Autonomous Ride-on Car

Ver. 1.3, May 2026

A STEM Project for High Schools

roblab.org/rideon

UWA Robotics & Automation Lab
Shaf Kashif, directed by Thomas Bräunl

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
CONTROL BOX COMPONENTS						
H0330	186Lx146Wx75Hmm IP65 Sealed ABS Enclosure	1	$33.25	$33.25	Altronics	Enclosure for electronics
AA1017A	FOB43301L PentaFOB Single Large Button 433Mhz Remote	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
T-Display-S3	LILYGO T-Display-S3 ESP32-S3 Development Board	1	$23.00	$23.00	Amazon	Controller
NEO-6M	NEO-6M GPS Breakout Module with Antenna	1	$17.95	$17.95	Altronics	GPS
DF-SEN0253	Gravity BNO055+BMP280 intelligent 10DOF AHRS IMU	1	$21.50	$21.50	Core Electronics	IMU
DFR0379	20W Adjustable DC-DC Buck Converter with Digital Display	1	$8.55	$8.55	Core Electronics	Step down voltage
GP2Y0A21YK0F	GP2Y0A21YK0F Analog Distance Sensor 10-80cm	3	$14.41	$43.23	DigiKey	Distance Sensor
P9407	7 Pin 5A Locking Female Line IP67 Waterproof Socket	1	$26.50	$26.50	Altronics	Mounting Electronics
P9467A	7 Pin 5A Locking Male Chassis IP67 Waterproof Plug	1	$18.50	$18.50	Altronics	Mounting Electronics
JST SH 4-pin	STEMMA QT / Qwiic JST SH 4-pin to Premium Male	1	$0.95	$0.95		Wiring
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						
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
W2712	9 Core Shielded Data Cable	3	$2.25	$6.75	Altronics	Electronics Cable
		TOTAL		**$1,039.23**		

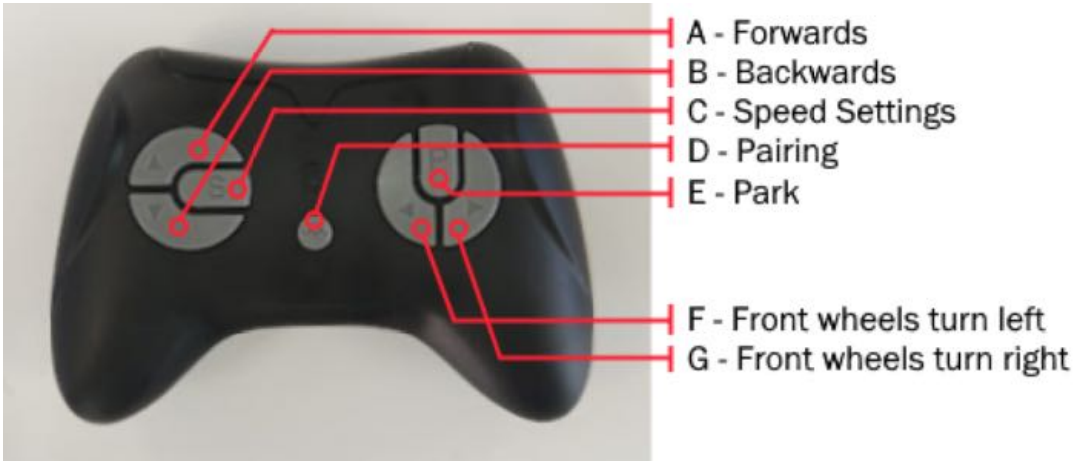
Link
https://www.artintoys.com.au/product/400-w-24-v-beach-buggy-dune-kids-ride-on-car-white/
Given to us free of charge from ArtInToys
https://www.altronics.com.au/p/h0330-ritec-186lx146wx75hmm-ip65-sealed-abs-enclosure/
https://www.altronics.com.au/p/aa1017a-elsema-pentafob-fob43301l-433mhz-single-large-button-remote-control/
https://www.altronics.com.au/p/a1025a-elsema-penta-pcr43301re-433mhz-1-channel-receiver/
https://www.amazon.com.au/LILYGO-T-Display-S3-ESP32-S3-Development-Soldering/dp/B09J112YR7?th=1
https://www.altronics.com.au/p/z6318-neo-6m-gps-breakout-module-with-antenna/
https://core-electronics.com.au/10-dof-mems-imu-sensor.html?
https://core-electronics.com.au/20w-adjustable-dc-dc-buck-converter-with-digital-display.html?
https://www.digikey.com.au/en/products/detail/sharp-socle-technology/GP2Y0A21YK0F/720159?
https://www.altronics.com.au/p/p9487-amphenol-ltw-7-pin-5a-locking-female-line-ip67-waterproof-socket/
https://www.altronics.com.au/p/p9467a-amphenol-ltw-7-pin-5a-locking-male-chassis-ip67-waterproof-plug/
https://www.adafruit.com/product/4209
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/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/
www.altronics.com.au/p/w2712-9-core-shielded-data-cable/

TOOLS + CONSUMABLES NEEDED	WHERE TO GET	COST
Soldering Iron	Altronics	$20.00
Leaded Solder	Altronics	$2.70
Hot Glue Gun	Altronics	$14.50
Hot Glue Sticks	Altronics	$5.65
Drill	Bunnings	$50.00
Screw Drivers	Altronics/Bunnings	$15.00
M3 Screws	Bunnings	$3.00
M3 Metal Spacers	JayCar/Bunnings	$14.00
Heatshrink	Altronics	$10.00
	TOTAL:	**$134.85**

Hardware Description

Remote Controller

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

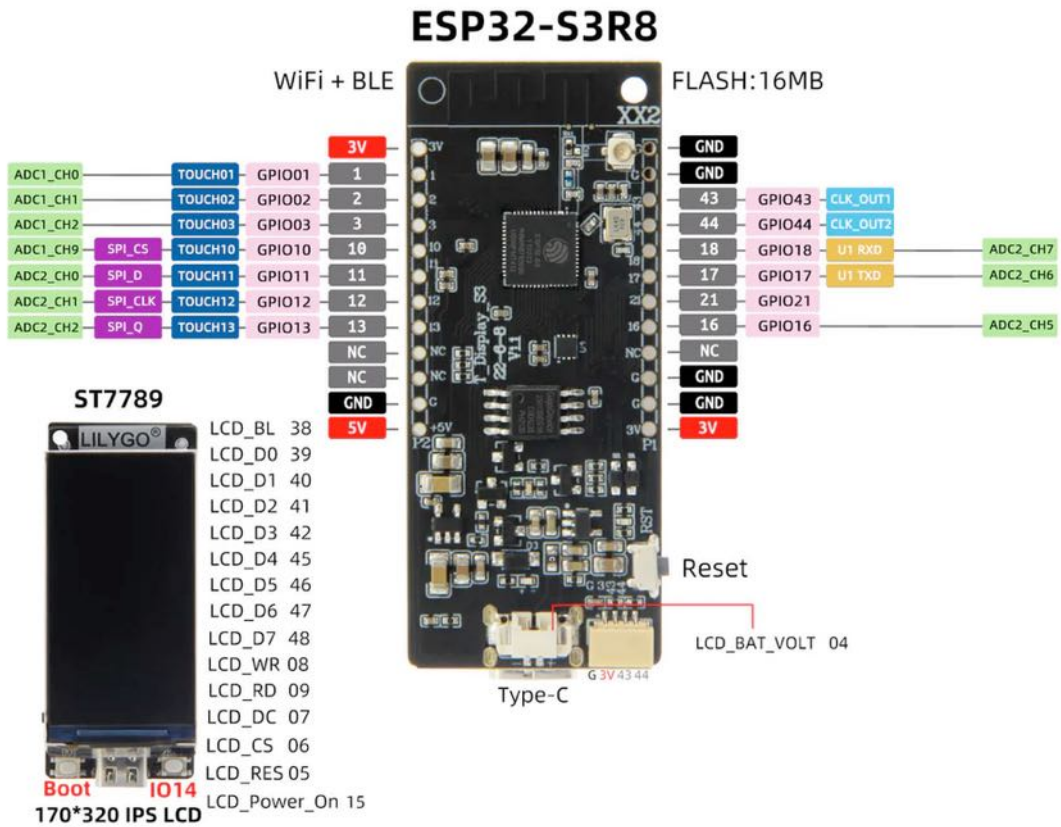


- NOTE: There are three LEDs on the controller which indicate information about the car.
- When changing the speed setting, 1 lit LED will indicate low speed, up to a maximum of 3 lit LEDs to indicate high speed.
- NOTE: To pair the remote with the car, the pairing button must be held down for 3 seconds, 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 ESP32’s 3v3 pin, which supplies 3.3V.

ESP32 LILYGO T-Displasy-S3

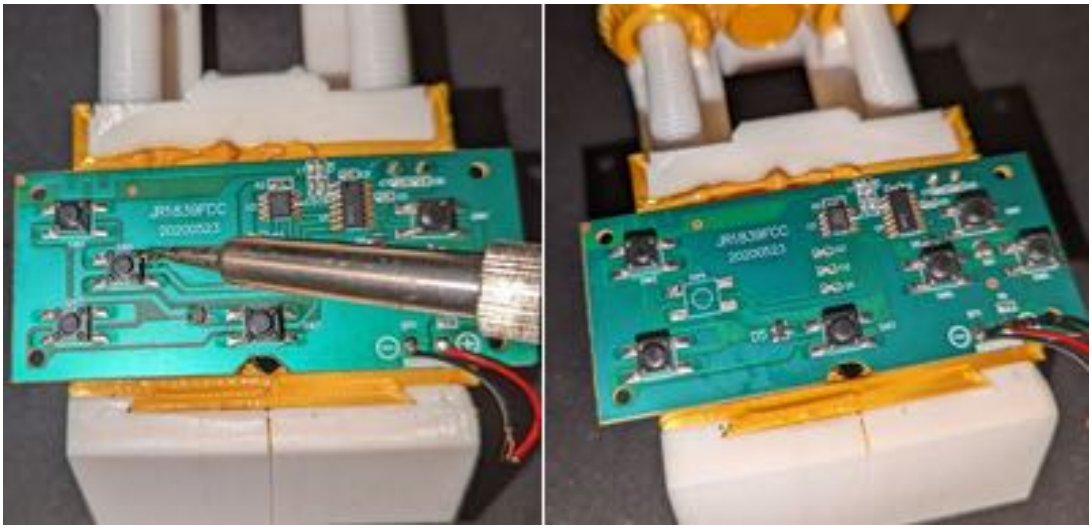
- This is a useful website to determine which output pins are suitable on the ESP32: https://github.com/Xinyuan-LilyGO/T-Display-S3
- Using GPIO pins to control the remote is recommended.
- NOTE: Each GPIO Pin may correspond to the same Wiring Pin number. The Wiring Pin number is what is used in the code, when any digitalWrite calls are made.
- NOTE: GPIO 1 – 13 are the only ADC meaning any analogue sensors must be connected to those.



