**Sensorized glove**

**Objective:**

To sensorize a glove for force and acceleration sensing of the hand during grasping and object manipulation.

**Components:**

1. Glove: two different layers. One for sensor installation and one for sensor protection.
2. Force sensors: force sensitive resistors (FSR) from Interlink Electronics (Westlake Village, CA, USA). Sensors used are [FSR 400](http://www.interlinkelectronics.com/FSR400.php), [FSR 400 Short](http://www.interlinkelectronics.com/FSR400short.php), and [FSR 402](http://www.interlinkelectronics.com/FSR402.php).
3. Accelerometer: MPU-6050.
4. Electrical wire: [NEF26-10546](http://www.coonerwire.com/hookup-wire-nef/), from Cooner Wire. This one is attached directly to the FSRs.
5. Ribbon cables to connect the NEF cable to the breadboard.
6. 16 pins flat connectors mounted on the breadboard, to interface with the ribbon cables.
7. Breadboard with the necessary components to interface the sensors to an Arduino.
8. [Arduino Mega](https://www.arduino.cc/en/Main/arduinoBoardMega) to handle signal acquisition and computer communication.
9. LoadCell (FRS166, Pressure Profile Systems, Los Angeles, CA, USA) used for calibration of the FSR.

**Sensors location:**

The accelerometer is placed at the center of the back of the hand, and 19 FSRs are used for sensorizing the fingers and the palm as shown in Figure 1.

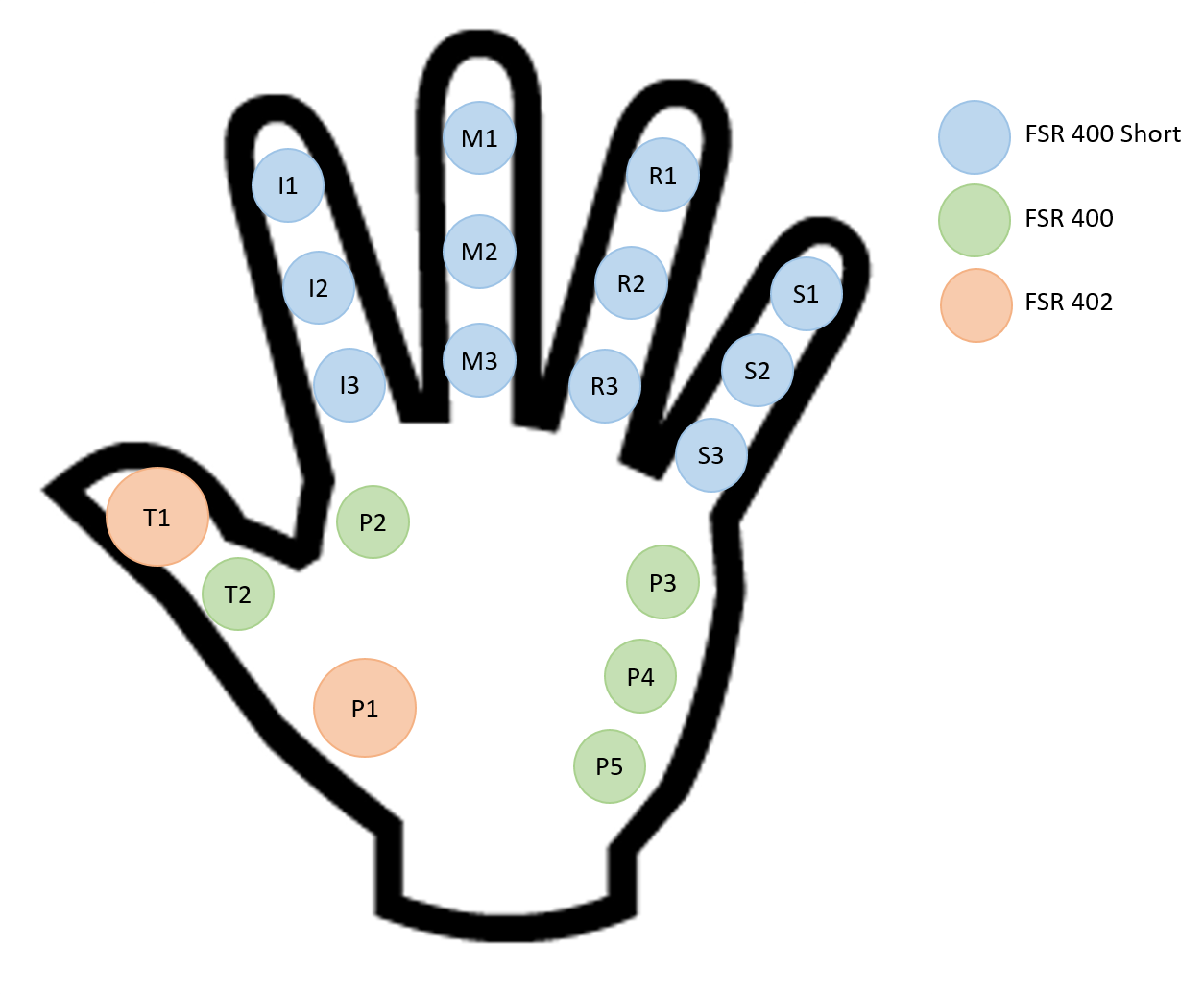


Figure 1. FSRs locations.

**Electronics design:**

The signals of the accelerometer and the FSRs are acquired with an Arduino Mega. The accelerometer is connected to an Arduino as shown in Figure 2, and the FSR is connected for analog input to an Arduino as shown in Figure 3. The sensorized glove developed has 19 FSRs; however, the Arduino Mega only has 16 analog inputs. In this case, for the current design, sensors I3, M2 and R2 are not connected to the Arduino. In a future version, a custom circuit board must be designed. This board must use a microcontroller that can manage all the input signals of both the accelerometer and the FSRs.

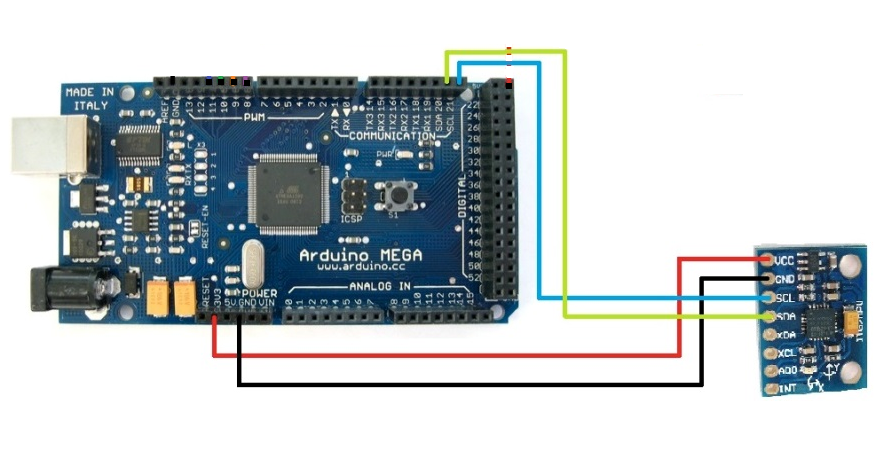


Figure 2. Connection between MPU-6050 and Arduino Mega.

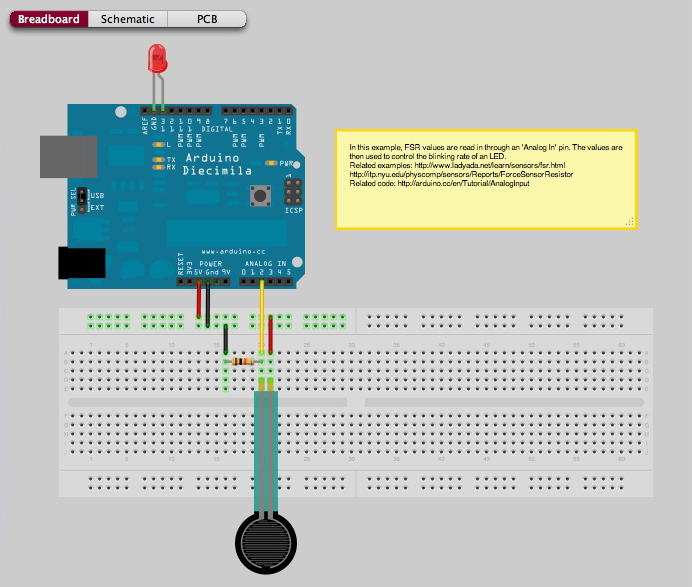


Figure 3. Connection between FSR and Arduino for analog input, using a breadboard.

Three ribbon cables, each one with 15 wires, are used to connect the sensors to a breadboard, using DB15 adapters for quick-release. These adapters are connected to three 16 pins flat connectors mounted on a breadboard. Figure 4 shows the relation between the sensors and the ribbon cables. Table 1-3 show the relation between the sensors, the DB15 pins, the 16 pins flat connectors, and the Arduino inputs.

Figure 4. Relation between the glove sensors and the ribbon cables/DB15 adapters.

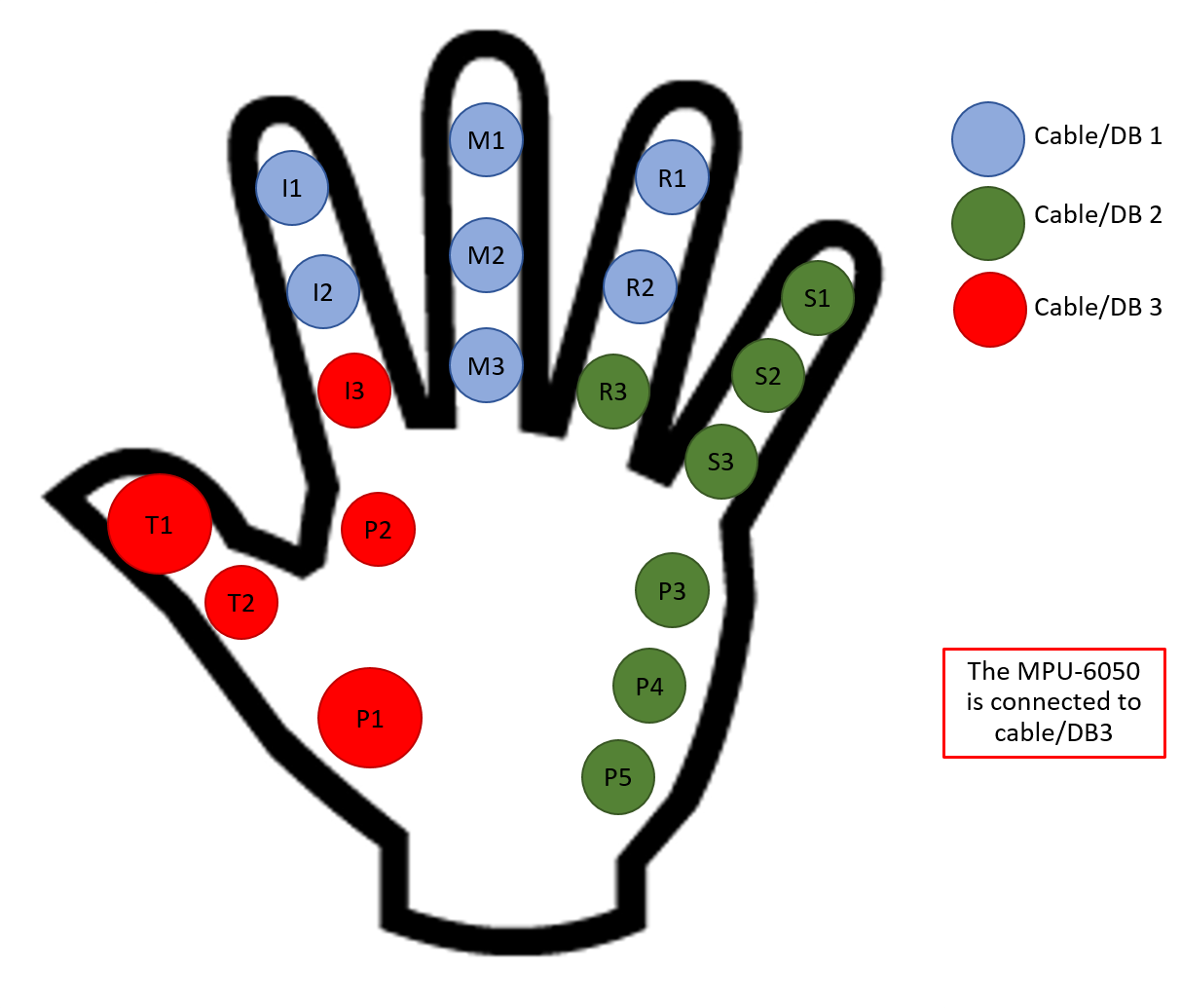


Table 1. Relation between ribbon cable/DB15.1 pins, sensors, and Arduino analog inputs.

|  |  |  |  |
| --- | --- | --- | --- |
| Cable/DB15.1 pin | 16 pin connector 1 | Sensor | Arduino input |
| 1 | 1 | R1 | A10 |
| 2 | 2 |
| 3 | 3 | M1 | A9 |
| 4 | 4 |
| 5 | 5 | M3 | A8 |
| 6 | 6 |
| 7 | 7 | I2 | A7 |
| 8 | 8 |
| 9 | 15 | Not connected/not use | |
| 10 | 14 | R2 | Not connected |
| 11 | 13 |
| 12 | 12 | M2 | Not connected |
| 13 | 11 |
| 14 | 10 | I1 | A13 |
| 15 | 9 |

Table 2. Relation between ribbon cable/DB15.2 pins, sensors, and Arduino analog inputs.

|  |  |  |  |
| --- | --- | --- | --- |
| Cable/DB15.1 pin | 16 pin connector 2 | Sensor | Arduino input |
| 1 | 8 | P5 | A12 |
| 2 | 7 |
| 3 | 6 | S2 | A11 |
| 4 | 5 |
| 5 | 4 | P4 | A15 |
| 6 | 3 |
| 7 | 2 | P3 | A14 |
| 8 | 1 |
| 9 | 9 | S1 | A6 |
| 10 | 10 |
| 11 | 11 | S3 | A5 |
| 12 | 12 |
| 13 | 13 | R3 | A4 |
| 14 | 14 |
| 15 | 15 | Not connected/not use | |

Table 1. Relation between ribbon cable/DB15.3 pins, sensors, and Arduino analog inputs.

|  |  |  |  |
| --- | --- | --- | --- |
| Cable/DB15.1 pin | 16 pin connector 3 | Sensor | Arduino input |
| 1 | 1 | I3 | Not connected |
| 2 | 2 |
| 3 | 3 | T1 | A2 |
| 4 | 4 |
| 5 | 5 | T2 | A1 |
| 6 | 6 |
| 7 | 7 | P1 | A0 |
| 8 | 8 |
| 9 | 15 | Not connected/not use | |
| 10 | 14 | P2 | A3 |
| 11 | 13 |
| 12 | 12 | MPU-6050: SDA | SDA |
| 13 | 11 | MPU-6050: SCL | SCL |
| 14 | 10 | MPU-6050: GND | GND |
| 15 | 9 | MPU-6050: VCC | VCC |

**Software design:**

Figure 5 shows the GUI developed to visualize the functioning of the sensorized glove. The GUI has a 2D view of the back of the right hand, with different areas representing the 19 FSRs installed on the glove. Each of these areas changes colour based on the force measured. The colour scale is displayed on the right side, going from O N to the maximum force used during calibration (default value is 10 N). The GUI allows the user to save the force and acceleration data in a text file. The software creates a new file each time the *Record* button is pressed. The default name of the file is “New Trial Timestamp.txt”; however, the user can choose a *Trial ID*, changing the name of the file to “Trial ID Timestamp.txt”. The headers of the save data are:

* The first 16 columns are the force values, going from thumb to small finger to palm sensors: T1, T2, I1, I2, M1, M3, R1, R3, S1, S2, S3, P1, P2, P3, P4, P5.
* The last 7 columns are the accelerometer sensor data: Accel\_X, Accel\_Y, Accel\_Z, Temperature, GYRO\_X, GYRO\_Y, and GYRO\_Z.

The GUI also displays the status of the Arduino and the Accelerometer and has a *log* to display important information about the functioning of the system to the user. In case, there is a disconnection of the Arduino, the GUI will alert the user, and will initialize a re-connection method. This method may fail due to hardware limitations intrinsic to the Arduino. In this case, the user has to check that the physical connection between the Arduino and the computer is working correctly and restart the application if necessary.

