτουργία επιτρέπει στο ιατρικό προσωπικό να φέρει το κρεβάτι σε επίπεδη θέση με το πάτημα ενός κουμπιού ή ενός μοχλού. Σε περίπτωση επείγων περιστατικού αυτό δημιουργεί γρήγορα μια επίπεδη και σκληρή επιφάνεια που απαιτείται για την διενέργεια ΚΑΡ.Π.Α. [17],[18]

### **1.4.3 Θερμοκοιτίδες**

Όπως αναφέρθηκε και σε προηγούμενη παράγραφο, τα πρόωρα βρέφη νοσηλεύονται στην MENN όπου οι θερμοκοιτίδες αποτελούν τον πυρήνα της. Μια νεογνική θερμοκοιτίδα είναι ένα κλειστό σύστημα όπου το πρόωρο βρέφος φυλάσσεται στο εσωτερικό της σε ένα καθαρό και ελεγχόμενο περιβάλλον, κατάλληλων περιβαλλοντικών συνθηκών θερμοκρασίας και υγρασίας, για επιτήρηση και φροντίδα. Οι ζωτικές λειτουργίες παρακολουθούνται ώστε να διασφαλιστεί η ασφάλεια των βρεφών και να αποφευχθούν τυχόν θάνατοι. Για την συνεχή παρακολούθηση των ζωτικών λειτουργιών (monitoring) των πρόωρων βρεφών στο νοσοκομείο, απαιτούνται ειδικοί αισθητήρες και ηλεκτρόδια τα οποία πρέπει να βρίσκονται σε επαφή με τον ασθενή καθώς και να μπορούν να απεικονίζονται σε μια οθόνη ή μόνιτορ. Οποιαδήποτε ανωμαλία στις παραμέτρους υποδεικνύεται από σύστημα συναγερμού.

Οι βασικές ζωτικές παράμετροι θεωρούνται να είναι η θερμοκρασία σώματος, η πίεση, ο καρδιακός και ο αναπνευστικός ρυθμός. Επίσης για μια θερμοκοιτίδα εκτός από τις φυσιολογικές παραμέτρους είναι πολύ σημαντική και η συνεχής επιτήρηση της θερμοκρασίας και της υγρασίας στο εσωτερικό της.

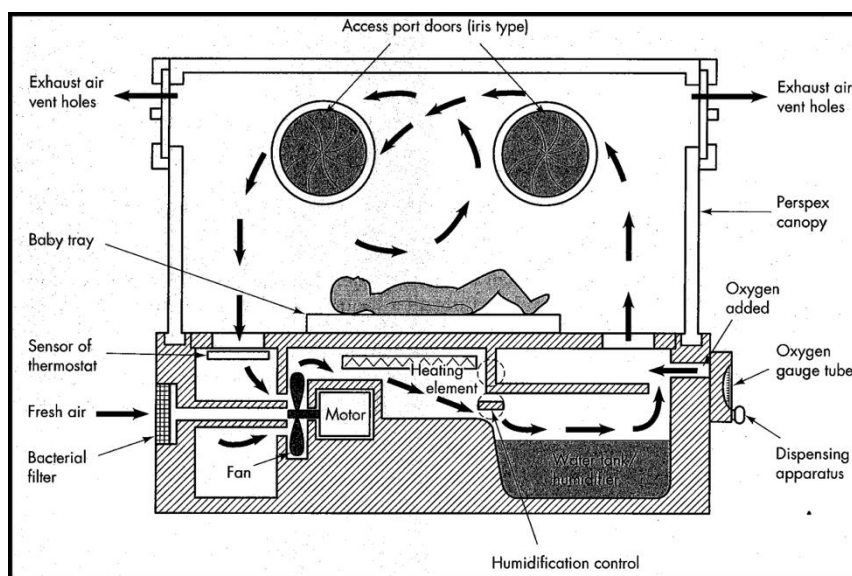
Βασικές λειτουργίες μια θερμοκοιτίδας είναι οι εξής:

- Η προστασία νεογνού από μικροβιακή λοίμωξη, άλλη μόλυνση ή θόρυβο
- Η διατήρηση υψηλής υγρασίας του αέρα για την αποτροπή μεγάλης απώλειας ύδατος από το δέρμα και την αναπνοή
- Η εξασφάλιση ασφαλούς μεταφοράς, όταν ένα άρρωστο πρόωρο νεογνό πρέπει να μετακινηθεί
- Η οξυγόνωση μέσω της χορήγησης οξυγόνου μέσω μηχανικού αερισμού, συνεχούς θετικής πίεσης αεραγωγών (CPAP) ή ρινικής κάνουλας.

Στην **εικόνα 7** παρουσιάζεται το τυπικό σχηματικό διάγραμμα μιας θερμοκοιτίδας. Προκειμένου να διατηρείται η θερμοκρασία σώματος του μωρού (36-37.2 Κελσίου) η θερμοκοιτίδα πρέπει να έχει τη δυνατότητα να δημιουργεί περιβαλλοντικό αέρα στους 34-38 Κελσίου με την υγρασία να είναι 40-80%.

Ένας ηλεκτρικός ανεμιστήρας τραβάει αέρα περιβάλλοντος μέσω ενός ειδικού φίλτρου βακτηρίων το οποίο αφαιρεί την σκόνη και τα βακτήρια. Ο φιλτραρισμένος περιβαλλοντικός αέρας κατευθύνεται σε ένα ηλεκτρικό θερμαντικό στοιχείο. Στη

συνέχεια ο φιλτραρισμένος και θερμαινόμενος αέρας περνάει πάνω από μια δεξαμενή νερού όπου αποκτά υγρασία. Έπειτα ρέει προς το κουβούκλιο της θερμοκοιτίδας. Το κουβούκλιο της θερμοκοιτίδας είναι ελαφρώς πεπιεσμένο και αυτό επιτρέπει στο διοξείδιο του άνθρακα να περάσει μέσω των οπών που έχει στο περίβλημα της στο περιβάλλον. [20],[21],[22]



Εικόνα 7. Βασική Δομή Θερμοκοιτίδας & Ροής Αέρα [19]

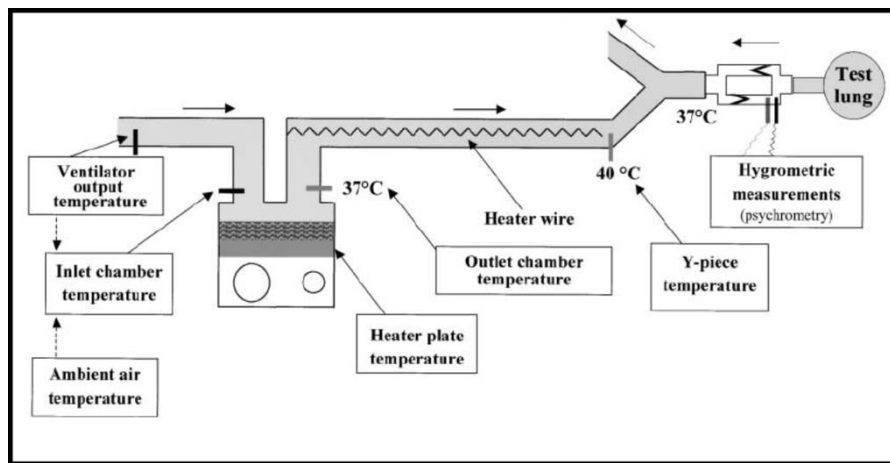
#### 1.4.4 Υγραντήρες

Τα θερμαινόμενα συστήματα ύγρανσης χρησιμοποιούνται ευρέως κατά τη διάρκεια παροχής μηχανικού αερισμού. Ο μηχανικός αερισμός παραμένει ο πρωταρχικός τρόπος αναπνευστικής υποστήριξης κατά τη διάρκεια χορήγησης γενικής ενδοτραχειακής αναισθησίας. Επιπλέον είναι χαρακτηριστικός τρόπος διαχείρισης ασθενών σε κρίσιμη κατάσταση που βρίσκονται στη ΜΕΘ. Ο μηχανικός αερισμός οδηγεί σε υπερβολική θέρμανση και απώλεια υγρασίας από το αναπνευστικό σύστημα, καθώς ο ενδοτραχειακός σωλήνας παρακάμπτει την ανώτερη αναπνευστική οδό. Αυτή η απώλεια αυξάνει το ιξώδες των εκκρίσεων της αναπνευστικής οδού και μεταβάλλει την δραστηριότητα της επιφανειοδραστικής ουσίας. Αυτά τα γεγονότα, μαζί με άλλους παράγοντες συμβάλλουν περαιτέρω στο να μη διατηρούνται φυσιολογικά επίπεδα ενυδάτωσης και θερμοκρασίας σε αναισθητοποιημένους ή κρίσιμα άρρωστους ασθενείς.

Για την επίλυση των προαναφερθέντων προβλημάτων, χρησιμοποιούνται σήμερα ιατροτεχνολογικές συσκευές που ονομάζονται υγραντήρες. Οι υγραντήρες μπορούν με ελεγχόμενο τρόπο να αντισταθμίσουν τα αποτελέσματα ψύξης και ξηρότητας των εισπνεόμενων αερίων στην αναπνευστική οδό ενός ασθενούς, προσθέτοντάς τους υγρασία και κατ' επέκταση ρυθμίζοντας την θερμοκρασία τους κατά το μηχανικό



αερισμό. Υπάρχουν αρκετοί τύποι υγραντήρων αλλά αυτοί που επικρατούν επί το πλείστον είναι οι θερμαινόμενοι υγραντήρες (ΘΥ).

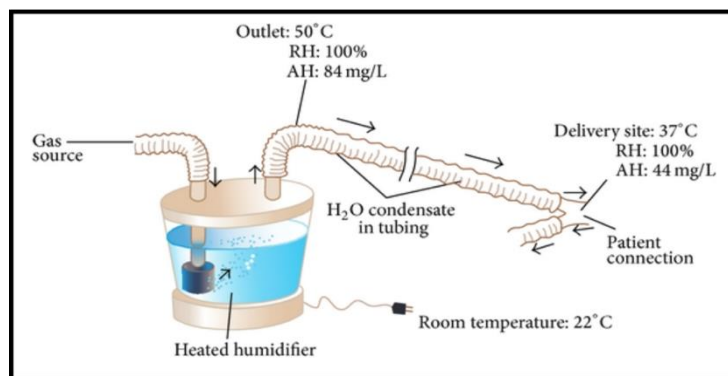


Εικόνα 8. Διάγραμμα Συστήματος Ρύθμισης Θερμοκρασίας Θερμαινόμενου Υγραντήρα [24]

Οι δυο μεγαλύτερες και πιο συχνά χρησιμοποιούμενες κατηγορίες ΘΥ είναι οι εξής:

### 1. Φυσαλίδων

Στους υγραντήρες φυσαλίδων το αέριο εξαναγκάζεται να κινηθεί χαμηλά εντός ενός σωλήνα ο οποίος βρίσκεται σε ένα δοχείο νερού. Το αέριο εξέρχεται από το ακροφύσιο του σωλήνα κάτω από την επιφάνεια του νερού σχηματίζοντας φυσαλίδες, αποκτώντας υγρασία καθώς ανεβαίνει την επιφάνεια του νερού. Η πλειοψηφία των υγραντήρων αποτελούνται από ένα ακροφύσιο το οποίο “σπάει” σε μικρές φυσαλίδες το αέριο. Το μέγεθος των φυσαλίδων επηρεάζει την ποσότητα υδρατμού στο αέριο, αφού όσο μικρότερες σε μέγεθος είναι οι φυσαλίδες, τόσο μεγαλώνει η αλληλεπίδραση μεταξύ του αερίου και του νερού. Επίσης ο ρυθμός ροής αλλά και η ποσότητα νερού που περιέχει το δοχείο επηρεάζουν το ποσοστό υδρατμών που θα περιέχει το τελικό αέριο. Σε γενικές γραμμές όσο μεγαλύτερη είναι η ποσότητα νερού στο δοχείο, τόσο μεγαλύτερη θα είναι η διεπαφής αερίου - νερού. Όσον αφορά το ρυθμό ροής, όταν διανέμονται χαμηλές ροές, υπάρχει περισσότερος χρόνος για ύγρανση του αερίου.

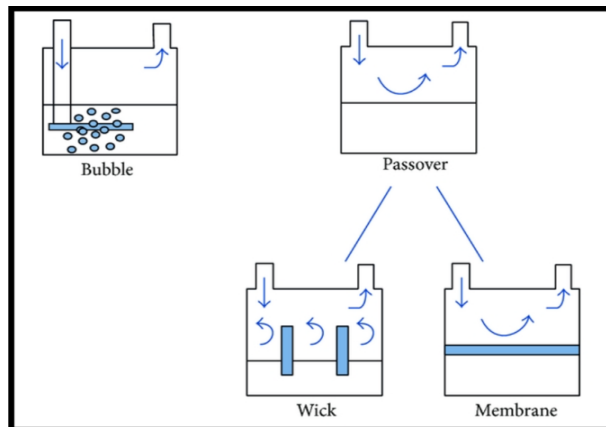


Εικόνα 9. Σύστημα Θερμαινόμενου Υγραντήρα [26]



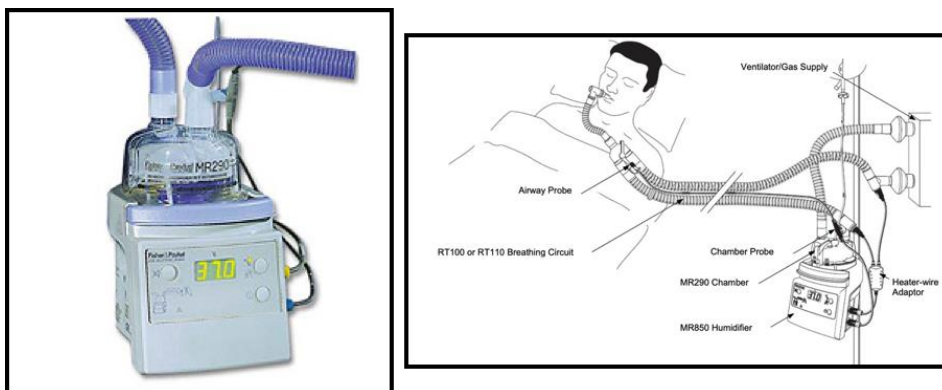
## II. Passover

Στους υγραντήρες διέλευσης, το αέριο εμπλουτίζεται με υδρατμούς και μεταφέρεται στον ασθενή περνώντας πάνω από ένα δοχείο που θερμαίνεται. Αυτοί συνήθως χρησιμοποιούνται για επεμβατικό και μη επεμβατικό μηχανικό αερισμό. Μια άλλη παραλλαγή των passover υγραντήρων είναι αυτή του “φτιλιού”. Οι υγραντήρες τύπου φτιλιού αποτελούνται από ένα δοχείο μέσα από το οποίο διέρχεται το αέριο. Στη συνέχεια το αέριο περνάει πάνω μια πορώδη επιφάνεια, μέρος της οποίας είναι βυθισμένο σε νερό. Η πορώδης επιφάνεια παρέχει μεγαλύτερη αλληλεπίδραση μεταξύ νερού και αερίου αυξάνοντας κατά αυτό τον τρόπο το ποσοστό υγρασίας συγκριτικά με απλούς υγραντήρες. Η δεξαμενή νερού τροφοδοτείται μέσω ενός κλειστού κυκλώματος είτε χειροκίνητα μέσω μιας θύρας είτε μέσω μπουκάλας νερού που βρίσκεται σε στατό που εξασφαλίζει ότι σε κάθε στιγμή η ποσότητα νερού είναι επαρκής. Καθώς το ξηρό αέριο εισέρχεται στον θάλαμο και περνάει μέσα από τα “φτιλίσια” η θερμοκρασία και η υγρασία του αερίου αυξάνεται. Ένας τρίτος τύπος Passover υγραντήρας περιλαμβάνει μια υδρόφοβη μεμβράνη. Όπως και με τη συσκευή “φτιλιού” ξηρός αέρας περνάει μέσα από μια μεμβράνη. Ωστόσο το υδρόφοβο χαρακτηριστικό της επιτρέπει μόνο τη διέλευση του υδρατμού, αποκλείοντας υγρό νερό να περάσει από μέσα της. [23],[24],[25],[26]



Εικόνα 10. Υγραντήρες Φυσαλίδων & Passover [26]

Στην **εικόνα 11** φαίνεται ένας από τους πιο κοινούς Passover υγραντήρες.



Εικόνα 11. Θερμαινόμενος Υγραντήρας Fisher & Paykel MR850 και η σύνδεσή του ασθενή [27]

## 2. Γενικές Αρχές

# Συστημάτων Επιτήρησης Ασθενούς

Με την εξέλιξη της σύγχρονης ιατρικής έγινε σαφές ότι ήταν αναγκαία η ανάπτυξη και η κατασκευή συσκευών συνεχούς παρακολούθησης ζωτικών παραμέτρων (Μόνιτορ Παρακολούθησης). Σκοπός των συστημάτων επιτήρησης ασθενούς, είναι η ποσοτική εκτίμηση σημαντικών φυσιολογικών μεταβλητών των ασθενών, στη διάρκεια μιας κρίσιμης, για τις βιολογικές λειτουργίες τους, περιόδου. Κύριο χαρακτηριστικό των σύγχρονων μόνιτορ είναι η παρακολούθηση πολλαπλών παραμέτρων την ίδια στιγμή, ενώ η λογική κατασκευής τους είναι “Modular”, δηλαδή με δυνατότητα εισαγωγής εν θερμό (κατά την λειτουργία της συσκευής), της βαθμίδας παρακολούθησης συγκεκριμένης παραμέτρου.

Τα μόνιτορ διαχωρίζονται στις 3 παρακάτω κατηγορίες:

- Τα παρακλίνια
- Τα χειρουργικά
- Τα μόνιτορ μεταφοράς

Τα μόνιτορ αποτελούνται από 4 βασικά τμήματα:

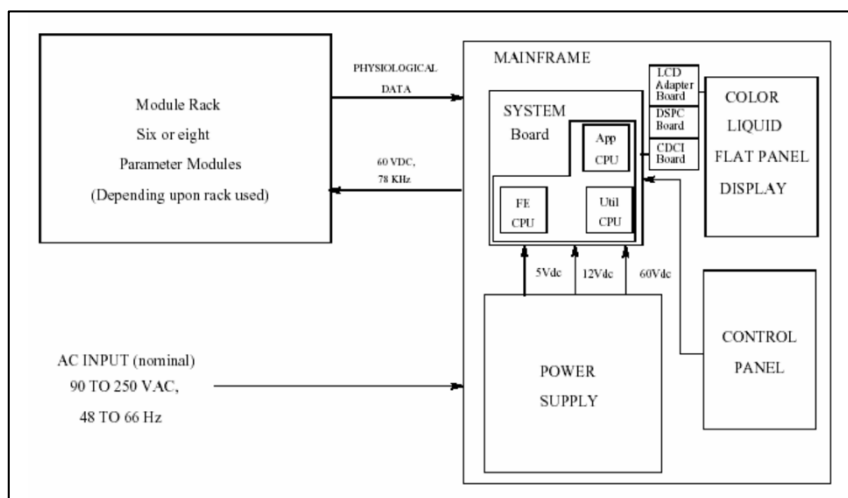
- A. Το πρώτο τμήμα είναι αυτό της απαγωγής φυσιολογικών παραμέτρων. Είναι ίσως από τα πιο βασικά τμήματα ενός μόνιτορ και είναι υποχρεωτικά ηλεκτρικά απομονωμένο από την υπόλοιπη συσκευή.
- B. Το δεύτερο είναι το τμήμα επεξεργασίας σημάτων. Αποτελείται από μια κεντρική μονάδα επεξεργασίας η οποία εφαρμόζει αλγορίθμους φιλτραρίσματος στα σήματα που ανιχνεύει η πρώτη βαθμίδα. Στα παλαιότερα μόνιτορ υπήρχε μια κεντρική μονάδα μετατροπής του Αναλογικού σήματος σε Ψηφιακό (A/D) αλλά πλέον η μετατροπή πραγματοποιείται σε κάθε τμήμα ξεχωριστά.
- C. Το επόμενο τμήμα είναι εκείνο της απεικόνισης των φυσιολογικών παραμέτρων. Η οθόνη απεικόνισης είναι τεχνολογίας TFT. Η περιοχή

απεικόνιση αποτελείται από την περιοχή γραφικής απεικόνισης των παραμέτρων (κυματομορφές φυσιολογικών παραμέτρων) και από την περιοχή απεικόνισης μεμονωμένων τιμών (παράμετροι όπου δεν απαιτείται γραφική απεικόνιση, όπως θερμοκρασία σώματος ασθενούς, συστολική/διαστολική/μέση αναίμακτη πίεση κλπ).

D. Τέλος υπάρχει το τμήμα χειρισμού του μόνιτορ. Αποτελείται από κομβία άμεσης λειτουργίας, εισαγωγής & χειρισμού των μενού του μόνιτορ. Η σύγχρονη τάση στον χειρισμό των μόνιτορ είναι με κομβίο περιστροφικής επιλογής ή με οθόνη αφής.

Όπως αναφέρθηκε προηγουμένως, τα μόνιτορ μπορούν να παρακολουθήσουν πολλαπλές παραμέτρους. Οι βασικές παράμετροι είναι οι εξής:

- Ηλεκτροκαρδιογράφημα 3-5-12πολικό
- Κορεσμός Οξυγόνου
- Αναίμακτη & Αιματηρή Πίεση
- Θερμοκρασία
- Καпноγραφία
- Αναπνοή
- Καρδιακή Παροχή [28]



Εικόνα 12. Σχηματικό Διάγραμμα ενός τυπικού μόνιτορ [28]

## 2.1 Αναίμακτη πίεση (NIBP)

Όταν η καρδιά τροφοδοτεί με αίμα τις αρτηρίες, το αίμα ρέει με μια δύναμη η οποία ασκεί πίεση στις αρτηρίες. Η πίεση του αίματος είναι το γινόμενο της ροής του αίματος με την αντίσταση των αιμοφόρων αγγείων.

Η πίεση του αίματος μετράται με περιχειρίδα πίεσης αίματος και μετράται με δυο τιμές, όπως 140/100 mm Hg (μιλιμέτρ υδραργύρου). Το πρώτο και μεγαλύτερο νούμερο στην μέτρηση ονομάζεται συστολική πίεση. Αυτή είναι η πίεση που παράγεται όταν η καρδιά συστέλλεται (τροφοδοτεί). Το δεύτερο και μικρότερο

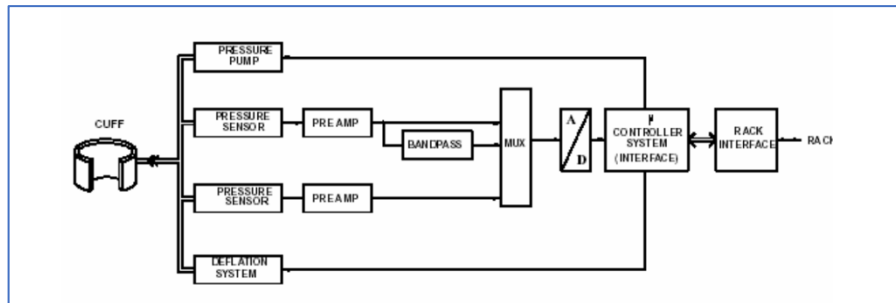
νούμερο ονομάζεται διαστολική πίεση. Αυτό το νούμερο αντικατοπτρίζει την πίεση στις αρτηρίες όσο η καρδιά γεμίζει και ξεκουράζεται ανάμεσα στους παλμούς.

Η πίεση του αίματος μπορεί να υπολογισθεί χρησιμοποιώντας μια ποικιλία μεθόδων, οι οποίες μπορούν να κατηγοριοποιηθούν σε δυο κύριες κατηγορίες:

- Την **αιματηρή**, όπου σε αυτή την περίπτωση εισέρχεται ένας καθετήρας εντός μιας αρτηρίας του ασθενή. Βασίζεται στην φυσική αρχή ότι τα υγρά είναι ασυμπίεστα και κατά συνέπεια οι μεταβολές της πίεσης μεταφέρονται και καταγράφονται οπουδήποτε σε ένα υγρό. Για το λόγο αυτό ο καθετήρας μπορεί να αποτελείται από ένα μεταλλάκτη πίεσης στην άκρη του ο οποίος περιέχει ένα διάφραγμα που πάλλεται με τις μεταβολές της πίεσης του υγρού που βρίσκεται σε επαφή με τη μεμβράνη. Την μετατόπιση της μεμβράνης την μετατρέπει το σύστημα σε ηλεκτρικό σήμα το οποίο στην συνέχεια ενισχύεται. Στην άλλη περίπτωση το κανάλι του καθετήρα είναι πληρωμένο με υγρό (συνήθως φυσιολογικός ορός) και συνδέει την πίεση του αίματος μέσω του υγρού σε έναν εξωτερικό μεταλλάκτη. Η αλλαγή της πίεσης του υγρού (πίεση του αίματος) στην αρτηρία του ασθενή μετράται αιματηρά. Αυτή η μέθοδος χαρακτηρίζεται επίσης και ως άμεση μέτρηση, καθώς η παράμετρος που μετράται είναι άμεσα συζευγμένη με τον εναλλάκτη.
- Την **αναίμακτη**, όπου σε αυτή τη περίπτωση η μέτρηση περιλαμβάνει τη χρήση μιας περιχειρίδας τοποθετημένη γύρω από κάποιο άκρο (συνήθως στο αριστερό μπράτσο) του ασθενή. Η περιχειρίδα φουσκώνει και ξεφουσκώνει σε ένα ελεγχόμενο ρυθμό (με βήμα 5 - 10 mmHg) και παρατηρούνται οι φυσικοί παράμετροι. Η ακουστική και η Ταλαντωσιομετρική τεχνική είναι οι πιο γνωστές αναίμακτες μέθοδοι. Αυτές οι μέθοδοι είναι έμμεσες αφού δεν συζευγνύονται άμεσα με την αρτηρία. Τα σύγχρονα μόνιτορ χρησιμοποιούν την ταλαντωσιομετρική μέθοδο για να μετρήσουν την αναίμακτη πίεση.

Σε αντίθεση με την ακουστική τεχνική, η οποία χρησιμοποιεί ήχους K (Korotkov) για να προσδιορίσει την πίεση του αίματος, η ταλαντωσιομετρική μέθοδος επιτηρεί τις αλλαγές στην πίεση της περιχειρίδας που προκαλούνται από την ροή του αίματος διαμέσου της αρτηρίας. Το μόνιτορ φουσκώνει την περιχειρίδα σε μια πίεση η οποία φράζει την αρτηρία. Το μόνιτορ μειώνει την πίεση της περιχειρίδας με ελεγχόμενο ρυθμό. Καθώς η πίεσή της μειώνεται, το αίμα αρχίζει να ρέει μέσω της αρτηρίας. Η αυξανόμενη ροή αίματος προκαλεί την αύξηση του πλάτους των σφυγμών της πίεσης στην περιχειρίδα. Οι σφυγμοί της πίεσης συνεχίζουν να αυξάνονται σε πλάτος με την μείωση της πίεσης της περιχειρίδας μέχρι να φτάσουν στο μέγιστο πλάτος τους, όπου σε εκείνο το σημείο αρχίζουν να μειώνονται με την μείωση της πίεσης της περιχειρίδας. Η πίεση της περιχειρίδας στην οποία το πλάτος των σφυγμών είναι το μέγιστο ονομάζεται Μέση Αρτηριακή Πίεση (ΜΑΠ).

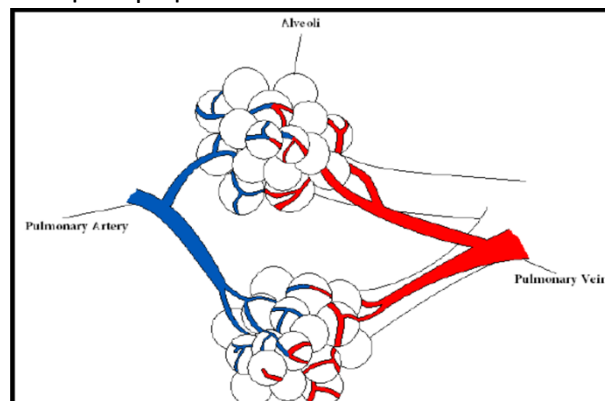
Στην **εικόνα 13** παρουσιάζεται το σχηματικό διάγραμμα του συστήματος μέτρησης αναίμακτης πίεσης. Διακρίνουμε την αντλία που «φουσκώνει» το περιβραχιόνιο, τον αισθητήρα πίεσης, τον μηχανισμό ελάττωσης πίεσης καθώς και το σύστημα μετατροπής του αναλογικού σήματος σε ψηφιακό. Η μέτρηση της πίεσης επαναλαμβάνεται σε διαστήματα που έχει επιλέξει ο χρήστης (πχ 15,30,60,120 min). [28],[29]



**Εικόνα 13.** Σχηματικό Διάγραμμα Συστήματος Μέτρησης Αναίμακτης Πίεσης [28]

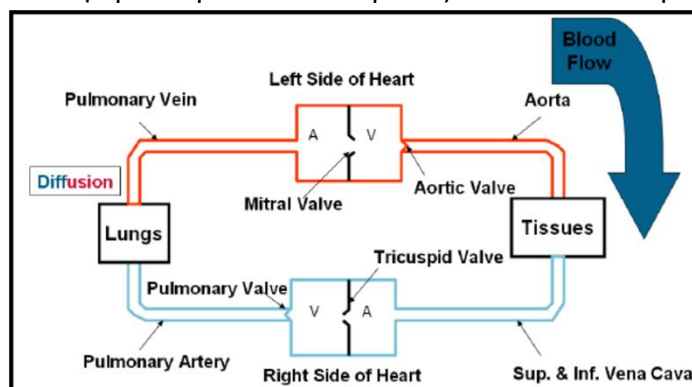
## 2.2 SpO<sub>2</sub>

Η επιτήρηση του ποσοστού κορεσμού του οξυγόνου στο αίμα (SpO<sub>2</sub>), έχει καθιερωθεί στην φροντίδα ασθενών παγκοσμίως. Σχεδόν κάθε μόνιτορ ασθενούς έχει ενσωματωμένη ή επισυναπτόμενη δυνατότητα για την παρακολούθηση αυτής της κομβικής ζωτικής παραμέτρου. Το SpO<sub>2</sub> είναι μια έμμεση και μη επεμβατική μέθοδος μέτρησης του κορεσμού του οξυγόνου στο αίμα. Η μεταφορά του οξυγόνου πραγματοποιείται μέσω του κυκλοφορικού συστήματος. Μη οξυγονωμένο αίμα εισέρχεται στην καρδιά και από εκεί διοχετεύεται στους πνεύμονες για να οξυγονωθεί. Στην διαδικασία της οξυγόνωσης, το αίμα περνάει μέσα από τις πνευμονικές κυψελίδες όπου γίνεται ανταλλαγή αερίων (διάχυση). Το διοξείδιο του άνθρακα (CO<sub>2</sub>) απελευθερώνεται και το αίμα οξυγονώνεται και ύστερα τροφοδοτείται πίσω στην αορτή.



**Εικόνα 14.** Διαδικασία Οξυγόνωσης του Αίματος [31]

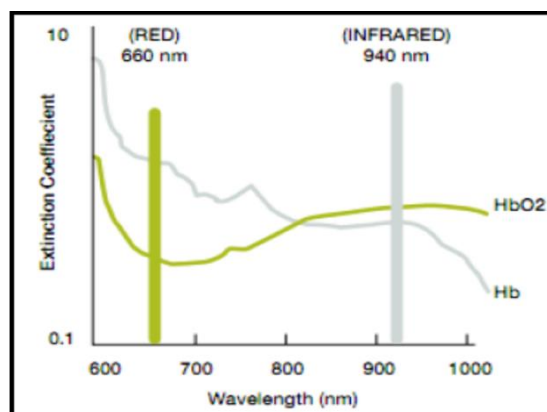
Τα ερυθρά αιμοσφαίρια περιέχουν μια πρωτεΐνη που ονομάζεται αιμοσφαιρίνη. Όταν το οξυγόνο αντιδρά με αυτή τη πρωτεΐνη, προσκολλάται σε αυτήν και παράγει οξυγονωμένη αιμοσφαιρίνη (HbO<sub>2</sub>). Τα ερυθρά αιμοσφαίρια με οξυγονωμένη αιμοσφαιρίνη κυκλοφορούν μέσω του αίματος σε όλο το σώμα, μεταφέροντας



Εικόνα 15. Διάγραμμα Κυκλοφορίας του Αίματος [31]

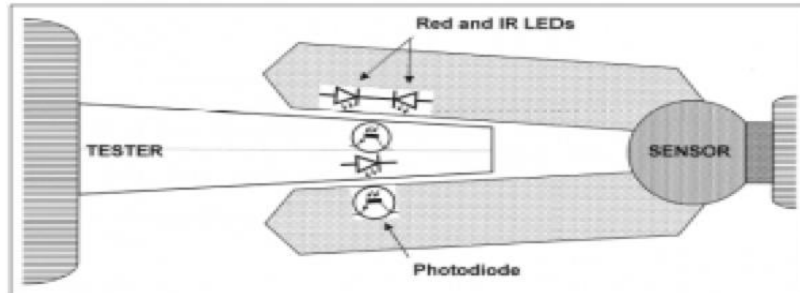
οξυγόνο στους ιστούς. Όταν το αίμα έρθει σε επαφή με ένα κύτταρο, η αιμοσφαιρίνη των ερυθρών αιμοσφαιρίων απελευθερώνει οξυγόνο και γίνεται μη οξυγονωμένη αιμοσφαιρίνη. Από αυτό το σημείο, το αίμα που δεν περιέχει πλέον οξυγόνο επιστρέφει στον δεξιό καρδιακό κόλπο της καρδιάς έτσι ώστε να επαναληφθεί η διαδικασία.

Το SpO<sub>2</sub> μετράται περιμετρικά, συνήθως σε δάχτυλο και δίνει σημαντικές πληροφορίες για το καρδιαγγειακό και αναπνευστικό σύστημα. Η συσκευή αποτελείται από μια πηγή φωτός η οποία εκπέμπει στο κόκκινο (660 nm) και μια στο υπέρυθρο (940 nm), από ανιχνευτές φωτός και το μέσο μέσα από το οποίο μεταδίδεται το φως (δάχτυλο ή λοβός). Η οξυγονωμένη (O<sub>2</sub>Hb) και η μη οξυγονωμένη (Hb) αιμοσφαιρίνη απορροφούν διαφορετικά το κόκκινο και το υπέρυθρο φως. Το ποσοστό κορεσμού της αιμοσφαιρίνης στο αρτηριακό αίμα μπορεί να υπολογιστεί μετρώντας τις αλλαγές στην απορρόφηση του φωτός που προκαλείται από τις διακυμάνσεις στη ροή του αίματος. Η μη οξυγονωμένη αιμοσφαιρίνη (Hb) παρουσιάζει υψηλότερη απορρόφηση στα 660 nm ενώ η οξυγονωμένη (HbO<sub>2</sub>) στα 940 nm.



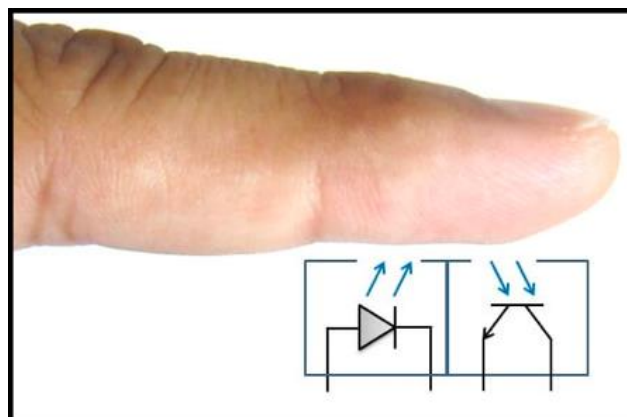
Εικόνα 16. Γράφημα Απορρόφησης Φωτός από την Αιμοσφαιρίνη [31]

Υπάρχουν δυο είδη τεχνολογίας για την ανίχνευση του SpO<sub>2</sub>: Αυτό της ανάκλασης και αυτό της διάδοσης. Από τις δυο αυτές μεθόδους, αυτή της διάδοσης χρησιμοποιείται επί το πλείστον. Όπως φαίνεται και στην εικόνα, στην τεχνολογία διάδοσης διαδίδεται κόκκινο και υπέρυθρο φως μέσω του δαχτύλου σε ένα φωτοανιχνευτή.



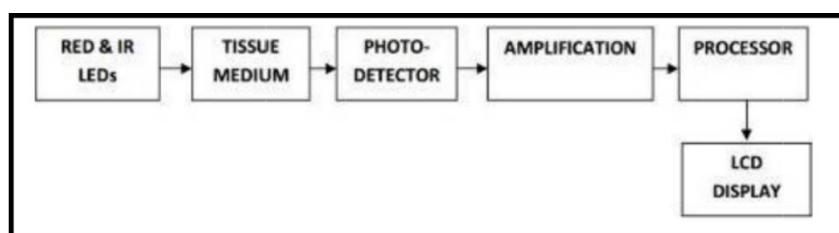
Εικόνα 17. Οξύμετρο Τεχνολογίας Διάδοσης [30]

Η άλλη μέθοδος είναι αυτή της ανάκλασης. Όπως φαίνεται και στην εικόνα, σε αυτή τη μέθοδο ο πομπός και ο δέκτης βρίσκονται στο ίδιο επίπεδο. Στην οξυμετρία ανάκλασης, οι αισθητήρες μπορούν να τοποθετηθούν και σε άλλες ανατομικές περιοχές όπως το πέλμα του ποδιού (σε νεογνά).



Εικόνα 18. Οξύμετρο με Τεχνολογία Ανάκλασης [30]

Ο υπολογισμός του επιπέδου κορεσμού στο αίμα βασίζεται στον νόμο των Beer - Lambert για την φασματική ανάλυση. Ο νόμος δηλώνει ότι η συγκέντρωση της απορροφώμενης ενέργειας στο διάλυμα μπορεί να προσδιοριστεί σαν μαθηματική συνάρτηση του φωτός που διαχέεται μέσα από το διάλυμα, δεδομένου ότι είναι γνωστά η ένταση του προσπίπτοντος φωτός, το μήκος της κυψελίδας και η σταθερά που χαρακτηρίζει το δείγμα σε ένα συγκεκριμένο μήκος κύματος. Το μπλοκ διάγραμμα ενός απλού Φασματοφωτομέτρου φαίνεται στην **εικόνα 19**:



Εικόνα 19. Μπλοκ Διάγραμμα Φασματοφωτομέτρου [28]

Η εξίσωση που περιγράφει το νόμο Beer-Lambert είναι η εξής [32]:

$$A_{\lambda 1} = \varepsilon_{o1}l_oC_o + \varepsilon_{r1}l_rC_r + \varepsilon_{x1}l_xC_x + A_{y1} \quad (1)$$

$$A_{\lambda 2} = \varepsilon_{o2}l_oC_o + \varepsilon_{r2}l_rC_r + \varepsilon_{x2}l_xC_x + A_{y2} \quad (2)$$

Όπου,

$A_{\lambda}$ : Απορροφητικότητα σε ένα διακριτό μήκος κύματος

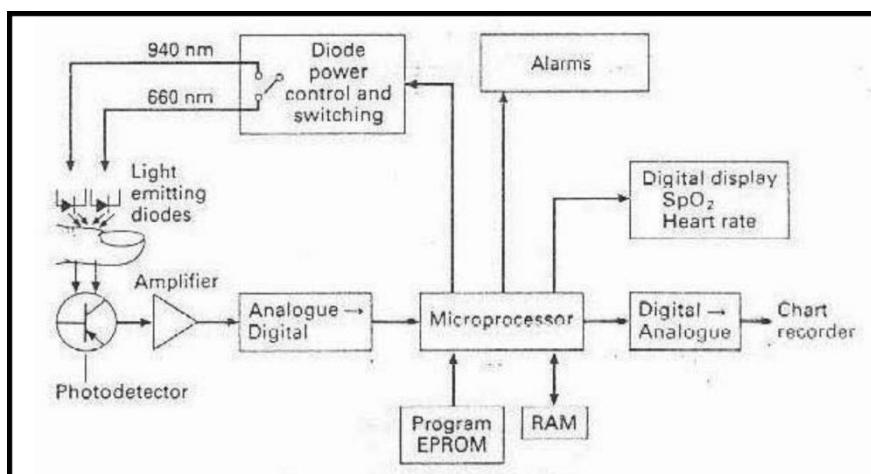
$\varepsilon_o$  = Συντελεστής Απόσβεσης

$l$  : Μήκος Διαδρομής

$C$  : Συγκέντρωση Αιμοσφαιρίνης

$\lambda$  : Μήκος Κύματος

Τέλος το βασικό κύκλωμα ενός σύγχρονου οξυμέτρου είναι το εξής:



Εικόνα 20. Τυπικό μπλοκ Διάγραμμα Αισθητήρα Οξυμετρίας [28]

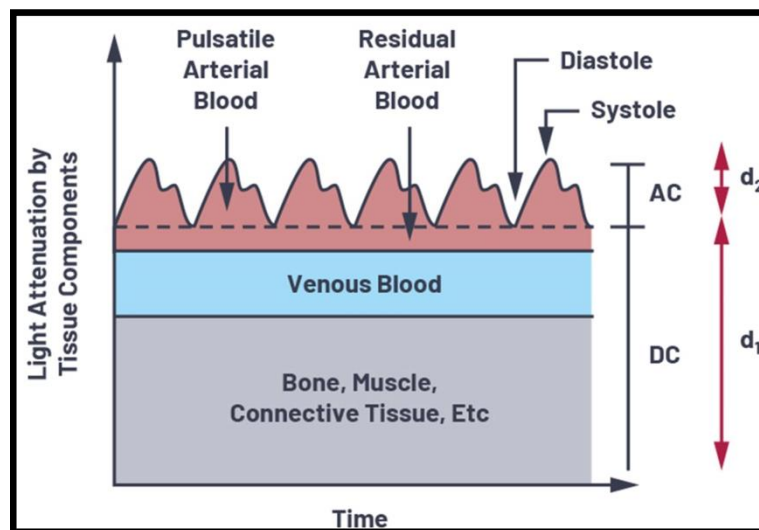
Το πιο απλό παλμικό οξύμετρο αποτελείται από δυο LED (ένα κόκκινο στα 660 nm και ένα υπέρυθρο στα 940 nm) και μια φωτοδίοδο σε διαμόρφωση διάδοσης ή ανάκλασης. Το εισερχόμενο σήμα από τον φωτοανιχνευτή είναι πολύ χαμηλό και ενισχύεται. Αρχικά θα εκπέμψει το κόκκινο LED και το σήμα θα μετρηθεί από τη φωτοδίοδο. Το ίδιο θα επαναληφθεί και για το υπέρυθρο LED. Στο τέλος απενεργοποιούνται και τα δύο LED έτσι ώστε να παρθεί μια βάση αναφοράς για οποιαδήποτε εξωτερική περιβαλλοντική πηγή φωτός. Όταν τα σήματα περάσουν από ηλεκτρονικά φίλτρα, η υψηλή συχνότητα (1KHz) διακόπτεται έτσι ώστε τα σήματα να εμφανίζονται ότι προέρχονται από συνεχής πηγή του κάθε μήκους κύματος. Το επίπεδο του περιβάλλοντος χώρου που ανιχνεύεται κατά την διάρκεια της σκοτεινής περιόδου, αφαιρείται από τα ρεύματα (DC) για αποφυγή λαθών. Εν συνεχεία ένα από τα δύο σήματα φωτοπληθυσμογραφίας μεταβάλλεται σε ενίσχυση έως ότου οι dc συνιστώσες και των δύο γίνουν ίσες.



