LED Light controller for enlargers

DISCLAIMER (sadly has to be said): *I take no responsibility for the contents of this manual. Any liability is expressly rejected. Anyone who uses this manual does so at his own risk. The handling of electrical voltages is not without danger and requires caution. No results are guaranteed by using this manual in any way.*

Version 1.0, 17.7.2020

1. Introduction

This instruction helps to build a LED light source for photographic enlargers.

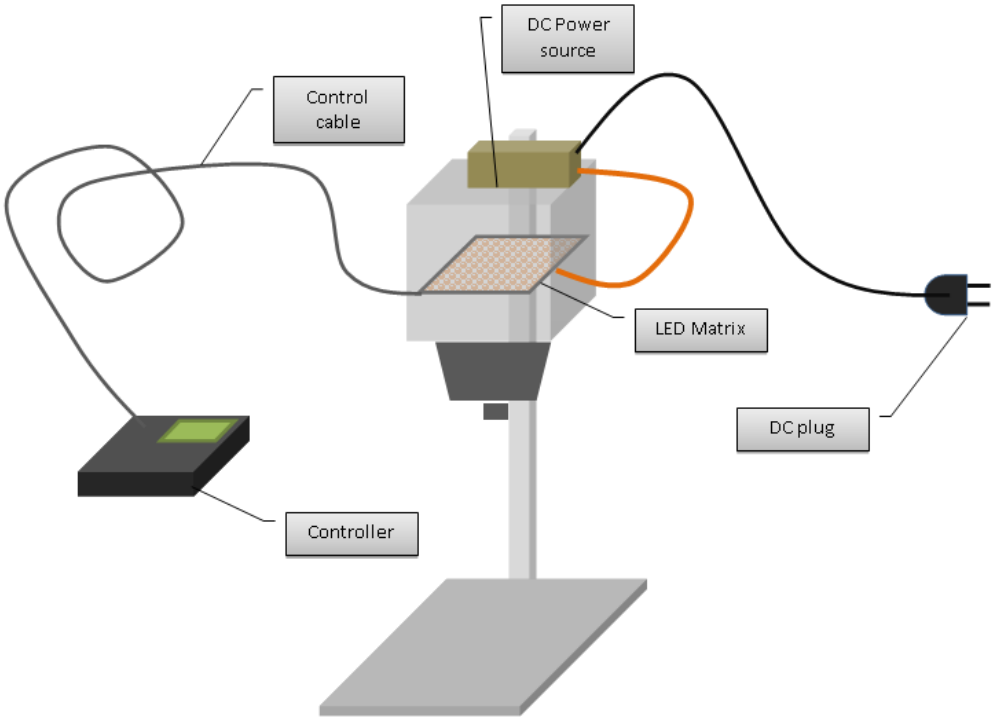
All content is free to be used for non-commercial purposes.

Any LED power setting can be used from 16 LED (small 35mm direct light source), 64 LED (normal 35mm/6x6 light source) up to 256-512 LED for LF enlargers.

The base is the use of the WS2812b LED. Each LED has a power consumption of 60mA (0.06A) keep this in mind when selecting an appropriate 5V power source.

The WS2812b LED has following wavelengths: Red: 620-625nm / Green 522-525nm / Blue 465-467nm.

The components for a LED light source setup are



a. Expected costs

This is not easy to say but based on my experience I can give a raw estimate for a LF 4x5” enlarger (<u>without any warranty</u>):

Arduino (14$) + 256 LED Matrix (18$) + LCD Display (14$) + Keypad (4$) + Buttons & Switches (6$) +Power supply (20$) + GX16 plug (4$) + Cables, enclosure and material (15$)
TOTAL about 90-100$, depending where you buy Even something less if you are good in searching.

b. Calculate the electrical power need of the LED source

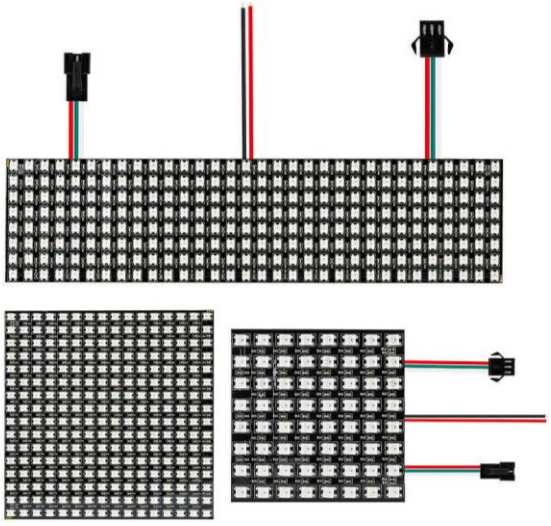
Most small enlargers have a light bulb that can be replaced with a smaller LED matrix. Some other enlargers can be adapted by bypassing the “old” tungsten light source with a new diffusion box that has an LED matrix included.

For the calculation of the needed power of the WS2812b modules, you may consider this table:

Amount of LED	**Size of module**	**Power consumption at 5V**	**Current**	**Equivalent Halogen**
16	3x3 to 4x4 cm	4-5W	0.9 A	25W
64	8x8 to 10x10 cm	19-21W	3.5 A	75-100W
256	16x16 to 17x17 cm	75-80W	15.5 A	250-300W

I am always cautious with this “equivalent” transformation, as the LED will not give all the power if modulated in the color needed. The amount of effective light can be up to 50% less. However, this is also the case if you use filters on a color head.

There are several other form factors of WS2812 LED matrixes or similar. There are round and circular, hexagonal and other shapes. Consider that a single WS2812 LED gives 0.25-0.3W of power and needs 0.06A of current.

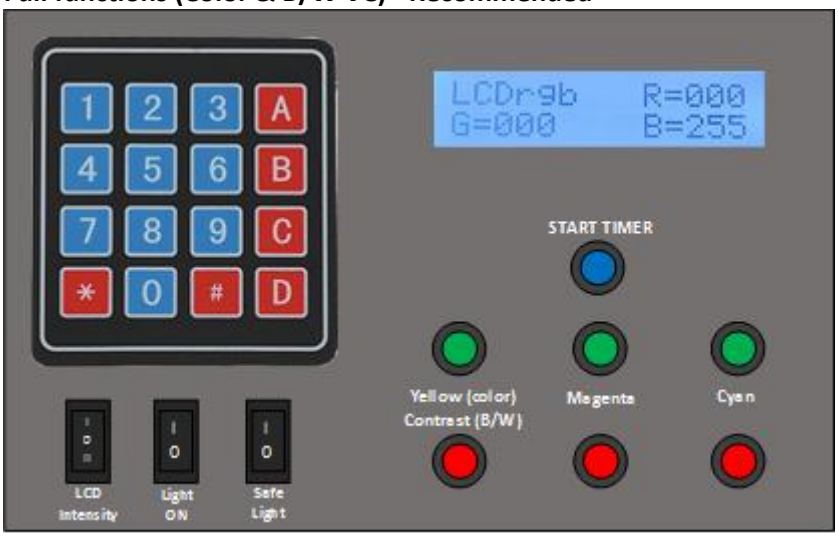


You can combine all shapes of modules into an array. This may occurs if you need a spaced light source of 4 modules. Every module needs to me connected to a power supply and the data signal have to be routed the correct way to the controller.

c. Possible panel layout of LED controller

The main Layout of the controller may look like this one (size shape may vary depending of the enclosure you use). You can choose any layout. I will describe mostly the full function layout. In the code some parameters can be set in order to be compatible to smaller layouts.

Full functions (Color & B/W VC) - Recommended



B/W VC only (example)

B/W compact (example)

2. Tools and material needed

- Soldering iron. Best some model for electronics around 25W. Soldering lead wire as well
- Shrink tube or insulating tape to isolate the soldering points
- Painter tape for marking cut out holes an protect the enclosure
- Coping saw to cut squares in enclosure
- Small file
- Drill with bits and a cone drill up to 16mm
- Dual tape for fixing things in enclosure
- Small screws to fix Arduino (3mm)
- A glue gun can be useful too
- If possible a multi meter to check connections and DC voltage

3. Bill of materials

Most of the material may be found for a convenient price at Aliexpress or eBay. Local online shops have these components too, but my experience is the parts are the same.

For the LED light source you need these components:

a. LED matrix:

Any WS2812 or WS2812b LED matrix. (But better no LED strips: These are difficult to wire and have no enough light per surface).
The amount of LED depends from the enlarger that you may want to convert. Consider the power of the original light source and use the table above (point 1.a) to see how many LED you need. Normally this will be a multiple of 64 as the most modules in an 8x8 size. Search of WS2812b on Aliexpress, eBay or any other online platform that sells similar products.

SEARCH TERMS: *WS2812b Matrix*

b. Arduino controller.

Any Arduino Mega(2560/1280) or mega pro. A smaller Arduino (Arduino Uno or similar) can be used with reduced amount of functions and buttons. Again, you may search on Aliexpress, eBay or any other online platform for such an Arduino Mega 2560. You will need a computer (Windows or MAC) to load the program on it. Normally an USB cable is provided too.

SEARCH TERMS: Arduino Mega 2560

c. Power source

A suitable power supply should match the amount of LED used. In the table at point 1.a is shown how many Amperes you need. The Ampere is the measure unit for the current (let's say – even if not really correct – the "power") of the module. The voltage is always 5V. Do NOT use other voltages because it will BURN the LED's. Always buy a power supply with at least 20% more power than you calculated. A power supply with higher Ampere (A) won't damage anything. An underpowered supply may won't allow the LED to light up.

Power supplies come in different form factors.

As module power supply:

Or as version with plastic enclosure

Both have advantages.

The module version comes in versions that are more powerful: up to 60A. **BE CATIOUS**: Use big sized cables for such power supplies. For powers over 15A user multiple cables. At least 16AWG (1.5mm2) cables. Using thin cables can cause short circuits and severe damage. This modules are best fixed close to the LED source on the enlarger head. Avoid power cable lengths aver 80cm. Any length under 30-40cm is optimal for high power projects.

The Version in Plastic enclosure is good for smaller projects (35mm enlargers) and do not needs to be fixed close to the head. Mostly any LED project with a power need up to 6-8A can be done with this type of power supply.

SEARCH TERMS: ***xA 5V power supply*** (change the 'x' with the amount of Ampere you need)

d. Keypad

For set the time of the timer and other functions a 4x4 membrane keypad is needed too. The program works with any matrix keypad with 8 pin connector. Anyway it's recommended to use this layout as it's one of the most common. If you use another 4x4 keypad it's recommended that it has the numbers at the same positions. In this instruction document all commands with the keypad use this layout as reference.

SEARCH TERMS: ***4x4 Keypad***

You will need a set of male-male jumper wire (better around 25cm length) in order to connect the keypad to the controller. Buy at least a set of 40 cables, as you will use them to connect the switches and push buttons. You

can also use 2.54mm pin headers (male) and solder wires on it in order to make a custom connector (these can be quiet like AWG 24 or AWG26). But pin headers should be long enough to get into the Arduino and keypad connector in order to make contact to the signal. Using universal jumper wire is easier.

SEARCH TERMS: ***male-male jumper wire***

e. LCD display

The program uses a specific 16x2 LCD RGB Backlight display (i2c interface) from Grove/Seedstudio. There is no (simple) way around to use another as this specific model. It cannot be found on Aliexpress (maybe on Amazon or eBay, for sure on rs-online.com). The main site to buy it is here: https://www.seeedstudio.com/Grove-LCD-RGB-Backlight.html

Some local online supplier for small electronic project have it too. DO NOT use any LCD you find elsewhere: It will not work with other LCD modules unless you change the code of the program (experienced Arduino users/programmers may can do it). In order to use the code/program as it is use ONLY this version of Seed Studio.

f. Buttons and switches

For the full version of the controller, you need:

- 7 push button switches (3 red, 3 green and one blue… or whatever you want as color)

SEARCH TERMS: ***push button switch***

- 4 Toggle Switches (3 simple ON-OFF, 1 ON-OFF-ON)

3x

1x

Alternatively you can use common toggle switches

SEARCH TERMS: ***OF-OFF switch*** or ***ON-OFF-ON switch***

g. Plugs and cables

To connect the controller to the LED matrix you need a 3 core cable (no need to be shielded) and a connector like this (GX16 3 pin):

You may use another connector or solder the cable direct.

For the connection between the power source and the LED matrix you may use an AWG18 - AWG16 wire. A 240V power cable with at least 1.5mm2 section can be used too.

The power supply will need an AC power cord with plug. Some thin signal wires are needed too.

