Live-Feed-over-LAN Pressure Sensor (LoLAN-PresS) Documentation

Members:
Kenneth V. Domingo
Rhei Joven G. Juan
Rene L. Principe Jr.

App Physics 185

1 Overview

The Live-Feed-over-LAN Pressure Sensor (LoLAN-PresS) is an implementation of velostat pressure/bend sensor which runs on an ESP8266-based microcontroller. On the hardware end, the sensor broadcasts through a local area network (LAN) using a pre-set IP address. On the software end, the feed can be retrieved, processed, and displayed in real-time through any Python interpreter on a device connected on the same network. The program depends on the following Python libraries:

- Numpy
- Matplotlib
- Scipy
- URLlib

2 Hardware setup

The hardware of LoLAN-PresS requires the following items along with their costs during Q2 2019:

Table 1: Cost of required materials.

Qty	Item	Source	Cost/pc (PhP)	Subtotal (PhP)
1	Velostat sheet	circuit.rocks	349.00	349.00
1	NodeMCU 1.0 Lua WiFi Board	circuit.rocks	325.00	325.00
			TOTAL	674.00

The velostat may come in small patches or a large sheet. Cutting this into the desired shape results in little change in its properties.

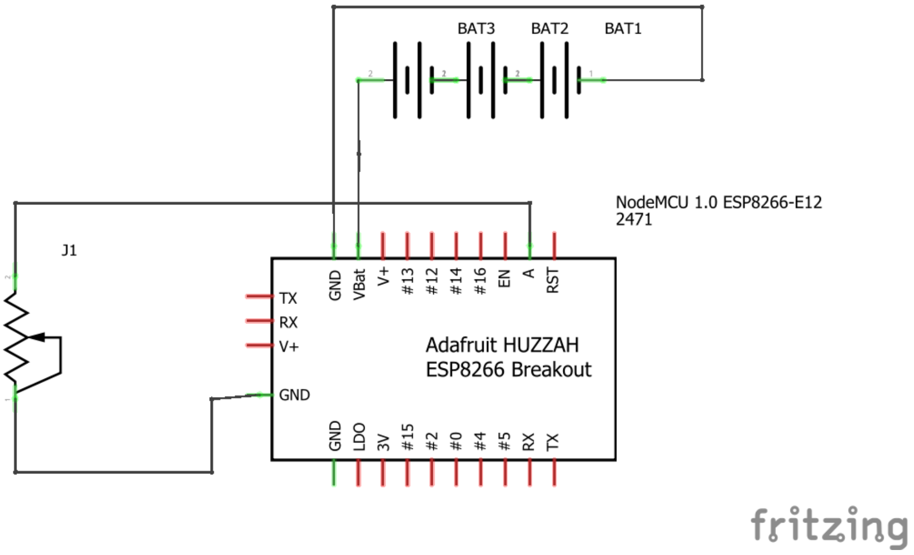

Figure 1: Schematic of the hardware setup.

3 Software setup

3.1 The PressureSensor class

The PressureSensor class contains the functions and attributes needed for communication with the microcontroller and live display of the analog data.

3.1.1 PressureSensor.__init__(url, calibration)

Instantiates the Spectrometer object and takes the calibration arguments.

Table 2: Program initialization.

Parameters	url : str Local IP address of the microcontroller. calibration : str Filename of the data to be used for calibration of voltage-to-force.

3.1.2 PressureSensor.plot_calibration()

Displays the calibration curve and the voltage-to-force equation.

3.1.3 PressureSensor.runlive()

Starts the live feed and saves all currently received data every 3 seconds to datalog.npy.

3.2 The characterization class

The characterization class allows the retrieval and processing of saved .npy files for processing later on.

3.2.1 characterization.__init__(filename, width, polyorder, lim)

Instantiates the Spectrometer object.

Table 3: Program initialization.

Parameters	filename : str Filename of the data to be processed. Accepts .txt, .log, and .npy formats. width : int Specifies the filter window length of the Savitzky-Golay filter. polyorder : int Specifies the order of the polynomial used to fit the samples in the Savitzky-Golay filter. lim : tuple Specifies the index range of the data to be processed.

The full code is available in the Appendix.

4 Demonstration

Appendix

Source code: