Prothese de main arduino

Prothese main arduino 001

Capteur pour muscle Myoware

Electrodes pour muscle

Main Robotisée HackBerry

Fabrication collaborative Main-robotisée site e-nable

Doc Main robotisée

Programme main robotisée HackBerry GitHub

Prothese main robot Arduino

Bionicohan FR

Capteurs EMG

Capteurs EMG c'est Quoi ? FR

EMG sur Alliexpress

DIY_-_Faites_le_vous-même-_Capteur_EMG.pdf

video capteur EMG EN

test capteurs EMG

Programme HackBerry.ino version V1.0 MK2

HackberryMk2v1.0.ino

```
/*
 *  Arduino micro code for HACKberry Mk2.
 *  Origially created by exiii Inc.
 *  edited by Kouki Shinjo on 2018/04/26
 */
#include <Servo.h>

//Settings
const boolean isRight = 0;//right:1, left:0
const int outThumbMax = 95;//right:open, left:close
const int outIndexMax = 100;//right:open, left:close
const int outOtherMax = 75;//right:open, left:close
const int outThumbMin = 10;//right:close, left:open
const int outIndexMin = 0;//right:close, left:open
const int outOtherMin = 20;//right:close, left:open
const int speedMax = 6;
const int speedMin = 0;
const int speedReverse = -3;
const int thSpeedReverse = 15;//0-100
const int thSpeedZero = 30;//0-100
const boolean onSerial = 0; // 1 is not recommended

//Pin
int pinButtonCalib; //start calibration
int pinButtonTBD; // No function implemented yet.
int pinButtonThumb; // open/close thumb
int pinButtonOther; //lock/unlock other three fingers
int pinServoIndex;
int pinServoOther;
int pinServoThumb;
int pinSensor; //sensor input

//Hardware
Servo servoIndex; //index finger
Servo servoOther; //other three fingers
Servo servoThumb; //thumb

//Software
boolean isThumbOpen = 1;
boolean isOtherLock = 0;
int swCount0,swCount1,swCount2,swCount3 = 0;
int sensorValue = 0; // value read from the sensor
int sensorMax = 700;
int sensorMin = 0;
int speed = 0;
int position = 0;
const int positionMax = 100;
const int positionMin = 0;
int prePosition = 0;
```

```
int outThumb,outIndex,outOther = 90;
int
outThumbOpen,outThumbClose,outIndexOpen,outIndexClose,outOtherOpen,outO
therClose;

void setup() {
    if (onSerial)  Serial.begin(9600);

    // Pin Configuration
    pinButtonCalib = A6;// A4 sur Arduino Uno pour test Calibration
    pinButtonTBD   = A7;// A5 sur Arduino Uno pour test
    pinButtonThumb = A0;// Verrouillage pouce
    pinButtonOther = 10;// Verouillage 3 doigts
    pinServoIndex  = 5;// Servo pour l'index
    pinServoOther  = 6;// Servo pour 3 doigts
    pinServoThumb  = 9;// Servo pour le pouce
    pinSensor      = A1; // Capteur de pression

    if(isRight){
        outThumbOpen=outThumbMax; outThumbClose=outThumbMin;
        outIndexOpen=outIndexMax; outIndexClose=outIndexMin;
        outOtherOpen=outOtherMax; outOtherClose=outOtherMin;
    } else {
        outThumbOpen=outThumbMin; outThumbClose=outThumbMax;
        outIndexOpen=outIndexMin; outIndexClose=outIndexMax;
        outOtherOpen=outOtherMin; outOtherClose=outOtherMax;
    }
    servoIndex.attach(pinServoIndex);//index
    servoOther.attach(pinServoOther);//other
    servoThumb.attach(pinServoThumb);//thumb

    pinMode(pinButtonCalib, INPUT_PULLUP);
    pinMode(pinButtonTBD,   INPUT_PULLUP);
    pinMode(pinButtonThumb, INPUT_PULLUP);
    pinMode(pinButtonOther, INPUT_PULLUP);
}

void loop() {
    //==waiting for calibration==
    if(onSerial) Serial.println("======Waiting for Calibration======");
    while (1) {
        servoIndex.write(outIndexOpen);
        servoOther.write(outOtherOpen);
        servoThumb.write(outThumbOpen);
        if(onSerial) serialMonitor();
        delay(10);
        if (readButton(pinButtonCalib) == LOW) {
            calibration();
            break;
        }
    }
```

```
    //==control==
    position = positionMin;
    prePosition = positionMin;
    while (1) {
        if (readButton(pinButtonCalib) == LOW) swCount0 += 1;
