Using the LED Matrix

The LED Matrix is a 16 x 32 LED display that stands in front of TimeSquAir.
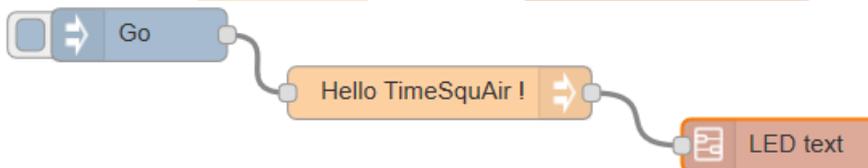
This tutorial shows how to use the provided nodes to use that display.

The LED Matrix "subflow"

A **Subflow** is a node made with a flow. There is two ways to make a node: one way is to use the javascript programming language, the second way is to use a flow.

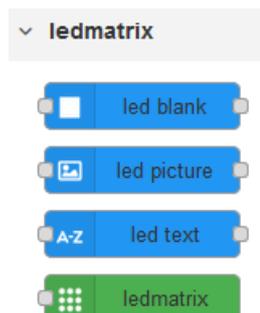To easily address the most frequent use, there is a orange node that does the job of displaying on the matrix:

You can use this node to display simple texts.

To do more, as mixing text and pictures, use the nodes below.

The LED Matrix nodes

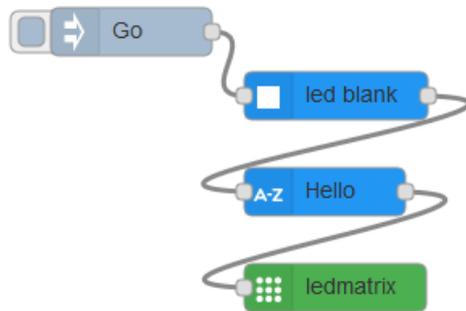You can find the LED Matrix nodes in the palette:

First use a LED Blank node to start. Then use a mix of LED Text and LED Picture nodes to compose the message. End with a LED

Matrix node and use a Go node to activate the whole.

Display a simple text

To display a simple text, a LED Text node is enough:

Double click the LED text node to edit the text to be shown.

See later if the text to display should be calculated.

Display a picture

Add the picture file on TimeSquAir

In order to add a picture on the LED matrix, you have to copy the picture on TimeSquAir.

Just get the destination folder using Samba: in your PC file browser, open:

```
\\myTimeSquAir\TheThingBox\root\userdir
```

where "myTimeSquAir" should be replaced by the name of your TimeSquAir.

Windows 10 asks you for credentials? Follow the following tutorial to fix your network: **https://tinkertry.com/how-to-change-windows-10-network-type-from-public-to-private**

You can do it by opening a file explorer window and changing its address by this one in the address bar. In the picture below, I access my TimeSquAir which name is "ttbbeacon" like that:
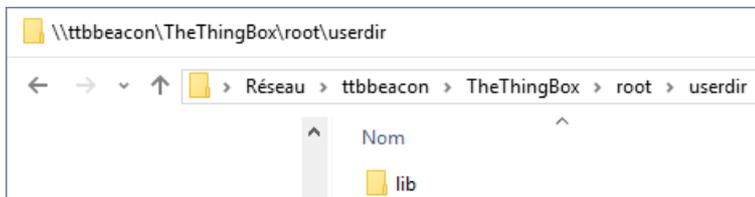
In you userdir, find or create a folder named `pictures` and put your pictures in it. You can use `png`, `jpg` or `ppm` images.
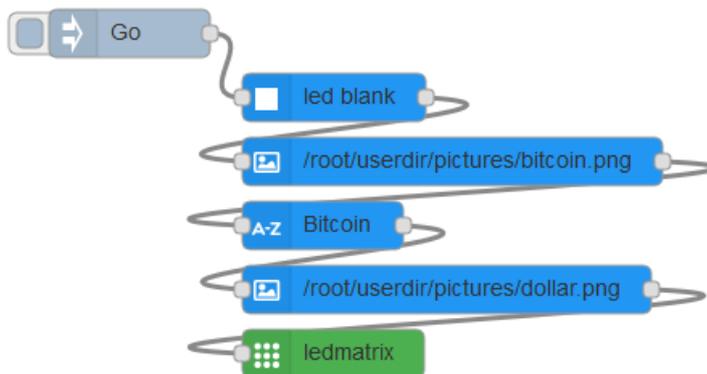
Modify the flow

