

4.3 Domótica

De nuevo vamos a tomar prestada una definición de la wikipedia:

Se llama domótica a los sistemas capaces de automatizar una vivienda o edificación de cualquier tipo, aportando servicios de gestión energética, seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, y cuyo control goza de cierta facilidad, desde dentro y fuera del hogar. Se podría definir como la integración de la tecnología en el diseño inteligente de un recinto cerrado.

<https://es.wikipedia.org/wiki/Dom%C3%B3tica>

En nuestro caso, dado que vamos a trabajar con una Raspberry Pi, vamos a ver una solución que yo considero perfecta para este ordenador. Se trata de Home Assistant pero antes de entrar en el meollo de la cuestión vamos a hablar de protocolos.

Protocolos de comunicación

En informática y telecomunicación, **un protocolo de comunicaciones es un sistema de reglas que permiten que dos o más entidades** (computadoras, teléfonos celulares, etc.) **de un sistema de comunicación se comuniquen entre ellas para transmitir información por medio de cualquier tipo de variación de una magnitud física.** Se trata de las reglas o el estándar que define la sintaxis, semántica y sincronización de la comunicación, así como también los posibles métodos de recuperación de errores. Los protocolos pueden ser implementados por hardware, por software, o por una combinación de ambos.

También se define como un conjunto de normas que permite la comunicación entre ordenadores, estableciendo la forma de identificación de estos en la red, la forma de transmisión de los datos y la forma en que la información debe procesarse.

Los sistemas de comunicación utilizan formatos bien definidos (protocolo) para intercambiar mensajes. Cada mensaje tiene un significado exacto destinado a obtener una respuesta de un rango de posibles respuestas predeterminadas para esa situación en particular. Normalmente, el comportamiento especificado es independiente de cómo se va a implementar. Los protocolos de comunicación tienen que estar acordados por las partes involucradas. Para llegar a dicho acuerdo, un protocolo puede ser desarrollado dentro de estándar técnico(...)

https://es.wikipedia.org/wiki/Protocolo_de_comunicaciones

En este punto puede ser que os estéis preguntando por qué os estoy contando qué son los protocolos de comunicaciones. En este apartado vamos a trabajar con distintos *gadget*, sensores, que pueden utilizar diferentes protocolos. Además, algunos de los sensores que voy a comentar funcionan con protocolos que la Raspberry Pi no soporta por lo que requieren de algún hardware adicional. Vamos a enumerar algunos de los protocolos que me parecen mas relevantes de cara a usar domótica en la Raspberry Pi:

- **Wi-Fi:** Soportado directamente por la raspberry pi.
- **Bluetooth:** Soportado directamente por la raspberry pi.
- **Zigbee:** No soportado por la raspberry pi directamente. Se trata de un protocolo de comunicación inalámbrica de bajo consumo. Ideal para dispositivos que requieren comunicaciones seguras con baja tasa de envío de datos y una gran vida útil de sus baterías. Necesita de un adaptador (USB) que conectaremos a la Raspberry Pi y que le dará a la Raspberry la posibilidad de comunicarse a través de este protocolo.
- **MQTT** (alias *mosquito*): es un protocolo de mensajería ligero con un consumo muy bajo de energía. Requiere de un *broker de mensajería*.

Zigbee

Mi adaptador zigbee-USB, un CC2531, lo adquirí por unos 6 € aquí <https://es.aliexpress.com/item/32971386349.html> requiere ser programado (*flasheado*). Se puede flashear "facilmente" con un dispositivo similar a <https://es.aliexpress.com/item/32976509073.html> que cuesta unos 8€ y estando obligados/as a utilizar Windows. También se puede flashear desde la Raspberry Pi utilizando este método https://www.zigbee2mqtt.io/guide/adapters/flashing/alternative_flashing_methods.html (yo no lo he probado)

