

Pre-amp Controller

for Mezmerize B1 with muses volume control

Carsten Grønning & Jan Abkjer Tofft | 2020

# Preface

The controller is designed to be able to interface to a Mezmerize B1 board with a Muses 72320 volume control with the support for remote control of volume and input selection etc.

Even though it was designed for a specific purpose the controller board is pretty generic and can be used to control other pre-amplifiers as well with some changes to the source code.

## Features

With the provided source code the board supports a lot of features:

* On board power supply
* IR remote control with learning functionality to be able to use (almost) any remote
* One or two rotary encoders to select volume, input, change settings etc.
* I2C based OLED display
* One extra I2C connector is exposed for future use
* Input selection of the Mezmerize B1 Ten Years After edition board (designed by Nelson Pass and Salas). See <https://diyaudiostore.com/collections/pre-amplifier/products/mezmerize-b1-buffer>
* Volume control via SPI interface to Muses72320 board (designed by Kim Bay Smidt). See <https://www.diyaudio.com/forums/pass-labs/334501-modular-control-unit-b1-rev-2-a.html#post5713852>
* Two separate 12V trigger outputs (to support mono block power amplifiers or other equipment)
* Two separate relay-based switches (to support mono block power amplifiers or other equipment like soft start circuits etc.). It is also possible to measure the voltage across the switch to determine the state of the switched circuit.
* Two analog inputs to measure resistance etc.
* The heart of the circuit is an Arduino Nano V3 – the source code is open source and available on github: <https://github.com/Gninnorg/Mezmerize-B1-Controller>

## The Schematics

The schematics can be downloaded from <https://github.com/Gninnorg/Mezmerize-B1-Controller/blob/master/hardware/schematic.pdf>

The circuit is divided into two parts: the onboard PSU and the controller part.

The PSU part is based on the design of sundberg84 and the board he has designed at <https://www.openhardware.io/view/504/HLK-PM01-breakout-board> to ensure safe usage of the HLK-PM AC to DC module.

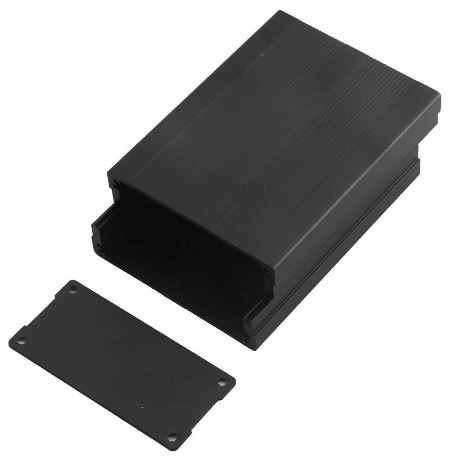
From a schematics point of view the design is identical but as we need 12V for the trigger outputs the HLK-PM12 is used. To supply the Arduino Nano and the rest of the digital part of the controller we have also added an 8V voltage regulator. Finally, two (optional) leds have been added to show if 12V and 8V are on or off.

## The PCB

#### Dimensions

The PCB dimensions is 100 x 71 mm which allows us to order cheap pcbs from one of the providers who offer small prototype batches of 5-10 PCBs for only 2-5$ + shipping costs. We have ordered from and have been fully satisfied with the quality and speed of delivery.

The width of 71 mm has also been selected to allow us to fit the pcb into the slots of a cheap aluminum project box. The purpose of the box is two-fold: as mains voltage is part of the board we want to protect ourselves from getting zapped and also we want the box to act as a shield when the controller is mounted in the same chassis as the B1 Mezmerize. The model we use can be found on ebay.com, aliexpress.com etc. if you search for “100\*76\*35mm aluminum”

As we’ll describe later in this guide we’ll modify the project box to allow access to the connectors outside the box. <TO DO>

The controller board can of course also be used without the project box and mounting holes are available for that purpose. Once again: PLEASE BE CAREFUL WHEN MAINS VOLTAGE IS USED!

