THE IMPLEMENTATION OF A RASPBERRY PI LOW COST SENSOR GLOVE

Valeriu Manuel IONESCU
University of Pitesti, Romania
valeriu.ionescu@upit.ro

Keywords: wired glove, Raspberry Pi, client-server application, sensor

Abstract: *Wired or sensor gloves are systems can be wore like a glove and are used to capture the physical movement of the fingers and the hand. They can be bought or can be manually crafted and their cost can range from cheap to very expensive. This paper presents the implementation of a low cost wired glove using handmade sensors, connected to a Raspberry Pi board and shows considerations regarding the quality and feasibility for a system that is controlled by finger and hand motion.*

1. INTRODUCTION

Capturing the movement of hands and fingers is used in virtual reality, biomechanics, animation, for controlling devices and transmitting information from the user to a computer system.

This is done by placing sensors that react by changing measurable properties to the variations of the physical or chemical properties of the environment. In the majority of cases passive sensors are used that measure acceleration, pressure, temperature, etc.

The first glove was created in 1976 and used flexible tubes with a light source at one end and a photocell at the other to register joint movement [1] but this was not an independent system, being connected via wires to a stationary machine as seen in Fig. 1. Since then, the microcontroller market has evolved allowing the costs to go down and the sensor quality to increase, while making the whole system portable.

The price of creating a system that capture these movements ranges from very cheap to extremely expensive, the main difference being the sensors type, number and the microcontroller gathering the data.

(a) (b)

Fig. 1. The evolution form the initial wired glove (a) to the novel CyberGlove II (b) has been in both hardware and portability

The expensive gloves such as CyberGlove II have a large number of sensors, detecting movement form each finger, palm arch and wrist movement. The sensor size is small and is nearly undetectable [2].

An novel approach for expensive gloves is made by the Sense Glove [3] where the glove is actually limiting the hand's movement according to the virtual reality objects that the hand is touched, giving this way the ability to "sense" virtual objects.

On the other end of the cost spectrum, there are many projects that aim to build a low cost sensor glove that use general purpose platforms such as Raspberry Pi [4], or Arduino, [5, 6]. The sensor number in this case is reduced, their visibility is noticeable and the type range from custom made to low cost sensors (such as capacitors, variable resistors, etc.)

This paper presents the implementation of a custom made, low cost, glove sensors connected to a Raspberry Pi board that communicates the data via wireless to a remote server. An altitude sensor - the Bosch BMP180 - was also used to determine significant hand up/down changes, due to the sensor resolution.

The sensor reading was made in Python language and the data is stored in CSV format that can be easily imported in other applications, including databases.

The paper presents the system structure, the sensor manufacturing, the application structure and finally concludes with observations regarding the implementation and future developments.

2. SYSTEM DESIGN

The system uses a Raspberry Pi model 3B and the operating system is Raspbian.

The glove motion capturing system operates in the following way:

-the glove catches an object. The moment is detected by the pressure sensor but also the resistive sensors that detect the bending of the fingers will detect the motion.

-the motion itself will be detected by the BM180 sensor that will give the information if the motion is going from down to up or the other way around.

-the object is placed on the resistive pressure sensitive sensor connected to a charge-discharge circuit. After the object is released, the fingers will move again and the information will also be available from this source.

-all the collected data is stored in a CSV file that is sent to a remote server via Wireless connection.

3. THE SENSORS

Resistive sensors are parametric sensors, meaning that they operate by changing the electrical resistance of the element when the properties of element that needs detection are changing.

The electrical resistance of a material is given by the equation (1).

$$R = \frac{\rho * l}{s} \tag{1}$$

Where: ρ – material resistivity; l - length [m]; s – section area [m^2]. Therefore any modifications for these factors influence the electrical resistance that can be recorded and used.

The pressure sensor was created using antistatic foam placed between to conductive wires as seen in Fig.1.

