



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Οδήγηση Bi-red μέσω διαδικτυακής εφαρμογής

Αγγέλης Α. Φώτης

Εισηγητής: Δρ Ιωάννης Έλληνας, Καθηγητής

ΑΘΗΝΑ
ΣΕΠΤΕΜΒΡΙΟΣ 2019

(κενή σελίδα)

Αφιερωμένο στην μητέρα μου Αστέρω

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Οδήγηση Bi-ped μέσω διαδικτυακής εφαρμογής

Αγγέλης Α. Φώτης
Α.Μ. 27278

Εισηγητής:
Δρ Ιωάννης Έλληνας, Καθηγητής

Εξεταστική Επιτροπή:
Αναστασία Βελώνη, Καθηγήτρια
Ιωάννης Αμοργίνος, Λέκτορας

Ημερομηνία εξέτασης: 09/2019

(κενή σελίδα)

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Αγγέλης Φώτης, του Αποστόλου, με αριθμό μητρώου 27278 φοιτητής του τμήματος ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ της σχολής ΜΗΧΑΝΙΚΩΝ, του ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ 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.»

(κενή σελίδα)

Ευχαριστίες

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε με την υποστήριξη του κ. Ιωάννη Έλληνα, τον οποίο θα ήθελα να ευχαριστήσω καθώς με κατεύθυνε με τον κατάλληλο τρόπο ώστε να καταφέρω να ολοκληρώσω την εργασία αυτή αρκετά χρόνια μετά την επιτυχή ολοκλήρωση του κύκλου μαθημάτων. Επίσης θα ήθελα να ευχαριστήσω την οικογένεια μου που με στήριξε οικονομικά και ψυχολογικά μέχρι και την ολοκλήρωση των σπουδών μου.

(κενή σελίδα)

Περίληψη

Η παρούσα πτυχιακή εργασία έρχεται σαν συνέχεια της πτυχιακής εργασίας των Αποστόλη Γαλανόπουλου και Αλέξανδρου Ντίτορα: Biped robot 9DOF controlled via android application (bluetooth and arduino). Στα πλαίσια της παρούσας εργασίας υλοποιήθηκαν νέα κυκλωματα και οι φυσικές τους διασυνδέσεις καθώς και το λογισμικό ώστε να οδηγούνται τα κινητά μέρη του biped μέσω μιας διαδικτυακής εφαρμογής, πιο συγκεκριμένα μιας ιστοσελίδας χρησιμοποιώντας υλικό της πλατφόρμας Arduino.

Abstract

This thesis comes as a follow-up to the thesis of Apostolis Galanopoulos and Alexandros Ditoras: Biped robot 9DOF controlled via android application (bluetooth and arduino). In the present work created: new circuits and their physical interfaces as well as the software needed to drive biped robot's mobile components through a network application, in particular a web site using hardware and the Arduino platform.

Keywords

arduino, servo, biped, web, http

Περιεχόμενα

1. Εισαγωγή	14
1.1 Προηγούμενη πτυχιακή εργασία	14
1.2 Περιγραφή αντικειμένου της πτυχιακής εργασίας	14
1.3 Δομή πτυχιακής εργασίας	14
Μέρος Πρώτο	15
2. Μικροελεγκτές και Arduino	15
2.1 Μικροελεγκτές	15
2.2 Arduino	16
2.2.1 Υλικό	16
2.2.1.1 Boards	17
2.2.1.2 Shields	17
2.2.2 Λογισμικό	18
2.2.2.1 Γλώσσα προγραμματισμού	18
2.2.2.1.1 Λειτουργίες	19
2.2.2.1.2 Τιμές	20
2.2.2.1.3 Δομή	20
2.2.2.2 Arduino IDE	21
2.2.2.2.1 Εγκατάσταση και παραμετροποίηση σε Mac OS X	22
2.3 Υλικό Arduino που χρησιμοποιήθηκε	23
2.3.1 Arduino Uno	23
2.3.1.2 Τεχνικά χαρακτηριστικά	24
2.3.1.3 Pins	24
2.3.1.3.1 Pins γενικής χρήσης	25
2.3.1.3.2 Pins ειδικών χρήσεων	25
2.3.1.4 Επικοινωνία	26
2.3.2 Ethernet Shield	27
2.3.2.1 Ιδιότητες και τεχνικά χαρακτηριστικά	27
2.3.2.2 Επεξήγηση πλακέτας και διαθέσιμων pins	28
3. Δίποδα (biped) ρομπότ	29
3.1 Κινητήρες τύπου servo	29
3.1.1 Πλεονεκτήματα και μειονεκτήματα έναντι άλλων τύπων κινητήρων	29
3.1.1.1 Πλεονεκτήματα	29
3.1.1.2 Μειονεκτήματα	29
3.1.2 Servo Tower Pro MG996R	30
3.1.2.1 Τεχνικά χαρακτηριστικά	30
3.3 Biped robots	31
3.3.1 Ρομπότ	31
3.3.2 Biped robot	32

4. Διαδικτυακές εφαρμογές	33
4.1 HTML	33
4.2 CSS	34
4.3 Διάθεση διαδικτυακής εφαρμογής	37
4.3.1 HTTP πρωτόκολλο	37
4.3.2 Web Servers	38
Μέρος Δεύτερο	39
5. Περιγραφή υλοποίησης και λειτουργίας	39
5.1 Βασική εγκατάσταση, παραμετροποίηση και δοκιμαστικό πρόγραμμα Arduino	39
5.1.1 Συνδεσμολογία	39
5.1.2 Υλοποίηση σε κώδικα	40
5.1.3 Συμπεράσματα/Σχόλια	41
5.2 Σύνδεση και χειρισμός ενός servo κινητήρα από το Arduino	42
5.2.1 Συνδεσμολογία	42
5.2.2 Υλοποίηση σε κώδικα	42
5.2.3 Συμπεράσματα/Σχόλια	43
5.3 Σύνδεση του Ethernet Shield και χειρισμός ενός servo μέσα από διαδικτυακή εφαρμογή	44
5.3.1 Συνδεσμολογία	44
5.3.2 Υλοποίηση σε κώδικα	45
5.3.3 Συμπεράσματα/Σχόλια	46
5.4 Προσθήκη όλων των servo του biped robot και χειρισμός τους από την διαδικτυακή εφαρμογή	48
5.4.1 Συνδεσμολογία	48
5.4.2 Υλοποίηση σε κώδικα	49
5.4.3 Συμπεράσματα/Σχόλια	50
6. Συμπεράσματα και προοπτικές	52
6.1 Συμπεράσματα/Επίλογος πτυχιακής εργασίας	52
6.2 Επόμενα βήματα	52
Παράρτημα	53
A. Fritzing	53
B. Gitlab	53
Γ. Amazon S3	54
Δ. Bootstrap framework	54
E. AngularJS	54
Z. Κώδικας Arduino και HTML σελίδα για τον χειρισμό όλων των servo	55
Βιβλιογραφικές αναφορές και σύνδεσμοι	60

Κατάλογος Πινάκων

[Πίνακας 2.1 Επίσημες πλακέτες Arduino](#)

[Πίνακας 2.2 Ορισμένα Arduino Shields](#)

[Πίνακας 2.3 Τεχνικά χαρακτηριστικά Arduino Uno rev3](#)

[Πίνακας 2.4 Τεχνικά χαρακτηριστικά Ethernet Shield v1.1](#)

[Πίνακας 3.1 Τεχνικά χαρακτηριστικά TowerPro MG996R](#)

[Πίνακας 3.2 Καλωδίωση Tower Pro MG996R](#)

[Πίνακας 4.1 Μερικά html tags](#)

[Πίνακας 4.2 Μερικές ιδιότητες css](#)

[Πίνακας 4.3 HTTP μέθοδοι](#)

Κατάλογος Σχημάτων και εικόνων

[Εικόνα 2.1 Σχήμα παραδείγματος μικροελεγκτή \(PIC16F\)](#)

[Εικόνα 2.2 Διαθέσιμες λειτουργίες για την διαχείριση Arduino board](#)

[Εικόνα 2.3 Τύποι δεδομένων, μεταβλητές και σταθερές για χρήση σε προγράμματα Arduino.](#)

[Εικόνα 2.4 Στοιχεία δομής Arduino προγράμματος](#)

[Εικόνα 2.5 Βασικό παράθυρο Arduino IDE](#)

[Εικόνα 2.6 Εκτελέσιμο Arduino Software \(desktop IDE\)](#)

[Εικόνα 2.7 Arduino Uno rev3](#)

[Εικόνα 2.8 Pins Arduino Uno rev3](#)

[Εικόνα 2.9 Arduino Ethernet Shield \(v1.1 Wiznet by Seed\)](#)

[Εικόνα 2.10 Pins Ethernet Shield \(v1.1 by Seed\)](#)

[Εικόνα 3.1 Παράδειγμα servo](#)

[Εικόνα 3.2 Servo TowerPro MG996R](#)

[Εικόνα 3.3 Ρομπότ Asimo \(2000\) κατασκευασμένο από την Honda](#)

[Εικόνα 3.4 Biped robot \(9DOF\) που χρησιμοποιήθηκε στην εργασία](#)

[Εικόνα 4.1 Hello World html page](#)

[Εικόνα 4.2 hello world html page with css](#)

[Εικόνα 4.3 HTTP επικοινωνία μεταξύ εξυπηρετητή και πελάτη](#)

[Εικόνα 5.1 Σύνδεση Arduino με υπολογιστή μέσω USB](#)

[Εικόνα 5.2 Παραμετροποίηση πλακέτας στο Arduino IDE](#)

[Εικόνα 5.3 Επιλογή θύρας στο Arduino IDE](#)

[Εικόνα 5.4 Hello World arduino program](#)

[Εικόνα 5.5 Άνοιγμα της εφαρμογής Serial Monitor](#)

[Εικόνα 5.6 Serial Monitor μετά την μεταφόρτωση του προγράμματος](#)

[Εικόνα 5.7 Έλεγχος ενός servo του biped από τον Arduino](#)

[Εικόνα 5.8 Κώδικας κίνησης ενός σέρβο Arduino IDE](#)

[Εικόνα 5.9 Servo χειρισμός μέσω διαδικτυακής εφαρμογής](#)

[Εικόνα 5.10 Εφαρμογή διαδικτύου \(σελίδα html\) για τον χειρισμό ενός από τα servo του biped robot](#)

[Εικόνα 5.11 Σύνδεση και των 8 servo στον Arduino Uno μέσω του Ethernet Shield](#)

[Εικόνα 5.12 Οθόνη διαδικτυακής εφαρμογής για την διαχείριση των οκτώ servo του biped robot](#)

Συντομογραφίες

9DOF nine degree-of-freedom
ROM read-only memory
EPROM erasable programmable read-only memory
RAM random access memory
LED light emitting diode
IDE integrated development environment
GPS Global Positioning System
USB Universal Serial Bus
LCD Liquid Crystal Display
DIY Do It Yourself
FTDI Future Technology Devices International
κτλ και τα λοιπά
UART universal asynchronous receiver-transmitter
TTL Time To Live
TCP Transmission Control Protocol
UDP User Datagram Protocol
PCB Printed Circuit Board
IC integrated circuit
URI Universal Resource Identifier
ICSP in-circuit serial programming
API application program interface

1. Εισαγωγή

1.1 Προηγούμενη πτυχιακή εργασία

Αφετηρία για την εργασία που πραγματοποιήσα ήταν η πτυχιακή εργασία των Αποστόλη Γαλανόπουλου και Αλέξανδρου Ντίτορα: Biped robot 9DOF controlled via android application (bluetooth and arduino). Σε αυτή την εργασία οι σπουδαστές κατασκεύασαν ένα biped (δίποδο) ρομπότ το οποίο ελεγχόταν από μια εφαρμογή android μέσω bluetooth. Για την υλοποίηση της εργασίας μου χρησιμοποίησα το δίποδο ρομπότ που κατασκευάστηκε στα πλαίσια της προαναφερθείσας εργασίας. Τα κυκλώματα της προηγούμενης εργασίας αποσυνδεθηκαν ώστε να μπορούν να χρησιμοποιηθούν ως έχουν στο μέλλον.

1.2 Περιγραφή αντικειμένου της πτυχιακής εργασίας

Αντικείμενο της παρούσας πτυχιακής άσκησης είναι να ελέγξουμε ένα δίποδο ρομπότ μέσα από μια διαδικτυακή εφαρμογή και πιο συγκεκριμένα τα κινούμενα μέρη του: χέρια, πόδια, κτλ. Η εφαρμογή αυτή θα φιλοξενηθεί στο arduino και θα είναι διαθέσιμη μέσα από το τοπικό δίκτυο που θα είναι συνδεδεμένη η πλακέτα του Arduino.

Στόχοι της εργασίας είναι:

- Η κατανόηση του τρόπου λειτουργίας και προγραμματισμού της πλατφόρμας Arduino και πώς μπορεί αυτή να χρησιμοποιηθεί για να λύσει προβλήματα του φυσικού κόσμου.
- Η διερεύνηση διαδικασιών και τεχνολογιών σχετικά με τους κινητήρες servo και τα δίποδα ρομπωτικά συστήματα.
- Η διερεύνηση της δημιουργίας διαδικτυακών εφαρμογών και η χρήση σύγχρονων εργαλείων για τον σκοπό αυτό.
- Η υλοποίηση ενός παραδείγματος ελέγχου δίποδου ρομπωτικού συστήματος με χρήση servo κινητήρων μέσω διαδικτυακής εφαρμογής χρησιμοποιώντας την πλατφόρμα Arduino και το υλικό που παρέχει.

1.3 Δομή πτυχιακής εργασίας

Η πτυχιακή εργασία χωρίστηκε σε δύο μέρη. Στο πρώτο μέρος θα αναλυθούν και εξηγηθούν σε θεωρητικό επίπεδο όλες οι τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν ώστε ο αναγνώστης να προετοιμαστεί για το δεύτερο μέρος. Σε αυτό θα αναλυθεί βήμα-βήμα η υλοποίηση και το πρακτικό μέρος της εργασίας.

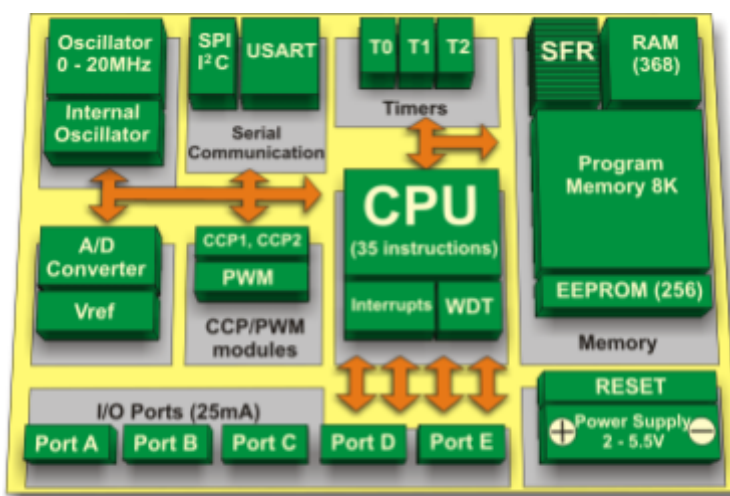
Μέρος Πρώτο

2. Μικροελεγκτές και Arduino

2.1 Μικροελεγκτές

Μικροελεγκτής (αγγλικά microcontroller), είναι ένα ολοκληρωμένο κύκλωμα το οποίο μπορεί να προγραμματιστεί. Το ολοκληρωμένο κύκλωμα αυτό περιέχει επεξεργαστή και άλλα περιφερειακά συστήματα (όπως μνήμη ROM/EPROM και RAM, κυκλώματα ελέγχου περιφερειακών συσκευών, θύρες εισόδου/εξόδου, κ.ά). Ο μικροελεγκτής λοιπόν σε σχέση με έναν μικροεπεξεργαστή μπορεί να λειτουργήσει με ελάχιστα εξωτερικά συστήματα καθώς έχει αρκετά ενσωματωμένα υποσυστήματα.

Οι μικροελεγκτές χρησιμοποιούνται σε όλα τα ενσωματωμένα συστήματα (embedded systems) ελέγχου χαμηλού και μεσαίου κόστους, δηλαδή σε συστήματα αυτοματισμού, σε ηλεκτρικές συσκευές, σε ηλεκτρονικές και ηλεκτρικές συσκευές, στην ρομποτική, κ.ά. Υπάρχουν αρκετές εταιρείες που κατασκευάζουν μικροελεγκτές σε όλο τον κόσμο, όπως οι Microchip, Atmel, Intel, Texas Instruments. Οι μικροελεγκτές προγραμματίζονται σε γλώσσα προγραμματισμού χαμηλού επιπέδου που βρίσκεται πιο κοντά στο υλικό (γλώσσα μηχανής, assembly), μπορεί όμως να χρησιμοποιηθεί και γλώσσα υψηλού επιπέδου και με τον κατάλληλο compiler να μετατραπεί σε κατάλληλη γλώσσα μηχανής για τον μικροελεγκτή.



Εικόνα 2.1 Σχήμα παραδείγματος μικροελεγκτή (PIC16F)

2.2 Arduino

Οι Massimo Banzι και David Cueartielles το 2005 ξεκίνησαν ένα σχέδιο για να φτιαχτεί μία συσκευή για τον έλεγχο προγραμμάτων διαδραστικών σχεδίων από μαθητές, η οποία θα ήταν πιο φθηνή από άλλα πρωτότυπα συστήματα διαθέσιμα εκείνη την περίοδο. Έτσι ονόμασαν το σχέδιο τους από τον Αρντουίνο της Ιβρέας και ξεκίνησαν να παράγουν πλακέτες σε ένα μικρό εργοστάσιο στην Ιβρέα, κωμόπολη της επαρχίας Τορίνο στην περιοχή Πεδεμόντιο της βορειοδυτικής Ιταλίας - την ίδια περιοχή στην οποία στεγαζόταν η εταιρεία υπολογιστών Olivetti.

Το Arduino είναι μία ανοικτού κώδικα¹ πλατφόρμα βασισμένη σε υλικό και λογισμικό που είναι εύκολο στη χρήση. Η πλατφόρμα περιλαμβάνει μια πλακέτα η οποία έχει προσαρμοσμένα έναν μικροελεγκτή και διαθέσιμες εισόδους και εξόδους. Ο προγραμματισμός αυτής της πλακέτας μπορεί να γίνει με την γλώσσα προγραμματισμού του Arduino που είναι βασισμένη στην Wiring (παρόμοια με τη γλώσσα προγραμματισμού C++² και ένα σύνολο από βιβλιοθήκες, υλοποιημένες επίσης στην C++). Το Arduino μπορεί να χρησιμοποιηθεί για την ανάπτυξη ανεξάρτητων διαδραστικών αντικειμένων αλλά και να συνδεθεί με υπολογιστή μέσω προγραμμάτων σε Processing, Max/MSP, Pure Data, SuperCollider.

Με την πλατφόρμα αυτή μπορούν να υλοποιηθούν πολλά διαφορετικά έργα που έχουν σαν αντικείμενο την ανάγνωση δεδομένων από κάποια είσοδο, την επεξεργασία τους και την έξοδο κάποιου αποτελέσματος. Μερικά παραδείγματα εισόδου είναι η ανίχνευση φωτός σε έναν αισθητήρα, το πάτημα ενός κουμπιού, ένα μήνυμα στο Twitter και μερικά παραδείγματα εξόδου είναι η ενεργοποίηση ενός κινητήρα, το άναμα ενός LED ή η ανάρτηση κάποιου κειμένου στο διαδίκτυο.