        else swCount0 = 0;
        if (swCount0 == 10) {
            swCount0 = 0;
            calibration();
        }
        if (readButton(pinButtonTBD) == LOW) swCount1 += 1;
        else swCount1 = 0;
        if (swCount1 == 10) {
            swCount1 = 0;
            // Do something here
            while (readButton(pinButtonTBD) == LOW) delay(1);
        }
        if (readButton(pinButtonThumb) == LOW) swCount2 += 1;
        else swCount2 = 0;
        if (swCount2 == 10) {
            swCount2 = 0;
            isThumbOpen = !isThumbOpen;
            while (readButton(pinButtonThumb) == LOW) delay(1);
        }
        if (readButton(pinButtonOther) == LOW) swCount3 += 1;//A3
        else swCount3 = 0;
        if (swCount3 == 10) {
            swCount3 = 0;
            isOtherLock = !isOtherLock;
            while (readButton(pinButtonOther) == LOW) delay(1);
        }

        sensorValue = readSensor();
        delay(25);
        if(sensorValue<sensorMin) sensorValue=sensorMin;
        else if(sensorValue>sensorMax) sensorValue=sensorMax;
        sensorToPosition();

        outIndex = map(position, positionMin, positionMax,
outIndexOpen, outIndexClose);
        servoIndex.write(outIndex);
        if (!isOtherLock){
            outOther = map(position, positionMin, positionMax,
outOtherOpen, outOtherClose);
            servoOther.write(outOther);
        }
        if(isThumbOpen) servoThumb.write(outThumbOpen);
        else servoThumb.write(outThumbClose);
        if(onSerial) serialMonitor();
    }
}
```

```
/*
* functions
*/
boolean isDigitalPin(const int pin) {
    return (pin >= 0) && (pin <= 19) ? true : false;
}

boolean readButton(const int pin) {
    if ( isDigitalPin(pin) ) {
        return digitalRead(pin);
    } else {
        if (analogRead(pin) > 512) return HIGH;
        else return LOW;
    }
}

int readSensor() {
    int i, sval;
    for (i = 0; i < 10; i++) {
        sval += analogRead(pinSensor);
    }
    sval = sval/10;
    return sval;
}

void sensorToPosition(){
    int tmpVal = map(sensorValue, sensorMin, sensorMax, 100, 0);
    if(tmpVal<thSpeedReverse) speed=speedReverse;
    else if(tmpVal<thSpeedZero) speed=speedMin;
    else speed=map(tmpVal,40,100,speedMin,speedMax);
    position = prePosition + speed;
    if (position < positionMin) position = positionMin;
    if (position > positionMax) position = positionMax;
    prePosition = position;
}

void calibration() {
    outIndex=outIndexOpen;
    servoIndex.write(outIndexOpen);
    servoOther.write(outOtherClose);
    servoThumb.write(outThumbOpen);
    position=positionMin;
    prePosition=positionMin;

    delay(200);
    if(onSerial) Serial.println("======calibration start======");

    sensorMax = readSensor();
    sensorMin = sensorMax - 50;
    unsigned long time = millis();
```

```
    while ( millis() < time + 4000 ) {
        sensorValue = readSensor();
        delay(25);
        if ( sensorValue < sensorMin ) sensorMin = sensorValue;
        else if ( sensorValue > sensorMax )sensorMax = sensorValue;

        sensorToPosition();
        outIndex = map(position, positionMin, positionMax,
outIndexOpen, outIndexClose);
        servoIndex.write(outIndex);

        if(onSerial) serialMonitor();
    }
    if(onSerial)  Serial.println("======calibration finish======");
    return;
}

void serialMonitor(){
    Serial.print("Min="); Serial.print(sensorMin);
    Serial.print(",Max="); Serial.print(sensorMax);
    Serial.print(",sensor="); Serial.print(sensorValue);
    Serial.print(",speed="); Serial.print(speed);
    Serial.print(",position="); Serial.print(position);
    Serial.print(",outIndex="); Serial.print(outIndex);
    Serial.print(",isThumbOpen="); Serial.print(isThumbOpen);
    Serial.print(",isOtherLock="); Serial.println(isOtherLock);
}
```

