

## Control de brazo robótico por Bluetooth (Compatible solo con dispositivos Android)

Se realizará el control de un brazo robótico mediante el módulo bluetooth, el brazo está conformado por 5 motores.

### Objetivo General.

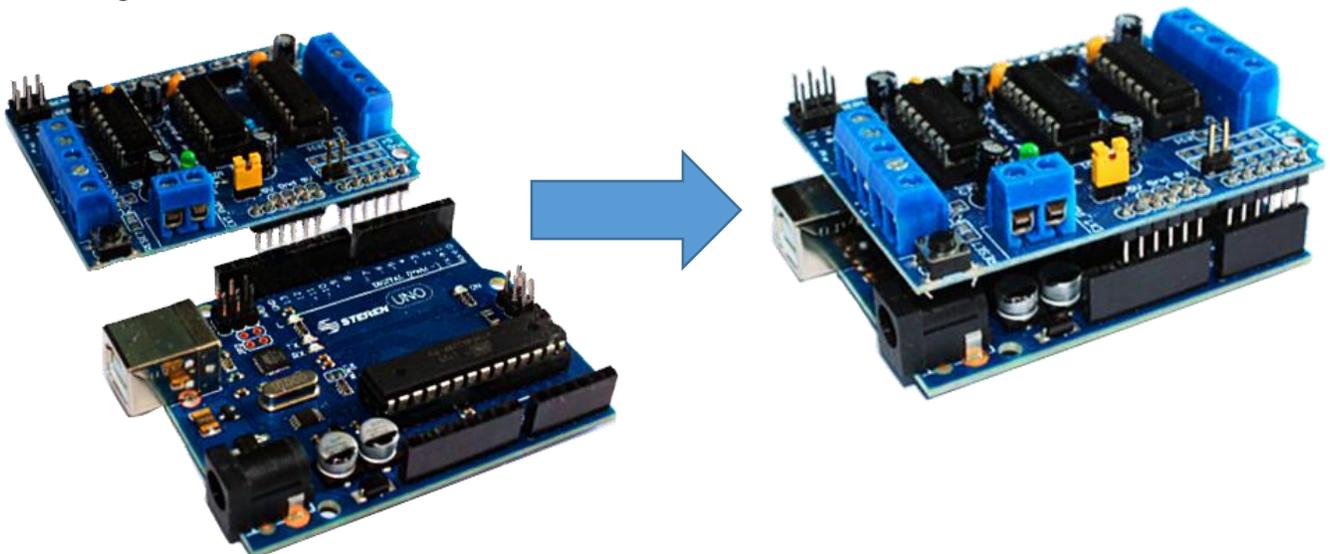
Desarrollar un programa que nos permita controlar el brazo mediante bluetooth haciéndole llegar señales al Arduino, y desde éste, enviar la señal a los motores.