Ο Κορεσμός Αιμοσφαιρίνης δίνεται από την παρακάτω εξίσωση, όπου οι τιμές  $HbO_2$  &  $Hb$ , αντιστοιχούν στις τιμές απορρόφησης για τα 2 διαφορετικά μήκη κύματος [28].

$$\text{Λειτουργικός } SaO_2 = \frac{HbO_2}{(HbO_2 + Hb)} \times 100$$

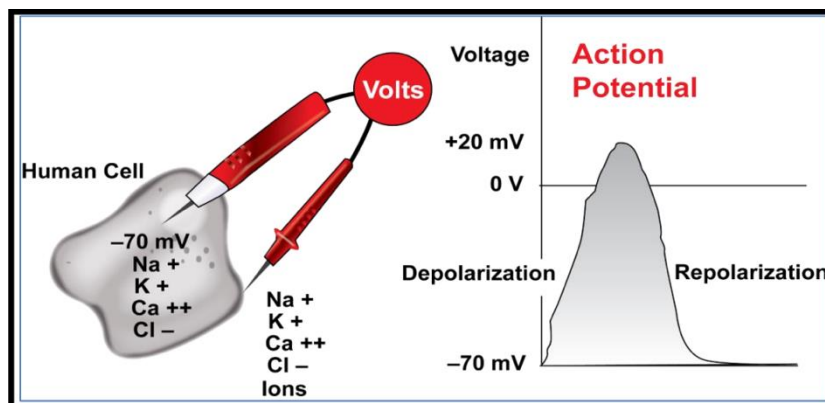
Με αυτό το τρόπο παράγεται το σήμα φωτοπληθυσμογραφίας και για τα δυο μήκη κύματος. Η DC συνιστώσα οφείλεται στη συνεχή ανακλαστική ύλη όπως το δέρμα, μύες, κόκαλα και φλεβικό αίμα. Η AC συνιστώσα αποτελείται κυρίως από ανακλώμενο φως που προκαλείται λόγω του παλμού του αρτηριακού αίματος. Η AC συνιστώσα εξαρτάται από τον καρδιακό ρυθμό και το πάχος της αρτηρίας, με το φως να ανακλάται ή να διαδίδεται περισσότερο συστολική. Κατά τη διάρκεια της συστολικής φάσης, το αίμα διοχετεύεται από την καρδιά, το οποίο αυξάνει την πίεση του κοιλιακού αίματος. Η αύξηση στην πίεση του αίματος διαστέλλει τις αρτηρίες και οδηγεί σε αύξηση του όγκου του κοιλιακού αίματος. Η αύξηση του αίματος προκαλεί αύξηση στην απορρόφηση του φωτός. Αντίστοιχα η πίεση του αίματος ελαττώνεται κατά τη διάρκεια της διαστολικής φάσης και συνεπώς ελαττώνεται και η απορρόφηση του φωτός. [30], [31],[32],[33]



Εικόνα 21. Απορρόφηση Φωτός μέσω των Ιστών [33]

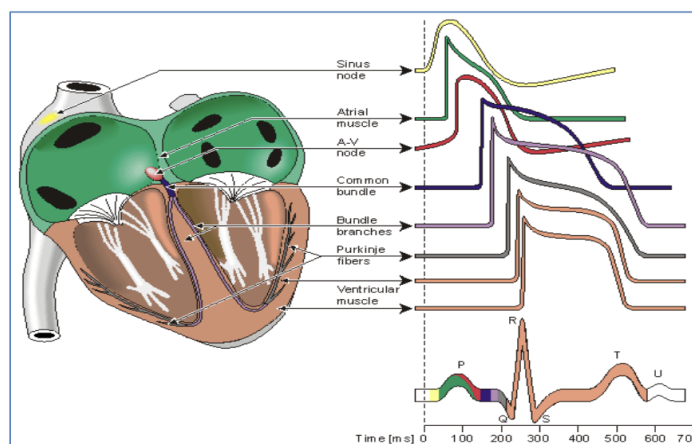
## 2.3 ECG

Τα κύτταρα στους ανθρώπους συμπεριφέρονται σαν μπαταρίες. Αυτά τα κύτταρα έχουν διαφορετικές συγκεντρώσεις ιόντων εντός και εκτός των μεμβρανών τους τα οποία δημιουργούν μικρά ηλεκτρικά δυναμικά που ονομάζονται βιοδυναμικά. Όταν υπάρχει αναταραχή σε ένα βιοδυναμικό, προκαλείται ένα δυναμικό ενέργειας που οφείλεται στην αποπόλωση και την επαναπόλωση του κυττάρου.



Εικόνα 22. Κάθε κύτταρο λειτουργεί σαν μια μικρή μπαταρία η οποία αποπολώνεται και επαναπολώνεται όταν υπάρχει κάποια αναταραχή [34]

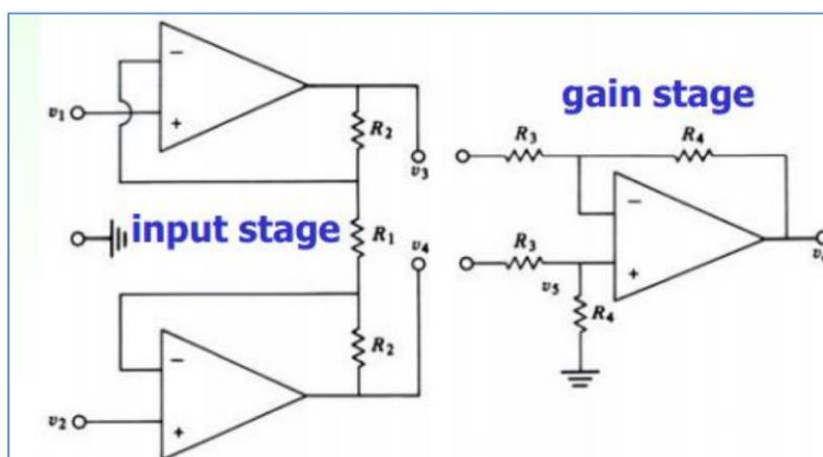
Ουσιαστικά, τα δυναμικά ενέργειας από διαφορετικούς κόμβους της καρδιάς είναι αυτά που συνθέτουν τα σήματα ηλεκτροκαρδιογραφήματος (ECG). Τα σήματα ECG αποτελούνται από την υπέρθεση διαφορετικών δυναμικών ενέργειας από τον χτύπο της καρδιάς. Τα μηχανήματα που καταγράφουν το ECG χρησιμοποιούν ηλεκτρόδια για να μετατρέψουν τα ιοντικά σήματα του σώματος σε ηλεκτρικά σήματα έτσι ώστε να απεικονιστούν και να χρησιμοποιηθούν σε ανάλυση δεδομένων. Οι καταγραφές ονομάζονται απαγωγές και λαμβάνονται ως συνάρτηση πολλαπλών ηλ. δυναμικών στην επιφάνεια του δέρματος. Οι βασικές απαγωγές είναι οι Διπολικές (I, II, III), οι Μονοπολικές (aVR, aVL, aVF) & προκάρδιες (V1 –V6). Ωστόσο εξαιτίας του μικρού μεγέθους των καταγραφών και του εξωτερικού θορύβου, το ECG χρειάζεται ενίσχυση και φιλτράρισμα για να παραχθεί υψηλής ποιότητας σήμα.



Εικόνα 23. Η υπέρθεση όλων των ενεργών δυναμικών παράγουν το σήμα ECG [34]

Η ενίσχυση επιτυγχάνεται κυρίως μέσω ενός διαφορικού ενισχυτή ενώ το φιλτράρισμα πραγματοποιείται μέσω κοινού και διαφορικού κόμβου. Τα σήματα ECG ποικίλουν από  $\mu\text{V}$  μέχρι  $\text{mV}$ . Εξαιτίας αυτού του μικρού εύρους, τα σήματα που μετρώνται πρέπει να ενισχυθούν έτσι ώστε να ερμηνευθούν καλύτερα. Οι τυπικοί ενισχυτές βιοδυναμικών έχουν υψηλή αντίσταση εισόδου και σχεδιάζονται με γνώμονα την ασφάλεια. Αυτό συμβαίνει γιατί το σήμα που ενισχύεται προέρχεται από ένα ζωντανό οργανισμό και πρέπει να παρθούν ειδικές προφυλάξεις έτσι ώστε να αποφευχθούν ατυχήματα. Συνήθως ένα κέρδος άνω του 1000 εφαρμόζεται σε κυκλώματα ενίσχυσης βιοσημάτων. Τέλος οι περισσότεροι ενισχυτές βιοσημάτων είναι διαφορικοί και χρησιμοποιούνται για να διασφαλιστεί ότι ο θόρυβος από τις εισόδους δεν θα ενισχυθεί, δίνοντας έτσι μεγαλύτερη ακρίβεια στο σήμα.

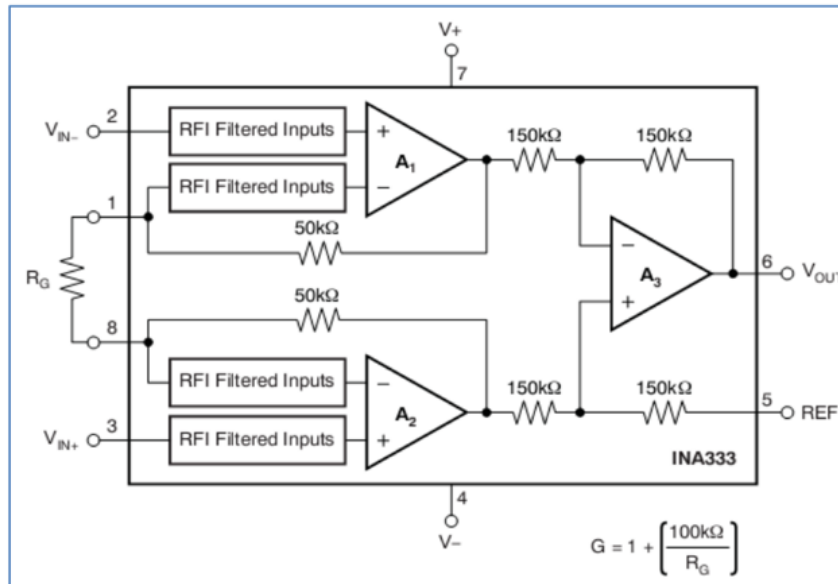
Στην **εικόνα 24** παρουσιάζεται ένα τυπικό κύκλωμα ενισχυτή οργανολογίας:



Εικόνα 24. Στάδια εισόδου και κέρδους ενός ενισχυτή οργανολογίας [34]

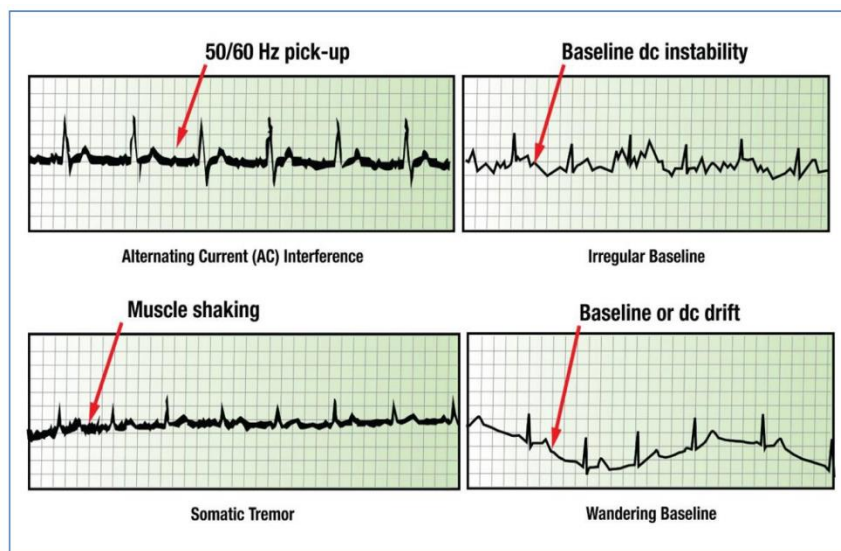
Στο στάδιο της εισόδου ιδανικά δεν παρέχεται καθόλου κέρδος κοινού σήματος, συνεπώς απαλείφεται ο θόρυβος κοινού σήματος. Στη συνέχεια οι έξοδοι του σταδίου εισόδου τροφοδοτούνται σαν εισοδοί στο στάδιο κέρδους. Το στάδιο κέρδους έχει χαμηλή σύνθετη αντίσταση και παρέχει διαφορικό κέρδος.

Ένα από τα πιο διαδεδομένα ολοκληρωμένα τσιπ και ευρέως διαθέσιμα είναι αυτό της Texas Instruments η οποία παρέχει το τσιπ “INA333”. Το τσιπ προσφέρει ευελιξία στη ρύθμιση του κέρδους ενώ μειώνει κατά πολύ το σχεδιαστικό σφάλμα του κυκλώματος.



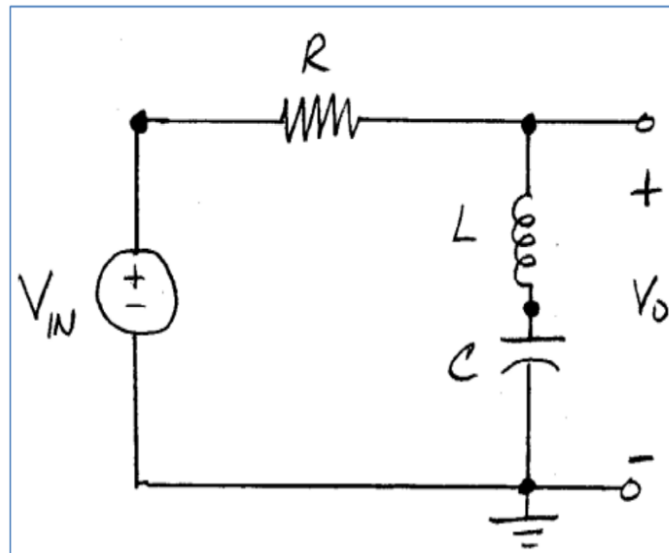
Εικόνα 25. Σχηματικό Διάγραμμα του INA333

Δυστυχώς ένα από τα προβλήματα που μπορεί να επηρεάσει το σήμα κατά την απαγωγή του, είναι αυτό του θορύβου. Αυτό έχει ως αποτέλεσμα να δημιουργούνται ψευδενδείξεις πάνω στο ωφέλιμο σήμα. Για να μπορέσει να ξεπεραστεί αυτό το πρόβλημα εφαρμόζονται ειδικά φίλτρα στο σήμα τα οποία αποκόπτουν κάθε είδους εσωτερικό ή εξωτερικό θόρυβο.



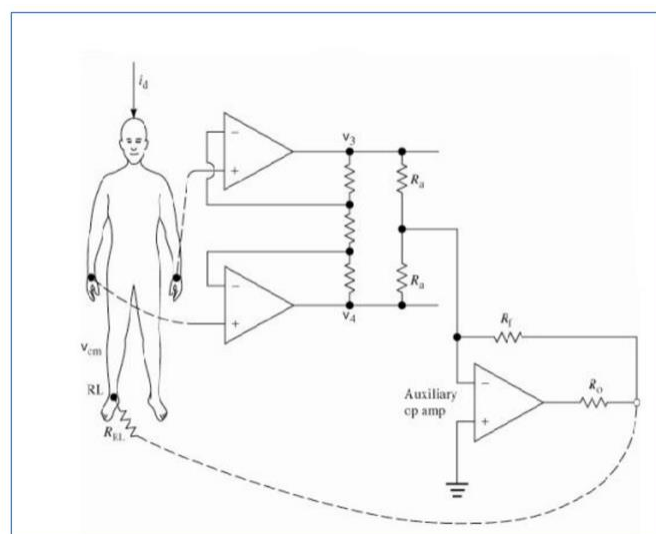
Εικόνα 26. Ψευδενδείξεις ECG [34]

Οι ψευδενδείξεις μπορούν να φιλτραριστούν είτε ψηφιακά είτε αναλογικά. Για παράδειγμα το ζωνοπερατό φίλτρο δευτέρου βαθμού που φαίνεται στην εικόνα μπορεί να τροποποιηθεί, για να αποκόψει τον θόρυβο των 60 Hz που προκαλείται από το δίκτυο ρεύματος, με βάση της τιμές αντίστασης, πυκνωτή ή πηνίου που θα επιλεγούν.



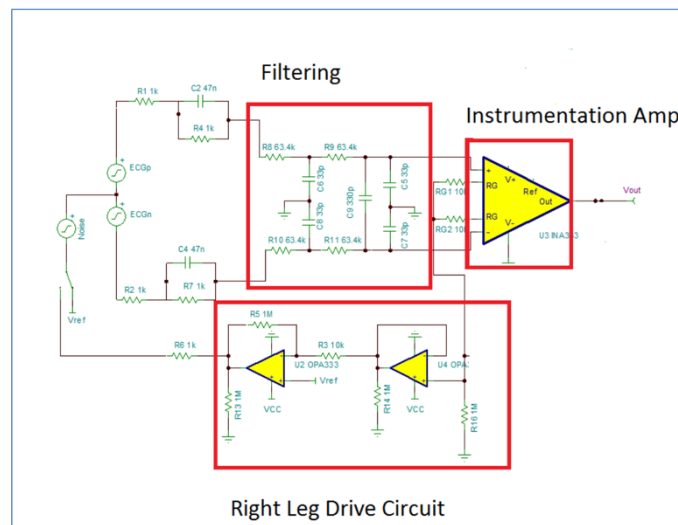
Εικόνα 27. Δευτέρου βαθμού ζωνοπερατό φίλτρο [34]

Τέλος μια μέθοδος για την μείωση του θορύβου είναι αυτή του οδηγού δεξιού ποδιού. Το κύκλωμα Right Leg Drive χρησιμοποιείται για την ελαχιστοποίηση της επίδρασης κοινού σήματος. Αναστρέφει και ενισχύει το μέσο κοινό σήμα και το ανατροφοδοτεί πίσω στο δεξί πόδι του ασθενή. Με αυτόν τον τρόπο ακυρώνεται ο θόρυβος των 60 Hz της εναλλασσόμενης τάσης και δημιουργεί ένα πιο καθαρό σήμα.



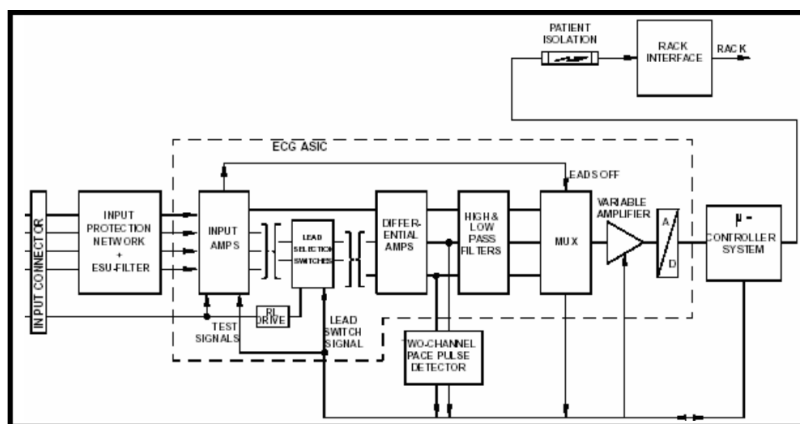
Εικόνα 28. Σύνδεση οδηγού δεξιού ποδιού [34]

Το τελικό κύκλωμα μέτρησης του ECG φαίνεται στην **εικόνα 29** .



Εικόνα 29. Κύκλωμα ECG [34]

Η βαθμίδα απαγωγής ECG ενός μόνιτορ απεικονίζεται στην **εικόνα 30**.

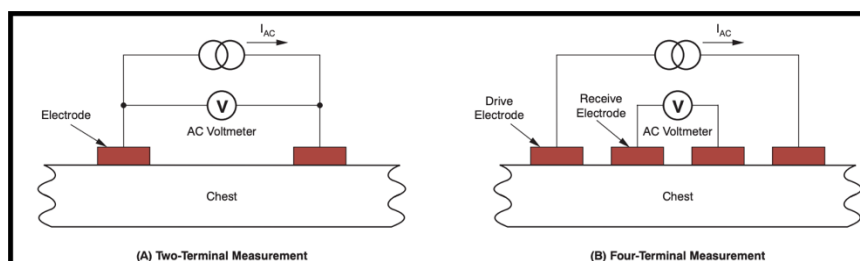


Εικόνα 30. Βαθμίδα απαγωγής ECG [34]

Τα καλώδια από τα ηλεκτρόδια του ασθενούς εισέρχονται στην βαθμίδα απαγωγής μέσω του κονέκτορα. Στη συνέχεια υπάρχει το κύκλωμα προστασίας από υπερτάσεις. Ακολουθεί το κομμάτι της ενίσχυσης και η βαθμίδα φιλτραρίσματος που αναλύθηκε παραπάνω. Οι επόμενες βαθμίδες χρησιμοποιούνται για την μετατροπή του αναλογικού σήματος σε ψηφιακό με τη χρήση πολυπλεκτών και αναλογικο-ψηφιακών μετατροπών. [34]

## 2.4 Ρυθμός αναπνοής

Η θωρακική εμπέδηση είναι μια κοινώς χρησιμοποιούμενη τεχνική για την επιτήρηση του ρυθμού αναπνοής ενός ατόμου. Εφαρμόζεται είτε με τη χρήση δυο ηλεκτροδίων είτε με τεσσάρων. Η βασική αρχή αυτής της τεχνικής είναι η μέτρηση της μεταβολής στην ηλεκτρική εμπέδηση στο θώρακα του ασθενή που προκαλείται κατά την αναπνοή.



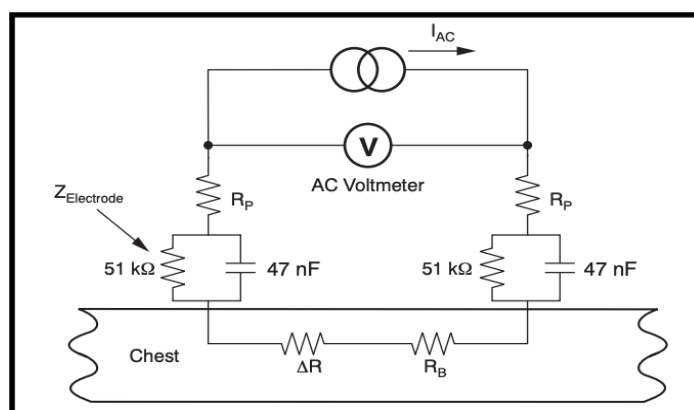
Εικόνα 31. Τοποθέτηση Ηλεκτροδίων για Πνευμογραφία Εμπέδησης [35]

Και στις δυο τεχνικές ένα υψίσυχο εναλλασσόμενο ρεύμα διαρρέει το δέρμα μέσω των ηλεκτροδίων. Το εναλλασσόμενο ρεύμα δημιουργεί μια διαφορά δυναμικού ανάμεσα σε δυο οποιαδήποτε σημεία μεταξύ των ηλεκτροδίων. Αυτή η διαφορά δυναμικού σχετίζεται με την αντίσταση του δέρματος μεταξύ των ηλεκτροδίων. Η ισοδύναμη αντίσταση ορίζεται ως η αναλογία της διαφοράς τάσης μεταξύ των δυο ηλεκτροδίων και το ρεύμα που ρέει στο δέρμα. Η συνδεσμολογία των τεσσάρων ηλεκτροδίων αποδίδει πιο ακριβή αποτελέσματα από αυτήν με δυο ηλεκτρόδια. Αυτό οφείλεται στο ότι τα σημεία που διαρρέονται από ρεύμα είναι φυσικά διαχωρισμένα από τα σημεία που μετράται η τάση. Παρόλα αυτά λόγω της ανάγκης να χρησιμοποιηθούν περισσότερα ηλεκτρόδια η συνδεσμολογία των δυο ηλεκτροδίων επικρατεί, αν λάβει κανείς υπόψιν ότι για την μέτρηση μπορούν χρησιμοποιηθούν δυο από τα ηλεκτρόδια που χρησιμοποιούνται για το ΗΚΓ.

Κατά τη διάρκεια μέτρησης της αναπνοής, ο θώρακας παρουσιάζει μια ηλεκτρική εμπέδηση στο ηλεκτρόδιο που αποτελείται από δυο συνιστώσες : μια με σχετικά σταθερή τιμή και μια με μεταβαλλόμενη. Η σταθερή τιμή της θωρακικής εμπέδησης, γνωστή και ως σύνθετη αντίσταση αναφοράς ( $R_b$ ), είναι  $500 \Omega$  ενώ η μεταβαλλόμενη τιμή είναι γνωστή και ως αναπνευστική αντίσταση ( $\Delta R$ ). Οι αλλαγές στην ηλεκτρική αντίσταση των πνευμόνων είναι κυρίως αποτέλεσμα δυο φαινομένων:

- Κατά τη διάρκεια της εισπνοής, υπάρχει αύξηση στον όγκο αερίων στον θώρακα σε σχέση με τον όγκο ρευστών. Αυτή η αύξηση προκαλεί πτώση στην αγωγιμότητα.
- Κατά τη διάρκεια της εισπνοής, η απόσταση της αγωγίμης περιοχής αυξάνεται εξαιτίας της διόγκωσης.

Λαμβάνοντας υπόψιν τα παραπάνω, και τα δυο αυτά φαινόμενα προκαλούν αύξηση στην ηλεκτρική εμπέδηση. Θα μπορούσε κανείς να συμπεράνει ότι η σχέση που συνδέει τον όγκο του εισπνεόμενου αέρα με την αλλαγή στη σύνθετη αντίσταση είναι σχεδόν γραμμική. Η μεταβαλλόμενη αντίσταση δημιουργεί μια μεταβαλλόμενη τάση ( $\Delta V$ ) όταν εφαρμόζεται ρεύμα. Αυτή η μεταβαλλόμενη τάση είναι η παράμετρος ενδιαφέροντος αφού μπορεί να χρησιμοποιηθεί ώστε να καθοριστεί ο ρυθμός αναπνοής ενός ανθρώπου. Συνήθως η  $\Delta R$  βρίσκεται στο εύρος  $0.1 \Omega$  με  $1 \Omega$ . Η  $\Delta V$  εξαρτάται από την ένταση του ρεύματος που εφαρμόζεται. Η εικόνα ( ) δείχνει το ηλεκτρικό μοντέλο ενός αναπνευστικού κύκλωματος. Το κύκλωμα περιλαμβάνει τις αντιστάσεις προστασίας απινιδωτή ( $R_p$ ) σε κάθε καλώδιο του ECG ( $1 \text{ k}\Omega$  / καλώδιο) και την σύνθετη αντίσταση του ηλεκτροδίου ( $Z_{\text{electrode}}$ ). [35]



Εικόνα 32. Ηλεκτρικό Μοντέλο Κυκλώματος Αναπνοής [35]

## 2.5 Καπνογραφία

Η παρακολούθηση της καπνογραφίας (End-Tidal carbon dioxide,  $ET_{CO_2}$ ) παρέχει πολύτιμες πληροφορίες σχετικά με την παραγωγή διοξειδίου του άνθρακα ( $CO_2$ ) και τον μηχανικό αερισμό. Επίσης γνωστή και ως καπνομετρία ή καπνογραφία, αυτή η τεχνική πραγματοποιείται συνήθως με ενδοτραχειακό αλλά και ρινικό σωλήνα. Η τελοεκπνευστικής καπνογραφίας παρέχει ανάλυση της κάθε αναπνοής και μια συνεχόμενη καταγραφή της αναπνευστικής κατάστασης. Παρέχοντας στιγμιαία ανάδραση για την αποτελεσματικότητα του μηχανικού αερισμού, η παρακολούθηση της  $ET_{CO_2}$  προσφέρει πρόωρη προειδοποίηση για βλάβη στο αναπνευστικό.

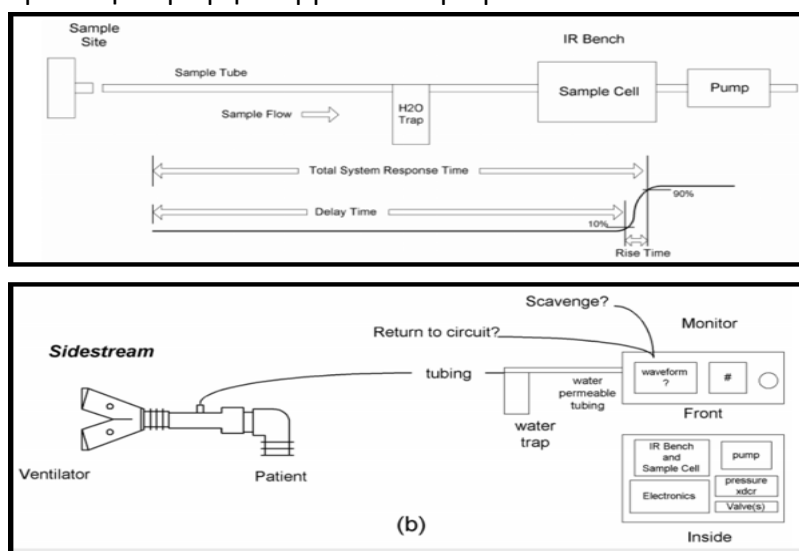
Ο βασικός τρόπος λειτουργίας των μόνιτορ καπνογραφίας είναι ότι εκπέμπουν υπέρυθρο φως πάνω στο αναπνευστικό κύκλωμα του ασθενούς. Το υπέρυθρο φως έχει την τάση να απορροφάται από το  $CO_2$ , έτσι η ανίχνευση μικρής ποσότητας φωτός από τον αισθητήρα υποδηλώνει την ύπαρξη μεγάλης ποσότητας  $CO_2$ . Με αυτό τον τρόπο, ο αισθητήρας μπορεί να καθορίσει το διοξείδιο που εκπνέεται από τον ασθενή. Στη συνέχεια τα δεδομένα που έχει καταγράψει ο αισθητήρας στέλνονται σε ένα μόνιτορ ώστε να γίνει απεικόνισή τους. Η πρώτη μορφή απεικόνισης παρουσιάζει την αυξομείωση των επίπεδων  $CO_2$  κατά τη διάρκεια της εκπνοής του ασθενούς με γραφική απεικόνιση. Έπειτα απεικονίζεται το



τελοεκπνευστικό CO<sub>2</sub>, το οποίο είναι συνήθως 30-45 mm Hg. Αυτό ονομάζεται καπνομετρία.

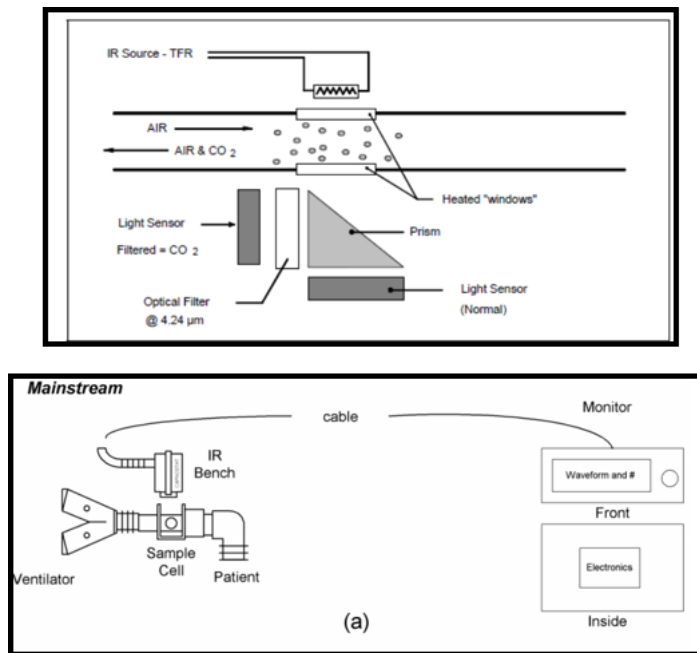
Υπάρχουν 3 κύριες τεχνολογίες για την καταγραφή της ETco<sub>2</sub>:

- Η πρώτη είναι η **Sidestream**. Μια από τις πιο διαδεδομένες μεθόδους για την μέτρηση της καπνογραφίας είναι η τεχνική Sidestream. Για την μέθοδο αυτή το αναπνευστικό κύκλωμα του ασθενή είναι συνδεδεμένο μέσω λεπτού σωλήνα, το οποίο καταλήγει σε ένα εξωτερικό μόνιτορ. Κατά την εκπνοή του ασθενούς, ένα μικρό δείγμα αερίου αναρροφάται από το Y του κυκλώματος του ασθενή και αναλύεται από το μόνιτορ. Λόγω του μεγάλου όγκου που αναρροφάται κατά τη δειγματοληψία (περίπου 150-200mL/λεπτό), αυτή η τεχνική δεν μπορεί να χρησιμοποιηθεί σε νεογνά ή βρέφη. Ένα από τα μειονεκτήματα της τεχνικής Sidestream είναι ότι έχουν μια μικρή καθυστέρηση στην ανάκτηση δεδομένων η οποία οφείλεται στον χρόνο που χρειάζεται το δείγμα για να μεταφερθεί από το αναπνευστικό κύκλωμα στο μόνιτορ. Η Sidestream τεχνική θεωρείται φτηνή λύση και μπορεί να χρησιμοποιηθεί με μη επεμβατικό αερισμό



Εικόνα 33. Μπλοκ Διάγραμμα & Λειτουργική Αναπαράσταση Sidestream Τεχνολογίας [36]

- Η δεύτερη τεχνολογία είναι η **Mainstream**. Σε αντίθεση με την Sidestream τεχνική, η Mainstream χαρακτηρίζεται από την αμεσότητα των αποτελεσμάτων. Αυτό οφείλεται στο ότι τοποθετείται απευθείας, παράλληλα με το κύκλωμα του ασθενή το σύστημα δειγματοληψίας για τη μέτρηση του διοξειδίου. Το πλεονέκτημα αυτής της μεθόδου είναι η άμεση εξαγωγή αποτελεσμάτων ενώ η ποσότητα δείγματος είναι πολύ μικρότερη. Το μειονέκτημα της μεθόδου είναι ότι αφενός προστίθεται βάρος στην αναπνευστική οδό, ενώ αυξάνεται και ο νεκρός μηχανικός χώρος, οποίος εξαρτάται από το μέγεθος του θαλάμου που χρησιμοποιείται για την ανίχνευση. Επίσης δεν μπορούν να χρησιμοποιηθούν για μη επεμβατικό μηχανικό αερισμό.



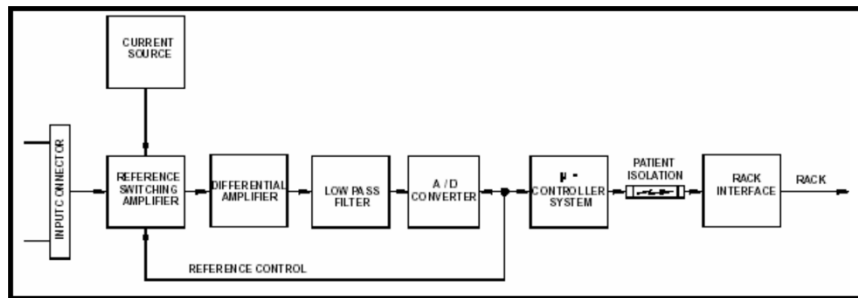
Εικόνα 34. Μπλοκ Διάγραμμα & Λειτουργική Αναπαράσταση Τεχνολογίας Mainstream [36]

- Η τρίτη τεχνολογία είναι η **Microstream**. Ενώ τα Mainstream και Sidestream μόνιτορ βασίζονται στην υπέρυθη απορρόφηση, τα νέου τύπου μόνιτορ ETco<sub>2</sub> χρησιμοποιούν φασματογραφία μοριακής συσχέτισης για μεγαλύτερη ακρίβεια. Το μόνιτορ **Microstream** έχει ταχύ χρόνο αντίδρασης και μπορεί να χρησιμοποιηθεί για επεμβατικό και μη επεμβατικό μηχανικό αερισμό. Οι κύριοι περιορισμοί του είναι είναι το κόστος και η ανάγκη για ξεχωριστό μόνιτορ από το παρακλίνιο μόνιτορ ή τον αναπνευστήρα. [37],[38],[39]

## 2.6 Θερμοκρασία

Η θερμοκρασία είναι μια από τις πιο απλές & βασικές φυσιολογικές παραμέτρους που μετράμε. Η μέτρηση της θερμοκρασίας γίνεται για τον έλεγχο του οργανισμού ως προς την δυνατότητα διατήρησης της θερμοκρασίας αλλά και την δυνατότητα απαγωγής αυτής. Εκτός από τις συνθήκες Υποθερμίας (<35oC) & Υπερθερμίας (>38oC), είναι ιδιαίτερα σημαντική η μέτρηση διαφοράς θερμοκρασίας σε διαφορετικές περιοχές του ανθρωπίνου σώματος. Για τον λόγο αυτό η βαθμίδα μέτρησης θερμοκρασίας συνήθως παρέχει την δυνατότητα μέτρησης 2 θερμοκρασιών και τις διαφορές αυτών. Για την μέτρηση χρησιμοποιείται Θερμίστορ (μεταλλάκτης, μεταβλητής αντίστασης με γραμμική απόκριση στην μεταβολή της θερμοκρασίας) το οποίο είτε τοποθετείται επιδερμικά ή εντός κοιλότητας (αυτί, πρωκτός). Στην παρακάτω εικόνα φαίνεται το σχηματικό διάγραμμα της βαθμίδας μέτρησης θερμοκρασίας. Η πηγή ρεύματος χρησιμοποιείται για την διέλευση

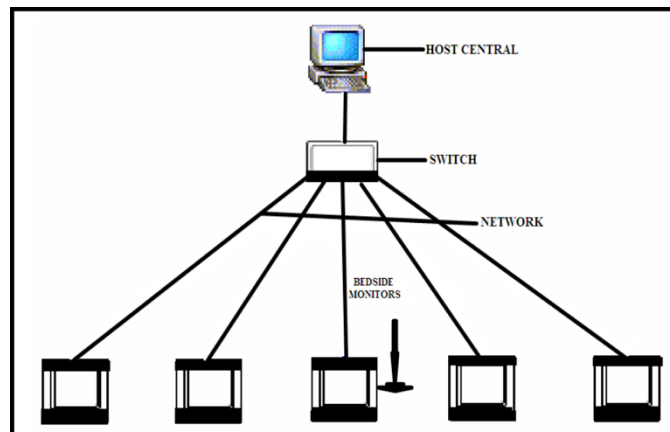
σταθερού ρεύματος μέσω του θερμίστορ. Η μέτρηση βασίζεται στην μεταβολή της πτώσης τάσης πάνω στην αντίσταση λόγω μεταβολής της τιμής της.



Εικόνα 35. Σχηματικό Διάγραμμα Βαθμίδας Μέτρησης Θερμοκρασίας [28]

## 2.7 Κεντρικό Σύστημα Παρακολούθησης (Central Monitoring System)

Καθημερινά οι απαιτήσεις στα σύγχρονα νοσοκομεία αυξάνονται ραγδαία αφού όλο και περισσότεροι ασθενείς επιζητούν ιατρική περίθαλψη για τα προβλήματα υγείας που αντιμετωπίζουν. Το γεγονός αυτό δημιουργεί την ανάγκη για συνεχή και αποτελεσματική επιτήρηση μεγάλου αριθμού ασθενών. Σε συνδυασμό με τους αυξανόμενους οικονομικούς περιορισμούς και το υπερφορτωμένο με καθήκοντα ιατρικό και νοσηλευτικό προσωπικό, ο στόχος του νοσοκομείου να προσφέρει όσο το δυνατόν καλύτερη ιατρική φροντίδα εξανεμίζεται. Τη λύση στα προαναφερθέντα προβλήματα δίνει το κεντρικό σύστημα επιτήρησης (ΚΣΕ). Το κεντρικό σύστημα επιτήρησης είναι μια συσκευή ιατρικών πληροφοριών η οποία εφαρμόζεται ευρέως στον τομέα της κλινικής παρακολούθησης. Για την κατασκευή ενός ΚΣΕ απαιτείται η δημιουργία ενός τοπικού δικτύου (LAN) το οποίο να απαρτίζεται από πολλαπλά παρακλίνια μόνιτορ και έναν κεντρικό υπολογιστή (server) ειδικών προδιαγραφών και ισχύος που έχει το ρόλο του σταθμού εργασίας. Όταν το κέντρο του δικτύου επιτήρησης, ο σταθμός εργασίας περιλαμβάνει εκτός από τον υπολογιστή διάφορα περιφερειακά συστήματα όπως πληκτρολόγιο, ποντίκι, ηχεία και, εκτυπωτή και μία με δυο οθόνες. Ο σταθμός εργασίας πραγματοποιεί την κεντρική επιτήρηση



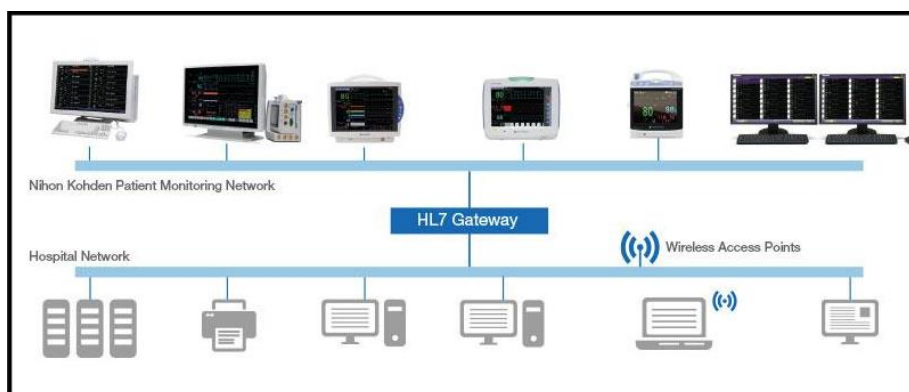
Εικόνα 36. Σχηματικό Διάγραμμα Κεντρικού Συστήματος Παρακολούθησης [41]

συλλέγοντας, παρουσιάζοντας και αναλύοντας τις φυσιολογικές παραμέτρους από τα πολλαπλά παρακλίγια μόνιτορ. Στο κάθε μόνιτορ ανατίθεται μια διαφορετική στατική IP διεύθυνση έτσι ώστε να αναγνωρίζεται από τον κεντρικό υπολογιστή. Στη συνέχεια τα μόνιτορ συνδέονται το κάθε ένα ξεχωριστά με καλώδια Ethernet τα οποία καταλήγουν σε ένα μεταγωγέα (switch). Τέλος ο μεταγωγέας συνδέεται με τον κεντρικό υπολογιστή όπως φαίνεται και στην **εικόνα 36**.

Σε αυτή τη συνδεσμολογία ο κεντρικός υπολογιστής επιτελεί 3 διαφορετικές λειτουργίες:

1. Οργανώνει σε πραγματικό χρόνο τις καταγραφές φυσιολογικών παραμέτρων από πολλαπλά παρακλίγια μόνιτορ.
2. Λειτουργεί σαν βάση δεδομένων. Επίσης μέσω της εφαρμογής που “τρέχει” μπορεί να προγραμματίσει καθώς και να ανανεώσει τα στοιχεία που εμφανίζονται σε κάθε παρακλίιο μόνιτορ.
3. Μπορεί να εκτυπώσει τα δεδομένα που λαμβάνει από τα παρακλίγια μόνιτορ.

Για την λειτουργία του ΚΣΕ δεν απαιτείται η σύνδεσή του στο δίκτυο του νοσοκομείου. Συνήθως το τοπικό δίκτυο είναι απομονωμένο από το υπόλοιπο νοσοκομείο. Η αποθήκευση των δεδομένων πραγματοποιείται στον κεντρικό υπολογιστή και μπορούν να ανακτηθούν και να εκτυπωθούν για μελέτη εντός 240 ωρών. Μετά το πέρας αυτού του χρονικού διαστήματος διαγράφονται. Φυσικά είναι εφικτό αν απαιτηθεί, μέσω μιας πύλης δικτύου (router), να συνδεθεί ο κεντρικός υπολογιστής μέσω του Νοσοκομειακού Πληροφοριακού Συστήματος στο διαδίκτυο. Έτσι για παράδειγμα θα μπορεί το ιατρικό προσωπικό του νοσοκομείου να έχει απομακρυσμένη πρόσβαση στο ιστορικό ενός ασθενούς καθώς και να παρακολουθεί ανά πάσα ώρα και στιγμή τις ζωτικές λειτουργίες του. Επίσης θα υπάρχει η δυνατότητα αποθήκευσης των δεδομένων σε βάση δεδομένων (Database) του νοσοκομείου για μελλοντική ανάκτηση και επεξεργασία.



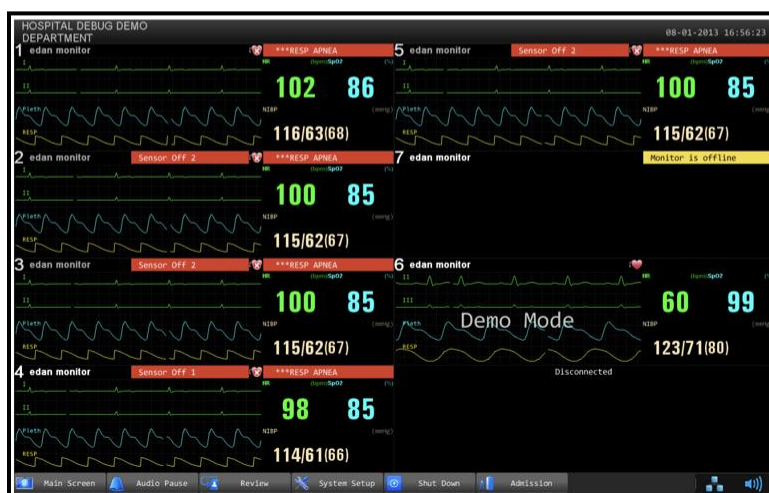
Εικόνα 37. Κεντρικό Σύστημα Απεικόνισης NIHON KOHDEN

Το κεντρικό σύστημα απελευθερώνει το ιατρικό και νοσηλευτικό προσωπικό από πολυάριθμο φόρτο εργασίας όσον αφορά την κλινική παρακολούθηση και βελτιώνει

ιδιαίτερα την ποιότητα επιτήρησης των ασθενών. Ένα μέσο σύστημα κεντρικής επιτήρησης είναι ικανό να συνδεθεί με 32-64 παρακλίνια μόνιτορ, και να απεικονίζει σε πραγματικό χρόνο πληροφορίες για τις φυσιολογικές παραμέτρους των ασθενών. Στο ΚΣΕ μπορούν να απεικονιστούν οι εξής κυματομορφές και μετρήσεις φυσιολογικών παραμέτρων:

- Αναπνοής
- Ηλεκτροκαρδιογραφήματος
- Πληθυσμογραφίας
- Αιματηρής / Μη αιματηρής Πίεσης
- Καпноγραφίας
- Θερμοκρασίας
- Οξυμετρίας

Επίσης το ΚΣΕ μπορεί να προσφέρει τόσο ηχητικές όσο και οπτικές ειδοποιήσεις για τυχόν συναγερμούς που μπορεί να προκύψουν σε κάποιο μόνιτορ έτσι ώστε να κινητοποιηθεί άμεσα το ιατρικό προσωπικό και να αποφευχθεί έγκαιρα ένα επείγον περιστατικό. [40],[41],[42],[43],[44]



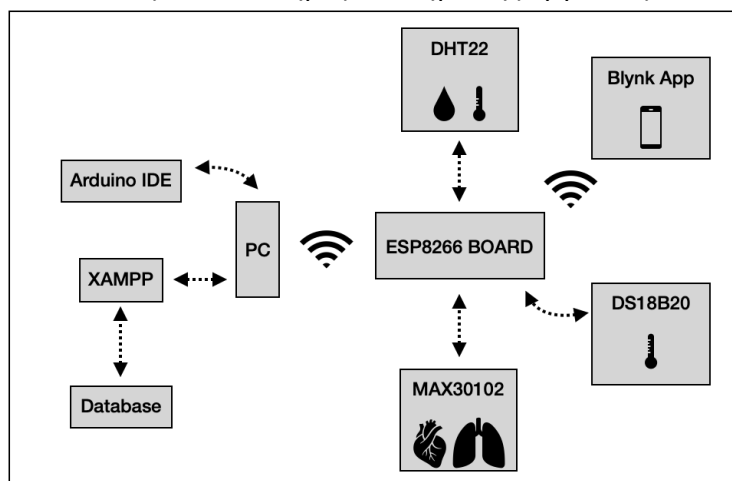
Εικόνα 38. Γραφικό Περιβάλλον Χρήσης Κεντρικού Σταθμού [40],[42]

# 3. ΥΛΙΣΜΙΚΟ

Στο παρόν κεφάλαιο παρουσιάζεται η δομή του συστήματος που αναπτύχθηκε για την επιτήρηση των ζωτικών λειτουργιών ενός ασθενούς και τις συνθήκες χώρου, σε πραγματικό χρόνο. Θα αναλυθούν οι έννοιες του μικροελεγκτή ESP8266, του αισθητήρα θερμοκρασίας και υγρασίας DHT22, του αισθητήρα για την μέτρηση οξυμετρίας και καρδιακών παλμών MAX30102 και του αισθητήρα θερμοκρασίας DS18B20 που χρησιμοποιήσα. Για την υλοποίηση της διπλωματικής εργασίας χρησιμοποιήσα τα εξής:

- PC
- ESP8266 Board
- Αισθητήρα DHT22
- Αισθητήρα MAX30102
- Αισθητήρα DS18B20
- Καλώδιο σύνδεσης ESP8266 - Laptop USB/Micro USB
- Breadboard
- Smartphone (iOS/Android)
- Male/Male Καλώδια
- Κολλητήρι
- Καλάι
- Αντίσταση 4.7 kΩ

Στην **εικόνα 39** απεικονίζεται το πλήρες σύστημα της εργασίας.



Εικόνα 39. Μπλοκ Διάγραμμα Πλήρους Συστήματος Εργασίας

### 3.1 Μικροελεγκτής ESP8266 NodeMCU

Τα τελευταία χρόνια ο κλάδος του IoT (Internet of Things ) έχει προσελκύσει το ενδιαφέρον του τεχνολογικού κόσμου για την δημιουργία project με εφαρμογές στο κλάδο της βιομηχανίας μέχρι την ιατρική. Πλέον τα φυσικά αντικείμενα και ο ψηφιακός κόσμος είναι ενωμένα πιο πολύ από ποτέ. Η εταιρία Espressif Systems έχει κυκλοφορήσει ένα πολύ μικρού μεγέθους μικροελεγκτή με ενσωματωμένη δυνατότητα WiFi, το ESP8266, το οποίο κοστίζει ~8€ κάνοντάς το ιδανικό για την επιτήρηση και τον έλεγχο συσκευών από παντού.

#### 3.1.1 Τμήμα ESP-12E

Η αναπτυξιακή πλατφόρμα αποτελείται από τη μονάδα ESP12-E η οποία περιλαμβάνει το SoC (System on Chip) που αποτελείται από τον μικροεπεξεργαστή Tensilica Xtensa® 32-bit LX106 RISC ο οποίος λειτουργεί στα 80 ή 160 MHz. Περιέχει επίσης 128 KB RAM και 4MB μνήμη Flash (για το πρόγραμμα και την αποθήκευση δεδομένων). Τέλος το ESP8266 ενσωματώνει τον πομποδέκτη 802.11b/g/n HT40 με αποτέλεσμα να μην μπορεί μόνο να συνδέεται σε ένα δίκτυο WiFi αλλά μπορεί επίσης να δημιουργήσει ένα δίκτυο από μόνο του επιτρέποντας άλλες συσκευές να συνδεθούν απευθείας σε αυτό κάνοντάς το ακόμα πιο ευέλικτο και πολυχρηστικό.

<b>ESP12-E</b>
Tensilica Xtensa® 32-bit LX106
80-160 MHz Συχνότητα χρονισμού
128 KB εσωτερική RAM
4 MB μνήμη Flash
Πομποδέκτης 802.11b/g/n HT40



Εικόνα 40. ESP-12E Chip [45]

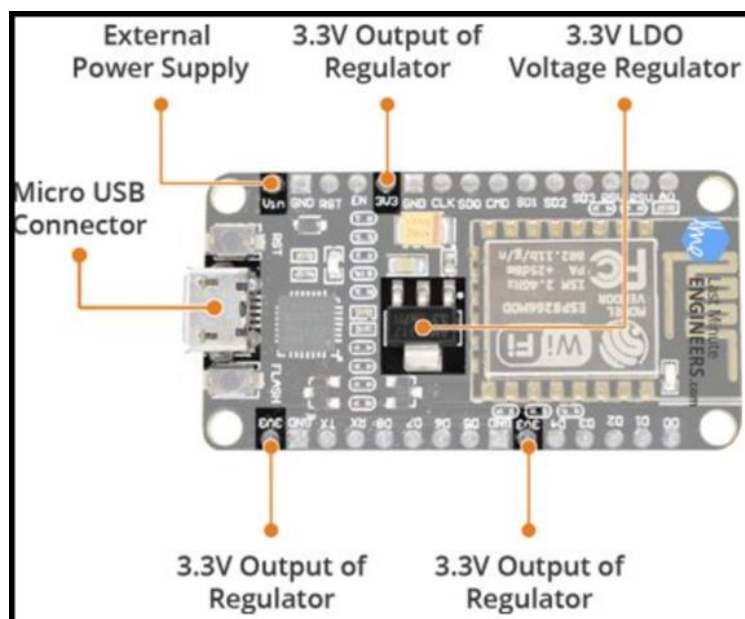


### 3.1.2 Απαιτήσεις ισχύος

Το εύρος τάσης λειτουργίας του ESP8266 κυμαίνεται από 3-3.6V και η πλακέτα είναι εξοπλισμένη ρυθμιστές τάσης ώστε να διατηρήσει την τάση σταθερή στα 3.3V. Επίσης ο μικροελεγκτής τροφοδοτείται με 800mA, τα οποία είναι υπέρ αρκετά αν αναλογιστεί κανείς ότι το ESP8266 καταναλώνει 80mA κατά την εκπομπή ραδιοσυχνοτήτων. Τέλος η έξοδος του ρυθμιστή τάσης, τροφοδοτεί και τα Pin τροφοδοσίας της πλατφόρμας έτσι ώστε να τροφοδοτούνται εξωτερικά στοιχεία όπως οι αισθητήρες της εργασίας.

Απαιτήσεις ισχύος
Τάση Λειτουργίας 2.5-3.6V
Ρυθμιστής Τάσης 3.3V, 800mA
Ρεύμα Λειτουργίας 80mA

Η τροφοδοσία στην πλακέτα παρέχεται μέσω βύσματος Micro USB. Διαφορετικά με μια εξωτερική πηγή τροφοδοσίας των 5V που συνδέεται στο Pin Vin μπορεί να τροφοδοτηθεί η πλακέτα και τα περιφερειακά της.



Εικόνα 41. Στοιχεία Τροφοδοσίας ESP8266 [46]

### 3.1.3 Περιφερειακά & Είσοδοι / Έξοδοι (I/O)

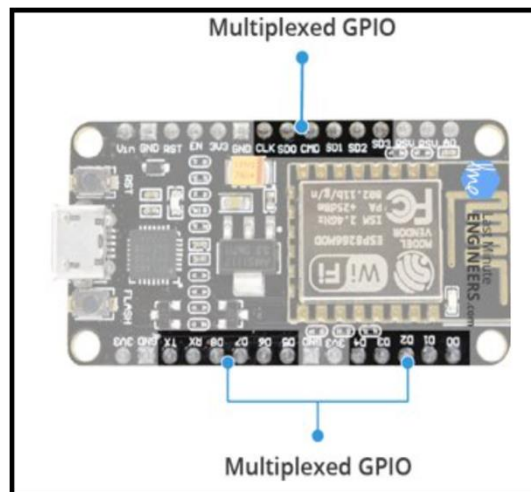
Το NodeMCU ESP8266 έχει συνολικά 17 GPIO Pins και στις 2 μεριές της πλακέτας. Σε αυτά τα Pin μπορούν να ανατεθούν όλων των ειδών περιφερειακά καθήκοντα όπως:

- **Κανάλι ADC** Καναλι ADC 10-bit
- **Διεπαφή UART** Χρησιμοποιείται για το ανέβασμα του κώδικα σειριακά



- **Έξοδοι Pulse Width Modulation (PWM)** Τα Pin Διαμόρφωσης Παλμού χρησιμοποιούνται για τη ρύθμιση φωτεινότητας των LED καθώς και τον έλεγχο σερβοκινητήρων.
- **Διεπαφές SPI, I2C & I2S** Οι διεπαφές SPI & I2C χρησιμοποιούνται για τη σύνδεση κάθε είδους αισθητήρα και περιφερειακών. Η διεπαφή I2S χρησιμοποιείται για την προσθήκη ήχου στο αντίστοιχο προτζεκτ.

<b>Είσοδοι / Έξοδοι</b>
1 κανάλι ADC
2 Διεπαφές UART
4 Έξοδοι PWM
Διεπαφές SPI, I2C & I2S

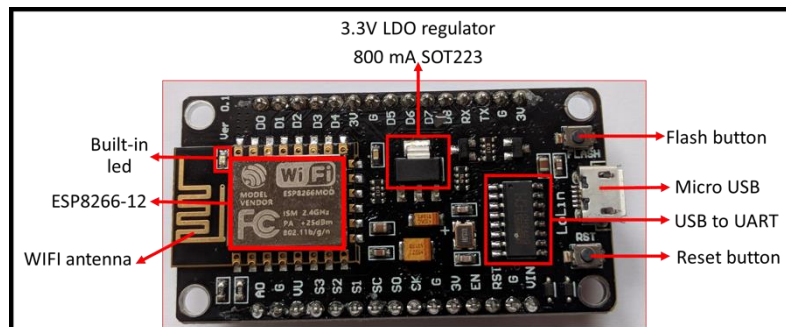


Εικόνα 42. Σύνθετες Διεπαφές [46]

### 3.1.4 Ενδείξεις LED & Κουμπιά στην Πλακέτα

Το ESP8266 αποτελείται από δυο κουμπιά. Το ένα με την ένδειξη RST που βρίσκεται στην κάτω δεξιά γωνία είναι το κουμπί RESET και όπως λέει και το όνομά του χρησιμοποιείται για να κάνει Reset το ολοκληρωμένο. Το δεύτερο κουμπί είναι το FLASH που βρίσκεται στην πάνω δεξιά γωνία και όταν είναι πατημένο χρησιμοποιείται για την αναβάθμιση του λογισμικού. Τέλος η πλακέτα έχει μια

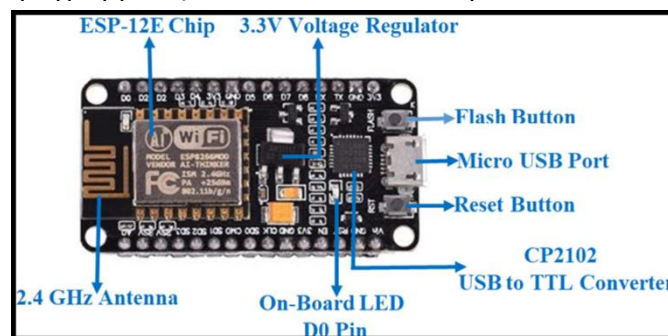
ένδειξη LED που μπορεί να προγραμματιστεί από το χρήστη και είναι συνδεδεμένη με με το Pin D0 της πλακέτας.



Εικόνα 43. Γενική Επισκόπηση Μικροελεγκτή ESP8266 [47]

### 3.1.5 Σειριακή Επικοινωνία

Η πλατφόρμα περιλαμβάνει τη γέφυρα ελέγχου CP2102 USB-to-UART η οποία μετατρέπει το σήμα που λαμβάνεται από το USB σε σειριακό και επιτρέπει στον υπολογιστή να προγραμματίζει και να επικοινωνεί με το ESP8266 chip.

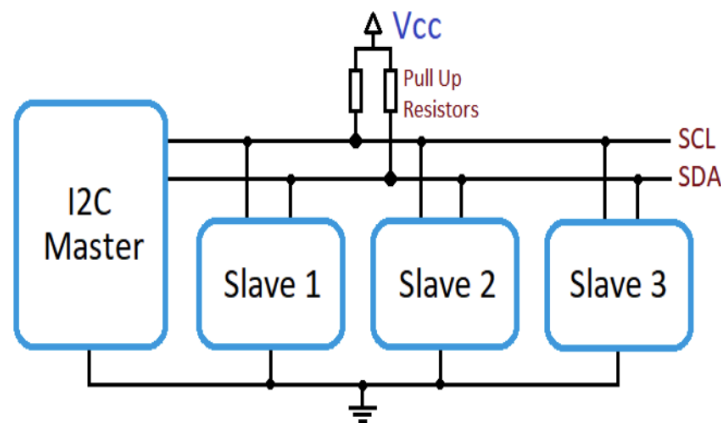


### 3.1.6 Ακροδέκτες του ESP8266 NodeMCU

Το ESP8266 έχει συνολικά 30 Pin για αλληλεπίδραση με εξαρτήματα ή αισθητήρες. Για μεγαλύτερη διευκόλυνση έχουν κατηγοριοποιηθεί ανάλογα με τη λειτουργία τους και παρουσιάζονται ως εξής:

- **Pins Τροφοδοσίας** Υπάρχουν τέσσερα Pins τροφοδοσίας, εκ των οποίων τρία Pin 3.3V & ένα Vin. Το Vin Pin μπορεί να χρησιμοποιηθεί για την απευθείας τροφοδοσία του ESP8266 και των περιφερειακών του με μια εξωτερική πηγή τάσης 5V. Τα Pin με έξοδο 3.3V προέρχονται από τους ρυθμιστές τάσης που βρίσκονται πάνω στην πλακέτα.
- **GND** Είναι τα Pin γείωσης της αναπτυξιακής πλατφόρμας ESP8266 NodeMCU.
- **I2C Pins** Ο διάυλος I2C χρησιμοποιείται ευρέως για την ανάκτηση μετρήσεων ή τον έλεγχο εξοπλισμού σε Arduino, ESP8266 και άλλους μικροελεγκτές με τη χρήση της βιβλιοθήκης Wire.h. Χρησιμοποιείται για την σύνδεση όλων των αισθητήρων και των περιφερειακών που υποστηρίζουν το πρωτόκολλο I2C. Υποστηρίζονται και Master και Slave I2C συσκευές. Ο διάυλος I2C χρειάζεται μόνο δυο καλώδια επικοινωνίας για τη σύνδεση των συσκευών μεταξύ τους.

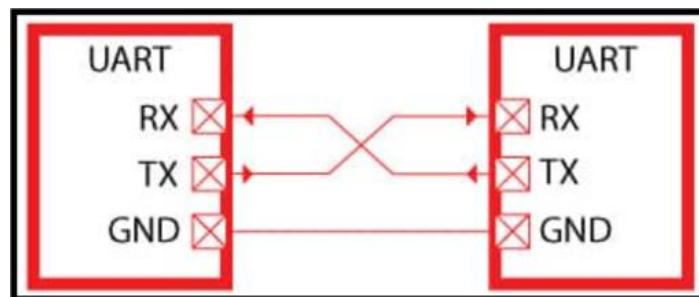
Το ένα καλώδιο είναι το **SDA** (Serial Data Line) το οποίο είναι αμφίδρομο καλώδιο μεταφοράς δεδομένων. Το άλλο είναι το **SCL** (Serial Clock Line) το οποίο είναι αμφίδρομο καλώδιο χρονισμού των επεξεργαστών. Ο μέγιστος αριθμός συσκευών που μπορούν να συνδεθούν εξαρτάται από τον αριθμό των διαθέσιμων διευθύνσεων. Συνήθως 128 συσκευές μπορούν να συνδεθούν ( $2^7$ ). Επίσης η λειτουργικότητα της διεπαφής I2C μπορεί να προγραμματιστεί με την συχνότητα του επεξεργαστή να φτάνει τα 100 kHz το μέγιστο. Τέλος πρέπει να σημειωθεί η συχνότητα ρολογιού της διεπαφής I2C πρέπει να είναι υψηλότερη από την Slave συσκευή με την πιο αργή συχνότητα ρολογιού.



Εικόνα 44. Δίαυλος I2C [48]

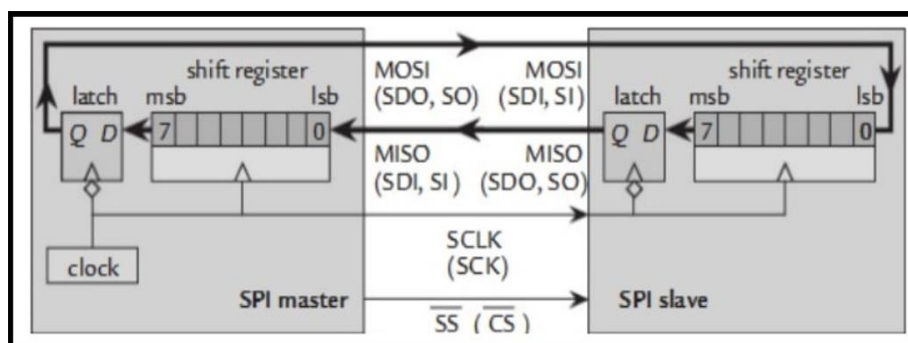
- **GPIO Pins** Ο μικροελεγκτής αποτελείται από 17 εισόδους και εξόδους, στις οποίες μπορούν να ανατεθούν λειτουργίες όπως Διαμόρφωσης Παλμού, Inter-Integrated Circuit (I2C), UART κ.ά για την διασύνδεση κάθε είδους αισθητήρα. Το κάθε ψηφιακό ενεργοποιημένο Pin μπορεί να διαμορφωθεί για εσωτερικό pull-up ή pull-down, ή να διαμορφωθεί για υψηλή αντίσταση εισόδου.
- **Κανάλι ADC** Επίσης ο μικροελεγκτής έχει ενσωματωμένο έναν αναλογικό σε ψηφιακό μετατροπέα (ADC) διακριτικής ικανότητας 10 bit. Με τον ADC μπορεί να ελεγχθεί η τάση εισόδου του pin TOUT καθώς και η τάση παροχής του Pin VDD3P3. Αυτές οι δυο λειτουργίες δεν μπορούν να λειτουργήσουν ταυτόχρονα.
- **UART Pins** Η επικοινωνία UART προέρχεται από το Universal asynchronous receiver-transmitter. Είναι ένα συγκεκριμένο τμήμα υλισμικού που πραγματοποιεί ασύγχρονη σειριακή επικοινωνία. Παρέχει δυνατότητες για την διαμόρφωση της μορφής των δεδομένων και ταχυτήτων διάδοσης σε διαφορετικούς ρυθμούς baud. Αποτελεί ένα από τα πιο απλά πρωτόκολλα επικοινωνίας μεταξύ δυο συσκευών. Με αυτό το πρωτόκολλο μεταφέρονται δεδομένα μεταξύ συσκευών συνδέοντας δυο καλώδια, το ένα το οποίο είναι το καλώδιο μετάδοσης ενώ το άλλο είναι το καλώδιο λήψης. Τα δεδομένα

μεταφέρονται ανά bit ψηφιακά από τη μια συσκευή στην άλλη. Το βασικό χαρακτηριστικό αυτής της τεχνολογίας είναι ότι οι συσκευές που εμπλέκονται μπορούν να λειτουργούν με διαφορετική συχνότητα. Για παράδειγμα, μέσω της σειριακής επικοινωνίας μπορούν να αλληλοεπιδράσουν δυο συσκευές ακόμα και αν έχουν διαφορετικές συχνότητες. Αυτό όμως που καθιστά δυνατή την επικοινωνία μεταξύ των δυο συσκευών είναι ο προκαθορισμένος ρυθμός μετάδοσης baud ο οποίος ορίζεται στη μνήμη Flash των συσκευών. Η πρώτη συσκευή που εκπέμπει, παίρνει τα bit από τα δεδομένα και τα μεταδίδει σε μια διαδοχική μορφή. Η δεύτερη συσκευή η οποία είναι ο δέκτης, επανασυναρμολογεί τα bits σε ένα ολοκληρωμένο byte. Το ESP8266 NodeMCU έχει 2 διεπαφές UART, το UART0 και το UART1, οι οποίες προσφέρουν ασύγχρονη επικοινωνία (RS232 & RS485) και μπορούν να επικοινωνήσουν στα 4.5 Mbps. Το UART0 (Pin TXD0, RXD0, RST0 & CTS0) μπορούν να χρησιμοποιηθούν για επικοινωνία. Ωστόσο το UART1 (Pin TXD1) χρησιμοποιείται μόνο για μεταφορά δεδομένων.



Εικόνα 45. Συνδεσμολογία UART επικοινωνίας [49]

- Pins SPI** Η επικοινωνία. SPI είναι ένα ψηφιακό πρωτόκολλο επικοινωνίας. Μέσω αυτού του πρωτοκόλλου η μεταφορά των δεδομένων γίνεται σειριακά στέλνοντας ένα bit τη φορά μεταξύ των συσκευών που εμπλέκονται. Πολλές αναλογικές συσκευές όπως αναλογικοί αισθητήρες, αισθητήρες θερμοκρασίας επίσης παρέχουν τη δυνατότητα επικοινωνία μέσω του πρωτοκόλλου SPI. Το πρωτόκολλο SPI όπως αναφέρθηκε είναι σειριακή και συγχρονισμένη διεπαφή. Συγχρονισμένη σημαίνει ότι χρειάζεται σήμα από το ρολόι για την μεταφορά και τη λήψη δεδομένων και το σήμα του ρολογιού



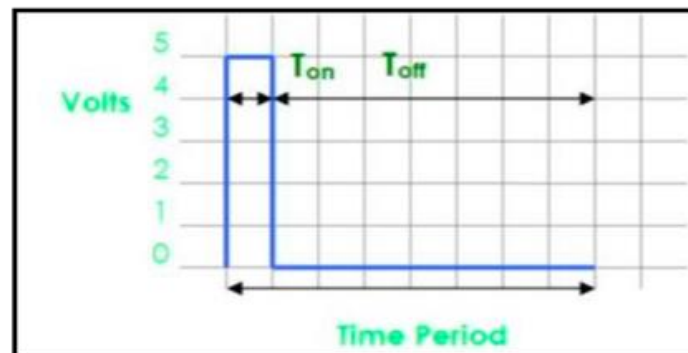
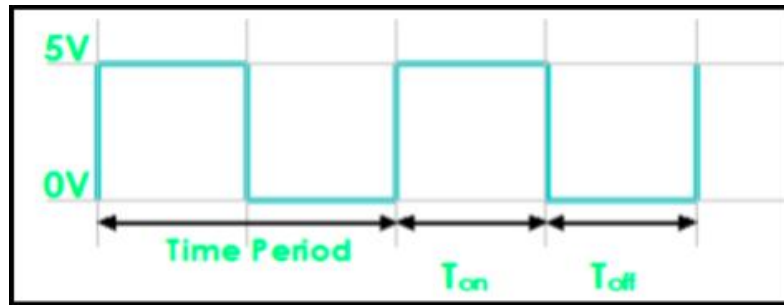
Εικόνα 46. Σύνδεση SPI μεταξύ 2 συσκευών [50]

συγχρονίζεται μεταξύ της συσκευής Master και Slave. Το σήμα του ρολογιού ελέγχει τότε πρέπει να σταλούν τα δεδομένα στη συσκευή Slave και τότε πρέπει να διαβαστούν από αυτή. Επίσης μόνο η συσκευή Master μπορεί να ελέγξει το ρολόι και να δημιουργήσει σήμα ρολογιού σε όλες τις συσκευές Slave. Και η Master και οι Slave συσκευές μπορούν να ανταλλάξουν δεδομένα μεταξύ τους αλλά πάντα πρέπει να υπάρχει σήμα ρολογιού. Το ESP8266 διαθέτει δυο SPI (SPI & HSPI) σε λειτουργίες Slave και Master.

- **PWM Pins** Η πλακέτα έχει 4 κανάλια Διαμόρφωσης Παλμού (PWM). Η έξοδος του PWM μπορεί να εφαρμοστεί προγραμματιστικά και να χρησιμοποιηθεί για τον έλεγχο σερβοκινητήρων και LED. Το εύρος συχνότητας του PWM μπορεί να μεταβληθεί από 1000μs σε 10000μs (100Hz-1kHz). Με την τεχνική της διαμόρφωσης εύρους παλμών ένα μεταβλητό αναλογικό σήμα δημιουργείται με τη χρήση ψηφιακού κυκλώματος. Η φιλοσοφία πίσω από το P.W.M είναι απλή, αφού βασίζεται στο άνοιγμα και το κλείσιμο ενός διακόπτη σε ομοιόμορφα και επαναλαμβανόμενα χρονικά διαστήματα. Τα τελευταία χρόνια η τεχνική P.W.M έχει εφαρμογή σε ευρύ φάσμα κλάδων και εφαρμογών όπως ο έλεγχος τάσης και ρεύματος, στους ηλεκτροκινητήρες, στα U.P.S κ.ά. Ένα από τα κυριότερα πλεονεκτήματα του ψηφιακού ελέγχου των αναλογικών κυκλωμάτων είναι η δραστική μείωση της κατανάλωσης ενέργειας. Με απλά λόγια το P.W.M είναι ένας πολύ αποτελεσματικός τρόπος για τον έλεγχο του αναλογικού σήματος σε ένα κύκλωμα και τα συστήματά του μέσω της ψηφιακής εξόδου των μικροελεγκτών. Τα κύρια στοιχεία που χαρακτηρίζουν ένα σήμα P,W.M είναι ο κύκλος και ο χρόνος λειτουργίας του. Ο κύκλος λειτουργίας μετριέται σε επί τις εκατό (%) και η συχνότητα διαμόρφωσης σε Hz. Το πλάτος του σήματος ελέγχου λαμβάνει 2 τιμές, μια για την κατάσταση αγωγιμότητας (ON) και μία για την κατάσταση διακοπής (OFF). Η PWM λειτουργεί βάσει ενός ρολογιού (timer) ώστε να βγάζει τετραγωνικό παλμό κάποιας συχνότητας. Η εντολή analogWrite() αλλάζει το εύρος του παλμού στον χρόνο (duty cycle) ενώ η συχνότητα παραμένει σταθερή. Το εύρος του παλμού (duty-cycle) ορίζεται από τον τύπο [51]:

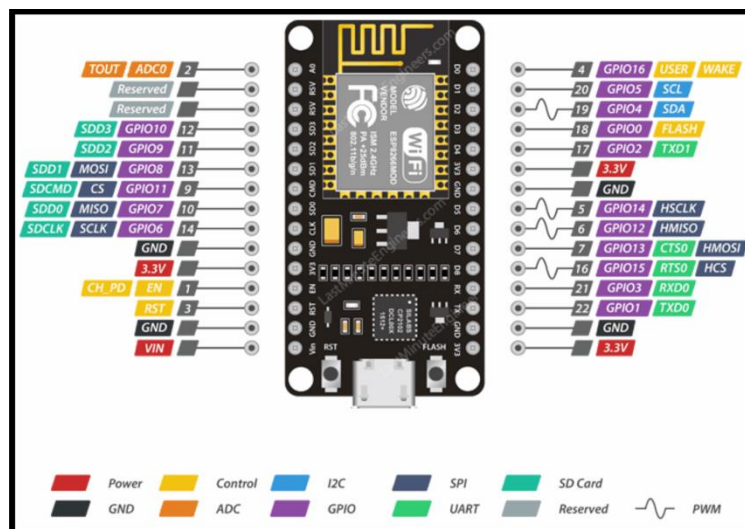
$$DutyCycle(\%) = \frac{T_{on}}{TimePeriod} \times 100$$

Παρακάτω ακολουθούν μερικά παραδείγματα διαμόρφωσης εύρους παλμού:



Εικόνα 47. Παραδείγματα Διαμόρφωσης Εύρους Παλμού [51]

- **Pin Ελέγχου** Αυτα τα Pin χρησιμοποιούνται για τον έλεγχο του chip ESP8266. Περιλαμβάνουν τα Pin “Chip Enable” (EN), RESET (RST) και WAKE.
- A. **EN Pin:** Το chip ESP8266 ενεργοποιείται όταν το Pin EN βρίσκεται σε υψηλή κατάσταση. Όταν βρίσκεται σε LOW κατάσταση το chip δουλεύει με ελάχιστη ισχύ.
- B. **RST Pin:** Το Pin RST χρησιμοποιείται για να γίνει RESET το chip.
- C. **WAKE Pin:** Το WAKE Pin χρησιμοποιείται για να αφυπνίσει το chip.



Εικόνα 48. Ακροδέκτες Μικροελεγκτή ESP8266 [46]

### 3.1.7 Σύγκριση ESP8266 NodeMCU με το Arduino

Διχως αμφιβολία η πιο διαδεδομένη αναπτυξιακή πλατφόρμα είναι αυτή του Arduino Uno. Για πολλά χρόνια το Arduino έχει υπάρξει ο “εγκέφαλος” για χιλιάδες εργασίες, από καθημερινά αντικείμενα μέχρι πολύ σύνθετα επιστημονικά μηχανήματα. Μία διεθνής κοινότητα από δημιουργούς, μαθητές, καλλιτέχνες, προγραμματιστές και επαγγελματίες ασχολήθηκαν με αυτή την ανοιχτού-κώδικα πλατφόρμα, οι συνεισφορές τους έχουν χρησιμοποιηθεί από πρωτάρηδες μέχρι επαγγελματίες. Από την άλλη μεριά το ESP8266 NodeMCU έχει αποκτήσει μεγάλη δημοφιλία τα τελευταία χρόνια με την αύξηση ενδιαφέροντος για IoT (Internet of Things) εφαρμογές. Παρακάτω θα παρουσιαστούν συνοπτικά τα πλεονεκτήματα αλλά και τα μειονεκτήματα της κάθε πλατφόρμας έναντι της άλλης.

- **Πλεονεκτήματα του ESP8266 NodeMCU έναντι του Arduino Uno**

1. **Καλύτερος επεξεργαστής & Μνήμη** Το ESP8266 NodeMCU έχει ταχύτητα ρολογιού 80 MHz και 4MB μνήμης Flash. Από την άλλη το Arduino έχει ταχύτητα ρολογιού 16 MHz και διατηρήσιμη μνήμη Flash 32 KB.
2. **Ενσωματωμένο Πρωτόκολλο Διαδικτυου TCP/IP - IoT** Το NodeMCU περιέχει πρωτόκολλο WiFi και μπορεί να συνδεθεί στο διαδίκτυο μέσω WiFi. Ενδείκνυται για εφαρμογές IoT.
3. **Φθηνότερο** Το NodeMCU είναι ένας μικροελεγκτής που βασίζεται στο chip ESP8266, το οποίο είναι πολύ πιο φθηνό σε σύγκριση με το Arduino Uno (8€ το NodeMCU έναντι 25€ το Arduino).
4. **Εύχρηστο με το Breadboard και μικρό σε μέγεθος** Το NodeMCU μπορεί εύκολα να ενσωματωθεί σε ένα Breadboard και να πραγματοποιηθούν πολλά σχέδια κυκλωμάτων. Από την άλλη το Arduino δεν μπορεί να ενσωματωθεί σε ένα Breadboard λόγω του μεγέθους του.

- **Μειονεκτήματα του ESP8266 NodeMCU έναντι του Arduino Uno**

1. **Εκμάθηση προγραμματισμού νέων λειτουργιών** Για τον προγραμματισμό του NodeMCU, είναι αναγκαία η εκμάθηση εφαρμογών διαδικτύου για την διεκπεραίωση νέων εργασιών.
2. **Αναλογικά Pin** Το NodeMCU έχει μόνο ένα Αναλογικό Pin εισόδου. Από την άλλη το Arduino UNO περιέχει 6 Αναλογικά Pin.
3. **Χαμηλότερο επίπεδο τάσης** Σε σπάνιες περιπτώσεις, εξωτερικές πλακέτες δεν είναι συμβατές με το NodeMCU.

- **Πλεονεκτήματα του Arduino Uno έναντι του ESP8266 NodeMCU**

1. **Εκτεταμένη υποστήριξη** Το Arduino Uno έχει εξαιρετικής ποιότητας περιεχόμενο, διαθέσιμο online καθώς περισσότεροι άνθρωποι σε όλο τον κόσμο συμμετέχουν στην κοινότητα του Arduino. Σε σύγκριση με το NodeMCU υπάρχει μεγαλύτερη πληθώρα περιεχομένου εκμάθησης.

2. **Εύκολη διεπαφή με αισθητήρες και μηχανισμούς** Το Arduino Uno μπορεί εύκολα να συνδεθεί με αισθητήρες και ενεργοποιητές καθώς υποστηρίζει μόντουλς με 3.3V και 5V σε σύγκριση με το NodeMCU.
3. **Δυνατότητα διαχείρισης τάσης** Το Arduino Uno μπορεί να διαχειριστεί τάση 20V, ενώ το NodeMCU μπορεί να αντέξει μέχρι 12V
4. **Ρεύμα ανά Pin** Το Arduino Uno μπορεί να διαχειριστεί ρεύμα μέχρι 40 mA ανά Pin. Το NodeMCU μπορεί να διαχειριστεί μέχρι 12 mA ρεύμα.

- **Μειονεκτήματα του Arduino Uno έναντι του ESP8266 NodeMCU**

1. **Περιορισμοί στην μνήμη και τον επεξεργαστή** Το Arduino Uno έχει μικρότερη μνήμη και επεξεργαστικές δυνατότητες σε σύγκριση με το NodeMCU. Επίσης δεν έχει ενσωματωμένες IoT δυνατότητες.
2. **Πιο ακριβό** Το Arduino είναι πιο ακριβό από το NodeMCU
3. **Δεν έχει δυνατότητα ενσωμάτωσης σε Breadboard** Το Arduino δεν μπορεί να ενσωματωθεί σε Breadboard λόγω του μεγέθους και του προσανατολισμού των pin του.

Εν κατακλείδι και οι δυο μικροελεγκτές είναι εξαιρετικοί, ο καθένας στον τομέα του. Ανάλογα με τις απαιτήσεις, ο καθένας μπορεί να επιλέξει τον καλύτερο μικροελεγκτή που ταιριάζει στις προδιαγραφές που επιθυμεί. Και οι δυο μικροελεγκτές έχουν τα πλεονεκτήματα και μειονεκτήματά τους. [45],[46],[47],[48],[49],[50],[51],[52],[53]

## 3.2 Αισθητήρας DS18B20

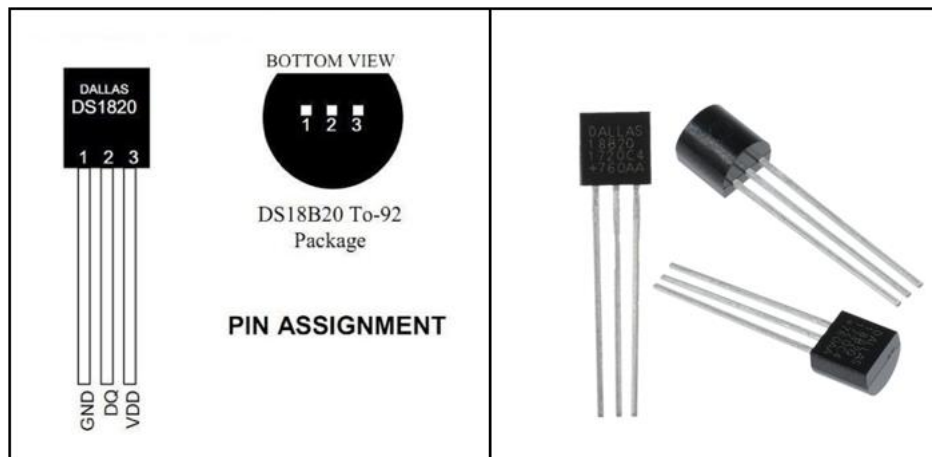
### 3.2.1 Γενικές πληροφορίες

Ένας από τους πιο εύκολους και φθηνούς τρόπους για την μέτρηση θερμοκρασίας σε ένα πρότζεκτ με χρήση του Arduino ή κάποιου παρεμφερούς μικροελεγκτή είναι η χρήση του αισθητήρα θερμοκρασίας DS18B20 1-Wire. Ο συγκεκριμένος αισθητήρας είναι αρκετά ακριβής στις μετρήσεις του και δεν απαιτεί επιπλέον εξωτερικά εξαρτήματα για να λειτουργήσει. Επίσης μπορεί να μετρήσει θερμοκρασίες οι οποίες κυμαίνονται από τους  $-55^{\circ}\text{C}$  μέχρι τους  $+125^{\circ}\text{C}$  με ακρίβεια  $\pm 0,5^{\circ}\text{C}$ . Σε σύγκριση με ένα τυπικό θερμίστορ, στον DS18B20 πραγματοποιείται άμεσα ο ψηφιακός υπολογισμός αρχικά και στη συνέχεια η αποθήκευση της τιμής της θερμοκρασίας στη μνήμη ROM μεγέθους 2 bytes ή 16 bits που διαθέτει. Η ακρίβεια της μετρούμενης τιμής μπορεί να προσαρμοστεί από 9-12 bit ανάλογα με τις απαιτήσεις του χρήστη. Η προεπιλεγμένη διακριτική ικανότητα του αισθητήρα από το εργοστάσιο είναι στα 12 bit (ακρίβεια  $0.0625^{\circ}\text{C}$ ). Τέλος ο αισθητήρας μπορεί να τροφοδοτηθεί με πηγή



τροφοδοσίας από 3.3 - 5V και καταναλώνει μόνο 1mA κατά την ενεργή περίοδο μέτρησης της θερμοκρασίας. Κατασκευάζεται από την Dallas Semiconductor Corp. και ένα από τα βασικά πλεονεκτήματα του αισθητήρα είναι ότι μπορεί να έχει αμφίδρομη επικοινωνία με τον μικροελεγκτή μέσω ενός μόνο ψηφιακού pin. Ο αισθητήρας συνήθως κατασκευάζεται σε δυο τύπους:

- Ο ένας που είναι σε περίβλημα TO-92 και μοιάζει με ένα συνηθισμένο τρανζίστορ.



