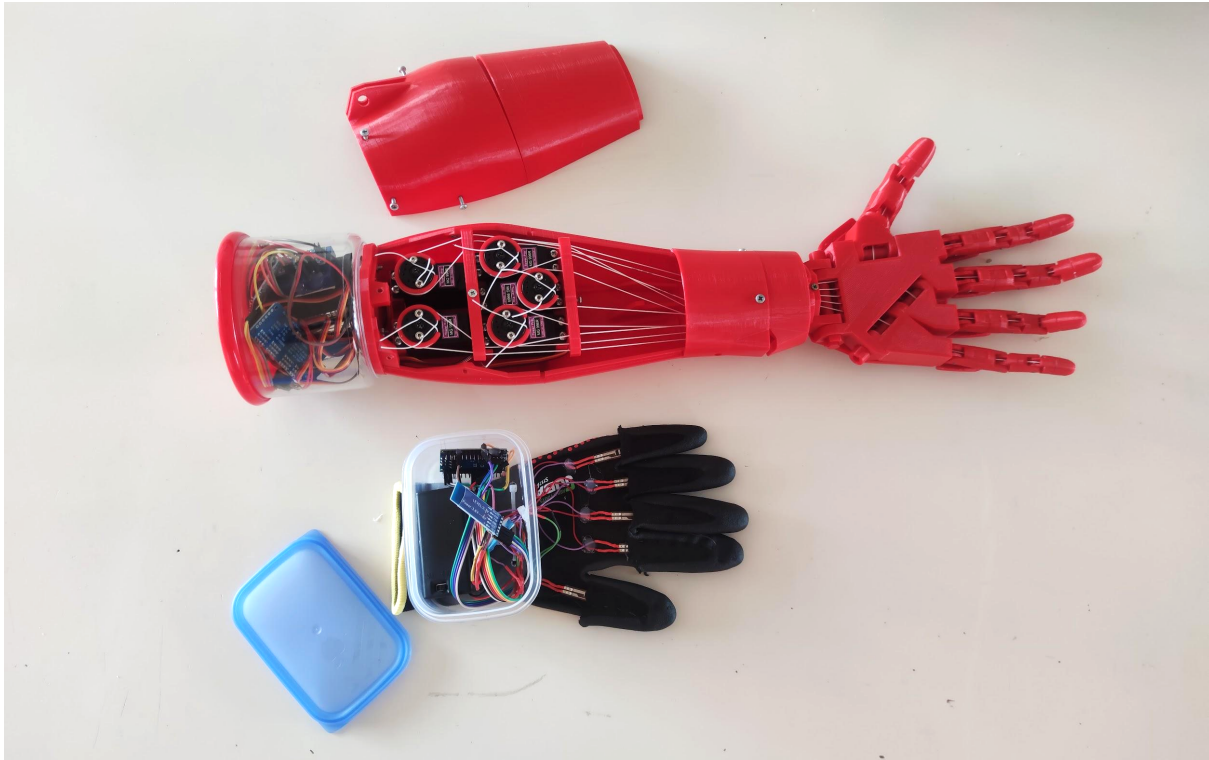


Brazo robótico inalámbrico con Arduino



Índice:

Introducción	2
Hardware	2
Montaje	3
Presupuesto	3
Conexionado	4
Software	5
Código	5
Configuración de los módulos bluetooth como maestro y esclavo.	5
Prueba de la mano, movimiento dedo a dedo:	6
Prueba de la mano, conexión por bluetooth	7
Prueba de la mano, envío de datos por bluetooth (emulando la función del guante)	9
Prueba del guante, visionado de los valores leídos por el sensor	10
Código final de la mano	12
Código final del guante	13
App móvil	14
Conclusión	15

Introducción

Este proyecto ha sido realizado por el alumnado de 4º ESO en la materia de Tecnología con el profesor Pablo González.

Consta de dos partes, el guante que viste el usuario y el brazo robótico que imita los movimientos del usuario.

Hardware

Los componentes necesarios para realizar este proyecto son los siguientes:

- 5 Servomotores de 180º MG996R
- 5 Sensores flex
- 5m de cuerda de bádminton
- 2 Arduino nano
- 2 Módulos bluetooth HC05
- Cables de conexionado
- Controlador servos PCA9685
- Portapilas 4xAA (6V)
- 2 Portapilas con interruptor 9V
- Guante de PVC
- 4 Pilas AA
- 2 Pilas 9V
- 2 Tuppers
- Filamento 3D Xataka PLA
- Hiilo, pegamento de cianoacrilato, tornillos, estaño y pegamento termoofusible.
- Piezas impresas en 3D:
 - Los archivos .stl se pueden descargar de la web oficial de Inmoov (<https://inmoov.fr/inmoov-stl-parts-viewer/?bodyparts=Left-Hand>):
 - * Mano izquierda
 - * Rotación muñeca
 - * Servo cama

Además, son necesarias las siguientes herramientas:

- Impresora 3D
- Destornillador
- Pelacables
- Soldador de estaño
- Tijeras
- Pistola de pegamento termofusible
- Aguja

Montaje

Es importante no mezclar las piezas impresas para no confundir las partes de cada dedo.