Για την υλοποίηση λοιπόν διαφόρων έργων χρειάζεται να συνταχθεί το κατάλληλο πρόγραμμα στην γλώσσα προγραμματισμού του Arduino και να εγκατασταθεί στον μικροελεγκτή της πλακέτας μέσω του προγράμματος Arduino IDE (βασισμένο στο Processing³).

2.2.1 Υλικό

Το υλικό της πλατφόρμας Arduino αποτελείται από πλακέτες (boards) και τυπωμένα boards επεκτάσεων κυκλωμάτων (shields).

¹ Το λογισμικό του οποίου ο πηγαίος κώδικας διατίθεται σε τρίτο για να τον εξετασει

² Η C++ (διαβάζεται: σι πλας πλας[1] [si: pləs pləs]) είναι μία γενικού σκοπού γλώσσα προγραμματισμού Η/Υ.

³ Processing είναι γλώσσα προγραμματισμού ανοικτού κώδικα και παράλληλα προγραμματιστικό περιβάλλον για ανθρώπους που θέλουν να προγραμματίσουν εικόνες, animation και ήχο

2.2.1.1 Boards

Μία πλακέτα (board) Arduino αποτελείται από ένα μικροελεγκτή Atmel AVR (ATmega328 και ATmega168 στις νεότερες εκδόσεις, ATmega8 στις παλαιότερες) και συμπληρωματικά εξαρτήματα για την διευκόλυνση του χρήστη στον προγραμματισμό και την ενσωμάτωσή του σε άλλα κυκλώματα. Όλες οι πλακέτες περιλαμβάνουν ένα γραμμικό ρυθμιστή τάσης 5V και έναν κρυσταλλικό ταλαντωτή 16MHz (ή κεραμικό αντηχητή σε κάποιες παραλλαγές). Ο μικροελεγκτής είναι από κατασκευής προγραμματισμένος με ένα bootloader, έτσι ώστε να μην χρειάζεται εξωτερικός προγραμματιστής.

Μέχρι στιγμής έχουν χρησιμοποιηθεί εμπορικά δεκαέξι διαφορετικές επίσημες πλακέτες Arduino. Τα σημερινά Arduino boards προγραμματίζονται μέσω USB, αυτό καθίσταται δυνατό μέσω της εφαρμογής προσαρμογών chip USB-to-Serial όπως το FTDI FT232.




Όνομα Πλακέτας	Περιγραφή
Serial Arduino	προγραμματισμένο με μία σειριακή DE-9 σύνδεση χρησιμοποιώντας τεχνολογία ATmega8.
Arduino Extreme	με ένα USB interface για προγραμματισμό χρησιμοποιώντας τεχνολογία ATmega8.
Arduino Mini	μία έκδοση μινιατούρας του Arduino χρησιμοποιώντας τεχνολογία surface-mounted ATmega168.
Arduino Nano	ένα ακόμα πιο μικρό, USB τροφοδοτούμενη εκδοχή του Arduino χρησιμοποιώντας τεχνολογία surface-mounted ATmega168 (ATmega328 για την νεότερη έκδοση).
LilyPad Arduino	ένα μινιμαλιστικό σχέδιο για εφαρμογές ένδυσης και E-textiles χρησιμοποιώντας τεχνολογία surface-mounted AT-mega328.
Arduino NG	με ένα USB interface για προγραμματισμό και χρησιμοποιώντας τεχνολογία ATmega8.
Arduino NG plus	με ένα USB interface για προγραμματισμό και χρησιμοποιώντας τεχνολογία ATmega168.
Arduino Bluetooth	με Bluetooth interface για προγραμματισμό χρησιμοποιώντας τεχνολογία ATmega168.
Arduino Diecimila	με ένα USB interface και χρησιμοποιεί τεχνολογία ATmega168 σε ένα DIP28 πακέτο.
Arduino Duemilanove	χρησιμοποιεί τεχνολογία ATmega168 (ATmega328 για την καινούργια έκδοση) και τροφοδοτείται μέσω ενέργειας USB/DC, αυτόματα εναλλασσόμενη.
Arduino Mega	χρησιμοποιώντας τεχνολογία surface-mounted ATmega1280 για περαιτέρω I/O και μνήμη.
Arduino Uno	χρησιμοποιώντας την ίδια τεχνολογία ATmega328 όπως το τελευταίο μοντέλο Duemilanove, αλλά ενώ το Duemilanove χρησιμοποιεί ένα FTDI chipset για το USB, το Uno χρησιμοποιεί τεχνολογία ATmega8U2 προγραμματισμένο ως σειριακός μετατροπέας.
Arduino Mega 2560	χρησιμοποιεί τεχνολογία surface-mounted ATmega2560 φέρνοντας την ολική μνήμη στα 256kB. Επίσης ενσωματώνει τη νέα τεχνολογία ATmega8U2 (ATmega16U2 σε αναθεώρηση τύπου 3) USB chipset.
Arduino Leonardo	με ένα ATmega32U4 chip που εξαλείφει την ανάγκη για συνδεσιμότητα μέσω USB και μπορεί να χρησιμοποιηθεί ως ψηφιακό πληκτρολόγιο ή ποντίκι. Κυκλοφόρησε στο Maker Faire Bay Area το 2012.
Arduino Esplora	με εμφάνιση που παραπέμπει σε χειριστήριο κονσόλας βιντεοπαιχνιδιών με joystick και ενσωματωμένους αισθητήρες για ήχο, φως, θερμοκρασία και επιτάχυνση.
Arduino Due	είναι ένα μικροχειριστήριο board βασισμένο στην τεχνολογία Atmel SAM3X8E ARM Cortex-M3 CPU. Είναι το πρώτο board της Arduino βασισμένη σε επεξεργαστή 32-bit ARM microcontroller

Πίνακας 2.1 Επίσημες πλακέτες Arduino

2.2.1.2 Shields

Τα Arduino και τα Arduino συμβατά boards χρησιμοποιούν την τεχνολογία των shields, τυπωμένων boards επεκτάσεων κυκλωμάτων που συνδέονται στα

κανονικά παρεχόμενα Arduino pin-headers. Τα shields μπορούν να παρέχουν έλεγχο στα motors, GPS, Ethernet, LCD εικόνας ή breadboarding (προτυποποίησης). Ένας αριθμός από shields μπορεί επίσης να γίνει και DIY.

Shield Name	Image	Usage
Arduino WiFi Shield		Το Arduino WiFi Shield συνδέει το Arduino στο διαδίκτυο ασύρματα.
Arduino Ethernet Shield		Το Arduino Ethernet Shield συνδέει το Arduino στο διαδίκτυο με ένα RJ45 καλώδιο.
Arduino Motor Shield		Το Arduino Motor Shield επιτρέπει την οδήγηση DC κινητήρων.

Πίνακας 2.2 Ορισμένα Arduino shields

2.2.2 Λογισμικό

2.2.2.1 Γλώσσα προγραμματισμού

Η γλώσσα προγραμματισμού που χρησιμοποιείται για τα Arduino προγράμματα βασίζεται στη γλώσσα Wiring, μια παραλλαγή της C/C++. Καθώς η γλώσσα αυτή προέρχεται από τη C/C++ οι εντολές, συναρτήσεις, μεταβλητές, σταθερές, τύποι δεδομένων, τελεστές και σύνταξη είναι ίδιες με την προσθήκη κάποιων ακόμα για την διαχείριση υλικού του Arduino. Ως compiler χρησιμοποιείται π AVR gcc και ως βασική βιβλιοθήκη της C χρησιμοποιείται η AVR libc.

Η γλώσσα προγραμματισμού του Arduino συμπεριλαμβάνει τρία βασικά μέρη: λειτουργίες, τιμές (μεταβλητές και σταθερές) και τη δομή.

2.2.2.1.1 Λειτουργίες

Υπάρχουν διαθέσιμες πάρα πολλές λειτουργίες (functions) για τον έλεγχο του Arduino board. Ακολουθεί μια λίστα από την ιστοσελίδα τεκμηρίωσης του Arduino⁴.

Digital I/O	Math	Random Numbers
digitalRead()	abs()	random()
digitalWrite()	constrain()	randomSeed()
pinMode()	map()	
	max()	Bits and Bytes
Analog I/O	min()	bit()
analogRead()	pow()	bitClear()
analogReference()	pow()	bitRead()
analogWrite()	sqrt()	bitSet()
		bitWrite()
Zero, Due & MKR Family	Trigonometry	highByte()
analogReadResolution()	cos()	lowByte()
analogWriteResolution()	sin()	
	tan()	External Interrupts
Advanced I/O	Characters	attachInterrupt()
noTone()	isAlpha()	detachInterrupt()
pulseIn()	isAlphaNumeric()	
pulseInLong()	isAscii()	Interrupts
shiftIn()	isControl()	interrupts()
shiftOut()	isDigit()	noInterrupts()
tone()	isGraph()	
	isHexadecimalDigit()	Communication
Time	isLowerCase()	Serial
delay()	isPrintable()	Stream
delayMicroseconds()	isPunct()	
micros()	isSpace()	USB
millis()	isUpperCase()	Keyboard
	isWhitespace()	Mouse

Εικόνα 2.2 Διαθέσιμες λειτουργίες για την διαχείριση Arduino board

Ένα απλό πρόγραμμα θα πρέπει να έχει οπωσδήποτε δύο λειτουργίες ορισμένες ώστε να εκτελεστεί:

- την `setup()`, μία συνάρτηση που τρέχει μία φορά στην αρχή του προγράμματος η οποία αρχικοποιεί τις ρυθμίσεις και
- την `loop()`, μία συνάρτηση που καλείται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί

```
// Ενσωματώσεις βιβλιοθηκών, δηλώσεις μεταβλητών...
void setup()
{
  // Λειτουργίες που ισχύουν για όλο το πρόγραμμα...
}
void loop()
{
  // Λειτουργίες που επαναλαμβάνονται κατά την λειτουργία του προγράμματος...
}
// Υπόλοιπες λειτουργίες...
```

⁴ <https://www.arduino.cc/reference/en/>

2.2.2.1.2 Τιμές

Οι τύποι δεδομένων, οι μεταβλητές και οι σταθερές που μπορούν να χρησιμοποιηθούν σε ένα πρόγραμμα Arduino, όπως αναφέραμε ήδη, είναι ίδιες με τις C/C++ με την προσθήκη ορισμένων ακόμα σχετικών με την διαχείριση του υλικού του Arduino. Ακολουθεί μια λίστα από την ιστοσελίδα τεκμηρίωσης του Arduino⁵

Constants	Conversion	float	Variable Scope & Qualifiers
Floating Point Constants	(unsigned int)	int	const
Integer Constants	(unsigned long)	long	scope
HIGH LOW	byte()	short	static
INPUT OUTPUT INPUT_PULLUP	char()	size_t	volatile
LED_BUILTIN	float()	string	
true false	int()	unsigned char	Utilities
	long()	unsigned int	PROGMEM
	word()	unsigned long	sizeof()
		void	
	Data Types	word	
	String()		
	array		
	bool		
	boolean		
	byte		
	char		
	double		

Εικόνα 2.3 Τύποι δεδομένων, μεταβλητές και σταθερές για χρήση σε προγράμματα Arduino.

2.2.2.1.3 Δομή

Ένα πρόγραμμα Arduino ονομάζεται sketch. Αυτό περιέχει τα κομμάτια κώδικα που θα εκτελεστούν από την πλακέτα Arduino. Όπως αναφέρθηκε παραπάνω πρέπει να συμπεριλαμβάνονται στην αρχή οι δηλώσεις μεταβλητών και η συμπερίληψη βιβλιοθηκών, οπωσδήποτε οι ειδικές λειτουργίες setup() και loop(). Υπάρχουν επίσης συντακτικοί κανόνες που πρέπει να ακολουθούνται, τελεστές σύγκρισης, αριθμητικοί τελεστές, λογικοί τελεστές, δομές ελέγχου, δομές ελέγχου και επανάληψης, και άλλα. Ακολουθεί μια λίστα από την ιστοσελίδα τεκμηρίωσης του Arduino⁶.

⁵ <https://www.arduino.cc/reference/en/>

⁶ <https://www.arduino.cc/reference/en/>

Sketch	Arithmetic Operators	Pointer Access Operators
loop()	% (remainder)	& (reference operator)
setup()	* (multiplication)	* (dereference operator)
	+ (addition)	
Control Structure	- (subtraction)	Bitwise Operators
break	/ (division)	&& (bitwise and)
continue	= (assignment operator)	<< (bitshift left)
do...while		>> (bitshift right)
else	Comparison Operators	^ (bitwise xor)
for	!= (not equal to)	(bitwise or)
goto	< (less than)	~ (bitwise not)
if	<= (less than or equal to)	
return	== (equal to)	Compound Operators
switch...case	> (greater than)	%= (compound remainder)
while	>= (greater than or equal to)	&= (compound bitwise and)
		*= (compound multiplication)
Further Syntax	Boolean Operators	++ (increment)
#define (define)	! (logical not)	+= (compound addition)
#include (include)	&& (logical and)	-- (decrement)
/* */ (block comment)	(logical or)	-= (compound subtraction)
// (single line comment)		/= (compound division)
;(semicolon)		^= (compound bitwise xor)
{ } (curly braces)		= (compound bitwise or)

Εικόνα 2.4 Στοιχεία δομής Arduino προγράμματος

2.2.2.2 Arduino IDE

Το ενσωματωμένο περιβάλλον ανάπτυξης του Arduino (IDE) είναι διαθέσιμο είτε κατευθείαν για χρήση μέσω διαδικτύου (online IDE⁷) είτε σαν εκτελέσιμο που μπορεί να εκτελεστεί τοπικά (desktop IDE⁸).

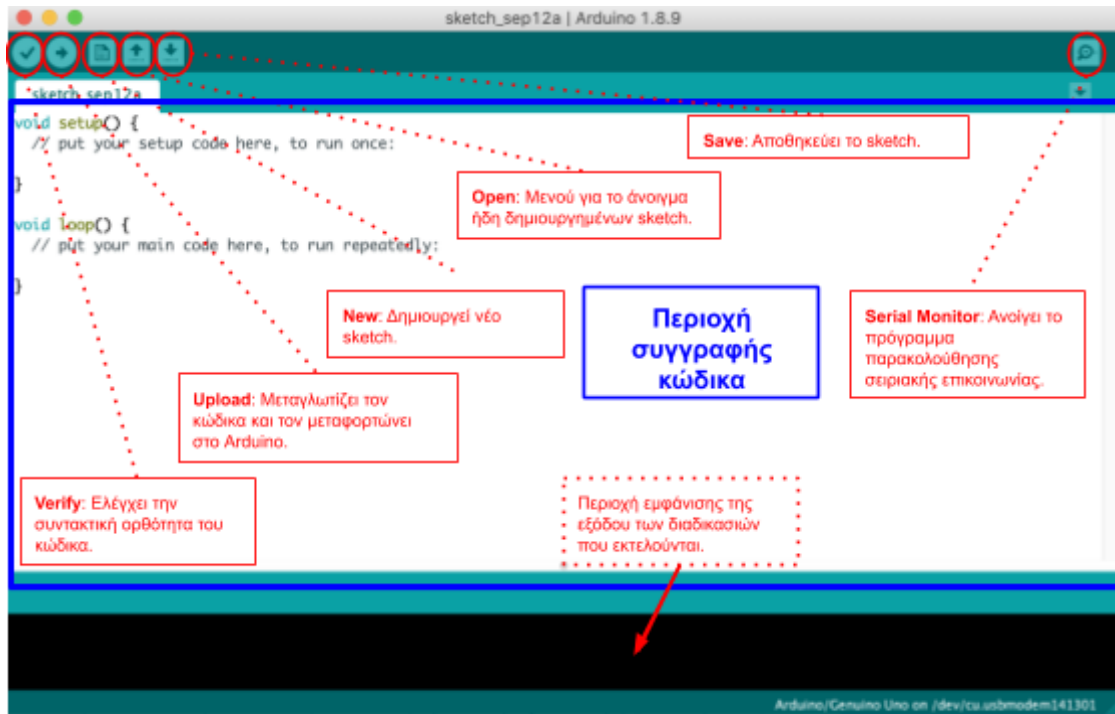
Για την παρούσα πτυχιακή εργασία χρησιμοποιήθηκε το desktop IDE που είναι μια εφαρμογή γραμμένη στη γλώσσα προγραμματισμού Java και χρησιμοποιείται για την εγγραφή, μεταγλώττιση και τη μεταφόρτωση προγραμμάτων σε πλακέτες Arduino. Το Arduino desktop IDE υπάρχει σε εκδόσεις για Windows, Mac και Linux και υπάρχει διαθέσιμο εντελώς δωρεάν στην επίσημη ιστοσελίδα του Arduino⁹.

Η εφαρμογή αυτή περιέχει ένα πρακτικό περιβάλλον για την συγγραφή των προγραμμάτων, με συντακτική χρωματική σημαση, μερικές έτοιμες βιβλιοθήκες, τον compiler για την μεταγλώττιση των sketch, ένα πρόγραμμα παρακολούθησης της σειριακής επικοινωνίας (serial monitor) και την απαραίτητη υλοποίηση για την μεταφόρτωση του συγγραφέντος προγράμματος στην πλακέτα του Arduino.

⁷ <https://create.arduino.cc/editor>

⁸ <https://www.arduino.cc/en/Main/Software#download>

⁹ <http://arduino.cc/en/Main/Software>

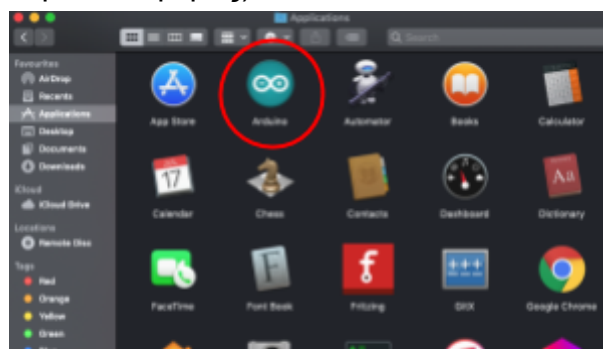


Εικόνα 2.5 Βασικό παράθυρο Arduino IDE

2.2.2.2.1 Εγκατάσταση και παραμετροποίηση σε Mac OS X

Το Arduino Software (IDE) που χρησιμοποιήθηκε για την υλοποίηση της παρούσας πτυχιακής εργασίας εγκαταστάθηκε σε τερματικό με λειτουργικό σύστημα Mac OS X. Τα βήματα που ακολουθήθηκαν ήταν:

- Λήψη της τελευταίας έκδοσης για το λειτουργικό Mac OS X¹⁰ και αποθήκευση στο δίσκο.
- Αποσυμπίεση του συμπιεσμένου καταλόγου.
- Αντιγραφή της εφαρμογής στον φάκελο Applications.
- Εκτέλεση της εφαρμογής.
- Παραμετροποίηση για την πλακέτα Arduino που χρησιμοποιήθηκε. (αναλυτικά τα βήματα στο πρακτικό μέρος)



Εικόνα 2.6 Εκτέλεση Arduino Software (desktop IDE)

¹⁰ https://www.arduino.cc/download_handler.php?f=/arduino-1.8.10-macosx.zip

2.3 Υλικό Arduino που χρησιμοποιήθηκε

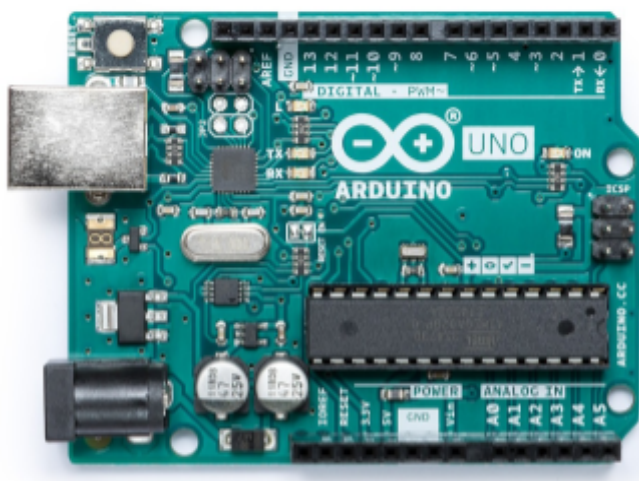
Για την υλοποίηση των απαραίτητων κυκλωμάτων για την πτυχιακή εργασία αυτή χρησιμοποιήθηκαν το Arduino Uno board για την οδήγηση των servo motors και το Ethernet Shield για την διασύνδεση στο δίκτυο και την φιλοξενία της ιστοσελίδας για την οδήγηση των servo των κινητών μερών του biped.

