mddp128u P/N MIDI-to-Parallel decoder *(for units produced since 1 Jan 2016!)* * user's guide *

1. Purpose and description

mddp128u MIDI–to-parallel decoder unit is new generation MIDI decoder, designed for controlling various objects (solenoids, lamps etc.) in various applications, by MIDI Note On/Off messages. Compared to its predecessors it has smaller size, weight and cost, faster response, all-in-one firmware, wider number of settings.

The decoder consists of one or two printed circuit boards per model, stacked together. The lower one is the main board where the processing micro controller is located together with MIDI interface and power regulating parts. It contains 8 driver arrays and connectors for 64 loads. The upper board (where available) is extension to the main board, containing 8 additional driver arrays and connectors for 64 more loads. The two boards are stacked together in normal use for making the unit more compact. When necessary the two boards can be "spread" so that they are entirely accessible. This may be suitable for testing purposes, but if necessary, the unit may work reliably in "spread" position as well.

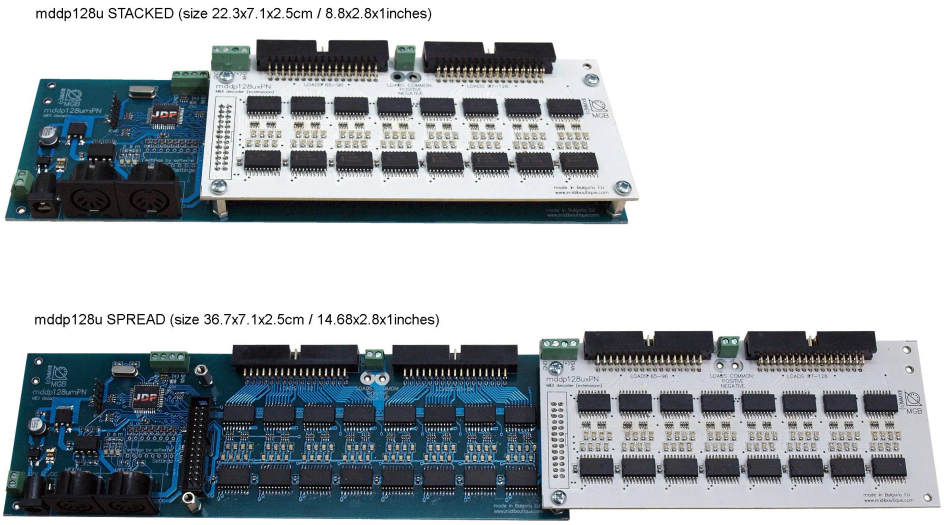


Fig.1 **mddp128u** - stacked vs. spread

mddp128u can be equipped with N-Darlington drivers for controlling loads wired to Common Positive line (*N-version*), or P-Darlington drivers for controlling loads wired to Common Negative line (*P-version*). The "P" or "N" capital letter in its name indicates the type of drivers used.

Decoder and driver boards can be stacked/spread in various combinations, for better space use.

mddp128u can be ordered with as many outputs as necessary (32, 64, 96 or 128), so that it comes as functional replacement of our older mdec32br-P/N, mdec64br-P/N, mdec96br-P/N and mdec128br-P/N decoders series. The unit responds to ***NoteOn***, ***NoteOff*** and ***AllNotesOff*** MIDI messages on user-selectable MIDI channel and ignores any other MIDI messages on the same MIDI channel and all channel messages on other MIDI channels, also Real time and SysEx messages (except product-specific SysEx message format, described in ***Appendix H***). It covers user-defined contiguous or non-contiguous range of notes (per mode) coming on user defined single or multiple channels (per mode). It runs all-in-one firmware, supporting

former *Standard*, *SAM, a*nd *PWM*&Pulse firmwares together. *In version 1.2016 two newer modes have been added, named PWF and SCN, described later on.* The user can select the mode they prefer with no need to place special order or request for particular firmware for each of above modes.

The hardware design is based high-performance micro controller, running original **mddp128u** (v.2016) firmware. It allows discrete (on/off) or regulated (by PWM or pulse width) MIDI control of up to 128 external loads.

