Flex Sensor Project

This project uses a flex sensor to describe the current tension of the sensor using the Arduino Uno board.

Initial goal:

We thought that using the flex sensor’s measurements we can calculate an approximate degree of its bending. This however failed, due to the inconsistency of the bending. The tension between its endpoints caused the middle of the sensor to bend in different ways at the same measurement time. This caused that we could not calibrate the sensor to display degrees.

Second goal:

After considering the flex sensor’s circumstances we thought a more descriptive value on the sensor’s current bend would be more appropriate. This has been achieved by taking measurement points, mapping them to values on a scale between 0 to 1, and linearly interpolating them on a range.

We marked fully bent as 1.0, strongly bent as 0.7, and slightly bent as 0.3. We then searched appropriate measurements for these values and jump between linear interpolations based on the current measurements. The value between 0 and 300 was mapped from 0.0 to 0.3, and from 300 to 800 was mapped from 0.3 to 0.7 and larger than 800 is mapped from 0.7 to 1.0. This simple solution proved to be sufficient and functional.

Later we thought that this project could be used as a peripheral to the computer, so we have implemented a mouse click functionality to the program, and thanks to the mapping we do not have double clicks, or any miss inputs.

Below is documentation for the code:  
The project uses 3 code files:

* v1.ino is the code that runs on the Arduino microcomputer.
* v1.py is the python code that runs on the local computer that processes the data the Arduino provides.
* DegreeConverter.py is a module in python that converts the data into usable measurements.

v1.ino

This Arduino sketch reads an analog input from a flex sensor connected to pin A0 and prints the raw sensor value to the Serial Monitor. It operates in a continuous loop, updating the value every 100 milliseconds.

* The analog value from the flex sensor ranges from 0 to 1023, representing the resistance change of the sensor as it bends.
* This sketch provides a basic framework for further development, such as mapping the sensor value to specific ranges or implementing additional functionality based on the sensor readings.

v1.py

This script interfaces with a serial device to read data and simulate mouse clicks based on processed input values. It utilizes the pyautogui library for mouse interactions and a custom scaling function from the DegreeConverter module.

Code Explanation

Imports

serial: Facilitates reading from the serial port.

custom\_scaling: Custom function for scaling data values.

pyautogui: Allows for mouse click simulation.

Serial Port Initialization

A close-up of a number

Description automatically generated

Initializes the serial connection on the specified COM port with a baud rate of 9600.

Sample Initialization

A computer screen shot of a computer code

Description automatically generated

Defines the sample size, initializes a list to store samples, and sets up variables for tracking changes and previous values.

Main Loop

A white background with black and white clouds

Description automatically generated with medium confidence

Enters an infinite loop to continuously read data from the serial port and processes it.

Data Handling

A white background with black text

Description automatically generated

Attempts to convert the read data to an integer, defaulting to 0 if unsuccessful.

Calculating Values

A white background with black text

Description automatically generated

Updates the current sample index, calculates the simple moving average, and applies the custom scaling function.

Output

A close-up of a computer code

Description automatically generated

Outputs the current raw and calculated values for monitoring.

DegreeConverter.py

The custom\_scaling function takes a numerical input and scales it based on predefined ranges. This function is useful for normalizing input values between specific thresholds, enabling consistent use in applications like data processing or control systems.

Parameters

x (int or float): A numerical input value that needs to be scaled. It must be in the range of 0 to 1000.

Returns

float: The scaled value corresponding to the input x. The output is constrained between 0 and 1.