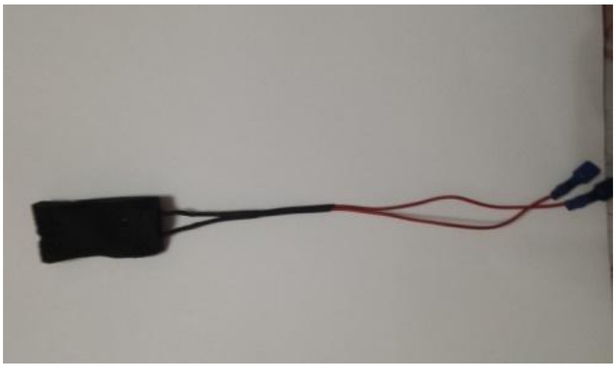
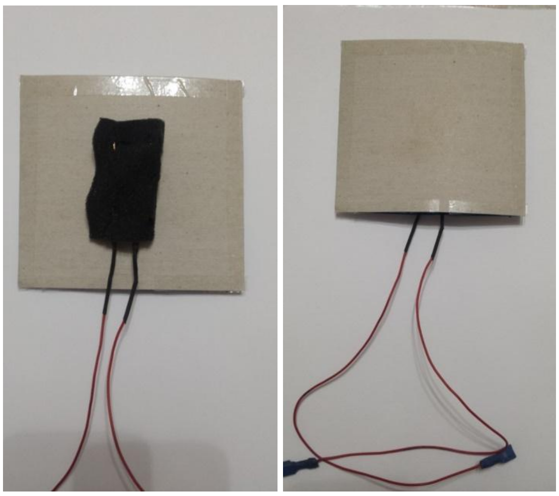


Fig. 1. The pressure sensor

A similar sensor was created for detecting the placement or removal of the weight that the glove was holding. The only difference is the size and placement of the foam as seen in Fig. 2.

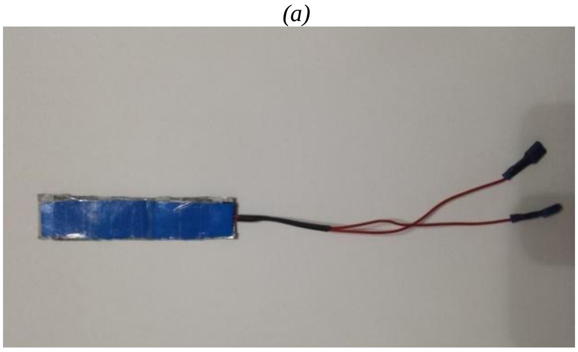
The bending / flexing sensor was implemented using a flexible potentiometer created using paper, aluminum foil and graphite as seen in Fig. 3.



(a) (b)

Fig. 2. Pressure sensor created with resistive pads (a) components, (b) assembled sensor

(a)



(b)

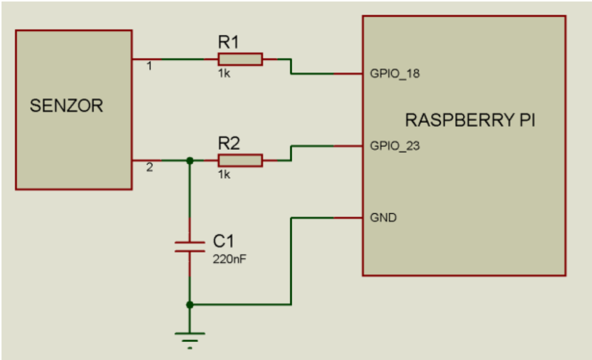
Fig. 3. Flexing sensor (a) components, (b) assembled sensor

Raspberry is not able to read analog data given by the pressure sensors because it does not have an ADC converter. That is why an charge/discahrge circuit was built using the Step Response technique. This is a RC circuit that has a timed response to an electrical pulse. A software counter gives information about the analog data being read. A higher signal value gives a longer time and a smaller signal gives a shorter time.

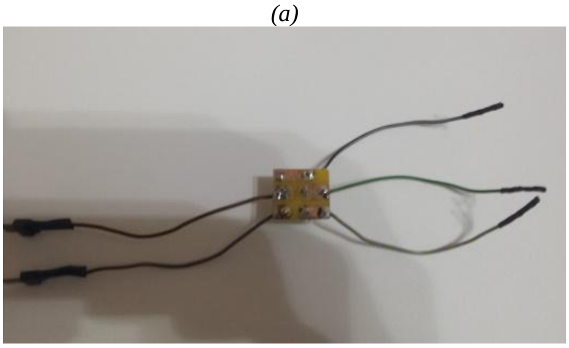
The circuit is presented in Fig. 4.

The final sensor is the altitude sensor was implemented using the Bosch BMP180 pressure sensor that is connected to the Raspberry Pi via I^2C interface. The resolution of this sensor is 0.25m, therefore significant hand movement was necessary to detect height change (Fig. 5).