Implementation Steps

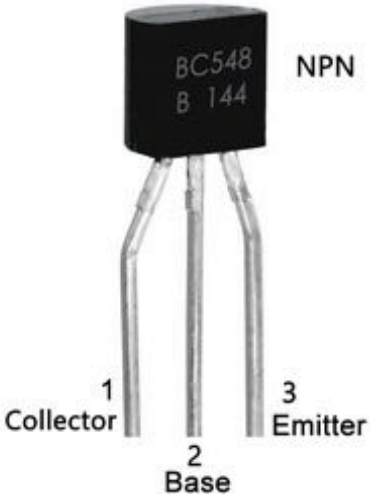
Remote Control Mod

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

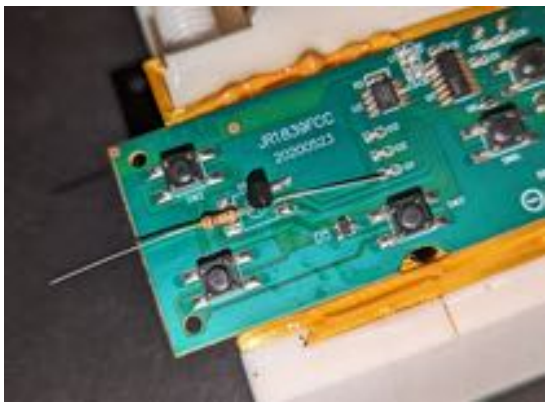


- Solder BC548 transistors in place of the buttons. Connect the emitter pin to a ground plate, and the collector pin to live.

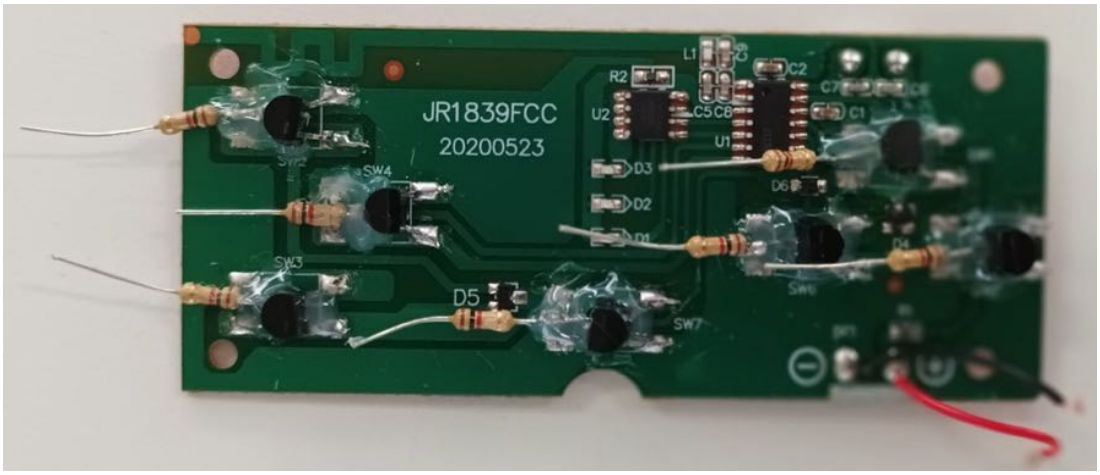
  NOTE: The light green sections of the circuit board are electrically connected. This can be used to find the ground side of the push buttons. The ground plates will be electrically connected to the section with a negative ground symbol. Similarly, the live plates are electrically connected to where the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



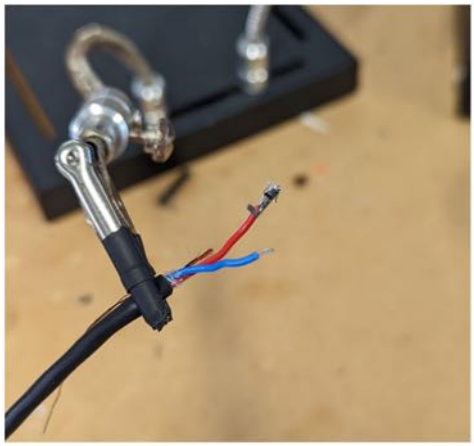
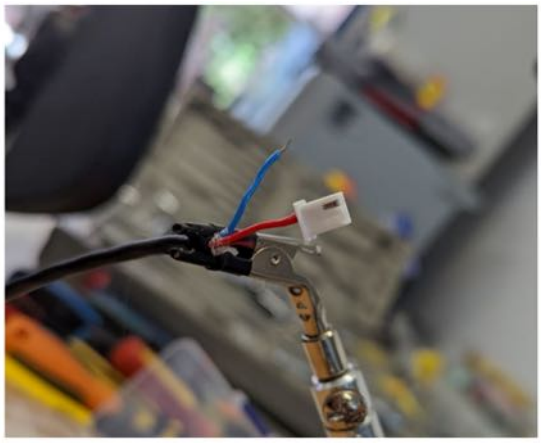
- Use hot glue to stabilize the transistors and prevent the resistor pins from touching the circuit board.



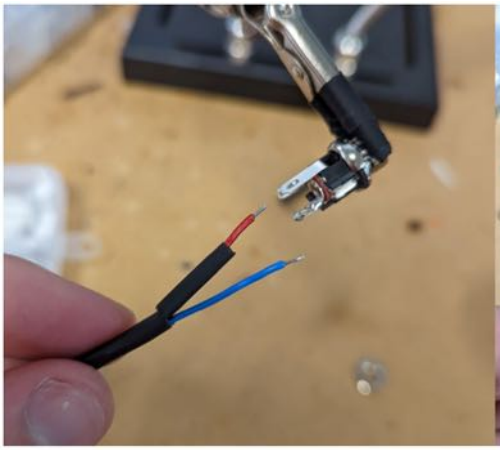
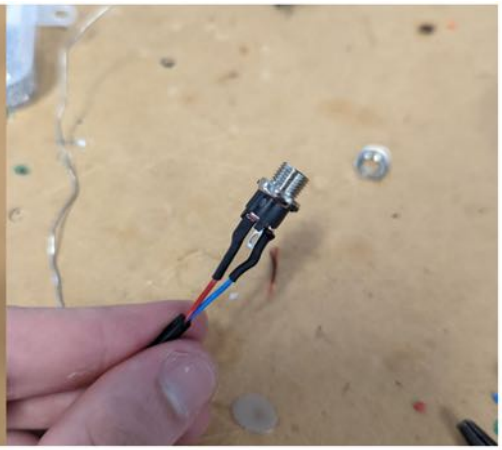
- On the circuit board, solder male/male jumper cables to the ground and positive terminals. Power can be supplied to the board using female/female jumper cables connected to the ESP32.
- Use female/female jumper cables to connect the resistors to different output pins on the ESP32. See the provided Table to see where each pin should be connected. Each pin can be used to simulate a button press on the remote controller. Now the car can be controlled remotely by uploading code to the ESP32.
  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

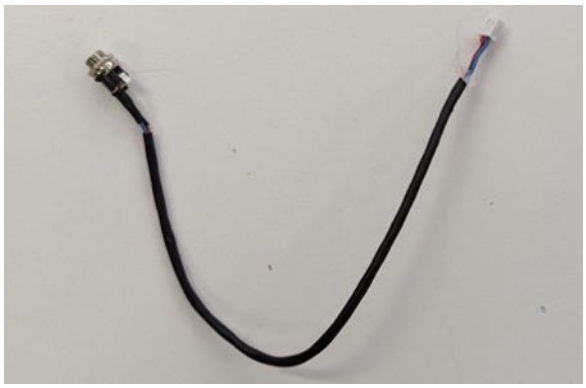
- To connect the car power to the control box we will make a couple of cables.
- Firstly, take dual core wire and strip a couple centimeters off each side like shown below. Then crimp a JST Crimp Pin to each wire. Insert the pins into the JST housing, make sure the positive side is connected as shown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

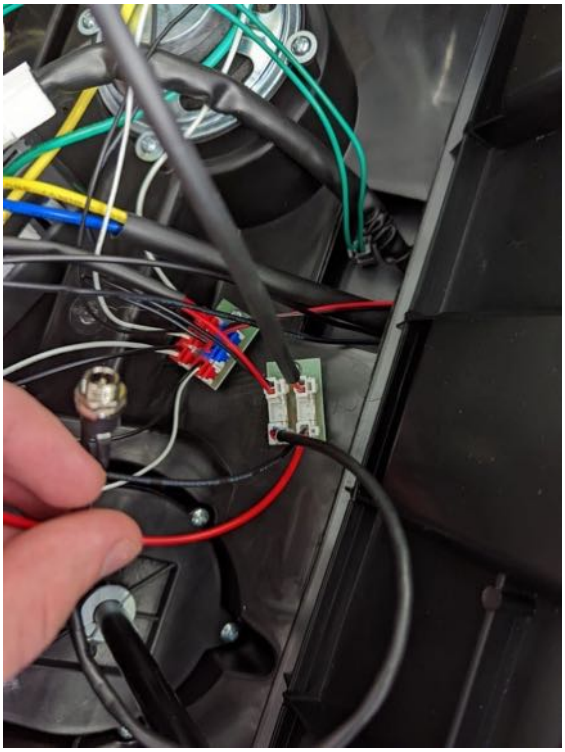



- The final cable is shown below.



