Megapixel Octoluminate Controller and 4 Port Receiver by CCR
Assembly And Installation Guide

Version 1.3

Content

Table of Contents

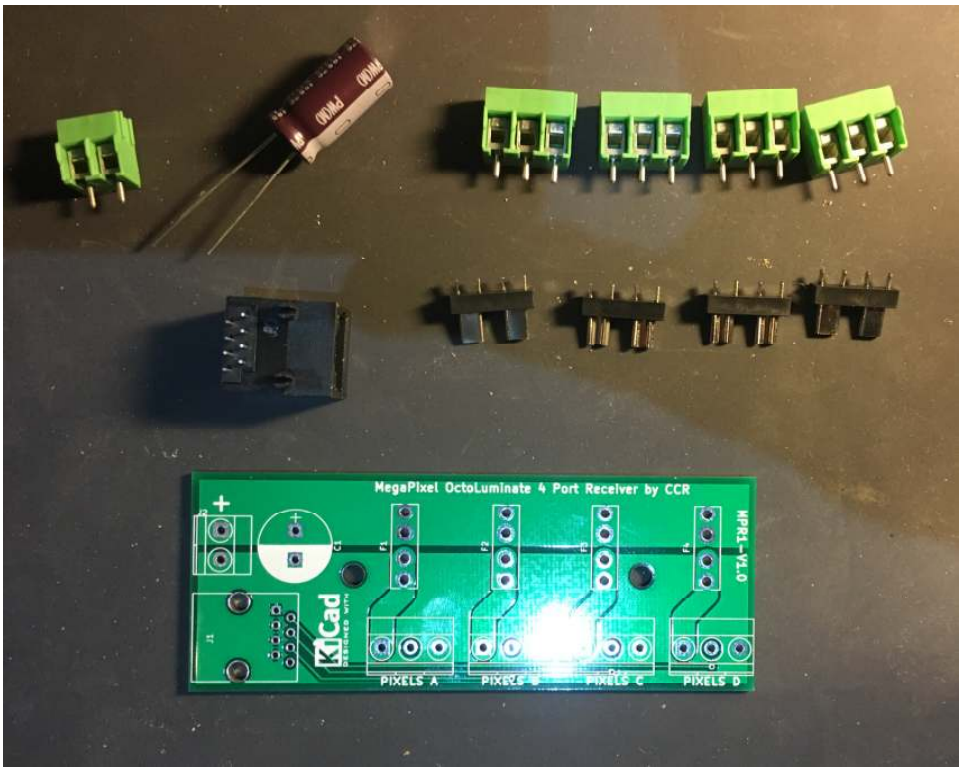
MegaPixel OctoLuminate 4 Port Receiver by CCR

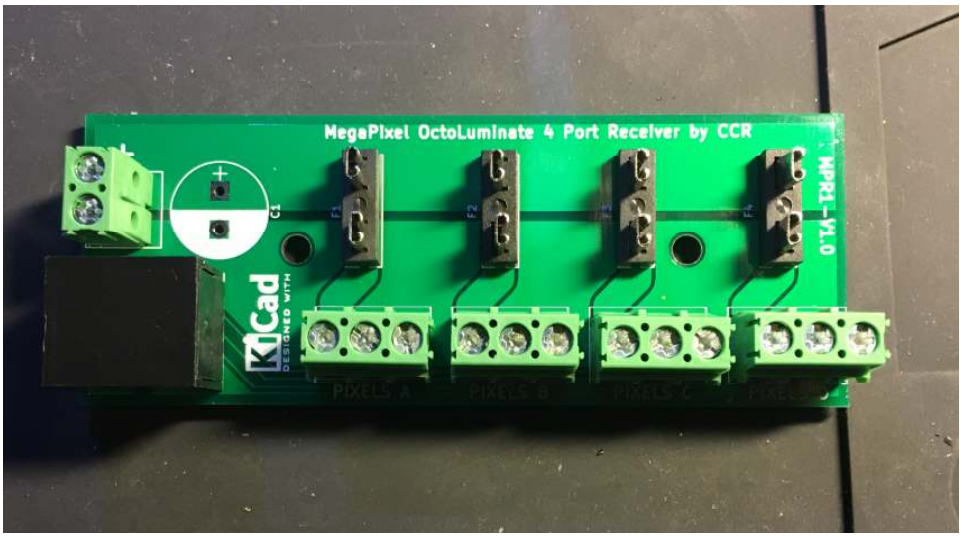
Parts List

Index	Part Number	Manufacturer Part Number	Description	QTY Needed	Board Location.
	BK-6013-ND	BK-6013	FUSE HOLDER BLADE PCB	4	F1-4
	609-1046-ND	54602-908LF	CONN MOD JACK 8P8C R/A UNSHLD	1	J1
	493-1790-ND	UPW1C102MPD	CAPACITOR ALUMINIUM 1000UF 20% 16 RADIAL	1	C1
	277-1667-ND	1935161	TERM BLOCK PCB 2POS 5.0MM GREEN	1	J2
	277-1578-ND	1935174	TERM BLOCK PCB 3POS 5.0MM GREEN	4	PIXELS A, B, C, D
			MINI BLADE FUSES (110mm) Appropriate rating for lights	4	

Assembly

Assemble 11 parts per receiver board.





Prior to fixing the capacitor, confirm

- No shorting across circuits
- Zero resistance across data line from J1 to Pixel . Note data is middle pin on pixel and each alternate pin on RJ45
- Each Data line is not shorted to ground
- +ve not shorted to ground
- NOTE PIN ORDER ON TERMINALS FROM LEFT TO RIGHT IS POSTIVE +V , DATA, GROUND.

Fuses

Use mini blade fuses (110mm size) of appropriate rating.

Each Pixel driver supports maximum of 4 universes ie 2040 channels.

Do not design each Pixels driver to be over loaded. Remember to inject power into the strings and connect to a common ground.

