Designing and Constructing - “Mars Rover”

Combining Hardware and Software

Ehren Chan || Anda Su

Computer Engineering 11 - TEJ3M

J. Radulovic

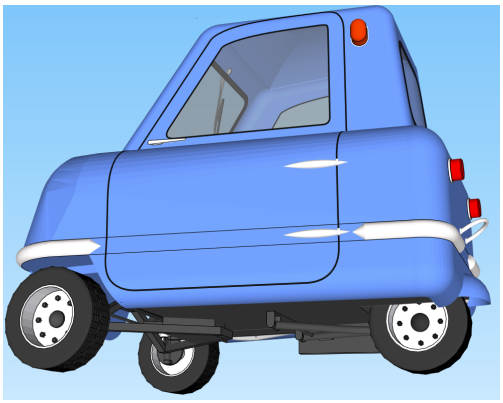
18/06/2018

TABLE OF CONTENTS:

1) Background Information

For this project, our goal is to design a rover that is fully functional while keeping costs as low as possible. This means, we will construct this with only three wheels — as that's all one would need for a stable structure. We also will use small wheels and motors, as that will lower the cost of the rover. Some other basic components include a battery, a raspberry pi microcontroller, a camera, a dual h-bridge, and couplers.

After constructing, it should look like a 3-wheeled car, where the front 2 wheels are motor driven and are responsible for moving forward, backward, and turning. The back wheel would only be responsible for moving freely. This will further cutdown on costs as we only need to drive 2 motors.

It should look like this model of a 3-wheel car:  (Model by kpat on Google Sketchup, link below)

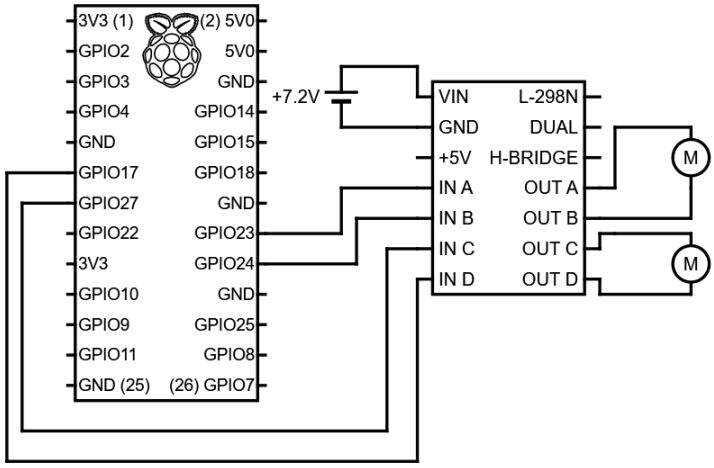
There are many milestones we need to check off before being able to drive this rover. This includes: driving motors via a microcontroller (GPIO pins on the raspberry pi), programming the commands to drive the motors (Python programs), connect and communicate the microcontroller wirelessly (Wi-Fi connection on the pi & SSH), wiring the components correctly (using the dual h-bridge, powering the microcontroller using a battery instead of a wall adapter), and live streaming video feed from the camera (VLC and reducing video delay on the network). After all these components are put together, we will finally be able to fine tune the movement of the rover (configure the motors to turn in a specific way) and then be able to drive it remotely.

2) Proposal

Step	Description	Days	Person responsible
1	set up pi, power on and connect to network, enable ssh	1	Anda/Ehren
2	connect to pi - ssh, install python and idle	1	Anda/Ehren
3	research : GPIO(Anda/Ehren), write a quick program and test signals with LEDs (Ehren)	1	Anda/Ehren, Ehren
4	configure wiring of motors (Anda), write a basic program to control motors (Ehren)	1	Anda, Ehren
5	research how to configure WiFi from ssh and configure network to be stable (static IP)	1-2	Anda/Ehren
6	design basic look of chassis (1 day) (Anda)	1	Anda
7	install camera and programs needed for a low latency live video stream	2-3	Ehren
8	construct chassis and attach parts	1-2	Anda
9	refine weight distribution on chassis and alignment of wheels, and test video and driving	1-2	Anda/Ehren
10	Compete!	1	Anda/Ehren

