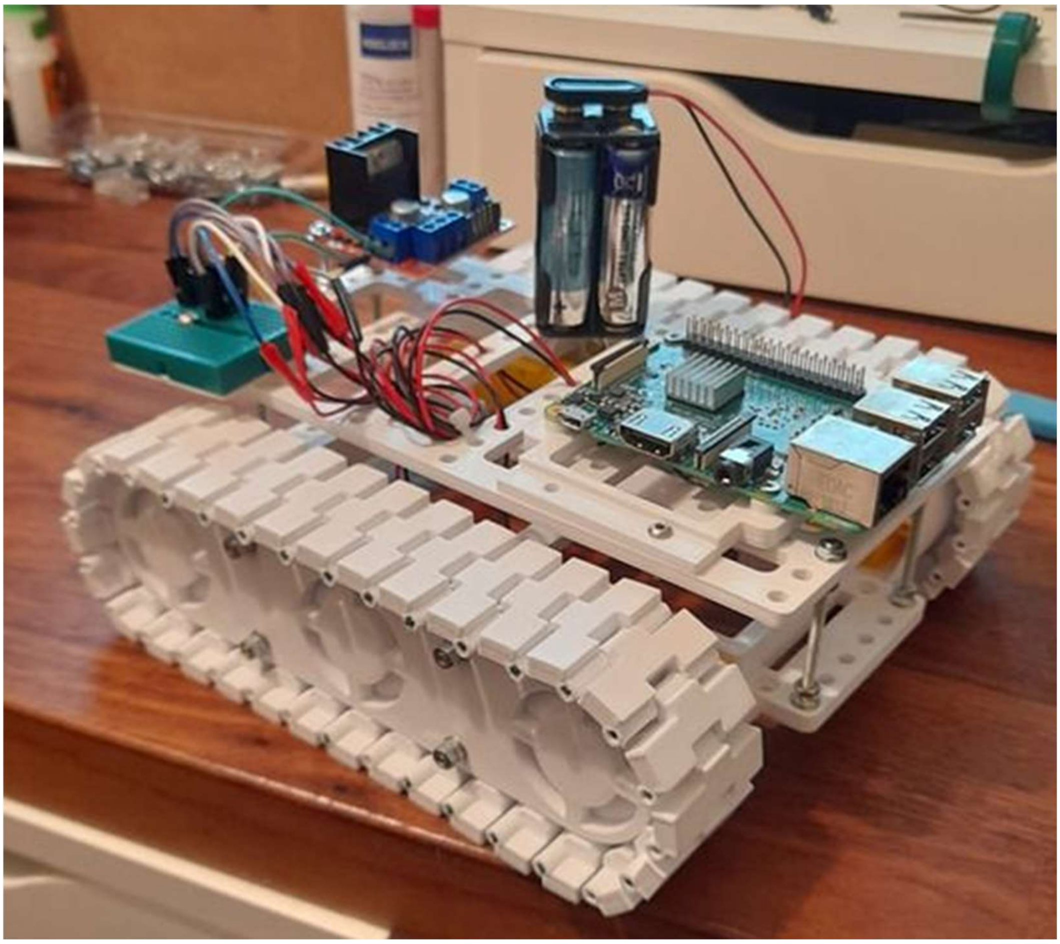
Here is a basic setup of the electronics for my Pi-tank.

I’m pretty sure someone else could do a much better job, but for the sake of keeping it simple so everyone can understand what’s happening. this doc will help explain how I made mine work.

PI-TANK



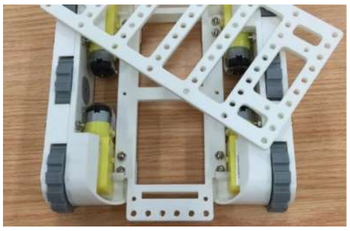
Index

Contents

Requirements.

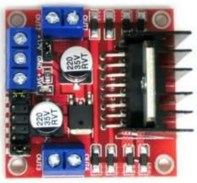
Below is a list of the components you need to make the basic tank.

1. Print the proto tank from thingivese - https://www.thingiverse.com/thing:972768.



2. A raspberry pi. (I'm using a pi 3)

3. A L298n motor controller board. (ebay link https://www.ebay.co.uk/itm/L298N-Dual-H-Bridge-Stepper-Motor-Driver-Controller-Board-Module-for-Arduino-UK/193603037174?)



