

Τίτλος έργου:

UGV απομακρυσμένου ελέγχου μέσω Διαδικτύου

Διεύθυνση αποθετηρίου στο Github: <https://github.com/gianakop/Web-UGV->

Εκπαιδευτικός: Νίκος Γιαννακόπουλος ΠΕ86 Πληροφορικής
gianakop@gmail.com



LETSROBOT.TV [LIVE devices](#) | [Shop](#) | [About](#) | [Community](#) | [Resources](#)

left right forward backward



Γενική περιγραφή ιδέας:

Το Ρομπότ απομακρυσμένου ελέγχου μέσω Διαδικτύου, είναι ένα μη επανδρωμένο όχημα εδάφους (UGV), το οποίο ο χειριστής το ελέγχει μέσω διαδικτύου από οπουδήποτε στον κόσμο, μέσα από μία ιστοσελίδα που του παρέχει πρόσβαση στον έλεγχο των λειτουργιών του ρομπότ. Η ίδια ιστοσελίδα είναι το περιβάλλον που έρχεται η

UGV απομακρυσμένου ελέγχου μέσω Διαδικτύου ~ 2 ~ Νίκος Γιαννακόπουλος
ανάδραση του ρομπότ, δηλαδή μέσα από αυτή ο χρήστης βλέπει και ακούει ότι βλέπει και ακούει το ρομπότ και φυσικά το κατευθύνει κατάλληλα σύμφωνα με τις επιθυμίες του.

Από τεχνικής άποψης το ρομπότ αποτελείται από ένα σασί με τους απαραίτητους κινητήρες για την κίνηση του, από έναν ελεγκτή των κινητήρων που θέτει τους κινητήρες κατάλληλη λειτουργία, από μία κάμερα με μικρόφωνο η οποία επιτρέπει στον χειριστή να βλέπει και να ακούει ότι βλέπει και ακούει το ρομπότ, από ένα ηχείο που επιτρέπει στον χειριστή να στέλνει τη φωνή του στο χώρο που κινείται το ρομπότ και τέλος, από έναν μικροελεγκτή που είναι ο βασικός εγκέφαλος του ρομπότ και ο οποίος συνδέεται και ελέγχει τη συντονισμένη λειτουργία όλων των παραπάνω μερών. Ο μικροελεγκτής αυτός, παρέχει την δυνατότητα ασύρματης σύνδεσης του ρομπότ με το διαδίκτυο μέσω Wi-Fi, ώστε να είναι δυνατός ο έλεγχος του ρομπότ μέσω του διαδικτύου. Τέλος ο χρήστης πρέπει να διαθέτει ηλεκτρονικό υπολογιστή με πρόσβαση στο διαδίκτυο, καθώς και την ηλεκτρονική διεύθυνση της ιστοσελίδας ελέγχου του ρομπότ η οποία είναι <https://letsrobot.tv/robocaster/gianakop/robot/59206902>. Η σελίδα αυτή, φιλοξενείται δωρεάν στο **LETSROBOTTV** <https://letsrobot.tv/> όπου φιλοξένει και πλήθος άλλα on-line ρομπότ και λειτουργεί μόνο όταν το ρομπότ είναι ανοικτό.)

Αναλυτική περιγραφή ιδέας:

Το Ρομπότ αποτελείται λοιπόν από τα παρακάτω υλικά μέρη των οποίων η χρήση και λειτουργία φαίνεται στο παρακάτω πίνακα :