Los pasos a seguir se pueden visualizar en los siguientes enlaces:

- Parte 1: https://youtu.be/vgtJY_rJWzo
- Parte 2: <https://youtu.be/ZXy00Oxoixcg>

Los servir se colocan inicialmente a 70°, se atornillan las poleas en esa posición de manera que queden centradas. Luego se mueven los servos a 5°, se pasa la tanza por las poleas y se aprieta con los tornillos.

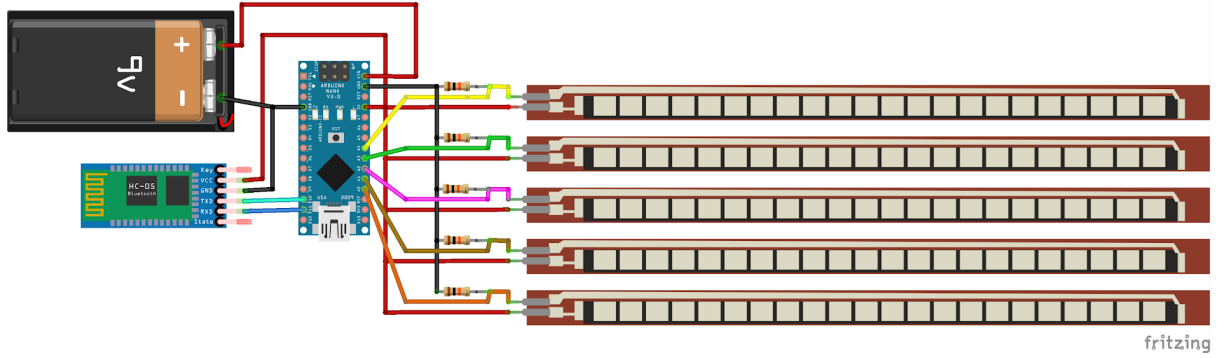
La mano estará estirada con 5° y encogida con 130°. Pero nunca debemos llegar a esos valores límite para no forzar el mecanismo.