Aideepen



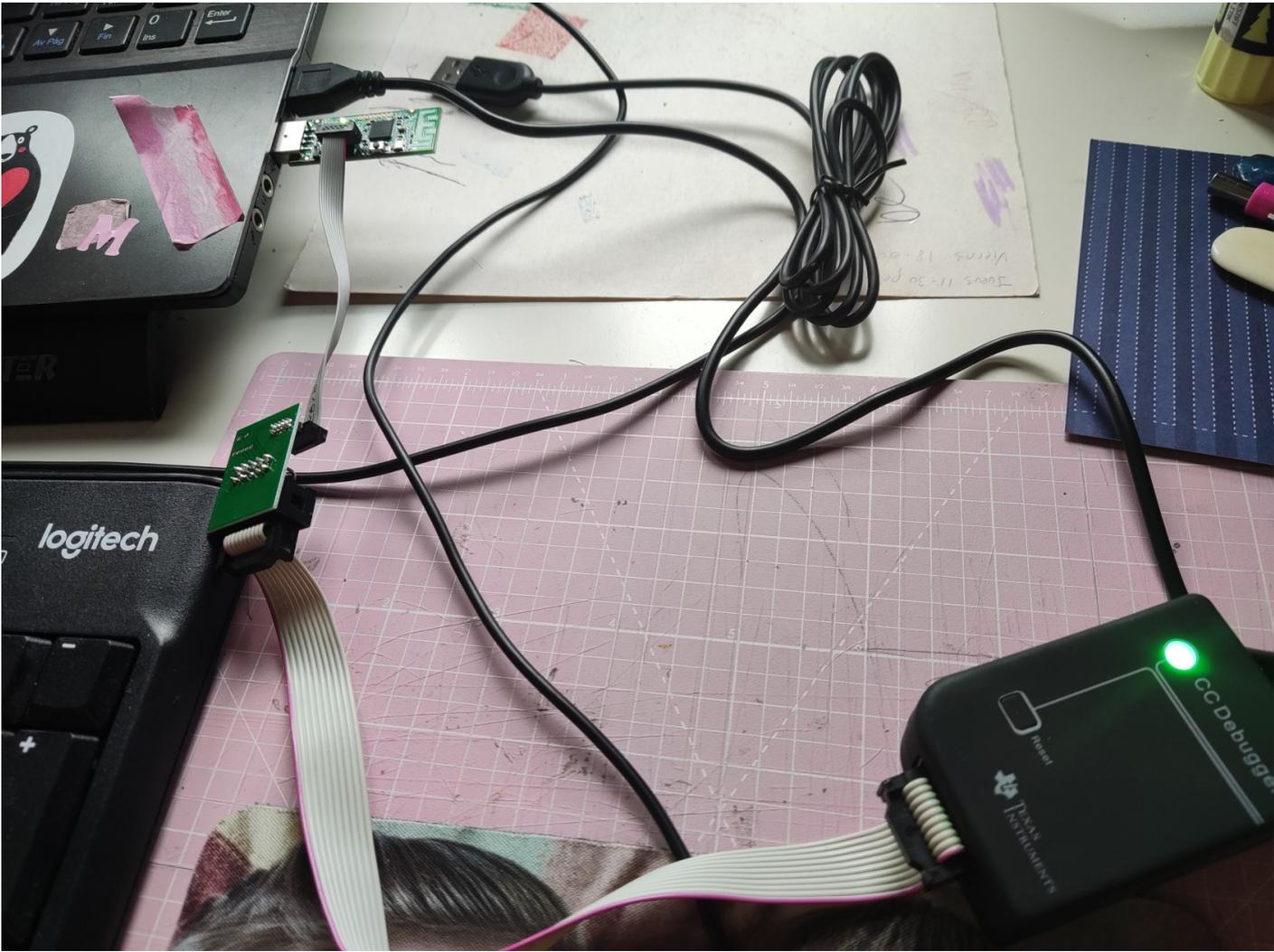
Imagen obtenida de <https://es.aliexpress.com/item/32971386349.html>

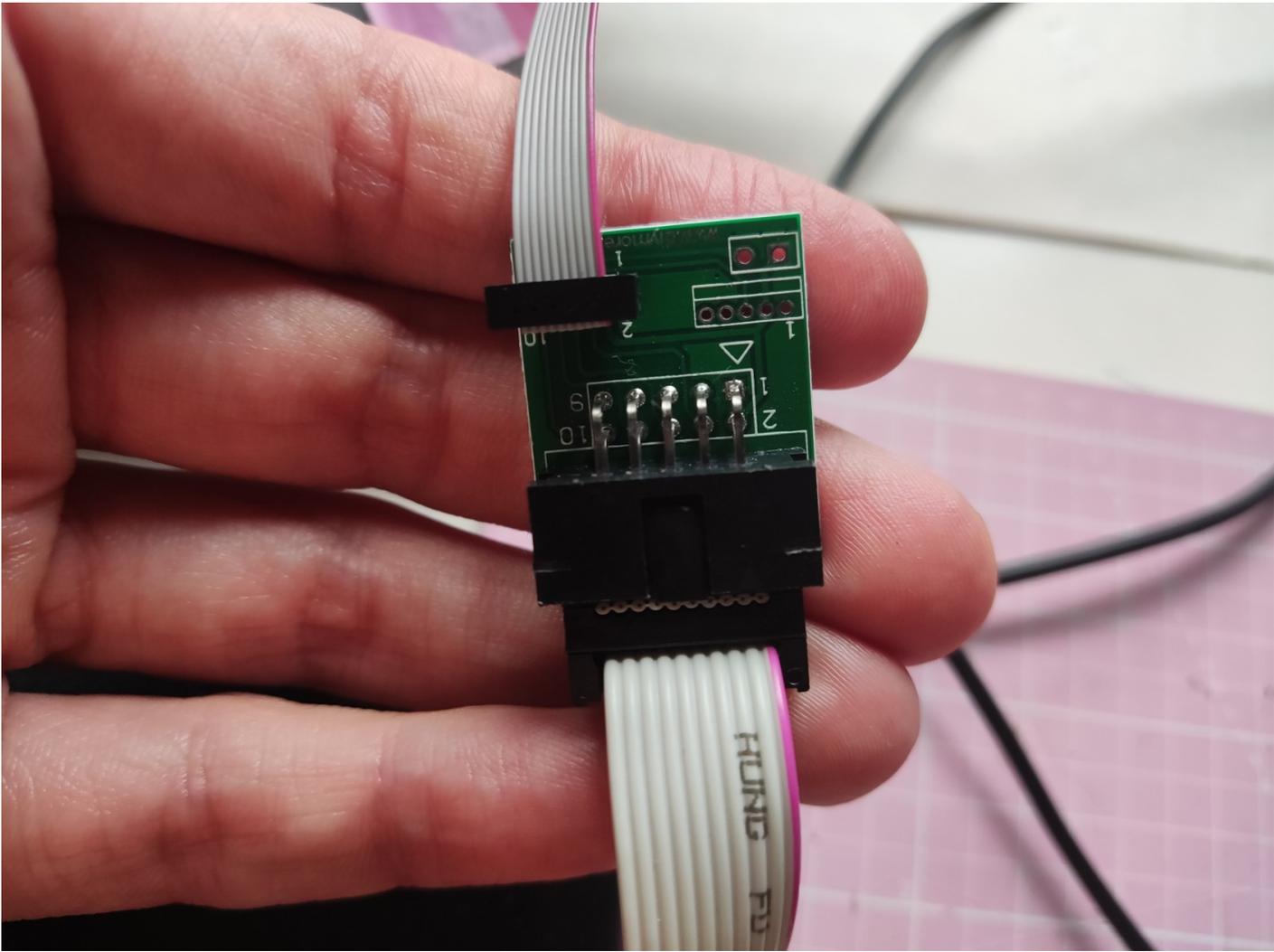
Puede ser que el dispositivo que tengas entre tus manos ya esté flasheado si otro compañero/a lo ha flasheado en una edición anterior del curso. Si así lo deseas puedes volver a flashearlos sin dañarlo.