2.3.1 Arduino Uno

Το Arduino Uno είναι ένας μικροελεγκτής ανοιχτού κώδικα που βασίζεται στον μικροελεγκτή Microchip ATmega328P και αναπτύχθηκε από το Arduino. Η πλακέτα είναι εξοπλισμένη με σετ ψηφιακών και αναλογικών εισόδων / εξόδων (I / O) ακίδων που μπορούν να διασυνδεθούν με διάφορες πλακέτες επέκτασης (shields) και άλλα κυκλώματα. Ο πίνακας έχει 14 ψηφιακές ακίδες, 6 αναλογικές ακίδες και προγραμματίζεται με το IDE του Arduino μέσω καλωδίου USB τύπου B . Μπορεί να τροφοδοτείται από το καλώδιο USB ή από εξωτερικό 9-volt μπαταρία , αν και δέχεται τάσεις μεταξύ 7 και 20 βολτ.

Η λέξη "uno" σημαίνει "ένα" στα ιταλικά και επιλέχθηκε για να σηματοδοτήσει την αρχική έκδοση του λογισμικού Arduino. Το ATmega328 στο board έρχεται προ-προγραμματισμένο με ένα bootloader που επιτρέπει την αποστολή νέου κώδικα σε αυτό χωρίς τη χρήση ενός εξωτερικού προγραμματιστή υλικού.

Ενώ το Uno επικοινωνεί χρησιμοποιώντας το αρχικό πρωτόκολλο STK500, διαφέρει από όλες τις προηγούμενες πλακέτες στο ότι δεν χρησιμοποιεί το chip driver FTDI USB-to-serial. Αντίθετα, χρησιμοποιεί το Atmega16U2 (Atmega8U2 μέχρι την έκδοση R2) προγραμματισμένο ως μετατροπέας USB σε σειριακό .



Εικόνα 2.7 Arduino Uno rev3

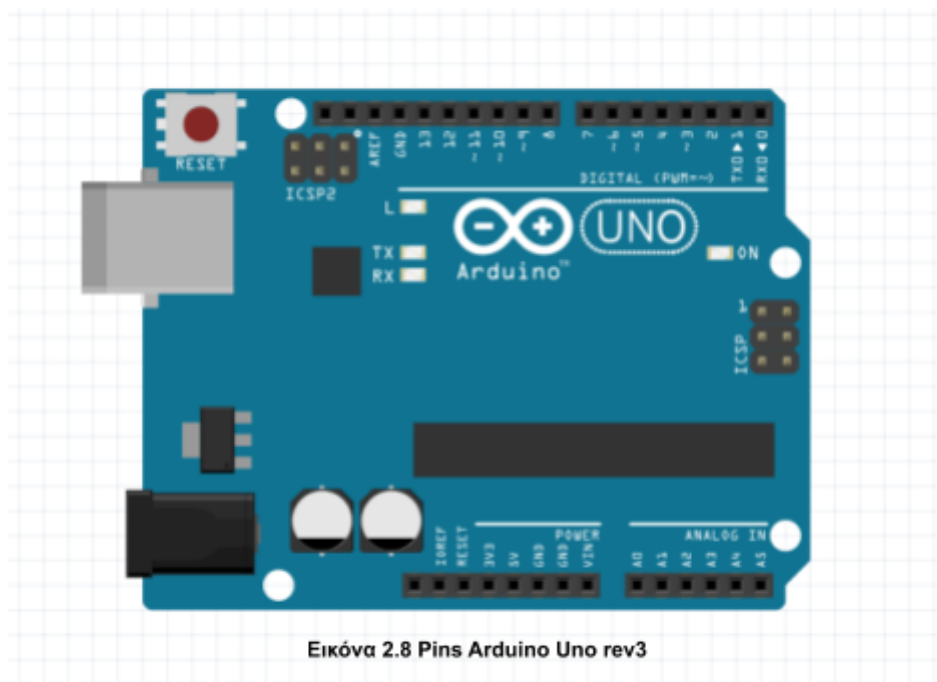
2.3.1.2 Τεχνικά χαρακτηριστικά

Υλικό/Ιδιότητα	Τιμή
Μικροελεγκτής	ATmega328P
Τάση λειτουργίας	5V
Τάση εισόδου (συνιστάται)	7-12V
Τάση εισόδου (όριο)	6-20V
Ψηφιακές ακίδες εισόδου / εξόδου	14 (εκ των οποίων 6 παρέχουν έξοδο PWM)
Ψηφιακές ακίδες εισόδου / εξόδου PWM	6
Αναλογικοί ακροδέκτες εισόδου	6
Τάση DC ανά είσοδο I / O	20 mA
DC ρεύμα για 3.3V ακίδων	50 mA
Μνήμη Flash	32 KB (ATmega328P) εκ των οποίων 0,5 KB χρησιμοποιείται από το bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Ταχύτητα ρολογιού	16 MHz
LED_BUILTIN	13
Μήκος	68,6 mm
Πλάτος	53,4 mm
Βάρος	25 g

Πίνακας 2.3 Τεχνικά χαρακτηριστικά Arduino Uno rev3

2.3.1.3 Pins

Ακολουθεί μια εικόνα των υποδοχών ακαι των ακροδεκτών της πλακέτας Arduino Uno rev3 καθώς και μια σύντομη επεξήγησή τους.



Εικόνα 2.8 Pins Arduino Uno rev3

2.3.1.3.1 Pins γενικής χρήσης

LED : Ενσωματωμένη λυχνία LED που ενεργοποιείται από την υποδοχή 13. Όταν ο ακροδέκτης έχει υψηλή τιμή, η λυχνία LED είναι αναμμένη, όταν έχει χαμηλή τιμή είναι απενεργοποιημένο.

VIN : Η τάση εισόδου στην πλακέτα Arduino όταν χρησιμοποιεί εξωτερική πηγή τροφοδοσίας.

5V : Αυτός ο ακροδέκτης εκπέμπει ρυθμισμένο 5V από τον ρυθμιστή του πίνακα. Η κάρτα μπορεί να τροφοδοτηθεί είτε από την υποδοχή τροφοδοσίας συνεχούς ρεύματος (7 - 20V), τον συνδετήρα USB (5V) ή τον ακροδέκτη VIN του πίνακα (7-20V). Η τροφοδοσία τάσης μέσω των ακροδεκτών 5V ή 3.3V παρακάμπτη τον ρυθμιστή και μπορεί να προκαλέσει βλάβη στην πλακέτα.

3V3 : Παροχή 3.3 volt από τον εποχούμενο ρυθμιστή. Η μέγιστη έλξη ρεύματος είναι 50 mA.

GND : Πείροι γείωσης.

IOREF : Αυτό το pin στον πίνακα Arduino παρέχει την αναφορά τάσης με την οποία λειτουργεί ο μικροελεγκτής.

Επαναφορά : Χρησιμοποιείται συνήθως για την προσθήκη ενός κουμπιού επαναφοράς σε shields που εμποδίζουν αυτό που υπάρχει στην πλακέτα.

2.3.1.3.2 Pins ειδικών χρήσεων

Κάθε μία από τις 14 ψηφιακές ακίδες και 6 αναλογικές ακίδες στο Uno μπορεί να χρησιμοποιηθεί ως είσοδος ή έξοδος, χρησιμοποιώντας λειτουργίες pinMode (), digitalWrite () και digitalRead (). Λειτουργούν με 5 βολτ. Κάθε ακροδέκτης μπορεί να παρέχει ή να λαμβάνει 20 mA ως συνιστώμενη κατάσταση λειτουργίας και έχει εσωτερική αντίσταση pull-up (αποσυνδεδεμένη από προεπιλογή) 20-50k ohm. Μέγιστο 40mA είναι η τιμή που δεν πρέπει να ξεπεραστεί σε οποιονδήποτε ακροδέκτη I / O για να αποφευχθεί η μόνιμη βλάβη του μικροελεγκτή.

Το Uno διαθέτει 6 αναλογικές εισόδους, με την ένδειξη A0 έως A5, καθένα από τα οποία παρέχει 10 ψηφία ανάλυσης (δηλαδή 1024 διαφορετικές τιμές).

Επιπλέον, μερικές καρφίτσες έχουν εξειδικευμένες λειτουργίες:

- Serial / UART : ακίδες 0 (RX) και 1 (TX). Χρησιμοποιείται για τη λήψη (RX) και τη μετάδοση (TX) TTL σειριακών δεδομένων. Οι ακίδες αυτές συνδέονται με τους αντίστοιχους ακροδέκτες του σειριακού τσιπ ATmega8U2 USB-to-TTL.
- Εξωτερικές διακοπές : οι ακίδες 2 και 3. Αυτές οι ακίδες μπορούν να διαμορφωθούν για να προκαλέσουν διακοπή σε χαμηλή τιμή, άνοδο ή πτώση της άκρης ή αλλαγή τιμής.
- PWM (διαμόρφωση εύρους παλμών): 3, 5, 6, 9, 10 και 11. Μπορεί να παρέχει έξοδο PWM 8 bit με τη λειτουργία analogWrite ().
- SPI (σειριακή περιφερειακή διασύνδεση): 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Αυτές οι ακίδες υποστηρίζουν την επικοινωνία SPI χρησιμοποιώντας τη βιβλιοθήκη SPI.

- TWI (διασύνδεση δύο καλωδίων) / I²C : ακίδα A4 ή SDA και ακίδα A5 ή SCL. Υποστήριξη επικοινωνίας TWI χρησιμοποιώντας τη βιβλιοθήκη καλωδίων.
- AREF (αναλογική αναφορά): Τάση αναφοράς για τις αναλογικές εισόδους.

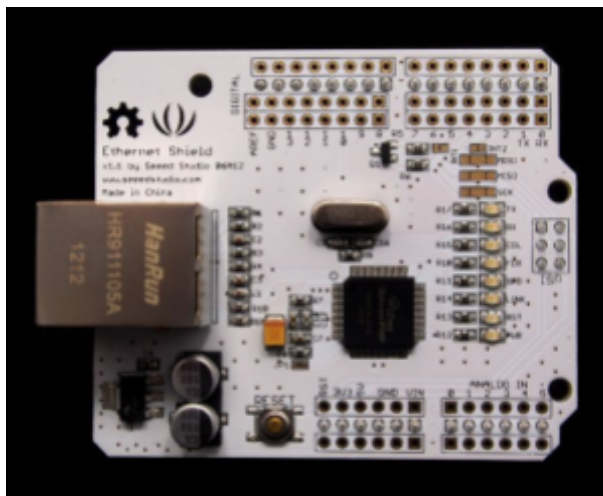
2.3.1.4 Επικοινωνία

Το Arduino Uno διαθέτει μια σειρά από εγκαταστάσεις επικοινωνίας με έναν υπολογιστή, έναν άλλο πίνακα Arduino / Geniino ή άλλους μικροελεγκτές. Το ATmega328 παρέχει σειριακή επικοινωνία UART TTL (5V), η οποία διατίθεται σε ψηφιακές ακίδες 0 (RX) και 1 (TX). Ένα ATmega16U2 στον πίνακα διοχετεύει αυτή τη σειριακή επικοινωνία μέσω USB και εμφανίζεται ως μια εικονική θύρα com για λογισμικό στον υπολογιστή. Το firmware 16U2 χρησιμοποιεί τα πρότυπα προγράμματα οδήγησης USB COM και δεν απαιτείται εξωτερικό πρόγραμμα οδήγησης. Ωστόσο, στα Windows, απαιτείται ένα αρχείο .inf.

Το λογισμικό Arduino (IDE) περιλαμβάνει μια σειριακή οθόνη που επιτρέπει την αποστολή απλών στοιχείων κειμένου προς και από την πλακέτα. Οι ενδεικτικές λυχνίες RX και TX στην πλακέτα θα αναβοσβήνουν όταν τα δεδομένα μεταδίδονται μέσω του τσιπ USB-to-serial και της σύνδεσης USB στον υπολογιστή (αλλά όχι για σειριακή επικοινωνία στις ακίδες 0 και 1).

2.3.2 Ethernet Shield

Για την διασύνδεση στο δίκτυο χρησιμοποιήθηκε το Ethernet Shield και και πιο συγκεκριμένα το Wiznet Ethernet Shield v.1.1 by Seed. Με τη χρήση του έγινε εύκολα άμεση σύνδεση στο διαδίκτυο.



Εικόνα 2.9 Arduino Ethernet Shield (v1.1 Wiznet by Seed)

2.3.2.1 Ιδιότητες και τεχνικά χαρακτηριστικά

Οι βασικές ιδιότητες αυτού του shield είναι:

- Συμβατότητα με Arduino και Arduino Mega.
- Συμβατότητα με τη βιβλιοθήκη Arduino Ethernet Library
- Τυπικό βύσμα Ethernet RJ45
- 16K byte εσωτερικού buffer
- Δείκτες LED για όλες τις λειτουργίες
- Εύκολο στη χρήση κουμπί επαναφοράς
- Βασικές κεφαλίδες για πινακες πρωτοτύπων πλέγματος 0.1"
- Έως 4 συνδέσεις δικτύου TCP / UDP

Τα κυριότερα τεχνικά χαρακτηριστικά συμπεριλαμβάνονται στον ακόλουθο πίνακα:

Parameter	Ethernet Shield V1.0 (V1.1)
Working Voltage	(+)5V
Control Mode	Control by seeeduino/Arduino
TCP/IP Ethernet Controller	Wiz5100
power supply	5V pin of Arduino/Seeeduino
Standard Shield	No
SD Card Socket	No
Grove Connector	No
Ethernet jack	Standard RJ45

Πίνακας 2.4 Τεχνικά χαρακτηριστικά Ethernet Shield v1.1

2.3.2.2 Επεξήγηση πλακέτας και διαθέσιμων pins

RJ45: θύρα Ethernet

Power LED: Ισχύς

RST LED (red): Κόκκινο όταν επαναφέρεται

LINK LED: Πάντα ενεργοποιείται όταν ο σύνδεσμος είναι εντάξει και αναβοσβήνει ενώ βρίσκεται σε κατάσταση TX ή RX

SPD LED: Το φως δείχνει ότι η ταχύτητα σύνδεσης είναι 100Mbps

LED FDX: Η λυχνία υποδεικνύει την κατάσταση της λειτουργίας πλήρους αμφίδρομης λειτουργίας.

Coil LED: Το φως δείχνει την παρουσία δραστηριότητας σύγκρουσης

LED Rx: Το φως υποδεικνύει την παρουσία δραστηριότητας λήψης

LED Tx: Το φως δείχνει την παρουσία δραστηριότητας μετάδοσης

Reset KEY: Επαναφορά της θωράκισης Ethernet και του Arduino όταν πατηθεί

D0: Αχρησιμοποίητο

D1: Αχρησιμοποίητο

D2: Συνδέει τον ακροδέκτη INT του W5100

D3: Ελέγξτε την επαναφορά του W5100

D4: Αχρησιμοποίητο

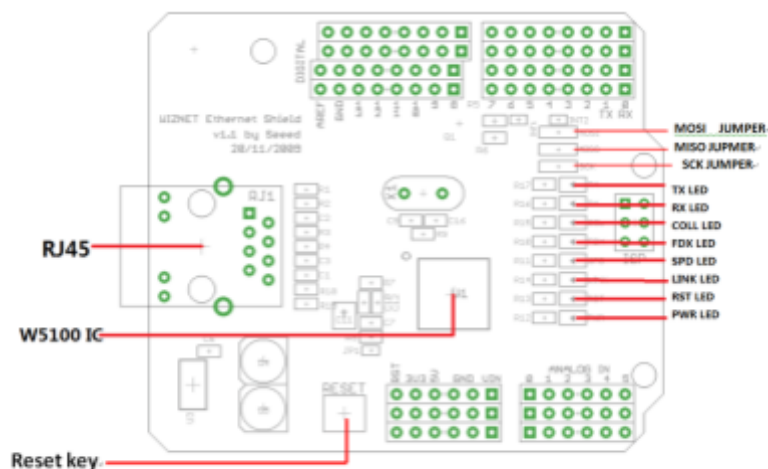
D5: Αχρησιμοποίητο

D6: Αχρησιμοποίητο

D7: Αχρησιμοποίητο

D8: Αχρησιμοποίητο

D9: Αχρησιμοποίητο



Εικόνα 2.10 Pins Ethernet Shield (v1.1 by Seed)

D10: Χρησιμοποιείται για επιλογή SPI Chip Select

D11: Χρησιμοποιείται για SPI MOSI

D12: Χρησιμοποιείται για το SPI MISO

D13: Χρησιμοποιείται για SPI SCK

D14 (A0): Αχρησιμοποίητο

D15 (A1): Δεν χρησιμοποιείται

D16 (A2): Αχρησιμοποίητο

D17 (A3): Αχρησιμοποίητο

D18 (A4): Αχρησιμοποίητο

D19 (A5): Δεν χρησιμοποιείται

3. Δίποδα (biped) ρομπότ

Για την κατασκευή του ρομπότ της πτυχιακής εργασίας (το οποίο παρελήφθη έτοιμο από προηγούμενη πτυχιακή εργασία) χρησιμοποιήθηκαν κινητήρες servo για την κίνηση των κινητών μερών του δίποδου ρομπότ και μεταλλικά εξαρτήματα για την μορφοποίηση του σώματος του.

3.1 Κινητήρες τύπου servo

Τα servo είναι ένα είδος ενεργοποιητή (actuator) που επιτρέπει τον ακριβή έλεγχο της γωνιακής ή γραμμικής θέσης, της ταχύτητας και της επιτάχυνσης. Περιέχει έναν κινητήρα, οδοντωτούς τροχούς, ένα κύκλωμα ελέγχου και έναν αισθητήρα θέσης (γωνίας στροφής). Όλα αυτά μαζί αποτελούν ένα σύστημα αυτομάτου ελέγχου κλειστού βρόχου.

Τα servo δεν είναι μια συγκεκριμένη κατηγορία κινητήρα, αν και ο όρος *σερβοκινητήρας* χρησιμοποιείται συχνά για να αναφέρεται σε έναν κινητήρα κατάλληλο για χρήση σε ένα σύστημα ελέγχου κλειστού βρόχου .

Τα χρησιμοποιούνται σε εφαρμογές όπως ρομποτική, μηχανήματα CNC ή αυτοματοποιημένη κατασκευή.



