Matthew Townsend

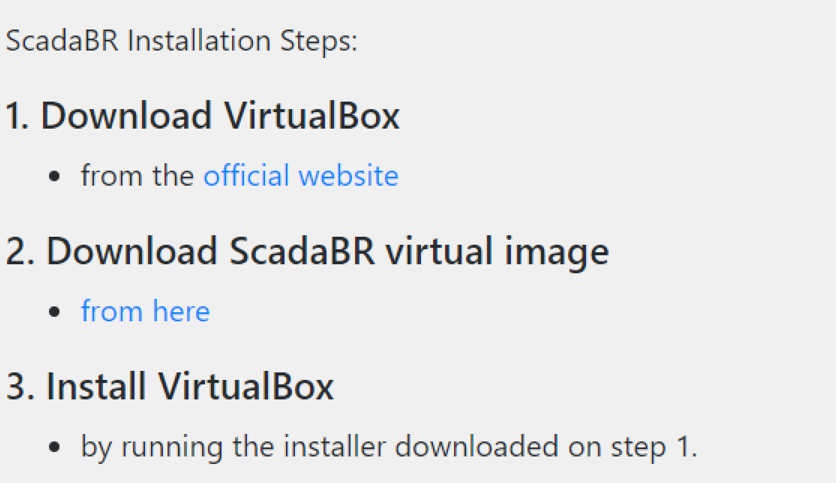
3/28/2023

Lab 5 HMI

In this Lab we used SCADABR along with the live circuit and create an HMI like the monitoring page on localhost:8080. SCADABR is like a web server but works in that it communicates with any IP given in the source page asking for what signals it is sending out on a GPIO through a plc runtime editor. For our lab it will communicate with the openplc runtime editor to collect this data. At its base, SCADABR is a server that runs on a computer that hosts a website for us to monitor the circuit on our local network. For this we all connected to a sing router so that the two systems will work together. To begin we went to

https://openplcproject.gitlab.io/reference/scada/installing-scadabr.html

and installed the image



ScadaBR Installation Steps:

1. Download VirtualBox

- from the official website

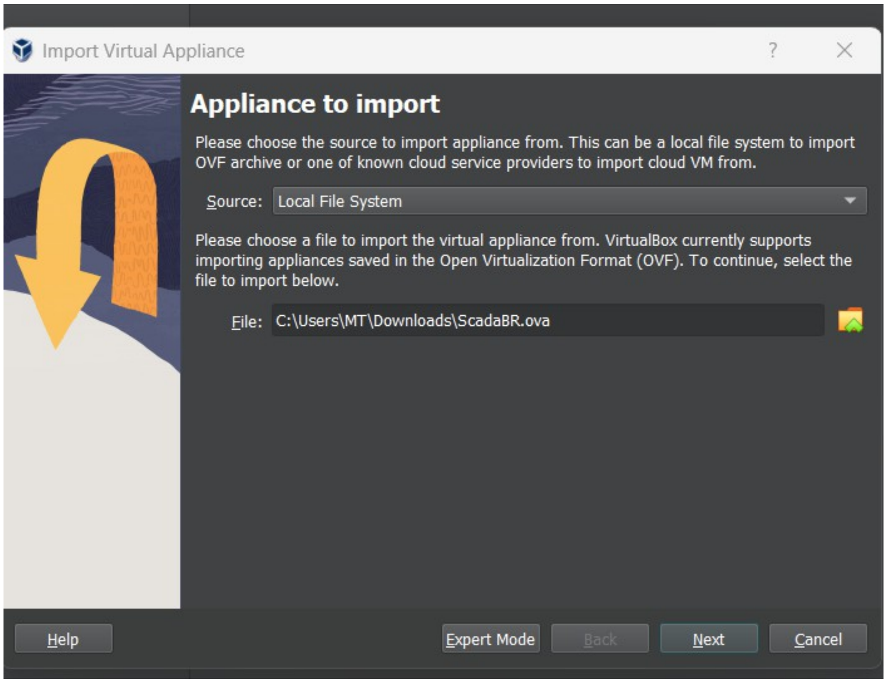
2. Download ScadaBR virtual image

- from here

3. Install VirtualBox

- by running the installer downloaded on step 1.

