Autonomous Ride-on Car

A STEM Project for High Schools

Ver. 1.0, March 2022

roblab.org/rideon

UWA Robotics Club
UWA Robotics & Automation Lab

Robotics is a fantastic research area and an ideal STEM project for high schools. As robot platforms are very expensive and model cars are rather limited, we decided to develop a robotics platform based on a kids' ride-on car. These cars already have drive-by-wire built-in and are fully remote controlled, so the only hardware interfacing required is to modify the remote transmitter and to add a remote emergency stop for safety reasons. The result is an open-ended autonomous vehicle project for high schools. The information given in this document can be freely distributed and we hope we will get a number of high schools to participate and build their own autonomous ride-on car (ARC), maybe even set up a competition.

The goal for each team is to let the car drive along a sequence of given GPS way points. We decided to not include too much detail on application software, because we want high school teams to develop their own programs rather than copy our solution.

Please feel free to contact us for any questions (see addresses at the end of this document) and, of course, send us some videos of your builds. We hope this project will create an interest to study Automation & Robotics Engineering at UWA:
`https://www.uwa.edu.au/study/courses/automation-and-robotics-engineering`

Bill of Material – *Your Shopping List*

(see separate file for easier access to web links)

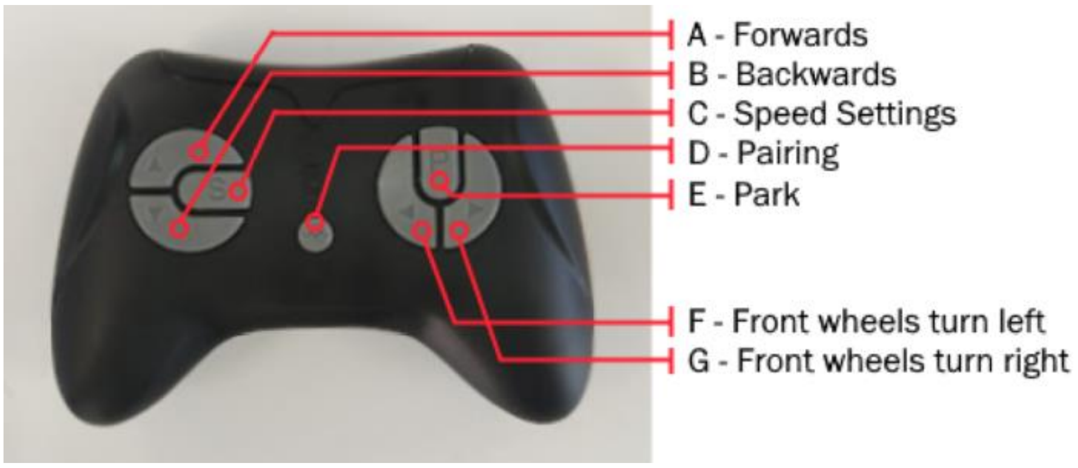
PART CODE	DESCRIPTION	QUANTITY	UNIT COST	TOTAL	SUPPLIER	USED FOR?
RIDE ON CAR	Beach Buggy Dune, 400W	1	$699.00	$699.00	ArtInToys	Car
aditional remote transmitter	additional transmitter for integrstion with embedded controller	1		$0.00	ArtInToys	Car
GPS	Columbus GPS module, USB	1	$65.00	$65.00	eBay	Car
Lidar	optional: Lidar distance sensor for Raspbery Pi	0	$121.67	$0.00	RS Online	Car
H0330	186Lx146Wx75Hmm IP65 Sealed ABS Enclosure	1	$33.25	$33.25	Altronics	Enclosure for electronics
AA1017A	FOB43301L PentaFOB Single Large Button 433Mhz Remote C	1	$45.00	$45.00	Altronics	Emergency stop switch
A1025A	PCR43301RE Penta Series 433Mhz 1 Channel Receiver	1	$69.95	$69.95	Altronics	Emergency stop switch
Z6302G	Raspberry Pi 4 Model B Single Board Computer 4GB	1	$114.40	$84.00	Element14	Controller
Z6513	5" LCD 800 x 480 HDMI Touchscreen For Raspberry Pi	1	$59.00	$59.00	Pi Australia	Controller
P7400	0.5m Thin High Speed HDMI with Ethernet Cable	1	$22.95	$22.95	Altronics	Controller
P1895A	0.3m A Male to Micro B Male USB 2.0 Cable	1	$6.30	$6.30	Altronics	Controller
P2240	2 Way Splicing Terminal Block	2	$1.40	$2.80	Altronics	Electronics Wiring
P2241	3 Way Splicing Terminal Block	2	$1.45	$2.90	Altronics	Electronics Wiring
Voltage converter	12V/24V TO 5V 10A 50W DC DC STEP DOWN CONVERTER V	1	$9.00	$9.00	DayGreen or eBay	Power Electronics
H1216	M3 x 10mm Tapped Spacer Pk 8	1	$4.50	$4.50	Altronics	Mounting Electronics
H3120A	M3 x 10mm Pan Pozi Nickel Bolt Pk 25	1	$2.60	$2.60	Altronics	Mounting Electronics
DA0328	Ultra Micro SDHC Card 16GB	1	$9.95	$9.95	eBay	Pi Operating System
CONTROLLER						
Z1042	NPN BC548B T092h General Purpose Transistor	7	$0.35	$2.45	Altronics	Controller Modification
R7046	1k 0.25W 5% Carbon Film Resistor PK 10	1	$0.60	$0.60	Altronics	Controller Modification
P1021	Pin To Socket 30 Way Prototyping Ribbon Strips 175mm	1	$5.20	$5.20	Altronics	Controller Modification
P1023	Socket To Socket 30 Way Prototyping Ribbon Strips 150mm	1	$5.20	$5.20	Altronics	Controller Modification
ADDITIONAL WIRING						
P5752	JST 2 Way 2.5mm Crimp Housing	1	$0.75	$0.75	Altronics	Car Wiring
P5750	Crimp Pin 2.5mm (Suit P 5752-P 5756) Single	2	$0.05	$0.10	Altronics	Car Wiring
P0622	2.1mm Metal Chassis Mount DC Power Socket	2	$4.50	$9.00	Altronics	Car Wiring
W2176	18AWG Black Double Insulated Speaker Cable	1	$1.40	$1.40	Altronics	Car Wiring
P6717	0.5m 2.1mm DC Plug to 2.1mm DC Plug Cable	1	$8.00	$8.00	Altronics	Car Wiring
			TOTAL	**$1'148.90**		

