



FACULTAD DE INFORMÁTICA
UNIVERSIDAD POLITÉCNICA DE MADRID

TESIS DE MÁSTER
MÁSTER UNIVERSITARIO EN INTELIGENCIA ARTIFICIAL

**Sistema IoT basado en el estándar BACNET
para el control del consumo eléctrico en
edificios de grandes dimensiones**

Autor: Jose Manuel García Fernández

Tutor: Miguel García Remesal

26 de junio de 2017

Resumen

El principal objetivo de esta tesis es realizar una investigación para comunicar de la mejor forma posible aparatos de conteo de consumo eléctrico de un edificio, siendo el problema principal que los contadores implementan un protocolo diferente al resto del sistema.

Se va realizar una investigación sobre todos los protocolos existentes que puedan servir para cumplir el objetivo anteriormente descrito, además también se deben de conocer las pasarelas comerciales existentes, que se encargan de realizar la conversión del protocolo ModBus al protocolo BACnet, que son los que se van a utilizar en este sistema.

Para realizar la traducción entre estos dos protocolos, se va crear un sistema básico con las tecnologías actuales, basado en las investigaciones realizadas previamente y tratando de mejorar algunos aspectos de las pasarelas comerciales existente, como puede ser un ahorro de dinero o la posibilidad de ampliación para que sea compatible con otros protocolos.

Por último, la aplicación va a ser desplegada en un servidor por lo que todo el mundo va a poder acceder a ella y utilizarla.

Palabras clave: Protocolo, Registro, Traducción, Pasarela Comercial

Abstract

The main objective of this thesis is to carry out an investigation to communicate the electrical counting devices of a building in the best possible way, the main problem is that the meters implement a different protocol to the rest of the system.

An investigation is going to be carried out on all the existing protocols that can serve the above objective. In addition, there are a lot of commercial gateways, which are responsible for converting the ModBus protocol to the BACnet protocol. These two protocols are going to be used in our system.

In order to carry out the translation between these two protocols, a basic system with the current technologies will be created, based on previous researches and trying to improve some aspects of the existing commercial gateways.

Finally, the application will be deployed on a server so everyone will be able to access it and use it.

Key words: Protocol, Register, Translation, Commercial Gateway

Índice general

Índice general	V
Índice de figuras	VII

1	Introducción	1
1.1	Motivación	2
1.2	Objetivos	3
1.3	Alcance	3
2	Estado del arte	5
2.1	Protocolos existentes	5
2.1.1	KNX	5
2.1.2	X2D/X3D	7
2.1.3	CANopen	7
2.1.4	ModBus	9
2.1.5	MBUS	10
2.1.6	X10	12
2.1.7	BACnet	13
2.1.8	XBee	15
2.2	Conversión de Modbus a BACnet	16
2.2.1	Pasarelas comerciales	16
2.2.2	Software	20
2.3	Formatos de mapping	20
3	Análisis y Desarrollo	23
3.1	Definición del sistema	23
3.2	Herramientas, Tecnologías y Plataformas	24
3.3	Interfaz de usuario	24
3.3.1	Página de inicio	24
3.3.2	Configuración del registro	25
3.3.3	Resumen de registros	26
3.4	Implementación del sistema	26
3.4.1	Diseño del fichero de salida	26
3.4.2	Elección de los datos a introducir por el usuario	28
3.4.3	Implementación de la aplicación	29
3.4.4	Despliegue de la aplicación Web	29
4	Discusión de los resultados y líneas futuras	31
5	Conclusiones	33
	Bibliografía	35

Apéndice		
A	Manual de usuario	37

A.1 Acceso a la web	37
A.2 Añadir un registro	37
A.3 Ver los registros introducidos	39
B Código fuente	41

Índice de figuras

1.1	Configuración descentralizada de dispositivos, obtención de datos de consumo y comunicación con dispositivos para controlarlo. [Figura].	2
2.1	Conexión bidireccional con KNX [Figura].	6
2.2	Bus CANopen [Figura].	8
2.3	Conexión Maestro-esclavo en ModBus [Figura].	9
2.4	Diagrama de bloques que muestra el principio del sistema M-Bus [Figura].	11
2.5	Transmisión de la señal en X10[Figura].	12
2.6	ANSI/ASHRAE Standard 135-2016(2006). Arquitectura colapsada de BACnet[Figura].	14
2.7	XBee Serie 1Figura.	15
2.8	Convertidor de protocolo Modbus a BACnet E8951[Figura].	18
2.9	Interfaz BACnet para objetos Modbus - Modulo[Figura].	19
2.10	Conexión de MODOBus/TCP a BACnet/IP[Figura].	19
2.11	Ejemplo de Fichero de mapeo en formato CSV.	21
2.12	Ejemplo de Fichero de mapeo en formato TXT.	21
3.1	Interfaz de inicio de la aplicación	25
3.2	Interfaz de configuración de un registro	25
3.3	Interfaz con el resumen de los datos introducidos	26
3.4	Fichero resultante de la aplicación	26
A.1	Interfaz de inicio de la aplicación	38
A.2	Interfaz para añadir una registro.	38
A.3	Interfaz para ver los registros añadidos hasta el momento.	39
A.4	Fichero resultante de la aplicación	39

CAPÍTULO 1

Introducción

En la actualidad está aumentando el uso de la domótica, para la automatización de tareas o el control de diversas actividades en los edificios, como puede ser el consumo eléctrico, la iluminación, la temperatura o la humedad. En este caso, nos vamos a centrar en comunicar aparatos para el conteo del consumo eléctrico de un edificio, con un diseño y organización parecido al de la figura 1.1, para conseguir esto hay una gran variedad de protocolos que nos pueden ayudar a conseguir el objetivo, así como una gran cantidad de dispositivos y contadores de diversas marcas y tipos.

Un factor muy importante en este tipo de tareas, es conocer todas las alternativas posibles, para que solución adoptada sea la mejor. Por ello se debe de realizar un análisis de todos los protocolos disponibles en el mercado y que sirvan para poder comunicar contadores de consumo eléctrico, una vez comparados entre ellos y tomada la decisión del protocolo a utilizar se deben de elegir los contadores a instalar. Hay una gran variedad de marcas que crean contadores y puede que muchos de ellos no sean compatibles con el protocolo que se ha elegido previamente. Pero aún en ese caso puede que la mejor solución sea realizar una traducción entre dos protocolos, uno el del contador y otro el del sistema.

En este caso nos vamos a centrar en construir un sistema con BACnet como protocolo principal, y los contadores de luz vas a tener implementado el protocolo ModBus, por lo que va a ser necesario crear un sistema que permita pasar de un protocolo a otro. Existen una gran cantidad de traductores de protocolos en el mercado, pero nos vamos a centrar solo en estos dos protocolos.

Primeramente, se va a hacer una breve descripción de cada uno de los protocolos que se van a utilizar:

- **Modbus:** Es un protocolo de solicitud-respuesta implementado usando una relación maestro-esclavo. En una relación maestro-esclavo, la comunicación siempre se produce en pares, un dispositivo debe iniciar una solicitud y luego esperar una respuesta y el dispositivo de inicio (el maestro) es responsable de iniciar cada interacción. Por lo general, el maestro es una interfaz humano-máquina (HMI) o sistema SCADA y el esclavo es un sensor, controlador lógico programable (PLC) o controlador de automatización programable (PAC). El contenido de estas solicitudes y respuestas, y las capas de la red a través de las cuales se envían estos mensajes, son definidas por las diferentes capas del protocolo

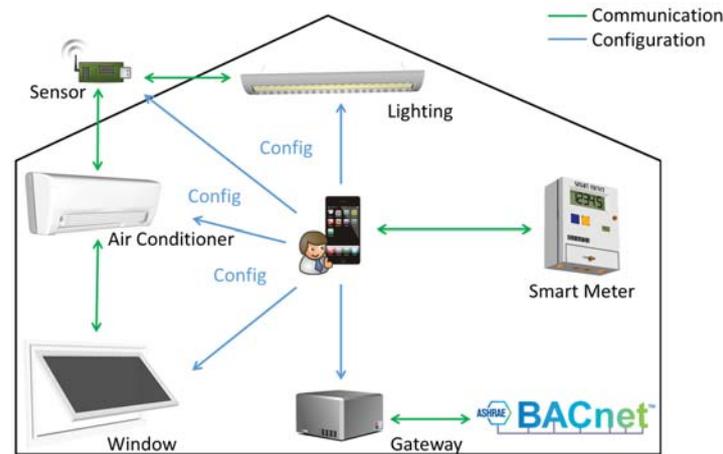


Figura 1.1: Configuración descentralizada de dispositivos, obtención de datos de consumo y comunicación con dispositivos para controlarlo. [Figura].

Recuperado de <http://www.ws4d.org>.

- BACnet:** Es una norma cuya finalidad es definir los servicios y protocolos de comunicación de datos entre equipos informáticos utilizados para vigilancia y control de HVAC&R y otros sistemas constructivos; definiendo una representación abstracta y orientada a objetos de la información comunicada entre dichos dispositivos. Por lo tanto esta norma facilita la aplicación y el uso de la tecnología de control digital en edificios. Este protocolo proporciona un conjunto completo de mensajes para transportar datos binarios, analógicos y alfanuméricos codificados entre dispositivos. Este protocolo modela cada computadora de automatización y control de edificios como una colección de estructuras de datos llamadas "objetos", cuyas propiedades representan varios aspectos del hardware, software y funcionamiento del dispositivo. Estos objetos proporcionan un medio para identificar y acceder a la información sin requerir el conocimiento de los detalles del diseño o configuración interna del dispositivo

1.1 Motivación

La motivación principal es que con el protocolo BACnet se puedan utilizar todo tipo de dispositivos, independientemente de la marca y del protocolo que tengas implementado internamente. Por lo tanto, se va a comenzar por dar soporte a los dispositivos que utilizan Modbus como protocolo predefinido. Para que esto sea posible se plantea el uso de una aplicación web sencilla, que sin tener demasiados conocimientos acerca de como funcionan ambos protocolos sea posible generar un fichero que realice la conversión entre un objeto de ModBus y uno de BACnet. Esta conversión es beneficiosa ya que el protocolo BACnet modela los objetos como una lista que puede ser visible para toda la red de dispositivos, lo que permite ver las propiedades de cada uno de ellos y entenderlas fácilmente, por el contrario con el protocolo Modbus es necesario un manual de usuario para comprender el significado de los registros.

Para conseguir esto, la idea principal es dar soporte al mayor número de objetos de cada protocolo, y que el usuario seleccione la conversión que desea realizar, para posteriormente introducir los datos necesarios para generar el fichero en el que se realiza la traducción.

1.2 Objetivos

El objetivo principal es realizar una investigación sobre todos los protocolos disponibles para comunicar aparatos de conteo de consumo, y decidir cual o cuáles de los protocolos son los más adecuados para resolver el problema planteado.

Un objetivo importante es la creación de una aplicación que realice la traducción entre el protocolo Modbus y el protocolo BACnet, dando como salida un fichero de texto. Para llevar esto a cabo de una forma correcta se deben de explorar las soluciones existentes en el mercado a este problema y tratar de crear algo que los mejore.

Otro de los objetivos principales es la disponibilidad del sistema, que debe de poder ser accesible por una gran cantidad de personas. Por otra parte, es muy importante que el sistema sea multiplataforma y pueda ejecutarse en varios sistemas operativos distintos.

El último de los objetivos planteados es que el sistema generado sea compatible con alguna de las implementaciones del protocolo BACnet existentes. Para que así sea más sencillo integrar nuestro sistema en una aplicación ya realizada y que utilice ese protocolo.

1.3 Alcance

La tesis se basa en realizar una investigación de los principales protocolos de comunicación que pueden ser utilizados para controlar el consumo de energía eléctrica de un edificio a través de varios contadores situados en él. Se va realizar una aplicación que va a permitir convertir objetos del protocolo Modbus a objetos del protocolo BACnet, el usuario podrá elegir el tipo de objeto en cada uno de los protocolos. La aplicación web va a tener una pantalla de inicio que será la de selección de los objetos y que posteriormente el usuario deberá de rellenar los atributos necesarios para realizar la conversión, el documento de salida de la aplicación será un fichero en formato csv que contenga la información necesaria para poder utilizar un objeto Modbus con el protocolo BACnet.