In order to avoid errors, the value was averaged for a series of 3 measurements, even if that implied certain latency in the detection process.



(a)



(b)

Fig. 4: A RC circuit used in the Step Response technique: (a)circuit, (b) implementation

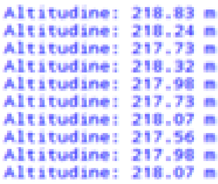


Fig. 5: The Bosch BMP180 pressure sensor response needs a rather large hand movement (0.25m) to signal the altitude change because of the resolution

The system with the connection of sensors is presented in Fig. 6.

The bending and pressure sensors are placed on the glove, while the pressure sensor is placed where the user will place the object.

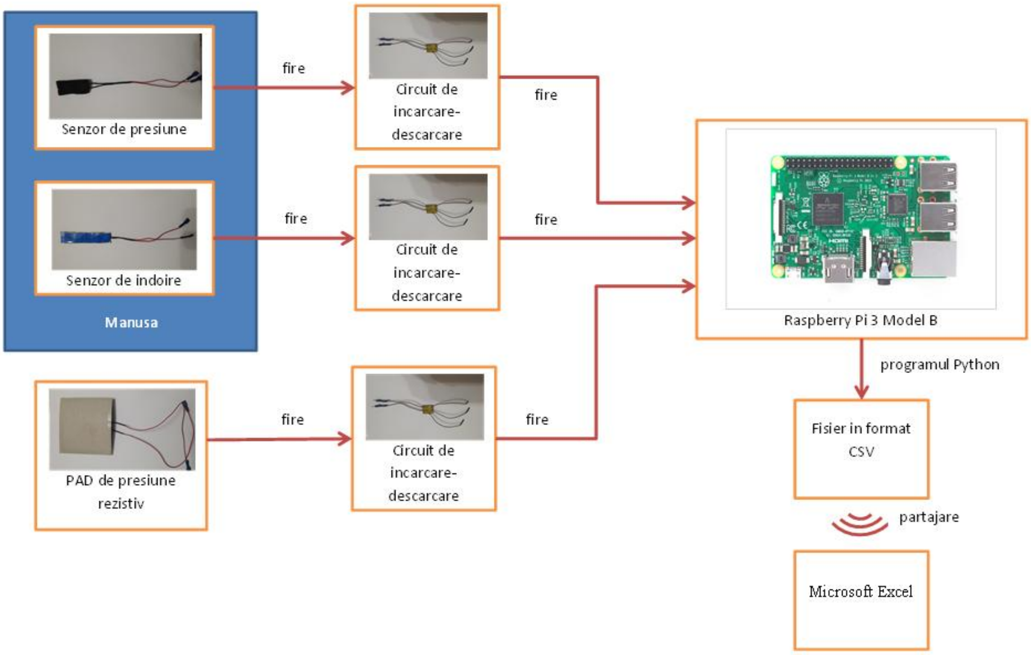


Fig. 6. System structure with connected sensors

4. THE APPLICATION

Python is a rather new programming language that appeared at the beginning of the '90s which accentuates code legibility and having a simpler syntax. For the platform used this is one of the methods that allow reading the data at a very high speed [7].

The program starts by importing the libraries that allow reading the GPIO data from the python program. [8]

After this, the sensor reading is made, which consists of two stages: discharging the step response circuit and counting the charge time that will give information regarding the analog value that is being present.

Finally the information is displayed on the screen and saved/appended to a CSV file that can be accessed by the server directly because the folder where the data is written is shared with the server computer via the wireless network. The program structure is presented in Fig. 7.

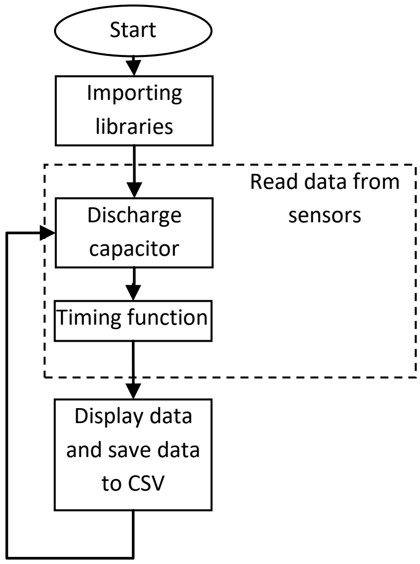


Fig. 7 Python Application structure

5. SYSTEM TESTING

The system was implemented and tested in order to have the following funcitons:

-detect the closure of the hand;

-detect the placement/removal of the weight on the pad;

-detect individual finger bending.