Programme HackBerry.ino version V3.1 MK2

Hackberryv31MK2_V31.ino

```
/*
 *  Arduino micro code for HACKberry.
 *  Origially created by exiii Inc.
 *  edited by Genta Kondo on 2017/6/11
 */
#include <Servo.h>

//Settings
const boolean isRight = 1;//right:1, left:0

//For right hand, find optimal values of ThumbMin, IndexMax and
OtherMax first.
//For left hand, find optimal values of ThumbMax, IndexMin and OtherMin
first.
//Then, calculate the remaining values by following rules.
//Difference of ThumbMin and ThumbMax is 86
```

```
//Difference of IndexMin and IndexMax is 117
//Difference of OtherMin and OtherMax is 55

const int outThumbMax = 170;//right:open, left:close
const int outIndexMax = 142;//right:open, left:close
const int outOtherMax = 96;//right:open, left:close

const int outThumbMin = 170-86;//right:close, left:open
const int outIndexMin = 140-117; //right:close, left:open
const int outOtherMin = 95-55;//right:close, left:open

const int speedMax = 6;
const int speedMin = 0;
const int speedReverse = -3;
const int thSpeedReverse = 15;//0-100
const int thSpeedZero = 30;//0-100
const boolean onSerial = 0;   //Mk2 doesn't use serial monitor

//Hardware
Servo servoIndex; //index finger
Servo servoOther; //other three fingers
Servo servoThumb; //thumb
int pinCalib; //start calibration
//int pinTBD;
int pinThumb; // open/close thumb
int pinOther; //lock/unlock other three fingers
int pinSensor = A1; //sensor input

//Software
boolean isThumbOpen = 1;
boolean isOtherLock = 0;
int swCount0,swCount1,swCount2,swCount3 = 0;
int sensorValue = 0; // value read from the sensor
int sensorMax = 700;
int sensorMin = 0;
int speed = 0;
int position = 0;
const int positionMax = 100;
const int positionMin = 0;
int prePosition = 0;
int outThumb,outIndex,outOther = 90;
int
outThumbOpen,outThumbClose,outIndexOpen,outIndexClose,outOtherOpen,outO
therClose;

void setup() {
  Serial.begin(9600);
  if(isRight){
    pinCalib =  A6;
    //pinTBD =  A7;
    pinThumb =  A0;
```

```
    pinOther =  10;
    outThumbOpen=outThumbMax; outThumbClose=outThumbMin;
    outIndexOpen=outIndexMax; outIndexClose=outIndexMin;
    outOtherOpen=outOtherMax; outOtherClose=outOtherMin;
  }
  else{
    pinCalib =  A6;
    //pinTBD =  A7;
    pinThumb =  A0;
    pinOther =  10;
    outThumbOpen=outThumbMin; outThumbClose=outThumbMax;
    outIndexOpen=outIndexMin; outIndexClose=outIndexMax;
    outOtherOpen=outOtherMin; outOtherClose=outOtherMax;
  }
  servoIndex.attach(5);//index
  servoOther.attach(6);//other
  servoThumb.attach(9);//thumb
  //pinMode(pinCalib, INPUT);//A6
  //digitalWrite(pinCalib, HIGH);
  //pinMode(pinTBD, INPUT);//A5
  //digitalWrite(pinTBD, HIGH);
  pinMode(pinThumb, INPUT);//A4
  digitalWrite(pinThumb, HIGH);
  pinMode(pinOther, INPUT);//A3
  digitalWrite(pinOther, HIGH);
}

void loop() {
//==waiting for calibration==
  if(onSerial) Serial.println("======Waiting for Calibration======");
  while (1) {
    servoIndex.write(outIndexOpen);
    servoOther.write(outOtherOpen);
    servoThumb.write(outThumbOpen);
    if(onSerial) serialMonitor();
    delay(10);
    if (DigitalRead(pinCalib) == LOW) {
      calibration();
      break;
    }
  }
 //==control==
  position = positionMin;
  prePosition = positionMin;
  while (1) {
    if (DigitalRead(pinCalib) == LOW) swCount0 += 1;
    else swCount0 = 0;
    if (swCount0 == 10) {
      swCount0 = 0;
      calibration();
    }
```