Then with this image we import it as an appliance as seen here then install it.



Import Virtual Appliance

Appliance to import

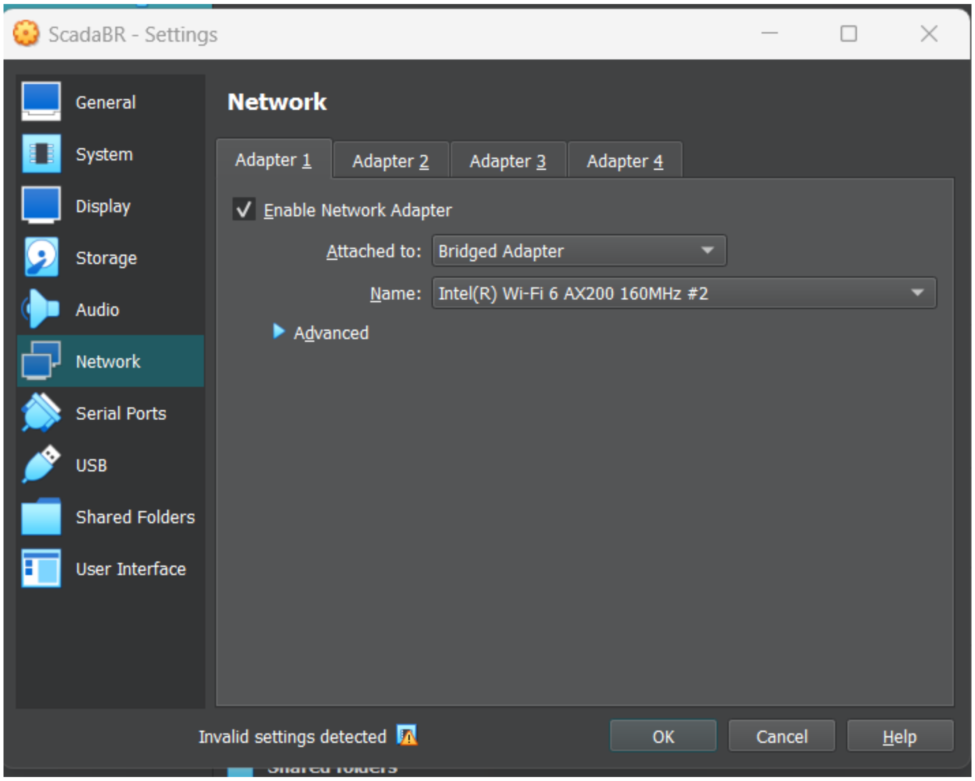
Please choose the source to import appliance from. This can be a local file system to import OVF archive or one of known cloud service providers to import cloud VM from.

Source: Local File System

Please choose a file to import the virtual appliance from. VirtualBox currently supports importing appliances saved in the Open Virtualization Format (OVF). To continue, select the file to import below.

File: C:\Users\MT\Downloads\ScadaBR.ova

After clicking through the screens and installing this we will need to change the network settings