MegaPixel OctoLuminate Controller by CCR

Parts List

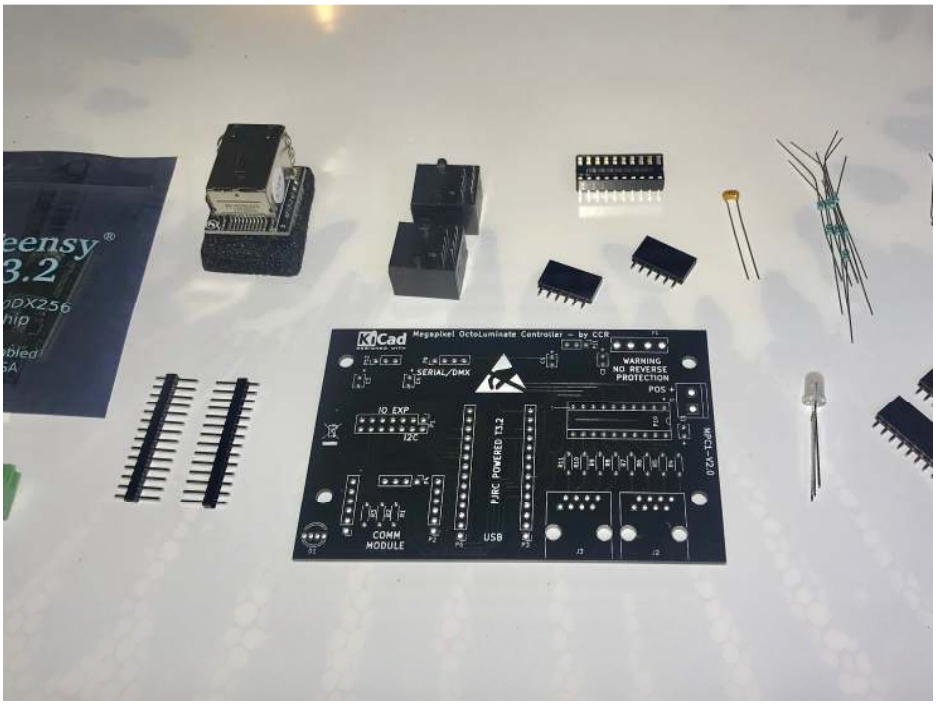
Part Number	Manufacturer Part Number	Description	QTY Needed	Board Location.
CF18JT68R0CT-ND	CF18JT68R0	RES 68 OHM 1/8W 5% CF AXIAL	1	R3
CF18JT15R0CT-ND	CF18JT15R0	RES 15 OHM 1/8W 5% CF AXIAL	2	R1, R2
609-1046-ND	54602-908LF	CONN MOD JACK 8P8C R/A UNSHLD	2	J2, J3
296-1476-5-ND	SN74AHCT245Nsn	IC BUS TRANSCEIVER 8BIT 20DIP	1	P10
AE9998-ND	A 20-LC-TT	CONN IC DIP SOCKET 20POS TIN	1	P10
BC1162CT-ND	K105Z20Y5VF5TL2	CAP CER 1UF 50V Y5V RADIAL	1	c1
277-1578-ND	1935174	TERM BLOCK PCB 3POS 5.0MM GREEN	1	J1
RNF18FTD100RCT-ND	RNF18FTD100R	RES 100 OHM 1/8W 1% AXIAL	8	R4-11
BK-6013-ND	BK-6013	FUSE HOLDER BLADE PCB	1	F1
5mm 4 pin Common Anode Diffused RGB Tri-Color		RGB Common **Anode** Diffused LED	1	D1
	PJRC Website	TEENSY 3.2	1	
	wiz820io	Wiznet W5200 or W5500 Module (note needs different software libraries)	1	P2
		Female 6 Pin 2.54 Socket	2	P1, P2
		Female 14 Pin 2.54 Socket	2	P5, P6
		MINI BLADE FUSES (110mm) Appropriate rating for controller & expansion boards	1	

Assembly

Assemble 22 parts per controller board.

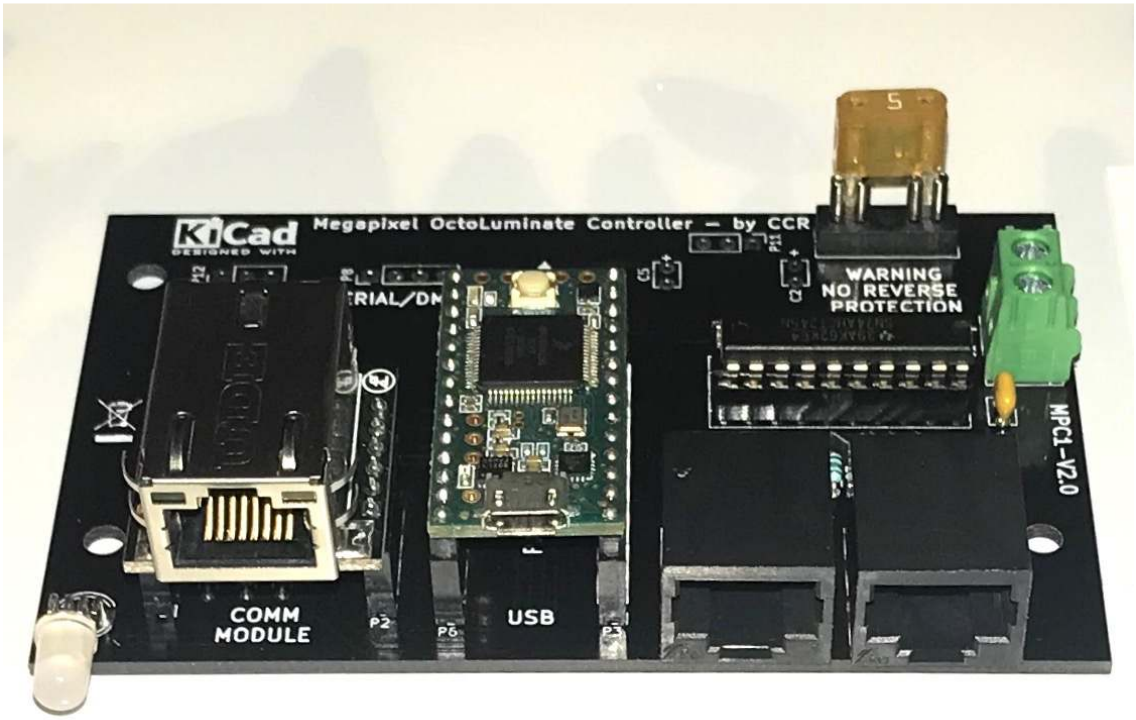
Soldering Teensy, Wiz820io and the transceiver IC to board is optional but Parts list includes headers.