The unit has opto-isolated MIDI input and standard MIDI-through output, allowing daisy-chaining of multiple units as well as inserting the unit in existing MIDI systems without need of additional interfacing devices like Splitters, Mergers, etc. The unit has on-board diode bridge and voltage regulator, so it can be powered from any AC/DC power supply adapter capable to source current of 500mA or more at typical voltage of 12 VAC/DC.

All settings such as Board Outputs number, Mode (former Firmware version), MIDI Channel, Starting Note, response Curve, Time-related parameters, PWM-related parameters and Resolution-related parameters, Channel/Note per output, are user-selectable by means of device-specific SysEx message, sent by ***mddp128uset.exe*** (v.2016) - Windows-based software application, which is free downloadable from our site. The format of this SysEx message is described in ***Appendix H,*** so that it can be implemented in third-party-built applications for Mac OS or other software platforms.

Three on-board LEDs indicate the device status:

"P" - Indicates power, lit when unit is powered;

"S" - Indicates normal function by pulsing;

"M" - Indicates any received MIDI message.

Appendices A,B,C,D,E and ***F*** show how external loads (electromagnetic valves, lamps, LEDs or SAM-operated switches) should be wired to the unit in case of Common Positive (N-drivers) and Common Negative (P-drivers). Note that there are integrated suppression diodes for each output driver, so no additional protection diodes are necessary when using it with inductive loads. An external power supply is necessary in any case, that is capable to source sufficient current at proper voltage for driving the loads. It is possible that the decoder board shares same power supply with loads.

2. Connection

There are few connectors/screw-terminals that are used for connecting the unit to external hardware. The connectors are functionally labeled on PCB. There are up to four 34-pin header connectors (BH34 type) for loads, as well as few screw terminals for connections to external power sources. Their use is illustrated for any case in wiring diagrams at the end of this document. There are two DIN-5 connectors for MIDI In and MIDI Out, and finally, there is coaxial connector for powering the board itself. It is doubled by screw terminal making easier to power the board from supplies without jacks, or spreading the power from same power supply to more than one board without need of jacks.

The input MIDI signal should be applied to MIDI In. The MIDI Out would repeat exactly any received MIDI byte/message, with minimal delay, without any processing.

The unit itself (without loads attached) can be powered by any type of wall-wart adapter with output voltage AC/DC between 7 and 24V (recommended 12VDC), capable to source at least 500mA of current. When no outputs are engaged, the unit would draw as small current as 50mA. During use the current it draws may increase up to 400mA, depending on the number of currently active outputs, since each output has on-board LED that draws current of few mA when lit.

When using DC adapter, the polarity does not matter as there is on-board diode bridge that maintains correct polarity.

Unlike the board itself that would accept AC power from transformer, the loads must be *always powered from **DC** source*. Before applying power to loads, please make sure that everything is wired in accordance with relevant wiring diagram!

3. Settings

All settings are done using ***mddp128uset.exe*** (v.2016) – a free-downloadable Windows application (*please read the note in red below!*).

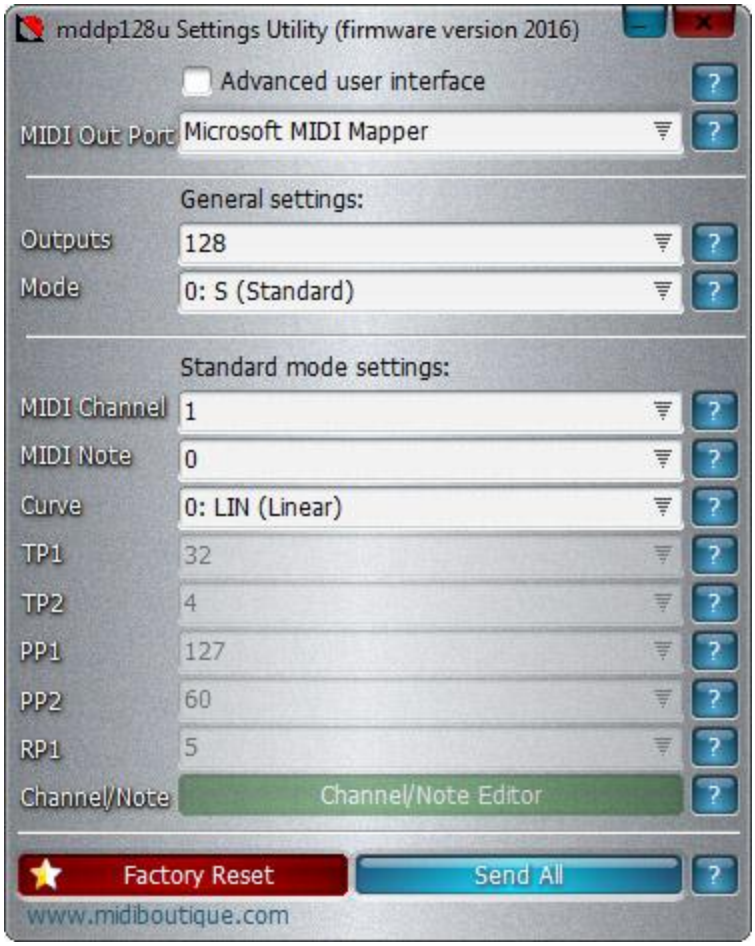


Fig.2 mddp128uset (v.2016) – screen shot

- **Outputs**

mddp128u can be ordered with various number of outputs. This number can be 8,16,24,32 etc., up to 128 in steps of 8. The time- related parameters are calibrated separately per each possible number of outputs. For best performance you should set Board Outputs parameter to be equal to the real number of outputs available on your board. Setting it to smaller value would make part of outputs non-responsive, setting it to bigger number would decrease board's performance. *This parameter is relevant to all modes, so please keep its value correct*!

- **Mode**

mddp128u supports the functionality of previously supported by three separate firmwares which were available only separately with its predecessors of **mdecXXXbr** series: Standard, SAM and PWM (where the last one supported PWM and Pulse output). **mddp128u** (v.2016) has 6 user-selectable modes, four of them covering the functionality of above former **mdecXXX** firmwares. All the six modes are explained in details in next chapter.

- **MIDI Channel**

This defines the MIDI Channel on which the unit will be listening. Only NoteOn/NoteOff/AllNotesOff, messages received on this particular channel will be responded to. Regardless this setting, the unit will re-transmit whatever MIDI messages it receives byte-by-byte, without any processing, so that other decoders

listening on other channels can be chained on same MIDI line. *The MIDI Channel parameter is relevant to all modes except SCN mode.*

- **Start Note**

Defines the note number assigned to first output. In case of SAM firmware defines the note number assigned to first couple of outputs since with SAM firmware each SAM switch takes two outputs (first/odd for ON solenoid, second/even for OFF solenoid). *The Start Note parameter is relevant to all modes except SCN mode.*

- **Curve Parameter**

This parameter is used in velocity-related modes VPWM and VP, where the *PWM duty* (in VPWM mode) or the *Pulse width* (in VP mode) is being controlled by note Velocity. The Velocity parameter of received and recognized MIDI note is mapped thru one of the available response curves for getting natural results in case of controlling solenoid-driven percussion etc. objects with non-linear mechanical response. Five predefined standard curves and one custom curve are available in firmware version 2016. The custom curve can be modified per request. All curves are illustrated in **Appendix G.** *The Curve parameter is relevant to velocity sensitive modes (VP and VPWM) only.*

- **TP1 and TP2: Time-related parameters**

These parameters are used for different purposes in SAM, VP and PWF modes. Their meaning is described per each mode below. *The TP1 is relevant to SAM and PWF modes only, the TP2 parameter is relevant to SAM, VP and PWF modes only.*

- **PP1 and PP2: PWM-related parameters**

These parameters are used in for different purposes in VPWM and PWF modes. Their meaning is described per each mode below. *The PP1 and PP2 parameters are relevant to PWF mode only.*

- **RP1: Resolution-related parameter**

This parameter is used controlling the bit resolution in PWM-related modes VPWM and PWF. Its meaning is described per each mode below. *The RP1 parameter is relevant to VPWM and PWF modes only.*

All above parameters are user-configurable using ***mddp128uset.exe*** (v.2016) software utility. The use of this utility is described step-by-step in ***Appendix I.*** Their values are kept in non-volatile memory even when **mddp128u** is not powered. The allowed value ranges and default values of all parameters are listed in ***Appendix H***, ***Table 2*****.**

