

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών *Εφαρμοσμένη
Πληροφορική της Υγείας & Τηλεϊατρική*

Μεταπτυχιακή Διατριβή



Τεχνολογίες Έξυπνου Σπιτιού για Εφαρμογές
Τηλεϊατρικής και Διαχείριση Έκτακτων Περιστατικών

Δημήτριος Κατσέλας

Επιβλέπων Καθηγητής
Θεοδόσιος Γούδας

Νοέμβριος 2019

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών *Εφαρμοσμένη*

Πληροφορική της Υγείας & Τηλεϊατρική

Μεταπτυχιακή Διατριβή

**Τεχνολογίες Έξυπνου Σπιτιού για Εφαρμογές
Τηλεϊατρικής και Διαχείριση Έκτακτων Περιστατικών**

Δημήτριος Κατσέλας

**Επιβλέπων Καθηγητής
Θεοδόσιος Γούδας**

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων για απόκτηση μεταπτυχιακού τίτλου σπουδών στην Εφαρμοσμένη Πληροφορική της Υγείας & Τηλεϊατρική από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών του Ανοικτού Πανεπιστημίου Κύπρου.

Νοέμβριος 2019

Περίληψη

Η υψηλού επιπέδου υγειονομική περίθαλψη αποτελεί σημαντική ανάγκη για τον ανθρώπινο πληθυσμό. Παρόλο που η ανάπτυξη της Ιατρικής τις τελευταίες δεκαετίες είναι ραγδαία, το πρόβλημα της επαρκούς και αποτελεσματικής περίθαλψης παραμένει. Οι κοινωνικές ομάδες που παρουσιάζουν τη μεγαλύτερη δυσκολία πρόσβασης σε αυτές τις υπηρεσίες είναι οι ηλικιωμένοι άνθρωποι, οι κάτοικοι απομακρυσμένων περιοχών και νησιών, οι άνθρωποι χαμηλότερου εισοδήματος και οι ασθενείς με χρόνιες παθήσεις. Η πρόοδος των κλάδων των Τηλεπικοινωνιών και της Πληροφορικής παρέχει μία πολύ δραστική λύση στα παραπάνω προβλήματα.

Σε αυτή τη διπλωματική εργασία θα υλοποιήσουμε ένα έργο αναπτύσσοντας ένα σύστημα παρακολούθησης της υγείας ασθενούς βασιζόμενο στο IoT. Το σύστημα αυτό καταγράφει τον ρυθμό της καρδιάς, τον αναπνευστικό ρυθμό και την θερμοκρασία του σώματος. Όλες αυτές οι τιμές αποστέλλονται σε μια πλατφόρμα IoT έτσι ώστε η κατάσταση της υγείας του ασθενούς να παρακολουθείται από οποιοδήποτε σημείο του κόσμου μέσω του διαδικτύου. Για την παρακολούθηση αυτών των βιοσημάτων θα χρησιμοποιήσουμε την πλατφόρμα ThingSpeak. Θα χρησιμοποιηθεί επίσης, η πλατφόρμα IFTTT για την παροχή στην ThingSpeak υπηρεσιών SMS και email έτσι ώστε να αποστέλλονται ειδοποιήσεις συναγερμού όταν λαμβάνονται τιμές εκτός προκαθορισμένων κρίσιμων ορίων.

Abstract

The need of high-level medical care is very crucial for people. Although the progress of medicine is rapid the last decades, the problem of sufficient and efficient health care still remains. The social groups that have the most difficulty accessing those services are the elderly, those who live at remote areas or islands and also those who have lower incomes. The development of Telecommunications and Computer Science can offer a very drastic solution to the above problems.

Here in this project, we will make an IOT based health monitoring system which records the patient heartbeat rate, body temperature and respiratory rate. Heartbeat rate, body temperature and respiratory values are recorded over Thingspeak so that the patient's health can be monitored from anywhere in the world over the internet. We will use Thingspeak monitor patient heartbeat, respiratory rate and temperature online using internet. We will also use IFTTT platform to connect Thingspeak to SMS so that alert messages can be sent whenever the patient is in critical state.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Θ. Γούδα για τις συμβουλές, την υποστήριξη και την παρότρυνση πάνω στο συγκεκριμένο αντικείμενο, καθώς και την εμπιστοσύνη που έδειξε προς το πρόσωπό μου αναθέτοντάς μου την παρούσα διπλωματική εργασία. Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου για τη συμπαράσταση και την υπομονή της καθ' όλη τη διάρκεια των σπουδών μου στο Ανοιχτό Πανεπιστήμιο Κύπρου.

Περιεχόμενα

<i>Περίληψη</i>	<i>i</i>
<i>Abstract</i>	<i>ii</i>
ΚΕΦΑΛΑΙΟ 1	1
<i>Εισαγωγή</i>	1
1.1 Σκοπός και Επιμέρους Στόχοι της Εργασίας	2
1.2 Οργάνωση και Δομή της Εργασίας	5
Κεφάλαιο 2	7
Θεωρητικό και Τεχνολογικό Υπόβαθρο – Βασικές Έννοιες	7
2.1 IoT – Internet of Things	7
2.1.1 Ορισμός του Internet of Things	8
2.2 Θεωρητικό Υπόβαθρο	8
2.3 Βιοσήματα	9
2.3.1 Η Θερμοκρασία του σώματος	9
2.3.2 Καρδιακός ρυθμός (pulse rate)	12
2.3.3 Ο αναπνευστικός ρυθμός	15
2.4 Τεχνολογικό Υπόβαθρο – Βασικές Έννοιες	18
2.4.1 Πλακέτες ανάπτυξης πρωτοτύπων (Development boards)	18
2.4.2 Μικροελεγκτής	19
2.4.3 Επικοινωνία μεταξύ Software και Hardware	20
2.4.4 Μετατροπέας Analog-to-Digital	21
2.4.5 Τελεστικοί Ενισχυτές	22
2.4.6 Cloud Υπηρεσία	24
2.4.7 Πλατφόρμες IoT	25
2.4.8 Διεπαφή Προγραμματισμού Εφαρμογών (Application Program Interfaces, APIs)	26
2.4.9 Το πρωτόκολλο HTTP	26
2.4.10 MATLAB	28
Κεφάλαιο 3	29
Βιβλιογραφική Επισκόπηση	29
Κεφάλαιο 4	45
Ανάλυση και Σχεδιασμός του Συστήματος	45
4.1 Γενική Περιγραφή	45
4.2 Λεπτομερής Περιγραφή του Συστήματος	46

4.3 Σχεδιασμός του Hardware.....	46
4.3.1 Μονάδα Data Acquisition.....	47
4.3.2 Microcontroller (MCU): Arduino UNO	56
4.3.3 Wi-Fi module ESP8266-01.....	58
4.3.4 Push Button.....	60
4.3.5 LED Καρδιακού ρυθμού.....	62
4.4 Λογισμικό.....	63
4.4.1 ARDUINO IDE.....	63
4.4.2 Η IoT πλατφόρμα ThingSpeak.....	65
4.4.3 Η πλατφόρμα IFTTT	68
4.4.4 ThingHTTP App.....	70
4.4.5 React App.....	70
4.4.6 Virtuino.....	71
Κεφάλαιο 5	72
Υλοποίηση του συστήματος.....	72
5.1 Ερμηνεία του Arduino κώδικα.....	72
5.2 Παραμετροποίηση της πλατφόρμας ThingSpeak και υλοποίηση της online παρακολούθησης.	82
5.2.1 Πλατφόρμα ThingSpeak.com.....	82
5.2.2 Εφαρμογές ThingHTTP	85
5.2.3 Ρύθμιση IFTTT για ενεργοποίηση αποστολής email/sms με βάση τις τιμές στο ThingSpeak.....	86
Κεφάλαιο 6	92
Αποτελέσματα και Ανάλυση Δεδομένων.....	92
6.1 Τελική Μορφή του Συστήματος Patient Monitor	92
6.2 Serial Monitor	93
6.3 Live Παρακολούθηση και Εξαγωγή Δεδομένων.....	94
6.4 Μετρήσεις Pulse Rate και Θερμοκρασίας.....	95
6.5 Ρυθμός Αναπνοής και Emergency Button.....	95
6.6 Ειδοποιήσεις και Συναγερμοί.....	96
6.7 Αποστολή SMS.....	98
6.8 VIRTUINO	98
Κεφάλαιο 7	100
Συμπεράσματα	100
7.1 Επίλογος - Συμπεράσματα	100
7.2 Παρατηρήσεις - Περιορισμοί	101
7.3 Θέματα για Περαιτέρω Διερεύνηση	101
Παράρτημα Α.....	105
Κώδικας Arduino Uno	105
Βιβλιογραφία.....	113

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

Τις τελευταίες δεκαετίες η εξέλιξη της τεχνολογίας φαίνεται να έχει επηρεάσει ριζικά τον τρόπο ζωής των ανθρώπων. Η τεχνολογία έχει κληθεί να προσφέρει λύση σε μία από τις βασικότερες ανάγκες της ανθρώπινης ύπαρξης, την υγεία. Σήμερα, οι αυτοματισμοί και το Διαδίκτυο των Πραγμάτων (Internet of Things, IoT) αλλάζουν τον κόσμο. Μέρα με τη μέρα η ανάπτυξη του IoT προκαλεί επανάσταση στις νέες τεχνολογίες, οι οποίες κάνουν τη ζωή μας άνετη και αυτοματοποιημένη. Η τεχνολογία αυτή ενσωματώνει ένα ευρύ φάσμα δικτυωμένων προϊόντων, συστημάτων και αισθητήρων, που εκμεταλλεύονται τις προόδους στην υπολογιστική ισχύ, στη μικρογραφία και στις διασυνδέσεις δικτύων για να προσφέρουν νέες δυνατότητες που δεν ήταν προηγουμένως δυνατές (Fernandez και Pallis, 2014).

Καθώς η υγεία είναι ένα από τα πιο σημαντικά θέματα στη ζωή του ανθρώπου το IoT θα μπορούσε να χρησιμοποιηθεί στον τομέα της υγείας ως ένα σύστημα συνεχούς παρακολούθησης της κατάστασης της υγείας του ασθενούς. Συγχρόνως, το Internet σήμερα είναι εύκολα διαθέσιμο για κινητές συσκευές, κάτι που κάνει την απομακρυσμένη παρακολούθηση ιδιαίτερα δημοφιλή. Έτσι, τα συστήματα που αναπτύσσονται προσφέρουν ένα νέο τρόπο παροχής ιατρικών υπηρεσιών επιτρέποντας την ισόνομη και εύκολη πρόσβαση ανθρώπων στις υγειονομικές υπηρεσίες μέσω της απομακρυσμένης και συστηματικής παρακολούθησης ασθενών με σκοπό την αντιμετώπιση επειγόντων περιστατικών που μέχρι πρότινος ήταν αδύνατη για ορισμένες κοινωνικές ομάδες. Ασθενείς με χρόνιες ασθένειες, όπως υπέρταση,

αναπνευστικές νόσους ή διαβήτη, χρειάζονται συχνότερες ιατρικές, νοσοκομειακές και επείγουσες υπηρεσίες σε σχέση με τους κανονικούς ασθενείς (Paré και άλλοι, 2010). Επίσης, αγροτικές και ορεινές περιοχές θα μπορούσαν να επωφεληθούν από αυτή την τεχνολογία.

Η απομακρυσμένη παρακολούθηση ασθενούς εξελίσσεται συνεχώς και έχει πολλά περιθώρια περαιτέρω εξέλιξης και ανάπτυξης. Είναι ζωτικής σημασίας για την παροχή αξιόπιστης φροντίδας στην αναπνευστική θεραπεία, την ανάνηψη, την κατ' οίκον θεραπεία, στις μεταφορές ασθενών καθώς και στην παρακολούθηση διαταραχών ύπνου στο σπίτι.

Όταν κάποιος ασθενής εισάγεται σε νοσοκομείο ή σε κάποια άλλη μονάδα υγείας, οι συγγενείς του ανησυχούν για την κατάσταση της υγείας του καθ' όλη τη διάρκεια της παραμονής του στους χώρους αυτούς. Ο συνδυασμός μιας πλακέτας ανάπτυξης, όπως είναι το Arduino, με το IoT έδωσε λύση σε τέτοιες καταστάσεις. Η εφαρμογή καινοτόμων τεχνολογιών στα υγειονομικά συστήματα έκανε δυνατή την απομακρυσμένη παρακολούθηση της κατάσταση υγείας του ασθενούς. Αισθητήρες λαμβάνουν δεδομένα για διάφορες παραμέτρους που αφορούν την υγεία του ασθενούς και το IoT αποθηκεύει τα δεδομένα και τα απεικονίζει μέσω ιστοσελίδας ή κάποιας κινητής συσκευής. Η χρήση αισθητήρων μειώνει την περίπτωση ανθρώπινου λάθους και το μέγεθος ενός τέτοιου συστήματος επιτρέπει την κατ' οίκον χρήση και παρακολούθηση.

Στο σύστημα που προτείνουμε σε αυτή την εργασία λαμβάνουμε μετρήσεις της θερμοκρασίας σώματος, του καρδιακού και του αναπνευστικού ρυθμού. Επίσης, παρέχουμε την δυνατότητα αποστολής ειδοποιήσεων μέσω email ή SMS όταν κάποια από τις παραμέτρους ξεπεράσει προκαθορισμένα όρια ή όταν ο ασθενής θεωρήσει ότι πρέπει να ειδοποιηθεί αυτός που τον παρακολουθεί.

1.1 Σκοπός και Επιμέρους Στόχοι της Εργασίας

Ο σκοπός της συγκεκριμένης διπλωματικής εργασίας είναι ο σχεδιασμός και η

ανάπτυξη ενός IoT συστήματος χαμηλού κόστους για την μέτρηση και απεικόνιση τριών από τα βασικότερα βιοσήματα, του καρδιακού παλμού, του αναπνευστικού ρυθμού και της θερμοκρασίας του σώματος ενός ασθενή με τη χρήση αισθητήρων, οι οποίοι τοποθετούνται με εύκολο τρόπο σε διάφορα σημεία του σώματος όπως π.χ. το δάκτυλο και η μύτη του ασθενούς. Το σύστημα που αναπτύσσουμε θα έχει, επίσης, την δυνατότητα αποστολής ειδοποιήσεων μέσω email στους συγγενείς και στον γιατρό που παρακολουθεί τον ασθενή όταν οι τιμές των μετρούμενων παραμέτρων ξεπεράσουν κάποια κρίσιμα όρια.

Επιπρόσθετα, ένα button (emergency button), το οποίο έχει τοποθετηθεί στο σύστημα, δίνει την δυνατότητα στον ασθενή, όταν δεν αισθάνεται καλά, να στείλει ειδοποίηση μέσω SMS στους συγγενείς ή στον γιατρό που τον παρακολουθεί.

Στην εργασία αναλύεται επίσης, ο τρόπος με τον οποίο η πλακέτα ανάπτυξης Arduino που χρησιμοποιούμε, λαμβάνει, επεξεργάζεται και αποστέλλει τα δεδομένα σε μια πλατφόρμα IoT, εν προκειμένω η πλατφόρμα ThingSpeak. Με την χρήση της ιστοσελίδας ThingSpeak.com μπορούμε να παρακολουθούμε τα δεδομένα που λαμβάνουμε και να ελέγχουμε το σύστημα μέσω του διαδικτύου με τη χρήση των καναλιών και των ιστοσελίδων που παρέχονται από την πλατφόρμα. Το ThingSpeak συλλέγει τα δεδομένα των αισθητήρων, τα αναλύει και τα απεικονίζει. Μπορεί επίσης, να προκαλέσει αλληλεπίδραση με άλλα συστατικά του συστήματος. Παράλληλα, χρησιμοποιείται η πλατφόρμα IFTTT για την σύνδεση του ThingSpeak με τις υπηρεσίες SMS και email έτσι ώστε να είναι δυνατή η αποστολή μηνυμάτων συναγερμού κάθε φορά που ο ασθενής βρίσκεται σε κρίσιμη κατάσταση.

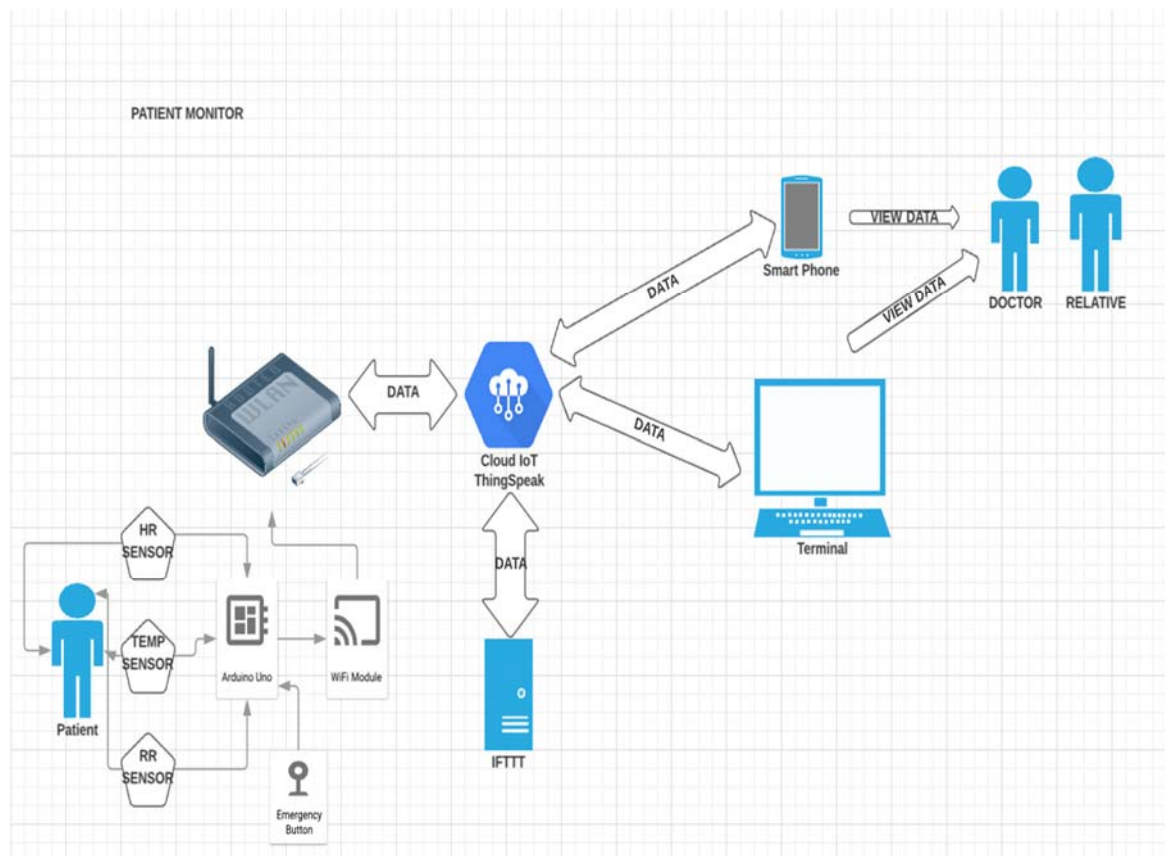
Στους ασθενείς θα παραχωρείται μια συσκευή Arduino στην οποία θα είναι συνδεδεμένοι οι αισθητήρες με τα παρελκόμενά τους, καθώς επίσης και η μονάδα Wi-Fi για τη διασύνδεση του Arduino με το ασύρματο δίκτυο του σπιτιού. Στους συγγενείς θα παρέχονται οδηγίες για τη σύνδεση του εξοπλισμού και οι κωδικοί πρόσβασης της πλατφόρμας ThingSpeak στην οποία

θα γίνεται η παρακολούθηση. Οι κωδικοί αυτοί θα παρέχονται και στον/στους γιατρούς που παρακολουθούν τον ασθενή.

Με βάση τα παραπάνω θα γίνει μελέτη των φαινομένων, των θεωριών των εφαρμογών γύρω από την τεχνολογία του IoT και των προκλήσεων στο σύγχρονο περιβάλλον υγείας. Συνεπώς, οι επιμέρους στόχοι είναι:

- Βιβλιογραφική ανασκόπηση του φαινομένου του Διαδικτύου των Πραγμάτων (IoT).
- Μελέτη του IoT στον τομέα της υγείας και εστίαση στις συσκευές που χρησιμοποιούνται.

Το Σχήμα 1 αποτελεί μια προσπάθεια για μια γενική και αφαιρετική οπτικοποίηση του συστήματος που υλοποιήθηκε, στο οποίο φαίνονται τόσο τα μέρη από τα οποία αποτελείται καθώς και οι κατηγορίες των χρηστών στους οποίους αναφέρεται.



Σχήμα 1. Σύστημα παρακολούθησης ασθενούς. Γενική αρχιτεκτονική του συστήματος.

Η ασφάλεια και η ιδιωτικότητα είναι επίσης σημαντικά ζητήματα, καθώς τα ιατρικά δεδομένα είναι εξ' ορισμού ευαίσθητης φύσεως και θα πρέπει να είναι διαθέσιμα μόνο σε εξουσιοδοτημένους χρήστες, πράγμα το οποίο σέβονται οι κανόνες και η πολιτική του παραπάνω συστήματος.

1.2 Οργάνωση και Δομή της Εργασίας

Η διπλωματική εργασία είναι οργανωμένη σε επτά κεφάλαια.

Στο **1ο Κεφάλαιο** γίνεται μια εισαγωγή στην διπλωματική εργασία, παρουσιάζονται ο σκοπός, οι στόχοι καθώς και η οργάνωση και δομή της εργασίας.

Στο **2ο Κεφάλαιο** παρουσιάζονται ορισμοί καθώς και οι σημαντικότερες έννοιες, θεωρητικές και τεχνικές, οι οποίες είναι άρρηκτα συνδεδεμένες με το έργο και βοηθούν στην διαμόρφωση μιας σφαιρικής εικόνας.

Στο **3ο Κεφάλαιο** παρουσιάζεται η βιβλιογραφική ανασκόπηση που πραγματοποιήθηκε για την περιγραφή του θεωρητικού υποβάθρου της εργασίας. Η μεθοδολογία που χρησιμοποιήθηκε είναι η ανασκόπηση σχετικών, μεμονωμένων μελετών υψηλής ποιότητας, που καλύπτουν έναν ή περισσότερους τομείς των τεχνολογιών που προαναφέρονται.

Στο **4ο Κεφάλαιο** παρουσιάζεται η αρχιτεκτονική του συστήματος, οι απαιτήσεις του και τα μέρη που το αποτελούν.

Στο **5ο Κεφάλαιο** παρουσιάζονται οι λεπτομέρειες της υλοποίησης καθώς και τα πιο σημαντικά ζητήματα, όπως οι μέθοδοι επικοινωνίας, οι τεχνικές που χρησιμοποιήθηκαν για τη λήψη δεδομένων, η λειτουργία των πλατφορμών που

χρησιμοποιήθηκαν, η διεπαφή του συστήματος καθώς και η επεξήγηση μερών του κώδικα.

Στο **6ο Κεφάλαιο** παρουσιάζονται οι μετρήσεις και η ανάλυση των δεδομένων.

Στο **7ο Κεφάλαιο** παρουσιάζονται τα συμπεράσματα από την υλοποίηση και χρήση του συστήματος, γίνεται μια συνολική αποτίμηση της εργασίας και δίνονται πιθανές προεκτάσεις-βελτιώσεις της λειτουργικότητας του συστήματος.

Στο **7ο Κεφάλαιο** παρατίθεται η βιβλιογραφία που χρησιμοποιήθηκε κατά τη συγγραφή της διπλωματικής εργασίας καθώς και παράρτημα με τον κώδικα.

Κεφάλαιο 2

Θεωρητικό και Τεχνολογικό Υπόβαθρο – Βασικές Έννοιες

Σκοπός του κεφαλαίου αυτού είναι να εισάγει τον αναγνώστη στις βασικότερες έννοιες οι οποίες χρησιμοποιούνται σε αυτή τη διπλωματική εργασία. Η εργασία αυτή συνδυάζει έννοιες από τρεις διαφορετικές επιστήμες (Ιατρική, Πληροφορική, Ηλεκτρονική) για την ανάπτυξη ενός IoT συστήματος παρακολούθησης ζωτικών σημάτων σε ασθενείς. Το κεφάλαιο αυτό χωρίζεται σε δύο ενότητες. Στην πρώτη αναφέρονται θεωρητικές κυρίως γνώσεις σχετικές με ιατρικά θέματα. Στην δεύτερη αναλύονται οι πιο βασικές τεχνολογίες που χρησιμοποιήθηκαν για την ολοκλήρωση της διπλωματικής εργασίας.

2.1 IoT – Internet of Things

Ο όρος «Διαδίκτυο των Πραγμάτων – Internet of Things(IoT)» χρησιμοποιήθηκε για πρώτη φορά το 1999 από τον Βρετανό Kevin Ashton, για να περιγράψει ένα σύστημα στο οποίο τα αντικείμενα του φυσικού κόσμου θα μπορούσαν να συνδεθούν με το Internet μέσω αισθητήρων. Σήμερα, το Διαδίκτυο των Πραγμάτων έχει γίνει ένας δημοφιλής όρος για την περιγραφή σεναρίων στα οποία η σύνδεση στο Internet και οι δυνατότητες των υπολογιστών εκτείνονται σε μια ποικιλία από αντικείμενα, συσκευές,

αισθητήρες και είδη καθημερινής χρήσης (Παπασταθοπούλου, 2017).

2.1.1 Ορισμός του Internet of Things

Παρά το μεγάλο παγκόσμιο ενδιαφέρον γύρω από το Διαδίκτυο των Πραγμάτων, δεν υπάρχει ένας διεθνώς αποδεκτός ορισμός για τον όρο. Σύμφωνα με τον οργανισμό ITU (International Telecommunication Union) ο ορισμός για το IoT είναι: “Μια παγκόσμια υποδομή για τον κόσμο της πληροφορίας η οποία ενεργοποιεί εξελιγμένες υπηρεσίες μέσω της διαδικτύωσης εικονικών και πραγματικών πραγμάτων βασιζόμενη στις υπάρχουσες και εξελισσόμενες διαλειτουργικές τεχνολογίες πληροφοριών και επικοινωνιών”.

Μέσω της αξιοποίησης των δυνατοτήτων αναγνώρισης , καταγραφής δεδομένων, την επεξεργασία και την επικοινωνία, το IoT κάνει πλήρη χρήση των πραγμάτων έτσι ώστε να προσφέρουν υπηρεσίες σε όλα τα είδη των εφαρμογών, εξασφαλίζοντας παράλληλα ότι πληρούνται οι απαιτήσεις της ασφάλειας και της προστασίας της ιδιωτικής ζωής (ό.π.). Θα μπορούσαμε να πούμε ότι το IoT αποτελεί περισσότερο ιδέα, επιστημονικό κλάδο ανάπτυξης και συνδυασμό τεχνολογιών, παρά αυτοτελή τεχνολογία.

2.2 Θεωρητικό Υπόβαθρο

Τα φυσικά μεγέθη που συνήθως παρακολουθούνται στα περισσότερα ιατρικά περιβάλλοντα είναι ο καρδιακός παλμός (pulse rate), ο ρυθμός αναπνοής (Respiratory Rate), η πίεση του αίματος (blood pressure), ο κορεσμός του οξυγόνου (SpO2) και η θερμοκρασία του σώματος. Η παρακολούθηση γίνεται μέσω των αντίστοιχων βιοσημάτων αυτών των μεγεθών. Κρίνεται, λοιπόν, απαραίτητο να οριστεί ο όρος «βιοσήμα» και παράλληλα, να παρουσιαστούν βασικοί ορισμοί των παραπάνω μεγεθών. Τέλος, θα παρουσιαστούν επίσης, και οι τρόποι συλλογής δεδομένων με χρήση αισθητήρων.

2.3 Βιοσήματα

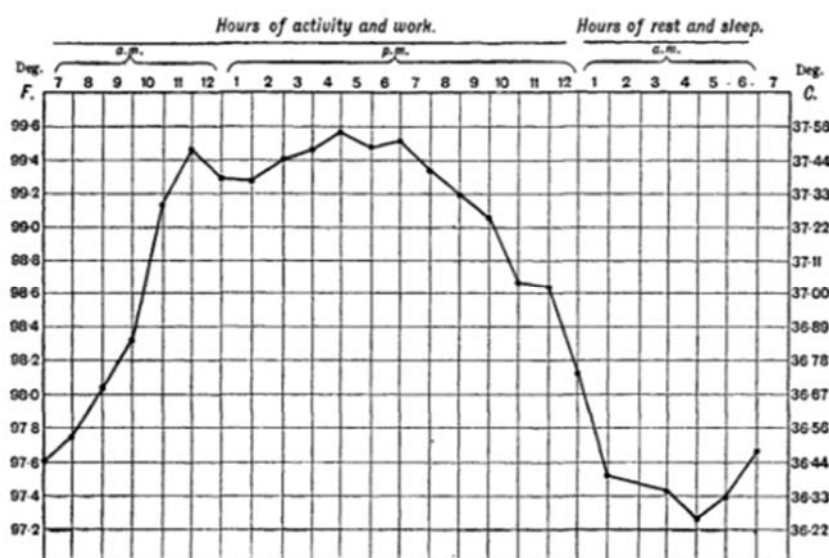
Το βιοσήμα (ή βιολογικό σήμα) είναι η χωροχρονική καταγραφή ενός βιολογικού γεγονότος όπως μια παλλόμενη καρδιά ή ένας συσπώμενος μυς. Η ηλεκτρική, χημική, και μηχανική δραστηριότητα που εμφανίζεται κατά τη διάρκεια αυτού του βιολογικού γεγονότος παράγει συχνά σήματα που μπορούν να μετρηθούν και να αναλυθούν. Τα βιολογικά σήματα, επομένως, περιέχουν τις πληροφορίες που μπορούν να χρησιμοποιηθούν για να εξηγήσουν τους φυσιολογικούς μηχανισμούς που υποκρύπτονται σε ένα συγκεκριμένο βιολογικό γεγονός ή ένα σύστημα. Τα βιοσήματα μπορούν να αποκτηθούν με ποικίλους τρόπους, όπως λόγω χάριν από το στήθοσκόπιο που χρησιμοποιεί ο γιατρός για να ακούσει τον ήχο της καρδιάς ενός ασθενή ή με την χρήση των ιδιαίτερα σύνθετων και τεχνολογικά προηγμένων βιοϊατρικών οργάνων (Clifford, 2008).

Στα πλαίσια της παρούσας εργασίας θα επικεντρωθούμε στα παρακάτω θεμελιώδη βιοσήματα: τον καρδιακό ρυθμό (pulse rate), τη θερμοκρασία του σώματος και την αναπνευστική συχνότητα.

2.3.1 Η Θερμοκρασία του σώματος

Η βιοχημική λειτουργικότητα του σώματος συνδέεται με τη σταθερότητα της θερμοκρασίας. Ο μηχανισμός ρύθμισης της θερμοκρασίας σε σταθερές τιμές αποτελεί ένα σύστημα ελέγχου που εξισορροπεί την απώλεια και την παραγωγή θερμότητας [6]. Η παραγωγή θερμότητας είναι παρεπόμενο των μεταβολικών διεργασιών του οργανισμού ενώ η απώλεια θερμότητας επιτυγχάνεται με τη μεταφορά της θερμότητας στο δέρμα και από εκεί στο περιβάλλον (Boron και Boulpaep, 2017).

Το σώμα συνεχώς παράγει θερμότητα ως προϊόν μεταβολισμού. Όταν η ποσότητα της παραγόμενης θερμότητας είναι ίση με την ποσότητα που χάθηκε το άτομο είναι σε θερμική ισορροπία. Η θερμοκρασία του σώματος ακολουθεί τον κερκάρδιο ρυθμό του σώματος και η μεταβολή της στο εύρος του 24ωρου ακολουθεί συγκεκριμένη εξέλιξη (Σχήμα 2). Μάλιστα, η εξέλιξή της θεωρείται τόσο ακριβής ώστε χρησιμοποιείται σαν μέτρο σύγκρισης με τους άλλους βιολογικούς ρυθμούς του σώματος για την εξακρίβωση της ρύθμισής τους. Η χαμηλότερη θερμοκρασία εντοπίζεται δύο περίπου ώρες πριν την αφύπνιση και η μεγαλύτερη προς το τέλος του απογεύματος, γύρω στις 6 το απόγευμα.



Σχήμα 2. Εξέλιξη της θερμοκρασίας κατά τη διάρκεια ενός 24-ώρου.

Οι βασικότεροι παράγοντες που επιδρούν στην παραγωγή θερμότητας είναι ο ρυθμός βασικού μεταβολισμού, η εκούσια αλλά και ακούσια μυϊκή δραστηριότητα, η θυροξίνη, η επινεφρίνη, η νοραδρελίνη κ.α.

Το κέντρο ρύθμισης της θερμοκρασίας είναι ο υποθάλαμος του εγκεφάλου. Ο υποθάλαμος ελέγχει τη θερμοκρασία του σώματος μέσω μηχανισμού ανάδρασης. Όταν η θερμοκρασία του σώματος αυξάνει, ο υποθάλαμος μέσω του κεντρικού νευρικού συστήματος (ΚΝΣ) προκαλεί διαστολή των αιμοφόρων αγγείων, εφίδρωση και αναστολή της παραγωγής θερμότητας. Όταν η θερμοκρασία του σώματος πέσει κάτω από τα φυσιολογικά όρια, ο

υποθάλαμος μέσω του ΚΝΣ προκαλεί συστολή των αιμοφόρων αγγείων (Wright κ.ά., 2002).

Υπό φυσιολογικές συνθήκες το θερμορυθμιστικό κέντρο διατηρεί τη θερμοκρασία του σώματος ανάμεσα στους 36 με 37,4 °C. Είναι δυνατόν να υπάρξουν μικρές διαφοροποιήσεις στη φυσιολογική θερμοκρασία του σώματος, οι οποίες μπορεί να εξαρτώνται από την ιδιοσυγκρασία του κάθε ατόμου αλλά και αλλαγές κατά τη διάρκεια της ημέρας.

Η θερμοκρασία διαφοροποιείται στα διάφορα σημεία του σώματος, με την εσωτερική θερμοκρασία να παραμένει πάντοτε υψηλότερη από τη θερμοκρασία της επιφάνειας. Η μέτρηση της εσωτερικής θερμοκρασίας γίνεται από τον ορθό ή τη μεμβράνη των τυμπάνων των αυτιών. Η θερμοκρασία της επιφάνειας μετριέται από το στόμα (υπογλώσσια), τη μασχάλη και το μέτωπο (4^η Υ.Π.Ε., 2014).

Υπάρχουν ποικίλοι τρόποι μέτρησης της θερμοκρασίας. Το παραδοσιακό θερμόμετρο στηρίζεται στην αύξηση του ύψους της στήλης υδραργύρου ανάλογα με τη θερμοκρασία. Η αύξηση της αντίστασης με την αύξηση της θερμοκρασίας διέπει τη λειτουργία τόσο των παραδοσιακών ηλεκτρικών θερμομέτρων όσο και των θερμίστορς. Τα πρώτα ενσωματώνουν μέταλλα ενώ τα δεύτερα ημιαγωγούς, οι οποίοι είναι πολύ πιο ευαίσθητοι στις μεταβολές της θερμοκρασίας. Μια εναλλακτική μέθοδος μέτρησης στηρίζεται στα θερμοζεύγη. Τα θερμοηλεκτρικά ζεύγη ή θερμοζεύγη (*thermocouples*) αποτελούν ένα εξαιρετικά διαδεδομένο είδος ανιχνευτών θερμοκρασίας, υψηλής ακρίβειας και χαμηλού κόστους. Αποτελούνται από δύο σύρματα διαφορετικών μετάλλων, τα οποία είναι ενωμένα σε δύο σημεία. Με την εξέλιξη της τεχνολογίας νέοι τρόποι μέτρησης εμφανίστηκαν με την ενσωμάτωση αισθητήρων σε ολοκληρωμένα κυκλώματα. Το ολοκληρωμένο κύκλωμα LM35 είναι ένας από τους πιο διαδεδομένους αισθητήρες θερμοκρασίας και αποτελεί τον αισθητήρα θερμοκρασίας που χρησιμοποιούμε στην παρούσα εργασία. Η λειτουργία του στηρίζεται στις ιδιότητες της επαφής *p-n*. Όπως είναι γνωστό, όταν η επαφή *p-n* είναι πολωμένη ανάστροφα, τότε το ανάστροφο ρεύμα κόρου είναι συνάρτηση

της θερμοκρασίας της επαφής. Τα ολοκληρωμένα αισθητήρια γενικά εμφανίζουν άριστη γραμμικότητα.

Το LM35 μεταβάλλει την έξοδό του κατά 10 mV όταν η θερμοκρασία μεταβάλλεται κατά 1 °C (LM 35 Datasheet).

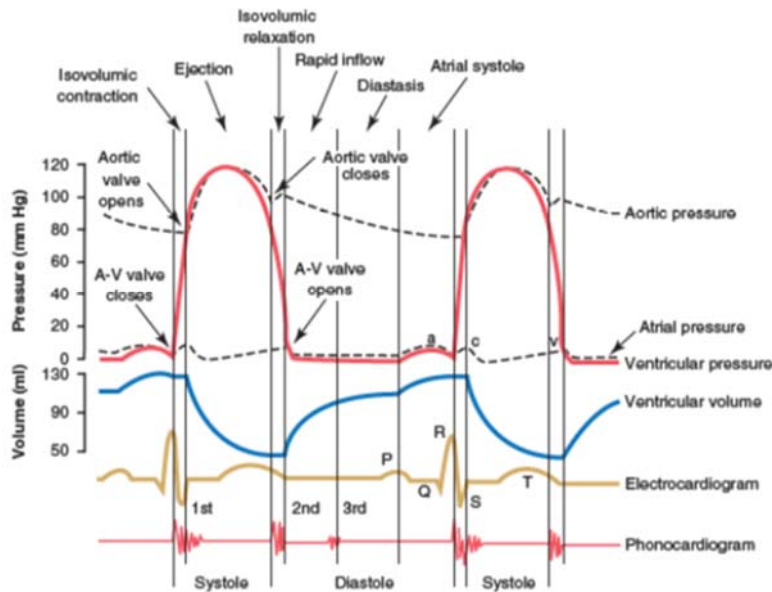
Οι μέσες φυσιολογικές τιμές της θερμοκρασίας διαφόρων σημείων του σώματος σε ενήλικες φαίνονται στον Πίνακα 1.

Πίνακας 1. Μέσες φυσιολογικές τιμές Θερμοκρασίας.

Στόμα	Ορθό	Μασχάλη	Τυμπανική Μembrάνη(αυτί)	Μέτωπο
37,4 °C - 37,7 °C	37,5 °C - 38 °C	36,5 °C - 37 °C	37,5 °C - 38,3 °C	34,4 °C - 36 °C

2.3.2 Ο καρδιακός ρυθμός (pulse rate)

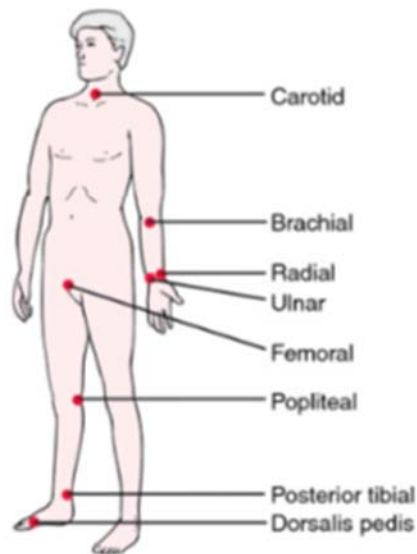
Η καρδιά είναι κοίλος μυς, ο οποίος, δίνοντας στο αίμα πίεση, το κάνει να κυκλοφορεί στο εσωτερικό των αρτηριών με τέτοιο τρόπο, ώστε να φτάνει σε όλα τα όργανα. Είναι φυσική αντλία που παίρνει το αίμα από τις φλέβες, στις οποίες βρίσκεται υπό χαμηλή πίεση, και το στέλνει στις αρτηρίες με υψηλή πίεση. Η καρδιά, εν γένει, λειτουργεί ακατάπαυστα καθ' όλη τη διάρκεια της ζωής του οργανισμού. Ένα μέτρο της κατάστασης της καρδιάς είναι ο καρδιακός ρυθμός. Ο καρδιακός ρυθμός αναφέρεται στον τρόπο αλληλουχίας των παλμών και των παύσεων που γίνονται αισθητοί στην κορυφή της καρδιάς ή στις περιφερειακές αρτηρίες. Ο ρυθμός αυτός φυσιολογικά αντιστοιχεί στο ρυθμό συστολής-διαστολής της καρδιάς. Στο Σχήμα 3 βλέπουμε τα γεγονότα που συμβαίνουν ανάμεσα σε δύο διαδοχικά χτυπήματα της καρδιάς για την αριστερή κοιλία της καρδιάς, σε διάφορες καμπύλες (Orie, 2004).



Σχήμα 3. Οι φάσεις του καρδιακού κύκλου.

Η μέτρηση του καρδιακού ρυθμού είναι μια από τις πιο σημαντικές παραμέτρους του ανθρώπινου καρδιαγγειακού συστήματος. Ο καρδιακός ρυθμός ενός υγιή ενήλικα σε κατάσταση ηρεμίας είναι περίπου 72 χτύποι ανά λεπτό (beats per minutes, bpm). Οι αθλητές έχουν συνήθως χαμηλότερο καρδιακό ρυθμό από ανθρώπους που δεν αθλούνται. Στα μωρά ο καρδιακός ρυθμός είναι υψηλότερος, περίπου 120 bpm, και στα μεγαλύτερα παιδιά περίπου 90 bpm. Ο καρδιακός ρυθμός αυξάνεται σταδιακά κατά τη διάρκεια της άσκησης και επιστρέφει αργά στην τιμή ηρεμίας μετά την άσκηση. Ο ρυθμός με τον οποίο ο παλμός επιστρέφει στις φυσιολογικές τιμές είναι ένδειξη της φυσικής κατάστασης του ατόμου. Οι τιμές άνω των 100bpm θεωρούνται ταχυκαρδία και κάτω των 60bpm βραδυκαρδία (Goljan, 2011).

Για την μέτρηση του καρδιακού ρυθμού χρησιμοποιούνται πολλοί τρόποι. Παραδοσιακά η μέτρηση γίνεται με ψηλάφηση της κερκιδικής αρτηρίας στην περιοχή του καρπού. Μπορεί όμως να μετρηθεί και στην καρωτίδα, τη μηριαία αρτηρία, την ιγνυακή αρτηρία, την οπίσθια κνημιαία αρτηρία και τη ραχιαία του ποδός (βλ. Εικόνα 1).



Εικόνα 1. Σημεία μέτρησης καρδιακού ρυθμού.

Μια πιο ακριβής μέθοδος μέτρησης των σφυγμών περιλαμβάνει τη χρήση του ηλεκτροκαρδιογραφήματος, κατά το οποίο τα ηλεκτρόδια στο δέρμα αποτυπώνουν το μοτίβο εκπόλωσης και επαναπόλωσης του καρδιακού μυός (Hall και Guyton, 2011). Ο καρδιακός ρυθμός προκύπτει άμεσα από το ηλεκτροκαρδιογράφημα.

Επίσης, αρκετά διαδεδομένη μέθοδο μέτρησης αποτελεί η χρήση αισθητήρα δακτύλου (fingertip sensor) τον οποίο χρησιμοποιούμε στην παρούσα εργασία. Ο αισθητήρας δακτύλου βασίζεται στη μέτρηση ενός φυσιολογικού σήματος που ονομάζεται φωτοπληθυσμογραφία (Photoplethysmography, PPG) (Azmal κ.ά, 2006), η οποία γίνεται με τη χρήση φωτός, είναι δηλαδή, μια οπτική μέτρηση της αλλαγής του όγκου του αίματος στις αρτηρίες.

Ο αισθητήρας δακτύλου λαμβάνει σήματα PPG εκπέμποντας φως με συγκεκριμένο μήκος κύματος μέσω των ιστών, και συγκρίνοντας τα χαρακτηριστικά απορρόφησης φωτός από το αίμα κάτω από αυτά τα μήκη κύματος (Photoplethysmogram Wiki, 2019).

2.3.3 Ο αναπνευστικός ρυθμός

Η βασική, αν και όχι μοναδική, λειτουργία του αναπνευστικού συστήματος είναι η αναπνοή με την οποία επιτυγχάνεται η παροχή στον οργανισμό του αναγκαίου για την αερόβια μεταβολική του δραστηριότητα ποσού οξυγόνου (περίπου 250ml/min) και η απαγωγή της ποσότητας διοξειδίου του άνθρακα (περίπου 224 ml/min) (Ganong, 2005). Κατά τη διάρκεια της αναπνοής αέρας, ο οποίος περιέχει οξυγόνο, εισέρχεται στους πνεύμονες, στους οποίους γίνεται η ανταλλαγή αερίων μέσω της μεμβράνης κυψελίδων-τριχοειδικών αγγείων. Μέρος αυτής της διαδικασίας αποτελεί και η έκκριση διοξειδίου του άνθρακα στον εξωτερικό αέρα μέσω του στόματός ή της μύτης.

Ο αναπνευστικός κύκλος αποτελείται από την εισπνοή και την εκπνοή και αποτελεί ένα ζωτικό σήμα που χρησιμοποιείται για την παρακολούθηση της εξέλιξης μιας ασθένειας. Επίσης, ένας μη φυσιολογικός αναπνευστικός ρυθμός είναι σημαντικός δείκτης σοβαρής ασθένειας.

Ο αναπνευστικός ρυθμός συνήθως μετρείται σε αναπνοές ανά λεπτό (breaths per minute, bpm). Ο τυπικά μέσος αναπνευστικός ρυθμός ποικίλει μεταξύ των ανθρώπων. Έτσι, ο μέσος αναπνευστικός ρυθμός για ένα παιδί 6 μηνών είναι μεταξύ 25 με 40 bpm, για ένα παιδί 3 ετών 20 με 30 bpm, ένα παιδί 10 ετών έχει 17 με 23 bpm και ένας ενήλικας 12 με 18 bpm (Respiratory rate Wiki, 2018).

Ο αναπνευστικός ρυθμός μαζί με τον καρδιακό παλμό και τη θερμοκρασία σώματος αποτελούν ένα δείκτη για την κατάσταση της υγείας ενός ατόμου. Τυπικά η μέτρησή του γίνεται σε ηρεμία υπολογίζοντας τον αριθμό των αναπνοών στο λεπτό με την μέτρηση των επαναλήψεων που ανεβαίνει και κατεβαίνει το στήθος του εξεταζόμενου. Ο αναπνευστικός ρυθμός μπορεί να ανέβει εξαιτίας πυρετού, ασθένειας ή οποιασδήποτε άλλης ιατρικής κατάστασης. Όταν η θερμοκρασία του σώματος ανέρχεται, η συχνότητα των αναπνοών αυξάνεται λόγω αύξησης του μεταβολισμού. Η συχνότητα αυξάνεται κατά 4 αναπνοές ανά λεπτό, σε κάθε αύξηση 0,6°C θερμοκρασίας πάνω από τα κανονικά επίπεδα. Οποιαδήποτε κατάσταση προκαλεί αύξηση του διοξειδίου









του άνθρακα και μείωση του οξυγόνου στο αίμα τείνει να αυξάνει τη συχνότητα και το βάθος των αναπνοών. Επίσης, συγκεκριμένα φάρμακα, ιδιαίτερα τα ναρκωτικά, καταστέλλουν τον αναπνευστικό ρυθμό. Ο αναπνευστικός ρυθμός είναι σημαντικός ζωτικός δείκτης καθώς η αλλαγή στο μοτίβο της αναπνοής είναι η πρώτη ένδειξη επιδείνωσης του επιπέδου οξυγόνου ενός ατόμου. Μια έρευνα των Fieselmann et al (1993) έδειξε ότι ένας ρυθμός αναπνοής μεγαλύτερος από 27 bpm είναι ένας καλύτερος προγνωστικός παράγοντας της καρδιακής ανακοπής εντός 72 ωρών σε σύγκριση με την αρτηριακή πίεση και τον καρδιακό ρυθμό[64]. Έχει επίσης, προταθεί ότι σε ασταθείς ασθενείς οι σχετικές αλλαγές στην αναπνευστική συχνότητα ήταν πολύ μεγαλύτερες από τις μεταβολές της καρδιακής συχνότητας ή της πίεσης και έτσι αποτελεί μία ασφαλή μέθοδο αξιολόγησης των ασθενών υψηλού κινδύνου (Subbe κ.ά., 2003).

Ένας φυσιολογικός αναπνευστικός ρυθμός είναι μεταξύ 12-18 αναπνοών ανά λεπτό και οποιαδήποτε τιμή μεγαλύτερη και μικρότερη από αυτή την περιοχή είναι ένας βασικός δείκτης για την επιδείνωση της υγείας του ατόμου. Ένα πρόβλημα με τον αναπνευστικό ρυθμό είναι ότι είναι δύσκολο να μετρηθεί η αναπνοή ενός ατόμου με ακρίβεια καθώς το άτομο μπορεί να αλλάζει τον ρυθμό της αναπνοής του.

Παρά τη μεγάλη σημασία του αναπνευστικού ρυθμού, υπάρχει έλλειψη απλών εργαλείων ή οργάνων μέτρησης του αναπνευστικού ρυθμού. Σήμερα, η μη αυτόματη μέτρηση του αναπνευστικού ρυθμού εξακολουθεί να χρησιμοποιείται, γεγονός που οδηγεί σε χρονοβόρες διαδικασίες για τον προσδιορισμό του (Fieselmann et al, 1993).

Για την μέτρηση του αναπνευστικού ρυθμού υπάρχουν διάφορες τεχνολογίες, οι οποίες μπορούν να ταξινομηθούν σε δύο μεγάλες κατηγορίες. Η μια κατηγορία είναι αυτή στην οποία οι αισθητήρες μέτρησης είναι σε επαφή με το σώμα και η άλλη αυτή στην οποία οι αισθητήρες δεν βρίσκονται σε επαφή με το σώμα. Στην πρώτη κατηγορία (αισθητήρες με επαφή) διακρίνουμε επτά διαφορετικές τεχνικές μέτρησης, οι οποίες κατηγοριοποιούνται με βάση το

αντικείμενο που μετριέται. Έτσι, οι επτά αυτές κατηγορίες είναι: Οι τεχνικές με βάση τη ροή αέρα της αναπνοής (Respiratory Airflow), τεχνικές με βάση τους αναπνευστικούς ήχους (Respiratory Sounds), τεχνικές με βάση τη θερμοκρασία του αέρα (Air Temperature), τεχνικές με βάση την υγρασία του αέρα (Air Humidity), τεχνικές με βάση συστατικά του αέρα, π.χ. CO₂ (Air Components), τεχνικές με βάση την ανάλυση της κίνησης του στέρνου (Chest Wall Movement Analysis), τεχνικές καρδιακής δραστηριότητας (Modulation of Cardiac Activity). Στην Εικόνα 2 βλέπουμε τις τεχνικές που χρησιμοποιούνται και τις υποκατηγορίες από τις οποίες αποτελούνται, καθώς επίσης και τις θέσεις επάνω στο σώμα όπου τοποθετούνται οι αισθητήρες (Massaroni et al, 2019).

Respiratory airflow	Respiratory sounds	Air Temperature	Air Humidity	Air components	Chest wall movements	Modulation cardiac activity	
Flow measurements	Acoustic measurements	Temperature measurements	Relative humidity measurements	CO ₂ measurements	Strain measurements	Biopotential measurements	
Differential flowmeters	Microphones	Thermistors	Capacitive sensors	Infrared sensors	Resistive sensors	ECG sensors	
Turbine flowmeters		Thermocouples	Resistive sensors	Fiber optic sensors	Capacitive sensors	Light intensity measurements	
Hot wire anemometers		Pyroelectric sensors	Nanocrystal and nanoparticles sensors		Inductive sensors	PPG sensors	
Fiber optic sensors		Fiber optic sensors	Fiber optic sensors		Fiber optic sensors		
						Impedance measurements	
					Transthoracic impedance sensors		
					Movement measurements		
					Accelerometers		
					Gyroscopes		
					Magnetometers		
							

Εικόνα 2. Οι πιο γνωστές τεχνικές (με βάση την επαφή) για την μέτρηση του αναπνευστικού ρυθμού και οι σχετικές περιοχές στο σώμα όπου τοποθετούνται οι αισθητήρες. PPG= Φωτοπληθυσμογραφία, ECG=Ηλεκτροκαρδιογράφημα.

Στην παρούσα διπλωματική εργασία επιλέχθηκε η τεχνική μέτρησης της θερμοκρασίας του αέρα με τη χρήση θερμίστορ. Πιο συγκεκριμένα, γίνεται χρήση ενός θερμίστορ το οποίο τοποθετείται σε μια ρινική μάσκα και ανιχνεύει τις αλλαγές θερμοκρασίας του αέρα μεταξύ εισπνοής και εκπνοής. Θερμίστορ

είναι ένας τύπος ηλεκτρικής αντίστασης του οποίου η τιμή της αντίστασης αλλάζει με τη μεταβολή της θερμοκρασίας. Υπάρχουν δύο τύποι θερμίστορες: (i) τα θερμίστορες αρνητικού συντελεστή θερμοκρασίας (negative temperature coefficient, NTC), στα οποία η αντίσταση πέφτει όταν η θερμοκρασία ανεβαίνει και, (ii) τα θερμίστορες θετικού συντελεστή θερμοκρασίας (positive temperature coefficient, PTC), τα οποία παρουσιάζουν την αντίθετη συμπεριφορά δηλαδή, η αντίσταση ανεβαίνει με την αύξηση της θερμοκρασίας. Τα θερμίστορες χαρακτηρίζονται από την υψηλή ευαισθησία στη θερμοκρασιακή περιοχή που ενδιαφέρει για την παρακολούθηση του αναπνευστικού ρυθμού (10-40 °C). Τα εμπορικά θερμίστορες είναι αρκετά μικρά σε μέγεθος (μερικά mm διάμετρο), έχουν όμως αρκετά μεγάλο χρόνο απόκρισης (τυπικά λίγα δευτερόλεπτα), γεγονός το οποίο μπορεί να θέσει σε κίνδυνο μια σωστή ανάλυση του μοτίβου αναπνοής, ειδικά σε περιπτώσεις υψηλού ρυθμού αναπνοής (Storck et al, 1996).

Αυτοί οι αισθητήρες μπορούν να χρησιμοποιηθούν για τη μέτρηση του αναπνευστικού σήματος στο επίπεδο της μύτης σε ενήλικες και βρέφη λόγω του μικρού τους μεγέθους και της χαμηλής διεισδυτικότητας.

2.4 Τεχνολογικό Υπόβαθρο – Βασικές Έννοιες

2.4.1 Πλακέτες ανάπτυξης πρωτοτύπων (Development boards)

Πλακέτα ανάπτυξης πρωτοτύπων είναι μια πλακέτα τυπωμένου κυκλώματος που περιέχει έναν μικροεπεξεργαστή και τον ελάχιστο αριθμό περιφερειακών εξαρτημάτων που απαιτούνται έτσι ώστε ένας μηχανικός να μπορέσει να εξοικειωθεί με τον μικροεπεξεργαστή που υπάρχει στην πλακέτα και να μάθει να τον προγραμματίζει. Παρέχει επίσης, τη δυνατότητα στους χρηστές να σχεδιάζουν πρωτότυπες εφαρμογές.

Σε αντίθεση με ένα σύστημα γενικού σκοπού, όπως είναι ο οικιακός υπολογιστής, μια πλακέτα ανάπτυξης περιέχει λίγο ή καθόλου υλικό

(hardware) για την διεπαφή χρήστη (Microprocessor development board Wiki, 2019).

Τυπικά μια πλακέτα ανάπτυξης αποτελείται από:

- Κύκλωμα τροφοδοσίας. Συνήθως η τροφοδοσία σε αυτές τις πλακέτες είναι 9V DC.
- Θύρα επικοινωνίας/προγραμματισμού. Είναι η θύρα μέσω της οποίας η πλακέτα συνδέεται με τον υπολογιστή για να γίνει ο προγραμματισμός του ενσωματωμένου μικροελεγκτή και η εγκαθίδρυση της επικοινωνίας.
- Ο μικροελεγκτής.
- Pins εισόδου-εξόδου στα οποία μπορούν να συνδεθούν αισθητήρες, μοτέρ, LCD οθόνες και πολλά άλλα.

Η πιο διάσημη και αναγνωρίσιμη πλακέτα ανάπτυξης είναι το Arduino. Το Arduino Board είναι ένας single-board μικροελεγκτής, δηλαδή, **μια απλή μητρική πλακέτα ανοικτού κώδικα**, με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους. Μπορεί να προγραμματιστεί με τη γλώσσα Wiring. Ουσιαστικά πρόκειται για τη γλώσσα προγραμματισμού C++ και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++.

2.4.2 Μικροελεγκτής

Ο **μικροελεγκτής** (*microcontroller*) είναι ένας τύπος επεξεργαστή, ουσιαστικά μια παραλλαγή μικροεπεξεργαστή, ο οποίος μπορεί να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα, λόγω των πολλών ενσωματωμένων υποσυστημάτων που διαθέτει. Χρησιμοποιείται ευρύτατα σε όλα τα ενσωματωμένα συστήματα (embedded systems) ελέγχου, χαμηλού και μεσαίου κόστους, όπως αυτά που χρησιμοποιούνται σε αυτοματισμούς, ηλεκτρονικά καταναλωτικά προϊόντα (από ψηφιακές φωτογραφικές μηχανές έως παιχνίδια), ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα (Μικροελεγκτής Wiki, 2019).

Κάθε μικροελεγκτής περιέχει μέσα σε ένα και μοναδικό ολοκληρωμένο κύκλωμα τα παρακάτω στοιχεία:

- έναν αριθμό από καταχωρητές ειδικού σκοπού (συσσωρευτή, καταχωρητή κατάστασης, μετρητή προγράμματος, καταχωρητή εντολών, καταχωρητή δείκτη)
- εσωτερικούς χρονιστές - απαριθμητές
- αριθμητική και λογική μονάδα (ALU)
- μονάδα αποκωδικοποίησης εντολών.

Βασικά στοιχεία ενός μικροελεγκτή αποτελούν:

- η μνήμη προγράμματος (ROM ή EPROM) και
- η μνήμη καταχωρητών / μεταβλητών (RAM).

Ο προγραμματισμός των μικροελεγκτών μπορεί να γίνει είτε από γλώσσες υψηλού επιπέδου, είτε σε γλώσσα κατανοητή από τον μικροελεγκτή (π.χ. Assembly). Οι πιο διαδεδομένες γλώσσες προγραμματισμού είναι οι C και C++.



Σχήμα 4: Βασική δομή ενός μικροελεγκτή.

2.4.3 Επικοινωνία μεταξύ Software και Hardware

Σε αυτή την εργασία η επικοινωνία μεταξύ Software και Hardware είναι μια σειριακή επικοινωνία δεδομένων. Η σειριακή επικοινωνία χρησιμοποιεί δύο μεθόδους:

1. Σύγχρονη
2. Ασύγχρονη

Στην περίπτωση της σύγχρονης επικοινωνίας μεταφέρεται ένα μπλοκ δεδομένων κάθε φορά ενώ, όταν χρησιμοποιείται η ασύγχρονη μέθοδος, γίνεται μεταφορά ενός μόνο byte τη φορά. Είναι δυνατόν να γραφτεί λογισμικό το οποίο θα χρησιμοποιεί και τις δύο μεθόδους. Το πρόγραμμα αυτό μπορεί να είναι κουραστικό και πολύ μεγάλο και για το λόγο αυτό μεγάλοι κατασκευαστές παράγουν ειδικά IC κυκλώματα για την σειριακή επικοινωνία δεδομένων. Τα κυκλώματα αυτά είναι ευρέως γνωστά ως Universal Asynchronous Transmitter/Receiver (**UART**).

2.4.4 Μετατροπέας Analog-to-Digital

Ο μικροελεγκτής που χρησιμοποιεί το Arduino διαθέτει έναν μετατροπέα από αναλογικό σε ψηφιακό στον οποίο καταλήγουν οι έξι αναλογικές εισοδοί του Arduino. Η μετατροπή αναλογικού σήματος σε ψηφιακό (*A/D conversion*) είναι η πλέον σημαντική διεργασία σε ένα ψηφιακό σύστημα που διασυνδέεται με τον πραγματικό κόσμο. Τα διάφορα φυσικά μεγέθη (πίεση, θερμοκρασία, τάση, απόσταση κλπ) μεταβάλλονται με αναλογικό τρόπο. Κατά συνέπεια, η μετατροπή τους σε τάση μέσω κάποιου αισθητηρίου δημιουργεί ένα αναλογικό σήμα τάσης, το οποίο για να εισαχθεί σε κάποιο ψηφιακό σύστημα επεξεργασίας πρέπει να κωδικοποιηθεί κατάλληλα. Το κύκλωμα που χρησιμοποιούμε γι' αυτήν την κωδικοποίηση το αποκαλούμε μετατροπέα A/D (*A/D converter*).

Ένας μετατροπέας A/D έχει κατ' αρχήν μία απλή ή διαφορική είσοδο, στην οποία τοποθετείται η αναλογική τάση που θέλουμε να μετατρέψουμε σε ψηφιακή, και έναν αριθμό από ακροδέκτες εξόδου που θα οδηγηθούν σε κατάλληλες λογικές καταστάσεις 0 ή 1 μετά τη μετατροπή. Σε πρακτικά κυκλώματα ο ελάχιστος αριθμός των δυαδικών ψηφίων από τα οποία αποτελείται η ψηφιακή λέξη μετά την μετατροπή είναι οκτώ, οπότε λέμε ότι το σύστημα είναι οκτάμπιτο και μπορεί να διακρίνει συνολικά 256 στάθμες.

Αντίστοιχα, μπορεί να έχουμε μετατροπείς A/D 10 bits (οπότε οι στάθμες είναι 1024), 12 bits (4096 στάθμες), των 16 bits, (65536 στάθμες) κ.ο.κ. (*Μετατροπή αναλογικών και Ψηφιακών Σημάτων, χ.η.*). Στην περίπτωση του Arduino ο A/D converter είναι 10bit και συνεπώς μπορεί να διακρίνει 1024 στάθμες. Στα συστήματα μετρήσεων σπάνια συντρέχει κάποιος πρακτικός λόγος για ανάλυση καλύτερη των 16 bits.

2.4.5 Τελεστικοί Ενισχυτές

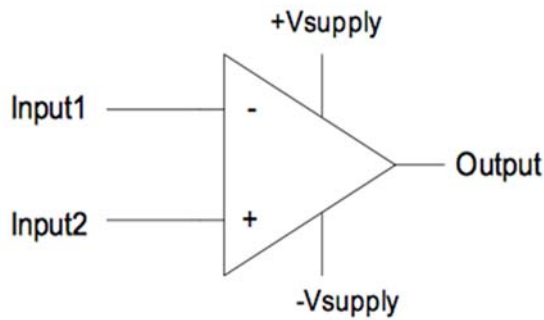
Ένας ενισχυτής στην απλούστερη μορφή του έχει στην είσοδο δύο ακροδέκτες, στα άκρα των οποίων εφαρμόζεται η τάση (σήμα) εισόδου . Έχει επίσης, στην έξοδο δύο ακροδέκτες, στα άκρα των οποίων λαμβάνεται η τάση (σήμα) εξόδου. Από τις σημαντικές παραμέτρους ενός ενισχυτή είναι η απολαβή τάσης η ρεύματος, η σύνθετη αντίσταση εισόδου, η σύνθετη αντίσταση εξόδου και η απόκριση συχνότητας.

Η απολαβή ή κέρδος τάσης(voltage gain) ορίζεται ως το πηλίκο της τάσης εξόδου V_{out} προς την τάση εισόδου V_{in} :

$$Gain = \frac{V_{out}}{V_{in}}$$

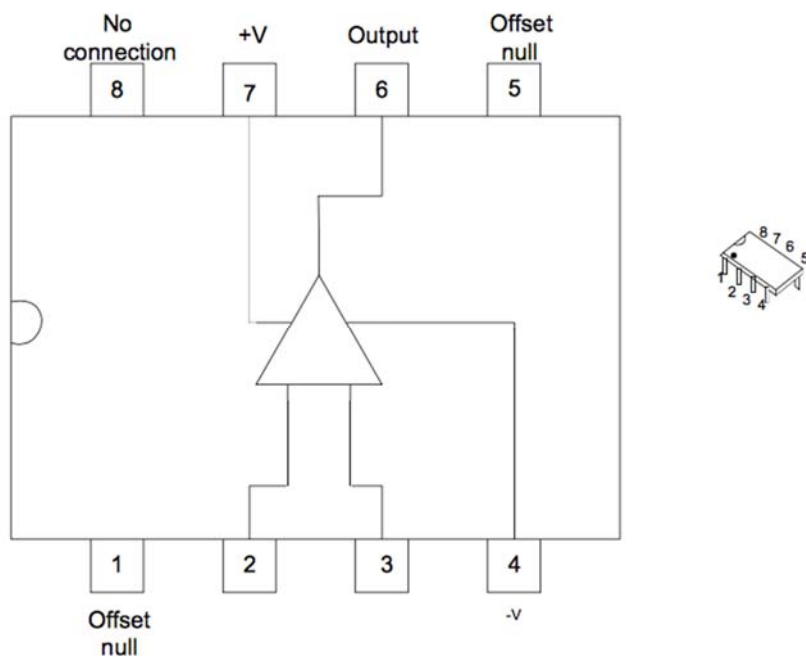
Στο σύστημα που αναπτύσσουμε και πιο συγκεκριμένα, στο κομμάτι μέτρησης αναπνευστικού ρυθμού χρησιμοποιούμε ένα είδος ενισχυτή, τον τελεστικό ενισχυτή.

Ο τελεστικός ενισχυτής αποτελεί τη βασική μονάδα των περισσότερων αναλογικών κυκλωμάτων. Στο παρακάτω σχήμα φαίνεται το σύμβολο του τελεστικού ενισχυτή.



Σχήμα 5: Σύμβολο Τελεστικού Ενισχυτή.

Ένας τελεστικός ενισχυτής έχει δύο εισόδους και μια έξοδο. Απαιτούνται επίσης, δύο επιπλέον ακροδέκτες για την τροφοδοσία του. Η τροφοδοσία ενός τελεστικού ενισχυτή μπορεί να γίνει είτε με την βοήθεια μιας μόνο πηγής τάσης, είτε με την βοήθεια δύο πηγών, οπότε οι τάσεις οι οποίες τροφοδοτούν τους αντίστοιχους ακροδέκτες θα είναι συμμετρικές ως προς την γείωση του κυκλώματος. Επιπλέον, μπορεί να υπάρχουν και άλλοι ακροδέκτες οι οποίοι να επιτρέπουν την προσπέλαση στο εσωτερικό κύκλωμα του τελεστικού ενισχυτή. Έτσι, στην ελάχιστη μορφή του, ένας τελεστικός ενισχυτής απαιτεί πέντε ακροδέκτες και τοποθετείται σε ένα κέλυφος οκτώ ακροδεκτών, οι οποίοι είναι διατεταγμένοι σε δύο σειρές (Dual in line, DIL8), όπως φαίνεται στα παρακάτω σχήματα.



Σχήμα 6: Μορφή και ακροδέκτες τελεστικού ενισχυτή.

Ο τελεστικός ενισχυτής ενισχύει και επεξεργάζεται την διαφορά των σημάτων που εφαρμόζεται στις δύο εισόδους και για αυτό το λόγο λέμε ότι ο τελεστικός ενισχυτής έχει διαφορική είσοδο (differential input). Οι εισοδοι ξεχωρίζουν μεταξύ τους με τα σημεία (+) που συμβολίζει τη μη αναστρέφουσα είσοδο (non-inverting input NI) και (-) που συμβολίζει την αναστρέφουσα είσοδο (inverting input I), (βλ. Σχήμα 6) (Karki, 1998).

2.4.6 Cloud Υπηρεσία

Υπηρεσία Cloud ονομάζεται η κατ' αίτηση διαδικτυακή κεντρική διάθεση υπολογιστικών πόρων (όπως δίκτυο, εξυπηρετητές, εφαρμογές και υπηρεσίες) με υψηλή ευελιξία, ελάχιστη προσπάθεια από τον χρήστη και υψηλή αυτοματοποίηση (NIST, 2011).

Στην cloud υπηρεσία η αποθήκευση, η επεξεργασία και η χρήση δεδομένων, λογισμικού και υπηρεσιών γίνεται διαδικτυακά, μέσω απομακρυσμένων υπολογιστών σε κεντρικά Datacenter. Υπηρεσίες όπως η κατ' αίτηση παροχή

εικονικών μηχανών, το διαδικτυακό ηλεκτρονικό ταχυδρομείο ή τα κοινωνικά δίκτυα συχνά βασίζονται στην τεχνολογία του Υπολογιστικού Νέφους.

Οι χρήστες εξοικονομούν πόρους από την αγορά και συντήρηση λογισμικού, τη συντήρηση ακριβών εξυπηρετητών και εγκαταστάσεων αποθήκευσης δεδομένων. Το SaaS (Software as a Service) αποτελεί μια από τις εκδοχές του Cloud και αναφέρεται σε Λογισμικό που προσφέρεται διαδικτυακά ως Υπηρεσία Cloud (ΕΠΣΕΤ, χ.η.).

2.4.7 Πλατφόρμες IoT

Για την απόκτηση/καταγραφή τιμών από ένα σύστημα IoT είναι πολύ βοηθητικό να διατίθεται μια πλατφόρμα για τη δημιουργία και διαχείριση εφαρμογών, την ανάλυση, την αποθήκευση και την ασφάλιση των δεδομένων.

Σε γενικές γραμμές, πλατφόρμα είναι ένας συνδυασμός hardware με λογισμικό, το οποίο μπορεί να περιλαμβάνει περιβάλλον εργασίας, αποθήκευση, υπολογιστική ισχύ, ασφάλεια, εργαλεία ανάπτυξης και πολλές άλλες λειτουργίες. Οι πλατφόρμες έχουν σχεδιαστεί για να υποστηρίζουν πολλά μικρότερα προγράμματα εφαρμογών (Making sense of Internet of Things platforms, 2017).

Στο Internet των πραγμάτων (IoT), οι πλατφόρμες έχουν σχεδιαστεί για την ανάπτυξη εφαρμογών που παρακολουθούν, διαχειρίζονται και ελέγχουν συνδεδεμένες συσκευές. Οι πλατφόρμες IoT πρέπει να χειρίζονται προβλήματα όπως τη σύνδεση και την εξαγωγή δεδομένων από έναν δυνητικά τεράστιο αριθμό endpoints, τα οποία μερικές φορές βρίσκονται τοποθετημένα σε δύσκολα σημεία με ασταθή συνδεσιμότητα. Λειτουργούν ως γέφυρα έτσι ώστε να υποστηρίξουν τα Application Program Interfaces (APIs), πρωτόκολλα και υπηρεσίες, το περιβάλλον που ενεργοποιεί τις συσκευές ώστε να επικοινωνούν μεταξύ τους καθώς επίσης, εφαρμογές και δίκτυα που είναι απαραίτητα για την επικοινωνία στο IoT. Μερικές από τις πιο γνωστές πλατφόρμες IoT είναι: Google Cloud Platform, ThigWorx, Amazon web Services, Cisco IoT Cloud

Connect, Thingspeak, Microsoft Azure IoT, Oracle Internet of Things και πολλές άλλες. Στην συγκεκριμένη εργασία θα χρησιμοποιήσουμε την πλατφόρμα ThingSpeak.

2.4.8 Διεπαφή Προγραμματισμού Εφαρμογών (Application Program Interfaces, APIs)

Διεπαφή προγραμματισμού εφαρμογών (Application Programming Interface, API) καλούμε «τη διεπαφή που ένα υπολογιστικό σύστημα, βιβλιοθήκη ή εφαρμογή παρέχει προκειμένου να επιτρέψει να γίνουν προς αυτό αιτήσεις από άλλα προγράμματα και/ή ανταλλαγή δεδομένων» (Διεπαφή προγραμματισμού εφαρμογών Wiki, 2017). Με απλά λόγια, API είναι ένα ενδιάμεσο λογισμικό που επιτρέπει την επικοινωνία μεταξύ δύο εφαρμογών. Είναι δηλαδή, ο φορέας που παραδίδει το αίτημα του χρήστη στον πάροχο που ανήκει ο χρήστης και στη συνέχεια επιστρέφει την απάντηση πίσω στον χρήστη.

Το Application Programming Interface (API) είναι ένα σύνολο από παραδοχές και ορισμούς, οι οποίες ορίζουν πώς μια υπηρεσία «καλείται» μέσα από ένα προγραμματιστικό περιβάλλον. Το API επιτρέπει στους προγραμματιστές να έχουν πρόσβαση στις λειτουργικότητες ενός προγράμματος μέσα από καλά καθορισμένες δομές δεδομένων. Με άλλα λόγια, το API περιγράφει ουσιαστικά τη λειτουργία και δομή ενός μέρους ή ολόκληρου προγράμματος και ορίζει τη διεπαφή αυτού με τον έξω κόσμο. Έχοντας την τεκμηρίωση ενός API είναι δυνατός ο προγραμματισμός ενός συστήματος χωρίς να έχουμε άλλου είδους πληροφορίες.

2.4.9 Το πρωτόκολλο HTTP

Το πρωτόκολλο HTTP (HyperText Transfer Protocol) είναι ένα πρωτόκολλο επικοινωνίας που ανήκει στο επίπεδο εφαρμογής και είναι το βασικό πρωτόκολλο στο οποίο στηρίζεται η λειτουργία του Παγκόσμιου Ιστού (World Wide Web). Χρησιμοποιεί κυρίως το πρωτόκολλο μεταφοράς TCP για την δημιουργία συνδέσεων επιπέδου μεταφοράς. Η τρέχουσα κύρια έκδοσή του

είναι η 1.1 που αντικατέστησε την προηγούμενη έκδοση 1.0. Επίσης, είναι σε εξέλιξη η προδιαγραφή και χρήση της επόμενης έκδοσης HTTP/2, η οποία ήδη υποστηρίζεται από πολλούς Εξυπηρετητές Ιστού (Web Servers) και Φυλλομετρητές (Web Browsers). Η λειτουργία του πρωτοκόλλου βασίζεται σε ανταλλαγή μηνυμάτων μεταξύ μιας εφαρμογής πελάτη και μια εφαρμογής εξυπηρετητή και καθορίζει την δομή και την σημασιολογία αυτών των μηνυμάτων. Τα μηνύματα που στέλνονται από την εφαρμογή πελάτη λέγονται Αιτήσεις HTTP (HTTP Requests) και τα μηνύματα που επιστρέφονται από τον Εξυπηρετητή λέγονται Αποκρίσεις HTTP (HTTP Responses) Ένας εξυπηρετητής που υποστηρίζει το πρωτόκολλο HTTP μπορεί να χρησιμοποιεί την προεπιλεγμένη θύρα 80 όπου θα περιμένει για μηνύματα από κάποια εφαρμογή πελάτη ή κάποια άλλη θύρα, η οποία θα πρέπει να αναφέρεται ρητά από την εφαρμογή πελάτη.

2.4.9.1 Αιτήσεις HTTP (HTTP Requests)

Ένα HTTP αίτημα αποτελείται από τρία βασικά τμήματα: την γραμμή αίτησης (Request Line), το Τμήμα Επικεφαλίδων (Header) και το Σώμα Μηνύματος (Message Body). Η γραμμή αίτησης αποτελείται από τρία τμήματα χωρισμένα με ένα κενό χαρακτήρα: (α) την μέθοδο της αίτησης, (β) το όνομα του πόρου αίτησης (URI) και (γ) την έκδοση του HTTP πρωτοκόλλου που ακολουθεί η αίτηση. Στο τέλος της γραμμής αίτησης υπάρχουν οι χαρακτήρες CarriageReturn (\r) και LineFeed (\n). Η μέθοδος μπορεί να είναι μια από τις: GET, POST, HEAD, PUT, DELETE, TRACE, OPTIONS (Εισαγωγή στο πρωτόκολλο HTTP, χ. η.). Στην συγκεκριμένη εργασία θα μας απασχολήσει η μέθοδος **POST**. Η μέθοδος αίτησης POST αιτείται από έναν διακομιστή ιστού να αποδεχθεί τα δεδομένα που περιέχονται στο σώμα του μηνύματος της αίτησης, πιθανότατα για την αποθήκευσή του. Συχνά χρησιμοποιείται κατά τη μεταφόρτωση ενός αρχείου (Post(HTTP) Wiki, 2019).

2.4.10 MATLAB

Το MATLAB (MATrix LABoratory) είναι ένα διαδραστικό περιβάλλον, που διαθέτει μια γλώσσα προγραμματισμού υψηλού επιπέδου και χρησιμοποιείται για αριθμητικούς υπολογισμούς, προγραμματισμό και οπτικοποίηση των αποτελεσμάτων. Επιτρέπει τη διαχείριση πινάκων, τη δημιουργία γραφικών παραστάσεων, την υλοποίηση αλγορίθμων, την ανάλυση δεδομένων, τη μοντελοποίηση συστημάτων, τη δημιουργία διεπιφανειών διασύνδεσης με άλλες γλώσσες προγραμματισμού (όπως C, C++, Java). Διαθέτει ενσωματωμένες εντολές και συναρτήσεις, που βοηθούν στην εκτέλεση μαθηματικών πράξεων και στην υλοποίηση αριθμητικών μεθόδων. Διαθέτει επίσης, πολλές πρόσθετες εργαλειοθήκες, που διευκολύνουν την ανάπτυξη ειδικών εφαρμογών, όπως συμβολικοί υπολογισμοί (symbolic computation), επεξεργασία εικόνων (image processing), στατιστική (statistics), έλεγχο συστημάτων (control system design), νευρωνικά δίκτυα (neural networks), ασαφή λογική (fuzzy logic). Στις πρόσφατες εκδόσεις του λογισμικού ο αριθμός των εργαλειοθηκών έχει ξεπεράσει τις 50 (Αρπατζάνης, χ.η.). Έχοντας την δυνατότητα να εκτελέσουμε κώδικα MATLAB εντός του ThingSpeak, μπορούμε να πραγματοποιούμε online ανάλυση και επεξεργασία των δεδομένων που λαμβάνει η πλατφόρμα.

Κεφάλαιο 3

Βιβλιογραφική Επισκόπηση

Στο πλαίσιο της παρούσας διπλωματικής εργασίας έγινε μια βιβλιογραφική επισκόπηση παρόμοιων συστημάτων-εφαρμογών καθώς και συστημάτων που ανήκουν στην επιστημονική περιοχή του IoT με σκοπό την υποστήριξη της εφαρμογής που αναπτύχθηκε.

Οι Spanò et al. (2016) προτείνουν ένα σύστημα για απομακρυσμένη παρακολούθηση του ηλεκτροκαρδιογραφήματος ECG το οποίο απευθύνεται σε ανθρώπους που χρειάζονται χρόνια παρακολούθηση της υγείας τους σε οικιακό περιβάλλον και το οποίο ενσωματώνεται σε μια ευρύτερη IoT υποδομή. Η χρήση του δεν απαιτεί καμία τεχνική κατάρτιση. Το πρωτότυπο αποτελεί μια πλήρη κάθετη λύση με μια σειρά πλεονεκτημάτων σε σχέση με τις τεχνολογίες αιχμής.

Τα χαρακτηριστικά των πρωτότυπων ECG αισθητήρων με χαμηλή κατανάλωση ενέργειας είναι:

- 1) Η αρχιτεκτονική που παρέχει χαμηλό κόστος ανά προστιθέμενο αισθητήρα/ χρήστη.
- 2) Η δυνατότητα απρόσκοπτης ενσωμάτωσης με άλλα έξυπνα οικιακά συστήματα μέσω μιας ενιαίας υποδομής IoT.

Στο επίπεδο της υποδομής, το σύστημα απομακρυσμένης παρακολούθησης του ECG μπορεί να συγχωνευθεί με άλλα συστήματα βιοϊατρικής παρακολούθησης και παρακολούθησης περιβάλλοντος, διότι η υποδομή IoT επιτρέπει την ομαλή προσθήκη διαφορετικών τύπων αισθητήρων.

Στο τοπικό επίπεδο ανάπτυξης, το σύστημα μπορεί να παρακολουθεί πολλούς ασθενείς στο ίδιο ασύρματο δίκτυο μειώνοντας έτσι το κόστος του συστήματος ανά ασθενή, την πολυπλοκότητα και το οριακό κόστος προσθήκης επιπλέον ασθενών προς παρακολούθηση.

Στο επίπεδο των επιμέρους αισθητήρων, ο ενιαίος ECG αισθητήρας έχει υψηλή ποιότητα σήματος και χαμηλή κατανάλωση ενέργειας παρουσιάζοντας την καλύτερη δυνατή απόδοση.

Η πλατφόρμα έχει τρία κύρια μέρη: α) το δίκτυο αισθητήρων-ενεργοποιητών, β) τον IoT server και γ) τη διασύνδεση χρήστη για απεικόνιση και διαχείριση. Οι αισθητήρες είναι φορητοί ECG αισθητήρες χαμηλού βάρους και αισθητήρες περιβάλλοντος, οι οποίοι συλλέγουν δεδομένα και τα αποστέλλουν σε πραγματικό χρόνο με χρήση ασύρματων πρωτοκόλλων (ZigBee, Bluetooth, WiFi) σε ένα gateway, συνδεδεμένο στο ADSL router του σπιτιού. Ο IoT server μετατρέπει τα ακατέργαστα δεδομένα που προέρχονται από ετερογενείς κόμβους σε μια "καθολική" μορφή, που περιέχει αναγνωριστικό αντικειμένου, τύπο αντικειμένου, μονάδα μέτρησης, πεδίο δεδομένων, γεωγραφική θέση και χρονική σήμανση. Το σύνολο του συστήματος δύναται να διαμορφώνεται και να ελέγχεται μέσω μιας web διεπαφής η οποία είναι προσβάσιμη από οποιονδήποτε υπολογιστή, έξυπνο τηλέφωνο ή tablet που είναι συνδεδεμένο στο διαδίκτυο. Τέλος, στον εξυπηρετητή IoT, τα δεδομένα υγείας μπορούν να συνδυαστούν με άλλα δεδομένα, να συγχωνευθούν, να υποβληθούν σε επεξεργασία από χρήστες ή/και από εξουσιοδοτημένους κλινικούς ιατρούς.

Στην εργασία τους οι Kaleem et al. (2016) αξιολόγησαν κριτικά την υπάρχουσα βιβλιογραφία, η οποία ασχολείται με τους αποτελεσματικούς τρόπους για την ανάπτυξη του IoT στον τομέα της ιατρικής και της έξυπνης υγειονομικής περίθαλψης (e-health). Στη συνέχεια πρότειναν ένα νέο e-Health σημασιολογικό μοντέλο ασθενών. Το προτεινόμενο μοντέλο, που ονομάζεται «k-Healthcare», χρησιμοποιεί τέσσερα επίπεδα. Το επίπεδο αισθητήρα, το επίπεδο δικτύου, το επίπεδο Internet και το επίπεδο υπηρεσιών.

Το επίπεδο αισθητήρα είναι το κατώτερο επίπεδο του συστήματος αποτελώντας την καρδιά του μοντέλου. Υπάρχουν διαφορετικά υποεπίπεδα αισθητήρων σε αυτό το επίπεδο: RTX-4100, ασύρματο EKG δύο απαγωγών, Arduino & Raspberry Pi, αισθητήρας οξυγόνου αίματος, παλμική οξυμετρία και αισθητήρες των Smart Phones.

Το επίπεδο δικτύου παίζει ρόλο-κλειδί στη επικοινωνία για τη διασύνδεση των συσκευών με το WAN με τη χρήση διαφορετικών πρωτοκόλλων (TCP/IP), τεχνολογιών, προτύπων όπως 3G, 4G, ADSL, DSLAM, και Routers.

Το επίπεδο Internet παρέχει τη λειτουργικότητα της αποθήκευσης και διαχείρισης δεδομένων. Όλα τα επίπεδα συνεργάζονται μεταξύ τους αποτελεσματικά και αποδοτικά με σκοπό να παρέχουν μια πλατφόρμα για πρόσβαση σε δεδομένα υγείας ασθενούς με τη χρήση έξυπνων τηλεφώνων. Το κίνητρο των Kaleem U. et al. είναι η χρήση των ενσωματωμένων αισθητήρων των έξυπνων τηλεφώνων για παρακολούθηση e-health. Το προτεινόμενο μοντέλο με το όνομα “k-Healthcare” παρέχει μια πλατφόρμα για αισθητήρες οι οποίοι είναι απευθείας συνδεδεμένοι με το smartphone του ασθενούς για τη λήψη δεδομένων κατά τον χρόνο εκτέλεσης. Τα δεδομένα αυτά αποθηκεύονται και επεξεργάζονται σε cloud πλατφόρμα και μπορούν να ανακτηθούν αργότερα από τους ιατρούς και το ιατρικό προσωπικό για την παρακολούθηση της υγείας των ασθενών.

Οι Tabish et al. (2014) ανέπτυξαν ένα universal σύστημα φροντίδας υγείας με βάση την αρχιτεκτονική 6LowPAN με το όνομα U-healthcare. Το σύστημα εκτελεί παρακολούθηση της κατάστασης της υγείας είτε σε εσωτερικό είτε σε εξωτερικό χώρο. Χρησιμοποιεί μια πλατφόρμα ζωντανής ροής δεδομένων για την ανάγνωση των απομακρυσμένων αισθητήρων παρακολούθησης του ECG και της θερμοκρασίας. Μπορεί να αποθηκεύσει τα δεδομένα των αισθητήρων σε απομακρυσμένους διακομιστές (servers) και χρησιμοποιεί υπηρεσίες cloud, όπως η UbuntuOne. Το προτεινόμενο σύστημα χρησιμοποιεί διάφορες συσκευές και τεχνολογίες όπως router, PC, πρωτόκολλο IPv6, Serial Line Internet Protocol (SLIP), 3G/4G, Microcontroller MSP430 και CC2420, TinyOS

και Contiki Open source operating system, ISR, και WiFi. Έχει τη δυνατότητα για online streaming όταν η ταχύτητα του Internet είναι ικανοποιητική καθώς επίσης και χρήσης σε επείγουσες καταστάσεις.

Στην εργασία τους οι Mshali et al. (2018) παρουσιάζουν μια διεξοδική μελέτη του context-aware computing στον τομέα της υγειονομικής φροντίδας ηλικιωμένων. Έχουν προταθεί πολλοί ορισμοί της έννοιας context στη βιβλιογραφία. Σύμφωνα με τον Dey (2001), το context αντιπροσωπεύει οποιαδήποτε πληροφορία μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει την κατάσταση μιας οντότητας. Μια οντότητα περιλαμβάνει ένα άτομο, έναν τόπο ή ένα αντικείμενο. Στην εργασία τους αναδεικνύονται οι τρέχουσες τεχνολογίες καθώς επίσης, οι απαιτήσεις και οι προκλήσεις στο σχεδιασμό Healthcare Monitoring Systems (HMS) σε «έξυπνα» περιβάλλοντα. Παρουσιάζονται οι κατευθυντήριες γραμμές για να βοηθήσουν τα ενδιαφερόμενα μέρη να κατανοήσουν τα θέματα που πρέπει να αντιμετωπιστούν για τη βελτίωση των υπηρεσιών ηλεκτρονικής υγειονομικής περίθαλψης. Περιγράφονται με λεπτομέρειες τα μέρη των HMS, το πλαίσιο των υπό παρακολούθηση αντικειμένων και ταξινομούνται οι κύριες δραστηριότητες που πρέπει να εξεταστούν για την επεξεργασία σεναρίων Health Smart Home (HSH). Επίσης, γίνεται προσπάθεια να παρουσιαστεί μια ενοποιημένη εικόνα των πιο σημαντικών λειτουργιών και υπηρεσιών που προσφέρει το HMS για την παρακολούθηση και ανίχνευση της ανθρώπινης συμπεριφοράς, συμπεριλαμβανομένων των εννοιών, προσεγγίσεων και τεχνικών επεξεργασίας. Γίνεται μια επισκόπηση της υπάρχουσας έρευνας και αξιολογείται η κατάσταση της τεχνολογίας για εφαρμογές και συστήματα παρακολούθησης της υγειονομικής περίθαλψης. Παρουσιάζονται μέθοδοι μοντελοποίησης και συλλογιστικής καθώς και αλγόριθμοι μάθησης που χρησιμοποιούνται για την ανάλυση και κατανόηση της ανθρώπινης συμπεριφοράς. Αυτές οι δραστηριότητες εξετάζονται σε σενάρια HMS και χρησιμοποιούνται σε διάφορες λειτουργίες που προσφέρονται από τα HMS. Τέλος, οι συγγραφείς κάνουν μια συζήτηση για τα πλεονεκτήματα και κάποιες ερευνητικές προκλήσεις για μελλοντικά υγειονομικά συστήματα.

Οι Baker et al. (2017) επιχειρούν με την εργασία τους να συμβάλλουν μοναδικά στον εντοπισμό όλων των βασικών συστατικών ενός end-to-end IoT συστήματος υγειονομικής περίθαλψης και προτείνουν ένα γενικό μοντέλο που θα μπορούσε να εφαρμοστεί σε όλα τα συστήματα υγειονομικής περίθαλψης που βασίζονται στο διαδίκτυο. Αυτό είναι ζωτικής σημασίας καθώς δεν υπάρχουν ακόμη στη βιβλιογραφία γνωστά end-to-end συστήματα για απομακρυσμένη παρακολούθηση της υγείας. Επιπλέον, η εργασία αυτή παρέχει μια ολοκληρωμένη έρευνα σχετικά με τις τεχνολογίες αιχμής που εμπίπτουν στο προτεινόμενο μοντέλο. Η προσοχή εστιάζεται στους αισθητήρες για την παρακολούθηση διαφόρων παραμέτρων υγείας, στα πρότυπα επικοινωνίας μικρής και μεγάλης εμβέλειας και στις τεχνολογίες cloud. Στην συγκεκριμένη εργασία εξετάζεται κάθε βασικό στοιχείο ενός IoT συστήματος υγειονομικής περίθαλψης, τόσο ξεχωριστά όσο και ως σύστημα. Παρουσιάζονται οι αισθητήρες που μπορούν να χρησιμοποιηθούν και οι οποίοι είναι μη επεμβατικοί. Οι βασικοί αισθητήρες είναι πέντε: τρεις για την παρακολούθηση των ζωτικών σημάτων παλμού, αναπνευστικού ρυθμού και θερμοκρασίας του σώματος και δύο επιπλέον για την παρακολούθηση της αρτηριακής πίεσης και του οξυγόνου του αίματος, παράμετροι, οι οποίες καταγράφονται συνήθως σε νοσοκομειακό περιβάλλον. Επίσης, γίνεται μια παρουσίαση επιπλέον αισθητήρων που μπορούν να χρησιμοποιηθούν όπως, οι αισθητήρες ηλεκτροκαρδιογραφήματος ECG, ηλεκτροεγκεφαλογραφήματος EEG καθώς και αισθητήρες για την ανίχνευση πτώσης (accelerometer, gyroscope, magnetometer and barometer). Σημαντική αναφορά γίνεται και στο κομμάτι των επικοινωνιών μικρής και μεγάλης εμβέλειας σε αυτά τα συστήματα.

Οι Smirek et.al. (2016) αναλύουν τις ομοιότητες και διαφορές μεταξύ δύο συστημάτων, του έργου Eclipse Smart Home (ESH) και του Universal Remote Control (URC), τα οποία χαρακτηρίζονται από την έλλειψη κατάλληλων διεπαφών χρήστη και παρουσιάζουν προβλήματα χαμηλής διαλειτουργικότητας μεταξύ διαφορετικών έξυπνων οικιακών συστημάτων. Το ESH παρέχει ένα ευέλικτο, διαμορφωμένο πλαίσιο για έξυπνες οικιακές λύσεις και λύσεις σε υποβοηθούμενο περιβάλλον (Ambient Assisted Living (AAL)), με έμφαση σε ετερογενή περιβάλλοντα. Στοχεύει στην αντιμετώπιση της

σημερινής πολύκατακερματισμένης αγοράς για έξυπνα οικιακά συστήματα και συσκευές IoT και όχι στο να ορίζει ένα κοινό πρωτόκολλο επικοινωνίας το οποίο πρέπει να υποστηρίζεται από διαφορετικές συσκευές. Το URC (OpenURC, χ.η.) είναι ένα πλαίσιο που σχεδιάστηκε με σκοπό να επιτρέπει εξατομικευμένες και ανταλλάξιμες διεπαφές χρηστών. Η κύρια ιδέα του URC είναι να επιτρέπει σε κάθε χρήστη να ελέγχει οποιαδήποτε συσκευή ή υπηρεσία (στόχο) με τη διεπαφή χρήστη που ταιριάζει καλύτερα στις ανάγκες του.

Οι Liu et.al. (2016) πραγματοποίησαν μια βιβλιογραφική ανασκόπηση για σπίτια που είναι εξοπλισμένα με αισθητήρες και τεχνολογίες που υποστηρίζουν την παρακολούθηση της υγείας στο σπίτι για ηλικιωμένους. Η μελέτη ενίσχυσε την προσπάθεια για εφαρμογή της παρακολούθησης της υγείας στο σπίτι και την παρακολούθηση της λειτουργίας και της κατάστασης της καρδιάς των ηλικιωμένων ασθενών.

Οι Cruz et.al. (2014) πραγματοποίησαν μια συστηματική ανασκόπηση η οποία αποσκοπούσε στο να παράσχει μια περιεκτική περιγραφή των μεθόδων που χρησιμοποιήθηκαν στις παρεμβάσεις τηλεπαρακολούθησης στο σπίτι για ασθενείς που πάσχουν από χρόνια αποφρακτική πνευμονοπάθεια (ΧΑΠ) (COPD). Επιπρόσθετα, διερευνήθηκε η προσήλωση και ικανοποίηση των ασθενών από τη χρήση συστημάτων τηλεπαρακολούθησης. Με την ανασκόπηση αυτή επισημάνθηκαν σημαντικές μεθοδολογικές πτυχές που πρέπει να λαμβάνονται υπόψη από τους ερευνητές και τους επαγγελματίες του τομέα υγείας όταν αναπτύσσονται συστήματα τηλεπαρακολούθησης στο σπίτι για ασθενείς με ΧΑΠ ενώ ταυτόχρονα παρέχονται συστάσεις για παρεμβάσεις στον τομέα της υγείας.

Οι Sailunaz et.al. (2016) προτείνουν ένα πλαίσιο που πληροί τις βασικές προϋποθέσεις για την παροχή ενός κατάλληλου συστήματος υγειονομικής περίθαλψης για ανθρώπους σε αγροτικές περιοχές αναπτυσσόμενων χωρών και το οποίο βασίζεται σε υπηρεσίες cloud. Το ιδιωτικό απόρρητο και η εμπιστευτικότητα των πληροφοριών για την υγεία του ασθενούς εξασφαλίζονται με τη χρήση μεθόδων κρυπτογράφησης με βάση την

ταυτότητα (Identity Based Encryption).

Στην εργασία τους οι Bennett et.al. (2017) πραγματοποιούν μια μελέτη στην οποία εξετάζουν το ιστορικό της υγειονομικής φροντίδας Smart Home, τους σημερινούς τομείς έρευνας και τις πιθανές περιοχές μελλοντικής έρευνας. Παρουσιάζονται μοναδικές κατηγοριοποιήσεις για τις Δραστηριότητες Καθημερινής Ζωής (Activities of Daily Living, ADL) και τους προσωπικούς αισθητήρες. Επιπλέον, αναφέρονται στην εφαρμογή τεχνολογίας «έξυπνου σπιτιού» για υγειονομική φροντίδα. Αυτές οι τεχνολογίες μπορούν να δημιουργήσουν λύσεις υγειονομικής περίθαλψης 24 ωρών/7 ημερών για τους ευάλωτους πληθυσμούς. Η ανάλυση των δεδομένων, είτε εντός του «έξυπνου σπιτιού» είτε εξωτερικά, από μια οντότητα επεξεργασίας δεδομένων μπορεί να οδηγήσει σε έγκαιρη διάγνωση πιθανών προβλημάτων καθώς και σε υποστήριξη έκτακτης ανάγκης εάν είναι απαραίτητο. Όλες αυτές οι λειτουργίες πρέπει να συνοδεύονται από ένα σύστημα προειδοποιήσεων. Η επίτευξη όλων αυτών ήταν εφικτή καθώς οι συσκευές γίνονται μικρότερες και ισχυρότερες, οι μεθοδολογίες λογισμικού ωριμάζουν και η υιοθέτηση της τεχνολογίας από τους πληθυσμούς αυξάνεται συνεχώς. Τέλος, γίνεται λεπτομερή παρουσίαση των κατηγοριών (περιβάλλοντος, φορετοί, κτλ) και των τύπων των αισθητήρων, των ελεγκτών, των gateways, των ενεργοποιητών (actuators), των έξυπνων συσκευών και του λογισμικού που χρησιμοποιείται στις εφαρμογές «έξυπνου σπιτιού».

Οι Duregger et al. (2014) πρότειναν ένα σύστημα για την παρακολούθηση παιδιατρικών ασθενών με νευροβλάστωμα στο σπίτι, που επικεντρώνεται σε ένα ευρύ φάσμα ζωτικών σημάτων (καρδιακού ρυθμού, αρτηριακής πίεσης, θερμοκρασίας, βάρος σώματος, αλλοιώσεις του δέρματος κ.ά.) με τη χρήση μιας εφαρμογής για κινητό τηλέφωνο και ένα set από αισθητήρες. Αυτό που είναι ιδιαίτερα ενδιαφέρον εν προκειμένω, είναι η συγχώνευση δεδομένων αισθητήρων με υποκειμενική ανατροφοδότηση για να δοθεί μια συνολική εικόνα της καλής κατάστασης του ασθενούς. Μια Android εφαρμογή με το όνομα MobileMonitor συλλέγει δεδομένα από τις συσκευές μέτρησης και τις μεταδίδει σε ένα backend σύστημα μέσω του πρωτοκόλλου HTTPS, το οποίο

προσθέτει ασφάλεια. Επιπλέον, το πρωτόκολλο Near-Field Communications (NFC) χρησιμοποιείται σε συνδυασμό με μια γραφική απεικόνιση A4 της κατάστασης των ασθενών (καλό, μεσαίο, κακό) για τη λήψη υποκειμενικών δεδομένων. Οι αισθητήρες μαζί με τα δεδομένα ευεξίας μπορούν να χρησιμοποιηθούν από επαγγελματίες υγείας, οι οποίοι με τη σειρά τους θα κρίνουν εάν ο ασθενής μπορεί να παραμείνει στο σπίτι για μεγαλύτερο χρονικό διάστημα μεταξύ των επισκέψεων στο νοσοκομείο. Σε περίπτωση που παρατηρηθούν συνθήκες επιδείνωσης οι ειδοποιήσεις δίδονται στον αντίστοιχο ογκολόγο.

Ένας από τους μεγαλύτερους κινδύνους που αντιμετωπίζουν οι ηλικιωμένοι στο σπίτι είναι η πιθανότητα πτώσης. Οι Li et al. (2014) προτείνουν μια συσκευή σε μορφή βραχιολιού, η οποία συνοδεύεται από μια διασύνδεση κινητής συσκευής για την ανίχνευση πτώσης των ηλικιωμένων σε πραγματικό χρόνο. Προκειμένου να ανιχνευθεί αν ο χρήστης πέφτει, οι υπάρχουσες μέθοδοι αναγνώρισης ανθρώπινης δραστηριότητας μπορούν να ταξινομηθούν στις εξής τέσσερις κατηγορίες: σταθερές μέθοδοι κατωφλίου, στρατηγικές αναγνώρισης προτύπων, συμβατική ή ασαφή λογική (fuzzy logic) και τεχνητά νευρωνικά δίκτυα. Το πρόβλημα είναι ότι με τις παραδοσιακές μεθόδους λογισμικού η ακρίβεια ανίχνευσης είναι συνήθως χαμηλή, γεγονός που οδηγεί σε ιατρικά μη ασφαλείς ενδείξεις. Επιπλέον, αυτοί οι αλγόριθμοι απαιτούν συχνά πολύπλοκους υπολογιστικούς πόρους, πράγμα που σημαίνει γρήγορη κατανάλωση των ενσωματωμένων μπαταριών. Έτσι, η πρόταση των Li et al. είναι ένα “έξυπνο” βραχιόλι που παρέχει μια 24ωρη υπηρεσία ανίχνευσης πτώσης με ρυθμό ανίχνευσης μέχρι 93% με απλοποιημένη λύση που εξοικονομεί διάρκεια ζωής της μπαταρίας. Αυτό μπορεί στη συνέχεια να συνδυαστεί με ένα σύστημα φορετού αερόσακου που ενεργοποιείται όταν ανιχνευθεί πτώση. Το βραχιόλι έχει αρχιτεκτονική σχεδίαση τριών επιπέδων: επίπεδο εφαρμογής (λογισμικό και στατιστικά στοιχεία), επίπεδο δικτύου (επικοινωνίες) και επίπεδο αισθητήρων (επιταχυνσιόμετρο 3 αξόνων, γυροσκόπιο 3 αξόνων, πυξίδα 3 αξόνων και αισθητήρας παλμών). Τα δεδομένα από τους αισθητήρες αποστέλλονται σε ένα κινητό τηλέφωνο μέσω Wi-Fi ή Bluetooth. Πρέπει να σημειωθεί ότι, παρόλο που η έρευνα αυτή επικεντρώνεται

στην ανίχνευση πτώσης, η τεχνολογία και η προσέγγιση μπορούν να χρησιμοποιηθούν για την επέκταση της έρευνας στη διάγνωση παθήσεων της καρδιάς και στην παρακολούθηση δραστηριότητας.

Το σύστημα μαθαίνει με βάση τα δεδομένα της εκπαίδευσης που εξάγονται για να καταλαβαίνει αποτελεσματικά τι μοιάζει με μια πτώση. Το σύστημα αυτό είναι παρόν στις περισσότερες εμπορικές λύσεις όπως αυτές της Fitbit, της Apple και της Garmin, των οποίων οι συσκευές εντοπισμού περιορίζονται από τη διάρκεια ζωής της μπαταρίας που μετράτε σε ώρες και όχι εβδομάδες.

Οι Bayo-Monton et.al. (2018) στην εργασία τους αξιολόγησαν με τη χρήση πρωτοτύπου μια λύση για γρήγορη συγχώνευση πληροφοριών με βάση πέντε αισθητήρες υγείας και δύο χαμηλού κόστους development boards: Arduino και Raspberry Pi. Η μελέτη τους αποσκοπούσε στην αξιολόγηση του βαθμού στον οποίο οι φορητές τεχνολογίες είναι σε θέση να ενσωματώσουν φορητούς αισθητήρες, συγκρίνοντας δύο σενάρια ανάπτυξης: το Raspberry Pi 3 και τον προσωπικό υπολογιστή. Η ανάλυσή τους επιβεβαίωσε ότι οι φορητές συσκευές ($p < 0.01$) είναι κατάλληλες για τη στήριξη της μετάδοσης και ανάλυσης των βιομετρικών σημάτων σε κλιμακούμενα συστήματα τηλεϊατρικής. Στην εργασία αυτή παρουσιάζεται και αξιολογείται ένα ενσωματωμένο κατανεμημένο σύστημα με ένα προσαρμοσμένο πρωτόκολλο Lite για τη σύνδεση φτηνών συσκευών υγείας βασισμένων σε λύσεις πρωτοτύπων eHealth (Arduino, Raspberry Pi και kit βιοσημάτων). Για να εκτιμηθεί ο βαθμός σύγκρισης των φορητών συσκευών με σταθερά συστήματα, η συνολική ανάπτυξη έγινε με δύο διαφορετικά σενάρια ανάπτυξης: (1) Υπολογιστής Desktop με λειτουργικό σύστημα Windows 10 και (2), Raspberry Pi με λειτουργικό σύστημα Windows 10 Core IoT. Στα δύο συστήματα ενσωματώθηκαν οι ίδιες λειτουργίες για την ανάκτηση δεδομένων από τους αισθητήρες υγείας και μια ιστοσελίδα HTML5 για να συγκριθούν και να προκύψει ένας βασικός δείκτης απόδοσης: η χρονική καθυστέρηση μεταξύ της απόκτησης και της εμφάνισης της βιοσήμανσης.

Το συμπέρασμα στο οποίο καταλήγουν είναι ότι οι φορητοί υπολογιστές προάγουν νέες ευκαιρίες για να επεκταθεί η χρήση φορητών συσκευών στην

έρευνα για την υγειονομική περίθαλψη. Πλεονεκτήματα του προτεινόμενου συστήματος είναι οι ανοιχτές προδιαγραφές του πρωτοκόλλου, η μέθοδος ανοιχτής επικοινωνίας με τη χρήση τυποποιημένων δομών επικοινωνίας (βάσει XML) και η χορογραφία, καθώς και η άμεση σύνδεση με υλικό ανοιχτού κώδικα. Τα αποτελέσματα αυτά καθιστούν δυνατή την ενίσχυση του συστήματος για άλλους τομείς, όπως αισθητήρες Ambient Assisted Living (AAL) και έξυπνοι οικιακοί αισθητήρες.

Οι Rahmani et.al. (2018) χρησιμοποιούν μια στρατηγική τοποθέτηση σημείων γεφύρωσης (gateways) στην άκρη του δικτύου για να παρέχουν υπηρεσίες υψηλού επιπέδου όπως, η τοπική αποθήκευση, η επεξεργασία τοπικών δεδομένων σε πραγματικό χρόνο και η ενσωματωμένη άντληση δεδομένων. Με τον τρόπο αυτό παρουσιάζεται μια πύλη για e-Health εφαρμογές. Στη συνέχεια, εφαρμόζουν ένα concept fog computing σε IoT συστήματα υγειονομικής περίθαλψης σχηματίζοντας ένα γεωδιανεμημένο ενδιάμεσο επίπεδο νοημοσύνης μεταξύ των κόμβων των αισθητήρων και του cloud.

Οι Gomez et al. (2016) μελέτησαν την ικανότητα ενός IoT συστήματος να ενσωματώνει συσκευές που μπορούν να συνδεθούν στο Internet. Παρείχαν, επίσης, πληροφορίες σχετικά με την κατάσταση υγείας των ασθενών και την ενημέρωση των γιατρών σε πραγματικό χρόνο. Στη μελέτη τους ανέπτυξαν μια αρχιτεκτονική που βασίζεται σε οντολογίες, η οποία έχει την ικανότητα να παρακολουθεί την υγεία και να παρέχει συστάσεις ρουτίνας στους ασθενείς με χρόνιες παθήσεις.

Οι Znjaj et.al. (2016) προτείνουν ένα σύστημα υπενθύμισης και παρακολούθησης λήψης φαρμάκων που αφορά στην ασφάλεια της υγείας, χρησιμοποιώντας ένα σύστημα IoT με αισθητήρια στοιχεία και ασύρματη μονάδα. Ένα IoT cloud ανοιχτού κώδικα αποτελεί μια αποτελεσματική προσέγγιση για την αποθήκευση δεδομένων των αισθητήρων.

Στην εργασία τους οι Deekshith Gupta et.al. (2015) ασχολούνται με το σχεδιασμό ενός συστήματος για την απομακρυσμένη συνεχή παρακολούθηση

του ηλεκτροκαρδιογραφήματος (ECG) και άλλων ζωτικών παραμέτρων με χρήση του Raspberry Pi. Τα δεδομένα των μετρήσεων αυτών αποθηκεύονται σε μια βάση δεδομένων και μπορούν να εμφανίζονται σε έναν ιστότοπο στον οποίο έχει πρόσβαση μόνο εξουσιοδοτημένο προσωπικό. Ο βασικός στόχος του συστήματος είναι η ενημέρωση των δεδομένων στη βάση δεδομένων και η ειδοποίηση των γιατρών για οποιαδήποτε διαταραχή. Το πρώτο επιτυγχάνεται με τη χρήση της MySQLdb για τη σύνδεση του Raspberry Pi στη βάση δεδομένων, ενώ για τις ειδοποιήσεις χρησιμοποιούνται συνδυαστικά το Raspberry Pi και μια μονάδα GSM. Το σύστημα αυτό έχει πολλά μελλοντικά πεδία, καθώς τα δεδομένα που συλλέγονται από την παρακολούθηση είναι πολύτιμα και μπορούν να χρησιμοποιηθούν για επιστημονική έρευνα από την ιατρική κοινότητα. Με τον προσδιορισμό των μοτίβων στις παρατηρούμενες παραμέτρους μπορεί να προβλεφθεί η φύση της ασθένειας. Η μελέτη δίνει έμφαση κυρίως στο σχεδιασμό του συστήματος και στον αλγόριθμο που χρησιμοποιείται για την εκτέλεση της εργασίας.

Η εργασία των Rahman et al (2019) προτείνει ένα ευφύες σύστημα παρακολούθησης ασθενούς το οποίο παρακολουθεί αυτόματα την κατάσταση υγείας του ασθενούς μέσω διαφόρων αισθητήρων. Στη συνέχεια, τα δεδομένα επεξεργάζονται χρησιμοποιώντας ένα Raspberry Pi και οι χρήσιμες πληροφορίες αποθηκεύονται σε ένα cloud του IoT συστήματος. Κατά κύριο λόγο, το σύστημα εξάγει το ECG βιοσήμα χρησιμοποιώντας έναν αισθητήρα ηλεκτροκαρδιογραφήματος (ECG). Μέσω της συνεχούς παρακολούθησης και του γραφήματος των πληροφοριών του ασθενούς, οι γιατροί/νοσηλευτές/συγγενείς μπορούν να ελέγξουν από απόσταση την κατάσταση του ασθενούς. Επιπλέον, εάν η κατάσταση γίνει κρίσιμη, αποστέλλεται μια ειδοποίηση με τη χρήση SMS σε έναν προκαθορισμένο αριθμό έκτακτης ανάγκης, έτσι ώστε ο γιατρός/νοσοκόμος/ συγγενής να ενημερωθεί και να έχει την ευκαιρία να ξεκινήσει βιντεοκλήση. Οι μεταβλητές που παρακολουθεί το σύστημα είναι: σήμα ηλεκτροκαρδιογραφήματος (ECG) με τη χρήση μιας απαγωγής και θερμοκρασία με τη χρήση αισθητήρα θερμοκρασίας.

Οι Parihar et al. (2017) υλοποίησαν ένα ασύρματο σύστημα μέτρησης του

καρδιακού ρυθμού και της θερμοκρασίας σώματος με βάση τον μικροελεγκτή ATmega328 (Arduino Uno). Η πρότασή τους περιλαμβάνει τους αισθητήρες οι οποίοι μετράνε τον καρδιακό ρυθμό και τη θερμοκρασία σώματος και ελέγχονται από τον μικροελεγκτή. Τα αποτελέσματα των μετρήσεων απεικονίζονται σε μια οθόνη LCD. Το ασύρματο μέρος του συστήματος που ανέπτυξαν χρησιμοποιείται για την ασύρματη αποστολή (χρήση ραδιοσυχνοτήτων) των δεδομένων μέτρησης σε μια απομακρυσμένη τοποθεσία. Για το λόγο αυτό, το σύστημα περιλαμβάνει ένα RF module για την αποστολή των δεδομένων και ένα TX module για την λήψη τους.

