**Internship Experience**

**Glove Project**

**Home Team Science and Technology Agency (HTX)**

Table of Contents

[**Project Information** 2](#_Toc110417735)

[Objective 2](#_Toc110417736)

[Resources/Materials 2](#_Toc110417737)

[1. Arduino Board 2](#_Toc110417738)

[2. Motor Drivers 2](#_Toc110417739)

[3. Stepper Motor 4](#_Toc110417740)

[4. FSR (Force Sensing Resistor) 5](#_Toc110417741)

[5. 7805 Voltage Regulator 5](#_Toc110417742)

[6. 3D Printed Structures 6](#_Toc110417743)

[7. Power Supply 6](#_Toc110417744)

[**Layout / Design (Mechanical & Electrical)** 7](#_Toc110417745)

[One finger 7](#_Toc110417746)

[Five fingers 9](#_Toc110417747)

[Circuit Connections 11](#_Toc110417748)

[**Bill of Materials** 13](#_Toc110417749)

[**Program Code** 14](#_Toc110417750)

[Force Sensing Resistor 14](#_Toc110417751)

[A4988 Motor Driver 14](#_Toc110417752)

[One Finger 15](#_Toc110417753)

[Five Fingers 15](#_Toc110417754)

# **Project Information**

## Objective

To use a force touch sensor to provide an equivalent force (haptic feedback) on the user’s fingers fitted in a glove.

## Resources/Materials

### Arduino Board

Due to the superior number of ports available with the Arduino MEGA, it is possible that the Arduino MEGA is more appropriate for usage as compared to the Arduino UNO. The table below shows the comparison:

|  |  |  |  |
| --- | --- | --- | --- |
| Board | Name | Arduino UNO R3 | Arduino Mega 2560 Rev3 |
| Pins | Digital I/O Pins | 14 | 54 |
| Analog input pins | 6 | 16 |
| PWM pins | 6 | 15 |

Arduino UNO: <https://docs.arduino.cc/hardware/uno-rev3>

Arduino MEGA: <https://docs.arduino.cc/hardware/mega-2560>

### Motor Drivers

There are 2 types of motor drivers, the L298N and the A4988. Either motor drivers can be utilised for the project.

* + L298N

According to a forum, the L298N is old and uses an inefficient technology. High current and low resistance motors will most likely be incompatible with this motor driver due to the absence of the current limiting feature.

Reference: <https://lastminuteengineers.com/stepper-motor-l298n-arduino-tutorial/> , <https://forum.pololu.com/t/which-to-use-l298n-h-bridge-or-the-a4988-stepper-motor-driver-carrier/9868>

Graphical user interface, application

Description automatically generated

* + A4988 (Stepper Driver)

VDD & GND is used for driving the internal logic circuitry (3V to 5.5V). VMOT & GND supplies power for the motor (8V to 35V).

A capacitor of 100uF (at least 47uF) should be used as a decoupling capacitor across the motor power supply pins to protect the driver from voltage spikes.

The MS1, MS2 & MS3 are three step size (resolution) selector inputs. The motor can be set to one of the five step resolutions by supplying the appropriate logic levels to these pins.

Table

Description automatically generated

Should these pins be disconnected, the motor will operate in full step mode.

There are 2 control inputs, STEP and DIR. STEP input controls the motor microsteps. DIR input controls the motor spinning direction – HIGH drives the motor clockwise, LOW drives the motor counterclockwise.

EN pin is an active low input – LOW will enable the A4988 driver. SLP pin is an active low input – LOW puts the driver in sleep mode, reducing the power consumption. RST is an active low input – when LOW, all STEP inputs are ignored until it becomes HIGH.

There are 4 output pins – 1B, 1A, 2A & 2B.

Current limiting is required before using the motor to limit the maximum amount of current flowing through the stepper coils and ultimately prevent it from exceeding the motor’s rated current. This is done by using a screwdriver and adjusting the current limiting potentiometer.

