

# Mano Biónica

## 1.- Prueba del servo:

- 1.1.- Breve introducción (Arduino y Servo).
- 1.2.- Conexión del Arduino a la placa protoboard.
- 1.3.- Interconexión entre Servo y Arduino.
- 1.4.- Alimentación de Arduino con el USB y Servo con una fuente de 6V.
- 1.5.- ¡Tierra común!
- 1.6.- Carga y ejecución del programa para probar el Servo.

## 2.- Montaje de la mano:

- 2.1.- Breve introducción a la impresión 3D.
- 2.2.- Montaje de dedos, cortado de cuerdas y paso por dedos.
- 2.3.- Unión dedos con palma y mano con antebrazo.
- 2.4.- Colocación del servo en caja.
- 2.5.- Calibrado del servo (utilizando el programa del punto 1).
- 2.6.- Atado de cuerdas al servo.

## 3.- Soldado del circuito EMG:

- 3.1.- Breve introducción EMG.
- 3.2.- Explicación del circuito y sus partes.
- 3.3.- Componentes electrónicos y soldado.
- 3.4.- Alimentación INA128P mediante pilas.
- 3.5.- Ejemplos de esquemas realizados.
- 3.6.- Prueba de la señal EMG usando un osciloscopio.

## 4.- Sistema completo:

- 4.1.- Carga y ejecución del programa para servo y EMG juntos.
- 4.2.- Probar el sistema completo.
- 4.3.- ¡Gozo y algarabía!

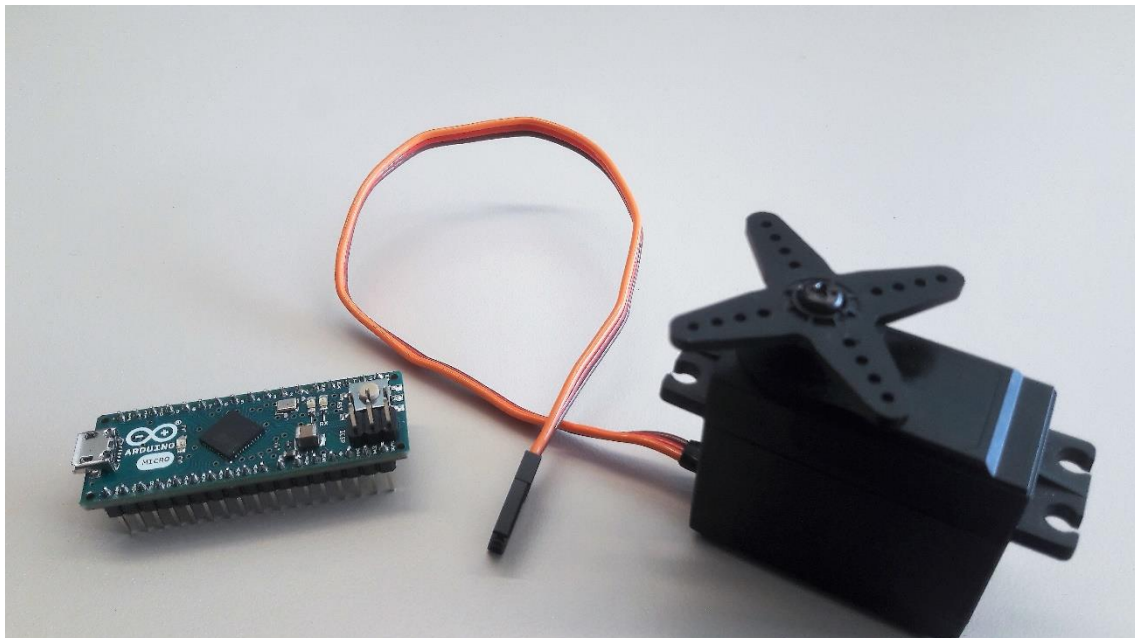
\* (2 y 3 son intercambiables en orden).

# 1.- Prueba del servo:

## 1.1.- Breve introducción (Arduino y Servo).

**Arduino** es una compañía que se dedica a realizar placas para desarrollo hardware. Los dos componentes principales de las placas son un **microcontrolador**, y una serie de puertos de entradas y salidas.

Un **servo** (servomotor), es un **motor** que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación y mantenerse estable en ella. Para ello dispone de un circuito de control electrónico que controla la dirección y la posición del motor.