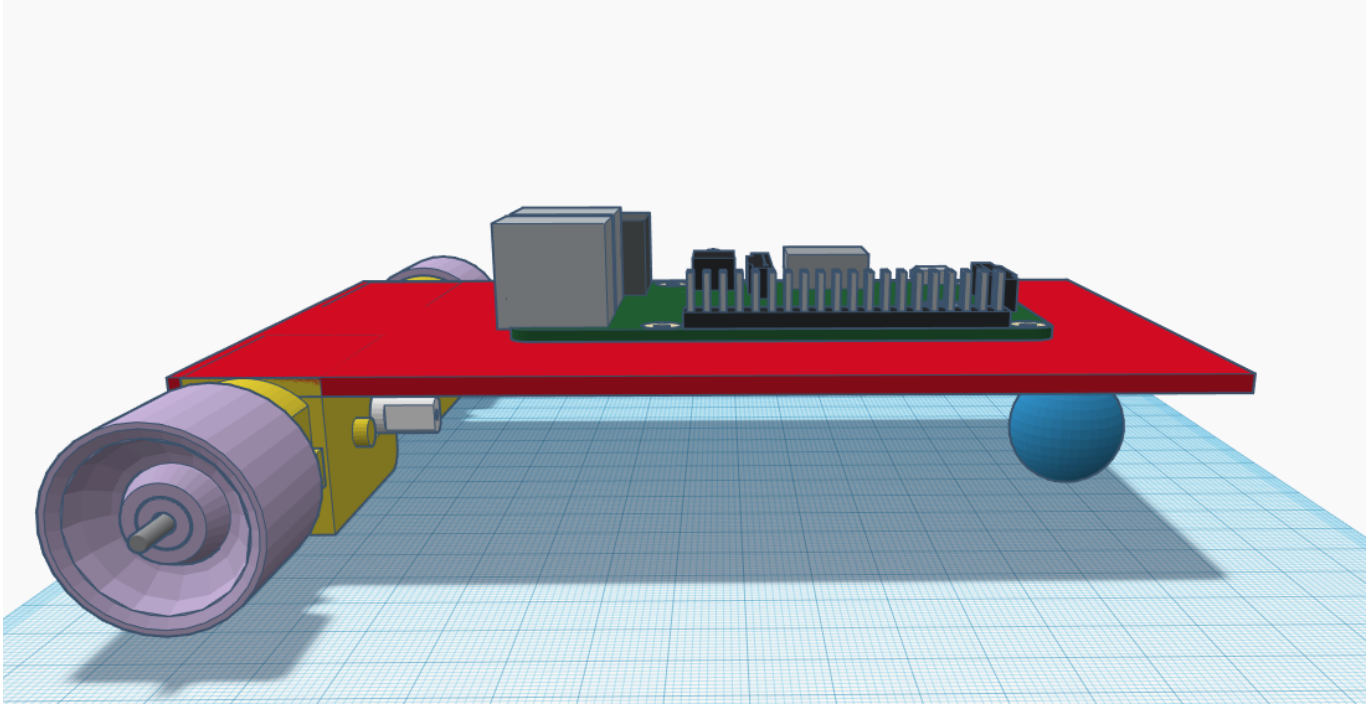
3) Design

Circuit Diagram



Like mentioned above, only 2 motors are being driven by the microcontroller. No resistors/voltage regulators are needed as there is already built-in power protection to the Raspberry Pi (via micro usb port) and the h-bridge. The Pi will eventually be powered via the same 7.2V battery, which is connected to the micro usb port (using a spliced cable).

3D Model of Chassis



All that's needed for this chassis is a flat sheet of stiff material. The motors and free-spinning wheel can be easily attached and will keep the vehicle standing. The microcontroller and dual h-bridge and camera will be mounted on top. A covering for the components is optional. All that is needed is something for the camera to rest against so that it stays upright.

Pseudocode of program

While active:

    If (key press: up)

        Move motors forward

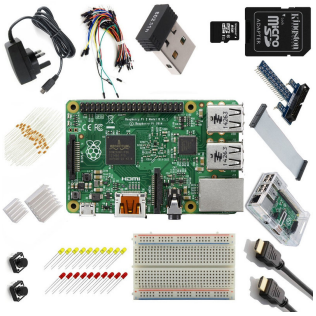
    Else if (key press: down)

    Move motors backward
Else if (key press: left)
    Move left motor backward, right motor forward
Else if (key press: right)
    Move right motor backward, left motor forward
Else
    Don’t move any motors

