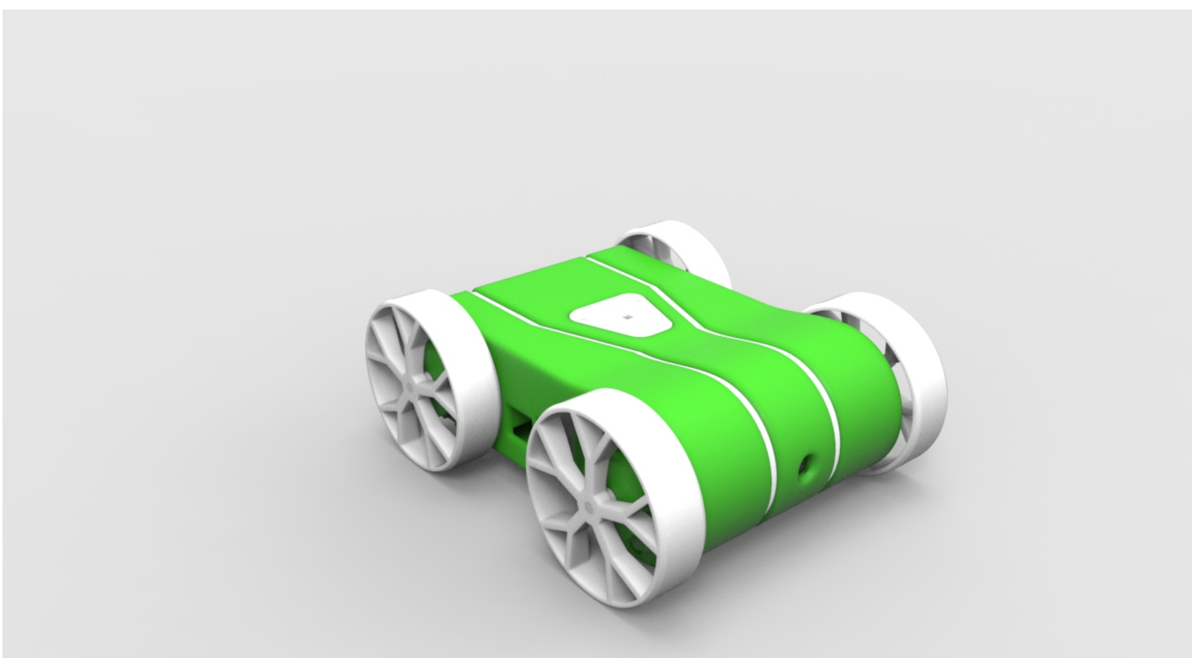
Razbot Build Instructions

Author:
Robbie Edwards



March 29, 2015

Chapter 1

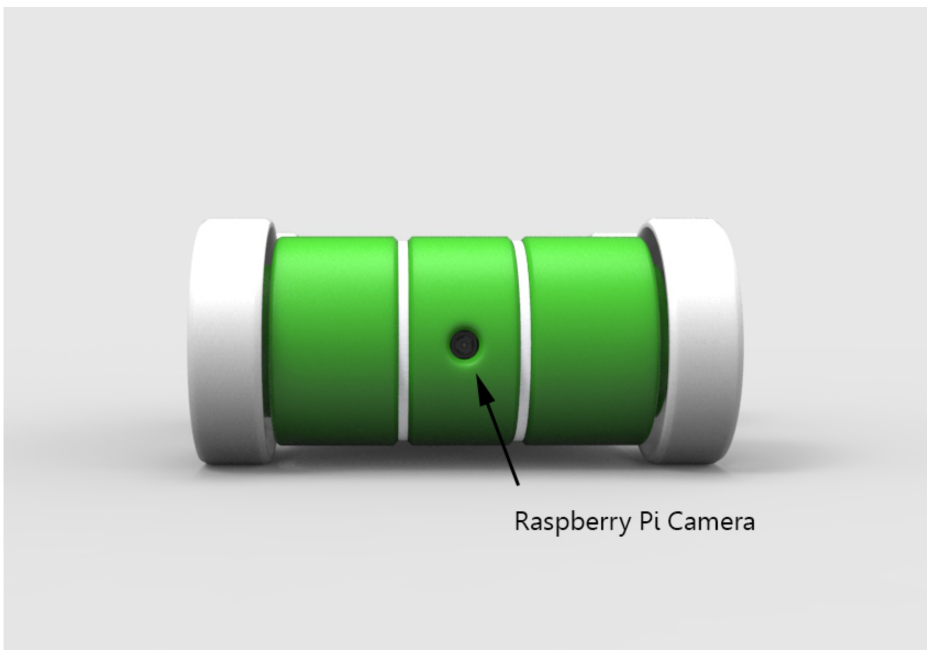
Overview

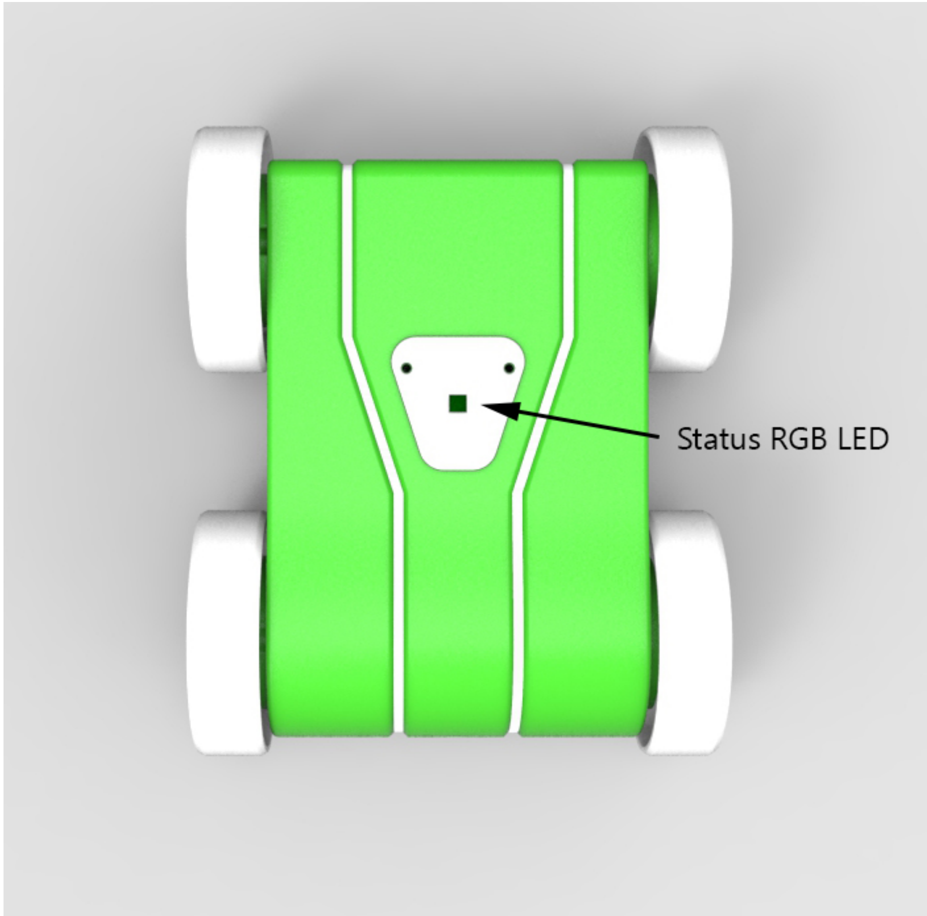
The Razbot was created to provide a low cost Linux/Robot Operating System (ROS) enabled mobile robot platform. The goal was to provide access to high quality software on a low cost platform which can be printed and assembled by the user. It currently runs ROS Indigo and Raspbian Jessie. The Robot Operating System is a BSD licensed meta-operating system which provides a convenient set of tools for developing software for robot applications. It is well documented online and explained in further detail in various books. Many open source ROS packages are freely available to carry out robot related software tasks ranging from interfacing with sensor hardware to carrying out navigation and mapping

