

Guía montaje brazo Keyestudio

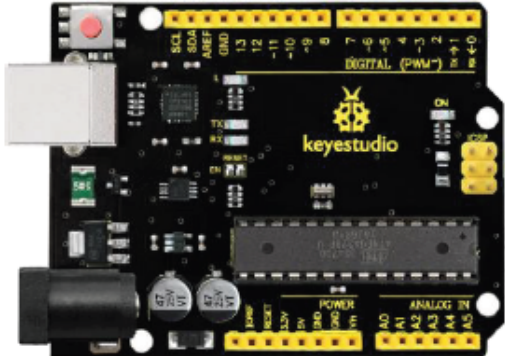
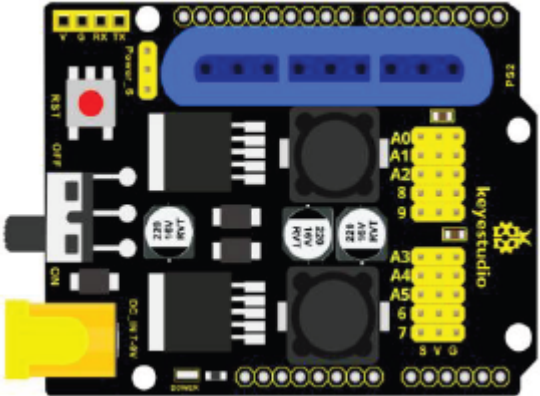
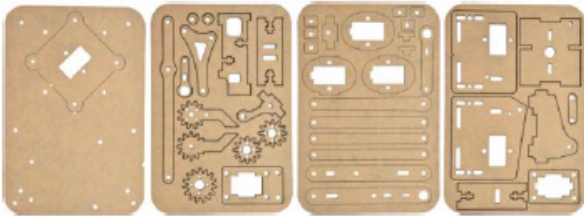


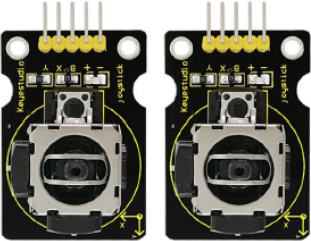
CFR Vigo

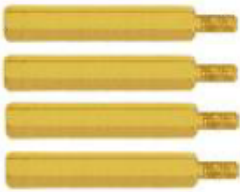
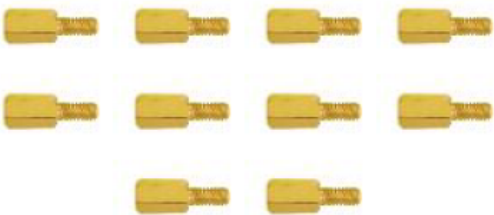
Antonio Rodríguez Nieto 2025

Elementos	3
Instalación drivers conexión Arduino Keyestudio	8
Cómo “configurar” los servos	10
Ensamblaje	12
1. Base del brazo robótico	12
2. Montaje del servo en la base	18
3. Montaje del servo izquierdo	21
4. Montaje del servo derecho	25
5. Integración de los brazos	34
6. Pinza (Servo de pinza)	48
7. Finalización de brazo (pinza y base)	56
8. Mando joysticks	62
9. Conexionado	66

Elementos

Elemento	Cantidad	Imagen
Keyestudio V4.0 placa de control (Arduino Uno) En inglés: Keyestudio V4.0 Control Board	1	 The image shows a black Keyestudio V4.0 Control Board, which is an Arduino Uno compatible board. It features a USB Type-C port on the left, a DC power jack, and various digital and analog pins along the edges. The board is populated with a microcontroller, memory chips, and other electronic components. The 'keyestudio' logo is visible in the center.
Keyestudio Servo driver En inglés: Keyestudio Servo Motor Driver Shield	1	 The image shows a Keyestudio Servo Motor Driver Shield. It is a black PCB with two large blue servo motor connectors. It includes a 5V regulator, a 3.3V regulator, and several status LEDs. The board is designed to be used as a shield on an Arduino board. The 'keyestudio' logo is visible on the right side.
Placas acrílicas En inglés: Acrylic Boards	1	 The image shows four different acrylic boards of various shapes and sizes. They are made of a light-colored material, likely acrylic, and have various holes and slots for components. The boards are arranged in a row.
Mando acrílico En inglés: Acrylic Handle	1	 The image shows an acrylic handle, which is a light-colored, curved piece of acrylic. It has several holes and slots for components. The text 'EXTRUDED ACRYLIC' is visible on the handle.

ABS soporte cilíndrico En inglés: MeArm ABS Cylindrical Holder	1	
Servos	4	
Módulo bluetooth BT-24	1	
Módulos de joystick	2	
Destornillador	1	
Llave	1	
M3*6mm tornillos cabeza redonda En inglés: M3*6MM Round Head Screws	12	

M3*10mm tornillos cabeza redonda En inglés: M3*10MM Round Head Screws	22	
M3*14mm tornillos cabeza plana En inglés: M3*14MM Flat Head Screws	2	
M3*12mm tornillos cabeza redonda En inglés: M3*12MM Round Head Screws	12	
M3*24mm + 6mm pilares En inglés: M3*24+6MM Copper Pillar	4	
M3*6mm + 6mm pilares En inglés: M3*6mm+6mm Copper Pillar	10	
M3 tuercas hexagonales inoxidables En inglés: M3 Stainless Steel Hex Nuts	22	

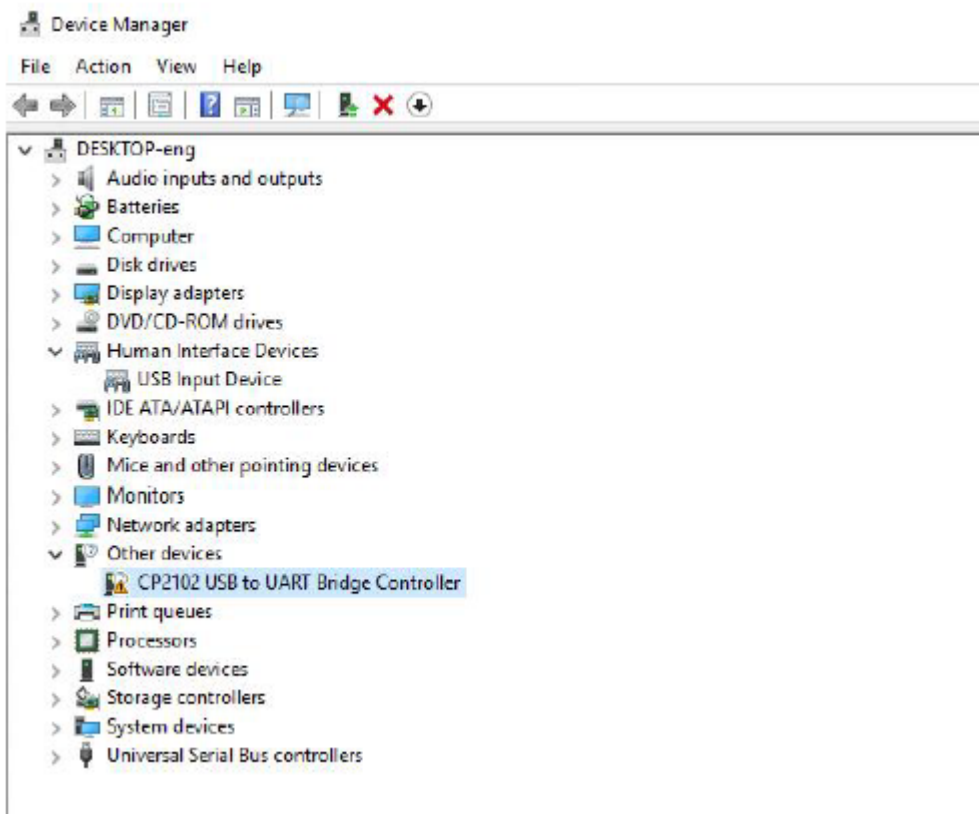
M3 tuercas hexagonales En inglés: M3 Hexagon Nuts	24	
M1.2x5mm tornillos phillips autorroscantes En inglés: M1.2x5MM Phillips Self-tapping Screws	8	
M2x5mm tornillos phillips autorroscantes En inglés: M2x5MM Phillips Self-tapping Screws	10	
M3 304 arandelas inoxidables En inglés: M3 304 Stainless Steel Flat Washer	10	
M2x8mm tornillos phillips autorroscantes En inglés: M2x8MM Phillips Self-tapping Screws	2	
M3x16mm tornillos cabeza plana En inglés: M3*16MM Flat Head Screws	2	

Cable macho-hembra 10cm	4	
Cable hembra-hembra 50cm	10	
Portapilas 18650	1	

Instalación drivers conexión Arduino Keyestudio

En el caso de que el ordenador no detecte el controlador de Keyestudio, sería necesario bajar los drivers de CP2102 para establecer la conexión correctamente.

Para ello, descargaríamos los drivers adjuntos, y yendo al administrador de dispositivos:



Clickamos dos veces en el controlador, y actualizamos el driver, seleccionando el archivo descargado:

← Update Drivers - CP2102 USB to UART Bridge Controller

Browse for drivers on your computer

Search for drivers in this location:

C:\Users\Administrator\Desktop\arduino-1.8.12\drivers\CP210x_6.7.0

Browse...

☒ Include subfolders

→ Let me pick from a list of available drivers on my computer

This list will show available drivers compatible with the device, and all drivers in the same category as the device.

Next

Cancel

De esta forma, ya estaría visible la conexión establecida con el controlador:

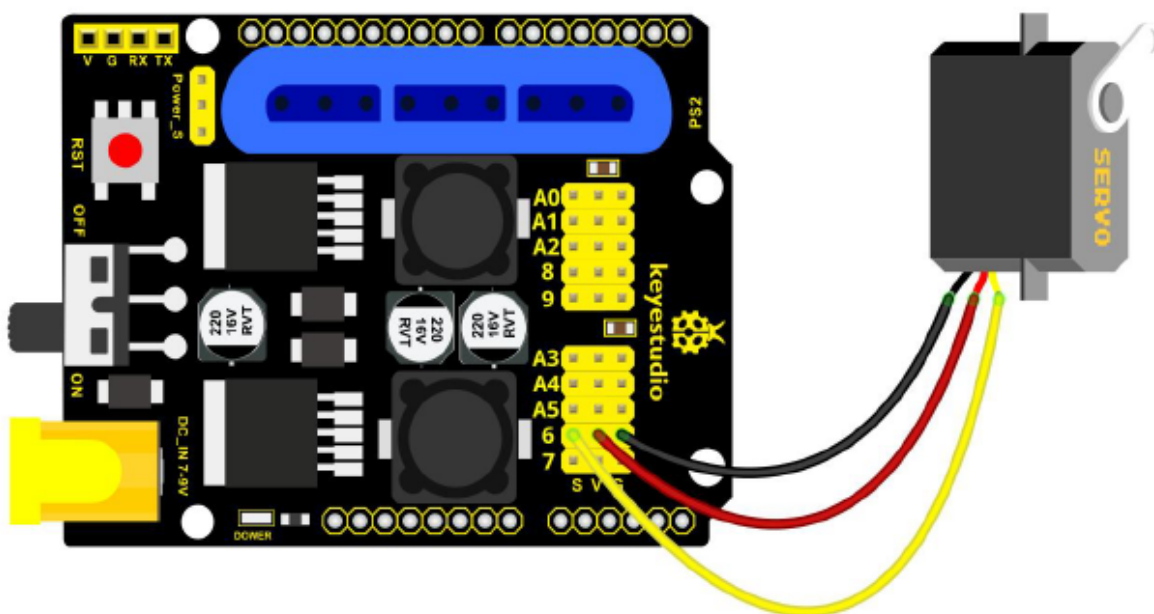
- USB Input Device
- > IDE ATA/ATAPI controllers
- > Keyboards
- > Mice and other pointing devices
- > Monitors
- > Network adapters
- ▼ Ports (COM & LPT)
 - Silicon Labs CP210x USB to UART Bridge (COM7)
- > Print queues
- > Processors
- > Software devices
- > Storage controllers
- > System devices
- > Universal Serial Bus controllers

Cómo “configurar” los servos

Teniendo en cuenta que los servos disponen de un rango de movimiento limitado (usualmente 0° - 180°), es necesario configurar su posición previo a anclar la cabeza del servo a la estructura.

Esto se debe a que si montamos el servo en una posición en la que se debería hallar en 0° , pero en cambio lo hemos fijado cuando está en una posición de 90° , nos quedaríamos con la mitad de rango de actuación (90° - 180° , en vez de 0° - 180°).

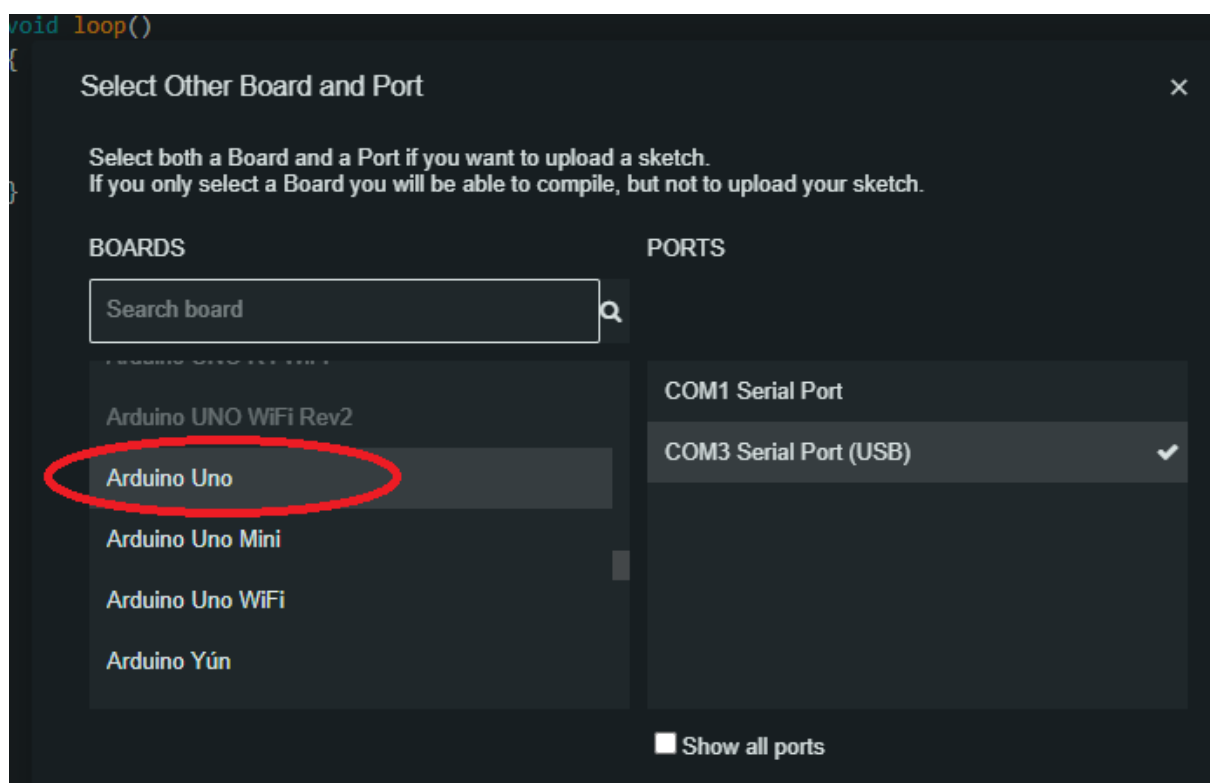
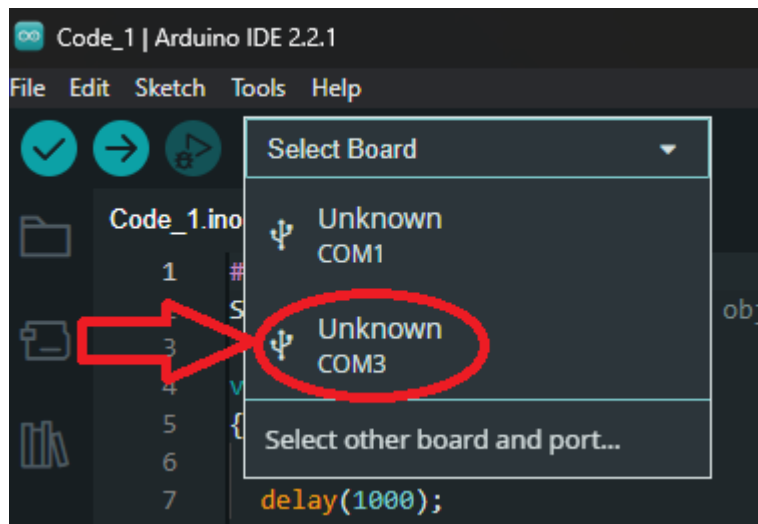
Por tanto, para ello, **habiendo instalado los drivers de keystudio** en el ordenador, conectamos el servo montado en el driver de servos, **en el pin sobre el que queremos actuar “X”**, tal que:



Cable naranja en la S de Señal.

Posteriormente, **conectamos el controlador al ordenador y la fuente de alimentación al driver del servo, cambiando el interruptor del driver a “ON”**.

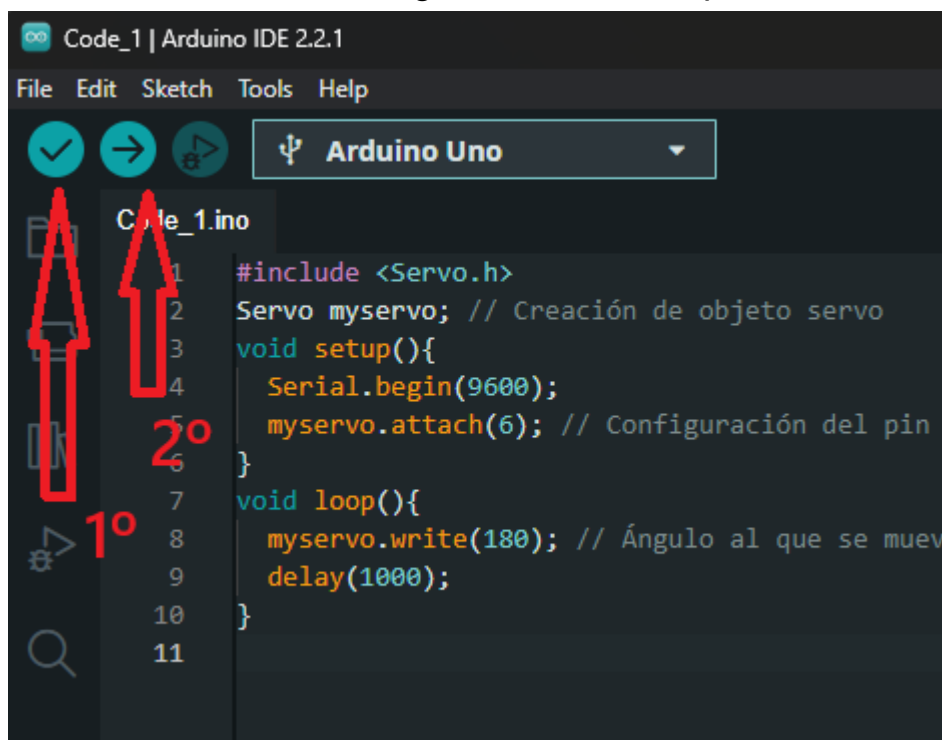
Por último, abrimos el **programa Arduino IDE**, seleccionamos el puerto en el que creemos que se encuentra, y como controlador **Arduino uno**:



Una vez configurado, copiamos el siguiente código en el programa, cambiando X por el pin en el que se haya conectado, e Y por los grados a los que queremos rotar el servo:

```
#include <Servo.h>
Servo myservo; // Creación de objeto servo
void setup(){
  Serial.begin(9600);
  myservo.attach(X); // Configuración del pin X
}
void loop(){
  myservo.write(Y); // Ángulo Y al que se mueve
  delay(1000);
}
```

Una vez tenemos el código de esa forma, pulsamos en verificar y subir:

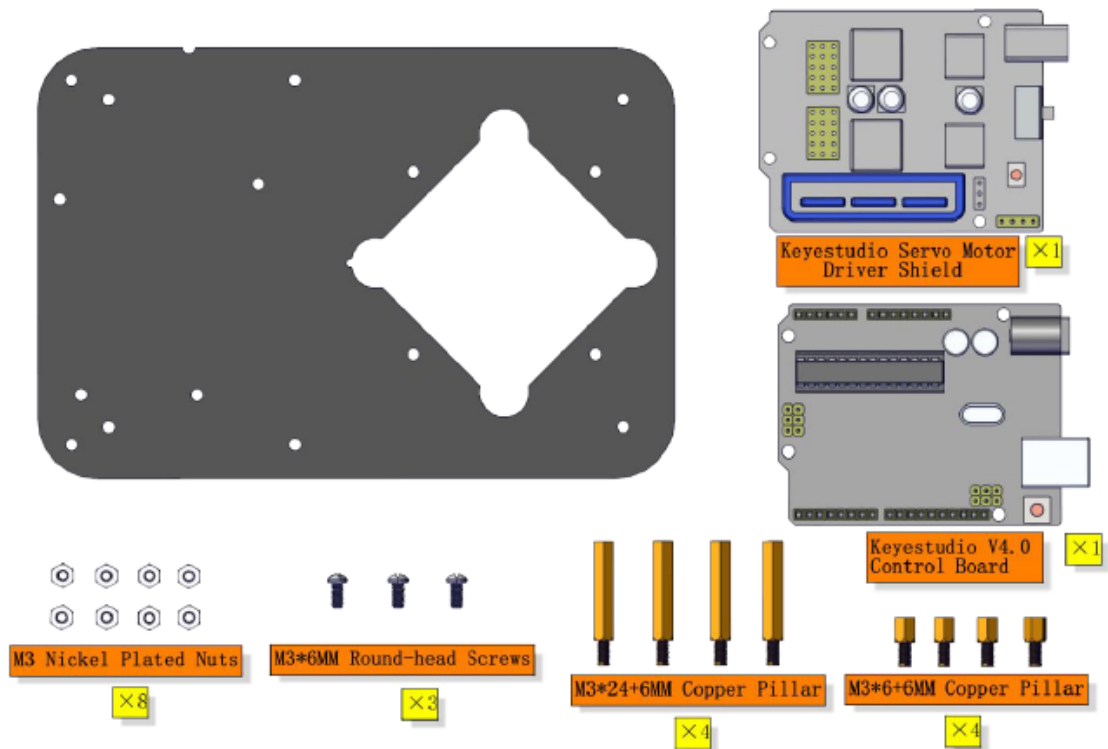


Si verificamos y todo fue bien, podremos subir sin problema el código al Arduino. En caso contrario, habría que corregir el error que está ocurriendo.

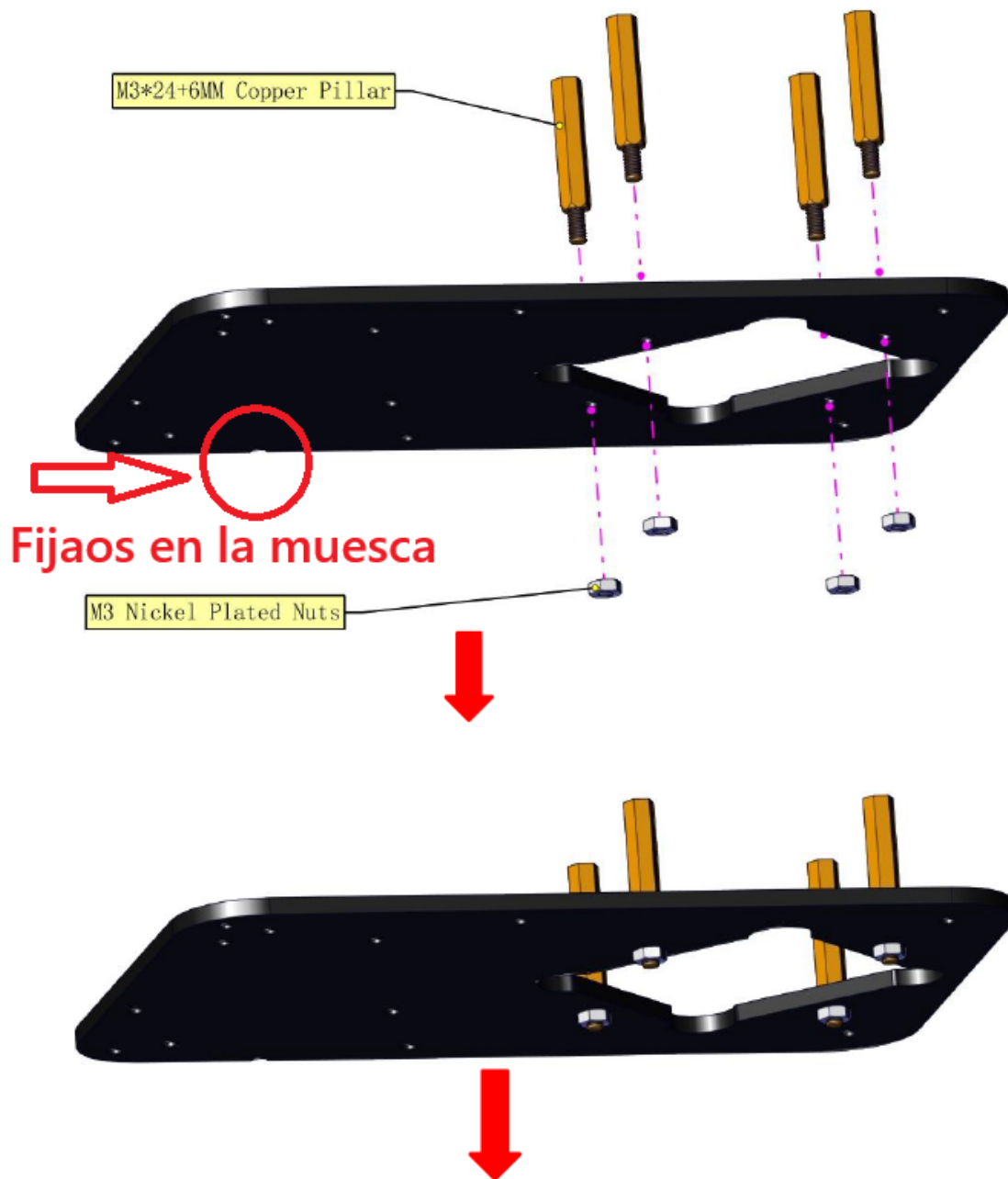
Ensamblaje

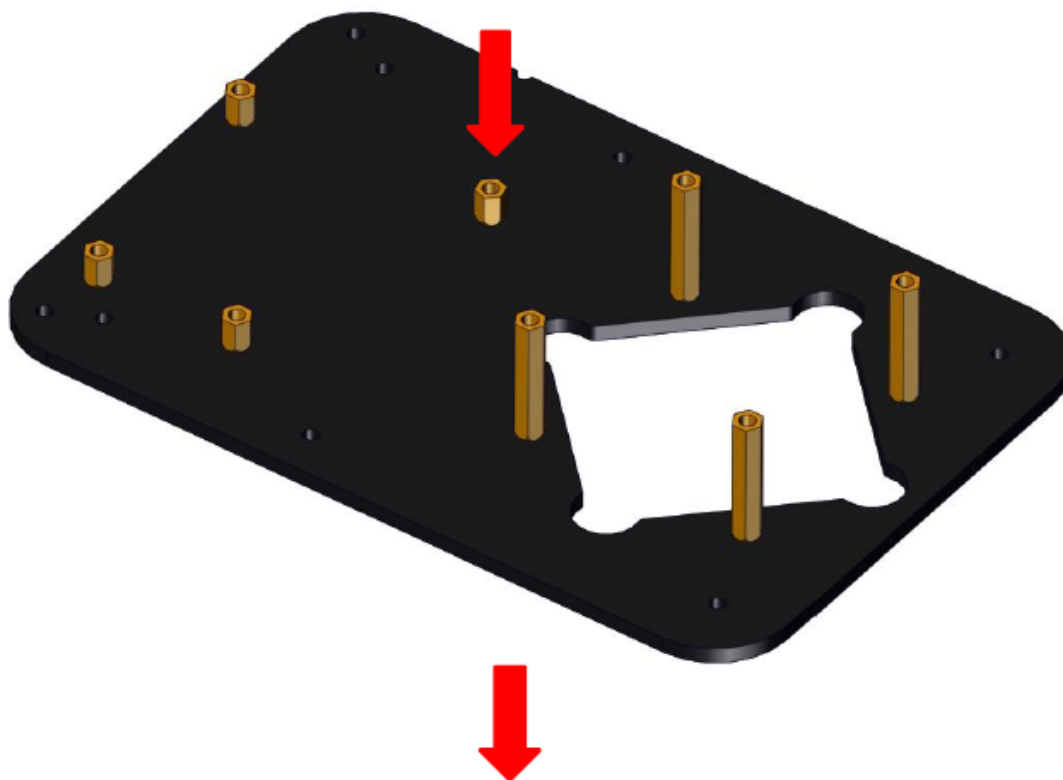
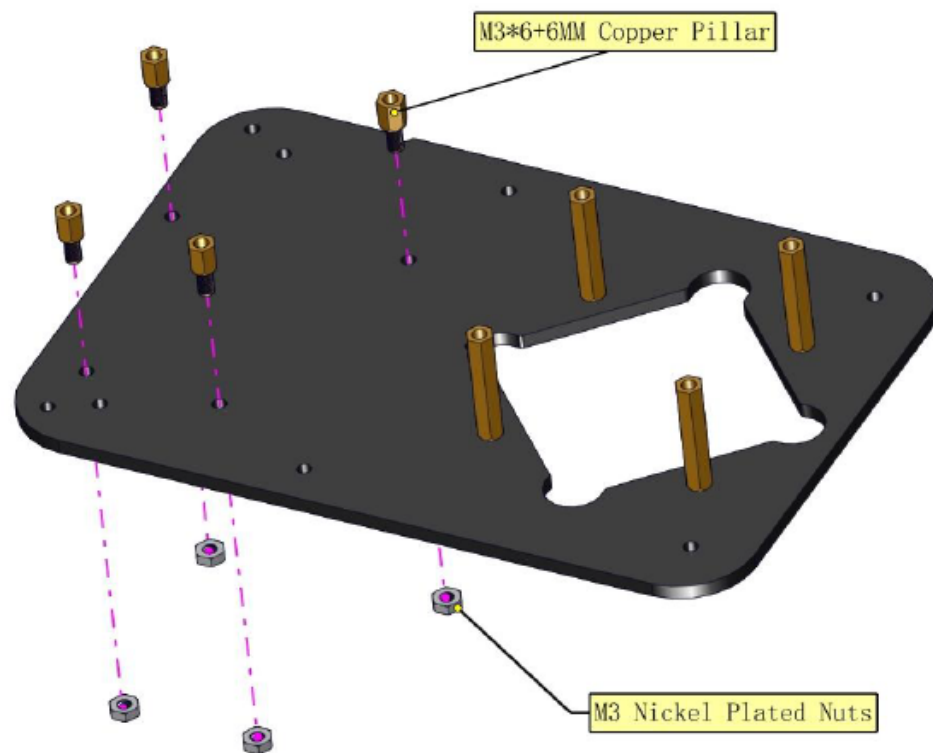
1. Base del brazo robótico

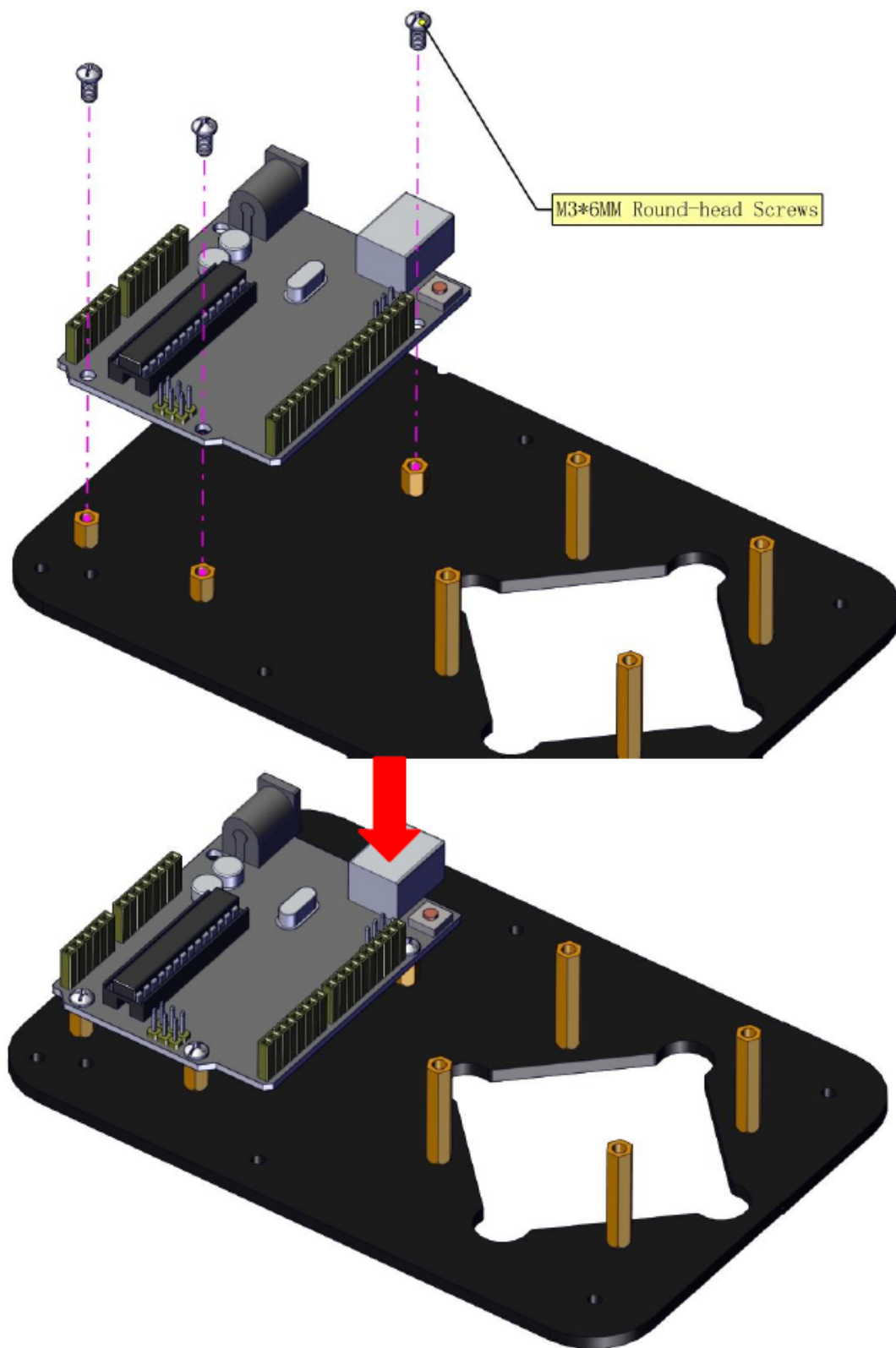
Componentes necesarios:

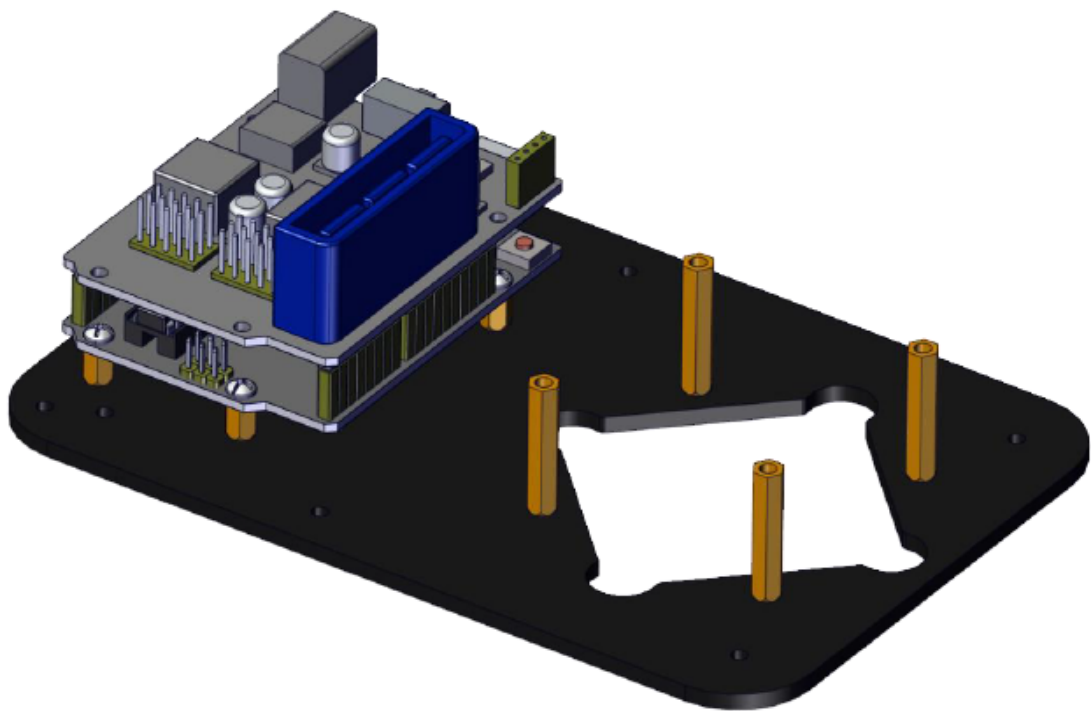
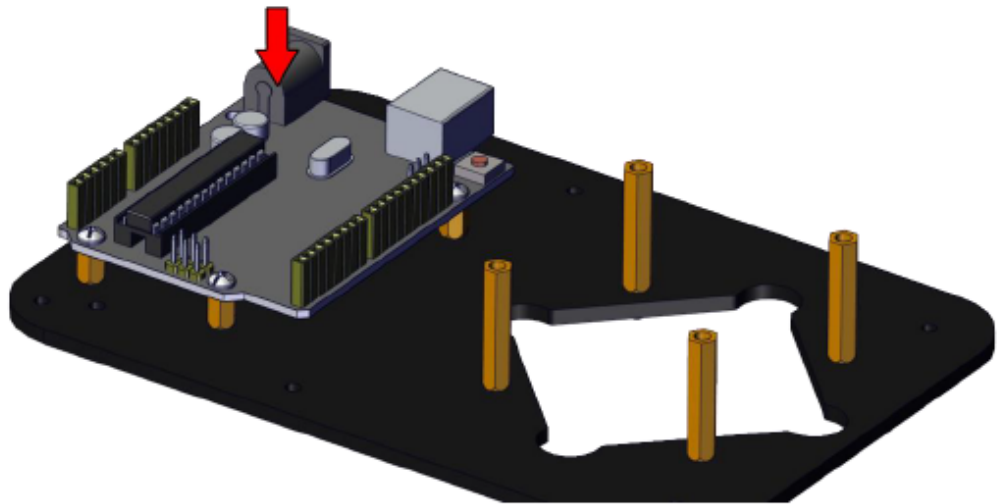
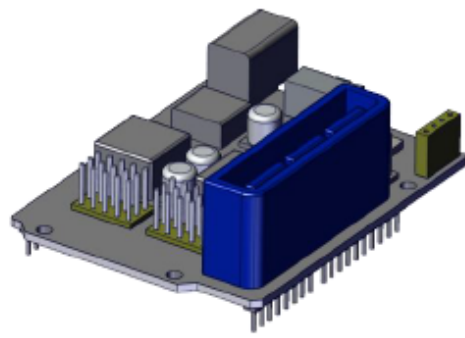


Procedimiento:



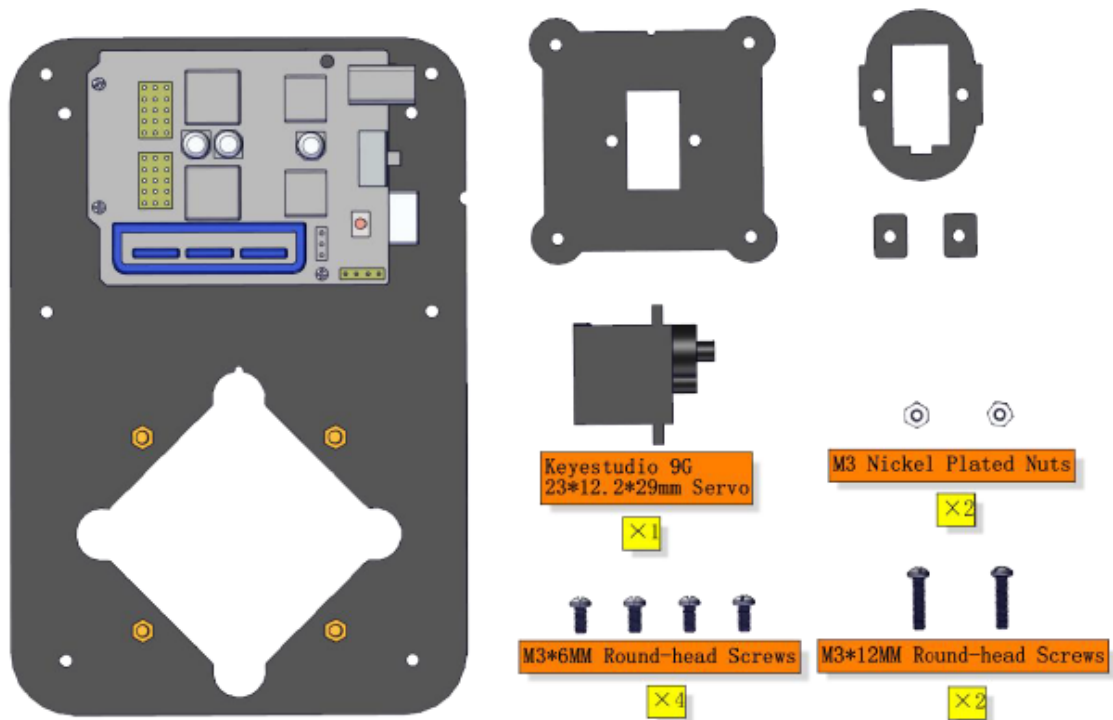






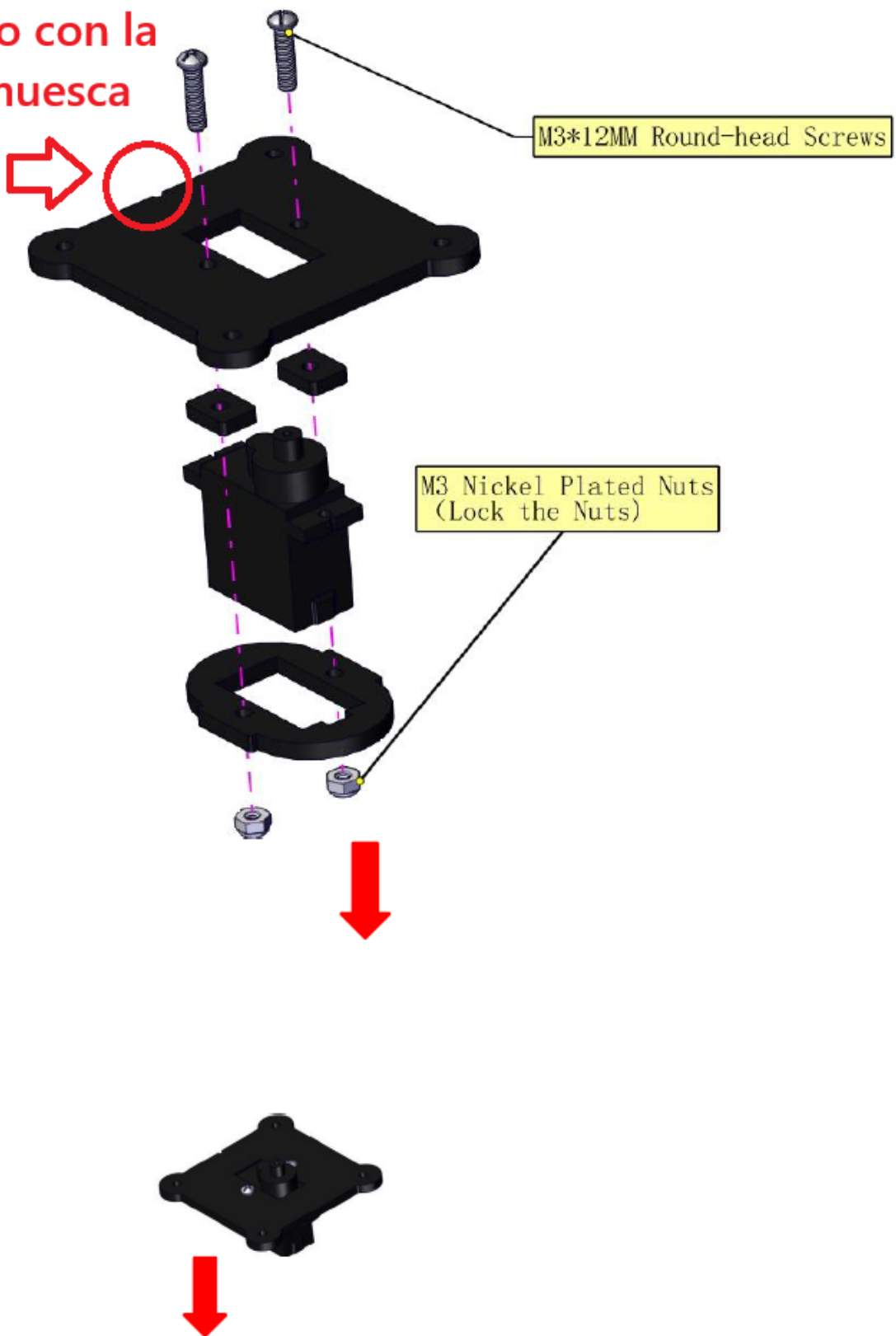
2. Montaje del servo en la base

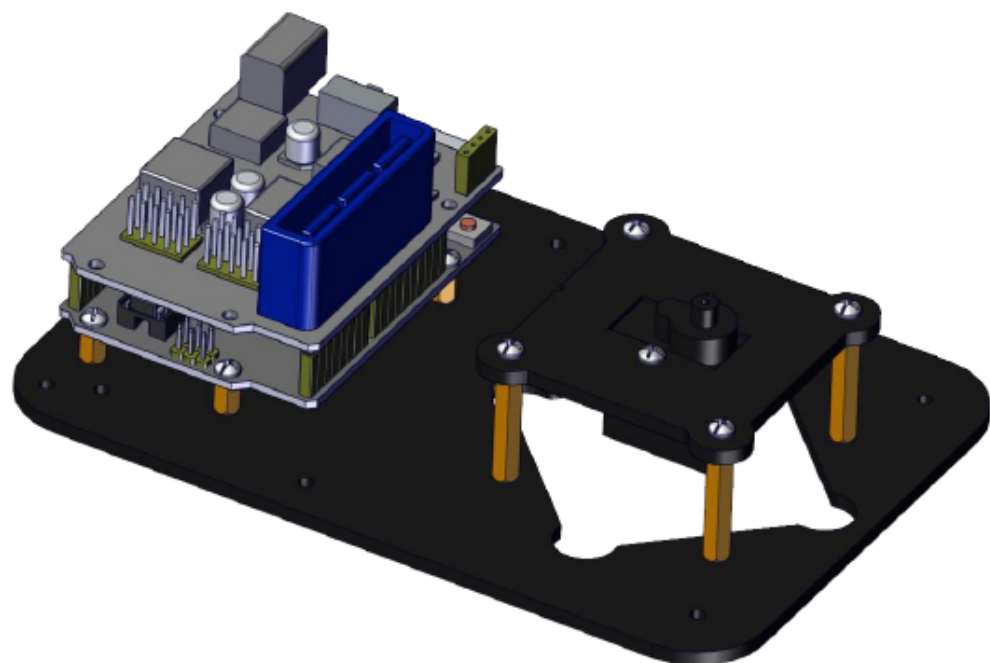
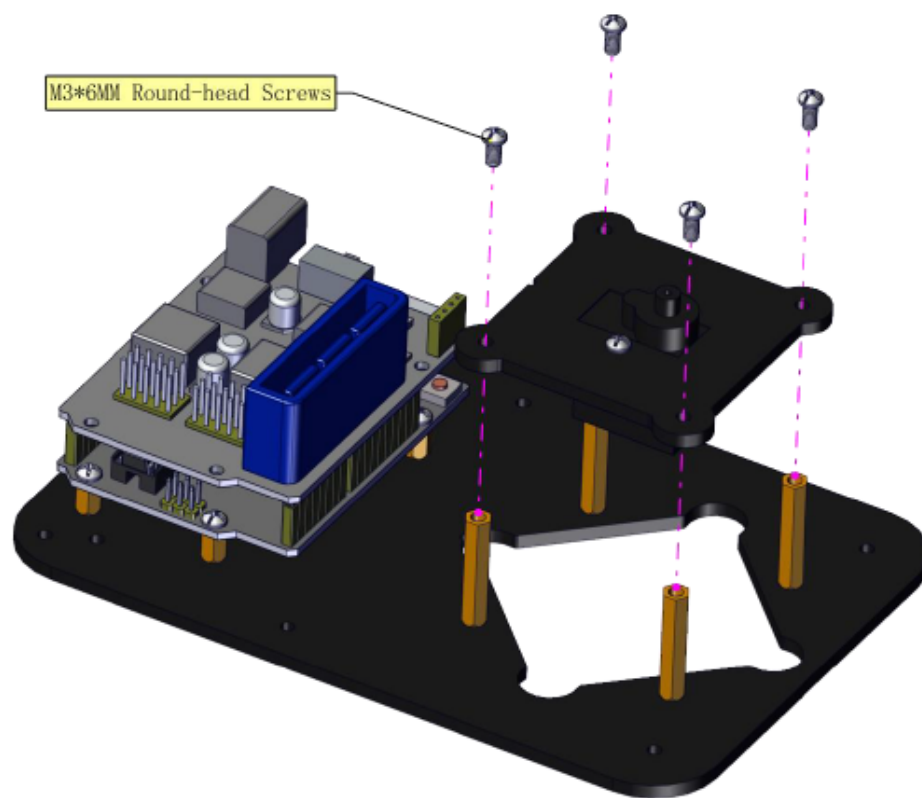
Componentes necesarios:



Procedimiento:

Ojo con la
muesca

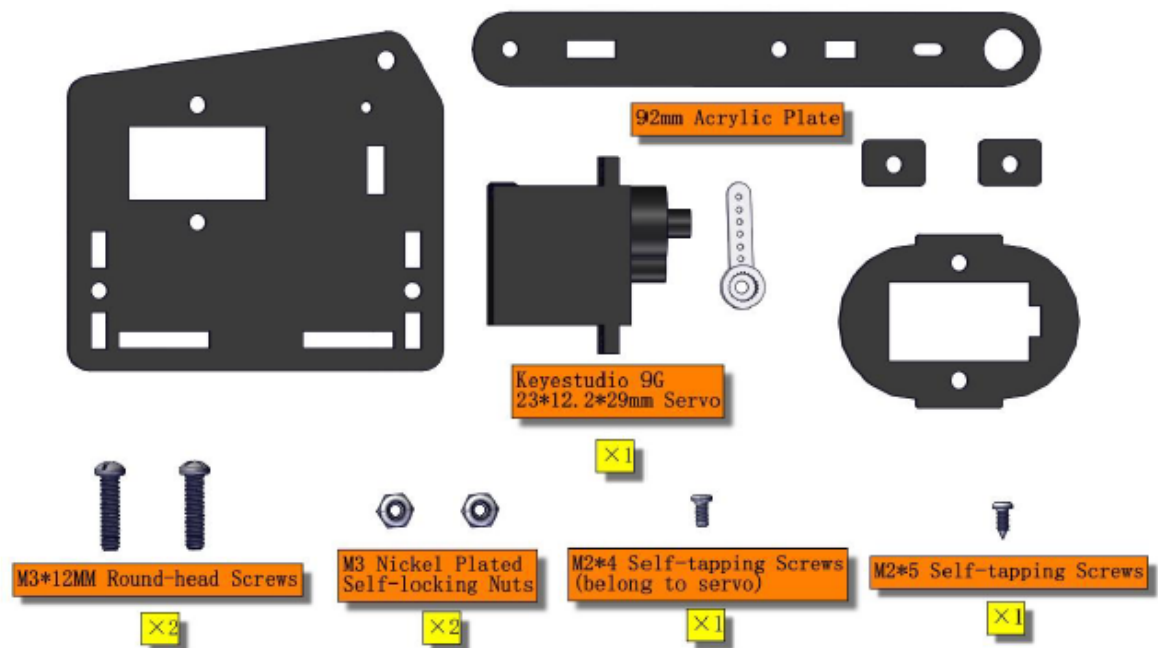




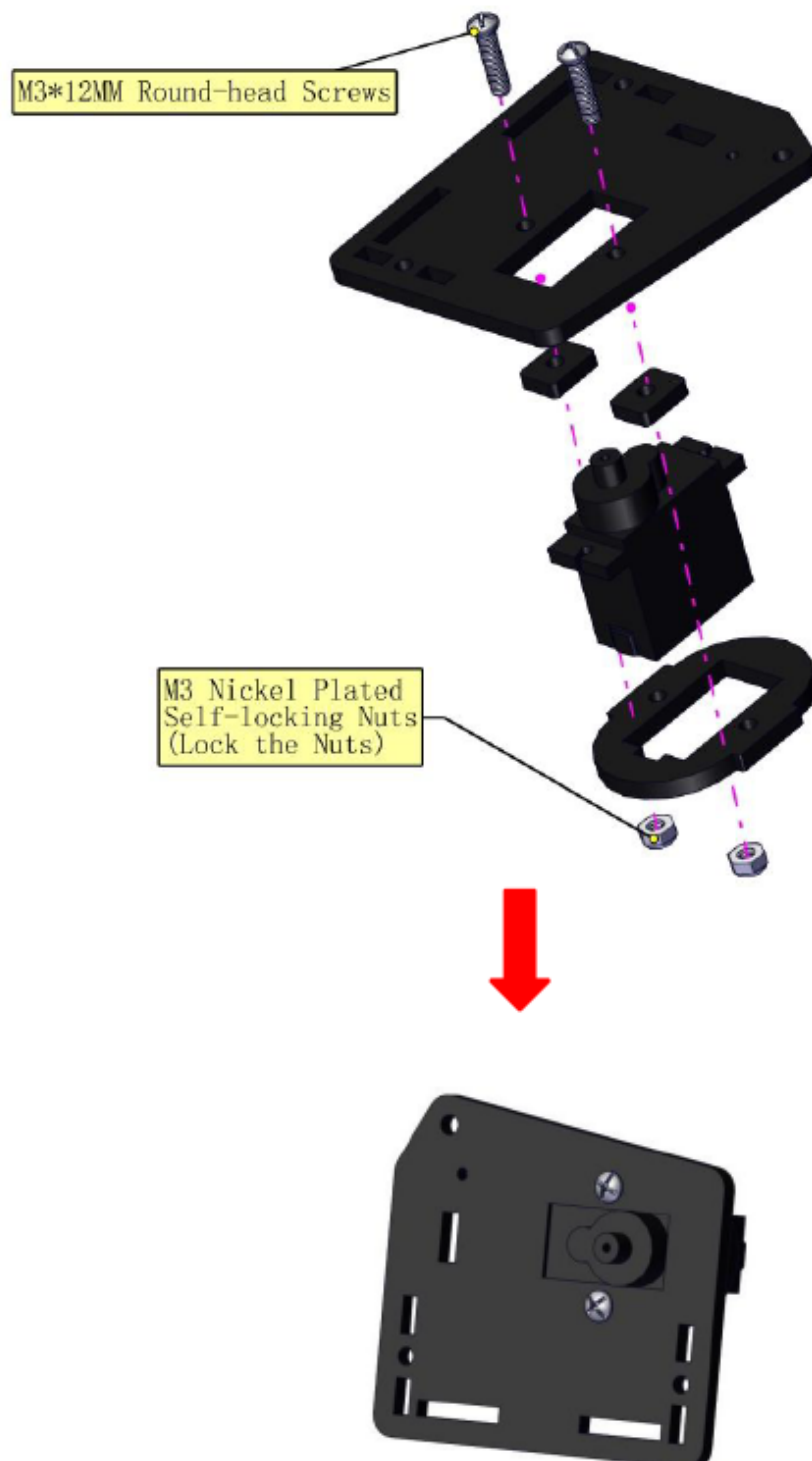
La base con el servo estaría lista

3. Montaje del servo izquierdo

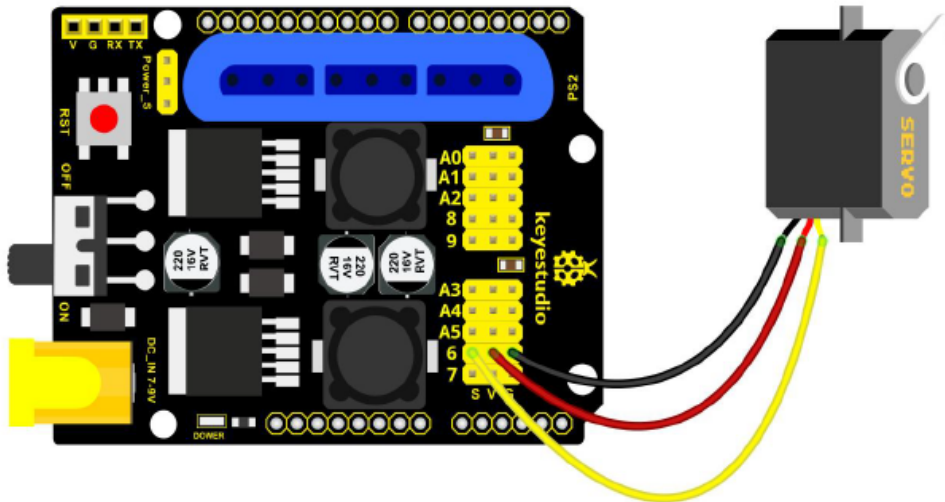
Componentes necesarios:



Procedimiento:



Ahora, para la configuración inicial del servo, como comentamos en el apartado previo, lo conectamos **en el pin 6**, tal que:



Cable naranja en la S de Señal.

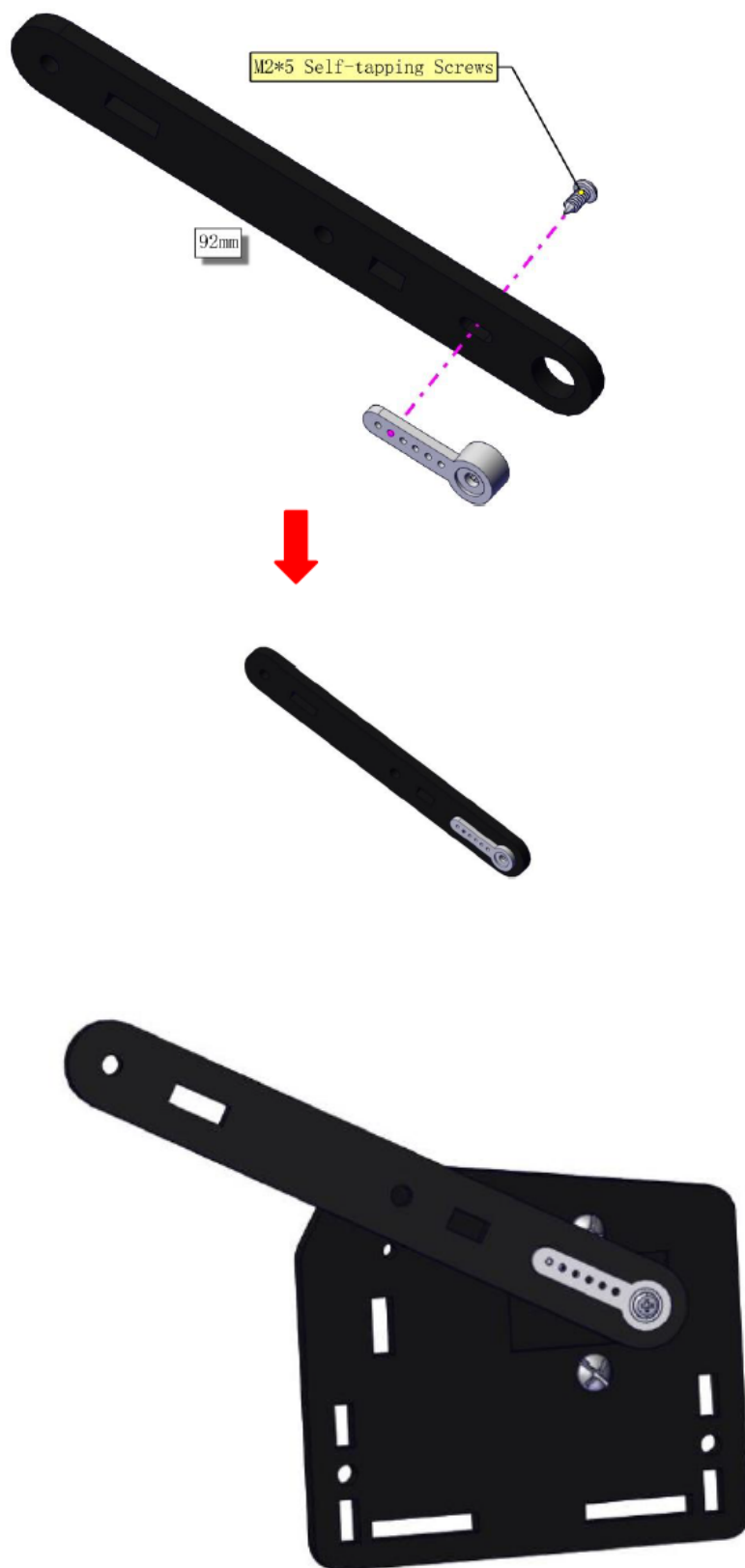
Posteriormente, **conectamos el controlador al ordenador y la fuente de alimentación al driver del servo, cambiando el interruptor del driver a “ON”**.

Con los pasos comentados en el apartado anterior hechos, copiamos el siguiente código en el programa:

```
#include <Servo.h>
Servo myservo; // Creación de objeto servo
void setup(){
  Serial.begin(9600);
  myservo.attach(6); // Configuración del pin 6
}
void loop(){
  myservo.write(180); // Ángulo al que se mueve (180)
  delay(1000);
}
```

Por tanto, se compilaría y subiría el código al Arduino, asegurándonos de que el servo se haya movido a esa posición.

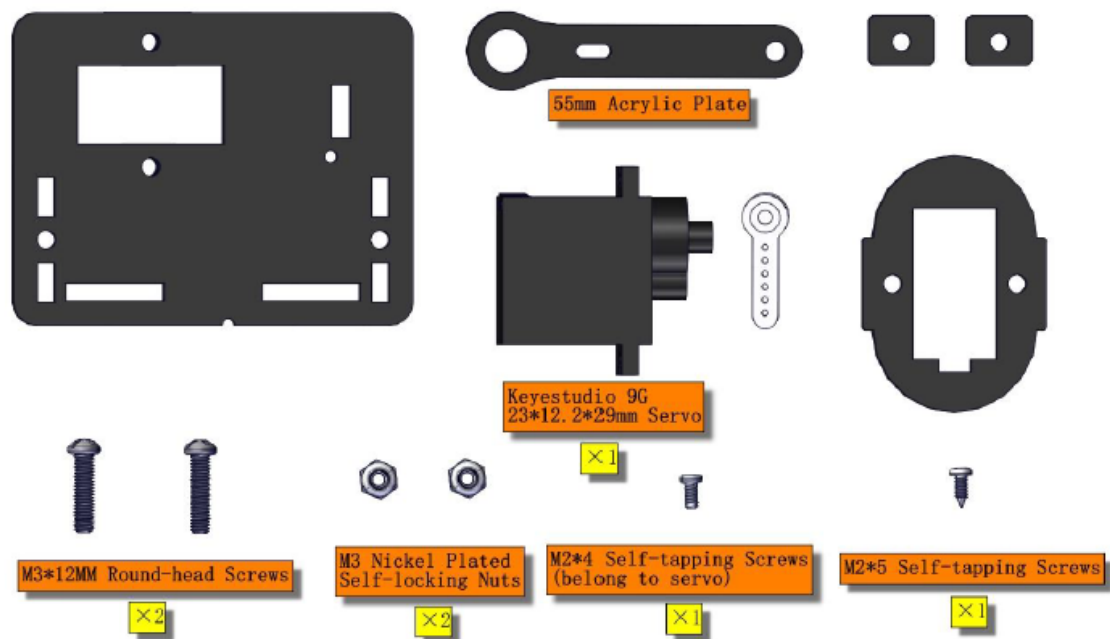
Para terminar el brazo:



El brazo izquierdo estaría listo.

Componentes necesarios:

Componentes necesarios:



Procedimiento:



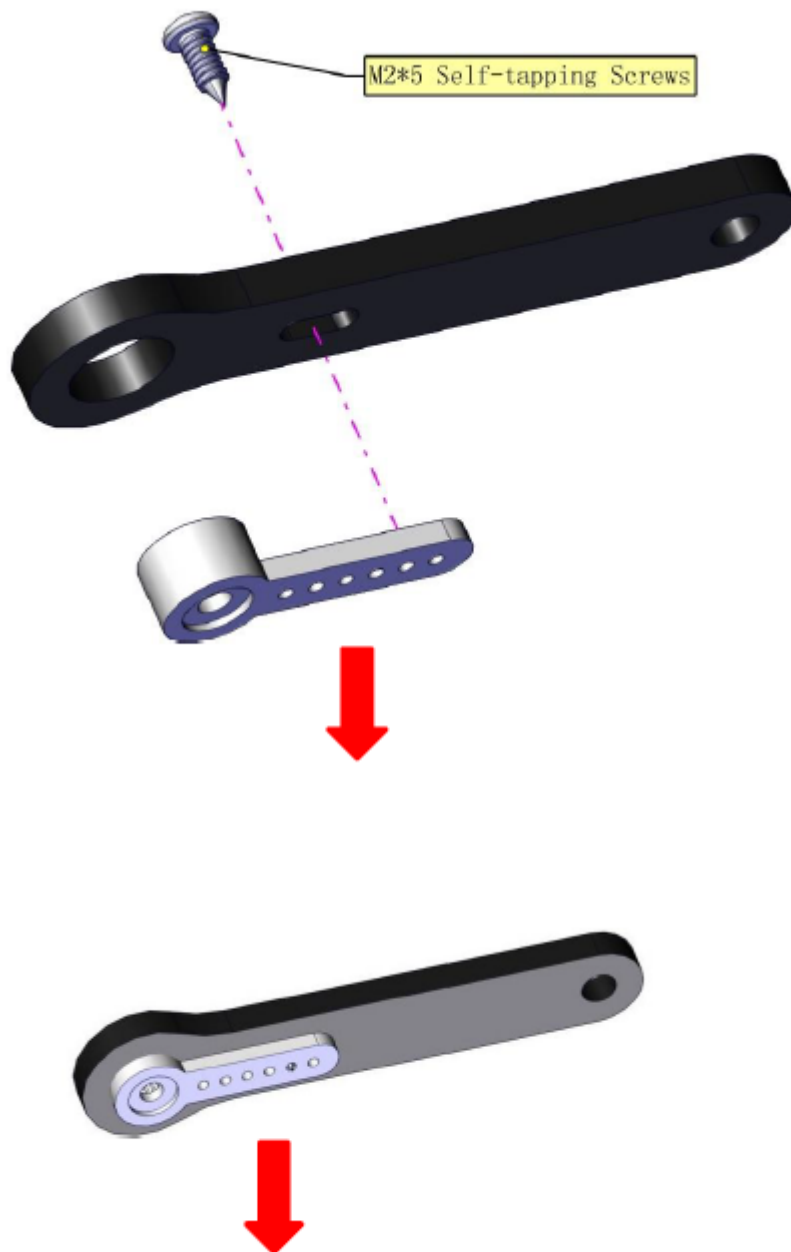
```
#include <Servo.h>

Servo myservo; // Creación de objeto servo

void setup(){
  Serial.begin(9600);
  myservo.attach(A0); // Configuración del pin A0
}

void loop(){
  myservo.write(0); // Ángulo al que se mueve (0)
  delay(1000);
}
```

Una vez se ha rotado, seguimos con el montaje:

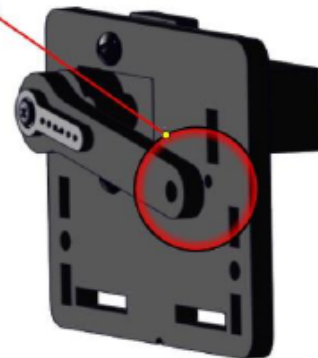
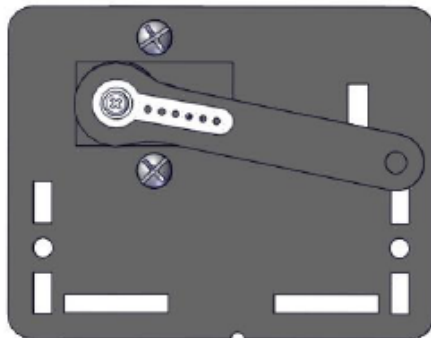




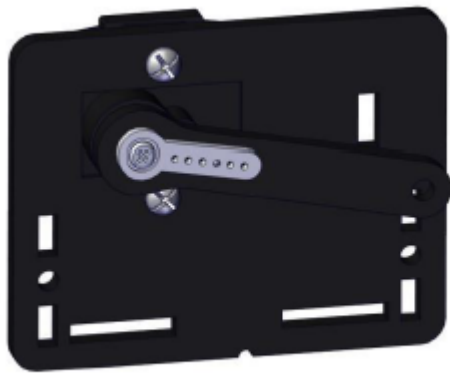
M2*4 Self-tapping Screws
(belong to servo)

El tornillo es de la bolsa del servo

keep the initial angle of servo unchanged,
and the acrylic bar is installed in this
position (in front of the hole) :

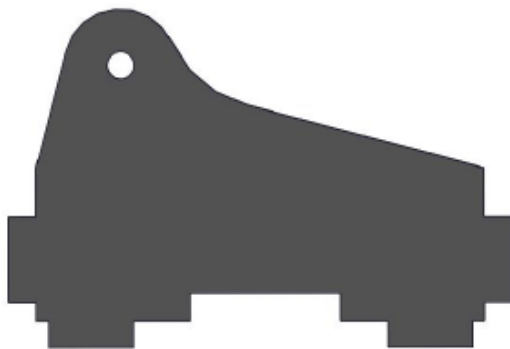


**Intentar no rotar mucho el servo en el
montaje, y que el agujero del brazo
coincida más o menos con el de la "placa"**



Avanzamos con la otra parte del brazo

Componentes necesarios:



M3*10MM Round-head Screws

×1



M3 Nickel Plated
Self-locking Nuts

×1

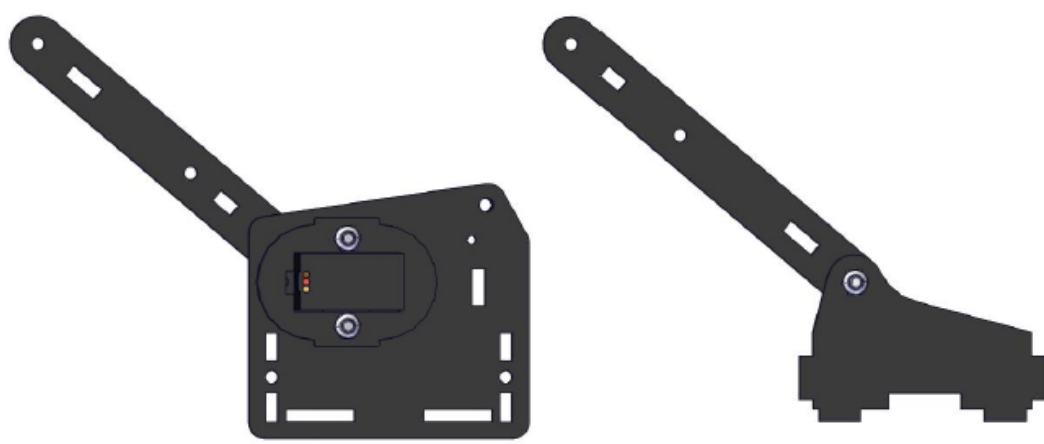


92mm Acrylic Plate

Procedimiento:



Terminamos el brazo, uniendo ambas partes:
Componentes necesarios:



M3 Nickel Plated Nuts

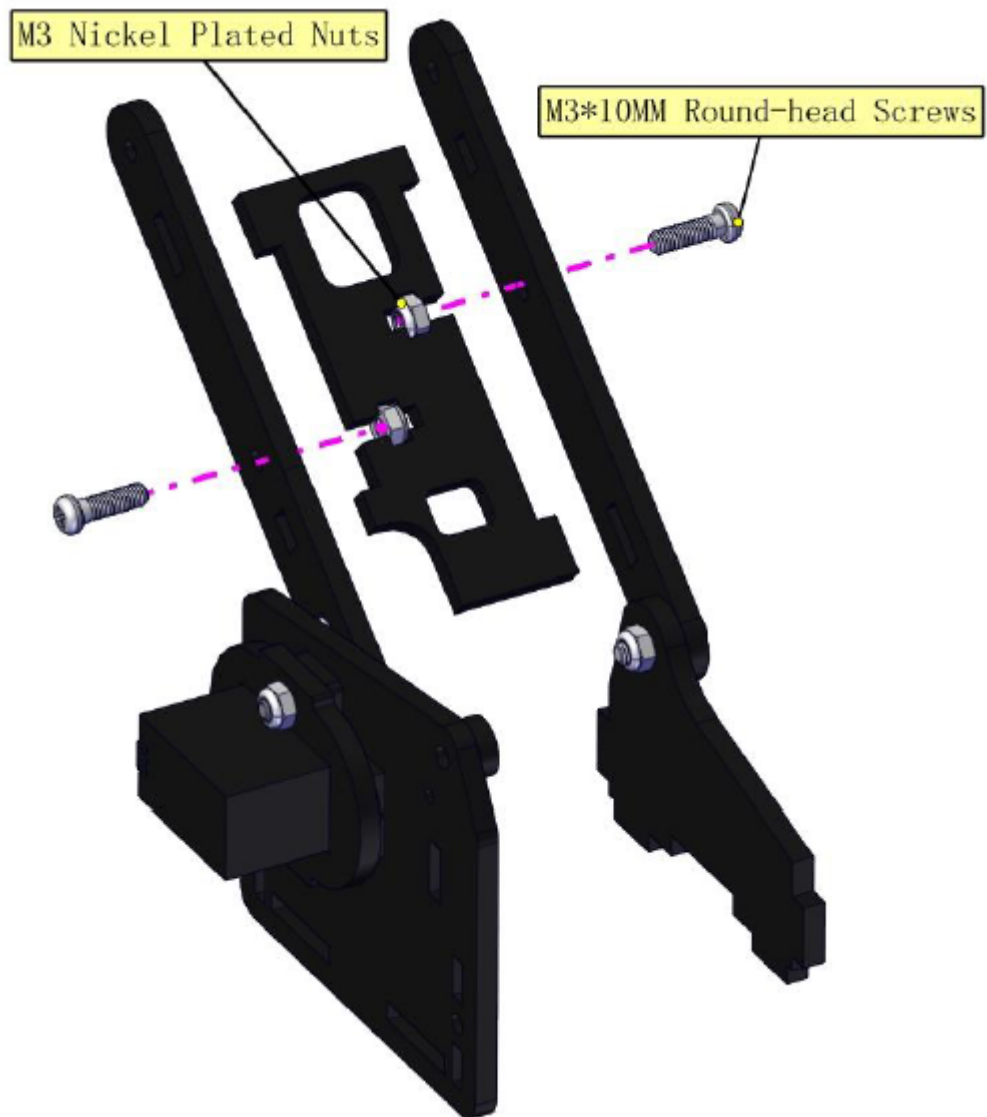
×2

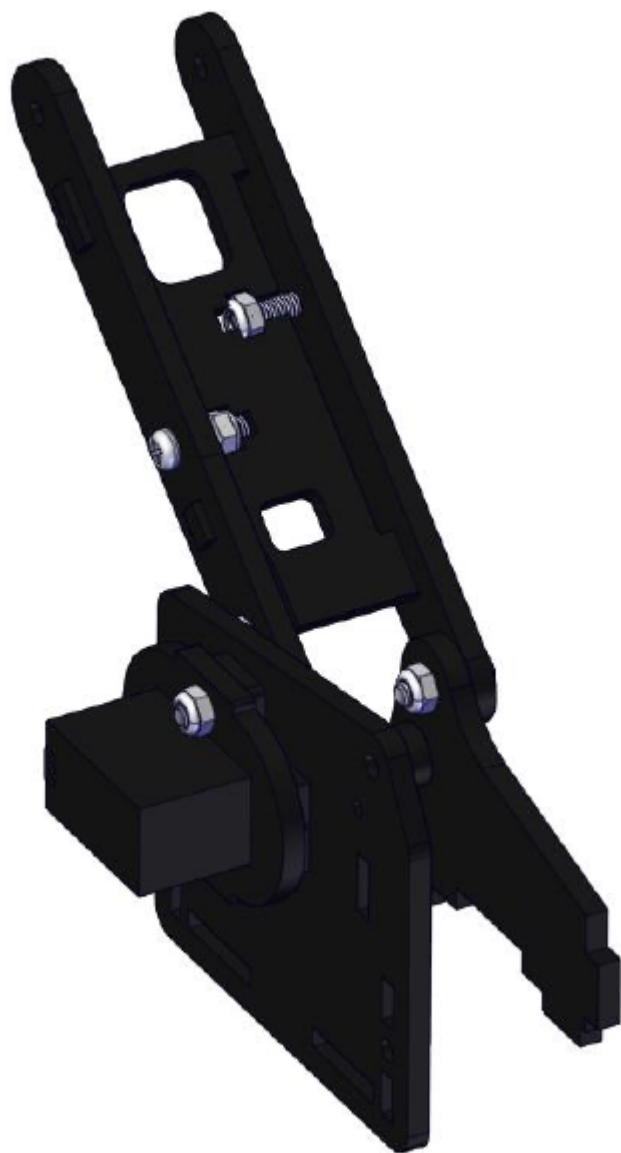


M3*10MM Round-head Screws

×2

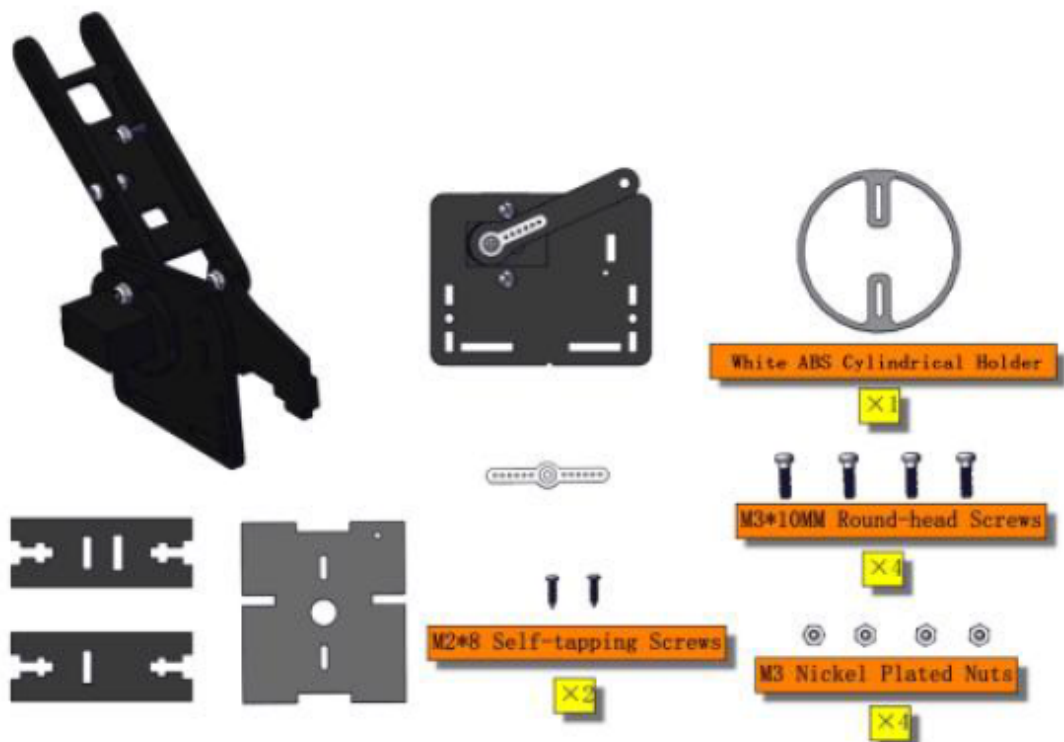
Procedimiento:



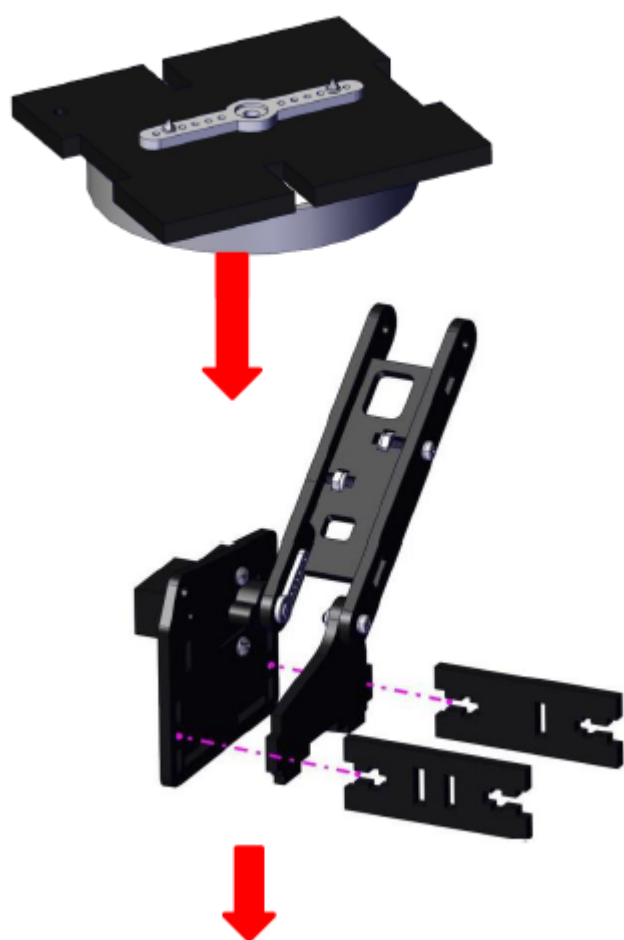
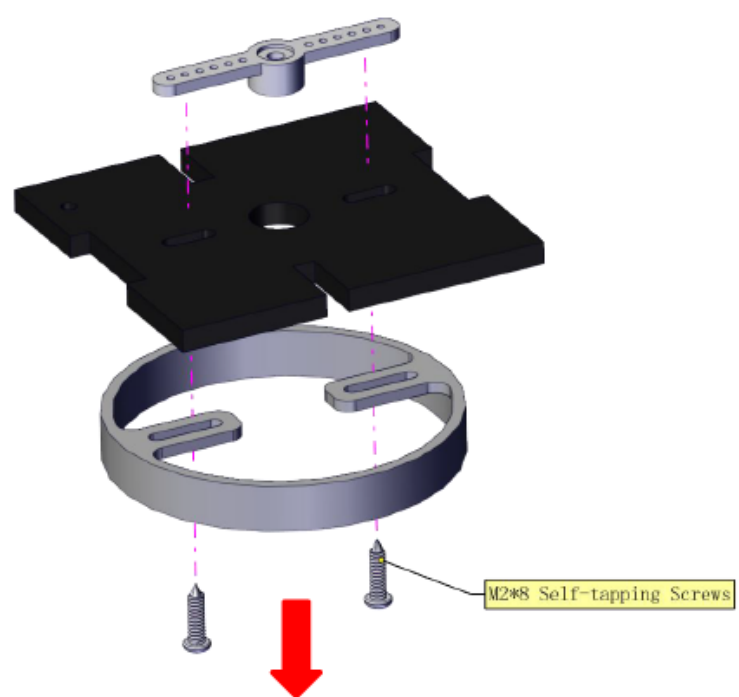


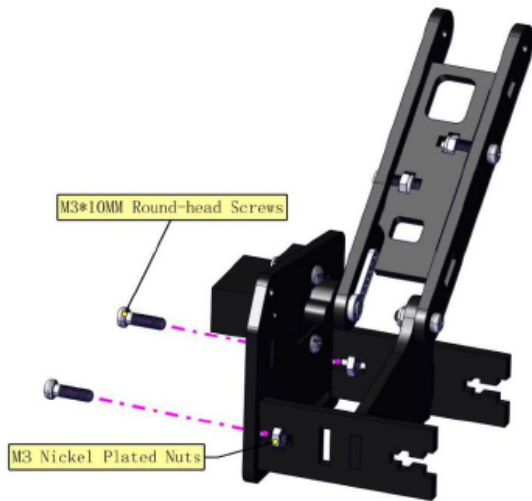
5. Integración de los brazos

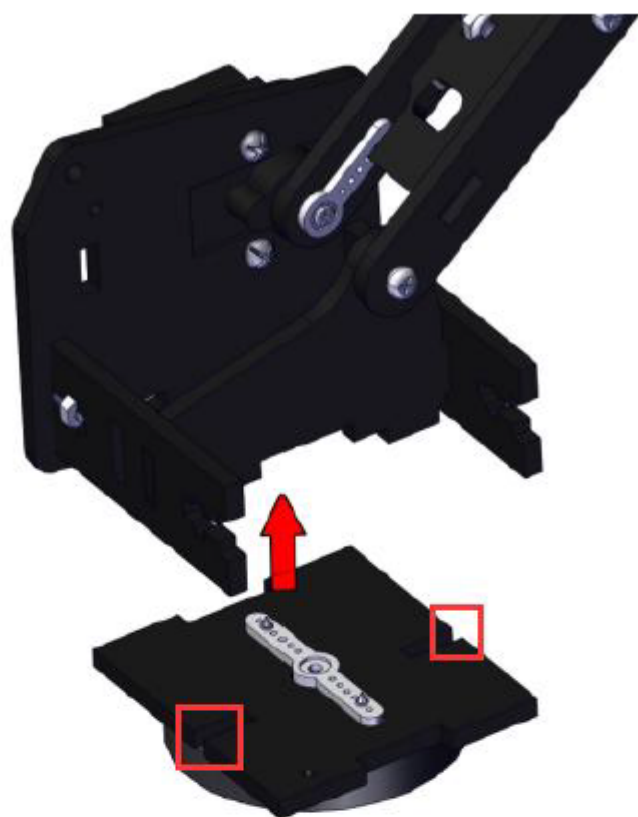
Componentes necesarios:

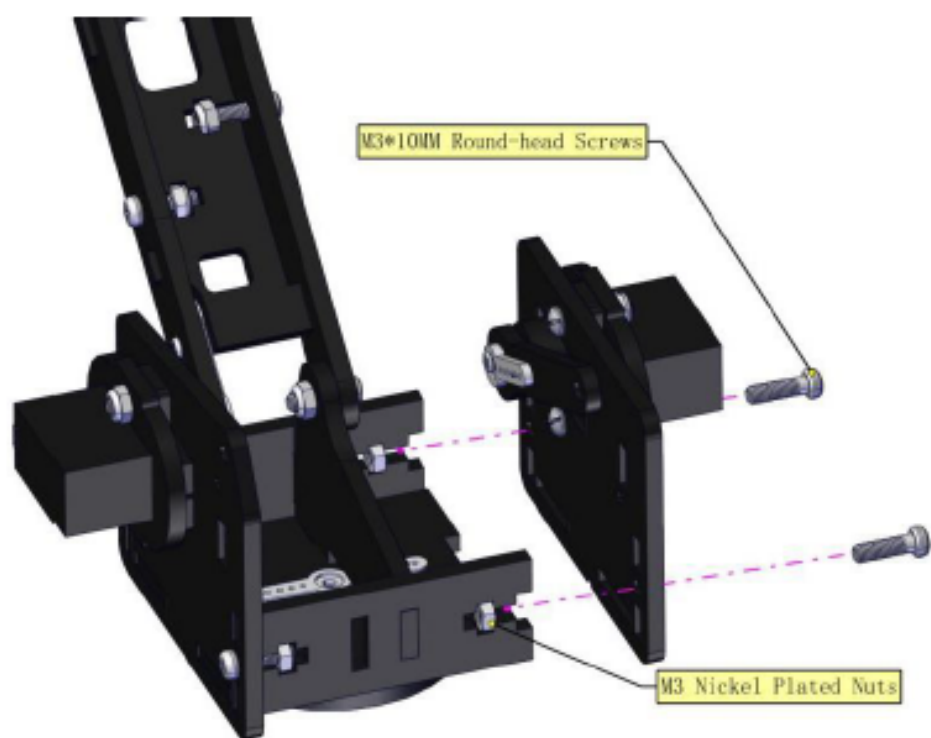


Procedimiento:

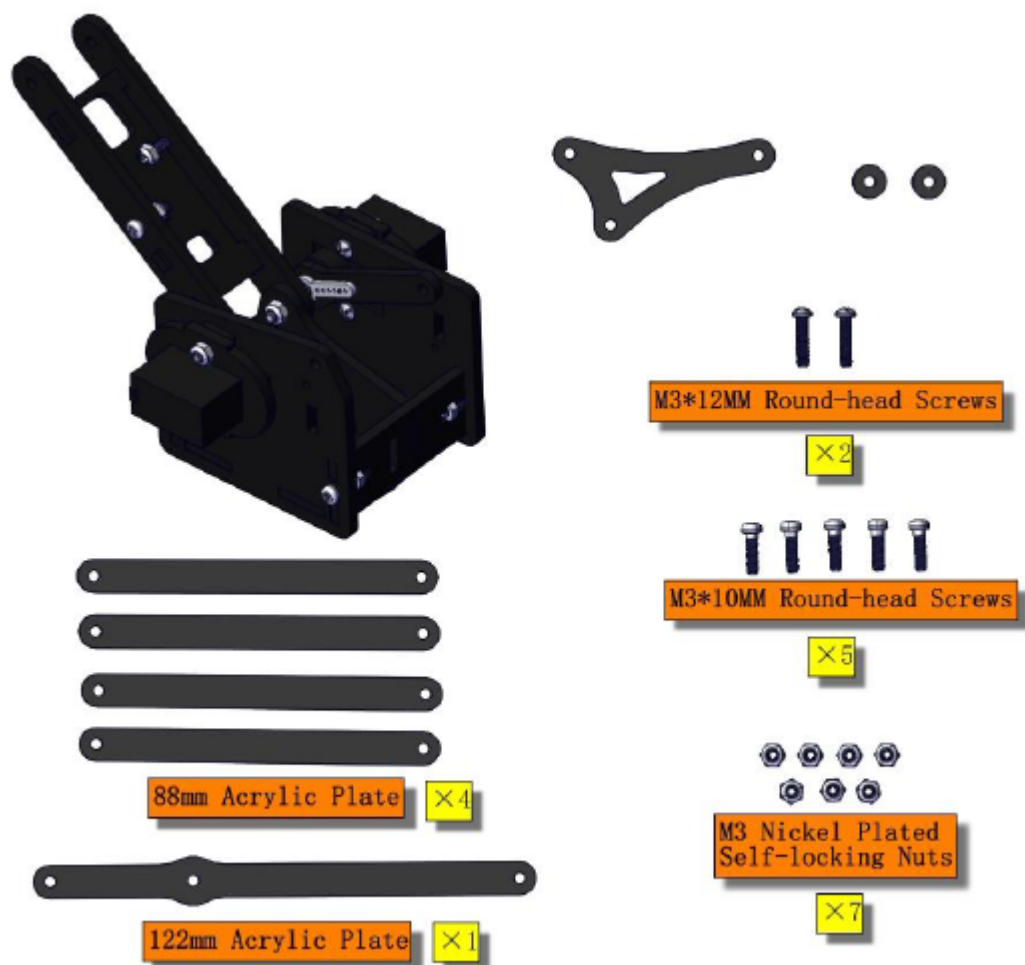






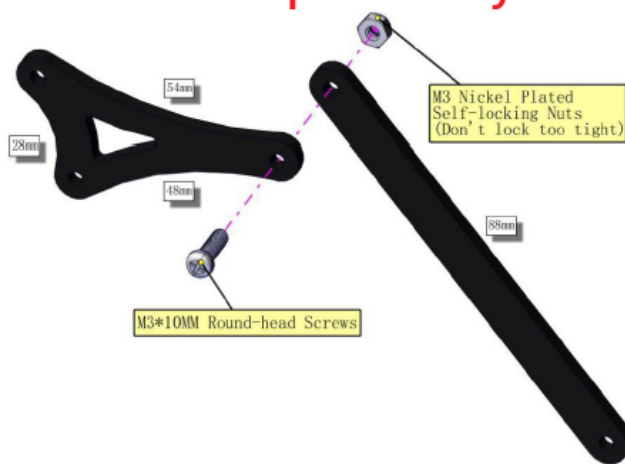


Terminamos el brazo con la última parte:
Componentes necesarios:

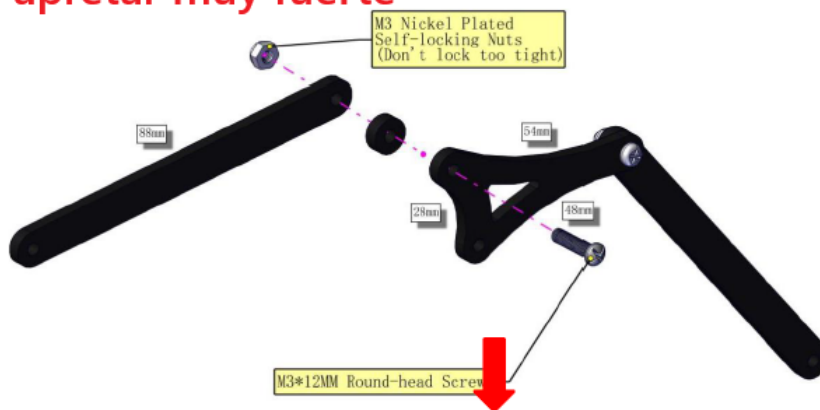


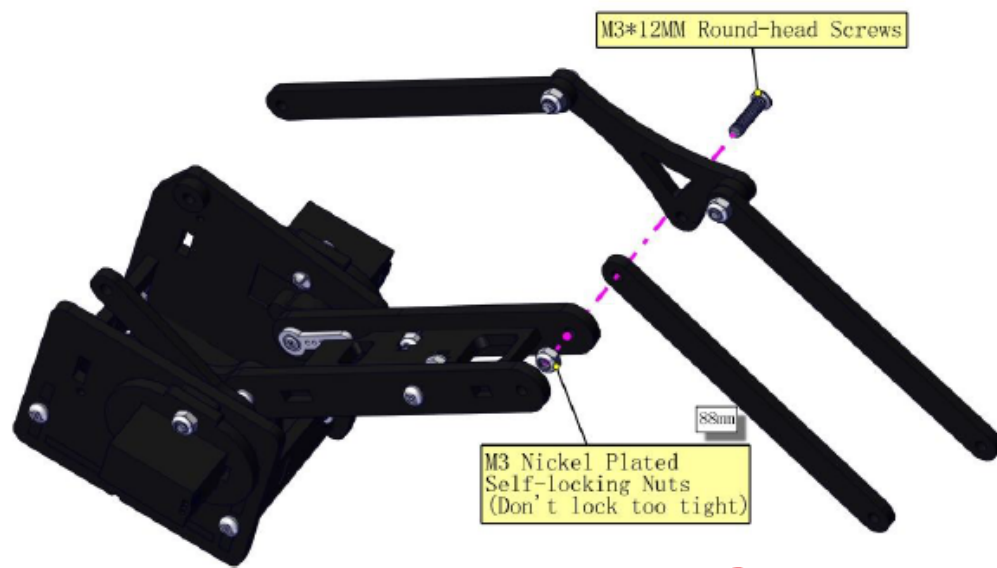
Procedimiento:

No apretar muy fuerte

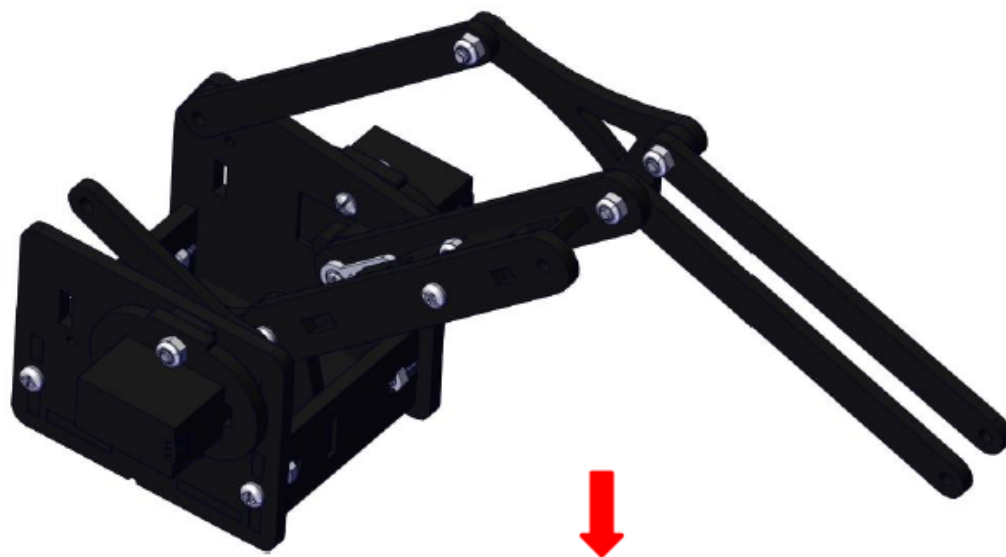


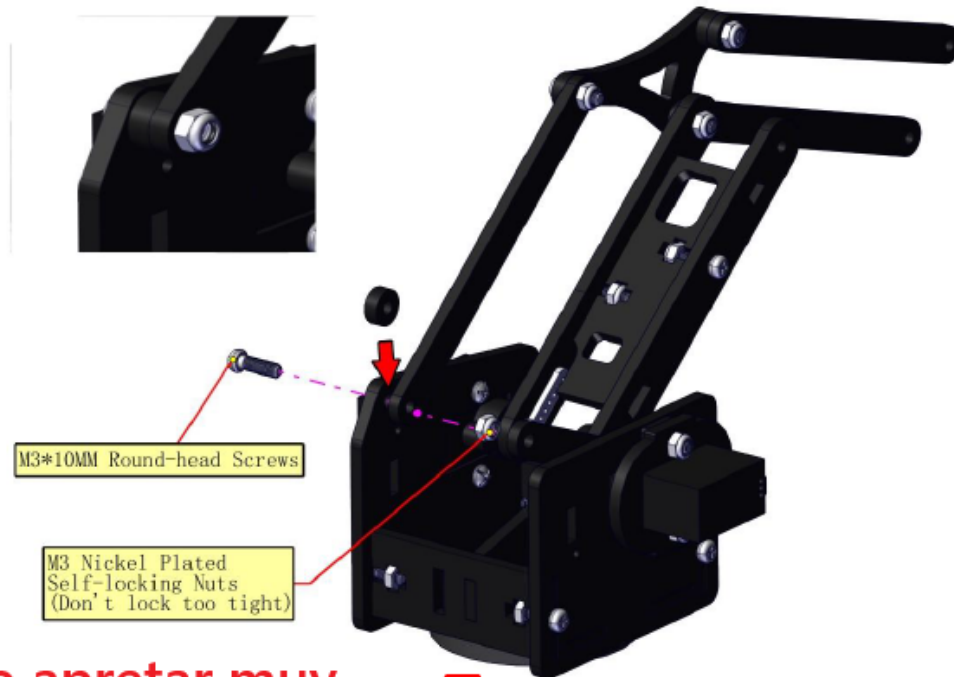
No apretar muy fuerte





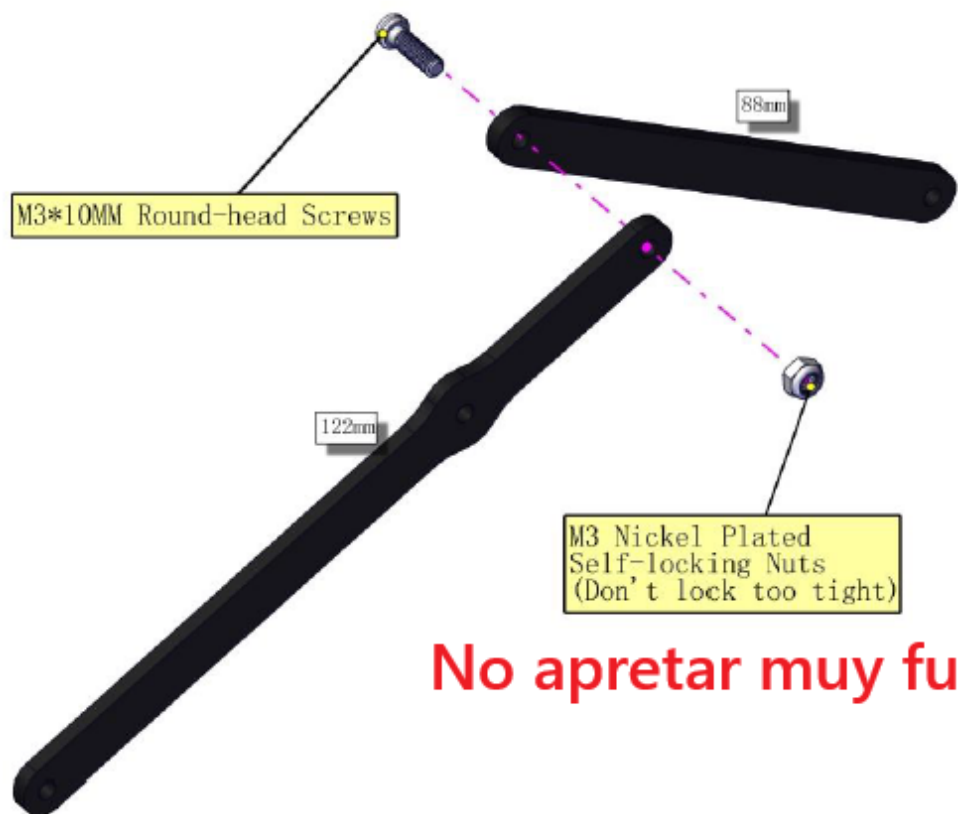
No apretar muy fuerte





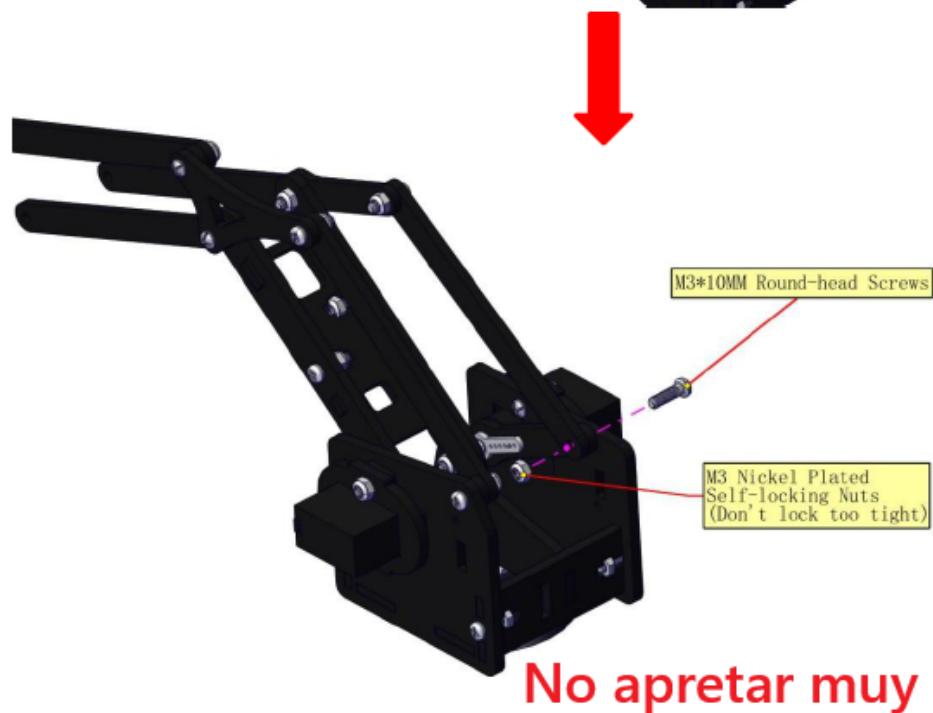
No apretar muy fuerte

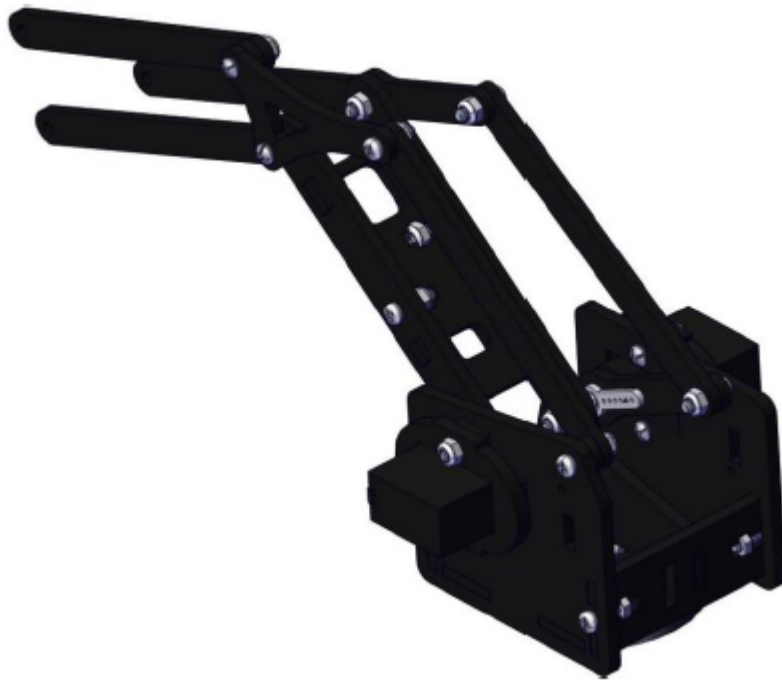




No apretar muy fuerte



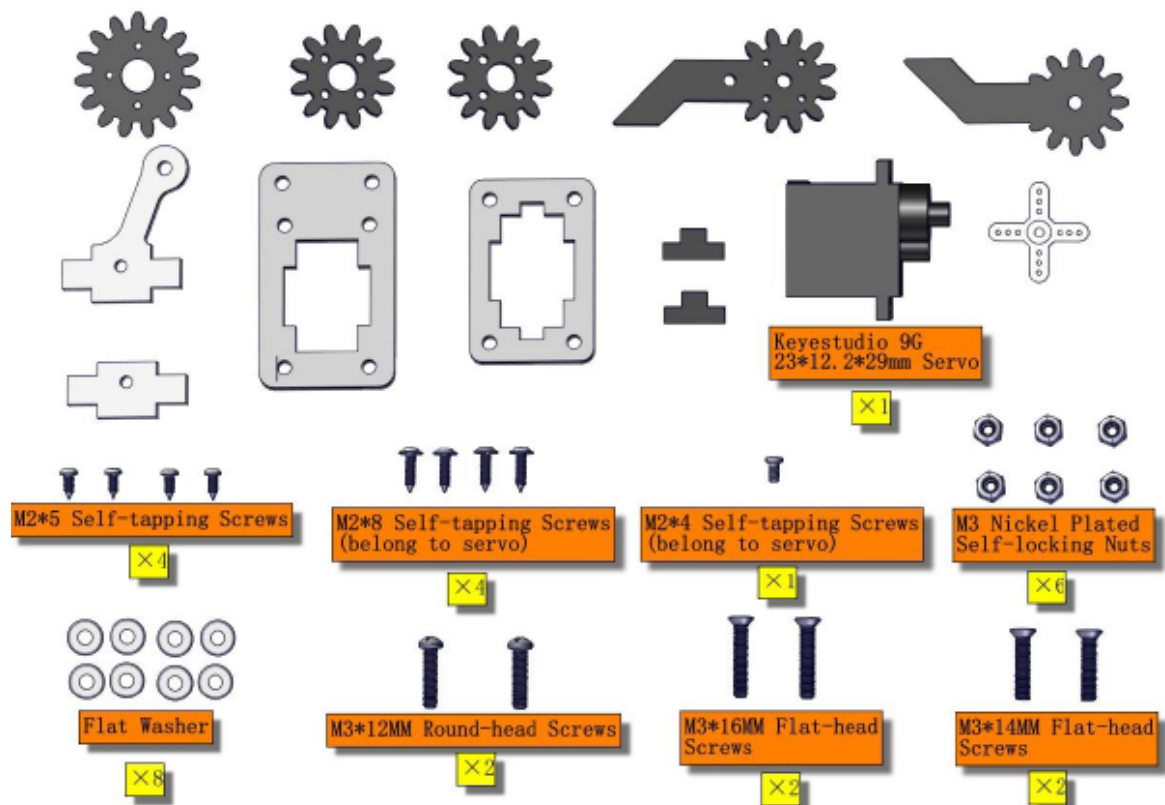




El brazo estaría terminado, faltando solo integrar la pinza.

6. Pinza (Servo de pinza)

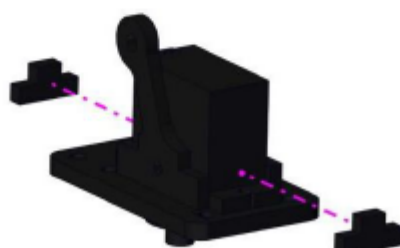
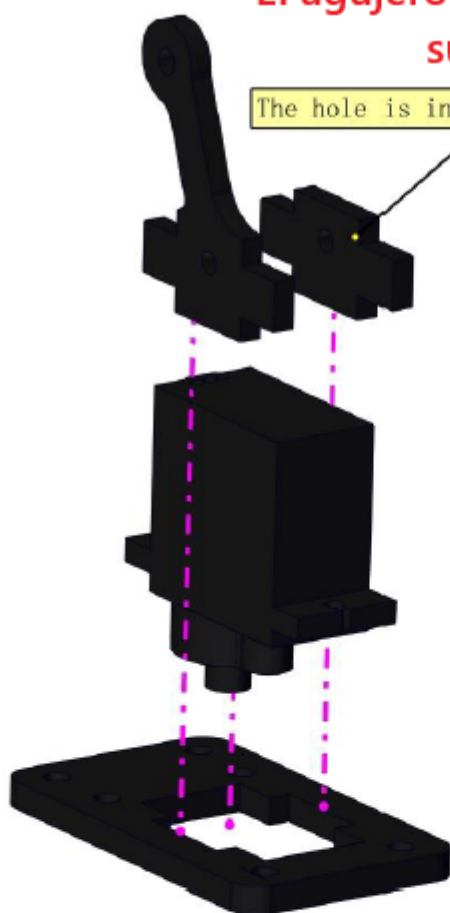
Componentes necesarios:

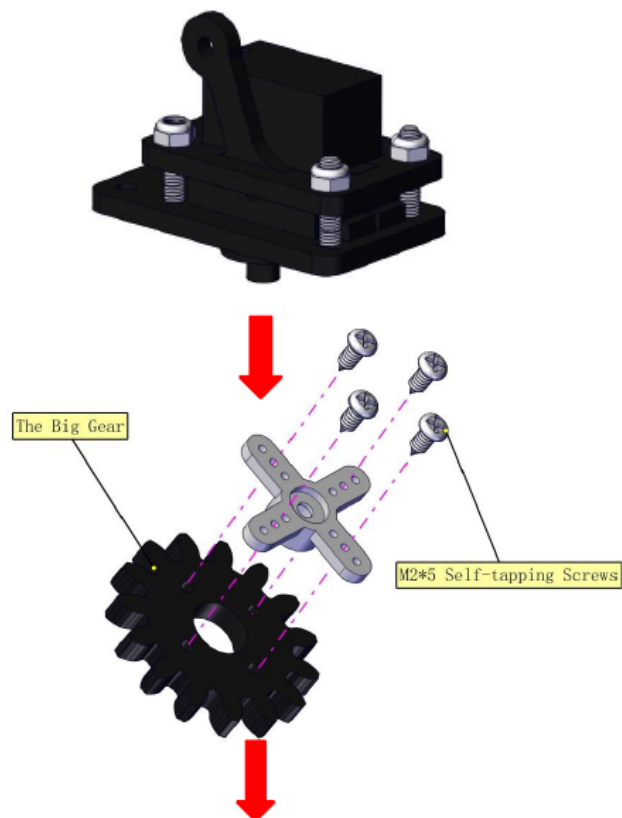
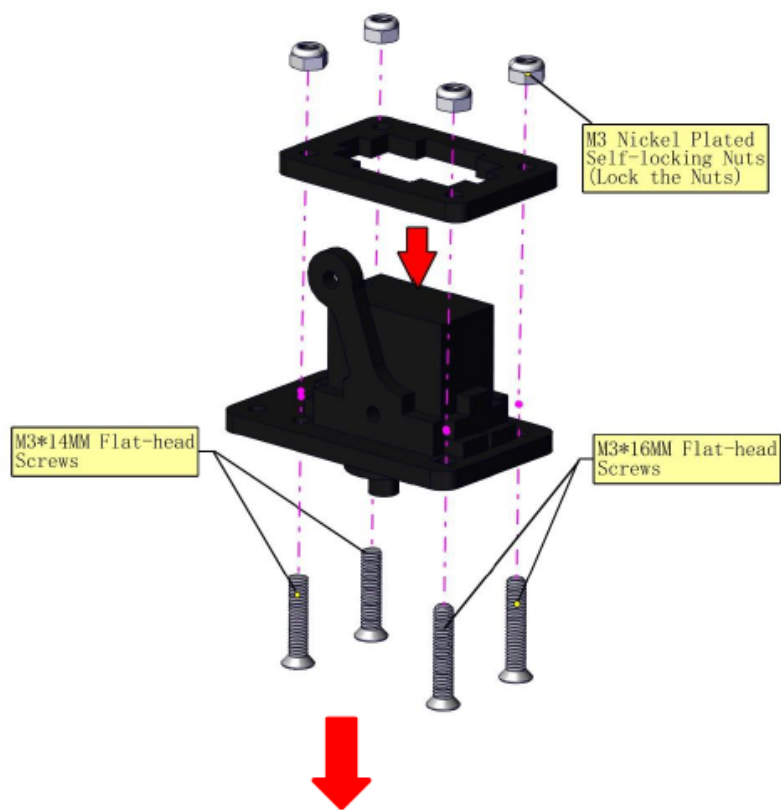


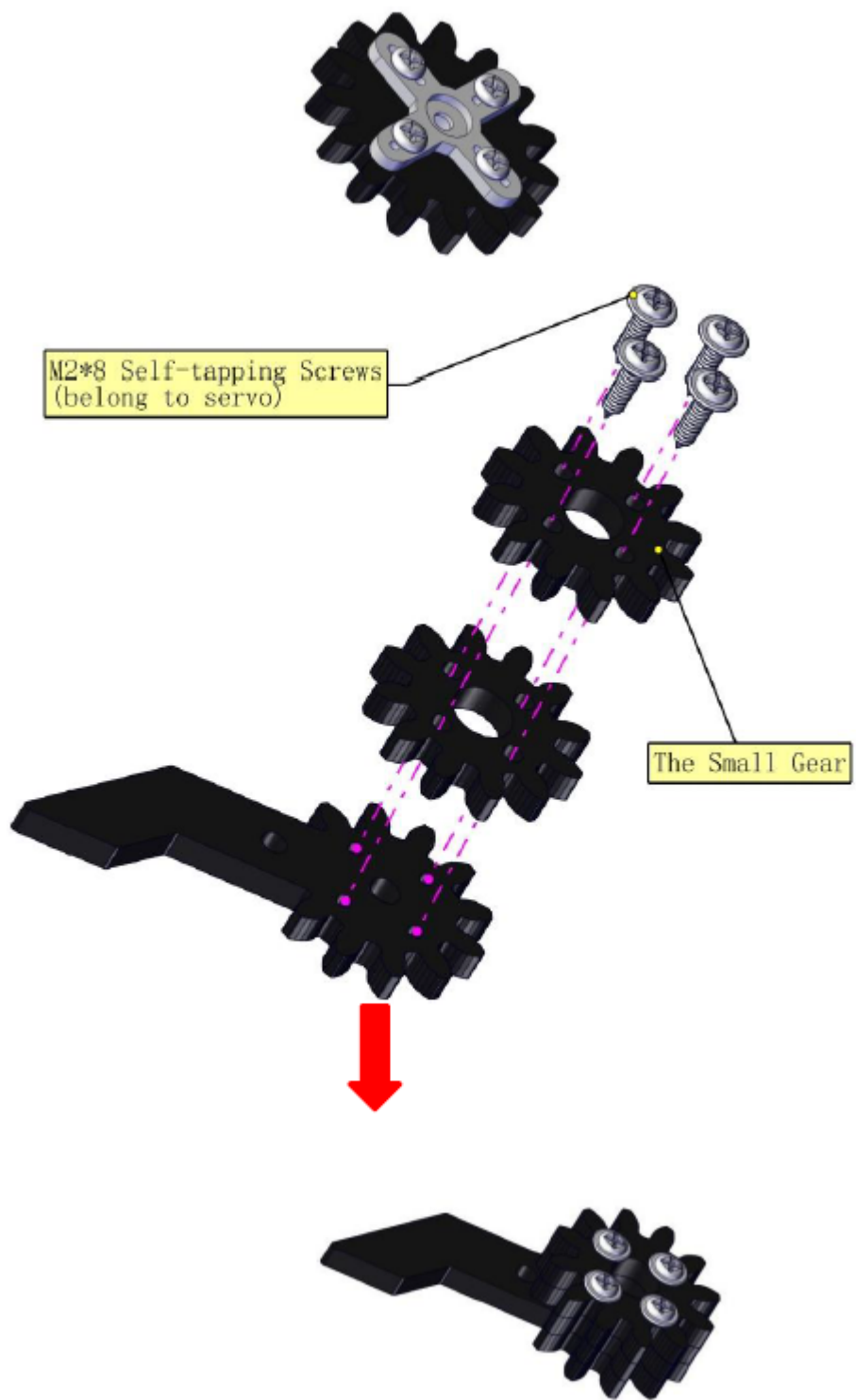
Procedimiento:

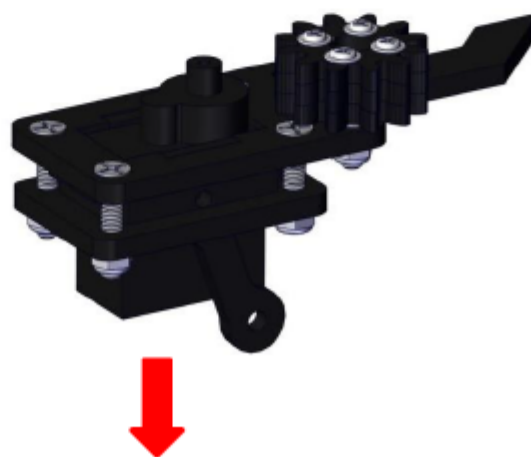
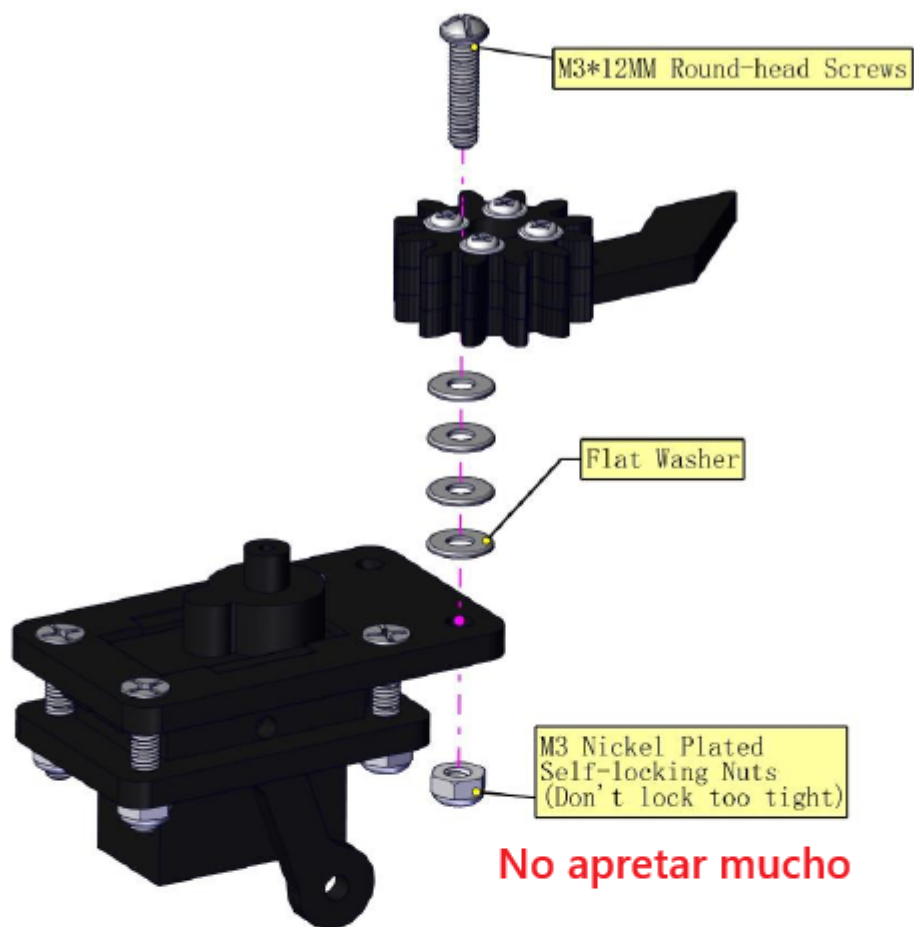
El agujero está en la parte superior

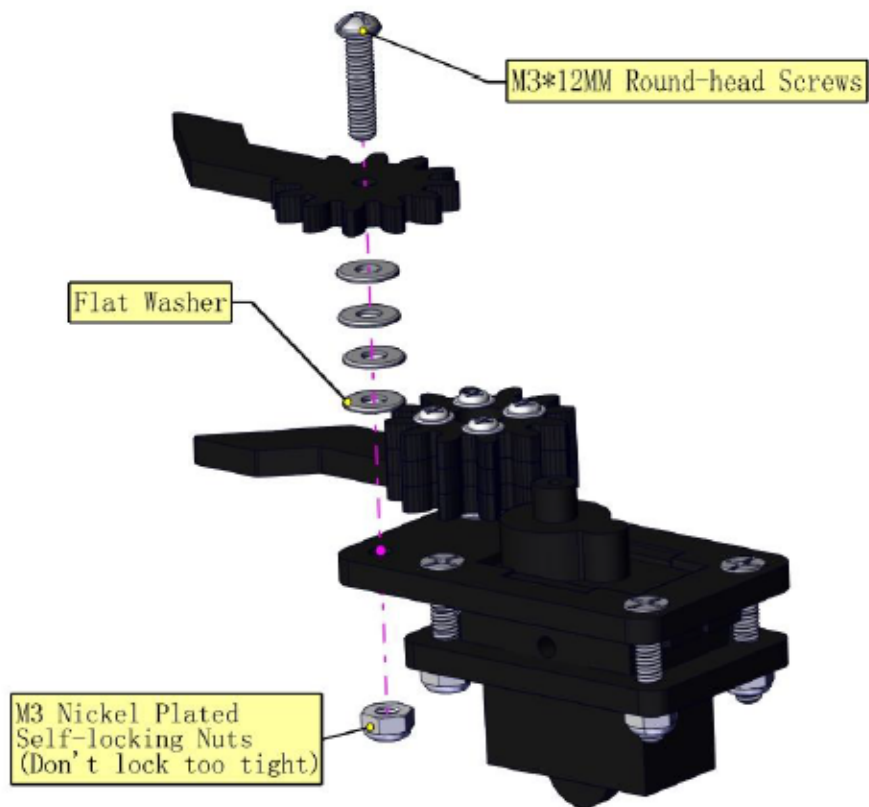
The hole is in an upward position







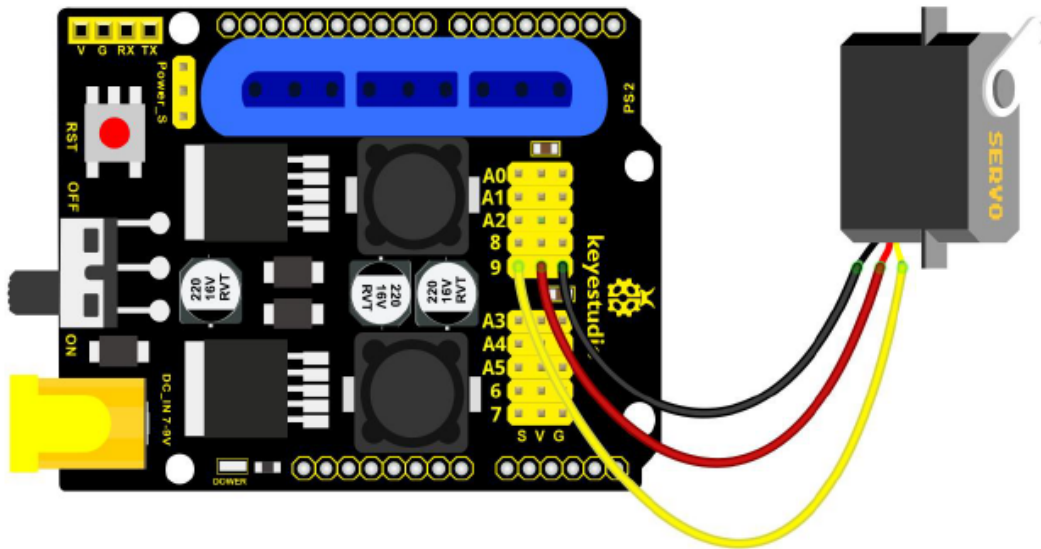




No apretar muy fuerte



Ahora que está el servo fijado, **rotamos a la posición de 0°**, conectándolo al **pin 9** tal que así:

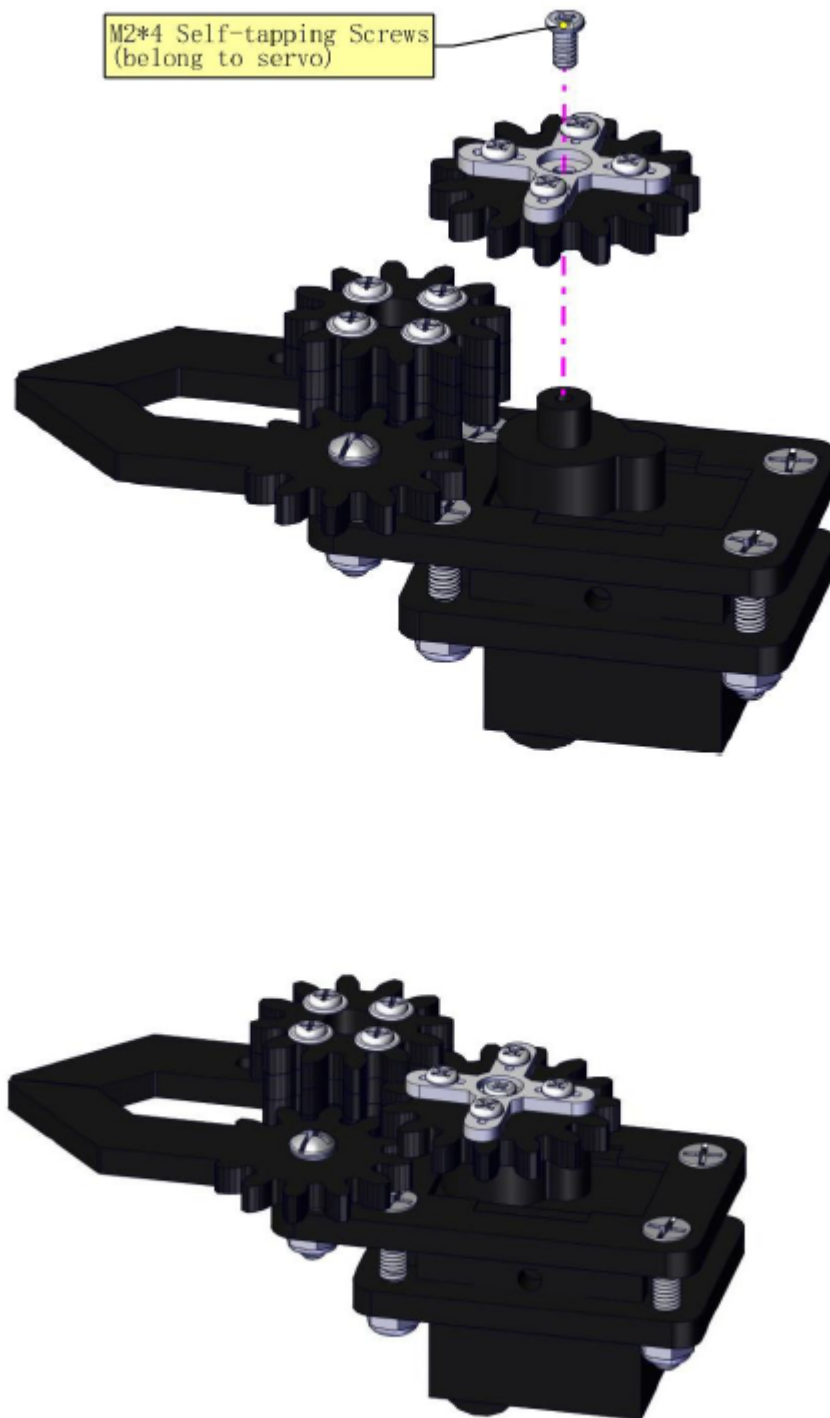


El código a subir:

```
#include <Servo.h>
Servo myservo; // Creación de objeto servo
void setup(){
  Serial.begin(9600);
  myservo.attach(9); // Configuración del pin 9
}
void loop(){
  myservo.write(0); // Ángulo al que se mueve (0)
  delay(1000);
}
```

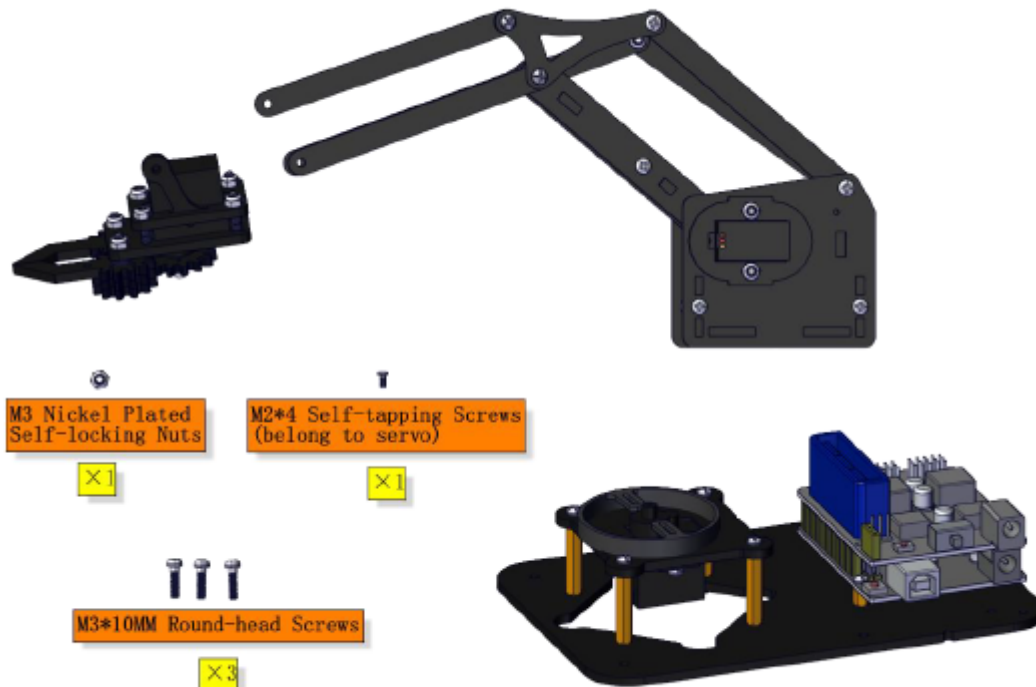
Una vez se ha rotado, seguimos con el montaje:

El tornillo es de la bolsa del servo



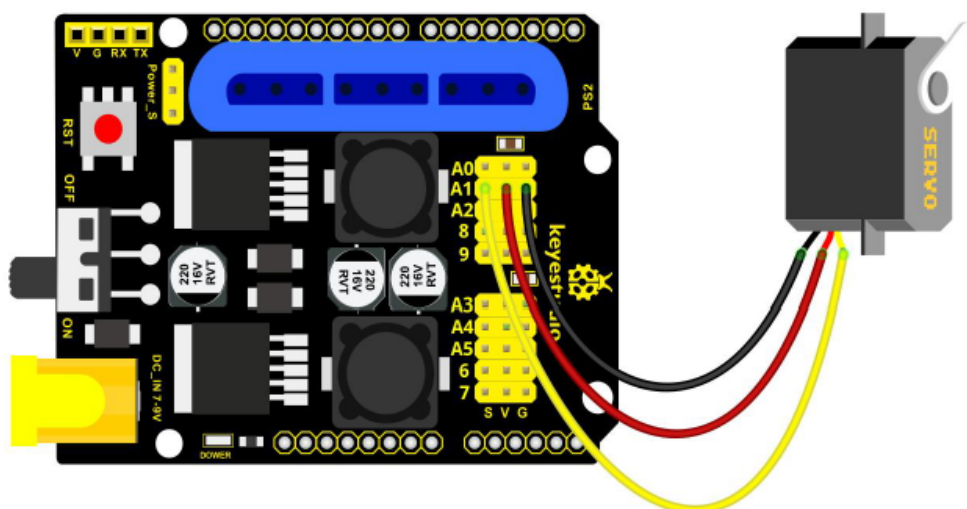
7. Finalización de brazo (pinza y base)

Componentes necesarios:



Procedimiento:

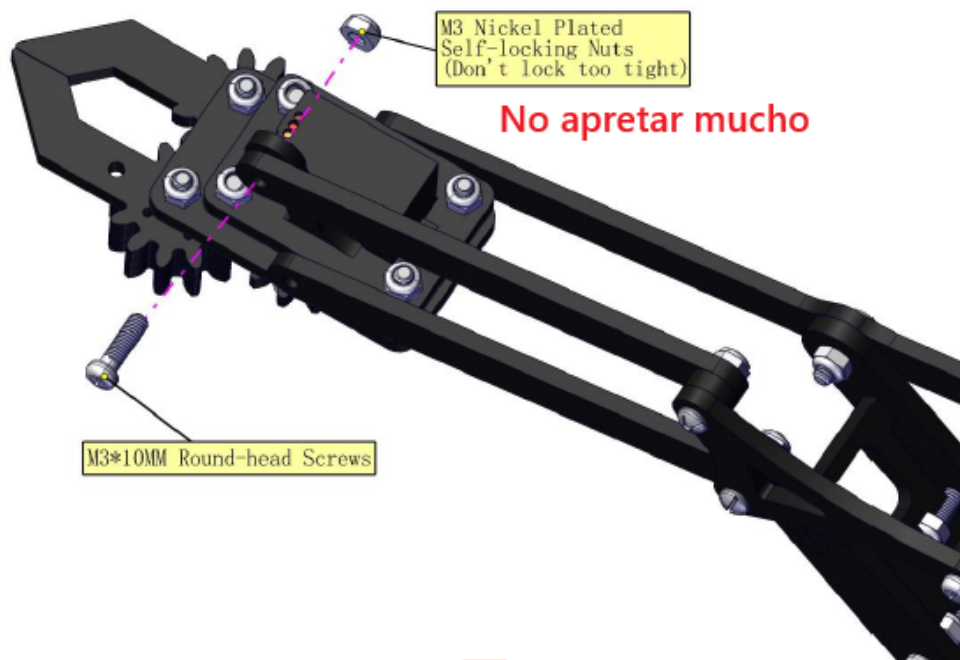
Vamos a poner la posición inicial del servo de la base, **rotando a la posición de 80°, conectándolo al pin A1** tal que así:

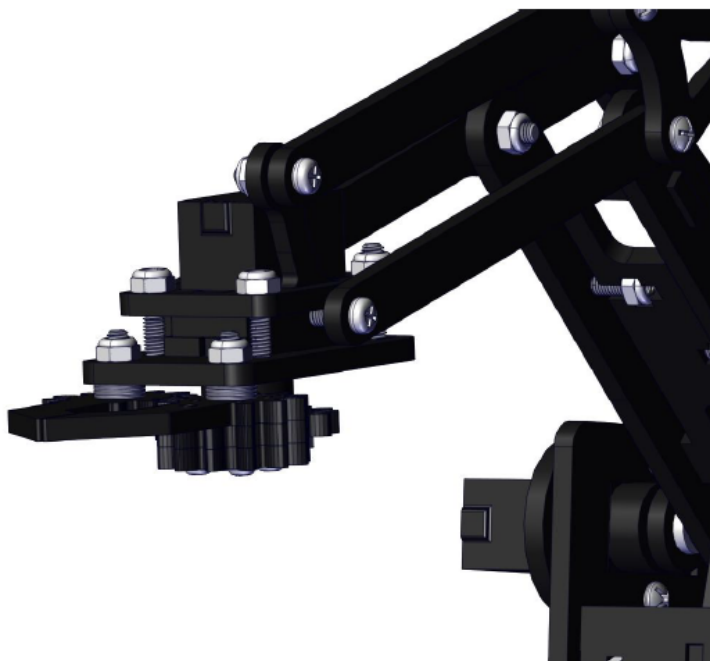
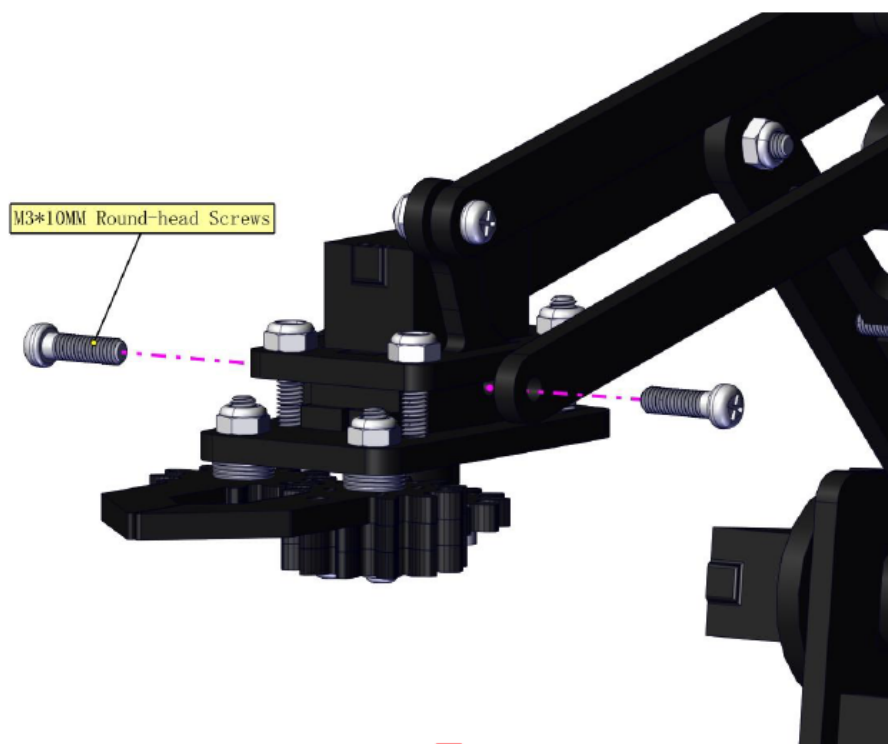


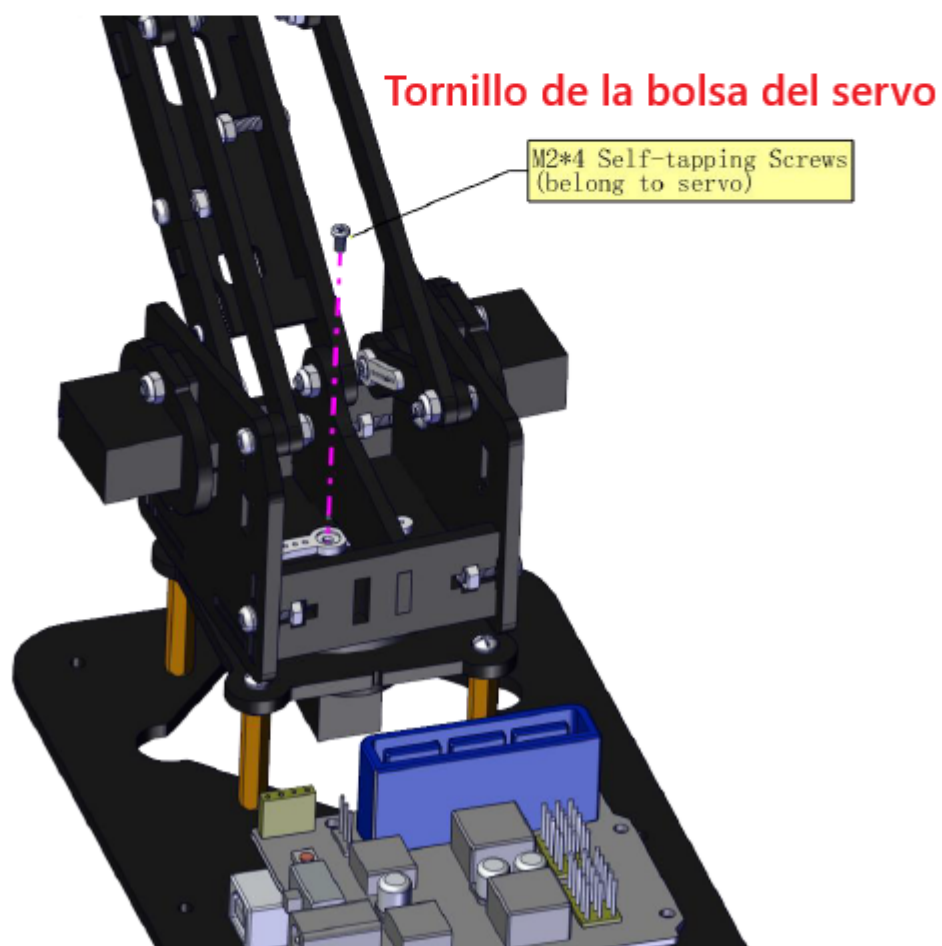
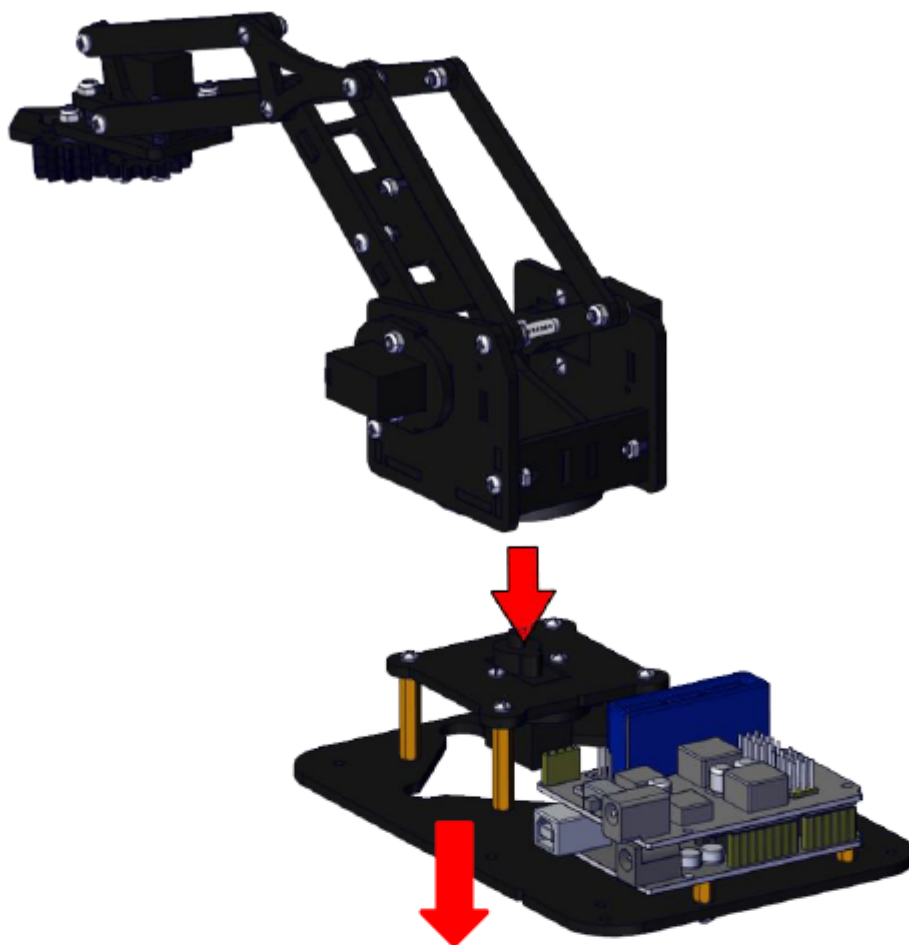
El código a subir:

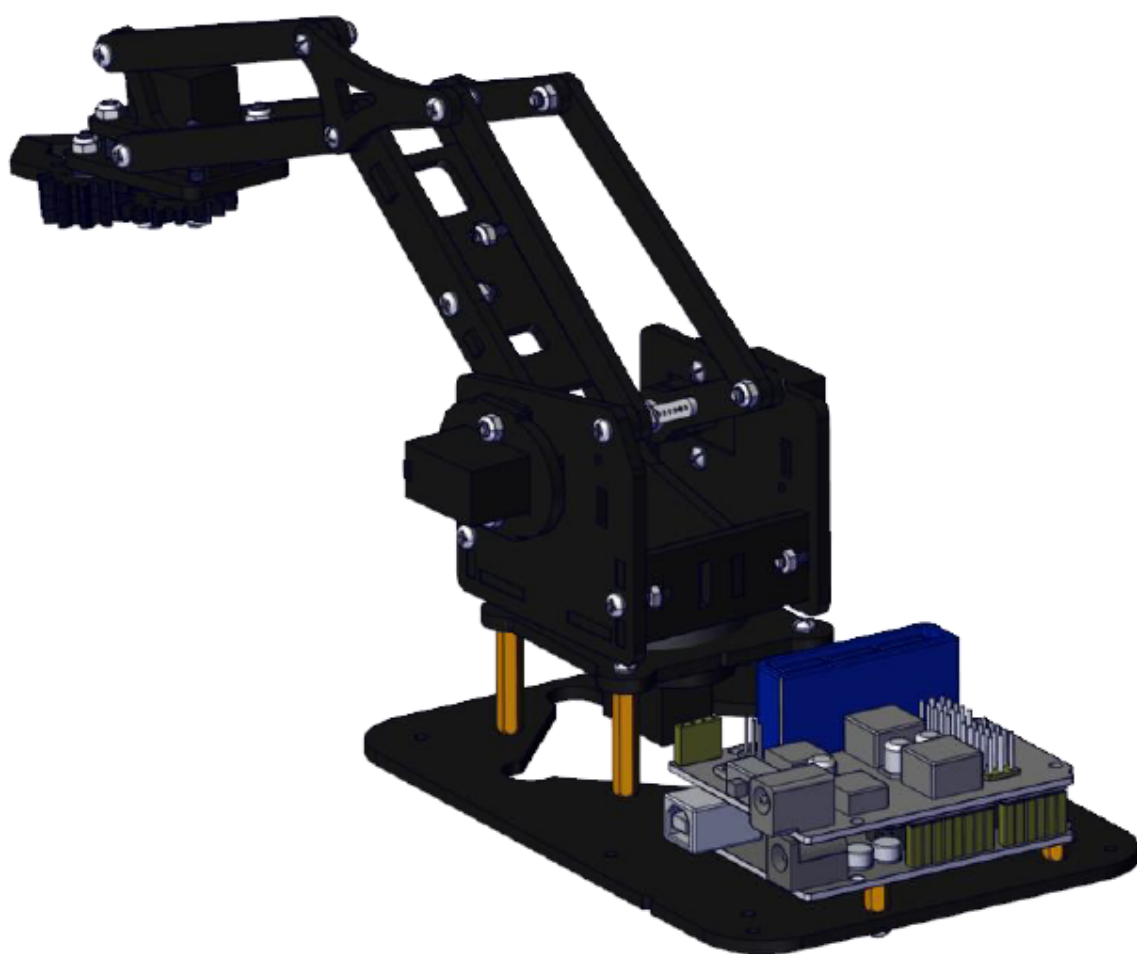
```
#include <Servo.h>
Servo myservo; // Creación de objeto servo
void setup(){
  Serial.begin(9600);
  myservo.attach(A1); // Configuración del pin A1
}
void loop(){
  myservo.write(80); // Ángulo al que se mueve (80)
  delay(1000);
}
```

Una vez se ha rotado, seguimos con el montaje:





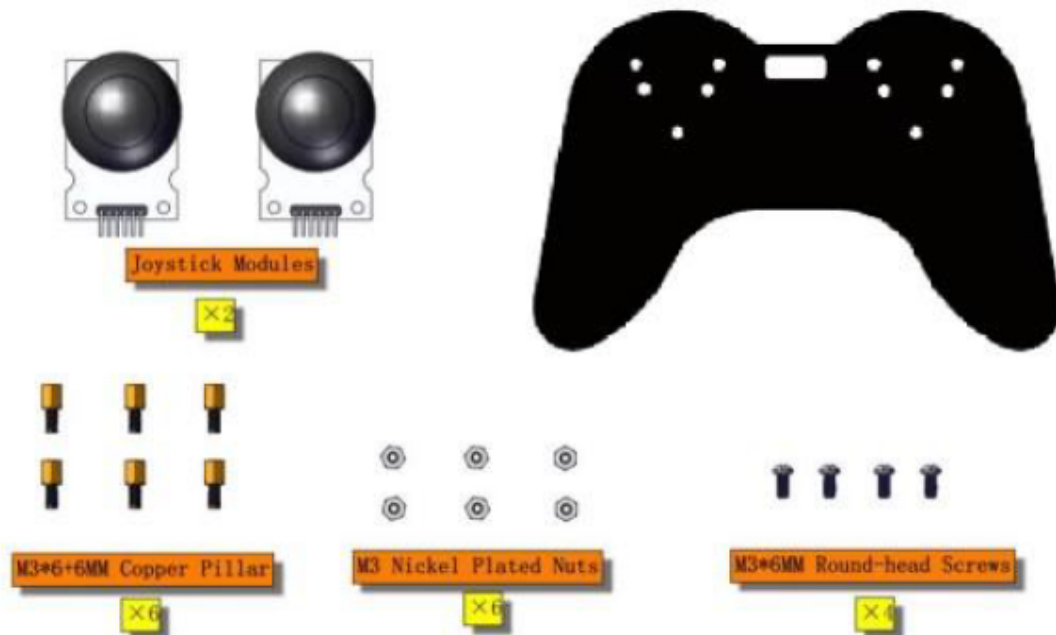




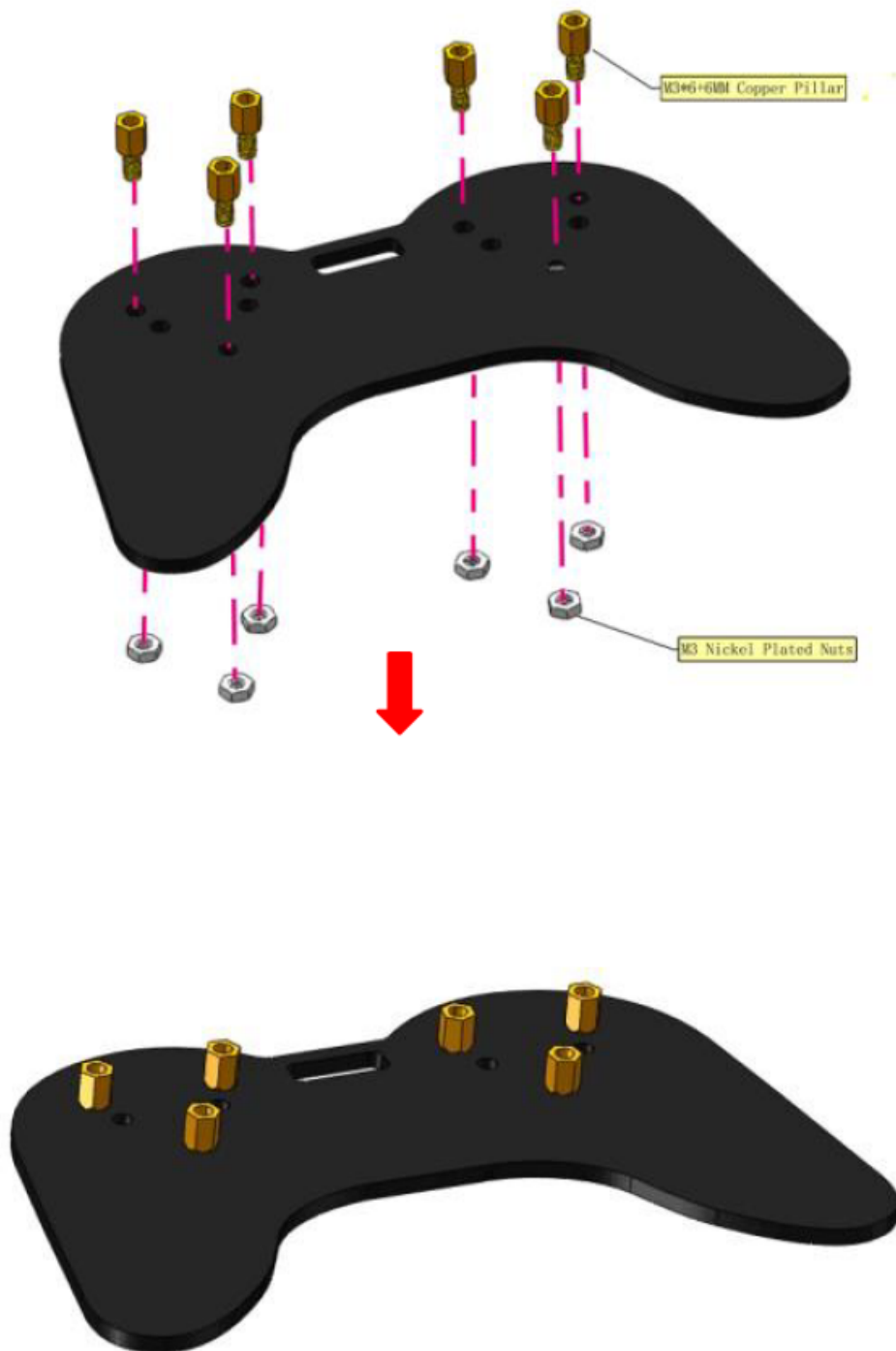
Brazo robótico montado.

8. Mando joysticks

Componentes necesarios:



Procedimiento:



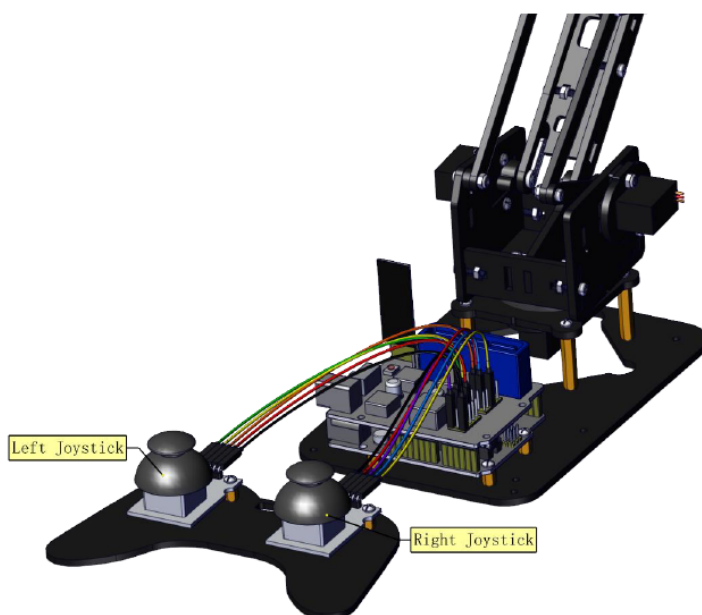
M3*6MM Round-head Screws



9. Conexionado

Mando:

Joystick derecho	Arduino
X	A2
Y	A5
B	7
V	V
G	G
Joystick izquierdo	Arduino
X	A3
Y	A4
B	8
V	V
G	G



Servos:

Rojo -> V

Marrón -> G

Naranja ->

	Pin (S)
Servo base (1)	A1
Servo derecho (2)	A0
Servo izquierdo (3)	6
Servo pinza (4)	9