

Diseño y Desarrollo de una Red MODBUS RTU basada en Arduino

DHI 2019~20 :: Alonso Rodriguez

Introducción

- **Objetivo**
 - Intercomunicar diferentes sensores y actuadores de diversa naturaleza, fabricantes, protocolos...
 - En definitiva: Abstraerlos
- **Obstáculo**
 - No todos los sensores y actuadores utilizan el mismo protocolo de transmisión de la información, y no se pueden acoplar en una red
- **Solución**
 - Acoplar un arduino para que actúe como “traductor”

Introducción: Tecnologías

MODBUS

Público y gratuito

Fácil de implementar, y
no requiere mucho
desarrollo

Maneja bloques de datos
sin restricciones

Sencillo y fiable

RS485

Estándar

Transmisión diferencial

Ampliamente conocido

Arduino

HW/SW Libre

Bajo coste

Versatilidad

Amplia comunidad

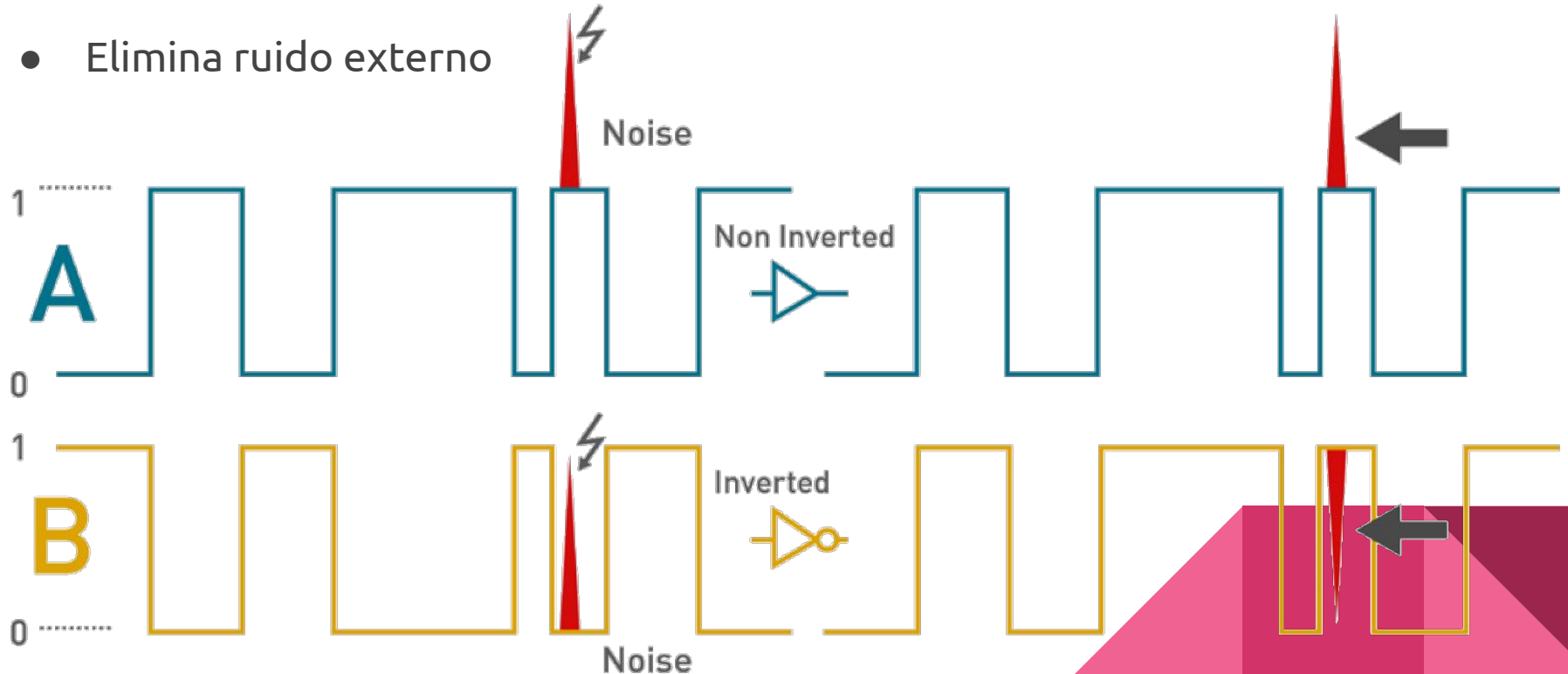
RS485

- Transmisión diferencial
- PtP, PtMP
- 256+ dispositivos
- Comunicación mediante par trenzado
- Serial
- Asíncrono
- $v = 300\text{baud/s} \rightarrow 19.2\text{kbaud/s}$

