

# Microcontrolador enviando datos a red local

## Índice

<b>Configuración de Node-red</b>	<b>6</b>
Instalación de node-red . . . . .	6
EJECUTAR NODE-RED . . . . .	6
INSTALACIÓN DE COMPLEMENTOS/MÓDULOS EN NODE-RED . . . . .	7
<b>Flujo de Node-red</b>	<b>7</b>
<b>Micropython y node-red (comunicación)</b>	<b>11</b>

!¿Enviar datos desde microcontrolador a red local con Micropython y Node Red

En este taller se aprenderá a compartir los datos de los sensores en una red Local como también dar órdenes a microcontroladores para que cambien el estado de algunos actuadores. En este ejemplo, los sensores son simulados por funciones de generación de datos aleatorios, por tanto, se invita a los estudiantes que después de comprender el funcionamiento de este ejemplo modifiquen el ejercicio agregando sensores y actuadores que han manejado en el transcurso del curso.

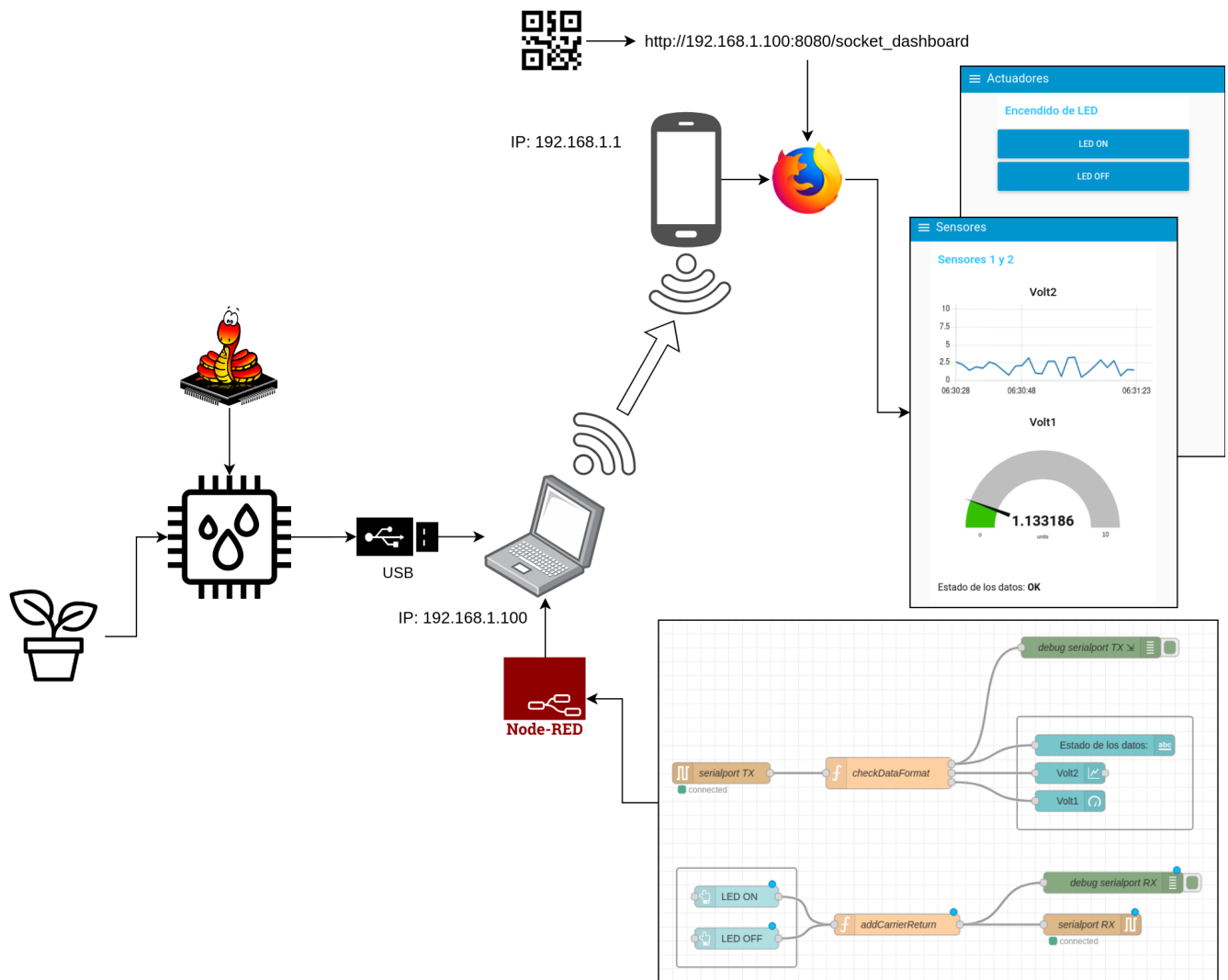
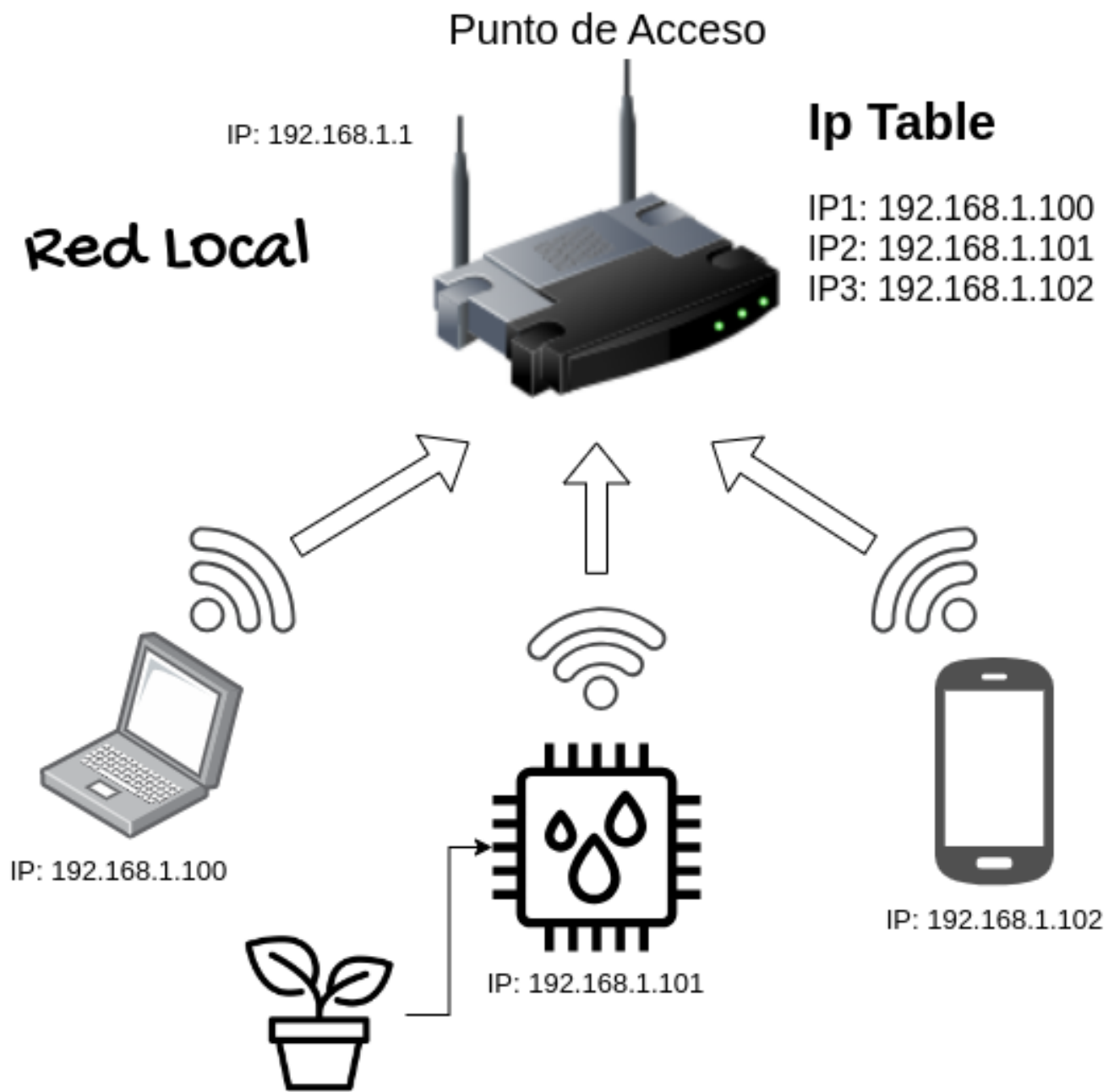


Figura 1: Envio de datos desde microcontrolador con micropython a navegador en red local