To connect the display and the power to the Arduino some 2.54mm pin head connectors are needed (about 12 pins)

h. Installation materials

- Gland connectors (for the cable connection to the enlarger head)
- 2.9mm screws to fix the Arduino board
- Dual side tape to fix the LCD display

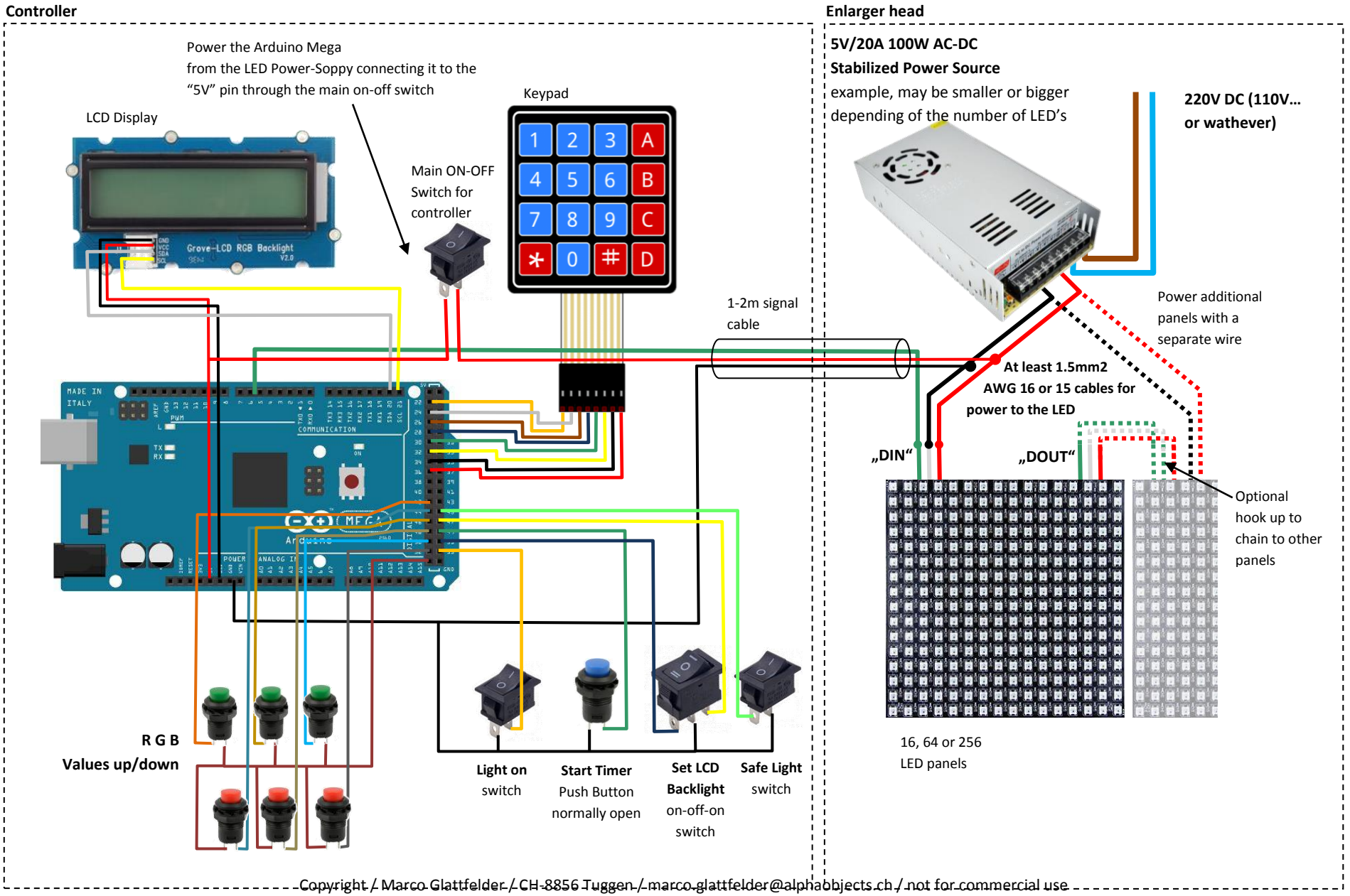
i. Enclosure

Any enclosure with a size of 15-18cm to 12-14cm and at least 4.5-5 cm height may be ok. I can recommend Hammond 1599KBK enclosures. You can use any metal, plastic or wood enclosure you can find or make yourself. I found RS components (rs-online.com) as a good source for enclosures.

The height of the enclosure is critical, as the jumper cables that go into the Arduino need some clearance. 90 degrees connectors may help for enclosures that have a height around 3.5 cm, but less than that will be difficult. You may need to get the rid of some plastic turrets on the inside of the enclosure panels. The thinner panel should be the bottom of the controller. All this preconfigured plastic parts for screws may affect the cut out of the square holes for the LCD display. Always check what is on the backside of the panels when you cut.

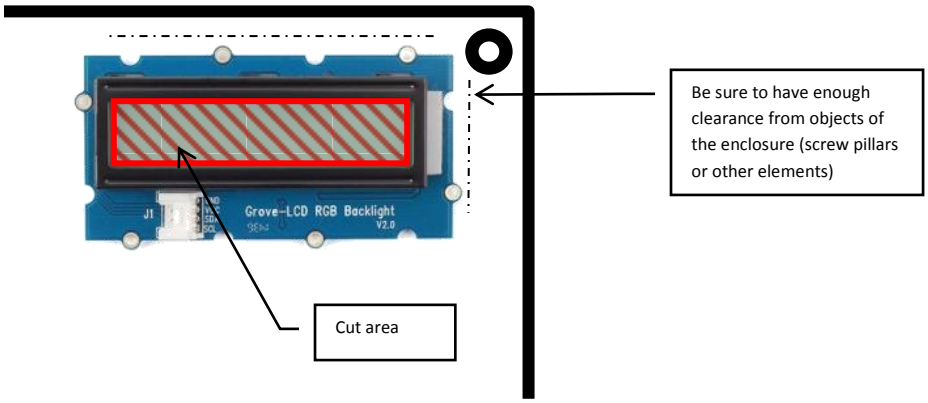
I used a175x125x45mm enclosure (the one I show in the pictures), and it was a bit too tight, especially the height. Better use enclosures (or make one) with more space.

4. Main circuit schema and wiring layout

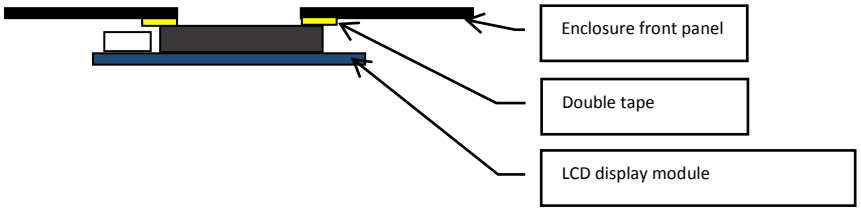


1. Installation in enclosure and construction tips

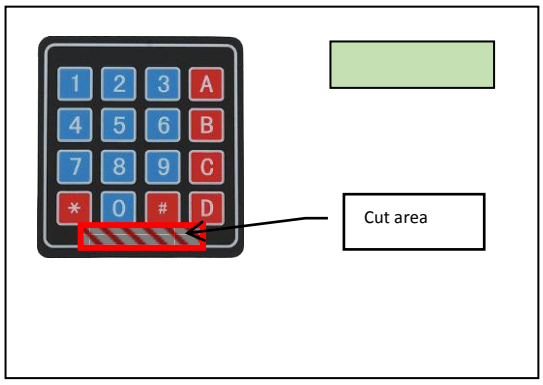
First, you should cut out a square hole for the LCD display on the top right corner of the enclosure front panel. Measure the size of the visible area of the LCD display (light green area) and cut out a hole of the same size. Be careful to consider the clearance of the whole LCD module size.

