



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Máster en Ingeniería Mecatrónica



Escuela Técnica Superior de Ingeniería del Diseño

Diseño Electrónico Avanzado

Comunicación serie RS-485 con Raspberry Pi



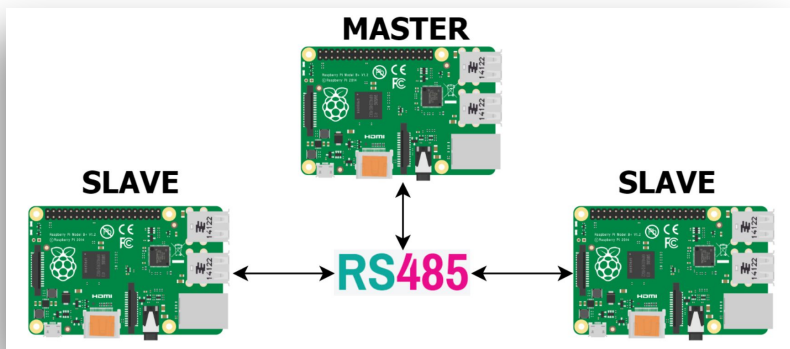
Aridane Álvarez Suárez
Jorge Molino Díaz

Valencia, Diciembre 2019

1. Objetivo
2. Solución adoptada
3. Programa esclavo
4. Programa maestro
5. Funcionamiento conjunto
6. Raspberry como Gateway
7. Conclusiones

1. Objetivo

Desarrollo de un protocolo de comunicación entre 3 Raspberry Pi sobre un bus serie RS-485 multipunto



- 1) Programa esclavo que reciba, procese y devuelva tramas
- 2) Programa maestro de orquestación
- 3) Depuración de funcionamiento conjunto

2. Solución adoptada

Formato de trama



Comando	Valor	Parámetro	Descripción
Hola	0x01	0	Respuesta de esclavo frente a la búsqueda
Cambio estado LED rojo	0x11	0	LED rojo en estado solicitado
Cambio estado LED amarillo	0x13	0	LED amarillo en estado solicitado
Lectura estado pulsador S1	0x31	0	Pulsador S1 sin presionar
		1	Pulsador S1 presionado
Lectura estado pulsador S2	0x33	0	Pulsador S2 sin presionar
		1	Pulsador S2 presionado
Lectura carga CPU	0x41	X	Informe del uso de CPU (en %)
Lectura temperatura CPU	0x43	X	Informe de la temperatura de la CPU esclavo (en C)

