ECE 445 Design Document

Spring 2019

"Learning and Labor"

FOAM PRESSURE-SENSOR BASED CONTROL METHOD FOR CONTROLLING PROSTHETIC HANDS

YANGGE LI

ZHOUSHI ZHU

ENLIANG LI

TEAM 19

TA: Amr Martini

Supervisor: Prof. Michael Oelze

3/7/19

Contents

1. Introduction

1.1 Statement of Purpose

Nowadays, prosthetic hands are commonly controlled by Electromyographic (EMG) method which evaluates the electrical activity produced by skeletal muscles. However, the traditional EMG method is not accurate enough, because the measurements of the electrical signal suffer from high level noises come from the users' skin. In addition, due to the physical layout and high cost of EMG sensors, the number of sensors is insufficient to acquire enough data to track the muscle movements precisely.

In this senior design project, we are collaborating with the PYONICS Inc., a customized prostheses manufacturing company, to design and prototype an interface platform to control a prosthetic hand based on foam pressure-sensor as an alternative of the controlling system based on EMG currently used by the PYONICS Inc. The project includes:

a. The design of the senor module which carries the electrodes array with its corresponding communication peripherals and the communication master device which processes data from sensor modules.

b. The programing of the communication protocol.

c. The soldering and assembly of all the components.

This project is sponsored by the PSYONIC Inc. The pressure sensor method is more accurate, less noisy and cheaper, and preliminary research[1] shows promising result regarding this pressure-sensing method.

1.2 Existing Work & Objective

The aforementioned paper introduces a methodology of controlling prosthetic hand based on pressure sensors located around testers arm. The researchers designed a tactile bracelet composed of 10 sensor boards which was deployed around the test subject's residual limb or forearm, in the case of able-bodied or disabled subjects. The pressure sensors are built with electric conducting foam and electrodes, utilizing the foam's property of changing resistance while compressed by pressure.

The experiment consisted of two phases: training phase and testing phase. During the training phase, the test subjects were asked to try to make the intended hand movement for a period of time while wearing the bracelet. Meanwhile, the sensors on the bracelet collected the pressure distributions caused by muscle movement on subjects' arm to train the classification algorithm which classify the pressure distribution into categories according to different hand movement. During the testing phase, the test subjects were asked to repeat the movements in training phase with the bracelet on and the pressure distribution collected were classified by the pre-trained classification algorithm and transferred. The classification results were compared with the ground truth hand movements used in the training phase and an average inference accuracy was calculate for both disabled subjects and able-bodied subjects. The researchers claimed that the average accuracy is 89.15% for able-bodied and 93.07% for disabled subjects.

The PSYONIC Inc. has a finished product of prosthetic hand based on the EMG method, thus we will integrate the prosthetic hand provided by the company with our prototype. The PSYONIC Inc. will also provide us technical support and funding for extra PCB orders separated from the course timeline.

We are planning to deploy the similar methodology introduced in the paper to build an interface platform which interacts with the electric conducting foam and prosthetic developed by PSYONIC Inc. The previous research only focused on classification results but we are moving one step forward to utilize the classification results to generate according command to control the prosthetic hand.

The main goal for designing and prototyping such a platform is to prove the efficacy of a new methodology and by reconducting similar experiment mention above, we are expecting to reproduce similar inference accuracy. Therefore, the classification algorithm that we are planning to develop is only expected to classify the pressure distribution into limited groups to achieve predefined simple hand motion control. PSYONIC Inc will potentially apply this new methodology in their products in the future depending the performance of the prototype and they will potentially implement more complex classifiers such neural networks to allow more sophisticated hand motion control. Some design details such as physical dimensions and foam choices are highly customized to fit a specific test subject's and may also need to be reconsidered in future applications on other individuals by PSYONICS Inc.

1.3 High-level requirements

There are three major requirements for our design:

- Our system should be able to sense and process one specific user's muscle movement and convert it to the command for prosthetic.
- The user should be able to control the prosthetic hand to complete motions including turning hand in four directions and clenching and releasing of the fist.
- The pressure sensor we design should be able to sense the muscle movements that compress the sensor by more than 0.1mm.

2. Design

2.1 Block Diagram

The system we designed will consist of a master device and some slave devices. The slave devices will hold the pressure sensors, collect pressure data from user and transmit them to the master device. The master device will collect data from slave devices and communicate with PSYONIC's prosthetic hand.

2.1.1 System Overview

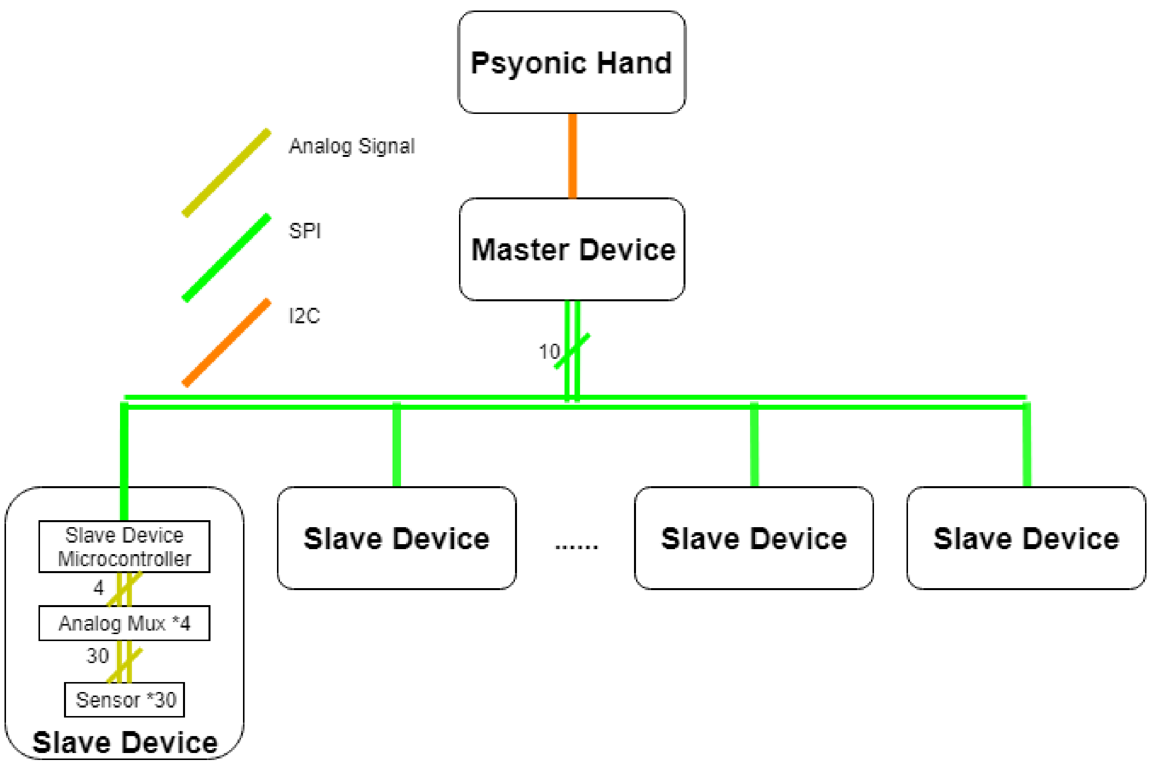


Figure 1: System Overview Block Diagram

2.1.2 Master Device

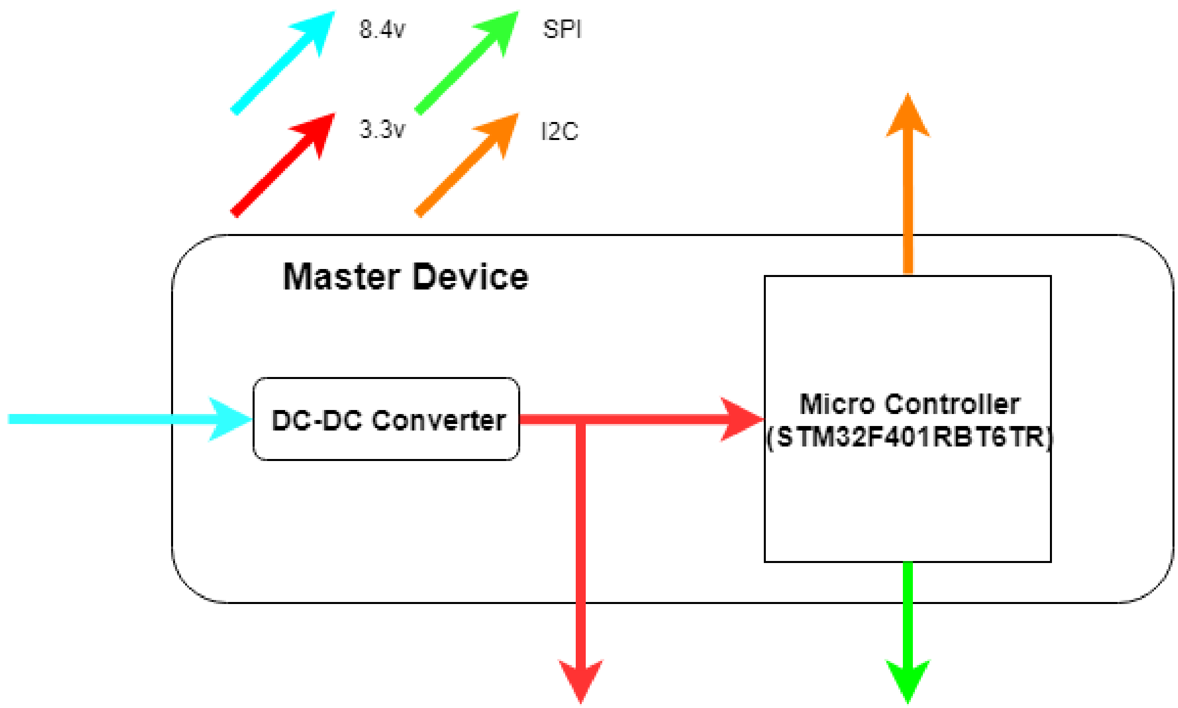


Figure 2: Master Device Block Diagram

2.1.3 Slave Device

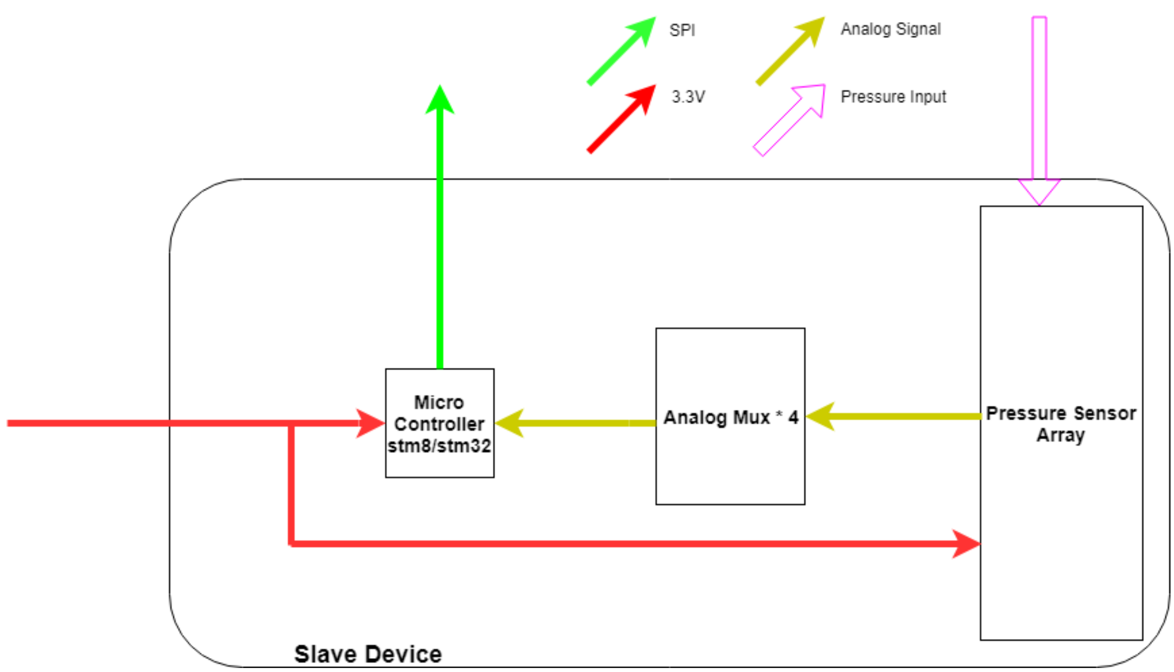


Figure 3: System Overview Block Diagram

2.2 Physical Design

The final prototype will be a bracelet which holds 10 pressure sensor modules and the master device as shown in figure 11. The bracelet should fit testers' arm thickness so that the pressure sensor can work in the optimal range. It should also provide enough tension and enable adjusting of the position of the sensor modules for testing purpose. Potential choices for the material of the bracelet include hook-and-loop fasteners or 3D printed plastic band similar to watch band.

Each sensor module will be contained in a 3D printed rectangular case. One side of

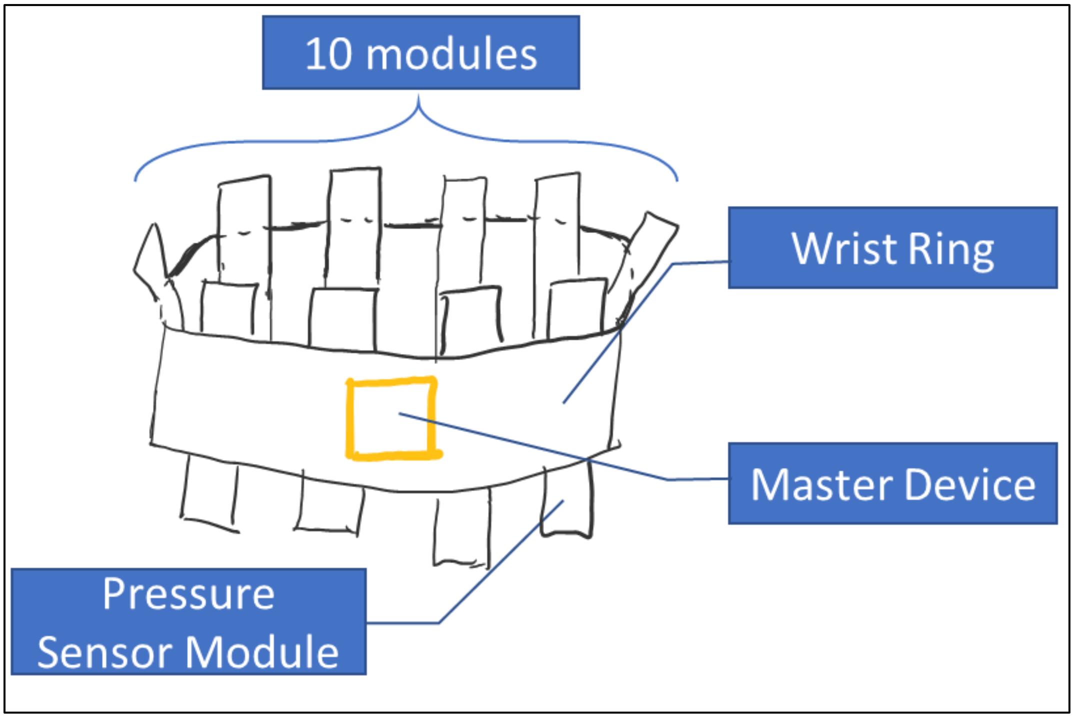


Figure 4: Physical Design Overview

the case will be attached to the bracelet via hook-and-loop fasteners or glue and the opposite side of the case will be left open allowing electric conducting foam to touch human skin. The master device should be attached to the outer surface the bracelet, i.e., the side opposite to the sensor modules, to avoid intervening the pressure sensing.

The dimension of the bracelet, including shape and circumference, and the sensor cases may be subject to change depending on the testers arm circumference. The size of the sensor module cases should be slightly larger than the sensor module PCB so the PCB can be contained and hold steadily. The length of the sensor module case can be extended for routing purpose, but the width of the case should be less than 3 cm, which is the maximum width of the PCB. This restriction is to ensure that 10 sensor modules can be placed around the tester's arm. The width of the bracelet should larger than the length of pressure sensor module PCB to provide uniform tension to the sensors.

2.3 Block Description

2.3.1 Master Device Microcontroller (STM32F401RB)

Input: 1.7V-3.6V power input.
Output: Command for PSYONIC's hand.
Communication: I2C communication with PSYONIC's hand; SPI communication with slave devices.
Description:
The STM32F401RB is using an ARM 32-bit Cortex-M4 CPU with floating-point unit (FPU). It has frequency up to 84 MHz, and contains up to 256 Kbytes of Flash memory, 512 bytes of OTP memory and up to 64 Kbytes of SRAM. [2]

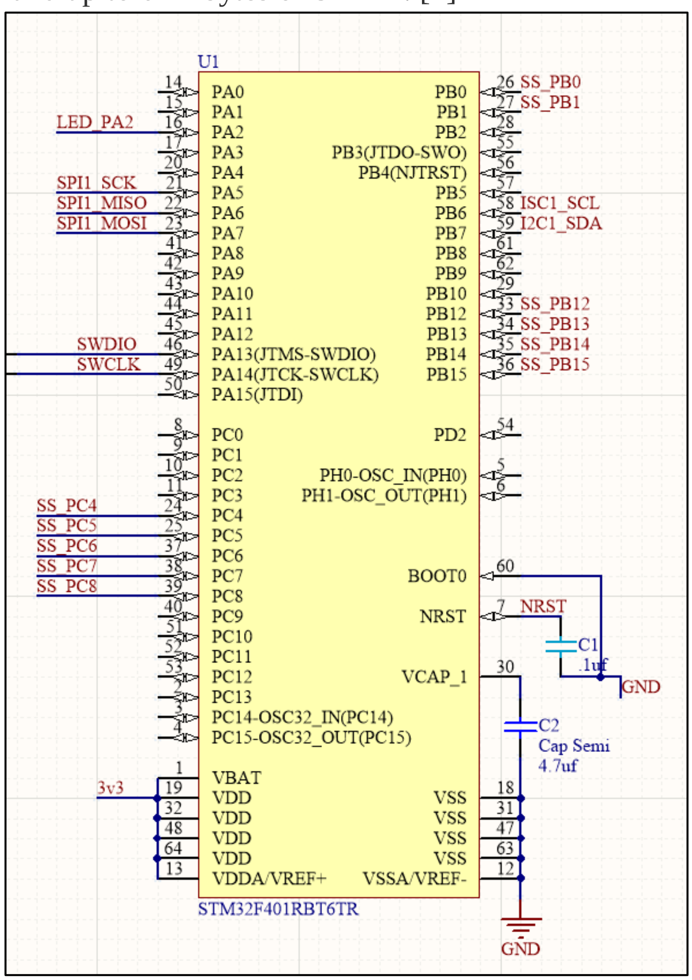


Figure 5: STM32F401RBT6 Schematic

This unit is supposed to collect data from all slave devices through SPI interface (MISO, MOSI, SCK, SS). It will then process the collected data and convert the data to command towards the PSYONIC's hand. The command will be transmitted through I2C interface (SDA and SCL) to the PSYONIC's hand.

This unit will be powered by a 3.3V power input regulated from an 8.4V voltage supplied from the PSYONIC's hand. A pin layout summary of these interfaces is shown in figure 5.

Pin	Function	Connection
1	External power supply for RTC, external clock oscillator and backup register	Connected to DC-DC Converter output
19, 32, 48, 64	External power supply for I/Os and the internal regulator	Connected to DC-DC Converter output
18, 31, 47, 63	Ground reference voltage	Connected to general ground
13	Positive reference voltage for ADC	Connected to VDD pins
12	Negative reference voltage for ADC	Connected to VSS pins
24, 25, 26, 27, 33, 34, 35, 36, 37, 38, 39	Slave selects	Connected to SS on slave microcontroller
21	SPI clock	Connected to clock pins on slave microcontroller
22	SPI master-in-slave-out bus	Connected to MISO pins on slave microcontroller
23	SPI master-out-slave-in bus	Connected to MOSI pins on slave microcontroller
30	Stabilization for the main regulator	Connected to general ground via 4.7 uF capacitor
46, 49	Programming pins for microcontroller; serial wire debug I/O and clock	Connected to PC for programming
58, 59	Communication pins with prosthetic hand control unit via I2C protocol	Connected to microcontroller on prosthetic hand
60	Bootloader selection pin	Connected to ground
7	External reset pin	Connected to ground via 1uF capacitor
16	Output pin for testing	Connected to a test LED

2.3.2 Slave Device Microcontroller (STM32F030K6)

Input: 2.4V-3.6V power input; Analog signals from MUXes.
Output: Digitized pressure sensor reading.
Communication: SPI communication with master device.
Description:
The STM32F401RB is using an ARM 32-bit Cortex-M0 CPU. It has frequency up to 48 MHz. It contains up to 256 Kbytes of Flash memory and up to 32 Kbytes of SRAM. In addition, the microcontroller also has one 12-bit ADC with up to 16 channels and conversion range 0 to 3.6 V. [3]

This unit is supposed to read analog voltage input from the MUXes. It will digitize the input using the built in 12-bit ADC of the microcontroller. The result data will be transmitted to the master device using SPI interface (MISO, MOSI, SCK, SS). In addition, four 8-to-1 analog MUXes are used to choose between different sensors' signal supporting up to 32 pressure sensors on each slave device. The microcontroller will also have three pins connected to all the MUXes to select between signals.