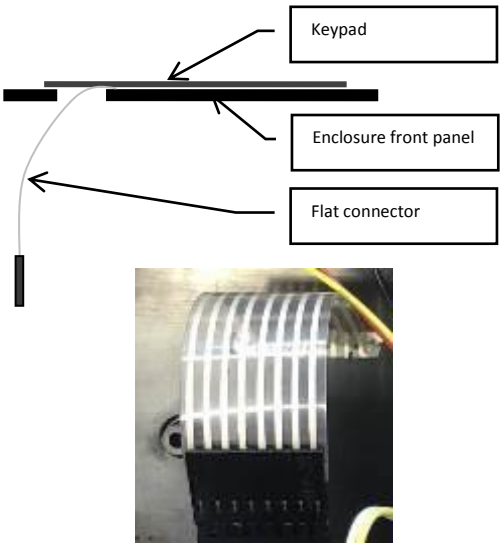


Put the connection cable (Groove connector) into the plug. The cable cannot be plugged in once the LCD is fixed with dual tape. Use some thin strips of dual tape to fix the LCD on the enclosure. You can also use screws to fix it, but I found the dual tape more practical.



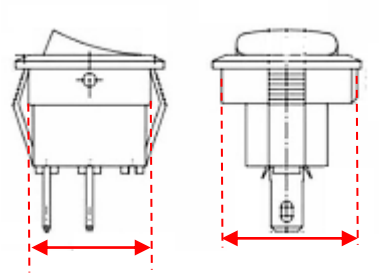
Now you have to cut a slit hole for the keypad cable. You can drill 3-4 holes and make the slit with a file. The keypad should be placed more or less on the top left corner of the enclosure front panel.

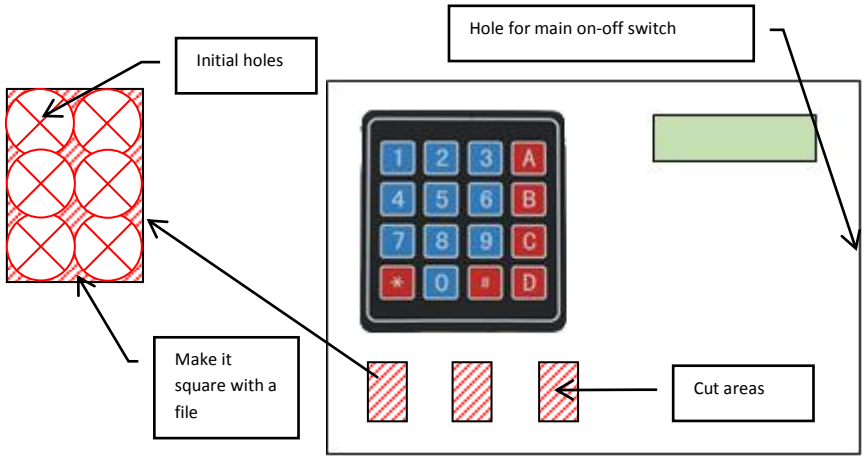




On the back of the keypad, you see where the flat connection tape goes out of the keypad. After you cut out the hole, remove the paper from the back of the keypad and stick it on the front panel of the enclosure. The flat connection tape should go through the hole you cut out.

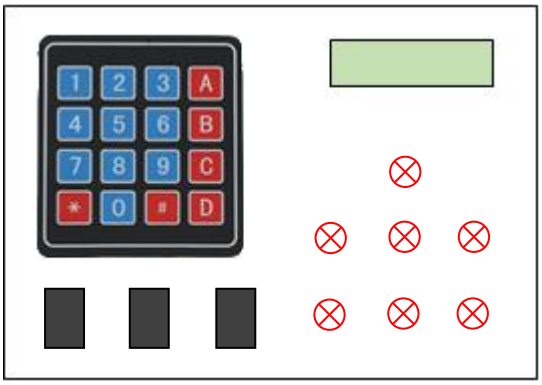
Now you have to cut out the holes for the switches. Measure the correct size of the switches hole. You can drill some holes (1 to 6) as start and make the final hole square with a file. You can use a precise saw for plastic or wood.



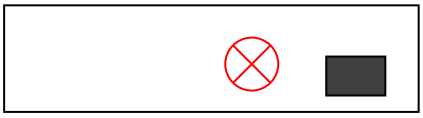


Make the same on the side of the enclosure for the main ON-OFF switch. If you use regular metal tip toggle switches instead of the plastic ones you may only need to drill round holes.

Now you have to drill the holes for the push buttons. 7 Holes in the appropriate diameter have to be drilled. Measure the diameter at the threaded part of the push button switch.

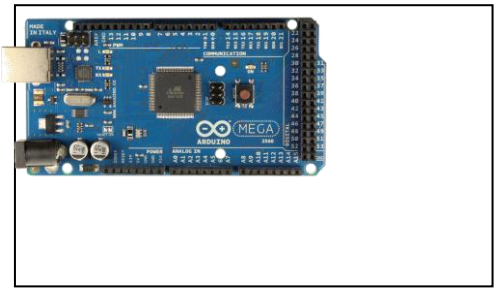


On right the side of the enclosure (beside the on-off switch) drill a 16mm hole for the signal cable connector.



On the left side you have to cut out square holes for the USB and power connector of the Arduino.

Now fix the Arduino on the bottom panel of the enclosure



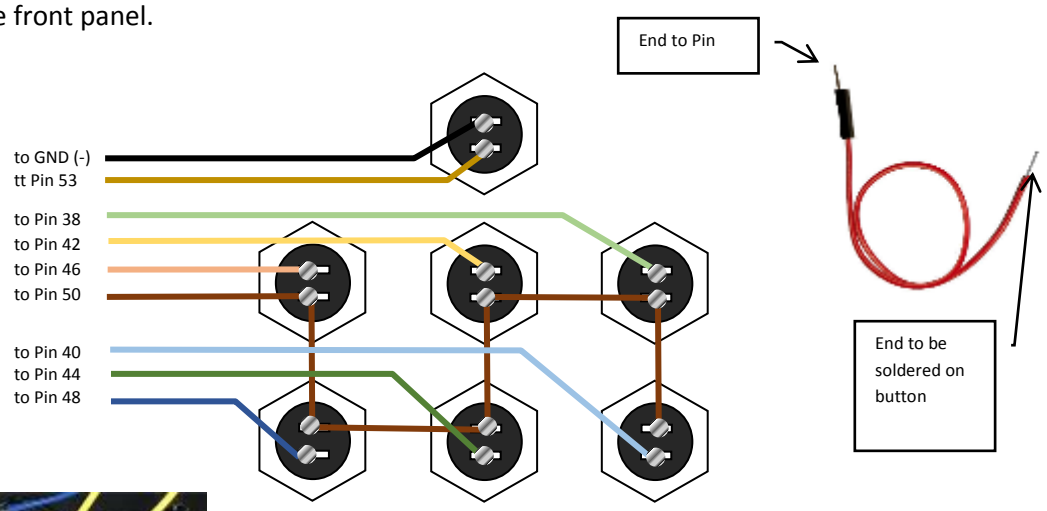
Use 3mm screws and make holes on the bottom panel. Be sure that the position and height of Arduino matches with the square holes you have formerly cut out on the left side.