CAPÍTULO 2

Estado del arte

La interoperabilidad proporciona la capacidad para construir sistemas de control o que dispositivos de varios proveedores puedan comunicarse entre sí a través de protocolos estándar abiertos. Esto permite a los clientes tener la suficiente flexibilidad para elegir los productos que mejor satisfacen sus necesidades independientemente del protocolo que implementen. Por lo tanto es importante poder pasar los datos de un determinado protocolo al formato de otro.

2.1 Protocolos existentes

En la actualidad existen una gran cantidad de protocolos orientados al "Internet de las cosas (IoT)", en este apartado se van a enumerar algunos de ellos orientado a la automatización. Una de las principales diferencias entre todo este tipo de protocolos es la forma en la que se modelan los distintos objetos, ya que cada elemento tienen una función diferente. Los principales protocolos que se van a describir son los de la compañía Delta Core como son KNX, XDD y X3D, junto con ModBus, MBUS, X10, BACnet y XBee.

2.1.1. KNX

El protocolo KNX fue publicado en 2002 por la 'KNX Association', y proporciona una tecnología de control de edificios, a través del control automatizado de procesos, dedicada a las necesidades de las aplicaciones del hogar y del edificio, con un enfoque descentralizado, de ahí el término red. La red de dispositivos KNX resulta de la fusión de los tres principales sistemas para la automatización de edificios (EIB, EHS, BatiBus).

KNX proporciona varias alternativas de comunicación en la capa física, para que los fabricantes, puedan elegir entre todas ellas o incluso combinarlas. Con la disponibilidad de routers, y combinado con el poderoso Interworking, multimedia, y también las configuraciones de varios proveedores se pueden construir. Los diferentes medios son:

- Cableado de par trenzado (heredado del EIB): Utiliza una red SELV y un sistema de suministro. Las características principales son: datos y transmisión

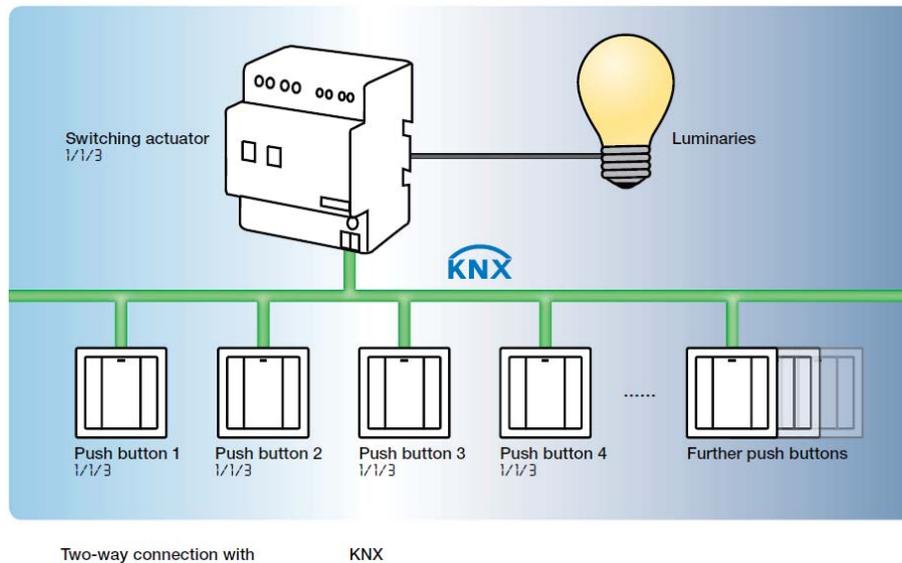


Figura 2.1: Conexión bidireccional con KNX [Figura].

Recuperado de <https://smart-automation.blogspot.com.es/p/eibknx.html>.

de potencia con un par (los dispositivos con consumo de energía limitado pueden ser alimentados por el bus), y transferencia de datos orientada a caracteres asíncronos y comunicación bidireccional semidúplex.

- Red eléctrica (heredado del EIB): permite la comunicación a través de la red de alimentación. Las principales características son: señalización de desplazamiento, por desplazamiento de frecuencia de propagación, transmisión asíncrona de paquetes de datos y comunicación bidireccional semidúplex. La distancia máxima que se puede lograr sin repetidor es de 600 metros.
- Radio: permite la comunicación a través de señales de radio en la banda de 868,3 MHz para dispositivos de corto alcance. Esta tecnología logra comunicar distancias de hasta 300 metros en campo abierto (condiciones ideales).
- Ethernet: usando el estándar Ethernet generalmente se utiliza como backbone entre segmentos, líneas y áreas KNX además de permitir la transferencia de telegramas a través del protocolo IP a ubicaciones remotas.

Aparte de las conexiones anteriores, todos los dispositivos KNX están conectados entre sí por un bus de dos hilos, permitiéndoles intercambiar datos. La función de los dispositivos de bus individuales se determina por su planificación de proyecto, que se puede cambiar y adaptar en cualquier momento. Se pueden identificar cuatro grupos de elementos:

- Actuadores: Los actuadores son los dispositivos que se conectan físicamente a los elementos que va a controlar en el edificio, y se encargan de realizar la traducción que viaja del mundo KNX al mundo físico conmutado, regulando o accionando los dispositivos que son controlados. Un ejemplo de ellos se puede ver en la figura 2.1.
- Sensores: Los sensores son los elementos que dentro del sistema se encargan de recoger datos o interpretar órdenes del usuario.

- Pasarelas: Las pasarelas o gateways son las encargadas de enlazar otros sistemas con otros protocolos de comunicación con KNX.
- Acopladores: Estos elementos se encargan de realizar una separación física dentro del bus agrupando los dispositivos en un segmento con unas características determinadas para la cantidad de equipos, ubicaciones físicas o funciones determinadas y conectarlo con otro segmento para obtener una mayor eficacia en el envío de datagramas a través del bus, para alcanzar mayores distancias (repetidores), además de darle un direccionamiento físico muy entendible utilizando la división de áreas, grupos y líneas.

2.1.2. X2D/X3D

El protocolo X2D o X3D [19], es un protocolo de comunicaciones desarrollado por Delta Dore y que surgió con el objetivo de gestionar técnicamente una instalación de cualquier tipo, ya sea calefacción, iluminación o riego. Para ello, hace uso de las líneas de corriente eléctrica como medio físico para la comunicación, con esto se reducen las labores de instalación de los equipos. En la actualidad, existen dos medios físicos sobre los cuales funciona este protocolo, las líneas de corriente eléctricas y la radiofrecuencia.

Los dispositivos Delta Dore que se utilizan en este sistema se pueden dividir en dos categorías:

- Receptores: Son dispositivos actuadores, y están disponibles en formato interruptor, sirven para controlar la iluminación, toldos, persianas... A cada uno de estos receptores se puede acceder desde ocho emisores diferentes.
- Emisores: Son los encargados de actuar a distancia sobre uno o varios receptores, posibilitando así el control de equipos desde múltiples puntos. Estos dispositivos no necesitan instalación y pueden ser colocados en cualquier lugar, ya que se alimentan a través de baterías. Pueden controlar tantos receptores como se deseen, facilitando la opción de crear ambientes, macros o escenarios en los que, con sólo una pulsación sobre un botón, se desaten una serie de acciones a gusto del usuario.

2.1.3. CANopen

CANopen [15] es un protocolo de comunicación de alto nivel, y está basado en el bus CAN. El estándar de CANopen consta de un esquema de direccionamiento, varios pequeños protocolos de comunicación y una capa de aplicación definida por un perfil de dispositivo. Los protocolos de comunicación tienen soporte para la gestión de red, supervisión de dispositivos y comunicación entre nodos, incluyendo una capa de transporte simple para la segmentación de mensajes. El protocolo de nivel inferior que implementa el enlace de datos y las capas físicas suele ser CAN (Controller Area Network), aunque los dispositivos que utilizan otros medios de comunicación (como Ethernet Powerlink, EtherCAT) también pueden implementar el perfil de dispositivo CANopen, las capas que forman este protocolo se pueden ver en la figura 2.2.

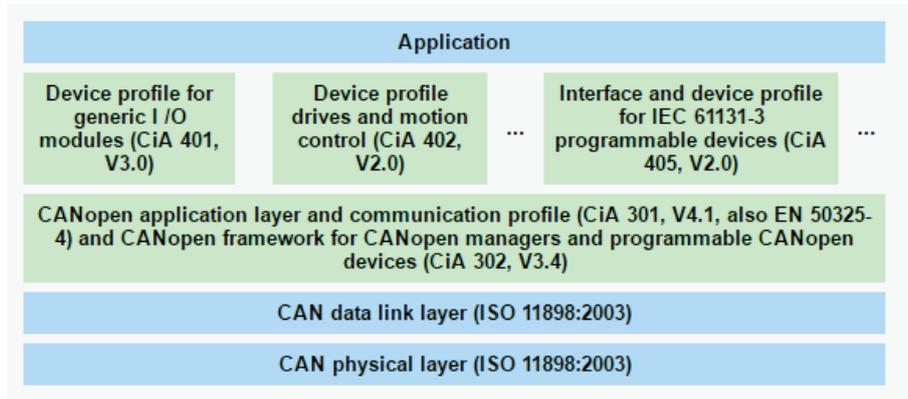


Figura 2.2: Bus CANopen [Figura].
Recuperado de www.canopensolutions.com.

El protocolo CANopen permite tanto la comunicación síncrona como asíncrona y admite tres tipos de relaciones de comunicación:

- **Maestro/Esclavo:** En todo momento hay un único maestro que dirige las comunicaciones para una determinada funcionalidad; el resto de dispositivos se consideran esclavos. El maestro hace una petición y el esclavo direccionado responde.
- **Cliente/Servidor:** Es una relación entre un sólo cliente y un sólo servidor. El cliente hace una petición que provoca que el servidor realice una determinada tarea. Tras finalizar la tarea, el servidor responde a la petición. En CANopen se utiliza para la parametrización de los dispositivos.
- **Productor/Consumidor:** Involucra a un productor y cero o más consumidores. Puede ser un modelo push, servicio no confirmado solicitado y ofrecido por el productor, que en CAN open se utiliza para transmitir datos a gran velocidad. O puede ser modelo pull, en el que el servicio confirmado es solicitado por el consumidor. En CANopen se puede usar para petición de datos con confirmación de entrega. No es tan rápido como el push.

Todos los dispositivos CANopen se pueden estructurar en tres unidades funcionales:

- **Comunicaciones:** Aquí se encuentran los objetos encargados de la comunicación, que proporcionan las funcionalidades necesarias para transportar los datos a través de la red.
- **Diccionario de objetos:** Se utiliza para la configuración y la comunicación con el dispositivo, y es una colección con todos los elementos de datos que tienen influencia en el comportamiento de los mismos. Cada objeto se direcciona utilizando un índice de 16 bits. Para permitir el acceso a elementos individuales de las estructuras de datos también existe un subíndice de 8 bits.
- **Aplicación:** Engloba la funcionalidad del dispositivo con respecto a la interacción con el proceso.

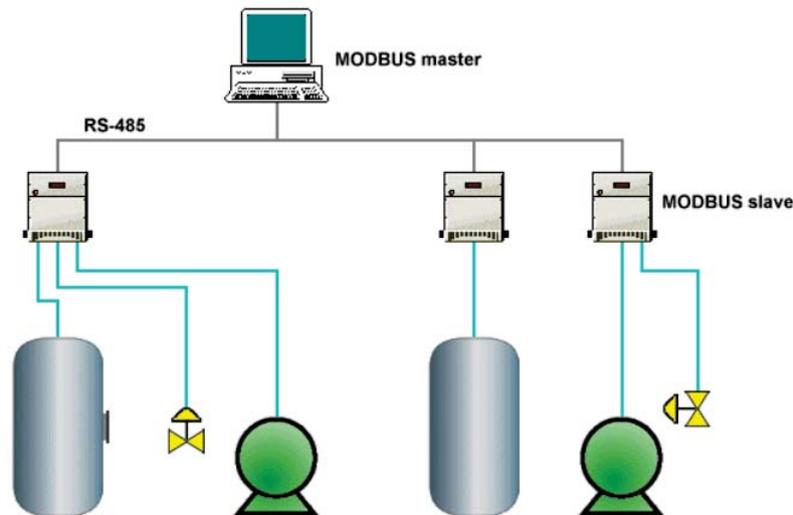


Figura 2.3: Conexión Maestro-esclavo en ModBus [Figura].