NOTE (01.2017): ***mddp128uset.exe*** *programming utility is not supported anymore. Please consider* ***UpSet.exe*** *utility.*

4. Modes

- ***Standard mode - Mode#0***

This mode is used for controlling static loads of any kind: lamps, LEDs, solenoids, relays etc. Possible applications are: pipe organ valves or solenoids, light shows, water shows, fireworks, etc. The unit will respond to NoteOn, NoteOff messages in contiguous note range, starting at user-selected *MIDI Note*, assigned to first output, as well as to AllNotesOff, all received on user-selected *MIDI Channel*.

Parameters relevant to Standard mode: MIDI Channel and MIDI Note.

- ***SAM mode - Mode#1***

This mode is intended for controlling Stop-Action Magnet (SAM) -switches. These are widely used with organ draw stops or motorized stop controls. Each SAM takes two consecutive outputs, the first (odd) one for controlling SAM's ON solenoid, the second (even) one – controlling SAM's OFF solenoid. For example, the first SAM will take outputs #1 and #2, the second - outputs #3 and #4 etc. Output#1 (as well as any output with odd number) will control SAM's ON solenoids in response to assigned NoteOn message, output#2 (as well as any output with even number) will control

SAM's OFF solenoid in response to NoteOff message. Each couple of outputs will respond to same Note number, so that the controlled SAM switch will move to ON position upon NoteOn and move to OFF position upon NoteOff of same note number. Therefore, an 128-output unit is able to control up to 64 SAM switches in response to 64 consecutive note numbers. The note number controlling the first couple of outputs is defined by *MIDI Note* parameter. The unit will be listening on user-selected *MIDI Channel*. The width of output pulses controlling SAM solenoids is equal to *TP1*TP2* in milliseconds. Upon AllNotesOff all SAMs will be moved to OFF position.

Parameters relevant to SAM mode: MIDI Channel, MIDI Note, TP1 and TP2.

- ***VP (velocity-controlled pulse) - Mode#2***

This mode is used for controlling solenoid-driven percussive instruments like chimes, bells, calliopes etc. Because of the mass/inertia of their striking mechanism, applying shorter or longer pulse may produce weaker or stronger hit. The unit outputs single pulses upon NoteOn message, with pulse *length proportional to MIDI note Velocity value, mapped thru selected Curve, then scaled (multiplied) by TP2 parameter. The resulting pulse will be long MV*TP2 mS, where MV is the Velocity value, mapped thru selected Curve*. If NoteOff arrives before the end of the pulse, the pulse is canceled immediately. If NoteOff arrives after the end of the pulse, nothing happens.

Parameters relevant to VP mode: MIDI Channel, MIDI Note, Curve and TP2.

- ***VPWM mode - Mode#3***

This mode is used for controlling solenoid-driven percussive string instruments, mainly pianos. Upon NoteOn the unit will output pulse-width modulated wave with constant period and *duty cycle proportional to Velocity value, mapped thru selected Curve*. The PWM frequency should always be kept at possible maximum. Since it is software-generated, it depends on the number of outputs and the velocity bit resolution, set by *RP1*. The more outputs - the lower PWM frequency. The higher resolution - the lower PWM frequency. Having the number outputs fixed for particular board, user can vary the *RP1* parameter to achieve higher frequency with smaller resolution or bigger resolution at lower frequency. The settings may need adjustment for finding the optimal values for frequency and resolution per each particular type of loads. Table 1 shows the relation between outputs number, PWM resolution (controlled by *RP1*) and the PWM frequency.

