Fly-by-Pi: Open source closed-loop control for geotechnical centrifuge testing applications

User guide

Witten by

André Broekman

Department of Civil Engineering

University of Pretoria

2020

This document serves as a high-level overview on the design and operation of the Fly-by-Pi centrifuge control system. Specifically the commands required in a terminal environment to use the controller and network configuration details.

Background

The Raspberry Pi Revision 3B was chosen as the platform of choice for the project. The Raspberry Pi platform has proven to be a valuable research tool (and data acquisition system) for scientific research projects. The Raspberry Pi provides the required interfacing options, processor capability and low power consumption required for centrifuge applications. These various components interface with the Raspberry Pi's GPIO pins for control data acquisition. Remote communications are performed through a wired local area network (LAN) configuration with the Ethernet interface.

Hardware

The two primary components of the control system are the analog-to-digital converter (ADC) and the high current motor driver. The ADC provides the interface between the centrifuge's DAQ and the Raspberry Pi. The centrifuge captures and processes the sensory information generated by the experiment. A conditioned signal is provided as an analog output on the centrifuge DAQ that provides-closed loop control of the entire system. The average closed-loop control frequency was measured as 63 Hz. This configuration balances the ADC sampling frequency with the resolution attained by the ADC (14-bits, +-2.048 V). The MCP3424 breakout board from DFRobot provides four differential input channels per board. Up to four ADC's can be installed on the same I2C bus using the adjustable addressing DIP switch.

The motor driver takes the form of Pololu's high current 24v13 breakout. The driver allows for a wide range of input Voltages, PWM speed control, on-off switch and directional control. The small form factor and power efficiency does not require active cooling or significant space to install. The motor driver is interfaced directly with the Raspberry Pi's pins, as the board supports 1.8V – 5V logic. The Raspberry Pi's dedicated PWM is used to drive the motor at a frequency of 300 Hz.

The circuit diagram illustrated the various components and their respective connections with the Raspberry Pi. Power is supplied from the centrifuge's 24V power rail. The 24V rail also serves as the power supply for the motor controller that power the linear actuator. The 24V rail is stepped down with a low cost LM2596 multi output power supply to provide 5V to the Raspberry Pi. The ADC is powered from the Raspberry Pi's 3.3V power supply to match the logic voltage from the Raspberry Pi.

The motor driver powers a CAHB-10 linear actuator. The motor of the actuator is connected through a gearbox to the actuator and is rated at 24V/2A. Multiple linear actuators can be controlled with a single motor driver using the centrifuge's relay multiplexer.

Figure 1 illustrates the electronic schematic for the Raspberry Pi and the auxiliary components. Figure 2 illustrates the current implementation of the controller. An expansion shield for the Raspberry Pi provides additional power pins required for the auxiliary electronics. The power and data pins of the ADC is kept as short as possible to minimise the noise and interference as far as possible. The output power of the motor driver is transferred to the multiplexer using two banana plugs installed on the top lid of the controller. This allows for easy removal of the entire controller, along with the ethernet cable and BNC cables (not illustrated).

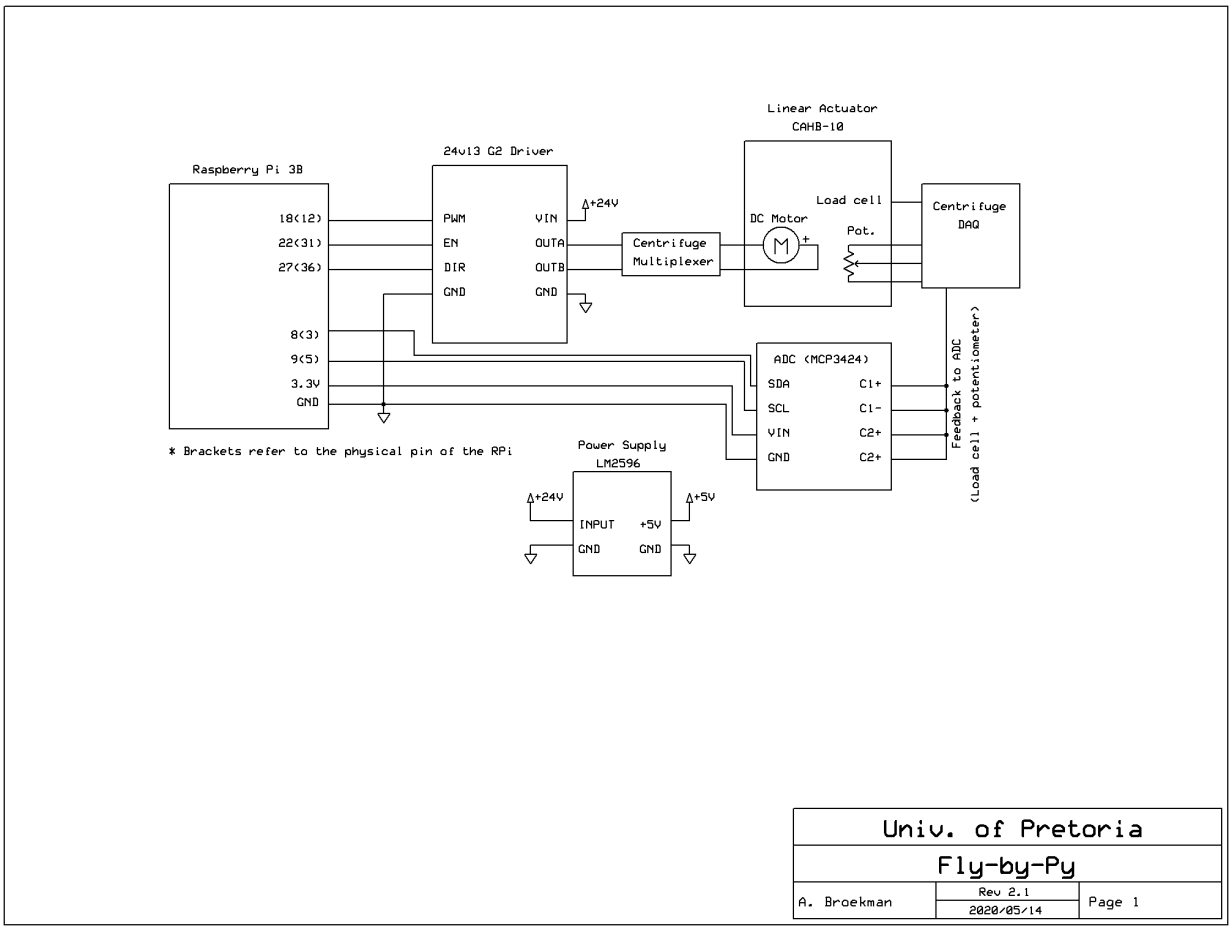

Figure 1: Fly-by-Pi electronic schematic

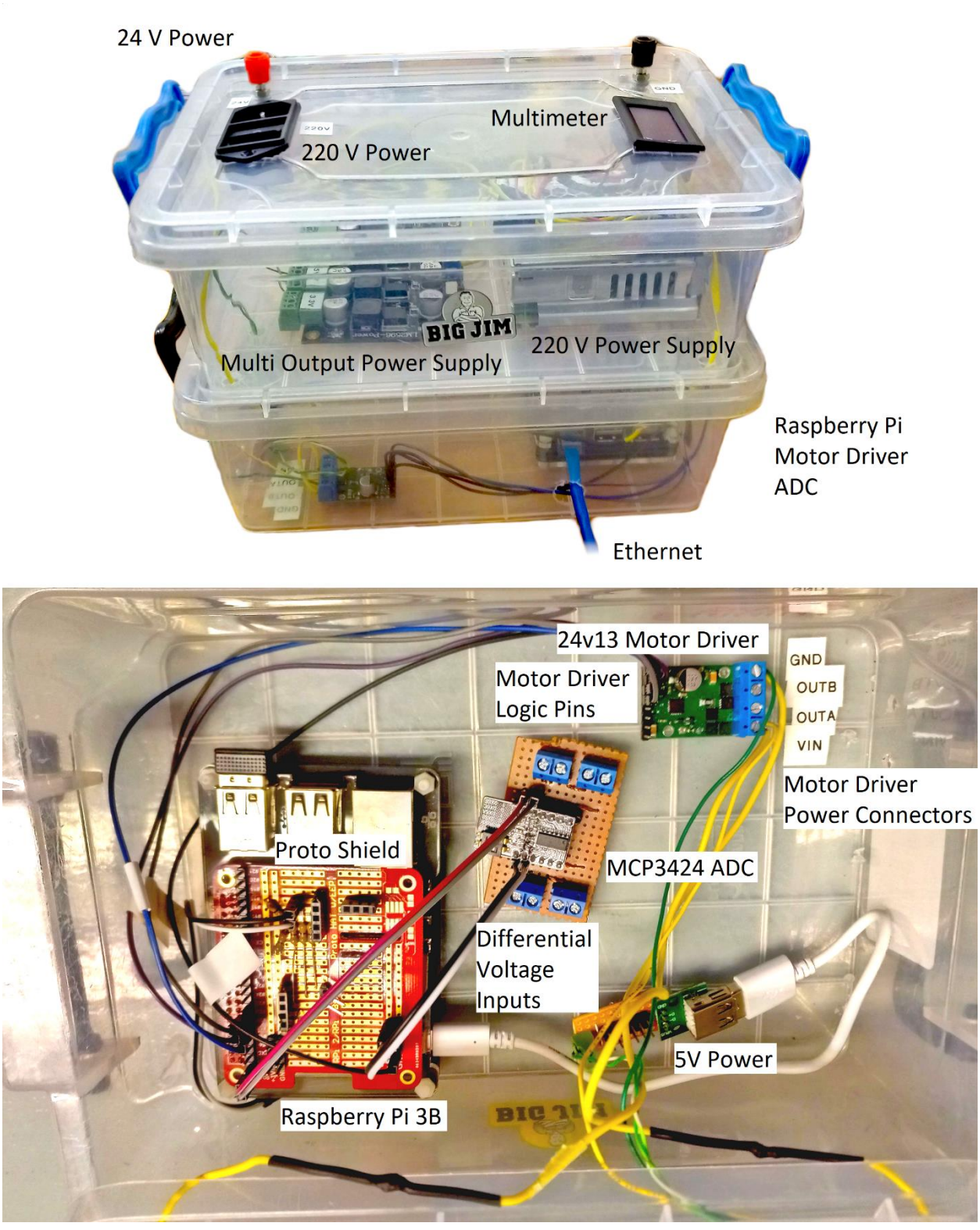

Figure 2: Fly-by-Pi controller with annotations

Software Installation

First update the Raspbian OS to the newest revision.

```
sudo apt—get update
sudo apt-get upgrade
```

Download and install VNC if it is not pre-installed:

```
sudo apt-get update
sudo apt-get install realvnc-vnc-server
```

Enable VNC for remote control:

```
sudo raspi-config
```

Upgrade to Python3 if it is not pre-installed:

```
sudo apt-get install python3-pip
```

To see the available I2C bus map:

```
sudo i2cdetect –y 1
```

This is useful to ensure that the ADC is installed and interfaced correctly the Raspberry Pi

Change the working directory to a suitable location and clone the FlyByPi software repository:

```
cd /home/pi/Desktop/
git clone https://github.com/andrebroekman/FlyByPi
```

Change the working directory to that of the *FlyByPy* folder, and test communications with ADC using the StressTestADC.py script:

```
cd ./FlyByPi
sudo python3 StressTestADCtest.py
```

Network Configuration

The Raspberry Pi is connected to the centrifuge's local area network (LAN). A default IP address is configured for the Raspberry Pi: 192.168.0.130. The default username of "pi" is used, along with a non-default password of "civil".

To assign a static IP address to the Raspberry Pi, open the `/ect/dhcpcd.conf` file with the nano editor:

```
sudo nano /etc/dhcpcd.conf
```

Scroll down the bottom of the file using the arrow keys on the keyboard and add the following information to the bottom of the file, such that it looks similar to the content below:

```
# define static profile

profile static_eth0

static ip_address=192.168.0.130/24

static routers=192.168.0.1

static domain_name_servers=192.168.0.1

# fallback to static profile on eth0

Interface eth0

fallback static_eth0
```

Exit the editor by pressing “Ctrl” and “X” together. Then press “y” to save the changes. Press “y” again to override the current configuration file. Reboot the Raspberry Pi using either a power cycle or:

```
sudo reboot
```

Putty is used to remotely control the device via SSH, using the static IP address defined above. The Raspberry Pi continued to operate and communicate normally even when subjected to high G-forces over extended periods of time, the highest magnitude reaching 50 G.

