



UNIVERSIDAD ALFONSO X EL SABIO

ESCUELA POLITÉCNICA SUPERIOR

GRADUADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL

MÓDULO DE DOMÓTICA CON ARDUINO EN RED DE IOT

ALUMNO: GOZALO DÍAZ DEL RÍO FONTÁN

N.P.: 121291

DIRECTOR DEL TRABAJO: MARIA ANTONIA SIMÓN

FECHA DE PRESENTACIÓN: (FECHA)

BREVE DESCRIPCIÓN:

DESARROLLO DE UN MÓDULO DE CONTROL REMOTO AUTOMÁTICO Y VERSÁTIL PARA APLICACIONES DE DOMÓTICA, ASÍ COMO DEL SISTEMA DE CONTROL Y GESTIÓN PARA ESTOS MÓDULOS MEDIANTE EL USO DE ARDUINO Y RASPBERRY PI. APLICACIONES DE IOT EN LA DOMÓTICA Y LA INFORMÁTICA INDUSTRIAL.

FIRMA DEL DIRECTOR

FIRMA DEL TUTOR

FIRMA DEL ALUMNO

1. ÍNDICE

| | |
|---|----|
| 1. Índice..... | 1 |
| 2. Introducción | 2 |
| 3. Estudio de concepto | 3 |
| 3.1 Internet de las Cosas..... | 3 |
| 4. Fundamento teórico..... | 4 |
| 4.1 Arduino | 4 |
| 4.1 a) Arduino Nano | 4 |
| 4.2 Raspberry Pi | 4 |
| 4.3 Node-Red | 4 |
| 4.4 MQTT | 5 |
| 4.5 PuTTY, FileZilla | 5 |
| 5. Materiales y programas Empleados | 6 |
| 5.1 Hardware | 6 |
| 5.2 Software | 6 |
| 5.2 a) Raspberry Pi | 6 |
| 5.2 b) ESP8266 | 6 |
| 5.2 c) PC | 6 |
| 6. Procedimiento | 7 |
| 6.1 Raspberry Pi | 7 |
| 6.2 Módulo ESP8266 | 8 |
| 6.3 Arduino Nano | 8 |
| 7. Conclusión..... | 10 |

2. INTRODUCCIÓN

En este proyecto se pretende diseñar un módulo controlador para ser empleado en ámbito de la domótica. Dicho módulo consistirá en una placa microcontroladora Arduino con un módulo de Wifi acoplado para permitir las comunicaciones inalámbricas. Estos módulos serán capaces de activar o desactivar actuadores en función de comandos originados en un ordenador central en la forma de una Raspberry Pi, así como de leer distintos sensores y comunicar esta lectura al ordenador.

En el consiguiente proyecto se realizará la programación y montaje de un módulo de Automatización mediante el uso de un Arduino Nano equipado con una antena de Wifi, así como la configuración de un Ordenador de Control.

El objetivo del proyecto es crear un módulo barato y accesible que permita controlar de forma inalámbrica cualquier dispositivo al que se conecte. Este control será posible realizarlo desde cualquier dispositivo capaz de comunicarse con una red Wifi.

El principal enfoque del proyecto es la versatilidad del producto. Se pretende que los módulos controladores puedan ser empleados para actuar sobre una gran variedad de dispositivos sin requerir alteraciones en su código. Para ello se pretende centralizar el control y la automatización en el ordenador central, minimizando la algoritmia contenida en los microcontroladores.

3. ESTUDIO DE CONCEPTO

Nos encontramos en una época en la que la automatización en el ámbito de la domótica está en pleno auge. En todos los comercios de electrónica se venden y promocionan los dispositivos “Smart-home”, que prometen emular la existencia de un mayordomo virtual que encienda o apague luces, abra y cierre puertas, controle la temperatura, realice videovigilancia... Decenas de productos y funciones distintas se ponen a nuestra disposición para facilitarnos una interacción más automática con nuestro día a día.

Sin embargo, estas múltiples soluciones se encuentran siempre asociadas a una marca comercial, con Amazon Echo y Google Home a la cabeza del mercado actual, y requieren que todos los componentes del sistema pertenezcan a la misma familia. Con este proyecto se pretende crear un dispositivo que nos permita ejercer este nivel de control sobre aparatos y electrodomésticos “no inteligentes”, sirviendo así de puente para permitir la automatización domótica sin necesidad de actualizar todos los aparatos ya presentes en casa.

