1 Estudio del problema. Justificación del tema elegido.

1.1 Introducción. Breve descripción del proyecto.

En la actualidad existen diversas marcas que ofrecen sistemas de climatización domotizadas, sin embargo una gran cantidad de empresas quedan fuera de este mercado por no incorporar sistemas que permitan controlar de modo remoto la climatización en el hogar, los motivos por los cuales no implementan sus productos con estos sistemas son varios, como por ejemplo:

* no pertenecer parte de los componentes a su ámbito habitual de trabajo,
* no contar con un departamento I+D capaz de dar respuesta a estas necesidades,
* el desconocimiento sobre las áreas de (internet de las cosas), lo cual necesariamente desemboca en un inminente desfase tecnológico.

El proyecto consiste en la creación de un prototipo, y el diseño de un sistema que permitan el control en tiempo real de una estufa eléctrica de aceite desde una aplicación web.

1.2 Finalidad. Qué queremos conseguir con el sistema a implementar.

La finalidad es demostrar que es posible adaptar a productos de origen (las actuales y desfasadas estufas) nuestro prototipo de mejora para el control remoto.

Con el sistema propuesto se pretende conseguir mayor comodidad para el usuario / cliente, pues se ofrece la posibilidad de monitorizar el estado previo de la estancia, y de controlar los cambios que se desean realizar. Se pueden controlar varios dispositivos por distintos usuarios dentro de una misma cuenta cliente.

1.3 Objetivos. Qué servicios ofrecerá el sistema una vez implementado.

Los objetivos, como ya se han descrito, son:

* actualizar un electrodoméstico el cual necesita de la presencia física de una persona y mejorarlo para posibilitar su uso, control y manejo de manera remota,
* dar a las empresas fabricantes acceso a un nicho de mercado que por el momento no está siendo explotado.

Los servicios que ofrece el sistema es el control on-line de dicho aparatos.

2. Recursos necesarios.

2.1 Recursos hardware.

Infraestructura

Como servidor web utilizare Raspberry Pi 2 Modelo B, un ordenador de placa reducida que incorpora un procesador con arquitectura ARM de bajo coste.



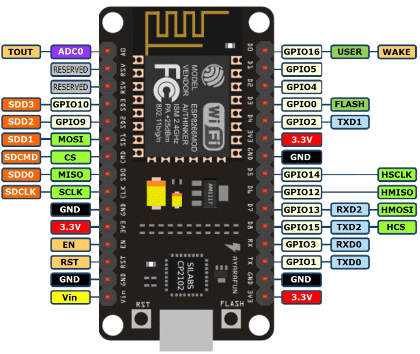
La elección de esta solución respecto a la tradicional, que sería un servidor compartido, es debido a la naturaleza del proyecto ya que en un servidor compartido no puedo ejecutar un demonio (aplicación que ofrezca un servicio en segundo plano), otra posible solución hubiese sido alquilar un servidor dedicado, aunque toda la configuración pero los costes hubieran sido elevados.

Proveedor de internet ONO a través del router -----marca modelo --- que se utilizara para conexión internet del servidor.

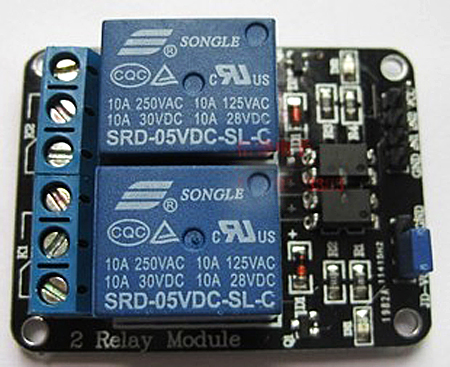


Dispositivo

1. Placa de desarrollo con chip ESP8266 Modelo NodeMCU, esta placa lleva incorporado el chip en el que se vasa nuestro proyecto, al ser una placa de desarrollo nos va a facilitar pines de conexión rápida y una conexión USB para mandar el código compilado desde el ordenador a la placa.



