ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΙΑΤΡΙΚΗΣ

ΣΥΣΤΗΜΑ ΥΠΟΒΟΗΘΗΣΗΣ ΓΙΑ ΑΤΟΜΑ ΠΟΥ ΠΑΣΧΟΥΝ ΑΠΟ ΑΛΤΣΧΑΙΜΕΡ

ΠΑΓΩΝΗΣ ΝΙΚΟΛΑΟΣ
Αριθμός Μητρώου:13073

<u>Επιβλέπων Καθηγητής</u>
Παντελεήμων Ασβεστάς,
Αναπληρωτής Καθηγητής

Αιγάλεω Ιούνιος 2022

Η Τριμελής Εξεταστική Επιτροπή

Ο Επιβλέπων Καθηγητής

Παντελεήμων Ασβεστάς	Δημήτριος Γκλώτσος	Σπυρίδων Κωστόπουλος
Αναπληρωτής Καθηγητής	Αναπληρωτής Καθηγητής	Αναπληρωτής Καθηγητής
[ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ]	[ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ]	[ΨΗΦΙΑΚΗ ΥΠΟΓΡΑΦΗ]
Digitally signed by PANTELEIMON ASVESTAS Date: 2022.07.20 10:29:55 +03'00'	Digitally signed by Dimitrios Gklotsos Date: 2022.07.20 13:50:21 +03'00'	Digitally signed by Spyridon Kostopoulos

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ/ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/η υπογράφων/ουσα Νικόλαος Παρώνης του Δημητρίου με αριθμό μητρώου 13073 φοιτητής/τρια του Τμήματος Μηχ. ΒΙΟΙΑΤΡΙΚΗΣ της Σχολής ΒΙΟΙΑΤΡΙΚΗΣ του Πανεπιστημίου Δυτικής Αττικής, δηλώνω υπεύθυνα ότι:

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

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

Ημερομηνία
20/7/2022

Ο/Η Δηλών/ούσα
Νικόλαος Παρώνης

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή Παντελεήμων Ασβεστά, για την συνεχή βοήθεια, στήριξη και ανταπόκρισή του, όταν αυτή ήταν απαραίτητη.

Περίληψη

Ο σκοπός της συγκεκριμένης εργασίας, είναι η δημιουργία μιας βοηθητικής συσκευής για άτομα που πάσχουν από τη νόσο Alzheimer ή αλλιώς Αλτσχάιμερ σε ελληνική μετάφραση, με κύριο στόχο την επιτήρησή τους. Η συσκευή αυτή θα αποτελεί τόσο ένα σύστημα παρακολούθησης του χώρου όσο και του ιδίου ασθενή, το οποίο θα απευθύνεται κυρίως σε υπερήλικες ενώ παράλληλα μέσω του συνδυασμού Arduino (μικροελεγκτής μονής πλακέτας) και μιας λεγόμενης IoT platform (δίκτυο που επιτρέπει την σύνδεση και την ανταλλαγή δεδομένων μεταξύ πολλών αντικειμένων) εν ονόματι Node - Red, θα παρακολουθεί τους ασθενείς και θα ενημερώνει με κατάλληλο τρόπο σε περίπτωση που παρουστιαστεί ανάγκη για βοήθεια. Δέκτες αυτής της ενημέρωσης, θα είναι οι συγγενείς ή οποιοδοήποτε άτομο που είναι υπεύθυνο για την επίβλεψη του ασθενή, η οποία θα πραγματοποιείται μέσω e-mail. Η αποστολή αυτού του e-mail θα είναι στην κρίση των αισθητήρων από τους οποίους αποτελείται το Alzheimer’s Assistant και θα συμβαίνει κάθε φορά που θα παραβιάζονται τα προκαθορισμένα όριά τους. Για την υλοποίηση όμως, της συγκεκριμένης συσκευής, χρειάστηκε μια συγκεκριμένη μεθοδολογία. Αυτή η μεθοδολογία, αρχικά προαπαιτούσε την σωστή λειτουργία των αισθητήρων ξεχωριστά και στην συνέχεια όλων μαζί ως σύνολο και να ελέγχονται από ένα και μόνο σύστημα. Βεβαίως αυτό δεν θα ήταν δυνατό, χωρίς τον κατάλληλο κώδικα για το Arduino είτε την δυνατότητά σύνδεσης του στο διαδίκτυο, πράγματα τα οποία το καθιερώνουν ως τον διαμεσολαβητή της επικοινωνίας μεταξύ της Node - Red, των αισθητήρων και των επιτηρητών του ασθενή. Τέλος, με την επικοινωνία πλέον εξασφαλισμένη και προσθέτοντας τις απαραίτητες εντολές στην Node - Red, βρισκόμαστε ένα βήμα πιο κοντά στην έγκαιρη πρόληψη και αντιμετώπιση των αντιξοοτήτων, της νόσου του Αλτσχάιμερ.

Λέξεις Κλειδιά: ***Αλτσχάιμερ, Arduino, Διαδίκτυο των Πραγμάτων, Node - Red, Σύστημα παρακολούθησης, Αισθητήρες, Ηλεκτρονικό Ταχυδρομείο.***

Abstract

The purpose of this work is to create an auxiliary device for people suffering from Alzheimer's or else Αλτσχάιμερ in Greek, with the main aim of supervising them. This device is an area monitoring and patient tracking system that is primarily aimed at elderly people, but also through the combination of Arduino (a single-board microcontroller) and a so-called IoT platform (a network that allows connection and data exchange between multiple objects) in the name of Node - Red, will monitor the patients and will properly inform you in case of need. Receivers of this update will be relatives or otherwise the person supervising the patient, which will be by e-mail. The sending of this email will be at the discretion of the Alzheimer's Assistant's sensors and will occur whenever their predefined limits are violated. To implement this device, however, a particular methodology was needed. This methodology initially requires the proper functioning of the sensors individually and then all together as a whole, to be controlled by a single system. Of course, this would not be possible, without the proper code for Arduino and its ability to connect to the internet, which establishes it as the mediator of communication between Node-RED, the sensors and the patient's supervisors. Finally, with communication now secured and adding the necessary commands to Node-RED, we are one step closer to early prevention and treatment of adversities, of the Alzheimer's disease.

Key Words: ***Alzheimer, Arduino, IoT (Internet of Things), Node - Red, Monitoring System, Sensors, E-mail.***

ΠΕΡΙΕΧΟΜΕΝΑ

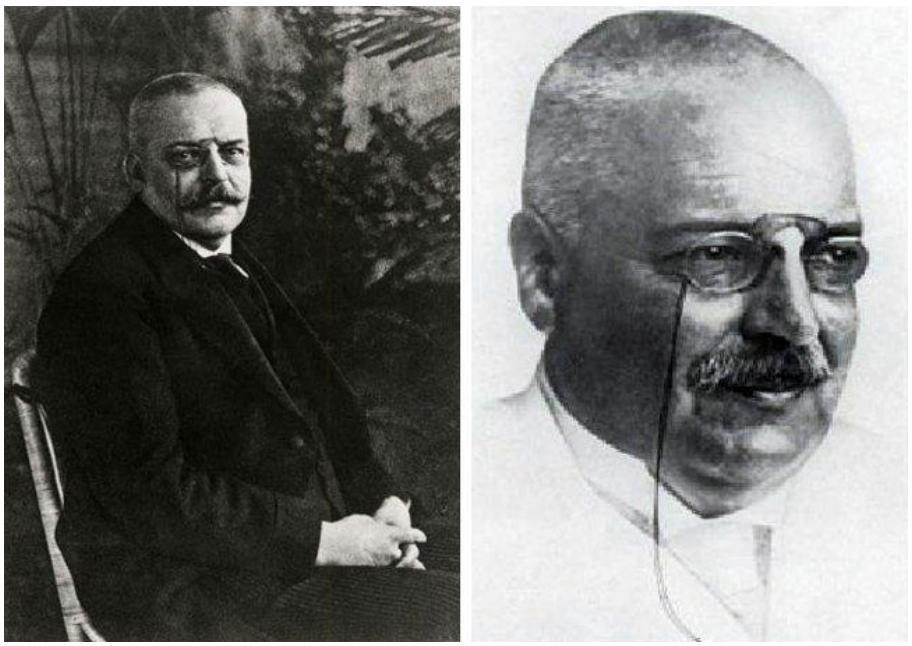
1. Εισαγωγή

Ξεκινώντας, είναι πολύ σημαντικό να γνωρίζουμε την ιστορική προέλευση μιας νόσου και το πως δημιουργήθηκε ή αλλιώς εμφανίστηκε, ώστε να μπορέσουμε ευκολότερα να κατανοήσουμε και αργότερα να ανακαλύψουμε τρόπους αντιμετώπισης της. Είναι απαραίτητο να γνωρίζουμε όσο το δυνατόν περισσότερα όσον αφορά τη συμπεριφορά της, τον τρόπο που και αν μεταδίδεται , τα συμπτώματα και τις παρενέργειες που αν τυχόν προκαλεί στον άνθρωπο αλλά και τους διαθέσιμους τρόπους αντιμετώπισής της, είτε αυτό αφορά κάποια φαρμακευτική αγωγή ή οποιοδήποτε άλλο τρόπο. Τέλος, εξίσου σημαντική είναι και η γνώση της πιθανότητας που έχει η πλειοψηφία του πληθυσμού, να προσβληθεί από αυτήν. Παρακάτω αναλύονται όχι μόνο όλα αυτά, αλλά και οι επιδιώξεις μου όσων αφορά την αντιμετώπιση της ασθένειας του Αλτσχάιμερ.

1.1 Ιστορική αναφορά στην νόσο Αλτσχάιμερ

Αρχικά η άνοια πριν καν ονομαστεί με αυτόν τον όρο προυπήρχε στην ιστορία ακόμα και την εποχή των Αιγύπτιων σε χρονολογία 2000 π.Χ., οι οποίοι γνώριζαν πως καθώς ο άνθρωπος γερνάει η μνήμη του εξασεθνεί. Ακόμα και αρχαίοι Έλληνες όπως ο Πυθαγόρας, ο Ιπποκράτης και ο Πλάτωνας πίστευαν πως η γήρανση έχει ως αποτέλεσμα τη φθίνουσα πορείας του ανθρώπινου μυαλού και σώματος. Ο Τόμας Γουίλις γιατρός της εποχής του 1600 επινόησε τη νευρολογία ως όρο ενώ το 1797 ο Φιλίπ Πινέλ κατέστησε την άνοια ως ιατρικό αποδεκτό όρο. Η νόσος του Αλτσχάιμερ περιγράφηκε πρώτη φορά από τον Γερμανό ψυχίατρο και νευροπαθολόγο Αλοΐσιο Αλτσχάιμερ το 1906. Νωρίτερα της ίδιας δεκαετίας και πιο συγκεκριμένα, το 1901, ο Αλοΐσιος στην ηλικία των 30 ετών εργαζόταν ως ψυχίατρος και πιο συγκεκριμμένα ως κλινικός ιατρός σε ένα άσυλο φρενοβλαβών στη Φρανκφούρτη και είχε ως στόχο να καταναοήσει πως αλληλοσυνδέονται μια νόσος του εγκεφάλου με τη ψυχική υγεία του ανθρώπου. Τότε γνώρισε την ασθενή Αουγκούστε Ντέτερ (Auguste Deter) η οποία σε ηλικία 51 ετών σύμφωνα με τα λεγόμενα του συζύγου της παρουσίαζε παράξενα συμπτώματα στη συμπεριφορά όπως φόβο, παράνοια, επιθετικότητα ενώ ταυτόχρονα είχε και μερική απώλεια μνήμης. Όταν εκείνη πέθανε το 1906, έχοντας θέση στελέχου σε ερευνητικό κέντρο στο Μόναχο, είχε την ιδέα να στείλει τον εγκέφαλο της για έρευνα, στον τότε γνωστό ψυχίατρο της εποχής και διευθυντή του ερευνητικού Εμίλ Κρέπελιν. Εκείνος μαζί με δύο Ιταλούς παθολόγους, χρησιμοποιώντας το μικροσκόπιο και τεχνικές χρώσης με τεχνολογία της εποχής διαπίστωσαν την ύπαρξη αυτών που εννούμε στη σημερινή εποχή αμυλοειδών πλακών, οι οποίες ευθύνονται για την καταστροφή εγκεφαλικών κυττάρων. Στις 3 Νοεμβρίου του 1906 παρουσίασε για πρώτη φορά στη βασιλική κλινική του πανεπιστημίου του Μονάχου τη παθολογία και τα συμπτώματα της νόσου Αλτσχάιμερ. Πλέον η νόσος αποτελεί ένα νευροεκφυλιστικό είδος άνοιας. Με τον όρο νευροεκφυλιστική , εννοείται μια κατηγορία παθήσεων οι οποίες στοχεύουν τους νευρώνες του ανθρώπινου εγκεφάλου. Με τον όρο άνοια περιγράφεται ένα σύνολο συμπτωμάτων που προκαλούνται από διαταραχές που επηρεάζουν τον εγκέφαλο με αποτέλεσμα ο ασθενής να έχει προβλήματα στην σκέψη, την μνήμη, την ομιλία, τη συμπεριφορά και γενικότερα στην εκτέλεση των καθημερινών του δραστηριοτήτων.

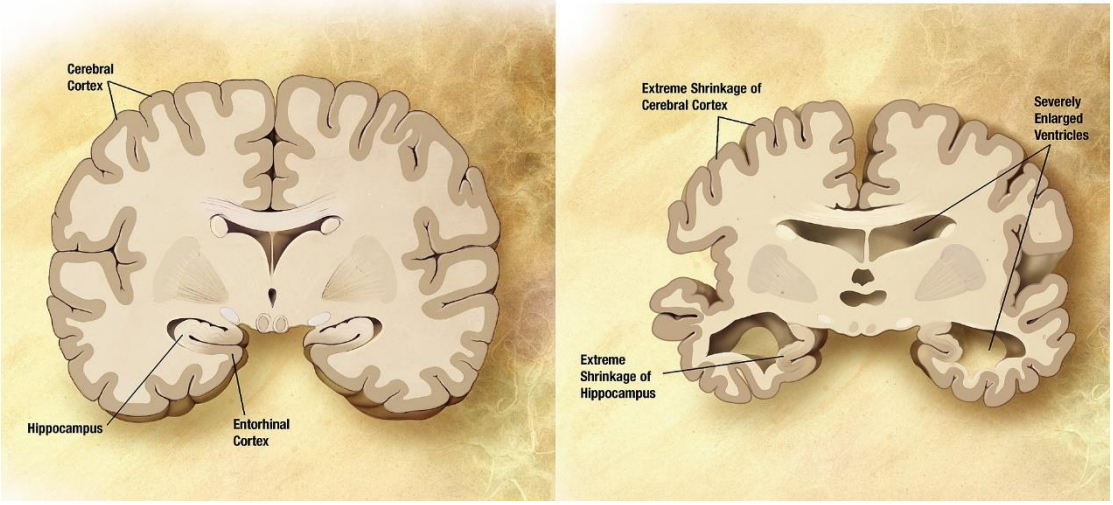
Πιο συγκεκριμένα η νόσος Αλτσχάιμερ στα πρώιμα στάδια της εκδηλώνεται με αργούς ρυθμούς, επηρεάζοντας κέντρα του νευρικού συστήματος, αλλά στη συνέχεια εξελίσσεται γρήγορα έχοντας ως τελικό αποτέλεσμα την καταστροφή των νευρικών κυττάρων και την ατροφία του εγκεφάλου. Κυρίως εμφανίζεται σε άτομα της τρίτης ηλικίας γύρω στα 65 έτη και ακόμα σε πιο σπάνιες περιπτώσεις μπορεί να εμφανιστεί πριν τα 50. Η ασθένεια αυτή αποτελεί μεγάλο κοινωνικό πρόβλημα τόσο για τον ασθενή του οποίου η ποιότητα ζωής υποβαθμίζεται, απομακρύνοντας τον από τους ρυθμούς τις καθημερινότητας του, αλλά την οικογένεια και τους συγγενείς του οι οποίοι φορτίζονται συναισθηματικά. Προς το παρών δεν έχει βρεθεί κάποια θεραπεία που να αντιμετωπίζει τις διαταραχές που συμβαίνουν στον εγκέφαλο. Οι θεραπείες που εφαρμόζονται είναι κυρίως φαρμακευτική αγωγή και η παροχή φροντίδας. Τα σύγχρονα φάρμακα για τη νόσο δεν μπορούν να σταματήσουν ή να αναστείλουν τη διαδικασία εξέλιξης της, αλλά μπορούν να ελαττώσουν τα συμπτώματα ενώ η παροχή φροντίδας καθιστά την καθημερινότητα του ασθενούς όσο το δυνατόν πιο εύκολη.



Εικόνα 1.1.1: Αλόϊσος Αλτσχάιμερ (14 Ιουνίου 1864 – 19 Δεκεμβρίου 1915).
Προσαρμόστηκε από: (Berrios, 1990)

1.2 Τα στάδια της νόσου Αλτσχάιμερ και τα συμπτώματα

Όπως αναφέρθηκε η συγκεκριμένη νόσος αρχικά εκδηλώνεται σε ήπια μορφή σε αντίθεση με τη συνέχεια όπου τα συμπτώματα εξελίσσονται ραγδαία. Η κατηγοριοπίηση των σταδίων της εξέλιξης της νόσου γίνεται είτε με τη μορφή επτά σταδίων είτε τριών φάσεων τα οποία θα αναφερθούν παρακάτω :



Εικόνα 1.2.1: Αριστερά η δομή του εγκεφάλου πριν την νόσο. Δεξιά η δομή του εγκεφάλου μετά τη νόσο. Προσαρμόστηκε από: (Garrondo, 2008)

Πρώιμο στάδιο (Mild Alzheimer's Disease)

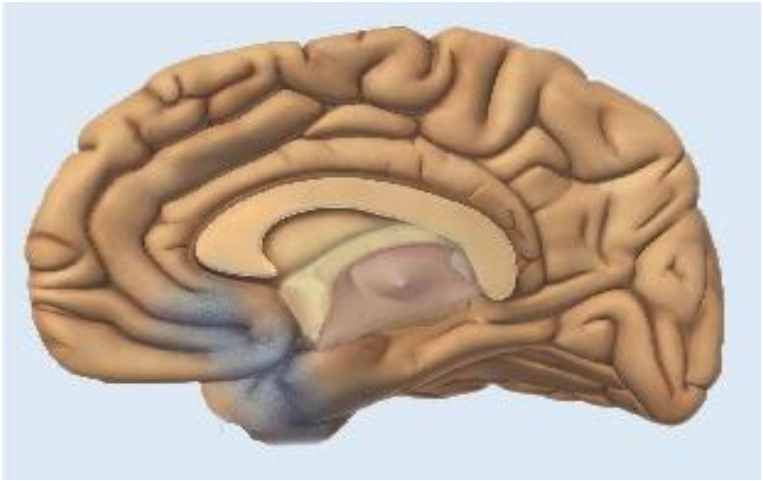
Φάση 1: Η νόσος του Αλτσχάιμερ αρχικά εκδηλώνεται πολύ ήπια, επηρεάζοντας μικρές περιοχές του εγκεφάλου χωρίς να παρατηρηθεί κάποια διαφορά στη συμπεριφορά του προσβαλόμενου ατόμου είτε ιδιαίτερα συμπτώματα. Ο μόνος τρόπος για να διαπιστωθεί αν κάποιος πάσχει από τη νόσο είναι μέσω τομογραφίας εκπομπής ποζιτρονίων ή αλλιώς PET scan.

Φάση 2: Ακόμα και σε αυτή τη περίπτωση οι αλλαγές στο τρόπο ζωής ενός ανθρώπου μπορεί να είναι αναιπαίσθητες ώστε να δυσκολεύεται να τα παραρήσει ο ίδιος ο συγγενής είτε κάποιος γιατρός. Συμπτώματα όπως να ξεχνιούνται λέξεις είτε να τοποθετούνται σε λάθος θέση αντικείμενα μπορεί να είναι και απλά σημάδια γήρανσης.

Φάση 3: Στη συγκεριμένη φάση παρατηρείται μερική απώλεια μνήμης κυρίως των πιο πρόσφατων γεγονότων. Επίσης μπορεί να ξεχνάει λόγια και ονόματα όταν μιλάει, είτε να εμφανίζει απόσπαση της προσοχής του αλλά είναι δυνατόν να πραγματοποιεί τις καθημερινές του δραστηριότητες όπως να εργάζεται και να ζει αυτόνομα. Στο τέλος του πρώτου σταδίου μπορεί να χαθεί σε άγνωστα μέρη, να δυσχεραίνεται περισσότερο η εύρεση των σωστών λέξεων είτε ακόμα να αποφεύγει κοινωνικές εκδηλώσεις. Συνήθως σε αυτό το στάδιο ο ασθενής είναι σε θέση να κατανοήσει και να αντιληφθεί πως του συμβαίνει κάτι και το αντιμετωπίζει με τρόπους όπως γράφοντας σημειώσεις για υπενθύμιση. Επιπλέον στοιχεία που εμφανίζονται στον ασθενή είναι:

- Διακρίνεται λανθασμένη αλληλουχία και έλλειψη λογικής όταν διηγείται μια ιστορία.
- Αρχίζουν να εμφανίζονται έλλειψη προσοχής και πρωτοβουλίας.
- Εμφανίζει δυσκολία στην ολοκλήρωση προτάσεων.
- Σωματικές δυνατότητες όπως ισορροπία και δύναμη υποβαθμίζονται.
- Επαναλαμβάνει ερωτήσεις.
- Μιλά λιγότερο σε σχέση με το παρελθόν.

- Δυσκολεύεται να θυμηθεί νέα πρόσωπα.
- Δεν πραγματοποιεί με την ίδια άνεση σχέδια και πλάνα.



Εικόνα 1.2.2: Απεικόνιση του εγκεφάλου στο πρώιμο στάδιο. Προσαρμόστηκε από: (Mrs. Eva Binder MBA CEO EB Lizenzverwertungs GmbH Pharma Division, 2005)

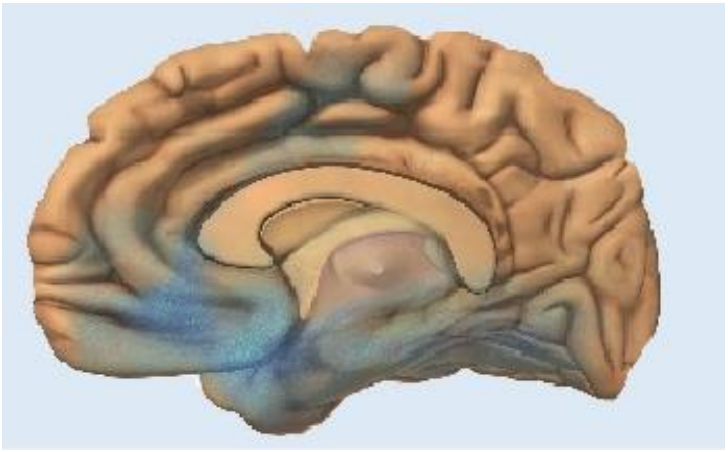
Μεσαίο στάδιο (Moderate Alzheimer's Disease)

Φάση 4: Σύνήθως αποτελεί κομβικό σημείο και στο οποίο γίνεται σαφές τόσο στην οικογένεια όσο και σε ένα γιατρό πως ένα άτομο αντιμετωπίζει σημαντικό πρόβλημα με τη μνήμη και τη σκέψη του, επηρεάζοντας την καθημερινότητά του. Υπάρχει μεγαλύτερη αύξηση των συμπτωμάτων σε σύκριση με εκείνων που παρατηρούνται στο πρώτο στάδιο. Η απώλεια μνήμης γίνεται εντονότερη, οι σωματικές ικανότητες μειώνονται, εμφανίζεται έλλειψη πρωτοβουλίας και διαταραχές ύπνου ενώ υπάρχει και δυσκολία στην επικοινωνία. Επιπλέον στοιχεία που εμφανίζονται στον ασθενή είναι:

- Αυξάνεται η δυσκολία στην εύρεση των σωστών λέξεων με αποτέλεσμα να συνθέτει δικές του.
- Το ενδιαφέρον για την υγιεινή αρχίζει σταδιακά να παραμελείται.
- Η έλλειψη κριτικής ικανότητας και πρωτοβουλίας, οδηγούν σε λανθασμένες αποφάσεις.
- Εμφανίζονται διαταραχές στον ύπνο.
- Οι σωματικές ικανότητες μειώνονται.
- Παράνοια, κυκλοθυμία και ψευδαισθήσεις είναι δυνατόν να εμφανιστούν και σε αυτό το στάδιο.
- Δυσκολία στην αναγνώριση οικείων προσώπων.
- Παρουσιάζει προβλήματα σε καθημερινές δραστηριότητες όπως ντύσιμο, ανάγνωση, γραφή και αριθμητική.
- Αποκτά καχυποψία και γίνεται πιο ευέξαπτος.

Φάση 5: Τα συμπτώματα γίνονται ακόμα πιο έντονα με συνέπεια το άτομο να χάνει σταδιακά την αίσθηση του χώρου και του χρόνου. Είναι δύσκολο να θυμηθεί προσωπικές του πληροφορίες όπως διεύθυνση ή αριθμό τηλεφώνου είτε να δυσκολεύεται να αναγνωρίσει σε τι εποχική περίοδο βρίσκεται και να φορά λανθασμένη ένδυση. Ως εκ τούτου καθημερινότητα του ασθενούς αρχίζει να γίνεται

ακόμα πιο δύσκολη τόσο σε κοινωνικό όσο και σε επαγγελματικό επίπεδο για τον ίδιο αλλά και την οικογένεια του.



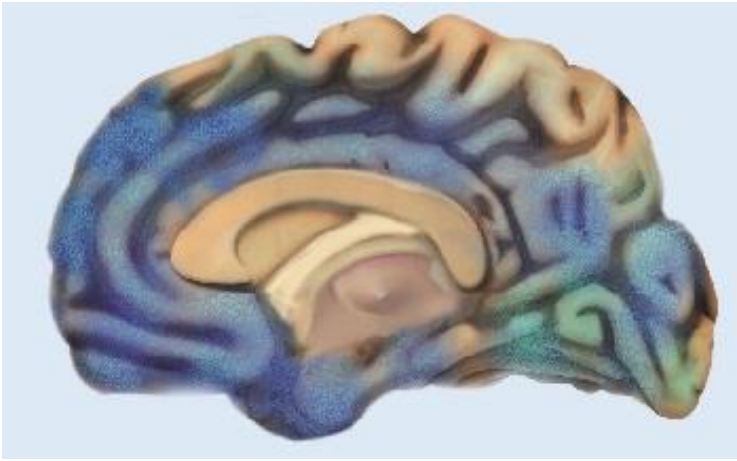
Εικόνα 1.2.3: Απεικόνιση του εγκεφάλου στο μεσαίο στάδιο. Προσαρμόστηκε από: (Mrs. Eva Binder MBA CEO EB Lizenzverwertungs GmbH Pharma Division, 2005)

Τρίτο στάδιο (Severe Alzheimer's Disease)

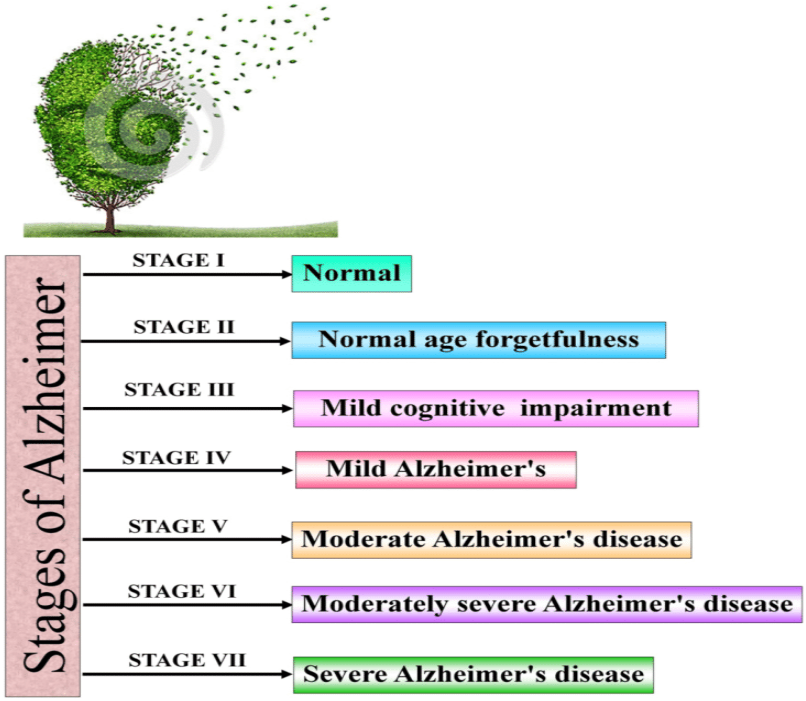
Φάση 6: Καθώς η ασθένεια προχωρά στο τελικό στάδιο βασικές και καθημερινές ανάγκες δυσχαιρένονται ακόμα περισσότερο. Υπάρχει εμφανής δυσκολία στο άτομο να συνδυάζει πρόσωπα μαζί με ονόματα, παραισθήσεις όπως το να θέλει να εργάζεται ενώ βρίσκεται σε συνταξιοδότηση. Έντονες δυσκολίες παρουσιάζονται στη κατάποση του φαγητού είτε να μπορούν να τραφούν και ντυθούν αυτόνομα.

Φάση 7:Αυτό το στάδιο χαρακτηρίζεται από την πλήρη απώλεια μνήμης του ασθενή και πλέον εξαρτάται από τη βοήθεια που του παρέχουν οι συγγενείς ώστε να ανταπεξέλθει στις καθημερινές δραστηριότητες. Επιπλέον δεν είναι σε θέση να περπατήσει χωρίς βοήθεια, η ομιλία μειώνεται σε λέξεις μέχρι να χαθεί τελείως ενώ ο εγκέφαλος χάνει τη δυνατότητα να ελέγχει βασικές βιολογικές λειτουργίες όπως η κατάποση. Επιπλέον στοιχεία που εμφανίζονται στον ασθενή είναι:

- Η ικανότητα της επικοινωνίας χάνεται τελείως.
- Ο εγκέφαλος δεν βρίσκεται σε θέση να ελέγξει βασικές βιολογικές λειτουργίες όπως η κατάποση.
- Περαιτέρω μείωση των σωματικών δυνατοτήτων.
- Χάνεται εύκολα η ισορροπία με αποτέλεσμα να κινείται με αναπηρικό καροτσάκι.
- Υπάρχει ακράτια ούρων.



Εικόνα 1.2.4: Απεικόνιση του εγκεφάλου στο τρίτο στάδιο. Προσαρμόστηκε από: (Mrs. Eva Binder MBA CEO EB Lizenzverwertungs GmbH Pharma Division, 2005)



Εικόνα 1.2.5: Απεικόνιση όλων των σταδίων της νόσου Αλτσχάιμερ.. Προσαρμόστηκε από: (Mohammad Kalim Ahmad Khan, 2016)

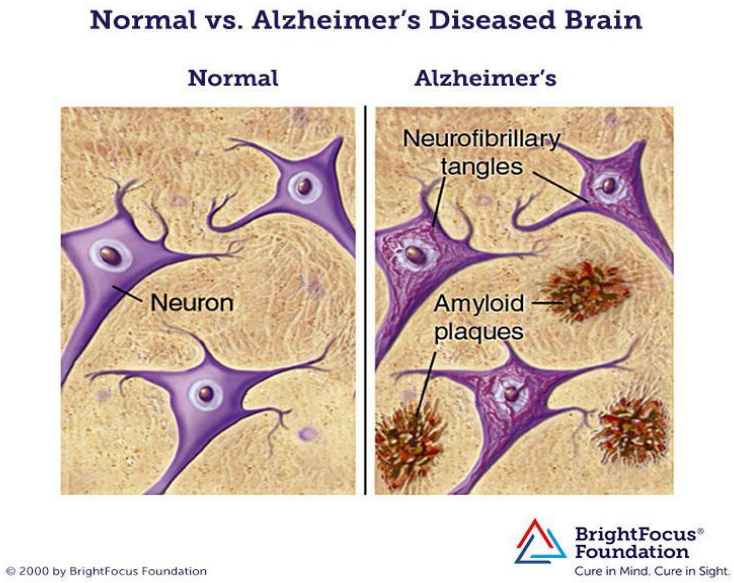
1.3 Αίτια πρόκλησης της νόσου Αλτσχάιμερ

Πολλοί επιστήμονες πιστεύουν πως δυο είναι οι κύριοι παράγοντες στους οποίους οφείλεται ο θάνατος των νευρώνων του εγκεφάλου και είναι οι παρακάτω:

1. **Αμυλοειδής πλάκες:** Ένα από τα χαρακτηριστικά της νόσου του Alzheimer είναι η συσσώρευση αμυλοειδών πλακών ανάμεσα στα νευρικά κύτταρα (νευρώνες) στον εγκέφαλο. Πιο συγκεκριμένα στην επιφάνεια της κυτταρικής μεμβάνης ενός νευρώνα του εγκεφάλου υπάρχει ένα συγκεκριμένο μόριο που

ονομάζεται πρόδρομη πρωτεΐνη αμυλοειδούς ή αλλιώς APP (Amyloid Precursor Protein) το οποίο βοηθά στην ανάπτυξη του νευρώνα είτε στην επιδιόρθωση τυχών βλαβών του. Όπως οι υπόλοιπες πρωτείνες μετά από καιρό χάνει τη λειτουργικότητα της και γι αυτό το λόγο πρέπει να αφαιρεθεί από τη κυτταρική μεμβράνη. Το ένας μέρος του μορίου αυτού βρίσκεται εσωτερικά της μεμβράνης και το άλλο εξωτερικά. Συγκεκριμένα ένζυμα αναλαμβάνουν να τεμαχίσουν το συγκεκριμένο μόριο σε μικρότερα πεπτίδια ώστε να είναι διαλυτά από τον οργανισμό και να απομακρυνθούν από αυτόν. Η α σεκρατάση αναλαμβάνει το εξωτερικό κομμάτι και η γ σεκρατάση το εσωτερικό. Ο νευρώνας όμως αλληλιπιδρά και με άλλα ένζυμα όπως η β σεκρατάση. Όταν εκείνη σε συνδυασμό με τη γ αναλαμβάνουν να τεμαχίσουν το APP, τότε τα κομμάτια στα οποία τεμαχίζεται το μόριο είναι μεγαλύτερα με αποτέλεσμα να δημιουργείται ένα μονομερές που ονομάζεται β αμυλοειδές και επιπλέον να είναι αδιάλυτο από τον οργανισμό. Όταν πολλά τέτοια μονομερή συγκεντρωθούν μαζί έξω από το νευρώνα ενώνονται και σχηματίζουν τις λεγόμενες αμυλοειδής πλάκες, μόρια αδιάλυτα, που εμποδίζουν την επικοινωνία μεταξύ των νευρώνων. Επίσης αυτές οι πλάκες μπορούν να προσκοληθούν πάνω σε αιμοφόρα αγγεία με συνέπεια τα τοιχώματα τους να αποδυναμώνονται και να αυξάνεται ο κίνδνος αιμορραγίας.

2. **<u>Νευροϊνιδιακοί σωροί:</u>** Οι νευροϊνιδιακοί σωροί είναι αδιάλυτες περιστρεφόμενες ίνες που βρίσκονται μέσα στα κύτταρα του εγκεφάλου. Η δομή ενός νευρώνα εσωτερικά αποτελείται από επιμέρους μικροσωληνίσκους οι οποίοι σχηματίζουν το κυταρροσκελετό και μέσω αυτών μεταφέρονται θρεπτικές ουσίες κατά μήκος του κυτταροσκελετού. Μια ειδική πρωτεΐνη που ονομάζεται tau ευθύνεται ώστε οι σωλήνες αυτοί να συγκρατούνται μεταξύ τους. Μπορείτε να φανταστείτε τους μικροσωληνίσκους σαν τις γραμμές του τρένου και την πρωτεΐνη ως εκείνη που ενώνει τις γραμμές αυτές. Οι ύπαρξη των αμυλοειδών πλακών έξω από τον νευρώνα έχεις ως επίδραση την ενεργοποίηση ενός ενζύμου που ονομάζεται κινάσημ εσωτερικά του κυττάρου. Το συγκεκριμένο ένζυμο μεταφέρει φωσφορικό άλας που αλληλιπεδρά με τις πρωτεΐνες tau που συγκρατούν τους σωλήνες και εκείνες με της σειρά τους αλλάζουν σχήμα, αποσπούνται από τους σωλήνες χάνοντας τη δομή τους και σχηματίζουν ενιαία σύμπλοκα ή αλλιώς νευροϊνιδιακούς σωρούς. Ως συνέπεια αυτής τη διαδικασίας συνεπάγεται ο εγκέφαλος να ατροφεί καθώς τα θρεπτικά συστατικά να μην μοιρούν να μεταφερθούν κατά μήκος των σωλήνων.



