Faculty of Science, Technology & Creative Arts

Using FSR Sensors to Provide Tactile Skin to the Humanoid Robot KASPAR

Guillermo Barbadillo
Prof. Dr. Kerstin Dautenhahn
Luke Wood

Technical Report 511

School of Computer Science, University of Hertfordshire

August 2011

Using FSR Sensors to Provide Tactile Skin to the Humanoid Robot KASPAR

Guillermo Barbadillo   Prof. Dr. Kerstin Dautenhahn   Luke Wood

Adaptive Systems Research Group, University of Hertfordshire

Abstract

The work presented in this paper was completed during a summer internship at the University of Hertfordshire. The project aimed to explore the possibility of using FSR Sensors with the humanoid robot KASPAR in order to gain tactile feedback. This paper covers all the processes from applying the sensors to monitoring their filtered values on the computer. In addition, details of the electronics used including the Arduino board used to receive the sensor data and send it to the computer are discussed.

1 State of The Art

1.1 KASPAR Robot

KASPAR is a child-sized robot which functions as a platform for Human-Robot-Interaction studies, using mainly physical bodily expressions (movements of the head and arms) and gestures to interact with a human. KASPAR is a 60 cm high robot and is fixed in a sitting position. The main body of the robot contains the electronic boards, batteries and motors. KASPAR has 8 degrees of freedom in the head and neck, and 6 DOF in the arms. The face is a silicon-rubber mask, which is supported by an aluminium frame. It has 2 DOF eyes fitted with video cameras; eye lids that can open and shut and a mouth capable of opening and smiling. KASPAR has several pre-programmed behaviours that include various facial expressions, hand waving and drumming on a tambourine that is placed on its legs[1].

Figure 1: KASPAR playing with one child

KASPAR is being used to study human-robot interaction as part of the European RobotCub Project, which aims to build an open-source robot platform for cognitive development research. The Adaptive Systems Research Group is investigating the use of gestures, expressions, synchronization and imitation.

KASPAR is currently also being used also in studies working with children with autism as part of the European FP6 IROMEC project and the European FP7 ROBOSKIN project. The IROMEC project acknowledges the important role of play in child development and targets children who are prevented from playing, either due to cognitive, developmental or physical impairments which affect their playing skills. This IROMEC project investigates how robotic toys can empower these children to discover a range of play styles from solitary to social and cooperative play. The ROBOSKIN project aims to develop new skin sensor technologies that can provide detailed tactile feedback from large areas of the robot body. The team at UH is currently developing cognitive mechanisms that use this tactile feedback to improve human-robot interaction capabilities and implementing this technology in the domain of robot assisted play for children with autism[2].

There are several versions of KASPAR, one of which has tactile sensors. More information about the tactile version of KASPAR along with other robots that use tactile sensors will be discussed later in this article[3]. The technology used in the tactile version of KASPAR was capacitive pressure sensors. However the approach used in this work employed resistive pressure sensors.

1.2 Project

The aim of this project is to explore the possibility of using FSR sensors to provide KASPAR with tactile feedback in order to improve the human-robot interaction. This project took place during a summer internship completed by Guillermo Barbadillo under the supervision of Luke Wood and Kerstin Dautenhahn.

2 FSR Sensors

2.1 How do they work?

Force Sensing Resistors (FSR) are a polymer thick film (PTF) device which exhibits a decrease in resistance with an increase in the force applied to the active surface[4].

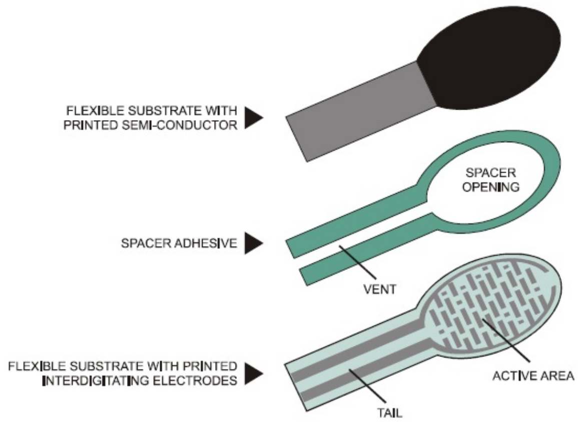

Figure 2: FSR Construction

When force is applied to the FSR sensor the gap between the conductive elements decreases and so does the resistance of the sensor. The resistance range of the sensor is typically from >1MW at no load to approximately 1kW at full load.