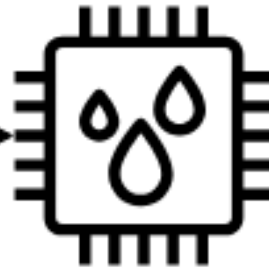


Red Local

IP: 192.168.1.1



IP: 192.168.1.100

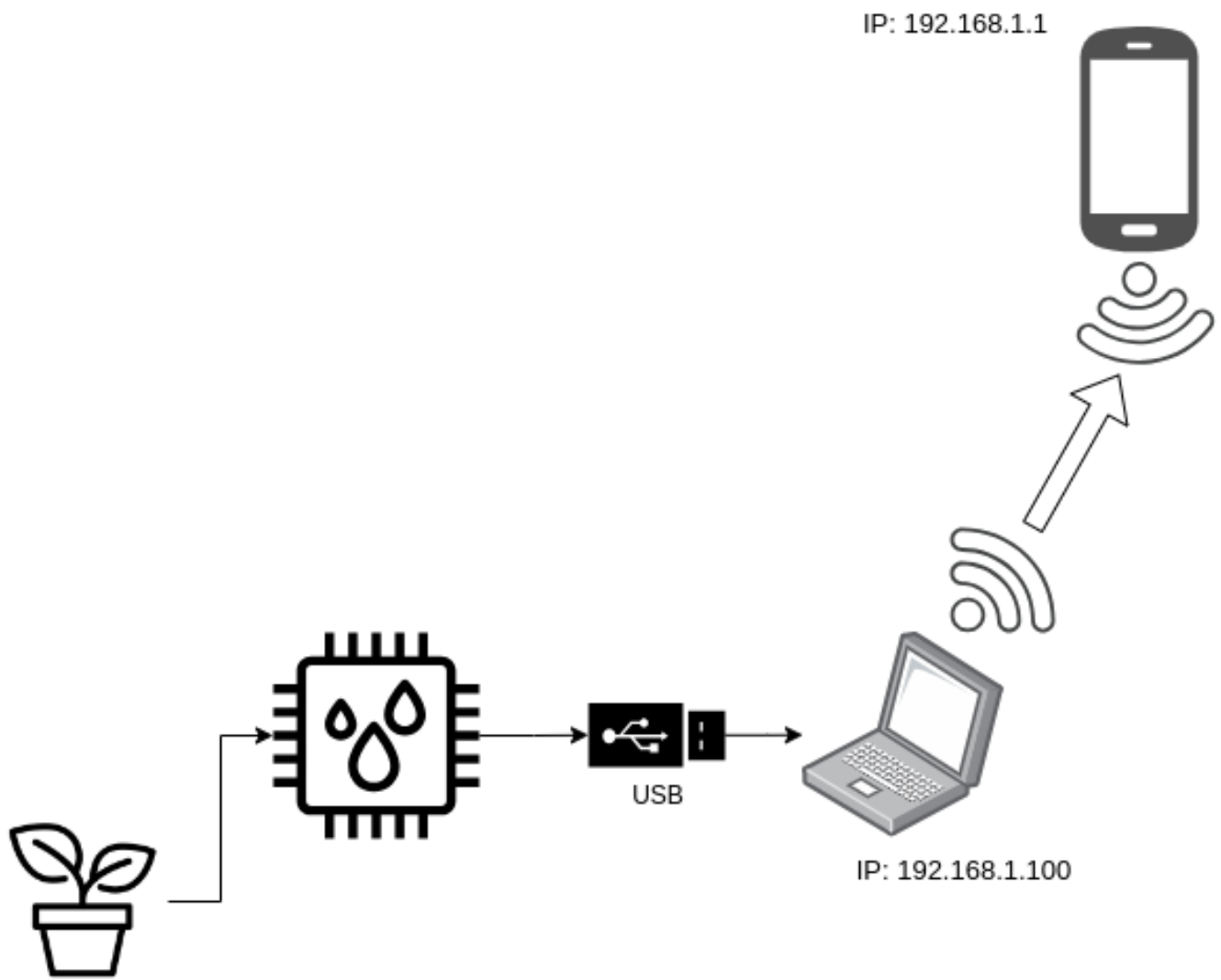


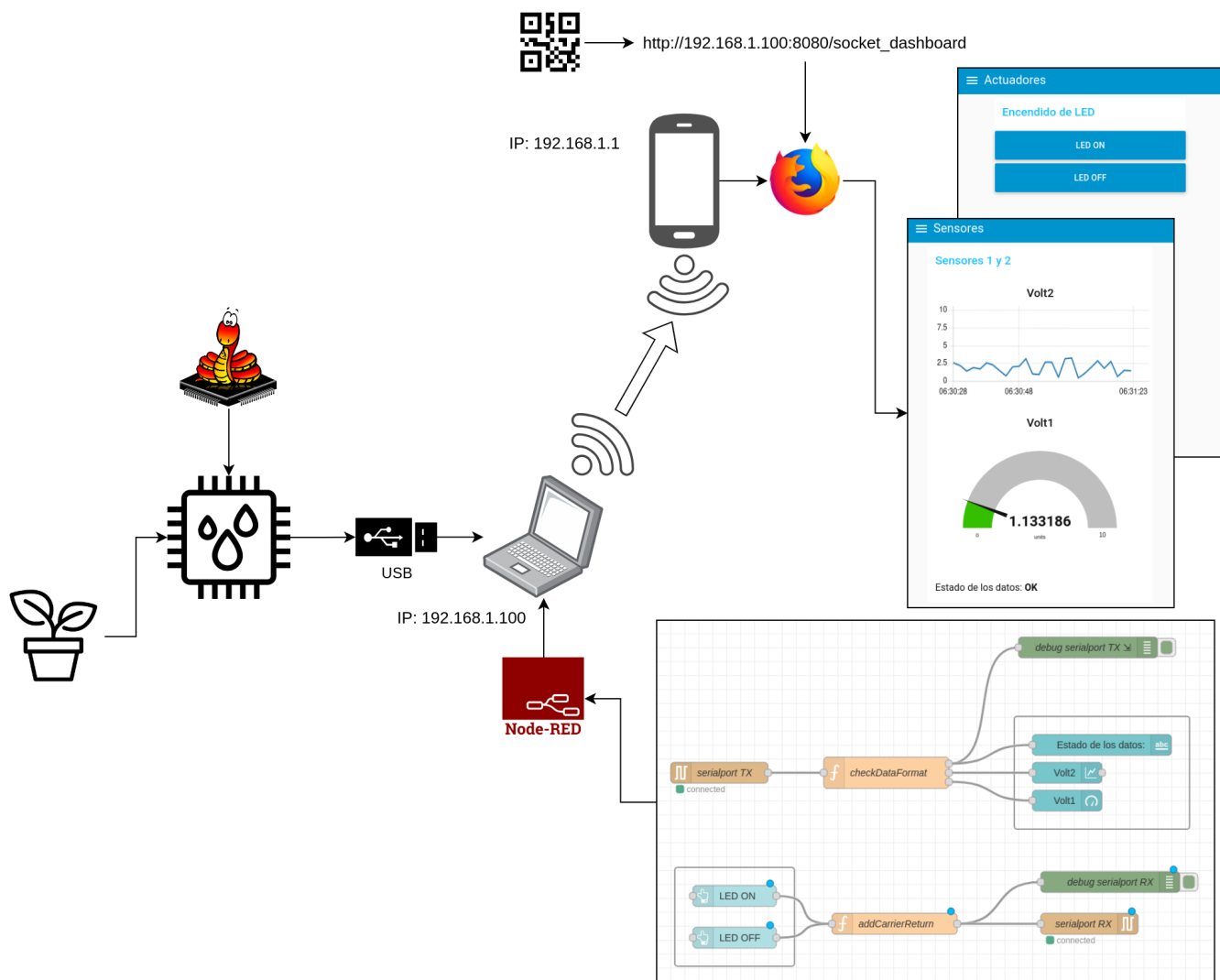
IP: 192.168.1.101

## Ip Table

IP1: 192.168.1.100

IP2: 192.168.1.101





## Configuración de Node-red

*Node-Red* es un software que permite conectar información de diferentes medios y protocolos ya sea para almacenar, presentar o controlar información de interés en un sistemas.

### Instalación de node-red

Para instalar Node-Red deberá primero instalar NodeJS para su sistema operativo, seguido, desde la línea de comandos de una terminal deberá instalar *node-red* con el gestor de paquetes de *nodejs* denominado **npm**. Según su sistema operativo siga las indicaciones planteadas en la documentación oficial:

- Instalar Nodejs
- Instalar Node-Red

Si tiene dificultades en la instalación puede buscar un ejemplo de instalación en Internet, Teniendo presente la documentación oficial ya que puede tener algunas variaciones en los comandos que deberá considerar, ejemplo de instalación de node-red

### EJECUTAR NODE-RED

Para ejecutar la aplicación nuevamente desde la terminal ejecute el siguiente comando:

```
node-red
```

Esto iniciará la aplicación y le pedirá que en el navegador que usted use escriba la dirección IP del servicio para que pueda ser consumido, ejemplo:

`http://127.0.0.1:1880`

Ponga la anterior dirección IP en el navegador para que *Node-red* esté listo para usarse.

## INSTALACIÓN DE COMPLEMENTOS/MÓDULOS EN NODE-RED

Para que el ejemplo pueda ser ejecutado requiere los módulos de *Dashboard* y *Serialport*. Dashboard permite crear una interfaz de usuario y Serialport la comunicación con el microcontrolador.

En la aplicación en la siguiente ruta busque e instale los módulos requeridos:

Ruta para instalación en la aplicación: Menu -> Manage Palette -> Install -> Option and Search

Módulos a buscar e instalar:

- node-red-dashboard
- node-red-node-serialport

Es posible que la aplicación le pida que deba reiniciar node-red, para tal fin, en la terminal, detenga la ejecución con **Control + C** y vuelva a iniciar el servicio de node-red con **node-red** en la misma terminal.

## Flujo de Node-red

A continuación se comparte el flujo diseñado en node-red que podrá descargar e importar en node-red y desplegar:

- node-red-flow.json

El flujo importado y desplegado en node red permite la comunicación bidireccional con el microcontrolador a través del puerto serial.

En la anterior imagen se puede observar los siguientes nodos:

- Comunicación serial bidireccional (serialport RX y serialport TX)
- Funciones para validar y dar formato a los mensajes que se envían y reciben (addCarrierReturn y checData-Format)
- Flujos de debug permiten darle seguimiento a los datos enviados o recibidos
- Nodos de dashboard los cuales se despliegan y son visualizados por el usuario en un formato legible, con estos nodos el usuario monitorea y controla el sistema (LED ON y LED OFF como entradas de información, y Estado de los datos, Volt1 y Volt2 como salida de información visual).

La lógica de los nodos desplegados está escrita en javascript, para conocer cómo funciona puede dar clip en el nodos y se mostrará una ventana con un menú e información de ello, ejemplo:

Para obtener la URL del dashboard desplegado puede dar clic en el ícono que encuentra en el menú del dashboard, ejemplo:

La anterior URL la puede compartir a otros dispositivos que estén en la misma red cambiando la ip de la URL por la IP del equipo donde está el servicio de *Node-red* que en este caso es el computador, ejemplo:

Si la URL generada es:

`http://127.0.0.1:1880/ui/#!/0?socketid=2xbT63rXMJDBZ5i3AAAB`

Y la ip del equipo es por ejemplo: 192.168.1.100 quedaría algo como lo siguiente:

`http://192.168.1.100:1880/ui/#!/0?socketid=2xbT63rXMJDBZ5i3AAAB`

Se recomienda codificar la URL en un código QR para que pueda ser compartido a un celular.

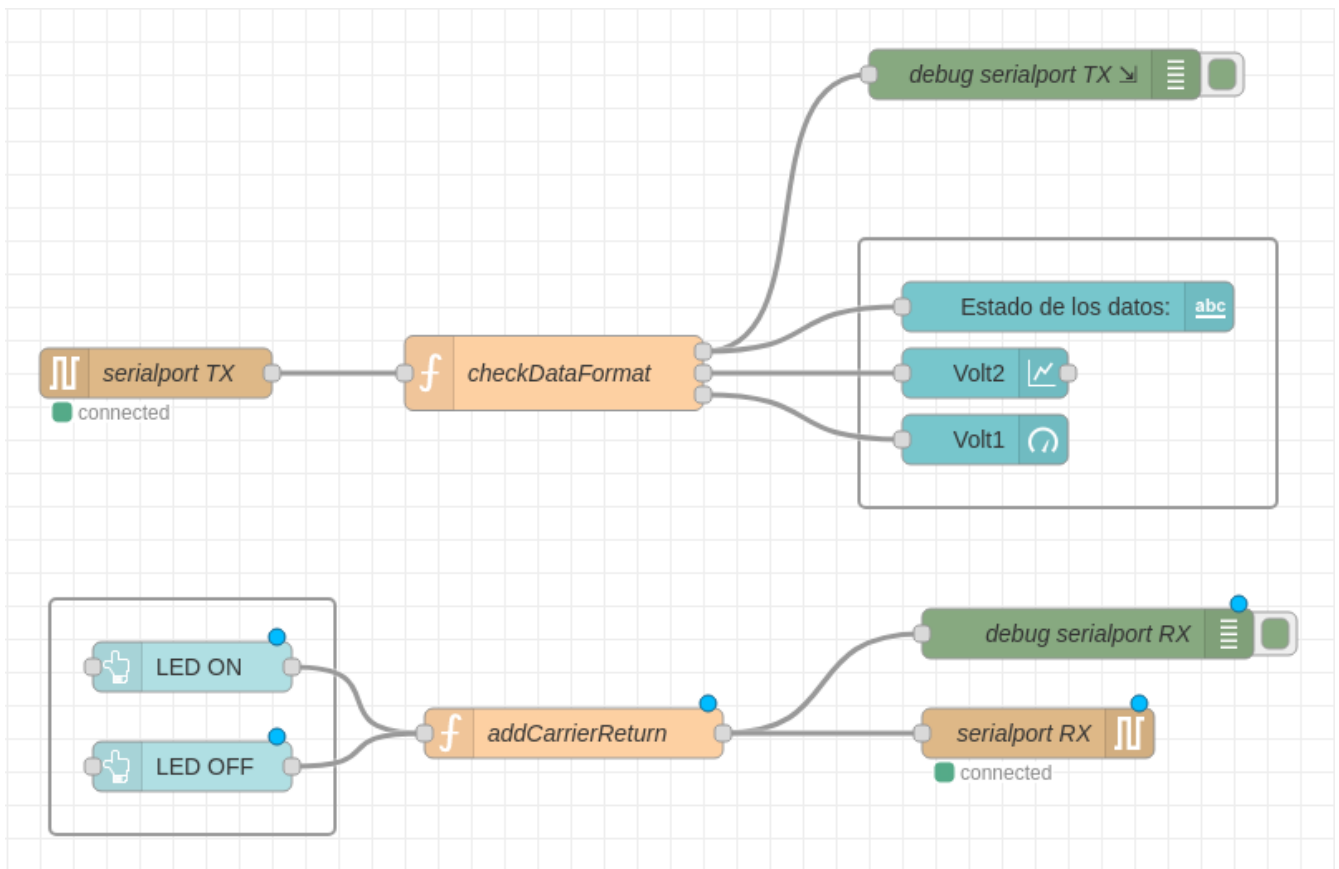


Figura 2: Flujo en node-red



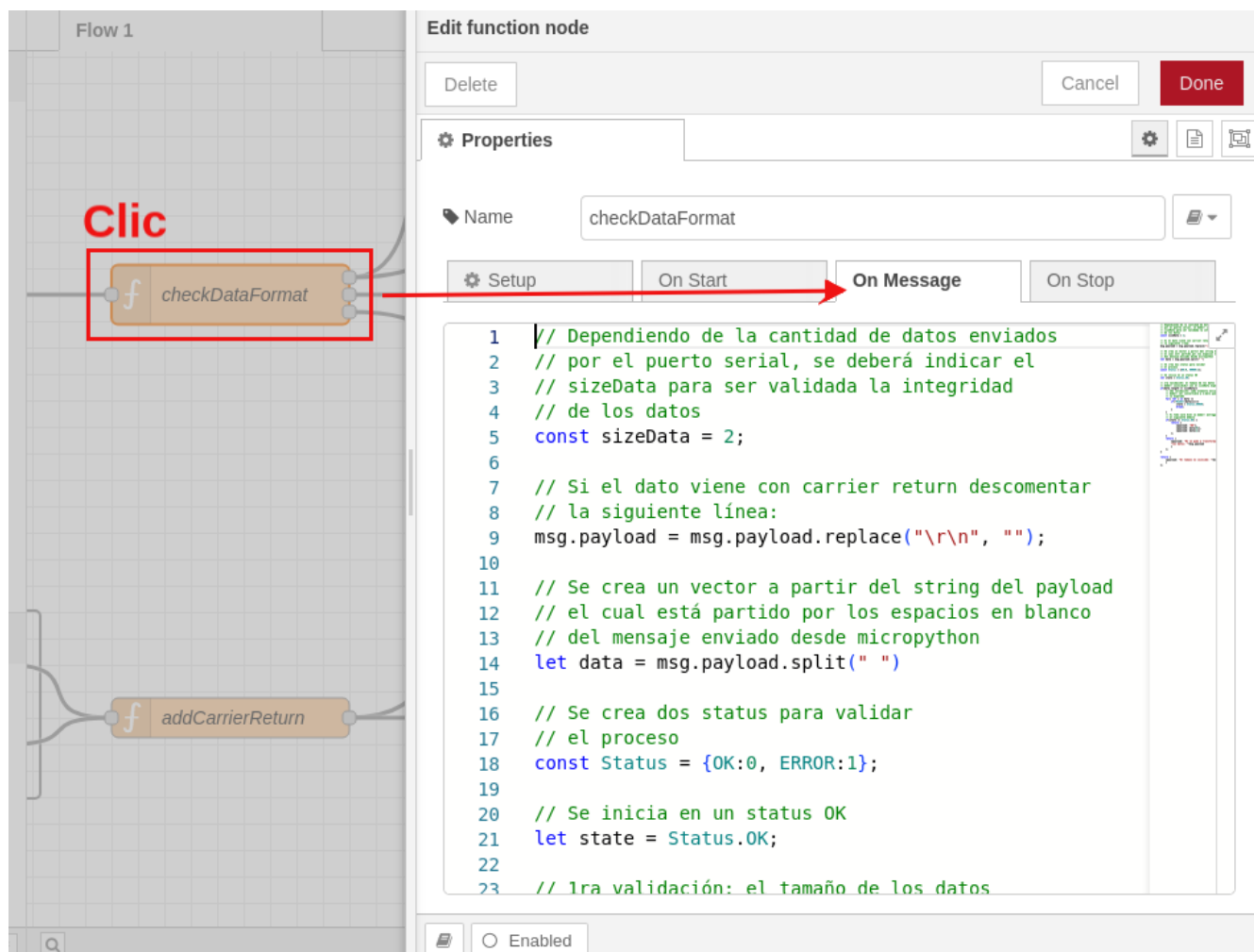


Figura 3: Lógica de un nodo en node-red

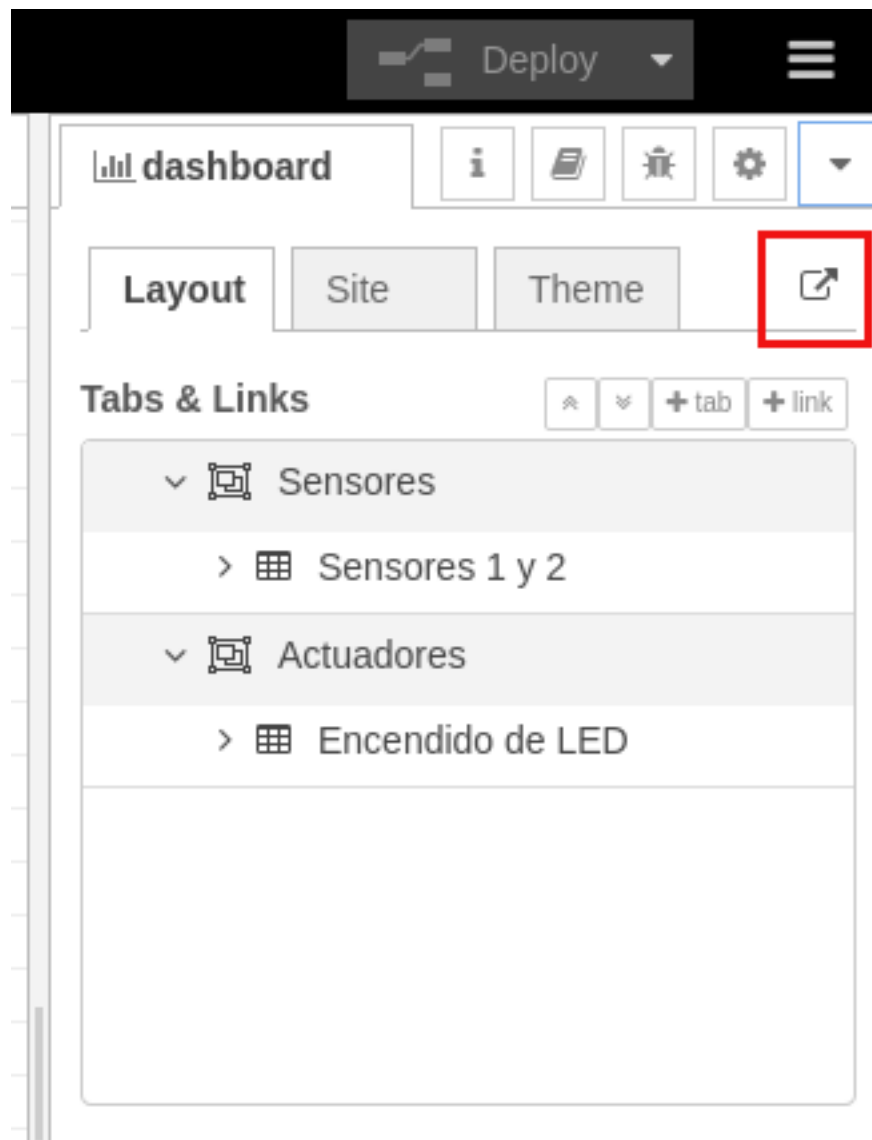


Figura 4: Obtener la url del dashboard desplegado

## Micropython y node-red (comunicación)

El siguiente programa que se ejecuta en el microcontrolador, permite la generación de los datos que se van a compartir en la red, como se hace uso de micropython es importante reconocer que la programación del microcontrolador se hace a través del REPL y ésta está compartida por el canal USB que será usado para el envío de datos, por tanto, aunque funciona se puede optimizar para otros propósitos.

Observe en el código que se comparte que los datos son enviados a través de la función `print()`, también observe que en este programa se hace uso de `threads` los cuales permiten la ejecución de tareas en loops distintos de manera “simultanea”.