Table 1. PWM frequency per **mddp128u** Outputs number and PWM resolution

Outputs	**PWM resolution,controlled via RP1, bits(levels)**						
	1(2)	**2(4)**	**3(8)**	**4(16)**	**5(32)**	**6(64)**	**7(128)**
8	10200Hz	5102Hz	2551Hz	1276Hz	641Hz	316Hz	160Hz
16	5814Hz	2907Hz	1429Hz	724Hz	367Hz	181Hz	91Hz
24	4098Hz	2049Hz	1020Hz	510Hz	255Hz	128Hz	64Hz
32	3145Hz	1562Hz	781Hz	390Hz	195Hz	96Hz	49Hz
40	2551Hz	1276Hz	625Hz	316Hz	156Hz	79Hz	39Hz
48	2128Hz	1064Hz	537Hz	263Hz	135Hz	67Hz	33Hz
56	1838Hz	926Hz	463Hz	231Hz	114Hz	57Hz	28Hz
64	1613Hz	806Hz	403Hz	201Hz	100Hz	51Hz	25Hz
72	1471Hz	735Hz	367Hz	178Hz	91Hz	45Hz	22Hz

80	1316Hz	658Hz	329Hz	164Hz	82Hz	41Hz	20Hz
88	1190Hz	609Hz	305Hz	149Hz	74Hz	37Hz	18Hz
96	1087Hz	556Hz	275Hz	138Hz	69Hz	34Hz	17Hz
104	1000Hz	500Hz	250Hz	125Hz	62Hz	31Hz	16Hz
112	961Hz	463Hz	238Hz	119Hz	59Hz	30Hz	15Hz
120	893Hz	446Hz	223Hz	109Hz	54Hz	28Hz	14Hz
128	833Hz	416Hz	208Hz	104Hz	52Hz	26Hz	13Hz

Parameters relevant to VPWM mode: MIDI Channel, MIDI Note, Curve and RP1.

- ***PWF mode - Mode#4***

This mode mode is used for controlling high-power solenoids. They require that upon NoteOn certain higher, user-defined amount of power (a.k.a. attack power) is applied initially on the load for user-defined fixed time (a.k.a. attack time), then power level is decreased for user-defined lower value (a.k.a. sustain power) until Note Off is received. The attack time, attack power and sustain power are user-configurable. In this mode they are controlled via *TP1* and *TP2* (attack time is approximately *TP1*TP2* mS), *PP1* (attack power) and *PP2* (sustain power). In fact the attack and sustain power are regulated by PWM, so that *PP1* and *PP2* actually control the PWM duty cycle for attack and sustain phases accordingly. The PWM frequency depends on number of outputs and *RP1*. Having outputs number fixed for each particular board, user can adjust the PWM frequency by finding optimal PWM resolution. The PWM resolution is controlled via *RP1* parameter. The relation between outputs number, PWM resolution and PWM frequency is shown in Table 2.

Parameters relevant to PWF mode: MIDI Channel, MIDI Note, TP1, TP2, PP1, PP2 and RP1.

- ***SCN mode - Mode#5***

In this mode the unit perfowms as in Standard mode (Mode#0), but ignoring *MIDI Channel* and *Start Note* parameters. Instead them, it uses user-assigned *MIDI Channel* and *Note* per each output. This allows assigning non-contiguous note ranges or different MIDI channels per user-defined groups of outputs. More than one outputs can be made responding to same *MIDI Channel*/*Note* combination.

- ***ROU mode - Mode#6***

In this (Router) mode, user can define one or more routing rules to control the way input notes control load outputs. It is suitable for implementation of Stop, Coupler, Transposer etc. organ specific functions, but not limited to this. User can define up to 32 rules.

Each rule has following parameters:

Control MIDI Channel and *Control Note*. The rule can be turned on/off during performance by sending NoteOn/NoteOff with <this> Note number on <this> Channel.

Input MIDI Channel - only notes received on <this> channel will be processed by this rule.

Process Start Note, Process End Note - All received notes within the range desined by <these> two note numbers will be processed by this rule.

First Output - the output associated with first note of above range. This defines the range of outputs mapped to above note range.

