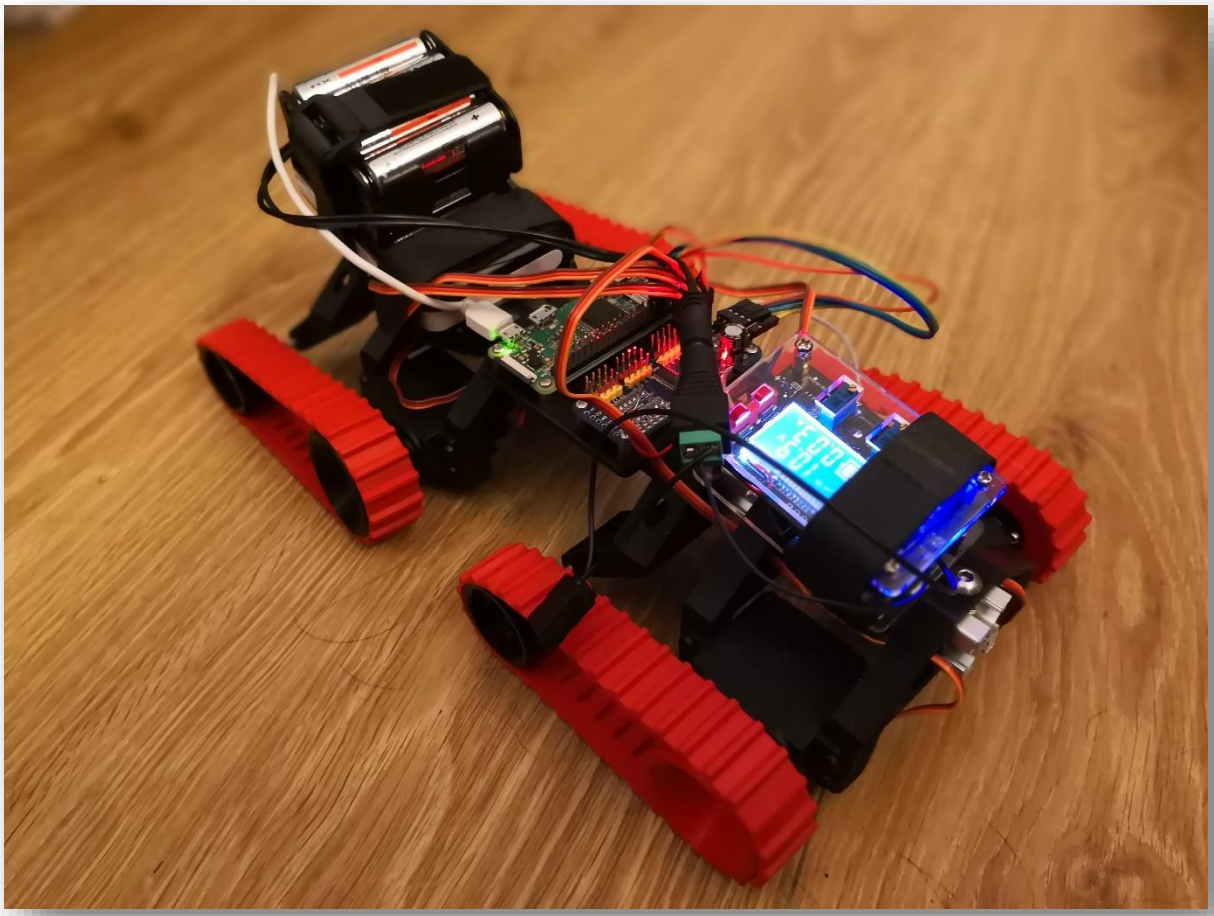
Pi Zero Robot (Zerobot)



Licence

Summary

The Pi Zero Robot (or Zerobot) is a modular robot platform based around 2020 extrusion. This gives it the flexibility to scale in size to suit various needs.