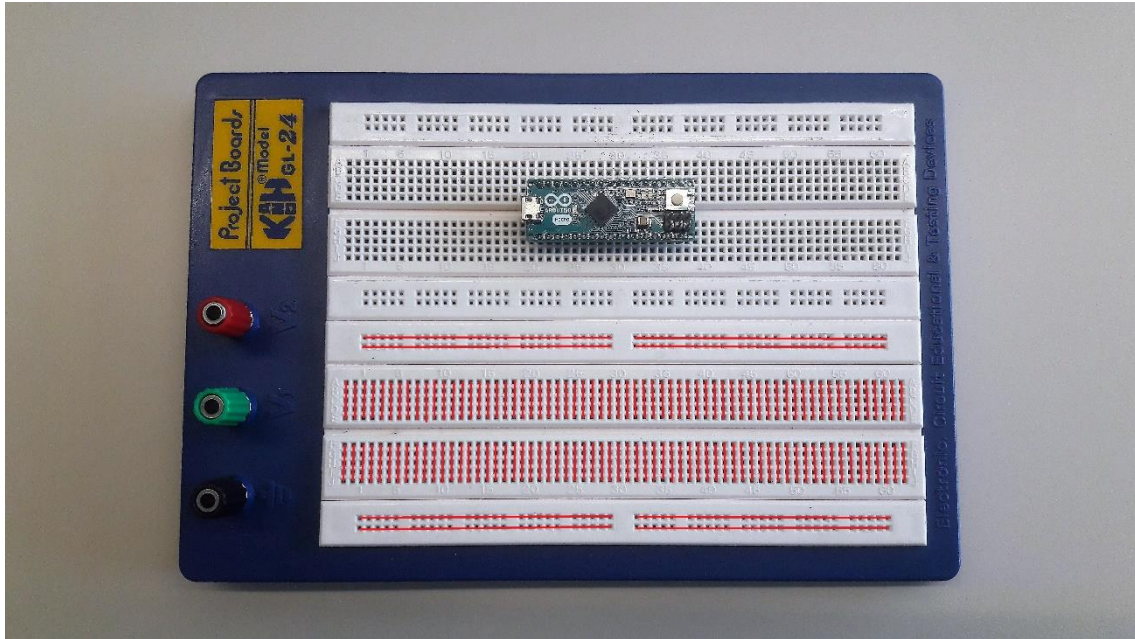


*Imagen Arduino MICRO y servomotor.*

En este proyecto vamos a utilizar una placa Arduino modelo MICRO a la que vamos a conectar en uno de sus puertos un servomotor. Después programaremos el microcontrolador para poder manejar el servo.

## 1.2.- Conexión del Arduino a la placa protoboard.

Una placa **protoboard** es una placa de pruebas para el prototipado de circuitos electrónicos. Está formada por un conjunto de orificios conectados eléctricamente de manera interna en posición horizontal y vertical.



*Imagen placa protoboard y Arduino MICRO.*

Para que cada PIN se encuentre aislado de los demás y podamos realizar las diferentes conexiones de nuestro circuito hay que colocar la placa Arduino de modo similar al de la imagen. En la parte inferior podemos observar en color rojo unas líneas que muestran el conexionado interno de la placa.

## 1.3.- Interconexión entre Servo y Arduino.

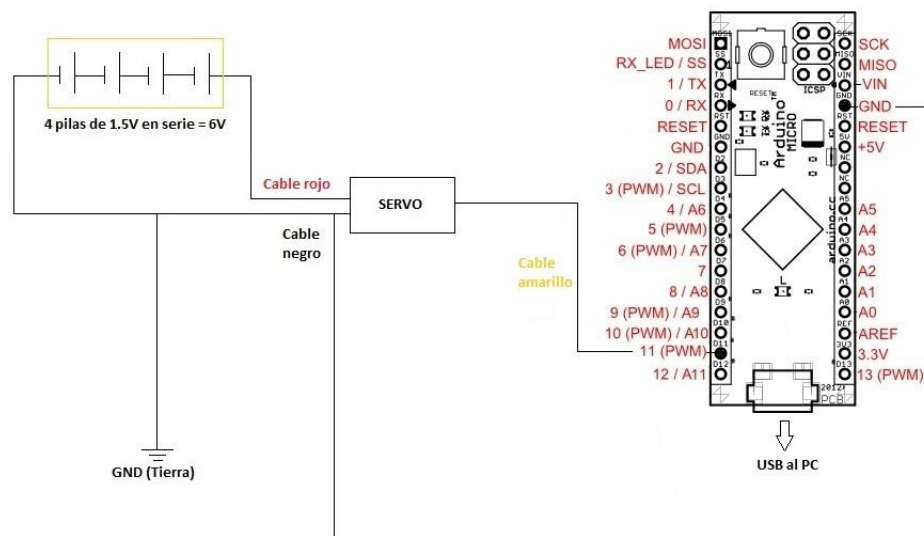
Para la conexión vamos a utilizar el **puerto 11** (PWM) de la placa Arduino. Lo conectamos al puerto de control del servo que se corresponde con el cable amarillo.

#### 1.4.- Alimentación de Arduino con el USB y Servo con una fuente de 6V.

Vamos a alimentar la placa Arduino mediante un cable USB con un PC. Para el Servo necesitaremos una fuente de alimentación externa de 6V, para ello vamos a utilizar 4 pilas de 1.5V conectadas en serie.

#### 1.5.- ¡Tierra común!

Para evitar posibles interferencias entre los diferentes componentes del circuito vamos a conectar el negativo de todos los elementos a un mismo punto denominado masa común o tierra.



*Esquema del circuito para prueba del Servo.*

## 1.6.- Carga y ejecución del programa para probar el Servo.

Tenemos que realizar un programa que nos sirva para manejar el servomotor y cargarlo en el Arduino. El lenguaje de programación que usa Arduino es una adaptación de C++ que incluye varias librerías que facilitan la programación de las entradas y salidas de la placa.

