Tapered beam scripts

Python 3 and Numpy 1.9.0 or later are required to run the scripts.

1. taperedbeam.py

The main script that runs the capacitance sensors and stores the data. Simultaneous video recording can be triggered by adding the option -c or –camera.

1. process\_touches.py

The main processing script that integrates the binary results from all four capacitance sensors and converts them to touches per electrode on the beam. It produces a \_touches.txt file containing all recorded touches for a single trial. More information can be found in the docstring of the script.

1. summarize.py

Integrates multiple \_touches.txt files into a single summary file, including the total number of foot faults, time to traverse the beam, as well as time and distance to the first foot fault. This script can be used after the \_touches.txt files have been corrected for false positives (either manually by comparing any warning messages to video recordings if available, or automatically by the process\_touches.py script). The output of this script, summary.txt, is in a convenient format for statistical analysis.

1. filters.py

Contains some code to filter out touches that are unlikely to be true foot faults. These filters can be applied from within process\_touches.py. For detailed information see docstrings of both scripts.

1. single\_sensor\_test.py

Prints the ‘raw’ ADC values for all 12 electrodes from a single sensor. Can be used to check whether all electrodes are well connected before applying protective finish. Normal *untouched* ADC values should range between 215-230, whereas they should decrease to about 50-90 when *touched*. Different sensors addresses can be selected by adding the -a or –-address option followed by either one of the following addresses: 0x5A, 0x5B, 0x5C, 0x5D (for sensors 1-4, respectively). Data can be saved to a .txt file through the -s or –-save option.

1. all\_sensors\_test.py

Prints the touch state of each of the four sensors. Can be used to check whether all electrodes are working correctly before starting measurements.

1. MPR121\_edited.py

Modified version of the standard MPR121.py script that comes with the installation. The edited script is a requirement for all above scripts as it increases the sensitivity of the capacitance sensors.

Installation steps

It may be most convenient to install the software and test the connection with a single capacitive touch sensor first, before proceeding to build the tapered beam apparatus.

1. Connect the MPR121 to the Raspberry Pi GPIO pins. A simple guide to the GPIO header can be found [here](http://www.raspberrypi-spy.co.uk/2012/06/simple-guide-to-the-rpi-gpio-header-and-pins/). The IRQ should be connected to GPIO26 when using all four sensors, but this can be left out when just testing one sensor.
2. Make sure the I2C interface is enabled by running sudo raspi-config in the command window. Select *Interfacing options* > *I2C* > *Enable*.
3. Install I2C tools by running sudo apt-get install -y i2c-tools.
4. Check the address of the connected sensor by running i2cdetect -y 1. The address shown should correspond to the address specified through the ADDR pin on the sensor chip (see the online [MPR121 tutorial](https://cdn-learn.adafruit.com/downloads/pdf/adafruit-mpr121-12-key-capacitive-touch-sensor-breakout-tutorial.pdf) by Adafruit for more information).
5. Install the MPR121 software for Python: git clone <https://github.com/adafruit/Adafruit_Python_MPR121.git>
6. In the newly installed Adafruit folder, add the MPR121\_edited.py script. This edited script enables autoconfiguration of the sensors with higher sensitivity necessary to detect mouse touches.
7. Launch a command window in the Adafruit folder and run sudo python3 setup.py install.
8. Done! Try running simpletest.py in the Adafruit/examples folder to check that everything works correctly.