Εικόνα 49. Ο Αισθητήρας DS18B20 σε περίβλημα τρανζίστορ

- Ο δεύτερος είναι σε μορφή αδιάβροχου probe το οποίο είναι πιο χρήσιμο όταν χρειάζεται να μετρηθεί κάτι που βρίσκεται σε μεγαλύτερη απόσταση, κάτω από νερό ή κάτω από το έδαφος.



Εικόνα 50. Ο Αισθητήρας DS18B20 σε μορφή Probe

Στον πίνακα παρουσιάζονται κάποιες από τις βασικές προδιαγραφές του αισθητήρα:

Τροφοδοσία	3.3 - 5 V
Κατανάλωση ρεύματος	1mA
Εύρος θερμοκρασίας	-55 - 125 °C
Ακρίβεια	±0.5°C
Διακριτική ικανότητα	9 με 12 bit (επιλέξιμο)
Χρόνος μέτρησης	< 750ms

### 3.2.2 Ακροδέκτες του DS18B20 και σύνδεση με τον μικροελεγκτή

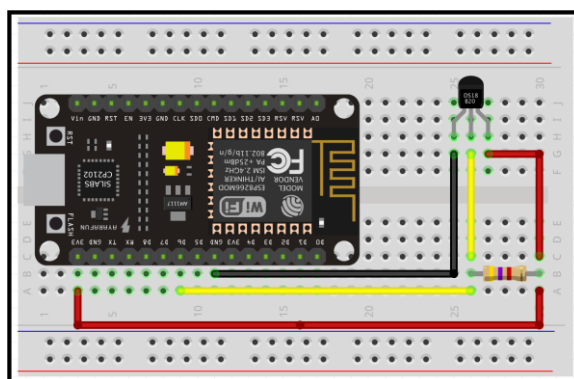
Και οι δυο τύποι αισθητήρων που αναφέρθηκαν προηγουμένως αποτελούνται από 3 ακροδέκτες.

- **GND** Είναι το Pin της γείωσης
- **DQ** Είναι το Pin του διαύλου δεδομένων 1-Wire που συνδέεται σε ψηφιακό Pin I/O του μικροελεγκτή και διαβάζει τις τιμές θερμοκρασίας.
- **VDD** Είναι το Pin που τροφοδοτεί τον αισθητήρα με τάση από 3.3 - 5 V.

Η συνδεσμολογία και των δυο τύπων αισθητήρων με τον μικροελεγκτή είναι αρκετά εύκολη και πανομοιότυπη. Ανάλογα με τον τύπο του αισθητήρα λοιπόν η συνδεσμολογία είναι η εξής:

- **Αισθητήρας με περίβλημα**

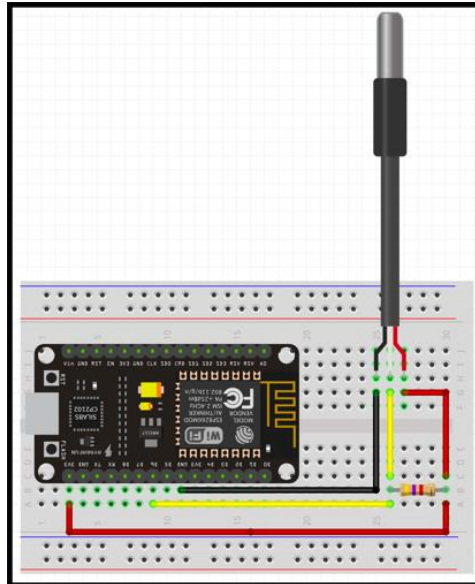
Αρχικά ο ακροδέκτης **VDD** συνδέεται με το Pin των **3.3V** μικροελεγκτή ESP8266. Ο ακροδέκτης **GND** συνδέεται με το Pin **GND** του μικροελεγκτή. Ο ακροδέκτης **DQ** συνδέεται με το ψηφιακό Pin **D7 (GPIO13)**. Τέλος συνδέεται η pull-up αντίσταση **4.7 kΩ** μεταξύ του **DQ** και των 3.3V έτσι ώστε να διατηρηθεί σταθερή η μεταφορά δεδομένων.



Εικόνα 51. Σύνδεση του αισθητήρα σε μορφή τρανζίστορ με το ESP8266 [54]

- Αισθητήρας σε μορφή Probe

Αντίστοιχα με τον αισθητήρα με περίβλημα γίνεται και η συνδεσμολογία του **Probe**.

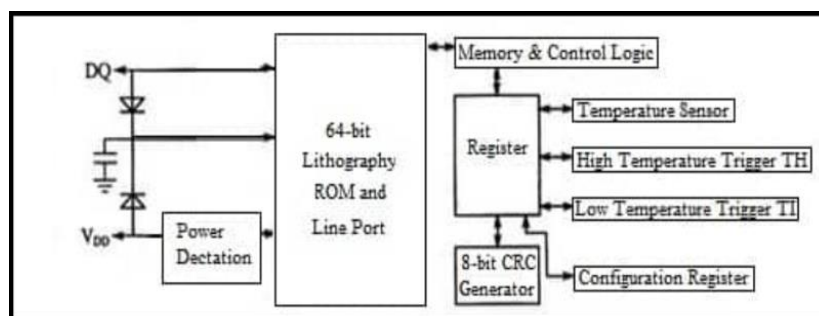


Εικόνα 52. Σύνδεση του αισθητήρα σε μορφή Probe με το ESP8266 [54]

### 3.2.3 Εσωτερική Δομή του DS18B20

Η εσωτερική δομή του DS18B20 αποτελείται κυρίως από 4 τμήματα:

- 64-bit ROM (Read-Only Memory) λιθογραφίας
- Αισθητήρα θερμοκρασίας
- Σύστημα συναγερμού θερμοκρασίας TH & TL
- Καταχωρητή διαμόρφωσης



Εικόνα 53. Εσωτερική Δομή του DS18B20 [55]

### 3.2.4 Αρχή λειτουργίας DS18B20

Όπως αναφέρθηκε και προηγουμένως, η μνήμη ROM κάθε αισθητήρα έχει μέγεθος δυο bytes ή 16 bits και έχουν ως σκοπό την αποθήκευση της μέτρησης της θερμοκρασίας. Το πρώτο byte ονομάζεται πιο σημαντικό (most significant byte ή

MSB) και το δεύτερο λιγότερο σημαντικό (least significant byte ή LSB). Στο MSB ανήκουν τα ψηφία από 8-15 ενώ στο LSB τα ψηφία από 0-7.

Στην εικόνα 57 φαίνεται η αναπαράσταση των ψηφίων της μνήμης ROM του αισθητήρα DS18B20. Μπορούμε να δούμε ότι τα bits 11, 12, 13, 14, 15 του MSB έχουν πάντα την ίδια τιμή. Ο ρόλος αυτών των bit είναι να δείχνουν το πρόσημο της μετρούμενης θερμοκρασίας. Αν για παράδειγμα μετρήθηκε αρνητική θερμοκρασία τότε αυτά τα bits παίρνουν τιμή “1” ενώ αν μετρηθεί θετική τιμή λαμβάνουν την τιμή “0”. Συνεπώς αν και το μέγεθος της μνήμης ROM είναι 16 bit έχουμε εν τέλει 12 διακριτά.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LS BYTE	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>-1</sup>	2 <sup>-2</sup>	2 <sup>-3</sup>	2 <sup>-4</sup>
	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
MS BYTE	S	S	S	S	S	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>

S = SIGN

Εικόνα 54. Αναπαράσταση των ψηφίων της μνήμης ROM του αισθητήρα DS18B20 [56]

Για το ακέραιο τμήμα κάθε μέτρησης χρησιμοποιούνται οκτώ bit (bit 4-11) ενώ για το δεκαδικό τέσσερα (bit 0-3). Τα ψηφία 4-11 χρησιμοποιούνται πάντα ενώ τα ψηφία 0-3 χρησιμοποιούνται ανάλογα με την ανάλυση που έχει επιλεγεί. Είναι εμφανές ότι η ανάλυση των 12 bit δίνει μετρήσεις με μεγαλύτερη διακριτική ικανότητα, παρόλα αυτά χρειάζεται μεγαλύτερο χρονικό διάστημα για να επιτευχθεί η μέτρηση. Όταν ζητείται η μετατροπή της μέτρησης της διαφοράς δυναμικού σε θερμοκρασία, ο αισθητήρας DS18B20 λαμβάνει πολλαπλές μετρήσεις και υπολογίζει τον μέσο όρο τους πριν αναφέρει το αποτέλεσμα. Έτσι όσο μεγαλύτερη είναι η ανάλυση τόσο μεγαλύτερος είναι και ο αριθμός των μετρήσεων των οποίων υπολογίζεται ο μέσος όρος άρα και μεγαλύτερος ο χρόνος για να ολοκληρωθεί η μετατροπή. Με βάση το Datasheet του κατασκευαστή η σχέση μεταξύ διακριτικής ικανότητας και χρόνου μετατροπής φαίνεται στον παρακάτω πίνακα(). Αφού ολοκληρωθεί η μετατροπή της διαφοράς δυναμικού σε θερμοκρασία, η τιμή αποθηκεύεται στη μνήμη ROM του αισθητήρα και μεταφέρεται στο μικροελεγκτή μέσω του διαύλου 1-Wire.

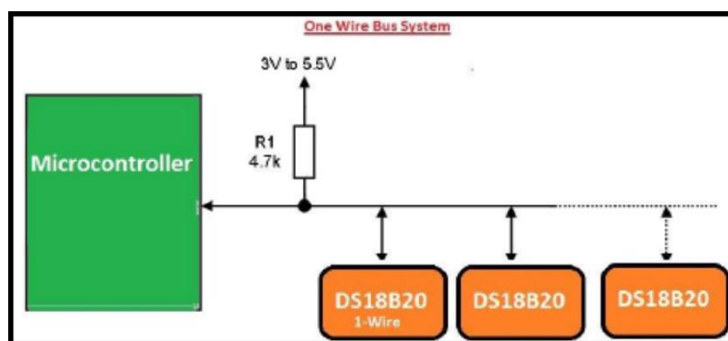
RESOLUTION (BITS)	MAX CONVERSION TIME	
9	93.75ms	(t <sub>CONV</sub> /8)
10	187.5ms	(t <sub>CONV</sub> /4)
11	375ms	(t <sub>CONV</sub> /2)
12	750ms	(t <sub>CONV</sub> )

Εικόνα 55. Αντιστοιχία ανάλυσης με χρόνο μετατροπής για τον DS18B20

### 3.2.5 One-Wire Bus

Ένα από τα μεγαλύτερα πλεονεκτήματα του αισθητήρα DS18B20 είναι ότι μπορούν να συνδεθούν πολλαπλοί τέτοιοι αισθητήρες μαζί στον ίδιο δίαυλο 1-Wire. Κάθε DS18B20 έχει ένα μοναδικό σειριακό κωδικό 64-bit (ROM) ενσωματωμένο σε αυτό κατά την κατασκευή στο εργοστάσιο που το διαφοροποιεί από ένα άλλο. Με απλά λόγια ο κάθε αισθητήρας έχει διαφορετική διεύθυνση και “επικαλώντας” την μπορεί κάποιος να δει τιμές από διαφορετικούς αισθητήρες. Αυτό το χαρακτηριστικό μπορεί να είναι μεγάλο πλεονέκτημα όταν απαιτηθεί ο έλεγχος πολλαπλών DS18B20 που είναι διανεμημένα σε μεγαλύτερο χώρο.

Το πρωτόκολλο 1-Wire ακολουθεί τη λογική του συστήματος Master-Slave, όπου 1 συσκευή Master μπορεί να ελέγχει ή να επικοινωνεί με όλες τις συσκευές Slave. Στην προκειμένη περίπτωση ο μικροελεγκτής έχει το ρόλο του Master και όλες οι συσκευές 1-Wire όπως ο DS18B20 λειτουργούν σαν Slave. Αν συνδεθεί μόνο μια συσκευή με τον μικροελεγκτή τότε το σύστημα λέγεται **Single Drop** ενώ αν συνδεθούν πολλαπλές συσκευές μέσω ενός μονού Pin τότε το σύστημα αποκαλείται Multi Drop.



Εικόνα 56. Σύστημα 1-Wire Bus [57]

Ο βασικός τρόπος λειτουργίας του αισθητήρα DS18B20 είναι ο άμεσος ψηφιακός υπολογισμός και η αποθήκευση της τιμής της θερμοκρασίας στη μνήμη ROM. Για να μπορέσει να πραγματοποιηθεί αυτή η διαδικασία πρέπει να προηγηθεί μια αλληλουχία ενεργειών από τον μικροελεγκτή προς τον αισθητήρα DS18B20 έτσι ώστε να αποκτήσει πρόσβαση στη μνήμη ROM. Η διαδικασία είναι η εξής:

1. **Αρχικοποίηση** Όλες οι διεργασίες που εκτελούνται στον δίαυλο 1-Wire ξεκινούν με την ακολουθία αρχικοποίησης. Η συσκευή Master, που στην προκειμένη περίπτωση είναι ο μικροελεγκτής, εκπέμπει ένα παλμό επαναφοράς (reset pulse) ενώ στη συνέχεια οι συσκευές Slave (πχ ο αισθητήρας DS18B20) εκπέμπουν παλμούς παρουσίας (presence pulse). Με τον παλμό παρουσίας η συσκευή Master αναγνωρίζει τις συσκευές με ρόλο Slave που βρίσκονται στο δίαυλο και είναι έτοιμες να εκτελέσουν τις εντολές της.
2. **Ανάγνωση κωδικών 64-bit** Μόλις ανιχνευτεί ο παλμός παρουσίας από τον μικροελεγκτή στον δίαυλο κάποιας συσκευής που είναι ενεργοποιημένη,

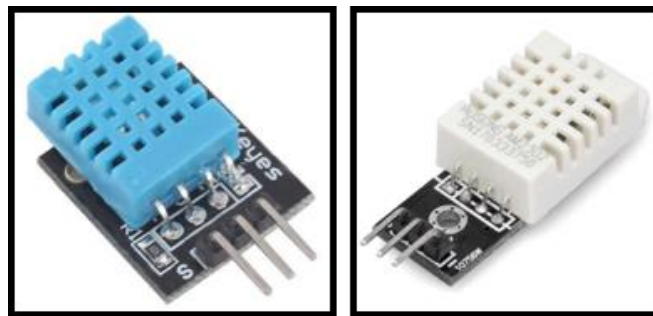
μπορεί να εκδοθεί η εντολή ανάγνωσης του μοναδικού κωδικού 64-bit που φέρει η κάθε Slave συσκευή. Με την εντολή ανάγνωσης ο μικροελεγκτής είναι σε θέση να αναγνωρίσει και να καθορίσει τον τύπο και το πλήθος των συσκευών Slave στο δίαυλο. Στην περίπτωση που υπάρχουν πολλές Slave συσκευές συνδεδεμένες, ο κωδικός 64-bit επιτρέπει στην Master συσκευή να έχει αποκλειστική επικοινωνία με αυτήν. Για την έκδοση κάποιας εντολής λειτουργίας του DS18B20, είναι αναγκαία η έκδοση της εντολής ανάγνωσης.

3. **Εντολή λειτουργίας DS18B20** Εφόσον υλοποιηθούν τα παραπάνω βήματα με τη συγκεκριμένη σειρά και ο μικροελεγκτής γνωρίζει το πλήθος των Slave συσκευών μπορεί πλέον να επικοινωνήσει με τον αισθητήρα DS18B20 και να εκδώσει μια από τις εντολές λειτουργίας του. Η πρώτη εντολή της Master συσκευής προς τον αισθητήρα είναι αυτή της μετατροπής της τάσης του αισθητήρα σε θερμοκρασία με την εντολή **Convert T[44h]**. Στη συνέχεια γίνεται ανάγνωση των περιεχομένων της μνήμης με την εντολή **Read Scratchpad [Beh]**. Με την εκτέλεση της εντολής **Convert T[44h]** τα θερμικά αποτελέσματα αποθηκεύονται στη μνήμη ROM και στη συνέχεια ο αισθητήρας εισέρχεται σε κατάσταση αναμονής. Με την **Read Scratchpad [Beh]** η Master συσκευή “διαβάζει” τη μνήμη ROM του αισθητήρα ώστε να λάβει την θερμοκρασία που έχει αποθηκευτεί σε αυτή. [54],[55],[56],[57],[58],[59],[60],[61],[62]

### 3.3 Αισθητήρας Θερμοκρασίας & Υγρασίας DHT22

#### 3.3.1 Γενικές πληροφορίες

Για την υλοποίηση της παρούσας εργασίας ήταν αναγκαία η καταγραφή της θερμοκρασίας και της υγρασίας του περιβάλλοντος χώρου. Μια από τις καλύτερες λύσεις για το σκοπό αυτό είναι ο αισθητήρας DHT22. Μαζί με τον αισθητήρα DHT11 αποτελούν τους πιο συχνά χρησιμοποιούμενους αισθητήρες για την επιτήρηση των συνθηκών χώρου λόγω του πολύ χαμηλού κόστους τους, της ευκολίας που έχουν στην διασύνδεσή τους με μικροελεγκτές και των ακριβέστατων αποτελεσμάτων που παρέχουν. Ένα πολύ σημαντικό χαρακτηριστικό αυτών των αισθητήρων είναι ότι οι



Εικόνα 57. Αισθητήρες Θερμοκρασίας - Υγρασίας, DHT11 & DHT22

μετρήσεις που παρέχουν έχουν ακρίβεια ενός δεκαδικού ψηφίου. Επίσης αυτοί οι αισθητήρες βαθμονομούνται εκ των προτέρων και δεν απαιτούν κάποιο επιπλέον εξάρτημα έτσι ώστε να ξεκινήσουν την μέτρηση της θερμοκρασίας και της σχετικής υγρασίας. Η σχετική υγρασία ορίζεται ως η ποσότητα νερού στον αέρα σε σχέση με την ποσότητα νερού που μπορεί να διατηρήσει ο αέρας σε μια συγκεκριμένη θερμοκρασία πολλαπλασιασμένη επί 100. Το μόνο μειονέκτημα που έχουν είναι ότι παίρνουν μέτρηση κάθε 1 ή 2 δευτερόλεπτα.

### 3.3.2 Σύγκριση DHT11 με DHT22

Όπως φαίνεται και στην εικόνα () οι δυο τύποι αισθητήρων DHTxx μοιάζουν αρκετά μεταξύ τους, τόσο εξωτερικά όσο και στη διαμόρφωση των Pin τους. Παρόλα αυτά έχουν κάποιες διαφοροποιήσεις. Αρχικά ο DHT22 είναι πιο ακριβός και ως επακόλουθο έχει καλύτερες προδιαγραφές. Πιο συγκεκριμένα το εύρος μέτρησης της θερμοκρασίας του κυμαίνεται από τους  $-40^{\circ}\text{C}$  στους  $+125^{\circ}\text{C}$  με ακρίβεια  $\pm 0.5$  βαθμούς, ενώ το εύρος μέτρησης της θερμοκρασίας του DHT11 κυμαίνεται από τους  $0^{\circ}\text{C}$  έως τους  $50^{\circ}\text{C}$  με ακρίβεια  $\pm 2$  βαθμούς. Επίσης ο DHT22 έχει καλύτερο εύρος μέτρησης της υγρασίας που κυμαίνεται από 0% μέχρι 100% με 2-5% ακρίβεια, ενώ το εύρος υγρασίας του DHT11 κυμαίνεται από 20 μέχρι 80% με 5% ακρίβεια. Στον πίνακα παρουσιάζονται συνοπτικά τα χαρακτηριστικά τους.

	<b>DHT11</b>	<b>DHT22</b>
Τάση Λειτουργίας	3 - 5V	3 - 5V
Μέγιστο Ρεύμα Λειτουργίας	2.5mA max	2.5mA max
Εύρος Υγρασίας	20 - 80% / 5%	0 - 100% / 2-5%
Εύρος Θερμοκρασίας	0 - 50°C / $\pm 2^{\circ}\text{C}$	-40 - 80°C / $\pm 0.5^{\circ}\text{C}$
Ρυθμός Δειγματοληψίας	1 Hz (κάθε δευτερόλεπτο)	0.5 Hz (κάθε 2 δευτερόλεπτα)
Μέγεθος	15.5mm x 12mm 5.5mm	15.1mm x 25mm x 7.7mm
Πλεονέκτημα	Πολύ χαμηλό κόστος	Μεγαλύτερη Ακρίβεια

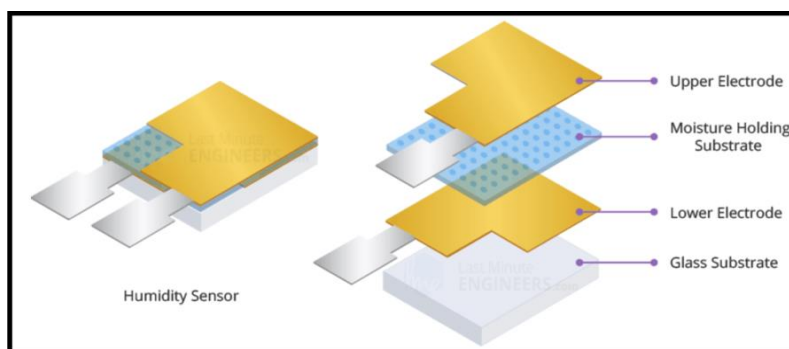
Εικόνα 58. Πίνακας Χαρακτηριστικών DHT11 & DHT22 [63]

### 3.3.3 Επισκόπηση υλισμικού

Στο εσωτερικό του κάθε αισθητήρα υπάρχει ένα αισθητήριο για την μέτρηση της υγρασίας μαζί με έναν αισθητήρα θερμοκρασίας NTC ή θερμίστορ. Ο αισθητήρας υγρασίας αποτελείται από δυο ηλεκτρόδια και ανάμεσά τους βρίσκεται ένα

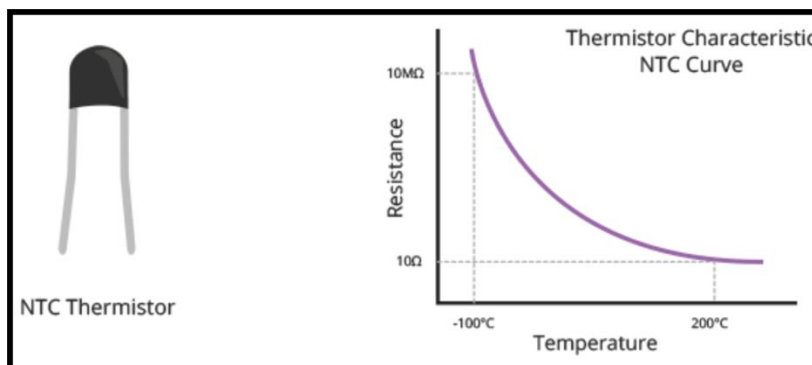


υπόστρωμα κατακράτησης υγρασίας (συνήθως κατασκευασμένος από κάποιο άλας ή κάποια αγώγιμα πλαστικά πολυμερή). Το υπόστρωμα απελευθερώνει ιόντα καθώς απορροφάει υδρατμούς, έχοντας ως αποτέλεσμα την αύξηση της αγωγιμότητας μεταξύ των ηλεκτροδίων. Η αλλαγή στην αντίσταση μεταξύ των δυο ηλεκτροδίων είναι ανάλογη της σχετικής υγρασίας. Όσο αυξάνεται η σχετική υγρασία τόσο μειώνεται η αντίσταση μεταξύ των ηλεκτροδίων, ενώ όσο μικραίνει η σχετική υγρασία τόσο αυξάνεται η αντίσταση μεταξύ των ηλεκτροδίων.



Εικόνα 59. Εσωτερική Δομή του αισθητηρίου υγρασίας [63]

Επιπλέον αποτελούνται από έναν αισθητήρα θερμοκρασίας NTC / Θερμίστορ για την μέτρηση της θερμοκρασίας. Το θερμίστορ είναι μια θερμική αντίσταση, δηλαδή μια αντίσταση που μεταβάλλει την αντίστασή της με την θερμοκρασία. Τα θερμίστορ κατασκευάζονται με τέτοιο τρόπο ώστε η αντίστασή τους να μεταβάλλεται δραστικά με την θερμοκρασία. Ο όρος NTC προέρχεται από το “Negative Temperature Coefficient” και σημαίνει “Αρνητικός Συντελεστής θερμοκρασίας”, δηλαδή ότι η αντίσταση μειώνεται με αύξηση της θερμοκρασίας.



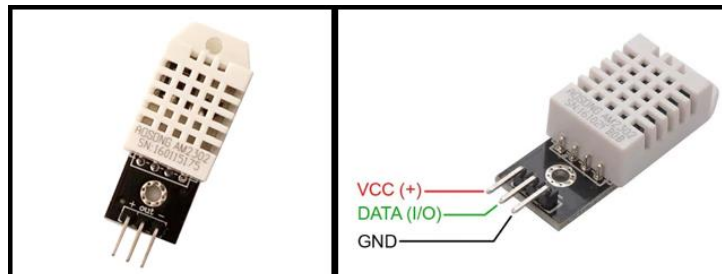
Εικόνα 60. NTC Θερμίστορ και χαρακτηριστική καμπύλη [63]

### 3.3.4 Μορφολογία Ακροδεκτών

Η συνδεσμολογία και των δυο αισθητήρων με τον μικροελεγκτή είναι ίδια. Αποτελούνται από 3 Pin:



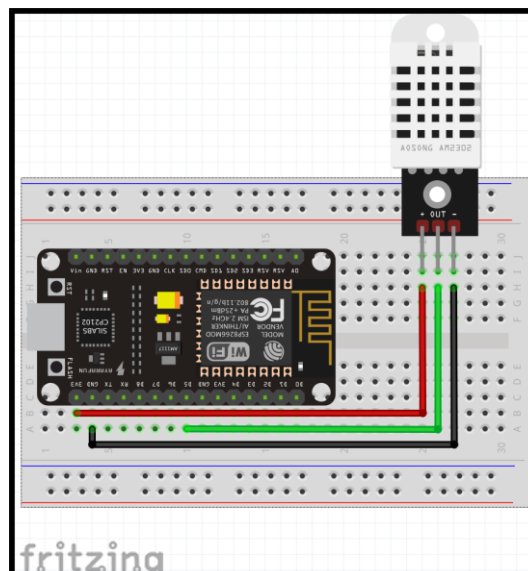
- **Vcc** Το Pin Vcc παρέχει τροφοδοσία στον αισθητήρα. Η τάση τροφοδοσίας κυμαίνεται από 3.3V μέχρι 5.5V.
- **Data** Το Pin Data χρησιμοποιείται για την επικοινωνία ανάμεσα στον αισθητήρα και στον μικροελεγκτή.
- **GND** Το **GND** πρέπει να συνδέεται με την γείωση του μικροελεγκτή.



Εικόνα 61. Ακροδέκτες αισθητήρα DHT22

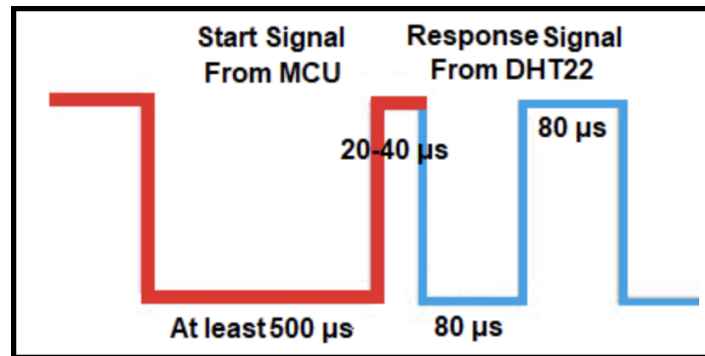
### 3.3.5 Συνδεσμολογία με τον μικροελεγκτή

Όπως αναφέρθηκε και προηγουμένως, για την εργασία χρησιμοποιήθηκε το ESP8266. Η συνδεσμολογία του αισθητήρα απεικονίζεται στην παρακάτω εικόνα. Το (-) του αισθητήρα είναι συνδεδεμένο με την γείωση, το (+) είναι συνδεδεμένο με την τροφοδοσία 3.3V και το **(OUT)** με το ψηφιακό Pin I/O **D5 (GPIO14)**. Να σημειωθεί ότι η module έκδοση του αισθητήρα που χρησιμοποιήθηκε για την εργασία, συμπεριλαμβάνει μια pull-up αντίσταση των 10kΩ μεταξύ του **OUT** και του **Vcc** έτσι ώστε να διατηρείται σταθερή η ροή των αποτελεσμάτων.



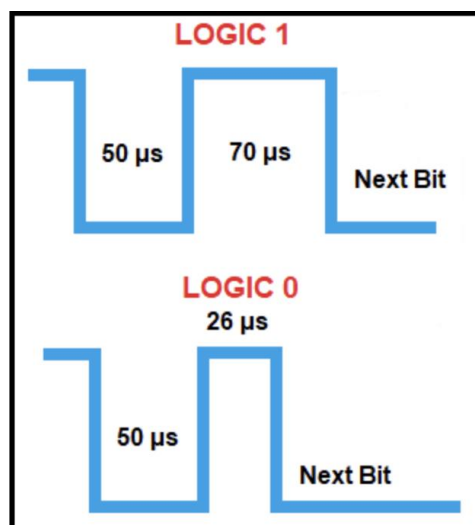
Εικόνα 62. Συνδεσμολογία DHT22 με το ESP8266 [63]

Για να ξεκινήσει η επικοινωνία με τον μικροελεγκτή έτσι ώστε να σταλούν τα δεδομένα θερμοκρασίας και υγρασίας, ο μικροελεγκτής πρέπει να ξεκινήσει το σήμα έναρξης, το οποίο είναι το λογικό "0" (χαμηλή κατάσταση) για τουλάχιστον 500 microseconds ακολουθούμενο από λογικό "1" (υψηλή κατάσταση). Ο DHT22 πρέπει τότε να απαντήσει ρίχνοντας το δίαυλο σε χαμηλή κατάσταση για 80 microseconds και επαναφέροντάς τον σε υψηλή κατάσταση για άλλα 80 microseconds πριν τον ρίξει ξανά σε χαμηλή κατάσταση.



Εικόνα 63. Διάγραμμα Παλμού Εκκίνησης DHT22 [64]

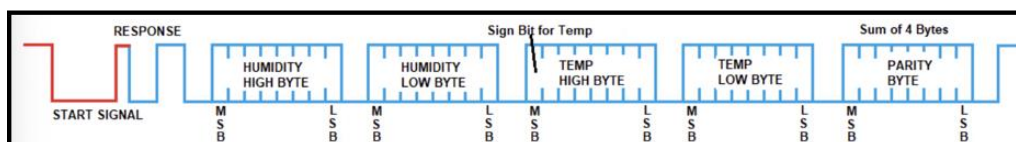
Ο DHT22 θα στείλει πρώτα τις τιμές υγρασίας ακολουθούμενες από τις τιμές θερμοκρασίας. Και οι δυο τιμές είναι 16-bit αλλά στέλνονται ανά 8-bit. Το πιο σημαντικό bit (MSB) του υψηλού byte στέλνεται πρώτο. Το MSB του υψηλού byte της θερμοκρασίας είναι το bit που δείχνει το πρόσημο. Εάν είναι "1" η θερμοκρασία είναι αρνητική αλλιώς είναι θετική. Το λογικό "1" είναι ένας χαμηλός παλμός διάρκειας 50 microseconds ακολουθούμενος από έναν υψηλό παλμό 70 microseconds. Το λογικό "0" είναι ένας χαμηλός παλμός διάρκειας 50 microseconds ακολουθούμενος από έναν υψηλό παλμό 26 microseconds.



Εικόνα 64. Απεικόνιση Παλμού με Λογικό 1 & Λογικό 0 [64]

Αφού σταλθούν τα δυο byte για την υγρασία και δυο byte για την θερμοκρασία, ο αισθητήρας σταματάει την μετάδοση με ένα byte “ισότητας”. Αυτό το byte ισούται με το άθροισμα των τεσσάρων byte. [63],[64],[65],[66],[67],[68],[69],[70]

$$\begin{aligned} \text{byte ισοτιμίας} &= \text{υψηλό byte υγρασίας} + \text{χαμηλό byte υγρασίας} + \\ &= \text{υψηλό byte θερμοκρασίας} + \text{χαμηλό byte θερμοκρασίας} \end{aligned}$$

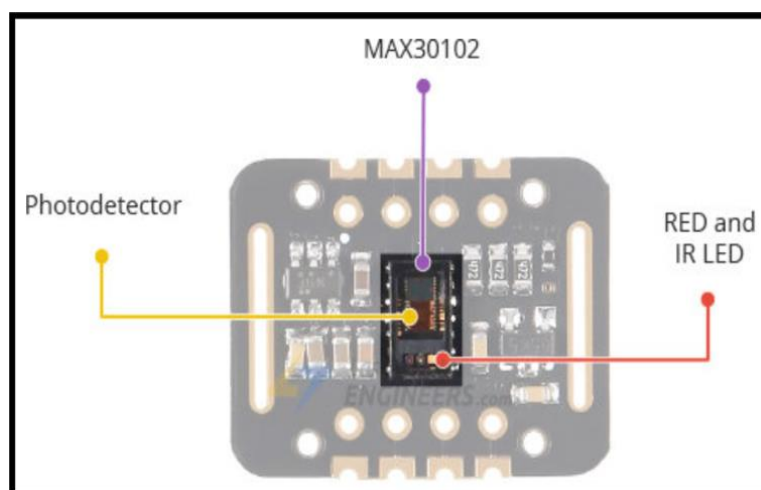


Εικόνα 65. Παράδειγμα Παλμού όταν ο DHT22 κάνει μια καινούρια μέτρηση Θερμοκρασίας & Υγρασίας [64]

### 3.4 Αισθητήρας SpO2 & Heart Rate MAX30102

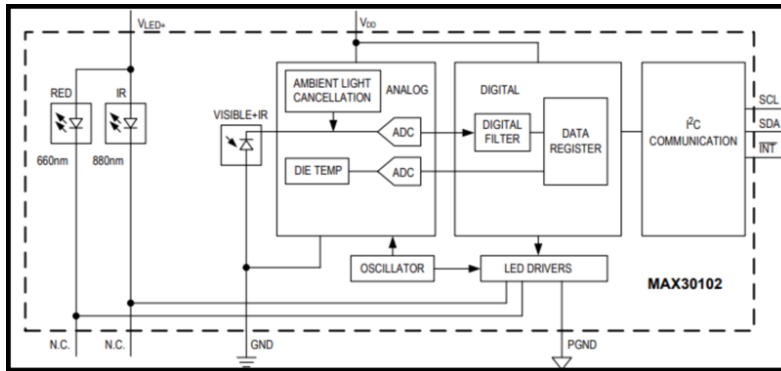
#### 3.4.1 Γενική επισκόπηση αισθητήρα

Ο MAX30102 αισθητήρας οξυμετρίας και μέτρησης καρδιακών παλμών είναι ένας I2C, χαμηλής κατανάλωσης βιομετρικός αισθητήρας. Περιλαμβάνει δυο LED, ένα φωτοανιχνευτή και επεξεργαστή αναλογικού σήματος χαμηλού θορύβου για την ανίχνευση του σήματος οξυμετρίας και του καρδιακού παλμού.



Εικόνα 66. Βασικά τμήματα του MAX30102 [71]

Από τη μια μεριά του τζαμιού, ο MAX30102 έχει δυο LED, ένα που εκπέμπει στο κόκκινο και ένα στο υπέρυθρο. Από την άλλη μεριά υπάρχει ένας πολύ ευαίσθητος φωτοανιχνευτής. Ο αισθητήρας ενεργοποιεί ένα LED κάθε φορά και ο ανιχνευτής καταγράφει το ποσό φωτός που επιστρέφει πίσω. Με βάση την αποτύπωση του φωτός γίνεται η μέτρηση του επιπέδου οξυγόνου στο αίμα και ο καρδιακός παλμός.

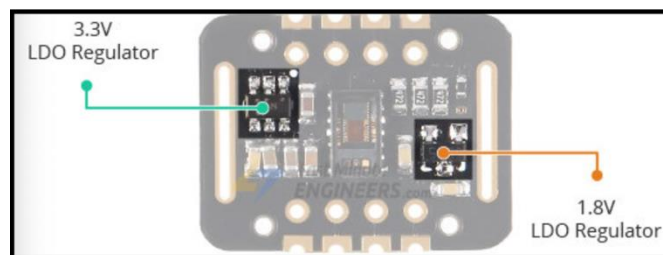


Εικόνα 67. Λειτουργικό μπλοκ διάγραμμα του MAX30102 [72]

### 3.4.2 Τεχνικά χαρακτηριστικά

- **Τροφοδοσία & κατανάλωση ισχύος**

Ο MAX30102 χρειάζεται δυο διαφορετικές τάσεις τροφοδοσίας, μια 1.8V για το ολοκληρωμένο και μια 3.3V για το κόκκινο και το υπέρυθρο LED. Έτσι ο αισθητήρας αποτελείται από ρυθμιστές τάσης 3.3V & 1.8V. Ένα από τα πιο σημαντικά χαρακτηριστικά του MAX30102 είναι η χαμηλή κατανάλωση ισχύος. Καταναλώνει λιγότερα από 600μΑ κατά τη διάρκεια της μέτρησης, ενώ γίνεται να εισέλθει και σε κατάσταση αναμονής όπου καταναλώνει μόνο 0.7μΑ. Η τόσο χαμηλή κατανάλωση ισχύος του επιτρέπει την ενσωμάτωσή του σε συσκευές που τροφοδοτούνται από μπαταρία όπως έξυπνα ρολόγια ή άλλες φορητές συσκευές.



Εικόνα 68. Ρυθμιστές τάσης 3.3V & 1.8V [71]

- **Διεπαφή I2C**

Ο αισθητήρας χρησιμοποιεί απλό πρωτόκολλο διεπαφής I2C δυο καλωδίων για επικοινωνία με τον μικροελεγκτή. Έχει μια καθορισμένη I2C διεύθυνση 0xAE<sub>HEX</sub> για λειτουργία γραψίματος και 0xAF<sub>HEX</sub> για ανάγνωση.

- **FIFO (First In First Out) Buffer**

Ο MAX30102 ενσωματώνει έναν FIFO Buffer για την αποθήκευση δειγμάτων δεδομένων. Ο FIFO έχει τράπεζα μνήμης 32 δειγμάτων που σημαίνει ότι μπορεί διατηρήσει 32 δείγματα οξυμετρίας και καρδιακού ρυθμού. Ο FIFO Buffer μπορεί να

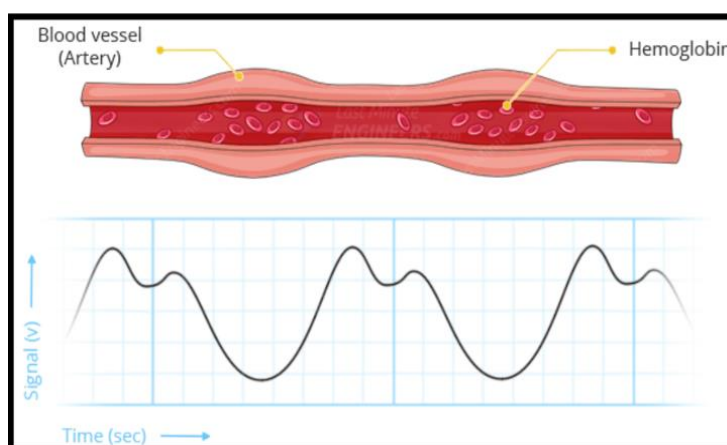
απαλλάξει τον μικροελεγκτή από το να διαβάζει το κάθε νέο δεδομένο από τον αισθητήρα εξοικονομώντας έτσι ενέργεια στο συνολικό σύστημα.

### 3.4.3 Τρόπος λειτουργίας MAX30102

Όπως κάθε οπτικό οξύμετρο και αισθητήρας καρδιακών παλμών, έτσι και ο MAX30102 αποτελείται από ένα ζεύγος LED υψηλής έντασης και ένα φωτοανιχνευτή. Το μήκος κύματος για το κόκκινο LED είναι στα 660nm ενώ για το υπέρυθρο στα 880nm. Ο αισθητήρας λειτουργεί εκπέμποντας φως με τα δυο LED στο δάχτυλο και μετρώντας το ανακλώμενο φως χρησιμοποιώντας τον φωτοανιχνευτή.

- **Μέτρηση καρδιακών παλμών**

Η πλούσια σε οξυγόνο αιμοσφαιρίνη στο αρτηριακό αίμα έχει το χαρακτηριστικό να απορροφάει υπέρυθρο φως. Όσο υψηλότερη η συγκέντρωση αιμοσφαιρίνης τόσο περισσότερο απορροφάει το υπέρυθρο φως. Καθώς το αίμα περνάει από το δάχτυλο με κάθε παλμό, η ποσότητα του ανακλώμενου φωτός αλλάζει, δημιουργώντας μια μεταβαλλόμενη κυματομορφή στην έξοδο του ανιχνευτή. Όσο συνεχίζεται αυτή η διαδικασία σχηματίζεται η μέτρηση του καρδιακού παλμού.

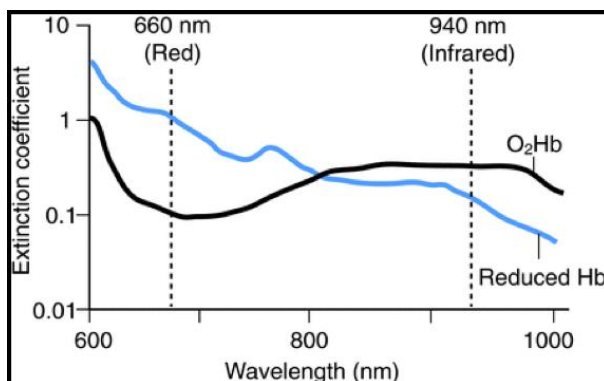


Εικόνα 69. Γραφική απεικόνιση μέτρησης καρδιακών παλμών [71]

- **Οξυμετρία**

Η οξυμετρία βασίζεται στην βασική αρχή ότι η ποσότητα κόκκινου και υπέρυθρου φωτός που απορροφάται διαφέρει ανάλογα με την ποσότητα οξυγόνου στο αίμα.

