



PROYECTO 2

REMOTE MANAGEMENT OF ROBOTICS ARM



FECHA:

31/05/2022

AUTORES:

Miguel de las Heras Fuentes

Diego Dorado Galán

Contenido

1. OBJETIVO DEL PROYECTO	3
2. COMPONENTES UTILIZADOS	3
3. ESQUEMA DEL CIRCUITO EN FRITZING	3
4. EXPLICACIÓN DETALLADA DE LA SOLUCIÓN	4
4.1. PLACA MASTER (JOYSTICK).....	4
4.2. PLACA SLAVE (SERVOMOTORES).....	6
5. IMAGEN DEL CIRCUITO	7

Ilustración 1: Componentes utilizados para la realización del proyecto	3
Ilustración 2: Esquema del circuito realizado en la plataforma "Fritzing"	3
Ilustración 3: Lógica de la solución.....	4
Ilustración 4: Imagen del circuito completo	7

1. OBJETIVO DEL PROYECTO

El objetivo de este proyecto es controlar un brazo robótico formado por dos servomotores mediante un joystick. Los dos servomotores serán controlados por una placa ESP32 y el joystick será controlado por otra ESP32 diferente, de manera que una placa le comunicará a la otra la posición en la que se tienen que situar los servomotores. Esta comunicación se realizará mediante bluetooth utilizando el patrón Master-Slave siendo el maestro la placa que controla el joystick y el esclavo la que controla los servos.

2. COMPONENTES UTILIZADOS

- 2 servomotores: Cada uno representará un eje de coordenadas(X, Y)
- 2 placas ESP32
- Joystick

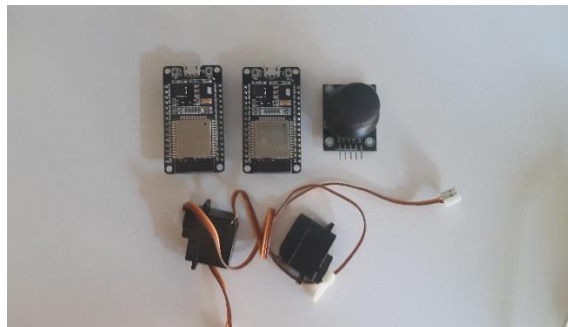


Ilustración 1: Componentes utilizados para la realización del proyecto

3. ESQUEMA DEL CIRCUITO EN FRITZING

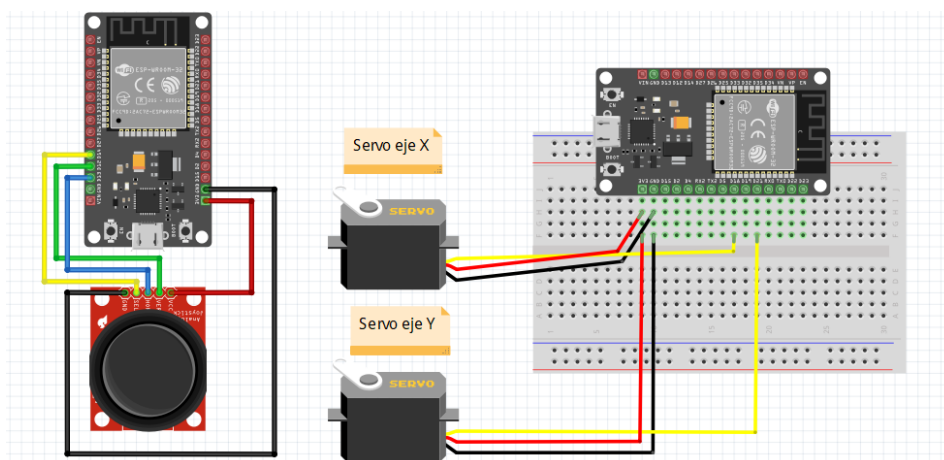


Ilustración 2: Esquema del circuito realizado en la plataforma "Fritzing"

