Primeros pasos con el PLC Raspberry Pi

Comandos útiles, consejos y trucos, aplicaciones interesantes.

Industrial Shields®

Índice

Cómo conectar la cámara USB al PLC Raspberry

Uso del PLC Industrial Raspberry

Explicación

TOMAR FOTOS

1. Conecta el PLC Raspberry a una red WIFi.

2. Abre una ventana de terminal y emite el siguiente comando para instalar el paquete fswebcam:

```
sudo apt update
sudo apt install fswebcam
```

3. Añade tu usuario al grupo de vídeo:

```
sudo usermod -a -F video <username>
```

4. Introduce el comando fswebcam seguido de un nombre de archivo. Se tomará una foto con la cámara web y se guardará en el nombre de archivo especificado:

```
fswebcam image.jpg
```

5. Especifica la resolución utilizando la bandera -r:

```
fswebcam -f 1280x720 image.jpg
```

Para más información, ve a:
https://www.raspberrypi.org/documentation/usage/webcams/

Explicación

GRABAR VIDEOS

1. Instala los paquetes ffmpeg:

```
sudo apt update
sudo apt install ffmpeg
```

2. Ejecuta el siguiente comando para grabar un vídeo desde un archivo de entrada y guardarlo en un archivo de salida:

```
ffmpeg -i <input file> <output file>
```

Consulta la documentación de ffmpeg para más detalles:
https://www.ffmpeg.org/ffmpeg.html

Enlaces relacionados

- Cómo conectar un Raspberry PLC al Wi-Fi
- Conceptos básicos sobre las salidas analógicas de un Raspberry Pi PLC
- Cómo encontrar tu PLC industrial perfecto
- Cómo programar en Python las entradas de interrupción del PLC industrial Raspberry
- Familia de productos Raspberry PLC
- Familia de productos TouchBerry Pi

Comandos más útiles de Raspberry Pi

Uso del Raspberry

Raspberry Pi

Introducción

Los comandos de Raspberry Pi nos permiten trabajar en una amplia gama de aplicaciones. Desde la construcción de un prototipo hasta el desarrollo de un software existente, Raspberry Pi puede proporcionar el apoyo.
En esta sección, aprenderás 5 herramientas de línea de comandos realmente útiles para utilizar tu Raspberry Pi o Raspberry PLC en un entorno seguro.

Explicación

Vcgencmd measure_temp

Vcgencmd es una utilidad de línea de comandos que puede obtener varios datos de la GPU VideoCore en la Raspberry Pi.

```
pi@raspberrypi:~ $ vcgencmd measure_temp
temp=49.6'C
```

Es importante conocer la temperatura de la Raspberry, porque un calor excesivo puede llevarle a situaciones no deseadas. De hecho, los que overclockean el procesador de la Raspberry deben comprobar la temperatura con frecuencia, ya que todos los modelos de Raspberry pi realizan un grado de gestión térmica para evitar el sobrecalentamiento bajo carga pesada. Los SoC tienen un sensor de temperatura interno, que el software de la GPU sondea para garantizar que las temperaturas no superen un límite predefinido.

Cuando la temperatura del núcleo está entre 80ºC y 85ºC, se muestra un icono de advertencia con un termómetro rojo a medio llenar, y los núcleos ARM se ralentizan progresivamente.

Por lo tanto, utiliza la opción **measure_temp**, para obtener la temperatura del SoC (System on Chip) medida por el sensor de temperatura de la placa, para ayudarte con el control de la temperatura de tu dispositivo.

Explicación

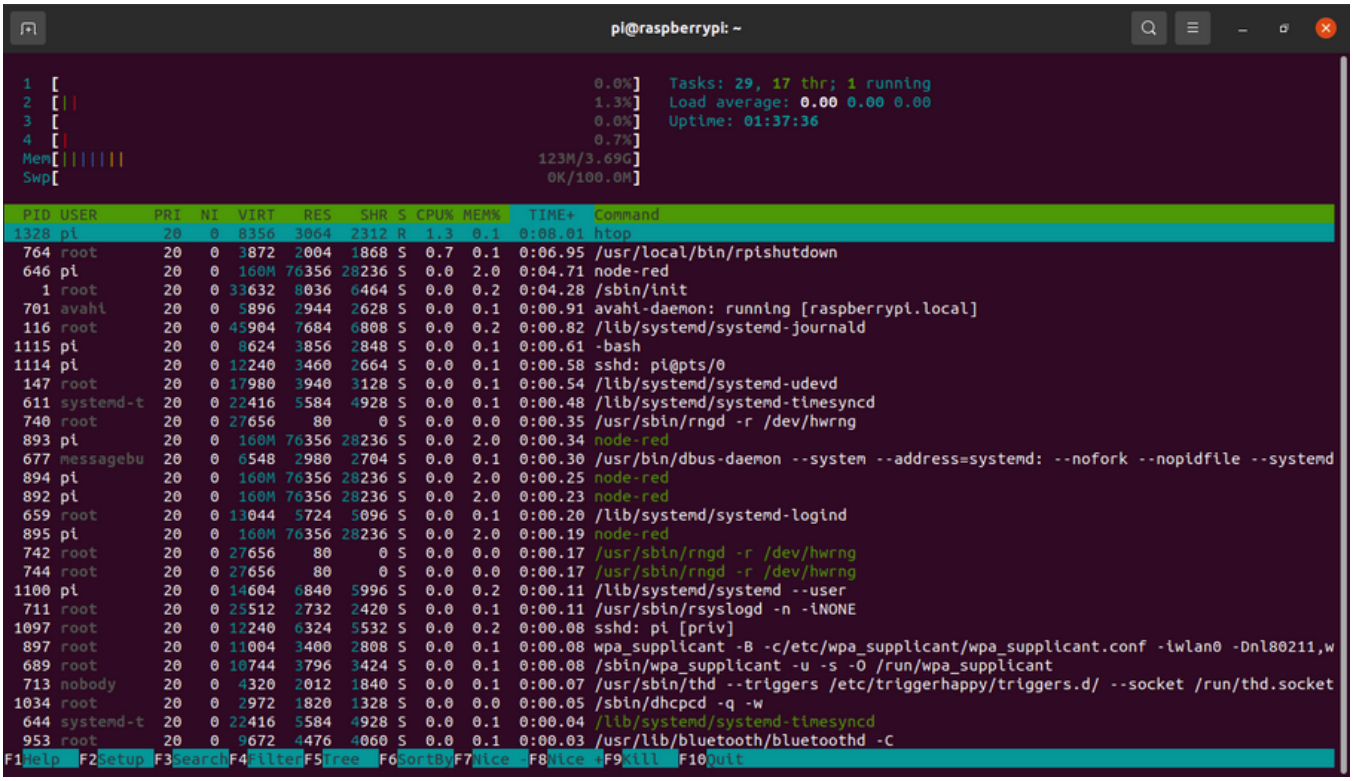
Consulta la siguiente URL para saber más sobre vcgencmd:
https://www.raspberrypi.org/documentation/raspbian/applications/vcgencmd.md

Htop

Htop es una utilidad de línea de comandos realmente potente que te permite monitorizar interactivamente los recursos vitales de tu sistema o los procesos del servidor en tiempo real.
Es bastante similar al comando *top*. Sin embargo, dado que *htop* es un programa más nuevo en comparación con *top*, ofrece muchas mejoras.
Htop también admite operaciones con el ratón, utiliza colores en sus salidas y da indicaciones visuales sobre el uso del procesador, la memoria y la swap.
También imprime líneas de comando completas para los procesos y permite desplazarse vertical y horizontalmente para los procesos y las líneas de comando respectivamente.
Puede instalarlo haciendo:

```
sudo apt update
sudo apt install htop
```

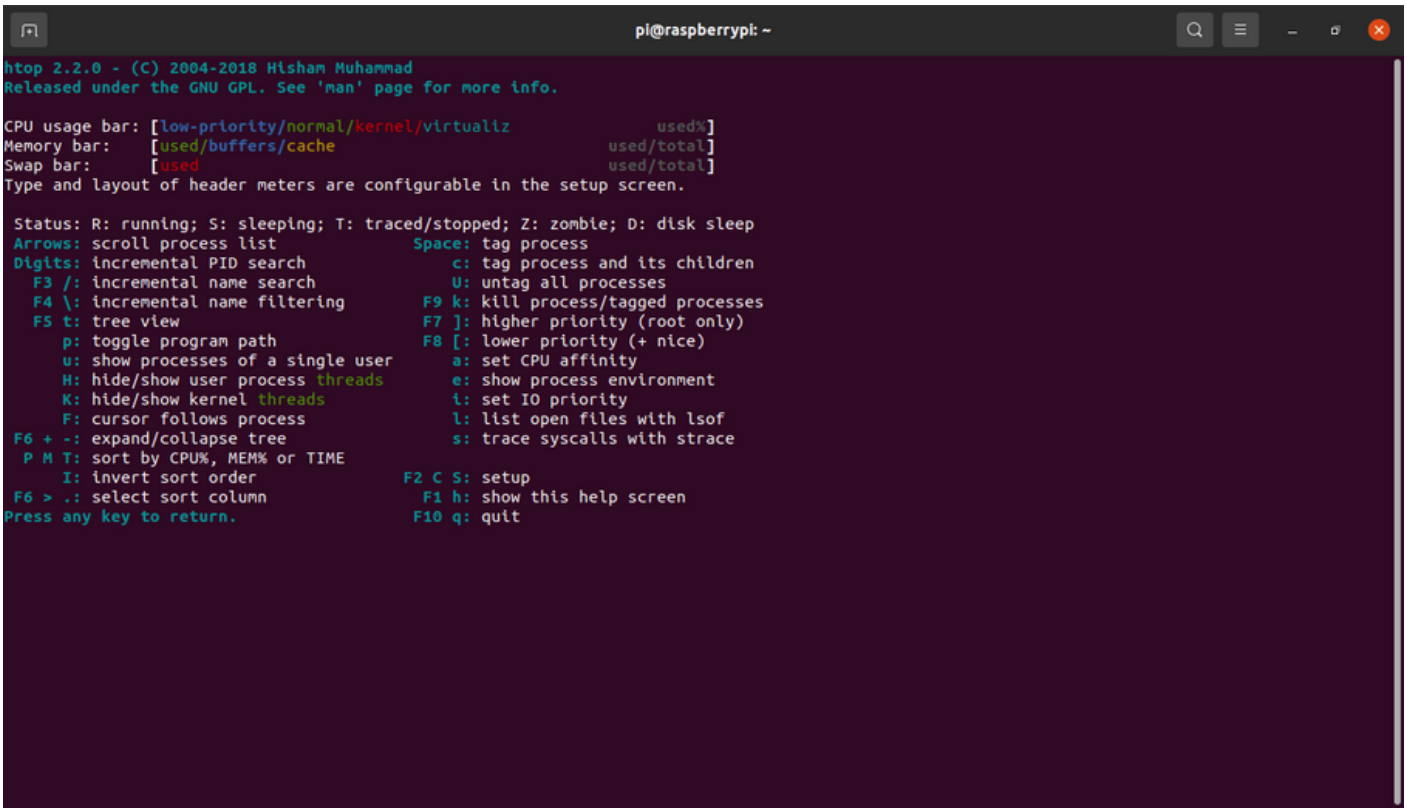
Así, si ejecuta **htop** en la línea de comandos, obtendrá algo como esto:



Explicación

Htop

Por último, si haces clic en **F1**, obtendrás ayuda sobre las posibilidades y verás cómo funciona.