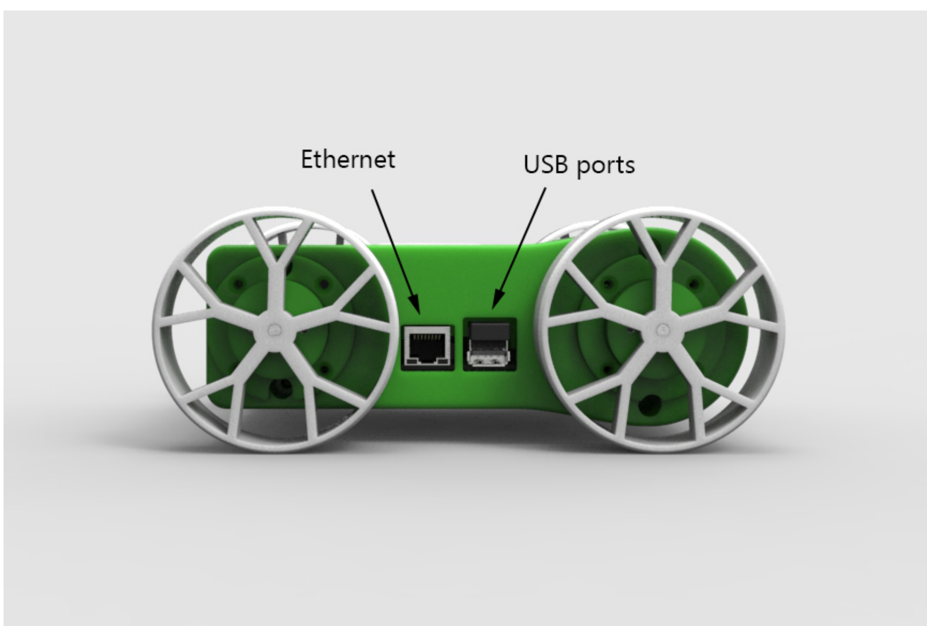
1.1 Features

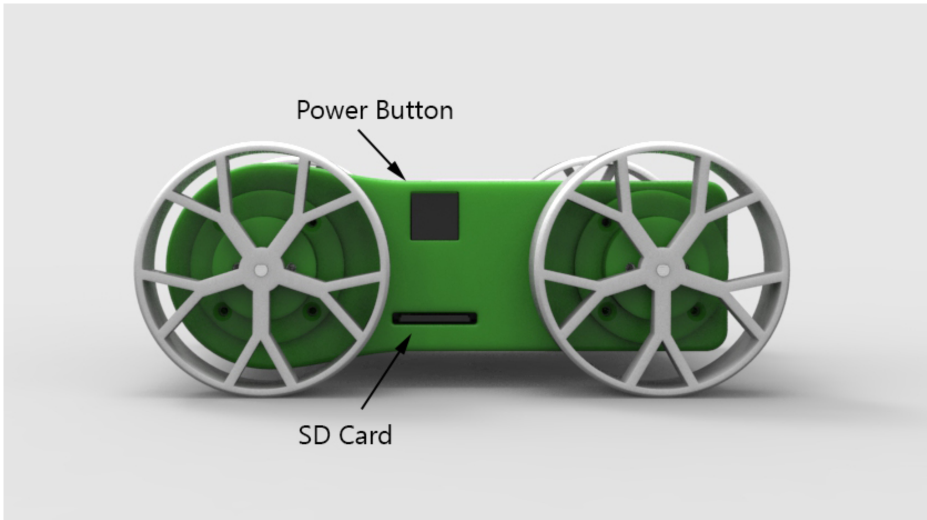
:

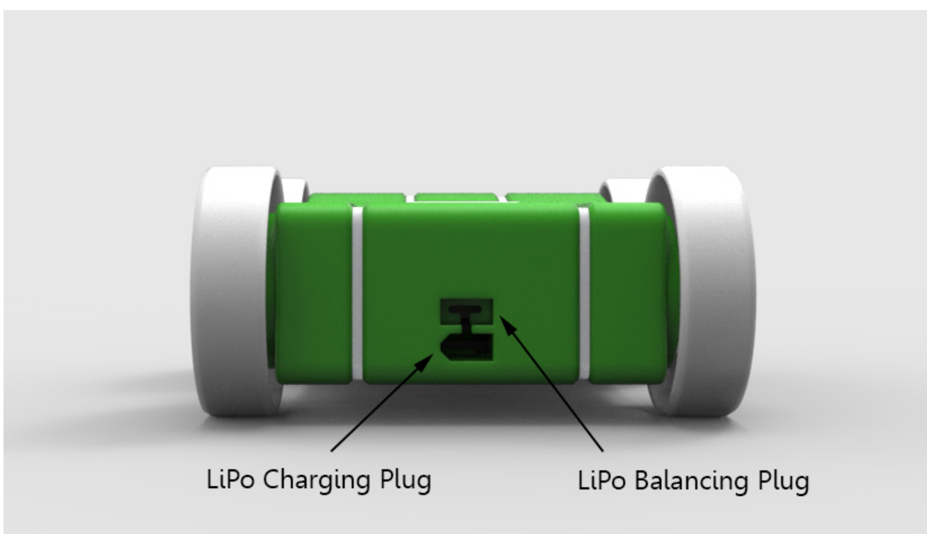
- Raspberry Pi with Raspbian Jessie
- ROS Indigo
- Independent Motor Control of each wheel
- WS2812b Status LED
- Raspberry Pi camera
- Lithium Battery: Turnigy Nanotech 3S 2200mAh,
- 2 hour or greater run time
- USB ports, Wifi
- Logitech F710 Game pad expandable