Εικόνα 3.1 Παράδειγμα servo

3.1.1 Πλεονεκτήματα και μειονεκτήματα έναντι άλλων τύπων κινητήρων

3.1.1.1 Πλεονεκτήματα

- Χαμηλό κόστος
- Μικρές διαστάσεις και εύχρηστο σχήμα: όλα τα τμήματά ενός σέρβο περιβάλλονται από ένα συμπαγές περίβλημα από το οποίο εξέρχει μόνο ο ελικός άξονας κίνησης
- Παράγουν υψηλές τιμές ροπής
- Δεν απαιτείται χρήση αισθητήρων και κυκλωμάτων ανάδρασης για τον προσδιορισμό της θέσης του άξονα κίνησης

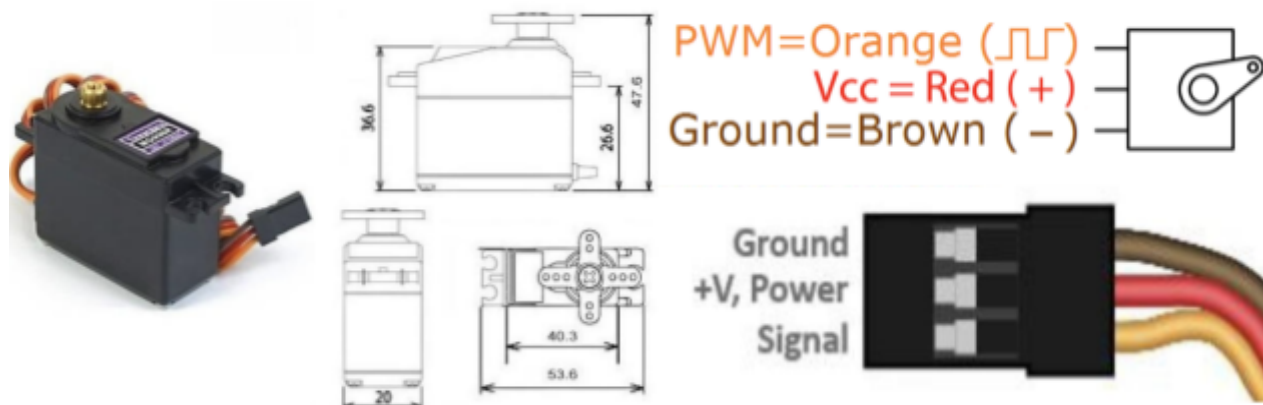
3.1.1.2 Μειονεκτήματα

- Αδυναμία εκτέλεσης πλήρους και συνεχούς περιστροφής

3.1.2 Servo Tower Pro MG996R

Για την κατασκευή του ρομποτ που χρησιμοποιήθηκε στην εργασία οι φοιτητές που το συναρμολόγησαν είχαν χρησιμοποιήσει τα servo Tower Pro MG996R.

Το MG996R διαθέτει μεταλλικά γρανάζια που οδηγούν σε επιπλέον 10kg παγίδα ροπής. Είναι ουσιαστικά μια αναβαθμισμένη έκδοση του γνωστό σέρβο MG995 και διαθέτει αναβαθμισμένη προστασία από τα κρούσματα και επανασχεδιασμένο PCB και IC σύστημα ελέγχου που το καθιστούν πολύ ακριβέστερο από τον προκάτοχό του. Το γρανάζι και ο κινητήρας έχουν επίσης αναβαθμιστεί για να βελτιώσουν το νεκρό εύρος ζώνης και το κεντράρισμα. Η μονάδα συμπεριλαμβάνει συρμάτινο καλώδιο 30cm και 3 ακίδων τύπου θηλυκού κεφαλίδας "S" που ταιριάζει στους περισσότερους δέκτες. Το servo αυτό μπορεί να περιστρέφεται περίπου 120 μοίρες (60 σε κάθε κατεύθυνση).



Εικόνα 3.2 Servo TowerPro MG996R

3.1.2.1 Τεχνικά χαρακτηριστικά

Property	Value
Weight	55g
Dimension	40,7×19,7×42,9mm
Stall torque	9.4kg/cm (4.8v); 11kg/cm (6.0v)
Operating speed	0.19sec/60degree (4.8v); 0.15sec/60degree (6.0v)
Operating voltage	4.8~ 6.6v
Gear Type	Metal gear
Temperature range	0- 55deg
Servo Plug	JR (Fits JR and Futaba)
Dead band width	1us
Servo wire length	32cm
Current draw at idle 10mA	
No load operating current draw 170mA	
Stall current draw 1400mA	

Πίνακας 3.1 Τεχνικά χαρακτηριστικά Tower Pro MG996R

Wire Number	Wire Colour	Description
1	Brown	Ground wire connected to the ground of system
2	Red	Powers the motor typically +5V is used
3	Orange	PWM signal is given in through this wire to drive the motor

Πίνακας 3.2 Καλωδίωση Tower Pro MG996R

3.3 Biped robots

3.3.1 Ρομπότ

Η λέξη ρομπότ προέρχεται από την τσέχικη λέξη *robot* (ρομπότα) που σημαίνει εργασία. Καθιερώθηκε ως όρος με την σημερινή του έννοια το 1920 από τον Τσέχο θεατρικό συγγραφέα Κάρελ Τσάπек στο έργο του "R.U.R." (Rossum's Universal Robots), όπου σατιρίζει την εξάρτηση της κοινωνίας από τους μηχανικούς εργάτες (ρομπότ) της τεχνολογικής εξέλιξης και που τελικά εξοντώνουν τους δημιουργούς τους. Σε πολλές σύγχρονες σλαβικές γλώσσες (π.χ. την πολωνική) χρησιμοποιείται σαν έκφραση της καθημερινότητας με την έννοια της σκληρής δουλειάς (αντίστοιχο του χαμαλίκι).

Ως ρομπότ ορίζεται μια μηχανή που έχει ανθρωπόμορφη συμπεριφορά και εκτελεί ανθρώπινες εργασίες με προγραμματισμένες εντολές του ανθρώπου. Δηλαδή "αισθάνεται", "σκέφτεται" και "επενεργεί". Ένα ρομποτικό σύστημα διαθέτει αισθητήρες, για την απόκτηση πληροφορίας είτε από το εξωτερικό περιβάλλον είτε σε σχέση με την εσωτερική του κατάσταση. Έχει τη δυνατότητα επεξεργασίας των πληροφοριών που συλλέγει και επενεργητές για την εκτέλεση κάποιας λειτουργίας στο περιβάλλον με βάση τις πληροφορίες που έλαβε ή τις εντολές που δέχεται ή και συνδυασμό αυτών.

Οι βασικές ιδιότητες ενός ρομπότ είναι:

- δυνατότητες επαναπρογραμματισμού
- δυνατότητες μηχανικής δράσης, εκτέλεση φυσικών εργασιών πάνω στο περιβάλλον
- προσαρμοστικότητα, ευελιξία, πολυσχιδής λειτουργικότητα

Ευρύτατη χρήση ρομπότ γίνεται σε πάρα πολλούς παραγωγικούς τομείς και κυρίως στη βιομηχανία (βιομηχανική ρομποτική), στην ιατρική, την αεροναυπηγική, την αεροδιαστημική κ.ά, γεγονός που έδωσε σημαντική ώθηση στην περαιτέρω ανάπτυξη της βιομηχανίας των ρομπότ, ιδιαίτερα στην Ιαπωνία και τις ΗΠΑ.



Εικόνα 3.3 Bipedal Ρομπότ (ανδρoειδής) Asimo (2000) κατασκευασμένο από την Honda

3.3.2 Biped robot

Ένα ζώο ή ένα μηχάνημα που συνήθως μετακινείται ή στέκεται σε δύο προσαρτήματα (τυπικά πόδια) είναι γνωστό ως 'biped' (/ bi.ped /), που σημαίνει "δύο πόδια" (λατινικά bi = δύο + ped = πόδι). Έτσι ένα ρομπότ που στέκεται και μετακινείται σε δύο πόδια ονομάζεται biped robot.

Το biped robot που χρησιμοποιήθηκε, όπως αναφέρθηκε και στην εισαγωγή, είχε κατασκευαστεί στα πλαίσια της πτυχιακής εργασίας των Αποστόλη Γαλανόπουλου και Αλέξανδρου Ντίτορα: Biped robot 9DOF controlled via android application (bluetooth and arduino).

Αποτελείται από 8 servo TowerPro MG996R τα οποία έχουν τοποθετηθεί έτσι ώστε να μπορούν να ελέγχονται/κινούνται οι πατούσες/αστράγαλοι, πόδια, ώμοι και αγκώνες. Όπως έχουμε αναφέρει σε προηγούμενο κεφάλαιο τα servo ελέγχονται από την πλακέτα Arduino Uno.



Εικόνα 3.4 Biped robot (9DOF) που χρησιμοποιήθηκε στην εργασία

4. Διαδικτυακές εφαρμογές

Μια εφαρμογή η οποία είναι διαθέσιμη στους χρήστες της μέσω διαδικτύου ή του ενδοδικτύου και ο χρήστης για να την χρησιμοποιήσει χρειάζεται μόνο ένα περιηγητή χαρακτηρίζεται ως διαδικτυακή εφαρμογή. Οι εφαρμογές αυτές συνήθως εκτελούνται σε υπολογιστικές μηχανές οι οποίες έχουν τον ρόλο του σταθμού εξυπηρέτησης και παρέχουν τις υπηρεσίες τους σε περισσότερους του ενός χρήστη.

Η διαδικτυακή εφαρμογή που θα υλοποιήσουμε για την εργασία αυτή θα φιλοξενείται στο Arduino Ethernet shield και θα κάνει χρήση των HTML και CSS.

4.1 HTML

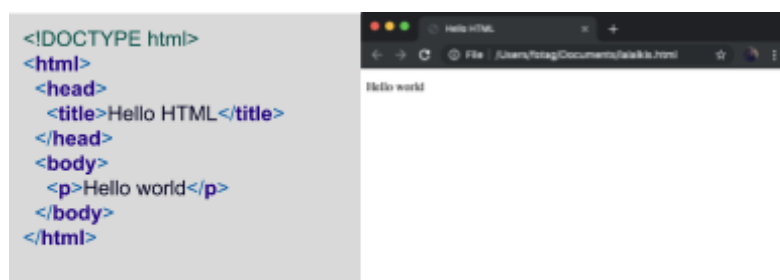
Η HTML (HyperText Markup Language, ελλ. Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων.

Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα <html>), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα <h1> και </h1>), με την πρώτη να ονομάζεται ετικέτα έναρξης (<tag_name>) και τη δεύτερη ετικέτα λήξης (</tag_name>). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ.

Ο σκοπός ενός web browser είναι να διαβάζει τα έγγραφα HTML και να τα συνθέσει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να παρουσιάσει το περιεχόμενο της σελίδας.

Οι Web browsers μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού. Ο οργανισμός W3C, ο οποίος δημιουργεί και συντηρεί τα πρότυπα για την HTML και τα CSS, ενθαρρύνει τη χρήση των CSS αντί διαφόρων στοιχείων της HTML για σκοπούς παρουσίασης του περιεχομένου.

Ένα παράδειγμα ενός στοιχείου html και ένας πίνακας με τα μερικά από τα βασικά html tags:



Εικόνα 4.1 Hello World html page

Tag	Description
<u><!DOCTYPE</u>	Ορίζει τον τύπο εγγράφου
<u>></u>	
<u><html></u>	Ορίζει ένα έγγραφο HTML
<u><head></u>	Ορίζει πληροφορίες σχετικά με το έγγραφο
<u><title></u>	Ορίζει έναν τίτλο για το έγγραφο
<u><body></u>	Ορίζει το σώμα του εγγράφου
<u><h1> to <h6></u>	Ορίζει τις κεφαλίδες HTML
<u><p></u>	Ορίζει μια παράγραφο
<u>
</u>	Εισάγει ένα σπάσιμο μιας γραμμής
<u><hr></u>	Ορίζει μια θεματική αλλαγή στο περιεχόμενο
<u><!--...--></u>	Ορίζει ένα σχόλιο
<u></u>	Ορίζει το έντονο κείμενο
<u><bdi></u>	Απομονώνει ένα μέρος του κειμένου που μπορεί να μορφοποιηθεί σε διαφορετική κατεύθυνση από το άλλο κείμενο εκτός αυτού
<u><bdo></u>	Καταργεί την τρέχουσα κατεύθυνση κειμένου
<u><blockquote></u>	Ορίζει μια ενότητα που αναφέρεται σε άλλη πηγή
<u><code></u>	Ορίζει ένα κομμάτι κώδικα υπολογιστή
<u></u>	Ορίζει κείμενο με έμφαση
<u><i></u>	Ορίζει ένα μέρος του κειμένου σε μια εναλλακτική φωνή ή διάθεση
<u><pre></u>	Ορίζει προκαθορισμένο κείμενο
<u></u>	Ορίζει σημαντικό κείμενο
<u><template></u>	Ορίζει ένα πρότυπο
<u><time></u>	Ορίζει μια ημερομηνία / ώρα

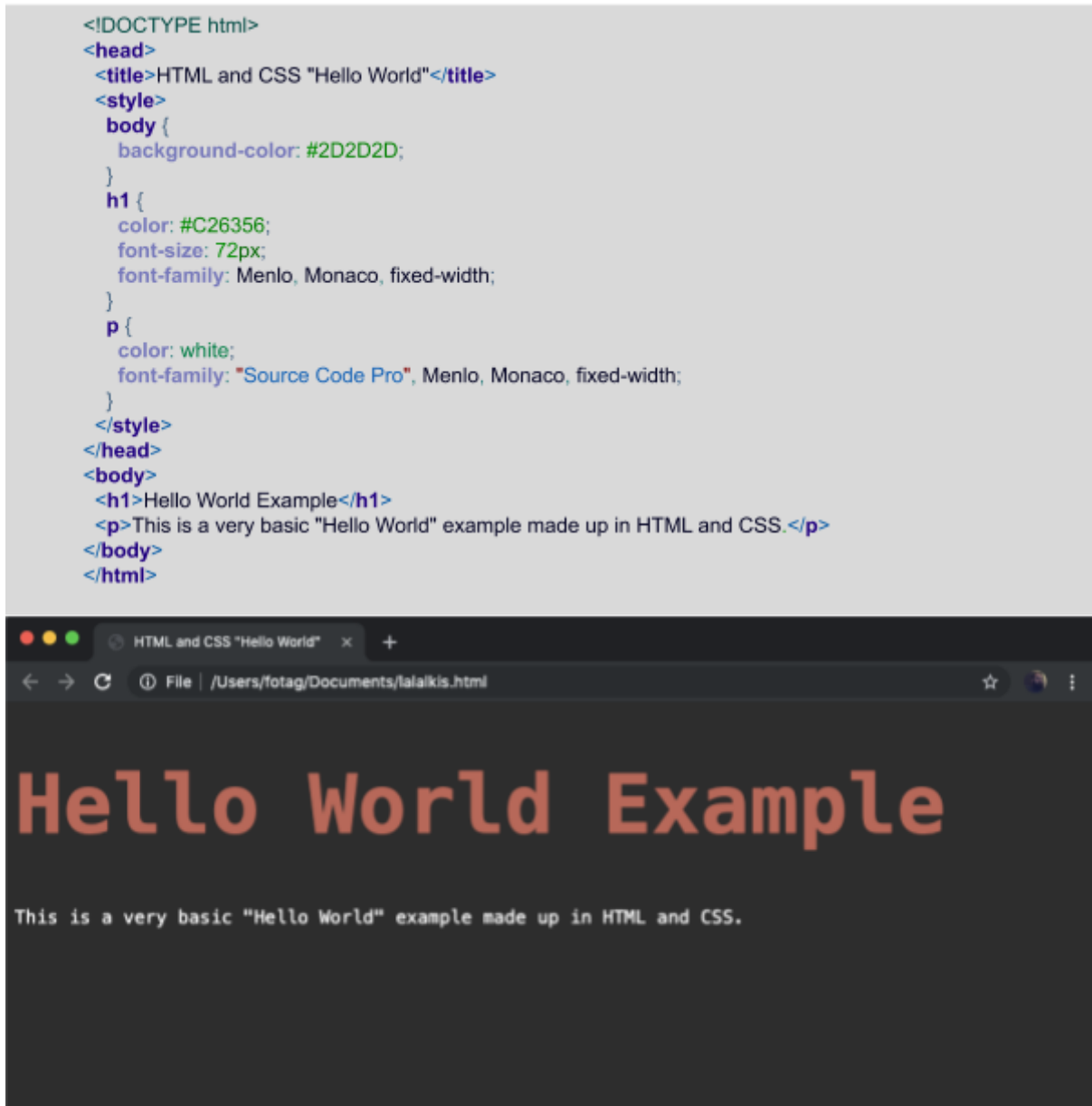
Πίνακας 4.1 Μερικά html tags

4.2 CSS

Η CSS (Cascading Style Sheets – διαδοχικά φύλλα ύφους ή επάλληλα φύλλα ύφους) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων ύφους που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης.

Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στυλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη.

Ένα παράδειγμα ενός στοιχείου html με την μορφοποίηση με CSS να περιλαμβάνεται στην HTML και ένας πίνακας με μερικές ιδιότητες css:



Εικόνα 4.2 hello world html page with css

Property	Description
color	Ορίζει το χρώμα ενός κειμένου
line-height	Ορίζει την απόσταση μεταξύ γραμμών
letter-spacing	Αυξήστε ή μειώστε το διάστημα μεταξύ των χαρακτήρων
text-align	Ευθυγραμμίζει το κείμενο σε ένα στοιχείο
text-decoration	Προσθέτει διακόσμηση στο κείμενο
text-indent	Εστέλλει την πρώτη γραμμή κειμένου σε ένα στοιχείο
text-transform	Ελέγχει τα γράμματα σε ένα στοιχείο
list-style	Ορίζει όλες τις ιδιότητες για μια λίστα σε μία δήλωση
list-style-image	Καθορίζει μια εικόνα ως δείκτη γραμματοσειράς
list-style-position	Καθορίζει πού να τοποθετηθεί ο δείκτης στοιχείου λίστας
list-style-type	Καθορίζει τον τύπο του δείκτη στοιχείου λίστας
border	Ορίζει όλες τις ιδιότητες των συνόρων σε μία δήλωση
border-color	Ορίζει το χρώμα των τεσσάρων συνόρων
border-style	Ορίζει το στυλ των τεσσάρων συνόρων
border-top	Ορίζει όλες τις ιδιότητες των κορυφαίων συνόρων σε μία δήλωση
border-top-color	Ορίζει το χρώμα του άνω περιθωρίου
border-top-style	Ορίζει το στυλ του άνω περιθωρίου
border-top-width	Ορίζει το πλάτος του άνω περιθωρίου
border-width	Ορίζει το πλάτος των τεσσάρων ορίων
font	Ορίζει όλες τις ιδιότητες γραμματοσειράς σε μία δήλωση
font-family	Καθορίζει την οικογένεια γραμματοσειρών για κείμενο
font-size	Καθορίζει το μέγεθος γραμματοσειράς του κειμένου
font-style	Καθορίζει το στυλ γραμματοσειράς για κείμενο
font-variant	Καθορίζει αν ένα κείμενο πρέπει να εμφανίζεται ή όχι σε μια μικρή γραμματοσειρά
font-weight	Καθορίζει το βάρος μιας γραμματοσειράς

Πίνακας 4.2 Μερικές ιδιότητες CSS

4.3 Διάθεση διαδικτυακής εφαρμογής

Για να χρησιμοποιήσουν οι χρήστες την διαδικτυακή εφαρμογή από τον browser τους χρησιμοποιείται το πρωτόκολλο http έτσι ώστε ο browser να επικοινωνήσει με τον εξυπηρετητή που φιλοξενεί την εφαρμογή.

