

Control de cámara de germinación hidropónica mediante IoT, broker Mosquitto y placas SBC en invernaderos de la provincia de Misiones

Marcelo Marinelli, Myriam Kurtz, Rubén R. Urquijo, Guillermo Wurm.
Instituto de Biotecnología de Misiones "María Ebe Reca". Campus U.Na.M. Ruta
12 Km 7,5, Miguel Lanús, Posadas, Misiones, Argentina.
{marcelo, myriam, chinourquijo, guillermow77}@gmail.com

Resumen. Se presenta la implementación de un sistema de control automático de sistemas hidropónicos en invernaderos de la provincia de Misiones. Para este propósito se utilizó tecnología IoT, hardware de bajo costo con placas tipo SBC Raspberry Pi y NodeMCU. En la placa Raspberry Pi se instaló un *broker* Mosquitto, el cual controla paquetes de datos MQTT, éste es un protocolo específico para IoT, ideal para trabajar con bajo ancho de banda y provee un sistema de seguridad con tres calidades de servicio. Con el programa NODE-RED se programó la administración de cuatro nodos NodeMCU que se encargan de administrar la lógica del control automático de los sistemas hidropónicos. También en NODE-RED se realiza la presentación de datos en un tablero de control, acceso a redes sociales, a repositorios virtuales y se envían los datos a una web de ThingSpeak para su visualización en tiempo real.

Palabras claves: IoT, Raspberry Pi, NodeMCU, MQTT, NODE-RED.

1. Introducción

Los cultivos hidropónicos son cada vez más utilizados por sus propiedades sanitarias debido a la no contaminación por contacto con el suelo, como los cultivos tradicionales, esta característica evita que los cultivos estén en contacto con bacterias, parásitos u otro tipo de contaminación. Esta clase de cultivos es conveniente realizarlo en dos etapas: la primera, es la germinación de las semillas hasta que se desarrolle un plantín de determinada característica, según el tipo de especie, para ello se desarrolló en [1] una cámara de germinación con sistema de control difuso. Luego, en la segunda etapa, se trasplantan los platines a otro contenedor que puede ser por sistema de raíz flotante o por sistema NFT (*Nutrient Film Technique*) [2], [3] para su desarrollo definitivo. Los emprendimientos de hidroponía de la provincia de Misiones se realizan en invernaderos para controlar las variables ambientales como humedad, temperatura y radiación solar.

En este trabajo se presenta la implementación de un sistema de control mediante IoT (*Internet of Things*) y dispositivos embebidos de bajo costo del tipo SBC (*Single Board Computer*).

Se utiliza un acceso a internet con router WiFi que provee de una cobertura en toda la superficie de trabajo de un invernadero tipo, que es de aproximadamente 1500 m². De esta forma se vinculan dispositivos inalámbricos de sensado y control de variables en distintos puntos del invernadero.

1.1 Hardware de control

Se realiza el sensado y control de dos subsistemas de cultivos hidropónicos: en las cámaras de germinación y en los sistemas NTF. Para esto se utilizan módulos NodeMCU [4] que son dispositivos de código abierto desarrollado para IoT basado en el circuito integrado ESP8266 de ESPRESSIF [5],[6], es un firmware basado en el lenguaje multiparadigma denominado Lua (Luna es portugués). Esta placa de desarrollo tiene conexión a WiFi, 13 puertos GPIO y una entrada analógica entre otras. Existen distintos fabricantes, ya que es de código abierto. Permite protocolos SPI, I²C, I²S con DMA, UART únicamente para transmisión que puede habilitarse a través del pin GPIO2, tiene dos leds indicadores y una fuente convertidora de continua a continua de 5 V a 3.3 V. El WiFi es compatible con el protocolo 802.11 b/g/n y soporta de autenticación WEP y WPA/WPA2 (Figura 1).

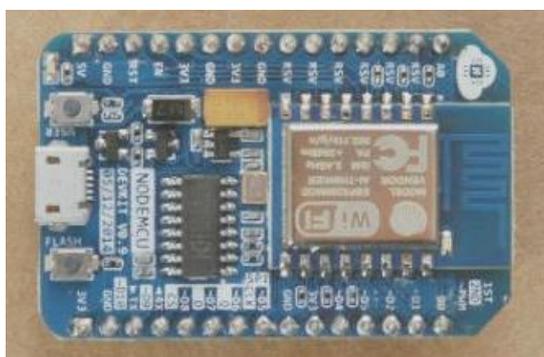


Figura 1. Esquema del NodeMCU[4].

