





## ¿QUÉ HACE ESTE CÓDIGO?

Imagina que tienes **varios sensores de temperatura** en diferentes silos. Este código convierte un **ESP32-C3** en el "JEFE" que:

-  **Pregunta a todos los sensores** "¿Quién está conectado?"
-  **Les pide que midan** temperatura y humedad
-  **Recoge todos los datos** y los organiza
-  **Envía la información** por radio (LoRa) a otro dispositivo

## ANALOGÍA SIMPLE:

text

## FÁBRICA DE SENSORES:

SENSORES (ESP8266) → SUPERVISOR (ESP32-C3) → GERENTE (Gateway)



Trabajadores

Jefe de planta

Director general

(Miden temperatura) (Recoge datos) (Recibe reportes)

---

## CONEXIONES FÍSICAS - PASO A PASO

### MATERIALES NECESARIOS:

- 1x ESP32-C3-DevKitM-1
- 1x Módulo MAX485 (RS485)
- 1x Módulo LoRa SX1278
- Cables dupont
- Fuente 3.3V

### CONEXIÓN RS485 (Para hablar con los sensores):

Módulo MAX485	ESP32-C3	Color Sugerido
---------------	----------	----------------

RO	GPIO4	Amarillo
----	-------	----------

Módulo MAX485	ESP32-C3	Color Sugerido
DI	GPIO5	Verde
DE/RE	GPIO6	Azul
VCC	3.3V	Rojo
GND	GND	Negro

#### ¿Para qué sirve cada pin?

- **RO (Receive Out):** Para **ESCUCHAR** lo que dicen los sensores
- **DI (Data In):** Para **HABLAR** con los sensores
- **DE/RE:** Como un **INTERRUPTOR** que cambia entre hablar/escuchar
- **VCC/GND:** Alimentación eléctrica

#### **CONEXIÓN LoRa (Para enviar datos lejos):**

Módulo LoRa	ESP32-C3	Función
SCK	GPIO2	Reloj
MISO	GPIO3	Recepción
MOSI	GPIO4	Transmisión
CS	GPIO7	Selección
RST	GPIO8	Reinicio
DIO0	GPIO9	Interrupción

---

## **CÓMO FUNCIONA EL CÓDIGO - PARTES PRINCIPALES**

### 1. **CONFIGURACIÓN INICIAL (Líneas 1-25)**

cpp

```
#define ENDPOINT_ID "EP01"
```

```
#define CICLO_COMPLETO 120000 // 2 minutos
```

### ¿Para qué sirve?

- ENDPOINT\_ID: Como ponerle **nombre** a este dispositivo
- CICLO\_COMPLETO: Cada **cuánto tiempo** hace todo el proceso

## 2. 📁 ALMACENAMIENTO DE DATOS (Líneas 33-41)

cpp

```
struct SensorData {  
    String address; // Dirección del sensor (ej: "01")  
    String location; // Ubicación (ej: "Silo_Norte")  
    float temperature; // Temperatura medida  
    float humidity; // Humedad medida  
};
```

Como una TABLA donde guarda información de cada sensor

## 3. 🎯 FUNCIONES IMPORTANTES

**ejecutarDiscovery()** - El "LLAMADO A LISTA"

cpp

```
void ejecutarDiscovery() {  
    broadcastRS485("DISCOVERY"); // Grita: "¡Respondan todos!"  
    // Espera respuestas por 10 segundos  
}
```

**Acción:** Pregunta "¿Quién está conectado?" y guarda quién responde

**ejecutarLecturaSensores()** - El "TOMA DE DATOS"

cpp

```
void ejecutarLecturaSensores() {  
    for (cada sensor conectado) {  
        enviarComandoRS485(sensor, "READ_SENSOR"); // "Mide ahora"  
        // Espera la respuesta con los datos
```

```
}  
}
```

**Acción:** Le pide a cada sensor que mida y le envíe los datos

#### **enviarDatosLoRa() - El "REPORTE FINAL"**

cpp

```
void enviarDatosLoRa() {  
    // Junta todos los datos en un paquete JSON  
    // Lo envía por radio LoRa  
}
```

**Acción:** Prepara un reporte con todos los datos y lo envía por radio

---

#### **CICLO DE TRABAJO (2 MINUTOS)**

##### **RELOJ AUTOMÁTICO:**

text

00:00 - 00:10 →  "LLAMADO A LISTA" (Discovery)

00:10 - 00:40 →  "TOMA DE DATOS" (Lectura sensores)

00:40 - 00:55 →  "ENVÍO POR RADIO" (LoRa)

00:55 - 02:00 →  "DESCANSO" (Espera próximo ciclo)

##### **CÓMO SE COMUNICA CON LOS SENSORES**

**Cuando el ESP32-C3 pregunta:**

text

"ADDR:BROADCAST|CMD:DISCOVERY"

(Significa: "¡Oigan todos! ¿Quién está conectado?")

**Los sensores responden:**

text

"ADDR:01|CMD:DISCOVERY\_RESP|LOC:Silo\_Norte|TEMP:25.5|HUM:60.2"

(Significa: "Yo soy el sensor 01, estoy en Silo Norte, mido 25.5°C y 60.2% humedad")

---

## CONFIGURACIÓN PARA TU PROYECTO

### CAMBIOS QUE PUEDES HACER:

#### Tiempos del ciclo (en milisegundos):

cpp

```
#define CICLO_COMPLETO 900000 // 15 minutos (en producción)
```

```
#define CICLO_COMPLETO 120000 // 2 minutos (para pruebas)
```

#### Identificación:

cpp

```
#define ENDPOINT_ID "EP01" // Cambiar por "EP02", "EP03", etc.
```

---

## SOLUCIÓN DE PROBLEMAS COMUNES

### Problema: No detecta sensores

#### Posibles causas:

- ✗ Conexiones RS485 incorrectas
- ✗ Sensores no alimentados
- ✗ Pines RO/DI intercambiados

#### Solución:

1. Verificar que el MAX485 tenga 3.3V
2. Revisar cables RO y DI
3. Probar con un solo sensor primero

### Problema: Error en LoRa

#### Solución:

1. Verificar que el módulo LoRa esté bien conectado
2. Confirmar que la antena esté conectada
3. Revisar pines SCK, MISO, MOSI

### Problema: El ciclo no avanza

#### Solución:

- Revisar el Monitor Serial para ver en qué fase está


- Verificar que el código se esté ejecutando


---


## QUÉ VER EN EL MONITOR SERIAL

**Cuando funciona correctamente:**

text


 ENDPOINT ESP32-C3 - INICIANDO


 Inicializando RS485... ☒ OK

 Inicializando LoRa... ☒ OK


 INICIANDO DESCUBRIMIENTO...

 RS485 >>> ADDR:BROADCAST | CMD:DISCOVERY

 RS485 <<< ADDR:01 | CMD:DISCOVERY\_RESP...

 Sensor: 01 - Silo\_Norte


 INICIANDO LECTURA...

 ENVIANDO POR LoRa...

☒ Datos enviados - Sensores: 1

---

## PRÓXIMOS PASOS

1. ☒ **CONECTAR LOS SENSORES ESP8266** al bus RS485
2. ☒ **ALIMENTAR TODO EL SISTEMA** con 3.3V estable
3. ☒ **VERIFICAR COMUNICACIÓN** en Monitor Serial
4.  **PREPARAR GATEWAY** para recibir datos LoRa