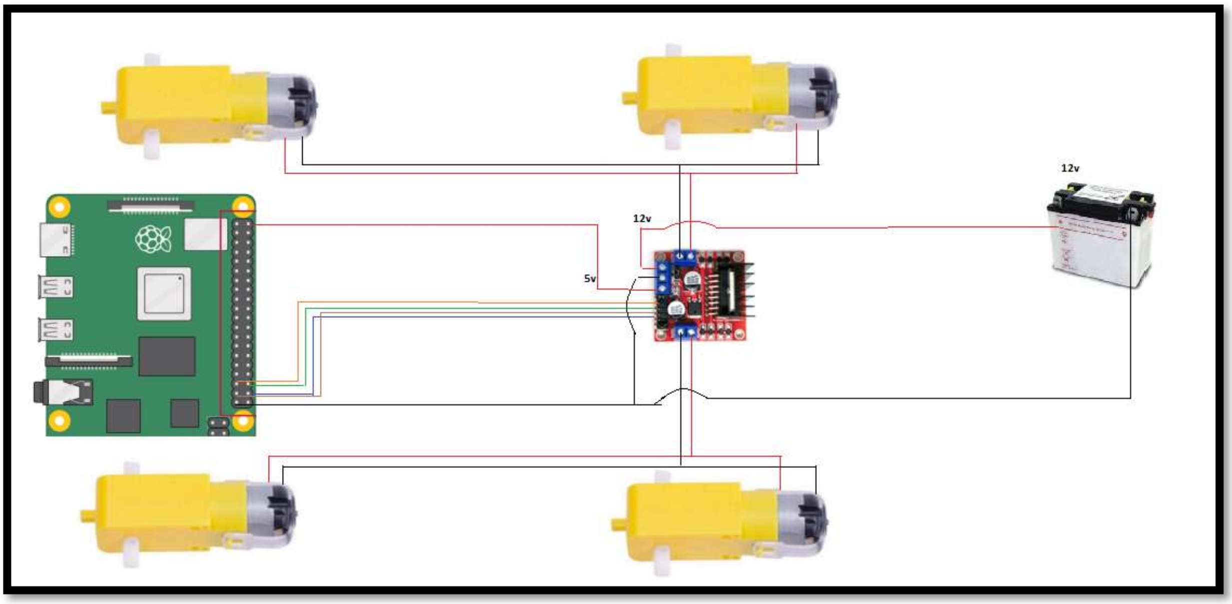
4. 4 motors for driving it. (ebay link https://www.ebay.co.uk/itm/Repairs-DIY-DC-3V-6V-Gear-TT-Motor-Tire-Wheel-Kit-For-Arduino-Smart-Car-Robot/303525513498)

5. A bunch of wires for connecting it all up 😊
6. 5v battery
7. 12v battery

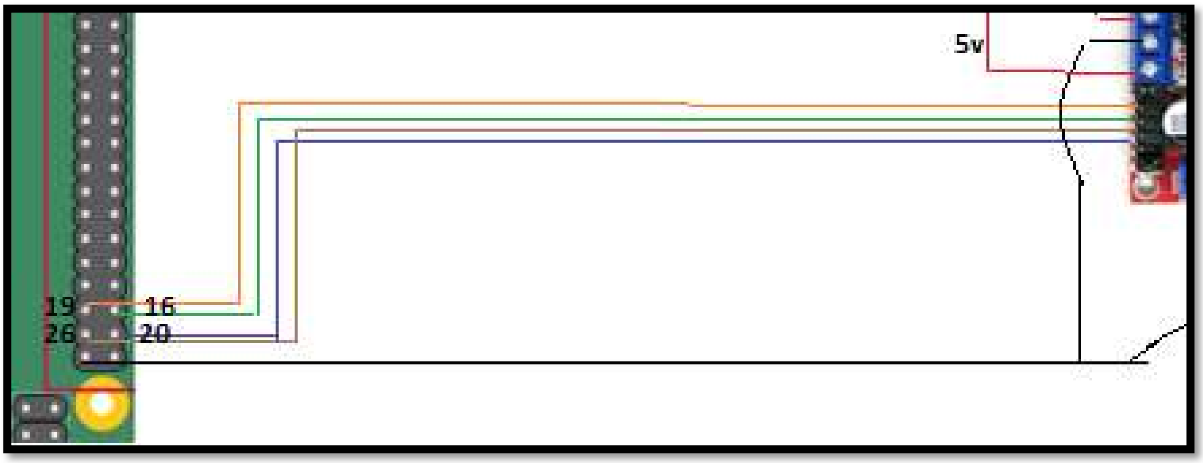
Connecting it all up (hardware)

The wires on my Pi-tank look messy, but its quite simple to wire up.
Below is a basic sketch of wiring.

