

Conexión MQTT, Broker - Cliente.

Gina Faisuly Castro Castro, Edward Fabián Hernández Ricaurte, Yeison Stiven Plazas

Acevedo, Álvaro Martínez Forero.

Fundación Universitaria de San Gil – Unisangil

Programa de Ingeniería Electrónica

Ginacastro4@unisangil.edu.co

edwardhernandez@unisangil.edu.co

Yeisonstiven@unisangil.edu.co

alvaromartinez@unisangil.edu.co

Resumen—La presente consideración tiene como fin evidenciar la funcionalidad de un sistema domótico a partir de un protocolo de conexión MQTT; para el desarrollo de la práctica se empleó la tarjeta ESP32 como interfaz de conexión entre el Broker y los dispositivos físicos que integran el sistema domótico (sensores y actuadores).

Los dispositivos físicos que integran el sistema domótico corresponden al sensor de temperatura DHT11, Actuador 1: Ventilador, actuador 2: Servomotor y el protocolo de comunicación broker MQTT (Cliente-Publicador). En primer lugar se realizó la programación en arduino para establecer la conexión entre el módulo ESP32 con la red, configuración de los pines para realizar la conexión del sensor dht11, actuadores, diodos led y realizar las peticiones a través de los tópicos del broker.

En segundo lugar el tópico raíz del broker MQTT realiza la lectura del sistema cada 25 segundos informando el estado de la temperatura y humedad, así mismo el servomotor está a cargo de la apertura y cierre de una puerta, a partir de un segundo tópico se determina si la puerta se encuentra abierta o cerrada. Por otro lado un tercer tópico permite configurar el umbral de temperatura, cuando la temperatura del ambiente sobrepasa el límite el ventilador se activa enfriando la habitación.

Finalmente un sistema de alerta se presenta en dos situaciones: cuando la temperatura sobrepasa el umbral, y cuando se activa la apertura de la puerta. Dicho sistema de alarma se evidencia con el envío de un correo electrónico generando los mensajes de alerta. Así mismo se hace el empleo de un repositorio con Github para compartir la documentación realizada.

Palabras claves— MQTT, Domótica, Internet de las Cosas, WiFi, Control basado en eventos.

Abstract— The present consideration aims to demonstrate the functionality of a home automation system from an MQTT connection protocol; For the development of the practice, the ESP32 card was used as a connection interface between the broker and the physical devices that integrate the home automation system (sensors and actuators).

The physical devices that integrate the home automation system correspond to the temperature sensor DHT11, Actuator 1: Fan, actuator 2: Servomotor and the communication protocol broker

MQTT (Client-Publisher). In the first place the programming was carried out in arduino to establish the connection between the module ESP32 with the network, configuration of the pins to make the connection of the sensor dht11, actuators, led diodes and to make the requests through the topics of the broker.

Second the root topic of the MQTT broker performs the system reading every 25 seconds reporting the state of temperature and humidity, likewise the servomotor is in charge of the opening and closing of a door, a second topic determines whether the door is open or closed. On the other hand a third topic allows to configure the threshold of temperature, when the temperature of the environment exceeds the limit the fan is activated cooling the room.

Finally, an alert system occurs in two situations: when the temperature exceeds the threshold, and when the door opening is activated. This alarm system is evidenced by the sending of an email generating the alert messages. It also makes use of a repository with GitHub to share the documentation made.

Keywords—MQTT, Domotics, Internet of Things, WiFi, Event-Based Control.

I. INTRODUCCIÓN

La domótica es la aplicación más popular dentro del Internet de las cosas, se considera una casa inteligente o Home Automation, a aquel espacio en donde todos sus dispositivos están interconectados entre sí mediante internet, de ahí sale el concepto de Internet de las Cosas, debido a que cualquier dispositivo eléctrico puede conectarse a través de la nube, esto permite el acceso a muchas herramientas que con el paso del tiempo se han ido desarrollando, como plataformas domóticas [1].