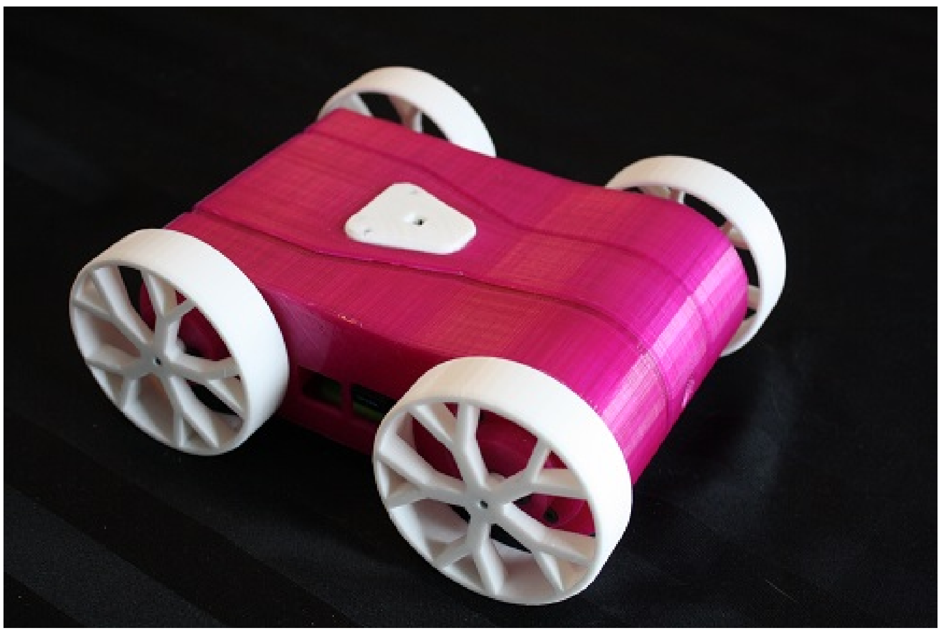

Raspberry Pi Camera


Status RGB LED


Ethernet
USB ports


Power Button
SD Card


LiPo Charging Plug
LiPo Balancing Plug



Chapter 2

Bill of Materials and Required Tools

2.1 Parts and Tools

The following tables provide a list of the parts and tools required to build the rover. The parts are currently sourced from many suppliers resulting in many orders and higher shipping costs. One day, someone can perhaps offer a single source supply for all components necessary to build the rover. Table 2.1 lists the major components of the rover. Table 2.2. The basic tools required are found in Table 2.3. A list of screws required for assembly is found in Table 2.4. The wiring components are in Table 2.5. The instructions for the assembly of the circuit board and a list of the required components are given in a later section.

2.2 3D printing

The Razbot is made up of 8 unique 3D printed parts and a total of 12 3D printed pieces. The table: 2.6 lists each part which is made with a 3D printer. The Razbot has been previously constructed with the printers listed and tested in Table: 2.7. Note that hex cap screws on this platform thread directly into 3D printed plastic. Holes have been sized appropriately to allow clearance for screws and appropriate diameters for threading. Appropriate hole sizing may vary printer to printer and should be checked on initial prints. Repeated screwing and unscrewing or forcing taps or screws may result in damage to 3D printed threaded holes.

Some parts will require support material. All parts have been found to function well when printed at 20% infill. 3D printing settings and parameter decisions will ultimately be left to the operator.

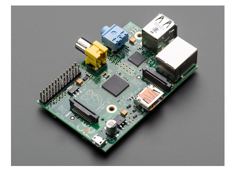
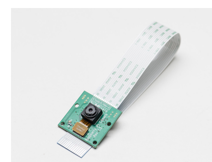
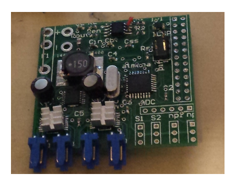
part	supplier	quantity	total cost	image
12V 100rpm 25mm gearmotors	ebay, aliexpress, amazon	4	$40	
Raspberry Pi B, B+, 2	adafruit, element14	1	$40-$60	
Raspberry Pi Wifi USB Adapter	adafruit, element14	1	$12	
Raspberry Pi Camera Board, NoIR	adafruit, element14	1	$30	
SD card		1	$15	NA
Turnigy 2200mAh 3S 20C LiPo Pack	HobbyKing	1	$10	
Razbot Electronics Board	None	1	N/A	
Lithium Polymer B3AC Battery Charger	HobbyKing	1	$8	
Wiring and connectors	NA	1	$15	NA
Screws and bolts	NA	1	$10	NA
3D printed parts	NA	12	$30 filament	NA

Table 2.1: Major Components Bill of Materials

part	supplier	quantity	total cost	image
Logitech F710 Gamepad	Bestbuy, computer stores	1	$45	

Table 2.2: Optional Equipment

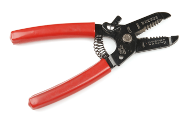
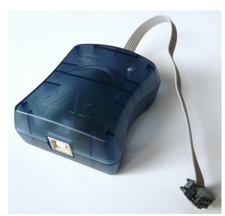
recommended tools	
Hex key set (metric) 1.5,2,2.5,3mm sizes	
Soldering Iron with medium and fine tips	
Hobby Knife	
Wire strippers	
AVR ISP mkii	
optional tools	
M3, M4 drill taps	
Hobby Tweezers	

Table 2.3: Recommended Tools

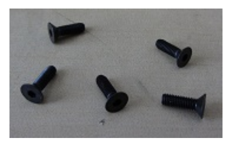
part	Spaenaur part number	quantity	image
Hex Socket Cap Screw M4 x 0.7mm x 25mm Full Thread	367-022	4	
Hex Socket Low Head Cap Screw M3 x 0.5mm x 5mm	366-1071	12	
Hex Socket Flat Head Cap Screw M3 x 0.5mm x 10mm	366-566	18	
Hex Socket Cap Screw M3 x 0.5mm x 10mm	367-B04-1P	5	
Hex Socket Cap Screw M2 x 0.4mm x 6mm	366-663	4	

Table 2.4: Bill of Nuts and Bolts

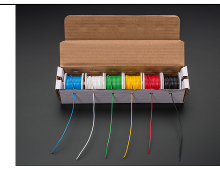
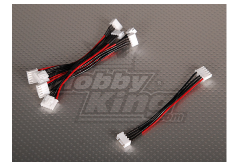
part	supplier	part number	quantity	image
Hook up wire 20-22AWG	various, adafruit	adafruit 1311	various colors	
Female JST-XH ¡-¿ Male Polyquest 3S 10cm (5pcs/bag)	HobbyKing	101b-103a-3s	1	
Female JST battery pigtail 12cm length	HobbyKing	AM-9017A	2	
Male JST battery pigtail 10cm length	HobbyKing	AM-9017B	2	
Power Switch SPST	SW627-ND	digikey	1	
2 pin connector: motor to MCU	Aliexpress, Digikey		4	

Table 2.5: Bill of Wires and Connectors

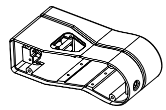
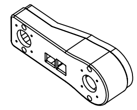
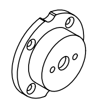
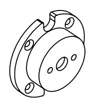
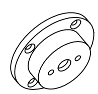
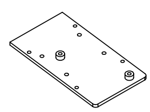
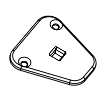
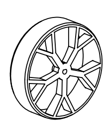
part	quantity	image	support material required
Side Left V01	1		yes
Side Right V01	1		not required
FR motor holder	1		yes
RR motor holder	1		yes
L motor holder	2		yes
PiTrayB	1		not required
Top Plate	1		not required
Wheel	4		not required

Table 2.6: Bill of 3D printed components

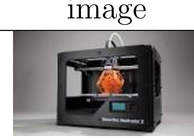
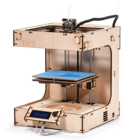
printer	build volume	image	result quality
Makerbot Replicator 2	28.5 L X 15.3 W X 15.5 H cm		excellent $\frac{10}{10}$
Tinkerines Ditto+	21.0 L X 18.5 W X 23.0 H cm		very good $\frac{9}{10}$

Table 2.7: 3D printers used

Chapter 3

Assembly

The rover is easy enough to assemble but it isn't perfect, yet. Many components and wires are packed tightly into a small space. Screws may be difficult to access with hex keys. Once the vehicle has been assembled and is running properly, minimal disassembly and maintenance will be required. Once all the 3D printed components are available with support material removed assembly can begin.