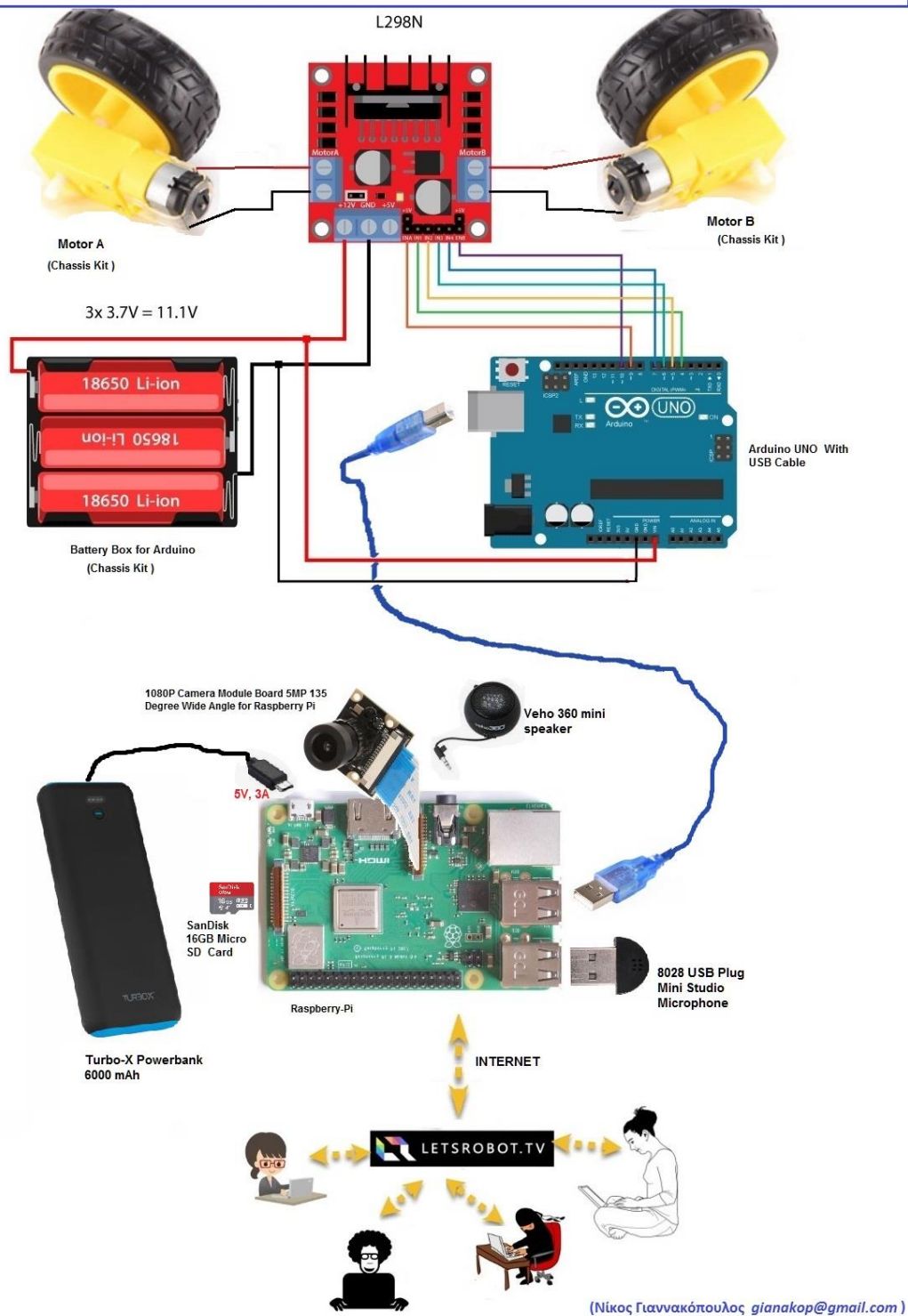
A.A.	ΠΡΟΪΟΝ (μέρη του ρομπότ)	ΛΕΙΤΟΥΡΓΙΑ & ΧΡΗΣΗ
1	Raspberry Pi 3 Model B+ (B plus) Quad Core 1.4GHz 64 bit CPU wifi & bluetooth	Η πλακέτα που κάνει τη διασύνδεση με το Internet (μέσω wifi) και στέλνει εικόνα και ήχο από το ρομπότ στον χρήστη του διαδικτύου και επιστρέφει τις εντολές του απομακρυσμένου χρήστη μέσω διαδικτύου στο ρομπότ
2	Turbo-X Powerbank 6000 mAh	Τροφοδοτεί με ρεύμα το Raspberry Pi

3	Camera Module Board 5MP Webcam Video 1080p 720p for Raspberry Pi 3	Λαμβάνει την εικόνα το τι βλέπει το ρομπότ
4	Mini 3.5mm Portable Stereo Speaker Amplifier For MP3/MP4/Mobile phone/Tablet	Μετατρέπει σε γραπτά κείμενα (text messages) που στέλνει ο απομακρυσμένος χρήστης μέσω διαδικτύου σε ήχο. (Ο ήχος αυτός ακούγεται πάνω στο ρομπότ). Έτσι ο χρήστης στέλνει κατά κάποιο τρόπο τη φωνή του.
5	8028 USB Plug Mini Studio Microphone	Στέλνει στον απομακρυσμένο χρήστη τους ήχους που ακούγονται στο χώρο που βρίσκεται το ρομπότ.
6	SanDisk 16GB Micro SD SDHC Card	Βρίσκεται τοποθετημένη στο Raspberry Pi και είναι απαραίτητη για τον προγραμματισμό του.
7	Arduino UNO With USB Cable	Η βασική πλακέτα που φέρει το firmware του ρομπότ δηλαδή τις βασικές εντολές κίνησής του με πιο απλό παράδειγμα κίνηση μπρος πίσω δεξιά και αριστερά. Μπορούν να προστεθούν επιπλέον λειτουργίες όπως φώτα ή ρομποτικοί βραχίονες κτλ.
8	2WD Motor Smart Robot Car Chassis Kit Speed Encoder Battery Box for Arduino W1Z3	Το σασί του ρομπότ όπου στην πιο απλή του έκδοση πρέπει να διαθέτει 2 κινητήρες με τροχούς μία μπαταριοθήκη και ένα σκελετό ικανό να κουβαλήσει και τα υπόλοιπα ηλεκτρονικά που αναφέραμε. *(Μπορούμε να χρησιμοποιήσουμε πολύ πιο πολύπλοκα και ακριβά σασί χωρίς να αλλάξει κάτι στη συνδεσμολογία εφόσον έχουν και αυτά 2 τροχούς). Στην εργασία αυτή βάλαμε ένα χειροποίητο σασί με 2 ποιο ακριβούς τροχούς.
9	L298N DC Motor Driver	Ο πιο απλός και φθηνός driver που θέτει σε λειτουργία το δύο μοτέρ του σασί. *(Μπορούμε να χρησιμοποιήσουμε πολύ πιο καλούς και ακριβούς motor driver)

Η κατασκευή η συνδεσμολογία και ο τρόπος λειτουργίας φαίνεται στο ακόλουθο σχεδιάγραμμα:

UGV απομακρυσμένου ελέγχου μέσω Διαδικτύου (συνδεσμολογία)

(Νίκος Γιαννακόπουλος gianakor@gmail.com)



Εικόνα 1: Συνδεσμολογία UGV απομακρυσμένου ελέγχου μέσω Διαδικτύου

Κόστος κατασκευής:

UGV απομακρυσμένου ελέγχου μέσω Διαδικτύου ~ 5 ~ Νίκος Γιαννακόπουλος
Το Συνολικό κόστος της κατασκευής στην πιο απλή του έκδοση (με το πιο απλό σασί) ανέρχεται στα **84 ευρώ** με συνυπολογισμένο κόστος μεταφοράς για προϊόντα από εξωτερικό μέσω e-bay. Παρακάτω φαίνεται ο πίνακας κόστους.