Os dejo un videotutorial que explica como *flashear* el dispositivo por si os animáis a utilizar la tecnología Zigbee. Os dejo también el enlace al vídeo ya que hay veces que no carga incrustado en este manual <https://youtu.be/70Eav-prgMk> En esta página del curso he anexionado (aparecen arriba a la izquierda) también los ficheros necesarios por si en el futuro dejan de estar disponibles en su ubicación actual.

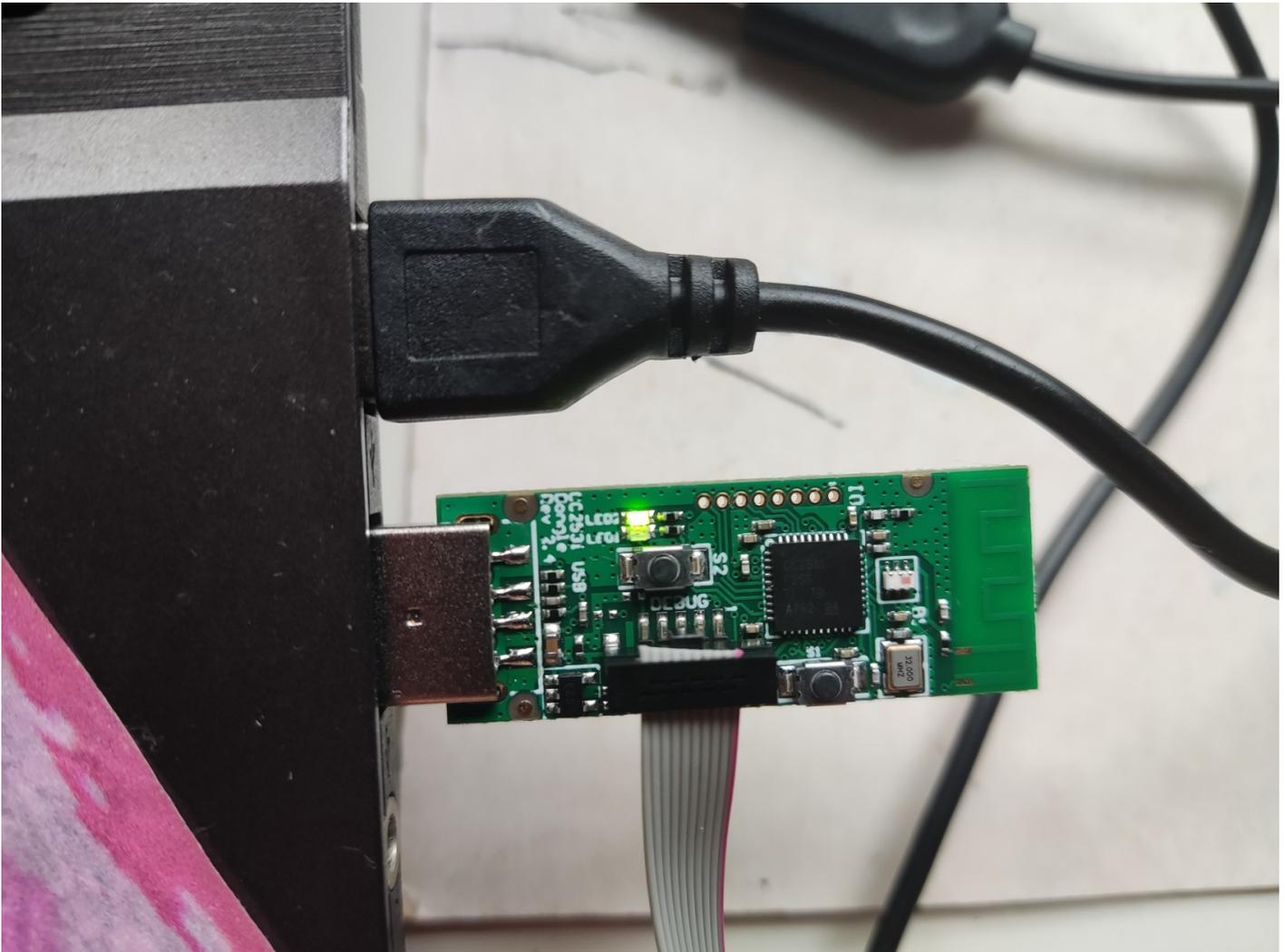
<https://www.youtube.com/embed/70Eav-prgMk>

Os dejo alguna foto de las conexiones del proceso de flasheo por si no lo veis correctamente en el vídeo:









Elaboración propia

El ccdebugger en ocasiones le cuesta MUCHO cambiar el led de rojo a verde. Es cuestión de insistir, reiniciar,... La primera vez que lo hice me costó 2 minutos. La segunda vez que lo hice me costó 20 minutos.

Os dejo unas capturas de pantalla del programa de flasheo:



What do you want to program?

Program CCxxxx SoC or MSP430

System-on-Chip | MSP430

EB ID	Chip type	EB type	EB firmware ID	EB firmware rev
2407	CC2531	CC Debugger	05CC	0041

Interface:

Fast

Flash image: C:\Users\Pablo\Desktop\Pablo - Borrarme\Z-Stack-firmware-master\coordina ...