3.1 Razbot MCU board

The Razbot MCU board is easy to make with the proper tools and soldering skills. Some of the components are quite small so proper tools and a good deal of patience is important.

1. The unpopulated PCB board can be ordered from a low cost PCB manufacturer such as SeeedStudio using the exported PCB design files or Gerber files in the electronics folder. The exported files are zipped in razbot_pcb_output.rar.

2. The electronics components can be ordered from a supplier such as Digikey or Newark. The component part numbers and quantities can be found in the excel document razbot BOM.xlsx in the electronics folder.

3. Once the boards and components arrive, the components can be soldered on to the PCB using a fine tipped, temperature controlled soldering iron. Tweezers will greatly assist in placing components. Proper use of flux will ensure easy soldering without jumping component pins. Solder the components on to the board following the designators on the board (also found in the razbot schematics pdf in the electronics folder) and the designator table in the Bill of Materials for the electronics components.
   The power regulator chip RT8250 has a ground pad on the bottom surface of the chip which requires an electrical connection with the board. A hot

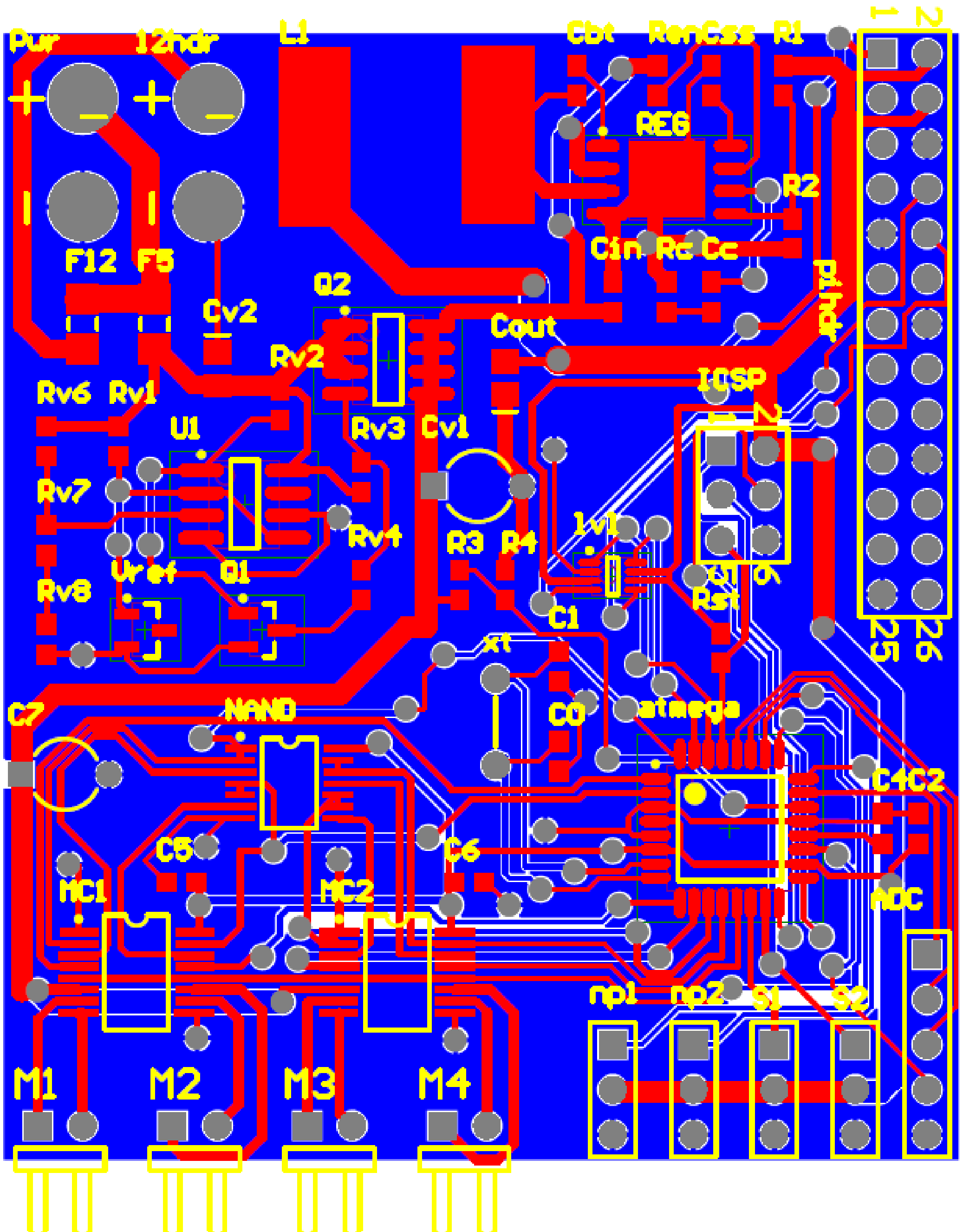


Figure 1: MCU board layout

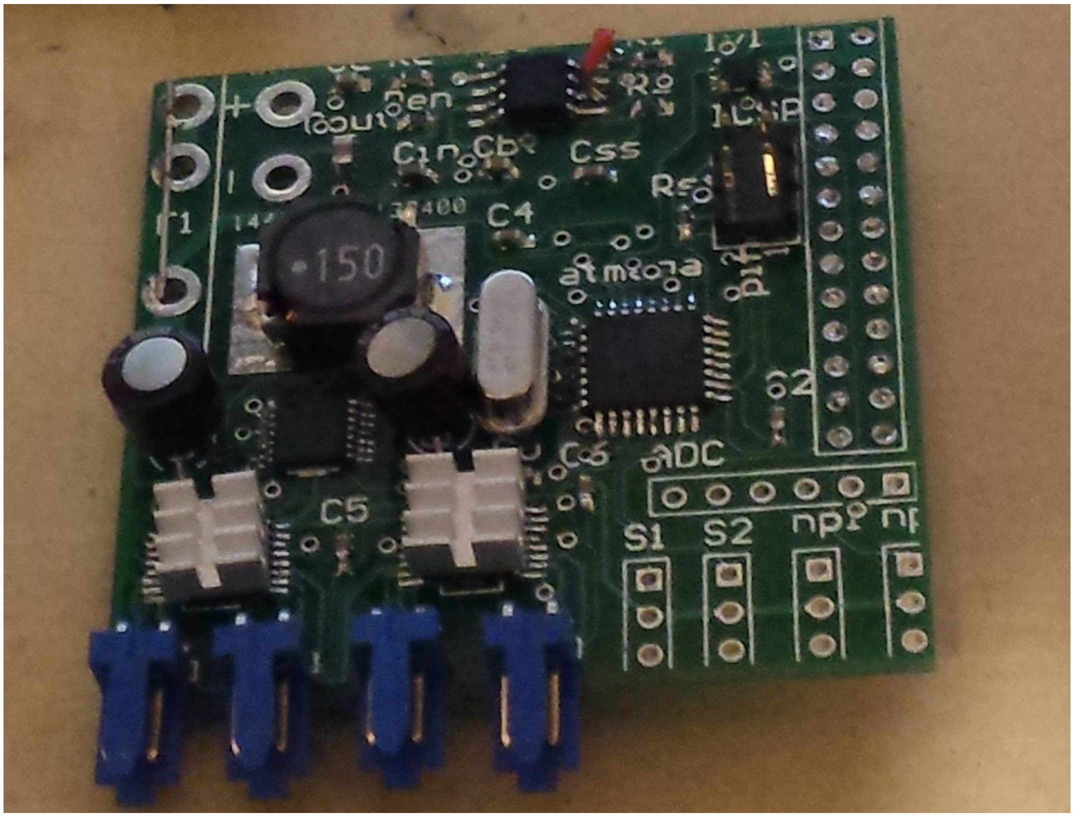


Figure 2: A prototype version of the Razbot PCB

air station can be used to reflow solder on the PCB and attach the chip. Conductive paste may also be an easier solution.

