MaxDuino R1.5
Design Walkthrough

Note: In some cases photos of a previous revision are used where there is no material difference to the main message.

MaxDuino Summary

MaxDuino is a carrier board for an Arduino Nano.
Components on the carrier board add significant functionality to the basic NANO

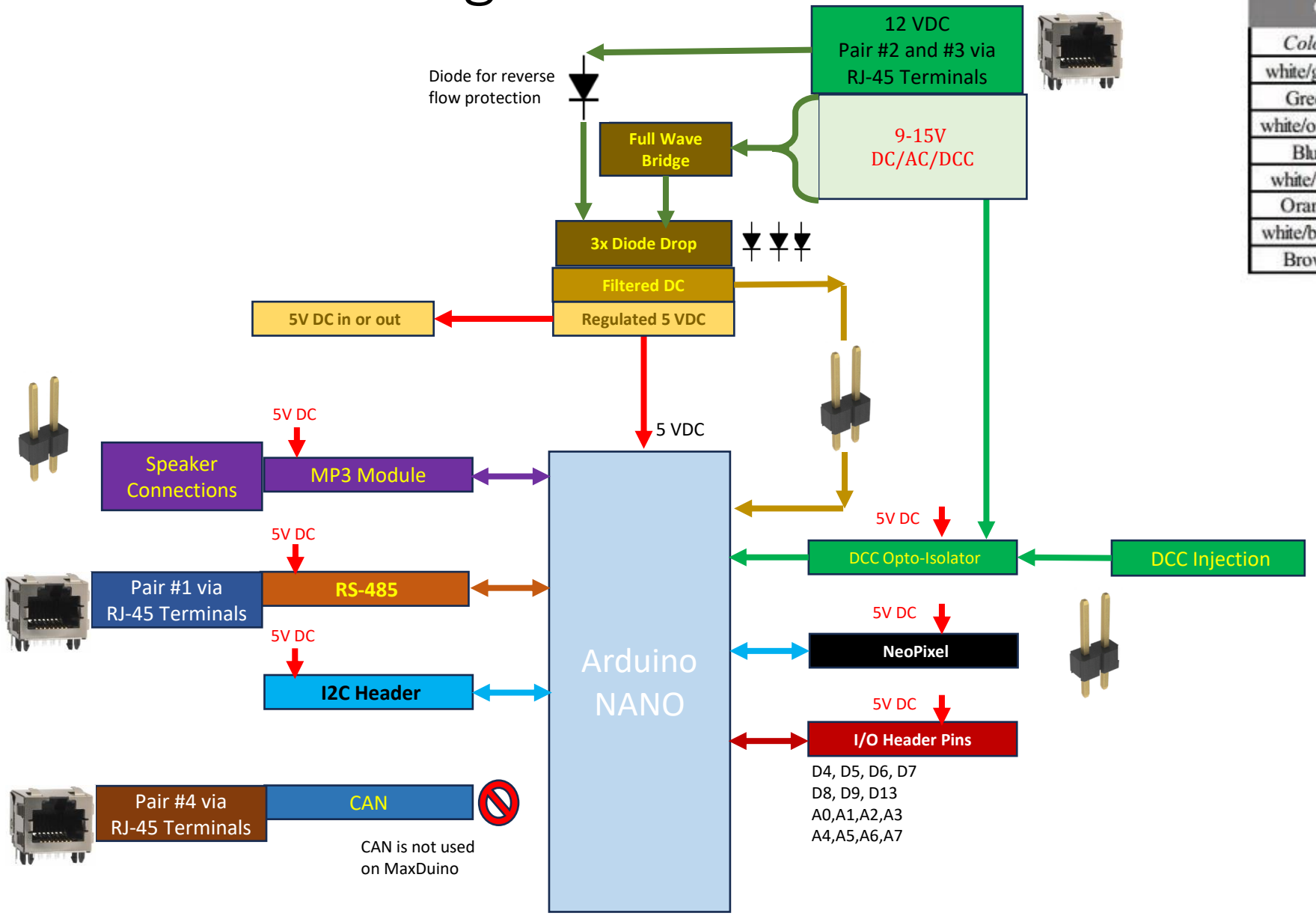
- Versatile power supply
- 3 pin I/O headers
- I2C header
- DCC decoding
- MP3 sound player
- RS485 communications
- NeoPixel (smart LED)

Version 1.5 replaces RS485 screw terminals with RJ45 jacks. This makes connecting multiple MaxDuinos together as easy as plugging in a standard ethernet cable. The one cable distributes power to the MaxDuino plus RS485 communications.

For full version history see the Management of Change pages.

** Be Aware **
As with any Arduino project be careful you do not set up an inadvertent ground loop. For example a USB cable plugged into an Arduino and a second USB cable plugged into a Arduino based DCC controller. When the DCC power is used to power the MaxDuino a ground loop is possible. (The use of one or more USB isolators is recommended)

MaxDuino 1.5 Block Diagram



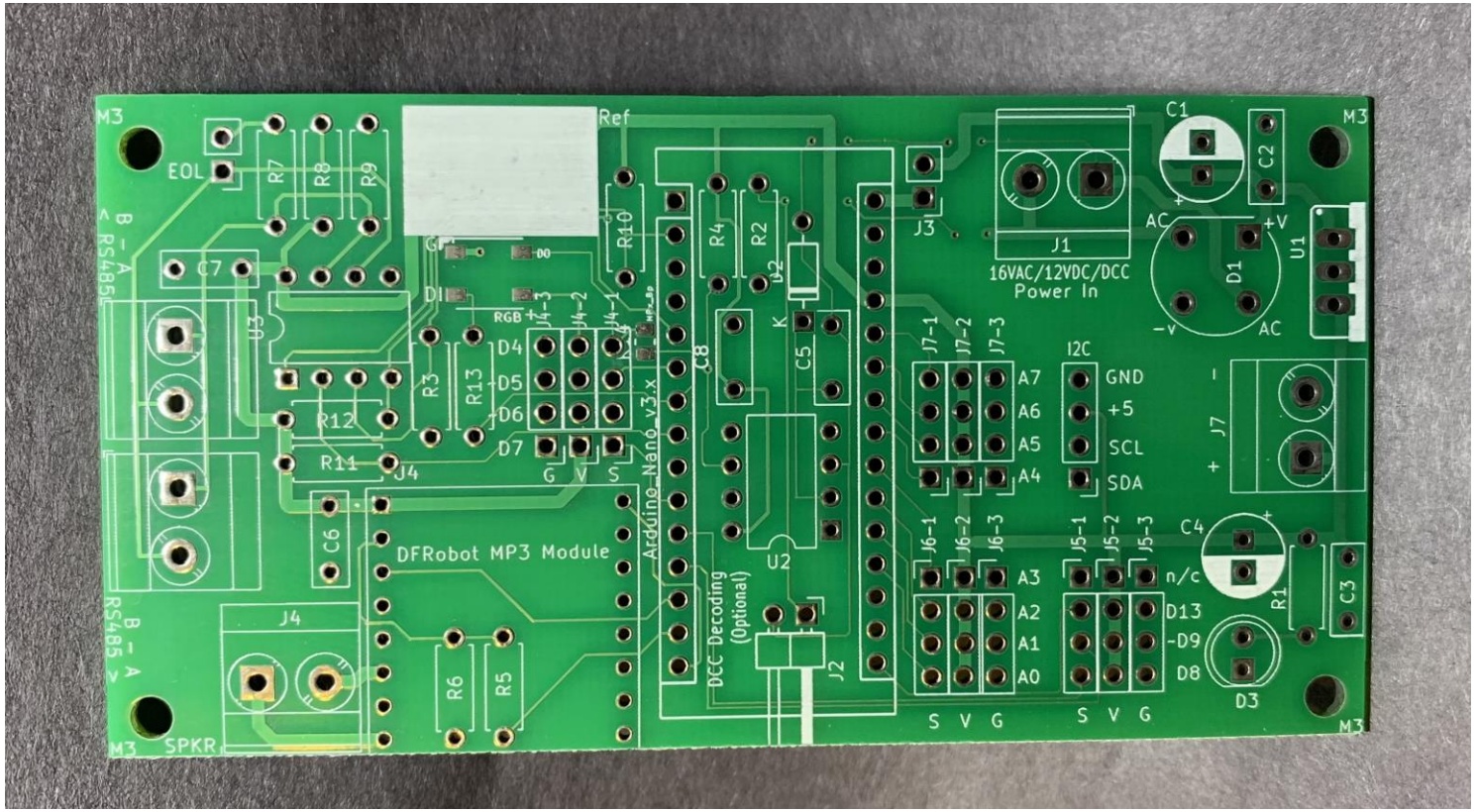
Straight LAN cable			
Color		RJ45	
Color	*Pair*	*Pin*	*Function*
white/green	3	1	Power 12+
Green	3	2	Power 0-
white/orange	2	3	Power 12+
Blue	1	4	RS485 A
white/blue	1	5	RS485 B
Orange	2	6	Power 0-
white/brown	4	7	CAN H
Brown	4	8	CAN L