Network

☑ Enable Network Adapter

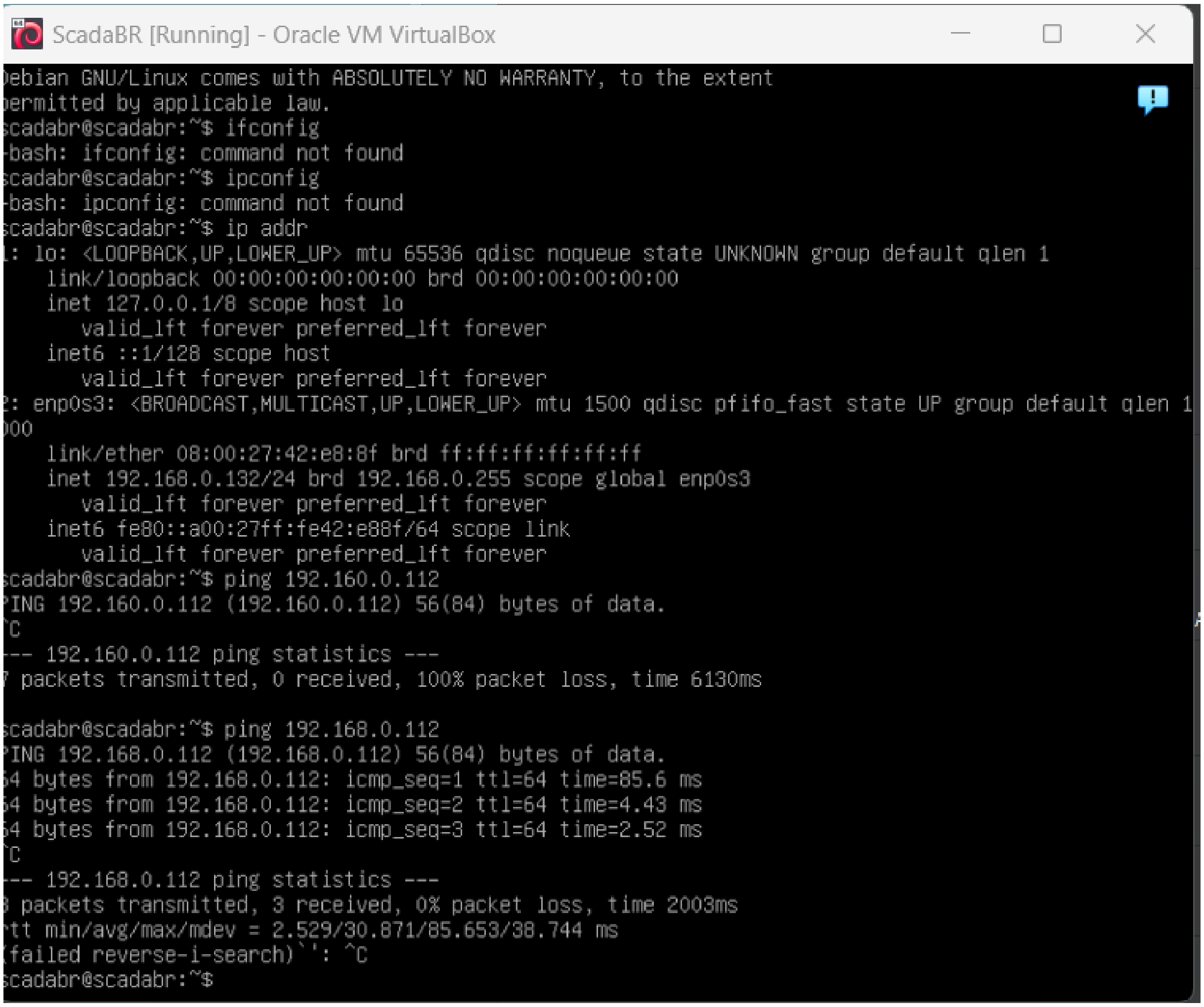
Attached to: Bridged Adapter

Name: Intel(R) Wi-Fi 6 AX200 160MHz #2

Invalid settings detected

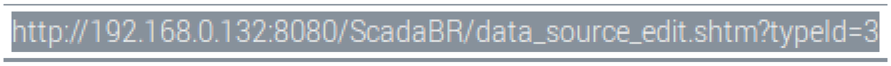
The setting s are the network to bridged, this is so that the machine will share the same IP as the computer so that it will be accessible on the ci246 network.

For those on the lab computer they had to store the virtual hard disk on a usb due to them being wiped every night. I used my laptop so I did not have to do his but its very simple, find the VM folders, find SCADABR then its disk and copy it out.