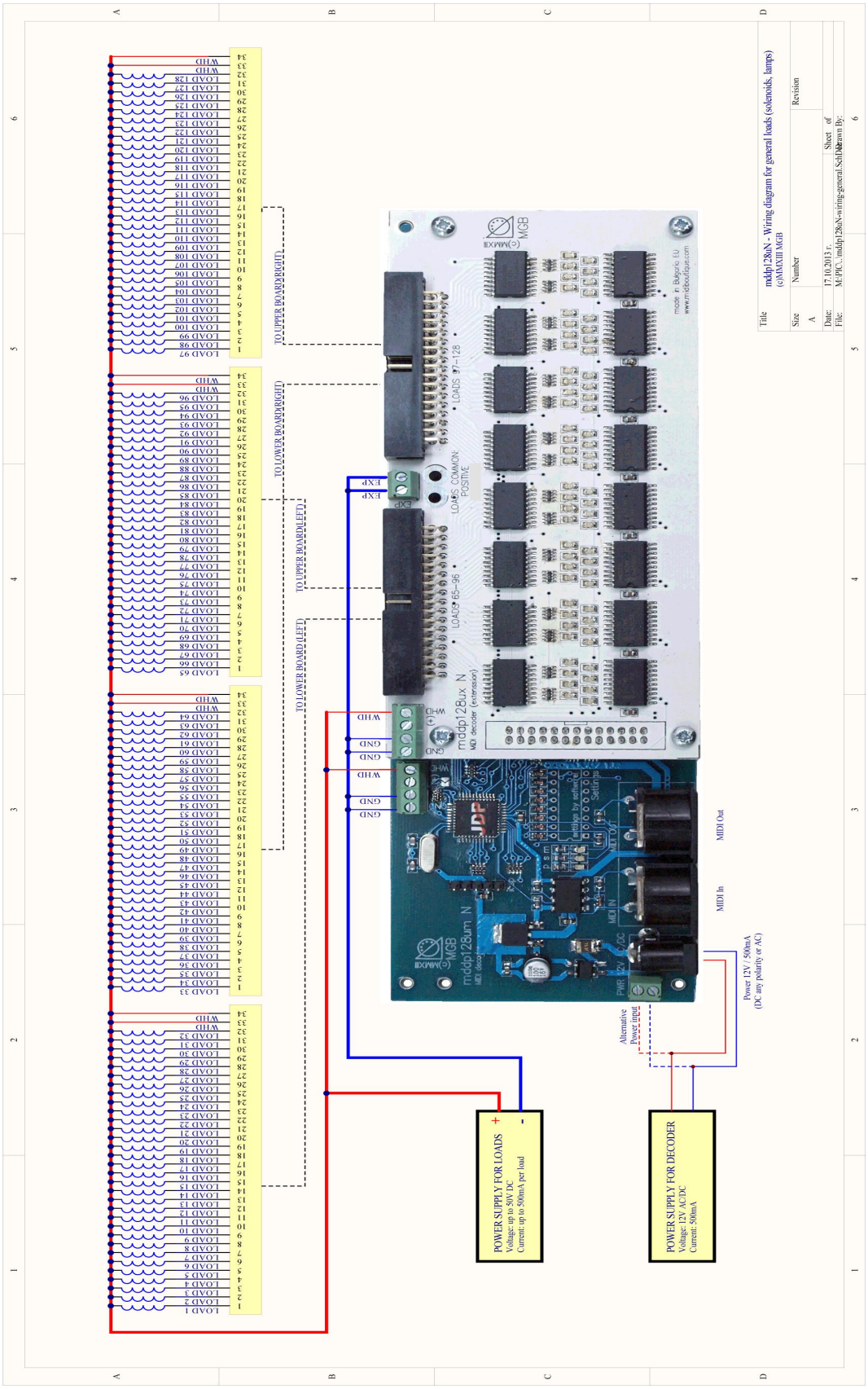
Rules can overlap on input note ranges, channels and output ranges, allowing various combinations. Any number of rules can be activated/de-activated at a time.

*NOTE: ROU mode can be configured only by using **UpSet.exe** progframing utility, downloadable from our siute for free. The **mddp128uset.exe** utility is discontinued and does not support ROU mode.*