UGV απομακρυσμένου ελέγχου μέσω Διαδικτύου (ΚΟΣΤΟΣ ΥΛΙΚΩΝ)
(Νίκος Γιαννακόπουλος gianakop@gmail.com)

A.A.	ΠΡΟΪΟΝ	ΤΙΜΗ ΣΕ EURO
1	Raspberry Pi 3 Model B+ (B plus) Quad Core 1.4GHz 64 bit CPU wifi & bluetooth	€ 43.00
2	Turbo-X Powerbank 6000 mAh	€ 13.00
3	Camera Module Board 5MP Webcam Video 1080p 720p for Raspberry Pi 3	€ 4.00
4	Mini 3.5mm Portable Stereo Speaker Amplifier For MP3/MP4/Mobile phone/Tablet	€ 4.00
5	8028 USB Plug Mini Studio Microphone	€ 2.00
6	SanDisk 16GB Micro SD SDHC Card	€ 5.00
7	Arduino UNO With USB Cable	€ 4.00
8	2WD Motor Smart Robot Car Chassis Kit Speed Encoder Battery Box for Arduino W1Z3	€ 7.00
9	L298N DC Motor Driver	€ 2.00
ΣΥΝΟΛΟ		€ 84.00

(Νίκος Γιαννακόπουλος gianakop@gmail.com)

Εικόνα 2: Κόστος UGV απομακρυσμένου ελέγχου μέσω Διαδικτύου.

Στη διεύθυνση αποθετηρίου του έργου στο Github:

<https://github.com/gianakop/Web-UGV-> υπάρχει ο σχετικός πίνακας excel, με στήλη που διαθέτει και τα εμπορικά links των παραπάνω προϊόντων και φυσικά και τις τιμές τους που αποδεικνύουν το κόστος της κατασκευής.

Εγκατάσταση λογισμικού:

Η εγκατάσταση του λογισμικού για να λειτουργήσει ένα ρομπότ απομακρυσμένου ελέγχου μέσω διαδικτύου περιλαμβάνει: 1) τη διαμόρφωση της SD κάρτας του Raspberry Pi και την εγκατάσταση του λειτουργικού συστήματος Raspbian, 2) Εγγραφή στον ιστότοπο του LETSROBOT.TV και 3) Εγκατάσταση του firmware του ρομπότ στην πλακέτα του Arduino. Για την διαμόρφωση και αρχικοποίηση του Raspberry Pi καλό είναι να έχει κανείς μια εμπειρία από το Linux αλλά αν αυτό δεν συμβαίνει δεν είναι εντελώς απαραίτητο μιας και τα βήματα περιγράφονται αναλυτικά παρακάτω:

1) Διαμόρφωση της SD κάρτας του Raspberry Pi

Επισκεπτόμαστε το <https://www.raspberrypi.org/downloads/raspbian/>.

Κατεβάζουμε τη συμπιεσμένη μορφή της τελευταίας έκδοσης του Raspbian

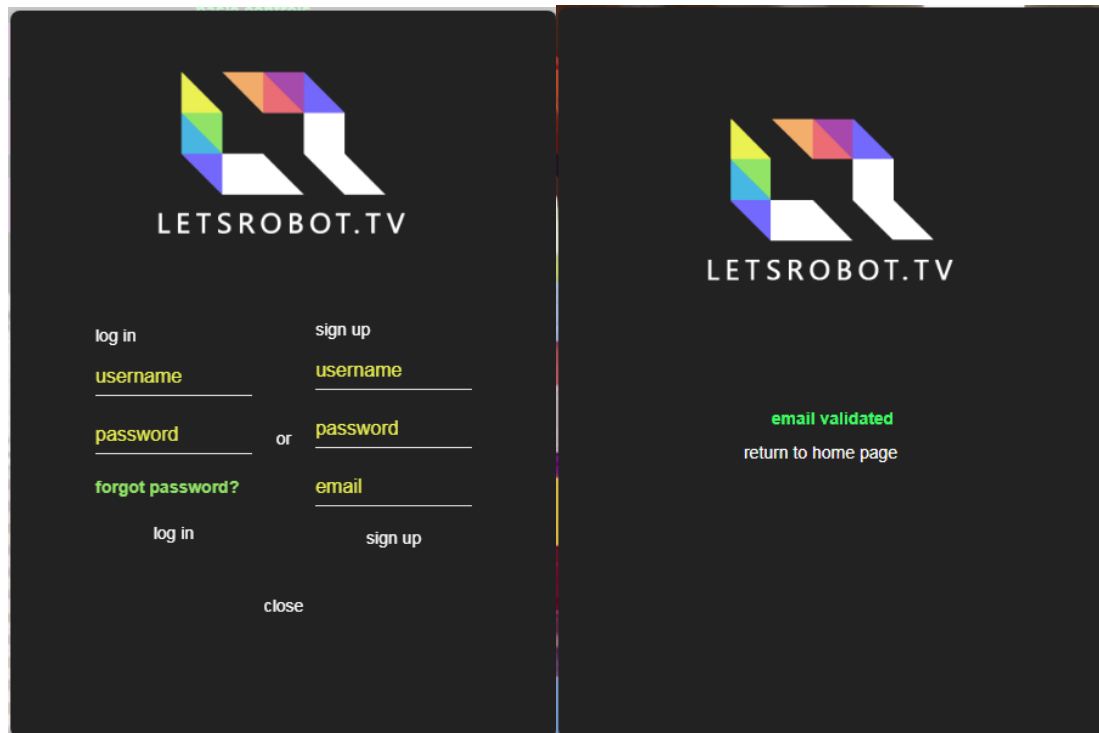


RASPBIAN STRETCH WITH DESKTOP

Με χρήση προγράμματος αποσυμπίεσης αρχείων ανοίγουμε το συμπιεσμένο αρχείο σε ένα φάκελο του υπολογιστή μας και κατόπιν μεταφέρουμε όλο το περιεχόμενο του στην κενή SD κάρτα μας που προορίζεται για το Raspberry Pi, και την οποία έχουμε τοποθετήσει στον υπολογιστή μας σε ένα card reader

Αφαιρούμε την SD Card από τον card reader του υπολογιστή και την τοποθετούμε στο microSD card slot του Raspberry Pi. Βλέπε σχετικό video: (Getting Started With The Raspberry Pi 3 <https://www.youtube.com/watch?v=gbJB3387xUw>)

2) Εγγραφή στον ιστότοπο του LETSROBOT.TV

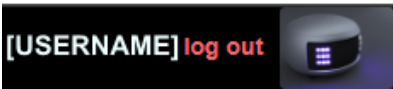



Εικόνα 3: Εγγραφή στο LETSROBOT.TV

Επισκεπτόμαστε τον ιστότοπο: <https://letsrobot.tv/login>.

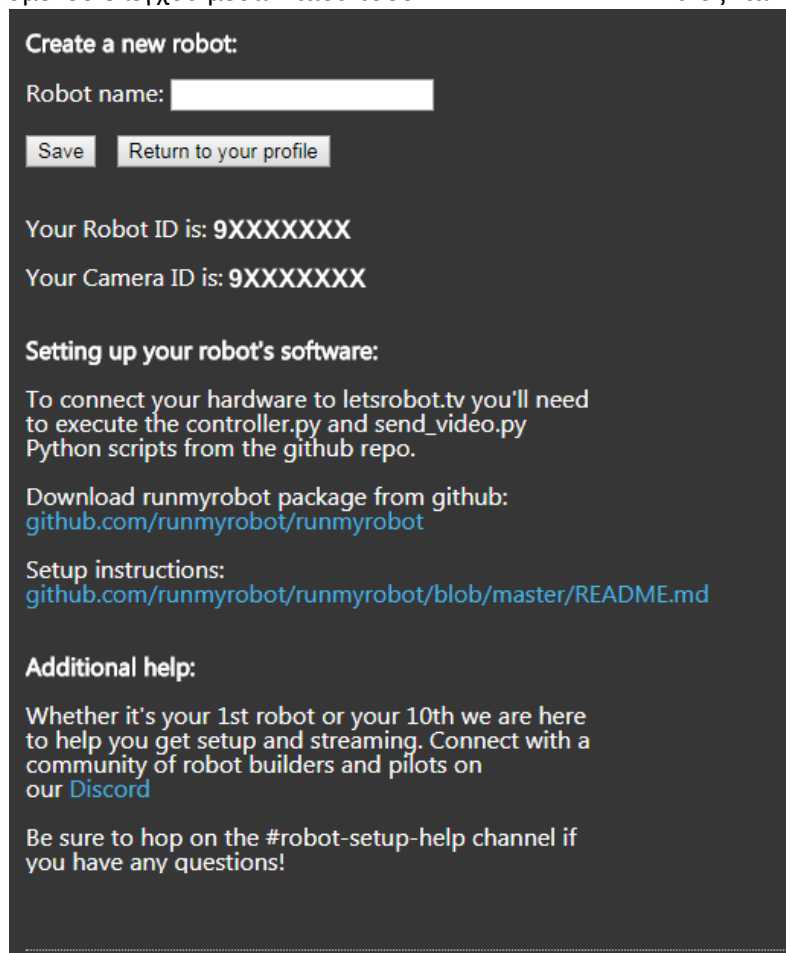
Εάν δεν έχουμε λογαριασμό δίνουμε Username, Password, και Email και πατάμε **sign up** κάτω από τις πληροφορίες που δώσαμε. Το email χρησιμεύει για επαλήθευση της εγγραφής μας. Αφού γραφτούμε επιτυχώς κάνουμε κλικ στο **return to home page** και είμαστε έτοιμοι να συνδέσουμε το Ρομπότ μας.

Website Bot Setup

Αρχίζουμε πατώντας στο Username  στην πάνω δεξιά μεριά του website, η οποία θα μας οδηγήσει στην σελίδα του προφίλ μας. Στη κάτω μεριά της σελίδας του προφίλ υπάρχει κουμπί για να προσθέσουμε ένα ρομπότ (**+ Add a new Robot!**)  το οποίο και πατάμε

Αυτό μας πάει στη σελίδα δημιουργίας νέου ρομπότ (Εικόνα 4)

∴



Create a new robot:

Robot name:

Your Robot ID is: 9XXXXXXX

Your Camera ID is: 9XXXXXXX

Setting up your robot's software:

To connect your hardware to letsrobot.tv you'll need to execute the controller.py and send_video.py Python scripts from the github repo.

