Appendix G

Manual Sensor Box

G.1 Sensor Box Version 1

G.1.1 Setup

Interfacing with the data acquisition tool is done using Python. A few steps are required to set up the module. Firstly, you will require a functional installation of python 3. Then, install the PySerial module using pip in a command prompt:

pip install pyserial

Start your python environment, and open the IAC_data_logging.py file [Available on Brightspace]. Run the script, and verify that you get output. Note that this is not real data, but instead data that is formatted in the same way for development purposes. The script will run perpetually, to halt it you can perform a keyboard interrupt using Ctrl+C. To acquire actual data, first power up and connect the device to your computer. The script will have provided some information about connected USB devices when you first ran it. If not, you can manually scan the USB ports by calling the port scan function in a python terminal:

port_scan()

This will provide a list of all attached USB serial devices. Identify the relevant device in the list[1] and note the USB port of the circuit. Fill in the USB port in the script options. Now, you can disable the development mode by setting:

dev = False

Try running the script with development mode disabled, and verify that you obtain output.

G.1.2 Data processing

Several preparations have to be done in order to successfully capture data during your experiment. While running, the program continuously acquires data through the USB connection. This data is structured as follows:
load_cell VALUE time_of_flight VALUE

At each step, this data is stored in the *line* variable. This data is stored as a *string*. Your first task is to process this string in such a way that the data gets stored in a useful manner. Keep in

[1] Usually, this is formatted as "COM#" with # being a number

mind that the script does not know when your experiment will end, and thus should be saving data continuously (not all at the end).

Secondly, you should design a procedure for calibrating the sensors. The program will output pure numerical values, not actual measurements. This means you will have to tackle two problems:

- The sensors (particularly the load cell) might return a non-zero value when no input is applied. For example, when not loading the sensor, you might get a reading of -5716.
- The sensors do not return values in proper units. For example, when loading the sensor with 20 kilograms, you might obtain a reading of 28371.

G.2 Sensor Box Version 2

G.2.1 Setup

Similar to version 1 of the data acquisition tool, Python is also used to interface with version 2 of the tool. The tool consists of the following items:

1x Adafruit MCP2221 (USB to GPIO/I2C/ADC/DAC microcontroller)
1x Adafruit VL53L0X (time of flight sensor)
1x Sparkfun Qwiic Scale (analog-to-digital converter for bridge sensors)

Again, a functional installation of python 3 is required, in addition to two scripts available on Brightspace: *cedargrove_nau7802.py* and *IAC_DAQ_MCP2221.py*.

Before attempting to install anything thoroughly read the entire manual.

Step 1 Install MCP2221
To install the MCP2221, follow the steps listed here: https://learn.adafruit.com/circuitpython-libraries-on-any-computer-with-mcp2221

Go through all steps, including post-install checks to make sure the module works.

On Windows, before proceeding with the post-install checks, in addition to setting the *BLINKA_MCP2221* environment variable to *1*, be sure to set the *BLINKA_MCP2221_RESET_DELAY* environment variable to *-1*.

Using the cmd prompt:
set BLINKA_MCP2221=1
and
set BLINKA_MCP2221_RESET_DELAY=-1

or

Using Windows Powershell:
env:BLINKA_MCP2221=1
and
env:BLINKA_MCP2221_RESET_DELAY=-1

Step 2 Install VL53L0X
To install the VL53L0X module follow all steps listed here: https://learn.adafruit.com/adafruit-vl53l0x-micro-lidar-distance-sensor-breakout/python-circuitpython

Step 3 Install Qwiic Scale
To use the Qwiic Scale board, copy both the *cedargrove_nau7802.py* and *IAC_DAQ_MCP2221.py* files to your working directory.

Step 4 Install adafruit-circuitpython-busdevice and adafruit-circuitpython-register
For the *cedargrove_nau7802.py* module to work, the *adafruit-circuitpython-busdevice* and *adafruit-circuitpython-register* Python modules should be installed using pip in a command prompt:

pip3 install adafruit-circuitpython-busdevice

pip3 install adafruit-circuitpython-register

Step 5 Run the data acquisition script
With all modules now installed, you should be able to run successfully run the *IAC_DAQ_MCP2221.py* script. The script will get the output values from the time of flight sensor and load cell and print them to your screen. The script will run perpetually, to halt it you can perform a keyboard interrupt using Ctrl+C.

G.2.2 Data processing

The *IAC_DAQ_MCP2221.py* script is provided to you as a start. You are free (and encouraged) to expand its functionality to meet your needs. As with version 1, values are not stored as actual measurements but as purely numerical values. It is therefore up to you to find a way to get useful measurement data from these “raw” values.