4) Implementation of Design

<u>Instructions:</u>

Tools: Soldering iron, Screwdriver, Wire Strippers, Hot Glue Gun



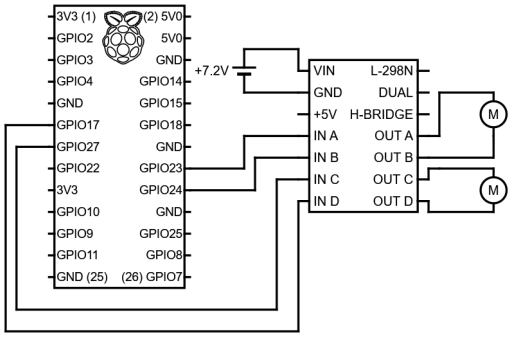
Materials: 2 Alligator clips, Raspberry Pi, 1 pack of Zip Ties, 3 Wheels (one is a 1 ¼" caster wheel), 2 Motors, VEX 7.2V 3000mAh battery, 1 Non-Stick Vegetable Grill, 5 Jumper Cables (4 F-F, 1 M-F), 2 Wooden Dowels, 2 Couplers, Tape, sticks of Hot Glue, Raspberry Pi Camera Module v2, Screws, Nuts, Solder, 1 metre of Single Core Breadboard Wire, Dual H-bridge

Total Cost: ~$193.87

Notes: You will need a network and access to the Wi-Fi network (password and SSID), along with the respective materials (eg. router, modem, ethernet cables)

Steps:

1) Open package and construct the Raspberry Pi
2) Create empty file named "ssh" and place it into the boot folder of the Raspberry Pi
   OR
   If you can boot into the GUI of the Raspberry Pi, you can enable it through launching "Raspberry Pi Configuration" from the "Preferences" menu
   OR
   Go to a terminal window and enter "sudo raspi-config", go to "Interfacing Options", and enable SSH
3) Connect to the Pi using: ssh -X pi@<IP address here>, and entering the password for the pi (default is "raspberry")
4) Install Python 2.7 on Raspberry Pi using *sudo apt-get install python*
5) Connect to the Pi using SSH and create a new python file using: nano filename.py
6) Wire the GPIO ports of the Raspberry PI to the H-Bridge using the Jumper cables, and following the circuit diagram shown here:



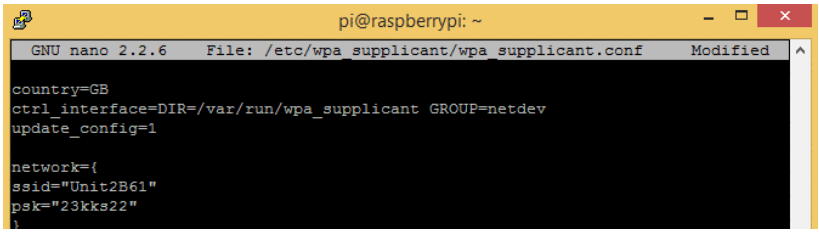
7) Connect 7.2V battery to H-bridge
8) Attach couplers to ends of motors and attach wheels to the couplers so that the motors can control the spinning of the wheels.
9) Wire the motors to the H-bridge

10) Install gpiozero using *sudo apt-get install gpiozero*

11) Create code for controlling motors using gpiozero using Pygame.
<See below ↓>

12) To set up wifi connection, input *sudo nano /etc/wpa_supplicant/wpa_supplicant.conf* into the command prompt.

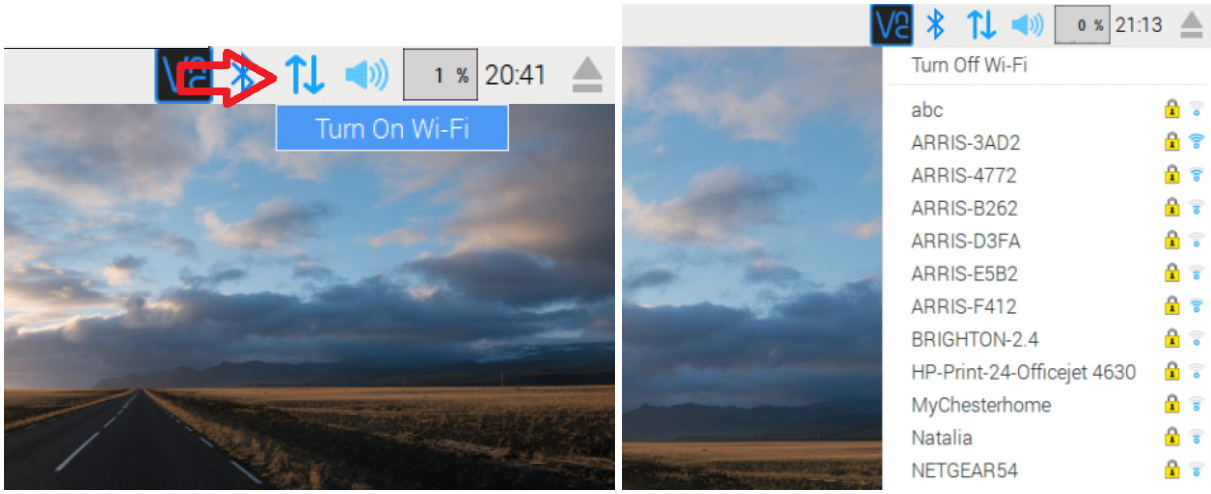
Change the SSID of the command prompt to the Wifi network name and the "psk" to the password as shown in the image to the right ->



