

# ***Monitorización de inversor solar***

Instalación del software	3
Programación del sistema	4
Programación del sistema 2 parte	6
Configurar Dashboard	7

# Instalación del software

Lo primero que tenemos que hacer es instalar node-red en nuestra raspberry pi, para hacer esto tenemos que abrir el terminal y teclear el siguiente comando

- ***apt-get install nodered***
- ***node-red-start***

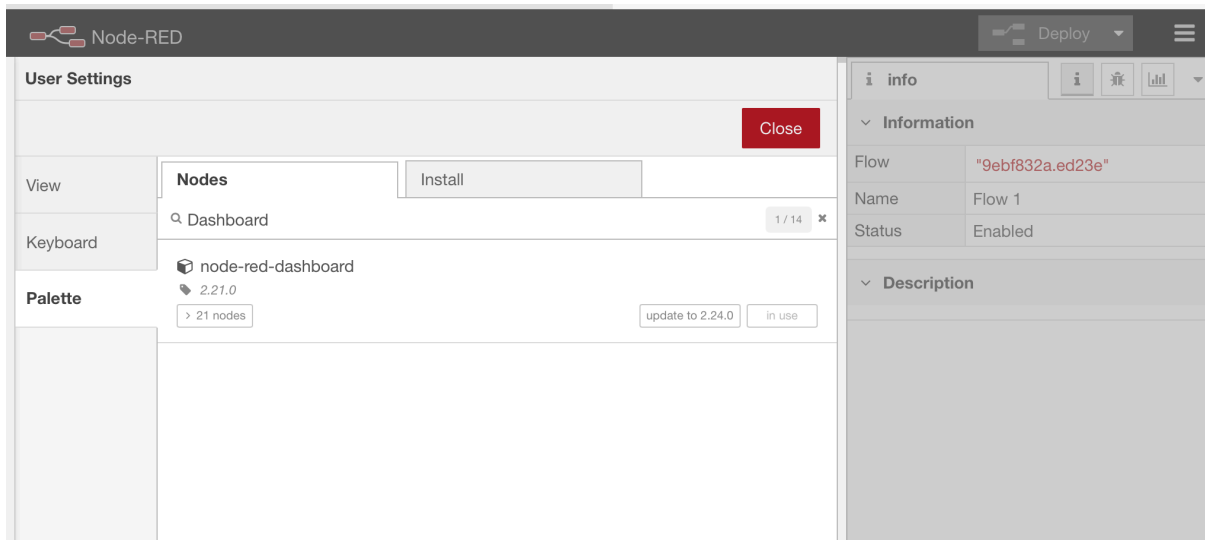
Ahora que tenemos node red instalado vamos a configurarlo para que se arranque al inicio

- sudo systemctl enable nodered.service***

Una vez que tengamos todo esto ya podemos acceder a la interfaz web de la siguiente manera

- ***http/"ip-raspberry":1880***

Dentro de la interfaz web, presionando el botón de las tres líneas horizontales accedemos al panel de "Manage palette " y buscamos la palabra "dashboard"e instalamos el plugin



Con esto tendremos todo lo que necesitamos para hacer nuestro proyecto de monitorización.

# Programación del sistema

Llegados a esta parte tengo que decir que hay mil maneras de hacer esto y que yo os definiré la mía propia.

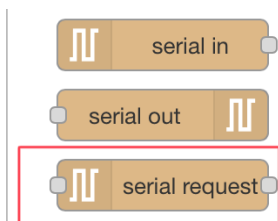
Para este ejemplo usaremos el inversor Axpert King pero es válido para cualquier tipo de inversor ya que solo tenéis que sustituir el comando que te devuelve los datos.

Lo primero que tenemos que definir es la configuración de nuestro puerto serie, en este caso es el siguiente

- Baud rate: 2400
- Data bit:8
- Parity bit: none
- Stop bit: 1

Estos datos los podéis ver en el protocolo de comunicación de vuestro inversor.