3.1 INTERNET DE LAS COSAS

El Internet de las Cosas (Internet of Things, IoT) es un sistema de dispositivos de computación, controladores mecánicos o digitales y otros aparatos interconectados entre sí con identificadores únicos. Esta red de dispositivos cuenta con la capacidad de transmitir datos entre sí y realizar diversas tareas sin necesidad de interacción humana. El Internet de las Cosas nace de los sistemas empotrados, la necesidad de análisis en tiempo real del machine learning.

Actualmente las aplicaciones del Internet de las Cosas están siendo constantemente expandidas y exploradas por la Industria para todo tipo de aplicaciones. Gracias al IoT, los sistemas cableados tradicionales y los complejos y delicados autómatas, omnipresentes en ámbitos industriales, están empezando a desaparecer para ser sustituidos por dispositivos más modernos con capacidades de IoT, mucho más versátiles y modulares.

4. FUNDAMENTO TEÓRICO

Para la realización de este proyecto se han empleado una variedad de tecnologías y lenguajes de programación, que se detallan a continuación:

4.1 ARDUINO

Las placas de desarrollo Arduino son una colección de microcontroladores con una memoria flash y varios pines de entrada/salida asociados recogidos bajo una licencia de hardware libre. Su principal uso es en el ámbito de la robótica y la automatización y son programables a través de un software de licencia libre empleando el lenguaje C++. Una placa de Arduino puede leer y modificar el estado de tensión de sus pines tanto de forma discreta digital, con dos niveles lógicos, como de forma analógica, permitiendo leer y emitir valores continuos.

Se trata de una pieza de hardware barata, resistente, sencilla de programar, muy personalizable y enormemente versátil. Existen numerosos modelos con diferencias en el número de pines, la potencia del microcontrolador, el tamaño de la memoria y el número de pines. Para este proyecto se empleará el modelo Nano debido a su pequeño tamaño, pese a que el funcionamiento es idéntico si se desea emplear el modelo UNO o MEGA.

4.1 a) Arduino Nano

Entre las múltiples variedades de placas de desarrollo de Arduino, el Nano es la versión más reducida de tamaño que sin embargo mantiene todas las funcionalidades del modelo UNO, el más tradicional. El Nano cuenta con un controlador ATmega328P de 16MHz, con 1KB de EEPROM, 32 KB de memoria Flash y 2KB de SRAM. Para funcionar requiere alimentación entre 7 y 12 Voltios y opera sus entradas y salidas a 5V. Cuenta con 14 pines que pueden ejercer de entradas / salidas digitales (6 que permiten salidas PWM) y 8 analógicos. Puede emitir hasta 40mA de corriente por cada pin. Puede conectarse por USB a un ordenador para su programación y alimentación o puede ser alimentado a través de uno de sus pines. A través de otro de sus pines puede alimentar dispositivos a 3.3 V.

4.2 RASPBERRY PI

Las Raspberry Pi son una serie de pequeños ordenadores de una sola placa. Sus aplicaciones son prácticamente ilimitadas y se encuentran presentes en aplicaciones industriales, educativas, investigadoras y en robótica, destacando gracias a su sorprendente potencia, versatilidad y portabilidad. Existen múltiples generaciones del producto, cada una con capacidades mejoradas. Para este proyecto se empleará una Raspberry Pi 4 Modelo B de 4 Gb de memoria, pese a que el proyecto se puede replicar a la perfección con cualquier modelo a partir de la Raspberry Pi 3.

Para su funcionamiento, como con cualquier ordenador, la Raspberry Pi requiere un Sistema Operativo. Este sistema se instala en una tarjeta SD que luego se inserta en la placa y actúa como disco duro del sistema. Hay numerosas imágenes distintas para Raspberry, dependiendo de la funcionalidad que se desee de ellas. Por su versatilidad y lo extendido de su uso, emplearemos Raspbian, una distribución de Debian adaptada a la Raspberry. Esto significa que, con muy pocas modificaciones, este proyecto puede funcionar con cualquier ordenador operando con una distribución de Linux, como por ejemplo Ubuntu.

4.3 NODE-RED

Node-Red es una herramienta de programación diseñada por IBM para conectar dispositivos de hardware, APIs y servicios online dentro del ámbito del IoT. Consiste en una interfaz web alojada

en un dispositivo que ejerce de anfitrión. En esta interfaz se pueden programar flujos de nodos que intercambian datos en forma de mensajes en formato JSON (JavaScript Object Notation). Al tratarse de una herramienta Open Source, cualquier usuario puede contribuir al proyecto sus propios nodos y funciones, por lo que el proyecto crece exponencialmente con su comunidad y es muy sencillo encontrar nodos y funciones personalizados para la aplicación que se desea. También cuenta con un nodo que permite programar funciones propias en lenguaje Javascript. Al haber sido desarrollado con IBM, su implementación junto con nodos de MQTT lo convierte en la más versátil herramienta para el IoT.