Once the information is correct, press CTRL-X, Y, and then enter to save and exit.

Restart the Wi-Fi adapter by inputting *sudo ifdown wlan0* followed by *sudo ifup wlan0* or simply reboot the Pi.

This can also be done on the desktop following these pictures:



You'll also want to make sure that you change your Pi to a static IP address by doing inputting *sudo nano /etc/network/interfaces* and making sure that your information follows this: https://www.swiftstack.com/docs/install/configure_networking.html

13) Connect Raspberry Pi Camera Module v2 to Raspberry Pi and ziptie the camera onto 2 wooden dowels for stability, which are to be glued to the case of the Pi

14) Install netcat and mplayer on both raspberry Pi and Desktop using *sudo apt-get install netcat* and *sudo apt-get install mplayer* respectively to set up video streaming

15) Run on terminal: *nc -l 2222 | mplayer -fps 200 -demuxer h264es -* <from>
16) Run on Raspberry Pi: */opt/vc/bin/raspivid -t 0 -w 300 -h 300 -hf -fps 23 -o - | nc <IP-OF-THE-Desktop> 2222*
to run at the ideal 23 frames per second. Video streaming is now active
17) Attach the coupler with the motor, and the coupler to the wheel
18) Zip-tie the free-turning wheel at the back-centre of the grill, and the other to at the front-left and right. The Pi goes in the centre of the grill, and the battery slightly back, and both get zip-tied. The h-bridge can sit on top of the Pi, can be secured simply with tape.

Software Component:

```
import pygame
from pygame.locals import *
from gpiozero import Robot
from gpiozero import Motor
import time

motorA = Motor(17, 22)  # your gpio pins go here
motorB = Motor(24, 23)

pygame.init()
screener = pygame.display.set_mode((400, 300))
quitng=False

while not quitng:
        for event in pygame.event.get():
        if event.type == pygame.QUIT:
        quit=True
        elif event.type == pygame.KEYDOWN:
        if event.key==K_UP :
                print("FORWARD")
                motorA.forward(1)
                motorB.forward(0.8)

        elif event.key==K_DOWN:
                print("BACKWARD")
                motorA.backward(1)
                motorB.backward(0.8)
        elif event.key==K_RIGHT:
                print("RIGHT")
                motorA.forward(1)
                motorB.backward(1)
```

```
        elif event.key==K_LEFT:
                print("LEFT")
                motorA.backward(1)
                motorB.forward(1)
        elif event.type == pygame.KEYUP:
        motorA.stop()
        motorB.stop()

pygame.dispay.quit()
pygame.quit()
```

RESOURCES

Model idea:
https://3dwarehouse.sketchup.com/model/u43437cf1-1ff7-4fc6-be6a-dab93cd716b2/Peel-p50
SSH: https://www.raspberrypi.org/documentation/remote-access/ssh/
Configuring Wi-Fi on the Raspberry Pi:
http://www.circuitbasics.com/how-to-set-up-wifi-on-the-raspberry-pi-3/
https://raspberrypi.stackexchange.com/questions/34914/wifi-not-working-on-startup
https://www.swiftstack.com/docs/install/configure_networking.html
Changing the name of the Raspberry PI:
https://www.howtogeek.com/167195/how-to-change-your-raspberry-pi-or-other-linux-devices-hostname/
GPIO pin diagram: https://pinout.xyz/#
Example programs of controlling components via GPIO pins on Python:
https://makezine.com/projects/tutorial-raspberry-pi-gpio-pins-and-python/
Set up mplayer streaming:
https://raspberrypi.stackexchange.com/questions/27082/how-to-stream-raspivid-to-linux-and-osx-using-gstreamer-vlc-or-netcat
/opt/vc/bin/raspivid -t 0 -w 300 -h 300 -hf -fps 20 -o - | nc <IP-OF-THE-CLIENT> 2222