ECE 5397 Final Project

WEB BASED CONTROLLER FOR OWI-ROBOT

Aurash Norouzi and Samuel Aragon

Wireless Web-based OWI Robot Control

The purpose of this project was to design and implement a web-based wireless controller to manipulate the OWI-robot joints while simultaneously providing the end-user with a live camera feed in both a first-person/virtual and third-person/overhead view. The project requires the user to emulate the procedures laid out in Benedict's manual for using the Arduino with the robot and the PCB board that was designed in fall 2017. The manual is included below. Our project made use of a Raspberry Pi 3 (RPI3) in order to communicate via the serial link to the Arduino and host the website on the local network which both the RPI3 and user device are connected to. We loaded the RPI3 with a version of Linux, dubbed, Ubuntu Mate. This is a lightweight version of Linux designed to be used with RPI's. In order to live-stream the camera feeds we used the *Motion* library that allows for streaming multiple cameras on local IP addresses which can be accessed via the HTML website. We utilize python and flask to host our website and process the user's commands from the controller. The flask app that is created in the python program allows commands such as pressed buttons on the website to correlate to events in our python environment. Once the event is captured in the python file, we use the serial module of python in order to send a command to the Arduino which is listening at a defined baud rate for our commands regarding which motor to rotate and when to stop rotation. In order to provide a convenient user experience, we designed the controller buttons as, "touch and hold", buttons which translates to an immediate stop upon unclicking the button. In other words, as long as the button is depressed the motor will rotate. Initially, this caused problems on the mobile site, yet these were solved by checking to make sure what type of device the user is employing. The most recent iteration that can be found at the provided GitHub link makes use of CSS, *cascading style sheets*, to provide the user with a more custom design. Unfortunately, while all features are accessible when used with a mobile phone, the screen dimensions cannot handle our buttons which may cause some distortion when used on mobile devices. Our first iterations that can be seen in the provided YouTube videos, lacked CSS, yet they were more user friendly on the mobile end. Nevertheless, the end user can control the OWI robot from any location within the local network so long as the RPI3 is connected to the same network. We will provide startup directions below after Benedict's manual. Our YouTube videos demonstrate these features and provide demos for use cases such as playing and pausing music or muting devices.