Link
https://www.artintoys.com.au/product/400-w-24-v-beach-buggy-dune-kids-ride-on-car-white/
Given to us free of charge from ArtInToys
https://www.ebay.com.au/itm/280873472724?hash=item41655d12d4:g:NQgAAOSwYjdgPDpo
https://au.rs-online.com/web/p/sensor-development-tools/2037609
https://www.altronics.com.au/p/h0330-ritec-186lx146wx75hmm-ip65-sealed-abs-enclosure/
https://www.altronics.com.au/p/aa1017a-elsema-pentafob-fob43301l-433mhz-single-large-button-remote-control/
https://www.altronics.com.au/p/a1025a-elsema-penta-pcr43301re-433mhz-1-channel-receiver/
https://au.element14.com/raspberry-pi/rpi4-modbp-4gb/raspberry-pi-4-model-b-4gb/dp/3051887
https://raspberry.piaustralia.com.au/products/5-inch-lcd-hdmi-touch-screen-display-for-raspberry-pi-4#description
https://www.altronics.com.au/p/p7400-dynalink-0.5m-thin-high-speed-hdmi-with-ethernet-cable/
https://www.altronics.com.au/p/p1895A-0.3m-a-male-to-micro-b-male-usb-2.0-patch-cable/
https://www.altronics.com.au/p/p2240-2way-32a-400v-ac-splicing-terminal-block/
https://www.altronics.com.au/p/p2241-3way-32a-400v-ac-splicing-terminal-block/
https://daygreen.com/products/12v-24v-to-5v-10a-50w-dc-dc-step-down-converter-voltage-regulator-1
https://www.altronics.com.au/p/h1216-m3-x-10mm-tapped-spacer-pk-8/
https://www.altronics.com.au/p/h3120a-m3-x-10mm-pan-pozi-nickel-bolt-pk-25/
https://www.ebay.com.au/itm/Micro-SD-Card-SanDisk-16GB-32GB-64GB-128GB-Ultra-Class-10-Mobile-Phone-Memory-A1/163988445093
https://www.altronics.com.au/p/z1042-npn-bc548-t092h-general-purpose-transistor/
https://www.altronics.com.au/p/r7048-1k5-0.25w-carbon-film-resistor-pk-10/
https://www.altronics.com.au/p/p1021-pin-to-socket-30-way-prototyping-ribbon-strips/
https://www.altronics.com.au/p/p1023-socket-to-socket-30-way-prototyping-ribbon-strips/
https://www.altronics.com.au/p/p5752-oupiin-2-way-2.5mm-crimp-housing/
https://www.altronics.com.au/p/p5750-crimp-pin-2.5mm-suit-p-5752-p-5756-single/
https://www.altronics.com.au/p/p0622-2.1mm-female-metal-chassis-mount-dc-power-socket/
https://www.altronics.com.au/p/w2176-24-0.2-black-double-insulated-figure-8-cable/
https://www.altronics.com.au/p/p6717-0.5m-2.1mm-dc-plug-to-2.1mm-dc-plug-cable/

Hardware Description

Remote Controller

- The controller has 7 buttons to correspond to different functions of the ride-on car, as shown below.

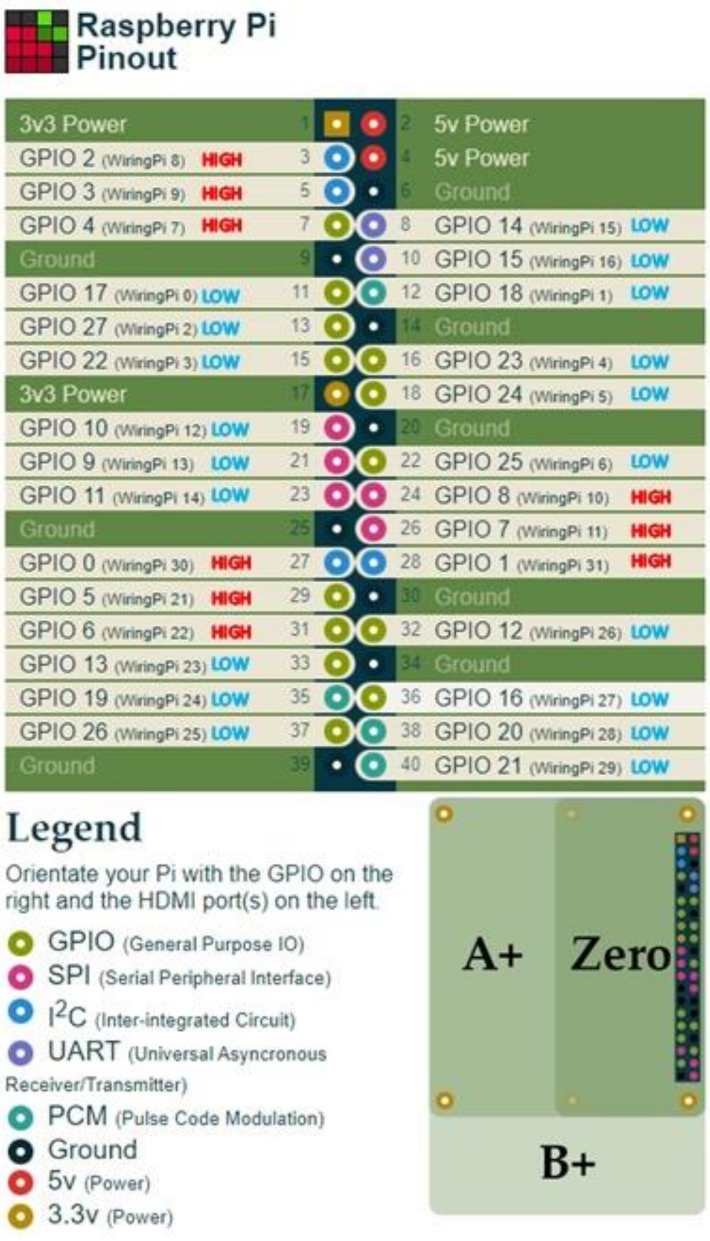


- NOTE: There are three LEDs on the controller which indicate information about the car.
- When changing the speed setting, 1 lit LED will indicate low speed, up to a maximum of 3 lit LEDs to indicate high speed.
- NOTE: To pair the remote with the car, the pairing button must be held down for 3 seconds, after which the controller LED begins to flash on and off. You must then power on the car to complete pairing. Once pairing completes, the LED will stop flashing.
- NOTE: If all three LEDs are flashing, the car is in parking mode.
- NOTE: The LEDs will "sleep" if no buttons are pressed in the last 5 or so seconds. This does not mean the pairing connection has been broken.
- NOTE: If attempting to hold down two buttons that act in opposite directions (such as pressing forwards and backwards), the remote will stop working for 3 seconds.

- The controller's circuit board is rated for ~3V. This should be supplied by the Raspberry Pi's 3v3 pin, which supplies 3.3V.

Raspberry Pi

- This is a useful website to determine which output pins are suitable on the Raspberry Pi: https://pinout.xyz/#
- Using GPIO pins to control the remote is recommended.
- NOTE: Each GPIO Pin may correspond to a different Wiring Pin number. The Wiring Pin number is what is used in the code, when any digitalWrite calls are made.
- NOTE: GPIOs 0 to 8 have a default HIGH signal every time the Pi is turned on. GPIOs 9 to 27 have a default state of LOW. For this reason, we avoid using GPIOs 0 to 8.



Raspberry Pi Pinout

3v3 Power	1	2	5v Power
GPIO 2 (WiringPi 8) HIGH	3	4	5v Power
GPIO 3 (WiringPi 9) HIGH	5	6	Ground
GPIO 4 (WiringPi 7) HIGH	7	8	GPIO 14 (WiringPi 15) LOW
Ground	9	10	GPIO 15 (WiringPi 16) LOW
GPIO 17 (WiringPi 0) LOW	11	12	GPIO 18 (WiringPi 1) LOW
GPIO 27 (WiringPi 2) LOW	13	14	Ground
GPIO 22 (WiringPi 3) LOW	15	16	GPIO 23 (WiringPi 4) LOW
3v3 Power	17	18	GPIO 24 (WiringPi 5) LOW
GPIO 10 (WiringPi 12) LOW	19	20	Ground
GPIO 9 (WiringPi 13) LOW	21	22	GPIO 25 (WiringPi 6) LOW
GPIO 11 (WiringPi 14) LOW	23	24	GPIO 8 (WiringPi 10) HIGH
Ground	25	26	GPIO 7 (WiringPi 11) HIGH
GPIO 0 (WiringPi 30) HIGH	27	28	GPIO 1 (WiringPi 31) HIGH
GPIO 5 (WiringPi 21) HIGH	29	30	Ground
GPIO 6 (WiringPi 22) HIGH	31	32	GPIO 12 (WiringPi 26) LOW
GPIO 13 (WiringPi 23) LOW	33	34	Ground
GPIO 19 (WiringPi 24) LOW	35	36	GPIO 16 (WiringPi 27) LOW
GPIO 26 (WiringPi 25) LOW	37	38	GPIO 20 (WiringPi 28) LOW
Ground	39	40	GPIO 21 (WiringPi 29) LOW

