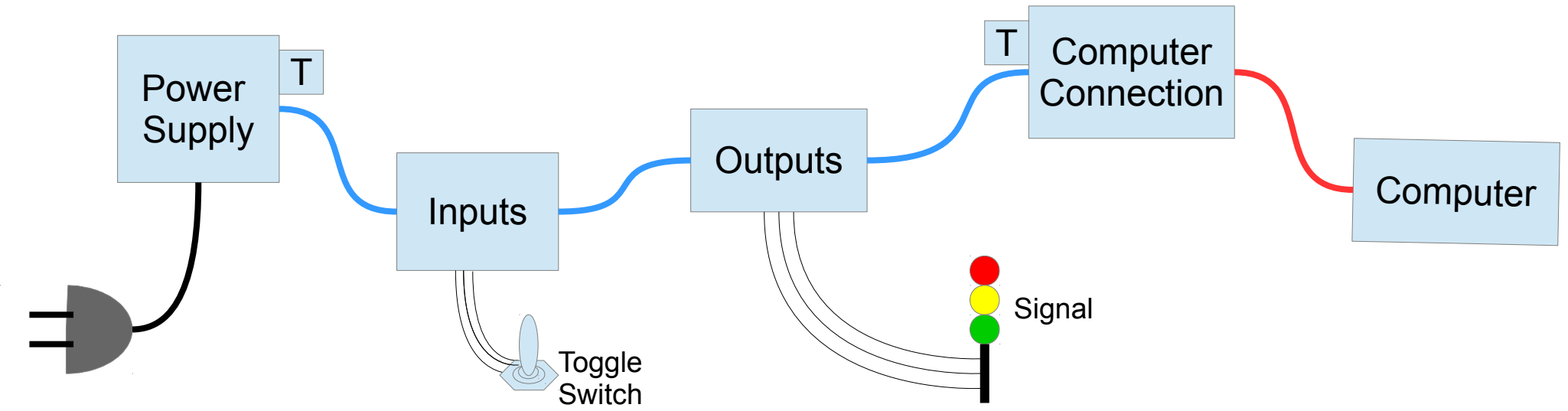
LCC / OpenLCB

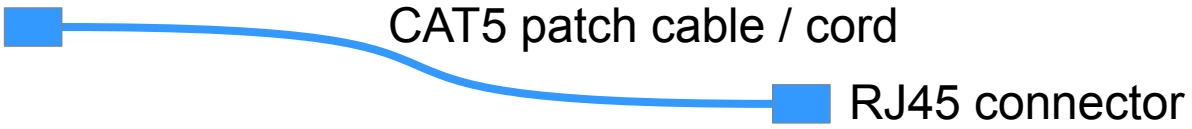
Introduction

Its a bus/network to connect accessories

For example:



CAT5 cables

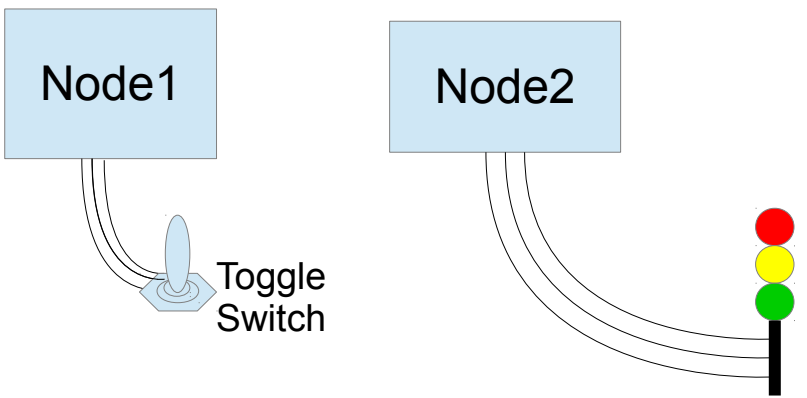


These connect the various network components, Normally one would use standard Ethernet patch cables / cords, available at many stores. You can build your own, if necessary, but the pre-made ones are inexpensive.

Each CAT5 cable contains eight pairs of wires. Some of these are used to carry limited power to the other components. For components need additional power, and they will have connections for it as necessary.

