



# **ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**

**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**

**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ**

**ΥΠΟΛΟΓΙΣΤΩΝ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Ανάπτυξη PID ελεγκτή θερμοκρασίας με τη βοήθεια Raspberry Pi  
και Matlab**

**Κρίγκας Π. Σπυρίδων**

**Εισηγητής: Δρ Ιωάννης Έλληνας, Καθηγητής**

**ΑΘΗΝΑ**

**ΙΟΥΝΙΟΣ 2020**



**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Ανάπτυξη PID ελεγκτή θερμοκρασίας με τη βοήθεια Raspberry Pi και Matlab**

**Σπυρίδων Π. Κρίγκας**

**A.M. 43214**

**Εισηγητής:**

**Δρ Ιωάννης Έλληνας, Καθηγητής**

**Εξεταστική Επιτροπή:**

**Ημερομηνία εξέτασης:**



## **ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

Ο κάτωθι υπογεγραμμένος Κρίγκας Σπυρίδων, του Παναγιώτη, με αριθμό μητρώου 43214 φοιτητής του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών του Πανεπιστημίου Δυτικής Αττικής πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού δμήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»



## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο την κατασκευή ανάπτυξη PID ελεγκτή θερμοκρασίας με τη βοήθεια Raspberry Pi και Matlab. Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου, Ιωάννης Έλληνας τον οποίο θα ήθελα να ευχαριστήσω.





## ΠΕΡΙΛΗΨΗ

Στη παρούσα πτυχιακή εργασία, βασικός σκοπός είναι η ανάπτυξη ενός PID Controller για τον έλεγχο της θερμοκρασίας χρησιμοποιώντας το Raspberry Pi και το MATLAB. Αρχικά, παρουσιάζονται τα βασικά χαρακτηριστικά του Raspberry Pi καθώς και τα πλεονεκτήματα του από τους υπόλοιπους της αγοράς. Στην συνέχεια, παρουσιάζεται το περιβάλλον του MATLAB και των δυνατοτήτων του. Αμέσως μετά παρουσιάζεται η λειτουργικότητα των διάφορων ειδών PID ελεγκτών και πως ο κάθε ένας ελεγκτής επηρεάζει ένα κλειστό κύκλωμα. Έπειτα, παρουσιάζεται η δημιουργία της κατασκευής, η σύνδεση του MATLAB με το Raspberry Pi καθώς και τα μοντέλα που αναπτύχθηκαν στο Simulink του Matlab με σκοπό την εύρεση των κατάλληλων όρων του PID Controller για την επίτευξη της λειτουργίας του συστήματος σύμφωνα με τις επιθυμητές προδιαγραφές.

## ABSTRACT

In this dissertation, the main purpose is to develop a PID Controller for the temperature control using Raspberry Pi and MATLAB. At the first, the basic characteristics of the Raspberry Pi as well as its advantages from the rest of the market are presented. Then, the MATLAB environment and its capabilities are presented. Immediately after, the functionality of the different types of PID controllers is presented and how each controller affects a closed circuit. Then, the creation of the construction, the connection of MATLAB with the Raspberry Pi as well as the models developed in Simulink of Matlab are presented in order to find the appropriate terms of the PID Controller to achieve the operation of the system according to the desired specifications.

## ΠΕΡΙΕΧΟΜΕΝΑ

1.	ΚΕΦΑΛΑΙΟ 1 .....	15
1.1.	Εισαγωγή .....	15
1.1.1.	Περιγραφή του αντικειμένου της πτυχιακής εργασίας .....	15
1.2.	Ιστορική Αναδρομή .....	16
1.2.1.	Ιστορική Αναδρομή Raspberry Pi .....	16
1.2.2.	Ιστορική Αναδρομή Matlab .....	17
1.2.3.	Ιστορική Αναδρομή PID Controller .....	18
2.	ΚΕΦΑΛΑΙΟ 2 .....	22
2.1.	Raspberry Pi .....	22
2.1.1.	Εισαγωγή στους Μικροελεγκτές .....	22
2.1.2.	Εισαγωγή στο Raspberry Pi .....	24
2.1.3.	Μοντέλα Raspberry Pi .....	25
2.1.4.	Ακίδες των Raspberry Pi .....	31
2.1.5.	Λογισμικό .....	32
2.1.6.	Πλεονεκτήματα και Μειονεκτήματα του Raspberry Pi .....	35
3.	ΚΕΦΑΛΑΙΟ 3 .....	36
3.1.	MATLAB .....	36
3.1.1.	Εισαγωγή στο Matlab .....	36
3.1.2.	Στοιχεία του Matlab .....	39
3.1.3.	Χρήση Του Matlab .....	42
3.1.4.	Support Package Matlab .....	44
4.	ΚΕΦΑΛΑΙΟ 4 .....	46
4.1.	PID Controller .....	46
4.1.1.	Εισαγωγή στους PID Controller .....	46
4.1.2.	Σχεδίαση Ελεγκτών PID .....	50
4.1.3.	Είδη Ελεγκτών PID .....	54
5.	ΚΕΦΑΛΑΙΟ 5 .....	60
5.1.	Υλικά Κατασκευής .....	60
5.1.1.	Raspberry Pi .....	61
5.1.2.	Micro SD Card .....	62
5.1.3.	LCD Module Display .....	63
5.1.4.	DALLAS 18B20 .....	63
5.1.5.	Fan 12V .....	65

5.1.6. Βαπτική Αντίσταση .....	65
5.1.7. Transistors.....	66
5.1.8. Υπόλοιπα υλικά .....	66
5.2. Εγκατάσταση Raspberry Pi .....	67
5.3. Κώδικας Εφαρμογής .....	72
6. ΒΙΒΛΙΟΓΡΑΦΙΑ.....	77

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1 – Raspberry Pi 4 .....	16
Εικόνα 2 – Matlab Logo .....	18
Εικόνα 3 – PID Controller .....	19
Εικόνα 4 – Logo Raspberry Pi .....	24
Εικόνα 5 – Raspberry Pi 1 Model B .....	25
Εικόνα 6 – Raspberry Pi 1 Model A .....	26
Εικόνα 7 – Raspberry Pi 1 Model B + .....	26
Εικόνα 8 – Raspberry Pi 1 Model A + .....	27
Εικόνα 9 – Raspberry Pi 2 Model B V1.2 .....	28
Εικόνα 10 – Raspberry Pi Zero .....	28
Εικόνα 11 – Raspberry Pi 3 Model B .....	29
Εικόνα 12 – Raspberry Pi 4 Model B .....	30
Εικόνα 13 – Ακίδες Raspberry Pi 3 Model B .....	31
Εικόνα 14 – Raspbian .....	33
Εικόνα 15 – Raspbmc .....	33
Εικόνα 16 – Pidora .....	34
Εικόνα 17 – Noobs .....	34
Εικόνα 18 – Matlab .....	37
Εικόνα 19 – Matlab Εντολή help .....	38
Εικόνα 20 – Matlab Εντολή helpdesk .....	38
Εικόνα 21 – Matlab Εντολή demo .....	39
Εικόνα 22 – Περιβάλλον Matlab .....	40
Εικόνα 23 – Περιβάλλον Simulink .....	43
Εικόνα 24 – Εγκατάσταση Support Package .....	44
Εικόνα 25 – Εγκατάσταση Support Package Raspberry .....	45
Εικόνα 26 – Τύποι PID Controller .....	46
Εικόνα 27 – Μαθηματικοί Τύποι PID Controller .....	48
Εικόνα 28 – PID Controller .....	55
Εικόνα 29 – Συνάρτηση Παραδείγματος .....	57
Εικόνα 30 – Εντολές Παραδείγματος στο Matlab .....	57
Εικόνα 31 – Απόκριση PID Controller .....	58
Εικόνα 32 – Απόκριση PI Controller .....	58
Εικόνα 33 – Απόκριση PD Controller .....	59
Εικόνα 34 – Κατασκευή .....	61
Εικόνα 35 – Raspberry Pi Model B .....	62
Εικόνα 36 – SD Card .....	62
Εικόνα 37 – LCD Display .....	63
Εικόνα 38 – DS18B20 .....	64
Εικόνα 39 – Fan 12V .....	65
Εικόνα 40 – Βαπτική Αντίσταση 10W10ΩJW .....	65
Εικόνα 41 – Transistor BD243C και 2N2222A .....	66

Εικόνα 42 – Εγκατάσταση Λειτουργικό στο Raspberry Pi .....	67
Εικόνα 43 – Εγκατάσταση Λειτουργικό στο Raspberry Pi .....	68
Εικόνα 44 – Εγκατάσταση Λειτουργικό στο Raspberry Pi .....	69
Εικόνα 45 – Αρχική οθόνη Raspberry Pi .....	69
Εικόνα 46 – Login χρδρ .....	70
Εικόνα 47 – Logo Python .....	72

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Επίδραση ελεγκτών P, I, D .....	49
Πίνακας 2: Παράμετροι Ziegler – Nichols .....	51
Πίνακας 3: Παράμετροι Tyreus – Luyben .....	51

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

**MATLAB:** Matrix Laboratory

**PID:** Proportional Integral Derivative

**LCD:** Liquid Crystal Display

**SD:** Secure Digital

**ROM:** Read Only Memory

**RAM:** Random Access Memory

**Mb:** Megabyte

**GB:** Gigabyte

**MHz:** Megaheartz

**GHz:** Gigaheartz

**USB:** Universal Serial Bus

**HDMI:** High Definition Multimedia Interface

**Wi-Fi:** Wireless Fidelity

**V:** Volt

## 1. ΚΕΦΑΛΑΙΟ 1

### 1.1. Εισαγωγή

#### 1.1.1. Περιγραφή του αντικειμένου της πτυχιακής εργασίας

Σε αυτό το κεφάλαιο θα εξηγήσουμε με απλά λόγια το θέμα της πτυχιακής εργασίας. Ο τίτλος της εργασίας είναι «Ανάπτυξη PID ελεγκτή θερμοκρασίας με τη βοήθεια Raspberry Pi και Matlab». Στην εργασία θα ασχοληθούμε με το Raspberry Pi, το Matlab και τον PID Controller.

Στόχος είναι η διατήρηση της θερμοκρασία σταθερή, σύμφωνα με την επιθυμητή τιμή που θα ορίσουμε και η απόκριση του συστήματος να είναι όσο το δυνατόν πιο ομαλή γίνεται. Για να μπορέσουμε να ελέγξουμε αυτή την θερμοκρασία, θα χρειαστούμε έναν αισθητήρα που θα μετράει την πραγματική θερμοκρασία. Το σύστημα θα υπολογίζει την διαφορά μεταξύ της επιθυμητής τιμής και της πραγματικής θερμοκρασίας.

Για την κατασκευή θα χρειαστούμε ορισμένα εξαρτήματα. Αρχικά, ένα Raspberry Pi, μια βαπτική αντίσταση, η οποία θα θερμαίνεται για να φτάσει η θερμοκρασία στην επιθυμητή θερμοκρασία που θέλουμε, ένα fun που θα κατεβάζει τη θερμοκρασία και έναν αισθητήρα θερμοκρασίας όπου θα ελέγχει αυτή την θερμοκρασία. Ακόμα, θα χρειαστούμε μια LCD Display η οποία θα μας εμφανίζει τη θερμοκρασία.

Για το MATLAB θα χρειαστούμε το Support Package for Raspberry Pi Hardware το οποίο μας επιτρέπει να δημιουργούμε και να εκτελούμε μοντέλα Simulink στον Raspberry Pi.



## 1.2. Ιστορική Αναδρομή

### 1.2.1. Ιστορική Αναδρομή Raspberry Pi

Το Raspberry Pi είναι ένας υπολογιστής σε μέγεθος πιστωτικής κάρτας. Δημιουργήθηκε στο εργαστήριο πληροφορικής του πανεπιστημίου Cambridge το 2006, από τον Eben Upton με τη βοήθεια από τους Rob Mullins, Jack Lang και Alan Mycroft, και δημιουργώντας την Raspberry Pi Foundation. Για τα επόμενα 6 χρόνια η ομάδα εργάστηκε για την κατασκευή μιας φθηνής και προσβάσιμης συσκευής με σκοπό την προώθηση της διδασκαλίας της επιστήμης των υπολογιστών στα σχολεία.

Η αρχική εμπορική κυκλοφορία του Raspberry Pi ήταν τον Φεβρουάριο του 2012. Είναι μια συσκευή που έχει πολλές δυνατότητες και λειτουργεί όπως ένας προσωπικός υπολογιστής, όπως πλοήγηση στο Internet, προβολή ταινιών, παιχνίδια κλπ. Έχουν κυκλοφορήσει πέντε μοντέλα, το Raspberry Pi 1, το Raspberry Pi 2, το Raspberry Pi Zero, το Raspberry Pi 3 ενώ πρόσφατα κυκλοφόρησε και το Raspberry Pi 4.



Εικόνα 1 – Raspberry Pi 4

Στο Raspberry Pi χρησιμοποιείς μια SD Card για αποθηκευτικό χώρο, καθώς είναι απαραίτητη για τη διαδικασία του boot, διότι όλα τα απαραίτητα αρχεία για την διαδικασία αυτή είναι αποθηκευμένα στην κάρτα αυτή με μορφή FAT32 Partition. Στο Raspberry Pi, υπάρχει ένας Header ο οποίος διαθέτει 40 ακίδες. Πρόκειται για τον P1 Header. Στην πλακέτα υπάρχουν ακόμη 5 Headers αλλά δεν είναι ενεργοποιημένοι.

Ο P1 Header λέγεται αλλιώς expansion header gpio connector P1. Στο Header υπάρχουν ακίδες τροφοδοσίας 5V, 3.3V, GND και ακίδες που επιτρέπουν την επικοινωνία και την σύνδεση του Raspberry Pi με άλλες συσκευές. Για την τροφοδοσία του Raspberry Pi έχει μία θύρα micro usb 5V.

### **1.2.2. Ιστορική Αναδρομή Matlab**

Το MATLAB (Matrix Laboratory) είναι ένα περιβάλλον αριθμητικής υπολογιστικής με προγραμματιστική γλώσσα τέταρτης γενιάς. Η τρέχουσα έκδοση του είναι η R2018b η οποία κυκλοφόρησε τον Σεπτέμβριο του 2018. Χρησιμοποιείται κατά κύριο λόγο για την επίλυση μαθηματικών προβλημάτων αλλά και τη σχεδίαση λειτουργιών, υλοποίηση αλγορίθμων, δημιουργία διεπαφών χρήστη και διασύνδεση προγραμμάτων γραμμένα σε άλλες γλώσσες, όπως C, C++, C#, Java, Fortran και Python.

Το MATLAB προορίζεται κυρίως για αριθμητικούς υπολογισμούς, παρόλο αυτά με την προσθήκη της βιβλιοθήκης MuPAD, επιτρέπουν την πρόσβαση σε συμβολικές υπολογιστικές ικανότητες. Ένα πρόσθετο πακέτο είναι το Simulink, στο οποίο μπορείς να προσθέσεις γραφικές προσομοιώσεις, σχεδίαση με βάση το μοντέλο για δυναμικά και ενσωματωμένα συστήματα.



Εικόνα 2 - Matlab Logo

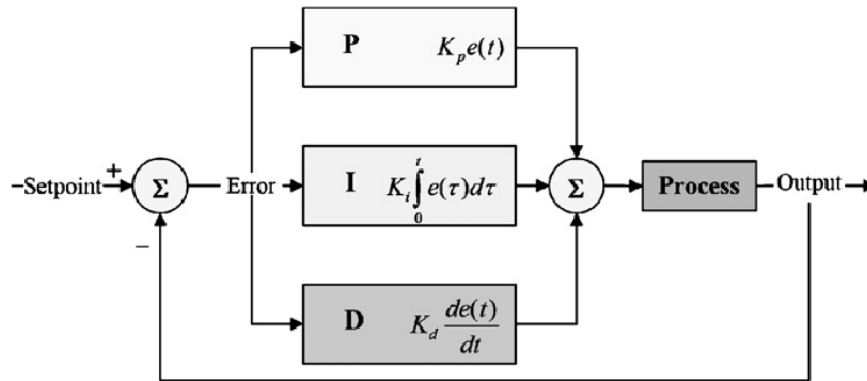
Το MATLAB ξεκίνησε στα τέλη της δεκαετίας του 1970 από τον Cleve Morel, πρόεδρος του τμήματος πληροφορικής του πανεπιστημίου του Νέου Μεξικού, για να δώσει πρόσβαση στους μαθητές του σε LINPACK και EISPACK χωρίς να χρειάζεται να μάθουν Fortran. Μετά από λίγο καιρό εξαπλώθηκε και σε άλλα πανεπιστήμια βρίσκοντας ένα ισχυρό κοινό στο πλαίσιο της εφαρμοσμένης μαθηματικής κοινότητας. Το 1984, ο Jack Little ενώθηκε με τον Cleve Morel και τον Steve Bangert και επανέγραψαν το MATLAB σε C και ίδρυσαν την MathWorks. Αυτές οι βιβλιοθήκες είναι γνωστές ως JACKPAC. Το 2000, το MATLAB ξαναγράφηκε για να χρησιμοποιήσει ένα νεότερο σύνολο βιβλιοθηκών, το LAPACK.

Το MATLAB υιοθετήθηκε για πρώτη φορά από ερευνητές και επαγγελματίες, αλλά γρήγορα εξαπλώθηκε σε πολλούς άλλους τομείς. Τώρα χρησιμοποιείται στην εκπαίδευση, ιδιαίτερα στη διδασκαλία της γραμμικής άλγεβρας και της αριθμητικής ανάλυσης. Είναι δημοφιλής στους επιστήμονες που ασχολούνται με την επεξεργασία εικόνων.

### 1.2.3. Ιστορική Αναδρομή PID Controller

Ο PID Controller (Proportional Integral Derivative) είναι ένας μηχανισμός ελέγχου συστήματος που χρησιμοποιείται ευρέως σε βιομηχανικά συστήματα ελέγχου και σε πολλές άλλες εφαρμογές που απαιτούν συνεχή διαμόρφωση ελέγχου. Ένας PID Controller υπολογίζει συνεχώς μια τιμή σφάλματος  $e(t)$  ως τη

διαφορά μεταξύ ενός επιθυμητού σημείου ρύθμισης (SP) και μιας μεταβλητής διεργασίας (PV) και εφαρμόζει μια διόρθωση βάσει αναλογικού, ολοκληρωμένου και παραγώγου όρο (P, I, D), αντίστοιχα.



Εικόνα 3 – PID Controller

Στην πράξη, εφαρμόζει αυτόματα ακριβή διόρθωση σε μια λειτουργία ελέγχου. Ένα καθημερινό παράδειγμα είναι το cruise control σε ένα αυτοκίνητο. Ο αλγόριθμος του PID Controller, επαναφέρει τη μετρούμενη ταχύτητα στην επιθυμητή ταχύτητα με ελάχιστη καθυστέρηση και υπέρβαση, αυξάνοντας την ισχύ του κινητήρα.

Αυτό το είδος ελέγχου εφευρέθηκε τον 17ο αιώνα από τον Christian Huyhens, ο οποίος ήθελε να ρυθμίσει το χάσμα μεταξύ των μυλόπετρών στους ανεμόμυλους ανάλογα με την ταχύτητα της περιστροφής με σκοπό να αντισταθμίσει την μεταβλητή ταχύτητα της τροφοδοσίας σιτηρών.

Όταν εφευρέθηκε η σταθερή ατμομηχανή υψηλής πίεσης, υπήρξε ανάγκη για τον αυτόματο έλεγχο της ταχύτητας. Δημιουργήθηκε από τον James Watt, ο αυτοδιαμορφωμένος ρυθμιστής conical pendulum, ένα σύνολο περιστρεφόμενων χαλύβδινων σφαιρών που συνδέονται με έναν κάθετο άξονα με βραχίονες σύνδεσης.

Τον 19ο αιώνα, ο James Clerk Maxwell περιέγραψε για πρώτη φορά την θεωρητική βάση για την λειτουργία των ρυθμιστών. Διερεύνησε τη μαθηματική βάση

για τη σταθερότητα του ελέγχου και έκανε βήματα προς μια λύση, αλλά έκανε έκκληση στους μαθηματικούς να εξετάσουν το πρόβλημα. Το πρόβλημα αυτό, εξετάστηκε από τον Edward Routh, τον Charles Sturm και τον Adolf Hurwitz. Όλοι οι παραπάνω βοήθησαν στην καθιέρωση των κριτηρίων σταθερότητας ελέγχου, όμως βελτιώθηκαν περαιτέρω, ιδίως από τον Αμερικανό επιστήμονα Willard Gibbs που ανέλυσε θεωρητικά τον ρυθμιστή του Watt.

Το 1922 αναπτύχθηκε για πρώτη φορά η επίσημη θεωρητική ανάλυση για τον PID ελεγκτή από τον μηχανικό Nicolas Minorsky. Ο Minorsky ερεύνησε και σχεδίασε την αυτόματη καθοδήγηση πλοίου του Πολεμικού Ναυτικού των ΗΠΑ, στηριζόμενος στην ανάλυση του σε παρατηρήσεις ενός πηδαλιούχου. Σημείωσε πως το πηδάλιο οδήγησε το πλοίο βάσει του συνδυασμού του τρέχοντος σφάλματος πορείας, του παρελθοντικού σφάλματος αλλά και του ρυθμού αλλαγής αυτού. Στην συνέχεια, αυτή η θεωρία μοντελοποιήθηκε μαθηματικά από τον ίδιο. Στόχος του ήταν η σταθερότητα. Παρότι ο αναλογικός έλεγχος παρείχε σταθερότητα έναντι των μικρών διαταραχών, δεν επαρκούσε για την αντιμετώπιση μιας σταθερής διαταραχής και ήταν απαραίτητη η προσθήκη του ολοκληρωτικού όρου. Τέλος, προστέθηκε ο διαφορικός όρος για την βελτίωση της σταθερότητας και του ελέγχου.

