

PRIMERA PARTE: Comprobación de conexiones

Comprobación de conexión entre Arduino y los servomotores

La verificación de la conexión entre el Arduino y los servomotores es esencial para asegurar el correcto funcionamiento del sistema y evitar posibles daños a los componentes, como se muestra en la **Figura 1**.

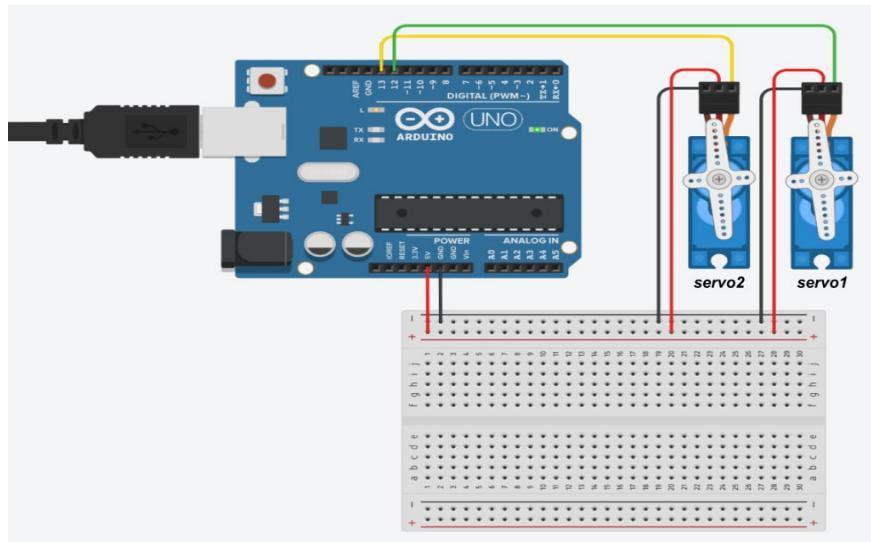


Figura 1: Diagrama de conexión de servomotores a la placa Arduino Uno.

A continuación, se presenta el código que debe cargarse en el microcontrolador Arduino para llevar a cabo la prueba de funcionamiento.

```
#include <Servo.h> // Incluye la biblioteca Servo para controlar los servomotores

Servo myservo1; // Crea un objeto Servo para controlar el primer servomotor (motor X)
Servo myservo2; // Crea un objeto Servo para controlar el segundo servomotor (motor Y)

void setup() {
  myservo1.attach(12); // Asocia el primer servo (motor X) al pin 12 de la placa Arduino
  myservo2.attach(13); // Asocia el segundo servo (motor Y) al pin 13 de la placa Arduino
}

void loop() {
  myservo1.write(0); // Mueve el motor X a la posición de 0 grados
  myservo2.write(0); // Mueve el motor Y a la posición de 0 grados
  delay(2000); // Espera dos segundo para que el servo se mueva

  myservo1.write(180); // Mueve el motor X a la posición de 180 grados
  myservo2.write(180); // Mueve el motor Y a la posición de 180 grados
  delay(2000); // Espera dos segundo para que el servo se mueva
}
```

Diagnóstico de posibles problemas

Si los servomotores no se mueven, se recomienda verificar las conexiones físicas, asegurándose de que los cables no estén sueltos o mal conectados.

Comprobación de la conexión entre Arduino y Raspberry Pi

Para verificar que la conexión entre el microcontrolador Arduino y la Raspberry Pi se ha establecido correctamente, se recomienda realizar una prueba cargando un código en el Arduino y ejecutando un script en Python desde la Raspberry Pi. Es crucial identificar correctamente el puerto USB que conecta ambos dispositivos; en este caso, se utiliza el puerto USB0. En la figura siguiente se presenta el esquema de conexión.

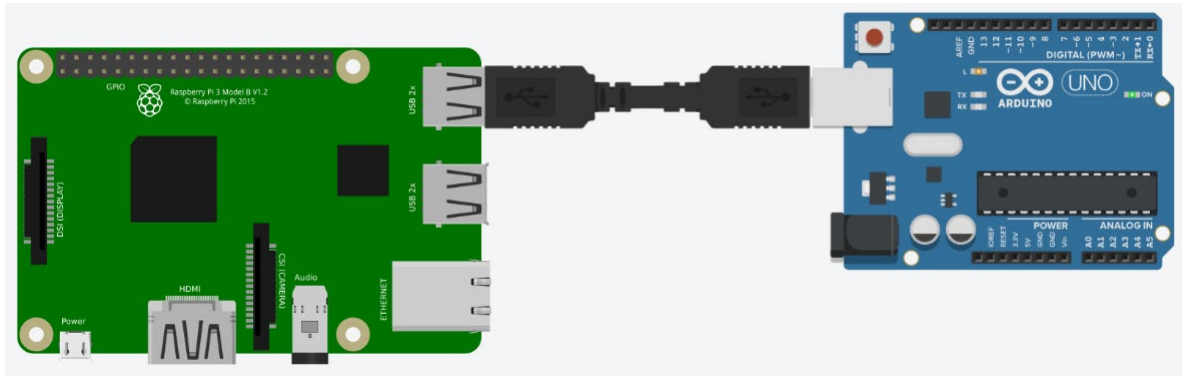


Figura 2: Diagrama de conexión entre microcontrolador Arduino y Raspberry Pi.

Código a cargar en el microcontrolador Arduino:

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  if (Serial.available() > 0) {
    String mensaje = Serial.readStringUntil('\n'); // Lee el mensaje
    Serial.println("El mensaje recibido es: " + mensaje); // Responde al Raspberry Pi
  }
}
```

A continuación, se presenta el código principal en Python que debe ejecutarse en la Raspberry Pi, asumiendo que el microcontrolador Arduino ya tiene cargado el programa correspondiente.

```
import serial
import time

# Configura el puerto serie
arduino = serial.Serial(port='/dev/ttyUSB0', baudrate=9600, timeout=1)

def send_and_receive(message):
    arduino.write(message.encode()) # Envía el mensaje al Arduino
    time.sleep(1) # Espera para que el Arduino procese
    response = arduino.readline().decode().strip() # Lee la respuesta
```

```

return response

try:
    while True:
        msg = input("Escribe un mensaje para enviar al Arduino: ")
        if msg.lower() == 'salir':
            break
        response = send_and_receive(msg)
        print(f"Respuesta del Arduino: {response}")
except Exception as e:
    print(f"Error: {e}")
finally:
    arduino.close()

```

Diagnóstico de posibles problemas

En caso de que no sea posible establecer comunicación y se produzcan errores durante la ejecución del script principal en Python, se recomienda verificar el puerto de comunicación entre la Raspberry Pi y el Arduino. Para ello, se puede utilizar el comando **ls /dev/tty*** en la consola, que lista los puertos seriales disponibles. Los puertos típicos asociados al Arduino suelen ser /dev/ttyUSB0 o /dev/ttyACM0.

Comprobación de conexión entre Raspberry Pi y Cámara Pi

Para verificar la correcta conexión entre el módulo Raspberry Pi y una cámara Pi, se puede ejecutar un código en Python que permita realizar una prueba de captura fotográfica. El diagrama de conexión se presenta en la **Figura 3**.

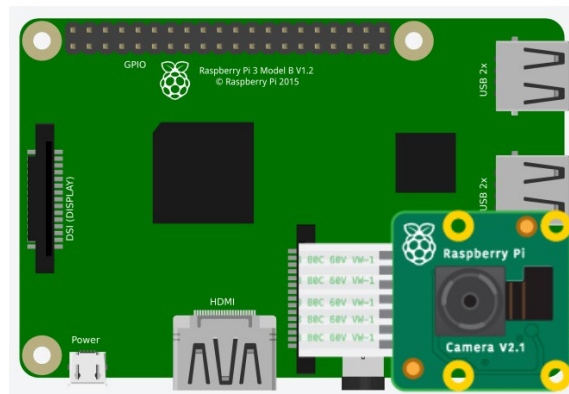


Figura 3: Diagrama de conexión entre Raspberry Pi y Cámara Pi.

