Autonomous Vehicle

Build Instructions

Khemar Bryan

# Contents

1. Correct Web Template Usage
2. System Diagram
3. Budget
4. Time Commitment
5. Mechanical Assembly
6. PCB/Soldering
7. Power Up
8. Unit Testing
9. Production Testing

**Web Template Usage**

The web page is fairly simple to use but to do this you must at least have basic knowledge of html coding.

To use the web template, you must:

* Clone the <https://github.com/six0four/six0four.github.io> repository
* Copy the indexcontent.hml, indexmenu.html, index.html and template folder to your own repository
* Open the web page by going to yourusername.github.io in a browser, if this does not work you should go to yourusername.github.io/indexcontent.html

After using the template, you can now make changes to the web page from the indexcontent and indexmenu files in the repository. It is not necessary to make nay changes to the index.html file after the first commit.

Adding a new section in the indexmenu.html file allows you to jump to section of the page after clicking the link in the menu

To add a new section to the menu in the in the project you should:

* **Step 1**

Go to indexmenu.html in your repository

* **Step 2**

Add a new section and label it

* **Step 3**

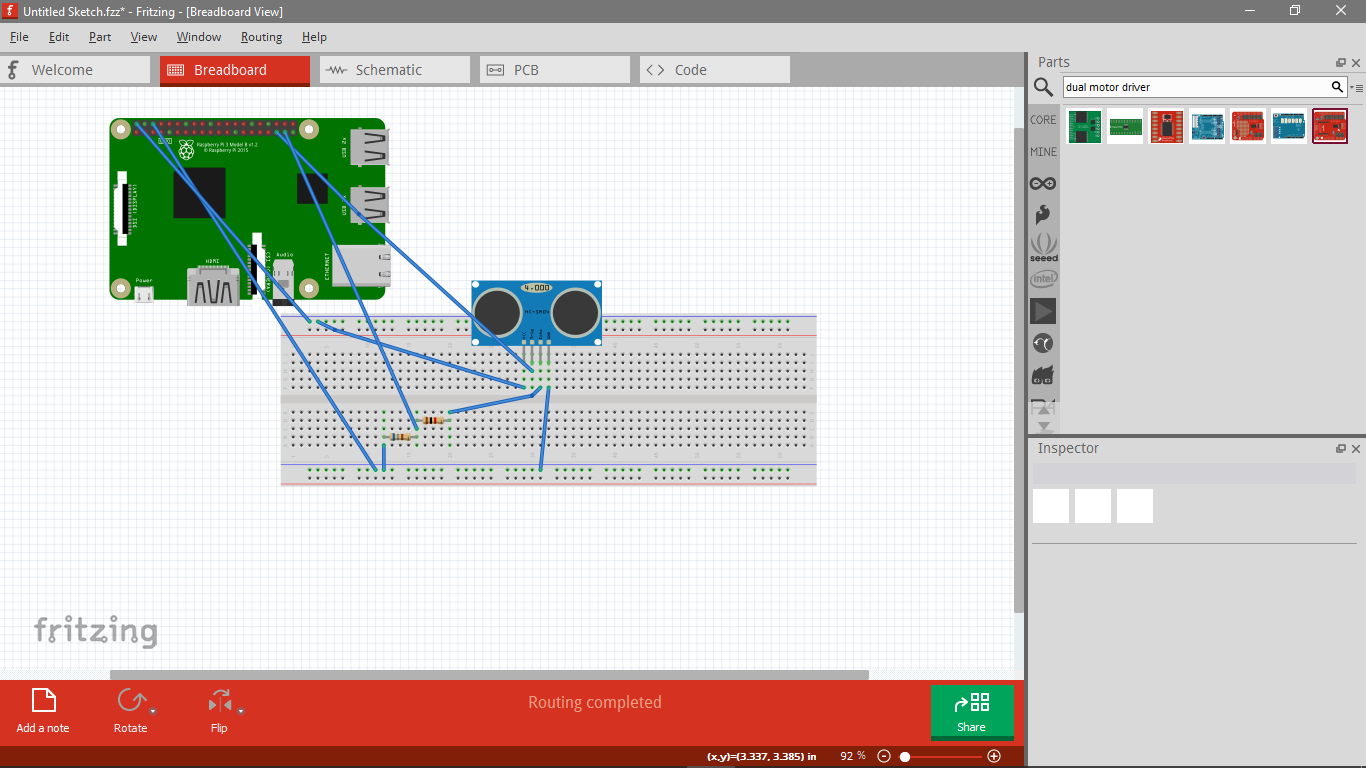
Commit the changes.