Recuperado de <https://technologyuk.net/telecommunications/industrial-networks/modbus.shtml>

2.1.4. ModBus

El estándar ModBus [17] define un protocolo de mensajería en la capa de aplicación, situado en el nivel 7 del modelo OSI que proporciona comunicaciones cliente / servidor.entre dispositivos conectados a diferentes tipos de buses o redes. También estandariza un protocolo específico en línea serie para intercambiar la petición ModBus entre un maestro y uno o varios esclavos, como se puede ver en la figura 2.3.

El protocolo ModBus permite una fácil comunicación en todos los tipos de arquitecturas de red. Además todos los tipos de dispositivos pueden utilizar este protocolo para iniciar una operación remota, y esta comunicación se puede realizar por el puerto serie o por Ethernet. Pero hay muchas variantes de protocolos ModBus entre las que destacan:

- ModBus ASCII: Esto se utiliza en la comunicación serial y hace uso de caracteres ASCII para la comunicación del protocolo. El formato ASCII utiliza un checksum de comprobación de redundancia longitudinal. Los mensajes Modbus ASCII están enmarcados por los dos puntos principales (":") y la nueva línea de salida (CR / LF).
- ModBus RTU: Es utilizado en la comunicación serial y hace uso de una representación binaria compacta de los datos para la comunicación del protocolo. El formato RTU sigue los comandos/datos con una suma de comprobación de redundancia cíclica que es un mecanismo de comprobación de errores para garantizar la fiabilidad de los datos. Modbus RTU es la implementación más común disponible para Modbus. Los mensajes son enmarcados (separados) por períodos inactivos.
- ModBus TCP/IP: Se trata de una variante Modbus utilizada para comunicaciones a través de redes TCP/IP, conectándose a través del puerto 502. No

requiere un cálculo de suma de comprobación, ya que las capas inferiores ya proporcionan protección de suma de comprobación.

El protocolo ModBus define una unidad de datos de protocolo (PDU) simple independiente de las capas de comunicación subyacentes. El mapeo del protocolo ModBus en buses o redes específicos puede introducir algunos campos adicionales en la unidad de datos de aplicación (ADU). La unidad de datos de aplicación ModBus es construida por el cliente que inicia una transacción ModBus. La función indica al servidor qué tipo de acción realizar. El protocolo de aplicación ModBus establece el formato de una solicitud iniciada por un cliente.

Los datos disponibles por medio de ModBus , son almacenados en uno de los cuatro tipos de bancos de datos que posee el protocolo [9] bobinas, entradas discretas, registros de retención y registros de entrada. Dependiendo del banco de datos utilizado, el tipo de datos a almacenar y los derechos de acceso a los mismos serán diferentes. Los dispositivos esclavos tienen acceso directo a estos datos, que son alojados localmente en los dispositivos. Los datos disponibles por medio de Modbus generalmente son un subconjunto de la memoria principal del dispositivo. Por el contrario, los maestros tienen que solicitar el acceso a estos datos a través de diversos códigos de función. El comportamiento de cada bloque de datos se define a continuación:

- Entradas discreta: Este tipo de datos puede ser proporcionado por sistemas de entrada y salida de datos. Este tipo de objetos son de solo lectura y su tamaño es de un bit.
- Bobinas: Este tipo de datos puede ser modificado por una aplicación. Este dato es de lectura y escritura y su tamaño es de un bit.
- Registros de entrada: Este tipo de datos puede ser proporcionado por sistemas de entrada y salida de datos. Solo puede ser accedido para su lectura y tiene un tamaño es una palabra de 16 bits.
- Registros de retención: Este tipo de datos puede ser modificado por una aplicación. Este dato es de lectura y escritura y su tamaño es una palabra de 16 bits.

Con estos bloques es posible restringir o permitir el acceso a los diferentes elementos de datos y también se puede proporcionar mecanismos simplificados en la capa de aplicación para el acceso a los datos. Estos bloques son completamente conceptuales, ya que pueden existir como direcciones de memoria separadas en un sistema determinado pero además también pueden trasladarse. El esquema de dirección se define completamente por el dispositivo esclavo y su interpretación de cada bloque de memoria es una parte importante del modelo de datos del dispositivo.

2.1.5. MBUS

M-Bus [16] fue desarrollado por el Profesor Dr. Horst Ziegler de la Universidad de Paderborn en cooperación con Texas Instruments Deutschland GmbH y

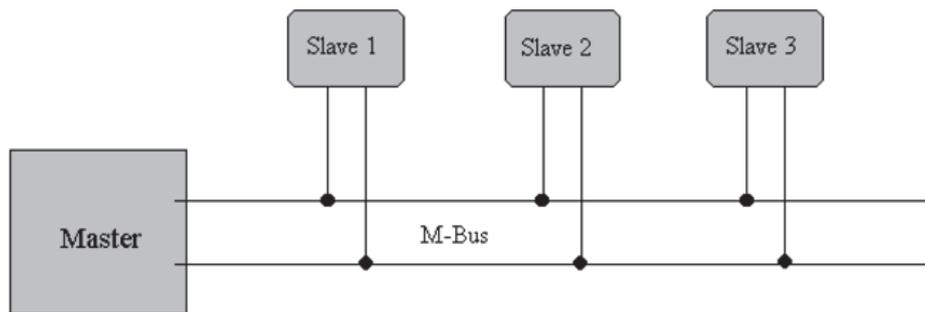


Figura 2.4: Diagrama de bloques que muestra el principio del sistema M-Bus [Figura]. Recuperado de <http://www.m-bus.com/mbusdoc/md4.php>.

Techem GmbH, es una norma europea para la lectura de forma remota de medidores de gas o electricidad, aunque también se utiliza para medidores de otro tipo de consumo. El M-Bus fue desarrollado para llenar la necesidad de un sistema para la conexión en red y lectura remota de medidores de utilidades, por ejemplo para medir el consumo de gas o agua en el hogar. Este bus cumple los requisitos especiales de los sistemas accionados remotamente o accionados por baterías, incluidos los medidores de utilidad para el consumidor. Cuando son interrogados, los medidores entregan los datos que han recogido a un maestro común, tal como un ordenador de mano, conectado en intervalos periódicos para leer todos los medidores de servicios públicos de un edificio. Un método alternativo de recolección centralizada de datos es transmitir lecturas de contador a través de un módem.

El concepto de M-Bus se basó en el modelo de referencia ISO-OSI, con el fin de realizar un sistema abierto que podría utilizar casi cualquier protocolo deseado. Dado que el M-Bus no es una red, y por lo tanto no necesita una capa de transporte o sesión, los niveles cuatro a seis del modelo OSI están vacíos. Por lo tanto, sólo el físico, el enlace de datos, la red y la capa de aplicación se proporcionan con funciones. Debido a que el cambio de parámetros como velocidad en baudios y dirección por capas superiores no está permitido en el Modelo ISO-OSI, se define una Capa de Gestión al lado y por encima de las siete Capas OSI. Por lo tanto, la dirección 254 y quizás 255 están reservadas para administrar la capa física y la dirección 253 (selección) para la capa de red, que sólo se utiliza en ciertos casos. Con esta nueva capa podemos gestionar directamente cada capa OSI para implementar características que no se ajustan al modelo OSI.

El M-Bus es un sistema jerárquico controlado por un maestro y con varios esclavos, como se puede ver en 2.4. La comunicación se lleva a cabo a través de un cable de dos hilos, con una transferencia de datos en serie. La transferencia de bits de maestro a esclavo se realiza mediante cambios de nivel de voltaje. Los bits enviados en la dirección contraria se codifican modulando el consumo de corriente del esclavo.

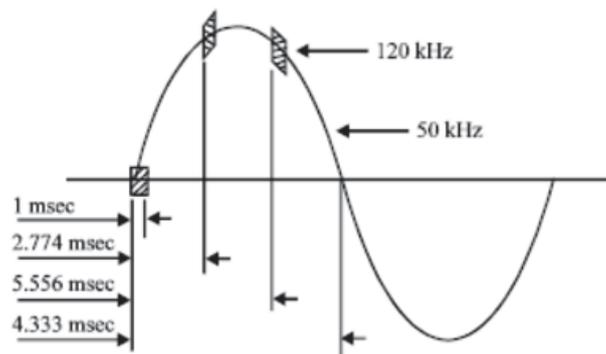


Figura 2.5: Transmisión de la señal en X10[Figura].

Recuperado de <http://scialert.net/fulltext/?doi=jas.2009.3562.3568>

2.1.6. X10

X10 [7] es un protocolo para el control remoto de dispositivos eléctricos desarrollada en 1978 por Pico Electronics. Este protocolo utiliza la red de corriente alterna convencional para la transmisión de la información, aunque podría haberse elegido otro, ya que el software es independiente de la red que se instale; sin embargo, las características especiales de este protocolo permiten reducir la instalación de los dispositivos del sistema domótico a la simple acción de enchufarlos a una toma de corriente eléctrica.

La transmisión de la información [8] está sincronizada con los pasos por cero de la corriente. Por lo tanto, un uno binario se corresponde con un pulso de 120 KHz durante un milisegundo y un cero binario se corresponde con la ausencia de ese pulso, como se puede ver en la imagen 2.5. La transmisión completa de un código X10 necesita 11 ciclos de corriente.

En el mercado hay una gran cantidad de dispositivos que implementan el protocolo X10, la instalación de casi todos ellos se basa en enchufarlos a una toma de corriente convencional, en caso de que el dispositivo sea inalámbrico es suficiente con fijarlo a una superficie. Otra parte importante es la conexión con el PC, ya que la mayoría de software comercial utilizado para los dispositivos X10 tiene características muy limitadas, permitiendo únicamente la programación y activación de algunas funciones a una hora prefijada.

Los dispositivos hardware que componen el sistema X10 se pueden dividir en tres categorías:

- Módulos de dispositivos: Existen una gran variedad de módulos que deben de variar en función de la carga que va a ser controlada. Para cargas de lámparas incandescentes, se deben de utilizar módulos de lámpara o interruptor de pared, que utilizan un interruptor TRIAC de estado sólido y también son capaces de atenuar la carga de la lámpara. Para cargas que no se correspondan con lámparas incandescentes se debe de utilizar un módulo de dispositivo, que se encargan de cambiar la electricidad mediante un impulso de relé. Muchos de los módulos de este tipo ofrecen la función 'control local', que sirve para que si el módulo está apagado, al accionar el interruptor de encendido de la lámpara o aparato, el módulo se enciende.

Hay otro tipo de módulos llamados de conmutador de pared que permiten regular el rango de brillo de la luz.

Los módulos de gama alta tienen características más avanzadas, como la programación de niveles, tipos de fundido personalizables o la capacidad de transmitir comandos mientras se utilizan

Hay módulos de sensores que detectan e informan de la temperatura, la luz, infrarrojos, movimiento, o aperturas y cierres de contacto. Los módulos de equipo incluyen termostatos, alarmas audibles y controladores para interruptores de baja tensión

- **Controladores:** Los controladores X10 pueden ser muy simples o muy complejos. Los transceptores son utilizados para recibir órdenes o ser programados por control remoto. Los controladores simples están preparados para controlar cuatro dispositivos X10 en cuatro direcciones secuenciales. Los controladores más sofisticados pueden controlar más unidades y/o incorporar temporizadores que realizan funciones preprogramadas en momentos específicos cada día. También están disponibles unidades que utilizan detectores de movimiento pasivos de infrarrojos o células fotoeléctricas para encender las luces sobre la base de las condiciones externas.
- **Puentes:** Este tipo de dispositivos se utiliza para traducir de X10 a otros estándares de domótica y viceversa.

2.1.7. BACnet

BACnet es un protocolo de comunicación para la red de comunicación y control de edificios, el desarrollo de este protocolo comenzó en Junio de 1987 en la reunión inaugural del comité del protocolo estándar (SPC), pero no fue hasta el 2006 cuando la Asociación de fabricantes de BACnet y el Grupo de interés de BACnet en Norteamérica se fusionaron en la llamada BACnet Internacional, que dio lugar a la especificación actual del protocolo [3].

La finalidad de esta norma es definir los servicios y protocolos de comunicación de datos para equipos informáticos utilizados para la automatización y control de servicios tales como sistemas de calefacción, ventilación, aire acondicionado, iluminación o acceso. Esta norma también permite realizar una representación abstracta y orientada a objetos de la información comunicada entre dichos equipos facilitando así la aplicación y el uso de la tecnología de control digital en edificios.

Este protocolo proporciona un conjunto completo de mensajes para transportar datos binarios, analógicos y alfanuméricos codificados entre dispositivos, incluyendo, pero sin limitarse a, valores de entrada y salida de hardware, valores de entrada y salida analógicos de hardware, valores binarios y analógicos de software, valores de cadena de texto, información de horarios, información de alarmas y eventos, archivos y lógica de control.

BACnet se basa en una arquitectura colapsada de cuatro capas que corresponde a las capas físicas, de enlace de datos, de red y de aplicación del modelo OSI, como se muestra en la figura 2.6. La capa de aplicación y una capa de red simple

BACnet Layers							Equivalent OSI Layers	
BACnet Application Layer							Application	
BACnet Network Layer							Network	
ISO 8802-2 (IEEE 802.3) Type 1		MS/TP	PTP	LonTalk	BVLL (Annex J)	BVLL (Annex U)	BZLL	Data Link
ISO 8802-3 (IEEE 802.3)	ARCNET	EIA-485	EIA-232		IPv4	IPv6	Zigbee	Physical

Figura 2.6: ANSI/ASHRAE Standard 135-2016(2006). Arquitectura colapsada de BACnet[Figura].

se definen en el estándar BACnet. BACnet proporciona las siguientes opciones que corresponden al enlace de datos OSI y a las capas físicas.

La capa física proporciona un medio para conectar los dispositivos y transmitir las señales electrónicas que transmiten los datos. Claramente la capa física es necesaria en un protocolo BAC.

La capa de enlace de datos organiza los datos en frames o paquetes, regula el acceso al medio, proporciona direccionamiento y gestiona la recuperación de errores y el control de flujo.

Las funciones proporcionadas por la capa de red incluyen traducción de direcciones globales a direcciones locales, encaminamiento de mensajes a través de una o más redes, acomodación de diferencias en tipos de red y en el tamaño máximo de mensaje permitido por dichas redes, secuenciación, control de flujo, control de errores y multiplexación. BACnet está diseñado para que sólo haya una ruta lógica entre dispositivos, eliminando así la necesidad de algoritmos de enrutamiento de ruta óptimos. Una red se compone de uno o más segmentos físicos conectados por repetidores o puentes pero con un solo espacio de dirección local.

La capa de aplicación del protocolo proporciona los servicios de comunicación requeridos por las aplicaciones para realizar sus funciones, en este caso monitoreo y control de los diferentes sistemas.

Por otra parte, este protocolo modela cada computadora de automatización y control de edificios como una colección de estructuras de datos llamadas "objetos", cuyas propiedades representan diversos aspectos del hardware, software y funcionamiento del dispositivo. Estos objetos proporcionan un medio para identificar y acceder a la información sin requerir el conocimiento de los detalles de el diseño o la configuración interna del dispositivo. En este documento se van a enumerar los cuatro objetos que se van a utilizar de BACnet, y que tiene varias propiedades en común, ya que todos tienen un nombre, un identificador, una descripción, un tipo y un valor. Los objetos son:

- Analog Input: El tipo de objeto de entrada analógica define un objeto estandarizado cuyas propiedades representan las características externamente visibles de una entrada analógica.



Figura 2.7: XBee Serie 1Figura.

Recuperado de <https://mcatronicauaslp.wordpress.com>

- **Analog Output:** El tipo de objeto Salida analógica define un objeto estandarizado cuyas propiedades representan las características externas visibles de una salida analógica.
- **Binary Input:** El tipo de objeto de entrada binaria define un objeto estandarizado cuyas propiedades representan las características externamente visibles de una entrada binaria. Una "entrada binaria" es un dispositivo físico o entrada de hardware que puede estar en sólo uno de dos estados distintos. En esta descripción, estos estados se denominan ACTIVO e INACTIVO. Un uso típico de una entrada binaria es indicar si una pieza particular de equipo mecánico, tal como un ventilador o una bomba, está funcionando o está inactiva. El estado ACTIVE corresponde a la situación cuando el equipo está encendido o en funcionamiento, y INACTIVO corresponde a la situación cuando el equipo está apagado o inactivo.
- **Binary Output:** El tipo de objeto de Salida binaria define un objeto estandarizado cuyas propiedades representan las características externamente visibles de una salida binaria. Una "salida binaria" es un dispositivo físico o salida de hardware que puede estar en sólo uno de dos estados distintos. En esta descripción, estos estados se denominan ACTIVO e INACTIVO. Un uso típico de una salida binaria es conmutar una pieza particular de equipo mecánico, tal como un ventilador o bomba, encendido o apagado. El estado ACTIVE corresponde a la situación cuando el equipo está encendido o en funcionamiento, y INACTIVE corresponde a la situación cuando el equipo está apagado o inactivo.

2.1.8. XBee

Los XBee son pequeñísimos chip azules, como los de la figura 2.7, capaces de comunicarse de forma inalámbrica unos con otros. Los primeros radios XBee [20] fueron introducidos bajo la marca MaxStream en 2005 y se basaron en la norma IEEE 802.15.4-2003 diseñada para comunicaciones punto a punto y estrella en velocidades de transmisión en baudios de 250 kbit/s. Dos modelos fueron inicialmente introducidos uno de costo menor de 1 mW XBee y uno con mayor potencia de 100 mW XBee-PRO. Desde la introducción inicial, se han introducido una serie de nuevos radios XBee y se ha desarrollado un ecosistema de módulos inalámbricos, pasarelas, adaptadores y software.

Los radios XBee se pueden utilizar con el número mínimo de conexiones, potencia (3.3 V), tierra, entrada de datos y salida de datos (UART), siendo las otras líneas recomendadas Reset y Sleep. Además, la mayoría de las familias de XBee tienen algún otro control de flujo, entradas/salidas, convertidor analógico a digital y líneas de indicador incorporadas. Existen dos tipos básicos de variedades de hardware, los Xbee serie 1 que no necesitan ser configurados y soportan la comunicación punto a punto y punto-multipunto; por otro lado el Xbee serie 2 (2.5, ZB, 2B) que soportan la comunicación punto a punto, punto-multipunto y redes mesh, pero por contra necesitan ser configurados. Destacar, que la comunicación entre estos dos tipos de dispositivos nos es posible. Tanto la Serie 1 y Serie 2 están disponibles en dos potencias de transmisión diferentes, la normal (simplemente Xbee) y la PRO (Xbee-PRO). La versión PRO consume más energía, son más caros y tienen más rango de distancia (de 1 a 15 millas).

2.2 Conversión de Modbus a BACnet

En este trabajo nos estamos centrando en los protocolos Modbus y BACnet, que como ya se ha visto Modbus es un protocolo muy utilizado en equipamiento mecánico, pero no es compatible con BACnet, por lo que en estos últimos tiempos se han estado desarrollando diferentes herramientas y sistemas que permitan realizar de una forma sencilla el "mapping" entre objetos de un protocolo y otro, sin perder información en el proceso.

Para realizar la conversión descrita hay que tener en cuenta que los dispositivos Modbus pueden comunicarse a través de ASCII, que es un viejo protocolo raramente utilizado en la actualidad, RTU, que es una interfaz de línea serial muy popular, o TCP, que opera a través de Ethernet. Por otro lado BACnet utiliza principalmente la IP para que sus dispositivos se comuniquen entre si.

En este apartado se va a hacer un estado del arte con todas las soluciones propuestas para integrar un dispositivo Modbus en una red BACnet.

2.2.1. Pasarelas comerciales

En el mercado existen multitud de pasarelas que se encargan de pasar los datos del protocolo Modbus a BACnet, únicamente conectando un dispositivo externo entre el dispositivo Modbus y el resto del sistema. Muchos de estos dispositivos desarrollados solo son compatibles con un determinado protocolo de comunicación y es este el criterio elegido para agrupar las distintas alternativas existentes en el mercado.

La mayoría de las pasarelas comerciales existentes en la actualidad solo son compatibles con BACnet/IP, aunque por el contrario muchas de ellas soportan varios protocolos de comunicación de Modbus. En [5] se propone un dispositivo que realiza la conversión de un esclavo Modbus RTU y un sistema de automatización de edificios con el protocolo BACnet/IP, para utilizar este dispositivo en una aplicación es necesario realizar un mapeo en el que los datos de registro Modbus se asignan a objetos analógicos BACnet y los datos de la bobina Modbus se asignan a objetos binarios en BACnet. La única pieza adicional requerida pa-

ra BACnet es la adición de metadatos que no es más que una descripción ASCII de cada objeto. Siguiendo con el mismo tipo de conversión, en [10] se propone otra pasarela que puede leer datos de Modbus desde sus dispositivos Modbus RTU y publicarlos como datos IP de BACnet. También puede escribir en los dispositivos Modbus RTU. Así como hacer lo contrario, es decir, leer los datos IP de BACnet y publicarlos como datos RTU de Modbus y la última funcionalidad que proporciona es que puede escribir en los dispositivos IP de BACnet, por lo tanto a diferencia del dispositivo explicado anteriormente este es capaz de realizar la traducción en ambas direcciones. La empresa FieldServer tiene el siguiente dispositivo [11] capaz de realizar la conversión entre varios protocolos, entre los que se encuentran Modbus RTU y BACnet, ya sea a través de IP o MS/TP, éste último no incluido en ninguna de las pasarelas anteriores. Otra de las diferencias de esta pasarela con respecto a las anteriores es que ésta proporciona una aplicación de escritorio para configurar la conversión.

Otra pasarela similar es BASgatewayLX descrita en [4], el principal objetivo es hacer que los dispositivos Modbus aparezcan como dispositivos de BACnet individuales. Este dispositivo tiene una conexión para el Modbus TCP o BACnet/IP además también cuenta con un puerto en serie que sirve para que sea compatible con dispositivos Modbus RTU y Modbus ASCII. Este dispositivo permite que más de 30 dispositivos Modbus puedan compartir el mismo puerto Modbus del BASgatewayLX, esto no es un problema, ya que virtualmente cada dispositivo Modbus conectado aparecerá como un dispositivo individual compatible con BACnet, lo único necesario es un perfil de dispositivo para cada dispositivo Modbus. Esto se consigue en la web de Contemporary Controls que posee una biblioteca de perfiles de dispositivos comunes, si uno en concreto no está disponible la web lo proporcionará bajo petición. Además todos los dispositivos personalizados se podrán cargar en el BASgatewayLX. Por lo tanto, utilizando páginas web y una base de datos residente de perfiles de dispositivos Modbus comunes, los puntos de datos Modbus de los dispositivos Modbus Serial o Modbus TCP pueden asignarse a objetos BACnet.

Al igual que en el caso anterior, existen pasarelas comerciales para los dispositivos Modbus TCP, entre ellas se encuentra la propuesta en [12], que crea una poderosa conexión entre dispositivos en una red Modbus TCP/IP y dispositivos BACnet. La versión de protocolo ProLinx BACnet/IP (Building Automation and Control) está diseñada específicamente para satisfacer las necesidades de comunicación de sistemas de automatización y control de edificios para aplicaciones tales como control de calefacción, ventilación y aire acondicionado, control de iluminación, control de acceso e incendio, sistemas de detección con soporte limitado de comandos BACnet/IP. Además proporciona una aplicación que gestiona los archivos de configuración con un diseño gráfico estructurado en árbol, con una amplia ayuda en línea para definir los parámetros de configuración y la configuración de la base de datos.

El convertidor de protocolo [14], que se puede ver en la imagen 2.8 permite la integración fácil de una amplia selección de medidores de la marca Veris con sistemas de automatización de edificios a través del protocolo BACnet. Cuando está conectado a la red, el E8951 detecta los medidores Modbus soportados y les proporciona un ID de dispositivo BACnet único y un conjunto completo de datos de medición y objetos de configuración. Simplemente hay que seleccionar



Figura 2.8: Convertidor de protocolo Modbus a BACnet E8951[Figura].

los ajustes de protocolo deseados mediante los interruptores DIP y el servidor web integral, y los medidores Veris Modbus soportados estarán disponibles como dispositivos BACnet totalmente compatibles.

