

Control de movimientos y rutinas para Mini Brazo Robótico por Bluetooth

(Compatible solo con dispositivos Android)

Objetivo General.

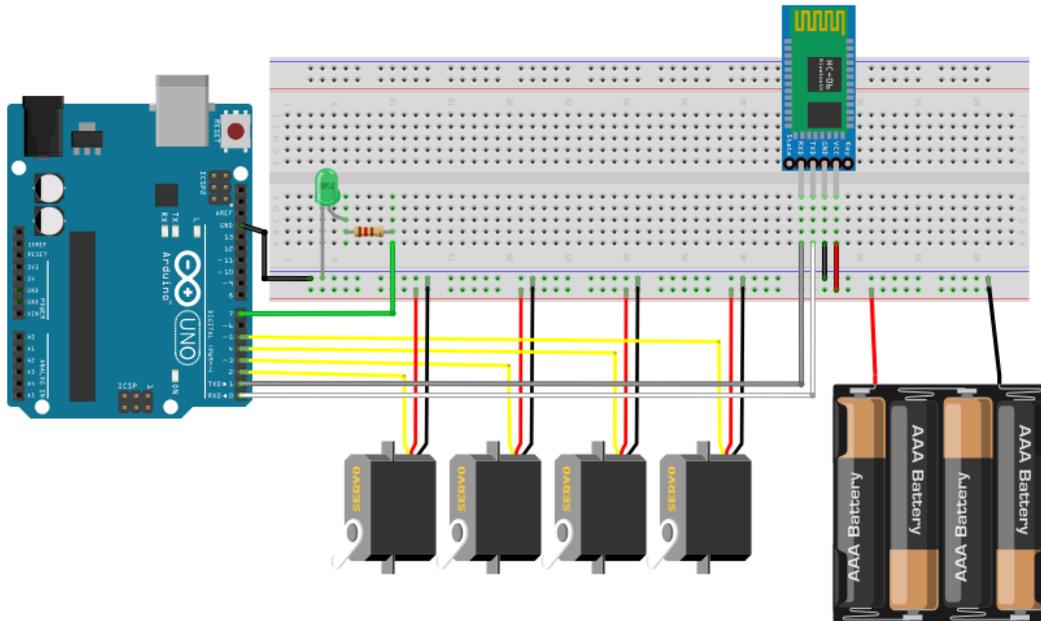
Desarrollar un programa que permita controlar movimientos del mini brazo robótico mediante un dispositivo móvil, utilizando como medio el Bluetooth entre el dispositivo móvil y la placa de Arduino.

Desarrollo de la actividad.

1. Desarrollar el código para el control de los servomotores del mini brazo robótico.
2. Conectar los servomotores, LED y Bluetooth con el Arduino, como se muestra en la imagen 1.
3. Cargar el Programa a la placa ARDUINO UNO
4. Descargar de la Play Store la APP:" ServoControl" para hacer movimientos y rutinas con el brazo robotico

Materiales	
Cantidad	Modelo
1	K-910
1	ARD-010
1	ARD-305
1	ARD-310
1	509-010
1	USB-490
1	E5/VER-C
1	R120 1/2
1	258-814
1	BAT-AA4

Diagrama de conexión.