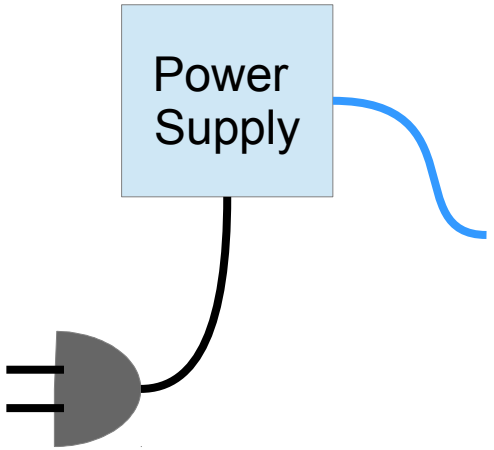
Nodes

These are the components connect to the network, and connect to your MRR accessories.



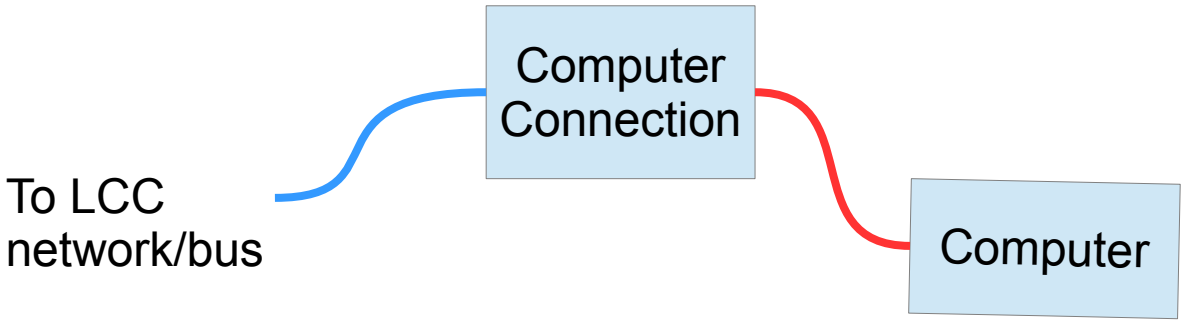
Power supply

LCC carries a limited amount of power in the CAT5 cable. This can be supplied from some Nodes, or from dedicated components, like RRCirkits's Power Point.



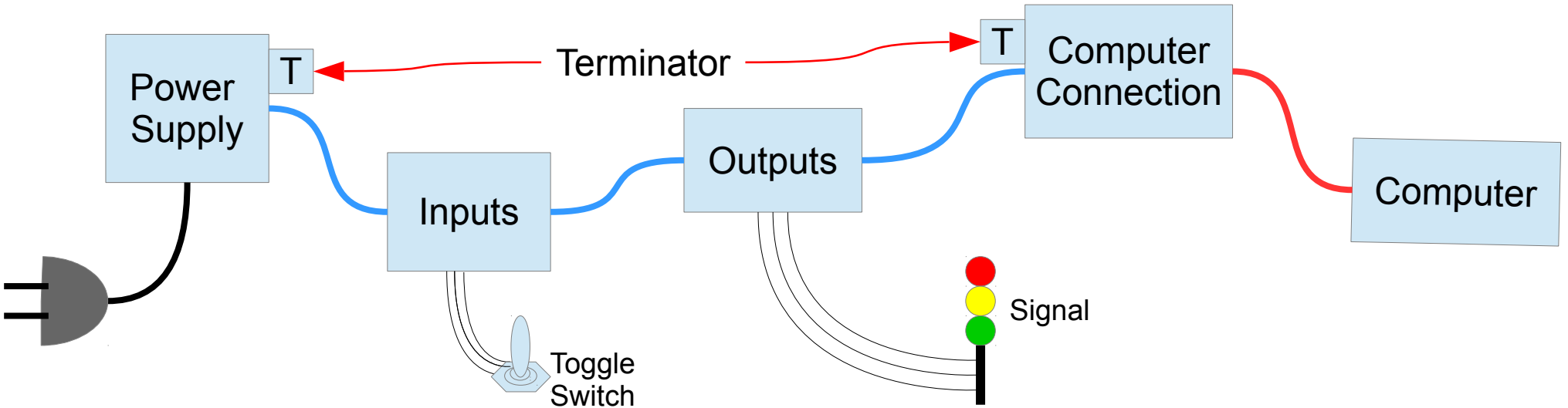
Computer connection

Computer connection: It is most often very useful to have a connection to a computer so that software can be used to configure and control the network.