For example, adding a high speed motor could allow this design to be adapted to become a robot lawnmower as easily as it could be adapted to be a small robot to explore small confined spaces. The hardware design is flexible enough to allow sensors, robot arms and cameras or anything else to be added.

The bot is coded with johnny-five.io which is a node.js based robotics tools. This makes it easy to add additional hardware modules (including an Arduino tethered to the Pi if it's needed).

As this is an open source project, please share any improvements you make to this by remixing it on Thingiverse or sharing your code on GitHub.

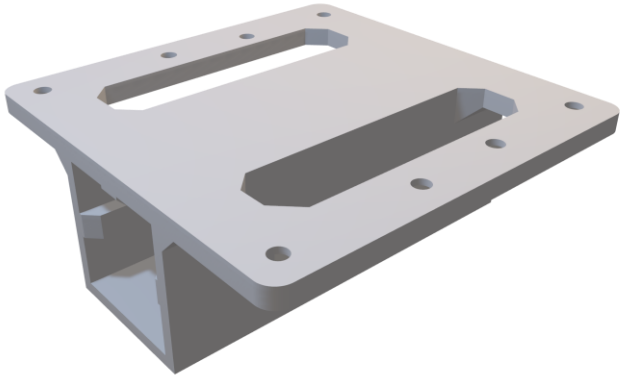
STL files?

All the STL files can be found at https://www.thingiverse.com/thing:3061928/.

I've designed these in DesignSpark Mechanical (DSM). The DSM files are in the github repository at https://github.com/stuartmunn/Pi-Zero-Robot. DSM software is free and you can use it to save the files in other formats if you'd prefer.

Pi_Zero_16_PWM_Base

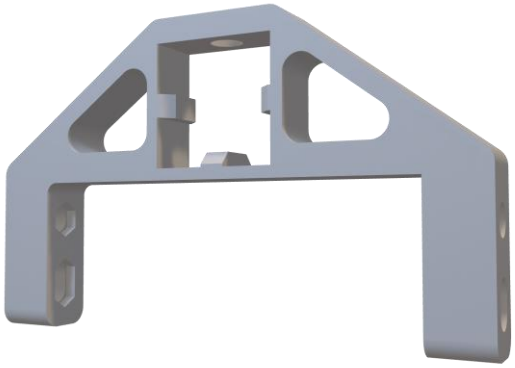
This is where you'll mount the Pi and Servo driver board. You'll need just one of these and some



M2.5 standoffs, bolts and nuts to attach the boards.

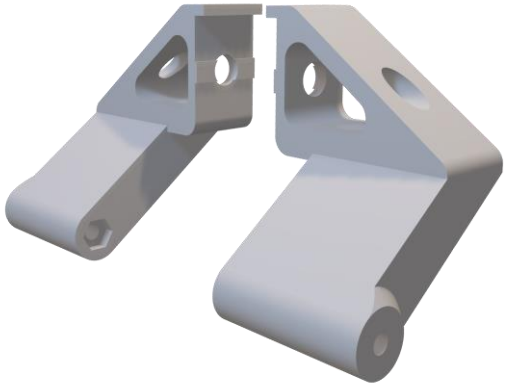
Double Servo Mount

These are designed to hold two servo motors either side of a 2020 bar. You'll need two of these for each pair of motors you use. Use M3 bolts and nuts to hold the servos to the mount and M5x8mm bolts and T-Slot nuts to hold it onto the 2020.

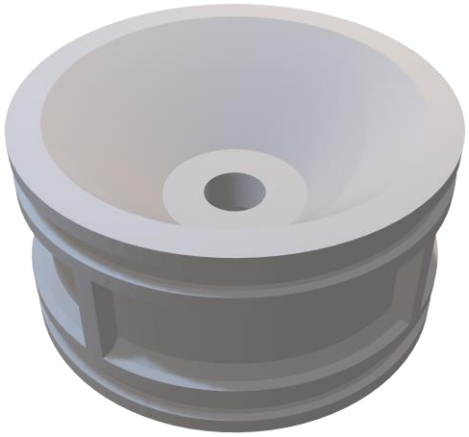


Idler Wheel Supports

If you're making a tracked vehicle these can hold the idler wheels. Long M3 bolts. These could do with some improvement. Use M5x8mm and T-Slot nuts to hold these on.



Tread Wheels



You can use any wheels you want. These ones are designed to work with the lego technics treads – the circular disc that comes with the servos should fit the back nicely.

Assembly of a 4WD simple tracked robot

The design of this robot is very modular, but it always helps to have an example to start with. These assembly instructions will give you a guide to building your robot.

Parts required

Apendix A has a link to all the parts I've used.

- 2020 Aluminium Extrusion (as much as you need and you can print 2020 profile if you need it)
- Tracks (while there are some printable tracks, I've chosen to use Lego ones)
- Idler wheels (again, I've used lego ones – no point re-inventing the wheel…)
- M5x8mm Bolts
- M5 T-Slot Nuts (you can print these if you prefer)
- M2.5 Standoffs, bolts and nuts.
- M3 Bolts (various lengths) and nuts
- 4 standard sized continuous servo motors

Print List

You'll need to print the following:

- 1 x Pi_Zero_16_PWM_Base.stl
- 4 x DoubleServoMount.stl
- 2 x IdlerWheelSupport.stl
- 4 x TreadWheels.stl

Putting it together

1. Use M2.5 standoffs to attach the Raspberry Pi Zero and the PCA9685 PWM board to the the Pi_Zero_16_PWM_Base print. There are M2.5 bolt shapes on the bottom for the bolts.
2. Attach one servo to one side of two DoubleServoMounts using M3 bolts and nuts.
3. Attach another servo to the other side using more M3 bolts and nuts.
4. Repeat 2 & 3 with remaining servos and mounts.
5. Attach your idler wheels to the idlerwheelsupports using a long M3 bolt and nuts.
6. Slide the parts onto your 2020 Extrusion bar in order:
   (L → R) ==Idler===ServoMount==Pi/PWM===Idler===ServoMount
7. Use M5 8mm bolts and M5 T-Slot nuts to secure parts in place. It's a good idea not to tighten the bolt too much yet – you may need to make some slight adjustments once the treads are on.

8. Attach the servo horns to your tread wheels (they are designed for the round horn). There's no holes for screwing these on, but your servo should come with appropriate screws for the job).
9. Attach the wheels to the servos
10. Put the treads on.
11. Slide servo mounts and idler supports about to take the slack out of the treads and tighten the bolts.

At present, the batter holders for this are still in design. Velcro straps do a pretty good job of holding batteries until this part is done.