Εικόνα 1.3.1: Αριστερά φυσιολογικός εγκέφαλος. Δεξιά εγκέφαλος με αμυλοειδής πλάκες και νευροϊνιδιακά πλέγματα.

Επίσης οι επιστήμονες διαχωρίζουν την ασθένεια του Αλτσχάιμερ σε δύο κατηγορίες:

1. **<u>Σποραδική:</u>** Σε αυτή τη περίπτωση η ακριβής αιτία της νόσου δεν έχει προσδιορισθεί και την αποδίδουν σε γενετικούς και περιβαλλοντολογικούς παράγοντες και απαντάται στο 90 – 95% του ποσοστού των ασθενών. Στις ηλικίες 60 - 65 ο κίνδυνος να εμφανιστεί η νόσος είναι περίπου 1% ενώ στις ηλικίες πάνω από 85 είναι περίπου 50%. Επίσης ένας άλλος παράγοντας να εμφανιστεί το Αλτσχάιμερ είναι το γονίδιο απολιποπρωτεΐνης APOEe4. Εάν για παράδειγμα ένας απόγονος κληρονομήσει το APOEe4 από τον ένα γονέα αυξάνεται η πιθανότητα να νοσήσει. Αν όμως το κληρονομήσει και από τους δύο γονείς τότε αυτή η πιθανότητα διπλασιάζεται.

2. **<u>Οικογενειακή:</u>** Σε αυτήν την περίπτωση κληρονομείται ένα γονίδιο το οποίο έχει μεταλλαχθεί που επιταχύνει την εκδήλωση της νόσου και αφορά το 5 - 10% των περιπτώσεων. Οι μεταλλάξεις των γονιδίων προσενιλίνη 1 και προσενιλίνη 2 στο χρωμόσωμα 14 και 1 αντίστοιχα, τα οποία είναι υποκατάστατα της πρωτεΐνης γάμα σεκρατάσης, επηρεάζουν την λειτουργία της συγκεκριμένης πρωτεΐνης. Η γ σεκρατάση οφείλεται στο κόψιμο των αμυλοειδών πλακών. Όταν υφίσταται τις αλλαγές αυτές η πρωτεΐνη αυτή κόβει μεγαλύτερα κομμάτια αμυλοειδών πλακών που ευθύνονται για την θανάτωση κυττάρων στον εγκέφαλο. Επίσης στο χρωμόσωμα 21 ή αλλιώς το σύνδρομο down βρίσκεται το γονίδιο που ευθύνεται για τη παραγωγή των αμυλοειδών πλακών. Αυτό σημαίνει πως άτομα με σύνδρομο down έχουν ένα επιπλέον γονίδιο αμυλοειδούς πλάκας και επομένως αυξημένη πιθανότητα εκδήλωσης αμυλοειδών πλακών. Σε αυτές τις περιπτώσεις το Αλτσχάιμερ μπορεί να εκδηλωθεί ακόμα και στην ηλικία των 40.

1.4 Στατιστικά στοιχεία για τη νόσο Αλτσχάιμερ

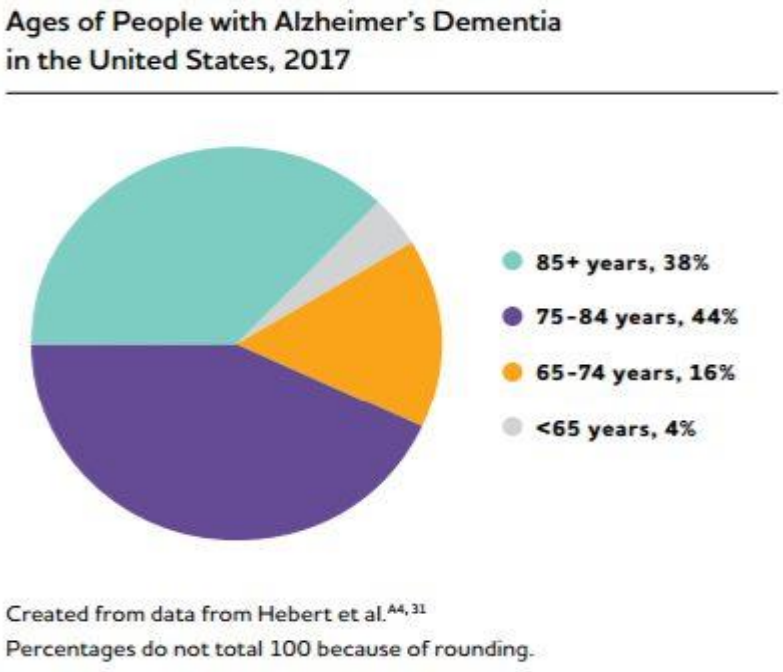
Τα παρακάτω στοιχεία παρουσιάζονται σύμφωνα με έρευνες που διεξήχθησαν το έτος 2016.

1. Το Αλτσχάιμερ παγκοσμίως:

- Παγκοσμίως, γύρω στα 44 εκατομμύρια άνθρωποι πάσχουν από τη νόσο αλτσχάιμερ ή σχετίζονται με κάποιου είδους άνοια.
- Συνήθως 1 στους 4 ανθρώπους με αλτσχάιμερ διαγνώστηκαν πως έχουν τη νόσο.
- Το αλτσχάιμερ και η άνοια είναι πιο συχνό φαινόμενο στη δυτική Ευρώπη ενώ σε πολύ μικρότερο βαθμό βρίσκεται στην Βόρεια Αμερική.
- Το αλτσχάιμερ είναι λιγότερο διαδεδομένο στην υποσαχάρια (κάτω από την έρημο Σαχάρα) Αφρική.

2. Το Αλτσχάιμερ στις Ηνωμένες Πολιτείες Αμερικής:

- 1 στους 9 Αμερικανούς άνω των 65 ετών πάσχει από τη νόσο.
- 1 στους 3 Αμερικανούς των 85 ετών και άνω πάσχουν από τη νόσο.
- Εκτιμάται ότι περισσότεροι από 5.3 εκατομμύρια Αμερικανοί θα έχουν την ασθένεια έως το 2050, εάν δεν έχει βρεθεί θεραπεία.
- Το τυπικό προσδόκιμο ζωής μετά από τη διάγνωση του Αλτσχάιμερ είναι 4 έως 8 χρόνια.
- Πιο συγκεκριμένα το ποσοστό των ασθενών σύμφωνα με την ηλικία τους στις Ηνωμένες Πολιτείες είναι:



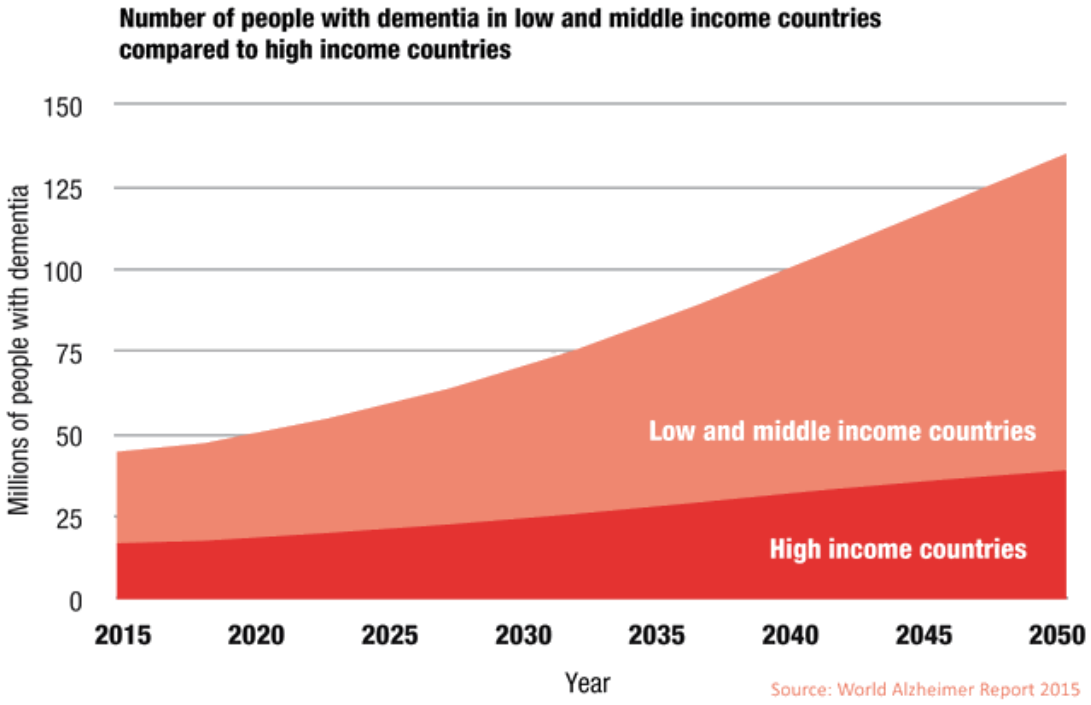
Εικόνα 1.4.1: Ποσοστά Alzheimer με βάση την ηλικία, στις Ηνωμένες Πολιτείες. Προσαρμόστηκε από: (Hebert et al., 2017)

3. Το κόστος φροντίδας για τη νόσο Αλτσχάιμερ:

- Το κόστος της φροντίδας των ασθενών με Αλτσχάιμερ στις Ηνωμένες Πολιτείες υπολογίζεται γύρω στα 236 δισ. Δολάρια το 2016.
- Το παγκόσμιο κόστος της νόσου του Αλτσχάιμερ και της άνοιας υπολογίζεται γύρω στα 605 δισ. Δολάρια.
- Το συνολικό κόστος που δαπανείται για τους Αμερικάνους από τις ηλικίες και άνω για προγράμματα φροντίδας είναι: Medicare 113 δισ. Δολάρια, Medicaid 41 δισ. Δολάρια, από προσωπικές δαπάνες 44 δισ. Δολάρια και για άλλους λόγους 29 δισ. Δολάρια .

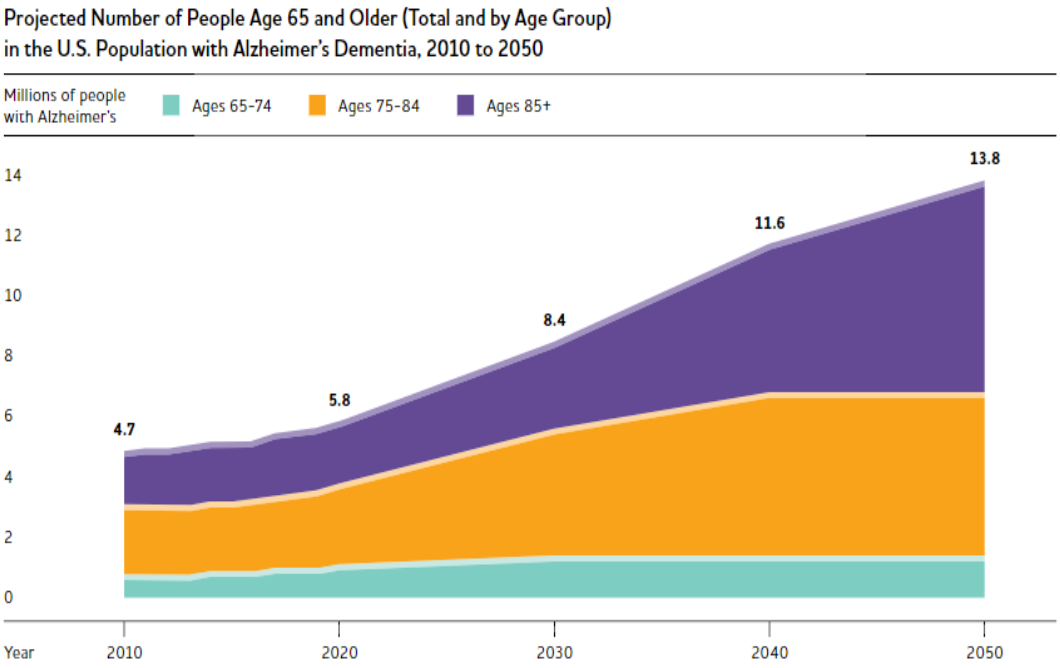
4. Προβλέψεις και εκτιμήσεις για την νόσο Αλτσχάιμερ στα προσεχή χρόνια:

- Στο παρακάτω γράφημα παρουσιάζονται τα προβλεπόμενα στατιστικά των ανθρώπων που θα πάσχουν από τη νόσο σε χώρες με χαμηλό και μεσαίο εισόδημα, συγκριτικά με εκείνες με υψηλό εισόδημα.



Εικόνα 1.4.2: Με έντονο κόκκινο οι χώρες με υψηλό εισόδημα. Με ανοικτό κόκκινο οι χώρες με χαμηλό και μεσαίο εισόδημα. Προσαρμόστηκε από: (World Alzheimer Report, 2015)

- Στο παρακάτω γράφημα παρουσιάζεται ο προβλεπόμενος αριθμός των ηλικιών που θα πάσχουν από τη νόσο Αλτσχάιμερ σε μελλοντικά έτη.



Εικόνα 1.4.3: Με πράσινο οι ηλικίες 65 έως 74. Με πορτοκαλί οι ηλικίες 75 έως 84. Με μπλε οι ηλικίες από 85 και άνω. Προσαρμόστηκε από: (World Alzheimer Report, 2010)

1.5 Τρόποι διάγνωσης και θεραπείας της νόσου

Μέχρι σήμερα δεν έχει ανακαλυφθεί η αποκλειστική εξέταση που θα ταυτοποιεί τη νόσο Αλτσχάιμερ. Επομένως ακολουθείται μια σειρά σε κάποιο νευρολογικό εργαστήριο με στόχο :

- Τη διαπίστωση του είδους και της βαρύτητας των συμπτωμάτων.
- Τον έλεγχο και τον αποκλεισμό συμπτωμάτων που φαίνονται να μοιάζουν με εκείνα της νόσου.
- Το σχεδιασμό της θεραπείας σύμφωνα με τη διάγνωση.

Οι κύριες διαγνωστικές μέθοδοι που εφαρμόζονται είναι:

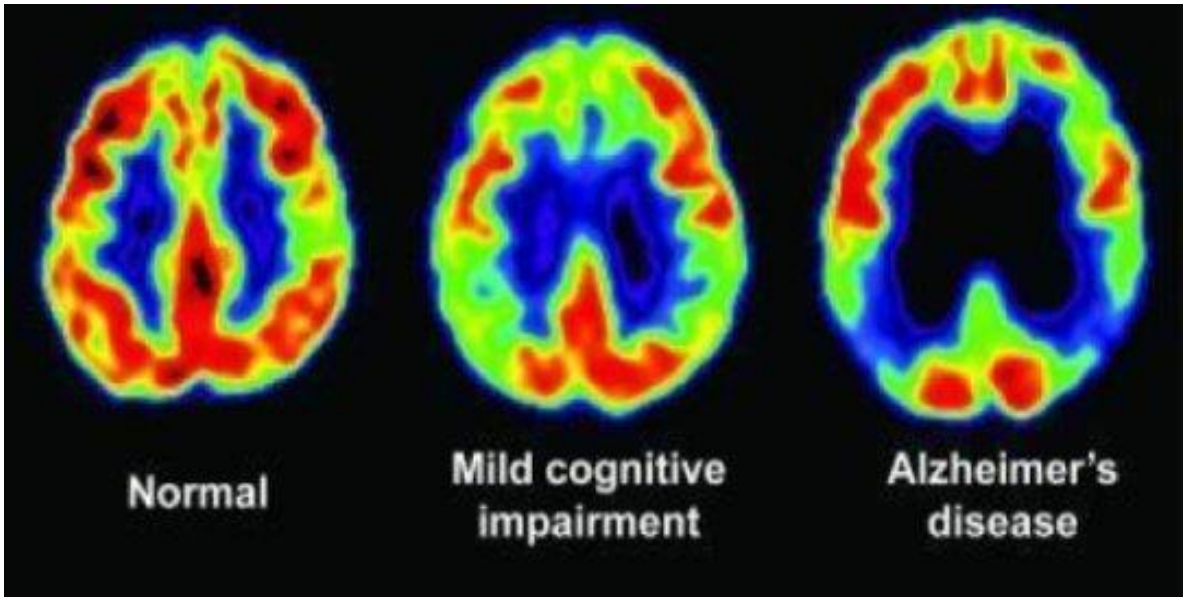
1. **Διεξοδική εξέταση του ιατρικού ιστορικού του ασθενούς:** Με αυτό τον τρόπο οι γιατροί χρησιμοποιούν συγκεκριμένες κλίμακες για να αξιολογήσουν τη συμπεριφορά και το γνωστικό επίπεδο του ασθενούς είτε αναζητούν προηγούμενες φαρμακευτικές αγωγές, όπως για την ύπαρξη ιστορικού κατάθλιψης.

2. **Εξετάσεις αίματος:** Όταν διαπιστώνεται πως έχουν επηρεαστεί οι νοητικές ικανότητες του ασθενούς, ακολουθείται μια σειρά από αιματολογικές εξετάσεις για να αποκλειστούν άλλα απλά αίτια άνοιας όπως ο υποθυρεοειδισμός ή έλλειψη βιταμινών.

3. **Απεικονιστικές εξετάσεις:** Συνήθως εφαρμόζονται αξονική ή τομογραφική εξέταση που παρέχουν πληροφορίες όπως την ύπαρξη συμπτωμάτων που μοιάζουν με τη νόσο αλτσχάιμερ είτε μεταβολές στην περιοχή του εγκεφάλου.

4. **<u>Ειδικές εξετάσεις:</u>** Αυτές οι εξετάσεις γίνονται μόνο σε ορισμένες περιπτώσεις όταν είναι ανάγκη να παρέχονται περισσότερα στοιχεία για τη διάγνωση όπως:

- <u>Σπινθηρογράφημα αιματικής ροής και τομογραφία εκπομπής ποζιτρονίων:</u> Με το σπινθηρογράφημα αιματικής ροής (SPECT) παρέχονται πληροφορίες για τις μεταβολικές διεργασίες του εγκεφάλου. Από την άλλη πλευρά υπάρχουν σύγχρονα ραδιοφάρμακα για Pet-scans τα οποία απεικονίζουν τις συγκεντρώσεις των αμυλοειδών πλακών στον εγκέφαλο.



Εικόνα 1.5.1: Απεικόνιση από Pet-Scan, που απεικονίζει τα πρώτα σημάδια του Αλτσχάιμερ. Προσαρμόστηκε από: (University of California – Berkeley, 2009)

- <u>Οσφυονωτιαία παρακέντηση:</u> Σε αυτή την εξέταση γίνεται εισαγωγή μιας λεπτής βελόνας μέσα στη σπονδυλική στήλη, στην περιοχή της οσφύος (μέσης) όπου βρίσκεται το εγκεφαλονωτιαίο υγρό. Από το εγκεφαλονωτιαίο υγρό μετρούνται κάποια ειδικά μόρια τα οποία αυξάνουν την ακρίβεια της διάγνωσης για την νόσο.

- <u>Γενετικός έλεγχος:</u> Αυτή η εξέταση γίνεται με σκοπό να διαπιστωθούν εάν υπάρχει μετάλλαξη στα 3 γονίδια (προσενιλίνη 1,προσενιλίνη 2, πρόδρομη πρωτεΐνη του αμυλοειδούς) που ευθύνονται για την πρόωρη εκδήλωση της νόσου. Παρόλο αυτά η διεκπεραίωση αυτής της εξέτασης είναι δύσκολη διότι γίνεται σε συγκεκριμένα κέντρα εξέτασης και πρέπει να έχει ακολουθήσει ο έλεγχος συγκεκριμένων πρωτοκόλλων.

Ακόμα δεν έχει υπάρξει τρόπος που να καταπολεμήσει οριστικά τη νόσο γι' αυτό οι δύο κύριες κατηγορίες που εφαρμόζονται είναι:

- Παροχή φροντίδας: Στα πρώτα στάδια της νόσου που ο ασθενής εμφανίζει μικρές απώλειες μνήμης η φροντίδα μπορεί συχνά να παρέχεται από σπίτι υπό τις οδηγίες του γιατρού. Η παροχή φροντίδας για ένα άτομο με νόσο Αλτσχάιμερ μπορεί να είναι δύσκολη και ψυχοφθόρος εμπειρία για τη σύζυγο και τα μέλη της οικογένειας. Αυτοί που παρέχουν φροντίδα χρειάζονται βοήθεια από οργανισμούς υποστήριξης είτε συγγενείς και φίλους. Η κατάλληλη διατροφή και η πρόσληψη υγρών είναι αρκετά σημαντική ενώ δε χρειάζονται μεγάλες αλλαγές στη καθημερινότητα του ασθενή όπως μια μετακόμιση. Όταν η κατάσταση του ασθενούς αρχίζει να επιδεινώνεται, τότε απαιτείται νοσηλευτική φροντίδα και είναι συνήθως απαραίτητο να μπει το άτομο σε κάποιο πρόγραμμα υπηρεσίας παροχής φροντίδας.

- Φαρμακευτική αγωγή: Τα σύγχρονα φάρμακα για τη νόσο δεν μπορούν να την αναστείλουν ή να τη σταματήσουν αλλά είναι ικανά έως κάποιο βαθμό να επιβραδύνουν την επιδείνωση των συμπτωμάτων. Όλα τα φάρμακα που έχουν εγκριθεί από τον Εθνικό Οργανισμό Φαρμάκων για τη θεραπεία της νόσου ανήκουν σε μία κατηγορία φαρμάκων που ονομάζονται αναστολείς χολινεστεράσης. Τα φάρμακα αυτά έχουν ως στόχο να βελτιώσουν τα επίπεδα της ακετυλοχολίνης στον εγκέφαλο, ενός νευροδιαβιβαστή που διαδραματίζει σημαντικό ρόλο στις δεξιότητες της μνήμης. Τέτοιου είδους φάρμακα είναι:

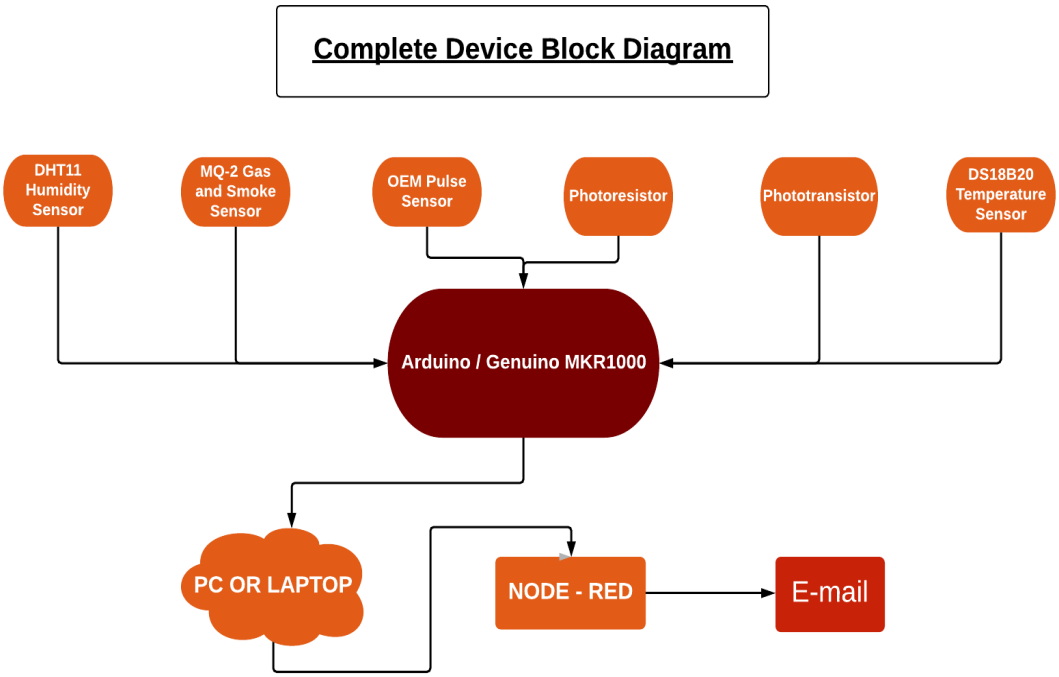
  1) Donepezil: Είναι το πιο συχνά χορηγούμενο φάρμακο από τους γιατρούς αλλά έχει παρενέργειες όπως ναυτία, διάρροια, ζαλάδα, οι οποίες δεν είναι έντονες και διαρκούν μικρό χρονικό διάστημα.
  2) Rivastigamine: Το φάρμακο αυτό έχει επίδραση όταν η νόσος βρίσκεται στο πρώτο μέχρι το μεσαίο στάδιο και οι παρενέργειες του είναι εμετός και ναυτία.
  3) Galanatamine: Είναι το νεότερο φάρμακο στην αγορά για το πρώτο έως το μεσαίο στάδιο της νόσου. Επιδρά στην συμπεριφορά και στις γνωστικές λειτουργίες. Οι παρενέργειες του βρίσκονται ακόμα υπό μελέτη.

1.6 Επιδίωξη πτυχιακής εργασίας

Στη σημερινή εποχή παρόλο που έχει σημειωθεί μεγάλη βελτίωση του βιοτικού επιπέδου, αύξηση του μέσου όρου ζωής και πρόοδος της τεχνολογίας παρατηρούμε πως υπάρχουν ασθένειες που πλήττουν την ανθρώπινη ζωή. Σε αυτή την πτυχιακή εργασία οι κύριοι στόχοι της είναι δύο. Ο πρώτος αφορά την κατανόηση της έννοιας αλλά και της σημασίας της νόσου Αλτσχάιμερ και το πώς επηρεάζει την καθημερινή διαβίωση. Ο δεύτερος αφορά την δημιουργία μιας βοηθητικής συσκευής για άτομα που πάσχουν από τη νόσο Αλτσχάιμερ με σκοπό την επιτήρηση και παρακολούθηση τους. Είναι ένας τρόπος ώστε να βελτιωθεί έως κάποιο βαθμό η καθημερινότητα του ασθενούς αλλά ταυτόχρονα να διευκολυνθούν και τα συγγενικά του πρόσωπα.

2. Hardware (Υλικά και Μεθοδολογία)

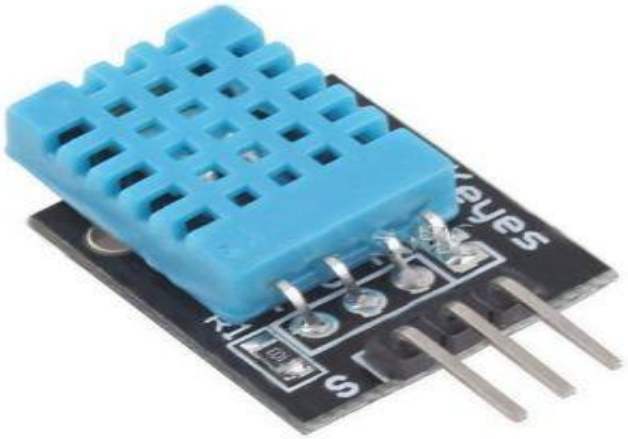
Το κομμάτι των υλικών στην συγκεκριμένη περίπτωση, αποτελεί τον πυρήνα της όλης διαδικασίας. Οι συσκευές που χρησιμοποιούνται και ο τρόπος λειτουργίας της κάθε μίας, καθορίζουν το αν η συσκευή που θα δημιουργηθεί θα είναι χρήσιμη ή όχι. Στο συγκεκριμένο κεφάλαιο, παρουσιάζονται οι δυνατότητες, ο τρόπος λειτουργίας, τα χαρακτηριστικά και η συνδεσμολογία, όλων των υλικών και εξαρτημάτων που θα χρησιμοποιηθούν, για την υλοποίηση της επιζητούμενης συσκευής. Ένα παράδειγμα της γενικής δομής του συστήματος απεικονίζεται στην παρακάτω εικόνα.



Εικόνα 2.1: Μπλοκ διάγραμμα γενικής δομής συστήματος. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

2.1 Υλικά και Εξαρτήματα

Για την κατασκευή της προαναφερόμενης συσκευής, χρησιμοποιήθηκαν 6 αισθητήρες, αντιστάσεις και LED, καθώς και μία πλακέτα μικροελεγκτή Arduino/Genuino MKR1000, σε συνδυασμό με μία IOT platform εν ονόματι Node-Red. Οι αισθητήρες συνδέθηκαν μέσω breadboard με το Arduino και με σύνδεση Wi-Fi, που διαθέτει το Arduino, έρχονται σε επικοινωνία με την Node-Red. Τέλος, η Node-Red είναι υπεύθυνη για την ολοκλήρωση της διαδικασίας, η οποία μέσω e-mail θα ενημερώνει το αρμόδιο άτομο για την οποιαδήποτε ανίχνευση από τους 6 αισθητήρες.



Εικόνα 2.1.1: Αισθητήρας Υγρασίας DHT11. Προσαρμόστηκε από: (GRobotronics, © 2011 - 2018)

Εικόνα 2.1.2: Αισθητήρας Αερίων και Καπνού MQ-2. Προσαρμόστηκε από: (GRobotronics, © 2011 - 2018)

Εικόνα 2.1.3: Αισθητήρας Παλμών της OEM. Προσαρμόστηκε από: (GRobotronics, © 2011 - 2018)

Εικόνα 2.1.4: Αισθητήρας Υπέρυθρης Ακτινοβολίας (Phototransistor). Προσαρμόστηκε από: (GRobotronics, © 2011 - 2018)

Εικόνα 2.1.5: Αισθητήρας Φωτός (Photoresistor). Προσαρμόστηκε από: (GRobotronics, © 2011 - 2018)

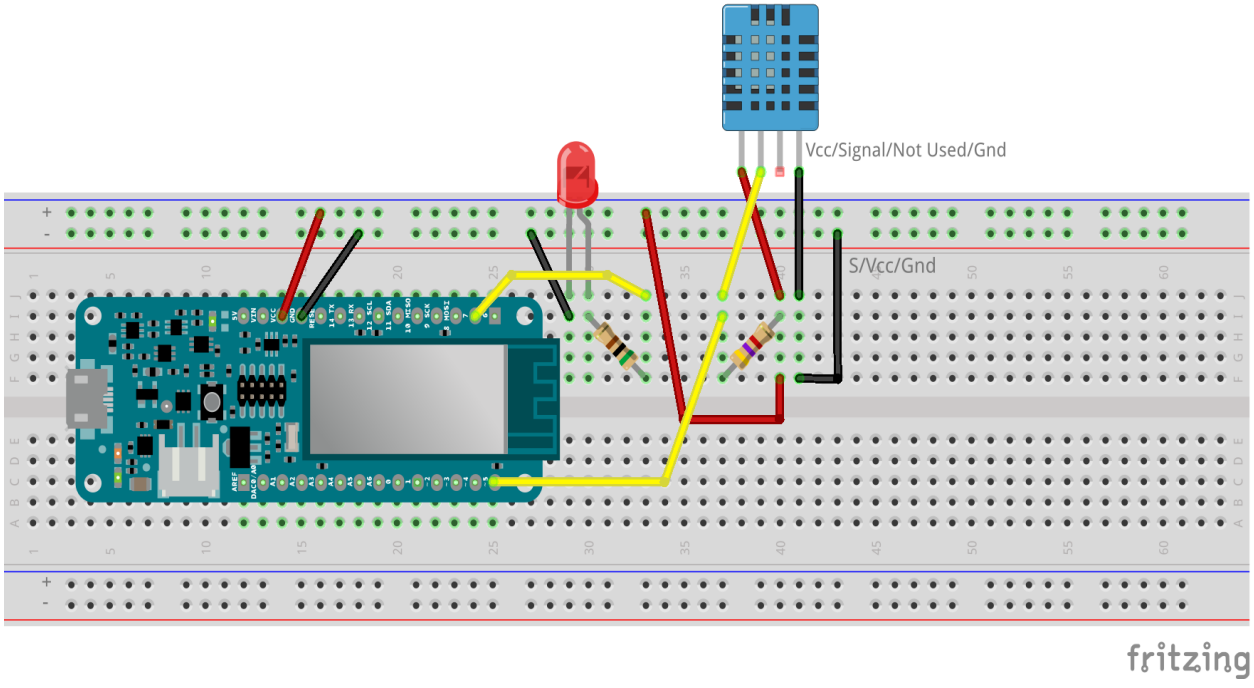
Εικόνα 2.1.6: Αισθητήρας Θερμοκρασίας (DS18B20). Προσαρμόστηκε από: (GRobotronics, © 2011 - 2018)

2.2 Αισθητήρες

2.2.1 Αισθητήρας Υγρασίας DHT11

Ο αισθητήρας υγρασίας DHT11 έχει διαστάσεις πλακέτας 21.9x16.6x7mm πάνω στην οποία είναι τοποθετημένος και διάκενο ανάμεσα στα pins 0.1”/2.54mm. Έχει ψηφιακή είσοδο και δειγματοληψία ανά δευτερόλεπτο (1Hz). Το εύρος υγρασίας που μετράει ιδανικά, είναι από 20% έως 80% με ακρίβεια 5% και δέχεται τροφοδοσία 3Volt έως 5Volt DC. Σημαντική προσθήκη, είναι ότι χρειάζεται μία αντίσταση 4.7 ΚΩ ανάμεσα στα pin της τάσης και της ψηφιακής εξόδου του και είναι συμβατός με τις περισσότερες αναπτυξιακές πλακέτες, όπως Arduino και Raspberry Pi. Τέλος, ο αισθητήρας στην δική μας περίπτωση, έχει ρυθμιστεί έτσι ώστε κάθε φορά που ξεπερνάει την μέτρηση υγρασίας της τάξης του 85%, να ανάβει ένα αντίστοιχο LED κόκκινου χρώματος.

Η ατομική συνδεσμολογία του με το Arduino/Genuino MKR1000, προβάλλεται στην παρακάτω εικόνα, η οποία έχει δημιουργηθεί και προσαρμοστεί μέσω ειδικού προγράμματος, που ονομάζεται Fritzing:

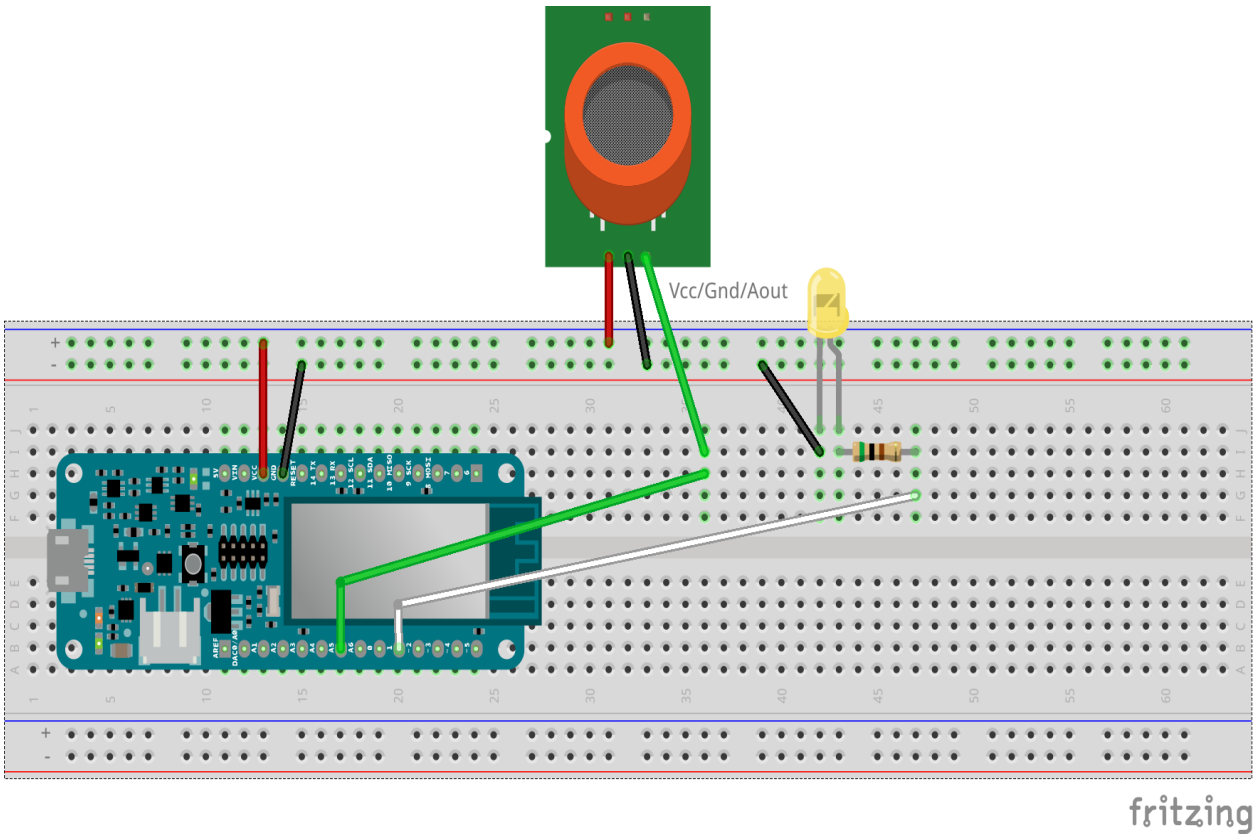


Εικόνα 2.2.1.1: Συνδεσμολογία Αισθητήρα Υγρασίας (DHT11) με Arduino Genuino MKR1000. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

2.2.2 Αισθητήρας Αερίων και Καπνού MQ-2

Ο αισθητήρας MQ-2, έχει την δυνατότητα ανίχνευσης υγραερίου (LPG), προπάνιου (Propane), και υδρογόνου (Hydrogen) και φυσικά καπνού (Smoke). Το μέγεθος των οπών στερέωσής του, είναι 2.0mm και οι διαστάσεις της πλακέτας στην οποία είναι τοποθετημένος, είναι 40x21mm. Η τάση εξόδου του, ενισχύεται με βάση την συγκέντρωση του αερίου που ανιχνεύει, πράγμα που σημαίνει ότι παρέχει γρήγορη απόκριση και ανάκτηση στις μετρήσεις του. Διαθέτει επιπλέον κύκλωμα ενίσχυσης, που προσθέτει την δυνατότητα αυξημένης ευαισθησίας, η οποία ευαισθησία, είναι και ρυθμιζόμενη. Η τάση λειτουργίας του είναι 3.3Volt έως 5Volt DC, είναι συμβατό με όλες τις πλακέτες Arduino και η διασύνδεση του με το Arduino/Genuino MKR1000, μπορεί να είναι είτε αναλογική είτε ψηφιακή. Στην δική μας κατασκευή όμως, χρησιμοποιούμε μόνο την αναλογική είσοδο. Τέλος, με την κάθε ανίχνευση οποιουδήποτε αερίου από τα όσα προαναφέρθηκαν ή καπνού, ανάβει ένα αντίστοιχο LED κίτρινου χρώματος.

Η ατομική συνδεσμολογία του με το Arduino/Genuino MKR1000, προβάλλεται στην παρακάτω εικόνα, η οποία έχει δημιουργηθεί και προσαρμοστεί μέσω ειδικού προγράμματος, που ονομάζεται Fritzing:

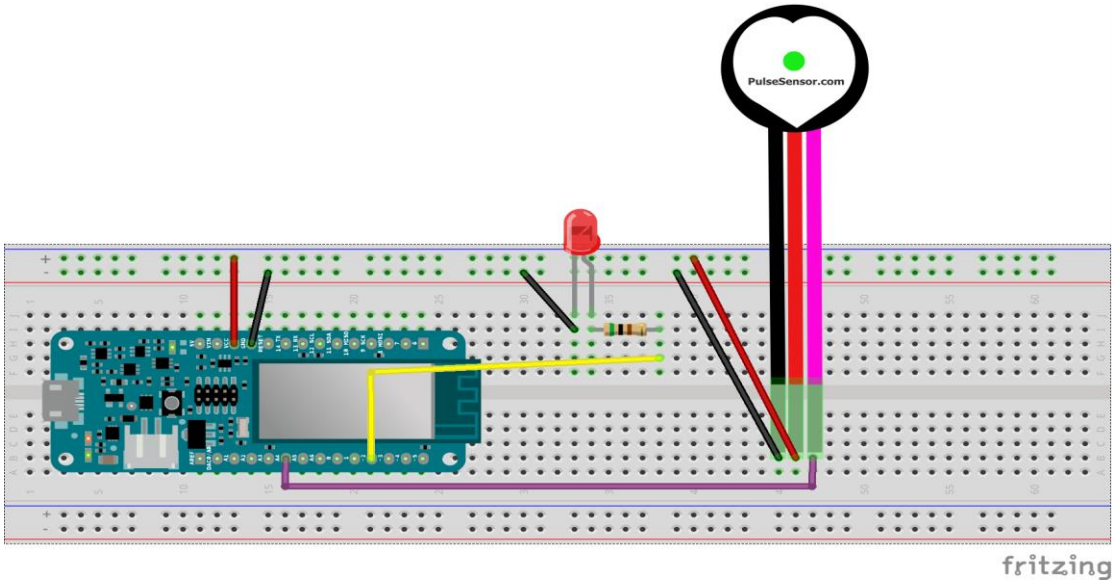


Εικόνα 2.2.2.1: Συνδεσμολογία Αισθητήρα Αερίων και Καπνού (MQ-2) με Arduino Genuino MKR1000. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

2.2.3 Αισθητήρας Παλμών της OEM

Ο αισθητήρας παλμών της OEM, είναι αισθητήρας plug-and-play για Arduino, όπου συνδυάζει έναν απλό οπτικό αισθητήρα παλμών, με ενίσχυση. Επιπλέον, διαθέτει κύκλωμα ακύρωσης θορύβου για εύκολη και γρήγορη μέτρηση και δέχεται τάση 5Volt DC. Πρόκειται για ένα 24-inch καλώδιο, έγχρωμα κωδικοποιημένο, με βασικές αρσενικές κεφαλές σύνδεσης, διάμετρο 0.625” (15.88mm) και πάχος 0.125” (3.18mm), το οποίο συνδέεται με αναλογική είσοδο και τοποθετείται στον δείκτη του αριστερού χεριού του ασθενούς, για την μέτρηση των παλμών του. Τέλος, με κάθε παλμό που ανιχνεύεται από τον αισθητήρα, ανάβει ένα αντίστοιχο LED κόκκινου χρώματος, υποδεικνύοντας ουσιαστικά τους παλμούς της καρδιάς του ασθενούς.

Η ατομική συνδεσμολογία του με το Arduino/Genuino MKR1000, προβάλλεται στην παρακάτω εικόνα, η οποία έχει δημιουργηθεί και προσαρμοστεί μέσω ειδικού προγράμματος, που ονομάζεται Fritzing:

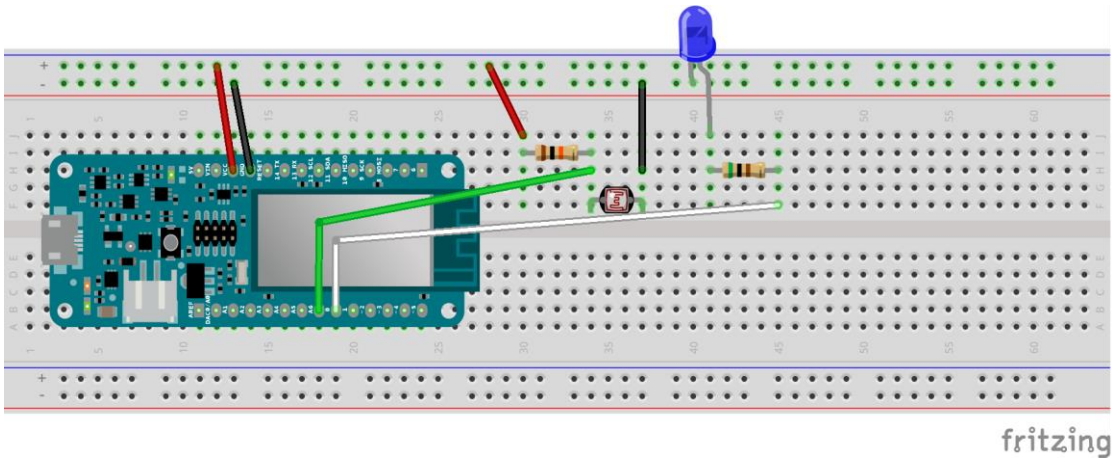


Εικόνα 2.2.3.1: Συνδεσμολογία Αισθητήρα Παλμών της OEM με Arduino Genuino MKR1000. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

2.2.4. Αισθητήρας Φωτοαντίστασης (Photoresistor)

Ο αισθητήρας φωτοαντίστασης, είναι στην ουσία μία αντίσταση, η οποία μεταβάλλει την τιμή της ανάλογα την έκθεσή της στο φως. Η αντίσταση του αισθητήρα, στο φως έχει την τιμή των 10ΚΩ, ενώ στο σκοτάδι την τιμή των 20ΚΩ. Σε συνδυασμό με μία αντίσταση 10ΚΩ, δημιουργούμε έναν διαιρέτη τάσης με αναλογική είσοδο, η οποία καταλήγει στο Arduino Genuino MKR1000. Κάθε φορά που η φωτοαντίσταση, έρχεται σε επαφή με φως, ένα αντίστοιχο LED μπλε χρώματος, ανάβει δείχνοντας την ύπαρξη φωτός, ενώ αντίστοιχα στην αντίθετη περίπτωση σβήνει. Οι διαστάσεις της φωτοαντίστασης είναι 2x4x5mm, 4mm μεταξύ των pins και η γραμμή μολύβδου της φωτοαντίστασης έχει μήκος 31mm.

Η ατομική συνδεσμολογία του με το Arduino/Genuino MKR1000, προβάλλεται στην παρακάτω εικόνα, η οποία έχει δημιουργηθεί και προσαρμοστεί μέσω ειδικού προγράμματος, που ονομάζεται Fritzing:

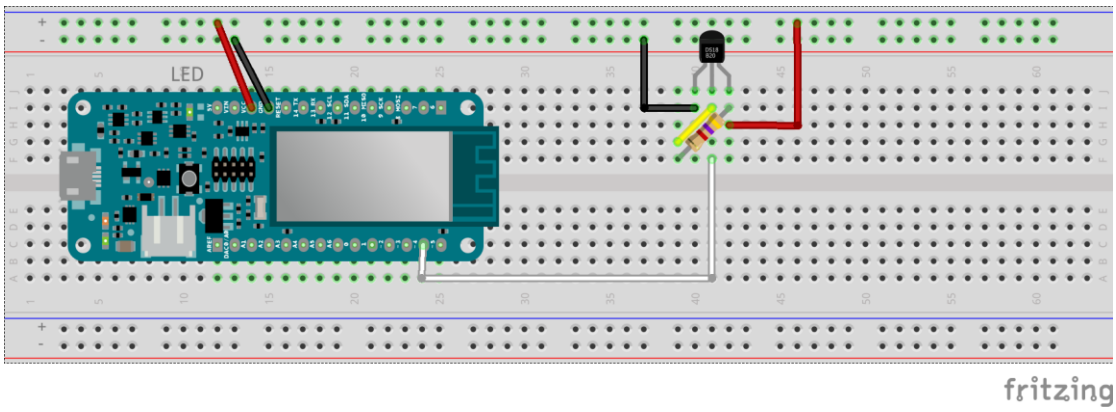


Εικόνα 2.2.4.1: Συνδεσμολογία Αισθητήρα Φωτοαντίστασης (Photoresistor) με Arduino Genuino MKR1000. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

2.2.5 Αισθητήρας Θερμοκρασίας DS18B20

Ο αισθητήρας θερμοκρασίας DS18B20, είναι ένας αισθητήρας με ψηφιακή έξοδο και χρησιμοποιείται με την προσθήκη της OneWire Library στις βιβλιοθήκες του λογισμικού του Arduino. Η θερμοκρασία που μετράει έχει ακρίβεια ± 0.5 °C σε θερμοκρασίες από (- 10 °C) έως (+ 85 °C) και συνολικό εύρος μέτρησης που κυμαίνεται από (- 55 °C) έως (+ 125 °C). Διαθέτει μοναδικό σειριακό αριθμό από 64-bit αποθηκευμένα στην ROM, που δίνει την δυνατότητα πολλαπλής σύνδεσης. Η τάση εισόδου που χρησιμοποιείται είναι 3V – 5.5V και η ανάλυσή του 9 – 12 bits. Ο DS18B20 είναι συμβατός με τις περισσότερες αναπτυξιακές πλακέτες (Arduino, Raspberry) και επιτυγχάνει της επικοινωνία με μόνο 1 pin. Επιπλέον, για την σύνδεσή του με μικροελεγκτή, απαιτείται συνήθως μία αντίσταση 4.7KΩ και η μέγιστη μετατροπή θερμοκρασίας που μπορεί να πετύχει, είναι 12 bits σε 750ms.

Η ατομική συνδεσμολογία του με το Arduino/Genuino MKR1000, προβάλλεται στην παρακάτω εικόνα, η οποία έχει δημιουργηθεί και προσαρμοστεί μέσω ειδικού προγράμματος, που ονομάζεται Fritzing:

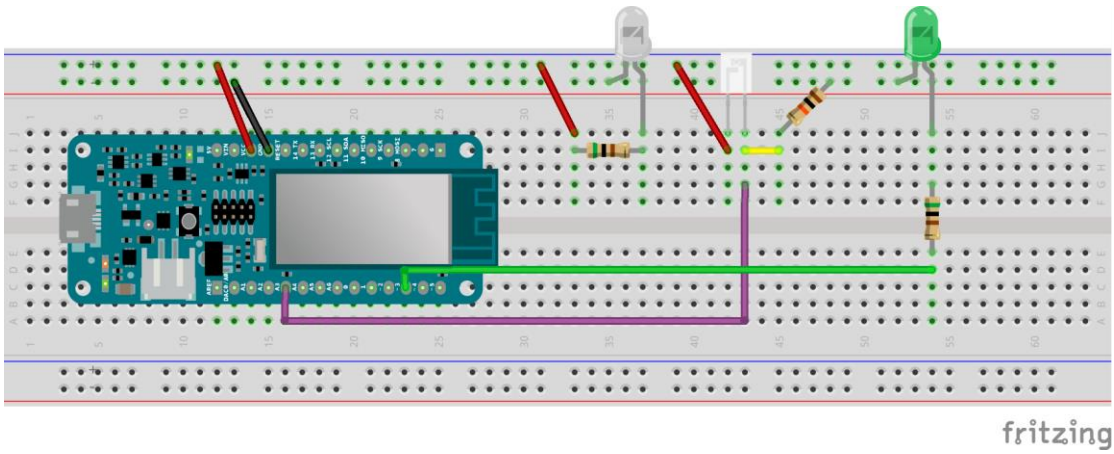


Εικόνα 2.2.5.1: Συνδεσμολογία Αισθητήρα Θερμοκρασίας (DS18B20) με Arduino Genuino MKR1000. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

2.2.6 Αισθητήρας Υπέρυθρης Ακτινοβολίας (Phototransistor)

Ο αισθητήρας υπέρυθρης ακτινοβολίας (Phototransistor) έχει σχηματικά την μορφή ενός απλού LED αλλά με μαύρο χρώμα και εντελώς διαφορετική λειτουργία. Πρόκειται για έναν ανιχνευτή με μαύρο φακό και IR (Infrared Radiation / Υπέρυθρη Ακτινοβολία) φίλτρο, το οποίο ανιχνεύει την υπέρυθρη ακτινοβολία και με την κατάλληλη συνδεσμολογία, μπορεί να μας υποδείξει πότε είναι εκτεθειμένο σε αυτή. Η οπτική γωνία που ανιχνεύει φτάνει μέχρι τις 20 μοίρες και έχει διάμετρο 3mm. Ο πιο συνηθισμένος τρόπος χρήσης του, τον οποίο και υιοθετήσαμε, είναι με την χρήση ενός LED υπερύθρων, που εκπέμπει στα 940nm και στην περίπτωση που η ακτινοβολία από το LED βρεθεί μέσα στο πεδίο ανίχνευσης του αισθητήρα υπερύθρων, τότε να ανάβει ένα άλλο LED (π.χ. κόκκινου χρώματος), το οποίο θα υποδηλώνει την ύπαρξη υπέρυθρης ακτινοβολίας.

Η ατομική συνδεσμολογία του με το Arduino/Genuino MKR1000, προβάλλεται στην παρακάτω εικόνα, η οποία έχει δημιουργηθεί και προσαρμοστεί μέσω ειδικού προγράμματος, που ονομάζεται Fritzing:



Εικόνα 2.2.6.1: Συνδεσμολογία Αισθητήρα Υπερύθρων (Phototransistor) με Arduino Genuino MKR1000. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

2.2.7 Προγραμματισμός και Μνήμη Arduino

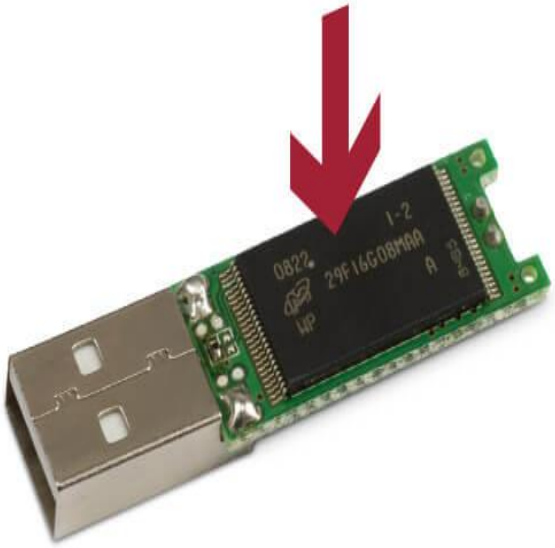
Ο προγραμματισμός του Arduino γίνεται σε γλώσσα προγραμματισμού C ή C^{++} στο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE). Το κυρίως πρόγραμμα που γράφεται για το Arduino ονομάζεται sketch και αποτελείται από δύο συναρτήσεις:

- **Void setup**: Αυτή η συνάρτηση εκτελείται μία φορά στην εκκίνηση του Arduino ή όταν γίνει επαναφορά με τη χρήση του κουμπιού reset. Τη χρησιμοποιούμε κυρίως για την αρχικοποίηση μεταβλητών, ακροδέκτες του Arduino (pin modes), τον ορισμό χρήσης κάποιας βιβλιοθήκης.

- **Void loop**: Σε αυτή τη συνάρτηση ο κώδικας που γράφεται εκτελείται συνεχώς και επαναλαμβανόμενα, μέχρι να το σταματήσει ο χρήστης.

Στο Arduino υπάρχουν τρεις ομάδες μνήμης που αποθηκεύονται τα δεδομένα:

- Μνήμη Φλας (Flash memory): Το πρόγραμμα (sketch) που δημιουργείται αποθηκεύεται στη μνήμη Flash, ένα είδος διατηρήσιμης (non-volatie) μνήμης. Τέτοιου είδους μνήμες έχουν την ιδιότητα να διατηρούν τα περιεχόμενα τους και μετά τη διακοπή της τροφοδοσίας και την επαναφόρτωση του προγράμματος με την επαναφορά της τροφοδοσίας.

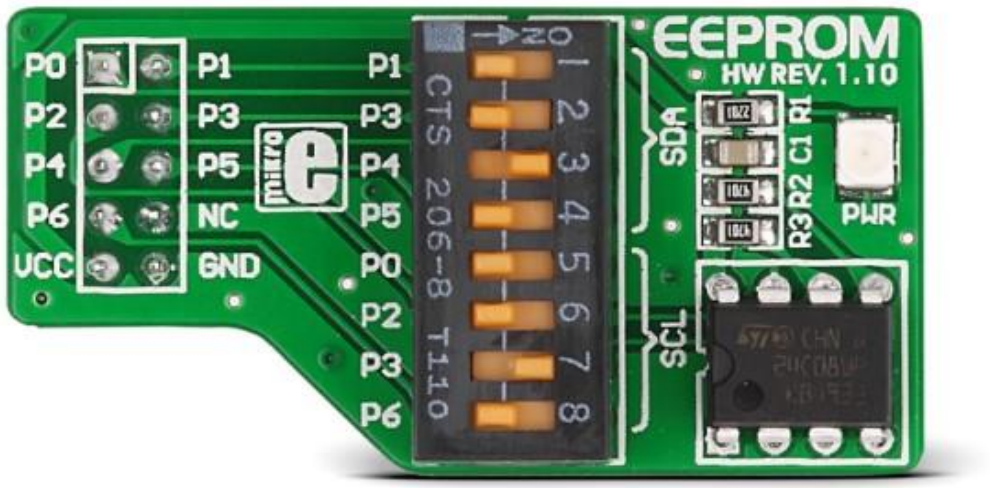


Εικόνα 2.2.7.1: Μνήμη Flash. Προσαρμόστηκε από: (© Flashbay ®, 2018)

- Μνήμη SRAM (Static Random Access Memory): Σε αυτή τη μνήμη αποθηκεύονται οι μεταβλητές όταν εκτελείται το κυρίως πρόγραμμα και είναι μη διατηρήσιμη.

Εικόνα 2.2.7.2: Μνήμη SRAM. Προσαρμόστηκε από: (© Technology Share ®, 2011)

- Μνήμη EEPROM: Αυτή η μνήμη ανήκει στη κατηγορία των διατηρήσιμων και συνήθως αποθηκεύονται μεταβλητές των οποίων τα δεδομένα θα χρησιμοποιούνται μακροπρόθεσμα από το χρήστη.



Εικόνα 2.2.7.3: Μνήμη EEPROM. Προσαρμόστηκε από: (© MikroElectronika d.o.o. ®, 2018)

Ανάλογα με το είδος των μεταβλητών που υπάρχουν στο πρόγραμμα του μικροελεγκτή καταλαμβάνουν και διαφορετικό χώρο στη μνήμη του. Πιο συγκεκριμένα:

Είδος Μεταβλητής	Επεξήγηση	Μέγεθος σε Byte	Εύρος Τιμών
Boolean	Αποθηκεύει δύο τιμές true ή false	1	True ή False
Char	Δεσμευμένη μεταβλητή που αποθηκεύονται μονοί χαρακτήρες ή αριθμοί	1	-128 έως 127
Unsigned char	Μη δεσμευμένη μεταβλητή που αποθηκεύεται συγκεκριμένο εύρος αριθμών	1	0 έως 255
Byte	Αποθηκεύει ένα μη δεσμευμένο αριθμό 8 bit	1	0 έως 255
Int	Αποθηκεύει σε μια μεταβλητή έναν αριθμό	2	-32.768 έως 32.767

Unsigned int	Αποθηκεύει σε μια μεταβλητή έναν αριθμό χωρίς αρνητικές τιμές	2	0 έως 65.535
Word	Αποθηκεύει ένα μη δεσμευμένο αριθμό 16 bit.	2	0 έως 65.535
Long	Αποθηκεύει αριθμούς μεγάλου μεγέθους 32 bit	4	-2.147.483.648 έως 2.147.483.647
Unsigned Long	Αποθηκεύει αριθμούς μεγάλου μεγέθους 32-bit χωρίς αρνητικές τιμές	4	0 έως 4.294.967.295
Float	Αποθηκεύει δεκαδικούς αριθμούς μέχρι 7 δεκαδικά ψηφία	4	-3.4028235+38 έως 3.4028235+38
Double	Αποθηκεύει δεκαδικούς αριθμούς μέχρι 16 δεκαδικά ψηφία	4	-3.4028235+38 έως 3.4028235+38
String	Αποθηκεύει μια σειρά από χαρακτήρων, συμβόλων ή αριθμών	Εξαρτάται από το μέγεθος του περιεχομένου του string	Συμβολοσειρά Χαρακτήρων
Array	Πίνακας με δεδομένα	Εξαρτάται από το μέγεθος του πίνακα	Πίνακας Τιμών
Void	Χρησιμοποιείται για τη δήλωση της συνάρτησης του προγράμματος που δεν επιστρέφει κάποια τιμή	0	Καμία

Οι μεταβλητές μπορούν να είναι:

- Τοπικές (local): Ορίζονται μέσα σε μία συνάρτηση του προγράμματος και είναι γνωστές μόνο μέσα σε αυτή.
- Καθολικές (global): Ορίζονται εκτός συνάρτησης του προγράμματος και είναι γνωστές σε κάθε συνάρτηση.

2.2.8 Arduino MKR1000

Το Arduino είναι μια πλατφόρμα ανοικτού κώδικα η οποία είναι ευέλικτη και εύκολη για χρήση τόσο στο κομμάτι του software όσο και του hardware. Απευθύνεται σε μεγάλο κοινό όπως σχεδιαστές, μηχανικούς, ηλεκτρονικούς δίνοντας τη δυνατότητα στους χρήστες να πειραματίζονται και να δημιουργούν διαδραστικά προγράμματα. Όλες οι πλακέτες του Arduino έχουν ως κοινό το προγραμματιστικό τους περιβάλλον γνωστό και ως Arduino IDE, με το οποίο αλληλεπιδρά ο χρήστης. Από αυτό το σημείο και έπειτα οι πλακέτες διαφοροποιούνται ανάλογα με τις ανάγκες που υπάρχουν. Διαφορές μπορεί να υπάρχουν ως προς:

Μικροελεγκτής/Επεξεργαστής: Αποτελεί τον εγκέφαλο της πλακέτας του Arduino. Υπάρχουν πολλών ειδών μικροελεγκτές που υποστηρίζουν διάφορες λειτουργίες και χαρακτηριστικά. Οι διαφορές ανάμεσα στους επεξεργαστές μπορεί να καθορίσει την ταχύτητα που λειτουργεί το Arduino.

Τάση εισόδου: Αυτή είναι η προτεινόμενη τάση λειτουργίας της πλακέτας. Φυσικά η πλακέτα μπορεί να δεχτεί ελάχιστα παραπάνω τάση αλλά είναι ασφαλέστερο να λειτουργεί ανάμεσα στα επιθυμητά όρια.

Ταχύτητα ρολογιού: Αυτή είναι η συχνότητα λειτουργίας του μικροελεγκτή και σχετίζεται με την ταχύτητα με την οποία μπορεί να εκτελεί εντολές. Οι περισσότεροι μικροελεγκτές που λειτουργούν σε 3V θα έχουν ρολόι (συχνότητα) στα 8MHz, ενώ οι περισσότεροι που λειτουργούν με 5V θα είναι χρονισμένοι στα 16MHz.

Ψηφιακές είσοδοι/έξοδοι: Αυτός είναι ο αριθμός των πινακίων-υποδοχών που υπάρχουν πάνω στη πλακέτα. Καθένα από αυτά μπορεί να επιλεχθεί σαν είσοδος είτε έξοδος.

Αναλογικές είσοδοι: Αυτός είναι ο αριθμός των πινακίων-υποδοχών αναλογικής εισόδου που είναι διαθέσιμες πάνω στην πλακέτα. Οι αναλογικοί ακροδέκτες έχουν την ένδειξη "A" ακολουθούμενος από τον αριθμό τους και επιτρέπουν να διαβάζονται αναλογικές τιμές χρησιμοποιώντας μετατροπέα αναλογικού προς ψηφιακό (ADC). Οι αναλογικές είσοδοι μπορούν επίσης να διαμορφωθούν ως περισσότερες ψηφιακές είσοδοι / έξοδοι.

Διαμόρφωση πλάτους παλμού: Είναι ο αριθμός ψηφιακών πινακίων - υποδοχών που μπορούν και παράγουν ένα παλμό διαμόρφωσης πλάτους. Αυτό σημαίνει πως το σήμα συμπεριφέρεται σαν αναλογική έξοδος και επιτρέπει στο Arduino να έχει μια αναλογική τάση μεταξύ μηδέν και τη τάση του συστήματος.

Μνήμη Flash: Αυτό είναι η χωρητικότητα της μνήμης προγράμματος που διαθέτει το Arduino για να αποθηκεύσει τα διάφορα προγράμματα. Δεν είναι διαθέσιμη όλη αυτή η μνήμη, καθώς ένα πολύ μικρό τμήμα καταλαμβάνεται από τον bootloader (συνήθως μεταξύ 0,5 και 2KB).

Ασπίδα: Η ασπίδα Wi-Fi του Arduino είναι ένα εξάρτημα που του επιτρέπει να συνδέεται στο Internet.

Λαμβάνοντας υπόψιν όλα τα παραπάνω θα γίνει σύγκριση των χαρακτηριστικών του Arduino MKR 1000 που χρησιμοποιήθηκε στην πτυχιακή εργασία σε σύγκριση με το πιο κοινό Arduino της αγοράς.

Είδος Arduino		
Τιμή	19.99$ - 23.00$	24.24$ - 71.00$
Διαστάσεις	68.6mm x 53.4mm	61.5mm x 25mm
Μικροελεγκτής	ATmega328P	SAMD21 Cortex-M0+
Ταχύτητα ρολογιού	16 MHz	32.769 kHz (RTC), 48 MHz
Μνήμη Flash	32 kB	256 kB
EEPROM	1 kB	ΚΑΝΕΝΑ
SRAM	2 kB	32 kB
Ψηφιακές είσοδοι	14	22
Αναλογικές είσοδοι	6	8
Διαμόρφωση πλάτος παλμού	6	12
Συμβατότητα ασπίδας	ΝΑΙ	Ειδικές ή MKR2UNO πλακέτα

Εικόνα 2.2.8.1: Σύγκριση κλασσικού Arduino UNO, με Arduino MKR1000. Προσαρμόστηκε από: (© Arrow Electronics. ®, 2018)

3. Software (Λογισμικό)

Στο κεφάλαιο αυτό, παρουσιάζονται οι προγραμματιστικές δυνατότητες των Arduino, ο τρόπος λειτουργίας τους καθώς και οι εντολές και συναρτήσεις που χρησιμοποιούν. Για ευκολότερη κατανόηση, έχουν χρησιμοποιηθεί πολλαπλά παραδείγματα με εικόνες.

3.1 Arduino

Το Arduino περιέχει το ολοκληρωμένο περιβάλλον ανάπτυξης (integrated development environment - IDE), δηλαδή ένα λογισμικό το οποίο προσφέρει τη δυνατότητα δημιουργίας διάφορων προγραμμάτων στον υπολογιστή. Τα κύρια μέρη ενός IDE είναι κυρίως:

- **Επεξεργαστής πηγαίου κώδικα**: Στην πληροφορική πηγαίος κώδικας είτε απλά κώδικας αποτελεί μια σειρά εντολών σε μορφή κειμένου, που γράφονται από ένα χρήστη σε μια γλώσσα προγραμματισμού. Είναι ένα πρόγραμμα το οποίο μπορεί να αποθηκευτεί σε αρχεία κειμένου και πιο σπάνια σε βάσεις δεδομένων.

- **Μεταγλωττιστής:** Ο μεταγλωττιστής (compiler) είναι ένα πρόγραμμα το οποίο μετατρέπει το κώδικα που είναι γραμμένος σε μια γλώσσα προγραμματισμού σε μια άλλη γλώσσα προγραμματισμού. Κατά κύριο λόγο γίνεται η μετατροπή τη γλώσσας προγραμματισμού του χρήστη στη γλώσσα μηχανής του υπολογιστή. Για να λειτουργήσει σωστά ο μεταγλωττιστής στηρίζεται πάνω σε μια συγκεκριμένη δομή της οποίας τα μέρη είναι:

1. **Εμπρόσθιο τμήμα (front end):** Κυρίως αναλαμβάνει τον έλεγχο του προγράμματος, εμφανίζοντας μηνύματα σφάλματος αν τυχόν υπάρχουν συντακτικά λάθη. Επίσης μπορεί να περιέχει έναν προεπεξεργαστή ,που παράγει ένα νέο κώδικα κατάλληλο για μεταγλώττιση αντικαθιστώντας εντολές του κώδικα που εμφανίζονται συχνά με λέξεις κλειδιά. Για παράδειγμα τα σχόλια αντικαθίστανται με κενά, οι λέξεις με σύμβολα, όπως φαίνεται παρακάτω:

   Πριν την επεξεργασία

```
#ifdef_WIN32 // _WIN32 is defined by all Windows 32 compliers
 x = max (q+r, s+t);
```

   Μετά την επεξεργασία

```
#ifdef_WIN32
x = ((q + r) > (r + s)? (q+r) : (s+t));
```

a. Λεκτικός αναλυτής: Χωρίζει το πηγαίο κώδικα σε επιμέρους κομμάτια που ονομάζονται λεκτικές μονάδες διαχωρίζοντας λέξεις κλειδιά, τιμές του προγράμματος είτε εντολές όπως φαίνεται παρακάτω:

Η εντολή x = a + b * 2; πρώτα διαχωρίζεται

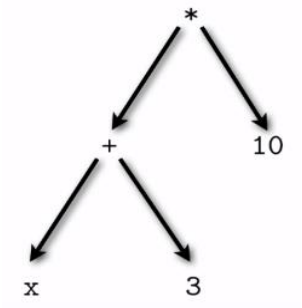
Ενδεικτικό Όνομα Λεκτικής Μονάδας	**Παράδειγμα Λεκτικών Μονάδων**
Αναγνωριστικό (identifier)	x, a, b
Τελεστής ανάθεσης	=
Τελεστής πρόσθεσης	+
Τελεστής πολλαπλασιασμού	*
Ακέραια τιμή	2
Τέλος εντολής	;

και μετατρέπεται.

[(identifier, x), (operator, =), (identifier, a), (operator, +), (identifier, b), (operator, *), (literal, 2), (separator, ;)]

b. Συντακτικός αναλυτής: Δέχεται ως είσοδο ένα πρόγραμμα με τη μορφή ακολουθίας λεκτικών μονάδων, ελέγχει αν το πρόγραμμα ανήκει στη ίδια γλώσσα με εκείνη του αναλυτή, σε περίπτωση συντακτικού λάθους ενημερώνει το χρήστη και παράγει το συντακτικό δένδρο που αντιστοιχεί στην ακολουθία εισόδου όπως φαίνεται παρακάτω:

Η εντολή
(x + 3) * 10
Μετατρέπεται σε



2. **Ενδιάμεσο τμήμα(middle end):** Πραγματοποιεί κάποιες βελτιστοποιήσεις με στόχο να καλυτερεύσει ο παραγόμενος κώδικας μηχανής. Τέτοιου είδους βελτιστοποιήσεις μπορούν να είναι:
   - Αφαίρεση άχρηστου κώδικα: Στη θεωρία των μεταγλωττιστών, η εξάλειψη του νεκρού κώδικα είναι μια βελτιστοποίηση που αφαιρείται κώδικας χωρίς να επηρεάζει τα αποτελέσματα του κυρίως προγράμματος. Αυτή η μέθοδος προσφέρει πλεονεκτήματα όπως η συρρίκνωση του μεγέθους του προγράμματος, αποφυγή εκτέλεσης άχρηστων λειτουργιών που καθυστερούν την εκτέλεση του προγράμματος. Έστω το παρακάτω πρόγραμμα σε γλώσσα C:

Παράδειγμα

```
int foo(void)
 {
   int a = 24;
   int b = 25; /* Assignment to dead variable */
   int c;
   c = a * 4;
   return c;
   b = 24; /* Unreachable code */
   return 0;
 }
```

Επεξήγηση

Η μεταβλητή b είναι μια τοπική μεταβλητή μέσα στο foo οπότε δε μπορεί να χρησιμοποιηθεί εκτός αυτού. Επίσης η τιμή της b, δηλαδή το 25, μετά την πρώτη ανάθεση δεν επανεμφανίζεται μέσα στο foo οπότε αυτή η μεταβλητή μπορεί να θεωρηθεί νεκρή και ο βελτιστοποιητής να απελευθερώσει χώρο στη μνήμη και να εξαλείψει την αρχικοποίησή της. Επίσης η πρώτη εντολή return εκτελείται ούτως ή άλλως χωρίς να προχωρά ο κώδικας στη μεταβλητή b η οποία είναι απρόσιτη και άρα μπορεί να αφαιρεθεί.