Install all switches and Push buttons in the holes you made and tighten all bolts and screws if there are any.

Now you have to solder the push buttons array. 6 buttons for the up/down values have one signal in common.

Here is how to solder the wires. Use the jumper cables with the pins at the end (see the list of material 2.d) and cut one of the pin at the end. Solder these ends on the push buttons as shown here below. The end with the pin have to be plugged in the analog inputs of the Arduino at the correspondent number. Only the ground (GND) wire (black) have is without pin as it has to be connected to the common ground (GND) set of wires. These have another cable with pin that will be connected to the GND pin of the Arduino.

Rear view of the front panel.



NOTE: If the buttons reacts somehow different as expected (e.g you press magenta '-') the wires may be inverted. You may invert the sequence in the pin group 38, 42, and 46 or the pin group 40, 44 and 48 and see if the reaction of the pressed button is right.

One pin of the Start/Stop button has to be connected to ground (GND). Leave the wire to ground free as it has to be soldered to other GND wires in order to collect all ground signals. The other wire of the Start/Stop button goes to Pin 53 of the Arduino.

NOTE: If you want a compact version (B/W only) of the controller, you may use only 2 up/down push buttons and the start button. Or even only the start button. Anyway this instruction focus on the full version with YMC color option.

Now take 8 jumper cables and plug them into the connector of the keypad (here an example not related to the construction of this controller)

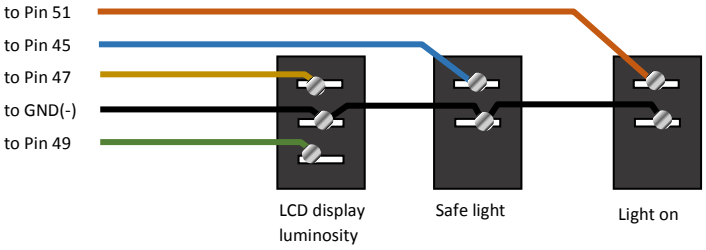
The other ends have to be plugged into pin 22, 24, 26, 28, 30, 32, 34, and 36 of the Arduino. Refer to the main circuit schema under point 4.

NOTE: If the keypad reacts somehow inverted (e.g You press '3' and you get '2' or '0') the wires may be inverted. You may invert the sequence in the pin block 22, 24, 26 and 28 or the pin block 30, 32, 34, and 36 and see if the reaction of the pressed keys is right.

Now solder the switches on the front panel. The switches have a common ground (GND) like the Start/Stop push button.

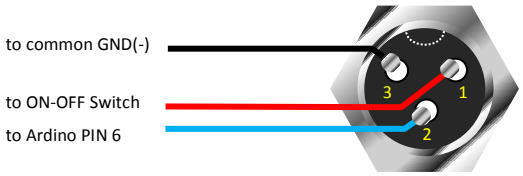
NOTE: The pictures I show are without insulating material (tape or shrink tube). This is not recommendable: better use insulation. My build was really quick&dirty.

Rear view of the switches on the front panel



Rear view of the ON-OFF switch on the right side

Now you solder the GX16 connector. Here is a rear view of the male panel connector. The pins are numbered. The signal pin is in the one middle opposed to the notch on the plug side with the number 2. Please refer to these numbers when you solder the plug for the connection cable. The 5V wire has to be soldered to pin 1 and the common ground (GND) to pin 3 of the GX16 male panel connector.



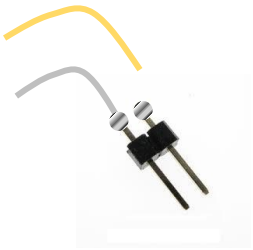
Now you have to solder the GX16 plug on the connection cable. Use a 3 core cable (3 wires) and remove the external insulation. Use a sharp knife and make a circular cut 1.5 cm under the end of the cable. DO NOT cut through the insulation but only half of its thickness. Bend the cable and break the insulation. You should have now the 3 internal wires free. These are still insulated. Strip these with a knife or scissor in order to have the core free for a length of about 3mm.

Now open the plug by removing the small screw on black body of the plug.

Solder the 3 wires using on the soldering tips of the black plug body. The 5V wire (red if possible) has to be soldered to pin 1 and the common ground (GND, black if possible. to pin 3. Connect the signal wire (any color) to the pin 2. Put some tape for insulation (shrink tube cannot be used here as the wires are too short. Close the plug again and feed the cable through the opening of the plug.

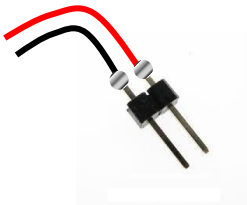
If you have a multi meter, check if there is electrical connection between the red wire (or whatever color you have chosen for the 5V) at the end of the connection cable and the 5V signal at the GX16 male panel connector on the enclosure. Check for short circuit between the pins of the connector and the wires. There should be no short circuit (if you measure some high ohm values it may come from the signal pin of the Arduino: that is ok).

Now you have to solder the pin header connectors on the end of the LCD display cable. Remove the end connector the Groove cable that comes from LCD display (cut it). Solder the white and the yellow cable on a 2 pin head connector. This connector have to be plugged to the pin 20 (white wire) and 21 (yellow cable) of the Arduino.



