Redesign of the E-Cone: A tool for treatment of hand disease

Haitao Wang, h.wang-1@student.utwente.nl, s1638874, MSc Embedded Systems

***Abstract*—E-Cone is a medical device that was developed to help patients with hand diseases exercise their hands and better distribute their hand force. Since the previous E-Cone had a lot of disadvantages, we redesigned the whole device with new hardware and software with cheaper solutions. First, we built our own pressure sensor with higher resolution than the previous one. Then, we designed a new control board with a new microcontroller and other electronics. We implemented wireless data transmission method instead of using the wired USB. After rewriting the software part, the new sensor is able to capture pressure image of our hands properly. Unlike the previous way using a specific software installed on PC to display the pressure image, a web server was set up on the new controller. We could access to the real-time pressure image through a web browser on any device. Analysis of the new sensor and software was made and existed problems were also pointed out. Yet, the final output of our work is a new pressure image acquisition system with stable function. The complete product is waiting to be finished.**

***Index Terms*—Hand disease, grip pattern, pressure sensor, finite element analysis, Raspberry Pi, webserver.**

I. INTRODUCTION

HAND function is very important to us both in work and our daily lives, especially in such a modern world where smartphones and computers are highly involved. However, there are many people who are suffering diseases and injuries cannot use their hands fully. For example, people with osteoarthritis (OA) and rheumatoid arthritis (RA) [1-2] are frequently suffering from an affliction of the joints of the hand and surrounding tissues consisting of pain and swelling, which causes decreased physical function when they perform activities requiring their hands. Sports injuries like muscle strains can also cause overall hand weakness and might eventually lead to loss of hand functions.

If left untreated, the long-term implication of reduced hand function will lead to serious consequences in a person's life. Fortunately, as long as the function disorder of the hands is not permanent, conditions can become better if patients are guided to exercise their hands frequently.

In order to help patients with hands diseases to exercise their hands and recover from disabilities, various rehabilitative devices show up in the market. From the simplest stress ball[1] to high-tech products with sensor technology [3], patients do have many choices. However, these devices also have disadvantages. Simple device cannot quantify the grip strength from which we can adjust our exercise strategy. High-tech products, normally, are very expensive and not handy to use. Thus, we are looking for such a device that can give us enough feedback and can be used easily.

Researchers have observed one phenomenon that patients with hand disabilities cannot distribute the pressure properly when handling objects. If the patient can be guided to better distribute their hand force, they will be able to use their hand more freely and can gradually recover. The best way to guide patients to use their hands is to have the grip pattern visualized. Then, a treatment can be developed to help patients better distribute their force according to the visualized pattern. The E-Cone[2] was developed based on this idea in 2011[4]. The E-Cone, as shown in figure 1, consists of two parts, a cone-shaped carrier and software part on the PC. A sensor mat with a matrix of pressure sensors is attached to the carrier. The control box at the bottom contains electronics that are used to read the pressure from the sensor mat. The pressure can then be converted into an image and be sent to a PC through a USB wire for visualization. Therefore, the patient can simply grip the carrier and continually adjust his hand force distribution based on the real-time feedback from the pressure image on the PC.

Figure. 1. E-Cone and pressure image. Black is no pressure, blue light pressure, yellow more and red strong pressure.

Although the current E-Cone is competent, manufacturing one single device costs more than 1000 euros. We are very interested in using cheaper material to rebuild the whole system. We also would like to optimize the electronics and software to make it more convenient to use and user-friendly.

II. RESEARCH PLANS

Considering the disadvantages of the current device, we plan

[1] A stress ball is a rubber ball, usually not more than 7 cm in diameter. Patients can squeeze it in the hand and manipulate it using fingers to help exercise the muscles of the hand.

[2] The E-Cone is an instrument that enables real-time high-resolution visualization of the grip pattern of the hand. Because of the direct feedback to the patient, the patient than can be instructed to adjust the distribution of the force on the fingers and hand.

to optimize the E-Cone in the following aspects.

1)New shape: The current hardware contains a control box at the bottom of the carrier, which makes the device heavy and inflexible. The sharp corners of the control box can also be a potential safety problem to our patients. One optimization we'd like to make is removing the control box and put all the electronics into the carrier itself. By doing this, the device will only have the one-part carrier left, which will make it more lightweight. What's more, it will become much easier for the patient to hold the device and make any movement they want.

2)DIY sensor mat: The current pressure sensor we are using is bought from Tekscan Inc. [5]. The sensor is expensive, one single sensor costs 150 euros. The sensor has a resolution of 44 × 44, which is good but we 'd like to improve it more. Also, 44 is not a very reasonable number. The connector of the sensor mat is not customized. We had to design a special circuit board to match the connector. We tried to look for other sensors from different companies. However, those sensors are either of too big size or too low resolution. It is totally possible to develop our own pressure sensor. Paper [6] shows a method of making the such a sensor based on carbon black/silicone rubber nanocomposite. Therefore, we'd like to develop our own pressure sensor. On one hand, we could reduce the cost by using cheaper material. On the other hand, we could make a sensor with higher resolution and customized connectors.

3)Redesign electronics: Since the sensor mat will be replaced, the old control board cannot be used anymore. We need to modify the design of the electronics, add new components and remove unnecessary parts. In the end, we also need to design the control board into a proper size and shape to fit the carrier.

4)Wireless communication: The current communication method between the E-Cone and the PC is based on wired connection through USB. The wire limits the range of motion and also adds inconvenience for holding the carrier when the patients want to hold it in a different way. Thus, we plan to change the wired communication into a wireless one. Technology like WIFI, Bluetooth, and ZigBee can be our choices. We choose WIFI in the end. By connecting to the WLAN, we can monitor and control the device more easily.

5)New microcontroller: The current control board of the E-Cone uses an FPGA as the processor. FPGA is very fast due to the parallel processing, but it also has a relatively high price. Therefore, we plan to use a cheaper microcontroller which is still competent to do the job. After investigating, we choose to use Raspberry Pi Zero Wireless as the processor. Raspberry Pi is widely used in all kinds of intelligent systems [7]. The newest version Zero Wireless does have a lot of advantages. First of all, Raspberry Pi Zero is cheap, which only costs 10 dollars. What's more, the Raspberry Pi Zero is tiny enough to get into the carrier. Raspberry Pi Zero Wireless has WIFI and Bluetooth module on board, which makes it more convenient for us to implement the wireless communication. Unlike FPGA, Raspberry Pi has an operating system. Based on that, we could program and develop other function easily.

6)Web-based terminal: The current E-Cone system uses a client software to display the pressure image. We are going to show the pressure image on a website. It will become more convenient to use, after all every PC has a web browser. No software needed to be installed in advance, the only thing we need to do is open the web browser and type in the address. People can even use a smartphone to view the pressure image. Then, patients don't have to stay in front of a PC when they use the E-cone.

The main work of redesigning the E-Cone is to build our own pressure sensor, including the related electronics. Without the sensor, we wouldn't be able to continue the following work. Building the sensor is the only hardware part of the whole work, which is particularly important. Once the sensor is working, we start to implement the software and then make optimizations. In the end, we integrate everything into a working system.

From what has been discussed above, we can simply divide our work into three parts, as shown in figure 2.