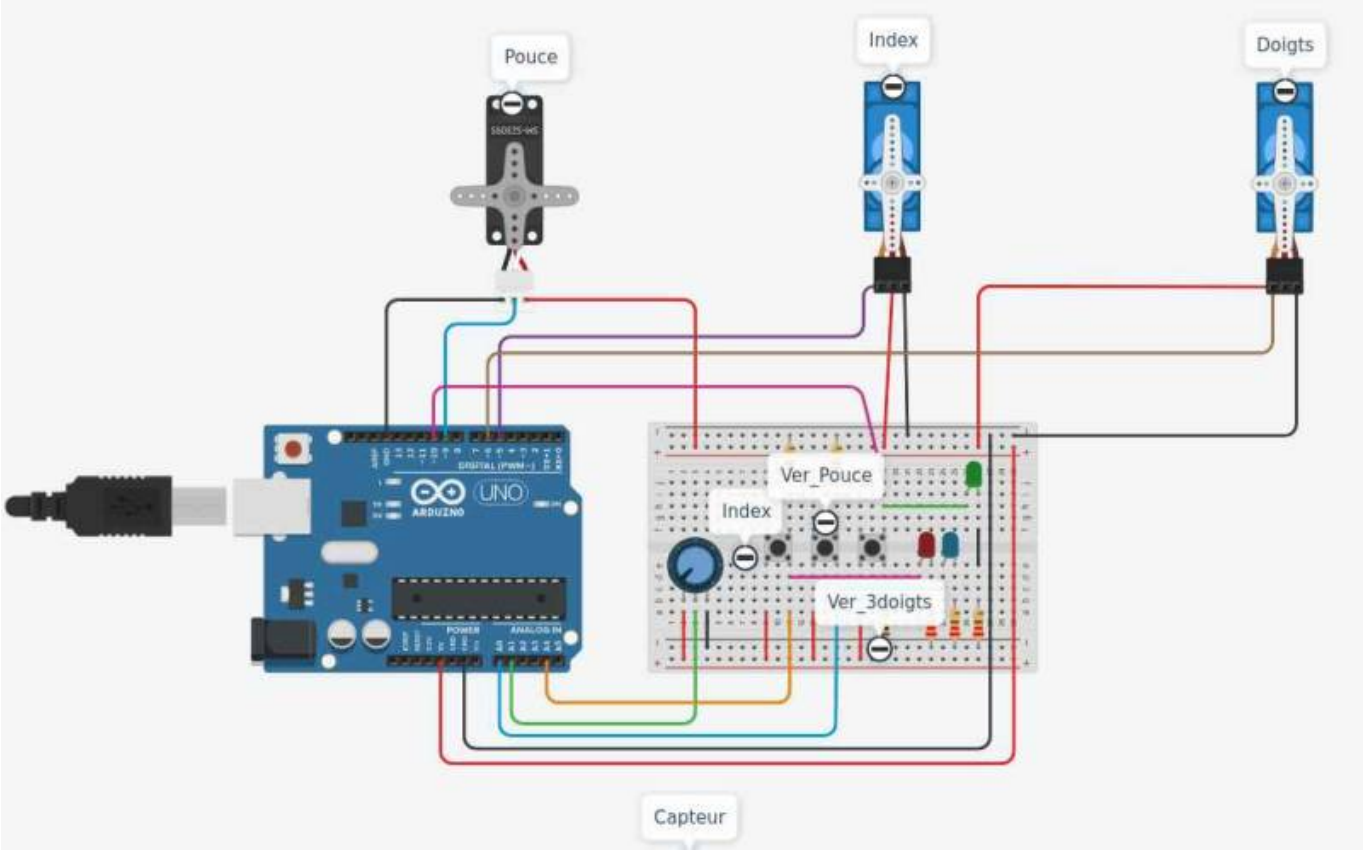
```
    if (digitalRead(pinThumb) == LOW) swCount2 += 1;
    else swCount2 = 0;
    if (swCount2 == 10) {
      swCount2 = 0;
      isThumbOpen = !isThumbOpen;
      while (digitalRead(pinThumb) == LOW) delay(1);
    }
    if (digitalRead(pinOther) == LOW) swCount3 += 1;//A3
    else swCount3 = 0;
    if (swCount3 == 10) {
      swCount3 = 0;
      isOtherLock = !isOtherLock;
      while (digitalRead(pinOther) == LOW) delay(1);
    }

    sensorValue = readSensor();
    delay(25);
    if(sensorValue<sensorMin) sensorValue=sensorMin;
    else if(sensorValue>sensorMax) sensorValue=sensorMax;
    sensorToPosition();

    outIndex = map(position, positionMin, positionMax, outIndexOpen,
outIndexClose);
    servoIndex.write(outIndex);
    if (!isOtherLock){
      outOther = map(position, positionMin, positionMax, outOtherOpen,
outOtherClose);
      servoOther.write(outOther);
    }
    if(isThumbOpen) servoThumb.write(outThumbOpen);
    else servoThumb.write(outThumbClose);
    if(onSerial) serialMonitor();
  }
}

/*
 * functions
 */
int readSensor() {
  int i, sval;
  for (i = 0; i < 10; i++) {
    sval += analogRead(pinSensor);
  }
  sval = sval/10;
  return sval;
}

void sensorToPosition(){
  int tmpVal = map(sensorValue, sensorMin, sensorMax, 100, 0);
  if(tmpVal<thSpeedReverse) speed=speedReverse;
  else if(tmpVal<thSpeedZero) speed=speedMin;
```

```
  else speed=map(tmpVal,40,100,speedMin,speedMax);
  position = prePosition + speed;