Read IEEE | Write IEEE

Location

Primary Secondary

IEEE 0x

Retain IEEE address when reprogramming the chip

View Info Page

Actions

- Erase
- Erase and program
- Erase, program and verify
- Append and verify
- Verify against hex-file
- Read flash into hex-file

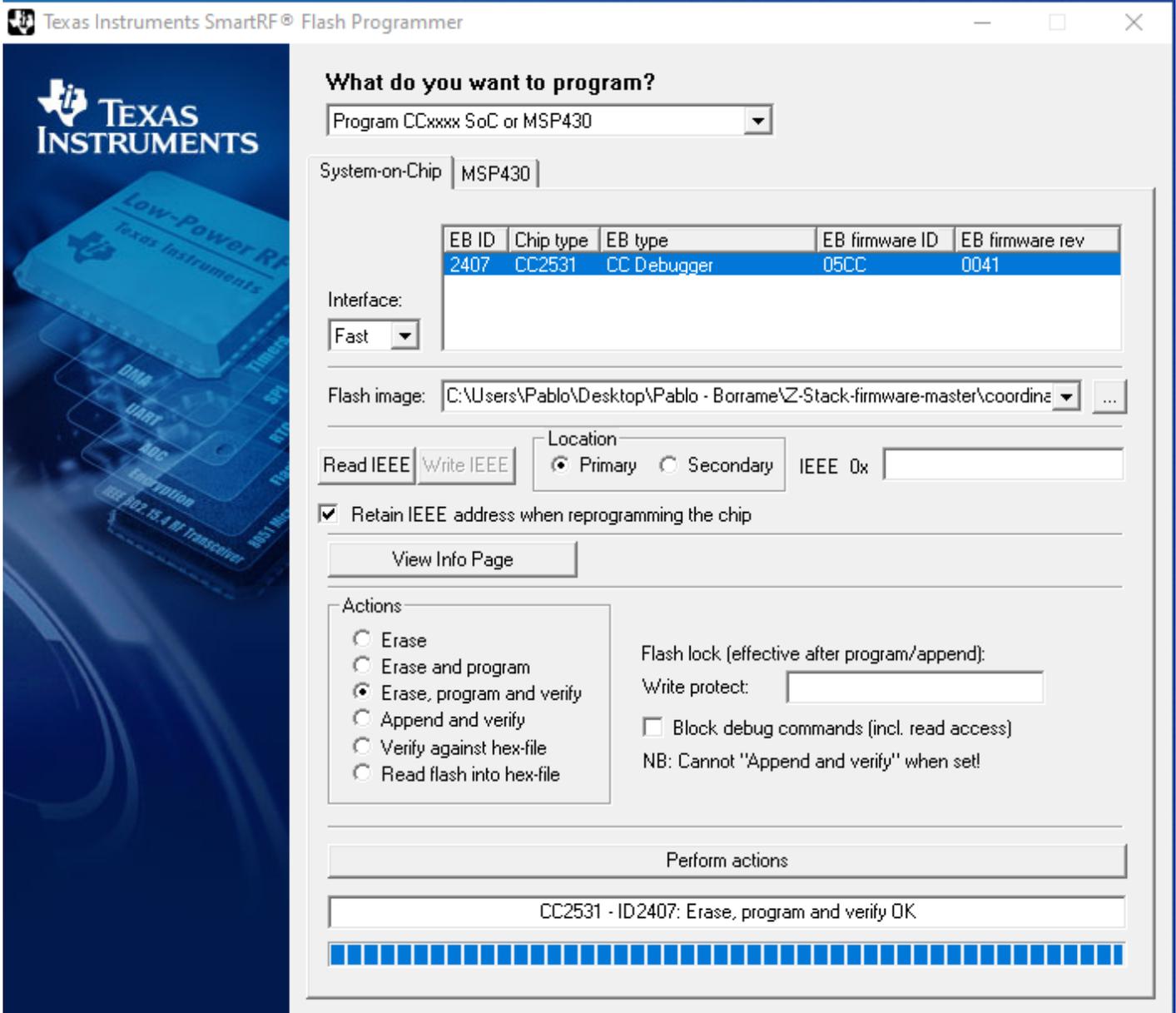
Flash lock (effective after program/append):

Write protect:

Block debug commands (incl. read access)

NB: Cannot "Append and verify" when set!

Perform actions



Elaboración propia

Tipos de dispositivos IoT

A la hora de adquirir dispositivos IoT los encontraremos de 2 tipos:

- conectados a la luz
- con batería/pilas

en algunos casos tiene sentido que estén conectados a la luz, es el caso de un termostato de la calefacción o un motor para subir/bajar las persianas, mientras que otros, como un sensor de temperatura o un actuador de radiador, tiene mas sentido que funcionen con batería/pilas en lugar de tener que llevar un enchufe a cada radiador o a cada punto dónde queramos medir

Además de la clasificación anterior, nos encontraremos básicamente dispositivos de 2 tipos: Wi-Fi y Zigbee pero no Bluetooth ¿por qué? por el consumo. Los dispositivos que utilizan el protocolo Bluetooth consumen mucha más batería que los que usan, por ejemplo, Zigbee. Nadie quiere un dispositivo IoT al que tiene que cambiarle las pilas cada mes cuando existe uno equivalente al que puede cambiarle la pila cada 2 años.