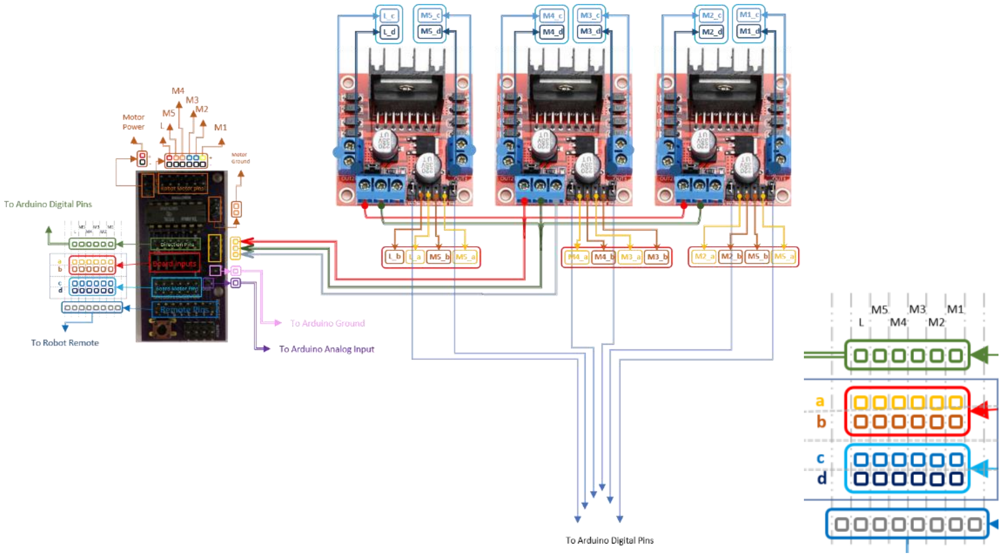
Wireless Web-based OWI Robot Control

Benedict Ischei’s Manual for Arduino Matlab startup below

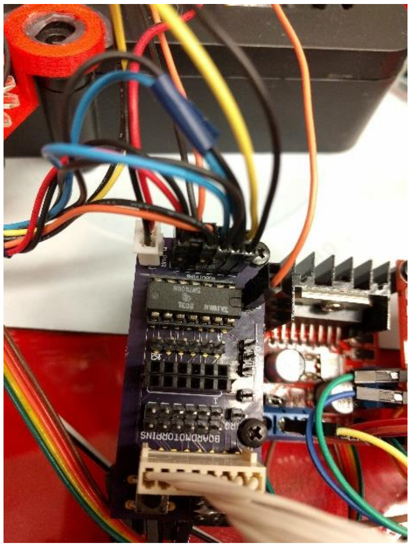
WARNING: DO NOT ATTEMPT TO CONTROL THE ARM WITH THE ARDUINO WHILE THE SHUNT IS ON THE PCB

Only one 5v driver terminal needs to be connected to the PCB

Part 1: Assembly



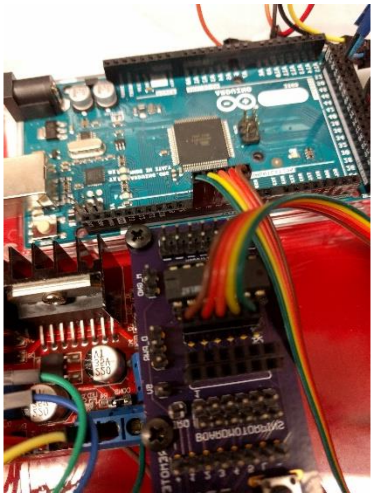
- First up, connect the Robot’s cables to the PCB.



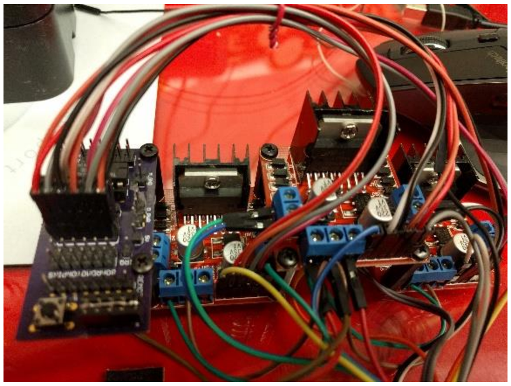
- Then connect the Remote Control AND shunt to the board and use them to ensure you have the pins setup correctly.

Wireless Web-based OWI Robot Control

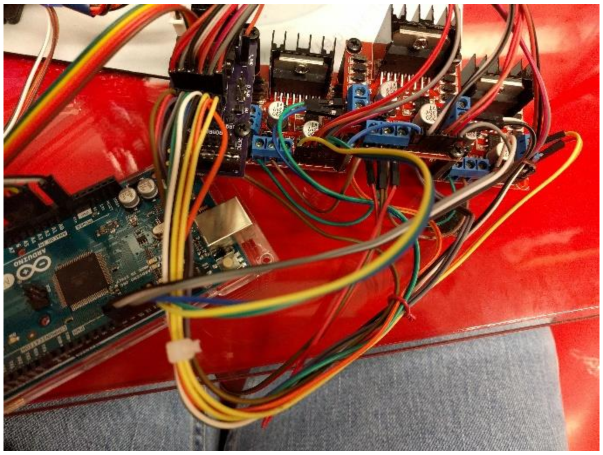
- Unplug the shunt and remote.
- Next, Connect the direction pins from the arduino to the PCB. You can ignore the pin for the light. (you only need 5 cables, since you don't need a direction for the light).