Οι δοκιμές πραγματοποιήθηκαν στο USS New Mexico με τους ελεγκτές να ελέγχουν την γωνιακή ταχύτητα του πηδαλίου. Έπειτα από τις δοκιμές, διαπιστώθηκε πως ο έλεγχος PI απέδωσε σταθερή στροφή  $\pm 2^\circ$ , ενώ η προσθήκη του διαφορικού όρου οδήγησε σε σφάλμα εκτροπής  $\pm 1/6^\circ$ . Τελικά όμως, το Πολεμικό Ναυτικό δεν υιοθέτησε το σύστημα λόγω της αντίστασης του προσωπικού.

Κατά τη διάρκεια του περασμένου μισού αιώνα, οι ερευνητές αναζητούσαν την επόμενη βασική τεχνολογία για τον συντονισμό PID Controller και την ομαλή του υλοποίηση. Η λειτουργία Computerizing, επιτρέπει την αυτόματη διεξαγωγή προσομοιώσεων, γεγονός που διευκολύνει την αναζήτηση των καλύτερων δυνατών ρυθμίσεων PID για την εφαρμογή. Μια προσέγγιση βασισμένη στην προσομοίωση, δεν απαιτεί τεχνητή ελαχιστοποίηση του εύρους ελέγχου και βοηθά στη βελτίωση της βραδείας παροδικής απόκρισης του ελεγκτή (Microcontroller, 2009). Κατά την αντιμετώπιση των προβλημάτων PID Controller, είναι επιθυμητό να

χρησιμοποιηθούν τυποποιημένες δομές PID για εύλογο φάσμα τύπων εγκαταστάσεων και λειτουργιών. Η προσαρμογή γύρω από τις τυποποιημένες δομές PID Controller, θα πρέπει επίσης να συμβάλει στη βελτίωση της οικονομικής αποδοτικότητας του ελέγχου και της συντήρησης PID. Με αυτόν τον τρόπο, μπορούν να αναπτυχθούν εξαιρετικά βέλτιστες μέθοδοι σχεδιασμού. Με τη ύπαρξη τεχνικών ταυτοποίησης του συστήματος, ολόκληρη η διαδικασία σχεδιασμού και συντονισμού PID Controller μπορεί να αυτοματοποιηθεί και μπορούν να διατεθούν αρθρωτά μπλοκ κώδικα για έγκαιρη εφαρμογή και προσαρμογή σε πραγματικό χρόνο.

Στη συνέχεια οι ελεγκτές PID χρησιμοποιήθηκαν για αυτόματο έλεγχο διεργασιών στη βιομηχανία. Σήμερα, το PID Controller χρησιμοποιείται σε εφαρμογές που απαιτούν ακριβές και βελτιστοποιημένο αυτόματο έλεγχο.

## 2. ΚΕΦΑΛΑΙΟ 2

### 2.1. Raspberry Pi

#### 2.1.1. Εισαγωγή στους Μικροελεγκτές

Ο μικροελεγκτής είναι ένας τύπος επεξεργαστή, ο οποίος μπορεί να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα, λόγω των πολλών ενσωματωμένων υποσυστημάτων που διαθέτει. Χρησιμοποιείται ευρύτατα σε όλα τα ενσωματωμένα συστήματα ελέγχου χαμηλού και μεσαίου κόστους, όπως αυτά που χρησιμοποιούνται σε αυτοματισμούς, ηλεκτρονικά καταναλωτικά προϊόντα, ηλεκτρικές συσκευές και κάθε είδους αυτοκινούμενα τροχοφόρα οχήματα.

Είναι απλούστερα σχεδιασμένος διότι είναι υπεύθυνος για την εκτέλεση μόνο μιας εργασίας ελέγχου ενός απλού συστήματος και έτσι περιλαμβάνει όλες τις απαιτούμενες λειτουργίες ενός ολοκληρωμένου κυκλώματος. Στην σημερινή εποχή, το κάθε προϊόν που αλληλοεπιδρά με έναν χρήστη περιλαμβάνει έναν μικροελεγκτή, ο οποίος παίζει το ρόλο του «εγκεφάλου» των ηλεκτρονικών κυκλωμάτων.

Στον μικροεπεξεργαστή, το ολοκληρωμένο κύκλωμα που τον αποτελεί περιέχει μόνο την Λογική και Αριθμητική Μονάδα (ALU), προσωρινή μνήμη RAM, πολύ υψηλής ταχύτητας (cache memory) και, κάποιες φορές, τον ελεγκτή μνήμης (memory controller). Όμως, για τη λειτουργία ενός πλήρους ενσωματωμένου υπολογιστικού συστήματος, απαιτούνται πολλά εξωτερικά υποσυστήματα και περιφερειακά. Κάποια από αυτά είναι:

- Μνήμη προγράμματος (τύπου ROM, FLASH, EPROM κλπ.) η οποία περιέχει το λογισμικό του συστήματος.
- Μεγάλο μέγεθος μνήμης RAM.
- Μόνιμη μνήμη αποθήκευσης παραμέτρων λειτουργίας (τύπου EEPROM ή NVRAM) η οποία να μπορεί να γράφεται στον πυρήνα του μικροελεγκτή.
- Κύκλωμα αρχικοποίησης (est).

- Διαχειριστή αιτήσεων διακοπής (interrupt request controller) από τα περιφερειακά.
- Κύκλωμα επιτήρησης λειτουργίας (watchdog timer) το οποίο αρχικοποιεί το σύστημα.
- Έναν ή περισσότερους χρονιστές-απαριθμητές υψηλής ταχύτητας (hardware timer-counter) για τη δημιουργία καθυστερήσεων, μέτρηση διάρκειας γεγονότων, απαρίθμηση γεγονότων και άλλων λειτουργιών ακριβούς χρονισμού.
- Ρολόι πραγματικού χρόνου (Real Time Clock, RTC) το οποίο τροφοδοτείται από ανεξάρτητη μπαταρία και για αυτό πρέπει να έχει πολύ χαμηλή κατανάλωση ρεύματος.
- Σειρά ανεξάρτητων ψηφιακών εισόδων και εξόδων (Parallel Input-Output, PIO).

Γενικά, όλες οι οικογένειες μικροελεγκτών ενσωματώνουν τα περισσότερα από τα παραπάνω περιφερειακά, με διαφοροποιήσεις κυρίως στην ύπαρξη ή μη εσωτερικής μνήμης προγράμματος και στο είδος της. Έτσι, υπάρχουν:

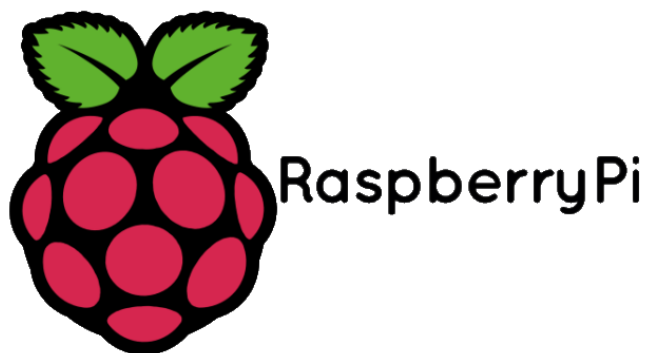
- Μικροελεγκτές χωρίς μνήμη προγράμματος, οι οποίοι χαρακτηρίζονται ως ROM-less. Αυτοί παρέχουν πάντοτε μια παράλληλη αρτηρία δεδομένων, πάνω στην οποία συνδέονται εξωτερικές μνήμες προγράμματος και RAM. Τέτοιοι τύποι μικροελεγκτών προορίζονται για πιο ισχυρά υπολογιστικά συστήματα ελέγχου, με μεγαλύτερες απαιτήσεις μνήμης.
- Μικροελεγκτές με μνήμη ROM, η οποία κατασκευάζεται με το λογισμικό της ή γράφεται μόνο μια φορά. Παρέχουν τη δυνατότητα πολύ χαμηλού κόστους, όταν αγοράζονται σε πολύ μεγάλες ποσότητες.
- Μικροελεγκτές με μνήμη FLASH, η οποία μπορούν συνήθως να προγραμματιστεί πολλές φορές. Αυτή είναι η πιο διαδεδομένη κατηγορία. Συχνά ο προγραμματισμός της μνήμης μπορεί να γίνει ακόμη και πάνω στο κύκλωμα της ενσωματωμένης εφαρμογής. Αυτοί οι μικροελεγκτές έχουν ουσιαστικά αντικαταστήσει τους παλαιότερους τύπους EPROM που έσβηναν με υπεριώδη ακτινοβολία.



Η πιο διαδεδομένη γλώσσα προγραμματισμού των μικροελεγκτών είναι η C, η C++ και οι παραλλαγές τους. Σε τμήματα του λογισμικού όπου απαιτείται μικρό μέγεθος χρησιμοποιούμενης μνήμης ή ταχύτητα, χρησιμοποιείται η Assembly. Ο προγραμματισμός σε C είναι πιο εύκολος από την Assembly και για αυτόν το λόγο η Assembly δεν χρησιμοποιείται συχνά.

### 2.1.2. Εισαγωγή στο Raspberry Pi

Το Raspberry Pi αναπτύχθηκε από την Raspberry Pi Foundation με σκοπό την προώθηση της διδασκαλίας της επιστήμης των υπολογιστών στα σχολεία. Είναι μια συσκευή που έχει πολλές δυνατότητες και λειτουργεί όπως ένας προσωπικός υπολογιστής, όπως πλοήγηση στο Internet, προβολή ταινιών, παιχνίδια κλπ. Έχουν κυκλοφορήσει πέντε μοντέλα, το Raspberry Pi 1, το Raspberry Pi 2, το Raspberry Pi Zero, το Raspberry Pi 3 ενώ πρόσφατα κυκλοφόρησε και το Raspberry Pi 4.



Εικόνα 4 – Logo Raspberry Pi

Το Raspberry Pi πρόκειται για έναν υπολογιστή σε μέγεθος της παλάμης, με πολύ μικρό κόστος αγοράς και χρήσης. Για να ολοκληρωθεί όμως και να λειτουργήσει ως συσκευή θα πρέπει να προμηθευτείτε κάποιες περιφερειακές μονάδες. Για αρχή θα χρειαστεί ρεύμα ενώ στην συνέχεια και ανάλογα την χρήση θα χρειαστείτε, οθόνη/τηλεόραση, πληκτρολόγιο, ποντίκι, αποθηκευτικό μέσο (π.χ. κάρτα μνήμης, USB stick). Επιπλέον, μπορείτε να προμηθευτείτε κουτί για την πλακέτα του Raspberry Pi και πολλές άλλες μονάδες επέκτασης που προσθέτουν

Ανάπτυξη PID ελεγκτή θερμοκρασίας με τη βοήθεια Raspberry Pi και Matlab

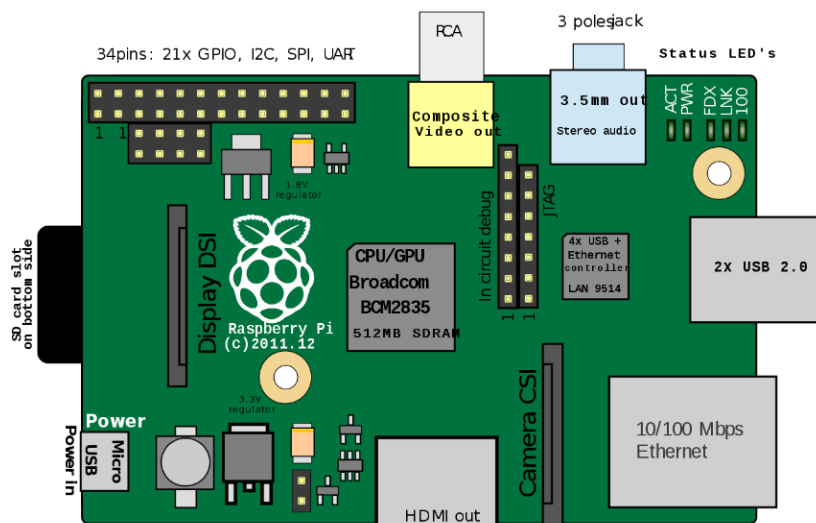
επιπλέον λειτουργίες στην πλακέτα και μας επιτρέπουν να εκτελούμε διαφορετικές εργασίες.

### 2.1.3. Μοντέλα Raspberry Pi

Μέχρι τώρα, έχουν κυκλοφορήσει πέντε μοντέλα Raspberry Pi, το Raspberry Pi 1, το Raspberry Pi 2, το Raspberry Pi Zero, το Raspberry Pi 3 και το Raspberry Pi 4 όπου το κάθε μοντέλο έχει τις δικές του εκδόσεις.

- **Raspberry Pi 1**

Τον Απρίλιο του 2012 κυκλοφόρησε το Raspberry Pi 1 Model B με τον μικροελεγκτή BCM2835, μονοπύρηνο επεξεργαστή ARM1176JZF-S (ARMv6Z 32-bit) στα 700MHz, μνήμη 256 Mb (τον Οκτώβριο του 2012 αλλάχτηκε σε 512Mb), 34 ακίδες, 2 θύρες USB 2.0, 1 θύρα Ethernet και 1 θύρα HDMI.

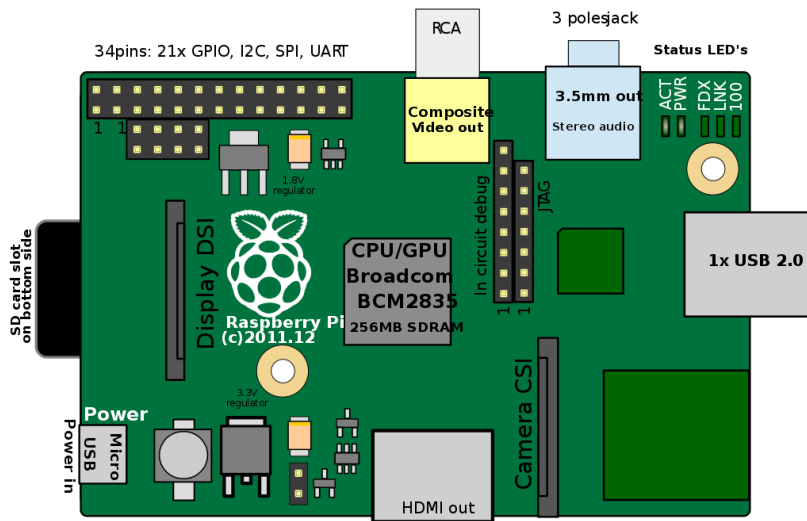


Εικόνα 5 – Raspberry Pi 1 Model B

Το Φεβρουάριο του 2013 κυκλοφόρησε το Raspberry Pi 1 Model A, το οποίο ήταν πιο οικονομικό από το προηγούμενο, με τον ίδιο μικροελεγκτή BCM2835 και μονοπύρηνο επεξεργαστή (ARM1176JZF-S (ARMv6Z 32-bit) στα 700MHz), μνήμη

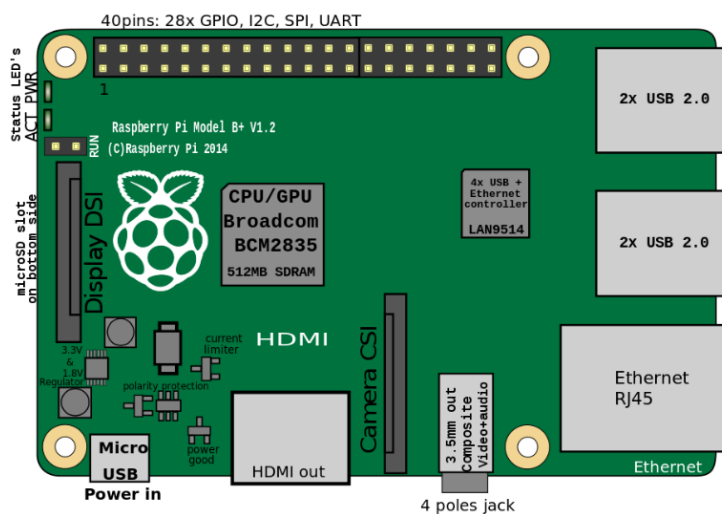
Ανάπτυξη PID ελεγκτή θερμοκρασίας με τη βοήθεια Raspberry Pi και Matlab

256 Mb, 34 ακίδες, 1 θύρα USB 2.0 και 1 θύρα HDMI. Η συγκεκριμένη έκδοση δεν είχε θύρα Ethernet.



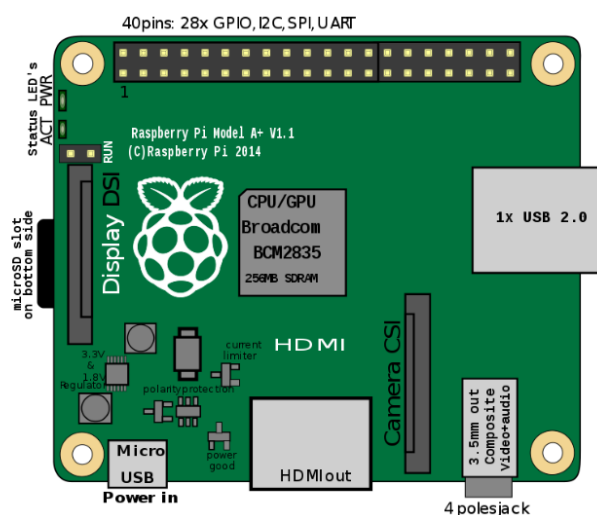
Εικόνα 6 – Raspberry Pi 1 Model A

Τον Ιούλιο του 2014 κυκλοφόρησε το Raspberry Pi 1 Model B + με τον μικροελεγκτή BCM2835, με μονοπύρηνο επεξεργαστή ARM1176JZF-S (ARMv6Z 32-bit) στα 700MHz, μνήμη 512 Mb, 40 ακίδες, 4 θύρες USB 2.0, 1 θύρα Ethernet και 1 θύρα HDMI.



Εικόνα 7 – Raspberry Pi 1 Model B +

Το Νοέμβριο του 2014 κυκλοφόρησε το Raspberry Pi 1 Model A + με τον μικροελεγκτή BCM2835, με μονοπύρηνο επεξεργαστή ARM1176JZF-S (ARMv6Z 32-bit) στα 700MHz, μνήμη 256 Mb (τον Αύγουστο του 2016 αλλάχτηκε σε 512Mb), 40 ακίδες, 1 θύρα USB 2.0 και 1 θύρα HDMI. Η συγκεκριμένη έκδοση δεν είχε θύρα Ethernet.

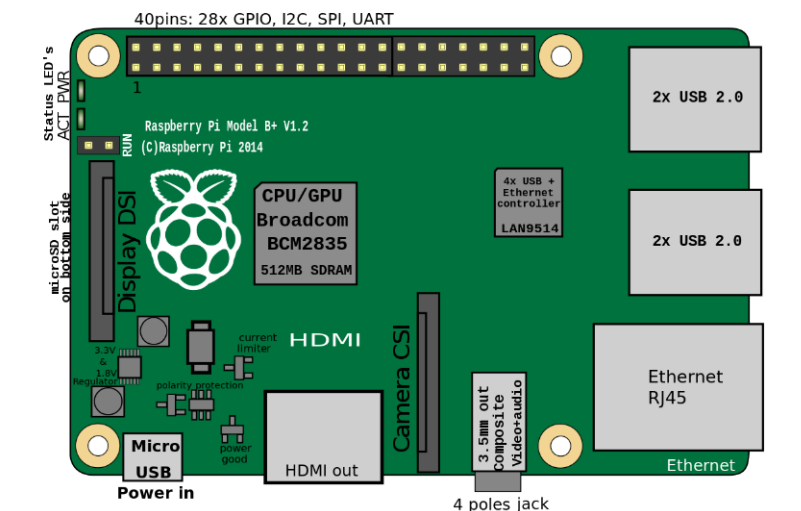


Εικόνα 8 – Raspberry Pi 1 Model A +

- **Raspberry Pi 2**

Τον Φεβρουάριο του 2015 κυκλοφόρησε το Raspberry Pi 2 Model B με τον μικροελεγκτή BCM2836, με τετραπύρηνο επεξεργαστή Cortex-A7 (ARMv7-A 32-bit) στα 900MHz, μνήμη 1 GB, 40 ακίδες, 4 θύρες USB 2.0, 1 θύρα Ethernet και 1 θύρα HDMI.

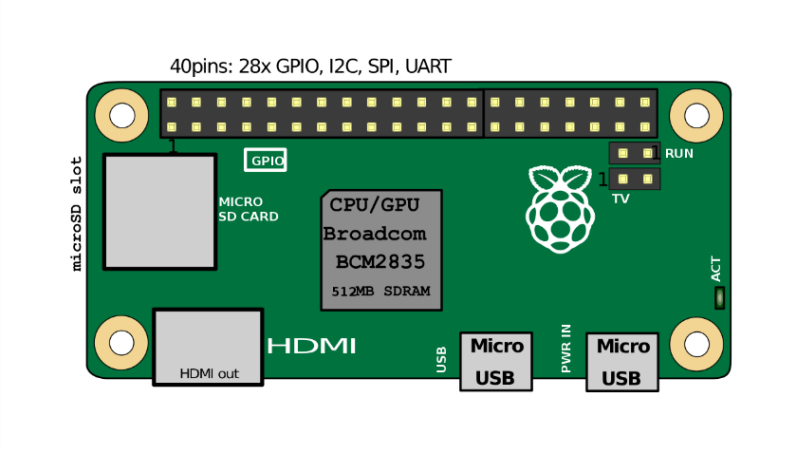
Τον Οκτώβριο του 2016 κυκλοφόρησε το Raspberry Pi 2 Model B V1.2 με τον μικροελεγκτή BCM2837, με τετραπύρηνο επεξεργαστή Cortex-A53 (ARMv8-A 64/32-bit) στα 900MHz, μνήμη 1 GB, 40 ακίδες, 4 θύρες USB 2.0, 1 θύρα Ethernet και 1 θύρα HDMI.



Εικόνα 9 – Raspberry Pi 2 Model B V1.2

- **Raspberry Pi Zero**

Τον Νοέμβριο του 2015 κυκλοφόρησε το Raspberry Pi Zero. Το Raspberry Pi Zero είναι πιο οικονομικό από όλα τα προηγούμενα μοντέλα και πιο μικρό σε μέγεθος. Το Raspberry Pi Zero περιέχει τον μικροελεγκτή BCM2835, μονοπύρηνου επεξεργαστή ARM1176JZF-S (ARMv6Z 32-bit) στα 1GHz, μνήμη 512 Mb, 40 ακίδες, 1 θύρες Micro-USB και 1 θύρα Mini-HDMI. Η συγκεκριμένη έκδοση δεν είχε θύρα Ethernet.