4.4 MQTT

MQTT (Message Queuing Telemetry Transport) es un protocolo de comunicaciones de red muy ligero basado en mecánicas de publicación-subscripción que transmite mensajes entre dispositivos, generalmente haciendo uso de conexiones TCP/IP (comunicaciones de red de Internet). Sin embargo, también está soportado por cualquier protocolo que ofrezca comunicaciones bidireccionales sin pérdidas. Su utilidad destaca en aplicaciones que requieran un código muy ligero o en sistemas con ancho de banda limitado.

El funcionamiento de MQTT es muy simple: los elementos que participan en la comunicación son un gestor de mensajes, el broker, y un número indeterminado de clientes. Los componentes de un mensaje MQTT son un tema y un contenido. Un cliente interactúa con el tema o temas deseados de dos maneras: puede **publicar** un mensaje en un tema concreto y puede **suscribirse** a uno o más temas. El gestor de mensajes se encarga de registrar los mensajes publicados en cada tema y distribuirlos de forma inmediata a todos los clientes suscritos a dicho tema. Existen funcionalidades que permiten asegurar distintas Calidades de Servicio (Quality of Service, QoS) para la comunicación:

- QoS 0: En modo QoS 0, el gestor envía los mensajes a los clientes suscritos en cuanto los recibe y después borra el mensaje. También se conoce como modo “Fire and Forget”.
- QoS 1: En modo QoS 1, el gestor al recibir un mensaje, lo reenviará a los clientes suscritos y esperará una confirmación de mensaje recibido antes de borrarlo. De no recibir la confirmación, volverá a enviar el mensaje de forma periódica hasta recibir la confirmación. Este modo se conoce también como “At least Once”, Al menos una vez.
- QoS 2: En modo QoS 2 el gestor se asegura de que el mensaje le llega al cliente esperando una confirmación por parte de este y luego vuelve a consultar con el cliente si la comunicación ha tenido éxito antes de borrar el mensaje. Este modo se conoce como “Once and Only Once”, Una vez y sólo una vez. QoS 2 también permite emplear la persistencia de mensajes, un sistema mediante el cual el gestor siempre almacena un registro de todos los mensajes enviados a cada tema para que si un nuevo cliente se suscribe a ese tema reciba todos los mensajes que se ha “perdido” antes de su suscripción.

Para la aplicación que nos ocupa emplearemos QoS 2, ya que la información que transmitiremos no es de naturaleza crítica.

4.5 PUTTY, FILEZILLA

PuTTY es un cliente de SSH que permite acceder a la consola de comandos de una máquina de Ubuntu (En este caso el sistema Raspbian de la Raspberry Pi) y manejarla a través de una red a la que esté conectada. FileZilla permite la transferencia de archivos entre dos equipos empleando SSH. Ambos programas son necesarios para operar la Raspberry Pi.

5. MATERIALES Y PROGRAMAS EMPLEADOS

5.1 HARDWARE

- Raspberry Pi 4 Modelo B (4Gb). Puede ser sustituida por cualquier modelo igual o posterior a la Raspberry Pi 3.
- Pantalla táctil LCD para Raspberry Pi (Opcional)
- Arduino Nano v.3.0
- Módulo de Wifi ESP8266 ES01
- Conversor FTDI Mini USB a TTL Serie
- Placas PCB perforadas para soldadura con estaño
- Soldador, estaño, cables
- Transformador de tensión 220V AC – 12 V DC
- Convertidores de tensión 12V – 5V y 5V – 3.3V
- Transistores
- Actuadores varios para demostración (Ventiladores, bombillas LED, etc.)

5.2 SOFTWARE

5.2 a) Raspberry Pi

- Distribución de Raspbian
- Node-Red
- Mosquitto
- Drivers para la pantalla LCD

5.2 b) ESP8266

- Bibliotecas *ESP8266WiFi.h* y *PubSubClient.h* de C++ (Compatibles con el IDE de Arduino)

5.2 c) PC

- IDE de Arduino
- Administrador de placas de ESP8266 para la IDE de Arduino
- PuTTY
- FileZilla

6. PROCEDIMIENTO

A continuación, se detallarán las fases del proyecto para su replicación y los requisitos para el desarrollo de las mismas. Cada fase va precedida de un código que indica a que parte del proyecto pertenece:

RP – Raspberry Pi, TH – Desarrollo teórico, ESP – Módulo Wifi esp8266, AN – Arduino NANO, POT – Potencia

6.1 RASPBERRY PI

RP1 – RASPBIAN

Requisito: Ninguno

En esta fase se instala el sistema operativo Raspbian en la Raspberry Pi.