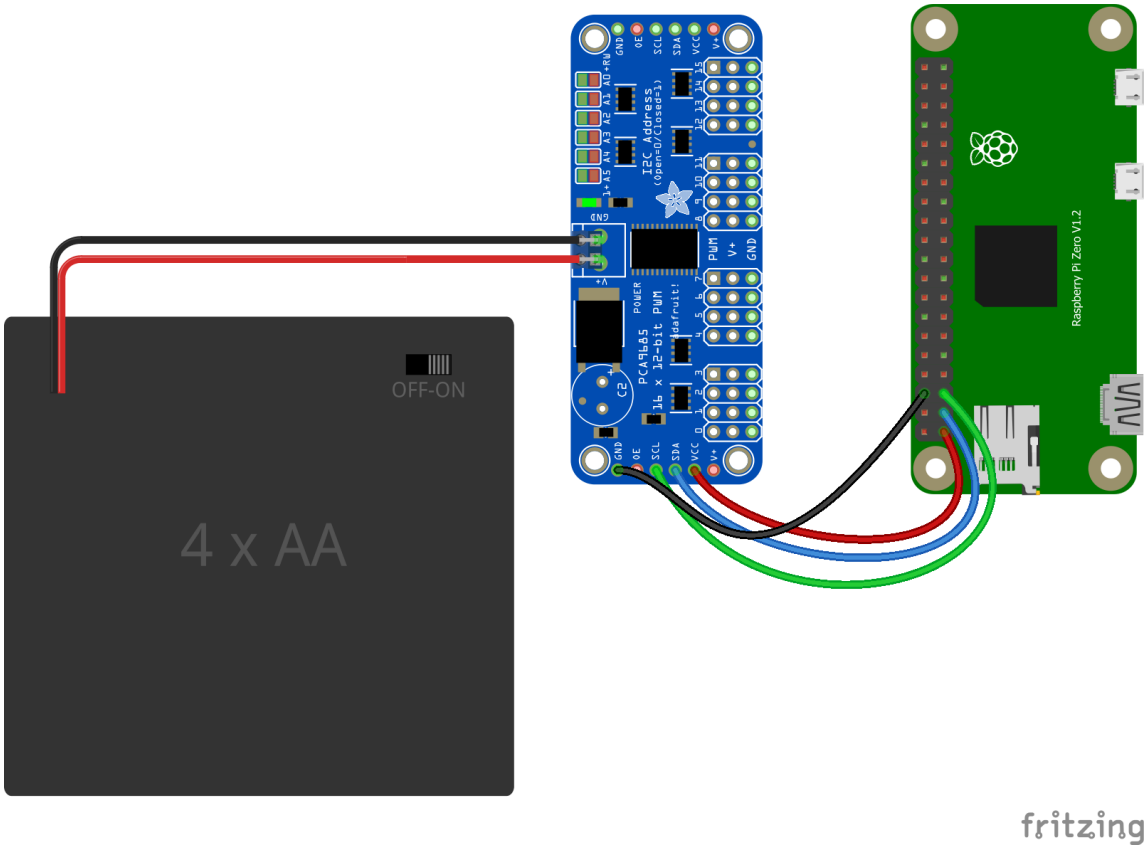
Preparing the Pi

Put a standard Raspbian Lite image onto the Pi and then follow the instructions at https://github.com/nebrius/raspi-io/wiki/Getting-a-Raspberry-Pi-ready-for-NodeBots.

Wiring it up

If you've got this far, you'll be looking at a bot that's largely assembled, but needs some wires.

Here's a wiring diagram to follow:



Connect the servos to the PWM board as follows:

Servo	PWM Pin
Front Left	0
Front Right	1
Rear Left	2
Rear Right	3

Code

There is some basic code to get the bot moving and test it on GitHub.

There will soon be some code to control the bot over SSH and hopefully over http soon. I'm also working on controlling it over RF (remote controlled lawnmower anyone?) as well as some autonomous code.

GitHub repository: https://github.com/stuartmunn/Pi-Zero-Robot

To clone this to your Raspberry Pi, use the following command:

git clone https://github.com/stuartmunn/Pi-Zero-Robot.git

To run the drivetest.js program type the following

cd ./Pi-Zero-Robot/zerobot/

sudo node drivetest.js

This should drive all wheels forward for 3 seconds and then stop. If it doesn't stop, you probably need to adjust the stop point on your servos until they stop moving. You can also issue some commands while drivertest.js is running:

Command	What it does
wheelsleft.cw(x)	Left wheels forward. X is the speed from 0 to 1.
wheelsright.cw(x)	Right wheels forward. Again x is the speed from 0 to 1.
wheelsleft.ccw(x)	Left wheels reverse. x is the speed from 0 to 1.
wheelsright.ccw(x)	Right wheels reverse. x is the speed from 0 to 1.
wheelsleft.stop()	Stops the left wheels
wheelsright.stop()	Stops the right wheels
.exit()	Stops the program – note doesn't stop the wheels moving.

There is also a shcontrol.js node script that uses the arrow keys for control and the space bar for stop.

Writing your own code

All the code provided is meant more as an example than as a finished product.

Half the fun is writing your own code. Also, I'm a terrible coder, so trust me, your better off writing your own!

There's plenty of example code at http://johnny-five.io – its well worth a look. Feel free to use the code I've provided as a starter. If you do write your own, I'd encourage you to fork the github repository and share your code with the world.