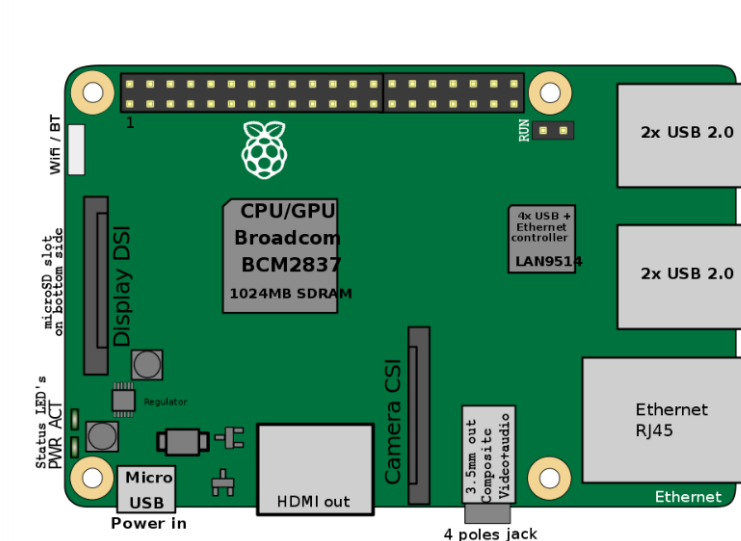


Εικόνα 10 – Raspberry Pi Zero

Τον Φεβρουάριο του 2017 κυκλοφόρησε το Raspberry Pi Zero W με τον μικροελεγκτή BCM2835, με μονοπύρηνου επεξεργαστή ARM1176JZF-S (ARMv6Z 32-bit) στα 1GHz, μνήμη 512 Mb, 40 ακίδες, 1 θύρα Micro-USB, Wi-Fi 802.11 wireless b/g/n single band 2.4GHz, Bluetooth 4.1 BLE και 1 θύρα Mini-HDMI.

- **Raspberry Pi 3**

Τον Φεβρουάριο του 2016 κυκλοφόρησε το Raspberry Pi 3 Model B με τον μικροελεγκτή BCM2837, με τετραπύρηνου επεξεργαστή Cortex-A53 (ARMv8-A 64/32-bit) στα 1.2GHz, μνήμη 1 GB, 40 ακίδες, 4 θύρες USB 2.0, 1 θύρα Ethernet, Wi-Fi 802.11 wireless b/g/n single band 2.4GHz, Bluetooth 4.1 BLE και 1 θύρα HDMI.



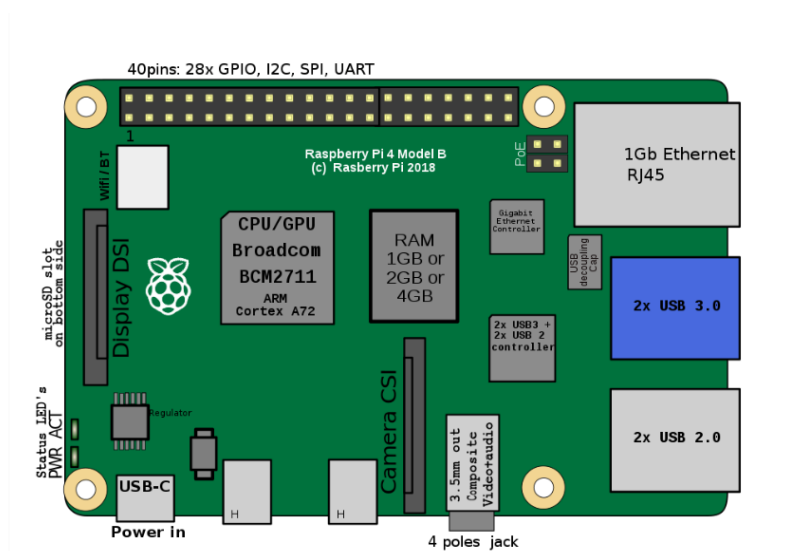
Εικόνα 11 – Raspberry Pi 3 Model B

Τον Μάρτιο του 2018 κυκλοφόρησε το Raspberry Pi 3 Model B + με τον μικροελεγκτή BCM2837B0, με τετραπύρηνου επεξεργαστή Cortex-A53 (ARMv8-A 64/32-bit) στα 1.4GHz, μνήμη 1 GB, 40 ακίδες, 4 θύρες USB 2.0, 1 θύρα Ethernet, Wi-Fi 802.11 wireless b/g/n/ac dual band 2.4/5GHz, Bluetooth 4.2 LS BLE και 1 θύρα HDMI.

Τον Νοέμβριο του 2018 κυκλοφόρησε το Raspberry Pi 3 Model A + με τον μικροελεγκτή BCM2837B0, με τετραπύρηνο επεξεργαστή Cortex-A53 (ARMv8 64-bit) στα 1.4GHz, μνήμη 512 Mb, 40 ακίδες, 1 θύρα USB 2.0, Wi-Fi 802.11 wireless b/g/n/ac dual band 2.4/5GHz, Bluetooth 4.2 BLE και 1 θύρα HDMI.

- **Raspberry Pi 4**

Τον Ιούνιο του 2019 κυκλοφόρησε το Raspberry Pi 4 Model B με 3 εκδόσεις που διαφέρουν στη μνήμη (1, 2, 4 GB), με τον μικροελεγκτή BCM2711, με τετραπύρηνο επεξεργαστή Cortex-A72 (ARMv8-A 64/32-bit) στα 1.5GHz, 40 ακίδες, 2 θύρες USB 2.0, 2 θύρες USB 3.0, 1 θύρα Ethernet, Wi-Fi 802.11 wireless b/g/n/ac dual band 2.4/5GHz, Bluetooth 5.0 και 2 θύρες HDMI.

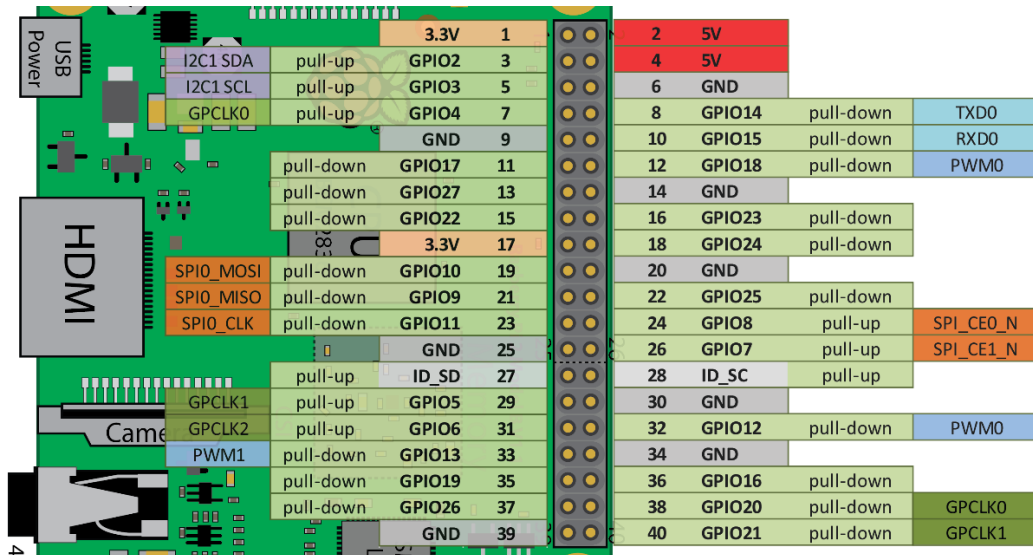


Εικόνα 12 – Raspberry Pi 4 Model B

Τον Μάιο 2020 κυκλοφόρησε νέα έκδοση του Raspberry Pi 4 Model B, με μνήμη 8GB και με τα ίδια χαρακτηριστικά, με τις προηγούμενες εκδόσεις.

## 2.1.4. Ακίδες των Raspberry Pi

Τα περισσότερα μοντέλα έχουν 40 ακίδες, τα πρώτα μοντέλα έχουν λιγότερες. Στο Raspberry Pi 3 Model B οι ακίδες είναι όπως φαίνονται παρακάτω.



Εικόνα 13 – Ακίδες Raspberry Pi 3 Model B

Όπως βλέπουμε στην (Εικόνα 13) υπάρχουν 40 ακίδες όπου η αρίθμηση ξεκινά από την πάνω αριστερά ακίδα. Η πάνω δεξιά αποτελεί την 2<sup>η</sup> ακίδα. Κάθετα από την 1<sup>η</sup> ακίδα υπάρχουν οι μονοί αριθμοί των ακίδων, ενώ κάθετα από την 2<sup>η</sup> ακίδα υπάρχουν αντίστοιχα οι ζυγοί αριθμοί. Όπως βλέπουμε και στην εικόνα η κάθε ακίδα έχει το δικό της χαρακτηριστικό. Συνολικά υπάρχουν 9 διαφορετικά είδη. Αναλυτικά βλέπουμε:

- 2 ακίδες για τάση 5V.
- 2 ακίδες για τάση 3.3V.
- 8 ακίδες για γείωση.
- 11 ακίδες για γενική χρήση.
- 2 ακίδες I2C.
- 5 ακίδες CLK.
- 5 ακίδες SPI.
- 3 ακίδες PWM.
- 2 ακίδες UART.



Τα χαρακτηριστικά των ακίδων είναι τα εξής:

- Οι ακίδες I2C χρησιμοποιούνται για την επικοινωνία μεταξύ πολλών απλών συσκευών και αισθητήρων μέσω μόνο δύο καλωδίων, με χαμηλές ταχύτητες.
- Οι ακίδες CLK χρησιμοποιούνται για χρονικά σήματα για να παρέχουν παλμούς με μέγιστη συχνότητα περίπου 75MHz.
- Οι ακίδες SPI χρησιμοποιούνται για να διαβάζουν περίπλοκους αισθητήρες, απλές οθόνες, ή επικοινωνία μεταξύ συσκευών. Η κάθε μετάδοση δεδομένων συγχρονίζεται από ένα χρονικό παλμό.
- Οι ακίδες PWM χρησιμοποιούνται για να ρυθμίζουν την ταχύτητα.
- Οι ακίδες UART χρησιμοποιούνται για πρόσβαση σε σειριακή κονσόλα, μετάδοση δεδομένων κατά τη σειριακή σύνδεση, μετατροπή των byte των δεδομένων σε bits.

### 2.1.5. Λογισμικό

Το Raspberry Pi Foundation παρέχει το Raspbian, μια έκδοση Linux που βασίζεται σε Debian (32-bit), καθώς και Ubuntu τρίτων κατασκευαστών. Υπάρχουν πολλά διαφορετικά λειτουργικά συστήματα για το Raspberry Pi όπως το Arch Linux ARM, το Pidora, το Raspbm, το Openelec και το Risc Os. Από το μοντέλο Raspberry Pi 3 και μετά μπορεί να έχει και Windows 10 σαν λειτουργικό σύστημα.

Το Raspbian είναι ένα ελεύθερο λειτουργικό σύστημα βασισμένο στο Debian και βελτιστοποιημένο για το Raspberry Pi. Αποτελεί το λειτουργικό όταν το Raspberry Pi πρόκειται να χρησιμοποιηθεί για γενική χρήση.



## Raspbian

Εικόνα 14 – Raspbian

Το Raspbian είναι μια διανομή του Linux βασισμένο και αυτό στο Debian που φέρνει το XBMC στο Raspberry Pi. Το Raspberry Pi μπορεί να χειριστεί την αναπαραγωγή πολυμέσων, κάνοντας το ιδανικό για την δημιουργία ενός Home Theater PC.



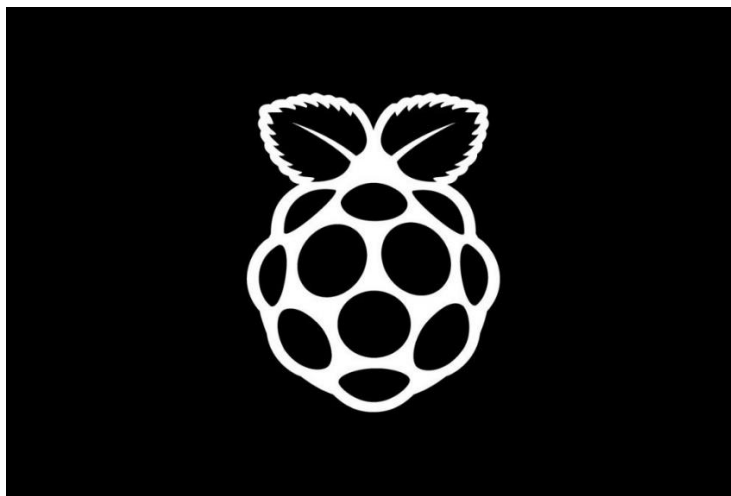
Εικόνα 15 – Raspbmc

Το Pidora είναι μια διανομή του Linux για το Raspberry Pi. Περιέχει πακέτα λογισμικού από το λειτουργικό Fedora ARM που έχει τροποποιηθεί ιδικά για το Raspberry Pi.



Εικόνα 16 – Pidora

Η προτεινόμενη εγκατάσταση από την Raspberry Pi Foundation είναι το Noobs. Γίνετε με αντιγραφή σε μια κάρτα micro SD και περιέχει τα παραπάνω λειτουργικά συστήματα. Στο άνοιγμα του Raspberry Pi επιλέγει ο χρήστης πιο λειτουργικό θα εγκαταστήσει.



Εικόνα 17 – Noobs

### **2.1.6. Πλεονεκτήματα και Μειονεκτήματα του Raspberry Pi**

Τα βασικά πλεονεκτήματα του Raspberry Pi είναι τα εξής:

- Μέγεθος αντίστοιχο με πιστωτική κάρτα.
- Πολύ μικρό κόστος.
- Χαμηλή κατανάλωση ενέργειας.
- Ευκολία στη χρήση.
- Εύκολη πρόσβαση στο Internet.
- Πρόσβαση στα Linux.
- Μεγάλη ποικιλία σε γλώσσες προγραμματισμού.
- Επέκταση της λειτουργίας μέσω των ακίδων

Τα βασικά μειονέκτημα του Raspberry Pi είναι τα εξής:

- Δεν μπορεί να αυξηθεί η μνήμη RAM.
- Μικρή τάση στους ακροδέκτες.
- Ο σχεδιασμός του υλικού είναι κλειστού τύπου

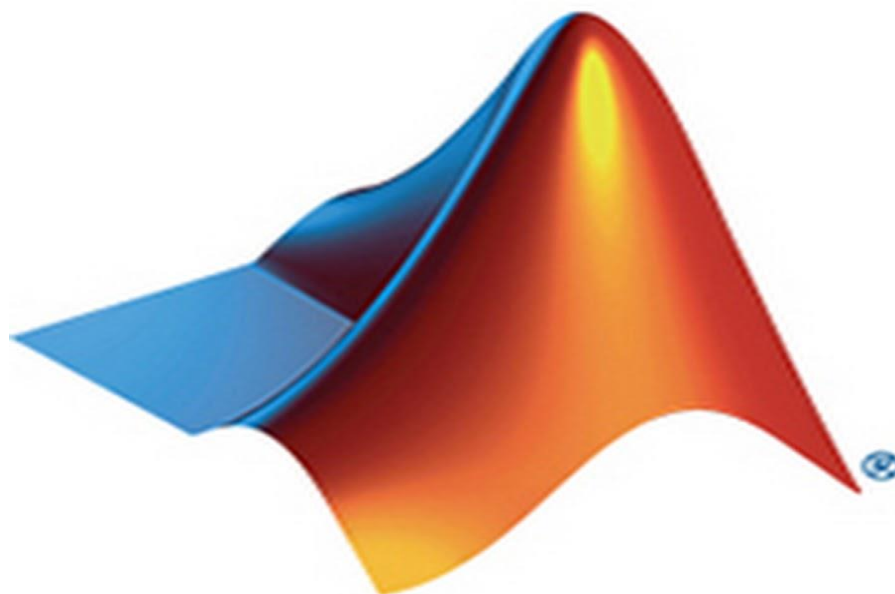
### **3. ΚΕΦΑΛΑΙΟ 3**

#### **3.1. MATLAB**

##### **3.1.1. Εισαγωγή στο Matlab**

Το MATLAB αποτελεί ένα εμπορικό εργαλείο το οποίο προσφέρει ένα διαδραστικό προγραμματιστικό περιβάλλον στον χρήστη και χρησιμοποιείται σε ένα μεγάλο εύρος εφαρμογών. Δημιουργήθηκε από τον Moler στη δεκαετία του 1970, αρχικά σαν εργαλείο διαχείρισης των βιβλιοθηκών της γλώσσας προγραμματισμού FORTRAN: Linpack (γραμμική άλγεβρα) και Eispack (ιδιο-τιμές και ιδιο-διανύσματα). Η έκδοση που χρησιμοποιείται σήμερα είναι η 9.8 και πέρα από τις δυνατότητες που παρείχαν οι προηγούμενες αυτή έχει εμπλουτιστεί με νέες βιβλιοθήκες αλλά και με καινούριο πιο φιλικό γραφικό περιβάλλον. Το Matlab χρησιμοποιείται κυρίως για μαθηματικούς υπολογισμούς, ανάπτυξη αλγορίθμων, προσομοίωση, μοντελοποίηση και προτυποποίηση. Επίσης, για ανάλυση δεδομένων, επιστημονικά και μηχανολογικά γραφικά και ανάπτυξη εφαρμογών. Είναι ένα διαδραστικό σύγχρονο εργαλείο του οποίου βασικό στοιχείο είναι η διάταξη (array). Αυτό μας επιτρέπει τη λύση πολλών υπολογιστικών προβλημάτων, ιδιαίτερα όσων περιλαμβάνουν σχηματισμούς πινάκων και διανυσμάτων σε χρόνο μικρότερο απ' ό τι απαιτούν οι γλώσσες προγραμματισμού Fortran και C.

Το Matlab περιέχει τη βιβλιοθήκη Simulink Browser. Το Matlab Simulink, είναι γλώσσα υψηλού επιπέδου για αριθμητικό υπολογισμό, οπτικοποίηση και ανάπτυξη εφαρμογών. Επίσης, το διαδραστικό περιβάλλον για την εξερεύνηση, τον σχεδιασμό και την επίλυση προβλημάτων. Το Matlab διαθέτει ενσωματωμένα γραφικά για την απεικόνιση δεδομένων όπως και εργαλεία ανάπτυξης για τη βελτίωση της ποιότητας και της βέλτιστη της εκτέλεση του.

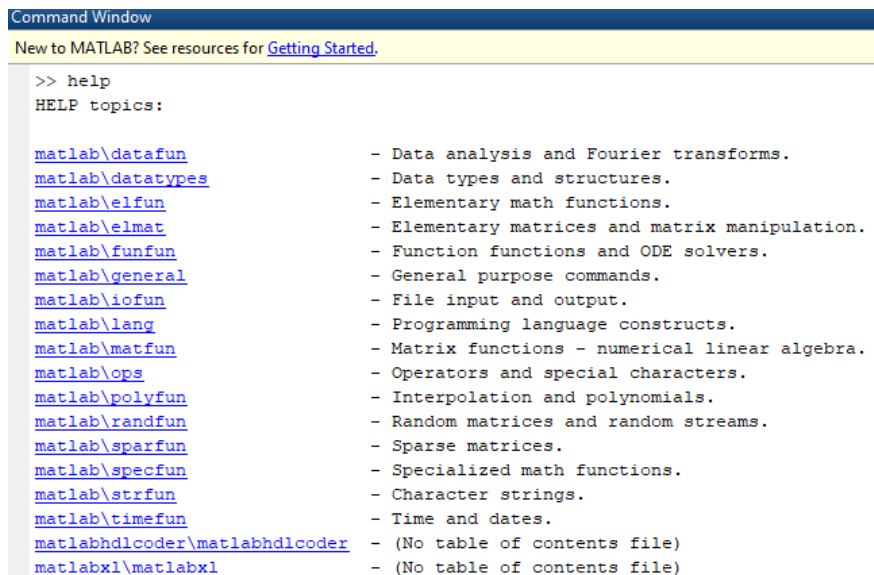


Εικόνα 18 – Matlab

Ενσωματώνει μια υψηλού επιπέδου γλώσσα προγραμματισμού, κατάλληλη για τη μοντελοποίηση και επίλυση σύνθετων μαθηματικών προβλημάτων. Βασίζεται εξ ολοκλήρου στη χρήση πινάκων, των οποίων μπορεί να είναι πραγματικοί ή μιγαδικοί αριθμοί (ακόμα και ένας μεμονωμένος αριθμός, όπως για παράδειγμα το 5, θεωρείται ως πίνακας με ένα στοιχείο). Ανάμεσα σε ένα πλήθος άλλων ευκολιών που προσφέρει επιτρέπει τον εύκολο χειρισμό πινάκων, τη γραφική απεικόνιση συναρτήσεων και δεδομένων, την υλοποίηση αλγορίθμων. Λόγω ότι το MATLAB βρίσκει εφαρμογή σε ποικίλα επιστημονικά πεδία (επεξεργασία σήματος, νευρωνικά δίκτυα, συστήματα ελέγχου κ.τ.λ.), πρόσθετα πακέτα, που ονομάζονται toolboxes, ενσωματώνονται σε αυτό και προσφέρουν χρήσιμες εξειδικευμένες συναρτήσεις.

Το Matlab διαχειρίζεται τα πάντα σαν να είναι πίνακες (πραγματικούς ή μιγαδικούς). Η γλώσσα του Matlab είναι άτυπη, δηλαδή δεν χρειάζεται να δηλώσεις τις μεταβλητές. Υπάρχουν κάποιες εντολές για βοήθεια στο Matlab όπως:

- help: εμφανίζει για λίστα βοήθειας.