Presupuesto

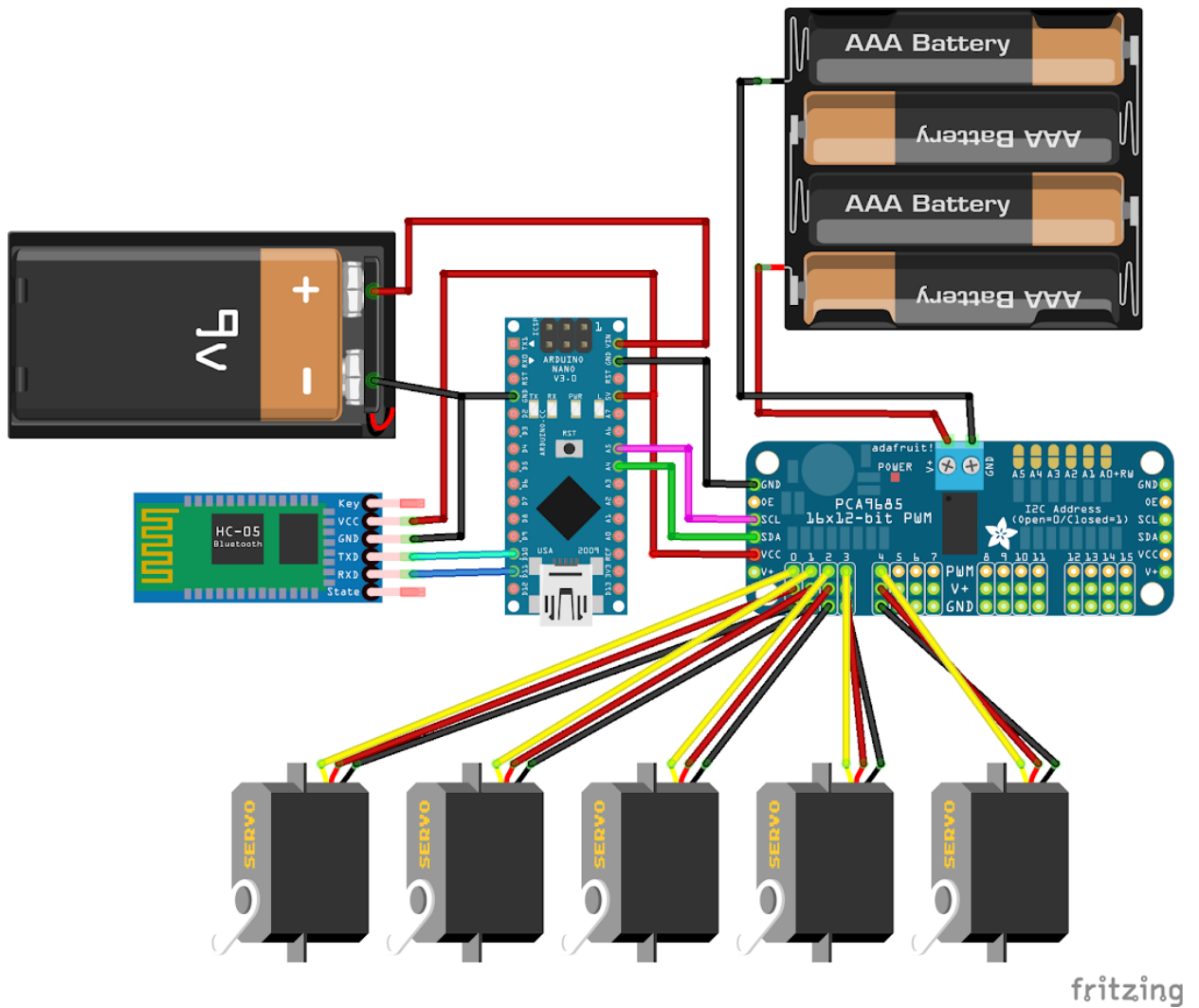
Componente	Precio	Enlace
- 5 Servomotores de 180° MG996R	25,35 €	https://es.aliexpress.com/item/33002954125.html
- 5 Sensores flex	41,70 €	https://es.aliexpress.com/item/33006987477.html
- 5m de sedal de pesca trenzado	6,99 €	https://pinguinofishing.es/w/mercado/producto/?nombre=MULTIFILAMENTO%20ASARI%20%20TRENZADO%20KAZUMA%20%20PE%208X&id=2521
- 2 Arduino nano	8,66 €	https://es.aliexpress.com/item/1005001780199515.html
- 2 Módulos bluetooth HC05	15,03 €	https://www.amazon.es/dp/B07J66PR6B?psc=1&ref=ppx_yo2ov_dt_b_product_details
- Cables de conexionado	6,98 €	https://www.amazon.es/dp/B0B1PD6SRH?psc=1&ref=ppx_yo2ov_dt_b_product_details
- Controlador servos PCA9685	4,34 €	https://es.aliexpress.com/item/4000302377081.html
- Portapilas 4xAA (6V)	1,54 €	https://www.amazon.es/dp/B07NDNKNLC?psc=1&ref=ppx_yo2ov_dt_b_product_details
- 2 Portapilas con interruptor 9V	9,69 €	https://www.amazon.es/dp/B094XSSYH2?ref=ppx_yo2ov_dt_b_product_details&th=1
- Guante de jardinería	4,69 €	https://amzn.eu/d/2AfDXFt
- 4 Pilas AA	1,99 €	https://supermercado.eroski.es/es/productdetail/18617423-pila-alcalina-lr06-aa-alkaline-pack-10-uds/
- 2 Pilas 9V	3,00 €	https://dmerka.com/?product=gadis-pila-super-alkalina-6lr61-9v-1-unidad-ifa-sabe
- 2 Toppers	3,00 €	
- Filamento 3D Sataca PLA	19,46 €	https://www.impresoras3d.com/producto/pla-850-sakata-3d-rojo/
otros (hilo, pegamento, etc)	7,00 €	
TOTAL	159,42 €	

Conexionado

- Guante:



- Mano:



Software

Para pasar el código a Arduino y poder comunicarnos con él es necesario disponer del software gratuito del IDE de Arduino (<https://www.arduino.cc/en/software>)
El plano del conexionado se ha realizado con fritzing (<https://fritzing.org>)

Código

Configuración de los módulos bluetooth como maestro y esclavo.