This unit will be powered by the 3.3 V input from the DC-DC Converter on the master device. A pin layout summary of these interfaces is shown in figure 6.

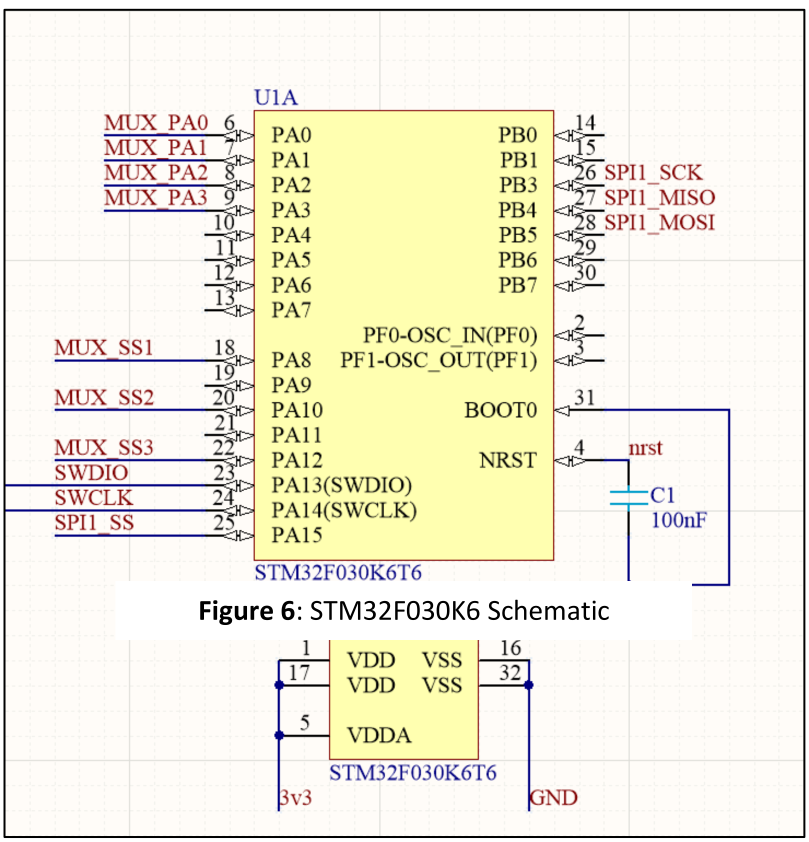


Figure 6: STM32F030K6 Schematic

Pin	Function	Connection
1, 17	External power supply for I/Os and the internal regulator	Connected to DC-DC Converter output
16, 32	Ground reference voltage	Connected to general ground
5	External analog power supplies for ADC	Connected to VDD pins
25	Slave Select	Connected to slave select pins on master microcontroller
26	SPI clock	Connected to SPI_CLK pin on master microcontroller
27	SPI master-in-slave-out bus	Connected to MISO pins on master microcontroller
28	SPI master-out-slave-in bus	Connected to MOSI pins on master microcontroller
6, 7, 8, 9	ADC pins for readings from sensors	Connected to output pins of multiplexers
18, 20, 22	Select pins for multiplexers	Connected to select pins of multiplexers
23, 24	Programming pins for microcontroller; serial wire debug I/O and clock	Connected to PC for programming
58, 59	Communication pins with prosthetic hand control unit via I2C protocol	Connected to microcontroller on prosthetic hand
31	Bootloader selection pin	Connected to ground
4	External reset pin	Connected to ground via 1uF capacitor

2.3.3 DC-DC Converter (TPS82140SILR)

Input: 3V-17V power supply, typically 8.4V (from PSYONIC's hand in our design)
Output: 0.9V to 6V Adjustable Output Voltage, in our case configured as 3.3V

Description:
The TPS82140 is a step-down converter MicroSiP™ power module optimized for small solution size and high efficiency. It supports input voltage from 3V to 17V and an adjustable voltage output from 0.9V to 6V. The converter support 2A continuous output current. [4]

This unit will take the 8.4V voltage from the PSYONIC's hand convert it to the 3.3V voltage that power the system we designed. It also supports other input power range from 3V to 17V. The 2A output current is enough to power all the rest of the system and a detailed calculation to justify this statement can be found in part 2.6.2 below. A pin layout summary of these interfaces is shown in figure 6.

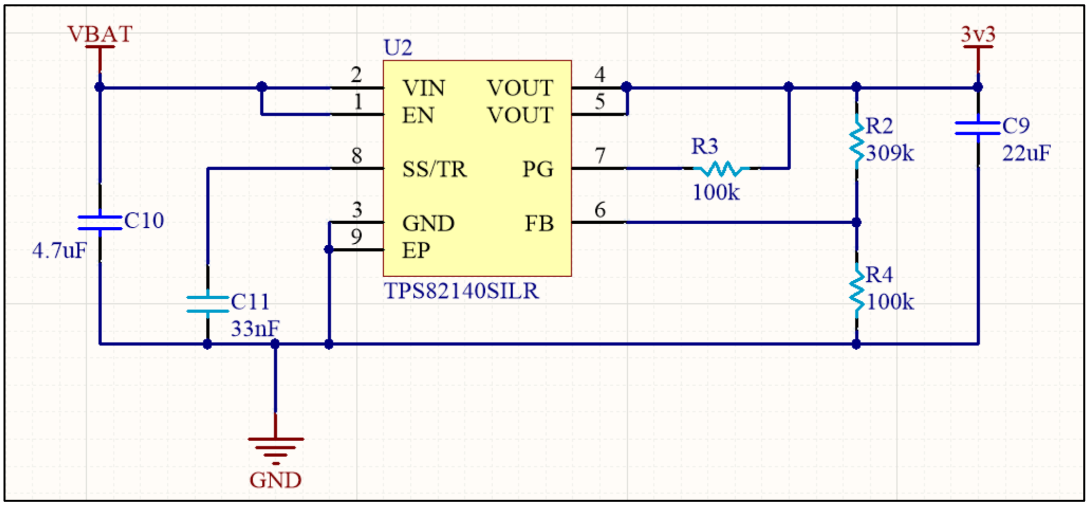


Figure 7: TPS82140SILR Schematic

Pin	Function	Connection
1	Enable pin	Connected to external DC power supply (7.2 - 8.4V)
2	Voltage Input	Connected to external DC power supply (7.2 - 8.4V)
3	Ground reference	Connected to ground
4, 5	Voltage Output	Connected to power supply pins on other microcontrollers
6	Feedback reference	Connected to feedback circuit
7	Power good open drain output	Connected to a pull-up resistor
8	Soft startup and voltage tracking	Connected to an external capacitor to set the internal reference voltage rising time
9	External thermal pad	Connected to ground to achieve appropriate power dissipation and mechanical reliability

2.3.4 Analog MUXs (NX3L4051HR,115)

Input: 1.4V-4.3V input power, typically 3.3V; Analog pressure sensor reading, range between 0-3.3V; Three selection signal (S1, S2, S3).
Output: Selected analog pressure sensor reading to the microcontroller.
Description:
The NX3L4051 is a low-ohmic 8-channel analog switch. It supports a 1.4V to 4.3V supply voltage and has a maximum of $900m\Omega$ on resistor. [5]

Since as we mentioned above, each slave device is going to hold up to 32 pressure sensors, which exceeds the 16 ADC channels we have on the microcontroller. Therefore, four 8-to-1 analog MUX(s) are used to select between different inputs. In case of including 30 pressure sensors on a slave module, three MUX(s) will each handle 8 analog signals and one MUX will handle 6 analog signals. All 4 MUX(s) will share the same selection signal from the microcontroller and connected to different ADC channel on the microcontroller. An example pin layout summary of these interfaces is shown in figure 7.

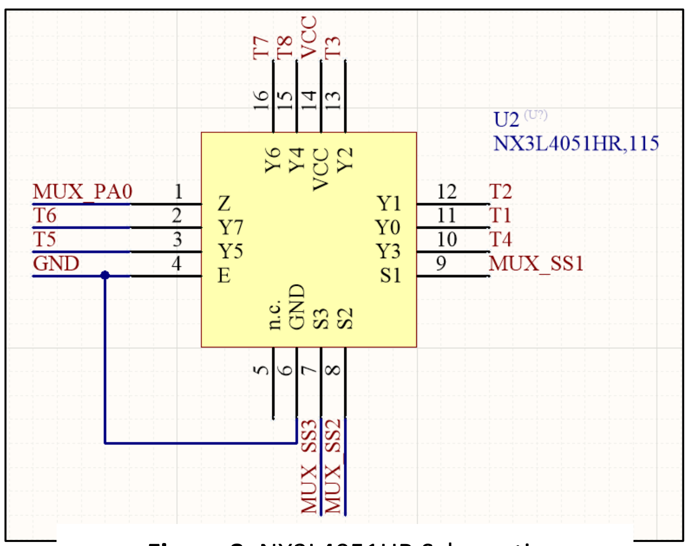


Figure 8: NX3L4051HR Schematic

Pin	Function	Connection
1	Output pin	Connected to MUX out pins on slave microcontroller
2, 3, 10, 11, 12, 13, 15, 16	Input pin	Connected to pressure sensor
4	Ground pin	Connected to ground
14	VCC pin	Connected to VDD on slave microcontroller
7, 8, 9	Select pin	Connected MUX select on slave microcontroller
5	Not used	Not connected

2.3.5 Pressure Sensor

Input: 3.3V input power; Pressure input from human muscle.
Output: Analog voltage change corresponding to change of pressure.
Description:
Each pressure sensor is based on a resistive working principle in which the interface resistivity between two surfaces changes according to the applied load. We will use metal trail on PCB as electrodes and use conductive foam as the sensor material. When load is applied the resistance between the electrodes will be changed and we can use the resistance change to sense pressure change. Therefore, we are going to apply the voltage division principle to convert the resistance change to voltage change, which is demonstrated in figure 8. A detailed explanation about the choice of foam and the resistance of R can be found in section 2.6.1 below.
Figure 9 from [1] illustrates the physical design of the pressure sensor.