Οι μετρήσεις της θερμοκρασίας και του καρδιακού ρυθμού ασθενούς με χρήση του IoT απασχόλησαν τους Bhaskar και Manage (2017). Πρόκειται για ένα σύστημα το οποίο μπορεί εύκολα να στέλνει πληροφορίες σε πραγματικό χρόνο σε πολλούς χρήστες, καθώς επίσης και μηνύματα συναγερμού στο γιατρό μέσω διαδικτύου στην περίπτωση που ο ασθενής βρίσκεται σε κρίσιμη κατάσταση. Επιπλέον, όταν οι τιμές των μετρήσεων ξεπεράσουν κάποια προκαθορισμένα όρια, ένας βομβητής θα ηχήσει ειδοποιώντας με τον τρόπο αυτό το άτομο που φροντίζει τον ασθενή. Για τις μετρήσεις των παραπάνω παραμέτρων χρησιμοποίησαν τον Linux based υπολογιστή Raspberry Pi3. Ο αισθητήρας θερμοκρασίας-υγρασίας DHT11 λαμβάνει την θερμοκρασία από το σώμα του ασθενή και στέλνει την πληροφορία στο Raspberry Pi3. Η πληροφορία που λαμβάνεται από τον αισθητήρα καρδιακού ρυθμού είναι σε αναλογική μορφή. Για το λόγο αυτό χρησιμοποίησαν έναν μετατροπέα Analog-to-Digital (ADS1115) για να δοθεί η πληροφορία στο Raspberry Pi3 σε ψηφιακή μορφή μέσω της σειριακής γραμμής δεδομένων του ADS1115. Για την απεικόνιση όλων αυτών των πληροφοριών έγινε χρήση της ιστοσελίδας Ubidots η οποία είναι ένας web server που προσφέρει υπηρεσίες IoT. Έτσι, με τη χρήση μόνο μιας διεύθυνσης IP ένας χρήστης μπορεί να παρακολουθήσει την κατάσταση υγείας του ασθενή από οποιοδήποτε μέρος του κόσμου μέσω ενός υπολογιστή ή μιας mobile συσκευής.

Οι Alam et al. (2019) στην εργασία τους ανέπτυξαν ένα σύστημα για την μέτρηση του ρυθμού της αναπνοής. Ο αναπνευστικός ρυθμός είναι ένας από

τους ζωτικούς δείκτες της κατάστασης υγείας ενός ατόμου. Είναι ο ρυθμός της αναπνοής του ατόμου σε ηρεμία και μετριέται σε αναπνοές το λεπτό (breaths per minute, **bpm**). Οι μετρήσεις έγιναν με τη χρήση ενός push switch το οποίο τοποθετήθηκε σε μια ζώνη που εφαρμόζεται στο στήθος του ασθενούς. Καθώς το στήθος κινείται κατά τη διάρκεια της αναπνοής, ο διακόπτης (push switch) του κυκλώματος πατιέται και έτσι ο αναπνευστικός ρυθμός του ατόμου μετριέται και απεικονίζεται σε μια Bluetooth εφαρμογή. Ο διακόπτης αυτός συνδέεται σε μια αναλογική είσοδο ενός μικροελεγκτή Arduino Uno. Το Arduino μετράει τις φορές που λαμβάνει σήμα από τον διακόπτη για μια χρονική διάρκεια 1 λεπτού και αυτή η τιμή αποτελεί τον ρυθμό της αναπνοής του ατόμου σε bpm. Στο Arduino συνδέεται ένα Bluetooth module (HC-05) το οποίο μεταδίδει τις μετρήσεις που γίνονται σε μια smart συσκευή όπως, για παράδειγμα, ένα smart κινητό τηλέφωνο. Στο κινητό τηλέφωνο υπάρχει εγκατεστημένη η εφαρμογή “Bluetooth Terminal” στην οποία απεικονίζονται τα δεδομένα.

Στην εργασία τους οι Rajani et al. (2016) ανέπτυξαν ένα σύστημα για τη συνεχή παρακολούθηση ασθενών με καρδιολογικά προβλήματα. Το σύστημά τους αποτελείται από δύο μέρη:

- 1) μια εφαρμογή για την παρακολούθηση του καρδιακού ρυθμού του ασθενούς η οποία έχει την δυνατότητα, όταν ανιχνεύεται ανωμαλία στον καρδιακό ρυθμό, να αποστέλλει μήνυμα στον γιατρό που παρακολουθεί τον ασθενή με τη χρήση ενός 3G Shield. Ο καρδιακός ρυθμός ανιχνεύεται με τη χρήση της τεχνικής της φωτοπληθυσμογραφίας (PPG)
- 2) ασύρματη επικοινωνία μέσω του Arduino Uno που αναπτύχθηκε με τέτοιο τρόπο ώστε να εκτελεί φωνητική κλήση, αποστολή SMS μέσω της διασύνδεσης με το 3G shield και με τη χρήση εντολών AT (AT commands). Το πρόγραμμα γράφτηκε με τέτοιο τρόπο ώστε να υπάρχουν πολλές επιλογές για την αποστολή μηνυμάτων συναγερμού όπως δυναμική κλήση, επείγουσα κλήση (αστυνομία, ασθενοφόρο, πυροσβεστική), αποστολή μηνύματος, λήψη κλήσης, αποσύνδεση κλήσης, επανάληψη κλήσης, προώθηση μηνύματος.

Οι Mohamad et al. (2019) σχεδίασαν και πρότειναν ένα IoT based σύστημα

παρακολούθησης της κατάστασης υγείας ασθενούς με την χρήση του αλγορίθμου ανάλυσης κύριου στοιχείου (Principal Component Analysis, PCA). Το προτεινόμενο IoT σύστημα στέλνει σήματα ECG που λαμβάνονται από τον ασθενή στην IoT cloud υπηρεσία Thingspeak. Ο αλγόριθμος ανάλυσης (PCA αλγόριθμος) εκτελείται στην IoT cloud πλατφόρμα ThingSpeak, και με τη χρήση του MATLAB ταξινομεί την καρδιακή ασθένεια συγκρίνοντας το ληφθέν σήμα με διάφορα σήματα ECG, που είναι αποθηκευμένα στις βάσεις δεδομένων των καναλιών του Thingspeak.

Τέλος, οι Koufi et al. (2018) σχεδίασαν και ανέπτυξαν ένα σύστημα συνεχούς παρακολούθησης της κατάστασης του ύπνου για ασθενείς που πάσχουν από άπνοιες κατά τη διάρκεια του ύπνου. Για το λόγο αυτό κατασκευάστηκε και σχεδιάστηκε ένα ηλεκτρονικό μοντέλο για την διάγνωση της άπνοιας κατά τη διάρκεια του ύπνου. Πρόκειται για μια συσκευή (μοντέλο) η οποία τοποθετείται στο στήθος του ασθενή και έχει τη δυνατότητα μέτρησης πολλών ζωτικών βιοσημάτων/παραμέτρων, όπως η αίσθηση της κίνησης στο στήθος (καρδιακός παλμός) και ο κορεσμός του οξυγόνου μέσω παλμικής οξυμετρίας, ο οποίος επηρεάζεται από τον ρυθμό της αναπνοής. Επιπλέον, γίνεται μέτρηση της θερμοκρασίας και της υγρασίας του ασθενούς με τη χρήση ειδικών αισθητήρων.

Το σύστημα που αναπτύξαμε σε αυτή την εργασία είναι ουσιαστικά μια εφαρμογή για την κατ' οίκον παρακολούθηση της κατάστασης της υγείας ασθενούς με δυνατότητες αντιμετώπισης επειγόντων καταστάσεων μέσω των ειδοποιήσεων και συναγεργμών που διαθέτει. Η νέα προτεινόμενη περιοχή, σε αυτή την εργασία, είναι η μέτρηση του αναπνευστικού ρυθμού (Respiratory Rate, RR) σε συνδυασμό με την μέτρηση των άλλων δύο παραμέτρων (καρδιακού παλμού, θερμοκρασία σώματος). Ειδικότερα η μέτρηση του αναπνευστικού ρυθμού επιτυγχάνεται με την χρήση ενός NTC θερμίστορ χαμηλού κόστους και χαμηλής ενεργειακής κατανάλωσης. Το γεγονός αυτό μας απαλλάσσει από τη χρήση πολύπλοκων κυκλωμάτων και ακριβών αισθητήρων τα οποία είναι απαραίτητα για την μέτρηση του αναπνευστικού ρυθμού μέσω

των άλλων μεθόδων που συναντήσαμε στην βιβλιογραφία και παρουσιάζουμε συνοπτικά στο δεύτερο κεφάλαιο.

Ο ρυθμός της αναπνοής είναι ένα βιοσήμα του οποίου η ανωμαλία είναι μια σημαντική ένδειξη για σοβαρά κλινικά συμβάντα. Η παρακολούθηση και η μέτρηση του αναπνευστικού ρυθμού ενός ασθενούς είναι διαδικασίες απαραίτητες για ποικίλους λόγους. Οποιαδήποτε διαταραχή στο σύστημα του σώματος που προκαλεί υποξαιμία ή υπερκαρδία μπορεί να ανιχνευθεί με μέτρηση του RR. Σημαντικές ενδείξεις μπορούν επίσης να καταγραφούν σε αρκετές περιπτώσεις ιατρικών εξετάσεων, όπως η μελέτη ύπνου, έλεγχος και παρακολούθηση πνευμονικής λειτουργίας, τεστ κοπώσεως και παιδιατρικής παρακολούθησης. Τα προβλήματα στην μέτρηση του RR, τα οποία εντοπίζονται και στη βιβλιογραφία, αφορούν κυρίως την αξιοπιστία της μέτρησης καθώς επίσης και τη δυσκολία αποδοχής της εξέτασης από πλευράς του ασθενούς εξαιτίας του επιβαρυντικού εξοπλισμού που πρέπει να φέρει κατά τη διάρκεια της εξέτασης. Η χρήση, ωστόσο, του θερμίστορ ως αισθητήρα τοποθετημένο σε μια ρινική μάσκα μειώνει σε μεγάλο βαθμό την αίσθηση δυσφορίας του ασθενούς κατά τη διάρκεια της μέτρησης διατηρώντας την αξιοπιστία της σε ικανοποιητικό επίπεδο. Για το λόγο αυτό το σύστημα που προτείνουμε στην παρούσα εργασία μπορεί υπό προϋποθέσεις να έχει εφαρμογή σε παιδιά-ασθενείς. Ο αναπνευστικός ρυθμός (RR) διαδραματίζει θεμελιώδη ρόλο στην κλινική αξιολόγηση ρουτίνας για τη διάγνωση, πρόγνωση και θεραπεία των ασθενειών σε παιδιά. Η ακριβής μέτρηση του αναπνευστικού ρυθμού είναι υψίστης σημασίας επειδή ο αυξημένος RR είναι δείκτης σοβαρής αναπνευστικής νόσου και είναι το κύριο διαγνωστικό κριτήριο για παιδική πνευμονία. Για τους παραπάνω λόγους το σύστημα που αναπτύξαμε θα μπορούσε να μετεξελιχθεί σε ένα σύστημα παρακολούθησης και αντιμετώπισης εκτάκτων περιστατικών εστιασμένο σε παιδιά.

Ένα δεύτερο, σημαντικό πλεονέκτημα του συστήματος που αναπτύξαμε είναι η χρήση της IoT πλατφόρμας ThingSpeak. Η πλατφόρμα ενσωματώνει όπως προ είπαμε τη γλώσσα MATLAB για την εύκολη και γρήγορη ανάλυση των IoT δεδομένων. Το MATLAB υποστηρίζει την ανάπτυξη cloud για IoT συστήματα μικρής έως μεσαίας κλίμακας, μέχρι μεγάλα συστήματα επιχειρήσεων. Το

Thingspeak είναι η μοναδική IoT web υπηρεσία ανοιχτού κώδικα που προσφέρει την ανάλυση δεδομένων στην πλατφόρμα Matlab.

Κεφάλαιο 4

Ανάλυση και Σχεδιασμός του Συστήματος

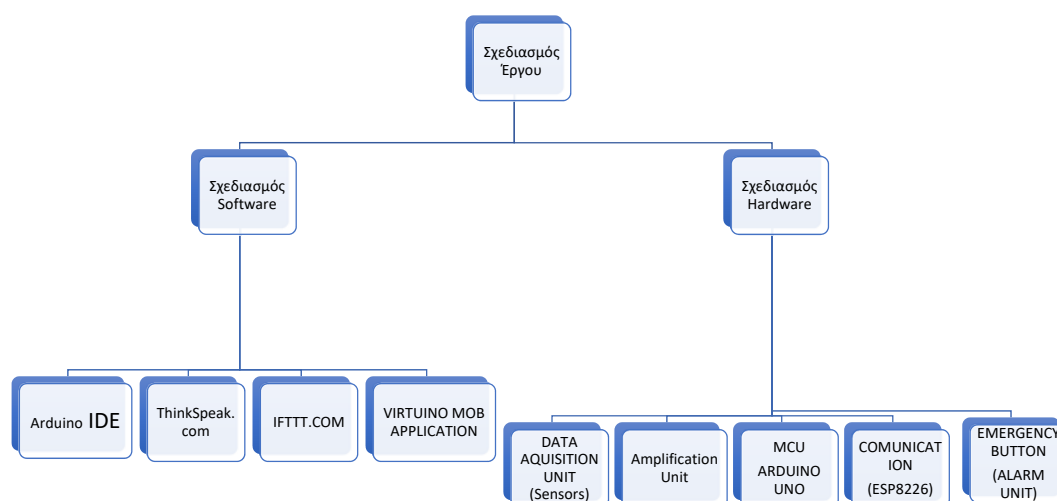
Σκοπός της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός ενός συστήματος IoT για την απομακρυσμένη παρακολούθηση διαφόρων βιοσημάτων σε ασθενείς με χρόνια προβλήματα υγείας. Πρόκειται για ένα ολοκληρωμένο σύστημα παρακολούθησης όπου οι χρήστες του συστήματος (ασθενείς, γιατροί, συγγενείς) μπορούν να έχουν πρόσβαση οποιαδήποτε στιγμή μέσω web ή mobile phone στα δεδομένα που καταγράφονται, στο ιστορικό των καταγραφών, καθώς επίσης να μπορούν να λαμβάνουν ειδοποιήσεις μέσω SMS ή email όταν παρουσιάζονται καταστάσεις που χρήζουν παρακολούθησης. Επιπλέον, ο χρήστης θα μπορεί να τοποθετεί μόνος του τη συσκευή παρακολούθησης για τη λήψη των μετρήσεων.

4.1 Γενική Περιγραφή

Στόχος του συστήματος που υλοποιούμε είναι η συνεχής λήψη των ζωτικών σημάτων του ασθενούς με στόχο την πρόωπη ανίχνευση επικίνδυνων καταστάσεων για την υγεία του ασθενούς και η άμεση επέμβαση για την αποφυγή μεταφοράς του στο νοσοκομείο.

4.2 Λεπτομερής Περιγραφή του Συστήματος

Αν θεωρήσουμε το έργο ως ένα σύστημα που περιέχει πολλά μέρη, το σύστημα που προτείνεται σε αυτή την εργασία μπορεί να χωριστεί σε δύο τμήματα. Το τμήμα Hardware και το τμήμα Software. Για περαιτέρω απλοποίηση κάθε ένα από τα τμήματα ταξινομείται σε κατηγορίες με βάση τις λειτουργίες που επιτελούνται (βλ. Σχήμα 7).



Σχήμα 7: Λεπτομερής περιγραφή του συστήματος.

4.3 Σχεδιασμός του Hardware

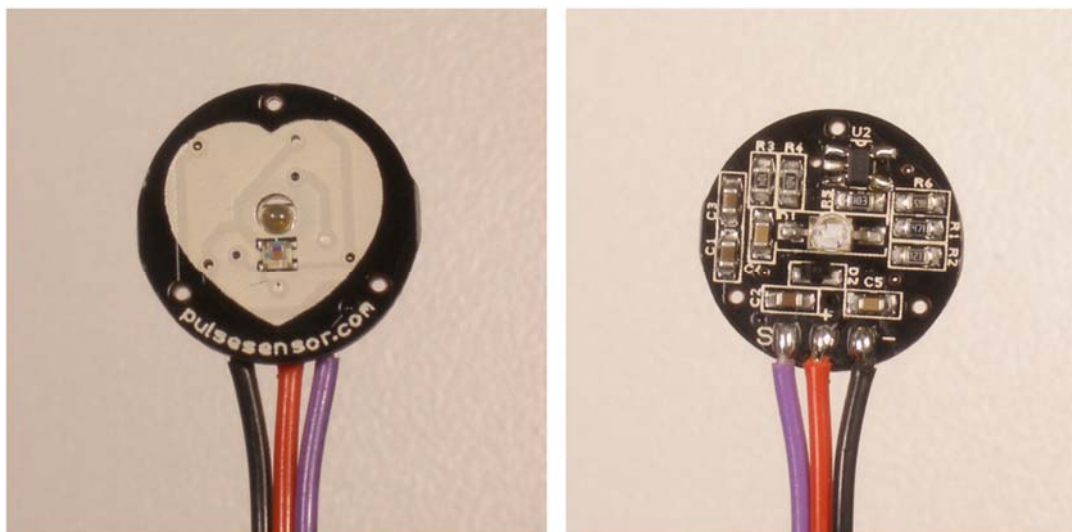
Το hardware του συστήματος που υλοποιούμε είναι ο συνδυασμός πέντε μονάδων: Της μονάδας data acquisition η οποία περιλαμβάνει τους αισθητήρες μέτρησης παλμού καρδιάς, θερμοκρασίας σώματος και ρυθμού αναπνοής, της μονάδας ενίσχυσης (amplification) σήματος του αισθητήρα ρυθμού αναπνοής, την μονάδα MCU (Arduino UNO), την μονάδα επικοινωνίας (Wi-Fi module ESP8226) και την μονάδα για την δημιουργία ειδοποιήσεων συναγερμού (Emergency Button) για περιπτώσεις έκτακτης ανάγκης.

4.3.1 Μονάδα Data Acquisition

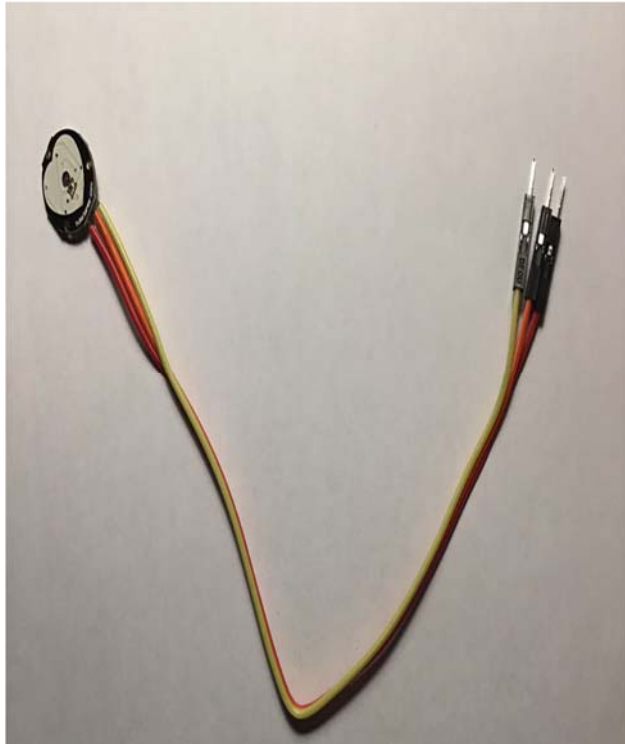
Όπως προαναφέραμε, η μονάδα Data Acquisition περιλαμβάνει τρεις αισθητήρες: Τον αισθητήρα παλμού καρδιάς (Pulse Sensor APDS-9008), τον αισθητήρα για την μέτρηση θερμοκρασίας σώματος (LM-35) και τον αισθητήρα για την μέτρηση του ρυθμού αναπνοής, ο οποίος, για το σύστημα που υλοποιούμε, είναι ένα NTC thermistor.

4.3.1.1 Αισθητήρας παλμού καρδιάς

Ο αισθητήρας παλμού καρδιάς είναι ένας καλοσχεδιασμένος plug and play αισθητήρας που μπορεί να συνδεθεί στην πλακέτα ανάπτυξης Arduino. Μπορεί να τοποθετηθεί για μέτρηση στην άκρη ενός δακτύλου ή στο λοβό του αυτιού. Με τη χρήση jumper wires συνδέεται στο Arduino.



Εικόνα 3: Οι δύο όψεις του αισθητήρα παλμού καρδιάς.



RED CABLE: GND

ORANGE CABLE: 3 to 5V

YELLOW CABLE: Signal

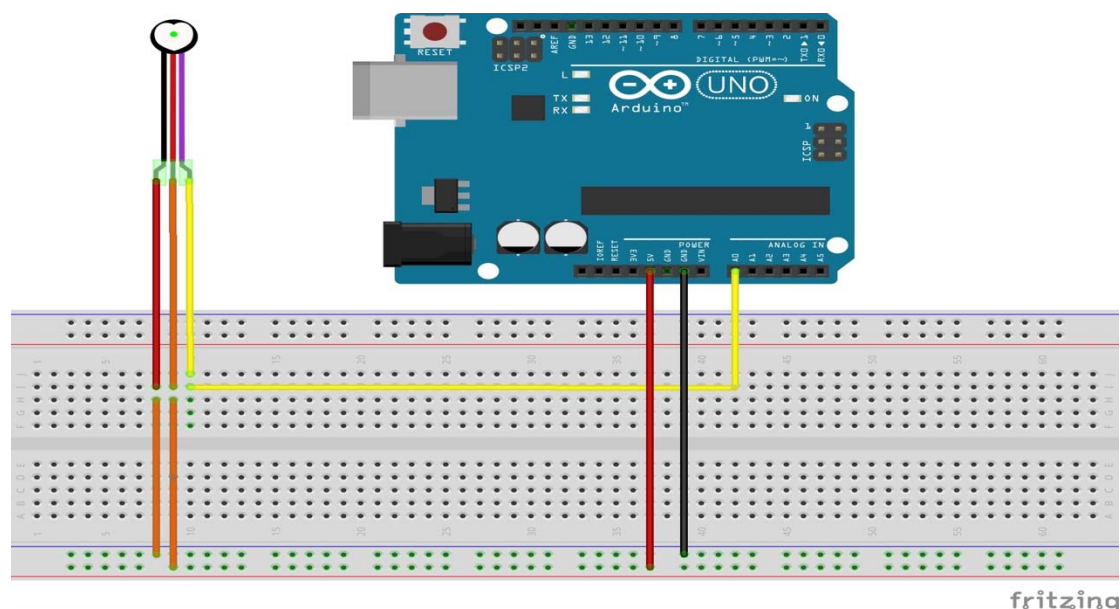
Εικόνα 4: Pinout του αισθητήρα παλμού καρδιάς.

Η μπροστά όψη του αισθητήρα είναι αυτή με το logo της καρδιάς. Αυτή είναι η πλευρά που έρχεται σε επαφή με το δέρμα. Παρατηρώντας αυτή την όψη βλέπουμε μια μικρή κυκλική τρύπα μέσω της οποίας περνάει το φως που παράγεται από το LED, το οποίο βρίσκεται από πίσω, καθώς επίσης και ένα μικρό τετράγωνο ακριβώς κάτω από την τρύπα. Το τετράγωνο αυτό είναι ο αισθητήρας περιβάλλοντος φωτός, ο οποίος είναι ακριβώς όπως οι αισθητήρες που βρίσκονται σε κινητά, tablets, φορητούς υπολογιστές και χρησιμοποιούνται για τη ρύθμιση της φωτεινότητας στις οθόνες σε διαφορετικές συνθήκες φωτισμού. Το LED εκπέμπει φως στην άκρη του δακτύλου ή στο λοβό του αυτιού ή σε οποιοδήποτε άλλο τριχοειδικό ιστό, και ο αισθητήρας διαβάζει το φως που ανακλάται από τον ιστό. Στην πίσω όψη του αισθητήρα είναι τοποθετημένα τα υπόλοιπα εξαρτήματα του κυκλώματος. Ο αισθητήρας παλμού μετατρέπει το φυσικό φωτοπληθυσμογράφημα (**photoplethysmogram (PPG)**) (Wikipedia, 2019) σε ηλεκτρικό σήμα.

Ο αισθητήρας εξάγει ένα ακατέργαστο σήμα τάσης αναλογικής διακύμανσης, το ενισχύει και το ομαλοποιεί. Σε κάθε χτύπο της καρδιάς, ένα κύμα παλμού μεταφέρεται μέσω των αρτηριών στους ιστούς πάνω από τους οποίους είναι τοποθετημένος ο αισθητήρας παλμού. Όταν αυτό το κύμα παλμού περνάει κάτω από τον αισθητήρα, το σήμα παρουσιάζει μια απότομη άνοδο στην τιμή του. Στη συνέχεια το σήμα πέφτει ξανά στην κανονική του τιμή πριν ο επόμενος παλμός περάσει κάτω από τον αισθητήρα (OP Amplifiers, χ.η).

Λόγω του χαρακτηριστικού της επαναληψιμότητας του κύματος παλμού, οι κορυφές της κυματομορφής επιλέγονται ως σημείο αναφοράς καθώς είναι αναγνωρίσιμες. Εφαρμόζοντας αλγόριθμο υπολογισμού στο χρόνο μεταξύ δύο διαδοχικών κορυφών μετριέται ο καρδιακός ρυθμός (Heart Rate). Ιδανικά θέλουμε να βρούμε την ακριβή στιγμή του καρδιακού χτύπου για ακριβείς μετρήσεις.

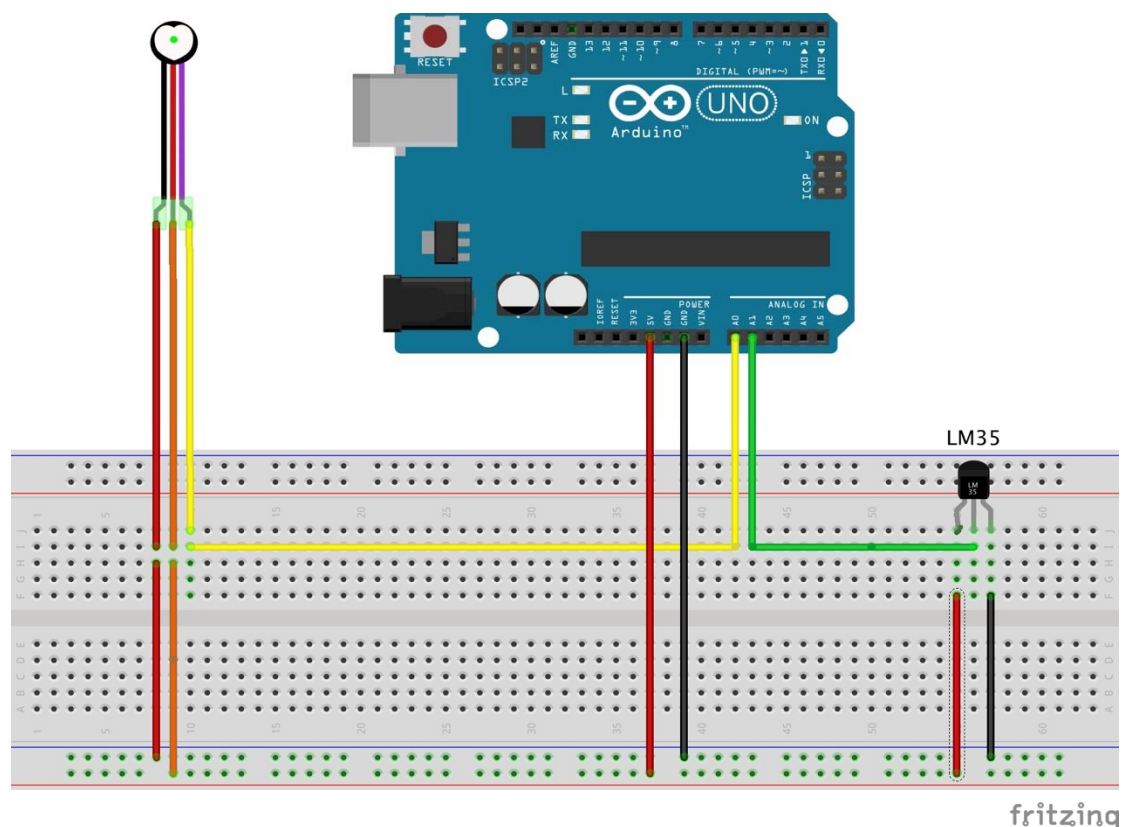
Για την επίτευξη όλων αυτών πρέπει να διασυνδέσουμε τον αισθητήρα παλμού με το Arduino UNO. Στο παρακάτω σχήμα δείχνουμε τον τρόπο σύνδεσης.



Σχήμα 8: Σύνδεση αισθητήρα παλμού καρδιάς με Arduino Uno μέσω breadboard (ράστερ).

της τάσης εισόδου (4V έως 30V), ο μεσαίος είναι η έξοδος του σήματος και ο δεξιός είναι η γείωση (GND) (LM35 Datasheet).

Στο παρακάτω σχήμα φαίνεται ο τρόπος σύνδεσης του LM35 με το Arduino UNO στο κύκλωμα στο οποίο έχει ήδη συνδεθεί ο αισθητήρας παλμού.



Σχήμα 10: Σύνδεση αισθητήρα θερμοκρασίας LM35 με Arduino Uno μέσω breadboard (ράστερ)

4.3.1.3 Thermistor

Στο έργο που υλοποιούμε επιλέξαμε τη χρήση thermistor ως αισθητήρα για τη μέτρηση του αναπνευστικού ρυθμού. Το thermistor είναι ένας τύπος αντίστασης (κατασκευασμένο από ορισμένα οξειδία μετάλλων) του οποίου η ωμική αντίσταση εξαρτάται από τη θερμοκρασία. Υπάρχουν δύο τύποι thermistors:(i) Τα thermistors αρνητικού συντελεστή θερμοκρασίας (NTC), των οποίων η αντίσταση πέφτει όταν η περιβάλλουσα θερμοκρασία ανεβαίνει, (ii) Τα thermistors θετικού συντελεστή θερμοκρασίας (PTC) των οποίων η

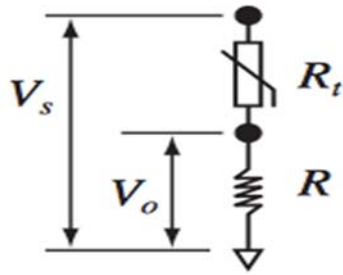
αντίσταση ανεβαίνει με την άνοδο της περιβάλλουσας θερμοκρασίας. Τα thermistors που κυκλοφορούν στο εμπόριο είναι αρκετά μικρά σε μέγεθος (μερικά mm διάμετρος), έχουν όμως υψηλό χρόνο απόκρισης (τυπικά λίγα δευτερόλεπτα) γεγονός που τα καθιστά ικανά για τη σωστή ανάλυση του pattern της αναπνοής και για monitoring σε πραγματικό χρόνο (Massaroni, 2019).

Για το σύστημα που αναπτύξαμε επιλέξαμε τη χρήση ενός thermistor NTC με τιμή 100KΩ στους 25 °C.



Εικόνα 5: Thermistor 100KΩ.

Όπως αναφέραμε παραπάνω, ένα thermistor δείχνει τη θερμοκρασία με τις αλλαγές στην ηλεκτρική του αντίσταση. Οι αναλογικές είσοδοι στο Arduino μπορούν να μετρήσουν μόνο ηλεκτρική τάση και συνεπώς, η ηλεκτρική αντίσταση του thermistor δεν μπορεί να μετρηθεί απευθείας. Μια απλή τεχνική για τη μετατροπή της αντίστασης του thermistor σε τάση είναι η χρήση του thermistor σε ένα διαιρέτη τάσης όπως φαίνεται στο Σχήμα 11:



Σχήμα 11: Διαίρετης τάσης.

Ο διαίρετης τάσης έχει δύο αντιστάσεις σε σειρά. Στο παραπάνω σχήμα η πάνω αντίσταση είναι το thermistor του οποίου η αντίσταση είναι μεταβλητή R_t και κάτω μια αντίσταση σταθερής τιμής R . Στον διαίρετη τάσης εφαρμόζεται μια τάση V_s . Η τάση εξόδου είναι η V_o και εξαρτάται από τις V_s και R των οποίων οι τιμές είναι γνωστές και από την R_t η οποία είναι μεταβλητή και άγνωστη. Η V_o είναι η τάση που θα μετριέται στην αναλογική είσοδο του Arduino. Η τάση V_o στα άκρα του θερμίστορ υπολογίζεται από τον τύπο (Recktenwald, 2013):

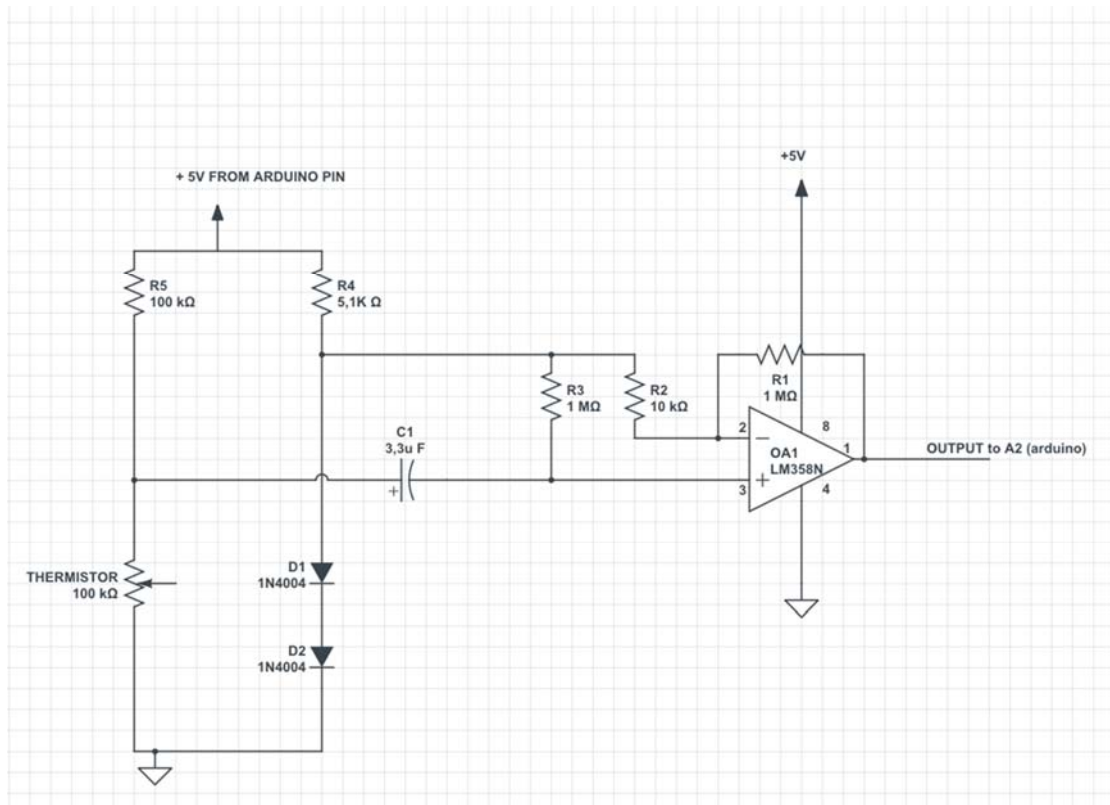
$$V_o = \frac{V_s R_t}{R_t + R}$$

Το thermistor θα το τοποθετήσουμε σε μια ρινική μάσκα οξυγόνου έτσι ώστε μέσω της ρινικής οδού να είναι σε απ' ευθείας επαφή με τον αέρα που εισπνέει και εκπνέει ο ασθενής.



Εικόνα 6: Τοποθέτηση thermistor σε ρινική μάσκα (ρινικά γυαλάκια).

Κατά τη διάρκεια της εισπνοής και της εκπνοής η μεταβολή της τάσης στις άκρες του thermistor είναι πολύ μικρή. Αυτή η μικρή μεταβολή της τάσης δεν είναι ανιχνεύσιμη από τον analog-to-digital μετατροπέα (ADC) του Arduino UNO. Για το λόγο αυτό, η τάση στο thermistor πρέπει να ενισχυθεί με τη χρήση ενός ενισχυτή (Mohan, 2015). Στο σύστημα που υλοποιούμε χρησιμοποιούμε έναν κλασικό τελεστικό ενισχυτή, LM 358 (LM 358 Op Amp, χ.η.). Στο Σχήμα 12 φαίνεται το κύκλωμα που υλοποιούμε για την ενίσχυση του σήματος.



Σχήμα 12: Κύκλωμα ενίσχυσης σήματος στο thermistor.

Ο τελεστικός ενισχυτής χρησιμοποιείται ως συγκριτής και οι αλλαγές της θερμοκρασίας μετριοούνται με το thermistor των 100KΩ. Ο τελεστικός ενισχυτής είναι ένας διαφορικός ενισχυτής, πράγμα που σημαίνει ότι ενισχύει την διαφορά τάσης που υπάρχει ανάμεσα στην αντιστρέψιμη και τη μη-αντιστρέψιμη είσοδο. Το κέρδος ενίσχυσης (gain) προκύπτει από τον λόγο των αντιστάσεων 1M/10K και στην περίπτωση μας είναι 100.