Privacidad

Una cuestión que yo considero MUY importante es el que el acceso a nuestros datos o a la funcionalidad del hardware no requiera de tener acceso a la nube de una determinada empresa por cuestiones ya no solo de privacidad sino por el hecho de que si esa empresa cierra su nube o establece un servicio de pago estamos condenados a pagar o cambiar el dispositivo con las implicaciones medioambientales y económicas que ello supone. Muchos fabricantes, como [tuya](#), plantean que para usar sus dispositivos Zigbee debes adquirir su hub (concentrador) y posteriormente este hub, a través de su app o web, te dará los datos que recopila o te permitirá interactuar con tus dispositivos. Frente a esta filosofía está la posibilidad de utilizar un software propio al que conectemos nuestros dispositivos (WiFi o Zigbee) y de la compañía que queramos (tuya, sonoff,...) sin depender de nubes de terceros. Además, con soluciones como las que hemos visto en el capítulo 3 para la creación de VPNs, podremos acceder a nuestro dispositivo desde cualquier lugar. Por ello en el siguiente apartado vamos a hablar de Home Assistant.

Home Assistant



Imagen obtenida de <https://design.home-assistant.io/#brand/logo>

Hassio, por si mismo, requeriría de un curso dedicado de una cantidad importante de horas por ello vamos a centrarnos en aquello más relevante.

Antes de comenzar voy a mostraros mi instalación de Home Assistant con los diferentes sensores, gráficas y automatizaciones que tengo así como por qué empecé a montar este sistema y qué

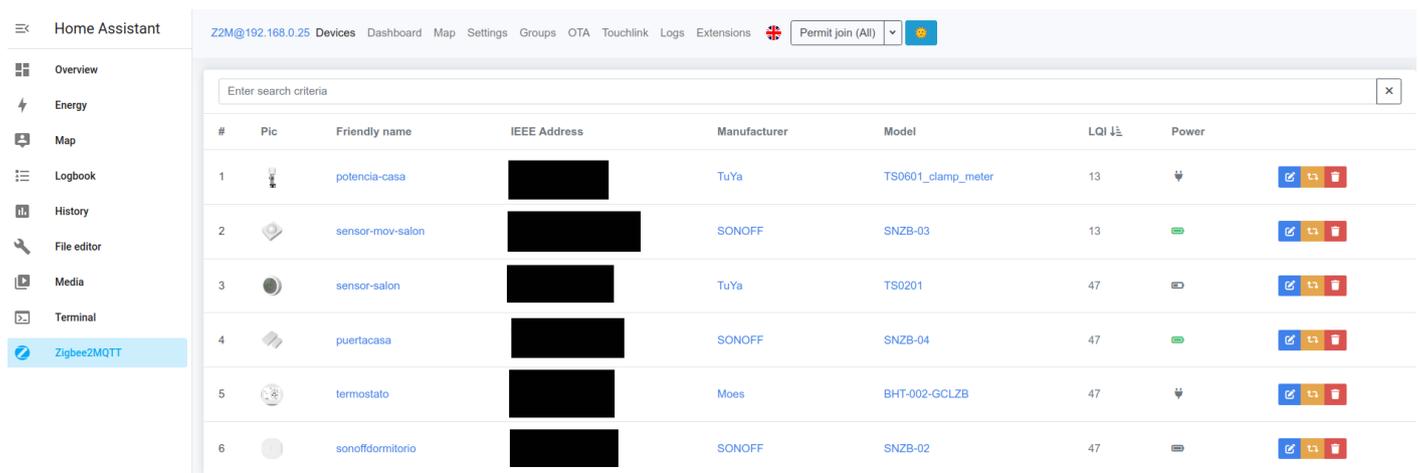
quiero hacer en el futuro.

Inicialmente monté esto para ubicar en cada dormitorio un sensor de temperatura y posteriormete ubicar un actuador de apertura/cierre en cada radiador. La idea era abrir o cerrar cada radiador de modo automático en función de la temperatura de cada estancia. Con posterioridad leí que estos actuadores con tanta apertura/cierre acababan estropeandose y por ello desistí de instalar estos actuadores en los radiadores pero la mecha ya estaba prendida.

En la actualidad tengo los siguientes sensores/actuadores:

- sensor de apertura: en la puerta de entrada a mi domicilio
- sensor de presencia: en la primera estancia a la que se accede desde la puerta de entrada
- sensor de temperatura y humedad en salón
- sensor de temperatura y humedad en el dormitorio principal
- termostato
- sensor de consumo: en el cuadro eléctrico de casa
- sensor de apertura: frigorífico
- receptor/emisor de infrarrojos
- detección de aparatos en red de casa

Todos, excepto el último, funcionan con zigbee mientras que el último hace uso de wifi.