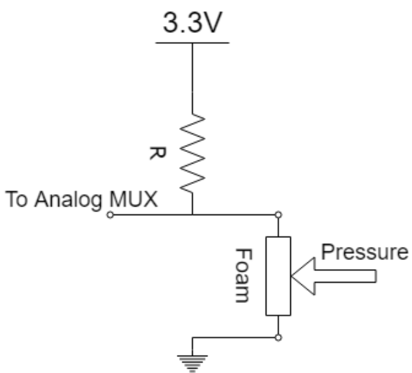


Figure 9: Pressure Sensor Model

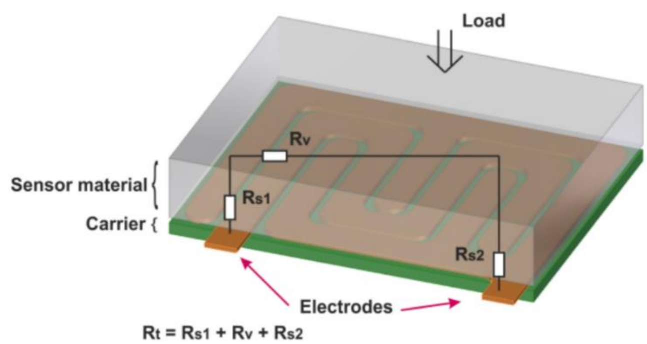
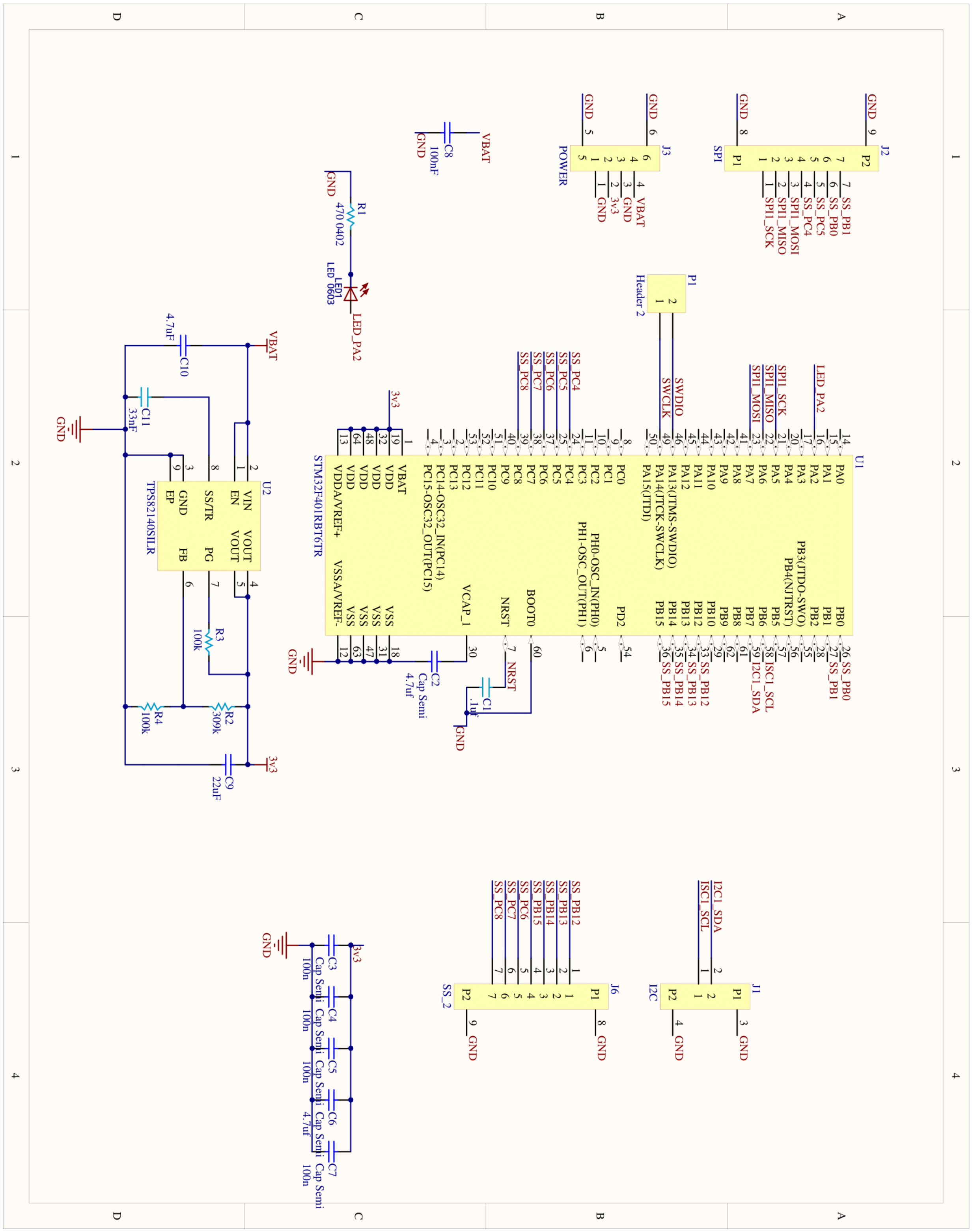


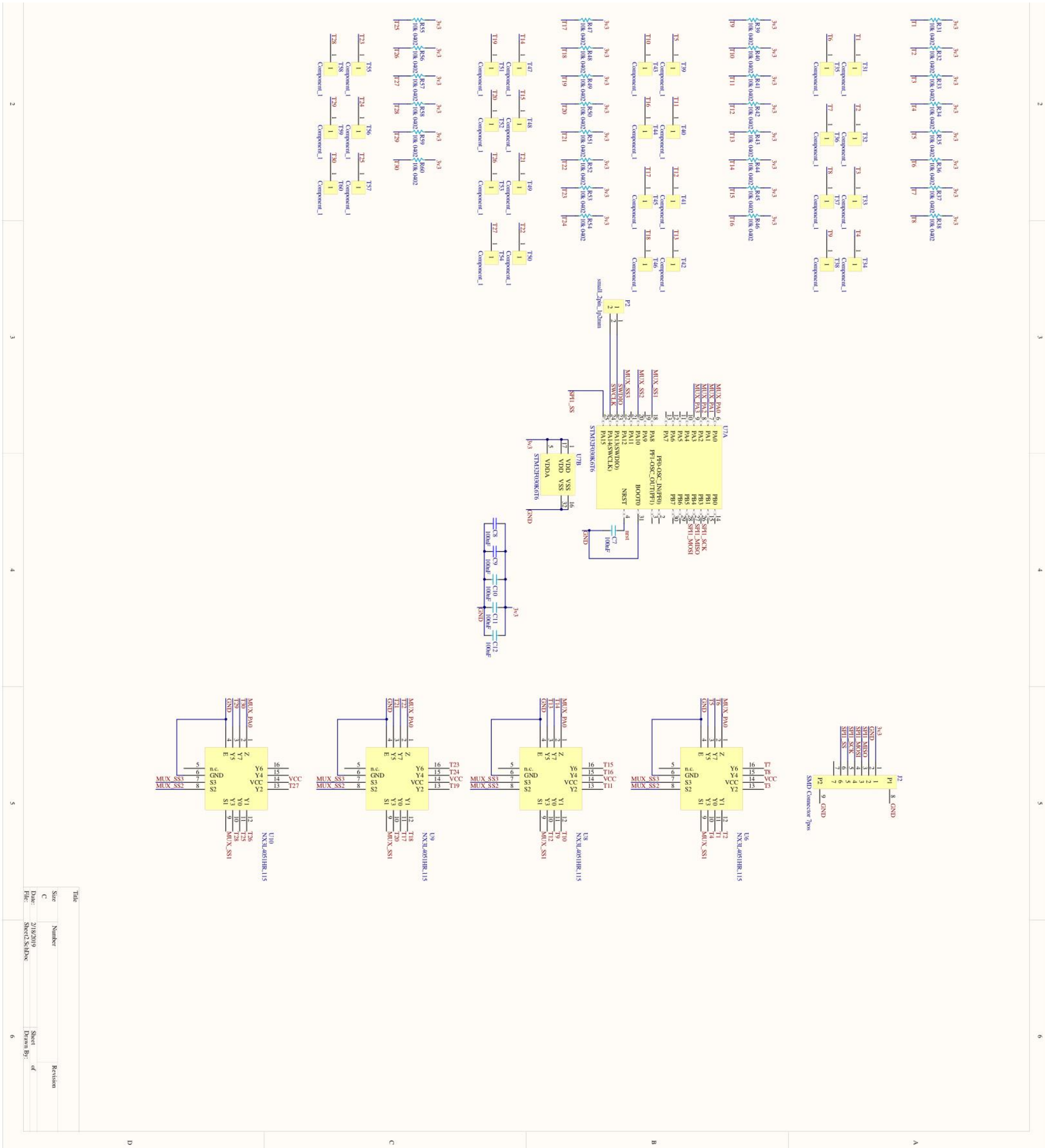
Figure 10: Pressure Sensor Physical Design

2.4 Schematics

2.4.1 Master Device Schematics



2.4.2 Slave Device Schematics



2.5 Software Description

2.5.1 Master Device Software

The microcontroller we are going to use as master in our system is STM32F401RBT6, which is a HAL C platform supporting the strict C89/C90 standard. Thus, we are going to use the "-std=c99 -pedantic-errors" compiling flag in this implementation.

Our system could be viewed as a common multiple Slaves single Master model. And three required control / data flow are attached as below, which will be controlled by our Master Device Software.

In a typical communication cycle, the Master Device Software will do the following,

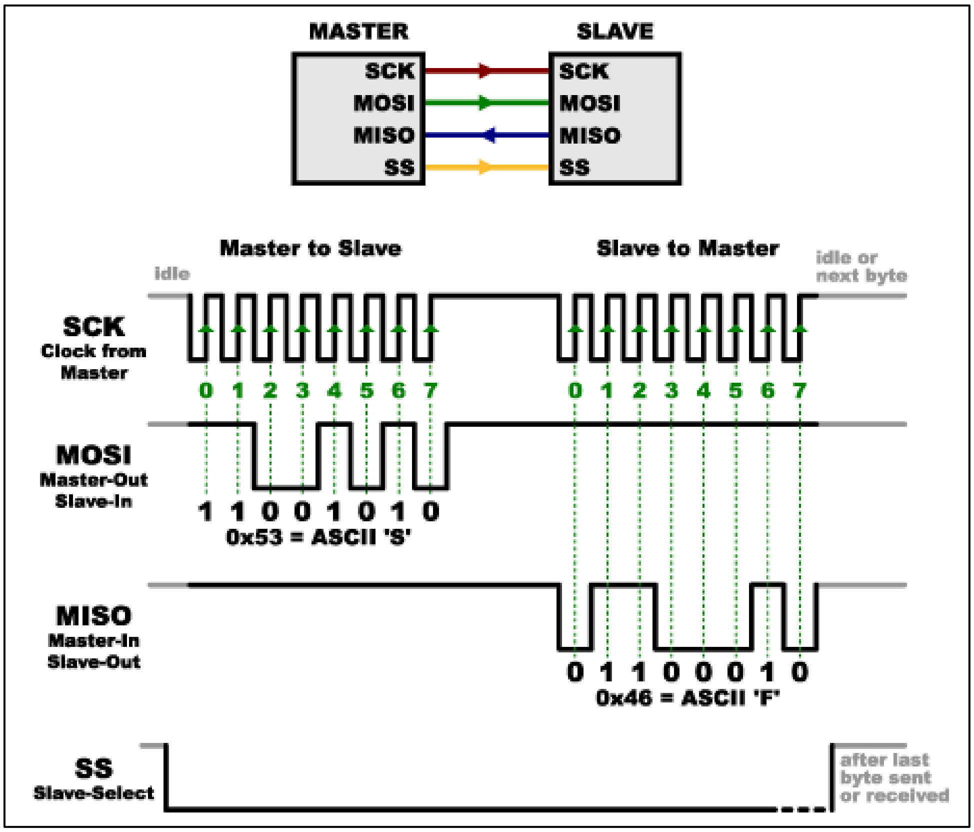


Figure 11: SPI Communication Model

a. Set the SS signal for the slave currently talking with to low (active low) using GPIO pin of the microcontroller, meanwhile, all other SS signals should remain at high to prevent MISO conflict.

b. When there is only one slave is in active, we will then instruct the MOSI to send a user-defined message to the slave indicating the start of the transmission process.

c. As the SCK ticking, master chip will wait for the response from slave by listening to the MISO.

d. SS for the current slave will be now set to high in order to end its permission to write to the MISO.

2.5.2 Slave Device Software

The microcontroller we are going to use as slave in our system is STM32F030K6T6, which is a HAL C platform supporting the strict C89/C90 standard. Thus, we are going to use the "-std=c99 -pedantic-errors" compiling flag in this implementation.

Compared to the Master Device Software, Slave Device Software has more tasks to complete.

The microcontroller on the Sensors Module Chip is expected to complete the pressure sensors signals' Analog-to-Digital Conversion (ADC) as well as the communication with master device.

For the ADC part, our Slave Device Software will do the following,

a. Scan 4 ADC pins at once and register their binary values.

b. Increment the counter to select next group of 4 sensors by giving correct combination of Chip-Select signal to the 8-to-1 MUX(s) we have.

c. Keep doing this until data from all 30 sensors are registered.

The task for communication with the master device is relatively simpler,

a. Waiting for the MOSI signal with the corresponding user-defined message indicating the start of transmission process.

b. Execute the scanning process and preserve only the latest data awaiting to be sent.

c. Write data to the MISO if its own SS is in low (active low).

The timing will be the most significant consideration when the time turns to the integration of the Master Device Software and Slave Device Software.

2.6 Calculation & Measurement

2.6.1 Pressure Sensor Characteristics

The design of pressure sensor is one of the most important part of our project. As we mentioned in section 1.2.5 above, our pressure sensor is based on a voltage division principle and using a conductive foam as the sensor material. In this section, we will explore the characteristic of the conductive foam and the value of the voltage divider resistor R. The final choice of foam is not part of our project that we will integrate our platform with the foam provide by PSYONIC Inc. The following experiments are just conducted to assist designing the sensor module PCB and PSYONIC Inc. will conduct further experiment to choose the best working foam.

2.6.1.1 Characteristic of the conductive foam

To choose the best type of foam, we acquire six different conductive foam and conduct some tests to choose the most suitable among them. We build a voltage divider as shown in figure 12 with supply voltage 8.06v and resistor R=1.492 kΩ and conducted the following three tests.

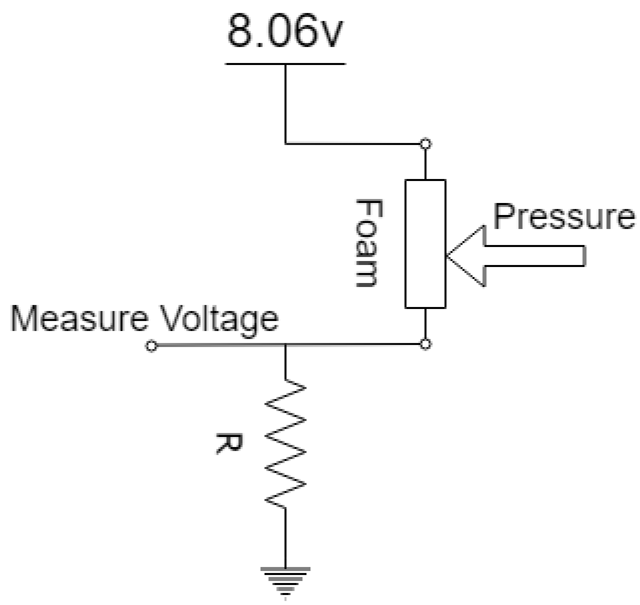


Figure 12: Pressure Sensor Testing Setup

a. Test 1 - establishes baseline, and test if voltage changes depending on how long foam is compressed
   1. Compress foam to 3mm at 3600 mm/min
   2. Hold in compressed state for 55 seconds
   3. Release foam at 3600 mm/min

b. Test 2 - tests if voltage is affected by release speed
   1. Compress foam to 3mm at 25 mm/min
   2. At the moment foam reaches 3mm compressed, release foam at 3600 mm/min

c. Test 3 - tests if voltage is affected by previous position
   1. Compress foam to 3mm at 600 mm/min
   2. Hold at 3mm compression for 5 seconds (establishes baseline of voltage at 3mm compression, in case it changed from test 1 or 2)
   3. Further compress foam to 6mm at 600 mm/min
   4. Release foam back to 3mm compression at 25 mm/min
   5. When foam reaches 3mm compression, release at 3600 mm/min

These three tests mainly test the influence of previous position and speed of compress/release to the resistance of the foam. The results for these three tests should be consistent for each foam. Foam 5 is showing the best result in these tests:

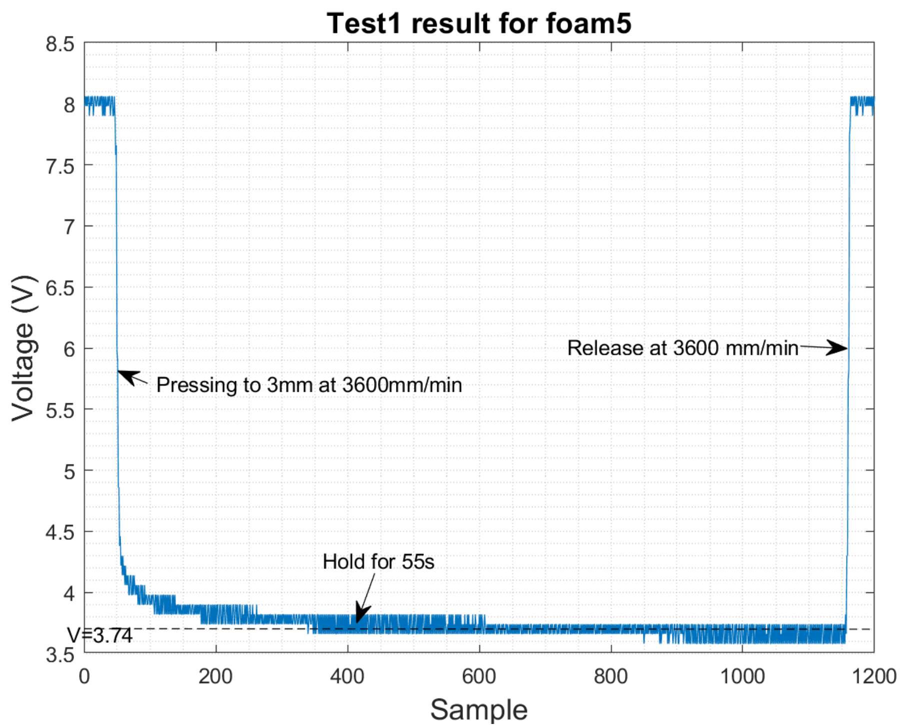


Figure 13: Test1 Result

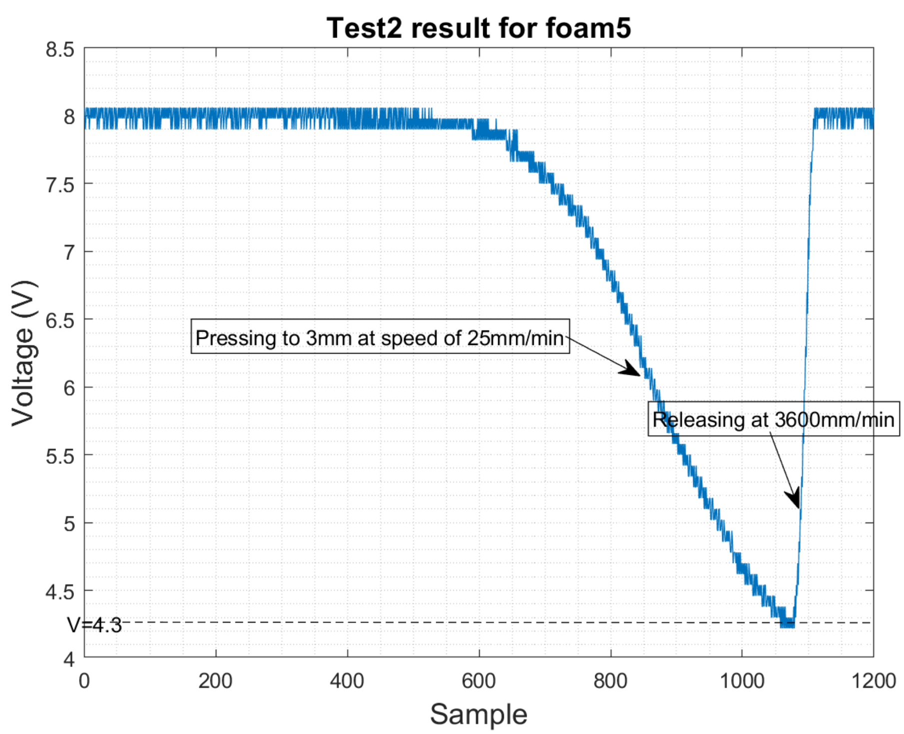


Figure 14: Test2 Result

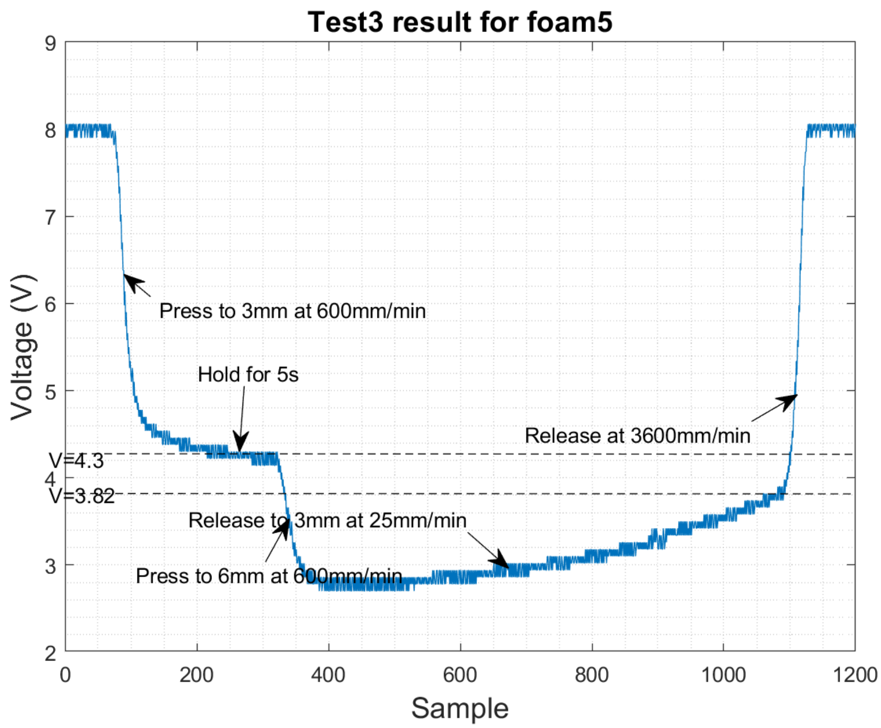


Figure 15: Test3 Result

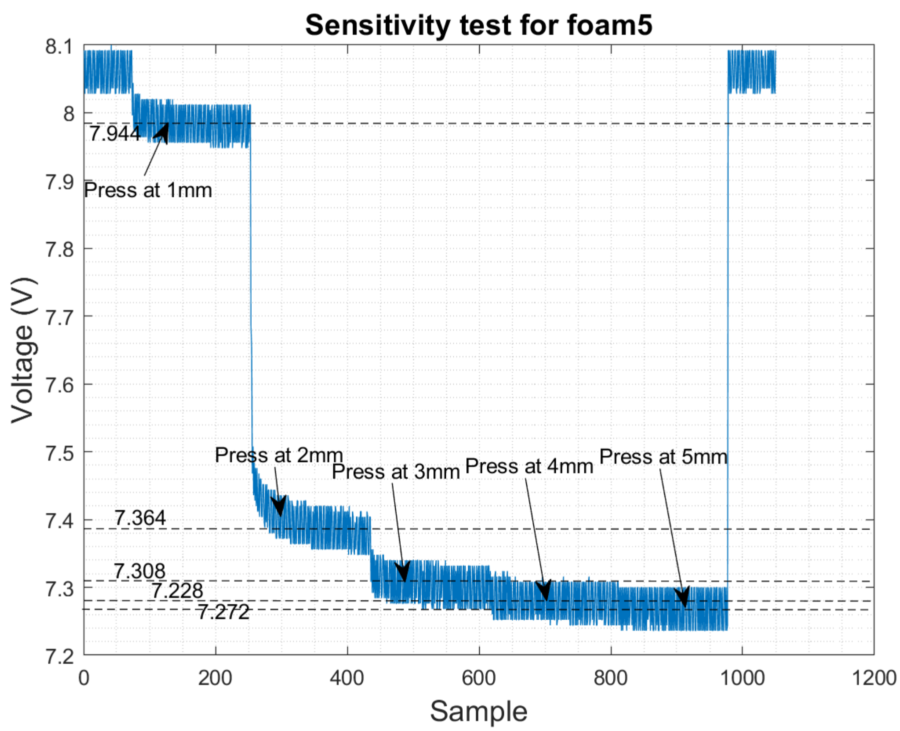


Figure 16: Sensitivity Test Result

The result shows that the voltage at 3mm compression is most consistent between the 3 tests in this foam. As we can see from the results above, the voltage is minimally affected by previous position or speed of compress/release. From the result above, we can also notice that voltage change between 0-3mm compression and 3-6mm compression drops drastically (test 3), showing that 3mm compression is probably the area of optimal compression for resistance change.

Motivated by this fact, we conduct an additional sensitivity test to foam 5. We build a voltage divider as shown in figure 8 with supply voltage 8.06v and resistance of R set to optimal value 25 kΩ (detailed information of this value can be found in section 1.6.1.2).

The foam begins with uncompressed. Then the foam is continuously compressed by 1mm at 90 second intervals (compressed by 1mm, hold for 90 secs, repeat) until reach 5mm. The result of the test is shown below:

As we can see from the result, the test shows that the resistance of foam 5 is most sensitive when the compressed distance is between 0-3mm. Therefore, can be the optimal operation region for this foam. The final choice of foam and the optimal working range of compression will be determined by PSYONIC Inc.

2.6.1.2 Value of the voltage divider resistor R

To determine the value for the voltage divider, tests are performed on foam 5 with a potentiometer.

The inner resistance of the potentiometer is set to $1.492k\Omega$, the resistance value used in previous test, and the foam is connected to the potentiometer while being compressed for $3mm$. After tuning the potentiometer to maximize the voltage across the foam, we discover that the 25 $k\Omega$ is the optimal resistance for R, assuming foam 5 will be the foam deployed on the sensors. The final value of R is subject to change according to the final electric conducting chosen by PSYONIC Inc.

2.6.2 Power Consumption

Since the power consumption for most electrical may vary under different condition, this is only an approximate calculation and we will mostly assume the maximum power consumption for components. In addition, we will assume our system have 10 slave devices.

2.6.2.1 Power Input

The power input of the system is determined by the DC-DC converter (TPS82140SILR) we used. The output voltage for the converter is 3.3V and according to the data sheet [4], the continuous output current is 2A. The total power input for the system is:

$$P_{in} = 3.3 * 2 = 6.6\ W$$

2.6.2.2 Master Device Microcontroller (STM32F401RB)

The input voltage to the STM32F401RB is 3.3V. According to datasheet [2] the current consumption for the microcontroller in run mode at maximum speed (84MHz) with all peripheral on is 22mA, the power consumption is:

$$P_{STM32F401RB} = 3.3 * 0.022 = 0.0726\ W$$

2.6.2.3 Slave Device Microcontroller (STM32F030K6)

The input voltage to the STM32F030K6 is 3.3V. According to datasheet [3] the power consumption for the microcontroller in maximum case when the supply voltage is 3.6V, microcontroller in Run mode at speed max speed 48MHz, code executing from RAM and all peripheral on, the max current is 30.2mA. Therefore, the power consumption for one microcontroller is:

$$P_{STM32F030} \leq 3.6 * 0.0302 = 0.10872\ W$$

The power consumption for all microcontroller (assuming 11 slave devices) is:

$$P_{SDM} \leq P_{STM32F030} * 10 = 1.0872\ W$$

2.6.2.4 Analog MUX (NX3L4051PW,118)

The input voltage to the NX3L4051PW,118 is 3.3V. According to data sheet [5] the maximum supply current for the IC at 3.6V is 5000nA. Therefore, the power consumption for one analog mux is:

$$P_{NX3L4051} \leq 3.6 * 0.000005 = 1.8e - 5\ W$$

The power consumption for all analog mux is:

$$P_{AM} = P_{NX3L4051} * 40 \ \leq \ 7.2e - 4\ W$$

2.6.2.5 Pressure sensor

As we mentioned in section 1.6.1 above, the resistance of the foam varies between 0-3000Ω and the resistance of R is $25k\Omega$. Therefore, explore the case with maximum power consumption and choose the lowest resistance of the foam. Neglect the current consumption by the ADC. Therefore, the power consumption for one pressure sensor is:

$$P_{sensor} = \frac{3.3^2}{25000} = 4.356e - 4\ W$$

The power consumption for all pressure sensors is:

$$P_{total} = P_{sensor} * 30 * 10 = 0.13068\ W$$

2.6.2.6 Conclusion on power consumption

According to the calculation above, the total power consumption of the system is approximately:

$$P_{all} = P_{STM32F401RB} + P_{SDM} + P_{AM} + P_{sensorAl} = 1.173W < P_{in} = 6.6W$$

Therefore, the power input to the system is enough for the system to functional properly.

2.6.3 Communication Rate

In the following section, a calculation about the communication rate between master device and slave devices are presented to show that our system is able to gather enough data to fulfill requirement to control the prosthetic hand.

The total conversion time for the ADC on the slave device microcontroller is $18\mu s$. The microcontroller can run as fast as 84MHz, which is way faster than the ADC conversion. Therefore, the programming running time can be ignored in this calculation. In addition, the latency coming from the analog MUX is less than 100ns, so this latency can also be ignored in this calculation. Therefore, each slave device can acquire all pressure sensor readings in:

$$18 * 30 = 540\mu s = 0.54ms$$

Assume only one slave device is performing ADC operation at a time to simplify calculation and perform worst time timing analysis, it will take

$$0.54 * 10 = 5.4ms$$

for all slave devices to have the pressure data ready.

The ADC we are using have a 12bits resolution. To simplify the design effort, we plan to zero padding the data to 16bits (2 bytes). Therefore, the total amount of data to be transmitted is

$$16 * 30 * 10 = 4800\ bits = 600\ bytes$$

Since SPI protocol between master device and slave devices can typically run at 12Mbits/s, the amount of time for all data to be transmitted is

$$\frac{4800}{12M} = 0.4ms$$

Therefore, the worst possible total amount of time for master to perform a pressure sensor reading is:

$$T_{total} = 5.4 + 0.4 = 5.8ms$$

In this paper (1), which demonstrates a similar work compared to our projects, the author mentioned that the data of their system is captured at 80 frames per second. In this context, a frame of data refers to the collection of the reading at each sensor at approximately the same time. Therefore, we will assume that a capture rate of 80 frames per second is enough for our system to function properly. Therefore, the maximum amount time between each sensor reading iteration will be:

$$T_{required} = \frac{1}{80} = 0.0125s = 12.5ms > T_{total}$$

Therefore, the rate for master to acquire all sensor data is good enough for the system to perform its task.

3. Requirement and Verification

3.1 Requirement & Verification

3.1.1 Master Device Microcontroller

Requirement	Verification
1. Able to both receive and transmit data over SPI at speeds greater than 12Mbit/s while running the proper master device software.	1. <u>SPI Verification</u> a. Connect the MISO, MOSI, SCK and SS signals to oscilloscope to verify there are signals. b. Using the signal from SCK to check if the speed of SPI is greater than 12Mbits/s. c. Use the debugger built inside the STM32 development environment to check the content of data from slave device is correct.
2. Able to both receive and transmit data over I2C while running the proper master device software.	2. <u>I2C Verification</u> a. Connect the SDA and SCL signals to oscilloscope to verify there are signals. b. Connect the master device to PSYONIC hand and send predetermined command to the hand and see if the hand is controlled. 3. <u>Computational power verification</u>

3. The microcontroller (STM32F401RB) clock frequency should be higher than 64MHz.	Connect the system core clock output to the oscilloscope. The frequency of that signal should be higher than 64MHz.

3.1.2 Slave Device Microcontroller

Requirement	Verification
1. Able to both receive and transmit data over SPI at speeds greater than 12Mbits/s while running the proper slave device software.	1. SPI verification Please refer to the SPI verification of master device. If the Master Device SPI is proofed to be working, the Slave Device SPI have to work.
2. Able to convert analog signals ranging from 0-3.3V to corresponding 12-bit digital signals.	2. ADC verification a. Connect the ADC channel of the slave device microcontroller to the DC power supply. b. Adjust the output voltage of the DC power supply to be value between 0-3.3V. c. Use the debugger built inside the STM32 development environment to check the correctness of converted data.

3.1.3 DC-DC Converter

Requirement	Verification
1. Able to convert voltage ranging from 7.2-8.4V to 3.3±0.3V.	1. Provide 7.2-8.4V input voltage from DC power supply to the component. The output voltage should be 3.3±0.3V
2. Able to provide 500mA current to power the system.	2. Check if the system powered by the DC-DC Converter can function properly.

3.1.4 Analog MUXs

Requirement	Verification
1. The component is able to select between 8 analog signal ranging from 0-3.3V.	1. MUX selection verification a. Program the slave microcontroller to specify the Select Signal for the Analog MUX. b. Apply different pressures to eight sensors that are connected to the MUX being tested.

2. The component should have a on resistance lower than 10Ω.	c. Use the debugger built inside the STM32 development environment to check if the output of MUX reflects the change of pressure on the selected sensor. 2. Connect the output pin of the component to a 10Ω resistor. Provide 3.3V signal to input 0 and change selection signal to select input 0. The voltage between input 0 and output should be less than 3.3/2=1.65V.

3.1.5 Pressure Sensor

Requirement	Verification
1. The pressure sensor characteristic should report approximately same signal for same compression of a properly chosen form and different sensors should report approximately same signal for same compression. The difference between different test trial on the same sensor should be less than 1% and the difference between different sensors should be less than 10%.	1. <u>Verification Process</u> a. Connect the master device to a PC via a UART bridge. b. Program the master microcontroller to collect digitized data from slaves and transfer the data to PC using UART protocol. c. Compress the foam on top of the all the sensor to a specific distance for five times. Choose 10 different distance within the working range of the foam and repeat the test for 10 time respectively. d. Compare the value transferred from master microcontroller in the PC and check if the differences are larger than expected. e. Port the data to Matlab and plot a heatmap of the reported pressure distribution for more visualized test results.

3.1.6 Mechanical Constraint

Requirement	Verification
1. The X and Y dimension of the master device PCB should both be smaller than 3cm.	1. Measure the dimension of the manufactured master device PCB to check if it is oversized.

2. The X dimension of the slave device PCB should be smaller than 3cm, and the Y dimension should be smaller than 10cm.	2. Measure the dimension of the manufactured slave device PCB to check if it is oversized.

3.2 Tolerance Analysis

The main error source in the system is that the analog reading from pressure sensor may be noisy and will potentially harm the categorization if the erroneous data are transferred to the master device. The noise may be caused by unstable contact between conducting foam and human skin or electrodes.

To estimate the error rate, the following assumptions are made:

a. The probability of one pressure sensor fails is $(1-\overline{\boldsymbol{p}})$ and its failure profile follows the Weibull distribution.
b. There are n sensors in total and if no less than m of the sensors reads uncorrupted signals at the same time, this set of data, i.e., a frame, is treated as valid and used by the classification program.
c. For each trial, i.e., collecting one frame of data from all sensors, the error is only due to random noise and $(1-\overline{\boldsymbol{p}})$ remains constant all the time

Therefore, $\boldsymbol{e(i)} = \binom{n}{i}\overline{\boldsymbol{p}}^{\boldsymbol{i}}(\boldsymbol{1}-\overline{\boldsymbol{p}})^{\boldsymbol{n-i}}$ denotes probability mass function of error as a function of the number of properly working sensor. It satisfies binomial distribution and the mean is $\boldsymbol{np}$. Define the probability that a frame is valid as below and denote it as $\boldsymbol{q}$:

$$\boldsymbol{q} = \boldsymbol{P(valid\ frame)} = \sum_{i=m}^{n}\binom{n}{i}\overline{\boldsymbol{p}}^{\boldsymbol{i}}(\boldsymbol{1}-\overline{\boldsymbol{p}})^{\boldsymbol{n-i}}$$

According to section 2.6.3, the maximum frames that the master microcontroller can process in one second is

$$\boldsymbol{floor}\left(\frac{\boldsymbol{1s}}{\boldsymbol{5.8ms}}\right) = \boldsymbol{172}$$

It does no harm to assume that we will tune the running speed of the system manually so 150 frames will be collected per second. The minimum frames per second needed to recognize hand motion is 80 so the system can tolerate dropping 70 frames per seconds. Therefore, the probability of the system failing in one second is:

$$\boldsymbol{P(fail)} = \boldsymbol{1} - \sum_{i=80}^{150}\binom{\boldsymbol{150}}{\boldsymbol{i}}\boldsymbol{q^i(1-q)^{150-i}}$$

While designing the sensor, the value of $\boldsymbol{P(fail)}$ should be minimized to a reasonably small number to be negligible. Plugging in the values for $\boldsymbol{n = 300, m = 240,}$ we discover that $\boldsymbol{P(fail)}$ approaches 0 rapidly in the range $\overline{\boldsymbol{p}} \in [\mathbf{0.8}, \mathbf{0.81}]$ and

$$\boldsymbol{P(fail) < 5.60 \times 10^{-6}\ for\ \overline{p} > 0.81}$$

We are planning to conduct tests on $\mathbf{300}$ sensors by applying a pressure on them and recording the number of uncorrupted ones. The calculate probability is denoted as $\boldsymbol{p}$ and we can assume that the sample is large so the distribution of $\boldsymbol{p}$ can be treated as approximately normal. The variance of $\boldsymbol{p}$ is $\boldsymbol{p(1-p)}$ and therefore the $\mathbf{95}\%$ confidence interval ($\boldsymbol{\alpha = 0.05}$) for $\boldsymbol{p}$ can then be calculated using the following formula:

$$\left\{\boldsymbol{p - u_\alpha} \times \sqrt{\frac{\boldsymbol{p(1-p)}}{\boldsymbol{n}}}, \infty\right\}$$

When $\boldsymbol{p = 0.85}$, the lower bound of the interval is $\mathbf{0.81609}$, so we are $\mathbf{95}\%$ confident that if the observed value of $\boldsymbol{p = 0.85}$, the real value of $\overline{\boldsymbol{p}} > \mathbf{0.81}$.

In conclusion, the probability of a sensor being **uncorrupted** observed during tests $\boldsymbol{p}$ must be larger than $\mathbf{0.85}$, i.e.,

$$\boldsymbol{p \geq 0.85}$$

Now assume the we want to keep this probability of a sensor being **uncorrupted** after $4000hrs$ of normal use. And given that the failure rate of a sensor follows the Weibull distribution, which has the following PDF

$$\boldsymbol{f(x|\lambda, k) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{(k-1)} e^{-(x/\lambda)^k} \quad (x \geq 0)}$$

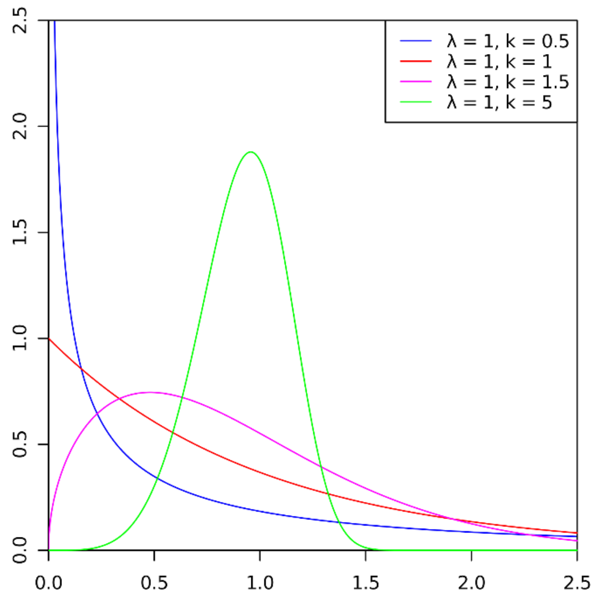


Figure 17: Different shapes of Weibull Distribution

By convention, the pressure sensors will experience a relatively higher failure rate in their early life, a phenomenon known as the early "infant mortality" failure. And we could further assume the shape parameter to be $[1.0, 1.5]$ and we will take $k = 1.3$ for our calculation.

Thus, we have the following equation to solve for the scale parameter λ

$$\mathbf{1 - (1 - e^{-(4000/\lambda)^{1.3}}) = 0.85}$$

Where $\boldsymbol{\lambda} = \mathbf{16182.9}$, so the Weibull Distribution of our product looks like,

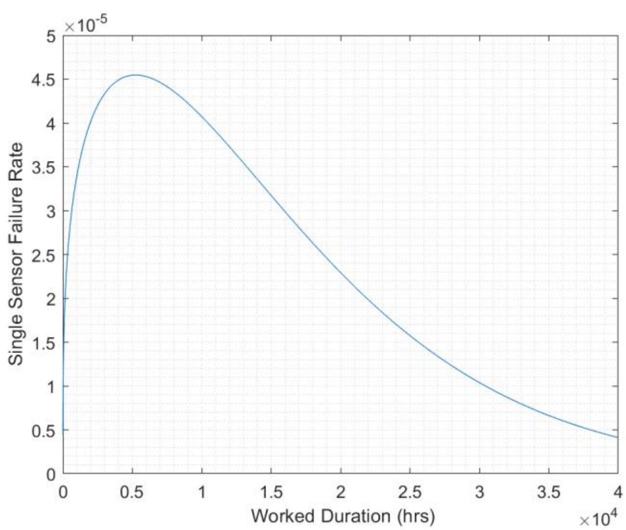


Figure 18: Single Sensor Failure Rate

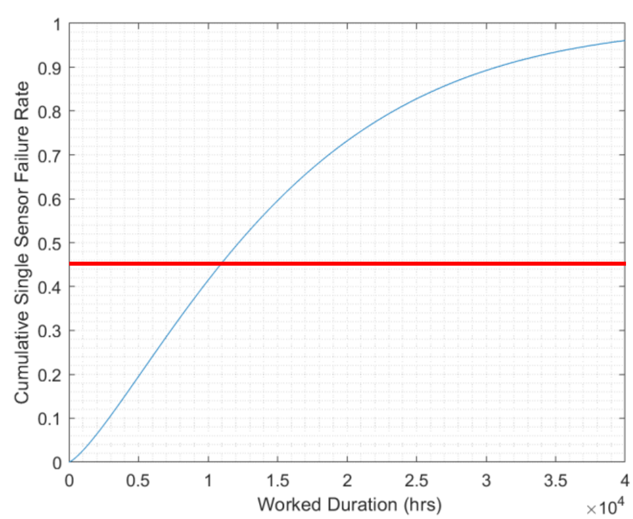


Figure 19: Cumulative Single Sensor Failure Rate

The red line on the CDF is the critical failure rate for our product, located at 0.4667.