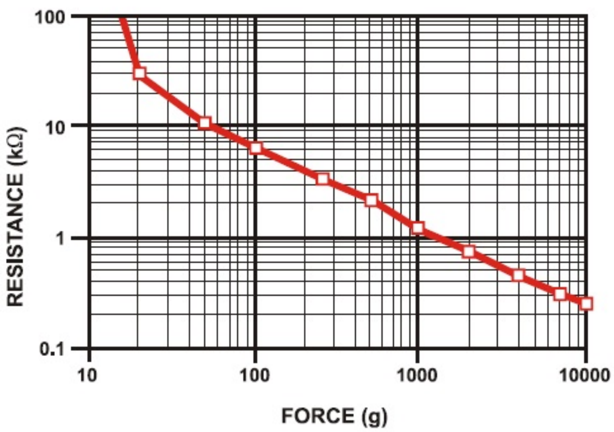

Figure 3: Resistance vs force

2.2 Types of FSR sensors

In this development two different types of FSR sensors were used, large and square sensors (FSR 406) and small round sensors(FSR 402). The smaller sensors were used in the hands, shoulders and head. The majority of sensors were the larger ones as locating the exact point of the physical contact was not relevant for the envisaged applications, the primary focus was to detect if there was contact or not with particular parts of the body.

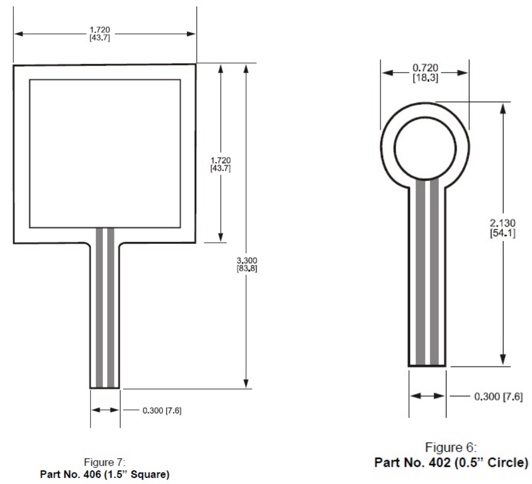

Figure 4: The two types of sensors used

There are many other different types of FSR sensors and custom FSR sensors could be manufactured where necessary.

2.3 Advantages and Disadvantages

The main advantages of FSR sensors is that they are cheap, self-adhesive and very robust enduring up to 10M actuations. FSR sensors are simple variable resistors, thus the circuit used for reading the resistor value is summarily very simple.

However the FSR sensors have a very long tail and careful consideration needs to be taken at the design stage to ensure there will be enough space for the tail connector. Theses sensors also have the disadvantage that if you blend them they change their resistor values. In addition, it was very hard to find a suitable connector for the sensor, and applying the sensor to curved surfaces can cause pre-loading. At the end of this paper a list detailing all the materials used in this work.

2.4 How to glue them and some tips

Placing FSR sensors on curved surfaces is possible provided they only have one direction of curvature and the diameter is not very small. However, if the surface is not regular the sensor will be pre-charged and will loose precision. It was necessary to glue the sensors on irregular surfaces as the feet and torso of Kaspar are irregular surfaces. The solution selected was to glue the sensors on rigid card with double-sided tape and then glue the card to the robot using a glue gun. In this way a flat surface had been created for the sensors but it was not possible to follow the natural shape of the body. Other type of sensors would be more effective when working over curved or irregular surfaces.

