



PodFlow – Getting Started

PodFlow is the graphical programming language developed by IKON AVS Ltd to facilitate the easy creation, modification and testing of programs for the Podule range controllers.

This guide is intended not only those who are new to PodFlow but also more experienced users who required a 'refresher' on some aspects of PodFlow.

Rather than a presenting a cold reference guide to all aspects of the software, this already exists, it takes the reader through a number of examples gradually building working systems utilising a broad range of Podules.

If you only need a refresher on certain aspects, just look it up in the index.

HARDWARE REQUIREMENTS	4
SOFTWARE INSTALLATION.....	4
GETTING STARTED WITH PODFLOW	5
TUTORIAL 1A – SIMPLE PROJECTOR CONTROL VIA RS232	6
Wiring	8
Projector OFF	9
Emulation.....	9
TUTORIAL 2B – EXPANDED PROJECTOR CONTROL VIA RS232	10
Flip-Flop Operation	12
Adding Projector Off	12
Separate On/ OFF Buttons	12
TUTORIAL 2C – ADDING TIMERS.....	13
Logical Gate.....	14
TUTORIAL 2D – FLASHING LEDs	17
Fast Flashing	17
Slow Flashing	18
Adding the Warm Up timer.....	18
Adding the Cooling Down Timer	19
TUTORIAL 3 – COMPLETING BASIC PROJECTOR CONTROL	20
TUTORIAL 3A – PROJECTOR INPUT SELECT	20
Enabling Buttons.....	20
Interlocking Buttons	20
TUTORIAL 3B – PICTURE BLANK.....	22
TUTORIAL 3C – VOLUME	22
Simple Control	22
Volume Ramp.....	23
TUTORIAL 4 – ADVANCED PROJECTOR CONTROL	24
Send until Acknowledged	24
Send then Interrogate	26
TUTORIAL 5 – PODNET	28
Message Processing	28
Message Numbering.....	28
TUTORIAL 5A – USING DIGITAL MESSAGES	29
RS232 INPUT.....	30
PODNET SIGNALS.....	31
PODNET INPUT	31
PODNET MESSAGE.....	31
PODNET OUTPUT	32
PODNET MESSAGE.....	32
LOGICAL GATE DEVICE	33
LOGICAL GATE EMULATOR	34
EMULATION.....	35
EMULATION TOOLBAR.....	35
INPUT EMULATION	36
EMULATION OUTPUTS	37
RUNNING THE EMULATION	37
APPENDIX 1	38
PODFLOW SCREEN.....	38

APPENDIX 2	39
<i>UNDERSTANDING RS232 MESSAGE FORMATS.</i>	39
APPENDIX 3	40
<i>TIMER</i>	40
APPENDIX 4	42
<i>PROGRAMMING THE POD</i>	42
MANUFACTURERS INFORMATION . ERROR! BOOKMARK NOT DEFINED.	

Hardware Requirements

There is no onerous requirement to allow the installation and running of PodFlow and most modern PC's and Laptops are suitable. The major requirement is the availability of either a Hardware serial (Com) port or a USB port with an USB-Serial adapter installed.

A minimum of 4MB is required for the Programs & libraries and a further 22MB for all the manuals.

PodFlow is suitable for use with Microsoft Windows only and is compatible with:- 98SE, 98ME, 2000, XP, XP-Pro, Media Centre. It will also run on Vista but you need to make certain that and USB-Serial converter used has Vista drivers available.

Software Installation

Before starting it is necessary to install the PodFlow software onto the PC or laptop.

On the CD and Website (www.ikonavs.com) there are two versions available.

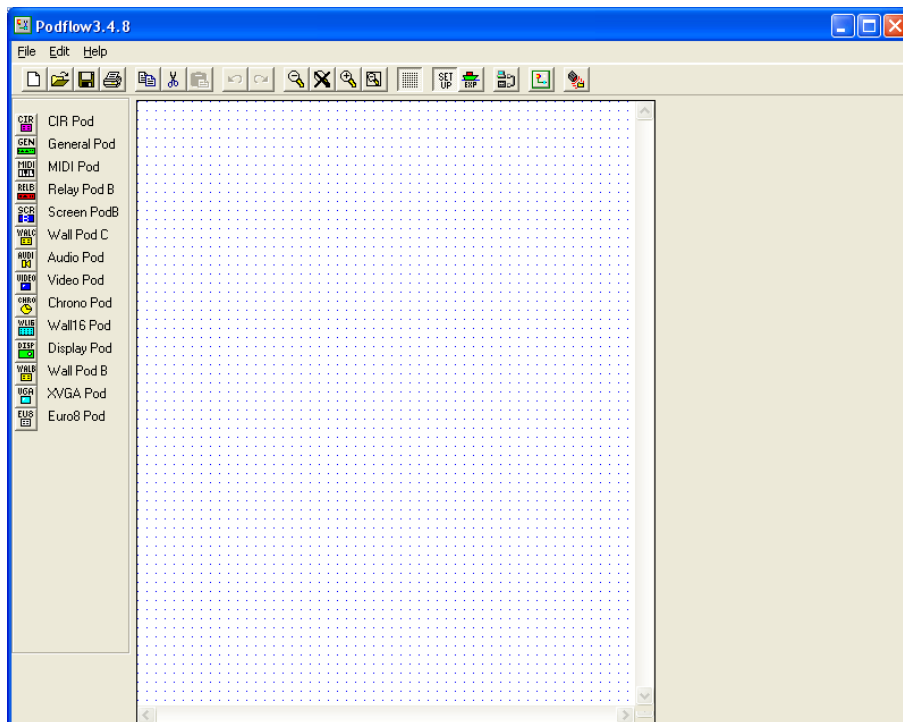
FULL Recommended for a first time install. By default it creates a PodFlow directory under 'Program Files' and installs PodFlow here. It also created two further subdirectories for the IR libraries and Manuals.

MINIMAL Only installs PodFlow and is intended for revision updates.

Download and run the required version following the instructions, when complete you are ready to continue.

Getting Started with PodFlow

Double click on the PodFlow icon to start and display the opening screen.

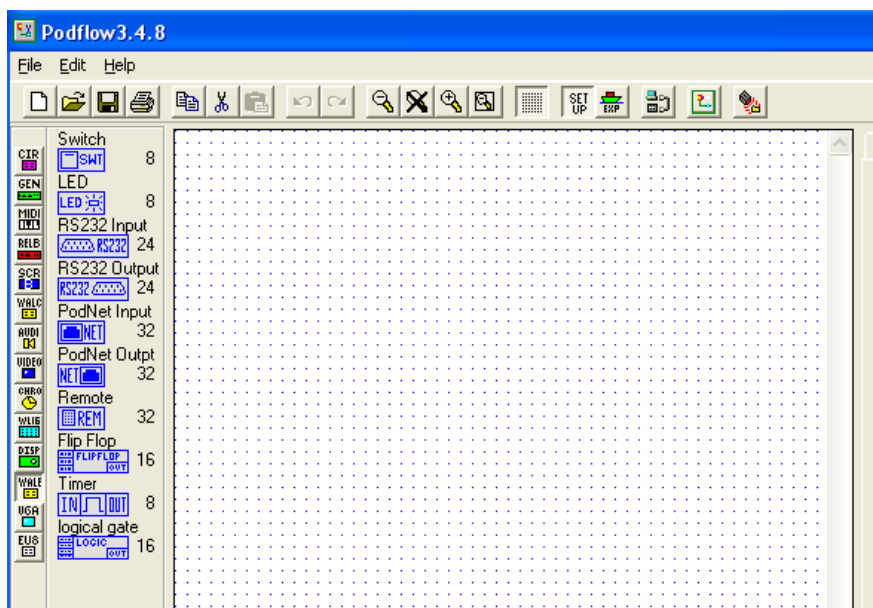


A full description of the screen layout is included as [Appendix 1](#). For now we will only concern ourselves with attributes as we use them.

To start the tutorial we are going to program a Wall Podule for the control of a video projector.


From the left icons select Wall Pod B.

The screen changes to the one below and shows to the left of the screen the available Input, Output and logic devices for the Wall Podule B.

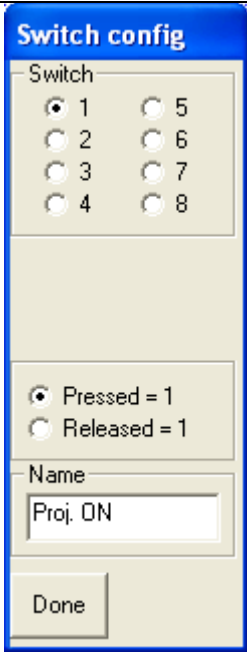


Tutorial 1a – Simple projector Control via RS232

In this tutorial we are going to look at a very basic method of controlling a projectors On / OFF functions via RS232. This is just the start to get you used to some of the basic techniques.


From the device list on the left, click on SWITCH  and drag onto the main working area (that's the space in the middle with a dotted grid). Drag about 1/3 of the way down and over to the left, release to drop here. Ok, from now on I will just refer to these actions as dragging a device onto the working area.

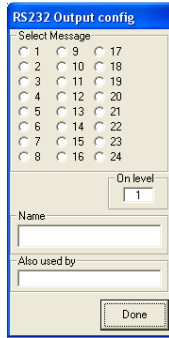
We now need to tell the system which switch this is – double click on the device to open.

	<p>Switch</p> <p>The switch location corresponds to the physical position of the switch on the Wall Podule. Switches are numbered in two columns with No. 1 at top left and No. 8 bottom right.</p> <p>Select the switch to be configured.</p> <p>Name</p> <p>Enter the name for the switch, in this case Proj. ON</p> <p>Click Done when complete.</p>
--	---

We have ignored the Pressed / Released option. During these tutorials, for simplicity, many of the unchanged attributes will be ignored unless they are relevant to the design. The 'Programmers Reference Guide' gives full details of all attributes and their functions.

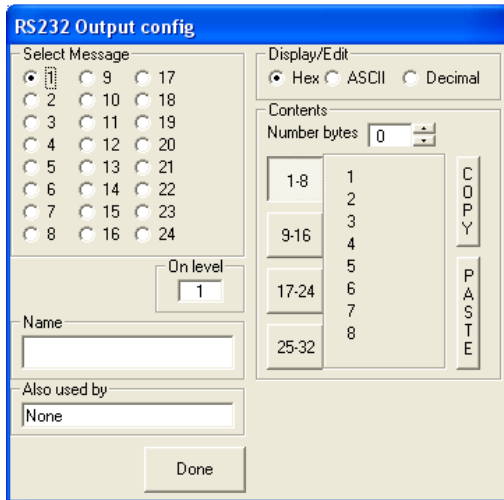
Also, whilst it is described as a switch, on a Wall Podule it is a push button and may be referred to as either within this guide.

Now Drag a RS232 Output  to the far side but inline with Switch. Double click to open revealing.



24 x RS232 output messages are available. Any messages already used will be 'greyed out' and cannot be selected

Click to select the next message, in this case message 1.