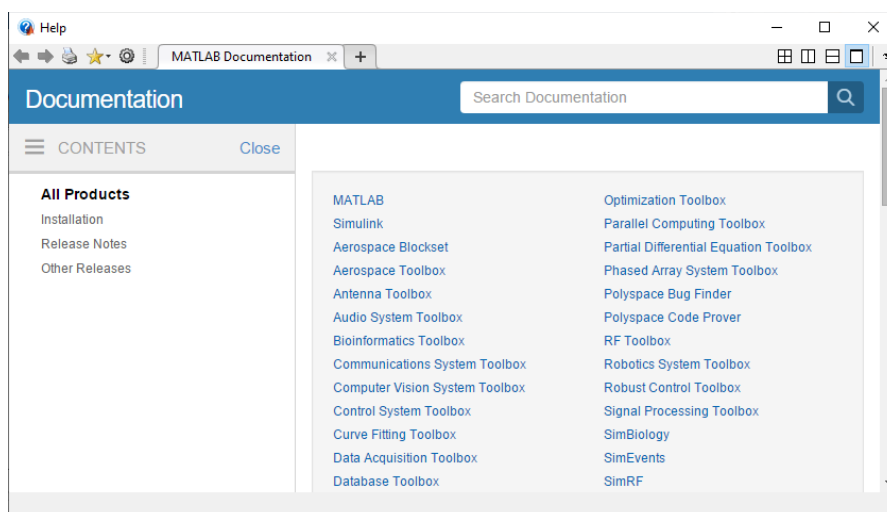


```
>> help
HELP topics:

matlab\datafun           - Data analysis and Fourier transforms.
matlab\datatypes        - Data types and structures.
matlab\elfun            - Elementary math functions.
matlab\elmat            - Elementary matrices and matrix manipulation.
matlab\funfun           - Function functions and ODE solvers.
matlab\general          - General purpose commands.
matlab\iofun            - File input and output.
matlab\lang             - Programming language constructs.
matlab\matfun           - Matrix functions - numerical linear algebra.
matlab\ops              - Operators and special characters.
matlab\polyfun          - Interpolation and polynomials.
matlab\randfun          - Random matrices and random streams.
matlab\sparfun          - Sparse matrices.
matlab\specfun          - Specialized math functions.
matlab\strfun           - Character strings.
matlab\timefun          - Time and dates.
matlabhdlcoder\matlabhdlcoder - (No table of contents file)
matlabxl\matlabxl      - (No table of contents file)
```

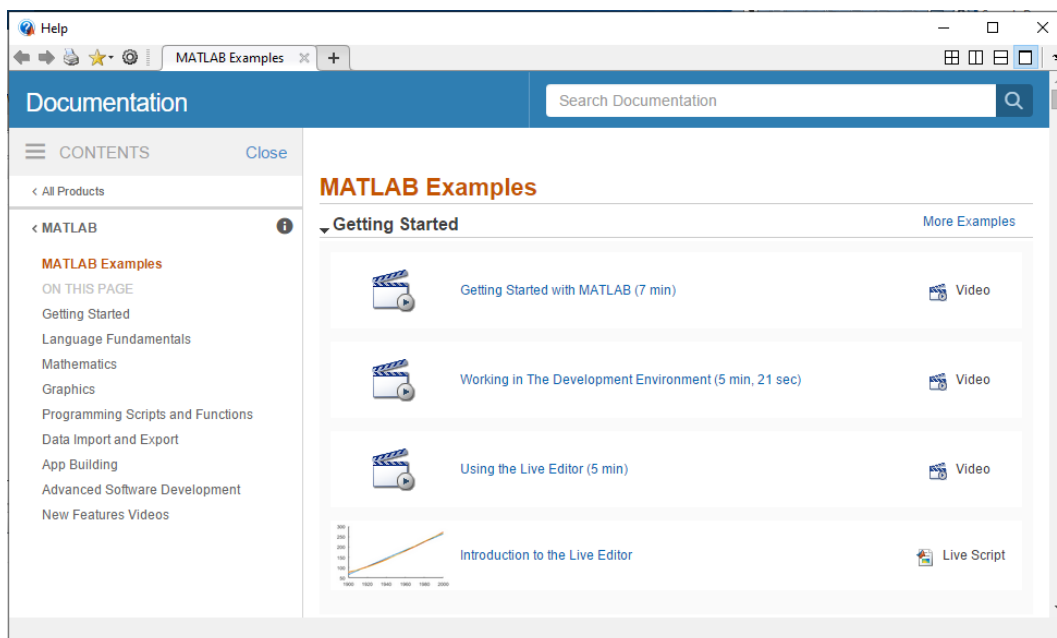
Εικόνα 19 – Matlab Εντολή help.

- help + θέμα: εμφανίζει τη βοήθεια του συγκεκριμένου θέματος ή της συνάρτησης.
- helpdesk: ανοίγει στο Web Browser αναλυτική τεκμηρίωση για το Matlab και τα Toolboxes του.



Εικόνα 20 – Matlab Εντολή helpdesk.

- lookfor + λέξη: Ψάχνει σε όλες τις συναρτήσεις για την λέξη που εισήγαμε και εμφανίζει τις συναρτήσεις που την περιέχουν.
- demo: ανοίγει μια οθόνη με διάφορα παραδείγματα και τις δυνατότητες του Matlab.



Εικόνα 21 – Matlab Εντολή demo.

### 3.1.2. Στοιχεία του Matlab

Τα βασικά χαρακτηριστικά του MATLAB είναι:

- Το Περιβάλλον Ανάπτυξης και τα αντίστοιχα εργαλεία του, περιλαμβάνει διάφορα παράθυρα, όπως το Παράθυρο Εντολών (Command Window) και το Ιστορικό Εντολών (Command History) και άλλα εργαλεία για αποσφαλμάτωση (Debugging), ανάλυση κώδικα και πλοήγηση στο σύστημα αρχείων.
- Η βιβλιοθήκη μαθηματικών συναρτήσεων, το πιο σημαντικό συστατικό του MATLAB, με στοιχειώδεις αριθμητικές συναρτήσεις, αλλά και πιο πολύπλοκες.

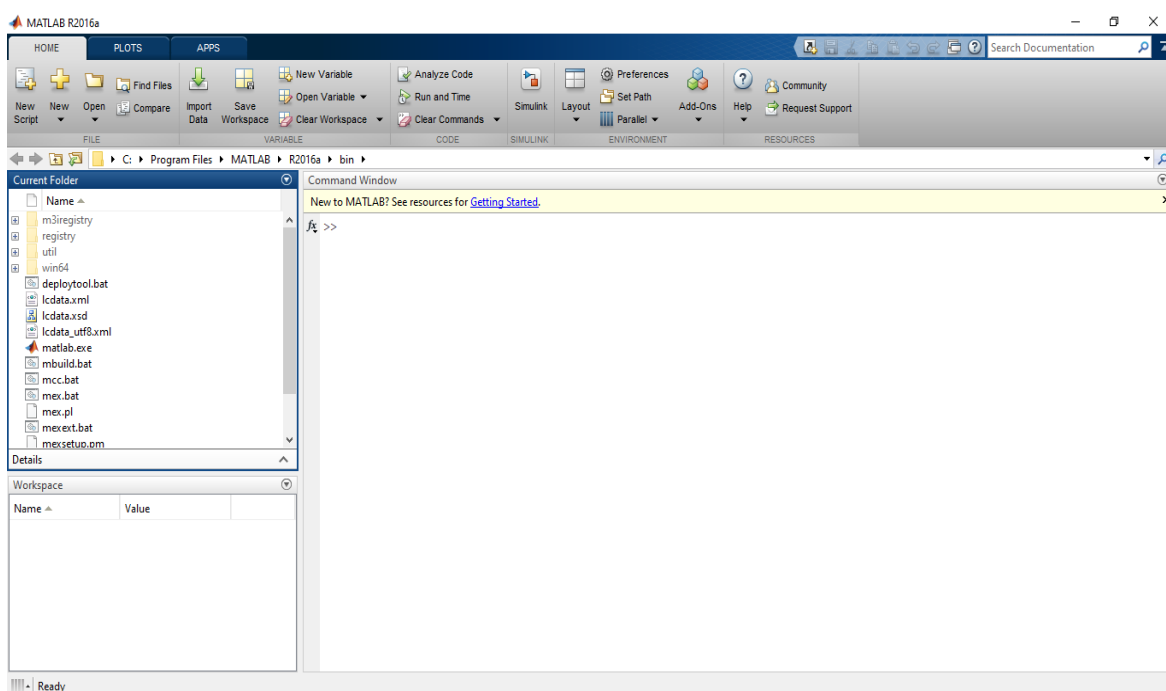


- Η γλώσσα προγραμματισμού είναι μια υψηλού επιπέδου προγραμματιστική γλώσσα με δομές δεδομένων, συναρτήσεις, εντολές ελέγχου ροής, εντολές εισόδου/εξόδου και στοιχεία από αντικειμενοστραφείς γλώσσες προγραμματισμού.
- Τα γραφικά του MATLAB παρέχουν μια πληθώρα δυνατοτήτων απεικόνισης διανυσμάτων, πινάκων και γραφημάτων στις 2 και 3 διαστάσεις.

Το σύστημα του Matlab αποτελείται από πέντε βασικά μέρη:

- **Περιβάλλον ανάπτυξης**

Είναι ένα σύνολο από εργαλεία που βοηθάνε στην χρήση των συναρτήσεων και των αρχείων του Matlab. Τα εργαλεία αυτά συμπεριλαμβάνουν την επιφάνεια εργασίας του Matlab (Matlab Desktop), τη γραμμή εργασιών (Command Window), το ιστορικό εντολών (Command History), έναν κειμενογράφο (Editor), έναν διορθωτή (Debugger), έναν περιηγητή για τη βοήθεια, το χώρο εργασίας (Workspace), τα αρχεία και την αναζήτηση.



Εικόνα 22 – Περιβάλλον Matlab

- **Βιβλιοθήκη μαθηματικών συναρτήσεων του Matlab**

Έχει μια τεράστια συλλογή από υπολογιστικούς αλγορίθμους που καλύπτουν ένα ευρύτατο φάσμα που ξεκίνα από στοιχειώδεις συναρτήσεις, όπως το άθροισμα, το ημίτονο, και εκτείνεται σε πιο σύνθετες συναρτήσεις, όπως αντιστροφή πίνακα, συναρτήσεις Bessel και μετασχηματισμούς Fourier.

- **Γλώσσα του Matlab**

Είναι μια υψηλού επιπέδου γλώσσα με δηλώσεις δομής ελέγχου, δομές δεδομένων, συναρτήσεις/εξόδους και χαρακτηριστικά αντικειμενοστραφούς προγραμματισμού.

- **Γραφικά**

Το Matlab μπορεί με ευκολία να αναπαραστήσει γραφικά διανύσματα και πίνακες και δίνει τη δυνατότητα σχολιασμού και εκτύπωσης αυτών των γραφημάτων. Περιλαμβάνει επιπέδου για απεικόνιση διανυσμάτων σε διαγράμματα δυο και τριών διαστάσεων, επεξεργασία εικόνας, δυναμική κίνηση ομοιωμάτων και γραφικά παρουσίασης. Περιλαμβάνει επιπλέον συναρτήσεις χαμηλού επιπέδου που επιτρέπουν στο χρήστη να προσαρμόσει πλήρως την εμφάνιση των γραφημάτων.

- **Περιβάλλον εφαρμογής προγραμμάτων (Application Program Interface - API)**

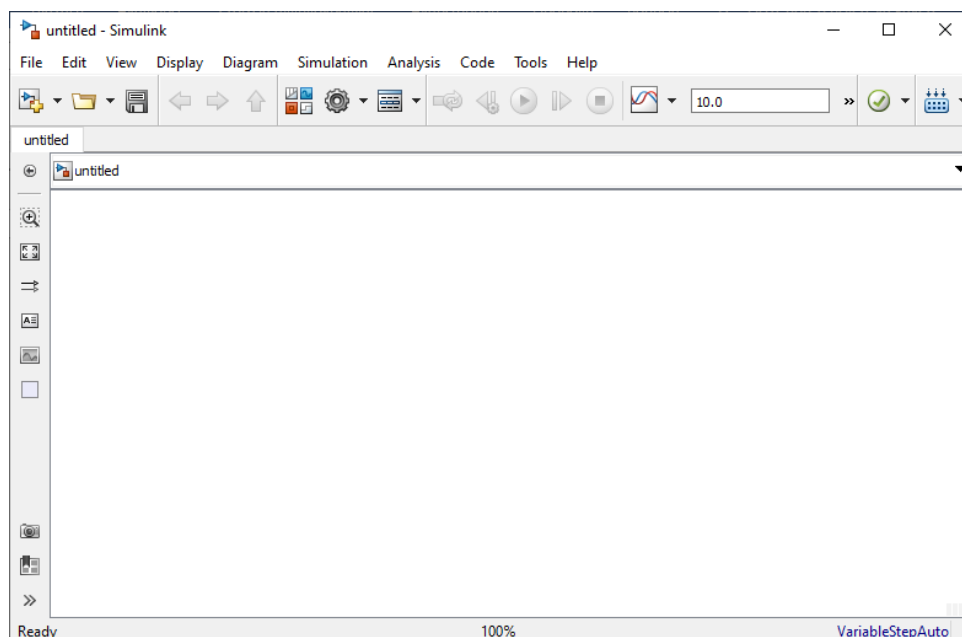
Είναι μια βιβλιοθήκη που επιτρέπει στο χρήστη να γράψει κώδικα στις γλώσσες προγραμματισμού C και Fortan που αλληλοεπιδρούν με το Matlab. Δίνει τη δυνατότητα κλήσης υπορουτίνων από το Matlab (Dynamic Linking), κλήση του

Matlab ως υπολογιστικής μηχανής και για την ανάγνωση και ανάπτυξη αρχείων τύπου MAT (MAX Files).

### 3.1.3. Χρήση Του Matlab

Το Matlab μπορεί να χρησιμοποιηθεί σε ένα μεγάλο εύρος εφαρμογών που απαιτούν σύνθετους μαθηματικούς υπολογισμούς. Είναι προγραμματισμένο για να λύνει αριθμητικά προβλήματα, δηλαδή με αριθμητική πεπερασμένη ακρίβεια. Χρησιμοποιείται για την ανάπτυξη αλγορίθμων που χρησιμοποιούνται για την επίλυση μαθηματικών προβλημάτων. Για την ανάπτυξη ενός αλγορίθμου απαιτείται ο ορισμός και η πλήρης κατανόηση του προβλήματος προς επίλυση, δηλαδή ο καθορισμός μιας συγκεκριμένης διαδικασίας, βημάτων, τα οποία οδηγούν σε συγκεκριμένα αποτελέσματα με ικανοποιητικά ελάχιστο δυνατό κόστος (μνήμης και χρόνου). Επίσης απαιτείται ο έλεγχος του προγράμματος για διάφορες περιπτώσεις δεδομένων και η χρήση υπορουτίνας / προγράμματος για επίλυση σχετικών προβλημάτων.

Μια άλλη εφαρμογή του Matlab είναι η μοντελοποίηση και η προσομοίωση προβλημάτων, κάνοντας χρήση του Simulink, το οποίο είναι ένα πρόγραμμα ενσωματωμένο στο Matlab. Η ανάλυση, η οπτικοποίηση δεδομένων και η δημιουργία γραφικών παραστάσεων. Το Matlab διαθέτει δυναμικές και ευέλικτες γραφικές ικανότητες. Σχήματα διαφόρων τύπων μπορούν να δημιουργηθούν σχετικά εύκολα. Τα Toolboxes (εργαλειοθήκες) του Matlab είναι ένα ευρύ σύνολο από ήδη ανεπτυγμένους αλγορίθμους που παρέχονται στο χρήστη με τη μορφή συναρτήσεων του προγράμματος και βοηθούν στη λύση ειδικής κατηγορίας προβλημάτων. Κάποια έτοιμα πακέτα τέτοιων αλγορίθμων είναι το control systems toolbox, το signal processing toolbox, το Simulink, το model predictive control toolbox και άλλα πολλά.



Εικόνα 23 – Περιβάλλον Simulink

Τα βασικά χαρακτηριστικά σου Simulink είναι τα εξής:

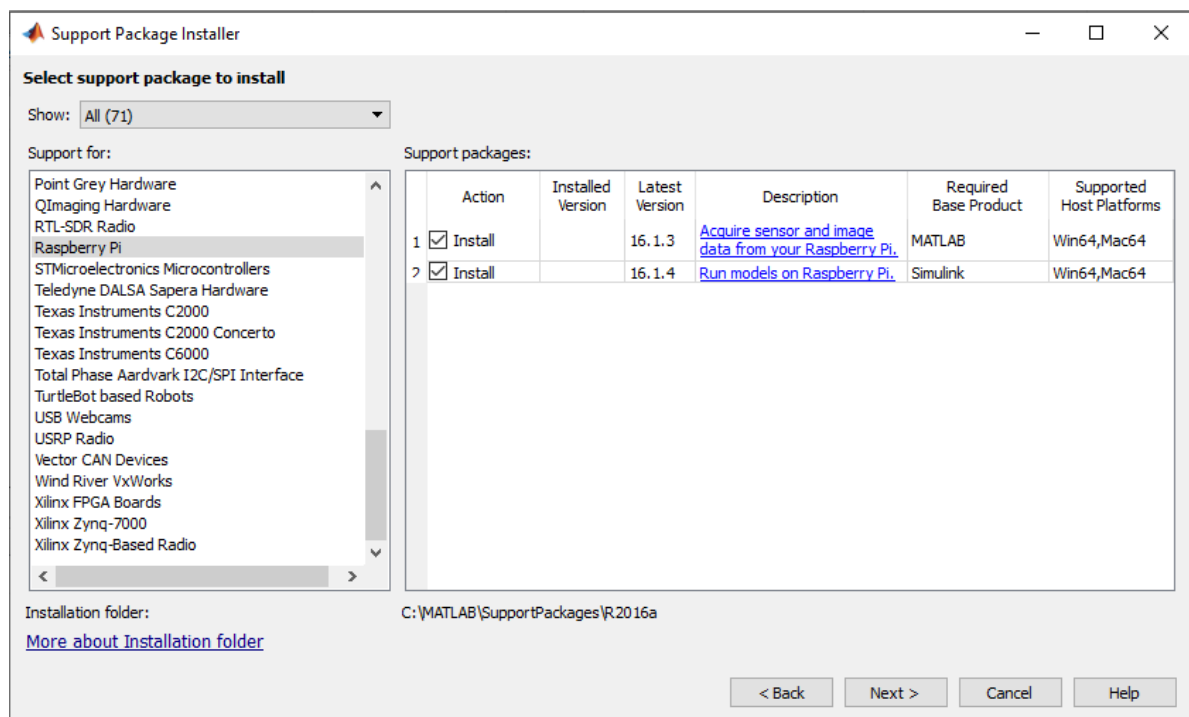
- Χρησιμοποιείται για τη δημιουργία μοντέλων προσομοιώσεων και ανάλυση φυσικών συστημάτων με ένα ευέλικτο και εύκολο στη χρήση περιβάλλον εργασίας.
- Δεν χρειάζεται προγραμματιστικές ικανότητες υψηλού επιπέδου.
- Επιλύει διάφορα είδη προβλημάτων που εμπίπτουν σε διάφορους επιστημονικούς τομείς λόγω της μαθηματικής επάρκειάς του.
- Ο χρήστης μπορεί να σχεδιάζει διαγράμματα ροής με το ποντίκι.
- Παρέχει μεγάλες δυνατότητες γραφικής απεικόνισης που απεικονίζει τα αποτελέσματα των προσομοιώσεων στην οθόνη του υπολογιστή.
- Υπάρχει πλήθος βιβλιογραφικών αναφορών και εφαρμογών αλλά και τεχνικής τεκμηρίωσης, που το καθιστά προσβάσιμο σε κάθε ενδιαφερόμενο.
- Υπάρχουν έτοιμες εφαρμογές Demo στο Matlab αλλά και στην ιστοσελίδα του MathWorks ([www.mathworks.com/](http://www.mathworks.com/)).

### 3.1.4. Support Package Matlab

Τα πακέτα υποστήριξης (Support Packages) είναι πρόσθετα πακέτα που χρησιμοποιούν ένα προϊόν MathWorks με συγκεκριμένο hardware και software τρίτων. Μερικά από τα Support Package είναι τα ακόλουθα:

- Arduino Hardware
- LEGO MINDSTORMS EV3 Hardware
- Raspberry Pi Hardware
- USB Camera

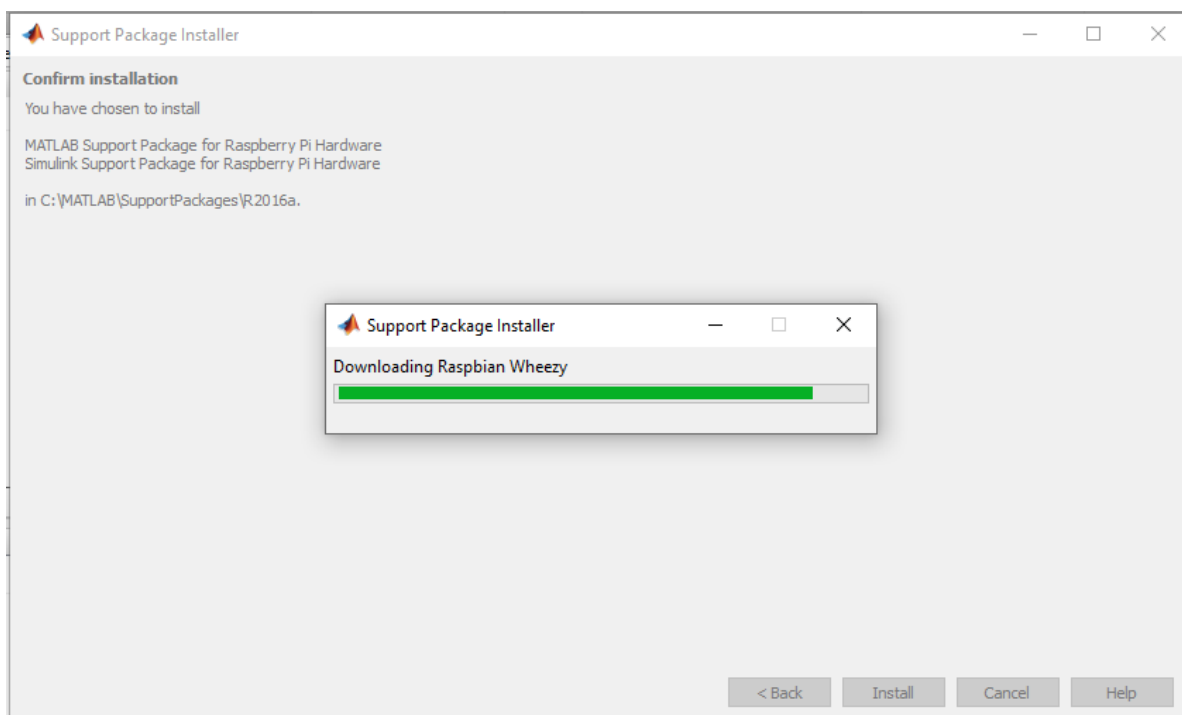
Για την εγκατάσταση ενός Support Package στο Matlab, επιλέγουμε από το μενού το «Add-Ons» και επιλέγουμε το «Get Hardware Support Package». Επιλέγουμε τη διαδικασία που επιθυμούμε (Install from Internet, Download from Internet, Install from folder, Uninstall). Έπειτα επιλέγουμε πιο Support Package που θέλουμε να εγκαταστήσουμε.