(*Importante: Para entrar en el modo AT, el botón del módulo debe estar pulsado a la hora de alimentarlo (Cargar el código en Arduino con el cable VCC desconectado y luego pulsar el botón y mantenerlo pulsado cuando se conecte)

```
//https://youtu.be/CIPu18Rbk6c

//Conectando un Bluetooth HC-05 a Arduino y vamos a los comandos AT para poder configurarlo.
//Se conecta RX al D11, TX al D10, Vcc a 5v dc y GND al arduino como siempre

/*PASOS PARA CARGAR EL PROGRAMA
* 1 Con el bluetooth apagado, cargamos el programa. VCC desconectada
* 2 Con el programa ya cargado presionamos el boton del modulo bluetooth y lo mantenemos pulsado
* 3 Manteniendo pulsado el botón, conectamos el VCC del bluetooth.
* 4 Vemos que parpadea el led más lento, quiere decir que está en modo configuración
*
* Si parpadea rápido es que está buscando módulos bluetooth.
*/

/*Comandos AT,

1 hay que ver que el panel de serie está en autoscroll, Ambos NL and CR y velocidad 38400 baudios
AT Para comprobar que entra en el modo de configuración
AT+NAME? Para conocer el nombre del dispositivo o cambiarselo. A VECES NO VA ESTE COMANDO
AT+NAME= nombre Para dar un nombre al bluetooth
AT+ADDR? Para conocer la mac del dispositivo (mano: 0022:12:000E1D)

AT+ROLE= Para configurarlo como maestro o esclavo. 0 ESCLAVO // 1 MAESTRO (normalmente esclavo)
AT+UART? Para la velocidad de comunicación
AT+PSWD? Para conocer el password. Normalmente 1234

AT+RESET Para resetear el bluetooth
AT+ORGL Para poner el módulo limpio de fábrica
AT+VERSION? Para saber la versión de firmware

Para emparejar maestros.

AT+CMODE=0 Para que empareje con una mac fija.
AT+BIND=0022,12,000E1D METEMOS LA MAC A EMPAREJAR. TODO MAYÚSUCLAS Y CAMBIANDO LOS : POR ,
AT+BIND? Dará la mac con la que está enlazado
*/

//Cuando cargamos el programa, el bluetooth debe estar descoenctado, apagado.
//después de cargarlo, lo conectamos

#include <SoftwareSerial.h> //Incluimos libreria de puerto serie. con esta podemos hacer pines digitales que sean de
//comunicación serie
SoftwareSerial BT(10, 11); // Estoy diciendo que mi puerto serie va a ir a los pines 10 y 11 como van en ese orden, quiere
//decir que el arduino, el pin 10 es RX y el arduino, el pin 11 es su TX. Lo que ocurre es que hay que invertirlos en el
//modulo bluetooth. Por eso el pin 10 de mi arduino va al TX del bluetooth y el pin 11 del arduino va al pin RX del bluetooth

void setup()
{ BT.begin(38400);
```

```

Serial.println("Listo"); // escribe Listo en el monitor para ver que hace algo
Serial.begin(38400); // Velocidad del puerto serie 38400
}

void loop()
{
  if (BT.available()) // Si hay disponible información desde el módulo
    Serial.write(BT.read()); // entonces lee bluetooth y lo envía al monitor serie
  if (Serial.available()) // Si hay info disponible desde el monitor serie
    BT.write(Serial.read()); // entonces lee lo del monitor serie y lo envía al bluetooth
}

```

En nuestro caso se debe pasar por consola lo siguiente:

MANO:

AT+ROLE= 0 (esclavo)

AT+UART= 38400 (puede ser una velocidad de comunicación mayor, pero tiene que coincidir con la del guante)

AT+NAME=Mano

AT+PSWD=1234 (puede ser otra contraseña, pero tiene que coincidir con la del guante)

AT+ADDR? (Se anota este valor, en nuestro caso 0022:12:000E1)

GUANTE:

AT+ROLE= 1 (maestro)

AT+UART= 38400 (puede ser una velocidad de comunicación mayor, pero tiene que coincidir con la de la mano))

AT+NAME=Guante

AT+PSWD=1234 (puede ser otra contraseña, pero tiene que coincidir con la de la mano)

AT+CMODE=0 (para que se empareje solo a una MAC)

AT+BIND=0022,12,000E1 (dirección MAC con la que tiene que enlazar, en nuestro caso 0022:12:000E1)

Prueba de la mano, movimiento dedo a dedo:

```

#include <Servo.h>
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

Adafruit_PWMServoDriver servos = Adafruit_PWMServoDriver(0x40);

unsigned int pos0=172; // ancho de pulso en cuentas para posición 0°
unsigned int pos180=565; // ancho de pulso en cuentas para la posición 180°
int estirados = 10;
int encogidos = 120;

int grados_1 = estirados;
int grados_2 = estirados;
int grados_3 = estirados;
int grados_4 = estirados;
int grados_5 = estirados;

void setup(){
  Serial.begin(9600); // Begin serial communication

  servos.begin();
  servos.setPWMPFreq(60); // Frecuencia PWM de 60Hz o T=16,66ms
}

```

```

Serial.println("Introduzca el dedo a mover (A-E) y los grados. Ejemplo: B110");
}

void setServo(uint8_t n_servo, int angulo) {
  int duty;
  duty=map(angulo,0,180,pos0, pos180);
  servos.setPWM(n_servo, 0, duty);
}

void loop(){

if (Serial.available() > 0) {

  switch (Serial.read()) {

    case 'A': grados_1 = Serial.parseInt();
    break;
    case 'B': grados_2 = Serial.parseInt();
    break;
    case 'C': grados_3 = Serial.parseInt();
    break;
    case 'D': grados_4 = Serial.parseInt();
    break;
    case 'E': grados_5 = Serial.parseInt();
    break;
  }

//-----

if (Serial.read() == '\n') {

  grados_1=grados_1;// Tiene en cuenta el último valor leído
  grados_2=grados_2;
  grados_3=grados_3;
  grados_4=grados_4;
  grados_5=grados_5;
}
}

setServo(0,grados_1);
Serial.print("Dedo 1 a: "); Serial.print(grados_1); Serial.print("\t");
setServo(1,grados_2);
Serial.print("Dedo 2 a: "); Serial.print(grados_2); Serial.print("\t");
setServo(2,grados_3);
Serial.print("Dedo 3 a: "); Serial.print(grados_3); Serial.print("\t");
setServo(3,grados_4);
Serial.print("Dedo 4 a: "); Serial.print(grados_4); Serial.print("\t");
setServo(4,grados_5);
Serial.println("Dedo 5 a: "); Serial.print(grados_5); Serial.print("\t");

  delay(100);
}