```

from picamera import PiCamera
from time import sleep

camera = PiCamera() # Crear un objeto de cámara

# Configuración básica
camera.resolution = (640, 480) # Resolución de la imagen
camera.framerate = 24          # Velocidad de fotogramas

```

```
try:
    print("Iniciando la cámara")
    camera.start_preview() # Inicia la previsualización de la cámara

    # Capturar una imagen
    camera.capture('/home/pi/test_image.jpg')
    print("Imagen capturada y guardada en /home/pi/test_image.jpg")

finally:
    camera.stop_preview() # Detiene la previsualización
    camera.close()       # Libera los recursos de la cámara
    print("Cámara desconectada")
```

Diagnóstico de posibles problemas

Los problemas en la prueba de conexión entre la cámara y la Raspberry Pi puede deberse a varias causas. En primer lugar, una conexión incorrecta de la cámara al puerto CSI de la Raspberry Pi podría ser la causa del fallo. Además, existe la posibilidad de que la cámara no esté activada en la configuración de la Raspberry Pi, lo que puede verificarse mediante el comando **sudo raspi-config** en la consola, seleccionando la opción Interfacing Options > Camera. Finalmente, la falta de instalación de la biblioteca picamera también podría generar inconvenientes. Para instalar esta biblioteca, se puede proceder desde la consola con los siguientes comandos: primero, **sudo apt update** para actualizar la información de los paquetes disponibles, y luego, **sudo apt install python3-picamera** para realizar la instalación.

Diagrama de conexión del sistema LitoScan.AR

La **Figura 4** muestra el diagrama de conexión completo del sistema LitoScan.AR. Este incluye la conexión de los servomotores al microcontrolador Arduino, la interconexión del Arduino con el módulo Raspberry Pi mediante un puerto USB, y la conexión de la cámara a la Raspberry a través del puerto CSI (Camera Serial Interface).

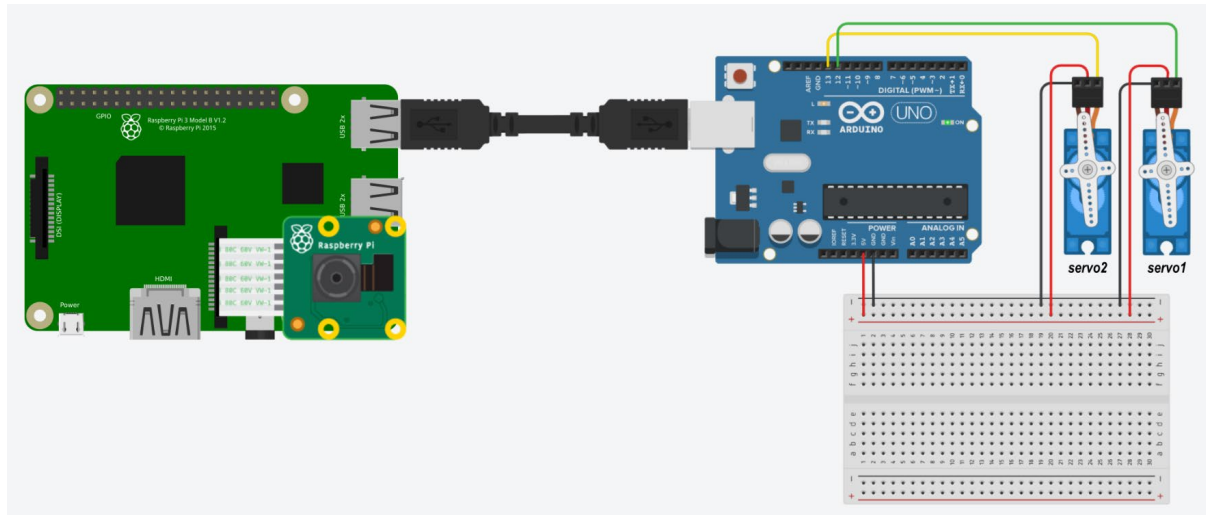


Figura 4: Esquema de conexión del sistema LitoScan.AR.