4.3.1 HTTP πρωτόκολλο

Το Πρωτόκολλο Μεταφοράς Υπερκειμένου (HyperText Transfer Protocol, HTTP) είναι ένα πρωτόκολλο επικοινωνίας. Αποτελεί το κύριο πρωτόκολλο που χρησιμοποιείται στους φυλλομετρητές (browser) του Παγκοσμίου Ιστού για να μεταφέρει δεδομένα ανάμεσα σε έναν διακομιστή (server) και έναν πελάτη (client).



Εικόνα 4.3 HTTP επικοινωνία μεταξύ εξυπηρετητή και πελάτη

Οι λειτουργίες που υποστηρίζει το HTTP ονομάζονται μέθοδοι. Κάθε αίτηση στον εξυπηρετητή αποτελείται από μία ή περισσότερες γραμμές κειμένου ASCII. Η πρώτη λέξη της πρώτης γραμμής της αίτησης είναι το όνομα της ζητούμενης μεθόδου.

Method verb	Περιγραφή
GET	Η μέθοδος GET ζητά από το διακομιστή να στείλει τη σελίδα. Η σελίδα κωδικοποιείται κατάλληλα σε μορφή MIME. Η πιο συνήθης μορφή της μεθόδου GET είναι η εξής: GET όνομα_αρχείου HTTP/1.1 όπου το όνομα_αρχείου προσδιορίζει το όνομα του πόρου που πρέπει να προσκομιστεί και το 1.1 είναι η έκδοση του πρωτοκόλλου που χρησιμοποιείται.
HEAD	Η μέθοδος HEAD ζητά μόνο την κεφαλίδα του μηνύματος, χωρίς τη πραγματική σελίδα. Η μέθοδος αυτή μπορεί να χρησιμοποιηθεί για τη συλλογή πληροφοριών για λόγους δεικτοδότησης ή απλώς και μόνο για τον έλεγχο εγκυρότητας μιας διεύθυνσης URL.
POST	Η μέθοδος POST χρησιμοποιείται κατά την υποβολή φερμών. Όπως και η μέθοδος GET, η POST περιέχει μια διεύθυνση URL αλλά αντί να ανακτά απλώς τη σελίδα μεταφέρει δεδομένα στον διακομιστή όπως για παράδειγμα τα περιεχόμενα της φόρμας. Έπειτα ο διακομιστής κάνει κάτι με αυτά τα δεδομένα ανάλογα με το URL. Τέλος, η μέθοδος επιστρέφει μια σελίδα που δείχνει το αποτέλεσμα.
PUT	Η μέθοδος PUT είναι η αντίστροφη της GET, δηλαδή αντί να διαβάζει τη σελίδα, γράφει τη σελίδα. Η μέθοδος αυτή κάνει εφικτή την κατασκευή μιας συλλογής ιστοσελίδων σε έναν απομακρυσμένο διακομιστή. Το σώμα της αίτησης περιέχει τη σελίδα. Μπορεί να κωδικοποιείται μέσω του MIME, οπότε οι γραμμές που ακολουθούν την PUT μπορεί να περιέχουν κεφαλίδες Content-Type και πιστοποίησης ταυτότητας, ώστε να αποδείξουν ότι ο αιτών έχει πραγματικά την άδεια να εκτελέσει τη ζητούμενη λειτουργία.
DELETE	Η μέθοδος DELETE καταργεί τη σελίδα ή τουλάχιστον δηλώνει ότι ο διακομιστής Ιστού έχει συμφωνήσει να καταργήσει τη σελίδα. Όπως και με την PUT και σε αυτή τη μέθοδο παίζουν μεγάλο ρόλο η πιστοποίηση της ταυτότητας και της άδειας εκτέλεσης της λειτουργίας.
TRACE	Η μέθοδος TRACE χρησιμοποιείται για αποσφαλμάτωση. Ζητά από τον διακομιστή να επιστρέψει την αίτηση. Η μέθοδος αυτή είναι χρήσιμη όταν η επεξεργασία των αιτήσεων δεν γίνεται σωστά και ο πελάτης θέλει να δει ποια αίτηση έλαβε πραγματικά ο διακομιστής.
CONNECT	Η μέθοδος CONNECT επιτρέπει στον χρήστη να πραγματοποιήσει σύνδεση με έναν διακομιστή Ιστού μέσω μιας ενδιάμεσης συσκευής, για παράδειγμα μέσω μια κρυφής μνήμης Ιστού.
OPTIONS	Η μέθοδος OPTIONS παρέχει έναν τρόπο ώστε ο πελάτης να στέλνει ερωτήματα στον διακομιστή σχετικά με μια σελίδα και να λαμβάνει τις μεθόδους και τις κεφαλίδες που μπορούν να χρησιμοποιηθούν με αυτή τη σελίδα.

Πίνακας 4.3 HTTP μέθοδοι

4.3.2 Web Servers

Ένας εξυπηρετητής ιστού είναι λογισμικό διακομιστή ή υλικό που προορίζεται για την εκτέλεση του εν λόγω λογισμικού, το οποίο μπορεί να ικανοποιήσει αιτήματα πελατών του World Wide Web . Ένας εξυπηρετητής ιστού μπορεί γενικά να περιέχει έναν ή περισσότερους ιστότοπους. Ένας εξυπηρετητής ιστού επεξεργάζεται εισερχόμενα αιτήματα δικτύου μέσω HTTP και πολλά άλλα σχετικά πρωτόκολλα .

Η κύρια λειτουργία ενός διακομιστή ιστού είναι η αποθήκευση, επεξεργασία και παράδοση ιστοσελίδων σε πελάτες. Η επικοινωνία μεταξύ πελάτη και διακομιστή πραγματοποιείται χρησιμοποιώντας το πρωτόκολλο μεταφοράς υπερκειμένου (HTTP) . Οι σελίδες που παραδίδονται είναι πιο συχνά έγγραφα HTML , τα οποία μπορεί να περιλαμβάνουν εικόνες, φύλλα στυλ και δέσμες ενεργειών εκτός από το περιεχόμενο κειμένου.

Ένας πράκτορας χρήστη, συνήθως ένα πρόγραμμα περιήγησης ιστού ή ανιχνευτής ιστού, ενεργοποιεί την επικοινωνία κάνοντας αίτηση για συγκεκριμένο πόρο χρησιμοποιώντας HTTP και ο διακομιστής ανταποκρίνεται με το περιεχόμενο αυτού του πόρου ή με μήνυμα σφάλματος αν δεν το κάνει. Ο πόρος είναι συνήθως ένα πραγματικό αρχείο στη δευτερεύουσα αποθήκευση του διακομιστή , αλλά αυτό δεν συμβαίνει απαραίτητα και εξαρτάται από τον τρόπο με τον οποίο υλοποιείται ο διακομιστής ιστού.

Μέρος Δεύτερο

5. Περιγραφή υλοποίησης και λειτουργίας

Στο κεφάλαιο αυτό θα περιγράψουμε το πρακτικό μέρος της πτυχιακής εργασίας και τα βήματα τα οποία έγιναν προκειμένου να καταλήξουμε στο επιθυμητό αποτέλεσμα.

5.1 Βασική εγκατάσταση, παραμετροποίηση και δοκιμαστικό πρόγραμμα Arduino

Το πρώτο βήμα του πρακτικού μέρους της πτυχιακής εργασίας ήταν η εξοικείωση με τον Arduino UNO¹¹. Για να το πετύχουμε αυτό θα προσπαθήσουμε να εκτυπώνουμε ένα μήνυμα (Hello World!) στην σειριακή έξοδο κάθε 5 δευτερόλεπτα.

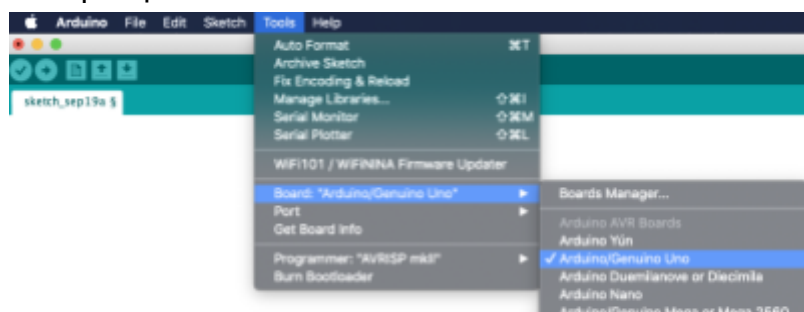
5.1.1 Συνδεσμολογία

Μετά την εγκατάσταση του Arduino IDE¹², συνδέσαμε με USB την Arduino UNO πλακέτα με τον υπολογιστή.



Εικόνα 5.1 Σύνδεση Arduino με υπολογιστή μέσω USB

- Παραμετροποιήσαμε το Arduino IDE για την πλακέτα που θα χρησιμοποιήσουμε: Tools > Board > Arduino/Genuino Uno

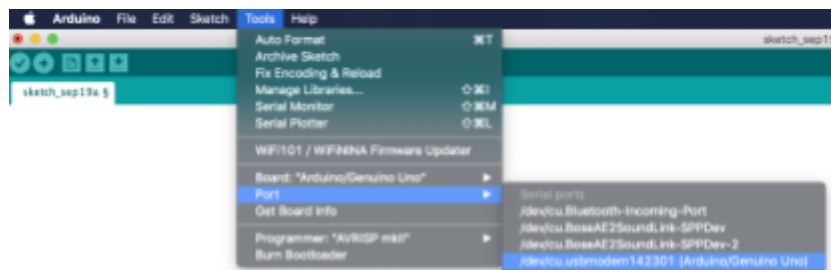


Εικόνα 5.2 Παραμετροποίηση πλακέτας στο Arduino IDE

¹¹ Βλέπετε [2.3.1 Arduino Uno](#)

¹² Βλέπετε [2.2.2.2.1 Εγκατάσταση και παραμετροποίηση σε Mac OS X](#)

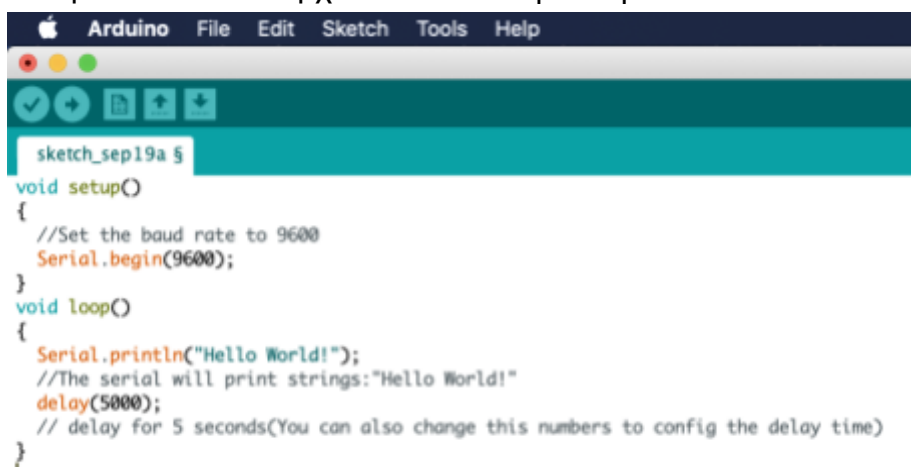
- Tools > Port > /dev/cu.usbmodem1423301 (Arduino/Genuino Uno), επιλογή θύρας επικοινωνίας (αλλάζει σε κάθε υπολογιστή ή νέα σύνδεση)




Εικόνα 5.3 Επιλογή θύρας στο Arduino IDE

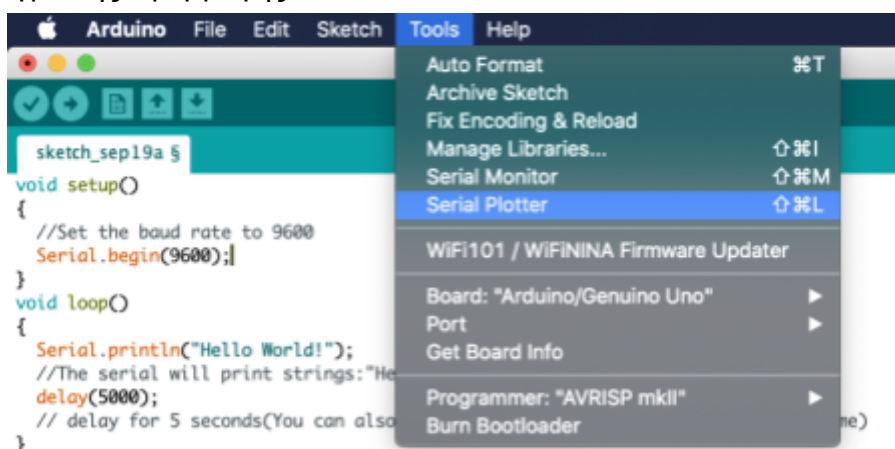
5.1.2 Υλοποίηση σε κώδικα

- Προετοιμασία του .ino αρχείου και αποθήκευση του Sketch στον τοπικό δίσκο



Εικόνα 5.4 Hello World arduino program

- Έλεγχος του κώδικα  για την ορθότητα του ως προς την σύνταξη.
- Άνοιγμα της εφαρμογής Serial Monitor: Tools > Serial Monitor



Εικόνα 5.5 Άνοιγμα της εφαρμογής Serial Monitor

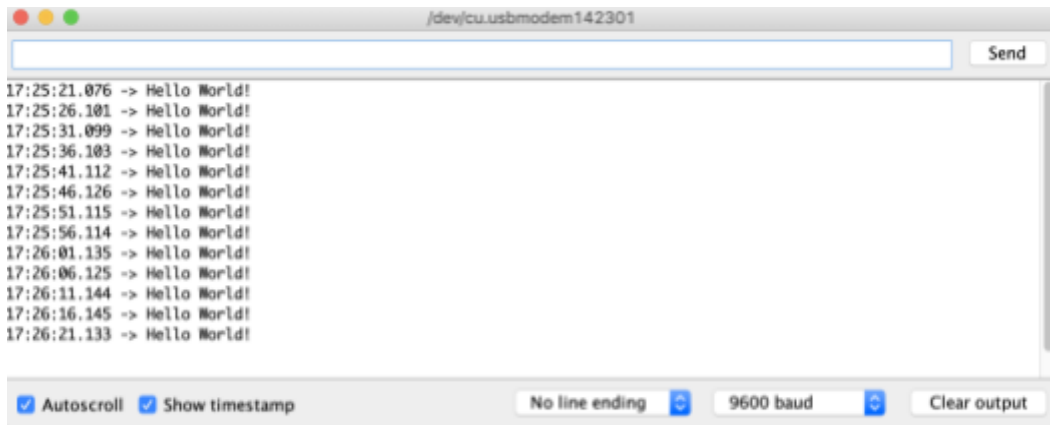
- Compile και μεταφόρτωση  του προγράμματος στην πλακέτα

5.1.3 Συμπεράσματα/Σχόλια

Στο βήμα αυτό λοιπόν έγινε εγκατάσταση του Arduino IDE και εξοικείωση με τον τρόπο συγγραφής κώδικα για προγράμματα arduino. Επίσης των διαδικασιών που χρειάζεται για να εγκατασταθεί το πρόγραμμα στο Arduino καθώς και πως μπορεί να γίνει εκσφαλμάτωση ή παρακολούθηση των αποτελεσμάτων.

Στόχος ήταν η εκτύπωση στην έξοδο του μηνύματος “Hello World!” κάθε 5 δευτερόλεπτα. Για να το πετύχουμε αυτό κάναμε χρήση της `println`¹³ μεθόδου στην serial port είδαμε τα αποτελέσματα με την εφαρμογή Serial Monitor του Arduino IDE (Εικόνα 5.5).

Επίσης δημιουργήθηκε αποθετήριο κώδικα στην πλατφόρμα Gitlab¹⁴, το οποίο θα αποθηκεύονται όλα τα σχετικά αρχεία της πτυχιακής εργασίας καθώς και τα διαγράμματα που μπορεί να χρειαστούν. Για την δημιουργία των διαγραμμάτων για τα κυκλώματα arduino χρησιμοποιήθηκε η πλατφόρμα Fritzing¹⁵.



Εικόνα 5.6 Serial Monitor μετά την μεταφόρτωση του προγράμματος

¹³ <https://www.arduino.cc/reference/en/language/functions/communication/serial/println/>

¹⁴ Βλέπε [Παράρτημα Β. Gitlab](#)

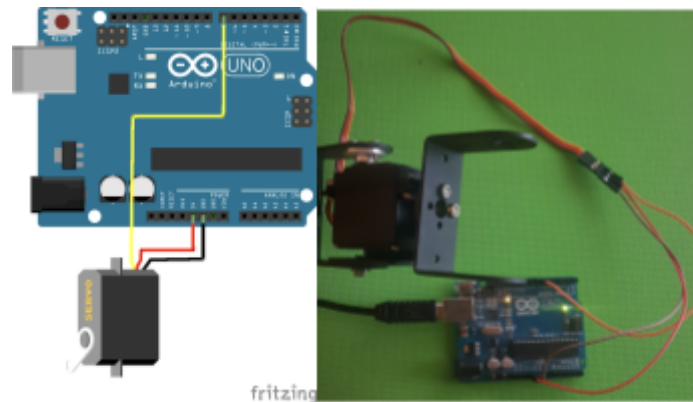
¹⁵ Βλέπε [Παράρτημα Α. Fritzing](#)

5.2 Σύνδεση και χειρισμός ενός servo κινητήρα από το Arduino

Στο δεύτερο βήμα θα προσπαθήσουμε να ελέγξουμε έναν servo κινητήρα¹⁶, συγκεκριμένα έναν από αυτούς που είναι ήδη εγκατεστημένοι στο biped robot¹⁷ (του αριστερού μπράτσου) από το Arduino. Η κίνηση που θα εκτελεί είναι μια συνεχόμενη κίνηση από 0 μέχρι 60 μοίρες.

5.2.1 Συνδεσμολογία

Συνδέουμε την γείωση του servo με την γείωση του Arduino, την τροφοδοσία του servo με τα 5V του Arduino και το control του servo με την υποδοχή 7 του Arduino. Στο βήμα αυτό ο Arduino τροφοδοτείται κατευθείαν από την USB θύρα που είναι ενωμένη στον υπολογιστή που τρέχουμε το Arduino IDE.



Εικόνα 5.7 Έλεγχος ενός servo του biped από τον Arduino

5.2.2 Υλοποίηση σε κώδικα

```

/*
 * Testing servo motor manipulation from Arduino
 */

#include <Servo.h>

Servo LeftArmServo; // Initialize servo at biped's left arm
int leftArmPin = 7; // Servo pulse at pin number 7 of arduino board

void setup() {
  LeftArmServo.attach(leftArmPin); // Attach servo motor pin
}

void loop() {
  // Move left arm 60 degrees
  for (int pos = 0; pos <= 60; ++pos) {
    LeftArmServo.write(pos);
    delay(50);
  }
}

```

Εικόνα 5.8 Κώδικας κίνησης ενός σέρβο Arduino IDE

¹⁶ Βλέπετε [3.1.2 Servo Tower Pro MG996R](#)

¹⁷ Βλέπετε [3.3.2 Biped robot](#)

5.2.3 Συμπεράσματα/Σχόλια

Για τον έλεγχο σου servo από τον Arduino, χρησιμοποιήθηκε η Servo.h βιβλιοθήκη. Η βιβλιοθήκη επιτρέπει στον Arduino να ελέγχει τους σερβοκινητήρες RC. Οι περισσότεροι σερβομηχανισμοί επιτρέπουν την τοποθέτηση του άξονα σε διάφορες γωνίες, συνήθως μεταξύ 0 και 180 μοιρών. Οι σερβομηχανές συνεχούς περιστροφής επιτρέπουν την περιστροφή του άξονα σε διάφορες ταχύτητες.