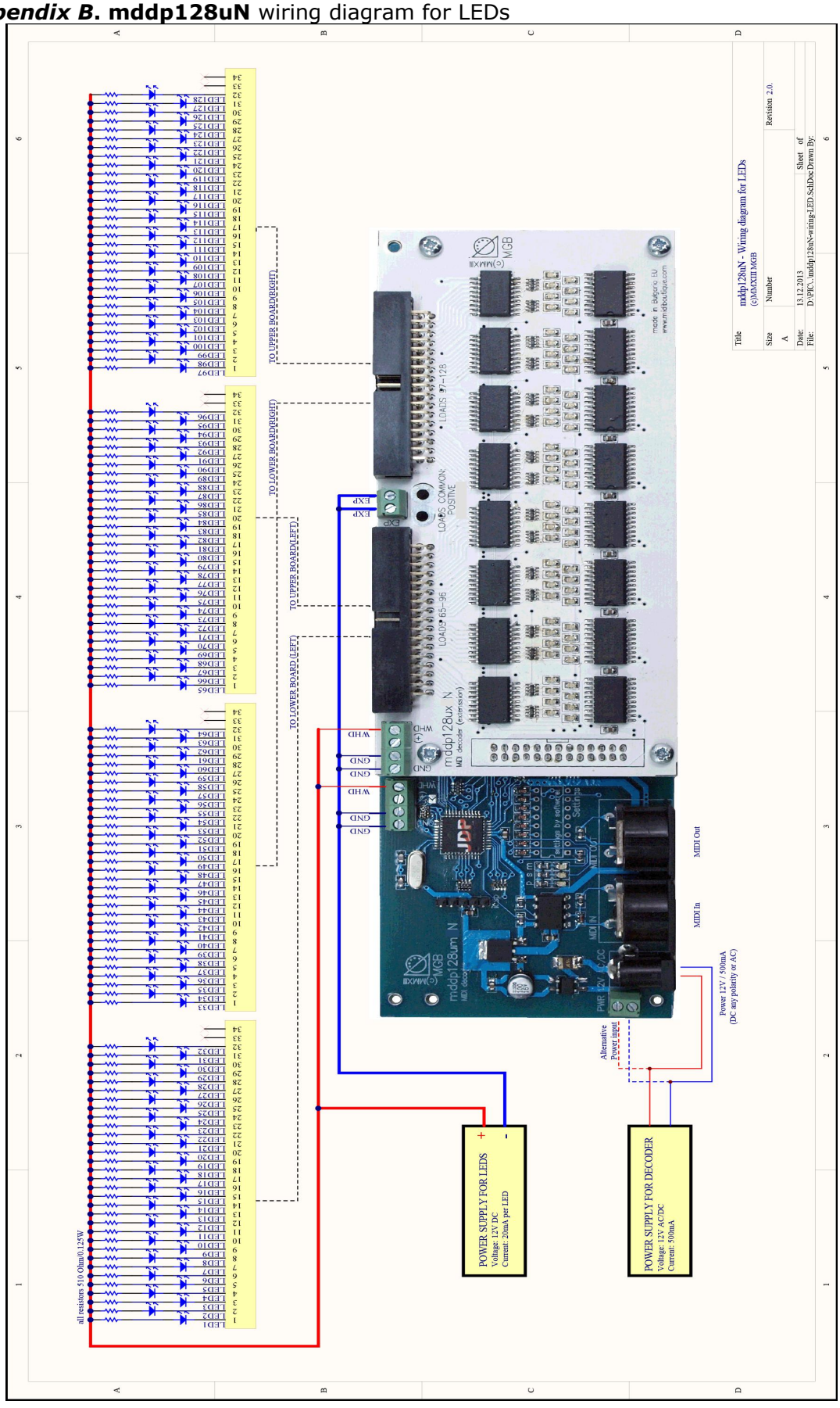
5. Device specifications

- Size: 22.3 x 7.1 x 2.5 cm (8.8 x 2.8 x 1 inches);
- Weight: 170g (6 oz)
- Power supply: AC/DC 12V/500 mA;
- Standard opto-isolated MIDI input;
- Buffered MIDI-through output;
- up to 128 parallel P- or N- type Darlington outputs for max.50V/max.500mA each, with integrated suppression diodes;
- MIDI note range: up to 128 (up to 64 for SAM mode) user-selectable contiguous notes, or user-defined list of notes (SCN mode only);
- Recognized MIDI messages: *NoteOn*, *NoteOff*, *AllNotesOff* on selected channel plus device-specific SysEx message used for Settings. Running status supported. NoteOn with Velocity = 0 recognized as NoteOff. All other messages are ignored. Any received MIDI byte is immediately re-transmitted without any processing/change;
- MIDI Channel – user selectable;
- Two Time-related parameters with various meanings – user selectable;
- Two Pulse-related parameters with various meanings – user selectable;
- One PWM Resolution-related parameter - user selectable;
- Velocity response Curve - user selectable.