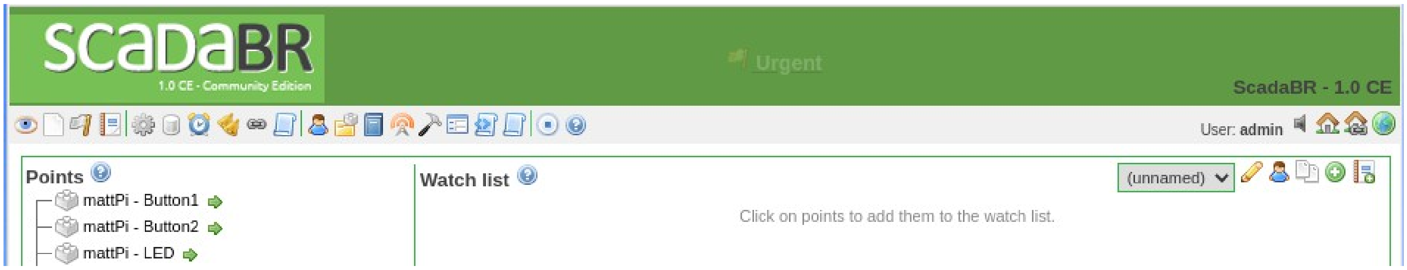
```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
scadabr@scadabr:~$ ifconfig
-bash: ifconfig: command not found
scadabr@scadabr:~$ ipconfig
-bash: ipconfig: command not found
scadabr@scadabr:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 08:00:27:42:e8:8f brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.132/24 brd 192.168.0.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe42:e88f/64 scope link
       valid_lft forever preferred_lft forever
scadabr@scadabr:~$ ping 192.160.0.112
PING 192.160.0.112 (192.160.0.112) 56(84) bytes of data.
^C
--- 192.160.0.112 ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 6130ms

scadabr@scadabr:~$ ping 192.168.0.112
PING 192.168.0.112 (192.168.0.112) 56(84) bytes of data.
64 bytes from 192.168.0.112: icmp_seq=1 ttl=64 time=85.6 ms
64 bytes from 192.168.0.112: icmp_seq=2 ttl=64 time=4.43 ms
64 bytes from 192.168.0.112: icmp_seq=3 ttl=64 time=2.52 ms
^C
--- 192.168.0.112 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 2.529/30.871/85.653/38.744 ms
(failed reverse-i-search)`': ^C
scadabr@scadabr:~$
```

It will look like this and in the `ip a` command you can see the IP of 192.168.0.132, the same as my laptop. SCADABR hosts a site which is accessible the following address, IP:8080/ScadaBR.

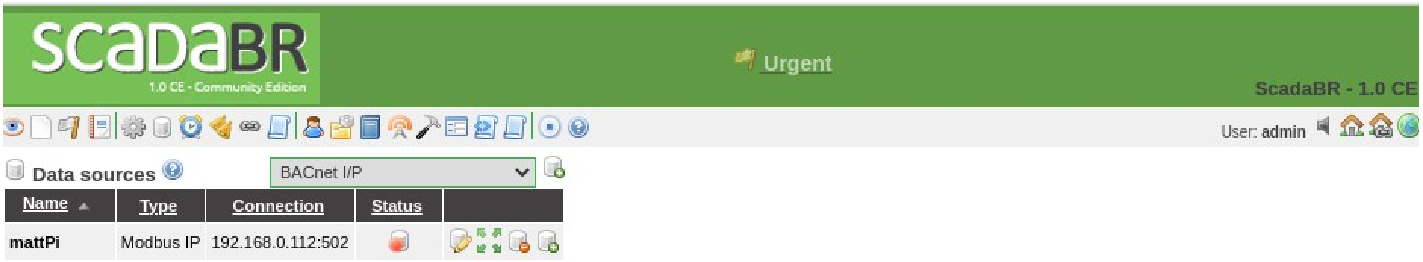


http://192.168.0.132:8080/ScadaBR/data_source_edit.shtm?typeId=3

You will be presented with a screen like this just without points.



Notice the bar of icons just under the logo, this is where all the work comes, press the white cylinder to reach the data sources.