- Next, connect the Direction pins from the motor drivers to the PCB

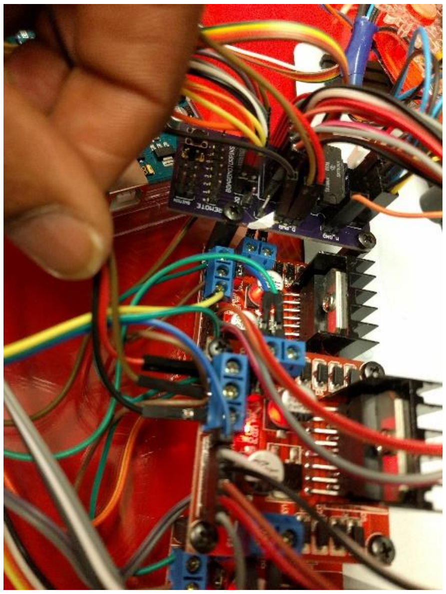


- Connect the Board motor pins on the PCB to the Driver outputs, and connect the enable pins to the arduino

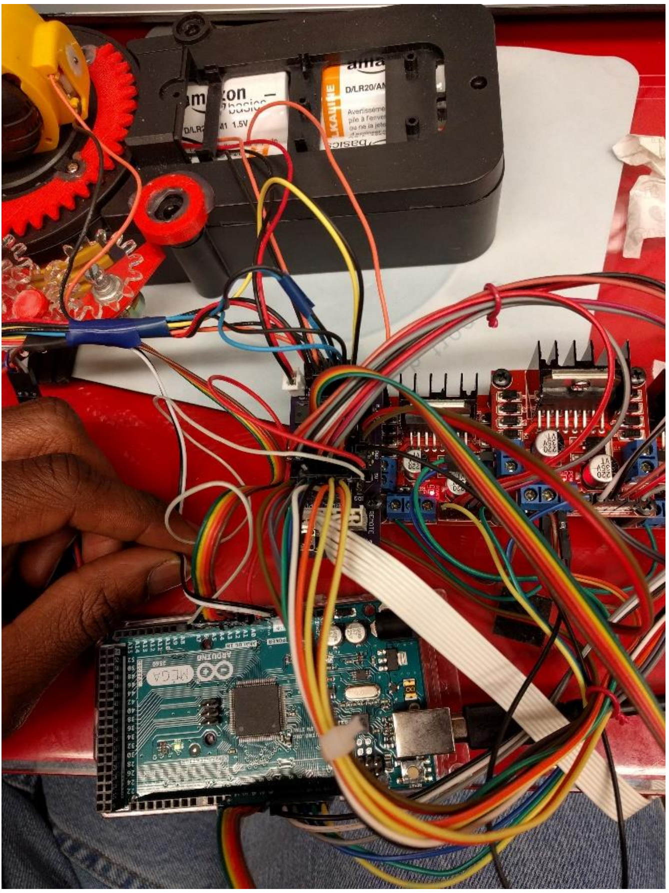


- Connect the driver power cables, and the boards should light up. Remember, you only need to connect ONE 5v terminal from the drivers to the PCB

Wireless Web-based OWI Robot Control



- Next, connect a single cable from the NEG pin on the PCB to an Arduino Ground pin, and the IRQ pin to an arduion analog input pin (I used A5)
- This is what the finished connection looks like.

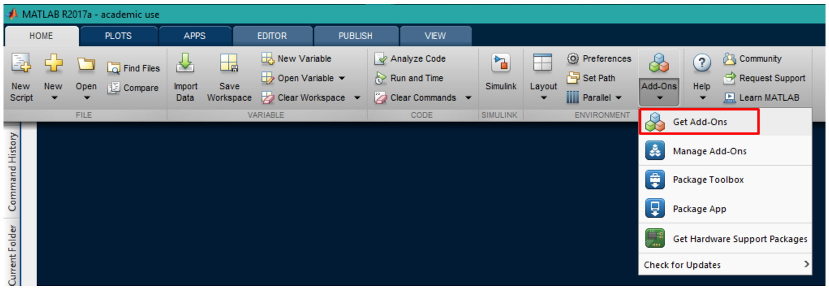


.

