



**ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Ανάπτυξη Συστήματος Απομακρυσμένης Διαχείρισης Αμαξιδίου με
Χρήση Arduino**

Χατζηλένας Χρήστος

Επιβλέπων: Γ. Δημητρακόπουλος

**Αθήνα
[Φεβρουάριος, 2015]**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

*Ανάπτυξη Συστήματος Απομακρυσμένης Διαχείρισης Αμαξιδίου με
Χρήση Arduino*

Χατζηλένας Χρήστος

ΑΜ: 21041

Μέλη τριμελούς επιτροπής: Γεώργιος Δημητρακόπουλος, Λέκτορας

Κωνσταντίνος Τσερπές, Λέκτορας

Δημοσθένης Αναγνωστόπουλος, Καθηγητής

Ευχαριστιές

Πρώτα από όλα, θέλω να ευχαριστήσω τον επιβλέποντα της πτυχιακής μου εργασίας, Λέκτορα κ. Γεώργιο Δημητρακόπουλο, για την πολύτιμη βοήθεια και καθοδήγηση κατά τη διάρκεια της δουλειάς μου. Επίσης, είμαι ευγνώμων στα υπόλοιπα μέλη της εξεταστικής επιτροπής της πτυχιακής μου εργασίας, Λέκτορα κ. Κωνσταντίνο Τσερπέ και Καθηγητή κ. Δημοσθένη Αναγνωστόπουλο για την προσεκτική ανάγνωση της εργασίας μου και για τη βαθμολόγησή της. Οφείλω επίσης ευχαριστίες στον Καθηγητή κ. Άγγελο Φλώρο, Αρχιτέκτονα-Σχεδιαστή Διαδραστικών Εφαρμογών, για το ότι ήταν πάντα διαθέσιμος να μου προσφέρει τις γνώσεις και την εμπειρία του, για τα εποικοδομητικά του σχόλια και τη συνεχή στήριξη και υπομονή του. Στη συνέχεια, ευχαριστώ ιδιαίτερα τη Βίκυ Μαρκαντώνη για το χρόνο που διέθεσε για να σχεδιάσει την ιστοσελίδα που συνοδεύει την πτυχιακή μου εργασία, καθώς και τον Αδάμ Λοίζο για όλες τις συζητήσεις μας και τις ιδέες που μου προσέφερε καθ' όλη τη διάρκεια εκπόνησης αυτής της εργασίας.

Σε αυτό το σημείο θέλω να αναφέρω ανθρώπους, εκτός του στενού ακαδημαϊκού περιβάλλοντος, που υπήρξαν σημαντικοί πόλοι στη ζωή μου, προσδίδοντας την απαιτούμενη ισορροπία. Θέλω αρχικά να ευχαριστήσω τον φίλο και συμφοιτητή μου Δημήτρη Παπαβασιλείου για τις χρήσιμες συμβουλές του πάνω στην πτυχιακή μου καθώς και όλους τους φίλους και τις φίλες των φοιτητικών μου χρόνων, που έκαναν τα χρόνια αυτά μία πραγματικά αξέχαστη εμπειρία. Έπειτα, θα ήθελα να ευχαρίστησω τη σχολική μου παρέα, που ήταν, και ελπίζω να είναι δίπλα μου και στο μέλλον. Βέβαια, το μεγαλύτερο ευχαριστώ το οφείλω στην οικογένεια μου και την κοπέλα μου, των οποίων η πίστη στις δυνατότητες μου αποτέλεσε αρωγός σε όλους τους στόχους και τα όνειρά μου.

Πίνακας Περιεχομένων

1	Εισαγωγή	1
1.1	Εισαγωγή	2
1.2	Μια σύντομη ιστορία του Arduino	3
1.3	Τι είναι Physical Computing ;	3
1.4	Τι είναι ο μικροελεγκτής ;	4
1.5	Η πλατφόρμα Arduino	4
1.5.1	Arduino Hardware	5
1.5.2	Ακροδέκτες εισόδου/εξόδου	5
1.6	Το λογισμικό (Arduino IDE)	7
1.7	Γλώσσα Arduino	7
2	Project Plan	8
2.1	Project Plan	9
3	Μέρος Πρώτο – Μελέτη κίνησης αμαξιδίου	10
3.1	Εισαγωγή	11
3.2	Υλικά	11
3.3	Λειτουργία L293DNE H-Bridge	14
3.4	Συνδεσμολογία	15
3.5	Κώδικας	17

4	Μέρος Δεύτερο – Μελέτη επικοινωνίας μέσω Wi-Fi	21
4.1	Εισαγωγή	22
4.2	Κατάσταση παραστατικής μεταφοράς - REpresentational State Transfer (REST)	22
4.3	Λειτουργία Arduino Yun	24
4.4	Υλικά	25
4.5	Συνδεσμολογία	26
4.6	Κώδικας	28
5	Μέρος Τρίτο – Μελέτη Web-Camera	36
5.1	Εισαγωγή	37
5.2	Υλικά	38
5.3	Συνδεσμολογία	40
5.4	Κώδικας	42
6	Μέρος Τέταρτο – Προσθήκη Αισθητήρα για τη διευκόλυνση του χρήστη	52
6.1	Εισαγωγή	53
6.2	Υλικά	54
6.3	Συνδεσμολογία	56
6.4	Κώδικας	58

7	Μέρος Πέμπτο – Προσθήκη ξύλινου σκελετού στο αμαξίδιο	68
7.1	Εισαγωγή	69
7.2	Υλικά	69
7.3	Κατασκευή	70

Εισαγωγή



1.1 Εισαγωγή

Το Arduino είναι μια physical computing ανοιχτού κώδικα πλατφόρμα που βασίζεται σε απλή είσοδο/έξοδο (I/O) και ένα περιβάλλον ανάπτυξης που χρησιμοποιεί την Processing. Επίσης, μπορεί να χρησιμοποιηθεί για την ανάπτυξη αυτόνομων διαδραστικών αντικειμένων ή να συνδυαστεί με κάποιο λογισμικό ενός H/Y (όπως Flash, Processing, VVVV ή Max/MSP). Το Arduino είναι ένα εργαλείο για την πραγματοποίηση διαδραστικών εφαρμογών, σχεδιασμένο για να απλοποιήσει το έργο αυτό για τους αρχάριους, αλλά εξακολουθεί να είναι αρκετά ευέλικτο για τους εμπειρογνώμονες για την ανάπτυξη σύνθετων έργων. Από την ίδρυσή του το 2005, έχουν πωληθεί περισσότερες από 200,000 πλακέτες Arduino και υπάρχει μια ραγδαία αύξηση του αριθμού των projects που το χρησιμοποιούν σαν υπολογιστικό πυρήνα. Η κοινότητα του Arduino είναι τεράστια οπότε, υπάρχει ένας μεγάλος αριθμός σεμιναρίων, βιβλίων και βιβλιοθηκών διαθέσιμος.

Το Arduino είναι διαφορετικό από τις άλλες πλατφόρμες για τους παρακάτω λόγους:

- Το προγραμματιστικό περιβάλλον του μπορεί να τρέξει σε διαφορετικές πλατφόρμες (Windows, Linux, Macintosh).
- Βασίζεται στην IDE Processing, ένα εύκολο στη χρήση περιβάλλον ανάπτυξης που χρησιμοποιείται από καλλιτέχνες και σχεδιαστές.
- Μπορεί να προγραμματιστεί μέσω ενός καλωδίου USB.
- Έχει ανοιχτό hardware και software. Οπότε, όποιος επιθυμεί μπορεί να κατεβάσει το διάγραμμα του κυκλώματος, να αγοράσει τα εξαρτήματα που χρειάζεται και να φτιάξει το δικό του.
- Το hardware είναι αρκετά φθηνό. Ένα USB board κοστίζει περίπου €20 και η αντικατάσταση ενός κατεστραμμένου κυκλώματος δεν κοστίζει παραπάνω από €5.
- Υπάρχει μια ενεργή κοινότητα χρηστών, έτσι υπάρχουν πολλοί άνθρωποι που μπορούν να βοηθήσουν.

1.2 Μια σύντομη ιστορία του Arduino

Το Arduino γεννήθηκε το 2005 στο Interaction Design Institute Ivrea, στην Ιταλία, ως μια ανοιχτού κώδικα πλατφόρμα. Οι ιδρυτές του έργου ήταν οι Massimo Banzi και David Cuartielles. Ο Hernando Barragan, ένας φοιτητής στο ίδιο ινστιτούτο, μαζί με τον Diego Gonzalez Joven είχε αναπτύξει την Wiring ως τη μεταπτυχιακή του διατριβή, η οποία ήταν υπό την επίβλεψη του Massimo Banzi και του Casey Reas (ένας από τους εμπνευστές της Processing). Η ιδέα πίσω από την Wiring ήταν να επιτρέπει την εύκολη χρήση του προγραμματισμού και της ηλεκτρονικής από καλλιτέχνες και σχεδιαστές – σε μια παρόμοια νοοτροπία με την οποία ο Casey Reas και ο Ben Fry είχαν αναπτύξει την Processing μερικά χρόνια νωρίτερα. Το Arduino χτίστηκε γύρω από την Wiring, αλλά αναπτύχθηκε ανεξάρτητα μετά το 2005.

1.3 Τι είναι Physical Computing;

Ο όρος Physical Computing περιγράφει τη ζεύξη του φυσικού χώρου με το χώρο των υπολογιστών. Πρόκειται για ένα ενδιάμεσο πεδίο μεταξύ τέχνης και τεχνολογίας. Μέσα από τη χρήση μικροεπεξεργαστών ανοικτού κώδικα και αισθητήρων, σε συνδυασμό με τη χρήση ειδικού λογισμικού δημιουργούνται πρότυπες και D.I.Y διαδραστικές κατασκευές, εφαρμογές και εγκαταστάσεις οι οποίες προτείνουν πρωτότυπες εμπειρίες φωτός, ήχου και εικόνας.

Περιλαμβάνει το σχεδιασμό διαδραστικών εφαρμογών που μπορούν να επικοινωνούν με ανθρώπους χρησιμοποιώντας αισθητήρες και ενεργοποιητές που ελέγχονται από ένα λογισμικό που τρέχει μέσα σε ένα μικροελεγκτή (ένας μικρός υπολογιστής με ένα μόνο ολοκληρωμένο κύκλωμα).

Στο παρελθόν, η χρήση της ηλεκτρονικής απαιτούσε να ασχοληθεί κανείς με τη μηχανική και απαιτούσε φυσικά και αρκετό χρόνο για την κατασκευή ενός κυκλώματος. Αυτά τα προβλήματα αποθάρρυναν δημιουργικούς ανθρώπους να ασχοληθούν με αυτά. Επίσης, τα περισσότερα εργαλεία που προορίζονται για μηχανικούς απαιτούσαν εκτεταμένη γνώση. Όμως τα τελευταία χρόνια, οι μικροελεγκτές έχουν γίνει φθηνότεροι και πιο εύκολοι στη χρήση επιτρέποντας έτσι τη δημιουργία καλύτερων εργαλείων.

Το Arduino, λοιπόν, έχει φέρει αυτά τα εργαλεία πιο κοντά σε έναν αρχάριο χρήστη, επιτρέποντάς του να αρχίσει να δημιουργεί μικρά projects μετά από 2 ή 3 ημέρες εργαστηρίου. Με το Arduino, ένας σχεδιαστής ή καλλιτέχνης μπορεί να μάθει βασικές γνώσεις ηλεκτρονικής και να γνωρίσει κάθε είδους αισθητήρα πολύ γρήγορα και να υλοποιήσει ένα project χωρίς μεγάλο κόστος.

1.4 Τι είναι ο μικροελεγκτής:

Ένας κεντρικός υπολογιστής ο οποίος μπορεί να χρησιμοποιηθεί για physical computing είναι ένας μικροελεγκτής. Είναι ένας πολύ απλός και μικρός υπολογιστής που είναι πολύ καλός σε τρία συγκεκριμένα πράγματα: λαμβάνει πληροφορίες από αισθητήρες, ελέγχει απλούς κινητήρες και άλλες συσκευές που προκαλούν φυσική αλλαγή και αποστέλει πληροφορίες σε άλλους υπολογιστές ή συσκευές. Ενεργεί ως πύλη μεταξύ του φυσικού κόσμου και του κόσμου των υπολογιστών. Οι μικροελεγκτές βρίσκονται παντού, από τα πλυντήρια μέχρι και τους διακόπτες για τα φώτα. Για αυτό το λόγο η τιμή τους έχει μειωθεί και η χρήση τους είναι ευκολότερη. Έτσι μπορεί κάποιος να εξερευνήσει ενσωματωμένα υπολογιστικά συστήματα στα πιο απίθανα μέρη ή να αναπτύξει απλές συσκευές με πιο σύνθετες λειτουργίες.

1.5 Η πλατφόρμα Arduino

Το Arduino αποτελείται από δύο κύρια μέρη: την πλακέτα Arduino, το οποίο είναι το κομμάτι του hardware για την κατασκευή ενός project, και το Arduino IDE, το κομμάτι του software που τρέχει στον υπολογιστή. Το IDE χρησιμοποιείται για το δημιουργία ενός σκίτσου (ένα μικρό πρόγραμμα), το οποίο στη συνέχεια “ανεβαίνει” στην πλακέτα Arduino. Το σκίτσο λέει στην πλακέτα Arduino τι πρέπει να κάνει.

Πριν λίγα χρόνια, το να δουλεύει κανείς πάνω στο hardware σήμαινε ότι έπρεπε να φτιάξει ένα κύκλωμα από το μηδέν. Όπως επίσης, να χρησιμοποιούσε εκατοντάδες διαφορετικά εξαρτήματα όπως αντιστάσεις, πυκνωτές, πηνία, τρανζίστορ, κ.ο.κ.. Κάθε κύκλωμα ήταν δύσκολο να αλλάξει, καθώς θα έπρεπε ο χρήστης να κόψει καλώδια, να κάνει συγκολλήσεις, και άλλα.

Με την εμφάνιση των ψηφιακών τεχνολογιών και των μικροελεγκτών αυτές οι λειτουργίες αντικαταστάθηκαν με προγράμματα λογισμικού. Το λογισμικό επιδέχεται ευκολότερα αλλαγές από το hardware. Με λίγα πατήματα πλήκτρων μπορεί να αλλάξει ολοκληρωτικά η λογική της συσκευής.

1.5.1 Arduino Hardware

Η πλακέτα Arduino είναι μια μικρή πλακέτα μικροελεγκτή, η οποία είναι ένα μικρό κύκλωμα που περιέχει ένα ολόκληρο υπολογιστή σε ένα μικρό ολοκληρωμένο κύκλωμα (το μικροελεγκτή). Μπορούμε να σκεφτούμε την πλακέτα Arduino ως ένα μικρό εγκέφαλο που μας επιτρέπει να συνδέσουμε μαζί ένα ευρύ φάσμα αισθητήρων και ενεργοποιητών. Η πλακέτα Arduino είναι χτισμένη γύρω από ένα 8-bit μικροελεγκτή ATmel AVR. Ανάλογα με την πλακέτα, βρίσκουμε διάφορες μάρκες, συμπεριλαμβανομένων των ATmega8, ATmega168, ATmega328, ATmega1280, και ATmega2560. Οι είσοδοι/έξοδοι μιας πλακέτας Arduino μπορούν να χρησιμοποιηθούν και σαν είσοδοι/έξοδοι για άλλα κυκλώματα χτισμένα γύρω από ένα Arduino. Το Arduino Uno είναι η πιο απλή πλακέτα που μπορούμε να χρησιμοποιήσουμε.

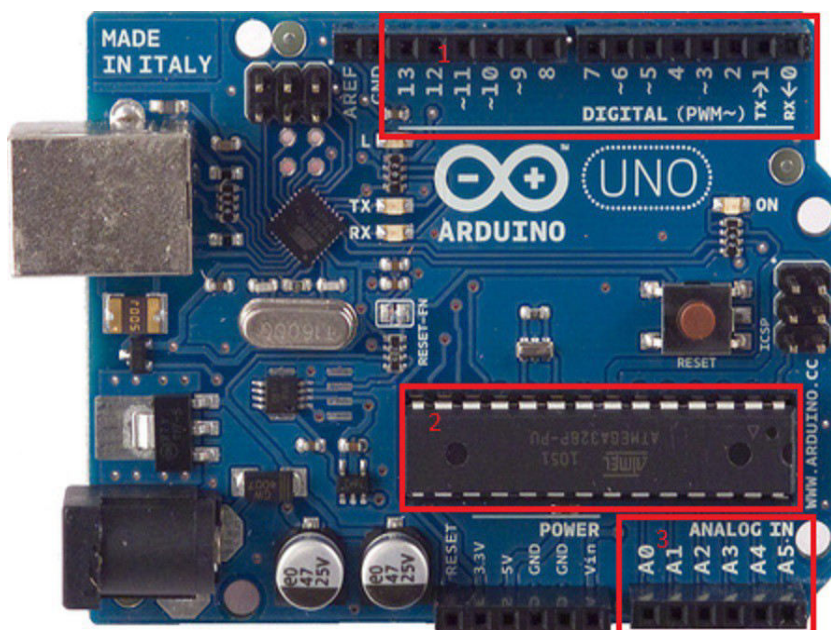
1.5.2 Ακροδέκτες εισόδου/εξόδου

Το Arduino Uno έχει 14 ακροδέκτες ψηφιακής εισόδου/εξόδου και 6 ακροδέκτες αναλογικής εισόδου, όπως φαίνεται στο παρακάτω σχήμα (Εικόνα 1). Αυτοί οι ακροδέκτες αντιστοιχούν στους ακροδέκτες εισόδου/εξόδου του ATmega μικροελεγκτή. Η αρχιτεκτονική της πλακέτας διαθέτει αυτούς τους ακροδέκτες για να είναι ευκολότερο να συνδεθεί η πλακέτα με εξωτερικά κυκλώματα. Οι ακροδέκτες μπορούν να οριστούν με 2 τρόπους ως εισόδου και εξόδου. Οι ακροδέκτες του ATmega είναι εξ' ορισμού ακροδέκτες εισόδου του Arduino, αν και θα πρέπει να αναφέρεται συχνά στον κώδικα για λόγους σαφήνειας.

1. Ψηφιακοί ακροδέκτες (digital pins): Το Arduino έχει 14 ψηφιακούς ακροδέκτες, οι οποίοι είναι αριθμημένοι από το 0 έως το 13 και μπορούν να οριστούν και ως ακροδέκτες εισόδου αλλά και ως εξόδου. Και στις 2 αυτές λειτουργίες οι ακροδέκτες μπορούν μόνο να στέλνουν ή να λαμβάνουν ψηφιακά σήματα τα οποία αποτελούνται από 2 καταστάσεις: ON (HIGH ή 5V) και OFF (LOW ή 0V). Οι ακροδέκτες που έχουν οριστεί ως έξοδοι μπορούν να παρέχουν ρεύμα σε εξωτερικές συσκευές ή

κυκλώματα. Αντίθετα, οι ακροδέκτες που έχουν οριστεί ως είσοδοι διαβάζουν ρεύματα από συσκευές που είναι συνδεδεμένες σε αυτούς. Τέλος, 6 από αυτούς του ακροδέκτες (PWM pins) μπορούν να χρησιμοποιηθούν και ως ακροδέκτες διαμόρφωσης εύρους παλμού (PWM). Με την χρήση αυτών των ακροδεκτών μπορούμε να εξομοιώσουμε αναλογικές τιμές.

2. Αναλογικοί ακροδέκτες εισόδου (analog input pins): Οι μικροελεγκτές ATmega, που χρησιμοποιούν οι πλακέτες Arduino, περιέχουν 6 κανάλια analog-to-digital μετατροπέα (ADC). Η λειτουργία αυτής της συσκευής είναι να μετατρέψει μία αναλογική τάση εισόδου σε ένα ψηφιακό αριθμό ανάλογο με το μέγεθος της τάσης εισόδου σε σχέση με την τάση αναφοράς (5V). Ο μετατροπέας ATmega για τις πλακέτες Arduino έχει ανάλυση 10-bit, το οποίο σημαίνει ότι θα επιστρέψει ακεραίους από 0 έως 1023. Για παράδειγμα, μια είσοδος 0V θα επιστρέψει 0, μια είσοδος 5V θα επιστρέψει 1023, ενώ μια είσοδος 2.5V θα επιστρέψει 512. Αυτοί οι ακροδέκτες μπορούν να οριστούν και ως ακροδέκτες εισόδου ή εξόδου ακριβώς όπως και οι ψηφιακοί ακροδέκτες. Αν σε κάποιο project χρειαστούν παραπάνω ψηφιακοί ακροδέκτες, μπορούμε να χρησιμοποιήσουμε και τους αναλογικούς ακροδέκτες σαν να είναι ψηφιακοί.



Εικόνα 1: 1.Ψηφιακοί ακροδέκτες, 2.ATmega328 μικροελεγκτής, 3.Αναλογικοί ακροδέκτες

1.6 Το Λογισμικό (IDE)

Το Arduino IDE είναι μια εφαρμογή cross-platform γραμμένη σε Java και προέρχεται από τον IDE για την Processing και την Wiring. Ένα IDE είναι ένα πρόγραμμα που έχει σχεδιαστεί για να διευκολύνει την ανάπτυξη λογισμικού. Το Arduino IDE περιλαμβάνει ένα πρόγραμμα επεξεργασίας πηγαίου κώδικα και έναν compiler που κάνει τον κώδικα κατανοητό στο μικροελεγκτή. Όταν, λοιπόν, “ανέβει” το σκίτσο (πρόγραμμα) στην πλακέτα: ο κώδικας μεταφράζεται στη γλώσσα C και περνά στο avr-gcc compiler, ένα σημαντικό κομμάτι του λογισμικού ανοιχτού κώδικα που κάνει την τελική μετάφραση και έτσι ο κώδικας γίνεται κατανοητός από το μικροελεγκτή.

Ο κύκλος προγραμματισμού για Arduino είναι ο εξής:

- Σύνδεση της πλακέτας σε μια θύρα USB ενός υπολογιστή.
- Γραφή σκίτσου.
- “Ανέβασμα” του σκίτσου στην πλακέτα μέσω της USB σύνδεσης και αναμονή για επανεκκίνηση της πλακέτας.
- Η πλακέτα εκτελεί το σκίτσο που μόλις “ανέβηκε”.

1.7 Γλώσσα Arduino

Η γλώσσα Arduino υλοποιείται σε C/C++ και βασίζεται στην Wiring. Όταν γράφεται ένα σκίτσο γίνεται χρήση της βιβλιοθήκης Wiring, η οποία περιλαμβάνεται στο Arduino IDE. Αυτό μας επιτρέπει να φτιάχνουμε εκτελέσιμα προγράμματα με δύο μόνο λειτουργίες: setup() και loop(). Όπως αναφέρθηκε, η Wiring είναι εμπνευσμένη από την Processing και η δομή της γλώσσας Arduino κληρονομείται από την Processing.

Project Plan



2.1 Project Plan

Η κύρια ιδέα της παρούσας πτυχιακής εργασίας ήταν να φτιάξω ένα αμαξίδιο το οποίο θα συνδεέται μέσω Wi-Fi με έναν υπολογιστή και από εκεί ο χρήστης θα μπορεί να το ελέγξει. Η επικοινωνία, δηλαδή, θα γίνεται μεταξύ ενός Server (αμαξίδιο) και ενός Client (ιστοσελίδα). Επίσης, σκέφτηκα να προσθέσω στο αμαξίδιο μία web-camera έτσι ώστε ο χρήστης να μπορεί να το κινεί ακόμα και αν βρίσκεται σε διαφορετικό χώρο από ότι το αμαξίδιο. Σε αυτό το σημείο πρέπει να αναφέρω ότι η επικοινωνία επιτυγχάνεται μόνο όταν ο Server (αμαξίδιο) και ο Client (ιστοσελίδα) βρίσκονται στο ίδιο δίκτυο. Τέλος, προχώρησα στην προσθήκη ενός αισθητήρα, μίας φωτοαντίστασης (ή φωτοκύτταρο), για να διευκολύνω επιπλέον το χρήστη. Τελικά, το project είχε τα παρακάτω στάδια:

1. Μελέτη κίνησης αμαξιδίου.
2. Μελέτη επικοινωνίας μέσω Wi-Fi.
3. Μελέτη ζωντανής μετάδοσης με web-camera.
4. Μελέτη αισθητήρα για τη διευκόλυνση του χρήστη.
5. Προσθήκη ξύλινου (κόντρα πλακέ) σκελετού στο αμαξίδιο.

Μέρος Πρώτο – Μελέτη κίνησης αμαξιδίου

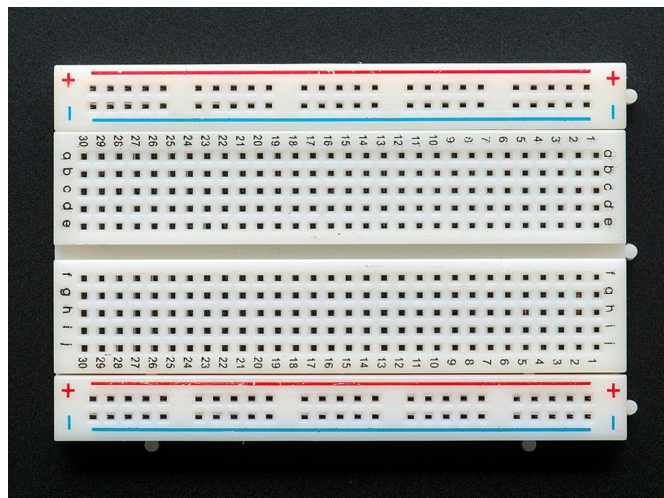


3.1 Εισαγωγή

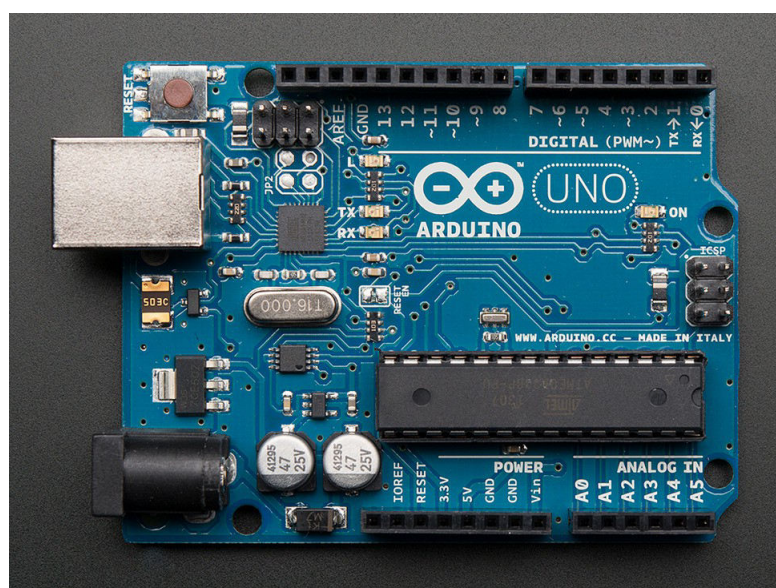
Στο αυτό μέρος της εργασίας μου μελέτησα την κίνηση του αμαξιδίου. Στο τέλος αυτού του μέρους σκοπός μου ήταν να φτιάξω ένα αμαξίδιο το οποίο θα κάνει προκαθορισμένες κινήσεις. Για παράδειγμα, αφού πρώτα θα κινούταν ευθεία για 5 δευτερόλεπτα, στη συνέχεια θα έστριβε για 5 δευτ. δεξιά, μετά αριστερά για τον ίδιο χρόνο και τέλος θα πήγαινε προς τα πίσω για 5 δευτ. και στο τέλος θα σταματούσε.

3.2 Υλικά

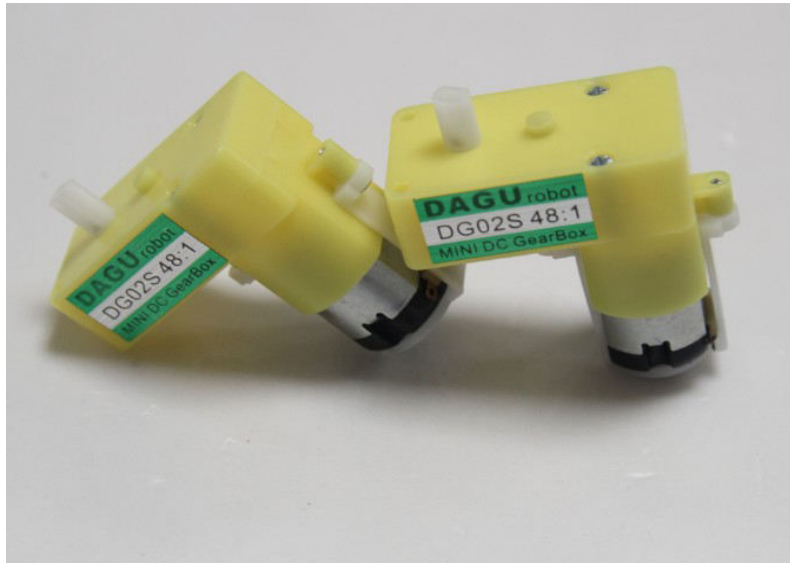
1. Μια breadboard.



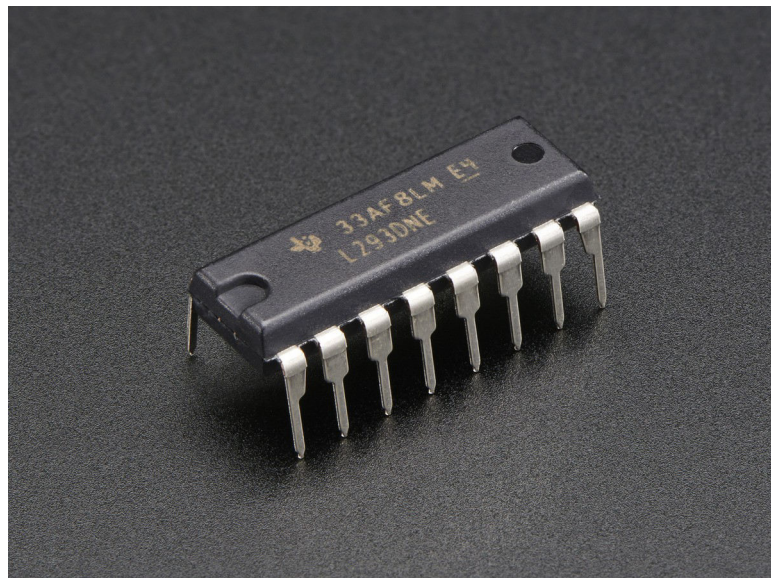
2. Ένα Arduino Uno. Δεν χρησιμοποίησα τόσο νωρίς το Arduino Yun, καθώς δεν χρειαζόταν ακόμα σύνδεση με Wi-Fi.



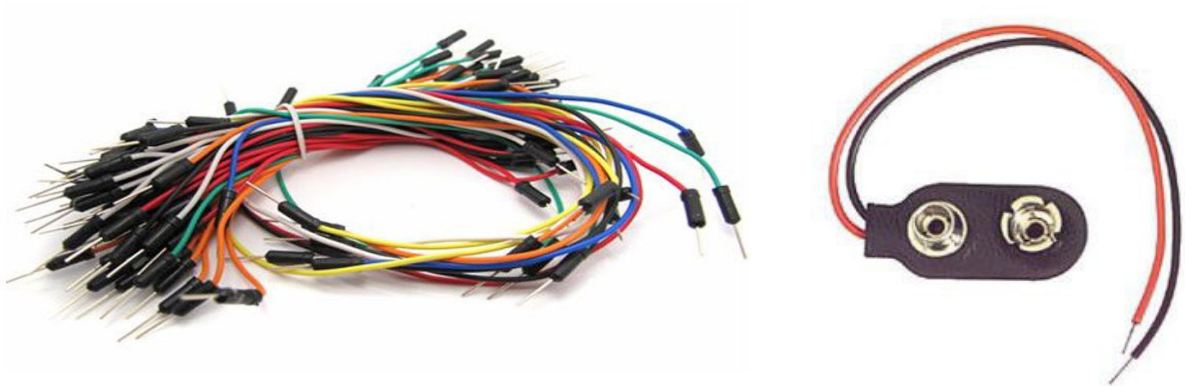
3. Δύο DC motors για την κίνηση των τροχών του αμαξιδίου.



4. Ένα L293DNE H-bridge κύκλωμα με το οποίο γίνεται ο έλεγχος της ταχύτητας και της κατεύθυνσης των DC motors για να κινηθούν οι τροχοί.



5. Breadboard jumper καλώδια και ένα clip για μπαταρία των 9V και μια μπαταρία 9V.



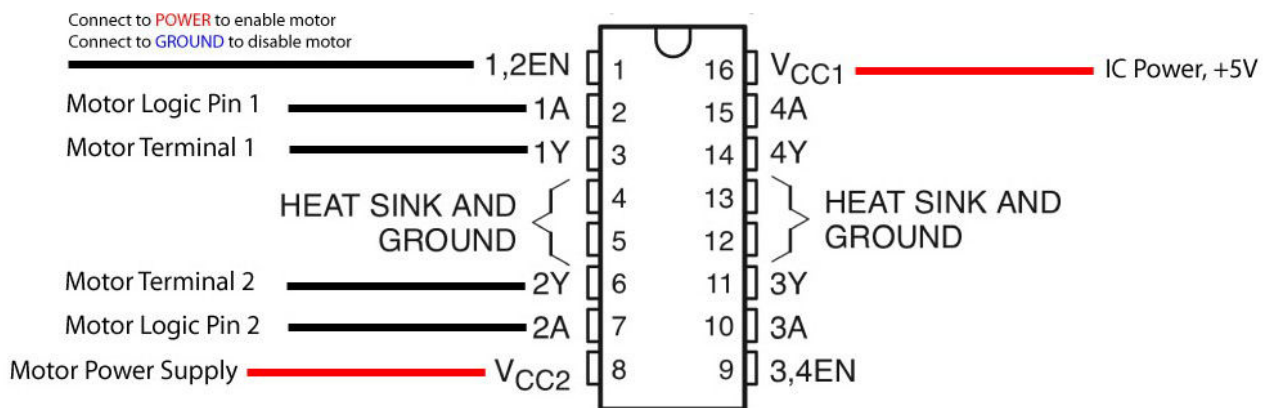
3.3 Λειτουργία L293DNE H-bridge

Σε αυτό το στάδιο το σημαντικότερο θέμα ήταν να καταλάβω πως λειτουργεί ένα κύκλωμα h-bridge έτσι ώστε να καταφέρω να ελέγξω την κίνηση των DC motors(DC κινητήρες). Η L293DNE H-bridge που χρησιμοποίησα είναι μια απλή h-γέφυρα. Έχει δύο γέφυρες, η μία είναι στην αριστερή πλευρά του chip, ενώ η άλλη στη δεξιά, και μπορεί να ελέγξει δύο DC motors.

Η h-bridge έχει τους παρακάτω ακροδέκτες:

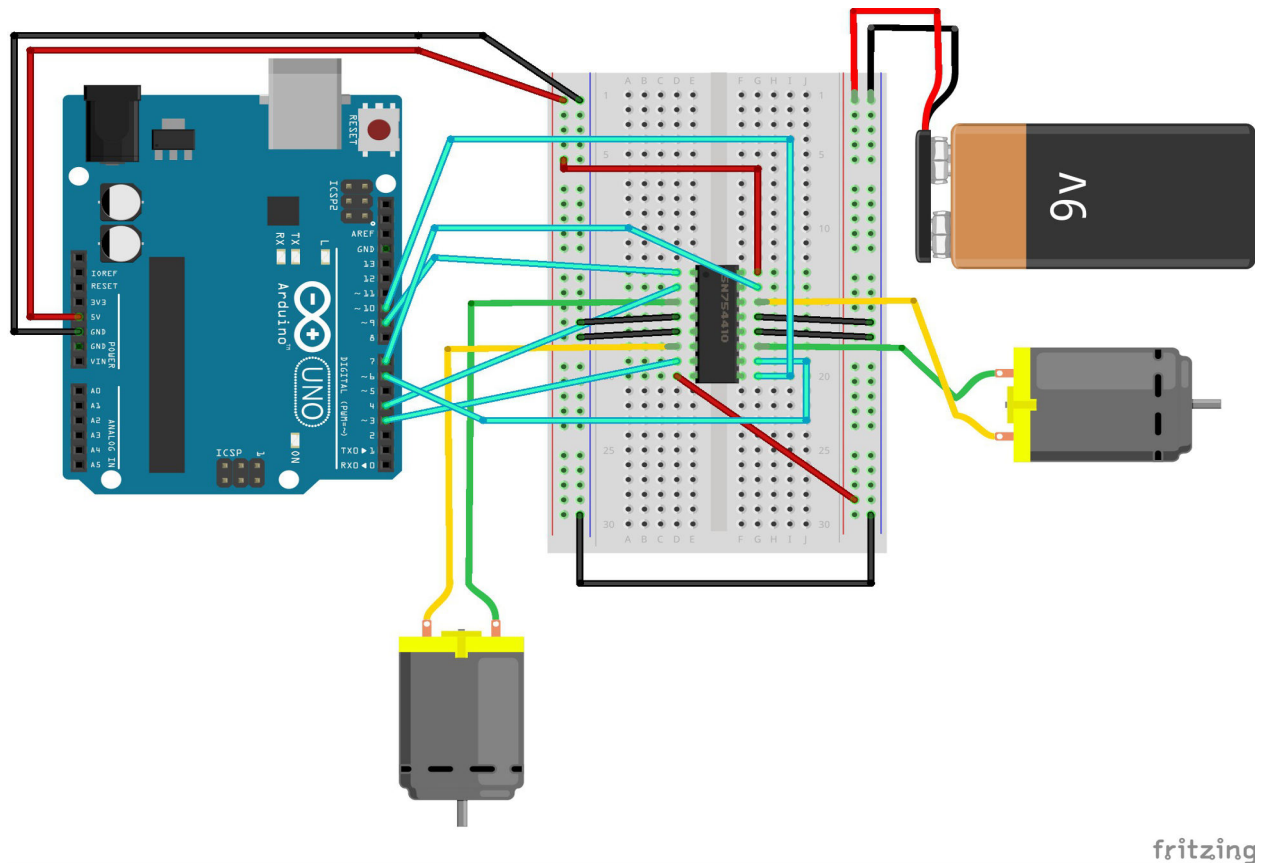
- Pin 1 (1,2EN) ενεργοποιεί ή απενεργοποιεί τον κινητήρα δίνοντας HIGH ή LOW
- Pin 2 (1A) είναι λογικός ακροδέκτης για τον κινητήρα (η είσοδος είναι HIGH ή LOW)
- Pin 3 (1Y) είναι για έναν από τους ακροδέκτες του κινητήρα
- Pin 4-5 είναι για τη γείωση
- Pin 6 (2Y) είναι για τον άλλον ακροδέκτη του κινητήρα
- Pin 7 (2A) είναι λογικός ακροδέκτης για τον κινητήρα (η είσοδος είναι HIGH ή LOW)
- Pin 8 (VCC2) είναι η τροφοδοσία του κινητήρα
- Pin 9 (3,4EN) ενεργοποιεί ή απενεργοποιεί τον κινητήρα δίνοντας HIGH ή LOW
- Pin 10 (3A) είναι λογικός ακροδέκτης για τον κινητήρα (η είσοδος είναι HIGH ή LOW)
- Pin 11 (3Y) είναι για έναν από τους ακροδέκτες του κινητήρα
- Pin 12-13 είναι για τη γείωση
- Pin 14 (4Y) είναι για τον άλλον ακροδέκτη του κινητήρα
- Pin 15 (4A) είναι λογικός ακροδέκτης για τον κινητήρα (η είσοδος είναι HIGH ή LOW)
- Pin 16 (VCC1) τροφοδοσία 5V από το Arduino

Παρακάτω βλέπετε ένα διάγραμμα μιας h-bridge.



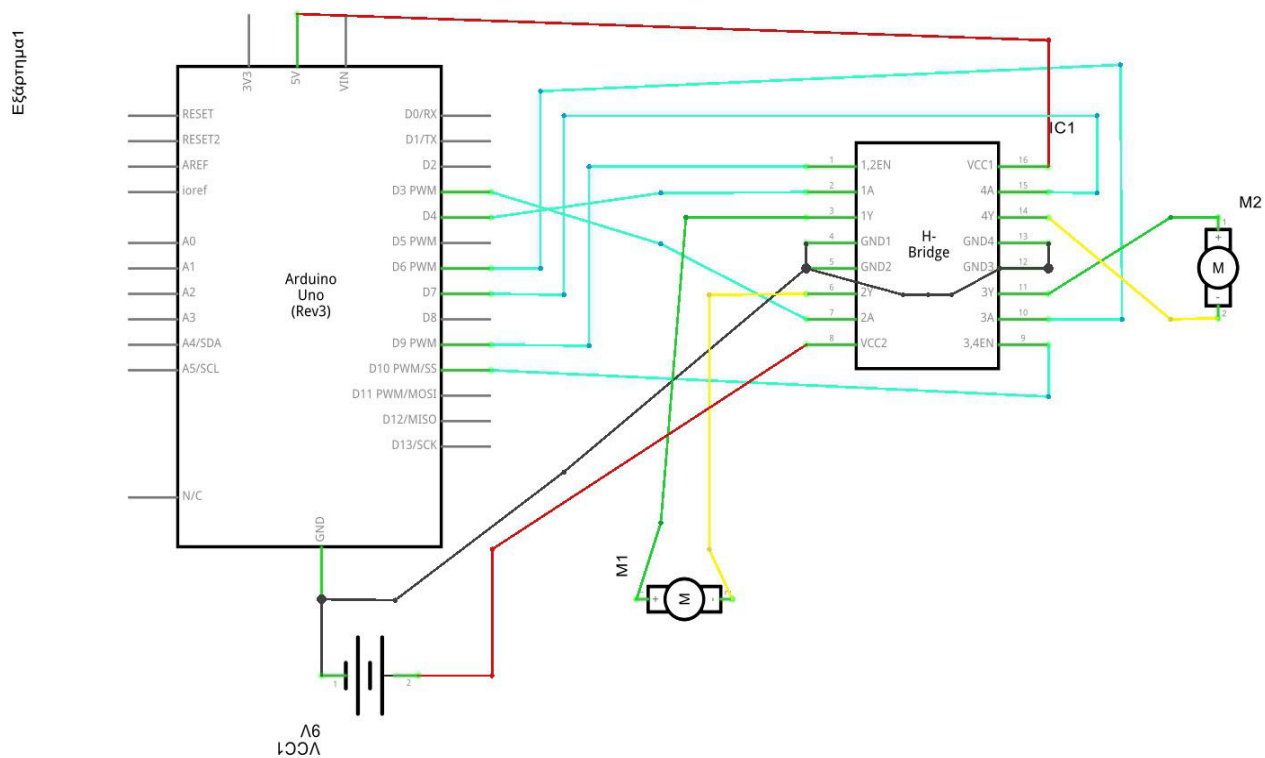
3.4 Συνδεσμολογία

Η συνδεσμολογία στη breadboard είναι η εξής:



Αρχικά, τοποθετούμε την L293DNE H-bridge στη breadboard έτσι ώστε το pin 1 να βρίσκεται στη θέση e13 της breadboard. Στη συνέχεια, τοποθετούμε ένα καλώδιο από το pin 1 στο digital pin ~9 του Arduino, από το pin 2 στο digital pin 4, από το pin 7 στο digital pin ~3 και από το pin 9 στο ρεύμα που μας δίνει η μπαταρία. Ακόμα, συνδέουμε το pin 9 με το digital pin ~10, το pin 10 με το digital pin ~6, το pin 15 με το digital pin 7 και το pin 16 με το ρεύμα του Arduino. Επιπλέον, τα pin 4, 5, 12 και 13 της h-bridge τα συνδέουμε στη γείωση που μας δίνει το Arduino, ενώ τα pin 3, 6 και 11, 14 τα συνδέουμε με τους DC κινητήρες. Τέλος, συνδέουμε το ρεύμα και τη γείωση για το Arduino και τη μπαταρία.

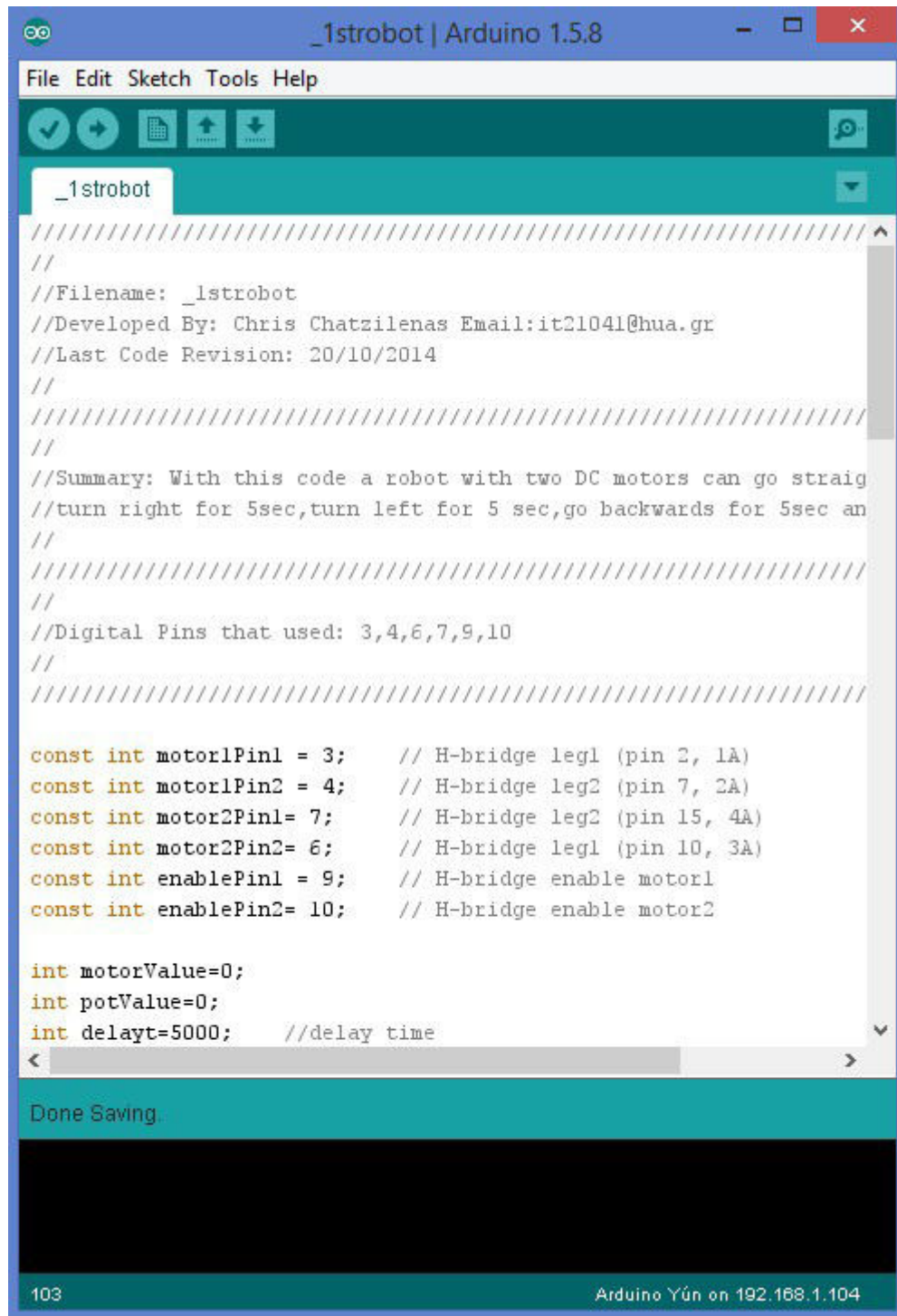
Το σχέδιο είναι το εξής:



fritzing

3.5 Κώδικας

Ο κώδικας είναι ο εξής:



```
File Edit Sketch Tools Help

_1strobot

////////////////////////////////////
//
//Filename: _1strobot
//Developed By: Chris Chatzilenas Email:it21041@hua.gr
//Last Code Revision: 20/10/2014
//
////////////////////////////////////
//
//Summary: With this code a robot with two DC motors can go straig
//turn right for 5sec,turn left for 5 sec,go backwards for 5sec an
//
////////////////////////////////////
//
//Digital Pins that used: 3,4,6,7,9,10
//
////////////////////////////////////

const int motor1Pin1 = 3;    // H-bridge leg1 (pin 2, 1A)
const int motor1Pin2 = 4;    // H-bridge leg2 (pin 7, 2A)
const int motor2Pin1= 7;     // H-bridge leg2 (pin 15, 4A)
const int motor2Pin2= 6;     // H-bridge leg1 (pin 10, 3A)
const int enablePin1 = 9;    // H-bridge enable motor1
const int enablePin2= 10;    // H-bridge enable motor2

int motorValue=0;
int potValue=0;
int delayt=5000;    //delay time

Done Saving.

103 Arduino Yun on 192.168.1.104
```



```
void setup() {  
  
    // set all the other pins you're using as outputs:  
    pinMode(motor1Pin1, OUTPUT);  
    pinMode(motor1Pin2, OUTPUT);  
    pinMode(motor2Pin1, OUTPUT);  
    pinMode(motor2Pin2, OUTPUT);  
    pinMode(enablePin1, OUTPUT);  
    pinMode(enablePin2, OUTPUT);  
  
    // set enablePin high so that motor can turn on:  
    digitalWrite(enablePin1, HIGH);  
    digitalWrite(enablePin2, HIGH);  
}  
  
void loop() {  
  
    //drive straight:  
    //set leg 2 of the 2 motors HIGH to go straight  
    digitalWrite(motor1Pin1, LOW);  
    digitalWrite(motor1Pin2, HIGH);  
  
    digitalWrite(motor2Pin1, HIGH);  
    digitalWrite(motor2Pin2, LOW);  
    delay(delayt);  
  
    //turn right:  
    //set leg 2 of the 2 motors HIGH  
    //set leg 1 of motor1 LOW  
    //set leg 1 of motor2 Higher than LOW  
    digitalWrite(motor1Pin1, LOW);  
    digitalWrite(motor1Pin2, HIGH);  
  
    potValue=350;  
    motorValue=map(potValue,0,1023,0,255);  
  
    digitalWrite(motor2Pin1, HIGH);  
    analogWrite(motor2Pin2, motorValue);  
    delay(delayt);  
}
```

1 Arduino Yun on 192.168.1.104

```
//turn left:
//set leg 2 of the 2 motors HIGH
//set leg 1 of motor2 LOW
//set leg 1 of motor1 Higher than LOW

potValue=350;
motorValue=map(potValue,0,1023,0,255);
analogWrite(motor1Pin1,motorValue);
digitalWrite(motor1Pin2, HIGH);

digitalWrite(motor2Pin1,HIGH);
digitalWrite(motor2Pin2, LOW);
delay(delayt);

//reverse
//set leg 1 of the 2 motors HIGH to go backwards
digitalWrite(motor1Pin2,LOW);
digitalWrite(motor1Pin1, HIGH);

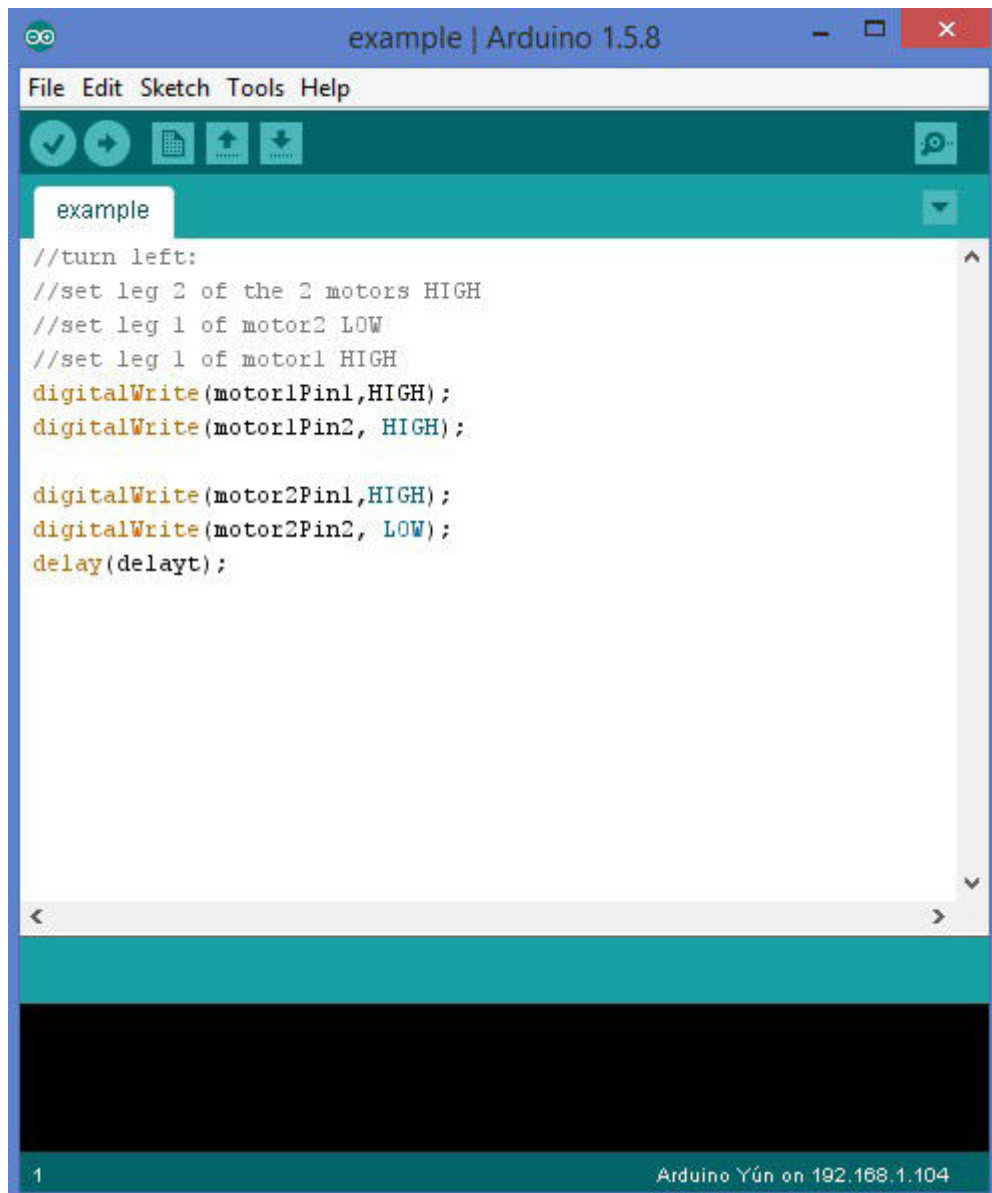
digitalWrite(motor2Pin2,HIGH);
digitalWrite(motor2Pin1, LOW);
delay(delayt);

//stop
//set leg 1,2 of the 2 motors HIGH or LOW to stop
digitalWrite(motor1Pin1,HIGH);
digitalWrite(motor1Pin2, HIGH);

digitalWrite(motor2Pin1,HIGH);
digitalWrite(motor2Pin2, HIGH);
delay(delayt);
}
```

1 Arduino Yún on 192.168.1.104

Για να στρίβει το αμαξίδιο θα μπορούσα πιο απλά να είχα γράψει:

A screenshot of the Arduino IDE interface. The title bar reads "example | Arduino 1.5.8". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for checking, running, saving, and uploading. A tab labeled "example" is active. The main text area contains the following code:

```
//turn left:
//set leg 2 of the 2 motors HIGH
//set leg 1 of motor2 LOW
//set leg 1 of motor1 HIGH
digitalWrite(motor1Pin1,HIGH);
digitalWrite(motor1Pin2, HIGH);

digitalWrite(motor2Pin1,HIGH);
digitalWrite(motor2Pin2, LOW);
delay(delayt);
```

The bottom status bar shows "1" on the left and "Arduino Yun on 192.168.1.104" on the right.

Με αυτόν τον τρόπο, όμως, το αμαξίδιο θα έστριβε πολύ απότομα, καθώς ο αριστερός κινητήρας θα έμενε ακίνητος. Για αυτό το λόγο, κινώντας με μικρότερη ταχύτητα τον αριστερό κινητήρα το αμαξίδιο στρίβει πιο ομαλά. Η τιμή της μεταβλητής `potValue` βρέθηκε μετά από πειραματισμό με διάφορες τιμές. Οπότε, με τη συνάρτηση `map(value, fromLow, fromHigh, toLow, toHigh)` αναγάγω την τιμή της `potValue` από μια περιοχή τιμών σε μία άλλη. Για παράδειγμα, ενώ η τιμή της `potValue` μπορεί να είναι 0-1023, με τη συνάρτηση `map` μπορεί να αναχθεί στις τιμές 0-255.

Μέρος Δεύτερο – Μελέτη επικοινωνίας μέσω Wi-Fi



4.1 Εισαγωγή

Στο δεύτερο μέρος της πτυχιακής μου μελέτησα την επικοινωνία μέσω Wi-Fi ενός Arduino Yun με μία ιστοσελίδα σε localhost. Σκοπός του βήματος αυτού, είναι να δίνονται εντολές στο αμαξίδιο μέσω μια ιστοσελίδας, η οποία αποτελεί τον Client, και εκείνο (Server) να κινείται αναλόγως. Η επικοινωνία βασίζεται στο REST (REpresentational State Transfer, Κατάσταση παραστατικής μεταφοράς) API που υποστηρίζει το Arduino Yun.

4.2 Κατάσταση παραστατικής μεταφοράς - REpresentational State Transfer (REST)¹

Το Representational State Transfer (REST) αναφέρεται σε ένα αρχιτεκτονικό στυλ για την υλοποίηση Web υπηρεσιών. Τέτοιες Web υπηρεσίες ονομάζονται Web υπηρεσίες RESTful. Αν και το ίδιο το REST δεν είναι πρότυπο, οι Web υπηρεσίες RESTful υλοποιούνται χρησιμοποιώντας Web πρότυπα.

Η βασική έννοια στην αρχιτεκτονική REST είναι εκείνη του πόρου. Ένας πόρος είναι οποιοδήποτε κομμάτι πληροφορίας το οποίο μπορεί να προσδιοριστεί με μοναδικό τρόπο. Στην αρχιτεκτονική REST οι αιτούντες και οι υπηρεσίες ανταλλάσσουν μηνύματα τα οποία συνήθως περιέχουν τόσο δεδομένα όσο και μεταδεδομένα. Το τμήμα των δεδομένων ενός μηνύματος αντιστοιχεί σε μια συγκεκριμένη αναπαράσταση του πόρου, όπως περιγράφεται από τα συνοδευτικά μεταδεδομένα (τη μορφή), τα οποία μπορεί επίσης να περιέχουν πρόσθετες οδηγίες επεξεργασίας. Η ανταλλαγή ενός πόρου μπορεί να γίνει σε πολλές αναπαραστάσεις. Οι δύο επικοινωνούντες συνεργάτες έχουν τη δυνατότητα να συμφωνήσουν εκ των προτέρων σε μια συγκεκριμένη αναπαράσταση.

Οι υπηρεσίες στην αρχιτεκτονική REST δεν διατηρούν την κατάσταση μιας αλληλεπίδρασης με έναν αιτούντα · δηλαδή, αν μια αλληλεπίδραση απαιτεί μια κατάσταση, όλες οι καταστάσεις πρέπει να αποτελούν μέρος των μηνυμάτων που ανταλλάσσονται. Με το να είναι οι υπηρεσίες μη καταστασιακές, αυξάνουν την αξιοπιστία τους, απλοποιώντας την επεξεργασία της

1. Deitel Paul and Deitel Harvey (2009)

Weerawarana Sanjiva, Curbera Fransisco, Leymann Frank, Storey Tony, Ferguson F. Donald (2005)

ανάκαμψης (Μια υπηρεσία που παρουσίασε βλάβη μπορεί να επανακάμψει γρήγορα). Επιπλέον, τέτοιες υπηρεσίες έχουν βελτιωμένη επεκτασιμότητα, επειδή δεν χρειάζεται να διατηρούν την κατάστασή τους, και έτσι δεν καταναλώνουν μνήμη, αντιπροσωπεύοντας ενεργές αλληλεπιδράσεις. Η αξιοπιστία και η επεκτασιμότητα αποτελούν απαραίτητες ιδιότητες του Ιστού. Η συμμόρφωση με το αρχιτεκτονικό ύφος REST επιτρέπει την κάλυψη αυτών των απαιτήσεων.

Η αρχιτεκτονική REST χρησιμοποιεί μια απλή διασύνδεση για το χειρισμό των πόρων κατά ένα γενικό τρόπο. Αυτή η διασύνδεση βασικά υποστηρίζει τη μέθοδο της δημιουργίας, ανάκτησης, ενημέρωσης, και διαγραφής (create, retrieve, update, and delete, CRUD). Τα μεταδεδομένα των αντίστοιχων μηνυμάτων περιλαμβάνουν το όνομα της μεθόδου, και το αναγνωριστικό του πόρου στον οποίο στοχεύει η μέθοδος. Εκτός από την περίπτωση της μεθόδου ανάκτησης, το μήνυμα περιλαμβάνει μια αναπαράσταση του πόρου. Επομένως, τα μηνύματα είναι αυτοπεριγραφικά.

Τα αναγνωριστικά που περιλαμβάνονται στα μηνύματα είναι θεμελιώδη για τη REST. Υποδηλώνουν περαιτέρω οφέλη αυτού του αρχιτεκτονικού ύφους. Για παράδειγμα, με το ρητό ορισμό του αναγνωριστικού του πόρου, η REST δίνει τη δυνατότητα για στρατηγικές αποθήκευσης στην κρυφή μνήμη (caching) σε διάφορα επίπεδα, και σε κατάλληλους ενδιάμεσους, κατά μήκος της διαδρομής των μηνυμάτων. Ένας ενδιάμεσος θα μπορούσε να διαπιστώσει ότι διαθέτει ένα έγκυρο αντίγραφο του πόρου, και είναι σε θέση να ικανοποιήσει μία αίτηση ανάκτησης χωρίς να μεταβιβάσει περαιτέρω την αίτηση. Αυτό συμβάλλει στην επεκτασιμότητα του γενικού περιβάλλοντος.

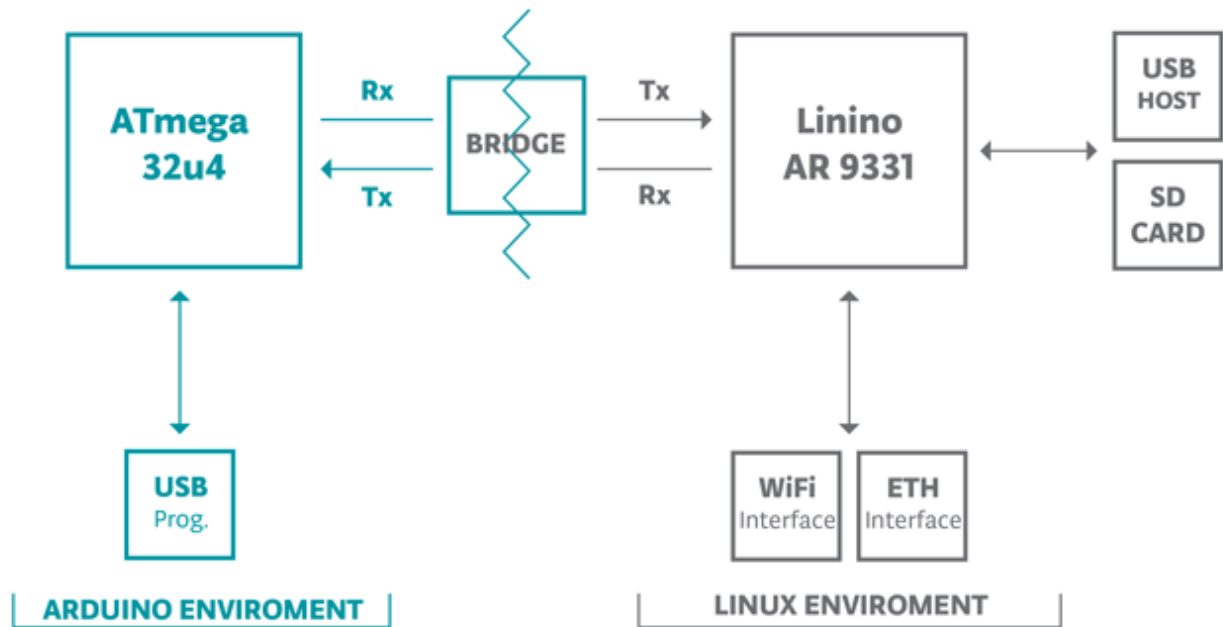
Κάθε μέθοδος σε μια RESTful Web υπηρεσία προσδιορίζεται από ένα μοναδικό URL. Κατά συνέπεια, όταν ο διακομιστής λαμβάνει ένα αίτημα, ξέρει αμέσως ποια λειτουργία να εκτελέσει. Τέτοιες Web υπηρεσίες μπορούν να χρησιμοποιηθούν σε ένα πρόγραμμα ή άμεσα από έναν Web browser.

4.3 Λειτουργία Arduino Yun

Το Arduino Yun είναι το πρώτο μέλος μια πρωτοποριακής γραμμής προϊόντων Wi-Fi που συνδυάζουν τη δύναμη του Linux με την ευκολία χρήσης του Arduino. Το πρώτο Arduino Yun βασίζεται σε έναν ATmega32u4 επεξεργαστή και τον Atheros AR9331. Ο επεξεργαστής Atheros υποστηρίζει μια διανομή Linux βασισμένη στο OpenWrt, ονομάζεται OpenWrt-Yun. Η πλακέτα έχει ενσωματωμένα Ethernet και WiFi υποστήριξη, μια θύρα USB-A, υποδοχή κάρτας micro-SD, 20 ψηφιακές εισόδου / εξόδου ακροδέκτες (εκ των οποίων 7 είναι δυνατόν να χρησιμοποιηθούν ως εξοδοί PWM και 12 ως αναλογικές εισοδοί), ένα 16 MHz κρύσταλλο ταλαντωτή, μία micro USB σύνδεση, μια κεφαλίδα ICSP, και 3 κουμπιά επαναφοράς.

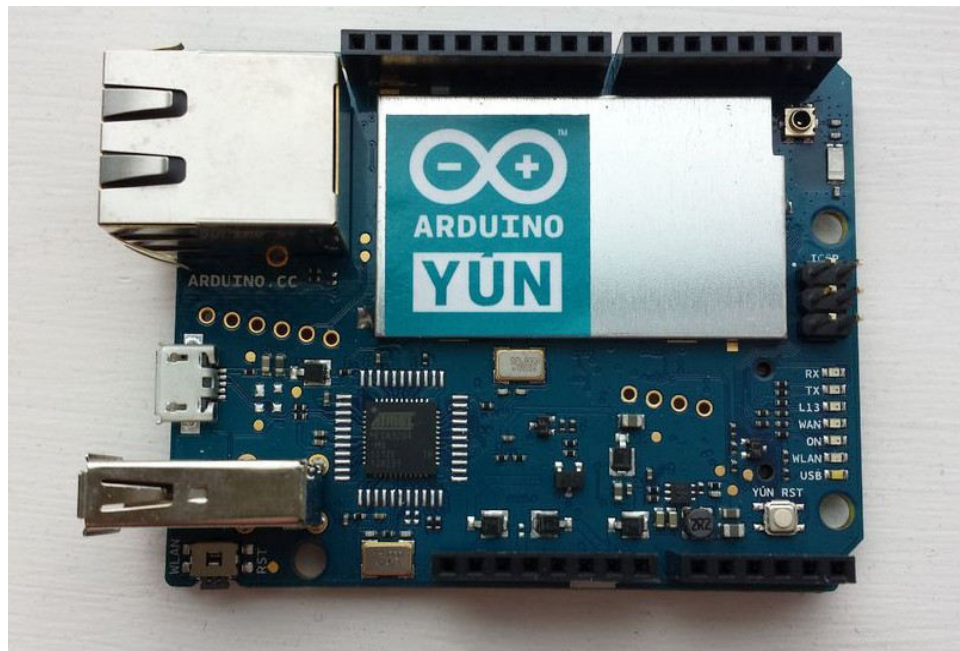
Το Yun ξεχωρίζει από τις άλλες πλακέτες Arduino στο ότι μπορεί να επικοινωνήσει με την διανομή Linux, προσφέροντας ένα ισχυρό δικτυωμένο υπολογιστή με την ευκολία του Arduino. Εκτός από τις εντολές Linux, όπως η cURL, μπορεί να γραφτεί και κώδικας σε python. Το Yun είναι παρόμοιο με το Arduino Leonardo στον επεξεργαστή ATmega32u4 που έχει ενσωματωμένη σύνδεση με USB, εξαλείφοντας έτσι την ανάγκη για ένα δεύτερο επεξεργαστή. Αυτό επιτρέπει στο Yun να εμφανιστεί σε έναν συνδεδεμένο υπολογιστή ως ποντίκι, πληκτρολόγιο ή ως virtual (CDC) serial / COM port.

Ιστορικά, η διασύνδεση του Arduino με πολύπλοκα web services ήταν μια πρόκληση λόγω της περιορισμένης διαθέσιμης μνήμης. Τα web services τείνουν να χρησιμοποιούν λεπτομερείς μορφές κειμένου, όπως XML, οι οποίες απαιτούν αρκετή ram για να αναλυθούν. Στο Arduino Yun υπάρχει η βιβλιοθήκη Bridge, η οποία μεταβιβάζει όλες τις συνδέσεις δικτύου και επεξεργασίας των συναλλαγών HTTP στο Linux επεξεργαστή. Δηλαδή, η βιβλιοθήκη Bridge διευκολύνει την επικοινωνία μεταξύ των δύο επεξεργαστών δίνοντας έτσι τη δυνατότητα σε προγράμματα του Arduino να τρέξουν shell scripts, να επικοινωνούν με διασυνδέσεις δικτύου και να λαμβάνουν πληροφορίες από τον επεξεργαστή AR9331. Παρακάτω ακολουθεί εικόνα η οποία εξηγεί την επικοινωνία μεταξύ των δύο επεξεργαστών μέσω της βιβλιοθήκης Bridge.



4.4 Υλικά

1. Ένα Arduino Yun για την σύνδεση μέσω Wi-Fi.

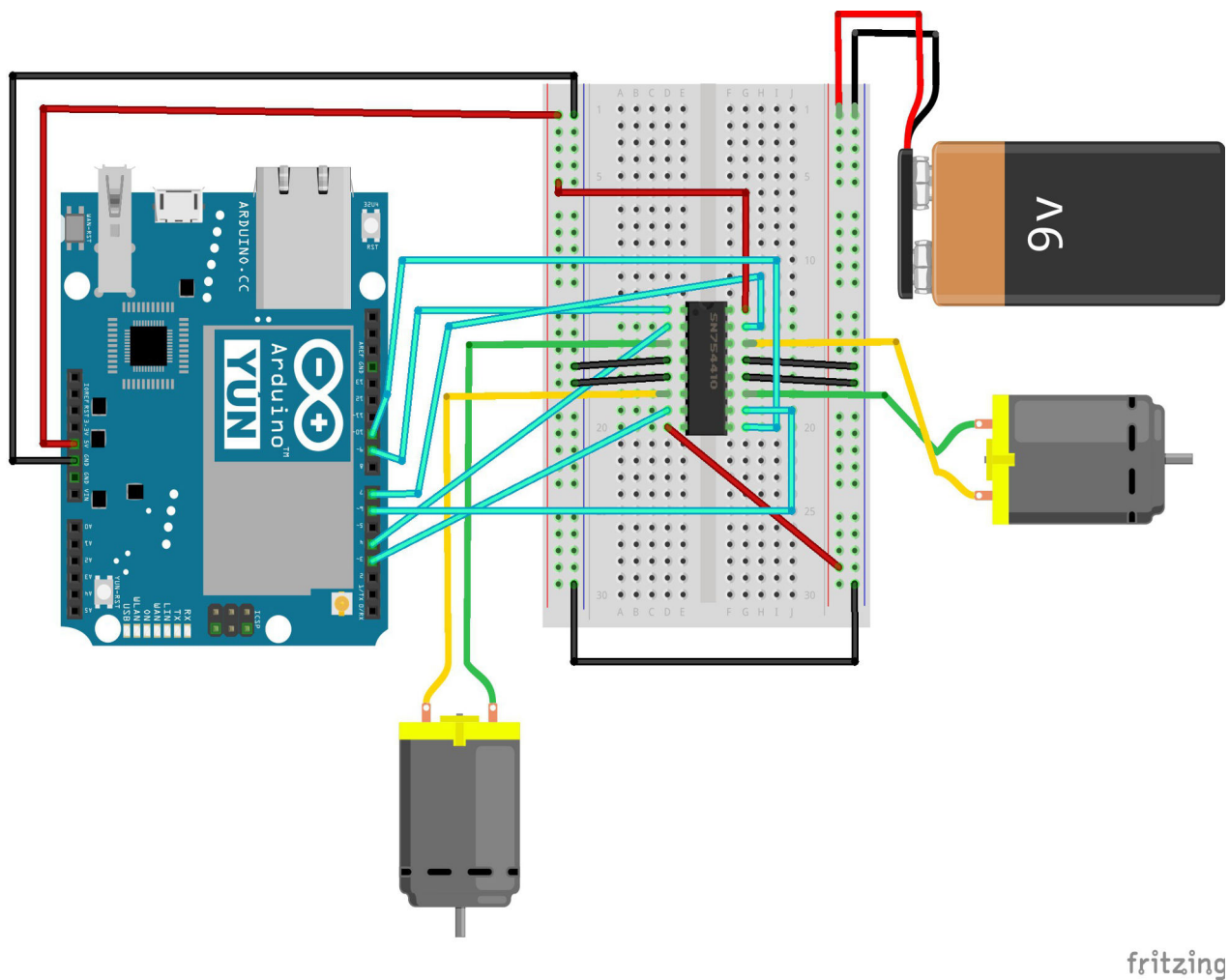


2. Τα υπόλοιπα υλικά που χρησιμοποιήθηκαν έχουν αναφερθεί στο πρώτο μέρος της εργασίας

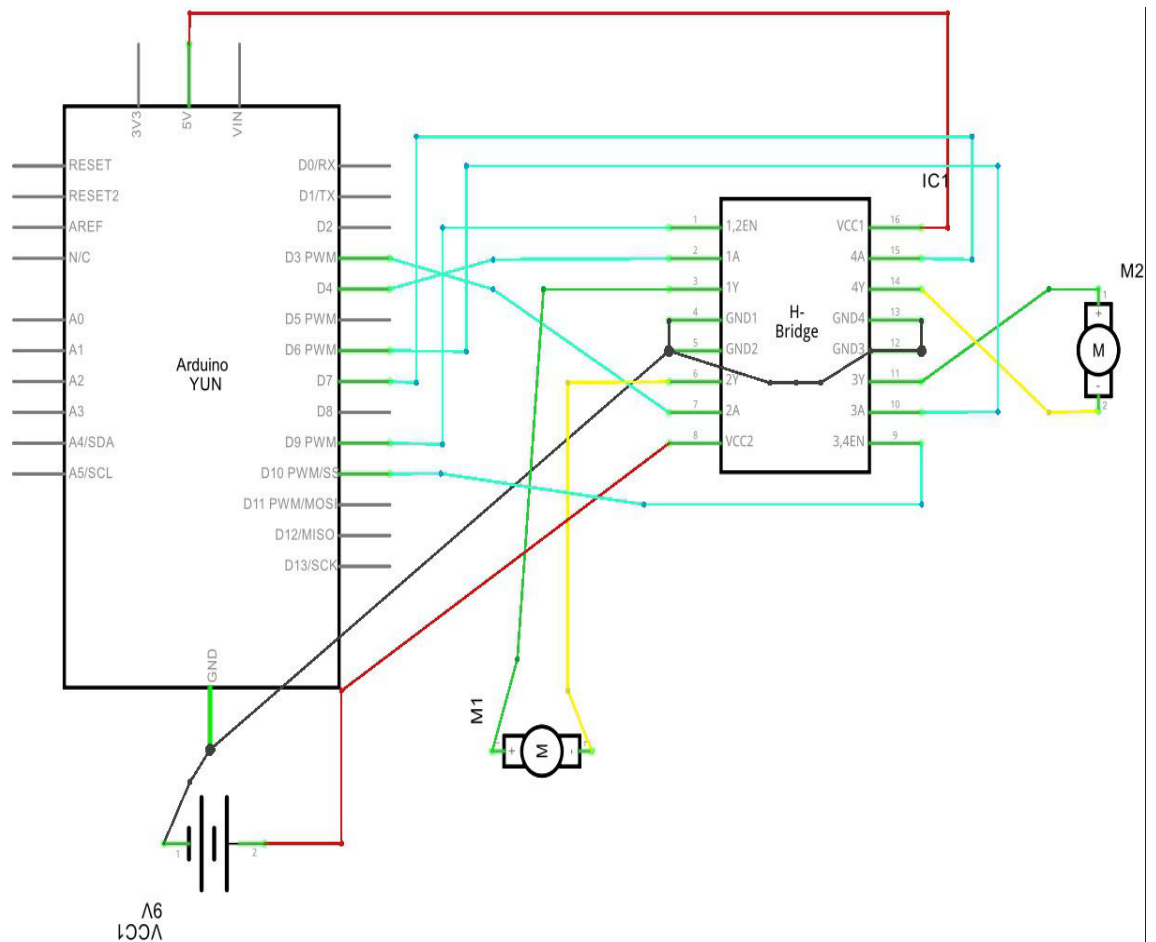
αυτής, όπου και μελετήθηκε η κίνηση του αμαξιδίου.