```
import random, time, machine
from machine import Pin

led = Pin(25, Pin.OUT)

# Esta función emula la lectura de un sensor
def readSensor():
    return random.random()*10

# variables globales
volt1 = 0
volt2 = 0

# Esta es una función que será usada para ejecutarse
# en un hilo independiente, puede crear otras
# funciones para otros propósitos si así lo desea
def thread_main():
    while True:
        time.sleep(2)
        volt1 = readSensor()
        volt2 = readSensor()
        print(volt1, volt2)

import _thread
# Inicio de un hilo de ejecución llamando a la función
# thread_main anteriormente creada
_thread.start_new_thread(thread_main, ())

# Función creada para recibir órdenes desde la
# comunicación serial, puede agregar las funciones que
# se desee realizar
def c(orden):
    if(orden == 1):
        led.on()
    elif(orden == 2):
        led.off()
```

Para el ejemplo, descargue el programa `main.py` y guárdelo con ese nombre en el microcontrolador que tiene micropython, libere el puerto de comunicación serial, en el caso de *Thonny* cambie de interprete o cierre la aplicación, si es necesario desconecte y conecte el microcontrolador del puerto USB.

Finalmente en Node-red reinicie los flujos y observe en el dashboard si se reciben datos desde el microcontrolador, interactúe con los botones de encendido y de apagado del LED.

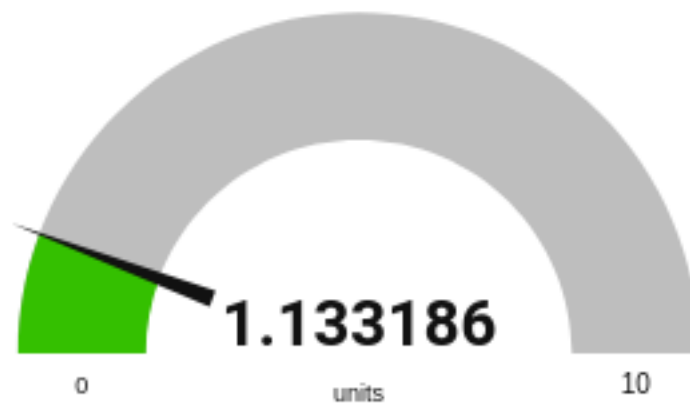
Si por alguna razón no se puede visualizar los datos, es recomendable reiniciar *Node-red* y refrescar las ventanas del navegador.

## Sensores 1 y 2

### Volt2



### Volt1



Estado de los datos: **OK**

Figura 5: Dashboard de node-red visualizadores

## Encendido de LED

LED ON

LED OFF

Figura 6: Dashboard de node-red botones