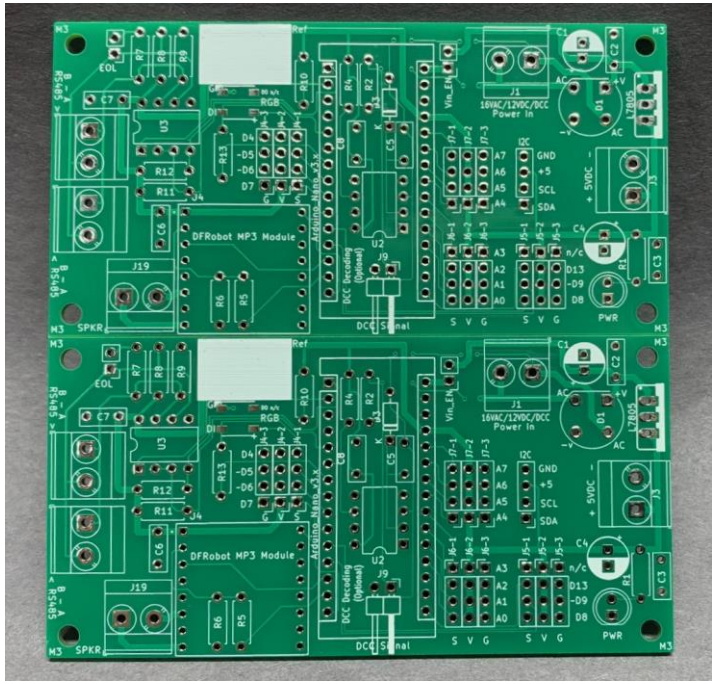
MaxDuino PCB

(Typical General arrangement)

One MaxDuino.

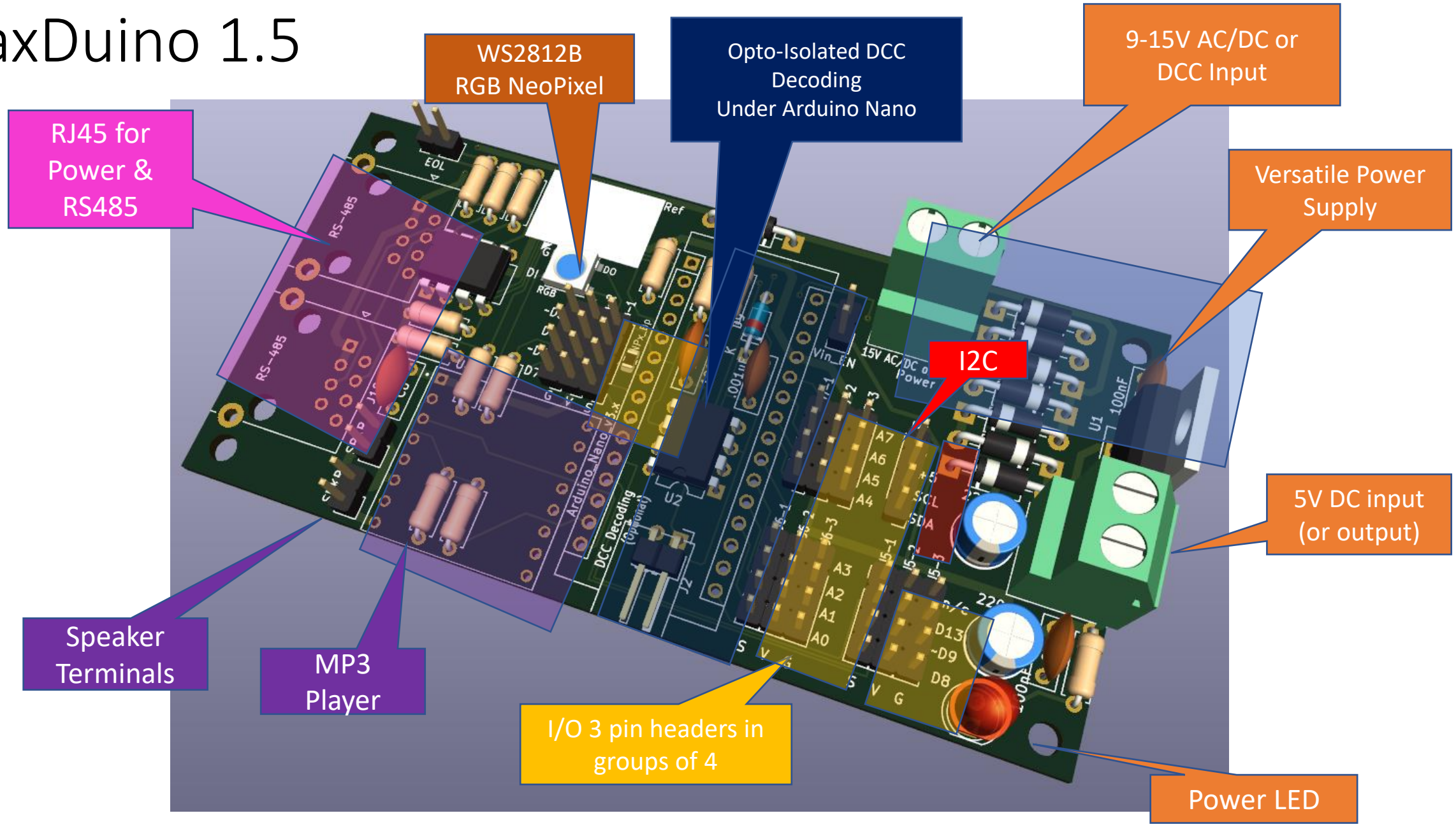
Vee cuts in the PCB allow clean separation with minimal force.