  if (position < positionMin) position = positionMin;
  if (position > positionMax) position = positionMax;
  prePosition = position;
}

void calibration() {
  outIndex=outIndexOpen;
  servoIndex.write(outIndexOpen);
  servoOther.write(outOtherClose);
  servoThumb.write(outThumbOpen);
  position=positionMin;
  prePosition=positionMin;

  delay(200);
  if(onSerial) Serial.println("======calibration start======");

  sensorMax = readSensor();
  sensorMin = sensorMax - 50;
  unsigned long time = millis();
  while ( millis() < time + 4000 ) {
    sensorValue = readSensor();
    delay(25);
    if ( sensorValue < sensorMin ) sensorMin = sensorValue;
    else if ( sensorValue > sensorMax )sensorMax = sensorValue;

    sensorToPosition();
    outIndex = map(position, positionMin, positionMax, outIndexOpen,
outIndexClose);
    servoIndex.write(outIndex);

    if(onSerial) serialMonitor();
  }
  if(onSerial)  Serial.println("======calibration finish======");
}

void serialMonitor(){
  Serial.print("Min="); Serial.print(sensorMin);
  Serial.print(",Max="); Serial.print(sensorMax);
  Serial.print(",sensor="); Serial.print(sensorValue);
  Serial.print(",speed="); Serial.print(speed);
  Serial.print(",position="); Serial.print(position);
  Serial.print(",outIndex="); Serial.print(outIndex);
  Serial.print(",isThumbOpen="); Serial.print(isThumbOpen);
  Serial.print(",isOtherLock="); Serial.println(isOtherLock);
}



boolean DigitalRead(const int pin) {
  if (analogRead(pin) > 512)return 1;
```