Legend

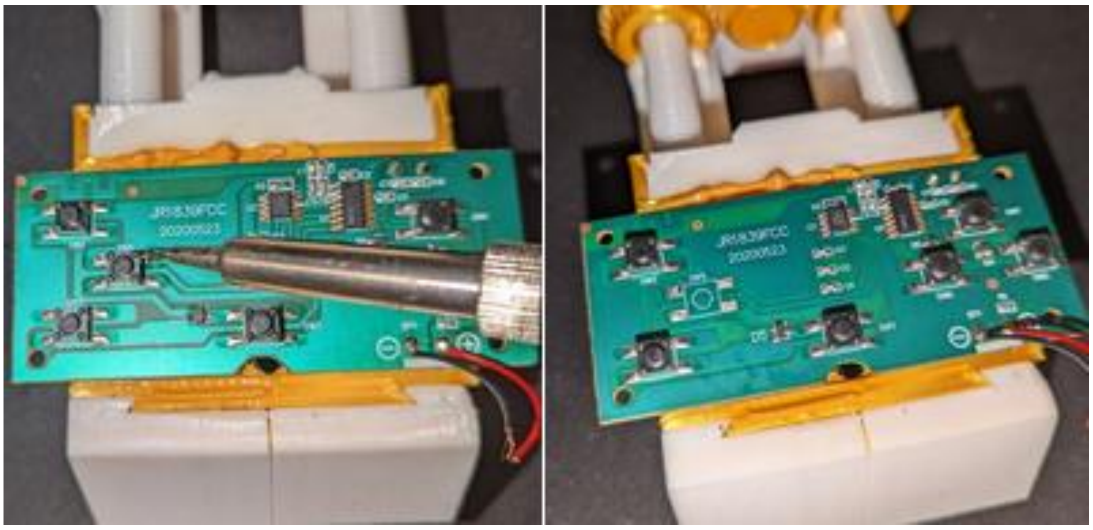
Orientate your Pi with the GPIO on the right and the HDMI port(s) on the left.

- ● GPIO (General Purpose IO)
- ● SPI (Serial Peripheral Interface)
- ● I²C (Inter-integrated Circuit)
- ● UART (Universal Asyncronous Receiver/Transmitter)
- ● PCM (Pulse Code Modulation)
- ● Ground
- ● 5v (Power)
- ● 3.3v (Power)

Implementation Steps

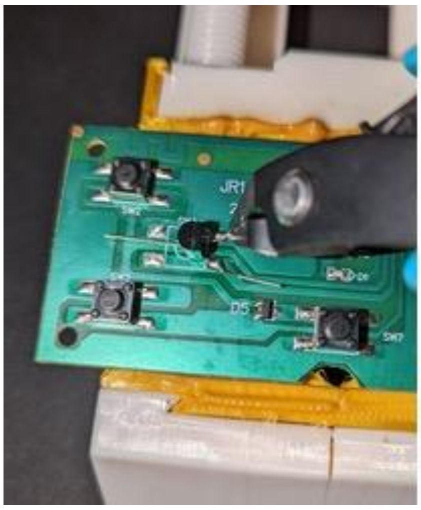
Remote Control Mod

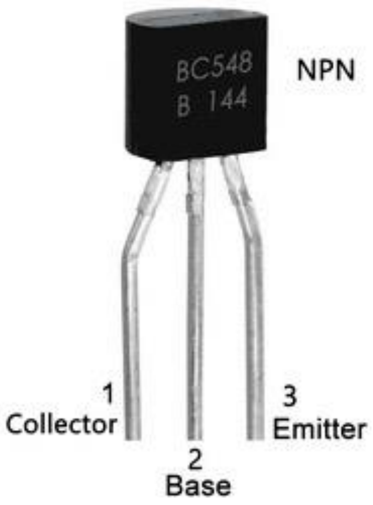
- Take out the circuit board from the remote controller. De-solder each of the 7 push buttons from the circuit board.



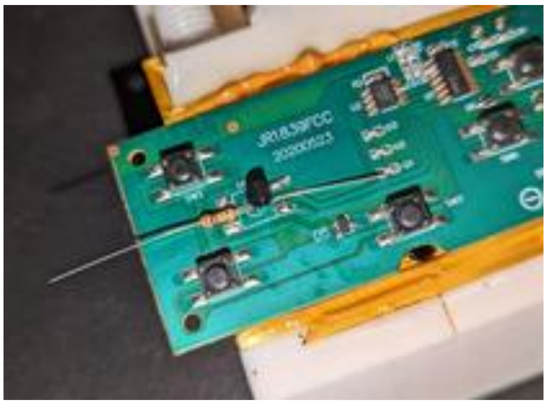
- Solder BC548 transistors in place of the buttons. Connect the emitter pin to a ground plate, and the collector pin to live.

  NOTE: The light green sections of the circuit board are electrically connected. This can be used to find the ground side of the push buttons. The ground plates will be electrically connected to the section with a negative ground symbol. Similarly, the live plates are electrically connected to where the live symbol is.

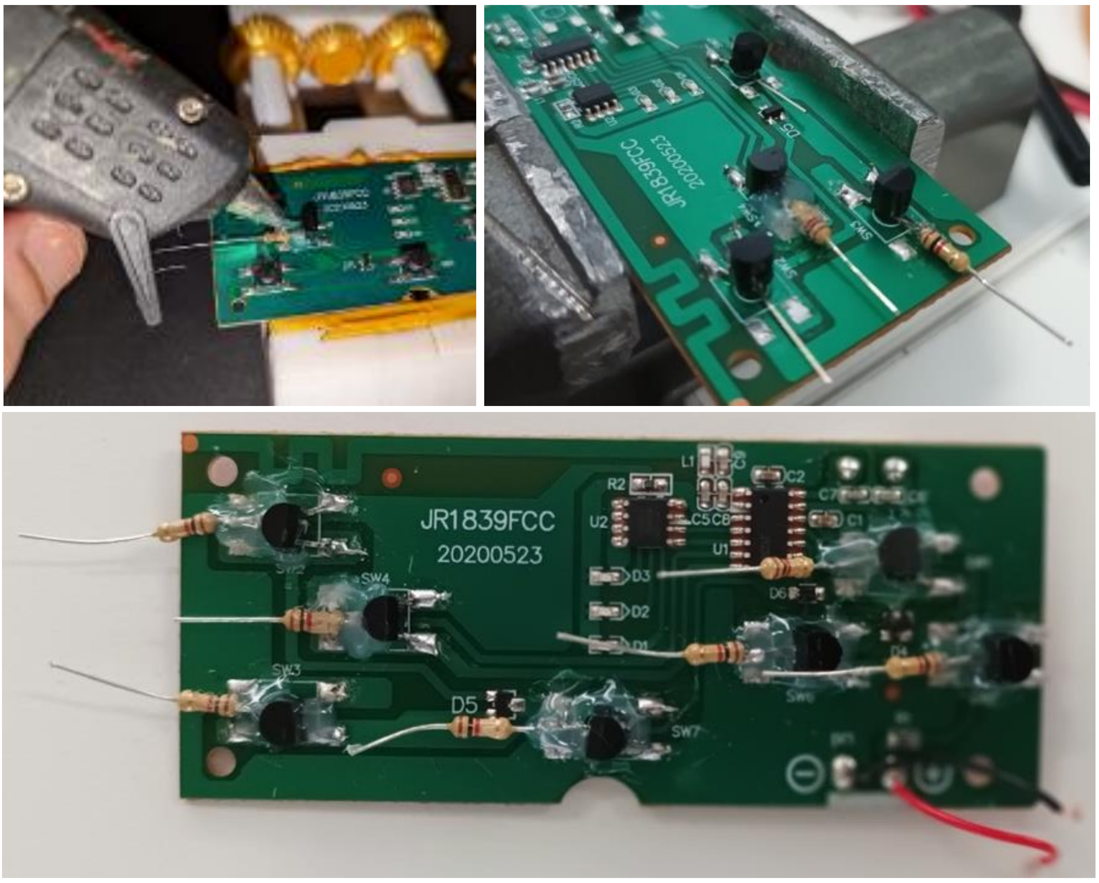




- On the base pin of the transistors, solder a 1KΩ resistor, making sure that the base pin does not touch the circuit board. Clip off the excess wire on the resistor if it is too long.



