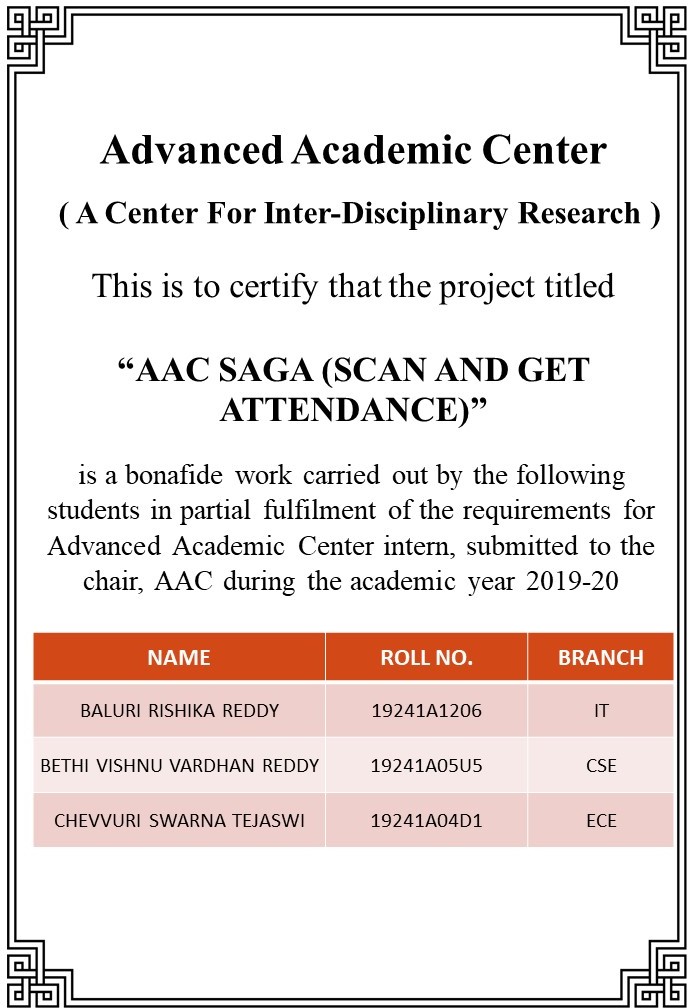


NAVEEN

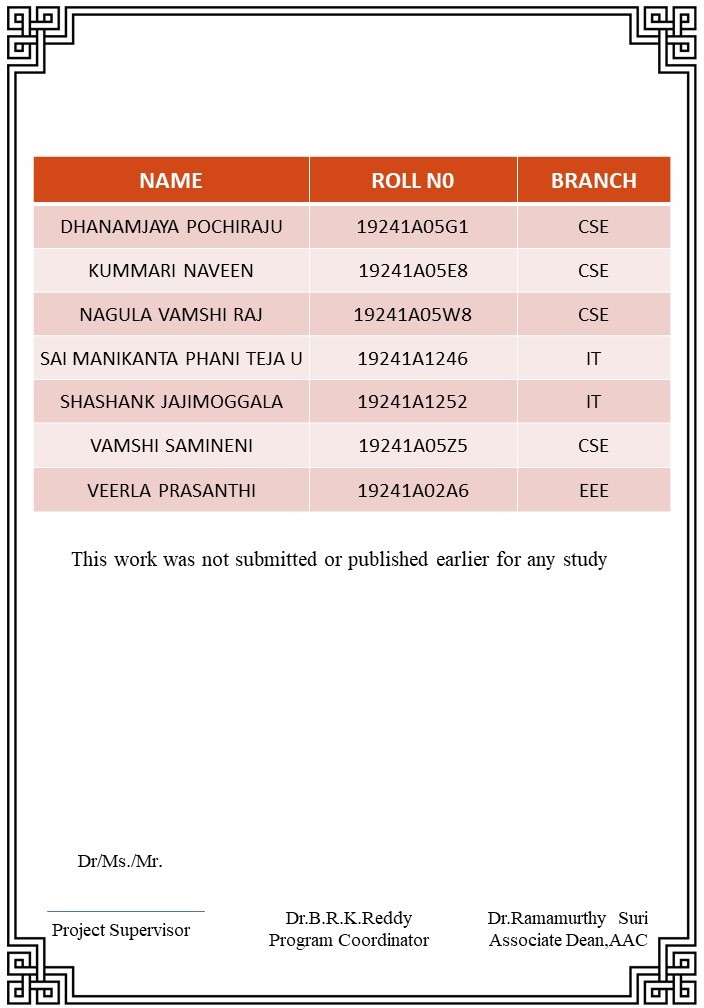
“ROBOTIC ARM POWERED BY EMG “

1



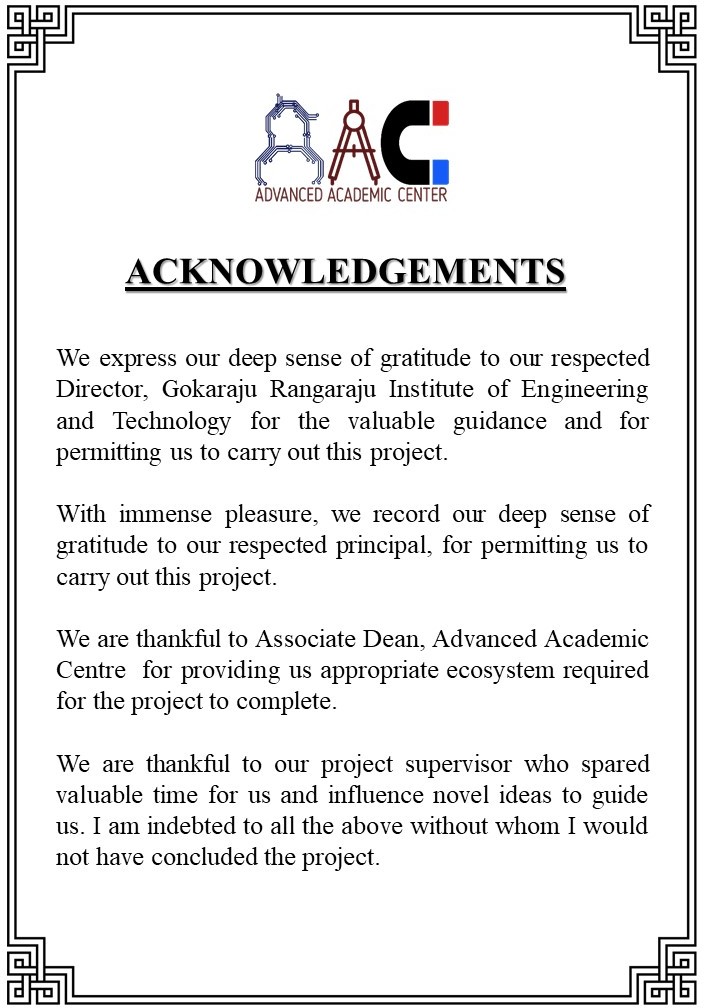
“ROBOTIC ARM POWERED BY EMG “

2



**This work was not submitted or published earlier for any study**

3



4

**1. Abstract**

***1.1 Problem Statement:***

To create a robotic arm, which is controlled by the sensory signals, captured by an EMG sensor.

***1.2 Problem solution:***

The robotic arm itself has been made out of cardboard. We have made a cardboard cutout of an arm, with sufficient space to accommodate the required circuitry within the cardboard model itself to eliminate the hassle of wires. Using a cardboard model also helped us to greatly reduce the overall cost of the project without compromising on the durability of the model. The cardboard arm contains a breadboard on the side, which is used to provide the necessary connections between the arm, servo motors, Arduino board, and the EMG sensor. The motion of the fingers is controlled by strings wound over each finger, which are in turn connected to servo motors.

***1.3 Functional Description:***

The entire circuitry has been assembled on the side of the model using a breadboard and wires. The connections are discussed in detail as we go further. The EMG sensor is to be attached to the user using three electrodes. The sensory signals obtained by the EMG sensor are sent as input to the Arduino board. The Arduino board, which has already been fed a cod which will be explained in detail later, rotates the servo motors after the signal received from the sensor cross a value which has been predetermined earlier in the code. The servo motors are in turn tied to strings, which are wound around the fingers of the model. Hence when the servo motors rotate, the fingers close, enabling the user to grip and release objects whenever they want.

We have used an Arduino Uno board in our project.

***1.4 Aim of The Project:***

* There is no commercially available solution for the paralyzed. The prosthetic limbs available for the amputees in the market are too expensive and only available in developed countries. With our project, we provide an affordable solution for the upper limb amputees.
* We provide a prosthesis which is much more functional than a typical hook or an artificial hand.