Download runmyrobot package from github:
github.com/runmyrobot/runmyrobot

Setup instructions:
github.com/runmyrobot/runmyrobot/blob/master/README.md

Additional help:

Whether it's your 1st robot or your 10th we are here to help you get setup and streaming. Connect with a community of robot builders and pilots on our [Discord](#)

Be sure to hop on the #robot-setup-help channel if you have any questions!

Εικόνα 4: Σελίδα νέου ρομπότ

Σε αυτή τη σελίδα δίνουμε όνομα στο ρομπότ μας και **σημειώνουμε το Robot ID και Camera ID** τα οποία θα χρειαστούμε για να αρχικοποιήσουμε το λογισμικό στην κάρτα του Raspberry Pi. Πατάμε **Save** button και μετά **Return to profile**. Τώρα στο κάτω μέρος της σελίδας του προφίλ υπάρχει επιλογή Robot Setting. Πρίν συνεχίσουμε στο κάτω μέρος πατάμε το κουμπί **Edit Button** στη δεξιά πλευρά του ονόματος του ρομπότ μας και κάνουμε τις απαραίτητες ρυθμίσεις. (* = Default setting)

Toggle Name	State	Description
Public	On*	Επιτρέπει το ρομπότ σου να είναι δημόσιο και να είναι ορατό στον κάθε ένα.
	Off	Επιτρέπει μόνο σε ανθρώπους που έχουν το URL του ρομπότ σου να αλληλεπιδρούν με αυτό.
Anonymous Control	On*	Επιτρέπει σε ανώνυμους (μη εγγεγραμμένους χρήστες) να οδηγούν το ρομπότ σου.
	Off	Επιτρέπει μόνο σε εγγεγραμμένους χρήστες να οδηγούν το ρομπότ σου.
Profanity Filter	On	Σβήνει μηνύματα που ίσως περιέχουν κάποιες λέξεις ακατάλληλες.

	Off*	Οτιδήποτε πληκτρολογείται επιτρέπεται στο chat.
Global Chat	On*	Το ρομπότ σου είναι μέρος του Globe Chat room μαζί με όλα τα υπόλοιπα ρομπότ.
	Off	Έχεις ιδιωτικό chat room για το ρομπότ σου
Show Exclusive Control Button	On*	<i>Μη διαθέσιμο</i> – Προσθέτει ένα κουμπί όπου κάποιος μπορεί να αγοράσει το δικαίωμα να είναι ο μοναδικός οδηγός του ρομπότ σου .
	Off	Το κουμπί αυτό δεν προστίθεται.
Mute Text-to-Speech	On	Απενεργοποιεί τη δυνατότητα να λέει το ρομπότ ότι τυπώνεται στο chat
	Off*	Το Text-to-Speech είναι ενεργοποιημένο
Mic Enabled	On*	Ενεργοποιεί το μικρόφωνο σε ανθρώπους που χρησιμοποιούν το site του ρομπότ σου.
	Off	Απενεργοποιεί το μικρόφωνο σε ανθρώπους που χρησιμοποιούν το site του ρομπότ σου.
Dev Mode	On	Μόνο εσύ οδηγείς το ρομπότ σου, αλλά το chat είναι ενεργοποιημένο για όλους.
	Off*	Το ρομπότ μπορεί να οδηγηθεί και από άλλους.
Custom Panels	On	Σου επιτρέπει να προσθέσεις εξειδικευμένες εντολές λειτουργίας ελέγχου στο ρομπότ σου.
	Off*	Επιτρέπει μόνο τις βασικές εντολές λειτουργίας οδήγησης του ρομπότ.

Αρχικά προτείνεται να θέσουμε το “Dev Mode” στο On και αργότερα να το γυρίσουμε στο off όταν έχουμε αρχικοποιήσει και ελέγξει επιτυχώς όλες τις λειτουργίες του ρομπότ μας ώστε να είναι έτοιμο για χρήση και από άλλους χρήστες. Κάθε φορά που αλλάζουμε τις ρυθμίσεις πατάμε **Save Robot Setting** και μετά **Close** στο παράθυρο που εμφανίζεται.

Άνοιγμα και λειτουργία του Raspberry Pi

Τώρα συνδέουμε και ανοίγουμε το Raspberry Pi, και εγκαθιστούμε το απαραίτητο λογισμικό. Αρχικά πρέπει να συνδέσουμε ένα κοινό πληκτρολόγιο και ποντίκι υπολογιστή στις δύο USB θύρες του Raspberry Pi και να συνδέσουμε την HDMI θύρα Raspberry Pi σε μία οθόνη υπολογιστή με HDMI καλώδιο. Προσοχή η τροφοδοσία του Raspberry Pi πρέπει να είναι 5V 3A το πολύ. Ιδανικό για την τροφοδοσία είναι το (Turbo-X Powerbank 6000 mAh) που περιγράφεται στον πίνακα υλικών του ρομπότ <https://www.plaisio.gr/tilefona/fortistes/powerbanks/Turbo-X-Powerbank-6000mah-2dot1a-Kokkino-S82.htm>

Αλλάζοντας τις τοπικές ρυθμίσεις στο Raspberry Pi

Όταν ανοίξει το Raspbian Desktop πάμε στο Main Menu. Και μετά στο **Preferences > Raspberry Pi Configuration**, .Εκεί κάνουμε κλικ στο Localisation και βάζουμε τις τοπικές ρυθμίσεις της χώρας μας.