Όπως φαίνεται και από το γράφημα, το μη οξυγονωμένο αίμα απορροφά περισσότερο κόκκινο φως (660nm), ενώ το οξυγονωμένο αίμα απορροφά περισσότερο υπέρυθρο φως (880nm). Με τον υπολογισμό της αναλογίας του υπέρυθρου και του κόκκινου φωτός που λήφθηκε από τον ανιχνευτή, υπολογίζεται το επίπεδο οξυγόνου στο αίμα.



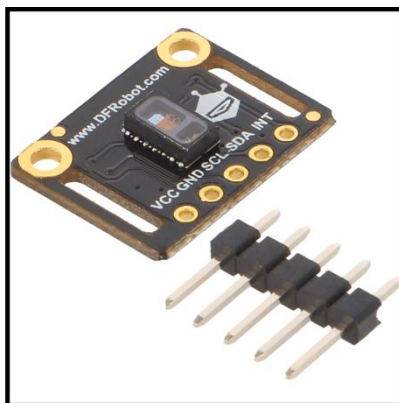
Εικόνα 70. Γράφημα απορρόφησης κόκκινου & υπέρυθρου φωτός στο αίμα

#### 3.4.4 Ακροδέκτες & συνδεσμολογία

Ο αισθητήρας MAX30102 αποτελείται από τους εξής ακροδέκτες:

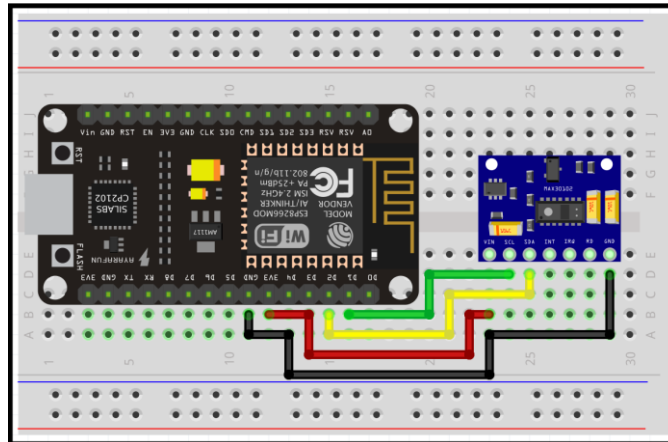
- **Vcc** Είναι το Pin τροφοδοσίας και μπορεί να συνδεθεί στα 3.3 ή 5V.
- **GND** Είναι το Pin της γείωσης.
- **SCL** Προέρχεται από το “Serial Clock” και είναι το Pin χρονισμού και συνδέεται την γραμμή χρονισμού του μικροελεγκτή.
- **SDA** Προέρχεται από το “Serial Data” και είναι το Pin αποστολή δεδομένων και συνδέεται με την γραμμή δεδομένων του μικροελεγκτή.

Η συνδεσμολογία του αισθητήρα MAX30102 είναι λίγο πιο περίπλοκη, αφού στην προκειμένη περίπτωση χρειάστηκε να γίνουν κάποιες επιπλέον διαδικασίες. Πιο συγκεκριμένα χρειάστηκε να γίνει χρήση του κολλητηρίου και καλά. Κατά την αγορά του, ο αισθητήρας έρχεται στη μορφή που φαίνεται στην εικόνα.



Εικόνα 71. Αισθητήρας MAX30102 DFrobot

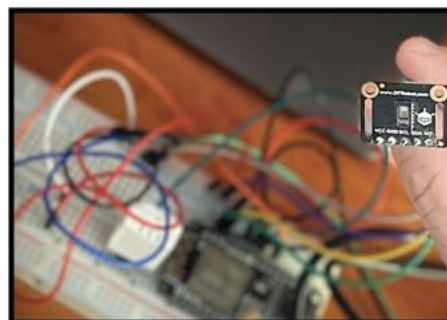
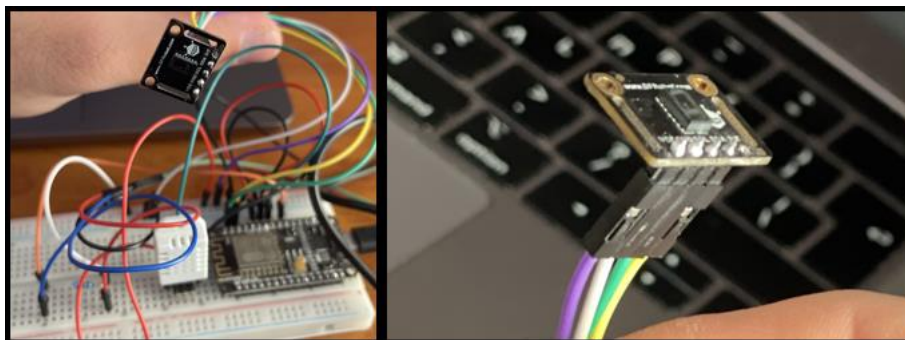
Έτσι για να μπορέσει να λειτουργήσει σωστά ο αισθητήρας και να έχει σταθερή απόδοση θα πρέπει να γίνει κόλληση των Pin με την πλακέτα του αισθητήρα όπως φαίνεται και στις παρακάτω εικόνες.



Εικόνα 72. Σύνδεση MAX30102 με το ESP8266 [71]

\*Να σημειωθεί ότι για τη συγκεκριμένη εργασία δεν απαιτείται η χρήση του Pin “INT” του αισθητήρα έτσι δεν έχει συνδεθεί.

Έτσι αφού γίνουν οι απαραίτητες διεργασίες, η τελική σύνδεση του αισθητήρα με τον μικροελεγκτή παρουσιάζεται στην εικόνα 76. [71],[72],[73],[74],[75]



Εικόνα 73. Συνδεσμολογία MAX30102



# 4. Λογισμικό

Σε αυτό το κεφάλαιο θα παρουσιαστούν τα πιο βασικά κομμάτια του κώδικα με επεξηγήσεις καθώς και η εισαγωγή των βιβλιοθηκών στα αντίστοιχα προγράμματα. Για τη συγκεκριμένη διπλωματική υπάρχουν 3 κύρια κομμάτια κώδικα:

- Το πρώτο είναι αυτό για τον προγραμματισμό του μικροελεγκτή ESP8266. Πιο συγκεκριμένα, υπάρχει δυνατότητα προγραμματισμού του μικροελεγκτή με τη χρήση του Arduino IDE.
- Το επόμενο κομμάτι κώδικα είναι αυτό για την δημιουργία της βάσης δεδομένων και την αποθήκευση των δεδομένων από τους αισθητήρες που χρησιμοποιήσα.
- Το τρίτο κομμάτι κώδικα γράφτηκε σε γλώσσα προγραμματισμού PHP. Ο κώδικας που γράφτηκε σε PHP είχε ως σκοπό την σύνδεση του Arduino IDE με τη βάση δεδομένων ώστε να μπορέσουν να αποθηκευτούν τα δεδομένα, αλλά και η δημιουργία μιας ιστοσελίδας όπου γίνεται δομημένη απεικόνιση των δεδομένων. Επίσης η ιστοσελίδα δίνει τη δυνατότητα φιλτραρίσματος των αποτελεσμάτων που αποθηκεύτηκαν για πιο αποτελεσματική αναζήτηση.

Τέλος, μια επιπλέον δυνατότητα στην παρούσα εργασία είναι η απεικόνιση όλων των μετρήσεων με γραφικό τρόπο, σε εφαρμογή του Smartphone. Για τον σκοπό αυτό χρησιμοποιήθηκε η εφαρμογή Blynk η οποία προσφέρει μια πληθώρα από εργαλεία για την δημιουργία ενός φιλικού προς το χρήστη περιβάλλοντος.

## 4.1 Προγραμματισμός στο Arduino IDE

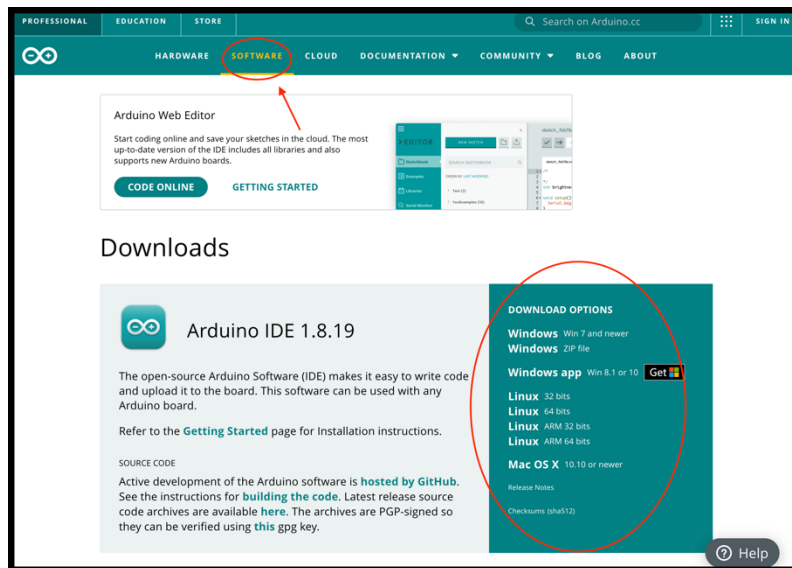
### 4.1.1 Εγκατάσταση του Arduino IDE

Το πρώτο κομμάτι της εργασίας ξεκινάει με τον εγκατάσταση του περιβάλλοντος προγραμματισμού του Arduino καθώς και των απαιτούμενων βιβλιοθηκών. Η διαδικασία είναι η εξής:

- Αρχικά από την ιστοσελίδα του Arduino (<https://www.arduino.cc/>), επιλέγουμε την ενότητα **Software**.
- Εκεί υπάρχουν όλες οι διαθέσιμες εκδόσεις του Arduino IDE ανάλογα με το λειτουργικό σύστημα που τρέχουμε στον υπολογιστή μας. Για τη

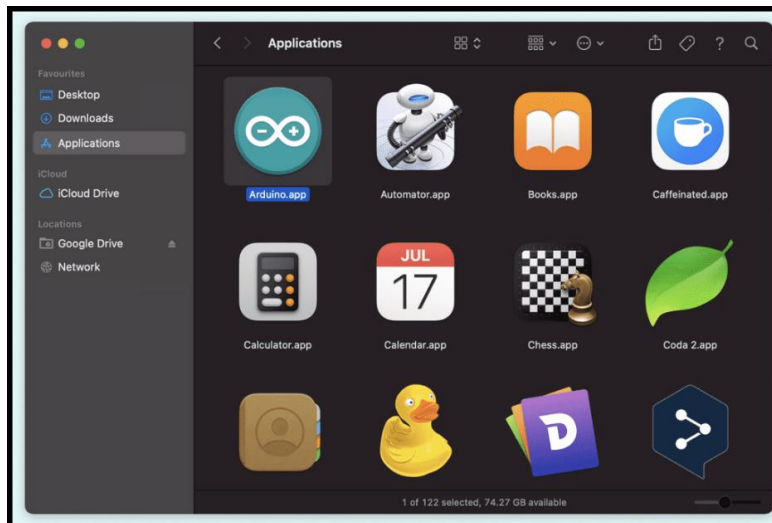


συγκεκριμένη εργασία χρησιμοποιήθηκε το λειτουργικό σύστημα Mac OS Big Sur.



Εικόνα 74. Arduino Software Download Page [76]

- Επιλέγουμε την έκδοση που μας ενδιαφέρει και κατεβαίνει στον υπολογιστή ένα αρχείο σε μορφή .zip. Αφού γίνει εξαγωγή του αρχείου από το φάκελο .zip, μεταφέρουμε το εικονίδιο στον φάκελο με τις εφαρμογές ώστε να γίνει εγκατάσταση. Τέλος ανοίγουμε την εφαρμογή.

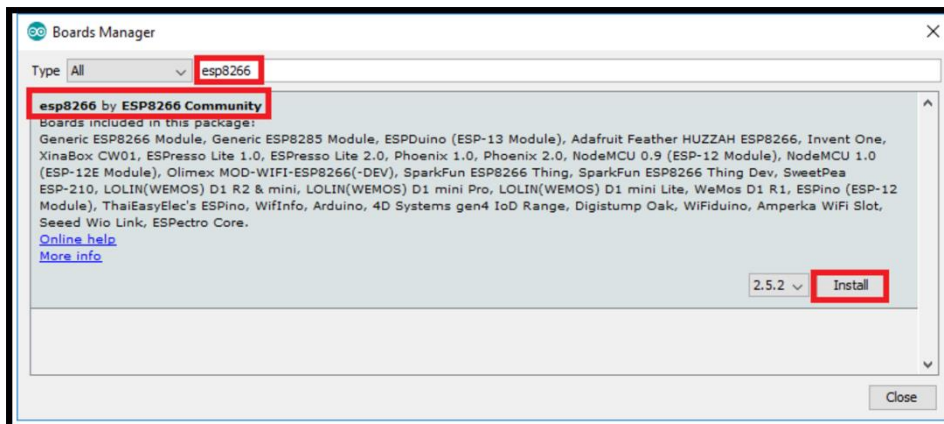


Εικόνα 75. Application Folder Mac OS

Για να μπορέσει να προγραμματιστεί το ESP8266 με την χρήση του Arduino IDE απαιτούνται κάποιες τελευταίες ρυθμίσεις.

- Στο Arduino IDE πηγαίνουμε στο **Preferences**. Στο πεδίο **“Additional Boards Manager URLs”** κάνουμε επικόλληση το εξής: [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) και πατάμε **“OK”**.

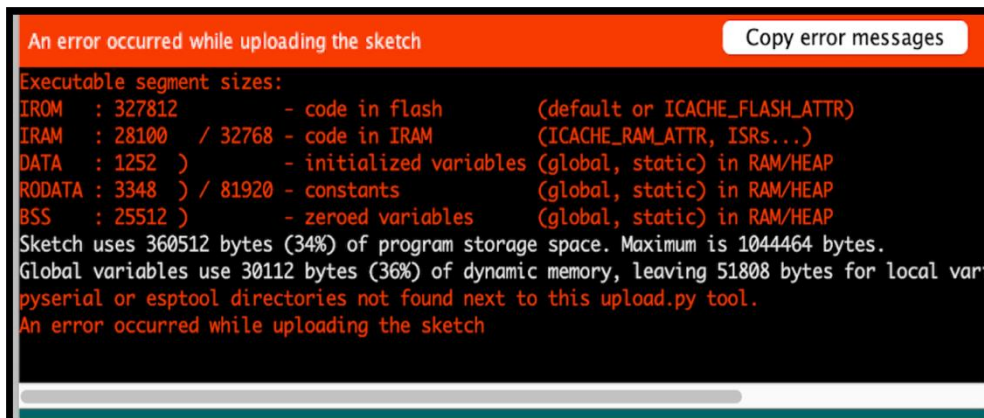
- Στην συνέχεια πηγαίνουμε στο path **Tools > Board > Boards Manager**. Στο πεδίο “Search” αναζητούμε το “**esp8266**” όπως φαίνεται στην εικόνα() και πατάμε “**install**”.



Εικόνα 76. Arduino Boards Manager

- Αφού ολοκληρώσουμε τη διαδικασία πρέπει να επιλέξουμε την πλακέτα NodeMCU. Από το path **Tools > Board**, επιλέγουμε το NodeMCU 1.0 (ESP-12E Module).
- Τέλος από το path **Tools > Port** επιλέγουμε την θύρα του υπολογιστή μας.

Παρόλο που έγιναν όλα τα προηγούμενα βήματα για την εγκατάσταση του ESP8266, κατά το ανέβασμα του κώδικα προέκυψε το παρακάτω πρόβλημα.



Εικόνα 77. Σφάλμα κατά το ανέβασμα του κώδικα στο ESP8266

Ύστερα από τη διερεύνηση του σφάλματος προέκυψε ότι το πρόβλημα σχετίζεται με το λογισμικό MacOS Big Sur της Apple. Το πρόβλημα επιλύθηκε με τον παρακάτω τρόπο:

- Από τον Finder επιλέγουμε **Μετάβαση > Μετάβαση σε φάκελο** και κάνουμε επικόλληση το εξής:

`/Library/Arduino15/packages/esp8266/hardware/esp8266/2.7.4/tools/pyserial/serial/tools/list_ports_osx.py`

- Ανοίγουμε το αρχείο με ένα απλό πρόγραμμα όπως το Text Editor.
- Βρίσκουμε τις εξής γραμμές κώδικα:

**iokit = ctypes.cdll.LoadLibrary(ctypes.util.find\_library('IOKit'))**

**cf = ctypes.cdll.LoadLibrary(ctypes.util.find\_library('CoreFoundation'))**

και χρησιμοποιούμε το σύμβολο (#) μπροστά από τις γραμμές κώδικα έτσι ώστε να μετατραπούν σε σχόλια.

Τέλος κάτω από αυτές τις γραμμές κώδικα κάνουμε επικόλληση τον εξής κώδικα:

**iokit=ctypes.cdll.LoadLibrary('/System/Library/Frameworks/IOKit.framework/IOKit')**

**cf=ctypes.cdll.LoadLibrary('/System/Library/Frameworks/CoreFoundation.framework/CoreFoundation')**

Ο τελικός κώδικας πρέπει να φαίνεται όπως στην φωτογραφία.

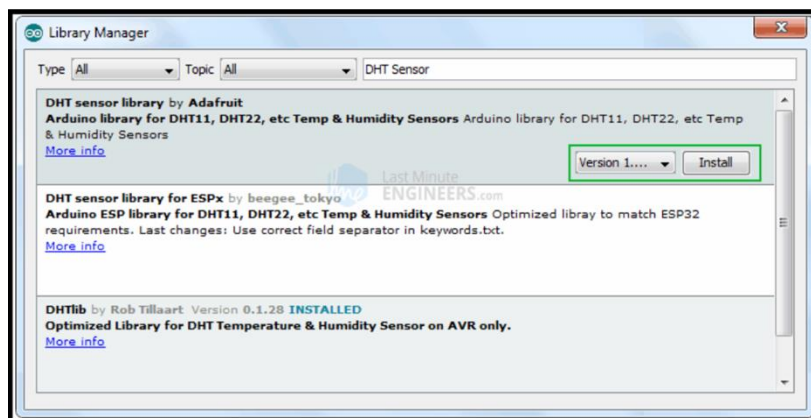
```
#iokit = ctypes.cdll.LoadLibrary(ctypes.util.find_library('IOKit'))
#cf = ctypes.cdll.LoadLibrary(ctypes.util.find_library('CoreFoundation'))
iokit = ctypes.cdll.LoadLibrary('/System/Library/Frameworks/IOKit.framework/IOKit')
cf = ctypes.cdll.LoadLibrary('/System/Library/Frameworks/CoreFoundation.framework/CoreFoundation')
```

Εικόνα 78. Επεξεργασμένος Κώδικας

#### 4.1.2 Εγκατάσταση βιβλιοθήκης DHT22

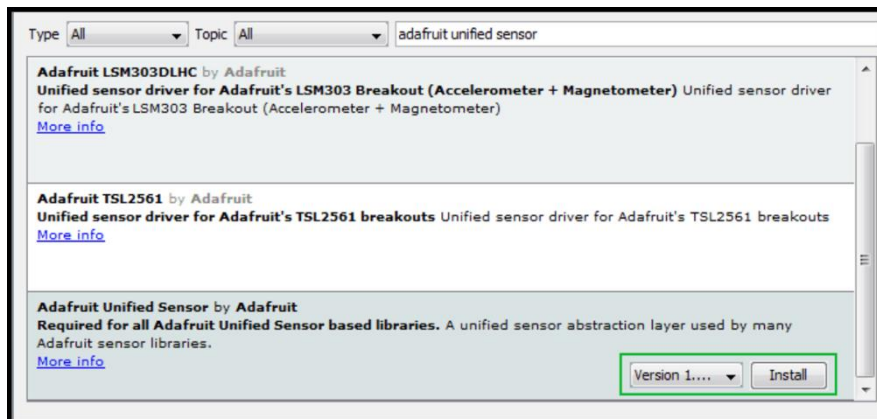
Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο, για την υλοποίηση της εργασίας θα χρησιμοποιηθεί ο αισθητήρας θερμοκρασίας και υγρασίας **DHT22**. Η διαδικασία για την εγκατάσταση της βιβλιοθήκης του αισθητήρα είναι η εξής:

- Από το Arduino IDE πηγαίνουμε στο path **Sketch > Include Library > Manage Libraries...**
- Στο παράθυρο που εμφανίζεται πληκτρολογούμε στο πεδίο αναζήτησης τον όρο **“DHT Sensor”**. Θα εμφανιστούν αρκετά αποτελέσματα αλλά μας ενδιαφέρει η βιβλιοθήκη που είναι φτιαγμένη από την **“Adafruit”**. Την εγκαθιστούμε.



Εικόνα 79. Εγκατάσταση βιβλιοθήκης DHT22

- Τέλος με τον ίδιο τρόπο, στο πεδίο αναζήτησης πληκτρολογούμε τον όρο “adafruit unified sensor”.

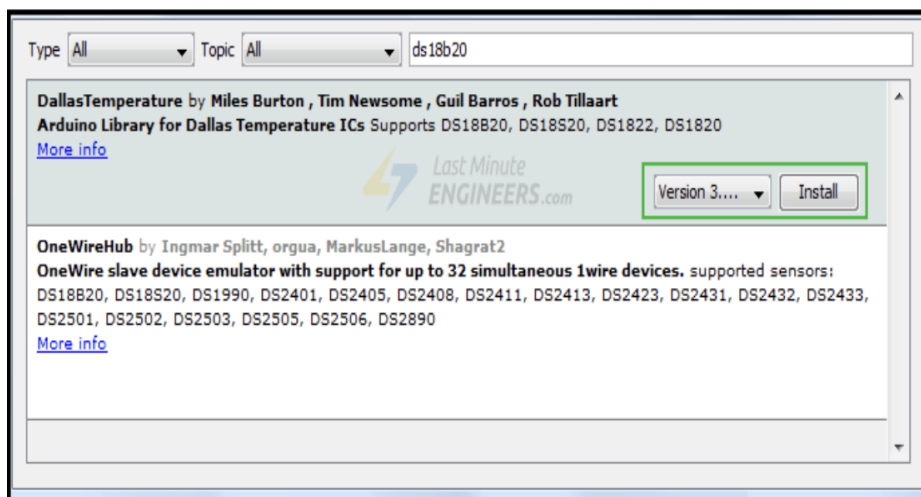


Εικόνα 80. Απαραίτητη βιβλιοθήκη για τη λειτουργία του DHT22

#### 4.1.3 Εγκατάσταση της βιβλιοθήκης για τον αισθητήρα DS18B20

Αντίστοιχα για τη χρήση του αισθητήρα DS18B20 πρέπει να εγκατασταθούν οι απαραίτητες βιβλιοθήκες.

- Από το Arduino IDE πηγαίνουμε στο **Sketch > Include Library > Manage Libraries...** . Στο πεδίο αναζήτησης πληκτρολογούμε “ds18b20” και κάνουμε εγκατάσταση της βιβλιοθήκης από την “DallasTemperature”.



Εικόνα 81. Εγκατάσταση Βιβλιοθήκης Αισθητήρα DS18B20

- Για να μπορέσει να λειτουργήσει η βιβλιοθήκη πρέπει να γίνει εγκατάσταση άλλης μια βιβλιοθήκης. Στο πεδίο αναζήτησης πληκτρολογούμε “onewire” και εγκαθιστούμε τη βιβλιοθήκη.



Εικόνα 82. Εγκατάσταση Βιβλιοθήκης One Wire

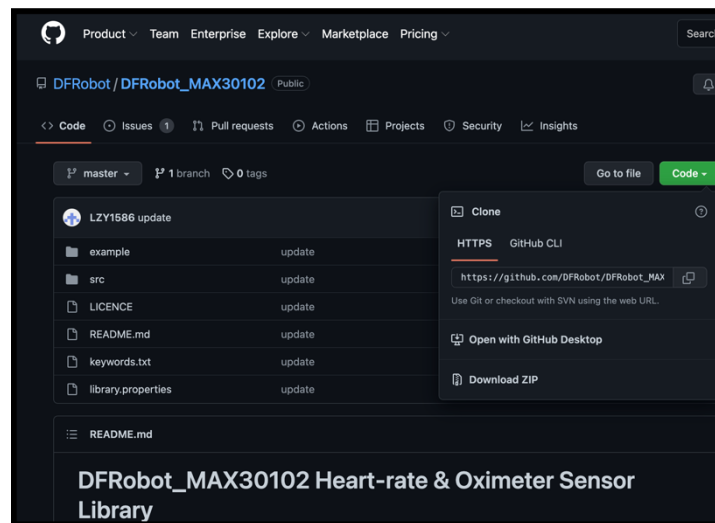
#### 4.1.4 Εγκατάστασης της βιβλιοθήκης για τον αισθητήρα MAX30102

Για την μέτρηση της οξυμετρίας και των καρδιακών παλμών χρησιμοποιήθηκε ο αισθητήρας **MAX30102** της **DFRobot**. Για την εγκατάστασή του απαιτούνται τα εξής βήματα.

- Επισκεπτόμαστε τη σελίδα του κατασκευαστή στη διεύθυνση

[https://github.com/DFRobot/DFRobot\\_MAX30102](https://github.com/DFRobot/DFRobot_MAX30102)

- Πατώντας το πράσινο πλήκτρο **“Code”**, επιλέγουμε το **“Download ZIP”**.



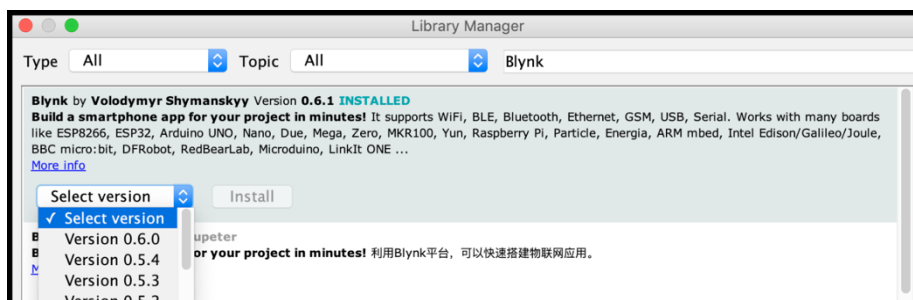
Εικόνα 83. Εγκατάσταση Βιβλιοθήκης MAX30102 DFRobot

- Στη συνέχεια κάνουμε εξαγωγή του .zip αρχείου, και αντιγράφουμε τον φάκελο που προκύπτει στο path που έχει δημιουργήσει το ESP8266 για τις βιβλιοθήκες του. Τέλος στο Arduino IDE πάμε στο path **Sketch > Include Library** και επιβεβαιώνουμε ότι η βιβλιοθήκη έχει εγκατασταθεί.

#### 4.1.5 Εγκατάσταση βιβλιοθήκης για την εφαρμογή Blynk

Ομοίως με τις ανωτέρω βιβλιοθήκες εγκαθιστούμε και την βιβλιοθήκη για την εφαρμογή Blynk.

- Στο path **Sketch > Include Library > Manage Libraries...** αναζητούμε τον όρο “blynk” και εγκαθιστούμε τη βιβλιοθήκη.

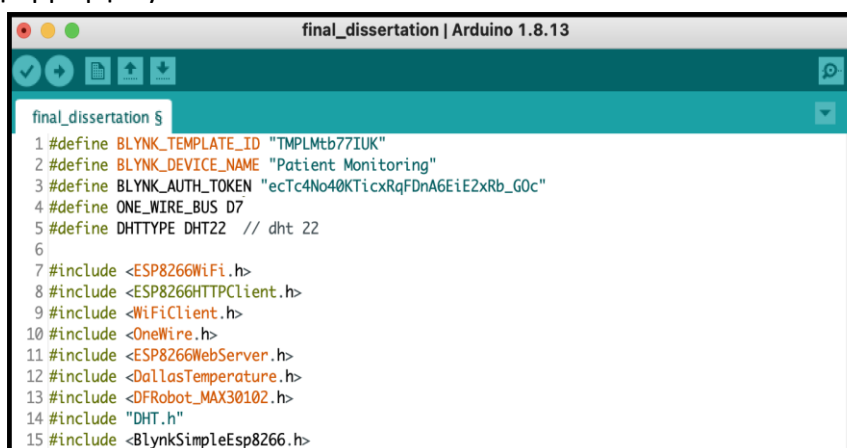


Εικόνα 84. Εγκατάσταση βιβλιοθήκης Blynk

#### 4.2 Επεξήγηση κώδικα Arduino IDE

Το κομμάτι κώδικα που θα επεξηγηθεί παρακάτω είναι υπεύθυνο για τον διαχείριση των αισθητήρων, την δημιουργία ενός Web Server για την real-time απεικόνιση των μετρήσεων των αισθητήρων, την επικοινωνία με την εφαρμογή Blynk στο Smartphone και την επικοινωνία με τη βάση δεδομένων για την αποθήκευση των δεδομένων.

- Στο παρακάτω τμήμα κώδικα στις γραμμές 1-5 ορίζονται οι τιμές των μεταβλητών που θα παραμείνουν σταθερές σε όλο τον κώδικα με τη χρήση της εντολής **#define**. Στις γραμμές 7-15 εισάγονται οι βιβλιοθήκες με την εντολή **#include**. Αυτές οι βιβλιοθήκες είναι αναγκαίες για την επικοινωνία του μικροελεγκτή με τους αισθητήρες, για την επικοινωνία του ESP8266 με το WiFi δίκτυο, τη δημιουργία του Web Server και την επικοινωνία με την εφαρμογή Blynk.



Εικόνα 85. Κώδικας Arduino



- Στη συνέχεια στις γραμμές **17-19**, εφόσον το ESP8266 θα συνδεθεί σε ένα υπάρχον δίκτυο WiFi, ορίζουμε το όνομα και τον κωδικό του δικτύου. Αυτές οι μεταβλητές ορίζονται σαν χαρακτήρες με τη χρήση της εντολής **“char”**.

```
17 // Your WiFi credentials.
18 char ssid[] = "test"; // Enter SSID here
19 char password[] = "12345678"; //Enter Password here
20
```

Εικόνα 86. Κώδικας Arduino

- Στις γραμμές 21-22 ο κωδικός εξουσιοδότησης της εφαρμογής Blynk γίνεται τύπου **“char”**. Στις γραμμές 24-29 ορίζονται οι τιμές για την επικοινωνία με τη βάση δεδομένων. Πιο συγκεκριμένα στη γραμμή **25** εισάγεται η IP διεύθυνση του υπολογιστή που χρησιμοποιούμε και το path για τον κώδικα PHP που επικοινωνεί με το Arduino IDE και συμβάλει στην αποθήκευση των δεδομένων από το **Βάση Δεδομένων**. Στη γραμμή **26** το **apikeyvalue** πρέπει να έχει ίδια τιμή με τον κώδικα στο αρχείο **PHP**.

```
24 //MySQL DataBase info
25 const char* serverName = "http://Your PC's IP/Sensor-data-test/post-sensor-data.php";
26 String apiKeyValue = "123";
27 // Example: tPmAT5Ab3j7F9
28 String sensorName = "DHT22";
29 String sensorLocation = "Hospital";
```

Εικόνα 87. Κώδικας Arduino

- Στις γραμμές **31-34, 36-37** ορίζονται αντικείμενα για τις αντίστοιχες κλάσεις. Με τις κλάσεις ορίζονται οι ιδιότητες και οι μέθοδοι των αντικειμένων. Στη γραμμή **31** το όρισμα στο αντικείμενο **server** είναι **80**. Αυτή είναι η προεπιλεγμένη θύρα για το **HTTP**, οπότε χρησιμοποιούμε αυτή. Στην γραμμή **34** οι ιδιότητες του αντικειμένου **oneWire** παίρνουν στο αντικείμενο που δημιουργήθηκε για την κλάση **DallasTemperature**. Στη γραμμή **35** ορίζεται ο ακροδέκτης που συνδέεται το Pin του αισθητήρα για την μεταφορά δεδομένων. Επίσης δημιουργείται αντικείμενο **dht** από την κλάση.

```
31 ESP8266WebServer server(80);
32 DFRobot_MAX30102 particleSensor;
33 OneWire oneWire(ONE_WIRE_BUS); // Setup a oneWire instance to communicate with any OneWire device
34 DallasTemperature sensors(&oneWire); // Pass oneWire reference to DallasTemperature library
35 uint8_t DHTPin = D5; // DHT Sensor
36 DHT dht(DHTPin, DHTTYPE); // Initialize DHT sensor.
37 BlynkTimer timer;
```

Εικόνα 88. Κώδικας Arduino

- Στις γραμμές **39-45** αρχικοποιούνται οι μεταβλητές που χρησιμοποιούνται στο πρόγραμμα.

```
39 int32_t SPO2; //SPO2
40 int8_t SPO2Valid; //Flag to display if SPO2 calculation is valid
41 int32_t heartRate; //Heart-rate
42 int8_t heartRateValid; //Flag to display if heart-rate calculation is valid
43
44 float Temperature, Humidity, tempC;
45 String val;
```

Εικόνα 89. Κώδικας Arduino

- Από τη γραμμή **47-96** εκτελείται η συνάρτηση **void setup()** . Αυτή η συνάρτηση εκτελείται μια φορά κατά την εκκίνηση του ESP8266 ή όταν γίνει επαναφορά με το κουμπί Reset. Στη γραμμή 49 γίνεται εκκίνηση του αισθητήρα θερμοκρασίας DS18B20. Στη γραμμή 51 ξεκινάει η σειριακή επικοινωνία στα 115200 bps (bits per second). Όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο η διακριτική ικανότητα του αισθητήρα DS18B20 μπορεί να καθοριστεί και προγραμματιστικά. Στην γραμμή 53 γίνεται αυτό ακριβώς. Στη γραμμή 57-60 γίνεται εκκίνηση του αισθητήρα οξυμετρίας MAX30102. Ο κώδικας έχει γραφτεί με τέτοιο τρόπο έτσι ώστε αν δεν υπάρξει επικοινωνία μεταξύ του μικροελεγκτή και του αισθητήρα να μην προχωρήσει το υπόλοιπο πρόγραμμα μέχρι να αποκατασταθεί η σύνδεση. Στη γραμμή 62-64 γίνεται η ρύθμιση των παραμέτρων για τον αισθητήρα οξυμετρίας-καρδιακών παλμών. Στη γραμμή 66 ορίζεται η κατάσταση του Pin του μικροελεγκτή έτσι ώστε να λειτουργεί σαν **είσοδος** για τις τιμές που στέλνει ο αισθητήρας θερμοκρασίας & υγρασίας DHT22. Στη γραμμή 70 ξεκινάει η επικοινωνία με τον DHT22. Στις γραμμές 72-76, 79-86 οι εντολές εκτυπώνουν στη σειριακή έξοδο ότι πραγματοποιείται σύνδεση στο δίκτυο WiFi. Αν η σύνδεση στο δίκτυο είναι επιτυχής, εκτυπώνεται η IP διεύθυνση του NodeMCU ESP8266.

```

47 void setup() {
48
49   sensors.begin();
50
51   Serial.begin(115200);
52
53   sensors.setResolution(9); /*Το αρχικο resolution του αισθητηρα είναι 12-bit δηλαδη ακριβεια 0.0625°C.
54   Αυτό καθυστερεί αρκετά την απεικόνιση της πραγματικής τιμής. Έτσι το αλλάζουμε σε 9-bit και ακριβεία 0.5 βαθμούς
55   ώστε να βλέπουμε πιο γρήγορα την πραγματική τιμή*/
56
57   while (!particleSensor.begin()) {
58     Serial.println("MAX30102 was not found");
59     delay(1000);
60   }
61
62   particleSensor.sensorConfiguration(/*ledBrightness=*/70, /*sampleAverage=*/SAMPLEAVG_16, \
63     /*ledMode=*/MODE_MULTILED, /*sampleRate=*/SAMPLERATE_400,
64     /*pulseWidth=*/PULSEWIDTH_118, /*adcRange=*/ADCRANGE_16384);
65
66   pinMode(DHTPin, INPUT);
67
68   //sensors.begin();
69
70   dht.begin();
71
72   Serial.println("Connecting to ");
73   Serial.println(ssid);
74
75   //connect to your local wi-fi network
76   WiFi.begin(ssid, password);
77   Blynk.config(auth);
78
79   //check wi-fi is connected to wi-fi network
80   while (WiFi.status() != WL_CONNECTED) {
81     delay(1000);
82     Serial.print(".");
83   }
84   Serial.println("");
85   Serial.println("WiFi connected..!");
86   Serial.print("Got IP: "); Serial.println(WiFi.localIP());

```

Εικόνα 90. Κώδικας Arduino

- Για να μπορέσουμε να διαχειριστούμε τα εισερχόμενα αιτήματα στον server, πρέπει να προσδιοριστεί ποιο κομμάτι κώδικα θα εκτελείται όταν ανοίγει ένα URL. Για να πραγματοποιηθεί αυτό, χρησιμοποιείται η μέθοδος **On**. Αυτή η μέθοδος παίρνει δυο παραμέτρους. Η πρώτη είναι το path του URL και η



δεύτερη είναι το όνομα της συνάρτησης που θέλουμε να εκτελείται όταν ανοίγει αυτό το URL. Ο κώδικας στις γραμμές 88-89 υποδεικνύει ότι όταν ο server λαμβάνει ένα HTTP αίτημα στο path (/), θα εκτελέσει την **handle\_OnConnect** συνάρτηση. Στη γραμμή 91-92 ξεκινάει ο server. Αντίστοιχα πρέπει να οριστεί τι θα κάνει ο server αν δεν λάβει κάποιο αίτημα. Το ρόλο αυτό τον αναλαμβάνει η μέθοδος **onNotFound** και η συνάρτηση **handle\_NotFound**. Τέλος στη γραμμή **95** έχει δημιουργηθεί μια συνάρτηση που τρέχει κάθε δευτερόλεπτο. Είναι απαραίτητη για το Blynk.

```
88 server.on("/", handle_OnConnect);
89 server.onNotFound(handle_NotFound);
90
91 server.begin();
92 Serial.println("HTTP server started");
93
94 // Setup a function to be called every second
95 timer.setInterval(1000L, sendSensor);
96
```

Εικόνα 91. Κώδικας Arduino

- Από τη γραμμή 98-106 τρέχει η συνάρτηση **void loop()**. Ο κώδικας που βρίσκεται μέσα σε αυτή τη συνάρτηση εκτελείται συνεχώς ως ένας αέναος βρόγχος. Στο εσωτερικό της καλούνται διαφορετικές συναρτήσεις που είναι υπεύθυνες για την λειτουργία του προγράμματος. Στη γραμμή 99 καλείται η συνάρτηση **values()** η οποία είναι υπεύθυνη για την λειτουργία των αισθητήρων και την απαγωγή των δεδομένων τους. Στη γραμμή 100 καλείται η μέθοδος **handleClient()** στο αντικείμενο **server** η οποία διαχειρίζεται τα εισερχόμενα αιτήματα HTTP. Στη συνέχεια στη γραμμή 101 καλείται η συνάρτηση **sql()** η οποία διαχειρίζεται τη σύνδεση με τη βάση δεδομένων. Από τη γραμμή **103-106** καλούνται συναρτήσεις που σχετίζονται με τη λειτουργία της εφαρμογής **Blynk**.

```
98 void loop(void) {
99   values();
100   server.handleClient();
101   sql();
102 |
103   Blynk.run();
104   timer.run(); // Initiates BlynkTimer
105   sendSensor();
106   delay(1000);
107
108 }
```

Εικόνα 92. Κώδικας Arduino

- Στη γραμμή **114-139** τρέχει η συνάρτηση **void values()** η οποία όπως αναφέρθηκε και προηγουμένως διαχειρίζεται τους αισθητήρες και τις μετρήσεις τους.

- Στη γραμμή **141-150** τρέχει η συνάρτηση **void sendSensor()**. Είναι υπεύθυνη για την αποστολή των δεδομένων και την επικοινωνία του μικροελεγκτή με την εφαρμογή **Blynk** στο smartphone. Το **V5-V9** στις εντολές **Blynk.virtualWrite()** είναι ροές δεδομένων (Datastreams) και θα εξηγηθούν σε επόμενη παράγραφο, για την διαμόρφωση της εφαρμογής.

```

141 void sendSensor() {
142
143     // You can send any value at any time.
144     // Please don't send more that 10 values per second.
145     Blynk.virtualWrite(V5, Temperature);
146     Blynk.virtualWrite(V6, Humidity);
147     Blynk.virtualWrite(V7, SP02);
148     Blynk.virtualWrite(V8, heartRate);
149     Blynk.virtualWrite(V9, tempC);
150 }

```

Εικόνα 93. Κώδικας Arduino

- Στη γραμμή 152-176 τρέχει η συνάρτηση **void sql()** η οποία διαχειρίζεται τη σύνδεση με τη βάση δεδομένων. Στη γραμμή 158 προετοιμάζεται το αίτημα τύπου **POST** με τις μετρήσεις των αισθητήρων ώστε να μεταβιβαστούν στη βάση δεδομένων. Η χρήση της μεταβλητής **apiKeyValue** μπορεί να έχει οποία τιμή θέλουμε. Ο μόνος λόγος που χρησιμοποιείται είναι για ασφάλεια, έτσι ώστε μόνο όποιος γνωρίζει την τιμή της μεταβλητής να μπορεί να δημοσιεύσει στη βάση δεδομένων. Στη συνέχεια στη γραμμή **162**, με τη μέθοδο **POST** στέλνεται το αίτημα με τις τιμές των αισθητήρων ώστε να ληφθεί από το κώδικα **PHP** και να αποθηκευτούν στη βάση δεδομένων.

```

158 String httpRequestData = "api_key=" + apiKeyValue +
159 //String httpRequestData = "api_key=theiotprojects&s
160 Serial.print("httpRequestData: ");
161 Serial.println(httpRequestData);
162 // Send HTTP POST request
163 int httpResponseCode = http.POST(httpRequestData);

```

Εικόνα 94. Κώδικας Arduino

- Στις γραμμές 178-185 βρίσκονται οι συναρτήσεις **void handle\_OnConnect()** και **void handle\_NotFound()**. Η πρώτη είναι υπεύθυνη για την αποστολή των τιμών των αισθητήρων στον Web Server για την real-time παρακολούθηση τους από κάποιον Browser. Η δεύτερη εμφανίζει το μήνυμα "Not found" αν δεν είναι επιτυχής η αποστολή.
- Τέλος στις γραμμές 188-307 πραγματοποιείται το γραφικό κομμάτι του Web Server. Ο κώδικας δημιουργεί εικονίδια για την πιο φιλική προς το χρήστη εμπειρία που συνοδεύονται από τις τιμές των αισθητήρων.