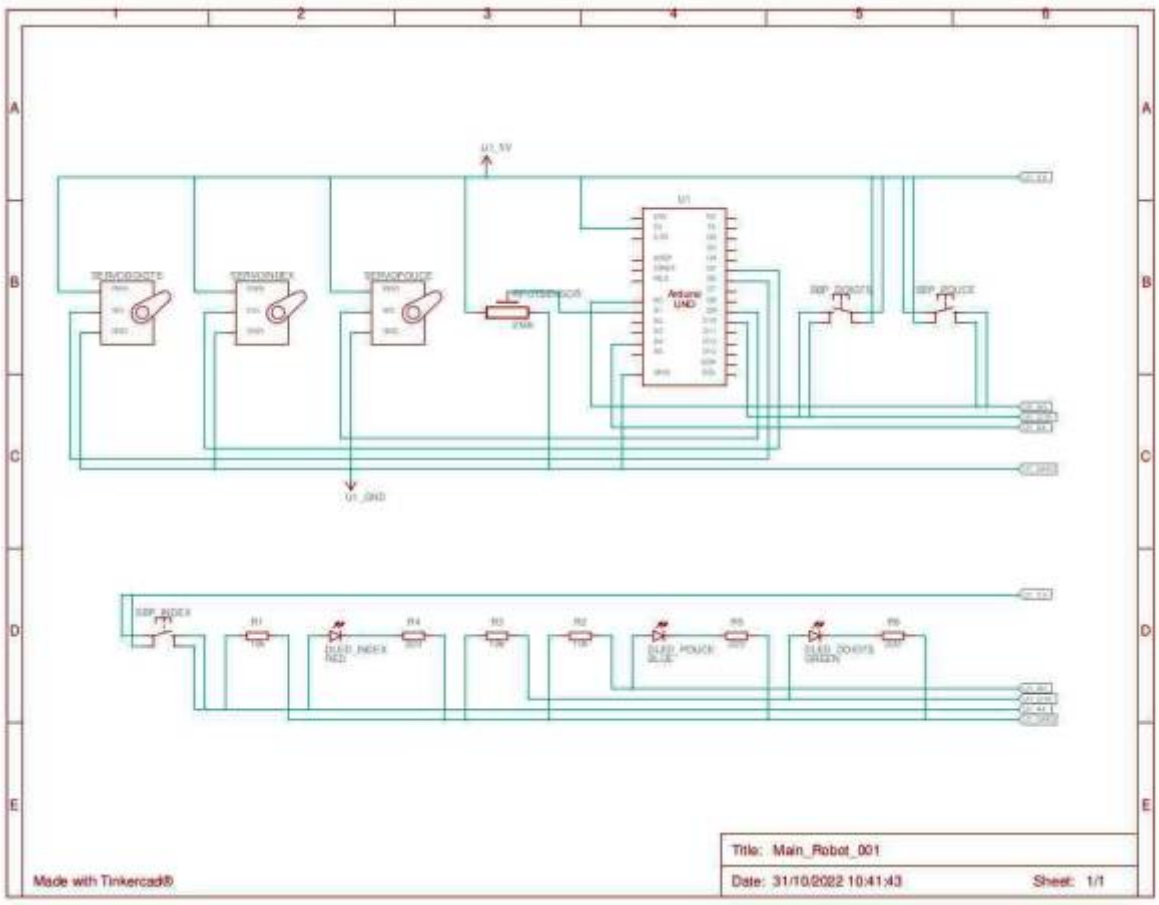
```
  else return 0;
}
```

Simulation main sur Tinkercad

Cablage Arduino Uno et servoMoteurs



Schema



Programme HackBerry_Test_tinkercad.ino version V1.0

Main_robot_0011.ino

```
#include <Servo.h>

//Pin
int pinButtonCalib; //start calibration
int pinButtonTBD; // No function implemented yet.
int pinButtonThumb; // open/close thumb
int pinButtonOther; //lock/unlock other three fingers
int pinServoIndex;
int pinServoOther;
int pinServoThumb;
int pinSensor; //sensor input

//Software
int sensorValue = 0;
int memPouce = 0;
int memIndex = 0;
int mem3Dgts = 0;

//Hardware
Servo servoIndex; //index
Servo servoOther; //3doigts
```

```
Servo servoThumb; //pouce

//Valeurs mini/max servos
const int outThumbMax = 170;//valeur max servo pouce
const int outIndexMax = 142;//valeur max servo index
const int outOtherMax = 96;//valeur max servo 3doigts

const int outThumbMin = 170-86;//valeur min servo pouce 170-86=84
const int outIndexMin = 140-117; //valeur min servo index 140-117= 23
const int outOtherMin = 95-55;//valeur min servo 3doigts 95-55 = 40

void setup() {
    Serial.begin(9600);

    // Pin Configuration
    pinButtonCalib = A4;// A6 sur Hackberry pour test Calibration --
validation index
    pinButtonTBD   = A5;// A7 sur Hackberry pour test
    pinButtonThumb = A0;// Verrouillage pouce -- validation pouce
    pinButtonOther = 10;// Verouillage 3 doigts -- validation  3doigts
    pinServoIndex  = 5;// Servo pour l'index
    pinServoOther  = 6;// Servo pour 3 doigts
    pinServoThumb  = 9;// Servo pour le pouce
    pinSensor      = A1; // Capteur de pression

    servoIndex.attach(pinServoIndex);//index
    servoOther.attach(pinServoOther);// 3 doigts
    servoThumb.attach(pinServoThumb);//Pouce
    pinMode(pinButtonCalib, INPUT_PULLUP);
    pinMode(pinButtonTBD,   INPUT_PULLUP);
    pinMode(pinButtonThumb, INPUT_PULLUP);
    pinMode(pinButtonOther, INPUT_PULLUP);
}

void loop() {
  sensorValue = 0;

  if ( digitalRead(pinButtonThumb) == HIGH ){
    memPouce = 1;
    memIndex = 0;
    mem3Dgts = 0;
  }
   if ( digitalRead(pinButtonCalib) == HIGH ){
    memPouce = 0;
    mem3Dgts = 0;
    memIndex = 1;
  }