- Use hot glue to stabilize the transistors and prevent the resistor pins from touching the circuit board.

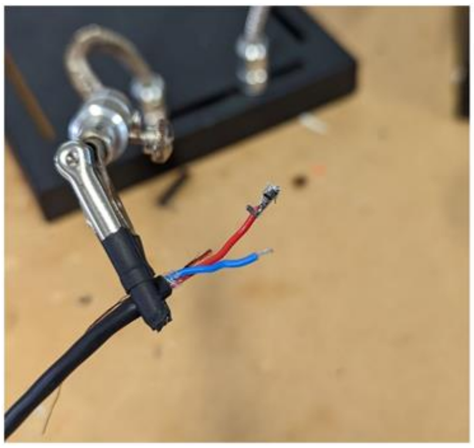
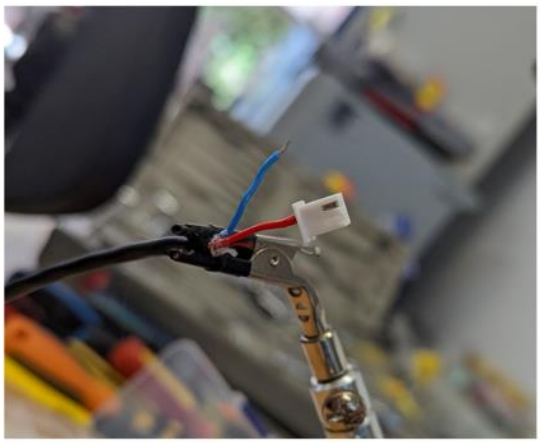


- On the circuit board, solder male/male jumper cables to the ground and positive terminals. Power can be supplied to the board using female/female jumper cables connected to the Raspberry Pi.
- Use female/female jumper cables to connect the resistors to different output pins on the Raspberry Pi. Each pin can be used to simulate a button press on the remote controller. Now the car can be controlled remotely by remotely logging into the Raspberry Pi via SSH.

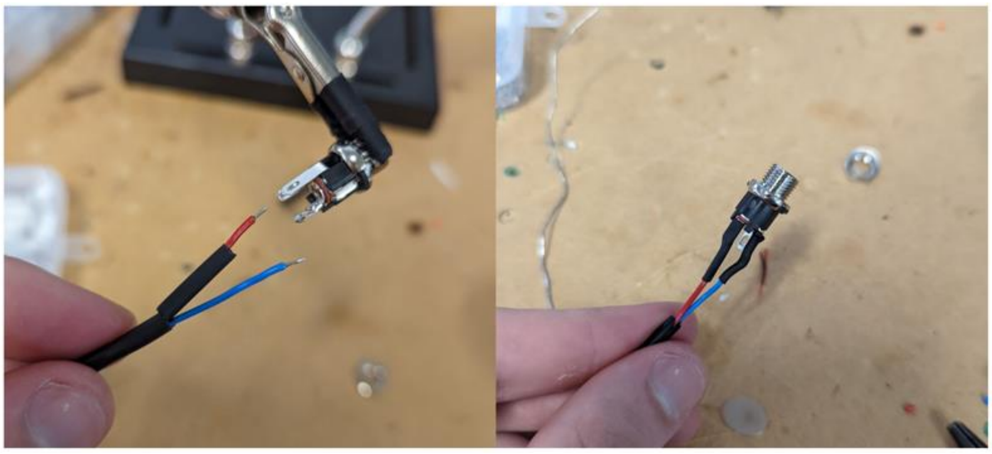
IMPORTANT Completing this step allows early testing of programs and making sure that the wiring is correct. Later you will want to solder male/female jumper cables to the resistors instead of using the temporary female/female cables.

Additional Soldering/Electronics

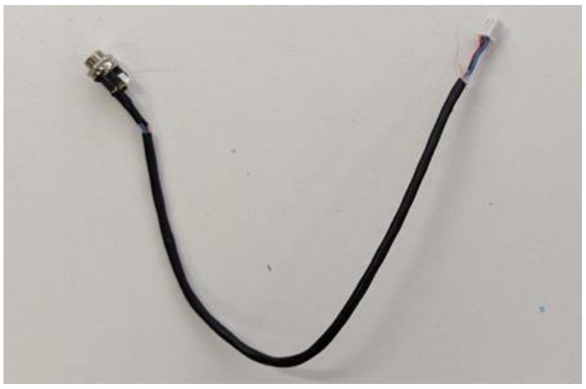
- To connect the car power to the control box we will make a couple of cables.
- Firstly, take dual core wire and strip a couple centimeters off each side like shown below. Then crimp a JST Crimp Pin to each wire. Insert the pins into the JST housing, make sure the positive side is connected as shown below.

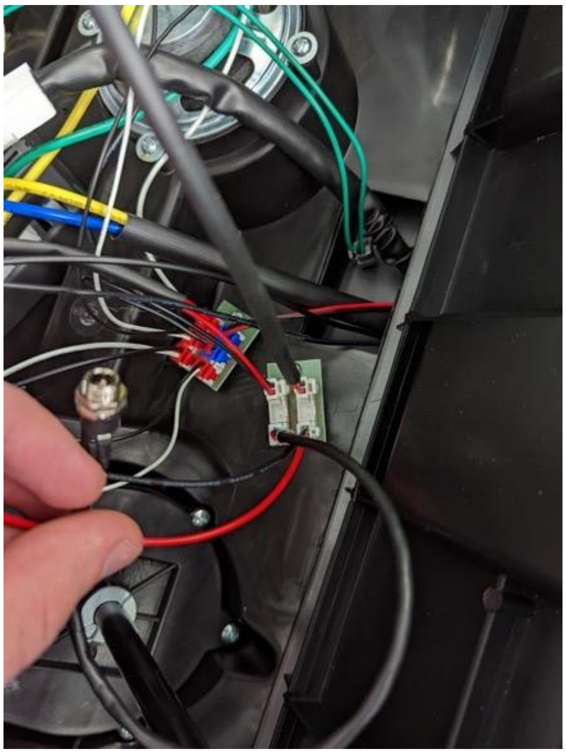
- On the other side of the cable, we are going to solder a 2.1mm chassis mount socket that will provide power to the outside of the car. Solder the positive wire to the centre pin of the connector and the negative wire to the outer pin. Use some heat shrink to cover the connections.



- The final cable is shown below.

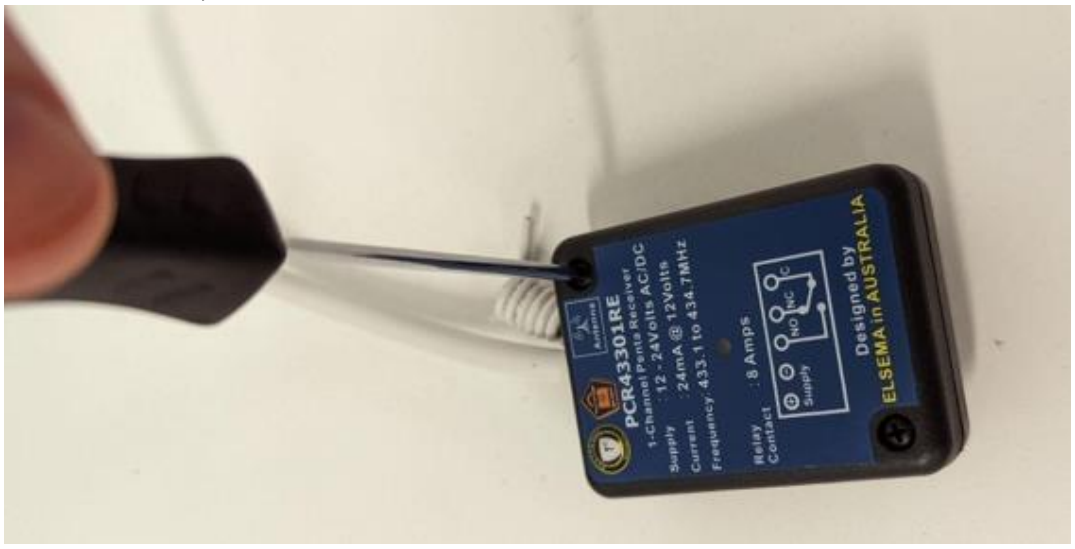


- Find the bus bar with JST sockets inside the hood of the car shown below, this is connected to the cars 24V battery. Plug in the JST connect of the cable you made and drill a hole in the foot well of the car for screwing in the 2.1mm socket. This will allow the 24V power to be accessed outside the hood of the car.