Εικόνα 24 – Εγκατάσταση Support Package

Ανάπτυξη PID ελεγκτή θερμοκρασίας με τη βοήθεια Raspberry Pi και Matlab

Πατάμε «Next» και περιμένουμε να εγκατασταθεί το Support Package.



Εικόνα 25 – Εγκατάσταση Support Package Raspberry

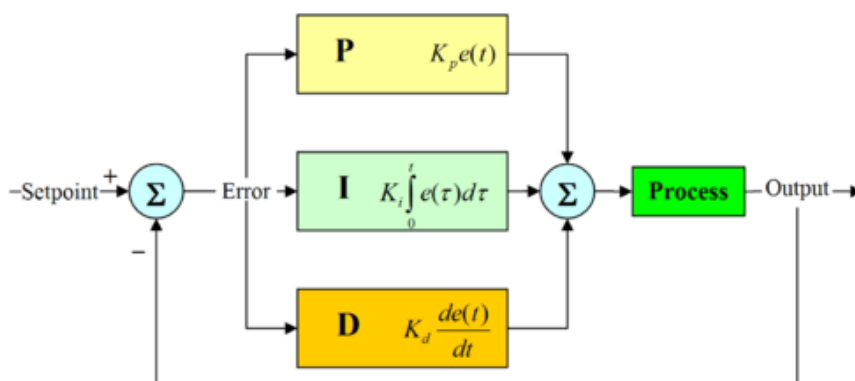
## 4. ΚΕΦΑΛΑΙΟ 4

### 4.1. PID Controller

#### 4.1.1. Εισαγωγή στους PID Controller

Το διακριτικό χαρακτηριστικό του ελεγκτή PID είναι η δυνατότητα χρήσης των τριών όρων ελέγχου της αναλογικής, ολοκληρωμένης και της παράγωγης επίδρασης στην έξοδο του ελεγκτή για την εφαρμογή ακριβούς και βέλτιστου ελέγχου. Ο ελεγκτής PID υπολογίζει συνεχώς μια τιμή σφάλματος  $e(t)$  ως τη διαφορά μεταξύ ενός επιθυμητού σημείου ρύθμισης  $SP = r(t)$  και μιας μετρούμενη μεταβλητή  $PV = y(t)$ . Ο ελεγκτής προσπαθεί να ελαχιστοποιήσει το σφάλμα με την πάροδο του χρόνου προσαρμόζοντας μια μεταβλητή ελέγχου  $u(t)$ , όπως το άνοιγμα μιας βαλβίδας ελέγχου, σε μια νέα τιμή που καθορίζεται από ένα σταθμισμένο άθροισμα των όρων ελέγχου.

Ο ελεγκτής PID αποτελεί αναπόσπαστο κομμάτι στα συστήματα αυτομάτου ελέγχου. Υπολογίζει συνεχώς μια τιμή σφάλματος  $e(t)$  ως τη διαφορά μεταξύ της επιθυμητής και της πραγματικής τιμής και εφαρμόζει διόρθωση στο σύστημα βασισμένη στους όρους P, I, D. Η έξοδος του ελεγκτή PID σχηματίζεται από το άθροισμα τριών όρων, ενός όρου P (Proportional) ανάλογου του σφάλματος, ενός όρου I (Integral) ανάλογου του ολοκληρώματος του σφάλματος και ενός όρου D (Derivative) ανάλογου της παραγώγου του σφάλματος όπως φαίνεται στο παρακάτω διάγραμμα.



Εικόνα 26 – Τύποι PID Controller

Ο όρος P (Proportional) προσαρμόζει την έξοδο του ελεγκτή σύμφωνα με το σύνολο του μεγέθους του σφάλματος. Το ρεύμα του κινητήρα ρυθμίζεται ανάλογα με το υπάρχον σφάλμα. Ωστόσο, αυτή η μέθοδος αποτυγχάνει εάν, για παράδειγμα, ο βραχίονας πρέπει να ανυψώσει διαφορετικά βάρη, ένα μεγαλύτερο βάρος χρειάζεται μεγαλύτερη δύναμη που εφαρμόζεται για το ίδιο, αλλά και μια μικρότερη δύναμη εάν το σφάλμα είναι ανάποδα.

Ο όρος I (Integral) χρησιμοποιείται για την αφαίρεση του σφάλματος σταθερής κατάστασης του συστήματος ελέγχου και τη βελτίωση της απόκρισης σταθερής κατάστασης. Αυξάνει την ενέργεια, όχι μόνο με το σφάλμα αλλά και με τον χρόνο για τον οποίο έχει επιμείνει. Επομένως, εάν η εφαρμοζόμενη δύναμη δεν είναι αρκετή για να μηδενίσει το σφάλμα, αυτή η δύναμη θα αυξηθεί όσο περνά ο χρόνος. Ένας ελεγκτής I θα μπορούσε να φέρει το σφάλμα στο μηδέν, αλλά η αντίδραση του θα είναι αργή. Επειδή η δράση είναι μικρή, χρειάζεται χρόνος. Η δράση αυξάνεται όσο το σφάλμα είναι θετικό, ακόμη και αν το σφάλμα έχει αρχίσει να πλησιάζει το μηδέν.

Ο όρος D (Derivative) χρησιμοποιείται για να προβλέψει την τάση του σφάλματος και να βελτιώσει την παροδική απόκριση του συστήματος. Στόχος του ελεγκτή είναι να φέρει το σφάλμα σε οριζόντια γραμμή, να μειώσει την ισχύ που ασκείται και να μειώσει την υπέρβαση. Η υπερβολική ώθηση, όταν το σφάλμα είναι μικρό και η μείωση, θα οδηγήσει σε υπέρβαση. Μετά την υπέρβαση, εάν ο ελεγκτής έπρεπε να εφαρμόσει μια μεγάλη διόρθωση στην αντίθετη κατεύθυνση και να ξεπεράσει επανειλημμένα την επιθυμητή θέση, η έξοδος θα ταλαντευόταν γύρω από το σημείο ρύθμισης είτε σε μια σταθερή, αναπτυσσόμενη ή αποσυντιθέμενη ημιτονοειδής. Εάν το πλάτος των ταλαντώσεων αυξάνεται με το χρόνο, το σύστημα είναι ασταθές. Εάν μειωθούν, το σύστημα είναι σταθερό. Εάν οι ταλαντώσεις παραμένουν σε σταθερό μέγεθος, το σύστημα είναι οριακά σταθερό.

Κάθε ένας από αυτούς τους όρους μπορεί να καθοριστεί από τον χρήστη. Αυτοί οι όροι πρέπει να προσαρμοστούν για να βελτιστοποιήσουν την ακρίβεια του ελέγχου και η διαδικασία προσδιορισμού των τιμών αυτών των παραμέτρων είναι γνωστή ως PID Tuning. Αυτές οι λειτουργίες είναι αρκετές, για τις περισσότερες διαδικασίες ελέγχου. Η δομή του PID Controller προτείνεται για αυτό το έργο καθώς



είναι απλή αλλά και η πιο εκτεταμένη μέθοδος ελέγχου που έχει χρησιμοποιηθεί στη βιομηχανία μέχρι στιγμής.

Η ισορροπία αυτών επιτυγχάνεται με το συντονισμό βρόχου για την παραγωγή της βέλτιστης λειτουργίας ελέγχου. Οι σταθερές συντονισμού παρουσιάζονται ως "K" και πρέπει να προέρχονται για κάθε εφαρμογή ελέγχου, καθώς εξαρτώνται από τα χαρακτηριστικά απόκρισης του πλήρους βρόχου εξωτερικά του ελεγκτή. Αυτά εξαρτώνται από τη συμπεριφορά του αισθητήρα μέτρησης στο τελικό στοιχείο ελέγχου. Οι τιμές των σταθερών τιμών μπορούν συνήθως να εισαχθούν γνωρίζοντας τον τύπο της εφαρμογής, αλλά κανονικά είναι βελτιωμένες ή συντονισμένες, κάνοντας «bumping» τη διαδικασία και παρατηρώντας την απόκριση του συστήματος. Ο PID Controller επιχειρεί συνεχώς να προσαρμόσει ένα κατάλληλο αναλογικό κέρδος ( $K_p$ ), ολοκληρωτικό κέρδος ( $K_i$ ) και διαφορικό κέρδος ( $K_d$ ) για την επίτευξη της βέλτιστης απόδοσης ελέγχου.

Το μαθηματικό μοντέλο και ο ελεγκτής χρησιμοποιούν μια άμεση ενέργεια ελέγχου για όλους τους όρους, πράγμα που σημαίνει ότι ένα αυξανόμενο θετικό σφάλμα οδηγεί σε μια αυξανόμενη θετική έξοδο ελέγχου, για την εφαρμογή της διόρθωσης σφαλμάτων. Ωστόσο, η έξοδος ονομάζεται αντίστροφη ενεργώντας εάν είναι απαραίτητο να εφαρμοστεί αρνητική διορθωτική ενέργεια. Για παράδειγμα, εάν η βαλβίδα στο βρόχο ροής ήταν άνοιγμα βαλβίδας 100-0% για έξοδο ελέγχου 0-100%, που σημαίνει ότι η ενέργεια του ελεγκτή πρέπει να αντιστραφεί. Ορισμένα σχήματα ελέγχου διαδικασίας και στοιχεία τελικού ελέγχου απαιτούν αυτήν την αντίστροφη ενέργεια.

Ο μαθηματικός τύπος είναι:

$$u(t) = MV(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

Εικόνα 27 – Μαθηματικός Τύποι PID Controller

Όπου το  $K_p$ ,  $K_i$  και  $K_d$ , όλοι μη αρνητικοί, δηλώνουν τους συντελεστές για τους αναλογικούς, αναπόσπαστους και παράγωγους όρους αντίστοιχα. Η χρήση του τύπου δεν εγγυάται τον βέλτιστο έλεγχο του συστήματος ή τη σταθερότητα του ελέγχου. Μπορεί να προκύψουν καταστάσεις όπου υπάρχουν υπερβολικές καθυστερήσεις. Η απόκριση του ελεγκτή μπορεί να περιγραφεί ως προς την ανταπόκρισή του σε ένα σφάλμα, το βαθμό στον οποίο το σύστημα ξεπερνά ένα σημείο ρύθμισης και τον βαθμό οποιασδήποτε ταλάντωσης του συστήματος. Ωστόσο, ο ελεγκτής PID είναι γενικά εφαρμόσιμος, καθώς βασίζεται μόνο στην απόκριση της μετρούμενης μεταβλητής διαδικασίας, όχι στη γνώση ή σε μοντέλο της υποκείμενης διαδικασίας.

Με την ρύθμιση των τριών παραμέτρων του μοντέλου, ένας ελεγκτής PID μπορεί να αντιμετωπίσει συγκεκριμένες απαιτήσεις επεξεργασίας. Σε ορισμένες εφαρμογές χρειάζονται μόνο ένας ή δύο όροι για την παροχή του κατάλληλου ελέγχου του συστήματος. Αυτό θα επιτευχθεί με το μηδενισμό των άλλων παραμέτρων. Ένας ελεγκτής PID ονομάζεται ελεγκτής PI, PD, P, I ή D όταν λείπουν οι αντίστοιχοι όροι. Πιο συχνά από όλους συναντάται ο PI ελεγκτής, σε εφαρμογές με μικρές καθυστερήσεις χρόνου ή σε περιπτώσεις όπου ο θόρυβος μέτρησης είναι σημαντικός ώστε να αποτρέπεται η χρήση του διαφορικού όρου, ή όταν δεν απαιτείται το κλειστό σύστημα να είναι αρκετά γρήγορο.

Τα αποτελέσματα καθενός από τους ελεγκτές P, I, D σε ένα σύστημα κλειστού βρόγχου φαίνονται στο πίνακα παρακάτω:

Τύπος Ελεγκτή	Χρόνος Ανόδου	Υπερύψωση	Χρόνος Αποκατάστασης	Μόνιμο Σφάλμα
<b>P</b>	Μείωση	Αύξηση	Μικρή Αλλαγή	Μείωση
<b>I</b>	Μείωση	Αύξηση	Αύξηση	Εξάλειψη
<b>D</b>	Μικρή Αλλαγή	Μείωση	Μείωση	Μικρή Αλλαγή

Πίνακας 1: Επίδραση ελεγκτών P, I, D

#### 4.1.2. Σχεδίαση Ελεγκτών PID

Υπάρχουν πολλές μέθοδοι για τη σχεδίαση ενός ελεγκτή PID. Οι πιο αποτελεσματικές μέθοδοι γενικά περιλαμβάνουν την ανάπτυξη κάποιας μορφής μοντέλου διεργασίας και στη συνέχεια την επιλογή των όρων  $K_p$ ,  $K_i$  και  $K_d$  με βάση τις δυναμικές παραμέτρους του μοντέλου. Οι μέθοδοι του χειροκίνητου συντονισμού μπορεί να είναι σχετικά χρονοβόρες.

Η επιλογή της μεθόδου εξαρτάτε σε μεγάλο βαθμό από το εάν ο ελεγκτής μπορεί να γίνει εκτός σύνδεσης για συντονισμό και από τον χρόνο απόκρισης του συστήματος. Εάν το σύστημα μπορεί να ληφθεί εκτός σύνδεσης, η καλύτερη μέθοδος συντονισμού συχνά συνεπάγεται με την υποβολή του συστήματος σε μια βαθμιαία αλλαγή στην είσοδο του. Η κακή ρύθμιση του PID μπορεί να προκαλέσει πολλά προβλήματα στην απόκριση του συστήματος όπως είναι οι αργοί χρόνοι ύψωσης, οι ταλαντώσεις και μη επαρκής απόσβεση.

- **Χειροκίνητη Ρύθμιση**

Εάν το σύστημα πρέπει να παραμείνει συνδεδεμένο, μια μέθοδος συντονισμού είναι να οριστούν πρώτα οι τιμές  $K_p$ ,  $K_i$  και  $K_d$  στο μηδέν. Έπειτα αυξάνουμε το  $K_p$  έως ότου το σύστημα παρουσιάσει μια μικρή ταλάντωση, τότε το  $K_p$  θα πρέπει να οριστεί στο μισό περίπου αυτής της τιμής. Στη συνέχεια, αυξάνουμε το  $K_i$  έως ότου διορθωθεί οποιαδήποτε μετατόπιση σε επαρκή χρόνο για τη διαδικασία. Αν αυξήσουμε πάρα πολύ το  $K_i$  θα προκαλέσει αστάθεια στο σύστημα. Τέλος, αυξάνουμε το  $K_d$ , εάν απαιτείται, έως ότου βελτιώσουμε την σταθερότητα του συστήματος με γνώμονα ότι το μεγάλο διαφορικό κέρδος θα προκαλέσει υπερβολική ανταπόκριση σε μεταβολές και θόρυβο. Αν αυξήσουμε πάρα πολύ το  $K_d$  θα προκαλέσει υπερβολική απόκριση και υπέρβαση στο σύστημα. Οι τιμές αυτές υπολογίζονται όταν καταλήξουμε στις τιμές που θα προσφέρουν στο σύστημα μας ικανοποιητική υπερύψωση, ταχεία απόκριση και μικρό μόνιμο σφάλμα. Ένας γρήγορος συντονισμός του ελεγκτή PID συνήθως ξεπερνά ελαφρώς για να φτάσει το σημείο ρύθμισης πιο γρήγορα. Ωστόσο, ορισμένα συστήματα δεν μπορούν να δεχτούν υπερβολική λήψη.

- **Μέθοδος Ziegler – Nichols**

Η μέθοδος Ziegler-Nichols, γνωστή και ως *μέθοδος ορίου ευστάθειας*, είναι μια από τις πιο γνωστές μεθόδους συντονισμού ενός ελεγκτή PID. Αναπτύχθηκε από τους John G. Ziegler και Nathaniel B. Nichols τη δεκαετία του 1940. Όπως και στην παραπάνω μέθοδο, το  $K_i$  και το  $K_d$  ορίζονται πρώτα στο μηδέν. Το αναλογικό κέρδος αυξάνεται έως ότου φτάσει στο απόλυτο κέρδος,  $K_u$ , στο οποίο η έξοδος του ελεγκτή αρχίζει να ταλαντεύεται συνεχώς. Το  $T_u$  είναι η περίοδος ταλάντωσης.

Οι παράμετροι του PID Controller μπορούν να προσδιοριστούν σύμφωνα με τον παρακάτω πίνακα βάσει των τιμών  $K_u$  και  $T_u$ :

Εκλεκτές	$K_p$	$K_i$	$K_d$
<b>P</b>	$0.50K_u$	-	-
<b>PI</b>	$0.45K_u$	$0.54K_u/T_u$	-
<b>PID</b>	$0.60K_u$	$1.2K_u/T_u$	$3K_uT_u/40$

Πίνακας 2 Παράμετροι Ziegler - Nichols

- **Μέθοδος Tyreus – Luyben**

Η μέθοδος Tyreus - Luyben μοιάζει αρκετά με τη μέθοδο Ziegler - Nichols αλλά οι ρυθμίσεις του τελικού ελεγκτή είναι διαφορετικές. Επίσης, αυτή η μέθοδος προτείνει μόνο ρυθμίσεις για ελεγκτές PI και PID. Αυτές οι ρυθμίσεις που βασίζονται στο απόλυτο κέρδος και την περίοδο δίνονται στο παρακάτω πίνακα:

Εκλεκτές	$K_p$	$K_i$	$K_d$
<b>PI</b>	$K_u/3.2$	$2.2T_u$	-
<b>PID</b>	$K_u/2.2$	$2.2T_u$	$T_u/6.3$

Πίνακας 3: Παράμετροι Tyreus - Luyben

Όπως και η μέθοδος Ziegler - Nichols, αυτή η μέθοδος είναι χρονοβόρα και αναγκάζει το σύστημα να περιθωριοποιήσει την αστάθεια. Πολλοί άλλοι αλγόριθμοι έχουν προταθεί για την επίλυση αυτών των προβλημάτων λαμβάνοντας κρίσιμα δεδομένα (απόλυτο κέρδος και συχνότητα) υπό πιο αποδεκτές συνθήκες.

- **Μέθοδος Cohen – Coon**

Η μέθοδος αυτή αναπτύχθηκε το 1953. Είναι παρόμοια με τη μέθοδο Ziegler – Nichols. Αναπτύχθηκε ένα σύνολο παραμέτρων συντονισμού για να δώσει μια απόκριση κλειστού συστήματος με συντελεστή απόσβεσής  $1/4$ . Αναμφισβήτητα το μεγαλύτερο πρόβλημα με αυτές τις παραμέτρους είναι ότι μια μικρή αλλαγή, θα μπορούσε να προκαλέσει ασταθές σύστημα κλειστού βρόχου.

- **Μέθοδος Astrom – Hagglund (Relay)**

Η μέθοδος αυτή δημοσιεύθηκε το 1984 από τον Karl Johan Åström και τον Tore Hägglund, Η μέθοδος ρελέ λειτουργεί προσωρινά τη διαδικασία χρησιμοποιώντας έλεγχο bang-bang και μετρά τις ταλαντώσεις που προκύπτουν. Η έξοδος αλλάζει (σαν ένα ρελέ, εξ ου και το όνομα) μεταξύ δύο τιμών της μεταβλητής ελέγχου. Οι τιμές πρέπει να επιλεγθούν έτσι ώστε η διαδικασία να ξεπεράσει το σημείο ρύθμισης, αλλά δεν πρέπει να είναι 0% και 100%. Επιλέγοντας κατάλληλες τιμές, μπορούν να αποφευχθούν οι επικίνδυνες ταλαντώσεις.

Εφόσον η μεταβλητή της διαδικασίας είναι κάτω από το σημείο ρύθμισης, η έξοδος του ελεγκτή ρυθμίζεται στην υψηλότερη τιμή. Μόλις ανεβεί πάνω από το σημείο ρύθμισης, η έξοδος ρυθμίζεται στην χαμηλότερη τιμή. Στην ιδανική περίπτωση, η κυματομορφή της εξόδου είναι σχεδόν τετράγωνη, περνώντας ίσο χρόνο πάνω και κάτω από το σημείο ρύθμισης. Η περίοδος και το πλάτος των ταλαντώσεων που προκύπτουν μετρούνται και χρησιμοποιούνται για τον υπολογισμό του τελικού κέρδους και της περιόδου, τα οποία στη συνέχεια τροφοδοτούνται στη μέθοδο Ziegler - Nichols.

Συγκεκριμένα, η τελική περίοδος  $T_u$  θεωρείται ότι είναι ίση με την παρατηρούμενη περίοδο και το τελικό κέρδος υπολογίζεται ως  $K_u = 4b / \pi a$ , όπου  $a$  είναι το πλάτος της μεταβλητής ταλάντωσης της διαδικασίας και  $b$  είναι το πλάτος της αλλαγής εξόδου ελέγχου που την προκάλεσε.

- **Μέθοδος Λογισμικού**

Οι περισσότερες σύγχρονες βιομηχανικές εγκαταστάσεις δεν συντονίζουν πλέον τα συστήματα χρησιμοποιώντας τις μεθόδους μη αυτόματου υπολογισμού που φαίνονται παραπάνω. Το λογισμικό βελτιστοποίησης PID και βελτιστοποίησης συστήματος χρησιμοποιείται για να εξασφαλίσει συνεπή αποτελέσματα. Αυτά τα πακέτα λογισμικού θα συλλέξουν τα δεδομένα, θα αναπτύξουν μοντέλα διεργασιών και θα προτείνουν το βέλτιστο συντονισμό. Ορισμένα πακέτα λογισμικού μπορούν ακόμη και να αναπτύξουν συντονισμό συλλέγοντας δεδομένα από αλλαγές αναφοράς.

Ο μαθηματικός συντονισμός ελεγκτή PID προκαλεί ώθηση στο σύστημα και στη συνέχεια χρησιμοποιεί την απόκριση συχνότητας του ελεγχόμενου συστήματος για να σχεδιάσει τις τιμές του ελεγκτή PID. Σε ελεγκτές με χρόνους απόκρισης αρκετών λεπτών, συνιστάται ο μαθηματικός συντονισμός, επειδή η δοκιμή και το σφάλμα μπορεί να διαρκέσουν ημέρες μόνο για να βρουν ένα σταθερό σύνολο τιμών ελεγκτών. Οι βέλτιστες τιμές είναι πιο δύσκολο να βρεθούν. Ορισμένοι ελεγκτές ψηφιακού βρόχου προσφέρουν δυνατότητα αυτόματου συντονισμού στην οποία αποστέλλονται στη διαδικασία πολύ μικρές αλλαγές σημείου ρύθμισης, επιτρέποντας στον ίδιο τον ελεγκτή να υπολογίσει τις βέλτιστες τιμές συντονισμού. Μια άλλη προσέγγιση υπολογίζει τις αρχικές τιμές μέσω της μεθόδου Ziegler – Nichols και χρησιμοποιεί μια τεχνική αριθμητικής βελτιστοποίησης για να βρει καλύτερους συντελεστές PID.

Άλλοι τύποι είναι διαθέσιμοι για το συντονισμό του βρόχου σύμφωνα με διαφορετικά κριτήρια απόδοσης. Πολλοί πατενταρισμένοι τύποι ενσωματώνονται τώρα σε μονάδες λογισμικού και υλικού συντονισμού PID. Οι εξελίξεις στο αυτοματοποιημένο λογισμικό συντονισμού ελεγκτή PID παρέχουν επίσης

αλγόριθμους για τον συντονισμό ελεγκτή PID σε ένα σενάριο δυναμικής ή μη σταθερής κατάστασης (NSS). Το λογισμικό θα μοντελοποιήσει τη δυναμική μιας διαδικασίας, μέσω μιας διαταραχής, και θα υπολογίσει τις παραμέτρους ελέγχου PID σε απόκριση.

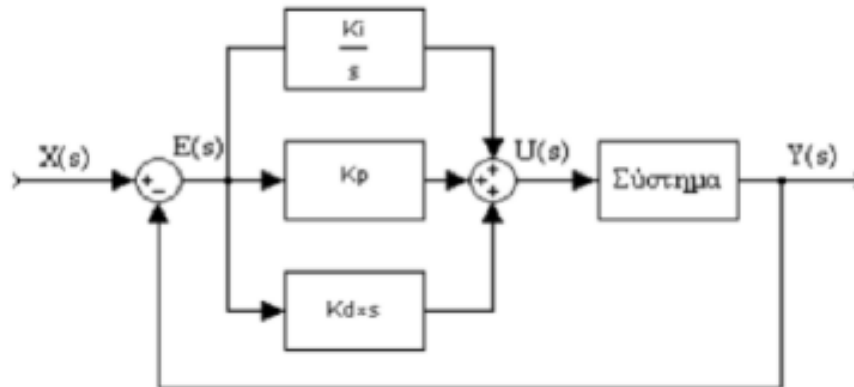
#### 4.1.3. Είδη Ελεγκτών PID

- **Ελεγκτής PID (Αναλογικός – Ολοκληρωτικός – Διαφορικός)**

Ο ελεγκτής PID είναι ευρέως διαδεδομένος και προτείνεται για αυτή την εργασία, λόγω της απλότητας του αλλά και της ορθής συμπεριφοράς του σε τεράστια κλίμακα συνθηκών. Ο PID Controller χρησιμοποιείται ευρέως στον έλεγχο κίνησης λόγω του απλού αλγόριθμου καθώς και της υψηλής αξιοπιστίας του. Όμως μερικά από τα ελεγχόμενα αντικείμενα του δεν έχουν ορισμένο κανένα ακριβές μαθηματικό μοντέλο, το οποίο να οδηγεί στην ρύθμιση των παραμέτρων PID. Έτσι, οι παράμετροι έχουν συνήθως κακή απόδοση και είναι δύσκολο να ικανοποιήσουν τον έλεγχο κίνησης υψηλής ακρίβειας του γραμμικού κινητήρα. Αν όμως χρησιμοποιηθεί ο αλγόριθμος, βασισμένος στο εκάστοτε σύστημα, για τον ορισμό των παραμέτρων PID, το σύστημα ελέγχου μπορεί όχι μόνο να διατηρήσει τα πλεονεκτήματα των απλών αρχών και τη χρήση του συμβατικού συστήματος ελέγχου PID, αλλά και να αποκτήσει χαρακτηριστικά όπως η ευελιξία και η προσαρμοστικότητα, τα οποία μπορούν να βελτιώσουν αποτελεσματικά την απόδοση του.

Ο αλγόριθμος PID είναι ο πιο δημοφιλής ελεγκτής ανάδρασης που χρησιμοποιείται ευρέως στο σύστημα ελέγχου. Η μέθοδος PID είναι ένα από τα πιο διαδεδομένα ανατροφοδοτούμενα συστήματα ελέγχου που έχει χρησιμοποιηθεί πριν από περισσότερα από 50 χρόνια. Ένα από τα πρώτα παραδείγματα ενός ελεγκτή τύπου PID αναπτύχθηκε από τον Elmer Sperry το 1911, ενώ η πρώτη δημοσιευμένη θεωρητική ανάλυση ενός PID Controller, ήταν από τον Ρώσο Αμερικάνο μηχανικό Nicolas Minorsky το 1922. Με την σωστή επιλογή των όρων  $K_d$ ,  $K_p$  και  $K_i$ , ο PID Controller συνδυάζει μικρό χρόνο ανόδου, μικρή υπερύψωση, μικρό χρόνο αποκατάστασης και μηδενικό σφάλμα στην μόνιμη κατάσταση.

Η μορφή ενός PID ελεγκτή είναι η ακόλουθη:



Εικόνα 28 – PID Controller

- **Ελεγκτής PI (Αναλογικός – Ολοκληρωτικός)**

Ο ελεγκτής PI επιτυγχάνετε με τον συνδυασμό αναλογικού και ολοκληρωτικού. Επιτυγχάνεται η εξάλειψη του σφάλματος στην μόνιμη κατάσταση και αυξάνεται η ευστάθεια του συστήματος. Με τον PI έλεγχο βελτιώνεται το σφάλμα στην μόνιμη κατάσταση. Αυξάνεται όμως η υπερύψωση των ταλαντώσεων και το σύστημα μπορεί να πέσει σε αστάθεια.

- **Ελεγκτής PD (Αναλογικός – Διαφορικός)**

Ο ελεγκτής PD επιτυγχάνετε με τον συνδυασμό αναλογικού και διαφορικού. Επιτυγχάνεται η καλύτερη απόσβεση του συστήματος άρα και κατά συνέπεια μείωση των ταλαντώσεων, αλλά και ελάττωση του σφάλματος στην μόνιμη κατάσταση.



- **Ελεγκτής P (Αναλογικός)**

Ο ελεγκτής P πολλαπλασιάζει το σφάλμα που δέχεται το σύστημα και ουσιαστικά επιταχύνει την επίτευξη της τελικής τιμής. Στα κλειστά συστήματα, με την σωστή επιλογή του αναλογικού όρου ( $K_p$ ), σταθεροποιείται το σήμα εξόδου λόγω της εξάλειψης των ταλαντώσεων στη μόνιμη κατάσταση. Πολλές φορές όμως, δημιουργείται σφάλμα μόνιμης κατάστασης που παρά την μείωση του, δεν μπορεί να εξαλειφθεί πλήρως. Εάν αυξήσουμε το κέρδος του ελεγκτή  $K_p$ , μπορεί να μειώνεται το σφάλμα στο σύστημα όμως ταυτόχρονα αυξάνονται οι ταλαντώσεις του.

- **Ελεγκτής I (Ολοκληρωτικός)**

Ο ελεγκτής I αθροίζει το σφάλμα στο χρόνο. Όσο το σφάλμα μειώνεται, το ολοκλήρωμα τείνει σε σταθερή τιμή που θα είναι η τελική τιμή. Προσθέτοντας τον ολοκληρωτικό όρο, το σφάλμα στην μόνιμη κατάσταση μηδενίζεται, όμως αυξάνεται ο χρόνος αποκατάστασης του συστήματος. Το ολοκλήρωμα του σφάλματος με σταθερό κέρδος προσφέρει στο σύστημα την εκμηδένιση του μόνιμου σφάλματος σε βηματική απόκριση. Όμως, η φυσική συχνότητα του συστήματος αυξάνεται με αποτέλεσμα να μειώνεται ο συντελεστής απόσβεσης του συστήματος και να χειροτερεύει η ευστάθειά του.

- **Ελεγκτής D (Διαφορικός)**

Ο ελεγκτής D προσθέτει τον διαφορικό όρο επιτυγχάνοντας την αύξηση της ευστάθειας του κλειστού συστήματος και η βελτίωση της μεταβατικής απόκρισης. Η διαφορίση του σφάλματος με σταθερό κέρδος επιφέρει στο σύστημα βελτίωση της ταχύτητας απόκρισης αλλά δεν επηρεάζει το σφάλμα στην μόνιμη κατάσταση. Γενικότερα, αποφεύγεται η χρήση του είτε συνηθίζεται να χρησιμοποιείται κάποιο φίλτρο συνδυαστικό με την παραγωγή.

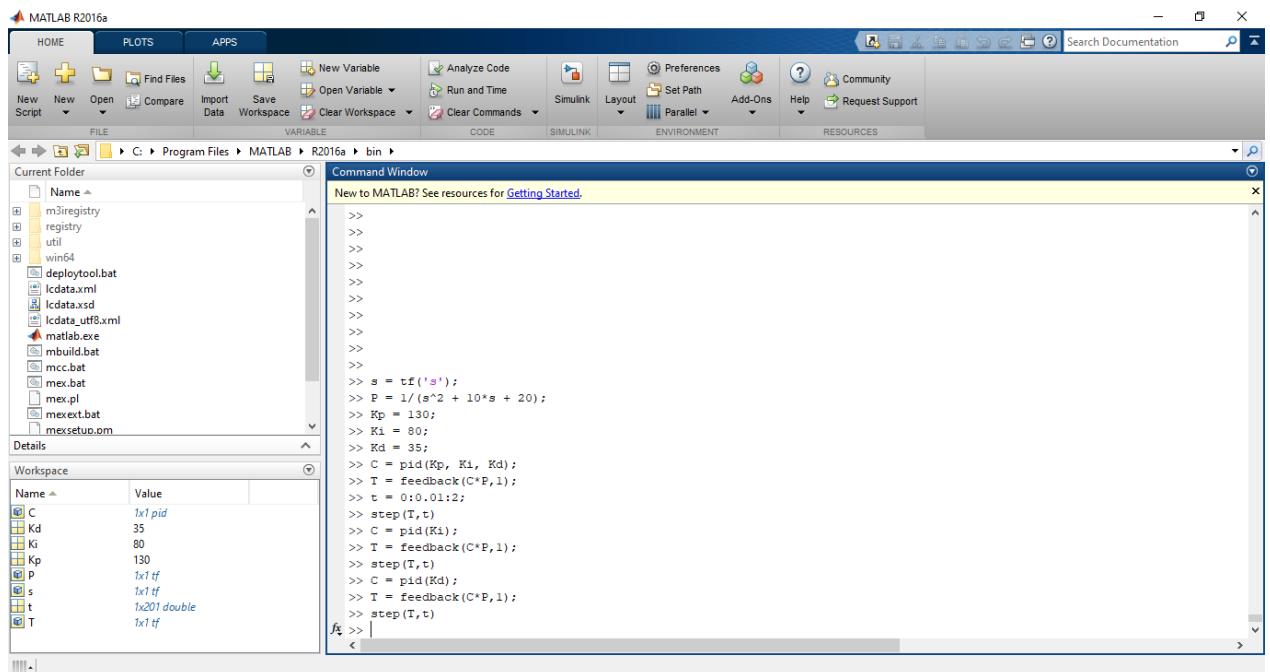
Ανάπτυξη PID ελεγκτή θερμοκρασίας με τη βοήθεια Raspberry Pi και Matlab

Ας δούμε τις γραφικές παραστάσεις απόκρισης του συστήματος με σταθερούς τους όρους  $K_p = 130$ ,  $K_i = 80$  και  $K_d = 35$ . Η συνάρτηση που θα χρησιμοποιήσουμε είναι η εξής:

$$\frac{X(s)}{F(s)} = \frac{1}{s^2 + 10s + 20}$$

Εικόνα 29 – Συνάρτηση Παραδείγματος

Οι εντολές στο Matlab είναι οι εξής:

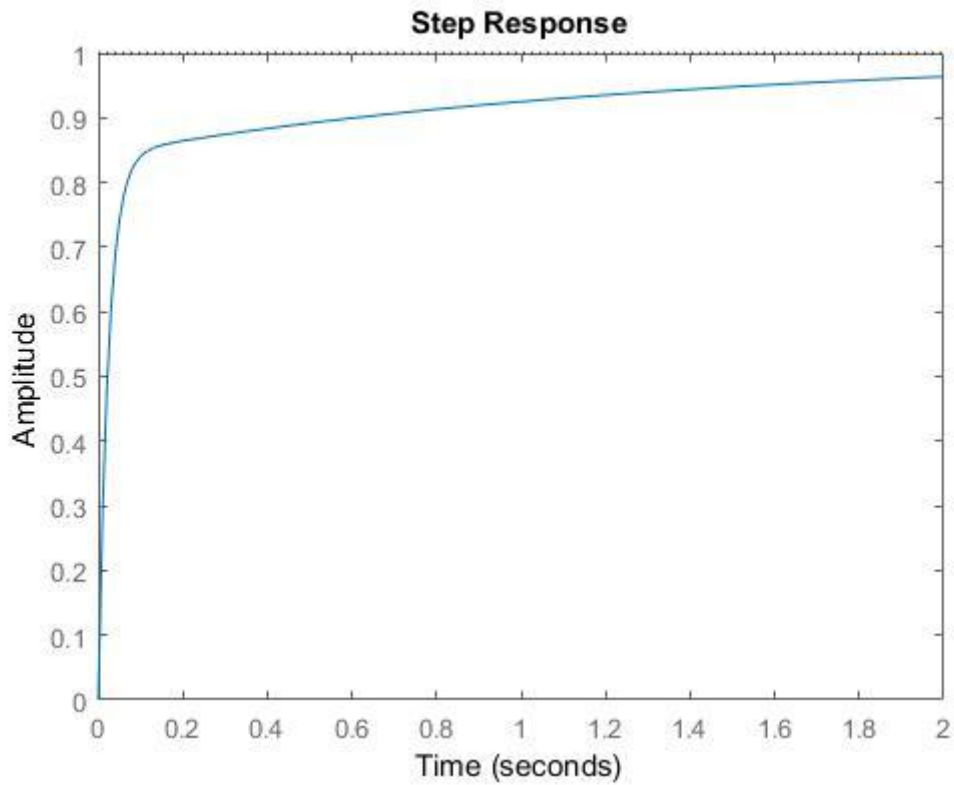


```
>> s = tf('s');
>> P = 1/(s^2 + 10*s + 20);
>> Kp = 130;
>> Ki = 80;
>> Kd = 35;
>> C = pid(Kp, Ki, Kd);
>> T = feedback(C*P,1);
>> t = 0:0.01:2;
>> step(T,t)
>> C = pid(Ki);
>> T = feedback(C*P,1);
>> step(T,t)
>> C = pid(Kd);
>> T = feedback(C*P,1);
>> step(T,t)
>>
```

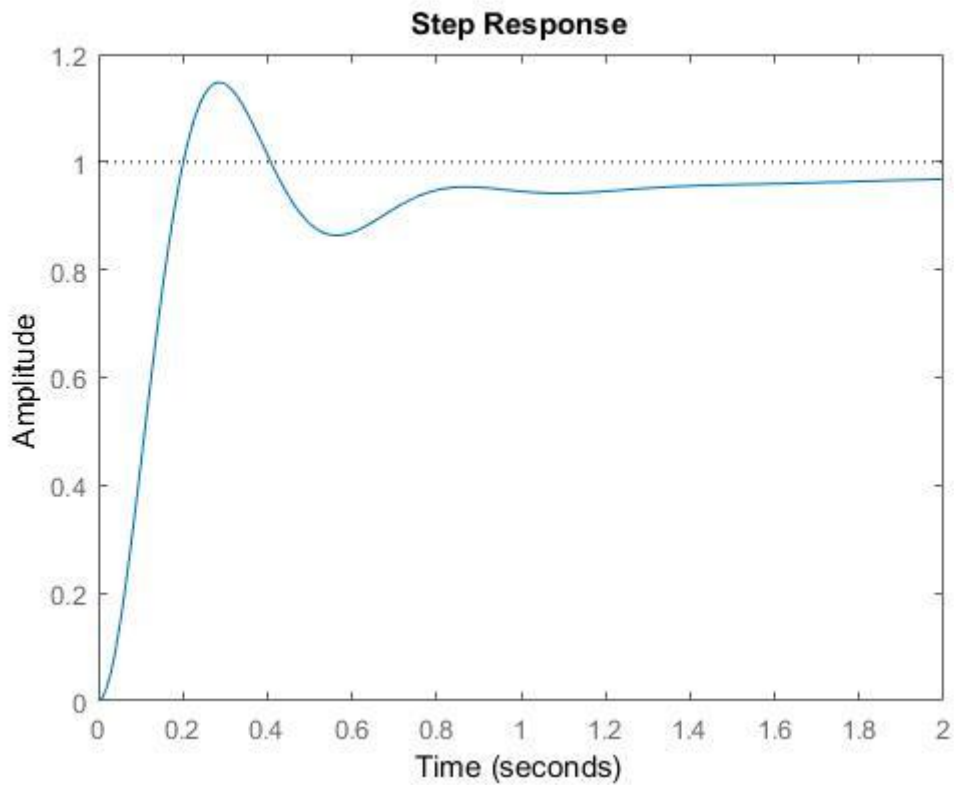
Name	Value
C	1x1 pid
Kd	35
Ki	80
Kp	130
P	1x1 tf
s	1x1 tf
t	1x201 double
T	1x1 tf

Εικόνα 30 – Εντολές Παραδείγματος στο Matlab.

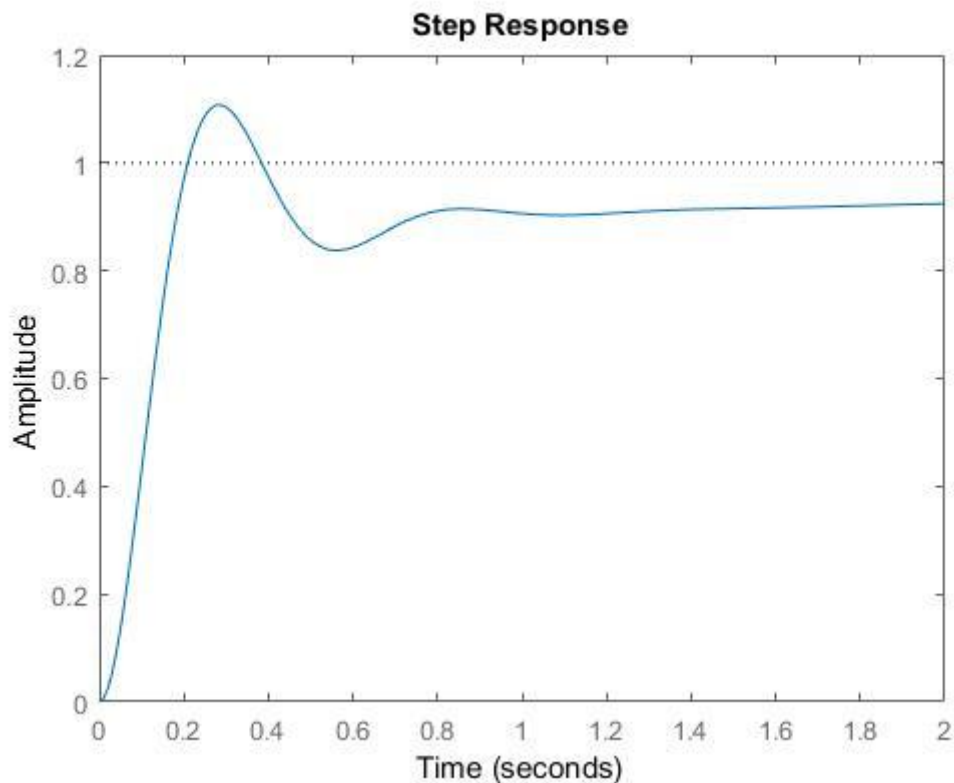
Από τις γραφικές παραστάσεις της απόκρισης φαίνεται ότι ο PID ελεγκτής παρουσιάζει τη βέλτιστη απόκριση ενώ ο PI και ο PD ελεγκτές μειώνουν την υπερύψωση και το χρόνο απόκρισης ενώ έχουν μικρή επιρροή στο χρόνο ανύψωσης και στο μόνιμο σφάλμα.