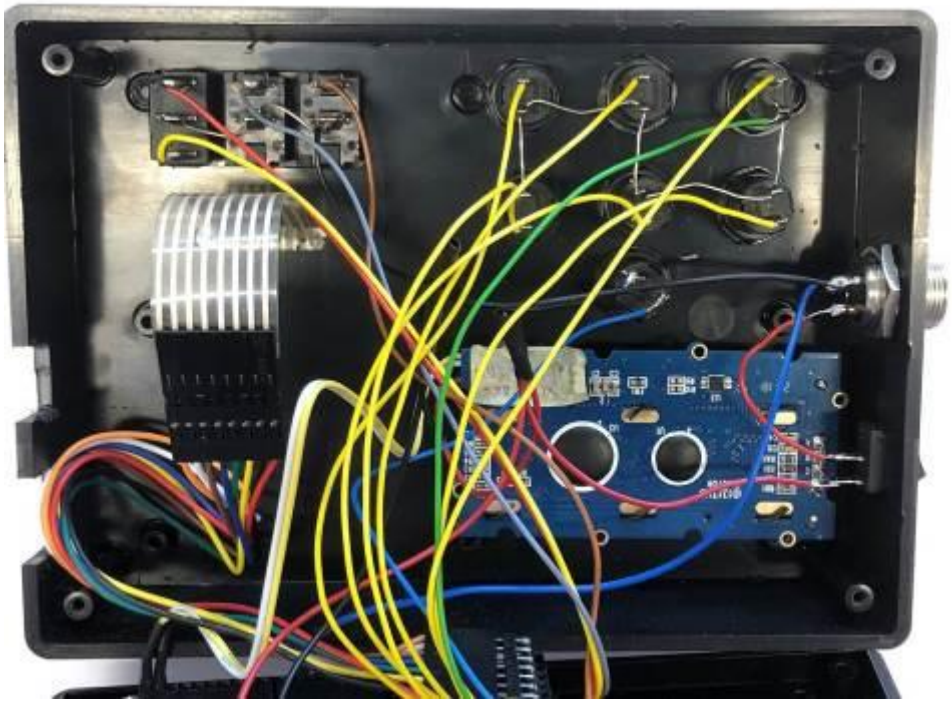
Solder all GND (ground) signal cables (coming from: start/stop button, common GND switches, black wire of the LCD display and the GND of the GX16 connector) together with another black wire of about 10cm. Insulate the soldering point. Make the same with 5V signals coming from: red wire of LCD display and red wire 5V from the ON-OFF switch) and use a 10 cm red wire. Use some tape or shrink tube to insulate the soldering points.

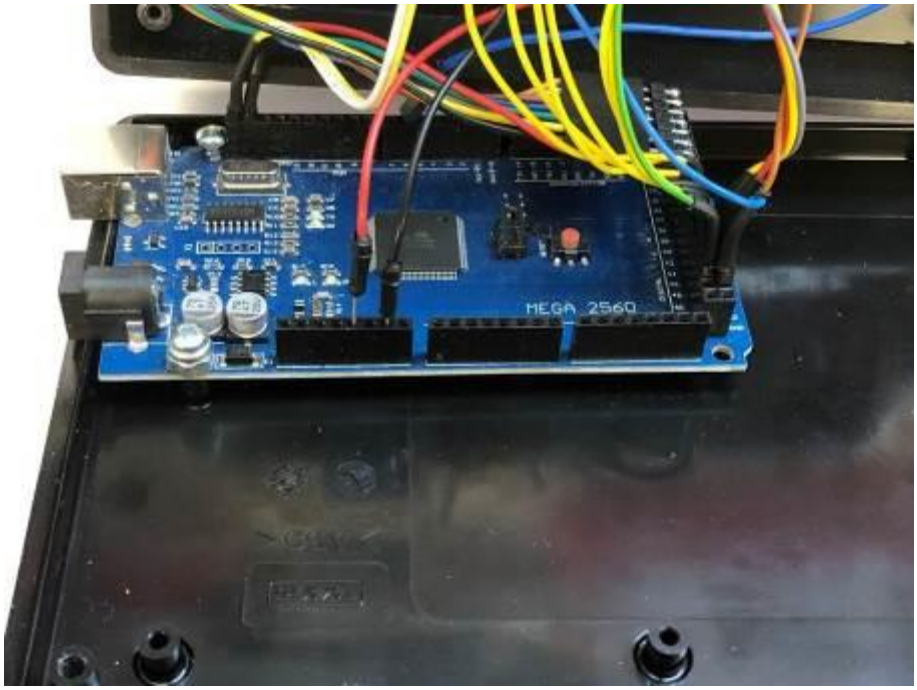
Solder the black and red wire coming from the soldered merging of the cables connection to a pin header connector. This have to plugged into the GND (black wire) and 5V (red wire) pin in the Arduino.



Now all soldering works on the controller side are made. Check that everything is wired as shown in the Main circuit schema and wiring layout picture. Be aware that in the picture the cable merging of the ground are slightly different.

This is a picture of a build





Note: The yellow/white cable of the LCD display is in other pins. These are connected parallel to pin 20 and 21. The 5V pin that goes in the Arduino was taken out for the picture and let not quiet in. I have used 90 degrees bended connector here because the enclosure wasn't so high.

2. Upload of the control software code on the Arduino

a. Install the Arduino IDE

The control software is based on the programming environment of Arduino.cc (https://www.arduino.cc/en/Main/Software). In order to be able to install the program you have to install the Arduino IDE on your computer. Download the last IDE from the link above and install it.

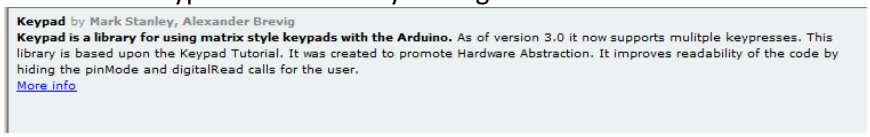
Download the code file of the LED controller. Normally it is included in the site where this project is published. Mostly as *.txt file. If you don't have send me an e-mail (marco.glattfelder@alphaobjects.ch)

If the file is still in *.txt format change the ending in *.ino.

Before you load the program, you have to upload some libraries. These are additional parts of software that are used to control the LED, the LCD display and so on. Follow the instructions here: https://www.arduino.cc/en/Guide/Libraries to install the needed libraries.

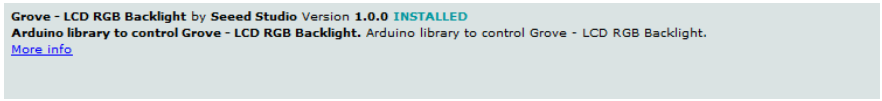
The needed libraries to install are:

- Keypad: https://github.com/Chris--A/Keypad
  Search for "keypad" in the library manager and install this one



- Neopixel: https://github.com/adafruit/Adafruit_NeoPixel
  Search for „neopixel" and install this one

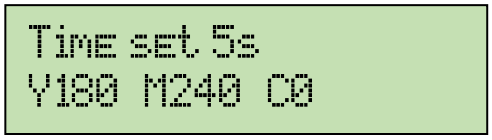
- Grove RGB LCD: https://github.com/Seeed-Studio/Grove_LCD_RGB_Backlight
  Search for „ Grove RGB" and install this one



b. Upload the controller software (.ino file)*

Connect the USB cable between the Arduino and the computer. If the IDE is connected correctly, the driver should be installed. Follow this instructions: https://www.arduino.cc/en/Guide/ArduinoMega2560

Select the *.ino file with the program of the controller and upload it. You should see the LCD display coming up with something like:



If you haven't built the LED adapter and/or led mixing chamber in the enlarger you cannot test the program yet.

3. Building of a LED panel carrier or mixing box

The building of a system or a box that supports the LED matrixes is different for any kind of enlarger. One of the best options is to build a light box that can be placed right above the negative holder. Not in every enlarger this can be done: some need the replacement of the original light source with an LED matrix in the same position.