4. EXPLICACIÓN DETALLADA DE LA SOLUCIÓN

Como hemos comentado en las secciones anteriores, nuestro objetivo es manejar dos servomotores desde una placa distinta a la que los controla. Para ello debemos utilizar algún protocolo de envío de datos. En esta ocasión hemos utilizado Bluetooth. La placa que controla el joystick(master) enviará mediante bluetooth a la placa que controla los servomotores(slave) los ángulos en los que deberán situarse los servos. En la siguiente imagen podemos ver la lógica de la solución propuesta:

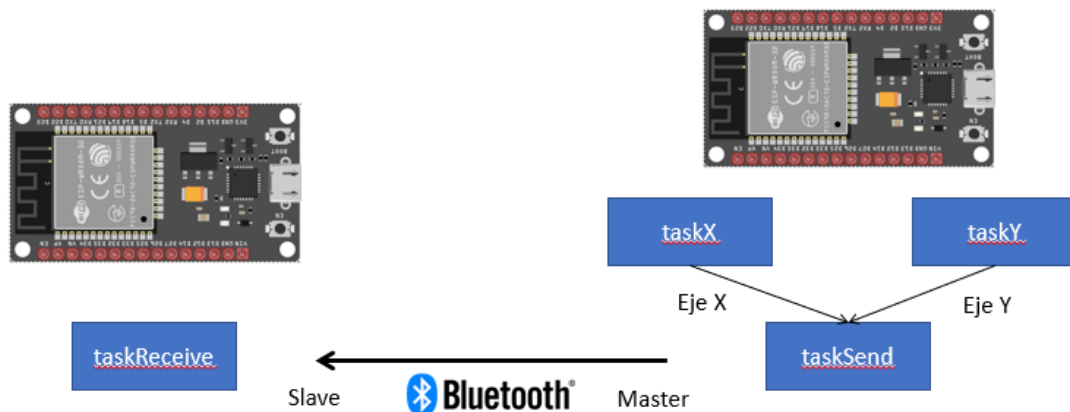


Ilustración 3: Lógica de la solución

A continuación se explica de manera detallada el código de cada placa.

4.1. PLACA MASTER (JOYSTICK)

Pines utilizados:

- P12: eje Y del joystick
- P13: eje X del joystick
- P14: botón del joystick

En esta placa lo primero que hemos implementado ha sido la comunicación bluetooth para que pueda conectarse a la placa Slave. Para iniciar la comunicación como maestro y conectarnos a la otra placa hemos utilizado los siguientes comandos.

```
#define name "ESP32Servos"

SerialBT.begin("ESP32Joystick",true)

SerialBT.connect(name);
```

Para el control de la velocidad hemos habilitado un interrupción en el pin 14 correspondiente al botón del joystick de manera que cada vez que se pulse se podrá cambiar entre dos posibles velocidades.

Para la lectura de valores se han creado dos tareas FreeRTOS; una para cada eje de coordenadas que queremos medir. Lo primero que haremos será medir el valor analógico del pin mediante:

```
lecturaX = analogRead(XPIN);
```

A continuación mapearemos ese valor(0-4095) en un valor entre 0 y 180 grados ya que son los ángulos mínimo y máximo en los que se puede posicionar el servo. Para ello utilizamos:

```
angleX = map(lecturaX,0,4095,0,180);
```

Por último, mediante una sucesión de if-else comprobamos si el joystick se ha movido y en caso positivo incrementamos o decrementamos la posición actual con la velocidad fijada en dicho momento.

```
xSemaphoreTake(xSemaforoX,xTicksToWait);  
if (angleX>=95)  
{  
    posicionX += velocidad;  
    if(posicionX > MAX_ANGLE) posicionX = 180;  
}  
else if (angleX <=75)  
{  
    posicionX -= velocidad;  
    if(posicionX < MIN_ANGLE) posicionX = 0;  
}  
xSemaphoreGive(xSemaforoX);
```

Como se puede apreciar; se ha utilizado un semáforo mutex para evitar condiciones de carrera en la actualización de la posición entre esta tarea y la de envío.

