Tracked Robot Design

16 Feb 2020

Nathan Lucas
Wayne State University
Electrical and Computer Engineering

1. Hardware

A tracked vehicle configuration with differential steering was selected for maneuverability within a relatively small test environment, and consistent turning clearances afforded by aligning the yaw axis with the geometric center of the platform. In order to reduce hardware variability and simplify maintenance, commercially available components were integrated to the maximum extent feasible and drill hole pattern templates were used where fabrication was necessary.

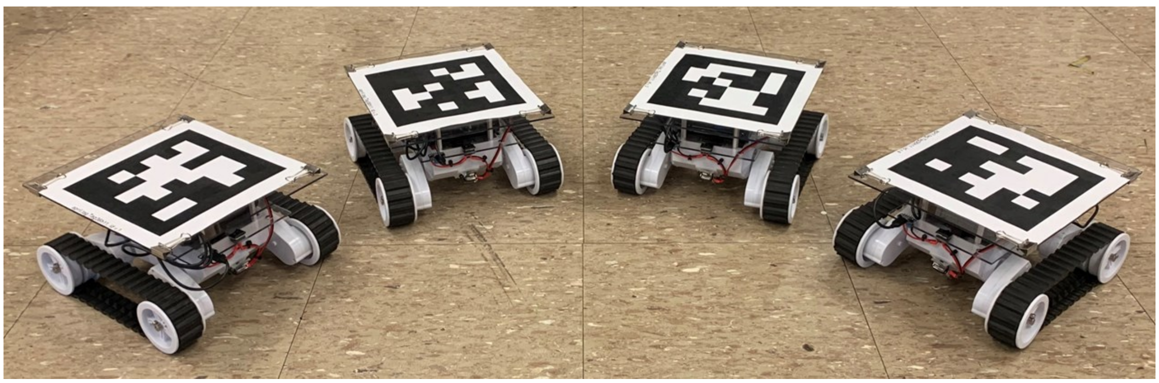

Figure 1-1: Tracked robots.
The tracked chassis provided a stable platform for the onboard controller, motor driver, battery packs, and other electronics. A unique fiducial on each robot was used for video-based localization and tracking.

Figure 1-2 provides an overview of the major components. A Dagu Rover 5 tracked chassis was used to provide a relatively robust, stable platform with a track and wheelbase of approximately 230 mm (9 in). The chassis came equipped with left and right motor assemblies with integrated gearboxes, wheels, rubber tracks, and electrical wiring. A Raspberry Pi 2 Model B (Raspberry Pi Foundation, http://www.raspberrypi.org/) served as the robot controller, with a DRV8835 dual motor driver shield installed on the general-purpose input/output (GPIO) header. An Edimax EW-7811Un USB Wi-Fi module connected the robot to the test platform network.

Two lithium-ion battery packs provided power to the robot. A 5.1 Ah pack powered the controller and other digital electronics. A 6.7 Ah pack supplied power for the motors via a USB Micro-B breakout board attached to the lower mounting plate. A single-throw toggle switch was installed in the lower plate between the USB breakout board and a 100 mA USB LED lamp mounted to the front of the robot. The lamp was inserted into the motor power circuit to prevent the battery pack from shutting down due to low current conditions, and also provided a visual indicator of the robot’s orientation. A complete bill of materials with quantities and dimensions can be found in Table 1-1.

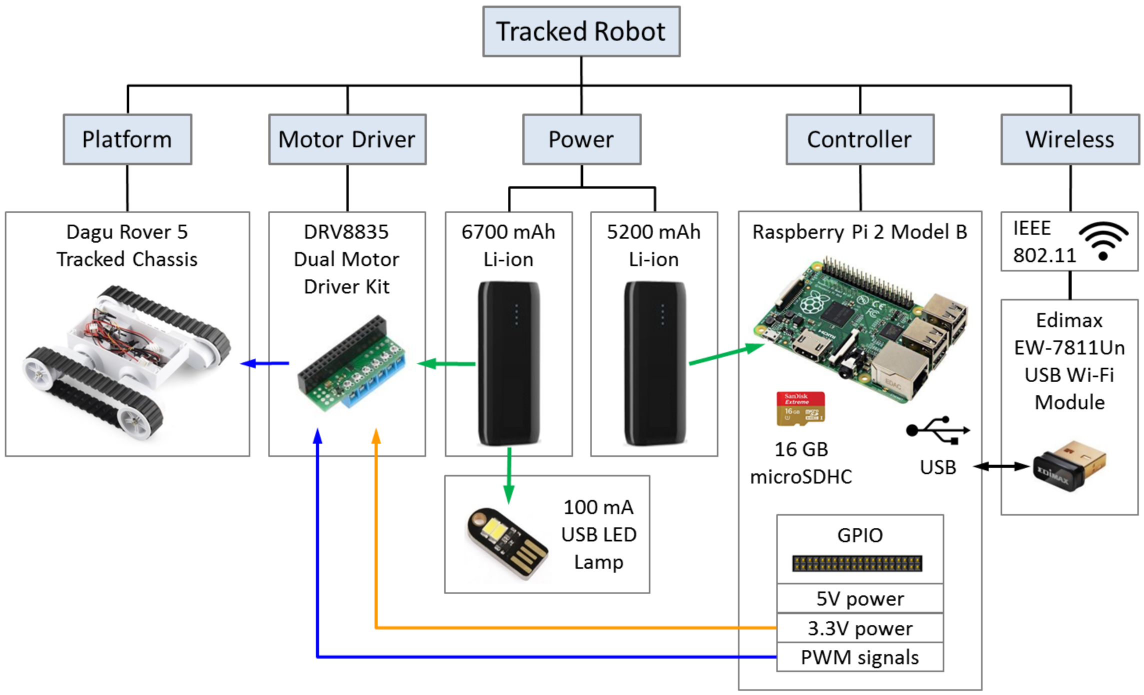

Figure 1-2: Tracked robot major hardware components.
The motor driver was installed on the general-purpose input/output (GPIO) header of the controller. Not shown: Electrical connectors, cables, toggle switch, polycarbonate mounting plates, fasteners, and other mounting hardware.

Table 1-1: Tracked robot hardware bill of materials

Component	Description	Qty
battery pack, digital power	Anker Astro E1 5200 mAh, A1211012	1
battery pack, motor power	Anker Astro E1 6700 mAh, A1211015	1
cable, power	USB A plug, USB Micro-B plug	2
chassis	Dagu Rover 5	1
connector, driver power, contact	18-22AWG, TE 1123721-2	2
connector, driver power, header	2 circuit, TE 1744048-2	1
connector, driver power, housing	2 circuit, TE 1744036-2	1
connector, motor, header	4 circuit, TE 1744048-4	1
connector, motor, housing	housing, 4 circuit, TE 1744036-4	1
controller board	Raspberry Pi 2 Model B	1
controller memory	16GB microSDHC	1
controller module, motor driver	DRV8835 Dual Motor Driver	1
controller module, wireless	Edimax EW-7811Un USB Wi-Fi	1
fiducial marker clamp	steel binder clip, 19 mm (3/4 in), silver finish	4
fiducial marker tag	AprilTag, black on white paper	1
hex hut, controller mount	N2.5-0.45, 2.1 mm thick, nylon	4
hex nut, controller mount	M2.5-0.45, steel	4
LED lamp board	100 mA, motor power circuit	1
LED lamp receptacle	USB A, motor power circuit	1
motor power receptacle	USB Micro-B breakout board	1
motor power switch, rocker, SPST	AC 250V 3A 2 pin on/off I/O SPST snap-in	1
motor power wire	2-conductor, 20 AWG, black-red	AR
mounting plate, controller	polycarbonate sheet, 0.093 × 8 × 5 in	1
mounting plate, fiducial	polycarbonate sheet, 0.093 × 10 × 8 in	1
screw, controller mount	M2.5-0.45 × 5 mm, pan head, nylon 6/6	4
screw, mounting plate	#6-32 × 3/8-in, flat head, zinc plated	4
standoff, controller mount	M2.5-0.45 × 6 mm Female × 6 mm Male, Nylon	4
standoff, mounting plate	#6-32 × 1.5-in, male/female, aluminum	4

2. Electrical Design

Figure 2-1 contains a schematic diagram of the motor driver circuit. The motor driver board could optionally supply power to the controller via the GPIO header, but this feature was not used. The controller was instead powered by a separate battery pack, which also provided power to the H-bridge integrated circuit via the GPIO’s regulated 3.3 V pin.

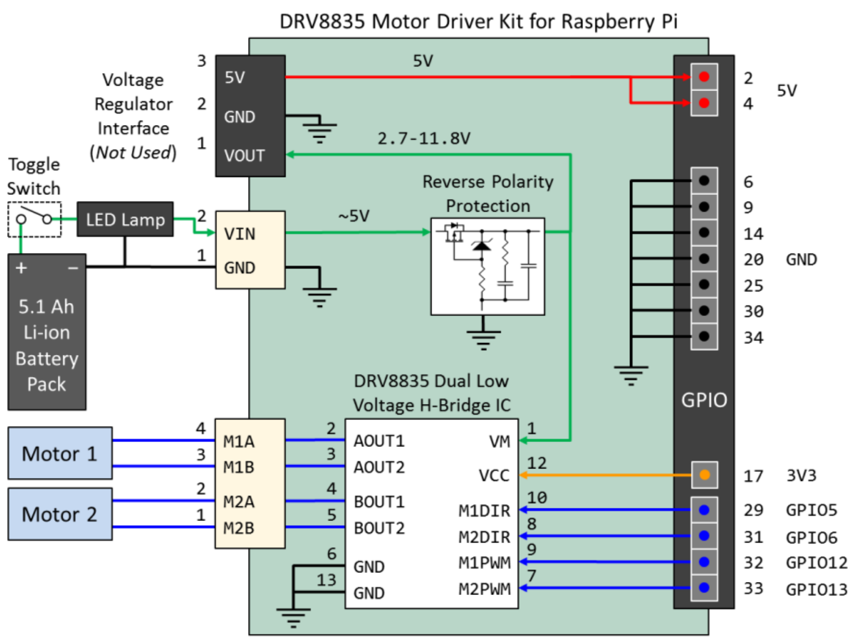

Figure 2-1: Schematic diagram of the motor driver circuit.

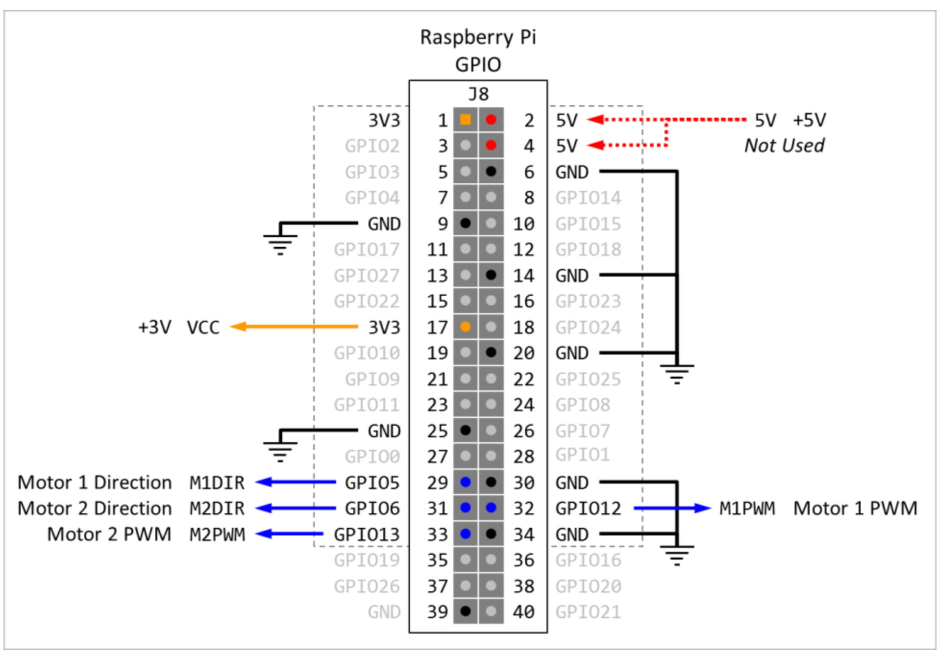

Figure 2-2: Motor driver interface.

3. Physical Integration

Two mounting plates were fabricated from 2.36 mm (0.093 in) thick polycarbonate sheets to integrate the components. Four 3.8 cm (1.5 in) aluminum standoffs were used to attached the upper and lower mounting plates to the chassis. The controller was mounted between the plates on four short nylon standoffs attached to the lower plate. Four 1.9 cm (3/4 in) wide binder clips were used to clamp a unique AprilTag fiducial to the top surface of the upper mounting plate.

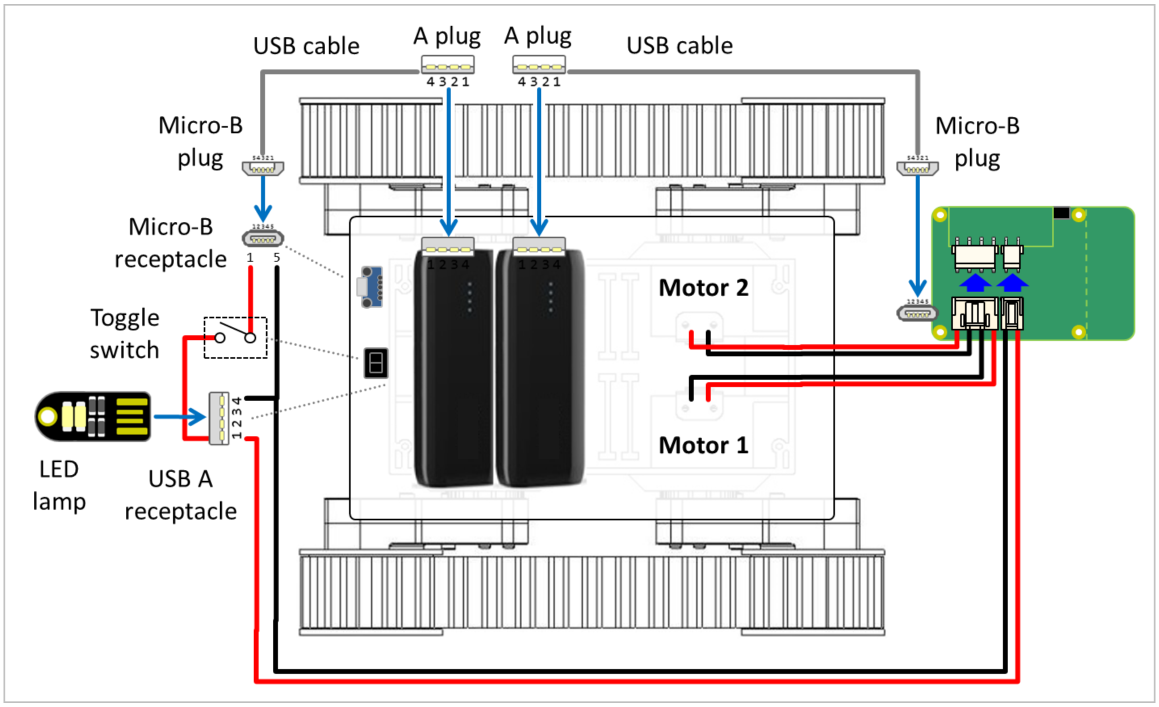

Figure 3-1: Electrical power circuit.