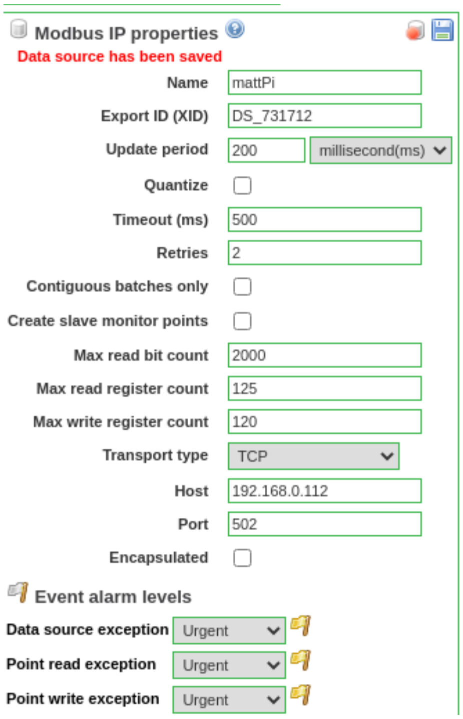
Data sources

BACnet I/P

Name	Type	Connection	Status
mattPi	Modbus IP	192.168.0.112:502	

This is where the plc information is stored, to create a source such as mattPi, it type should be changed from BACnet I/P to modbus IP. This is due to the communication protocol PLCs use.

Press the cylinder with a green arrow next to it. This will take to the creation of a source.



Modbus IP properties

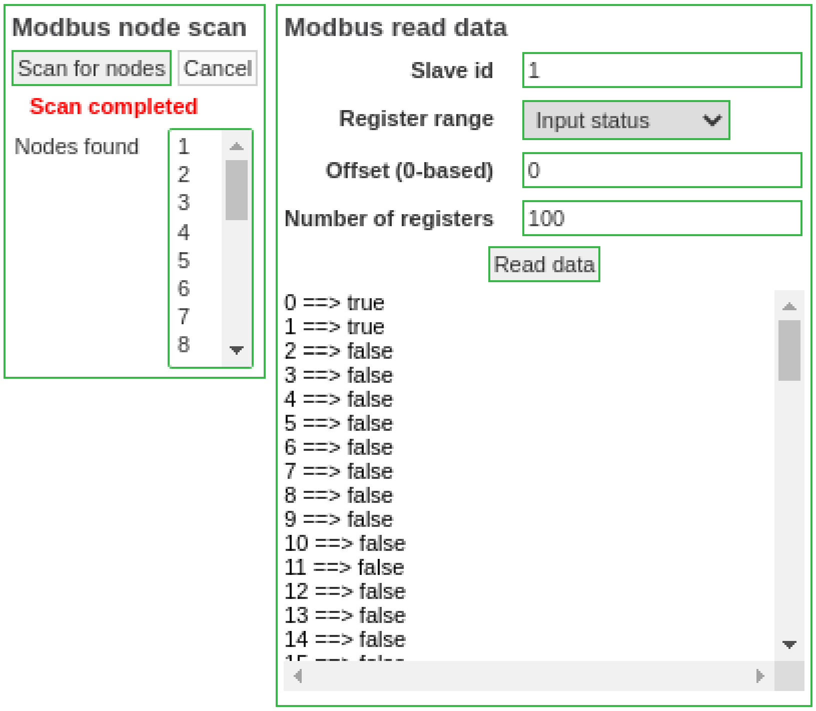
Data source has been saved

Name	mattPi
Export ID (XID)	DS_731712
Update period	200 millisecond(ms)
Quantize	□
Timeout (ms)	500
Retries	2
Contiguous batches only	□
Create slave monitor points	□
Max read bit count	2000
Max read register count	125
Max write register count	120
Transport type	TCP
Host	192.168.0.112
Port	502
Encapsulated	□

Event alarm levels

Data source exception	Urgent
Point read exception	Urgent
Point write exception	Urgent

Here you will name it, change update period from 5 minutes to a either seconds or milliseconds, then add the IP address of your pi. Then press the blue save icon, then the source called mattPi will be saved but it will monitor nothing as no points were added. To locate then add the points you will use the modbus scan and read data to discover the points of the data