4.5 Συνδεσμολογία

Το μόνο που αλλάζει σε αυτό το δεύτερο μέρος στη συνδεσμολογία είναι η πλακέτα Arduino, καθώς εδώ χρησιμοποίησα Arduino Yun. Οπότε, η συνδεσμολογία στη breadboard είναι η εξής:



Ενώ το σχέδιο είναι το εξής:



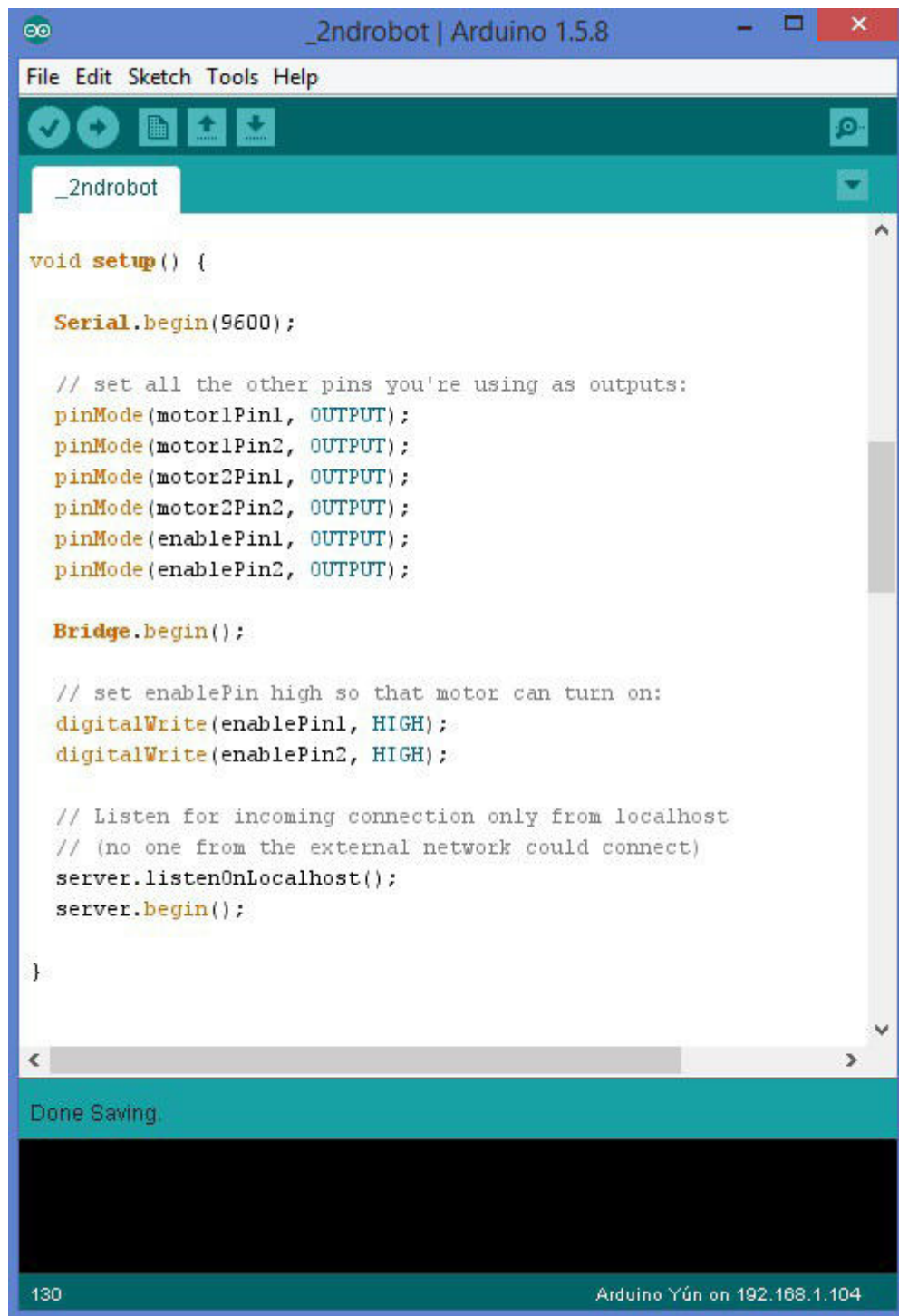
fritzing

4.6 Κώδικας

Για να υπάρξει επικοινωνία μέσω Wi-Fi θα πρέπει να υπάρχει ένας Client και ένας Server. Οπότε, στην πλακέτα βρίσκεται ο Server, ενώ μιά ιστοσελίδα αποτελεί τον Client. Παρακάτω βρίσκεται ο κώδικας για τον Server:

The image is a screenshot of the Arduino IDE interface. The title bar at the top reads "_2ndrobot | Arduino 1.5.8". Below the title bar is a menu bar with "File", "Edit", "Sketch", "Tools", and "Help". Underneath the menu bar is a toolbar with icons for opening, saving, and uploading files. The main text area contains the code for the "_2ndrobot" sketch. The code includes comments about the filename, developer, and revision, followed by a summary of the code's functionality. It then includes the necessary libraries: <Bridge.h>, <YunServer.h>, and <YunClient.h>. The code defines a YunServer object and a readString function. It also defines constants for motor pins and enable pins. Finally, it declares variables for motor value, pot value, and delay time. The status bar at the bottom shows "1" on the left and "Arduino Yun on 192.168.1.104" on the right.

```
//  
//  
//Filename: _2ndrobot  
//Developed By: Chris Chatzilenas Email:it21041@hua.gr  
//Last Code Revision: 06/11/2014  
//  
/////////////////////////////////////  
//  
//Summary: With this code a robot with two DC motors can go straig  
//turn right,turn left ,go backwards and stop  
//  
/////////////////////////////////////  
//  
//Digital Pins that used: 3,4,6,7,9,10  
//  
/////////////////////////////////////  
  
#include <Bridge.h>  
#include <YunServer.h>  
#include <YunClient.h>  
  
YunServer server;  
String readString;  
  
> const int motor1Pin1 = 3;    // H-bridge leg1 (pin 2, 1A)  
const int motor1Pin2 = 4;    // H-bridge leg2 (pin 7, 2A)  
const int motor2Pin1= 7;    // H-bridge leg2 (pin 15, 4A)  
const int motor2Pin2= 6;    // H-bridge leg1 (pin 10, 3A)  
const int enablePin1 = 9;    // H-bridge enable motor1  
const int enablePin2= 10;    // H-bridge enable motor2  
  
int motorValue=0;  
int potValue=0;  
int delayt=5000;    //delay time  
  
< 
```





```
void loop() {

  // Get clients coming from server
  YunClient client = server.accept();

  // There is a new client?
  if (client) {
    // read the command
    String command = client.readString();
    command.trim();          //kill whitespace
    Serial.println(command);

    if(command=="forward"){
      //drive straight:
      //set leg 2 of the 2 motors HIGH to go straight
      digitalWrite(motor1Pin1, LOW);
      digitalWrite(motor1Pin2, HIGH);

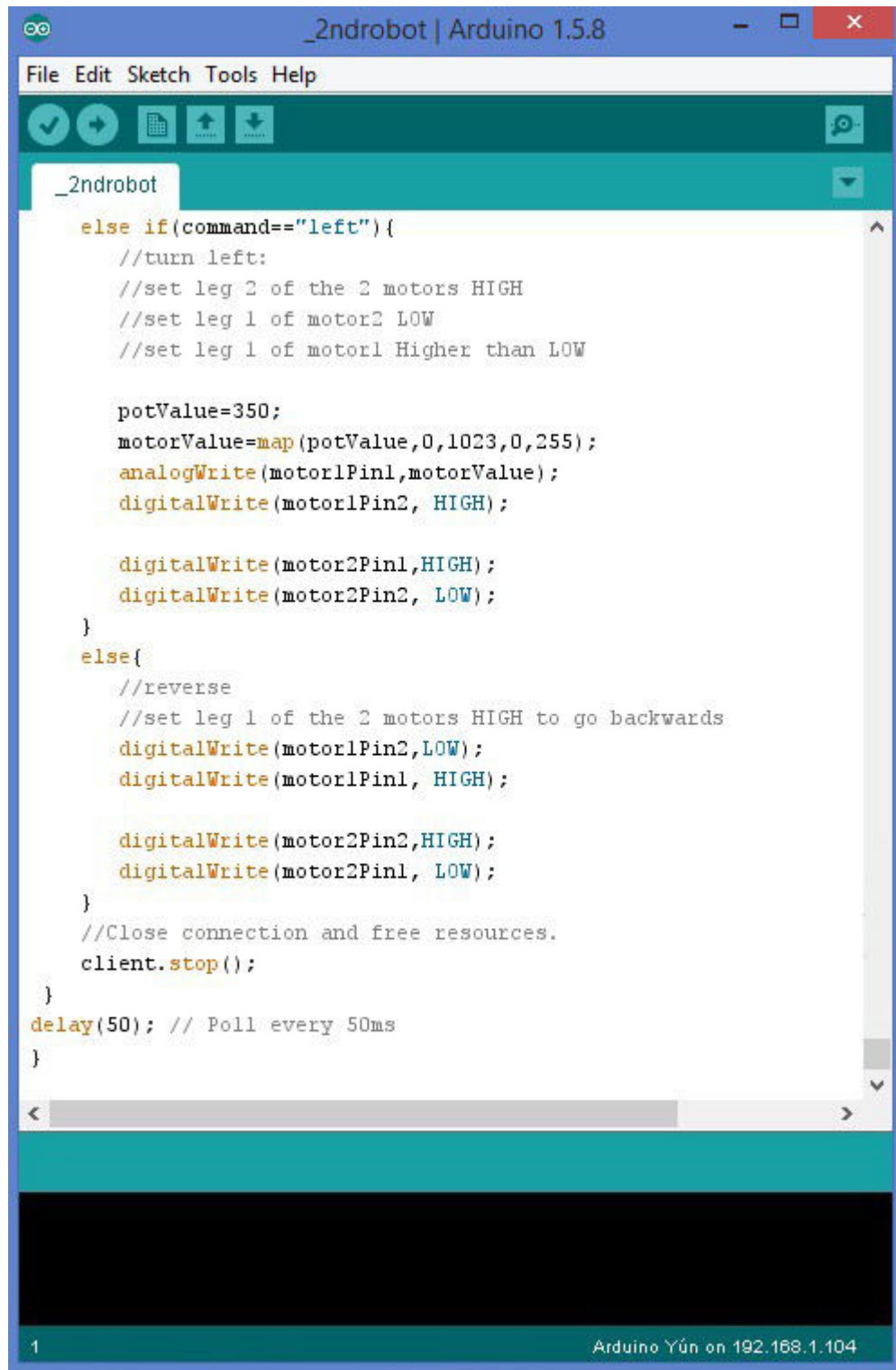
      digitalWrite(motor2Pin1, HIGH);
      digitalWrite(motor2Pin2, LOW);
    }
    else if(command=="stop"){
      //stop
      //set leg 1,2 of the 2 motors HIGH or LOW to stop
      digitalWrite(motor1Pin1, HIGH);
      digitalWrite(motor1Pin2, HIGH);

      digitalWrite(motor2Pin1, HIGH);
      digitalWrite(motor2Pin2, HIGH);
    }
    else if(command=="right"){
      //turn right:
      //set leg 2 of the 2 motors HIGH
      //set leg 1 of motor1 LOW
      //set leg 1 of motor2 Higher than LOW
      digitalWrite(motor1Pin1, LOW);
      digitalWrite(motor1Pin2, HIGH);

      potValue=350;
      motorValue=map(potValue,0,1023,0,255);

      digitalWrite(motor2Pin1, HIGH);
      analogWrite(motor2Pin2, motorValue);
    }
  }
}
```

1 Arduino Yun on 192.168.1.104



```
File Edit Sketch Tools Help

_2ndrobot

else if(command=="left"){
    //turn left:
    //set leg 2 of the 2 motors HIGH
    //set leg 1 of motor2 LOW
    //set leg 1 of motor1 Higher than LOW

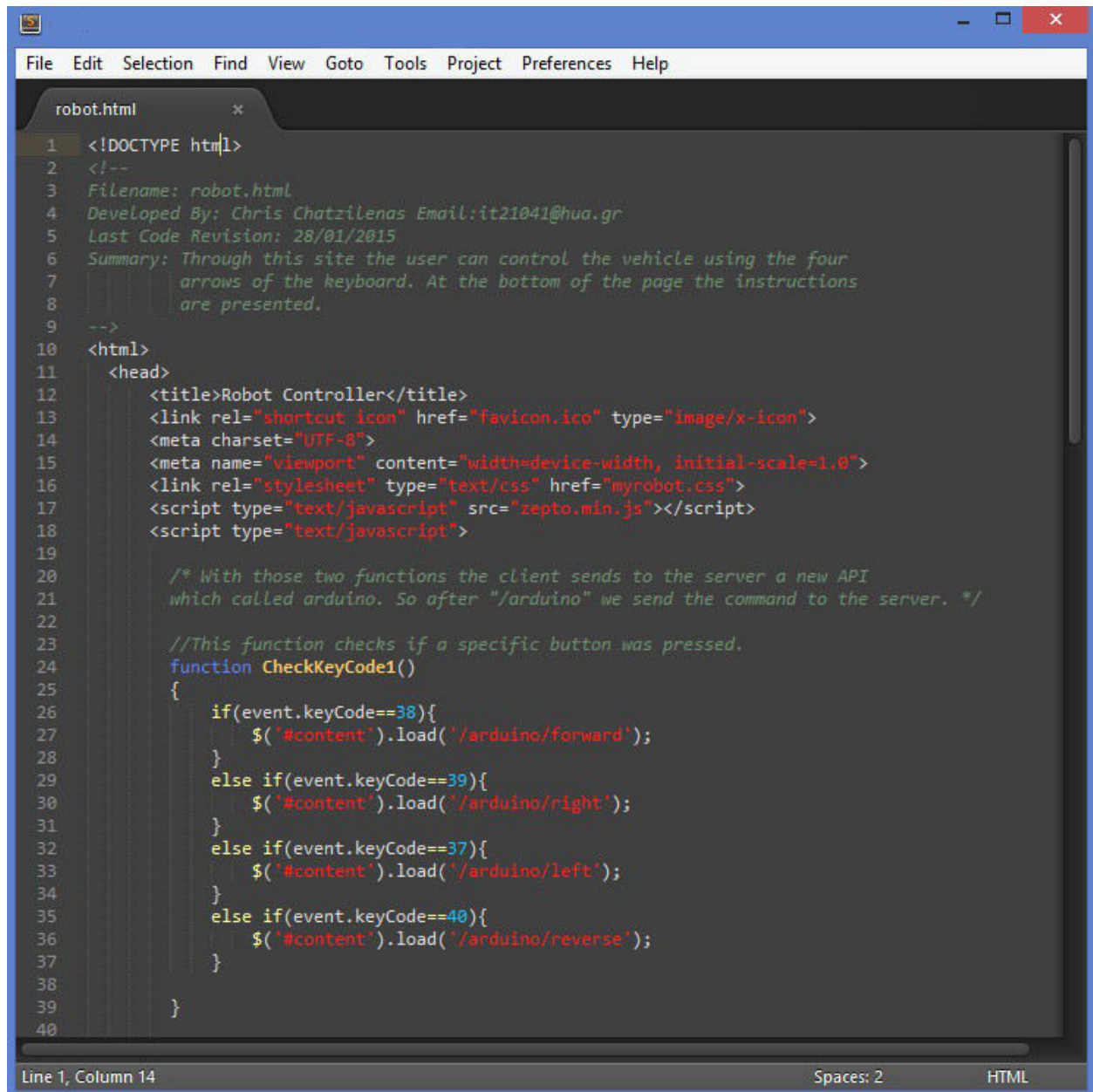
    potValue=350;
    motorValue=map(potValue,0,1023,0,255);
    analogWrite(motor1Pin1,motorValue);
    digitalWrite(motor1Pin2, HIGH);

    digitalWrite(motor2Pin1,HIGH);
    digitalWrite(motor2Pin2, LOW);
}
else{
    //reverse
    //set leg 1 of the 2 motors HIGH to go backwards
    digitalWrite(motor1Pin2,LOW);
    digitalWrite(motor1Pin1, HIGH);

    digitalWrite(motor2Pin2,HIGH);
    digitalWrite(motor2Pin1, LOW);
}
//Close connection and free resources.
client.stop();
}
delay(50); // Poll every 50ms
}
```

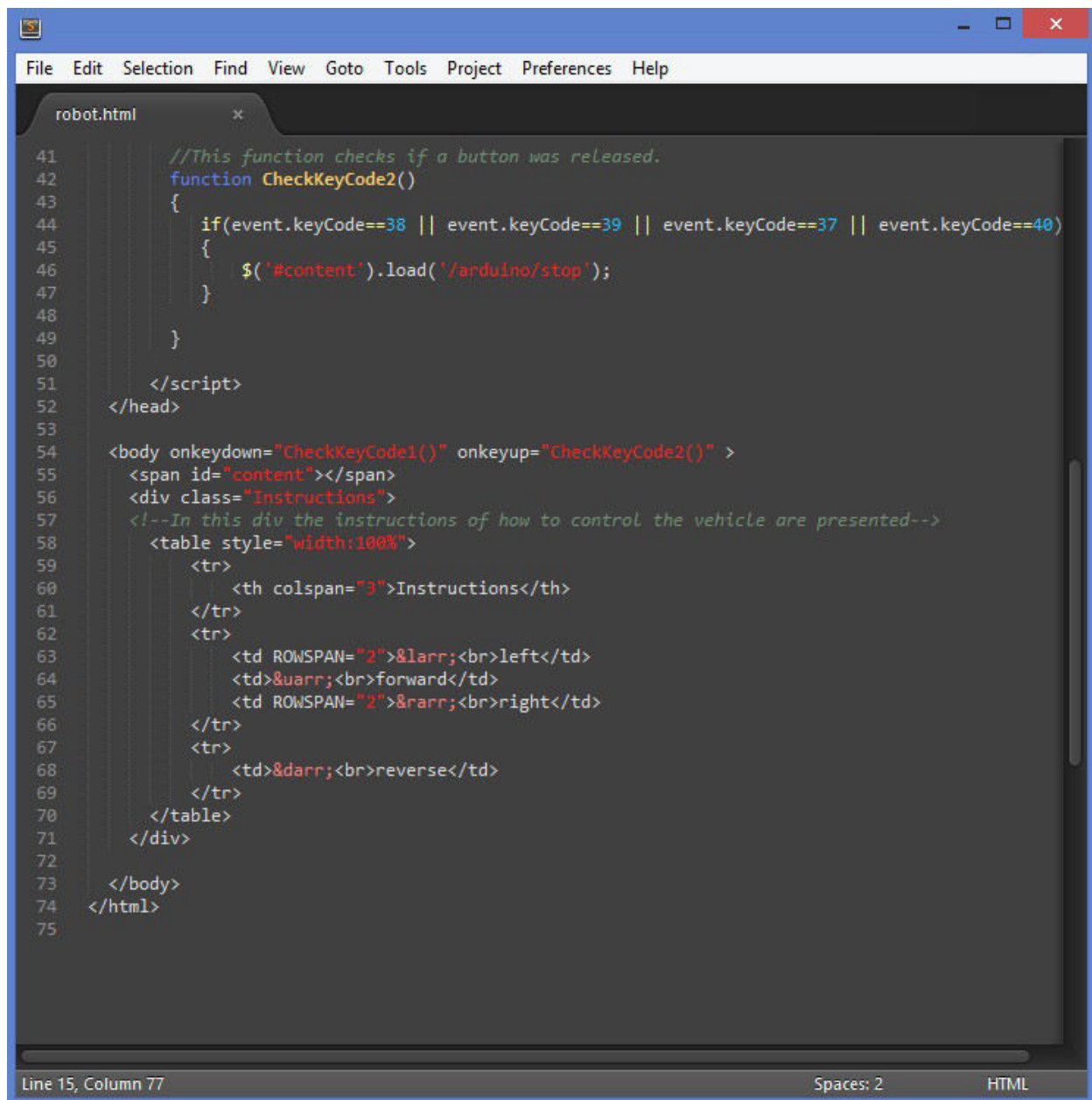
1 Arduino Yún on 192.168.1.104

Ακολουθεί ο κώδικας της ιστοσελίδας (Client):



```
1 <!DOCTYPE html>
2 <!--
3  Filename: robot.html
4  Developed By: Chris Chatzilenas Email:it21041@hua.gr
5  Last Code Revision: 28/01/2015
6  Summary: Through this site the user can control the vehicle using the four
7           arrows of the keyboard. At the bottom of the page the instructions
8           are presented.
9  -->
10 <html>
11   <head>
12     <title>Robot Controller</title>
13     <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
14     <meta charset="UTF-8">
15     <meta name="viewport" content="width=device-width, initial-scale=1.0">
16     <link rel="stylesheet" type="text/css" href="myrobot.css">
17     <script type="text/javascript" src="zepto.min.js"></script>
18     <script type="text/javascript">
19
20       /* With those two functions the client sends to the server a new API
21        which called arduino. So after "/arduino" we send the command to the server. */
22
23       //This function checks if a specific button was pressed.
24       function CheckKeyCode1()
25       {
26         if(event.keyCode==38){
27           $('#content').load('/arduino/forward');
28         }
29         else if(event.keyCode==39){
30           $('#content').load('/arduino/right');
31         }
32         else if(event.keyCode==37){
33           $('#content').load('/arduino/left');
34         }
35         else if(event.keyCode==40){
36           $('#content').load('/arduino/reverse');
37         }
38       }
39
40     </script>
11   </head>
```

Line 1, Column 14 Spaces: 2 HTML

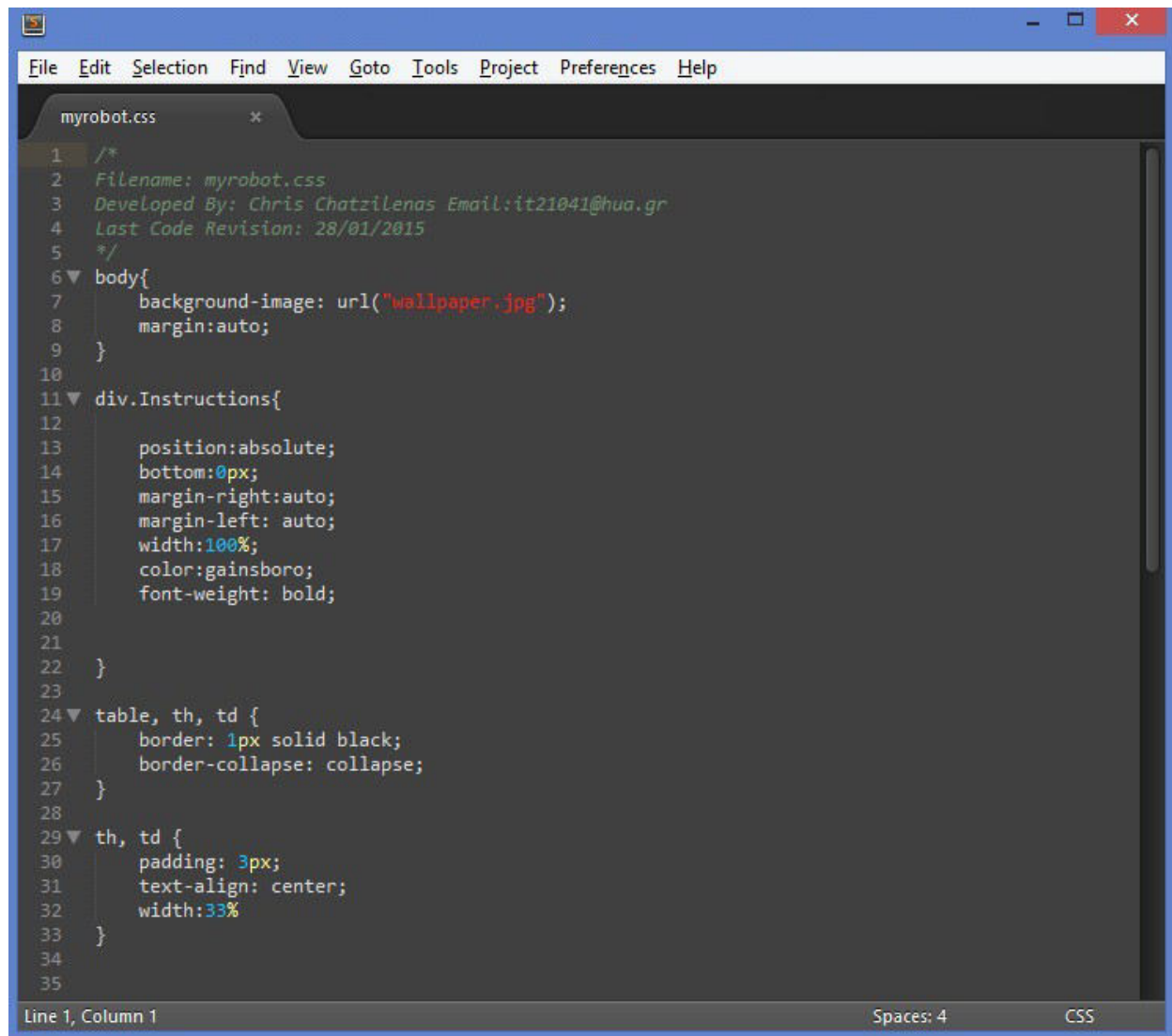


```
41 //This function checks if a button was released.
42 function CheckKeyCode2()
43 {
44     if(event.keyCode==38 || event.keyCode==39 || event.keyCode==37 || event.keyCode==40)
45     {
46         $('#content').load('/arduino/stop');
47     }
48 }
49
50
51 </script>
52 </head>
53
54 <body onkeydown="CheckKeyCode1()" onkeyup="CheckKeyCode2()" >
55     <span id="content"></span>
56     <div class="Instructions">
57         <!--In this div the instructions of how to control the vehicle are presented-->
58         <table style="width:100%">
59             <tr>
60                 <th colspan="3">Instructions</th>
61             </tr>
62             <tr>
63                 <td ROWSPAN="2">&larr;<br>left</td>
64                 <td>&uarr;<br>forward</td>
65                 <td ROWSPAN="2">&rarr;<br>right</td>
66             </tr>
67             <tr>
68                 <td>&darr;<br>reverse</td>
69             </tr>
70         </table>
71     </div>
72
73 </body>
74 </html>
75
```

Line 15, Column 77 Spaces: 2 HTML

Σε αυτό το σημείο θα ήταν σωστό να αναφερθώ στο αρχείο zepto.min.js που χρησιμοποίησα. Το Zepto, λοιπόν, είναι μια μινιμαλιστική βιβλιοθήκη Javascript για σύγχρονα προγράμματα περιήγησης, με jQuery-συμβατό API. Μοιάζει αρκετά με jQuery, οπότε είναι εύκολο να χρησιμοποιηθεί. Ενώ, η 100% κάλυψη του jQuery δεν είναι ο στόχος του σχεδιασμού, τα APIs που παρέχονται ταιριάζουν με αντίστοιχα jQuery. Ο στόχος του Zepto είναι να έχουμε μια ~ 5-10k βιβλιοθήκη που κατεβάζει και εκτελεί γρήγορα, με ένα γνώριμο και ευέλικτο API, ώστε ο χρήστης να μπορεί να επικεντρωθεί στην ολοκλήρωση των υπόλοιπων εργασιών του. Το Zepto είναι λογισμικό ανοιχτού κώδικα και διατίθεται σε προγραμματιστές και εταιρίες με την άδεια του MIT (MIT license).

Ακολουθεί το css της ιστοσελίδας:



```
1  /*
2  Filename: myrobot.css
3  Developed By: Chris Chatzilenas Email:it21041@hua.gr
4  Last Code Revision: 28/01/2015
5  */
6  body{
7      background-image: url("wallpaper.jpg");
8      margin:auto;
9  }
10
11  div.Instructions{
12
13      position:absolute;
14      bottom:0px;
15      margin-right:auto;
16      margin-left: auto;
17      width:100%;
18      color:gainsboro;
19      font-weight: bold;
20
21
22  }
23
24  table, th, td {
25      border: 1px solid black;
26      border-collapse: collapse;
27  }
28
29  th, td {
30      padding: 3px;
31      text-align: center;
32      width:33%
33  }
34
35
```

Line 1, Column 1

Spaces: 4

CSS

Η μορφή της ιστοσελίδας σε αυτό το στάδιο ήταν:

Webpage Screenshot



Μέρος Τρίτο – Μελέτη Web-Camera



5.1 Εισαγωγή

Στο τρίτο μέρος πρόσθεσα στο αμαξίδιο μια ψηφιακή κάμερα με την οποία θα κάνω ζωντανή μετάδοση στην ιστοσελίδα που έχω φτιάξει, αλλά θα μπορεί ο χρήστης να τραβήξει και φωτογραφίες, οι οποίες αποθηκεύονται στην κάρτα μνήμης του Arduino Yun. Για να μπορέσει να λειτουργήσει η κάμερα έπρεπε να εγκαταστήσω κάποια πακέτα λογισμικού. Συνδέθηκα με την πλακέτα μέσω ssh, στον linux επεξεργαστή και «έτρεξα» την παρακάτω εντολή έτσι ώστε να εγκαταστήσω κάποιους οδηγούς UVC:

```
opkg install kmod-video-uv
```

Στη συνέχεια, «έτρεξα» την παρακάτω εντολή έτσι ώστε να εγκαταστήσω τη βιβλιοθήκη fswebcam με την οποία μπορεί εύκολα οποιοσδήποτε να τραβήξει φωτογραφία από το τερματικό:

```
opkg install fswebcam
```

Οπότε, με την εντολή:

```
fswebcam mnt/sda1/test.png
```

μπορεί εύκολα να βγεί μια φωτογραφία από το τερματικό. Στη συνέχεια, εγκατέστησα τη βιβλιοθήκη mjpg streaming για να μπορέσω να κάνω τη ζωντανή μετάδοση με την ακόλουθη εντολή:

```
opkg install mjpg-streamer
```

Μετά από αυτό, λόγω του πακέτου που εγκατέστησα είναι εύκολο να γίνει streaming τοπικά, δίνοντας την παρακάτω εντολή στον επεξεργαστή linux:

```
mjpg_streamer -i "input_uv.so -d /dev/video0 -r 640x480 -f 25" -o "output_http.so -p 8080  
-w /www/webcam" &.
```

Το γνώρισμα **-r** αρχικοποιεί την ανάλυση του βίντεο (χρησιμοποίησα τη χαμηλότερη γιατί σε υψηλότερη ανάλυση το βίντεο καθυστερούσε), ενώ το γνώρισμα **-p** αρχικοποιεί τη θύρα στην οποία θα είναι διαθέσιμο το stream. Εκτός από τη ψηφιακή κάμερα, συνέδεσα στην πλακέτα ένα

servo των 180 μοιρών, έτσι ώστε πάνω του να τοποθετηθεί η κάμερα και να γυρνά ανάλογα με τις εντολές που δίνει ο χρήστης από το πληκτρολόγιο. Οι εντολές που θα μπορεί να δώσει ο χρήστης είναι:

1. να στρίψει την κάμερα δεξιά ή αριστερά κατά 20 μοίρες κάθε φορά
2. να την επαναφέρει στο κέντρο

5.2 Υλικά

1. Servo των 180 μοιρών έτσι ώστε η κάμερα να γυρνά ανάλογα με τις εντολές που δίνει ο χρήστης.



2. Έναν πυκνωτή 100uF 25V.

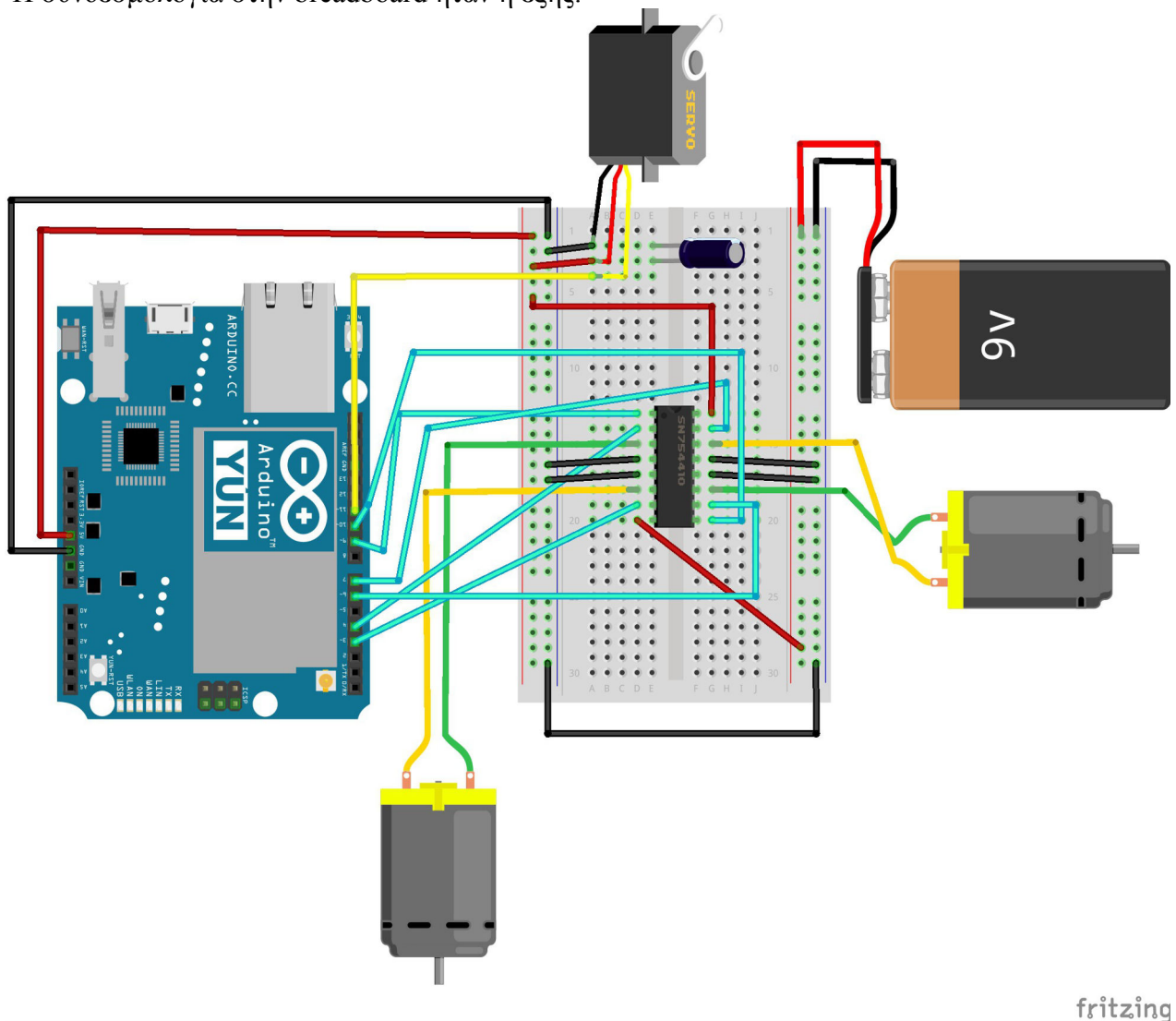


3. Μια κάμερα Logitech Webcam C160.



5.3 Συνδεσμολογία

Η συνδεσμολογία στην breadboard ήταν η εξής:

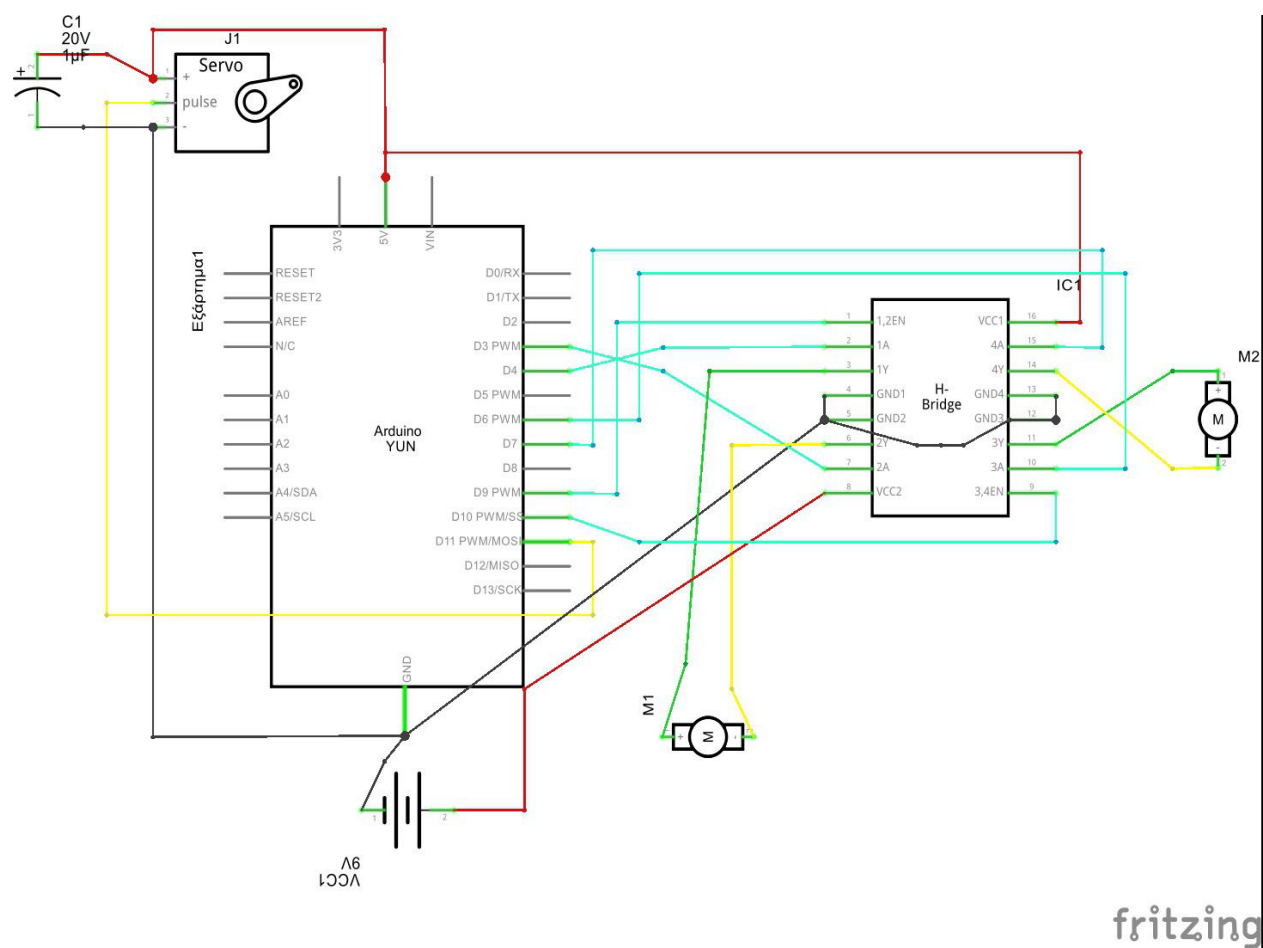


fritzing

Η μοναδική αλλαγή από το προηγούμενο μέρος είναι η προσθήκη του Servo στη θέση a1 της breadboard (το καλώδιο της γείωσης του Servo). Όπως φαίνεται και στο σχήμα, το μαύρο καλώδιο του Servo είναι η γείωση, το κόκκινο καλώδιο είναι το ρεύμα και το κίτρινο καλώδιο είναι συνδεδεμένο με το pin 11 του Arduino. Τέλος, πρόσθεσα ένα πυκνωτή ανάμεσα στο ρεύμα και στη γείωση, διότι όταν ένας σερβοκινητήρας αρχίζει να κινείται, αντλεί περισσότερο ρεύμα απ' ότι αν ήταν ήδη σε κίνηση. Με την τοποθέτηση ενός πυκνωτή μπορούμε να εξομαλύνουμε τις αλλαγές τάσης που μπορεί να προκύψουν. Αυτοί οι πυκνωτές ονομάζονται πυκνωτές απόζευξης επειδή μειώνουν ή εξομαλύνουν τις αλλαγές που προκαλούνται από τα εξαρτήματα που αποτελούν το υπόλοιπο κύκλωμα. Σε αυτό το σημείο, έπρεπε να προσέξω έτσι ώστε να συνδέσω την κάθοδο του πυκνωτή στη γείωση και την άνοδο στο ρεύμα. Σε αντίθετη

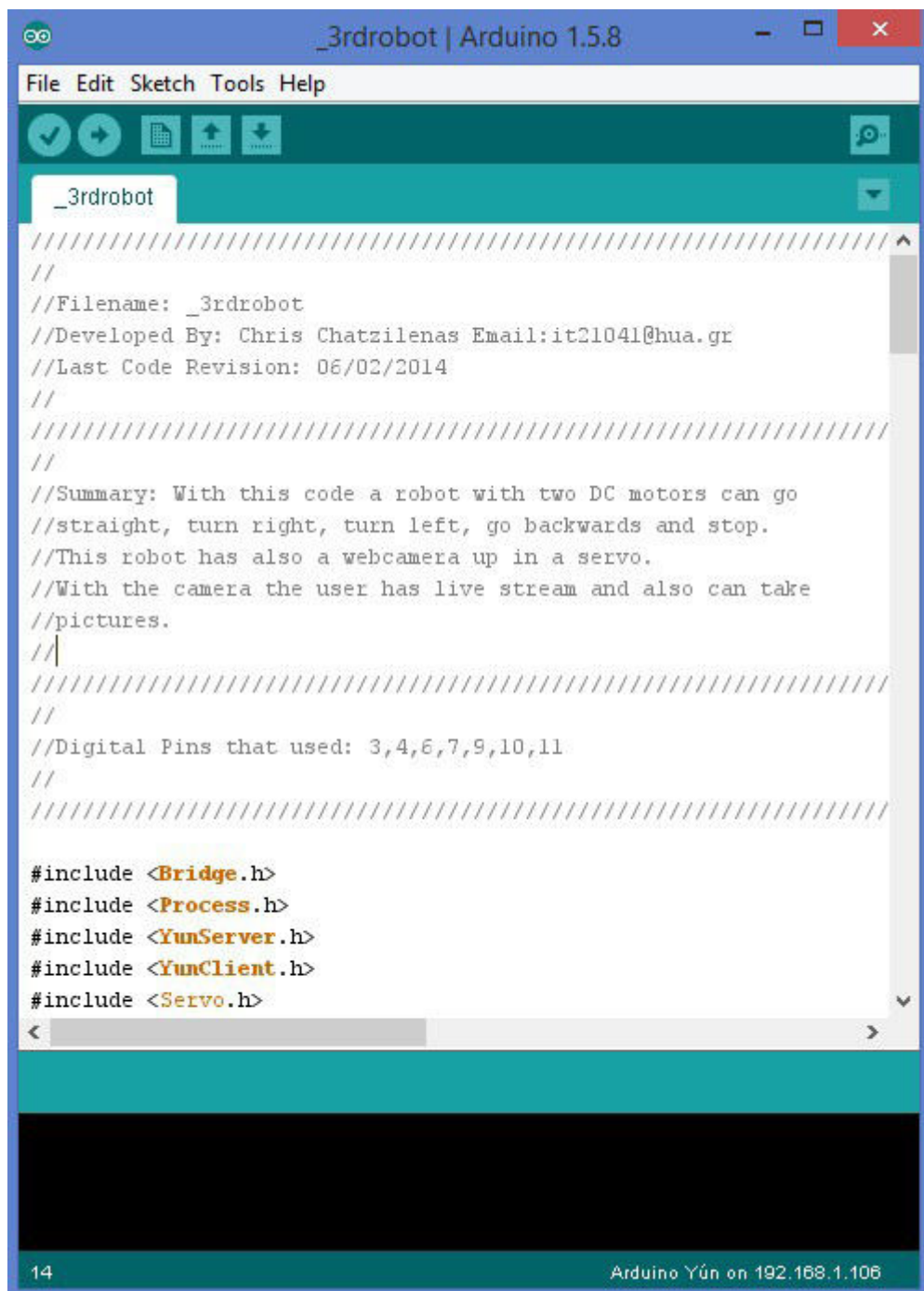
περίπτωση, μπορεί να εκραγούν.