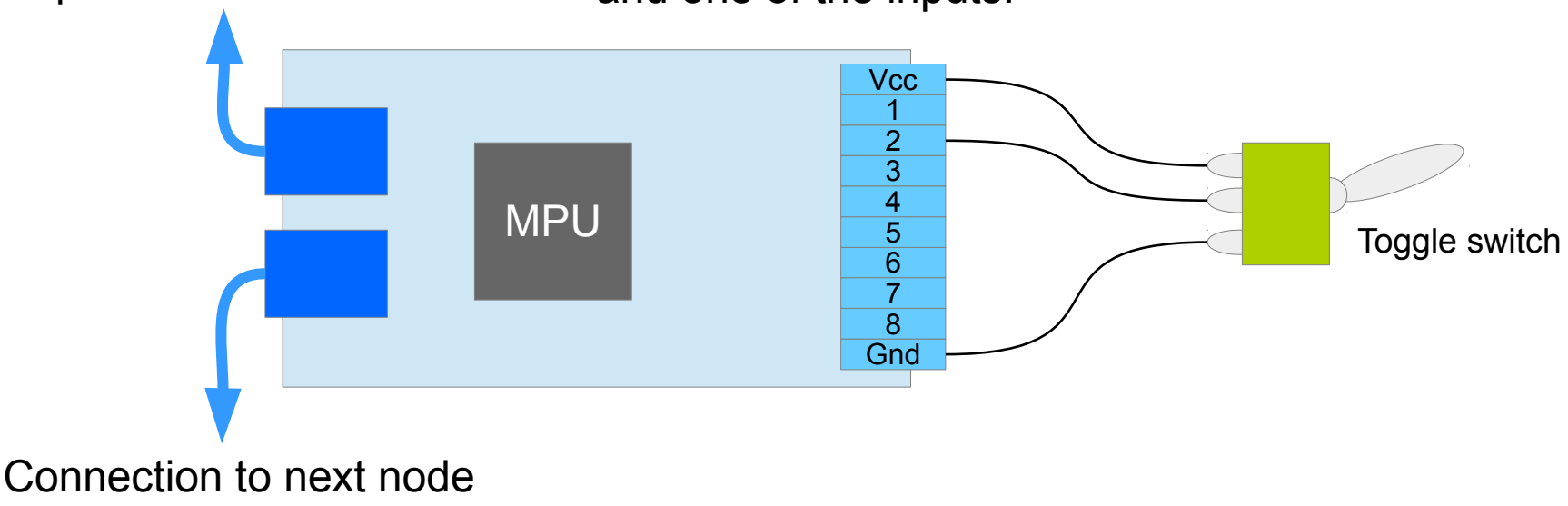
Bus terminators

One requirement of the LCC CAN bus is that it be terminated at both ends. Doing this helps to reduce reflections on the bus that lets it work better. Some nodes will allow the user to activate termination on them. Special terminators can be plugged into the two empty connector RJ45 connectors, eg RRCirkits's terminator.