```

Prueba de la mano, conexión por bluetooth

```

/*
Codigo escrito por Pablo
Mano robótica
Programa a instalar en el arduino de la mano
*/
#include <Servo.h>
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include <SoftwareSerial.h> //Incluimos libreria de puerto serie. con esta podemos hacer pines digitales que sean de
//comunicación serie

SoftwareSerial BT(10, 11); // Estoy diciendo que mi puerto serie va a ir a los pines 10 y 11 como van en ese orden, quiere
//decir que el arduino, el pin 10 es RX y el arduino, el pin 11 es su TX. Lo que ocurre es que hay que invertirlos en el
//modulo bluetooth. Por eso el pin 10 de mi arduino va al TX del bluetooth y el pin 11 del arduino va al pin RX del bluetooth

Adafruit_PWMServoDriver servos = Adafruit_PWMServoDriver(0x40);

```

```

unsigned int pos0=172; // ancho de pulso en cuentas para pocicion 0°
unsigned int pos180=565; // ancho de pulso en cuentas para la pocicion 180°

int grados_1 = 70;
int grados_2 = 70;
int grados_3 = 70;
int grados_4 = 70;
int grados_5 = 70;

void setup() {
  // initialize serial:

  BT.begin(38400);
  //Serial.begin(9600);

  servos.begin();
  servos.setPWMPFreq(60); //Frecuecia PWM de 60Hz o T=16,66ms
}

void setServo(uint8_t n_servo, int angulo) {
  int duty;
  duty=map(angulo,0,180,pos0, pos180);
  servos.setPWM(n_servo, 0, duty);
}

void loop() {

  if (BT.available() > 0) {

    if (BT.read() == 'A') {

      grados_1 = BT.parseInt();
    }

    if (BT.read() == 'B') {
      grados_2 = BT.parseInt();
    }

    if (BT.read() == 'C') {
      grados_3 = BT.parseInt();
    }

    if (BT.read() == 'D') {
      grados_4 = BT.parseInt();
    }

    if (BT.read() == 'E') {
      grados_5 = BT.parseInt();
    }

  }

  //-----
  if (BT.read() == '\n') {

    if(grados_1>60 && grados_1<180){grados_1=grados_1;} // Tiene en cuenta el último valor leído siempre que sea válido
    if(grados_2>60 && grados_2<180){grados_2=grados_2;}
    if(grados_3>60 && grados_3<180){grados_3=grados_3;}
    if(grados_4>60 && grados_4<180){grados_4=grados_4;}
    if(grados_5>60 && grados_5<180){grados_5=grados_5;}

  }
}

delay(5);
BT.print("A");delay(20);BT.print(grados_1);delay(20);
BT.print("B");delay(20); BT.print(grados_2);delay(20);
BT.print("C");delay(20); BT.print(grados_3);delay(20);
BT.print("D");delay(20); BT.print(grados_4);delay(20);
BT.print("E");delay(20); BT.print(grados_5);delay(20);
BT.print("\n"); delay(20);

```



```

setServo(3,grados_1);//pulgar
setServo(0,grados_2);//indice
setServo(1,grados_3);//corazon
setServo(2,grados_4);//anular
setServo(4,grados_5);//meñique

delay(100);
}

```

Prueba de la mano, envío de datos por bluetooth (emulando la función del guante)

```

/*
Codigo escrito por Pablo
Mano robótica con sensores flex
Programa a ejecutar en el arduino del guante
*/

#include <SoftwareSerial.h>
#include <Wire.h>

SoftwareSerial BT(10, 11); // Estoy diciendo que mi puerto serie va a ir a los pines 10 y 11 como van en ese orden, quiere
//decir que el arduino, el pin 10 es RX y el arduino, el pin 11 es su TX. Lo que ocurre es que hay que invertirlos en el
//modulo bluetooth. Por eso el pin 10 de mi arduino va al TX del bluetooth y el pin 11 del arduino va al pin RX del bluetooth

int sensor1;
int sensor2;
int sensor3;
int sensor4;
int sensor5;

int grados1 = 70;
int grados2 = 70;
int grados3 = 70;
int grados4 = 70;
int grados5 = 70;

//-----

void setup() {

  BT.begin(38400);
  Serial.begin(9600);

  Serial.println("Introduzca los valores de los sensores (A-E) entre 65 y 170 . Ejemplo: A78B81C155D1105E95");
}

void loop() {

if (Serial.available() > 0){
if(Serial.read() == 'A'){grados1 = Serial.parseInt();}
if(Serial.read() == 'B'){grados2 = Serial.parseInt();}
if(Serial.read() == 'C'){grados3 = Serial.parseInt();}
if(Serial.read() == 'D'){grados4 = Serial.parseInt();}
if(Serial.read() == 'E'){grados5 = Serial.parseInt();}
//-----
if(Serial.read() == '\n') {
grados1=grados1;// Tiene en cuenta el último valor leído
grados2=grados2;
grados3=grados3;
grados4=grados4;
grados5=grados5;
}}

BT.print("A"); //BT.write??? o BT.print??

