

DISEÑO Y DESARROLLO DE UNA RED MODBUS RTU BASADA EN ARDUINO

Alonso Rodriguez Iglesias – DHI 2019/20 – TGR 1

ÍNDICE

Introducción	3
RS485	4
MODBUS	5
Arduino y Componentes del Sistema	7
Conclusiones y Crítica	10

ÍNDICE DE FIGURAS

- Figura 1: Cabecera DB9.
- Figura 2: Tipos de Objeto en MODBUS.
- Figura 3: Estructura de Trama MODBUS.
- Figura 4: Envío correcto de varias tramas.
- Figura 5: Envío incorrecto de una trama.
- Figura 6: Diagrama de comunicaciones sobre el bus.
- Figura 7: Primer escenario.
- Figura 8: Segundo escenario

INTRODUCCIÓN

Este documento está complementado por las explicaciones que se dan en la exposición de la presentación, por eso mismo he prácticamente prescindido de las imágenes puesto que ya se encuentran en la presentación de diapositivas.

El objetivo del TFG que he leído parece ser el explicar como funciona el protocolo MODBUS, y posteriormente realizar un tutorial de como establecer una red de comunicaciones MODBUS RTU con Arduino, que se empleará en este caso para leer el estado de sensores, y en base a este, modificar el estado de los actuadores.

Las principales tecnologías que empleamos son MODBUS, protocolo estandarizado y ampliamente conocido en la industria, publico, y gratuito a la vez que sencillo y fiable, que nos permite manejar bloques de datos sin imponer restricciones y es sencillo de implementar, ya que no requiere mucho desarrollo.

Este MODBUS está desplegado sobre el protocolo de comunicación RS485, también ampliamente conocido y estandarizado, y que emplea transmisión diferencial para alcanzar distancias de hasta 1200m sin repetidores.

Finalmente realizaremos la interacción con la red utilizando los Arduinos como coordinadores, y productos de su ecosistema como el módulo para RS485; que como bien ya se habrán hartado de comentar mis compañeros, destaca por ser un ecosistema completamente libre, en hardware y software, de bajo coste y con una gran versatilidad y comunidad. Sin embargo, los productos que se emplean en este sistema (Arduino UNO Rev.3) no destacan por su alta eficiencia energética.

RS485

Este es un protocolo como bien dije arriba estándar y ampliamente conocido.

Sus características más atractivas son la transmisión diferencial (y por tanto su capacidad de llegar más lejos que una versión sin este método de transmisión), y que se puede configurar en modo PtP (Point to Point), PtMP (Point to MultiPoint) y es multidrop, es decir, que múltiples máquinas pueden enviar tramas en el medio.

Soporta 256 dispositivos en su versión, o más aún si añadimos repetidores o sistemas similares. Además, en los extremos del bus debemos añadir resistencias de terminación para evitar reflexiones en el bus.

La comunicación se puede realizar mediante un par trenzado, o mediante un cable terminado en una cabecera DB9 como el mostrado en la imagen a continuación.

Es serial y asíncrono y soporta una velocidad de símbolo de 300 baudios por segundo hasta 19.2 kilobaudios por segundo.



Figura 1: Cabecera DB9

MODBUS

Este protocolo desarrollado por MODICON en 1979 por MODICON para integrar sus dispositivos PLC en el año 1979, publico, gratuito y altamente conocido y fácil de implementar, viene en dos versiones, MODBUS Serie, que trabaja en Capa II; y MODBUS TCP/IP, que trabaja en Capa VII y que se suele desplegar sobre redes Ethernet.

Tiene una topología de maestro a esclavo(s), y dos modos de transmisión, ASCII y RTU.

Puede operar a una velocidad de símbolo desde 75 baud/s hasta 19.2 kbaud/s.

Cuenta con cuatro tipos de objetos que son los siguientes:

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

Figura 2: Tipos de Objeto en MODBUS

El funcionamiento del protocolo consiste en el envío de tramas con una estructura:



Figura 3: Estructura de Trama MODBUS

En modo RTU, independientemente del modo de paridad que empleemos enviamos 11 bits en el canal por cada 8 de datos.