Otra pasarela existente es Onyx Edge MB311 [18], que está diseñado para integrar los sistemas de comunicación Modbus TCP/RTU a los sistemas o dispositivos BACnet IP que permitan monitorear y controlar dispositivos y sistemas integrados. Esta pasarela tiene incorporado un framework llamado Helixx, que lleva la traducción del protocolo Modbus a BACnet al siguiente nivel, no sólo traduciendo hasta 2500 puntos Modbus a puntos BACnet manejables; sino que convierte los dispositivos esclavos Modbus en dispositivos virtuales BACnet mediante enrutamiento virtual BACnet. Actuando como el dispositivo maestro Modbus, Helixx gestiona todos los dispositivos esclavos Modbus RTU o TCP conectados a él. Para cada dispositivo esclavo Modbus que se añade, Helixx crea un dispositivo BACnet virtual correspondiente. Helixx garantiza que el dispositivo BACnet virtual emule siempre el dispositivo Modbus que está puenteando y protege la integridad de la red BACnet asignando automáticamente una instancia de dispositivo BACnet única/ ID de objeto a cada dispositivo BACnet virtual utilizado para identificarlo en la red BACnet. Los puntos Modbus deseados se añaden simplemente bajo el dispositivo esclavo Modbus. Helixx crea y mapea automáticamente el objeto BACnet virtual correcto en su dispositivo BACnet virtual correspondiente. Por lo tanto con este framework el dispositivo Onyx no necesita software de terceros para acceder y configurar la pasarela.

También se han desarrollado pasarelas compatibles con todos los tipos de redes de comunicación soportados por Modbus, la puerta de entrada [1] permite conectar dispositivos Modbus a una red BACnet. El dispositivo físico correspondiente se puede ver en la figura 2.9. La puerta de enlace funciona como un intérprete entre las dos redes. Los dispositivos que soportan Modbus (RTU, ASCII o TCP), aparecen en la red BACnet/IP como dispositivos individuales BACnet compatibles. Esto le permite controlar de forma centralizada los dispositivos Modbus, por ejemplo en un edificio y el monitor. La puerta de enlace funciona de la siguiente forma, las redes Modbus RTU y Modbus ASCII están conectados al puerto de serie de la puerta de enlace. Modbus TCP y BACnet/IP redes se conectan al puerto Ethernet, un perfil de dispositivo debe ser creado y descargado en la



Figura 2.9: Interfaz BACnet para objetos Modbus - Módulo[Figura]. Recuperado de <https://www.cimetrics.com/products/b6131>

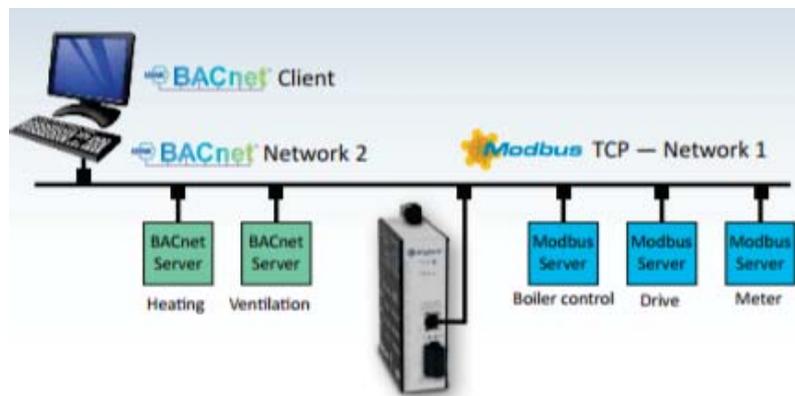


Figura 2.10: Conexión de MODbus/TCP a BACnet/IP[Figura]. Recuperado de www.anybus.com

puerta de entrada para cada dispositivo Modbus. En la imagen 2.10 se muestra la red resultante al realizar la conversión de ModBus/TCP a BACnet/IP utilizando la pasarela de la marca Anybus.

Ya que la mayoría de los dispositivos ModBus deben de conectarse a una red BACnet, la compañía Cimetrics [2] ofrece varios convertidores de protocolos, el último es el B6131. El B6131 actúa como interfaz Modbus a BACnet, tomando los datos del registro Modbus y exponiéndolo a BACnet. También incluye plantillas de ejemplo de la gama de metros conocidos por lo que a menudo no es necesario pasar tiempo eligiendo qué registros para exponer desde cada dispositivo. Además, el B6131 permite a los usuarios crear sus propias plantillas cargables para asignar un dispositivo Modbus RTU o Modbus TCP a BACnet/IP. El B6131 incluye ejemplos de algunas plantillas de dispositivos Modbus para proporcionar las mejores prácticas de usuario al elegir nombres, unidades y descripciones. Todos los objetos de BACnet tienen un nombre significativo y están estandarizados para facilitar la lectura e integración en los sistemas de automatización de edificios. Los pasos de configuración incluyen el uso de una herramienta Windows para crear una plantilla (la herramienta tiene menús desplegables para facilitar la creación de nombres y la elección de unidades), especificando la dirección IP

de un medidor (si es un medidor TCP/IP), especificando el ID Modbus del dispositivo y Cargando una plantilla creada/suministrada con la herramienta. La configuración se puede guardar en un archivo .csv para el mantenimiento de registros y análisis de datos.

Una de las pasarelas ms importantes es la llamada BASremote [6], de la compañía ContemporaryControls. Esta pasarela es capaz de convertir dispositivos ModBus ASCII, RTU y TCP a un dispositivo que siga el protocolo BACnet/IP. Esta pasarela tiene asociada un aplicación en la que se puede desarrollar un perfil de dispositivo ModBus. Una vez que el perfil del dispositivo se desarrolla se instala en el BASremote mediante la creación de una lista de escaneo. Esos puntos ModBus en la lista de exploración serán accesibles desde un cliente BACnet. Los perfiles de dispositivos Modbus son desarrollados por Contemporary Controls como una conveniencia para sus clientes. Además el cliente puede modificar el perfil del dispositivo para ajustarse a los requisitos únicos de una instalación o desarrollar un perfil de dispositivo completamente nuevo.

2.2.2. Software

Otra forma de realizar la conversión entre ambos protocolos es a través de un programa software, que se pueda ejecutar en un ordenador, Raspberry o en un router de BACnet. En [13] se presenta una aplicación en la que se puede configurar los dispositivos Modbus que se van a conectar, así como su tipo de red y el puerto de conexión correspondiente. La dirección Modbus, tipo de datos, registro de tipo y orden de bytes se debe seleccionar para cada punto de Modbus que necesita ser añadido. Para asignar los puntos Modbus a puntos BACnet se crea una extensión BACnet para cada uno de los componentes Modbus. Una vez terminado el proceso de configuración y mapeo, los componentes Modbus aparecerán en la aplicación para comprobar que todo ha ido correctamente. Los puntos BACnet resultantes pueden ser vistos desde cualquier cliente BACnet conectado a la red.

2.3 Formatos de mapping

Uno de los aspectos muy importantes que se debe de tener en cuenta es el formato en el que se guardan los datos correspondientes al mapeo entre dos protocolos. En este caso se va a investigar solo acerca de los mappings entre los protocolos ModBus y BACnet, pero estos formatos se pueden utilizar en traducciones entre cualquier par de protocolos.

El formato más utilizado en el mercado para realizar este tipo de mapping, son las tablas, por lo que la información se guarda en un fichero con formato csv, lo que se corresponde con que cada una de las filas de la tabla se corresponde con una fila del fichero, en la que el contenido de cada columna está separado por una coma, como se puede ver en la figura 2.11. Este tipo de mapeo se ha utilizado en la mayoría de dispositivos entre los que destaca por importancia [6],

El otro formato que se suele utilizar, aunque en menor medida, para guardar el resultado de los mappings es el txt, el esquema de dicho fichero suele ser como el que se muestra en la figura 2.12, en el que cada propiedad aparece en una línea

```

YES,0, Flow in Eng unit ,11,X400001,32 Bit Unsigned Integer LO/HI,ANALOG_INPUT,1001,NO_UNITS,Not Defined ,None,0,1,0,1,1,0
YES,0, Total ,11,X400003,32 Bit Unsigned Integer LO/HI,ANALOG_INPUT,1002,NO_UNITS,Not Defined ,None,0,1,0,1,1,0
YES,0, Temperature ,11,X400005,32 Bit Unsigned Integer LO/HI,ANALOG_INPUT,1003,NO_UNITS,Not Defined ,None,0,1,0,0,1,1,0
YES,0, Elapsed time ,11,X400007,32 Bit Unsigned Integer LO/HI,ANALOG_INPUT,1004,NO_UNITS,Not Defined ,None,0,1,0,0,1,1,0
YES,0, Velocity ,11,X400009,32 Bit Unsigned Integer LO/HI,ANALOG_INPUT,1005,NO_UNITS,Not Defined ,None,0,1,0,1,1,0
YES,0, Flow in Eng unit * 10 ,11,X400011,16 Bit Unsigned Integer,ANALOG_INPUT,1006,NO_UNITS,Not Defined ,None,0,1,0,1,1,0
YES,0, Flow in Eng unit *100 ,11,X400012,16 Bit Unsigned Integer,ANALOG_INPUT,1007,NO_UNITS,Not Defined ,None,0,1,0,1,1,0
YES,0, Total *100 ,11,X400013,16 Bit Unsigned Integer,ANALOG_INPUT,1008,NO_UNITS,Not Defined ,None,0,1,0,1,1,0
YES,0, Total2 ,11,X400014,32 Bit Unsigned Integer LO/HI,ANALOG_INPUT,1009,NO_UNITS,Not Defined ,None,0,1,0,1,1,0
YES,0, Status Power up indication ,11,X400016,Bit16,BINARY_INPUT,1010,NO_UNITS,Not Defined ,0,0,1,0,1,1,0
YES,0, Status Flow rate reached high limit threshold ,11,X400016,Bit16,BINARY_INPUT,1011,NO_UNITS,Not Defined ,1,0,1,0,1,1,0
YES,0, Status Flow rate reached high limit threshold ,11,X400016,Bit16,BINARY_INPUT,1012,NO_UNITS,Not Defined ,2,0,1,0,1,1,0
YES,0, Status Temperature reached high limit threshold ,11,X400016,Bit16,BINARY_INPUT,1013,NO_UNITS,Not Defined ,3,0,1,0,1,1,0
YES,0, Status Temperature reached low limit threshold ,11,X400016,Bit16,BINARY_INPUT,1014,NO_UNITS,Not Defined ,4,0,1,0,1,1,0
YES,0, Status Sensor reading is out of range ,11,X400016,Bit16,BINARY_INPUT,1015,NO_UNITS,Not Defined ,5,0,1,0,1,1,0
YES,0, Status Velocity flow rate outside of calibration table ,11,X400016,Bit16,BINARY_INPUT,1016,NO_UNITS,Not Defined ,6,0,1,0,1,1,0

```

Figura 2.11: Ejemplo de Fichero de mapeo en formato CSV.
Recuperado de BASremote

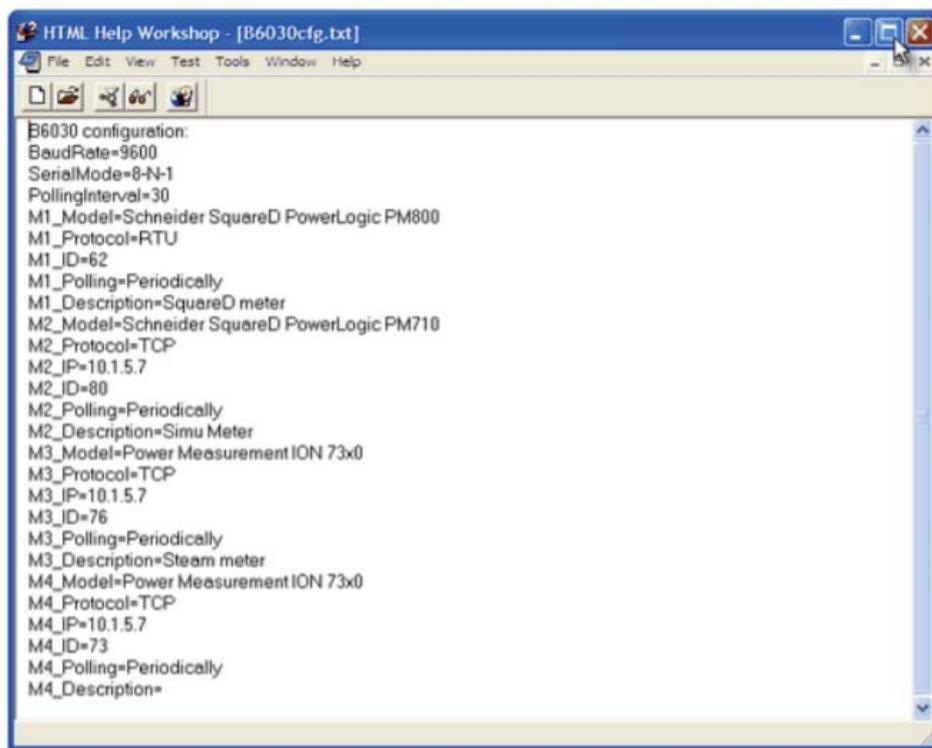


Figura 2.12: Ejemplo de Fichero de mapeo en formato TXT.
Recuperado de Cimetrics

del fichero, en el formato "título:valor". Este tipo de formato se ha utilizado en aplicación [2]

CAPÍTULO 3

Análisis y Desarrollo

3.1 Definición del sistema

Este sistema va a ser una aplicación que proporcione un fichero con el que realizar la traducción de un objeto ModBus a un objeto BACnet, teniendo en cuenta los registros de cada uno de ellos. En esta versión se podrán mapear los registros del objeto ModBus a los registros del objetos BACnet, por lo tanto los datos se pasarán de un registro al otro, el sistema va a permitir al usuario seleccionar varios parámetros en función del tipo de registro y de dato que se esté tratando. Por otra parte, el sistema mostrará en todo momento los registros que se han añadido hasta el momento.

Será necesario un navegador web en el que se pueda desplegar la aplicación. Por otra parte, la seguridad no va a ser uno de los objetivos, por ello no se van a implementar medidas de seguridad adicionales a las que incluyan las librerías utilizadas.

La aplicación consta de tres interfaces de usuario, una a modo de pantalla principal de la aplicación desde la que el usuario puede dar un nombre a la conversión, que también dará nombre al fichero de salida.

En la segunda interfaz, el usuario debe configurar el mapeo de un registro ModBus a un registro BACnet, rellenando todos los parámetros que se piden, como son el tipo, nombre, dirección del esclavo, descripción...

Por último en la interfaz final el usuario podrá ver en un tabla los registros añadidos así como los datos asociados a cada uno de ellos, además dispondrá de tres opciones extras como son la de eliminar todos los registros y comenzar de nuevo en la pantalla principal, añadir un nuevo registro a través de la interfaz descrita anteriormente o la de generar un fichero en formato CSV con todos los datos que ha introducido anteriormente y que puede ver en la tabla.

3.2 Herramientas, Tecnologías y Plataformas

En este apartado se van a enumerar las herramientas, las tecnologías y las plataformas utilizadas para la realización del proyecto.

- **Bootstrap:** Framework que permite crear interfaces web con CSS y JavaScript, utilizada para realizar el desarrollo front-end de la aplicación y con el fin de que tenga un diseño más bonito y usable.
- **Flask:** Es un microframework cuyo lenguaje de programación es python que permite crear aplicaciones web de una forma rápida y con un bajo número de líneas de código. Este framework también incluye un servidor web e desarrollo con el que poder probar las aplicaciones desarrolladas sin necesidad de instalar un servidor externo.
- **Atom:** Editor de texto desarrollado por GitHub, compatible con varios lenguajes de programación. Esta herramienta se ha utilizado para desarrollar y gestionar el código.
- **GitHub:** Plataforma de desarrollo colaborativo utilizada como método de control de versiones a través de la herramienta de escritorio que proporciona.
- **Navegador Web:** Se han utilizado distintos navegadores web como Safari o Google Chrome para comprobar que la aplicación se comporta igual en todos los navegadores
- **PythonAnywhere:** Es un entorno de desarrollo integrado en línea y un servicio de alojamiento basado en el lenguaje de programación Python. Proporciona acceso desde el navegador a las interfaces de línea de comandos Python y Bash del servidor, junto con un editor de código con resaltado de sintaxis. Los archivos de la aplicación se pueden transferir al servidor utilizando el navegador del usuario. Las aplicaciones Web alojadas por el servicio se pueden escribir utilizando cualquier marco de aplicación basado en WSGI. Se ha utilizado esta herramienta para desplegar la aplicación realizada y que así esté disponible en Internet.

3.3 Interfaz de usuario

3.3.1. Página de inicio

En la imagen 3.1 se puede ver la interfaz principal de la aplicación que aparece al abrirla, en la parte superior hay un hueco en el que el usuario debe de introducir el nombre que le quiere dar a la traducción que se va a realizar, este nombre es el que dará nombre al fichero de salida. En la parte de abajo aparece una tabla con los registros introducidos por el usuario, como se acaba de iniciar la aplicación la tabla se encuentra vacía.

En la parte inferior aparece un botón para pasar a la siguiente interfaz, una vez introducido el nombre deseado.

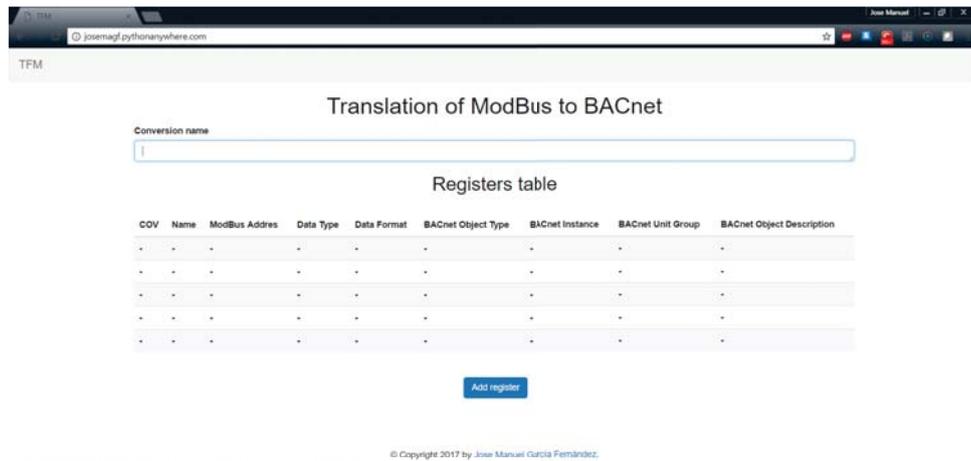


Figura 3.1: Interfaz de inicio de la aplicación

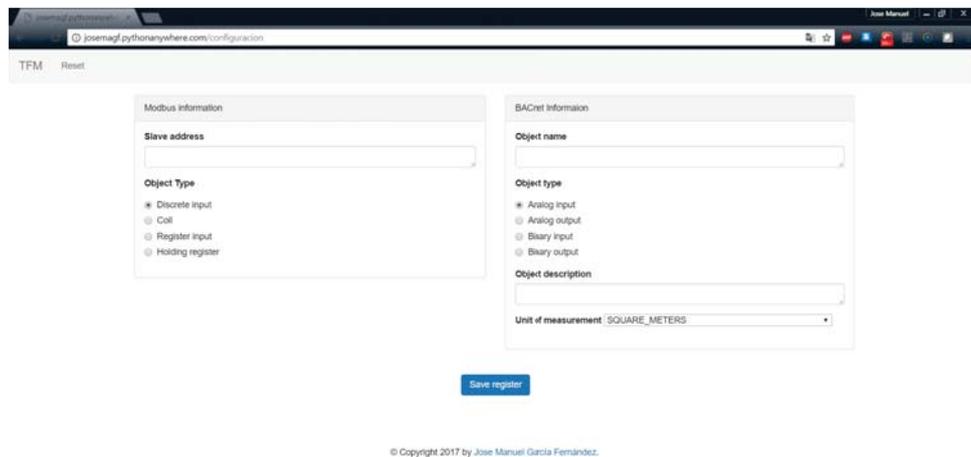


Figura 3.2: Interfaz de configuración de un registro

3.3.2. Configuración del registro

La interfaz de configuración de registro que se muestra en 3.2 está compuesta por una barra de navegación en la parte superior, con una única funcionalidad, que es la de volver a la página de inicio y borrar todo lo realizado hasta el momento.

En la parte central aparece un formulario, que está dividido en dos partes diferenciadas, por un lado los datos referentes al protocolo ModBus y por otro lado los referentes al protocolo BACnet. Estos formularios constan de cuadros de texto, radiobotones y comboboxes.

Por último en la parte inferior aparece un botón, para poder pasar a la siguiente interfaz y guardar los avances.

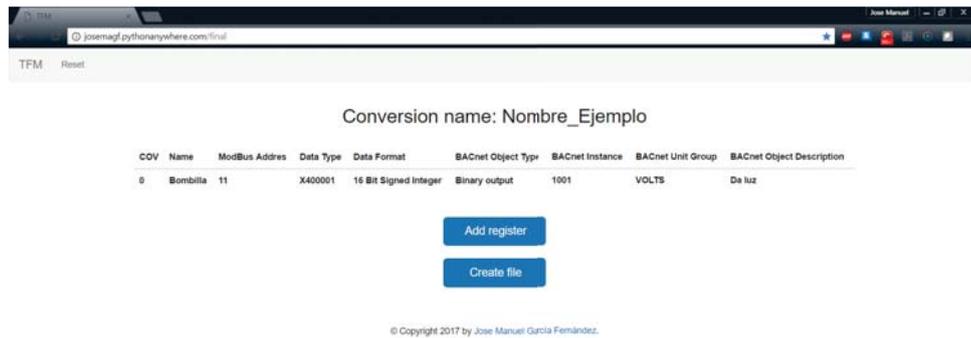


Figura 3.3: Interfaz con el resumen de los datos introducidos

```

1 COV,Name,ModBus Address,Data Type,Data Format,BACnet Object Type,BACnet Instance,BACnet Unit Group,BACnet Object Description,
2 0,Ejemplo1,11,X400001,Bit,Binary output,1001,MEGAVOLT AMPERES,-,
3 0,Example2,11,X400002,16 Bit Signed Integer,Analog output,1002,SQUARE METERS,Description ,
4 0,Example3,11,X400003,Bit,Analog input,1003,PERCENT PER SECOND,-,
5

```

Figura 3.4: Fichero resultante de la aplicación

3.3.3. Resumen de registros

La interfaz final de la aplicación se puede ver en 3.3, tiene una barra de navegación en la parte superior, con una única funcionalidad, que es la de volver a la página de inicio y borrar todo lo realizado hasta el momento.

En la parte central aparece el nombre con el que se creará el fichero resultante, y justo debajo una tabla con todos los registros detallados añadidos por el usuario, que será el contenido del fichero.

En la parte inferior hay una botonera en la que el usuario podrá decidir si añadir un nuevo registro o finalizar el proceso.

3.4 Implementación del sistema

3.4.1. Diseño del fichero de salida

El diseño y selección de los datos que debe de contener el fichero de salida es la primera cuestión a tener en cuenta a la hora de comenzar el proyecto, ya que debe contener todos los datos necesarios para poder realizar la traducción de un objeto ModBus a un objeto BACnet, pero sin excesos de información. Primeramente se ha elegido el formato CSV, ya que al ser una tabla es mucho más sencillo tanto de entender como de utilizar posteriormente, y como se ha visto anteriormente es el tipo de fichero más utilizado en este tipo de tareas. Por lo tanto el fichero de salida tendrá un aspecto similar al que se muestra en la imagen 3.4.