Replacement of the original light source (Durst B30)

… and replacing with a light box right above the negative holder (Durst M605)

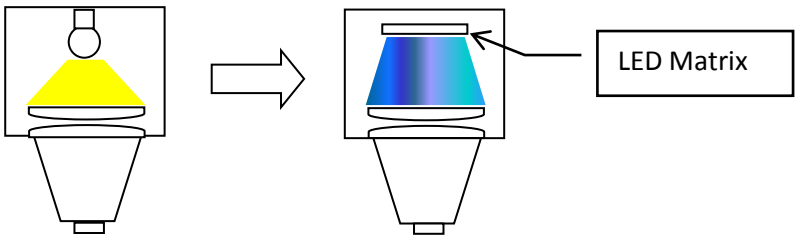
Often there is not enough space for a new LED matrix big enough to deliver sufficient light.

An enlarger with a removable light box can be upgraded with a mixing light box. If it's not a diffusion but a condenser type you may want to keep the condensers. But a diffusion box has advantages too.

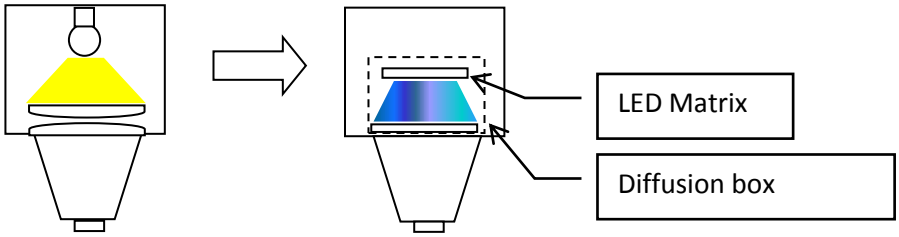
Single or double condenser enlargers for the medium and large format have mostly enough space for a new light source. Here you need a support that fits into the upper part of the housing more or less at the height of the original lamp or bulb. It can by a round or square plywood piece with some holes. If the housing was meant to be used as diffusion chamber, it does not matter if you seal the upper part of the chamber as the LED have a more homogenous light.

There are various arrangements possible

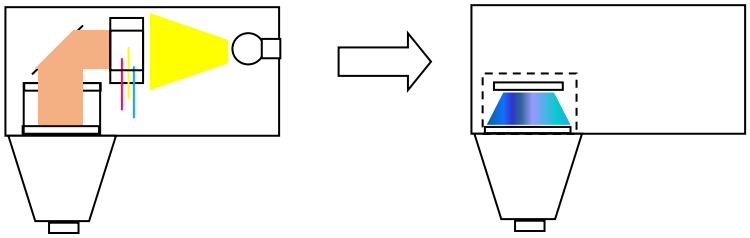
- Replacement of light bulb / adaptation of condenser enlarger



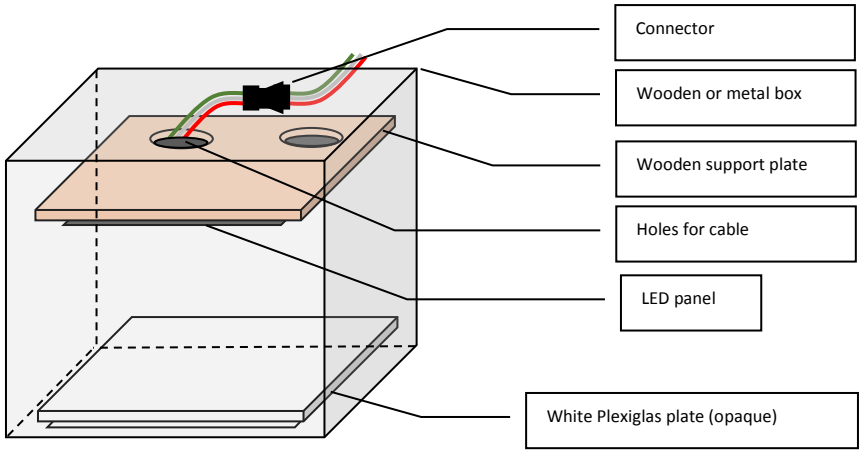
- Insertion/building of a LED diffusion light box (instead of condenser)



- Bypass/Replace color/vc head with LED diffusion light box



The ideal of the LED light source is in a diffusion box. Some condenser enlargers allow a LED source quiet easily. Consider that the box should have at last 30mm height in order to mix the single LED spots. It is not critical as other light sources that need to white filters.



I will not give any instruction on how upgrade a specific enlarger. It's more or less on you how to do it.

4. Installation of the LED matrix and the power source

Use 2 gland connectors to let the cable from the controller and from the power source pass through the walls of your enlarger. You may need to drill a hole in the enlarger enclosure. Anyway, in most of the cases you can use the opening for the lamp cable or other spaces.

The wiring of the panels is crucial. If you are installing more than one LED matrix panel

- Always power each panel with the 5V coming from the power supply separately.
- Connect the data cable from the controller (at the end of the long connection cable) to the **DIN** soldering pad of the first panel. Connect the **DOUT** soldering pad of to the first panel to the DIN soldering pad of the second panel. And so on if there are more panels. The panels have mostly already connectors that match.

The first connector at the DIN of the first panel has to be cut away in order to solder the wires coming from the controller.

Use a solid plywood base with holes where the soldering pads of the panels are and stick the panels with dual tape on it.

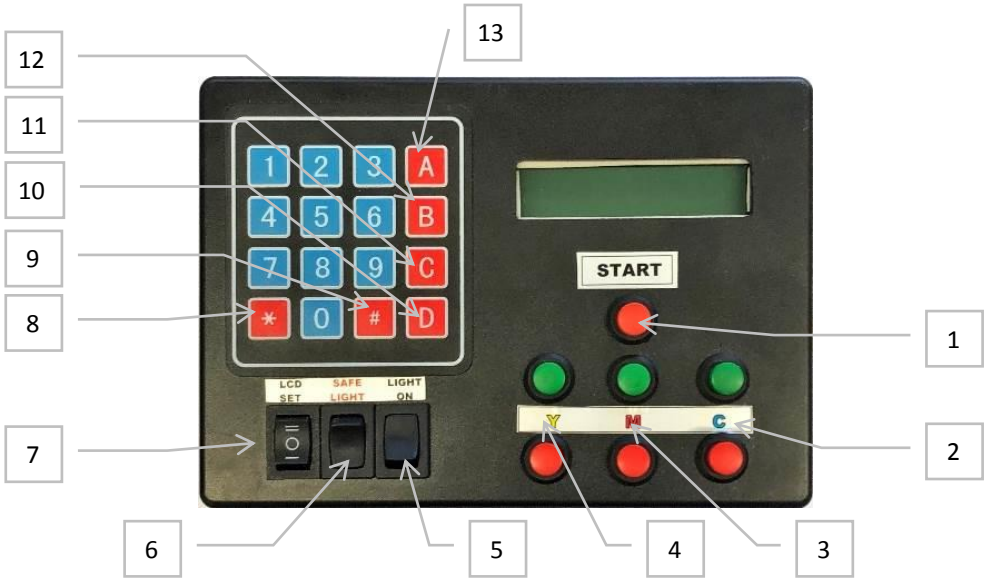
Place the power supply CLOSE to the head of the enlarger. The power sources are mostly not so heavy. Only for power sources under 8A longer cables of 1m may be used. Fix is on a plywood and fix this plywood with a suitable method to the enlarger head.