4. Use the AVRISP-mkii programmer with the Arduino IDE to load the Arduino bootloader on to the ATmega328P on the MCU board by connecting to the ICSP header on the board. This will set the proper fuse bits on the ATmega328P for use with Arduino code.

5. Use the programmer again to load on the razbot arduino V02 code located in the razbot Software MCU folder. The ros lib library must be played in your Arduino IDE's libraries folder to compile the code.

MCU Board Alternatives

Ordering and assembling your own PCB may seem like a daunting task. Razbot was originally developed using off the shelf components connected to a Raspberry Pi. This may be appealing to users not wanting to create their own PCB so some

Designator	Digikey PN	Qty /	Desc	Package
NAND	MM74HC132MTCXCT-ND	1	IC NAND 4ch	14-TSSOP
Atmega	2 ATMEGA328P-AURCT-ND	1	IC MCU 8BIT 16KB	32-TQFP
xt	CTX1084-ND	1	XTAL 16MHz 20pf	THRU
REG	1028-1002-1-ND	1	IC REG BUCK SYNC ADJ 3A	8SOP
lvl	296-22862-1-ND	1	IC VOLT-LVL TRANSL BI-DIR	US8
F5,F12	507-1194-1-ND	1	FUSE BOARD MNT 5A 63VAC/VDC 1206	1206
Css,C2,C4,C5,C6		5	CAP CER 0.1uF 25V	603
Cout		1	CAP CER 22uF 10V	805
C7,Cv1		2	CAR EL 22+uF 50V	RAD
Cv2		0	unpopulated	1206
R3		1	RES 150K 1/10W 0603	603
R4,Ren, Rv6		2	RES 100K 1/10W 0603	603
R2, Rst, Rv1, Rv2, Rv4,		5	RES 10K 1/8W	603
Rv3		1	RES 50K 1/8W	603
Rv7		1	RES 27K 1/8W	603
Rv8		1	RES 6.8K 1/8W	603
R1		1	RES 45.3K 1/10W	603
Rc		1	RES 13K	603
ISCP		1	6 POS VERT HEADER	0.1' hdr
L1	SRP1040-150MCT-ND	1	FIXED IND 15UH 5A 57 MOHM SMD	
Cc	399-7913-1-ND	1	CAP CER 3.9nF	603
Cin	490-7198-1-ND	1	CAP CER 10UF 16V	603
Cbt		1	CAP CER 10nF	603
MC1 MC2	TB6552FNG(O8EL)CT-ND	2	800mA Motor Driver 2CH	16SSOP
C0, C1	1276-1187-1-ND	2	20pF 0603	603
pihdr		1	13x2 0.1	0.1' hdr
PWR	SW627-ND	1	SWITCH PUSHBUTTON SPST 3A 125V	
U1	296-12848-1-ND	1	dual comparator	SOIC8
Q1	568-6492-1-ND	1	TRANS NPN 140V 0.3A SOT23	SOT23
Vref	LM4040AIM3-2.5/NOPBCT-ND	1	2.5voltage ref	SOT23-3
Q2	785-1022-1-ND	1	MOSFET P-CH 30V 12A 8SOIC	SOIC8

Figure 3: MCU electronics board bill of materials

guidelines are presented here. Note that the increase in wiring can result in larger potential for error and possibly more work than building the PCB in the first place.

Razbot was initially build with an Arduino Micro mounted to the GPIO pins on the Raspberry Pi connected to the serial port. This could have also been connected over USB. It was used to control an L298 2 channel motor driver as well as other servos. A Turnigy 5V 5A switching regulator was used to supply 5V to the robot from the 11.1V Lithium battery. Users may be able to do away with the Arduino Micro and simply drive the L298 motor controller with the Raspberry Pi GPIO.

Squeezing components into the small Razbot chassis may prove difficult. The Razbot has 4 motors, thus the 2 motors on each side of the vehicle must be wired in parallel removing potential for future omnidirectional upgrades.

The components for the MCU board alternative solution are listed in 3.1.

part	supplier	cost	image
Arduino Micro	Adafruit	$22	
L298 2 ch motor controller	Aliexpress, Ebay, Seeedstudio	$4-20	s
Turnigy 5V 5A Switching Regulator	HobbyKing	$5	

Table 3.1: Parts for MCU alternative robot

3.2 Electronics Assembly and Test

1. Download the latest Razbot SD card image from SD card image on Google Drive. Extract it with Winrar and use a program such as Win32diskimager to mount it on to an 8GB SD card. Later when the image is booted, raspi-config can be used to expand the file system to fill the full 8GB of storage.
   Raspberry Pi B only!
   The current SD card image does not support the Raspberry Pi 2. It can be easily updated to run on the RPi2 by updating the firmware following instructions on the internet such as (**INSERT LINK**). The next image update will be RPi 2 compatible.

2. Plug the Raspberry Pi Camera and Wifi dongle into the Pi and insert the SD card for now. Connect the unit to an Ethernet port on your router. Use conventional wall power to power the unit.

   a) Using your router's user interface, locate the IP address of the Raspberry Pi. The initial hostname will appear as ¡color¿pibot. This can be changed with raspi-config at a later time.

   b) SSH into the Raspberry Pi using the IP address from step 2.1. In windows this can be done with a free program such as Putty. Username

and password as usual:
username: pi
password: raspberry

c) Verify that ROS is running (robot upstart is used to start ROS and required nodes at boot up).

```
$rostopic list
```

d) Verify the mjpg-streamer software has started and is streaming video from the camera. Point a web-browser to: Your bot's IP address:8080 Configure the Wifi on the Pi to automatically connect to your network using wicd-curses.

```
$sudo wicd-curses
```

3. Attach the wire ends of a male JST pigtail connector to the Lithium Polymer battery.

   a) Using a hobby knife, remove a small amount of shrink wrap from the battery ground (black) wire connection to the yellow XT-60 connector.
   b) Solder on the ground (black) wire to the JST connector
   c) repeat this process with the positive (red) connection on the yellow XT-60 connector
   d) Solder on the positive (red) lead from the JST connector

WARNING

Take care not to short the battery terminals together while making this modification. Do not let the knife short the terminals while cutting!

WARNING

Take care to check the polarity of the connection, red to red, black to black.

4. Mount the MCU electronics board on top of the Raspberry-Pi. Ensure the GPIO connectors line up with the female header on the MCU board.