Two panels are on one PCB
under 100mm x 100mm

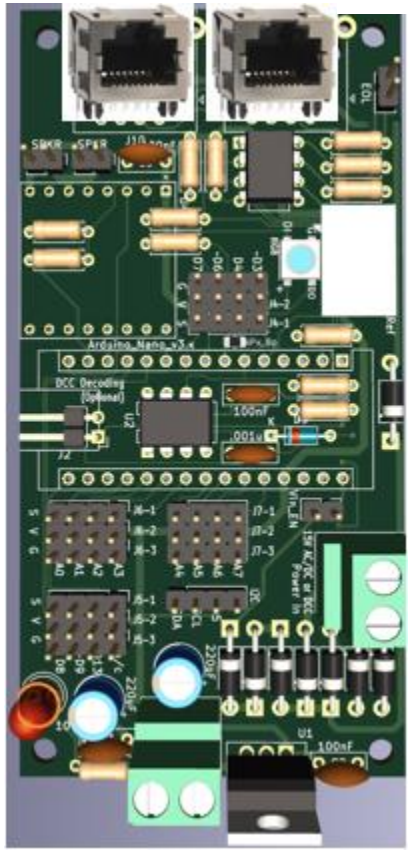
MaxDuino 1.5

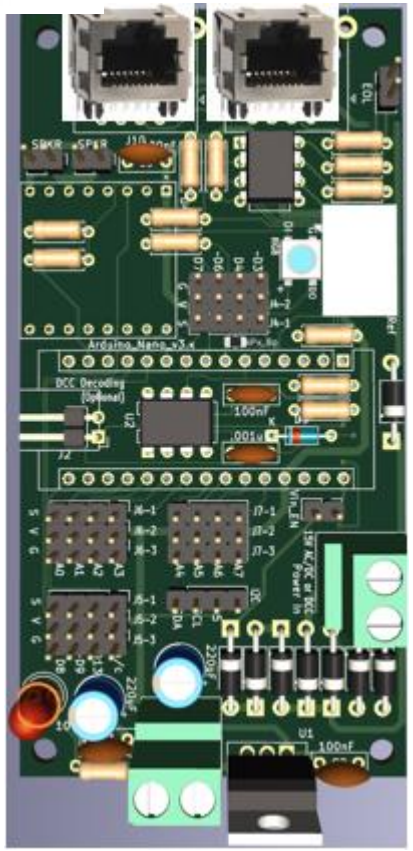

WS2812B
RGB NeoPixel
Opto-Isolated DCC
Decoding
Under Arduino Nano
9-15V AC/DC or
DCC Input
RJ45 for
Power &
RS485
Versatile Power
Supply
I2C
5V DC input
(or output)
Speaker
Terminals
MP3
Player
I/O 3 pin headers in
groups of 4
Power LED

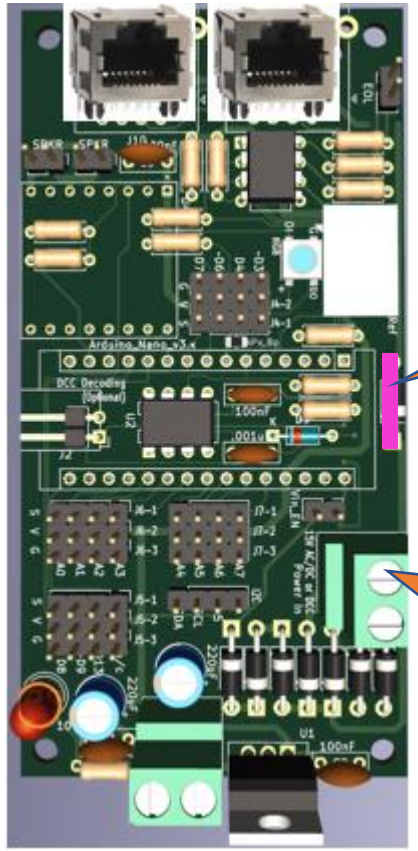
MaxDuino General Architecture

RJ45 cable distributes both Power & RS485

USB from JMRI hops onto network cable (2 wires)



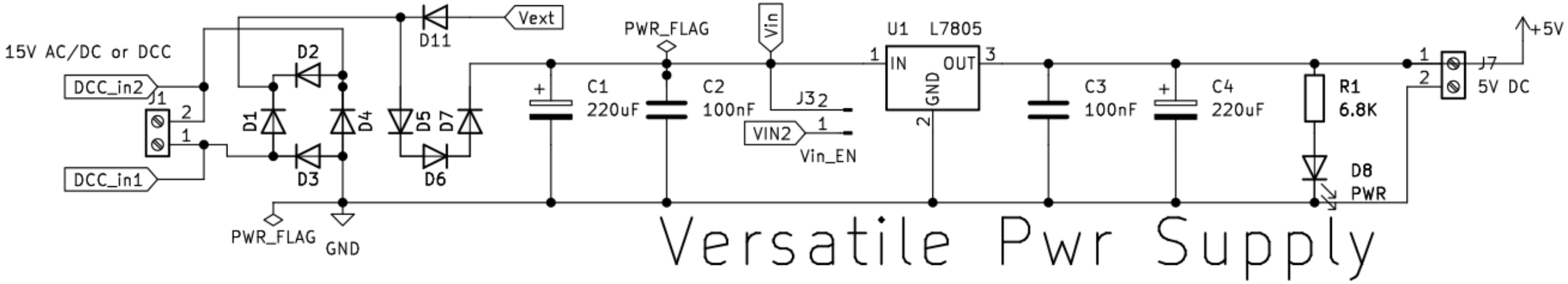




On one MaxDuino install a jumper instead of this diode. This allows power ‘out’ onto the network cable.

Power provided here is then supplied to all MaxDuinos.