Αρχειοποιώντας Wifi and I2C

Επανεκκινούμε το Raspberry Pi πάμε στο Terminal window και πληκτρολογούμε το παρακάτω και πατάμε enter:

```
1. sudo apt-get install -y python-smbus
```

Κατόπιν πάλι στο Terminal window πληκτρολογούμε το παρακάτω και πατάμε enter:

```
1. sudo apt-get install -y i2c-tools
```

Κατόπιν πάλι στο Terminal window πληκτρολογούμε το παρακάτω και πατάμε enter:

```
1. sudo raspi-config
```

Αυτό θα εκτελέσει το Raspberry Pi Software Configuration Tool σε μία οθόνη όπου θα ενεργοποιήσουμε το I2c. Χρησιμοποιούμε τα βελάκια και πάμε στην επιλογή "5. Interfacing Options" και πατάμε Enter ώστε να ενεργοποιήσουμε το I2C. Αφού το ενεργοποιήσουμε κάνουμε reboot:

```
1. sudo reboot
```

Εγκατάσταση λογισμικού

Αφού κάναμε την επανεκκίνηση το Raspberry Pi πρέπει να εγκαταστήσουμε το software που χρειάζεται το ρομπότ. Θα χρησιμοποιήσουμε το script που βρίσκεται στο [Run My Robot Github](https://github.com/runmyrobot/runmyrobot). Για να εγκαταστήσουμε το λογισμικό αυτό Terminal window πληκτρολογούμε το παρακάτω και πατάμε enter:

```
1. sudo wget https://raw.githubusercontent.com/runmyrobot/runmyrobot/master/scripts/install.sh -O /tmp/install.sh && bash /tmp/install.sh
```

Μόλις το script αρχίσει να τρέχει μας ζητάει να βάλουμε το **Robot ID**, το οποίο έχουμε σημειώσει κάπου από το βήμα της δημιουργίας της σελίδας του ρομπότ μας. Εάν το όχετε ξεχάσει αυτό καθώς και το **Camera ID** μπορείτε να πάτε πάλι στο προφίλ χρήστη στο letsrobot.tv να πάτε στο κάτω μέρος και να πάρετε την πληροφορία. Πληκτρολογήστε το Robot ID στο Terminal window, ελέγξτε το με προσοχή ότι είναι σωστό και πατήστε Enter.

Μετά το script θα σας ζητήσει να πληκτρολογήσετε το Camera ID. Όταν βάλετε και το Camera ID στο Terminal window, ελέγξτε το και πάλι με προσοχή ότι είναι σωστό και πατήστε Enter.

Το script θα αρχίσει να εκτελείται για κάμπωση ώρα (από 20 λεπτά μέχρι 1 ώρα εξαρτάται από πολλούς παράγοντες) στην διάρκεια αυτή θα εγκαταστήσει όλο το απαιτούμενο λογισμικό που χρειάζεται το ρομπότ σου ώστε να κάνει live streaming.

Τροποποιώντας το Controller.py και ξεκινώντας το ρομπότ.

Αφού εγκατασταθεί το script και σταματήσει να τρέχει η διαδικασία εγκατάστασης , θα εμφανιστεί στο terminal `pi@raspberrypi:~$` . Τότε κλείστε το terminal window πατώντας το X στο πάνω δεξιό μέρος του Terminal window.

Μετά κάντε κλικ στο εικονίδιο με τους δύο φακέλους στην πάνω αριστερή μεριά του Raspbian Desktop. Αυτό θα ανοίξει το file explorer του Raspbian στο `/home/pi`. Πηγαίνετε μέσα στο φάκελο `runmyrobot` .Μετά κάντε δεξί κλικ στο αρχείο `Controller.py` και στο μενού που ανοίξει πατήστε "Open". Αυτό θα ανοίξει το Thonny, το οποίο είναι ένας python script editor, και με το οποίο μπορείτε να κάνετε μικρές αλλαγές στο `Controller.py` ανάλογα με τους drivers που χρησιμοποιεί το ρομπότ σας. Πηγαίνετε στην γραμμή 29 και αλλάξτε το `.4` σε `.3`, ώστε η διορθωμένη γραμμή να είναι η παρακάτω:

```
1. parser.add_argument('--turn-delay', type=float, default=0.3)
```

Αποθηκεύστε τις αλλαγές σας στο **File>Save**. Τότε κλείστε το terminal window πατώντας το X στο πάνω δεξιό μέρος του Terminal window

Το τελευταίο που μένει είναι να τσεκάρετε αν το ρομπότ σας αυτόματα συνδέεται με το `letsrobot.tv` όταν ανοίγεται το διακόπτη τροφοδοσίας του. Για να το επαληθεύσετε ανοίξτε το Terminal window. πληκτρολογούμε το παρακάτω και πατάμε enter:

```
1. sudo nano home/pi/start_robot
```

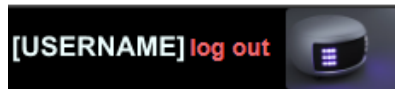
Αυτό θα ανοίξει έναν text editor στο Terminal που ονομάζεται Nano, ο οποίος θα ανοίξει τις πληροφορίες που περιέχει το αρχείο `start_robot`. *Εκεί θα πρέπει να δείτε τα Robot ID και Camera ID του ρομπότ σας.* Αφού επαληθεύσετε ότι το `start_robot` έχει τα σωστά IDs πατήστε **Ctrl+X** να βγείτε από το nano. Τότε στο Terminal window πληκτρολογούμε το παρακάτω και πατάμε enter:

```
1. sudo shutdown -h now
```

Αυτό θα τερματίσει τη λειτουργία του Raspberry Pi. Όταν το Pi τερματίσει αποσυνδέστε την τροφοδοσία, το πληκτρολόγιο το ποντίκι και το καλώδιο HDMI από το Raspberry Pi.