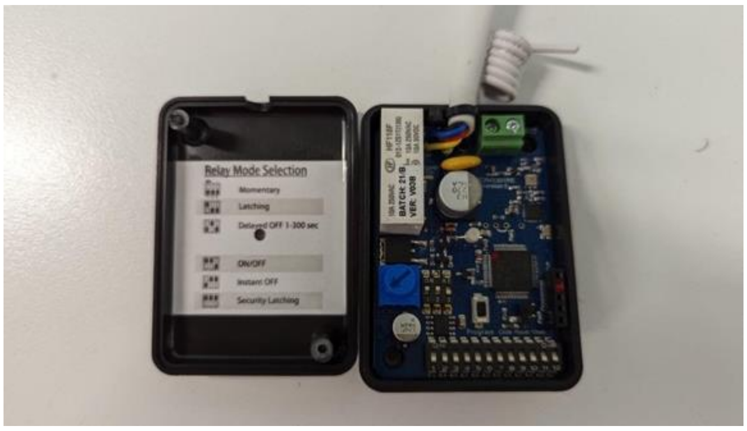


Setting up the E-Stop Receiver

- Unscrew the top off the PentaFob Receiver



- Follow the instructions inside to set the FOB into ON/OFF mode.

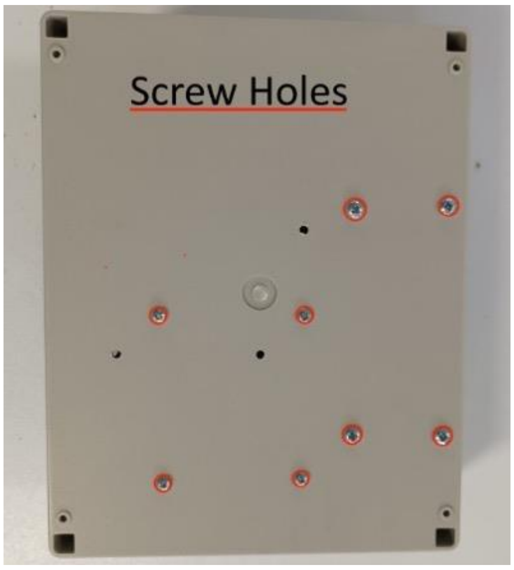


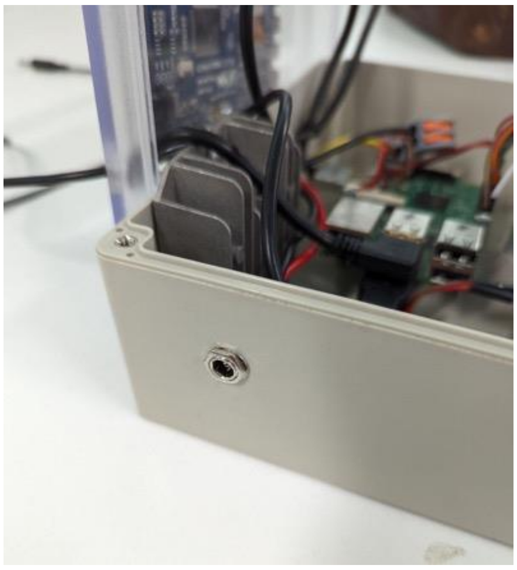
- When powered with 24V the FOB key will allow the car to be stopped wirelessly.

Housing the Electronics

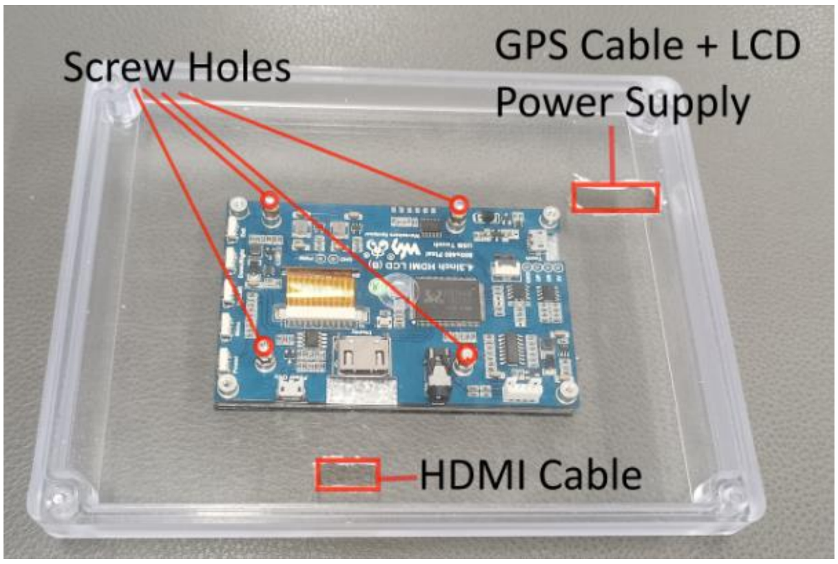
- Drill the appropriate holes into the electronics box base to secure both the Raspberry Pi and the controller board. There should be screw holes located on the four corners of both parts. Keep in mind that cables will need to be plugged into the Pi when choosing where to drill the holes.

- Drill a hole into the wall suitable for the 2.1mm power socket.