5. Wire connectors to the electric motors

a) Cut 10cm lengths of two different colors of 20-22awg wire for each of the motors (4x color 1, 4x color 2)

b) Solder one end of each wire to the motor terminals, ensure consistency of polarity. Check for the red dot on the motor by the terminal to find the positive connection.

c) Build the 2pin blue connector (PART NUMBERS) on the other ends of the wires for each motor. Again, ensure consistency of polarity. Place positive on the right and side of the connector (connector slot faces upwards)

6. Attach the motors to the MCU board according to the diagram in **??**.

7. Power up the system by connecting the Battery JST connector to the JST connector on the MCU board. The LED will illuminate in blue. The Pi should power on and boot as usual.

8. Verify the system has booted normally, SSH into the Pi and verify ROS and the camera have started.

9. Verify the motors are operational by manually publishing their control ROS-topic

   a) Set all motors to full forward speed

   ```
   $ rostopic pub /piBot_motors std_msgs/ColorRGBA 255 255 255 255
   ```

   b) Set all motors to full reverse speed

   ```
   $ rostopic pub /piBot_motors std_msgs/ColorRGBA 0 0 0 0
   ```

   c) Set all motors to zero speed

   ```
   $ rostopic pub /piBot_motors std_msgs/ColorRGBA 127 127 127 127
   ```

   Consider the location of each motor on the vehicle when noting the direction of travel of each motor.

10. Verify the LED is responsive by publishing color topics. Set the light to red+green

```
$ rostopic pub /piBot_led std_msgs/ColorRGBA 180 120 0 0
```