**2.Project Work Flow:**

CREATING A CARDBOARD MODEL OF A HAND WHICH CAN ACCOMMODATE ALL THE REQUIRED COMPONENTS AND CIRCUITRY

PROGRAMMING THE ARDUINO USING A CODE WHICH ROTATES THE SERVOS AFTER THE SENSORY SIGNAL CROSSES A PREDETERMINED VALUE

ASSEMBLING THE MODEL AND THE CIRCUITS TO FINISH THE WORKING PRODUCT.

The Project has been divided into three sub tasks to enable efficient working.

The sub tasks are as follows:

***Connecting the Arduino to EMG sensor:***

Khaja Faizan has successfully worked to give connections from Arduino to EMG sensor with the help of external sources, which was the crucial part of the project. And also, he worked to get readings of the muscle, which were required for working of the prosthetic arm.

***Programming the Arduino board:***

Kireeti has worked to write a code, which is suitable for working of the prosthetic arm. He wrote the code based on the readings given by Sai Gopi. The code is in such a way that – when the muscle contracts, fingers of the prosthetic arm will be closed and if the muscle is relaxed position, fingers of the prosthetic arm will be in open position.

***Prosthetic arm model:***

Jayanth designed a model of prosthetic arm, which is suitable to carry Arduino, EMG sensor with ease, which doesn’t disturb the person (person who uses this prosthetic arm) from their day-to-day activities.

The entire team has contributed wholeheartedly in the creation of this project.

Our mentor Naveen has constantly pointed us in the right direction and cleared our doubts. His support was instrumental in the making of this project

**3.Components and Connections:**

***3.1Required Hardware:***

Arduino Board (UNO or NANO)

Servo Motor (5 pcs)

EMG sensor

9V Battery (2Pcs)

9V Battery Connector (3Pcs)

Breadboard

Jumper Wires (Required number)

We have used Arduino UNO in our project. If one prefers to, they can also use Arduino NANO in this project as there are no major changes in the program.

**What is Arduino?**

Arduino is an open-source project that has designed a family of simple computers that provide an interface between sensors and control of physical devices. Because the schematics to design the products are open-source, other companies may freely reproduce the models as well. These unofficial devices are often cheaper and generally reliable. The Arduino “Uno” is a good starter model with a fair amount of inputs and outputs.

Arduino devices are programmed using a simplified version of the C language. A user writes code on a computer and transfers this code to the Arduino by USB. Then, USB or a battery can power the Arduino without the need for a computer.

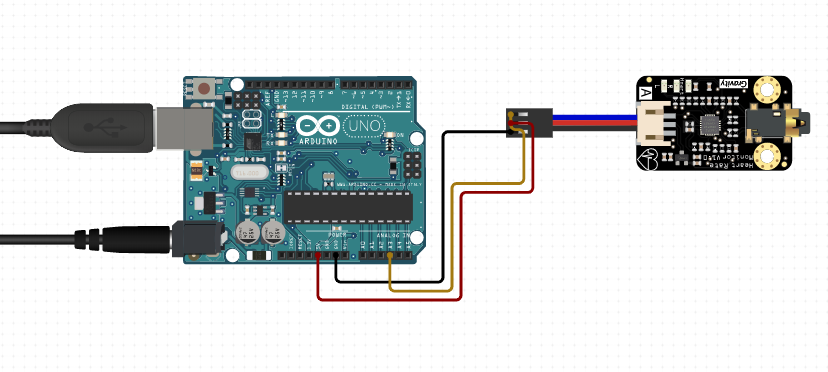
**What is an EMG Sensor?**

An electromyogram (EMG) measures the electrical activity of muscles at rest and during contraction.

EMG signals are used in many clinical and biomedical applications. EMG is used as a diagnostics tool for identifying neuromuscular diseases, assessing low-back pain, kinesiology, and disorders of motor control. EMG signals are also used as a control signal for prosthetic devices such as prosthetic hands, arms, and lower limbs.

***3.2.Connections:***

***3.2.1Connections from Arduino to EMG sensor:***



EMG sensor:

|  |  |
| --- | --- |
|  |  |

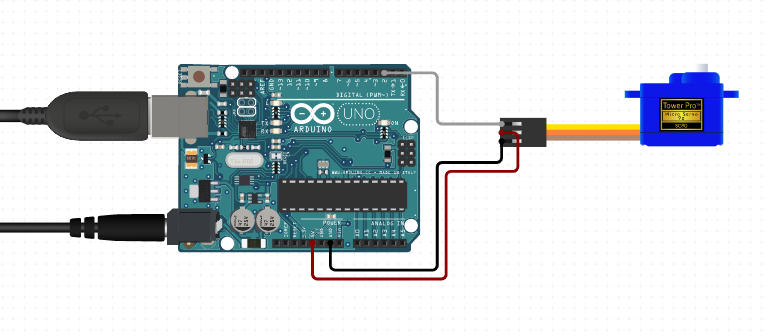
The connections are as follows:

* Firstly, we connect two 9V batteries in series. Then we connect positive end of the battery to ‘+Vs’ pin of the EMG sensor, negative end of the battery to ‘-Vs’ pin of the EMG sensor and the wires which remain are connected to the ground pin of the EMG sensor. This connection is responsible for the power supply to the sensor

**Caution:** This sensor needs a positive and negative reference voltage. Two power supplies are required. The sensor has a maximum operating voltage of ±18 V; however, we recommend using no higher than ±9 V to minimize the risk of electric shock.

* Now we connect ‘sig’ pin of the EMG sensor to any analogue pin of the Arduino and ‘GND’ pin of the EMG sensor to GND pin of the Arduino. This connection is responsible for the transfer of signals between the sensor and Arduino.
* EMG sensor is connected to our hand by using electrodes. These electrodes are connected to the sensor using an AUX cable.

***3.2.2.Connections from Arduino to Servo motors:***



SERVO motors:



The connections are as follows:

* For one servo motor, we can connect it directly to Arduino. Since we want to connect 5 servo motors, we use breadboard for this.
* We connect ‘5V’ point of the Arduino to a point in breadboard. This point is connected to servo motor using jumper cables.
* Similarly, we connect ‘GND’ point of the Arduino to another point of breadboard and this point is connected to grounding wire of servo motor using jumper cables.
* We connect remaining pin of the servo motor to ‘PWM’ pin of the Arduino. Muscle signal will be sent to servo motor through this connection.
* For most of them, Red wire will be connected to ‘5V’ pin, brown wire to ‘GND’ pin and orange wire to ‘PWM’ pin.

**4. Programming the Arduino Board:**

***4.1 Program:***

The program is the most essential part of any IOT project. Code for this prosthetic arm is based on a simple logic.

Let us consider ‘x’ be the reading when muscle is in relaxed position, ‘y’ be the reading when muscle is contracted. Then, let ‘z’ be the value such that

Z = (x + y)/2

The working of the code is in such a way that – if readings of EMG is less than ‘z’, the servo motors are rotated at an angle of 10 degrees and if readings of EMG is greater than ‘z’, the servo motors are rotated at an angle of 170 degrees. Since these servos are connected to fingers of the prosthetic arm with the help of strings, fingers of the model will be open when servo is at 10 degrees angle and fingers are closed when servo is at 170 degrees angle, creating a movement similar to gripping.

***4.2 Libraries used:***

We include only a single library - #include<servo.h>

This library will be helpful to control servo motors and makes the code simple.

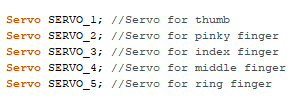


* Here we defined ‘Threshold’ as the mid value of our muscle readings as explained above. That is,

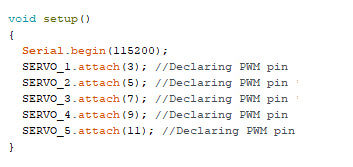
Threshold value = (value when muscle is in contract position +relaxed position)/2

For our project, threshold value is 500. This value varies from person to person since this is based on muscle structure of the individual

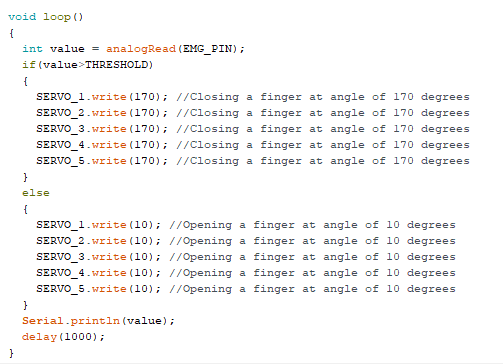
* Here we defined ‘EMG\_PIN’ as ‘0’. This represents that we are connecting ‘SIG’ pin of the EMG sensor to analog pin ‘0’ of the Arduino.