Modbus node scan

Scan completed

Nodes found

```
1
2
3
4
5
6
7
8
```

Modbus read data

Slave id: 1

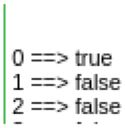
Register range: Input status

Offset (0-based): 0

Number of registers: 100

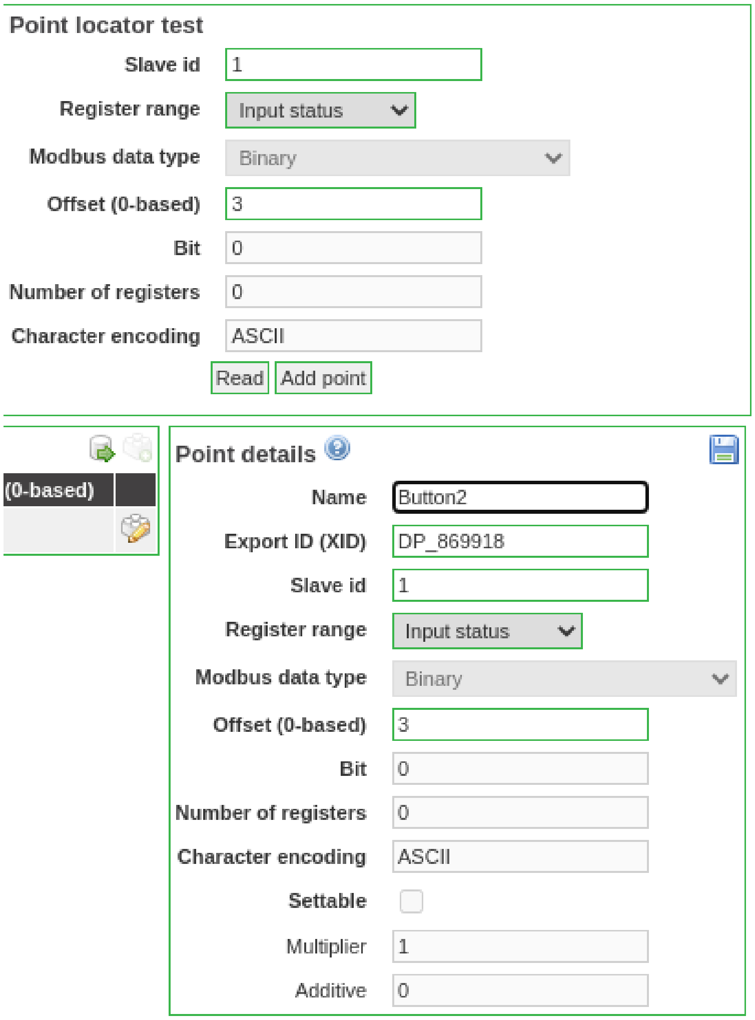
```
0 ==> true
1 ==> true
2 ==> false
3 ==> false
4 ==> false
5 ==> false
6 ==> false
7 ==> false
8 ==> false
9 ==> false
10 ==> false
11 ==> false
12 ==> false
13 ==> false
14 ==> false
```

It is as simple as pressing Scan for Nodes then read data, if there is an error it means theres a problem with your IP or the plc. Here you can see that it reads two inputs as true, this is the two buttons being detected. Change the Input Status to coil status it will look like this



```
0 ==> true
1 ==> false
2 ==> false
```

With this data you can create the points which will be used to monitor the data. I requires a name and the offset of 2 and 3 for the inputs, because remember from the live circuit labs we can not use the first two input pins. For button two it will look like this.