Para la programación del módulo se utilizó el IDE Arduino, configurado con la placa NodeMCU 1.0 (ESP-12E Module) como indica la Figura 2.

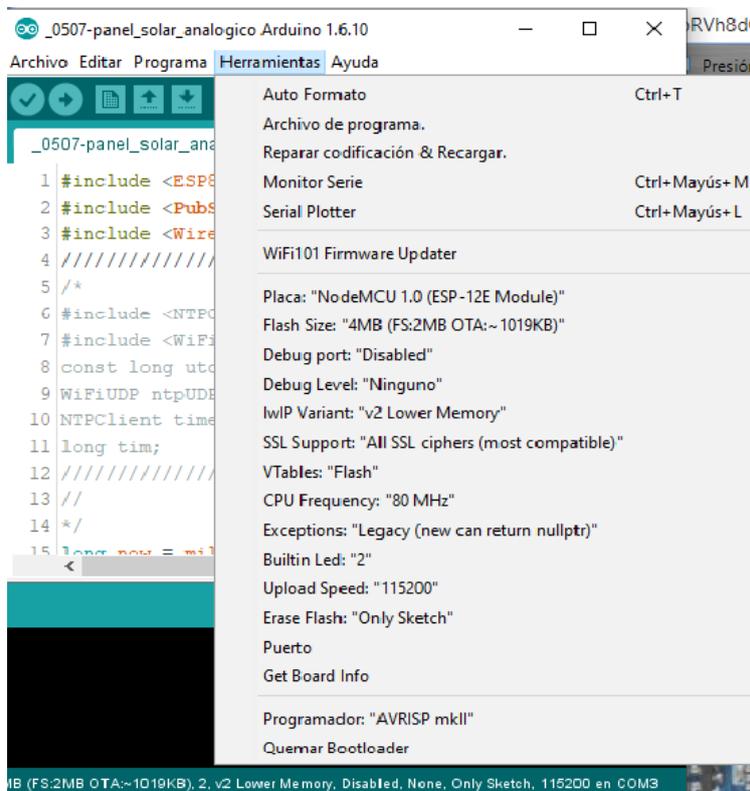


Figura 2. Captura de pantalla de la configuración del IDE Arduino.

1.2 Placa Raspberry Pi

Es una placa del tipo SBC (*Single Board Computer*), de bajo costo y soporta distintos sistemas operativo: Linux ARM (Debian, Fedora, Arch Linux), RISC OS2, Windows 10 IoT Core, entre otros. Su tamaño es de 85 mm de longitud por 56 mm de ancho. Requiere alimentación de 5 V y tiene una salida HDMI, una de audio y video compuesto, 4 puertos USB y un conector Ethernet. Tiene una CPU ARM1176JZF-S (armv6k) a 700 MHz, GPU Broadcom VideoCore IV3, memoria RAM de 512 MBytes, capacidad de almacenamiento por medio de Tarjeta SD o SDHC (Figura 3 [6]).

En el conector de expansión posee puertos GPIO (*General Purpose Input/Output*), los cuales representan la interfaz entre la Raspberry Pi y el exterior [7], en total son 17 puertos programables como entrada o salida. Los puertos utilizados son GPIO 11, 12, para sensores de humedad y temperatura y 25 como entrada digital para la obtención del sensor de nivel de nutrientes y el puerto 18 como salida para activar la bomba.

1.3 Sensores

Como sensores de humedad y temperatura se utilizó el DHT22, que posee un rango de operación de 0 a 100% HR y de -40 a 80°C y cuenta con una precisión de humedad de 2% RH y de temperatura de 0,5% (Figura 4. a.).

Para medir la temperatura de la solución de nutrientes se empleó el DS18B20 (Figura 4.b.), que posee encapsulado metálico y puede sumergirse sin alterar la

medición. Para el sensado de nivel, fue instalado un sensor con flotante y actuador magnético (Figura 4.c.).



Figura 3. Raspberry Pi 2 modelo B [6].