-detect hand lifting and lowering

-save all data in CSV format and communicate with the server.

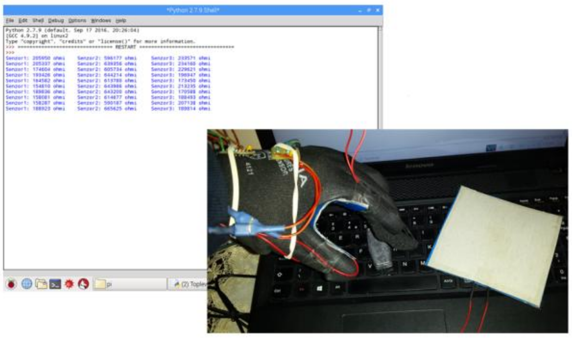
The data output at the moment when the glove was tested, with the three stages of the system operation, is presented in Fig. 8.

The sensor response to placing the object on the pad is presented in Fig.9 (a) and (b). The moment when the weight is placed on the pad is clearly detected. The response to two of the bending sensors placed along the finger length is presented in Fig. 9 (c).

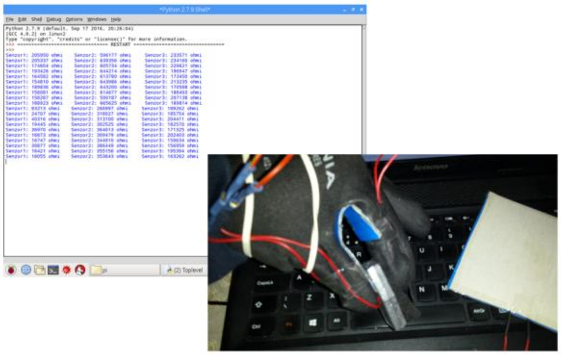
In the moment when the weight is picked up is clearly seen.

The different responses from the sensors are due to the manual implementation of the resistive sensors and the different bending of the fingers that pick the object.

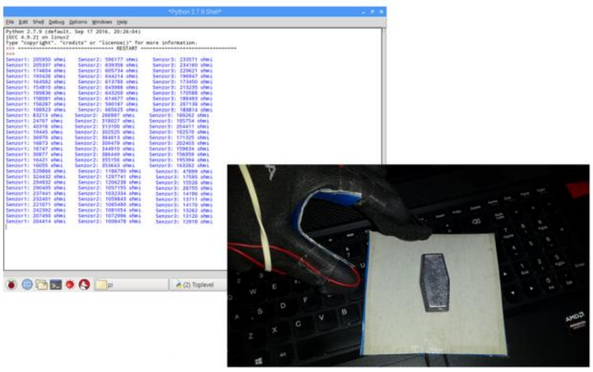
After the calibration (establishing the level when a finger is flexed or not) it was easy to write a routine to show what the action of each finger.



(a)

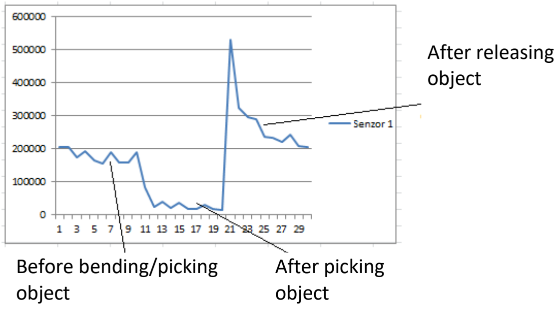


(b)

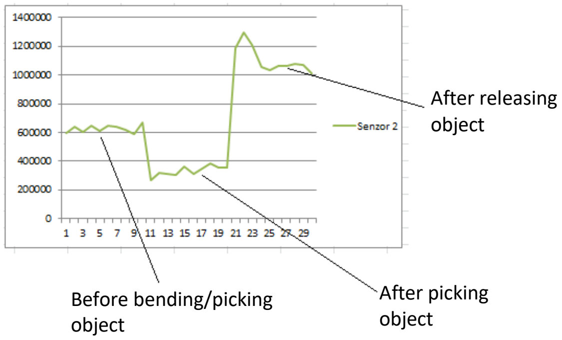


(c)

