

Investigación de los Protocolos de Comunicación Industrial

TAREA

1. Investigación de los protocolos de comunicación industrial

1.1 Serial RS485

El protocolo **RS485** es un estándar de comunicación serie utilizado en entornos industriales debido a su robustez y capacidad para manejar múltiples dispositivos en un mismo bus.

Características principales:

- Es un protocolo **dúplex** que permite la comunicación bidireccional en un solo par de cables.
- Soporta **hasta 32 dispositivos** en un mismo bus sin necesidad de un repetidor.
- Funciona en un esquema **diferencial**, lo que le permite resistir interferencias electromagnéticas.
- Permite transmisiones a largas distancias, de hasta **1200 metros**, dependiendo de la velocidad de comunicación.

Comparación con RS232 y RS422:

Característica	RS232	RS422	RS485
Tipo de señal	Unipolar	Diferencial	Diferencial
Distancia máxima	15 metros	1200 metros	1200 metros
Número de dispositivos	1 transmisor y 1 receptor	1 transmisor y 10 receptores	Hasta 32 dispositivos (o más con repetidores)
Inmunidad al ruido	Baja	Alta	Muy alta

Aplicaciones en la industria:

- Comunicación entre PLCs (Controladores Lógicos Programables).
- Redes de sensores industriales.
- Control de motores y variadores de frecuencia.
- Sistemas SCADA para monitoreo de procesos.

Ventajas y desventajas:

✓ Ventajas:

- Comunicación confiable en entornos industriales.
- Soporta múltiples dispositivos en un solo bus.
- Alto alcance sin perder calidad de señal.

✗ Desventajas:

- Requiere protocolo de control para evitar colisiones en la comunicación.
- No es plug-and-play como USB o Ethernet.

1.2 Software MODBUS

El protocolo **MODBUS** es un estándar de comunicación industrial desarrollado en 1979 por Modicon para permitir la comunicación entre dispositivos electrónicos.

Tipos de MODBUS:

1. **MODBUS RTU:** Utiliza comunicación serie (RS485 o RS232) con formato binario, lo que lo hace más eficiente en términos de transmisión.
2. **MODBUS ASCII:** Similar al RTU, pero los datos se envían en formato ASCII, lo que facilita la lectura pero reduce la velocidad de transmisión.
3. **MODBUS TCP/IP:** Adaptación de MODBUS para redes Ethernet, lo que permite comunicación a través de internet.

Casos de uso en la industria:

- Control de procesos industriales con PLCs.
 - Integración de sensores y actuadores en sistemas de monitoreo.
 - Interfaz entre SCADA y equipos de automatización.
-

1.3 Wireless MQTT

El protocolo **MQTT (Message Queuing Telemetry Transport)** es un estándar de comunicación inalámbrica basado en el modelo **publicación/suscripción**, ideal para IoT y sistemas industriales conectados.

Principios de funcionamiento:

- En lugar de comunicación directa entre dispositivos, MQTT usa un **servidor central llamado "broker"** que distribuye los mensajes entre los clientes.
- Cada dispositivo puede actuar como **publicador** (envía datos) o **suscriptor** (recibe datos).
- Utiliza un sistema de **"topics"** para organizar la información (por ejemplo, `planta/temperatura`).

Uso en IoT y automatización industrial:

- Monitoreo de sensores inalámbricos en fábricas.
 - Comunicación entre robots autónomos y servidores de control.
 - Supervisión de procesos remotos mediante aplicaciones en la nube.
-

2. Descripción de funcionamiento

RS485:

1. Un maestro envía una señal diferencial en el bus.
2. Todos los dispositivos escuchan, pero solo el esclavo con la dirección correcta responde.
3. La respuesta del esclavo se envía de regreso al maestro.

MODBUS RTU:

1. Un PLC maestro envía una solicitud a un esclavo específico.
2. El esclavo procesa la solicitud y responde con los datos requeridos.
3. Se verifica la integridad de los datos mediante el código CRC.

MQTT:

1. Un sensor (cliente) publica datos en un topic (ejemplo: `fabrica/temperatura`).
2. El broker recibe el mensaje y lo reenvía a todos los suscriptores interesados.
3. Un sistema de control central suscrito al topic `fabrica/temperatura` recibe los datos y toma decisiones.

3. Esquemas eléctricos de implementación

Esquema RS485 con un microcontrolador (ESP32 o Arduino)

Diagrama de conexión básico:

ESP32		MAX485	Dispositivo RS485
TX	----->	DI	
RX	<-----	RO	
DE	----->	RE	(conectados juntos y controlados por GPIO)
A	----->	A	
B	----->	B	
GND	----->	GND	

Esquema MQTT con ESP32 y un broker

Ejemplo de conexión de ESP32 con un servidor MQTT usando Wi-Fi:

```
#include <WiFi.h>
#include <PubSubClient.h>

const char* ssid = "TuRedWiFi";
const char* password = "TuContraseña";
const char* mqttServer = "broker.hivemq.com";

WiFiClient espClient;
PubSubClient client(espClient);

void setup() {
  WiFi.begin(ssid, password);
  client.setServer(mqttServer, 1883);
}

void loop() {
  client.publish("fabrica/temperatura", "25°C");
  delay(5000);
}
```