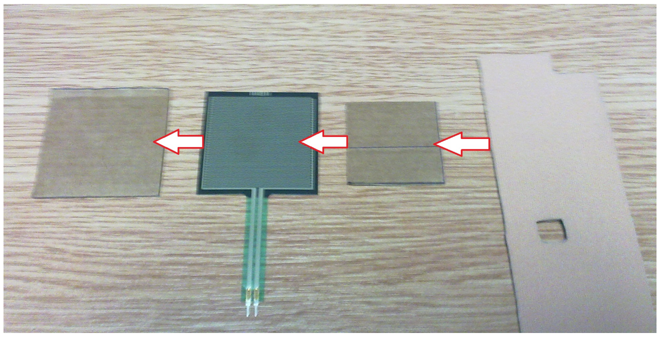

Figure 5: Material used to paste the sensors

All the sensors were covered with neoprene foam. This foam was chosen because of its minimal thickness (only 3mm) and also because it was flesh coloured. The foam was glued to the body using a glue gun. This provides robust performance along with the ability to easily remove the sensors using a screwdriver or a cutter. The foam provides protection to the sensors without interfering with the readings.

Here there are some tips that can be useful:

• It is useful to check the value of the FSR resistor once it has been glued and cover with the foam. The fastest and easiest way to do this is using a polimeter. If the value is higher than 200k everything is OK, otherwise check that the foam is not applying pressure to the sensor or that the sensor is bent.

• The neoprene foam used for this work is highly recommended because it does not oppose to the pressure and translates all the force to the sensor.

• Sometimes is useful to glue a small rubber or card actuator above the sensor. This helps to distribute the pressure across the sensor and increases the repeatability.

2.5 Final Result

26 sensors were applied to the robot.

- Head: 2
- Shoulder: 2
- Torso: 4
- Arm: 8
- Hand: 6
- Feet: 4

The glued sensors can be seen below. Only the sensors of the hands remain uncovered by the clothes so these were the only ones which were designed with a rounded shape.

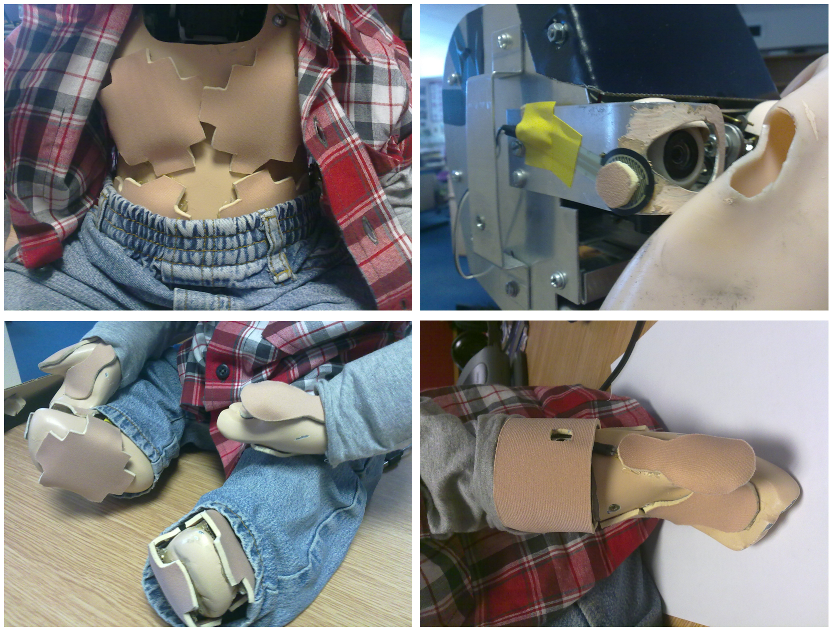

Figure 6: Sensors covered by foam

3 Electronic

3.1 Circuit used

In this document there are some suggested electrical interfaces to obtain the data from the sensors in the FSR Guide[4]. The simplest option was chosen for this work.

