

Trabajo Fin de Grado en Ingeniería de las Tecnologías Industriales

Integración de dispositivos Modbus en Home Assistant para monitorización y gestión energética

Autor: Diego Campos Colinet

Tutor: Manuel Garrido Satué

**Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla**

Sevilla, 2025



Trabajo Fin de Grado
en Ingeniería de las Tecnologías Industriales

Integración de dispositivos Modbus en Home Assistant para monitorización y gestión energética

Autor:

Diego Campos Colinet

Tutor:

Manuel Garrido Satué

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2025

Trabajo Fin de Grado: Integración de dispositivos Modbus en Home Assistant para monitorización y gestión energética

Autor: Diego Campos Colinet
Tutor: Manuel Garrido Satué

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

Me gustaría comenzar agradeciendo a mi familia, que ha sido la más importante durante mi etapa universitaria. A mis padres, mi hermano, mi tía y a mi abuela, por su apoyo diario, cariño, compañía y preocupación por mi en todo momento.

También quiero agradecer a mi tutor, Manuel, por su gran ayuda para poder realizar este trabajo.

Por último, gracias a mis amigos y compañeros de carrera, por estar siempre presentes y hacer más llevaderos momentos duros durante esta etapa.

Diego Campos Colinet

Sevilla, 2025

Resumen

Debido al notable incremento en el número de instalaciones fotovoltaicas de autoconsumo que se está produciendo en los últimos años, surge la necesidad de optimizar su funcionamiento maximizando el aprovechamiento de la energía generada, con el objetivo de lograr un mayor ahorro económico. La solución planteada en este trabajo consiste en el desarrollo de una instalación domótica, manejada en el software Home Assistant, capaz de monitorizar los datos de la instalación fotovoltaica y actuar sobre distintos dispositivos de la vivienda. Entre las diferentes opciones disponibles en el mercado, se ha optado por integrar todos los dispositivos, tanto sensores como actuadores, mediante el uso del protocolo de comunicación Modbus.

En el desarrollo del proyecto se presentan dos alternativas para la toma de datos del medidor, una conexión directa mediante cableado y una conexión inalámbrica mediante una placa ESP32. Tras la toma de datos, se amplía la instalación con un módulo de relés y un termostato digital, demostrando la capacidad del sistema para controlar remotamente algunos de los consumos del hogar como los provenientes de la climatización o la iluminación.

El proyecto sienta así las bases para una gestión de la energía más eficiente, orientada a aumentar el gasto de energía producida por la fotovoltaica y reducir la importada desde la red.

Abstract

Due to the significant increase in the number of self-consumption photovoltaic installations in recent years, there is a need to optimize their operation by maximizing the use of the energy generated, with the goal of achieving greater economic savings. The solution proposed in this work consists of developing a home automation system, managed using Home Assistant software, capable of monitoring photovoltaic installation data and acting on various devices in the home. Among the different options available on the market, the choice was made to integrate all the devices, both sensors and actuators, using the Modbus communication protocol.

During the project development, two alternatives are presented for collecting data from the meter, a direct connection via wiring and a wireless connection using an ESP32 board. After data collection, the installation is expanded with a relay module and a digital thermostat, demonstrating the system's capability to remotely control some of the household's consumption, such as air conditioning or lighting.

The project thus lays the foundation for more efficient energy management, aimed at increasing the consumption of energy produced by photovoltaics and reducing the amount imported from the grid.

Índice Abreviado

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
1 Introducción	1
1.1 Objetivo	4
2 Modbus	5
2.1 Arquitectura	5
2.2 Capa aplicación	6
2.3 Modbus serie	8
2.4 Capa física	9
3 Dispositivo para la toma de datos y Home Assistant	11
3.1 Eastron SDM230 Modbus	11
3.2 Home Assistant Green	15
3.3 Home Assistant OS	16
4 Alternativas de conexión para la toma de datos	19
4.1 Alternativa 1	19
4.2 Alternativa 2	23
4.3 Comparación de alternativas	30
5 Ampliación de la instalación	31
5.1 Modbus rtu 16-CH relay module	32
5.2 Termostato CX3-S4DM	36
5.3 Propuesta de instalación	39
6 Conclusiones	41
6.1 Objetivos alcanzados	41
6.2 Posibles ampliaciones	41
A Anexo: Códigos	43
A.1 Eastron SDM230 Modbus: Alternativa 1	43
A.2 Eastron SDM230 Modbus: Alternativa 2	44
A.3 Modbus rtu 16-CH relay module	45
A.4 Termostato CX3-S4DM	47
B Anexo: Hojas de datos	49
B.1 Eastron SDM230 Modbus	49
B.2 Home Assistant Green	58

B.3	Termostato CX3-S4DM	61
	<i>Índice de Figuras</i>	67
	<i>Índice de Tablas</i>	69
	<i>Índice de Códigos</i>	71
	<i>Bibliografía</i>	73

Índice

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
1 Introducción	1
1.1 Objetivo	4
2 Modbus	5
2.1 Arquitectura	5
2.2 Capa aplicación	6
2.2.1 Código de función	6
2.2.2 Mensajes de excepción	7
2.3 Modbus serie	8
2.3.1 Tramas modo RTU	8
2.4 Capa física	9
3 Dispositivo para la toma de datos y Home Assistant	11
3.1 Easton SDM230 Modbus	11
3.1.1 Comunicación	12
3.1.2 Conexionado	13
3.2 Home Assistant Green	15
3.2.1 Conexionado	16
3.3 Home Assistant OS	16
4 Alternativas de conexión para la toma de datos	19
4.1 Alternativa 1	19
4.1.1 Programación	20
4.2 Alternativa 2	23
4.2.1 ESP32 DevKit V1	23
4.2.2 Módulo MAX485	24
4.2.3 Conexionado	25
4.2.4 Programación	26
4.3 Comparación de alternativas	30
5 Ampliación de la instalación	31
5.1 Modbus rtu 16-CH relay module	32
5.1.1 Conexionado	32
5.1.2 Comunicación Modbus	33
5.1.3 Programación	34
5.2 Termostato CX3-S4DM	36
5.2.1 Conexionado	36

5.2.2	Comunicación Modbus	37
5.2.3	Programación	37
5.3	Propuesta de instalación	39
6	Conclusiones	41
6.1	Objetivos alcanzados	41
6.2	Posibles ampliaciones	41
A	Anexo: Códigos	43
A.1	Eastron SDM230 Modbus: Alternativa 1	43
A.2	Eastron SDM230 Modbus: Alternativa 2	44
A.3	Modbus rtu 16-CH relay module	45
A.4	Termostato CX3-S4DM	47
B	Anexo: Hojas de datos	49
B.1	Eastron SDM230 Modbus	49
B.2	Home Assistant Green	58
B.3	Termostato CX3-S4DM	61
	<i>Índice de Figuras</i>	67
	<i>Índice de Tablas</i>	69
	<i>Índice de Códigos</i>	71
	<i>Bibliografía</i>	73

1 Introducción

La producción energética en España está sufriendo grandes cambios en los últimos años, debido al crecimiento de la producción energética proveniente de energías renovables. En 2024 el 56.8 % del total de la energía producida en España fue de origen renovable, habiendo aumentado en los últimos años de forma continuada la potencia instalada de energía generada por fuentes renovables hasta alcanzar los 85000 MW aproximadamente.

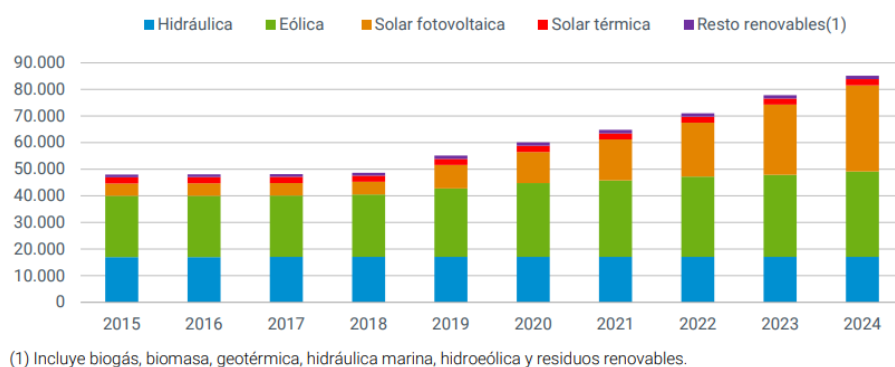


Figura 1.1 Evolución de la potencia instalada de generación renovable en MW [4].

En la figura anterior, también se puede observar como la energía solar fotovoltaica ha ido ganando importancia con el transcurso del tiempo, alcanzando en estos últimos años los niveles de potencia instalada de la energía eólica, la cual lleva dominando esta estadística durante años. En cuanto a producción energética la solar fotovoltaica también se sitúa como la segunda de las renovables con un 17 % del total de la producción energética española por detrás de la eólica con un 23.2 %. Aunque observando la tendencia de la potencia instalada, es cuestión de tiempo que se establezca la energía solar fotovoltaica como la mayor fuente de producción energética del país.

Este incremento de la producción solar fotovoltaica viene de la mano con un gran aumento en el ámbito de las instalaciones fotovoltaicas de autoconsumo en viviendas, ya que en apenas 7 años se ha multiplicado por 200 el número de este tipo de instalaciones en España, llegando en 2024 a 2281 MW de potencia instalada en viviendas.

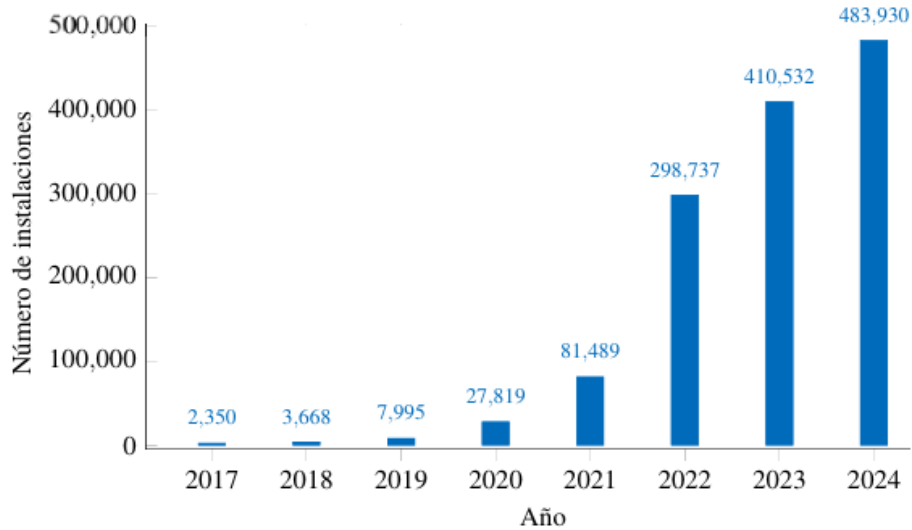


Figura 1.2 Evolución del número total de instalaciones fotovoltaicas residenciales en España [3].

Este aumento se debe a múltiples factores como los avances tecnológicos en los paneles, las nuevas políticas gubernamentales que fomentan el abandono de combustibles fósiles, el reparto de subvenciones y ayudas para instalaciones o la reducción de costes para realizar una instalación que se ha dado en los últimos 10 años. Además, económicamente para los consumidores llegan a ser rentables a no muy largo plazo. En 2024 las instalaciones de autoconsumo supusieron un ahorro promedio estimado de 157€ por kW instalado, suficiente para recuperar la inversión inicial en menos de 10 años, otro dato que refuerza el porqué de este aumento tan pronunciado en el número de instalaciones.

Con el aumento de instalaciones mostrado, una tarea pendiente es el aprovechamiento de los datos que se pueden obtener de las instalaciones. Debido a que en muchos casos no se instalan baterías que permitan almacenar la energía generada, el análisis de datos puede ser clave para conseguir un mayor ahorro económico, reduciendo consumos evitables que se produzcan en horarios que no haya producción energética.

Para una mejor comprensión de este problema, se van a mostrar ejemplos de mi propia vivienda, en la cual no se aprovechan los datos para la optimización del consumo. Las gráficas y datos que se van a mostrar a continuación se han obtenido de Fusion Solar, aplicación que registra datos de la instalación como la producción fotovoltaica, el autoconsumo o la cantidad de electricidad exportada o importada desde la red. En primer lugar, se muestra en la Figura 1.3 como es la curva de producción eléctrica en un día soleado.

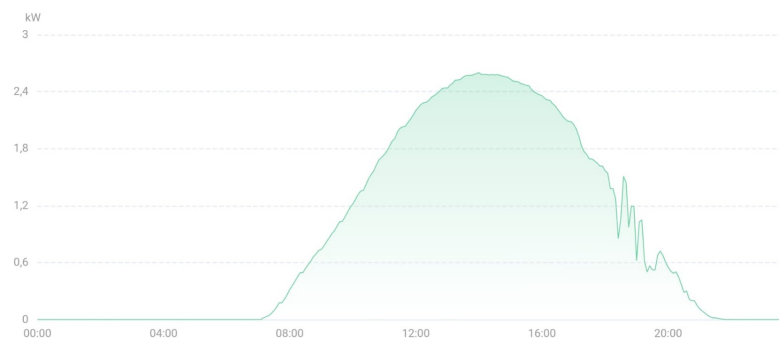


Figura 1.3 Curva diaria de producción fotovoltaica.

Se observa como la producción fotovoltaica comienza al amanecer, llegando a los máximos al mediodía y a partir de ahí reduciéndose progresivamente hasta el atardecer. Lo que interesa económicamente es aprovechar en mayor medida la energía producida y reducir lo máximo posible la importación de energía desde la red.

Al no poseer una instalación que aproveche los datos de la fotovoltaica, sumado a que los días laborables no hay nadie en la vivienda durante la mañana, mucha de la energía producida no se aprovecha y se acaba exportando a la red. Esta situación se puede ver en la siguiente figura, siendo los tramos de exportación en los que no coinciden las gráficas verde y naranja.

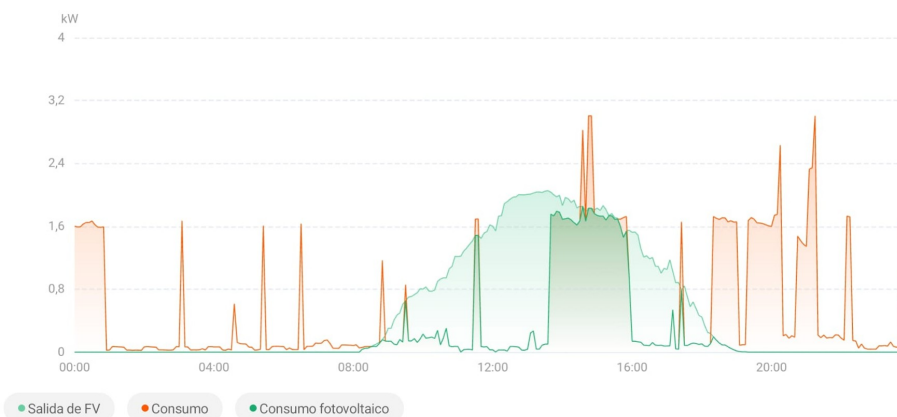


Figura 1.4 Datos de producción y consumo fotovoltaico.

Se observa cómo se realizan consumos importantes en horas sin producción y se desaprovecha energía en tramos de alta producción eléctrica. Esta situación repetida de forma continuada da como resultados las estadísticas de la Figura 1.5 que reflejan los datos del aprovechamiento de la energía producida en los 5 primeros meses de 2025.



Figura 1.5 Estadísticas de consumo y exportación de energía.

Únicamente se acaba consumiendo en la vivienda el 31.57% de la energía producida e importando desde la red el 60.38% de la consumida. Aunque los datos de energía exportada sean bastante altos, al exportar la energía producida, se está vendiendo la electricidad a un precio más bajo del que cuesta importarla desde la red, por lo que se puede sacar como conclusión, que lo más rentable económicamente, sería aumentar el porcentaje de autoconsumo y autosuficiencia.

Para aumentar estos porcentajes, lo ideal sería el desplazamiento del consumo en la medida de lo posible hacia el horario de producción fotovoltaica, para que cuando llegue las horas en las que se reduce la producción, se reduzca en la manera de lo posible la electricidad importada desde la red. Es verdad que hay consumos que no se pueden automatizar, como los provenientes de cocinar. Pero otros como la climatización del hogar o el control del consumo de un termo eléctrico son tareas de alto consumo energético que sí podrían ser optimizadas. Este problema podría ser solucionado a través de una instalación domótica.

La domótica es un ámbito de la tecnología diseñada para la mejora de nuestras vidas en muchos ámbitos como la seguridad, accesibilidad, ahorro energético, ahorro económico o comodidad. Todo esto mediante la

agrupación de los diferentes sistemas de la vivienda, de forma que la información que proporcionan pueda ser utilizada de forma conjunta por la totalidad de los sistemas agrupados. [5]

En el ámbito de la domótica, existen diversos protocolos de comunicación que permiten la interconexión de dispositivos dentro del hogar.

- KNX: uno de los estándares más consolidados en Europa, utilizado principalmente en instalaciones cableadas por su fiabilidad y escalabilidad, ideal para sistemas complejos en viviendas y edificios.
- Modbus: originado en el sector industrial, también se emplea en domótica para integrar dispositivos como sensores, controladores y sistemas de climatización, gracias a su simplicidad y compatibilidad con múltiples plataformas.
- Zigbee y Z-Wave: son una opción para realizar instalaciones inalámbricas, tienen un bajo consumo energético y un precio reducido respecto a otras tecnologías. Son sistemas fiables de comunicaciones con topología mallada.
- LonWorks: usado sobre todo en el mercado de Estados Unidos, destaca por su arquitectura descentralizada y su facilidad de expansión y escalabilidad. Se forma por una red de nodos independientes interconectados.

En relación con el control de instalaciones de hogares inteligentes para poder monitorear sensores, realizar automatizaciones o manejar dispositivos de forma remota existen algunos softwares como Home Assistant, OpenHAB, Domoticz, Jeedom o Fibaro Home Center.

1.1 Objetivo

El objetivo de este proyecto es dar el primer paso para el montaje de una instalación domótica que permita aprovechar de forma más eficiente la energía eléctrica producida por una instalación fotovoltaica, con el fin de conseguir los beneficios ya mencionados previamente. Este primer paso consiste en la toma de datos de los excedentes de la producción fotovoltaica y gasto de electricidad proveniente de la red eléctrica de una vivienda. Para ello, se explicarán dos alternativas para la realización de la toma de datos. Además, se va a complementar mediante una ampliación de la instalación, en la cual se explicará la instalación y programación de dos dispositivos actuadores. Todos los dispositivos usados se integrarán y gestionarán en el software de Home Assistant, y se comunicarán entre ellos mediante el protocolo de comunicaciones Modbus. En la siguiente figura se muestra un esquema del proyecto de instalación.

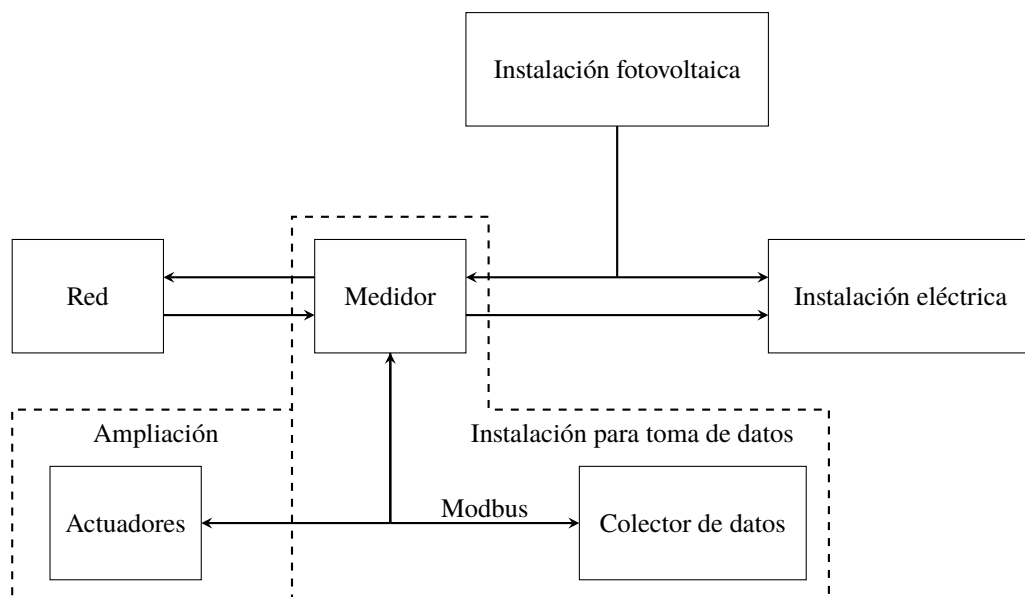


Figura 1.6 Esquema de la instalación.

2 Modbus

El protocolo de comunicaciones Modbus surge en el 1979 desarrollado por Modicon, es usado para la comunicación entre dispositivos electrónicos y posee una arquitectura maestro-esclavo. En este tipo de sistemas existe un maestro que puede leer o cambiar datos de los esclavos a través de peticiones. Una vez recibida la petición por el esclavo será procesada y emitirá una respuesta.

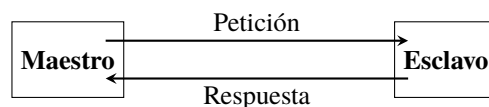


Figura 2.1 Esquema Petición-respuesta.

A los datos de los esclavos se les denomina registros y se agrupan en zonas de memoria, dependiendo del tipo de dato que sea. Existen cuatro zonas de memoria diferentes, cada una con un rango de números de registro, un rango de direcciones, un tamaño de registro y unos permisos de acceso. Estos permisos limitan las acciones que podrá realizar el maestro mediante una petición. [6]

Tabla 2.1 Características de las zonas de memoria.

Zona de memoria	Números de registro	Dirección	Tamaño (bits)	Permisos
Salida digital	1-9999	0000-270E	1	Leer/Escribir
Entrada digital	10001-19999	0000-270E	1	Leer
Entrada analógica	30001-39999	0000-270E	16	Leer
Salida analógica	40001-49999	0000-270E	16	Leer/Escribir

2.1 Arquitectura

En relación a los niveles del modelo OSI, que divide los protocolos en 7 capas separadas por sus funcionalidades, este protocolo de comunicaciones ocupa los niveles 1, 2 y 7, los cuales corresponden a la capa física, capa de enlace y capa de aplicación respectivamente.

- Nivel 1 Capa física: abarca todo lo relacionado al medio en el que se transmiten los datos.
- Nivel 2 Capa de enlace: gestiona como se empaquetan los datos que se envían en una comunicación.
- Nivel 7 Capa de aplicación: relaciona las funcionalidades que tiene el protocolo de comunicación.

En la Figura 2.2 se recoge los diferentes tipos de comunicación Modbus según las diferentes capas ya explicadas. En el proyecto únicamente se usa Modbus serie, en concreto el modo RTU, por lo que más adelante este es el único que se explicará.

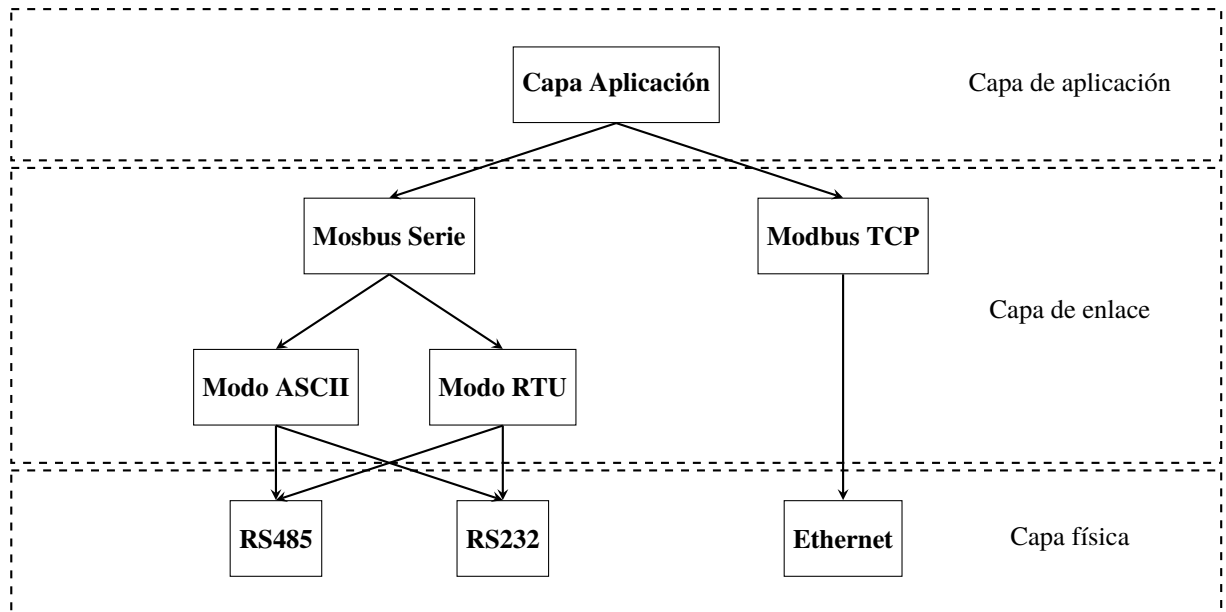


Figura 2.2 Esquema arquitectura modbus.

2.2 Capa aplicación

Como se ha mencionado anteriormente, esta capa define las diferentes funciones que se les puede asociar a cada petición enviada por un maestro, como los esclavos deben responder a una petición dependiendo de la función usada y como se codifican las solicitudes en un mensaje, también denominadas APDU (Application layer Protocol Data Unit). Dependiendo de la utilidad que posean, en el protocolo Modbus existen 3 tipos diferentes de APDU, siendo el tamaño máximo de una APDU 253 bytes.

- APDU de solicitud: es el enviado desde el maestro al esclavo con el que se pretenda establecer comunicación. Contiene el código de función y los datos de la solicitud.
- APDU de respuesta: es el enviado de esclavo a maestro en caso de comunicación exitosa. Contiene el código de función y los datos solicitados por la función.
- APDU de excepción: es el enviado de esclavo a maestro en caso de comunicación fallida. Contiene un código de función de excepción y el código de excepción asociado al fallo ocurrido.

2.2.1 Código de función

Las funciones definen que acción debe hacer y como debe responder el esclavo a una petición que ha sido recibida. Todas las funciones tienen un código de función asociado, que es el mencionado en la composición de las APDU. Cada código de función tiene asociada una estructura propia para la solicitud y otra para la respuesta, esta estructura será la que defina el tamaño de los datos enviados. Se ha recogido en la siguiente tabla algunas de las funciones principales de comunicación junto a su respectivo código de función. Además, para esas funciones se muestra en la Figura 2.2 la estructura de solicitud y respuesta correspondiente, correspondiendo cada cuadrado a un tamaño de un byte.

Tabla 2.2 Códigos de función.

Código Función	Función
01	Leer salidas digitales
02	Leer entradas digitales
03	Leer salidas analógicas
04	Leer entradas analógicas
05	Cambiar una salida digital
06	Cambiar una salida analógica
0F	Cambiar múltiples salidas digitales
10	Cambiar múltiples salidas analógicas

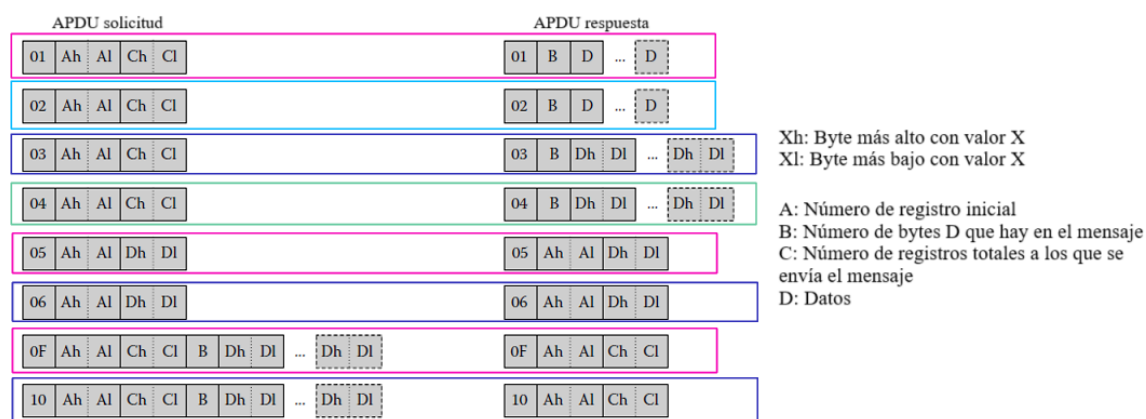


Figura 2.3 Estructura de mensajes Modbus según la función [6].

2.2.2 Mensajes de excepción

Hasta ahora se han explicado los casos en la que la comunicación se produce de manera correcta, pero hay casos en los que no es exitosa y puede ser por 3 motivos.

- El mensaje no llega al destinatario, por lo que no hay respuesta.
- El mensaje llega al destinatario con fallos de paridad o de CRC, en este caso no hay respuesta debido a que el esclavo ignora la petición.
- La petición llega sin errores al destinatario, pero este no puede atenderlo por algún motivo. Este es el caso en el que el esclavo responde con una APDU de excepción.

Como se ha explicado antes, las APDU de excepción tienen un código de función de excepción y un código de excepción asociado al fallo ocurrido. Los códigos de función de excepción están asociados a cada código de función y para calcularlos se pone a uno el bit más significativo del código de función y se pasa a hexagesimal. En la siguiente tabla se recogen dos ejemplos de este cálculo.

Tabla 2.3 Cálculo del código de función de excepción.

CF	CF binario	CF excepción binario	CF excepción
01	00000001	10000001	81
0F	00001111	10001111	8F

Los códigos de excepción explican el fallo que ha sucedido para que la comunicación no haya sido exitosa. Cada función puede tener unos fallos concretos, a continuación se muestran los códigos y su significado para

las funciones que se han explicado

Tabla 2.4 Códigos de excepción.

Código excepción	Significado
01	Función no válida para el esclavo
02	Número de registro no válido para el esclavo
03	Datos no válidos para el esclavo
04	Fallo del dispositivo esclavo
08	Error de paridad

2.3 Modbus serie

Este tipo de comunicación conecta todos los dispositivos a un único bus, para que el dispositivo maestro haga peticiones de forma individual o a todos los dispositivos del bus, esta segunda forma de comunicación también es llamada broadcast. En cuanto a los dispositivos, este tipo de comunicación lo forman un único maestro junto a un número de esclavos entre 1 y 247.

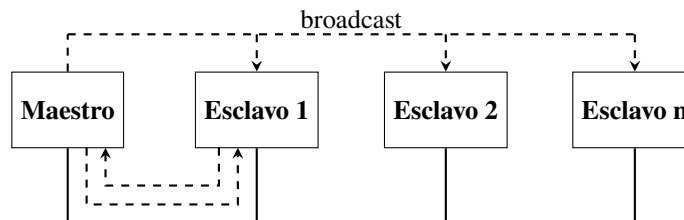


Figura 2.4 Esquema modbus serie.

2.3.1 Tramas modo RTU

Una trama es la forma en la que van empaquetados los datos en una comunicación. Como se ha visto en la Figura 2.2 dentro de Modbus serie los dos modos para realizar los envíos de datos son RTU y ASCII. Como es el usado, el que se va a explicar es el modo RTU aunque no son muy diferentes.

Las tramas RTU tienen la siguiente estructura:

- 1 Byte de dirección: Indica la dirección asociada al esclavo que se desea que reciba la petición, si es broadcast el byte será 00.
- 1 Byte de función: Indica el código de función.
- 0 - 252 Bytes de datos: Contienen la información que complementa a la función, como cambios en registros del esclavo o información solicitada.
- 2 Bytes de CRC: Mecanismo de detección de errores. Se calcula con todos los bytes anteriores que forman la trama.

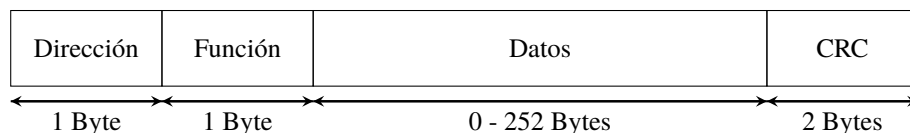


Figura 2.5 Transmisión de Trama RTU.

Por cada byte transmitido en un trama es necesario transmitir 11 bits, del siguiente tipo:

- 1 Bit de inicio (BI): Realiza la sincronización inicial e indica el comienzo de la trama.
- 8 Bit de datos (BD): Contiene los datos en binario, el bit menos significativo se envía en primer lugar.
- 1 Bit de paridad (BP): Permite la detección de errores.
- 1 Bit de parada (BS): Asegura un periodo entre transmisión de caracteres.

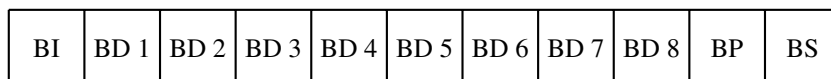


Figura 2.6 Transmisión de 1 byte.

Respecto a los requisitos de tiempo es necesario que tanto los caracteres de una misma trama, como unas tramas con otras estén separadas para evitar errores en la difusión de un mensaje. La distancia de separación entre caracteres es de 1.5 veces el tiempo de difusión de un caracter, y de 3.5 veces en el caso de separación entre tramas.

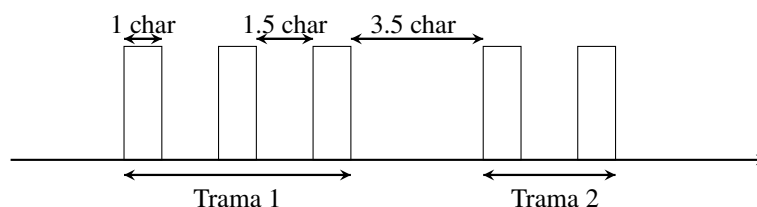


Figura 2.7 Requisitos de tiempo de Trama RTU.

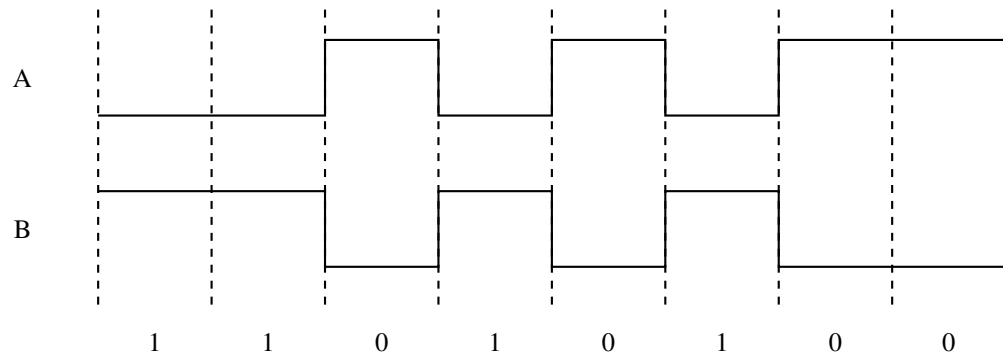
2.4 Capa física

La comunicación Modbus Serie soporta dos tipos de capa física que son RS232 y RS485.

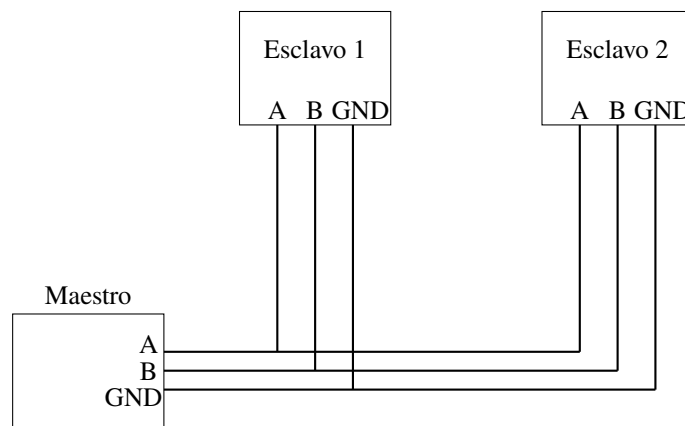
- RS232 es usado en distancias cortas, tiene una velocidad de transmisión media baja y solo puede ser conectado un único maestro con un único esclavo.
- RS485 soporta distancias más largas de hasta 1000 metros con una velocidad alta, dependiendo la velocidad máxima de la distancia. Es capaz de comunicar hasta 32 dispositivos por segmento, pudiéndose utilizar repetidores para conectar más segmentos.

Debido a sus características se ha escogido RS485 como método de conexión entre dispositivos. Algo a tener en cuenta es que para que funcione la comunicación correctamente los dispositivos de una misma red deben tener todos la misma configuración de modo de transmisión (ASCII/RTU), tipo de paridad (par/impar/sin paridad) y velocidad de transmisión (tasa de baudios).

RS485 es una línea diferencial en la que existen 2 fases. La fase A es baja para la lógica 1 y alta para la lógica 0. La fase B es alta para la lógica 1 y baja para la lógica 0. Además, opcionalmente puede haber en la conexión una línea de tierra. Para entenderlo mejor se representa en la figura como es la transmisión en A y B de una serie de bits. [1]

**Figura 2.8** Líneas A y B.

Para la conexión entre dispositivos se tiene que conectar las fases de un dispositivo con las fases del mismo nombre del siguiente. En la figura se puede ver con un ejemplo de 3 dispositivos.

**Figura 2.9** Conexión RS485.

3 Dispositivo para la toma de datos y Home Assistant

En el capítulo introductorio se pudo ver en la Figura 1.6 el esquema de la instalación que se pretende realizar, pero sin entrar en los dispositivos específicos que se van a usar. En este capítulo se recoge la descripción y conexionado del dispositivo que realiza la toma de datos, el Eastron SDM230 Modbus, el cual medirá la electricidad consumida en la vivienda proveniente de la red y el excedente de la instalación fotovoltaica. Por otra parte también se desarrolla el dispositivo que recibirá los datos siendo el maestro de la comunicación Modbus, el cual es el Home Assistant Green. Este dispositivo funciona con el software Home Assistant Operating System, que también será explicado.

3.1 Eastron SDM230 Modbus



Figura 3.1 Dispositivo Eastron SDM230 Modbus.

Este dispositivo es un medidor fabricado por la empresa Eastron, el cual tiene la función de medir parámetros eléctricos de una instalación eléctrica de una sola fase, y mostrarlos mediante la pantalla LCD que incorpora. Es capaz de realizar medidas de energía bidireccionales, por lo que es un dispositivo idóneo para una instalación fotovoltaica, ya que mide la energía consumida por la vivienda proveniente de la red y la energía exportada por ser excedentes. También posee un puerto de comunicación para la visualización remota de los parámetros medidos, algo que será necesario para la comunicación Modbus. A continuación, se recogen en la tabla algunas de sus características principales.

Tabla 3.1 Características Eastron SDM230 Modbus.

Físicas	
Ancho	36 mm
Alto	99 mm
Profundidad	63 mm
Eléctricas	
Rango de voltaje	176-276 V
Corriente máxima	100 A
Corriente mínima	0,5 A
Frecuencia	50/60 Hz
Consumo	<2 W/10 VA

Como se ha podido ver en la Figura 3.1 este dispositivo tiene 2 botones, el izquierdo cambia la medida mostrada en la pantalla cuando se pulsa, mientras que el botón derecho sirve para acceder a la configuración del dispositivo manteniéndose pulsado 3 segundos. Al acceder a la configuración primero se tiene que introducir la contraseña, que por defecto es 1000.

3.1.1 Comunicación

Para la visualización de datos de forma remota, el dispositivo posee dos opciones de comunicación, a través de una salida de pulsos o mediante una salida RS485 para Modbus RTU, la cual será la utilizada. Para que las peticiones Modbus se realicen correctamente se deben configurar los parámetros de comunicación entre la opciones disponibles. La siguiente tabla muestra tanto las opciones como la configuración elegida. Como se explicó en el capítulo anterior todos los dispositivos en una red RS485 tienen que tener las mismas características de comunicación, por lo que las características escogidas en este dispositivo se repetirán en todos los equipos que se configuren.

Tabla 3.2 Características de comunicación.

Parámetro	Opciones	Elección
Tasa de Baudios	1200/2400/4800/9600	9600
Paridad	Par/Impar/Ninguna	Ninguna
Bits de Parada	1/2	1
Dirección	1-247	1

Una vez configurados los parámetros de comunicación, es necesario conocer el registro y la dirección de los parámetros que mide el dispositivo para poder acceder a cada uno de ellos. Se recogen en la siguiente tabla los registros y direcciones de los parámetros que se han considerado interesantes de monitorear para posteriormente poder realizar automatizaciones. Si se desea obtener un parámetro adicional, en la Sección B.1 del Apéndice B se pueden encontrar todos los parámetros que mide el dispositivo junto a sus características.

Tabla 3.3 Características de parámetros.

Registro	Parámetro	Unidad	Dirección protocolo Modbus
30001	Tensión	V	0000
30013	Potencia activa	W	000C
30073	Energía activa exportada	kWh	0048
30075	Energía activa importada	kWh	004A

3.1.2 Conexionado

Respecto al conexionado para el funcionamiento del dispositivo y para la comunicación Modbus, se indica en la Figura 3.2 la función y localización física de cada puerto del dispositivo.



Figura 3.2 Localización y función de los puertos.

En una instalación real este dispositivo se coloca en el cuadro eléctrico de la vivienda entre el interruptor general automático y el interruptor diferencial. En el siguiente esquema se puede ver como es la conexión.

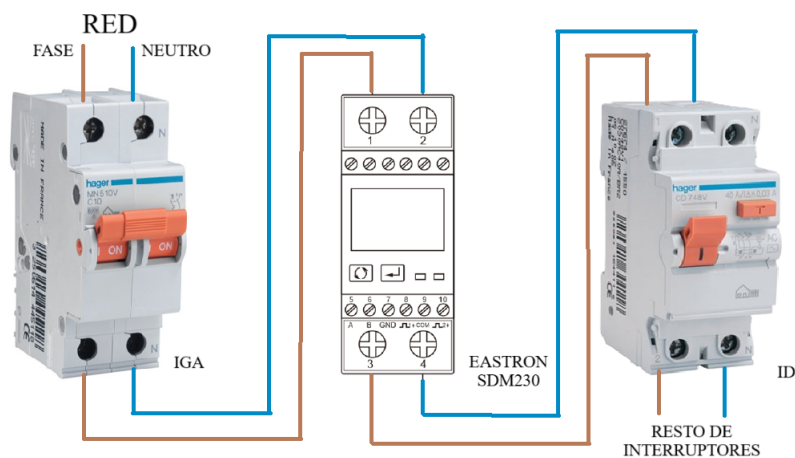


Figura 3.3 Instalación en cuadro eléctrico.

Debido a que instalarlo en la red de la vivienda sería algo tedioso, para poder probar su funcionamiento se ha optado por cortar una alargadera e instalar el dispositivo en donde se ha realizado el corte. Al conectar la alargadera a un enchufe y conectar algún electrodoméstico en el extremo de la alargadera se simula la introducción de electricidad desde la red a la vivienda, interpretándose el consumo que realiza el electrodoméstico como el consumo total de la vivienda. Siguiendo las indicaciones de la Figura 3.2 se realiza la conexión en la alargadera. Debido a que el dispositivo no tiene una entrada para el cable de tierra (verde/amarillo), se ha empalmado por fuera del dispositivo. En la siguiente imagen se puede ver como se ha conectado y como realiza una medición de la tensión de la red.

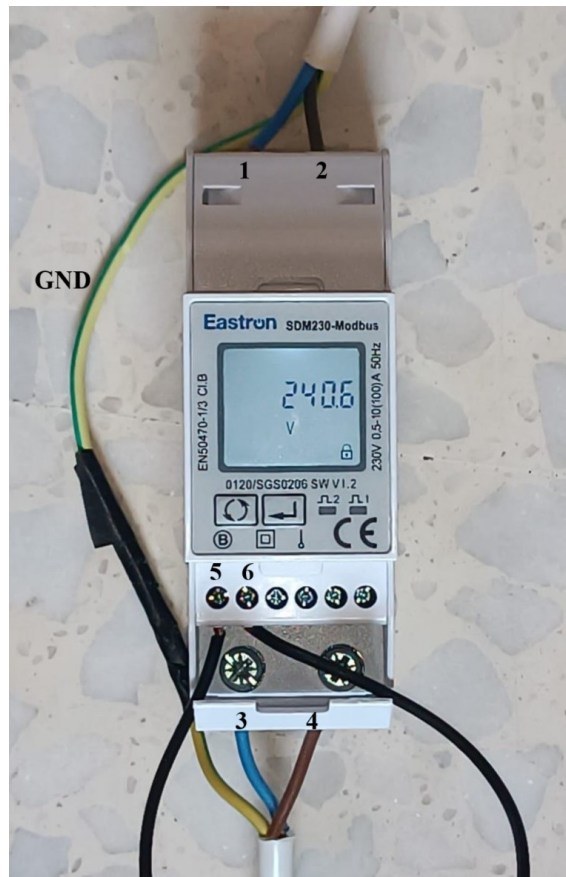


Figura 3.4 Conexión realizada.

3.2 Home Assistant Green

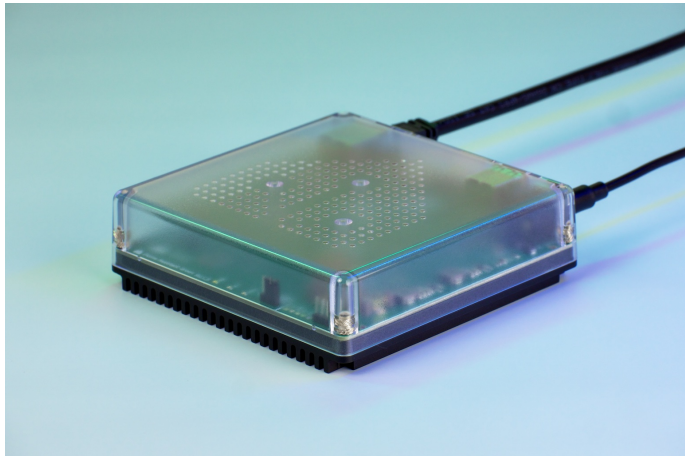


Figura 3.5 Home Assistant Green. Fuente: Internet.

Home Assistant Green es un dispositivo fabricado por la empresa Nabu Casa con el fin de acercar la domótica a todo tipo de usuarios, ya que este tiene una instalación muy sencilla y su configuración es bastante intuitiva. Su función es ser el controlador principal de una instalación domótica, recibiendo datos de sensores y ejecutando automatizaciones para hacer funcionar actuadores instalados en la vivienda cuando sea necesario. En la instalación tendrá la función de maestro de la comunicación Modbus.

Usa como sistema operativo Home Assistant OS que será desarrollada en el siguiente apartado. En cuanto a características del dispositivo se han recogido las más importantes en la siguiente tabla.

Tabla 3.4 Características Home Assistant Green.

Físicas	
Ancho	112 mm
Alto	32 mm
Profundidad	112 mm
Eléctricas	
Alimentación	12 V
Consumo	1.7 - 3 W
Técnicas	
Procesador	Rockchip RK3566
Almacenamiento interno	32 GB
Memoria RAM	4 GB
Puertos	
Ethernet	
2x USB 3.0 Tipo A	
Micro SD	
HDMI	

3.2.1 Conexionado

Para que el dispositivo funcione se conectará a la red con el cable de alimentación que viene incluido. Además, para la posterior programación es necesario conectar el dispositivo con un router Wifi vía cable Ethernet. Una vez conectado se podrá acceder a la configuración del dispositivo desde internet en el puerto 8123.



Figura 3.6 Puertos Home Assistant Green. *Fuente: Internet.*

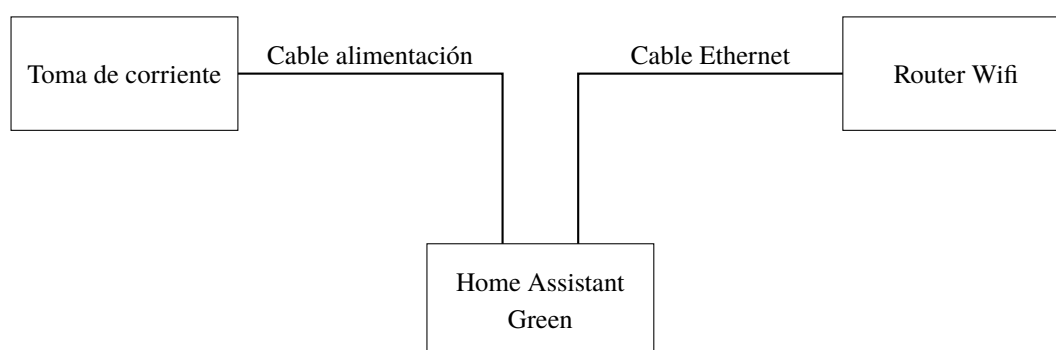


Figura 3.7 Esquema conexión Home Assistant Green.

3.3 Home Assistant OS

Como ya se ha mencionado anteriormente Home Assistant OS es un software usado para gestionar una instalación domótica y es usado por los dispositivos Home Assistant, aunque también es posible instalarlo en placas de desarrollo como Raspberry Pi. En este apartado se va a hacer una introducción a este software explicando la interfaz, mostrando las extensiones que se van a usar posteriormente y explicando como se programa la configuración de comunicación.

En primer lugar, para acceder al programa, el ordenador que se use tiene que estar conectado a la señal Wifi emitida por el router Wifi que esté conectado por cable ethernet al Home Assistant Green. Una vez conectado, se tiene que acceder al puerto de internet 8123. Esto se puede hacer introduciendo la siguiente url en un buscador como Google Chrome “<http://homeassistant.local:8123/>”. Una vez se acceda será necesario crear o introducir una cuenta de Home Assistant para acceder al programa.

Cuando se accede al programa se mostrará por pantalla algo parecido a la Figura 3.8. En la zona izquierda del programa se muestran diferentes ventanas y complementos que se pueden descargar para añadir funcionalidades. En el resto de la pantalla se muestra el contenido de la ventana seleccionada. Por ejemplo en la Figura 3.8 está seleccionada la ventana resumen y en la derecha se puede ver el contenido, que consiste en parámetros que se han configurado para que aparezcan en esta ventana.

Los complementos que no están incorporadas de serie, y por lo tanto es necesaria su descarga son “File Editor” y “ESPHome Device Builder”. Para poder instalarlas hay que acceder a la ventana “Configuración”, a partir de ahí entrar en “Complementos” y por último en “Tienda de Complementos”. Una vez se llegue solo será necesario buscar los dos complementos mencionados e instalarlos. El uso que se les dará y su funcionamiento se explicará en el siguiente capítulo.

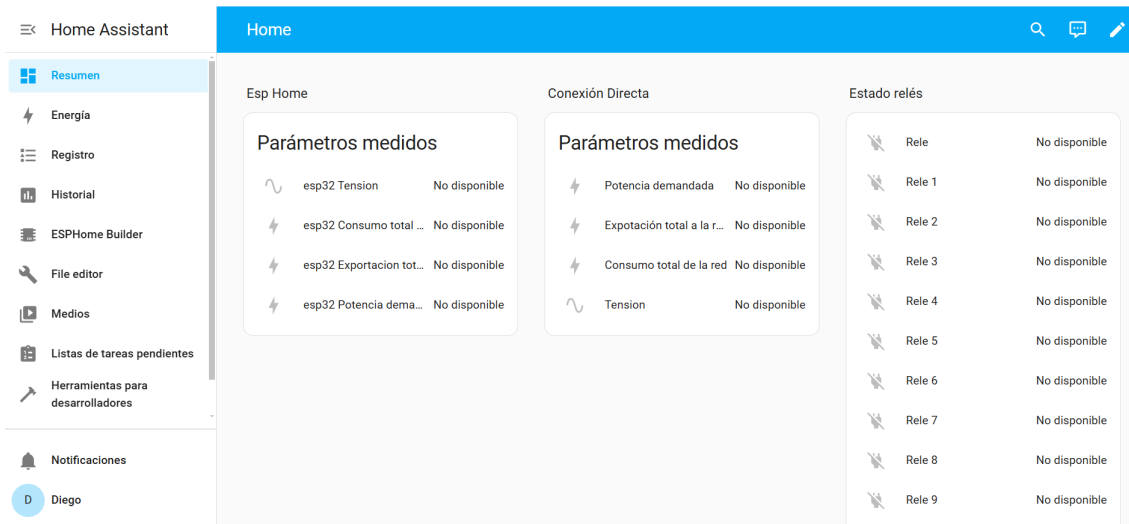


Figura 3.8 Interfaz de Home Assistant OS.

Respecto a como se programa en este software, se debe saber que en los dos complementos que se van a usar y en muchas otras aplicaciones de Home Assistant OS, para programar se usan archivos YAML. Este es un lenguaje diseñado para estructurar datos de forma que sea legible para una persona, ya que posee una sintaxis muy sencilla, la cual se basa en la sangría para representar los datos. Los elementos que se crean se definen mediante características y se sigue una jerarquía en función de la sangría.

Código 3.1 Ejemplo archivo YAML.

```
protocolo_comunicacion:
- nombre: Modbuscom
  tipo: Mosbus_rtu
  sensores:
    - nombre: sensor1
      dirección: 1
    - nombre: sensor2
      dirección: 2
```

En este código se ejemplifica como es un archivo de este tipo. En primer lugar, se indica que se va a crear un protocolo de comunicación y separándose mediante sangría comienza una lista de características en la que cada vez que se introduzca un guión (-) significará el comienzo de la definición de un protocolo de comunicación diferente. Tras definir las características, se indica la creación de un listado de sensores y se sigue el mismo procedimiento anterior separando con sangría e introduciendo un listado de características para cada sensor. Siempre que se defina una característica, esta debe ir seguida de dos puntos (:) y un espacio antes de asignarle un valor. Cuando se cargue el código en Home Assistant cada sensor creado aparecerá definido en el programa como una entidad.

4 Alternativas de conexión para la toma de datos

Este capítulo trata uno de los temas principales del trabajo, que es la explicación de diferentes conexiones entre el dispositivo medidor y el colector de datos. Se van a describir 2 alternativas para la conexión, finalizando con una comparativa entre ambas.

Para cada alternativa se hará una explicación de como se realiza la conexión, que dispositivos auxiliares son necesarios, se mostrarán imágenes reales de la instalación, y por último se explicará el código realizado en el software Home Assistant para la transmisión de datos.

4.1 Alternativa 1

La primera forma de conexión es la más simple de las dos, ya que consiste en conectar directamente el medidor Eastron con el Home Assistant Green para transmitir los datos. Como se ha podido ver anteriormente este dispositivo no posee entradas para la conexión RS485, por lo que para adaptar la entrada se conecta a uno de los puertos USB del Home Assistant un convertidor USB a RS485.



Figura 4.1 Adaptador USB a RS485 *Fuente: Internet.*

Para conectar el adaptador con el medidor Eastron basta con conectar las fases con la misma letra entre sí, A con A y B con B. Debido a que el adaptador USB no tiene entrada de tierra no se incluye cable GND.

En la siguientes figuras se pueden ver el esquema de conexión de esta alternativa, además del montaje realizado con los dispositivos reales.

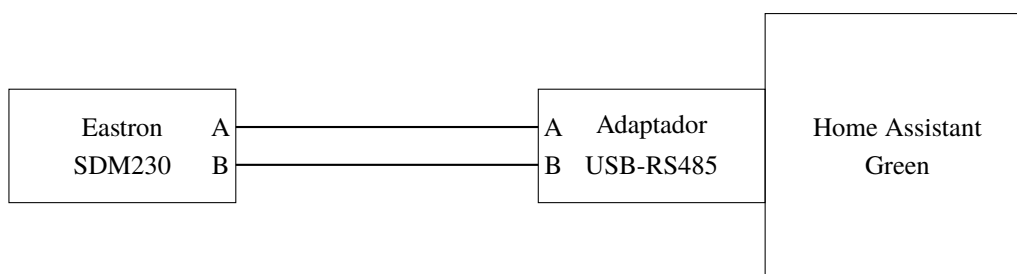


Figura 4.2 Conexión alternativa 1.

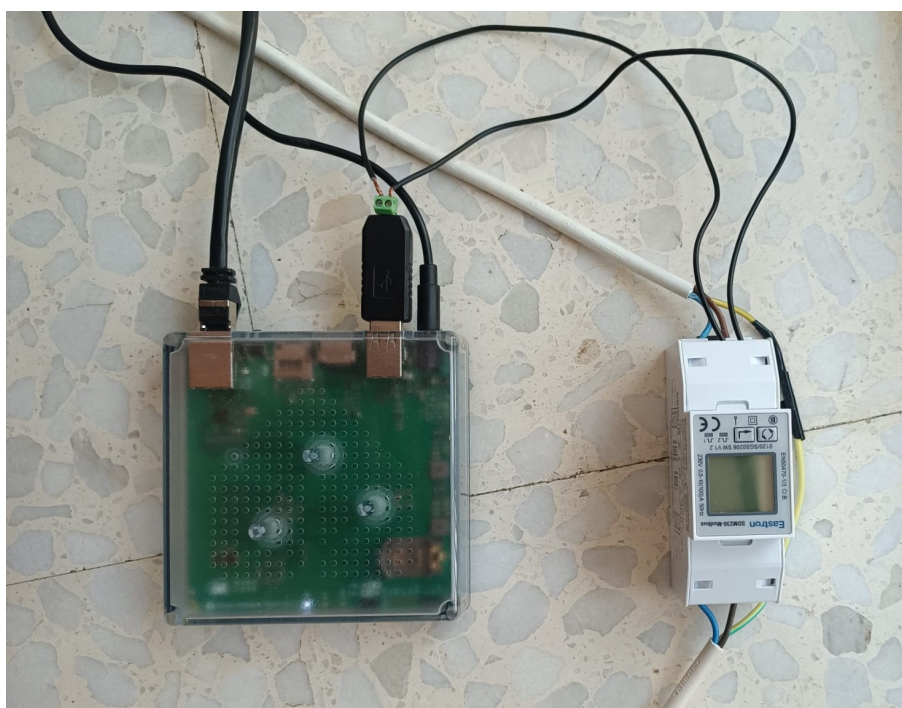


Figura 4.3 Montaje real alternativa 1.

4.1.1 Programación

La programación de este montaje se realiza desde el complemento “File Editor” de Home Assistant OS. Para ello se accede al archivo “configuration.yaml” y se edita directamente, prestando atención a no borrar el código escrito por defecto, y comenzando por debajo de este.

Es necesario saber que siempre que se realice una modificación del archivo en el que se esté programando, se deberá guardar la modificación y reiniciar el Home Assistant Green en la ventana “Herramientas para desarrolladores”. Esto se tiene que hacer para que el cambio se actualice en la comunicación entre dispositivos, y no solo en el archivo modificado.

El primer paso para la programación de la comunicación con el dispositivo Eastron SDM230 es la definición del tipo y las características del protocolo de comunicación que se va a establecer. En Home Assistant OS existe la orden “modbus” que indica el comienzo de la definición de las características de un protocolo de comunicación Modbus. En la Tabla 4.1 se recogen las características necesarias para una correcta configuración, junto a sus definiciones. Sabiendo que se va a realizar una comunicación Modbus RTU, como se indicó en el capítulo 2, junto a la configuración del dispositivo recogida en la tabla Tabla 3.2 del capítulo anterior, se le ha asignado su respectivo tipo o valor a las características en el Código 4.1.

Tabla 4.1 Características para configuración Modbus.

name	Nombre asignado a esta comunicación Modbus
type	Tipo de comunicación Modbus según la capa de enlace
port	Puerto físico del Home Assistant Green por el que se va a hacer la comunicación
baudrate	Velocidad de envío de datos de la comunicación serie en bits/s
bytesize	Número de bits de datos por byte
method	Tipo de comunicación Modbus serie
parity	Tipo de paridad usada
stopbits	Número de bits de parada
delay	Tiempo de espera en segundos para mandar el primer mensaje una vez hecha la conexión

Código 4.1 Configuración Modbus.

```

modbus:
- name: contador
  type: serial
  port: /dev/ttyUSB0
  baudrate: 9600
  bytesize: 8
  method: rtu
  parity: N
  stopbits: 1
  delay: 1

```

Ya realizada la configuración del tipo y los parámetros de la comunicación se pueden definir los datos que se van a medir. En este caso, los registros que pretendan medir se definen en Home Assistant como “sensors” debido a que son valores analógicos.

En la siguiente tabla se recogen las variables de configuración que se deben usar para acceder al registro y realizar una buena representación de la medida. En el programa realizado se han tomado las 4 medidas expuestas en la Tabla 3.3, pero debido a ser una programación muy repetitiva y poco interesante, en el Código 4.2 solo se muestra la configuración para el registro de la tensión. El programa completo, el cual recoge la configuración de todas la medidas y de la comunicación se puede consultar en en la Sección A.1 del Apéndice A.

Tabla 4.2 Ordenes para configuración de sensores.

name	Nombre asignado a la medida
adress	Número del registro de la medida que se pretenda consultar
slave	Dirección asignada al dispositivo al que se pretenda mandar la petición
data_type	Tipo de formato de la variable a consultar
device_class	Tipo de medida del registro
input_type	Indica la zona de memoria del registro
precision	Número de decimales en la medida
unit_of_measurement	Unidades del registro medido

Código 4.2 Configuración para lectura de entrada analógica.

```

sensors:
- name: Tension
  address: 0

```

```

slave: 1
data_type: float32
device_class: voltage
input_type: input
precision: 2
unit_of_measurement: V

```

Una vez realizada la programación para la obtención de los datos recogidos por el medidor, en la ventana “Resumen” de Home Assistant OS se podrán ver los parámetros medidos en cada instante. Además, pulsando en cualquier parámetro se podrá ver una gráfica que registra la evolución de la medida con el paso del tiempo. La siguiente imagen es una captura de los 4 parámetros que se han configurado para medirse y una gráfica de la evolución de la tensión durante una hora.



Figura 4.4 Registro y evolución de medidas.

Para la visualización de datos como la exportación y el consumo total es muy útil usar la ventana “energía”, esta permite introducir diferentes datos sobre la utilización de energía en la vivienda y mostrarlos de forma muy visual. Para mostrar como recoge los datos se ha introducido el consumo total de la red y se enchufado en la alargadera donde se encuentra conectado el dispositivo medidor un aire acondicionado durante 5 horas aproximadamente. En la siguiente gráfica de barras se observa la medición realizada por horas.

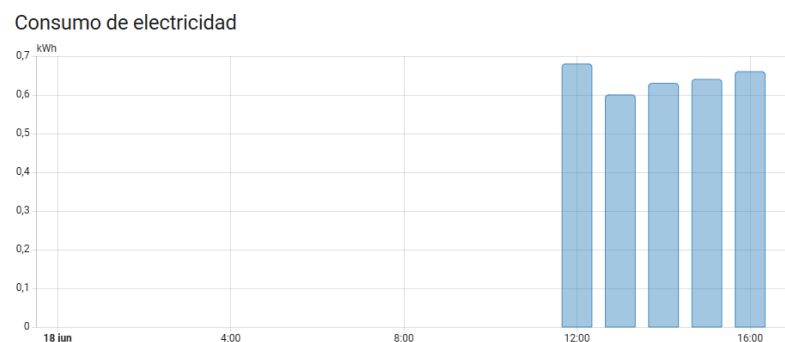


Figura 4.5 Gráfica consumo energético de la red.

4.2 Alternativa 2

Mediante la primera alternativa explicada se ha conseguido realizar la comunicación de una forma directa y únicamente a través del protocolo Modbus. En esta segunda alternativa se ha optado por el uso de un dispositivo intermedio, el cual se comunica con el medidor por comunicación Modbus, recibe los datos, y finalmente los envía por señal Wifi al Home Assistant Green, consiguiendo así obtener los datos de forma inalámbrica. El dispositivo intermedio es la placa ESP32 DevKit V1, el cual se ha elegido por la posibilidad de configuración en Home Assistant OS gracias al complemento ESPHome Device Builder.

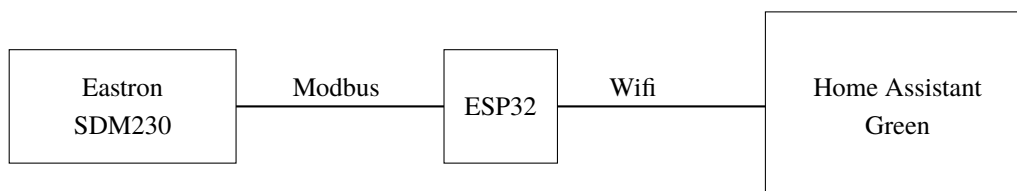


Figura 4.6 Esquema de dispositivos alternativa 2.

4.2.1 ESP32 DevKit V1

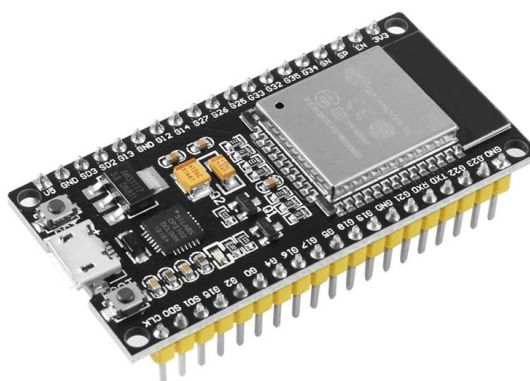


Figura 4.7 ESP32 DevKit V1. Fuente: Internet.

Este dispositivo es una placa de desarrollo, que usa como módulo de placa un ESP-WROOM-32, que tiene como base el chip ESP32. Permite usar protocolos de comunicación inalámbrica como puede ser Wifi o Bluetooth, característica por la que se ha escogido, ya que va a permitir obtener los datos en Home Assistant sin tener que realizar un cableado que vaya del cuadro eléctrico hasta donde este situado el Home Assistant Green. La placa dispone de 30 pines, con diferentes funcionalidades, como pines de alimentación (VIN y 3V3), tierra (GND), GPIO (General Purpose Input/Output), pines analógicos (ADC), PWM (Pulse Width Modulation), SPI (Serial Peripheral Interface), I2C (Inter-Integrated Circuit), pines de táctiles (Touch) o pines de comunicación serial (Tx y Rx). [2]

En cuanto a las opciones para la alimentación, esta placa puede ser alimentada directamente por puerto micro-USB o usando una fuente de alimentación de 5V o 3.3V. Para las pruebas de funcionamiento se alimenta por el puerto micro-USB.

Como se puede ver en la Figura 4.8 un problema de la placa ESP32 es que no tiene entradas para RS485, sino que tiene pines para comunicación serial en la que hay un pin de transmisión de datos denominado “Tx” y otro de recepción “Rx”. Para solucionar este problema se usa el módulo MAX485, capaz de transformar señal serie a RS485.

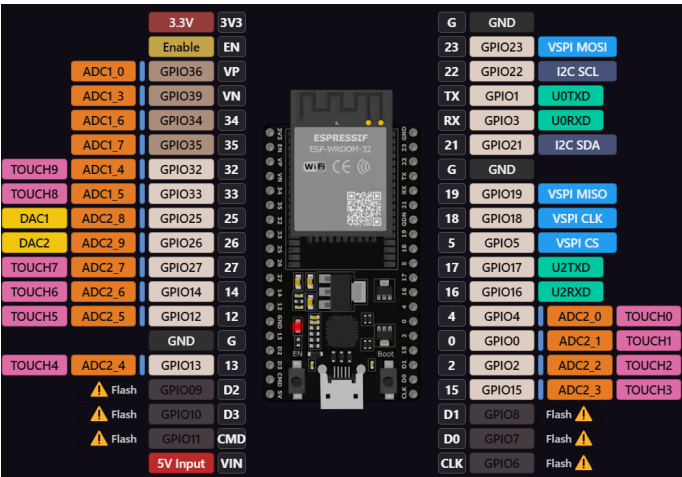


Figura 4.8 Pines ESP32. Fuente: Internet.

4.2.2 Módulo MAX485



Figura 4.9 Módulo MAX485. Fuente: Internet.

El módulo MAX485 es un transceptor de línea RS485 utilizado para implementar comunicaciones RS485 en microcontroladores como el ESP32 explicado en el apartado anterior. Es necesario su uso para transformar la señal serie emitida por el ESP32 a RS485 y viceversa. En la siguiente tabla se recogen los pines de este dispositivo.

Tabla 4.3 Pines MAX485.

Nombre del pin	Función
Vcc	Alimentación
GND	Tierra
A	Entrada señal A RS485
B	Entrada señal B RS485
RO (Receiver output)	Pin por el que se recibe información
RE (Receiver output enable)	Habilita la recepción de información
DE (Driver output enable)	Habilita la transmisión de información
DI (Driver input)	Pin por el que se transmite información

Dependiendo de la conexión de los pines RO, DI, RE y DE la comunicación que se establece podrá ser de diferente forma. Como se verá en las siguientes imágenes se pueden conseguir conexiones en las que el ESP32 funcione como transmisor de datos, receptor de datos y ambas.

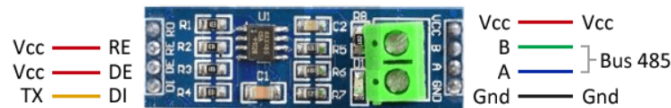


Figura 4.10 Conexión para ESP32 receptor.



Figura 4.11 Conexión para ESP32 transmisor.



Figura 4.12 Conexión para ESP32 receptor-transmisor.

4.2.3 Conexionado

Una vez se conocen todos los dispositivos que se van a usar y como funcionan se ha adaptado la Figura 4.6 para poder ver como es realmente la conexión entre los dispositivos de esta alternativa.

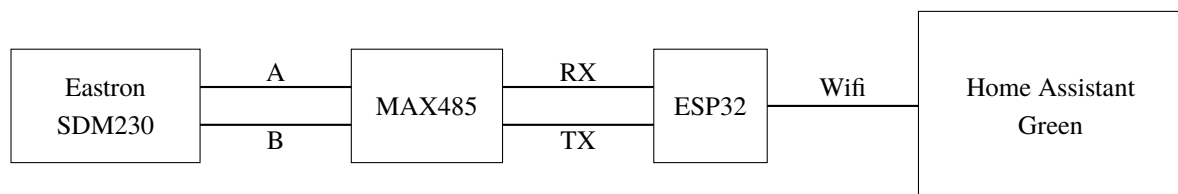


Figura 4.13 Conexión alternativa 1.

Debido a que en esta aplicación se necesita que el ESP32 transmita y reciba datos, la conexión usada entre ESP32 y MAX485 tiene que ser la indicada en la Figura 4.12. Se muestra a continuación un esquema del conexionado entre ESP32 y MAX485 indicando los pines que es necesario conectar entre ambos dispositivos.

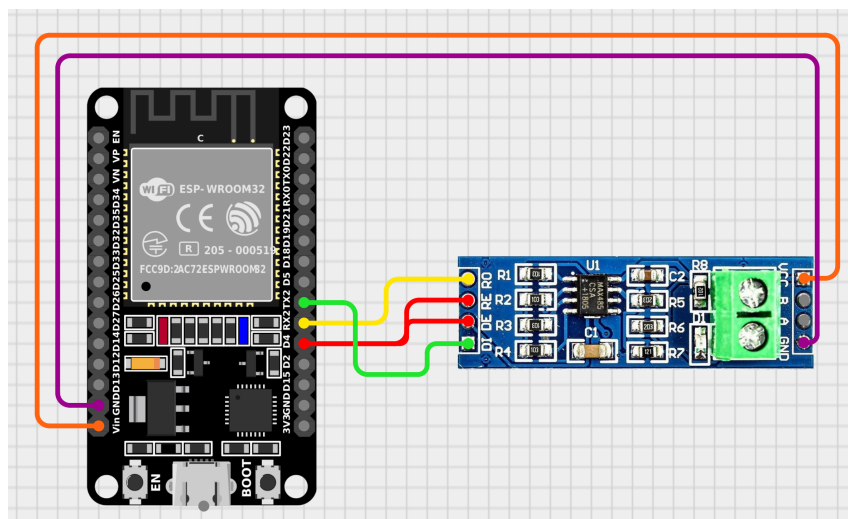


Figura 4.14 Conexión ESP32-MAX485.

Para conectar los pines entre las placas se han usado cables Dupont hembra-hembra como los que se pueden ver en la siguiente imagen junto a la conexión con todos los dispositivos reales.

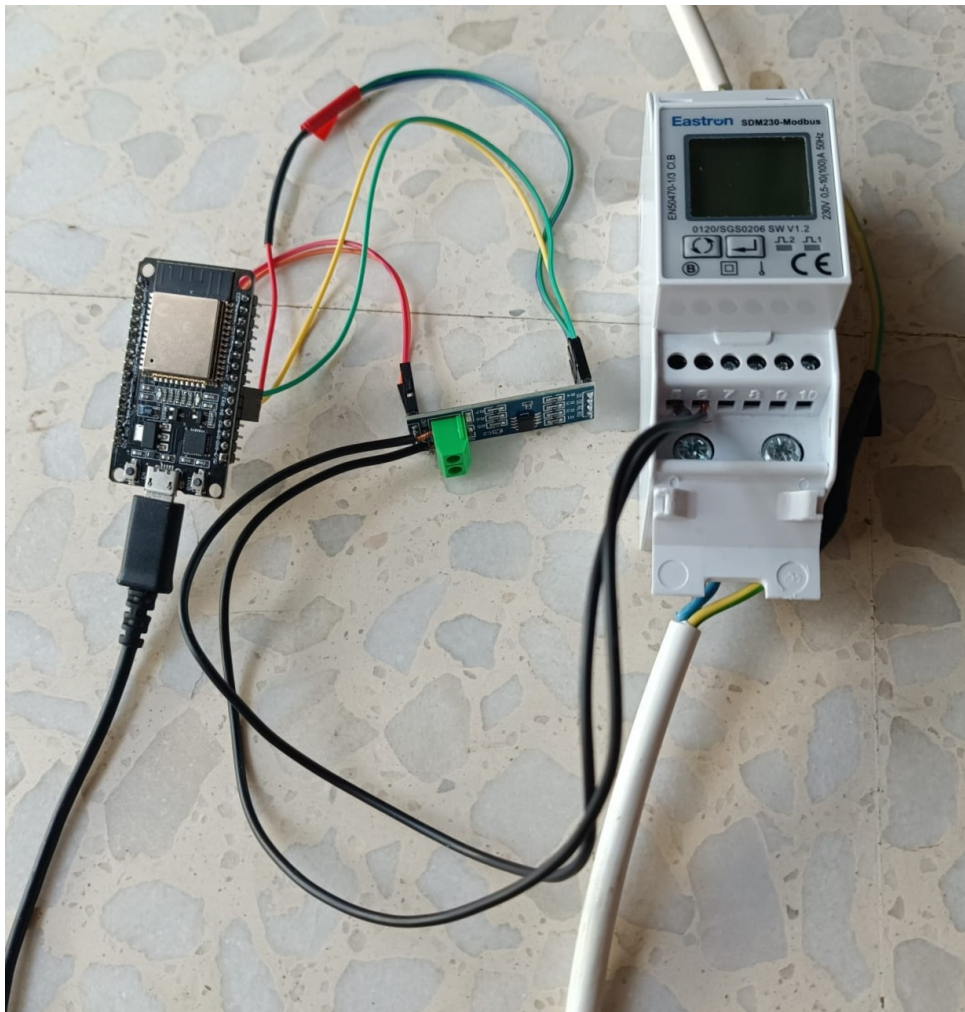


Figura 4.15 Conexión real alternativa 2.

4.2.4 Programación

Como se ha mencionado previamente, el proceso de programación y configuración se va a realizar en el complemento ESPHome Device Builder, el cual permite programar microcontroladores ESP para integrarlos en Home Assistant. El complemento funciona creando un dispositivo virtual que se configura para ser un reflejo del microcontrolador. En ese dispositivo virtual se realiza la programación que se desea hacer y finalmente se carga el archivo en el microcontrolador enviándolo por señal Wifi.

Una vez se entiende como funciona el complemento en esta sección se va a explicar primero como se configura un dispositivo virtual, ya que es una tarea algo tediosa, y luego como se programa el archivo .yaml de la misma manera que se ha hecho en la primera alternativa.

Cuando se inicia por primera vez el complemento es necesario añadir el nombre y la contraseña de la señal Wifi que se vaya a usar. Esto se hace para que después se incluya automáticamente la configuración de la señal Wifi en los dispositivos virtuales que se usen.

Para añadir una nueva placa se pulsa en el botón “New Device” situado en la zona inferior derecha, se le asigna un nombre y se selecciona el tipo de placa que sea, en este caso ESP32. Tras estos pasos aparecen cuatro opciones para poder descargar la configuración que se cargue posteriormente en el microcontrolador y le permita trabajar con Home Assistant.

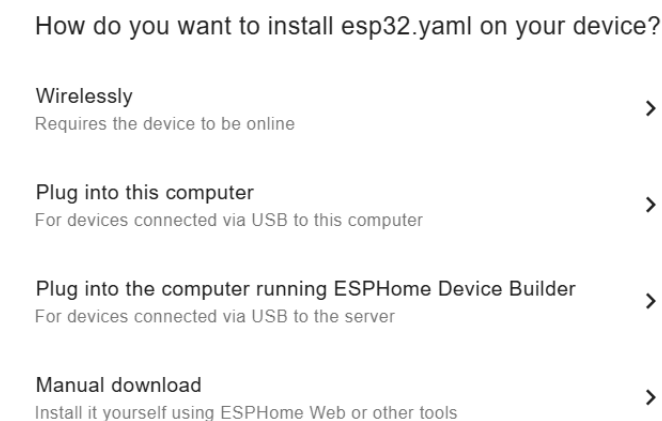


Figura 4.16 Opciones de instalación.

La primera vez que se haga se tiene que seleccionar la opción “Manual downlaod”, al pulsar en ella comienza un proceso que puede tardar varios minutos. Al terminar da a elegir entre dos versiones de descarga en la que se tiene seleccionar “Factory format (Previously Modern)”, esto descarga un archivo .lib. Una vez descargado se tiene que acceder a ESPHome Web en la siguiente url:

-<https://web.esphome.io/>

En esta página web se podrá cargar la configuración descargada del archivo .lib en el ESP32. Para ello se conecta la placa al ordenador con un cable USB que pueda ser usado para intercambio de datos, tras conectarlo se pulsa el botón “Connect” y se selecciona el puerto en el que se haya conectado la placa, de los que aparezcan disponibles. Tras hacer esto aparecerá la placa como conectada y se puede proceder a cargar la configuración. Esto se hace pulsando “install”, seleccionando el archivo previamente descargado y finalmente pulsando otra vez “install” mientras se mantiene apretado el botón “BOOT” de la placa ESP32. Una vez realizados estos pasos comenzará la instalación del archivo que podrá unos minutos, al finalizar la placa ESP32 ya estará lista para ser programada con ESPHome y estará conectada por Wifi al dispositivo Home Assistant Green. Para comprobar que el proceso se ha realizado correctamente se debe acceder al complemento y asegurarse que el dispositivo virtual añadido aparezca como online.

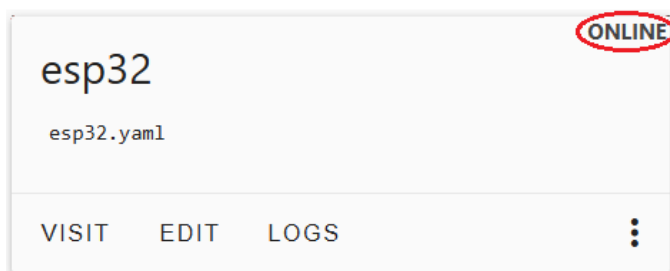


Figura 4.17 Dispositivo online.

Finalizada la configuración se puede empezar a programar el archivo .yaml, para acceder a este se pulsa en el botón “Edit” del dispositivo virtual y se comienza a editar el archivo justo por debajo del código que está escrito por defecto.

Al igual que en la primera alternativa es necesario comenzar por definir como va a ser la configuración Modbus. En este caso se realiza en tres pasos, primero se define la comunicación entre el ESP32 y el módulo MAX485 con la orden “uart”, indicando características de las tramas y los pines Rx y Tx que se vayan a usar. Después con la orden “modbus” e incluyendo las características anteriores se define el pin del ESP32 que habilita la comunicación receptor-transmisor con el MAX485. Por último se especifica la dirección del dispositivo que se pretenda monitorear y alguna característica más mediante la orden “modbus_controller”.

Tabla 4.4 Ordenes para configuración de comunicación Modbus en ESPHome.**UART**

id	Identificación de la configuración que esté siendo definida
tx_pin	Pin por el que se envían los datos del ESP32 al MAX485
rx_pin	Pin por el que se reciben los datos en el ESP32
baud_rate	Velocidad de envío de datos en bits/s
stop_bits	Número de bits de parada
data_bits	Número de bits de datos por byte
parity	Tipo de paridad usada

Modbus

id	Identificación de la configuración que esté siendo definida
uart_id	id de la configuración uart previamente definida
flow_control_pin	Pin del ESP32 que habilita la comunicación receptor-transmisor

Modbus_Controller

id	Identificación de la configuración que esté siendo definida
modbus_id	id de la configuración modbus previamente definida
address	Dirección del dispositivo a monitorear
update_interval	Tiempo entre toma de medidas en segundos

Código 4.3 Programación de la configuración Modbus en esp32.

```

uart:
  id: modbus_uart
  tx_pin: GPIO17
  rx_pin: GPIO16
  baud_rate: 9600
  stop_bits: 1
  data_bits: 8
  parity: NONE

modbus:
  - id: modbus1
    uart_id: modbus_uart
    flow_control_pin: GPIO4

modbus_controller:
  - id: modbus_cont
    modbus_id: modbus1
    address: 3
    update_interval: 5s

```

Una vez definida la comunicación se pasa a la obtención de las medidas. en ESPHome la configuración de sensores es parecido a como se hace en File Editor pero no exactamente igual en este caso se define la entidad como “sensor”, se le añade el id de la entidad “modbus_controller” previamente definida y seguidamente se le añaden sus características con las ordenes de la siguiente tabla.

Tabla 4.5 Ordenes para configuración de sensores en ESPHome.

platform	Tipo de tecnología que usa el sensor
modbus_controller_id	id de la configuración modbus previamente definida
register_type	Indica la zona de memoria del registro
address	Número del registro de la medida a consultar
device_class	Tipo de medida
accuracy_decimals	Número de decimales de la medida
name	Nombre asignado a la medida
unit_of_measurement	Unidades del registro medido
value_type	Formato de la variable

Código 4.4 Programación de sensor en esp32.

```

sensor:
  - platform: modbus_controller
    modbus_controller_id: modbus_cont
    register_type: read
    address: 0
    device_class: voltage
    accuracy_decimals: 2
    name: Tension
    unit_of_measurement: V
    value_type: FP32
    state_class: measurement

```

Una vez editado el código para que se apliquen los cambios se pulsa en “Save” para guardarlo y seguidamente en “Install”, aparecerá las misma opciones que en la Figura 4.16 pero esta vez al estar ya configurado se pueden aplicar los cambios por señal Wifi pulsando en “Wireless”.

Cuando se termine de instalar se podrá ver en la ventana resumen las medidas definidas por el código, de la misma forma que en la alternativa 1.

**Figura 4.18** Registro de medidas.

4.3 Comparación de alternativas

Ambas alternativas permiten alcanzar con éxito el objetivo de tomar datos del medidor Easton y visualizarlos en Home Assistant de igual forma. La Alternativa 1 destaca por su simplicidad en la programación y estabilidad al no depender de una señal Wifi que pueda tener problemas. Esta opción ideal para instalaciones en las que no haya problema en realizar un cableado largo entre ambos dispositivos. En cambio, la Alternativa 2 aporta una mayor flexibilidad, al poder colocar los dispositivos sin preocuparse del cableado, siempre teniendo en cuenta que la señal no sufra interferencias. Esta alternativa es mejor para viviendas donde el Home Assistant no se encuentre cerca del cuadro eléctrico. En cuanto a la inversión económica son muy similares, ya que todos los dispositivos auxiliares que se han mostrado son baratos. La elección entre ambas dependerá de las condiciones específicas de la instalación y las prioridades del usuario.

5 Ampliación de la instalación

Una vez se han presentado las dos alternativas para hacer llegar los datos de la instalación al Home Assistant, se ha visto oportuno seleccionar algunos dispositivos con comunicación Modbus que podrían ser usados para programar el funcionamiento de distintos dispositivos de la vivienda como sistemas de climatización, luces, persianas o toldos.

Los dispositivos que se usarán para esta ampliación son un módulo con 16 relés integrados y un termostato digital de pared para manejar fancoils. En la siguiente figura se puede ver un esquema de como quedaría la ampliación de la instalación.

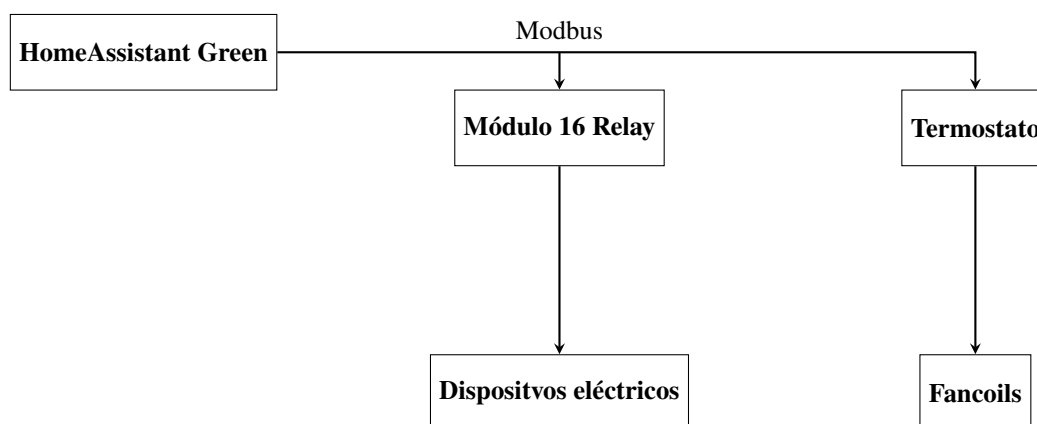


Figura 5.1 Esquema ampliación.

5.1 Modbus rtu 16-CH relay module

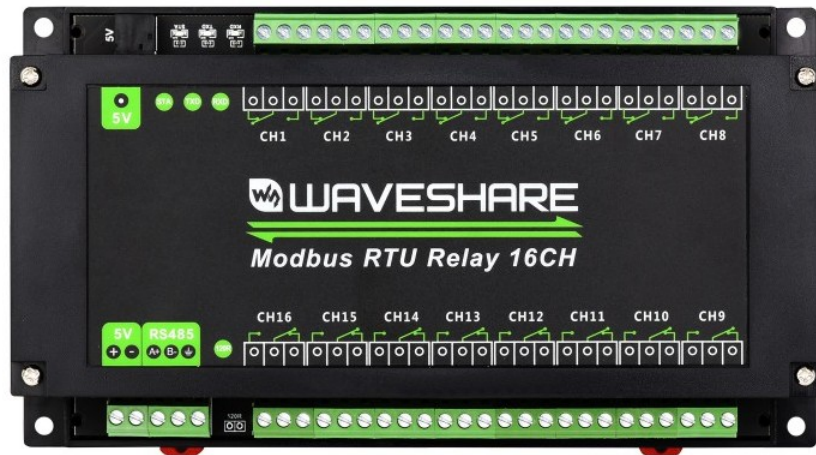


Figura 5.2 Dispositivo Modbus RTU 16-CH relay module.

Este equipo es un módulo conformado por 16 relés con conexión Modbus RTU fabricado por la empresa Waveshare electronics. Para poder cambiar de estado los relés será necesario enviar desde un dispositivo maestro una petición Modbus al registro correspondiente de cada relé, los cuales van desde el 0 al 15, correspondiendo el 0 al relé CH1 y el 15 al CH16. En cuanto a la alimentación, funciona con una tensión de 5V, la cual se puede introducir con un cable de alimentación por la parte superior izquierda, o también se puede alimentar por la bornera de tornillo en la zona inferior izquierda. Respecto a la entrada RS485 para comunicación Modbus, esta se sitúa en la zona inferior izquierda con sus respectivas entradas A, B y GND.

5.1.1 Conexionado

Como se ha podido ver en la Figura 5.2 cada canal del dispositivo representa un relé, que está formado por 3 puertos. El puerto central es el común, y es al que tiene que llegar la toma de corriente. Los puertos laterales son los de salida y van conectados a los dispositivos que se pretendan hacer funcionar. Dependiendo del estado del relé, deja pasar la corriente al puerto izquierdo o al derecho. Por defecto el estado del relé conecta el común con el puerto izquierdo, si se cambia el estado con una petición Modbus habilitando el puerto derecho, se notifica encendiéndose un led azul situado tras los puertos. En la siguiente figura se puede ver como realizar la conexión eléctrica para un dispositivo.

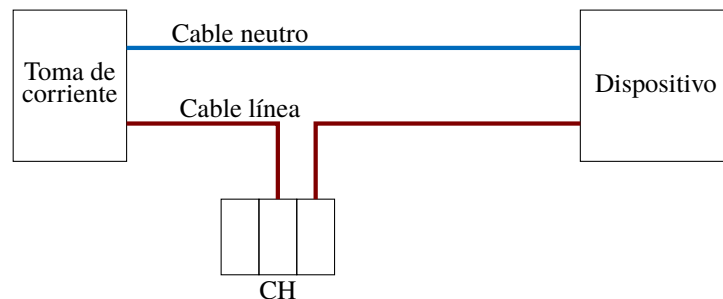


Figura 5.3 Conexionado para dispositivo de 2 estados.

El caso anterior serviría para un dispositivo que solo tenga 2 estados (encendido/apagado) como podría ser una bombilla, de forma que si el relé está conectado con el puerto izquierdo la luz está apagada, y si se cambia al derecho se enciende.

En otras aplicaciones de la vivienda como el motor de una persiana está conexión no es válida, ya que se necesita un estado más, debido a que el motor puede estar parado, moviéndose para abrir la persiana o para cerrarla. En este caso, se necesita hacer uso de dos de los relés siguiendo el conexionado de la siguiente figura. Para una mejor comprensión en la Tabla 5.1 se recoge el funcionamiento del motor dependiendo del estado de los relés.

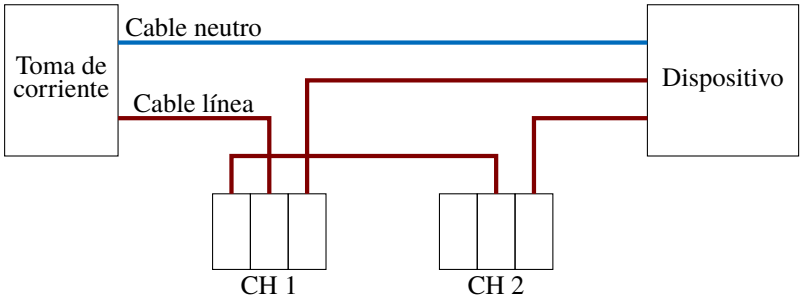


Figura 5.4 Conexionado para dispositivo de 3 estados.

Tabla 5.1 Funcionamiento del motor de una persiana según estados de los relés.

CH 1	CH 2	Motor
Izquierda	Izquierda	Apagado
Derecha	Izquierda	Subiendo
Derecha	Derecha	Subiendo
Izquierda	Derecha	Bajando

5.1.2 Comunicación Modbus

Como se explicó en el Capítulo 2 en la comunicación Modbus serie todos los dispositivos de una red Modbus serie tienen que tener las mismas características de comunicación. Por defecto, este dispositivo tiene las características iguales a la configuración aplicada al medidor Eastron, así que no hace falta cambiarlas.

Debido a que la dirección sí tiene que ser diferente para cada esclavo esta característica si es necesario modificarla, ya que por defecto es 1. Para cambiar la dirección se ha descargado en el ordenador la aplicación Modbus Poll software, que permite enviar tramas Modbus. Se ha conectado el módulo de relés al ordenador con el adaptador USB a RS485 y se le ha enviado la siguiente trama al dispositivo.

Código 5.1 Trama para cambiar la dirección.

00 06 40 00 00 02 1C 1A

Usando la dirección broadcast 00 se le envía el mensaje a todas las direcciones, la función 06 sirve para la modificación de registros analógicos, 40 00 es el registro donde se almacena la dirección del dispositivo, 00 02 es la nueva dirección y 1C 1A es el CRC.

Tras esta modificación las características de comunicación quedan de la siguiente forma.

Tabla 5.2 Características de comunicación.

Parámetro	Configuración
Tasa de Baudios	9600
Paridad	Sin paridad
Bits de Parada	1
Dirección	2

5.1.3 Programación

El código se realiza en el File Editor de Home Assistant OS como ya se hizo para la programación de la Alternativa 1. En primer lugar, se tiene que definir la comunicación Modbus con sus parámetros. Este paso es exactamente igual al que se programó en el Código 4.1 para la primera alternativa.

Una vez configurada la comunicación Modbus se define el tipo de datos que se va a medir, la cantidad de ellos y la dirección del dispositivo. En este caso, al ser relés, el tipo de dato en Home Assistant es “binary_sensors”. Como se ha hecho en anteriores explicaciones, en la Tabla 5.3 se muestra para que sirve cada comando de la configuración y en el Código 5.2 el valor que se le ha dado a cada comando.

Tabla 5.3 Ordenes para lectura de sensores binarios.

name	Nombre asignado a la medida
slave	Dirección asignada al dispositivo al que se pretenda mandar la petición
address	Número del primer registro que se pretenda consultar
slave_count	Número del último registro que se pretenda consultar
device_class	Tipo de medida del registro
input_type	Indica la zona de memoria del registro
scan_interval	Tiempo en segundos entre actualizaciones de los sensores
unique_id	Identificador único para los sensores definidos

Código 5.2 Mostrar estados de los relés.

```
binary_sensors:
  - name: R1e1e
    slave: 2
    address: 0
    slave_count: 15
    device_class: plug
    input_type: coil
    scan_interval: 1
    unique_id: reles_16ch
```

Este código nos permite la visualización de los estados de los relés a través de la lectura de los registros. Para poder cambiar los estados es necesario definir uno a uno cada registro como un interruptor binario, los cuales se definen en Home Assistant como “lights”. En el Código 5.3 se definen solo los dos primeros relés para ejemplificar como se debe hacer.

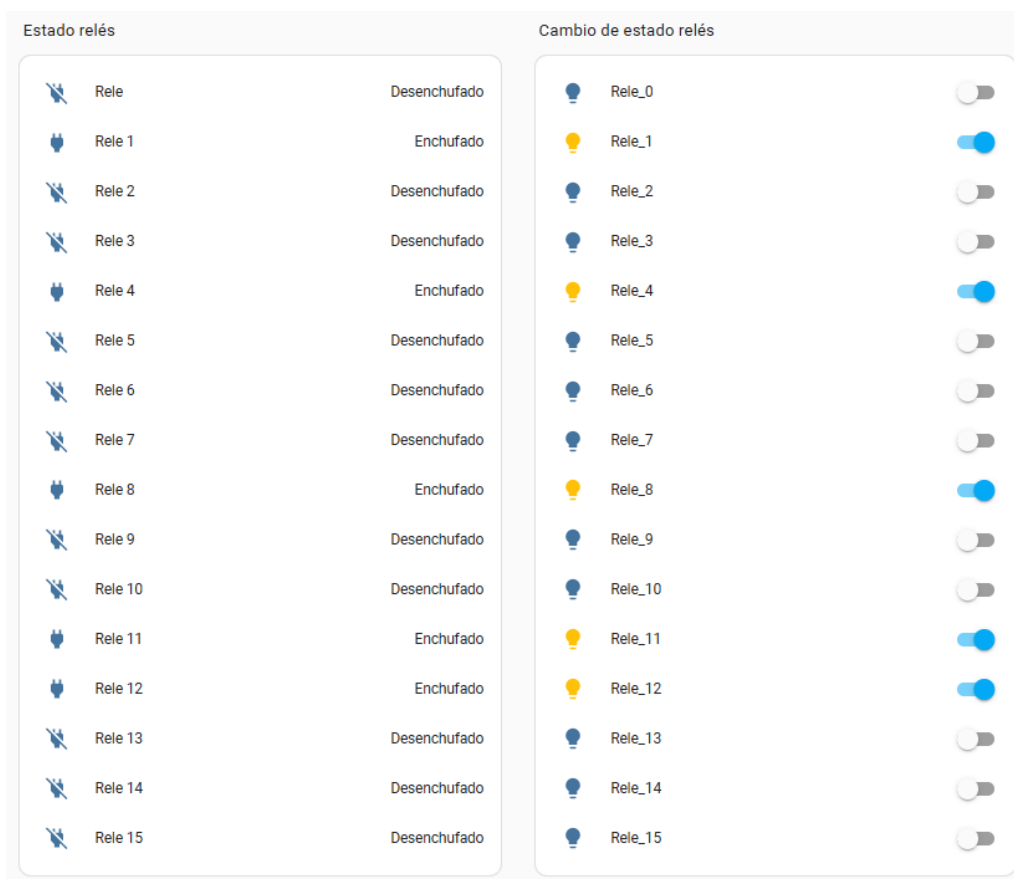
Tabla 5.4 Ordenes para modificar sensores binarios.

name	Nombre asignado a la medida
address	Número registro que se pretenda modificar
write_type	Indica la zona de memoria del registro
slave	Dirección asignada al dispositivo al que se pretenda mandar la petición

Código 5.3 Cambiar estados de relés.

```
lights:
- name: "Rele_0"
  address: 0
  write_type: coil
  slave: 2
- name: "Rele_1"
  address: 1
  write_type: coil
  slave: 2
```

Gracias a esta programación, la cual se encuentra completa en la Sección A.3 del Apéndice A, se pueden ver y modificar los estados de los relés. En la siguiente imagen se puede ver la ventana resumen en la que se ha creado una zona para ver los estados de cada relé en la parte izquierda y en la derecha los interruptores para cambiar de estado cada relé.

**Figura 5.5** Control de relés en Home Assistant OS.

5.2 Termostato CX3-S4DM



Figura 5.6 Dispositivo Termostato CX3-S4DM. *Fuente: Internet.*

Este dispositivo es un termostato de pared usado para modificar la configuración de una instalación de fancoils, válido para instalaciones de 2 o 4 tubos. Tiene posibilidad de control remoto mediante comunicación Modbus a través de RS485, lo que lo hace un dispositivo idóneo para el trabajo. En la siguiente tabla se recogen algunas de sus características.

Tabla 5.5 Características Home Assistant Green.

Físicas	
Ancho	86 mm
Alto	86 mm
Profundidad	30 mm
Eléctricas	
Alimentación	220 V $\pm$ 10 %
Rango de ajuste de temperatura	5 - 35 °C
Resolución de temperatura	± 0.5 °C
Precisión del control de temperatura	± 1 °C
Consumo	<2 W

5.2.1 Conexionado

La conexión necesaria para la instalación se puede observar en la Figura 5.7 mediante un esquema sacado de la Sección B.3 del Apéndice B. Se precisan de algunas indicaciones para una mejor comprensión.

- Dependiendo del tipo de instalación (2 o 4 tubos) se realiza una conexión diferente. En el caso de ser de 2 tubos, V1 es la válvula de frío/calor. Si es de 4 tubos V1 es la válvula de frío y V2 la de calor.
- S1 es una entrada por si se desea conectar un interruptor al dispositivo que lo encienda o apague.
- S2 es una entrada para poder añadir un sensor externo al dispositivo.
- A y B son las entradas de la comunicación Modbus.

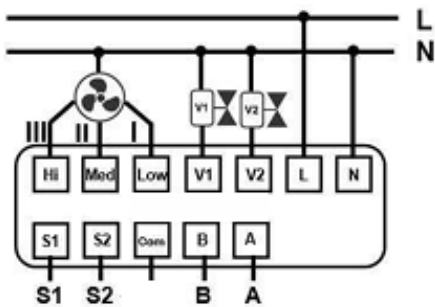


Figura 5.7 Conexionado Termostato CX3-S4DM.

5.2.2 Comunicación Modbus

Como se ha visto de forma repetida a lo largo del trabajo, cada dispositivo tiene unas opciones de configuración, pero se tienen que configurar de forma que todo los dispositivos del bus tengan la misma configuración. En la siguiente tabla se recogen las opciones y la elección de configuración para este dispositivo. Se ha escogido la dirección 3 para que no coincida con ninguna de las anteriores.

A diferencia del dispositivo anterior, en este se pueden cambiar los parámetros directamente desde el display del dispositivo sin tener que hacerlo mediante el envío de tramas Modbus. Antes de empezar a usarlo se debe configurar ya que las características que se van a usar no son las que configuradas por defecto.

Tabla 5.6 Características de comunicación.

Parámetro	Opciones	Elección
Tasa de Baudios	4800/9600/19200	9600
Paridad	Par/Impar/Ninguna	Ninguna
Bits de Parada	1	1
Dirección	1-32	3

5.2.3 Programación

Tras revisar los diferentes registros de este dispositivo, recogidos en el ANEXO, se ha considerado que los datos más interesantes para posteriormente realizar automatizaciones son los de la Figura 5.7. En la tabla también se añade el registro de cada dato y el valor que se debe escribir en el registro dependiendo de la acción que se dee realizar.

Tabla 5.7 Características de comunicación.

Parámetro	Registro	Valor a escribir
Encender Termostato	2	01
Apagar Termostato	2	00
Temperatura de funcionamiento	3	5 - 35
Leer Temperatura ambiente	9	-
Modo funcionamiento frío	1	1
Modo funcionamiento calor	1	2
Modo funcionamiento ventilación	1	3

En primer lugar, como en anteriores dispositivos se tiene que realizar la configuración Modbus RTU ya explicada previamente en el Código 4.1.

Para este tipo de dispositivos existe en Home Assistant la entidad “climates”, la cual permite añadir

características como las de la Tabla 5.8 para configurar de forma personalizada el termostato. En el Código 5.4 se muestran ya los parámetros configurados gracias a las diferentes características mostradas en la Tabla 5.5 y Tabla 5.7.

Tabla 5.8 Ordenes para configuración de climatizadores.

name	Nombre asignado a la medida
slave	Dirección asignada al dispositivo al que se pretenda mandar las peticiones
address	Número de registro que contenga la medida de temperatura
temperature_unit	Unidades de medida de temperatura
precision	Número de decimales aplicados
max_temp	Temperatura máxima a la que se pueda configurar el termostato
min_temp	Temperatura mínima a la que se pueda configurar el termostato
data_type	Formato de los datos
input_type	Zona de memoria del registro
target_temp_register	Registro de la temperatura de funcionamiento
target_temp_write_registers	Habilita a target_temp_register
temp_step	Precisión del control de temperatura
hvac_mode_register	Indica comienzo de la defición de los modos del dispositivo
values	Indica comienzo de la defición de los valores a escribir en el registro para activar los modos
state_cool	Número a escribir para activar modo frío
state_heat	Número a escribir para activar modo calor
state_fan_only	Número a escribir para activar modo ventilación
hvac_onoff_register	Registro en el que se habilita el encendido/apagado del dispositivo
hvac_on_value	Valor a escribir en el registro para encender el dispositivo
hvac_off_value	Valor a escribir en el registro para apagar el dispositivo

Código 5.4 Cambiar estados de relés.

```
climates:
- name: Termostato
  slave: 3
  address: 9
  temperature_unit: C
  precision: 1
  max_temp: 35
  min_temp: 5
  data_type: float32
  input_type: holding
  target_temp_register: 3
  target_temp_write_registers: true
  temp_step: 1
  hvac_mode_register:
    address: 1
    values:
      state_cool: 1
      state_heat: 2
      state_fan_only: 3
  hvac_onoff_register: 2
  hvac_on_value: 1
  hvac_off_value: 0
```

Gracias a esta programación, se podrá comprobar la temperatura ambiente, modificar la temperatura de funcionamiento y cambiar los modos de funcionamiento. En la siguiente imágenes se pueden ver el display que se crea gracias a la entidad “climates”. A la izquierda se ve la medición de temperatura y como se puede variar la temperatura de funcionamiento dentro del rango establecido. Y en la imagen de la derecha como existe una pestaña para alternar el modo de funcionamiento o poder apagar el termostato. Al no poseer este dispositivo para poder hacer pruebas, la temperatura ambiente es 0, ya que no se está leyendo ningún registro.

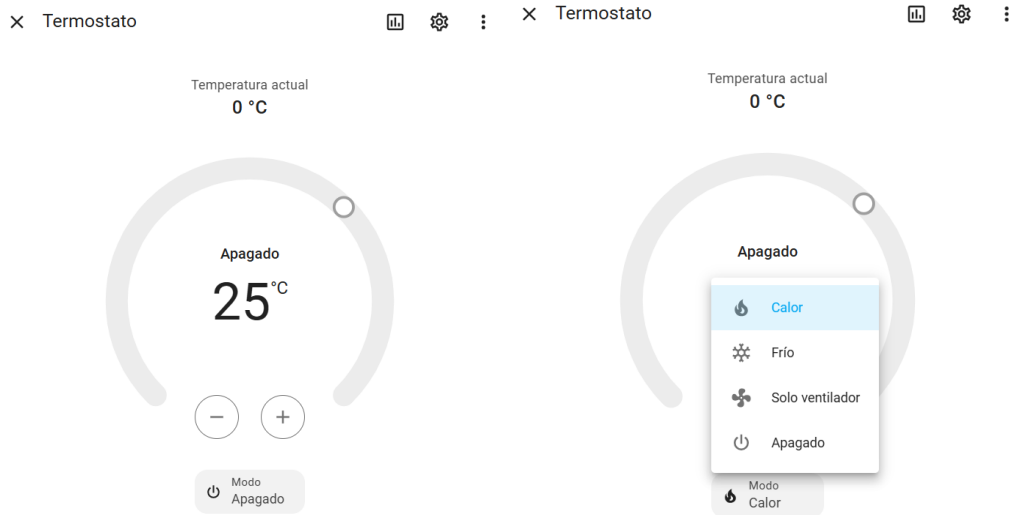


Figura 5.8

5.3 Propuesta de instalación

Una vez explicados como programar la comunicación y los conexiones de los dispositivos, en la siguiente figura se muestra una propuesta de lo que podría ser una instalación domótica con los dispositivos explicados, para una vivienda que permita el ahorro energético a través de la gestión de los excedentes de producción fotovoltaica.

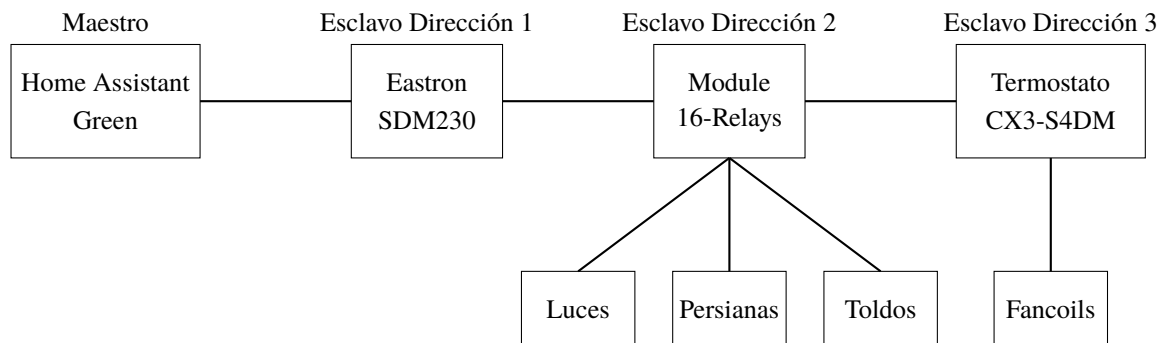


Figura 5.9 Esquema completo de la instalación domótica.

6 Conclusiones

Para culminar este trabajo, se presentan las conclusiones extraídas y algunas posibles ampliaciones del trabajo.

6.1 Objetivos alcanzados

A lo largo de este trabajo se ha desarrollado una instalación domótica basada en el protocolo de comunicaciones Modbus, con el objetivo de optimizar el consumo energético en viviendas con instalaciones fotovoltaicas de autoconsumo. Se ha demostrado cómo mediante la integración de dispositivos de medida y actuación, controlados a través de Home Assistant, es posible monitorizar y gestionar el consumo eléctrico.

A continuación, se exponen algunas conclusiones alcanzadas:

- Se ha realizado con éxito la comunicación entre el medidor Eastron SDM230 Modbus y el sistema Home Assistant, usando dos alternativas de conexión, con conexión directa mediante cableado y otra de forma inalámbrica. Obteniendo en ambas los mismos resultados, por lo que ambas son buenas opciones.
- Se ha ampliado la instalación mediante un módulo de 16 relés y un termostato CX3-S4DM, lo que ha permitido validar la capacidad del sistema para controlar de forma remota sistemas de climatización o iluminación.
- Se ha explicado la conexión con placas de desarrollo ESP, las cuales podrían cumplir otras funciones en una instalación como por ejemplo para crear sensores.
- Se ha demostrado como el protocolo de comunicación Modbus sigue siendo una opción perfectamente válida para entornos domésticos, entre la multitud de tecnologías para instalaciones domóticas que coexisten actualmente. Además su integración con Home Assistant es sencilla de implementar gracias al uso de archivos YAML en complementos como File Editor y ESPHome Builder.
- Se ha presentado un esquema de instalación domótica que podría aplicarse en entornos reales, sirviendo como base para futuras automatizaciones que busquen la eficiencia energética, mediante la gestión del consumo.

6.2 Posibles ampliaciones

Pese haber cumplido los objetivos que se propusieron en el comienzo del proyecto, siempre existen aspectos a mejorar que se podrían desarrollar en un trabajo futuro para complementar la instalación planteada. A continuación, se exponen algunos de los posibles trabajos para complementar.

- Realizar automatizaciones en Home Assistant que manejen los dispositivos ya programados para aumentar el autoconsumo y reducir el consumo de electricidad proveniente de la red eléctrica.

- Incorporar nuevos sensores y actuadores para recopilar más datos del entorno y poder diseñar automatizaciones que estén mejor adaptadas al uso real de la vivienda.
- Realizar integraciones con dispositivos que usen otras tecnologías domóticas. Esto facilitaría una instalación más flexible, permitiendo combinar distintos tipos de dispositivo.

A Anexo: Códigos

En este anexo se muestran todos los códigos que se han realizado para poder configurar los dispositivos descritos.

A.1 Eastron SDM230 Modbus: Alternativa 1

Código A.1 Toma de medidas del Eastron SDM230 Modbus en File Editor.

```
modbus:
  name: contador
  type: serial
  port: /dev/ttyUSB0
  baudrate: 9600
  bytesize: 8
  method: rtu
  parity: N
  stopbits: 1
  delay: 1
  sensors:
    - name: Tension
      address: 0
      slave: 1
      data_type: float32
      device_class: voltage
      input_type: input
      precision: 2
      unit_of_measurement: V
    - name: Potencia demandada
      address: 12
      slave: 1
      data_type: float32
      device_class: energy
      input_type: input
      precision: 2
      unit_of_measurement: W
    - name: Consumo total de la red
      address: 72
      slave: 1
      data_type: float32
      device_class: energy
```

```
    input_type: input
    precision: 2
    unit_of_measurement: kWh
  - name: Exportación total a la red
    address: 74
    slave: 1
    data_type: float32
    device_class: energy
    input_type: input
    precision: 2
    unit_of_measurement: kWh
```

A.2 Eastron SDM230 Modbus: Alternativa 2

Código A.2 Toma de medidas del Eastron SDM230 Modbus en ESPHome.

```
uart:
  id: modbus_uart
  tx_pin: GPIO17
  rx_pin: GPIO16
  baud_rate: 9600
  stop_bits: 1
  data_bits: 8
  parity: NONE

modbus:
  - uart_id: modbus_uart
    flow_control_pin: GPIO4
    id: modbus1

modbus_controller:
  - id: modbus_cont
    modbus_id: modbus1
    address: 1
    update_interval: 5s

sensor:
  - platform: modbus_controller
    modbus_controller_id: modbus_cont
    register_type: read
    address: 0
    device_class: voltage
    accuracy_decimals: 2
    name: Tension
    unit_of_measurement: V
    value_type: FP32

  - platform: modbus_controller
    modbus_controller_id: modbus_cont
    register_type: read
    address: 12
```

```

device_class: energy
accuracy_decimals: 2
name: Potencia demandada
unit_of_measurement: W
value_type: FP32

- platform: modbus_controller
  modbus_controller_id: modbus_cont
  register_type: read
  address: 72
  device_class: energy
  accuracy_decimals: 2
  name: Consumo total de la red
  unit_of_measurement: kWh
  value_type: FP32

- platform: modbus_controller
  modbus_controller_id: modbus_cont
  register_type: read
  address: 74
  device_class: energy
  accuracy_decimals: 2
  name: Exportacion total a la red
  unit_of_measurement: kWh
  value_type: FP32

```

A.3 Modbus rtu 16-CH relay module

Código A.3 Configuración del Modbus rtu 16-CH relay module.

```

modbus:
  name: contador
  type: serial
  port: /dev/ttyUSB0
  baudrate: 9600
  bytesize: 8
  method: rtu
  parity: N
  stopbits: 1
  delay: 1
  binary_sensors:
    - name: Rele
      address: 0
      device_class: plug
      input_type: coil
      scan_interval: 1
      slave: 2
      slave_count: 15
      unique_id: reles_16ch
  lights:
    - name: "Rele_0"
      address: 0

```

```
    write_type: coil
    slave: 2
- name: "Rele_1"
  address: 1
  write_type: coil
  slave: 2
- name: "Rele_2"
  address: 2
  write_type: coil
  slave: 2
- name: "Rele_3"
  address: 3
  write_type: coil
  slave: 2
- name: "Rele_4"
  address: 4
  write_type: coil
  slave: 2
- name: "Rele_5"
  address: 5
  write_type: coil
  slave: 2
- name: "Rele_6"
  address: 6
  write_type: coil
  slave: 2
- name: "Rele_7"
  address: 7
  write_type: coil
  slave: 2
- name: "Rele_8"
  address: 8
  write_type: coil
  slave: 2
- name: "Rele_9"
  address: 9
  write_type: coil
  slave: 2
- name: "Rele_10"
  address: 10
  write_type: coil
  slave: 2
- name: "Rele_11"
  address: 11
  write_type: coil
  slave: 2
- name: "Rele_12"
  address: 12
  write_type: coil
  slave: 2
- name: "Rele_13"
  address: 13
  write_type: coil
  slave: 2
- name: "Rele_14"
  address: 14
  write_type: coil
```

```
    slave: 2
  - name: "Rele_15"
    address: 15
    write_type: coil
    slave: 2
```

A.4 Termostato CX3-S4DM

Código A.4 Configuración del termostato CX3-S4DM.

```
modbus:
  name: contador
  type: serial
  port: /dev/ttyUSB0
  baudrate: 9600
  bytesize: 8
  method: rtu
  parity: N
  stopbits: 1
  delay: 1
  climates:
    - name: Termostato
      slave: 3
      address: 9
      temperature_unit: C
      precision: 1
      max_temp: 35
      min_temp: 5
      data_type: float32
      input_type: holding
      target_temp_register: 3
      target_temp_write_registers: true
      temp_step: 1
      hvac_mode_register:
        address: 1
        values:
          state_cool: 1
          state_heat: 2
          state_fan_only: 3
      hvac_onoff_register: 2
      hvac_on_value: 1
      hvac_off_value: 0
```


B Anexo: Hojas de datos

B.1 Eastron SDM230 Modbus

Eastron SDM230-Modbus User Manual



SDM230-Modbus

Single-Phase Two Module DIN rail Meters



- Measures kWh, Kvarh, KW, Kvar, KVA, PF, Hz, dmd, V, A, etc.
- Bi-directional measurement IMP & EXP
- Two pulse outputs
- RS485 Modbus
- Din rail mounting 35mm
- 100A direct connection
- Better than Class 1 / B accuracy

User Manual V1.4

2015

- 1 -

Application

The energy-meters “with a blue back-lighted LCD screen for prefect reading” are used to measure single-phase like residential, Utility and Industrial application. The unit measures and displays various important electrical parameters, and provide a communication port for remote reading and monitoring. Bi-directional energy measurement makes the unit a good choice for solar PV energy metering.

PART 1 Specification**General Specifications**

Voltage AC (Un)	230V
Voltage Range	176~276V AC
Base Current (Ib)	10A
Max. Current (Imax)	100A
Mini Current (Imin)	0.5A
Starting current	0.4% of Ib
Power consumption	<2W/10VA
Frequency	50/60Hz(±10%)
AC voltage withstand	4KV for 1 minute
Impulse voltage withstand	6KV-1.2uS waveform
Overcurrent withstand	30Imax for 0.01s
Pulse output rate	1000imp/kWh (default) 1000/100/10/1 imp/kWh/kVarh (configurable)
Display	LCD with blue backlit
Max. Reading	999999.9kWh

Accuracy

Voltage	0.5% of range maximum
Current	0.5% of nominal
Frequency	0.2% of mid-frequency
Power factor	1% of Unity
Active power	1% of range maximum
Reactive power	1% of range maximum
Apparent power	1% of range maximum
Active energy	Class 1 IEC62053-21 Class B EN50470-3
Reactive energy	1% of range maximum

Environment

Operating temperature	-25℃ to +55℃
Storage and transportation temperature	-40℃ to +70℃
Reference temperature	23℃ ± 2℃
Relative humidity	0 to 95%, non-condensing
Altitude	up to 2500m
Warm up time	10s
Installation category	CAT III
Mechanical Environment	M1
Electromagnetic environment	E2
Degree of pollution	2

Output**Pulse Output**

The meter provides two pulse outputs. Both pulse outputs are passive type.

Pulse output 1 is configurable. The pulse output can be set to generate pulses to represent total / import/export kWh or kVarh.

The pulse constant can be set to generate 1 pulse per: 0.001(default) / 0.01 / 0.1 / 1 kWh/kVarh.

Pulse width: 200/100/60ms

Pulse output 2 is non-configurable. It is fixed up with import kWh. The constant is 1000imp/kWh.

RS485 output for Modbus RTU

The meter provides a RS485 port for remote communication. Modbus RTU is the protocol applied. For Modbus RTU, the following RS485 communication parameters can be configured from the Set-up menu.