The board contains several connections which are for future expansion and one section which is used when powering from an external 12v supply. (see Power section below)



Notes

- P1 is not labeled but is opposite side of expansion module, parallel to P1
- LED is common ANODE
- When testing, do not connect the Teensy, Wiz820io and transceiver.
- Confirm…
  - No shorting across circuits
  - Each Data line is not shorted to ground
  - Each data line is connected without a break or a cross circuit.
    - Pins 11-18 connect to R11-R4 respectively and no cross circuits.
    - Teensy pins 2,14,7,8,6,20,21,5 connect to IC pins 2-9 respectively and no cross circuits.
  - +ve not shorted to ground at Teensy
- After power option selected, power the board and confirm Teensy Vin pin is receiving 5v and the transceiver (Level Shifter Chip) across pins 10 and 20 are receiving 5v.



Fuses

Use mini blade fuses (110mm size) of appropriate rating.

Each Pixel driver supports maximum of 4 universes ie 2040 channels.

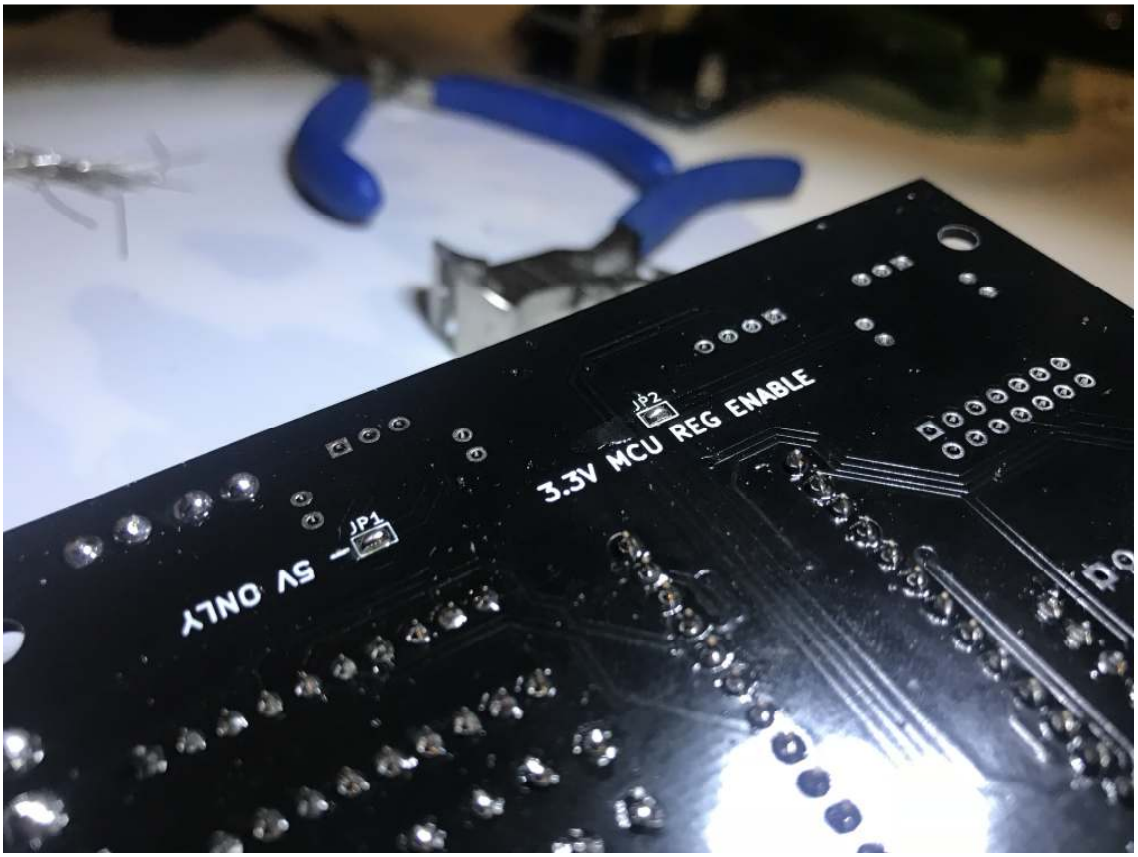
Do not design each Pixels driver to be over loaded. Remember to inject power into the strings and connect to a common ground.

