

Setting Up SmartSign To Work With The Broadcast Radio Hardware Service

The Broadcast Radio Hardware Service is a software application used to relay hardware events to our fifth generation software products, using your existing IT network. In the context of SmartSign Lite, this is most commonly used to allow a physical hardware input on one SmartSign to trigger events on multiple other SmartSigns on your network (for example a second SmartSign outside the studio that warns when the microphones are 'live').

The Broadcast Radio Hardware Service is compatible with the following software.

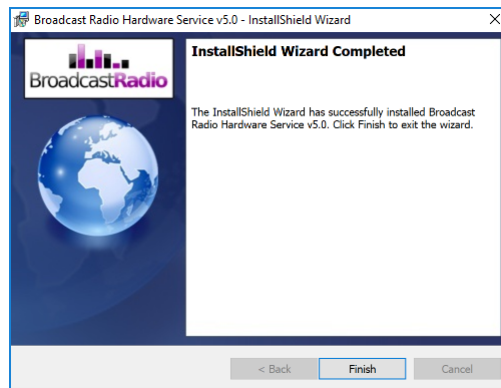
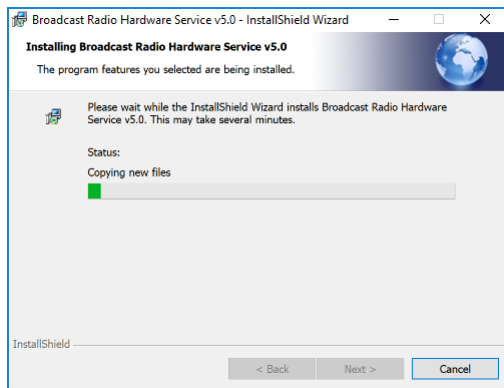
- Myriad 5 Payout
- Myriad 5 Logging
- SmartSign Lite

In order to use this facility you need to download and install the Broadcast Radio Hardware Service which you can obtain for free from our website. You will also need to install it on a PC that is on the same network as the SmartSigns that need to use it.

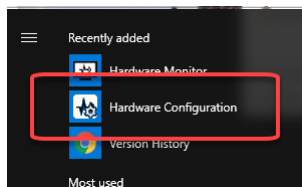
Installing & Configuring The Hardware Service

Visit: <http://www.broadcastradio.com/support/myriad-5-logging-support/myriad-5-logging-downloads/>

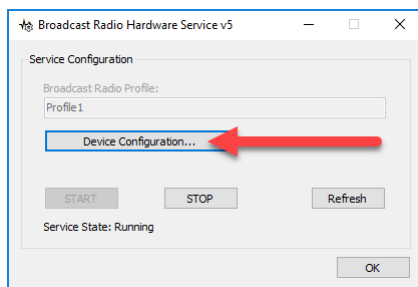
Expand the 'Related Downloads' option and then click on the **Download** button. The software will then download to your usual downloads folder. Once the download is complete, double click on the file to begin the installation.



Next, press the Windows Start button on your keyboard and locate the **Hardware Configuration** option which should be listed under **Broadcast Radio Tools** (and in the Recently Added list).



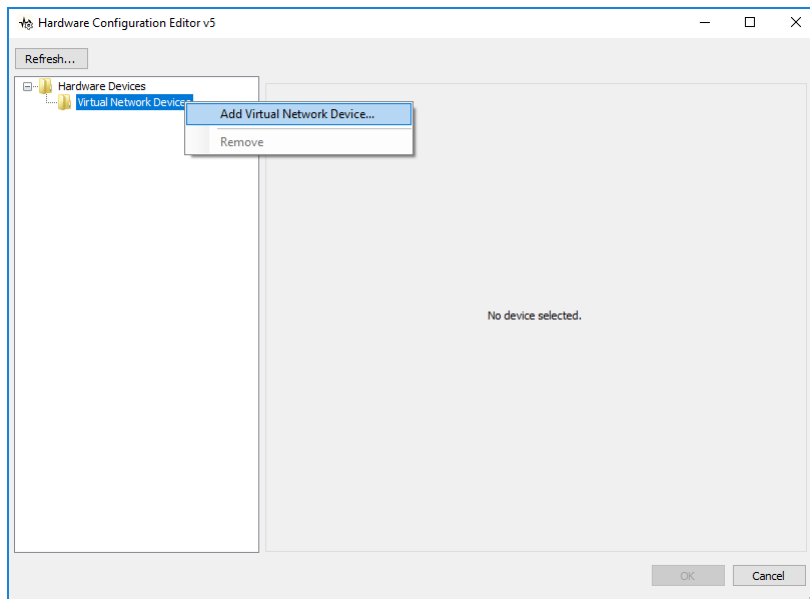
As the software starts, you may be asked to allow the software to configure a hardware device by Windows (depending on your user account access), if you are, just click 'yes'.