Baud rate: 1200, 2400, 4800, 9600

Parity: NONE/EVEN/ODD

Stop bits: 1 or 2

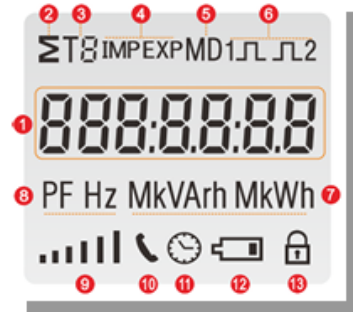
Modbus Address: 1 to 247

Mechanics

Din rail dimensions	36x99x63 (WxHxD) DIN 43880
Mounting	DIN rail 35mm
Sealing	IP51 (indoor)
Material	self-extinguishing UL94V-0

LCD display

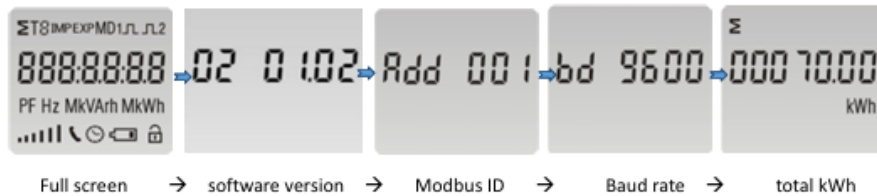
Item	Descriptions
1	7 digits used to display measured values or RTC
2	Total value
4	Import information, Export information
5	Max. Demand for Power or Current
6	Pulse output 1 and Pulse output 2
7	Measurement units
8	PF = power factor Hz = frequency
9	Bar display of Power
10	Communication indicator
11	Time information
12	Low battery warning
13	Lock symbol



PART 2 Operation

Initialization Display

When it is powered on, the meter will initialize and do self-checking.



Scroll display by Button

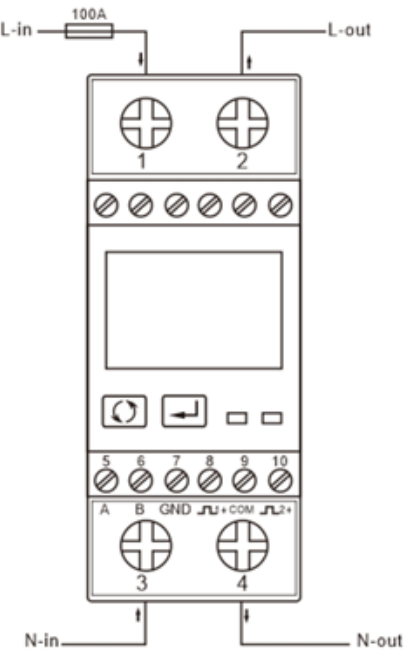
After initialization and self-checking program, the meter display the measured values. The default page is total kWh. If the user wants to check other information, he needs to press the scroll button on the front panel.

The display order by scroll button

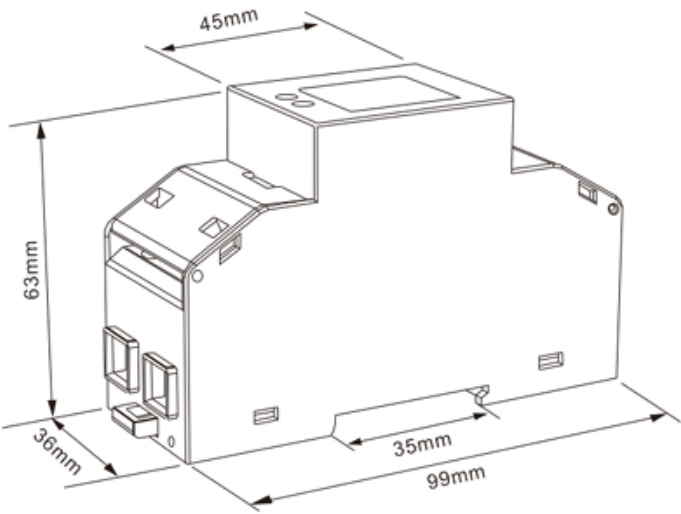


Total kWh → import kWh → export kWh → resettable kWh → total kWh → import kWh → export kWh → resettable kWh → Max. power demand → voltage → current → W → Var → VA → power factor → frequency → pulse constant → Modbus ID → baud rate → continuous running time.

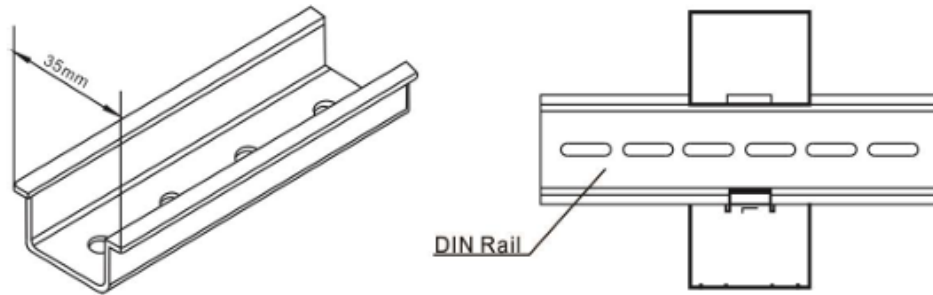
Wiring diagram



Dimensions



Installation



Eastron SDM230Modbus Protocol V1.2

1.Eastron SDM230Modbus Smart Meter Modbus Protocol Implementation V1.2

1.1 Modbus Protocol Overview

This section provides basic information for interfacing the Eastron Smart meter to a Modbus Protocol network. If background information or more details of the Eastron implementation is required, please refer to section 2 and 3 of this document.

Eastron offers the option of an RS485 communication facility for direct connection to SCADA or other communications systems using the Modbus Protocol RTU slave protocol. The Modbus Protocol establishes the format for the master's query by placing into it the device address, a function code defining the requested action, any data to be sent, and an error checking field. The slave's response message is also constructed using Modbus Protocol. It contains fields confirming the action taken, any data to be returned, and an error-checking field. If an error occurs in receipt of the message, SDM230Modbus will make no response. If the SDM230Modbus is unable to perform the requested action, it will construct an error message and send it as the response.

The electrical interface is 2-wire RS485, via 2 screw terminals. Connection should be made using twisted pair screened cable (Typically 22 gauge Belden 8761 or equivalent). All "A" and "B" connections are daisy chained together. Line topology may or may not require terminating loads depending on the type and length of cable used. Loop (ring) topology does not require any termination load. The impedance of the termination load should match the impedance of the cable and be at both ends of the line. The cable should be terminated at each end with a 120 ohm (0.25 Watt min.) resistor. A total maximum length of 1200 meters is allowed for the RS485 network. A maximum of 32 electrical nodes can be connected, including the controller. The address of each Eastron can be set to any value between 1 and 247. Broadcast mode (address 0) is supported.

The format for each byte in RTU mode is:

Coding System:	8-bit per byte
Data Format:	4 bytes (2 registers) per parameter. Floating point format (to IEEE 754) Most significant register first (Default). The default may be changed if required -See Holding Register "Register Order" parameter.
Error Check Field:	2 byte Cyclical Redundancy Check (CRC)
Framing:	1 start bit 8 data bits, least significant bit sent first 1 bit for even/odd parity (or no parity) 1 stop bit if parity is used; 1 or 2 bits if no parity

Data Coding

All data values in the SDM230Modbus smart meter are transferred as 32 bit IEEE754

Eastron SDM230Modbus Protocol V1.2

floating point numbers, (input and output) therefore each SDM230Modbus meter value is transferred using two Modbus Protocol registers. All register read requests and data write requests must specify an even number of registers. Attempts to read/write an odd number of registers prompt the SDM230Modbus smart meter to return a Modbus Protocol exception message. However, for compatibility with some SCADA systems, SDM230Modbus Smart meter will response to any single input or holding register read with an instrument type specific value.

The SDM230Modbus can transfer a maximum of 40 values in a single transaction; therefore the maximum number of registers requestable is 80. Exceeding this limit prompts the SDM230Modbus to generate an exception response.

Data transmission speed is selectable between 1200, 2400, 4800, 9600

1.2 Input register

Input registers are used to indicate the present values of the measured and calculated electrical quantities. Each parameter is held in two consecutive 16 bit register. The following table details the 3X register address, and the values of the address bytes within the message. A (*) in the column indicates that the parameter is valid for the particular wiring system. Any parameter with a cross(X) will return the value zero. Each parameter is held in the 3X registers. Modbus Protocol function code 04 is used to access all parameters.

For example, to request:	Amps 1	Start address=0006
		No. of registers =0002
	Amps 2	Start address=0008
		No. of registers=0002

Each request for data must be restricted to 40 parameters or less. Exceeding the 40 parameter limit will cause a Modbus Protocol exception code to be returned.

Address (Register)	SDM230-Modbus Input Register Parameter		Modbus Protocol Start Address Hex	
	Description	Units	Hi Byte	Lo Byte
30001	Line to neutral volts.	Volts	00	00
30007	Current.	Amps	00	06
30013	Active power.	Watts	00	0C
30019	Apparent power	VoltAmps	00	12
30025	Reactive power	VAr	00	18
30031	Power factor	None	00	1E
30037	Phase angle.	Degree	00	24
30071	Frequency	Hz	00	46

Easton SDM230Modbus Protocol V1.2

30073	Import active energy	kwh	00	48
30075	Export active energy	kwh	00	4A
30077	Import reactive energy	kvarh	00	4C
30079	Export reactive energy	kvarh	00	4E
30085	Total system power demand	W	00	54
30087	Maximum total system power demand	W	00	56
30089	Current system positive power demand	W	00	58
30091	Maximum system positive power demand	W	00	5A
30093	Current system reverse power demand	W	00	5C
30095	Maximum system reverse power demand	W	00	5E
30259	Current demand.	Amps	01	02
30265	Maximum current demand.	Amps	01	08
30343	Total active energy	kwh	01	56
30345	Total reactive energy	kvarh	01	58
30385	Current resettable total active energy	kwh	01	80
30387	Current resettable total reactive energy	kvarh	01	82

1.3 Modbus Protocol Holding Registers and Digital meter set up

Holding registers are used to store and display instrument configuration settings. All holding registers not listed in the table below should be considered as reserved for manufacturer use and no attempt should be made to modify their values.

The holding register parameters may be viewed or changed using the Modbus Protocol. Each parameter is held in two consecutive 4X registers. Modbus Protocol Function Code 03 is used to read the parameter and Function Code 16 is used to write. Write to only one parameter per message.

Address Register	Parameter	Modbus Protocol Start Address Hex		Valid range	Mode
		High Byte	Low Byte		
40013	Pulse output 1 Width	00	0C	Write relay on period in Milliseconds: 60, 100 or 200, default 100. Data Format: float (length: 4 byte)	r/w
40019	Network Parity	00	12	Write the network port parity/stop bits for MODBUS Protocol, where: 0 = One stop bit	r/w

B.2 Home Assistant Green



Home Assistant Green Datasheet

Product

Model No.	NC-GREEN-1175
Conformity	CE, FCC
Origin	Made in PRC

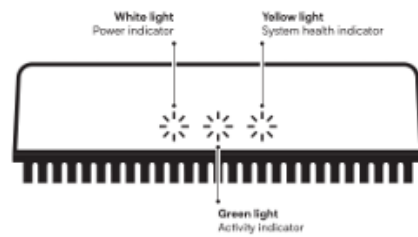
Product details

SoC	Rockchip RK3566 SoC with quad-core Arm Cortex-A55 CPU
Frequency	1.8 Ghz
Storage	32 GB eMMC flash drive
Memory	4 GB LPDDR4X
Environmental conditions for operation	Indoor use only 0 °C to 40 °C 32 °F to 104 °F Humidity: non-condensing Keep in dry, not excessively dusty environment as this can cause damage to the unit
Material	Enclosure • Polycarbonate plastic • Color: Semi-transparent

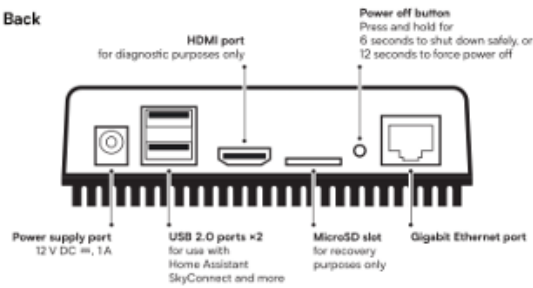


Home Assistant Green Datasheet

Front



Back



Interfaces

Status LEDs	• White (power indicator) • Green (activity indicator) • Yellow (system health indicator)
Power supply	DC barrel connector, 5.5 mm * 2.1 mm 12 V DC, 1 A
Battery	CR2032. Not included. Optional.
Energy consumption	Idle: ~1.7 W at 12 V Load: ~3 W at 12 V
USB	2x USB 2.0 Type-A Host ports, 5 V up to 2 A (combined)
Display	HDMI port, for diagnostic purposes only
External storage	MicroSD slot
Networking	Gigabit Ethernet port



Dimensions and weight, including accessories and packaging

Length	218 mm / 8.58 inches
Width	128 mm / 5.04 inches
Height	58 mm / 2.28 inches
Weight	655 g / 23.10 oz

Dimensions and weight

Length	112 mm / 4.41 inches
Width	112 mm / 4.41 inches
Height	32 mm / 1.26 inches
Weight	340 g / 12 oz

What's in the box

Home Assistant Green smart home hub	1x
Ethernet cable	1x
Universal power supply: 12 V DC, 1 A	
Power plug adapters (EU, US, UK)	1 each
Warranty and Safety Information	1x
Quick Start Guide	1x

B.3 Termostato CX3-S4DM

CX3-S Manual, montaje en superficie



Gracias por comprar nuestro producto. Su soporte y confianza nos han llevado a más de 25 años de éxito.

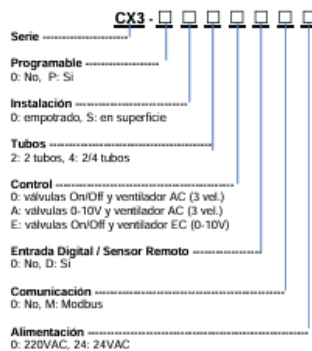
Los termostatos de Fan Coil, ampliamente usados en edificios residenciales, comerciales, de oficinas, hoteles, hospitales y otros usos industriales, controlan la temperatura ambiente con precisión, procurando un ambiente confortable con un consumo reducido de energía, que puede alcanzar hasta el 80%.

Gama:

En esta serie hay modelos para aplicación a:

- 2, 4 y 2/4 tubos
- Ventilador con motor AC (3-velocidades) o EC (0-10V)
- Válvulas ON/OFF o 0-10V
- Comunicación Modbus
- Contacto seco para Ventana, Puerta o Tarjetero de Hotel
- Sensor remote para la temperatura del aire o de tubería

Nomenclatura



Nota: los dígitos 0 pueden ser omitidos

Ejemplo:

CX3-PS4EDM significa: programable, montaje en superficie, 4 tubos, válvulas On/Off, motor EC del ventilador, contacto de entrada y sensor remoto, comunicación Modbus. Alimentación 220VAC

Especificaciones

1. Alimentación: 24VAC/220VAC $\pm$ 10%, 50/60 Hz
2. Rango de ajuste de la Temperatura: 5-35°C
3. Resolución de la Temperatura: $\pm$ 0.5°C
4. Precisión del Control de la Temperatura: $\pm$ 1°C
5. Tipos de ventilador: AC (3 velocidades) o EC (0-10V)
6. Carga de los contactos: 5A ohmica, 3A inductiva
7. Control de las Válvulas: ON/OFF o 0-10V
8. Sensor Interno: NTC100K
9. Sensor Remoto: NTC10K
10. Comunicación: Modbus 4800,9600,19200bps tasa de bauds
11. Contacto de entrada libre de tensión para Ventana, puerta o tarjetero de ocupación de hotel
12. Temperatura de almacenamiento: -20°C-60°C
13. Temperatura de servicio: 0°-50°C
14. Humedad de servicio: 5-95%
15. Consumo de potencia: <2W
16. Distancia entre taladros de fijación: 60mm
17. Grado de protección: IP 40
18. Retroiluminación: blanca
19. Color de la carcasa: blanco RAL9010
20. Tipo de instalación standard: superficie (S)
21. Dimensiones: 86x86x30mm
22. Condiciones de storage: IEC 60721-3-1, Class 1K3
23. Condiciones de transporte: IEC 60721-3-2, Class 2K3
24. Condiciones de servicio: IEC 60721-3-3, Class 3K5

Guía para el usuario:

On/Off: Presione para conmutar On/Off

Ajuste de la Temperatura de Consigna

Presione "A" para aumentarla, "V" para reducirla

Modo: Presione "M" para cambiar el modo entre Frio , Calor y Ventilación . El modo se confirmará al presionar otro botón o automáticamente tras 6 segundos.

Ventilador: Seleccione las velocidades o Auto , presionando . En Auto la velocidad será Baja si la diferencia entre ambiente y consigna excede de 1°C, será Media si excede de 2°C y Alta si la diferencia es superior a 3°C.

Ajuste del voltaje de salida para ventiladores 0-10V:

Conmute a Off el termostato y presione "M" y "V" 5 segundos para entrar en el menú de ajuste de parámetros. Presione "M" o para cambiar el parámetro, y presione "V" o "A" para cambiar el valor: (40°10%-4V)

Nota: si el valor es 40, el voltaje de salida es 4V.

Par.01: Baja vel.; (Rango de 11-40 Por defecto =40)

Par.02: Vel. Media; (Rango de 41-70 Por defecto=70)

Par.03: Alta vel.; (Rango de 71-99 Por defecto= 90)

Pulse o espere 30s para salir y guardar todos los ajustes.

Retroiluminación: Se activará al tocar cualquier botón. Se desactivará tras 6s después del último toque.

Bloqueo del teclado: (SOLO posible si el Par. 11 se ajusta a 1/2/3): Presione "V" y "A" por 3 segundos para bloquear los botones y presiónelos de nuevo igualmente para desbloquear.

Reloj (Solo CX3-PS):

Encienda el termostato y presione "M" por 3 segundos

1. Parpadearán las horas, presione "V" o "A" para ajustarla

2. De nuevo presione "M": parpadearán los minutos, presione "V" o "A" para ajustarlos

3. Presione "M" y parpadeará el día de la semana, presione "V" o "A" para ajustarlo

4. Presione "M" para confirmar el ajuste

Presione el botón de encendido o espere 30s para salir y guardar todos los ajustes

Temporización On/Off (Solo CX3-PS):

Repita los pasos 1-4 anteriores para On u Off del temporizador

1. La hora parpadea, presione "V" o "A" para ajustarla

2. Presione "M" para los minutos y ajustados igual que la hora

3. Presione "M". Parpadearán Timer Off y la hora, ajústela pulsando "V" o "A"

4. Presionando de nuevo "M", parpadearán los minutos, que se ajustan con "V" o "A"

5. Presione "M" o espere 6 seg. para confirmar los cambios. Pulse el botón o espere 30s para salir y guardar los ajustes.

Nota: La temporización actúa solo una vez y para el día en curso.