El primer nodo que tenemos que utilizar es el "**serial request**" que está dentro del menú network



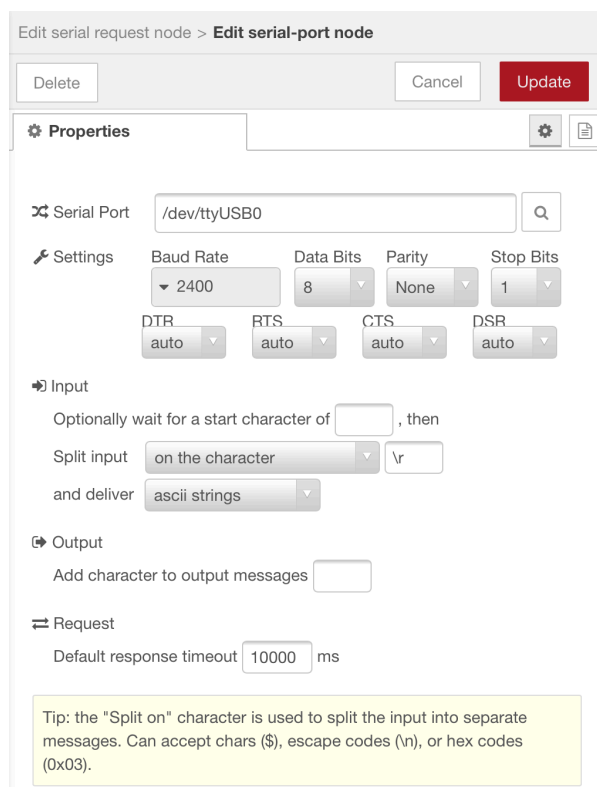
Una vez tengamos el nodo puesto en nuestro tablero tendremos que configurar la conexión, le pincharemos doble click y presionaremos sobre el icono del lápiz

Para mi gusto este es el paso más importante de todos.

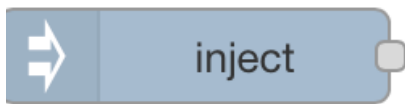
Serial port: hace referencia donde está conectado nuestro cable de comunicación, en mi caso tengo un conversor a usb que se conecta a unos de los puertos de la raspberry USB. Simplemente presionamos el botón de la lupa y nos aparecerá los puertos serie disponibles

Settings: simplemente ponemos los datos que hemos recogido en el protocolo de comunicación del inversor.

Es importante poner el `\r` en el campo de split input que hace referencia a un retorno de carro



Una vez tenemos configurado nuestro puerto serie ahora tenemos que configurar la entrada del mismo, para esto usaremos el nodo "**inject**" del menú **common**.



En este nodo insertamos el comando que queremos enviarle al inversor, en nuestro ejemplo usaremos el comando QPIGS que nos devuelve es estado general del inversor. Si no usáis el mismo inversor solo tenéis que iros al protocolo de comunicación y ver el comando que os interesa.

Es importante configurar el campo **repeat** para que el comando se envíe cada X tiempo.

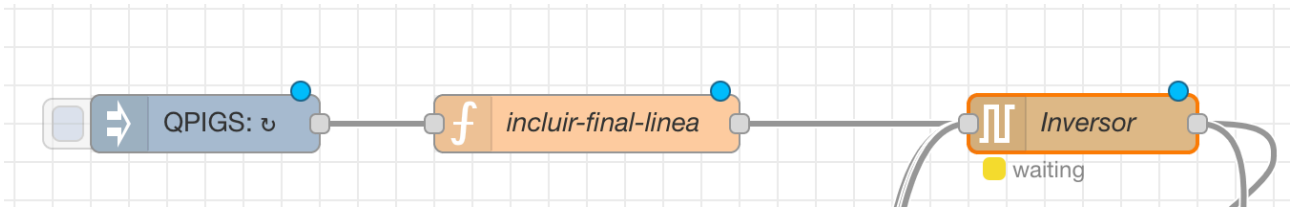
A dialog box titled "Edit inject node" with a "Delete" button on the left, "Cancel" and "Done" buttons on the right. Below the title bar is a "Properties" section with a gear icon, a document icon, and a refresh icon. The "Payload" field has a dropdown menu showing "az" and "QPIGS". The "Topic" field is empty. There is a checkbox for "Inject once after 0.1 seconds, then" which is currently unchecked. The "Repeat" field has a dropdown menu showing "none". The "Name" field contains the text "Name".