11. Verify the battery voltage is being monitored properly.

```
$ rostopic echo /piBot_bat
```

The electronics bring-up is complete

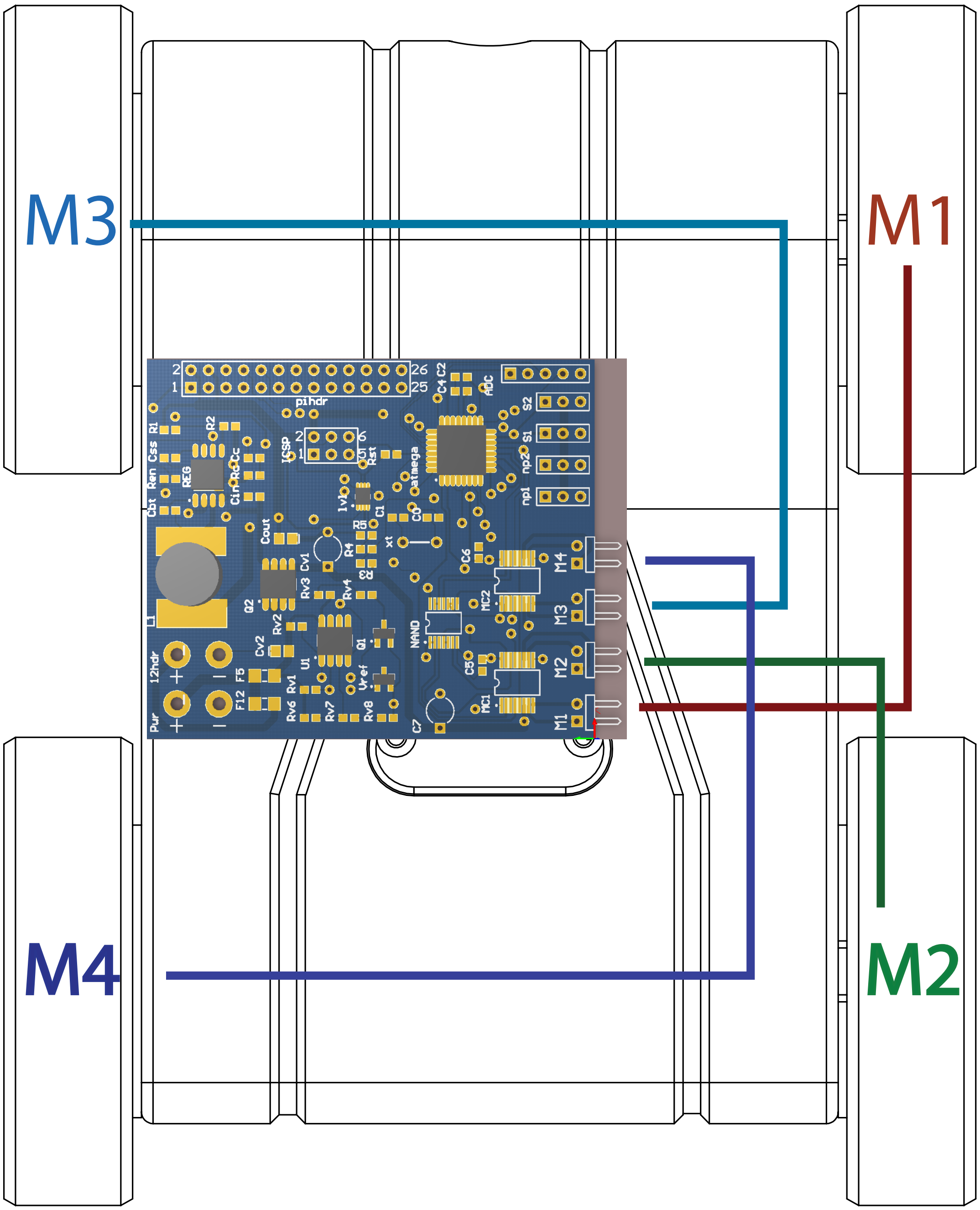


Figure 4: Motor wiring connections

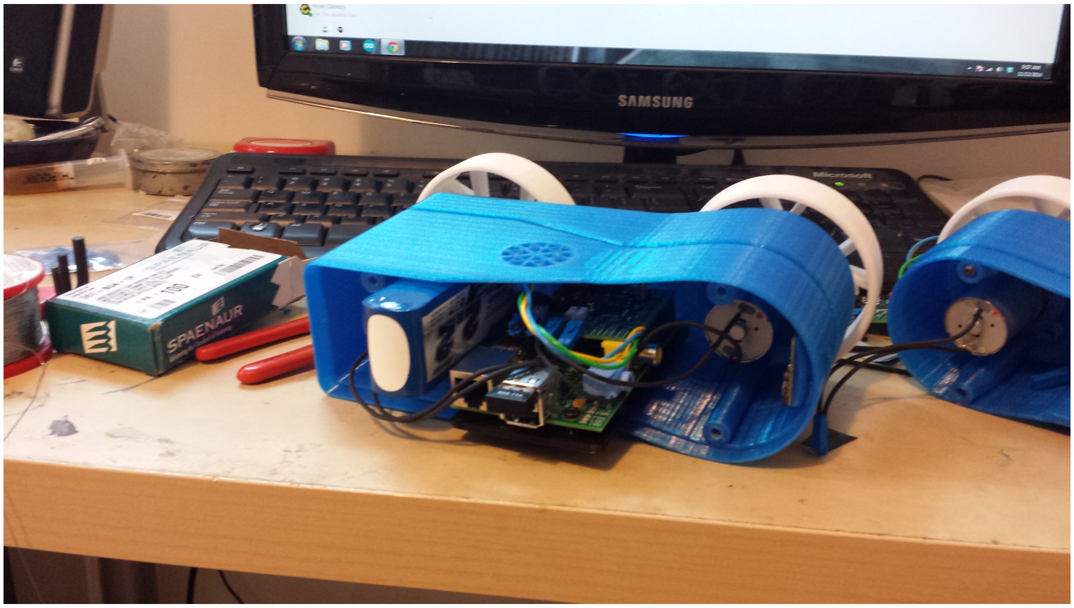
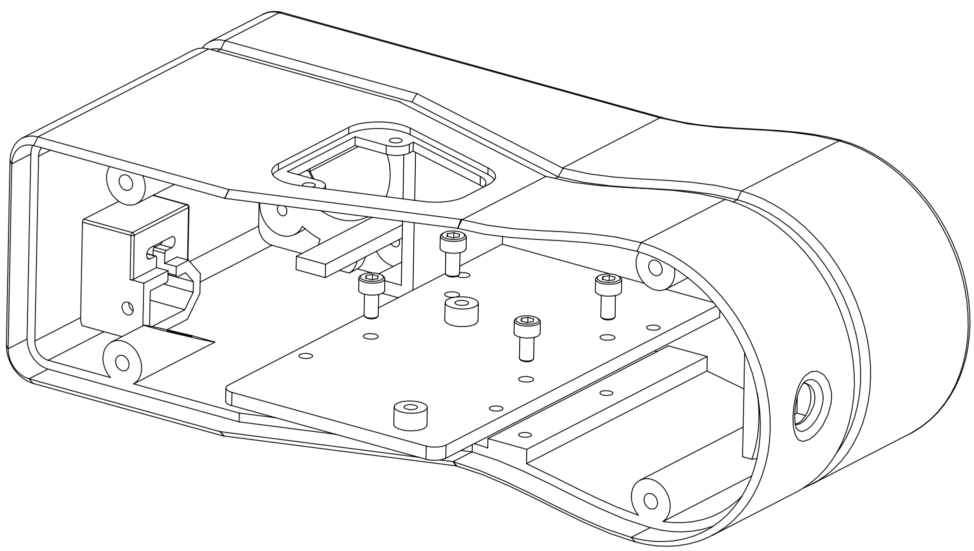


Figure 5: Electronics assembled, mounted in chassis

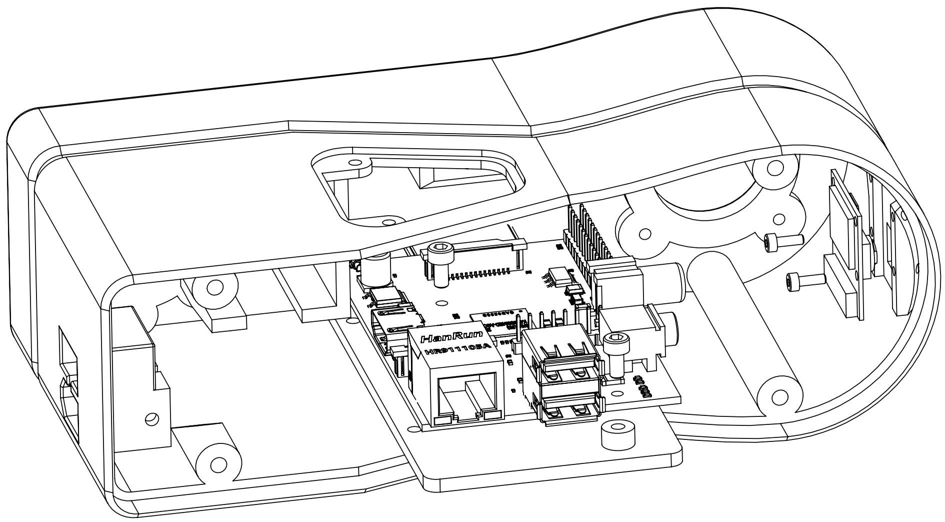
3.3 Mechanical Assembly

1. Attach the Pi-Tray to the left side of the chassis (PiTrayB to Side Left V01) using 2-4 Hex Socket Low Head Cap Screw M3 x 0.5mm x 5mm screws.



2. Mount the Pi-Camera to the inside of Side Left V01 using M2 Hex Cap Screws. Leave the ribbon cable attached to the camera board but not to the Pi during this process.

3. Install the Pi on the Pi mounting tray using M3 x 0.5 x 6mm Low Head Hex



Cap screws [h]

4. Attach the JST connectors to the power switch as shown in Figure ¡MAKE A FIGURE¿. Positive power should run though the switch.

5. Install the switch into the slot in Side Left V01

6. Install the battery into Side Left V01. Before fully inserting the battery, run the balancing plug out the rear connection socket of the robot. This can be difficult, take care not to damage the wires. Next run the XT 60 out the corresponding rear connection socket. Run the JST back in the robot towards the Pi mounting tray. Fully insert the battery.

    a) Lock the XT-60 connector in place with the M3 x 0.5 x 6mm Low Head Hex Cap set-screw

7. Connect the battery JST to the switch.

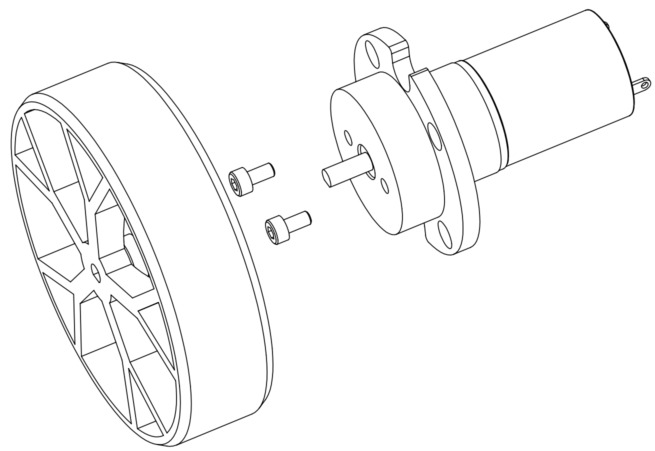
8. Connect the switched JST power to the MCU board. Test the functionality by turning it on. Ensure the electronics are not in contact with metal surface or other potential shorts during testing. Remove the MCU Board.

9. Connect the Pi camera ribbon cable to the Pi.

10. Push the LED and wiring through the hole in Side Left V01. Attach the LED to the Top Plate using M2 screws if possible or just hot glue. Screw the Top plate down to Side Left V01 using M3 x 0.5 x 10mm Flat Head Hex Cap screws.

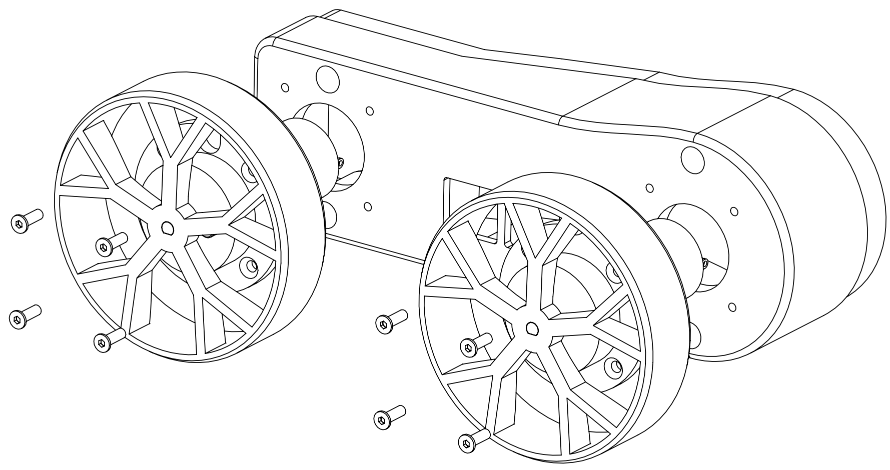
11. Install the motors in each of the motor holders using M3 x 0.5 x 6mm Low Head Hex Cap screws.