Dmesg

El kernel de Linux es el núcleo del sistema operativo que controla el acceso a los recursos del sistema, como la CPU, los dispositivos de E/S, la memoria física y los sistemas de archivos. El kernel escribe varios mensajes en el ring buffer del kernel durante el proceso de arranque y cuando el sistema está funcionando.

El ring buffer del kernel es una porción de la memoria física que contiene los mensajes de registro del kernel. Tiene un tamaño fijo, lo que significa que una vez que el buffer está lleno, los registros más antiguos se sobrescriben.

Explicación

Dmesg

Dmesg se utiliza para examinar o controlar el buffer de anillo del kernel. Es realmente útil para examinar los mensajes de arranque del kernel y depurar problemas relacionados con el hardware. La acción por defecto es mostrar todos los mensajes del ring buffer del kernel.

Por lo tanto, simplemente ejecútalo como:

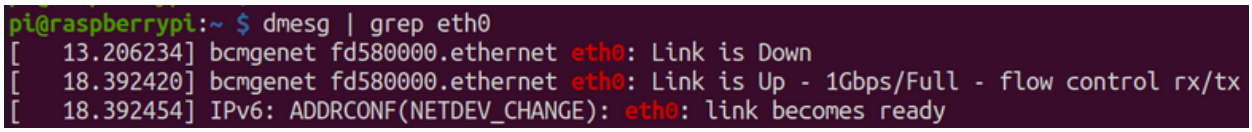
```
dmesg
```

Uso:

```
dmesg [options]
```

Si ves que el comando dmesg muestra demasiados mensajes que no eres capaz de leer, entonces encuentra las palabras que realmente quieres encontrar usando | grep:

```
dmesg | grep eth0
```



Para saber más sobre este comando y sus opciones, basta con escribir:

```
dmesg -h
or
man dmesg
```

Explicación

Operadores de coma y corchetes

Podemos hacer muchas cosas con las operaciones de comas y corchetes, para hacernos la vida más fácil, veamos algunos usos:

- **Operaciones de cambio de nombre y de copia de seguridad con operadores de comas y llaves.**
- **Comparación de patrones con el operador coma y llaves.**
- **Operaciones de cambio de nombre y copia de seguridad (prefijando el nombre) en nombres de archivo largos.**
- **Para copiar archivos de un directorio padre sin escribir dos veces la ruta larga**

1. Para hacer una copia de seguridad de hola.txt a hola.txt.bak:

```
cp hello.txt{,.bak,}
```

2. Para revertir el archivo de hello.txt.bak a hello.txt:

```
mv hello.txt{.bak,}
```

3. Para renombrar el archivo con el prefijo "1-":

```
cp hello.txt 1-!#^
```

4. Para copiar archivos de un directorio principal sin escribir dos veces la ruta larga:

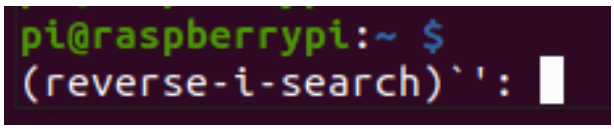
```
cp firstDir/secondDir/thirdDir/{hello.txt,bye.txt}
```

Explicación

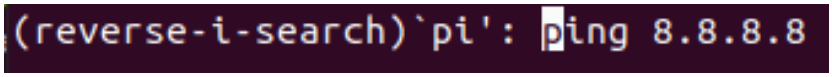
Ctrl + R

¿Te imaginas que pudieras autocompletar tus comandos con los que has escrito antes? Algo así como Google Autocomplete. Sería realmente útil, ¿verdad? Eso es posible abriendo una ventana de terminal y probando lo siguiente:

1.Ctrl + R



2. Comienza a escribir tu comando y aparecerán algunas sugerencias:

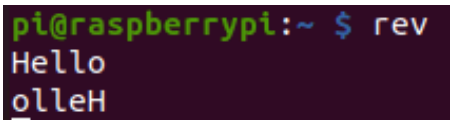


3. Si quieres escribir el comando sugerido, entonces haz clic en el tabulador, o en la flecha derecha de tu teclado, y ese comando sugerido se pondrá en tu línea de comandos listo para ser usado. En caso de que quieras ver más comandos sugeridos, entonces intenta Ctrl + R de nuevo hasta que veas el comando que quieres ejecutar.

Por último, algunos comandos complicados

1.Escribe "rev" para invertir el mensaje a escribir:

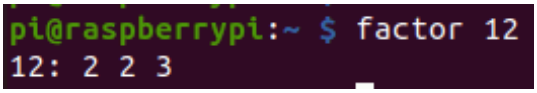
```
rev
```



Explicación

Por último, algunos comandos complicados

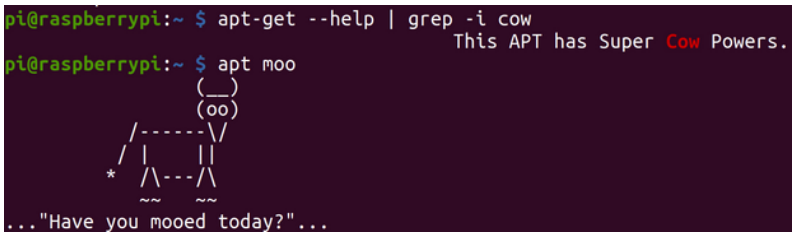
2. Escribe "factor <número>" para factorizar cualquier número:



3. Por último, escribe lo siguiente y verás lo que ocurre:

```
apt moo

apt-get --help | grep -i cow
```



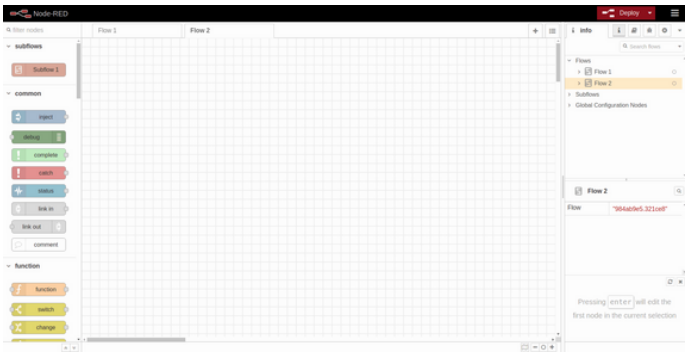
Enlaces relacionados

- Cómo conectar un Raspberry PLC al Wi-Fi
- Conceptos básicos sobre las salidas analógicas de un Raspberry Pi PLC
- Cómo encontrar tu PLC industrial perfecto
- Cómo programar en Python las entradas de interrupción del PLC industrial Raspberry
- Familia de productos Raspberry PLC
- Familia de productos TouchBerry Pi

Tutorial de Node-RED y Raspberry: Cómo capturar los datos del sensor

Aplicaciones industriales Raspberry PLC y Node-RED: captura de valores del sensor de peso

Introducción



To know more about Node-RED click on the image

Node-RED es una herramienta de programación para conectar dispositivos de hardware. API y servicios en línea de formas nuevas e interesantes.
Proporciona un editor basado en el navegador que facilita el cableado de los flujos utilizando la amplia gama de nodos de la paleta que se puede desplegar a su tiempo de ejecución en un solo clic.

En esta sección, aprenderás a desarrollar la aplicación Node-RED que se muestra en este enlace.