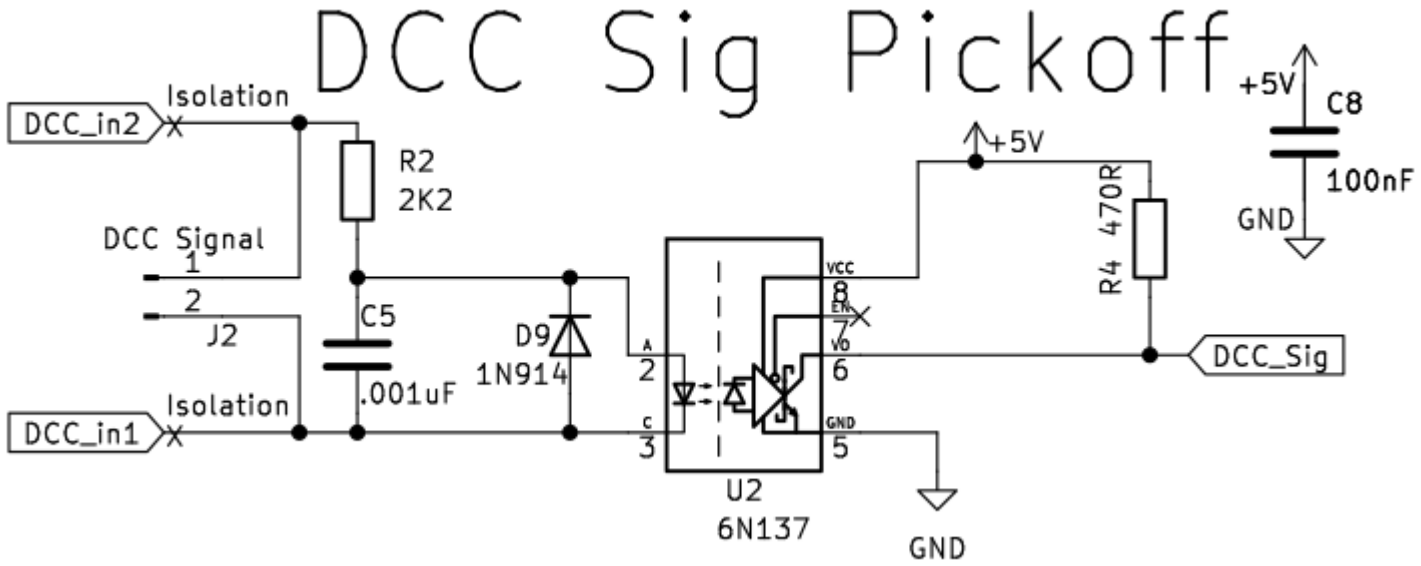
Design Review - Power Supply



Power comes in from the left. It can be 9-15V AC/DC, or a DCC feed. It then goes through a full wave bridge rectifier D1-D4 implemented as discrete diodes for flexibility. There are 3 additional diodes in series D5,D6 and D7 that drop the supplied voltage to the rest of the circuit. The filtered and reduced DC voltage is input to the 5 volt regulator chip L7805. External power from the RJ45 jack enters via D11. Current will only flow if the external voltage exceeds any locally supplied voltage. The 5 volt regulator is capable of providing 1.5 amps (the preceding diodes may be limiting at less than this). The 5 volt output is filtered with C3 and C4 to produce a steady 5 volt supply. Note that C4 is needed to ensure stability of the voltage regulator. A LED is used to indicate the presence of 5 volts (even without any Arduino present.) On the right hand side is a second set of terminals which can be used as a 5 volt output to supply other circuits, or this can be used as a 5 volt input (in which case the L7805 regulator and everything to the left of it is not required.

Notes: If DCC is used as a power source the use of a heat sink on the L7805 is strongly recommended. According to your preference high speed diodes may be used for D1-D4 (Usually at the expense of power handling). If DCC is used as the power source that signal is by default routed to the DCC optoisolator circuit. The input to the voltage regulator chip Vin can be routed to the Vin pin of the Arduino (labelled Vin2) by inserting a jumper at Vin_En. (You must verify Vin is 12VDC or less else the Arduinos on board regulator could be damaged).

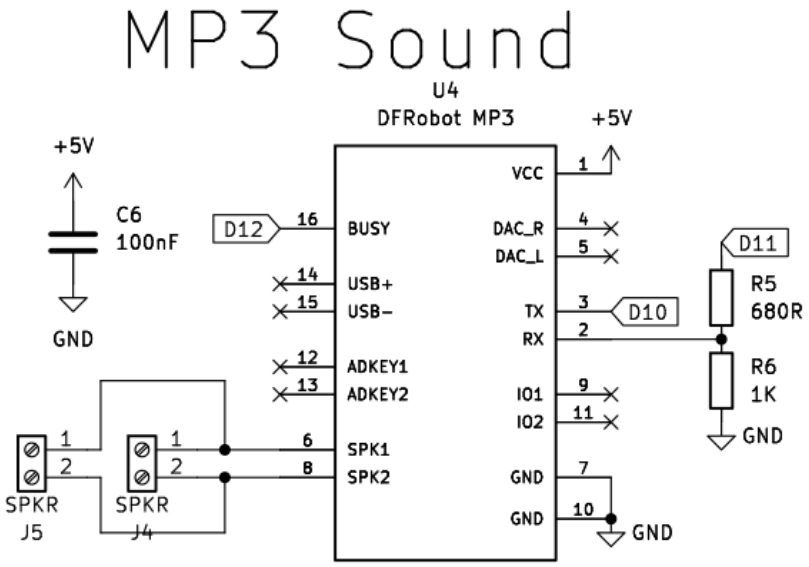
Design Review – DCC Decoder



The DCC signal from the power circuit comes in on the left as DCC_in1 and DCC_in2. These are routed through two isolation points which are thin traces and are clearly marked on the bottom of the MaxDuino PCB.
The DCC signal goes onto the 2K2 current limiting resistor R2 and then through the optoisolator U2. D9 provides reverse polarity protection for the optoisolator diode. C5 (in conjunction with R2) provide transient noise rejection. Pin 7 is not used as the optoisolator has an internal pull-up. The output of the optoisolator goes to Arduino pin D3 and is pulled high by R4 (470 ohm) which is needed to create a fast risetime on the output. The use of pin D3 is very important because this is only one of two pins on the Nano that can have interrupts attached which is important for the needed time critical decoding.

NOTES: If DCC is not used as the MaxDuino power supply but you still want to decode dcc signals then the two DCC isolation traces must be made open circuit. Once these traces are opened then the MaxDuino power source and the DCC decoding are isolated from each other. An independent DCC signal for decoding can be input via J2 which is a pair of horizontal pins located on the PCB under the Arduino.

Design Review – MP3 Player



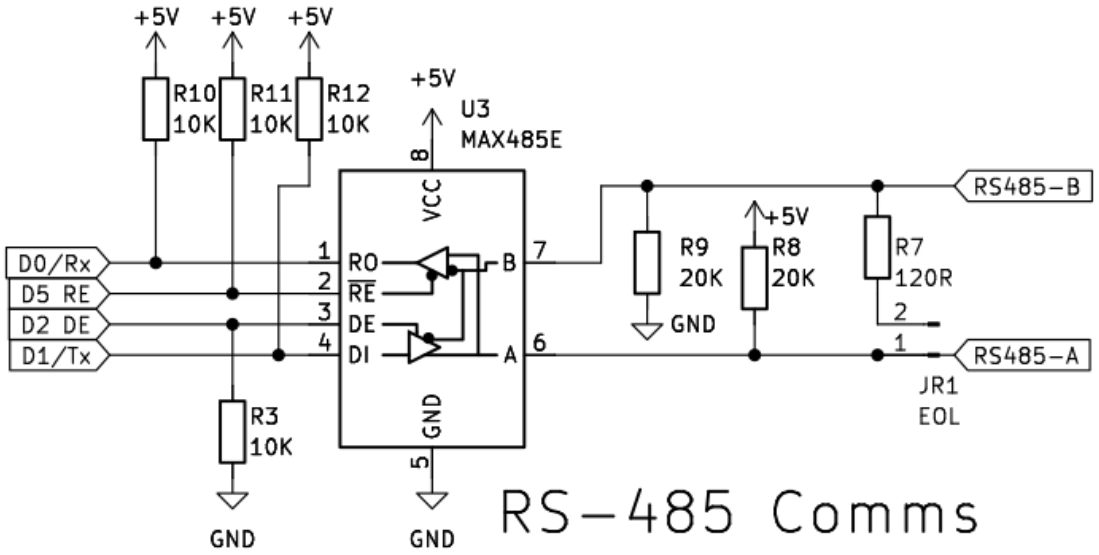
This circuit utilizes the common DFRobot MP3 player module. This module has an SD card reader and so it can play MP3's located on a memory card. These can be sound effects, station announcements, or any other sounds as required. Any small 8 Ohm speaker rated at 3 watts or more can be used.