Η βιβλιοθήκη Servo υποστηρίζει έως και 12 κινητήρες στις περισσότερες πλακέτες Arduino. Σε πίνακες εκτός του Mega, η χρήση της βιβλιοθήκης απενεργοποιεί τη λειτουργία analogWrite () (PWM) στις ακίδες 9 και 10, ανεξάρτητα από το αν υπάρχει ή όχι Servo σε αυτές τις υποδοχές.

Σημειώστε ότι οι servo αντλούν σημαντική ισχύ, οπότε αν χρειάζεται να οδηγήσετε περισσότερους από έναν ή δύο, ίσως χρειαστεί να τους τροφοδοτήσετε από μια ξεχωριστή τροφοδοσία (δηλαδή όχι από τα +5V του Arduino). Επίσης πάντα πρέπει να είναι συνδεδεμένη η γείωση του arduino με την γείωση των servo και της εξωτερικής τροφοδοσίας ρεύματος.

Για να χρησιμοποιηθεί ένας servo σε ένα πρόγραμμα arduino πρέπει πρώτα να αρχικοποιηθεί (`Servo ServoName`) και μετά να γίνει attach στο pin που θα συνδέσουμε το pulse του servo (`ServoName.attach(ArduinoPin)`). Για να δώσουμε τον αριθμό των μοιρών που θέλουμε να περισταγεί το servo χρησιμοποιούμε την `write()` (`ServoName.write(NumberOfDegrees)`).

Διαθέσιμες από την βιβλιοθήκη είναι επίσης και οι παρακάτω λειτουργίες:

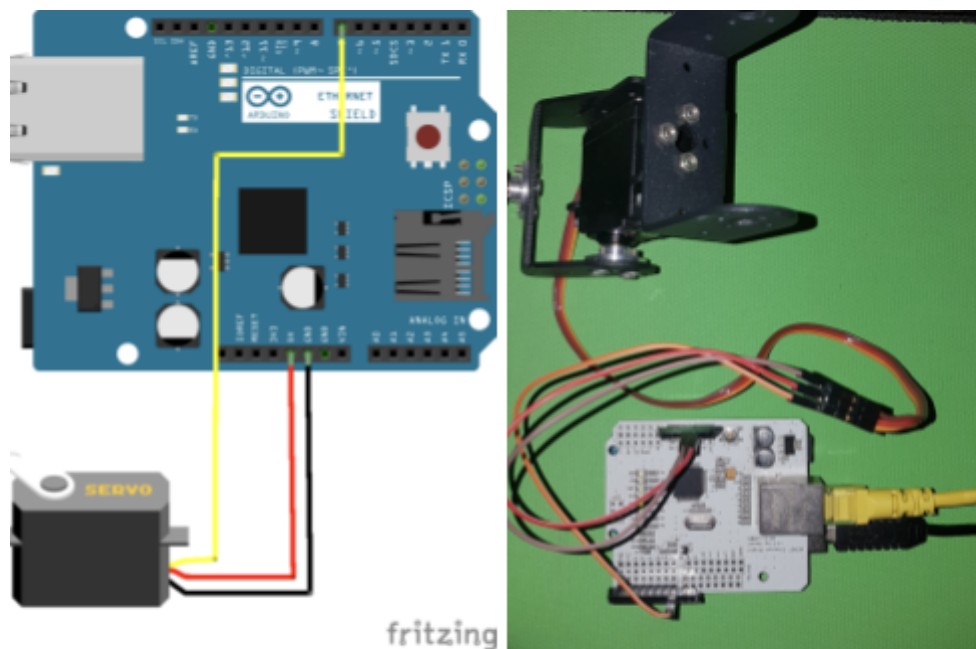
- `writeMicroseconds()`: παρόμοια με την `write`
- `read()`: διαβάζει την θέση που βρίσκεται ο άξονας του servo
- `attached()`: ελέγχει αν το servo έχει γίνει attach
- `detach()`: κάνει detach το servo από το pin του arduino

5.3 Σύνδεση του Ethernet Shield και χειρισμός ενός servo μέσα από διαδικτυακή εφαρμογή

Στο επόμενο βήμα θα προσπαθήσουμε να ελέγξουμε πάλι τον ίδιο servo (του αριστερού μπράτσου του biped robot) από μια διαδικτυακή εφαρμογή. Συγκεκριμένα θα χρησιμοποιήσουμε το Ethernet Shield¹⁸, στο οποίο θα φιλοξενείται μια html¹⁹ σελίδα από την οποία θα δίνουμε εντολές στον servo.

5.3.1 Συνδεσμολογία

Ενώνουμε το Ethernet shield με το Arduino Uno, τοποθετώντας το ακριβώς από πάνω από τον Arduino έτσι ώστε να εφαρμόσουν τα pin του ICSP και τα digital, analog και power. Έπειτα συνδέουμε την γείωση του servo με την γείωση του Arduino, την τροφοδοσία του servo με τα 5V του Arduino και το control του servo με την υποδοχή 7 του Arduino, όλα από τις υποδοχές του Ethernet shield. Επίσης σκαι εδώ ο Arduino τροφοδοτείται κατευθείαν από την USB θύρα που είναι ενωμένη στον υπολογιστή που τρέχουμε το Arduino IDE. Συνδέουμε με ένα UTP καλώδιο το Ethernet shield με το router/gateway.



Εικόνα 5.9 Servo χειρισμός μέσω διαδικτυακής εφαρμογής

¹⁸ Βλέπε [2.3.2 Ethernet Shield](#)

¹⁹ Βλέπε [4.1 HTML](#)

5.3.2 Υλοποίηση σε κώδικα

```

/*
 * Testing servo motor manipulation from a local webpage by
 using Ethernet arduino shield
 * Inspired from:
 https://randomnerdtutorials.com/arduino-webserver-with-an-arduino-ethernet-shield/
 * Custom css created and uploaded to an s3 bucket:
 https://s3-eu-west-1.amazonaws.com/static.cambot.allaboutsadsadmin.com/stylesheets/cambot.css
 */
#include <SPI.h>
#include <Ethernet.h>
#include <Servo.h>

Servo LeftArmServo; // Initialize servo

int pos = 0; // variable for servo position
int leftArmPin = 7; // Servo pulse at pin number 7 of arduino board

// LAN connection initialization
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
byte ip[] = {192, 168, 1, 178};
byte gateway[] = {192, 168, 1, 1};
byte subnet[] = {255, 255, 255, 0};
EthernetServer server(5555); // webserver accessible at 5555
TCP port
String readString;

void setup() {
  Serial.begin(9600); // Initialize connection
  while (!Serial) {; // wait for serial port to connect. Needed for Leonardo only
  }
  LeftArmServo.attach(leftArmPin); // Attach servo motor pin
  // Begin ethernet connection
  Ethernet.begin(mac, ip, gateway, subnet);
  server.begin();
  Serial.print("Server available at: ");
  Serial.println(Ethernet.localIP());
}

void loop() {
  // Create a client connection
  EthernetClient client = server.available();
  if (client) {
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        //read char by char HTTP request
        if (readString.length() < 100) {
          //store characters to string
          readString += c;
          //Serial.print(c); //print to serial monitor for debugging
        }
        //if HTTP request has ended
        if (c == '\n') {
          //Serial.println(readString); //print to serial monitor for debugging
          client.println("HTTP/1.1 200 OK"); //send new page
          client.println("Content-Type: text/html");
          client.println();
          client.println("<HTML>");
          client.println("<HEAD>");
          client.println(
            "<link rel='stylesheet' type='text/css' href='https://s3-eu-west-1.amazonaws.com/static.cambot.allaboutsadsadmin.com/stylesheets/cambot.css' />");
          client.println("<TITLE>Fotag</TITLE>");
          client.println("</HEAD>");
          client.println("<BODY>");
          client.println("<H2>Move Left Arm</H2>");
          client.println("<br />");
          client.println("<a href='/?rotate_left'>Rotate Left</a>");
          client.println("<a href='/?rotate_right'>Rotate Right</a><br />");
          client.println("<p>Created by fotag</p>");
          client.println("<br />");
          client.println("</BODY>");
          client.println("</HTML>");
          delay(1);
          //stopping client
          client.stop();
          //controls the Arduino if you press the buttons
          if (readString.indexOf("?rotate_left") > 0) {
            for (pos = 0; pos < 45; pos += 3) // goes from 0 degrees to 45 degrees
              {
                // in steps of 1 degree
                LeftArmServo.write(pos); // tell servo to go to position in variable 'pos'
                delay(15); // waits 15ms for the servo to reach the position
              }
          }
          if (readString.indexOf("?rotate_right") > 0) {
            for (pos = 45; pos >= 1; pos -= 3) // goes from 45 degrees to 0 degrees
              {
                LeftArmServo.write(pos);
                delay(15);
              }
          }
          //clearing string for next read
          readString = "";
        }
      }
    }
  }
}

```

5.3.3 Συμπεράσματα/Σχόλια

Σε αυτό το βήμα προστέθηκαν δύο βιβλιοθήκες ακόμα. Η SPI.h, που είναι μια βιβλιοθήκη για την σύγχρονη σειριακή επικοινωνία για ανταλλαγή δεδομένων. Στο συγκεκριμένο κύκλωμα χρησιμοποιείται στην σειριακή επικοινωνία uno-ethernet shield και uno-arduino software και η επικοινωνία γίνεται initialize με την μεθοδο *begin()*. Η SPI έχει ακόμα αρκετές μεθόδους διαθέσιμες για την παραμετροποίηση της επικοινωνίας μεταξύ δύο μικροελεγκτων(*SPISettings*), την εκτελεση transactions(*beginTransaction()*, *endTransaction()*), την μεταφορά δεδομένων (*transfer()*) και άλλες.

Επίσης χρησιμοποιήθηκε η Ethernet.h για την διαχειριση του Ethernet shield. Στην αρχή δηλώνονται κάποιες μεταβλητές για την IP (*192.168.1.178*) και την MAC (*DE:AD:BE:EF:FE:ED*) διεύθυνση, το gateway (*192.168.1.1*) και το η μάσκα (*255.255.255.0*) του υποδικτύου που θα ανήκει το Ethernet shield. Με τη βοήθεια της *EthernetServer()* ξεκινάμε ένα server, πρώτα θέτωνατας την πόρτα (*5555*) και μετά ξεκινώντας τον server με την *begin()*.

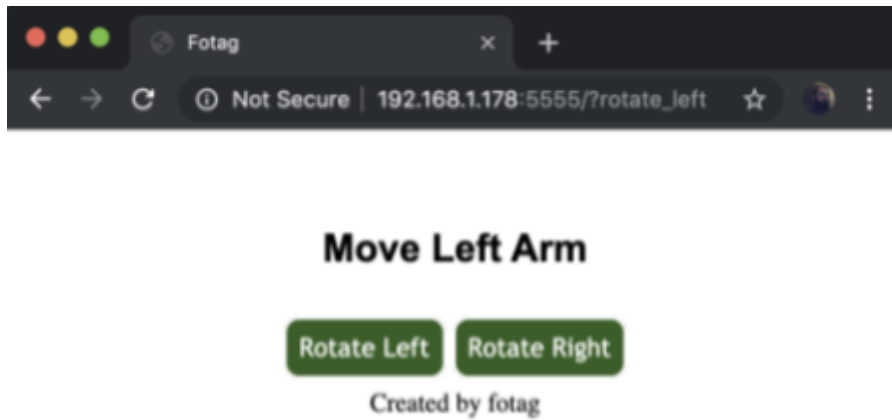
Επειτα δημιουργούμε ένα client αντικείμενο για τις συνδέσεις που θα γίνουν και αρχικοποιούμε την μεταβλητή *c* που θα δέχεται τα HTTP requests και αφού μετατραπεί σε τύπο string αποθηκευεται στην μεταβλητη *readString*. Σε κάθε request, επιστρέφεται μια html σελίδα (Εικόνα 5.7) όπου ο χρήστης έχει την επιλογή μεταξύ δύο υπερσυνδέσμων, του Rotate Left και του Rotate Right. Η μορφοποίηση της html σελίδας, γίνεται με ένα css αρχείο που έχουμε αποθηκεύσει στο Amazon S3²⁰. Αναλόγως της επιλογής καλείται άλλο URI (*/?rotate_left* ή */?rotate_right*), εφόσον το request γίνεται στον ίδιο server (Ethernet shield) που φιλοξενείται η html σελίδα το path είναι σχετικό και δεν χρειάζεται διεύθυνση IP. Το request αυτό αφού παραληφθεί, αποθηκεύει στην *readString* τα δεδομένα. Με την χρήση της *indexOf()*²¹, η οποία ελέγχει για σειρά χαρακτήρων σε μια άλλη σειρά χαρακτήρων και επιστρέφει έναν αριθμό (την θέση που βρίσκεται) ή -1 αν δεν βρει την σειρά, ελέγχουμε αν στο request υπάρχουν τα *?rotate_left* ή *?rotate_right* και αναλόγως στέλνουμε στο servo είτε +45 μοίρες με βήμα 3, είτε -45 μοίρες με βήμα -3.

Για να προσπελάσουμε την ιστοσελίδα από browser, σε υπολογιστή που είναι συνδεδεμένος στο τοπικό δίκτυο και έχει πρόσβαση σε επικοινωνία TCP στην πόρτα 5555, καλούμε την διεύθυνση <http://192.168.1.178:5555>.

Για να στρίψει το μπράτσο δεξιά πατάμε στο “κουμπί/σύνδεσμο” Rotate Right και για να στρίψει αριστερα στο Rotate Left.

²⁰ Βλέπε Παράρτημα Γ. [Amazon S3](#)

²¹ <https://www.arduino.cc/reference/en/language/variables/data-types/string/functions/indexof/>



Εικόνα 5.10 Εφαρμογή διαδικτύου (σελίδα html) για τον χειρισμό ενός από τα servo του biped robot

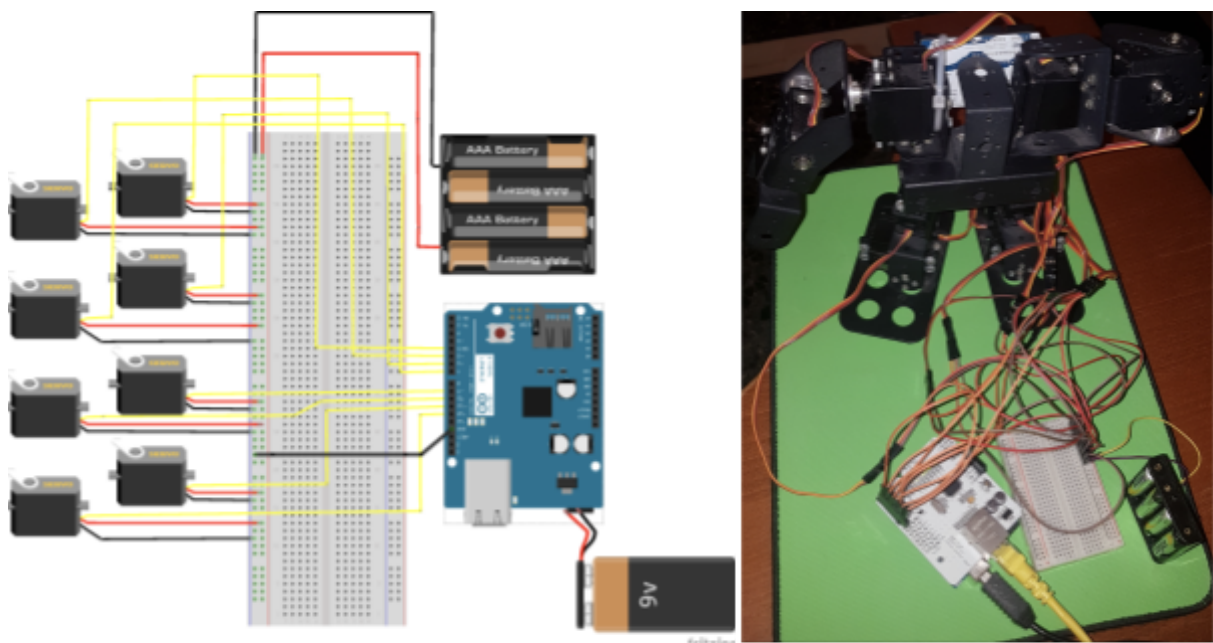
Έτσι χρησιμοποιώντας το URI του HTTP request μπορούμε εύκολα να επικοινωνήσουμε την ενέργεια που θέλουμε να εκτελέσει ο servo μέσω της διαδικτυακής εφαρμογής στον Arduino και έπειτα να μετατραπεί σε ενέργεια του servo.

5.4 Προσθήκη όλων των servo του biped robot και χειρισμός τους από την διαδικτυακή εφαρμογή

Σε αυτό το βήμα θα ενώσουμε με τον Arduino όλα τα servo του ρομπότ και θα επεκτείνουμε την διαδικτυακή εφαρμογή να μπορεί να στείλει εντολές σε όλα τα servo.

5.4.1 Συνδεσμολογία

Ενώνουμε το Ethernet shield με το Arduino Uno, τοποθετώντας το ακριβώς από πάνω από τον Arduino έτσι ώστε να εφαρμόσουν τα pin του ICSP και τα digital, analog και power. Έπειτα σε ένα breadboard, συνδέουμε όλες τις γειώσεις των servo με την γείωση του Arduino και το αρνητικό πόλο της τροφοδοσίας (4xAAA batteries), την τροφοδοσία των servo με το θετικό πόλο των μπαταριών. Τα control των servo με τις ψηφιακές υποδοχές του Arduino (μέσω του ethernet shield), στις θέσεις που φαίνεται στον κώδικα. Επίσης σκαι εδώ ο Arduino τροφοδοτείται κατευθείαν από την USB θύρα που είναι ενωμένη στον υπολογιστή που τρέχουμε το Arduino IDE, μετά τον προγραμματισμό χρησιμοποιήθηκε μπαταρία 9V και αποσυνδέθηκε το USB. Συνδέουμε με ένα UTP καλώδιο το Ethernet shield με το router/gateway.



