

Implementación de SCADA en sistema de bombas domesticó

Grupo Automatizacion

20 de agosto de 2019

1. Objetivo

El objeto de este proyecto inicial es familiarizarse con las distintas tecnologías, tanto de software como de hardware, que la compañía utilizara para implementar soluciones de IoT a nivel industrial.

2. Tecnologías

2.1. Hardware

2.1.1. NodeMCU

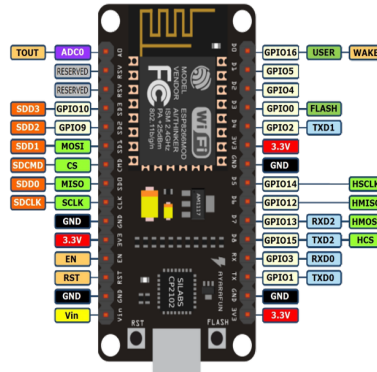
Se utilizara un PLC de NodeMCU, el cual cuenta con un modulo de WiFi, 10 pines de GPIO y un firmware hecho para implementar soluciones con IoT.

Especificaciones:

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1

- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 Mhz
- Wi-Fi: IEEE 802.11 b/g/n

Figura 1: pinout del PLC de NodeMCU



2.1.2. Raspberry π

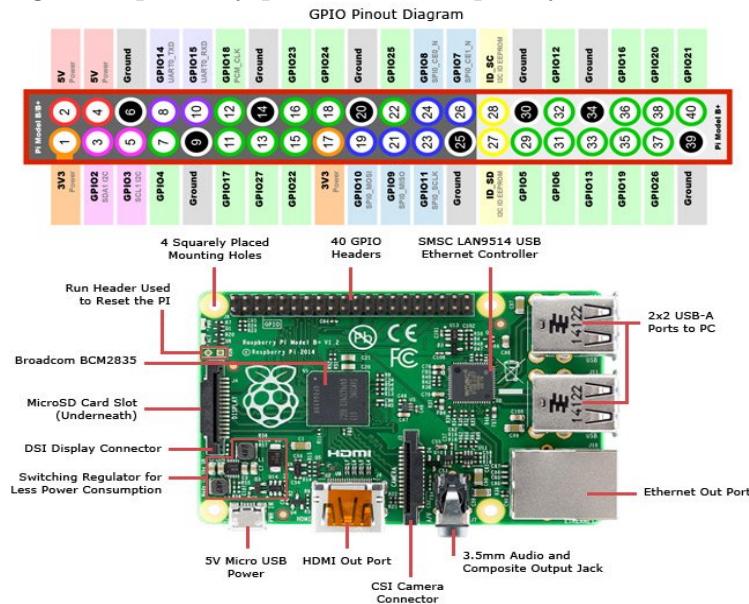
Se utilizara un Raspberry Pi model B+ con Ignition Edge como broker entre el NodeMCU y el servidor central de Ignition.

Especificaciones:

- Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header

- Full-size HDMI
- 4 USB 2.0 ports
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- 4-pole stereo output and composite video port
- Micro SD port for loading your operating system and storing data
- 5V/2.5A DC power input
- Power-over-Ethernet (PoE) support (requires separate PoE HAT)

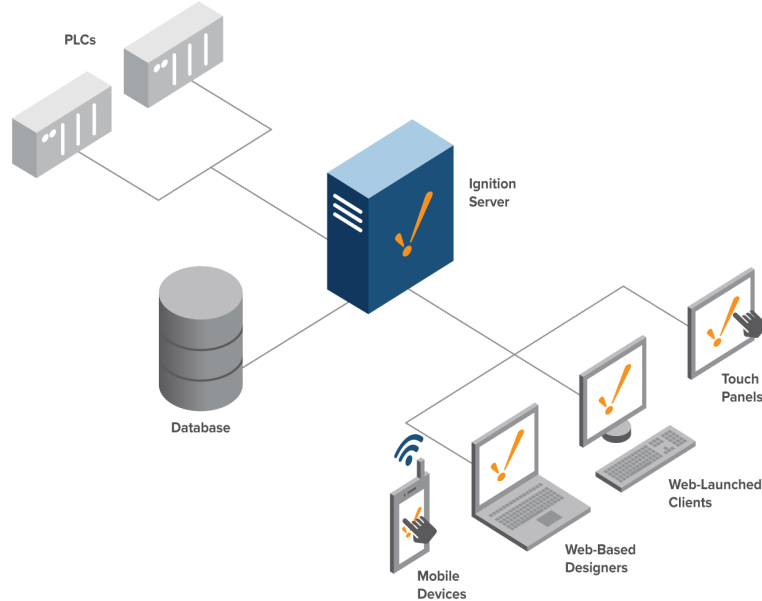
Figura 2: pinout y puertos del Raspberry Pi Model B+



2.2. software

En cuanto a software, se utilizara Ignition como SCADA, el cual tiene un modulo que implementa MQTT como protocolo de comunicaci3n entre el PLC y el servidor de SCADA.

Figura 3: Arquitectura estándar de Ignition



2.2.1. Instalación de Ignition Edge en raspberry pi:

Una guía detallada de como instalar Ignition Edge en el raspberry Pi puede encontrarse en:

<https://support.inductiveautomation.com/index.php?/Knowledgebase/Article/View/118/2/installation-ignition-edge-on-raspberry-pi>

Sin embargo, hay ciertos pasos que se tienen que modificar:

Paso 2 de la guía:

En la guía, el comando para hacer unzip del archivo descargado dice que hay que usar el comando:

```
sudo unzip ./Ignition-Linux-armhf-7.9.2 -d /usr/local/ignition
```

Este comando se debe reemplazar por:

```
sudo unzip ./Ignition-Linux-armhf-8.0.3 -d /usr/local/ignition
```

puesto la versión que se descargo (y que esta en el directorio que se tiene) es la 8.0.3, no la 7.9.2

Paso 4 de la guía:

En la guía, el comando que sale para hacer que los files sean ejecutables es:

```
sudo chmod +x ignition.sh ignition-gateway gcu.sh
```

sin embargo, el comando que se debe usar (sale en el archivo llamado README, el cual esta en la carpeta de Edge que se descargo) es:

```
sudo chmod +x *.sh
```

Paso 5 de la guía:

Si el comando:

```
sudo ./ignition.sh start
```

no funciona, se tiene que ganar privilegios de superuser, para lo cual se hace el comando:

```
sudo su
```

Seguido de:

```
./ignition.sh start
```

3. Objetivos

1. Lograr comunicación bidireccional entre el PLC de NodeMCU y el servidor de Ignition
2. Hacer análisis de data obtenida usando la plataforma de Ignition
3. Crear algoritmo que ajuste parámetros de funcionamiento automáticamente
4. Implementar sistema de alarmas en base a funcionamiento del sistema (ej, Esta entrando agua, etc)

4. Arquitectura a usar

Se utilizara una arquitectura simple donde el servidor central de Ignition sera un PC, seguido de un gateway que servirá como broker de datos el cual sera el raspberry pi el cual tendrá conexión con el PLC NodeMCU mediante el TCP Modbus o mediante MQTT.