- Συνεχής αναδίπλωση (constant folding): Η συνεχής αναδίπλωση είναι μια διαδικασία αναγνώρισης και αξιολόγησης των σταθερών ενός προγράμματος κατά τον χρόνο σύνταξης, αντί να υπολογιστούν κατά τη διάρκεια που εκτελείται το πρόγραμμα. Έστω η παρακάτω έκφραση:

  y = 2 * 3 * 4;

  Αυτό το οποίο θα κάνει ο μεταγλωττιστής είναι να αναγνωρίσει τη συγκεκριμένη έκφραση και να αποθηκεύσει την τιμή της(στη προκειμένη περίπτωση 24) στη μεταβλητή στο χρόνο της σύνταξης, αντί να γίνουν οι υπολογισμοί πολλαπλασιασμού και ύστερα να αποθηκευτούν την ώρα της εκτέλεσης του προγράμματος. Επίσης μπορεί να έχει και εφαρμογή σε συμβολοσειρές. Για παράδειγμά:

  Η εντολή
  "giorgo" + "fuge"
  Μετατρέπεται σε
  "giorgofuge"

- Συνεχής διάδοση (constant propagation): Η συνεχής διάδοση είναι μια διαδικασία όπου αντικαθίστανται οι τιμές γνωστών σταθερών σε εκφράσεις κατά τον χρόνο της σύνταξης. Έστω η παρακάτω έκφραση:

Αρχική Μορφή
Int x = 14;
Int y = 7 – x / 2;
Return y * (28 / 14 +2);

Ενδιάμεση Μορφή
Int x = 14;
Int y = 7 – 14 / 2;
Return y * (28 / 14 +2);

Τελική Μορφή
Int x = 14;
Int y = 0;
Return 0;

- Συνδυασμός συνεχής αναδίπλωσης και διάδοσης: Η μέθοδος αυτή χρησιμοποιείται για να γίνουν περισσότερες μειώσεις και βελτιστοποιήσεις. Έστω η παρακάτω έκφραση:

Αρχικά
Int a = 30;
Int b = 9 – (a / 5);
Int c;
C = b * 4;
If (c > 10) {
C = c – 10;
}
Return c * (60 / a);

Ύστερα από συνεχή αναδίπλωση και διάδοση
Int a = 30;
Int b = 3
Int c;
C = b * 4;
If (c > 10) {
C = c – 10;
}
Return c * 2

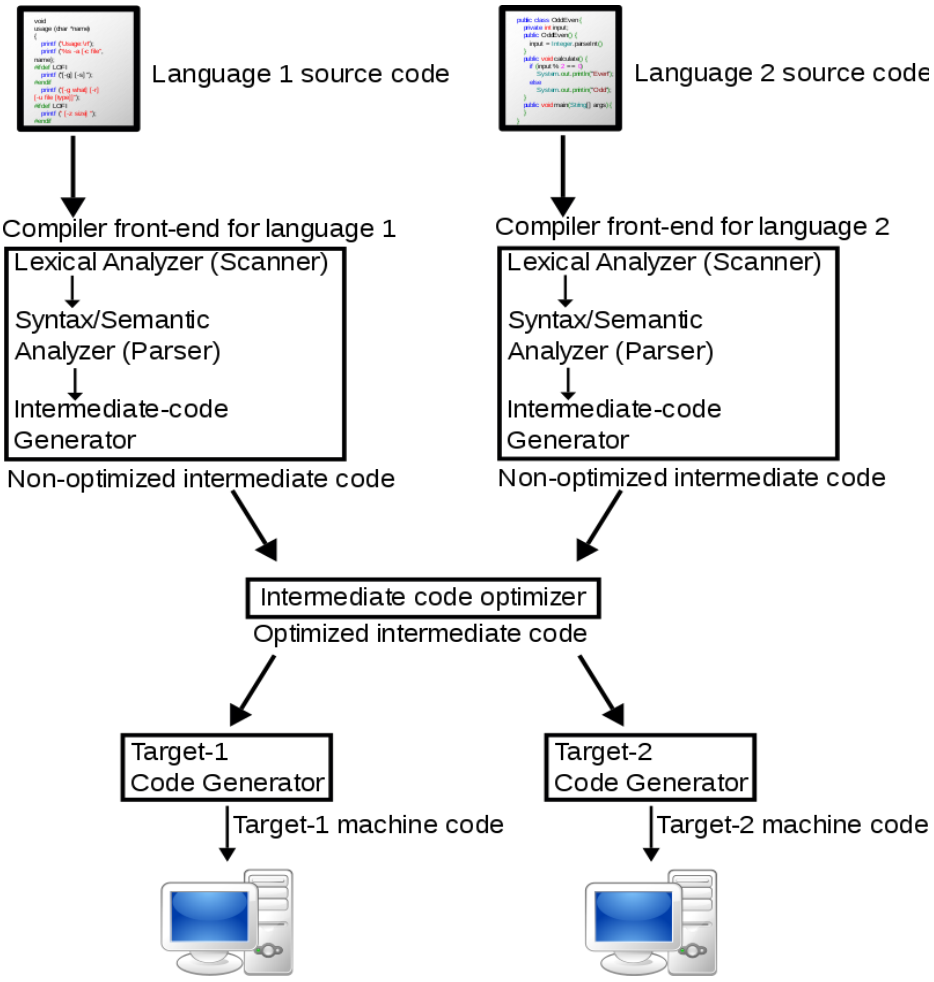
Εφαρμόζοντας ξανά την ίδια μέθοδο
Int a = 30;
Int b = 3
Int c;
C = 12;
If (true) {
C = 2;
}
Return c * 2

Τα a και b έχοντας απλοποιηθεί εφαρμόζεται εξάλειψη νεκρού κώδικα

```
Int c;
C = 12;
If (true) {
C = 2;
}
Return c * 2
```

Το ενδιάμεσο τμήμα παράγει στη συνέχεια μια άλλη ενδιάμεση αναπαράσταση, για το οπίσθιο τμήμα. Οι περισσότερες βελτιστοποιήσεις έχουν ήδη γίνει στο ενδιάμεσο τμήμα.

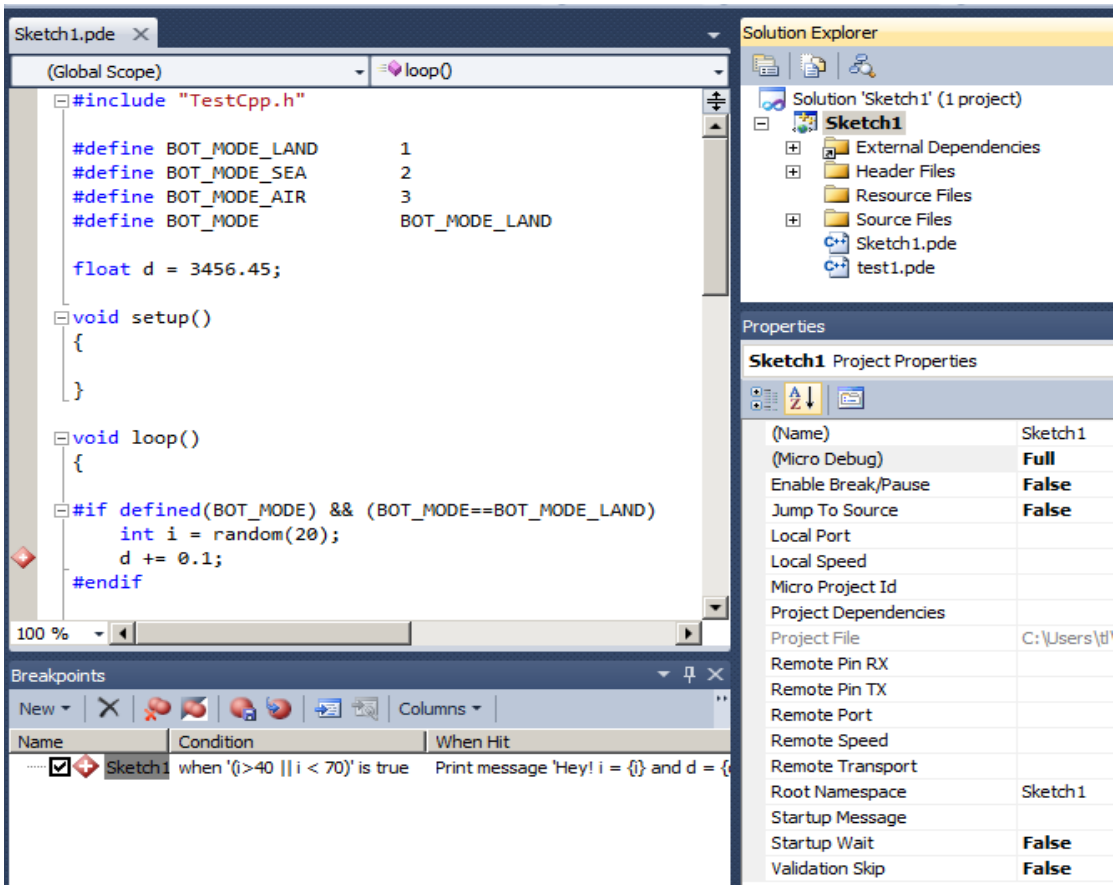
3. **Οπίσθιο τμήμα (back end):** Είναι υπεύθυνο για τη μετάφραση της ενδιάμεσης αναπαράστασης του ενδιάμεσου τμήματος σε γλώσσα μηχανής, συμβολική γλώσσα και γλώσσα προγραμματισμού (όπως η C). Κάθε εντολή της ενδιάμεσης αναπαράστασης αντιστοιχεί σε κάποιες συμβολικές εντολές. Οι βελτιστοποιήσεις που γίνονται εξαρτώνται από την αρχιτεκτονική της CPU του υπολογιστή. Αυτό αφορά τις δυνατότητες που παραχωρεί ο υπολογιστής σε χώρους μνήμης όπου θα αποθηκεύονται οι μεταβλητές.



Εικόνα 3.1.1: Παράδειγμα Οπίσθιου τμήματος. Προσαρμόστηκε από: (© Visual Micro ®, 2012)

- **<u>Εργαλεία αυτόματης παραγωγής κώδικα:</u>** Είναι τα διάφορα μέσα που προσφέρονται δημιουργία του κώδικα (εντολές, σύμβολα).

- **<u>Αποσφαλματωτής:</u>** Είναι ένα πρόγραμμα το οποίο εξετάζει την ορθότητα του προγράμματος με εργαλεία που διαθέτει ,όπως αναλυτή συμβόλων (symbol resolver), πλατφόρμα εντοπισμού σφαλμάτων, το οποίο έχει γραφεί στη γλώσσα προγραμματισμού. Εάν υπάρχει λάθος στο κώδικα επισημαίνει τα σημεία που υπάρχουν τα λάθη. Προσφέρει τη δυνατότητα το πρόγραμμα να εκτελείται βήμα προς βήμα ή να διακόπτεται σε κάποιο σημείο ώστε να διαπιστωθεί το λάθος.



Εικόνα 3.1.2: Αποσφαλματωτής. Προσαρμόστηκε από: (© Visual Micro ®, 2012)

- **<u>Βιβλιοθήκες:</u>** Είναι μια σειρά από έτοιμα μικρά προγράμματα ή αλλιώς υποπρογράμματα που συμβάλουν στην ανάπτυξη του λογισμικού.

- **<u>Σύστημα ελέγχου εκδόσεων:</u>** Αποτελεί έναν τρόπο αναβάθμισης των εκδόσεων κάθε λογισμικού που χρησιμοποιεί το Arduino, για την εκτέλεση οποιασδήποτε εντολής, με την νεότερη έκδοσή του.

- **<u>Γραφικό περιβάλλον χρήστη:</u>** Είναι το σύνολο από τα γραφικά στοιχεία που παρουσιάζονται στην οθόνη του υπολογιστή με στόχο την επικοινωνία χρήστη και υπολογιστή. Χαρακτηριστικό παράδειγμα αποτελεί το πληκτρολόγιο ή το ποντίκι.

Το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino βρίσκεται σε γλώσσα προγραμματισμού Java, η οποία βρίσκει χρήση σε πολλές πλατφόρμες και υποστηρίζει τη γλώσσα προγραμματισμού Processing και το σχέδιο Wiring. Το Processing είναι μια γλώσσα προγραμματισμού ανοιχτού κώδικα και γενικότερα ένα προγραμματιστικό περιβάλλον που αφορά τους χρήστες που θέλουν να προγραμματίσουν εικόνες, animation και ήχο. Το Wiring είναι ένα προγραμματιστικό περιβάλλον που αφορά τους χρήστες που θέλουν να προγραμματίσουν σε μορφή σχεδίου ηλεκτρονικά κυκλώματα.

3.2 Συναρτήσεις Arduino

- **Include**

Περιγραφή: Χρησιμοποιείται με στόχο να συμπεριληφθούν οι εξωτερικές βιβλιοθήκες στο κυρίως πρόγραμμα (sketch). Οι βιβλιοθήκες περιέχουν μικρότερα εκτελέσιμα προγράμματα, δηλαδή συναρτήσεις που διευκολύνουν το χρήστη τόσο στην εξοικονόμηση χρόνου όσο και χώρου για το κυρίως πρόγραμμα.

Σύνταξη: #include < name of the library > ή #include "name of the library".

Παράμετροι: Η διαφορά των συμβόλων για να συμπεριληφθεί η βιβλιοθήκη αφορά την αναζήτηση της τοποθεσίας των αρχείων. Τα < > υποδηλώνουν πως το αρχείο είναι τοπικό (local), ενώ τα " " υποδηλώνουν πως το αρχείο βρίσκεται σε συγκεκριμένη τοποθεσία.

Παράδειγμα:

Complete_Program §

```
#include <OneWire.h> // Γίνεται αναγνώριση και επικοινωνεί ο αισθητήρας με το Arduino
#include "DHT.h"     // Πρόγραμμα για τη μεταρατροπή θερμοκρασίας από τον αισθητήρα
```

Εικόνα 3.2.1: Παράδειγμα include. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

- **Define**

Περιγραφή: Χρησιμοποιείται για να δώσει ένα όνομα σε μια σταθερή τιμή πριν συνταχθεί το κυρίως πρόγραμμα. Οι σταθερές τιμές δε καταλαμβάνουν καθόλου χώρο μνήμης στον μικροελεγκτή.

Σύνταξη: #define constant name value

Παράμετροι:

- constant name: Όνομα της μεταβλητής.
- value: Τιμή που αντικαθιστά την μεταβλητή.

Παράδειγμα:

```
#define DHTPIN 5 // Ο μεταγλωττιστής θα αντικαταστήσει κάθε αναφορά του DHTPIN με την τιμή 5
```

Εικόνα 3.2.2: Παράδειγμα define. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

- **Serial**

Το Arduino περιέχει ένα πλήθος συναρτήσεων που στόχος τους είναι η σειριακή επικοινωνία μεταξύ του μικροελεγκτή και άλλων συσκευών. Η συνάρτηση που χρησιμοποιείται για να υποδηλώσει την σειριακή επικοινωνία καλείται Serial και χωρίζεται σε διάφορες κατηγορίες που είναι:

1. **Serial.begin**

Περιγραφή: Καθορίζει τον ρυθμό μετάδοσης των δεδομένων σε bits/sec για τη σειριακή μετάδοση. Οι πιο συχνά χρησιμοποιούμενες τιμές είναι: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 57600, 115200.

Σύνταξη: Serial.begin(speed)

Παράμετροι: Στο speed αντικαθίστανται οι παραπάνω αριθμοί υποδηλώνοντας τον ρυθμό μετάδοσης.

Παράδειγμα:

```
void setup() {
Serial.begin(9600); // Ρυθμός μετάδοσης 9600 bits/sec
}
```

Εικόνα 3.2.3: Παράδειγμα Serial.begin. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

2. **Serial.print**

Περιγραφή: Επιτυγχάνεται η αντιστοίχιση των δεδομένων στην μορφή ASCII κειμένου. Πιο συγκεκριμένα στους ακέραιους αριθμούς αντιστοιχίζεται ένας χαρακτήρας ASCII για κάθε ψηφίο. Στους δεκαδικούς ακολουθείται η ίδια διαδικασία με τη διαφορά πως εκτυπώνονται μόνο τα 2 πρώτα δεκαδικά ψηφία. Οι χαρακτήρες και οι συμβολοσειρές εκτυπώνονται όπως ήταν αρχικά.

Σύνταξη: Serial.print(val,format)

Παράμετροι:

- val: Καθορίζει τη τιμή που θα τυπωθεί.
- format: Όταν ο αριθμός είναι ακέραιος καθορίζει τη βάση του αριθμητικού συστήματος ενώ στους δεκαδικούς αριθμούς υποδηλώνει το πλήθος των δεκαδικών ψηφίων.

Παράδειγμα:

```
void loop() {
  Serial.print(40,BIN); // Εκτυπώνει την τιμή 40 σε δυαδική μορφή: 101000
  Serial.print(40,OCT); // Εκτυπώνει την τιμή 40 σε οκταδική μορφή: 50
  Serial.print(40,DEC); // Εκτυπώνει την τιμή 40 σε δεκαδική μορφή: 40
  Serial.print(40,HEX); // Εκτυπώνει την τιμή 40 σε δεκαεξαδική μορφή: 28
}
```

Εικόνα 3.2.4: Παράδειγμα Serial.print. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

3. Serial.println

Όμοια σύνταξη και περιγραφή με τη διαφορά πως όταν τυπώνεται η εντολή γίνεται αυτόματα αλλαγή γραμμής.

Παράδειγμα:

```
void loop() {
  Serial.println("Light Detected"); // Εκτυπώνει τη φράση Light Detected
}
```

Εικόνα 3.2.5: Παράδειγμα Serial.println. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

- **pinMode**

Περιγραφή: Καθορίζει εάν κάποιος ακροδέκτης του μικροελεγκτή θα λειτουργεί ως είσοδος ή έξοδος.

Σύνταξη: pinMode(pin,mode)

Παράμετροι:

- pin: Ορίζεται ο αριθμός του ακροδέκτη.
- mode: INPUT λειτουργεί ως είσοδος, OUTPUT λειτουργεί ως έξοδος και INPUT_PULLUP λειτουργεί ως είσοδος με εσωτερική αντίσταση Pull-up.

Παράδειγμα:

```
void setup() {
  Serial.begin(9600);
  pinMode(0, OUTPUT);  // Ορίζει τον ακροδέκτη 0 ως έξοδο
  pinMode(A3, INPUT);  // Ορίζει τον ακροδέκτη A3 ως είσοδο
 }
```

Εικόνα 3.2.6: Παράδειγμα pinMode. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

- **int**

Περιγραφή: Αποθηκεύει σε μία μεταβλητή έναν αριθμό.

Σύνταξη: inta var = val

Παράμετροι:

- var: Ορίζεται το όνομα της μεταβλητής.
- val: Ορίζεται η τιμή που ανατίθεται στη μεταβλητή.

Παράδειγμα:

```
int LedHeat = 4; // Η μεταβλητή LedHeat παίρνει την τιμή 4
int value = 0;  //  Η μεταβλητή value παίρνει την τιμή 0
```

Εικόνα 3.2.7: Παράδειγμα int. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

- **delay**

Περιγραφή: σταματά την εκτέλεση του προγράμματος ανάλογα με το χρόνο που θα οριστεί από τη παράμετρο.

Σύνταξη: delay(ms)

Παράμετροι: Ο αριθμός των millisecond που θα σταματήσει η εκτέλεση του προγράμματος.

Παράδειγμα:

```
delay(10); // Γίνεται καθυστέρηση 10 ms
```

Εικόνα 3.2.8: Παράδειγμα delay. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

- **digitalWrite**

Περιγραφή: Αναθέτει σε έναν ακροδέκτη του μικροελεγκτή την τιμή HIGH (5V) ή LOW (0V).

Σύνταξη: digitalWrite(pin,value)

Παράμετροι:

- pin: Ο αριθμός του ακροδέκτη όπου θα γίνει η εγγραφή της τιμής.
- value: Τιμή High για 5V και LOW για 0V.

Παράδειγμα:

```
void setup()
{
  pinMode(3, OUTPUT);           // Ορίζει τον ακροδέκτη 3 ως έξοδο
}

void loop()
{
  digitalWrite(3, HIGH);        // Ο ακροδέκτης 3 πάιρνει την τιμή 5V
  delay(2000);                  // Καθυστέρηση 2 sec
  digitalWrite(3, LOW);         // Ο ακροδέκτης 3 πάιρνει την τιμή 0V
  delay(2000);                  // Καθυστέρηση 2 sec
}
```

Εικόνα 3.2.9: Παράδειγμα digitalWrite. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

- **digitalRead**

Περιγραφή: Ανάγνωση της τιμής από ένα συγκεκριμένο ακροδέκτη. Ύστερα επιστρέφει τη τιμή HIGH (5V) ή LOW (0V).

Σύνταξη: digitalRead(pin)

Παράμετροι:

- pin: Ο αριθμός του ακροδέκτη όπου θα γίνει η ανάγνωση της τιμής.

Παράδειγμα:

```
int ledPin = 10;    // LED συνδεδεμένο στον ακροδέκτη 10
int inPin = 4;      // pushbutton συνδεδεμένο στον ψηφιακό ακροδέκτη 7
int val = 0;        // αρχικοποίηση της μεταβλητής που θα διαβαστεί

void setup()
{
  pinMode(ledPin, OUTPUT);      // Ορίζει τον ακροδέκτη 10 ως έξοδο
  pinMode(inPin, INPUT);        // Ορίζει τον ακροδέκτη 4 ως είσοδο
}

void loop()
{
  val = digitalRead(inPin);     // Διαβάζει την τιμή του ακροδέκτη 4, δηλαδή 0 ή 5V
  digitalWrite(ledPin, val);    // Το Led παίρνει την ίδια τιμή με το pushbutton. Αν το Pushbutton είναι πατημένο,
                                // έχοντας τιμή 5V τότε και το Led θα έχει τιμή 5V, δηλαδή αναμένο.
}
```

Εικόνα 3.2.10: Παράδειγμα digitalRead. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

- **analogRead**

Περιγραφή: Γίνεται ανάγνωση της τιμής σε ένα από τους 6 αναλογικούς ακροδέκτες εισόδου του μικροελεγκτή. Οι είσοδοι μετατρέπουν την τάση που διαβάζουν από 0 έως 5V σε ακέραιες τιμές από 0 έως 1023. Αυτό καθορίζει και τη διακριτική ικανότητα του μικροελεγκτή, δηλαδή τα αποτελέσματα των μετρήσεων είναι 5V/1024 ή αλλιώς 4.9mV ανά μονάδα. Εάν η αναλογική είσοδος δεν είναι συνδεδεμένη σε κάποια τάση επιστρέφει τυχαίες τιμές.

Σύνταξη: analogRead(pin)

Παράμετροι:

- pin: Ο αριθμός του ακροδέκτη όπου θα γίνει η ανάγνωση της τιμής.

Παράδειγμα:

```
int LDR_Pin = A6; // Η μεταβλητή LDR_Pin παίρνει την τιμή του αναλογικού ακροδέκτη A6
void setup() {
  Serial.begin(9600); // Ρυθμός μετάδοσης δεδομένων 9600 bits/sec
  pinMode(0, OUTPUT); // Ορίζει τον ψηφιακό ακροδέκτη 0 ως έξοδο
}
void loop() {
  int LDRReading = analogRead(LDR_Pin);  // Αποθηκεύεται στη μεταβλητή LDRReading η τιμή που διαβάζεται από τον ακροδέκτη A6
  if (LDRReading < 800) {               // Συγκρίνει εάν η τιμή που έχει διαβαστεί είναι μικρότερη από 800
    Serial.println("Light Detected");   // Τυπώνει τη φράση Light Detected
    Serial.println(LDRReading);         // Τυπώνει την τιμή του LDRReading
    Serial.println();                   // Αλλαγή γραμμής
    digitalWrite(0, HIGH);              // Ο ακροδέκτης 0 θα πάρει την τιμή 5V
  }
  else {                                 // Συγκρίνει εάν η τιμή που έχει διαβαστεί είναι μεγαλύτερη από 800 και άνω
    Serial.println("The room is dark");  // Τυπώνει τη φράση Light Detected
    Serial.println(LDRReading);         // Τυπώνει την τιμή του LDRReading
    Serial.println();                   // Αλλαγή γραμμής
    digitalWrite(0, LOW);               // Ο ακροδέκτης 0 θα πάρει την τιμή 0V
  }
}
```

Εικόνα 3.2.11: Παράδειγμα analogRead. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

- **if**

Περιγραφή: Ελέγχει κάποια συνθήκη ή ένα σύνολο συνθηκών τις οποίες θα εκτελεί αν είναι αληθής (true) ή δε θα εκτελεί αν είναι ψευδής (false).

Σύνταξη:
```
if (condition1)
{
  // do a thing A
 }
 else if (condition2)
{
   // do a thing B
 }
 Else
{
  // do a thing B
 }
```

Παράμετροι:

- if: έλεγχος μια συνθήκης Α.
- else if: έλεγχος μια συνθήκης Β. Όταν υπάρχουν συνήθως περισσότερες από δύο συνθήκες χρησιμοποιούμε αυτή τη συνάρτηση.
- else: έλεγχος μια συνθήκης Γ. Όταν υπάρχουν συνήθως δύο συνθήκες χρησιμοποιούμε αυτή τη συνάρτηση.

Παράδειγμα:

```
int val;          //Αρχικοποίηση της μεταβλητής val
int tempPin = A1;  // Η μεταβλητή παίρνει την τιμή του αναλογικού ακροδέκτη A1
int LedPin = 5;    // Led συνδεδεμένο στον ακροδέκτη 5

void setup() {
pinMode(tempPin,INPUT); // Ορίζει τον ακροδέκτη A1 ως είσοδο
pinMode(LedPin,OUTPUT); // Ορίζει τον ακροδέκτη 5 ως έξοδο
}

void loop() {
val=analogRead(tempPin);               // Ανάγνωση της τιμής από τον ακροδέκτη A1 και αποθήκευση στη μεταβλητή val
float cel= ((((val/1024)*5)-0.22)*100);  // Τύπος μετατροπής σε θερμοκρασία. Αποθήκευση στην μεταβλητή cel τη τιμή της θερμοκρασίας με δεκαδικά ψηφία.
if (cel > 25)                          // Συγκρίνει αν η τιμή της θερμοκρασίας είναι πάνω από 25 βαθμούς
{
  digitalWrite(LedPin,HIGH);          // Ο ακροδέκτης 5 παίρνει την τιμή 5V, δηλαδή το Led ανάβει
}
else                                  // Συγκρίνει αν η τιμή της θερμοκρασίας είναι από 25 βαθμούς και κάτω
{
  digitalWrite(LedPin,LOW);         // Ο ακροδέκτης 5 παίρνει την τιμή 0V, δηλαδή το Led σβήνει
}
delay(3000);                        // Καθυστέρηση 3 sec
}
```

Εικόνα 3.2.12: Παράδειγμα if. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

- **switch**

Περιγραφή: Παρόμοια με την if, συγκρίνει τη τιμή μιας μεταβλητής με τη τιμή μιας μεταβλητής μέσα σε μια συνθήκη. Όσο αυτές οι δύο ταιριάζουν το πρόγραμμα εκτελείται κανονικά.

Σύνταξη: switch (var) {
case 1
// do something when var is equal to 1
break
case 2
// do something when var is equal to 2
break
case 3
// do something when var is equal to
break
default
}

Παράμετροι:

- var: Η αρχική μεταβλητή η οποία συγκρίνεται με εκείνες μέσα στις συνθήκες.
- case: Σταθερές προς σύγκριση.
- break: Χρησιμοποιείται για να σταματήσει η επανάληψη του κώδικα.

Παράδειγμα:

```
int val;           // Αρχικοποίηση της μεταβλητής val
int tempPin = A1;  // Η μεταβλητή παίρνει την τιμή του αναλογικού ακροδέκτη A1
int LedPin = 5;    // Led συνδεδεμένο στον ακροδέκτη 5

void setup() {
pinMode(tempPin,INPUT); // Ορίζει τον ακροδέκτη A1 ως είσοδο
pinMode(LedPin,OUTPUT); // Ορίζει τον ακροδέκτη 5 ως έξοδο
}

void loop() {
val=analogRead(tempPin);               // Ανάγνωση της τιμής από τον ακροδέκτη A1 και αποθήκευση στη μεταβλητή val
float cel= ((((val/1024)*5)-0.22)*100); // Τύπος μετατροπής σε θερμοκρασία. Αποθήκευση στην μεταβλητή cel τη τιμή της θερμοκρασίας με δεκαδικά ψηφία.
switch (cel) {                         // Μεταβλητή με την οποία θα ελεγχούν οι συνθήκες
  case 30:                            // Εξετάζει εάν η θερμοκρασία είναι 30 βαθμοί
  digitalWrite(LedPin,HIGH);           // Ο ακροδέκτης 5 παίρνει την τιμή 5V, δηλαδή το Led ανάβει
  delay(3000);                        // Καθυστέρηση 3 sec
  break;                              // Διακοπή της επανάληψης
  case 20:                            // Εξετάζει εάν η θερμοκρασία είναι 20 βαθμοί
  digitalWrite(LedPin,LOW);           // Ο ακροδέκτης 5 παίρνει την τιμή 0V, δηλαδή το Led σβήνει
  delay(3000);                        // Καθυστέρηση 3 sec
  break;                              // Διακοπή της επανάληψης
}
```

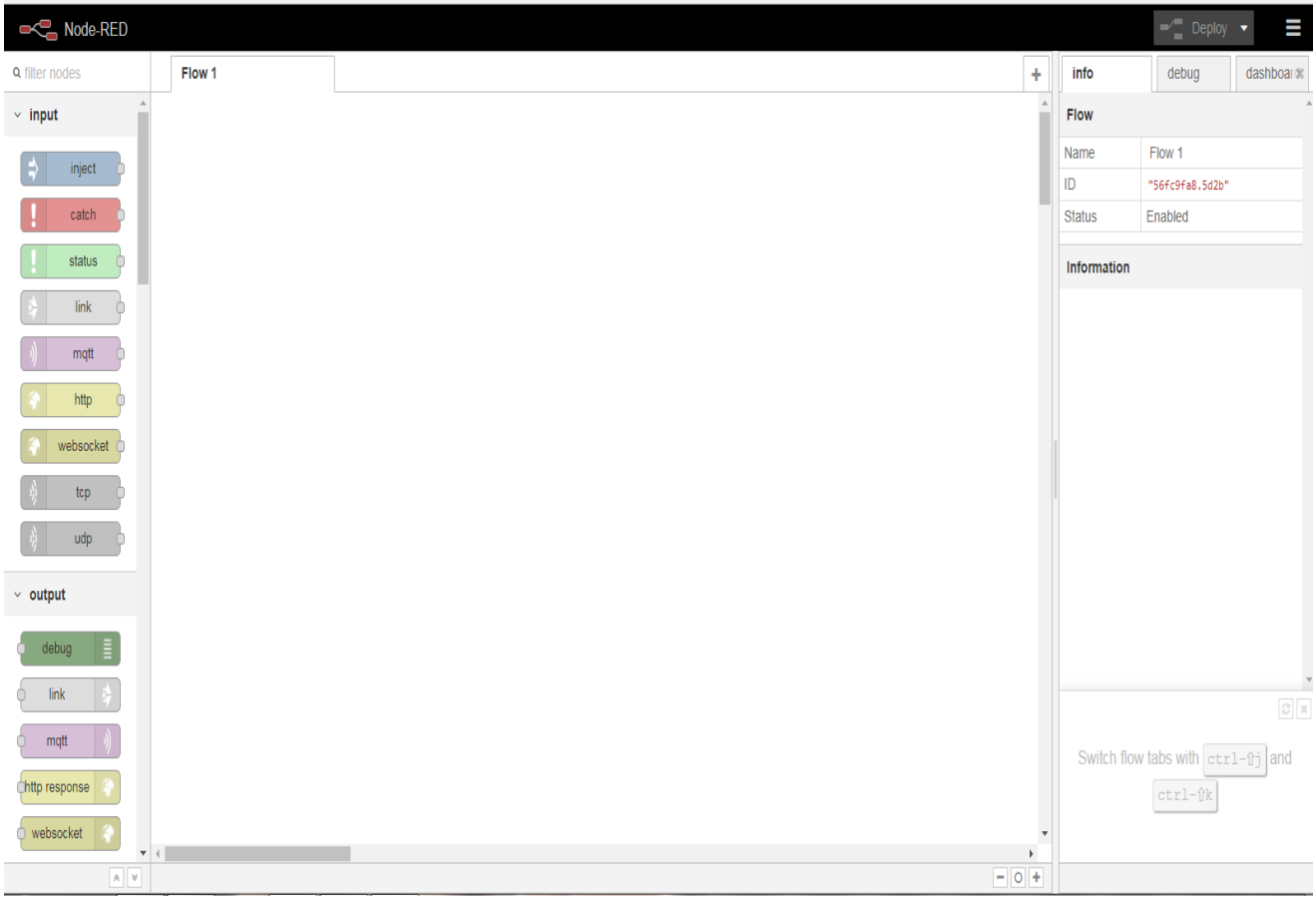
Εικόνα 3.2.13: Παράδειγμα switch. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

4. Node – Red

Στο συγκεκριμένο κεφάλαιο γίνεται ανάλυση των δυνατοτήτων της Node – Red, καθώς και προβολή των χαρακτηριστικών της, αλλά και παρουσιάζεται βήμα – βήμα ο τρόπος σύνδεσής της με έναν ηλεκτρονικό υπολογιστή. Επιπλέον, προβάλλεται και το τελικό συνδυαστικό πρόγραμμα, μεταξύ Arduino και Node – Red.

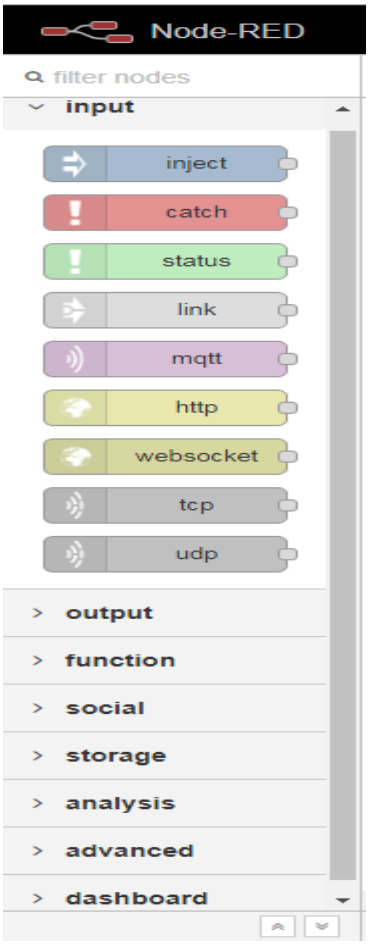
4.1 Γνωριμία με την Node - Red

Η Node Red είναι ένα προγραμματιστικό εργαλείο, που εξασφαλίζει την επικοινωνία hardware συσκευών και online υπηρεσιών με ποικίλους τρόπους. Επινοήθηκε από τον J.Paul Morrison στη δεκαετία του 1970 και είναι ένα είδος προγραμματισμού βάση ροής (flow based programming). Στην Node Red υπάρχει ένα δίκτυο μαύρων κουτιών ή κόμβοι (nodes), όπως ονομάζονται αλλιώς στην Node Red. Ο κάθε κόμβος έχει μια συγκεκριμένη λειτουργία, όπου δέχεται δεδομένα, τα επεξεργάζεται, ύστερα τα μεταφέρει σε άλλους κόμβους και η ροή των δεδομένων μεταξύ των κόμβων αλλά και η επικοινωνία τους, επιτυγχάνεται εξαιτίας του διαδικτύου. Αυτή είναι και η έννοια του προγραμματισμού βάση ροής (flow based programming). Η πρόσβαση στο προγραμματιστικό περιβάλλον της Node Red γίνεται μέσω ενός φυλλομετρητή. Τα κύρια μέρη που απαρτίζουν το προγραμματιστικό περιβάλλον της Node Red είναι:



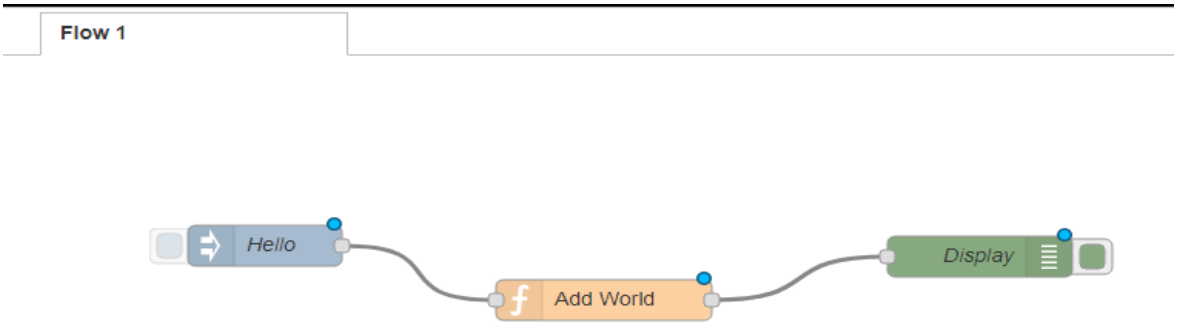
Εικόνα 4.1.1: Γενική εικόνα Node - Red. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

- Παλέτα (Pallet): Βρίσκεται αριστερά στον editor και περιέχει ένα σύνολο κόμβων (nodes), καθένας για συγκεκριμένη εργασία.