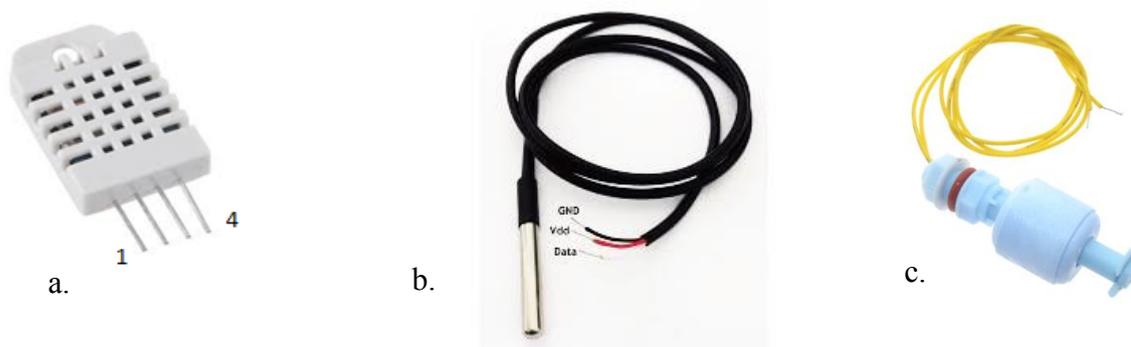


Figura 4. Sensores: a) DHT22 b) DS18B20 c) Nivel magnético.

2. Protocolo MQTT (Message Queue Telemetry Transport)

Es un protocolo, desarrollado por IBM y Arcom en 1999, de mensajes del tipo publicación/suscripción para trabajar con redes de bajo ancho de banda y alta latencia, lo que lo hace ideal para trabajo en entornos “*machine-to-machine*” (M2M) o “*Internet of Things*” [8]. Debido a sus características, este protocolo es ideal para entrega de paquetes de telemetría en forma confiable, en Versiones 3.1 o superior puede usarse nombre de usuario y contraseña en paquetes MQTT [9].

Los mensajes constan de tres partes: el encabezado fijo de 2 bytes, el encabezado variable de 4 bytes que puede no estar y la carga útil que puede tener hasta un máximo de 256 Mb. Tiene tres opciones de calidad de servicio [10], [11]: QoS 0: At most once, QoS 1: At least once y QoS 2: Exactly once.

Para la implementación del protocolo se necesita un “*broker*”, este administra la comunicación entre los dispositivos que se suscriben para recibir mensajes y los que publican mensajes, cada uno de estos se identifican por un “*topic*”.

En el ejemplo de la Figura 5 se utiliza el tópico camara para recibir y transmitir mensajes entre el cliente, en este caso cámara I y el *broker*.

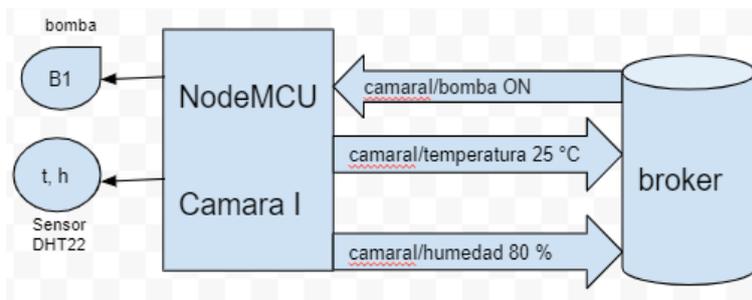


Figura 5. Esquema de intercambio de mensajes MQTT.

A nivel de protocolo, para iniciar la comunicación de un cliente con el *broker*, este envía un paquete de control denominado CONNECT, el *broker* contesta con un paquete CONNACK, así se establece un enlace TCP/IP utilizando el puerto 1883 (si utiliza TLS el 8883). El enlace se mantiene hasta la desconexión del cliente. Luego el cliente puede enviar un paquete PUBLISH que contiene el tópico y el contenido. En nuestro ejemplo la cámara I se suscribe con el tópico *camaraI* y envía un mensaje con el valor de la variable humedad, luego la variable temperatura y recibe del *broker* el mensaje para encender la bomba B.

camaraI/humedad 85%

camaraI/temperatura 20°C

El *broker* también puede enviar mensajes al cliente mediante el mensaje PUBLISH, por ejemplo si pretendemos enviar un mensaje para activar la bomba de la cámara I debe ser:

camaraI/bomba ON

3. Broker Eclipse Mosquitto

Mosquitto forma parte de la Fundación Eclipse, es un proyecto cuyo sitio web es iot.eclipse.org [12], [13]. Es un agente de mensajes de código abierto que implementa las versiones 5.0, 3.1.1 y 3.1 del protocolo MQTT.