* **Step 4**

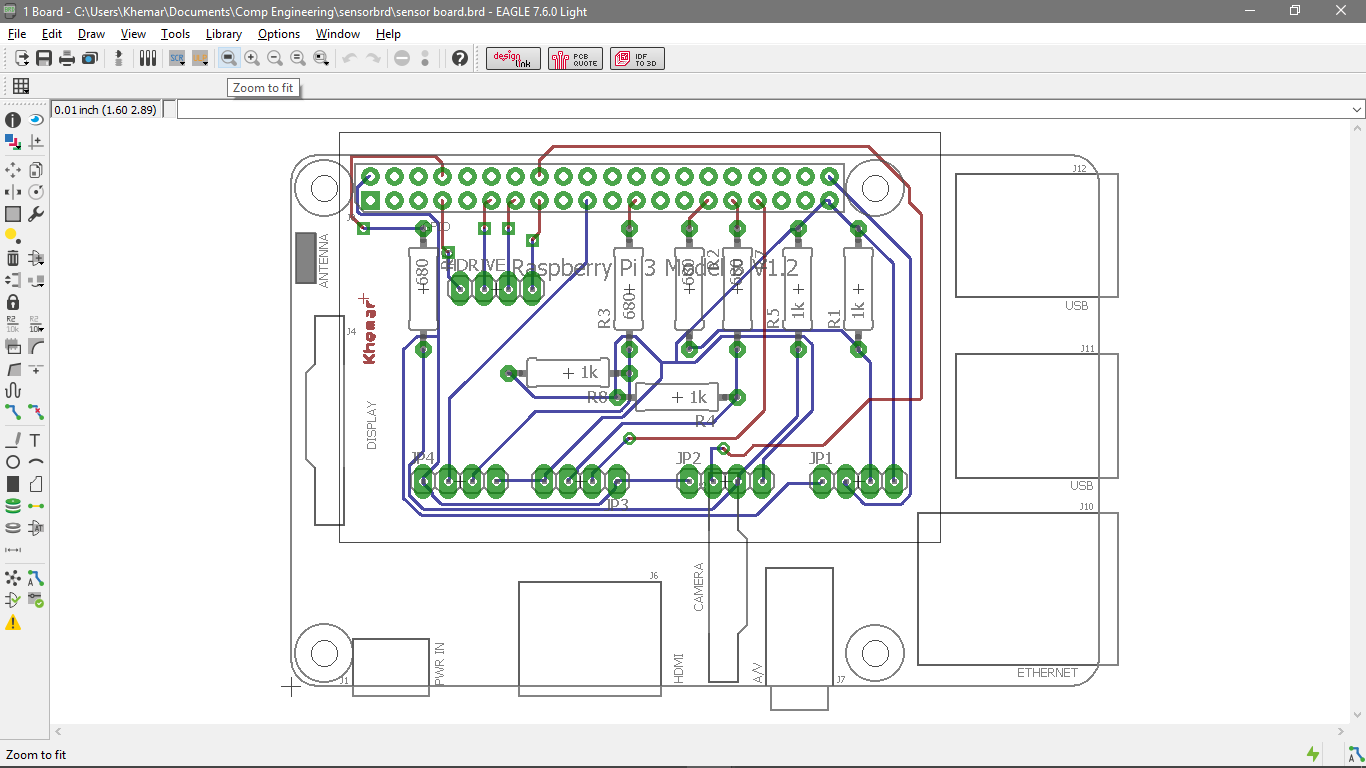
These changes allowed you to access the new week from the side menu, you must now create the content in the indexcontent.html file for the menu link to jump to.

