Operating manual: test data acquisition

February 24, 2022

1 Background

Firstly some background...

1.1 Load cell

A load cell is a device which.... etc etc. Long story short: voltage is linear with load for large enough loads.

1.2 Time of flight sensor

Distance sensor

1.3 I2C connections

Protocol for communicating between smalll devices (such as arduino's) using only 2 wires.

2 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. 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. To acquire actual data, first power up and connect the device to your computer. Then, run the port scan function:

```
port_scan()
```

This will provide a list of all attached USB serial devices. Probably, this will only list the data acquisition circuit. Note the USB port of the circuit, and fill it in. 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.

3 Data processing

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

```
load_cell_1 VALUE load_cell_2 VALUE load_cell_3 VALUE
load_cell_4 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 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 cells) will return a non-zero value when no load 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.