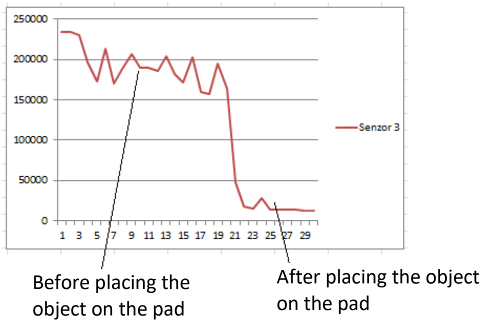
Fig. 8 The three stages of the system operation: (a) glove with no object and no weight present on the pad sensor (b)glove with object and bent finger sensors (c)glove with no object and weight present on the pad sensor



(a)



(b)



(c)

Fig. 9 The sensor response to (a) bending/flexing one finger (b) bending/flexing a different finger (c) placing the object o the pad

The final implementation of the sensor glove is presented in Fig. 10.

Fig. 10 The glove with attached sensors and the step response circuit. The bending sensors are placed on the fingers inside area

6. CONCLUSIONS

This paper presented a sensor glove implementation based on Raspberry Pi.

The sensors are low cost and custom made. The usability of these sensors proved to be at an acceptable level and the detection of individual finders of the glove was possible. The pressure sensor however proved to be difficult to use due to the artisanal nature of the implementation, needing significant pressure to be applied to record changes, and after a short period of testing it had to be replaced due the mechanical deformation of the sponge used inside.

An improvement for the system in weight, power consumption and sensor reading frequency would be to use Raspberry Pi zero that has an even higher GPIO reading rate, due to the size and decreased complexity. Another improvement would be to switch from Linux to Ultibo (that is not an operating system but an open source baremetal environment based on Lazarus) that is specially oriented for capturing data, or even an RT OS such as FreeRTOS or ChibiOS/RT. This behavior is due to the fact that the computer latency has increased for the modern operating systems (even if the hardware itself is much faster) because there are many processes that need to be queued and they interfere with the data collection [9]. During the debugging and testing of the system it was observed that not showing the data output in the console can speed the data collection process as this inserts a large lag into the data query loop.

7. ACKNOWLEDGMENT

The author acknowledges the contribution of Sarpe Florin Cristian in testing the ideas presented in this paper and making a working system prototype**.**

8. REFERENCES

[1]. Sturman, D.J., Zeltzer, D. "A survey of glove-based input". IEEE Computer Graphics and Applications. 14 (1): 30–39., 1994, doi:10.1109/38.250916

[2] M. V. Liarokapis, P. K. Artemiadis and K. J. Kyriakopoulos, "Telemanipulation with the DLR/HIT II robot hand using a dataglove and a low cost force feedback device," 21st Mediterranean Conference on Control and Automation, Chania, 2013, pp. 431-436., doi: 10.1109/MED.2013.6608758

[3] Gijs den Butter, “Sense Glove Press Release”, web, Accessed: 01.12.2017, http://www.senseglove.com/#_features

[4]. Dan Aldred, “Project New York: the Social Media Pi Glove”, web, Accessed: 01.12.2017, http://www.tecoed.co.uk/project-new-york.html

[5] Hunter Scott, “How to build a pair of sensor-gloves for 40”, web, Accessed: 01.12.2017, https://www.hscott.net/how-to-build-a-pair-of-15000-sensor-gloves-for-40/

[6] Jonathan Cates, Nobufumi Takahashi, Ryan Barton, “Arduino Flex Sensor Glove”, web, Accessed: 01.12.2017, http://www.instructables.com/id/Arduino-Flex-Sensor-Glove/

[7] Joonas Pihlajamaa, “Raspberry Pi 2 vs. 1 GPIO Benchmark”, web, Accessed: 01.12.2017, http://codeandlife.com/2015/03/25/raspberry-pi-2-vs-1-gpio-benchmark/

[8] Python Software Foundation, “RPi.GPIO”, web, Accessed: 01.12.2017 https://pypi.python.org/pypi/RPi.GPIO

[9] Dan Weinreb, “Computer latency: 1977-2017”, web, Accessed: 01.12.2017, https://danluu.com/input-lag/