Programación (CX3PS solo):

Encienda el termostato, pulse "M" y por 3 segundos

1. Aparece No.1 (evento 1) y "MON TUE WED THU FRI"

2. Pulse "V" o "A" para ajustar la hora de comienzo

3. Pulse "M" y use "V" o "A" para ajustar el minuto de inicio

4. Pulse "M" y use "V" o "A" para ajustar la consigna

5. Pulse "M" para entrar en 2/3/4 y repita los pasos 2-4

6. Pulse "M" para editar "Sat & Sun" y repita los pasos 1-5

7. Pulse "M" o espere 6s para confirmar los ajustes anteriores.

8. Pulse o espere 30s para salir y conservar los cambios

Ajuste de Parámetros:

Apague el termostato y pulse "M" y "A" 5 seg. para entrar en el programa de ajuste de parámetros. Pulse "M" o para cambiar el parámetro y pulse "V" o "A" para cambiar su valor. Pulse o espere 30s para salir y conservar todos los ajustes

1	Tipo de equipo	0: Fan Coil 2-tubos (por defecto) 1: Fan Coil 4-tubos 2: Frio: FC, Calor: FC+ Suelo Radiante 3: Frio: FC, Calor: Suelo Radiante
2	Modo Auto	0: Manual (por defecto) 1: Auto
3	Sensor	1: Solo el interno (por defecto) 2: Solo externo 3: Sensor interno y externo
4	Control del Ventilador	0: Para alcanzar la consigna (p. defecto) 1: Continúa en marcha en confort
5	Estado de Desocupación	0: Corta todas las salidas 1: Reduce el confort de la consigna
6	Display de la Temperatura	0-Muestra temperatura ambiente (p. defecto) 1-Muestra la consigna
7	Offset de la Temperatura	-4°C - 4°C (por defecto = 0)
8	Diferencial	1° - 4°C (por defecto = 1)

9	Temp. cambio Auto a Calor en 2-tubos	21°-40°C (22°C p. defecto) en 2-tubos. Modo Auto solo con sensor remoto
10	Temp. cambio Auto a Frio en 2-tubos	10°-20°C (18°C por defecto) en 2-tubos. Modo Auto solo con sensor remoto a tubería
11	Bloqueo de Teclado	0: Todas las teclas libres (p. defecto) 1: Todos los botones bloqueados 2: Botón de On/Off bloqueado 3: Todos bloqueados menos el de On/Off
12	Estado al rearme (tras un corte)	0: Rearme en Off (defecto) 1: Vuelve al estado previo al corte 2: Se rearma en On
13	Contacto de entrada libre de tensión (tarjetero)	0: Con tarjeta insertada, S1 y S2 abiertos (por defecto contacto NC) 1: Con tarjeta insertada, S1 y S2 cerrados (contacto NO)
14	Protección heladas	0: Deshabilitada (por defecto) 1: Habilitada
15	Programable	1: No-programable 2: Temporización On/Off (por defecto) 3: Programable 4: Temporizable y Programable
16	Pre calentamiento	21°-50°C (38°C por defecto)
17	Consigna mínima	9°-18°C (5°C por defecto)
18	Consigna ECO Frio (Desocupación)	22°-32°C (28°C por defecto)
19	Consigna máxima	28°-35°C (35°C por defecto)
20	Consigna ECO Calor (Desocupación)	10°-21°C (18°C por defecto)
21	Retardo en Calor	6-4 minutos (0 por defecto)
22	Retardo Protección del Compresor	0-4 minutos (0 por defecto)
23	Fahrenheit/Celsius	F/C (Celsius por defecto)

Explicación de los parámetros:

1- **Tipo de equipo:** Control de Fan Coil de 2 / 4-tubos

2- **Modo Auto:** Se puede ajustar la unidad de 2 o 4 tubos para funcionar con cambio manual o automatico de modo

3- **Sensor:** 1: Solo sensor interno. 2: Solo sensor externo. 3: El sensor externo se adhiere a la tubería de agua para medir su temperatura y cambiar el modo Calor/Frio automáticamente. Esta función solo es posible para 2-tubos (Par. 1=0), cambio automático (Par. 2=1) y con sensor interno y externo (Par. 3=3). El modo será Frio si la temperatura del agua es inferior a un cierto valor ajustado por el parámetro 10 (18°C por defecto) y será Calor si la temperatura del agua es superior a un valor ajustado por el parámetro 9 (22°C por defecto).

4- **Control del Ventilador:** cuando se alcanza la temperatura de consigna se puede ajustar que el ventilador pare o que continúe funcionando.

5- **Estado desocupado:** al extraer la tarjeta del tarjetero, el estado de desocupación será: 0: Off, ninguna salida; 1: ECO con las consignas reducidas ajustadas por los parámetros 18 y 20 y el ventilador funcionará a baja velocidad

7- **Offset de la Temperatura:** se puede calibrar la lectura de temperatura en un rango de -4°C a 4°C

8- **Diferencial:** el frio o calor se activarán cuando la diferencia entre la temperatura ambiente y la de consigna sea $\geq$ que el diferencial. Por ejemplo, en modo Frio, si la consigna es 25°C y el diferencial =1°C, el arranque del frio se producirá a 26°C.

9 & 10- **Temperatura de Cambio Automático en 2-tubos:** Como se ha descrito con el parámetro 3 = 3, solo es posible si además 01 =0 (2-tubos) y 2=1 (modo Auto) al mismo tiempo. Con este ajuste el ventilador solo se activará si el agua está suficientemente fría o caliente, para evitar la sensación de frio en el modo Calor y ahorrar energía en el modo Frio si las temperaturas del agua no van a producir un intercambio confortable y eficiente

11- **Bloqueo del teclado:** se puede ajustar en 4 niveles diferentes, el símbolo aparecerá en la pantalla

12- **Estado tras Rearme:** Si falla la alimentación, al volver la corriente, el termostato puede retomar tres estados seleccionables:

0: se mantiene en Off. 1: recupera el estado que tenía previo al corte 2: conmuta a On automáticamente

13- Contacto de entrada. Función ECO

Un contacto seco (como p.e. el de un tarjetero de hotel) puede activar el modo ECO de ahorro de energía con aparición del como en la pantalla.

14. **Protección contra heladas:** En este modo (no disponible en aplicaciones solo frio) estando el termostato en Off si la temperatura desciende a 6°C el termostato se pondrá automáticamente en On, modo Calor hasta que la temperatura ambiente alcance 8°C

17 & 19. **Limites de temperatura de consigna:** se puede ajustar un mínimo en Frio y un máximo en Calor, para ahorrar energía

18 & 20: en el modo ECO de ahorro de energía (activada según el parámetro 13) el termostato funcionará con las consignas ajustadas para ahorro energético

21&22- Solo disponible si el parámetro 1-01(4-tubos) o para el modelo CX3(S)-AC, para compresor **Relé de Calor**: tras alcanzar la temperatura de confort, por razones de seguridad y de ahorro energético, el ventilador puede seguir funcionando entre 0-4mins.

23- Solo disponible para los modelos CX3-S4, CX3-S4D y CX3-S4DM

Nota: los parámetros 3,5,9,10,13,16,17,18,19,20 son solo para los termostatos con entrada digital /sensor remoto identificados con -D. El parámetro 15 es solo para los programables, identificados con una P.

Ajustes del Modbus (para termostatos con comunicación -M):

Apague el termostato y pulse sostenidamente "M" durante 5 seg. para entrar en los parámetros del Modbus

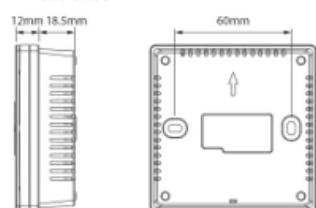
Par. 01 "Dirección": De 01-32, pulse "M" 3 veces para pasar al siguiente

Par. 02 "tasa baud", 01:4800, 02: 9600, 03:19,200.

Par. 03 "Paridad", 00: Sin paridad, 01: Impar, 02: Par

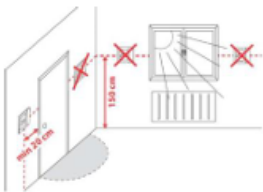
Pulsando el botón  o esperando 30s saldrá del menú guardando todos los ajustes

Dimensiones:



Montaje e instalación

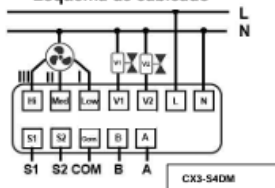
Instale el termostato en la pared pero no en nichos o estanterías, detrás de cortinas ni encima o cerca de fuentes de calor, aproximadamente por encima de 1.5 m del suelo.



El montaje se debe realizar en un espacio interior seco y limpio, no expuesto a golpes o corrientes de aire de focos fríos o calientes.

Antes de retirar la cubierta, desconecte la alimentación. El cableado se debe realizar cumpliendo la normativa.

Esquema de cableado



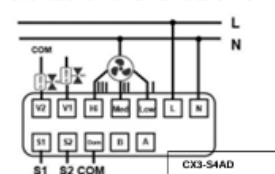
2-tubos: V1: válvula Frio/Calor

4-tubos: V1: Frio, V2: Calor

S1: contacto seco, S2: sensor remoto,

A/B: Modbus

Ventilador 3-vel. & Válvulas 0-10V

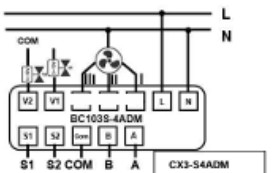


2-tubos: V1-COM: válvula 0-10V Frio/Calor

4-tubos: V1-COM: válvula 0-10V Frio

V2-COM: válvula 0-10V Calor

S1: contacto seco, S2: sensor remoto



2-tubos: V1-COM: 0-10V Frio/Calor

4-tubos: V1-COM: válvula 0-10V Frio

V2-COM: válvula 0-10V Calor

S1: contacto seco S2: sensor remoto,

A/B: Modbus

Retirada de deshecho



El dispositivo se considera equipo electrónico a efectos de cumplimiento de las directivas sobre retirada de desechos y no se puede eliminar como basura doméstica

--Se debe retirar por los canales adecuados (Puntos Limpios)

--Se deben cumplir las normas locales en vigor

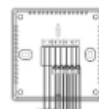
Instalación

1



Separe la cubierta apretando los pestillos que hay en su canto inferior con un destornillador plano

2

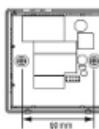


Para facilitar la instalación, el primer paso es cablear los hilos libres de tensión, luego los de bajo voltaje y a continuación los de alto voltaje. Compruebe que han sido correctamente borneados.

Para evitar contactos indeseados, utilice punteras para los hilos en el extremo de los cables



3



Montaje en superficie con taladros separados a 60mm. Asegúrese de que la tapa posterior está en la posición correcta según la flecha que lo indica

4



Fije la tapa frontal de la carcasa en el borde superior asegurándose de que los pin están correctamente posicionados y empujela sobre la carcasa hasta oír un click de encaje..

MODBUS PROTOCOL

Modbus Transport Protocols

Thermostats come with the option of Modbus communication. The support of Modbus communication allows simple integration of the thermostats with a building management system using standard Modbus serial communication.

Thermostats communicate as a Modbus RTU slave device over a serial RS-485 connection, allowing for the transfer of real-time data. RS-485 communication parameters such as baud rate, parity check and Modbus address can be adjusted.

These parameters are defined for each thermostat in the Parameter Settings Table. If required, adjust the settings to disable the Modbus connection. Connection to the RS-485 network is made via dedicated terminals on the back of the thermostat and marked A (+) and B (-). The following Modbus register types and formats are supported:

Serial number	Specification	Protocol Specification
1	Media	RS485
2	Baud Rate	19200BPS/9600 BPS /4800 BPS
3	Transmit Mode	RTU
4	Data Unit	Additional address + Function code+ Data 1.....N+ CRC High byte, CRC Low byte
5	Address	1-32
6	Function Code	3, 6, 16
7	Data Qty	<255
8	DATA	0-255
9	CRC check	CRC-16
10	Byte Format	11 digits: 1 start digit+8 data digit+ odd parity digit+ 1 stop digit
11	Check Method	CRC-16
12	0 Address	Broadcast Address
13	Port Definition	A (+), B (-)

03/06/16 Command message:

Function Code	Holding Register Address	Fan Coil	Data Description
03/06/16	1(0)	Fan mode	00: high speed 01: Medium speed 02: low speed 03: Auto Speed
03/06/16	2(1)	Mode	1: Cool 2: Heat 3: Ventilation
03/06/16	3(2)	Thermostat Status	00: Thermostat OFF 01: Thermostat ON 02: Frost Protect (read only)
03/06/16	4(3)	Set Room Temperature	Temperature (5~35℃)
03/06/16	5(4)	Timer On "Hour"	(0~24)
03/06/16	6(5)	Timer On "Minute"	(0~60)
03/06/16	7(6)	Timer Off "Hour"	(0~24)
03/06/16	8(7)	Timer Off "Minute"	(0~60)
03	9(8)	Communication check	Communicate between 0/1
03	10(9)	Room Temperature	0-50
03	11(10)	Output	Bit0: Cooling valve(4-pipe) 0: OFF, 1: ON Bit1: Fan low speed 0: OFF, 1: ON Bit2: Fan medium speed 0: OFF, 1: ON Bit3: Fan high speed 0: OFF, 1: ON Bit4: Heating valve (4-pipe) 0: OFF, 1: ON Bit5-7: Reserved
03	12(11)	Error Information	Bit0: Internal sensor error 0: OK, 1 Error Bit1: External sensor error 0: OK, 1 Error Bit2: Reserved Bit3: Reserved Bit4- Bit7: Reserved
03	13(12)	External sensor	Temperature Range (0~+99℃)
03	14(13)	Reserved	
03/06/16	15(14)	Reserved	
03/06/16	16(15)	Enable Frost Protection	00: OFF, 01: ON
03/06/16	17(16)	Programmable	01: Manual 02: Timer 03: Programmable 04: Timer+Programmable
03/06/16	18(17)	Thermostat status after Power Recover	0: OFF 1: Back to status before the power failure 2: ON
03/06/16	19(18)	Keypad lockout	00: Disable

			01: Lock all buttons 02: Lock ON/OFF button 03: Lock all buttons except ON/OFF button
03/06/16	20(19)	Temperature Display	00: Show room temperature 01: Show set Temperature
03/06/16	21(20)	Set Min Temperature	Set Range (5~18℃)
03/06/16	22(21)	Set Max Temperature	Set Range (20~40℃)
03/06/16	23(22)	Differential Temperature	Set Range (1~4℃)
03/06/16	24(23)	Sensor Selection	01: Internal 02: External 03: Internal & External
03/06/16	25(24)	Occupancy/dry contact close/open	00: Occupied when open-circuit 01: Occupied when short-circuit
03/06/16	26(25)	Unoccupied Status	00: cut off all outputs 01: entry ECO mode
03/06/16	27(26)	Unoccupied Heating set temperature range	Set Range (10~21℃)
03/06/16	28(27)	Unoccupied cooling set temperature range	Set Range (22~32℃)
03/06/16	29(28)	Fan operation after setting temperature is reached	00: fan off 01: fan on
03/06/16	30(29)	2/4 pipe selection	00: 2-pipe 01: 4-pipe
03/06/16	31(30)	Heat/Cool changeover	00: Manual 01: Auto 02: Reserved
03/06/16	32(31)	Preheat Temp. Setting	Set Range (21~50℃)
03/06/16	33(32)	Auto Cool Temp. Setting	Set Range (10~20℃)
03/06/16	34(33)	Auto Heat Temp. Setting	Set Range (21~40℃)

Temperature Value:

E.G: Temperature : 25.5℃, the date(255) : 00 FF

Temperature : 5.0℃, the date(50) : 00 32

Índice de Figuras

1.1	Evolución de la potencia instalada de generación renovable en MW [4]	1
1.2	Evolución del número total de instalaciones fotovoltaicas residenciales en España [3]	2
1.3	Curva diaria de producción fotovoltaica	2
1.4	Datos de producción y consumo fotovoltaico	3
1.5	Estadísticas de consumo y exportación de energía	3
1.6	Esquema de la instalación	4
2.1	Esquema Petición-respuesta	5
2.2	Esquema arquitectura modbus	6
2.3	Estructura de mensajes Modbus según la función [6]	7
2.4	Esquema modbus serie	8
2.5	Transmisión de Trama RTU	8
2.6	Transmisión de 1 byte	9
2.7	Requisitos de tiempo de Trama RTU	9
2.8	Líneas A y B	10
2.9	Conexión RS485	10
3.1	Dispositivo Eastron SDM230 Modbus	11
3.2	Localización y función de los puertos	13
3.3	Instalación en cuadro eléctrico	13
3.4	Conexiónado realizado	14
3.5	Home Assistant Green. <i>Fuente: Internet</i>	15
3.6	Puertos Home Assistant Green. <i>Fuente: Internet</i>	16
3.7	Esquema conexión Home Assistant Green	16
3.8	Interfaz de Home Assistant OS	17
4.1	Adaptador USB a RS485 <i>Fuente: Internet</i>	19
4.2	Conexión alternativa 1	20
4.3	Montaje real alternativa 1	20
4.4	Registro y evolución de medidas	22
4.5	Gráfica consumo energético de la red	22
4.6	Esquema de dispositivos alternativa 2	23
4.7	ESP32 DevKit V1. <i>Fuente: Internet</i>	23
4.8	Pines ESP32. <i>Fuente: Internet</i>	24
4.9	Módulo MAX485. <i>Fuente: Internet</i>	24
4.10	Conexión para ESP32 receptor	25
4.11	Conexión para ESP32 transmisor	25
4.12	Conexión para ESP32 receptor-transmisor	25
4.13	Conexión alternativa 1	25
4.14	Conexión ESP32-MAX485	25
4.15	Conexión real alternativa 2	26

4.16	Opciones de instalación	27
4.17	Dispositivo online	27
4.18	Registro de medidas	29
5.1	Esquema ampliación	31
5.2	Dispositivo Modbus RTU 16-CH relay module	32
5.3	Conexionado para dispositivo de 2 estados	32
5.4	Conexionado para dispositivo de 3 estados	33
5.5	Control de relés en Home Assistant OS	35
5.6	Dispositivo Termostato CX3-S4DM. <i>Fuente: Internet</i>	36
5.7	Conexionado Termostato CX3-S4DM	37
5.8		39
5.9	Esquema completo de la instalación domótica	39

Índice de Tablas

2.1	Características de las zonas de memoria	5
2.2	Códigos de función	7
2.3	Cálculo del código de función de excepción	7
2.4	Códigos de excepción	8
3.1	Características Easton SDM230 Modbus	12
3.2	Características de comunicación	12
3.3	Características de parámetros	12
3.4	Características Home Assistant Green	15
4.1	Características para configuración Modbus	21
4.2	Ordenes para configuración de sensores	21
4.3	Pines MAX485	24
4.4	Ordenes para configuración de comunicación Modbus en ESPHome	28
4.5	Ordenes para configuración de sensores en ESPHome	29
5.1	Funcionamiento del motor de una persiana según estados de los relés	33
5.2	Características de comunicación	34
5.3	Ordenes para lectura de sensores binarios	34
5.4	Ordenes para modificar sensores binarios	35
5.5	Características Home Assistant Green	36
5.6	Características de comunicación	37
5.7	Características de comunicación	37
5.8	Ordenes para configuración de climatizadores	38

Índice de Códigos

3.1	Ejemplo archivo YAML	17
4.1	Configuración Modbus	21
4.2	Configuración para lectura de entrada analógica	21
4.3	Programación de la configuración Modbus en esp32	28
4.4	Programación de sensor en esp32	29
5.1	Trama para cambiar la dirección	33
5.2	Mostrar estados de los relés	34
5.3	Cambiar estados de relés	35
5.4	Cambiar estados de relés	38
A.1	Toma de medidas del Eastron SDM230 Modbus en File Editor	43
A.2	Toma de medidas del Eastron SDM230 Modbus en ESPHome	44
A.3	Configuración del Modbus rtu 16-CH relay module	45
A.4	Configuración del termostato CX3-S4DM	47

Bibliografía

- [1] *Modbus*, wikipedia, <https://es.wikipedia.org/wiki/Modbus>, Consulta: 14 de mayo de 2025.
- [2] *Naylamp mechatronics*, <https://naylampmechatronics.com/>, Consulta: 8 de mayo de 2025.
- [3] APPA, *Iii informe anual del autoconsumo fotovoltaico (2024).*, 2025.
- [4] Red eléctrica, *Informe del sistema eléctrico: Informe resumen de energías renovables 2024.*, 2025.
- [5] Francisco Guzmán Navarro, *Domótica.*, Madrid: RA-MA Editorial, 2015.
- [6] B. M. Wilamowski and J. D. Irwin, *The industrial electronics handbook. fundamentals of industrial electronics*, 2nd ed., Boca Raton (Florida): CRC Press.