- 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

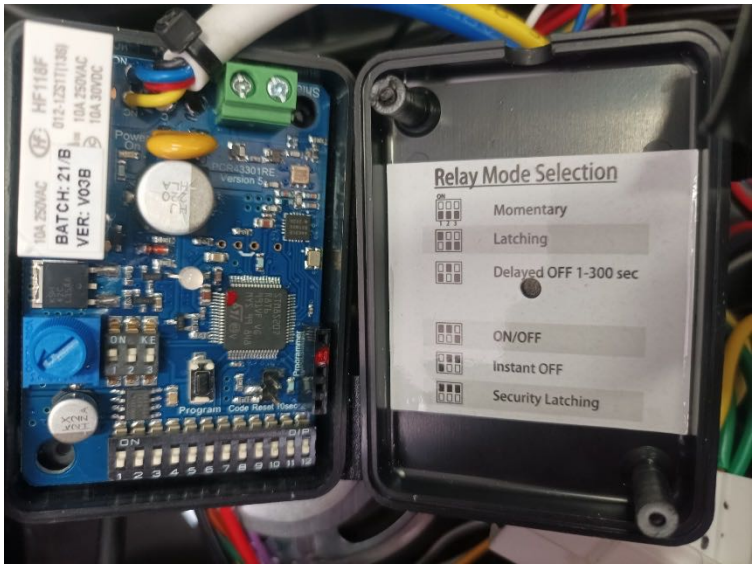
a hole in the foot well of the car for screwing in the 2.1mm socket. This will allow the 24V power to be accessed outside the hood of the car.



Setting up the E-Stop Receiver

- Unscrew the top off the PentaFob Receiver

- Follow the instructions inside to set the FOB into LATCHING mode.



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

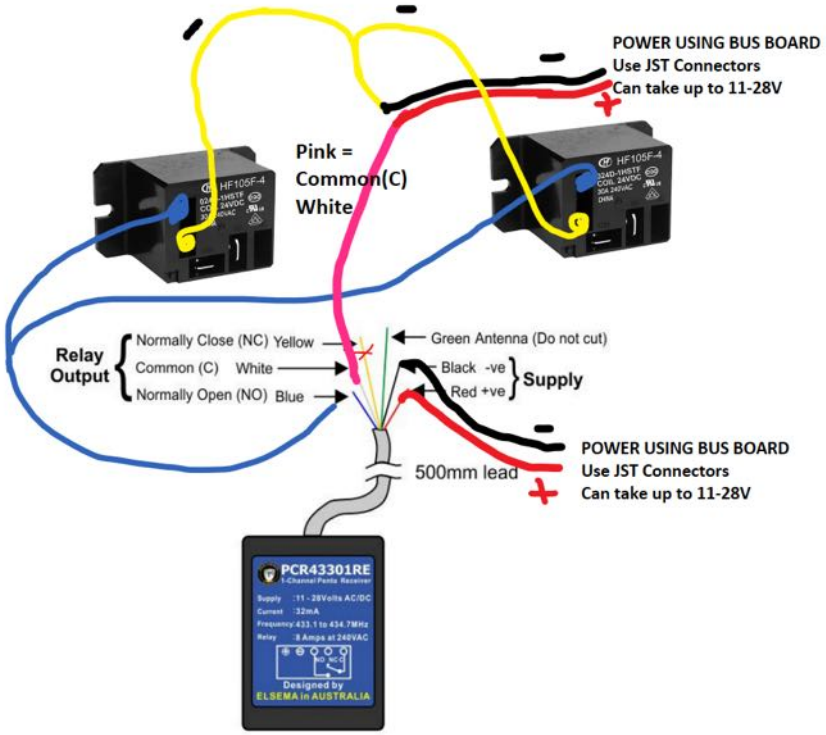
Wiring up the Emergency Stop:

Parts you need:

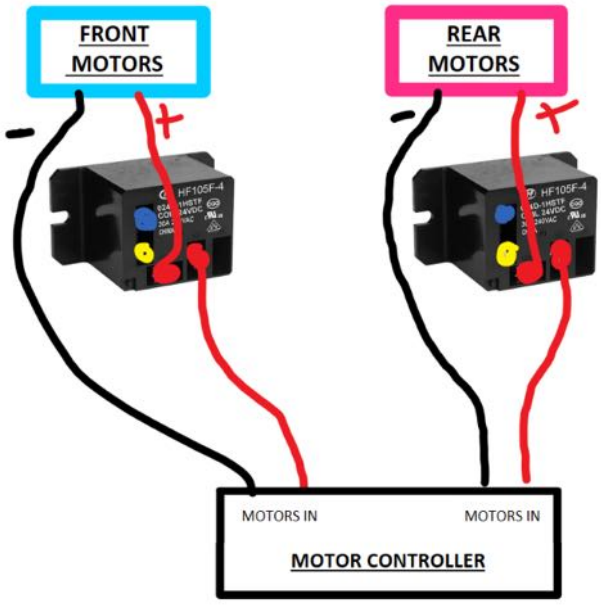
- E-Stop and Key fob (look at the BOM)
- Two Relays (look at the BOM)
- Car's Internal Motor Controller
- Cables and suitable crimps/connectors.

Step 1: Open and set the E-Stop to the correct mode, flick the switches according to the diagram to **LATCHING** mode. Find the Car's motor cables and motor controller and disconnect the motors from the controller.

Step 3: Wiring up the E-Stop to the power and Relays. (One relay for front two motors, and another for rear motors). Power for the E-Stop is from the car's bus board. Use crimp connectors and JST connectors to make these connections. [Check diagrams below and the pictures provided]



Step 4: Wiring up the Relays to the motor controller and motors. When wiring up the motors to the relay, please ensure that you use the cables that are suitable for the current rating of the motors. The relays act as a switch triggered by the E-stop, and they help stop the motors in case of an emergency. [Check diagrams below and the pictures provided]



Step 5: Testing and Usage.

You will need the key fob to activate the E-Stop.

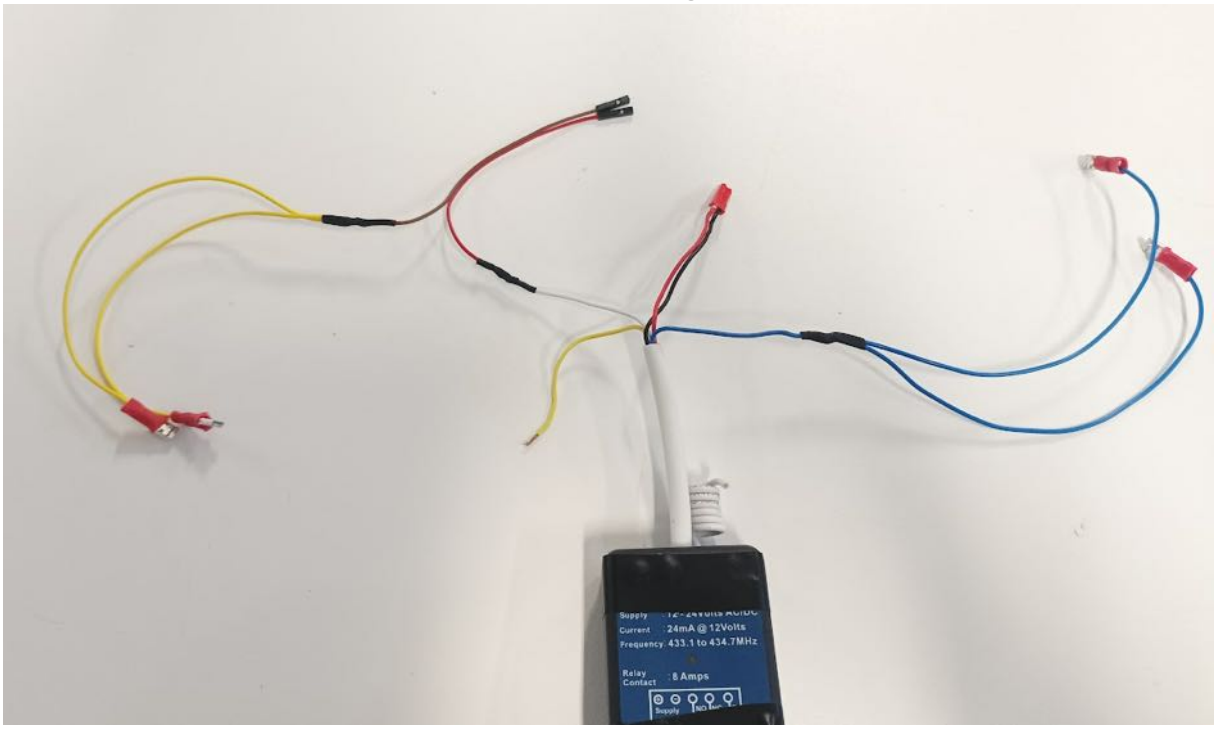
When the Car starts, you will have to press the fob to let the relays allow current flow to the motors. You should be able to hear an audible click which indicate that the relays are switching and the red light on the E-Stop. After the key press, the car should be able to move.

If an emergency arises while in operation, please press the fob key to activate the E-stop and trigger the relays which cut the power to the motors.