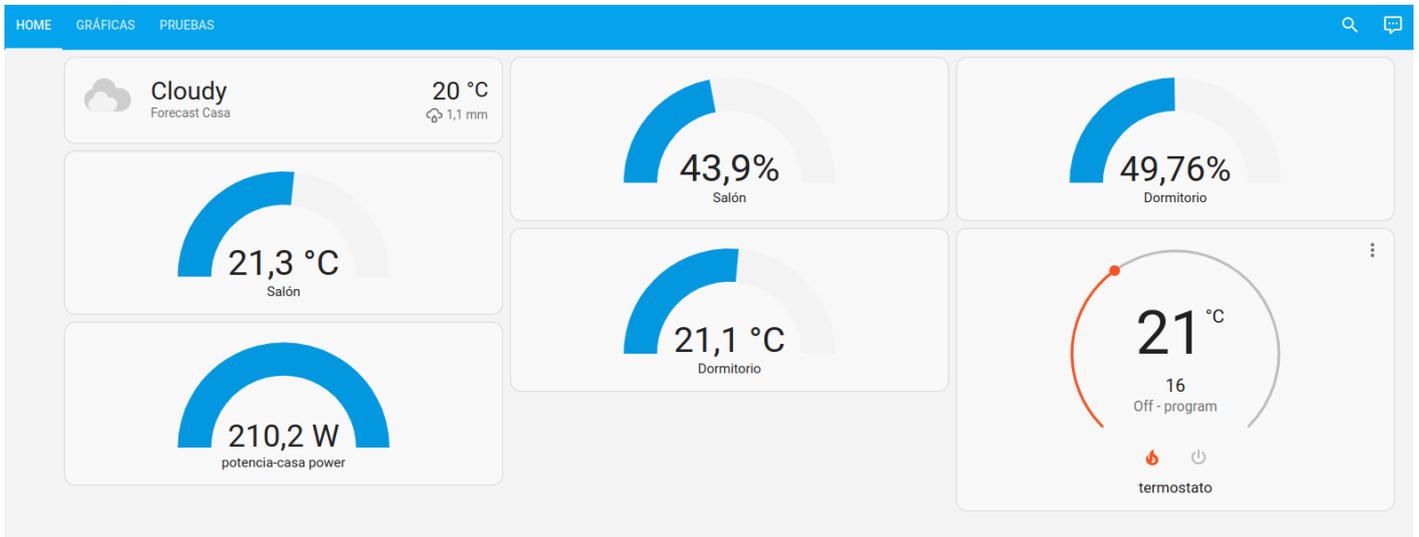


The screenshot shows the Home Assistant interface with the Zigbee2MQTT extension active. The main panel displays a table of devices connected via Zigbee. The table has columns for ID, Pic, Friendly name, IEEE Address, Manufacturer, Model, LQI, and Power. There are 6 devices listed.

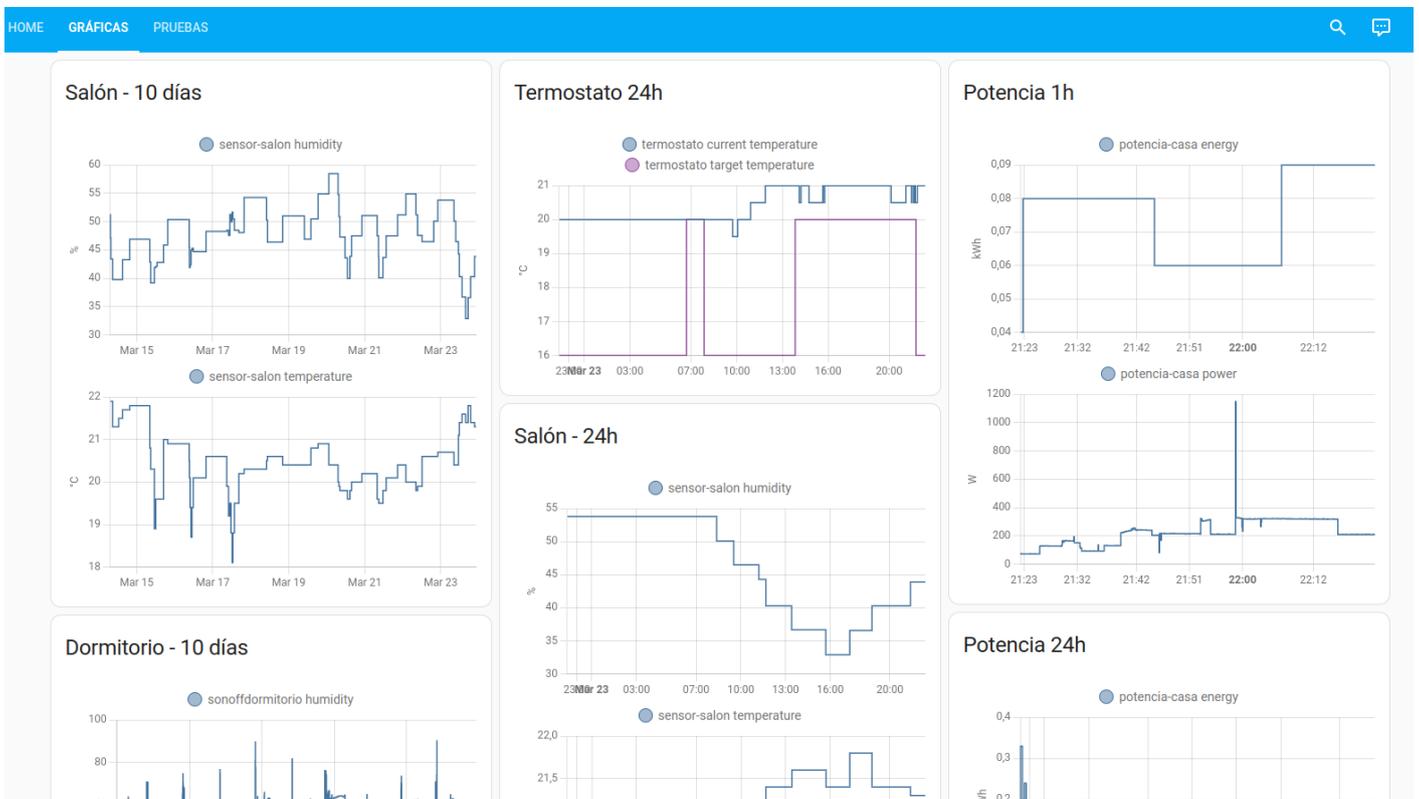
#	Pic	Friendly name	IEEE Address	Manufacturer	Model	LQI ↓↑	Power
1		potencia-casa	[REDACTED]	TuYa	TS0601_clamp_meter	13	⚡
2		sensor-mov-salon	[REDACTED]	SONOFF	SNZB-03	13	🟢
3		sensor-salon	[REDACTED]	TuYa	TS0201	47	🟡
4		puertacasa	[REDACTED]	SONOFF	SNZB-04	47	🟢
5		termostato	[REDACTED]	Moes	BHT-002-GCLZB	47	⚡
6		sonoffdormitorio	[REDACTED]	SONOFF	SNZB-02	47	🟢