Una plataforma domótica es el complemento perfecto dentro de un sistema domótico, ya que esta permite la interoperabilidad de casi todos los dispositivos inteligentes, además de poder monitorearlos y controlarlos a través de sistemas que el usuario puede desarrollar, también cuenta con acceso remoto, esto lo hace operable desde cualquier parte del mundo, solo se necesita una conexión a internet estable [1].

En este contexto han surgido una serie de protocolos de comunicación que dan soporte a las necesidades anteriormente descritas y encajan en el paradigma del IoT, tales como MQTT

(Message Queue Telemetry Transport). MQTT es un protocolo de mensajería asíncrona que disocia al emisor y al receptor tanto en espacio como en tiempo. Utiliza un modelo de publicación y suscripción, el cual es no-bloqueante, lo que permite que no sea necesaria una red muy fiable. Es un protocolo liviano y flexible, lo que hace que pueda implementarse en dispositivos con recursos limitados y se adapte a situaciones con diferente demanda de recursos. La comunicación es posible gracias a un intermediario de mensajes entre los distintos clientes, que puede ser cualquier cosa capaz de enviar o recibir mensajes. El cliente publica un mensaje en un tema o topic, que se lo envía al intermediario, o también llamado broker, el cual redirige este mensaje al cliente o clientes que estén suscritos a ese topic. Puesto que los mensajes están organizados por temas proporciona una jerarquía a la estructura del sistema [2].

Es por lo mencionado en la presente introducción, que se va a desarrollar un sistema domótico basado en IoT, utilizando como protocolo de comunicación: MQTT.

II. MATERIALES Y EQUIPOS

- Fuente de voltaje DC
- Protoboard
- Conectores macho-macho, macho-hembra.
- Diodos Led
- Multímetro
- Arduino Uno o ESP32
- Sensor DHT11
- Módulo Relé
- Ventilador a 5V

III. PROCEDIMIENTOS

Diseño e implemente un sistema de domótico de apertura de una puerta, control de temperatura y notificación por medio de correo, la figura 1 expone el sistema en general aplicando el protocolo de comunicación MQTT e implementado un repositorio en GitHub para subir los archivos que contienen el contenido del laboratorio.

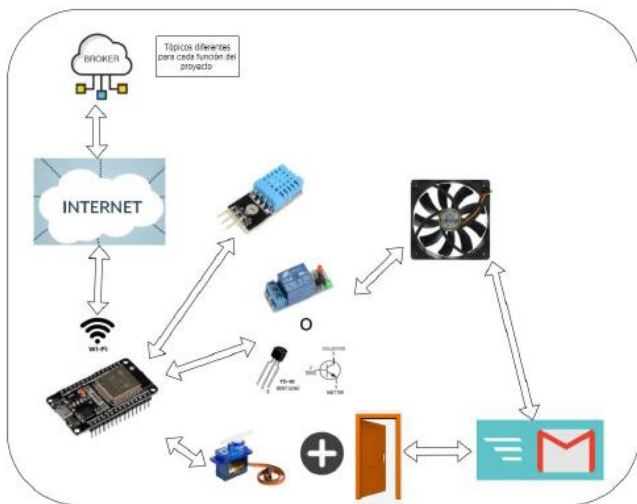


Fig. 1. Sistema de control y monitoreo de temperatura de una incubadora.

Para el desarrollo de la práctica de laboratorio se implementó el sistema señalado en la figura 1. Para lo cual el desarrollo de la práctica se realizó en las siguientes fases:

1. Implementación del montaje físico en la protoboard

Para realizar las respectivas conexiones en la protoboard se debe definir el sistema funcional del proyecto el cual se plantea de la siguiente manera: El sistema domótico está conformado por el sensor de temperatura DHT11, actuador1: ventilador, actuador2: servomotor. El sensor DHT11 informará acerca del estado de la temperatura y humedad actual en el ambiente, si el sistema detecta un aumento elevado en la temperatura, se encenderá un led testigo (informa que se ha sobrepasado el umbral de temperatura asignado) y el ventilador se activará para bajar la temperatura del ambiente. En segundo lugar el servomotor en posición de 0° mantendrá la puerta del sistema domótico cerrada, al realizar un giro de 90° el servomotor abrirá la puerta del sistema.

