

Examen Interactivo: Práctica 7

Preguntas y Respuestas

Pregunta 1: ¿Qué es el protocolo I2C y para qué se utiliza?

✓ Respuesta: Es un protocolo de comunicación serie que permite la conexión de múltiples dispositivos usando solo dos líneas: SDA y SCL.

Intentos: 1

Pregunta 2: ¿Cuáles son las principales líneas de comunicación en I2C?

✓ Respuesta: SDA (datos) y SCL (reloj).

Intentos: 1

Pregunta 3: ¿Cuál es la diferencia entre direcciones de 7 bits y 10 bits en I2C?

✓ Respuesta: La dirección de 7 bits es la más común y permite hasta 128 dispositivos, mientras que la de 10 bits amplía la cantidad de dispositivos conectados.

Intentos: 1

Pregunta 4: ¿Cuál es la función de la resistencia pull-up en las líneas SDA y SCL?

✗ Respuesta: Reducir el consumo de energía.

✓ Respuesta: Mantener las líneas en nivel alto cuando no hay comunicación. Intentos: 2

Pregunta 5: ¿Qué biblioteca se usa en Arduino para trabajar con I2C?

✓ Respuesta: Wire.h

Intentos: 1

Pregunta 6: ¿Qué función se usa para iniciar la comunicación I2C en ESP32?

✗ Respuesta: I2C.open()

✓ Respuesta: Wire.begin() Intentos: 2

Pregunta 7: ¿Cuál es la ventaja de I2C frente a SPI?

✓ Respuesta: Usa menos pines y permite conectar varios dispositivos con una menor cantidad de conexiones.

Intentos: 1

Pregunta 8: ¿Qué dispositivo se usó en la práctica para la comunicación I2C?

☒ Respuesta: Un sensor de temperatura y humedad.

Intentos: 1

Pregunta 9: ¿Qué velocidad de comunicación soporta el protocolo I2C?

☒ Respuesta: 10 Mbps

☒ Respuesta: Estándar de 100 kHz y rápido de hasta 400 kHz.

Intentos: 2

Pregunta 10: ¿Cómo se obtiene la dirección de un dispositivo I2C en Arduino?

☒ Respuesta: Usando un escáner I2C con la biblioteca Wire.h.

Intentos: 1

Pregunta 11: ¿Cuál es la función principal de Wire.requestFrom() en Arduino?

☒ Respuesta: Enviar datos a un esclavo.

☒ Respuesta: Solicitar datos desde un dispositivo esclavo.

Intentos: 2

Pregunta 12: ¿Qué ocurre si dos dispositivos intentan transmitir datos al mismo tiempo en I2C?

☒ Respuesta: Se usa un mecanismo de arbitraje para evitar colisiones y garantizar la comunicación correcta.

Intentos: 1

Pregunta 13: ¿Qué dispositivo actúa como maestro en la comunicación I2C en ESP32?

☒ Respuesta: El ESP32.

Intentos: 1

Pregunta 14: ¿Por qué es importante la dirección en I2C?

☒ Respuesta: Para identificar a qué dispositivo se está enviando o recibiendo datos.

Intentos: 1

Pregunta 15: ¿Qué ventaja tiene la comunicación I2C en sensores?

☒ Respuesta: Permite transmitir a mayores distancias.

☒ Respuesta: Facilita la conexión de múltiples sensores con solo dos cables.

Intentos: 2

Pregunta 16: ¿Cuál es un problema común en I2C y cómo se soluciona?

☒ Respuesta: Problemas de ruido en las líneas SDA y SCL, que se pueden solucionar con resistencias pull-up adecuadas.

Intentos: 1

Pregunta 17: ¿Qué pasa si se conecta más de un dispositivo con la misma dirección en I2C?

☒ Respuesta: Se generarían conflictos en la comunicación y los datos podrían corromperse.

Intentos: 1

Pregunta 18: ¿Cómo se cambia la dirección de un dispositivo I2C si hay conflicto?

☐ Respuesta: No se puede cambiar.

☒ Respuesta: Algunos dispositivos permiten cambiar la dirección mediante pines de configuración.

Intentos: 2

Pregunta 19: ¿Qué ocurre si la velocidad de I2C es demasiado alta para un dispositivo esclavo?


☒ Respuesta: Puede generar errores en la transmisión y fallos en la recepción de datos.

Intentos: 1

Pregunta 20: ¿Cómo podemos verificar la comunicación I2C en ESP32?

☒ Respuesta: Usando un escáner I2C y monitoreando la comunicación con herramientas de depuración.

Intentos: 1

 **Informe Final Total de preguntas: 20 Total de intentos realizados: 25 Porcentaje de aciertos en el primer intento: 16/20 = 80%**