Cuadro 2: Tabla de comandos que envían los esclavos

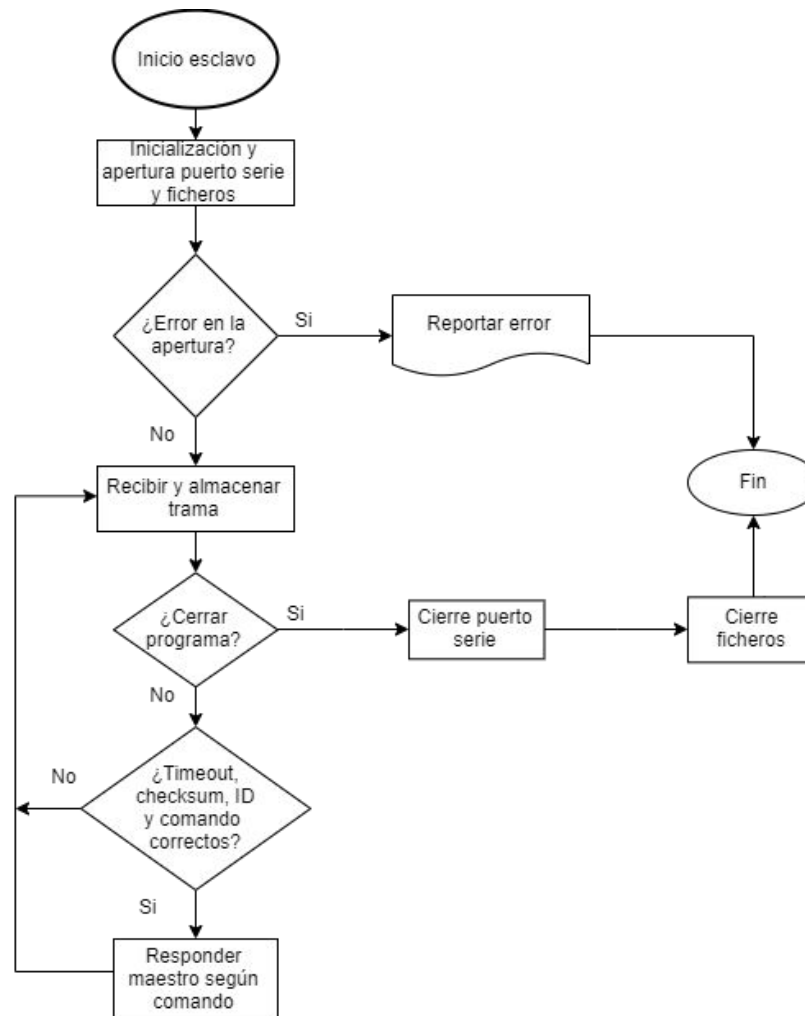
2. Solución adoptada

Comprobación de trama recibida:

- 1) Almacenar mensajes en buffer
- 2) Número de caracteres 6
- 3) Comprobar checksum
- 4) Comprobar dirección de destino
- 5) Procesar mensaje

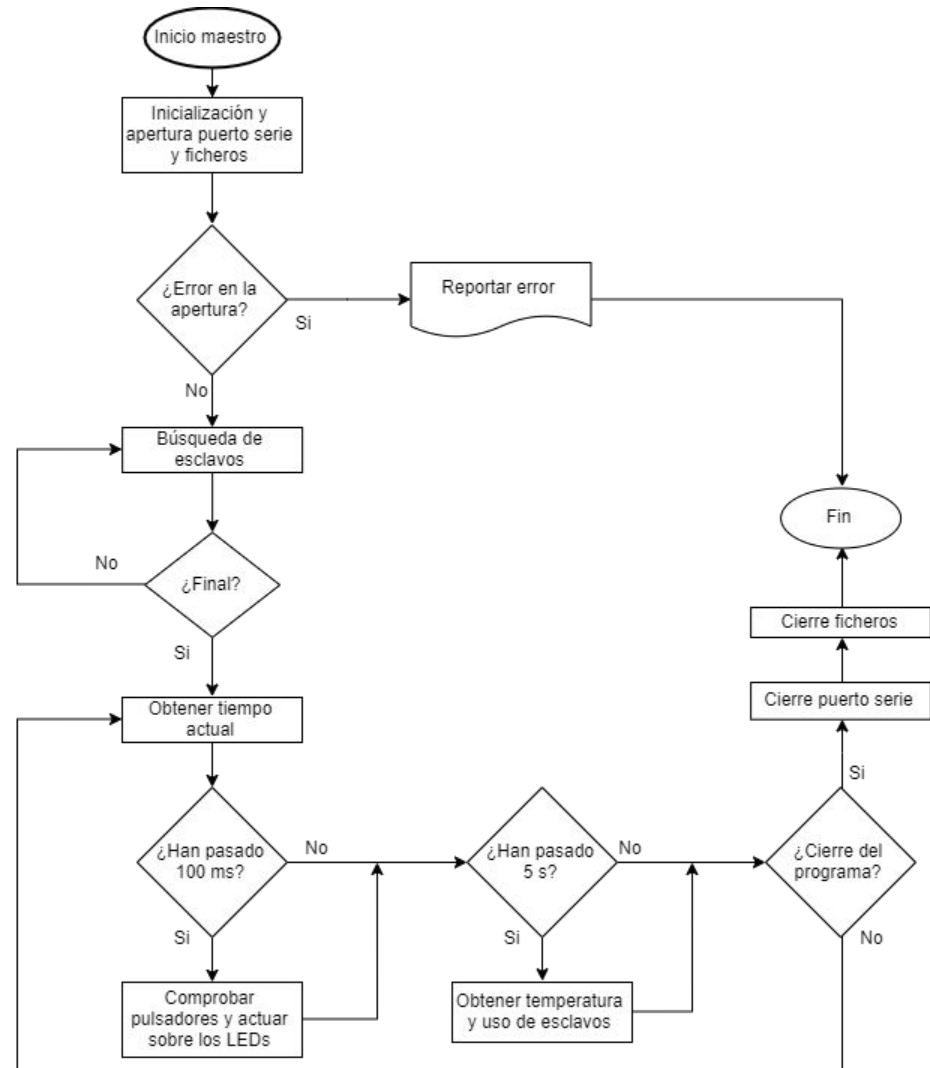
```
1 timeout_occurred = serial_read(serial, buffer_in, 6, timeout);
2 if (timeout_occurred!=0) {
3     if(checksum(buffer_in)){
4         if(buffer_in[1] == ID) {
5             //PROCESADO
6         }
7     }
8 }
```