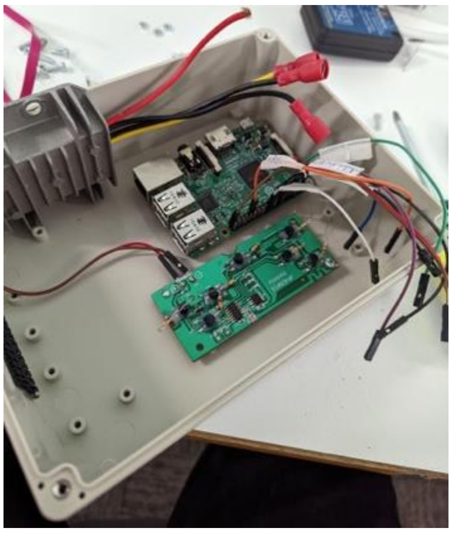




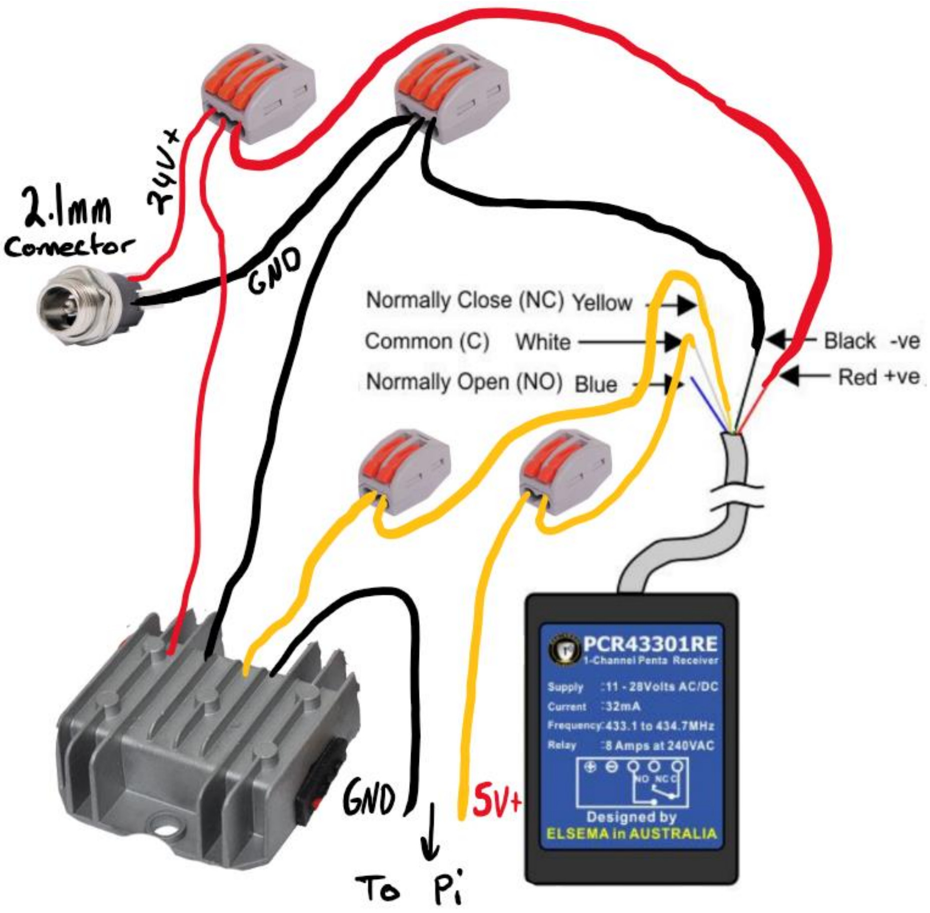
- Drill holes into the lid for the HDMI cable and the LCD power cable.



- Secure the Raspberry Pi and controller board using tapped spacers between them and the base for elevation. Reconnect all jumper cables between the Pi and the controller board.

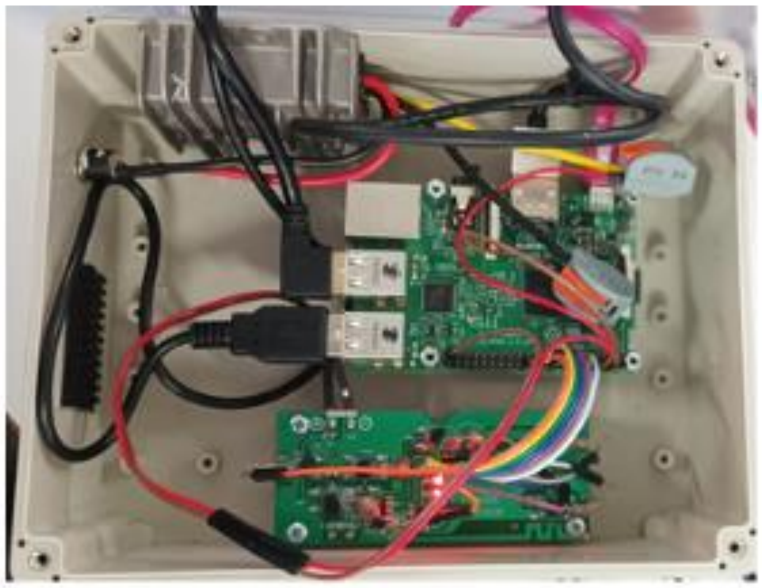


- Use Velcro or double-sided tape to secure the 24V-5V converter and E-Stop receiver to the sides of the enclosure.
- The wiring diagram for the power wiring is shown below, the 5V power from the regulator goes to the 5V pins on the Raspberry Pi board.



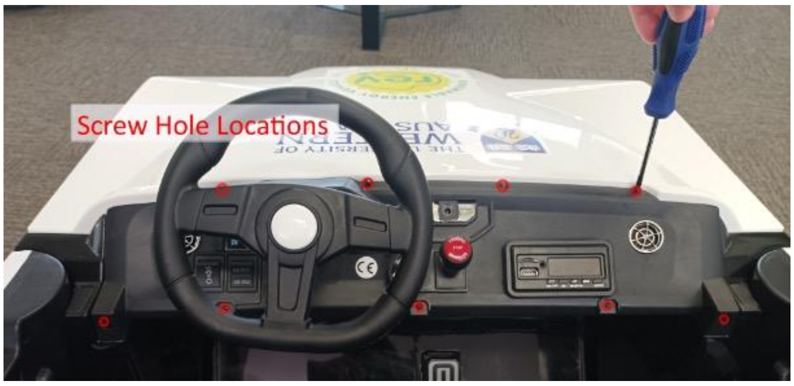
- Wire the 5V to the Pi's 5V and Gnd pins, wire the Pi's 3.3V output to the controller circuit board's positive input and connect the Gnd as well. Then wire up all the resistors to suitable GPIO pins on the Pi.

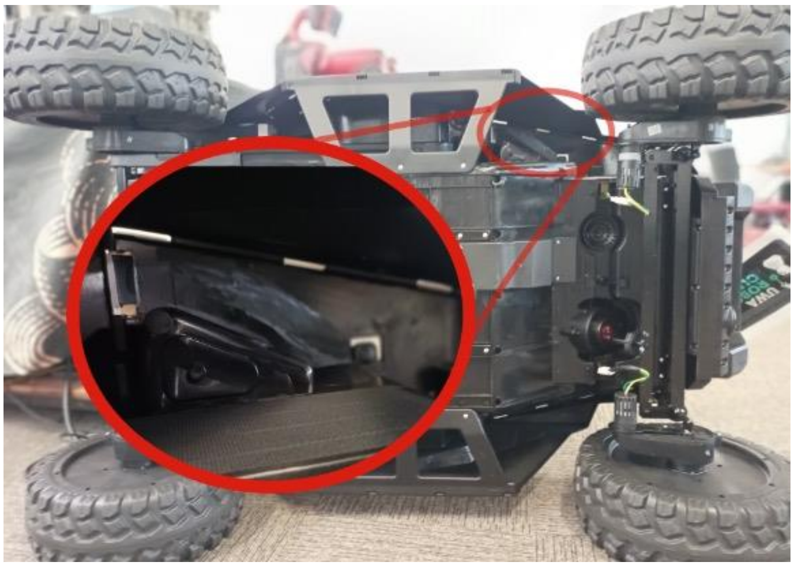
- Connect the GPS receiver, HDMI and USB for the LCD screen to the Pi, then close up the box.



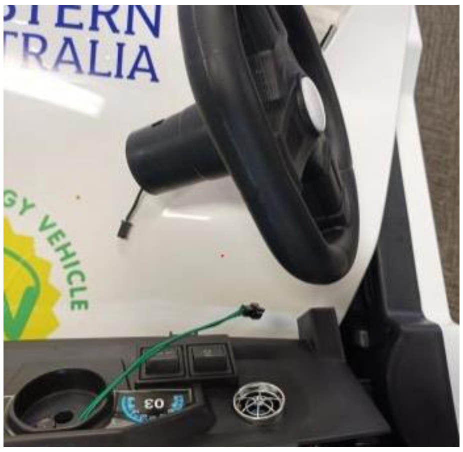
Final Modifications to the Car

- Unscrew the car's dashboard and pry out the clips on the hood to expose the internal wiring. To unclip the plastic hood, it may be easier to rotate the car onto its side.