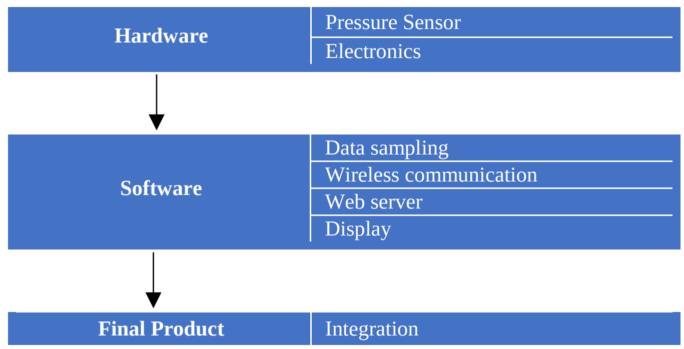

Figure. 2. Division of work

III. PRESSURE SENSOR

A. *Pressure sensor principle*

Paper [8] and [9] both illustrate the principle while applying a pressure sensor to a smart gun. The principle of the pressure sensor mat can be simply explained through figure 3. We use a small 4 × 4 array to illustrate the principle. The whole mat has three layers. On the top layer, we have four electrode strips that are placed horizontally, and on the bottom layer, there are four strips placed vertically. There is a thin mat which is pizoresistive between the two layers. In such a construction, a resistive element exists at each crossing point. The resistance of the element is sensitive to the pressure. Thus, we can measure the resistance to indicate the pressure. The resistance of the piezoresistive mat decreases with increasing pressure.

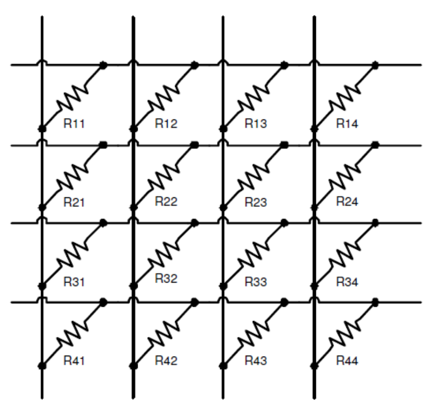

Figure. 3. Schematics of pressure sensor array

Each crossing point of the sensor array is referred to as one sensor cell. To measure the resistance of each sensor cell, we need an analog measuring circuit, like what's shown in figure 4. This figure shows the basic theory of our sampling circuit. The rows of the sensor array are connected to a 4-to-1 analog switch, and the columns are connected to four 1-to-2 analog switches. By select a specific row and a column and connect them into a circuit, we can calculate the resistance of a specific sensor cell. The selected column is the one which is connected to the test voltage V_{test}, other columns are connected to ground. The selected row is the one which is connected to the input of the amplifier, other rows are left open. In our circuit, R_{33} is the resistance of the selected cell. We could know the resistance(R_k) of the selected cell is proportional to the output of the amplifier V_{out}, which is:

$$R_k = -R_{33} \times \frac{V_{out}}{V_{test}} \tag{1}$$

By using analog multiplexers as switches, we can control them in time sequence. Thus, the sensor cells are able to be scanned in time sequence, we can easily get the resistance of every single cell. And a pressure map can be generated. The output of the amplifier is connected to the input of an AD converter, which transfers the data to the microcontroller for digital analysis.

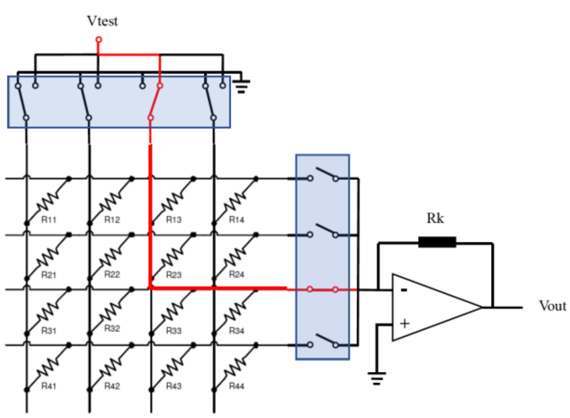

Figure. 4. Schematics of measuring circuit

B. Finite element modeling

In figure 4, we assume the current only flows through the sensor cell we selected. Other cells should have no influence or have little influence on the behavior of the selected one. But there does exist other paths for the current. In order to figure out what exactly happens when the current goes through the sensor array, we did some simulations by using the software Ansys[3].

1) Simulation with only 2 cells

At first, we made a model with only two cells to check whether the current will follow the correct path. As shown in figure 5. The model contains three layers. Two electrode strips lie horizontally on the top layer. One electrode that represents the selected column is set to 5 volts and the other is set to ground. The bottom layer is an electrode, which represents the selected row, placed in another direction. It has been set to the virtual ground, the same as the negative input of the amplifier. The inner layer is the resistive material with an electric resistance of 5000 ohm-m, which is a value between insulator and conductor.

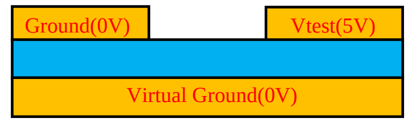

Figure. 5. The sensor model with only two cells

Figure 6 shows the calculated current density after finite element simulation. We can see that current only flow through the selected cell. In the simulation of figure 6, the resistance of the inner layer is fixed. We all know that when we press the selected the cell, the resistance of that area should decrease. Thus, we reduce the resistance of that area to 2000 ohm-m to see the changes. The new result is shown in figure 7. The current density gets enhanced. According to the two simulations, we draw the conclusion that the current will flows through the selected cell with little leakage.

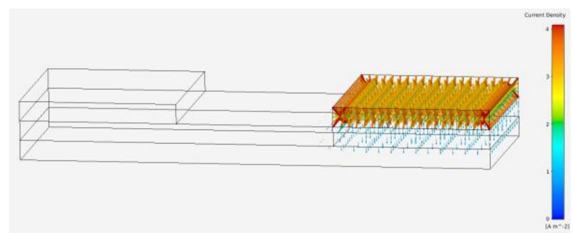
Figure. 6. The current density of a 2 × 1 sensor. (without force applied)

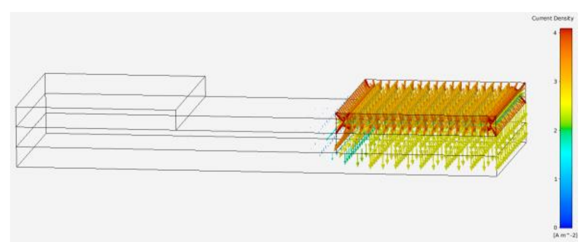
Figure. 7. The current density of a 2 × 1 sensor. (with force applied)

2) Influence of the geometry of the resistive material.

In the experiment above, the current has two paths to go, one to the ground and another to the virtual ground. But the results show that almost all current chose the second path. The reason for this is that the current density is influenced by the geometry of the resistive material. Compared with the thickness of the material, the contact area at the crossing point is much bigger. If the size of the cross section is relatively the same as the contact area, the result will be different. To prove that, we had another simulation with a model that has a much thick inner layer. Figure 8 shows the result. We can see a lot of current goes

[3] Ansys is a software used for finite element analysis in different physical fields.

to the top left electrode. Although the main part of the current is still on the correct path, we cannot ignore the current that goes to the other way. To get a better result, the thickness of the resistive material we use must be small enough. In another word, we cannot make the width of the electrode too small.

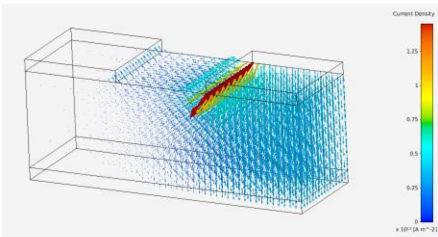

Figure. 8. The current density of a 2 × 1 sensor with a thick inner layer

3) Simulation with an 8 × 8 sensor array.