1. I've connected the left 2 motors together, and same on the right.
2. Then feed the two pairs of wires into the L298n controller board.
3. 12v connects to the L298n board to power the motors. 5v to power the IC chip and raspberry pi.
4. (*The picture doesn't show it. But make sure you power up the PI as you usually would* – or connect up a battery to power the Pi.)



Closer look at the pins.

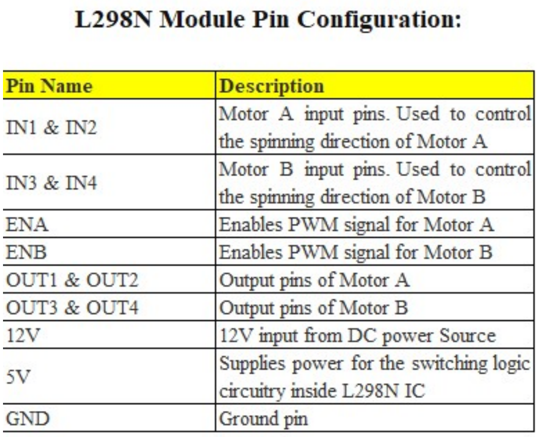


About the L298n board

Basically, this board has 2 sets of input pins per motor.. (the pins that go to the raspberry pi)
This is to control forwards and backward actions.

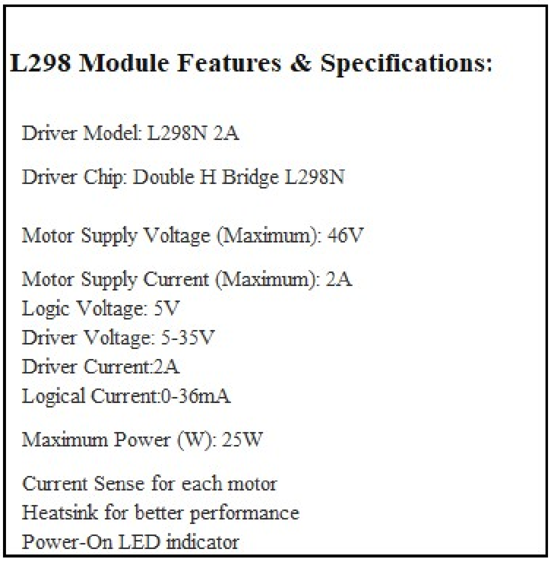
e.g.
for Motor A, you have to set pin1 **on** and pin2 **off**. = Forwards action.
for Motor A, you have to set pin1 **off** and pin2 **on**. = Backwards action.
for Motor A, you have to set pin1 **off** and pin2 **off**. = stop action.

for Motor B, you have to set pin3 **on** and pin4 **off**. = Forwards action.
for Motor B, you have to set pin3 **off** and pin4 **on**. = Backwards action.
for Motor B, you have to set pin3 **off** and pin4 **off**. = stop action.



L298N Module Pin Configuration:

Pin Name	Description
IN1 & IN2	Motor A input pins. Used to control the spinning direction of Motor A
IN3 & IN4	Motor B input pins. Used to control the spinning direction of Motor B
ENA	Enables PWM signal for Motor A
ENB	Enables PWM signal for Motor B
OUT1 & OUT2	Output pins of Motor A
OUT3 & OUT4	Output pins of Motor B
12V	12V input from DC power Source
5V	Supplies power for the switching logic circuitry inside L298N IC
GND	Ground pin



L298 Module Features & Specifications:

Driver Model: L298N 2A

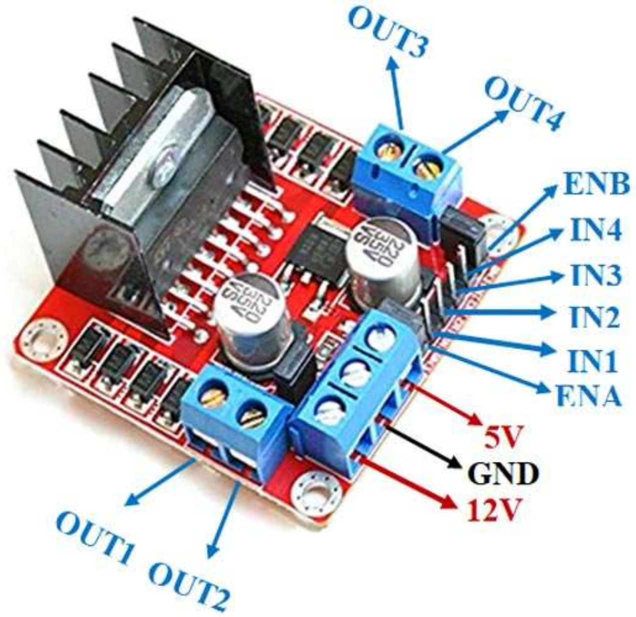
Driver Chip: Double H Bridge L298N

Motor Supply Voltage (Maximum): 46V

Motor Supply Current (Maximum): 2A
Logic Voltage: 5V
Driver Voltage: 5-35V
Driver Current:2A
Logical Current:0-36mA

Maximum Power (W): 25W

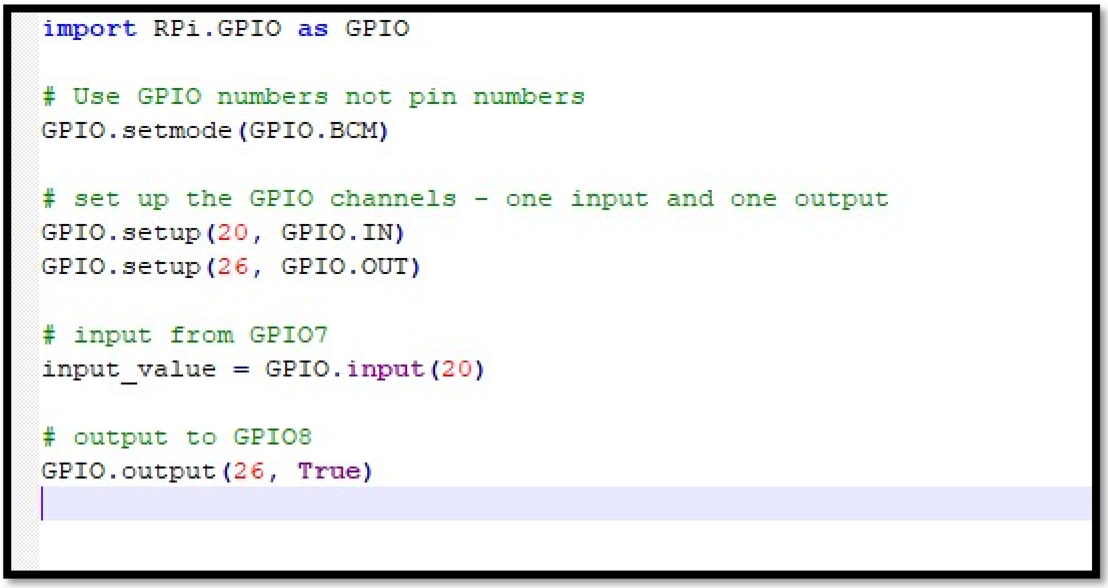
Current Sense for each motor
Heatsink for better performance
Power-On LED indicator



So. Using code, how do we make it move?

Let’s explain how to send a basic command to the controller board

I created a folder on my raspberry pi and created a file called PiBackwardsLeft.py. (*because I wanted my tanks left wheels to turn in reverse)*
Inside that file I created this code.



```
import RPi.GPIO as GPIO

# Use GPIO numbers not pin numbers
GPIO.setmode(GPIO.BCM)

# set up the GPIO channels - one input and one output
GPIO.setup(20, GPIO.IN)
GPIO.setup(26, GPIO.OUT)

# input from GPIO7
input_value = GPIO.input(20)

# output to GPIO8
GPIO.output(26, True)
```

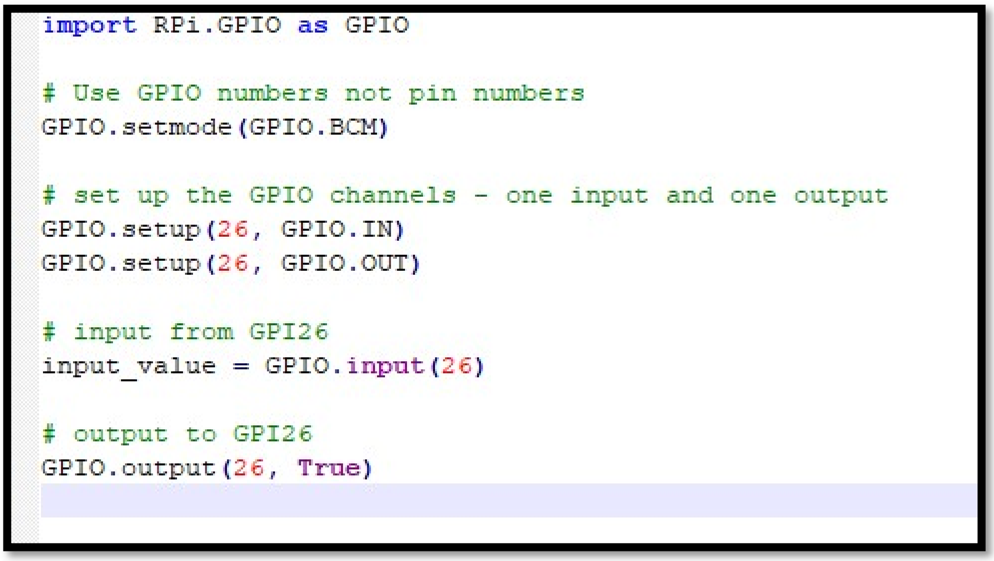
We have to say which pins we’re using, so I specified pins 20 and 26.

If you remember I said we need to have one pin **On** and one pin **Off.** Which is what we’re doing here. I’ve turned in 26 on, and left 20 off.

To execute this script on my raspberry pi I type “python PiBackwardsLeft.py “

Stopping the Motor..

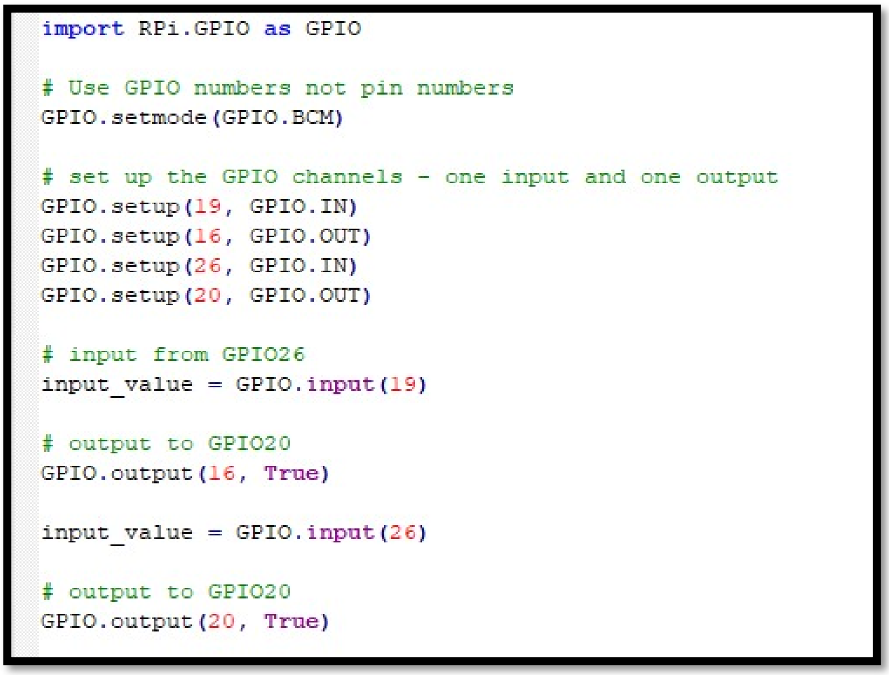
To stop the motor we have to set the pins to be the same (*e.g. set them both to on or both to off*)



```
import RPi.GPIO as GPIO

# Use GPIO numbers not pin numbers
GPIO.setmode(GPIO.BCM)

# set up the GPIO channels - one input and one output
GPIO.setup(26, GPIO.IN)
GPIO.setup(26, GPIO.OUT)

# input from GPI26
input_value = GPIO.input(26)

# output to GPI26
GPIO.output(26, True)
```

Script - Combining all together:

Using the method above you are only controlling one side of the tank. So to move the tank forward you have to do that twice (once for each side)



```
import RPi.GPIO as GPIO

# Use GPIO numbers not pin numbers
GPIO.setmode(GPIO.BCM)

# set up the GPIO channels - one input and one output
GPIO.setup(19, GPIO.IN)
GPIO.setup(16, GPIO.OUT)
GPIO.setup(26, GPIO.IN)
GPIO.setup(20, GPIO.OUT)

# input from GPIO26
input_value = GPIO.input(19)

# output to GPIO20
GPIO.output(16, True)

input_value = GPIO.input(26)

# output to GPIO20
GPIO.output(20, True)
```

Like wise, if we want to stop the tank going forward, we need to stop both left and right sites..