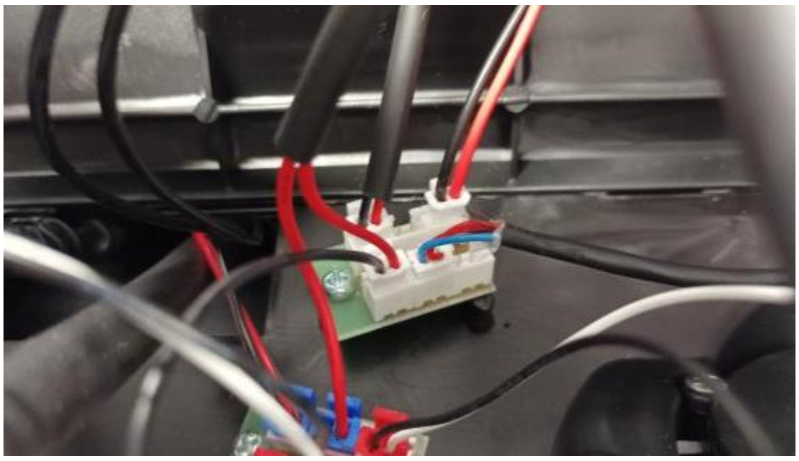


- Disconnect the cable leading to the steering wheel horn. Disconnect the speaker cables to prevent loud audio playing from the speakers.

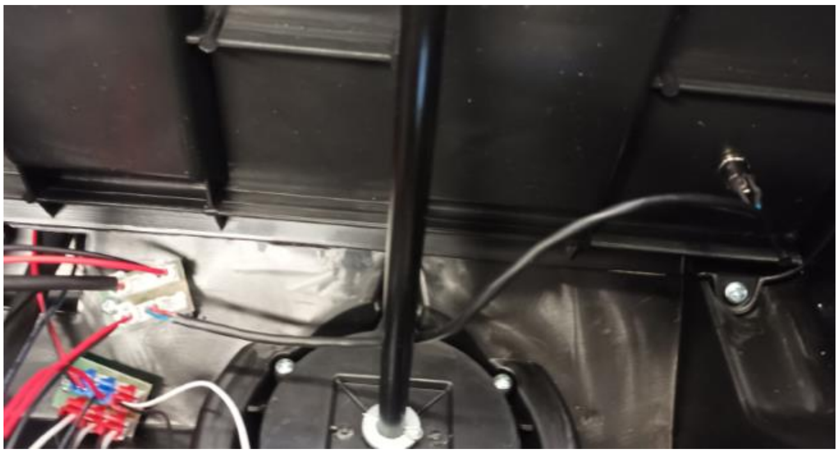


- You can now reach the interior wall of the car footrest. Drill a hole suitable for the 5.5mm socket such that it is accessible from the exterior side of the footrest.

- Locate the 24V JST connector from inside the car, which looks like this:



- Attach the modified JST cable and fix the soldered 2.1mm power socket through a hole in the footrest.

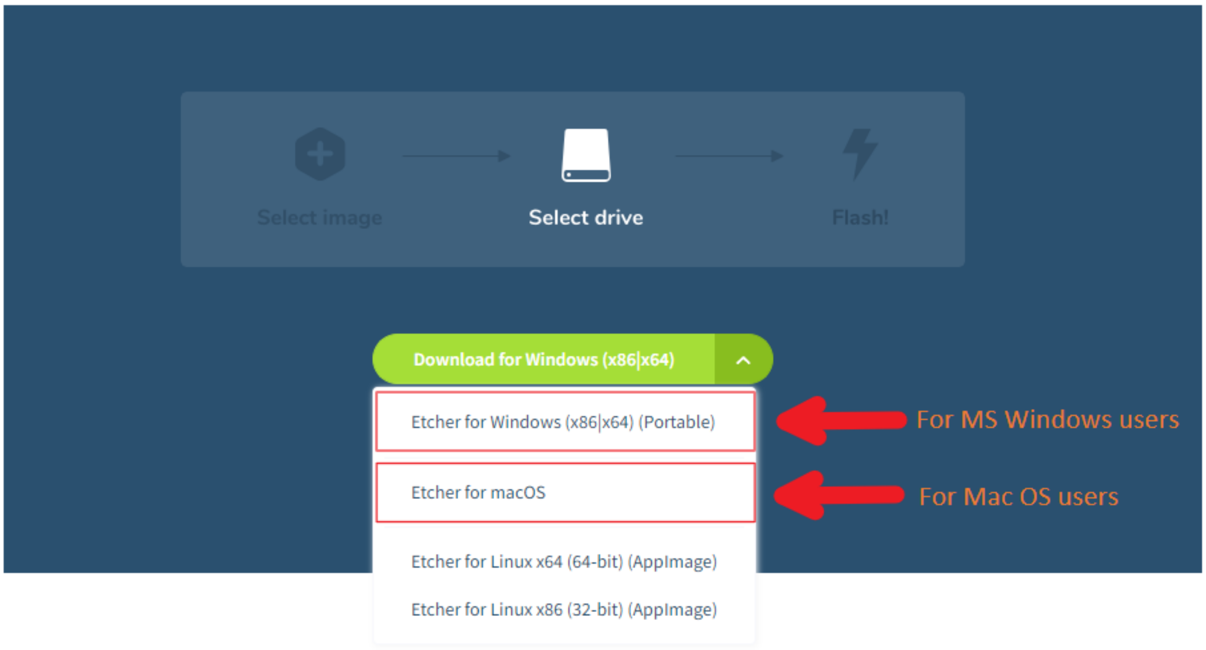


- From the inside of the car, you can now connect the electronics box and the power socket using the 2.1mm barrel plug cable.
- The GPS can be optionally secured to a more open area of the car to improve reception.
- The dashboard and hood can now be screwed back in.
- To explain this procedure: we are routing the 24V supply from the car battery to the power socket fixed into the footrest. This is connected to the 24-5V converter which provides the 5V required for the Raspberry Pi. The electronics box can now be left inside the car's footrest.

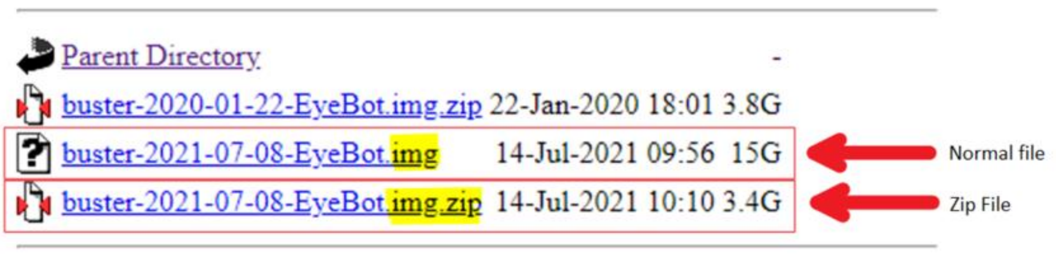
Setting up the Raspberry Pi

The Raspberry Pi needs to be set up properly before it can run the code to control the car.

- Install *balenaEtcher* on your computer, from the following link:
  https://www.balena.io/etcher/
  Download the appropriate version for your machine. This is used to etch operating system images onto an SD Card, which will be plugged into the Raspberry Pi so that it can run the operating system that we want.


Select image
Select drive
Flash!
Download for Windows (x86|x64)
Etcher for Windows (x86|x64) (Portable)
For MS Windows users
Etcher for macOS
For Mac OS users
Etcher for Linux x64 (64-bit) (AppImage)
Etcher for Linux x86 (32-bit) (AppImage)