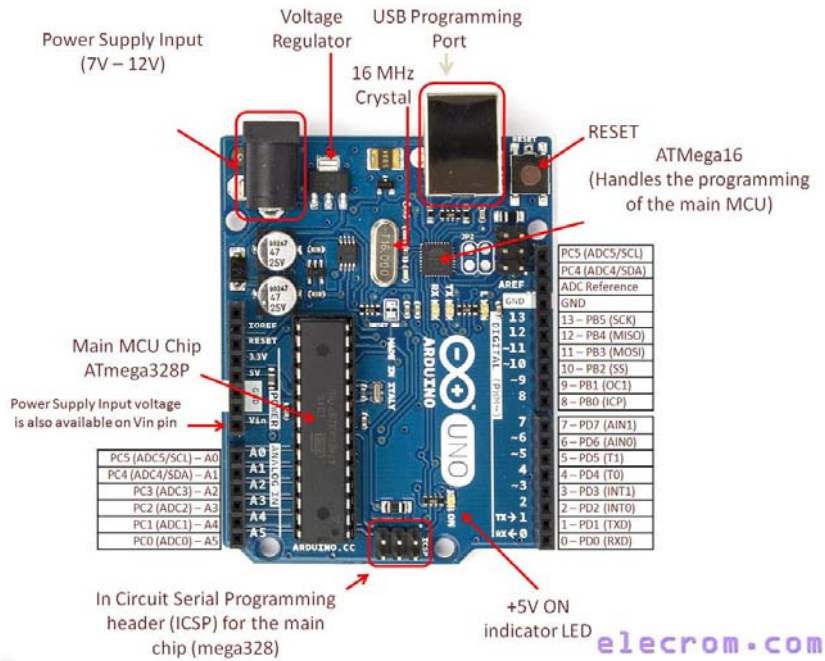
Η έξοδος ενός τελεστικού ενισχυτή είναι πολύ ισχυρή και υπερκαλύπτει οποιαδήποτε τάση πόλωσης (bias) (How To Bias An Op-Amp, χ.η.) που εφαρμόζεται εκεί. Επειδή όμως η έξοδος αντανακλά τις τάσεις που υπάρχουν στην είσοδο, εφαρμόζουμε μια bias τάση 1,4V στην είσοδο (-) με τη χρήση των δύο διόδων 1N4004 και με την αρνητική ανατροφοδότηση φέρνουμε την έξοδο στην τάση που επιθυμούμε. Το ενισχυμένο σήμα από το pin 1 του τελεστικού ενισχυτή οδηγείται στην αναλογική είσοδο A2 του Arduino UNO.

4.3.2 Microcontroller (MCU): Arduino UNO

Ως μονάδα μικροελεγκτή στο σύστημα που αναπτύχθηκε χρησιμοποιούμε την πλακέτα (board) ανάπτυξης Arduino Uno που βασίζεται στον μικροελεγκτή ATmega328P της Atmel.

Το Arduino UNO διαθέτει 14 ψηφιακές εισόδους/εξόδους από τις οποίες 6 υποστηρίζουν PWM (Pulse Width Modulation). Επίσης, διαθέτει 6 αναλογικές εισόδους οι οποίες καταλήγουν σε έναν μετατροπέα A/D των 10 bit. Η μνήμη προγράμματος (flash rom) είναι 32KB, από τα οποία τα 0.5KB δεσμεύονται από τον bootloader, με τον οποίο είναι δυνατή η μεταφορά (upload) του κώδικα (sketch) από τον υπολογιστή στο Arduino με χρήση της θύρας USB και χωρίς επιπλέον Hardware. Το υπόλοιπο της μνήμης flash χρησιμοποιείται για την αποθήκευση των προγραμμάτων που φορτώνονται. Διαθέτει επίσης, μνήμη EEPROM 1KB για ανάγνωση και εγγραφή δεδομένων (τα οποία και διατηρεί μετά από διακοπή τροφοδοσίας) και μνήμη SRAM 2KB για την αποθήκευση δεδομένων (πίνακες, μεταβλητές) κατά τη διάρκεια λειτουργίας, και τα χάνει όταν η παροχή ρεύματος στο Arduino διακοπεί ή αν γίνει reset. Για την επικοινωνία χρησιμοποιείται το πρωτόκολλο STK500. Ο χρονισμός του κυκλώματος γίνεται με ένα ρολόι στα 16MHz. Μπορεί να τροφοδοτηθεί από το καλώδιο διασύνδεσης USB που χρησιμοποιείται για τον προγραμματισμό του ή από εξωτερικό τροφοδοτικό (+6V–20V DC). Εμείς χρησιμοποιούμε την τροφοδοσία που παρέχεται από το καλώδιο USB (Arduino Board, Product Overview, 2016).

Το Arduino διαθέτει 14 θηλυκά pins αριθμημένα από το 0-13 που μπορούν να οριστούν είτε ως ψηφιακές εισοδοί αλλά και έξοδοι (Εικόνα 7). Ως έξοδοι, αλλάζουν κατάσταση μεταξύ λογικού «1» και «0» βάσει του προγράμματος που έχει φορτωθεί, ώστε να στέλνουν το ανάλογο σήμα στα συνδεδεμένα εξαρτήματα. Ως εισοδοί, δίνεται η δυνατότητα στο πρόγραμμα να δεχθεί πληροφορίες από κάποιο εξάρτημα, διαβάζοντας την κατάσταση του pin.



Εικόνα 7: Το Arduino Uno.

Τα 5 power pins (τροφοδοσίας) έχουν τα εξής χαρακτηριστικά:

Vin: Η τάση εισόδου του κυκλώματος

5V: Τάση εξόδου

3V3: Τροφοδοσία 3,3V που παράγεται από τον voltage regulator του board

GND: Τα pin της γείωσης

IOREF: Παρέχει την τάση με την οποία λειτουργεί ο μικροελεγκτής

Τα υπόλοιπα pins:

Pin 0: Rx (pin 2/RxD του μC)

Pin 1: Tx (pin 3/TxD του μC)

Pin 2: General I/O (pin 4/PD2/INT0 του μC)

Pin 3: General I/O ή PWM (pin 5/PD3/INT1 του μC)

Pin 4: General I/O (pin 6/PD4/T0 του μC)

Pin 5: General I/O ή PWM (pin 11/PD5/T1 του μC)

Pin 6: General I/O ή PWM (pin 12/PD6/AIN0 του μC)

Pin 7: General I/O (pin 13/PD7/AIN1 του μC)

- Pin 8: General I/O (pin 14/PB0/ICP του μC)
- Pin 9: General I/O ή PWM (pin 15/PB1/OC1 του μC)
- Pin 10: General I/O ή PWM (pin 16/PB2/SS του μC)
- Pin 11: General I/O ή PWM (pin 17/PB3/MOSI του μC)
- Pin 12: General I/O (pin 18/PB4/MISO του μC)
- Pin 13: General I/O (pin 19/PB5/SCK του μC)

Εκτός από τα παραπάνω, υπάρχει και η σειρά με τα 6 pins αναλογικής εισόδου (ANALOG IN) τα οποία καταλήγουν σε ένα μετατροπέα Αναλογικού – Ψηφιακού Analog to Digital Converter-ADC) και μπορούν να λειτουργούν ως αναλογικοί είσοδοι (Χατζηευθυμίου, 2015). Στο σύστημα που αναπτύσσουμε χρησιμοποιούμε τις τρεις πρώτες αναλογικές εισόδους A0, A1, A2 για την ανάγνωση δεδομένων από τους αισθητήρες ρυθμού παλμού, θερμοκρασίας σώματος και ρυθμού αναπνοής.

4.3.3 Wi-Fi module ESP8266-01

Το ESP8266, παρόλο που είναι ευρέως γνωστό ως Wi Fi module, στην πραγματικότητα είναι ένας μικροελεγκτής, ο οποίος προσφέρει ολοκληρωμένες λύσεις δικτύωσης. Παρέχει συνδεσιμότητα Wi Fi σε οποιονδήποτε μικροελεγκτή μέσω του πλήρους πακέτου TCP/IP που διαθέτει. Η διασύνδεση του ESP8266 με τον μικροελεγκτή (Arduino UNO) μπορεί να επιτευχθεί μέσω UART και με τη βοήθεια μιας μεγάλης γκάμας εντολών AT, ο μικροελεγκτής μπορεί να ελέγχει το ESP module.

Οι εντολές AT του ESP8266 Wi-Fi module είναι υπεύθυνες για τον έλεγχο όλων των λειτουργιών του module όπως restart, σύνδεση στο Wi-Fi, αλλαγή mode λειτουργίας κ.ο.κ. Η κεραία που διαθέτει είναι τυπωμένη επάνω στην πλακέτα του module και παρέχει πολύ καλή κάλυψη (ESP8266 AT Commands, χ.η.). Επίσης, διαθέτει 8 pins (Εικόνα 8) τα οποία είναι τοποθετημένα σε δύο σειρές των 4 pins ανά σειρά.

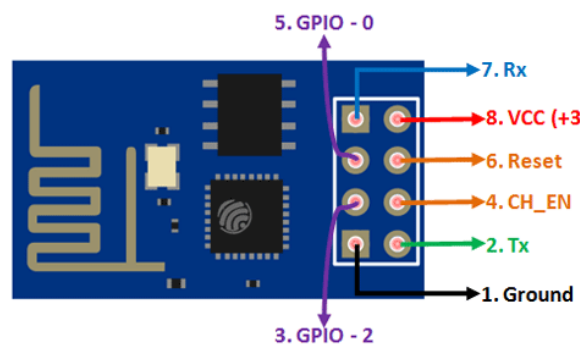
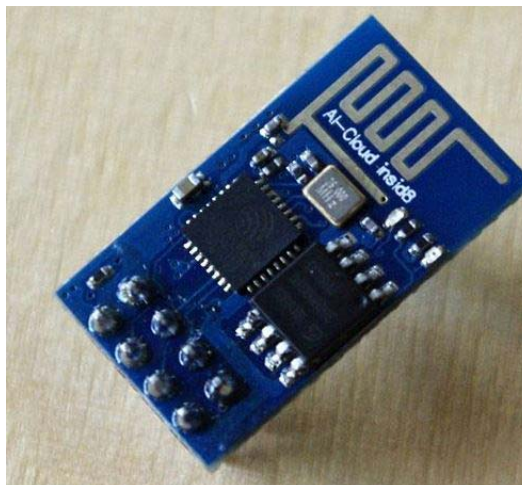
Περιγραφή pins:

Πρώτη Σειρά:

1. **GND** (pin γείωσης)
2. **GPIO2** (Ψηφιακή προγραμματιζόμενη I/O)
3. **GPIO0** (Ψηφιακή προγραμματιζόμενη I/O, χρησιμοποιείται επίσης σε BOOT modes)
4. **RX** – UART Κανάλι Λήψης

Δεύτερη Σειρά:

1. **TX** – UART Κανάλι Μετάδοσης
2. **CH_PD** (Ενεργοποίηση/Απενεργοποίηση τροφοδοσίας, πρέπει να συνδεθεί είτε απευθείας είτε μέσω αντίστασης στα 3.3V)
3. **REST** – reset, πρέπει να συνδεθεί στα 3.3V
4. **VCC -3.3V** Τροφοδοσία

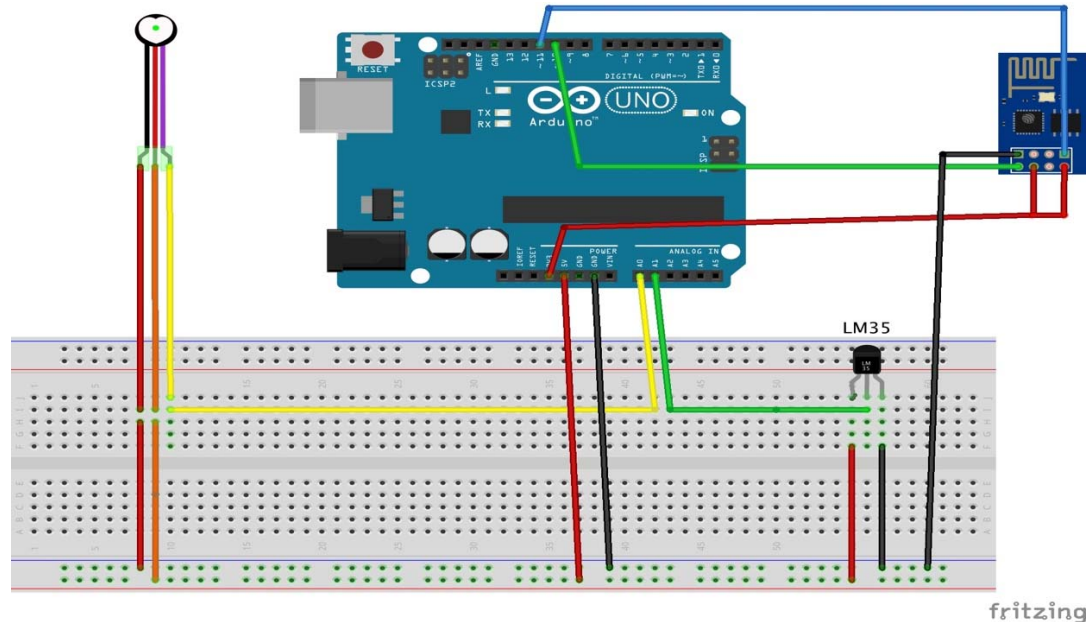


Εικόνα 8: ESP8266-01 Module.

Στο σύστημα που αναπτύξαμε το ESP8266 module λειτουργεί σε mode station για να μπορεί το σύστημα να συνδεθεί στο ασύρματο δίκτυο του σπιτιού και

στη συνέχεια τα δεδομένα να αποστέλλονται στην IoT πλατφόρμα που έχουμε επιλέξει.

Η παρακάτω εικόνα απεικονίζει τη συνδεσμολογία του ESP module με το Arduino UNO.



Σχήμα 13: Σύνδεση Wi-Fi module με Arduino Uno μέσω breadboard (ράστερ).

UNO	ESP
11	RX
10	TX
3.3V	VCC
GND	GND
3.3V	CH_EN

Πίνακας 2: Αντιστοιχία pins Arduino UNO με ESP8266

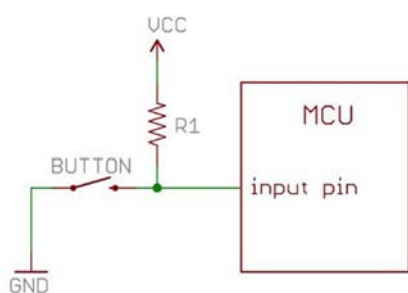
4.3.4 Push Button

Όπως προαναφέραμε, το σύστημα που αναπτύσσουμε θα δίνει τη δυνατότητα στον χρήστη-ασθενή να στέλνει ειδοποιήσεις συναγερμού στους οικείους του ή στον γιατρό του με το πάτημα ενός κουμπιού, του emergency button. Σχεδιαστικά, για να το επιτύχουμε, θα χρησιμοποιήσουμε ένα pushbutton και μια pull-up αντίσταση (εδώ 10KΩ).

Το pushbutton είναι ένα εξάρτημα το οποίο όταν πατηθεί συνδέει δύο σημεία σε ένα κύκλωμα. Η σύνδεση με το Arduino γίνεται με τρία καλώδια. Το ένα συνδέει τον ένα ακροδέκτη του pushbutton με την τροφοδοσία των 5V μέσω μιας pull-up αντίστασης (10KΩ). Το δεύτερο συνδέει τον άλλο ακροδέκτη με την γείωση. Το τρίτο συνδέει τον ακροδέκτη που πάει στην pull-up αντίσταση με μια από τις ψηφιακές εισόδους/εξόδους του Arduino.

Όταν το pushbutton είναι open (δεν είναι πατημένο), δεν υπάρχει σύνδεση μεταξύ των δύο ακροδεκτών του, και έτσι το pin του pushbutton είναι συνδεδεμένο στα 5V (μέσω της pull-up αντίστασης) και η κατάσταση του pin που διαβάζουμε είναι HIGH. Όταν το button είναι κλειστό (πατημένο), υπάρχει σύνδεση μεταξύ των δύο ακροδεκτών συνδέοντας έτσι το pin στην γείωση και με αυτό τον τρόπο διαβάζουμε LOW. (Το pin είναι ακόμα συνδεδεμένο στα 5V αλλά η αντίσταση μεταξύ τους σημαίνει ότι το pin “είναι πιο κοντά” στη γείωση.

Οι pull-up αντιστάσεις χρησιμοποιούνται συχνά με buttons και διακόπτες.

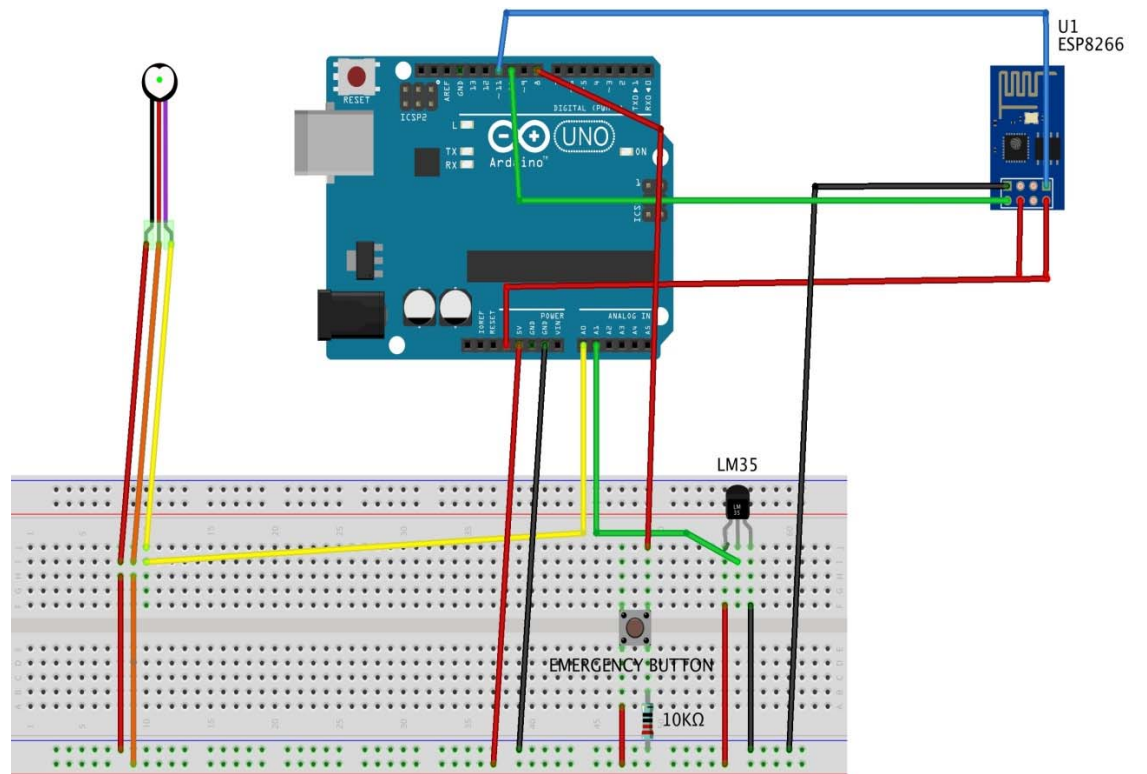


Σχήμα 14: Χρήση pull-up αντίστασης.

Με την pull-up αντίσταση, το input pin θα διαβάζει κατάσταση HIGH όταν το button δεν είναι πατημένο. Με άλλα λόγια, μια μικρή ποσότητα ρεύματος ρέει μεταξύ της VCC και του Input pin (όχι της γείωσης), και γι' αυτό το input pin διαβάζει την VCC, δηλαδή HIGH. Όταν πατηθεί το button, το input pin συνδέεται απευθείας στην γείωση και για το λόγο αυτό στο input pin διαβάζουμε κατάσταση Low. Σημειώνουμε ότι αν η αντίσταση δεν ήταν εκεί με

το πάτημα του button, η VCC θα ενώνονταν απευθείας με τη γείωση το οποίο σημαίνει βραχυκύκλωμα (Pull-up Resistors, 2019).

Σχεδιάζοντας και το κομμάτι του emergency button καταλήγουμε στην τελική μορφή του κυκλώματος (Σχήμα 15).

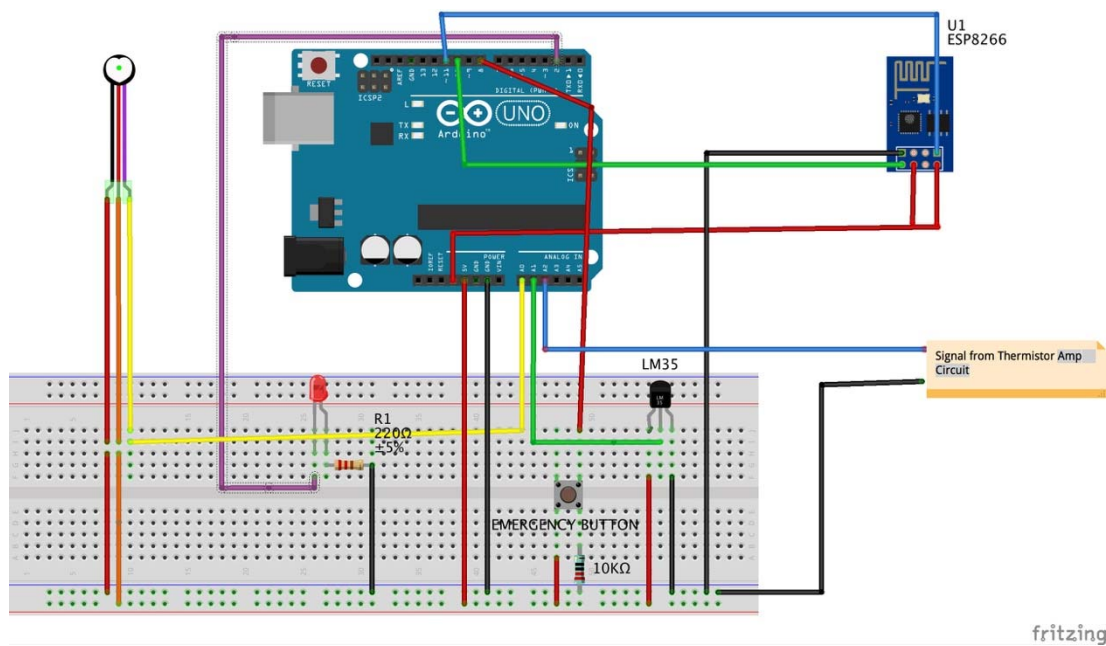


fritzing

Σχήμα 15: Σύνδεση Emergency Button και pull-up αντίστασης με το υπόλοιπο κύκλωμα.

4.3.5 LED Καρδιακού ρυθμού

Στο κύκλωμα που υλοποιούμε έχει προστεθεί και ένα LED το οποίο θα αναβοσβήνει με το ρυθμό παλμού έτσι ο ώστε ο ασθενής να έχει μια ένδειξη για την σωστή λειτουργία του κυκλώματος. Φυσικά, δεν αποτελεί ένδειξη για τη συνολική λειτουργία του συστήματος καθώς αυτό περιλαμβάνει και άλλα τμήματα. Όσον αφορά τη λειτουργία του LED, τα πράγματα είναι απλά. Το led συνδέεται μέσω μιας αντίστασης περιορισμού ρεύματος στην ψηφιακή I/O 2 του Arduino. Στο λογισμικό θέτουμε το pin 2 ως έξοδο και οδηγούμε σε αυτή τον καρδιακό παλμό που λαμβάνουμε από τον αισθητήρα καρδιακού ρυθμού.



Σχήμα 16: Σύνδεση LED και αντίστασης περιορισμού ρεύματος με το υπόλοιπο κύκλωμα.

4.4 Λογισμικό

Κατά τη διάρκεια ανάπτυξης και χρήσης του συστήματος αυτού χρησιμοποιήθηκαν αρκετά εργαλεία λογισμικού για τον προγραμματισμό του Arduino UNO, το οποίο θεωρείται η καρδιά του συστήματος, καθώς επίσης και για την απεικόνιση των δεδομένων, την αποστολή ειδοποιήσεων συναγερμού και για την παρακολούθηση των δεδομένων μέσω mobile συσκευών.

4.4.1 ARDUINO IDE

Το ακρώνυμο IDE προέρχεται από το “Integrated Development Environment”, δηλαδή Ολοκληρωμένο Περιβάλλον Ανάπτυξης. Είναι το επίσημο λογισμικό που εισήχθη από την Arduino.cc και χρησιμοποιείται κυρίως για επεξεργασία, μεταγλώττιση και ανέβασμα του κώδικα σε μια συσκευή Arduino. Σχεδόν όλα τα Arduino modules είναι συμβατά με αυτό το λογισμικό το οποίο είναι ανοιχτού κώδικα και είναι άμεσα διαθέσιμο για εγκατάσταση και εκκίνηση σύνταξης κώδικα.

4.4.1.1 Ορισμός του Arduino IDE

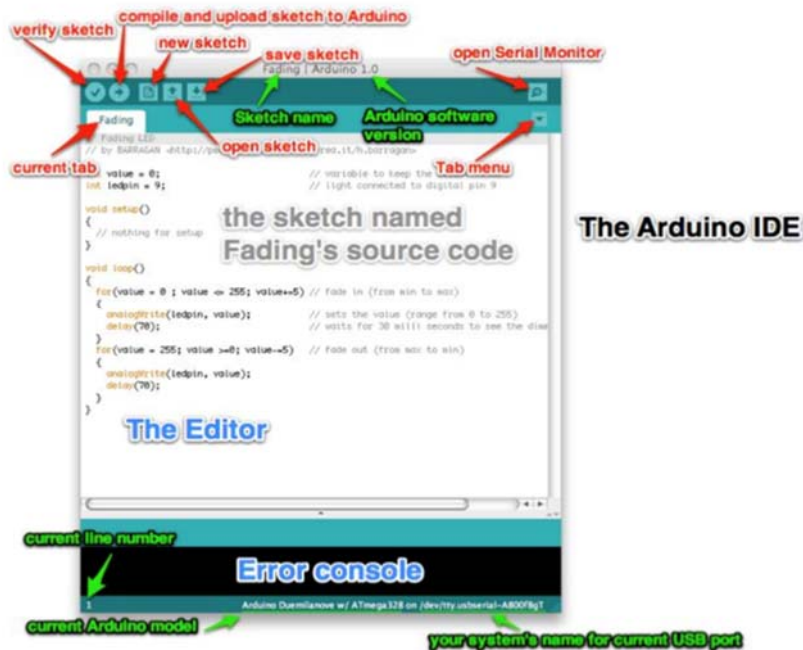
1. Το Arduino IDE είναι το επίσημο λογισμικό για Arduino συσκευές, το οποίο κάνει την σύνταξη κώδικα εύκολη ακόμα και για χρήστες οι οποίοι δεν είναι εξοικειωμένοι με τον προγραμματισμό και την σύνταξη κώδικα.
2. Είναι διαθέσιμο για λειτουργικά συστήματα όπως MAC, Windows, Linux και τρέχει σε μια Java πλατφόρμα η οποία προσφέρεται με ενσωματωμένες συναρτήσεις και εντολές, οι οποίες έχουν ζωτικό ρόλο στην αποσφαλμάτωση, επεξεργασία και μεταγλώττισης του κώδικα σε αυτό το περιβάλλον.
3. Διατίθεται μια μεγάλη γκάμα Arduino modules με πιο χαρακτηριστικά τα: Arduino UNO το οποίο είναι και το module που χρησιμοποιούμε στο σύστημα που αναπτύξαμε, το Arduino Mega, Arduino Leonardo, Arduino nano, Arduino Micro και πολλά άλλα.
4. Κάθε ένα από αυτά συμπεριλαμβάνει ένα μικροελεγκτή ο οποίος ουσιαστικά είναι αυτός που προγραμματίζεται και δέχεται πληροφορίες με τη μορφή κώδικα.
5. Ο κύριος κώδικας, επίσης γνωστός και ως sketch, που δημιουργείται στην πλατφόρμα IDE, θα δημιουργήσει τελικά ένα HEX αρχείο το οποίο στη συνέχεια μεταφέρεται και φορτώνεται στον ελεγκτή της πλακέτας.
6. Το περιβάλλον IDE περιλαμβάνει κυρίως δύο βασικά μέρη: Επεξεργαστή και μεταγλωττιστή, ο οποίος χρησιμοποιείται πρωτίστως για την σύνταξη του απαιτούμενου κώδικα και στη συνέχεια για τη μεταγλώττιση και τη μεταφόρτωση του κώδικα σε μια συσκευή Arduino.
7. Το περιβάλλον υποστηρίζει εξίσου τις γλώσσες προγραμματισμού C και C++.

Το περιβάλλον IDE χωρίζεται σε τρεις ενότητες:

1. Menu Bar
2. Text Editor

3. Output Pane

Το περιβάλλον του λογισμικού φαίνεται στην εικόνα 9 (Fezari,2018).

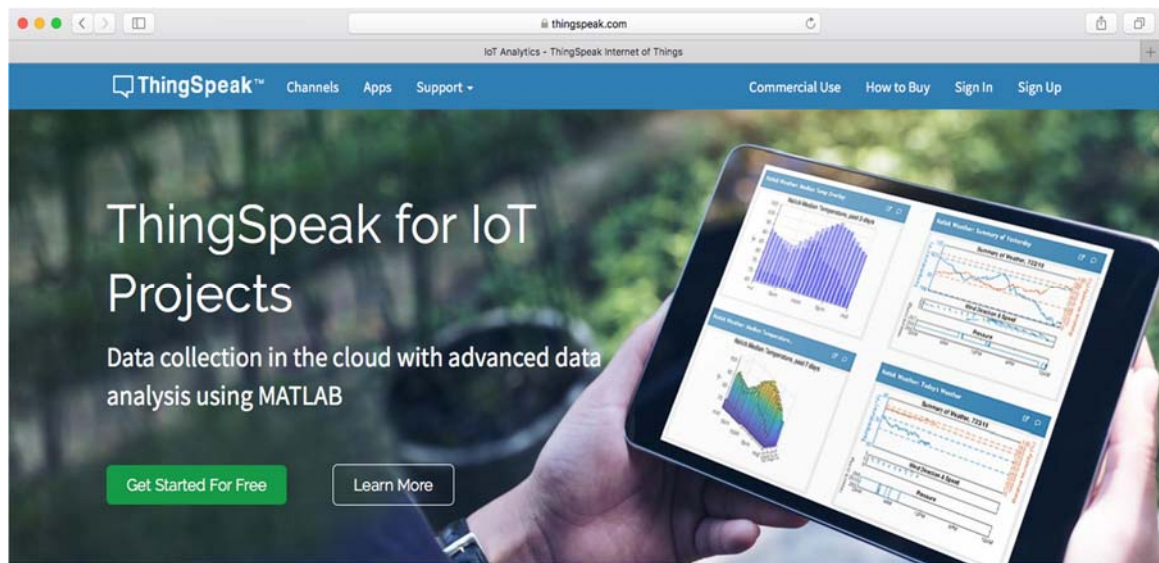


Εικόνα 9: Περιβάλλον Arduino IDE.

4.4.2 Η IoT πλατφόρμα ThingSpeak

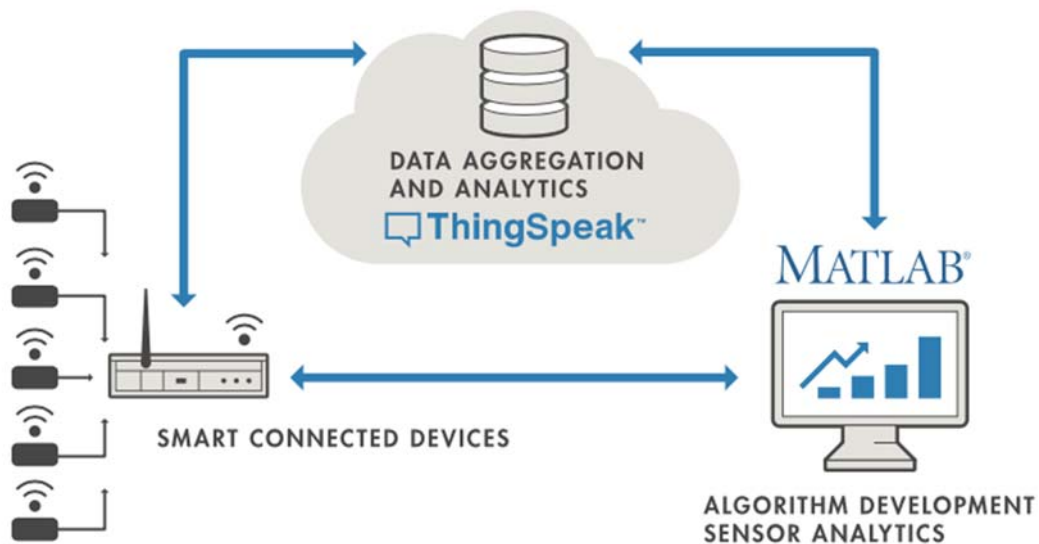
Το ThingSpeak είναι μια δωρεάν διαδικτυακή πλατφόρμα (Εικόνα 10), που επιτρέπει την συλλογή και την αποθήκευση δεδομένων στο σύννεφο (cloud) αλλά και την ανάπτυξη IoT εφαρμογών. Είναι ανοιχτού κώδικα (open source) και εισήχθη για πρώτη φορά το 2010 από την IoBridge ως υποστηρικτική υπηρεσία για IoT εφαρμογές. Παρέχει άμεση απεικόνιση δεδομένων που λαμβάνει από συσκευές σε γραφήματα μέσω MATLAB, ώστε μετέπειτα να υπάρχει η δυνατότητα για online ανάλυση και επεξεργασία. Οι συσκευές επικοινωνούν με το ThingSpeak μέσω ενός RESTful API, το οποίο χρησιμοποιεί το πρωτόκολλο μεταφοράς υπερκειμένου (HTTP) για αιτήσεις τύπου GET, PUT, POST, DELETE για την επεξεργασία των δεδομένων (What is RESTful API?, 2019). Η ιδιαιτερότητα της συγκεκριμένης κατηγορίας API είναι το χαμηλό

bandwidth που απαιτούν, η οποία τα καθιστά ιδανικά για χρήση σε ένα δίκτυο, χωρίς να προκαλούν συμφόρηση σε αυτό.



Εικόνα 10: Η πλατφόρμα ThingSpeak.com.

Σε υψηλό επίπεδο, πολλά συστήματα IoT μπορούν να περιγραφούν χρησιμοποιώντας το διάγραμμα του Σχήματος 17:



Σχήμα 17: Διάγραμμα IoT συστήματος.

Στην αριστερή πλευρά του διαγράμματος έχουμε τις «έξυπνες» συσκευές (τα “things” σε ένα IoT) τα οποία συλλέγουν τα δεδομένα και για το σύστημα που

σχεδιάζουμε αντιπροσωπεύουν τους αισθητήρες ρυθμού παλμού, θερμοκρασίας και thermistor για την μέτρηση του αναπνευστικού ρυθμού.

Στη μέση, έχουμε το cloud που συγκεντρώνονται και αναλύονται σε πραγματικό χρόνο τα δεδομένα από όλες τις πηγές. Στην περίπτωση μας το cloud είναι η IoT πλατφόρμα ThingSpeak. Στη δεξιά πλευρά του διαγράμματος απεικονίζεται η ανάπτυξη αλγορίθμου που σχετίζεται με την εφαρμογή IoT.

Ένα σύστημα IoT περιλαμβάνει όλα αυτά τα στοιχεία. Το ThingSpeak ταιριάζει, όπως αναφέραμε παραπάνω, στο τμήμα cloud του διαγράμματος και παρέχει μια πλατφόρμα για τη γρήγορη συλλογή και ανάλυση δεδομένων από αισθητήρες συνδεδεμένους στο διαδίκτυο (Thingspeak.com, 2019).

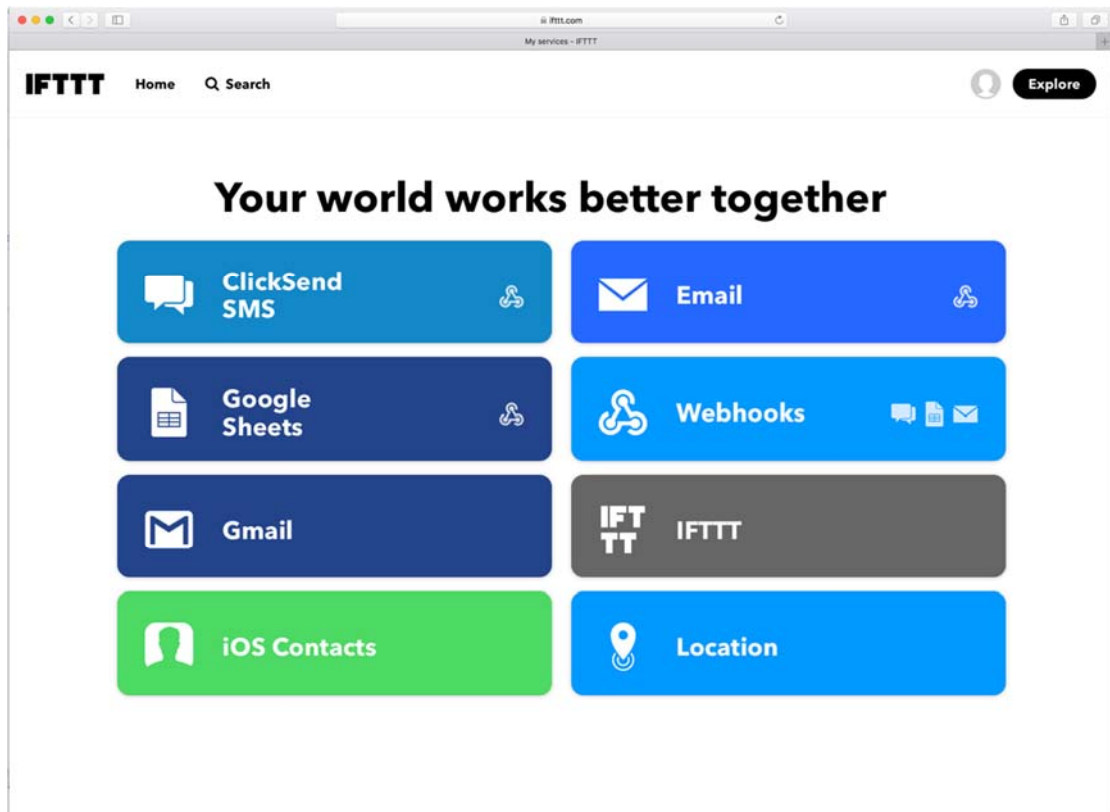
Το ThingSpeak έχει πολλές δυνατότητες από τις οποίες ορισμένες περιλαμβάνουν τα ακόλουθα:

- Εύκολη παραμετροποίηση: Παραμετροποίηση συσκευών για την αποστολή δεδομένων στο ThingSpeak με τη χρήση πρωτοκόλλων IoT.
- Απεικόνιση: Απεικόνιση δεδομένων αισθητήρων που συλλέγονται σε πραγματικό χρόνο.
- Συσσωμάτωση: συγκεντρωτικά δεδομένα κατόπιν αιτήματος από πηγές τρίτων.
- Ανάλυση: Εκτέλεση της αυτόματης IoT ανάλυσης η οποία βασίζεται σε events ή προγραμματισμένες ενέργειες.
- Πρότυπα: Κατασκευή και προτυποποίηση συστημάτων IoT χωρίς να χρειάζεται εγκατάσταση servers ή ανάπτυξη web λογισμικού.
- Αυτοματισμός: Αυτοματοποιημένος χειρισμός των δεδομένων και επικοινωνία με χρήση υπηρεσιών τρίτων όπως του Twitter ή του IFTTT (Mohamad et al., 2019).

4.4.3 Η πλατφόρμα IFTTT

Το IFTTT (ακρώνυμο του **If This Then That**) είναι μια δωρεάν web-based υπηρεσία για την δημιουργία αλυσίδων conditional statements, γνωστών ως applets (Εικόνα 11).

Ένα applet ενεργοποιείται από αλλαγές που συμβαίνουν μέσα σε μια άλλη web υπηρεσία όπως είναι οι Gmail, Facebook, Instagram, Pinterest κ.α. Στο σύστημα που αναπτύξαμε τα applets που δημιουργήσαμε και θα μελετήσουμε αναλυτικά στο κεφάλαιο 4 ενεργοποιούνται εντός της πλατφόρμας ThingSpeak.com. Αυτό γίνεται με την χρήση των εφαρμογών ThingHTTP που διαθέτει η πλατφόρμα ThingSpeak. Οι εφαρμογές ThingHTTP ενεργοποιούν τα IFTTT applets που έχουμε δημιουργήσει για την αποστολή SMS σε έναν ή περισσότερους αριθμούς κινητού τηλεφώνου που έχουμε ορίσει όταν ο χρήστης-ασθενής πατάει το κουμπί (emergency button) ενεργοποίησης ειδοποιήσεων. Με τον ίδιο τρόπο ενεργοποιείται το IFTTT applet για αποστολή μηνύματος email πάλι σε προκαθορισμένους παραλήπτες, όταν οι τιμές των αισθητήρων που εισέρχονται στην πλατφόρμα ξεπερνούν ή υπολείπονται των ορίων συναγερμού που έχουμε θέσει (Wiki, 2019).



Εικόνα 11: Η πλατφόρμα IFTTT.com.

Συνοψίζοντας θα λέγαμε ότι το IFTTT είναι ένα εργαλείο λογισμικού με καθόλου ή ελάχιστο κώδικα για την δημιουργία μικρών προγραμμάτων (applets) αποτελούμενων από ενεργοποιητές (triggers) (if this) και λειτουργίες (actions)(then that). Κάποιες από τις έννοιες που χρησιμοποιούνται είναι:

- Οι υπηρεσίες είναι το βασικό δομικό συστατικό του IFTTT. Περιγράφουν κυρίως μια σειρά δεδομένων από μια συγκεκριμένη υπηρεσία ιστού, όπως Gmail, YouTube κ.α. Οι υπηρεσίες μπορούν επίσης, να περιγράψουν ενέργειες που ελέγχονται με συγκεκριμένα APIs, όπως τα SMS. Κάθε υπηρεσία έχει ένα συγκεκριμένο σύνολο ενεργοποιήσεων και λειτουργιών.
- Ενεργοποιητές/Πυροδοτητές (triggers) είναι το μέρος “this” ενός applet. Είναι ουσιαστικά αυτά τα στοιχεία που ενεργοποιούν μια λειτουργία (action).
- Λειτουργίες είναι το μέρος “that” του applet. Είναι η έξοδος που προκύπτει από την είσοδο ενός trigger.

- Applets (μικρές εφαρμογές) είναι μικρά προγράμματα λογισμικού τα οποία υποστηρίζουν μεγαλύτερα προγράμματα εφαρμογών. Στο παρελθόν, ο όρος applet ήταν συχνά συνδεδεμένος με την γλώσσα προγραμματισμού Java. Σήμερα, ο όρος συχνά συνδέεται με την υπηρεσία IFTTT. Το IFTTT χρησιμοποιεί κοινή λογική προγραμματισμού ώστε να επιτρέψει σε συγκεκριμένα γεγονότα να ενεργοποιηθούν από ένα λογισμικό ως μια υπηρεσία η οποία θα προκαλεί αντίδραση σε μια άλλη υπηρεσία cloud. Αυτό επιτρέπει σε έναν τελικό χρήστη χωρίς τεχνική κατάρτιση να αυτοματοποιήσει καθημερινές εργασίες προγραμματίζοντας τα triggers και τα actions (What is applet, 2017).

4.4.4 ThingHTTP App

Οι εφαρμογές ThingHTTP ενεργοποιούν την επικοινωνία μεταξύ συσκευών, ιστοσελίδων, υπηρεσιών διαδικτύου χωρίς να χρειάζεται η εφαρμογή πρωτοκόλλου στο επίπεδο της συσκευής. Καθορίζουμε ενέργειες (actions) στο ThingHTTP, τις οποίες πυροδοτούμε χρησιμοποιώντας άλλες ThingSpeak εφαρμογές όπως οι TweetControl, TimeControl και React. Στην συγκεκριμένη εργασία χρησιμοποιούμε τις εφαρμογές React για την πυροδότηση των action μιας ThingHTTP εφαρμογής (Mathworks.com, 2014). Η ThingHTTP υποστηρίζει μεθόδους GET, POST, PUT και DELETE, SSL καθώς επίσης και βασική αυθεντικοποίηση, προσαρμοσμένο hostname, προσαρμοσμένες κεφαλίδες, διαφορετικούς τύπους περιεχομένου. Επιπρόσθετα, υποστηρίζει και τις εκδόσεις 1.0 και 1.1 του HTTP πρωτοκόλλου.

4.4.5 React App

Οι εφαρμογές React επιτρέπουν την πυροδότηση ThingHTTP αιτημάτων ή στέλνουν ένα tweet με τη χρήση του ThingTweet όταν το κανάλι του ThingSpeak πληροί συγκεκριμένες προϋποθέσεις (ThingSpeak IoT Community, 2012). Για παράδειγμα, μπορούμε να θέσουμε ένα θερμοστάτη σε λειτουργία τη στιγμή που επιστρέφουμε στο σπίτι από τη δουλειά δημιουργώντας ένα react εντοπισμού θέσης (Geo Location React). Μπορούμε να δημιουργήσουμε πολλές

διαφορετικές αντιδράσεις στα δεδομένα αισθητήρων με τη χρήση των ThingSpeak εφαρμογών React.

4.4.6 Virtuino

Η εφαρμογή Virtuino είναι μια πλατφόρμα HMI για εξυπηρετητές IoT, Arduino ESP και άλλα παρόμοια boards, MQTT brokers, PLC και διακομιστές Modbus.

Μπορούμε να δημιουργήσουμε εικονικές οθόνες στο τηλέφωνο ή στο tablet για να ελέγχουμε το σύστημα που αναπτύξαμε, μέσω Bluetooth, WiFi ή Web (Virtuino App, χ.η.). Η εφαρμογή είναι διαθέσιμη δωρεάν στο Play Store αλλά και σαν .APK στο Virtuino Server μέσω της ιστοσελίδας virtuino.com.



Εικόνα 12: Η Android εφαρμογή Virtuino.

Κεφάλαιο 5

Υλοποίηση του συστήματος

5.1 Ερμηνεία του Arduino κώδικα

Ο Arduino κώδικας που χρησιμοποιήθηκε παρατίθεται ολοκληρωμένος στο παράρτημα του 7^{ου} κεφαλαίου. Σε αυτή την ενότητα θα επεξηγήσουμε τα σημαντικότερα μέρη του.

Αρχικά εισάγουμε τρεις βιβλιοθήκες οι οποίες είναι απαραίτητες για τη λειτουργία του κώδικα και κατά συνέπεια του συστήματος (Εικόνα 13).

```
9 #include <SoftwareSerial.h>           // Βιβλιοθήκες απαραίτητες για τη λειτουργία
10 #include "Timer.h"
11 #include <PulseSensorPlayground.h>  |
```

Εικόνα 13. Βιβλιοθήκες

Το Arduino έχει ενσωματωμένη υποστήριξη σειριακής επικοινωνίας στα pins 0, 1. Η σειριακή αυτή επικοινωνία γίνεται με την χρήση hardware, το οποίο ενσωματώνεται σε ένα chip που ονομάζεται UART. Το hardware αυτό επιτρέπει στον μικροελεγκτή ATmega να λαμβάνει σειριακή επικοινωνία ακόμα και όταν εκτελεί άλλες εργασίες όσο υπάρχει χώρος στα 64 bytes του serial buffer. Η βιβλιοθήκη SoftwareSerial.h αναπτύχθηκε με σκοπό να επιτρέπεται η σειριακή επικοινωνία και σε άλλα pins του Arduino, με τη χρήση software το οποίο εξομοιώνει τη λειτουργία αυτή (εξ ου και το όνομα “SoftwareSerial”). Με την χρήση αυτής της βιβλιοθήκης είναι δυνατόν να έχουμε πολλαπλές Software

σειριακές θύρες με ταχύτητες έως και 115200 bps (Arduino.cc, 2019). Στο σύστημα που αναπτύξαμε χρησιμοποιούμε Software σειριακή θύρα για την επικοινωνία του Arduino με το ESP8266 Wi-Fi module.

Για την λειτουργία του αισθητήρα παλμού που χρησιμοποιούμε εισάγουμε την βιβλιοθήκη PulseSensorPlayground. Η βιβλιοθήκη αυτή εμπεριέχει όλο τον απαραίτητο κώδικα που θα χρειαστούμε για να εξάγουμε το HR σήμα από τον αισθητήρα. Η βιβλιοθήκη Timer.h περιέχει τον κατάλληλο κώδικα που μας επιτρέπει να διαχειριστούμε οτιδήποτε έχει να κάνει με χρόνους και διακοπές (intervals) στον κώδικα μας.

Χρησιμοποιούμε τις παρακάτω εντολές (Εικόνα 14) για να δημιουργήσουμε αντικείμενα (όψεις) των κλάσεων που εισαγάγαμε στον κώδικα.

```
15 Timer timer;  
16 PulseSensorPlayground pulseRate;  
17 SoftwareSerial esp8266(10,11);
```

Εικόνα 14. Όψεις των κλάσεων Timer, PulseSensorPlayground και SoftwareSerial.

Χρησιμοποιώντας τα εργαλεία που μας προσφέρει η βιβλιοθήκη PulseSensorPlayground θέτουμε την σταθερά USE_ARDUINO_INTERRUPTS true για πιο ακριβή αντιστοίχιση του BPM που λαμβάνεται από τον αισθητήρα. Επίσης, ενεργοποιούμε το DEBUG έτσι ώστε να εμφανίζονται οι τρέχουσες εντολές στο Serial Monitor του Arduino IDE (Εικόνα 15).

```
3 #define USE_ARDUINO_INTERRUPTS true  
4 #define DEBUG true
```

Εικόνα 15. Δήλωση σταθερών 1.

Επίσης καθορίζουμε ως σταθερές το όνομα του ασύρματου δικτύου (SSID) στο οποίο θα συνδέσουμε τη συσκευή μας, το password του δικτύου (WPA Key) και την IP διεύθυνση της IoT πλατφόρμας ThingSpeak (Εικόνα 16).

```
#define SSID "SVPA" // "SSID- WiFi name"
#define PASS "6942431740" // "password-WPA Key"
#define IP "184.106.153.149" // thingspeak.com ip address
```

Εικόνα 16. Δήλωση σταθερών 2.

Για να ενημερώνεται το κανάλι στην πλατφόρμα ThingSpeak.com με τα δεδομένα που προέρχονται από τους αισθητήρες, δηλώνουμε μια μεταβλητή (msg) τύπου string στην οποία χρησιμοποιούμε τη μέθοδο GET και το api key του Thingspeak καναλιού (Εικόνα 17). Παρακάτω θα δείξουμε τον τρόπο με τον οποίο λαμβάνουμε το API key του καναλιού.

```
String msg = "GET /update?key=ZYRYVE09K5G61864";
```

Εικόνα 17. API Key

Στην συνάρτηση **setup()**, θέτουμε το baud rate για την επικοινωνία του Arduino Serial Monitor με το esp8266. Για την εφαρμογή μας η ταχύτητα είναι 115200 bps. Η επικοινωνία ξεκινάει δίνοντας την εντολή "AT" στο ESP: **esp8266.println("AT")**. Αν η απάντηση που λαμβάνει το Serial Monitor του Arduino είναι "OK", το αποτέλεσμα, δηλαδή, της γραμμής κώδικα **if (esp8266.find("OK"))** είναι True, τότε καλείται η συνάρτηση **connectWiFi()** για να γίνει η σύνδεση του esp με το Wi-Fi δίκτυο. Μετά από όλα αυτά αρχικοποιούμε τους χρονιστές (timers) με τη χρήση της μεθόδου **every** για το αντικείμενο timer της κλάσης **Timer (timer.every(time_interval,do_this))**. Όπου **time_interval** βάζουμε την τιμή 16000 καθώς το ThingSpeak χρειάζεται 15 sec μεταξύ των ενημερώσεων που λαμβάνει. Η τιμή δίνεται σε χιλιάδες γιατί ο κώδικας του Arduino αναγνωρίζει τους χρόνους σε msec και για το λόγο αυτό, έχουμε 15s=15000msec, 16s=16000msec. Επίσης, όπου **do_this**, θα βάλουμε στη μια περίπτωση την κλήση της συνάρτησης **getData** με την οποία το Arduino λαμβάνει τα δεδομένα από τους αισθητήρες, και στην άλλη την συνάρτηση **updateValues** με την οποία το Arduino ενημερώνει τις τιμές στο ThingSpeak.com [Εικόνα 18].

```

void setup()
{
  pinMode(2, OUTPUT);
  Serial.begin(9600);
  //esp8266.begin(9600);
  esp8266.begin(115200);
  pulseRate.analogInput(PulseWire);
  pulseRate.blinkOnPulse(2); //Το LED που συνδέουμε στο pin 2 Arduino αναβοσβήνει με το ρυθμό της καρδιάς.
  pulseRate.setThreshold(Threshold);
  // Έλεγχος αν δημιουργήθηκε το αντικείμενο "pulseSensor" και εάν ξεκίνησε "began" να βλέπει το σήμα.
  if (pulseRate.begin()) {
    Serial.println("pulseRate Object !"); //Αυτό τυπώνεται στο Serial Monitor μαι φορά κατά την εκκίνηση ή το reset του Arduino.
  }
  Serial.println("AT");
  esp8266.println("AT");
  delay(3000);
  if(esp8266.find("OK"))
  {
    connectWiFi();
  }
  timer.every(16000, getData);// thingspeak χρειάζεται 15 sec καθυστέρηση μεταξύ των ενημερώσεων.
  timer.every(16000, updateValues);
}

```

Εικόνα 18. Συνάρτηση setup()

Η συνάρτηση **connectWiFi()** δηλώνεται ως **Boolean** και κατά συνέπεια, οι τιμές που επιστρέφει είναι True ή False. Έτσι, εάν το Wi-Fi είναι συνδεδεμένο επιστρέφει την τιμή True ενώ στην αντίθετη περίπτωση επιστρέφει false. Με την εντολή **AT+CWMODE=1** κάνουμε το esp8266 module να λειτουργεί σε κατάσταση station. Επιπλέον, η εντολή **AT+CWJAP=** (Ravi, 2018) χρησιμοποιείται για την σύνδεση του module και κατ'έπекταση του συστήματος στο Access Point (δηλ. Το WiFi Router) του χώρου που θα γίνεται η παρακολούθηση του ασθενούς (Εικόνα 19).

```

boolean connectWiFi()
{
  Serial.println("AT+CWMODE=1");// Δήλωση του esp8266 σε station mode
  esp8266.println("AT+CWMODE=1");
  delay(2000);
  String cmd="AT+CWJAP=\""; // Σύνδεση στο Access Point
  cmd+=SSID;
  cmd+="\", \"";
  cmd+=PASS;
  cmd+="\"";
  Serial.println(cmd);
  esp8266.println(cmd); // Αποστολή στο esp της μεταβλητής cmd
  delay(5000);
  if(esp8266.find("OK")) // Αν η απάντηση στο serial monitor είναι OK η συνάρτηση
  {
    return true;          // επιστρέφει true αλλιώς false
  }
  else
  {
    return false;
  }
}

```

Εικόνα 19. Συνάρτηση connectWiFi() για την σύνδεση του ESP με το Access Point του σπιτιού.

Η συνάρτηση getData() λαμβάνει δεδομένα από τον αισθητήρα παλμού και από τον αισθητήρα θερμοκρασίας LM35 (Εικόνα 20).

```

void getData(){
  analog_Temp = analogRead(A1);
  Voltage = (analog_Temp / 1023.0) * 5000; // 5000 για να πάρουμε τιμή voltage σε millivots.
  tempC = Voltage * 0.1;                  // Μετατροπή της τάσης (voltage) σε τιμή θερμοκρασίας
  Serial.println(tempC);
  int myBPM = pulseRate.getBeatsPerMinute(); // Κλήση της συνάρτησης του αντικειμένου pulseRate η οποία επιστρέφει BPM ως "int".
                                              // Αυτή η τιμή BPM αποθηκεύεται στη μεταβλητή "myBPM"
  if (pulseRate.sawStartOfBeat()) {        // Συνάρτηση του αντικειμένου η οποία κάνει συνεχώς δοκιμές
                                              // για το εάν "εμφανίζεται νέος παλμός καρδιάς".
  Serial.println(myBPM);                  // Εμφάνιση των BPM στο Serial Monitor
  }
}

```

Εικόνα 20: Η συνάρτηση getData()

Το Arduino διαβάζει την τιμή στην αναλογική είσοδο A1 και την αποθηκεύει στην μεταβλητή analog_Temp. Ο Analog-to-Digital μετατροπέας του Arduino είναι 10bit και για το λόγο αυτό, διαιρούμε την μεταβλητή analog_Temp με 1024 (2^{10}). Στην συνέχεια πολλαπλασιάζουμε με το 5000 το οποίο είναι τα 5V

της τάσης λειτουργίας του Arduino εκφρασμένα σε mV. Για να μετατρέψουμε την τιμή αυτή από Volt σε τιμή θερμοκρασίας (στην κλίμακα C^o) πολλαπλασιάζουμε με το (0.1) καθώς ο Scale factor του LM35 είναι 10mV/ °C . Η συνάρτηση getBeatPerMinute του αντικειμένου PulseRate λαμβάνει την τιμή BPM και την αποθηκεύει στη μεταβλητή myBPM.

Οι τιμές αυτών των μεταβλητών που είναι float μετατρέπονται σε string με τη χρήση της συνάρτησης **dtostrf()** (Hobbytronics.co.uk, 2019) (Εικόνα 21).

```
char outstr1[10];           // outstr1[10], outstr2[10] είναι οι πίνακες που αποθηκεύονται οι τιμές σε string format
char outstr2[10];           //
BPM = dtostrf(myBPM, 4, 1, outstr1); // Μετατροπή των μεταβλητών myBPM και tempC σε string
temp = dtostrf(tempC, 4, 1, outstr2); // dtostrf() function http://www.hobbytronics.co.uk/arduino-float-vars
```

Εικόνα 21. Μετατροπή τιμών σε string.

Δημιουργούμε την συνάρτηση updateValues() για να ενημερώνουμε το κανάλι στον ThingSpeak Server με τις τιμές των αισθητήρων.

Η χρήση της AT εντολής "AT+CIPSTART=\\"TCP\\",\\""" δημιουργεί μια σύνδεση TCP στην θύρα 80 (M2MSupport.net., 2013) (Εικόνα 22).

```
void updateValues()
{
  String cmd = "AT+CIPSTART=\\"TCP\\",\\""; // Χρήση της εντολής AT+CIPSTART για εγκαθίδρυση TCP επικοινωνίας
  cmd += IP;
  cmd += "\",80"; // Port 80
  Serial.println(cmd);
  esp8266.println(cmd);
  delay(2000);
  if(esp8266.find("Error")) // Η εκτέλεση της AT+CIPSTAR επιστρέφει OK(επιτυχή σύνδεσης)
  { // ή "error" στην αντίθετη περίπτωση.
    return;
  }
}
```

Εικόνα 22. Σύνδεση TCP.

Στη συνέχεια οι μετρήσεις που λαμβάνουν οι αισθητήρες συνδέονται με την μέθοδο GET URL (έχει αποθηκευτεί στην μεταβλητή msg) με τη χρήση των "&field1=" για τις μετρήσεις των παλμών, "&field2=" για τις μετρήσεις θερμοκρασίας και "&field4=" για τις μετρήσεις του ρυθμού αναπνοής

(Respiratory Rate). Η αποστολή όλων αυτών των πληροφοριών γίνεται με τη χρήση της εντολής “AT+CIPSEND=” (Ravi, 2018)

```

cmd = msg ;
cmd += "&field1="; //field 1 για ρυθμό παλμού (BPM)
cmd += "BPM";
cmd += "&field2="; //field 2 για θερμοκρασία
cmd += "temp";
cmd += "&field4="; //field 4 για το ρυθμό αναπνοής
cmd += "respRate";
cmd += "\r\n";
Serial.print("AT+CIPSEND="); // Εντολή AT+CIPSEND https://github.com/espressif/ESP8266\_AT/wiki/CIPSEND
esp8266.print("AT+CIPSEND=");
Serial.println(cmd.length());
esp8266.println(cmd.length());
if(esp8266.find(">")) // Μετά την εκτέλεση της εντολής επιστρέφεται ">" που δείχνει ότι το module είναι έτοιμο
// να δεχθεί δεδομένα με μήκος που υπολογίσε απο την προηγούμενη εντολή (cmd.length()).
{
  Serial.print(cmd);
  esp8266.print(cmd);
}
else
{
  Serial.println("AT+CIPCLOSE");
  esp8266.println("AT+CIPCLOSE"); // Κλείνει τη σύνδεση TCP|
  //Επανάληψη αποστολής...
  error=1;
}
}

```

Εικόνα 23. Αποστολή δεδομένων στην IoT πλατφόρμα

Ίδια λογική χρησιμοποιείται για τη δημιουργία της συνάρτησης emergency_button(). Όταν το button πάρει την τιμή HIGH, το esp8266 module στέλνει την πληροφορία στον server χρησιμοποιώντας τις εντολές AT+CIPSTART και AT+CIPSEND.

```

void emergency_button(){ // Η συνάρτηση αυτή λειτουργεί με τον ίδιο τρόπο που περιγράψαμε
  emergency = digitalRead(8); // για τη συνάρτηση updateValues()
  if(emergency == HIGH){
    Serial.println(emergency);
    /*Serial.println("PANIC");
    digitalWrite(2, HIGH); // ενεργοποίηση του LED στην θύρα 2 (HIGH είναι το επίπεδο της τάσης)
    delay(2000); // παύση για δύο δευτερόλεπτα
    digitalWrite(2, LOW); // απενεργοποίηση του LED κάνοντας την τάση στην θύρα 2 LOW
    delay(2000);
    */
    String cmd = "AT+CIPSTART=\\"TCP\\","\\"; // εγκαθίδρυση TCP επικοινωνίας
    cmd += "IP";
    cmd += "\",80";
    Serial.println(cmd);
    esp8266.println(cmd);
  }
}

```

Εικόνα 24. Η συνάρτηση emergency_button()

Η σύνδεση με την GET URL γίνεται με χρήση του “&field3=”.

```

cmd = msg ;
cmd += "&field3=";

```

Εικόνα 25. Αποστολή τιμής για το πάτημα του button.

Στο κομμάτι του κώδικα που αφορά τη μέτρηση και καταγραφή του ρυθμού της αναπνοής ξεκινάμε με την δήλωση, της μεταβλητής, **respStatus**, η οποία είναι τύπου **enum** (Arduino.cc, 2018) δηλαδή μεταβλητή η οποία μπορεί να λάβει ένα σετ τιμών. Στην Εικόνα 26 φαίνονται οι τιμές που μπορεί να λάβει.

```
enum respStatus { // Δημιουργία μεταβλητής η οποία μπορεί να λάβει ένα σετ απο τιμές/  
    Inhaling,      // https://playground.arduino.cc/Code/Enum  
    Exhaling  
};
```

Εικόνα 26. Μεταβλητή enum.

Στο κεφάλαιο 3 αναφέραμε ότι για την μέτρηση του αναπνευστικού ρυθμού χρησιμοποιούμε ένα NTC thermistor 100KΩ. Για το λόγο αυτό, όταν η θερμοκρασία αυξάνεται, η αντίσταση του thermistor μειώνεται, ενώ στην αντίθετη περίπτωση, όταν η θερμοκρασία μειώνεται, η αντίσταση αυξάνεται. Αύξηση της θερμοκρασίας παρατηρείται κατά την εκπνοή καθώς ο αέρας που εκπνέει ο ασθενής έχει τη θερμοκρασία του σώματος, η οποία είναι μεγαλύτερη από την θερμοκρασία του εξωτερικού περιβάλλοντος στο οποίο βρίσκεται ο αισθητήρας. Αντίστοιχα μείωση της θερμοκρασίας παρατηρείται κατά την εισπνοή. Οι τιμές του αισθητήρα που διαβάζει το Arduino στην είσοδο A2 είναι >400 κατά την εισπνοή και <100 κατά την εκπνοή. Οι τιμές αυτές είναι τιμές τάσης εκφρασμένες σε ψηφιακές μονάδες στο εύρος 0 – 1024 μιας και ο A/D μετατροπέας του Arduino είναι 10 bit ($2^{10}=1024$). Την πληροφορία αυτή θα χρησιμοποιήσουμε στον κώδικα για να ανιχνεύουμε την εναλλαγή εισπνοής/εκπνοής του ασθενούς (Εικόνα 27).

```

int sensorValue = analogRead(A2);
//sensorValue = (1023 / sensorValue) - 1;
// sensorValue = SERIESRESISTOR / sensorValue;
//Serial.print("Thermistor resistance ");
//Serial.println(sensorValue);

//delay(2000);
//}
if ((sensorValue > 400) && (currentRespState == Exhaling)) // 400 είναι η τιμή που διαβάζει ο A/D μετατροπέας
  { // στην είσοδο A2
    currentRespState = Inhaling;
    prevRespTime = RespTime; // Αποθήκευση χρόνου του τελευταίου κύκλου της αναπνοής
    RespTime = millis();

    float period_respiration = RespTime - prevRespTime; // Περίοδος της αναπνοής

    if (period_respiration == 0){
      breathrate = 0;
      Serial.println(breathrate);
    }
    else {
      breathrate = 60000 / period_respiration; // αναγωγή του ρυθμού αναπνοής στο λεπτό

    }

  }

else if ((currentRespState == Inhaling) && (sensorValue < 100))
  {
    currentRespState = Exhaling;
    Serial.println("Exhaling");
  }
}

```

Εικόνα 27. Μέτρηση ρυθμού αναπνοής.

Με τον παραπάνω κώδικα γίνεται υπολογισμός της στιγμιαίας τιμής του αναπνευστικού ρυθμού. Αρχικά με τη χρήση της συνάρτησης `millis()` καταχωρείται στη μεταβλητή `RespTime` ο χρόνος που έχει παρέλθει από την έναρξη εκτέλεσης του κώδικα. Σε κάθε νέο κύκλο (εκτέλεση `loop`) η τιμή αυτή αντικαθίσταται με την νέα τιμή του χρόνου που έχει παρέλθει. Στη συνέχεια με την εντολή **period_respiration** υπολογίζουμε την περίοδο της αναπνοής και τέλος, με την εντολή **breathrate=60000/period_respiration** κάνουμε την αναγωγή στο 1 λεπτό για να υπολογίσουμε τον αναπνευστικό ρυθμό σε bpm (breath per minutes). Με τον τρόπο αυτό γίνεται μεν η μέτρηση του αναπνευστικού ρυθμού, οι τιμές όμως μπορεί να έχουν μεγάλες αποκλίσεις από κύκλο σε κύκλο μέτρησης. Αυτό συμβαίνει γιατί δεν έχουμε ορίσει κάποιο είδος υστέρησης μεταξύ των τιμών, δηλαδή την εξάρτηση που έχει η τρέχουσα τιμή με τις προηγούμενες. Είναι το επόμενο βήμα που εφαρμόζεται για να έχουμε κάποιο είδος εξομάλυνσης στις τιμές. Για το λόγο, αυτό τις τιμές που υπολογίζουμε με τον τρόπο αυτό τις εισάγουμε σε ένα αλγόριθμο ψηφιακού low pass φίλτρου για τον υπολογισμό των τιμών κινητού μέσου όρου (Εικόνα 28).


```

if (respRate == 0)
  respRate = breathrate;
else {
  respRate = (int) ((0.2 * breathrate) + (0.8 * respRate)); //https://helpful.knobs-dials.com/index.php/Low-pass_filter
//https://www.norwegiancreations.com/2016/03/arduino-tutorial-simple-high-pass-band-pass-and-band-stop-filtering/
}

```

Εικόνα 28. Εφαρμογή με κώδικα φίλτρου low-pass.

Η μέθοδος κινητού μέσου όρου (moving average) αντιστοιχεί στον υπολογισμό του μέσου όρου του δείγματος των μετρήσεων και στην χρησιμοποίηση αυτού για την απόρριψη τιμών που απέχουν-αποκλίνουν αρκετά. Ο όρος «κινητός μέσος όρος» χρησιμοποιείται αφού για κάθε νέα ή νέες μετρήσεις που προστίθενται στο δείγμα των μετρήσεων, υπολογίζεται ένας νέος μέσος όρος αντικαθιστώντας τον παλαιότερο. Καθώς υπολογίζεται κάθε νέος μέσος όρος, η διαδικασία της απόρριψης τιμών λόγω σημαντικής απόκλισης από αυτόν επαναλαμβάνεται. Η μέθοδος του κινητού μέσου όρου δεν είναι ιδανική, είναι όμως πιο αποτελεσματική από τη χρησιμοποίηση του απλού αριθμητικού μέσου ολόκληρου του δείγματος για την απόρριψη τιμών (Βαρδάκας κ.ά., 2017).

Ο τύπος που χρησιμοποιούμε για να υπολογίσουμε την φιλτραρισμένη έξοδο είναι (Knobs-dials.com, 2019)

$$\text{filtered_output}[i] = \alpha * \text{raw_input}[i] + (1 - \alpha) * \text{filtered_output}[i - 1]$$

Στην συνάρτηση loop() γίνεται κλήση της emergency_button() και των timers με τη χρήση της εντολής timer.update() (Εικόνα 29).

```

void loop()
{
  emergency_button();
  startingPoint: //label
  error=0;
  timer.update();
  //Επανάληψη αποστολής αν η μετάδοση δεν ολοκληρωθεί
  if (error==1)
  {
    goto startingPoint; //Πήγανε στο label "startingPoint"
  }
  //delay(4000);
}

```

Εικόνα 29. Κλήση συνάρτησης emergency_button() και των timers.

Η συνάρτηση **timer.update()** πρέπει να εκτελείται μέσω της **loop**. Με την κλήση της, εκτελούνται όλα τα events τα οποία είναι συνδεδεμένα με το αντικείμενο timer.

Οι timers στον κώδικα είναι:

```

timer.every(5000, getData);
timer.every(5000, updateValues);

```

Εικόνα 30. Timers

5.2 Παραμετροποίηση της πλατφόρμας ThingSpeak και υλοποίηση της online παρακολούθησης.

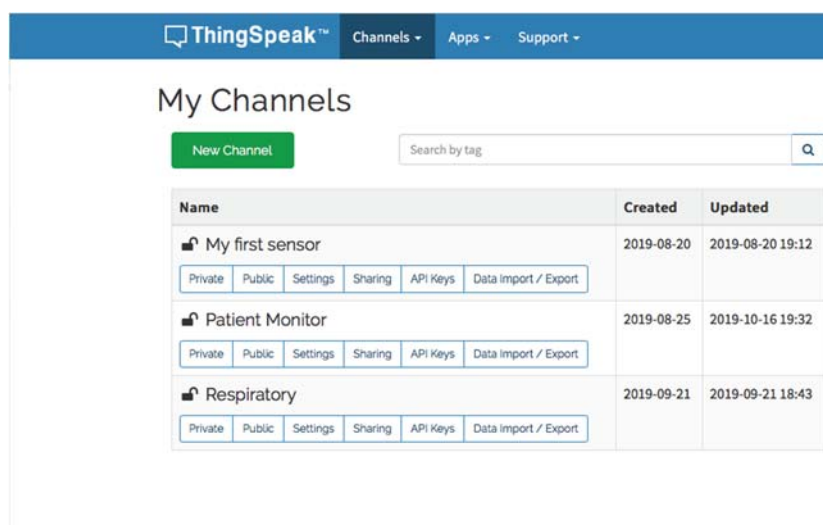
5.2.1 Πλατφόρμα ThingSpeak.com

Όπως αναφέραμε στο κεφάλαιο 4 θα χρησιμοποιήσουμε την IoT πλατφόρμα ThingSpeak για την online παρακολούθηση των ζωτικών παραμέτρων (καρδιακοί παλμοί, θερμοκρασία σώματος, ρυθμός αναπνοής) του ασθενή μέσω διαδικτύου. Θα χρησιμοποιήσουμε επίσης, την πλατφόρμα **IFTTT.com** για την σύνδεση του ThingSpeak με υπηρεσίες email/SMS έτσι ώστε να υπάρχει

δυνατότητα αποστολής μηνυμάτων/ειδοποιήσεων έκτακτης ανάγκης κάθε φορά που ο ασθενής βρίσκεται σε κρίσιμη κατάσταση.

Αρχικά θα δημιουργήσουμε μέσω μιας απλής διαδικασίας ένα λογαριασμό χρήστη στο ThingSpeak.com και θα κάνουμε σύνδεση με τα στοιχεία αυτά. Χρήστης μπορεί να είναι ο γιατρός ή μια ομάδα γιατρών που παρακολουθεί τον ασθενή, κάποιος συγγενής ή οποιοσδήποτε που είναι σε θέση να παρακολουθεί τα δεδομένα αυτά. Για το λόγο αυτό, username και κωδικοί πρόσβασης πρέπει να παρέχονται στους παραπάνω.

Θα δημιουργήσουμε ένα κανάλι παρακολούθησης για τον ασθενή. Σε περίπτωση που κάποιος γιατρός παρακολουθεί πολλούς ασθενείς μπορούμε να δημιουργήσουμε ένα κανάλι παρακολούθησης ανά ασθενή. Χρησιμοποιώντας την φόρμα (Εικόνα 31) δημιουργίας καναλιού θα δώσουμε στο κανάλι ένα όνομα και μια περιγραφή.



Name	Created	Updated
My first sensor	2019-08-20	2019-08-20 19:12
Patient Monitor	2019-08-25	2019-10-16 19:32
Respiratory	2019-09-21	2019-09-21 18:43

Εικόνα 31. Κανάλια στην πλατφόρμα ThingSpeak.com

Το κανάλι που υλοποιούμε είναι το Patient Monitor. Στη συνέχεια συμπληρώνουμε τα πεδία field 1, field 2, field 3, field 4 με τις παραμέτρους Pulse Rate, Temperature, emergency και Respiratory αντίστοιχα (Εικόνα 32).

Channel Settings

Percentage complete 30%

Channel ID 852326

Name

Description

Field 1

Field 2

Field 3

Field 4

Field 5

Field 6

Field 7

Field 8

Metadata

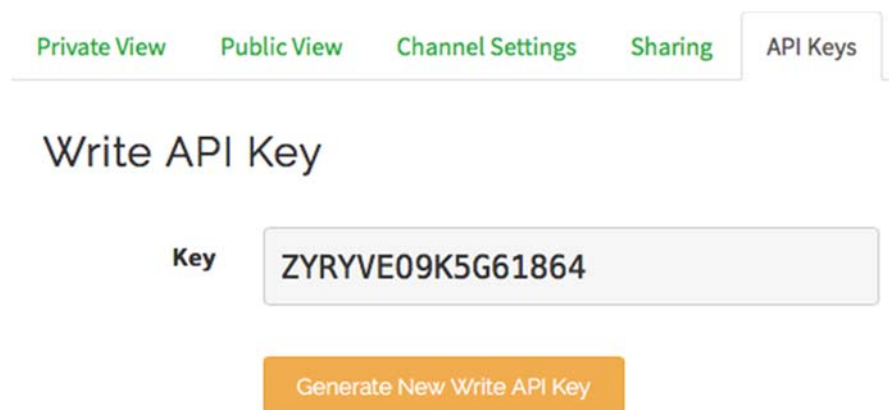
Εικόνα 32. Ρυθμίσεις καναλιού Patient Monitor

Με βάση τις παραπάνω ρυθμίσεις θα εμφανιστούν τέσσερα γραφήματα, ένα για κάθε παράμετρο (Εικόνα 33).



Εικόνα 33. Γραφήματα των παραμέτρων

Σημειώνεται ότι το **Write API Key** (Εικόνα 34) που δημιουργείται για το συγκεκριμένο κανάλι είναι αυτό που χρησιμοποιούμε στον κώδικα του Arduino.



Εικόνα 34. API key

5.2.2 Εφαρμογές ThingHTTP

Για την ενεργοποίηση/σκανδαλισμό των IFTTT applets με σκοπό την αποστολή email/SMS θα χρησιμοποιήσουμε τις εφαρμογές ThingHTTP της πλατφόρμας. Οι εφαρμογές ThingHTTP ενεργοποιούν την επικοινωνία μεταξύ συσκευών, ιστοσελίδων, υπηρεσιών διαδικτύου χωρίς να είναι απαραίτητη η εφαρμογή του πρωτοκόλλου στο επίπεδο συσκευής. Μπορούν να καθοριστούν ενέργειες που θέλουμε να ενεργοποιήσουμε στην ThingHTTP με την χρήση άλλων ThingSpeak εφαρμογών όπως τα **React**.

Για τη δημιουργία νέου ThingHTTP (Εικόνα 35) θα χρειαστούμε ένα URL για τον σκανδαλισμό τον οποίο θα πάρουμε από την πλατφόρμα IFTTT.

Apps / ThingHTTP

New ThingHTTP

Name	Created
Patient Monitor <input type="button" value="View"/> <input type="button" value="Edit"/>	2019-08-25
Emergency <input type="button" value="View"/> <input type="button" value="Edit"/>	2019-08-25
resp <input type="button" value="View"/> <input type="button" value="Edit"/>	2019-10-14
respsms <input type="button" value="View"/> <input type="button" value="Edit"/>	2019-10-14

Εικόνα 35. Εφαρμογές ThingHTTP

5.2.3 Ρύθμιση IFTTT για ενεργοποίηση αποστολής email/sms με βάση τις τιμές στο ThingSpeak

Στην ιστοσελίδα **IFTTT.com** δημιουργούμε ένα λογαριασμό με τον οποίο μπορούμε να εισέλθουμε. Αρχικά μέσω της υπηρεσίας **Webhooks** δημιουργούμε ένα event το οποίο θα ενεργοποιούμε με την χρήση μεθόδων POST ή GET web request. Ειδικότερα θα δημιουργήσουμε αρχικά ένα event για την περίπτωση που ο ρυθμός αναπνοής του ασθενή βγει εκτός ορίων (< από 8 bpm και > από 16 bpm). Η υπηρεσία Webhooks μας δίνει το URL το οποίο θα χρησιμοποιήσουμε στο ThingHTTP (Εικόνα 36).



Your key is: **brtazMbEWOjZKFdb-GSFpW**

[← Back to service](#)

To trigger an Event

Make a POST or GET web request to:

```
https://maker.ifttt.com/trigger/respiratory/with/key/brtazMbEWOjZKFdb-GSFpW
```

With an optional JSON body of:

```
{ "value1" : " ", "value2" : " ", "value3" : " " }
```

The data is completely optional, and you can also pass value1, value2, and value3 as query parameters or form variables. This content will be passed on to the Action in your Recipe.

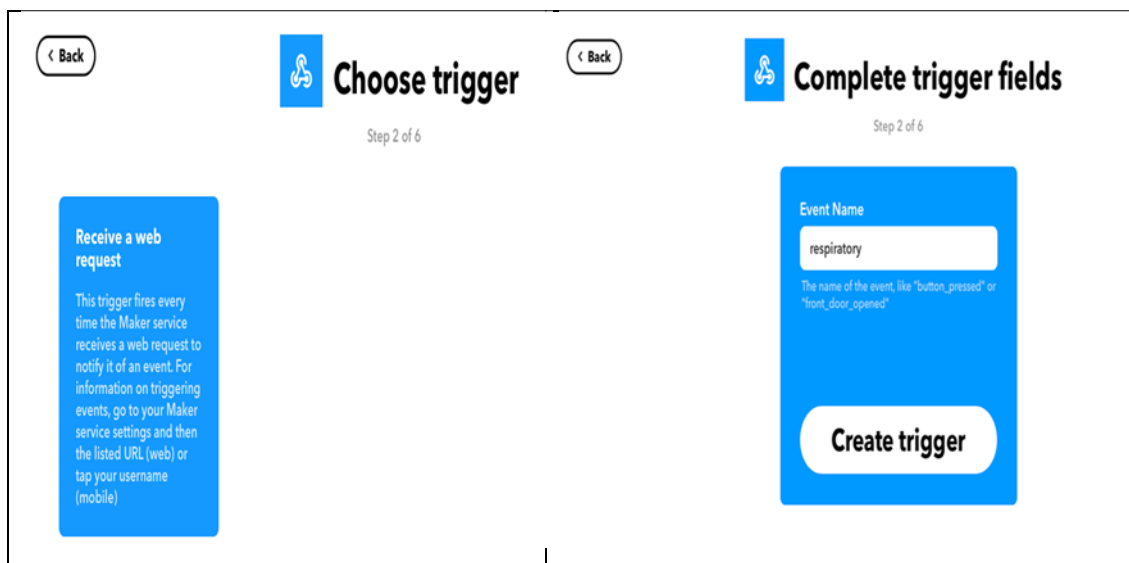
You can also try it with `curl` from a command line.

```
curl -X POST https://maker.ifttt.com/trigger/respiratory/with/key/brtazMbEWOjZKFdb-GSFpW
```

Εικόνα 36. Δημιουργία URL για την πυροδότηση του event respiratory

Με τον ίδιο τρόπο δημιουργούμε ένα URL για το event emergency button.

Στο επόμενο βήμα δείχνουμε πως θα δημιουργήσουμε τα applets για να μπορούν οι εφαρμογές ThingHTTP να αποστέλλουν email/SMS. Η διαδικασία είναι απλή. Ξεκινάμε ρυθμίζοντας τον τρόπο με τον οποίο θα γίνεται ο σκανδαλισμός των events που για την δικιά μας περίπτωση είναι το “Receive a web request” (Εικόνα 37).



Εικόνα 37. Web requests και Trigger

Το όνομα στο Event Name πρέπει να είναι ίδιο με το όνομα που δώσαμε στο URL που δημιουργήσαμε με την υπηρεσία WebHooks. Έτσι, μέσω του Gmail θα στέλνεται email σε παραλήπτες που θα ορίσουμε εμείς κάθε φορά που θα συμβαίνει σκανδαλισμός στο γεγονός respiratory (Εικόνα 38).

To address
dkatselas@yahoo.gr
Accepts up to twenty email addresses, each separated with a space or comma **Add ingredient**

CC address
Accepts up to twenty email addresses, each separated with a space or comma **Add ingredient**

BCC address
Accepts up to twenty email addresses, each separated with a space or comma **Add ingredient**

Subject
The event named respiratory occurred on the Maker Webhooks service **Add ingredient**

Body
respiratory
Patient: Dimitris Katselas
What: respiratory

When: {{OccurredAt}}

Extra Data: {{Value1}}, {{Value2}},

Review and finish

Step 6 of 6

If Maker Event "respiratory", then Send an email from dkatselas@gmail.com 73/140
by dkatselas

Receive notifications when this Applet runs

Finish

Εικόνα 38. Δημιουργία event και ρυθμίσεις

Όπως αναφέρθηκε παραπάνω θα δημιουργήσουμε ένα δεύτερο web request για την αποστολή SMS όταν ο ασθενής πατάει το button έκτακτης ανάγκης. Η υπηρεσία Webhooks μας δίνει το URL το οποίο θα χρησιμοποιήσουμε στο ThingHTTP (Εικόνα 39).



Your key is: **brtazMbEWOjZKFdb-GSFpW**

[Back to service](#)

To trigger an Event

Make a POST or GET web request to:

```
https://maker.ifttt.com/trigger/emergency/with/key/brtazMbEWOjZKFdb-GSFpW
```

With an optional JSON body of:

```
{ "value1" : " ", "value2" : " ", "value3" : " " }
```

The data is completely optional, and you can also pass value1, value2, and value3 as query parameters or form variables. This content will be passed on to the Action in your Recipe.

You can also try it with `curl` from a command line.

```
curl -X POST https://maker.ifttt.com/trigger/emergency/with/key/brtazMbEWOjZKFdb-GSFpW
```

Test it

Εικόνα 39. Δημιουργία URL για την πυροδότηση του event emergency

Στο applet (Εικόνα 40) που θα δημιουργήσουμε για την αποστολή SMS θα χρησιμοποιήσουμε την υπηρεσία (ClickSend SMS, 2019).

The image shows two side-by-side screenshots of the IFTTT applet creation interface. The left screenshot is titled "Complete action fields" (Step 5 of 6) and contains three input sections: "To" with the number "+306932454798", "From" with "Patient: Patient Name", and "Message" with "Value 1: Patient name
Value 2: Alert
". Each section has an "Add ingredient" button. A "Create action" button is at the bottom. The right screenshot is titled "Review and finish" (Step 6 of 6) and shows a blue card with the trigger logic: "If Maker Event 'emergency', then Send an SMS message" by dkatselas (52/140). Below the card is a toggle for "Receive notifications when this Applet runs" which is turned on. A large "Finish" button is at the bottom.

Εικόνα 40. Δημιουργία event "emergency" και ρυθμίσεις.

Το επόμενο βήμα μετά την ολοκλήρωση της δημιουργίας των applets είναι να δημιουργήσουμε δύο εφαρμογές ThingHTTP μέσω της πλατφόρμας ThingSpeak. Έτσι, θα δημιουργήσουμε τις ThingHTTP εφαρμογές respiratory και emergency στις οποίες θα αντιγράψουμε τα δυο URL που δημιουργήσαμε μέσω της υπηρεσίας webhooks. Συμπληρώνουμε τις υπόλοιπες πληροφορίες όπως φαίνεται στην εικόνα 41.

The image shows two side-by-side screenshots of the ThingHTTP configuration interface. The left screenshot is for an app named 'resp' and the right is for an app named 'Emergency'. Both interfaces have the same layout with the following fields:

- Name:** resp (left), Emergency (right)
- API Key:** E1VQFLPCYTC16AMI (left), 2MXN4BEXQBMI7VJW (right)
- URL:** https://maker.ifttt.com/trigger/respiratory/with/key/brtazMbE (left), https://maker.ifttt.com/trigger/emergency/with/key/brtazMbE (right)
- HTTP Auth Username:** (empty)
- HTTP Auth Password:** (empty)
- Method:** POST (dropdown)
- Content Type:** application/json
- HTTP Version:** 1.1 (dropdown)
- Host:** (empty)
- Headers:** A table with 'Name' and 'Value' columns, and a 'remove header' button. Below the table is an 'add new header' button.
- Body:** A text area containing the JSON string: {"value1": "%channel_852326_field_4%"} (left), (empty) (right)
- Parse String:** (empty)
- Save ThingHTTP:** A green button at the bottom of each form.

Εικόνα 41. ThingHTTP εφαρμογές resp και Emergency

Στο ThingHTTP resp και συγκεκριμένα στο πεδίο Body η τιμή 852326 είναι το ID του καναλιού και field_4 είναι η μεταβλητή που λαμβάνει τις τιμές του ρυθμού αναπνοής από τον αισθητήρα (thermistor).

Έχοντας στη διάθεση μας τα ThingHTTP μπορούμε να δημιουργήσουμε τα React για τον σκανδαλισμό των URLs. Ένα React συνεργάζεται με μία εφαρμογή ThingHTTP για την εκτέλεση μιας ενέργειας (action) όταν τα δεδομένα του καναλιού πληρούν κάποιες συγκεκριμένες προϋποθέσεις. Δημιουργούμε αντίστοιχα δύο Reacts δίνοντας τα ονόματα Emergency και resp και συνδέοντάς τα με τα αντίστοιχα ThingHTTP. Συμπληρώνουμε τις υπόλοιπες πληροφορίες όπως φαίνεται στην εικόνα 42.

Με βάση την βιβλιογραφία όταν ο ρυθμός αναπνοής είναι μικρότερος από 8 bpm και μεγαλύτερος από 16 bpm θεωρείται μη φυσιολογικός. Για το λόγο αυτό, στο react θέσαμε αυτά τα όρια. Αντίστοιχα, στο react emergency η προϋπόθεση για την εκτέλεση ενός Action είναι: is equal to 1, δηλαδή το button έχει πατηθεί και είναι σε κατάσταση High.

The image shows two side-by-side configuration panels for Reacts in the ThingHTTP interface. Both panels have the following fields:

- React Name:** A text input field containing 'resp' (left) and 'Emergency' (right).
- Condition Type:** A dropdown menu set to 'Numeric'.
- Test Frequency:** A dropdown menu set to 'On Data Insertion'.
- Condition:** A dropdown menu set to 'If channel'.
- Channel:** A dropdown menu set to 'Patient Monitor (852326)'.
- Field:** A dropdown menu. On the left, it is set to '4 (Respiratory)'. On the right, it is set to '3 (emergency)'.
- Operator:** A dropdown menu. On the left, it is set to 'is greater than'. On the right, it is set to 'is equal to'.
- Value:** A text input field. On the left, it contains '16'. On the right, it contains '1'.
- Action:** A dropdown menu set to 'ThingHTTP'.
- then perform ThingHTTP:** A dropdown menu. On the left, it is set to 'resp'. On the right, it is set to 'Emergency'.
- Options:** Two radio buttons. The first is 'Run action only the first time the condition is met' (unselected). The second is 'Run action each time condition is met' (selected).
- Save React:** A green button at the bottom of each panel.

Εικόνα 42. Ρυθμίσεις εφαρμογών React

Κεφάλαιο 6

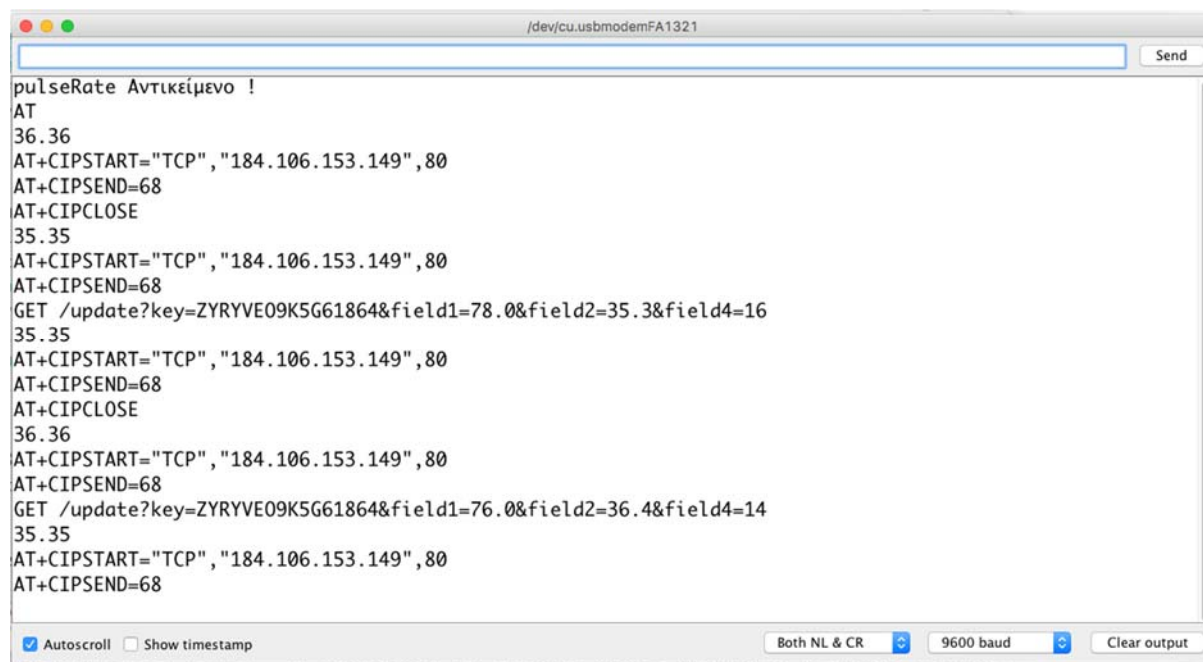
Αποτελέσματα και Ανάλυση Δεδομένων

6.1 Τελική Μορφή του Συστήματος Patient Monitor

Για την εγκατάσταση του hardware χρησιμοποιήθηκαν αρκετά καλώδια (jumper wires) και breadboards για την τοποθέτηση των εξαρτημάτων.

Μετά τη σύνδεση και τον προγραμματισμό όλων των εξαρτημάτων και των μονάδων μεταξύ τους, έγινε ο έλεγχος λειτουργίας. Όπως αναφέρθηκε στα προηγούμενα κεφάλαια, το σύστημα που αναπτύξαμε είναι ένα IoT σύστημα παρακολούθησης της υγείας ασθενούς τα συστατικά του οποίου είναι η πλακέτα ανάπτυξης Arduino Uno, αισθητήρες μέτρησης καρδιακού παλμού, αναπνευστικού ρυθμού, θερμοκρασίας σώματος, η μονάδα Wi-Fi για τις επικοινωνίες και το κύκλωμα ενίσχυσης του σήματος που λαμβάνεται από το NTC θερμίστορ. Στην παρακάτω εικόνα βλέπουμε την τελική εγκατάσταση του όλου συστήματος.

TCP επικοινωνία με τον server ο οποίος έχει IP Address 184.106.153.149 μέσω της πόρτας 80. Επίσης, η εντολή AT+CIPSEND δίνει τις τιμές των παραμέτρων Pulse Rate, Temperature και Respiratory Rate που το Arduino στέλνει στον server. Τέλος, βλέπουμε και το API key που χρησιμοποιείται.



```

/dev/cu.usbmodemFA1321
pulseRate Αντικείμενο !
AT
36.36
AT+CIPSTART="TCP", "184.106.153.149", 80
AT+CIPSEND=68
AT+CIPCLOSE
35.35
AT+CIPSTART="TCP", "184.106.153.149", 80
AT+CIPSEND=68
GET /update?key=ZYRYVE09K5G61864&field1=78.0&field2=35.3&field4=16
35.35
AT+CIPSTART="TCP", "184.106.153.149", 80
AT+CIPSEND=68
AT+CIPCLOSE
36.36
AT+CIPSTART="TCP", "184.106.153.149", 80
AT+CIPSEND=68
GET /update?key=ZYRYVE09K5G61864&field1=76.0&field2=36.4&field4=14
35.35
AT+CIPSTART="TCP", "184.106.153.149", 80
AT+CIPSEND=68

```

Εικόνα 44. Το Serial Monitor του Arduino IDE

6.3 Live Παρακολούθηση και Εξαγωγή Δεδομένων

Τα δεδομένα συλλέγονται από τους αισθητήρες και συγχρόνως μεταφορτώνονται στον cloud server (ThingSpeak) (εικόνα 45). Τα δεδομένα απεικονίζονται σε γραφήματα σε αναλογία με το χρόνο. Ο server με βάση τις προδιαγραφές του ανανεώνει τα δεδομένα κάθε 15sec. Αυτό όμως δεν συμβαίνει στην πραγματικότητα. Ο χρόνος ανανέωσης είναι μεγαλύτερος καθώς επηρεάζεται και από άλλους παράγοντες (ταχύτητα δικτύου, απόκριση αισθητήρων κ.ά.). Μπορεί να γίνει εξαγωγή αυτών των δεδομένων οποιαδήποτε στιγμή χρησιμοποιώντας τις δυνατότητες εξαγωγής που προσφέρει ο server. Ο γιατρός ή ο συγγενής μπορεί να δει τα δεδομένα

οποιαδήποτε στιγμή από οπουδήποτε αρκεί να έχει πρόσβαση στην πλατφόρμα.

6.4 Μετρήσεις Pulse Rate και Θερμοκρασίας

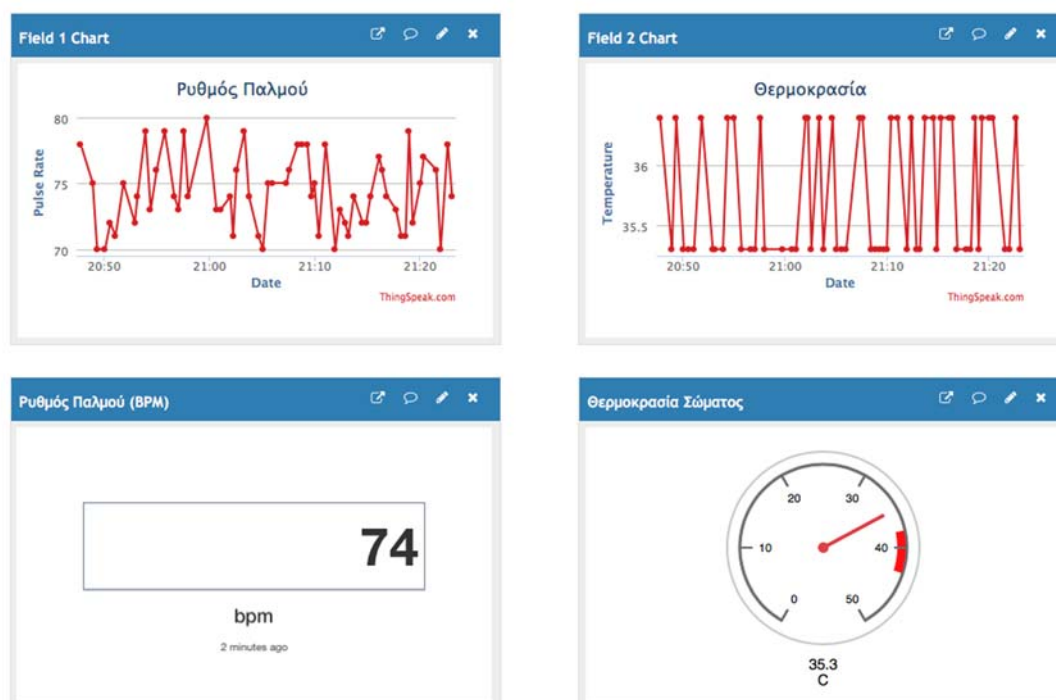
Στην παρακάτω εικόνα βλέπουμε τα γραφήματα για τον Ρυθμό Παλμού της καρδιάς και την θερμοκρασία. Συμπληρωματικά προστέθηκαν δύο Widgets στα οποία βλέπουμε με διαφορετικό τρόπο τις τελευταίες τιμές που έχει λάβει ο Server.

Channel Stats

Created: 2 months ago

Last entry: less than a minute ago

Entries: 1499

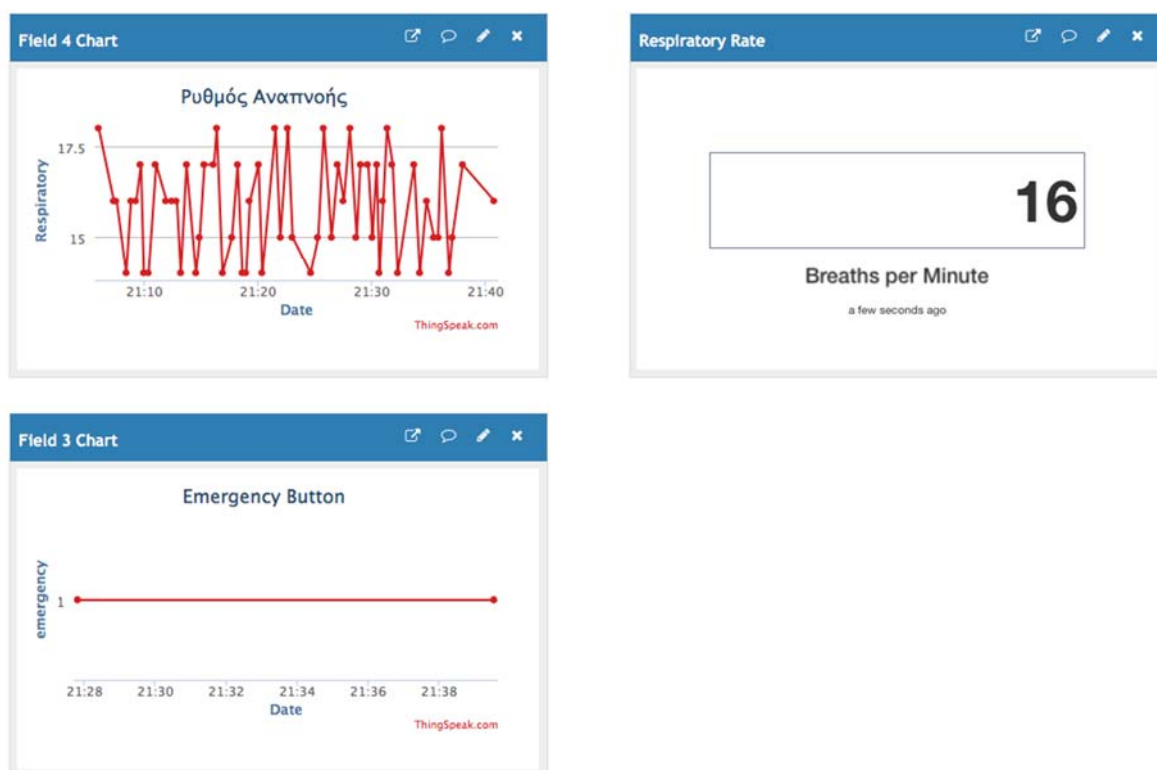


Εικόνα 45. Γραφήματα και Widgets για Ρυθμό παλμού και θερμοκρασία σώματος

6.5 Ρυθμός Αναπνοής και Emergency Button

Με τον ίδιο τρόπο απεικονίζονται στην παρακάτω εικόνα τα γραφήματα του Ρυθμού Αναπνοής και του Emergency Button. Ειδικά για την περίπτωση του

Emergency Button σημειώνεται ότι η κόκκινη βούλα που εμφανίζεται στο επίπεδο 1 σηματοδοτεί το πάτημα του Button.



Εικόνα 46. Γραφήματα και Widgets για Ρυθμό Αναπνοής και emergency button

6.6 Ειδοποιήσεις και Συναγερμοί

Στο κεφάλαιο 5 αναφερθήκαμε στα applets που δημιουργήθηκαν στην πλατφόρμα IFTTT.com με σκοπό την παροχή ειδοποιήσεων συναγερμού. Έτσι, για καταστάσεις που χρήζουν της προσοχής του γιατρού ή για επείγουσες καταστάσεις το σύστημα έχει τη δυνατότητα να στέλνει ειδοποιήσεις είτε με email είτε με SMS.

Manage your Applets

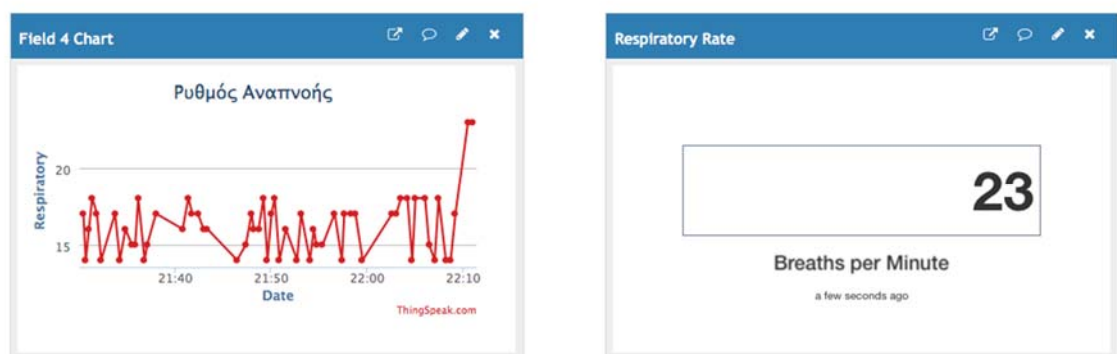
You can scroll through a list of them here or create your own. [Learn more](#)

Got it

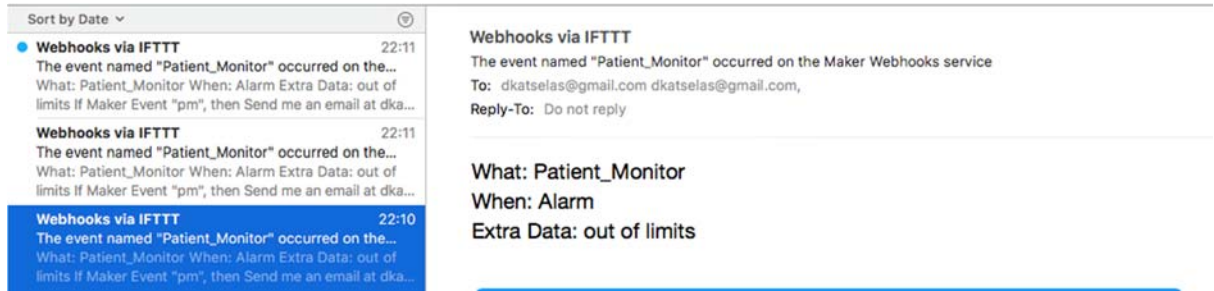
My Applets

Εικόνα 47. IFTTT Applets.

Για την περίπτωση που ο ασθενής που παρακολουθείται έχει ταχύπνοια (αυξημένος αριθμός αναπνοών) δημιουργήθηκε μια εφαρμογή ThingHTTP η οποία στέλνει ένα email σε μια προκαθορισμένη διεύθυνση όταν ο ρυθμός αναπνοής είναι μεγαλύτερος από 22 αναπνοές ανά λεπτό. Στις εικόνες 48 και 49 βλέπουμε τα emails που λαμβάνονται όταν ο ρυθμός αναπνοής είναι 23 bpm. (Σημείωση: Όταν μιλάμε για αυξημένο αριθμό αναπνοών εννοούμε παραπάνω από 20 αναπνοές το λεπτό που μπορεί να φτάσουν και ως τις 30 με 35. Η επιλογή για 22 αναπνοές είναι απλά για την εκτέλεση του πειράματος).

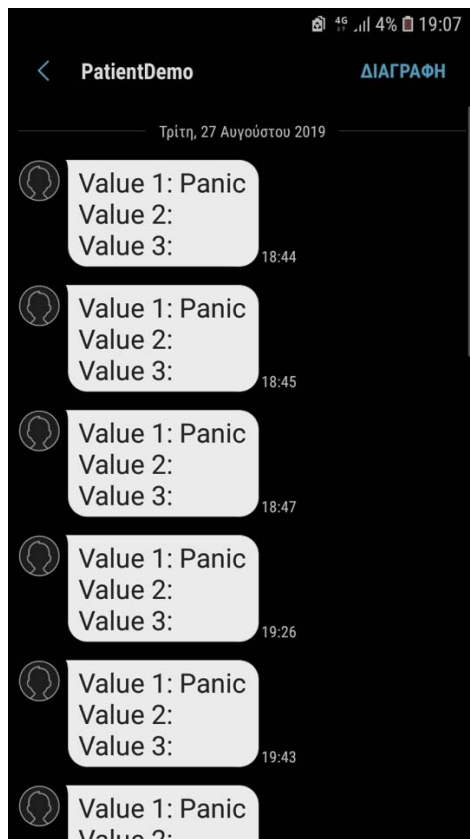


Εικόνα 48. Γραφήματα όταν ο ρυθμός αναπνοής είναι μεγαλύτερος από 22bpm



Εικόνα 49. Emails με ειδοποίηση συναγερμού υπέρβασης ορίων για τον αναπνευστικό ρυθμό.

6.7 Αποστολή SMS

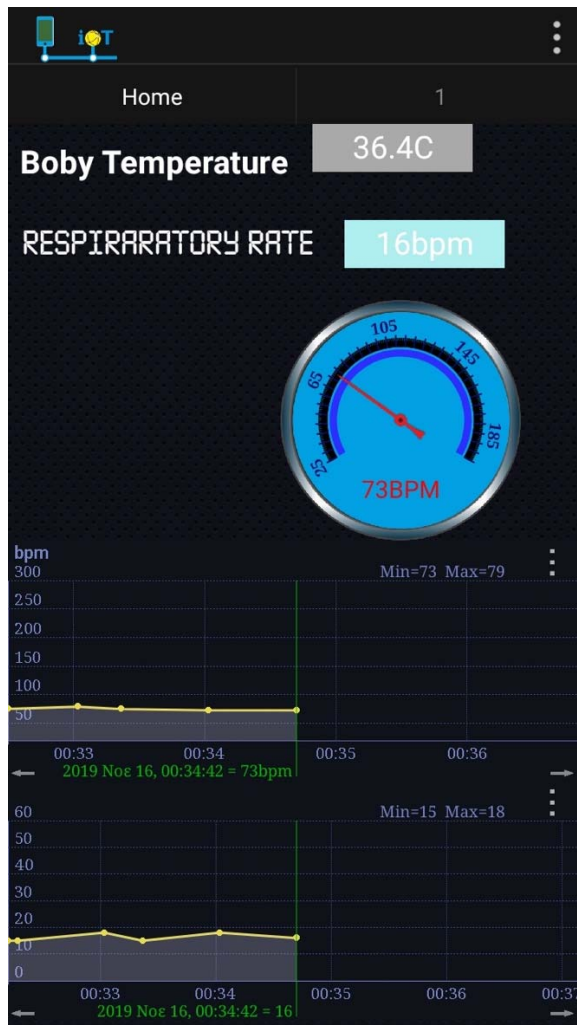


Εικόνα 50. SMS που λαμβάνονται όταν ο ασθενής πατάει το Emergency Button

6.8 VIRTUINO

Ο γιατρός ή ο συγγενής που παρακολουθεί τον ασθενή έχει τη δυνατότητα να βλέπει τα δεδομένα από μια κινητή συσκευή με τη χρήση της εφαρμογής

Virtuino για Android. Στην παρακάτω εικόνα βλέπουμε το πάνελ που δημιουργήσαμε έτσι ώστε ο χρήστης να μπορεί να βλέπει τις τιμές Pulse Rate, Θερμοκρασία Σώματος, Respiratory Rate σε πραγματικό χρόνο αλλά και μέσω γραφημάτων την εξέλιξη τους στο χρόνο.



Εικόνα 51. Πάνελ της εφαρμογής Virtuino με τις μετρήσεις πραγματικού χρόνου για όλες τις παραμέτρους και τα γραφήματα του καρδιακού και του αναπνευστικού ρυθμού.

Κεφάλαιο 7

Συμπεράσματα

7.1 Επίλογος – Συμπεράσματα

Στην παρούσα διπλωματική εργασία έγινε μια προσπάθεια να υλοποιηθεί ένα σύστημα απομακρυσμένης παρακολούθησης ασθενών με χρόνιες παθήσεις. Το πιο σημαντικό χαρακτηριστικό αυτού του συστήματος είναι ότι η κατάσταση της υγείας του ασθενούς μπορεί να παρακολουθείται ενώ ο ίδιος βρίσκεται στο σπίτι του καθώς και το ότι θα μπορούσαν να ληφθούν τα απαραίτητα μέτρα στην περίπτωση αδιαθεσίας του.

Ο σχεδιασμός έγινε με τέτοιο τρόπο ώστε το σύστημα να είναι εύχρηστο πρωτίστως για τον ασθενή αλλά και για το άτομο που θα αναλάβει την παρακολούθησή του. Σε αυτό συνετέλεσαν κυρίως οι τεχνολογίες που χρησιμοποιήθηκαν. Η εξέλιξη του διαδικτύου σε συνδυασμό με την εξέλιξη στις online IoT πλατφόρμες καθώς και στους προσωπικούς υπολογιστές δίνει την δυνατότητα δημιουργίας και εφαρμογής συστημάτων τα οποία πριν από μερικά χρόνια απαιτούσαν πολύ υψηλό κόστος και εξειδικευμένο εξοπλισμό. Επιπρόσθετα, η εκμάθηση όλων αυτών των εργαλείων είναι πλέον σχετικά σύντομη και εύκολη, ενώ οι δυνατότητες που παρέχουν είναι πολύ εξελιγμένες. Το σύστημα που υλοποιήσαμε μπορεί να παράσχει στους ίδιους τους ασθενείς την δυνατότητα να παρακολουθούν μόνοι τους (ή με τη βοήθεια τρίτων) την υγεία τους. Το χαμηλό κόστος αγοράς και το μικρό μέγεθος του Arduino Uno, καθιστά εύκολη την απόκτησή του και τη χρήση του σε οποιαδήποτε τοποθεσία διαθέτει Wi-Fi σύνδεση στο διαδίκτυο. Επιπλέον, τα δεδομένα των μετρήσεων ενημερώνονται συνεχώς στην υπηρεσία cloud σε κανονική περίοδο χρόνου. Αυτό βοηθάει τους γιατρούς ή τους συγγενείς του ασθενούς να παρακολουθούν οποιαδήποτε στιγμή την κατάσταση της υγείας του ασθενούς,

καθώς επίσης και στο να λαμβάνουν μέτρα που μπορεί να χρειαστούν την κατάλληλη στιγμή. Το σύστημα έχει επίσης τη δυνατότητα να στέλνει στο γιατρό ή στους συγγενείς ειδοποιήσεις συναγερμού μέσω sms και email, όταν οι τιμές των παραμέτρων που παρακολουθούνται ξεφύγουν από προκαθορισμένα όρια. Σημαντική είναι επίσης, η μειωμένη πιθανότητα ανθρώπινου λάθους καθώς τα βιοσήματα λαμβάνονται από αισθητήρες, καθώς και η συμβολή του συστήματος στη βελτίωση του τομέα της υγειονομικής περίθαλψης μειώνοντας τις δαπάνες θεραπείας. Ένα τέτοιο σύστημα βοηθά στη βελτίωση της υγείας του ασθενούς μειώνοντας την ανάγκη για εισαγωγή του ασθενούς στο νοσοκομείο παρέχοντας τη δυνατότητα παρακολούθησης στο οικείο περιβάλλον του σπιτιού του.

7.2 Παρατηρήσεις – Περιορισμοί

Το σύστημα δεν μπορεί να θεωρηθεί ένα πλήρες σύστημα τηλεπαρακολούθησης ασθενών αλλά σίγουρα αποτελεί μια καλή βάση ώστε πάνω σε αυτή να αναπτυχθούν και να προστεθούν νέες δυνατότητες. Οι αισθητήρες που χρησιμοποιήσαμε είναι κυρίως για ανάπτυξη και δεν έχουν την ποιότητα των αισθητήρων που βρίσκουμε στις ιατρικές συσκευές. Επιπλέον, το σύστημα χρειάζεται το κατάλληλο εύρος ζώνης (bandwidth) δεδομένου ότι οι ειδοποιήσεις συναγερμού μέσω email και η επίσκεψη στην ιστοσελίδα για την απομακρυσμένη παρακολούθηση των δεδομένων μέσω διαδικτύου εξαρτώνται από το σωστό εύρος ζώνης της σύνδεσης στο διαδίκτυο.

7.3 Θέματα για Περαιτέρω Διερεύνηση

Τα διδάγματα που αντλήθηκαν από την εκπόνηση αυτής της εργασίας υπογραμμίζουν διάφορους τομείς για περαιτέρω έρευνα.

Όσον αφορά τους αισθητήρες, παρόλη την πρόοδο που έχει επιτευχθεί στην εξέλιξή τους δεν έχουν φτάσει ακόμη το επίπεδο των αισθητήρων hospital-grade που χρησιμοποιούν οι νοσοκομειακές συσκευές. Ειδικά η βελτίωση στους

αισθητήρες του αναπνευστικού ρυθμού θα ήταν ανεκτίμητη στον τομέα της ιατρικής. Για το λόγο αυτό, πρέπει να γίνει περαιτέρω έρευνα για την βελτίωση της ποιότητας τους έτσι ώστε να καταστεί δυνατή η κατασκευή αισθητήρων υψηλής ακρίβειας, αξιοπιστίας και άνεσης όταν φοριούνται. Στην εργασία μας μελλοντικός στόχος θα μπορούσε να είναι η μείωση της επίδρασης της κίνησης στους αισθητήρες, ιδιαίτερα στους αισθητήρες αναπνευστικού ρυθμού και παλμών. Η προσθήκη επιπλέον αισθητήρων μέτρησης για την βελτίωση της αποτελεσματικότητας του συστήματος θα ήταν ένας δεύτερος στόχος. Έτσι, θα μπορούσαν να προστεθούν αισθητήρες παλμικής οξυμετρίας για την μέτρηση του κορεσμού οξυγόνου στο αίμα, αισθητήρες μέτρησης αρτηριακής πίεσης που ειδικά σε ασθενείς με καρδιαγγειακές νόσους θα μπορούσαν να φανούν πολύτιμοι, αισθητήρες μέτρησης σακχάρου στο αίμα, καθώς και αισθητήρες κίνησης για την ανίχνευση πτώσης ή της θέσης του ασθενή. Επιπρόσθετα, μπορούν να προστεθούν και αισθητήρες περιβάλλοντος για μετρήσεις θερμοκρασίας και υγρασίας του χώρου στον οποίο ζει ο ασθενής.

Όσον αφορά τις επικοινωνίες μια πρώτη βελτίωση που θα μπορούσε να γίνει θα ήταν η μείωση των καλωδιώσεων, γεγονός που θα επιφέρει επιπλέον άνεση στον ασθενή. Αυτό μπορεί να επιτευχθεί με τη χρήση ασύρματων αισθητήρων των οποίων το κόστος προς το παρόν είναι ιδιαίτερα υψηλό για χρήση σε τέτοιου είδους εφαρμογές. Τα ασύρματα δίκτυα που χρησιμοποιούνται, στην περίπτωση μας ένα WI-Fi οικιακό δίκτυο, θα μπορούσαν να αποτελέσουν ένα ακόμη πεδίο βελτίωσης για τέτοια συστήματα. Έτσι, θα ήταν χρήσιμο να αναπτυχθούν φορητά συστήματα υγειονομικής περίθαλψης που εξαρτώνται από το αναδυόμενο πρότυπο NB-IoT (Wikipedia, 2019) Επειδή πρόκειται για ένα εξαιρετικά νέο πρότυπο, δεν υπάρχουν εργασίες για την εφαρμογή του σε ένα περιβάλλον υγειονομικής περίθαλψης παρά τα προφανή πλεονεκτήματα του σε αυτόν τον τομέα. Τα πλεονεκτήματα του σε σχέση με τα υφιστάμενα δίκτυα είναι:

- Η τεχνολογία NB-IoT προσφέρει χαμηλότερη κατανάλωση ενέργειας. Αυτό επιτρέπει μεγάλη ενεργειακή αυτονομία των συνδεδεμένων

αισθητήρων & συσκευών η οποία μπορεί να διαρκέσει έως και **10 χρόνια** με μία μόνο φόρτιση.

- Μπορεί να προσφέρει αξιόπιστη και ισχυρή κάλυψη ακόμα κι όταν οι συσκευές βρίσκονται σε υπόγεια ή κάτω από το έδαφος.

Τέλος, θα μπορούσαμε να τροποποιήσουμε το σύστημά μας για να του δώσουμε δυνατότητες φορητότητας. Αυτό θα μπορούσε να γίνει με την προσθήκη ενός 3G/4G module το οποίο θα αναλάμβανε την επικοινωνία μεταξύ του Arduino με τις υπηρεσίες cloud. Συγχρόνως η ανάπτυξη μιας Android εφαρμογής για τον έλεγχο και την διεπαφή του συστήματος θα προσέδιδε μεγάλη ευελιξία.

Η αποθήκευση δεδομένων με χρήση τεχνολογιών cloud έχει εξεταστεί εκτενώς, αλλά η επεξεργασία δεδομένων είναι ένας τομέας στον οποίο πρέπει να διεξαχθεί περαιτέρω έρευνα. Πρέπει να συνεχιστεί η ανάπτυξη cloud based αλγορίθμων οι οποίοι είναι σε θέση να επεξεργάζονται ακατέργαστα δεδομένα από σύνθετους αισθητήρες και να εξάγουν σημαντικές πληροφορίες σχετικά με την υγεία ενός ατόμου.

Η μηχανική μάθηση (Machine learning) είναι ένας άλλος κλάδος επεξεργασίας δεδομένων που θα ήταν εξαιρετικά πολύτιμος σε σενάρια υγειονομικής περίθαλψης. Εάν εφαρμοστεί στο υψηλής υπολογιστικής ισχύος περιβάλλον του cloud, θα μπορούσε να προσφέρει υπηρεσίες διάγνωσης για τους ασθενείς, να κάνει νέες ανακαλύψεις σχετικά με τις τάσεις των ασθενειών και να βοηθήσει στην ανάπτυξη μεθόδων θεραπείας. Παρ' όλα τα σαφή πλεονεκτήματα, η μηχανική μάθηση δεν έχει εξερευνηθεί ακόμη ευρέως για εφαρμογές υγειονομικής περίθαλψης, παρέχοντας έτσι μια σημαντική ερευνητική ευκαιρία.

Μια άλλη επέκταση του συστήματος θα μπορούσε να είναι η χρήση της σειριακής επικοινωνίας που διαθέτει το Arduino. Μια σειριακή έξοδος μπορεί να συνδεθεί σε ένα υπολογιστή έτσι ώστε να έχουμε τη δυνατότητα καταγραφής των δεδομένων στην περίπτωση που το σύστημα τεθεί offline. Με τον τρόπο αυτό ο γιατρός θα έχει τη δυνατότητα να ανατρέξει σε καταγραφές που σε διαφορετική περίπτωση θα είχαν χαθεί.

Τέλος, η αξιοπιστία του συστήματος μπορεί να βελτιωθεί περαιτέρω με την προσθήκη αυστηρών πρωτοκόλλων ασφαλείας, όπως η σάρωση δακτυλικών αποτυπωμάτων και η προστασία με κωδικό πρόσβασης.

Παράρτημα Α

Κώδικας Arduino Uno

```
#define USE_ARDUINO_INTERRUPTS true
#define DEBUG true
#define SSID "SVPA" // "SSID- WiFi name"
#define PASS "6942431740" // "password-WPA Key"
#define IP "184.106.153.149" // thingspeak.com ip address
#define SERIESRESISTOR 77800 // Η αντίσταση του NTC thermistor
// στους 23 βαθμούς κελσίου είναι 77,8ΚΩ
#include <SoftwareSerial.h> //
#include "Timer.h" // Βιβλιοθήκες απαραίτητες για τη λειτουργία
#include <PulseSensorPlayground.h> //
```

```
Timer timer;
```

```
PulseSensorPlayground pulseRate;
```

```
SoftwareSerial esp8266(10,11);
```

```
String msg = "GET /update?key=ZYRYVE09K5G61864";
```

```
// API key για την αποστολή δεδομένων σε ThingSpeak.com
```

```
//Variables
```

```
const int PulseWire = A0; // Το κίτρινο καλώδιο του PulseSensor
```

```
// συνδέεται στο ANALOG PIN 0
```

```
int Threshold = 550; // για τον αισθητήρα heart rate
```

```
//float myTemp;
```

```
int myBPM;
```

```
String BPM;
```

```
String temp;
```

```
String RR;
```

```
int error;
```

```
int emergency;
```

```
int analog_Temp;
```

```

float Voltage;
float tempC;
//Variables for RR
enum respStatus { // Δημιουργία μεταβλητής η οποία μπορεί να λάβει
                  // ένα σετ απο τιμές
    Inhaling, // https://playground.arduino.cc/Code/Enum
    Exhaling };
unsigned long prevRespTime = 0;
unsigned long RespTime = 0;
respStatus currentRespState = Exhaling;
float breathrate = 0;
int respRate = 0;

void setup()
{
  pinMode(2, OUTPUT);
  Serial.begin(9600);
  //esp8266.begin(9600);
  esp8266.begin(115200); // Η ταχύτητα του ESP ρυθμίζεται στα 115200
  pulseRate.analogInput(PulseWire);
  pulseRate.blinkOnPulse(2); // Το LED που συνδέουμε στο I/O pin 2
                             // του Arduino αναβοσβήνει με το
                             // ρυθμό της καρδιάς.
  pulseRate.setThreshold(Threshold);
  // Έλεγχος αν δημιουργήθηκε το αντικείμενο
  // "pulseSensor" και εάν ξεκίνησε "begin" να βλέπει το σήμα.
  if (pulseRate.begin()) {
    Serial.println("pulseRate Αντικείμενο !");
    //Αυτό τυπώνεται στο Serial Monitor μόνο μια φορά κατά την
    // εκκίνηση ή το reset του Arduino.

  }
  Serial.println("AT");
  esp8266.println("AT");
  delay(3000);
  if(esp8266.find("OK"))
  {
    connectWiFi();
  }
  timer.every(15000, getData); // thingspeak χρειάζεται 15 sec

```

```

        // χρόνοκαθυστέρηση μεταξύ των
        // ενημερώσεων

    timer.every(15000, updateValues);
}
void loop()
{
    emergency_button();
    respiratory();
    startingPoint: //label
        error=0;
    timer.update();
    //Επανάληψη αποστολής αν η μετάδοση δεν ολοκληρωθεί
    if (error==1)
    {
        goto startingPoint; //Πήγανε στο label "start"
    }
    //delay(4000);
}
void updateValues()
{
    String cmd = "AT+CIPSTART=\\"TCP\\",\\"";
    // Χρήση της εντολής AT+CIPSTART
    // για εγκαθίδρυση TCP επικοινωνίας
    cmd += IP;
    cmd += "\",80"; // Port 80
    Serial.println(cmd);
    esp8266.println(cmd);
    delay(2000);
    if(esp8266.find("Error")) // Η εκτέλεση της AT+CIPSTAR επιστρέφει
    {
        // OK(επιτυχή σύνδεσης) ή "error" στην
        // αντίθετη περίπτωση.

        return;
    }

    cmd = msg ;
    cmd += "&field1="; //field 1 για ρυθμό παλμού (BPM)
    cmd += BPM;
    cmd += "&field2="; //field 2 για θερμοκρασία
    cmd += temp;

```

```

cmd += "&field4="; //field 4 για το ρυθμό αναπνοής
cmd += RR;
cmd += "\r\n";
Serial.print("AT+CIPSEND=");
// Εντολή AT+CIPSEND
//https://github.com/espressif/ESP8266_AT/wiki/CIPSEND
esp8266.print("AT+CIPSEND=");
Serial.println(cmd.length());
esp8266.println(cmd.length());
if(esp8266.find(">")) // Μετά την εκτέλεση της εντολής
// επιστρέφεται ">" που δείχνει
// ότι το module είναι έτοιμο να δεχθεί δεδομένα με μήκος
// που υπολογισε απο την προηγούμενη εντολή (cmd.length()).
{
  Serial.print(cmd);
  esp8266.print(cmd);
}
else
{
  Serial.println("AT+CIPCLOSE");
  esp8266.println("AT+CIPCLOSE"); // Κλείνει τη σύνδεση TCP
                                     //Επανάληψη αποστολής...

  error=1;
}
}
boolean connectWiFi()
{
  Serial.println("AT+CWMODE=1");// Δήλωση του esp8266 σε station mode
  esp8266.println("AT+CWMODE=1");
  delay(2000);
  String cmd="AT+CWJAP=\""; // Σύνδεση στο Access Point
  cmd+=SSID;
  cmd+="\", \"";
  cmd+=PASS;
  cmd+="\" ";
  Serial.println(cmd);
  esp8266.println(cmd); // Αποστολή στο esp της μεταβλητής cmd
  delay(5000);
  if(esp8266.find("OK")) // Αν η απάντηση στο serial monitor είναι OK
                                     //η συνάρτηση επιστρέφει true αλλιώς false

```

```

{
  return true;
}
else
{
  return false;
}
}

void getData(){
  analog_Temp = analogRead(A1);
  Voltage = (analog_Temp / 1023.0) * 5000;
  // 5000 για να πάρουμε τιμή voltage σε millivots.
  tempC = Voltage * 0.1; // Μετατροπή της τάσης (voltage)
                        // σε τιμή θερμοκρασίας με βάση προδιαγραφές LM35

  Serial.println(tempC);
  int myBPM = pulseRate.getBeatsPerMinute();
  // Κλήση της συνάρτησης του αντικειμένου
  // pulseRate η οποία επιστρέφει BPM ως "int".

  delay(20);
  char outstr1[10]; // outstr1[10], outstr2[10] είναι οι
                  // πίνακες που αποθηκεύονται οι τιμες σε
                  // string format

  char outstr2[10];
  BPM = dtostrf(myBPM, 4, 1, outstr1);
  // Μετατροπή των μεταβλητών myBPM και tempC σε string
  temp = dtostrf(tempC, 4, 1, outstr2);
  // dtostrf() Συνάρτηση
  // http://www.hobbytronics.co.uk/arduino-float-vars
}

void emergency_button() { // Η συνάρτηση αυτή λειτουργεί με
                          // τον ίδιο τρόπο που περιγράψαμε

  emergency = digitalRead(8); // για τη συνάρτηση updateValues()
  if(emergency == HIGH){
    Serial.println(emergency);
  }
}

```

```

digitalWrite(2, HIGH); // ενεργοποίηση του LED στην θύρα 2 (HIGH
                        // είναι το επίπεδο της τάσης)
delay(2000);          // παύση για δύο δευτερόλεπτα
digitalWrite(2, LOW); // απενεργοποίηση του LED κάνοντας την τάση
                        // στην θύρα 2 LOW

delay(2000);

String cmd = "AT+CIPSTART=\"TCP\",\","; // εγκαθίδρυση TCP
                                           // επικοινωνίας

cmd += IP;
cmd += "\",80";
Serial.println(cmd);
esp8266.println(cmd);
delay(2000);
if(esp8266.find("Error"))
{
    return;
}
cmd = msg;
cmd += "&field3=";
cmd += emergency;
cmd += "\r\n";
Serial.print("AT+CIPSEND="); // αποστολή πληροφοριών στον server
esp8266.print("AT+CIPSEND=");
Serial.println(cmd.length());
esp8266.println(cmd.length());
if(esp8266.find(">"))
{
    Serial.print(cmd);
    esp8266.print(cmd);
}
else
{
    Serial.println("AT+CIPCLOSE"); // τερματισμός TCP επικοινωνίας
    esp8266.println("AT+CIPCLOSE");
    error=1;
}
}
}
}

```

```

// Μέτρηση Respiratory Rate
void respiratory(){
  int sensorValue = analogRead(A2);
  //sensorValue = (1023 / sensorValue) - 1;
  // sensorValue = SERIESRESISTOR / sensorValue;// Υπολογισμός τιμής της αντίστασης του
  thermistor
  //Serial.print("Thermistor resistance ");
  //Serial.println(sensorValue);

  //delay(2000);
  //}
  if ((sensorValue > 400) && (currentRespState == Exhaling))
  // 400 είναι τιμή που διαβάζει ο A/D μετατροπέας στην είσοδο A2
  {
    currentRespState = Inhaling;
    prevRespTime = RespTime; // Αποθήκευση τιμής χρόνου του
                                // τελευταίου κύκλου της αναπνοής
    RespTime = millis();

    float period_respiration = RespTime - prevRespTime;

    if (period_respiration == 0){
      breathrate = 0;
      Serial.println(breathrate);

    }
    else {
      breathrate = 60000 / period_respiration;
  // αναγωγή του ρυθμού αναπνοής στο λεπτό
  /*Serial.print("Analog reading ");    // Ενεργοποίηση για να
                                          // βλέπουμε τις τιμές
      Serial.println(sensorValue);    // στο Serial Monitor
      Serial.print("Inhaling /tResp Rate: ");
      Serial.println(breathrate);
      Serial.print("Current Tme:");
      Serial.println(RespTime);
      Serial.print("averange:");
      Serial.println(respRate);
    }
  }

```

```

*/
if (respRate == 0)
    respRate = breathrate;
else {
    respRate = (int) ((0.2 * breathrate) + (0.8 * respRate));
    //https://helpful.knobs-dials.com/index.php/Low-pass_filter
    //https://www.norwegiancreations.com/2016/03/arduino-tutorial-simple-high-pass-band-
    pass-and-band-stop-filtering/

    //Serial.print("/tAvg Resp Rate: ");
    //Serial.println(respRate);
    char outstr3[10]; //
    RR = dtostrf(respRate, 4, 1, outstr3); // Μετατροπή της μεταβλητής respRate σε string
    }
}
}
else if ((currentRespState == Inhaling) && (sensorValue < 100))
{
    currentRespState = Exhaling;
}
}

```


Βιβλιογραφία

A.H. Mohamad, et al., A. (2019). Thingspeak Cloud Computing Platform Based ECG Diagnose System. *International Journal of Computing and Digital Systems*, 8(1), pp.11-18.

Alam, M., Hussain, M. and Amin, A. (2019). A Novel Design of a Respiratory Rate Monitoring System using a Push Switch Circuit and Arduino Micocontroller. [International Conference on Robotics, Electrical and Signal Processing Techniques](#). Dhaka, Bangladesh: IEEE, pp.470-473. *Anaesthesia*, 58(8), pp. 797-802.

Azmal, G. M., Al-Jumaily, A. και Al-Jaafreh, M. (2006). Continuous Measurement of Oxygen Saturation Level using Photoplethysmography Signal. *Proceedings of the 2006 International Conference on Biomedical and Pharmaceutical Engineering*. University of Technology Sydney, 2006-12-01. Singapore.

Baker, S., Xiang, W. and Atkinson, I. (2017). Internet of Things for Smart Healthcare: Technologies, Challenges, and Opportunities. *IEEE Access*, 5, pp.26521-26544.

Bayo-Monton, J., Martinez-Millana, A., Han, W., Fernandez-Llatas, C., Sun, Y. and Traver, V. (2018). Wearable Sensors Integrated with Internet of Things for Advancing eHealth Care. *Sensors*, 18(6), p.1851.

Bennett, J., Rokas, O. και Chen, L. (2017). Healthcare in the Smart Home: A Study of Past, Present and Future. *Sustainability*, 9(5), p.840.

Bhaskar, L. and Manage, P. (2017). IOT based Patient Health Monitoring System using Raspberry Pi 3. *International Research Journal of Engineering and Technology (IRJET)*. 4(7), pp. 2593-2597

Boron, W. και Boulpaep, E. (2017). *Medical physiology*. Philadelphia: Elsevier.

Clifford, G.D. (2008). *Biomedical Signal and Image Processing*. Spring. Κεφάλαιο 15.

Cruz, J., Brooks, D. and Marques, A. (2014). Home telemonitoring in COPD: A systematic review of methodologies and patients' adherence. *International Journal of Medical Informatics*, 83(4), pp.249-263.

Deekshith Gupta, M.S., Patchava, V. and Menezes, V. (2015). Healthcare based on IoT using Raspberry Pi. *International Conference on Green Computing and Internet of Things (ICGCloT)*. Noida, India: IEEE. Pp.796-799

Dey, A. (2001). Understanding and Using Context. *Personal and Ubiquitous Computing*, 5(1), pp.4-7.

Duregger, K., Hayn, D., Morak, J., Ladenstein, R. και Schreier, G. (2014). An mHealth system for toxicity monitoring of paediatric oncological patients using Near Field Communication technology. *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Milan: IEEE, pp. 6848-6851.

Fernandez, F. και Pallis, G.C. (2014). *Opportunities and challenges of the Internet of Things for healthcare Systems engineering perspective*. International Conference on Wireless Mobile Communication and Healthcare (Mobihealth). Athens: IEEE.

Fezari, M. and Dahoud, A. (2018). *Integrated Development Environment "IDE" For Arduino*. Al Zaytoonah University, Jordan. Διαθέσιμο σε <https://www.researchgate.net> [Πρόσβαση: 12 Σεπ. 2019].

Fieselmann, J. F., Hendryx, M.S., Helms, C.M, και Wakefield, D.S. (1993). Respiratory rate predicts cardiopulmonary arrest for internal medicine inpatients. *Journal of General Internal Medicine*, 8(7), pp. 354–360.

Ganong, W. (2005). *Review of medical physiology*. New York: McGraw-Hill Medical.

Goljan, E. (2011). *Pathology*. Philadelphia, PA: Mosby/Elsevier.

Gómez, J., Oviedo, B. and Zhuma, E. (2016). Patient Monitoring System Based on Internet of Things. *Procedia Computer Science*, 83, pp.90-97.

Hall, J. και Guyton, A. (2011). *Textbook of medical physiology*. Philadelphia, Pa.: Saunders.

Karki, J. (1998). *Mixed Signal and Analog Operational Amplifiers*. Διαθέσιμο σε <http://www.ti.com/lit/an/sloa011/sloa011.pdf> [Πρόσβαση: 12 Αυγ 2019].

Koufi, D. Srouji, K. Aldukka, A. Saltaji, M.M. Παπασταθοπούλου, A. (2018). *Sleep Apnea Monitoring and diagnosis system*. Dep. Biomedical Engineering. Near East University. Κύπρος.

Li, Z. και Huang, A. (2014). Fall perception for elderly care: A fall detection algorithm in SmartWristlet mHealth system. *In Proceedings of the 2014 IEEE International Conference on Communications (ICC)*. Sydney, Australia: IEEE, pp. 4270–4274.

Liu, L., Stroulia, E., Nikolaidis, I., Miguel-Cruz, A. and Rios Rincon, A. (2016). Smart homes and home health monitoring technologies for older adults: A systematic review. *International Journal of Medical Informatics*, 91, pp.44-59.

Massaroni, C., Nicolò, A., Lo Presti, D., Sacchetti, M., Silvestri, S. and Schena, E. (2019). Contact-Based Methods for Measuring Respiratory Rate. *Sensors*, 19(4), p.908.

Massaroni, C., Nicolò, A., Lo Presti, D., Sacchetti, M., Silvestri, S. and Schena, E. (2019). Contact-Based Methods for Measuring Respiratory Rate. *Sensors*, [online] 19(4), p.908. Διαθέσιμο σε <https://www.mdpi.com/1424-8220/19/4/908/pdf> [Πρόσβαση: 05 Αύγ. 2019].

Mohan Ra, K. (2015). DESIGN AND DEVELOPMENT OF REAL TIME RESPIRATORY RATE MONITOR USING NON-INVASIVE BIOSENSOR. *International Journal of Research in Engineering and Technology*, 04(06), pp.437-442.

Mshali, H., Lemlouma, T., Moloney, M. and Magoni, D. (2018). A survey on health monitoring systems for health smart homes. *International Journal of Industrial Ergonomics*, 66, pp.26-56.

Opie, L. (2004). *Heart physiology*. Philadelphia: Lippincott Williams & Wilkins.

Paré, G., Moqadem, K., Pineau, G. και St-Hilaire, C. (2010). Clinical effects of home telemonitoring in the context of diabetes, asthma, heart failure and hypertension: a systematic review. *J Med Internet Research*. Vol. 12(21).

Parihar, V.R., Tonge, A.Y. and Ganorkar, P.D. (2017). Heartbeat and Temperature Monitoring System for Remote Patients using Arduino. *International Journal of Advanced Engineering Research and Science*, 4(5), pp.55-58.

Rahman, A., Rahman, T., Ghani, N.H., Hossain, M.S., & Uddin, J. (2019). IoT Based Patient Monitoring System Using ECG Sensor. *2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, 378-382.

Rahmani, A., Gia, T., Negash, B., Anzanpour, A., Azimi, I., Jiang, M. and Liljeberg, P. (2018). Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach. *Future Generation Computer Systems*, 78, pp.641-658.

Rajani Kumari, L.V., Padma Sai, Y. and Balaji, N. (2016). Arduino Based Abnormal Heart Rate Detection and Wireless Communication. *International Journal on Cybernetics & Informatics*, 5(4), pp.47-53.

Ravi (2018). *ESP8266 AT Commands (AT Instruction Set) - Electronics Hub*. [online] Electronics Hub. Διαθέσιμο σε <https://www.electronicshub.org> [Πρόσβαση: 1 Σεπ 2019].

Recktenwald, G. (2013). *Temperature Measurement with a Thermistor and an Arduino*. Portland State University, Mechanical and Materials Engineering

Sailunaz, K., Alhussein, M., Shahiduzzaman, M., Anowar, F. and Mamun, K. (2016). CMED: Cloud based medical system framework for rural health monitoring in developing countries. *Computers & Electrical Engineering*, 53, pp.469-481.

Smirek, L., Zimmermann, G. and Beigl, M. (2016). Just a Smart Home or Your Smart Home – A Framework for Personalized User Interfaces Based on Eclipse Smart Home and Universal Remote Console. *Procedia Computer Science*, 98, pp.107-116.

Spano, E., Di Pascoli, S. and Iannaccone, G. (2016). Low-Power Wearable ECG Monitoring System for Multiple-Patient Remote Monitoring. *IEEE Sensors Journal*, 16(13), pp.5452-5462.

Storck, K., Karlsson, M., Ask, P. and Loyd, D. (1996). Heat transfer evaluation of the nasal thermistor technique. *IEEE Transactions on Biomedical Engineering*, 43(12), pp.1187-1191.

Subbe, C., Davies, R., Williams, E., Rutherford, P. και Gemmell, L. (2003). Effect of introducing the Modified Early Warning score on clinical outcomes, cardio-pulmonary arrests and intensive care utilisation in acute medical admissions.

Tabish, R., Ghaleb, A.M., Hussein, R., Touati, F. et al. (2014). A 3G/WiFi-enabled 6LoWPAN-based U-healthcare system for ubiquitous real-time monitoring and data logging. *Middle East Conference in Biomedical Engineering (MECBME)*. Doha: IEEE, pp. 277-280.

Ullah, K., Shah, M.A. and Zhang, S. (2016). Effective ways to use Internet of Things in the field of medical and smart health care. *International Conference on Intelligent Systems Engineering (ICISE)*. Islamabad: IEEE, pp. 372-379.

Wright, K., Hull, J. και Czeisler, C. (2002). Relationship between alertness, performance, and body temperature in humans. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 283(6), pp. R1370-R1377.

Zanjal, S.V. and Talmale, G.R. (2016). Medicine Reminder and Monitoring System for Secure Health Using IOT. *Procedia Computer Science*, 78, pp.471–476.

Arduino Board, Product Overview (2016). Διαθέσιμο σε: <https://www.arduino.cc> [Πρόσβαση: 15 Οκτ. 2019].

Arduino.cc. (2018). *Arduino Playground - Enum Resource*. [online] Διαθέσιμο σε <https://playground.arduino.cc/Code/Enum> [Πρόσβαση 4 Όκτ. 2019].

Arduino.cc. (2019). *Arduino-SoftwareSerial*. [online] Διαθέσιμο σε <https://www.arduino.cc> [Πρόσβαση: 12 Οκτ 2019].

ClickSend. (2019). *Bulk SMS Gateway & Email Marketing Platform – Email & SMS Marketing*. [online] Διαθέσιμο σε: <https://www.clicksend.com/eu/> [Πρόσβαση: 26 Σεπ 2019].

ESP8266 AT Commands. Διαθέσιμο σε <https://www.electronicshub.org> [Πρόσβαση: 23 Αυγ. 2019].

Hobbytronics.co.uk. (2019). *Displaying FLOAT variables in Arduino*. [online] Διαθέσιμο σε <http://www.hobbytronics.co.uk> [Πρόσβαση: 16 Σεπ 2019].

How To Bias An Op-Amp. Διαθέσιμο σε <http://fab.cba.mit.edu> [Πρόσβαση: 12 Σεπ. 2019].

https://en.wikipedia.org/wiki/Microprocessor_development_board
[Πρόσβαση: 11 Νοε 2019].

Knobs-dials.com. (2019). *Low-pass filter - Helpful*. [online] Διαθέσιμο σε <https://helpful.knobs-dials.com> [Πρόσβαση: 19 Σεπ. 2019].

LM 358 Op Amp. Διαθέσιμο σε <http://www.electronics123.net> [Πρόσβαση: 13 Σεπ. 2019].

LM35 Datasheet. Διαθέσιμο σε <http://www.ti.com/lit/ds/symlink/lm35.pdf> [Πρόσβαση: 11 Αύγ. 2019].

M2MSupport.net. (2013). *AT+CIPSTART – Start up TCP or UDP Connection*. [online] Διαθέσιμο σε <https://m2msupport.net/m2msupport/atcipstart-start-up-tcp-or-udp-connection/> [Πρόσβαση 12 Οκτ. 2019].

Making sense of Internet of Things platforms <https://www.mckinsey.com> [Πρόσβαση: 13 Αυγ 2019].

Mathworks.com. (2014). [online] Διαθέσιμο σε <https://www.mathworks.com> [Πρόσβαση: 13 Οκτ. 2019].

Microprocessor development board. 31 Οκτ 2019 [Άρθρο Wiki]. Διαθέσιμο σε National Institute of Standards and Technology (NIST) (2011). *Final Version of NIST Cloud Computing Definition Published*. Διαθέσιμο σε <https://www.nist.gov>. [Πρόσβαση: 23 Αυγ 2019].

OP Amplifiers. /Διαθέσιμο σε <https://www.generationrobots.com> [Πρόσβαση: 21 Σεπ. 2019].

OpenURC Alliance. OpenURC, <http://www.openurc.org>. [Πρόσβαση: 21 Σεπ. 2019]

Photoplethysmogram. 5 Σεπ 2019. [Άρθρο Wiki]. Διαθέσιμο σε <https://en.wikipedia.org> [Πρόσβαση 22 Σεπτεμβρίου 2019].

Post (HTTP). 9 Σεπ 2019. [Άρθρο Wiki]. Διαθέσιμο σε <https://en.wikipedia.org> [Πρόσβαση: 03 Οκτ 2019].

Respiratory rate. 29 Αυγ 2018. [Άρθρο Wiki]. Διαθέσιμο σε <https://en.wikipedia.org>. [Πρόσβαση: 29 Σεπ 2019].

Sparkfun.com. (2019). *Pull-up Resistors - learn.sparkfun.com*. [online] Διαθέσιμο σε <https://learn.sparkfun.com/> [Πρόσβαση: 24 Αυγ. 2019].

ThingSpeak IoT Community. (2012). *React*. [online] Διαθέσιμο σε <https://community.thingspeak.com> [Πρόσβαση: 13 Οκτ 2019].

Thingspeak.com. (2019). *Learn More - ThingSpeak IoT*. [online] Διαθέσιμο σε https://thingspeak.com/pages/learn_more [Πρόσβαση: 02 Σεπ. 2019].

Virtuino App. Διαθέσιμο σε <http://virtuino.com/index.php> [Πρόσβαση: 22 Οκτ. 2019].

What is applet? - Definition from WhatIs.com (2017). [online] WhatIs.com. Διαθέσιμο σε <https://whatis.techtarget.com> [Πρόσβαση: 4 Οκτ. 2019].

What is RESTful API? - Definition from WhatIs.com (2019). [online] SearchAppArchitecture. Διαθέσιμο σε <https://searchapparchitecture>. [Πρόσβαση: 11 Οκτ. 2019].

Wikipedia (2019). *IFTTT*. [Άρθρο Wiki]. Διαθέσιμο σε : <https://en.wikipedia.org> [Πρόσβαση: 12 Οκτ 2019].

Wikipedia (2019). *Photoplethysmogram*. [Άρθρο Wiki]. Διαθέσιμο σε <https://en.wikipedia.org/wiki/Photoplethysmogram> [Πρόσβαση: 22 Αυγ. 2019].

Wikipedia Contributors (2019b). *Narrowband IoT*. [Άρθρο Wiki] Διαθέσιμο σε <https://en.wikipedia.org> [Accessed 24 Nov. 2019].

Αρπατζάνης, Ν. (χ.η). *Εισαγωγή στο MATLAB. Εργαστηριακές Σημειώσεις*. Τ.Ε.Ι. Κ. Μακεδονίας. Διαθέσιμο σε <https://elearning.teicm.gr/> [Πρόσβαση: 22 Αυγ 2019].

Βαρδάκας, Μ. και Κοσμαδάκης, Ι-Μ. (2017). *Μελέτη και υλοποίηση ενσωματωμένου συστήματος αισθητήρων επιτήρησης βιοϊατρικών παραμέτρων*

για *e-Health monitoring εφαρμογές*, Τμήμα Ηλεκτρολογων Μηχανικων Και Τεχνολογιας Υπολογιστων, Πανεπιστήμιο Πατρών.

Παπασταθοπούλου, Α. (2017). *Internet of Things*. Τμήμα Εφαρμοσμένης Πληροφορικής. Πανεπιστήμιο Μακεδονίας, Ελλάδα.

Χατζηευθυμίου, Σ. (2015). *Μελέτη Του Διαδικτύου Των Πραγμάτων: Αρχιτεκτονική, Εφαρμογές Και Μελλοντικό Όραμα*. Τμήμα Φυσικής, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης.

4^η Υ.Π.Ε. Μακεδονίας και Θράκης. (2014) *Λήψη Ζωτικών Σημείων σε Ενήλικες*. Θεσσαλονίκη: Υπουργείο Υγείας. Ανακτήθηκε από: [Lipsi zwtikwn simeiwv se enilikes.pdf](#) [Πρόσβαση 20 Σεπτεμβρίου 2019].

Διεπαφή προγραμματισμού εφαρμογών. 25 Μαΐου 2017. [Άρθρο Wiki]. Διαθέσιμο σε <https://el.wikipedia.org/wiki/> [Πρόσβαση: 12 Αυγ 2019].

Εθνικό Πληροφοριακό Σύστημα Έρευνας και Τεχνολογίας (ΕΠΣΕΤ) (χ.η.). *Υπολογιστικό Νέφος (Cloud Computing)*. Διαθέσιμο σε <http://www.epset.gr> [Πρόσβαση: 11 Ιουλ 2019].

Εισαγωγή στο πρωτόκολλο HTTP <https://www.dit.uoi.gr> [Πρόσβαση: 22 Αυγ 2019].

Μετατροπή αναλογικών και Ψηφιακών Σημάτων. Διαθέσιμο σε <http://meleththrio.teicm.gr> [Πρόσβαση: 10 Οκτ 2019].

Μικροελεγκτής. 23 Ιουλ 2019. [Άρθρο Wiki]. Διαθέσιμο σε <https://el.wikipedia.org/wiki/Μικροελεγκτής> [Πρόσβαση: 12 Οκτ 2019].