PuTTY is used to create a secure terminal connection to the remote device that is connected on the LAN (Figure 3). Enter the IP address of the Raspberry Pi (192.168.0.130) to connect. The credentials of the root user account are entered next. First the username (pi) and the password (civil).

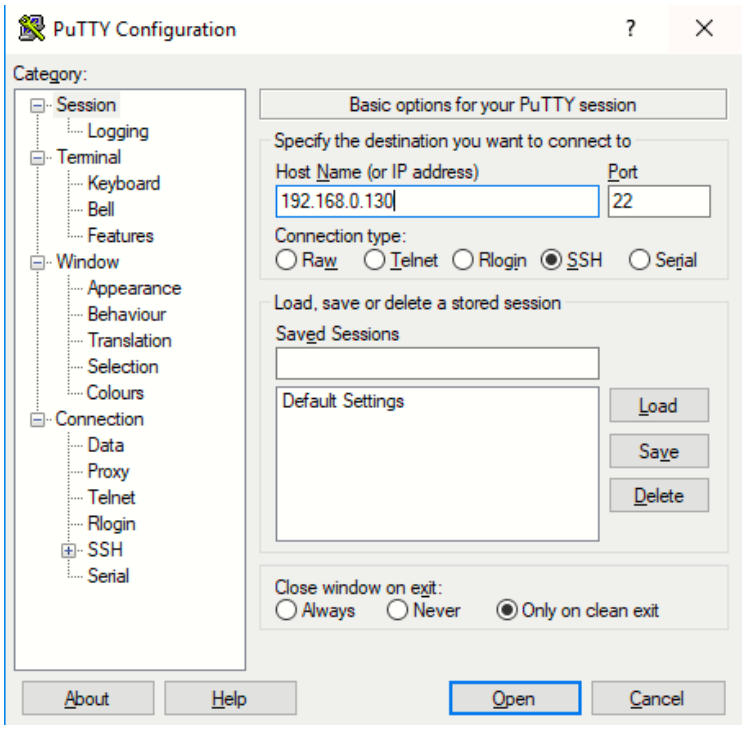

Figure 3: PuTTY configuration window

Next, change the current working directory to the correct folder using the command (Figure 4):

```
cd /home/pi/Desktop/FlyByPi
```

This is the folder where the software scripts reside and where all the data files will be stored. To run any python script, e.g. test.py, use the following command:

```
sudo python3 StressTestADCtest.py
```

This specifies the interpreter to use Python3 and to run the script in super user mode (`sudo`). To stop any Python script, press “Ctrl” and “C” together – multiple attempts may be required. When a script is terminated, the motor driver is configured to disable to motor for safety.

Pressing the “up” arrow key shows all prior user commands in the terminal. This eases navigation significantly and reduces typing errors that prevent scripts from executing. Typing “`history`” into the terminal will show all the commands used since the operating system was installed.

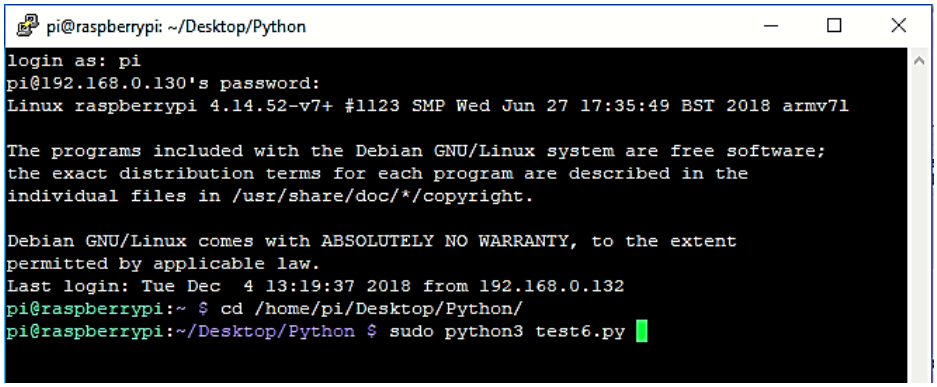

Figure 4: SSH terminal interface

To transfer files between the computer in the control room and the Raspberry Pi, WinSCP is used (Figure 5). First specify the remote network connection (identical to the details already provided). Next, navigate to the respective file structures on both the host computer and Raspberry Pi. Files can be transferred in both directions.