The conclusion gotten from the previous simulation is that the current follows the way we expect. But we would like to add the number of rows and columns to prove the universality of the result because things might be different for more sensor cells. Thus, we made an 8 × 8 sensor model as shown in figure 9. We set the width of the electrode as 0.8mm and the same value for the gap between electrodes. The reason for choosing such a small value is that the width of a real product will be less than 1mm if we want to reach a much higher sensor resolution and keep a proper size at the same time. The thickness of the inner material is set to 0.1 mm to make sure the current in the material won't flow in the horizontal direction.

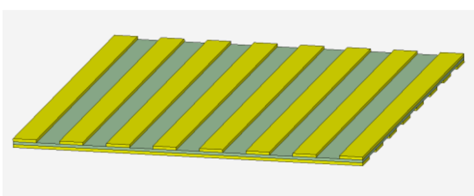

Figure. 9. The 8 × 8 sensor model

By selecting the second column from the left on the top layer, and the second row from the bottom on the bottom layer. We get the current density as shown in figure 10. We can see that most of the current flow through the selected sensor cell. Although leakage current exists, they are little compared with the main part. We can ignore the leakage current.

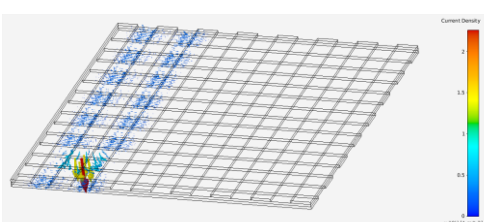

Figure. 10. The current density of the 8 × 8 sensor model

Since the simulation of the 8 × 8 sensor meets our expectation, we can safely extend the result to a higher resolution. Because it's only a matter of adding more rows and columns. From all the simulations made above, we can ensure that making our own sensor mat is theoretically feasible.

IV. Pressure sensor Experiments

The simulations showed that the sensor should work in a proper way. But nothing could be guaranteed without real experiment. In order to find out what's the actual behaviors of the sensor will be, we have made some sensor prototypes and have done several experiments.

A. *Two by two sensor experiment*

To start from the most basic situation, a 2 × 2 sensor array was made at first, as shown in figure 11. For the conductive electrode strips, the brass foil was used. The pressure sensitive layer is a whole piece of resistive sheet as shown in figure 12. All the electronics needed here is an amplifier. The rows and columns are selected manually by using jumpers. We use a multimeter to measure the output voltage of the amplifier while we are pressing the sensor.

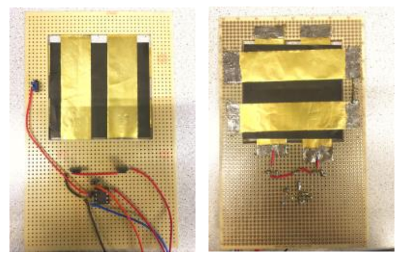

Figure. 11. The two by two sensor array

Figure. 12. The pressure-sensitive conductive sheet

We get several findings by playing with this self-made sensor.

1) With the increase of pressure, the output voltage (absolute value) of the amplifier increases. That means the output current increases which also means the resistance of the selected cell drops.

2) For a specific selected sensor cell, the output of the amplifier won't change if we press its surrounding cells. That means when we select the first row and the first column, the output voltage will change accordingly if we apply force to the top left sensor cell. But nothing would happen if we press other cells.

3) For this sensor, whether we use one whole conductive sheet for all crossing points or separate sheets for each, like what's shown in figure 13, the output stays the same. This means that the current flow follows the path we want it to follow. The leakage current to other paths is very small. But this finding only applies to this sensor array here whose sensor cells are of a relatively big area.

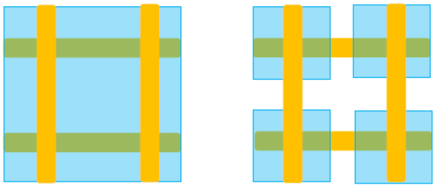

Figure. 13. One whole conductive sheet and separate sheets

4) Pressing only one sensor cell that is not selected won't affect the output. However, pressing the selected cell and the other cell on the same row at the same time, the output would be smaller than that of pressing selected cell only. The reason will be discussed in Chapter VI.

5) The output current of the amplifier has a limit. If you press the sensor too hard at a big test voltage, the current will exceed the maximum value and the voltage will rebound. We have to be very careful about the test voltage V_{test} and the feedback resistance $R_{k.}$ because by adjusting these two values we can avoid the problem.

B. Eight by eight sensor experiment

The experiment above shows that the sensor we made is functional and it behaves as we expect. In order to get a better idea of how the sensor behaves with more sensor cells and to get familiar with the working ways of multiplexers, we made an 8 × 8 sensor array. The sensor works well just like the 2 × 2 sensor. We can already generate a pressure image from this sensor. We use Arduino as the control unit and three multiplexers for cell selecting. Figure 14 shows the 8 × 8 sensor and the measuring circuit.

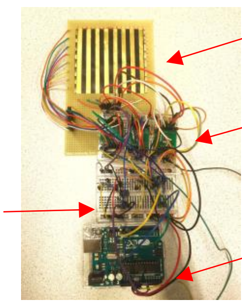

Figure. 14. The eight by eight sensor and the measuring circuit

We used one 1-to-8 multiplexer for the rows to choose one row to be connected. Two 2-to-8 multiplexers were used for columns to choose from being connected to V_{test} or to the ground. According to the output value of the AD converter, we use different colors to indicate the pressure magnitude. The generated pressure image is shown figure 15. Here we just use Processing[4] as the software to display the image. By testing the 8 × 8 sensor we built by ourselves, we already have an overall view of our system. And the test results show that building a high-resolution sensor by ourselves is feasible.

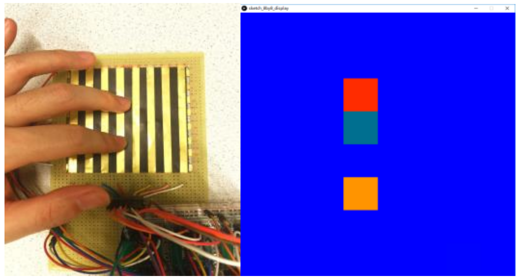

Figure. 15. Pressure image generated from the 8 × 8 sensor matrix

C. 64 by 64 sensor

The current sensor bought from Tekscan Inc. has a resolution of 44 × 44. Since it's already competent to do its job, we don't need to make a new sensor with much higher resolution. And in order to have a more reasonable number than 44, we decide to build a sensor with a resolution of 64 × 64. The new sensor's resolution is about 50% higher than the old one. Also, 64 is a much reasonable number for us since we are using binary code to control the multiplexers.

The second finding in section A shows that pressing sensor cells except the selected one has no influence on the output. The Situation might be different for a sensor with higher resolution. But according to our simulation in Chapter III, there should not be any problem. In our simulation, the width of the electrode is 0.8 mm and the thickness of the resistive material is 0.1 mm. We have the exact same parameters for our 64 × 64 sensor.

We use flexible printed circuit board (FPC) to build the sensor because the minimum thickness that manufactures can produce nowadays is as thin as 0.1 mm. After designing and manufacturing, we got the prototype of our 64 × 64 sensor mat, as shown in figure 16.

Figure. 16. The 64 × 64 sensor mat

The resistive material we use is the same as what we use in the previous experiment. It has a thickness of 0.1mm. Plus the thickness of the FPC, the sensor itself should have a thickness of 0.3mm. The resistive sheet has a volume resistivity of 500 ohm-cm and surface resistivity of 31000 ohms/sq.cm.