En este proyecto se utilizó el *broker* Mosquitto instalado en una placa Raspberry Pi. Como está disponible en el repositorio, se instala con el comando:

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

En caso contrario, puede descargarse del repositorio: <https://mosquitto.org/blog/2013/01/mosquitto-debian-repository/>.

Una vez instalado en la Raspberry, Mosquitto se puede probar por consola los siguientes comandos: *mosquitto_sub* y *mosquitto_pub*, el primero permite suscribirse a un topic del *broker*.

```
mosquitto_sub -h BROKER -t TOPIC (ej. mosquitto_sub -h localhost -t camaraI)
```

El segundo permite publicar mensajes en el *broker* sobre un *topic* determinado y los suscriptores del mismo *topic* lo van a recibir.

mosquitto_pub -h BROKER -t TOPIC -m MENSAJE (ej.
 mosquitto_pub -h localhost -t camarai -m "temperatura =
 25 °C")

En la Figura 6 se aprecian la captura de pantalla del ejemplo anterior.

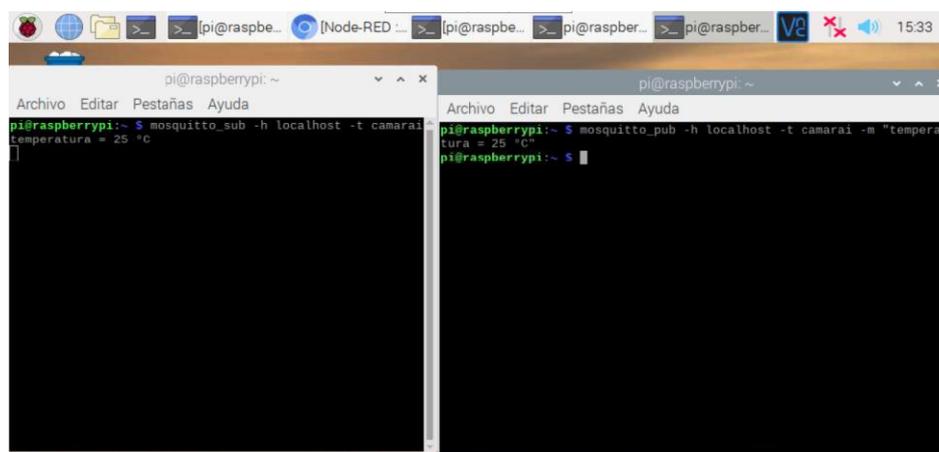


Figura 6. Captura de pantalla del ejemplo.

4. Topología del sistema de control y telemetría de un invernadero en la provincia de Misiones

Debido a las características del invernadero es necesario controlar dos cámaras de germinación, un sistema NFT y los servos de apertura y cierre de las ventanas para recirculación de aire desde el exterior y un sistema de cobertura (media sombra) para disminuir la radiación solar en horas donde el sol se encuentra perpendicular al invernadero Figura 7.

Para el software de control se utilizó NODE-RED, que es una herramienta de programación basada en web para IoT que permite manejar hardware utilizando nodos específicos para cada dispositivo. Es de código abierto, desarrollado por IBM y trabaja con nodos que representan APIs de web, servicios de redes sociales o dispositivos de hardware, se unen mediante conexiones que representan el flujo de datos [13].

En la versión utilizada del sistema operativo Raspbian usado para la Raspberry Pi ya viene instalado. Posee una interface web del tipo tablero de control en donde se pueden representar los datos y controles en forma gráfica.