Elaboración propia

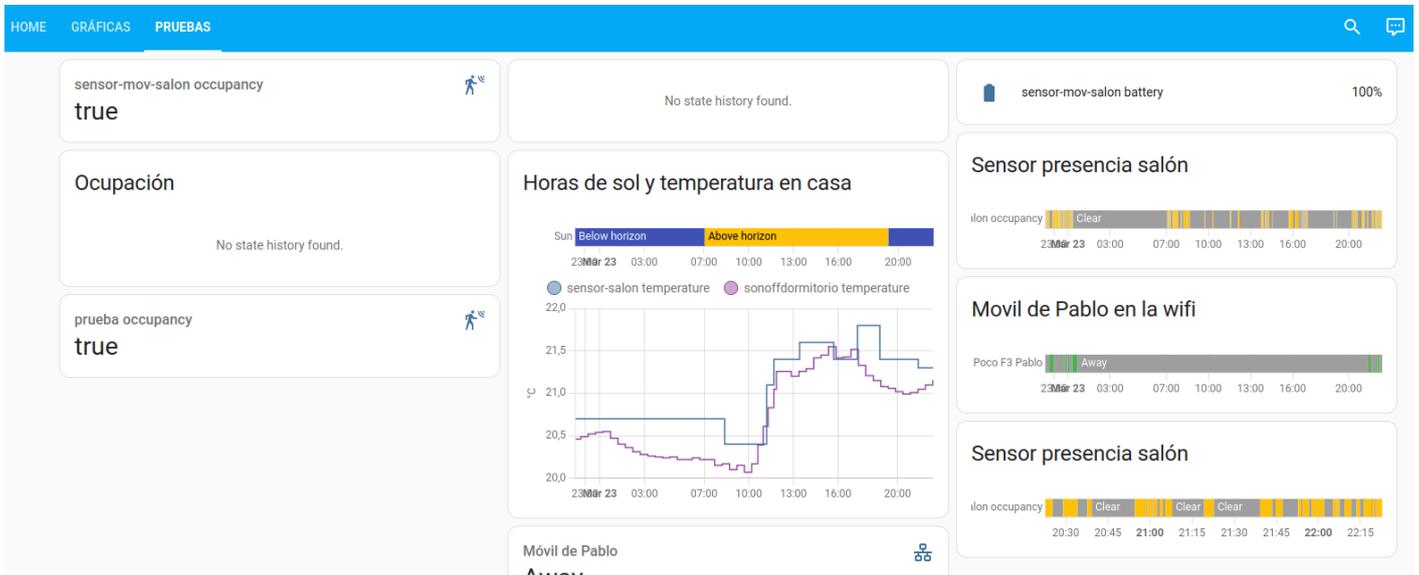
Estos sensores me permiten obtener valores del instante en que realice la consulta pero también históricos. También me permite actuar directamente sobre los sensores como el termostato. Todo ello se configura en unos dashboards muy sencillos como los que vemos a continuación



Elaboración propia



Elaboración propia



Elaboración propia

A partir de lo anterior se pueden crear diferentes automatizaciones. Por ejemplo yo tengo creadas las siguientes:

- Frigo abierto: Mi frigorífico no tiene sensor avisador si te dejas la puerta abierta por lo que intenté hacer un avisador en caso de que estuviese abierto mas de 10s. Mis hijas arrancaron el sensor antes de ponerlo en marcha.
- Frigo cerrado:
- Puerta de casa abierta: Me manda un mensaje por telegram cada vez la puerta de casa se abre. Lo uso a modo de alarma
- Puerta de casa cerrada:
- Termostato temperatura medida cambió por debajo de 19°C: Lo tengo creado para pruebas con el termostato