<TO DO> Board dimensions and placement of mounting holes

The circuit is divided into two parts: the onboard PSU and the controller part.

The PSU part is based on the design of sundberg84 and the board he has designed at <https://www.openhardware.io/view/504/HLK-PM01-breakout-board> to ensure safe usage of the HLK-PM AC to DC module.

From a schematics point of view the design is identical but as we need 12V for the trigger outputs the HLK-PM12 is used. To supply the Arduino Nano and the rest of the digital part of the controller we have also added an 8V voltage regulator. Finally, two (optional) leds have been added to show if 12V and 8V are on or off.

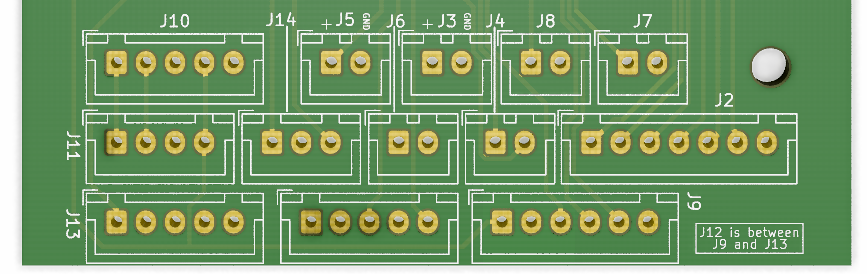
Et billede, der indeholder elektronik, kredsløb

Automatisk genereret beskrivelse

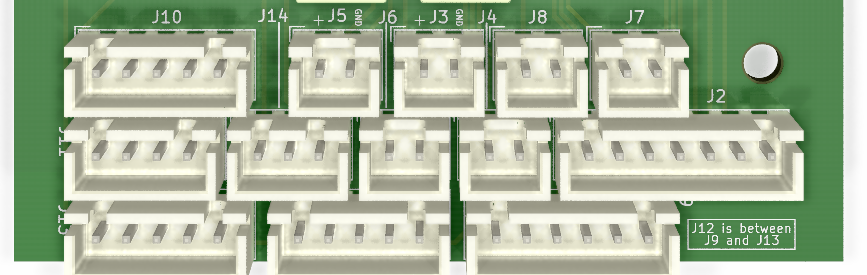
Et billede, der indeholder elektronik, kredsløb

Automatisk genereret beskrivelse

#### Connectors

For all connectors: the pin with a square shape is pin 1. 

If the JST-XH female connectors are used, pin 1 is (still) the leftmost pin of each connector if the board is viewed from the connector end:



**J1 – AC input**

|  |  |
| --- | --- |
| Pin | Description |
| 1 | Live |
| 2 | Neutral |

**J2 – Control of input relays on Mesmerize boards**

|  |  |
| --- | --- |
| Pin | Description |
| 1 | Input 1 |
| 2 | Input 2 |
| 3 | Input 3 |
| 4 | Input 4 |
| 5 | Input 5 |
| 6 | Input 6 |
| 7 | GND |

<TO DO> Show mapping of pins to B1 Mezmerize board

Note: If the connector is used to control relays on other boards than the Mezmerize remember to mount flyback diodes as we do not use the in-built diodes of the ULN2803. The reason is that the on-board relays on the controller and the relays on the Mezmerize do not run at the same voltage. Because of that the COM pin on the ULN2803 can’t be used.

**J3 – 12V trigger output #1**

Please note that J3 and J4 works in unison – if J3 is activated so is J4.

|  |  |
| --- | --- |
| Pin | Description |
| 1 | +12V |
| 2 | GND |

**J4 – Switch #1**

This output can be used as a simple switch (either as a momentary switch or a latching switch depending on the software – in our version of the software it is programmed to be a momentary switch). Please note that J3 and J4 works in unison – if J3 is activated so is J4.

The pcb is designed with the possibility to measure the voltage of the switched circuit. R1 and R2 is a voltage divider to protect the analog input of the Arduino. The default values of R1 and R2 are selected to allow measurement of a voltage up to a bit more than +12V. Do not mount R1 and R2 if you do not need to measure the voltage of the switched circuit and recalculate the values of R1 and R2 if the voltage to be measured is larger than +12V.

|  |  |
| --- | --- |
| Pin | Description |
| 1 | Positive side of the circuit that must be switched |
| 2 | GND side of the circuit that must be switched |

**J5 – 12V trigger output #2**

Please note that J5 and J6 works in unison – if J5 is activated so is J6.

|  |  |
| --- | --- |
| Pin | Description |
| 1 | +12V |
| 2 | GND |

**J6 – Switch #2**

This output can be used as a simple switch (either as a momentary switch or a latching switch depending on the software – in our version of the software it is programmed to be a momentary switch). Please note that J5 and J6 works in unison – if J5 is activated so is J6.

The pcb is designed with the possibility to measure the voltage of the switched circuit. R3 and R4 is a voltage divider to protect the analog input of the Arduino. The default values of R3 and R4 are selected to allow measurement of a voltage up to a bit more than +12V. Do not mount R3 and R4 if you do not need to measure the voltage of the switched circuit and recalculate the values of R3 and R4 if the voltage to be measured is larger than +12V.

|  |  |
| --- | --- |
| Pin | Description |
| 1 | Positive side of the circuit that must be switched |
| 2 | GND side of the circuit that must be switched |

**J7 – NTC measurement L**

J7 and J8 are a bit special and customized to read the resistance of a connected 4.7K NTC resistor placed on the heat sinks in my, Jan’s, Pass F5 Turbo V3 monoblocks. The purpose is to be able to show if the amplifiers have reached their ‘optimal’ working temperature (approx.. 55C) – and to shut them down if they get too hot during the summer ;-) I am fully aware that this is not a precise measurement by any means but it should suffice for the purpose.

|  |  |
| --- | --- |
| Pin | Description |
| 1 | Resistor pin 1 |
| 2 | Resistor pin 2 |

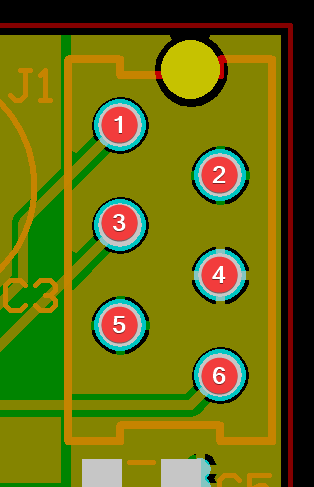
**J8 – NTC measurement R**

See J7 for explanation.

|  |  |
| --- | --- |
| Pin | Description |
| 1 | Resistor pin 1 |
| 2 | Resistor pin 2 |

**J9 – SPI output (used to control the Muses72320 volume control board)**

|  |  |
| --- | --- |
| Pin | Description |
| 1 | +5V |
| 2 | GND |
| 3 | SS |
| 4 | MOSI |
| 5 | MISO |
| 6 | SCK |

If you use the Muses72320 board designed by Kim Bay Smidt the mapping of pins from J9 on the controller board to J1 on the Muses board is:

|  |  |  |  |
| --- | --- | --- | --- |
| J9 on controller board | | J1 on Muses board | |
| Pin | Description | Pin | Description |
| 1 | +5V | 4 | VDD |
| 2 | GND | 2 | GND |
| 3 | SS | 1 | Latch |
| 4 | MOSI | 6 | SDATA |
| 6 | SCK | 3 | SCLK |

Note: pin 5 (MISO) of the controller and pin 5 (GND) on the Muses board are NOT used = unconnected.

**J10 – I2C to control display or another device**

Please note the extra pin that can be used to adjust the contrast if a I2C LCD display is used. As we use an OLED-display we have no use for the contrast adjustment by the extra pin, but it’s there if you need it.

|  |  |
| --- | --- |
| Pin | Description |
| 1 | SCL |
| 2 | SDA |
| 3 | GND |
| 4 | +5V |
| 5 | Contrast control connected to pin D9 on the Arduino Nano |

**J11 – I2C (for future use)**

|  |  |
| --- | --- |
| Pin | Description |
| 1 | SCL |
| 2 | SDA |
| 3 | GND |
| 4 | +5V |

**J12 – Rotary encoder with switch #1**

|  |  |
| --- | --- |
| Pin | Description |
| 1 | SW |
| 2 | GND |
| 3 | CCW |
| 4 | GND |
| 5 | CW |

Used for volume control and menu navigation after activation of the switch

**J13 – Rotary encoder with switch #2**

|  |  |
| --- | --- |
| Pin | Description |
| 1 | SW |
| 2 | GND |
| 3 | CCW |
| 4 | GND |
| 5 | CW |

Used for input selection and standby mode on/off (by activation of the switch)

**J14 – IR receiver**

|  |  |
| --- | --- |
| Pin | Description |
| 1 | Output from IR receiver |
| 2 | GND |
| 3 | +3.3V |

## BOM

<TO DO>

## Build guide

Solder SMT components on the front of the board. The tricky parts are U6 (EEPROM) and even more the MCP23008 (U5) and ULN2803 (U3) because of the small distance between their legs. It is possible to drag-solder the IC’s, but we use hot-air and a solder stencil ordered with the pcbs (cost 7$) as it is much faster, easier and guaranties a nice result.

<TO DO describe direction of LEDS>

Et billede, der indeholder elektronik, kredsløb

Automatisk genereret beskrivelse

Then solder the two SMT diodes on the back of the board.

<TO DO describe direction of the diodes>

Et billede, der indeholder elektronik, kredsløb

Automatisk genereret beskrivelse

Finally, it is time to solder the THT components (but wait a little bit with the connectors).

The thermal fuse breaks at 73C so it needs to be thermal sinked when soldered. One way of doing it is to attach a crocodile clip or two to the leg while it is soldered, but even then, you must solder quickly. It helps that the legs of thermal fuse must be kept pretty long, as you must be able to bend the legs, so the body of the thermal fuse touches the side of the Hi-Link module. For the best result glue it to the side of the Hi-Link module – but do NOT use a glue-gun ;-) Measure the resistance of the thermal fuse with your DMM after soldering it in: it must be close to zero Ohms. If the resistance is infinite, shown as OL on your DMM, the thermal fuse has had too much heat and is broken…

If you prefer it, the thermal fuse can be left out, as it is only an extra safety measure to switch the power to the circuit of, if the temperature of the Hi-Link module rises to an abnormal temperature. If you decide to leave out the thermal fuse you have to solder a piece of wire in its place.

We recommend soldering female headers to the pcb and male headers to the Arduino Nano as shown on the 3D-rendering on the front page of this guide. There are three reasons for this: the most obvious reason is that there are components placed on the board below the Arduino. The other reason is that using female headers allows the Arduino to be plugged in and out in case of failure. The third reason is that a few new Arduinos, Arduino Nano 33 IOT/BLT, has just been released and as they are pin-compatible with the Arduino Nano V3 it opens the possibility for upgrading in the future to allow remote control via Bluetooth or wireless network.

If you want to use the project box shown earlier in this guide you must cut an opening in the front plate and then slide it onto the controller pcb BEFORE the connectors are soldered. That is the only way to have the front plate of the project box placed between the relays and the connectors.

<TO DO insert photo of pcb with the front plate in place>

Et billede, der indeholder elektronik, kredsløb

Automatisk genereret beskrivelse

## Software for the arduino nano

<TO DO>