[#include](https://www.roboticskanti.com/blog/hashtags/include" \t "_self) <Servo.h>

[#define](https://www.roboticskanti.com/blog/hashtags/define) THRESHOLD 150 //In order to determine the state of the hand (opened/closed)

[#define](https://www.roboticskanti.com/blog/hashtags/define) EMGPIN 3 //Analog pin connected to Muscle Sensor V3 Board

[#define](https://www.roboticskanti.com/blog/hashtags/define) LITTLEPIN 2 //Digital pin used by Little servo

[#define](https://www.roboticskanti.com/blog/hashtags/define) RINGPIN 3 //Digital pin used by Ring servo

[#define](https://www.roboticskanti.com/blog/hashtags/define) MIDDLEPIN 4 //Digital pin used by Middle servo

[#define](https://www.roboticskanti.com/blog/hashtags/define) INDEXPIN 5 //Digital pin used by Index servo

[#define](https://www.roboticskanti.com/blog/hashtags/define) THUMBPIN 6 //Digital pin used by Thumb servo

//Constants used to open and close the fingers

[#define](https://www.roboticskanti.com/blog/hashtags/define) LITTLE 1

[#define](https://www.roboticskanti.com/blog/hashtags/define) RING 2

[#define](https://www.roboticskanti.com/blog/hashtags/define) MIDDLE 3

[#define](https://www.roboticskanti.com/blog/hashtags/define) INDEX 4

[#define](https://www.roboticskanti.com/blog/hashtags/define) THUMB 5

Servo servoLittleFinger; // Define servo fingers

Servo servoRingFinger; // Define servo fingers

Servo servoMiddleFinger; // Define servo fingers

Servo servoIndexFinger; // Define servo fingers

Servo servoThumbFinger; // Define servo fingers

int finger;

// Motion routines for handopen and handclose

void openhand(){for(finger = 1; finger < 6; finger++){openFinger(finger);}}

void closehand(){for(finger = 1; finger < 6 ; finger++){closeFinger(finger);}}

// You have to rewrite properly the functions to open and close the fingers

// according of your assembly

// In my case, middle and index fingers are opened when servo is at 170 degrees and the others when servo is at 0 degrees

// I have used Towardpro MG996R servos

void openFinger(int finger){

if(finger==LITTLE){servoLittleFinger.write(0);} // Little finger

else if(finger==RING){servoRingFinger.write(170);}// Ring finger

else if(finger==MIDDLE){servoMiddleFinger.write(170);}// Middle finger

else if(finger==INDEX){servoIndexFinger.write(170);}// Index finger

else if(finger==THUMB){servoThumbFinger.write(0);}//Thumb finger

}

void closeFinger(int finger){

if(finger==LITTLE){servoLittleFinger.write(170);} // Little finger

else if(finger==RING){servoRingFinger.write(0);}// Ring finger

else if(finger==MIDDLE){servoMiddleFinger.write(0);}// Middle finger

else if(finger==INDEX){servoIndexFinger.write(0);}// Index finger

else if(finger==THUMB){servoThumbFinger.write(170);}//Thumb finger

}

void setup(){

Serial.begin(115200); //BAUDRATE set to 115200, remember it to set monitor serial properly (used this BaudRate and "NL&CR" option to visualize the values correctly)

servoLittleFinger.attach(LITTLEPIN); // Set Little finger servo to digital pin 3

servoRingFinger.attach(RINGPIN); // Set Ring finger servo to digital pin 5

servoMiddleFinger.attach(MIDDLEPIN); // Set Middle finger servo to digital pin 6

servoIndexFinger.attach(INDEXPIN); // Set Index finger servo to digital pin 9

servoThumbFinger.attach(THUMBPIN); // Set Thumb finger servo to digital pin 10

}//end setup

void loop() {// Nothing to do here, all is done in the interrupt function

int value = analogRead(EMGPIN); //Sampling analog signal

if(value>THRESHOLD) //If the value of the sample is greater than THRESHOLD means that the hand has been closed

{closehand();}

else //Otherwise the hand is open

{openhand();}

Serial.println(value); //You can use serial monitor to set THRESHOLD properly, comparing the values shown when you open and close your hand

}