Del sitio web de Raspberry se descarga el software Raspberry Pi Imager. Una vez descargado y habiendo insertado una tarjeta SD formateada en el ordenador, se ejecuta el programa y se elige grabar en la tarjeta el sistema NOOBS (New Out Of the Box Software), que permite instalar una variedad de imágenes.

Una vez grabado el sistema NOOBS, se debe conectar la Raspberry a un monitor y un teclado, que emplearemos para navegar por el menú de NOOBS hasta instalar Raspbian lite. Una vez instalado debemos acceder a las opciones de configuración del sistema mediante el comando `sudo raspi-config` y activar la comunicación SSH en las opciones de Interfaz.

RP2 – WIFI

Requisito: RP1

En esta fase la Raspberry Pi se configura como punto de acceso Wifi, generando la red “RaspiHouse”.

Para facilitar este proceso, se ha escrito un programa que contiene los comandos bash necesarios para realizar la configuración. Solo es necesario transferir mediante FileZilla el archivo wifiAPconfig.sh al directorio de usuario de la Raspberry y ejecutar desde ese directorio el comando `sudo bash wifiAPconfig.sh`.

Si se desea configurar el SSID y la contraseña de la red wifi, se puede editar el ejecutable con `sudo nano wifiAPconfig.sh` y cambiarlos en las líneas:

```
sudo echo -e "ssid=RaspiHouse" >> /etc/hostapd/hostapd.conf
sudo echo -e "wpa_passphrase=MyIoTProject" >> /etc/hostapd/hostapd.conf
```

RP3 – MQTT

Requisito: RP1

En esta fase se instala y configura mosquitto, el software gestor de mensajes de MQTT que emplearemos. Simplemente es necesario ejecutar los siguientes comandos con la Raspberry conectada a internet:

```
sudo apt-get update
```



```
sudo apt-get install mosquitto
```

RP4 – NODE-RED

Requisito: RP3

En esta fase se instala Node-Red junto con los servicios en segundo plano que permiten añadir nodos nuevos y gestionar el funcionamiento del servicio. Simplemente es necesario ejecutar los siguientes comandos con la raspberry conectada a internet:

```
sudo bash < (curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

```
sudo systemctl enable nodered.service
```

```
sudo npm install node-red-dashboard
```

Por último es necesario transferir mediante FileZilla el archivo flows.json al directorio /home/pi/.nodered de la raspberry

6.2 MÓDULO ESP8266

ESP1 – PROGRAMACIÓN

Requisito: RP4

Se programará el módulo ESP8266 empleando el IDE de Arduino. Para ello se debe montar el siguiente circuito con el ESP8266 y el adaptador de FTDI a TTL.

Una vez montado, desde la IDE de Arduino se debe configurar el módulo. Para ello se accede al menú Archivo -> Preferencias y en el campo de gestor de otras placas se añade la línea http://arduino.esp8266.com/stable/package_esp8266com_index.json

Reiniciamos el IDE de Arduino y ahora en “Herramientas” seleccionamos como tarjeta el módulo genérico de ESP8266. Conectamos el cable al adaptador de FTDI y escribimos el código contenido en el archivo espCode.ino. En este momento se determina el identificador único de este módulo ESP: en el código, bajo “MQTT Configuration” se debe especificar el identificador que se le asignará al módulo (Ver imagen).

```
// MQTT Configuration
const String clientId = "ESP1";
const char* inTopic = "raspihouse/esp1/1";
const char* outTopic = "raspihouse/espout/1";
```

Se observa en la imagen que el nombre del cliente que se ha proporcionado es ESP1 y que para la comunicación MQTT, se suscribirá al tema *raspihouse/esp1/1* y publicará en *raspihouse/espout/2*. Estos valores pueden ser alterados como se desee. Si se van a alterar los temas de MQTT se recomienda leer el Anexo1 para información acerca de la sintaxis de MQTT.

6.3 ARDUINO NANO

AN1 – PROGRAMACIÓN

Requisito: ESP1

Se programará el Arduino Nano haciendo uso de la IDE de Arduino. Simplemente es necesario conectarlo al PC, seleccionar la placa Nano en el selector de placas y escribir en él el código `arduinoCode.ino`

7. PROGRAMACIÓN Y USO DEL SISTEMA

8. CONCLUSIÓN

9. REFERENCIAS

10. ANEXO 1: SINTAXIS DE MQTT