One of our goals is to make the new sensor cheaper than the previous one. The previous sensor was bought from Tekscan Inc. Each sensor costs €150. The sensor we made consist of three parts, two flexible printed circuits and the resistive

[4] Processing is a flexible software sketchbook and a language for learning how to code within the context of the visual arts.

material. The FPC cost €400 for 3 prototypes, so each cost about €133. The resistive material only costs €3.95 for a dimension of 280mm x 280mm. Thus, we can see the self-made sensor is cheaper than the previous one. And this is only for the prototype. If we are going to use it for the real products, it could be much cheaper.

D. *The electronics for the 64 by 64 sensor*

1) Microcontroller

We choose Raspberry Pi Zero Wireless as our processor. The initial reason we choose Raspberry Pi Zero Wireless is because of its tiny size. With a dimension of 65mm × 30mm, we can easily get it into the body of the cone. Raspberry Pi Zero Wireless is also very cheap, which makes it a better choice. Compared to its previous version Raspberry Pi Zero , it has WIFI and Bluetooth on board. We can use these wireless communication methods to do a lot of extensions to our device. And the most important advantage, compared with other general microcontrollers, is that it has its own operating system. It makes the programming process much easier.

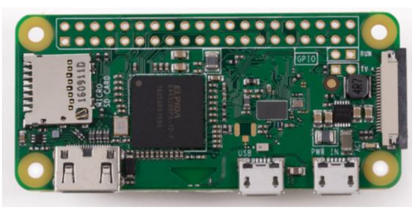

Figure. 17. Raspberry Pi Zero Wireless

We investigated other microcontroller boards as well, like BeagleBoard, Gumstix, UDOO, and CHIP. But those boards have many disadvantages compared to the Raspberry Pi Zero. BeagleBoard has a big size and its storage is very limited. Gumstix is of a small size, but it doesn't have IO pins on-board. UDOO is compatible with Arduino. People who are familiar with Arduino can directly use it as an Arduino board and program it in old ways. But it also has a big size. What's more, these three kinds of boards are very expensive. CHIP doesn't cost much and has a very small size. However, the on-board storage is limited. Combined, Raspberry Pi Zero Wireless is the best choice.

2) Multiplexers

The multiplexer we choose is ADG739. It is a CMOS analog matrix switch with dual 4-channel. It has an 8-bit shift register to control the states of the 8 switches, as shown in figure 18. The maximum switch speed can reach 30MHz. For the rows of the sensor, we need to connect one of the 64 rows to the input of the amplifier and others keep open. That is one switch for one row. Thus, 8 multiplexers are needed for the rows. For the columns, we need to connect each column either to V_{test} or ground. That is two switches for one column. 16 multiplexers are needed for the columns. The output of the shift register can be connected to the input of register of the next one, which makes it possible to have a number of these multiplexers daisy-chained. Using SPI protocol, we can easily control the states of each switch by shifting 8 bytes for rows and 16 bytes for columns.

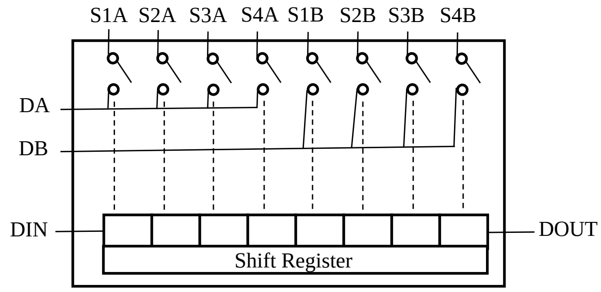

Figure. 18. Principle of the multiplexer

3) Analog-digital converter

The ADC we use has a resolution of 16-bit. The external clock cycle ranges between 24kHz and 2.4MHz. The maximum throughput rate is 100KHz. The ADC can also be controlled with SPI, which can be shared with multiplexers.

After designing and manufacturing, we got the prototype of our electronics board, as shown in figure 19.

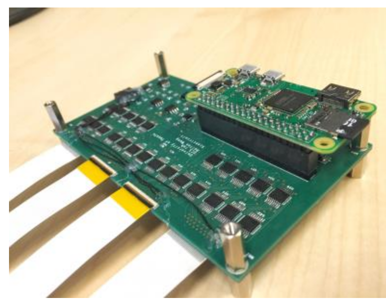

Figure. 19. Electronics board.

E. *64 by 64 sensor experiment*

To reduce the cost of the prototype, we use FPC cable to connect the sensor to the electronics board. Figure 20 shows the combination of the sensor and electronics board, which makes the final hardware system of a working pressure sensor.

Figure. 20. Electronics board.

After finishing the software part, which will be discussed in Chapter V, we get the pressure image. Figure 21 shows the pressure image of a hand. Different color represents different force level.

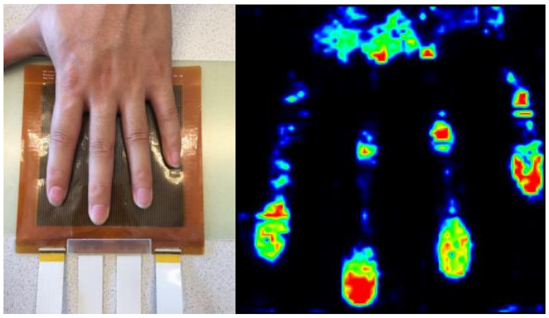

Figure. 21. Pressure image of a hand.

V. SOFTWARE DESIGN

The sensor should be sensitive to the pressure. The sampling circuit should be able to generate a corresponding output and send it to the controller. That's all the responsibilities of hardware. When it comes to software, it should be responsible for converting all the sample values to a readable format and convert it into an image. A pressure image is more intuitive than a bunch of numbers or a curve diagram. The generated image should be transmitted to a computer and be displayed. The overall working process of software is illustrated in figure 22.

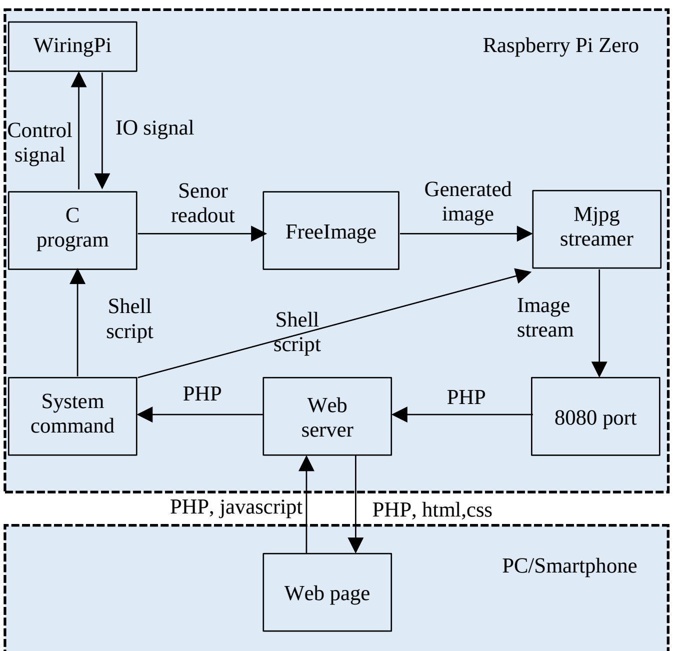

Figure. 22. Working process of software.