  if ( digitalRead(pinButtonOther) == HIGH ){
    memPouce = 0;
    mem3Dgts = 1;
    memIndex = 0;
```

```
  }


  if (memPouce == 1){
    sensorValue = analogRead(pinSensor);
    delay(25);
    sensorValue = map(sensorValue, 0, 1023, outThumbMin, outThumbMax);
    Serial.print("sensor pouce= "); Serial.println(sensorValue);
    servoThumb.write(sensorValue);
    delay(15);
  }
  if (memIndex == 1){
    sensorValue = analogRead(pinSensor);
    delay(25);
    sensorValue = map(sensorValue, 0, 1023,outIndexMin, outIndexMax);
    Serial.print("sensor index= "); Serial.println(sensorValue);
    servoIndex.write(sensorValue);
    delay(15);
  }
  if (mem3Dgts == 1){
    sensorValue = analogRead(pinSensor);
    delay(25);
    sensorValue = map(sensorValue, 0, 1023,outOtherMin, outOtherMax);
    Serial.print("sensor doigts= "); Serial.println(sensorValue);
    servoOther.write(sensorValue);
    delay(15);
  }

}
```

Test main.ino

testmain.ino

```
//EMG sensor robotic hand controller
//This code is for controlling a robotic hand with
//an EMG sensor.
//
//© Au Robots 8.4.2017


//Necessary for controlling the servos
#include <Servo.h>

const int x = ///// This is the reference value and it
//will depend on your setup. You have to find it out
//yourself by looking at the serial monitor and finding
//a value between the maximum and minimum value.
```

```
//Naming the servos
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;
Servo servo5;
Servo servo6;

void setup()
{
//Starting the serial monitor
Serial.begin(9600);

//Configuring servo pins
servo2.attach(10); // pinky
servo3.attach(11); //ring
servo4.attach(3); // middle
servo5.attach(6); //index
servo6.attach(5); //thumb
}


void loop()
{
//Printing the EMG data
Serial.println(analogRead(5));

//If the EMG data is greater than x the hand closes
  if(analogRead(5) > x) {
    servo2.write(180);
    servo3.write(148);
    servo4.write(89);
    servo5.write(180);
    servo6.write(180);
  }

//If the EMG data is lower than x the hand opens
  else if (analogRead(5) < x) {
    servo2.write(38);
    servo3.write(10);
    servo4.write(0);
    servo5.write(16);
    servo6.write(16);
  }

//A delay to slow down the process
  delay(100);
}
```