Εικόνα 31 – Απόκριση PID Controller



Εικόνα 32 – Απόκριση PI Controller



Εικόνα 33 – Απόκριση PD Controller

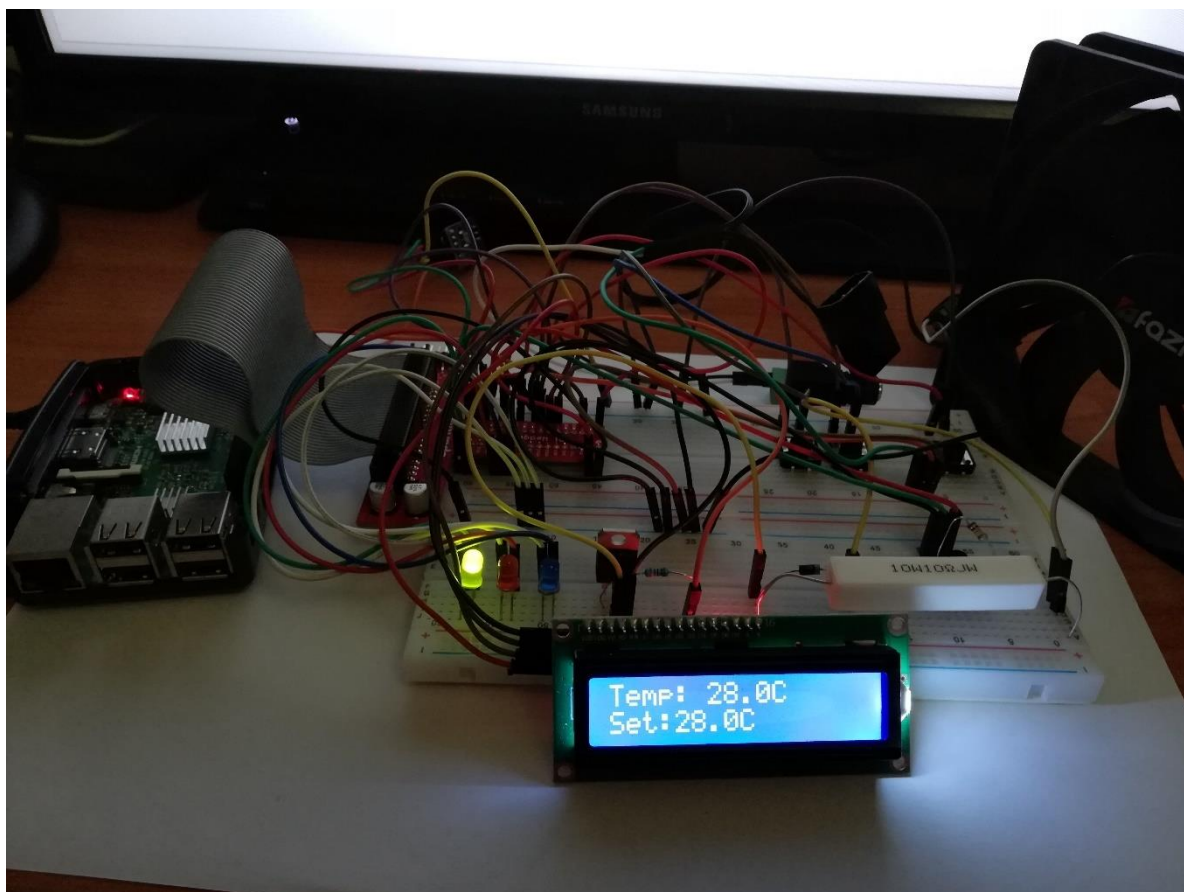
Εν κατακλείδι, με την αύξηση του αναλογικού κέρδους, το μόνιμο σφάλμα ελαττώνεται. Εάν όμως, το αυξήσουμε πολύ, το σύστημα γίνεται ασταθές. Με τον ολοκληρωτικό έλεγχο, επιτυγχάνεται μείωση του μόνιμου σφάλματος, όμως συχνά μειώνεται η ευστάθεια του συστήματος και με τον ολοκληρωτικό έλεγχο, επιτυγχάνεται μείωση του μόνιμου σφάλματος, όμως συχνά μειώνεται η ευστάθεια του συστήματος.

## 5. ΚΕΦΑΛΑΙΟ 5

### 5.1. Υλικά Κατασκευής

Τα υλικά που χρησιμοποιήθηκαν για την κατασκευή είναι τα εξής:

- Raspberry Pi 3 Model B
- Micro SD card
- LCD Module Display Blue 16X2 Character
- DALLAS 18B20 (αισθητήρας θερμοκρασίας)
- Fan 12V
- Transistor NPN 2N2222A
- Transistor BD243C
- Βαττική Αντίσταση (10W10ΩJW)
- Αντίσταση (10KΩ)
- 3 Κουμπιά
- 3 Λαμπάκια Led (Πράσινο, Κόκκινο, Μπλε)
- Θηλυκά και αρσενικά καλώδια (jumpers)
- 3 Εξωτερικοί φορτιστές μπαταριών (5V, 9V, 12V)
- Πλακέτα (Breadboard)
- Εξωτερική θήκη raspberry pi
- Οθόνη
- Πληκτρολόγιο
- Ποντίκι



Εικόνα 34 – Κατασκευή

### 5.1.1. Raspberry Pi

Για την πτυχιακή εργασία χρησιμοποιήθηκε το Raspberry Pi Model B. Κυκλοφόρησε τον Φεβρουάριο του 2016 με τον μικροελεγκτή BCM2837, με τετραπύρρηνο επεξεργαστή Cortex-A53 (ARMv8-A 64/32-bit) στα 1.2GHz, μνήμη 1 GB, 4 θύρες USB 2.0, 1 θύρα Ethernet, Wi-Fi 802.11 wireless b/g/n single band 2.4GHz, Bluetooth 4.1 BLE, 1 θύρα HDMI, 1 CSI Camera Port, 1 θύρα για micro SD Card, 1 DSI Display Port και 40 ακίδες.

Από τις 40 ακίδες που έχει, 2 δίνουν 5V ρεύμα, 2 δίνουν 3.3V ρεύμα, 8 είναι για γείωση (Ground) και τα υπόλοιπα 28 Pin έχουν κάποιες συγκεκριμένες λειτουργίες για την επικοινωνία και την σύνδεση του Raspberry Pi με άλλες συσκευές. Για την τροφοδοσία του Raspberry Pi χιάζεται 5V από την θύρα micro usb.



Εικόνα 35 – Raspberry Pi Model B

### 5.1.2. Micro SD Card

Στην Micro SD Card είναι αποθηκευμένο το λογισμικό αλλά και γενικά ο εσωτερικός αποθηκευτικός χώρος του Raspberry Pi.

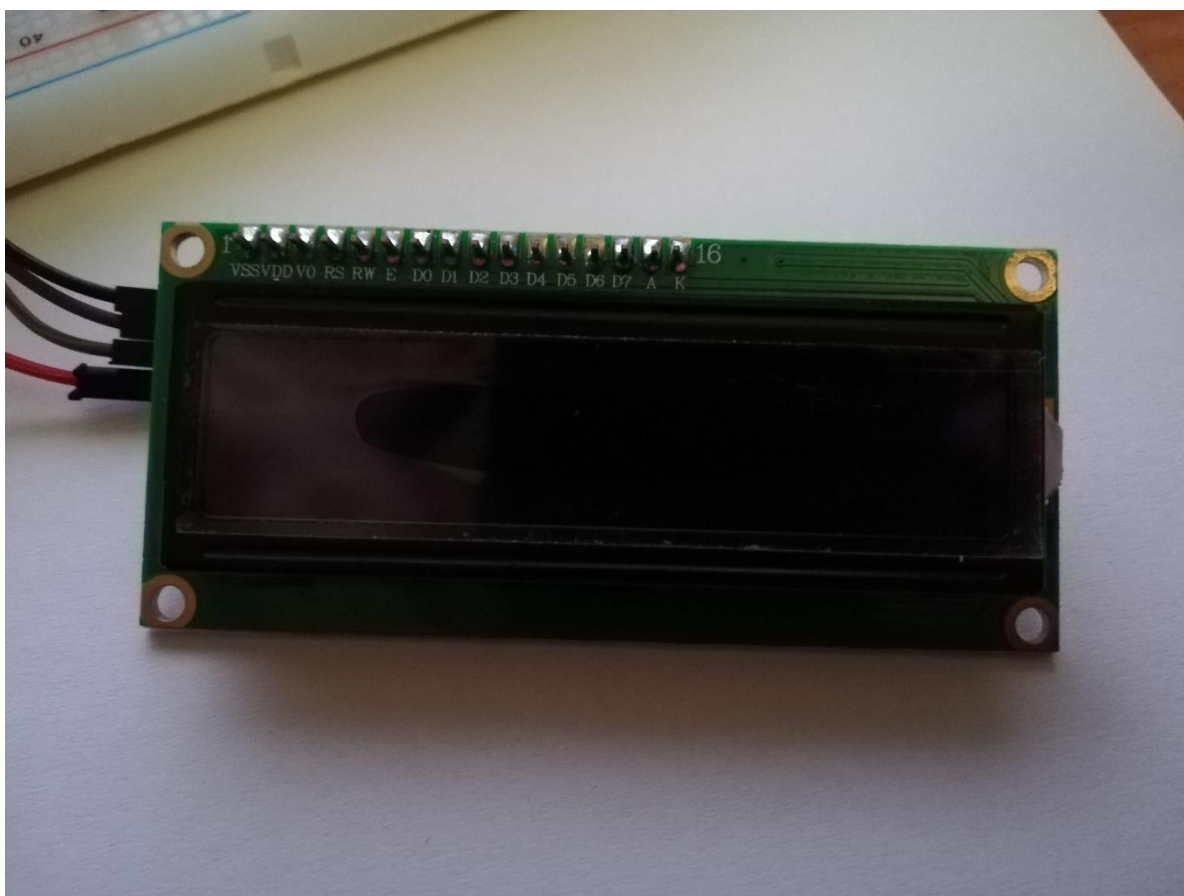


Εικόνα 36 – SD Card



### 5.1.3. LCD Module Display

Η LCD (Liquid Crystal Display) είναι μια επίπεδη οθόνη και μπορεί να εμφανίσει 2 γραμμές από 16 χαρακτήρες η κάθε μία. Η οθόνη LCD θα εμφανίζει τις τιμές που θέλουμε να υπολογίσουμε (θερμοκρασία,  $K_p$ ,  $K_i$ ,  $K_d$ ). Η LCD έχει από την πίσω πλευρά ένα I2C backpack για την σύνδεση. Έχει 4 ακίδες (GND, VCC, SDA, SCL).



Εικόνα 37 – LCD Display

### 5.1.4. DALLAS 18B20

Ο αισθητήρας θερμοκρασίας DALLAS 18B20 είναι ιδανικός διότι λίγοι αισθητήρες είναι τόσο εύκολο να εγκατασταθούν στο Raspberry Pi. Έχει το ίδιο μέγεθος με ένα τρανζίστορ και χρησιμοποιούν μόνο ένα καλώδιο για το σήμα δεδομένων. Είναι επίσης εξαιρετικά ακριβείς και λαμβάνουν γρήγορα μετρήσεις. Το

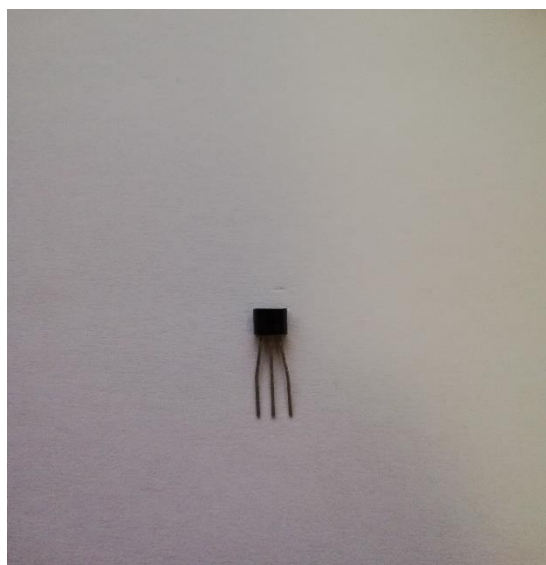


μόνο άλλο συστατικό που χρειάζεστε είναι μια αντίσταση 4,7K Ohm ή 10K Ohm. Το DS18B20 επικοινωνεί με το πρωτόκολλο επικοινωνίας «One-Wire», ένα ιδιόκτητο πρωτόκολλο σειριακής επικοινωνίας που χρησιμοποιεί μόνο ένα καλώδιο για τη μετάδοση των μετρήσεων θερμοκρασίας στον μικροελεγκτή.

Το DS18B20 μπορεί να λειτουργήσει με τον τρόπο που είναι γνωστός ως τρόπος λειτουργίας παρασίτων . Κανονικά το DS18B20 χρειάζεται τρία καλώδια για λειτουργία: τα καλώδια VCC, Ground και data wires. Στη λειτουργία παρασίτων, χρησιμοποιούνται μόνο οι γραμμές Ground και data wires και τροφοδοτείται μέσω του data wires. Διαθέτει επίσης μια λειτουργία συναγερμού που μπορεί να ρυθμιστεί ώστε να εξάγει ένα σήμα όταν η θερμοκρασία υπερβαίνει ένα υψηλό ή χαμηλό επίπεδο που έχει ρυθμίσει ο χρήστης.

Τα χαρακτηριστικά του DS18B20 είναι:

- Θερμοκρασία από -55 ° C έως 125 ° C.
- Τάση λειτουργίας από 3,0V έως 5,0V.
- Δείγμα κάθε 750 ms
- 64 bit μοναδική διεύθυνση
- Πρωτόκολλο επικοινωνίας One-Wire



Εικόνα 38 – DS18B20

### 5.1.5. Fan 12V

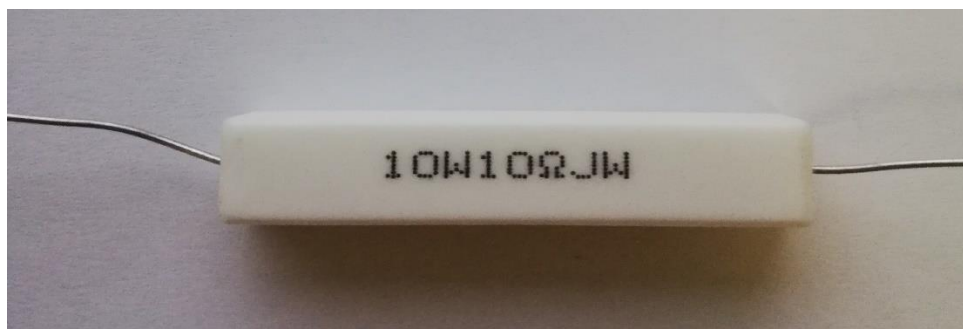
Το ανεμιστήρα 12V θα τον χρειαστούμε για να κατεβάσουμε την θερμοκρασία.



Εικόνα 39 – Fan 12V

### 5.1.6. Βαττική Αντίσταση

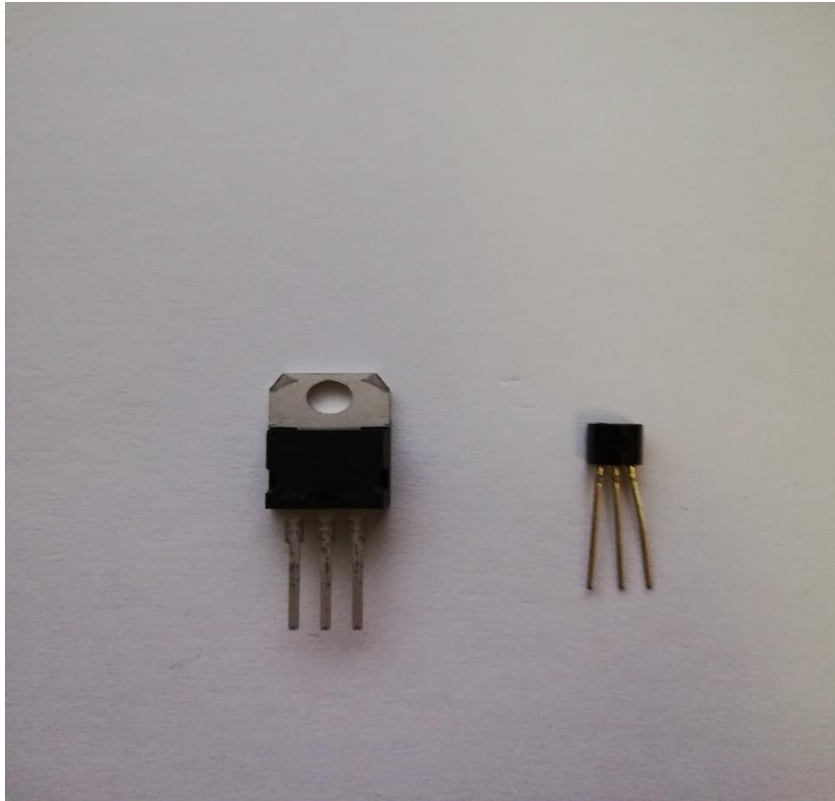
Την Βαττική Αντίσταση (Power Resistor) θα την χρειαστούμε για να ανεβάζουμε την θερμοκρασία. Η βαττική αντίσταση θα ενεργοποιείται από ένα transistor ηρη. Χρειάζεται εξωτερική τροφοδοσία 12V. Η αντίσταση που θα χρησιμοποιήσουμε είναι η (10W10ΩJW) που βγαίνει από τον τύπο  $Ισχύς = V^2/R$ . Για παράδειγμα, η τάση στα άκρα της αντίστασης είναι 12V αφαιρούμε την τάση στο transistor (π.χ. 2V) η αντίσταση είναι 10W άρα έχουμε  $10 = 10^2/R$ ,  $R = 10Ω$ .



Εικόνα 40 – Βαττική Αντίσταση 10W10ΩJW

### 5.1.7. Transistors

Χρησιμοποιήθηκαν 2 Transistors το 2N2222A και το BD243C. Έχουν 3 ακίδες, τον εκπομπό που είναι συνδεδεμένη με την γείωση, την βάση η οποία χρησιμοποιείται για την ενεργοποίηση/απενεργοποίηση του τρανζίστορ και τον συλλέκτη.



Εικόνα 41 – Transistors BD243C και 2N2222A.

### 5.1.8. Υπόλοιπα υλικά

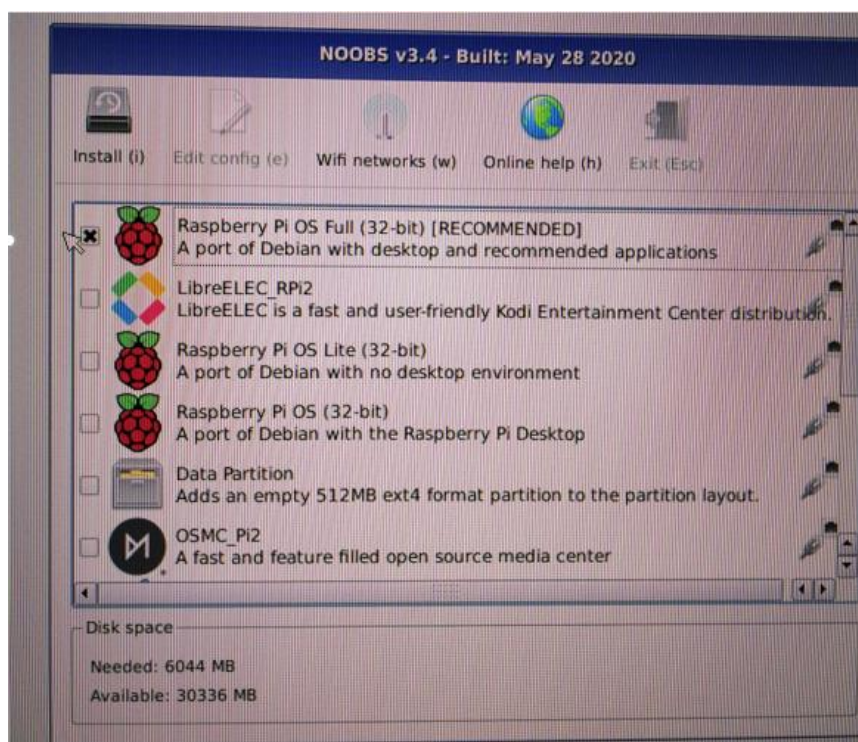
Τα υπόλοιπα υλικά που χρησιμοποιήθηκαν είναι αντιστάσεις, 3 κουμπιά για την ρύθμιση της θερμοκρασίας, 3 Led λαμπάκια (Πράσινο, Κόκκινο, Μπλε) που ανάβουν ανάλογα με την θερμοκρασία του αισθητήρα ( $t$ ) και την επιθυμητή θερμοκρασία ( $st$ ). Αν  $st > t$  ανάβει το μπλε, αν  $st < t$  ανάβει το κόκκινο και αν  $st = t$  ανάβει το μπλε. Θηλυκά και αρσενικά καλώδια (jumpers) για την συνδεσμολογία

των υλικών με το Raspberry Pi, πλακέτα (Breadboard) για την τοποθέτηση των υλικών.

## 5.2. Εγκατάσταση Raspberry Pi

Αρχικά πρέπει να εγκαταστήσουμε το λειτουργικό στο Raspberry Pi. Για την εγκατάσταση του λειτουργικού θα κατεβάσουμε από το site του Raspberry ([www.raspberrypi.org](http://www.raspberrypi.org)) το Noobs που είναι προτεινόμενη εγκατάσταση για αρχάριους. Βάζουμε την micro SD card στον υπολογιστή και με το SD Card Formatter κάνουμε format την κάρτα έχοντας επιλεγμένο το CHS format size adjustment. Έπειτα αντιγράφουμε το Noobs στην micro SD card.