Εικόνα 4.1.2: Παλέτα Node - Red. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

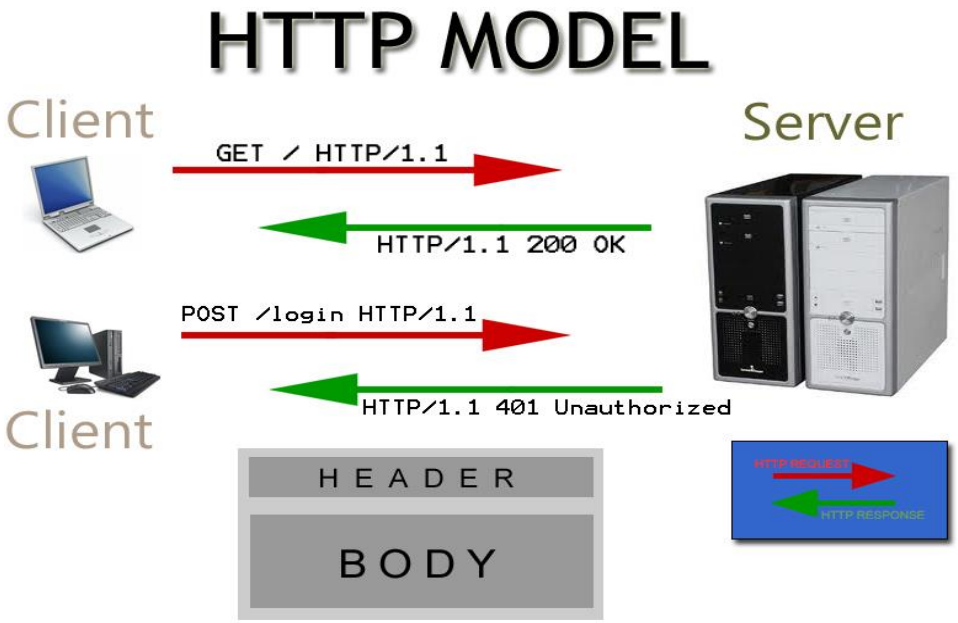
- Ροή (Flow): Καλείται το σύνολο των κόμβων, που ανασύρονται από την παλέτα στο κυρίως πρόγραμμα, οι οποίοι συνδέονται μεταξύ τους και εκτελούν την εργασία που τους ανατίθενται. Μία ροή για να ξεκινήσει χρειάζεται μια είσοδο (input) και για να τελειώσει μια έξοδο (output).



Εικόνα 4.1.3: Παράδειγμα ροής Node - Red. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

Ως είσοδοι και έξοδοι μιας ροής μπορούν να είναι:

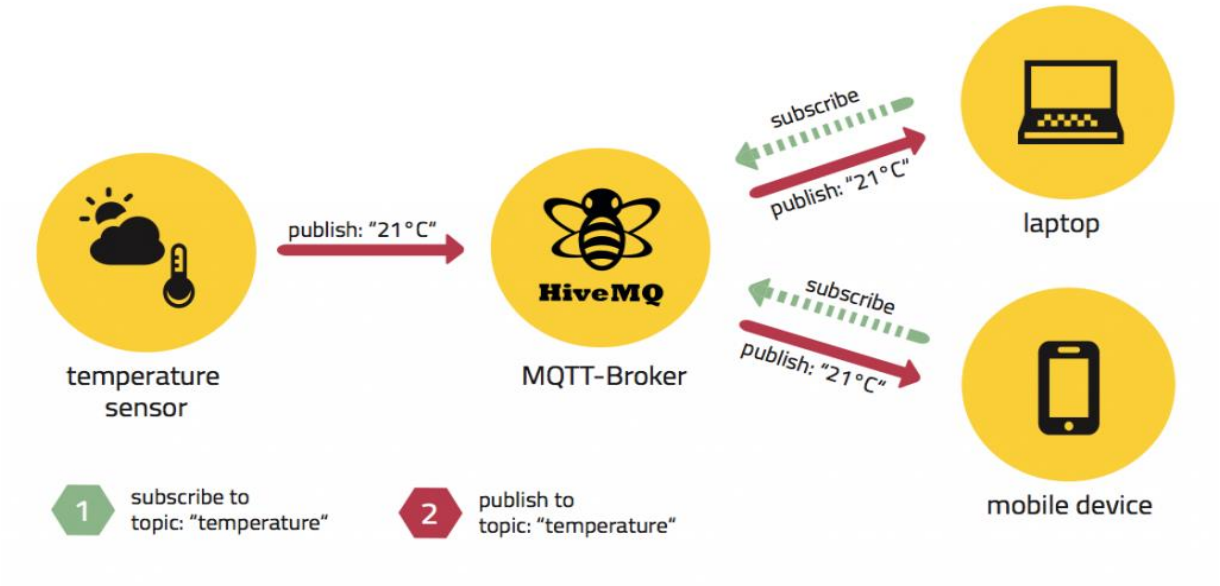
i. **Http:** Δημιουργείται ένα τελικό σημείο, δηλαδή μια διεύθυνση URL στην οποία υπάρχει επικοινωνία μεταξύ του πελάτη (Client) και του διακομιστή (Server). Γενικότερα το πρωτόκολλο μεταφοράς υπερκειμένου (http) λειτουργεί ως ένα πρωτόκολλο αίτησης απάντησης μεταξύ των δύο. Για παράδειγμα, ως πελάτης μπορεί να λειτουργήσει ένας αισθητήρας πάνω στο Arduino το οποίο υποβάλλει αίτημα στο διακομιστή, δηλαδή σε μια ιστοσελίδα URL που έχει δημιουργηθεί από τη Node Red. Έτσι επιτυγχάνεται η αποστολή δεδομένων από τον αισθητήρα στο διακομιστή. Οι δύο πιο γνωστές μέθοδοι αίτησης - απάντησης για την επικοινωνία μεταξύ πελάτη και διακομιστή είναι η get και η post. Με τη μέθοδο get ο πελάτης αιτείται από το διακομιστή τη λήψη εγγράφων και οι παράμετροι, δηλαδή οι πληροφορίες που ζητά, είναι κωδικοποιημένες στο URL και συνήθως περιγράφουν πιο έγγραφο ψάχνουν ή σε ποια σελίδα είναι. Η μέθοδος post χρησιμοποιείται για να στέλνει πληροφορίες, ενημερώνοντας τον Server και οι παράμετροι βρίσκονται στο κυρίως σώμα.



Εικόνα 4.1.4: Μοντέλο HTTP. Προσαρμόστηκε από: (© MediaWiki ®, 2012)

ii. **Mqtt:** Είναι ένα σύστημα δημοσίευσης – ανταλλαγής (publish – subscribe) ή αλλιώς πρωτόκολλο μηνυμάτων (messaging protocol) που δίνει τη δυνατότητα στο πελάτη (Client) να δημοσιεύει και να λαμβάνει μηνύματα και καθιστά πολύ εύκολη την επικοινωνία μεταξύ πολλαπλών συσκευών. Προσφέρει τη δυνατότητα αποστολής κάποιας εντολής, ώστε να ελεγχθεί μια έξοδος, όπως ο φωτισμός ενός Led, είτε να διαβαστούν δεδομένα, όπως οι τιμές ενός αισθητήρα και να δημοσιευθούν. Το κομμάτι που παρεμβάλλεται μεταξύ των πελατών και των συσκευών είναι ο μεσίτης και είναι υπεύθυνος για την ανταλλαγή μηνυμάτων μεταξύ τους. Αποτελεί δηλαδή τον κεντρικό

κόμβο επικοινωνίας. Όταν ο πελάτης δημοσιεύει ένα μήνυμα στο μεσίτη αυτό περιέχει και μία θεματολογία ή απλά ένα θέμα. Το θέμα προσδιορίζει που θα δημοσιευθεί το μήνυμα. Ο πελάτης που θέλει να λαμβάνει μηνύματα κάνει εγγραφή σε ένα συγκεκριμένο θέμα και ο μεσίτης παραδίδει όλα τα μηνύματα που ταιριάζουν στο συγκεκριμένο θέμα, στο πελάτη. Επομένως, αυτό που έχει σημασία για τον πελάτη είναι να γνωρίζει το θέμα, ώστε να λάβει τη πληροφορία που επιθυμεί.

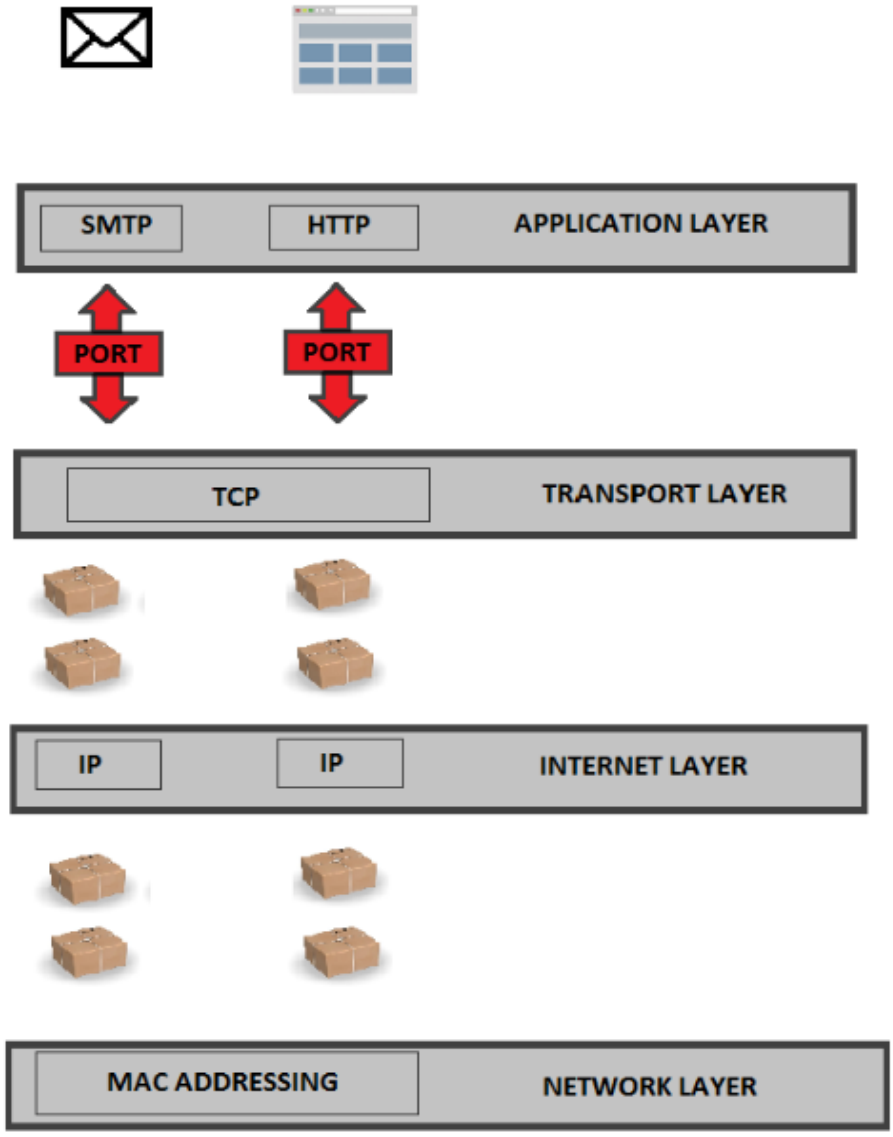


Εικόνα 4.1.5: Παράδειγμα συστήματος MQTT. Προσαρμόστηκε από: (© dc-square GmbH, 2018)

iii. **Tcp:** Είναι ένα πρωτόκολλο επικοινωνίας του διαδικτύου το οποίο ορίζει τη δρομολόγηση των δεδομένων, δηλαδή το πώς θα μεταφερθούν από το ένα μέρος στο άλλο με ασφαλή τρόπο, μέσω κάποιων επιπέδων. Έστω ότι ένας υπολογιστής από την Ελλάδα θέλει να στείλει δεδομένα σε ένα υπολογιστή στην Βοστώνη. Το πρώτο επίπεδο που ονομάζεται επίπεδο εφαρμογής, αφού λάβει τα δεδομένα από τον υπολογιστή και τα διαχωρίσει ανάλογα με το είδος τους, αν είναι ένα email σε smtp ή περιηγητής σε http, επικοινωνεί με το δεύτερο επίπεδο που ονομάζεται επίπεδο μεταφοράς μέσω κάποιων θυρών. Κάθε θύρα έχει δικό της αριθμό ώστε να γνωρίζει το τύπο των δεδομένων και τη προέλευση τους. Για παράδειγμα, από τις θύρες με αριθμό 80 περνά η πληροφορία τύπου http, δηλαδή ενός περιηγητή. Ύστερα το επίπεδο μεταφοράς διαχωρίζει τη πληροφορία σε διάφορα κομμάτια, πακέτα δεδομένων όπως ονομάζονται, έχοντας το καθένα το δικό του. Κάθε πακέτο ακολουθεί ξεχωριστά διαφορετική διαδρομή για να φτάσει στον τελικό υπολογιστή. Οι ονομασίες που δίνονται στα επιμέρους μέρη του χρησιμεύουν ώστε η πληροφορία να συναρμολογηθεί ξανά από τον υπολογιστή δέκτη όπως ήταν αρχικά, αλλά και για να γίνει έλεγχος κομματιών που τυχών λείπουν με στόχο να επαναποσταλθούν. Στη συνέχεια στο τρίτο επίπεδο που ονομάζεται

επίπεδο Internet, τα πακέτα ταυτοποιούνται με ip διευθύνσεις οι οποίες υποδεικνύουν τη πηγή προέλευσης τους, δηλαδή στη περίπτωση μας τον υπολογιστή στην Ελλάδα, αλλά και τον τελικό προορισμό, δηλαδή τον υπολογιστή στη Βοστώνη. Τέλος, το τέταρτο επίπεδο που ονομάζεται επίπεδο διαδικτύου, προσδίδει στα πακέτα mac διευθύνσεις ώστε εκείνες να δρομολογηθούν στο τελικό φυσικό μέσο, που είναι ο υπολογιστής αποδέκτης σε αυτή τη περίπτωση.

Παρακάτω παρουσιάζεται ο μηχανισμός λειτουργίας του πρωτοκόλλου επικοινωνίας TCP.



Εικόνα 4.1.6: **Μηχανισμός λειτουργίας του πρωτοκόλλου επικοινωνίας TCP.** ***Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)***

Σε μια ροή τόσο οι είσοδοι όσο και οι έξοδοι μπορούν να είναι περισσότερες από μια και μεταξύ τους παρεμβάλλονται κόμβοι που επεξεργάζονται τα δεδομένα και μπορεί να είναι:

1. **Function:** Ο συγκεκριμένος κόμβος χρησιμοποιείται για τη γραφή κάποιου κώδικα. Έστω ότι ένας αισθητήρας στέλνει τις καταγραφές του σε μια ιστοσελίδα που έχει δημιουργηθεί από τη node red. Οι καταγραφές αυτές θα έχουν τη μορφή συμβολοσειρών. Εάν τοποθετήσουμε το κόμβο function μπορούμε να γράψουμε κώδικα που θα μετατρέψει τη συμβολοσειρά σε αριθμούς.

2. **Switch:** Σε αυτό το κόμβο φτάνουν μηνύματα, πληροφορίες και ανάλογα με το αν τηρούν τις προϋποθέσεις που ορίζονται μέσα σε αυτόν προωθούνται περαιτέρω ή σταματούν. Μια εφαρμογή αυτού του κόμβου είναι να χρησιμοποιηθεί ως έλεγχος κατωφλίου. Έστω ότι ένας αισθητήρας θερμοκρασίες στέλνει πληροφορίες για την θερμοκρασία του περιβάλλοντος. Μέσα στον κόμβο μπορεί να οριστεί μια συνθήκη ώστε οι θερμοκρασίες που ξεπερνούν τους 26 βαθμούς να προχωρούν σε μια τελική έξοδο και οι υπόλοιπες να αποκόπτονται.

3. **Trigger:** Αυτός ο κόμβος στέλνει ένα μήνυμα αν δεν ληφθεί τίποτα μέσα σε ένα προκαθορισμένο χρονικό διάστημα ή μέχρι να γίνει κάποια επαναφορά. Μια χρήσιμη εφαρμογή αυτού του κόμβου είναι το αναβόσμησμα ενός led. Από προεπιλογή όταν δέχεται ένα μήνυμα στέλνει ένα μήνυμα με αριθμό 1, περιμένει 250ms και στέλνει ένα δεύτερο μήνυμα 0. Επομένως καθορίζοντας το χρόνο που θα περιμένει ελέγχουμε πότε θα αναβοσβήνει το led, ανάβοντας με το μήνυμα 1 και σβήνοντας με το μήνυμα 0.

4. **Delay:** Αυτός ο κόμβος παρέχει τη δυνατότητα να καθυστερούν τα μηνύματα που περνάνε μέσα από αυτόν, για χρονικό διάστημα που ορίζεται μέσα στη συνάρτηση και μπορεί να είναι μερικά δευτερόλεπτα έως ώρες ή μέρες.

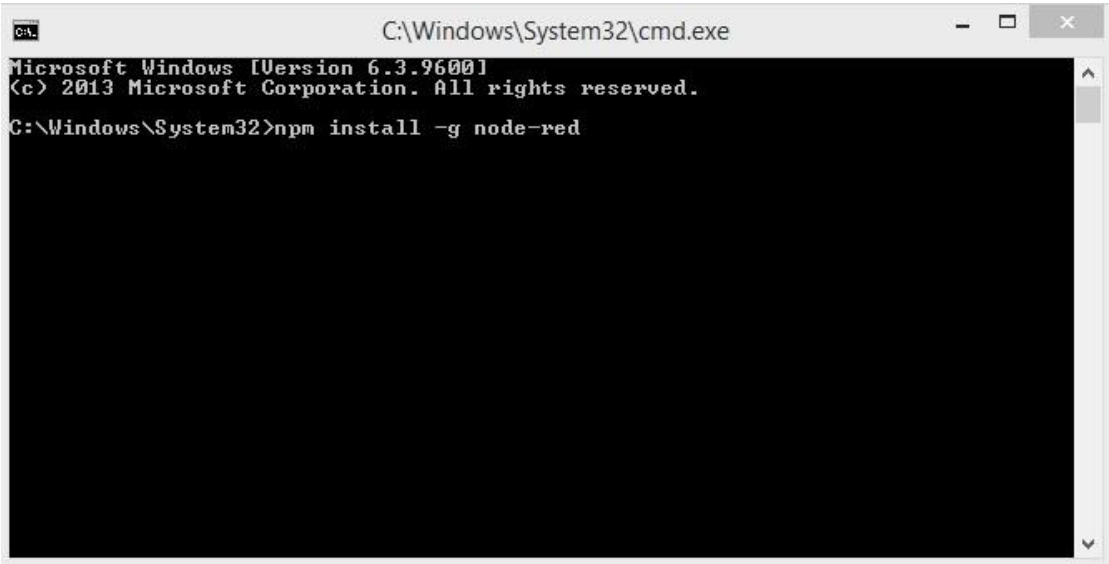
5. **Email:** Αυτός ο κόμβος χρησιμοποιείται για να προωθεί email στους διάφορους λογαριασμούς που επιθυμεί ο χρήστης.

6. **File:** Αυτός ο κόμβος έχεις δύο χρήσεις. Ως πρώτη δέχεται ως είσοδο το όνομα ενός αρχείου, το διαβάζει παράγοντας στην έξοδο του συμβολοσειρές ή δυαδικές ακολουθίες. Ως δεύτερη χρήση μπορεί να προσθέσει πληροφορίες σε ένα υπάρχον αρχείο είτε να το διαβάσει.

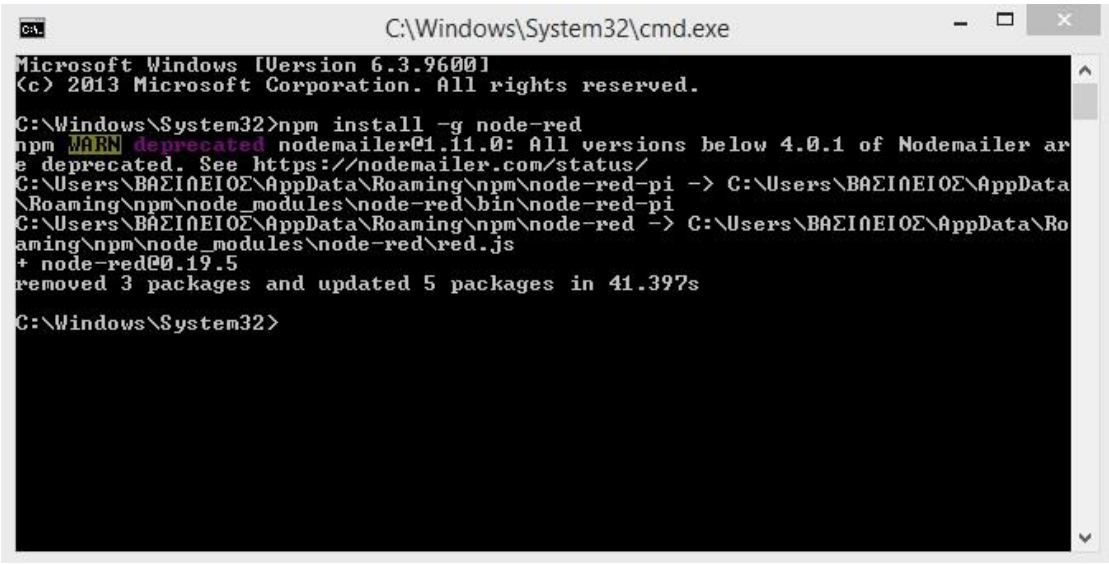
7. **Audio Out:** Αυτός ο κόμβος μπορεί να αναπαράγει ήχους ή να μετατρέψει ένα κείμενο σε ήχο.

4.2 Σύνδεση Node – Red

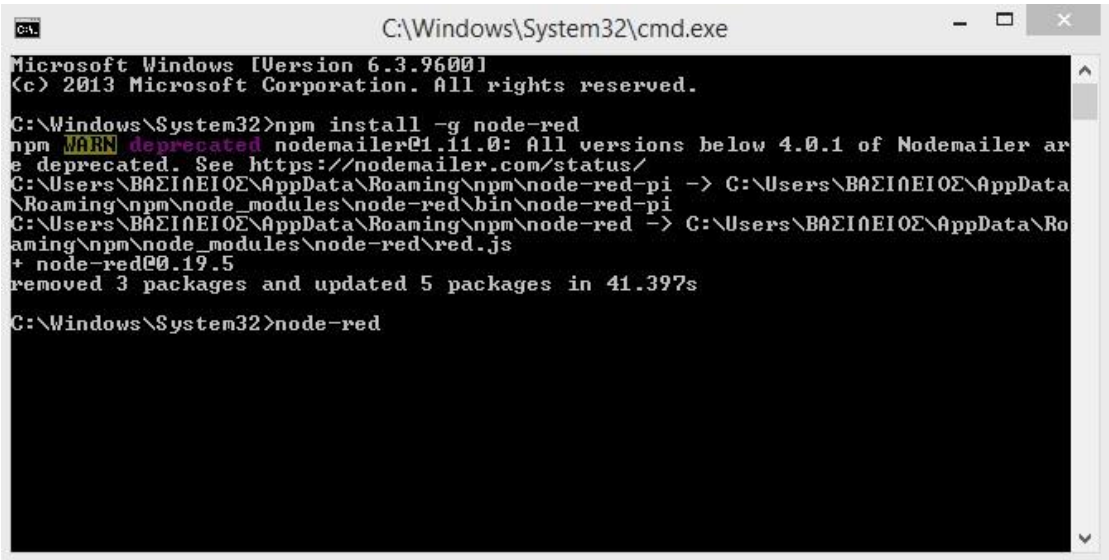
Για την σύνδεση της Node – Red με τον ηλεκτρονικό υπολογιστή στον οποίο θα εγκαταστήσουμε την συσκευή, είναι απαραίτητες η λήψη και εγκατάσταση ενός προγράμματος εν ονόματι node.js, από το σύνδεσμο: https://nodejs.org/en/. Στην συνέχεια, θα πρέπει να ανοιχτεί το command window του ηλεκτρονικού υπολογιστή και να τοποθετηθούν οι απαραίτητες εντολές, για να επέλθει η σύνδεση με την Node – Red. Η διαδικασία και οι εντολές, απεικονίζονται παρακάτω.



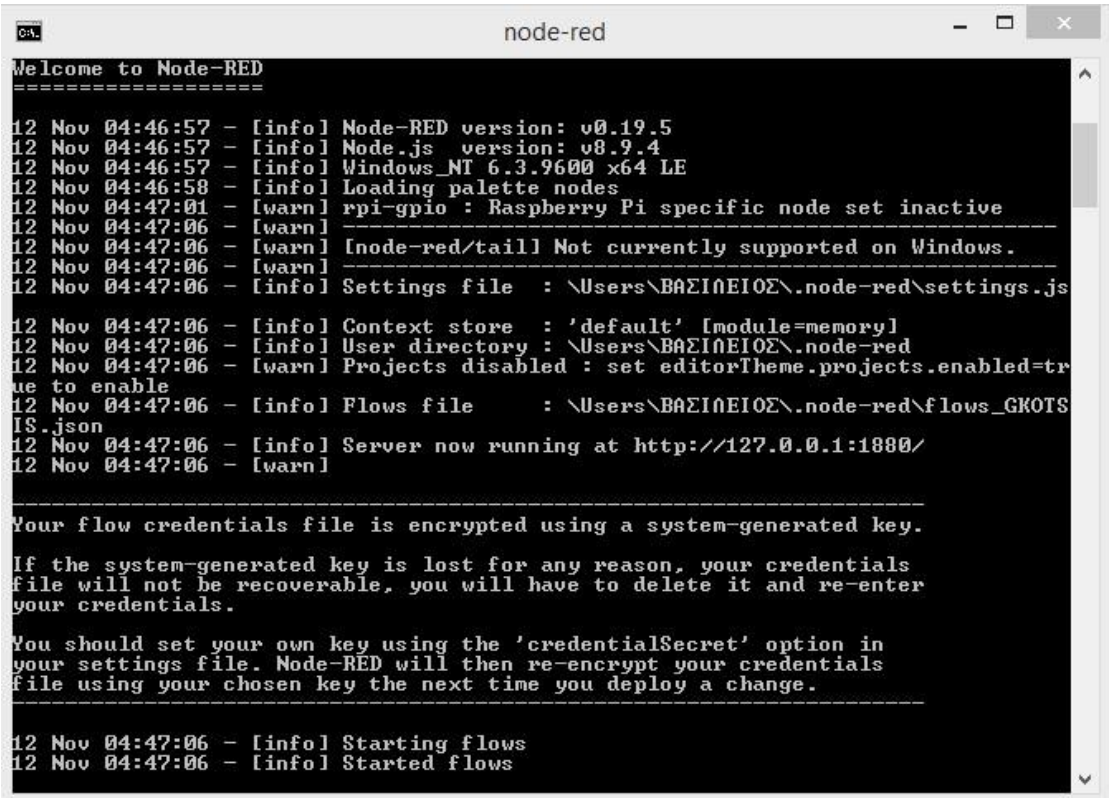
Εικόνα 4.2.1: Αρχική εντολή σύνδεσης Node – Red. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



Εικόνα 4.2.2: Τροποποίηση που πραγματοποιήθηκε μετά την πρώτη εντολή. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



Εικόνα 4.2.3: Τοποθέτηση δεύτερης εντολής σύνδεσης. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

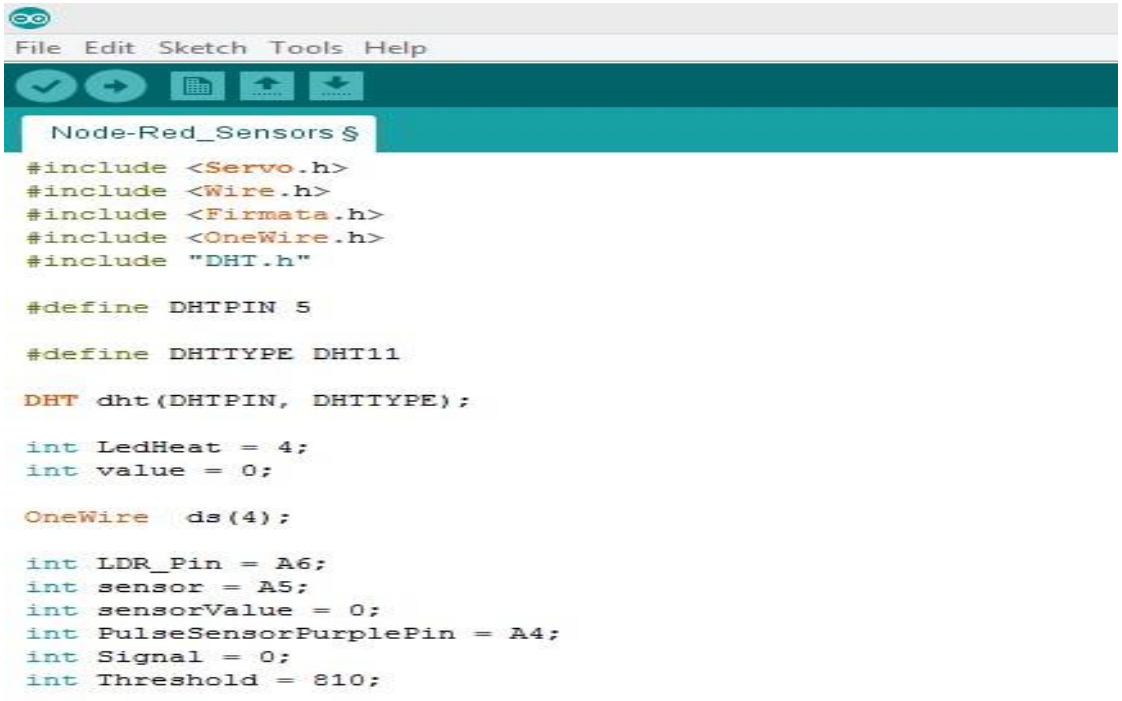


Εικόνα 4.2.4: Τελική κατάληξη command window μετά τις απαραίτητες εντολές και επιτυχημένη σύνδεση με την Node - Red. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

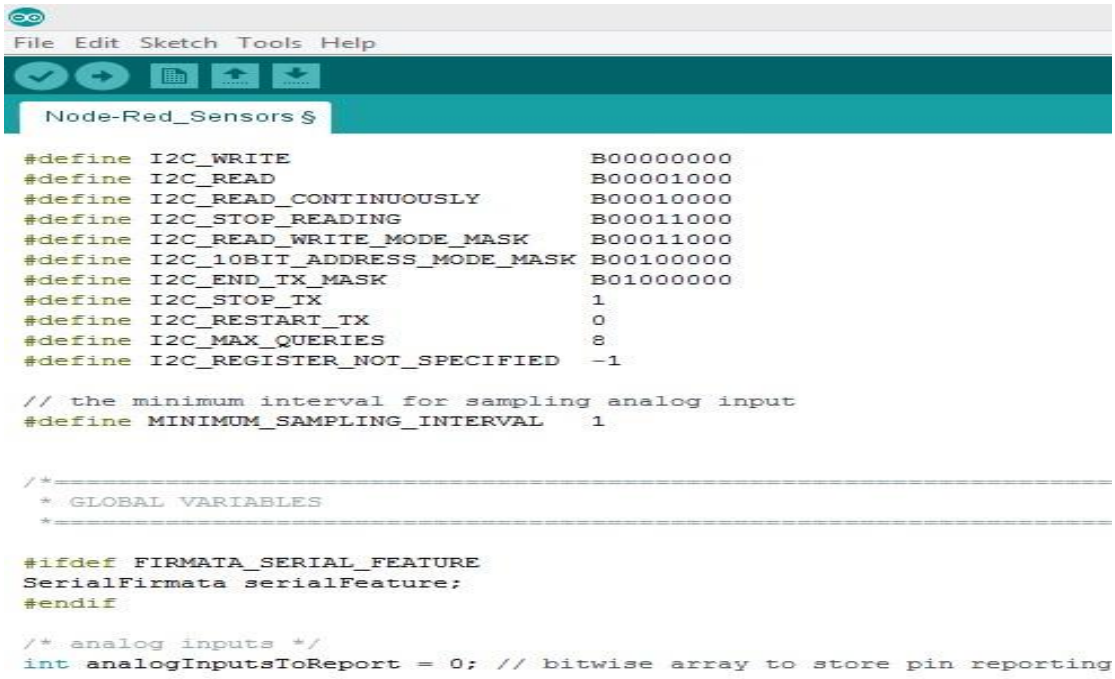
Προσοχή: Η σύνδεση με την Node – Red, θα συνεχίσει να υφίσταται, μόνο για όσο χρονικό διάστημα το command window, αφού τοποθετηθούν σε αυτό οι απαραίτητες εντολές, παραμείνει ανοιχτό.

4.3 Τελικό Πρόγραμμα (Arduino & Node – Red)

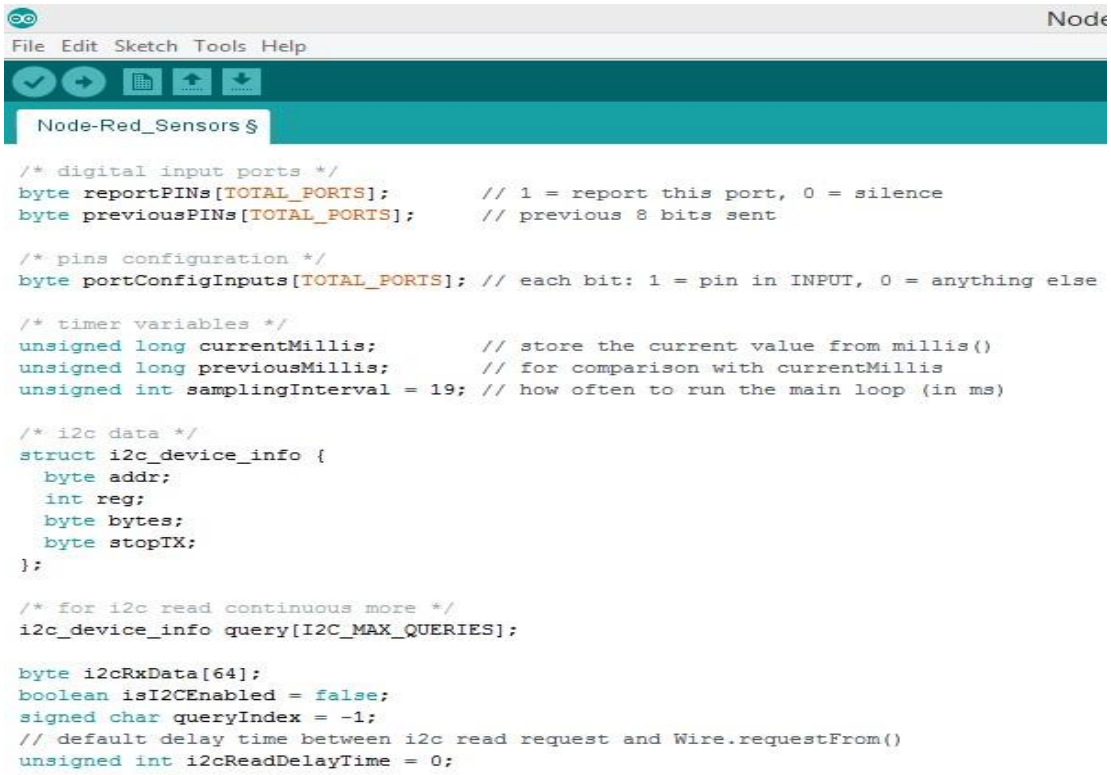
Στο συγκεκριμένο κεφάλαιο, παρουσιάζονται αναλυτικά ολόκληρο το πρόγραμμα που χρησιμοποιήθηκε τόσο στο προγραμματιστικό περιβάλλον του Arduino (IDE), όσο και στην σελίδα προγραμματισμού της Node – Red (http://localhost:1880/).