Una vez decidido el formato hay que seleccionar los datos que van a aparecer que son los siguientes:

- Columna 1: Contiene el valor COV(cambio de valor). Para una variable binaria su valor va a ser 0, porque un cambio de estado siempre se transmite si el host ha establecido una conexión COV. Para una variable analógica el valor puede ser 0 o algún valor como 10, 20.56, etc que represente el incremento COV.
- Columna 2: Contiene el nombre del objeto BACnet. Debe de ser único y debe de tener una extensión de 1 a 64 caracteres, entre los que no puede estar la coma.
- Columna 3: Identifica la dirección del esclavo Modbus, que debe de estar en un rango entre 10 y 247.
- Columna 4: Contiene el tipo de datos y los elementos de la tabla de datos. La dirección está precedida por una X mayúscula seguida por el tipo de datos Modbus (0,1,3,4), posteriormente el número de elemento por la tabla de datos 1-65535. El ajuste de la tabla de datos comienza con 1 (denominado direccionamiento del PLC). Sin embargo, los datos se almacenarán realmente en ubicaciones que son menos que la dirección del PLC
- Columna 5: Contiene el formato de los datos que puede ser:
 - Bit: Es para las referencias 0aaaaa y 1aaaaa y para la selección de bits de las referencias 3aaaaa o 4aaaaa. Como se mencionó anteriormente, un esquema de direccionamiento de 6 dígitos se utiliza con .aaaaa" que representa una dirección de 1 a 65535.
 - 16 Bit Signed Integer: 16 bits leídos como -32.768 a 36.767.
- Columna 6: Contiene el tipo de objeto BACnet al que se se quiere hacer la traducción. Puede ser uno de los siguientes:
 - Binary Input.
 - Binary Output.
 - Analog Input.
 - Analog Output.
- Columna 7: Contiene el número de instancia del objeto BACnet. Este número secuencial es generado por el programa de hoja de cálculo que comienza con 1000. Cada instancia de objeto es única independientemente del tipo de objeto. Sólo asigna números en el rango de 1000-8000. El resto de números están reservados.
- Columna 8: Contiene la unidad de medida válida de BACnet, hay una gran cantidad de medidas soportadas, divididas en las siguientes categorías:
 - Superficie.
 - Eléctrica.

- Energía.
 - Entalpía.
 - Frecuencia.
 - Humedad.
 - Longitud.
 - Luz.
 - Masa.
 - Potencia.
 - Presión.
 - Temperatura.
 - Tiempo.
 - Velocidad.
 - Volumen.
 - Otros
- Columna 9: Contiene la descripción del objeto BACnet. Este es un campo opcional en el objeto BACnet y no es necesario, pero no se puede dejar el campo vacío, debe contener al menos un carácter.

3.4.2. Elección de los datos a introducir por el usuario

Una vez que ya se tienen los datos que deben aparecer en el fichero de salida de la aplicación hay que buscar la forma de obtenerlos. El principal objetivo es que el usuario deba introducir el menor número de datos posibles, ya que se busca que la aplicación sea fácil de utilizar y rápida. Para llevar a cabo esto se ha decidido separar los datos a pedir en dos bloques, por un lado los relativos al objeto ModBus y por otro los del objeto BACnet. Los datos que se van a pedir con respecto al protocolo ModBus son los siguientes:

- Dirección del esclavo: El usuario tendrá un campo de texto para introducirla con el formato explicado en el apartado anterior.
- Tipo de Objeto: El usuario podrá elegir el tipo de objeto entre los cuatro que va a soportar la aplicación, se hará de esta forma para que no se pueda introducir un valor erróneo.

Los datos que se van a pedir al usuario relativos al objeto BACnet son los siguientes:

- Nombre del objeto: El usuario tendrá un campo de texto para introducir el nombre del objeto.
- Tipo de objeto: El usuario podrá elegir uno de los cuatro tipos de objetos BACnet soportados por la aplicación,, esto se hace para que no se pueda introducir un valor erróneo.

- Descripción del objeto: El usuario tendrá un campo de texto para introducir la descripción del objeto.
- Unidad de medida: El usuario podrá elegir a través de un combobox una de las medidas soportadas, destacar que todas las medidas soportadas aparecen clasificadas por las categorías expuestas en el apartado anterior. Se emplea esta técnica para elegir la unidad de medida para que no se introduzcan valores erróneos que puedan hacer fallar la conversión en pasos futuros.

Con estos datos pedidos al usuario se van a obtener todos los campos del fichero de salida, ya que sabiendo el tipo de objeto que es de uno u otro protocolo se pueden deducir casi todos los demás campos. Esto se debe a que en esta aplicación solo están soportados 4 tipos de objetos BACnet, de los más de 20 existentes, por lo que saber el tipo de datos que maneja cada uno es relativamente sencillo. Por otra parte hay un identificador único que es generado automáticamente por la aplicación de forma que ningún registro tenga el mismo identificador.

3.4.3. Implementación de la aplicación

Ahora que ya se tienen los parámetros que debe introducir el usuario y los campos que deben aparecer en el fichero de salida de la aplicación, es momento de comenzar con la programación.

Lo primero a tener en cuenta en este apartado es la elección del lenguaje de programación en el que se va a crear la aplicación. En este caso se ha decidido utilizar Python, ya que el protocolo BACnet soporta este lenguaje, por lo que en un futuro se podría integrar esta aplicación con el protocolo y es más sencillo si está construida en el mismo lenguaje de programación. Otro de los factores clave a la hora de la elección del lenguaje es la posibilidad de desplegar la web de forma gratuita, algo que se comentará en el siguiente apartado. Por lo tanto se ha decidido utilizar el microframework Flask para la creación de la aplicación, ya que además tiene soporte para Bootstrap, que se ha utilizado para mejorar el aspecto visual.

3.4.4. Despliegue de la aplicación Web

Por último, para que la aplicación esté disponible hay que subirla a un servidor. En este caso se ha decidido utilizar la plataforma PythonAnywhere que ofrece servicios de hosting gratuitos para distintos tipos de aplicaciones.

En este caso, la plataforma elegida tienen compatibilidad con aplicaciones desarrolladas en flask, por lo tanto no hay que realizar ninguna modificación al código inicial para poder tener la aplicación en la red. Por lo tanto el proceso a llevar a cabo es crear un proyecto en PythonAnywhere y subir los ficheros a través del asistente que proporciona el servicio utilizado. Por tanto, la aplicación está actualmente disponible en a través de un navegador web.

CAPÍTULO 4

Discusión de los resultados y líneas futuras

En este apartado se hará una discusión de los resultados obtenidos durante la investigación y el sistema creado.

Se ha implementado un sistema básico con las tecnologías actuales, para realizar el mapeo de objetos del protocolo ModBus a objetos del protocolo BACnet. Para intentar mejorar lo ya existente se ha realizado una profunda investigación del mercado y del estado actual en el ámbito de la automatización. Como se ha visto hay una gran cantidad de protocolos que pueden ser utilizados para este tipo de tareas, pero se ha decidido la utilización de BACnet ya que es un protocolo muy utilizado en la actualidad y que tiene una implementación compatible con varios lenguajes de programación, por otra parte se ha tenido que utilizar el protocolo ModBus en los contadores de luz, ya que es el protocolo que traen instalado por defecto. Una vez con el problema planteado, el principal problema a resolver es como realizar la traducción y como interpretar esa traducción, por lo que se ha investigado acerca de todas las soluciones que ofrece el mercado. La mayoría de empresas proponen utilizar una pasarela comercial, la mayoría de las veces con un precio elevado, rondando los 150 euros, y que en la mayoría de los casos no proporcionan una forma sencilla de configurar el dispositivo, ya que únicamente ofrecen un fichero a rellenar o incluso a veces ya vienen con los mapeos predefinidos y que no se pueden modificar. En el sistema que se ha realizado se ha intentado mejorar esto, y que el usuario de una forma rápida y sencilla pueda configurar la traducción como quiera. Esto se consigue a través de una conceptualización de los objetos, ya que por ejemplo el usuario no debe de introducir todos y cada uno de los datos del fichero de salida, si no que a partir de unos pocos datos se realiza un proceso de inferencia y se obtiene el resto. Un aspecto muy importante que se ha mejorado con la aplicación realizada es la compatibilidad con todos los tipos del protocolo ModBus y BACnet existentes, como puede ser RTU, ASCII, IP... Esto se ha conseguido ya que la aplicación puede ser ejecutada en cualquier dispositivo, ya sea un ordenador, una Raspberry, un Arduino y todos estos dispositivos tienen la capacidad de poder conectarse a los objetos del sistema a través de distintos tipos de puertos, por lo tanto es una de las principales mejoras del sistema realizado respecto a la mayoría de los dispositivos existentes en el mercado.

Otro aspecto a tener en cuenta es el formato de salida del mapping, en la aplicación desarrollada se ha decidido utilizar un fichero CSV como en la mayoría de sistemas de este tipo, pero este formato puede ser mejorado, una de las alternativas es la utilización de ontologías, con esto se conseguiría realizar esta traducción de un forma automática. Esto se conseguiría con la definición de una ontología que relacione los objetos ModBus con los Objetos BACnet.

Para reducir el costo se ha decidido realizar una aplicación que puede ser ejecutada en cualquier dispositivo, y cuyo fichero de salida con el mapping correspondiente, puede ser utilizado en un computador de placa reducida, como por ejemplo una Raspberry Pi. El pequeño programa que se ejecutaría en esa placa no está implementado, pero sería acceder a los datos que hay en el documento CSV y a partir del objeto ModBus que llega crear un objeto BACnet, ya que están especificadas todas las direcciones y registros correspondientes del objeto de entrada al de salida.

Otra posible ampliación que se podría realizar y que no soportan el resto de pasarelas del mercado actual, es incluir otros protocolos de entrada, con esto se conseguiría tener una integración total de varios tipos de protocolos, lo que hace que se puedan utilizar un mayor número de objetos distintos dentro del mismo sistema.

CAPÍTULO 5

Conclusiones

Se ha realizado una investigación para comunicar aparatos de conteo del consumo eléctrico de una casa, pero el sistema principal utiliza un protocolo diferente al de los contadores. Para solucionar este problema se ha implementado un sistema, que realiza la traducción del protocolo ModBus al protocolo BACnet. La aplicación genera un fichero CSV correspondiente a como crear los objetos del sistema principal con todos los datos de los contadores.

La principal ventaja del sistema creado respecto al resto de aplicaciones similares existentes en el mercado, es la simplicidad de la interfaz, lo que hace que casi toda la gente que la utilice entienda como funciona la aplicación de una forma rápida. Además, se ha tratado de optimizar el fichero de salida, para que tengas el mínimo número de datos posibles para poder realizar el proceso correctamente, y que se pueda realizar la conversión sin necesidad de comprar un dispositivo específico.

Respecto a la tesis, creo que se han conseguido alcanzar los objetivos previstos y la simplicidad buscada. Se ha conseguido diseñar una solución que satisfaga el problema propuesto, mejorando en algunos aspectos las soluciones existentes en el mercado actualmente, y se han conseguido proponer nuevas líneas de investigación futuras para seguir mejorando el sistema.

Tras la investigación, análisis, diseño y desarrollo de este sistema, considero que se han alcanzado los requisitos que se esperaban de esta Tesis de Fin de Máster, e incluso se han contemplado ciertos atributos de calidad como la posibilidad de utilizar la aplicación en distintos dispositivos (Portabilidad) o la simplicidad de uso (Usabilidad). Además, se han llegado a las siguientes conclusiones:

Por otro lado, con respecto a la aplicación Web, creo que el uso de la librería "Bootstrap" ha sido un acierto. Dado que se buscaba la simplicidad, no era necesaria una gran personalización de la misma y así se pudo realizar una web responsiva. También se ha utilizado librería Flask junto con la herramienta PythonAnywhere para hacer que la aplicación pueda ser desplegada en un servidor, lo que hace que haya podido aprender una librería nueva y esto me ha ayudado a poder desplegar la aplicación en la nube.

Por último, me gustaría comentar que por primera vez me he enfrentado a una investigación "real", ya que además de realizar el estado del arte había que analizarlo tratar de buscar una solución que mejore lo ya existente, y creo que he

sabido adaptarme con facilidad a los diferentes problemas que han ido surgiendo en el desarrollo. Por ello estoy satisfecho de la tesis y del trabajo realizado.

Bibliografía

- [1] AnyBus BACnet/Modbus Gateway. Consultado en <https://www.cimetrics.com/products/b6131>. Último Acceso: 24/05/2017.
- [2] BACnet Interface to Modbus Devices - Module. Consultado en http://www.anybus.de/products/bacnet_modbus_gateway.shtml. Último Acceso: 24/05/2017.
- [3] ANSI. *A Data Communication Protocol for Building Automation and Control Networks*. ANSI/ASHRAE Standard 135-2016.
- [4] BASautomation. *BASgatewayLX - ModBus to BACnet Converter*. Contemporary Controls, DS-BASGTWY0-BC3, October, 2013.
- [5] Connect Modbus RTU Slaves to BACnet/IP. Consultado en <http://www.rtaautomation.com/product/460mbs/>. Último Acceso: 24/05/2017.
- [6] Contemporary Controls. *Mapping of Modbus registers to BACnet objects using the BASremote*. Contemporary Controls, AN-BASREMO01-BA1, Agosto, 2014.
- [7] Domótica mediante Protocolo X10. Consultado en <http://www.monografias.com/trabajos101/domotica-protocolo-x10/domotica-protocolo-x10.shtml>. Último Acceso: 24/05/2017.
- [8] How X10 Works. Consultado en <http://www.smarthomeusa.com/how-x10-works/>. Último Acceso: 24/05/2017.
- [9] Información Detallada sobre el Protocolo Modbus. Consultado en <http://www.ni.com/white-paper/52134/es/>. Último Acceso: 24/05/2017.
- [10] Modbus RTU to BACnet IP Gateway | BACnet IP to Modbus RTU Protocol Converter. Consultado en <http://www.protoconvert.com/DirectSolutions/GatewayGrid/ModbusRTUtoBACnetIPGateway%7C%7CBACnetIPtoMod.aspx>. Último Acceso: 24/05/2017.
- [11] Modbus RTU to BACnet MS/TP or BACnet/IP QuickServer Gateways. Consultado en <http://www.prosoft-technology.com/Products/Gateways/Building-Automation-Systems/Modbus-RTU-to-BACnet-MS-TP-or-BACnet-IP-QuickServer-Gateways>. Último Acceso: 24/05/2017.

- [12] Modbus TCP/IP to BACnet/IP Client Gateway. Consultado en <http://www.prosoft-technology.com/Products/Gateways/Modbus-TCP-IP/Modbus-TCP-IP-to-BACnet-IP-Client-Gateway>. Último Acceso: 24/05/2017.
- [13] Modbus to BACnet Gateway. Consultado en <http://www.cbmsstudio.com/modbus-to-bacnet-gateway.html>. Último Acceso: 24/05/2017.
- [14] Modbus to BACnet Protocol Converter. Consultado en http://www2.schneider-electric.com/resources/sites/SCHNEIDER_ELECTRIC/content/live/FAQS/294000/FA294960/en_US/e8951_d0116_SpecSheet.pdf. Último Acceso: 24/05/2017.
- [15] Raquel Sánchez Díaz . *CANopen*. Departamento de Informática y Automática, Facultad de Ciencias, Universidad de Salamanca, junio, 2007.
- [16] The M-Bus. Consultado en <http://www.m-bus.com>. Último Acceso: 24/05/2017.
- [17] Modbus-IDA. *ModBus Application Protocol Specification*. <http://www.Modbus-IDA.org>, 28/12/2006.
- [18] Modbus to BACnet Bridge. Consultado en https://www.niagaramarketplace.com/asb_ListingDetail?listingId=a0nD000000CJdQBIA1&appType=Product. Último Acceso: 29/05/2017.
- [19] X3D. Consultado en <https://domotique.ooreka.fr/astuce/voir/155607/le-protocole-x2d-x3d-comment-ca-marche-pour-qui>. Último Acceso: 24/05/2017.
- [20] ¿Qué es XBee?. Consultado en <http://xbee.cl/que-es-xbee/>. Último Acceso: 12/06/2017.
- [Selwyn Piramuthu, 2008] Selwyn Piramuthu (2008) Input data for decision trees. *Expert Systems with Applications*, 34, 1220–1226.
- [21] Jennifer S. Light. When computers were women. *Technology and Culture*, 40:3:455–483, juliol, 1999.
- [22] Georges Ifrah. *Historia universal de las cifras*. Espasa Calpe, S.A., Madrid, sisena edició, 2008.

APÉNDICE A

Manual de usuario

A.1 Acceso a la web

Para acceder a la aplicación Web, el usuario debe acceder a la misma utilizando el protocolo HTTPS independientemente de si utiliza el navegador de un ordenador o un dispositivo móvil.

La dirección Web de la aplicación es la siguiente:

`http://josemagf.pythonanywhere.com/`

Una vez accedido se le mostrará al usuario la página principal en la que debe de indicar el nombre que se le va a dar al proceso.

EN esta pantalla el usuario deberá introducir el nombre que quiera dar a la traducción de ModBus a BACnet en el campo de texto de la parte superior, cabe destacar que la cadena de caracteres que se introduzca va a dar nombre al fichero de salida de la aplicación. Por otra parte en la parte inferior se podría ver una tabla con los registros añadido, como se acaba de iniciar el proceso la tabla se encuentra vacía.

Una vez rellenado ese campo y para pasar al siguiente paso el usuario debe de pulsar el botón que parece debajo de la tabla que pone ".Add register".

A.2 Añadir un registro

Una vez en la pestaña de añadir un registro, el usuario puede ver un formulario que debe de rellenar con distintos tipos de campos. Este formulario está dividido en dos. Por un lado tenemos los datos relativos al objeto ModBus, que son los que se ven en el recuadro verde. En este caso debe de indicar en el primer cuadro de texto la dirección del esclavo t debe de elegir el tipo de objeto que es, a través de unos radiobotones.

Por otro lado, en el recuadro rojo se pueden ver los datos que el usuario debe de rellenar, relativos al objeto BACnet. Lo primero que debe indicar es el nombre que va a recibir el objeto, esta cadena de texto no puede contener comas y debe de tener una extensión entre 1 y 64 caracteres. El usuario también debe de indicar

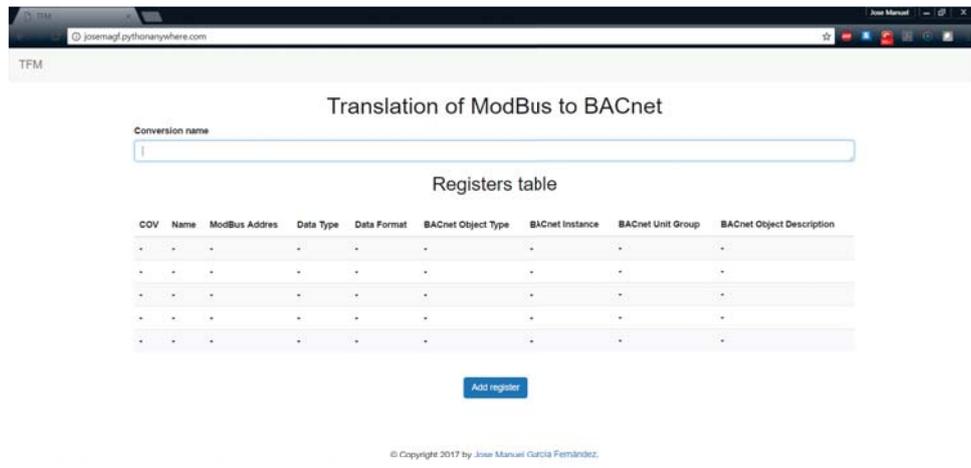


Figura A.1: Interfaz de inicio de la aplicación

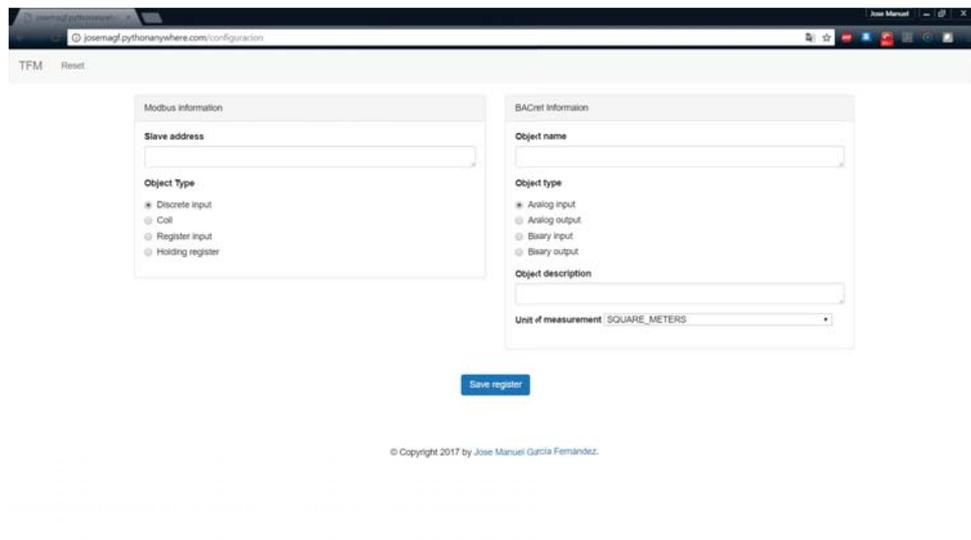


Figura A.2: Interfaz para añadir una registro.

el tipo de objeto BACnet del que se trata, a través de unos radiobotones. Otro campo de texto que debe de rellenar el usuario es la descripción del objeto, esta puede ser opcional, pero el campo no puede dejarse en blanco, por lo que al menos hay que introducir un caracter. Y el último valor a elegir es la unidad de medida del objeto, en este caso el usuario debe de elegir una en una lista clasificada por categorías, con todas las unidades de medida existentes.

Una vez rellenados todos los campos el usuario podrá añadir un registro con esos datos pulsando en el botón que aparece en la parte inferior y que pone "Save register".

En la parte superior aparece una barra de navegación en la que el usuario puede volver al inicio de la aplicación y borrar todos los datos introducidos hasta el momento, pulsando el botón que pone "Reset".

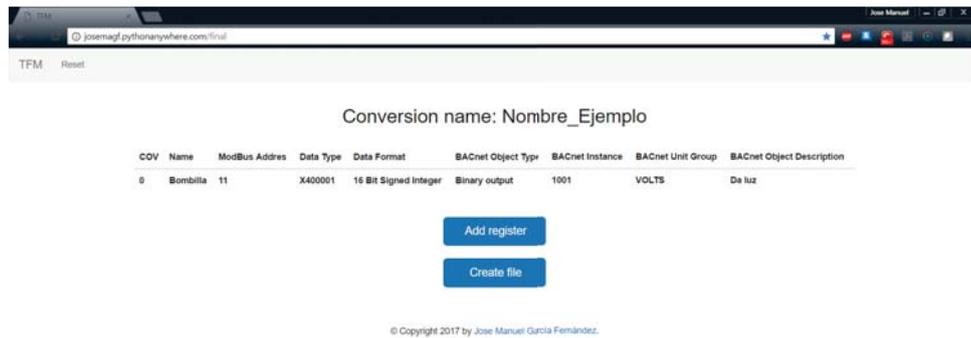


Figura A.3: Interfaz para ver los registros añadidos hasta el momento.

```

1 COV,Name,ModBus Address,Data Type,Data Format,BACnet Object Type,BACnet Instance,BACnet Unit Group,BACnet Object Description,
2 0,Ejemplo1,11,X400001,Bit,Binary output,1001,MEGAVOLT_AMPERES,-,
3 0,Example2,11,X400002,16 Bit Signed Integer,Analog output,1002,SQUARE_METERS,Description ,
4 0,Example3,11,X400003,Bit,Analog input,1003,PERCENT_PER_SECOND,-,
5

```

Figura A.4: Fichero resultante de la aplicación

A.3 Ver los registros introducidos

Esta es la última interfaz de la aplicación, en la parte superior se puede ver el nombre que se le ha dado en el primer paso a la traducción, justo debajo se puede ver una tabla con todos los registros introducidos anteriormente, en cada columna de la tabla se pueden todas y cada una de las características correspondientes a cada uno de ellos, toda la información que se puede ver en la tabla es la que va a aparecer en el fichero que se va a generar.

Debajo de la tabla hay dos botones, uno que pone "Add register" que sirve para añadir un nuevo registro, por lo que se vuelve a la interfaz descrita en el apartado anterior, el otro botón es el de "Create file", este es el paso final para terminar el proceso, ya que se generará un fichero CSV con todos los datos de la tabla y con el nombre que se ha indicado, este fichero se descargará automáticamente.

En la imagen A.4 se puede ver el formato del fichero de salida, en la primera fila se puede ver el título de cada columna, y las posteriores filas se corresponden con todos los registros añadidos, cada fila es un registro.

En la parte superior aparece una barra de navegación en la que el usuario puede volver al inicio de la aplicación y borrar todos los datos introducidos hasta el momento, pulsando el botón que pone "Reset".

APÉNDICE B

Código fuente

El código fuente de la aplicación creada se encuentra en el siguiente repositorio:

<https://github.com/josemanuelgarcia/TFM>