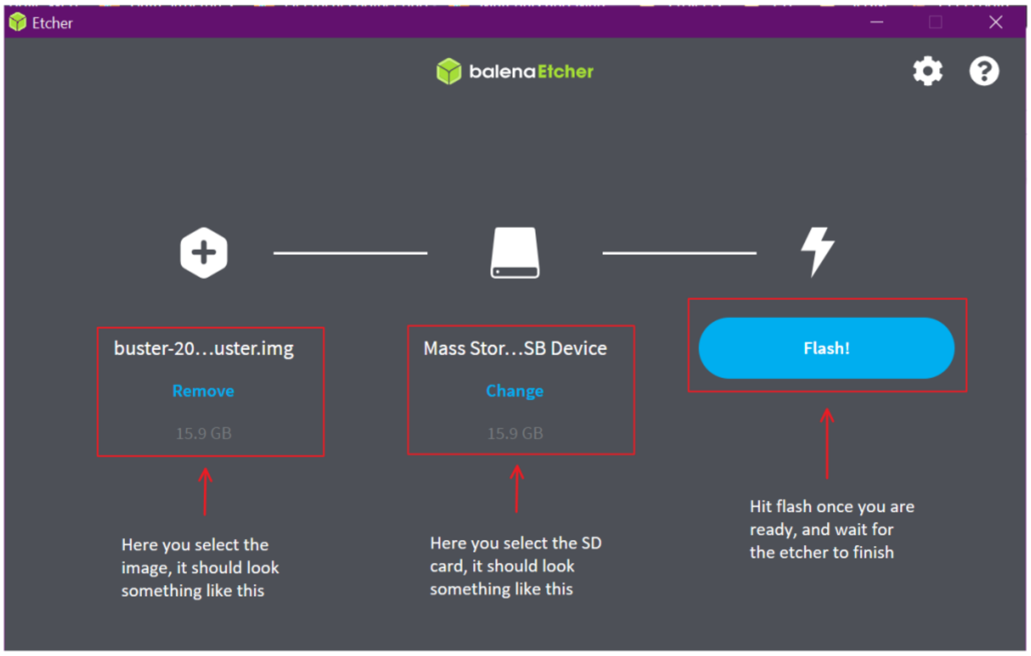
- The Raspberry Pi will run on the Eyebot OS version 7, the download link and the documentation link as following (respectively):
  https://robotics.ee.uwa.edu.au/rasp/images-pi4_Buster/
  https://robotics.ee.uwa.edu.au/eyebot//
  Download the Eyebot img.



Parent Directory		-
buster-2020-01-22-EyeBot.img.zip	22-Jan-2020 18:01	3.8G
buster-2021-07-08-EyeBot.img	14-Jul-2021 09:56	15G
buster-2021-07-08-EyeBot.img.zip	14-Jul-2021 10:10	3.4G

If your laptop is quite fast, it can be faster to download the zip, and then unzip it. If it is not, or you are unsure, then simply download the .img file and do not worry about any additional steps.

- Insert a blank SD card into your computer and open the *balenaEtcher*.
  Click on flash from file and select the downloaded Eyebot image.
  Click on select target, and select the blank SD card you previously inserted.
  Hit the flash button and wait for the process to complete.



Here you select the image, it should look something like this

Here you select the SD card, it should look something like this

Hit flash once you are ready, and wait for the etcher to finish

IMPORTANT. Once the flashing is complete, your computer may not recognize the SD card anymore, or send an error telling you the adaptor is empty. This is perfectly normal, the SD card is no longer formatted in a way that your computer is able to understand, so the computer is unable to read it, but this is not an issue. Your computer may ask if it can reformat the SD card to something it is able to understand, do not allow it to do this or you will not be able to use it anymore, and will have to flash the image again.

- Insert the SD card into the Raspberry Pi and power up the Raspberry Pi. If you have not set up the hardware on the car properly yet then you can simply plug a micro USB cable into the Pi and plug the other end into the computer, and this will supply it with power.

- Connect a touchscreen via the HDMI and USB port, or connect a monitor and a mouse via HDMI and USB. You may choose whatever you prefer, but you need a way to display the Raspberry Pi display and give it inputs. From this point, all you wish to do is connect the Raspberry Pi to a network, so that you can connect to it remotely from your laptop. Just note that the Raspberry Pi and the computer have to be connected to the same network.

- Determine what the IP address of the Raspberry Pi is. This is typically a number with dots in it that will look something like this, 192.168.x.xxx. The best way to find the IP is to open your Router homepage and find the section that lists all the connected devices. From here you should see something called Pi, this is the Raspberry Pi and the router should list its IP. Note this number down.

- Install Putty. This is a client which will allow you to connect to the Raspberry Pi from your own computer. It will only allow you to access the terminal from the Raspberry Pi however, but this is sufficient for our purpose and is much faster than using the Windows Remote Desktop Connection. Putty can be installed from here: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

It is recommended you download the 64-bit x86 file. Execute the installer and follow the on-screen instructions to install putty.

- Open Putty. Under the Host Name section, write in the IP address of the Raspberry Pi. Leave everything else as is and hit connect. The username should be “**pi**” and the password “**rasp**”.

Programming

This is the demo code that can be run on the Pi to control the car. It essentially pushes the buttons the same way you would on the remote, but instead of pushing with fingers, it pushes the buttons by sending them electrical signals.
This code is written in C.

```
#include "eyebot.h"

void turnLeft(int);
void turnRight(int);
void moveForward(int);
void moveBackward(int);
void pairing();
void pushButton(int, int);

#define RIGHT 6 //physical pin 22, wiring pin 6
#define LEFT 4 //physical pin 16, wiring pin 4
#define PARK 21 //physical pin 29, wiring pin 21
#define CON 3 //physical pin 15, wp 3
#define REVERSE 0 //physical 11, wp 0
#define FORWARD 26 //physical 32, wiring pin 26
#define SPEED 2 //physical pin 13, wiring pin 2

#define SECOND 1000000

int main() {
        wiringPiSetup();
        pinMode(RIGHT, OUTPUT); //Turn right
        pinMode(LEFT, OUTPUT); //Turn left
        pinMode(PARK, OUTPUT); //Park
        pinMode(CON, OUTPUT); //Pairing/remote on
        pinMode(REVERSE, OUTPUT); //Down
        pinMode(FORWARD, OUTPUT); //Up
        pinMode(SPEED, OUTPUT); //Speed

        //This is a demo of the different ways to control the car
```

```
 //This pairing process only has to be done once and
 //and then the car will always remember the remote
 //Pushing the pairing button for 4 seconds
 pushButton(CON, 4);
 //Waiting for 3 seconds
 usleep(3*SECOND);

 //Pushing reverse for three seconds.
 pushButton(REVERSE, 3);

 //Pushing forwards for three seconds.
 pushButton(FORWARD, 3);

 //Pushing left for three seconds.
 pushButton(LEFT, 3);

 //Pushing right for three seconds.
 pushButton(RIGHT, 3);

 //Pushing park toggles the parking mode on.
 //Pushing park for 1 second
 pushButton(PARK, 1);
 //The car is now parked and will not move.

 //Pushing park for 1 second.
 pushButton(PARK, 1);
 //The car is no longer parked and will now move.

//The speedmode button cycles through each speedmode.
//Each press increases the speed by 1, from 1 - 3.
//Pressing it while it is at 3 simply brings it back
//down to 1.

//Push speed for one second, assuming we are at 1
pushButton(SPEED, 1);
//We are now in speedmode 2
//Push speed for one second
pushButton(SPEED, 1);
//We are now in speedmode 3

//Push speed for one second
pushMode(SPEED, 1);
//We are now back in speedmode 1

}
```

```
//Pushes the given button for the given duration (in seconds)
void pushButton(int inPin, int duration) {
       digitalWrite(inPin, HIGH);
       usleep(duration*SECOND);
       digitalWrite(inPin, LOW);
}
```

More info on pins for remote control via software:

https://robotics.ee.uwa.edu.au/rasp/
https://pinout.xyz/pinout/5v_power

Copy this program as file “ride.c” to directory “/home/pi/usr”, e.g. using the Filezilla tool. Then open a terminal window on the Raspberry Pi and enter the following commands (from the command line) to go to the right directory, edit the source code, and compile the program:

```
cd /home/pi/usr
nano ride.c
gccarm -g -o ride.x ride.c
```

To run the compiled program, type:.

```
./ride.x
```

Contact

UWA Robotics Club

https://uwarobotics.com.au
club@uwarobotics.com.au

Jake Lorkin
Tiziano Wehrli
Won Chen Qin
Agnibho Gangopadhyay

UWA Robotics & Automation Research Lab

https://roblab.org *or* https://robotics.ee.uwa.edu.au
tb@ee.uwa.edu.au

Professor Thomas Bräunl

CLOUGH ENGINEERING STUDENT CENTRE
ELECTRIC VEHICLE EV CHARGING STATION