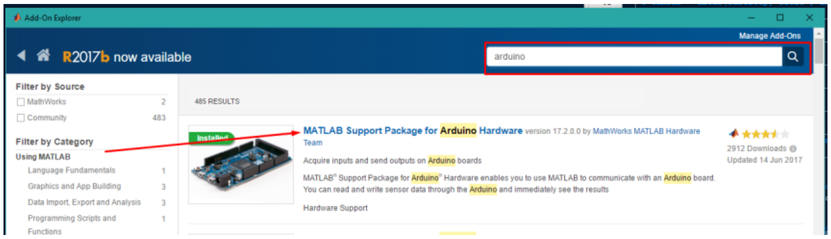
Wireless Web-based OWI Robot Control

Part 2: Matlab

Navigate to Add- Ons in the Home tab group, and select Get Add- Ons



Search for Arduino, then install the Support Package for Arduino Hardware



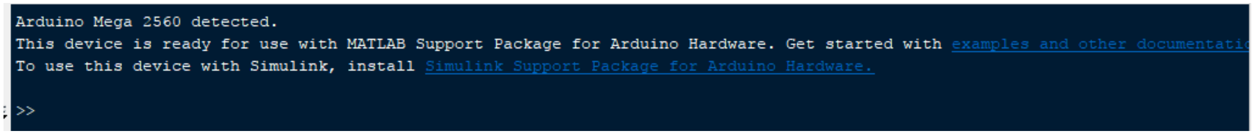
MATLAB Support Package for Arduino Hardware version 17.2.0.0 by MathWorks MATLAB Hardware Team

Acquire inputs and send outputs on Arduino boards

MATLAB® Support Package for Arduino® Hardware enables you to use MATLAB to communicate with an Arduino board. You can read and write sensor data through the Arduino and immediately see the results

Hardware Support

If it installs correctly, you should get a message everytime you plug an arduino into your computer.

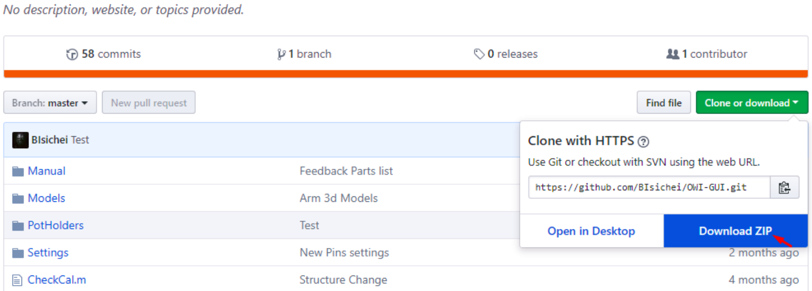


```
Arduino Mega 2560 detected.
This device is ready for use with MATLAB Support Package for Arduino Hardware. Get started with examples and other documentatio
To use this device with Simulink, install Simulink Support Package for Arduino Hardware.

>>
```

With the included PCB, the motor directions can be controlled by a 1 or 0 sent to the Direction pins(Green group in fig. 1)

GUI:

- If you choose to use the GUI, Download the files from https://github.com/BIsichei/OWI-GUI



No description, website, or topics provided.