Power

The Controller has three power options to suit different conditions.

Depending on the power option, you will need to solder jumper pad or add extra components.

There are 2 (two) solder jumpers on the back of the board which are clearly identified.



3.3v MCU Reg Enabled: when the 3.3v regulator on the Teensy board is used to power the controller board. In MOST cases this is always soldered. If you plant to add something more than the ethernet module then you may need solder in an independent 3.3v regulator**.

5v only: Solder this when an external 5v power supply is used to power both the Teensy and the controller board. If using 12+v leave the 5v unsoldered! Leaving this soldered with voltages higher than 5+ volts will damage the components on the board!

NOTE:

DO NOT power the Teensy from USB and an external source at the same time.

DO NOT power the board from an external 12v supply when 5v pad is soldered!!!

Option 1 – Teensy powered by USB

This option is for people wish to power the Teensy and the controller board from the USB port on the Teensy.

The board has a jumper pad labelled “3.3v MCU Reg Enabled”. Solder this pad to enable the board ethernet module to be powered from the Teensy.

**The Teensy regulator has a maximum output of 250mA and will be able to power an Ethernet module but is unlikely to power any future expansion modules. In the future a regulator and external supply may need to be fitted to allow for further expansion.

NOTE!! If using an older Teensy 3.1, you must leave the 3.3v jumper unsoldered and install a 3.3v regulator. This is because the Teensy 3.1 has a week regulator unable to power any expansion models such as the Ethernet module.

Option 2 – 5v

This option os for people who will have an external 5v power supply to power the Controller board and all it's components.

The board has a jumper pad labelled "5v only". Solder this pad to enable the board to use the external 5v supply.

Check the Teensy has 5v to the Vin pin BEFORE connecting the Teensy.

DO NOT power the Teensy on both USB and the 5v supply.

DO NOT connect this pad if you are running any other external supply voltage. Any voltage higher than 6v is likely to damage the Teensy.

Option 3 – 12v

A 12v regulator must be added to the board when using a 12v supply.

The following components are required.

Part Number	Manufacturer Part Number	Description	QTY Needed	Board Location.
811-2196-5-ND	OKI-78SR-5	Non-Isolated Switching Regulator DC-DC 7v-36v IN/5v OUT	1	p11
445-173302-3-ND	FG20X7S1H106KRT06	CAP CER 10UF 50V X7S RADIAL	2	c2, c5

The jumper solder pads on the back of the board MUST NOT be connected.

Check the Teensy has 5v to the Vin pin BEFORE connecting the Teensy.

DO NOT power the Teensy on both USB and the 12v supply.

Network Configuration

The Wiz5200 and the Wiz5500 need to use a special library for Ethernet to achieve maximum performance.