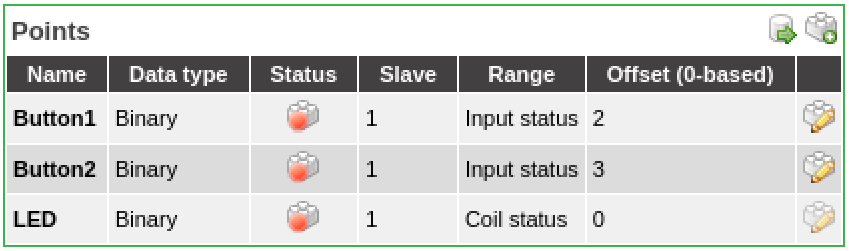
Point locator test

Slave id	1
Register range	Input status
Modbus data type	Binary
Offset (0-based)	3
Bit	0
Number of registers	0
Character encoding	ASCII

Point details

Name	Button2
Export ID (XID)	DP_869918
Slave id	1
Register range	Input status
Modbus data type	Binary
Offset (0-based)	3
Bit	0
Number of registers	0
Character encoding	ASCII
Settable	□
Multiplier	1
Additive	0

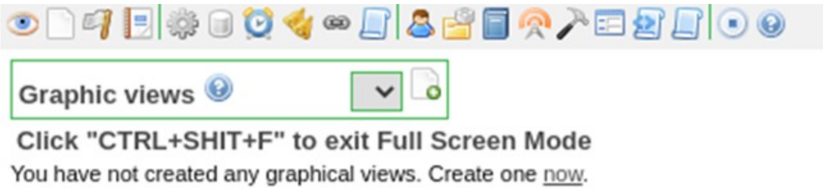
This will create the point to which we can create an HMI and monitor it whether its on or off. Follow this process to create the points on our circuit, then we will move onto the graphical view. It should look like this at the bottom of the page. Make sure you press the status, so they turn green for later monitoring, do the same on the data source page.



Points

Name	Data type	Status	Slave	Range	Offset (0-based)	
Button1	Binary	●	1	Input status	2	
Button2	Binary	●	1	Input status	3	
LED	Binary	●	1	Coil status	0	

The Graphical view is the HMI for our plc, it is the white paper icon on the task bar,

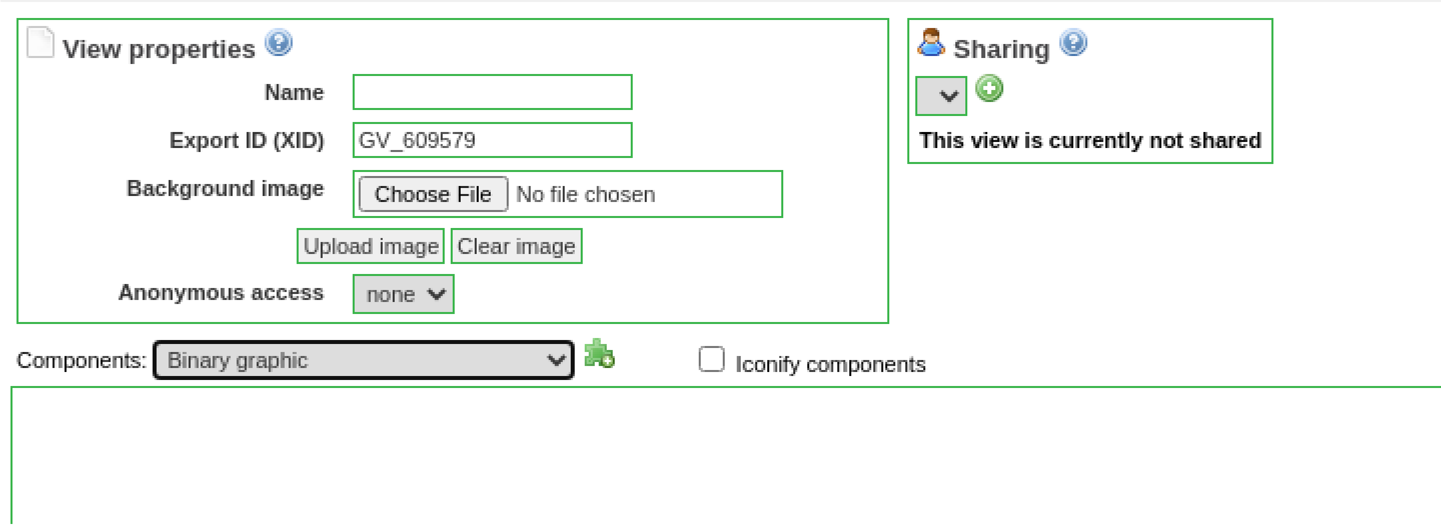


Graphic views

Click "CTRL+SHIT+F" to exit Full Screen Mode

You have not created any graphical views. Create one now.