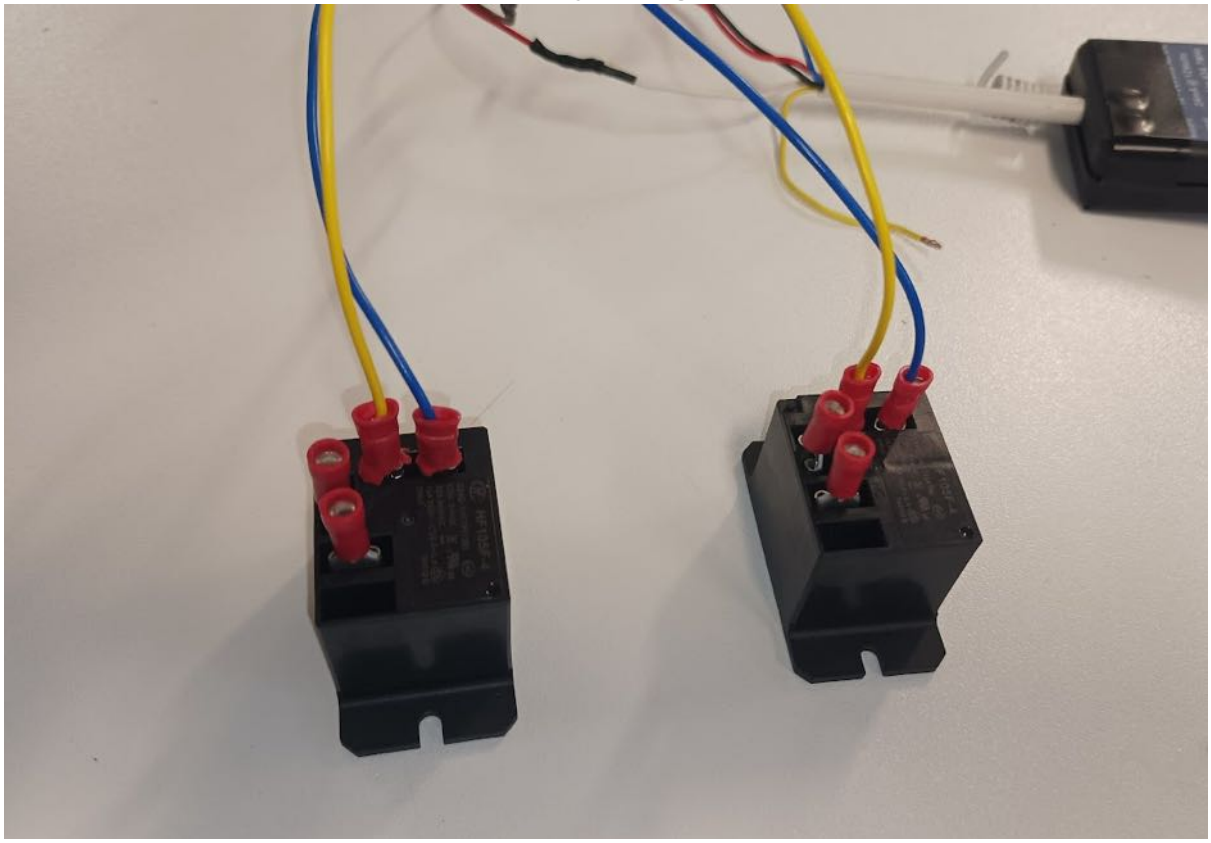
PLEASE NOTE: With this configuration, the E-Stop only cuts power to the motors. The remaining electronic hardware on the car is still active and powered.

Photos:

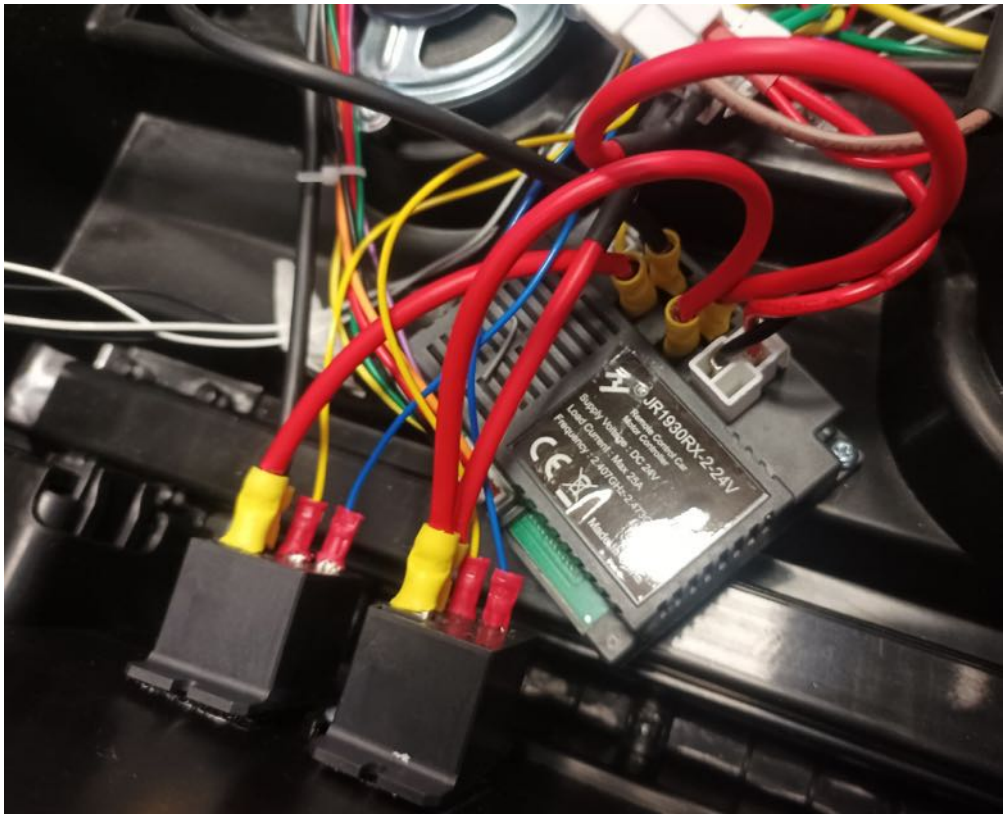
E-Stop Wiring:



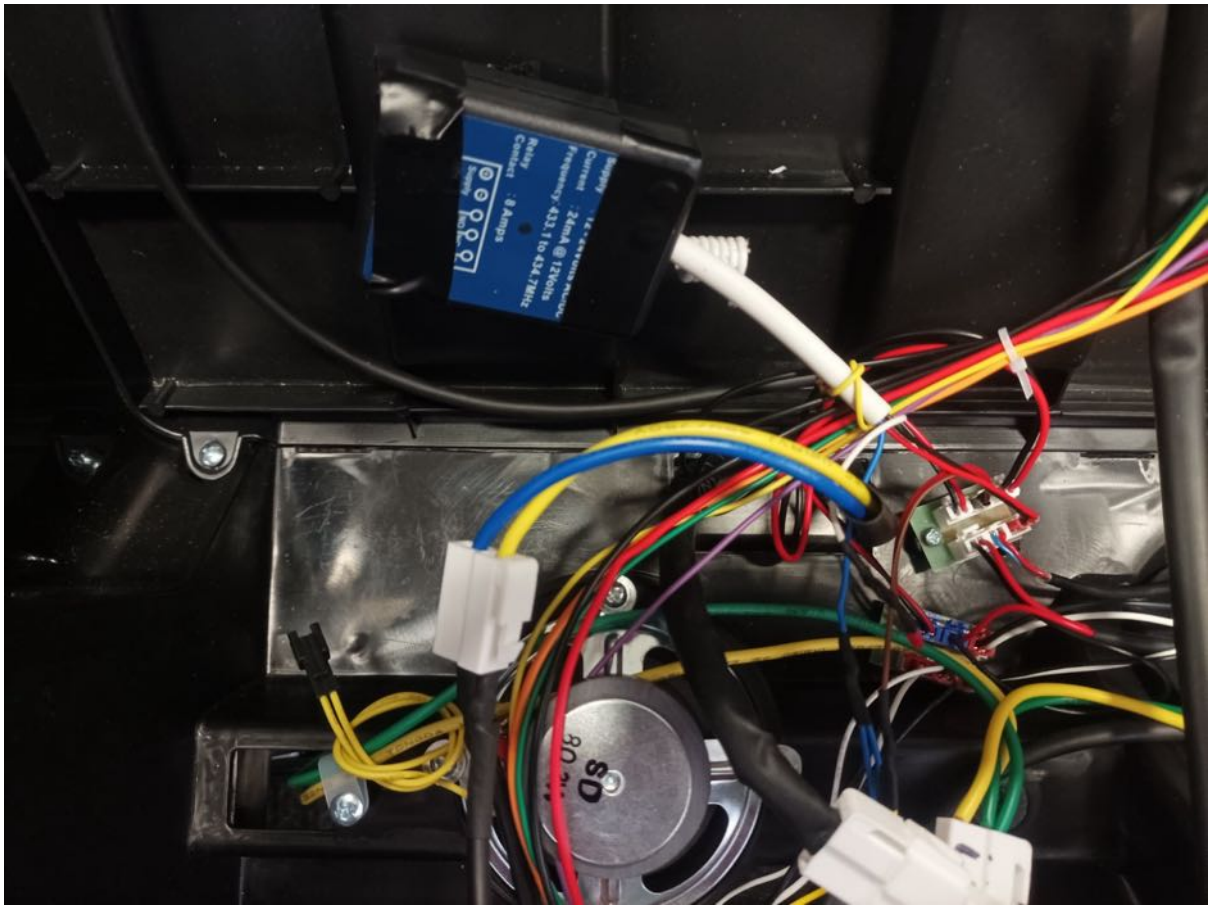
Relay Wiring:



Relay Wiring to Motor Controller and Mounting of Relays:



E-Stop Mounting:



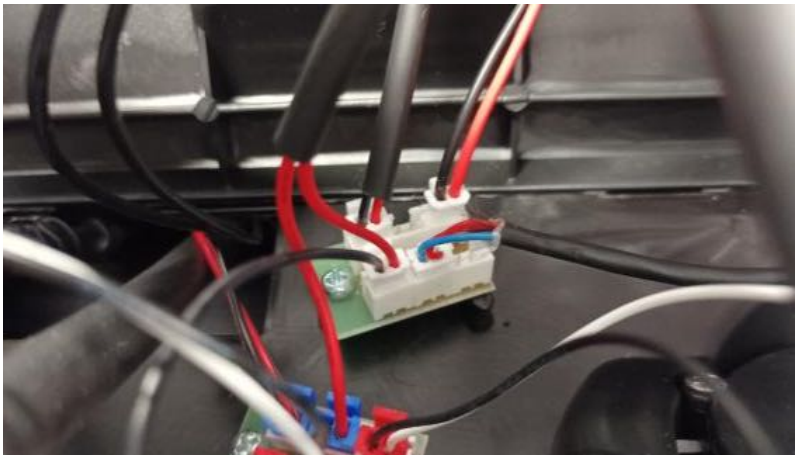
Final Modifications to the Car

- 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:

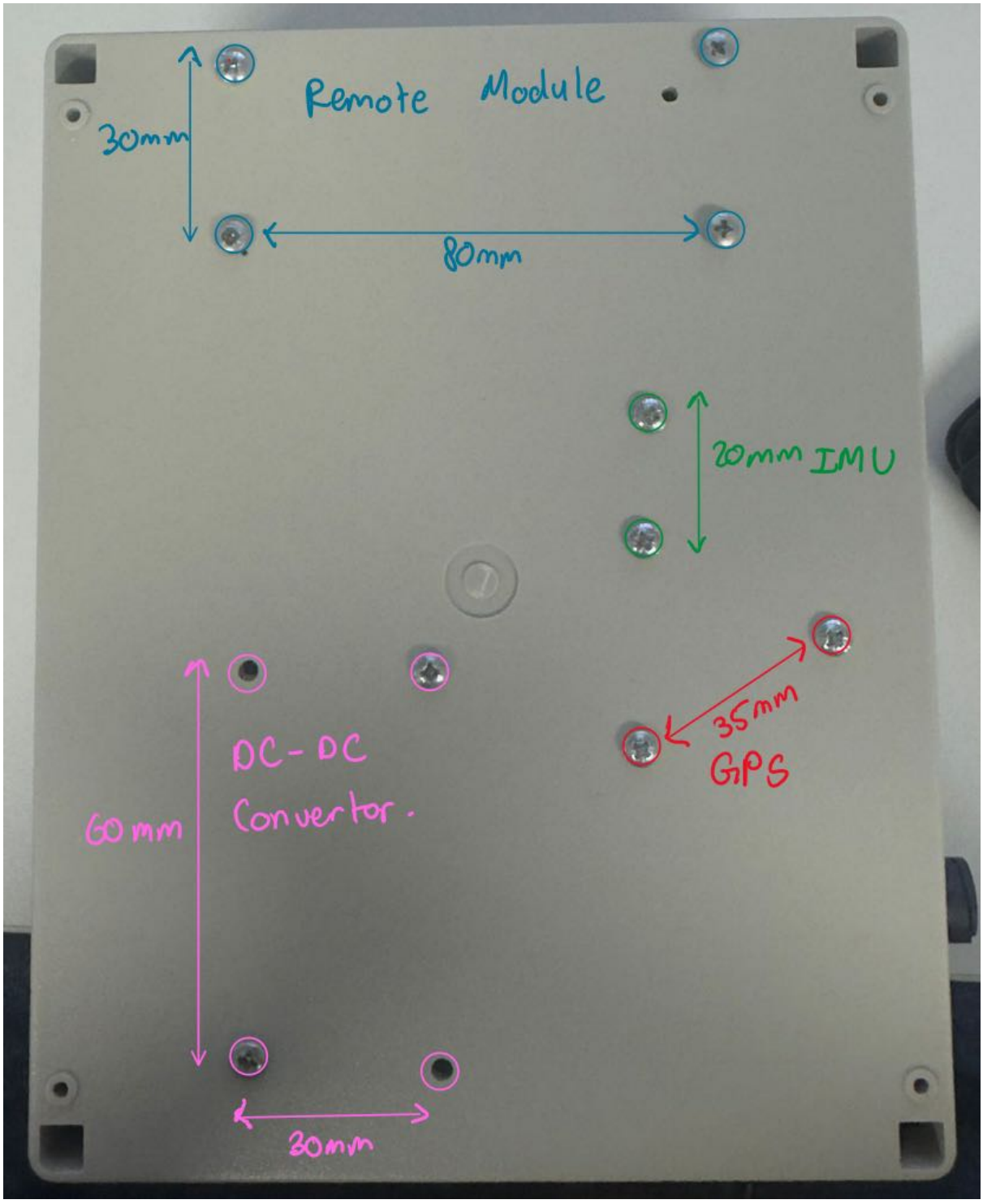


- Attach the modified JST cable and fix the soldered 2.1mm power socket through a hole in the footrest.

- 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 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 ESP32. The electronics box can now be left behind the seat with the power cable routed on the side next to the seats.

Housing Electronics

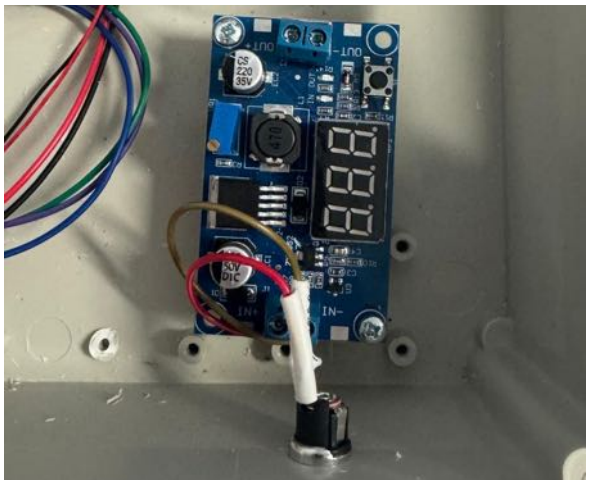
Screw Hole and spacing figure



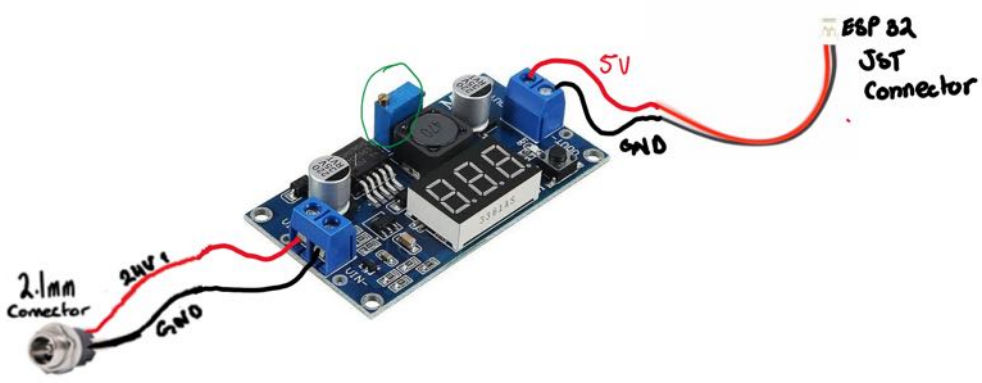
- Drill the appropriate holes using a 3.0mm drill bit into the control box base as per the figure shown above. This is to secure the electrical components such as the GPS, IMU, remote module and the DC- DC convertor to the box. Once drilled place a M3 screw from the bottom of the box and screw a M3 spacer from the inside, mounting the spacer to the floor of the control box, do that for the rest of the holes.

- Once the spacers are mounted, place the components as mentioned in the Figure above on top of the spacers aligning the mounting holes of the component with the spacer hole and screw them in using M3 screws

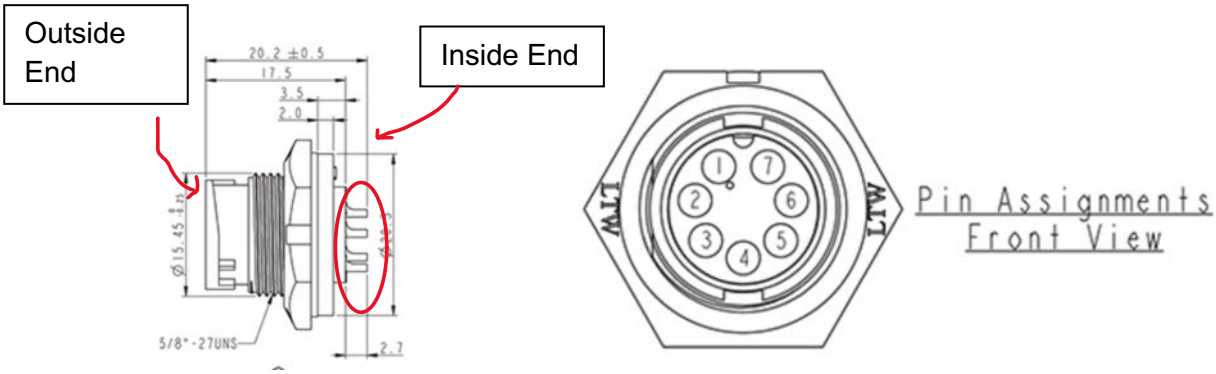
- Drill a hole into the wall suitable for the 2.1mm power socket and screw the socket as shown in the Figure below.



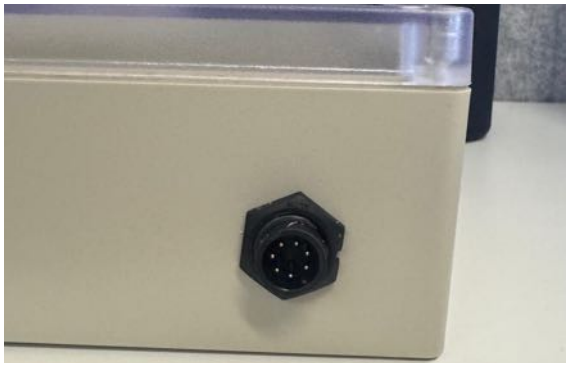
- The wiring diagram for the power is shown below. The 24V and GND wire coming from the 2.1 barrel connector is to be placed inside the DC-DC convertor's VIN+ and VIN- respectively, once placed power the module and adjust the voltage using the potentiometer (circled green in the Figure) until the output voltage reaches 5V as seen on the LCD screen. Then connect the ESP32's JST power connector to the convertor's VOUT terminal block as seen in the Figure.

