

Test No	Area under test	Test Code	Physical Parts Needed	Connections	Steps	Expected Outcome	Comments
1	IO Header Pins	MaxDuino_IO_Check	A 3 wire sensor with 3 pin dupont header in order S-V-G	plug in sensor observing pin order	By activating a one pin at a time the on board neopixel should change colour.  <b>Do not ever jumper the +5 and ground pins together - smoke will result.</b>	The NeoPixel will display a different colour for each pin activated. A6/A7 and D4/D13 are not tested.	Most of the pins are configured as digital inputs by the sketch. A6/A7 do not have this capability. The purpose is only to test if the header pins are working.
2	Neopixel	MaxDuino_Onboard_NeoBlink.ino	none	none	Load Sketch	on board Neopixel will cycle Red-Green-Blue	NeoPixel is on pin D4
3	I2C	Maxduino_I2C_check_OLED_128x32	128x32 OLED display F-F Dupont jumper wire (Qty 4)	Connect display to I2C header, matching the 4 pins	Load Sketch	Display will cycle through some tests under control of the sketch.	I2C OLED displays have an I2C address of 0x3C or 0x3D
4	MP3	MaxDuino_MP3_Sound_Check.ino	Small 8 ohm speaker micro SD card with mp3 folder containing mp3 sound tracks 3 wire sensor (eg IR sensor) to activate sound effects.	speaker connected to the speaker terminals 3 wire sensor on D9 (ensure correct Pin Out)	Load Sketch	A sound track should play upon power up. Sensor activation will advance to next track.	play order is determined by time stamp of file copy, NOT the file name or visible file time stamp.
5	RS-485	MaxDuino_RS485_MP3	SD card with MP3 sound tracks USB to RS485 converter. Pair of wires for RS485 USB cable for Arduino speaker and wires Computer running JMRI	Connect speaker wires to the speaker terminals. Use another pair of wires to connect the RS485 USB converter to the module (matching the A and B terminals). Connect the USB converter to the PC	Establish a connection from JMRI to USB converter so that becomes available to JMRI. (See Screen shot at right) The CMRI node in Arduino test code is listening on address 0. Therefore in the lights table (1) define 'lights' for CMRI addresses 1 => 7 (2) Toggling any of bits 2-7 will trigger sound effects. (3)	Bits 1 to 7 being toggled in JMRI/CMRI will cause sounds to play on module. (Only 7 sample sound effects are on SD card, As written the Arduino code is only looking at first 7 CMRI bits)	This test shows a minimum of functionality as a proof of operation.
6	DCC Decoder	MaxDuino_DCC_Decoder	Jumper wires to power the MaxDuino from DCC	DCC Power to the power input connections on MaxDuino. Note: The use of a heat sink on the 5 Volt regulator (Part 7805) is highly recommended. Depending on the DCC voltage used this chip can get warm.	Load the test sketch Connect MaxDuino to a DCC source. (Leave unpowered). Initially nothing is being decoded as MaxDuino is unpowered. Apply power to the module from a DCC supply. Decoding will commence as soon as Arduino powers up.	When power is applied to the DCC the Arduino is also powered up. It will decode the DCC packets and display some summary information (counts of distinct packets) on the serial monitor. (at 19.2Kbps)	This test shows a bare minimum of functionality as a proof of hardware operation.

Test #5 - JMRI showing a CMRI connection to the USB / RS485 adaptor



Test #5 - In the lights table (1)

Add a series of lights at addresses 1 through 7 (2)

Toggle the bits using the buttons (3).

Several buttons have sounds associated with them assuming the SD card is installed.