BIsichei Test

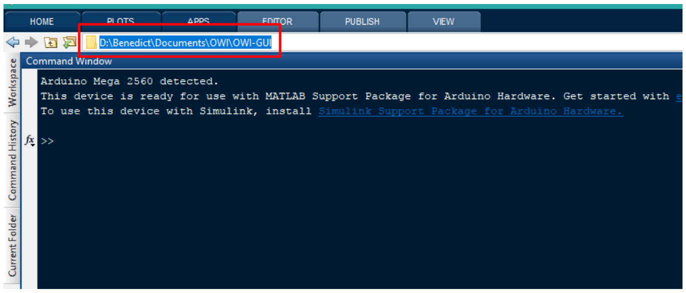
Use Git or checkout with SVN using the web URL.

https://github.com/BIsichei/OWI-GUI.git

Manual	Feedback Parts list	
Models	Arm 3d Models	
PotHolders	Test	
Settings	New Pins settings	2 months ago
CheckCal.m	Structure Change	4 months ago

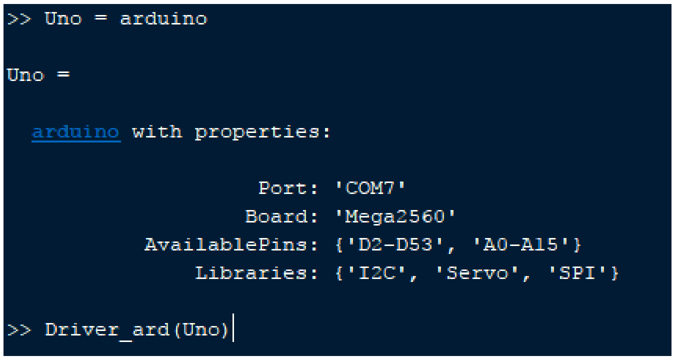
- Unizip the files, and navigate to the download location in matlab

Wireless Web-based OWI Robot Control



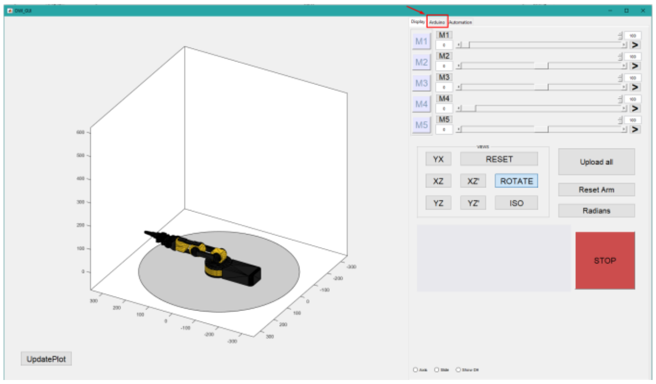
```
Arduino Mega 2560 detected.
This device is ready for use with MATLAB Support Package for Arduino Hardware. Get started with e
To use this device with Simulink, install Simulink Support Package for Arduino Hardware.

>>
```

- Next, create an arduino object, and pass it to the Driver_ard function

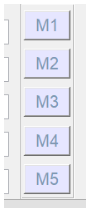


```
>> Uno = arduino

Uno = 

  arduino with properties:

                    Port: 'COM7'
                   Board: 'Mega2560'
           AvailablePins: {'D2-D53', 'A0-A15'}
               Libraries: {'I2C', 'Servo', 'SPI'}

>> Driver_ard(Uno)
```

- The GUI should popup.
- Navigate to the Arduino Tab



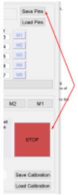
- Click load Pins, and select the fall2017pins.mat file in the settings folder.
- Select Apply All Values.
- Cross check, and make sure the pin values are correct
- To ensure your connection has been setup correctly, try the purple Virtual remote controls.



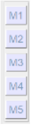
- If they differ from the movement you'd get if you were using the physical remote (moving down when you push up), reverse the appropriate wire pair on the BoardMotorpins section of the PCB.
- Next, click one of the Motors in the Calibration tab, then follow the instructions to calibrate the robot. You must calibrate all motors to progress.