3. Programa esclavo



4. Programa maestro

- Evitar bloqueo de delays en programa.
- Aproximación de multitarea con millis()



5. Funcionamiento conjunto

- Funcionamiento correcto de los dos programas mediante la utilidad hércules
- Tramas erróneas en esclavo 2 y 3 enviadas por el maestro al juntar los 3 programas
- Colisiones en half-duplex al enviar varios dispositivos tramas al mismo tiempo
- Necesidad de arbitraje
- Modificación del maestro

```
01, 03, 12, 00, 01, 17
DB, FD, 9D, FF, FF, FF
01, 02, 30, 00, 00, 33
01, 02, 30, 00, 00, 33
01, 03, 12, 00, 00, 16
EF, FD, 77, FF, FD, 00
01, 02, 30, 00, 00, 33
01, 02, 30, 00, 00, 33
01, 03, 12, 00, 01, 17
03, 01, 13, 00, 00, 17
01, 02, 30, 00, 00, 33
01, 02, 30, 00, 00, 33
01, 03, 12, 00, 01, 17
F7, FF, 67, FF, FE, 00
```


5. Funcionamiento conjunto

Códigos de error en la espera de respuesta del esclavo

- 255: Timeout en recepción de datos
- 254: Checksum de la trama incorrecto
- 253: ID de trama recibida no coincide
- 252: Comando del mensaje recibido no coincide con el esperado

```
1 // Esperar trama de respuesta
2 char receive_msg(char com) {
3     serial_read(...);
4     return check();
5 }
```

5. Funcionamiento conjunto

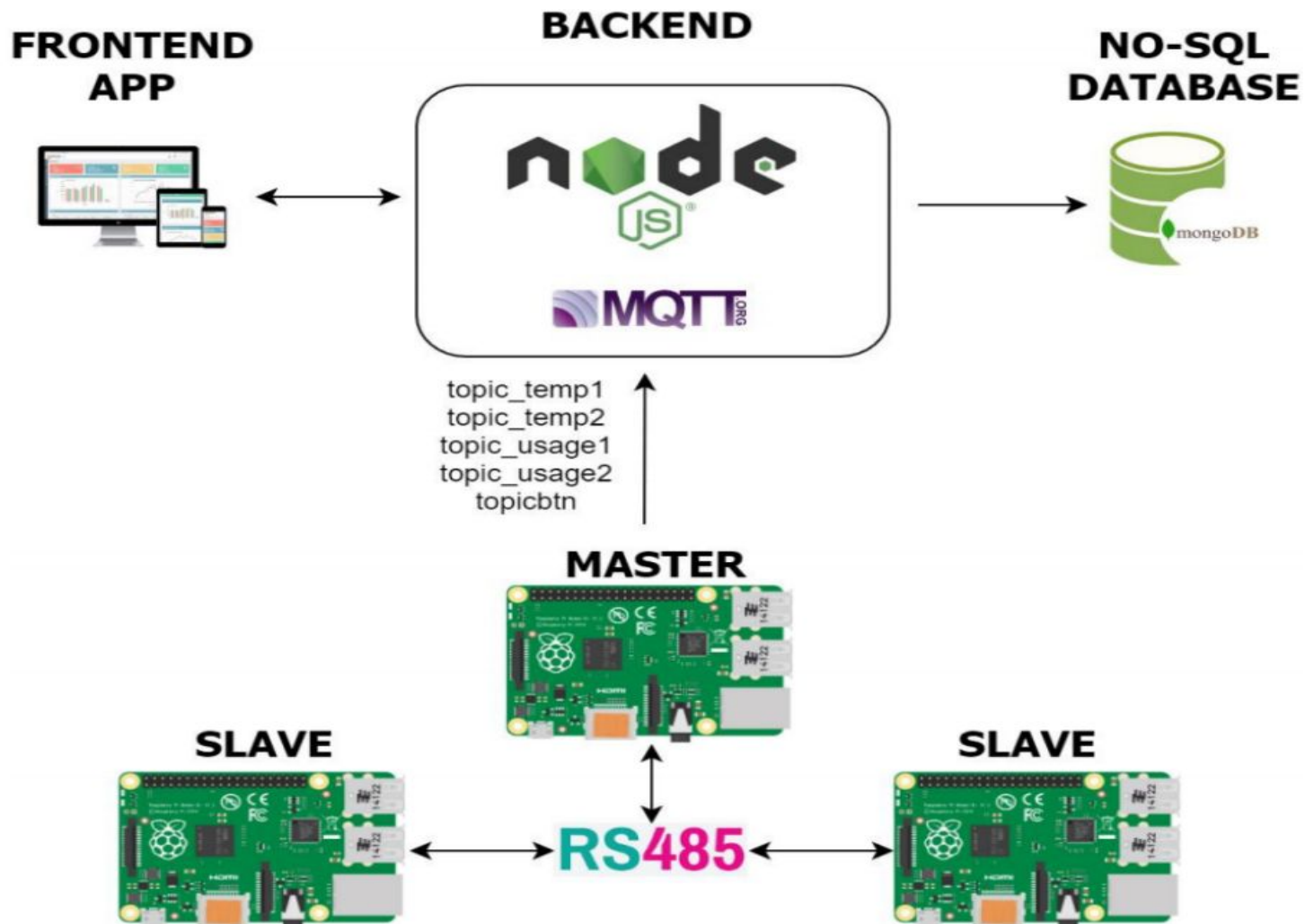
Aplicación final sin errores de tramas

```
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se han encontrado suficientes raspberrys para el programa!

-----
(program exited with code: 255)
Press return to continue
```

```
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se ha producido un timeout en la espera del mensaje
Se han encontrado 2 raspberrys con IDs: 5, 10
Uso de CPU de raspberry con ID 5: 0 %
Temperatura de raspberry con ID 5: 44 °C
Uso de CPU de raspberry con ID 10: 47 %
Temperatura de raspberry con ID 10: 42 °C
```