Παρακάτω ακολουθεί το σχέδιο:



5.4 Κώδικας

Ο κώδικας για το Server:



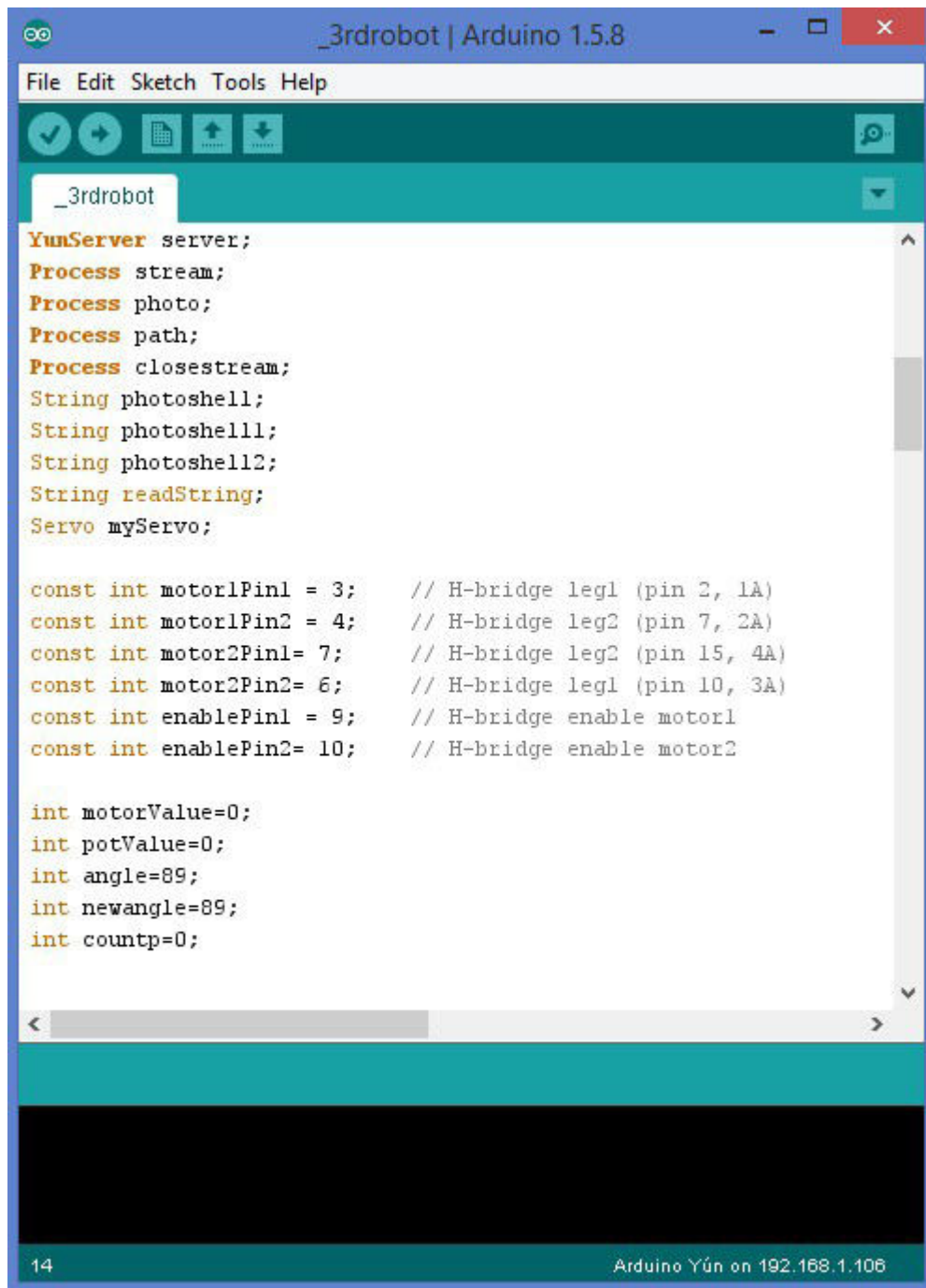
```
File Edit Sketch Tools Help

_3drobot

////////////////////////////////////
//
//Filename: _3drobot
//Developed By: Chris Chatzilenas Email:it21041@hua.gr
//Last Code Revision: 06/02/2014
//
////////////////////////////////////
//
//Summary: With this code a robot with two DC motors can go
//straight, turn right, turn left, go backwards and stop.
//This robot has also a webcam up in a servo.
//With the camera the user has live stream and also can take
//pictures.
//
////////////////////////////////////
//
//Digital Pins that used: 3,4,6,7,9,10,11
//
////////////////////////////////////

#include <Bridge.h>
#include <Process.h>
#include <YunServer.h>
#include <YunClient.h>
#include <Servo.h>
```

14 Arduino Yun on 192.168.1.106



```
File Edit Sketch Tools Help

_YunServer server;
Process stream;
Process photo;
Process path;
Process closestream;
String photoshell;
String photoshell1;
String photoshell2;
String readString;
Servo myServo;

const int motor1Pin1 = 3; // H-bridge leg1 (pin 2, 1A)
const int motor1Pin2 = 4; // H-bridge leg2 (pin 7, 2A)
const int motor2Pin1 = 7; // H-bridge leg2 (pin 15, 4A)
const int motor2Pin2 = 6; // H-bridge leg1 (pin 10, 3A)
const int enablePin1 = 9; // H-bridge enable motor1
const int enablePin2 = 10; // H-bridge enable motor2

int motorValue=0;
int potValue=0;
int angle=89;
int newangle=89;
int countp=0;
```

14 Arduino Yun on 192.168.1.106



```
void setup() {  
  
  Serial.begin(9600);  
  myServo.attach(11);  
  
  // set all the other pins you're using as outputs:  
  pinMode(motor1Pin1, OUTPUT);  
  pinMode(motor1Pin2, OUTPUT);  
  pinMode(motor2Pin1, OUTPUT);  
  pinMode(motor2Pin2, OUTPUT);  
  pinMode(enablePin1, OUTPUT);  
  pinMode(enablePin2, OUTPUT);  
  
  Bridge.begin();  
  
  //open live stream  
  stream.runShellCommand("mjpg_streamer -i 'input_uvc.so -d  
/dev/video0 -r 640x480 -f 25' -o 'output_http.so -p 8080 -w /www/webcam' &");  
  // set enablePin high so that motor can turn on:  
  digitalWrite(enablePin1, HIGH);  
  digitalWrite(enablePin2, HIGH);  
  
  //put servo to 89  
  myServo.write(angle);  
  
  // Listen for incoming connection only from localhost  
  // (no one from the external network could connect)  
  server.listenOnLocalhost();  
  server.begin();  
  
}
```

14 Arduino Yun on 192.168.1.106



```
void loop() {

  // Get clients coming from server
  YumClient client = server.accept();

  // There is a new client?
  if (client) {
    // read the command
    String command = client.readString();
    command.trim();           //kill whitespace
    Serial.println(command);

    if(command=="forward"){
      //drive straight:
      //set leg 2 of the 2 motors HIGH to go straight
      digitalWrite(motor1Pin1,LOW);
      digitalWrite(motor1Pin2, HIGH);

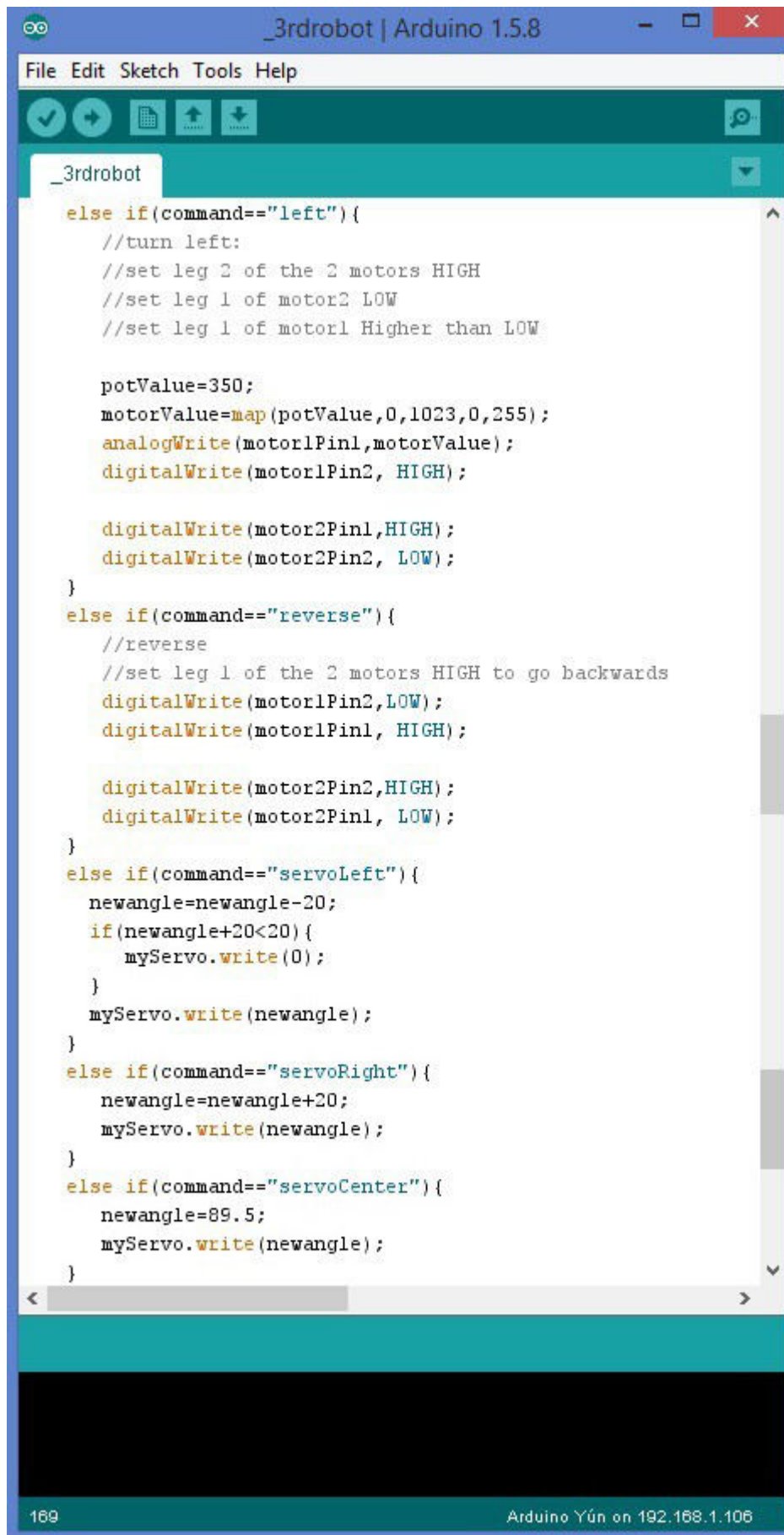
      digitalWrite(motor2Pin1,HIGH);
      digitalWrite(motor2Pin2, LOW);
    }
    else if(command=="stop"){
      //stop
      //set leg 1,2 of the 2 motors HIGH or LOW to stop
      digitalWrite(motor1Pin1,HIGH);
      digitalWrite(motor1Pin2, HIGH);

      digitalWrite(motor2Pin1,HIGH);
      digitalWrite(motor2Pin2, HIGH);
    }
    else if(command=="right"){
      //turn right:
      //set leg 2 of the 2 motors HIGH
      //set leg 1 of motor1 LOW
      //set leg 1 of motor2 Higher than LOW
      digitalWrite(motor1Pin1,LOW);
      digitalWrite(motor1Pin2, HIGH);

      potValue=350;
      motorValue=map(potValue,0,1023,0,255);

      digitalWrite(motor2Pin1,HIGH);
      analogWrite(motor2Pin2, motorValue);
    }
  }
}
```

14 Arduino Yún on 192.168.1.106



```
File Edit Sketch Tools Help

_3drobot

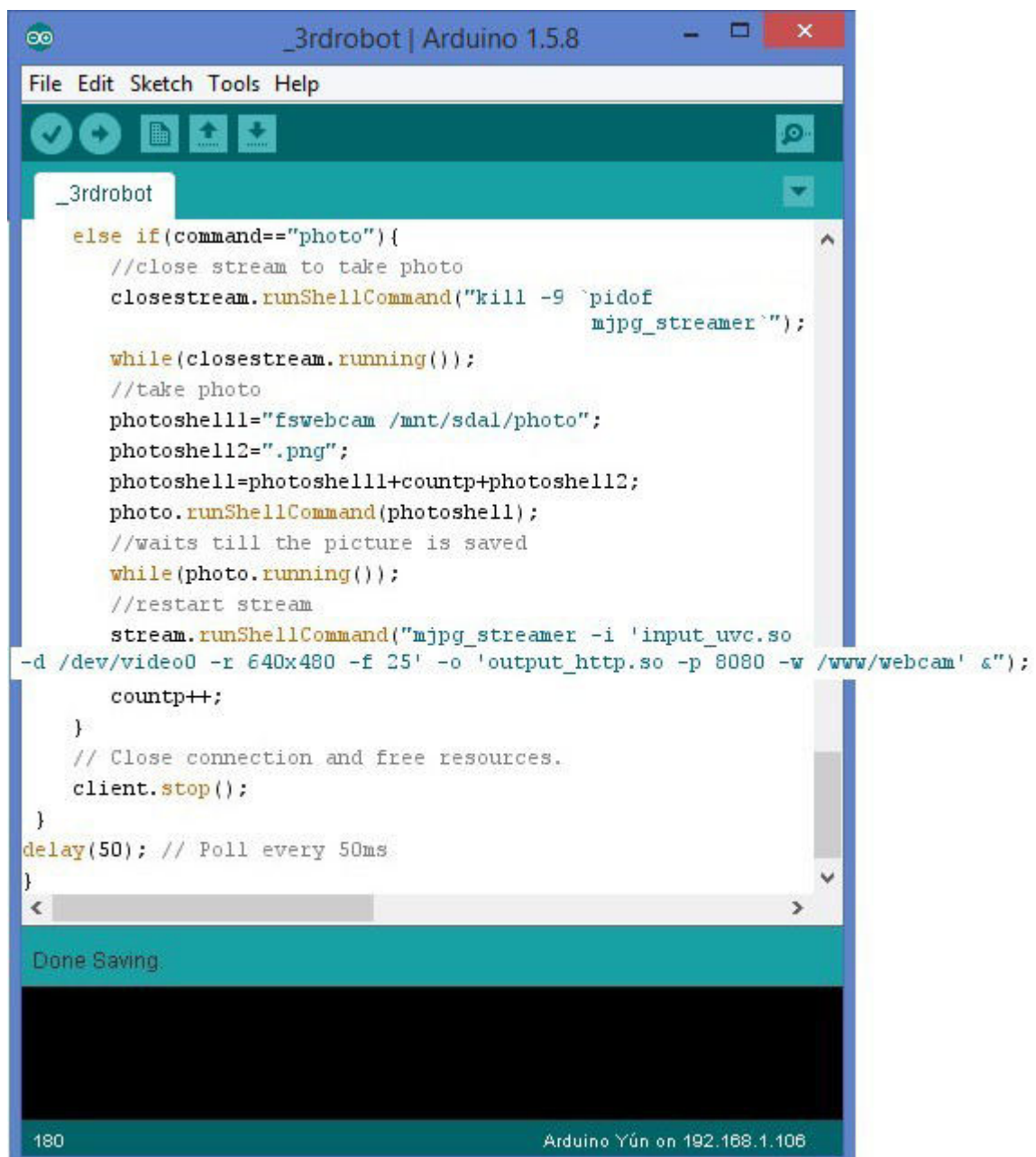
else if(command=="left"){
  //turn left:
  //set leg 2 of the 2 motors HIGH
  //set leg 1 of motor2 LOW
  //set leg 1 of motor1 Higher than LOW

  potValue=350;
  motorValue=map(potValue,0,1023,0,255);
  analogWrite(motor1Pin1,motorValue);
  digitalWrite(motor1Pin2, HIGH);

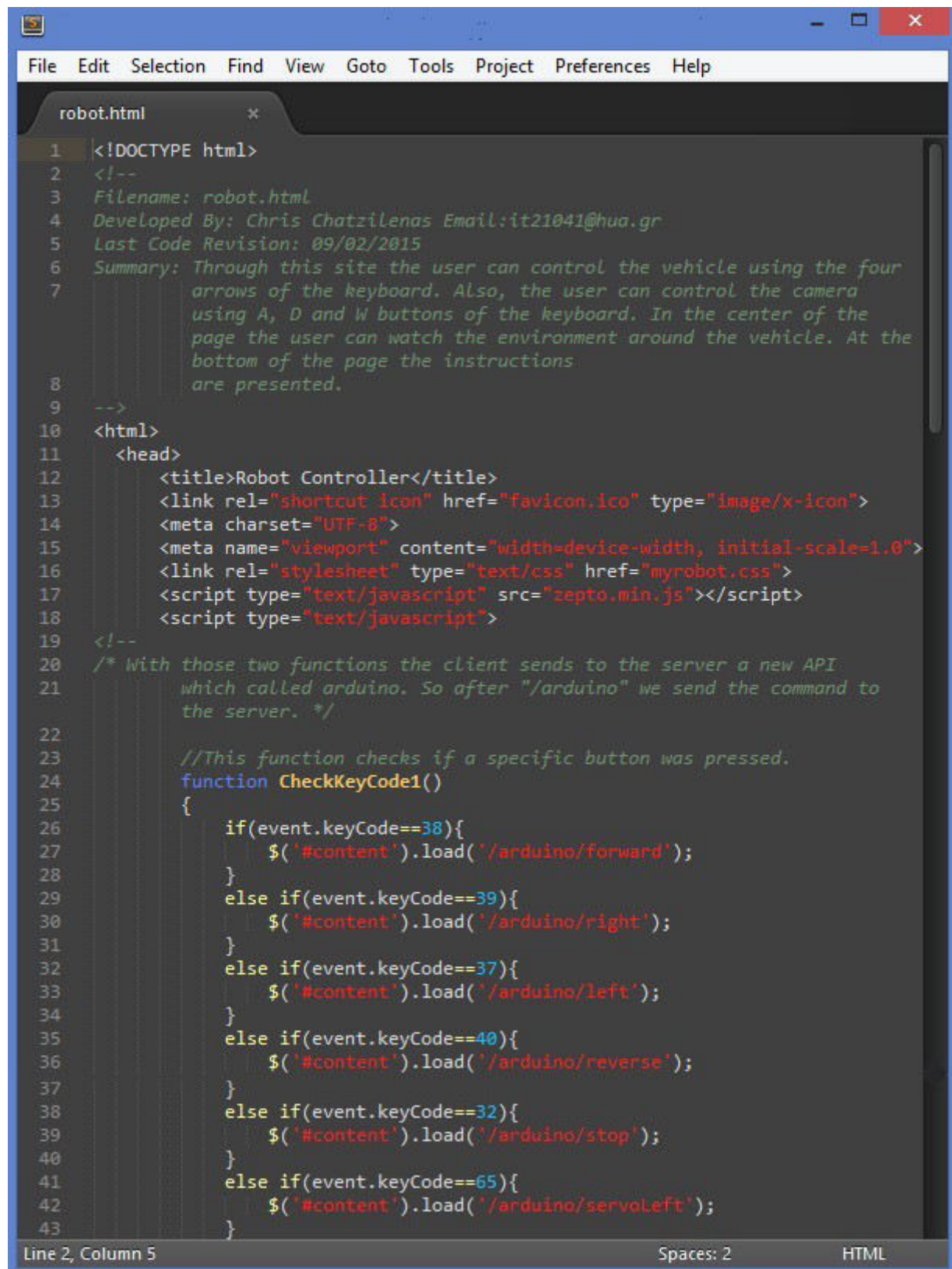
  digitalWrite(motor2Pin1,HIGH);
  digitalWrite(motor2Pin2, LOW);
}
else if(command=="reverse"){
  //reverse
  //set leg 1 of the 2 motors HIGH to go backwards
  digitalWrite(motor1Pin2,LOW);
  digitalWrite(motor1Pin1, HIGH);

  digitalWrite(motor2Pin2,HIGH);
  digitalWrite(motor2Pin1, LOW);
}
else if(command=="servoLeft"){
  newangle=newangle-20;
  if(newangle+20<20){
    myServo.write(0);
  }
  myServo.write(newangle);
}
else if(command=="servoRight"){
  newangle=newangle+20;
  myServo.write(newangle);
}
else if(command=="servoCenter"){
  newangle=89.5;
  myServo.write(newangle);
}
}

169 Arduino Yún on 192.168.1.106
```



Ακολουθεί ο κώδικας της ιστοσελίδας (Client):



```
1 <!DOCTYPE html>
2 <!--
3  Filename: robot.html
4  Developed By: Chris Chatzilenas Email:it21041@hua.gr
5  Last Code Revision: 09/02/2015
6  Summary: Through this site the user can control the vehicle using the four
7           arrows of the keyboard. Also, the user can control the camera
8           using A, D and W buttons of the keyboard. In the center of the
9           page the user can watch the environment around the vehicle. At the
10          bottom of the page the instructions
11          are presented.
12 -->
13 <html>
14   <head>
15     <title>Robot Controller</title>
16     <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
17     <meta charset="UTF-8">
18     <meta name="viewport" content="width=device-width, initial-scale=1.0">
19     <link rel="stylesheet" type="text/css" href="myrobot.css">
20     <script type="text/javascript" src="zepto.min.js"></script>
21     <script type="text/javascript">
22
23       /* With those two functions the client sends to the server a new API
24       which called arduino. So after "/arduino" we send the command to
25       the server. */
26
27       //This function checks if a specific button was pressed.
28       function CheckKeyCode1()
29       {
30         if(event.keyCode==38){
31           $('#content').load('/arduino/forward');
32         }
33         else if(event.keyCode==39){
34           $('#content').load('/arduino/right');
35         }
36         else if(event.keyCode==37){
37           $('#content').load('/arduino/left');
38         }
39         else if(event.keyCode==40){
40           $('#content').load('/arduino/reverse');
41         }
42         else if(event.keyCode==32){
43           $('#content').load('/arduino/stop');
44         }
45         else if(event.keyCode==65){
46           $('#content').load('/arduino/servoLeft');
47         }
48       }
49     </script>
50   </head>
51 </html>
```

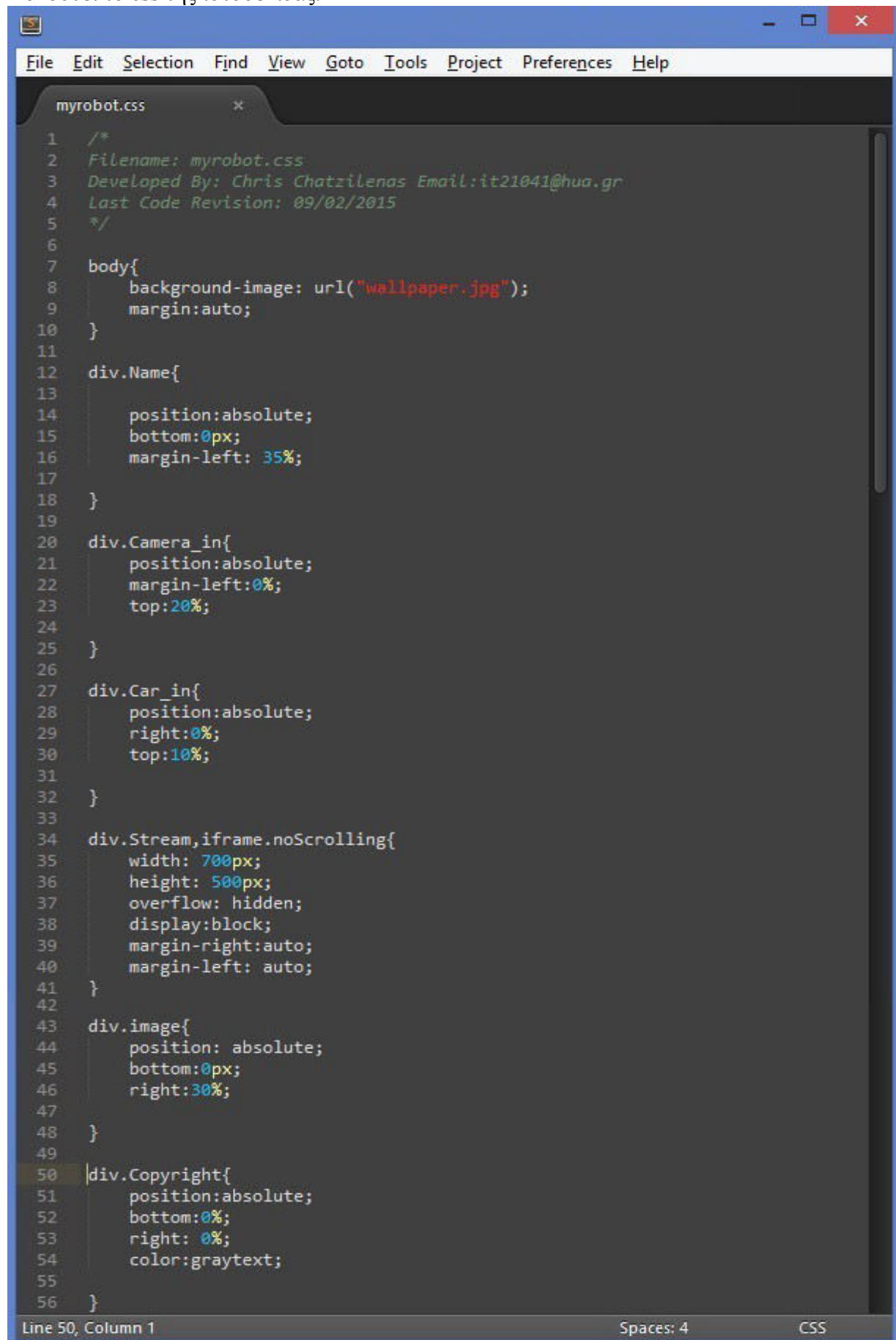
Line 2, Column 5 Spaces: 2 HTML

```
File Edit Selection Find View Goto Tools Project Preferences Help

robot.html x
44 | else if(event.keyCode==68){
45 |     $('#content').load('/arduino/servoRight');
46 | }
47 | else if(event.keyCode==87){
48 |     $('#content').load('/arduino/servoCenter');
49 | }
50 | else if(event.keyCode==80){
51 |     $('#content').load('/arduino/photo');
52 | }
53 | }
54 </script>
55 </head>
56 <body onkeydown="CheckKeyCode1()">
57 | <span id="content"></span>
58 | <div class="image">
59 | | 
60 | | </div>
61 | <div class="Camera_in">
62 | | 
63 | | </div>
64 | <div class="Car_in">
65 | | 
66 | | </div>
67 | <div class="Name">
68 | | 
69 | | </div>
70 | <div class="Stream">
71 | | <!--In this div the user can watch the environment around the vehicle.
72 | | -->
73 | | <iframe src="http://192.168.1.104:8080/stream_simple.html" class="
74 | | noScrolling">
75 | | <p>Your browser does not support iframes.</p>
76 | | </iframe>
77 | | </div>
78 | | <div class="Copyright">
79 | | | Desinged by Vicky Markantoni.<br>
80 | | | Copyright &copy; 2014 Chris Hatzilenas.<br>
81 | | | All Rights Reserved.
82 | </div>
83 </body>
84 </html>

Line 44, Column 11 Spaces: 2 HTML
```

Ακολουθεί το css της ιστοσελίδας:



```
1  /*
2  Filename: myrobot.css
3  Developed By: Chris Chatzilenas Email:it21041@hua.gr
4  Last Code Revision: 09/02/2015
5  */
6
7  body{
8      background-image: url("wallpaper.jpg");
9      margin:auto;
10 }
11
12 div.Name{
13
14     position:absolute;
15     bottom:0px;
16     margin-left: 35%;
17 }
18
19
20 div.Camera_in{
21     position:absolute;
22     margin-left:0%;
23     top:20%;
24 }
25
26
27 div.Car_in{
28     position:absolute;
29     right:0%;
30     top:10%;
31 }
32
33
34 div.Stream,iframe.noScrolling{
35     width: 700px;
36     height: 500px;
37     overflow: hidden;
38     display:block;
39     margin-right:auto;
40     margin-left: auto;
41 }
42
43 div.image{
44     position: absolute;
45     bottom:0px;
46     right:30%;
47 }
48
49
50 div.Copyright{
51     position:absolute;
52     bottom:0%;
53     right: 0%;
54     color:graytext;
55 }
56 }
```

Line 50, Column 1 Spaces: 4 CSS

Η μορφή της ιστοσελίδας σε αυτό το στάδιο ήταν:



Μέρος Τέταρτο – Προσθήκη αισθητήρα για τη διευκόλυνση του χρήστη



6.1 Εισαγωγή

Σε αυτό το τέταρτο μέρος πρόσθεσα στο αμαξίδιο μία φωτοαντίσταση. Η φωτοαντίσταση (ή φωτοκύτταρο) αλλάζει την αντίστασή της με βάση την ποσότητα του φωτός που πέφτει πάνω της. Στο προηγούμενο μέρος, χρησιμοποίησα μια ψηφιακή κάμερα με την οποία θα μπορεί εύκολα ο χρήστης να χειρίζεται το αμαξίδιο ακόμα και αν δεν έχει οπτική επαφή με αυτό.