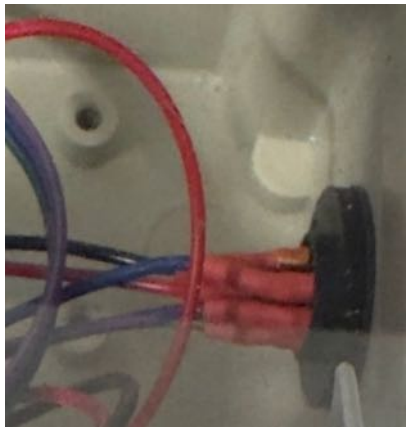


- Using female Dupont cables of different colours (red, black, green, purple, blue), cut and splice the end of one side of the cables.
- We then will be working on the 7 Pin 5A Locking Male Chassis IP67 Waterproof Plug, see images below of what they look like.
- The soldering end of the pins are the one circles in red in the below picture, we can classify that as the inside end of the connector
- Solder the splice end of the red cable to pin 1, black cable to pin 7, and the various colors green, purple, blue to pins 3,4,5 respectively. After soldering heat shrink each cable individually to prevent any short circuiting



- Drill a large hole of size approximately 21mm on the side of the control box for the 7 Pin 5A Locking Male Chassis IP67 Waterproof Plug and mount it with the inside end facing inside the control box and outside end of the connector facing outside the control box as seen in figure below



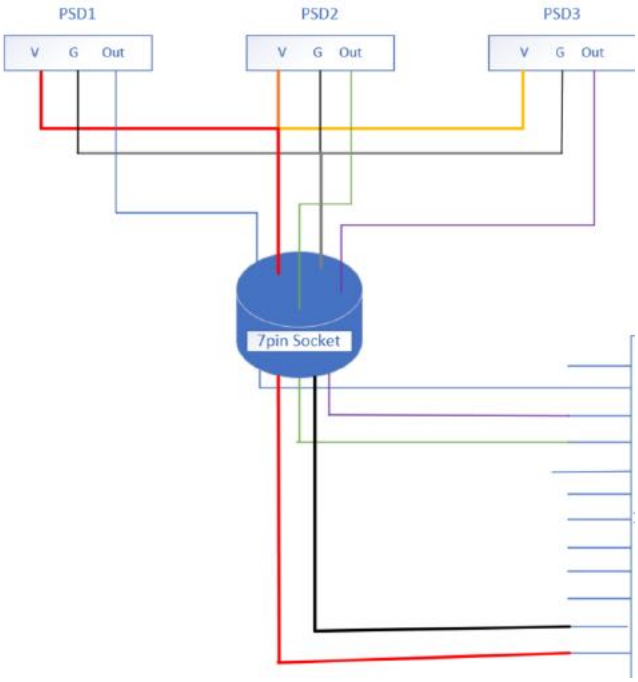


- Now we will be working on the 7 Pin 5A Screw-On Female Line IP67 Waterproof Socket & the 9 Core Shielded Data Cable.
- Using the 9 core shield data cable, identify the colours (red, black, green, purple and blue inside the cable), the rest of the 4 cables will not be used and can be cut off or taped off.

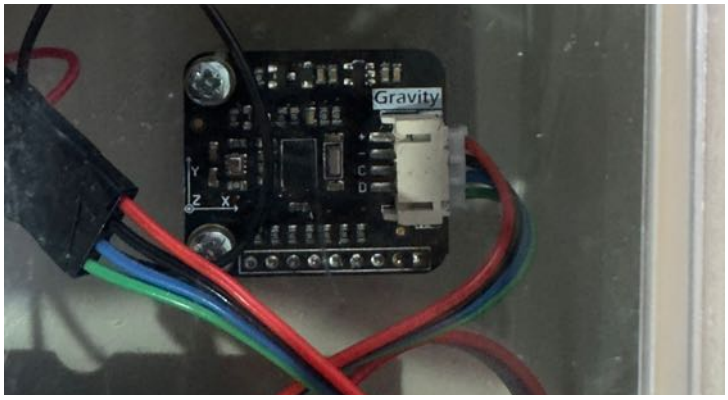
- Using the Pin assignment listed below, solder one end of the 9-core shield data cables each useful wire accordingly to the 7 Pin 5A Screw-On Female Line IP67 Waterproof

Socket, make sure to solder each colour to each pin as it was done to the male socket,

- Note that the male pin assignment and female pin assignment are not the same and are a mirror of each other, critically thinking why this is important, visualise how each pin will align with the other end, hence it is important to keep the red cable to pin 1, black cable to pin 7, and the various colours green, purple, blue to pins 3,4,5 respectively. After soldering heat shrink each cable individually to prevent any short circuiting
- Now we will be working on the other end of the 9-core shield cable and attach it to the 3 PSD sensors.
- Take a PSD sensor and cut and splice the end of the cable so there is exposed wire for power, ground and signal.
- Solder the signal wire of the PSD to one of the various 3 colours, then repeat this for the other 2 PSD for the other 2 colours left
- Then take a spare red wire (for extension) and splice both ends for exposed wire, perform a 4 way solder between each red wire of the PSD and one end of the extension wire, then solder the other end of the extension wire to the red wire coming from the 9-core shield cable
- Perform the same step as above for the black (ground wire)

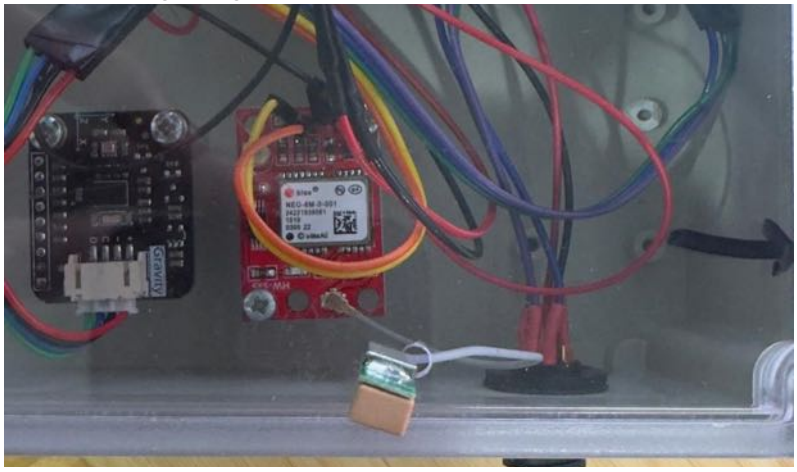


- Attach the provided IMU cable (Gravity: 4Pin I2C/UART Sensor Cable for Arduino) to the Gravity BBNO IMU as shown below

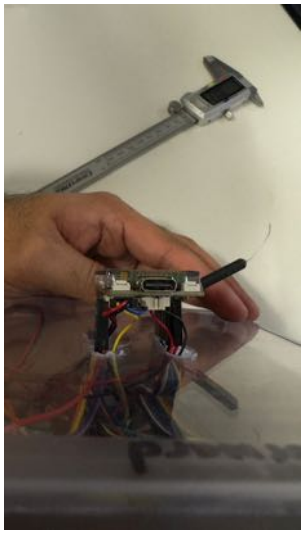


- Now take 4 Male to Female Dupont Cables in the colours of red, black, yellow and orange. Using the image below plug the male end of the red cable into the GPS's VCC solder port, the black cable into GND solder port, orange cable into the TXD solder port and the yellow cable into the RXD solder port.
- Additionally drill a small hole of size 3mm into the lid of the control box exactly above the antenna port circles in the picture and route the antenna connector through the hole and plug it into the antenna port

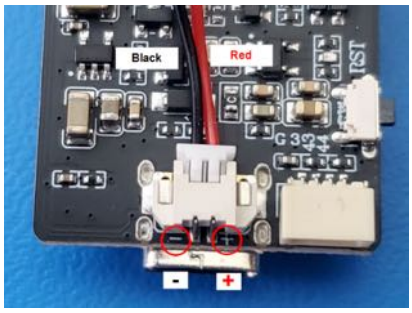
The resulting image should look like this:



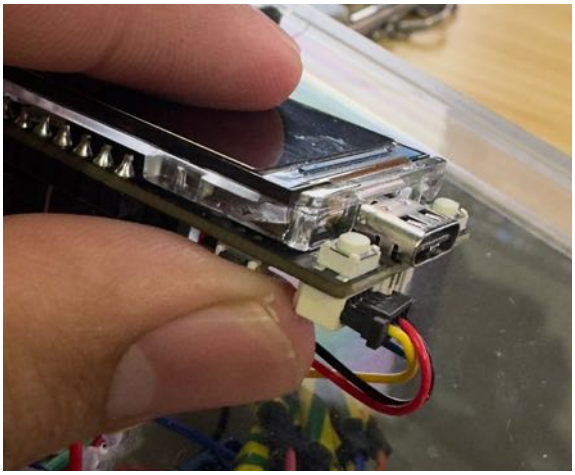
- On the lid either using a drill or a Dremel create 2 pockets for wiring to be routed and the ESP32's pins to be routed towards the inside of the box. This will allow the ESP32 to sit flush on the lid whilst making its pins accessible towards the inside of the box for connectivity to components and power.



