

Operating manual: test data acquisition

March 28, 2022

1 Background

Firstly some background...

1.1 Load cell

A load cell is a device which can measure applied loads. Provided the applied load is sufficiently large, the output of the load cell f_{cell} is a linear function of the applied load F , such that:

$$f_{cell}(F) = a \cdot F + b \quad (1)$$

Where a and b are values to be obtained from calibration.

1.2 Time of flight sensor

Distance sensor

1.3 I2C connections

Protocol for communicating between small 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. 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 list¹ 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.

¹Usually, this is formatted as “COM” with being a number

3 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 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.

4 Schematic

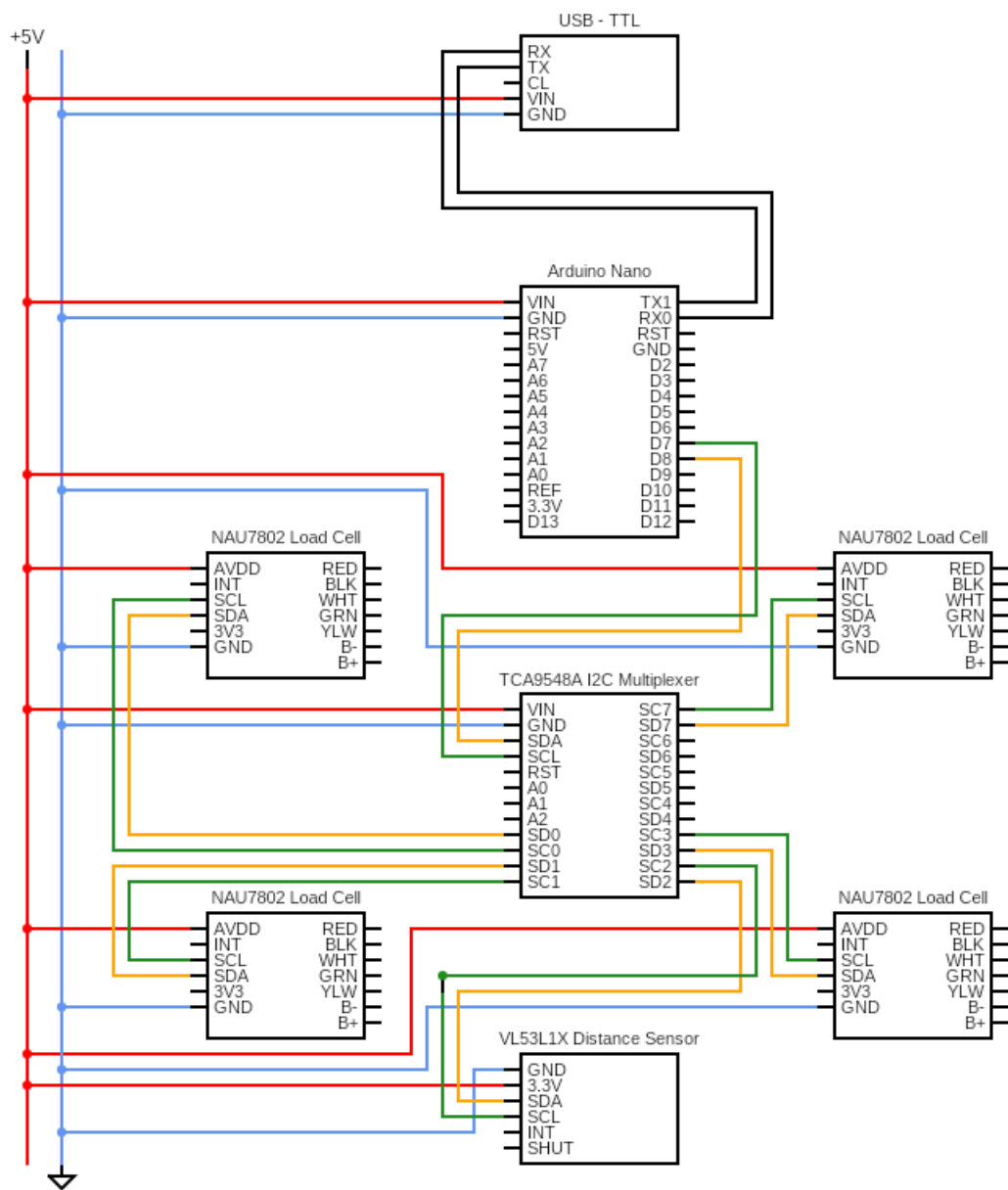


Figure 1: diagram