Τι γίνεται όμως όταν το αμαξίδιο οδηγείται σε ένα χώρο ο οποίος δεν έχει αρκετό φως; Για αυτόν ακριβώς το λόγο χρησιμοποίησα το φωτοκύτταρο, για να μπορέσω να διαχειριστώ τις τιμές του αισθητήρα αυτού και μετά να ελέγξω τη λειτουργία κάποιων leds (light-emitting diode) τα οποία θα βοηθούν το χρήστη να κινεί το αμαξίδιο σε περιοχές χωρίς πολύ φωτισμό. Για να μην γίνεται εύκολα αντιληπτό το αμαξίδιο από άλλους χρησιμοποίησα υπέρυθρα leds (ir-leds), καθώς η υπέρυθη ακτινοβολία που εκπέμπουν δεν ανήκει στο φάσμα του ορατού φωτός.

Για να καταφέρω να εκμεταλλευτώ πλήρως τα ir-leds έπρεπε να αφαιρέσω το φίλτρο υπέρυθρης ακτινοβολία που είχε η κάμερα και μετά να προσθέσω στη θέση του ένα κομμάτι από αχρησιμοποίητο φίλμ αναλογικής φωτογραφικής μηχανής, αφού το εκθέσω πρώτα για κάποια λεπτά σε δυνατό φως. Η προσθήκη αυτού του αυτοσχέδιου φίλτρου έγινε, διότι χωρίς αυτό αν η ψηφιακή κάμερα εκτιθόταν σε δυνατό φως τότε ο χρήστης δεν θα έβλεπε στην οθόνη του streaming τίποτα παραμόνο λευκό φως.

Εκτός από τα παραπάνω, έπρεπε να προσθέσω στην κάμερα σε κάποιο σημείο τα ir-leds. Υπενθυμίζω ότι η κάμερα έχει σφαιρικό σχήμα, οπότε χρειαζόμουν μια πλακέτα σε σχήμα δακτυλίου. Μετά από έρευνα αγοράς δεν κατάφερα να βρω κάποιο σχετικό προϊόν, οπότε αποφάσισα να φτιάξω μία αυτοσχέδια από χοντρό χαρτόνι (περίπου 3mm). Μετά από αυτό, κόλλησα με κολλητήρι και καλά τα ir-leds καθώς και τις αντιστάσεις που χρειαζόμουν πάνω στην «πλακέτα» παράλληλα έτσι ώστε να περνά από όλα τα ir-leds ίσο ρεύμα και να μην υπάρχουν διαφορές στη φωτεινότητα. Στη συνέχεια κόλλησα την «πλακέτα» με θερμόκολλα στη ψηφιακή κάμερα.

Τέλος, χρησιμοποιώντας μια πλακέτα κατασκευών, τη φωτοαντίσταση, μία αντίσταση, καλώδια, κολλητήρι και καλάνι κατάφερα να εφαρμόσω την πλακέτα στο πάνω μέρος του αμαξιτίου για να έχω σωστότερα δεδομένα.

6.2 Υλικά

1. Φωτοαντίσταση (ή φωτοκύτταρο) LDR 5mm.



2. 10 υπέρυθρα leds 5mm-950nm.



3. 10 αντιστάσεις 330Ω και 1 αντίσταση $10k\Omega$.

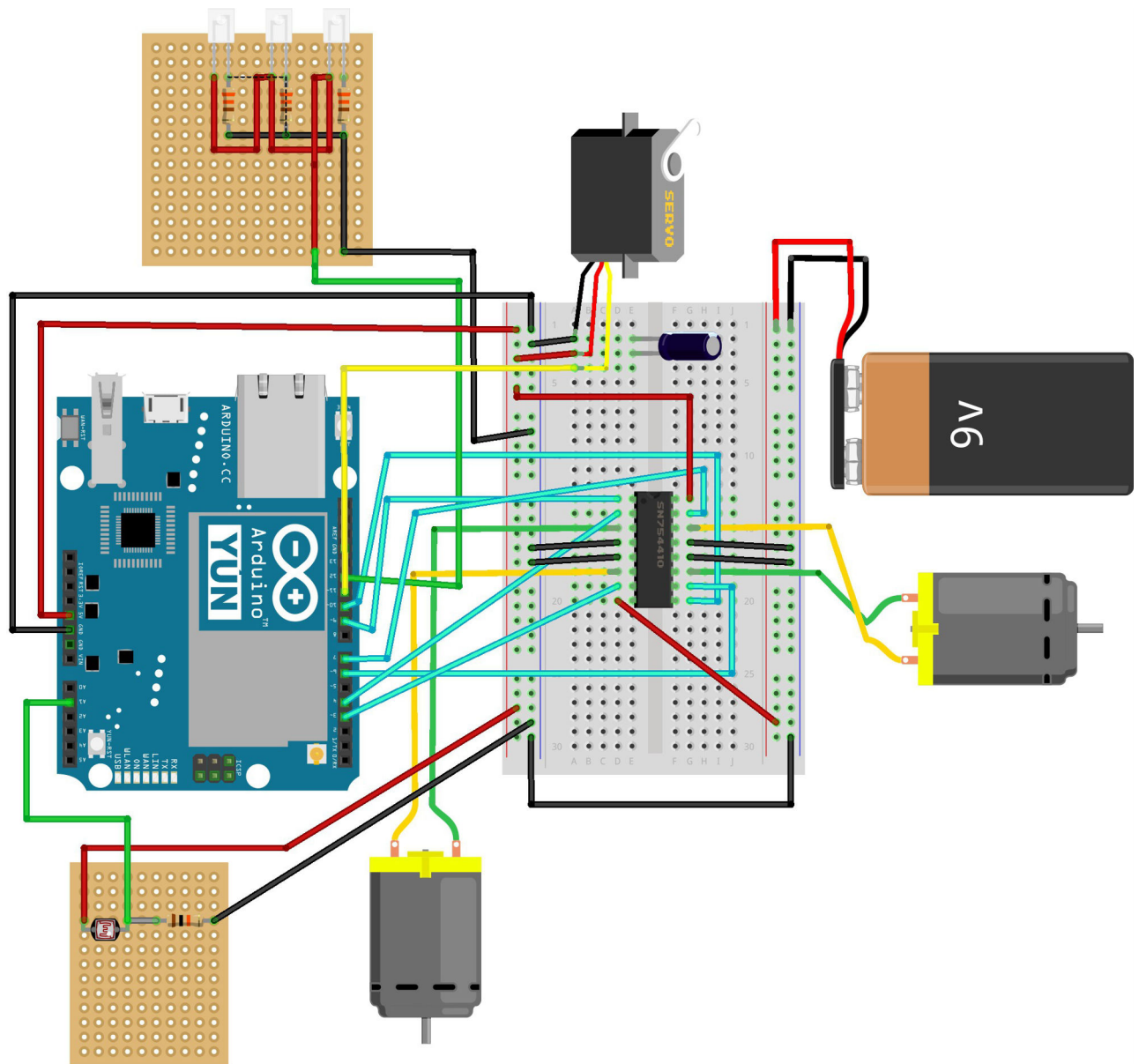


5. Καλώδια, κολλητήρι, καλάι, θερμόκολλα και πλακέτα κατασκευής.

6. Χαρτόνι μεγέθους $200 \times 200 \times 3$ mm.

6.3 Συνδεσμολογία

Η τελική συνδεσμολογία είναι η εξής:

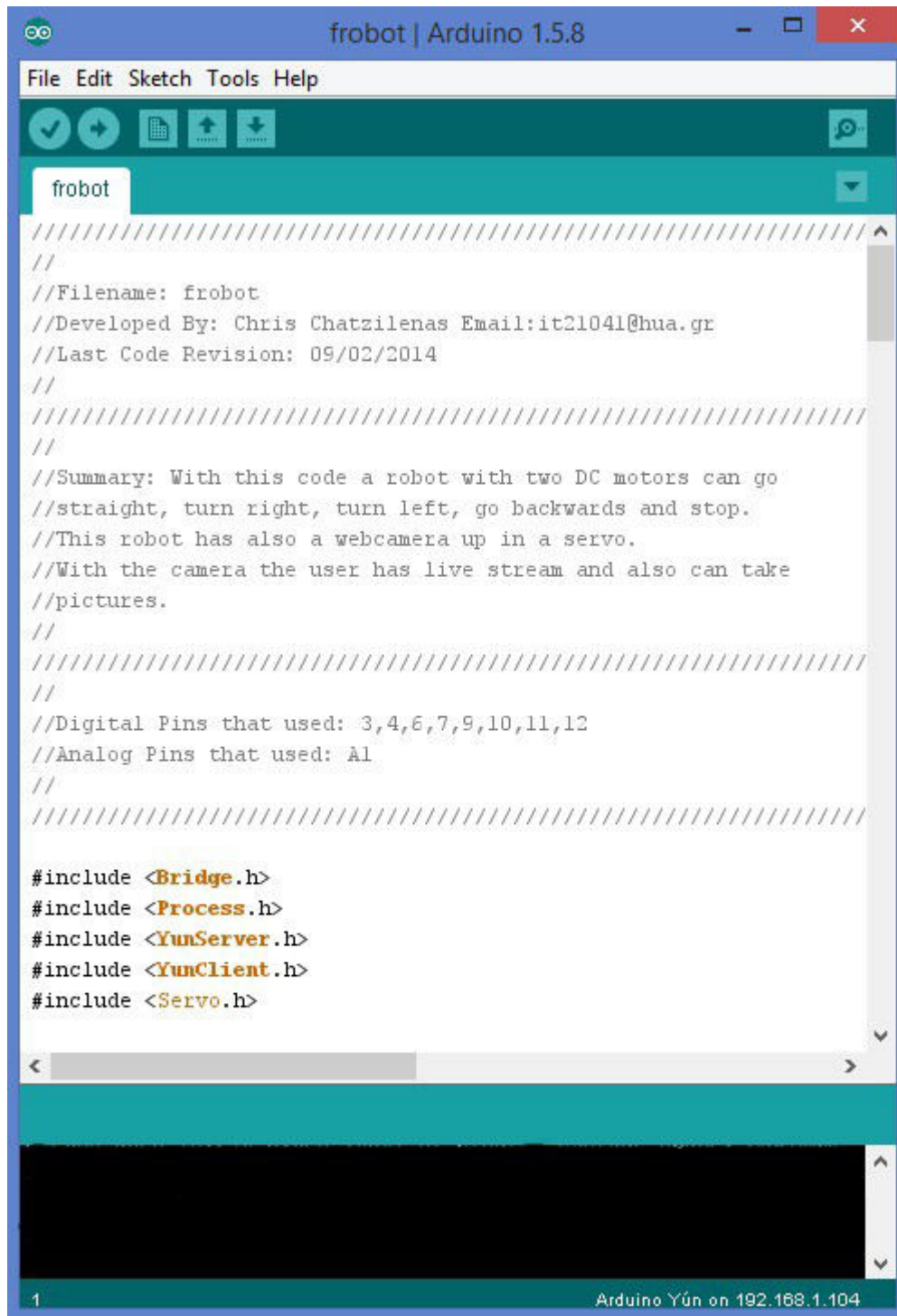


fritzing

Η συνδεσμολογία που περιγράφεται στην παραπάνω εικόνα δεν αντιστοιχεί απόλυτα στην πραγματική «πλακέτα» που σχεδίασα μόνος μου, γιατί το σχήμα της θα γινόταν αρκετά περίπλοκο. Παρόλα αυτά, σχεδίασα μια πλακέτα με 3 ir-leds (από 10) και 3 αντιστάσεις (από 10) σε παράλληλη σύνδεση. Οπότε, όπως φαίνεται και στην εικόνα, συνέδεσα στη γείωση της «πλακέτας», τη γείωση του Arduino, ενώ το ρεύμα της το συνέδεσα με το pin 12 του Arduino. Τέλος, στην πλακέτα με τον αισθητήρα συνέδεσα τη μία άκρη της φωτοαντίστασης με το ρεύμα του Arduino, ενώ την άλλη άκρη τη συνέδεσα με το αναλογικό pin A1 του Arduino, αλλά και με την αντίσταση 10 kΩ την οποία με τη σειρά της την συνέδεσα με τη γείωση του Arduino.

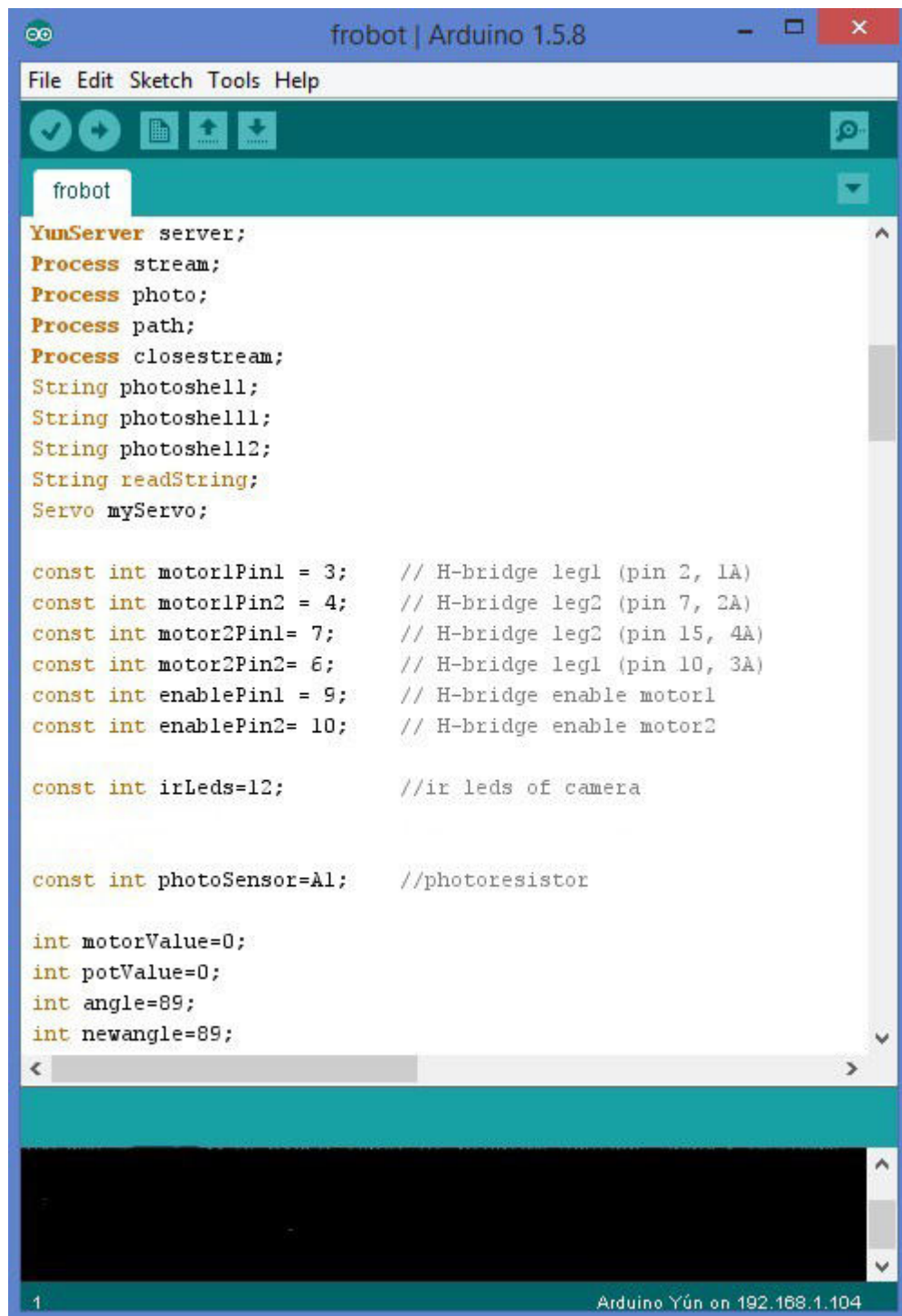
6.4 Κώδικας

Ο τελικός κώδικας για το Server είναι ο εξής:



```
Arduino IDE - frobot | Arduino 1.5.8
File Edit Sketch Tools Help

//
//
//Filename: frobot
//Developed By: Chris Chatzilenas Email:it21041@hua.gr
//Last Code Revision: 09/02/2014
//
//
//Summary: With this code a robot with two DC motors can go
//straight, turn right, turn left, go backwards and stop.
//This robot has also a webcamera up in a servo.
//With the camera the user has live stream and also can take
//pictures.
//
//
//Digital Pins that used: 3,4,6,7,9,10,11,12
//Analog Pins that used: A1
//
//
#include <Bridge.h>
#include <Process.h>
#include <YunServer.h>
#include <YunClient.h>
#include <Servo.h>
```



The screenshot shows the Arduino IDE interface with a sketch named "frobot" open. The sketch contains the following code:

```
YunServer server;
Process stream;
Process photo;
Process path;
Process closestream;
String photoshell;
String photoshell1;
String photoshell2;
String readString;
Servo myServo;

const int motor1Pin1 = 3;    // H-bridge leg1 (pin 2, 1A)
const int motor1Pin2 = 4;    // H-bridge leg2 (pin 7, 2A)
const int motor2Pin1 = 7;    // H-bridge leg2 (pin 15, 4A)
const int motor2Pin2 = 6;    // H-bridge leg1 (pin 10, 3A)
const int enablePin1 = 9;    // H-bridge enable motor1
const int enablePin2 = 10;   // H-bridge enable motor2

const int irLeds=12;         //ir leds of camera

const int photoSensor=A1;    //photoresistor

int motorValue=0;
int potValue=0;
int angle=89;
int newangle=89;
```

The IDE window title is "frobot | Arduino 1.5.8". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for saving, running, uploading, and downloading. The status bar at the bottom indicates "1" and "Arduino Yun on 192.168.1.104".



```
void setup() {  
  
  Serial.begin(9600);  
  myServo.attach(11);  
  
  // set all the other pins you're using as outputs:  
  pinMode(motor1Pin1, OUTPUT);  
  pinMode(motor1Pin2, OUTPUT);  
  pinMode(motor2Pin1, OUTPUT);  
  pinMode(motor2Pin2, OUTPUT);  
  pinMode(enablePin1, OUTPUT);  
  pinMode(enablePin2, OUTPUT);  
  
  //Ir Leds  
  pinMode(irLeds, OUTPUT);  
  digitalWrite(irLeds, LOW);  
  
  Bridge.begin();  
  
  //open live stream  
  stream.runShellCommand("mjpg_streamer -i 'input_uvc.so -d  
/dev/video0 -r 640x480 -f 25' -o 'output_http.so -p 8080 -w /www/webcam' &");  
  // set enablePin high so that motor can turn on:  
  digitalWrite(enablePin1, HIGH);  
  digitalWrite(enablePin2, HIGH);  
  
  //put servo to 89  
  myServo.write(angle);  
  // Listen for incoming connection only from localhost  
  // (no one from the external network could connect)  
  server.listenOnLocalhost();  
  server.begin();  
  
}
```

1

Arduino Yún on 192.168.1.104

The image shows a screenshot of the Arduino IDE interface. The title bar at the top reads "frobot | Arduino 1.5.8". Below the title bar is a menu bar with "File", "Edit", "Sketch", "Tools", and "Help". Under the "Sketch" menu, there are icons for a checkmark, a circular arrow, a document, and upload/download arrows. A tab labeled "frobot" is active. The main text area contains the following C++ code:

```
void loop() {  
  
  int sensorValph=analogRead(photoSensor);  
  delay(5);  
  int sV=sensorValph/4;    //convert from 0-1023 to 0-255  
  
  if(sV<127)  
  {  
    digitalWrite(irLeds,HIGH);  //enable ir leds  
  }  
  else  
  {  
    digitalWrite(irLeds,LOW);   //disable ir leds  
  }  
  
  // Get clients coming from server  
  YumClient client = server.accept();  
  
  // There is a new client?  
  if (client) {  
    // read the command  
    String command = client.readString();  
    command.trim();           //kill whitespace  
    Serial.println(command);  
    if(command=="forward"){  
      //drive straight:  
      //set leg 2 of the 2 motors HIGH to go straight  
      digitalWrite(motor1Pin1,LOW);  
      digitalWrite(motor1Pin2, HIGH);  
  
      digitalWrite(motor2Pin1,HIGH);  
      digitalWrite(motor2Pin2, LOW);  
    }  
  }  
}
```

At the bottom of the IDE, there is a status bar showing "1" on the left and "Arduino Yún on 192.168.1.104" on the right.

```
Arduino IDE - frobot | Arduino 1.5.8
File Edit Sketch Tools Help

frobot

else if(command=="stop"){
    //stop
    //set leg 1,2 of the 2 motors HIGH or LOW to stop
    digitalWrite(motor1Pin1,HIGH);
    digitalWrite(motor1Pin2, HIGH);

    digitalWrite(motor2Pin1,HIGH);
    digitalWrite(motor2Pin2, HIGH);
}
else if(command=="right"){
    //turn right:
    //set leg 2 of the 2 motors HIGH
    //set leg 1 of motor1 LOW
    //set leg 1 of motor2 Higher than LOW
    digitalWrite(motor1Pin1,LOW);
    digitalWrite(motor1Pin2, HIGH);

    potValue=350;
    motorValue=map(potValue,0,1023,0,255);

    digitalWrite(motor2Pin1,HIGH);
    analogWrite(motor2Pin2, motorValue);
}
else if(command=="left"){
    //turn left:
    //set leg 2 of the 2 motors HIGH
    //set leg 1 of motor2 LOW
    //set leg 1 of motor1 Higher than LOW

    potValue=350;
    motorValue=map(potValue,0,1023,0,255);
    analogWrite(motor1Pin1,motorValue);
    digitalWrite(motor1Pin2, HIGH);

    digitalWrite(motor2Pin1,HIGH);
    digitalWrite(motor2Pin2, LOW);
}
}

159 Arduino Yún on 192.168.1.104
```

```

frobot | Arduino 1.5.8
File Edit Sketch Tools Help

frobot

else if(command=="reverse"){
    //reverse
    //set leg 1 of the 2 motors HIGH to go backwards
    digitalWrite(motor1Pin2,LOW);
    digitalWrite(motor1Pin1, HIGH);

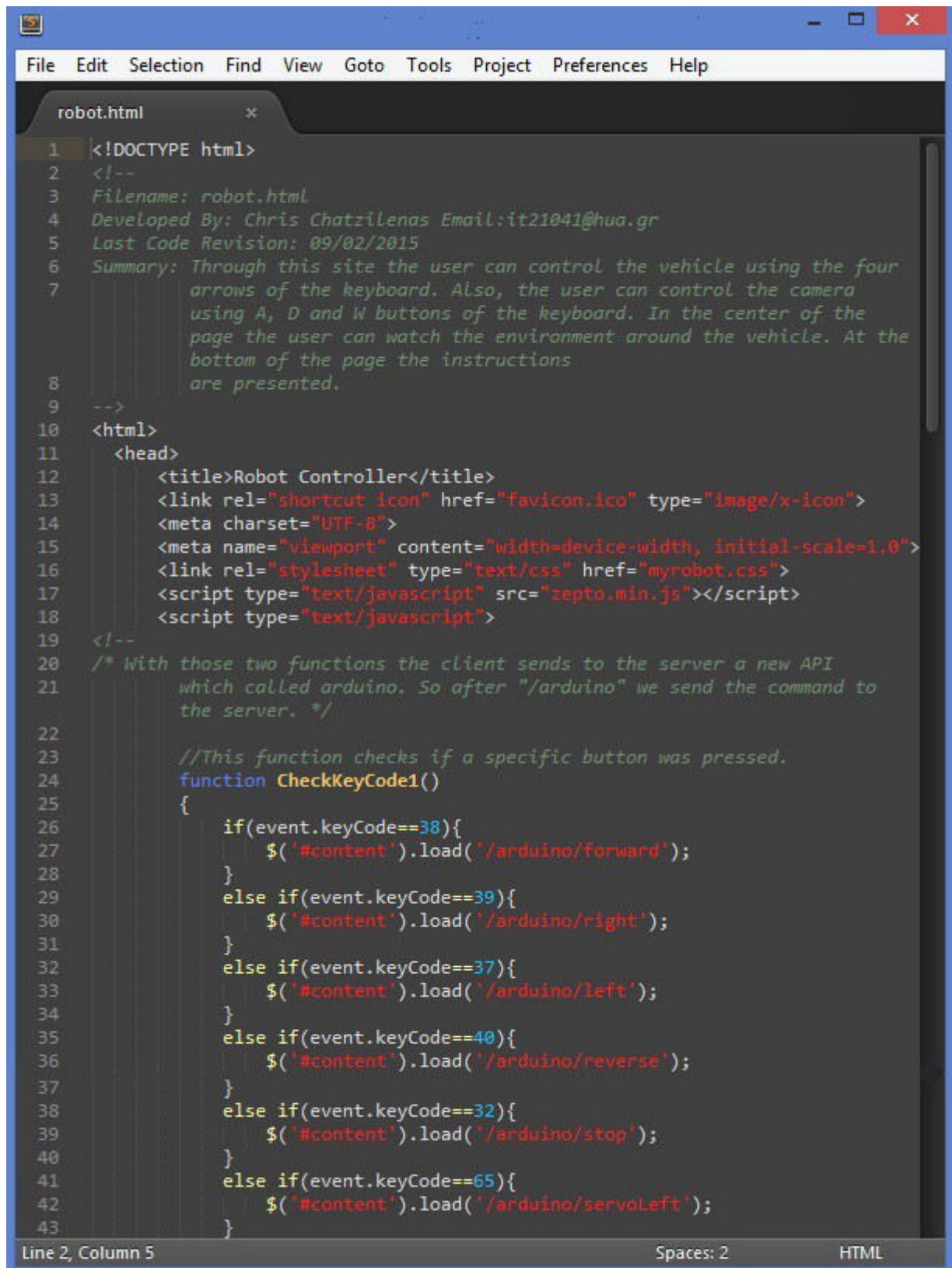
    digitalWrite(motor2Pin2,HIGH);
    digitalWrite(motor2Pin1, LOW);
}
else if(command=="servoLeft"){
    newangle=newangle-20;
    if(newangle+20<20){
        myServo.write(0);
    }
    myServo.write(newangle);
}
else if(command=="servoRight"){
    newangle=newangle+20;
    myServo.write(newangle);
}
else if(command=="servoCenter"){
    newangle=89.5;
    myServo.write(newangle);
}
else if(command=="photo"){
    //close stream to take photo
    closestream.runShellCommand("kill -9 `pidof mjpg_streamer`");

    while(closestream.running());
    //take photo
    photoshell1="fswebcam /mnt/sdal/photo";
    photoshell2=".png";
    photoshell=photoshell1+countp+photoshell2;
    photo.runShellCommand(photoshell);
    //waits till the picture is saved
    while(photo.running());
    //restart stream
    stream.runShellCommand("mjpg_streamer -i 'input_uvc.so -d
/dev/video0 -r 640x480 -f 25' -o 'output_http.so -p 8080 -w /www/webcam' &");
    countp++;
}
// Close connection and free resources.
client.stop();
}
delay(50); // Poll every 50ms
}
< >

186 Arduino Yún on 192.168.1.104