Explicación

Conceptos básicos de Node-RED

Como dijimos en la introducción, Node-RED proporciona un editor basado en el navegador que facilita la unión de flujos utilizando la amplia gama de nodos de la paleta que pueden ser desplegados en su tiempo de ejecución en un solo clic.
Así pues, vamos a descubrir los aspectos básicos:

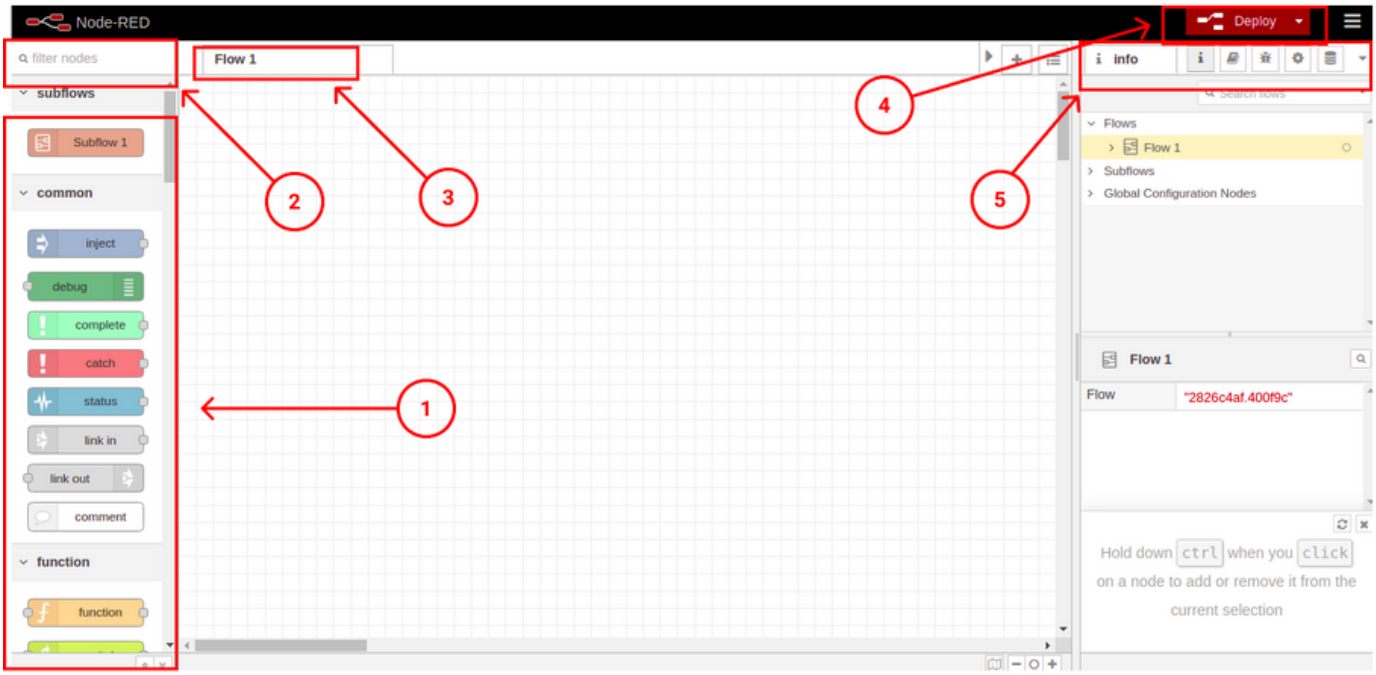
1. Node-RED dispone de una **amplia gama de nodos** que te ofrece muchas posibilidades. Si vas al menú de nodos de la izquierda, encontrarás los nodos que vienen por defecto. Son fáciles de usar; sólo tienes que arrastrarlos y soltarlos en tu flujo para poder empezar a utilizarlos.
2. Además, si ya sabes qué nodo quieres, hay una **barra de búsqueda** para filtrar los nodos y encontrar exactamente el que quieres.
3. Si haces doble clic en la pestaña **Flow 1**, se mostrará una ventana de configuración en la que podrás cambiar su nombre o desactivarlo, por ejemplo. En la misma barra, hay una pestaña + que añade otra pestaña de Flujo, para que puedas usar tantas como quieras.
4. Una vez que tengas tus nodos conectados y quieras **desplegar tus cambios**, haz clic en el botón ***Deploy***.

Explicación

Conceptos básicos de Node-RED

5. Por último, en la barra de la derecha, donde aparece la **pestaña de información**, hay más pestañas importantes como:

- **Información:** para obtener información general sobre los flujos.
- **Pestaña de ayuda:** que te da información sobre el nodo en el que has hecho clic.
- **Mensajes de depuración:** es una pestaña realmente útil para conocer los errores que se han producido, o para visualizar los mensajes de depuración del nodo.
- **Nodos de configuración:** Muestra los nodos de configuración de los flujos.
- **Panel de control:** Esta pestaña te permite establecer el diseño del tablero, la configuración del sitio y el tema.



Explicación

Nuestros nodos

Así que, ahora que sabes lo básico, vamos a introducir los nodos que vamos a utilizar:

- **Nodo Ui_button:** Añade un botón a la interfaz de usuario. Al hacer clic en el botón se genera un mensaje con msg.payload establecido en el campo Payload. Si no se especifica ningún payload, se utiliza el id del nodo.

- **Nodo de función:** Una función JavaScript que se ejecuta contra los mensajes que recibe el nodo. Los mensajes se pasan como un objeto JavaScript llamado msg. Por convención, tendrá una propiedad msg.payload que contiene el cuerpo del mensaje.

- **Nodo Exec:** Ejecuta un comando del sistema y devuelve su salida. El nodo puede ser configurado para esperar hasta que el comando se complete, o para enviar su salida a medida que el comando la genera. El comando que se ejecuta puede ser configurado en el nodo o proporcionado por el mensaje recibido.

- **Cambiar el nodo:** Establece, cambia, elimina o mueve las propiedades de un mensaje, contexto de flujo o contexto global. El nodo puede especificar múltiples reglas que se aplicarán en el orden en que se definan.

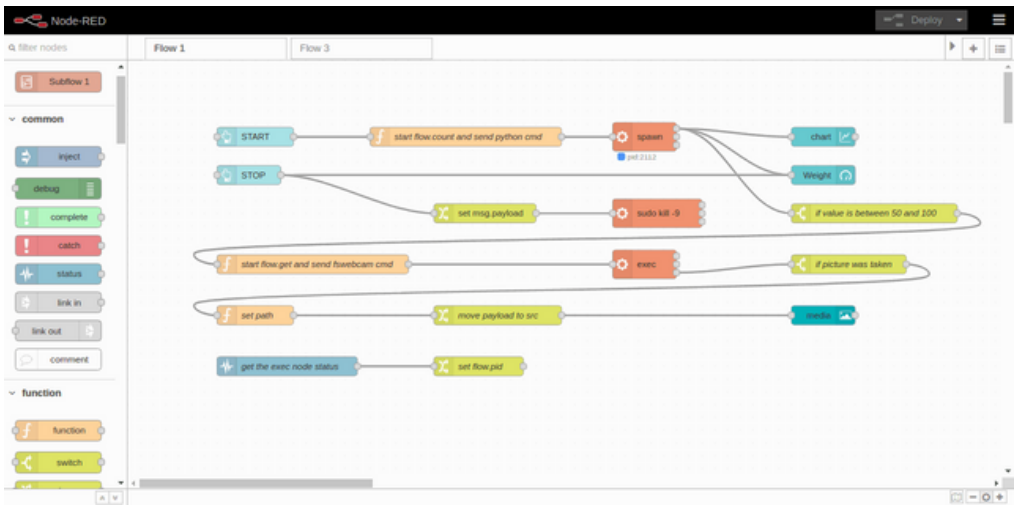
- **Nodo de cambio:** Encamina los mensajes en función de los valores de sus propiedades o de su posición en la secuencia.

- **Nodo Ui_chart:** Traza los valores de entrada en un gráfico. Puede ser un gráfico de líneas basado en el tiempo, un gráfico de barras (vertical u horizontal) o un gráfico circular.

- **Nodo Ui_gauge:** Añade un widget de tipo indicador a la interfaz de usuario. Se busca en el msg.payload un valor numérico y se formatea de acuerdo con el Formato de Valor definido.

- **Nodo Ui_media:** Muestra archivos multimedia y URLs en el panel de control.

- **Nodo Status:** Informa de los mensajes de estado de otros nodos en la misma pestaña.



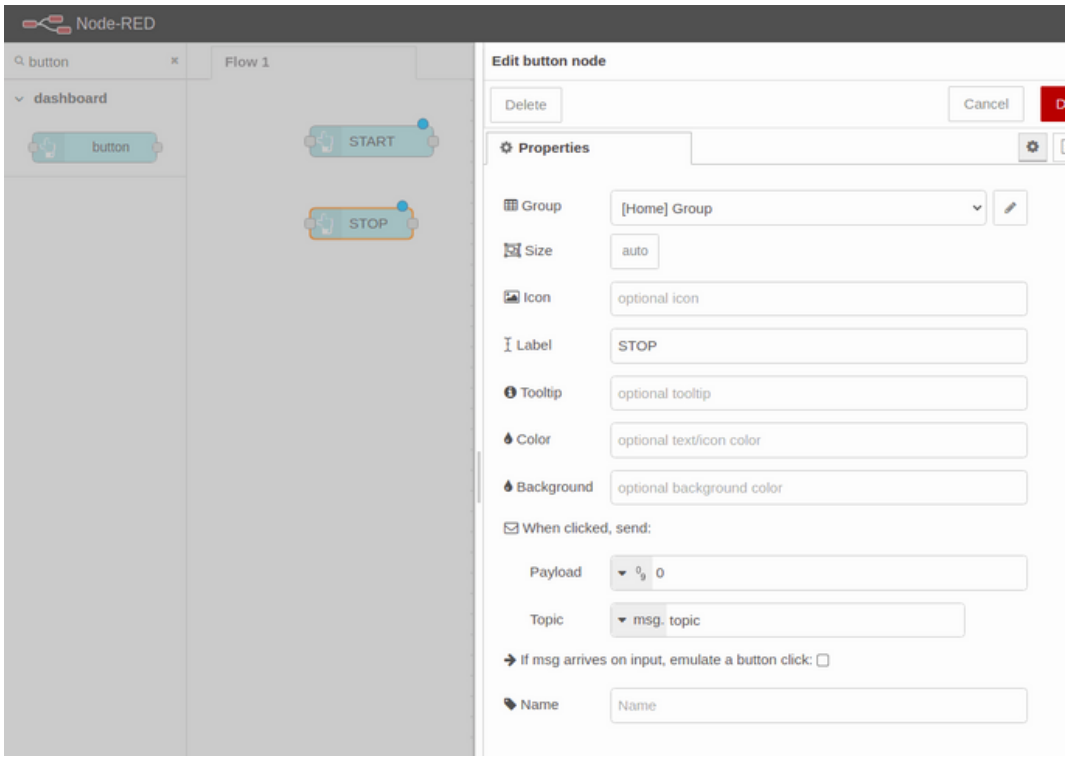
Explicación

Obtener el valor del peso

Lo que vamos a hacer es empezar a obtener los valores de un sensor de peso, y cuando la aplicación encuentre el valor que hemos establecido, la cámara USB tomará una foto.

Así que, ¡vamos a empezar a desarrollar nuestra aplicación!

