



OpenLCB/NMRAnet Developer and Early Adopter Kit UNDERSTANDING DevKit *Ios*



The *Ios* in the DevKit have firmware loaded into them that does specific things. It makes use only of the *Ios* Input and Output connectors.

DevKit *Ios* are Special

In order to make the *Ios* in the DevKit immediately demonstrable, the *Ios* in the DevKit are programmed such that their inputs and outputs share the same event-numbers. This means that changes to inputs will be followed by the outputs of all boards. This is not typical *Io* behaviour, but we will show you how to revert the *Ios* to 'normal' behaviour.

Inputs

In response to a change in the voltage on an Input from 0V to 5V or 5V to 0V, the *Io* will send the event-number associated with state-change. For example, if you ground an input 0, then the *Io* will send a specific event-number, and will send a different one when you release the Input.

Outputs

Similarly, each Output is associated with two event-numbers. When the first of these is received from the network, the LED associated with that output will turn on, and the output goes to ground. When the other event-number is received, the LED turns off and the output goes to +5V.

How DevKit *Ios* are Special

Normally, each input state and output state has its own unique event-number associated with it. In order to make the DevKit *Ios* immediately usable in a demo mode, every *Io* has the same event-number assigned to similarly numbered Inputs and Outputs, thus:

Default	On	Off
Input 0 and Output 0	05.02.01.02.02.00.00.00	05.02.01.02.02.00.00.01
Input 1 and Output 1	05.02.01.02.02.00.00.02	05.02.01.02.02.00.00.03
Input 2 and Output 2	05.02.01.02.02.00.00.04	05.02.01.02.02.00.00.05
Input 3 and Output 3	05.02.01.02.02.00.00.06	05.02.01.02.02.00.00.07
Input 4 and Output 4	05.02.01.02.02.00.00.08	05.02.01.02.02.00.00.09
Input 5 and Output 5	05.02.01.02.02.00.00.0A	05.02.01.02.02.00.00.0B
Input 6 and Output 6	05.02.01.02.02.00.00.0C	05.02.01.02.02.00.00.0D
Input 7 and Output 7	05.02.01.02.02.00.00.0E	05.02.01.02.02.00.00.0F

The consequences of this are that the Output LEDs of all *Ios* will follow the Input states of all *Ios*. This allows you to set up an interaction between *Ios* by simply plugging them together.

Restoring your *Ios* to normal behaviour.

Normally *Ios* have a set of event-numbers that are unique to each node and to each input and output state. They can be reset to this behaviour by holding down both the blue and gold buttons for 8 seconds. At about 3 seconds the blue and gold LEDs will start flashing rapidly – this is to warn you that a reload is taking place. Each time you do this a new set of unique event numbers will be loaded into the node.

Doing this the first time you perform this reload, it will result in the following event-numbers, where xx will be the last two digits of your Io node ID.

Default	On	Off
Input 0	05.02.01.02.02.xx.00.00	05.02.01.02.02.xx.00.01
Input 1	05.02.01.02.02.xx.00.02	05.02.01.02.02.xx.00.03
Input 2	05.02.01.02.02.xx.00.04	05.02.01.02.02.xx.00.05
Input 3	05.02.01.02.02.xx.00.06	05.02.01.02.02.xx.00.07
Input 4	05.02.01.02.02.xx.00.08	05.02.01.02.02.xx.00.09
Input 5	05.02.01.02.02.xx.00.0A	05.02.01.02.02.xx.00.0B
Input 6	05.02.01.02.02.xx.00.0C	05.02.01.02.02.xx.00.0D
Input 7	05.02.01.02.02.xx.00.0E	05.02.01.02.02.xx.00.0F
Output 0	05.02.01.02.02.xx.00.10	05.02.01.02.02.xx.00.11
Output 1	05.02.01.02.02.xx.00.12	05.02.01.02.02.xx.00.13
Output 2	05.02.01.02.02.xx.00.14	05.02.01.02.02.xx.00.15
Output 3	05.02.01.02.02.xx.00.16	05.02.01.02.02.xx.00.17
Output 4	05.02.01.02.02.xx.00.18	05.02.01.02.02.xx.00.19
Output 5	05.02.01.02.02.xx.00.1A	05.02.01.02.02.xx.00.1B
Output 6	05.02.01.02.02.xx.00.1C	05.02.01.02.02.xx.00.1D
Output 7	05.02.01.02.02.xx.00.1E	05.02.01.02.02.xx.00.1F

The next reload will produce:

Default	On	Off
Input 0	05.02.01.02.02.xx.00.20	05.02.01.02.02.xx.00.21
Input 1	05.02.01.02.02.xx.00.22	05.02.01.02.02.xx.00.23
Input 2	05.02.01.02.02.xx.00.24	05.02.01.02.02.xx.00.25

Input 3	05.02.01.02.02.xx.00.26	05.02.01.02.02.xx.00.27
Input 4	05.02.01.02.02.xx.00.28	05.02.01.02.02.xx.00.29
Input 5	05.02.01.02.02.xx.00.2A	05.02.01.02.02.xx.00.2B
Input 6	05.02.01.02.02.xx.00.2C	05.02.01.02.02.xx.00.2D
Input 7	05.02.01.02.02.xx.00.3E	05.02.01.02.02.xx.00.2F
Output 0	05.02.01.02.02.xx.00.30	05.02.01.02.02.xx.00.31
Output 1	05.02.01.02.02.xx.00.32	05.02.01.02.02.xx.00.33
Output 2	05.02.01.02.02.xx.00.34	05.02.01.02.02.xx.00.35
Output 3	05.02.01.02.02.xx.00.36	05.02.01.02.02.xx.00.37
Output 4	05.02.01.02.02.xx.00.38	05.02.01.02.02.xx.00.39
Output 5	05.02.01.02.02.xx.00.3A	05.02.01.02.02.xx.00.3B
Output 6	05.02.01.02.02.xx.00.3C	05.02.01.02.02.xx.00.3D
Output 7	05.02.01.02.02.xx.00.3E	05.02.01.02.02.xx.00.3F

And so on. You can do this 2000 times before you will run out of event-numbers. However, do not over-use this facility. You should only do so if you are reusing your node for something completely different, or if you are selling it. Since this node's event-numbers can be taught to other nodes, and those nodes can teach that event-numbers to others