```

Ακολουθεί ο τελικός κώδικας της ιστοσελίδας (Client):



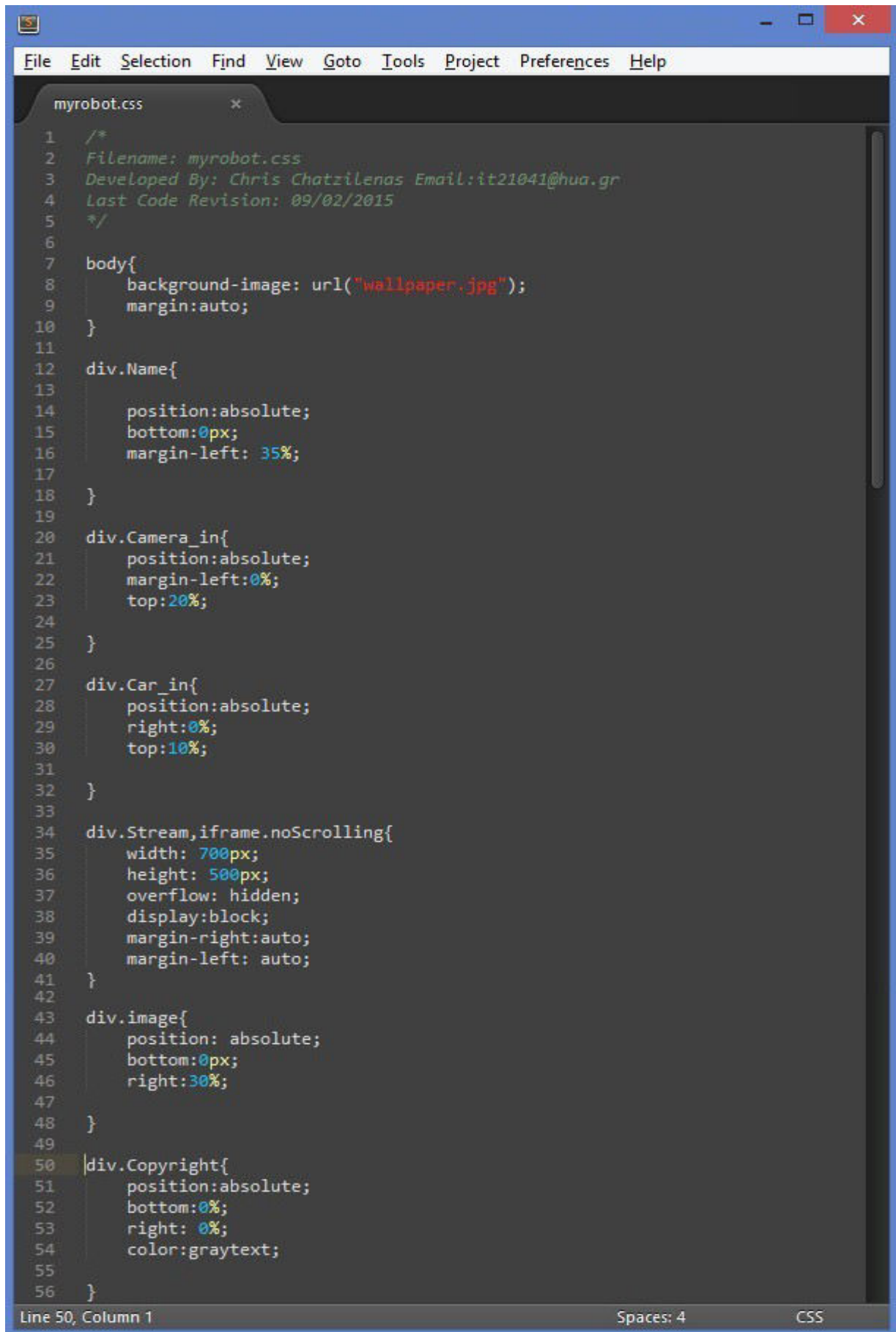
```
1 <!DOCTYPE html>
2 <!--
3  Filename: robot.html
4  Developed By: Chris Chatzilenas Email:it21041@hua.gr
5  Last Code Revision: 09/02/2015
6  Summary: Through this site the user can control the vehicle using the four
7           arrows of the keyboard. Also, the user can control the camera
8           using A, D and W buttons of the keyboard. In the center of the
9           page the user can watch the environment around the vehicle. At the
10          bottom of the page the instructions
11          are presented.
12 -->
13 <html>
14   <head>
15     <title>Robot Controller</title>
16     <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
17     <meta charset="UTF-8">
18     <meta name="viewport" content="width=device-width, initial-scale=1.0">
19     <link rel="stylesheet" type="text/css" href="myrobot.css">
20     <script type="text/javascript" src="zepto.min.js"></script>
21     <script type="text/javascript">
22       /* With those two functions the client sends to the server a new API
23        which called arduino. So after "/arduino" we send the command to
24        the server. */
25
26       //This function checks if a specific button was pressed.
27       function CheckKeyCode1()
28       {
29         if(event.keyCode==38){
30           $('#content').load('/arduino/forward');
31         }
32         else if(event.keyCode==39){
33           $('#content').load('/arduino/right');
34         }
35         else if(event.keyCode==37){
36           $('#content').load('/arduino/left');
37         }
38         else if(event.keyCode==40){
39           $('#content').load('/arduino/reverse');
40         }
41         else if(event.keyCode==32){
42           $('#content').load('/arduino/stop');
43         }
44         else if(event.keyCode==65){
45           $('#content').load('/arduino/servoLeft');
46         }
47       }
48     </script>
49   </head>
50   <body>
51     <div id="content">
52     </div>
53   </body>
54 </html>
```

```
robot.html x
File Edit Selection Find View Goto Tools Project Preferences Help

44 | else if(event.keyCode==68){
45 |     $('#content').load('/arduino/servoRight');
46 | }
47 | else if(event.keyCode==87){
48 |     $('#content').load('/arduino/servoCenter');
49 | }
50 | else if(event.keyCode==80){
51 |     $('#content').load('/arduino/photo');
52 | }
53 | }
54 </script>
55 </head>
56 <body onkeydown="CheckKeyCode1()">
57   <span id="content"></span>
58   <div class="image">
59     
60   </div>
61   <div class="Camera_in">
62     
63   </div>
64   <div class="Car_in">
65     
66   </div>
67   <div class="Name">
68     
69   </div>
70   <div class="Stream">
71     <!--In this div the user can watch the environment around the vehicle.
72     -->
73     <iframe src="http://192.168.1.104:8080/stream_simple.html" class="
74       noScrolling">
75       <p>Your browser does not support iframes.</p>
76     </iframe>
77   </div>
78   <div class="Copyright">
79     Desinged by Vicky Markantoni.<br>
80     Copyright &copy; 2014 Chris Hatzilenas.<br>
81     All Rights Reserved.
82   </div>
83 </body>
84 </html>

Line 44, Column 11 Spaces: 2 HTML
```

Το τελικό css της ιστοσελίδας:

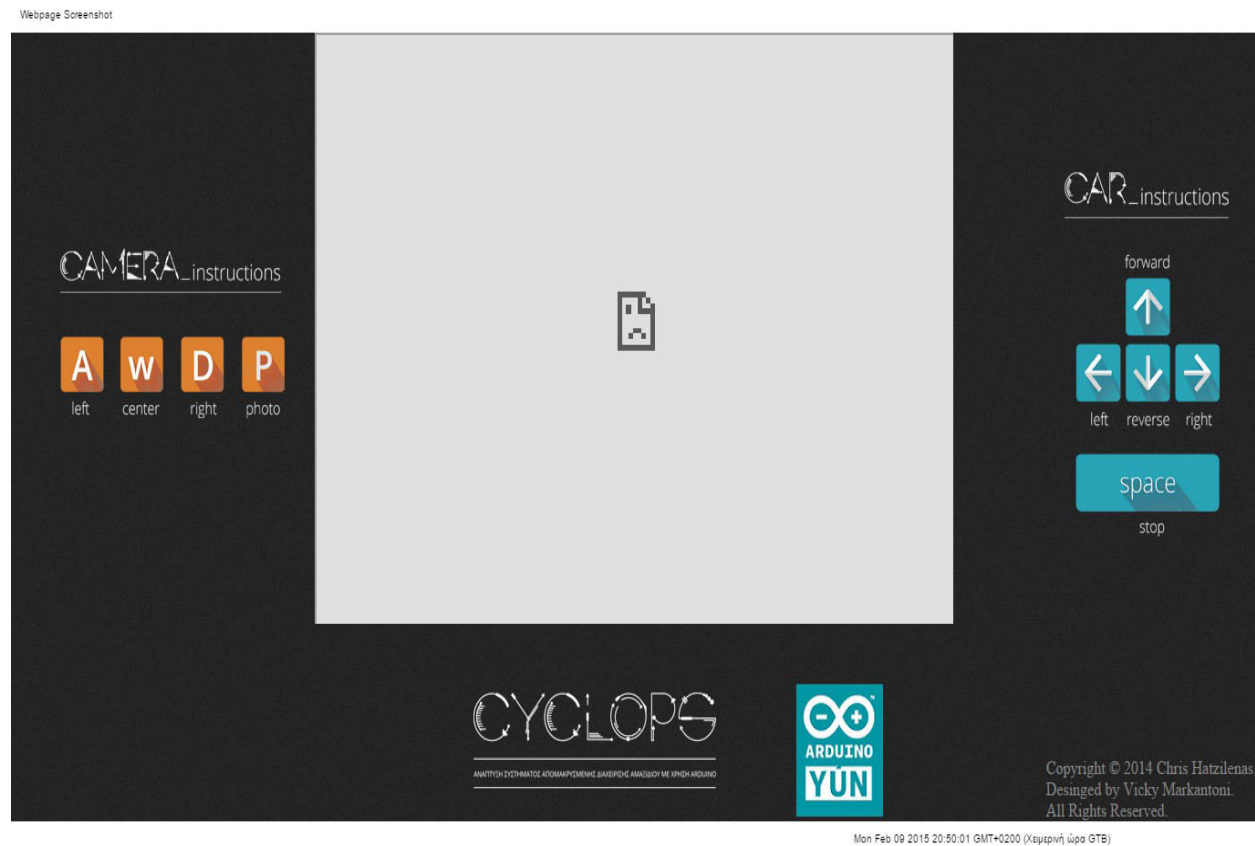


The screenshot shows a code editor window titled 'myrobot.css'. The editor contains the following CSS code:

```
1  /*
2  Filename: myrobot.css
3  Developed By: Chris Chatzilenas Email:it21041@hua.gr
4  Last Code Revision: 09/02/2015
5  */
6
7  body{
8      background-image: url("wallpaper.jpg");
9      margin:auto;
10 }
11
12 div.Name{
13
14     position:absolute;
15     bottom:0px;
16     margin-left: 35%;
17 }
18
19
20 div.Camera_in{
21     position:absolute;
22     margin-left:0%;
23     top:20%;
24 }
25
26
27 div.Car_in{
28     position:absolute;
29     right:0%;
30     top:10%;
31 }
32
33
34 div.Stream,iframe.noScrolling{
35     width: 700px;
36     height: 500px;
37     overflow: hidden;
38     display:block;
39     margin-right:auto;
40     margin-left: auto;
41 }
42
43 div.image{
44     position: absolute;
45     bottom:0px;
46     right:30%;
47 }
48
49
50 div.Copyright{
51     position:absolute;
52     bottom:0%;
53     right: 0%;
54     color:graytext;
55 }
56 }
```

The status bar at the bottom indicates 'Line 50, Column 1', 'Spaces: 4', and 'CSS'.

Το τελικό αποτέλεσμα της ιστοσελίδας ήταν το εξής:



Μέρος Πέμπτο – Προσθήκη ξύλινου σκελετού στο αμαξίδιο

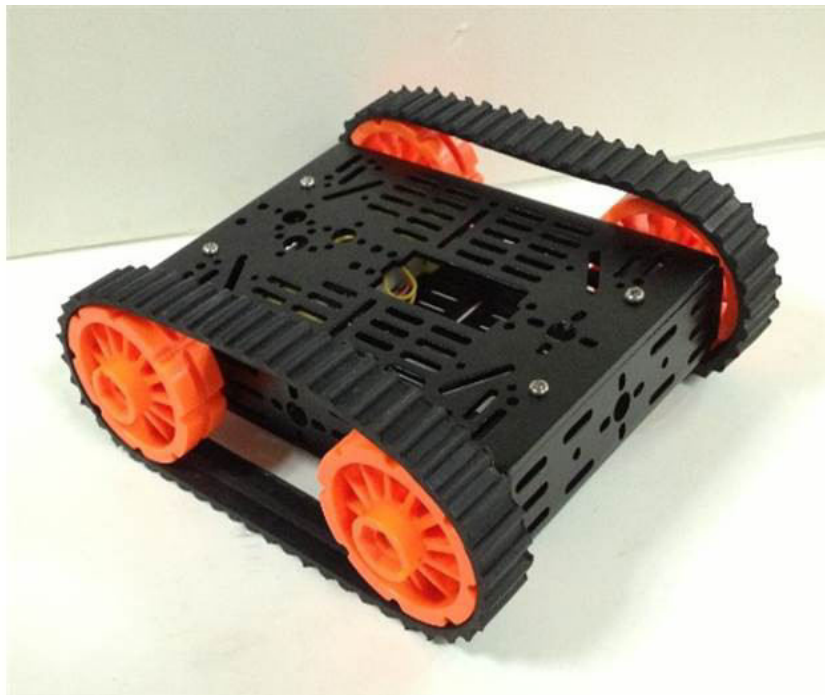


7.1 Εισαγωγή

Αυτό το τελευταίο μέρος είναι περισσότερο συμπληρωματικό. Παρακάτω λοιπόν θα σας δείξω πως κατασκεύασα το ξύλινο σκελετό που πρόσθεσα στο αμαξίδιο.

7.2 Υλικά

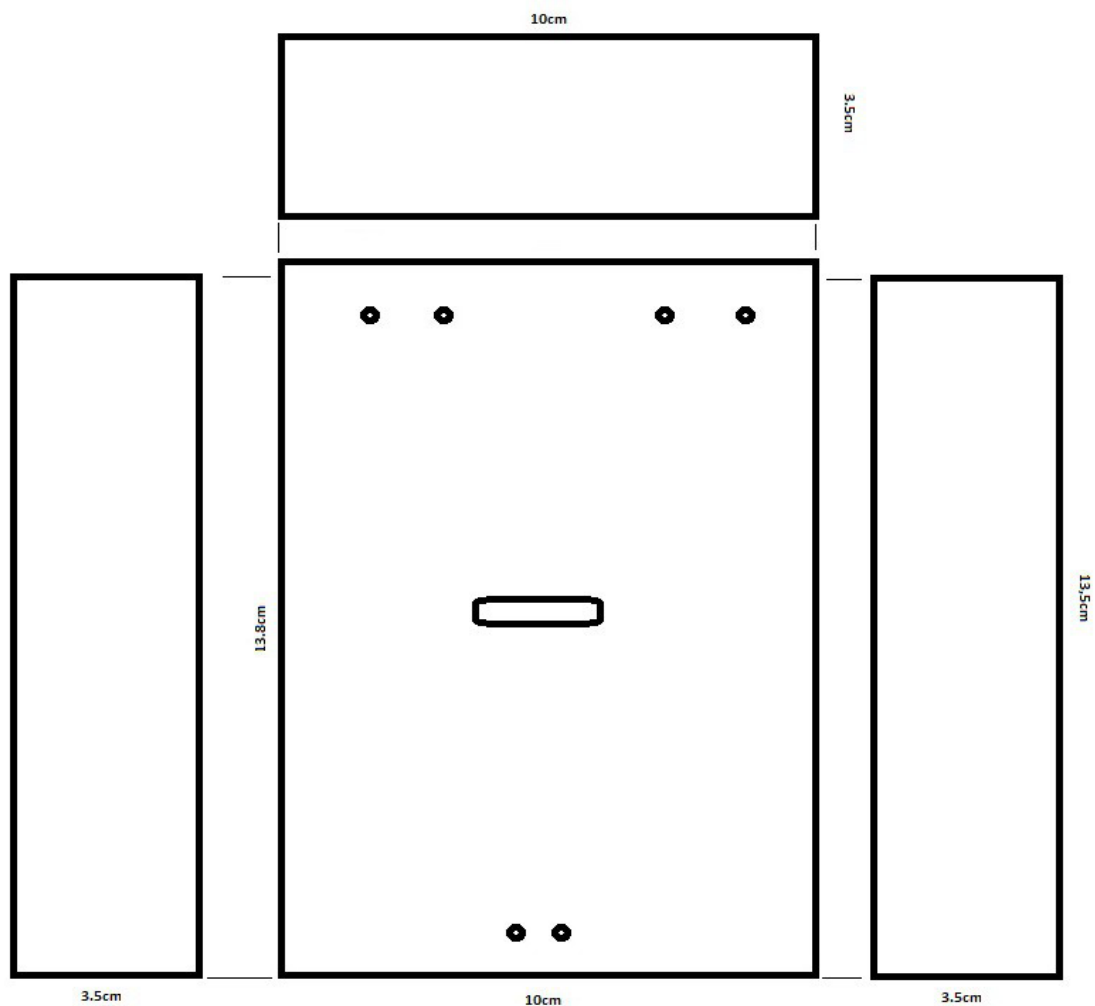
1. Το έτοιμο αμαξίδιο.



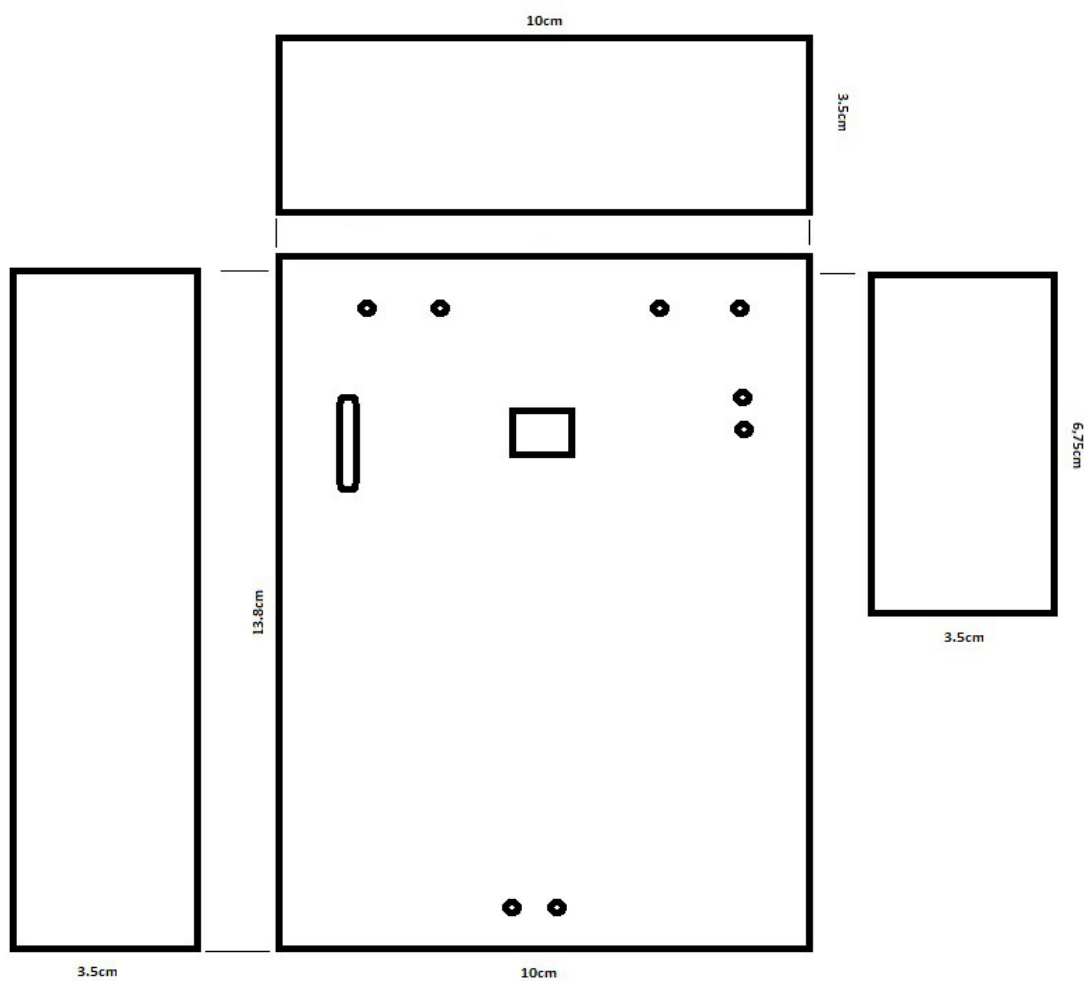
2. Κόντρα πλακέ διαστάσεων 500x500x3 mm.
3. Βίδες M3 8mm, μεταλικούς αποστάτες M/F 30mm, μεταλικούς αποστάτες M/F 5mm, παξιμάδια M3

7.3 Κατασκευή

Για να γίνει εύκολα αντιληπτή η κατασκευή θα ακολουθήσουν εικόνες. Στο πρώτο επίπεδο χρησιμοποίησα τον προκατασκευασμένο σκελετό του αμαξιδίου, αλλά χωρίς το πάνω μέρος. Σε αυτό το επίπεδο είναι, όπως είναι φυσικό, τα DC motors για να κινηθεί το αμαξίδιο και μια μπαταρία 9V για τους κινητήρες (DC motors). Αυτό το έκανα για να μπορέσω να προσαρτήσω το δεύτερο επίπεδο το οποίο και κατασκεύασα με κόντρα πλακέ. Οπότε, το δεύτερο επίπεδο είναι το εξής:



Σε αυτό το δεύτερο επίπεδο τοποθέτησα τη breadboard και το Arduino Yun. Το τρίτο επίπεδο το κατασκεύασα ως εξής:



Σε αυτό το επίπεδο έχω προσθέσει μια επαναφορτιζόμενη μπαταρία για το Arduino Yun. Τέλος, στο τέταρτο και τελευταίο επίπεδο, το οποίο αποτελείται από το πάνω μέρος του προκατασκευασμένου σκελετού του αμαξιδίου, πρόσθεσα την κάμερα και την πλακέτα με τον αισθητήρα.

Βιβλιογραφία

1. Banzi Massimo, Getting Started with Arduino, Make:Books, 2009.
2. Deitel Paul and Deitel Harvey, Java™ How to Program Eighth Edition, Prentice Hall; 8 edition, 2009.
3. Melgar Enrique Ramos and Diez Ciriaco Castro with Jaworski Przemek, Arduino and Kinect Projects, Technology In Action, 2012.
4. O' Sallivan Dan and Igoe Tom, Physical Computing, Thomson Course Technology PTR, 2004.
5. Weerawarana Sanjiva, Curbera Fransisco, Leymann Frank, Storey Tony, Ferguson F. Donald, Web Services Platform Architecture, Prentice Hall; 1 edition, 2005.
6. <http://predic8.com/rest-webservices.htm>
7. <http://zeptojs.com/>