12. Push the wheels on to the motor shafts. This may require a tap from a hammer. Use M3 x 0.5 x 10mm set screws in the wheels if required.



[h]

13. Install the motor holder assemblies into the vehicle chassis sides (Side Left V01, Side Right V01) using the M3 x 0.5 x 10mm flat head cap screws.



[h]

14. Connect the motor wires to the MCU board in proper order shown in Figure XXX. You may wish to verify proper operation of the electronics one more time at this points.

15. Push the two sides of the vehicle together while taking care not to tangle or stress wiring. This may take patience.

16. Use four M4 x 0.7 x 25mm bolts to attach the chassis sides together.

Mechanical assembly is complete.

Chapter 4

Operation

The Razbot is easy to use! Simply turn it on and begin enjoying your ROS-robot experience. Razbot uses the ROS robot upstart package to launch its ROS nodes.

1. Press the Red power button on the left side of robot. The status LED will illuminate Blue

2. Give the robot some time to boot up. The red camera light will come on and shine through the 3D printed chassis at the front when launch is complete. Joystick control will also be available if a USB gamepad joystick is plugged in.

3. Locate your robot's IP address with your router. Log into your router, typically located at 192.168.1.1 and find the IP address assigned to the razbot hostname. In some case you will be able to use the hostname (razbot) in place of the IP address

4. Navigate to the server hosted by mjpg-streamer at (robot-IP-address):8080 or (robot-hostname):8080.

5. To control the robot using a browser and view the video feed, navigate to (robot-IP-address):8080/ROScontrol.html. (Currently Unfinished but works)

4.1 ROS Operation

The Razbot will start some of it's own software when the Raspberry Pi boots up. This will allow it to connect to a USB joystick and drive the rover as well as view the camera and accept commands from a web browser.

To view the ROS nodes which are run on start-up type:

```
$rosnode list
```

The following nodes should be enumerated

- pibot - subscribes to the joy topic and publishes piBot motors, piBot light
- joy - publishes the joy topic using a Linux Joystick device
- rosbridge_server - opens a websocket to receive topics from ROSlibjs
- rosserial_python - interface with the MCU board to transmit topics

A script is also used to start Mjpg Streamer with the Pi Camera

To begin working in more depth with the robot and developing your own software, you can SSH into the robot to see what is really going on and run your own code. Use a program such as Putty in Windows to connect to the robot's IP address (determined from your router's DHCP client list). In some cases using the robot's hostname (razbot) will also work. From Linux, simply open a terminal and type:

```
ssh pi@<IP address or hostname>
```

```
ssh pi@192.168.1.101
```

```
ssh pi@razbot
```

The following ROS topics will be visible after start-up by running:

```
$rostopic list
```

- piBot_motors
- piBot_light
- piBot_bat
- joy

To view the data being transmitted in a given topic type:

```
$rostopic echo joy
```

Move the joystick around and watch the data update. Try this for different topics.

software disclaimer

With the electrical and mechanical design now in a somewhat stable state, development work will now focus on updating some of the software on the robot. There may be frequent changes to the razbot packages and the topics and instructions listed above may not always be recent or accurate. Some of the nodes will be reimplemented in a more *standard* fashion of the Robot Operating System. For example: the pibot package will likely be reimplemented using a standard diff-drive controller outputting the cmd vel topic. This will be subsequently used by a razbot base package and a different message type to control the motors.

4.2 Simulation and Visualization

Paul Bovbel has created a URDF (Universal Robot Description File) and a ROS package which allows the Razbot to be simulated in the Gazebo physics simulation environment. Gazebo is a popular tool for roboticist to simulate their platforms, software and sensor arrangements in a virtual environment.

Paul has created tutorials and files which will guide users through setting up the simulation. These are partially available for now (awaiting permission) on Github.

Chapter 5

Future Work

There remains much to do and much to work on with the Razbot, the following are a few ideas.

5.1 software

1. update ROS packages to standard format, consistent with gazebo simulation
2. rename ROS topics and packages from pibot to razbot
3. update RPi SD image to be compatible with RPi2
4. Provide a demonstration or control interface with RVIZ
5. Continue development of the Web based camera and control interface. Improve latency
6. update the MCU firmware to accommodate all of the above

5.2 electrical hardware

1. integrate one or two optical flow sensors to provide odometry to the robot. Update the MCU board to support the sensors. Attempt to do this experimentally on an existing Razbot with and additional Arduino Micro connected over USB.
2. Provide breakouts for additional power and communications on the robot chassis. Break-out the fused 12V power on the robot chassis and make the servo PWM outputs available on the chassis and as a ROStopic for control
3. Add an inertial measurement unit such as the MPU-6000 to the MCU board if cost feasible on the next revision.

5.3 mechanical hardware

1. Develop or adapt 3D printed Mecanum wheels for the razbot. Implement the appropriate control hardware.
2. Improve assembly ergonomics, access to screws required for assembly, reliability of screw hardware in 3D printed holes
3. Create new industrial design options so users can chose the look of their Razbot
4. Develop better sensor and additional hardware capabilities
5. Modify the wheel design and material to improve *off-road* and carpet capability

5.4 general robotics

1. Borrow a laser scanner such as an RP-600, SICK TIM 551 or Hokuyo UST 10LX and attempt to run SLAM (Simultaneous Localization and Mapping) as well as autonomy on the robot.
2. Do some form of computer vision or visual SLAM with the RPi camera.
3. offload intensive processing to a server from the RPi.

5.5 Final Notes

If you are attempting to build your own Razbot or are even thinking of building your own system, please drop me a line! Rob.a.Edwards@gmail.com.

Don't hesitate to send any suggestions for improvements, new ideas, questions, requests for support or even complaints. If the documentation is insufficient, let me know or post an issue to the github with where you are having trouble building your bot.

If you do anything cool with your bot, please share! Send pictures, videos, software.

Good luck! There is still a long road ahead to get razbot to the point where it may be useful as an education tool and for roboticists without the fair amount of multidisciplinary design and assembly skills required to get a robot working from the ground up. It may never get to that point, but with your help it will be one step closer.

Appendix A

Special Thanks

Paul Bovbel Created the razbot tutorials for learning about URDFs (Universal Robot Description Files) and the Gazebo Simulation Environment
Ilia Baranov Provided useful advice throughout development as well as sanity checks and emotional support in the form of an unstoppable Starcraft opponent for most in the office.
Mike Palmer Provided feedback on electronics board designs and contributed the low voltage protection circuit to the design.
Mike Purvis Provded advice which helped significantly with getting robot upstart to run.
Martin Hummel Provided a full ROS Indigo built on Raspbian Jessie image on answers.ros.org

Thank you all, this would not be possible without your input and support.