Whilst it is possible to use the existing Ethernet library for the Wiz5200 and the Ethernet2 library for wiz5500, this will cause problems and you may not achieve maximum performance. Errors could include missed frames or missed lights during a sequence (due to packet loss).

Installing the correct Ethernet library

You need to use a modified version of the 1-socket Ethernet library for Teensy from alex-Arc. This library reduces the SPI overhead to give back SRAM for TCP and UDP packets and as such increase performance.

Libraries are typically located in one of three locations.

- Library folder where the Sketches are saved
- The libraries folder in programs
  C:\Program Files (x86)\Arduino\libraries
- The hardware folder for teensy
  C:\Program Files (x86)\Arduino\hardware\teensy\avr\libraries

It is important to remove (you can either delete or move to a temp folder elsewhere), the Ethernet folder from these directories.

To install the new library

1. Download the zip file containing the **Ethernet-1-socket** library.
2. In Arduino, go to Sketch->Include Library->Add .ZIP Library
3. Select the Ethernet-1-socket zip file and add.

DHCP

If you are using DHCP and setting the IP address to a fixed one in the DHCP tables, please comment out the appropriate TWO sections of code in the top and in the begin

```
// Uncomment if you want to use static IP
//*********************************************************
// ethernet interface ip address
// IPAddress ip(192, 168, 10, 100);  //IP address of ethernet shield
//*********************************************************
 // ***
```

```
// ** if using static IP use following
// Ethernet.begin(mac,ip);
// ***
// if using DHCP use following
Ethernet.begin(mac);
```

LED Colour Order and other strip settings

Setting the Colour order by changing the code accordingly…

```
// Using different LEDs or colour order? Change here...
// ********************************************************
LEDS.addLeds<OCTOWS2811, BRG>(leds, NUM_LEDS_PER_STRIP);
LEDS.setBrightness(100);
 // ********************************************************
```

Use lighting software to set the colour to a single static colour and rotate through each of red, green and blue to confirm the correct colour order.

Matrix size

You should not change the settings without an understanding of what these settings do.

As per OctoWS2811 library all SPI Ports (8) will have the same pixel count (680 if using max).

There is no benefit in reducing the number of LEDs per universe; the universe count or number of LED per strip. (if these are changed you will also need to change other items such as number of leds and the channel count.)

```
/// DONT CHANGE unless you know the consequences...
#define ETHERNET_BUFFER 636 //540
#define CHANNEL_COUNT 16320 //because it divides by 3 nicely
#define NUM_LEDS 5440 // cannot go higher than this - Runs out of SRAM
#define NUM_LEDS_PER_STRIP 680
#define NUM_STRIPS 8
#define UNIVERSE_COUNT 32
#define LEDS_PER_UNIVERSE 170
```

Compiling the software

It is very important you DO NOT cache the compiled core. You WILL experience problems if you do this.

Go to file->preferences and ensure '**Aggressively cache compiled core'** is NOT checked.

You must also compile with the following settings

- Tools->CPU Speed -> 120Mhz (overclock)
- Tools->Optimize->Fastest

If you receive the error, **'SPIFIFO.begin(ss_pin, SPI_CLOCK_30MHz); // W5100 is 14 MHz max'**, you have the incorrect Ethernet library installed.

Lighting software

When setting lighting software such a Jinx, Glediator, xLights, it is important to set the number of channels per universe to 510. Refer to the software help files to find out how to set the number of channels per universe.

Please not, when using Jinx we observed that the 1 pixel wide radar scanner did not light every pixel. This was tested using the Glediator feed on the serial port as well. This is assumed to be something with Jinx and not the controller board.

Debugging software

There should be no reason to use Serial.print to debug the software. However if you do choose to debug the software this will dramatically slow down the frame rate and may cause other problems due to reduction memory for the buffer.

There are a number of serial.print commands which have been commented out which are used for debugging purposes.

Controller board LED

During the power up cycle of the controller board, the integrated LED will turn blue for around 10 seconds and then turn off.

Whilst the board is receiving information from the Ethernet connection (specifically DMX frames) then the LED will flash green.