Απενεργοποίηση του Dev Mode

Αφού το ρομπότ έχει πια ρυθμιστεί σωστά απενεργοποιούμε το Dev Mode ώστε να μπορούν να το χειρίζονται και άλλοι εκτός από εμάς.



για πρόσβαση στη σελίδα του προφίλ σας. Μετά στο κάτω μέρος της σελίδας του προφίλ κάντε κλικ στο Edit Button στη δεξιά μεριά του ονόματος του ρομπότ. Στο κάτω μέρος του panel αλλάξτε το "Dev Mode" σε Off (και πατήστε **Save Robot Setting** και μετά **Close**).

Συγχαρητήρια έχετε δημιουργήσει ένα ρομπότ απομακρυσμένου ελέγχου μέσω διαδικτύου με τη βοήθεια του letsrobot.tv. ,έτοιμο να ελεγχθεί και να αλληλεπιδράσει με ανθρώπους απ' όλο τον πλανήτη μέσω του διαδικτύου.

Για τυχόν απορίες και βοήθεια μπορείτε ανατρέξετε στο Διεύθυνση αποθετηρίου στο Github: <https://github.com/gianakop/Web-UGV-> ή να επικοινωνήσετε με το δημιουργό του έργου Νίκο Γιαννακόπουλο στο gianakop@gmail.com

3) Εγκατάσταση του firmware του ρομπότ στην πλακέτα του Arduino

Το firmware του ρομπότ μπορεί να ποικίλει ανάλογα με την κατασκευή μας και φυσικά εξαρτάται από τα pins του Arduino όπου έχουμε συνδέσει τον έλεγχο των κινητήρων το είδος του motor controller, τις διάφορες επιπλέον συσκευές που έχουμε τοποθετήσει πάνω στο ρομπότ (βραχίονες φώτα κτλ.) καθώς και τους τυχόν αισθητήρες. Παρακάτω παρουσιάζεται ένα ενδεικτικό firmware που λειτουργεί με την πιο απλή βασική κατασκευή και συνδεσμολογία που περιγράφεται στην εικόνα 1 σελ 4.

Ο βασικός κώδικας γραμμένος σε Arduino IDE για κίνηση του ρομπότ στις 4 βασικές κατευθύνσεις (εμπρός πίσω δεξιά και αριστερά σύμφωνα με τις εντολές που θα λαμβάνει μέσω διαδικτύου από το δικτυακό interface του LETSROBOT TV) είναι ο παρακάτω (Εικόνα5):



```
arduino_letsrobot
#include "simple_hbridge.h"

int fspeed= 255;
int hspeed = 125;
int bufferSize = 0;
long finishCommandTimestamp = 0;
long currentTimestamp = 0;

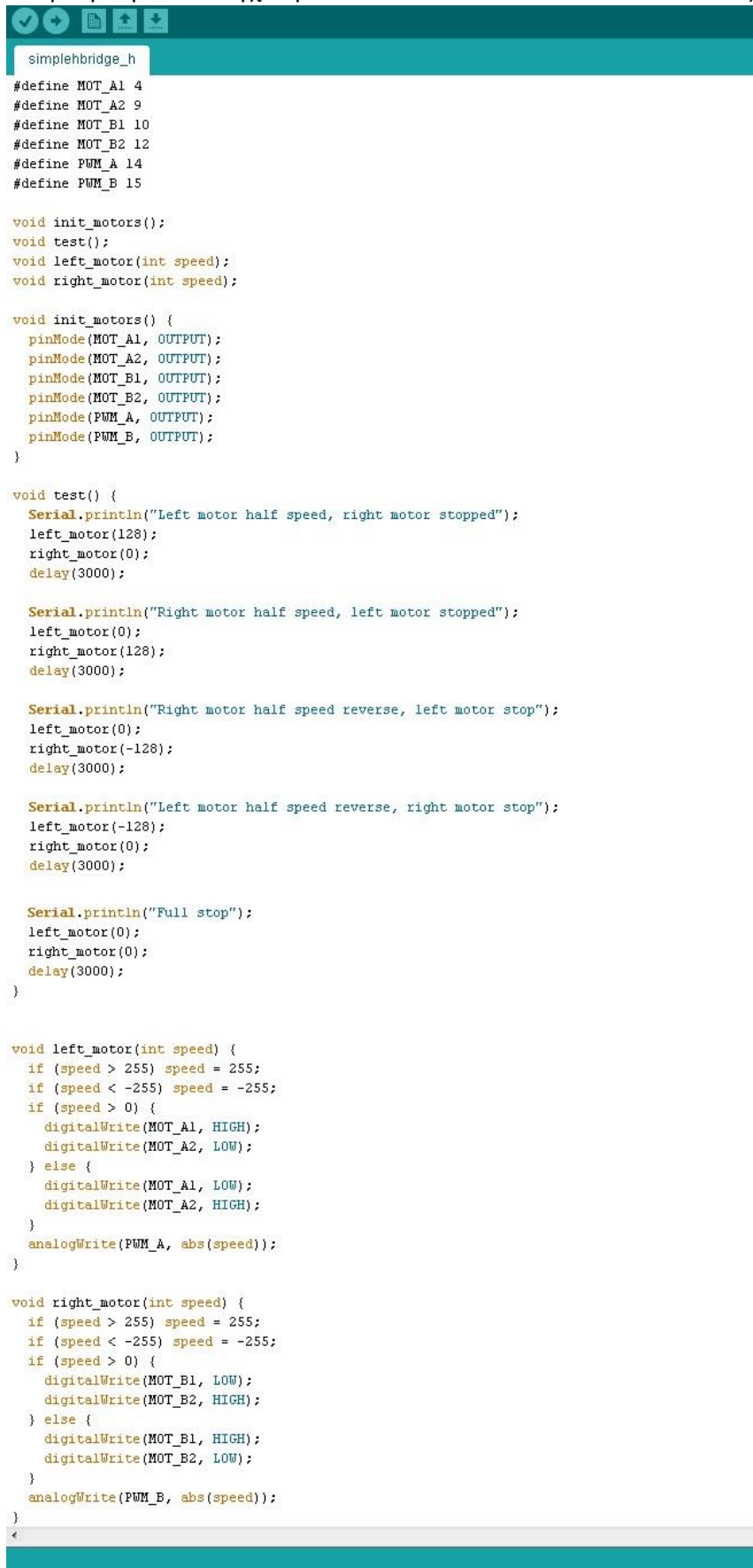
void setup() {
  Serial.begin(9600);
  init_motors();
  finishCommandTimestamp = millis();
  currentTimestamp = millis();
}

void loop() {
  currentTimestamp = millis();
  if (currentTimestamp > finishCommandTimestamp) {
    left_motor(0);
    right_motor(0);
  }
  if (Serial.available()) {
    char inByte = Serial.read();
    delay(2);
    while (Serial.available()) {
      char tempByte = Serial.read();
      delay(2);
      if (tempByte != 10 && tempByte != 13) inByte = tempByte;
    }
    switch (inByte) {
      //_____Motors_____
      case 'f': // forward
        left_motor(fspeed);
        right_motor(fspeed);
        reset_ending_timestamp();
        break;
      case 'b': // back
        left_motor(-fspeed);
        right_motor(-fspeed);
        reset_ending_timestamp();
        break;
      case 'r': // right
        left_motor(fspeed);
        right_motor(hspeed);
        reset_ending_timestamp();
        break;
      case 'l': // left
        left_motor(hspeed);
        right_motor(fspeed);
        reset_ending_timestamp();
        break;
      default:
        delay(10);
    }
  }
}

void reset_ending_timestamp() {
  finishCommandTimestamp = millis()+500;
}
```