1. En primer lugar, vas a añadir dos botones del panel de control: el primero para iniciar la aplicación, y el otro para detenerla.
Para ello, ve a la barra de búsqueda de los nodos de filtrado y escribe: botón. Añade dos botones al flujo, y haz doble clic para editarlos.
En el primero, debes crear un Grupo UI y una Pestaña UI para mostrar nuestro dashboard. Una vez hecho, funcionará para todos los nodos del Dashboard, por lo que sólo es necesario una vez. Después de eso, escribirá una etiqueta para ser mostrada, en nuestro caso: START.
Del mismo modo, el botón de parada tendrá la misma configuración; seleccionaremos el grupo y la pestaña donde queremos mostrarlo, escribiremos: STOP como etiqueta y añadiremos un 0 a la carga útil, para que el valor del indicador se ponga a 0 cuando la aplicación se detenga, en lugar de detenerse en el último valor.



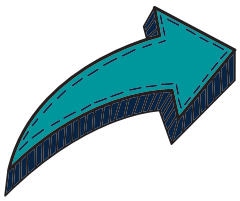
Explicación

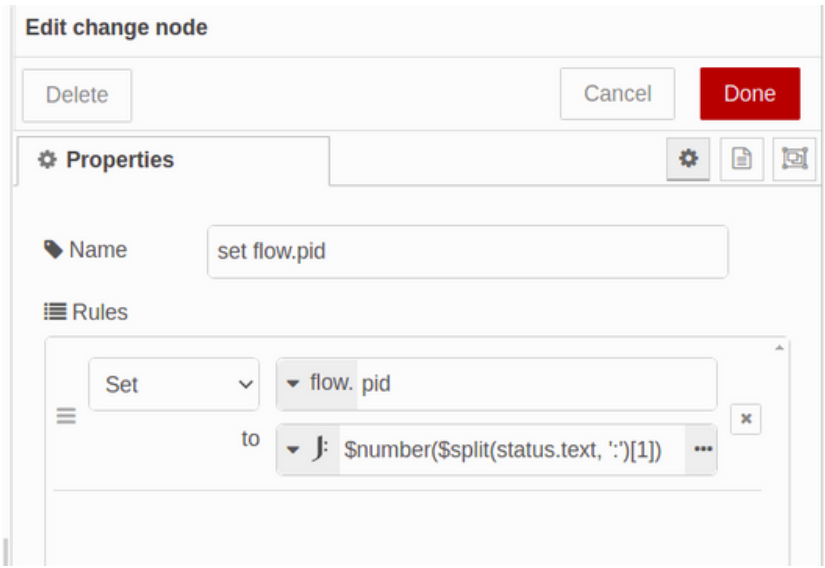
Obtener el valor del peso

2. Vas a añadir un nodo de función junto al botón de inicio y lo vas a cablear.
En el nodo de inicio, vas a inicializar una variable de flujo llamada count a 0, que vas a utilizar más adelante cuando nombres las imágenes, y vas a enviar el mensaje con el comando a ejecutar para que la app se inicie.

```
var count = flow.get('count')||0;
flow.set('count', count);
var newMsg = {payload: "python -u /home/pi/hx711py/example.py"};
return newMsg;
```

Puede dar un nombre al nodo de la función como le gustaría verlo en su flujo. En este caso: iniciar flow.count y enviar python cmd. Finalmente, cablea un nodo Exec y edítalo. Seleccione la salida: "mientras se ejecuta el comando - modo spawn", y haz clic en la casilla para añadir el msg.payload.

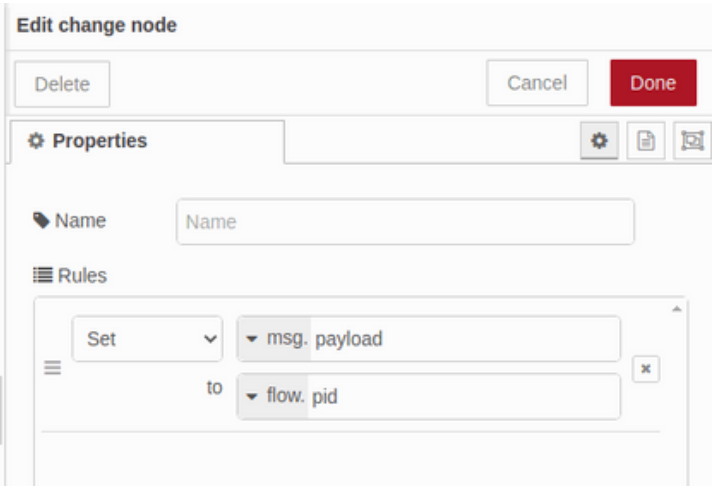




3. Cuando hay un nodo exec ejecutándose como modo spawn, eso genera un PID del proceso en ejecución, que tendrás que obtener para poder matarlo. Así que eso es lo que vas a hacer ahora.
Añade un nodo de estado, ve a "Reportar estado desde" y selecciona "Nodos seleccionados". Elija el nodo de ejecución, y haga clic en Hecho. Después de eso, cablea un nodo Change, y edítalo para establecer el flow.pid como se muestra a continuación:

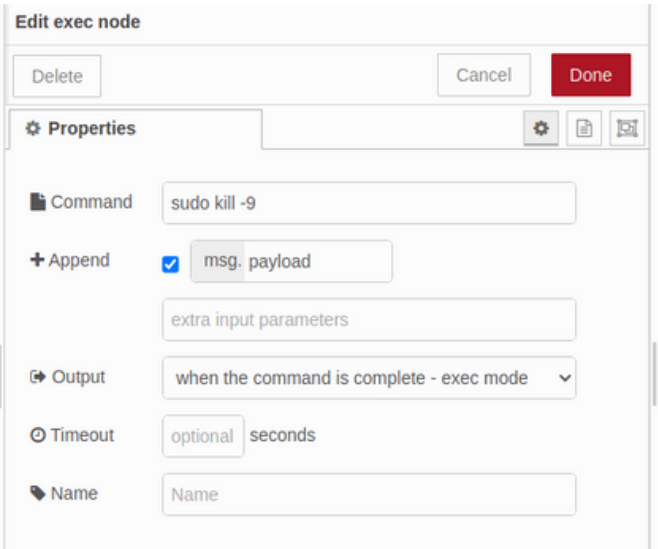
Explicación

Obtener el valor del peso



Finalmente, añade otro nodo de cambio junto al botón de parada y conéctalos. Como hemos establecido el flow.pid en el nodo de cambio anterior, ahora vamos a establecer el msg.payload al flow.pid. Haciendo esto, al pulsar el botón de parada, el msg.payload se enviará a través del nodo.

Entonces, ahora el pid es el msg.payload. Añade un nodo exec como modo exec para matar el pid, y edítalo:



Explicación

Obtener el valor del peso

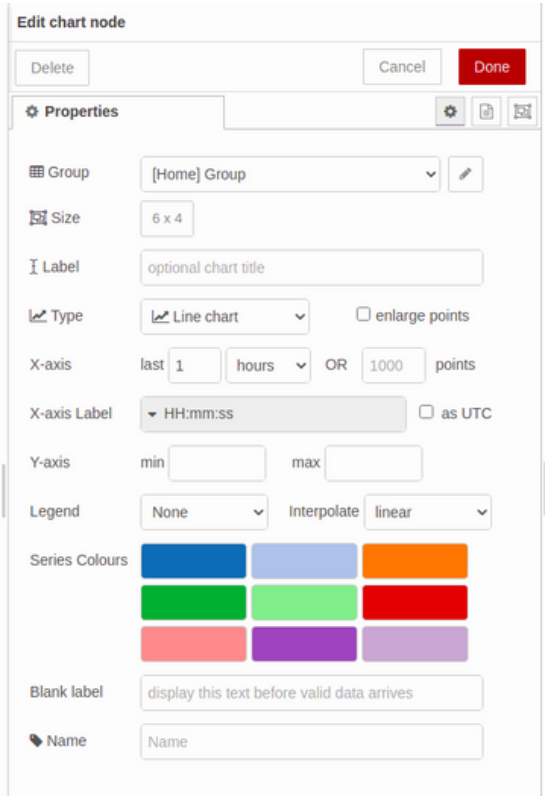
Por el momento, tu flujo se verá así:

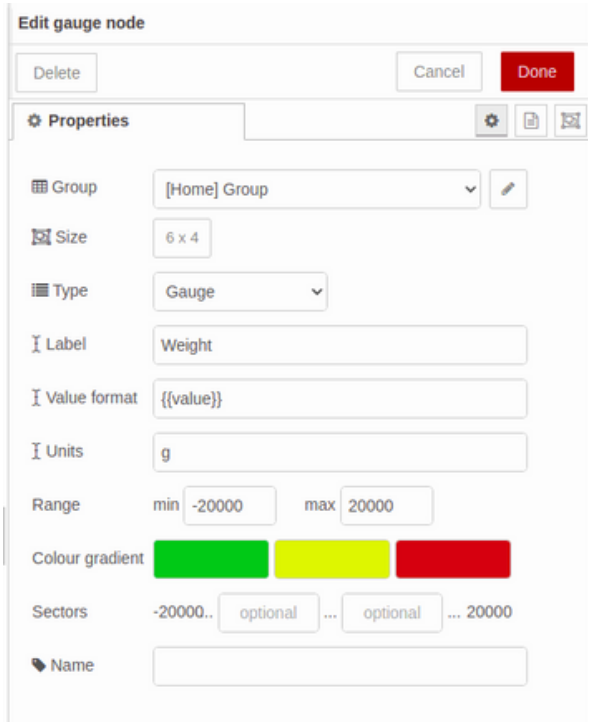

```
[{"id":"2826c4af.400f9c","type":"tab","label":"Flow
1","disabled":false,"info":""},
{"id":"bce0df4f.bc788","type":"ui_button","z":"2826c4af.400f9c","name":"","gro
up":"c4c1bcc1.49c24","order":16,"width":"7","height":"2","passthru":false,"lab
el":"START
","tooltip":"","color":"","bgcolor":"","icon":"","payload":"","payloadType":"s
tr","topic":"topic","topicType":"msg","x":140,"y":140,"wires":
[["882b392c.ab71b8"]]},
{"id":"222e70bc.56f6","type":"ui_button","z":"2826c4af.400f9c","name":"","grou
p":"c4c1bcc1.49c24","order":15,"width":0,"height":0,"passthru":false,"label":"
STOP","tooltip":"","color":"","bgcolor":"","icon":"","payload":"0","payloadTyp
e":"num","topic":"topic","topicType":"msg","x":130,"y":220,"wires":
[["63e42d5b.dee384"]]},
{"id":"882b392c.ab71b8","type":"function","z":"2826c4af.400f9c","name":"start
flow.count      and      send      python      cmd","func":"var      count      =
flow.get('count')||0;\nflow.set('count',  count);\n\nvar  newMsg  =  {payload:
\"python              -u              /home/pi/hx711py/example.py\"};\nreturn
newMsg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":
[],"x":410,"y":140,"wires":[["9628a2eb.2a5d3"]]},
{"id":"2abcf1ce.f1931e","type":"status","z":"2826c4af.400f9c","name":"","scope
":["9628a2eb.2a5d3"],"x":140,"y":60,"wires":[["b7fab428.f4fb78"]]},
{"id":"9628a2eb.2a5d3","type":"exec","z":"2826c4af.400f9c","command":"","addpa
y":"payload","append":"","useSpawn":"true","timer":"","oldrc":false,"name":"",
"x":690,"y":140,"wires":[[],[],[]]},
{"id":"b7fab428.f4fb78","type":"change","z":"2826c4af.400f9c","name":"","rules
":[{"t":"set","p":"pid","pt":"flow","to":"$number($split(status.text,      ':')
[1])","tot":"jsonata"}],"action":"","property":"","from":"","to":"","reg":fals
e,"x":410,"y":60,"wires":[[]]},
{"id":"63e42d5b.dee384","type":"change","z":"2826c4af.400f9c","name":"","rules
":
[{"t":"set","p":"payload","pt":"msg","to":"pid","tot":"flow"}],"action":"","pr
operty":"","from":"","to":"","reg":false,"x":420,"y":220,"wires":
[["46ba8b75.815004"]]},
{"id":"46ba8b75.815004","type":"exec","z":"2826c4af.400f9c","command":"sudo
kill
-9","addpay":"payload","append":"","useSpawn":"false","timer":"","oldrc":false
,"name":"","x":690,"y":220,"wires":[[],[],[]]},
{"id":"c4c1bcc1.49c24","type":"ui_group","name":"Group","tab":"cbda5f28.c75ad"
,"order":1,"disp":true,"width":"20","collapse":false},
{"id":"cbda5f28.c75ad","type":"ui_tab","name":"Home","icon":"dashboard","disab
led":false,"hidden":false}]
```

Explicación

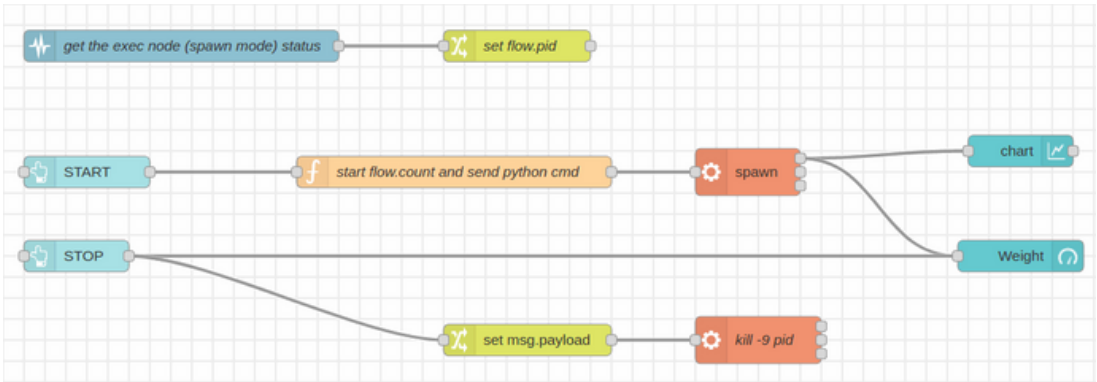
Obtener el valor del peso

4. Ahora, vas a ver los valores de la última 1 hora en un gráfico de líneas, y también en tiempo real en un indicador.
Entonces, arrastra y suelta un nodo de gráfico y un nodo de indicador, y vamos a editarlos.
En el nodo del gráfico, establece el eje X en la última hora, o el tiempo que quieras registrar, añade la pestaña y el grupo que quieras mostrar y haz clic en Hecho.
Edita el nodo del indicador eligiendo la misma pestaña y grupo y estableciendo una etiqueta para mostrar como su título, también escribe las unidades. Por último, establece el valor mínimo y el máximo para fijar el rango:





Finalmente, conéctalos como se muestra a continuación:



Explicación

Weight! ¡Haz una foto a esto!

Una vez que tienes los valores de nuestra báscula Raspberry y los has mostrado en tu Dashboard, es el momento de hacer algunas fotos.

Para los siguientes pasos es necesario instalar el node-red-contrib-ui-media, así que si no lo hiciste aún, por favor ve al último post para saber cómo: **https://www.industrialshields.com/es_ES/blog/raspberry-pi-for-industry-26/post/como-hacer-una-foto-cuando-se-detecta-un-valor-de-la-celula-de-carga-289**.

5. Ahora, vas a añadir un nodo interruptor y establecer si una propiedad está entre 50 y 100 para tomar una foto. Los valores dependen de ti, sólo tienes que elegir las reglas de valor, elegir el campo de número, y añadir el número que quieres presentar. Conecta este nodo con el nodo spawn.

6. Conectado a la salida del último nodo de cambio, añada un nodo de función para enviar el comando fswebcam y configure el flujo.count para nombrar las fotos con un contador de la siguiente manera:

```
var count = flow.get('count');
count++;

msg.payload = "fswebcam -r 1280x720 --no-banner /home/pi/images/image" + count +
".jpg";

flow.set('count', count);
return msg;
```

Debes añadir tres parámetros al comando fswebcam:ç

1. **- r** para establecer la resolución de la foto.
2. **--no-banner** para omitir el banner de la cámara
3. La **ruta** para decir dónde guardar las imágenes, y cómo se van a nombrar.

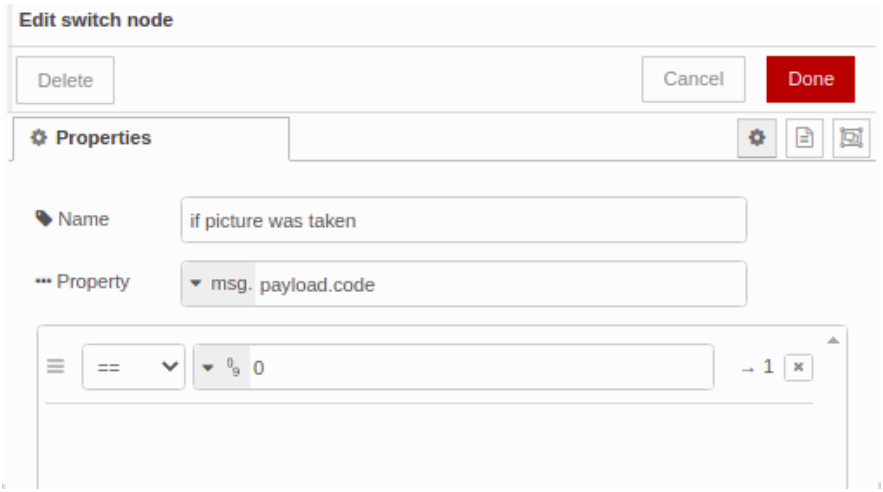
7. El nodo de la función enviará un msg.payload, por lo que vas a añadir un nodo exec anexando el msg.payload para ejecutar el comando en tu controlador industrial Raspberry Pi PLC.

Explicación

Weight! ¡Haz una foto a esto!

8. El modo exec tiene tres salidas. La primera devuelve la stdout, la segunda devuelve la stderr y la última, devuelve el código de retorno. Así que en este caso, conecta la tercera salida, el código de retorno, a un nodo switch para continuar con el flujo si no hubo error.

Así que, en el nodo switch, establece la propiedad a msg.payload.code y establece la regla de valor igual al número 0, para estar seguro de que el comando fwwebcam se ejecutó sin errores.



9. Después, conecte un nodo de función para enviar el nombre de la foto que se acaba de tomar para que se pueda mostrar en el Tablero de Control de Node-RED. Una vez editado como se muestra a continuación, conecte un nodo de cambio para mover el msg.payload a msg.src:

```
let count = flow.get('count');
msg.payload = "/image" + count + ".jpg";
return msg;
```

Explicación

Weight! ¡Haz una foto a esto!