Código.

```
#include <Servo.h> //Librería para trabajar con los servomotores

//Creando los objetos de la librería Servo
Servo servo_1, servo_2, servo_3, servo_4;

//Declaración de variables y matrices globales
int num_rut = 0;
int retardo = 10; //Entre más pequeño el valor, los servomotores se moverán más rápido.
int cont = 0;
char dato;
int ser[4] = {90,90,90,90};
int ser_rut[4][12];
void setup(){
    //Asignación de los Pines con los servomotores a utilizar
    servo_1.attach(2);
    servo_2.attach(3);
    servo_3.attach(4);
    servo_4.attach(5);
}
```

```

//Calibrando los servomotores a una posición inicial
servo_1.write (ser[0]);
servo_2.write (ser[1]);
servo_3.write (ser[2]);
servo_4.write (ser[3]);

//Declaración de pines de salida
pinMode(7,OUTPUT);

//Se inicia la comunicación con el Serial
Serial.begin(9600);
}

//Declaración del método "leer_dato" para la lectura de datos provenientes del dispositivo móvil
void leer_dato(){
    if(Serial.available()>=1) {
        dato = Serial.read();
        delay(30);
        dato = Serial.read();
    }
}

//Declaración del método "modo_rutina"
void modo_rutina(){
    if(dato == '0'){
        leer_dato();
        //Cuando el valor es '0' significa que se ha guardado un valor
        //Lectura del dato
    }
    if(num_rut < 12 ){
        //Condición que indica el número máximo de movimientos
        if(dato == 'W'){
            //Condición para guardar el movimiento del servomotor 1
            ser_rut[0][num_rut] = ser[0]; //Se almacena la posición actual del servomotor 1
        }
        else if(dato == 'X'){
            //Condición para guardar el movimiento del servomotor 2
            ser_rut[1][num_rut] = ser[1]; //Se almacena la posición actual del servomotor 2
        }
        else if(dato == 'Y'){
            //Condición para guardar el movimiento del servomotor 3
            ser_rut[2][num_rut] = ser[2]; //Se almacena la posición actual del servomotor 3
        }
        else if(dato == 'Z'){
            //Condición para guardar el movimiento del servomotor 4

```

```

    ser_rut[3][num_rut] = ser[3];    //Se almacena la posición actual del servomotor 4
}
num_rut++;                          //Se le suma '1' a la variable "num_rut", cada vez que un
                                    //movimiento sea guardado
}
}
else if(dato == 'R') {              //Cuando el valor es 'R' significa la activación de la rutina
num_rut = 0;
while(num_rut < 12 && dato == 'R'){ //Ciclo para repetir la rutina hasta que se envíe otro dato
    digitalWrite(7,1);              //Se enciende el LED que indica que entró en el modo rutina
    //Instrucciones para ejecutar los movimientos almacenados para el servomotor 1
    if(ser_rut[0][num_rut] > 0) {
        if(ser[0] <= ser_rut[0][num_rut]) {
            while(ser_rut[0][num_rut] >= ser[0]) { //Ciclo para ejecutar el movimiento
                servo_1.write(ser[0]);
                ser[0]++;
                delay(retardo);
            }
            delay(500);
        }
        else {
            while(ser_rut[0][num_rut] <= ser[0]) { //Ciclo para ejecutar el movimiento
                servo_1.write(ser[0]);
                ser[0]--;
                delay(retardo);
            }
            delay(500);
        }
    }
}
//Instrucciones para ejecutar los movimientos almacenados para el servomotor 2
else if(ser_rut[1][num_rut] > 0) {
    if(ser[1] <= ser_rut[1][num_rut]) {
        while(ser_rut[1][num_rut] >= ser[1]) { //Ciclo para ejecutar el movimiento

```

```

servo_2.write(ser[1]);
ser[1]++;
delay(retardo);
}
delay(500);
}
else {
while(ser_rut[1][num_rut] <= ser[1]) {           //Ciclo para ejecutar el movimiento
servo_2.write(ser[1]);
ser[1] --;
delay(retardo);
}
delay(500);
}
}
//Instrucciones para ejecutar los movimientos almacenados para el servomotor 3
else if(ser_rut[2][num_rut] > 0)
{
if(ser[2] <= ser_rut[2][num_rut])
{
while(ser_rut[2][num_rut] >= ser[2]) {           //Ciclo para ejecutar el movimiento
servo_3.write(ser[2]);
ser[2]++;
delay(retardo);
}
delay(500);
}
}
else {
while(ser_rut[2][num_rut] <= ser[2]) {           //Ciclo para ejecutar el movimiento
servo_3.write(ser[2]);
ser[2] --;
delay(retardo);
}
}
}

