

# Brazo robotico que sigue la luz

## Reporte

Universidad de Guadalajara

Seminario de programación de sistemas embebidos D01-I9893

### Equipo

Ángel de Jesús Vázquez Zarate

Naresh Satoshi López Ojeda

Daniel Esau Gonzalez Molina

Fernanda Hermosillo Gonzáles

**Resumen**-en esta práctica se diseñó y se construyó un brazo robótico que sigue la luz a base de foto-resistencias.

### **Materiales**

- ESP32 microcontrolador
- servomotor mg90s
- kit chasis de brazo robótico 4 grados de libertad
- foto-resistencia (LDR)
- Resistencia de 10KOhms a 1/4 watt
- transistores 2n2222 (4)
- resistencias 10kOhms
- cable 22awg estañado
- termofit en varias medidas y colores
- "shield" para placa de desarrollo esp32-Devkit

### **Marco teórico**

#### **ESP32 Microcontrolador**

El ESP32 es un microcontrolador económico y de alto rendimiento desarrollado por Espressif Systems. Se utiliza ampliamente en proyectos de electrónica e IoT (Internet de las Cosas) gracias a sus avanzadas funciones y versatilidad. Se puede programar con Arduino IDE, MicroPython o Espressif IDF y se aplica en diversos proyectos, como controles remotos, domótica, sistemas de monitorización, sensores conectados a la nube,

pantallas (como en su proyecto de contador con formato de reloj), robots, drones y otros dispositivos inteligentes.



#### **kit chasis de brazo robótico 4 grados de libertad**

Un brazo robótico también conocido como brazo mecánico, para armar, cuenta con funciones similares a las de un brazo humano. Los brazos robóticos son la suma de varios mecanismos que a su vez pueden unirse y formar un robot más complejo. Estos brazos robóticos están conectados por medio de articulaciones que permiten el movimiento rotacional de estos mismos, también pueden tener movimientos traslacionales o desplazamientos lineales.



### *servomotor mg90s*

El **MG90S** es un **servomotor** compacto y de alto torque, muy popular en proyectos de robótica, automatización y modelismo. Funciona mediante una señal PWM (modulación por ancho de pulso) que permite controlar con precisión la posición del eje dentro de un rango típico de 0° a 180°. Este servomotor se caracteriza por tener **engranajes metálicos**, lo que le da mayor durabilidad y resistencia comparado con modelos de plástico como el SG90. Opera generalmente a **4.8V–6V** y puede entregar un torque de hasta **2.2 kg·cm** a 6V. Es ideal para aplicaciones donde se requiere más fuerza y resistencia que la que ofrecen micro servos básicos.



### *Fotorresistencia*

Una fotorresistencia o LDR (Light Dependent Resistor) es un componente electrónico cuya resistencia varía según la cantidad de luz que incide sobre él. Cuando hay mucha luz, su resistencia disminuye, permitiendo el paso de corriente; cuando hay poca luz, su resistencia aumenta, dificultando el paso de corriente.



### *transistor*

El 2N2222 es un transistor de unión bipolar (BJT) NPN que permite la transferencia adecuada de energía

eléctrica entre diferentes partes de un circuito eléctrico.



### *resistencia*

componente electrónico que limita el flujo de corriente eléctrica en un circuito.



### *Código*

```
#define maX 180
#define maY 65
#define maxP 15
#define minP 0
```

Define los **ángulos máximos o mínimos** de movimiento para cada servo. Por ejemplo, maY = 65 limita cuánto se puede mover en el eje Y.

```
int x = 90 , y = 45, z = 90, claw = 0;
```

Posiciones iniciales de los servos:

- x, y, z → articulaciones.
- claw → pinza (cerrada).

```
void setup() {
  // put your setup code here, to
  run once:
  Serial.begin(115200);
  motX.attach(seMoX);
  motY.attach(seMoY);
  motZ.attach(seMoZ);
  pinza.attach(seMoP);
}
```

```
motX.write(x);
motY.write(y);
motZ.write(z);
pinza.write(claw);
delay(500);
Serial.println("PINZA INICIADA");
delay(1000);
}
```

Inicializa los servos, configura sus posiciones y establece comunicación por el puerto serial para depuración.

```
void loop() {
    // put your main code here, to
    run repeatedly:
    if(analogRead(inXP)>(analogRead(i
nXM)+umPho)){
        if(x>(0)) x-=5;
    }
}
```

Si hay **más luz en la derecha (XP)** que en la izquierda (XM) con una diferencia mayor al umbral, el servo gira hacia esa dirección disminuyendo x.

```
if(analogRead(inXM)>(analogRead(i
nXP)+150+umPho)){
    if(x<(170)) x+=5;
}
```

Similar al anterior, pero ahora si la luz está más a la izquierda (XM), gira en la otra dirección.

```
if(analogRead(inYP)>(analogRead(i
nYM)+umPho-50)){
    if(y>(10)) y-=5;
}
```

Detecta luz en la parte superior (YP) y baja y (el brazo se mueve hacia arriba).

```
if(analogRead(inYM)>(analogRead(in
YP)+umPho-50)){
    if(y<(maY)) y+=5;
}
```

Lo mismo pero si la luz está abajo, aumenta y (el brazo baja).

```
motX.write(x);
```

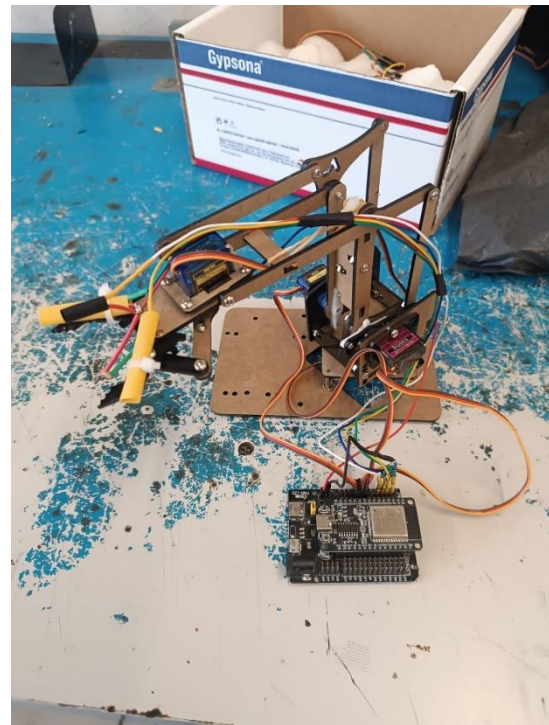
```
motZ.write(z);
delay(10);
motY.write(y);
```

Mueve los servos a las nuevas posiciones actualizadas según la luz detectada. z no cambia en este código, pero puedes modificarlo si necesitas articulación adicional.

```
Serial.print("X");
Serial.println(x);
Serial.print("Y");
Serial.println(y);
delay(100);
}
```

Imprime las posiciones actuales de los servos x e y para monitorear el movimiento del brazo.

## Resultados



## conclusion

Con esta práctica, logramos construir un brazo robotico que sigue fuentes de luz utilizando fotorresistencias. A través del análisis de las lecturas analógicas de cuatro fotorresistencias ubicadas en una disposición en cruz, el

sistema determina en qué dirección se encuentra la mayor intensidad luminosa y ajusta progresivamente la posición de los servos en los ejes X e Y para orientarse hacia ella y pues esto permitió comprender cómo implementar un sistema de seguimiento reactivo basado en sensores analógicos, el control básico de servomotores con ESP32, y la importancia del procesamiento de señales para la toma de decisiones en sistemas embebidos.

## Referencias

- Arduino. (n.d.). *ESP32 Documentation*. Retrieved from <https://www.arduino.cc>