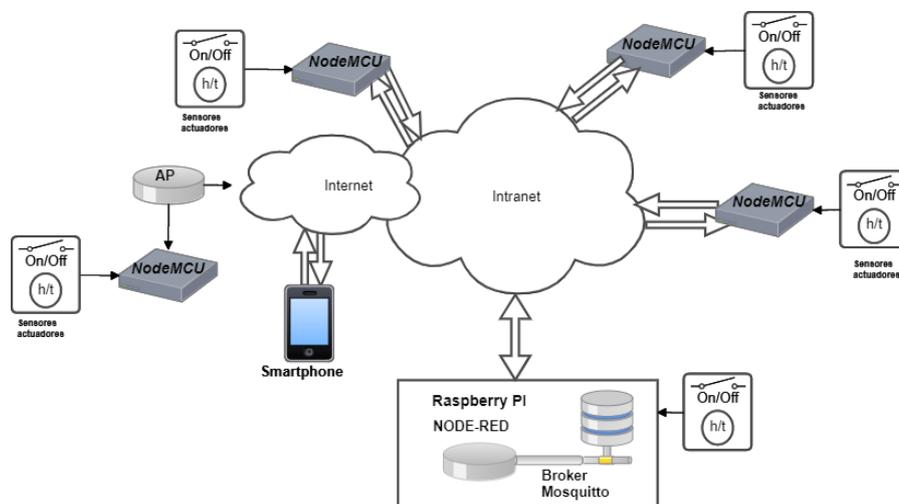


Figura 7. Topología del sistema de control de invernadero.

La placa Raspberry Pi, en la que se instaló el *broker* Mosquitto, también tiene las siguientes funciones:

- 1) Controla una de las cámaras de germinación mediante un sensor de humedad y temperatura DHT22 (Figura 8), un sensor de temperatura de nutrientes DS18B20 (Figura 9), sensor de nivel de nutrientes y control de la bomba de circulación.
- 2) Almacena los datos en archivos de texto plano.
- 3) Envía a un disco virtual Dropbox los archivos de datos de todas las variables con la fecha y la hora de sensado.
- 4) Envía a ThingSpeak (ver el apartado 4.1) los datos para almacenamiento y visualización en la web.
- 5) Envía los datos a la interfaz Dashboard para visualizar los datos en formato tipo Gage y Chart.

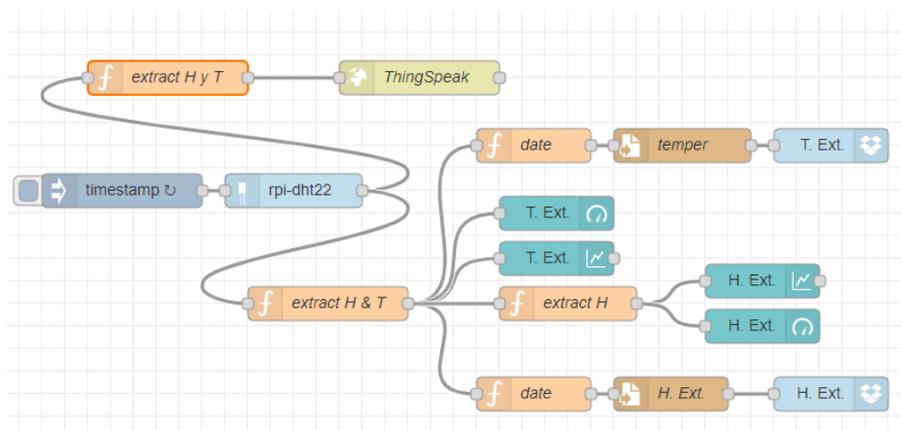


Figura 8. Control del sensor DHT22 de la cámara I.

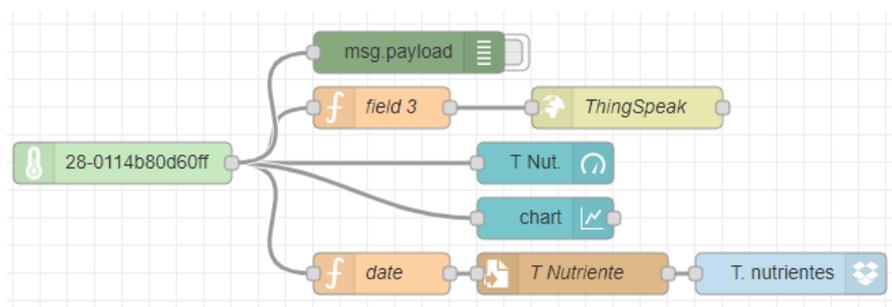


Figura 9. Control del sensor ds18b20 de la cámara I.