Εικόνα 5: Κώδικας για κίνηση του ρομπότ στις 4 βασικές κατευθύνσεις

Ο κώδικας της βασικής βιβλιοθήκης simple_hbridge.h είναι ο παρακάτω:



```
simplehbridge_h

#define MOT_A1 4
#define MOT_A2 9
#define MOT_B1 10
#define MOT_B2 12
#define PWM_A 14
#define PWM_B 15

void init_motors();
void test();
void left_motor(int speed);
void right_motor(int speed);

void init_motors() {
    pinMode(MOT_A1, OUTPUT);
    pinMode(MOT_A2, OUTPUT);
    pinMode(MOT_B1, OUTPUT);
    pinMode(MOT_B2, OUTPUT);
    pinMode(PWM_A, OUTPUT);
    pinMode(PWM_B, OUTPUT);
}

void test() {
    Serial.println("Left motor half speed, right motor stopped");
    left_motor(128);
    right_motor(0);
    delay(3000);

    Serial.println("Right motor half speed, left motor stopped");
    left_motor(0);
    right_motor(128);
    delay(3000);

    Serial.println("Right motor half speed reverse, left motor stop");
    left_motor(0);
    right_motor(-128);
    delay(3000);

    Serial.println("Left motor half speed reverse, right motor stop");
    left_motor(-128);
    right_motor(0);
    delay(3000);

    Serial.println("Full stop");
    left_motor(0);
    right_motor(0);
    delay(3000);
}

void left_motor(int speed) {
    if (speed > 255) speed = 255;
    if (speed < -255) speed = -255;
    if (speed > 0) {
        digitalWrite(MOT_A1, HIGH);
        digitalWrite(MOT_A2, LOW);
    } else {
        digitalWrite(MOT_A1, LOW);
        digitalWrite(MOT_A2, HIGH);
    }
    analogWrite(PWM_A, abs(speed));
}

void right_motor(int speed) {
    if (speed > 255) speed = 255;
    if (speed < -255) speed = -255;
    if (speed > 0) {
        digitalWrite(MOT_B1, LOW);
        digitalWrite(MOT_B2, HIGH);
    } else {
        digitalWrite(MOT_B1, HIGH);
        digitalWrite(MOT_B2, LOW);
    }
    analogWrite(PWM_B, abs(speed));
}
```

Εικόνα 6: Κώδικας της βασικής βιβλιοθήκης simple_hbridge.h

Τελειώνοντας υπενθυμίζω ότι μπορείτε να οδηγήσετε live το ρομπότ στην παρακάτω σελίδα:

<https://letsrobot.tv/robocaster/gianakop/robot/59206902>.

Ευχαριστώ,

Νίκος Γιαννακόπουλος (καθηγητής πληροφορικής ΠΕ86)

gianakop@gmail.com