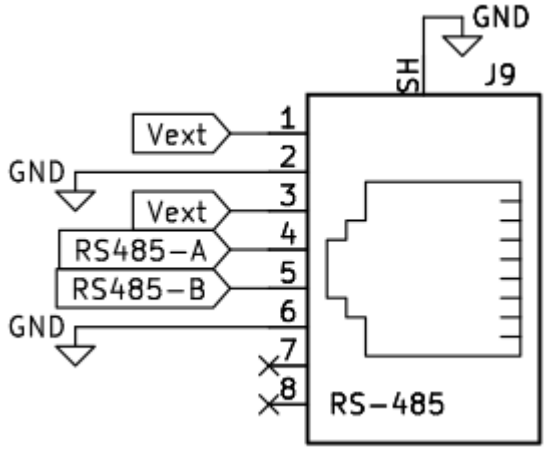
The Tx pin from this module connects to D10 on the Arduino while the Rx pin is connected to D11. The use of software serial is required to implement this but as the transmission speed is a relatively slow 9600 bps this is not an issue. A pair of resistors are used to lower the voltage received by this module as it can glitch if the Rx voltage is even slightly over its 5 volt limit.
(Alternate design which I have not tested – omit R6, use 1K resistor for R5)

The busy pin is taken back to the Arduino on pin D12 to provide real time status information.
It is possible to 'ask' the module for its status via software also. Both approaches have been tested and work as expected.
The following library has been tested `#include <DFPlayerMini_Fast.h>`

Note: The **DFRobot MP3** player module is recommended. There is a similar module with part number MP3-TF-16P but it did not work well for me. Specifically It did not respond to some serial commands. (e.g.: setting volume)

Design Review



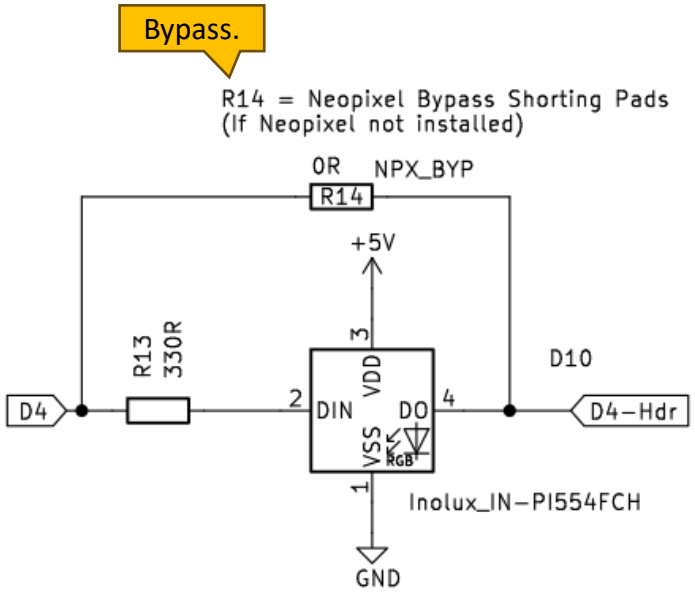


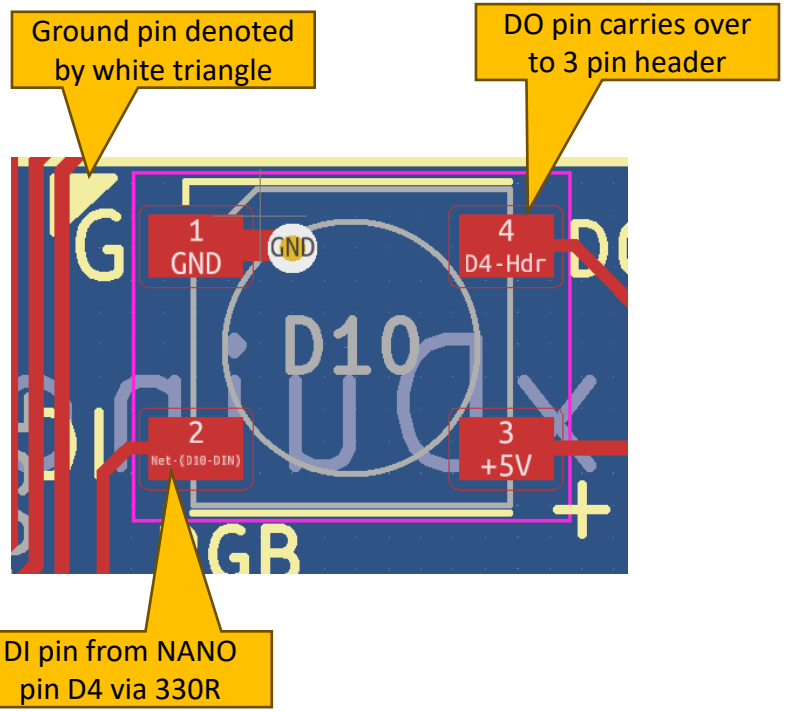
Typical of two on board.

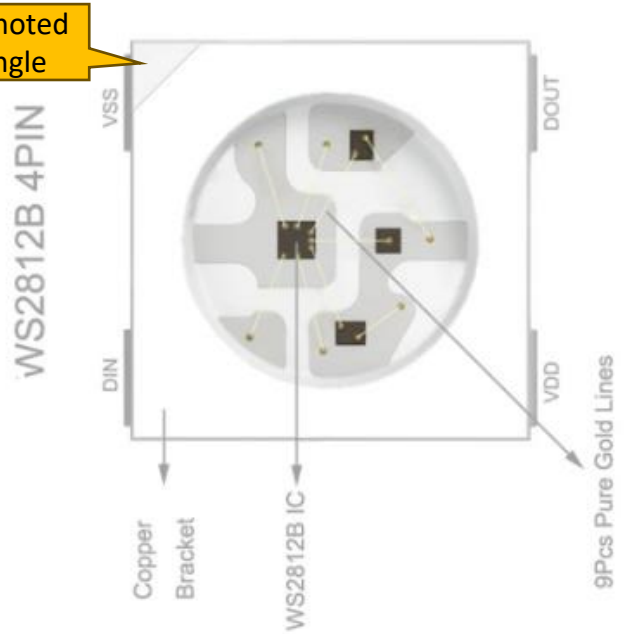
This circuit utilizes the common MAX485 chip to receive RS485 communications coming in from the right and then converts them into standard logic level signals that the Arduino can read. As of MaxDuino V1.5 there are two RJ45 jacks in parallel allowing for easy daisy chaining of both comms and power. R7 is an EOL (End of Line) termination resistor. The EOL jumper at R7 should only be installed on one of the MaxDuino boards and preferably the one farthest from the sending device (Typically a USB/RS485 interface). This circuit interfaces with the Arduino Tx/Rx pins to take advantage of the high speed Arduino UART. Note the direction of transmission (DE and RE) are implemented on separate Arduino pins. Doing this and using pull up and pull down resistors R3 and R11 means that in circuit serial programming can work without disrupting other RS485 communications on the wires. (In the instance of a large network for example).
The following libraries were used in testing

```
#include <Auto485.h>
#include <CMRI.h>
```

Design Review – NeoPixel







Pinout for WS2812B NeoPixel chip.
Note **Vss = Ground**. **VDD** = **+ 5V**
The orientation shown matches the PCB