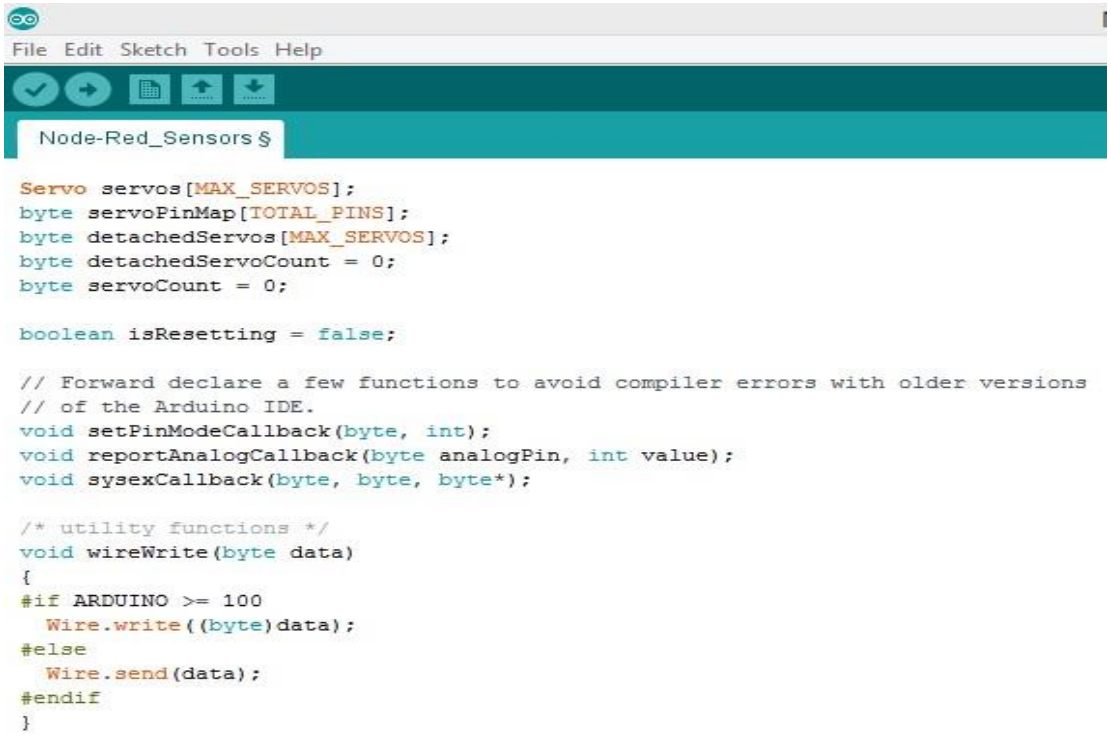
Εικόνα 4.3.1: Αρχικοποίηση βιβλιοθηκών και παραμέτρων που θα χρησιμοποιηθούν από το πρόγραμμα. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



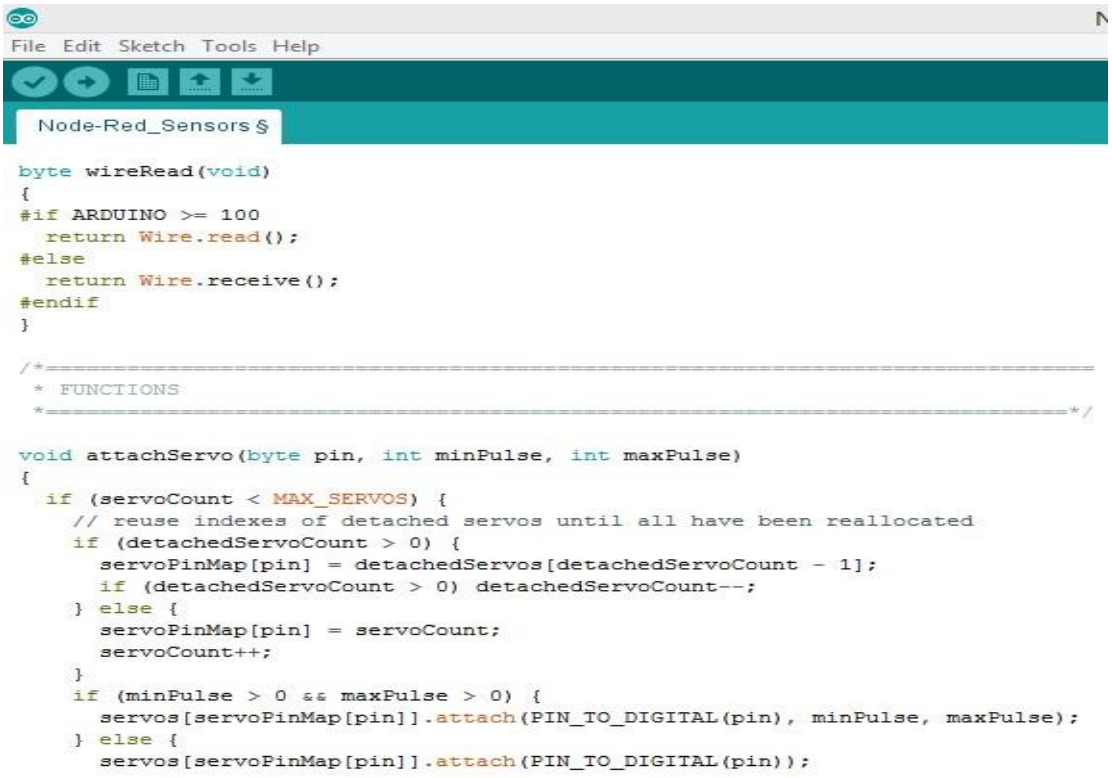
Εικόνα 4.3.2: Επιπλέον αρχικοποίηση παραμέτρων που θα χρησιμοποιηθούν από το πρόγραμμα. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



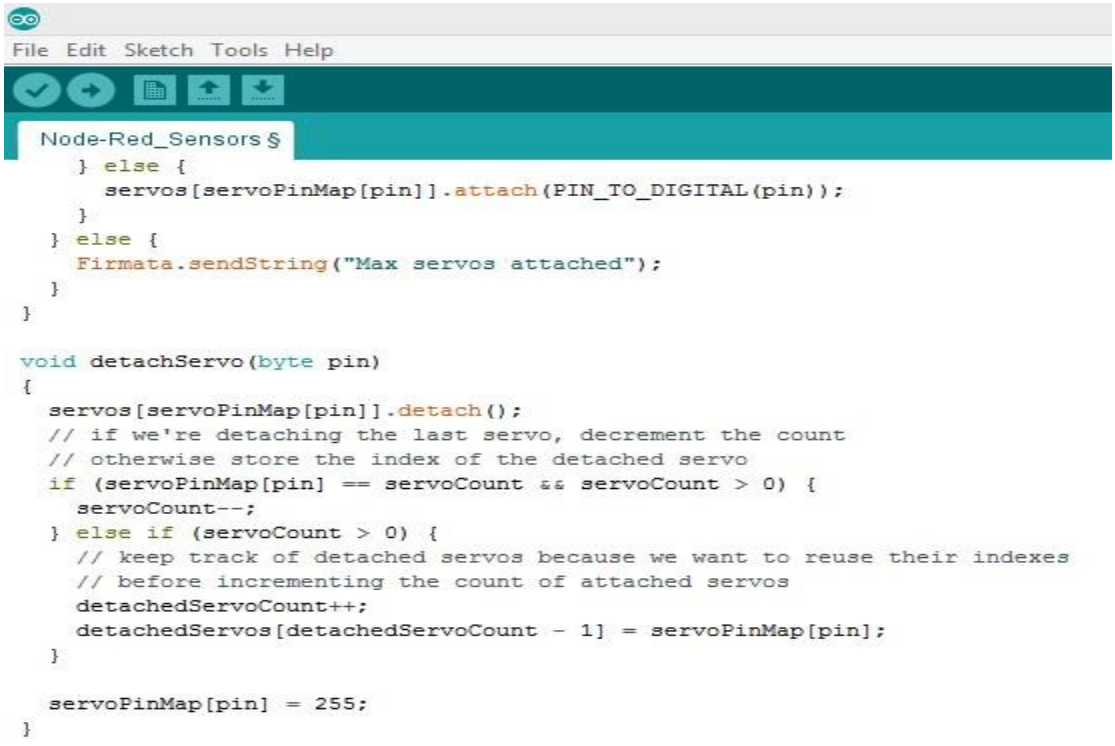
Εικόνα 4.3.3: Επιπλέον αρχικοποίηση παραμέτρων που θα χρησιμοποιηθούν από το πρόγραμμα. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



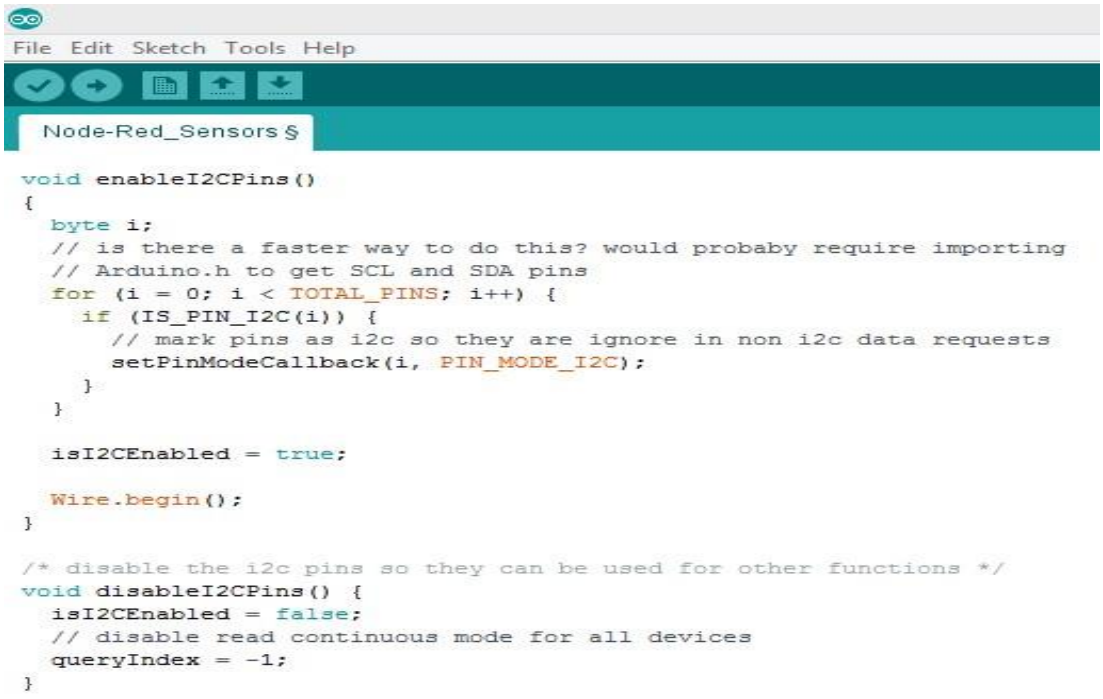
Εικόνα 4.3.4: Επιπλέον αρχικοποίηση παραμέτρων που θα χρησιμοποιηθούν από το πρόγραμμα.. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



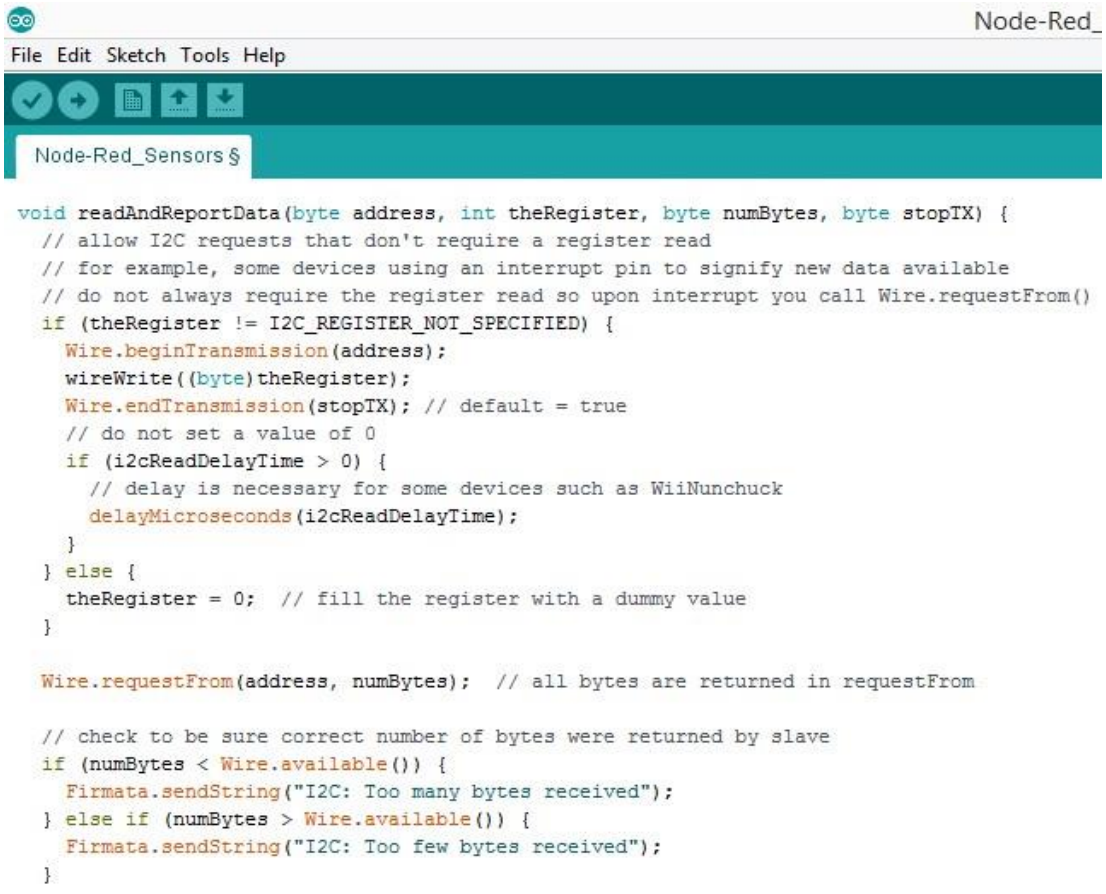
Εικόνα 4.3.5: Τοποθέτηση απαραίτητων συναρτήσεων. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



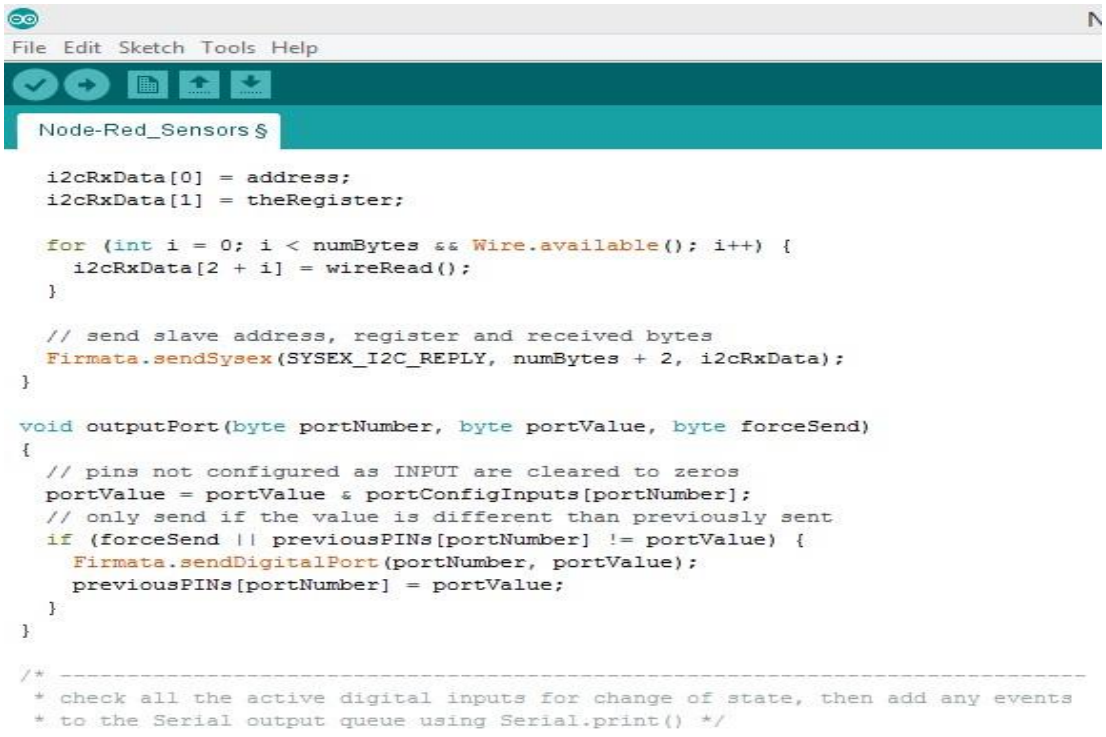
Εικόνα 4.3.6: Τοποθέτηση απαραίτητων συναρτήσεων. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



Εικόνα 4.3.7: Τοποθέτηση απαραίτητων συναρτήσεων. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

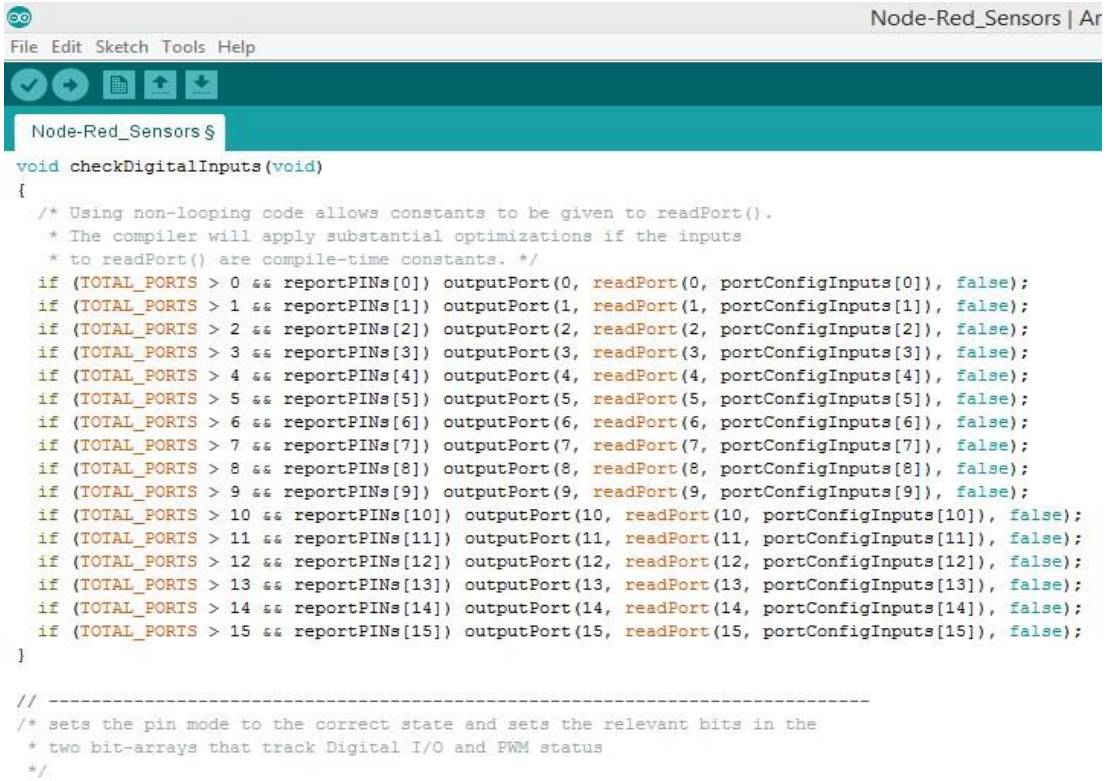


Εικόνα 4.3.8: Τοποθέτηση απαραίτητων συναρτήσεων. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



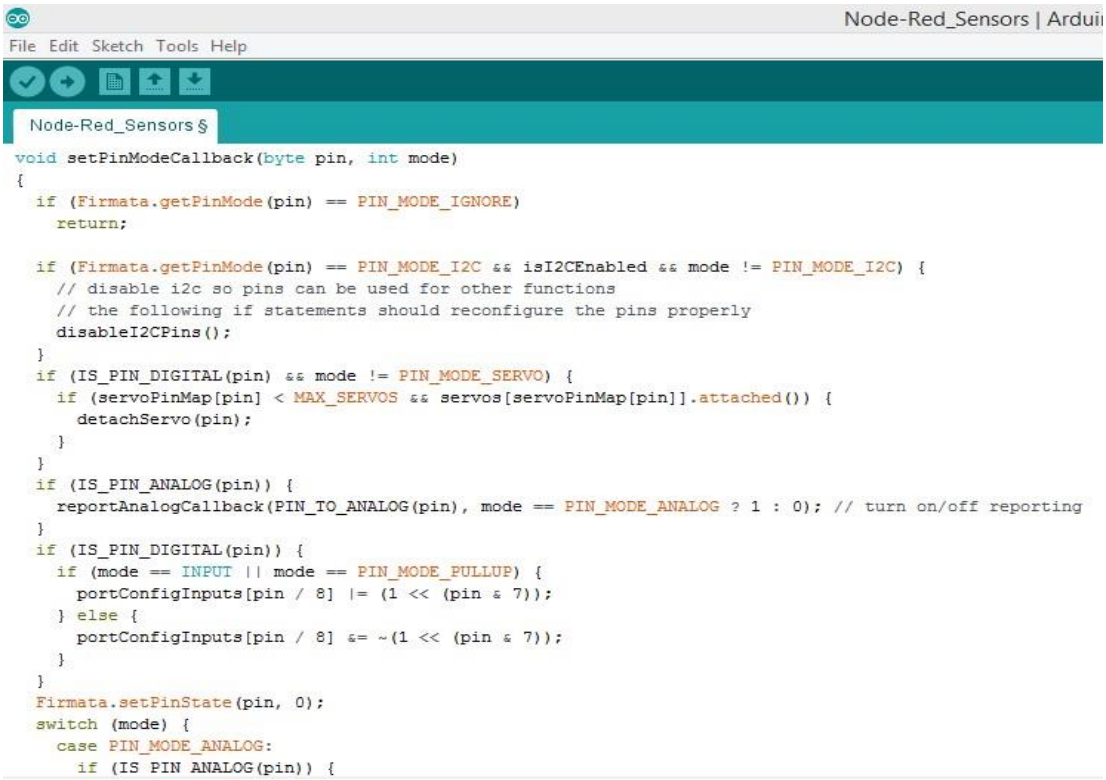

```
  i2cRxData[0] = address;
  i2cRxData[1] = theRegister;

  for (int i = 0; i < numBytes && Wire.available(); i++) {
    i2cRxData[2 + i] = wireRead();
  }

  // send slave address, register and received bytes
  Firmata.sendSysex(SYSEX_I2C_REPLY, numBytes + 2, i2cRxData);
}

void outputPort(byte portNumber, byte portValue, byte forceSend)
{
  // pins not configured as INPUT are cleared to zeros
  portValue = portValue & portConfigInputs[portNumber];
  // only send if the value is different than previously sent
  if (forceSend || previousPINs[portNumber] != portValue) {
    Firmata.sendDigitalPort(portNumber, portValue);
    previousPINs[portNumber] = portValue;
  }
}

/* -----------------------------------------------------------------------------
 * check all the active digital inputs for change of state, then add any events
 * to the Serial output queue using Serial.print() */
```

Εικόνα 4.3.9: Τοποθέτηση απαραίτητων συναρτήσεων. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

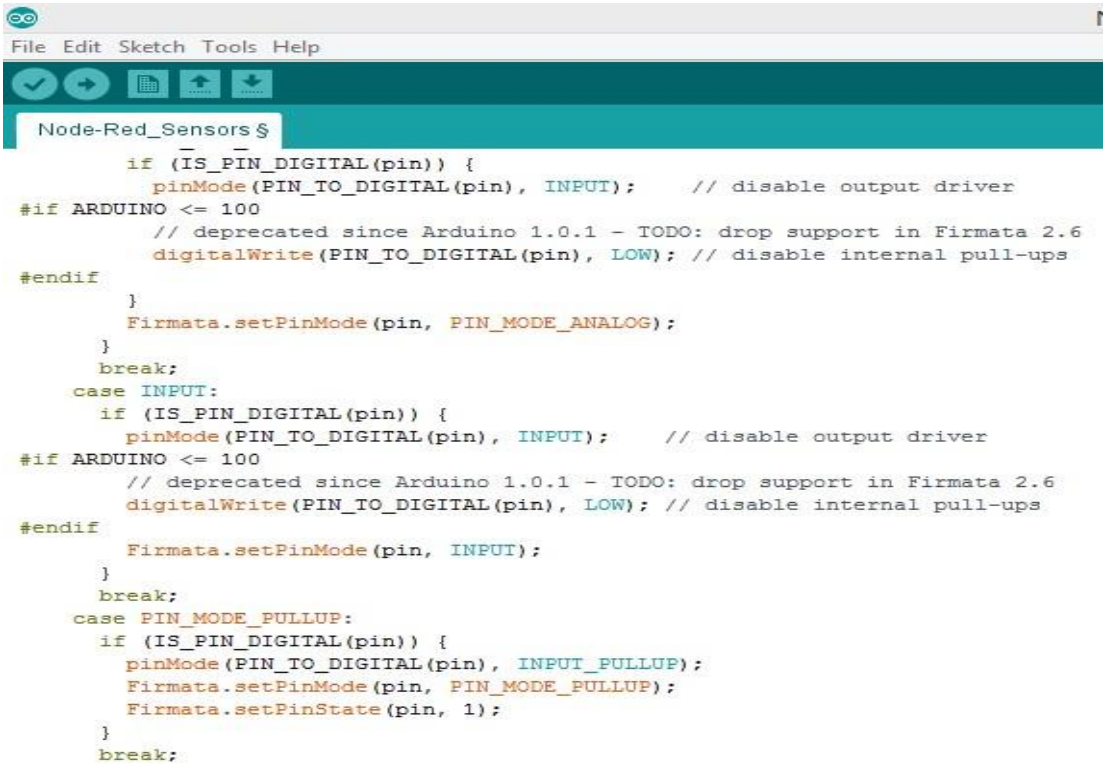



```
void checkDigitalInputs(void)
{
  /* Using non-looping code allows constants to be given to readPort().
   * The compiler will apply substantial optimizations if the inputs
   * to readPort() are compile-time constants. */
  if (TOTAL_PORTS > 0 && reportPINs[0]) outputPort(0, readPort(0, portConfigInputs[0]), false);
  if (TOTAL_PORTS > 1 && reportPINs[1]) outputPort(1, readPort(1, portConfigInputs[1]), false);
  if (TOTAL_PORTS > 2 && reportPINs[2]) outputPort(2, readPort(2, portConfigInputs[2]), false);
  if (TOTAL_PORTS > 3 && reportPINs[3]) outputPort(3, readPort(3, portConfigInputs[3]), false);
  if (TOTAL_PORTS > 4 && reportPINs[4]) outputPort(4, readPort(4, portConfigInputs[4]), false);
  if (TOTAL_PORTS > 5 && reportPINs[5]) outputPort(5, readPort(5, portConfigInputs[5]), false);
  if (TOTAL_PORTS > 6 && reportPINs[6]) outputPort(6, readPort(6, portConfigInputs[6]), false);
  if (TOTAL_PORTS > 7 && reportPINs[7]) outputPort(7, readPort(7, portConfigInputs[7]), false);
  if (TOTAL_PORTS > 8 && reportPINs[8]) outputPort(8, readPort(8, portConfigInputs[8]), false);
  if (TOTAL_PORTS > 9 && reportPINs[9]) outputPort(9, readPort(9, portConfigInputs[9]), false);
  if (TOTAL_PORTS > 10 && reportPINs[10]) outputPort(10, readPort(10, portConfigInputs[10]), false);
  if (TOTAL_PORTS > 11 && reportPINs[11]) outputPort(11, readPort(11, portConfigInputs[11]), false);
  if (TOTAL_PORTS > 12 && reportPINs[12]) outputPort(12, readPort(12, portConfigInputs[12]), false);
  if (TOTAL_PORTS > 13 && reportPINs[13]) outputPort(13, readPort(13, portConfigInputs[13]), false);
  if (TOTAL_PORTS > 14 && reportPINs[14]) outputPort(14, readPort(14, portConfigInputs[14]), false);
  if (TOTAL_PORTS > 15 && reportPINs[15]) outputPort(15, readPort(15, portConfigInputs[15]), false);
}

// -----------------------------------------------------------------------------
/* sets the pin mode to the correct state and sets the relevant bits in the
 * two bit-arrays that track Digital I/O and PWM status
 */
```

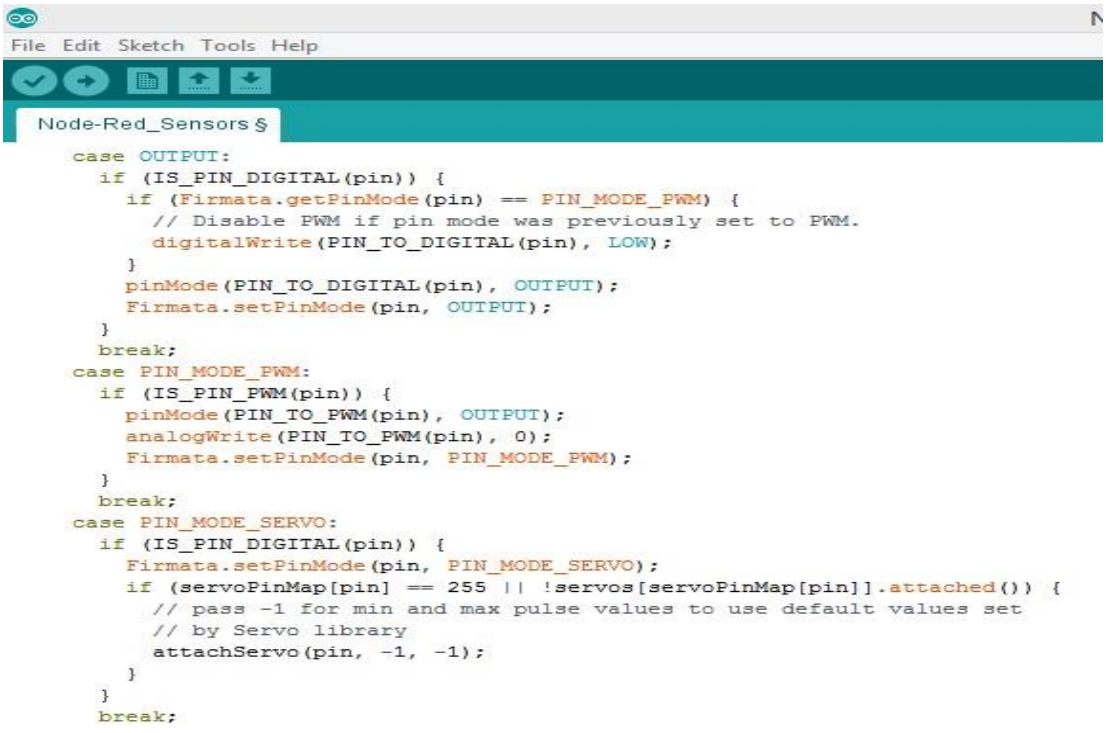
Εικόνα 4.3.10: Έλεγχος ψηφιακών εισόδων και προσθήκη συμβάντος σε περίπτωση αλλαγής κατάστασης. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



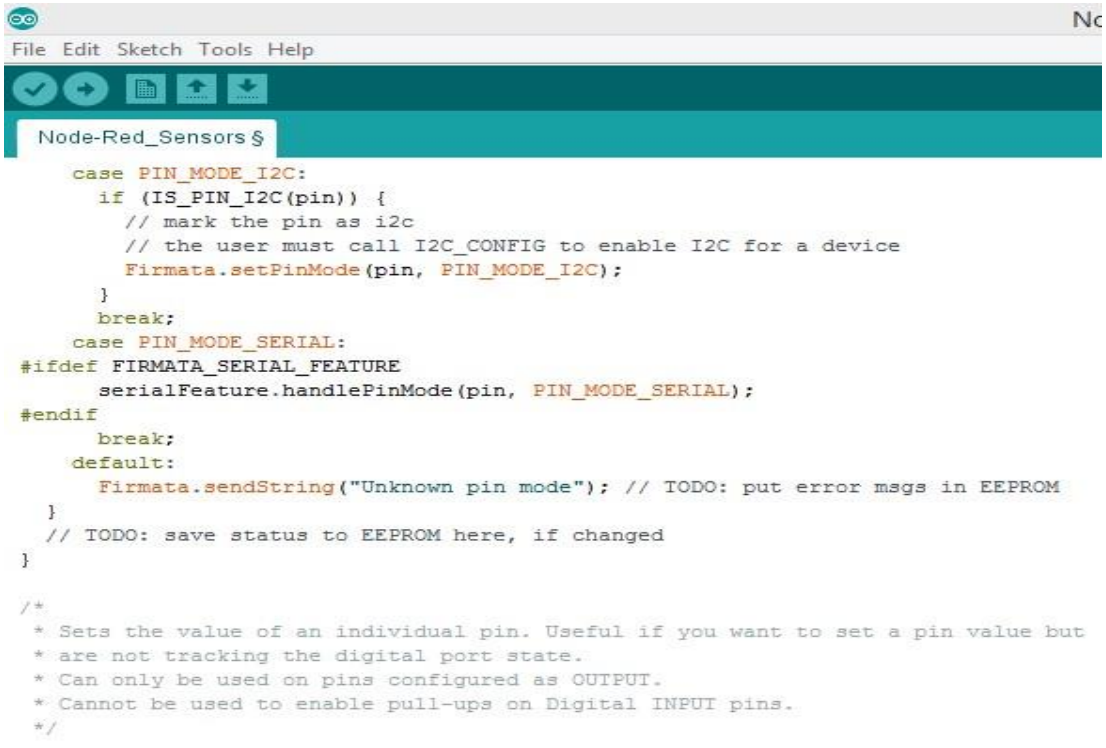
Εικόνα 4.3.11: Ρύθμιση των pins στη σωστή κατάσταση και καταχώρηση των σχετικών bits σε συστοιχία δύο bit, που ανιχνεύει ψηφιακή είσοδο και έξοδο. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



Εικόνα 4.3.12: Ρύθμιση των pins στη σωστή κατάσταση και καταχώρηση των σχετικών bits σε συστοιχία δύο bit, που ανιχνεύει ψηφιακή είσοδο και έξοδο. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

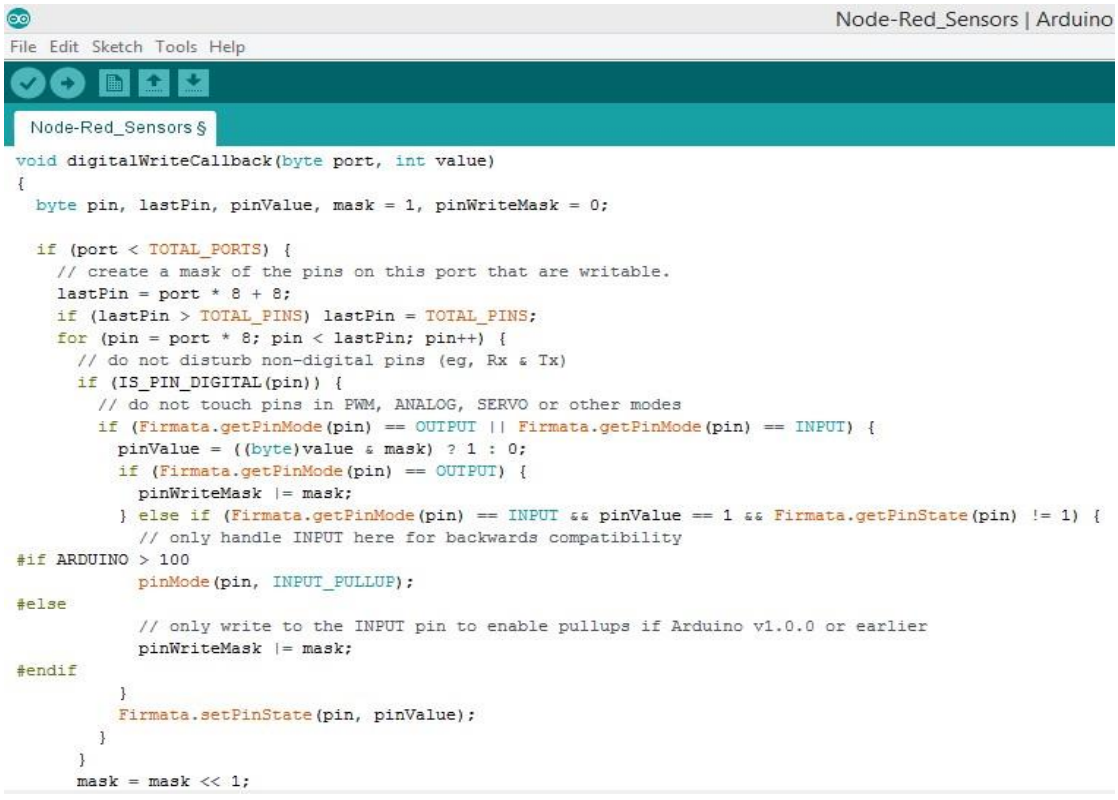


Εικόνα 4.3.13: Ρύθμιση των pins στη σωστή κατάσταση και καταχώρηση των σχετικών bits σε συστοιχία δύο bit, που ανιχνεύει ψηφιακή είσοδο και έξοδο. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

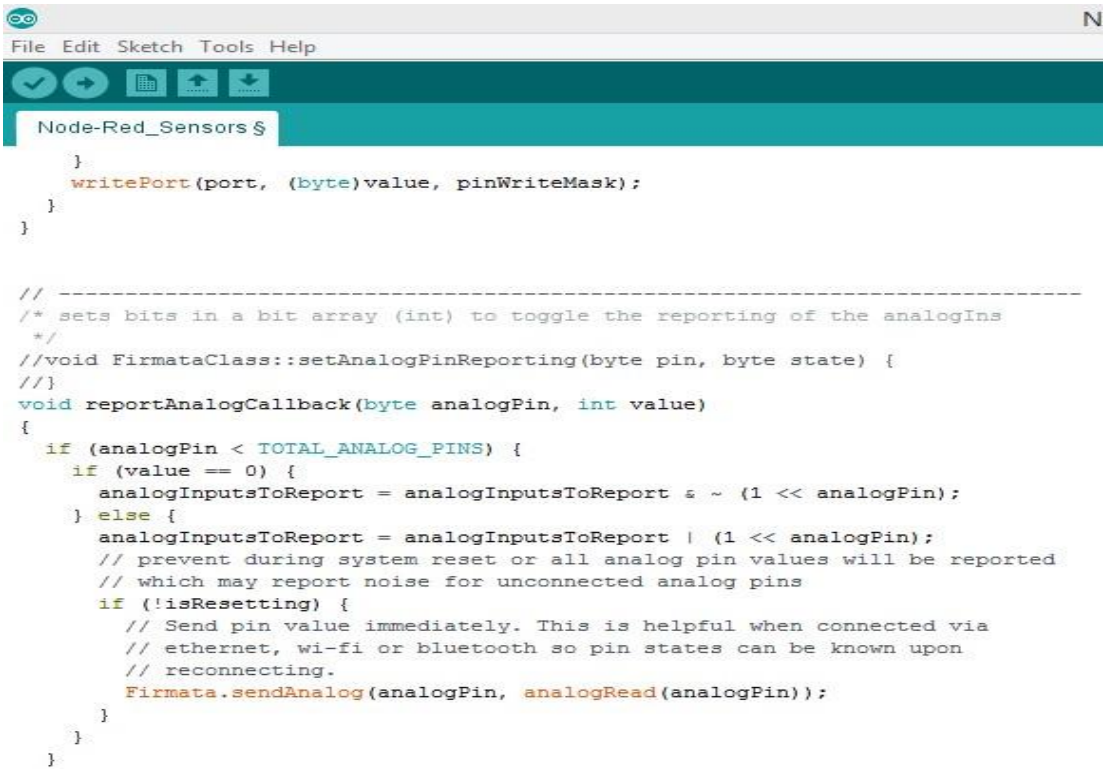


Εικόνα 4.3.14: Ρύθμιση των pins στη σωστή κατάσταση και καταχώρηση των σχετικών bits σε συστοιχία δύο bit, που ανιχνεύει ψηφιακή είσοδο και έξοδο. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

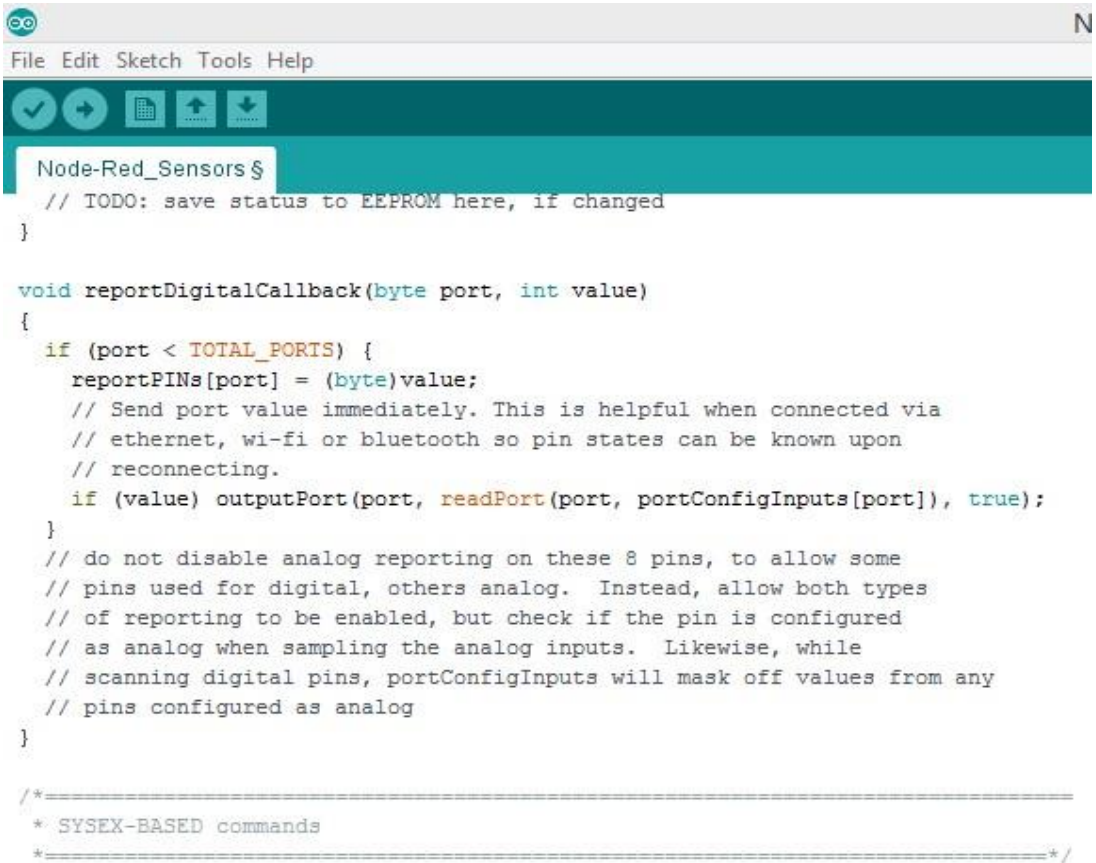
Εικόνα 4.3.15: Προσθήκη τιμών στα pin. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



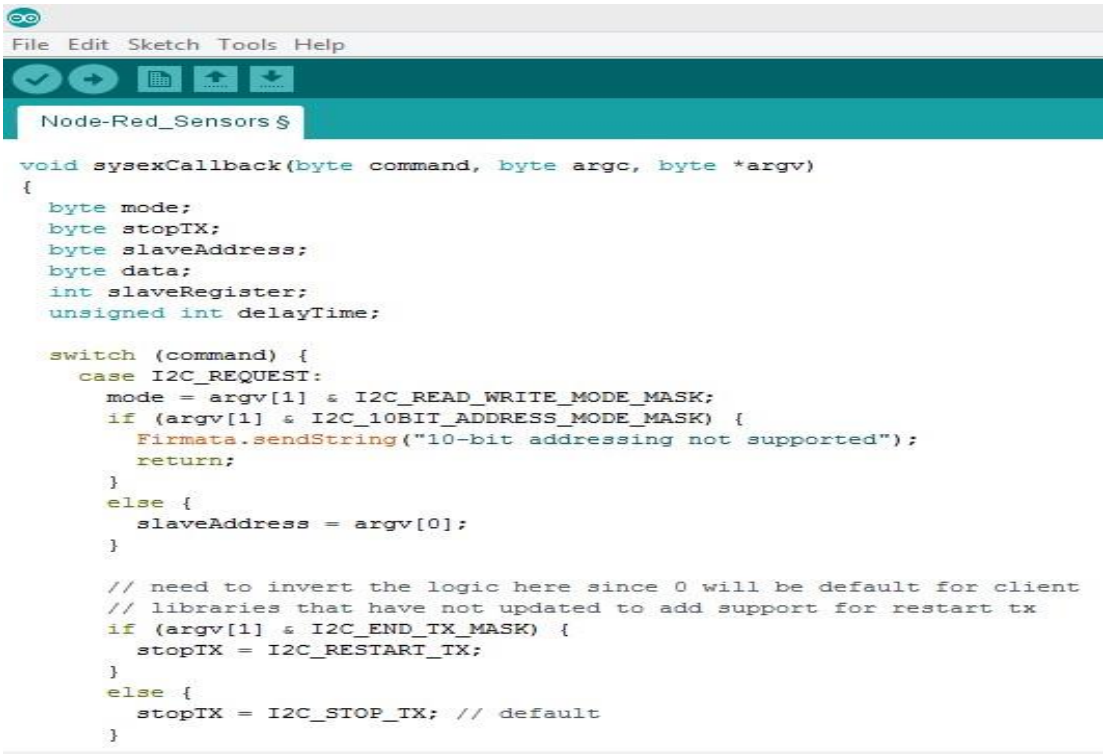
Εικόνα 4.3.16: Προσθήκη τιμών στα pin. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



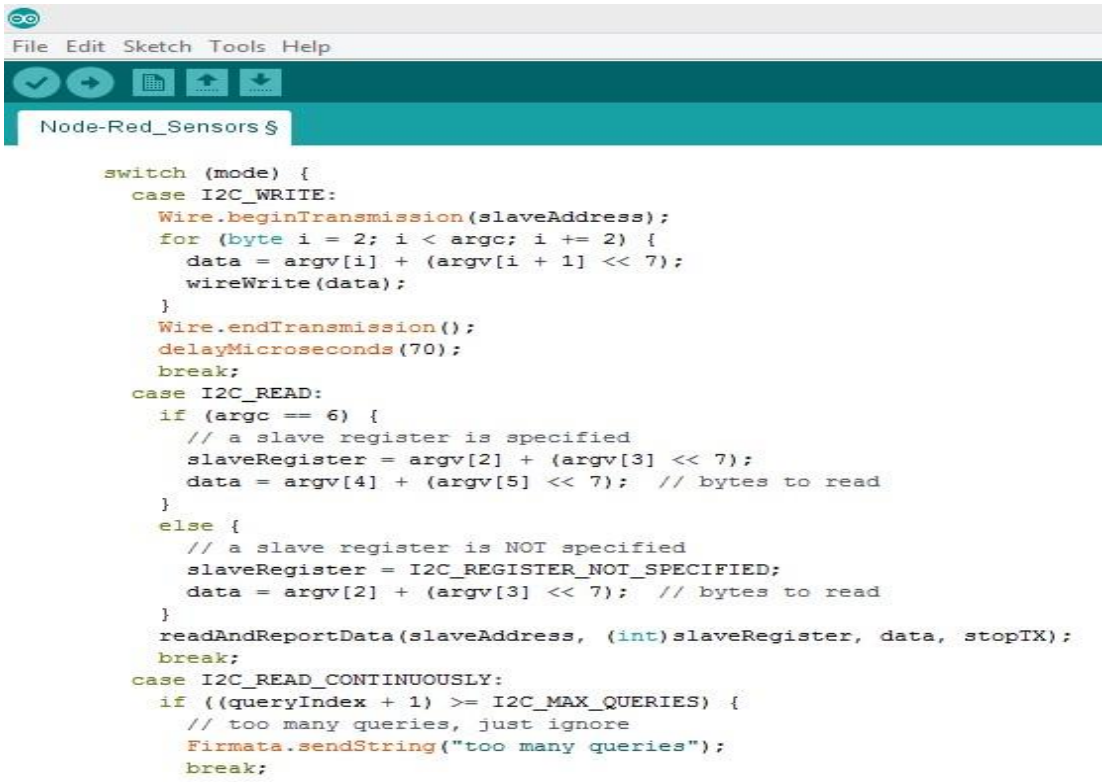
Εικόνα 4.3.17: Προσθήκη τιμών στα pin. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



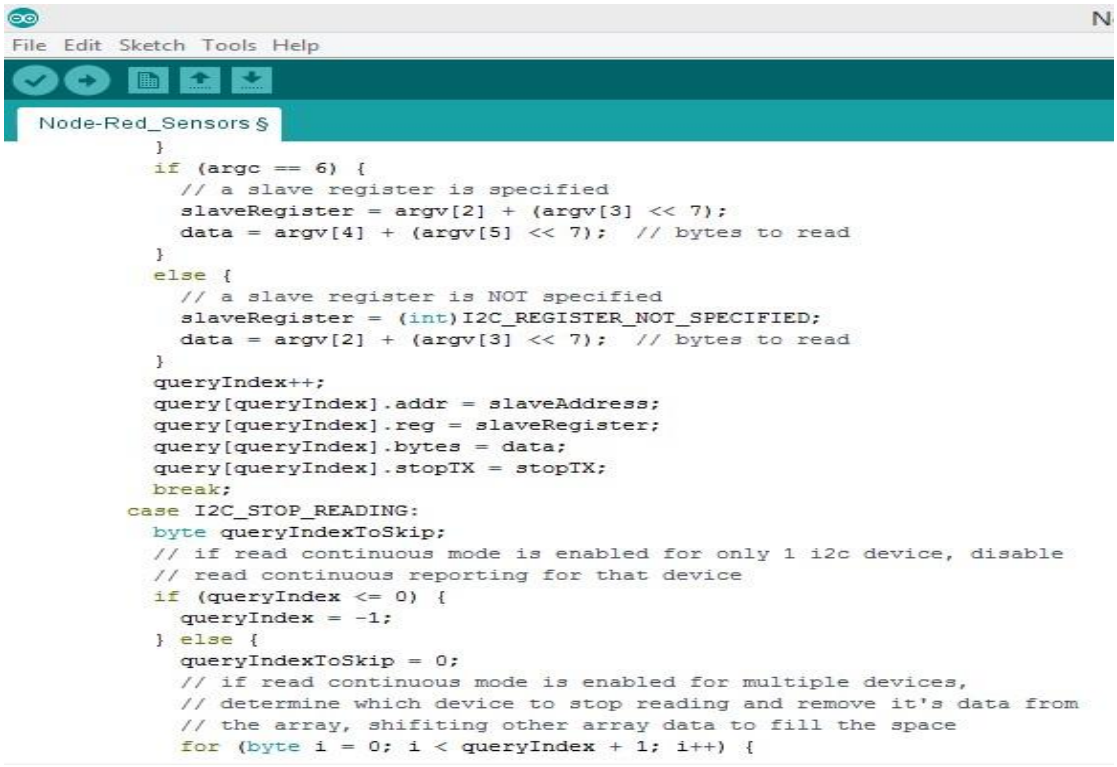
Εικόνα 4.3.18: Προσθήκη τιμών στα pin. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



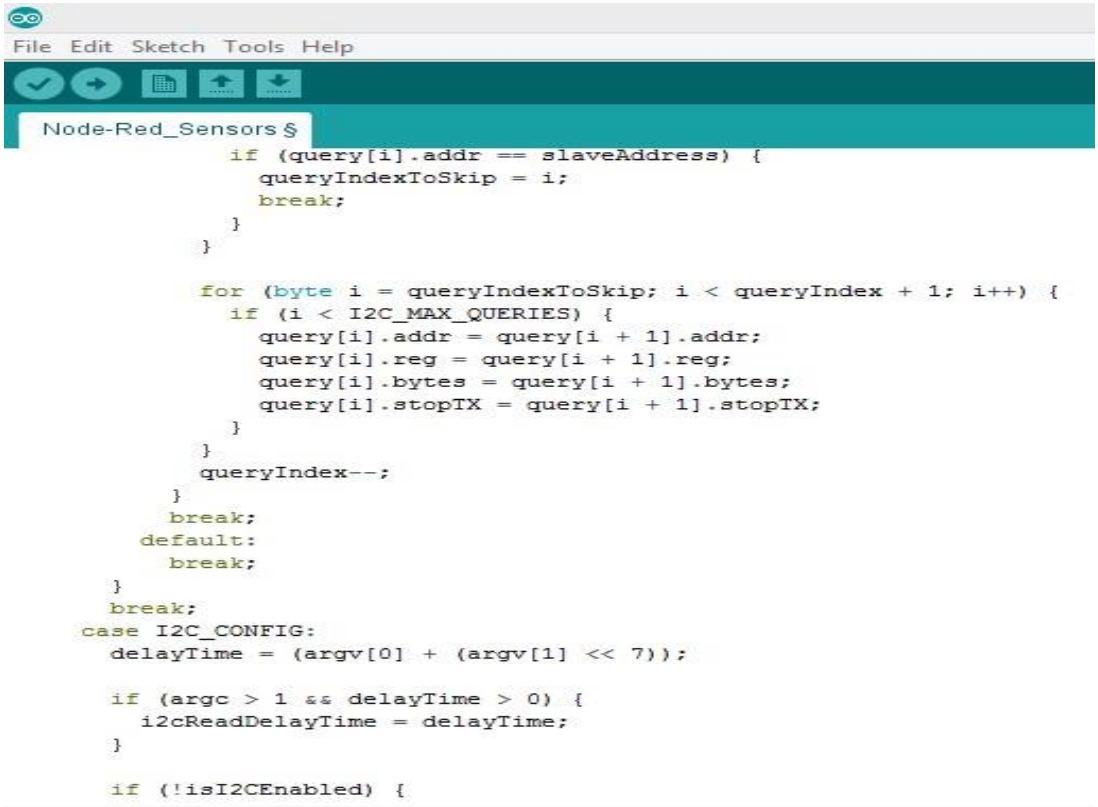
Εικόνα 4.3.19: Απαραίτητες εντολές λειτουργίας προγράμματος. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



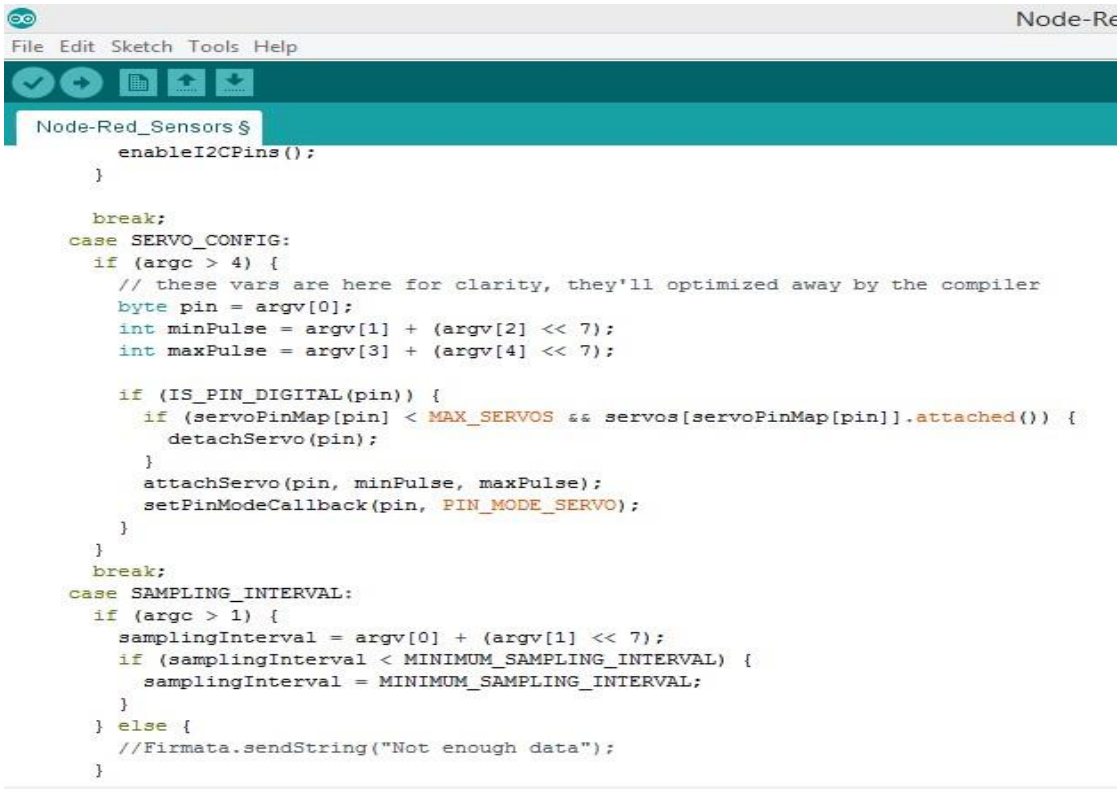
Εικόνα 4.3.20: Απαραίτητες εντολές λειτουργίας προγράμματος. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



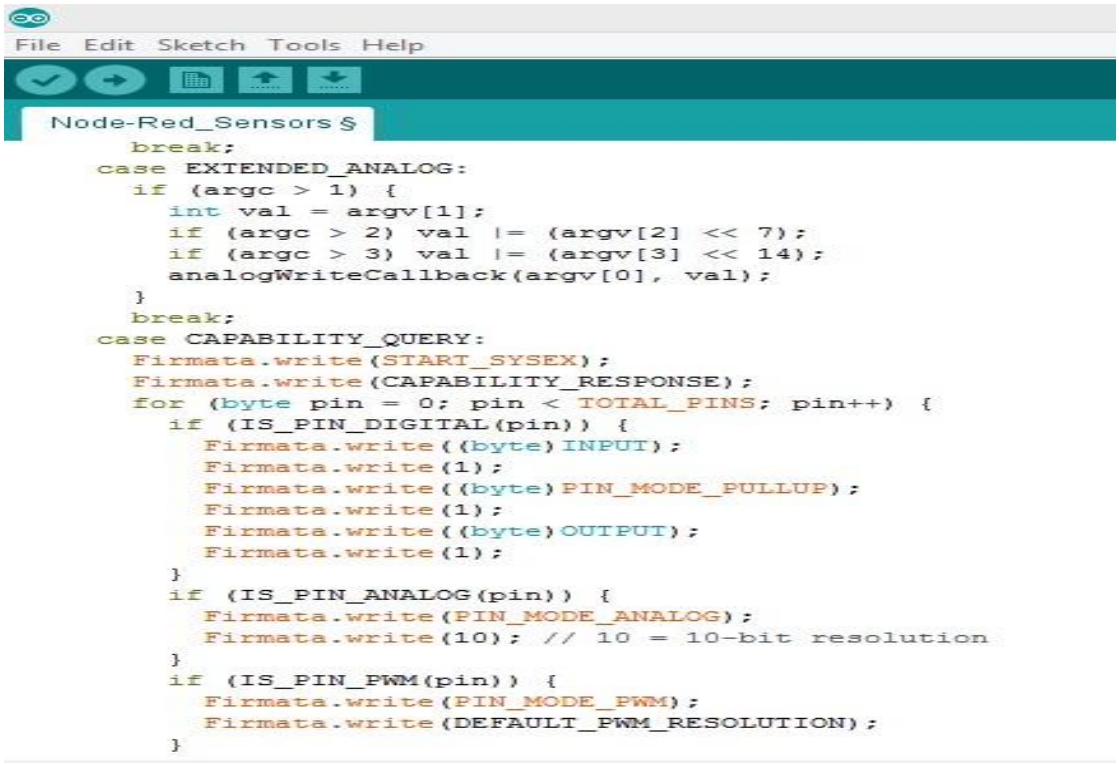
Εικόνα 4.3.21: Απαραίτητες εντολές λειτουργίας προγράμματος. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



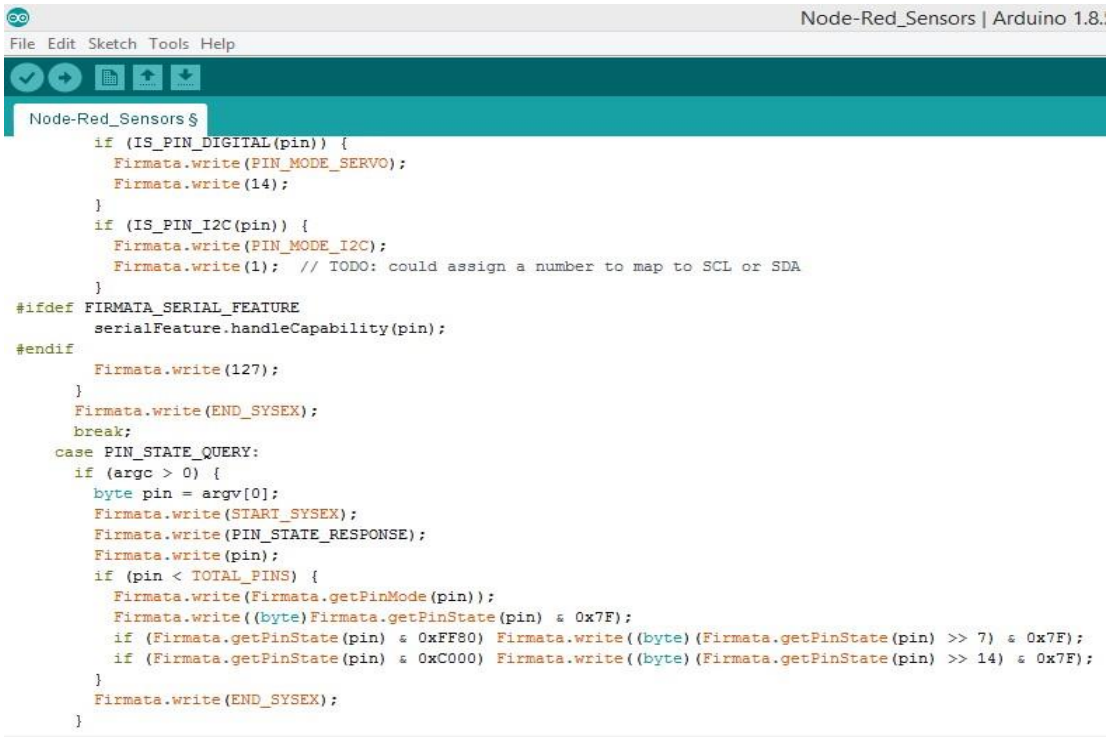
Εικόνα 4.3.22: Απαραίτητες εντολές λειτουργίας προγράμματος. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



Εικόνα 4.3.23: Απαραίτητες εντολές λειτουργίας προγράμματος. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

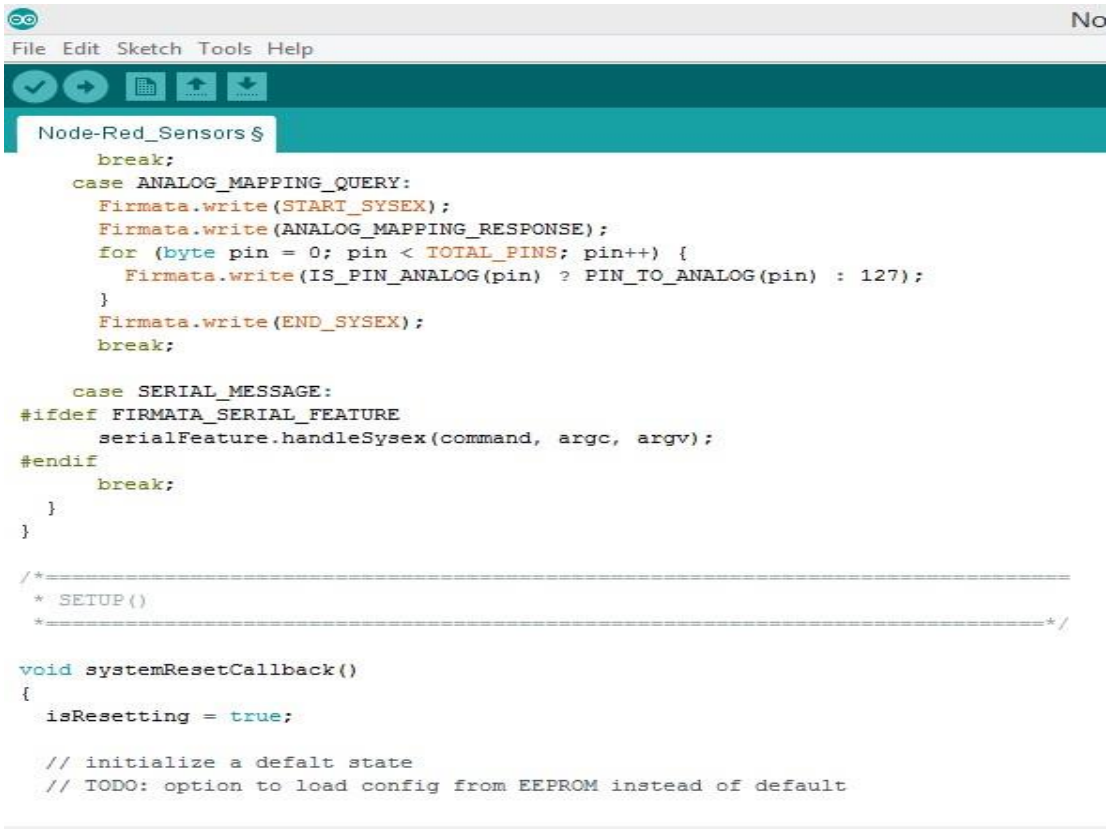


Εικόνα 4.3.24: Απαραίτητες εντολές λειτουργίας προγράμματος. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



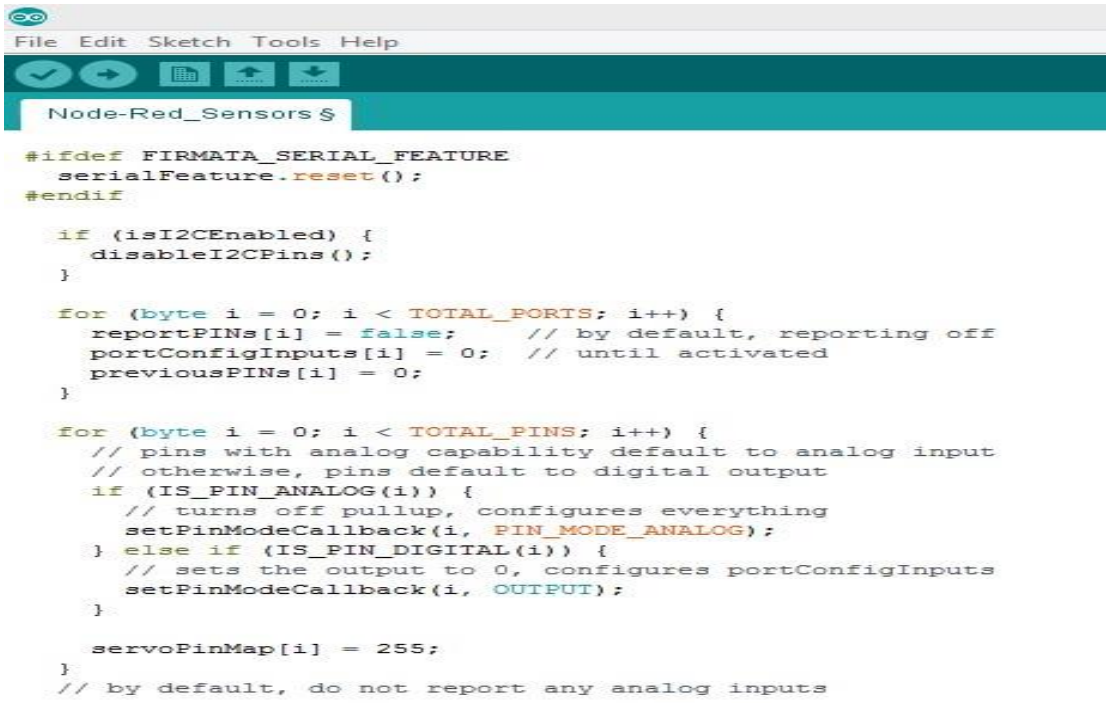

```
        if (IS_PIN_DIGITAL(pin)) {
          Firmata.write(PIN_MODE_SERVO);
          Firmata.write(14);
        }
        if (IS_PIN_I2C(pin)) {
          Firmata.write(PIN_MODE_I2C);
          Firmata.write(1);  // TODO: could assign a number to map to SCL or SDA
        }
#ifdef FIRMATA_SERIAL_FEATURE
        serialFeature.handleCapability(pin);
#endif
        Firmata.write(127);
      }
      Firmata.write(END_SYSEX);
      break;
    case PIN_STATE_QUERY:
      if (argc > 0) {
        byte pin = argv[0];
        Firmata.write(START_SYSEX);
        Firmata.write(PIN_STATE_RESPONSE);
        Firmata.write(pin);
        if (pin < TOTAL_PINS) {
          Firmata.write(Firmata.getPinMode(pin));
          Firmata.write((byte)Firmata.getPinState(pin) & 0x7F);
          if (Firmata.getPinState(pin) & 0xFF80) Firmata.write((byte)(Firmata.getPinState(pin) >> 7) & 0x7F);
          if (Firmata.getPinState(pin) & 0xC000) Firmata.write((byte)(Firmata.getPinState(pin) >> 14) & 0x7F);
        }
        Firmata.write(END_SYSEX);
      }
```

Εικόνα 4.3.25: Απαραίτητες εντολές λειτουργίας προγράμματος. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

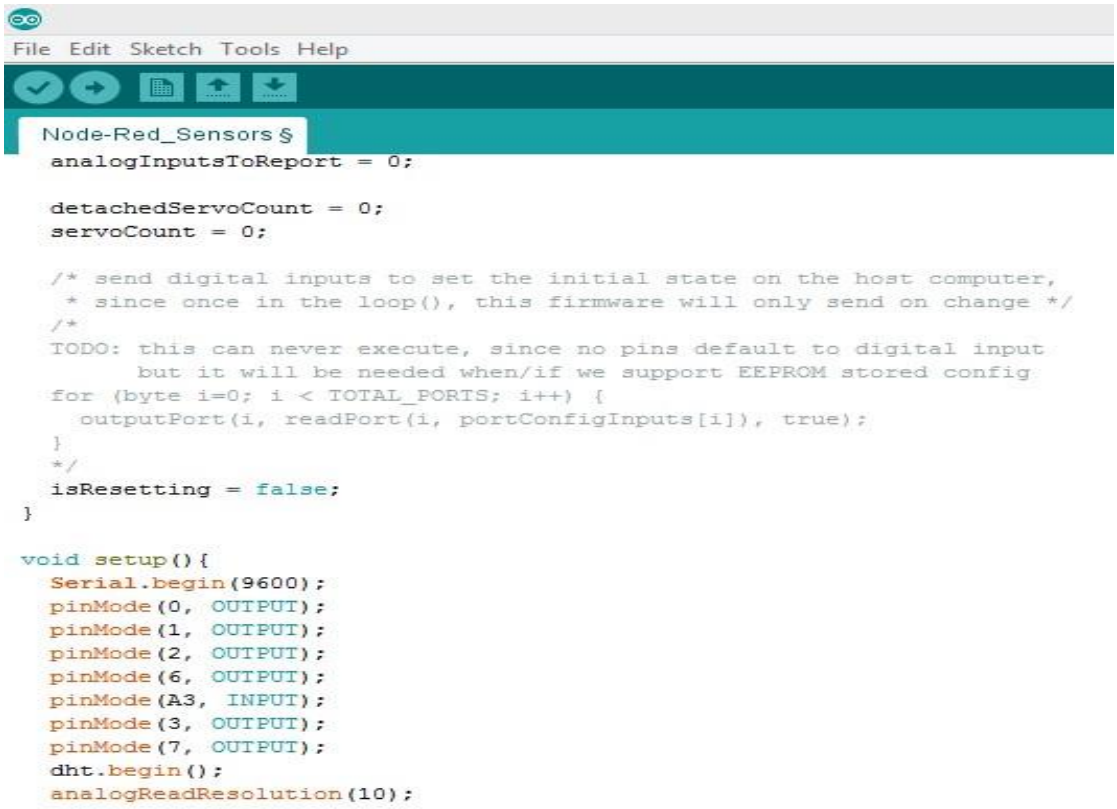



```
        break;
    case ANALOG_MAPPING_QUERY:
      Firmata.write(START_SYSEX);
      Firmata.write(ANALOG_MAPPING_RESPONSE);
      for (byte pin = 0; pin < TOTAL_PINS; pin++) {
        Firmata.write(IS_PIN_ANALOG(pin) ? PIN_TO_ANALOG(pin) : 127);
      }
      Firmata.write(END_SYSEX);
      break;

    case SERIAL_MESSAGE:
#ifdef FIRMATA_SERIAL_FEATURE
      serialFeature.handleSysex(command, argc, argv);
#endif
      break;
  }
}

/*==============================================================================
 * SETUP()
 *============================================================================*/

void systemResetCallback()
{
  isResetting = true;

  // initialize a defalt state
  // TODO: option to load config from EEPROM instead of default
```