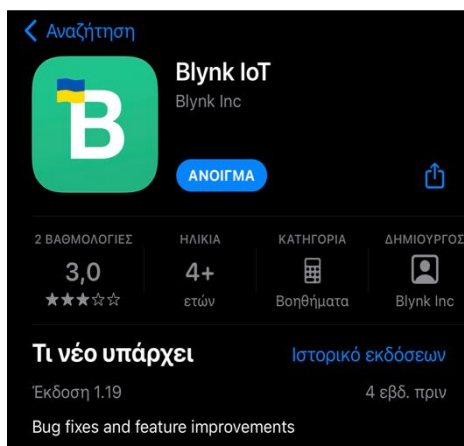
**Ολόκληρος ο κώδικας βρίσκεται στο τέλος της εργασίας.**

## 4.3 Εγκατάσταση και ρύθμιση της εφαρμογής Blynk

### 4.3.1 Εφαρμογή στο Smartphone

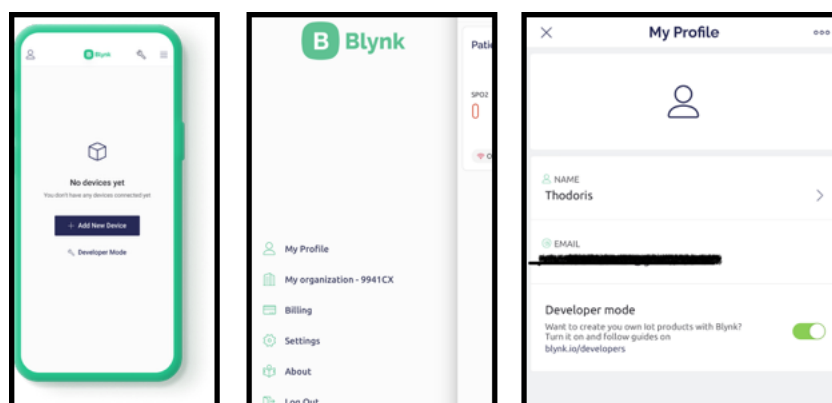
Το Blynk είναι μια απλή αλλά ισχυρή εφαρμογή, που δεν χρειάζεται την γνώση προγραμματισμού. Με αυτήν μπορεί κάποιος να πρωτοτυπήσει, να κατασκευάσει και να διαχειριστεί τις συνδεδεμένες ηλεκτρονικές συσκευές οποιουδήποτε μεγέθους. Υποστηρίζει συσκευές όπως το Arduino, ESP8266, ESP32, Raspberry Pi κ.ά. Παρακάτω θα περιγραφούν τα βήματα για την εγκατάσταση και τη διαμόρφωση της εφαρμογής σε Smartphone. Για την υλοποίηση της συγκεκριμένης εργασίας έγινε χρήση του λογισμικού συστήματος iOS, αλλά με τα ίδια βήματα η εφαρμογή μπορεί να ρυθμιστεί και σε συσκευές Android.

- Αρχικά είναι απαραίτητη η εγκατάσταση της εφαρμογής. Για το σκοπό αυτό ο χρήστης πρέπει μέσα από την εφαρμογή **App Store** να κατεβάσει και να εγκαταστήσει την εφαρμογή **“Blynk IoT”**.



Εικόνα 95. Blynk IoT app στο App Store

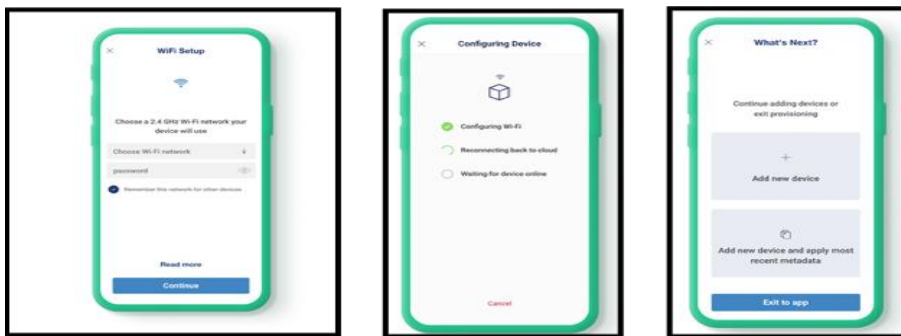
- Αφού ανοίξουμε την εφαρμογή, εμφανίζεται η πρώτη οθόνη που έχει δυο επιλογές. Είτε να συνδεθούμε με ένα υπάρχον λογαριασμό είτε να φτιάξουμε ένα καινούριο. Ακολουθώντας τα βήματα που εμφανίζονται φτιάχνουμε ένα



Εικόνα 96. Ενεργοποίηση Developer Mode στο Blynk app

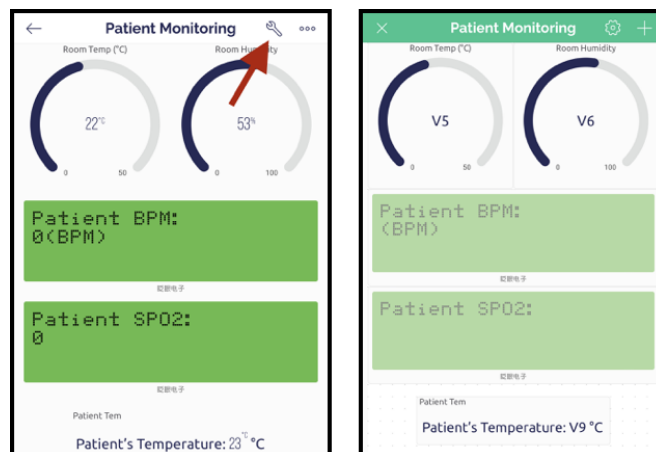
νέο λογαριασμό. Ο σκοπός της δημιουργίας account είναι για λόγους ασφαλείας. Με αυτό τον τρόπο μπορεί κάποιος να εισέλθει από οποιαδήποτε συσκευή στο δικό του project. Στη συνέχεια εμφανίζεται η οθόνη για την προσθήκη μιας νέας συσκευής. Στην περίπτωση που δεν είναι ενεργοποιημένη η επιλογή “Developer Mode”, από την αριστερή στήλη την ενεργοποιούμε στο path **My Profile > Developer Mode**

- Έχοντας ανεβάσει τον κώδικα Arduino στον μικροελεγκτή, επιστρέφουμε στην αρχική οθόνη, πατάμε το **“Add New Device”** και περιμένουμε να εμφανιστεί το όνομα του μικροελεγκτή μας, στην προκειμένη περίπτωση το ESP8266 στη λίστα. Αφού συνδεθεί η συσκευή, η εφαρμογή ζητάει τα credentials για να συνδεθεί στο δίκτυο WiFi που είναι συνδεδεμένος ο μικροελεγκτής. Πρέπει να επισημανθεί ότι το δίκτυο πρέπει να είναι 2.4 GHz αλλιώς δεν θα πραγματοποιηθεί η σύνδεση. Αφού ολοκληρωθεί η διαδικασία γίνονται κάποιοι τελευταίοι έλεγχοι από την εφαρμογή. Τέλος αλλάζουμε το όνομα της εφαρμογής μας και πατάμε **“Exit to app”**.



Εικόνα 97. Σύνδεση ESP8266 με το Blynk app

- Στη συνέχεια πραγματοποιείται η παραμετροποίηση της εφαρμογής. Εντός του project πατάμε στο σύμβολο του “κλειδιού” στην πάνω δεξιά γωνία. Στην επόμενη οθόνη εμφανίζεται ο “καμβάς” που μπορούμε να παραμετροποιήσουμε το project. Πατώντας στην οθόνη εμφανίζονται όλες οι επιλογές γραφικών, μετρητών και πινάκων που είναι διαθέσιμες στην



Εικόνα 98. Παραμετροποίηση Εφαρμογής

εφαρμογή. Όπως φαίνεται και στην εικόνα χρησιμοποιήθηκαν δυο μετρητές τύπου “Gauge”, δυο οθόνες τύπου “LCD” και ένα απλό κουτί προβολής τιμών.

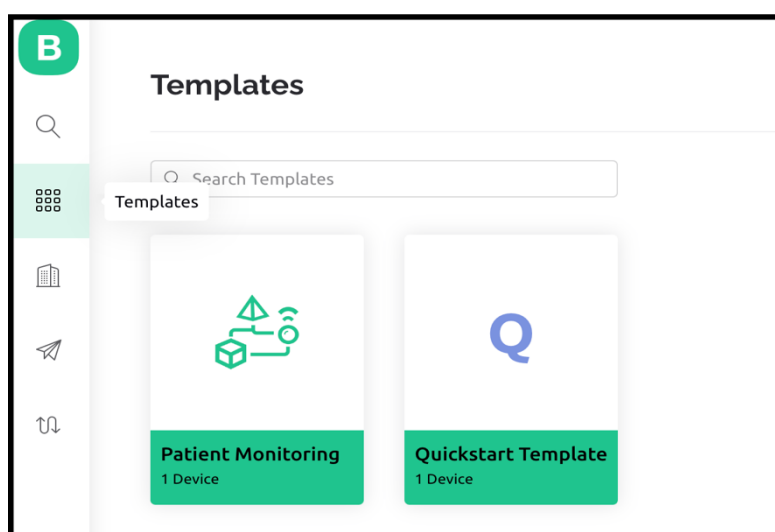
- Για τις τιμές των αισθητήρων επέλεξα τις εξής ροές δεδομένων:

Θερμοκρασία Δωματίου	Datastream V5
Υγρασία Δωματίου	Datastream V6
Οξυμετρία (SpO2)	Datastream V7
Καρδιακοί Παλμοί (Bpm)	Datastream V8
Θερμοκρασία Ασθενούς	Datastream V9

#### 4.3.2 Ρύθμιση του Blynk στην Web εφαρμογή

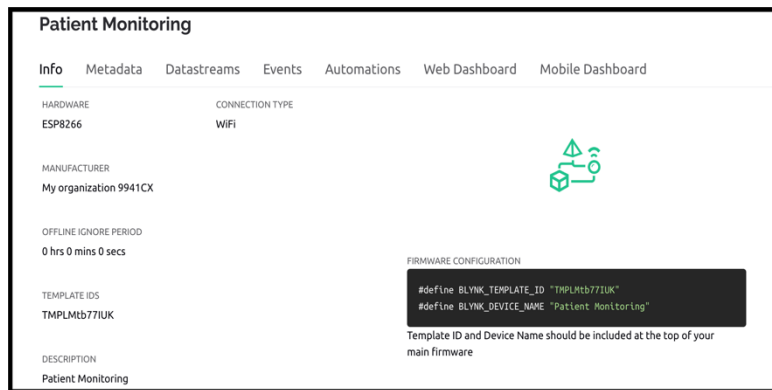
Για να λειτουργήσει το Blynk πρέπει να γίνουν κάποιες τελευταίες ρυθμίσεις στην ιστοσελίδα του project. Αυτές οι ρυθμίσεις δεν είναι διαθέσιμες στην εφαρμογή.

- Αρχικά μπαίνουμε στη διεύθυνση <https://blynk.cloud/> και κάνουμε είσοδο στο site με το account που δημιουργήσαμε στην εφαρμογή. Στη συνέχεια από την αριστερή στήλη μεταβαίνουμε στο Tab “**Templates**” επιλέγουμε το project μας. Στην προκειμένη περίπτωση το “**Patient Monitoring**”.



Εικόνα 99. Ρύθμιση Blynk από τον Web Browser PC

- Στο tab “Info” εμφανίζονται κάποιες πολύ σημαντικές πληροφορίες για την επικοινωνία του μικροελεγκτή με την εφαρμογή Blynk. Για την ακρίβεια το “**BLYNK\_TEMPLATE\_ID**” καθώς και το “**BLYNK\_DEVICE\_NAME**”. Αυτές οι δυο γραμμές πρέπει να αντιγραφούν και να γίνουν επικόλληση στον κώδικα του ESP8266.

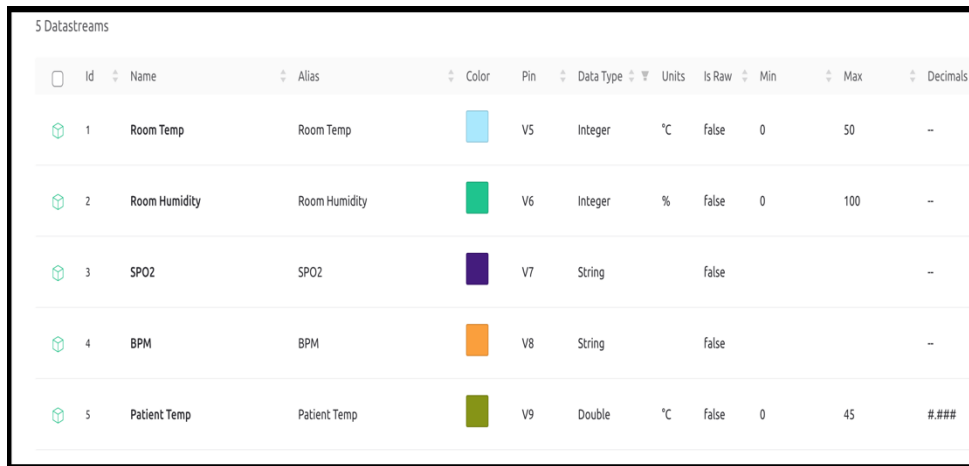


Εικόνα 100. Παράμετροι Blynk για τον κώδικα του Arduino

- Στη συνέχεια επιλέγοντας το tab “Datastream”, πατάμε το “edit” και μετά “New Datastream”. Με αυτόν τον τρόπο δηλώνουμε έναν-έναν τα στοιχεία των αισθητήρων που θα λαμβάνει το Blynk από το ESP8266. Κάποιες από τις πληροφορίες είναι το όνομα που θα εμφανίζει ο κάθε αισθητήρας στην εφαρμογή, ο τύπος των δεδομένων που θα λαμβάνονται (π.χ ακέραιος, με υποδιαστολή ή αλφαριθμητικό) καθώς το Datastream στο οποίο ανατίθεται. Το Datastream που θα δηλωθεί για τον κάθε αισθητήρα θα πρέπει να περαστεί στη συνέχεια και στο πρόγραμμα του **ESP8266** όπως αναλύθηκε προηγουμένως.

Εικόνα 101. Παραμετροποίηση των Datastream

Η συνολική εικόνα των αισθητήρων πρέπει να είναι αυτή:



Id	Name	Alias	Color	Pin	Data Type	Units	Is Raw	Min	Max	Decimals
1	Room Temp	Room Temp	Light Blue	V5	Integer	°C	false	0	50	--
2	Room Humidity	Room Humidity	Green	V6	Integer	%	false	0	100	--
3	SPO2	SPO2	Purple	V7	String		false			--
4	BPM	BPM	Orange	V8	String		false			--
5	Patient Temp	Patient Temp	Olive Green	V9	Double	°C	false	0	45	###

Εικόνα 102. Τελική Εικόνα των Datastream

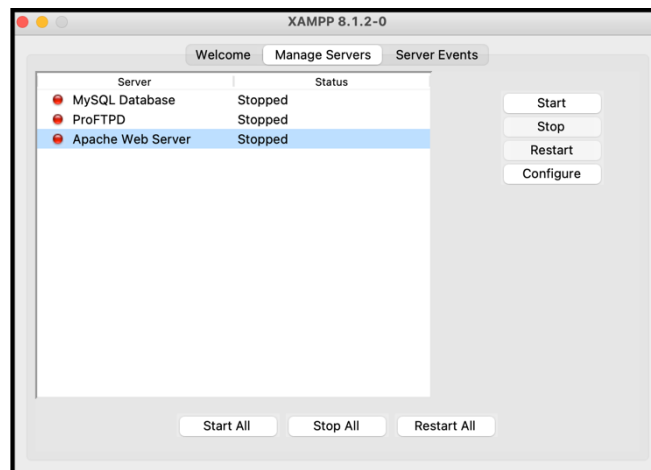
Με αυτό τον τρόπο η εφαρμογή Blynk είναι πλήρως λειτουργική για χρήση μεταξύ του ESP8266 και της εφαρμογής.

#### 4.4 Δημιουργία της Βάσης Δεδομένων

Αρχικά για την υλοποίηση της βάσης δεδομένων πρέπει να κατεβάσουμε την εφαρμογή “XAMPP”. Το “XAMPP” είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοιχτού κώδικα και ανεξάρτητου πλατφόρμας το οποίο περιέχει τον εξυπηρετητή ιστοσελίδων HTTP APACHE, τη βάση δεδομένων MySQL και ένα διερμηνέα για κώδικα γραμμένο σε PHP και Perl. Για την υλοποίηση της εφαρμογής έγινε λήψη της εφαρμογής “xampp-osx-8.1.2-0-installer” από τη διεύθυνση

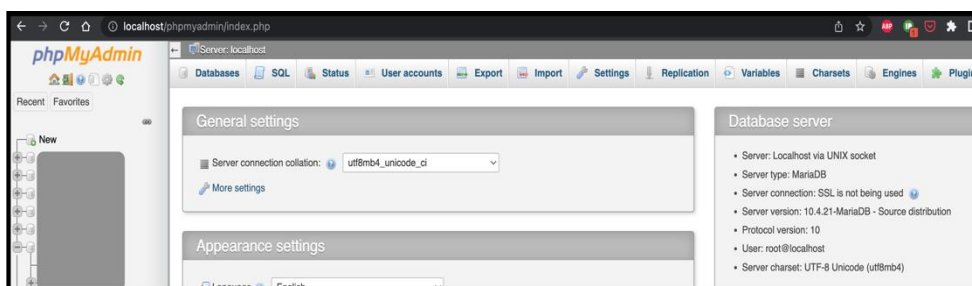
<https://sourceforge.net/projects/xampp/files/XAMPP%20Mac%20OS%20X/>.

Αφού κατεβάσουμε και εγκαταστήσουμε την εφαρμογή, ανοίγουμε το πρόγραμμα. Στη συνέχεια εκκινούμε και τους τρεις Server με το “Start”.



Εικόνα 103. Περιβάλλον Χρήσης XAMPP, MacOS

Στη συνέχεια από τον Browser του υπολογιστή πληκτρολογούμε το URL <http://localhost/phpmyadmin/> , ώστε να εισέλθουμε στο phpMyAdmin και να μπορέσουμε να διαχειριστούμε και να προγραμματίσουμε τη βάση δεδομένων. Την πρώτη φορά που θα εισέλθουμε θα μας ζητηθεί να εισάγουμε ένα **Username & Password**. Αυτά τα στοιχεία θα χρησιμοποιηθούν και αργότερα για τον κώδικα PHP.



Εικόνα 104. Αρχική σελίδα phpMyAdmin

Με το **New > Enter database name> Create** φτιάχνουμε μια νέα βάση δεδομένων όπως φαίνεται στην εικόνα. Για την παρούσα εργασία η βάση δεδομένων ονομάστηκε **“sensor\_db”**.



Εικόνα 105. Δημιουργία της Βάσης Δεδομένων

Αφού δημιουργηθεί η βάση δεδομένων είναι σημαντικό να ενεργοποιήσουμε όλα τα δικαιώματα (privileges) έτσι ώστε να μπορούν να εισαχθούν δεδομένα στη βάση δεδομένων.

Αφού δημιουργηθεί η βάση δεδομένων, την επιλέγουμε από την αριστερή στήλη και στην ενότητα **“SQL”** πληκτρολογούμε τον παρακάτω κώδικα και πατάμε **GO**.

```
CREATE TABLE SensorData (
  id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  location VARCHAR(15) NOT NULL,
  roomtemp VARCHAR(15) NOT NULL,
  roomhumid VARCHAR(10),
  spo2 VARCHAR(10),
  bpm VARCHAR(10),
  bodytemp VARCHAR(10),
  reading_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
)
```

Εικόνα 106. Κώδικας SQL για τον προγραμματισμό της Βάσης Δεδομένων

Αρχικά με τον εντολή **CREATE TABLE** δημιουργούμε νέο πίνακα. Στη συνέχεια επειδή μας ενδιαφέρει όλα τα σεντ μετρήσεων να έχουν μια τιμή που θα τις προσδιορίζει με



μοναδικό τρόπο χρησιμοποιούμε την μεταβλητή και της προσδίδουμε τις ιδιότητες **πρωτεύοντος κλειδιού (PRIMARY KEY)**. Επίσης θέλουμε σε κάθε σετ μετρήσεων το κλειδί να αυξάνει κατά μια μονάδα οπότε χρησιμοποιούμε την εντολή **AUTO\_INCREMENT**. Με την εντολή **UNSIGNED** το κλειδί μας δεν έχει αρνητικές τιμές ενώ με την χρήση του **INT** παίρνει μόνο ακέραιες τιμές. Στη συνέχεια ορίζονται οι μεταβλητές **location, roomtemp, roomhumid, spo2, bpm, bodytemp**. Οι μεταβλητές αυτές έχουν οριστεί ως **VARCHAR**, δηλαδή αλφαριθμητικές τιμές. Οι μεταβλητές **location** και **roomtemp** μπορούν να πάρουν μέχρι 15 χαρακτήρες ενώ οι άλλες μεταβλητές μέχρι 10. Τέλος επειδή θέλουμε να ξέρουμε την ακριβή ημερομηνία και την ώρα που πάρθηκε η κάθε μέτρηση, προσδίδουμε στην μεταβλητή **reading\_time** τον τύπο **TIMESTAMP (χρονοσφραγίδα)**. Με το **CURRENT\_TIMESTAMP** η πρώτη τιμή δεδομένων παίρνει την ημερομηνία και την ώρα που είχε εκείνη τη στιγμή ο **server**.

## 4.5 Επεξήγηση Κώδικα PHP

Αφού ολοκληρώθηκε το κομμάτι της βάσης δεδομένων, τώρα πρέπει να γίνει η σύνδεση της βάσης με τον μικροελεγκτή ώστε να αποθηκεύονται τα δεδομένα. Το αρχείο θα γραφτεί σε μορφή PHP και πρέπει να αποθηκευτεί με το όνομα "**post-sensor-data.php**". Αν αλλάξει το όνομα του αρχείου θα πρέπει να αλλάξει και στον κώδικα του ESP8266 αντίστοιχα. Επίσης το δεύτερο αρχείο που θα γραφτεί σε PHP είναι υπεύθυνο για την απεικόνιση των αποτελεσμάτων της βάσης δεδομένων σε μια ιστοσελίδα. Σε αυτή την ιστοσελίδα θα δίνεται η δυνατότητα στον χρήστη να κάνει αναζήτηση των αποτελεσμάτων με διάφορα φίλτρα για μεγαλύτερη αποτελεσματικότητα.

### 4.5.1 Σύνδεση της βάσης δεδομένων με τον μικροελεγκτή μέσω PHP

Αρχικά στο πρόγραμμα ορίζονται κάποιες μεταβλητές ώστε να μπορέσει να επιτευχθεί η σύνδεση μεταξύ του κώδικα PHP και του μικροελεγκτή και του κώδικα PHP με την βάση δεδομένων. Αρχικά ορίζεται η **μεταβλητή \$servername = "localhost"** (**localhost** διότι η βάση δεδομένων φιλοξενείται στον ίδιο **Server** που φιλοξενείται και το αρχείο **PHP**). Οι μεταβλητές **\$dbname, \$username, \$password** παίρνουν τις ίδιες τιμές από τις αντίστοιχες που ορίσαμε στη βάση δεδομένων στο **phpMyAdmin**. Το **\$api\_key\_value** πρέπει να έχει την ίδια τιμή με την τιμή που ορίσαμε στον κώδικα του ESP8266 για τους λόγους που αναφέρθηκαν σε προηγούμενη ενότητα. Επίσης δημιουργούνται οι μεταβλητές **\$api\_key = \$location =**

**\$roomtemp = \$roomhumid = \$spo2 = \$bpm = \$bodytemp =**. Αυτές οι μεταβλητές θα “υποδεχτούν” τις τιμές των αντίστοιχων μεταβλητών από τον κώδικα του **ESP8266**.

```
$servername = "localhost";
// REPLACE with your Database name
$dbname = "sensor_db";
// REPLACE with Database user
$username = "admin";
// REPLACE with Database user password
$password = "1234";

// Keep this API Key value to be compatible with the ESP8266 code provided in the project page.
// If you change this value, the ESP8266 sketch needs to match
$api_key_value = "123";

$api_key = $location = $roomtemp = $roomhumid = $spo2 = $bpm = $bodytemp = "";
```

Εικόνα 107. *post-sensor-data.php* Κώδικας

Στο παρακάτω κομμάτι κώδικα ελέγχουμε αρχικά αν έχουμε λάβει κάποιο αίτημα μέσω της μεθόδου **POST**. Εφόσον δεν υπάρχει κάποιο πρόβλημα από τη μεριά του κώδικα στο Arduino IDE, ο κώδικας συνεχίζει ελέγχοντας ότι η μεταβλητή **\$api\_key** δεν περιέχει περιττούς χαρακτήρες όπως κενά ή ανάποδη κάθετο κ.ά καλώντας τη συνάρτηση **test\_input()**. Σε περίπτωση που έχει περιττούς χαρακτήρες τους αφαιρεί. Στη συνέχεια ελέγχεται αν η τιμή που στάλθηκε από το Arduino IDE είναι ίδια με την τιμή ελέγχου μας. Εφόσον δεν προκύπτει κάποιο πρόβλημα καλείται η συνάρτηση **test\_input()** για όλες τις μεταβλητές που στέλνονται από το **Arduino IDE** και οι τιμές αποθηκεύονται στις αντίστοιχες μεταβλητές που δημιουργήσαμε προηγουμένως.

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $api_key = test_input($_POST["api_key"]);
    if($api_key == $api_key_value) {
        $location = test_input($_POST["location"]);
        $roomtemp = test_input($_POST["Room_Temp"]);
        $roomhumid = test_input($_POST["humidity"]);
        $spo2 = test_input($_POST["SpO2"]);
        $bpm = test_input($_POST["BPM"]);
        $bodytemp = test_input($_POST["Body_Temp"]);
    }
}
```

Εικόνα 108. Κώδικας *post-sensor-data.php*

Στο επόμενο κομμάτι δημιουργείται η επικοινωνία με τη βάση δεδομένων. Εάν η επικοινωνία είναι επιτυχής, δημιουργείται η μεταβλητή **\$sql** η οποία εισάγει στον πίνακα **SensorData** που φτιάξαμε στη **Βάση Δεδομένων**, τις τιμές των αισθητήρων.

```
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO SensorData (location, roomtemp, roomhumid, spo2, bpm, bodytemp)
VALUES ('" . $location . "', '" . $roomtemp . "', '" . $roomhumid . "', '" . $spo2 . "', '" . $bpm . "', '" . $bodytemp . "')";
```

Εικόνα 109. . Κώδικας *post-sensor-data.php*

**Ολόκληρος ο κώδικας βρίσκεται στο τέλος της εργασίας.**

#### 4.5.2 Απεικόνιση και φιλτράρισμα των αποτελεσμάτων της βάσης δεδομένων σε ιστοσελίδα

Εφόσον τα αποτελέσματα αποθηκεύονται στη βάση δεδομένων, είναι πολύ χρήσιμο να μπορούν να απεικονιστούν με οργανωμένο τρόπο. Επίσης λόγω του μεγάλου όγκου δεδομένων είναι απαραίτητη η χρήση φίλτρων που θα καθιστούν δυνατή την εύρεση συγκεκριμένων αποτελεσμάτων. Αρχικά όπως και στο προηγούμενο αρχείο PHP, ορίζονται οι μεταβλητές που είναι υπεύθυνες για τη σύνδεση με τη βάση δεδομένων και ύστερα πραγματοποιείται η σύνδεση.

```
$servername = "localhost";
// REPLACE with your Database name
$dbname = "sensor_db";
// REPLACE with Database user
$username = "admin";
// REPLACE with Database user password
$password = "1234";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

Εικόνα 110. Κώδικας *Sensor-data.php*

Στη συνέχεια δημιουργούνται οι μεταβλητές **\$datefrom**, **\$dateto**, **\$spo2from**, **\$spo2to**, **\$bpmfrom**, **\$bpmtto**, **\$bodyfrom**, **\$bodyto** και **\$where** οι οποίες στην αρχή δεν έχουν κάποια τιμή. Επίσης δημιουργείται ένας πίνακας που τον ονομάζουμε **\$options** και στην αρχή είναι και αυτός κενός.

```
$datefrom='';
if (isset($_POST['datefrom'])) {
    $datefrom=$_POST['datefrom'];
}
$dateto='';
if (isset($_POST['dateto'])) {
    $dateto=$_POST['dateto'];
}

$spo2from='';
if (isset($_POST['spo2from'])) {
    $spo2from=$_POST['spo2from'];
}
$spo2to='';
if (isset($_POST['spo2to'])) {
    $spo2to=$_POST['spo2to'];
}

$bpmfrom='';
if (isset($_POST['bpmfrom'])) {
    $bpmfrom=$_POST['bpmfrom'];
}
$bpmtto='';
if (isset($_POST['bpmtto'])) {
    $bpmtto=$_POST['bpmtto'];
}

$bodyfrom='';
if (isset($_POST['bodyfrom'])) {
    $bodyfrom=$_POST['bodyfrom'];
}
$bodyto='';
if (isset($_POST['bodyto'])) {
    $bodyto=$_POST['bodyto'];
}

$where = '';

$options=array();
```

Εικόνα 111. Κώδικας *sensor-data.php*

Η συνάρτηση “isset()” χρησιμοποιείται για να ελέγξουμε αν μια μεταβλητή έχει δηλωθεί ή όχι. Στη συνέχεια το “\$\_POST” αποτελεί μια καθολική μεταβλητή που χρησιμοποιείται για την συλλογή δεδομένων μετά την υποβολή της φόρμας HTML με τη **method=“post”**. Στην ιστοσελίδα έχω φτιάξει μια φόρμα με οχτώ πεδία εισαγωγής κειμένου, ένα κουμπί τύπου “submit” και ένα τύπου “reset”. Όταν ο χρήστης υποβάλλει τα δεδομένα πατώντας το κουμπί “Search the table”, τα στοιχεία στέλνονται στον κώδικα PHP του αρχείου που έχουμε δηλώσει (σε αυτή την περίπτωση του **sensor-data.php**) και μέσω του “\$\_POST” που αναφέρθηκε προηγουμένως συλλέγονται τα δεδομένα και αποθηκεύονται στις αντίστοιχες μεταβλητές.

```
<form name="search_term" method="post" action="sensor-data.php">
<table border="0" cellpadding="2" cellspacing="2">
<tr>
<td>Date:</td><td><input type="text" name="datefrom" value="" size="20"></td><td><input type="text"
name="dateto" value="" size="20"></td>
<td><?php echo $datefrom; ?> - <?php echo $dateto; ?></td>
</tr>
<tr>
<td>SpO2:</td><td><input type="text" name="spo2from" value="" size="20"></td><td><input type="text"
name="spo2to" value="" size="20"></td>
<td><?php echo $spo2from; ?> - <?php echo $spo2to; ?></td>
</tr>
<tr>
<td>BPM:</td><td><input type="text" name="bpmfrom" value="" size="20"></td><td><input type="text"
name="bpmtto" value="" size="20"></td>
<td><?php echo $bpmfrom; ?> - <?php echo $bpmtto; ?></td>
</tr>
<tr>
<td>Body Temp:</td><td><input type="text" name="bodyfrom" value="" size="20"></td><td><input type="text"
name="bodyto" value="" size="20"></td>
<td><?php echo $bodyfrom; ?> - <?php echo $bodyto; ?></td>
</tr>
</table><br>
<input type="submit" name="search" value="Search the table...">
<input type="reset" name="reset" value="Reset">
</form>
```

Εικόνα 112. Κώδικας sensor-data.php

Εφόσον οι τιμές που έχουν υποβληθεί δεν είναι κενές, γίνονται διασταυρώσεις των δεδομένων με την εντολή

```
$options=array();
if($datefrom!='')
{
//options["reading_time >="2022-04-12 12:00:00";
$options["reading_time >=".=mysql_real_escape_string($conn,$datefrom)."";
}
if($dateto!='')
{
//options["reading_time <="2022-04-12 12:00:00";
$options["reading_time <=".=mysql_real_escape_string($conn,$dateto)."";
}
if($spo2from!='')
{
//options["spo2 >="5";
$options["spo2=".=mysql_real_escape_string($conn,$spo2from)."";
}
if($spo2to!='')
{
//options["spo2 <="5";
$options["spo2<=".=mysql_real_escape_string($conn,$spo2to)."";
}
if($bpmfrom!='')
{
//options["bpm >="5";
$options["bpm>=".=mysql_real_escape_string($conn,$bpmfrom)."";
}
if($bpmtto!='')
{
//options["bpm <="5";
$options["bpm<=".=mysql_real_escape_string($conn,$bpmtto)."";
}
if($bodyfrom!='')
{
//options["bodytemp >="5";
$options["bodytemp=".=mysql_real_escape_string($conn,$bodyfrom)."";
}
if($bodyto!='')
{
//options["bodytemp <="5";
$options["bodytemp<=".=mysql_real_escape_string($conn,$bodyto)."";
}
if(count($options))
{
$where="WHERE ".implode(" AND ",$options);
}
$sql = "SELECT id, location, roomtemp, roomhumid, spo2, bpm, bodytemp, reading_time FROM SensorData $where ORDER BY id DESC";
```

Εικόνα 113. Κώδικας sensor-data.php

“`mysqli_real_escape_string`” και αποθηκεύονται στον πίνακα `$options`. Με τη συνάρτηση `count()` βλέπουμε αν ο πίνακας είναι άδειος ή έχει στοιχεία. Εφόσον έχει στοιχεία φιλτράρονται τα δεδομένα του πίνακα `$options` με τη χρήση της συνάρτησης `implode()` και του λογικού τελεστή “AND” και επιστρέφουν μόνο οι μετρήσεις που ικανοποιούν τη συνθήκη. Στη συνέχεια χρησιμοποιείται η μεταβλητή `$sql` η οποία προετοιμάζει με βάση τα κριτήρια που έχουμε επιλέξει ποιες τιμές θα “ζητηθούν” από τη βάση δεδομένων.

Τέλος στο παρακάτω κομμάτι κώδικα πραγματοποιείται η σύνδεση με τη βάση δεδομένων και παραλαμβάνονται τα αποτελέσματα.

```
if ($result = $conn->query($sql)) {
    while ($row = $result->fetch_assoc()) {
        $row_id = $row["id"];
        $row_location = $row["location"];
        $row_roomtemp = $row["roomtemp"];
        $row_roomhumid = $row["roomhumid"];
        $row_spo2 = $row["spo2"];
        $row_bpm = $row["bpm"];
        $row_bodytemp = $row["bodytemp"];
        $row_reading_time = $row["reading_time"];
    }
}
```

Εικόνα 114. Κώδικας `sensor-data.php`

Είναι πολύ σημαντικό να σημειωθεί ότι για να μπορέσει να λειτουργήσει ο κώδικας “`post-sensor-data.php`” και “`sensor-data.php`” πρέπει να τοποθετηθούν στο path [/Applications/XAMPP/xamppfiles/htdocs](#) .

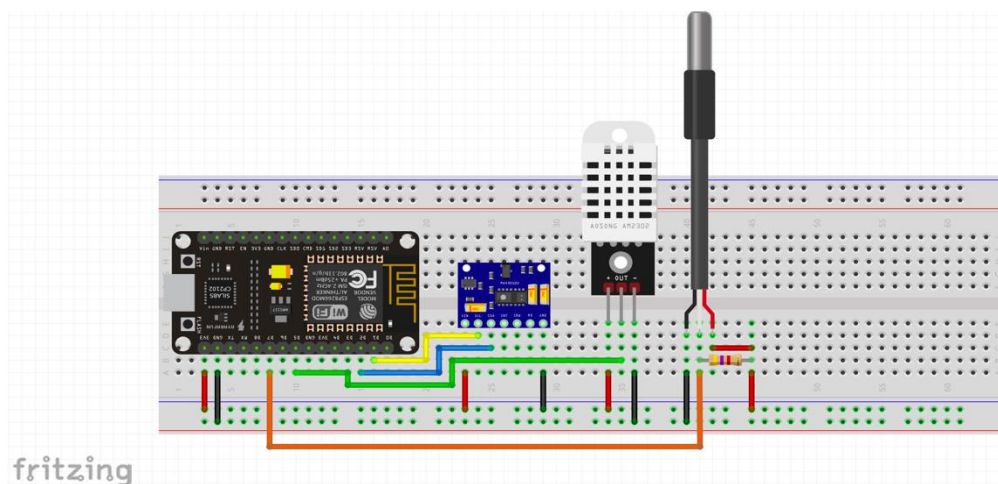
[76],[77],[78],[79],[80],[81],[82],[83],[84],[85],[86],[87],[88],[89],[90],[91],[92],[93],[94],[95],[96]

# 5. ΑΠΟΤΕΛΕΣΜΑΤΑ

Σε αυτό το κεφάλαιο θα παρουσιαστεί το ολοκληρωμένο κατασκευαστικό σύστημα της εργασίας και οι διάφορες λειτουργίες του. Επίσης θα παρουσιαστούν κάποια πειράματα που διενεργήθηκαν ώστε να αξιολογηθεί η ορθή λειτουργία του.

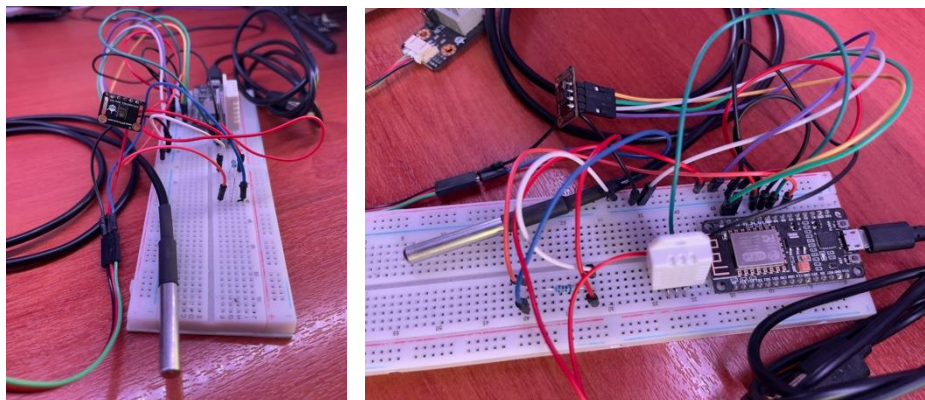
## 5.1 Ολοκληρωμένο Κατασκευαστικό Σύστημα

Στα προηγούμενα κεφάλαια παρουσιάστηκαν αναλυτικά τα επιμέρους εξαρτήματα που αποτελούν την συγκεκριμένη εργασία καθώς και ο κώδικας προγραμματισμού. Έτσι λοιπόν το ολοκληρωμένο κατασκευαστικό σύστημα φαίνεται στην **εικόνα 115**.



Εικόνα 115. Ολοκληρωμένο κατασκευαστικό σύστημα στο fritzing

Επίσης το ολοκληρωμένο κύκλωμα που κατασκεύασα σε Breadboard φαίνεται στην **εικόνα 116**.

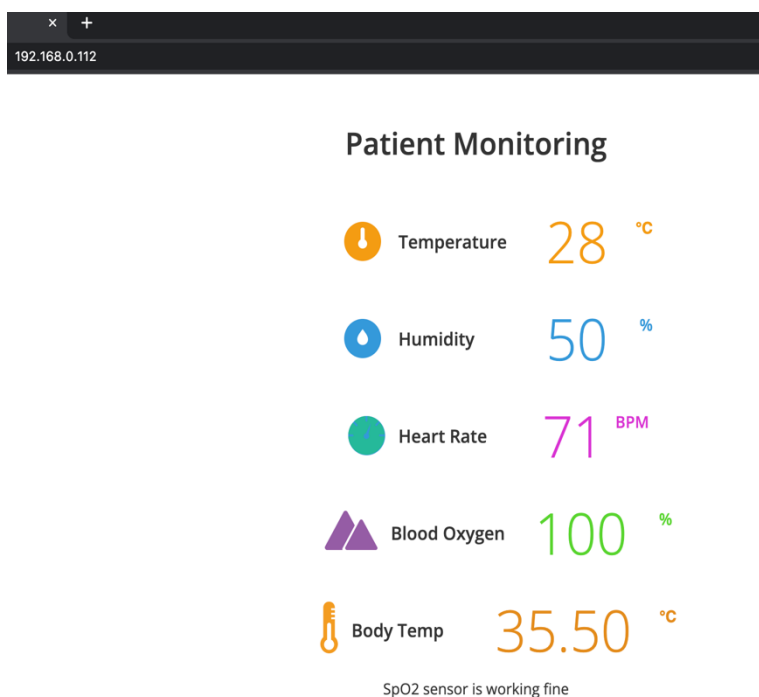


Εικόνα 116. Ολοκληρωμένο κατασκευαστικό σύστημα σε Breadboard

Εκτός από το Υλισμικό και το κατασκευαστικό κομμάτι, το σύστημα έχει κάποιες λειτουργίες και δυνατότητες οι οποίες αναλύθηκαν εκτενώς στο κεφάλαιο του Λογισμικού. Εν συντομία αυτές οι λειτουργίες είναι οι εξής:

- Η δημιουργία ενός Web Server για την Real-Time απεικόνιση των μετρήσεων των αισθητήρων σε Φυλλομετρητή Η/Υ καθώς και στην εφαρμογή για Smartphone Blynk,
- Η δημιουργία μιας βάσης δεδομένων για την αποθήκευση των αποτελεσμάτων και
- Η δημιουργία ξεχωριστής ιστοσελίδας όπου μπορεί κάποιος να ανατρέξει στο πλήρες ιστορικό των μετρήσεων και με τη βοήθεια φίλτρων αναζήτησης να ανατρέξουμε σε συγκεκριμένα αποτελέσματα.

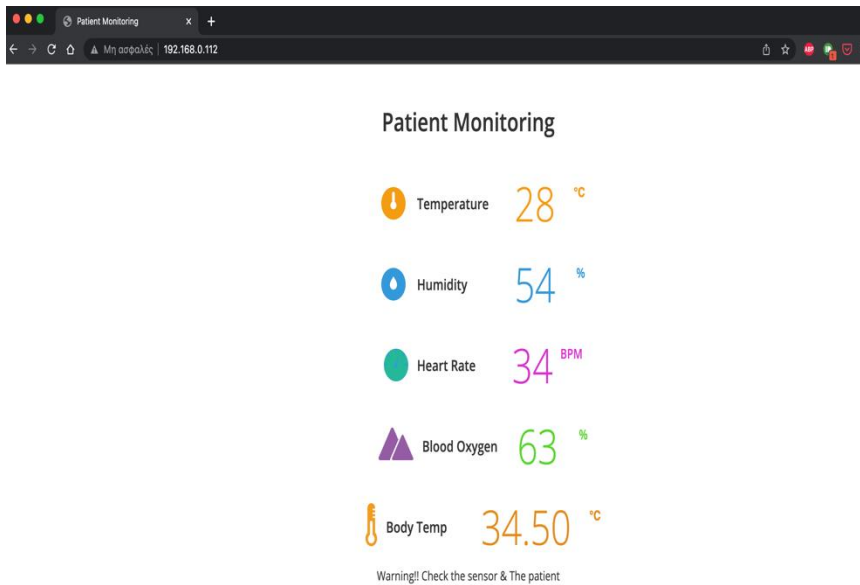
Στην **εικόνα 117** παρουσιάζεται η Real-Time απεικόνιση των αποτελεσμάτων όταν ο αισθητήρας MAX30102 έχει τοποθετηθεί στο δάχτυλο και έχει ανιχνεύσει σήμα. Όπως βλέπουμε εμφανίζεται μήνυμα που μας ενημερώνει ότι ο αισθητήρας λειτουργεί σωστά.