4. Cost and Schedule

4.1 Cost Analysis

4.1.1 Cost of All Parts (Currency in USD)

Part	Part Number	Cost/Unit	Quantity	Subtotal	Provider
10 x Slave Module					
4 layers PCB	with ENIG finish	3.5	10	**35**	JLCPCB
Microcontroller	STM32F030K6T6	1.26	10	**12.6**	STMicroelectronics
0.1 uF Capacitor	CC0402JRX5R6BB104	0.039	40	**1.56**	Yageo
4.7 uF Capacitor	EMK107ABJ475MA-T	0.124	20	**2.48**	Taiyo Yuden
7-pin Connector	53261-0771	1.26	10	**12.6**	Molex
2-pin SMD Header	N/A	0	10	**0**	N/A
10k Resistor	SFR01MZPJ103	0.023	300	**6.9**	ROHM Semiconductor
8-input MUX	NX3L4051HR	0.849	40	**33.96**	NXP Semiconductors
1 x Master Module					
PCB	with HASL finish	0.2	1	**0.2**	JLCPCB
Microcontroller	STM32F401RBT6	5.33	1	**5.33**	STMicroelectronics
SMD LED	SML-LXFT0603UPGCTR	0.806	20	**16.12**	Lumex
2-pin SMD Header	N/A	0	1	**0**	N/A
2-pin Connector	53261-0271	0.776	1	**0.776**	Molex
4-pin Connector	53261-0471	0.983	1	**0.983**	Molex
7-pin Connector	53261-0771	1.26	2	**2.52**	Molex
Linear Regulator	TPS82140SILR	3.38	1	**3.38**	Texas Instruments
0.1 uF Capacitor	CC0402JRX5R6BB104	0.039	5	**0.195**	Yageo
4.7 uF Capacitor	EMK107ABJ475MA-T	0.124	4	**0.496**	Taiyo Yuden
22 uF Capacitor	GRM188R61A226ME15J	0.35	1	**0.35**	Murata Electronics
33 nF Capacitor	AC0402KRX7R8BB333	0.053	1	**0.053**	Yageo
470 Resistor	RR0510P-471-D	0.076	1	**0.076**	Susumu
309k Resistor	RC0402FR-07309KL	0.012	1	**0.012**	Yageo
100k Resistor	RT0402FRE07100KL	0.056	2	**0.112**	Yageo
MISC					
ST-LINK in-circuit debugger/programmer	N/A	8.6	1	**8.6**	N/A
FTDI FT232 USB to UART Converter	N/A	10.25	1	**10.25**	N/A
			Total:	**154.553**	

4.1.2 Cost of Labor (Currency in USD)

Worker	Pay Rate (Semi-Skill)	Weekly Hours	Total Pay
Yangge Li	25	12	300
Enliang Li	25	12	300
Zhoushi Zhu	25	12	300
		Total:	**900**

If we assume the actual working period takes about 12 weeks, the total cost of labor is about $\$900 * 12 * 2.5 = \$27000$

4.2 Timeline

4.2.1 Group Member 1: Yangge Li

YANGGE LI	Monday	Tuesday	Wednesday	Thursday	Friday
Week 8 (3/4/2019)	Teamwork Evaluation	Soldering the Master Chip	Soldering the Master Chip	Meeting with machine learning guys	Master Chip Ver 2.0 Sentout [if needed]
Week 9 (3/11/2019)	Soldering the Sensors Chip	Verify the hardware of Master Chip Ver 1.0		Gereral meeting with PSYONIC INC.,	
Week 10 (3/18/2019)		Sensors Chip Ver 2.0 Sentout [if needed]	Soldering the Sensors Chip	Soldering the Sensors Chip	
Week 11 (3/25/2019)	Individual progress reports		Testing the foam used for pressure sensing	Hardware Progress Report to PSYONIC INC.,	Testing the foam used for pressure sensing
Week 12 (4/1/2019)		Finalize the design of both PCBs (Master + Sensors)			
Week 13 (4/8/2019)			Integration between software and hardware	Integration between software and hardware	Integration between software and hardware

4.2.2 Group Member 2: Enliang Li

ENLIANG LI	Monday	Tuesday	Wednesday	Thursday	Friday
Week 8 (3/4/2019)	Teamwork Evaluation	Design Document Modifications		Meeting with machine learning guys	Master Chip Ver 2.0 Sentout [if needed]
Week 9 (3/11/2019)		Verify the MCU on Master Chip using R&V table		Gereral meeting with PSYONIC INC.,	
Week 10 (3/18/2019)		Sensors Chip Ver 2.0 Sentout [if needed]	First version of SPI communication software compiled		
Week 11 (3/25/2019)	Individual progress reports			Communication Software Ver 1.0	Downloading the code to master chip MCU
Week 12 (4/1/2019)		Debug and Testing on the combined software		Software Progress Report to PSYONIC INC.,	
Week 13 (4/8/2019)		Communication Software Ver 2.0	Integration between software and hardware	Integration between software and hardware	Integration between software and hardware

4.2.3 Group Member 3: Zhoushi Zhu

ZHOUSHI ZHU	Monday	Tuesday	Wednesday	Thursday	Friday
Week 8 (3/4/2019)	Teamwork Evaluation	Design Document Modifications		Meeting with machine learning guys	Master Chip Ver 2.0 Sentout [if needed]
Week 9 (3/11/2019)	Soldering Sensors Chip	Verify the hardware of Sensors Chip Ver 1.0		Gereral meeting with PSYONIC INC.,	
Week 10 (3/18/2019)		Sensors Chip Ver 2.0 Sentout [if needed]	First version of slave device software compiled		
Week 11 (3/25/2019)	Individual progress reports			Communication Software Ver 1.0	Downloading the code to sensor chip MCU
Week 12 (4/1/2019)		Debug and Testing on the combined software		Software Progress Report to PSYONIC INC.,	
Week 13 (4/8/2019)		Communication Software Ver 2.0	Integration between software and hardware	Integration between software and hardware	Integration between software and hardware

5. Safety and Ethics

5.1 Safety

Our major safety concern during the design process is the potential circuit short which may lead to extreme high temperature due to current surge if the PCB carries serious bugs or human error while testing. Circuit short hazard could burn down the PCB or scald our skins, and thus, it needs to be taken seriously. We won't allow any one in our group working alone with the PCB(s).

Since the prosthetic hand is comprised of many mechanical components, we also need to take precautions to avoid any possible cutting injuries caused by improper operations.

5.2 Ethics

After reviewing the IEEE and ACM ethics, we agree on the following concerns to be presented in our project proposal,

5.2.1 IEEE Policies, Section 7, 7.8 IEEE Code of Ethics

8. to treat fairly all persons and to not engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression [6]

We need to be honest on the performance of our product. Due to current technical limitation, both EMG and pressure-sensing method could only work for the disabled who had amputation surgery below the elbow (and still have working residual limbs). We expect our product to work poorly for those who experienced amputation surgery above the elbow.

This may be a discrimination regarding the degree of disability.

9. to avoid injuring others, their property, reputation, or employment by false or malicious action [6]

&

ACM General Ethical Principles 1.2 Avoid harm. [7]

Our product may injure/harm (bring negative consequence) the user, or the objects hold by the prosthetic hand due to incorrect responds that against user's will. Furthermore, since user could not always control the force it applies to the project precisely, it's very possible that soft items are squeezed.

5.2.2 ACM General Ethical Principles

1.6 Respect privacy. [7]

As part of the training process, we need to collect sensitive data from user, such as the pressure patterns for different hand movements and store them into the master chip, which assumes the possibility of user's data leakage if the product hacked or missing.

References

[1] C. Castellini, R. Kõiva, C. Pasluosta, C. Viegas, and B. M. Eskofier, "Tactile Myography: An Off-Line Assessment of Able-Bodied Subjects and One Upper-Limb Amputee," *MDPI*, 23-Mar-2018. [Online]. Available: http://www.mdpi.com/2227-7080/6/2/38. [Accessed: 01-Feb-2019].

[2] STMicroelectronics, "STM32F401xB STM32F401xC Datasheet," *STMicroelectronics*, 06-Sep-2013. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32f401cb.pdf. [Accessed: 01-Feb-2019].

[3] STMicroelectronics, "STM32F030x4 STM32F030x6 STM32F030x8 STM32F030xC Datasheet," *STMicroelectronics*, 04-Jul-2013. [Online]. Available: https://www.st.com/resource/en/datasheet/stm32f030k6.pdf. [Accessed: 01-Feb-2019].

[4] Texas Instruments, "TPS84210 2.95-V to 6-V Input, 2-A Synchronous Buck, Integrated Power Solution," *Texas Instruments*, 11-Apr-2018. [Online]. Available: http://www.ti.com/lit/ds/symlink/tps84210.pdf. [Accessed: 01-Feb-2019].

[5] NXP Semiconductors N.V., "NX3L4051 Product data sheet," NXP Semiconductors N.V., 03-Jul-2012. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/NX3L4051.pdf. [Accessed: 01-Feb-2019].

[6] Ieee.org, "IEEE IEEE Code of Ethics", 2016. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 1- Feb- 2019].

[7] "ACM Code of Ethics and Professional Conduct," *ACM Code of Ethics and Professional Conduct*. [Online]. Available: https://www.acm.org/binaries/content/assets/about/acm-code-of-ethics-and-professional-conduct.pdf. [Accessed: 01-Feb-2019].

[8] Zhang, C. (2019). How to run deep learning model on microcontroller with CMSIS-NN (Part 1). [Blog] DLology. Available at: https://www.dlology.com/blog/how-to-run-deep-learning-model-on-microcontroller-with-cmsis-nn/ [Accessed 19 Feb. 2019].