Esto puede llevar a confusión ya que en el modo ASCII enviamos 10 bits, pero nada más lejos de la realidad, en modo ASCII necesitamos dos tramas para enviar un byte de datos, por lo que necesitaríamos 20 bits.

Es decir, el modo ASCII es menos eficiente que el modo RTU, y este solo se utiliza cuando por razones de compatibilidad o similares, no se puede desplegar un MODBUS RTU.

Para el envío de la trama en bloques de 11 (RTU) o 10 (ASCII) bits, se envían con una pequeña diferencia temporal entre cada bloque de datos consecutivo, que nunca será mayor a la duración equivalente a transmitir 3.5 caracteres. Cuando se supera ese tiempo la trama se da por transmitida en el receptor, y en caso de que sea incorrecta, se descartará.

Para el envío de datos, el maestro del bus puede realizar un envío broadcast o un envío request. El primero es un envío a todos los esclavos para que ejecuten una acción todos los esclavos, y el segundo es una petición individualizada a la cual el esclavo responde.

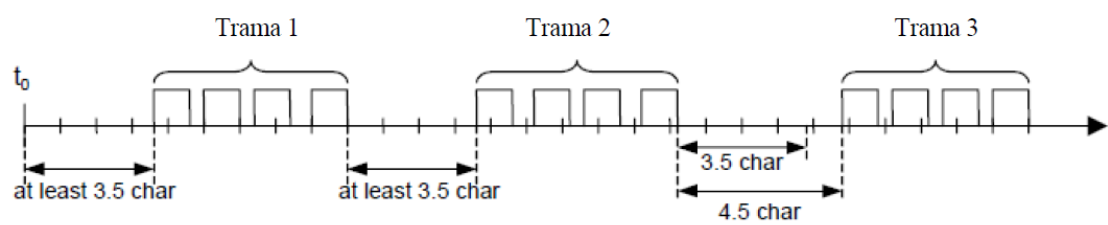


Figura 4: Envío correcto de varias tramas

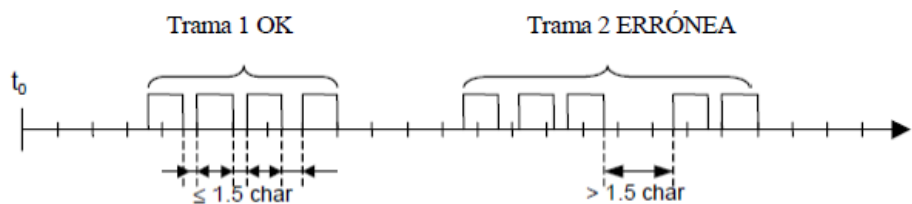


Figura 5: Envío incorrecto de una trama

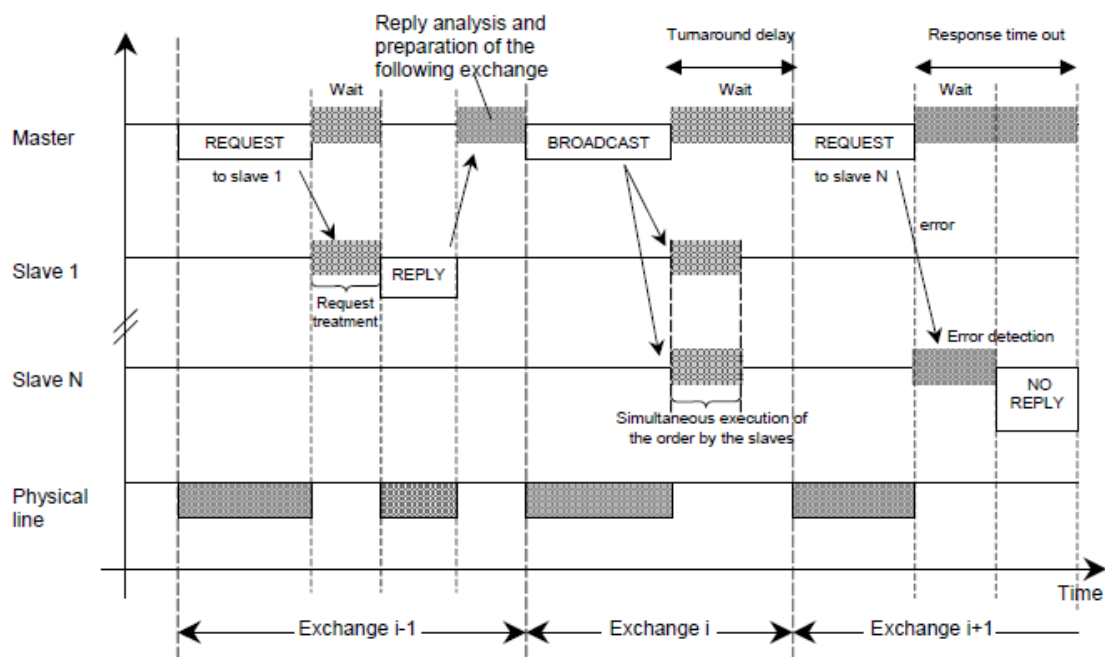


Figura 6: Diagrama de comunicaciones sobre el bus

ARDUINO Y COMPONENTES DEL SISTEMA

Arduino es una plataforma de Hardware y Software libre que surge fruto de la liberación del proyecto interno del instituto Ivrea en Italia.

En cuanto a los componentes, el Arduino que empleamos en este proyecto es el Arduino UNO Revisión 3, su modelo más famoso, que incorpora un ATmega AVR 328P.

Así, el listado de componentes necesarios será:

- Arduino UNO Rev.3
 - Módulo Multiprotocol
 - Módulo MODBUS/RS-485
 - Sensor PIR (Passive InfraRed)
 - Servo motor
 - Protoboard
 - Cables jumper para protoboard
 - Diodo LED
 - Potenciómetro
-
- Destacar también el uso de la librería `<ArduinoRS485-MODBUS>`.

Realmente el TFG realiza un tutorial (en mi opinión excesivamente extenso) de cómo instalar el software Arduino IDE, y me da la sensación de que habla poco del apartado del montaje. Continúo ahora, y en conclusiones y críticas comento más al respecto.

Primer escenario

El primer escenario que realizó el alumno es un modelo simplificado para facilidad en la depuración de errores y un acercamiento progresivo para entender el funcionamiento de la librería, en el cual simplemente establece conexión entre un maestro y un esclavo, para realizar pruebas de lectura y escritura. En este caso el maestro se limita a dar ordenes y el esclavo a ejecutarlas.

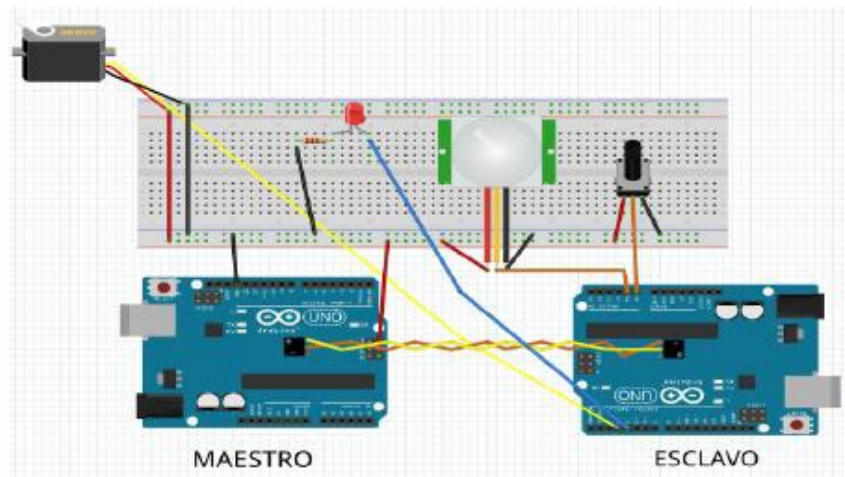


Figura 7: Primer escenario

Segundo escenario

En este segundo escenario, sigue siendo un proyecto de juguete, pero ya se puede apreciar mucho mejor la utilidad de este proyecto:

Se conecta un maestro a dos esclavos. Un esclavo cuenta con un PIR y un Servo, y el otro un LED.

Aquí tenemos un flujo de datos tal que:

1. Maestro pregunta repetidamente a esclavo por estado de PIR
2. Maestro envía señal para encender LED si hay movimiento, y para situar al servo en +90 grados.
3. Una vez no hay movimiento, se apaga el LED, y se devuelve el servo a su posición original de -90 grados.

Tendríamos un sistema de detección del movimiento por polling (encuesta).

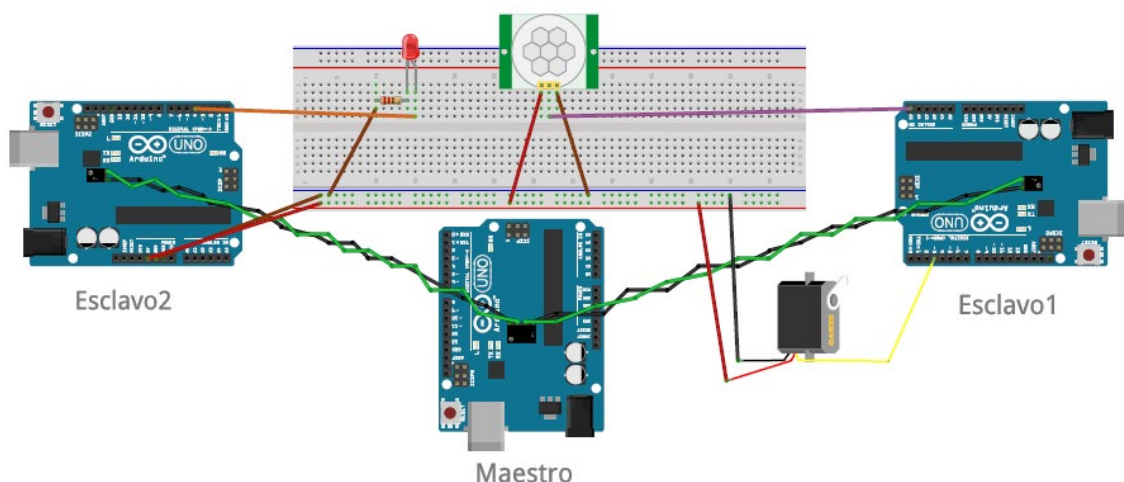


Figura 8: Segundo escenario

CONCLUSIONES Y CRÍTICA

El proyecto desde luego me parece de gran utilidad, en parte por mi desconocimiento del mundo industrial, ya que no sé si este tipo de soluciones ya existen, pero independientemente de esto último, el desarrollo de buses, comunicaciones, y cualquier medio de transmisión de información son una de mis ramas favoritas de la ingeniería, y me parece que este alumno ha realizado un buen trabajo abstrayendo mediante un Arduino varios dispositivos. En este caso son dos proyectos de juguete, pero por ejemplo podríamos tener algún sistema más complejo que se beneficiase enormemente de este sistema. Pongamos por ejemplo un sistema de cronómetros múltiples centralizados (se emplea en algunas tiendas de Zara para contabilizar tiempo en probadores); esta sería una muy buena solución en mi opinión, ya que se puede emplear el Arduino para implementar el funcionamiento del propio cronómetro, dándole más versatilidad a la solución, y emplearlo también como nodo del bus.

Por otro lado, quizás me ha parecido demasiado denso en la parte de explicación de MODBUS y muy superficial en la parte de montaje y componentes, que es precisamente la que he de comentar. Así que he decidido hacer una pequeña introducción a MODBUS para entender mejor qué está ocurriendo por debajo, que a nivel personal es algo que me interesa mucho por las razones que he expuesto arriba, y cuya sección en el TFG he disfrutado leyendo (a pesar de que algunas partes estén un poco ambiguas).

He echado en falta que se hable de la eficiencia, (es un tema que creo que a la mayoría de ingenieros nos preocupa) y de la posible capacidad de cambiar estos Arduinos por alternativas de más bajo consumo y/o coste.

También he echado en falta esquemáticos de los circuitos, pero son circuitos sencillos así que con las imágenes proporcionadas no ha habido mayor problema.

En mi opinión es un buen trabajo, sencillo, pero con potencial.

Alonso Rodriguez Iglesias. 30-Marzo-2020