At first, our C program sends signals to the GPIO of Raspberry Pi to control the states of multiplexers by using WiringPi library, which is a pin based GPIO access library written in C for Raspberry Pi. Also, IO signals read from ADC can be captured. After processing those captured signals we get sensor readout. FreeImage, which is a library for processing images in all kinds of formats, is used to generate images continuously. Those images are sent to the 8080 port of the processor in stream by using mjpg-streamer. Mjpg-streamer is a command line application that copies JPEG frames from one or more input plugins to multiple output plugins. The pre-installed web server reads the image stream by access 8080 port. Then, a web page is built up with the image shown on it. We can access the web page from any device that has a web browser. In order to make the sampling process controllable from the client interface, on the server side we use PHP file to react to the request from the web page. This PHP file can call system command to control the running state of the sampling process. So far, the whole software loop is presented. We are going to discuss details in the following sections.

A. *Sensor scanning*

We write a program to control the multiplexers and sample the pressure value. Apparently, we cannot get the resistance of every sensor cell at the same time, we are only allowed to have one row and one column to be connected to the sampling circuit each time. To get the resistance of each sensor cell and generate a pressure image, we have to scan the sensor array in a certain way. What's more, to have a real-time response, we have to scan the sensor matrix fast enough.

Notice the multiplexers are daisy-chained. In total, we have 24 multiplexers, 8 for the rows and 16 for the columns. We chain these two set of multiplexers separately on purpose when we design the hardware. Technically, we can get the 24 multiplexers in one chain and use only one sequence of bits to set them up, like what's shown in figure 15. Although it is easier to implement, it's inefficient. Every time we update the states of multiplexers to select one sensor cell, we don't need to update all switches. By separating the rows and columns, we just need to update the rows and keep the column unchanged for all sensor cells in one column. We calculated the processing time for both methods to illustrate the difference. In our case, we have 64 rows and 64 columns. By writing a nested for-loop, we have to do the switches updating process for 4096 times. Assuming the time used to transmit one byte of data is T. For the first method, we transmit 24 bytes each time.

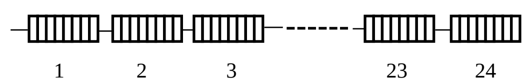

Figure. 15. Twenty-four multiplexers connected in one chain.

Total time needed for one scanning of all sensor cells is:

$$4096 \times 24 \times T = 98304\text{T}$$

For the second method, we choose the column first and keep it unchanged. So, we transmit 16 bytes for columns and we only do this for 64 times. We transmit 8 bytes for rows and this process happens 4096 times. In total, one scanning of all sensor cells takes:

$$64 \times 16 \times T + 4096 \times 8 \times T = 33792\text{T}$$

We can see, the second method is 2.9 times faster than the first method.

Although we speed up the scanning process, that is not the best we can do. In the previous chapter when we introduce the multiplexer, we mentioned that for each row only one switch is needed. It is either connected to the input of the amplifier or kept open. Assuming we have 8 rows in total, the first row is selected. Then, the value of the shift register should be "10000000" where 1 represents switch on and 0 represents switch off. "01000000" for the second row and "0010000" for the third row. You may notice we don't need to shift a totally new 8-bits value to the register, we only need to shift one bit "0". Thus, a new scanning time can be calculated if we choose to implement this method. That is:

$$64 \times 16 \times T + 4096 \times 1 \times T/8 = 1536\mathrm{T}$$

Which is 64 times faster than the first method and 22 times faster than the second method. That's a considerable improvement and will make our sensor more sensitive.

Table 1 shows the real difference between those three methods after the experiment, from which we can see the improvements are smaller in practice than that in theory. The third method is 13 times faster than the second method and 37 times faster than the first method.

TABLE1
SPEED COMPARISON BETWEEN DIFFERENT SCANNING METHODS

First method	Second method	Third method
585955us	205209us	15787us

The time indicates the time for one iteration of scanning all the 4096 cells.

B. Image generation

The general idea about generating a pressure image is that for every value we read from the ADC, we convert it into RGB value for one pixel. Since we only have 64 × 64 sensor cells, we generate a 64 × 64 pixels image. The RGB value of each pixel is converted from the ADC output of the corresponding sensor cell. We'd like to let the color changes from blue (low pressure) to red (high pressure). The ADC has a 16-bit output, whose output value ranges from 0 to 665535. We simply divide the range into 6 parts. In each part, how the RGB changes regarding to the ADC readout is shown in Table 1.

TABLE 2
RGB ASSIGNMENTS

ADC readout	RGB	Color
0 ~ 12000	0.0.0 ~ 0.0.255	Black ~ Blue
12000 ~ 24000	0.0.255 ~ 0.255.255	Blue ~ Cyan
24000 ~ 36000	0.255.255 ~ 0.255.0	Cyan ~ Green
36000 ~ 48000	0.255.0 ~ 255.255.0	Green ~ Yellow
48000 ~ 60000	255.255.0 ~ 255.0.0	Yellow ~ Red
60000 ~ 65535	255.0.0	Red

To get a complete pressure image for one complete scanning iteration, we draw the image pixels in the sequence of scanning. At first, we allocate memory for a 64 × 64 image. Every time we select one row and one column, we sample the ADC output and then set the corresponding pixel to the correct RGB value. After 4096 iterations, every sensor cell has been sampled and every pixel of the pressure image has gotten its value. Then we write the image to a jpeg/png file. The whole process is finished in a fast speed. By choosing the fastest scanning method in section A, it only takes 60ms to generate one single image. It only 6ms to write the image into a file. By generating images continuously, we can reach a frame rate up to 15fps.

C. Image interpolation

By applying the method mentioned in section B, we can generate a pressure image like what's shown in Figure 23.

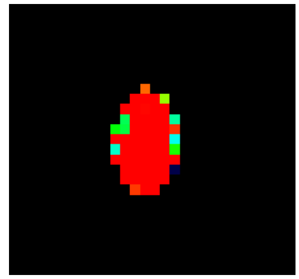

Figure. 23. Pressure image of a fingertip

As you can see, the displaying result doesn't look nice. Although our pressure sensor has a relatively high resolution of 64 × 64, it would be terrible if we simply display the image in the same resolution. To get an elegant, good-looking image, we have to apply interpolation to the original image. Another advantage of using interpolation is that we can eliminate the noise at the same time. In our case, we extend our image to a resolution of 256 × 256.

There are many interpolation methods we can use. We have tried bilinear interpolation and bicubic interpolation. And the results are almost the same for our image. Figure 24 shows the result. However, with the increasing of the image resolution, the frame rate drops. That's because a bigger image needs to be write to the file, which takes more time. Also, the calculations of the interpolation take a lot of time. The calculations of the bicubic method are more complex than that of the bilinear interpolation, so it takes more time. And other interpolation methods like Lanczos sampling is even more time-consuming, but it, of course, can offer a better result. In our case, the bilinear interpolation is already good enough. We don't need to sacrifice the frame rate for a better interpolation method.

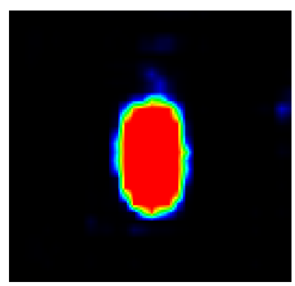

Figure. 24. Pressure image of a fingertip after interpolation

D. Mjpeg streamer and web server

From what has been discussed in section B, we can continuously write images into a specific directory. What Mjpeg-streamer does is watching the directory all the time. At any time it detects a new image file was written to the directory, it will stream that file. Mjpeg-streamer then can start a small web server and send the stream to the 8080 port of the Raspberry Pi. We can use any device to connect to the IP address of the Raspberry Pi and watch the image stream at 8080 port. Mjpeg-streamer uses plugins to do all the work. For example, "input_file.so" is the plugin used to watch a directory and "output_http.so" is the plugin used for http streaming. To run the streamer, only two lines of command line are needed. We can write a simple shell script to do this.