```

```

    delay(500);
}
}

//Instrucciones para ejecutar los movimientos almacenados para el servomotor 4
else if (ser_rut[3][num_rut] > 0) {
    if(ser[3] <= ser_rut[3][num_rut]) {
        while(ser_rut[3][num_rut] >= ser[3]) { //Ciclo para ejecutar el movimiento
            servo_4.write(ser[3]);
            ser[3]++;
            delay(retardo);
        }
        delay(500);
    }
    else {
        while(ser_rut[3][num_rut] <= ser[3]) { //Ciclo para ejecutar el movimiento
            servo_4.write(ser[3]);
            ser[3]--;
            delay(retardo);
        }
        delay(500);
    }
}
num_rut++;
leer_dato();
}
}

else if(dato == 'S') { //Cuando el valor es 'S' significa la eliminación de la rutina
    for(int i=0;i<2;i++) { //Proceso FOR para limpiar la matriz de donde se almacenan
        //las rutinas.
        for(int j=0;j<12;j++) {
            ser_rut[i][j] = - 1;
            Serial.println(ser_rut[i][j]);
        }
    }
}

```

```

    }
    num_rut = 0;
}
}

void modo_manual(){
    digitalWrite(7,0);          //Se apaga el LED que entró en el modo rutina
    ///////////ServoMotor 1//////////
    if (dato == 'A') {          //Si el dato es "A", el servo '1' gira en sentido
                                //a las manecillas de reloj

        if (ser[0] <= 175 ) {
            servo_1.write(ser[0]);
            ser[0]++;
            delay(retardo);
        }
        leer_dato();
    }
    else if (dato == 'a') {      //Si el dato es "a", el servo '1' gira en sentido
                                //contrario a las manecillas de reloj

        if (ser[0] >= 5 ) {
            servo_1.write(ser[0]);
            ser[0] --;
            delay(retardo);
        }
        leer_dato();
    }
    ///////////ServoMotor 2//////////
    if (dato == 'B') {          //Si el dato es "B", el servo '2' gira en sentido
                                //a las manecillas de reloj

        if (ser[1] <= 175 ) {
            servo_2.write(ser[1]);
            ser[1]++;
            delay(retardo);
        }
    }
}

```



```

if(ser[3] <= 175 ) {
  servo_4.write(ser[3]);
  ser[3]++;
  delay(retardo);
}
leer_dato();
}
else if(dato == 'd') {           //Si el dato es "d", el servo '4' gira en sentido
                                //contrario a las manecillas de reloj

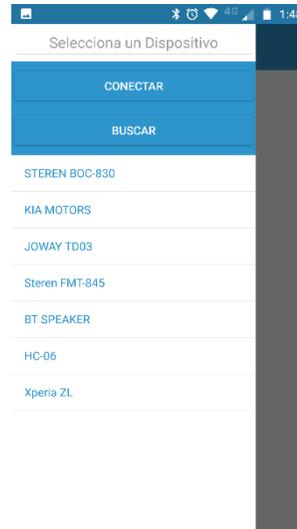
  if(ser[3] >= 5 ) {
    servo_4.write(ser[3]);
    ser[3] --;
    delay(retardo);
  }
  leer_dato();
}
}
void loop()
{
  leer_dato();                 //Ejecución del método para leer el dato
  modo_manual();              //Ejecución del método para el modo manual
  modo_rutina();              //Ejecución del método para el modo rutina
}

```

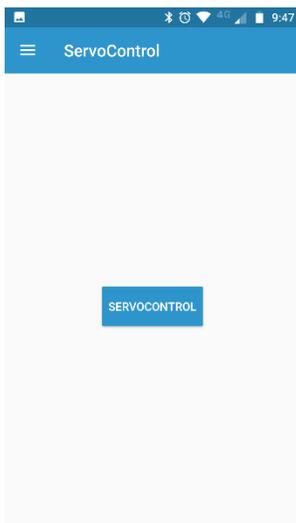
Descarga APP para el control del brazo robótico.



Descarga la aplicación "ServoControl" de la Play Store



Abre la aplicación, empareja tu dispositivo HC-06 y haz conexión.



Abre la opción ServoControl



Controla tu brazo robótico presionando los botones y programa rutinas.