Here you will click the page with a green pluss in the corner to create the new view which will look like this.



View properties

Name:

Export ID (XID): GV_609579

Background image: Choose File No file chosen

Upload image | Clear image

Anonymous access: none

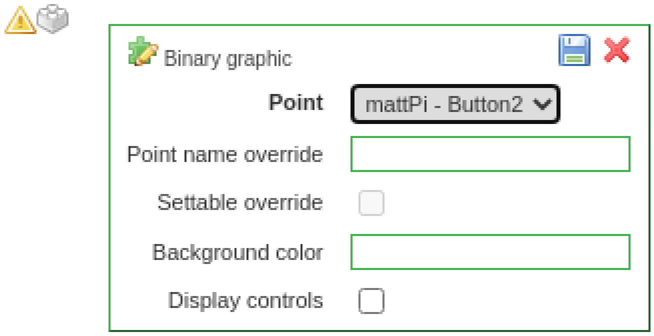
Sharing

This view is currently not shared

Components: Binary graphic

□ Iconify components

Here you will name the view, then add your components which will be the points from the data sources. You do this by pressing the green puzzle piece for each pointer. It will add one blank object which you will name and give an image.



Binary graphic

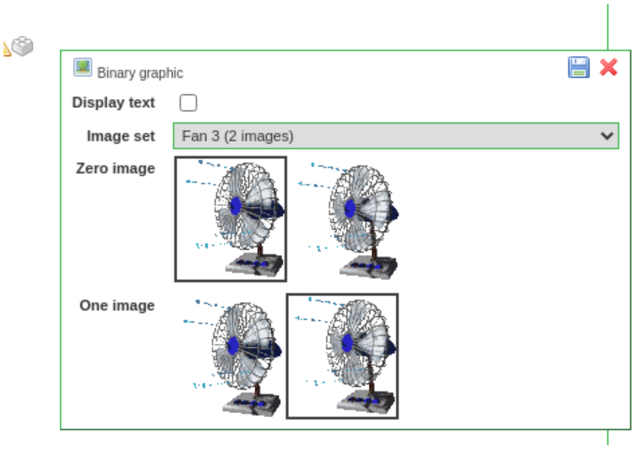
Point mattPi - Button2

Point name override

Settable override □

Background color

Display controls □



Binary graphic

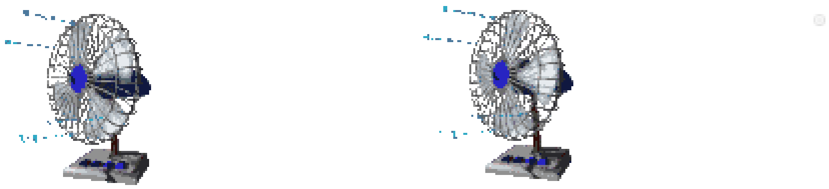
Display text □

Image set Fan 3 (2 images)

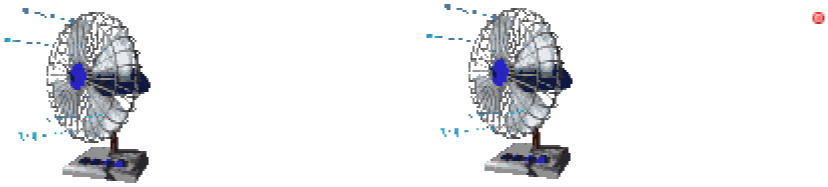
Zero image

One image

After setting up after each pointer for this we will be able to view all three working in a view.



All buttons + LED off



Post button 1, LED on.