RS-485	
Estándar	TIA/EIA-485-A
Medio físico	Par trenzado
Topología de red	Punto a punto, punto a multipunto, multi-drop
Modo de comunicación	Semiduplex, dúplex
Máximo de dispositivos	Originalmente 32, actualmente 256 e incluso más usando repetidores
Modo de operación	Diferencial
Niveles de tensión	-7V / +12V
"1" Lógico	Tensión positiva (B-A > +200mV)
"0" Lógico	Tensión negativa (B-A < -200mV)

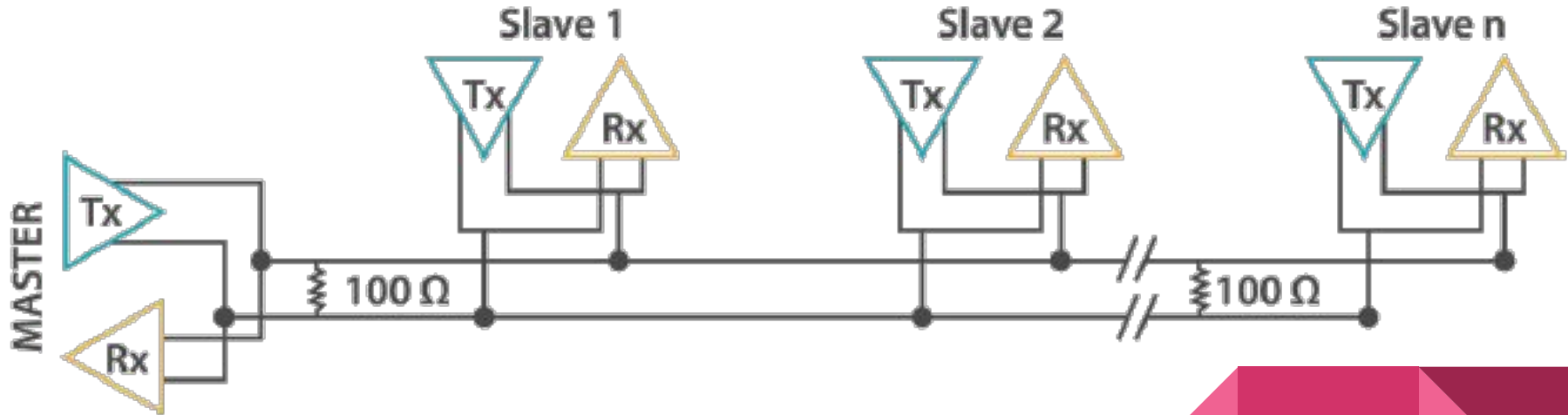
RS485: Transmisión diferencial

- Elimina ruido externo



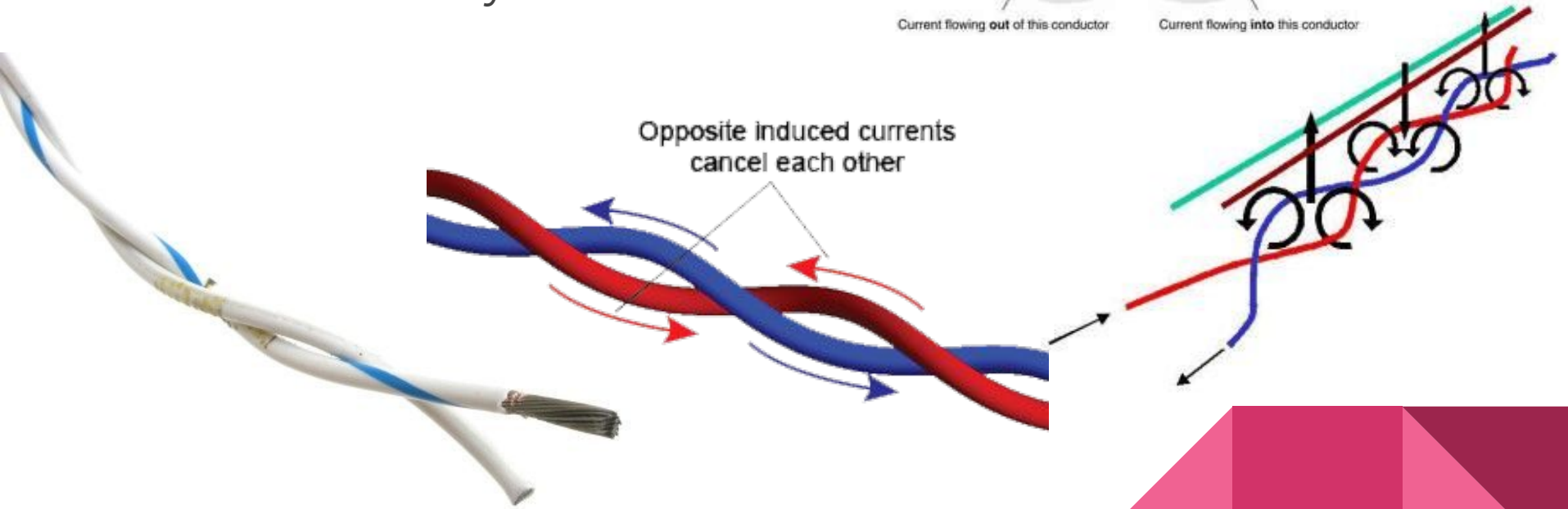
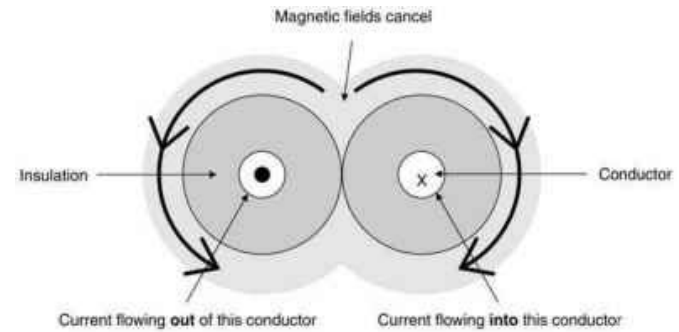
RS485: PtMP

- Todas las comunicaciones pasan por el maestro



RS485: Par Trenzado

- Minimiza crosstalk y el ruido externo



MODBUS

- Protocolo MODBUS Serie (Capa II), y MODBUS TCP/IP (Capa VII)
- Topología Maestro Esclavo(s)
- Dos modos
 - ASCII
 - RTU
- $v = 75\text{baud/s} \rightarrow 19.2\text{kbaud/s}$

Tipo de objeto	Acceso	Tamaño
Discrete input	Solo leer	1 bit
Coil	Leer/escribir	1 bit
Input register	Solo leer	16 bits
Holding register	Leer/escribir	16 bits

MODBUS: Trama

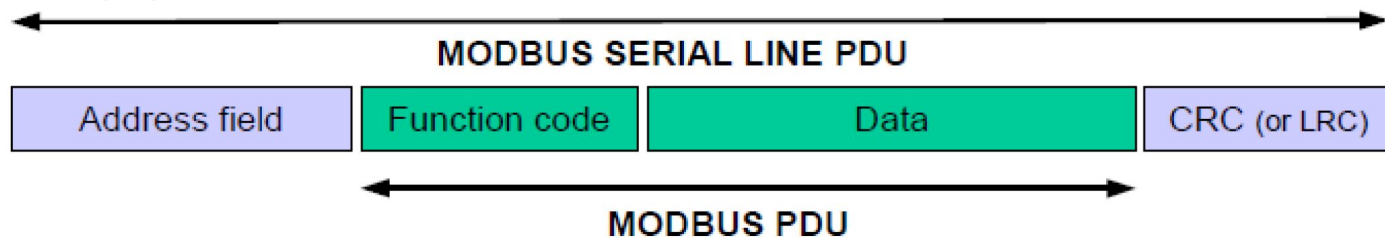
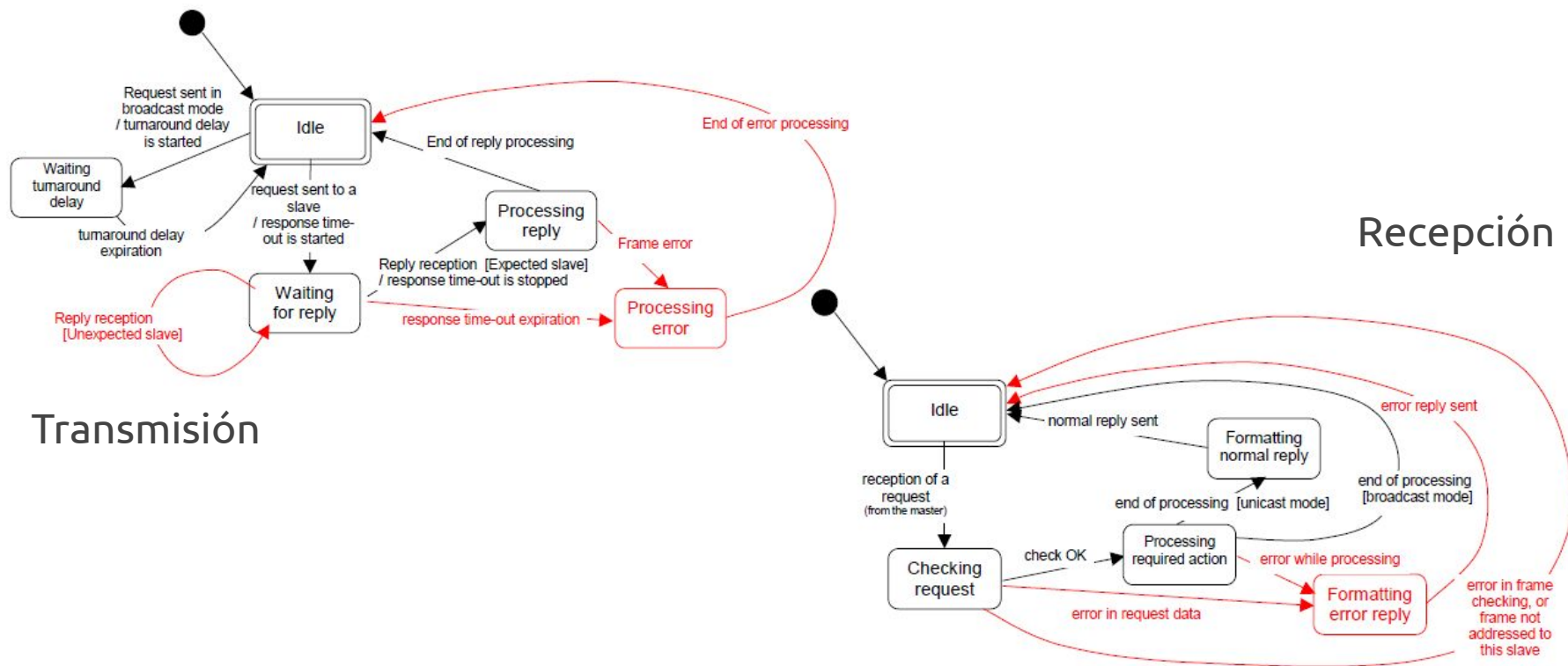


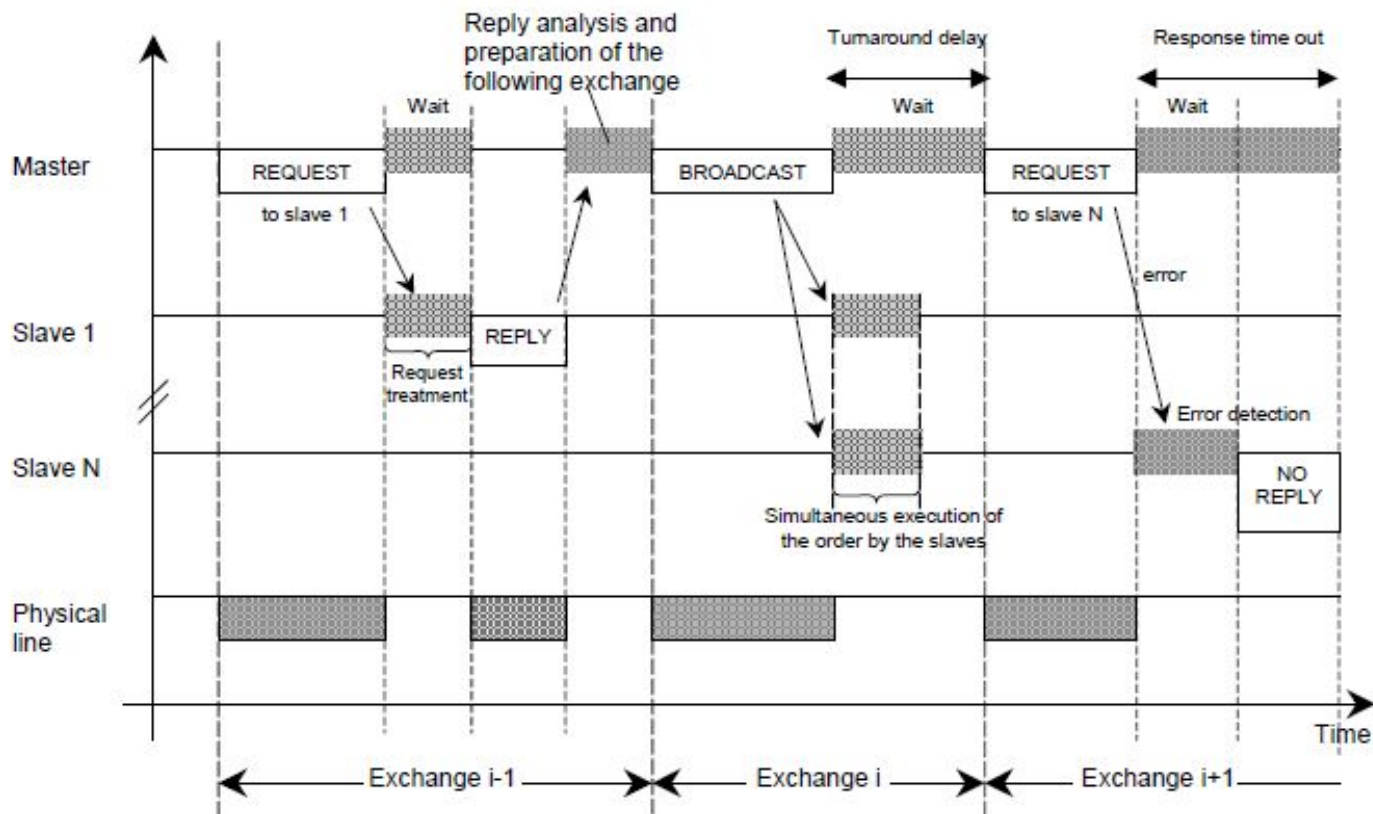
Figura 2-4: trama MODBUS serie

- **Address field o Campo de dirección:** Sirve para indicar la dirección del esclavo al que va dirigida la trama. El rango válido va desde 0 a 247, siendo el 0 la dirección de *Broadcast* y quedan reservadas las direcciones 248 a 255. Cuando el esclavo recibe una trama dirigida a él; construye la respuesta y pone su propia dirección en este campo, para que el maestro sepa de qué esclavo viene la respuesta.
- **Function code o Código de función:** Indica el código de la operación que el maestro solicita al esclavo; por ejemplo, leer un determinado registro.
- **Data o Campo de datos:** Lleva la información que se necesite para realizar determinada función; por ejemplo, escribir un valor en el registro indicado.
- **CRC o LRC:** Chequeo de redundancia cíclica o chequeo de redundancia longitudinal: sirve para asegurarse de que la información llega sin errores.

MODBUS: Diagrama de Estados



MODBUS: Transmisión



MODBUS: Formato de Trama - RTU

Con chequeo de paridad

Inicio	1	2	3	4	5	6	7	8	Par	Stop
--------	---	---	---	---	---	---	---	---	-----	------

Sin chequeo de paridad

Inicio	1	2	3	4	5	6	7	8	Stop	Stop
--------	---	---	---	---	---	---	---	---	------	------

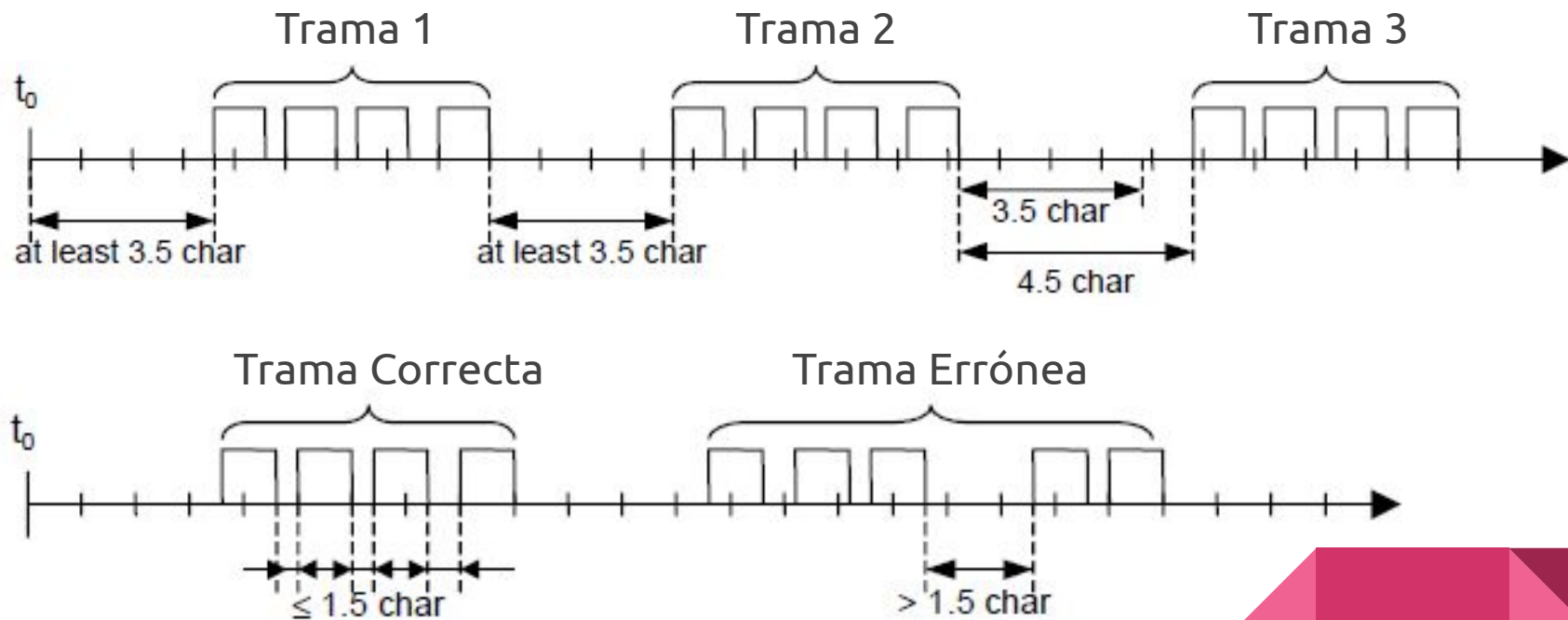
MODBUS message

Start
≥ 3.5 char

Address	Function	Data	CRC Check
8 bits	8 bits	N x 8 bits	16 bits

End
≥ 3.5 char

MODBUS: Formato de Trama - RTU (II)



MODBUS: Formato de Trama - ASCII

Con chequeo de paridad

Inicio	1	2	3	4	5	6	7	Par	Stop
--------	---	---	---	---	---	---	---	-----	------

Sin chequeo de paridad

Inicio	1	2	3	4	5	6	7	Stop	Stop
--------	---	---	---	---	---	---	---	------	------

Start	Address	Function	Data	LRC	End
1 char :	2 chars	2 chars	0 up to 2x252 char(s)	2 chars	2 chars CR,LF

MODBUS: Formato de Trama - ASCII vs RTU

RTU: 11 bits vs ASCII: 10 bits



MODBUS: Formato de Trama - ASCII vs RTU

RTU: 11 bits

ASCII: 10 bits



MODBUS: Formato de Trama - ASCII vs RTU

RTU

Con chequeo de paridad

Inicio	1	2	3	4	5	6	7	8	Par	Stop
--------	---	---	---	---	---	---	---	---	-----	------

Sin chequeo de paridad

Inicio	1	2	3	4	5	6	7	8	Stop	Stop
--------	---	---	---	---	---	---	---	---	------	------

ASCII


Con chequeo de paridad

Inicio	1	2	3	4	5	6	7	Par	Stop
--------	---	---	---	---	---	---	---	-----	------

Sin chequeo de paridad

Inicio	1	2	3	4	5	6	7	Stop	Stop
--------	---	---	---	---	---	---	---	------	------

Arduino

- Hardware y Software Libre
 - Bajo coste
 - Versatilidad
 - Amplia comunidad
 - Algunos modelos no destacan por su elevada eficiencia energética
 - etc
- 

Material

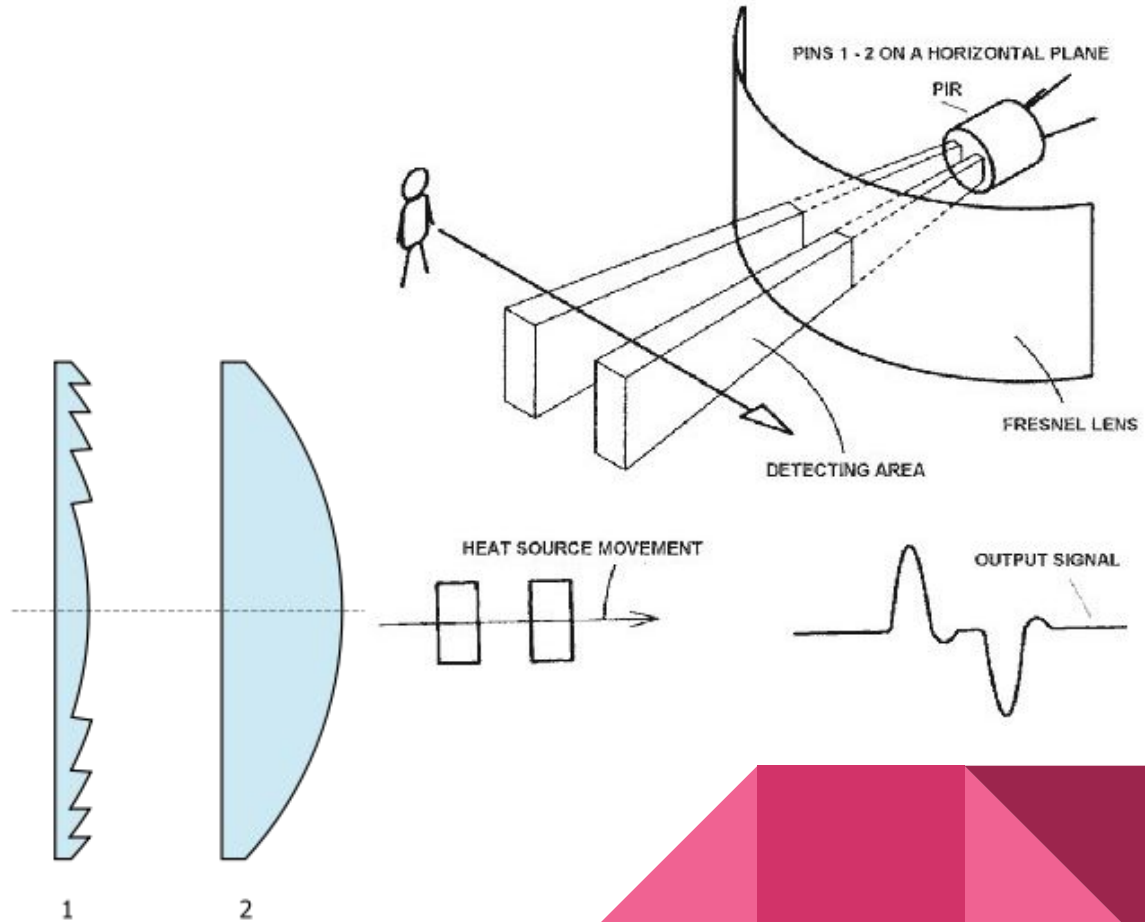
- Arduino UNO Rev3
- Módulo Multiprotocol
- Módulo MODBUS/RS-485
- Sensor PIR
- Servo Motor
- Protoboard
- Cables jumper para protoboard
- Diodo LED
- Potenciómetro

- Librería <ArduinoRS485-MODBUS>

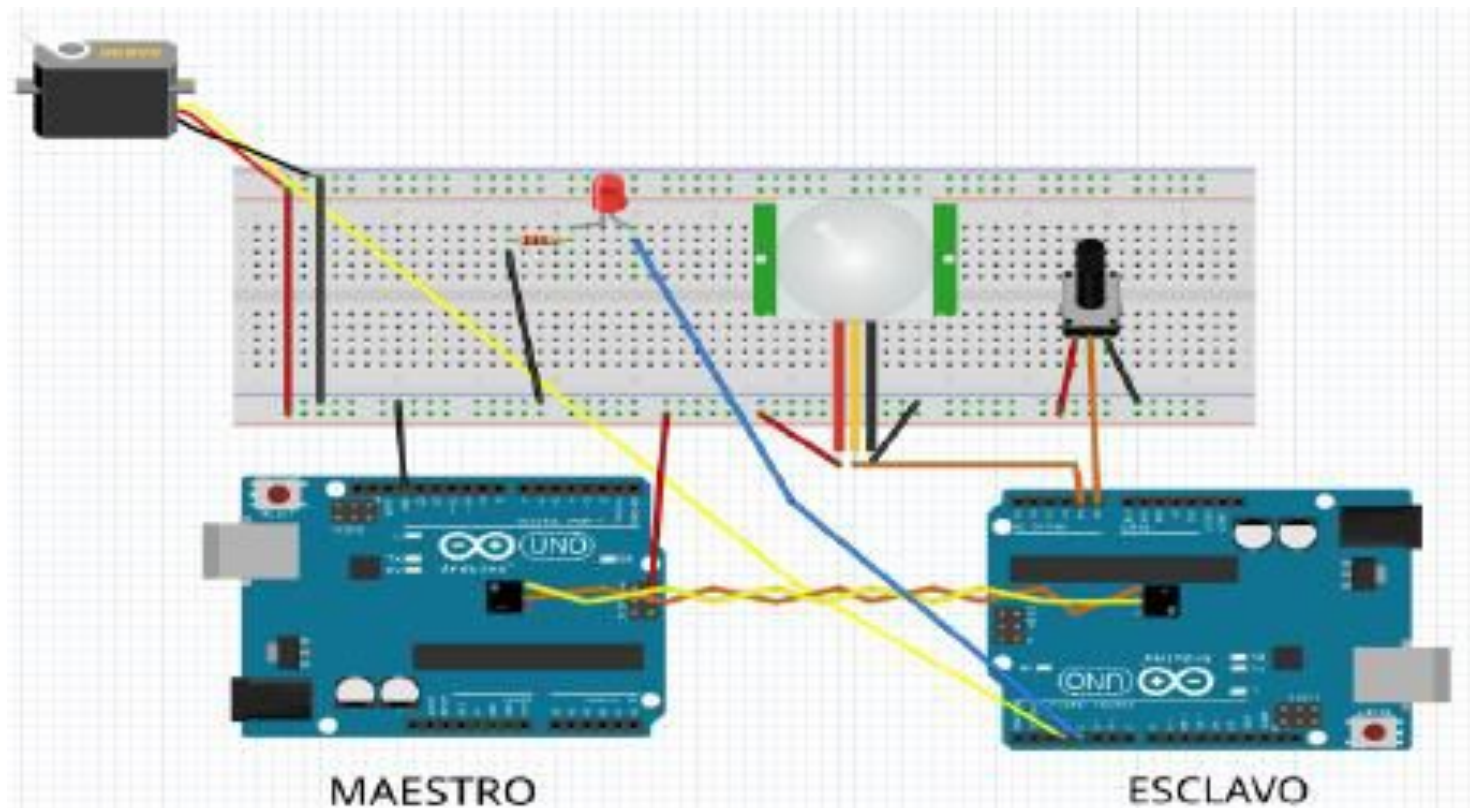


Sensor PIR

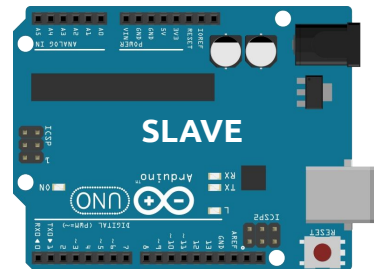
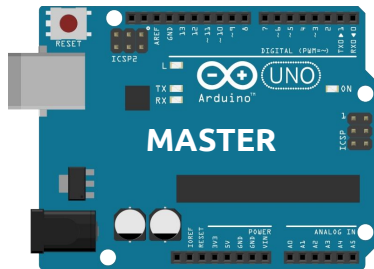
- Passive InfraRed sensor



Primer escenario



Flujo de tramas



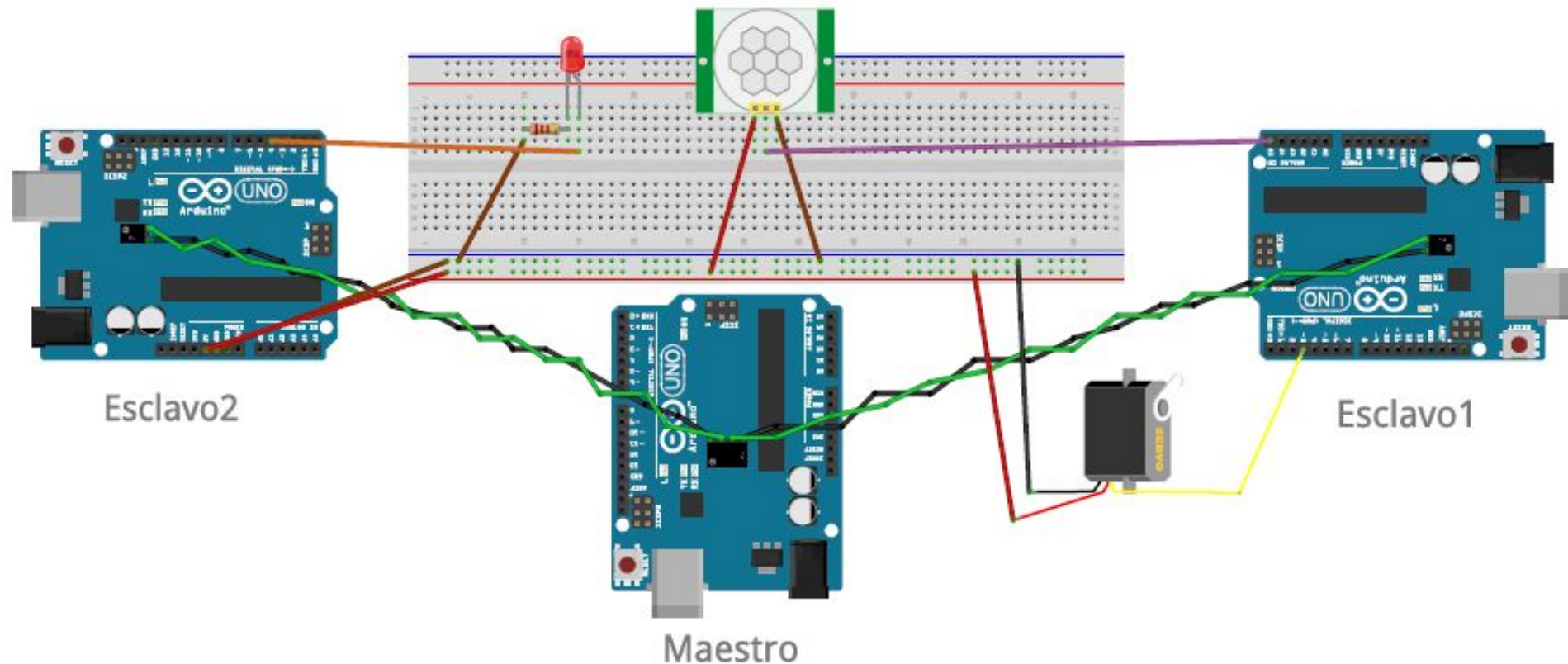
Dirección esclavo	Código de función	Dirección de inicio	Nº de registros	CRC
01	03	00 00	00 01	----

Dirección esclavo	Código de función	Bytes de respuesta	Estado de los bits	CRC
01	03	02	00 00	----

Dirección esclavo	Código de función	Dirección del registro	Valor a escribir	CRC
01	06	00 00	00 00	----

Dirección esclavo	Código de función	Dirección del registro	Valor a escribir	CRC
01	06	00 00	00 00	----

Segundo escenario



Conclusiones

- Mucha teoría de MODBUS, esquemas claros y concisos
- Teoría > Práctica
- Teoría + Tutorial
- Nuevos conceptos que desconocía
- Base robusta para los proyectos expuestos en otras sesiones
- Expansibilidad del bus para mayor modularidad
 - > Desoldar resistencia de 120Ω en los transceptores intermedios



Crítica

- Teoría > Práctica
- Eficiencia energética
- Esquemáticos



?