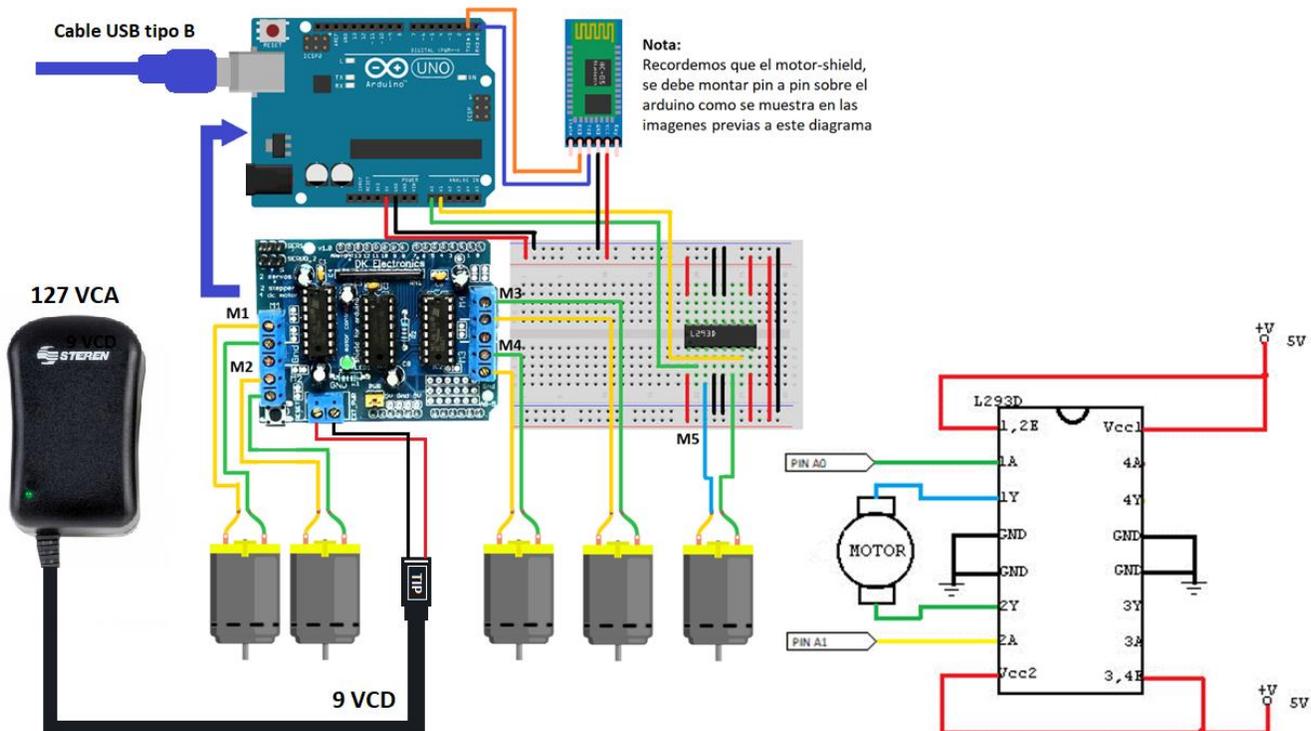
### Desarrollo de la actividad práctica

1. Realizar código de control con Bluetooth y utilizar librerías para el controlador de motores
2. Hacer Conexión del controlador de motores y Bluetooth en la Protoboard
3. Cargar el Programa a la Placa Arduino UNO
4. Descargar de Play Store la App para el generar los botones que manden la señal Bluetooth
5. Comprobar funcionamiento de componentes y placa Arduino
6. Comprobar funcionamiento del brazo

Materiales	
Cantidad	Modelo
1	K-680
1	ARD-010
1	ARD-340
1	ARD-310
1	ARD-305
1	L293D
1	509-010
1	250-687
1	ELI-1200
1	USB-490

### Diagrama de Conexión





**Nota del eliminador**

En el plug del eliminador ELI-1200, considerar la polaridad especificada en el diagrama, ya que orientando el letrero "TIP" hacia nosotros, el polo Positivo (+) estará a la derecha y el Negativo (-) a la izquierda

**Breve Descripción del L293D**

El integrado L293D incluye cuatro circuitos para manejar cargas de potencia media, en especial pequeños motores y cargas inductivas, con la capacidad de controlar corriente hasta 600mA en cada circuito y una tensión entre 4,5V a 36V.

La tarjeta para control de motores tiene la función de controlar cuatro motores en una sola placa con la finalidad de tener un menor conexionado haciéndolo más sencillo y menos robusto para nuestro proyecto.

Para esta práctica se utilizaron cinco motores, por lo que fue requerido un circuito integrado L293D adicional a la tarjeta para controlar el quinto motor.

## Código

```
#include <AFMotor.h>//Librería para trabajar la shield de control de motores
//asignacion de motores a controlar.
AF_DCMotor motor1 (1);
AF_DCMotor motor2 (2);
AF_DCMotor motor3 (3);
AF_DCMotor motor4 (4);
//declaración de variables globales.
int pinza1=14;
int pinza2=15;
char Val;

void setup() {

    Serial.begin(38400);//iniciamos una comunicación serial.
    //Declarar velocidad con la que trabajara el motor
    motor1.setSpeed(255);
    motor2.setSpeed(255);
    motor3.setSpeed(255);
    motor4.setSpeed(255);
    //Declaracion de pines como salida
    pinMode(pinza1, OUTPUT);
    pinMode(pinza2, OUTPUT);
}

void loop() {

    if (Serial.available() > 0) { //mandar a iniciar la conexión serial
        Val = Serial.read();
        switch (Val) { //utilizar switch case para cuando reciba una variable en el
            serial se direcciona a un caso
            //caso para girar el motor 1 hacia adelante
            case '1':
                Adelantem1();
                break;
            //caso para girar el motor 1 hacia la derecha
            case '2':
                Atrasml();
                break;
            //caso para girar el motor 2 hacia la izquierda
            case '3':
                Adelantem2();
                break;
            //caso para girar el motor 2 hacia atrás
            case '4':
                Atrasml2();
                break;
            //caso para girar el motor 3 hacia adelante
            case '5':
                Adelantem3();
                break;
            //caso para girar el motor 3 hacia atrás
            case '6':
                Atrasml3();
                break;
            //caso para girar el motor 4 hacia adelante
            case '7':
                Adelantem4();
                break;
            //caso para girar el motor 4 hacia atrás
            case '8':
                Atrasml4();
                break;
        }
    }
}
```

```

        break;
//caso para girar el motor 5 y cerrar la pinza
    case '9':
        cerrar();
        break;
//caso para girar el motor 5 y abrir la pinza
    case 'A':
        abrir();
        break;
//caso para detener los motores
    case 'B':
        STOP();
        break;
    }
}
loop();
}
//girar el motor 1 hacia la derecha
void Adelantem1() {
    motor1.run(FORWARD);
}
//girar el motor 1 hacia la izquierda
void Atrasm1() {
    motor1.run(BACKWARD);
}
//girar el motor 2 hacia adelante
void Adelantem2() {
    motor2.run(FORWARD);
}
//girar el motor 2 hacia atrás
void Atrasm2() {
    motor2.run(BACKWARD);
}
//girar el motor 3 hacia adelante
void Adelantem3() {
    motor3.run(FORWARD);
}
//girar el motor 3 hacia atrás
void Atrasm3() {
    motor3.run(BACKWARD);
}
//girar el motor 4 hacia adelante
void Adelantem4() {
    motor4.run(FORWARD);
}
//girar el motor 4 hacia atrás
void Atrasm4() {
    motor4.run(BACKWARD);
}
//girar el motor 5 y cerrar la pinza
void cerrar() {
    digitalWrite(pinza1, HIGH);
    digitalWrite(pinza2, LOW);
}
//girar el motor 5 y abrir la pinza
void abrir() {
    digitalWrite(pinza1, LOW);
    digitalWrite(pinza2, HIGH);
}
//detener todos los motores
void STOP() {
    motor1.run(RELEASE);
    motor2.run(RELEASE);
}

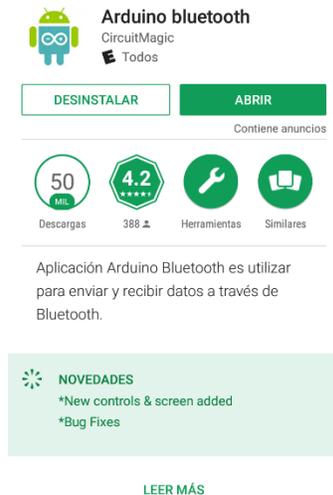
```

```

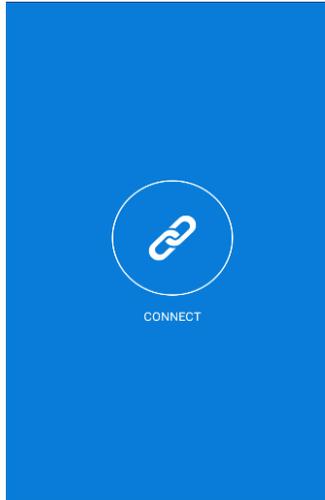
motor3.run(RELEASE);
motor4.run(RELEASE);
digitalWrite(pinza1, LOW);
digitalWrite(pinza2, LOW);

```

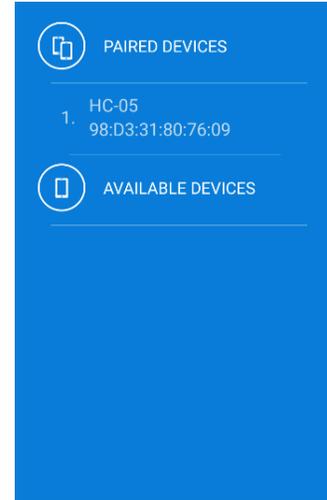
## Descarga de la App para el control Bluetooth



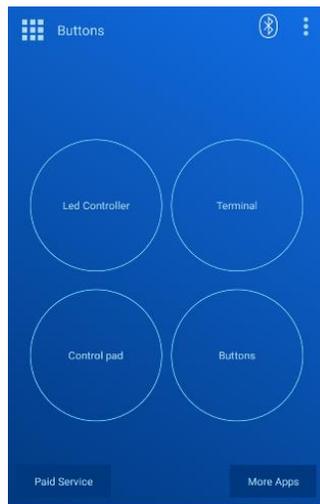
1.- Descarga la App "Arduino Bluetooth"



2.- Abre la App y conecta



3.- Selecciona el módulo HC-05



4.- Selecciona la opción "Buttons"

Button label	On Release	On Press
IzquierdaM1	B	1
DerechaM1	B	2
SubirM2	B	4
BajarM2	B	3
SubirM3	B	6
BajarM3	B	5
SubirM4	B	8
BajarM4	B	7
Abrir	B	A
Cerrar	B	9

**Save**

5.- Configura los botones como se muestra



6.- El control BT está listo para trabajar