Ahora añadiremos un nodo intermediario entre inject y el puerto serie que se llama **función**, este nodo nos permitirá añadir los saltos de línea (\n) para que el comando funcione correctamente.

A dialog box titled "Edit function node" with a "Delete" button on the left, "Cancel" and "Done" buttons on the right. Below the title bar is a "Properties" section with a gear icon, a document icon, and a refresh icon. The "Name" field contains the text "incluir-final-linea". The "Function" field contains a code editor with two lines of code:

```
1 msg.payload="\n" + "QPIGS" + "\n"
2 return msg;
```

Y así quedaría la primera parte del diagrama



## Programación del sistema 2 parte

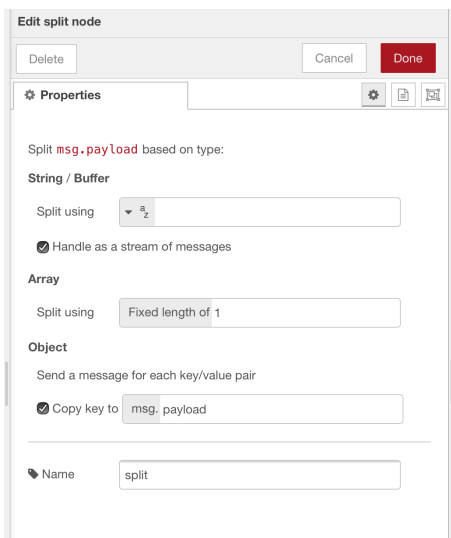
Ahora mismo ya tenemos en la salida una cadena de caracteres como esta

(BBB.B CC.C DDD.D EE.E FFFF GGGG HHH III JJ.JJ KKK OOO TTTT EE.E UUU.U  
WW.WW PPPPP b7b6b5b4b3b2b1b0 QQ VV MMMMM b10b9b8

Letra	Descripción	Tipo dato
B	Voltaje del grid	Decimal voltios
C	Intensidad del grid	Decimal amperios
D	Voltaje AC salida	Decimal voltios
E	Frecuencia AC salida	Decimal Hz
F	Intensidad AC salida	Decimal Amperios
Etc	Etc	Etc

\*Para ver todo los parámetros consultar el protocolo de comunicación

Pero todo junto no nos sirve de nada por lo que tenemos que separarlo, para esto usaremos la función "split"



Aquí os adjunto la configuración de esta función, tener en cuenta que el campo **"split using"** no está vacío sino que hay un espacio en blanco, sin este espacio la función no funcionará.

Justo después de esta función usaremos la función join que nos transformará la cadena troceada en un array, esto nos permitirá acceder a los datos más cómodamente

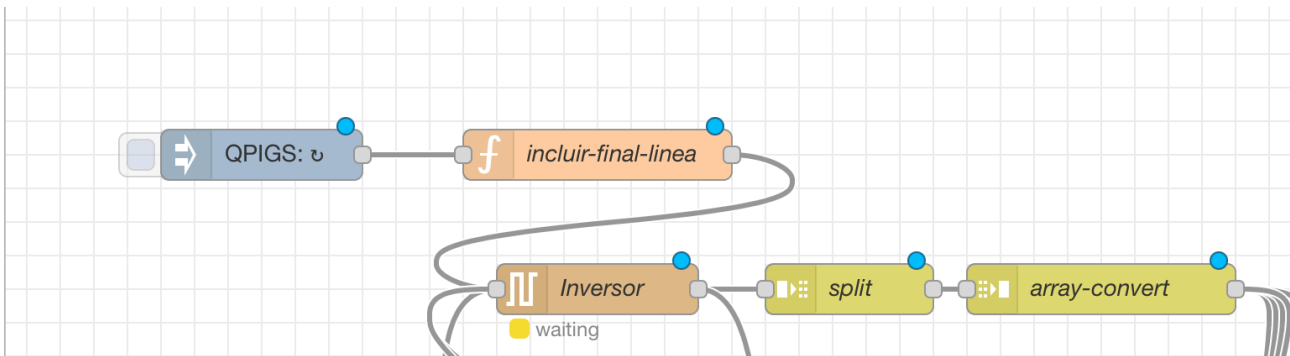
La configuración de este nodo es la siguiente:

- Mode: manual
- Combine each: msg.payload
- to create: an Array

# Configurar Dashboard

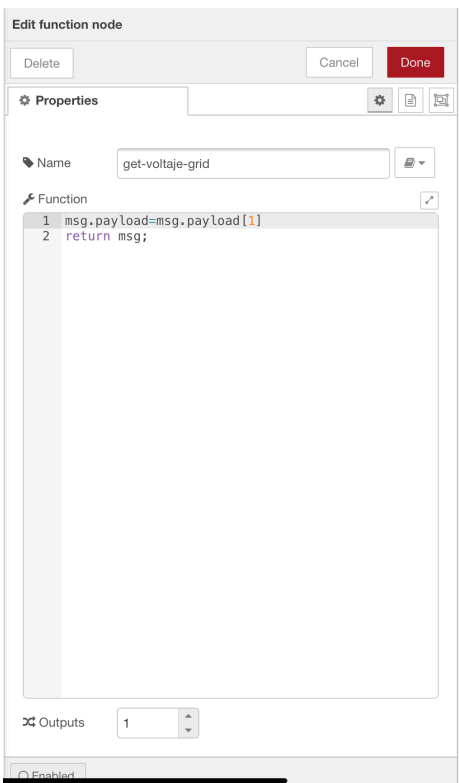
Actualmente ya tenemos todos los datos separados, solamente nos queda darle formato y visualizarlos en nuestro navegador

Ahora mismo nuestro panel de node red es el siguiente



Para dar formato a nuestros datos usaremos el nodo función que ya hemos visto anteriormente, hay que entender que ahora mismo tenemos un array o tabla de una sola fila con todos los datos

Grid voltaje	Grid current	AC Ouput	AC frecuencia	Etc
--------------	--------------	----------	---------------	-----



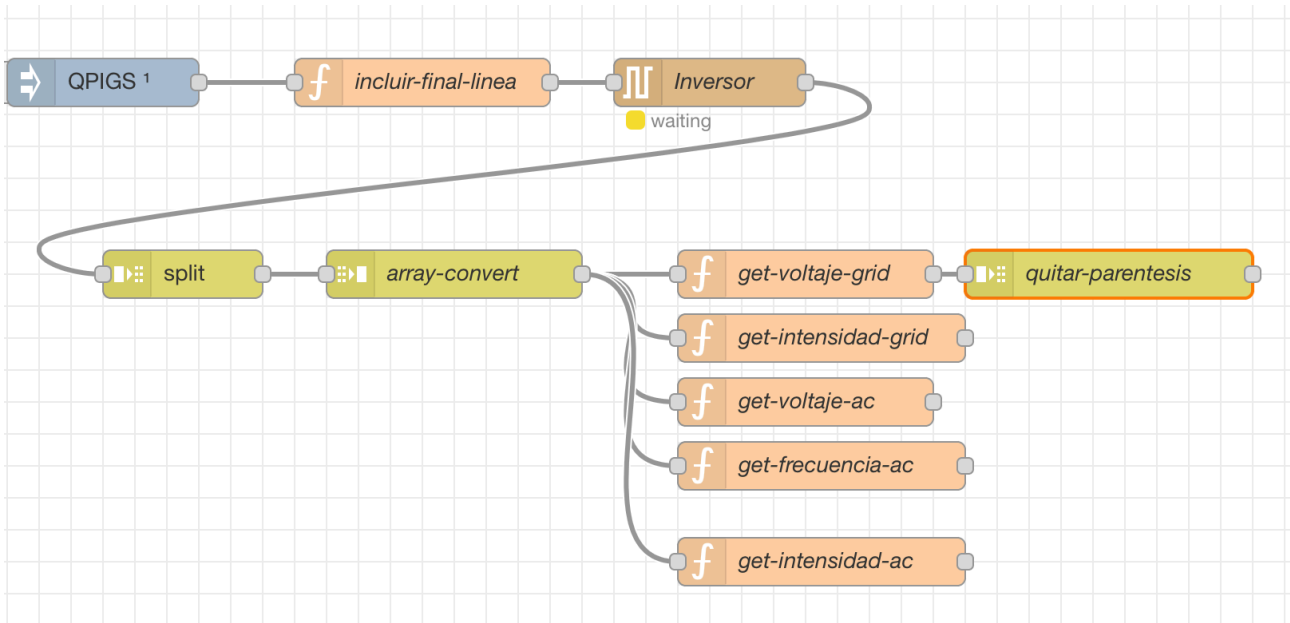
Cada valor está ordenado numéricamente en este caso empieza por 1, por ejemplo el "AC Ouput" es el número 3

La función que necesitamos implementar es la siguiente Como podéis ver el número que está entre los corchetes hace referencia al dato que queremos obtener, si quisiéramos sacar el "AC ouput" dentro del corchete tendríamos que tener un numero 3.

Hay que implementar una nodo de función por cada dato que queremos obtener.

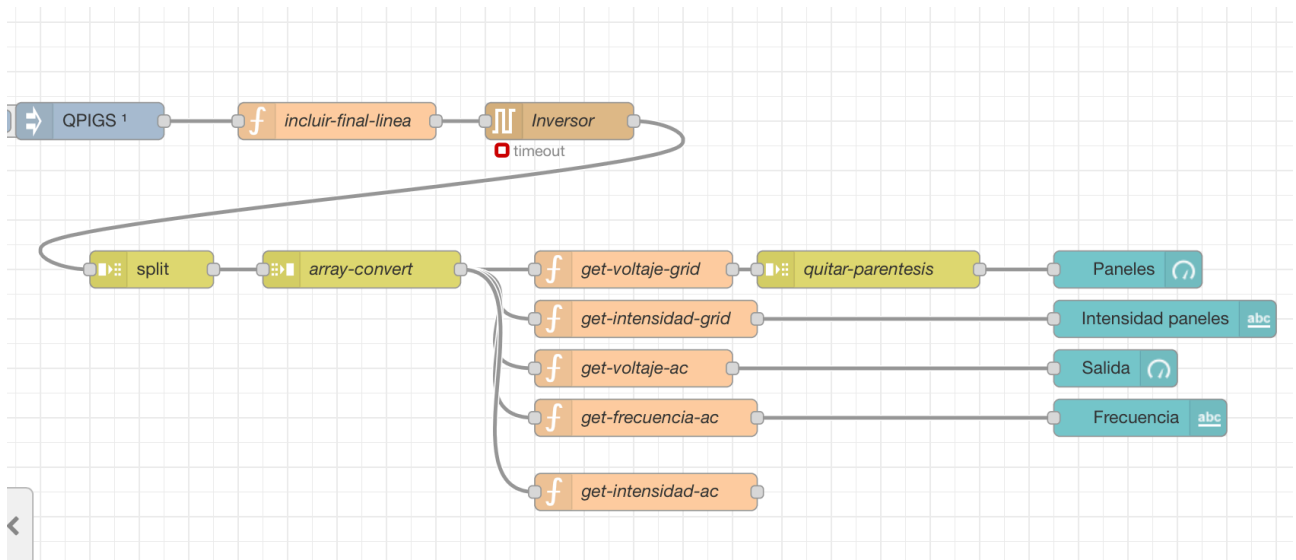
Solo para el primer dato de nuestro array que es el "grid voltaje" hay que implementar una función extra ya que el dato viene con el carácter "(", para solucionar esto usaremos en nodo "split", la configuración del nodo es el siguiente  
-split using: (