Εικόνα 5.11 Σύνδεση και των 8 servo στον Arduino Uno μέσω του Ethernet Shield

5.4.2 Υλοποίηση σε κώδικα

```

#include <SPI.h>
#include <Ethernet.h>
#include <Servo.h>

// Initialize all servos
Servo leftAnkleServo;
Servo leftLegServo;
Servo rightAnkleServo;
Servo rightLegServo;
Servo leftElbowServo;
Servo leftShoulderServo;
Servo rightElbowServo;
Servo rightShoulderServo;

// Initialize all variables
int pos = 0; // variable will be used for servos position

// Pins to attach servos
int leftAnklePin = 6;
int leftLegPin = 9;
int rightAnklePin = 10;
int rightLegPin = 11;
int leftElbowPin = 4;
int leftShoulderPin = 5;
int rightElbowPin = 2;
int rightShoulderPin = 3;

// Servos initial positions
int leftAnklePos = 65;
int leftLegPos = 80;
int rightAnklePos = 80;
int rightLegPos = 120;
int leftElbowPos = 105;
int leftShoulderPos = 5;
int rightElbowPos = 10;
int rightShoulderPos = 100;

// Network related variables
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
IPAddress ip(192, 168, 1, 178); // static ip address for the ethernet
shield
EthernetServer server(80); // server port for shield's web server
String readString; // string variable to read HTTP requests

void setup() {

  // Attach all servos
  // Feet
  leftAnkleServo.attach(leftAnklePin);
  leftLegServo.attach(leftLegPin);
  rightAnkleServo.attach(rightAnklePin);
  rightLegServo.attach(rightLegPin);

  // Hands
  leftElbowServo.attach(leftElbowPin);
  leftShoulderServo.attach(leftShoulderPin);
  rightElbowServo.attach(rightElbowPin);
  rightShoulderServo.attach(rightShoulderPin);

  // Initialize all servos
  // Feet
  leftAnkleServo.write(leftAnklePos);
  leftLegServo.write(leftLegPos);
  rightAnkleServo.write(rightAnklePos);
  rightLegServo.write(rightLegPos);

  // Hands
  leftElbowServo.write(leftElbowPos);
  leftShoulderServo.write(leftShoulderPos);
  rightElbowServo.write(rightElbowPos);
  rightShoulderServo.write(rightShoulderPos);

  // Initialize serial connection
  Serial.begin(9600);
  while (!Serial) { ; // wait for serial port to connect.
}

  // Begin ethernet connection
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.print("Server available at: ");
  Serial.println(Ethernet.localIP());
}

void loop() {
  EthernetClient client = server.available();
  if (client) {
    while (client.connected()) {
      if (client.available()) {
        char reqChar = client.read(); //save request in variable

        //read char by char HTTP request
        if (readString.length() < 100) {
          //store characters to string
          readString += reqChar;
          //Serial.print(reqChar); // ONLY for DEBUG
        }

        //if HTTP request has ended
        if (reqChar == '\n') {
          //Serial.println(readString); // ONLY for DEBUG

          delay(1);
          client.println("HTTP/1.1 200 OK");
          client.println("Connection: close");
          // this is the CORS header. Very important. Google CORS.
          client.println("Access-Control-Allow-Origin: *");
          client.println();
          //stopping client
          client.stop();

          // Right side
          // Hands
          if (readString.indexOf("?le_rotate_left") > 0) {
            for (pos = 0; pos < 45; pos += 5) // goes from 0
degrees to 45 degrees
            {
              // in steps of 1 degree
              leftElbowServo.write(pos); // tell servo to go to
position in variable 'pos'
              delay(50); // waits 15ms for the servo to
reach the position
            }
          }
          if (readString.indexOf("?le_rotate_up") > 0) {
            for (pos = 45; pos >= 1; pos -= 5) // goes from 45
degrees to 0 degrees
            {
              leftElbowServo.write(pos);
              delay(50);
            }
          }
          ..... Ολόκληρος ο κώδικας στο αρχείο
          biiped_via_http/all_servos_via_http/biped.ino και στο
          Παράρτημα Ζ.
        }
      }
    }
  }
}

```

5.4.3 Συμπεράσματα/Σχόλια

Για την συμπερίληψη της διαχείρισης όλων το servo, έγινε προσπάθεια προσθήκης τους στο υπάρχον πρόγραμμα. Αρχικοποιώντας όλα τα servo, κανοντας τα attach και δίνοντας τους αρχική θέση. Για την διαδικτυακή εφαρμογή έγινε προσθήκη στην σελίδα των απαραίτητων κουμπιών για την διαχείριση όλων τον servo. Όταν έγινε μεταφόρτωση του προγράμματος στον arduino υπήρξε σφάλμα και δεν μπορούσε να ολοκληρωθεί. Συγκεκριμένα η μνήμη του Arduino δεν επαρκούσε για αποθηκευσει τις μεταβλητες και τα δεδομένα για την html σελιδα που θα σερβιρε το Ethernet Shield καθώς είχε μεγαλώσει αρκετά.

Ετσι αφαιρέθηκε η html σελίδα από τον arduino κώδικα και στην θέση της έμεινε ένας webserver που ακούει στην πόρτα 80 και απαντάει με 200 OK (http code) όταν παραλαμβάνει ένα request για να ξέρει ο browser ;oti το request παραλήφθηκε. Η υλοποίηση για τις ενέργειες των servo παραμένει η ίδια με το προηγούμενο βήμα, απλά δημιουργήθηκαν τόσα if statements όσα και τα servos και οι επιθυμητες ενεργειες που θέλαμε να γίνουν (στροφή αριστερά-δεξιά ή πάνω-κάτω).

Στην html σελίδα χρησιμοποιήθηκε η συλλογή ανοιχτού κώδικα Bootstrap 4²². Δημιουργήθηκε header και footer για την καλύτερη παρουσίαση της σελίδας (χρησιμοποιήθηκε το λογότυπο του ΠΑΔΑ). Για κάθε ενέργεια των servo δημιουργήθηκαν δύο κουμπιά (buttons) και για κάθε servo μια κάρτα (card) που συμπεριλαμβάνει τα δύο κουμπιά.

Το κάθε κουμπί στην ενέργεια κλικ (*ng-click*) καλεί μια συναρτηση angularjs αναλόγως την κίνηση που θέλουμε να πραγματοποιήσουμε και ποιο servo θα την εκτελέσει. Έχουν υλοποιηθεί δύο λειτουργίες (functions), με το angularjs, για κάθε servo. Έχουν ονομαστεί αναλόγως του servo που ελέγχουν και της κίνησης που εκτελούν (για παράδειγμα *leftElbowRotateLeft*, *leftElbowRotateRight*, etc). Όταν κληθεί αυτή η function τότε εκτελεί ένα http get request στον webserver του arduino με παράμετρο το αντίστοιχο της επιθυμητης κίνησης λεκτικό. Για παράδειγμα αν θέλουμε να κουνηθει αριστερα ο αριστερος αγκώνας και πατήσουμε το πλήκτρο 'Move left' της κάρτας *Left Elbow*, θα εκτελεστεί η *leftElbowRotateLeft* function και θα κληθει ο webserver του arduino με το παρακάτω URI: http://192.168.1.178:80/?le_rotate_left. Ο arduino κωδικας θα αντιληφθει το *'?le_rotate_left'* και θα ζητήσει από το *leftElbowServo* σερβο να μετακινηθει 45 μοιρες με βήμα 5.

Το angularjs module που δημιουργήθηκε και συμπεριλήφθηκε στην html σελίδα είναι το *readArduinoApp* και ο controller που υλοποιεί τις λειτουργίες για τα servo είναι ο *ReadArduinoController*. Στο body html tag ενημερώνουμε για το module `<body ng-app="readArduinoApp">` και στο div tag που συμπεριλαμβάνονται όλα τα

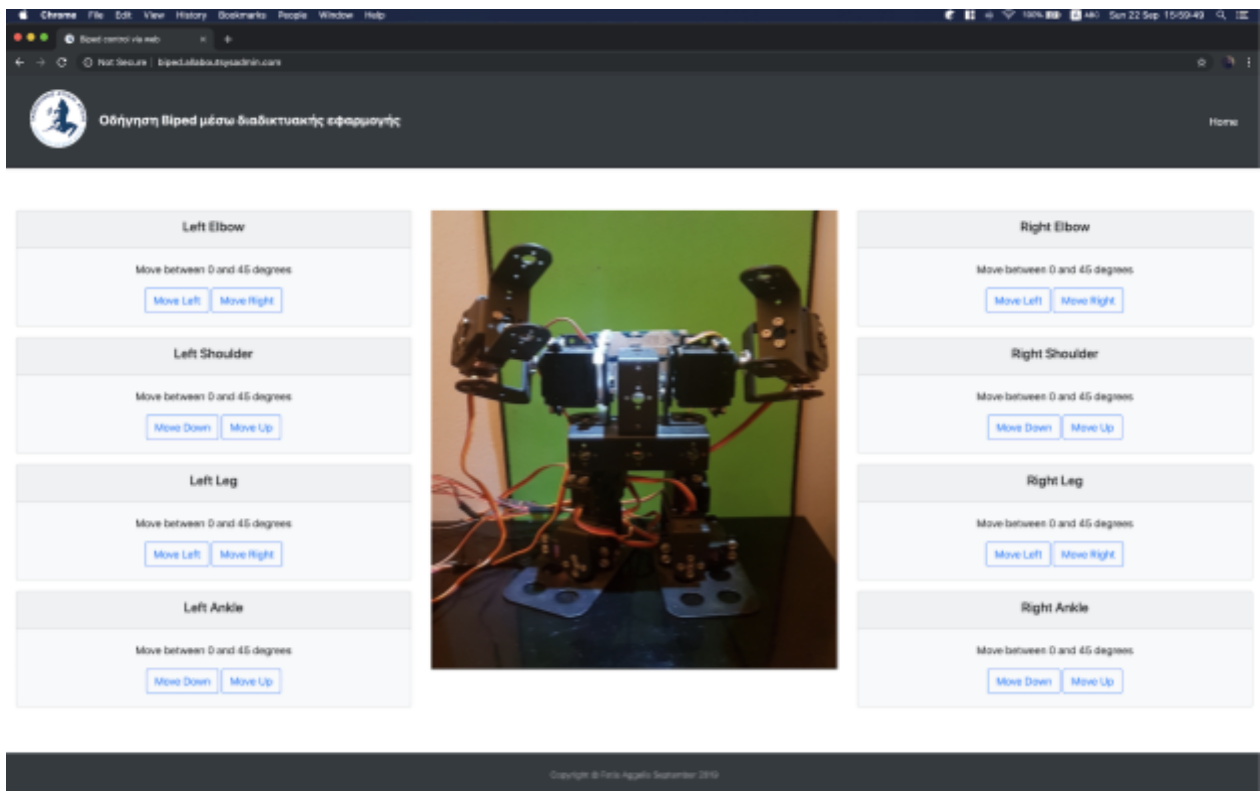
²² Βλέπε [Παράρτημα Δ. Bootstrap framework](#)

κουμπια που καλούν τις λειτουργίες τον controller που τις περιλαμβάνει `<div class="row" ng-controller="ReadArduinoController">`.

Η html σελίδα μεταφορτώθηκε σε ένα bucket στο Amazon S3 το οποίο παραμετροποιήθηκε κατάλληλα για να λειτουργεί σαν webserver στατικών ιστοσελίδων και ήταν διαθέσιμη για συγκεκριμένο χρονικό διάστημα στο url: <http://biped.allaboutsystadmin.com/>

Η σελίδα αυτή έχει στατικά μέσα στον κωδικά js την IP address του arduino, οπότε αν και είναι διαθέσιμη στο internet τα requests θα καταλήξουν στον arduino μόνο αν επισκεφθείτε την ιστοσελίδα από browser που είναι στο ίδιο τοπικό δίκτυο με τον arduino.

Στο μέλλον αυτή η σελίδα μπορεί να φιλοξενηθεί οπουδήποτε ακόμα και στον υπολογιστή που είναι ο browser που θα την χρησιμοποιήσουμε.



Εικόνα 5.12 Οθόνη διαδικτυακής εφαρμογής για την διαχείριση των οκτώ servo του biped robot

6. Συμπεράσματα και προοπτικές

6.1 Συμπεράσματα/Επίλογος πτυχιακής εργασίας

Μετά λοιπόν από την ολοκλήρωση όλων των βημάτων καταφέραμε να ελέγξουμε τα σέρβο του δίποδου ρομπότ από την προηγούμενη πτυχιακή εργασία από μια διαδικτυακή εφαρμογή (html σελίδα τελικά φιλοξενούμενη στο Amazon S3 που συμπεριλαμβάνει javascript script στηριζόμενο στο AngularJS framework) χρησιμοποιώντας την πλακέτα arduino uno και το ethernet shield από το τοπικό δίκτυο.

Στην διάρκεια της εκπόνησης της εργασίας αποκτήθηκαν ή επεκτάθηκαν γνώσεις γύρω από τις διαδικτυακές εφαρμογες, τους σερβοκινητήρες και την πλατφόρμα arduino.

6.2 Επόμενα βήματα

Πιθανά επόμενα βήματα για την παρούσα πτυχιακή εργασία μπορεί να είναι:

- Βελτιστοποίηση της υπάρχουσας, χωρίς προσθήκη νέας λειτουργίας: βελτιστοποίηση arduino και javascript κώδικα ώστε να μην γίνεται επανάληψη του ίδιου κώδικα αλλά αν εκτελούνται δομές επανάληψης ή υλοποιηθούν μέθοδοι που θα δέχονται σαν ορίσματα τα δεδομένα που αλλάζουν (σέρβο, κίνηση), παραμετροποίηση της IP διεύθυνσης του arduino στην html ώστε να δίνει ο χρήστης την IP διεύθυνση του arduino, καλύτερος έλεγχος συμβολοσειράς στον arduino.
- Προσθήκη κάμερας, μικροελεγκτή και παροχή ζωντανής εικόνας στην διαδικτυακή εφαρμογή
- Υλοποίηση ανθρώπινων κινήσεων από το ρομπότ (περπάτημα, χρήση αντικειμένων)
- Δημιουργία Διεπαφής προγραμματισμού εφαρμογών (API) που θα δέχεται τα αιτήματα από διαφορετικές διεπαφές χρήστη, θα τα αποθηκεύει σε βάση δεδομένων και ο kvndiakaw arduino θα αναζητά μέσω του API διαθέσιμες ενέργειες.

Παράρτημα

Στα πλαίσια της πτυχιακής εργασίας χρησιμοποιήθηκαν και κάποια άλλα εργαλεία για συγκεκριμένες διαδικασίες, όπως τον σχεδιασμό των κυκλωμάτων, την αποθήκευση και διαχείριση των αρχείων του κώδικα και την αποθήκευση των CSS αρχείων στο internet, περιβάλλοντα ανοικτού λογισμικού για html/css, και άλλα. Στο παράρτημα θα κάνουμε μια συνοπτική περιγραφή τους και θα συμπεριλάβουμε και τα αρχεία κώδικα για το τελευταίο βήμα που ήταν αρκετά μεγάλα για να τα συμπεριλάβουμε στο κείμενο της ανάλυσης.

A. [Fritzing](#)

Το Fritzing είναι μια πρωτοβουλία ανοιχτού κώδικα για την ανάπτυξη ερασιτεχνικού λογισμικού CAD για το σχεδιασμό του ηλεκτρονικού εξοπλισμού, για να υποστηρίξει τους σχεδιαστές και τους καλλιτέχνες στον πειραματισμό με ένα πρωτότυπο για την δημιουργία ενός πιο μόνιμου κυκλώματος. Αναπτύχθηκε στο Πανεπιστήμιο των εφαρμοσμένων επιστημών του Πότσταμ.

B. [Gitlab](#)

GitLab είναι μια web-based πλατφόρμα. Μια από τις διαθέσιμες υπηρεσίες είναι η δημιουργία αποθετηρίου κώδικα (Git). Το Git (/git/) είναι ένα σύστημα ελέγχου εκδόσεων (λέγεται και σύστημα ελέγχου αναθεωρήσεων ή σύστημα ελέγχου πηγαίου κώδικα) με έμφαση στην ταχύτητα, στην ακεραιότητα των δεδομένων και στην υποστήριξη για κατανεμημένες μη γραμμικές ροές εργασίας. Το Git σχεδιάστηκε και αναπτύχθηκε αρχικά από τον Λίνους Τόρβαλντς για τη ανάπτυξη του πυρήνα Linux το 2005 και έχει γίνει από τότε το πιο διαδεδομένο σύστημα ελέγχου εκδόσεων για ανάπτυξη λογισμικού. Άλλες υπηρεσίες που παρέχει το Gitlab είναι η παροχή wiki, CI/CD pipelines χρησιμοποιώντας open-source άδεια. Το λογισμικό δημιουργήθηκε από τον Dmitry Zaporozhets και Valery Sizov, και χρησιμοποιείται από αρκετές μεγάλες εταιρείες τεχνολογίας όπως οι IBM, Sony, Jülich Research Center, NASA, Alibaba, Oracle, Invincea, O'Reilly Media, Leibniz-Rechenzentrum (LRZ), CERN, Ευρωπαϊκό XFEL, GNOME Foundation, Boeing, Autodesk και SpaceX.

Η σημερινή τεχνολογία στοίβα του λογισμικού που περιλαμβάνει Go, Ruby on Rails και Vue.js. Το λογισμικό ακολουθεί ένα μοντέλο ανάπτυξης ανοιχτού πυρήνα, όπου η βασική λειτουργικότητα απελευθερώνεται με άδεια ανοικτού κώδικα (MIT), ενώ η πρόσθετη λειτουργικότητα είναι υπό άδειες ιδιοκτησίας.

Γ. [Amazon S3](#)

Η υπηρεσία απλής αποθήκευσης Amazon (Amazon S3) είναι μια υπηρεσία αποθήκευσης αντικειμένων που προσφέρει κορυφαία κλιμάκωση, διαθεσιμότητα δεδομένων, ασφάλεια και απόδοση. Αυτό σημαίνει ότι οι πελάτες όλων των μεγεθών και βιομηχανιών μπορούν να το χρησιμοποιήσουν για την αποθήκευση και προστασία οποιουδήποτε όγκου δεδομένων για μια σειρά περιπτώσεων χρήσης, όπως ιστότοποι, εφαρμογές για κινητά, backup και αποκατάσταση, αρχεία, επιχειρησιακές εφαρμογές, συσκευές IoT και μεγάλες αναλύσεις δεδομένων. Το Amazon S3 παρέχει εύχρηστες λειτουργίες διαχείρισης, ώστε να μπορείτε να οργανώνετε τα δεδομένα σας και να διαμορφώνετε λεπτές ρυθμίσεις ελέγχου πρόσβασης για να ικανοποιείτε τις συγκεκριμένες επιχειρηματικές, οργανωτικές και απαιτήσεις συμμόρφωσης. Το Amazon S3 έχει σχεδιαστεί για αντοχή 99,999999999% (11 9) και αποθηκεύει δεδομένα για εκατομμύρια εφαρμογές για επιχειρήσεις σε όλο τον κόσμο.

Δ. [Bootstrap framework](#)

Το Bootstrap είναι μια συλλογή εργαλείων ανοιχτού κώδικα (Ελεύθερο λογισμικό) για τη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Περιέχει HTML και CSS για τις μορφές τυπογραφίας, κουμπιά πλοήγησης και άλλων στοιχείων του περιβάλλοντος, καθώς και προαιρετικές επεκτάσεις JavaScript. Είναι το πιο δημοφιλές πρόγραμμα στο GitHub και έχει χρησιμοποιηθεί από τη NASA και το MSNBC, μεταξύ άλλων.

Ε. [AngularJS](#)

Το AngularJS είναι βασισμένο σε JavaScript πλαίσιο front-end ανοιχτού κώδικα που υποστηρίζεται κυρίως από την Google και από μια κοινότητα ατόμων και εταιρειών για την αντιμετώπιση πολλών από τις προκλήσεις που αντιμετωπίζουν στην ανάπτυξη εφαρμογών μιας σελίδας. Στόχος του είναι να απλοποιήσει τόσο την ανάπτυξη όσο και τη δοκιμή τέτοιων εφαρμογών, παρέχοντας ένα πλαίσιο για αρχιτεκτονικές μοντέλου-προβολέα -ελεγκτή (MVC) και μοντέλο προβολής-προβολής (MVVM), μαζί με τα συστατικά που χρησιμοποιούνται συνήθως σε πλούσιες εφαρμογές Διαδικτύου .

Το AngularJS είναι το τμήμα frontend της στοίβας MEAN, που αποτελείται από τη βάση δεδομένων MongoDB , το πλαίσιο εξυπηρετητή ιστού Express.js , το ίδιο το Angular.js και το περιβάλλον χρόνου εκτέλεσης του διακομιστή Node.js.