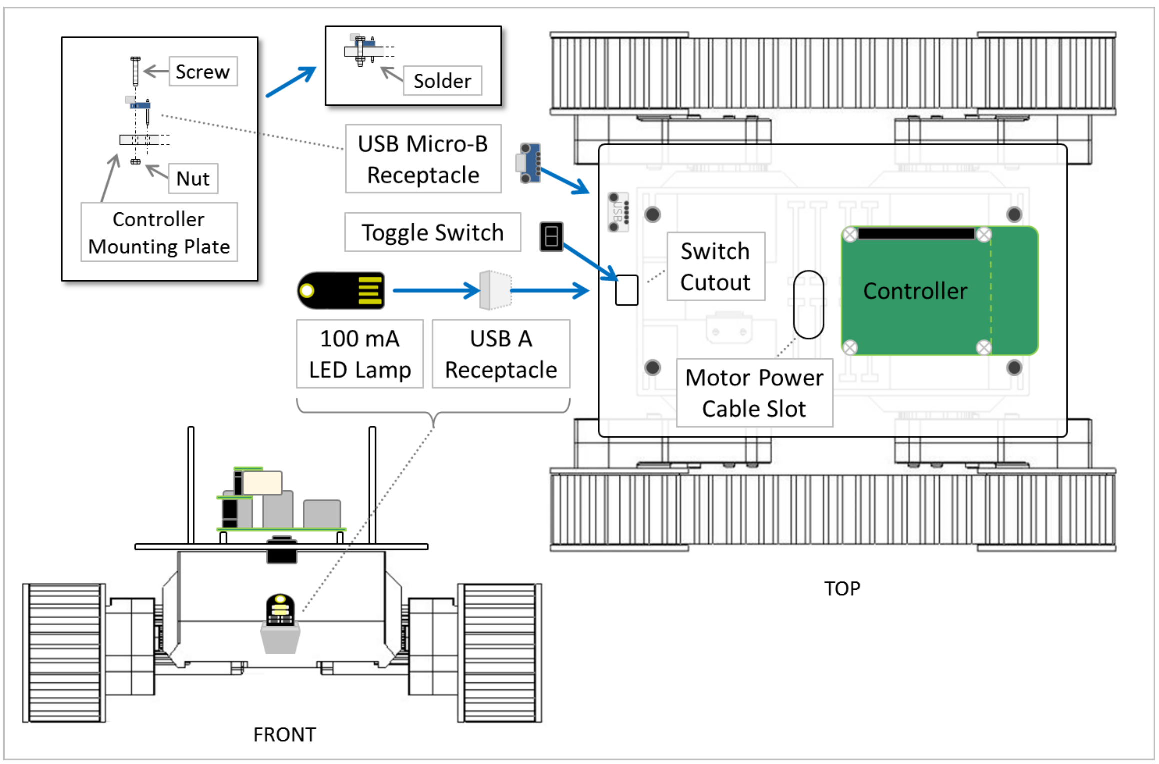

Figure 3-2: Electrical power component integration.

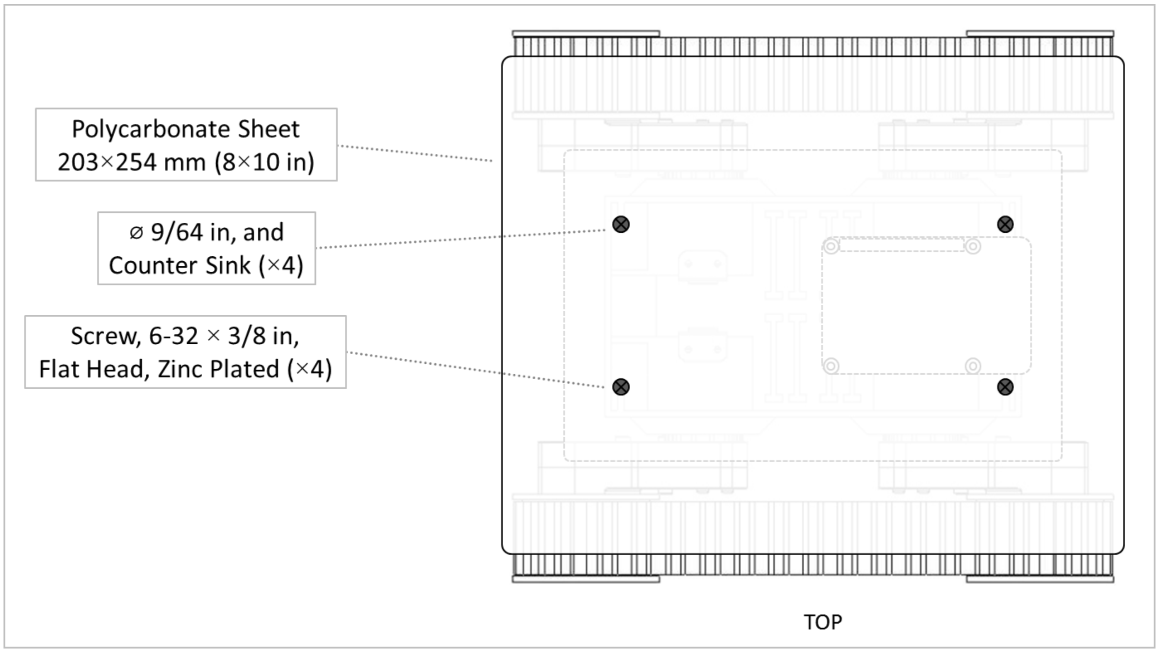

Figure 3-3: Fiducial mounting plate.

4. Software

The Raspbian operating system was installed on the controller to support the software onboard the robots. Table 4-1 contains a summary of software component.

Table 4-1: Tracked robot software

Software	Purpose
`launch-robot-#.sh`	Shell script to launch robot-client at startup and shutdown OS upon exit
`robot-client-#.py`	Python script for TCP I/O and motor commands
`Pololu_drv8835_rpi`	Python library for DRV8835 dual motor driver
`WiringPi2-Python`	Functions for managing IO expanders
`Python`	Script language interpreter
`WiringPi`	GPIO access library for the BCM2835 SoC

'#' in script names refers to the robot number (1, 2, 3, or 4)

Robot functionality was distributed between the onboard controller and the centralized control interface software. Onboard software was minimal because the control interface was responsible for motion planning and sent motor speed values to the robot via TCM messages. Figure 4-1 illustrates the onboard controller software and interfaces.

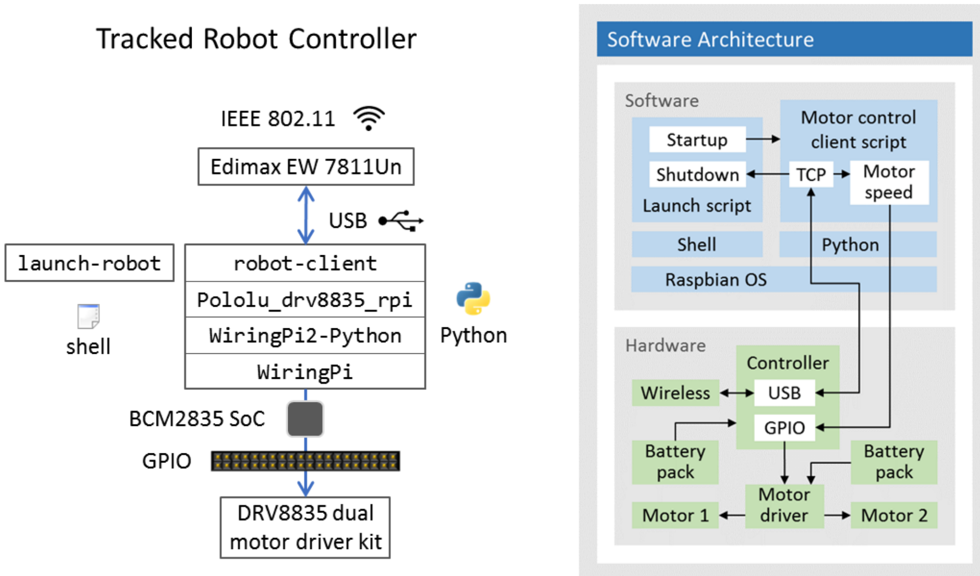

Figure 4-1: Tracked robot software components and interfaces.

`launch-robot` was a shell script which managed software startup and shutdown. `robot-client` was a Python script which processed TCP communication from the control interface and issued commands to the motor driver via the `Pololu_drv8835_rpi` library. These scripts contained a unique identification number for each robot and were named accordingly. For example, `launch-robot-1` and `robot-client-1` were installed on robot 1.

A `cron` task was scheduled on each robot to execute `launch-robot` each time the controller booted (see Figure 4-2). `launch-robot` simply launched `robot-client`, waited for it to complete, then issued a shutdown command to the operating system (OS). In addition to processing motor commands, `robot-client` listened for a shutdown command to be issued by the control interface. Upon receiving the shutdown command, `robot-client` stopped processing and returned execution back to `launch-robot`, which then issued a shutdown command to the controller OS. Thus, the onboard software ensured the main `robot-client` script always ran when the robot was powered on, and an orderly shutdown occurred before the robot was powered down.

```
#!/bin/sh
# launch-robot-1.sh
echo `date +"%Y-%m-%d %H:%M:%S"` `hostname`
#sleep 20
cd /
cd /home/pi
echo `date +"%Y-%m-%d %H:%M:%S"` robot-client-1.py
sudo python3 robot-client-1.py $1
sudo shutdown -h now
```

```
@reboot sh /home/pi/launch-robot-1.sh >/home/pi/log/cronlog 2>&1
```

Figure 4-2: Tracked robot software launch script (top) and `cron` task (bottom).