Connect the DC cables from the power supply to the LED matrix: 5V to “+V” and GND to “-V” (choose the right size: Do not use thin cables!). Connect the 5V and GND coming from the controller too.

Fix an AC cable to the power supply (with common ground if possible) and install a plug.

If you have a multimeter: BEFORE you put the GX16 plug from the controller cable into the controller, power up the power supply and measure the voltage on the plug. You should put the black measuring tip into pin 3 and the red one into pin 1. You should measure something around 4.95-5.2 Volt. If you measure a minus value (-) then you have inverted the V+ and V- or some other cable.

5. Controller use and program instructions



The program start default in color mode (see below point 6 to change this if you want the B/W mode first).

With the 6 buttons (2,3 and 4) you can change the Yellow, Magenta and Cyan values.

INPORTANT: This are absolute color values that go from 0 to 255. This are **NOT** the usual 0-130 or 0-170 density values you find on an enlarger color head as the white balance values are not 0 but around 220 yellow and 170 magenta.

The yellow values are increased by 1 while the magenta values by 5. Empirical tests gave the result that this are the best values. Cyan is increased by 1 as well.

With the switch **5** you can set the light to be permanently on. The light goes otherwise usually on during the operation of the timer.

With the button **1** you start the timer.

The timer time can set with the keyboard (or the buttons 2 in B/W mode). The actual value can be erased with the keypad key '*' (8)

With the switch **6** you can set the safe light (full red). This light is safe for most B/W papers. The red light is only on if

From the default mode you can access the other modes:

- The keypad key 'A' gives access to the YMC manual values enter mode
- The keypad key 'B' gives access to the RGB manual values enter mode
- The keypad key 'C' gives access to the B/W Contrast Control mode
- The keypad key 'D' gives access to the EEPROM memory for writing an retrieving the YMC values. Up to 400 storage places can be set and read.

a. The B/W mode

By pressing the keypad key ‘C‘ from the default (color) mode you have access to the B/W contrast control mode.

The contrast values are going from 0 to 5 in 0.5 steps. Other than the color mode values, the contrast values should more or less be similar to what you find on regular enlargers.

You can increase or decrease the contrast values with the keypad keys ‘A’ or ‘B‘. Or you can use the buttons **4**.

The time is set a usual with the keypad or the buttons **2**.

Keypad key ‘#’ (**9**) key sends you back to the default color mode.

There is a simple split contrast function if you press again the keypad key ‘C‘. This is only a rudimental function that starts the timer twice on the maximal and then on the minimal contrast.

a. YMC manual values mode

By pressing the keypad key ‘A‘ from the default (color) mode you have a mode that allows you to set the YMC values with the keypad.

The timer cannot be operated in this mode. You can set a value from 0 to 255 with the keypad

- The Keypad key ‘*’ key erases the values and sets to 0
- The Keypad key ‘A’ assigns the entered value to the Y channel. This value stays if you change the mode to default.
- The Keypad key ‘B’ assigns the entered value to the M channel. This value stays if you change the mode to default.
- The Keypad key ‘C’ assigns the entered value to the C channel. This value stays if you change the mode to default.
- The Keypad key ‘#’ key sends you back to the default mode

b. RGB manual values mode

By pressing the keypad key ‘B‘ from the default (color) mode you have a mode that allows you to set the color values in RGB format with the keypad.

The timer cannot be operated in this mode. You can set a value from 0 to 255 with the keypad

- The Keypad key ‘*’ key erases the values and sets to 0
- The Keypad key ‘A’ assigns the entered value to the R channel. This value stays if you change the mode to default.
- The Keypad key ‘B’ assigns the entered value to the G channel. This value stays if you change the mode to default.
- The Keypad key ‘C’ assigns the entered value to the B channel. This value stays if you change the mode to default.
- The Keypad key ‘#’ key sends you back to the default mode

c. EEPROM memory storage

By pressing the keypad key ‘D‘ you can store and retrieve YMC values to and from the memory. Up to 400 storage places can be used.

- Set a value from 0 to 400 for the storage place
- The Keypad key ‘A’ writes the actual YMC values to the selected EEPROM storage place
- The Keypad key ‘B’ retrieves YMC values from the selected EEPROM storage place assign these to the actual YMC values
- The Keypad key ‘*’ key erases the values and sets to 0
- The Keypad key ‘#’ key sends you back to the default mode

6. Setup and changes in the code

In order to adapt the software for different layouts and uses you may must change some parameters in the code. Scroll down in the Arduino IDE till the section here below:

```
//-------------------------------------------------------------------------------------------------------------------------
//------------------------- PARAMETERS TO BE CHANGED -----------------------------------------------------------------------

#define NUMPIXELS 256 // Amount of the LED on the matrix. Change it if you have a smaller matrix or daisy chained more
matrix
int operate_mode = 0; // Start mode. Set 0 if you want to start with color mode. Set 3 if you want to start with B/W mode
int no_color = 0; // set this if want to stay only in B/W mode. This is the case if your enclosure is compact.
int rgb_min = 1;  // min brightness for the LCD luminosity switch position 1
int rgb_mid = 20; // mid brightness for the LCD luminosity switch position 2
int color_min = 0; // color for the min position for the LCD luminosity switch position 1 -> 1=green or 0=red
int color_mid = 0; // color for the min position for the LCD luminosity switch position 2 -> 1=green or 0=red
int color_def = 0; // Start color fot the LCD -> 1=green or 0=red

//------------------------- END PARAMETERS TO BE CHANGED -------------------------------------------------------------------
//-------------------------------------------------------------------------------------------------------------------------
```

Important is the value `NUMPIXELS 256` . This is the number of LED you use on the matrix. You can set more LED than you have , but if you set less only a part of the LED will light up. For example, change to `#define NUMPIXELS 64` if you have a 64 LED matrix module. If you use more modules you need to calculate the total and set it as parameter.
Other parameters:

- For starting the controller in B/W mode
  change: `int operate_mode = 0;` to `int operate_mode = 3;`
- For using the controller only in B/W mode without any access to color mode
  change: `int no_color = 0;` to `int no_color = 1;`

With the switch 7 you can set the LCD display on mid or low brightness. The values are stored in the parameter `int rgb_min = 1` or `int rgb_mid = 20;` you may want to change. A value of 50 is max brightness. The lowest still visible light has the value 1.

For the other parameter, see the description in the code.

Save and upload the code to the Arduino in order to activate the parameters.