### Ejemplo de programa para controlar el Servo:

```
/*
 * Controla el servo
 * "a" en la consola para abrir la mano
 * "c" en la consola para cerrar la mano
 */

#include <Servo.h>

int servo_pin = 11;
boolean hand_opened = true;

int opened_angle = 45, closed_angle = 0;
Servo servo;
int servo_timer = 0;
int timer_threshold = 500;

void setup()
{
  Serial.begin(9600);
  servo.attach(servo_pin);
  servo.write(opened_angle);
}

void loop()
{
  if (Serial.available() > 0) {
    int inByte = Serial.read();
    switch (inByte) {
      case 'a':
        if(!hand_opened) {
          for(int pos = closed_angle * 2; pos < opened_angle * 2; pos+=2)
          {
            servo.write(pos);
            delay(2);
          }
          hand_opened=true;
        }
        servo_timer = 0;
        break;
      case 'c':
        if(hand_opened){
          for(int pos = opened_angle * 2; pos > closed_angle * 2; pos-=2)
          {
            servo.write(pos);
            delay(2);
          }
          hand_opened=false;
        }
        servo_timer = 0;
        break;
      default:
        delay(10);
    }

    if (servo_timer < timer_threshold)
      servo_timer++;

    delay(1);
  }
}
```

## 2.- Montaje de la Mano:

### 2.1.- Breve introducción a la impresión 3D.

La impresión 3D consiste en crear objetos físicos en base a un modelo digital. Existen varias tecnologías, la más utilizada es la **FDM** (Modelado por Deposición Fundida) que consiste en la superposición de capas de material fundido a gran temperatura hasta formar el objeto deseado. Hay multitud de materiales para la impresión en 3D, los más comunes y que mejor funcionan con la tecnología FDM son los **termoplásticos** como por ejemplo el PLA o el ABS.