Normally, the web server is enough to help us watch the stream. However, we cannot run a PHP script on this server. To make the software easier to use, it would be great to control the software from the client side which is the web page. PHP is necessary for controlling the backend from the frontend. Thus, we established another web server to do the job. For Raspberry Pi, we choose Nginx as the web server because it requires little resources. The working flow of this part can be shown in figure 25.

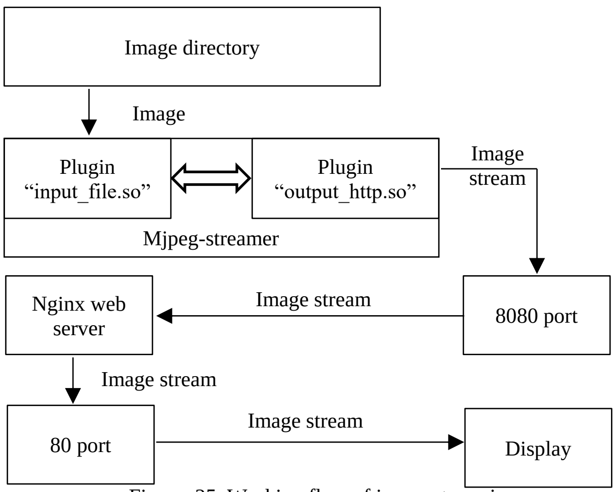

Figure. 25. Working flow of image streaming

E. Website Display

Once we get the streamer and web server work, we can watch the image stream on a PC or a smartphone. After writing some basic web files, we get a simple web page. Figure 26 shows the web page on a PC browser and figure 27 shows that on a smartphone browser. With the button “run”, we could execute the sampling and streaming program and show the pressure image at any time. With the button “stop”, we could shut down all the related running program. Compared with the display method used on the previous E-cone, using the web to show the pressure image is much more convenient. We are no longer constrained by the USB wire. And we don’t need to install the displaying software on every device. As long as the device has a web browser and connected to the same network of the E-cone, we can watch the pressure image easily.

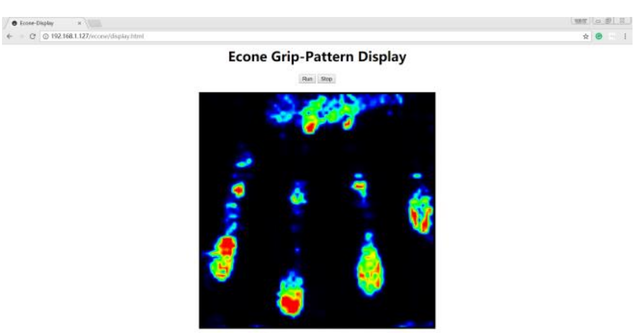

Figure. 26. Pressure image shown on a PC web browser

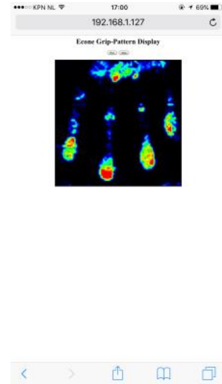

Figure. 27. Pressure image shown on a smartphone browser

To make the website work, several web technologies have been applied. The basic element showing on the web is described in a html file. By writing a CSS file, we unified the style of the web page. The interaction between the web page and the server is based on javascript and PHP. After one button is clicked, javascript responds to the action and another PHP file is activated. Then the PHP file calls system command to run the shell script, which controls the sampling and streaming program. Table 3 shows the related files on the server side.

TABLE 3
DISPLAY RELATED FILES

File name	FUNCTION
display.html	Web page content
display.css	Web page style
start_econe.php	Being called to run the shell script “start_econe.sh”
stop_econe.php	Being called to run the shell script “stop_econe.sh”
start_econe.sh	Starts the sampling and streaming program on the server side
stop_econe.sh	Stops all the related programs on the server side

VI. PRESSURE SENSOR ANALYSIS

A. Sensor Resolution

Apparently, the resolution of our sensor is 64 × 64. But we still need to verify this resolution. Because that the 64 × 64 resolution might be something else in practice. To ensure the resolution, we have to have every sensor cell work in the first place. The experiment shows that every sensor can work normally. Since the three layers of the sensor mat are of a whole piece, the sensor itself couldn’t have any physical problems. However, some sensor cells may not work for other reasons. Figure 28 shows the pressure image of a fingertip. We notice that two lines are missing. No matter how hard you press, nothing happens on these two lines. So, we can say the sensor cells on these two lines don’t work. The real resolution drops. This problem is caused by the bad connection between the sensor and the control board, which makes two rows of the sensor are not connected. Problems like this could be solved easily by having a better connection.

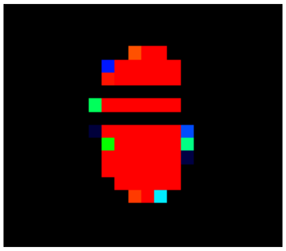

Figure. 28. An incomplete pressure image of a fingertip

Next, we don't only want all sensor cells to work, but work in a proper way. When we press different cells on the sensor, the pressure image should react in different areas. If pressing cell A and cell B has the same reaction, the real resolution should be lower. 4096 cells should correspond to 4096 areas on the pressure image. The experiment shows that every sensor cell can work separately. Thus, the 64×64 resolution is correct.

Compare with the current E-cone, our resolution is 50% higher.

B. *Influence of nearby sensor cells*

In section A, we know that pressing one cell won't show the image of another cell. But, when we press two or more cells on the sensor mat, will them have an influence on each other? The answer is yes. This problem is already mentioned in Chapter IV when we test the 2×2 sensor array. The phenomenon observed here is that when we press two areas that are horizontally aligned, they will have an influence on each other. Pressing one of the two areas harder will make the pressure values of the other area drop. Pressure image of two fingertips in figure 29 shows the phenomenon.

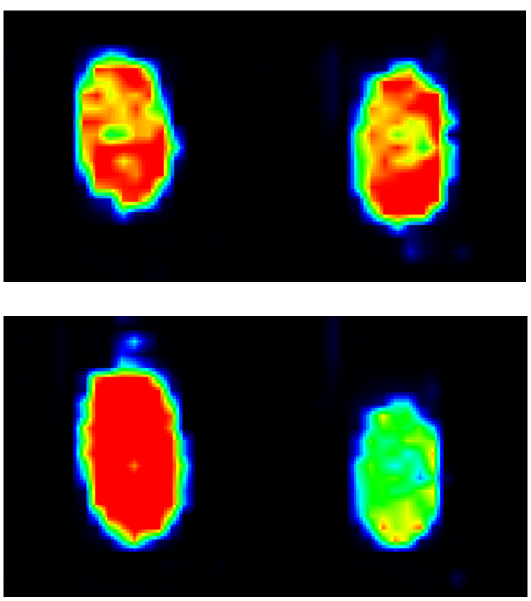

Figure. 29. Influence of horizontally aligned areas.
The upper image is the original image and bottom image is the resulting image. When we start to press the left area harder and keep the force on the right area unchanged. The right area starts to fade with the increasing force applied on the left area.