- Connect the Esp32 power JST connector coming from the DC-DC convertor like shown below



- Use the 4pin i2C JST connector and connect it to the bottom of the ESP32 as shown below

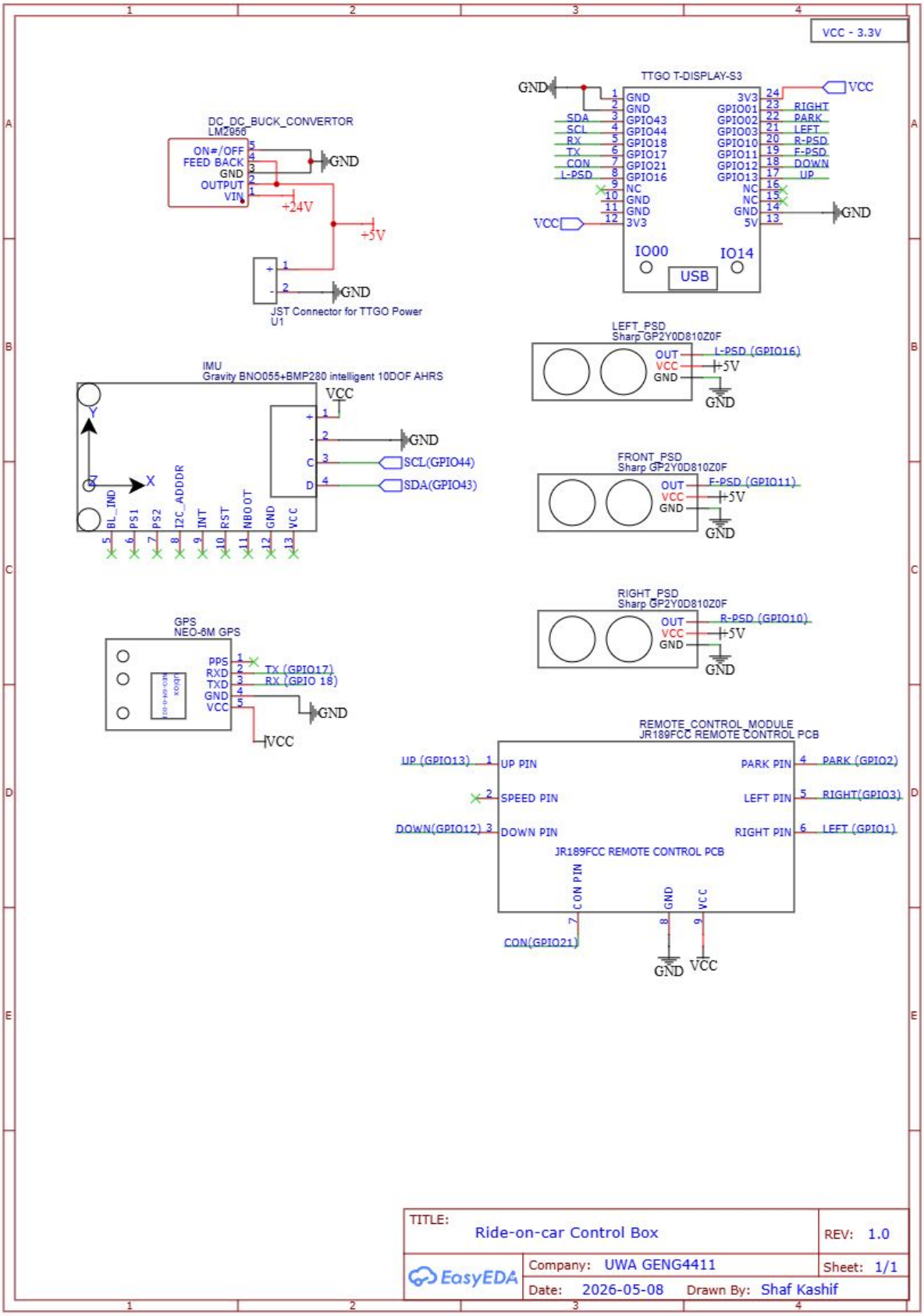


Now it's time to connect all the components to the ESP32. Using the Table below connect each component to the right pin.

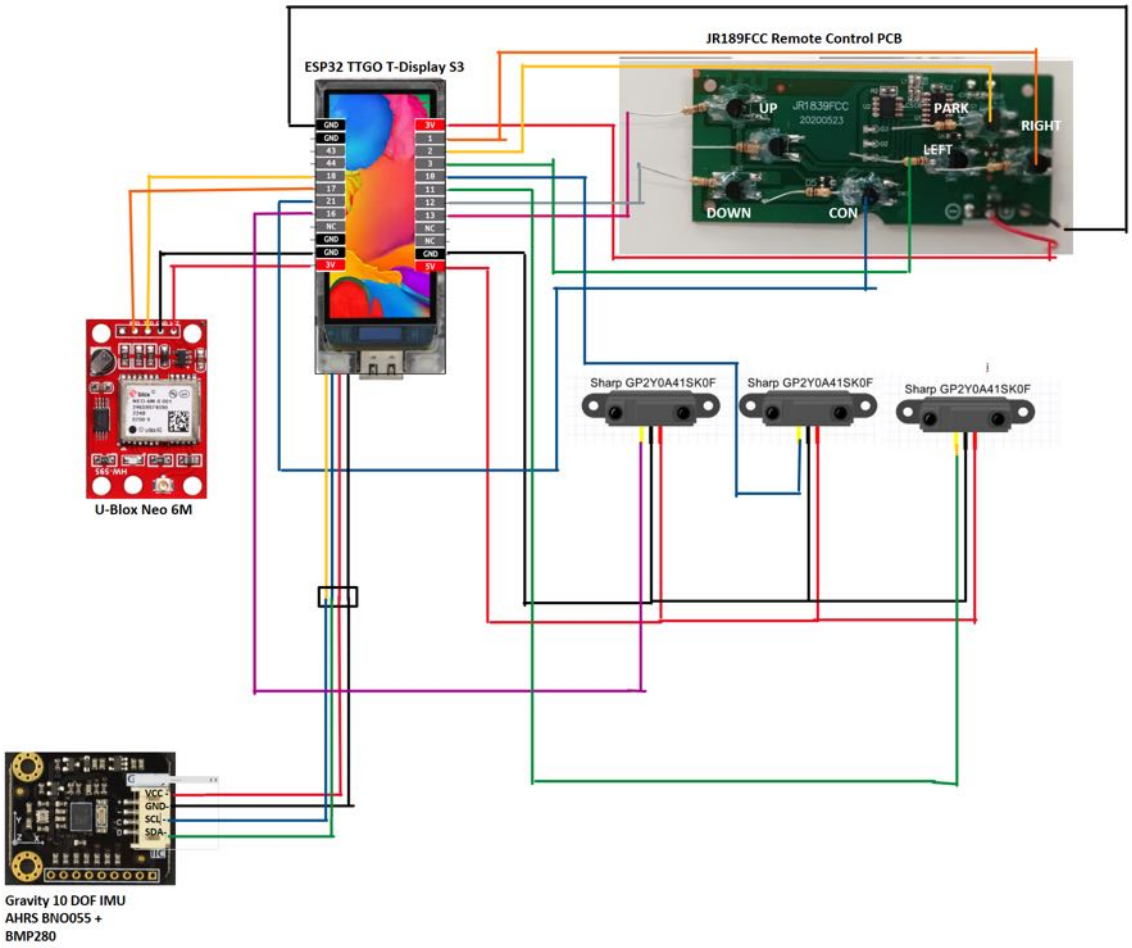
- Note for the IMU, the i2c cable that comes with IMU uses red for VCC, black for GND, blue for SCL and green for SDA, additionally the i2c JST cable for the ESP32 uses red for VCC and black for ground, however, uses blue for SDA and yellow for SCL.
- Note for the UART GPS connection, the RXD on GPS is to be connected to the TX on the ESP32 (orange wire) and the TXD on the GPS is supposed to be connected to the RX on the ESP32 (Yellow wire)
-

TTGO T-Display S3 Pin	**Colour / where the wire is coming from**	**Peripheral Pin/Description**
43	*Coming from the i2c blue cable*	*IMU Sensor SDA (Green cable from i2c)*
44	*Coming from the i2c yellow cable*	*IMU Sensor SCL (Blue cable from i2c)*
18	*Yellow cable coming from GPS*	*GPS UART RX*
17	*Orange cable coming from GPS*	*GPS UART TX*
16	*Green cable coming from 9-core shield cable*	*Left PSD*
10	*Blue cable coming from 9-core shield cable*	*Right PSD*
11	*Purple cable coming from 9-core shield cable*	*Front PSD*
12	*Grey wire coming from remote*	*Remote Down PIN (Backwards)*
13	*Purple wire coming from remote*	*Remote Up PIN (Forward)*
1	*Orange wire coming from remote*	*Remote Right PIN*
2	*Yellow wire coming from remote*	*Remote Park PIN*
3	*Green wire coming from remote*	*Remote Left PIN*
21	*Blue wire coming from remote*	*Remote Connection PIN*

- Connect the GPS and remote-control red wire (VCC) to the 3.3V on the ESP32 using the female Dupont cables attached to the components
- The IMU VCC cable is already supplied with 3.3V through the 2 i2c cables and the grounds are attached.
- The PSD sensors and thus the 7-pin connector that has the female Dupont cables attached, the red wire should be connected to 5V on the esp32
- All grounds of the components can be connected to any grounds on the esp32
- The wiring diagram can be seen below, additionally a simplified colour version of the diagrams can be seen below