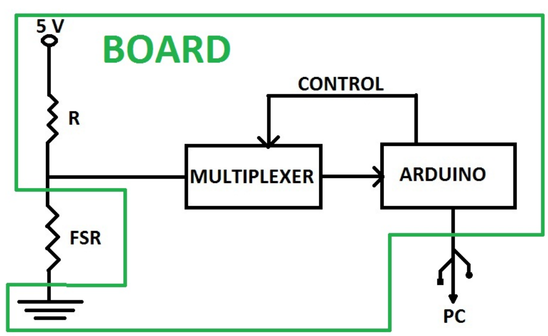

Figure 7: Scheme of the circuit used to read the FSR sensors

One resistor connected in series with the FSR sensor was used for this work. We read the voltage between the two resistors and sent it to a 8-multiplexer. Finally an Arduino board reads the output of the multiplexers.

3.2 Connections

As seen in the diagram, two wires per sensor are required. However, a common ground can be used for all the sensors and reduce the number of wires needed. Each multiplexer has 8 inputs so only 9 wires are required for each multiplexer, however two ground wires were used so there were 10 wires per multiplexer.

The FSR sensors cannot be soldered directly to the wires, therefore using the proper connectors is crucial for building a robust system. The best option is to solder the connectors to the wires and protect them with heat shrink.

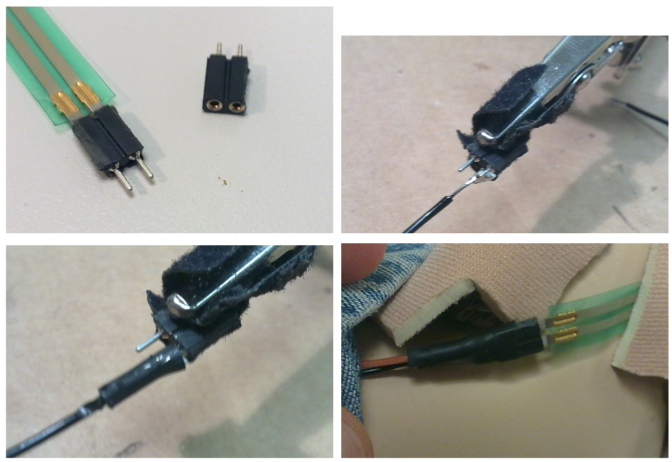

Figure 8: The making of the connectors

3.3 Multiplexer

In electronics, a multiplexer or mux is a device that selects one of several analogue or digital input signals and forwards the selected input into a single line. This was the best way to gain access to the sensors because we were going to use up to 32 sensors. In this work 4 multiplexers were used with 8 inputs each.

3.4 Resistor

The value of the resistor determines the range of weight that can be read by the Arduino board. We decided to use a 10K resistor because it will enable us to read forces of 20g or less. Choosing a higher value resistor will give the ability to read smaller forces, but would decrease the ability to distinguish between greater forces. Network resistors were used because they are small and fitted perfectly next to the multiplexer.

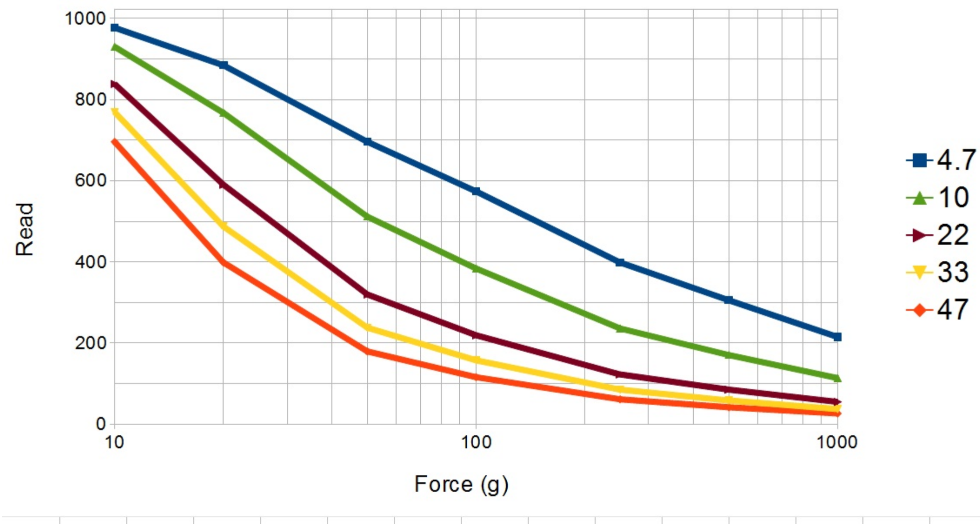

Figure 9: Read graphs of using different resistors

3.5 Arduino

In this work 4 multiplexers were used which are controlled by an Arduino board. The Arduino board selects the channel of the multiplexer and also reads the output. 3 digital outputs are used to select the input of the multiplexer and 4 analogue input to receive the signal from each multiplexer. The Arduino board is simple to program and very powerful, it is highly recommended because it is well documented online[5].

3.6 Board

The circuit was implemented on a prototyping board. Network resistors were used because it was the best solution for improving the use of space. A complete list of the materials used can be found at the end of this paper.

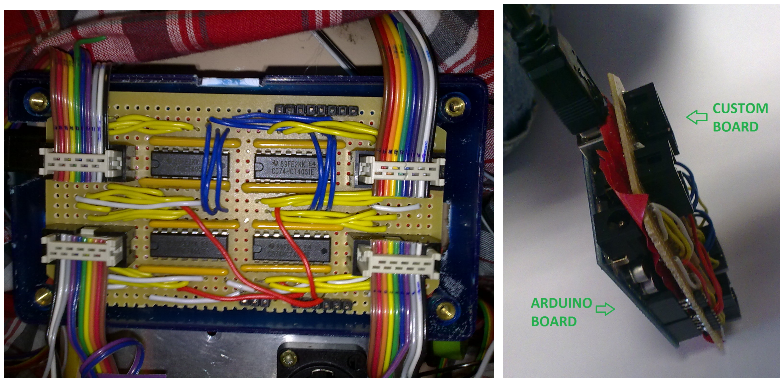

Figure 10: The boards

4 Software

4.1 Arduino Software

Arduino provides an environment to program the board in a programming language similar to C. The board was programmed to do these steps in an infinite loop:

1. Send a start byte (0xAB) through the serial port. Thus we know that we are going to receive the sensor data next.
2. Select the channel of the multiplexer
3. Select a multiplexer
4. Read the 10 bit value from the sensor
5. Split the value into two bytes
6. Send the two bytes value through the serial port.

We were able to work with a baud rate of 19200 without having problems. We reached a frequency of approximately 30Hz, which was found to be a very good speed. However, the speed can be doubled if necessary by sending only the 8 bigger bits from the ADC.

The Arduino environment also allows the data from the serial port to bee seen. However, it is in plain text and that's not the better way to see the values of 32 sensors. Therefore we decided to create a program to visualise the values in a more intuitive environment.

Detailed documentation of Arduino can be found online[6].

4.2 Visual Basic Software

We decided to use Visual Basic because communicating with the serial port is simple and the easiest way is using a default SerialPort object[7]. We used Visual Studio 2010 because it has some objects that allow development of a good looking program fast, and we were looking to make a prototype of the program.

The program reads the bytes sent by the Arduino board and waits until it gets the start byte (0xAB). It then stores the next 64 bytes and finally it calculates the 10 bit value from each sensor.

Later we included the possibility of filtering the FSR data. A very simple yet powerful filter was used which worked very well:

$$FilterData = PostScaler \cdot (RawData - Offset)$$

To obtain the filter's value it was only necessary to measure the offset. This was achieved by reading the FSR value for 10 times and choosing the smallest value. Then the postscaler is calculated as shown below:

$$PostScaler = \frac{1023}{1023 - Offset}$$

Thus the filter signal has the same original's signal range (0-1023).

For the interface a picture of a boy was used in the background and progress bars above it in the relevant places. When the sensors were pressed the progress bars fill, and there is also a chart to follow the variation of the sensors throughout time.