Wireless Web-based OWI Robot Control

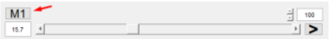
- When done, save Your new Calibration and/or your pin settings (if you changed them)



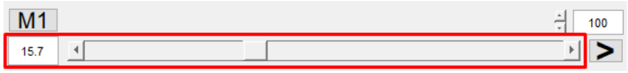
Button Functions:



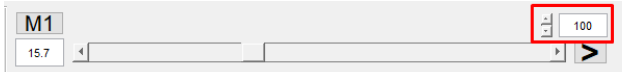
Virtual Remote



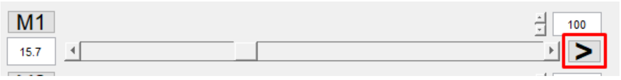
reset joint



Joint value and slider.



Joint power value and slider.



Upload Joint to board

Wireless Web-based OWI Robot Control

Manual for setting up our web based controller

After you have used the manual above to setup your connections, if possible test the connections with the matlab driver that is provided. Also note the PWM pins for each motor and the direction enable pins as well. These will be useful upon setting up the Arduino ino file that we have provided. Next, I will provide the steps to re-create our project.

SUPPLIES:

1. ***OWI Robot Arm set up as described above***
2. ***Raspberry PI 3***
3. ***Micro SD Card for RPI3 software (UBUNTU MATE)***
4. ***Two USB Webcams***
5. ***HDMI and monitor to configure RPI3***

STEP 1:

Download UBUNTU MATE for RPI3 using the following website as a guide.

https://ubuntu-mate.org/blog/ubuntu-mate-for-raspberry-pi-3/

STEP 2:

Login to RPI3 and download motion by doing the following

1. Run the following command to update the OS on Raspberry Pi:

   sudo apt-get update

2. Next you will have to install the "motion" library

   sudo apt-get install motion

3. Set Motion daemon to "yes" by editing: */etc/default/motion* to make it look like the following: "start_motion_daemon=yes".

   sudo nano /etc/default/motion

4. Now access the etc folder with root privileges and change the motion.conf file with the motion.conf provided in our github. Also, be sure to add the two files named thread0.conf and thread1.conf

5. Performing the following will begin to host both cameras on two available ports on the local network.

```
sudo /etc/init.d/motion restart
```

Step 3:

Install flask on the RPI3 in order to host our website.

```
sudo pip install Flask
```

Step 4:

Download our code from the GitHub repository at the below link.

https://github.com/aurashn/ECE-5397-Final-Project

Step 5:

You will have to change the pins on the pi_owi.ino file to your respective pin configuration. I will provide one example below.

```
if(in=='l'){ //rotate left
digitalWrite(30,0);
digitalWrite(2,1);
```

The above code checks for the left rotation command and sets the direction pin 30 to 0 or LOW while changing the PWM pin to 1 or HIGH. Thus, the user should change 30 to the direction pin of the base rotation motor and test to make sure a LOW will cause left rotation. Also, the pin 2 should be changed to the correct PWM pin for that motor. This procedure should be carried out for each motor and each command condition in the .ino file. Upon completing this step, be sure to load this file onto your Arduino.

Step 6:

Change the img src links in the following line of code within the robot.html in the PI_OWI directory to the local IP address of your RPI3 appended by the port 8081 and 8082 for each camera that is to be displayed by the website.

```
<

    img id="picture"  class="picture" src=http://10.0.0.10:8081

>

<

    img id="picture2"  class="picture" src=http://10.0.0.10:8082

>
```

Step 7:

Using the terminal on the RPI3 access the PI_OWI directory and run the following command:

```
sudo python pi_owi.py
```

This command will launch the website which can be accessed at the local IP address appended by a :5010.

Step 8:

Enjoy! Feel free to make improvements to the code. Also, this code provides a very solid base to create a mobile robot. Just add two motors and four wheels while making a few modifications to the code to have a mobile robot with dual livestream feeds. We believe this project allows the user to gain a solid understanding of how numerous systems can communicate and provides an opportunity to work on web development for use in robotic applications. This project was most definitely worth doing for us as we learned valuable skills in regards to robot control, serial data transfer, local website hosting, HTML, CSS, and user design for applications aimed at robotics. Thus, I would recommend future students to use our code as a guide to develop other projects that aim to have a wireless interface.