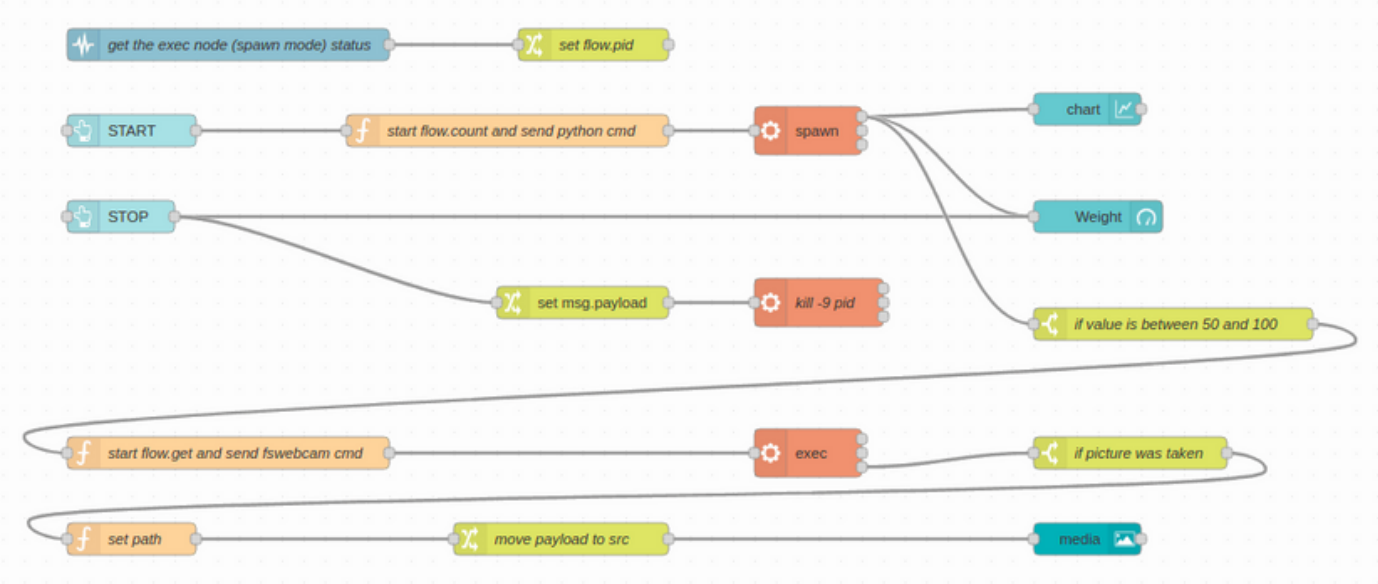
10. Por último, añade el nodo de los medios de comunicación y simplemente añade un Grupo a él, o configura el diseño como quieras.


```
[{"id":"2826c4af.400f9c","type":"tab","label":"Flow 1","disabled":false,"info":""},
{"id":"9b234a13.0256e8","type":"exec","z":"2826c4af.400f9c","command":"","addpay":"payload","append":"","useSpawn":"false","timer":"","oldrc":false,"name":"","x":510,"y":260,"wires":[[],[],["3f27e8b4.d02378"]]},
{"id":"ec5481a.4fbf28","type":"exec","z":"2826c4af.400f9c","command":"","addpay":"payload","append":"","useSpawn":"true","timer":"","oldrc":false,"name":"","x":870,"y":60,"wires":[["36307784.3144e8","372e8f7b.f9752","d4e34cd5.f423e"],[],[]]},
{"id":"36307784.3144e8","type":"switch","z":"2826c4af.400f9c","name":"if value is between 50 and 100","property":"payload","propertyType":"msg","rules":[{"t":"btwn","v":"50","vt":"num","v2":"100","v2t":"num"}],"checkall":"true","repair":false,"outputs":1,"x":990,"y":160,"wires":[["fb666c7f.c2ff9"]]},
{"id":"3f27e8b4.d02378","type":"switch","z":"2826c4af.400f9c","name":"if picture was taken","property":"payload.code","propertyType":"msg","rules":[{"t":"eq","v":"0","vt":"num"}],"checkall":"true","repair":false,"outputs":1,"x":1030,"y":260,"wires":[["7646b60e.83a318"]]},
{"id":"7646b60e.83a318","type":"function","z":"2826c4af.400f9c","name":"set path","func":"let count = flow.get('count');\nmsg.payload = \"/image\" + count + \".jpg\";\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":140,"y":360,"wires":[["a4078c82.e803a"]]},{"id":"a4078c82.e803a","type":"change","z":"2826c4af.400f9c","name":"move payload to src","rules":[{"t":"move","p":"payload","pt":"msg","to":"src","tot":"msg"}],"action":"","property":"","from":"","to":"","reg":false,"x":560,"y":360,"wires":[["3385b59c.06c81a"]]},
{"id":"8771f8be.e44f68","type":"ui_button","z":"2826c4af.400f9c","name":"","group":"c4c1bcc1.49c24","order":5,"width":7,"height":2,"passthru":false,"label":"STOP LOAD CELL","tooltip":"","color":"","bgcolor":"","icon":"","payload":"0","payloadType":"num","topic":"","topicType":"str","x":170,"y":160,"wires":[["e053ecae.bca31","d4e34cd5.f423e"]]},
{"id":"d4e34cd5.f423e","type":"ui_gauge","z":"2826c4af.400f9c","name":"","group":"c4c1bcc1.49c24","order":13,"width":6,"height":4,"gtype":"gage","title":"Weight","label":"g","format":"{{value}}","min":"-2000","max":"2000","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":1070,"y":100,"wires":[]},
{"id":"372e8f7b.f9752","type":"ui_chart","z":"2826c4af.400f9c","name":"","group":"c4c1bcc1.49c24","order":11,"width":6,"height":4,"label":"","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpolate":"linear","nodata":"","dot":false,"ymin":"","ymax":"","removeOlder":1,"removeOlderPoints":"","removeOlderUnit":"3600","cutout":0,"useOneColor":false,"useUTC":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"outputs":1,"useDifferentColor":false,"x":1070,"y":60,"wires":[[]]},
{"id":"99be7e29.78696","type":"ui_button","z":"2826c4af.400f9c","name":"","group":"c4c1bcc1.49c24","order":3,"width":7,"height":2,"passthru":false,"label":"START LOAD CELL","tooltip":"","color":"","bgcolor":"","icon":"","payload":"","payloadType":"str","topic":"","topicType":"str","x":180,"y":60,"wires":[["615b1ef6.53963"]]},
{"id":"615b1ef6.53963","type":"function","z":"2826c4af.400f9c","name":"start flow.count and send python cmd","func":"var count = flow.get('count')||0;\nflow.set('count', count);\n\nvar newMsg = {payload: \"sudo python -u /home/pi/hx711py/example.py\"};\nreturn newMsg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":610,"y":60,"wires":[["ec5481a.4fbf28"]]},
{"id":"3385b59c.06c81a","type":"ui_media","z":"2826c4af.400f9c","group":"c4c1bcc1.49c24","name":"","width":6,"height":4,"order":15,"category":"","file":"","layout":"expand","showcontrols":true,"loop":true,"onstart":false,"scope":"local","tooltip":"","x":1070,"y":360,"wires":[[]]},
{"id":"16de31a2.e4a6de","type":"status","z":"2826c4af.400f9c","name":"get the exec node status","scope":["ec5481a.4fbf28"],"x":190,"y":600,"wires":[["9bb820b3.87fbe"]]},
{"id":"9bb820b3.87fbe","type":"change","z":"2826c4af.400f9c","name":"set flow.pid","rules":[{"t":"set","p":"pid","pt":"flow","to":"$number($split(status.text, ':')[1])","tot":"jsonata"}],"action":"","property":"","from":"","to":"","reg":false,"x":410,"y":600,"wires":[["b99c988c.86bf98"]]},
{"id":"b99c988c.86bf98","type":"function","z":"2826c4af.400f9c","name":"set kill cmd","func":"let pid = flow.get('pid');\nvar kill = \"kill -9 \" + pid;\nflow.set('kill', kill);\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":590,"y":600,"wires":[[]]},
{"id":"e053ecae.bca31","type":"function","z":"2826c4af.400f9c","name":"killall python","func":"msg.payload = \"sudo killall python\";\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":530,"y":160,"wires":[["ec5481a.4fbf28"]]},{"id":"fb666c7f.c2ff9","type":"function","z":"2826c4af.400f9c","name":"start flow.get and send fswebcam cmd","func":"var count = flow.get('count');\ncount++;\n\nmsg.payload = \"fswebcam -r 1280x720 --no-banner /home/pi/images/image\" + count + \".jpg\";\n\nflow.set('count', count);\n\nreturn msg;\n\n","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":230,"y":260,"wires":[["9b234a13.0256e8"]]},{"id":"5771d86a.220b58","type":"comment","z":"2826c4af.400f9c","name":"In case you want to kill the flow pid and not the python processes, replace the \"killall python\" function node, for the \"killall pid\" function node -->","info":"","x":550,"y":540,"wires":[]},
{"id":"b88b67eb.03f068","type":"function","z":"2826c4af.400f9c","name":"killall pid","func":"msg.payload = flow.get('kill');\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":[],"x":1100,"y":540,"wires":[[]]},
{"id":"c4c1bcc1.49c24","type":"ui_group","name":"","tab":"cbda5f28.c75ad","order":1,"disp":true,"width":"20","collapse":false},
{"id":"cbda5f28.c75ad","type":"ui_tab","name":"Home","icon":"dashboard","disabled":false,"hidden":false}]
```

Explicación

Weight! ¡Haz una foto a esto!

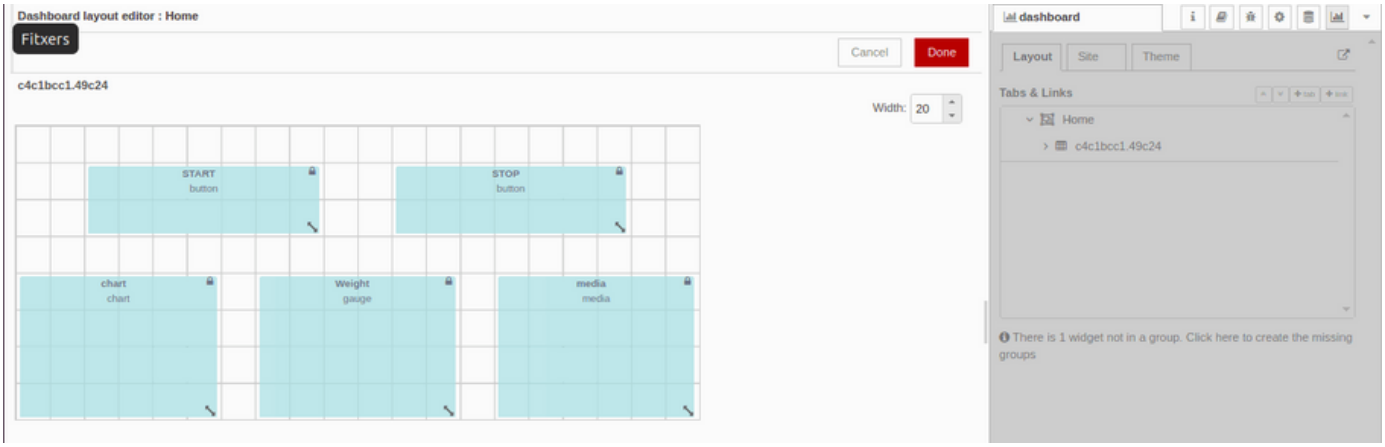
Ahora, tu aplicación Node-RED debería tener este aspecto:



Explicación

Consejos

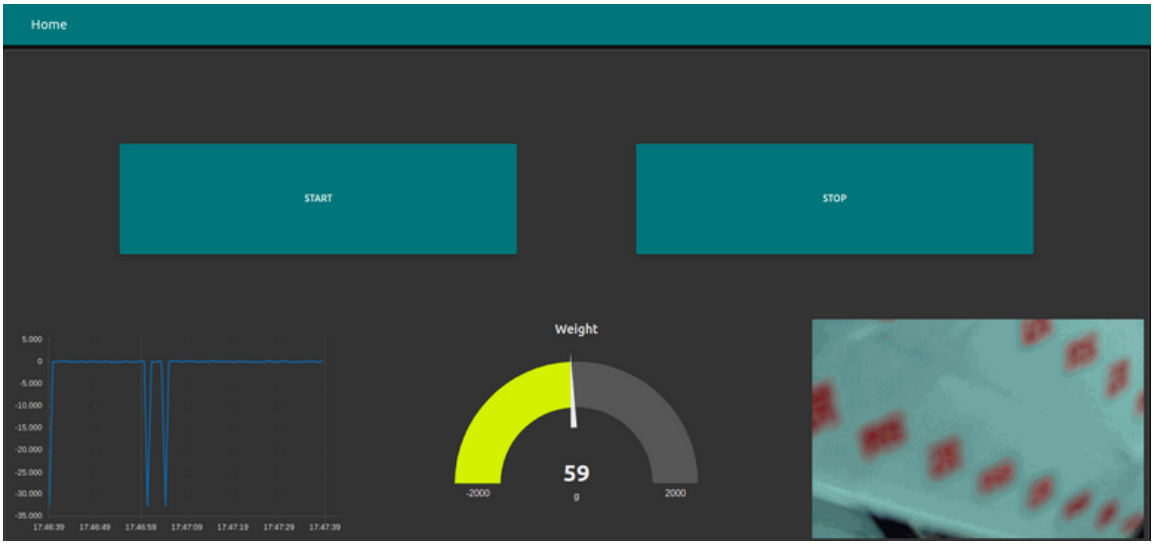
Si vas al menú de la derecha, en la pestaña Dashboard, y pasas el ratón por encima de tu pestaña, verás aparecer tres botones: group, edit y layout. Así, si haces clic en diseño, verás el editor de diseño del Dashboard, donde es posible mostrar tus nodos ui como quieras.



Si ves que no puedes cambiar el **tamaño de los widgets**, ve a cada nodo del Dashboard, y en la sección de Size, verás que está configurado como automático, así que simplemente establece cualquier tamaño manual, vuelve al editor de diseño del Dashboard, donde se aplicarán los cambios.

Finalmente, ve a http://10.10.10.20:1880/ui/ para comprobar tu panel de control Node-RED.

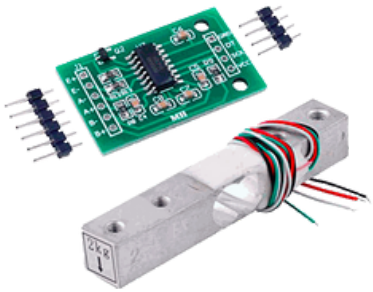
ERR_CONNECTION_TIMED_OUT



Cómo hacer una foto cuando se detecta un valor de la célula de carga

Aplicaciones industriales Raspberry PLC y Node-RED

Introducción



Los sensores vestibles son un mercado emergente que está ganando rápidamente reconocimiento en el mercado global, especialmente en el sector industrial.
La familia de PLCs basados en Raspberry Pi ofrece un amplio abanico de posibilidades para controlar y supervisar esta innovadora tecnología.
En esta sección, aprenderás a tomar una foto usando una cámara USB cuando una célula de carga detecta un valor específico, controlado por un PLC industrial Raspberry Pi y monitorizándolo usando Node-RED.

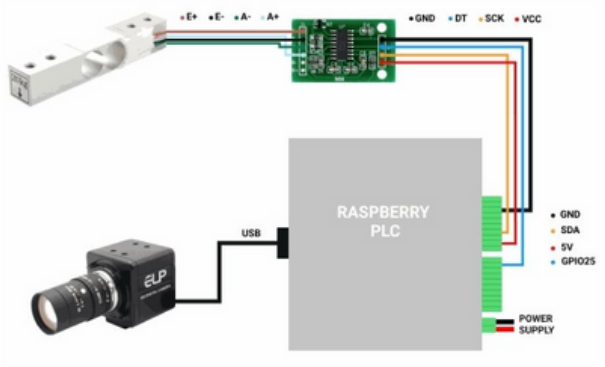
Explicación

Requisitos

- 1x Sensor de peso
- 1x Módulo transmisor de célula de carga HX711
- 4x cables
- 1x cámara USB
- 1x PLC industrial Raspberry Pi

Conexión del hardware

Conecta el hardware como se muestra a continuación para proceder con el software:



WEIGHTING SENSOR	HX711	RASPBERRY PLC	USB CAMERA
RED CABLE	E+		
WHITE CABLE	A+		
BLACK CABLE	E-		
GREEN CABLE	A-		
	GND	GND	
	DT	GPIO 25	
	SCK	SDA	
	VCC	5V	
		USB	USB CABLE

Explicación

Configuración del sensor de peso

1. Primero, vamos a clonar un proyecto que contenga un archivo de ejemplo que muestre una función de la librería. Para ello, abre una ventana de terminal en tu controlador PLC Raspberry Pi y escribe lo siguiente:

```
git clone https://github.com/tatobari/hx711py
```

2. Una vez clonado el repositorio, aparecerá un directorio llamado hx711py con el archivo llamado example.py. Por lo tanto, ve a ese archivo para ajustar algunos cambios:

```
cd hx711py
sudo nano example.py
```

3. Dentro del archivo, modifiquemos algunas líneas, para que el código tenga el siguiente aspecto:

```
#! /usr/bin/python2

import time
import sys

EMULATE_HX711=False

referenceUnit = -1

if not EMULATE_HX711:
    import RPi.GPIO as GPIO
    from hx711 import HX711
else:
    from emulated_hx711 import HX711

def cleanAndExit():
    print("Cleaning...")

    if not EMULATE_HX711:
        GPIO.cleanup()

    print("Bye!")
    sys.exit()
```

Explicación

Configuración del sensor de peso

```
hx = HX711(25, 2)
hx.set_reference_unit(referenceUnit)
hx.reset()

hx.tare()

print("Tare done! Add weight now...")

def func():
    while True:
 try:
 val = hx.get_weight(5)
 yield val
 hx.power_down()
  hx.power_up()
  time.sleep(0.1)

 except (KeyboardInterrupt, SystemExit):
        cleanAndExit()

function = func()
for i in function:
    print(i)
```

4. Una vez modificado el código, sal con Ctrl + X, escribe 'Y' para guardar el archivo con el mismo nombre y Enter.

Prueba de la báscula Raspberry

1. Para una correcta calibración y poder obtener el peso correcto, se necesita un objeto de comparación cuyo peso se conozca. Se recomienda elegir un valor medio del máximo que puede obtener la célula de carga. Por ejemplo, si su báscula puede obtener hasta 20 kgs, entonces se recomienda elegir un objeto cuyo peso sea de 10 kgs.

2. En primer lugar, es necesario comentar la siguiente línea de la siguiente manera:

```
#hx.set_reference_unit(referenceUnit)
```

Explicación

Prueba de la báscula Raspberry

3. A continuación, coloca el objeto en la balanza y ejecútalo con el siguiente comando:

```
sudo python example.py
```

4. Verás que los valores mostrados serán tanto positivos como negativos. En este caso, se muestran en 24500 valores alrededor de -22200. Así que referenciamos los valores como:

$$-22200 \div 24500 = -0.9$$

5. Después de obtener ese valor, vuelve a la línea que comentamos antes, y descoméntala eliminando el hashtag, y escribiendo el valor que obtuviste como unidad de referencia como:

```
hx.set_reference_unit(referenceUnit)
referenceUnit = -1
```

Si has conseguido una calibración correcta, verás que el valor de la balanza de peso estará en torno a 0. Si no, puedes modificar la variable referenceUnit para conseguir la calibración correcta.

Obtener los valores con Node-RED

1. Por defecto, nuestras imágenes del PLC Raspberry tienen Node-RED ya instalado. Si no tienes el Node-RED instalado en tu dispositivo, primero ve a la siguiente URL para proceder a la instalación: **https://nodered.org/docs/getting-started/raspberrypi**

2. Una vez que Node-RED esté listo, desde tu ordenador ve a tu navegador y escribe: **http://10.10.10.20:1880/** si estás conectado a través de Ethernet, o **http://wlan0-pi-address:1880/** si quieres conectarte a través de WiFi.