VCC - 3.3V
DC_DC_BUCK_CONVERTOR
LM2956
ON#/OFF
FEED BACK
GND
OUTPUT
VIN
GND
+24V
+5V
GND
JST Connector for TTGO Power
U1
TTGO T-DISPLAY-S3
GND
VCC
SDA
SCL
RX
TX
CON
L-PSD
GPIO43
GPIO44
GPIO18
GPIO17
GPIO21
GPIO16
3V3
GPIO001
GPIO002
GPIO003
GPIO010
GPIO011
GPIO012
GPIO013
RIGHT
PARK
LEFT
R-PSD
F-PSD
DOWN
UP
NC
5V
IO00
IO14
USB
LEFT_PSD
Sharp GP2Y0D810Z0F
OUT
VCC
GND
L-PSD (GPIO16)
+5V
IMU
Gravity BNO055+BMP280 intelligent 10DOF AHRS
SCL(GPIO44)
SDA(GPIO43)
BL_IND
PS1
PS2
I2C_ADDDR
INT
RST
NBOOT
FRONT_PSD
Sharp GP2Y0D810Z0F
F-PSD (GPIO11)
RIGHT_PSD
Sharp GP2Y0D810Z0F
R-PSD (GPIO10)
GPS
NEO-6M GPS
PPS
RXD
TXD
TX (GPIO17)
RX (GPIO 18)
REMOTE_CONTROL_MODULE
JR189FCC REMOTE CONTROL PCB
UP (GPIO13)
UP PIN
SPEED PIN
DOWN(GPIO12)
DOWN PIN
PARK PIN
PARK (GPIO2)
LEFT PIN
RIGHT(GPIO3)
RIGHT PIN
LEFT (GPIO1)
CON PIN
CON(GPIO21)
TITLE:
Ride-on-car Control Box
REV: 1.0
EasyEDA
Company: UWA GENG4411
Sheet: 1/1
Date: 2026-05-08
Drawn By: Shaf Kashif

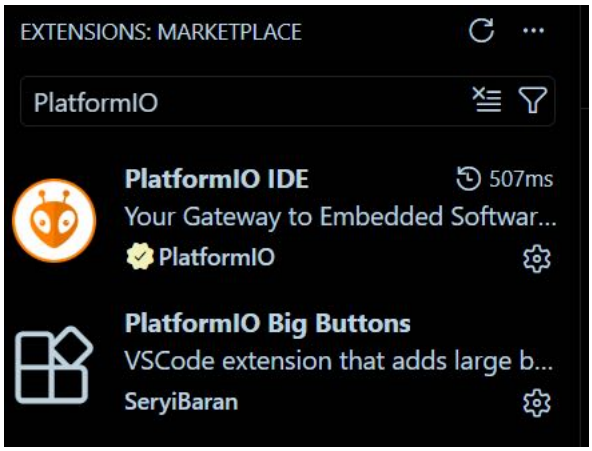
-


JR189FCC Remote Control PCB
ESP32 TTGO T-Display S3
UP
JR1839FCC
20200523
PARK
RIGHT
LEFT
DOWN
CON
Sharp GP2Y0A41SK0F
Sharp GP2Y0A41SK0F
Sharp GP2Y0A41SK0F
U-Blox Neo 6M
VCC
GND
SCL
SDA
Gravity 10 DOF IMU
AHRS BNO055 +
BMP280

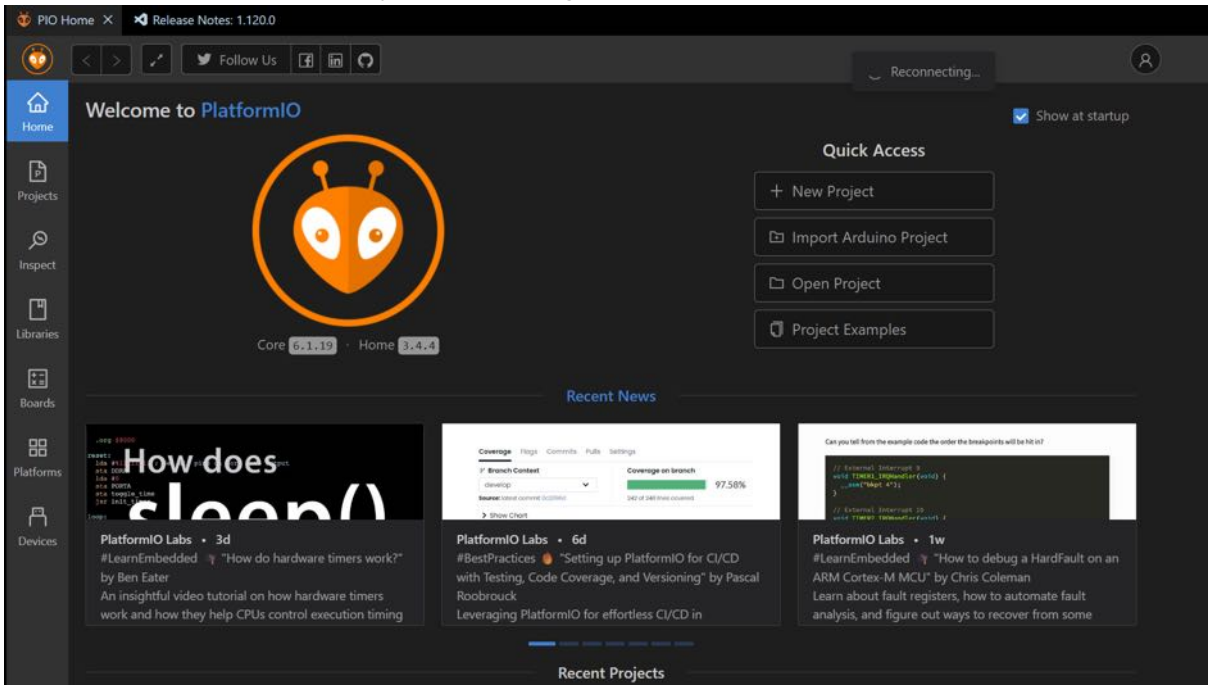
Software Side

- Using a laptop or computer desktop, install VS Code.

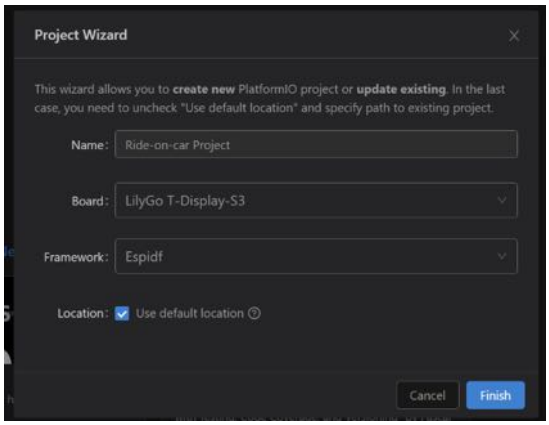
- Once VS Code is installed, go to extensions and install PlatformIO IDE & PlatformIO big buttons



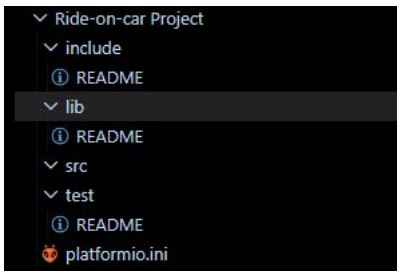
-
- Once installed it should land you on this page



- Click on the New Project Button and Enter the following details



- Once completed it should open up the workspace as follow:



- Click on the platformio.ini file and copy across the following

```
- [env:lilygo-t-display-s3]
- platform = espressif32
- board = lilygo-t-display-s3
- framework = arduino
- monitor_speed = 115200
- board_build.filesystem = littlefs
-
- lib_deps =
-   Bodmer/TFT_eSPI@^2.5.35
-   adafruit/Adafruit BNO055 @ ^1.6.4
-   adafruit/Adafruit Unified Sensor @ ^1.1.9
-   mikalhart/TinyGPSPlus@^1.0.3
-   https://github.com/me-no-dev/ESPAsyncWebServer.git
-   https://github.com/me-no-dev/AsyncTCP.git
-
-
- build_flags =
-   -DUSER_SETUP_LOADED=1
-   -DUSER_SETUP_FILE="TTGO_T_Display_S3.h"
-   -include
  $PROJECT_LIBDEPS_DIR/$PIOENV/TFT_eSPI/User_Setups/Setup206_LilyGo_T_Display_S3.h
-
-
- ; (Optional but handy)
- monitor_filters = esp32_exception_decoder, time
-
```

- Then click on the src folder and create a new file and save it as main.cpp

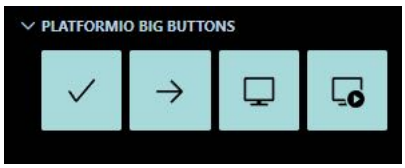
- This is the main file where you would do the primary coding

- Add the following libraries and pin allocation

```
include <ESPAsyncWebServer.h>
#include <Arduino.h>
#include <Wire.h>
#include <TFT_eSPI.h>
#include <SPI.h>
#include <TinyGPS++.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BNO055.h>
#include <utility/imumaths.h>
#include <math.h>

#define TFT_BL 38
TFT_eSPI tft = TFT_eSPI();
const int pinFrontPSD = 11;
const int pinLeftPSD  = 16;
const int pinRightPSD = 10;

// I2C pins (shared by BNO055)
const int I2C_SDA = 43;
const int I2C_SCL = 44;

// ---------------- Serial Pins for GPS ----------------
const int pinRX = 18; // RX of GPS
const int pinTX = 17; // TX of GPS

// ---------------- Controller Pins ----------------
const int pinLeft  = 3;   // Green wire
const int pinRight = 1;   // Orange wire
const int pinUp    = 13;  // White wire
const int pinDown  = 12;  // Grey wire
const int CON      = 21;  // Blue wire
const int parkPin  = 2;   // Yellow wire
//const int speedPin = 11;  // Purple wire (disabled for now)


// ---------------- GPS ----------------
#define GPS_BAUD 9600
TinyGPSPlus gps;
HardwareSerial gpsSerial(2);
```

- You are now ready to start coding the ride-on-car

- Once you have some code you want to upload click on the tick button to compile and → button to upload the code on to the ESP32, make sure you have a USB cable plugged from your computer on the usb c port on the ESP32.



Contact

UWA Robotics & Automation Research Lab

https://roblab.org *or* https://robotics.ee.uwa.edu.au
thomas.braunl@uwa.edu.au
Professor Thomas Bräunl
Shaf Kashif

UWA Robotics Club

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