* Here we are declaring 5 servo motors for our movement of 5 fingers of the prosthetic arm. These servos can be attached directly to finger or by connecting threads for servo to finger based on prosthetic hand model.



* Here we declare PWM pin for each servo motor. That is, we are attaching Servo\_1 to PWM pin ‘3’. Similarly, Servo\_2 to PWM pin ‘5’, Servo\_3 to PWM pin ‘7’, Servo\_4 to PWM pin ‘9’, Servo\_5 to PWM pin ‘11’.
* This connection helps us to get data from Arduino which receives information from EMG.



This is the logic that makes the prosthetic arm work.

Int value = analogRead(EMG\_PIN);

This part of the code will help us to read the values that will be given by the EMG sensor and it is stored in the keyword ‘value’. The values (given by EMG sensor) which is stored in the keyword ‘value’ keeps on changing as it continuously gets the values from the sensor.

* The working of the code is as such:

If value is greater than the threshold value, all the servo motor will rotate at an angle of 170 degrees. Hence servo motor will pull the thread which is connected to the corresponding fingers making them close.

Similarly, if value is less than the threshold value, all servo motors will rotate at an angle of 10 degrees causing all the fingers to open.

Here delay(1000) represents, we want values from EMG sensor at a time difference of 1000 milli seconds.

This completes the code of our robotic arm

Final code of this project is:

#include <servo.h>

#define THRESHOLD 500

#define EMG\_PIN 0 //Declaring analog pin

Servo SERVO\_1; //Servo for thumb

Servo SERVO\_2; //Servo for pinky finger

Servo SERVO\_3; //Servo for index finger

Servo SERVO\_4; //Servo for middle finger

Servo SERVO\_5; //Servo for ring finger

void setup()

{

Serial.begin(115200);

SERVO\_1.attach(3); //Declaring PWM pin for

SERVO\_2.attach(5); //Declaring PWM pin for

SERVO\_3.attach(7); //Declaring PWM pin for

SERVO\_4.attach(9); //Declaring PWM pin for

SERVO\_5.attach(11); //Declaring PWM pin for

}

void loop()

{

int value = analogRead(EMG\_PIN);

if(value>THRESHOLD)

{

SERVO\_1.write(170); //Closing a finger at angle of 170 degrees

SERVO\_2.write(170); //Closing a finger at angle of 170 degrees

SERVO\_3.write(170); //Closing a finger at angle of 170 degrees

SERVO\_4.write(170); //Closing a finger at angle of 170 degrees

SERVO\_5.write(170); //Closing a finger at angle of 170 degrees

}

else

{

SERVO\_1.write(10); //Opening a finger at angle of 10 degrees

SERVO\_2.write(10); //Opening a finger at angle of 10 degrees

SERVO\_3.write(10); //Opening a finger at angle of 10 degrees

SERVO\_4.write(10); //Opening a finger at angle of 10 degrees

SERVO\_5.write(10); //Opening a finger at angle of 10 degrees

}

Serial.println(value);

delay(1000);

}

**5. Scope for Future Developments:**

* Though the code is written for the operation of five servos, in the working model of our project, we have incorporated only two. Increasing the number of servo motors used can reduce the strain on the servos.
* Having more than one threshold value can enable us to control the movement of each finger separately, making the model much more efficient and closer to a real hand.
* A stronger and more durable material can be used to make the prosthetic arm.
* This technology can also be extended to control robotic arms remotely using a Bluetooth or a Wi-Fi module.

**6. Applications:**

* This robotic arm is a cost effective and extremely useful prosthesis for upper limb amputees. As we use the sensory signals generated in the muscles of amputees themselves, it makes it much more easier and effective to use.
* Though our primary aim was developing a prosthesis operated by the amputee in his vicinity, with the help of a Wi-Fi or a Bluetooth module, this technology can also be used to control robotic arms remotely. This enables people working with hazardous materials and environments to be safer in their workplaces.

**7. References:**

* <https://youtu.be/D-6GDlvAMCI>
* <https://youtu.be/c9FuPdl3xCE>
* <https://imotions.com/blog/electromyography-101/>
* <https://www.intechopen.com/books/computational-intelligence-in-electromyography-analysis-a-perspective-on-current-applications-and-future-challenges/signal-acquisition-using-surface-emg-and-circuit-design-considerations-for-robotic-prosthesis>