La alimentación del sistema la proporciona el módulo ESP32 (3.3V) y GND. El pin 22 alimenta al led testigo de temperatura, el sensor DHT11 está comunicado al pin digital 4, el servomotor recibe alimentación del módulo y se comunica mediante el pin 23, el ventilador recibe alimentación a través de un relé que a su vez se alimenta del módulo ESP32; la comunicación entre el ventilador y la ESP32 se realiza por el pin 22. El montaje con las respectivas conexiones se presenta en la figura 2.

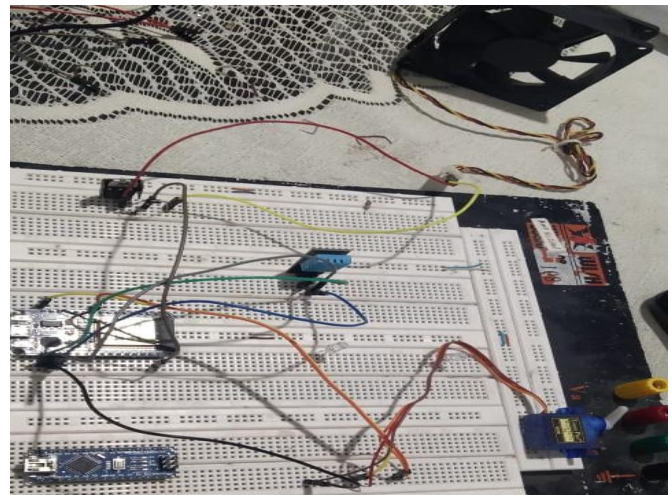


Fig. 2. Montaje inicial para verificar conexiones y realizar pruebas a los dispositivos.

2. Programación del Módulo ESP32

La programación del módulo ESP32 se realiza mediante el software arduino, a continuación se presenta la programación de las principales variables.

En la figura 3 se declaran las variables correspondientes a cada uno de los tópicos del bróker, de igual manera se define la cuenta en Gmail para recibir las notificaciones de alerta de los bróker.

```

//***** MQTT CONFIG *****
//***** MQTT CONFIG *****
//***** MQTT CONFIG *****
const char *mqtt_server = "ioticos.org";
const int mqtt_port = 1883;
const char *mqtt_user = "lkHt4mdQ7o690FZ";
const char *mqtt_pass = "S9rgQGDHwEu4MK";
const char *root_topic_subscribe = "284KB86PZde5n2L/input";
const char *root_topic_publish = "284KB86PZde5n2L/output";

const char *root_topic_subscribe_door = "284KB86PZde5n2L/door";
const char *root_topic_subscribe_modifyTemp = "284KB86PZde5n2L/modifyTemp";
const char *root_topic_publish_data = "284KB86PZde5n2L/data";
const char *root_topic_publish_temp = "284KB86PZde5n2L/temp";
const char *root_topic_publish_humidity = "284KB86PZde5n2L/humidity";

//Config mail
#define emailSenderAccount "ioticosafse90@gmail.com"
#define emailSenderPassword "alfasted90"
#define emailRecipient "casanareiot@gmail.com"
#define smtpServer "smtp.gmail.com"
#define smtpServerPort 465

String emailSubject = "ESP32 Test - PRIMER ENVIO";
String emailBodyMessage = "<div style='color:#2f4468;'><h1>Hello World!</h1><p>";

```

Fig. 3. Declaración de variables para la funcionalidad del bróker MQTT.

En la figura 4 se aprecia el ciclo condicional para activar la funcionalidad del actuador2: Servomotor, en este ciclo si el estado del servomotor es "1", este girará 90° accionando la apertura de la puerta, seguidamente llegara una notificación de correo electrónico informando que la puerta se encuentra abierta. Si el estado del servomotor es "0" el servomotor se encontrara en 0 grados de posición, por lo tanto la puerta permanecerá cerrada.

```

void readTopicDoor(){
  Serial.println(topic);
  if(topic!=""){
    String readTopic = topic.substring(16,20);
    Serial.println(readTopic);
    if(readTopic=="door"){
      if(body=="abrir"){
        Serial.println("readTopic -> ok");
        Serial.println(body);
        myservo.write(180);
        emailSubject = "ESP32 ALERTA - PUERTA ABIERTA";
        emailBodyMessage = "<div style='color:#2f4468;'><h1>Hola esto es una";
        ConfigSendMailTo();
        topic="";
        stateServo = 1;
      }
      if(body=="cerrar"){
        Serial.println("readTopic -> ok");
        Serial.println(body);
        myservo.write(1);
        topic="";
        stateServo = 0;
      }
      if(body=="enviarCorreo"){
        Serial.println("enviarCorreo -> ok");
        Serial.println(body);
        ConfigSendMailTo();
      }
    }
  }
}

```

Fig. 4. Ciclo condicional para accionar el giro del servomotor.

En la figura 5 se presenta el ciclo que modifica la variable temperatura, esto con el objetivo de definir un umbral de temperatura para realizar las pruebas con el ventilador. Si la temperatura sobrepasa el umbral definido se activa el ventilador, el cual se encargara de enfriar la habitación y mantener la temperatura ambiente.

Cabe mencionar que la variable primero realiza la lectura del estado de la temperatura, seguidamente si la temperatura se encuentra por debajo del umbral el ciclo finaliza e inicia uno nuevo. Si la temperatura sobrepasa el umbral se activara el ventilador y se envía

un mensaje de notificación al correo electrónico del usuario informando que el estado de la temperatura es alto y por lo cual se procede a ventilar la casa.

```

void readModifyTemp(){
  Serial.println(topic);
  if(topic!=""){
    String readTopic = topic.substring(16,26);
    Serial.println(readTopic);
    if(readTopic=="modifyTemp"){
      if(body.toInt()>=0){
        Serial.println("readModifyTemp -> ok");
        Serial.println(body.toInt());
        tempModify = body.toInt();
        topic="";
      }
    }
  }
}

void controllerTemp(){
  if(temperature>tempModify){
    Serial.println("Encendido Relay -> ok");
    digitalWrite(rele,HIGH);
    emailSubject = "ESP32 ALERTA - VENTILADOR ENCENDIDO";
    emailBodyMessage = "<div style='color:#2f4468;'><h1>Hola señor usuario,";
    if(banderaControllerTempMail == 0){
      ConfigSendMailTo();
      banderaControllerTempMail = 1;
    }
  }
  else{
    Serial.println("Apagado Relay -> ok");
  }
}

```

Fig. 5. Ciclo condicional que lee y controla el estado de la temperatura en la casa.

En la figura 6 se presenta el ciclo que realiza la lectura de las variables de temperatura y humedad cada 25 segundos, de igual manera se llaman a los tópicos los cuales mostraran en el bróker la lectura de la temperatura y humedad actual.

```

void tempPublish(){
  //My timer
  unsigned long topLoop = millis();
  // this is setInterval
  if (topLoop - lastPublishHumidityAll >= publishHumidityAll) {
    lastPublishHumidityAll = topLoop;
    String str = "La Temperatura es -> " + String(temperature) + "°C";
    str.toCharArray(msg, 30);
    client.publish(root_topic_publish_temp,msg);
    delay(200);
    //delay(25000);
  }
}

void humidityPublish(){
  //My timer
  unsigned long topLoop2 = millis();
  // this is setInterval
  if (topLoop2 - lastPublishTempAll >= publishTempAll) {
    lastPublishTempAll = topLoop2;
    String str = "La Humedad es -> " + String(humidity) + "%";
    str.toCharArray(msg, 30);
    client.publish(root_topic_publish_humidity,msg);
    str="";
    delay(200);
    //delay(25000);
  }
}

```

Fig. 6. Ciclo condicional de lectura del estado de la temperatura y humedad actual del sistema domótico.