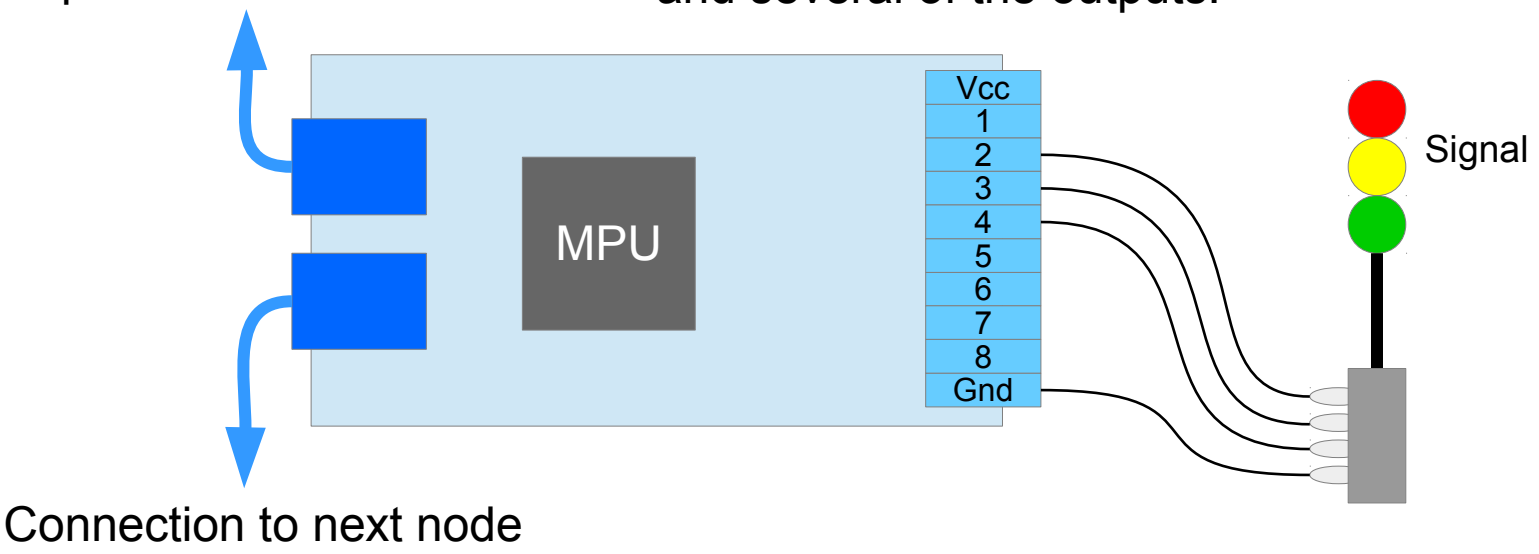
Node components: inputs

Node may have input connectors, whether pins, screw terminals etc. Input devices are connected with wires to these. In this example a toggle switch connects to power and one of the inputs.