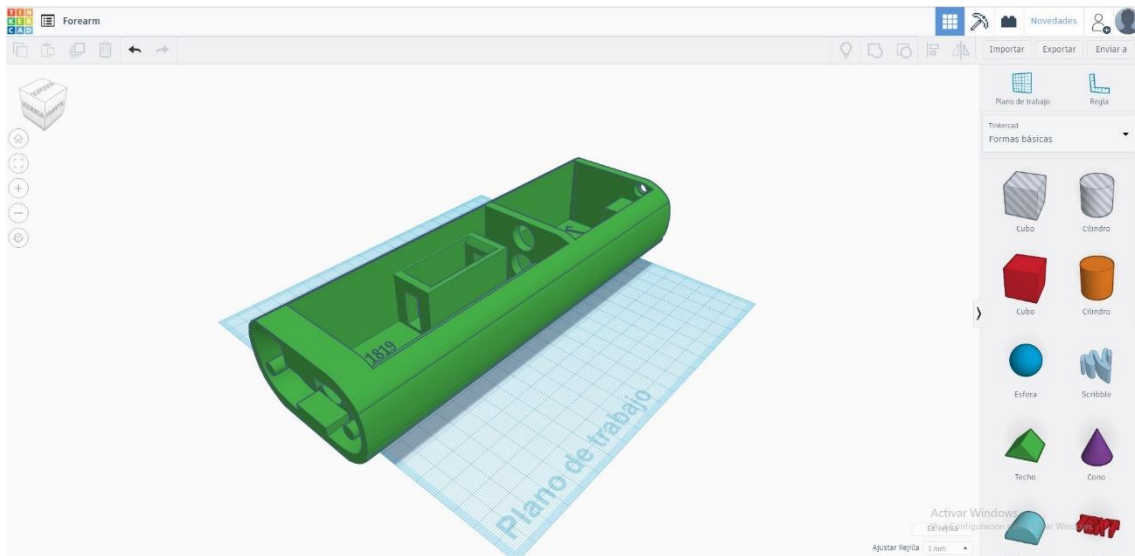


*Pieza impresa al 1%, al 50% y al 100%.*



*Imagen de la pieza durante la impresión.*

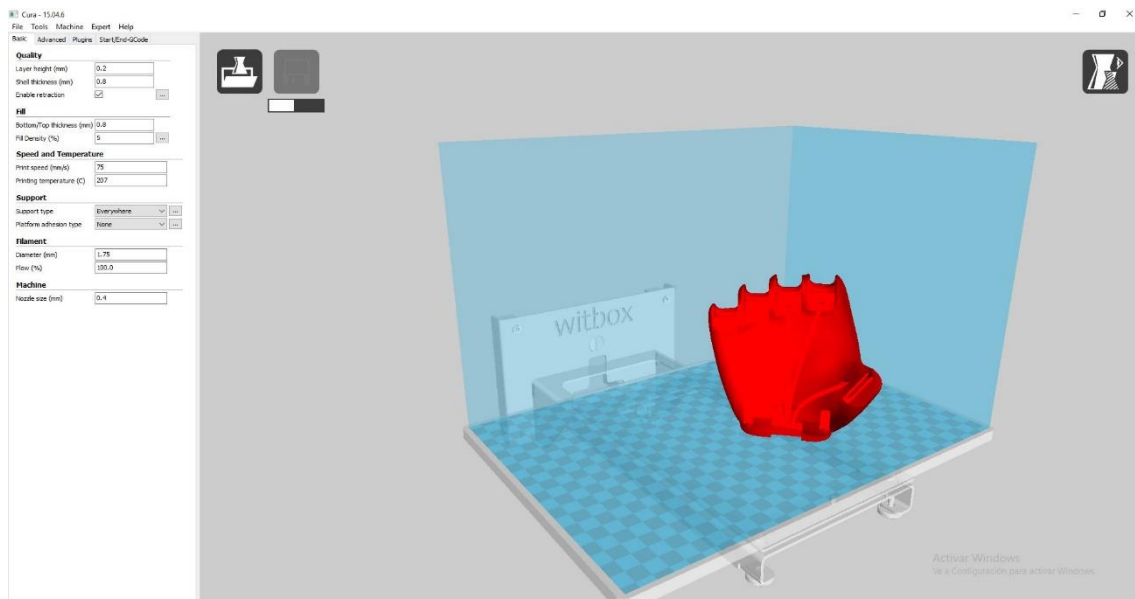
Antes de comenzar a imprimir necesitamos un modelo digital del objeto. El modelo lo crearemos mediante algún programa de diseño 3D, existen multitud, incluso gratuitos. Algunos de los programas más utilizados son: TinkerCAD, FreeCAD, SketchUp o AutoCAD. Otra opción es escanear el objeto si disponemos de un escáner 3D.



*Captura de pantalla TinkerCAD.*

El último paso antes de imprimir es “traducir” nuestro modelo digital a un lenguaje que entienda la impresora, y para ello necesitamos de otro software denominado **laminador**.

El laminador nos va a crear un fichero G-code en el que están configuradas las coordenadas de impresión y diferentes opciones de configuración. Entre estas opciones se selecciona el modelo de impresora y su área de impresión, el grosor del borde de la pieza, la densidad del relleno interior, la velocidad y la temperatura de impresión, etc. Los más utilizados actualmente son Cura, Slic3r y KISSlicer.

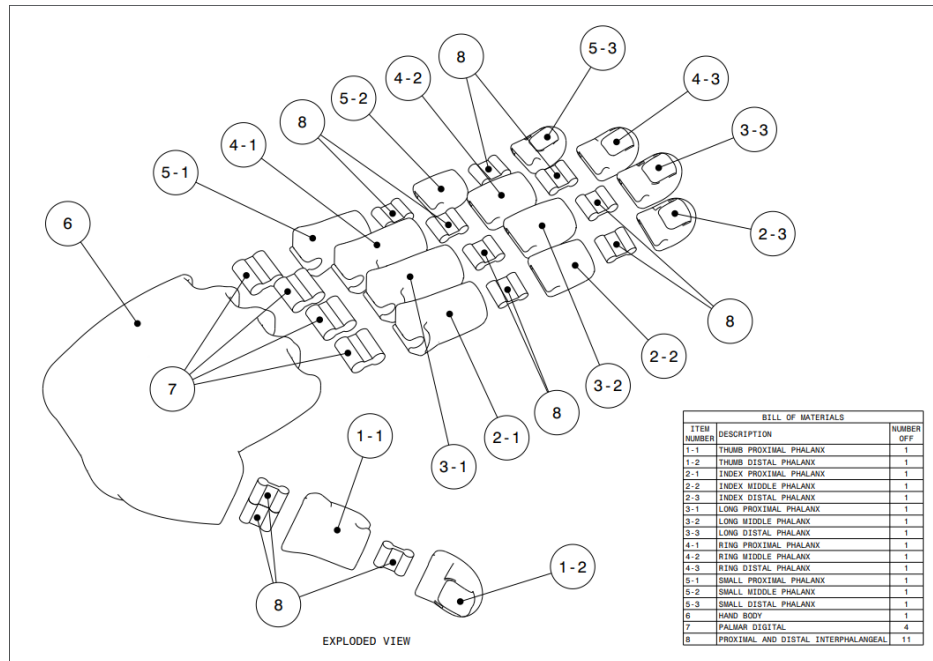


*Captura de pantalla Cura.*



## 2.2.- Montaje de dedos, cortado de cuerdas y paso por dedos.

Para montar los dedos vamos a unirlos mediante las articulaciones impresas con material flexible.



*Esquema de montaje de la mano.*

Cortaremos cinco cuerdas muy largas (un metro aproximadamente) para poder pasarlas por los dedos, la palma de la mano y atarlas al servomotor con facilidad. Primero utilizamos un cable fuerte, como por ejemplo hilo de pesca doblado a la mitad desde uno de los orificios de la punta del dedo y lo pasamos hacia abajo por el resto de los orificios. Ahora vamos a utilizar este cable para tirar de la cuerda y después repetir la operación por el otro orificio de la punta del dedo.

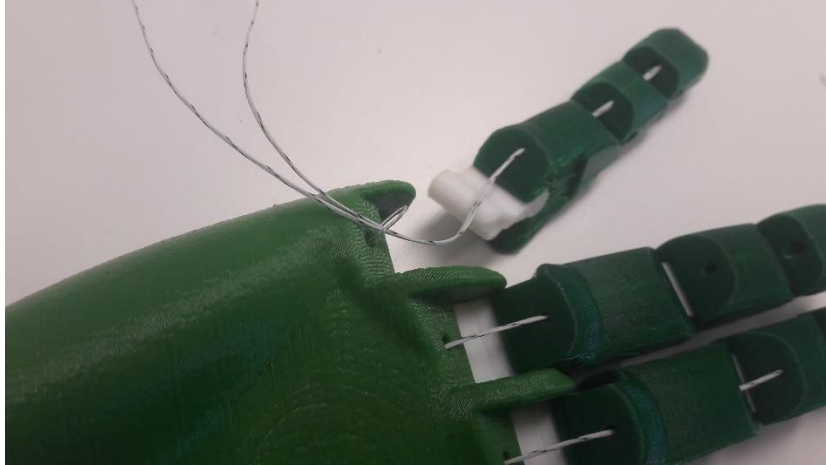


*Imagen pasando los hilos.*



### 2.3.- Unión dedos con palma y mano con antebrazo.

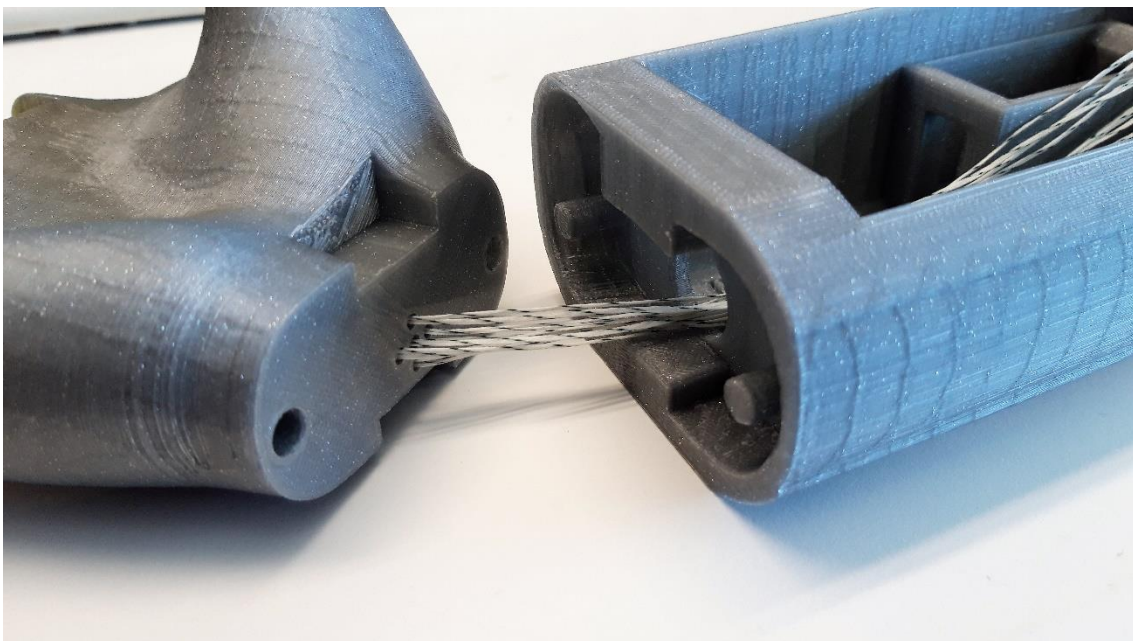
Al igual que hicimos con los dedos pasamos primero por la palma de la mano un cable fuerte y lo doblamos a la mitad para después tirar de los hilos que tenemos en los dedos.



*Imagen unión dedo y palma.*

Conectamos los dedos mediante la articulación a la palma de la mano y cuando tengamos los 5 procedemos a pasar los hilos por el orificio del antebrazo.

Hacemos coincidir en su posición la palma con el antebrazo y apretamos con suavidad, tienen que quedar unos cuantos milímetros entre las ranuras de las palmas y el antebrazo, para que cuando se quieran desmontar poder introducir un destornillador plano.



*Imagen unión mano y antebrazo.*

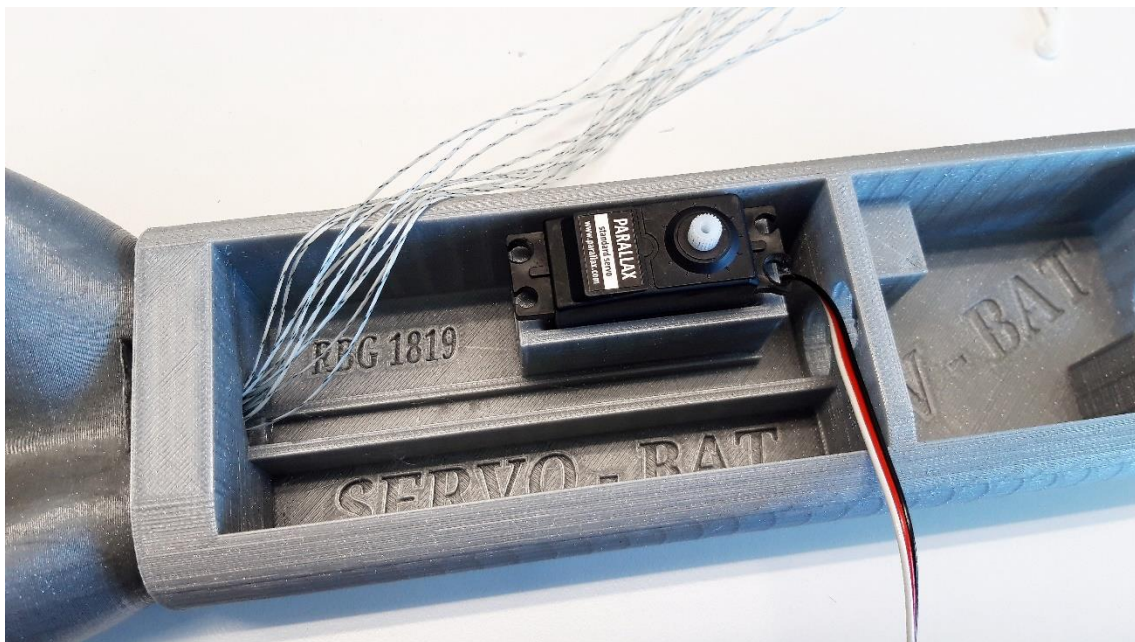
## 2.4.- Colocación del servo en caja.

Pasamos los hilos por la pieza unida al servo y después la atornillamos, pero sin llegar a atarlos.



*Imagen pasando los hilos.*

Ahora es el momento de posicionar el servo en el antebrazo, pasamos el cable por el hueco inferior y colocamos el servo apretando hacia abajo.



*Imagen servo en antebrazo.*

## 2.5.- Calibrado del servo (utilizando el programa del punto 1).

Antes de atar los hilos vamos a utilizar el programa para calibrar los grados entre los que queremos que se mueva el servo. Observamos el movimiento y ajustamos aumentando o disminuyendo los ángulos de apertura y cierre. En el ejemplo el ángulo de apertura es 45 y el de cierre 0 grados, tendremos que modificarlos en la línea del programa en la que aparece lo siguiente:

```
int opened_angle = 45, closed_angle = 0;
```

## 2.6. – Atado de cuerdas al servo.

Cuando ya tenemos calibrado el servo atamos las cuerdas de manera que cuando el servo se encuentre en posición de reposo estén tensas, pero sin tirar de los dedos. Cuando el servo pase a la otra posición tirará de las cuerdas cerrando la mano.



*Imagen mano abierta y cerrada.*



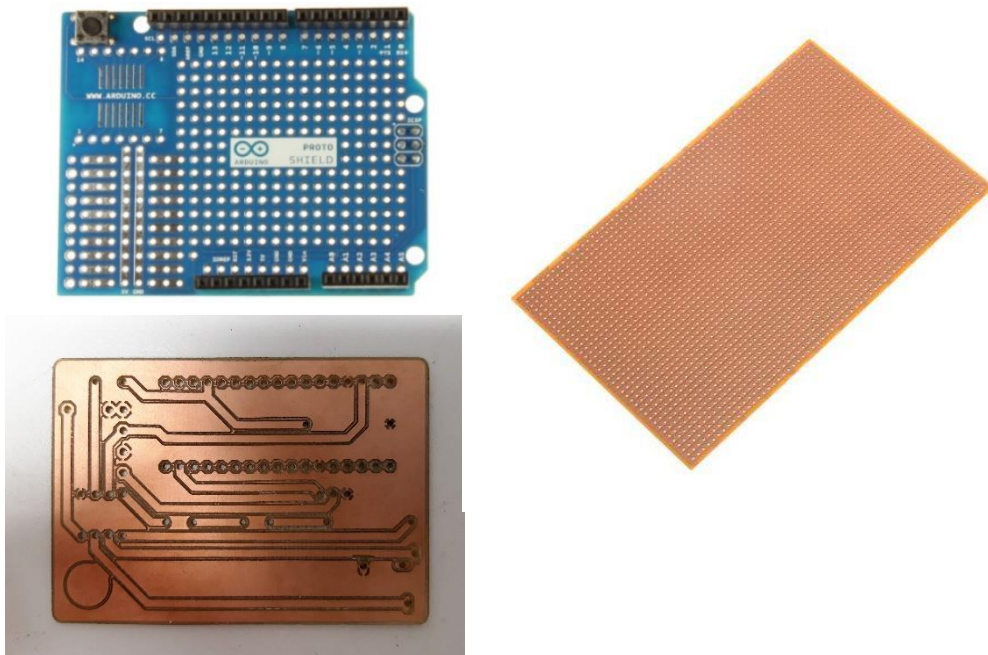


### 3.3. – Componentes electrónicos y soldado.

Para realizar la placa vamos a necesitar los siguientes componentes electrónicos:

- Placa base.
- Arduino Micro.
- Cables.
- 2 pilas de 9V.
- 4 pilas de 1,5V AA.
- 3 conectores de pilas.
- Portapilas de 4 AA.
- Cable de tres electrodos con conector tipo Jack macho.
- Conector Jack hembra.
- Tira de zócalos macho-hembra.
- Tira de zócalos macho-macho.
- Resistencia de 1 M $\Omega$ .
- Resistencia 120  $\Omega$ .
- Condensador 10 nF.
- 2 Condesadores 100  $\mu$ F.
- Zócalo para integrado de 8 patas.
- Integrado INA128P (Amplificador de instrumentación).
- Interruptor (opcional).

Tenemos varias opciones de placa base, la primera sería una placa de prototipado suministrada por Arduino, también tenemos la opción de realizar nuestra propia placa cortando con las dimensiones que deseemos de una placa de prototipado estándar, o incluso realizar nuestra propia placa PCB.



*Imagen placas placa base Arduino PROTO SHIELD, placa de prototipado estándar y PCB realizada con CNC.*

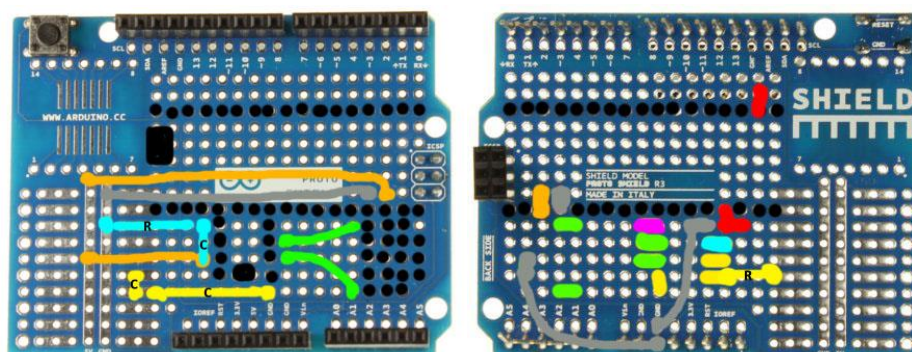
En cuanto al soldado de los componentes tenemos que prestar atención si utilizamos la placa estándar en separar las pistas mediante un cúter para aislar unas pistas de otras, ya que como ocurría en la placa protoboard los orificios están conectados eléctricamente en línea recta si miramos la placa de modo horizontal o vertical.

### 3.4. – Alimentación INA128P mediante pilas.

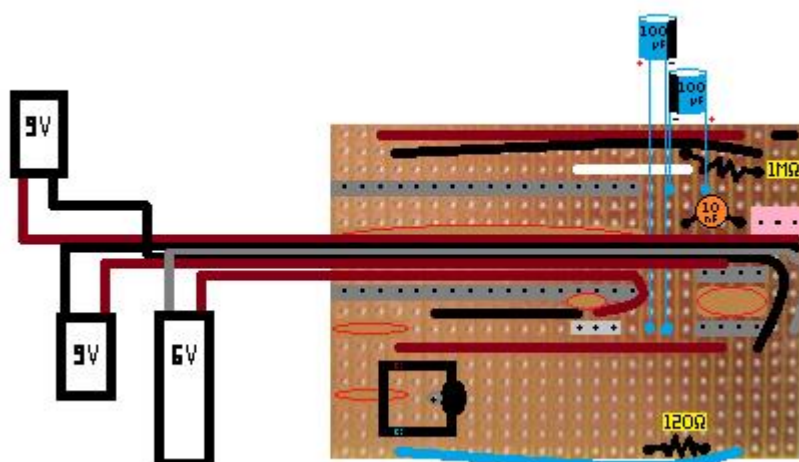
El INA128 es un amplificador de instrumentación muy versátil que vamos a utilizar en el módulo EMG. Viene en un formato de integrado de ocho patas y para su alimentación vamos a necesitar +9V, -9V y GND.

El modo para conseguirlo aparece en el esquema general de todo el circuito que vimos anteriormente, para ello unimos el positivo y el negativo de cada pila y esta unión la llegamos también a tierra (GND) y a la patilla 5, después llevaremos el positivo que nos queda libre de una pila a la patilla 7 (+9V), y el negativo que nos queda libre en la otra pila a la patilla 4 (-9V).