Automations

Scenes

Scripts

Blueprints



Search



↑ Name

Last triggered



Frigo abierto

Apr 26, 07:39



Frigo cerrado

Dec 11, 12:18



Puerta de casa abierta

1 hour ago



Puerta de casa cerrada

1 hour ago

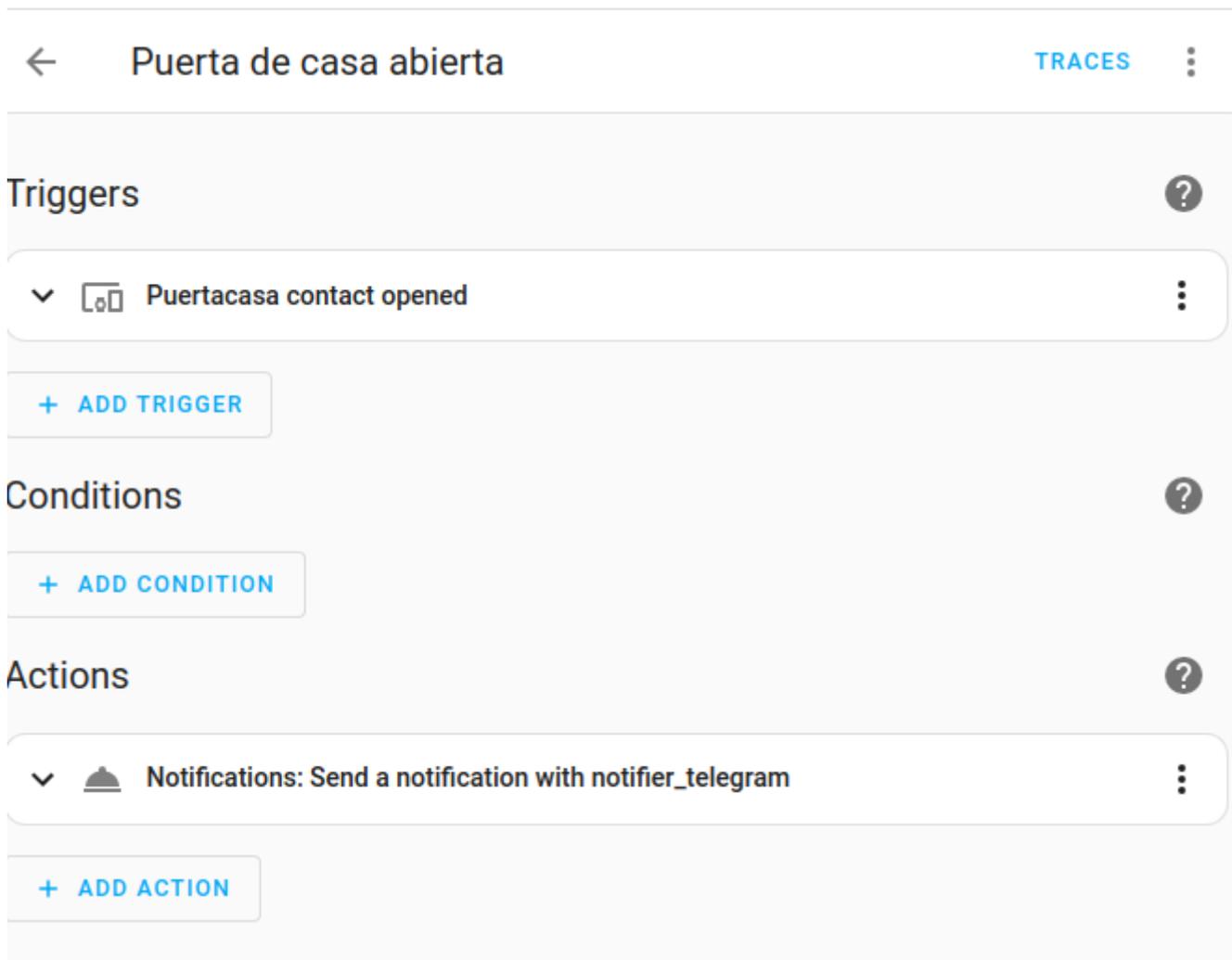


Termostato temperatura medida cambió por ...

2 days ago



Elaboración propia



Elaboración propia

Visto lo que yo tengo es momento de empezar por la **instalación**: Home Assistant (<https://www.home-assistant.io/>) puede ser instalado dockerizado o puede ser instalado directamente como un sistema operativo en una tarjeta SD dedicada. En mi caso opto por la 2ª opción pues como vemos en la imagen posterior el hecho de tenerlo instalado de este modo me permite disfrutar de todas las posibilidades de la herramienta.

Compare Installation Methods

	OS	Container	Core	Supervised
Automations	✓	✓	✓	✓
Dashboards	✓	✓	✓	✓
Integrations	✓	✓	✓	✓
Blueprints	✓	✓	✓	✓
Uses container	✓	✓	✗	✓
Supervisor	✓	✗	✗	✓
Add-ons	✓	✗	✗	✓
Backups	✓	✓ ¹	✓ ¹	✓
Managed OS	✓	✗	✗	✗

1: Backups for Home Assistant Core and Home Assistant Container is provided by the [backup integration](#).

Imagen obtenida de <https://www.home-assistant.io/installation/>

En la dirección <https://www.home-assistant.io/installation/> tenemos a nuestra disposición las diferentes formas de instalación. En el caso de elegir la instalación vía sistema operativo descargaremos la imagen y la instalaremos tal como vimos en el [capítulo 2.1.1](#). Os dejo también un tutorial por si os resulta de utilidad <https://domoticaencasa.es/video-tutorial-instalacion-de-home-assistant/>

<https://www.youtube.com/embed/Ncg9unb2gZs>

Si vais a trabajar con dispositivos Zigbee necesitaréis conocer qué es Zigbee2Mqtt por ello os dejo este enlace <https://domoticaencasa.es/video-funciona-mqtt-integrar-home-assistant/> a un tutorial que explica como funciona y como integrarlo en Home Assistant. Os dejo también un vídeo:

<https://www.youtube.com/embed/YPS5xI6Bx3I>

El protocolo MQTT se usa en robótica educativa, por ejemplo [aquí en el curso ESP32 en el Aula con Arduinoblocks](#)

Si te quedas con ganas de mas voy a recomendarte esta serie de 8 vídeos dónde indican como proceder con hassio y zigbee:

<https://www.youtube.com/embed/FGTLpdgvfPI>

Una función muy interesante es la de que hassio nos envíe notificaciones en Telegram por ello os dejo un enlace a este tutorial <https://domoticaencasa.es/home-assistant-9-notificaciones-telegram/> dónde explica como hacerlo. Este canal os puede resultar también de gran utilidad para los servicios que hemos desplegado a lo largo del capítulo 3.

Integraciones

Si además de las integraciones que pone por defecto a tu disposición Home Assistant quieres hacer uso de otras no oficiales puedes instalar HACS que nos permite acceder a cientos, si no miles, de proyectos que la comunidad ha creado. Para instalar HACS es suficiente con seguir estas instrucciones:

<https://www.youtube.com/embed/4-ujjhKU8CE>

Revision #17

Created 4 February 2023 10:59:24 by Pablo Ruiz

Updated 11 January 2024 18:28:21 by Javier Quintana