Add a LEDPicture node and fill it with the path of the picture:
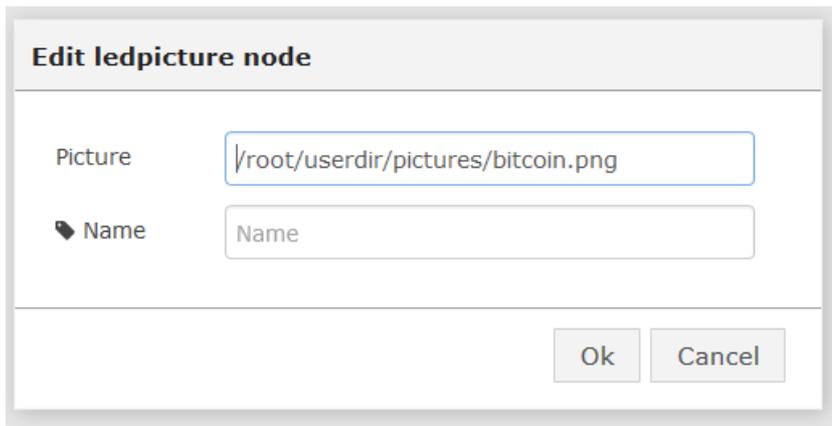
and insert it in the flow.

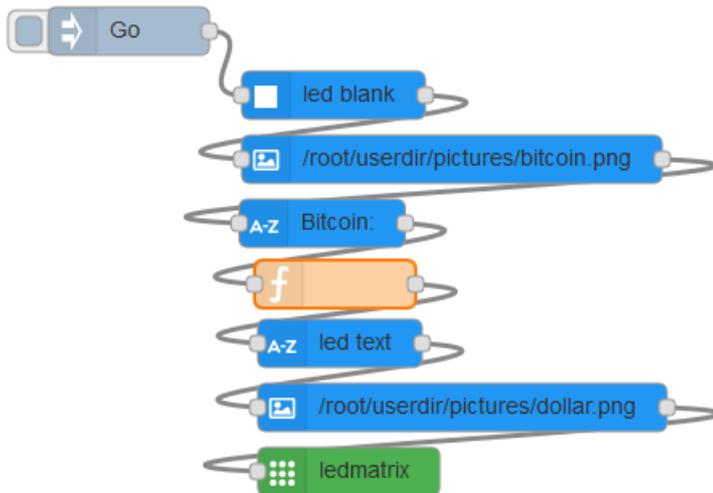Below is an example of mixed text and pictures:

⊚ Using calculated values

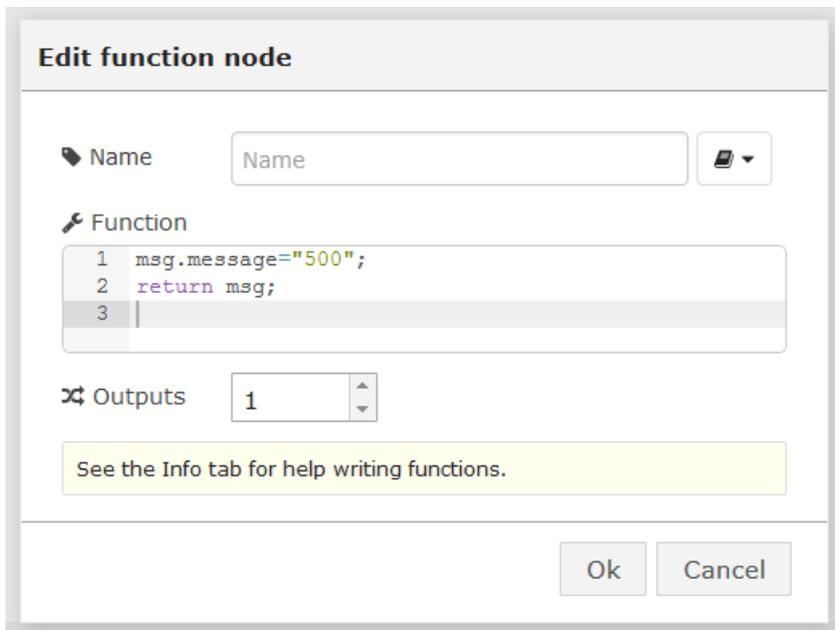Of course, some value may be unknown at this time, thus should be calculated at runtime.

The solution is to use a function node and calculate the value in javascript (an easy software language).

A basic example

Here is the flow with a new function node that sets the message to "500":

Double click on the function node to add the javascript:

- **line 1**: the `message` property of the flow (the msg object) is set to "500". We do that because we know (its written in the info tab) that the LEDText node uses that property to find the text to display. You can do any calculation using javascript to calculate the accurate message.

- **line 2**: we return the modified msg object to let it follow the rest of the flow.

Get a value from the cloud

Make it more useful and go fetch the bitcoin value on the web.

We get the value by making an HTTP request to **https://paymium.com/api/v1/data/eur/ticker** which returns data

like:

```
1 {
2   high: 374.99,
3   low: 365.63,
4   volume: 43.36850015,
5   bid: 367.01,
6   ask: 369.07,
7   midpoint: 368.04,
8   vwap: 368.91341449,
9   at: 1459516121,
10  price: 367,
11  open: 365.62,
12  variation: 0.3774,
13  currency: "EUR",
14  trade_id: "1f7d36f2-b66e-46a4-8c33-8568f0d7f8e0",
15  size: 0.682
16 }
```

To do that, we use an "HTTP Request" node and fill it with the address above:

Edit http request node

☰ Method	GET
🌐 URL	https://paymium.com/api/v1/data/eur/ticker
	☐ Use basic authentication?
← Return	a parsed JSON object
🏷 Name	Name

Tip: If the JSON parse fails the fetched string is returned as-is.

Ok   Cancel

Then we link a function node to feed the msg.message as above:

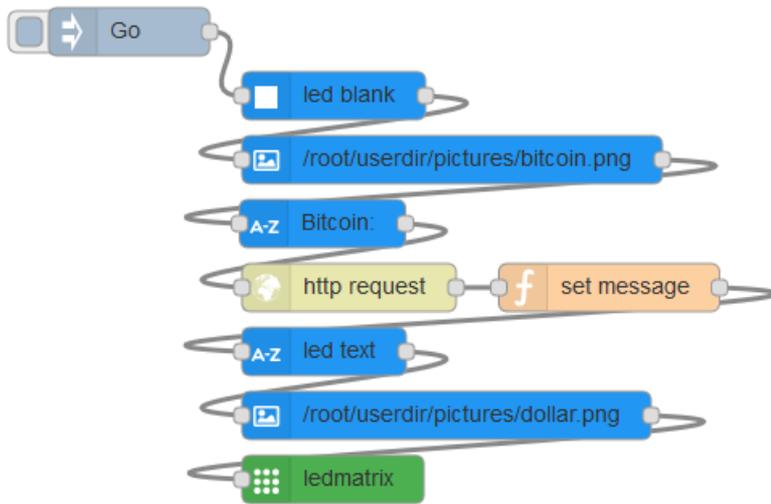**Edit function node**

🏷 Name   set message

🔧 Function

```
1  msg.message=msg.payload.ask;
2  return msg;
```

⤨ Outputs   1

See the Info tab for help writing functions.

Ok   Cancel

Here is the final flow:

○ Change display settings

By editing the LEDMatrix node, you can change the *speed* and the *intensity* of the display.