Finalmente, se ha creado un tarea freeRTOS para el envío de los datos a la placa Slave. Para ello se accede tanto a la posición X como a la posición Y mediante un semáforo mutex para garantizar la exclusividad y se envían ambos datos mediante los comandos:

```
SerialBT.write(posicionX);  
SerialBT.write(posicionY);
```

4.2. PLACA SLAVE (SERVOMOTORES)

Pines utilizados:

- P18: Servomotor eje X
- P21: Servomotor eje Y

En esta placa al igual que en la anterior; lo primero que hacemos es configurar la conexión bluetooth para habilitar la recepción de datos por parte de la placa master. Para ello utilizamos el comando:

```
SerialBT.begin("ESP32Servos");
```

Como se puede apreciar, aquí a diferencia de la anterior placa; no se ha utilizado el argumento true ya que no la estamos configurando como master.

Una vez realizada la conexión bluetooth, habilitamos los pines correspondiente a cada servo:

```
servoX.attach(XPIN);  
servoY.attach(YPIN);
```

En la función loop utilizaremos una sentencia if para comprobar si hay nuevos datos recibidos en el bluetoothSerial y en caso positivo procederemos a su lectura. Para ello llamaremos dos veces a la función:

```
SerialBT.read();
```

En la primera llamada leeremos el valor del eje X y en la segunda el del eje Y. Una vez obtenidos estos valores le comunicaremos a cada servo su nueva posición mediante:

```
servoX.write(incomingX);  
servoY.write(incomingY);
```

5. IMAGEN DEL CIRCUITO

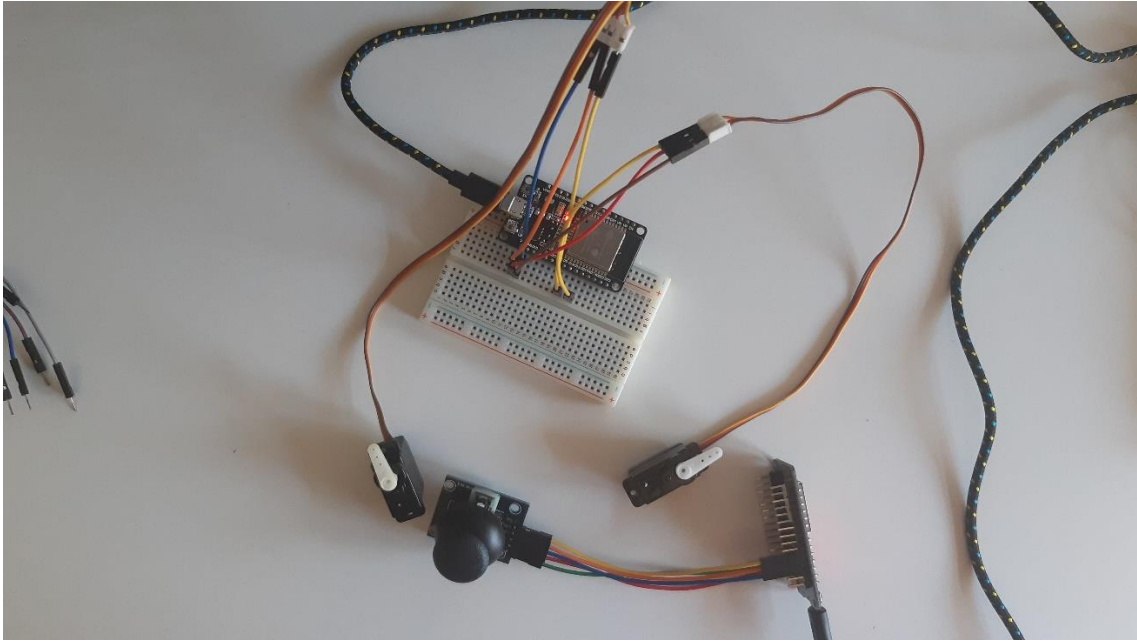


Ilustración 4: Imagen del circuito completo