4.1 ThingSpeak

Es una plataforma web desarrollada para almacenamiento de datos en la nube para IoT. También permite realizar distintos tratamientos de datos con Matlab. Puede accederse en forma gratuita en <https://thingspeak.com/>.

Para enviar los datos a un canal de ThingSpeak debe programarse un nodo función con los datos del canal y un token que proporciona la cuenta. En el siguiente ejemplo, se envían los datos que llegan al nodo proveniente de un sensor y se reemplazan las X por el token. Luego se conecta con un nodo HTTP para que realice la función GET y envíe el dato correspondiente.

```
var field3 = msg.payload;
msg.url="https://api.thingspeak.com/update?api_key=XXXXXXXXXXXX&
field3="+field3;
return msg;
```

La representación de los datos se realiza en tiempo real, puede apreciarse en la Figura 10.

4.2 Nodos NodeMCU

Cada nodo "NodeMCU" de la Figura 7, está compuesto por los sensores descritos anteriormente con la diferencia que el conexionado se hace por los puertos GPIO del dispositivo denominados D4 con la librería `DallasTemperature.h` para el DS18b20, D3 para el DHT22 con la librería `DHT.h`, D5 para el sensor de nivel de nutrientes y D6 para el control de la bomba.

Para la conexión WiFi, se programan los datos del AP: nombre del ESID y la password, para la conexión con el *broker* se indica la dirección IP del mismo. De esta manera y mediante la librería `PubSubClient.h`, se pueden enviar y recibir paquetes MQTT:

```
client.subscribe("invernadero/camaraiii");

client.publish("invernadero/camaraiii", temperatura)
```

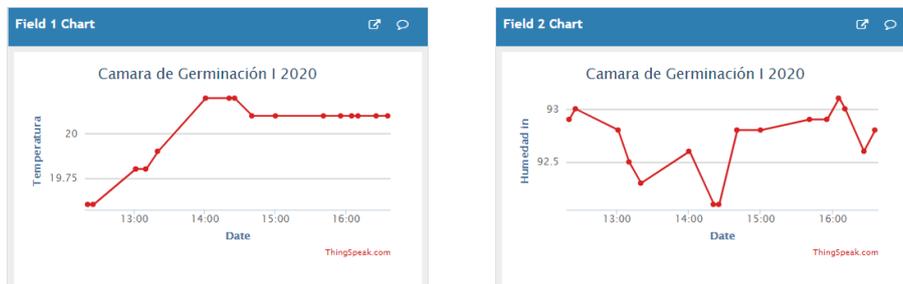


Figura 10. Captura de pantalla de los gráficos de ThingSpeak.

4.3 Control automático y mensajería NODE-RED

El *broker* recibe los datos, de los nodos NodeMCU, de cada sector del invernadero (Figura 11), los almacena en archivo de texto plano y en la nube con Dropbox, los presenta en forma gráfica en *Dashboard* e ingresan a un nodo de control que, en función de los datos obtenidos, produce señales de control que por medio de los nodos de salida MQTT envían al respectivo NodeMCU. Con respecto al nivel, la señal recibida es del tipo binaria y en caso de que el nivel esté bajo, envía un mail al administrador y activa señales sonoras y de texto en el tablero de control.