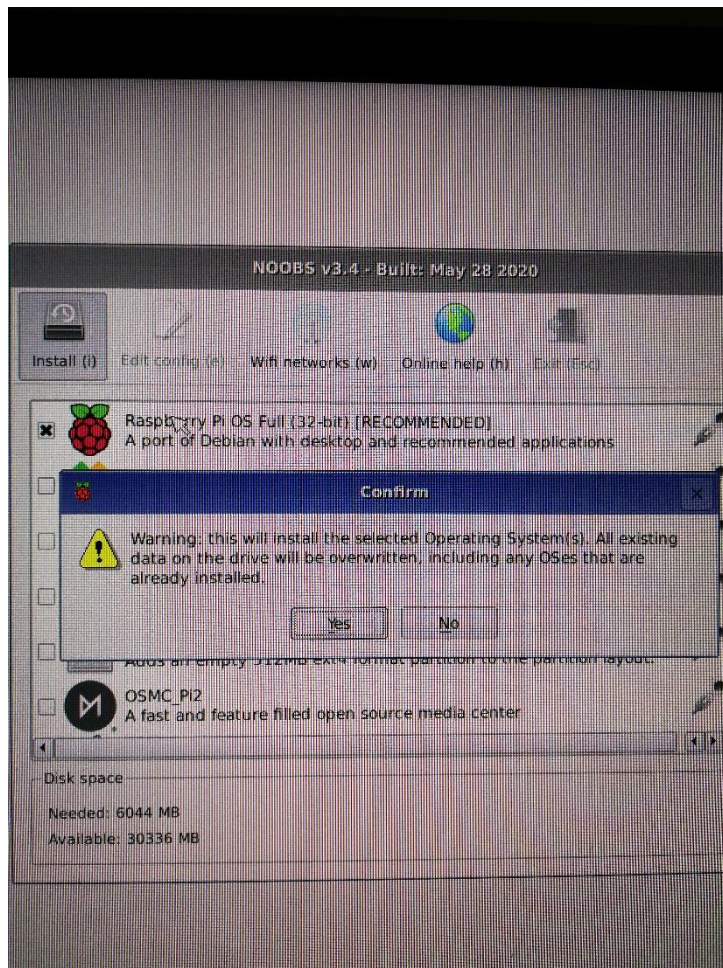
Βάζουμε την micro SD card στο Raspberry Pi. Συνδέουμε μέσω της θύρας HDMI σε μια οθόνη, στις θύρες USB συνδέουμε πληκτρολόγιο και ποντίκι. Έπειτα συνδέουμε το Raspberry Pi με το ρεύμα και στο Internet μέσω της θύρας Ethernet. Όταν ανοίξει, μας εμφανίζει στην οθόνη να επιλέξουμε το λειτουργικό σύστημα που θέλουμε να εγκαταστήσουμε.



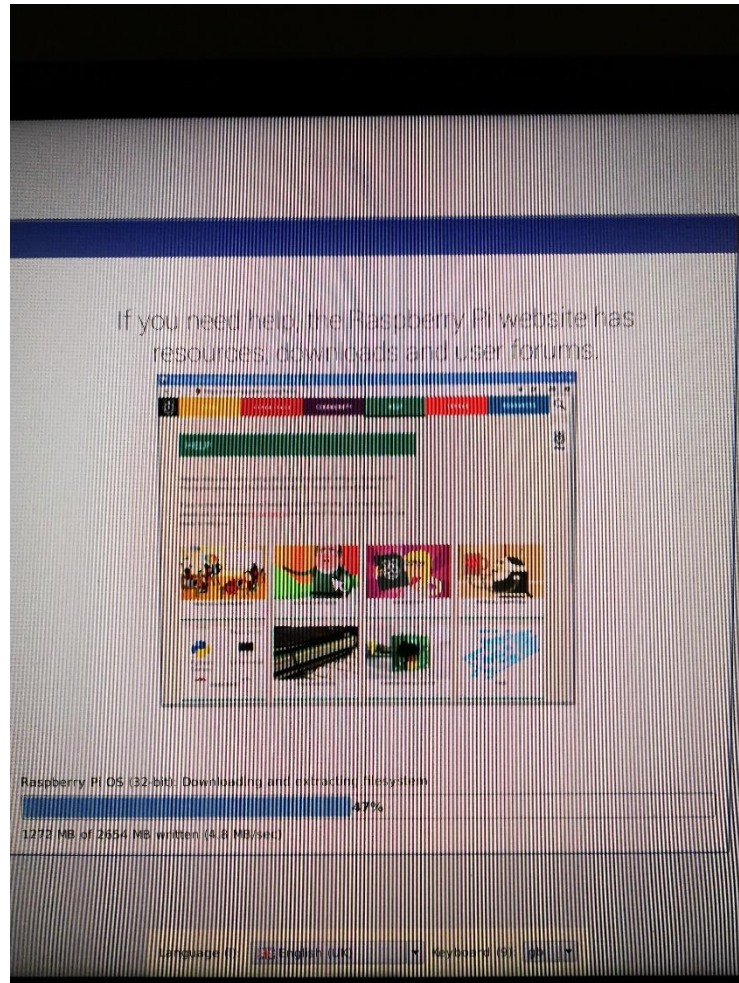
Εικόνα 42 – Εγκατάσταση Λειτουργικό στο Raspberry Pi.



Επιλέγουμε το Raspberry Pi OS Full (32-bit) και πατάμε «Install». Μας βγάζει ένα μήνυμα ότι θα διαγραφούν όλα τα περιεχόμενα της micro SD. Επιλέγουμε «Yes» για να ξεκινήσει η εγκατάσταση του λειτουργικό. Περιμένουμε να εγκατασταθεί το λειτουργικό σύστημα. Ο χρόνος της εγκατάστασης διαφέρει ανάλογα με την ταχύτητα του Internet.

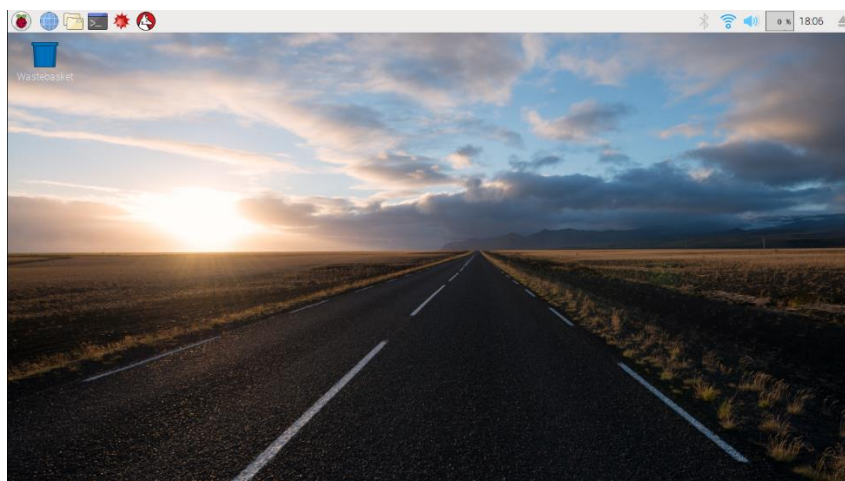


Εικόνα 43 – Εγκατάσταση Λειτουργικό στο Raspberry Pi.



Εικόνα 44 – Εγκατάσταση Λειτουργικό στο Raspberry Pi.

Όταν τελειώσει επιτυχώς η εγκατάσταση θα μας εμφανίσει ένα μήνυμα ότι το λειτουργικό σύστημα εγκαταστάθηκε επιτυχώς. Πατάμε το «OK». Η αρχική οθόνη του Raspberry Pi είναι η εξής:



Εικόνα 45 – Αρχική οθόνη Raspberry Pi.

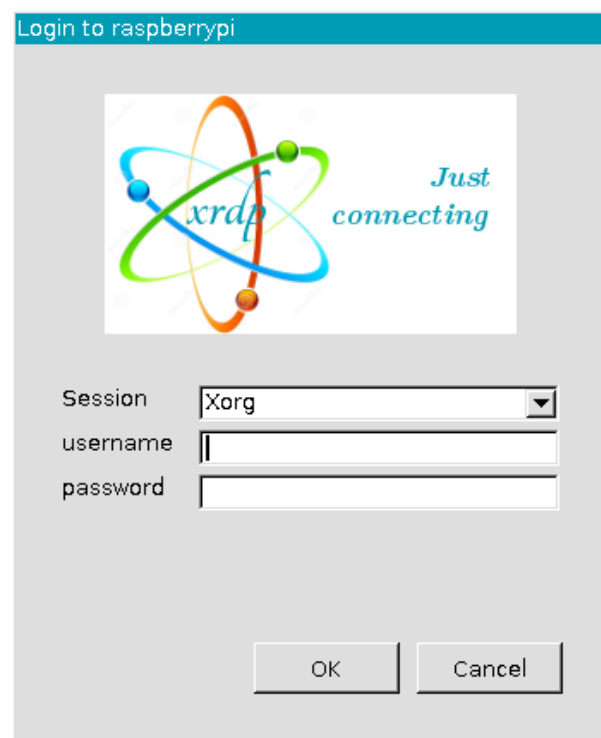
Αρχικά πρέπει να κάνουμε update τι βιβλιοθήκες του Raspberry Pi. Ανοίγουμε το Terminal και τρέχουμε τις ακόλουθες εντολές:

- `sudo apt-get update`
- `sudo apt-get disk-upgrade`

Για να συνδέσουμε το Raspberry Pi με Remote Desktop από τον υπολογιστή πρέπει να εγκαταστήσουμε το xrdp. Τρέχουμε την ακόλουθη εντολή στο Terminal και μετά κάνουμε reboot το Raspberry Pi:

- `sudo apt-get install xrdp`
- `sudo reboot`

Όταν ανοίξει το Raspberry Pi μπορούμε να συνδεθούμε με Remote Desktop, βάζοντας την IP Addressee. Τα Default, Username και Password, είναι pi και raspberry αντίστοιχα.



Εικόνα 46 – Login xrdp.

Έπειτα πρέπει να εγκαταστήσουμε τις βιβλιοθήκες για την LCD οθόνη και για τον αισθητήρα θερμοκρασίας DS18B20.

Για την οθόνη πρέπει να τρέξουμε στο Terminal τα εξής:

- `sudo apt-get install -y python-smbus`
- `sudo apt-get install -y i2c-tools`
- `sudo nano /etc/modules`

Προσθέτουμε τις 2 γραμμές στο τέλος

- `i2c-bcm2708`
- `i2c-dev`

- `sudo nano /etc/modprobe.d/raspi-blacklist.conf`

Προσθέτουμε τις 2 γραμμές στο τέλος

- `blacklist spi-bcm2708`
- `blacklist i2c-bcm2708`

- `sudo nano /boot/config.txt`

Προσθέτουμε τις 2 γραμμές στο τέλος

- `dtoverlay = i2c1=on`
- `dtoverlay = i2c_arm=on`

Για τον αισθητήρα θερμοκρασίας DS18B20 τρέχουμε στο Terminal τα εξής:

- `sudo nano /boot/config.txt`

Προσθέτουμε την τη γραμμή στο τέλος

- `dtoverlay = w1-gpio`

Έπειτα κάνουμε επανεκκίνηση το Raspberry Pi.



### 5.3. Κώδικας Εφαρμογής

Ο κώδικας της εφαρμογής έχει γραφτεί σε γλώσσα Python. Η Python είναι διερμηνευόμενη, γενικού σκοπού και υψηλού επιπέδου, γλώσσα προγραμματισμού. Ανήκει στις γλώσσες προστακτικού προγραμματισμού και υποστηρίζει τόσο το διαδικαστικό όσο και το αντικειμενοστραφές προγραμματιστικό. Δημιουργήθηκε από τον Ολλανδό Guido van Rossum στο ερευνητικό κέντρο Centrum Wiskunde & Informatica (CWI) το 1989 και κυκλοφόρησε για πρώτη φορά το 1991.

Ο κύριος στόχος της είναι η αναγνωσιμότητα του κώδικά της και η ευκολία χρήσης της. Το συντακτικό της επιτρέπει στους προγραμματιστές να εκφράσουν έννοιες σε λιγότερες γραμμές κώδικα από ότι θα ήταν δυνατόν σε γλώσσες όπως η C++ ή η Java. Διακρίνεται λόγω του ότι έχει πολλές βιβλιοθήκες που διευκολύνουν ιδιαίτερα αρκετές συνηθισμένες εργασίες και για την ταχύτητα εκμάθησής της. Μειονεκτεί στο ότι επειδή είναι διερμηνευόμενη είναι πιο αργή από τις μεταγλωττιζόμενες γλώσσες όπως η C και η C++. Για αυτό το λόγο δεν είναι κατάλληλη για γραφή λειτουργικών συστημάτων.

Οι διερμηνευτές της Python είναι διαθέσιμοι για εγκατάσταση σε πολλά λειτουργικά συστήματα, επιτρέποντας στην Python την εκτέλεση κώδικα σε ευρεία γκάμα συστημάτων. Η Python αναπτύσσεται ως ανοιχτό λογισμικό (open source) και η διαχείρισή της γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation. Ο κώδικας διανέμεται με την άδεια Python Software Foundation License η οποία είναι συμβατή με την GPL. Το όνομα της γλώσσας προέρχεται από την ομάδα των Άγγλων κωμικών Μόντυ Πάιθον και δεν έχει καμιά σχέση με το φίδι πύθωνα, παρότι το λογότυπό της παραπέμπει σε κάτι τέτοιο.



Εικόνα 47 – Logo Python.

Ο κώδικας που χρησιμοποιήσαμε για την εφαρμογή είναι ο εξής:

### pid.py

```
import smbus
import time
import os
import glob
import RPi.GPIO as GPIO

# Define some device parameters
I2C_ADDR = 0x3F # I2C device address
LCD_WIDTH = 16 # Maximum characters per line

# Define some device constants
LCD_CHR = 1 # Mode - Sending data
LCD_CMD = 0 # Mode - Sending command
LCD_LINE_1 = 0x80 # LCD RAM address for the 1st line
LCD_LINE_2 = 0xC0 # LCD RAM address for the 2nd line
LCD_BACKLIGHT = 0x08 # On

ENABLE = 0b00000100 # Enable bit

# Timing constants
E_PULSE = 0.0005
E_DELAY = 0.0005

#Open I2C interface
bus = smbus.SMBus(1) # Rev 2 Pi uses 1

#temp
os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')
base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'

#rotary
RoAPin = 17 # CLK Pin
RoBPin = 18 # DT Pin
BtnPin = 27 # Button Pin

#pin
GREEN_LED = 16
RED_LED = 6
BLUE_LED = 5
Fan = 19
Power = 20
ButtonDown = 22
ButtonUp = 26
BottonReset = 24

def setup():
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(RED_LED, GPIO.OUT)
    GPIO.setup(BLUE_LED, GPIO.OUT)
    GPIO.setup(GREEN_LED, GPIO.OUT)
    GPIO.setup(Fan, GPIO.OUT)
    GPIO.setup(Power, GPIO.OUT)
    GPIO.setup(RoAPin, GPIO.IN)
    GPIO.setup(RoBPin, GPIO.IN)
    GPIO.setup(BtnPin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
    GPIO.setup(ButtonDown, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
    GPIO.setup(ButtonUp, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
    GPIO.setup(BottonReset, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)

# Initialise display
def lcd_init():
    lcd_byte(0x33,LCD_CMD)
    lcd_byte(0x32,LCD_CMD)
    lcd_byte(0x06,LCD_CMD)
    lcd_byte(0x0C,LCD_CMD)
    lcd_byte(0x28,LCD_CMD)
```

## Ανάπτυξη PID ελεγκτή θερμοκρασίας με τη βοήθεια Raspberry Pi και Matlab

```
lcd_byte(0x01,LCD_CMD)
time.sleep(E_DELAY)

def lcd_byte(bits, mode):
    bits_high = mode | (bits & 0xF0) | LCD_BACKLIGHT
    bits_low = mode | ((bits<<4) & 0xF0) | LCD_BACKLIGHT
    # High bits
    bus.write_byte(I2C_ADDR, bits_high)
    lcd_toggle_enable(bits_high)
    # Low bits
    bus.write_byte(I2C_ADDR, bits_low)
    lcd_toggle_enable(bits_low)

def lcd_toggle_enable(bits):
    # Toggle enable
    time.sleep(E_DELAY)
    bus.write_byte(I2C_ADDR, (bits | ENABLE))
    time.sleep(E_PULSE)
    bus.write_byte(I2C_ADDR,(bits & ~ENABLE))
    time.sleep(E_DELAY)

def lcd_string(message,line):
    # Send string to display
    message = message.ljust(LCD_WIDTH," ")
    lcd_byte(line, LCD_CMD)
    for i in range(LCD_WIDTH):
        lcd_byte(ord(message[i]),LCD_CHR)

def read_temp_raw():
    f = open(device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

def read_temp_c():
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2)
        lines = read_temp_raw()
    equals_pos = lines[1].find('=')
    if equals_pos != -1:
        temp_string = lines[1][equals_pos+2:]
        temp_c = int(temp_string) / 1000.0 # TEMP_STRING IS THE SENSOR OUTPUT, MAKE SURE IT'S AN INTEGER TO
DO THE MATH
        return temp_c

def temp_to_string():
    temp_c = read_temp_c()
    temp_c = str(round(temp_c, 1)) # ROUND THE RESULT TO 1 PLACE AFTER THE DECIMAL, THEN CONVERT IT TO A
STRING
    return temp_c

lcd_init()

def rotaryDeal(globalCounter):
    globalCounter = str(round(globalCounter, 1))
    return globalCounter

def btnUp(channel):
    global globalCounter
    globalCounter = globalCounter + 0.1
    lsd()

def btnDown(channel):
    global globalCounter
    globalCounter = globalCounter - 0.1
    lsd()

def btnReset(channel):
    global globalCounter
    globalCounter = read_temp_c()
    lsd()

def lsd():
```

## Ανάπτυξη PID ελεγκτή θερμοκρασίας με τη βοήθεια Raspberry Pi και Matlab

```
global globalCounter
lcd_string("Temp: " + temp_to_string() + "C",LCD_LINE_1)
lcd_string("Set:" + rotaryDeal(globalCounter) + "C",LCD_LINE_2)
def loop():
    global globalCounter
    globalCounter = read_temp_c()
    GPIO.add_event_detect(ButtonDown, GPIO.RISING, callback=btnDown)
    GPIO.add_event_detect(ButtonUp, GPIO.RISING, callback=btnUp)
    GPIO.add_event_detect(ButtonReset, GPIO.RISING, callback=btnReset)
    while True:
        if GPIO.input(24) == GPIO.HIGH:
            GPIO.output(RED_LED, True)
            GPIO.output(BLUE_LED, True)
            GPIO.output(GREEN_LED, True)
            a = read_temp_c()
            if a > globalCounter: #hot start fan
                GPIO.output(RED_LED, True)
                GPIO.output(BLUE_LED, False)
                GPIO.output(GREEN_LED, False)
                GPIO.output(Fan, True)
                GPIO.output(Power, False)
            elif a < globalCounter: #coold start power
                GPIO.output(RED_LED, False)
                GPIO.output(BLUE_LED, True)
                GPIO.output(GREEN_LED, False)
                GPIO.output(Fan, False)
                GPIO.output(Power, True)
            elif a == globalCounter: #correct temperature
                GPIO.output(RED_LED, False)
                GPIO.output(BLUE_LED, False)
                GPIO.output(GREEN_LED, True)
                GPIO.output(Fan, False)
                GPIO.output(Power, False)

def destroy():
    GPIO.cleanup()          # Release resource

if __name__ == '__main__': # Program start from here
    setup()
    try:
        loop()
    except KeyboardInterrupt: # When 'Ctrl+C' is pressed, the child program destroy() will be executed.
        destroy()
```

Για να βάλουμε να τρέχει ο κώδικας του αρχείου μας (pid.py) κατευθείαν μόλις ανοίξει το Raspberry Pi τρέχουμε στο Terminal τα εξής:

- cd pid
- nano launcher.sh

Προσθέτουμε τις γραμμές

- #!/Bin/sh
- #launcher.sh
- #navigate to home directory, then to this directory, then execute python script, then back home
  
- cd/

## Ανάπτυξη PID ελεγκτή θερμοκρασίας με τη βοήθεια Raspberry Pi και Matlab

- `cd home/pi/pid`
- `sudo python pid.py`
- `cd/`

## 6. ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] A. P. Singh, —Speed Control of DC Motor using PID Controller Based on Matlab, vol. 4.
- [2] Σήματα και Συστήματα με Matlab, Παλαμίδης – Βελώνη – Σύγχρονη Εκδοτική - 2008.
- [3] Συστήματα Αυτομάτου Ελέγχου, Βελώνη Αναστασία – Εκδόσεις Τζιόλα
- [4] Gardner, J.W 2000 Μικροαισθητήρες – Αρχές και Εφαρμογές. Θεσσαλονίκη : Εκδόσεις Τζιόλα
- [5] Σήματα και Συστήματα για Τεχνολόγους, Φωτόπουλος – Βελώνη – Σύγχρονη Εκδοτική
- [6] Comparison of PID Controller Tuning Methods – Semantic
- [7] <https://www.raspberrypi.org/>
- [8] <https://www.raspberrypi-spy.co.uk/2015/05/using-an-i2c-enabled-lcd-screen-with-the-raspberry-pi/>
- [9] <https://computertechinfo.gr/ti-einai-to-raspberry-pi/>
- [10] <https://www.circuitbasics.com/raspberry-pi-ds18b20-temperature-sensor-tutorial/>
- [11] <https://en.wikipedia.org/wiki/MATLAB>
- [12] <https://uk.mathworks.com/>
- [13] <https://en.wikipedia.org/wiki/MathWorks>
- [14] <https://en.wikipedia.org/wiki/Microcontroller>
- [15] [https://en.wikipedia.org/wiki/PID\\_controller#Manual\\_tuning](https://en.wikipedia.org/wiki/PID_controller#Manual_tuning)
- [16] <https://www.sdcard.org/downloads/formatter/>
- [17] [https://uk.mathworks.com/help/matlab/matlab\\_external/support-package-installation.html](https://uk.mathworks.com/help/matlab/matlab_external/support-package-installation.html)
- [18] <https://pdfs.semanticscholar.org/116c/e07bcb202562606884c853fd1d19169a0b16.pdf>
- [19] <https://www.python.org/>
- [20] <https://el.wikipedia.org/wiki/Python>