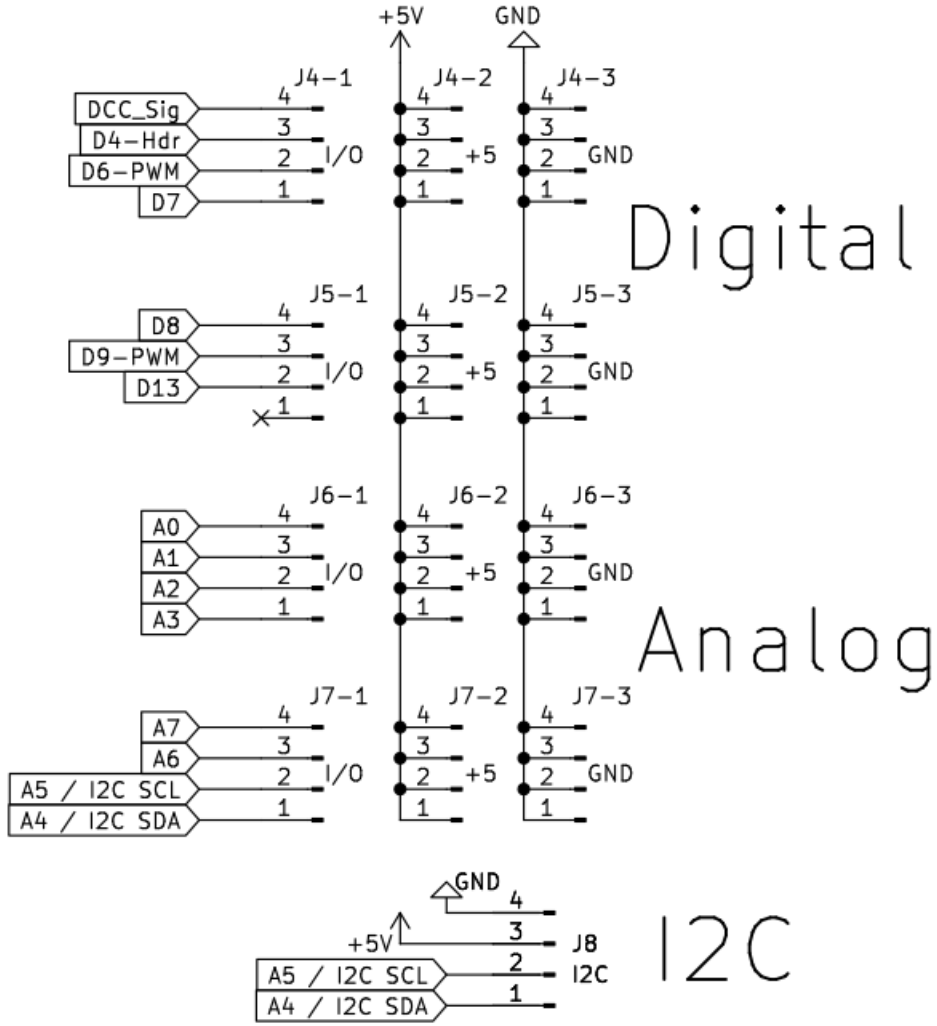
This simple circuit utilizes the common addressable LED WS2812B NeoPixel chip to provide a wide range of colour feedback options. The circuit utilizes one Arduino pin (D4) and has an on board recommended 330 ohm series resistor for the input pin. The three pin header associated with D4 is also available and suitable for longer strings of NeoPixel addressable LED's. Be sure to consider power requirements when using long (more than 10 elements) strings of NeoPixels. Note a bypass set of pads exist (R14 is specified as a SMD component but is not normally installed – only the solder pads are used and these can shorted if the NeoPixel is not installed. Doing so connects D4 directly to the header pin bypassing the onboard Neo Pixel. If the header pin D4 is used for additional NeoPixels the onboard NeoPixel is index 0, and the external NeoPixels start at index 1. The following library was used in testing `#include <FastLED.h>.`

Design Review – Header Pins

The I/O pins are arranged as four groups of four spaced to enables colour coding the pins. Each of the I/O pins is this presented as a 3 pin header.
The pin order is Signal, +5V, Ground or S V G for short.
This pin order is a common but not universal
(so always check your modules)

On the PCB some pins are marked with a tilde (~).
This is indicating these pins are PWM enabled.

The DCC decoder circuit ties into **pin D3**.
Pin D4 is normally the output of the on-board NeoPixel but from their it returns to the header pins (for connecting additional external neopixels).
A4/A5 pins are also part of the I2C header.
The I2C header provides +5, Ground, SCL and SDA
(In the same order as commonly found on OLED displays.)
Be sure to check your pin order when using I2C.



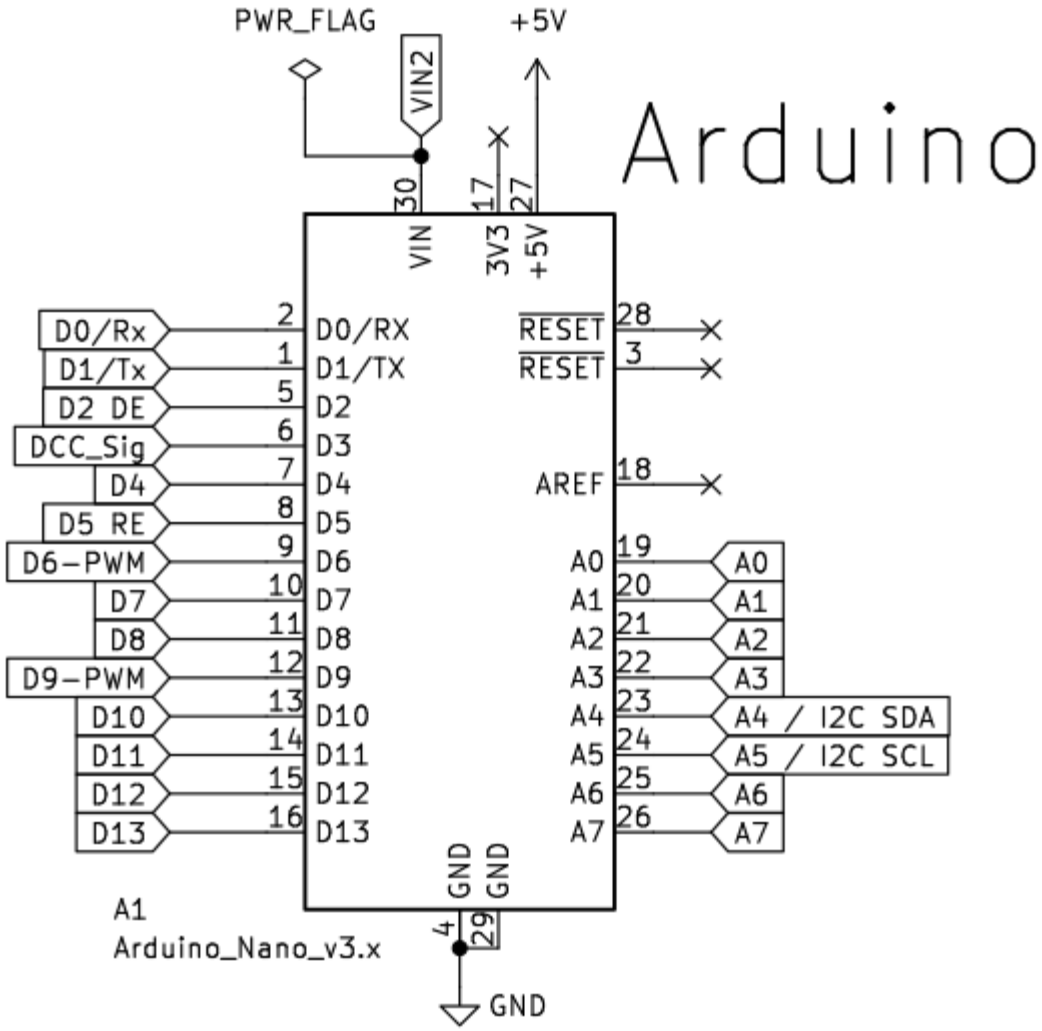
Design Review – Arduino

The Arduino NANO module is almost fully utilized as shown. Only 4 pins have no connections to them.