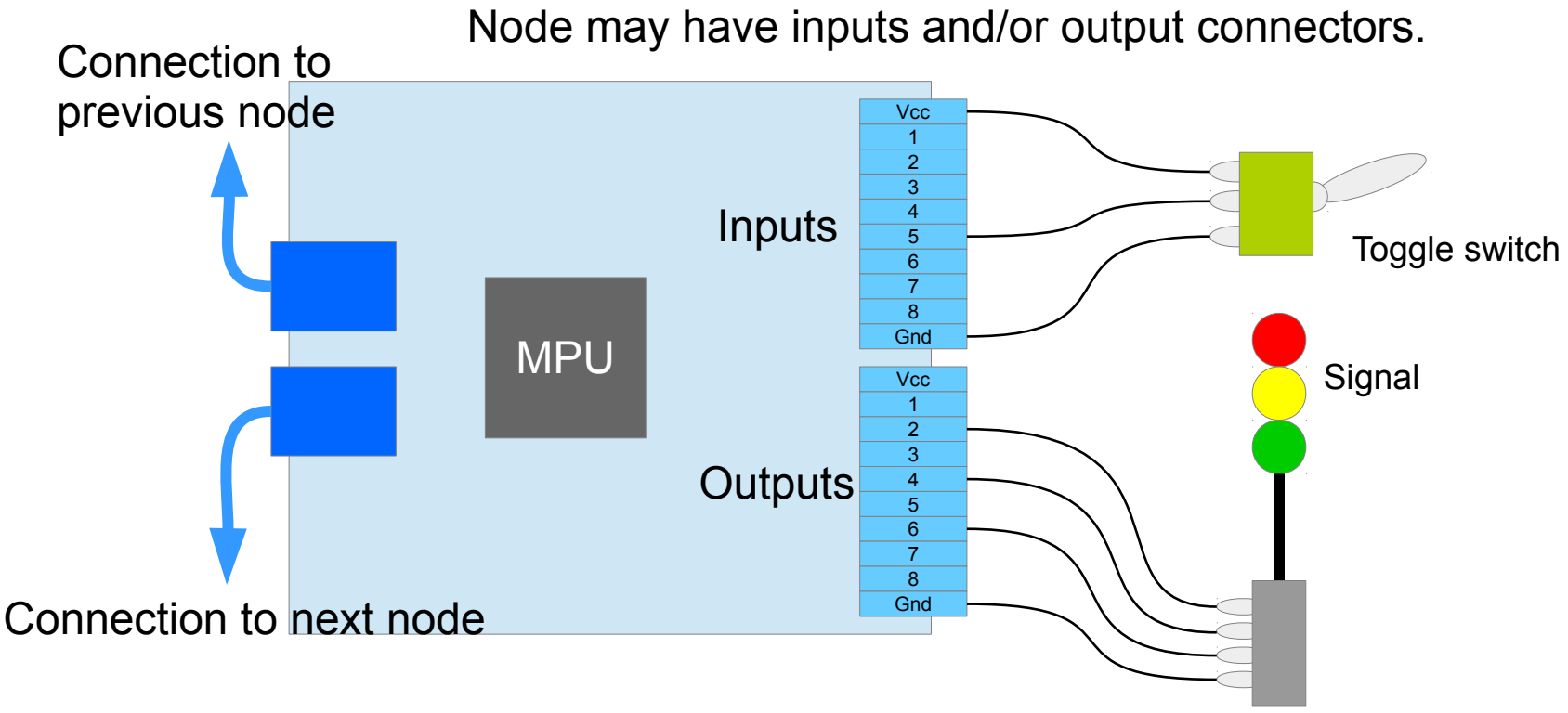


Node components: outputs

Node may have output connectors, whether pins, screw terminals etc. Output devices are connected with wires to these. In this example a Signal mast connects to ground and several of the outputs.

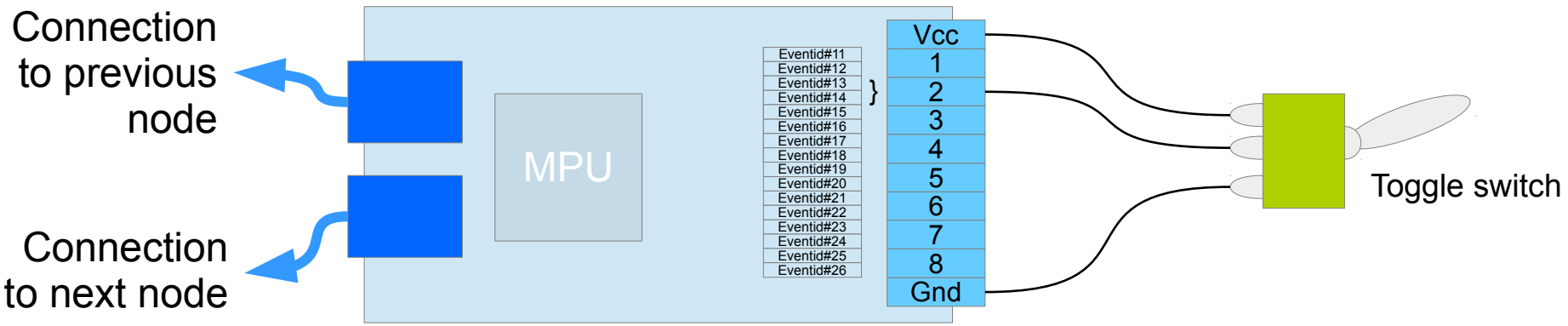


Node components: inputs and/or outputs



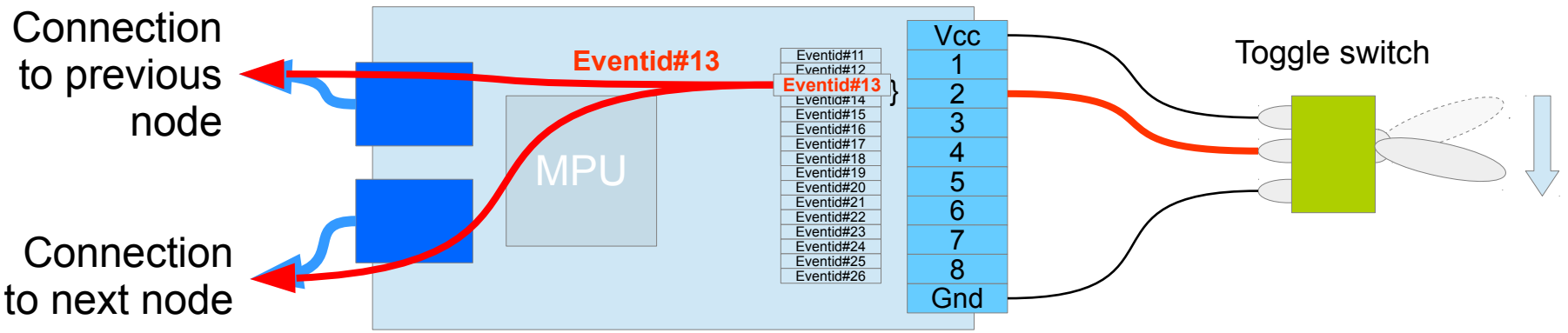
EvenIDs

Eventids are identifiers associated with inputs and outputs. For example, in the input-node below, there is a pair of eventids for each input --- one for when the toggle switch changes to one position, and another when it changes to the other position. In this example, eventids #13 and 14 are associated with inout#1. Eventid#13 is produced and sent to the bus when the toggle is thrown one way, and eventid#14 is produced when it is thrown the other way.



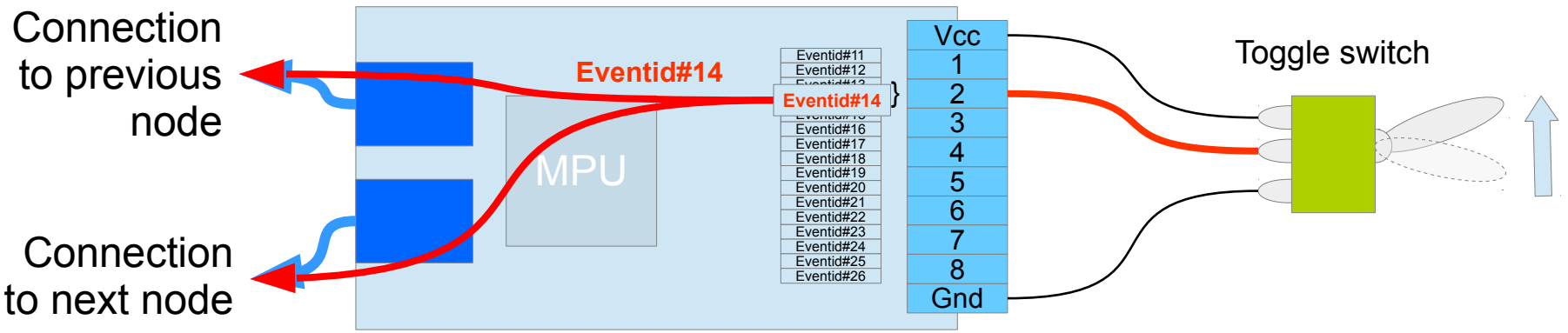
Sending one eventid

Moving the toggle down produces eventid#13, and it is send over the bus.



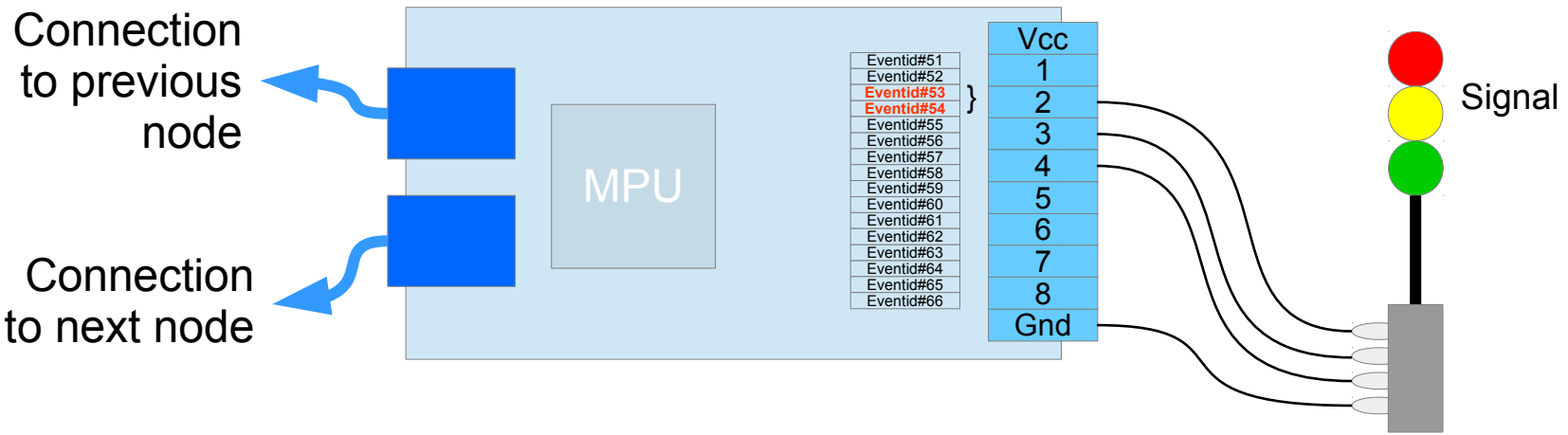
Sending the other eventid

Moving the toggle down produces eventid#14, and it is send over the bus.



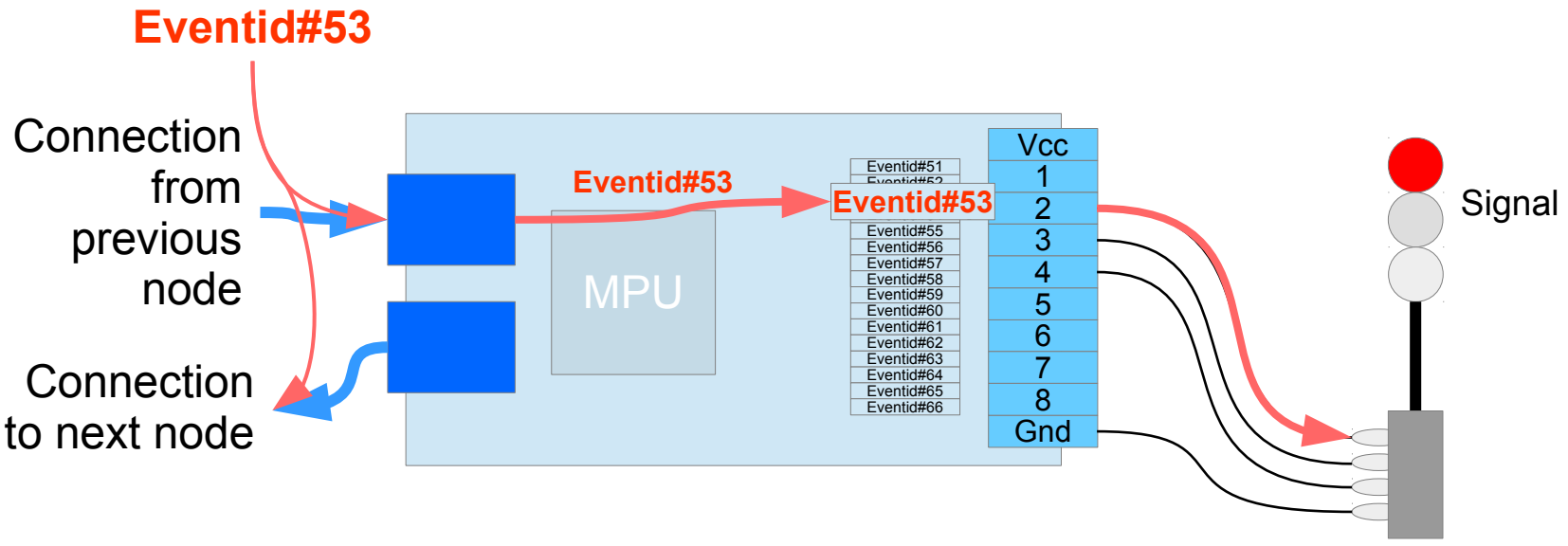
Eventids and outputs

Outputs are also associated with eventids. Often these are in pairs – one turns on a LED and the other turns it off.
Consider the node below. Eventid#53 turns on the red LED and eventid#54 turns it off.



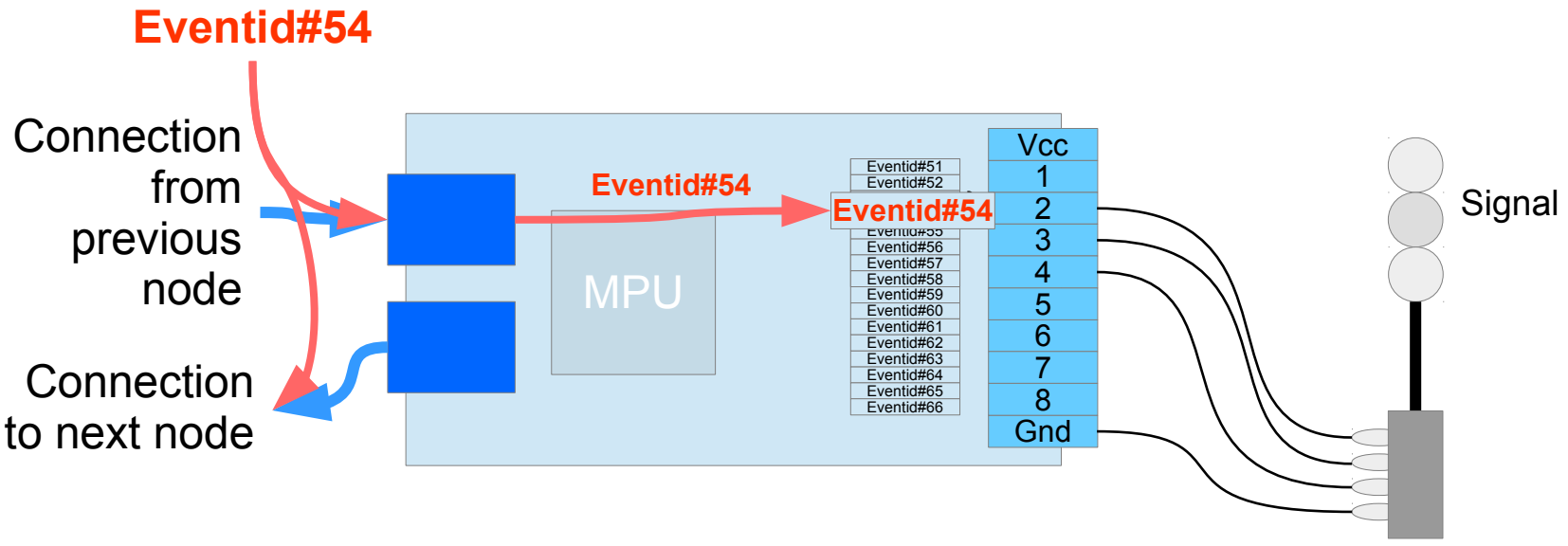
Eventids and outputs

If the node receives (or consumes) eventid#53, it matches the eventids in its internal table, and causes the red aspect to be lit.



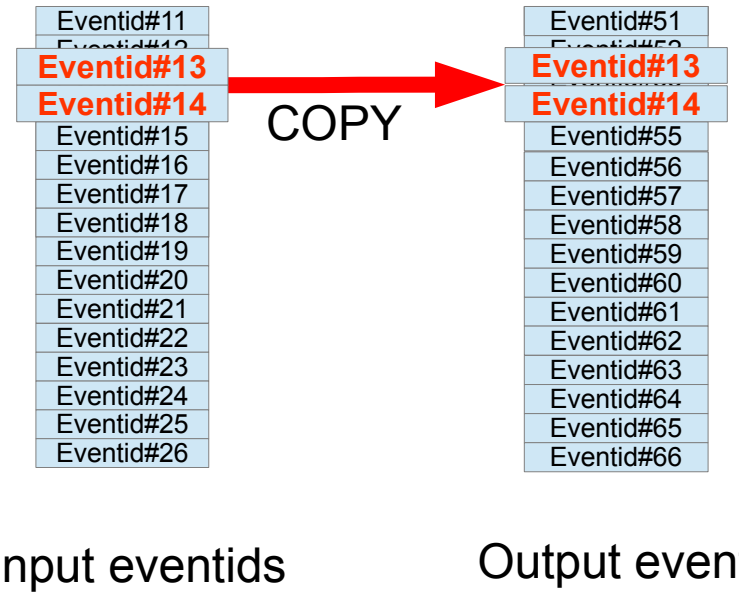
Eventids and outputs

If the node consumes eventid#54, it matches the internal table, and then the red aspect is unlit.



Connecting inputs to outputs

To cause any specific input to affect any specific output, they simply have to share the same eventid. This is done by copying the eventids from one to the other. In this example we have copied eventid#13 and eventid#14 onto the Output events.



Connecting inputs to outputs

Shared eventid