Diagram, schematic

Description automatically generated

Reference: <https://lastminuteengineers.com/a4988-stepper-motor-driver-arduino-tutorial/>

### Stepper Motor

The Arduino library will not be needed to be included should the A4988 Motor Driver be used.

Arduino Library: <https://www.arduino.cc/reference/en/libraries/stepper/stepper/>

### FSR (Force Sensing Resistor)

The force sensor is a resistor that changes its resistance, depending on how much it has been pressed. The harder the sensor is pressed (higher force), the lower the resistance between the two terminals will be.

Pins A0 to A5 on the Arduino UNO work as the analog input (A0 to A15 for Arduino MEGA). Analog input pins convert the voltage (between 0V and Vcc) into integer values (between 0 and 1023) referred to as ADC value or analog value.

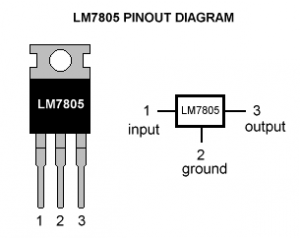
Connecting a pin of the force sensor to one of the analog input pins will allow us to read the analog value from the pin by using the analogRead() function, thereby giving us knowledge on how much has been pressed.



Reference: <https://arduinogetstarted.com/tutorials/arduino-force-sensor> , <https://lastminuteengineers.com/fsr-arduino-tutorial/#:~:text=Force%20Sensing%20Resistors%20are%20also,pressure%2C%20squeeze%2C%20and%20weight>.

### 7805 Voltage Regulator

The LM7805 is essentially a voltage regulator, providing a constant output of +5V. Being a three-pin IC, the input pin is for accepting incoming DC voltage, the ground pin acting as a ground, and the output pin that supplies +5V.



For the Glove application, the input voltage will be +12V. Decoupling capacitors will be included between the input and ground, as well as, the output and the ground.

Reference: <https://www.seeedstudio.com/blog/2019/10/30/lm7805-voltage-regulator-features-comparisons-lm317-and-more/#:~:text=The%20LM7805%20is%20a%20voltage,3%2DTerminal%20Regulators>

### 3D Printed Structures

* + End (Motor Side)
  + End (Opposite Side)
  + Plunger Holder
  + Slider
  + Syringe Holder

### Power Supply

A constant DC voltage of 12V will need to be supplied to the Stepper Motors and its corresponding drivers. For this, a power supply unit that converts the standard 230V to 12V is used.

Graphical user interface

Description automatically generated

# **Layout / Design (Mechanical & Electrical)**

## One finger

#### 1st Version

The design for a module of the mechanism is shown in the diagram below. To trigger the syringe, a push and pull mechanism is required. This is provided via the movable slider which relies on the motor rod and two corresponding guiding rods.

A picture containing timeline

Description automatically generated

Another method of guiding the slider is illustrated below. In this method, a simple modification is made. The two guiding rods are replaced by guiding walls, thereby acting as another way to secure the movement of the slider in a single movable axis. To facilitate smooth movements with minimal friction, a form of lubrication can be applied to the walls.

Graphical user interface

Description automatically generated

#### 2nd Version

After further discussion, the design was reviewed and changed. This version utilised an aluminium profile as the main sliding avenue, equipped with a 3D printed syringe holder that could move to and fro as required.

Diagram

Description automatically generated

However, after several tests, it was found that the syringe holders were unable to slide smoothly back and forth. This happened despite having conducted extensive fine tuning of the syringe holder.

#### 3rd Version

With the knowledge and experience acquired from designing the previous two versions, a third version was made. This version incorporates the use of a linear rail, which acts as the main sliding mechanism for the syringe holder. The linear rail also contains a sliding block attached to its rails, allowing additional parts to be attached to it, such as the syringe holder for this application. Diagrams depicting the third version can be found below.

Diagram

Description automatically generated

The syringe holder and plunger holder were also redesigned for integration with this version. A hex nut slot was carved in the middle of the syringe holder, to allow the motor threaded rod to be connected to the syringe holder. Also, a motor mount was included to allow the motor to sit above the linear rail and be positioned at an appropriate height for smooth operations.

Diagram, engineering drawing

Description automatically generated

## Five fingers

One way to assemble the five modules to the glove is by simply arranging the five modules together in a straight vertical line.

Diagram

Description automatically generated

The illustration below is another configuration that could also be used. It involves the modules to be placed in a formation that surrounds the upper half of the glove, resembling a semi-circle.

Diagram, engineering drawing

Description automatically generated

## Circuit Connections

The diagram below illustrates the circuit connections for one finger using the A4988 Motor Driver. The power circuit incorporates decoupling capacitors and an LM7805 Voltage Regulator to supply the 5V Output to the Arduino UNO board and the A4988 Motor Driver.

The analog input pin 0 (A0) is used as the input for detecting the force sensing resistor (FSR) values. Digital pins 2 and 3 are used for the STEP and DIR connectors on the A4988 Motor Driver respectively.

Diagram, schematic

Description automatically generated

0.33uF – LM7805 datasheet

47uF – A4988 Motor Driver

The circuit configuration for the integrated five fingers is similar. Five Stepper Motors, A4988 Motor Drivers, FSRs and 10k Ohm Resistors will be used. Nonetheless, there are some exceptions. There will only be one power circuit that supplies the 12V and 5V to the rest of the components.

Since five analog pins and 10 digital pins will be used; one Arduino UNO board appears capable of supporting the system. An Arduino MEGA could also come in handy should more functionalities be added.

If the L298N Motor Driver is used, the circuit connections will be tweaked slightly. An additional 2 digital pins on the Arduino board will be taken up. Another thing to take note is the A+, A-, B+ and B- wires of the NEMA 17 motor used. This can be determined through the manufacturer’s datasheet or by conducting a continuity test using a multimeter. The diagram below illustrates the circuit configuration.

Diagram, schematic

Description automatically generated

It is also to be noted that the LM7805 Voltage regulator will possibly not be used as the 5V can be supplied directly by the Arduino board. Hence, only a 12V input including a decoupling capacitor of 47uF will be used should this be the case.

# **Bill of Materials**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Item | Manufacturer | Qty. | Remarks (if any) |
| 1 | Arduino MEGA | Arduino | 1 |  |
| 2 | Arduino UNO |  | - | One in the lab |
| 3 | Stepper Motor | [From Sim Lim] | 5 |  |
| 4 | A4988 Motor Driver | [From Shopee] | 5 | Sim Lim Sunlight |
| 5 | Syringe | [From Shopee] | 5 | Check length and size |
| 6 | Tube | [From Shopee] | 5 | Check length and size |
| 7 | Force Sensing Resistor | [From Shopee] | 5 | Sim Lim Sunlight |
| 8 | 10k Ohm Resistor | [From Sim Lim] | 5 | Sim Lim Sunlight  Check in lab |
| 9 | LM7085 Voltage Regulator | - | - |  |
| 10 | 0.33uF/47uF Capacitor | [From Sim Lim] | - | Choose either one  Check in lab |
| 11 | 0.1uF Capacitor | [From Sim Lim] | - | Check in lab |
| 12 | 12V Power Supply | [From Sim Lim] | 1 |  |
| 13 | Linear Rail | Misumi | 5 | Check measurements |

# **Program Code**

## Force Sensing Resistor

Graphical user interface, text, application

Description automatically generated

## A4988 Motor Driver

For controlling multiple steppers, it is possible that that the AccelStepper library must be utilised. The documentation can be found [here](http://www.airspayce.com/mikem/arduino/AccelStepper/MultipleSteppers_8pde-example.html).

Graphical user interface, text, application

Description automatically generated

## One Finger

## Five Fingers