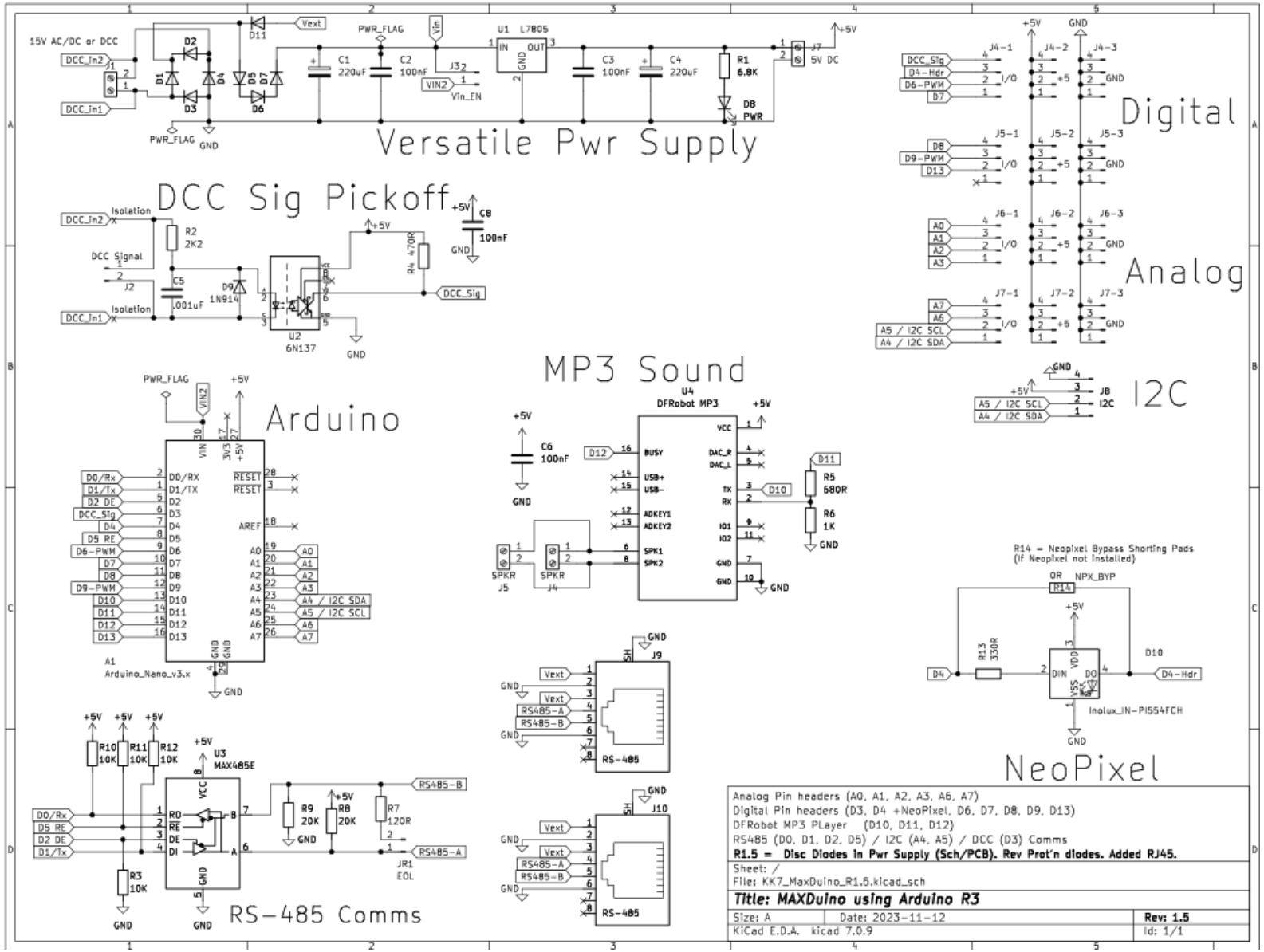
As of R1.3 both D2 and D5 are used for RS485 direction control.

It is recommended to install female header pins on the PCB so that the NANO module can be simply plugged into those header pins. This also provides ample clearance for the DCC decoder circuit located under the Arduino.

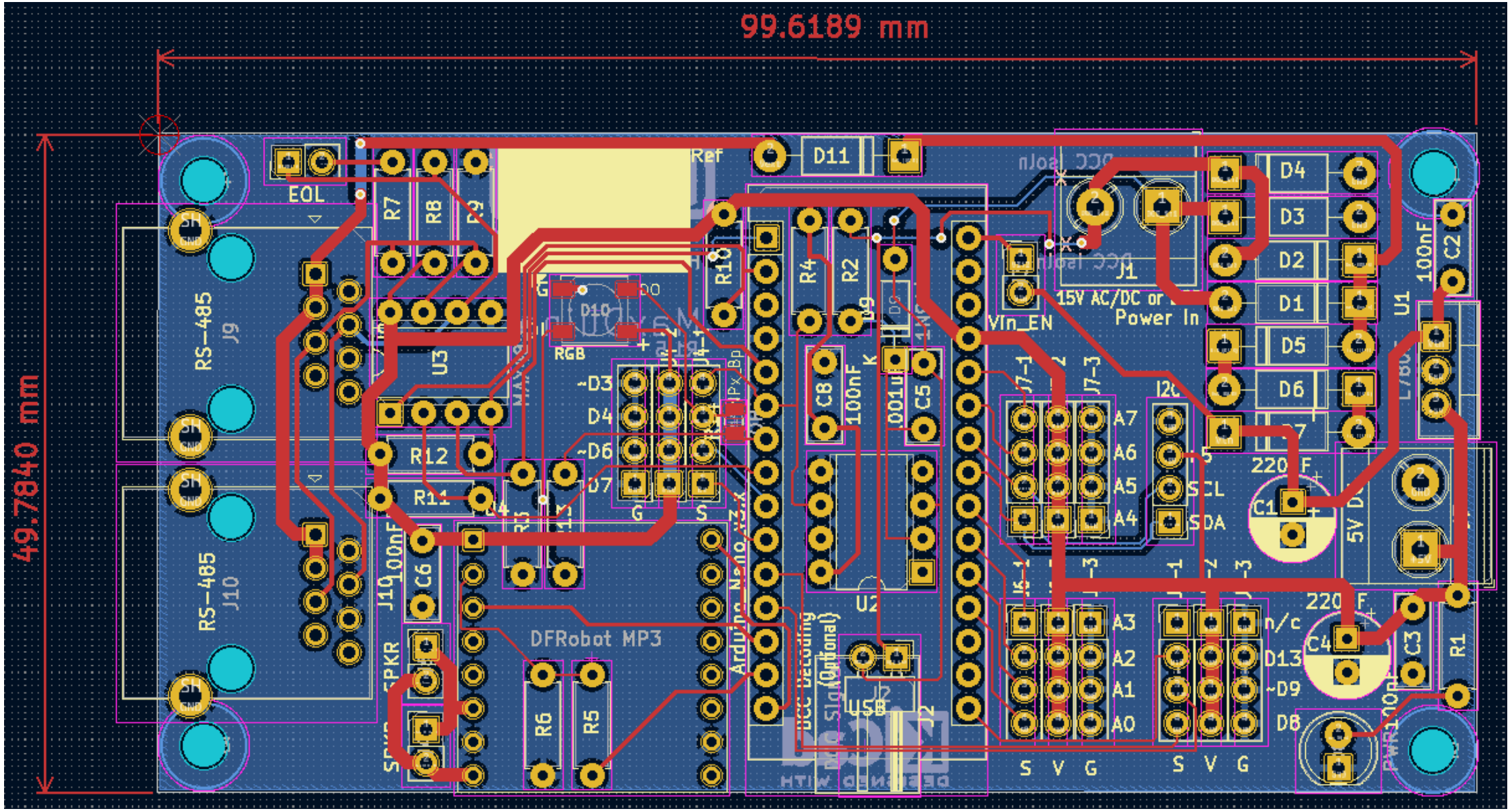
For ease of testing I utilized separate NANOs preprogrammed with the test sketches. (labelled appropriately). This eliminated the need for re-flashing them all the time.