*Εικόνα 117. Real-Time απεικόνιση των αποτελεσμάτων όταν ο αισθητήρας SpO2 λειτουργεί σωστά*

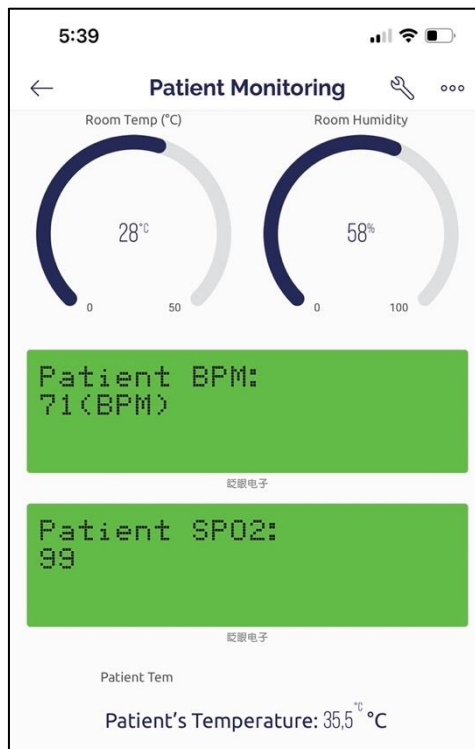
Αντίστοιχα στην **εικόνα 118** φαίνεται η Real-Time απεικόνιση των αποτελεσμάτων όταν ο αισθητήρας MAX30102 έχει αποσυνδεθεί ή έχει κάποιο τεχνικό πρόβλημα ή κάποια από τις μετρήσεις Οξυμετρίας ή Καρδιακών Παλμών είναι εκτός των φυσιολογικών ορίων που έχουμε ορίσει στον κώδικα. .





Εικόνα 118. Απεικόνιση των αποτελεσμάτων όταν ο αισθητήρας SpO2 έχει αποσυνδεθεί ή ο ασθενής παρουσιάζει κάποιο πρόβλημα

Στην εικόνα 119 παρουσιάζεται η απεικόνιση των μετρήσεων στην εφαρμογή Blynk.



Εικόνα 119. Απεικόνιση των αποτελεσμάτων στην εφαρμογή Blynk



Στην **εικόνα 120** φαίνεται η βάση δεδομένων με κάποια από τα αποτελέσματα που έχει αποθηκεύσει.

```
SELECT * FROM `SensorData` ORDER BY `reading_time` DESC
```

id	location	roomtemp	roomhumid	spo2	bpm	bodytemp	reading_time
621	Hospital	28.70	56.90	99	78	28.00	2022-09-11 17:59:06
620	Hospital	28.60	56.70	95	78	28.00	2022-09-11 17:59:01
619	Hospital	28.60	57.30	99	93	28.00	2022-09-11 17:58:55
618	Hospital	28.60	57.00	97	93	28.00	2022-09-11 17:58:50
617	Hospital	28.60	56.80	99	78	28.00	2022-09-11 17:58:45
616	Hospital	28.60	57.10	100	78	28.00	2022-09-11 17:58:39
615	Hospital	28.60	56.60	98	68	28.00	2022-09-11 17:58:34
614	Hospital	28.70	56.60	-1	-1	28.00	2022-09-11 17:58:28
613	Hospital	28.70	56.90	100	75	28.00	2022-09-11 17:58:23
612	Hospital	28.60	56.60	100	93	28.00	2022-09-11 17:58:18
611	Hospital	28.60	56.50	99	75	28.00	2022-09-11 17:58:12
610	Hospital	28.60	56.90	100	93	28.00	2022-09-11 17:58:07
609	Hospital	28.60	56.90	100	88	28.00	2022-09-11 17:58:02
608	Hospital	28.60	57.10	100	88	28.00	2022-09-11 17:57:56

**Εικόνα 120.** Αποθηκευμένες μετρήσεις στη βάση δεδομένων

Τέλος στην **εικόνα 121** φαίνεται η ιστοσελίδα όπου μπορούν να προβληθούν όλα τα δεδομένα που έχουν αποθηκευτεί και να γίνει φιλτράρισμα σε αυτά με κριτήρια αναζήτησης. Στο συγκεκριμένο παράδειγμα επιλέχτηκε να εμφανιστούν όλα τα σετ μετρήσεων όπου η τιμή του SpO2 είναι μεγαλύτερη από 95 και οι καρδιακοί παλμοί πάνω από 70.

**Patient Monitoring**

Date:  -  -

SpO2:  -  95 -

BPM:  -  70 -

Body Temp:  -  -

Search the table...

A/A	Location	Room Temperature	Room Humidity	SpO2	BPM	Body Temperature	Timestamp
621	Hospital	28.70	56.90	99	78	28.00	2022-09-11 17:59:06
620	Hospital	28.60	56.70	95	78	28.00	2022-09-11 17:59:01
619	Hospital	28.60	57.30	99	93	28.00	2022-09-11 17:58:55
618	Hospital	28.60	57.00	97	93	28.00	2022-09-11 17:58:50
617	Hospital	28.60	56.80	99	78	28.00	2022-09-11 17:58:45
616	Hospital	28.60	57.10	100	78	28.00	2022-09-11 17:58:39
613	Hospital	28.70	56.90	100	75	28.00	2022-09-11 17:58:23
612	Hospital	28.60	56.60	100	93	28.00	2022-09-11 17:58:18
611	Hospital	28.60	56.50	99	75	28.00	2022-09-11 17:58:12
610	Hospital	28.60	56.90	100	93	28.00	2022-09-11 17:58:07

**Εικόνα 121.** Ιστοσελίδα προβολής δεδομένων με κριτήρια αναζήτησης

## 5.2 Πειράματα - Έλεγχοι

### 5.2.1 Έλεγχος για την αξιοπιστία των μετρήσεων SpO2 & Heart Rate

Για την παρούσα εργασία πραγματοποιήθηκαν κάποιες μετρήσεις και πειράματα ώστε να ελεγχθεί η πιστότητα των αποτελεσμάτων. Τα στατιστικά εργαλεία που χρησιμοποιήθηκαν είναι η **Τυπική Απόκλιση Πληθυσμού**, ο **Μέσος Όρος** και το **%Σφάλμα**. Για να μπορέσουν να συγκριθούν τα αποτελέσματα, ειδικά της Οξυμετρίας και των Καρδιακών Παλμών χρησιμοποιήθηκε ένα κοινό Οξύμετρο το οποίο πωλείται σε Φαρμακεία. Το Οξύμετρο αυτό αξιολογήθηκε για την πιστότητα στις μετρήσεις του με τη χρήση ενός διακριβωμένου οργάνου και πιο συγκεκριμένα του **Rigel Uni-SiM Patient Simulator**. Αφού εξακριβώθηκε η ορθότητα των μετρήσεων του κοινού Οξυμέτρου, χρησιμοποιήθηκε για την αξιολόγηση του πειραματικού οξυμέτρου. Για την πραγματοποίηση των μετρήσεων **SpO2**, **Heart Rate** συμμετείχαν 6 άνθρωποι διαφορετικού φύλου και ηλικίας, στους οποίους τοποθετήθηκε στο ένα δάχτυλο το κοινό οξύμετρο και στο άλλο αυτός της εργασίας. Με αυτό τον τρόπο κατεγράφησαν μετρήσεις για αυτές τις δυο παραμέτρους ταυτόχρονα.

Στις **εικόνες 122, 123, 124** παρουσιάζονται κάποιες από τις μετρήσεις που έγιναν με το διακριβωμένο όργανο και το οξύμετρο του εμπορίου έτσι ώστε να διαπιστωθεί η ορθότητα των μετρήσεων τόσο στην Οξυμετρία όσο και στους Καρδιακούς Παλμούς.



Εικόνα 122. Έλεγχος κοινού Οξυμέτρου με το Rigel Uni-SiM

Στην **εικόνα 122** βλέπουμε ότι το tester ρυθμίστηκε σε SpO2=99% και Heart Rate=72 bpm. Το προς αξιολόγηση Οξύμετρο ανταποκρίθηκε άριστα, δείχνοντας ακριβώς τις ίδιες τιμές.

Αντίστοιχα στην **εικόνα 123** οι τιμές του tester ορίστηκαν σε SpO2=97% και Heart Rate=75 bpm. Ξανά το προς αξιολόγηση Οξύμετρο ανταποκρίθηκε άριστα, δείχνοντας και πάλι ακριβώς τις ίδιες τιμές.



*Εικόνα 123. Έλεγχος κοινού Οξύμέτρου με το Rigel Uni-SiM*

Τέλος στην **εικόνα 124** οι τιμές του tester ορίστηκαν σε SpO2=89% και Heart Rate=68 bpm. Ξανά το προς αξιολόγηση Οξύμετρο ανταποκρίθηκε άριστα, δείχνοντας και πάλι την ίδια τιμή για το SpO2, ενώ στο Heart Rate είχε απόκλιση μίας μονάδας.



*Εικόνα 124. Έλεγχος κοινού Οξύμέτρου με το Rigel Uni-SiM*

Από τις μετρήσεις βλέπουμε ότι το Οξύμετρο του εμπορίου είναι πλήρως αξιόπιστο και μπορεί να χρησιμοποιηθεί για τον έλεγχο αξιοπιστίας του του MAX30102. Ο λόγος που δεν εξακριβώθηκε κατευθείαν το Οξύμετρο της εργασίας με το διακριβωμένο όργανο είναι λόγω της κατασκευής του MAX30102, το οποίο είναι τεχνολογίας ανάκλασης και δεν υποστηρίζεται από το Rigel Uni-SiM.

	SpO2 %			Heart Rate (bpm)		
α/α	Uni-SiM	Κοινό Οξύμετρο	%Error	Uni-SiM	Κοινό Οξύμετρο	%Error
1	89	89	0	68	67	1,47
2	90	89	1,11	70	70	0
3	91	91	0	72	72	0
4	92	92	0	73	74	1,37
5	93	93	0	75	75	0
6	94	94	0	78	78	0
7	95	95	0	81	80	1,23
8	97	97	0	84	84	0
9	99	99	0	87	87	0
10	100	100	0	90	90	0
	<b>Average %Error</b>		0,11	<b>Average %Error</b>		0,41
	<b>Mean (Uni-SiM/Οξύμετρο)</b>		94/93.9	<b>Mean (Uni-SiM/Οξύμετρο)</b>		77.8/77.7
	<b>Standard Deviation</b>		3.55/3.67	<b>Standard Deviation</b>		7.10/7.14

Όπως φαίνεται από τις μετρήσεις και την στατιστική ανάλυση το **Μέσο SpO2 %Σφάλμα=0,11**, ενώ το **Μέσο HR %Σφάλμα=0,41**. Δηλαδή η ακρίβεια των μετρήσεων κυμαίνεται στο **99.89%** και στο **99.59%** αντίστοιχα. Επίσης από τον **Μέσο Όρο** και την **Τυπική Απόκλιση** βλέπουμε ότι η πιστότητα είναι εξαιρετική και οι τιμές παρουσιάζουν σχεδόν ίδια μεταβλητότητα χωρίς να υπάρχουν ακραίες τιμές.

### 5.2.2 Έλεγχοι και αποτελέσματα πειραματικής διάταξης SpO2 & Heart Rate

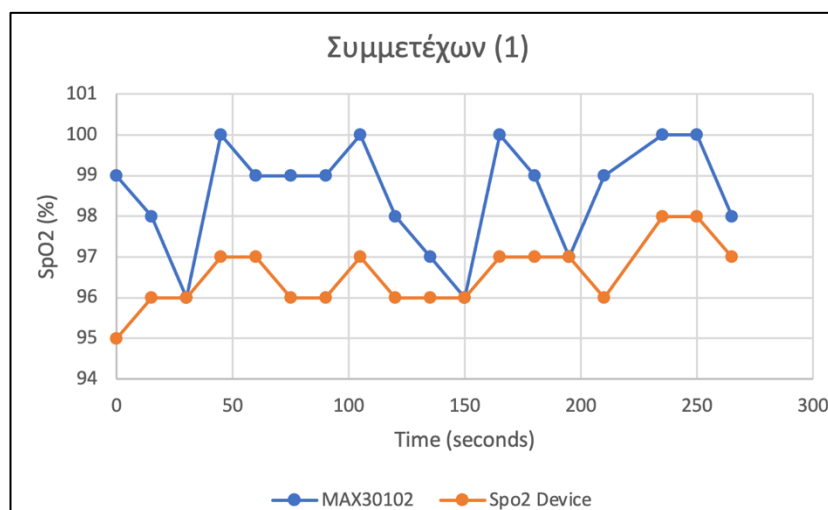
Στον παρακάτω Πίνακα παρουσιάζονται οι μετρήσεις SpO2 που έγιναν με το **MAX30102** και το **Οξύμετρο του εμπορίου** έτσι ώστε να καθοριστούν η ακρίβεια και η αξιοπιστία.

	Μέση Τιμή %SpO2 (18 μετρήσεις/άτομο)			
α/α	Κοινό Οξύμετρο	MAX30102	%Error	Standard Deviation
Male 28 (1)	96.55	98.55	2.07	0.76/1.30
Female 53 (2)	97.11	98.94	1.89	0.87/0.97
Male 12 (3)	97	98.89	1.95	0.75/1.03
Female 42 (4)	96.33	97.89	1.61	0.88/1.59
Male 64 (5)	96.67	98.61	2.01	1.00/2.00
Female 75 (6)	95.89	98.72	2.95	1.2/0.87

Πίνακας αποτελεσμάτων Οξυμετρίας

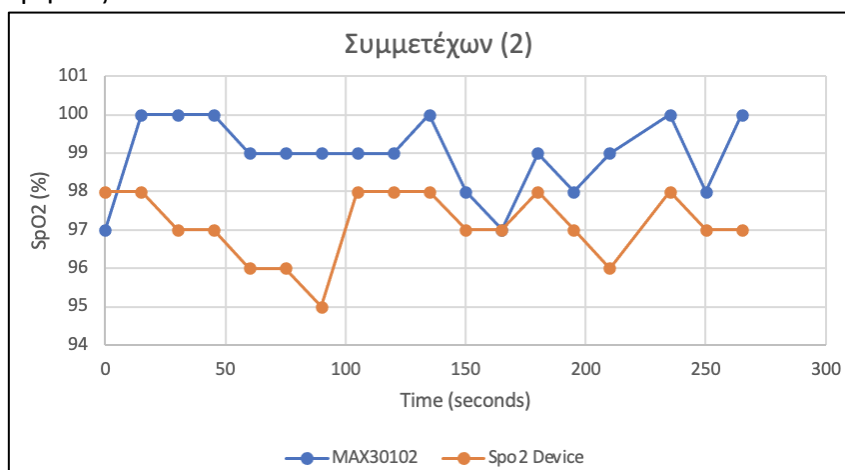
- Ο συμμετέχων (1) είχε μέση μέτρηση **SpO2 96.55%** και **98.55%** με το **Οξύμετρο Εμπορίου** και το **MAX30102** αντίστοιχα. Το **%Σφάλμα** που παρουσίασαν οι μετρήσεις είναι ίσο με **2.07%**. Σύμφωνα με τα διεθνή πρότυπα και κανονισμούς η διαφορά ανάμεσα στη θεωρητική και την πειραματική τιμή Οξυμετρίας πρέπει να είναι μικρότερη ή ίση με **2%**. Παρόλο που τα αποτελέσματα έχουν πολύ καλή ακρίβεια, παρατηρούμε ότι η Τυπική απόκλιση μεταξύ θεωρητικής και πειραματικής τιμής είναι σχεδόν διπλάσια. Αυτό υποδεικνύει ότι η πιστότητα των μετρήσεων του **MAX30102** δεν είναι

καλή, κάτι το οποίο οφείλεται στην παρουσία ακραίων τιμών. Η **εικόνα 127** παρουσιάζει τις μετρήσεις.



Εικόνα 125. Μετρήσεις SpO<sub>2</sub>, Συμμετέχων (1)

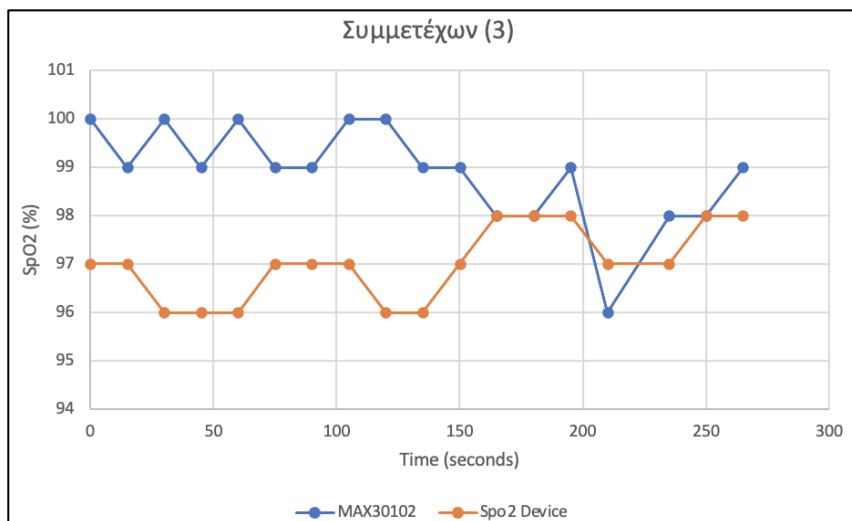
- Η συμμετέχων (2) είχε μέση μέτρηση **SpO<sub>2</sub> 97.11** και **98.94** με το **Οξύμετρο Εμπορίου** και το **MAX30102** αντίστοιχα. Το **%Σφάλμα** που παρουσίασαν οι μετρήσεις είναι ίσο με **1.89%**, δηλαδή μικρότερο από **2%** που ορίζουν τα πρωτόκολλα καθιστώντας την ακρίβεια μέτρησης πολύ καλή. Η **Τυπική απόκλιση** μεταξύ θεωρητικής και πειραματικής τιμής είναι σχεδόν ίδια. Αυτό υποδεικνύει ότι σε αυτή τη περίπτωση η πιστότητα των μετρήσεων του **MAX30102** είναι αρκετά καλή. Απουσιάζουν στο μεγαλύτερο εύρος των μετρήσεων ακραίες τιμές και η πειραματική διάταξη ανταποκρίθηκε στη μεταβλητότητα του φυσικού μεγέθους. Η **εικόνα 128** παρουσιάζει τις μετρήσεις.



Εικόνα 126. Μετρήσεις SpO<sub>2</sub>, Συμμετέχων (2)

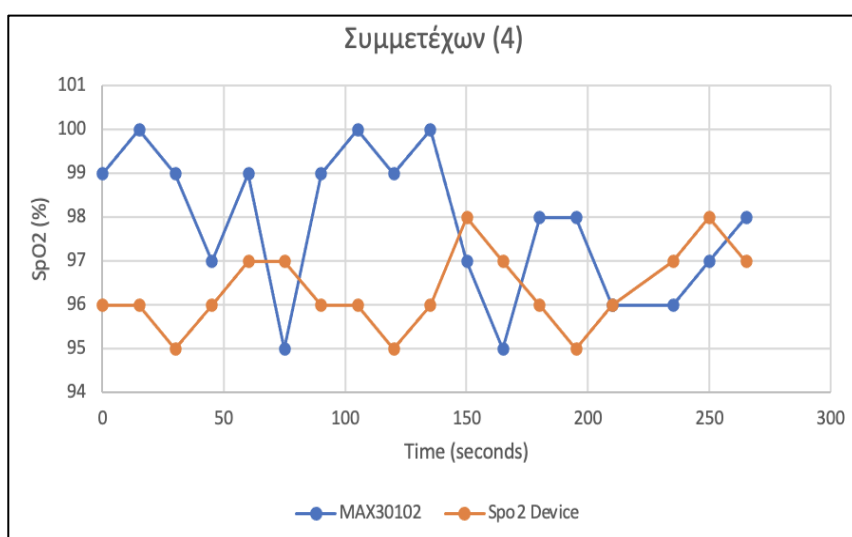
- Ο συμμετέχων (3) είχε μέση μέτρηση **SpO<sub>2</sub> 97** και **98.89** με το **Οξύμετρο Εμπορίου** και το **MAX30102** αντίστοιχα. Το **%Σφάλμα** που παρουσίασαν οι μετρήσεις είναι ίσο με **1.95%**, δηλαδή μικρότερο από **2%** που ορίζουν τα πρωτόκολλα καθιστώντας την ακρίβεια μέτρησης εξαιρετική. Η **Τυπική**

**απόκλιση** μεταξύ θεωρητικής και πειραματικής τιμής είναι σχεδόν **25%**. Αυτό υποδεικνύει ότι η πιστότητα των μετρήσεων του **MAX30102** δεν είναι καθόλου καλή. Οι τιμές παρουσιάζουν μεγάλη μεταβλητότητα σε σχέση με το διακριβωμένο Οξύμετρο καθιστώντας τις μετρήσεις μη αξιόπιστες. Η **εικόνα 129** παρουσιάζει τις μετρήσεις.



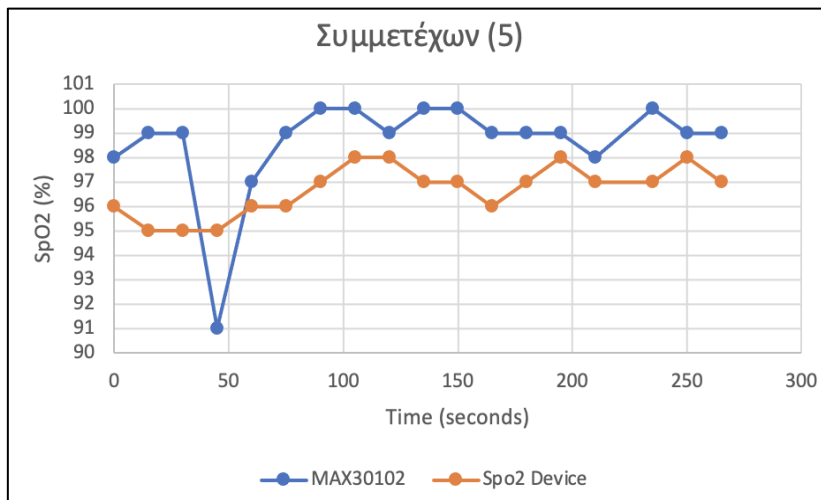
Εικόνα 127. Μετρήσεις Οξυμετρίας, Συμμετέχων (3)

- Η συμμετέχων (4) είχε μέση μέτρηση **SpO2 96.33** και **97.89** με το **Οξύμετρο Εμπορίου** και το **MAX30102** αντίστοιχα. Το **%Σφάλμα** που παρουσίασαν οι μετρήσεις είναι ίσο με **1.61%**. Παρόλο που το **%Σφάλμα** είναι μικρότερο από **2%**, η **Τυπική απόκλιση** μεταξύ θεωρητικής και πειραματικής τιμής είναι σχεδόν διπλάσια. Αυτό δείχνει ότι οι μετρήσεις του **MAX30102** δεν έχουν μεγάλη πιστότητα αφού οι τιμές κυμαίνονται αρκετά υψηλότερα από τον μέσο όρο. Η **εικόνα 130** παρουσιάζει τις μετρήσεις.



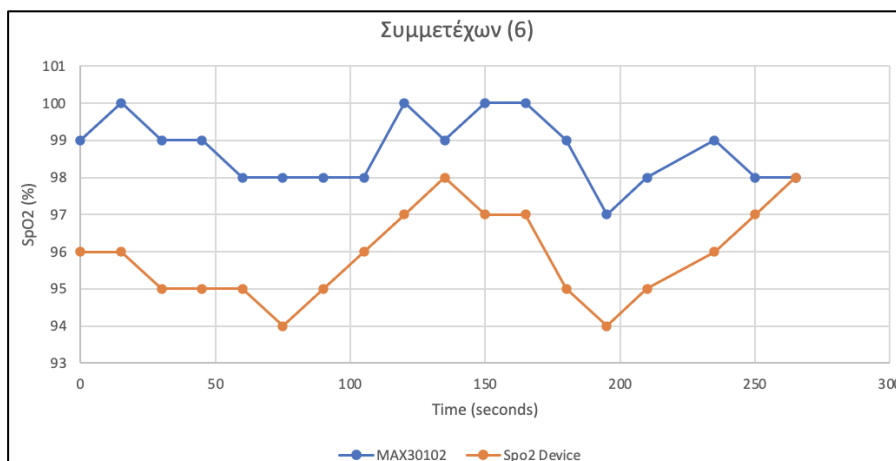
Εικόνα 128. Μετρήσεις Οξυμετρίας, Συμμετέχων (4)

- Ο συμμετέχων (5) είχε μέση μέτρηση **SpO2 96.67** και **98.61** με το **Οξύμετρο Εμπορίου** και το **MAX30102** αντίστοιχα. Το **%Σφάλμα** που παρουσίασαν οι μετρήσεις είναι ίσο με **2.01%**. Το **%Σφάλμα** είναι σε αποδεκτά επίπεδα, αλλά η τυπική απόκλιση των πειραματικών μετρήσεων είναι διπλάσια της θεωρητικής. Σε γενικές γραμμές οι μετρήσεις του **MAX30102** είχαν ομοιομορφία και δεν απείχαν πολύ από τις θεωρητικές τιμές όπως φαίνεται και στην **εικόνα 131**. Ωστόσο υπάρχει μια ακραία τιμή (91%) που χαλάει την ολική αξιοπιστία των μετρήσεων.



Εικόνα 129. Μετρήσεις Οξυμετρίας, Συμμετέχων (5)

- Τέλος η συμμετέχων (6) είχε μέση μέτρηση **SpO2 95.89** και **98.72** με το **Οξύμετρο Εμπορίου** και το **MAX30102** αντίστοιχα. Το **%Σφάλμα** που παρουσίασαν οι μετρήσεις είναι ίσο με **2.95%**, το οποίο φανερώνει κακή ακρίβεια στις μετρήσεις. Επίσης, η **Τυπική απόκλιση** του **MAX30102** είναι μικρότερη από την αντίστοιχη θεωρητική κάτι που φανερώνει ότι το MAX30102 δεν προσαρμοζόταν επαρκώς σε διακυμάνσεις και αλλαγές του φυσικού φαινομένου όπως φαίνεται και στην **εικόνα 132**.



Εικόνα 130. Μετρήσεις Οξυμετρίας, Συμμετέχων (6)



Στον παρακάτω Πίνακα παρουσιάζονται οι μετρήσεις **Καρδιακού Παλμού** που έγιναν με το **MAX30102** και το **Οξύμετρο του εμπορίου** έτσι ώστε να καθοριστούν η ακρίβεια και η αξιοπιστία.

	<b>Μέση Τιμή HR (bpm) (18 μετρήσεις/άτομο)</b>			
<b>α/α</b>	<b>Κοινό Οξύμετρο</b>	<b>MAX30102</b>	<b>%Error</b>	<b>Standard Deviation</b>
<b>Male 28 (1)</b>	85.94	87.89	2.26	1.93/5.30
<b>Female 53 (2)</b>	77.77	82.06	5.58	3.94/6.88
<b>Male 12 (3)</b>	75.17	78.89	4.95	3.29/9.41
<b>Female 42 (4)</b>	74.61	80.78	8.27	3.18/5.33
<b>Male 64 (5)</b>	74.72	78.94	5.65	2.71/3.01
<b>Female 75 (6)</b>	71.67	76.95	7.36	2.3/4.17

Πίνακας αποτελεσμάτων Καρδιακών Παλμών

Στις μετρήσεις των καρδιακών παλμών είναι ευκολότερο να εκτιμηθούν συνολικά τα αποτελέσματα, αφού όλες οι μετρήσεις παρουσιάζουν ίδιες τάσεις. Σε όλους τους ελέγχους που πραγματοποιήθηκαν το μέσο **%Σφάλμα** κυμάνθηκε πάνω από το όριο του **2%** που είναι αποδεκτό σαν σφάλμα. Πιο συγκεκριμένα το ελάχιστο **%Σφάλμα** που καταγράφηκε ήταν **2.26%**, ενώ το υψηλότερο **8.27%**. Εύκολα συμπεραίνουμε ότι η πλειοψηφία των τιμών που καταγράφησαν από το **MAX30102** ήταν αρκετά υψηλότερες από τις πραγματικές και άρα η ακρίβεια των μετρήσεων είναι πολύ κακή.

Όσον αφορά τις μετρήσεις της **τυπικής απόκλισης**, σε όλες σχεδόν τις περιπτώσεις οι τιμές του **MAX30102** παρουσίασαν τουλάχιστον διπλάσιο αποτέλεσμα. Από όλες τις μετρήσεις που πραγματοποιήθηκαν με το διακριβωμένο Οξύμετρο, η **μέγιστη τυπική απόκλιση** που σημειώθηκε ήταν **3.94** ενώ η **ελάχιστη** που σημειώθηκε με το **MAX30102** ήταν **3.01**. Συμπεραίνουμε ότι το **MAX30102** δεν απέδωσε τα αποτελέσματα με καθόλου καλή πιστότητα, παρουσιάζοντας μεγάλη μεταβλητότητα στις μετρήσεις.

- Συνολικά ο αισθητήρας **MAX30102** δεν απέδωσε όπως έπρεπε. Η ακρίβεια στις μετρήσεις του SpO2 ήταν αρκετά καλή, παρόλα αυτά οι τιμές ήταν συστηματικά υψηλότερες από τις θεωρητικές. Στις μετρήσεις των Καρδιακών Παλμών, δεν υπήρχε καθόλου καλή ακρίβεια και παρουσιάζονταν πολύ μεγάλες διακυμάνσεις. Με βάση τις μετρήσεις θεωρώ ότι ο **MAX30102** δεν είναι αξιόπιστος και η χρήση του πρέπει να περιορίζεται μόνο σε εργασίες όπως και η παρούσα.

### 5.2.3 Έλεγχοι και αποτελέσματα πειραματικής διάταξης για την μέτρηση θερμοκρασίας σώματος

Σε αυτή την ενότητα θα παρουσιαστούν τα αποτελέσματα και η στατιστική ανάλυση για τις μετρήσεις θερμοκρασίας σώματος. Όπως αναφέρθηκε και σε προηγούμενες ενότητες, για την πειραματική μέτρηση χρησιμοποιήθηκε ο αισθητήρας DS18B20 σε μορφή Probe ενώ σαν πρότυπο χρησιμοποιήθηκε ένα κοινό ψηφιακό θερμόμετρο που μπορεί να αγοράσει κάποιος σε φαρμακεία. Οι μετρήσεις πραγματοποιήθηκαν στα ίδια 6 άτομα που έγιναν και οι προηγούμενοι έλεγχοι. Στον πίνακα παρακάτω παρουσιάζονται οι μετρήσεις.

Μέτρηση Θερμοκρασίας Σώματος °C (3 μετρήσεις/άτομο)					
α/α	Θεωρητική/Πειραματική Μέτρηση			Μ.Ο Μετρήσεων	%Σφάλμα
	1 <sup>η</sup> Μέτρηση	2 <sup>η</sup> Μέτρηση	3 <sup>η</sup> Μέτρηση		
<b>Male 28 (1)</b>	35.7/35.6	35.9/35.8	36.0/35.9	35.9/35.8	0.28
<b>Female 53 (2)</b>	36.3/35.8	36.0/35.8	35.9/35.7	36.1/35.8	0.83
<b>Male 12 (3)</b>	36.6/36.5	36.7/36.5	36.5/36.5	36.6/36.5	0.27
<b>Female 42 (4)</b>	36.3/36	36.4/36.2	36.2/36	36.3/36.1	0.64
<b>Male 64 (5)</b>	36.5/36.5	36.6/36.6	36.8/36.8	36.6/36.5	0.36
<b>Female 75 (6)</b>	35.8/36	35.9/36.1	36/36.1	35.9/36.1	0.46

Πίνακας αποτελεσμάτων Θερμοκρασίας Σώματος

Από τα αποτελέσματα των μετρήσεων είναι εμφανές ότι ο αισθητήρας DS18B20 απέδωσε εξαιρετικά στην μέτρηση της θερμοκρασίας σώματος. Η μεγαλύτερη διαφορά που σημειώθηκε ανάμεσα σε θεωρητική και πειραματική τιμή ήταν 0.5 °C, κάτι που υποδηλώνει υψηλή ακρίβεια και πιστότητα στις μετρήσεις. Οι μέσοι όροι των μετρήσεων είναι σχεδόν ίδιοι και το **%Σφάλμα** δεν ξεπέρασε ποτέ το **0.87%**. Είναι ασφαλές λοιπόν να συμπεράνουμε ότι ο αισθητήρας είναι πολύ αξιόπιστος και θα μπορούσε να χρησιμοποιηθεί σε καταστάσεις που απαιτείται μεγάλη ακρίβεια και πιστότητα στην μέτρηση θερμοκρασίας.

#### 5.2.4 Έλεγχοι και αποτελέσματα πειραματικής διάταξης για την μέτρηση θερμοκρασίας και υγρασίας χώρου

Σε αυτή την ενότητα θα παρουσιαστούν τα αποτελέσματα και η στατιστική ανάλυση για τις μετρήσεις θερμοκρασίας & υγρασίας χώρου. Όπως αναφέρθηκε και σε προηγούμενες ενότητες, για την πειραματική μέτρηση χρησιμοποιήθηκε ο αισθητήρας DHT22 ενώ σαν πρότυπο χρησιμοποιήθηκε ένα κοινό ψηφιακό θερμόμετρο & υγρασιόμετρο που μπορεί να αγοράσει κάποιος σε φαρμακεία. Οι μετρήσεις πραγματοποιήθηκαν σε 5 διαφορετικούς χώρους με διαφορετικές συνθήκες ο καθένας. Για τον κάθε χώρο πάρθηκαν 2 μετρήσεις θερμοκρασίας και υγρασίας. Επίσης και η θεωρητική και η πειραματική διάταξη είχαν 5' ώστε να προσαρμοστούν στον εκάστοτε χώρο. Στον πίνακα παρακάτω παρουσιάζονται οι μετρήσεις.

Θερμοκρασία (°C) & Υγρασία (RH%) Χώρου					
α/α		1 <sup>η</sup> Μέτρηση	2 <sup>η</sup> Μέτρηση	Μ.Ο Μετρήσεων	%Σφάλμα
Χώρος 1	Θεωρητική	23/48	23.2/48	23.1/48	0/1.25
	Πειραματική	22.9/49	23.2/48.2	23.1/48.6	
Χώρος 2α	Θεωρητική	25.4/62	26.7/59	26.1/60.5	1.9/0.5
	Πειραματική	25.9/61.4	27.2/59	26.6/60.2	
Χώρος 3	Θεωρητική	23.5/45	23/45	23.3/45	1.3/1.1
	Πειραματική	23/45.8	22.9/43.1	23/44.5	
Χώρος 4	Θεωρητική	27.1/58	27.2/59	27.2/58.5	1.8/0.9
	Πειραματική	27.7/59	27.7/59	27.7/59	
Χώρος 5	Θεωρητική	25/67	25.4/65	25.2/66	2/1.5
	Πειραματική	25.6/68	25.7/66	25.7/67	

Πίνακας αποτελεσμάτων Θερμοκρασίας & Υγρασίας Χώρου

Από τα αποτελέσματα των μετρήσεων βλέπουμε ότι οι διαφορές μεταξύ Θεωρητικού και Πειραματικού οργάνου είναι πολύ μικρές. Παρατηρούμε ότι η μέγιστη διαφορά που σημειώνεται μεταξύ θεωρητικής και πειραματικής θερμοκρασίας είναι **0.7 °C**, ενώ για την υγρασία είναι **1.9%**. Επίσης το **%Σφάλμα** για την μέτρηση Θερμοκρασίας κυμάνθηκε από **0-2%** ενώ για την υγρασία από **0.5-1.5%**. Μπορούμε λοιπόν να συμπεράνουμε ότι ο αισθητήρας έχει εξαιρετική ακρίβεια και πιστότητα στις μετρήσεις του και θα μπορούσε να χρησιμοποιηθεί σε πραγματικές συνθήκες για την επιτήρηση της Θερμοκρασίας και της Υγρασίας ενός χώρου.

# 6. Συμπεράσματα & Μελλοντική Εργασία

## 6.1 Συμπεράσματα

Η δυνατότητα εξατομικευμένης επιτήρησης των ασθενών σε ένα νοσοκομείο είναι πολύ σημαντική διαδικασία, αφού με αυτόν τον τρόπο ο κάθε ασθενής δέχεται την καλύτερη δυνατή ποιότητα υπηρεσιών καθώς και άμεση φροντίδα σε περίπτωση έκτακτης ανάγκης. Παρόλα αυτά, η εξατομικευμένη επιτήρηση δεν είναι πάντα εύκολη λόγω του μειωμένου νοσηλευτικού και ιατρικού προσωπικού στα νοσοκομεία αλλά και του μεγάλου κόστους που απαιτείται για την εγκατάσταση ενός Κεντρικού Συστήματος Ενημέρωσης. Στην παρούσα εργασία έγινε μια προσπάθεια για την δημιουργία ενός συστήματος επιτήρησης ζωτικών παραμέτρων και συνθηκών χώρου, χαμηλού κόστους.

Μετά την ολοκλήρωση της εργασίας και τη σωστή λειτουργία του συστήματος εντοπίστηκαν κάποια θετικά όπως και αρνητικά χαρακτηριστικά.

### Θετικά

1. Το κόστος για την υλοποίηση της εργασίας είναι αρκετά χαμηλό και πιο συγκεκριμένα κάτω των 60€.
2. Απαιτούνται μόνο κάποιες απλές γνώσεις ηλεκτρονικών και συγκεκριμένα η συνδεσμολογία μεταξύ του ESP8266 και των τριών αισθητήρων. Επίσης η χρήση κολλητηριού είναι σε πολύ βασικό επίπεδο.
3. Υπάρχει μεγάλος αριθμός βιβλιογραφίας η οποία μπορεί να σε καθοδηγήσει σε πολλά σημεία προγραμματισμού αλλά και υλοποίησης.
4. Η υλοποίηση της εργασίας περιλαμβάνει τον προγραμματισμό της βάσης δεδομένων, του μικροελεγκτή αλλά και ιστοσελίδας. Με αυτό τον τρόπο κάποιος αποκτά γνώσεις σε ένα ευρύ φάσμα γλωσσών προγραμματισμού.
5. Διατηρώντας τον πυρήνα της εργασίας και τροποποιώντας μόνο κάποιον αισθητήρα, το σύστημα μπορεί να χρησιμοποιηθεί σε πολλούς τομείς.

### Αρνητικά

1. Απαιτείται καλή γνώση προγραμματισμού

2. Ο αισθητήρας MAX30102 δεν είναι εύκολος στην χρήση και δεν έχει μεγάλη αξιοπιστία.

## 6.2 Μελλοντικές Επεκτάσεις

- Με την χρήση ενός πιο αξιόπιστου αισθητήρα Οξυμετρίας αλλά και αισθητήρων για την παρακολούθηση περισσότερων ζωτικών παραμέτρων, το σύστημα μπορεί να χρησιμοποιηθεί για την κατ' οίκων επιτήρηση ασθενών.
- Εισαγωγή γραφημάτων στην real-time παρακολούθηση των παραμέτρων θα κάνει το σύστημα ακόμα πιο εύκολο στην κατανόηση των αποτελεσμάτων.
- Στην ιστοσελίδα απεικόνισης και φιλτραρίσματος μπορούν να τοποθετηθούν Drop-Down κουμπιά για την επιλογή μεμονωμένων παραμέτρων.
- Τροποποίηση του κώδικα του ESP8266 ώστε να μπορεί να ελέγχεται από οπουδήποτε και όχι μόνο όταν βρίσκεται κάποιος στο ίδιο WiFi δίκτυο.

# Βιβλιογραφία

[1] John C. Marshall, Laura Bosco, Neill K. Adhikari, Bronwen Connolly, Janet V. Diaz, Todd Dorman, Robert A. Fowler, Geert Meyfroidt, Satoshi Nakagawa, Paolo Pelosi, Jean-Louis Vincent, Kathleen Vollman, Janice Zimmerman, “What is an intensive care unit? A report of the task force of the World Federation of Societies of Intensive and Critical Care Medicine”, Journal of Critical Care, Volume 37, 2017, Pages 270-276, ISSN 0883-9441, <https://doi.org/10.1016/j.jcrc.2016.07.015>.