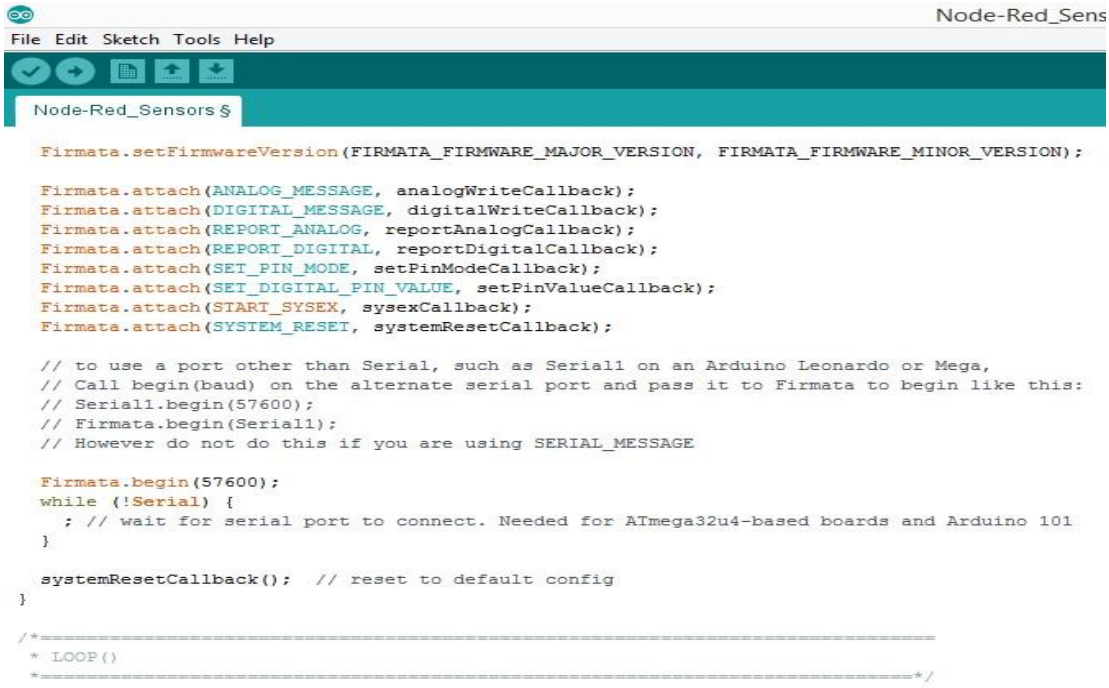
Εικόνα 4.3.26: Απαραίτητες εντολές λειτουργίας προγράμματος. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



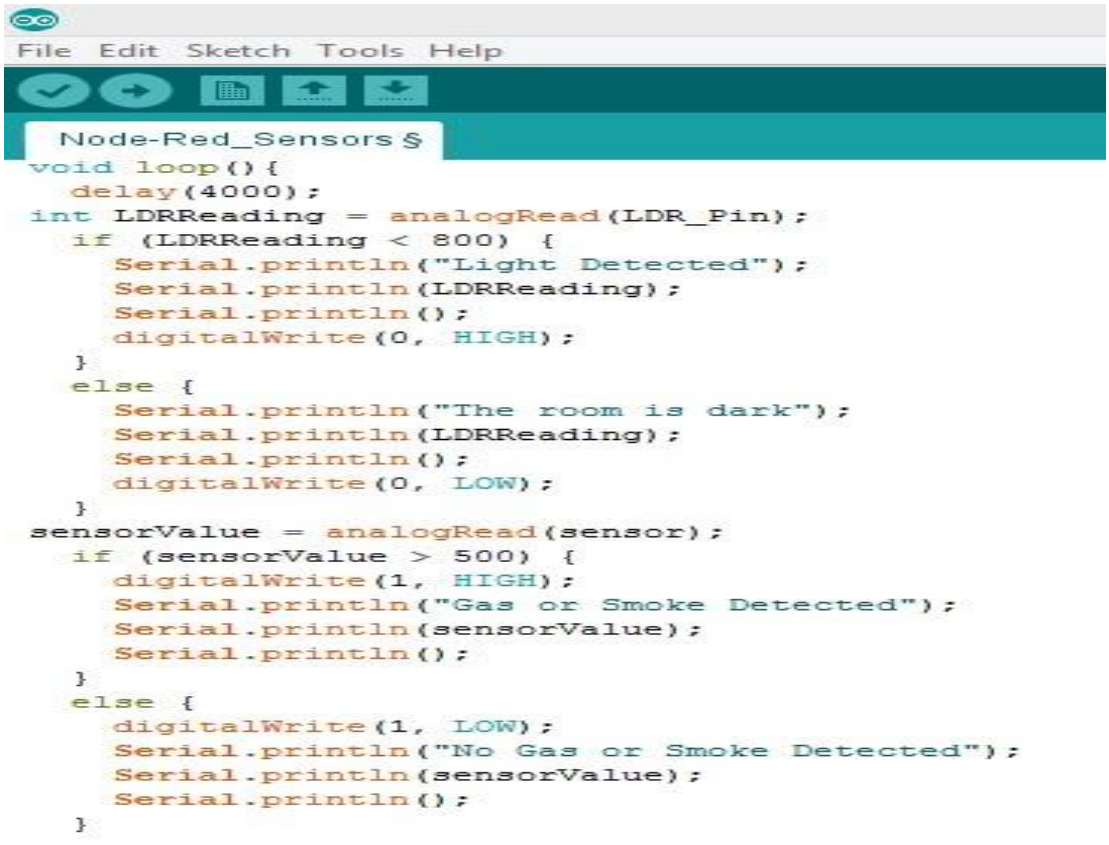
Εικόνα 4.3.27: Απαραίτητες εντολές λειτουργίας προγράμματος. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



Εικόνα 4.3.28: Δημιουργία της διάταξης του προγράμματος (setup) και δήλωση εισόδων, εξόδων. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

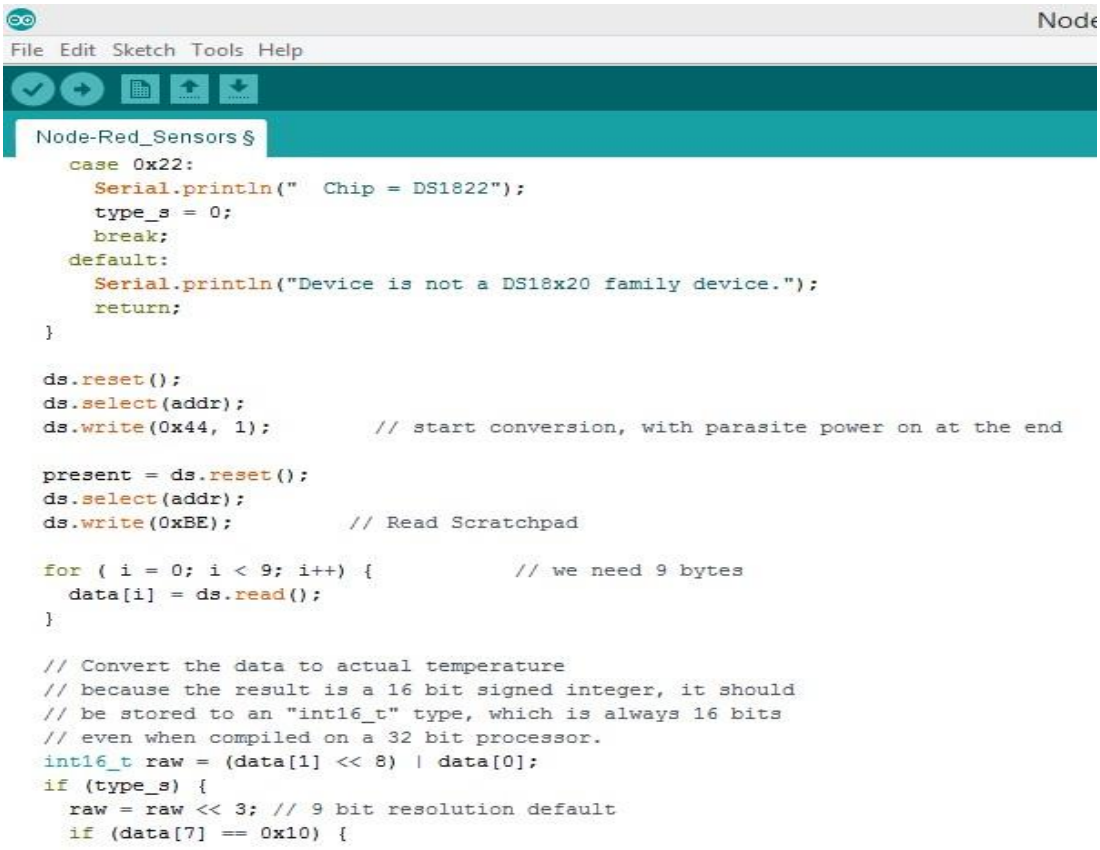


Εικόνα 4.3.29: Τοποθέτηση εντολών μηνυμάτων. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



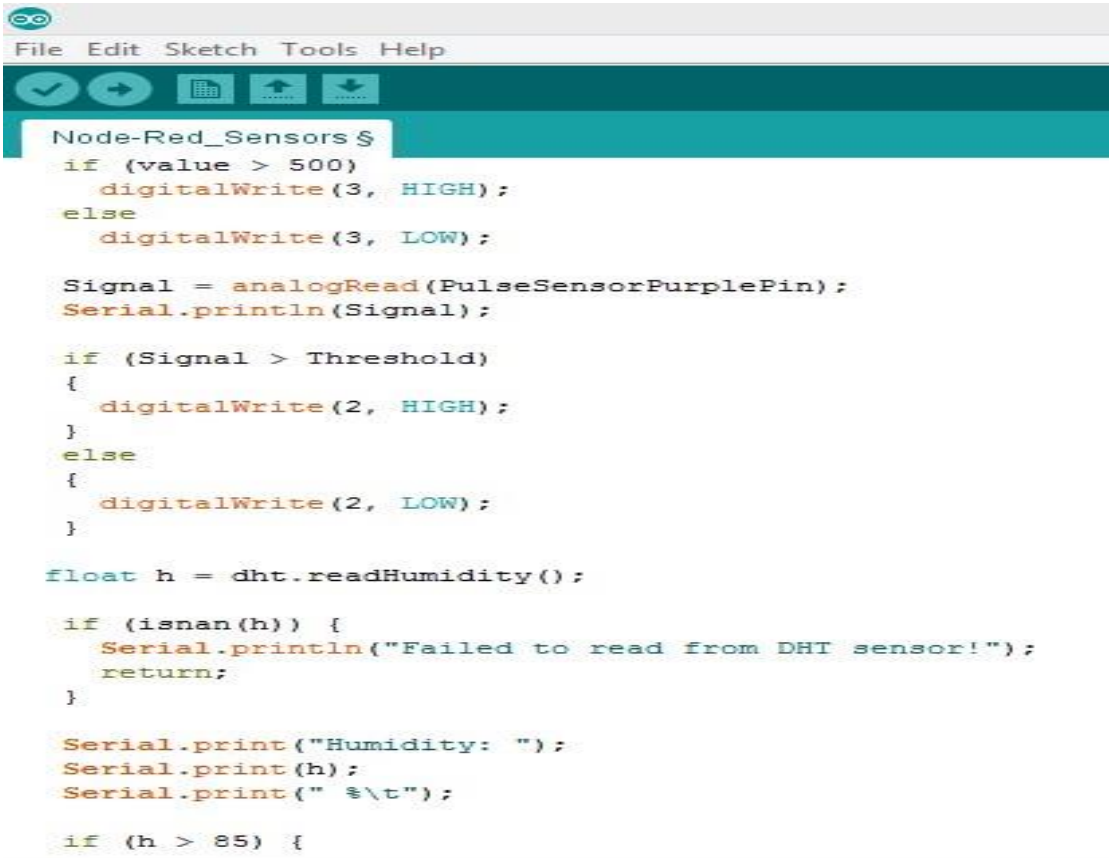
Εικόνα 4.3.30: Εντολές λειτουργίας αισθητήρα φωτός. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

Εικόνα 4.3.31: Εντολές λειτουργίας αισθητήρα θερμοκρασίας. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



Εικόνα 4.3.32: Εντολές λειτουργίας αισθητήρα θερμοκρασίας. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

Εικόνα 4.3.33: Εντολές λειτουργίας αισθητήρα θερμοκρασίας. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)



Εικόνα 4.3.34: Εντολές λειτουργίας αισθητήρα μέτρησης καρδιακών παλμών και υγρασίας. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

Εικόνα 4.3.35: Εντολές λειτουργίας αισθητήρα υγρασίας και έλεγχος είδους* pin. *Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

Εικόνα 4.3.36: Έλεγχος είδους* pin. *Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

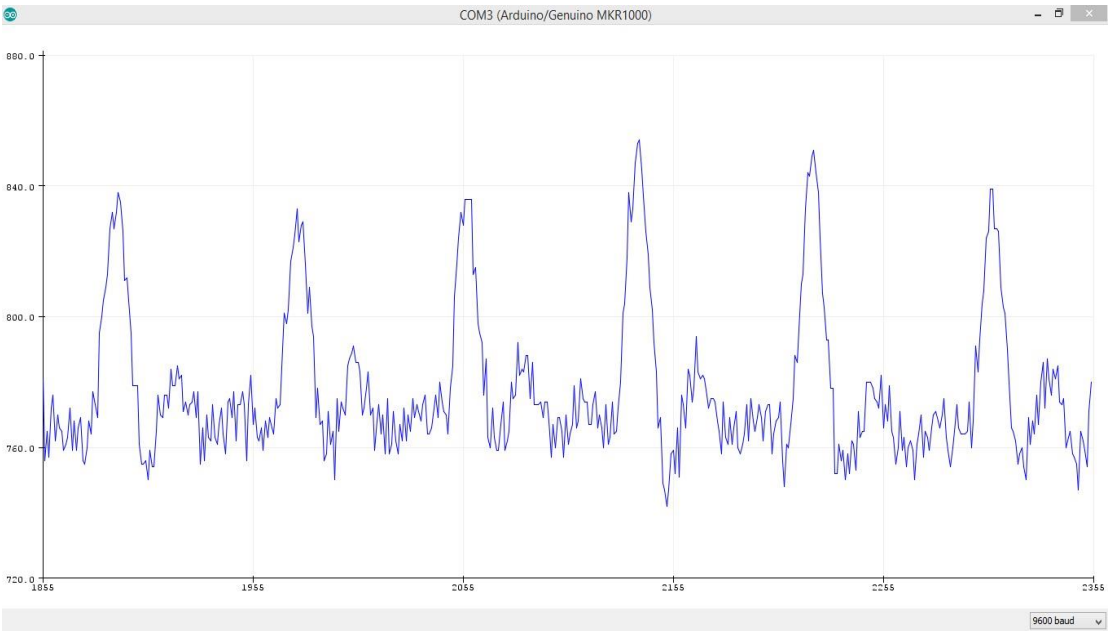
Εικόνα 4.3.37: Συνδεσμολογία και σχηματική απεικόνιση εντολών της Node – Red, για επιτυχή αποστολή email και λειτουργίας, όλων των αισθητήρων συνδυαστικά με εκείνη. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

5. Αποτελέσματα & Συμπεράσματα

Συνοψίζοντας τα όσα έχουμε δει μέχρι τώρα στα προηγούμενα κεφάλαια και λαμβάνοντας υπόψιν μας την συμπεριφορά του κάθε αισθητήρα στα ερεθίσματα που τους δώσαμε, υπάρχει η δυνατότητα να προβούμε σε ορισμένα συμπεράσματα και παράλληλα να παρουσιάσουμε τα αποτελέσματα που συλλέχθηκαν, τα οποία αναλύονται παρακάτω.

Αρχικά, με τους συγκεκριμένους αισθητήρες που χρησιμοποιήθηκαν, δεν είναι δυνατό να μετράμε καρδιακούς παλμούς σε πραγματικό χρόνο και παράλληλα να χρησιμοποιούμε και τους υπόλοιπους πέντε αισθητήρες. Αυτό συμβαίνει, διότι για την χρήση των υπόλοιπων πέντε αισθητήρων (εκτός του αισθητήρα παλμών) και την αποστολή email στο ηλεκτρονικό ταχυδρομείο του συγγενή, πρέπει να προσθέσουμε μία καθυστέρηση (delay) στο πρόγραμμα του Arduino, που θα δίνει ένα περιθώριο ανάμεσα σε κάθε τι που θα ανιχνεύουν οι αισθητήρες. Θα αφήνει πιο απλά ένα κενό ορισμένων δευτερολέπτων το οποίο ορίζει ο χρήστης, έως ότου οι αισθητήρες ξαναπάρουν μέτρηση. Χωρίς την χρήση αυτού του delay, οι μετρήσεις θα είναι πιο ανακριβείς. Ακριβώς για αυτό, ο αισθητήρας καρδιακών παλμών δεν μπορεί να χρησιμοποιείται ταυτόχρονα με τους υπόλοιπους. Η μέτρηση των καρδιακών παλμών θα πρέπει να είναι όσο πιο ακριβής γίνεται και παράλληλα άμεση, χωρίς κανένα delay. Επομένως, η λύση στο συγκεκριμένο πρόβλημα θα μπορούσε να είναι ένας άλλος αισθητήρας που να δίνει την δυνατότητα αυτή, ή ακόμη και το χώρισμα των έξι αισθητήρων σε δύο συσκευές. Μία αυτόνομη, που θα αναλαμβάνει την μέτρηση μόνο των καρδιακών παλμών και άλλη μία που θα πληρεί τις υπόλοιπες λειτουργίες.

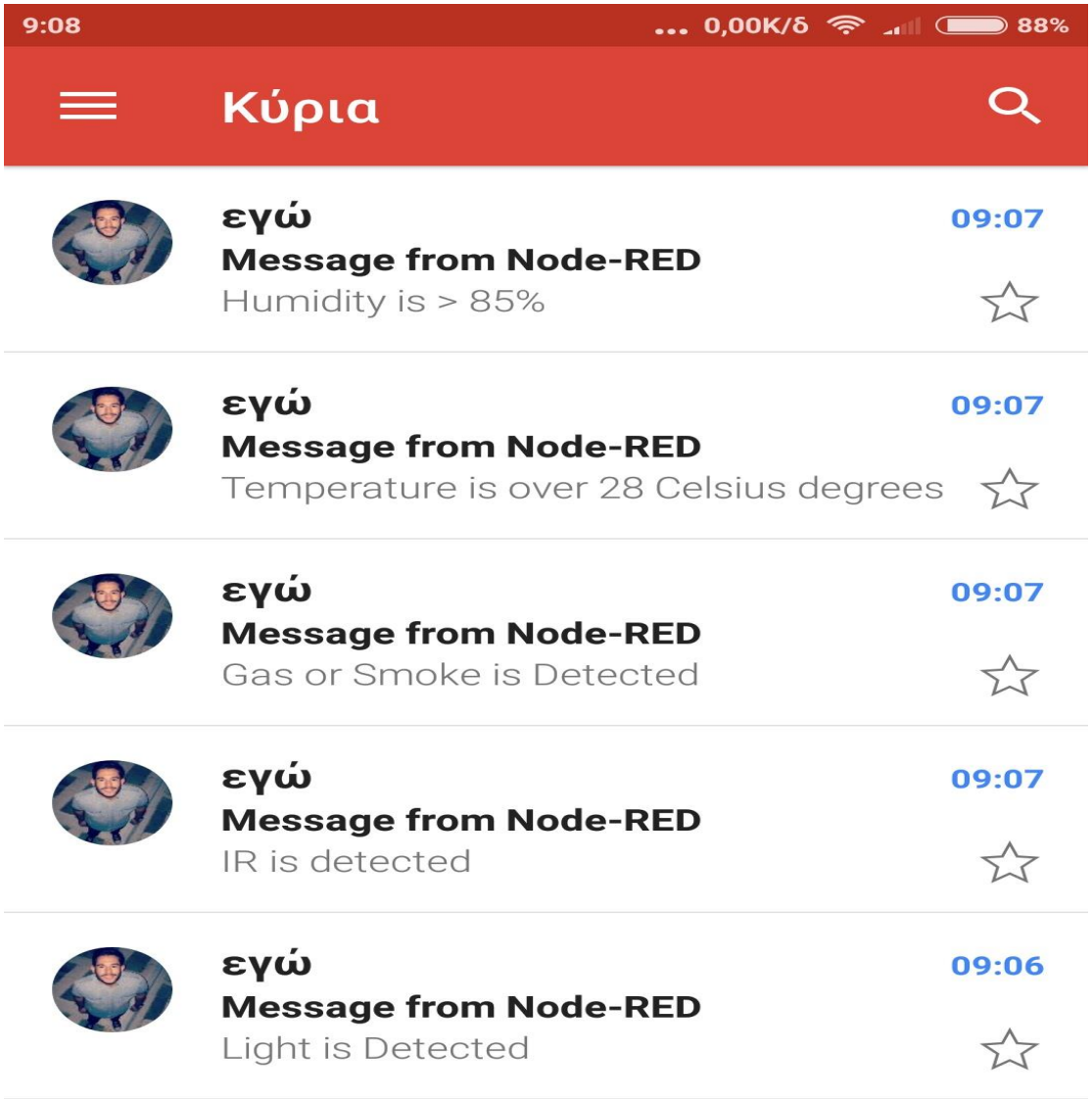
Όσων αφορά τον συγκεκριμένο αισθητήρα παλμών όμως, η γραφική απεικόνιση που δίνει όταν λειτουργεί, είναι ένα τέλειο καρδιογράφημα, πράγμα που μας δείχνει πως ακόμα και αν χρειαστούν δύο συσκευές για την ταυτόχρονη μέτρηση και των έξι αισθητήρων, θα είναι αξιόπιστες.



Εικόνα 5.1: Γραφική απεικόνιση καρδιογραφήματος από τον αισθητήρα παλμών. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

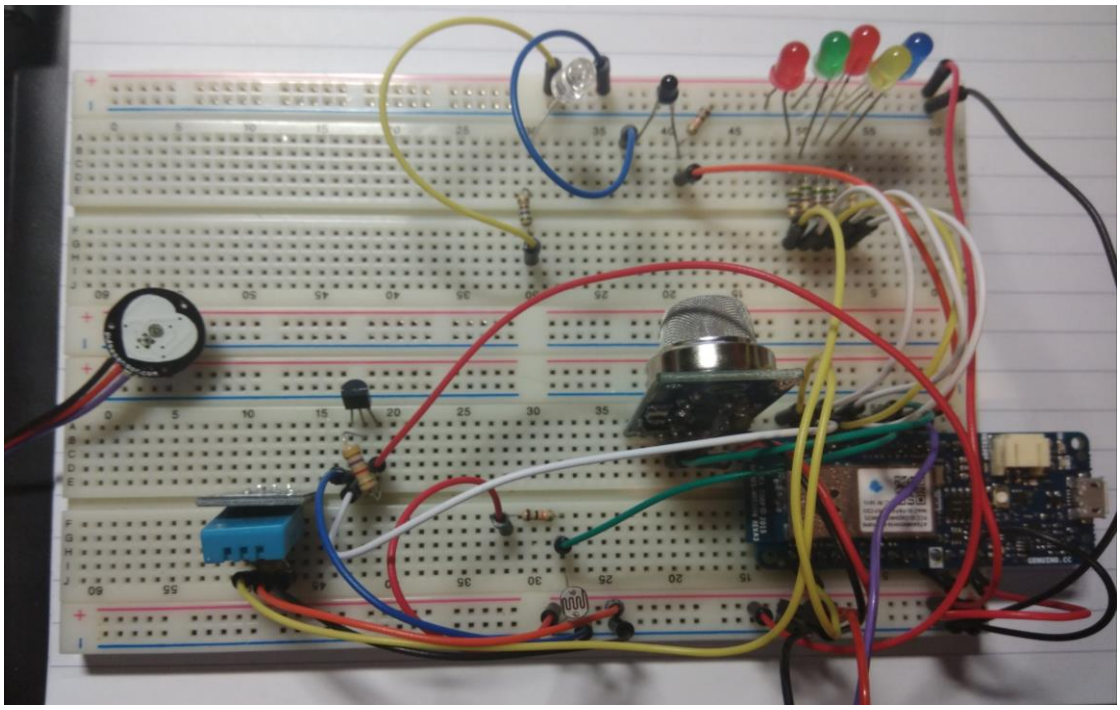
Σημαντική επισήμανση, που πρέπει να ληφθεί υπόψιν, είναι πως ο αισθητήρας υπερύθρων επηρεάζεται από την ηλιακή ακτινοβολία. Έπειτα από δοκιμές της συσκευής όταν εκείνη είχε φτάσει στο τελικό της στάδιο, παρατηρήθηκε πως στην έκθεση στην ηλιακή ακτινοβολία, ο αισθητήρας υπερύθρων (phototransistor) συμπεριφέρεται, ακριβώς όπως όταν στρέψουμε προς το μέρος του ένα Led υπερύθρων ή ένα τηλεκοντρόλ πατώντας τα κουμπιά του. Αυτό συμβαίνει διότι η ηλιακή ακτινοβολία, είναι και αυτή σε μεγάλο ποσοστό υπέρυθρη, πράγμα που θα πρέπει να συλλογιστεί κάποιος, όσων αφορά την χρήση που θα κάνει στον συγκεκριμένο αισθητήρα.

Επιπροσθέτως, οι τέσσερεις εναπομείναντες αισθητήρες, έδειξαν να λειτουργούν σε κάθε δοκιμή που τους έγινε και φυσικά πως είναι ικανοί να ανιχνεύσουν ο καθένας ξεχωριστά, αυτό για το οποίο αναγράφεται χωρίς κάποιο πρόβλημα. Ένα παράδειγμα των ειδοποιήσεων, όπου θα λαμβάνει ένα συγγενικό πρόσωπο του ασθενή στο ηλεκτρονικό του ταχυδρομείο μέσω της πλατφόρμας Node - Red, απεικονίζεται στην παρακάτω εικόνα.



Εικόνα 5.2: Παράδειγμα ειδοποίησης ηλεκτρονικού ταχυδρομείου συγγενή του ασθενή, μέσω της Node - Red. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

Κλείνοντας, στην παρακάτω εικόνα απεικονίζεται η ολοκληρωμένη συσκευή, που προέκυψε από τον συνδυασμό των έξι αισθητήρων και του Arduino / Genuino MKR1000.



Εικόνα 5.3: Ολοκληρωμένη συσκευή επιτήρησης ασθενή. Προσαρμόστηκε από: (Γκότσης & Παγώνης, 2018)

6. Συζήτηση & Μελλοντικές Προοπτικές

Βασική ιδέα της πτυχιακής εργασίας ήταν η κατασκευή μια συσκευής που θα απευθύνεται σε άτομα που πάσχουν από τη νόσο Αλτσχάιμερ με σκοπό τη διευκόλυνση της καθημερινότητάς τους. Για να επιτευχθεί αυτό, αρχικά χρειάστηκαν γνώσεις σχετικά με τη νόσο, τον τρόπο που επηρεάζει εκείνους που πάσχουν, τα είδη των προβλημάτων και δυσκολιών για τα ίδια τα άτομα αλλά και τον περίγυρό τους, ώστε οι παράμετροι της κατασκευής αυτής να συμβάλλουν εάν όχι στη εξάλειψη, έστω στη καταπολέμησή τους. Εφόσον όλα όσα αναφέρθηκαν προηγουμένως έγιναν κατανοητά, το επόμενο στάδιο ήταν η υλοποίηση της ιδέας, η οποία ήταν μια συσκευή που επιτηρεί τη συμπεριφορά του ασθενή, ενώ ταυτόχρονα κάποιο συγγενικό του πρόσωπο να μπορεί να έχει πρόσβαση στη συσκευή ακόμα και εξ ’αποστάσεως, ώστε να ενημερώνεται και να παρακολουθεί. Στη πράξη ο μηχανισμός αυτός, θα έπρεπε να είναι προσιτός έχοντας εύκολη χρήση, τόσο από τον ίδιο τον ασθενή, όσο και από τους επιτηρητές του. Πηγές έμπνευσης για την συγκεκριμένη πτυχιακή εργασία, υπήρξαν ήδη υπάρχοντα τεχνολογικά επιτεύγματα. Τέτοιου είδους καινοτομίες αποτελούν συσκευές, όπου έχουν ηχογραφηθεί ηχητικά μηνύματα για να υπενθυμίζουν στον ασθενή δραστηριότητες που ξεχνά να πραγματοποιήσει, είτε ανιχνευτές κίνησης όπως για παράδειγμα, όταν κάποιος φεύγει από το σπίτι να του υπενθυμίζει να κλειδώνει την εξώπορτα. Επίσης, υπάρχουν έξυπνα ρολόγια που αναγράφουν την ώρα, την ημερομηνία, τον μήνα και το έτος διαθέτοντας λογισμικό που προσφέρει ποικίλες δυνατότητες, όπως επιλογές μεγέθους γραμματοσειράς, ειδική διάταξη στο χώρο και φωτεινά χρώματα, ανάλογα με τον τρόπο διευκόλυνσης. Χρήσιμες είναι και οι συσκευές gps για τον συνεχή εντοπισμό της θέσης του ασθενούς. Έχοντας όλες αυτές τις πληροφορίες, έγινε η προσπάθεια εκπόνησης μιας παρόμοιας κατασκευής. Βέβαια υπήρχαν και δυσκολίες στην διεκπεραίωση αυτής της διαδικασίας. Ένα από αυτά ήταν ο περιορισμός των χρημάτων που διαθέτει το πανεπιστήμιο για την αγορά υλικών που χρειάζονταν για τη κατασκευή και γι’ αυτό χρησιμοποιήθηκαν πιο φθηνά, που θα προσέγγιζαν όμως τη βασική της ιδέα. Επίσης, ο διατιθέμενος χρόνος δεν άφηνε περιθώρια για περαιτέρω μελέτη πάνω στο αντικείμενο, που θα συνέβαλε σε συνεχείς βελτιστοποιήσεις και διορθώσεις της κατασκευής, για την πιο αποτελεσματική της λειτουργία. Δεδομένου όλο αυτών των παραγόντων, ο μηχανισμός κατάφερε να πετύχει το στόχο του, δηλαδή να πραγματοποιείται επιτήρηση ατόμων που πάσχουν από Αλτσχάιμερ, αλλά να βρίσκεται σε πρώιμο προπτυχιακό επίπεδο. Εάν υπήρχε η κατάλληλη τεχνογνωσία, αλλά και η διάθεση χρόνου θα μπορούσε να αξιοποιηθεί ακόμα περισσότερο. Όσων αφορά το κομμάτι των υλικών, θα χρησιμοποιούνταν καλύτερης ποιότητας εξαρτήματα και το κυριότερο θα γινόταν σμίκρυνση της κατασκευής. Αυτό θα μπορούσε να έχει εφαρμογή σε ένα έξυπνο ρολόι, όπου όλοι οι αισθητήρες και τα ηλεκτρονικά συστήματα θα βρίσκονται πάνω σε αυτό έχοντας τις ίδιες δυνατότητες, με όφελος την εξοικονόμηση του χώρου αλλά και την διευκόλυνση της χρήσης. Από την άλλη όσον αφορά το κομμάτι του λογισμικού, με τις κατάλληλες γνώσεις προγραμματισμού θα υπήρχε δυνατότητα σχεδιασμού ενός κώδικα, όπου ο προγραμματισμός λειτουργιών της συσκευής θα ήταν πιο εύκολος και η χρήση του θα ήταν πιο προσιτή στον χρήστη. Τέλος, η πολιτική του πανεπιστημίου που κρατά ως αρχείο όλες τις πτυχιακές εργασίες, παρέχει τη δυνατότητα στους φοιτητές να εργαστούν με ήδη προϋπάρχοντες εργασίες, κάνοντας έτσι τροποποιήσεις με σκοπό την βελτίωσή τους, καθώς και την εν καιρό δυνατότητα χρήσης τους στο μέλλον.

Αναφορές – Πηγές

[1] Maurer K., Maurer U. (2003). "*Alzheimer: The Life of a Physician and Career of a Disease*". New York: Columbia University Press. **ISBN 0-231-11896-1**

[2] Marder K (2012). "*Dementia and memory loss. In JCM Brust, ed., Current Diagnosis and Treatment Neurology*", 2nd ed., pp. 78-101. New York: NcGraw-Hill.

[3] McKhann GM, et al. (2011). "*The diagnosis of dementia due to Alzheimer's disease: Recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease*". Alzheimer's and Dementia, 7 (3): pp. 263-269.

[4] Sperling RA, et al. (2011). "*Toward defining the preclinical stages of Alzheimer's disease: Recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease*". Alzheimer's and Dementia, 7 (3): pp. 280-292.

[5] David Kushner (2011-10-26). "*The Making of Arduino*". IEEE Spectrum.

[6] Braak H, Braak E (1991). "*Neuropathological staging of Alzheimer-related changes*". Acta Neuropathol 82: pp. 239–259.

[7] Association, A. (2011). "*2011 Alzheimer's disease facts and figures*". Alzheimer's & Dementia, 7(2), pp. 208–244.

[8] Bertram, L., McQueen, M. B., Mullin, K., Blacker, D., & Tanzi, R. E. (2007). "*Systematic meta-analyses of Alzheimer disease genetic association studies: the AlzGene database*". Nature Genetics, 39(1), pp. 17–23.

[9] Clay Andres, et al. (2011). "*Beginning Arduino Programming*". Springer Science+Business Media. **ISBN 978-1-4302-3777-8**

[10] Steve Anglin, et al. (2009*).* "*Practical Arduino: Cool Projects for Open Source Hardware*". Springer-Verlag New York. **ISBN 978-1-4302-2477-8**

[11] Jonathan Gennick, et al. (2015). "*Beginning C for Arduino, Second Edition: Learn C Programming for the Arduino*". Springer Science+Business Media New York. **ISBN 978-1-4842-0941-7**

[12] Robert Hutchinson, et al. (2016). "*Building Arduino Projects for the Internet of Things: Experiments with Real-World Applications*". Springer Science+Business Media New York. **ISBN 978-1-4842-1939-3**

[13] Steve Anglin, et al. (2013). "*Beginning Arduino*". Springer Science+Business Media New York. **ISBN 978-1-4302-5016-6**

[14] Massimo Banzi (2011). "*Getting Started with Arduino*". Make: Books, an imprint of Maker Media, a division of O'Reilly Media. **ISBN: 978-1-449-309879**

[15] Holsinger RM, McLean CA, Beyreuther K, Masters CL, Evin G.(2002). "*Increased expression of the amyloid precursor beta-secretase in Alzheimer's disease. Ann Neurol*". **ISBN 51:783–786**

[16] Gotz J, Chen F, van Dorpe J, Nitsch RM(2001). "*Formation of neurofibrillary tangles in P301l tau transgenic mice induced by Abeta 42 fibrils".* **ISBN 293:1491–1495**