On the contrary, no influence exists when we press two areas that are vertically aligned. This phenomenon is caused by the low resistance of the resistive material in our sensor and this is inevitable in our case. Figure 30, with two columns and two rows, shows the basic theory behind this. Row1 is connected to ground and Row2 is connected to VCC. Col1 is kept open and Col2 is connected to virtual ground. Like what's shown in the figure, sensor cell A is selected. As what has been discussed in the Chapter III, most of the current should flow following the red arrow. But if we start to press an area, for example cell B and C, that is horizontally aligned with cell A, (Cell B and C are often pressed at the same time because they are close to each other), the resistance of these two cells will decrease gradually. The inner material we use in our sensor can reach a minimum resistance of 30ohm which is too small. When the resistance of cell B becomes small enough, the electrical potential of electrode Col1 will be close to VCC. An electrical potential difference forms at cell C. Therefore, leakage current will flow following the green arrow. The input current of the amplifier decreases, making it seem like the pressure on the cell A drops. But the truth is the real pressure stays the same, what decreases is the pressure value we measured. However, pressing a vertically aligned area, for example Cell C and D, won't form any electrical potential difference at these two cells. That's why the problem only happens between areas that are horizontally aligned. To get rid of this problem, we have to look for another kind of material that has higher resistance. In this way, the electrical potential of electrode Col1 wouldn't increase too much. Although this phenomenon is not what we want, normally we can still get a nice pressure image. Because it only happens when the pressure applied on the two areas have a big difference.

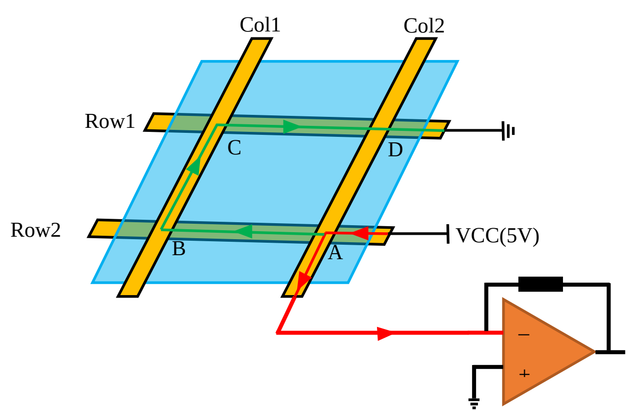

Figure. 30. Current split on horizontally aligned areas.

The two cells shown in figure 30 are adjacent to each other. That doesn't mean only the adjacent cells have an influence on each other. Figure 29 already shows that no matter how far the distance between the two areas, as long as they are horizontally aligned they will influence each other. That's because the leakage current is following the electrode strips on the bottom layer. The electrode strips are conductors with very low resistance, which makes the distance have no effect on the result.

C. *Sensor-Pressure relationship*

The ADC we use has 16-bit output, which ranges from 0 to 65535. Assuming the resistance of the conductive material can change linearly from the minimum value to the maximum value. Then we should get the corresponding pressure linearly changing from the maximum to the minimum. Let:

$$minimum\ pressure = P_{min}$$

$$minimum\ pressure = P_{max}$$

Then the pressure resolution should be:

$$\text{Pressure}\ resolution = \frac{P_{max} - P_{min}}{65535} \tag{2}$$

However, equation (2) is based on the assumption that the resistance can change continuously and it won’t influence the functionality of other electronics. We are not sure about the property of the resistive material yet. Thus, we need to do some test to have it figured out.

1)Resistance range

By using the multimeter to measure the resistance of the sensor cells, we got a general but not strict resistance range. The resistance of a single sensor cell can change from tens of ohms to thousands of ohms. Because the resistance of the material is not distributed strictly even, some cells could have a little bit higher or lower limit.

2)Pressure range

The pressure range of the sensor is not a fixed value. There are several factors can affect the result. The first one is the test voltage V_{test}. According to equation (1), when we have a smaller V_{test}, the R_k should be bigger to get the same amount of output of the amplifier. That means less pressure needed to apply on the sensor because the R_k increases with the decrease of the pressure. Since the range of the resistance won’t change, the pressure range is extended. The second factor is the size of the area on which the force is applied. Besides the reason that a bigger area will reduce the force applied to each cell. According to what has been discussed in section B, one single cell can bear more pressure in a bigger area with more cells that are horizontally aligned because of current leakage.

We designed an experiment to test the properties of the sensor mat. We use numbers of weights as pressure generator. By putting the weight (referred to as G kg) on a specific area (referred to as S m^2) of the sensor, we get the applied pressure as:

$$Applied\ pressure = \frac{G}{S}\ \ \text{kg/m}^2 \tag{3}$$

we can easily evaluate the behavior of the sensor by increasing or decreasing the number of weights. In our experiment, we have 24 brass weights in total, each weight is about 78 grams with little bias. By having S = 1cm^2, we got the experiment result as shown in figure 31.

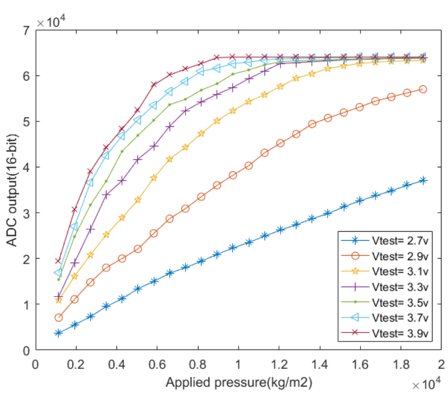

Figure. 31. The relationship between sensor readout and applied pressure for different test voltage. S = 1cm^2, the ADC readout is the average value of all the cells in this area.

10861	9216	63701	63983	63885	35777
4760	7079	44768	64053	64023	35408
1451	3123	12883	39942	10381	1983
639	1049	5675	16815	2063	127
55	75	879	3477	4919	153
99	199	871	2335	1859	187

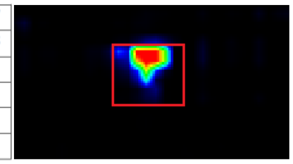

Figure. 32. The unevenly distributed pressure in a 1cm^2 area. The middle part has much higher value than the surroundings. The left side shows the output value of all sensor cells. The right side shows the pressure image after interpolation. The red rectangle is the area where we measure.

For a small V_{test} = 2.7v, the curve is almost linear. But for a bigger test voltage, the slope of the curve decreases with the increase of the pressure. This is caused by the uneven force distribution on the 1cm^2 area. The pressure in the middle of the area is always higher than the surroundings. Like what’s shown in figure 32. After the value of the middle cells reaching the limit, the increasing speed is slowed down. This also explains why the relationship for V_{test} = 2.7v is linear because the pressure in the middle still doesn’t reach the maximum value with all the 24 weights. We can expect a slope decrease for V_{test} = 2.7v with more weights added. Derived from the linear curve for V_{test} = 2.7v, we could say that for a single sensor, the readout has linear change with the applied pressure as long as the pressure is below the maximum.

In figure 31, with the increase of test voltage, the slope of the curves increases and the curves seem to be closer to each other. The increasing slope means with a higher test voltage the sensor is more sensitive. The curves becoming closer means that when the test voltage cross over a certain line, adding a certain amount of force will have a smaller influence on the result than that with a small voltage. Figure 33 illustrates this phenomenon very well. We can see the slope of the curve stays the same at first and decreases with the increasing test voltage after a certain point. The reason for this is what has been discussed in the fourth findings in section A, Chapter IV. For a big test voltage, the output current of the amplifier is over the limit in our circuit, making the test voltage drops to a lower value. We didn’t use test voltage below 2.7V, that’s because a small voltage lower than 2.7v couldn’t generate much input current for the amplifier. Even a huge amount pressure is applied to the sensor, the ADC readout cannot reach the maximum.

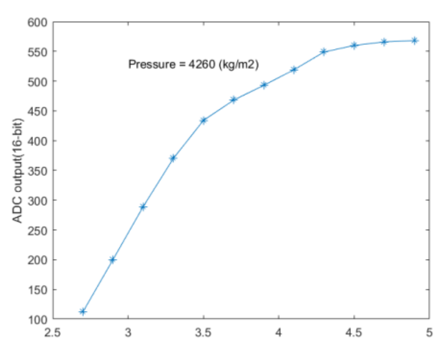

Figure. 33. The relationship between sensor readout and test voltage.