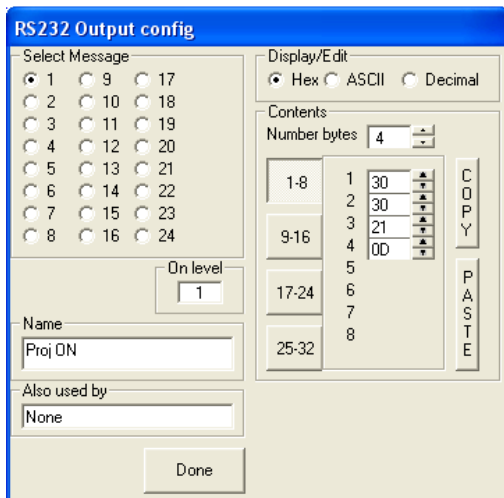


We now need to not only give the message a name but also enter the serial string needed to turn on the projector.

This information is usually available from the projectors manual or on the manufacturers website. If you are not certain how to interpret this have a look at [Appendix 2](#) for help.

1. Click in the Name field and enter the message name of Proj ON.
2. Look to verify that under 'Display/Edit' the Hex box is selected, if not select it.
3. Under Contents, against 'Number bytes' click the down arrow to select 4.
4. Against 1,2,3 & 4 there will now be a white box displaying 00.
5. Highlight the contents of box (byte)1 and enter 30.
6. Repeat for bytes 2,3 & 4 entering 30, 21, 0D respectively.

The RS232 config should not look like:-



This set of 4 codes is actually the ON command for a Mitsubishi projector. I have selected this one only because it is all Hex format and quite short.

The 0D byte is the command for carriage return and is quite common as the last byte for many projectors. If you accidentally entered 0d you should have noticed that the software automatically corrects this for you to 0D.

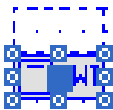
If / when you RS232 Output config look like the above click on 'Done' to close.

There is one final operation to do for setting up the RS232 and this is to set the baud Rate.

To the right of the screen there are four page tabs, Pod, RS232,----- & Notes. Full explanation in [Appendix 1](#), but for now click on the 'Pod' tab. Under this tab you will see PORT 1. From here you can set the Baudrate and Parity. For the Mitsubishi projector they are 9K6 and None.

Wiring

OK, we now have two devices but these need wiring together. Single click on the Switch, this highlights the device with a solid square at the centre.

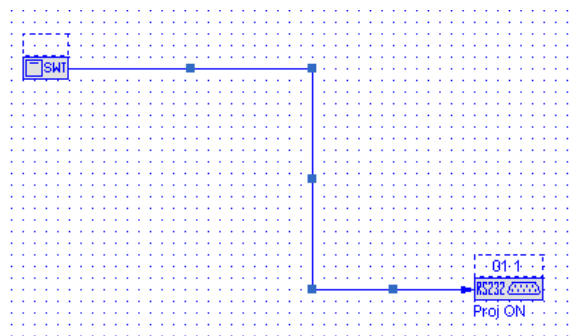


If you position the cursor on this black square the pointer changes to a vertical arrow. Click and drag a line onto the RS232 device then release.

You should now have a blue line connecting the two devices with a blue solid square in the middle. These devices are now connected.

For practice click on the RS232 device and drag it so that it is lower than the Switch, say about 20 grid spaces. Now click on the line (wire) to display the centre node. Click on this and drag the node so it is horizontal with the Switch but at least 3 grid marks away.

Two more nodes appear, one each side of the node you have moved. These allow further routing. Select the node nearest the RS232 device and move to create a right angle below the previous node and the RS2323 device - like this:-



You will see later that using this method allows devices to be laid out and wired logically to make the logic diagram easier to follow.

A further trick, useful later as you will see, is to change the colour of the line. If the line is still selected just click on the mouse's right button, or select first then right click. This opens a dialogue box, select the last entry Colour and then Selection. A table of eight colours opens, select the colour for the line and the menu closes after changing the line to the selected colour.

Right, playtime over, back to controlling our projector.

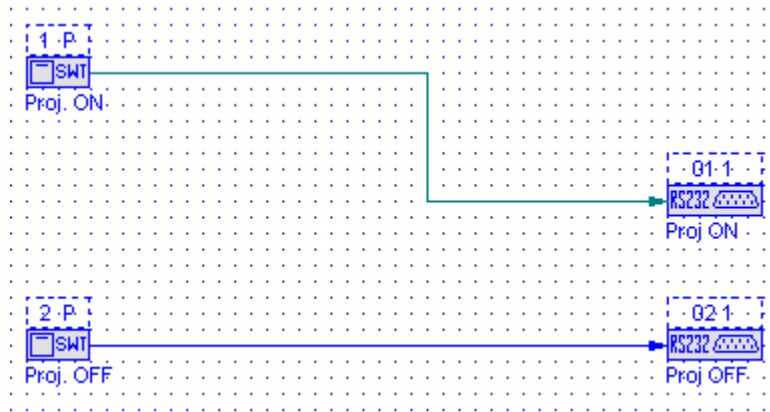
Projector OFF

So far we have assigned a push button (Switch) for turning on the projector and attached a RS232 output command to send the ON message, now we will add an off command.

The process is identical to the previous but this time we will use Button 2. The steps are:-


1. Drag and drop a Switch onto the working area.
2. Configure the switch as No. 2 and name Proj. OFF.
3. Drag and drop a RS232 Output onto the working area.
4. Set as message 2 and name a Proj OFF
5. Set for 4 x Hex bytes and enter the four bytes as 30, 30, 22 & 0D, there is no need to change the Baudrate or Parity.
6. Wire the Off switch to the RS232 Output.


You should now have a working area of the screen looking something like:-

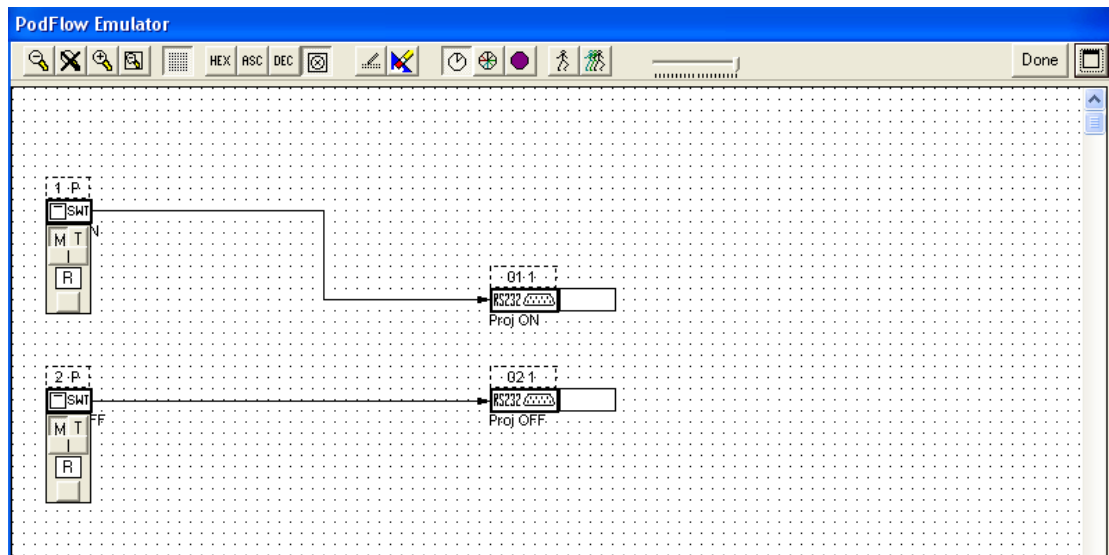


Emulation


One of the powerful features of PodFlow is the ability to emulate the design on the PC before downloading to the Podule. Selecting the emulation feature

does this, but before running, SAVE the file by clicking on the  button or from the file menu selecting save. If this is the first time of saving you will be prompted for a filename.

When saved select emulation by clicking the  button opening a new screen.



Before starting the emulator select HEX from the top tool bar to set the viewed message in Hex. Then reset all inputs by clicking on the button below the 'R' on both the switches to the left of the screen. This ensures that all inputs are set to off before starting.

You start the emulation by clicking the  button.

If you now press the button below the 'R' you should observe the attached RS232 command sending. The data box that opens when the send is activated shows the baudrate, parity and the bytes sent.

Press DONE to end and exit.

Congratulations, you have created your first working PodFlow program.

Now whilst we can download this to a Podule for use, it's not very interesting, so let's take the design further into a more useful program.

Tutorial 2b – Expanded projector Control via RS232

Building upon Tutorial 1, we are going to expand the design to incorporate the LED indicators within the push buttons. We will also include timers for Projector 'warm up' and 'cool down' timing.

Returning to the current PodFlow diagram, either select in turn the two wires or. Position the cursor to the top left above the wires only (not devices) hold down the left mouse button and drag a box to the bottom right of the wires only to select both wires.

Now press DELETE on the keyboard to remove both wires.

This technique of boxing can be used to select multiple devices & wiring to delete or move.

Next we are going to use a Flip-Flop to latch the Projector ON status.



From the Devices on the screen left drag and drop a Flip-Flop to the right of Button 1 (ON).

Take a wire from Switch 1 to the CLK of the Flip-Flop. Then click on the flip-flops OUT text to highlighted the solid block for a wire connection.

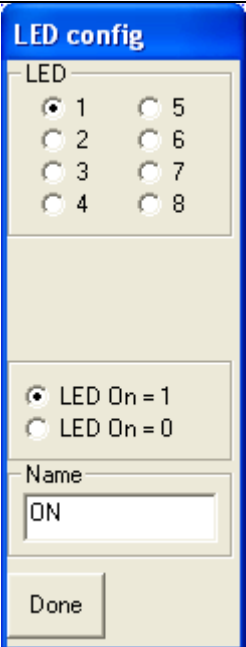


On devices with multiple connections, only the OUT connection allows a wire to be initiated from it.

From here drag a wire to the RS232 Output 1 (ON).

Now we will add an indicator to show when the Projector is ON.

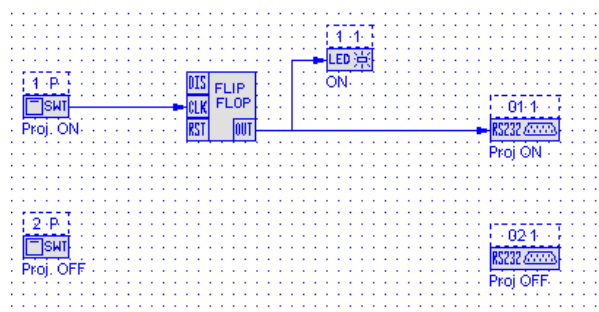
From the Devices drag and drop a LED  to the right of the Flip-Flop and double click to open the device parameters.

	<p>LED</p> <p>The Led location corresponds to the physical position of the switch into which it is fitted.</p> <p>Select the LED to be configured.</p> <p>Name</p> <p>Enter the name for the LED, in this case ON</p> <p>Click Done when complete.</p>
--	--

Now take another wired from the OUT of the Flip-Flop and link to this LED.

This is an opportunity to add corners.

It should look something like:-



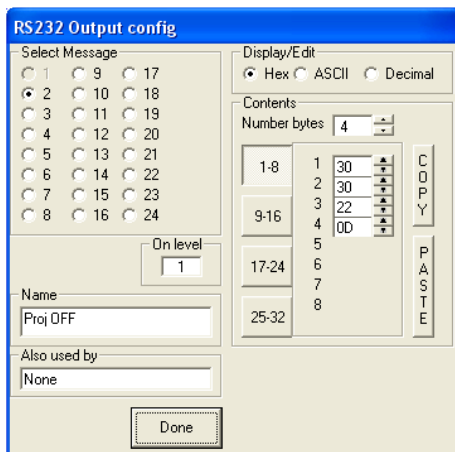
Flip-Flop Operation

Connected in this way, as soon as the Flip-Flops CLK connection receives an input from operation of the Switch it's OUT goes high, sending the RS232 ON message and lighting the LED. A second operation will turn out the LED but not send the message. A third operation will send the message and illuminate the LED again.

The reason for this is that with only an input on CLK, the Flip-Flops output changes state on each Low to High input change. The RS232 also only transmits on a Low to High change. The LED however shows the status of the signal connected to it.

Adding Projector Off

To integrate the projector OFF command we need to link the Flip-Flops output to the second RS232 send, the one for OFF. Now this would result in both ON and OFF being sent together, not very useful, so when you have wired the connection double click on the RS232 device.



Take a look at the box 'On level'. This will contain a 1 indicating that the message will be sent on a 0 to 1 (low to High / Off to On) transition on the devices input.

Click on the box to change to '0', a 1 to 0 / High to Low / On to Off transition.

With the above change the OFF message will only be send when the Flip-Flops output turns off.

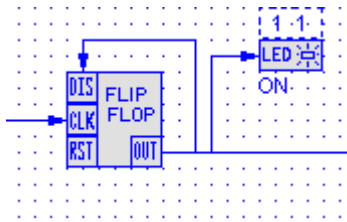
Try emulating this to confirm the results.

This latest change changed the On button an On/Off button, you can re label it but if you still prefer separate buttons – read on.

Separate On/ OFF Buttons

The RST input of the Flip-Flop is used to force the output Low (reset it) so if you connect the OFF switch to this input you can use the ON push to turn on and the OFF push to turn OFF.

Now the astute among you will have noticed that the ON button can still toggle between ON and OFF on alternate operations, not an ideal situation. This is where the last input of the Flip-Flop is used.



By connecting the OUT back to the DIS input changes the operation. Now the CLK input can ONLY set the Flip-Flop on whilst the RST input can only turn it off.

It has now become a Set / Reset latch.

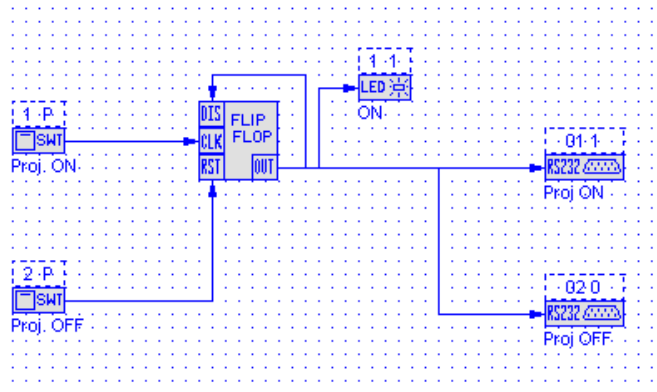
Try emulating to verify operation.

Whilst this is a very workable solution for projector control, we can add some very useful 'bells & whistles' to make the design not only more elegant but also much more user friendly.


Tutorial 2C – Adding Timers


We will start by adding timers for the Warm Up and Cool Down times and link these to the LEDs.

You should be starting with:-



Click on the wire connecting the Proj. OFF switch to the Flip-Flop and delete.

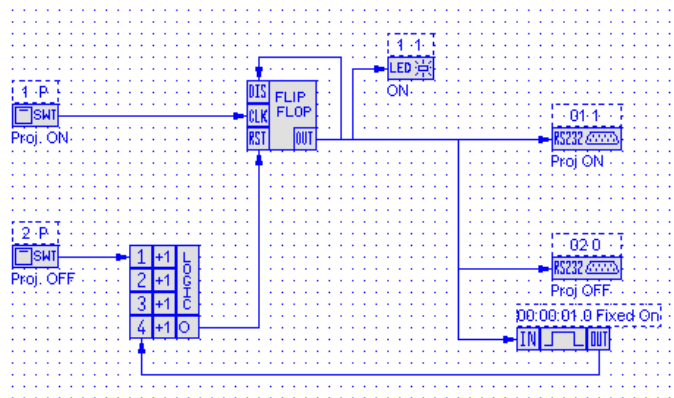
From the Devices select a Timer  and drag and drop just below the Proj OFF RS232.

Next select a Logical Gate  and place this to the right of the Proj.OFF switch with the 1 input in line with the Switch.

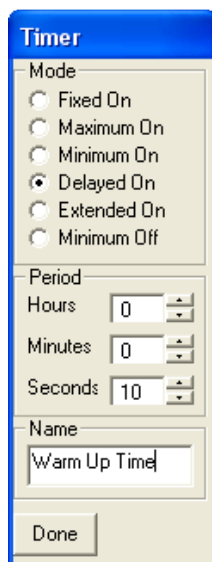
Take a wire from the Flip-Flops OUT and connect to the IN of the timer. Now take a wire from the Timers OUT and link this to the 4 input of the Logical Gate. You may need to add corners to visually simplify the routing.

Just two more wires, one from The Proj. OFF switch to the 1 input of the Logical Gate and finally one from the 0 (out)of the Logical Gate to the RST input of the Flip-Flop.

The diagram should look something like:-



We now need to configure the timer. Double click on it to open.



Mode

You can set the timer to operate in different ways, for a full explanation refer to [Appendix 3](#) or the 'Programmers Reference Guide'.

For this application select Delayed On

Set the Period to 10 Seconds by either typing the value in the box or using the Up and Down arrow buttons.

Give the Timer a name, in this example 'Warm Up Time'.

Setting to 'Delayed On' sets the timer to only turn its output ON / High / 1 the set Period after the IN is triggered.

The Period is set to 10 seconds, as this is only a demo. In reality it would be as specified by the projectors manufacturer, typically 40-50 seconds.

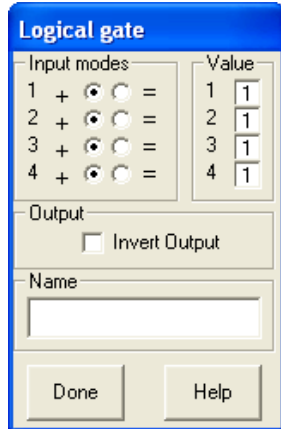
The operation is that now when the Flip-Flops OUT goes High, after operating the Proj.ON switch, the timers output will not go High until 10 seconds later, ie when the projector has warmed up.

Logical Gate

Lets now turn our attention to the Logical Gate.

This is quite a complex logic device as it can replace a number of different AND, OR, NAND, NOR and EXOR functions, many of them simultaneously.

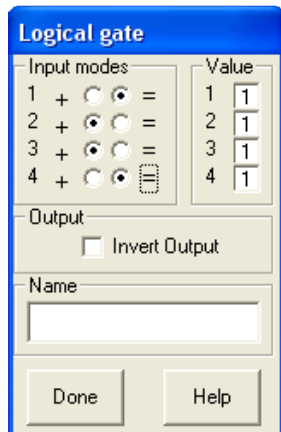
Double click on the device to open.



When opened all the inputs will be set to + and the Value at 1 creating a 4 input OR gate.

To ignore an input leave at the default setting of + and 1.

For our application we are using inputs 1 and 4 and need to make an AND gate, that is input 1 AND input 4 must be High for an output. Clock in the = box for both inputs.



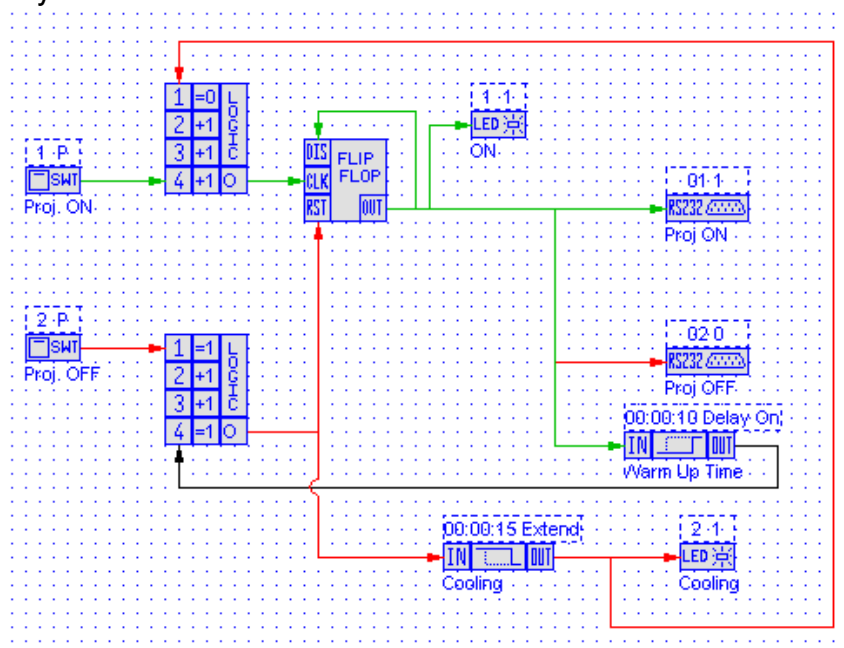
With these two set if the Proj. OFF push is pressed but not timer has not completed it's 'warm up' period nothing happens. Similarly if the timer has finished but the button is not pressed, nothing happens.

In this way we stop someone trying to turn off the projector until it has finished its warm up cycle.

Emulate the design so far to make certain you understand what is happening.

In a similar way we should really stop someone trying to turn on the projector whilst it is cooling down. It would also be a good idea to give some indication of this.

This is easily achieved by adding a further time and gate as below. By now I am certain you know how to add and connect the devices.



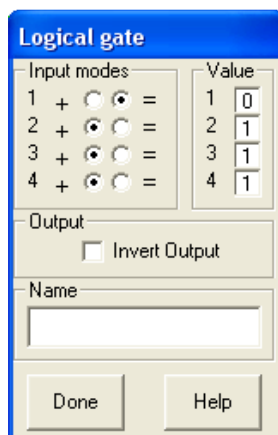
For the first time I have used colours to aid understanding the diagram. My personal convention is to use Green for 'ON' related commands, Red for 'OFF' related commands and Black for 'Enable' commands. It is purely arbitrary.

Lets take a look at the new additions.

Cooling Timer This is triggered when the OFF command is sent to the Flip-Flop. I have used an 'Extended On' timer with a 15 second period. This type of timer needs only a momentary input trigger, as you will get from pressing and releasing the OFF push. The intention is that its Output will remain HIGH for long enough to allow the projector to cool. I have set this to 15 seconds for the demo but in reality it will be between 1 and 2 minutes.

The timer output feed a LED, the one within switch 2 so this is illuminated whist the projector is cooling. The output also feed the additional Logical Gate between the ON push and the Flip-Flop.

Logical Gate The settings on this gate are different to before. Previously we needed two HIGH inputs to allow the OFF gate to trigger, in this case we need the feed from the OFF timer to be low to enable the ON button to operate.



By setting input 1 to = and 0 we are saying that input 1 MUST be at 0 to allow any of the other inputs to be active.

We could set input 4 to = as before but there is no need. With input 1 as an AND input (=) this is achieved.


OK, try emulating again – but don't forget to save first.

There is one final timer to add.

It's most annoying when you accidentally press the OFF button and the projector starts shutting down, those precious minutes waiting for it to cool down and restart can appear very long.

Why not introduce a simple 'Press and Hold' into the off function.

To do this delete the wired from the Proj.OFF push and insert a Timer device between it and the Logical Gate. IT may be that you need to create some room so use the same boxing technique as earlier (when we deleted multiple wires) but this time select ass the diagram and drag it to the right. Now drag the Pushes back over to the left to give you room.

Probably part of the diagram is not hidden under the right tab fields, you can either live with this and use the windows scroll bars to navigate or for now turn off the tab fields. To do this just click on the  button on the top menu bar. This toggles on and off the tab fields.

Having created room, connect the Proj.OFF button to the timer's input and the timer's output to input 1 of the Logical Gate. Now double click on the timer device to open and set for Delayed ON and 3 Seconds.

Now to turn off the projector you need to press and hold the OFF button for 3 seconds before it operated.

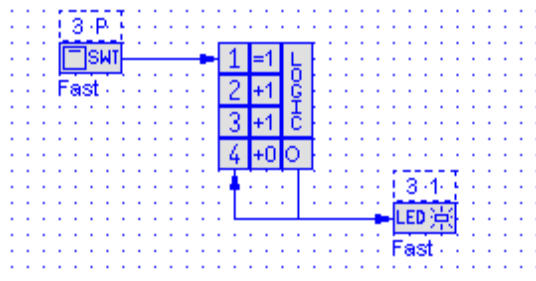
Tutorial 2d – Flashing LEDs

In this final part relating only to Projector ON / OFF functions, we will make the ON and OFF LEDs flash during Warn Up and Cool, Down. Again this is really more 'Bells and Whistles' but it does give very useful user feedback so that they know something is happening.

To create a flashing circuit we use the Logical Gate and another timer as per the examples below.

In both these I have used another Switch and its LED just for demo.

Fast Flashing



Here the Logical Gate has input 1 set to provide an AND function. When input 1 is High and Input 4 is low, the output is High.

When Input 1 goes high the LED attached to the output flashes fast.

Explanation

For those of you who want to know how it works. Not necessary at this stage, but here goes – just skip if not interested just now.

The output is looped back to input 4 to exploit the software delay through the device, about 100mS. At rest as Input 1 is Low the output, and therefore Input 4 will be low and the LED will be off.

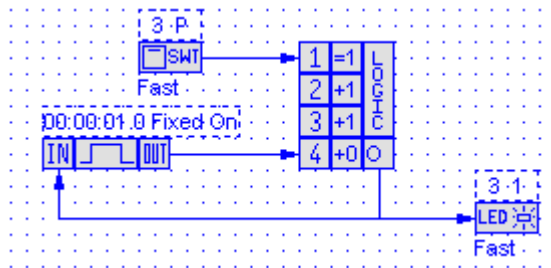
As soon as input 1 goes high, as input 4 is low the output goes high. Now this actually happens after the maths has been done by the software so the output actually changes 100mS after Input 1 goes high.

Now as Input 4 is now High it forces, after 100mS the Output low – and off we go again causing the LED to flash at 100mS intervals.

Try emulating this to see the results; the flashing may be slightly erratic under the emulation as it has to interact with the PC. On the Podule itself it is fine and even.

Slow Flashing

Now we can expand this by inserting a timer:-




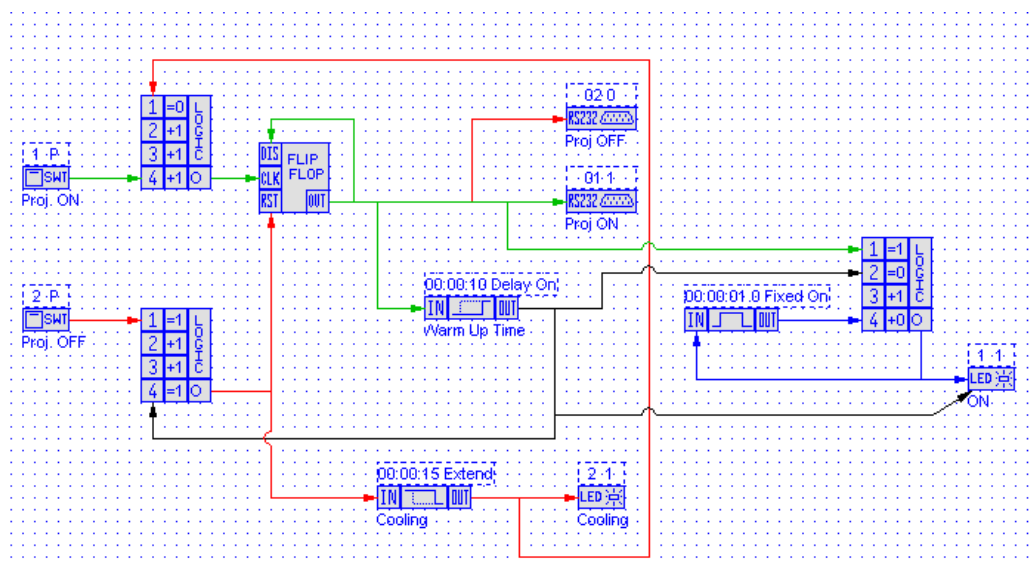
By inserting the timer, set for 1 second, we have introduced a delay into the flashing to give a slow, 1 second, flash.

Adding the Warm Up timer

To add the 'Warm Up' timer to the diagram you need to drag and drop a timer and a logic gate. These need configuring and wiring as shown above.

The following diagram has these added and connected. I have created more

space by turning off the right hand 'tab' menus by pressing the  button on the top tool bar. I have also rearranged some devices to create more room.



The ON led indicator has been moved from the Flip-Flop output and reconnected to the output of the logic gate.

Logic Gate Inputs

In addition to input 4, which is configured for flashing, inputs 1 and 2 are now used.

Input 1 is set to be an AND gate (the = sign) and so the flashing will only occur when this is high (logic 1).

Input 2 is also set as an AND gate but this time it need to be lo (logic 0) for the flasher to operate.

Operation

As soon as the main Flip-Flop is set ON, sending the RS232 ON string, Input 1 of the ON leds logic gate is set high and, as input 2 is lo, the flashing circuit starts and flashing the ON led in approx 1 second intervals.

At the same time the 'Warm Up' timer is activated and after the set time (10 seconds in the demo) its output goes to a high state. As this now changed the logic gates input 2 from lo to high the flashing of the ON led will stop.

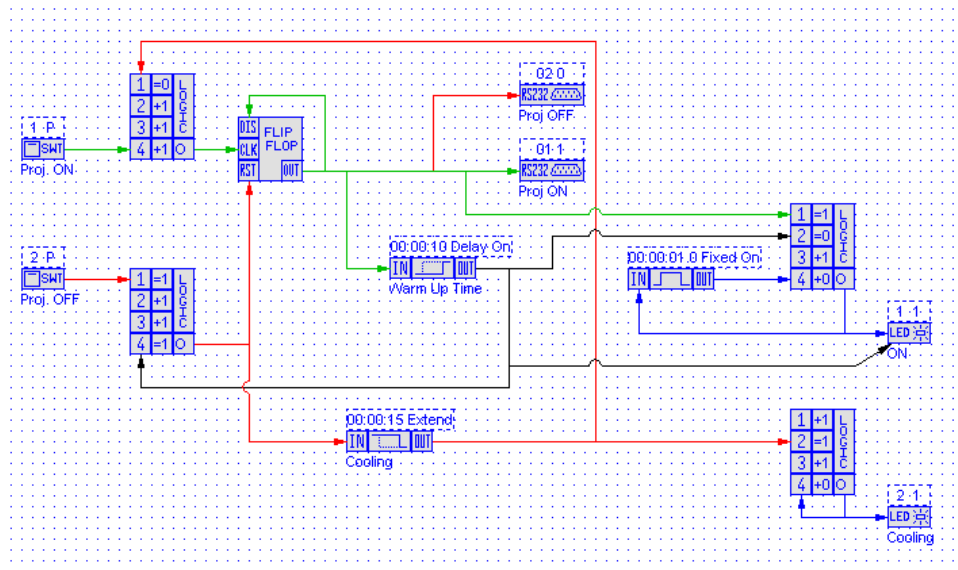
We obviously need the ON led to stay on whilst the projector is on. To achieve this it is simply a matter of connecting the (now high) output of the 'warm up' timer to the led.

As is common with most PodFlow devices, inputs are an OR gate allowing multiple connections without any additional logic.

Adding the Cooling Down Timer

To add the 'Cooling Down' timer to the diagram you need to drag and drop another logic gate. These need configuring and wiring as shown above.

The following diagram has these added and connected.



As with the 'Warm Up' timer, the LED has been disconnected from its previous connection and connected to the logic gates output. Also the output of the extended on 'cooling' timer has been connected to input 2 of this logic gate. With this input set as an AND gate, the led will only flash whilst input 2 is high.

So, whilst the cooling cycle is active and the cooling timers output is high, the Cooling led will flash fast. As soon as the timing cycle completes the led is off.

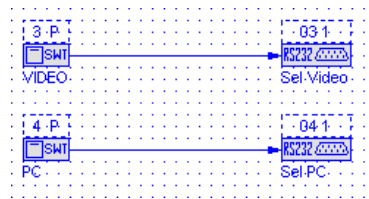
Tutorial 3 – Completing Basic Projector Control

This tutorial will further expand control of the projector (or other RS232 display device) with the addition of input selection, image blanking and control over volume.

Tutorial 3a – Projector Input Select

Having now completed the On and OFF control of the projector, we need to add the ability to select different inputs

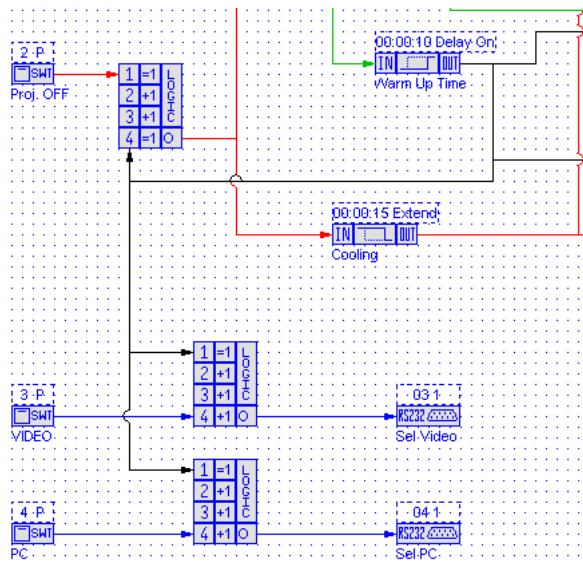
We could quite simply add buttons linked to RS232 outputs:-



The problem with this is that if you send the command whilst the projector is warming up it will most probably be ignored. Also, there is nor indication of what input is currently active.

Enabling Buttons

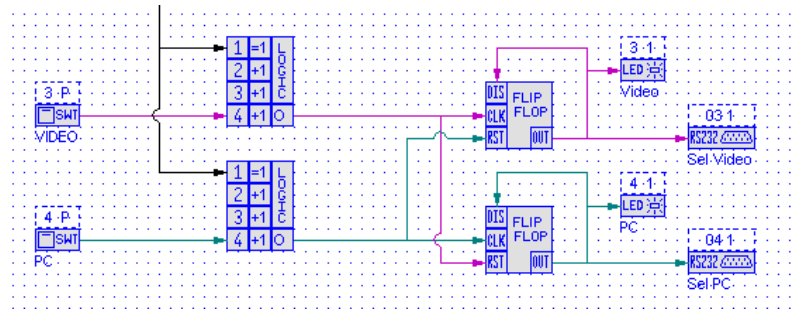
We can use the 'warm up' timer to enable buttons.



Here we are using the output of the 'warm up' timer to enable the operation of the Video and PC select buttons by connecting the timers output to input 1 of a logic gate. The logic gate is configured with input 1 as an AND function, so it must be high (logic 1) for the button connected to input 4 to send the RS232 command.

Interlocking Buttons

Expanding further on this we can add additional flip-flops to control led button indicators for conformation of the input selected.



The flip-flops have been inserted between the buttons logic gate and the RS232 transmit. OUT has been connected back to DIS to create a Set & Reset Flip-Flop.

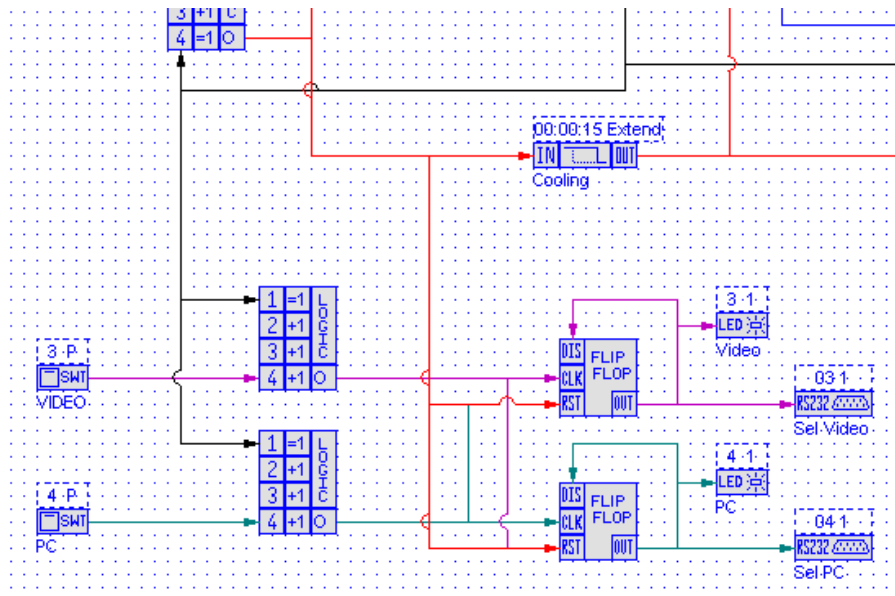
Assuming the projector is on and therefore the two logic gates are enabled, pressing VIDEO will set its Flip-flop ON and transmit the RS232 string. It will also illuminate the LED.

Repeating this for the PC push will sets its flip-flop ON, transmit the string and turn the led on. However, as the output from the buttons logic gate is also connected to RST (reset) of the Video flip-flop, it will turn this off and therefore the associated led off.

This can be expanded to cover more inputs with the logic gate out of each button connect to the RST connection of all other flip-flops.

Wire it up and give it a try under emulation.

One final bit to add in this section. This is the automatic turning off of the input select leds when the projector is turned off.

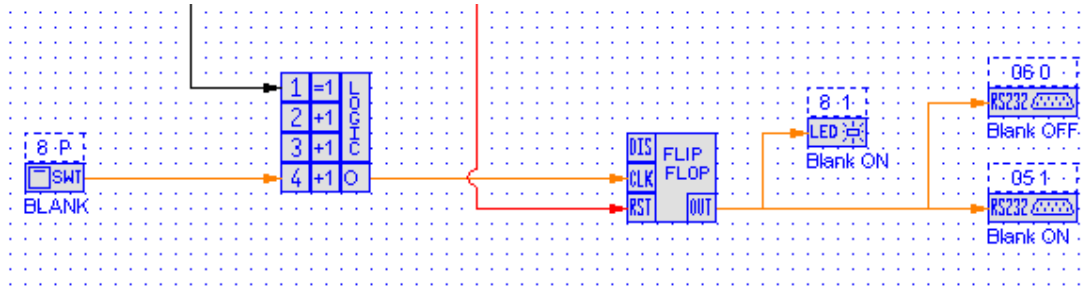


Connecting the output of the RESET flip-flop to all RST inputs of the input select flip-flops will turn all these off when the projector is turned off.

Tutorial 3b – Picture Blank

Picture mute commands are normally a toggle action, send one code to blank the image and another to reveal it again.

The logic required is very similar to that for turning the projector on and off, it just need incorporating in a slightly different way. Have a look at the diagram below:-



Here we have used a logic gate to enable the buttons operation when the projector is on and warmed up (the black line to the top left that is fed from the warm up timer).

Successive operation of the Blank button will send the Blank on and Blank off RS232 strings illuminating the Blank led when in Blank mode.

When the projector is turned off, as with the input select buttons, the blank flip-flop is reset.

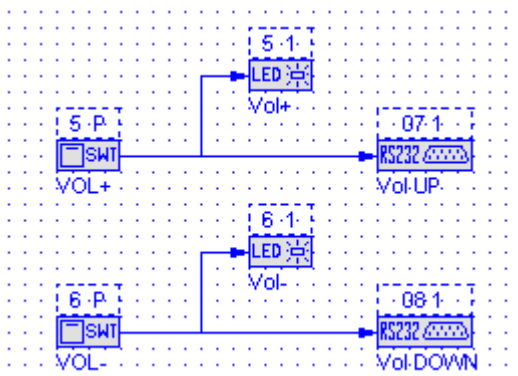
Tutorial 3c – Volume

Control over volume on a projector or other display is only possible if the unit supports incremental adjustment. That is you send a command to increase or decrease the volume by a set amount.

If the unit requires you to send an absolute value (i.e. go to -10dB, go to 0dB etc) you cannot control it from any Podule other than the touch Podule.

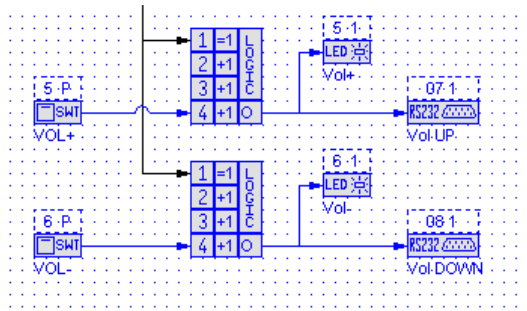
OK, Disclaimers over, lets get down to how.

Simple Control



The simplest way is to just connect the appropriate buttons to a RS232 transmit for the command.

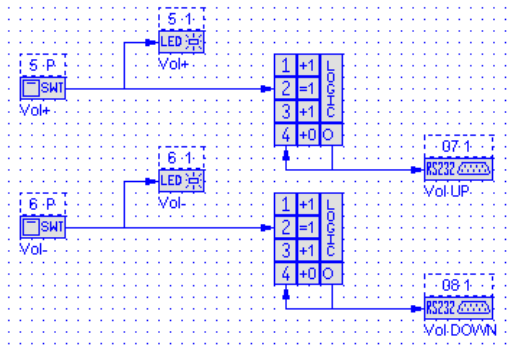
The button leds can also be added.



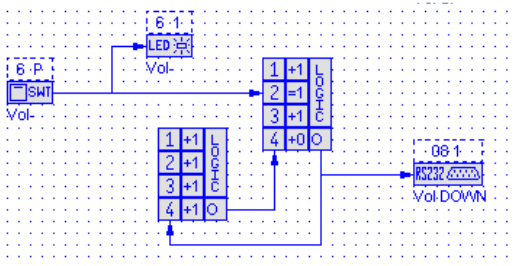
You can inhibit operation of the volume buttons as per the input select buttons but it's normally not worth it, as the volume buttons will have no effect if the display is off.

Volume Ramp

On both the above you need to press the volume button each time to adjust the volume, so if you need a large volume change or the incremental steps are small, you will need multiple button pushes.



Incorporating the same logic as used for fast flashing will send the RS232 command once for a quick button push or repeatedly whilst the button is held.



If the speed of volume ramp is too quick, try inserting an additional gate in the feedback to half the rate.

Tutorial 4 – Advanced Projector Control

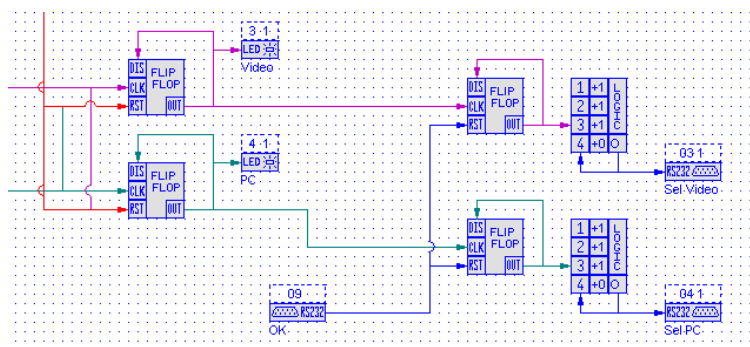
Whilst many projectors are quite happy to operate with a received RS232 input only, there are a growing number that ignore RS232 comms if they are busy doing other tasks (typically adjusting input parameters).

Using the Podules bi-directional RS232 control you can configure a Podule to create two-way communications with the projector to circumvent any problem this may create.

Send until Acknowledged

Some projectors, typically some Sony and NEC models among others, require the continual transmission of a input command with this stopping as soon as the projector acknowledges it has been received.

To achieve this, once an input is selected, we need to create two states for the output, one where the RS232 signal is continually sent and another then the transmission is acknowledged. This is easily achieved by incorporating a further flip-flop to 'latch' the 'continue to send' state with this reset when an acknowledge signal is received.

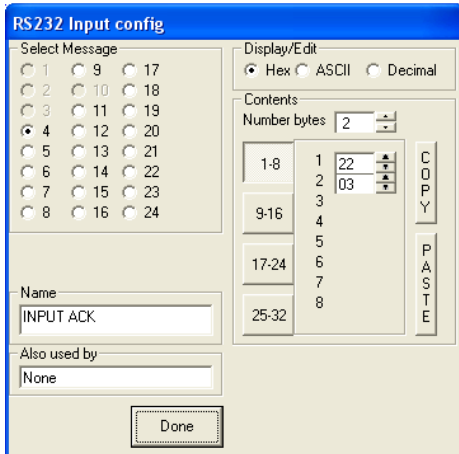


Based upon our previous Wall Podule program, here a flip-flop has been added after the input select flip-flop. This flip-flop has its OUT connected back to DIS to make it a Set / Reset type. Flip-Flops are edge triggered and only recognise an input when it changes from Lo (0) to High (1). It ignores both a continuous Lo and High state. It will set ONLY when the preceding flop-flops output goes High.

This new flip-flop enables a circuit similar to the one used to fast flash a led only this time continually transmitting the RS232 string. [Like the flip-flop, the RS232 transmit is also edge triggered so it will send on each Lo to High input transition].

The RS232 transmission continues until the projector accepts the command and replies with the OK string. This is received by the RS232 input device and used to reset the flip-flops.

You will obviously need to know the reply string from the projector. In the example above it is the same string for any input. You can use separate RS232 inputs per flip-flop if they are different.

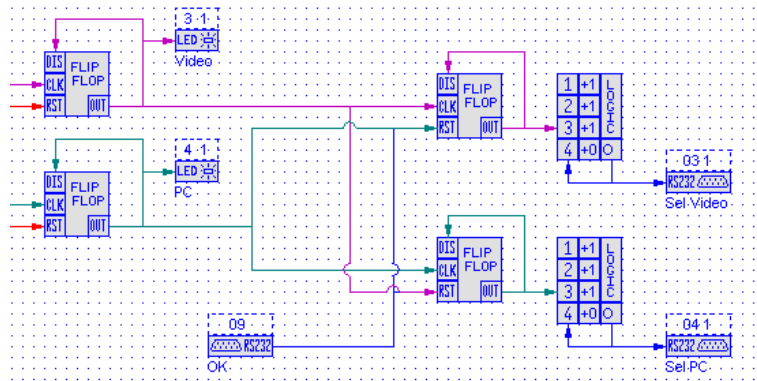


The expected receive strings are entered into the RS232 Input Config in the same way as the output strings.

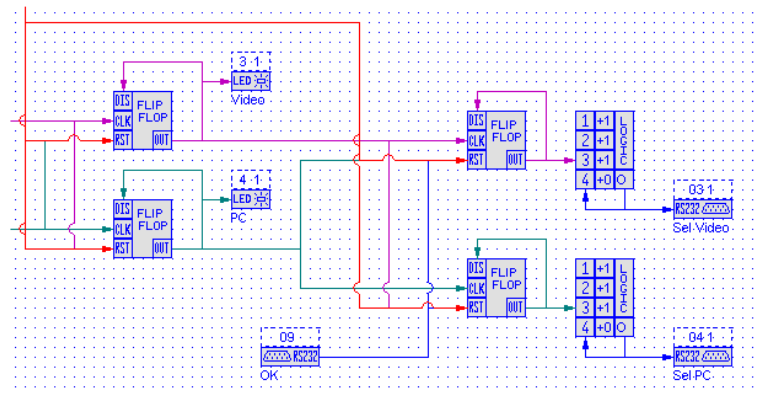
Tip. It is only necessary to enter the bytes up to and including any that are different for the acknowledgements and omit those that follow.

Now, whilst what we have created is a working solution, it lacks certain refinements for reliable operation. For instance, if you select video and before the projector changes then select PC, both strings are being sent, not an ideal situation.

The solution is to cross connect the inputs and RES, just as we did with the input select buttons in Tutorial 3a.



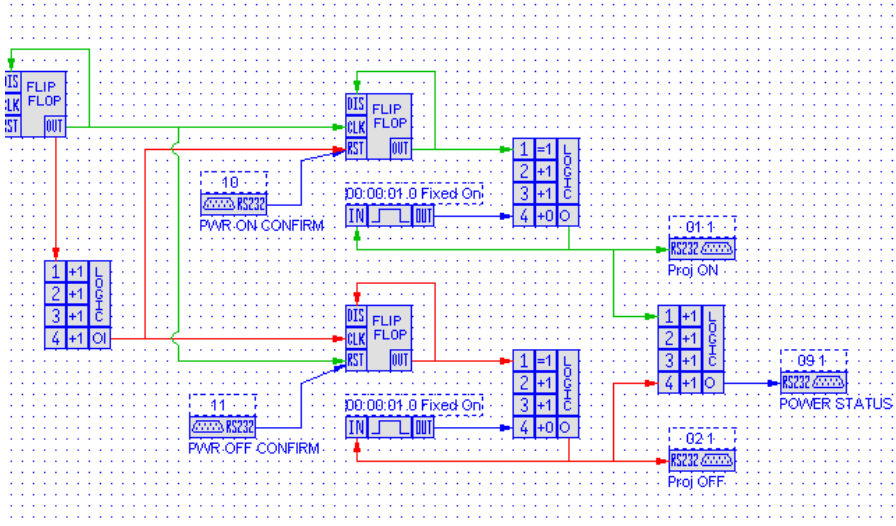
A final refinement is to ensure that if you turn off the projector before an input selection is made, the transmissions are ended. This is achieved by also connecting the Reset to the new flip-flops RES input.



Send then Interrogate

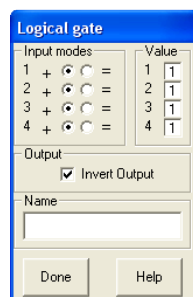
With some projectors, if you send a Power On command whilst it is still cooling, it will respond with an acknowledgement of reception but ignore the message as it is cooling. You now have a situation where the Podule thinks the system should be on whilst the projector is off.

Here we need to send an ON command and then ask the projector for its status. If this is incorrect, send the ON command again and ask again for status.



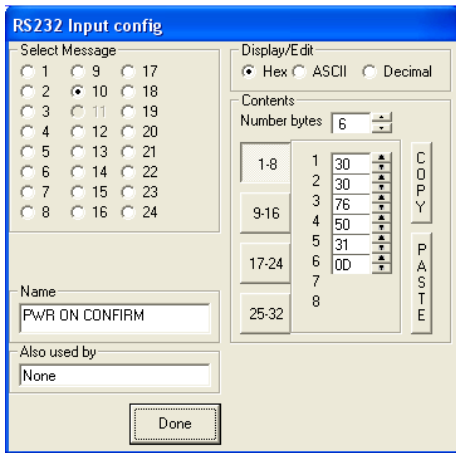
In principle this is an extension of the previous example. Lets first take a look at the flip-flop and logic gate to the right.

The flip-flop is the main projector On/Off latch, as used in Tutorial 2c. The main difference though is that whilst we previously used the Lo to High transition to send the Projector ON string and the High to Lo transition to send the Projector OFF string, here we cant do so. Instead we still use the Lo to Hi for initiating the ON string but use a logic gate as an inverter to create a Lo to Hi transition for the OFF string when the flip-flops output goes Lo.



To create a simple inverter within the Logic Gate, just click on the Invert Output box. Now whichever input goes high, the output goes low. Likewise as the input goes lo the output goes high.

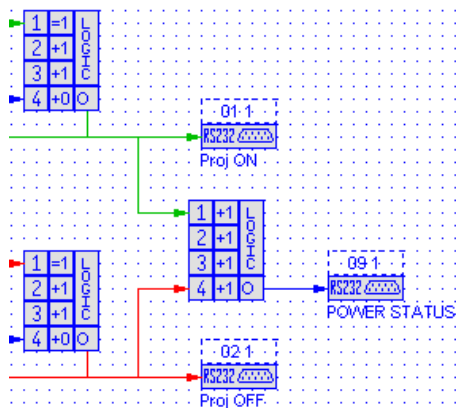
The rest of the Projector ON logic is similar to the previous example except we have added a 1 second timer to the pulse circuit created around the logic gate. This will now send the RS232 string approximately every second. The flip-flop is cleared not by the message-received string but by Power On confirmation string (see below).



This string, for a Mitsubishi projector, is only sent by the projector when the projector has started its warm up cycle.

It is sent in response to a status query command.

Returning to the main diagram.



In addition to sending the Proj ON string, the pulse generators output also sends a Power Status request string.

As this string is after a logic gate it is delayed approximately 100mS behind the ON string.

The same gate and RS232 output is used for the OFF string.

Tutorial 5 – PodNet

PodNet is the proprietary network used to link Podules together. It is NOT Ethernet and should never be connected to Ethernet or damage may occur (to the Ethernet devices).

PodNet can carry Digital and Analogue messages but the creation and use of these is determined by the Podules connected.

For example, the Wall Podule only supports switches and RS232 messages, so all PodNet IN and OUT messages will be Digital. However the CIR Podule has both Digital and Analogue inputs and outputs so it can use both Digital and Analogue messages.

For this set of tutorials we will concentrate on Digital PodNet messages only. Analogue messages are covered later on in Tutorial #.

Message Processing

Each PodNet message is given a message number. At present Podules support messages from 1 to 1024.

PodNet is carried between all Podules connected together. Only if a Podule has a PodNet inputs will it look for the message numbers on the network and respond accordingly when received.

Message Numbering

This can be quite arbitrary but you should only have one Podule origination a message with as many as required receiving the message. On the Ikon website we have a spreadsheet available for download (PodNet Addresses.xls) that simplifies keeping track of messages on larger systems.

The format is:-

PROJECT =

DATE:

PodNet	USE	Origin	Dest. 1	Dest. 2	Dest. 3
1	Select DVD	Wall 8	Screen	CIR	Audio
2					
3					
4					

Tutorial 5a – Using Digital messages

Consider the following:-

You have a Wall Podule controlling a projector and you now want to add an electric screen. Using a Screen or Relay Podules, connect via PodNet this is very easy to achieve.

RS232 Input



The screenshot shows a dialog box titled "RS232 Input config". It has several sections:

- Select Message:** A grid of radio buttons for selecting a message number from 1 to 24. Message 1 is selected.
- Display/Edit:** Radio buttons for "Hex", "ASCII", and "Decimal". "Hex" is selected.
- Size:** A field labeled "Number bytes" with a value of 2 and a spinner control.
- Contents:** A table with 8 rows and 2 columns. The first two rows contain hex values "0A" and "0D". To the right of the table are vertical buttons for "COPY" and "PASTE".
- Name:** A text input field.
- Also used by:** A dropdown menu currently showing "None".
- Done:** A button at the bottom.

When the RS232 input Device symbol is double clicked it displays a 'Select Message' window allowing the assignment of the message it is to respond to.

When a message is selected a side window opens, this is a repeat of the RS232 Tab and allows the setting up, or verification of the message length and content. A warning is also given if the message is already assigned within the Podule.

PodNet Signals

The PodNet currently supports up to 255 simultaneous messages with these a combination of Digital and Analog. The Wall Podule only supports Digital PodNet messages. You should only have a single Podule originating the message but all Podules can receive it. On any Podule, if the status of a message is changed the Podule transmits this change immediately, or if the network is busy in the next available time slot, this is generally within 5mS.

All Podules automatically retransmit their locally originated PodNet messages at an interval of approximately 35 seconds to ensure complete synchronisation between units.

PodNet input

A screenshot of a software dialog box titled "PodNet Message". It has a blue title bar. Below the title bar, there is a section labeled "PodNet address number" with a text input field containing the number "1" and a small spinner control. Below that is a section labeled "Name" with a larger text input field. At the bottom of the dialog is a "Done" button.

PodNet Message

Enter the message number to be used either by typing in the message number or using the buttons.

For convenience you can give the PodNet input a suitable name.

PodNet Output



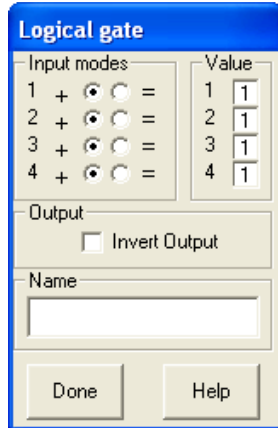
PodNet Message

Enter the message number to be used either by typing in the message number or using the buttons.

For convenience you can give the PodNet input a suitable name.

Podules will not 'see' on their input a transmitted message that uses the same message number as a PodNet output used on the same Podule. This is to eliminate the possibility of creating an infinite feedback loop.

Logical Gate Device

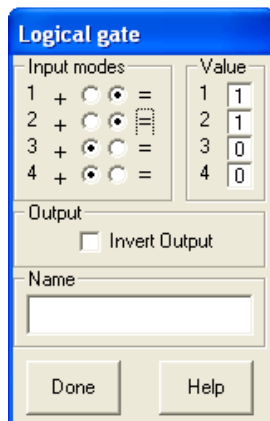


Each input can be set as an AND gate or an OR gate.

If, as shown, the + button is selected it is part of the OR gate.

As part of the OR gate the input will be active if the signal is equal to the number on the right. The example as shown, if any of the four inputs are Hi (1) for the gate output will be high.

The Invert Output box will change the status of the output from Hi to Lo when all inputs are Hi.



Inputs 3 & 4 are still part of the OR gate but looking for a low (0) to be valid - if unconnected effectively not used.

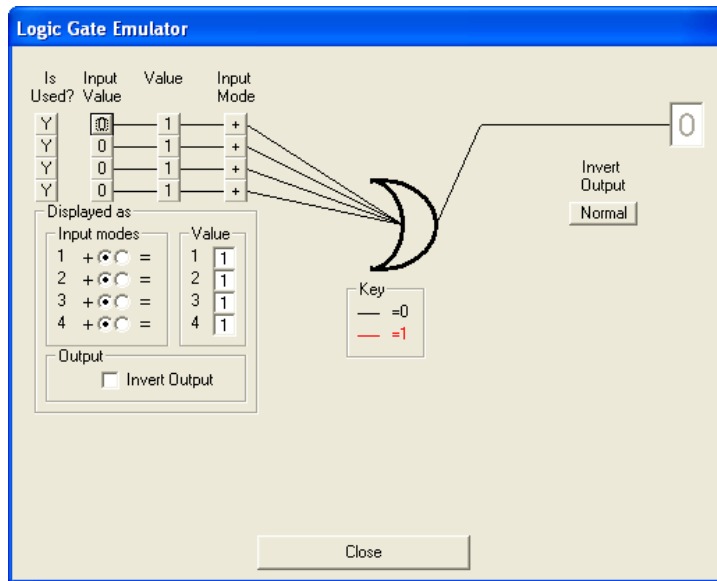
Inputs 1 & 2 have the = button selected and are part of the AND gate.

As part of the AND gate if either input goes high the output will be high.

The Invert Output box will change the status of the output from Hi to Lo when either input 1 or 2 are Hi.

Logical Gate Emulator

Under 'Help' an emulator allows a simulation of operation.



Is Used?

Select if the gate input is to be used or not.

Input Value

Set the logic level at the input, either 1 or 0.

Value

Set the level the input is to respond to.

Input Mode

This toggles between + for an 'OR' input and = for an 'AND' input. Changing these will alter the logic diagram to suit the required logic.

Invert Output

Lets you invert the output if required.

Displayed As

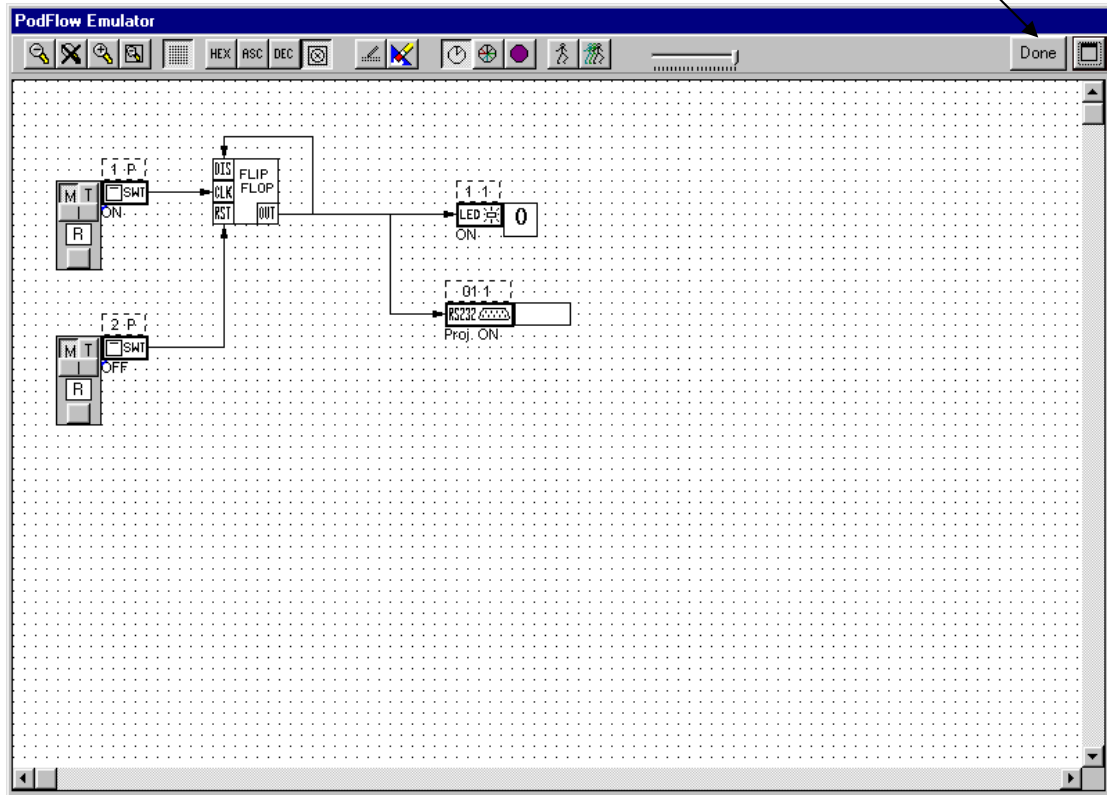
This group of radio buttons change to reflect the emulation settings and directly correlate to how the device itself needs to be set to achieve the emulated logic.



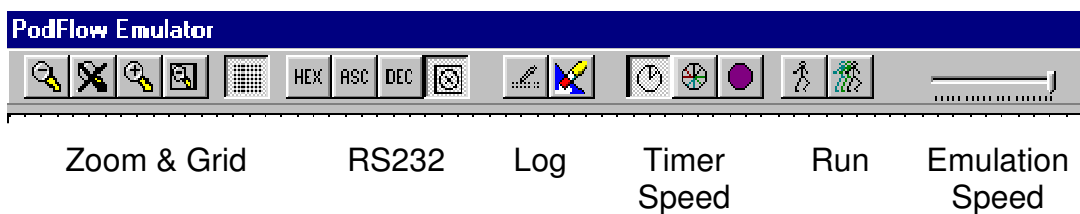
Emulation

The Emulation mode allows you to test your PodFlow configuration prior to loading into a Podule. Click on the Emulate icon on the toolbar to open the emulation.



EXIT Emulation











Emulation Toolbar



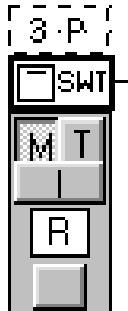
These operate as per the main PodFlow program.

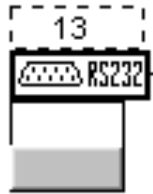
RS232	
	Select the format for RS232 messages.
	Shows devices as device image.

<p>LOG</p>  	<p>RS 232 output events are logged during the emulation.</p> <p>Use to view and hide the log.</p> <p>Clear the log.</p>
<p>Timer Speed</p>   	<p>Use to speed up timer operation during emulation.</p> <p>Run timers in real time, (default setting).</p> <p>Run timers at x10.</p> <p>Run timers at x60, i.e. 1 min in 1 second.</p>
<p>RUN</p>  	<p>Start the emulation.</p> <p>Start the emulation running.</p> <p>Step through each operation manually.</p>
<p>Emulation Speed</p> 	<p>Allows the emulation to be slowed down.</p> <p>As shown emulation is at 100% speed. Slide to left to slow to 5% minimum.</p>

Input Emulation

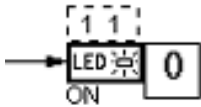
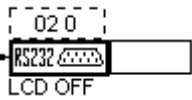

Each input has a set of controls to set the input state. These change between different types of inputs.

<p>Push Button Input & Digital PodNet Input</p> 	<p>When set for Momentary operation the Invert button is available. The devices output is only high when the lower button is operated. If Invert has been changed then the output is high until the lower button is operated.</p> <p>When set for Toddle the Invert button is hidden and each operation of the lower button toddles the output status.</p>
--	---


<p style="text-align: center;">RS232 Input</p> 	<p>RS232 inputs are emulated by operation of the button below the symbol. When operated PULSE is displayed in the window above the button and the devices output pulses high.</p>
---	---

Emulation Outputs

Each type of output device has it's own method of displaying the current status.

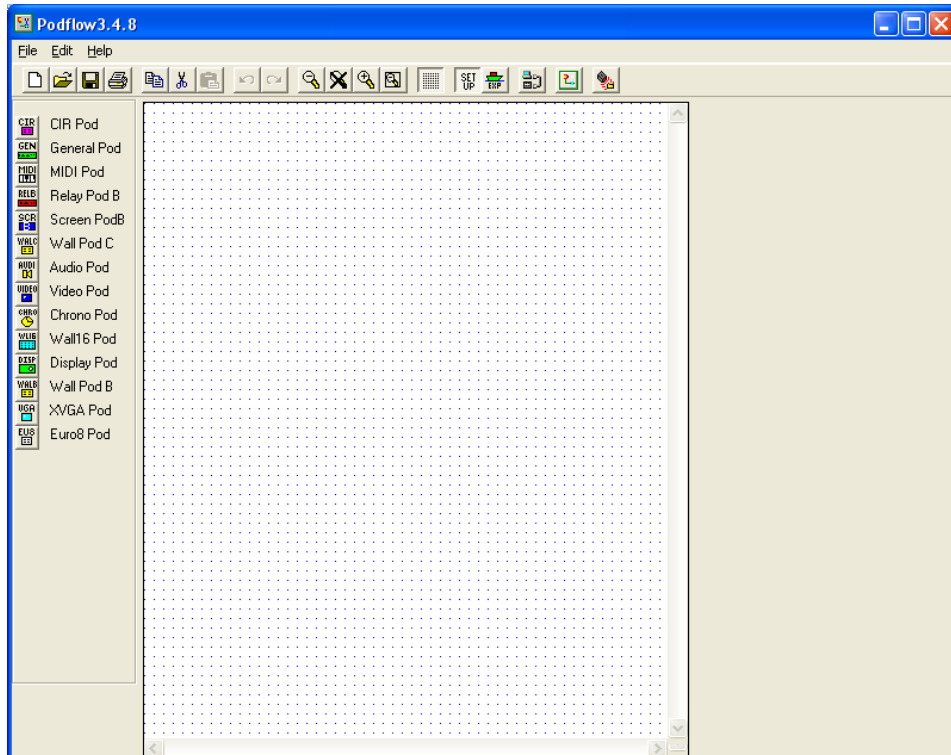
<p style="text-align: center;">LED Outputs</p> 	<p>The LED outputs display their current status as a flag showing 0 for off and 1 for on.</p>
<p style="text-align: center;">RS232 Outputs</p> 	<p>When an RS232 output is activated the message will be displayed to the right of the symbol. It is also logged.</p>
<p style="text-align: center;">PodNet Outputs</p> 	<p>Digital PodNet messages are shown with a '0' or '1' flag dependent upon their status.</p>

Running the Emulation

- 1 Zoom, maximise, turn on/off the grid etc. as per personal preference.
- 2 Set all inputs to the start condition, by default this is Momentary & Inverted.
- 3 Select RUN  .
- 4 Manipulate inputs as appropriate.
- 5 Change the emulation speed, timer speed etc if required.

APPENDIX 1

PodFlow Screen

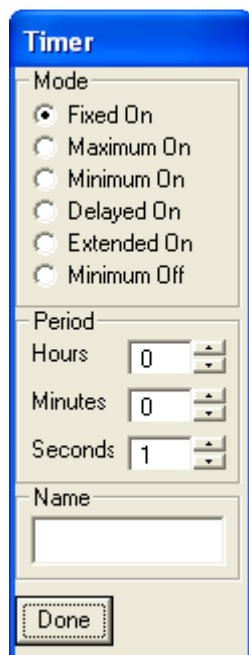


APPENDIX 2

Understanding RS232 message formats.

APPENDIX 3

Timer



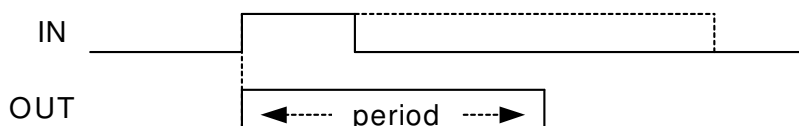
The device will only accept a single input for triggering and can be used in multiple timing modes. The time period can be set from 1 second to 18 hours.

The time period set along with the type of timer selected is displayed above the device.

Any name is shown below the device.

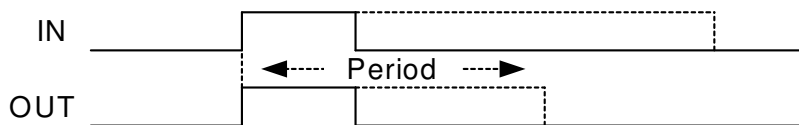
Fixed On

The output will follow the input state. It will remain High for the on timer period, independent of any other input changes.



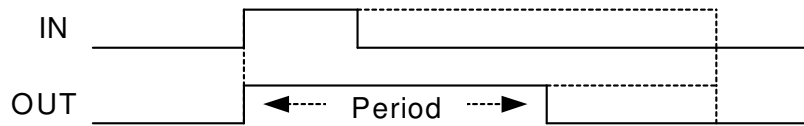
Maximum on

The output will follow the input state up to a maximum time set by the on timer.



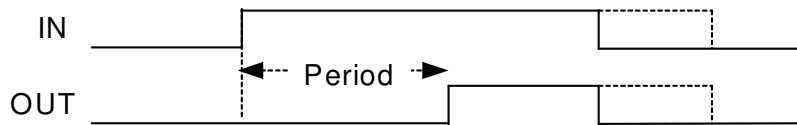
Minimum On

On for a set time even if the trigger is removed. The output remains high whilst the input remains high, even if the time period has expired.



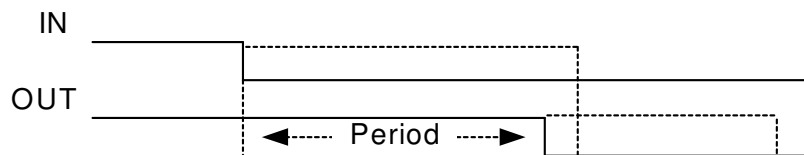
Delayed On

Output remains low for the set period after a Lo to Hi transition and then goes high whilst the input remains Hi.



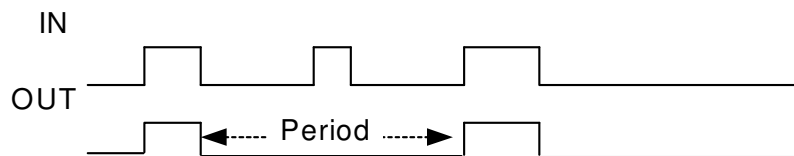
Extended On

Output remains High for the set period after a Hi to low trigger.



Min Off

Output will remain low for the set period, even if the input is re-triggered.



Period

Adjustable time in Seconds, Minutes or Hours

1 to 59 Seconds

1 to 59 Minutes

1 to 17 Hours

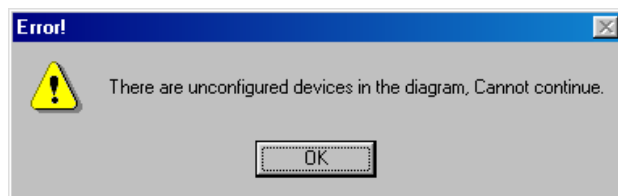
If no time is set it defaults to 1 second.

APPENDIX 4


Programming the Pod

Use a serial lead as described in Appendix D to program any of the Podules.

When you click on this icon  you may get a message



This is because some of the devices have not been set up properly. You may have an input device on you project that has no terminal selected.

Once all the devices have been corrected and you then press  A window will appear as it searches for pods, Fig 1, if none are found you will get a window like Fig 2, check cable connection and power to the Pod. Once communications has been established a window like Fig 3 will appear. This shows that there is a CIR Pod connected to com port 1. Click the radio button next to the Pod you wish to program.

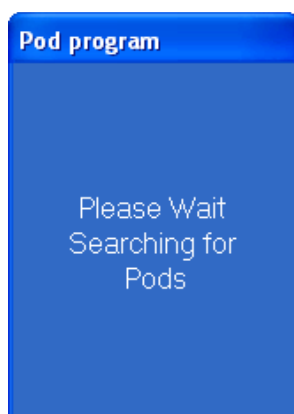


Fig 1

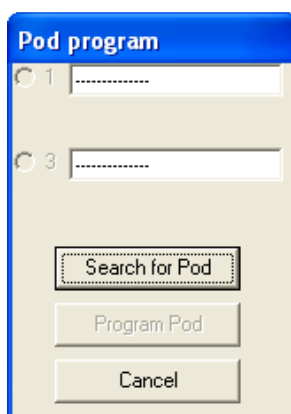


Fig 2

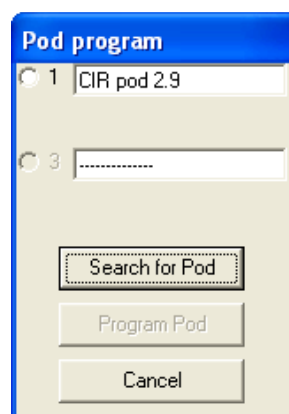


Fig 3

Note:- Fig 3, also shows the revision of the firmware within the POD, New versions, along with PodBootstrap to load them are available on our web site at www.ikonavs.com .

Whilst the PodFlow data is being sent the progress display as per Fig 4 is displayed with this changing to Fig 5 when programming is complete allowing the programming of a second Podule or exit.

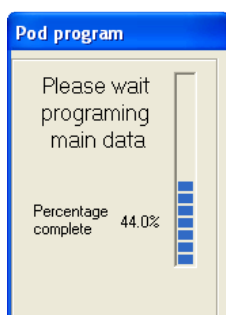


Fig 4

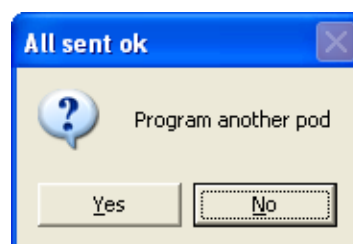


Fig 5

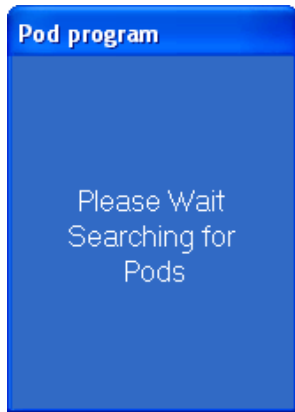


Fig 1

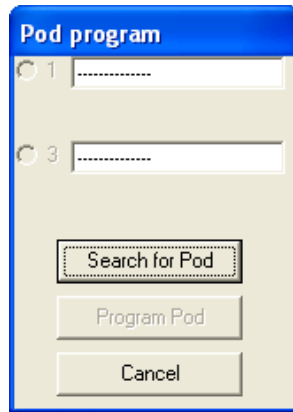


Fig 2

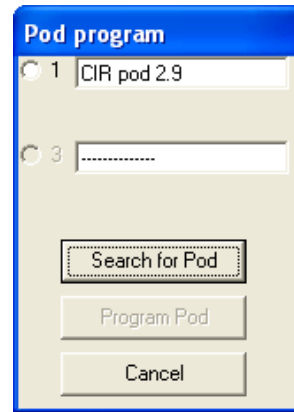


Fig 3

Note:- Fig 3, also shows the revision of the firmware within the POD, New versions, along with PodBootstrap to load them are available on our web site at www.ikonavs.com .

Whilst the PodFlow data is being sent the progress display as per Fig 4 is displayed with this changing to Fig 5 when programming is complete allowing the programming of a second Podule or exit.

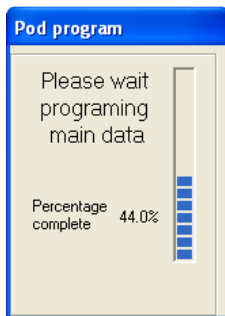


Fig 4

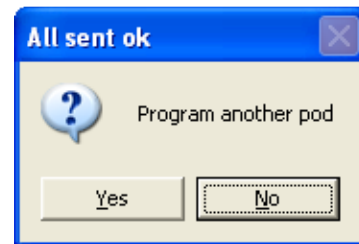


Fig 5

Manufacturers Information

The iKON AVS PODULES are manufactured in England by Nebula Audio Ltd.

For service or warranty advice please initially contact your supplier.
Alternatively contact the manufactures at:-

Nebula Audio Ltd
40 Saint Dunstons Close
Worcester
WR5 2AJ

Telephone: +44 (0) 1299 250991

Fax: +44 (0) 1299 250983

Website www.ikonavs.com

Technical support e-mail

support@ikonavs.com