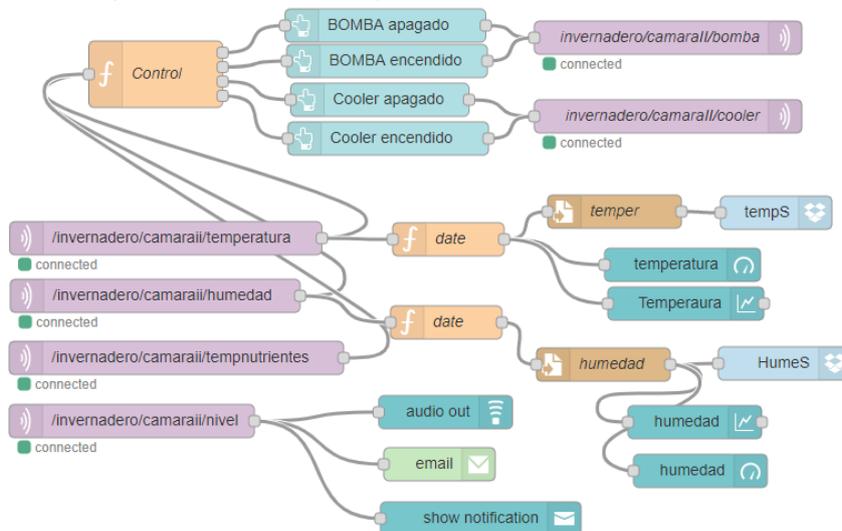


Figura 11. Control de la cámara II con los datos recibidos por MQTT desde un NodeMCU.

5. Conclusiones

Se realizó la implementación de un sistema de control para invernaderos de la provincia de Misiones utilizando placas SBC de bajo costo Raspberry Pi y prototipos NodeMCU basados en ESP8266 conectados a un WiFi, utilizando protocolo MQTT con un *broker* Mosquitto. Se diseñó la conexión de los sensores a cada placa y se programó el software embebido en cada caso. Mediante NODE-RED se administran los datos representándolos en un tablero de control y se los almacena en archivos locales y en la nube, utilizando Dropbox y ThingSpeak.

En la topología se aprecia un acceso a un proveedor de Internet, lo que permite visualizar el tablero de control desde un Smartphone o una PC.

6. Referencias

1. Marinelli, M., Acosta, N., Toloza, J. M., & Kornuta, C. (2017). Fuzzy Control of a Germination Chamber. *Journal of Computer Science and Technology*, 17(01), 74-78.
2. Hernández, C. J., & Hernández, J. L. (2005). Valoración productiva de lechuga hidropónica con la técnica de película de nutrientes (nft). *Naturaleza y Desarrollo*, 3, 1, 11-16.
3. Garzón López, S. S., & Arévalo de Gauggel, G. (2006). *Evaluación del rendimiento de tres variedades de lechuga bajo el sistema NFT (nutrient film technique) de hidroponía con dos soluciones de nutrientes* (No. T2244). ESCUELA AGRÍCOLA PANAMERICANA.
4. NodeMCU Connect Things EASY, http://www.nodemcu.com/index_en.html [Ultimo acceso 8/6/2020].
5. Espressif System ESP8266 Series, <https://www.espressif.com/> [acceso 8/6/2020].
6. Raspberry Pi Model 2B <https://www.raspberrypi.org> [acceso 10/6/2020].
7. Jain, S., Vaibhav, A., & Goyal, L. (2014, February). Raspberry Pi based interactive home automation system through E-mail. In *Optimization, Reliability, and Information Technology (ICROIT), 2014 International Conference on* (pp. 277-280). IEEE.
8. Ceja, J., Renteira, R., Ruelas, R., & Ochoa, G. (2017). Módulo ESP8266 y sus aplicaciones en el internet de las cosas. *Revista de Ingeniería Eléctrica*, 24-36.
9. Protocolo MQTT (Message Queue Telemetry Transport), <http://mqtt.org> [acceso 6/6/2020].
10. Defossé, N., López, R. A., Gómez, M. E., Konstantinoff, P., Wahler, S. P., Castro, L., & Harris, G. (2017, August). Implementación de middleware publicador/subscriptor para aplicaciones web de monitoreo. In *XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires)*.
11. Naik, N. (2017, October). Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. In *2017 IEEE international systems engineering symposium (ISSE)* (pp. 1-7). IEEE.
12. Eclipse Mosquitto™. An open source MQTT broker, <https://mosquitto.org> [Ultimo acceso 10/9/2020].
13. Open Source for IoT, <https://iot.eclipse.org>, [Ultimo acceso 10/9/2020].
14. Marinelli, M., & Urquijo, R. (2017). Sistema de Control de una cámara de germinación hidropónica con IoT. In *XXIII Congreso Argentino de Ciencias de la Computación (La Plata, 2017)*.
15. Chanthakit, S., & Rattanapoka, C. (2018, July). Mqtt based air quality monitoring system using node MCU and node-red. In *2018 Seventh ICT International Student Project Conference (ICT-ISPC)* (pp. 1-5). IEEE.