To add new content to the page itself you must do through the indexcontent.hmtl file. You must create a section in the file and use the same section label you created in the indexmenu file eg. (if your label in indexmenu is section #13 then it must be section #13 in indexcontent). After doing this you can add anything you want to the section. (Remember to close all tags)

**System Diagram**



HC SR04 Diagram



You can choose between using a breadboard or building a PCB from that schematic.

**Budget**

For this project, you will need to budget at least $300 as prices may vary at time of purchase

|  |  |  |  |
| --- | --- | --- | --- |
| **Autonomous RC Car Budget 2016** | | | |
| **Item** | **Cost (Before Taxes)** | **Taxes (15%)** | **Cost (After Taxes)** |
| Raspberry Pi Kit | $ 99.99 | $ 13.00 | $ 112.99 |
| Hwydo 4WD Robot Smart Car Chassis | $ 38.00 | $ 4.94 | $ 42.94 |
| Energizer Max AA Batteries (4 Pack) | $ 4.97 | $ 0.65 | $ 5.62 |
| HC-SR04 Ultrasonic Sensors(x2) | $ 4.99 | $ 0.65 | $ 5.64 |
| 6" Male to Male Jumper Wires(x10) | $ 2.97 | $ 0.39 | $ 3.36 |
| 6" Male to Female Jumper Wires(x10) | $ 2.98 | $ 0.39 | $ 3.37 |
| 6" Female to Female Jumper Wires(x10) | $ 2.99 | $ 0.39 | $ 3.38 |
| L298 Dual Motor Driver -2A | $ 14.00 | $ 1.82 | $ 15.82 |
| Total | $ 170.89 |  | $ 193.11 |

\*N.B – this budget does not account for breadboards and resistors which you will also need.

**Time Commitment**

Approximate time for project completion : 1 year

Time required for first stage:12 weeks (10 weeks ideally)

Amount of hours required per week: 6 hours

**Mechanical Assembly**

To assemble the car chassis, you will need a screw driver and all the parts listed below:

Parts included in the kit:

* 2x Acrylic car floor (Body of the car)
* 8x Acrylic fasteners
* 4x coded disks
* 28x M3 Screw
* 14x M3 nuts
* 6x M3-30 copper columns
* 6x M3 screws
* 4 wheels
* 4x gear motors

Follow the steps in the video below to build the car.

<https://www.youtube.com/watch?v=uW8YVcBjPGU>

It is recommended that you connect the motors for the wheels that are on the same side.

The motor driver(L298) and the raspberry pi can be mounted anywhere on the car using the screw holes in each.

To connect the Motor Driver to the motors, follow these steps:

1. Loosen the screws on the OUT ports (OUT1, OUT2, OUT3, OUT4)
2. Put the red and black leads from the right motors of the car into OUT2 and OUT1 respectively then tighten the screws.
3. Place the red and black wires from the left side into OUT4 and OUT3 respectively then tighten the screws.

After the motor driver is connected to the motors you must now connect the driver to the raspberry pi and the power supply

1. Loosen the screws on the 5V, round and VCC ports
2. Take the red lead from the power supply and place it into the VCC port then take the black lead and push it into the ground port. Tighten the screws on the VCC port but leave the ground port open for now
3. Take a male to female wire and plug the header onto pin 2 on your raspberry pi (this is the 5V out pin) then push the male head into the 5V port on the motor driver.
4. Using another male to female wire, plug the header onto pin 6 on the raspberry pi and push the male head into the ground port on the motor driver.
5. You can now tighten the screws on the 5V and ground ports

To set up the sensors you can either use a breadboard and follow the breadboard layout seen in the system diagram above. Basic knowledge of connections are made on a breadboard is required. You will need the following

* 1K and 680 Ohm resistors
* The HC SR04 Sensor
* Male to male wires
* Male to female wires

**PCB/Soldering**

Soldering onto the PCB is a simple task that even beginners can carry out but it is recommended that you have at least basic knowledge of circuitry. Practice before you solder onto your pcb to minimize mistakes

**Step 1**

* Get a soldering iron, a PCB holder and some solder.

**Step 2**

* Place the PCB into the PCB holder and plug the soldering iron in to heat it up

**Step 3**

* When the LED on the soldering iron has turned green this means you can begin soldering
* Start by cleaning the putting some solder onto the iron and wipe it off on a sponge (this is done to clean the iron)

**Step 4**

* Look at your schematic then begin soldering on the necessary parts. (MAKE SURE TO FOLLOW SCHEMATIC)

**Power Up**

In this phase, you will be powering up your car and PCB(if you chose to use one)

**Powering up the PCB**

* The raspberrypi by default comes loaded with an OS but it is recommended you reinstall it
* Format the SD card then use <http://sourceforge.net/projects/win32diskimager/> to write the image file <https://downloads.raspberrypi.org/raspbian/images/raspbian-2016-09-28/2016-09-23-raspbian-jessie.zip> onto it.
* Using the HDMI cable from the rapsberrypi kit connect the pi to a monitor.
* Using an Ethernet cable connect the pi to a source of internet.(or use wifi)
* Put the sd card into your raspberrypi and power it on.
* When the raspberry pi is powered on look for the ip address (hover cursor over internet icon or use ifconifg in the terminal)
* Open remote pc, enter the ip address and connect to the pi.
* Go to the six0four.github.io repository and look in the [ModularSenseHatStripped](https://github.com/six0four/six0four.github.io/tree/master/ModularSenseHatStripped) folder for a file called traffic 2B.c. Download the file to your desktop.
* Put the PCB header onto the raspberry pi’s pins
* Go to the terminal and run the traffic 2B.c program, the LED’s on the PCB should now start changing colours.

**Powering up the Car**

* Put the batteries into the battery holder
* If the H Drive is receiving power from the batteries the LED should illuminate.
* Remove one of the batteries from the battery holder to turn of the motors as you will need to test if the raspberry pi is powering the motor driver
* Plug in the raspberry pi and the LED on the motor driver should illuminate once again
* Now put the batteries back into the battery holder and all components should be powered up

**Unit Testing**

**Testing the H Drive & Battery Power Up**

* Place the batteries into battery holder and the LED should turn on and shine red

**Testing H Drive & Raspberry Pi Power Up**

* Plug the raspberry pi into a power source and the LED should turn on.

**Testing Raspberry Pi**

* Plug the raspberry pi in and connect it to a monitor.
* You should be greeted by a screen asking you to enter your login information
* If you haven’t made any modifications then the default credentials are

Username: pi Password: raspberry

* Open the browser and go to any website
* If the pi is connected to internet you should have no problem

**Testing the motors**

* Ensure all connections are made according to the build instructions
* Power up the raspberry pi and login
* Go onto my repository github.com/Khemar1/khemar1.github.io/Documentation and download the fwdback.py program
* Go to your terminal and go to the directory where you downloaded fwback.py
* Run the program using sudo python fwdback.py
* If all connections were made correctly then the wheels should move forward and backward

**Testing the Sensor**

* To test the sensor the steps are similar to that of the motors.
* Power up the raspberry pi and login
* Go to the Documentation folder on the repository and download the program sensor2.py
* Open your terminal and go into the directory where you downloaded the program
* Run the program using sensor2.py
* Place an object in front of the sensor and it should return the distance of the object to the screen.

**Product Testing**

When unit testing has been complete and all problems resolved, you can now test the car as a whole. Follow these steps to test that all components work together:

* Power up the raspberry pi
* Put in the batteries
* Connect the raspberry pi to a monitor or use remote pc to view the desktop
* Login to the raspberry pi
* Open your browser and go to to the documentations folder in my repository and download the program auto1.py
* Elevate the car so that it isn’t touching the ground (You can place it on top of something)
* Go to your terminal and navigate to where you downloaded auto1.py
* Run the program using sudo python auto1.py
* The wheels should move forward continuously
* Wave your hand in front of the sensors and the wheels should change direction

You can further test the car by placing it in an open area and watching it go

* Follow the previous steps but use remote pc to connect instead of using a monitor. This is important because you want the car to move freely
* Place the car in the open area
* Run the program again
* The car should move forward continuously
* When the car senses an obstacle it should reverse and change direction(whether it be left or right. This is choses randomly by the program)
* If it senses an obstacle in the direction it turned to then it will turn once again.
* The car will continuously do this until it cannot find a path to continue