3. Creación de los Tópicos en MQTT

A continuación se presentan los tópicos publicadores que realizan la lectura de las variables del sistema.

En la figura 7 se observa la terminal principal del Usuario Publicador.

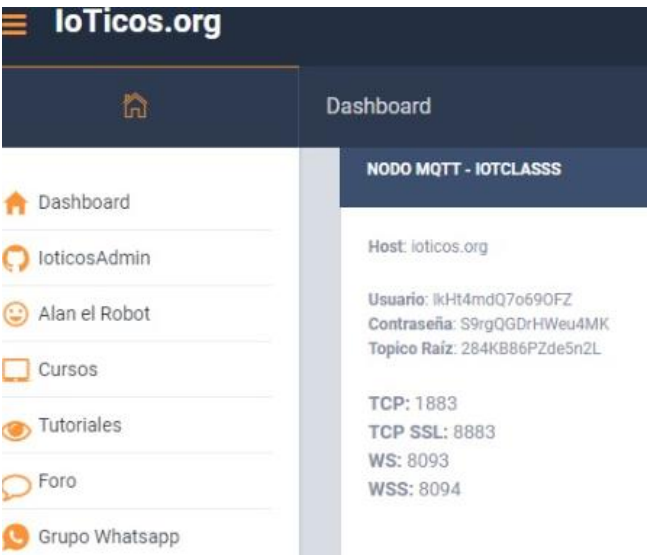


Fig. 7. Broker MQTT de la terminal cliente-publicador.

Tópico Raíz: 284KB86PZde5n2L

Tópico principal: 284KB86PZde5n2L/data

Informa acerca de la información del sistema actual: temperatura, humedad y si la puerta está abierta o cerrada.

Tópico Puerta: 284KB86PZde5n2L/door

Informa acerca del estado de puerta cerrada o abierta. Cuando se publica desde el bróker a este tópico, se acciona el servomotor, el cual realiza la apertura o cierre de la puerta.

Tópico Temperatura: 284KB86PZde5n2L/temp

Informa acerca de la temperatura actual actualizada cada 25

Tópico Humedad: 284KB86PZde5n2L/humidity

Informa acerca de la humedad actual actualizada cada 25 segundos

Tópico para modificar Temperatura:
284KB86PZde5n2L/modifyTemp

Cuando se publica desde el broker a este tópico permite configurar el sistema de control de temperatura.

4. Pruebas funcionales del sistema domótico

A continuación se presenta la funcionalidad del sistema domótico a través del broker MQTT.

En la terminal principal del broker de la plataforma ioticos se observa la lectura de la variables temperatura, humedad y estado de

la puerta. Este sistema es visualizado al publicar el topico 284KB86PZde5n2L/data.

Seguidamente cuando se publica en el broker el topico 284KB86PZde5n2L/modifyTemp se procede a modificar la temperatura para accionar el ventilador. Finalmente cuando se publica en el broker el topico 284KB86PZde5n2L/door se procede a accionar el servomotor para abrir y cerrar la puerta. La respuesta de la publicación de los tópicos en la terminal se evidencia en la figura 8.

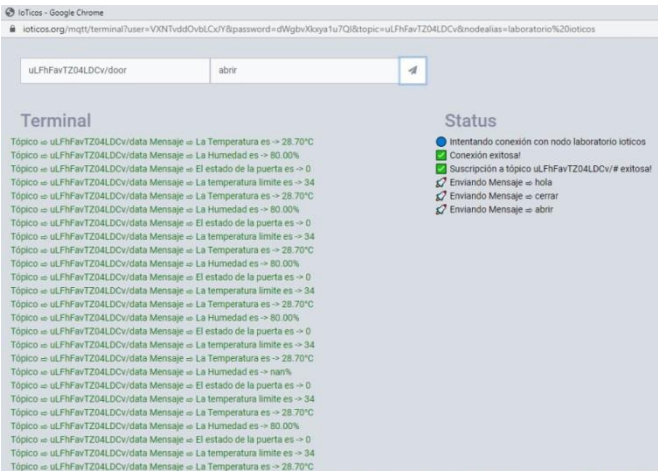


Fig.8 Terminal bróker cliente-Publicador con la respuesta a los tópicos.