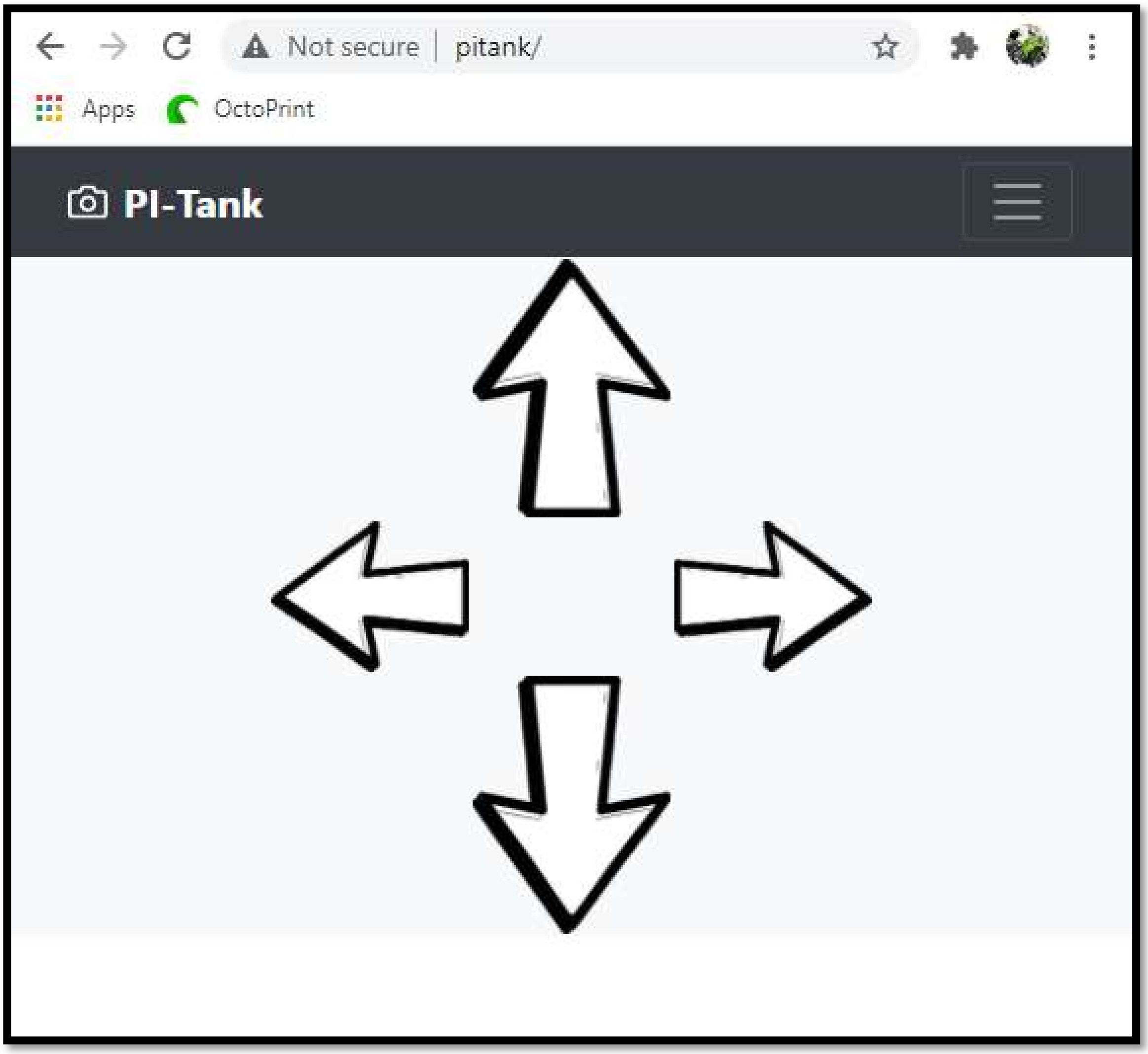
(handy if you want the tank to turn whilst moving forward, just stop one side)

Make a remote control for it?

I made a very simple webpage on the raspberry pi, that just executes the Python scripts from above.

When I click up, it submits the forward action. I then have another button that appears to cancel the forward command.

Then the same for the other actions, left, right, back etc.



I'll drop my example of this on github along with the python scripts.