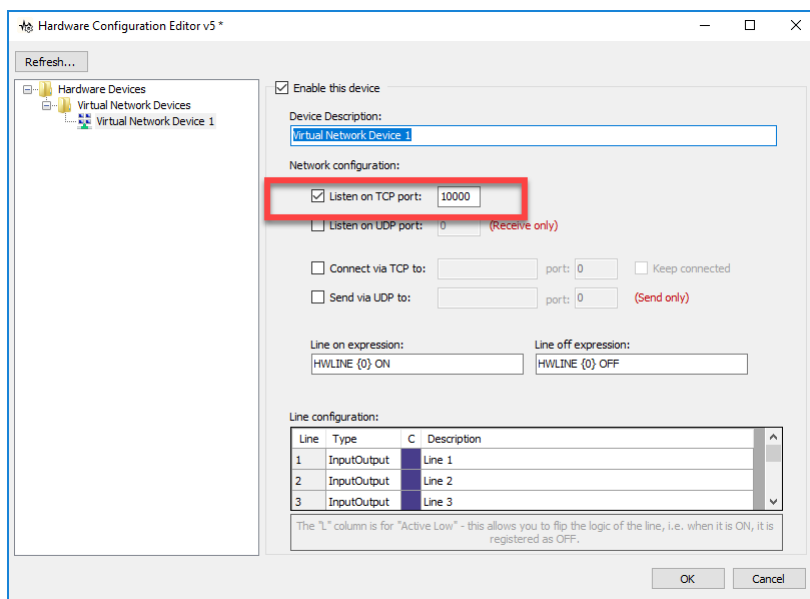
This window shows you the status of the Hardware Service and allows you to configure the devices. In this case you want to click on the **Device Configuration** button.

Next you will see an empty screen with 'No Device Selected' in the centre because we have not setup a device yet. Obviously if you have previously setup a device (physical or virtual) then it will be displayed here.

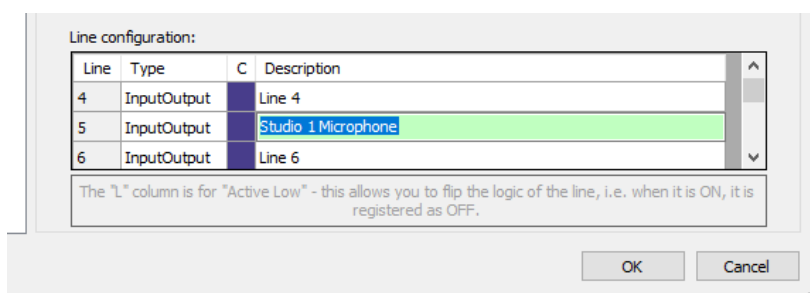
Right Click on the **Virtual Network Devices** folder and select **Add Virtual Network Device**



This will create a new Virtual Network Device with the default settings. The settings are covered in separate documentation but in most cases you can leave the defaults in place. You will need to make note of the **Listen On TCP Port** number which is set to **10000** but default. You will need that number when we come to configure the SmartSigns.



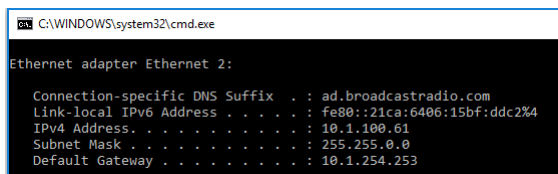
NOTE: You can change the hardware line descriptions if you like to make testing easier. In this example we are going to use line 5 for the 'mic live' indicator in Studio 1 so I have altered the description for line 5.