The second factor influence the pressure range is the size of the area. We did another experiment with $S = 2cm^2$ to see the real influence. Figure 34 shows the result. For a certain pressure value, the sensor readout for $S = 2cm^2$ is not equal to the half of the readout for $S = 1cm^2$ but close to it. It means the influence of leakage current is not much. In section B we mentioned that the leakage current will have a big influence only when the pressure applied on different cells have a big difference. In our case, we apply pressure to one area, sensor cells are close to each other, thus there won't be a very big difference. The result is totally acceptable.

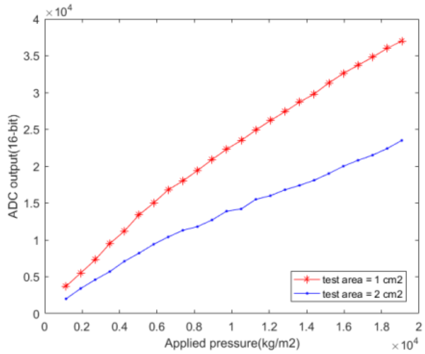

Figure.34. The relationship between sensor readout and applied pressure for different size of areas with force applied.

In a small summary, the sensor output is linear with the applied pressure for a single sensor cell. We cannot give out a specific range of the pressure that we can apply. Because it depends on the test voltage we use and how much area you apply your force on. From the experiment we did, we can see the approximate maximum pressure for $V_{test} = 3.9v$, $S = 1cm^2$ is 9000 kg/m^2. And for $V_{test} = 2.7v$, $S = 1cm^2$, the maximum pressure is over 20000 kg/m^2. Of course, we could calculate the maximum pressure value for every test voltage, but we need more weights and it will take time.

D. Frame rate

In the beginning, while we were still using the first scanning method, the frame rate is only 1.5 fps. The second scanning method doubled the frame rate to 3 fps. A frame rate of 3 fps is merely acceptable. The lag isn't much but we can still feel it. However, a frame rate of 1.5 fps is totally unacceptable. The lag is too much, making the system not real-time anymore. By using the third scanning method, the frame rate of the pressure image we can get now is 15fps. It's about 30% higher than the current E-cone which a frame rate of 10 fps.

However, if we take the interpolation calculation into account, the frame rate drops to 5.5fps for the bilinear interpolation and 2fps for bicubic interpolation. The previous E-cone generates the image at a speed of 10fps and also displays the image in the same speed of 10fps even it extends the resolution from 44 × 44 to 704 × 704. Because it does the interpolation calculations on the PC using the graphic card. But in our new design, all the calculations are done on the Raspberry Pi. The PC is only for display.

Probably in the future, the frame rate can be optimized further if we make a better scanning method or have a more efficient algorithm.

VII. Further Work

So far, we have built our own pressure sensor which can work in a reasonable way; we have redesigned the electronics to match the new sensor; we have implemented the wireless data transmission and implemented the web-based display; Also, we have done a lot of experiments to analyze the property of the hardware and software.

We already have a complete working system, but it's only a prototype that is used to do the test. It's not a finished product yet. To get to the final step, we have to integrate everything into the cone. To do that, we have to build a cone-shaped carrier that has enough inside space for electronics. We have to redesign the shape of the electronics board to match the carrier. We have to find a way to attach our sensor to the surface of the carrier. Although there are some things left to be done, the most important part which is building a working system with new sensor has already been finished. What left is only assembling.

As what has been discussed in section VI, the low resistance of the conductive material limits the performance of our sensor. In the future, we have to look for another kind of material with higher resistance. The material should still have a relatively high resistance when high pressure is applied to it. Therefore, the current flowing through one cell wouldn't be split by other cells.

VIII. Conclusion

E-cone is a medical device that is used to capture grab pattern of the patients with hand disease. It was developed several years ago at the University of Twente. Although the current E-cone is competent, there are disadvantages exist and it can be optimized in many aspects. For example, the cost could be reduced and it could be made easier to use. In this report, we discussed all the work we have done for redesigning the current device. In the end, a working system was set up and it worked as we expected. The final product was not produced due to the lack of time. It could be finished in the future.

We use the pressure sensor mat to get the pressure image of one's hand. In order to get a higher resolution and reduce the cost, we built our own pressure sensor, which has a resolution of 64 × 64. Compared with the 44 × 44 resolution of the current sensor, we have improved the resolution by about 50%. With the newly designed electronics, the frame rate of the pressure sensor can reach 15 fps, which is about 30 % higher than the current one. With proper set-up, the sensor can bear a pressure over 20000 kg/m^2. Our work points out a way of making pressure sensor by ourselves. It is totally feasible to make sensors using much cheaper material and make it to a higher resolution. In the future work that requires measuring pressure, we could use the method in this report.

We used WIFI as the communication method between the E-cone and the displaying terminal instead of using the USB wire. Thus, the E-cone became totally independent. Patients can hold the device in all kinds of ways without limitation. We established a web server and used a web page to display the pressure image we collected, instead of using a specialized software. It makes the device more convenient to use. Any device that has a web browser can be used as the displaying terminal.

The whole system can work very well. Pressure can be sampled by the sensor and the image can be generated at a fast speed. The feedback pressure image can be shown on the website in real-time. Although we have completed the core part of the design and gotten a working system, the final product is waiting to be finished. Of course, better result can be achieved with further optimizations.

REFERENCES

[1] Cross M, Smith E, Hoy D et al. The global burden of hip and knee osteoarthritis: estimates from the global burden of disease 2010 study. Ann Rheum Dis 2014; 73:1323–30.

[2] Scott DL, Wolfe F, Huizinga TW (Sep 25, 2010). "Rheumatoid arthritis". *Lancet*. **376** (9746): 1094–108.

[3] Carly Brown, Sasha Cai Lesher Perez, Lee Linstroth, and Nathan Kleinhans. Hand rehabilitation device for patients during acute phase of stroke. University of Wisconsin-Madison. Dec 19, 2007.

[4] L. Spreeuwers, A. Aarden, A. Hoeksma, L. Roorda, P. Kroon, and D. Spreeuwers. Ontwikkeling van een exercise cone met druksensoren voor revalidatie en behoud van werkvermogen van reumapatiÃnnten. verslag van de haalbaarheidsstudie, 2008.

[5] R. Veldhuis, A. Bazen, J. Kauffman, P. Hartel. Biometric verification based on grip-pattern recognition. In Security, Steganography, and Watermarking of Multimedia Contents, pages 634-641, 2004

[6] L. Wang, T. Ding, P. Wang. Thin Flexible Pressure Sensor Array Based on Carbon Black/Silicon Rubber Nanocomposite. IEEE Sensors Journal, VOL. 9, No. 9, 1130-1135, September 2009.

[7] S. V. Gawande, P. R. Deshmukh. Raspberry Pi Technology. International Journal of Advanced Research in Computer Science and Software Enginering.VOL. 5, Issue 4, 37-40, April 2015.

[8] J. A. Kauffman, A. M. Bazen, S. H. Gerez and R. N. J. Veldhuis, Grip-pattern recognition for smart guns, 14th Annual Workshop on Circuits, Systems and Signal Processing (ProRISC), Veldhoven, The Netherlans, 2003, pp. 379-384.

[9] William L. Maness et al. Pressure and contact sensor for measuring dental occlusion. United States Patent; 4,856,993, Aug. 1989. Tekscan, Inc.