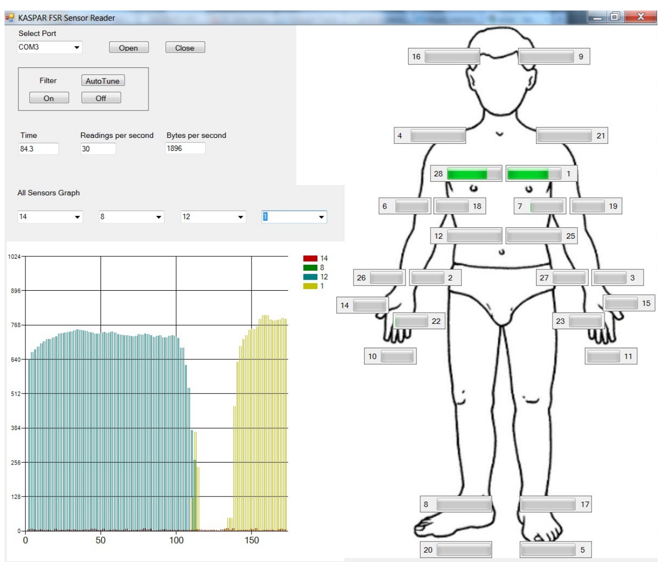

Figure 11: Picture of the software running

4.3 Java Software

There is a Kaspar GUI to control the robot, it is therefore necessary to integrate the FSR sensors into the program. Working with the serial port in java it is not as intuitive as in VB but it is still relatively easy. We worked on the base of a class described in this page[8].

The Kaspar GUI was created with Netbeans so we continued to use it to create the FSR module. We created a class that reads the values from the Arduino board and also filtered them if necessary. Finally an additional tab on the original program was created to see the values from the sensors.

5 List of Materials

FSR 406 The big and square sensors.

FSR 402 The small and round sensors.

4609X-101-103LF The resistors used in series with the FSR sensors.

CD74HCT4051E The multiplexer used to receive voltage values from different FSR sensors.

008290010001011 Socket for 10 wires.

MC9A12-1034 Header for 10 wires.

RE520-HP Prototyping board

135-2801-010 10 way cable.

801-43-032-10-001000 The connectors for the FSR sensors. They were very difficult to find.

Arduino Board Simply and powerful.

SMALL Neoprene Sheets 3mm Double Lined 300mm x 300mm - Flesh Coloured Used to cover the sensors

Double sided tape

Glue gun

Rigid Card

6 Contact

- Kerstin Dautenhahn

k.dautenhahn@herts.ac.uk

- Luke Wood

l.wood@herts.ac.uk

- Guillermo Barbadillo

guillermobarbadillo@gmail.com

Contact me if you want to see the source code of the programs, also you can download it directly from http://www.megaupload.com/?d=06AAMWTE

References

[1] Michael L. Walters Ben Robins Hatice Kose-Bagci N. Assif Mirza Mike Blow Kerstin Dautenhahn, Chrystopher L. Nehaniv. Kaspar a minimally expressive humanoid robot for human- robot interaction research. *APPLIED BIONICS AND BIOMECHANICS*, 2009.

[2] http://kaspar.feis.herts.ac.uk/. Kaspar kinesics and synchronisation in personal assistant robotics.

[3] Marco Maggiali Lorenzo Natale Giorgio Cannata Alexander Schmitz, Perla Maiolino and Giorgio Metta. Methods and technologies for the implementation of large-scale robot tactile sensors. *IEEE TRANSACTIONS ON ROBOTICS*, 27, 2011.

[4] http://www.media.mit.edu/resenv/classes/MAS836/Readings/fsrguide.pdf. Fsr integration guide.

[5] http://www.arduino.cc/. Arduino webpage.

[6] http://www.arduino.cc/en/Reference/HomePage. Arduino language reference.

[7] http://www.electronicsblog.net/digital-voltmeter-arduino-ant-pc-visual-c-comunication-via-serial port/. Arduino and pc (visual c++) comunication via serial port.

[8] http://www.arduino.cc/playground/Interfacing/Java. Arduino and java.