The Poor Man's Drum Machine
Build Guide
v4.1.0

The Poor Man's Drum Machine is an open source, Arduino based, 16 step DIY drum sequencer. It was initially designed to control the 11 TR-909 drum modules, however, it can be used to trigger any set of analog drum modules.

Features:

- Open source
- Triggers a maximum of 16 drum voices
- High & Low accent, adjustable for each voice
- Shuffle and Flam, for each voice
- Create patterns using Pattern Write mode or Tap Write mode
- Pattern Play mode: create tracks by playing different patterns on the fly
- Create, edit and play rhythm tracks (48 different patterns[1], 224 measures per track)
- Save upto 36 tracks on an EEPROM
- Voice Mute mode: mute and un-mute voices on the fly for extra variation and improvisation
- MIDI in [2]
    - Play a rhythm track that was programmed in a DAW
    - Sync the PMDM to a DAW
- MIDI out [2]
    - Use the PMDM as a Master
    - Export a rhythm track into a DAW as a MIDI track
- Clock with internal clock or with external LFO / MIDI
- External Reset input
- Clock Out output
- Create random rhythm patterns
- Random improvisation on existing patterns
- Can be configured to work with the TR-909 or TR-808 Hihat modules
- Customizable through a simple configuration file

[1] When 11 voices are used. If more voices are used, this number will be lower, if less voices are used, this number can be higher (see below).

[2] MIDI functionality only tested with Ableton Live.

The PMDM is highly customizable. Using a simple configuration file, several paramaters can be adjusted, e.g. the number of drumvoices and the names of the drumvoices. It can be configured to work with TR-909 Hihat module, a TR808 Hihat module or neither of them.

A little on how it works:

The PMDM sequencer is built around an Arduino Nano Every. The Nano Every has the same pinout as a regular Nano v3, but it has more SRAM memory (6KB instead of 2KB).

Two 74HC595 (8 bit shift registers) are used to generate the 5V triggers. One TLC5940 (16 channel PWM LED driver) (U7) is used to drive the 16 LEDs. A second TLC5940 (U4) is used to generate the Control Voltages (CV) for the accents. The PWM signals on the outputs of U6 are low pass filtered and buffered. By using a high PWM frequency (31,4 kHz), the response time of the fluctuating accent CV is short enough to be usable in a drum sequencer.

Using this setup, the PMDM sequencer could trigger a maximum of 16 drum modules (provided that they accept 5V triggers and 0-5V for the accent CV). By default, the PMDM firmware is configured to control 11 drum modules, just like the original TR-909. However, it is quite easy to adjust the circuit as well as the firmware to work with more (upto 16), or less drum modules.

Before you start to build

Before you start to build you have to decide two things:

1) what voltage range do your drum modules need for the accent CV? In many cases a range of 0-5 Volt will be enough. If so, R1...R32 can be omitted and R1...R16 should be replaced by a wire jumper. If 0-15 Volt is required, R1...R32 should be used (see below: Accent Control Voltage).

1) Do you want to use all 16 voices? If not, you can leave out opamp buffers. For instance, if 12 voices are enough for your needs, U4 can be left out, as wel as all the resistors and capacitors that are associated with it. An advantage of using less drumvoices is that you will be able to make drum tracks with more different patterns (see below: Adjusting the number of drum voices)

Configuring the firmware

By default, the firmware is configured to work with the 11 drumvoices from the TR-909 rhythm composer:

Voice   Name
1   Bass drum
2   Snare drum
3   Low Tom
4   Mid Tom
5   High Tom
6   Rim Shot
7   Clap
8   Open Hihat
9   Closed Hihat
10   Ride Cymbal
11   Crash Cymbal

It is easy to increase the number of voices or to change the names of the voices by changing the values in the configuration file: PMDS_config.h. Several other paramaters can also be changed in the configuration file:

-Number of voices
-Number of patterns per track
-Number of measures per track
-Names of the drumvoices
-Brightness of the LEDs for low and high accents
-Size of the EEPROM
-Use a TR-909 Hihat module, a TR-808 Hihat module or neither of them

Adjusting the number of drum voices

The PMDM can control up to 16 drum modules (voices). When using 16 voices, the maximal number of different patterns that a drum track can consist of is 33. If you want to make tracks with more different patterns, you have to reduce the number of drum voices. For instance, if you use 11 drum voices, the maximal number of patterns is 48 (see table 1).

By default, the PMDM firmware is configured to control 11 drum modules. Edit the PMDS_config.h file if you want to use more or less drum voices. Change the values for MaxVoices and MaxPatterns according to the values shown below.

MaxVoices	MaxPatterns	MaxComposition
8	66	224
9	59	224
10	53	224
11	48	224
12	44	224
13	41	224
14	38	224
15	35	224
16	33	224

MaxVoices   Defines the number of voices that you want to use   (default = 11)
MaxComposition   Defines the length (in measures) of a drum track   (default = 224)
MaxPatterns   Defines the number of different patterns in a track (default = 48)

WARNING: When changes are made to these values in the configuration file, the rhythm tracks that were previously saved on the EEPROM will become unusable!

## Changing the names for the drum voices

The names of the drum voices as they appear on the LCD screen can be customized. See the instructions in the configuration file for more information on how to do this.

Using the TR-909 and TR-808 Hihat modules

The TR-909 and TR-808 Hihat modules are special modules that can produce two different sounds (either Open HH or Closed HH sounds).

The TR-909 Hihat module has a trigger input, a CV input and a HIHAT_SELECT input. If HIHAT_SELECT is low (0V), the module produces an Open Hihat sound, when HIHAT_SELECT is high (5V), it produces a closed hihat sound. The TR-808 Hihat module does not have a HIHAT_SELECT pin, instead, it has two trigger pins (one for the OH and one for the CH sound) and a single CV pin.

In the PMDS_config.h file, you can specify wheter you want to use a TR-909 Hihat module, a TR-808 Hihat module, or neither of them. See the instructions in the configuration file for more information. Connect the TR-909 or the TR-808 HH module to the PMDM as shown below. Voice 8 will be OH, voice 9 will be CH.
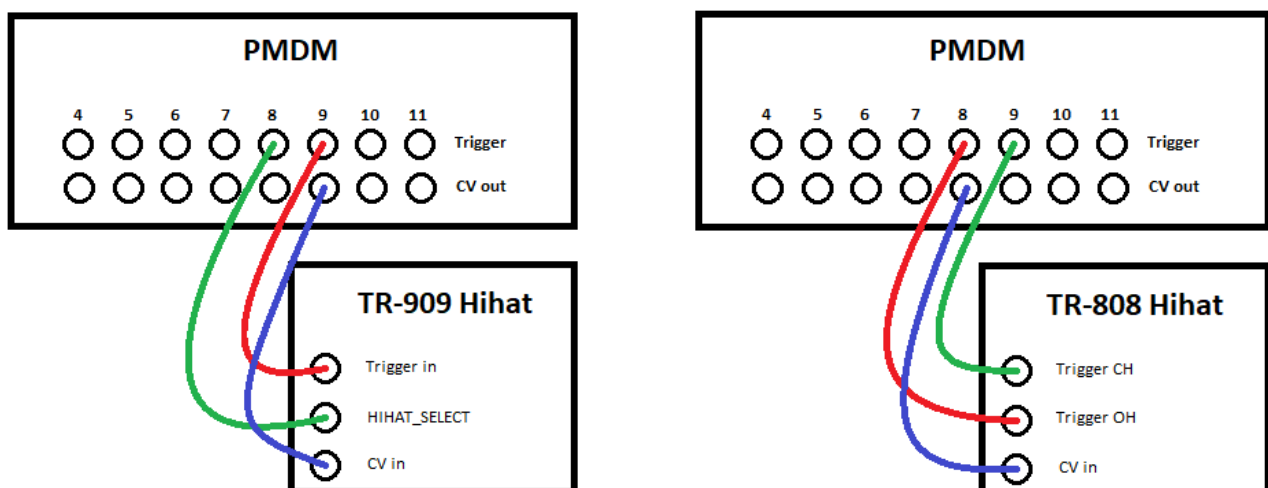
Accent Control Voltage

Most drum modules require an accent control voltage ranging from 0 to 5 Volt. Some drum modules (e.g. the TR-808 drum modules) require CV's ranging from 0 to 15 Volt. When CV's of 0-5 Volt are sufficient, the 20K resistors (R1 to R16) and the 10K resistors (R17 to R32) at the opamp buffers U1 …U4 can be omitted (see fig. 1B). If not, the resistors should be used (see fig. 1A).
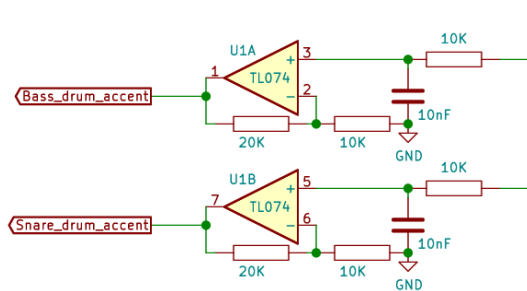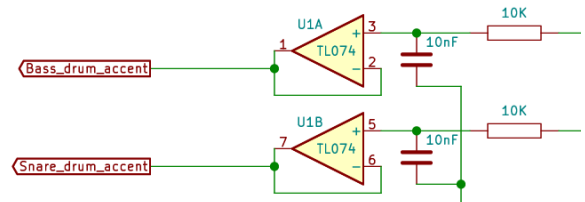

Fig 1A: 0 – 15 Volt accent CV.

Fig 1B: 0 – 5 Volt accent CV. Leave out the 20K resistor and the 10K resistor to ground. Replace the 20K resistor with a wire.

LCD screen

Use a standard 16x2 LCD screen with an I2C interface. You might need to adjust the address of the I2C adapter in the firmware, depending on the type of the I2C interface. The type is written on the chip on the I2C interface.

Type of I2C adapter	address
PCF8574(T)	0X27
PCF8574A(T)	0X37

EEPROM

It is advised to use a 1025Kb EEPROM. Smaller EEPROMS can be used (256 Kb or 512 Kb) but you will be able to store less drum tracks, and you will have to adjust the configuration file (PMDM_config.h).

When using the default values (11 voices and 48 patterns) the numbers of drum tracks that can be saved are as follows:

EEPROM	Drum tracks
1025 Kbit	36
512 Kbit	18
256 Kbit	9

When less voices or less patterns are used, the number of stored drum tracks can be higher. When more voices/patterns are used, the number will be lower. Use the PMPM Calculation Sheet to calculate the maximal number of drum tracks that can be saved on the EEPROM with the settings of your choise.

NB: the EEPROM has to be I2C compatible.

Compiling the sketch

The sketch makes use of two libraries that should be installed in your Arduino IDE:

a) LiquidCrystal_I2C.h   (https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library)
b) Tlc5940_AVR0.h   (https://github.com/ScharreSoft/Tlc5940_AVR0)

Two adjustments should be made in the Tlc5940_config.h file. This file can be found in the Documents\Arduino\ libraries\Tlc5940_AVR0\ folder. Open the file and make these two adjustments:

1: We are using 2 TLC5940s in our circuit: set the number of daisy chained TLCs to 2:

     #define NUM_TLCS 2

2: We need a high PWM frequency: set the Grayscale Resolution to 511:

     #define GRAYSCALE_RESOLUTION 511

NB: because we are using the Tlc5940_AVR0 library, Arduino pin D12 can not be used as a GPIO.

Optional: Remove the warning from the LiquidCrystal_I2C library

When compiling the LiquidCrystal_I2C library by fdebrabander you will get a warning. When you compile the sketch again, the warning will disappear and it will work just fine. However, if you want to get rid of the warning proceed as follows:

1) Go to the directory of the LiquidCrystal_I2C library. You can find it in the ..\documents\Arduino\libraries folder

2) Open the file LiquidCrystal_I2C.h

3) Locate this section:

#define En B00000100 // Enable bit
#define Rw B00000010 // Read/Write bit
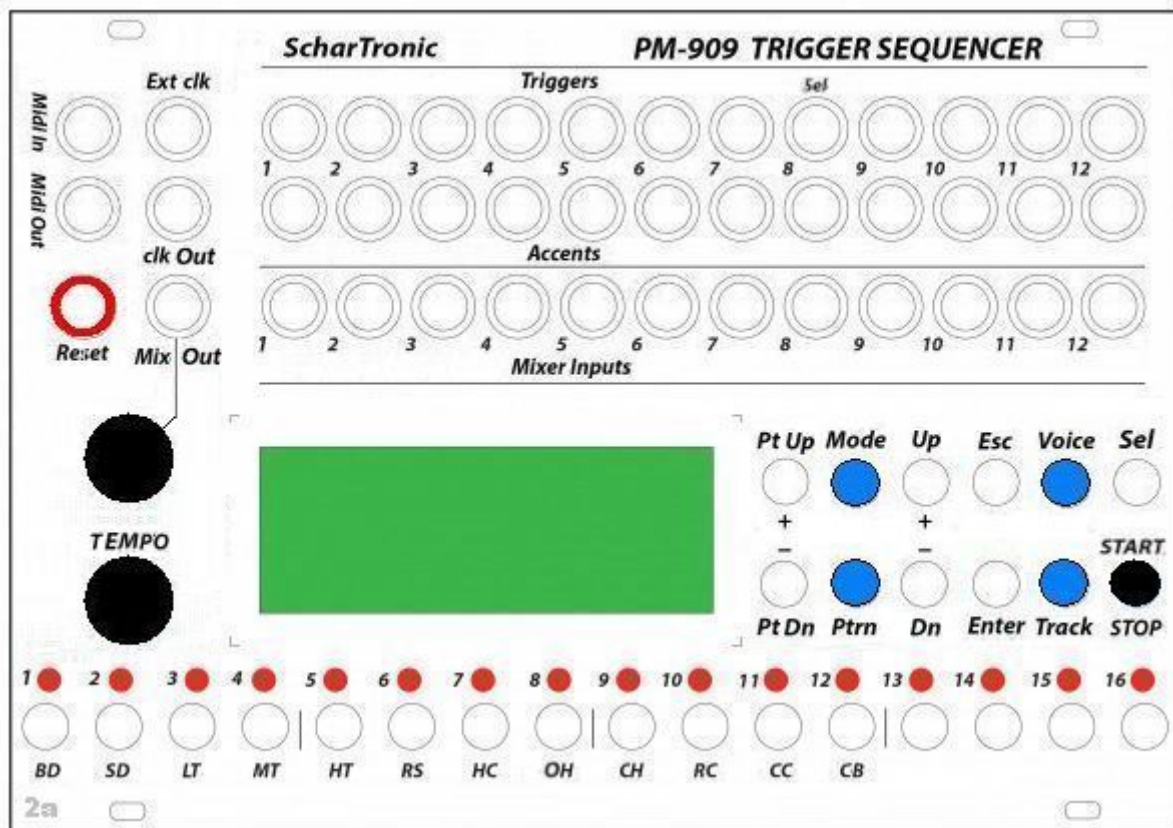#define Rs B00000001 // Register select bit

4) Replace it with this:

#define En 0b00000100 // Enable bit
#define Rw 0b00000010 // Read/Write bit
#define Rs 0b00000001 // Register select bit

5) Save the file.

Now the warning should disappear.

IMPORTANT!

It is highly advised to use an appropriate heat sink for the LM7805. Depending on the current that your LEDs draw, it can become quite hot.

A 36HP panel for Eurorack, designed by ModWiggler members Ayab and KSS.

More info:
https://modwiggler.com/forum/viewtopic.php?t=258027
https://github.com/ScharreSoft/The-Poor-Mans-Drum-Machine
https://www.youtube.com/channel/UCwJOUJiDiWwPXO-47SPIHYQ