[2] Chidile Muonwe, The Intensive Care Unit, Physiopedia, [https://www.physio-pedia.com/The Intensive Care Unit](https://www.physio-pedia.com/The_Intensive_Care_Unit)

[3] Μονάδα Εντατικής Θεραπείας Ενηλίκων (ΜΕΘ), Νοσοκομείο Μητέρα, <https://www.mitera.gr/department/polydynami-monada-entatikis-therapeias-enilikon-m-e-th/>

[4] Neonatal intensive care unit (NICU), <https://www.pregnancybirthbaby.org.au/neonatal-intensive-care-unit>

[5] Lucinda Hampton, Neonatal Intensive Care Unit (NICU), Physiopedia, [https://www.physio-pedia.com/Neonatal Intensive Care Unit \(NICU\)](https://www.physio-pedia.com/Neonatal_Intensive_Care_Unit_(NICU))

[6] <https://www.mitera.gr/department/monada-entatikis-nosileias-neognon-m-e-n-n/>

[7] Dr. Janine Zee-Cheng, What is the Pediatric Intensive Care Unit? An introduction (Aug 27, 2018), <https://medicine.iu.edu/blogs/pediatrics/what-is-the-pediatric-intensive-care-unit-an-introduction>

[8] <https://www.mitera.gr/department/paidokardiocheiroyrgiki-meth/>

[9] Rohrig SAH, Lance MD, Faisal Malmstrom M. Surgical intensive care - current and future challenges? Qatar Med J. 2020 Jan 13;2019(2):3. doi: 10.5339/qmj.2019.qccc.3. PMID: 31976309; PMCID: PMC6958059, <https://pubmed.ncbi.nlm.nih.gov/31976309/>

[10] The ICU: life-saving care during COVID-19 and every day (December 7, 2020) <https://health.ucdavis.edu/coronavirus/news/headlines/the-icu-life-saving-care-during-covid-19-and-every-day-/2020/12>

- [11] Ken Harris, What is the ICU and why do COVID-19 patients end up there?( January 7, 2021), <https://www.osfhealthcare.org/blog/what-is-the-icu-and-why-do-covid-19-patients-end-up-there/>
- [12] Chatburn RL, Mireles-Cabodevila E. Chapter 3. Basic Principles of Ventilator Design. In: Tobin MJ. eds. *Principles and Practice of Mechanical Ventilation*, 3e. McGraw Hill; 2013. Accessed September 30, 2022. <https://accessmedicine.mhmedical.com/content.aspx?bookid=520&sectionid=41692239>
- [13] Zimmerman D, & Junker C (2014). Mechanical ventilation: principles of action. Freeman B.S., & Berger J.S.(Eds.), *Anesthesiology Core Review: Part One Basic Exam*. McGrawHill. <https://accessanesthesiology.mhmedical.com/content.aspx?bookid=974&sectionid=61587243><https://accessanesthesiology.mhmedical.com/content.aspx?sectionid=61587243&bookid=974>
- [14] Basic Principles of Ventilator Design, <https://aneskey.com/basic-principles-of-ventilator-design/>
- [15] Motta, Daniel & Amaral, Luiz & Silva, Bruno & Gomes, Lucas & Barbosa, Willams & Coelho, Rodrigo & Machado, Bruna. (2021). Collaborative and Structured Network for Maintenance of Mechanical Ventilators during the SARS-CoV-2 Pandemic. *Healthcare*. 9. 754. 10.3390/healthcare9060754. [https://www.researchgate.net/publication/352526625\\_Collaborative\\_and\\_Structured\\_Network\\_for\\_Maintenance\\_of\\_Mechanical\\_Ventilators\\_during\\_the\\_SARS-CoV-2\\_Pandemic](https://www.researchgate.net/publication/352526625_Collaborative_and_Structured_Network_for_Maintenance_of_Mechanical_Ventilators_during_the_SARS-CoV-2_Pandemic)
- [16]<https://en.wikipedia.org/wiki/Ventilator>
- [17] Lucie Hudson, What is the Trendelenburg position and why do we use it?(8 November 2016), <https://www.innovacareconcepts.com/en/blog/what-is-the-trendelenburg-position-and-why-do-we-use-it/>
- [18] [Leanne Godfrey](#), **Are ICU beds different from other hospital beds? Specifications & Features(16 April 2020)**, <https://www.innovacareconcepts.com/en/blog/icu-beds-vs-hospital-beds-specifications/>
- [19]<https://bmet.ewh.org/bitstream/handle/20.500.12091/178/Infant%20Incubators%2003.pdf?sequence=1&isAllowed=y>
- [20] Βασίλειος Σπυρόπουλος, Εισαγωγή στην Τεχνολογία Χειρουργείου, Εντατικής και Επείγουσας Ιατρικής.pdf
- [21] A Rajalakshmi et al 2019 J. Phys.: Conf. Ser. 1362 012128, “A Survey on Neonatal Incubator Monitoring System”



- [22] Frank Weithöner, Infant Incubators, [http://www.frankshospitalworkshop.com/equipment/infant\\_incubators\\_equipment.html](http://www.frankshospitalworkshop.com/equipment/infant_incubators_equipment.html)
- [23] Kelly T. Shannon, Sivam Ramanathan, Chapter 132 - Humidifiers, Editor(s): John L. Atlee, Complications in Anesthesia (Second Edition), W.B. Saunders, 2007, Pages 543-545, ISBN 9781416022152, <https://doi.org/10.1016/B978-1-4160-2215-2.50137-X>, <https://www.sciencedirect.com/science/article/pii/B978141602215250137X>
- [24] Ricard, Jean-Damien & LE MIÈRE, ERIC & Markowicz, Philippe & LASRY, SERGE & SAUMON, GEORGES & DJEDAÏNI, KAMEL & COSTE, FRANÇOIS & Dreyfuss, Didier. (2000). Efficiency and Safety of Mechanical Ventilation with a Heat and Moisture Exchanger Changed Only Once a Week. American journal of respiratory and critical care medicine. 161. 104-9. 10.1164/ajrccm.161.1.9902062.
- [25] CHRISTIAN LARSSON DAVID YANG, Humidification of Inspiration Gas in a Ventilator, <https://www.diva-portal.org/smash/get/diva2:542303/fulltext01.pdf>
- [26] Haitham S. Al Ashry, Ariel M. Modrykamien, "Humidification during Mechanical Ventilation in the Adult Patient", BioMed Research International, vol. 2014, Article ID 715434, 12 pages, 2014. <https://doi.org/10.1155/2014/715434>
- [27] MR850 Technical Manual 185041713 Rev J, Issued November 2005, Ref. 185041713, [http://www.frankshospitalworkshop.com/equipment/documents/ventilators/service\\_manuals/Fisher\\_Paykel\\_Humidifiers\\_MR-850\\_-\\_Service\\_manual.pdf](http://www.frankshospitalworkshop.com/equipment/documents/ventilators/service_manuals/Fisher_Paykel_Humidifiers_MR-850_-_Service_manual.pdf)
- [28] ΣΗΜΕΙΩΣΕΙΣ ΠΑΡΑΚΛΙΝΙΩΝ ΜΟΝΙΤΟΡ ΑΣΘΕΝΟΥΣ Α. ΤΖΑΒΑΡΑΣ
- [29] [https://www.celyontecnica.es/var/celyon-293-LIT284\\_NIBP\\_White\\_Paper\\_V5\\_-\\_final.pdf](https://www.celyontecnica.es/var/celyon-293-LIT284_NIBP_White_Paper_V5_-_final.pdf)
- [30] **Andrew Clay, Fluke Biomedical.** Understanding SpO2 Sensor Testing, <https://24x7mag.com/medical-equipment/testing-calibration/general-testing-equipment/understanding-spo2-sensor-testing/>
- [31] Santiago Lopez, Pulse Oximeter Fundamentals and Design, Freescale Semiconductor Application Note, Document Number: AN4327 Rev. 2, 11/2012
- [32] Shakhreet, B. & Shakhreet, M.. (2015). DESIGN OF PULSE OXIMETRY SYSTEM. Journal of Basic and Applied Research International. 7. 90 - 100. [https://www.researchgate.net/publication/275153148\\_DESIGN\\_OF\\_PULSE\\_OXIMETRY\\_SYSTEM](https://www.researchgate.net/publication/275153148_DESIGN_OF_PULSE_OXIMETRY_SYSTEM)
- [33] Robert Finnerty, How to Design a Better Pulse Oximeter, <https://www.analog.com/en/technical-articles/how-to-design-a-better-pulse-oximeter.html>

- [34] Nathan M Kesto, Electrocardiography Circuit Design 4/5/2013, <https://www.egr.msu.edu/classes/ece480/capstone/spring13/group03/documents/ElectrocardiographyCircuitDesign.pdf>
- [35] Amit K. Gupta, Respiration Rate Measurement Based on Impedance Pneumography, Application Report SBAA181–February 2011, Texas Instruments, [https://www.ti.com/lit/an/sbaa181/sbaa181.pdf?ts=1658830828184&ref\\_url=https%252F%252Fwww.google.pt%252F](https://www.ti.com/lit/an/sbaa181/sbaa181.pdf?ts=1658830828184&ref_url=https%252F%252Fwww.google.pt%252F)
- [36] Δρ. Α. Τζαβάρας, Η ΒΙΤ της κατ' οίκον Επιτήρησης Υγείας και Νοσηλείας και Διαδικτυακή Διαχείριση Παροχής Υπηρεσιών Υγείας, Ενότητα 1.5: Παρακολούθηση Ζωτικών Παραμέτρων, <http://medisp.bme.teiath.gr/eclass/courses/>
- [37] Samuel M. Galvagno Jr., D.O. Bhavani Shankar Kodali, M.D., Capnography in Intensive Care Unit <https://www.capnography.com/icu/capnography-in-icu>
- [38] <https://www.myamericannurse.com/understanding-end-tidal-co2-monitoring/>
- [39] How Is CO2 Monitored? Your Complete Carbon Dioxide Guide <https://infiniummedical.com/how-is-co2-monitored-your-complete-carbon-dioxide-guide/>
- [40] EDAN MFM-CMS Central Monitoring System User Manual
- [41] SYED MUHAMMAD YASIR JAFRI, IP BASED PATIENT MONITORING SYSTEMS, University of Boras, <https://hb.diva-portal.org/smash/get/diva2:1311731/FULLTEXT01>
- [42] <https://www.draeger.com/Library/Content/vista-120-cms-pi-9068120-en-master.pdf>
- [43] [https://eu.nihonkohden.com/en/products/patientmonitoring/central\\_nursing\\_station-cns9101.html](https://eu.nihonkohden.com/en/products/patientmonitoring/central_nursing_station-cns9101.html)
- [44] <https://nkuniversity.org/covid-19/wp-content/uploads/2020/01/nk-healthprotect-pop-up-icu-solution-setup-manual.pdf>
- [45] Stefan, NodeMCU, ESP12, ESP8266 - What is the difference?( Sep 19, 2021) <https://blog.spacehuhn.com/nodemcu-vs-esp8266>
- [46] <https://lastminuteengineers.com/esp8266-nodemcu-arduino-tutorial/>
- [47] [https://annefou.github.io/loT\\_introduction/02-ESP8266/index.html](https://annefou.github.io/loT_introduction/02-ESP8266/index.html)
- [48] Νούσης Βασίλης, Ηγουμενίτσα 2019, Arduino για αρχάριους με έμφαση στη χρήση του στο εργαστήριο Φυσικών Επιστημών <http://3gym-igoum.thesp.sch.gr/images/Arduino/book.pdf>
- [49] <https://microcontrollerslab.com/uart-communication-working-applications/>

- [50] <https://microcontrollerslab.com/introduction-to-spi-communication-protocol/>
- [51] Κόχυλας Δημήτρης - Τσάλτας Θωδωρής, “Χειρισμός Ρομποτικού Βραχίονα Μέσω Κάμερας Kinect”
- [52] <https://robocraze.com/blogs/post/arduino-vs-nodemcu>
- [53] <https://developer.electricimp.com/resources/i2c>
- [54] <https://lastminuteengineers.com/ds18b20-arduino-tutorial/>
- [55] <https://www.apogeeweb.net/circuitry/ds18b20-working-principle.html>
- [56] Κορίτσογλου Κυριάκος, Μια απλή, χαμηλού κόστους μέθοδος για τη βελτίωση της ακρίβειας των ψηφιακών αισθητήρων θερμοκρασίας DS18B20 με τη χρήση γραμμικής παλινδρόμησης., [https://apothesis.eap.gr/bitstream/repo/45881/1/90443\\_%ce%9a%ce%9f%ce%a1%ce%99%ce%a4%ce%a3%ce%9f%ce%93%ce%9b%ce%9f%ce%a5\\_%ce%9a%ce%a5%ce%a1%ce%99%ce%91%ce%9a%ce%9f%ce%a3.pdf](https://apothesis.eap.gr/bitstream/repo/45881/1/90443_%ce%9a%ce%9f%ce%a1%ce%99%ce%a4%ce%a3%ce%9f%ce%93%ce%9b%ce%9f%ce%a5_%ce%9a%ce%a5%ce%a1%ce%99%ce%91%ce%9a%ce%9f%ce%a3.pdf)
- [57] Zahid Ali, Introduction to DS18B20 (25/01/2019), <https://www.theengineeringprojects.com/2019/01/introduction-to-ds18b20.html>
- [58] Fezari, Mohamed & Al Dahoud, Ali. (2019). Exploring One-wire Temperature sensor "DS18B20" with Microcontrollers.
- [59] <https://www.elprocus.com/ds18b20-temperature-sensor/>
- [60] <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>
- [61] Debashis Das, How Does a DS18b20 Temperature Sensor Work and how to Interface it with Arduino?(April 25, 2022), <https://circuitdigest.com/microcontroller-projects/interfacing-ds18b20-sensor-with-arduino>
- [62] <https://www.best-microcontroller-projects.com/ds18b20.html>
- [63] <https://lastminuteengineers.com/dht11-dht22-arduino-tutorial/>
- [64] Roland Pelayo, How the DHT22 Sensor Works, <https://www.teachmemicro.com/how-dht22-sensor-works/>
- [65] Dejan, DHT11 & DHT22 Sensors Temperature and Humidity Tutorial using Arduino, <https://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/>
- [66] FABIO NELLI, DHT11 and DHT22 sensors – Measure humidity and temperature with Arduino 12 OCTOBER 2020 , <https://www.meccanismocomplesso.org/en/dht11-and-dht22-sensors-measure-humidity-and-temperature-with-arduino/>
- [67] DHT22 Temperature-Humidity Sensor, [https://www.waveshare.com/wiki/DHT22\\_Temperature-Humidity\\_Sensor](https://www.waveshare.com/wiki/DHT22_Temperature-Humidity_Sensor)

- [68] DHT22 – Temperature and Humidity Sensor  
<https://components101.com/sensors/dht22-pinout-specs-datasheet>
- [69] <https://www.mouser.com/datasheet/2/737/dht-932870.pdf>
- [70] Thomas Liu, Digital relative humidity & temperature sensor AM2302/DHT22,  
<https://cdn-shop.adafruit.com/datasheets/Digital+humidity+and+temperature+sensor+AM2302.pdf>
- [71]<https://lastminuteengineers.com/max30102-pulse-oximeter-heart-rate-sensor-arduino-tutorial/>
- [72] <https://www.utmel.com/components/max30102-sensor-datasheet-pinout-and-schematic?id=346>
- [73] Debashis Das, How MAX30102 Pulse Oximeter and Heart Rate Sensor Works and how to Interface it with Arduino?( May 4, 2022),  
<https://circuitdigest.com/microcontroller-projects/how-max30102-pulse-oximeter-and-heart-rate-sensor-works-and-how-to-interface-with-arduino>
- [74]<https://pdfserv.maximintegrated.com/en/an/AN6409.pdf>
- [75] Radwa Sameh et al 2020 J. Phys.: Conf. Ser. 1447 012004, Design and Implementation of an SPO2 Based Sensor for Heart Monitoring Using an Android Application,  
<https://iopscience.iop.org/article/10.1088/1742-6596/1447/1/012004/pdf>
- [76] <https://www.arduino.cc/>
- [77]<https://examples.blynk.cc/?board=ESP8266&shield=ESP8266%20WiFi&example=GettingStarted%2FVirtualPinRead>
- [78]<https://github.com/blynkkk/blynk-library/releases/tag/v1.0.1>
- [79]<https://docs.blynk.io/en/getting-started/what-do-i-need-to-blynk/how-quickstart-device-was-made>
- [80]<https://docs.blynk.io/en/getting-started/template-quick-setup/set-up-web-dashboard>
- [81]<https://forum.arduino.cc/t/pyserial-and-esptools-directory-error/671804/5>
- [82] <https://blynk.io/blog/how-to-get-started-with-blynk>
- [83]<https://mytectutor.com/how-to-connect-esp8266-to-mysql-database-using-php-and-arduino-ide/>
- [84]<https://randomnerdtutorials.com/esp32-esp8266-mysql-database-php/>
- [85] [https://georgochr.sites.sch.gr/?page\\_id=2087](https://georgochr.sites.sch.gr/?page_id=2087)

- [86] Alsan Parajuli, Insert Data into MySQL Database with ESP8266 Development Board <https://theiotprojects.com/insert-data-into-mysql-database-with-esp8266/>
- [87] <https://how2electronics.com/iot-based-patient-health-monitoring-esp32-web-server/>
- [88]<https://lastminuteengineers.com/esp8266-dht11-dht22-web-server-tutorial/>
- [89]<https://docs.blynk.io/en/blynk.edgent-firmware-api/installation/what-do-i-need-to-blynk>
- [90]Marcelo Rovai, IoT Made Simple: Monitoring Multiple Sensors(August 25, 2017), <https://www.hackster.io/mjrobot/iot-made-simple-monitoring-multiple-sensors-db887f>
- [91]<https://github.com/RuiSantosdotme/ESP32-ESP8266-PHP-MySQL/blob/master/code/post-esp-data.php>
- [92] Sahil Kazi, Log Data into MySQL Database using NodeMCU Development Board(April 2, 2021), <https://circuitdigest.com/microcontroller-projects/nodemcu-mysql-data-logging>
- [93]<https://dev.mysql.com/doc/refman/8.0/en/timestamp-initialization.html>
- [94] Muhammad Aqib, Logging Data to Database Using Arduino Ethernet Shield(December 18, 2018), [https://create.arduino.cc/projecthub/muhammad-aqib/logging-data-to-database-using-arduino-ethernet-shield-3e9a0e?ref=tag&ref\\_id=database&offset=3](https://create.arduino.cc/projecthub/muhammad-aqib/logging-data-to-database-using-arduino-ethernet-shield-3e9a0e?ref=tag&ref_id=database&offset=3)
- [95]<https://stackoverflow.com/questions/27183416/what-exactly-does-the-php-function-test-input-do>
- [96]<https://stackoverflow.com/questions/5471561/what-is-this-isset-post-code-doing>

# Κώδικας

## Κώδικας Arduino

```
#define BLYNK_TEMPLATE_ID "TMPLMtb77IUK"
#define BLYNK_DEVICE_NAME "Patient Monitoring"
#define BLYNK_AUTH_TOKEN "ecTc4No40KTicxRqFDnA6EIE2xRb_GOc"
#define ONE_WIRE_BUS D7
//#define DHTTYPE DHT11 // DHT 11
#define DHTTYPE DHT22 // dht 22

#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <OneWire.h>
#include <ESP8266WebServer.h>
#include <DallasTemperature.h>
#include <DFRobot_MAX30102.h>
#include "DHT.h"
#include <BlynkSimpleEsp8266.h>

// Your WiFi credentials.
char ssid[] = "TP-Link_F98F"; // Enter SSID here
char password[] = "tplink24GHz!!"; //Enter Password here

//Blynk app authorisation token
char auth[] = BLYNK_AUTH_TOKEN;

//MySQL DataBase info
```

```

const char* serverName = "http://192.168.0.104/Sensor-data-test/post-sensor-data.php";
String apiKeyValue = "123";
// Example: tPmAT5Ab3j7F9
String sensorName = "DHT11";
String sensorLocation = "Hospital";

ESP8266WebServer server(80);
DFRobot_MAX30102 particleSensor;
OneWire oneWire(ONE_WIRE_BUS); // Setup a oneWire instance to communicate with any OneWire
device
DallasTemperature sensors(&oneWire); // Pass oneWire reference to DallasTemperature library
uint8_t DHTPin = D5; // DHT Sensor
DHT dht(DHTPin, DHTTYPE); // Initialize DHT sensor.
BlynkTimer timer;

int32_t SPO2; //SPO2
int8_t SPO2Valid; //Flag to display if SPO2 calculation is valid
int32_t heartRate; //Heart-rate
int8_t heartRateValid; //Flag to display if heart-rate calculation is valid

float Temperature, Humidity, tempC;
String val;

void setup() {

sensors.begin();

Serial.begin(115200);

sensors.setResolution(10); /*Το αρχικο resolution του αισθητηρα ειναι 12-bit δηλαδη ακριβεια
0.0625°C.
Αυτο καθυστερει αρκετα την απεικονιση της πραγματικης τιμης. Ετσι το αλλαζουμε σε 9-bit και
ακριβεια 0.5 βαθμους
ωστε να βλεπουμε πιο γρηγορα την πραγματικη τιμη*/

while (!particleSensor.begin()) {

```

```

Serial.println("MAX30102 was not found");
delay(1000);
}

particleSensor.sensorConfiguration(/*ledBrightness=*/70, /*sampleAverage=*/SAMPLEAVG_16, \
/*ledMode=*/MODE_MULTILED, /*sampleRate=*/SAMPLERATE_400,
/*pulseWidth=*/PULSEWIDTH_118, /*adcRange=*/ADCRANGE_16384);

pinMode(DHTPin, INPUT);

sensors.begin();

dht.begin();

Serial.println("Connecting to ");
Serial.println(ssid);

//connect to your local wi-fi network
WiFi.begin(ssid, password);
Blynk.config(auth);

//check wi-fi is connected to wi-fi network
while (WiFi.status() != WL_CONNECTED) {
delay(1000);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected..!");
Serial.print("Got IP: "); Serial.println(WiFi.localIP());

server.on("/", handle_OnConnect);
server.onNotFound(handle_NotFound);

server.begin();
Serial.println("HTTP server started");

```



```

// Setup a function to be called every second
timer.setInterval(1000L, sendSensor);
}

void loop(void) {
values();
server.handleClient();
sql();

Blynk.run();
timer.run(); // Initiates BlynkTimer
sendSensor();
delay(1000);

}

void values() {
sensors.requestTemperatures();
tempC = sensors.getTempCByIndex(0);
particleSensor.heartRateAndOxygenSaturation(**SPO2=*)&SPO2, /**SPO2Valid=*)&SPO2Valid,
/**heartRate=*)&heartRate, /**heartRateValid=*)&heartRateValid);
if (SPO2 < 0) {
SPO2 = -1;
}
if (heartRate < 0) {
heartRate = -1;
}
if (heartRate < 100 && heartRate > 60) {
val = "SpO2 sensor is working properly";
}
else {
val = "Warning!! Check the sensor & The patient";
}
//Serial.println(SPO2);
//Serial.println(heartRate);

```

```

Temperature = dht.readTemperature(); // Gets the values of the temperature
Humidity = dht.readHumidity(); // Gets the values o the humidity
delay(200);
}
void sendSensor() {
// You can send any value at any time.
// Please don't send more that 10 values per second.
Blynk.virtualWrite(V5, Temperature);
Blynk.virtualWrite(V6, Humidity);
Blynk.virtualWrite(V7, SPO2);
Blynk.virtualWrite(V8, heartRate);
Blynk.virtualWrite(V9, tempC);
}
void sql() {
WiFiClient client;
HTTPClient http;
http.begin(client, serverName);
http.addHeader("Content-Type", "application/x-www-form-urlencoded");
String httpRequestData = "api_key=" + apiKeyValue + "&location=" + sensorLocation +
"&Room_Temp=" + String(Temperature) + "&humidity=" + String(Humidity) + "&SpO2=" + String(SPO2)
+ "&BPM=" + String(heartRate) + "&Body_Temp=" + String(tempC) + "";
//String      httpRequestData      =      "api_key=theiotprojects&sensor=HC-
SR04&location=Home&distance=24.75";
Serial.print("httpRequestData: ");
Serial.println(httpRequestData);
// Send HTTP POST request
int httpResponseCode = http.POST(httpRequestData);

if (httpResponseCode > 0) {
Serial.print("HTTP Response code: ");
Serial.println(httpResponseCode);
}
else {
Serial.print("Error code: ");
Serial.println(httpResponseCode);
}
}

```

```

// Free resources
http.end();
delay(200);
}
void handle_OnConnect() {
server.send(200, "text/html", SendHTML(Temperature, Humidity, heartRate, SPO2, tempC, val));
}
void handle_NotFound() {
server.send(404, "text/plain", "Not found");
}
String SendHTML(float TempCstat, float Humiditystat, float BPM, float SpO2, float tempC, String val) {
String ptr = "<!DOCTYPE html> <html>\n";
ptr += "<head><meta http-equiv=\"refresh\" content=\"3\"> <meta name=\"viewport\"
content=\"width=device-width, initial-scale=1.0, user-scalable=no\">\n";
ptr += "<link href=\"https://fonts.googleapis.com/css?family=Open+Sans:300,400,600\"
rel=\"stylesheet\">\n";
ptr += "<title>Patient Monitoring</title>\n";
ptr += "<style>html { font-family: 'Open Sans', sans-serif; display: block; margin: 0px auto; text-align:
center;color: #333333;}\n";
ptr += "body{margin-top: 50px;}\n";
ptr += "h1 {margin: 50px auto 30px;}\n";
ptr += ".side-by-side{display: inline-block;vertical-align: middle;position: relative;}\n";
ptr += ".humidity-icon{background-color: #3498db;width: 41px;height: 41px;border-radius: 50%;line-
height: 44px;}\n";
ptr += ".humidity-text{font-weight: 600;padding-left: 15px;font-size: 19px;width: 160px;text-align:
left;}\n";
ptr += ".humidity{font-weight: 300;font-size: 60px;color: #3498db;}\n";
ptr += ".temperature-icon{background-color: #f39c12;width: 41px;height: 41px;border-radius:
50%;line-height: 44px;}\n";
ptr += ".temperature-text{font-weight: 600;padding-left: 15px;font-size: 19px;width: 160px;text-align:
left;}\n";
ptr += ".temperature{font-weight: 300;font-size: 60px;color: #f39c12;}\n";
ptr += ".bpm-icon{background-color: #3498db;width: 41px;height: 41px;border-radius: 50%;line-
height: 20px;}\n";
ptr += ".bpm-text{font-weight: 600;padding-left: 15px;font-size: 19px;width: 160px;text-align: left;}\n";
ptr += ".bpm{font-weight: 150;font-size: 60px;color: #d932d3;}\n";
//ptr += ".bo-icon{background-color: #3498db;width: 41px;height: 41px;line-height: 20px;}\n";

```

```

ptr += ".bo-text{font-weight: 600;padding-left: 15px;font-size: 19px;width: 160px;text-align: left;}\n";
ptr += ".bo{font-weight: 300;font-size: 60px;color: #59d62f;}\n";
//ptr += ".bt-icon{background-color: #3498db;width: 41px;height: 41px;border-radius: 50%;line-height: 44px;}\n";
ptr += ".bt-text{font-weight: 600;padding-left: 15px;font-size: 19px;width: 160px;text-align: left;}\n";
ptr += ".bt{font-weight: 300;font-size: 60px;color: #e38c1b;}\n";
ptr += ".superscript{font-size: 17px;font-weight: 600;position: absolute;right: -50px;top: 15px;}\n";
ptr += ".data{padding: 10px;}\n";
ptr += "</style>\n";
ptr += "</head>\n";
ptr += "<body>\n";

ptr += "<div id=\"webpage\">\n";

ptr += "<h1>Patient Monitoring</h1>\n";

ptr += "<div class=\"data\">\n";
ptr += "<div class=\"side-by-side temperature-icon\">\n";
ptr += "<svg version=\"1.1\" id=\"Layer_1\" xmlns=\"http://www.w3.org/2000/svg\"
xmlns:xlink=\"http://www.w3.org/1999/xlink\" x=\"0px\" y=\"0px\"
ptr += \"width=\"9.915px\" height=\"22px\" viewBox=\"0 0 9.915 22\" enable-background=\"new 0 0 9.915 22\" xml:space=\"preserve\">\n";
ptr += "<path fill=\"#FFFFFF\" d=\"M3.498,0.53c0.377-0.331,0.877-0.501,1.374-0.527C5.697-0.04,6.522,0.421,6.924,1.142\n";
ptr += "c0.237,0.399,0.315,0.871,0.311,1.33C7.229,5.856,7.245,9.24,7.227,12.625c1.019,0.539,1.855,1.424,2.301,2.491\n";
ptr += "c0.491,1.163,0.518,2.514,0.062,3.693c-0.414,1.102-1.24,2.038-2.276,2.594c-1.056,0.583-2.331,0.743-3.501,0.463\n";
ptr += "c-1.417-0.323-2.659-1.314-3.3-2.617C0.014,18.26-0.115,17.104,0.1,16.022c0.296-1.443,1.274-2.717,2.58-3.394\n";
ptr += "c0.013-3.44,0-6.881,0.007-10.322C2.674,1.634,2.974,0.955,3.498,0.53z\"/>\n";
ptr += "</svg>\n";
ptr += "</div>\n";
ptr += "<div class=\"side-by-side temperature-text\">Temperature</div>\n";
ptr += "<div class=\"side-by-side temperature\">\n";
ptr += (int)TempCstat;

```

```

ptr += "<span class=\"superscript\">&#8451;</span></div>\n";
ptr += "</div>\n";

ptr += "<div class=\"data\">\n";
ptr += "<div class=\"side-by-side humidity-icon\">\n";
ptr += "<svg version=\"1.1\" id=\"Layer_2\" xmlns=\"http://www.w3.org/2000/svg\"
xmlns:xlink=\"http://www.w3.org/1999/xlink\" x=\"0px\" y=\"0px\" width=\"12px\"
height=\"17.955px\" viewBox=\"0 0 13 17.955\" enable-background=\"new 0 0 13 17.955\"
xml:space=\"preserve\">\n";
ptr += "<path fill=\"#FFFFFF\"
d=\"M1.819,6.217C3.139,4.064,6.5,0,6.5,0s3.363,4.064,4.681,6.217c1.793,2.926,2.133,5.05,1.571,7.0
57\n";
ptr += "c-0.438,1.574-2.264,4.681-6.252,4.681c-3.988,0-5.813-3.107-6.252-4.681C-
0.313,11.267,0.026,9.143,1.819,6.217\"></path>\n";
ptr += "</svg>\n";
ptr += "</div>\n";
ptr += "<div class=\"side-by-side humidity-text\">Humidity</div>\n";
ptr += "<div class=\"side-by-side humidity\">";
ptr += (int)Humiditystat;
ptr += "<span class=\"superscript\">%</span></div>\n";
ptr += "</div>\n";

ptr += "<div class='data Heart Rate'>";
ptr += "<div class='side-by-side bpm-icon'>";
ptr += "<svg enable-background='new 0 0 40.542 40.541'height=40.541px id=Layer_1 version=1.1
viewBox='0 0 40.542 40.541'width=40.542px x=0px xml:space=preserve
xmlns=http://www.w3.org/2000/svg xmlns:xlink=http://www.w3.org/1999/xlink y=0px><g><path
d='M34.313,20.271c0-0.552,0.447-1,1-1h5.178c-0.236-4.841-2.163-9.228-5.214-12.593l-
3.425,3.424\";
ptr += "c-0.195,0.195-0.451,0.293-0.707,0.293s-0.512-0.098-0.707-0.293c-0.391-0.391-0.391-1.023,0-
1.414l3.425-3.424\";
ptr += "c-3.375-3.059-7.776-4.987-12.634-5.215c0.015,0.067,0.041,0.13,0.041,0.202v4.687c0,0.552-
0.447,1-1,1s-1-0.448-1-1v0.25\";
ptr += "c0-0.071,0.026-0.134,0.041-
0.202C14.39,0.279,9.936,2.256,6.544,5.385l3.576,3.577c0.391,0.391,0.391,1.024,0,1.414\";
ptr += "c-0.195,0.195-0.451,0.293-0.707,0.293s-0.512-0.098-0.707-0.293L5.142,6.812c-2.98,3.348-
4.858,7.682-5.092,12.459h4.804\";
ptr += "c0.552,0,1,0.448,1,1s-0.448,1-
1,1H0.05c0.525,10.728,9.362,19.271,20.22,19.271c10.857,0,19.696-8.543,20.22-19.271h-5.178\";

```

```

ptr += "C34.76,21.271,34.313,20.823,34.313,20.271z M23.084,22.037c-0.559,1.561-2.274,2.372-
3.833,1.814";

ptr += "c-1.561-0.557-2.373-2.272-1.815-3.833c0.372-1.041,1.263-1.737,2.277-
1.928L25.2,7.202L22.497,19.05";

ptr += "C23.196,19.843,23.464,20.973,23.084,22.037z'fill=#26B999 /></g></svg>";

ptr += "</div>";

ptr += "<div class='side-by-side bpm-text'>Heart Rate</div>";

ptr += "<div class='side-by-side bpm'>";

ptr += (int)BPM;

ptr += "<span class='superscript'>BPM</span></div>";

ptr += "</div>";

ptr += "<div class='data Blood Oxygen'>";

ptr += "<div class='side-by-side bo-icon'>";

ptr += "<svg enable-background='new 0 0 58.422 40.639'height=40.639px id=Layer_1 version=1.1
viewBox='0 0 58.422 40.639'width=58.422px x=0px xml:space=preserve
xmlns=http://www.w3.org/2000/svg xmlns:xlink=http://www.w3.org/1999/xlink y=0px><g><path
d='M58.203,37.754l0.007-0.004L42.09,9.935l-0.001,0.001c-0.356-0.543-0.969-0.902-1.667-0.902";

ptr += "c-0.655,0-1.231,0.32-1.595,0.808l-0.011-0.007l-0.039,0.067c-0.021,0.03-0.035,0.063-
0.054,0.094L22.78,37.692l0.008,0.004";

ptr += "c-0.149,0.28-0.242,0.594-
0.242,0.934c0,1.102,0.894,1.995,1.994,1.995v0.015h31.888c1.101,0,1.994-0.893,1.994-1.994";

ptr += "C58.422,38.323,58.339,38.024,58.203,37.754z'fill=#955BA5 /><path d='M19.704,38.674l-
0.013-0.004l13.544-23.522L25.13,1.156l-0.002,0.001C24.671,0.459,23.885,0,22.985,0";

ptr += "c-0.84,0-1.582,0.41-2.051,1.038l-0.016-0.011L20.87,1.114c-0.025,0.039-0.046,0.082-
0.068,0.124L0.299,36.851l0.013,0.004";

ptr += "C0.117,37.215,0,37.62,0,38.059c0,1.412,1.147,2.565,2.565,2.565v0.015h16.989c-0.091-0.256-
0.149-0.526-0.149-0.813";

ptr += "C19.405,39.407,19.518,39.019,19.704,38.674z'fill=#955BA5 /></g></svg>";

ptr += "</div>";

ptr += "<div class='side-by-side bo-text'>Blood Oxygen</div>";

ptr += "<div class='side-by-side bo'>";

ptr += (int)SpO2;

ptr += "<span class='superscript'>%</span></div>";

ptr += "</div>";

ptr += "<div class='data Body Temp'>";

ptr += "<div class='side-by-side bt-icon'>";

```

```

ptr += "<svg enable-background='new 0 0 19.438 54.003'height=54.003px id=Layer_1 version=1.1
viewBox='0 0 19.438 54.003'width=19.438px x=0px xml:space=preserve
xmlns=http://www.w3.org/2000/svg xmlns:xlink=http://www.w3.org/1999/xlink y=0px><g><path
d='M11.976,8.82v-
2h4.084V6.063C16.06,2.715,13.345,0,9.996,0H9.313C5.965,0,3.252,2.715,3.252,6.063v30.982";

ptr += "C1.261,38.825,0,41.403,0,44.286c0,5.367,4.351,9.718,9.719,9.718c5.368,0,9.719-4.351,9.719-
9.718";

ptr += "c0-2.943-1.312-5.574-3.378-7.355V18.436h-3.914v-2h3.914v-2.808h-4.084v-
2h4.084V8.82H11.976z M15.302,44.833";

ptr += "c0,3.083-2.5,5.583-5.583,5.583s-5.583-2.5-5.583-5.583c0-2.279,1.368-4.236,3.326-
5.104V24.257C7.462,23.01,8.472,22,9.719,22";

ptr += "s2.257,1.01,2.257,2.257V39.73C13.934,40.597,15.302,42.554,15.302,44.833z'fill=#F29C21
/></g></svg>";

ptr += "</div>";

ptr += "<div class='side-by-side bt-text'>Body Temp</div>";

ptr += "<div class='side-by-side bt'>";

ptr += (float) tempC;

ptr += "<span class='superscript'>&#8451;</span></div>\n";

ptr += "</div>";

ptr += (String) val;
//ptr += "<h1>val</h1>\n";

ptr += "</div>\n";

ptr += "</body>\n";

ptr += "</html>\n";

return ptr;
}

```

## Κώδικας PHP (sensor-data.php)

```
<?php
$servername = "localhost";
// REPLACE with your Database name
$dbname = "sensor_db";
// REPLACE with Database user
$username = "admin";
// REPLACE with Database user password
$password = "1234";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$datefrom="";
if (isset($_POST['datefrom'])) {
    $datefrom=$_POST['datefrom'];
}
$dateto="";
if (isset($_POST['dateto'])) {
    $dateto=$_POST['dateto'];
}

$spo2from="";
if (isset($_POST['spo2from'])) {
    $spo2from=$_POST['spo2from'];
}
$spo2to="";
if (isset($_POST['spo2to'])) {
    $spo2to=$_POST['spo2to'];
}
```



```

$bpmfrom="";
if (isset($_POST['bpmfrom'])) {
    $bpmfrom=$_POST['bpmfrom'];
}
$bpmto="";
if (isset($_POST['bpmto'])) {
    $bpmto=$_POST['bpmto'];
}

$bodyfrom="";
if (isset($_POST['bodyfrom'])) {
    $bodyfrom=$_POST['bodyfrom'];
}

$bodyto="";
if (isset($_POST['bodyto'])) {
    $bodyto=$_POST['bodyto'];
}

$where = "";

$options=array();

if($datefrom!="")
{
    //$options[]="reading_time >='2022-04-12 12:00:00'";
    $options[]="reading_time >='".mysql_real_escape_string($conn,$datefrom)."'";
}

if($dateto!="")
{
    //$options[]="reading_time <='2022-04-12 12:00:00'";
    $options[]="reading_time <='".mysql_real_escape_string($conn,$dateto)."'";
}

```

```

if($spo2from!="")
{
    //$options[]="spo2 >=5";
    $options[]="spo2>=".mysql_real_escape_string($conn,$spo2from)."";
}

if($spo2to!="")
{
    //$options[]="spo2 <=5";
    $options[]="spo2<=".mysql_real_escape_string($conn,$spo2to)."";
}

if($bpmfrom!="")
{
    //$options[]="bpm >=5";
    $options[]="bpm>=".mysql_real_escape_string($conn,$bpmfrom)."";
}

if($bpmtto!="")
{
    //$options[]="bpm <=5";
    $options[]="bpm<=".mysql_real_escape_string($conn,$bpmtto)."";
}

if($bodyfrom!="")
{
    //$options[]="bodytemp >=5";
    $options[]="bodytemp>=".mysql_real_escape_string($conn,$bodyfrom)."";
}

if($bodyto!="")
{
    //$options[]="bodytemp <=5";
    $options[]="bodytemp<=".mysql_real_escape_string($conn,$bodyto)."";
}

```

```

if(count($options))
{
    $where="WHERE ".implode(" AND ", $options);
}
$sql = "SELECT id, location, roomtemp, roomhumid, spo2, bpm, bodytemp, reading_time FROM
SensorData $where ORDER BY id DESC";
$mytable = '<table width="50%" cellpadding="5" cellspacing="5">
<tr>
    <td><strong>A/A<strong></td>
    <td><strong>Location<strong></td>
    <td><strong>Room Temperature<strong></td>
    <td><strong>Room Humidity<strong></td>
    <td><strong>SpO2<strong></td>
    <td><strong>BPM<strong></td>
    <td><strong>Body Temperature<strong></td>
    <td><strong>Timestamp<strong></td>
</tr>;
if ($result = $conn->query($sql)) {
    while ($row = $result->fetch_assoc()) {
        $row_id = $row["id"];
        $row_location = $row["location"];
        $row_roomtemp = $row["roomtemp"];
        $row_roomhumid = $row["roomhumid"];
        $row_spo2 = $row["spo2"];
        $row_bpm = $row["bpm"];
        $row_bodytemp = $row["bodytemp"];
        $row_reading_time = $row["reading_time"];

        $mytable .= '<tr>
            <td>' . $row_id . '</td>
            <td>' . $row_location . '</td>
            <td>' . $row_roomtemp . '</td>
            <td>' . $row_roomhumid . '</td>
            <td>' . $row_spo2 . '</td>
            <td>' . $row_bpm . '</td>

```

```

        <td>' . $row_bodytemp . '</td>
        <td>' . $row_reading_time . '</td>
    </tr>';
}
$result->free();
}

$mytable .= '</table>';

$conn->close();
?><!DOCTYPE html>
<html>
<head>
<title>
    Patient Monitoring
</title>
<body>
<h1>Patient Monitoring</h1>
<form name="search_term" method="post" action="sensor-data.php">
<table border="0" cellpadding="2" cellspacing="2">
<tr>
    <td>Date:</td><td><input type="text" name="datefrom" value="" size="20"></td><td>-
</td><td><input type="text" name="dateto" value="" size="20"></td>
    <td><?php echo $datefrom; ?> - <?php echo $dateto; ?></td>
</tr>

<tr>
    <td>SpO2:</td><td><input type="text" name="spo2from" value="" size="20"></td><td>-
</td><td><input type="text" name="spo2to" value="" size="20"></td>
    <td><?php echo $spo2from; ?> - <?php echo $spo2to; ?></td>
</tr>

<tr>
    <td>BPM:</td><td><input type="text" name="bpmfrom" value="" size="20"></td><td>-
</td><td><input type="text" name="bpmtto" value="" size="20"></td>
    <td><?php echo $bpmfrom; ?> - <?php echo $bpmtto; ?></td>

```

```
</tr>
```

```
<tr>
```

```
    <td>Body    Temp:</td><td><input    type="text"    name="bodyfrom"    value=""  
size="20"></td><td><input type="text" name="bodyto" value="" size="20"></td>
```

```
    <td><?php echo $bodyfrom; ?> - <?php echo $bodyto; ?></td>
```

```
</tr>
```

```
</table><br>
```

```
<input type="submit" name="search" value="Search the table...">
```

```
<input type="reset" name="reset" value="Reset">
```

```
</form>
```

```
<?php echo $mytable; ?>
```

```
</body>
```

```
</html>
```

## Κώδικας PHP (post-sensor-data.php)

```
<?php
$servername = "localhost";
// REPLACE with your Database name
$dbname = "sensor_db";
// REPLACE with Database user
$username = "admin";
// REPLACE with Database user password
$password = "1234";
// Keep this API Key value to be compatible with the ESP8266 code provided in the project page.
// If you change this value, the ESP8266 sketch needs to match
$api_key_value = "123";
$api_key = $location = $roomtemp = $roomhumid = $spo2 = $bpm = $bodytemp = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $api_key = test_input($_POST["api_key"]);
    if($api_key == $api_key_value) {
        $location = test_input($_POST["location"]);
        $roomtemp = test_input($_POST["Room_Temp"]);
        $roomhumid = test_input($_POST["humidity"]);
        $spo2 = test_input($_POST["SpO2"]);
        $bpm = test_input($_POST["BPM"]);
        $bodytemp = test_input($_POST["Body_Temp"]);
        // Create connection
        $conn = new mysqli($servername, $username, $password, $dbname);
        // Check connection
        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }

        $sql = "INSERT INTO SensorData (location, roomtemp, roomhumid, spo2, bpm, bodytemp)
        VALUES ('" . $location . "', '" . $roomtemp . "', '" . $roomhumid . "', '" . $spo2 . "', '" . $bpm . "', '" .
        $bodytemp . "')";

        if ($conn->query($sql) === TRUE) {
            echo "New record created successfully";
        }
    }
}
```

```
}  
else {  
    echo "Error: " . $sql . "<br>" . $conn->error;  
}  
  
$conn->close();  
}  
else {  
    echo "Wrong API Key provided."  
}  
  
}  
else {  
    echo "No data posted with HTTP POST."  
}  
  
function test_input($data) {  
    $data = trim($data); //  
    $data = stripslashes($data);  
    $data = htmlspecialchars($data);  
    return $data;  
}
```