Fig.9. Puerta cerrada – Posición 0° del servomotor – Tópico 284KB86PZde5n2L/door



Fig.10. Puerta abierta – Posición 90° del servomotor – Tópico 284KB86PZde5n2L/door publicado



Fig.11. Ventilador accionado para bajar el umbral de temperatura – Tópico 284KB86PZde5n2L/door publicado.

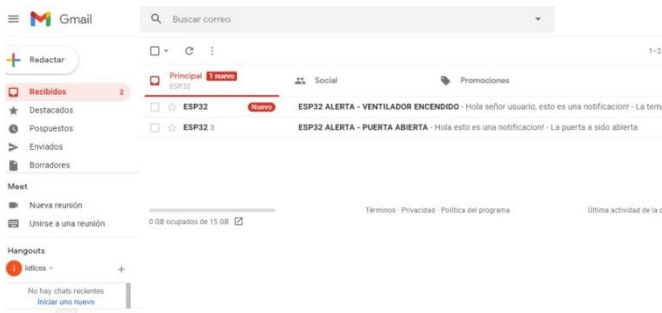


Fig.12. Correos de alerta señalando el aumento de la temperatura y el estado de la puerta.

5. Repositorio en GitHub

A continuación se comparte el link de la cuenta en GitHub del repositorio que contendrá los archivos correspondientes al laboratorio 3 de la asignatura Internet de las cosas.

Link: https://github.com/Faisuly-Castro/Repositorio_lab3_IOT.git

IV. CONCLUSIONES

- El desarrollo del presente laboratorio nos permitió desarrollar un sistema domótico. Este sistema utiliza el protocolo MQTT para coordinar de forma asíncrona el funcionamiento de varios dispositivos para conseguir una solución de control de sensores y actuadores de una casa que plantea las bases de un control domótico más complejo. El protocolo MQTT permite a través de la publicación de tópicos visualizar y modificar las variables de estado proporcionadas por los sensores y actuadores. El bróker cuenta con un tópico raíz público, a través de él generan los tópicos publicadores auxiliares: como por ejemplo el tópico 284KB86PZde5n2L/modifyTemp que nos permite modificar el umbral de temperatura. La funcionalidad del bróker depende del estado activo de la plataforma en la que se creó el bróker, en nuestro caso fue ioticos.org; si la plataforma se encuentra caída o con problemas de conectividad no será posible trabajar con el bróker a nivel local y WAN.
- Para el desarrollo del laboratorio se creó un repositorio con la plataforma GitHub a partir de las practicas vistas en clase, por lo cual con la creación de este repositorio se facilita el sistema para compartir y almacenar archivos a través de internet haciéndoles accesibles para todo público y en todo momento, salvo en excepciones donde se agregue o modifique archivos en el repositorio local y no se realice la debida actualización de las ramas del repositorio en la nube.
- Las plataformas IoT son una herramienta muy importante ya que con ellas se puede implementar códigos y toma de datos en tiempo real, donde al utilizar los demás componentes que estas proporcionan se puede llegar a crear proyectos muy beneficiosos para la humanidad. Obteniéndose resultados satisfactorios y siendo tomada más en cuenta en el presente para almacenamiento en la nube y control de sistemas a través de internet.

BIBLIOGRAFÍA

- [1] C. A. Tinoco Alejandro, «Diseño e implementación de un sistema domótico basado en iot,» 21 Diciembre 2020. [En línea]. Available: <http://repositorio.utmachala.edu.ec/handle/48000/16089>. [Último acceso: 2021 Mayo 16].
- [2] J. De la cal calleja, «Control Domótico basado en el protocolo MQTT,» Julio 2019. [En línea]. Available:

<https://core.ac.uk/download/pdf/228074048.pdf>. [Último acceso: 2021 Mayo 16].