### 3.5. – Ejemplos de esquemas realizados.

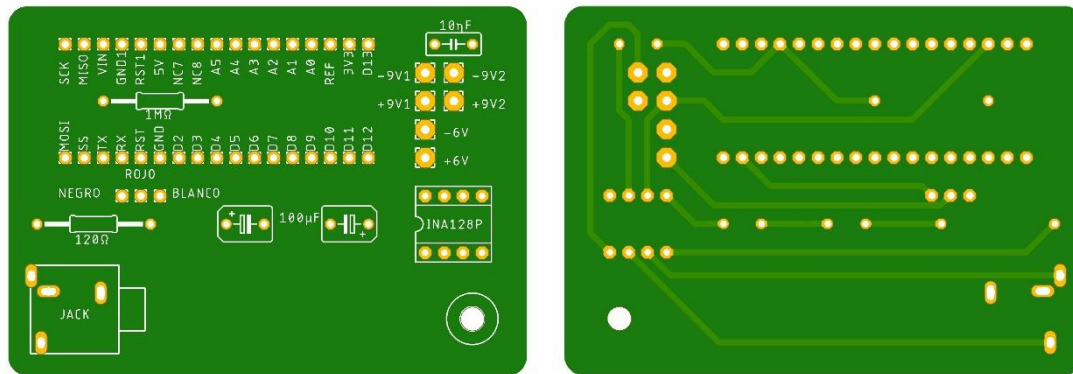


*Vista frontal y trasera del circuito montado sobre placa Arduino PROTO SHIELD.*



*Croquis del circuito sobre placa de prototipado estándar.*





*Vista frontal y trasera de tarjeta PCB diseñada con Eagle.*

### 3.6. – Prueba de la señal EMG usando un osciloscopio.

Para probar la señal que nos genera el módulo EMG podemos utilizar un osciloscopio. Conectamos una sonda y la ponemos en A0, sin olvidarnos de colocar la pinza del cocodrilo a masa. Cuando nos conectemos los electrodos y movamos el brazo podremos observar la señal que vamos a utilizar para controlar el servo con el Arduino.

## 4.- Sistema completo:

### 4.1.- Carga y ejecución del programa para servo y EMG juntos.

Creemos y cargamos un nuevo programa en Arduino para controlar el sistema completo.

#### Ejemplo de programa para servo y EMG:

```
/*
 * Controla el servo usando el EMG
 * Cada pico del EMG abre la mano si estaba cerrada
 * o la cierra si estaba abierta
 */

#include <Servo.h>

int emg_pin=0; //Analog 0
int valor_emg=0;
int led = 13;
int servo_pin = 11;
Servo servo;
int estado=0;

void setup()
{
  Serial.begin(9600);
  pinMode(led, OUTPUT);
  servo.attach(servo_pin);
}

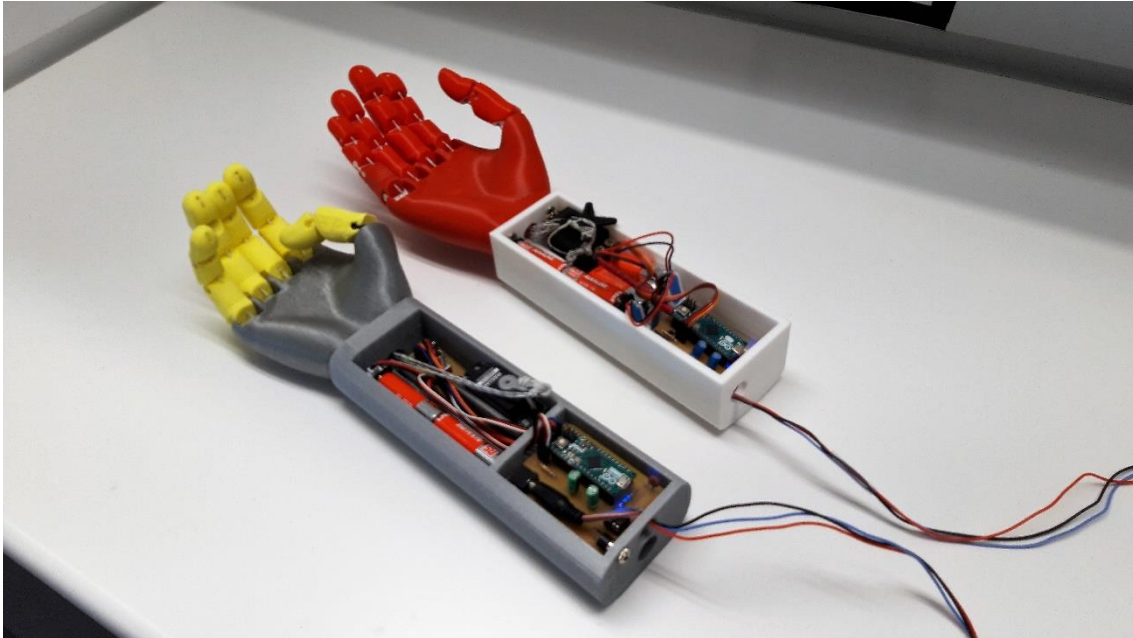
void loop()
{
  valor_emg=0;
  valor_emg=analogRead(emg_pin);
  Serial.println(valor_emg);
  if (valor_emg>100 && estado==0)
  {
    digitalWrite(led, HIGH);
    servo.write(20);
    estado=1;
    delay(1000);
  }
  else if (valor_emg>100 && estado==1)
  {
    digitalWrite(led, LOW);
    servo.write(160);
    estado=0;
    delay(1000);
  }
  Serial.println(valor_emg);
  delay(50);
}
```

### 4.2. – Probar el sistema completo.

Una vez cargado el nuevo programa ya podremos probar nuestro sistema al completo, dependiendo de la persona que lo utilice quizás tengamos que ajustar `valor_emg>100` del bucle `if` y `else if`.

#### 4.3. – ¡Gozo y algarabía!

Hemos conseguido que nuestro sistema funcione correctamente, cuando cerramos el brazo nuestra mano biónica se cierra, al volverlo a cerrar la mano se abre de nuevo.



*Imagen de dos versiones del sistema completo.*