```

```

delay(20);
BT.print(grados1);
delay(20);

BT.write("B");
delay(20);
BT.write(grados2);
delay(20);

BT.print("C");
delay(20);
BT.print(grados3);
delay(20);

BT.print("D");
delay(20);
BT.print(grados4);
delay(20);

BT.print("E");
delay(20);
BT.print(grados5);
delay(20);

BT.print("\n");
delay(20);

//-----

if (BT.available()) {
  Serial.write(BT.read()); //muestra por pantalla los valores de los grados que envia la MANO
}
}

```

Prueba del guante, visionado de los valores leídos por el sensor

```

/*
Codigo escrito por Pablo
Mano robótica con sensores flex
Programa a ejecutar en el arduino del guante
*/

#include <SoftwareSerial.h>
#include <Wire.h>

SoftwareSerial BT(10, 11); // Estoy diciendo que mi puerto serie va a ir a los pines 10 y 11 como van en ese orden, quiere
//decir que el arduino, el pin 10 es RX y el arduino, el pin 11 es su TX. Lo que ocurre es que hay que invertirlos en el
//modulo bluetooth. Por eso el pin 10 de mi arduino va al TX del bluetooth y el pin 11 del arduino va al pin RX del bluetooth

int flex1 = A0; //el primer sensor conectado a la entrada analógica A=
int flex2 = A1;
int flex3 = A2;
int flex4 = A3;
int flex5 = A4;

int sensor1; //declaro la variable sensor para guardar el valor leído por el sensor (700 a 900) (0-1023 normalmente en otros sensores)
int sensor2;
int sensor3;
int sensor4;
int sensor5;

int grados1 = 70; //posición de seguridad para los servos
int grados2 = 70;
int grados3 = 70;
int grados4 = 70;
int grados5 = 70;

void setup() {

```

```

Serial.begin(38400); //inicia el bluetooth a 38400 baudios (parametro que ya se ha ajustado previamente en el BT al cambiar AT*UART=38400,0,0)
}

void loop() {

  // Visionado de los valores de los sensores _____
  Serial.print("A"); //Enviamos la letra A para indicar que luego pasaremos el valor del primer dedo. BT.write() no se usa porque modifica el valor de los
  grados por cambiar el formato
  delay(20); //esperamos unos milisegundos para no saturar el buffer del bluetooth
  Serial.print(sensor1); //enviamos el valor de grados que se ha obtenido para que la mano lo lea
  delay(20); //esperamos unos milisegundos para no saturar el buffer del bluetooth

  Serial.print("B");
  delay(20);
  Serial.print(sensor2);
  delay(20);

  Serial.print("C");
  delay(20);
  Serial.print(sensor3);
  delay(20);

  Serial.print("D");
  delay(20);
  Serial.print(sensor4);
  delay(20);

  Serial.print("E");
  delay(20);
  Serial.print(sensor5);
  delay(20);

  Serial.print("\n"); //indicamos que ya hemos pasado todos los valores leidos
  delay(20);
}

```

Valores leídos de los sensores con la mano estirada:

```

Pulgar229Índice239Corazón224Anular223Meñique218
Pulgar229Índice240Corazón225Anular223Meñique218
Pulgar229Índice240Corazón226Anular224Meñique219
Pulgar229Índice240Corazón226Anular224Meñique219
Pulgar229Índice240Corazón226Anular225Meñique219
Pulgar229Índice237Corazón223Anular226Meñique215
Pulgar229Índice238Corazón223Anular225Meñique216
Pulgar228Índice240Corazón220Anular225Meñique216
Pulgar226Índice240Corazón220Anular223Meñique217
Pulgar223Índice239Corazón221Anular223Meñique218
Pulgar227Índice240Corazón221Anular223Meñique218
Pulgar227Índice240Corazón218Anular221Meñique218
Pulgar223Índice239Corazón218Anular221Meñique218
Pulgar223Índice239Corazón218Anular221Meñique219
Pulgar225Índice239Corazón217Anular220Meñique218
Pulgar225Índice239Corazón216Anular215Meñique218

```

Valores leídos de los sensores con la mano encogida:

```

Pulgar100Índice104Corazón98Anular104Meñique98
Pulgar100Índice105Corazón98Anular104Meñique99
Pulgar101Índice105Corazón98Anular104Meñique99
Pulgar101Índice105Corazón99Anular104Meñique98
Pulgar99Índice108Corazón96Anular105Meñique100
Pulgar98Índice106Corazón100Anular100Meñique101
Pulgar102Índice104Corazón102Anular104Meñique96
Pulgar103Índice100Corazón99Anular106Meñique96
Pulgar99Índice109Corazón96Anular103Meñique101
Pulgar100Índice103Corazón102Anular100Meñique101
Pulgar100Índice107Corazón101Anular100Meñique101
Pulgar100Índice105Corazón101Anular100Meñique97
Pulgar104Índice101Corazón103Anular105Meñique96
Pulgar107Índice105Corazón96Anular107Meñique97
Pulgar99Índice111Corazón97Anular103Meñique102
Pulgar102Índice105Corazón102Anular101Meñique101

```

Se obtienen entonces los siguientes valores máximos y mínimos de cada sensor después de varias comprobaciones:

Dedo	Extendido	Encogido
pulgar	245	80
índice	246	84
corazón	250	64
anular	263	90
meñique	240	85

Código final de la mano

```

/*
Mano robótica
Programa a instalar en el arduino de la mano
*/

#include <Servo.h>
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>
#include <SoftwareSerial.h> //Incluimos libreria de puerto serie. con esta podemos hacer pines digitales que sean de
//comunicación serie

SoftwareSerial BT(10, 11); // Estoy diciendo que mi puerto serie va a ir a los pines 10 y 11 como van en ese orden, quiere
//decir que el arduino, el pin 10 es RX y el arduino, el pin 11 es su TX. Lo que ocurre es que hay que invertirlos en el
//modulo bluetooth. Por eso el pin 10 de mi arduino va al TX del bluetooth y el pin 11 del arduino va al pin RX del bluetooth

Adafruit_PWMServoDriver servos = Adafruit_PWMServoDriver(0x40); //permite controlar los servos a través del módulo de conexión de 16 pines

unsigned int pos0=172; // ancho de pulso en cuentas para posición 0°
unsigned int pos180=565; // ancho de pulso en cuentas para la posición 180°

int dedo_extendido = 10; //valor en grados donde el dedo está estirado
int dedo_encogido = 120;

int grados_1 = dedo_extendido; //posición neutra de los servos
int grados_2 = dedo_extendido;
int grados_3 = dedo_extendido;
int grados_4 = dedo_extendido;
int grados_5 = dedo_extendido;

int grados1 = dedo_extendido; //posición neutra de los servos
int grados2 = dedo_extendido;
int grados3 = dedo_extendido;
int grados4 = dedo_extendido;
int grados5 = dedo_extendido;

void setup() {
  // initialize serial:

  BT.begin(38400); //inicia el bluetooth a 38400 baudios (parametro que ya se ha ajustado previamente en el BT al cambiar AT*UART=38400,0,0)

  servos.begin();
  servos.setPWMPFreq(60); //Frecuencia PWM de 60Hz o T=16,66ms
}

void setServo(uint8_t n_servo, int angulo) { //adapta los pulsos a angulos

```

```

int duty;
duty=map(angulo,0,180,pos0, pos180);
servos.setPWM(n_servo, 0, duty);
}

void loop() {

if (BT.available() > 0) {

if (BT.read() == 'A') { //si lee una letra A...
  grados_1 = BT.parseInt(); //el siguiente valor entero es el ángulo del primer dedo

if (BT.read() == 'B') {
  grados_2 = BT.parseInt();}

if (BT.read() == 'C') {
  grados_3 = BT.parseInt();}

if (BT.read() == 'D') {
  grados_4 = BT.parseInt();}

if (BT.read() == 'E') {
  grados_5 = BT.parseInt();}

if (BT.read() == '\n') { //cuando el envío de la cadena termina...
  if(grados_1>dedo_extendido && grados_1<dedo_encogido){grados1=grados_1;} // Tiene en cuenta el último valor leído siempre que sea válido
  if(grados_2>dedo_extendido && grados_2<dedo_encogido){grados2=grados_2;}
  if(grados_3>dedo_extendido && grados_3<dedo_encogido){grados3=grados_3;}
  if(grados_4>dedo_extendido && grados_4<dedo_encogido){grados4=grados_4;}
  if(grados_5>dedo_extendido && grados_5<dedo_encogido){grados5=grados_5;}
}

setServo(3,grados1);//mueve el dedo pulgar
setServo(0,grados2);//índice
setServo(1,grados3);//corazon
setServo(2,grados4);//anular
setServo(4,grados5);//meñique
}

```

Código final del guante

```

/*
Codigo escrito por Pablo
Mano robótica con sensores flex
Programa a ejecutar en el arduino del guante
*/

#include <SoftwareSerial.h>
#include <Wire.h>

SoftwareSerial BT(10, 11); // Estoy diciendo que mi puerto serie va a ir a los pines 10 y 11 como van en ese orden, quiere
//decir que el arduino, el pin 10 es RX y el arduino, el pin 11 es su TX. Lo que ocurre es que hay que invertirlos en el
//modulo bluetooth. Por eso el pin 10 de mi arduino va al TX del bluetooth y el pin 11 del arduino va al pin RX del bluetooth

//comprobar qué sensor está conectado a cada entrada
int flex4 = A0; // anular
int flex1 = A1; // pulgar
int flex3 = A2; // corazón
int flex5 = A3; // meñique
int flex2 = A4; // índice

int dedo_extendido = 10; //valor en grados donde el dedo está estirado
int dedo_encogido = 120;

int sensor1; //declaro la variable sensor para guardar el valor leído por el sensor (700 a 900) (0-1023 normalmente en otros sensores)
int sensor2;
int sensor3;
int sensor4;
int sensor5;