3. Después de abrir una nueva ventana de Node-RED, ve al menú de la derecha, haz clic en **Gestionar paleta > Instalar > Tipo: node-red-contrib-ui-media e instálalo.**

Explicación

Obtener los valores con Node-RED

4. Después de eso, haz clic en Importar, y pega el siguiente flows.json y haz clic en el botón Desplegar:


```
[{"id":"2826c4af.400f9c","type":"tab","label":"Flow 1","disabled":false,"info":""},
{"id":"9b234a13.0256e8","type":"exec","z":"2826c4af.400f9c","command":"","addpay":"payloa
d","append":"","useSpawn":"false","timer":"","oldrc":false,"name":"","x":510,"y":260,"wir
es":[[],[],["3f27e8b4.d02378"]]},
{"id":"ec5481a.4fbf28","type":"exec","z":"2826c4af.400f9c","command":"","addpay":"payload
","append":"","useSpawn":"true","timer":"","oldrc":false,"name":"","x":870,"y":60,"wires"
:[["36307784.3144e8","372e8f7b.f9752","d4e34cd5.f423e"],[],[]]},
{"id":"36307784.3144e8","type":"switch","z":"2826c4af.400f9c","name":"if value is between
50 and 100","property":"payload","propertyType":"msg","rules":
[{"t":"btwn","v":"50","vt":"num","v2":"100","v2t":"num"}],"checkall":"true","repair":fals
e,"outputs":1,"x":990,"y":160,"wires":[["fb666c7f.c2ff9"]]},
{"id":"3f27e8b4.d02378","type":"switch","z":"2826c4af.400f9c","name":"if picture was
taken","property":"payload.code","propertyType":"msg","rules":
[{"t":"eq","v":"0","vt":"num"}],"checkall":"true","repair":false,"outputs":1,"x":1030,"y"
:260,"wires":[["7646b60e.83a318"]]},
{"id":"7646b60e.83a318","type":"function","z":"2826c4af.400f9c","name":"set
path","func":"let count = flow.get('count');\nmsg.payload = \"/image\" + count +
\".jpg\";\nreturn msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":
[],"x":140,"y":360,"wires":[["a4078c82.e803a"]]},
{"id":"a4078c82.e803a","type":"change","z":"2826c4af.400f9c","name":"move payload to
src","rules":
[{"t":"move","p":"payload","pt":"msg","to":"src","tot":"msg"}],"action":"","property":"",
"from":"","to":"","reg":false,"x":560,"y":360,"wires":[["3385b59c.06c81a"]]},
{"id":"8771f8be.e44f68","type":"ui_button","z":"2826c4af.400f9c","name":"","group":"c4c1b
cc1.49c24","order":5,"width":7,"height":2,"passthru":false,"label":"STOP LOAD
CELL","tooltip":"","color":"","bgcolor":"","icon":"","payload":"0","payloadType":"num","t
opic":"","topicType":"str","x":170,"y":160,"wires":
[["e053ecae.bca31","d4e34cd5.f423e"]]},
{"id":"d4e34cd5.f423e","type":"ui_gauge","z":"2826c4af.400f9c","name":"","group":"c4c1bcc
1.49c24","order":13,"width":6,"height":4,"gtype":"gage","title":"Weight","label":"g","for
mat":"{{value}}","min":"-2000","max":"2000","colors":
["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":1070,"y":100,"wires":[]},
{"id":"372e8f7b.f9752","type":"ui_chart","z":"2826c4af.400f9c","name":"","group":"c4c1bcc
1.49c24","order":11,"width":6,"height":4,"label":"","chartType":"line","legend":"false","
xformat":"HH:mm:ss","interpolate":"linear","nodata":"","dot":false,"ymin":"","ymax":"","r
emoveOlder":1,"removeOlderPoints":"","removeOlderUnit":"3600","cutout":0,"useOneColor":fa
lse,"useUTC":false,"colors":
["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5
"],"outputs":1,"useDifferentColor":false,"x":1070,"y":60,"wires":[[]]},
{"id":"99be7e29.78696","type":"ui_button","z":"2826c4af.400f9c","name":"","group":"c4c1bc
c1.49c24","order":3,"width":7,"height":2,"passthru":false,"label":"START LOAD
CELL","tooltip":"","color":"","bgcolor":"","icon":"","payload":"","payloadType":"str","to
pic":"","topicType":"str","x":180,"y":60,"wires":[["615b1ef6.53963"]]},
{"id":"615b1ef6.53963","type":"function","z":"2826c4af.400f9c","name":"start flow.count
and send python cmd","func":"var count = flow.get('count')||0;\nflow.set('count',
count);\n\nvar newMsg = {payload: \"sudo python -u
/home/pi/hx711py/example.py\"};\nreturn
newMsg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":
[],"x":610,"y":60,"wires":[["ec5481a.4fbf28"]]},
```

Explicación

Obtener los valores con Node-RED


```
{"id":"3385b59c.06c81a","type":"ui_media","z":"2826c4af.400f9c","group":"c4c1bcc1.49
c24","name":"","width":6,"height":4,"order":15,"category":"","file":"","layout":"exp
and","showcontrols":true,"loop":true,"onstart":false,"scope":"local","tooltip":"","x
":1070,"y":360,"wires":[[]]},
{"id":"16de31a2.e4a6de","type":"status","z":"2826c4af.400f9c","name":"get the exec
node status","scope":["ec5481a.4fbf28"],"x":190,"y":600,"wires":
[["9bb820b3.87fbe"]]},
{"id":"9bb820b3.87fbe","type":"change","z":"2826c4af.400f9c","name":"set
flow.pid","rules":
[{"t":"set","p":"pid","pt":"flow","to":"$number($split(status.text, ':')
[1])","tot":"jsonata"}],"action":"","property":"","from":"","to":"","reg":false,"x":
410,"y":600,"wires":[["b99c988c.86bf98"]]},
{"id":"b99c988c.86bf98","type":"function","z":"2826c4af.400f9c","name":"set kill
cmd","func":"let pid = flow.get('pid');\nvar kill = \"kill -9 \" +
pid;\nflow.set('kill', kill);\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":
[],"x":590,"y":600,"wires":[[]]},
{"id":"e053ecae.bca31","type":"function","z":"2826c4af.400f9c","name":"killall
python","func":"msg.payload = \"sudo killall python\";\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":
[],"x":530,"y":160,"wires":[["ec5481a.4fbf28"]]},
{"id":"fb666c7f.c2ff9","type":"function","z":"2826c4af.400f9c","name":"start
flow.get and send fswebcam cmd","func":"var count =
flow.get('count');\ncount++;\n\nmsg.payload = \"fswebcam -r 1280x720 --no-banner
/home/pi/images/image\" + count + \".jpg\";\n\nflow.set('count', count);\n\nreturn
msg;\n\n","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":
[],"x":230,"y":260,"wires":[["9b234a13.0256e8"]]},
{"id":"5771d86a.220b58","type":"comment","z":"2826c4af.400f9c","name":"In case you
want to kill the flow pid and not the python processes, replace the \"killall
python\" function node, for the \"killall pid\" function node --
>","info":"","x":550,"y":540,"wires":[]},
{"id":"b88b67eb.03f068","type":"function","z":"2826c4af.400f9c","name":"killall
pid","func":"msg.payload = flow.get('kill');\nreturn
msg;","outputs":1,"noerr":0,"initialize":"","finalize":"","libs":
[],"x":1100,"y":540,"wires":[[]]},
{"id":"c4c1bcc1.49c24","type":"ui_group","name":"","tab":"cbda5f28.c75ad","order":1,
"disp":true,"width":"20","collapse":false},
{"id":"cbda5f28.c75ad","type":"ui_tab","name":"Home","icon":"dashboard","disabled":f
alse,"hidden":false}]
```


5. Ve a tu terminal Raspberry de nuevo, y escribe:

```
cd
mkdir images
sudo nano /home/pi/.node-red/settings.js
```

- ***cd:*** Lleva al directorio /home/pi.
- ***mkdir imágenes:*** Este comando crea un nuevo directorio llamado 'images' en el directorio actual. Este directorio almacenará todas las imágenes tomadas por la cámara.

Por último, vamos a modificar una línea para poder obtener las imágenes de una fuente.

Explicación

Obtener los valores con Node-RED

6. Encuentra la línea con el parámetro httpStatic y haz lo siguiente:

```
//httpStatic:   '/home/nol/node-red-static/',   ------------>   Replace
this
httpStatic: '/home/pi/images',                ------------> For this
```

7. Reinicia tu PLC de código abierto Raspberry Pi.

8. Después de reiniciar, ve de nuevo a tu navegador y escribe http://10.10.10.20:1880/ui.

9. Si todo ha ido bien, puedes probar y disfrutar de tu aplicación.

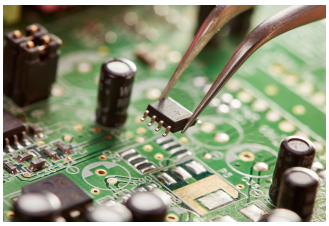
Enlaces relacionados

- Cómo conectar un Raspberry PLC al Wi-Fi
- Conceptos básicos sobre las salidas analógicas de un Raspberry Pi PLC
- Cómo encontrar tu PLC industrial perfecto
- Cómo programar en Python las entradas de interrupción del PLC industrial Raspberry
- Familia de productos Raspberry PLC
- Familia de productos TouchBerry Pi

Cómo conectar el PLC industrial Raspberry al Wi-Fi

Comunicaciones industriales

Introducción



Los dispositivos de la familia PLC basados en Raspberry Pi tienen conectividad inalámbrica Wi-Fi por defecto.
Utiliza las dos frecuencias más comunes: 2,4GHz y 5GHz. También utiliza las bandas IEEE 802.11.b/g/n/ac. Y, para conectarlos a la red Wi-Fi, hay que seguir unos pasos específicos.

Requisitos

Los puntos clave a tener en cuenta son los siguientes:

- Familia de PLCs industriales Raspberry Pi
- **Acceso al PLC:** shh. Un tutorial sobre cómo acceder al dispositivo a través de Linux o Windows se puede encontrar en la Guía del usuario del Raspberry Pi PLC controller User Guide

Explicación

Para conectar este PLC con su red Wi-Fi, debe modificar el archivo wpa_supplicant dentro de la Raspberry:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Y hay que configurar el archivo con los parámetros de configuración de la red Wi-Fi (puede cambiar dependiendo de la configuración específica de cada caso):

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=GB

network={
ssid="NETWORK SSID"
psk="NETWORK PASSWORD"
key_mgmt=WPA-PSK
}
```

Explicación

Y eso es todo, después debes reiniciar los servicios de red para aplicar los cambios:

```
sudo service networking restart
```

O puede reiniciar el sistema para aplicar los cambios:

```
sudo reboot
```

Para conectar este PLC con tu red Wi-Fi, debes modificar el archivo wpa_supplicant dentro de la Raspberry:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Y hay que configurar el archivo con los parámetros de configuración de la red Wi-Fi (puede cambiar dependiendo de la configuración específica de cada caso):

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=GB

network={
ssid="NETWORK SSID"
psk="NETWORK PASSWORD"
key_mgmt=WPA-PSK
}
```

Industrial Shields®

(+34) 938 760 191

info@industrialshields.com

Camí del Grau, 25
08272 Sant Fruitós de Bages (Barcelona)
Spain

www.industrialshields.com