Click on **OK** to close the window, you will be asked to confirm you wish to overwrite the previous settings. Once you have done that, you can click on **OK** on the Service Configuration window to close it. At this stage it will disappear but don't worry, the Hardware Service runs as a **Service** which means that it is running in the background all the time but you will not be able to see it working unless you run the **Hardware Monitor** application.

Now the Hardware Service is running and configured, you need to find out one more thing before you can configure your SmartSigns, you need to find out the IP address of the computer you have just installed onto. The quickest way is:

1. Windows Key + R (this opens the Run box)
2. Type in CMD and press the <Enter> key on your keyboard (this opens a command line window).
3. Type in **ipconfig** and press <Enter> (this will list your IP address.



```
C:\WINDOWS\system32\cmd.exe

Ethernet adapter Ethernet 2:

    Connection-specific DNS Suffix  . : ad.broadcastradio.com
    Link-local IPv6 Address . . . . . : fe80::21ca:6406:15bf:ddc2%4
    IPv4 Address. . . . . : 10.1.100.61
    Subnet Mask . . . . . : 255.255.0.0
    Default Gateway . . . . . : 10.1.254.253
```

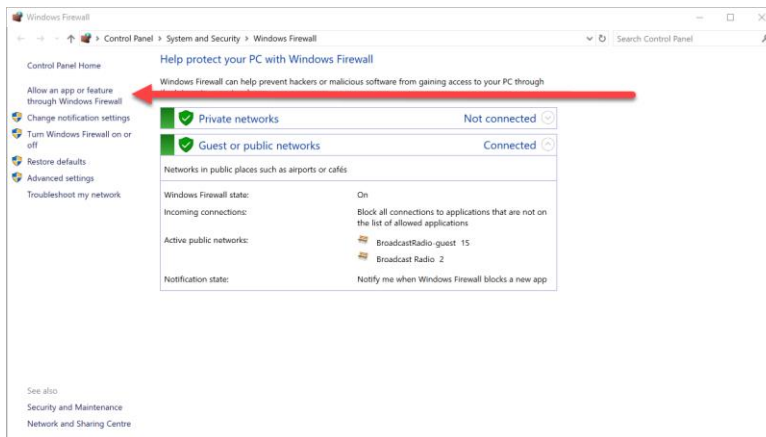
4. Write down the IP Address on the **IPv4** line. This will be in the format xxx.xxx.xxx.xxx (in the example above it is 10.1.100.61).
5. Type **Exit** and press <Enter> to close the window.

You now have everything ready to setup SmartSign Lite to work with the Hardware Service.

Important: You Need To Update Your Windows Firewall

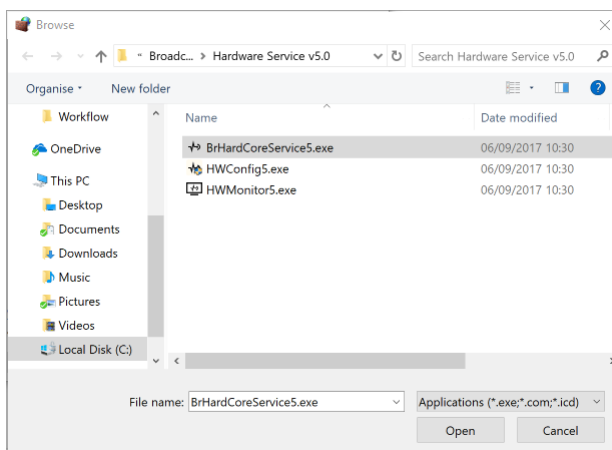
IMPORTANT: Whilst not technically part of the Hardware Service installation process, by default, you **Windows Firewall** will block your Hardware Service from communicating with the SmartSigns. You need to allow add in an 'exception' to allow the service to safely send and receive data from other connecting systems.

1. Press the Windows key on your keyboard and type in Firewall. You should see an option called **Windows Firewall**, run that option.
2. Click on the '**Allow An App Or Feature Through Windows Firewall**' option on the left hand list.

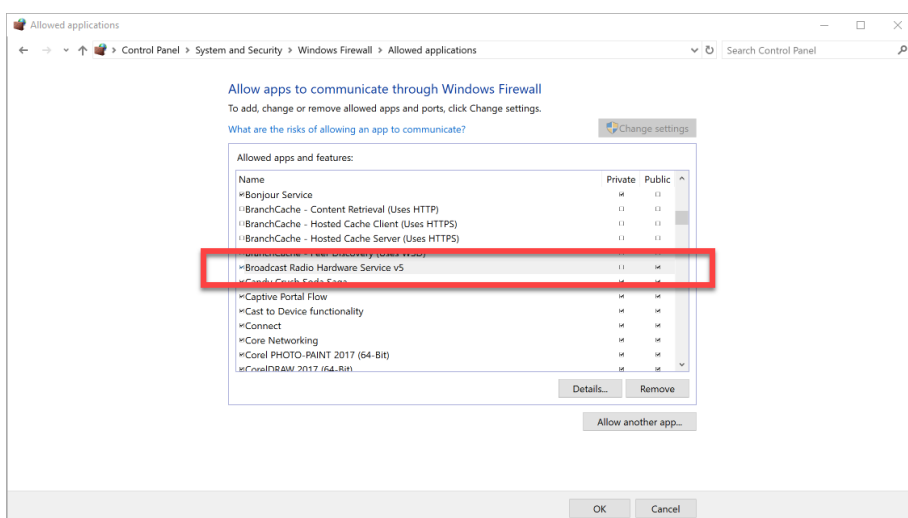


3. Click on the **Change Settings** button (top right) and then the **Allow Another App** button (lower right).
4. You now need to browse to the location where the Hardware Service is installed and select **BrHardCoreService5.exe**. The default location is:

C:\Program Files (x86)\Broadcast Radio\Hardware Service v5.0\BrHardCoreService5.exe



5. Now click on the **Add** button to add the app to the Firewall exceptions list. You should see it listed in the list of allowed applications.

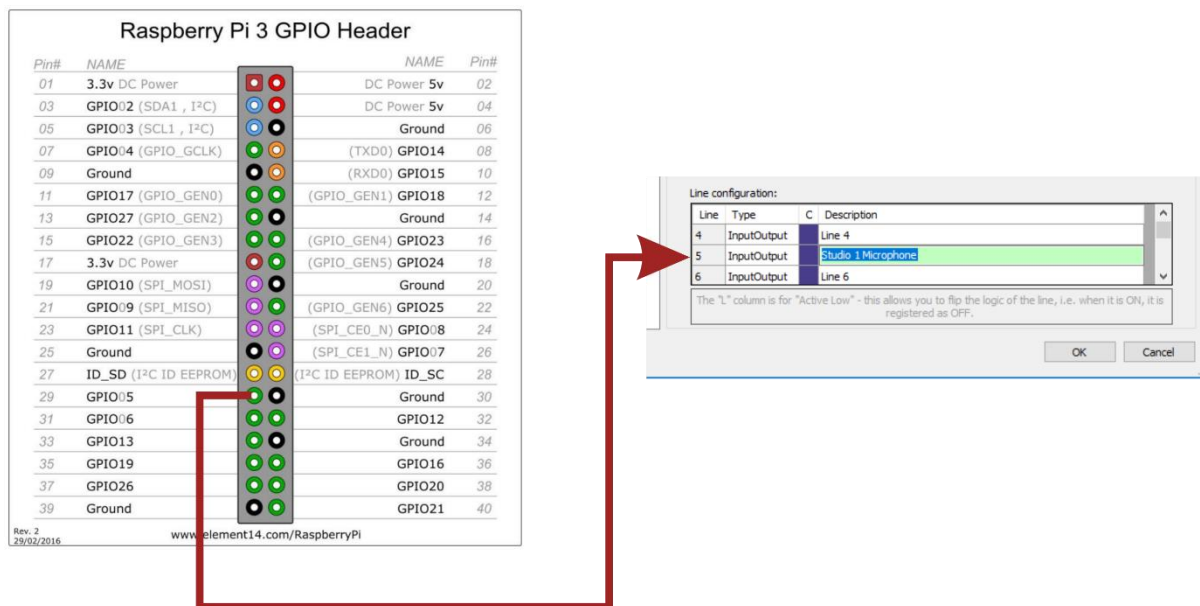


Your Hardware Service should now be able to communicate with the other Broadcast Radio software and systems on your network.

Setting Up The First SmartSign

The Theory

The way this works is that the Hardware Service creates 100 virtual hardware lines that can be used for inputs or outputs. The first 27 of these lines are also directly linked to the corresponding GPIO inputs / outputs on the Raspberry Pi such that Virtual Line 5 in the Hardware Service is intrinsically linked to GPIO 5 on the Raspberry Pi (keep in mind this is not the same thing as pin 5 as the inputs are not mapped directly on Raspberry Pi's, in fact GPIO Line 5 on a Pi is actually physically pin 29 on the connector!)



The diagram above shows how GPIO Line 5 on the Pi (physical pin 29) is directly linked to Virtual Line 5 in the hardware service, so if you trigger an input on GPIO Line 5 on the Pi, Virtual Line 5 will automatically reflect that in the Hardware Service.

In practice, GPIO Lines 1-4 on the Pi are reserved for board use so GPIO 5 should be considered the first usable GPIO.

In order to understand how this works in the real world, we use the example of a 'Mic Live' indicator in a studio that needs to be duplicated to a second SmartSign located outside the studio to prevent people opening the door when the microphone is on.

Within the studio it is a very simple setup. The 'Mic Live' logic generated by the mixing desk whenever the microphone channel is 'on' is connected to GPIO Line 5 on the Raspberry Pi (usually this is actually a closure between Line 5 (pin 29) and a ground pin (there is conveniently a ground pin on pin 30 directly beside pin 29). So when the microphone channel is 'opened' on the mixer, the mixing desk logic output creates a 'closure' between GPIO Line 5 (pin 29) and ground (pin 30).

You then configure your SmartSign to have a Smart Display Tile that changes 'state' whenever GPIO Line 5 is triggered.



So when you turn on the microphone, the Smart Display Tile on the SmartSign will change accordingly.

So far this is nothing new but we now want to have the same thing happen on a second SmartSign outside the studio but we want to use a virtual hardware line to trigger the Smart Display Tile. Here are the steps we need to take:

1. Setup the first SmartSign to 'broadcast' its GPIO Hardware Events to the Hardware Service.
2. On the second SmartSign, setup the corresponding GPIO line as an **Output** (the reason why is explained later).
3. Setup a Smart Display Tile on the second SmartSign to react to the corresponding GPIO Line as if it were a normal physical input.

So lets get started.

Configuring The SmartSign In The Studio To Broadcast Its Hardware Status

As already discussed, Hardware Service has 100 virtual lines and lines 5 to 27 are directly mapped to the corresponding physical GPIO Lines on any Pi that is connected to the service.

So when a physical input connected on GPIO Line 5 is triggered on any SmartSign connected to the Hardware Service, the corresponding Virtual Line 5 will also trigger and all other SmartSigns that are 'subscribing' to the service will also see that trigger.

In practice that means you are safest assigning a GPIO / Virtual Line for each single trigger or event you want to use throughout your system eg:

- Line 5 = Mic in studio1
- Line 6 = Mic in studio 2
- Line 8 = Studio 1 has on air control
- Line 9 = Studio 2 has on air control etc

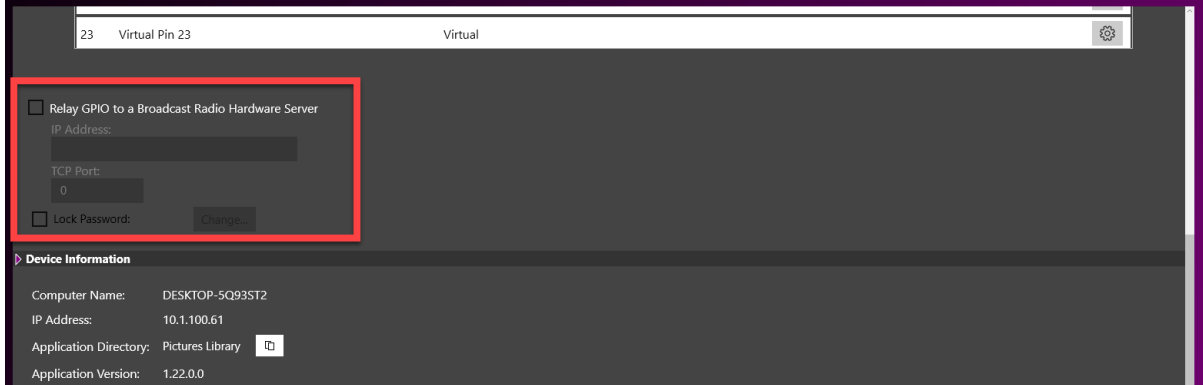
That means only the SmartSign in Studio 1 is using physically GPIO Line 5 on the it's Pi with the SmartSign in studio 2 using a different line altogether.

In this example, we are assuming that we already have the SmartSign in studio 1 setup to receive a physical GPIO input on Line 5 and have a Smart Display Tile setup to change when it happens (mic live). Now we need to 'broadcast' this event via the Hardware Service.

On your SmartSign, go into **Layout** mode and click on the **App Settings** option located on the extreme left.

Scroll down until you see the **Relay GPIO To A Broadcast Radio Hardware Service** option.

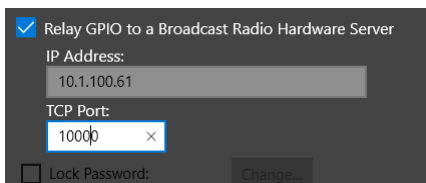
Application Settings



Tick the box to enable and add the following information (which you noted when we setup the Hardware Service).

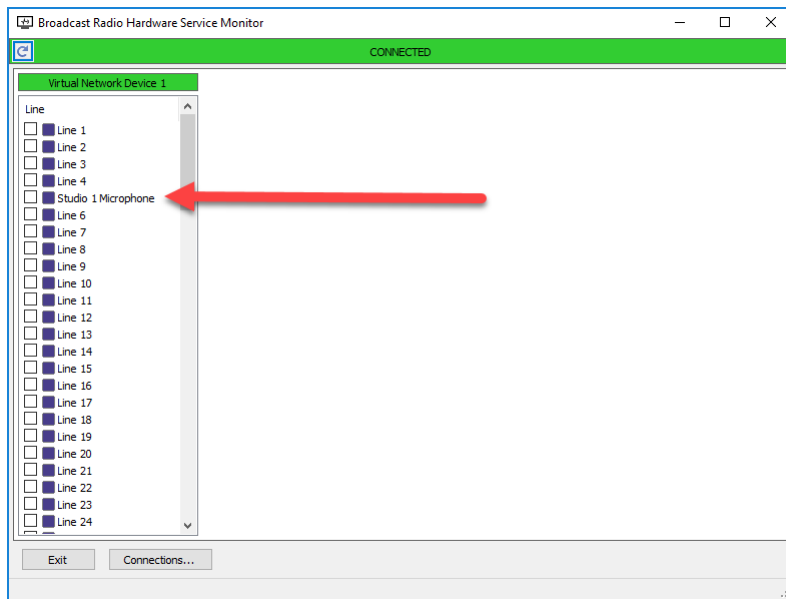
- IP Address Of PC Running The Hardware Service
- TCP Port Used For The Virtual Device

In the example we looked at when setting up the Hardware Service they were 10.1.100.61 and 10000 respectively. Your IP address will be different but your default port should be 10000 unless you altered it when setting up the Virtual Network Device.



Click on the **Back** button at the bottom left of the **Applications Settings** screen to save your changes and return to the **Layout Page**. You can now click on **Go** to restart your SmartSign.

Tip: You can now run the Hardware Monitoring application on the PC that the service is running and you should be able to 'see' when the virtual hardware is triggered. Try running the Monitoring Application and then switching the microphone on and off in the studio, you should see the corresponding Virtual Line switching on and off.



Configuring Other SmartSigns To Subscribe To Your Virtual Hardware

So now that we are ‘broadcasting’ the trigger via the Hardware Service, we need to setup the other SmartSign to react to the **Virtual Hardware Line**.

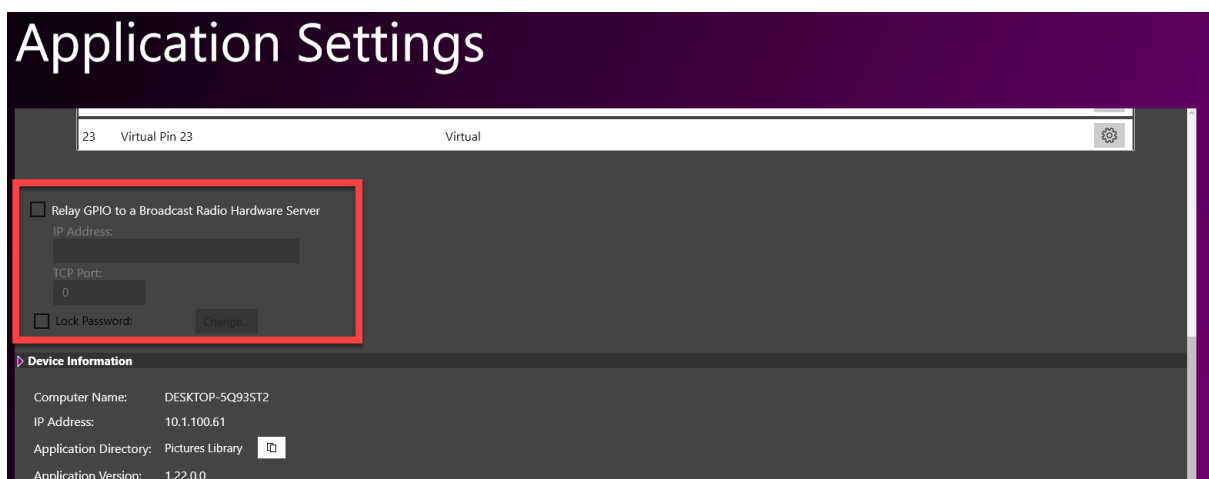
Assuming it is plugged into the same network as the Hardware Service, it is simply a case of setting the **IP Address & Port** for the **Hardware Service** (just like we did on the other SmartSign) and then set the GPIO Line we are interested in (5 in this case) to be an **Output**.

Why an Output I hear you say? Good question! There is a complex technical reason which involves instructing the internal chipset on the PI to ignore physical inputs and instead rely on virtualised ones provided in software but it is easier to just accept, for this to work you need to set the GPIO Line we are interested in to be an **Output**.

So first thing is to add in the IP Address and Port for the Hardware Service using the same method as on the first SmartSign:

Go into **Layout** mode and click on the **App Settings** option located on the extreme left.

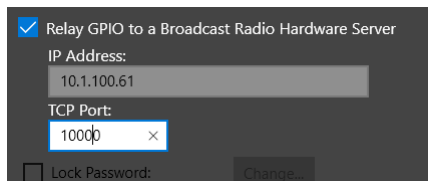
Scroll down until you see the **Relay GPIO To A Broadcast Radio Hardware Service** option.



Tick the box to enable and add the following information (which you noted when we setup the Hardware Service).

- IP Address Of PC Running The Hardware Service
- TCP Port Used For The Virtual Device

In the example we looked at when setting up the Hardware Service they were 10.1.100.61 and 10000 respectively. Your IP address will be different but your default port should be 10000 unless you altered it when setting up the Virtual Network Device.



☒ Relay GPIO to a Broadcast Radio Hardware Server

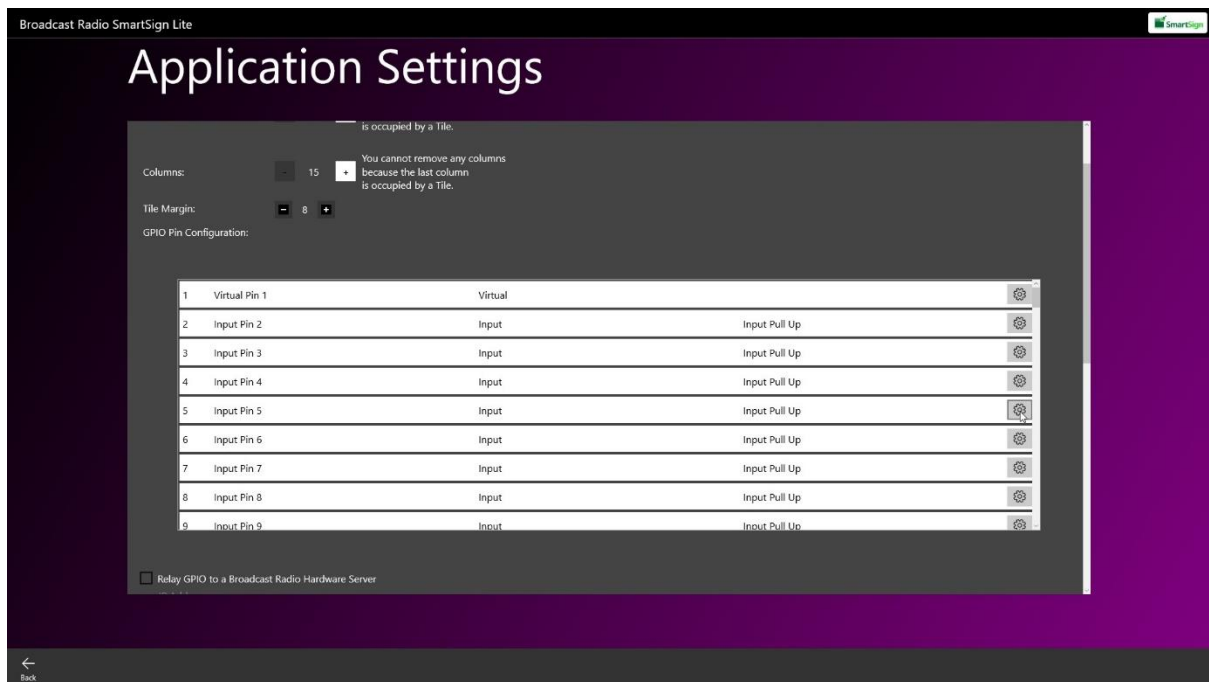
IP Address:
10.1.100.61

TCP Port:
10000

☐ Lock Password: Change...

But we also need to set GPIO Line 5 to be an **Output**.

Scroll up until you locate the **GPIO Pin Configuration** section.



Broadcast Radio SmartSign Lite

Application Settings

Columns: 15
Tile Margin: 8

GPIO Pin Configuration:

Pin	Label	Type	Configuration	Icon
1	Virtual Pin 1	Virtual		
2	Input Pin 2	Input	Input Pull Up	
3	Input Pin 3	Input	Input Pull Up	
4	Input Pin 4	Input	Input Pull Up	
5	Input Pin 5	Input	Input Pull Up	
6	Input Pin 6	Input	Input Pull Up	
7	Input Pin 7	Input	Input Pull Up	
8	Input Pin 8	Input	Input Pull Up	
9	Input Pin 9	Input	Input Pull Up	

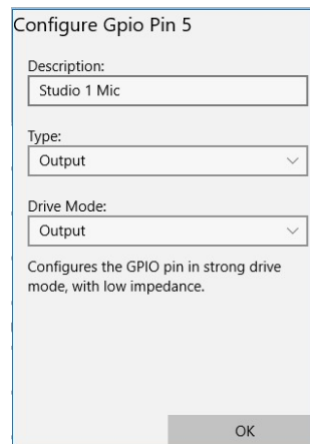
☐ Relay GPIO to a Broadcast Radio Hardware Server



Click on the icon to edit the setting for the GPIO Line you are interested in (5 in this example).

On the config window.

1. Set the Description to something useful.
2. Change the **Type** to **Output**
3. Change the **Drive Mode** to **Output**
4. Click on **OK**



Should now look something like this:

4	Input Pin 4	Input	Input Pull Up
5	Studio 1 Mic	Output	Output
6	Input Pin 6	Input	Input Pull Up

Your SmartSign is now configured to 'subscribe' to the Virtual Hardware Line 5 and you can setup a **Smart Display Tile** to react to GPIO Line 5 changing just like it was a physically connected input.