```

```

int grados1 = dedo_extendido;//posición de seguridad para los servos
int grados2 = dedo_extendido;
int grados3 = dedo_extendido;
int grados4 = dedo_extendido;
int grados5 = dedo_extendido;

int sensor_extendido = 240; // valor máximo leído en la prueba de sensores con la mano estirada
int sensor_encogido = 90; // valor míno leído en la prueba de sensores con la mano encogida

void setup() {

  BT.begin(38400);//inicia el bluetooth a 38400 baudios (parámetro que ya se ha ajustado previamente en el BT al cambiar AT*UART=38400,0,0)

}

void loop() {

sensor1 = analogRead(flex1); //lee el sensor conectado a la entrada...
grados1 = map(sensor1, sensor_encogido, sensor_extendido, dedo_encogido, dedo_extendido);// (sensor1, 700, 900, 65, 175) y reescala su valor entre los posibles ángulos que pueden tomar los servos (65 estirada - 175 encogida)

sensor2 = analogRead(flex2);
grados2 = map(sensor2, sensor_encogido, sensor_extendido, dedo_encogido, dedo_extendido);// (sensor2, 90, 250, 120, 10)

sensor3 = analogRead(flex3);
grados3 = map(sensor3, sensor_encogido, sensor_extendido, dedo_encogido, dedo_extendido);

sensor4 = analogRead(flex4);
grados4 = map(sensor4, sensor_encogido, sensor_extendido, dedo_encogido, dedo_extendido);

sensor5 = analogRead(flex5);
grados5 = map(sensor5, sensor_encogido, sensor_extendido, dedo_encogido, dedo_extendido);

// _____

BT.print('A'); //Enviamos la letra A para indicar que luego pasaremos el valor del primer dedo. BT.write() no se usa porque modifica el valor de los grados por cambiar el formato
//delay(5); //esperamos unos milisegundos para no saturar el buffer del bluetooth
BT.print(grados1); //enviamos el valor de grados que se ha obtenido para que la mano lo lea
//delay(5); //esperamos unos milisegundos para no saturar el buffer del bluetooth

BT.print('B');
BT.print(grados2);

BT.print('C');
BT.print(grados3);

BT.print('D');
BT.print(grados4);

BT.print('E');
BT.print(grados5);

BT.print('\n'); //indicamos que ya hemos pasado todos los valores leídos
delay(10);

}

```

App móvil

Para probar la conexión bluetooth de la mano y el visionado de los valores leídos por ella se ha diseñado una aplicación móvil para android en MIT app inventor

(<https://gallery.appinventor.mit.edu/?galleryid=c4b3d25b-29b3-452b-b8e2-8abfc885193e>)



Se puede descargar el archivo .apk para instalar la aplicación en un smartphone android en el siguiente enlace:

- https://drive.google.com/file/d/1tnGnlcagPdsVlzCg8JsruBQrAxC03p_r/view?usp=share_link



También se ha creado otra aplicación, sin visionado, para que el usuario también pueda mover cada dedo de la mano sin tener que ponerse el guante.

-

Conclusión

En la elaboración de este proyecto se han encontrado las siguientes dificultades:

- Las cuerdas de bádmiton (https://www.decathlon.es/es/p/cuerda-de-badminton-bst-930-roja/_/R-p-307308?mc=854402&c=BLANCO) que manejan los dedos se salen de las poleas:
 - Se ha procedido a tensarlas mucho más y tampoco funcionó. Se procedió a cambiarlas por sedal trenzado de pesca.
- El guante de jardinería utilizado en un primer momento era demasiado flexible por lo que al retirarlo de la mano, el sensor se salía de las costuras

- Se ha cambiado por un guante de PVC mucho más rígido y menos ajustado, pero tampoco funcionó. Finalmente se optó por cubrir los sensores con los dedos cortados del guante derecho.
- Los módulos bluetooth no cambiaban a modo maestro
 - Se han comprado otros dos módulos nuevos puesto que los primeros debían estar estropeados.

Hemos trabajado como un grupo unido hasta el final y eso ha dado sus frutos.

Este proyecto podría tener diversas aplicaciones. Entre ellas podríamos destacar la de realizar una prótesis. Para ello solo deberíamos sustituir los sensores flex por sensores que detectan los impulsos nerviosos del cerebro, como es el caso del brazalete de control de gestos de Myo (EMG MYO). Esta opción no hemos podido desarrollarla debido al alto precio de dichos sensores.