Design Review Full Schematic



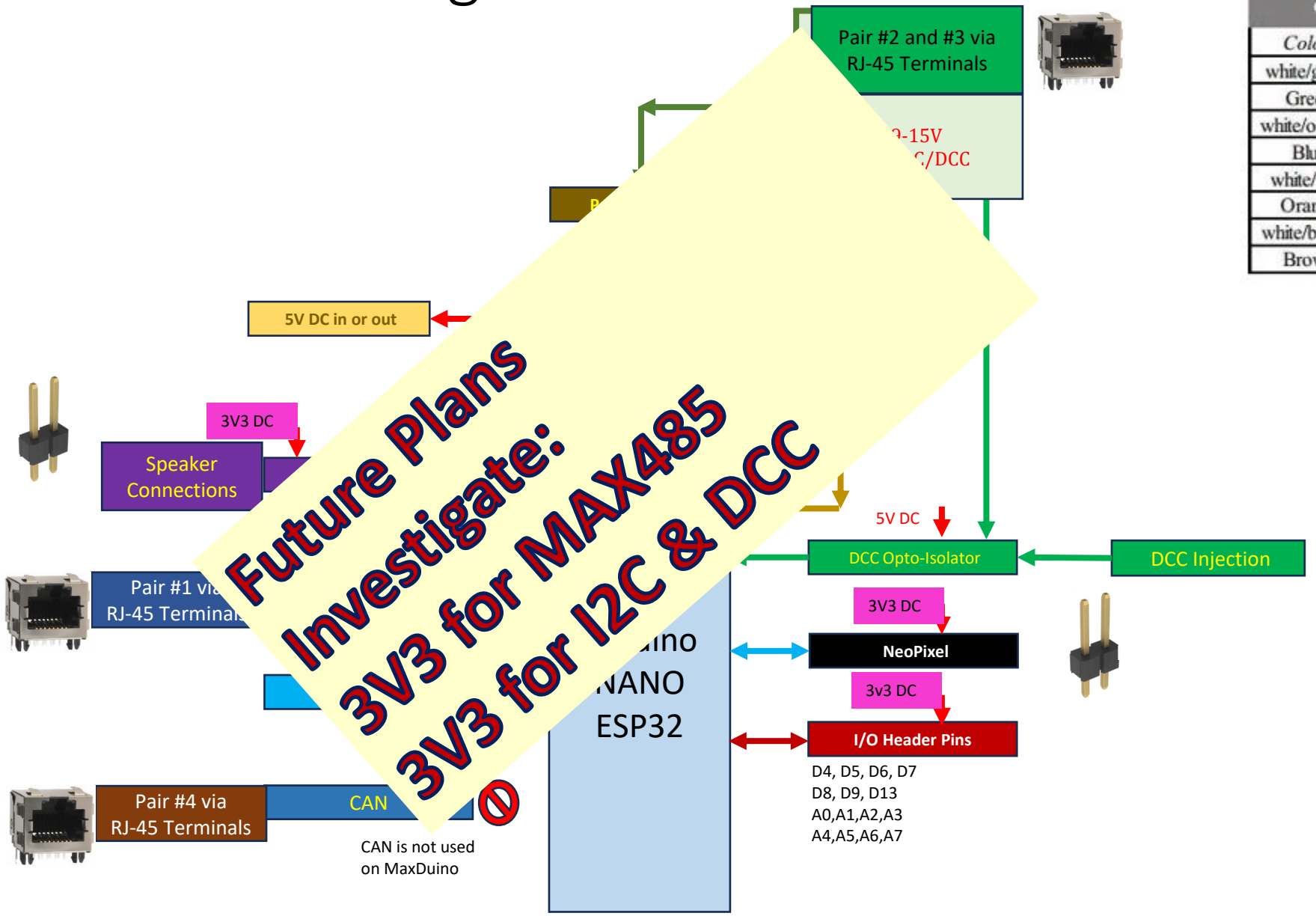
Design Review Full PCB



Management of Change

Rev 0 – Dec 2022 first 5 PCB's (enough for 10 prototypes) ordered. Platform for developing test code.
Rev 1 – May 29 2023 second batch of 5 PCB's ordered. Tried fixing DCC decoding, added 5 volt power input, added NeoPixel, changed 3 pin header and I2C header arrangement, Larger power traces, clarified stenciling. Found flaw in DCC Decoder section. One PCB (Two MaxDuino) were given to Ian for evaluation.
Rev 1.1 Revised DCC Decoder circuit. Additional stencil near RGB NeoPixel to aid with orientation, additional stencil for double isolation of DCC signal pickoff from power supply section.
Ordered two batches of 5 PCB's June 29 2023 from JLCPCB. All testing passed.
Rev 1.2 Revised RS485 circuit to use 2 pins for enabling direction control. This puts both into high impedance state when doing ICSP. Also added flow through for the on board NeoPixel and ability to bypass the NeoPixel if not needed.
Rev 1.3 Previous revision changed the pin for DCC decoding – to one that does not support interrupts. This version moves DCC decoding back to pin D3 while retaining 2 pin direction control on RS485. All testing passed on October 14 2023.
Rev 1.4 During final prep of MERG article editor suggested minor changes that in turn suggested minor fixes to front and back stenciling. Circuit was electrically unchanged from version 1.3
Rev 1.5 Swapped screw terminals for RJ45 carrying combined power and RS485 A&B signals. Changed full wave rectifier into 4 discrete diodes (Can now use high speed diodes according to design preference). Added voltage drop diodes to reduce loading on regulator. Removed speaker screw terminals to make room for above (replaced with 2 pin headers).

MaxDuino R2 Block Diagram



Straight LAN cable			
Color		RJ45	
Color	*Pair*	*Pin*	*Function*
white/green	3	1	Power +
Green	3	2	Power -
white/orange	2	3	Power +
Blue	1	4	RS485 A
white/blue	1	5	RS485 B
Orange	2	6	Power -
white/brown	4	7	CAN H
Brown	4	8	CAN L