Z. Κώδικας Arduino και HTML σελίδα για τον χειρισμό όλων των servo

biped.ino

```

/*
 * This program will prepare 8 servos and the ethernet shield in order to
 have network connectivity
 * It will start a web server on Ethernet Shield (port 80) which listen to
 requests and respond with 200 OK
 * Then on every request will receive it will check in the URI for specific
 strings in order to send actions to
 * specific servos. Right now it will only increase and decrease degrees
 between 0 and 45 with a not recursive and reusable
 * way of coding for all servos.
 *
 *
 * IMPORTANT NOTE Sides (left/right) when looking biped from
 behind
 * Inspired from https://playground.arduino.cc/Code/WebServerJson/
 */

#include <SPI.h>
#include <Ethernet.h>
#include <Servo.h>

// Initialize all servos
Servo leftAnkleServo;
Servo leftLegServo;
Servo rightAnkleServo;
Servo rightLegServo;
Servo leftElbowServo;
Servo leftShoulderServo;
Servo rightElbowServo;
Servo rightShoulderServo;

// Initialize all variables
int pos = 0; // variable will be used for servos position

// Pins to attach servos
int leftAnklePin = 6;
int leftLegPin = 9;
int rightAnklePin = 10;
int rightLegPin = 11;
int leftElbowPin = 4;
int leftShoulderPin = 5;
int rightElbowPin = 2;
int rightShoulderPin = 3;

// Servos initial positions
int leftAnklePos = 65;
int leftLegPos = 80;
int rightAnklePos = 80;
int rightLegPos = 120;
int leftElbowPos = 105;
int leftShoulderPos = 5;
int rightElbowPos = 10;
int rightShoulderPos = 100;

// Network related variables
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
IPAddress ip(192, 168, 1, 178); // static ip address for the ethernet
shield
EthernetServer server(80); // server port for shield's web server
String readString; // string variable to read HTTP requests

void setup() {

  // Attach all servos
  // Feet
  leftAnkleServo.attach(leftAnklePin);
  leftLegServo.attach(leftLegPin);
  rightAnkleServo.attach(rightAnklePin);

  rightLegServo.attach(rightLegPin);

  // Hands
  leftElbowServo.attach(leftElbowPin);
  leftShoulderServo.attach(leftShoulderPin);
  rightElbowServo.attach(rightElbowPin);
  rightShoulderServo.attach(rightShoulderPin);

  // Initialize all servos
  // Feet
  leftAnkleServo.write(leftAnklePos);
  leftLegServo.write(leftLegPos);
  rightAnkleServo.write(rightAnklePos);
  rightLegServo.write(rightLegPos);

  // Hands
  leftElbowServo.write(leftElbowPos);
  leftShoulderServo.write(leftShoulderPos);
  rightElbowServo.write(rightElbowPos);
  rightShoulderServo.write(rightShoulderPos);

  // Initialize serial connection
  Serial.begin(9600);
  while (!Serial) { ; // wait for serial port to connect.
}

  // Begin ethernet connection
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.print("Server available at: ");
  Serial.println(Ethernet.localIP());
}

void loop() {
  EthernetClient client = server.available();
  if (client) {
    while (client.connected()) {
      if (client.available()) {
        char reqChar = client.read(); //save request in variable

        //read char by char HTTP request
        if (readString.length() < 100) {
          //store characters to string
          readString += reqChar;
          //Serial.print(reqChar); // ONLY for DEBUG
        }

        //if HTTP request has ended
        if (reqChar == '\n') {
          //Serial.println(readString); // ONLY for DEBUG

          delay(1);
          client.println("HTTP/1.1 200 OK");
          client.println("Connection: close");
          // this is the CORS header. Yery important. Google CORS.
          client.println("Access-Control-Allow-Origin: *");
          client.println();
          //stopping client
          client.stop();

          // Right side
          // Hands
          if (readString.indexOf("?le_rotate_left") > 0) {
            for (pos = 0; pos < 45; pos += 5) // goes from 0
degrees to 45 degrees
            {
              // in steps of 1 degree
              leftElbowServo.write(pos); // tell servo to go to
position in variable 'pos'
              delay(50); // waits 15ms for the servo to
reach the position
            }
          }
        }
      }
    }
  }
}

```



```

    }
}
if (readString.indexOf("?le_rotate_up") > 0) {
    for (pos = 45; pos >= 1; pos -= 5) // goes from 45
degrees to 0 degrees
    {
        leftElbowServo.write(pos);
        delay(50);
    }
}
if (readString.indexOf("?ls_rotate_down") > 0) {
    for (pos = 0; pos < 45; pos += 5)
    {
        leftShoulderServo.write(pos);
        delay(50);
    }
}
if (readString.indexOf("?ls_rotate_right") > 0) {
    for (pos = 45; pos >= 1; pos -= 5)
    {
        leftShoulderServo.write(pos);
        delay(50);
    }
}

// Feet
if (readString.indexOf("?la_rotate_up") > 0) {
    for (pos = 0; pos < 45; pos += 5)
    {
        leftAnkleServo.write(pos);
        delay(50);
    }
}
if (readString.indexOf("?la_rotate_down") > 0) {
    for (pos = 45; pos >= 1; pos -= 5)
    {
        leftAnkleServo.write(pos);
        delay(50);
    }
}
if (readString.indexOf("?ll_rotate_left") > 0) {
    for (pos = 0; pos < 45; pos += 5)
    {
        leftLegServo.write(pos);
        delay(50);
    }
}
if (readString.indexOf("?ll_rotate_right") > 0) {
    for (pos = 45; pos >= 1; pos -= 5)
    {
        leftLegServo.write(pos);
        delay(50);
    }
}

// Right side

// Hands
if (readString.indexOf("?re_rotate_left") > 0) {
    for (pos = 0; pos < 45; pos += 5)
    {
        rightElbowServo.write(pos);
        delay(50);
    }
}

}
}
}
}
}
}
}
}
if (readString.indexOf("?re_rotate_up") > 0) {
    for (pos = 45; pos >= 1; pos -= 5)
    {
        rightElbowServo.write(pos);
        delay(50);
    }
}
if (readString.indexOf("?rs_rotate_down") > 0) {
    for (pos = 0; pos < 45; pos += 5)
    {
        rightShoulderServo.write(pos);
        delay(50);
    }
}
if (readString.indexOf("?rs_rotate_right") > 0) {
    for (pos = 45; pos >= 1; pos -= 5)
    {
        rightShoulderServo.write(pos);
        delay(50);
    }
}

// Feet
if (readString.indexOf("?ra_rotate_up") > 0) {
    for (pos = 0; pos < 45; pos += 5)
    {
        rightAnkleServo.write(pos);
        delay(50);
    }
}
if (readString.indexOf("?ra_rotate_down") > 0) {
    for (pos = 45; pos >= 1; pos -= 5)
    {
        rightAnkleServo.write(pos);
        delay(50);
    }
}
if (readString.indexOf("?rl_rotate_left") > 0) {
    for (pos = 0; pos < 45; pos += 5)
    {
        rightLegServo.write(pos);
        delay(50);
    }
}
if (readString.indexOf("?rl_rotate_right") > 0) {
    for (pos = 45; pos >= 1; pos -= 5)
    {
        rightLegServo.write(pos);
        delay(50);
    }
}

//clearing string for next read
readString = "";

```

biped.html

```

<HTML>
<HEAD>
  <!--
    In case will be reused in functions of script below, we
    have to change appropriately the ip address of
    Arduino ethernet shield!
    TODO: In AngularJS script, take Arduino IP address as
    variable, ask user to add it in a filed at the page
    TODO: Make AngularJS functions reusable and give
    direction and part as parameters
  -->
  <meta charset="UTF-8">
  <title>Biped control via web</title>
  <link rel="shortcut icon" type="image/x-icon"
href="https://www.uniwa.gr/wp-content/uploads/2018/11/logo-pada.png">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
  <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
  <nav class="navbar navbar-expand-lg py-3 navbar-dark bg-dark shadow-sm">
    <div class="container-fluid">
      <a href="#" class="navbar-brand">
        
        <span class="font-weight-bold">Οδήγηση Biped
μέσω διαδικτυακής εφαρμογής</span>
      </a>
      <button type="button" data-toggle="collapse"
data-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation"
class="navbar-toggler"><span
class="navbar-toggler-icon"></span></button>
      <div id="navbarSupportedContent"
class="collapse navbar-collapse">
        <ul class="navbar-nav ml-auto">
          <li class="nav-item active"><a href="#"
class="nav-link">Home <span
class="sr-only">(current)</span></a></li>
        </ul>
      </div>
    </div>
  </nav>

  <script type="text/javascript">
    var app = angular.module("readArduinoApp", []);

    app.controller("ReadArduinoController", [ "$scope",
$!http,
function($scope, $http) {
  $scope.ok = false;
  $scope.fetching = false;
  $scope.status = "no data yet";

  <!--
  Left Side
  -->

```

```

$scope.leftElbowRotateLeft = function() {
  $scope.fetching = true;

  $http.get('http://192.168.1.178:80/?le_rotate_left');
};

$scope.leftElbowRotateRight = function() {
  $scope.fetching = true;

  $http.get('http://192.168.1.178:80/?le_rotate_right');
};

$scope.leftShoulderRotateDown = function() {
  $scope.fetching = true;

  $http.get('http://192.168.1.178:80/?ls_rotate_down');
};

$scope.leftShoulderRotateUp = function() {
  $scope.fetching = true;

  $http.get('http://192.168.1.178:80/?ls_rotate_up');
};

$scope.leftLegRotateLeft = function() {
  $scope.fetching = true;

  $http.get('http://192.168.1.178:80/?ll_rotate_left');
};

$scope.leftLegRotateRight = function() {
  $scope.fetching = true;

  $http.get('http://192.168.1.178:80/?ll_rotate_right');
};

$scope.leftAnkleRotateDown = function() {
  $scope.fetching = true;

  $http.get('http://192.168.1.178:80/?la_rotate_down');
};

$scope.leftAnkleRotateUp = function() {
  $scope.fetching = true;

  $http.get('http://192.168.1.178:80/?la_rotate_up');
};

  <!--
  Right Side
  -->

  $scope.rightElbowRotateLeft = function() {
  $scope.fetching = true;

  $http.get('http://192.168.1.178:80/?le_rotate_left');
};

  $scope.rightElbowRotateRight = function() {
  $scope.fetching = true;

  $http.get('http://192.168.1.178:80/?le_rotate_right');
};

```

```

$scope.rightShoulderRotateDown = function()
    $scope.fetching = true;
$http.get('http://192.168.1.178:80/?ls_rotate_down');
    $scope.rightShoulderRotateUp = function()
    $scope.fetching = true;
$http.get('http://192.168.1.178:80/?ls_rotate_up');
    $scope.rightLegRotateLeft = function()
    $scope.fetching = true;
$http.get('http://192.168.1.178:80/?ll_rotate_left');
    $scope.rightLegRotateRight = function()
    $scope.fetching = true;
$http.get('http://192.168.1.178:80/?ll_rotate_right');
    $scope.rightAnkleRotateDown = function()
    $scope.fetching = true;
$http.get('http://192.168.1.178:80/?la_rotate_down');
    $scope.rightAnkleRotateUp = function()
    $scope.fetching = true;
$http.get('http://192.168.1.178:80/?la_rotate_up');
    ]
);
</script>
</HEAD>
<BODY ng-app="readArduinoApp">

```

```

<div class="row" ng-controller="ReadArduinoController">
  <div class="col mt-5">
    <div class="card bg-light text-center mb-3 ml-3 mt-3">
      <div class="card-header text-center"><h5>Left Elbow</h5></div>
      <div class="card-body">
        <p class="card-text">Move between 0 and 45 degrees</p>
        <button type="button" class="btn btn-outline-primary" ng-click="leftElbowRotateLeft()">Move Left</button>
        <button class="btn btn-outline-primary" ng-click="leftElbowRotateRight()">Move Right</button>
      </div>
    </div>
    <div class="card bg-light text-center mb-3 ml-3">
      <div class="card-header text-center"><h5>Left Shoulder</h5></div>
      <div class="card-body">

```

```

<p class="card-text">Move between 0 and 45 degrees</p>
        <button type="button" class="btn btn-outline-primary" ng-click="leftShoulderRotateDown()">Move Down</button>
        <button class="btn btn-outline-primary" ng-click="leftShoulderRotateUp()">Move Up</button>
      </div>
    </div>
    <div class="card bg-light text-center mb-3 ml-3">
      <div class="card-header text-center"><h5>Left Leg</h5></div>
      <div class="card-body">
        <p class="card-text">Move between 0 and 45 degrees</p>
        <button type="button" class="btn btn-outline-primary" ng-click="leftLegRotateLeft()">Move Left</button>
        <button class="btn btn-outline-primary" ng-click="leftLegRotateRight()">Move Right</button>
      </div>
    </div>
    <div class="card bg-light text-center mb-3 ml-3">
      <div class="card-header text-center"><h5>Left Ankle</h5></div>
      <div class="card-body">
        <p class="card-text">Move between 0 and 45 degrees</p>
        <button type="button" class="btn btn-outline-primary" ng-click="leftAnkleRotateDown()">Move Down</button>
        <button class="btn btn-outline-primary" ng-click="leftAnkleRotateUp()">Move Up</button>
      </div>
    </div>
  </div>
</div>
<div class="col mt-5">
  
</div>

```

```

<div class="col mt-5">
  <div class="card bg-light text-center mb-3 mr-3 mt-3">
    <div class="card-header text-center"><h5>Right Elbow</h5></div>
    <div class="card-body">
      <p class="card-text">Move between 0 and 45 degrees</p>
      <button type="button" class="btn btn-outline-primary" ng-click="rightElbowRotateLeft()">Move Left</button>
      <button class="btn btn-outline-primary" ng-click="rightElbowRotateRight()">Move Right</button>
    </div>
  </div>
  <div class="card bg-light text-center mb-3 mr-3">
    <div class="card-header text-center"><h5>Right Shoulder</h5></div>
    <div class="card-body">

```

```

<p class="card-text">Move between 0 and 45
degrees</p>
<button type="button" class="btn
btn-outline-primary"
ng-click="rightShoulderRotateDown()">Move
Down</button> <button class="btn btn-outline-primary"
ng-click="rightShoulderRotateUp()">Move Up</button>
</div>
</div>
<div class="card bg-light text-center mb-3 mr-3">
<div class="card-header text-center"><h5>Right
Leg</h5></div>
<div class="card-body">
<p class="card-text">Move between 0 and 45
degrees</p>
<button type="button" class="btn
btn-outline-primary" ng-click="rightLegRotateLeft()">Move
Left</button> <button class="btn btn-outline-primary"
ng-click="rightLegRotateRight()">Move Right</button>
</div>
</div>
<div class="card bg-light text-center mb-3 mr-3">

```

```

<div class="card-header text-center"><h5>Right
Ankle</h5></div>
<div class="card-body">
<p class="card-text">Move between 0 and 45
degrees</p>
<button type="button" class="btn
btn-outline-primary"
ng-click="rightAnkleRotateDown()">Move Down</button>
<button class="btn btn-outline-primary"
ng-click="rightAnkleRotateUp()">Move Up</button>
</div>
</div>
</div>
<div id="sticky-footer" class="py-4 bg-dark
text-white-50 fixed-bottom">
<div class="container-fluid text-center">
<small>Copyright © Fotis Aggelis September
2019</small>
</div>
</footer>
</BODY>
</HTML>

```

Βιβλιογραφικές αναφορές και σύνδεσμοι

Βίντεο προηγούμενης πτυχιακής εργασίας	https://www.youtube.com/watch?v=BBKsu3uqPwE
Μικροελεγκτές	https://el.wikipedia.org/wiki/Μικροελεγκτής http://www.eln.teilam.gr/sites/default/files/Lesson03.pdf http://dallas-ic.com/images/rf_soc/fig1-3.gif
Arduino	https://el.wikipedia.org/wiki/Arduino https://www.arduino.cc/en/Guide/Introduction http://enigma.freebsdworld.gr/sites/default/files/enigma-part2.pdf http://online-seminaria.gr/eclass/modules/document/file.php/EDU102/Εισαγωγή%20και%20θεωρία.pdf https://processing.org/
Arduino programming	https://www.arduino.cc/reference/en/ https://www.arduino.cc/en/Main/Software
	https://arch.ict.e.uowm.gr/mdasyg/pdfs/dimitriou_thesis.pdf https://slideplayer.gr/slide/11197527/ https://www.robotshop.com/media/files/PDF/dfrobot-arduino-shields-manual.pdf
Arduino Uno	https://store.arduino.cc/arduino-uno-rev3
Ethernet Shield	http://wiki.seeedstudio.com/Ethernet_Shield_V1.0/
Servo	https://eclass.uoa.gr/modules/document/file.php/PRIMEDU478/1.%20Genika_%20Rompotiki%20kai%20Ekpaideytiki%20Rompotiki/klp/4_STEPPER%26SERVOS_LAB.pdf https://robotronics.com/mechanics-and-wheels/servo/servo-el/ https://elearning.teicm.gr/file.php/467/Ergasthrio4.pdf https://en.wikipedia.org/wiki/Servomotor
TowerPro MG996R	https://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf http://www.towerpro.com.tw/product/mg996r/
Code syntax highlight	https://tohtml.com/cpp/
Robot	http://users.softlab.ntua.gr/~ktzaf/Courses/robotics-I-shmmy-kinematics-1.pdf https://el.wikipedia.org/wiki/Ρομπότ http://users.sch.gr/jenyk/index.php/artificialintelligence/ai-historicalreview/11-robotics/17-whatisroboticswhatisrobot https://en.wikipedia.org/wiki/Robot#/media/File:HONDA_ASIMO.jpg
Biped robots	https://en.wikipedia.org/wiki/Humanoid_robot
Διαδικτυακές εφαρμογές	https://el.wikipedia.org/wiki/Διαδικτυακή_εφαρμογή

HTML	https://el.wikipedia.org/wiki/HTML https://www.w3schools.com/tryit/tryit.asp?filename=tryhtml_default https://www.w3schools.com/tags/
CSS	https://el.wikipedia.org/wiki/CSS https://www.w3schools.com/tryit/tryit.asp?filename=tryhtml_default http://web.simmons.edu/~grabiner/comm244/weekthree/css-basic-properties.html
HTTP	https://el.wikipedia.org/wiki/Πρωτόκολλο_Μεταφοράς_Υπερκειμένου https://searchengineland.com/figz/wp-content/uploads/2018/07/client-server.png
Web server	https://en.wikipedia.org/wiki/Web_server
Fritzing	https://en.wikipedia.org/wiki/Fritzing https://fritzing.org/home/
GitLab	https://en.wikipedia.org/wiki/GitLab
Amazon S3	https://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html
Arduino Hello world	http://osoyoo.com/2017/06/23/arduino-lesson-hello-world/
Arduino Servo	https://www.arduino.cc/en/reference/servo
Arduino/Servo/ Web	https://www.arduino.cc/en/reference/ethernet https://randomnerdtutorials.com/arduino-webserver-with-an-arduino-ethernet-shield/ https://www.arduino.cc/en/Tutorial/WebClientRepeating https://www.arduino.cc/en/reference/SPI https://www.arduino.cc/en/reference/ethernet
Arduino/AllServos/Web	https://playground.arduino.cc/Code/WebServerJson/
AngularJS	https://en.wikipedia.org/wiki/AngularJS https://web-mate.gr/ti-einai-i-angular/ https://angularjs.org/
Bootstrap framework	https://getbootstrap.com/2.3.2/ https://el.wikipedia.org/wiki/Bootstrap https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)