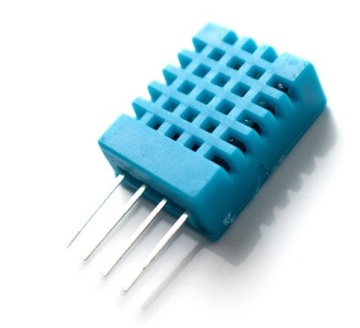
# 2. 2 vías relé de protección con el módulo optoacoplador tarjeta de expansión de relé, será la encargada de dar corriente a las resistencias de la estufa.



3. Desplaye oled 0.96 I2C EL0443, donde se imprimir los datos de la estufa.



Sensor de temperatura y humedad DHT11, muy común para realizar proyectos en los cuales se quiera conocer la temperatura.



2.2 Recursos software.

Como base de datos se ha escogido MySQL, que es una de las más utilizadas en el mudo del desarrollo de bases de datos y de la cual existe mucha documentación. La herramienta escogida para gestionar la base de datos es Workbench, que es de la misma empresa que desarrolla MySQL.

Los lenguajes de programación que se van a utilizar son:

* Para el servidor php, lenguaje interpretado, lo cual nos evita el tener que compilar el código antes de ejecutarlo, enfocado a la programación web, muy bien documentado y con soporte para programación orientada a objetos.
* En el lado del cliente un clásico como javaScript, también es un lenguaje interpretado soportado por todos los navegadores, altamente extendido y dentro del estándar de HTML5.
* Para la programación del chip ESP8266, dentro de las pocas opciones que nos encontramos, escogemos un lenguaje basado en C++, el cual se utiliza para la programación de los dispositivos de la plataforma Ardunio, altamente extendida para la crear prototipos electrónicos, esta elección viene motivada por la gran cantidad de librerías que existen para componentes electrónicos, en los que se encuentran los que vamos a utilizar.
* Para la parte del navegador HTML para la estatus y css.

Para gestionar las dependencias de las librerías utilizadas en el servidor, programado con php, se ha optado por COMPOUSER, esta herramienta es ideal ya que en un único archivo de configuración podemos definir, todas las librerías que necesitemos, especificando una versión en concreto si es necesario, también podremos definir los espacios de nombre que vamos a utilizar en nuestro proyecto, para evitar las posibles colisiones que puede haber entre el nombre de nuestras clases y los de las librerías utilizadas. Después de haber especificado todos los parámetros en el archivo de configuración, el programa se encarga de descargar las librería y las dependencias que estas tengan asociadas. También nos genera un autolouder vasado en el entandar psr-4.

El sistema operativo (SO) en el que voy hacer funcionar la Raspberry es el Ubuntu Core, SO basado en Debian y adaptado a la arquitectura ARM de la Raspberry, si bien no es SO que el desarrollador de la Raspberry desarrolla para su producto sí que se encuentra en su página web para su descarga.

Se ha escogido Slim que es un micro framework, rápido que nos permite abstraernos del manejo de las peticiones HTTP, muy documentado y sencillo de utilizar.

Como servidor web HTTP Apache server software de código abierto y multiplataforma, siendo muy flexible, altamente configurable y el más utilizado.

Se ha escogido GIT como sistema de control de versiones. Nos permite tener versionado el código tanto de forma local como en un servidor que en este caso va ha ser github.

HERRAMIENTAS PARA EL DESARROYO

Para la edición del código se va a utilizar atom, rápido y ligero, con un montón de extensiones que nos permiten configurarlo a nuestro gusto.

Para el desarrollo voy a utilizar una maquina virtual con montada con el programa Virtualbox, para facilitar la compatibilidad con la Raspberry, se va a instalar en la maquina el SO Ubuntu server.

Para automatizar el proceso de montaje de la máquina virtual se va utilizar Vagrant, este software funciona definiendo un archivo de configuración con las características que tiene que tener la máquina virtual (sistema operativo, carpetas compartidas, tipo de conexión,…) y el programa se encarga de montarla, también se ha definido un script con todos los programas necesario para que funcione la aplicación que Vagrant se encarga de ejecutar después de la instalación del SO y su configuración. También configura una conexión ssh, con la cual podremos acceder a la maquina desde cualquier punto de la red, esto nos servirá para pequeñas modificación y comprobaciones de funcionamiento.

Fase de desarrollo:

Virtual vox para la creación de maquinas virtuales.

Es una herramienta para la creación y configuración de entornos de desarrollo virtualizados.

Como gestor de control de versiones local git y github para la gestión remota.