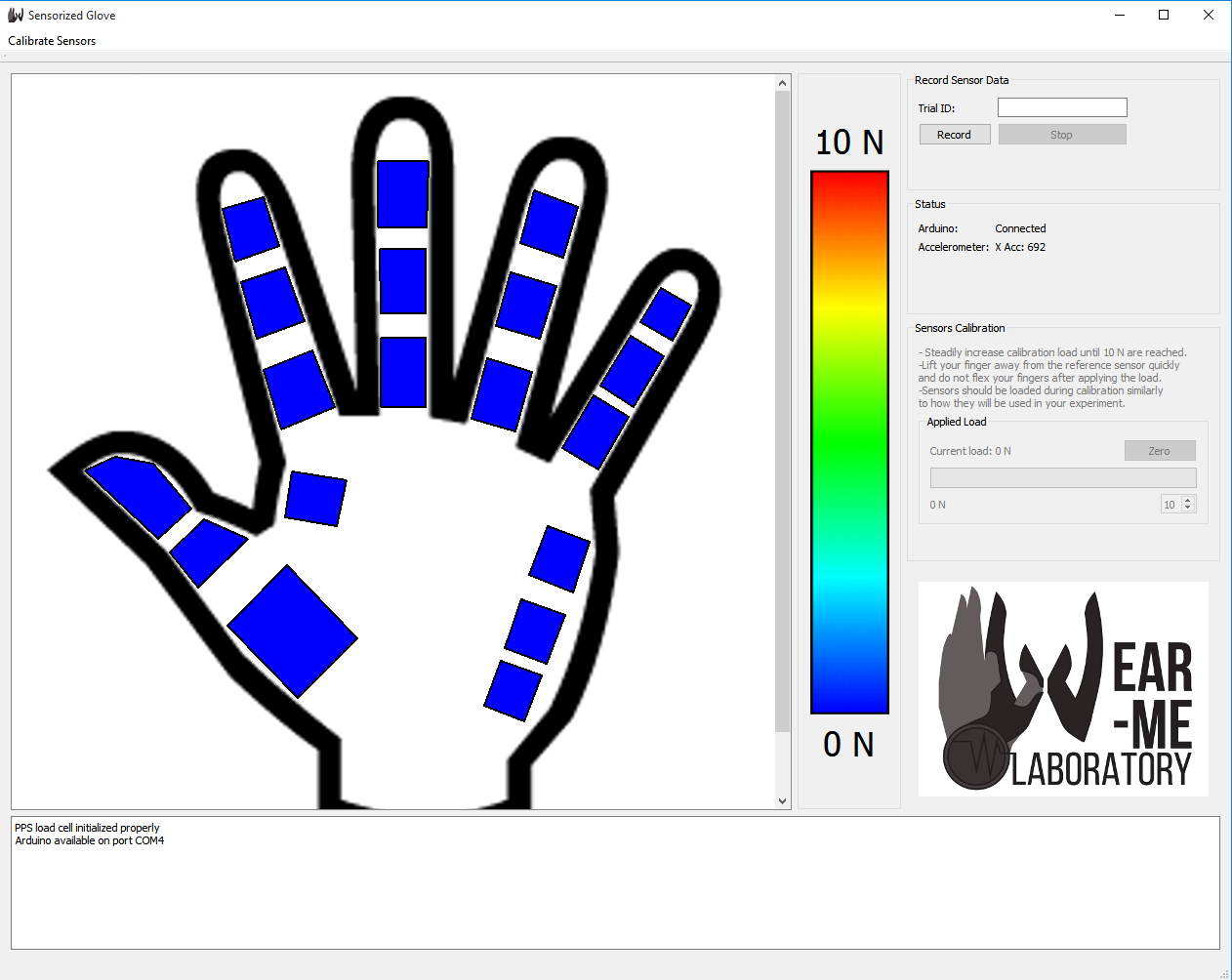


Figure 5. GUI of the sensorized glove.

**FSR calibration and assessment:**

The software allows the user to calibrate each FSR and perform an assessment of this calibration. The calibration values are stored in the file “./Setup/FSR\_calibration\_values.txt” and are loaded once the software is started. In the menu bar, when the user presses the option *Calibrate Sensors/Calibrate*, the panel *Sensor Calibration*, becomes enable. This panel provides the user with some information about how to calibrate the sensors and displays the current force applied on the load cell and the current conductance of the FSR under calibration. In addition, the sensors located on the hand image of the GUI will flash, letting the user know which sensor needs to be calibrated.

*In order to calibrate a sensor:*

1. Check the GUI to see which sensor needs to be calibrated.
2. Press against the load cell with the desired sensor, ensuring full contact is made between the sensor and the load cell.
3. Press until the maximum calibration force is reached.
4. The software will calculate a linear regression to determine the calibration parameters. If the intersection of the linear equation is less than ±0.5 N and if the *r2* is more than ±0.92, the user will be notified that the sensor was calibrated correctly and a new sensor will start flashing in the GUI. Note: if the user desires to skip the calibration of a particular sensor, or accept the calibration values (even though they do not satisfy the constrains above), this can be done by pressing the respective buttons.
5. Repeat this process for all 16 sensors.
6. When calibration is completed, the user is notified, the panel calibration becomes disable and the calibration values are stored in the calibration file.