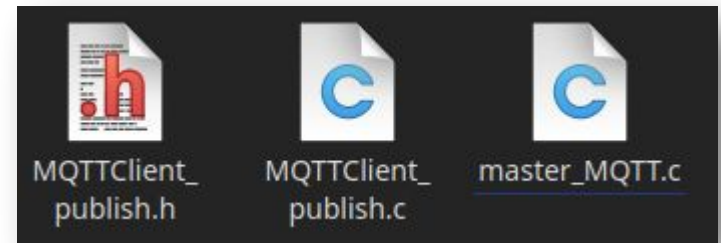
6. Raspberry como Gateway



6. Raspberry como Gateway

Modificación maestro

Creación de ficheros que faciliten el uso de la librería MQTT



```
1 #define ADDRESS      "ws://rpi-gateway.herokuapp.com:80/"
2 #define CLIENTID      "RPI_master"
3 #define QOS           0
4 #define TIMEOUT       10000L
5
6 void init_MQTT();
7 int send_string_MQTT(char *msg, char *topic);
```

6. Raspberry como Gateway

Despliegue aplicación

Ejecución en la nube mediante PaaS



HEROKU

+



mongoDB®

Atlas

```
_id: ObjectId("5dd42178bd47e60017ab3e10")  
log: "02, 01, 31, 00, 00, 34"  
fecha: "Tue Nov 19 2019 19:08:08 GMT+0000 (Coordinated Universal Time)"
```

```
▼ rpi-gateway.herokuapp.com  
log = 02, 01, 31, 00, 00, 34  
► $SYS (1 topic, 10 messages)  
topic_btn = 2
```

6. Raspberry como Gateway

Aplicación frontend

- App multiplataforma mediante framework Flutter:
 - Basado en Dart (POO)
 - Desarrollado en C++
- Librería “mqtt_client” de Dart
- Diseño en Adobe XD
- Respuesta en tiempo real al estado de los botones de los esclavos



7. Conclusiones

- Se ha logrado el objetivo principal tras cumplir los objetivos parciales y los problemas que se van generando
- El uso de la herramienta hércules ha sido esencial para corrección de errores
- Los programas se ejecutan con fluidez debido a evitar bloqueos con `Delay()`
- Se ha conseguido utilizar la Raspberry como puerta de enlace entre un protocolo clásico, bus serie RS-485, y otro moderno como MQTT



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA