# Configuring a JMRI / CMRI Node

## A Step By Step Guide

**To be used in for testing proper hardware operation of the MaxDuino PCB**

The default Arduino RS-485 test sketch is preconfigured for:

* Speed of 19,200 bps
* CMRI Node Address 0 (with default settings this provides 24 ‘lights’ and ‘48’ sensors)
* The test sketch is listening for individual bits 1-7 coming from JMRI (lights) and then sets or clears the matching bit going to JMRI (the Sensor table)
* It also uses the on board NeoPixel on the MaxDuino PCB to indicate activity.

|  |  |  |
| --- | --- | --- |
| 1 | C:\Users\allom\AppData\Local\Microsoft\Windows\INetCache\Content.Word\00 JMRI-CMRI Connection Preferences.jpg | Make a new connection of Type CMRI. The COM port will be the one your USB/RS485 adaptor is plugged into.  Check additional settings and ensure speed is set to 19200 bps as shown. This must match the Arduino sketch and that speed has been pre configured in the sketch. |
|  |  |  |
| 2 | C:\Users\allom\AppData\Local\Microsoft\Windows\INetCache\Content.Word\01 JMRI-CMRI Configure Nodes.jpg | Notice at the bottom of the previous screen you have the option to configure nodes. If you click it the screen at right will appear (but without any rows). On this screen you will click Add Node at the bottom. |
|  |  |  |
| 3 | C:\Users\allom\AppData\Local\Microsoft\Windows\INetCache\Content.Word\02 JMRI-CMRI Add Nodes.jpg | This pop up then appears. It looks rather busy but most can be ignored. The important item is the node address which defaults to -1 (meaning not configured). Change the address to 0 and click “Enable Polling at Start-up” and then “Add Node”.  After this you should return to the previous screen and now the row for address 0 should be present. |
|  |  |  |
| 4 | C:\Users\allom\AppData\Local\Microsoft\Windows\INetCache\Content.Word\03 JMRI-CMRI Adding Lights To Table.jpg | The previous steps have told JMRI we have a piece of hardware out there at address 0.  Next we will configure individual devices at that address. Specifically go to the Lights table in JMRI and add a light. Use the CMRI connection and click add a sequential range. Start at address 1 and add 7 items |
|  |  |  |
| 5 | C:\Users\allom\AppData\Local\Microsoft\Windows\INetCache\Content.Word\04 JMRI-CMRI Lights Table.jpg | After the previous step the lights table should look like this.  In JMRI a light is really and digital output, it does not matter if it used for a light or not. |
|  |  |  |
| 6 |  | In a similar manner on the sensors table add a block of 7 sensors. |
|  |  |  |
| 7 | C:\Users\allom\AppData\Local\Microsoft\Windows\INetCache\Content.Word\04 JMRI-CMRI Lights Table.jpg | By clicking the state of a given light you toggle it from off to on and vice versa.  That command goes out on the RS485, is detected by the Arduino running the CMRI emulation library at address 0.  The sketch acts on that bit by turning on the NeoPixel and also turn on the matching bit in the sensor table. |
| 8 | If toggling a bit in the lights table causes the matching bit in the sensor table to toggle you can be assured a command is going to the Arduino and being returned back.  This is proof that basic communications is working. While this happening activity should be visible on the USB interface which normally has LED’s flickering to indicate traffic.  Adding additional nodes at additional addresses is simply repeating these steps and modifying the sketch to be listening on the new address. Dozens of nodes and hundreds of lights or sensors is possible. |  |
|  |  |  |
|  |  |  |