Finalmente el esquema de nuestros nodos quedará así

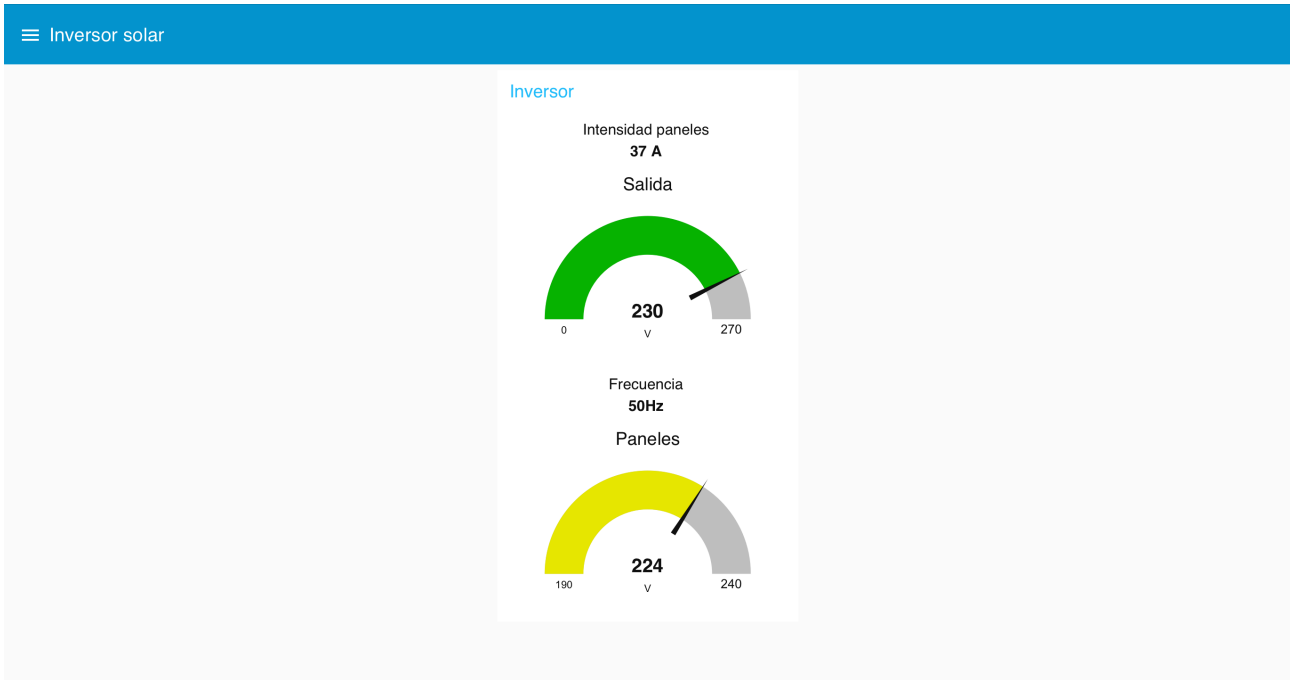


Ahora viene la mejor parte de todas, vamos a trasladar todo esto a nuestra interfaz gráfica que la podremos ver en cualquier navegador, para esto usaremos todos los nodos que hemos instalado de la sección "**Dashboard**", aquí es un poquito a gusto del consumidor, si lo queréis mostrar en texto plano o en gráfico, en este caso he hecho dos gráficos y dos textos para que lo veáis

La URL de la interfaz web sería : [http://\"ip raspberry\":1880/ui](http://\)







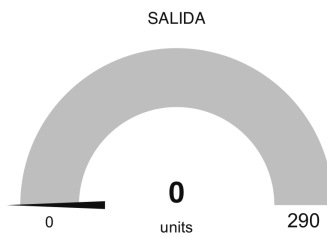
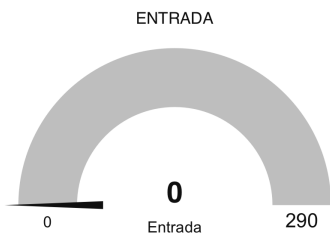
El resultado final sería este

Aquí os adjunto otro proyecto que estoy haciendo, toda la parte visual solo es dedicarle un poco de tiempo moviendo los gráficos y textos

AUTONOMIA

Autonomia: minutos  
Consumo: W

DATOS TECNICOS



ENTRADA