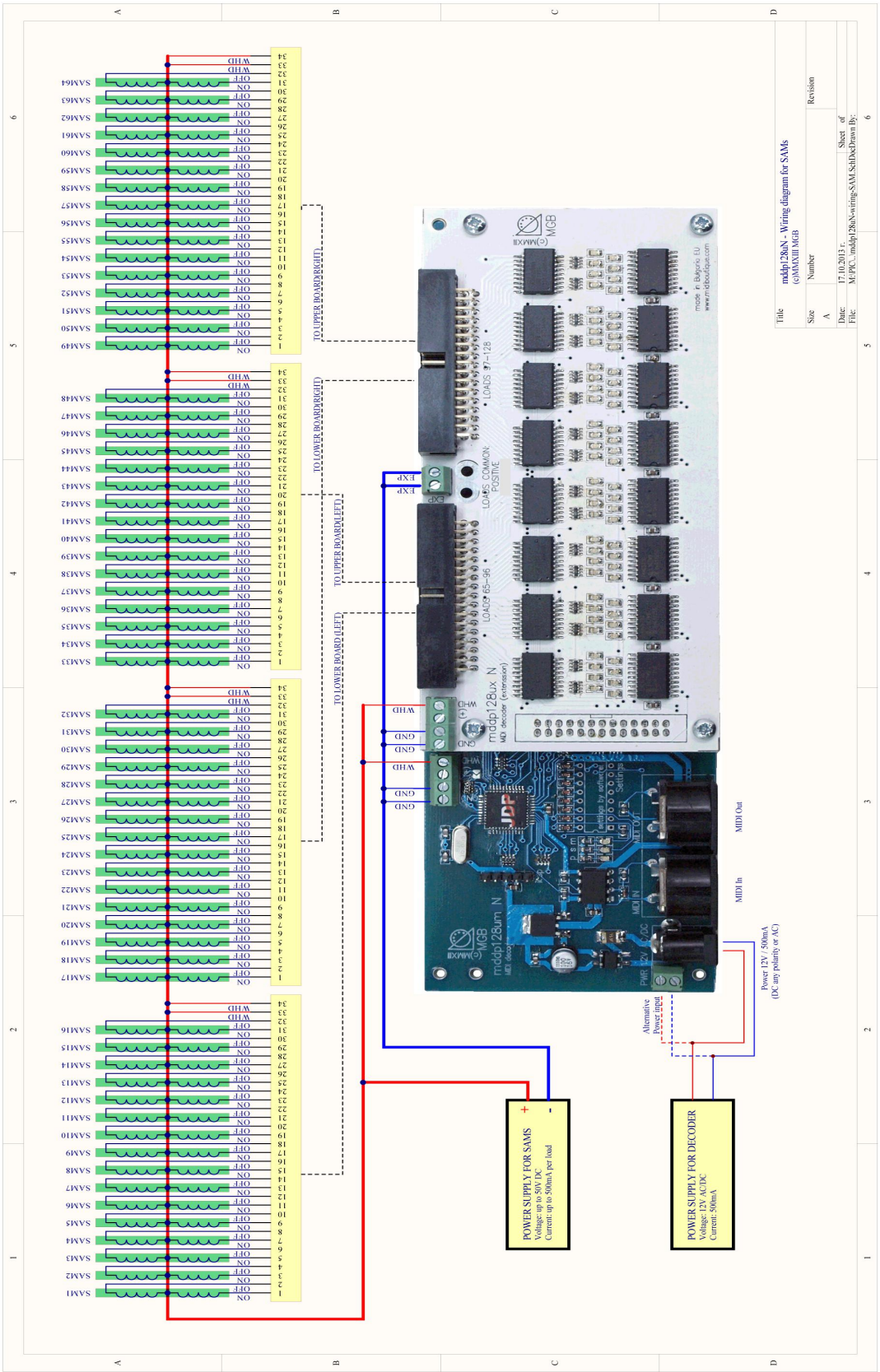
***Appendix A*. mddp128uN** wiring diagram for general loads: solenoids, lamps etc.



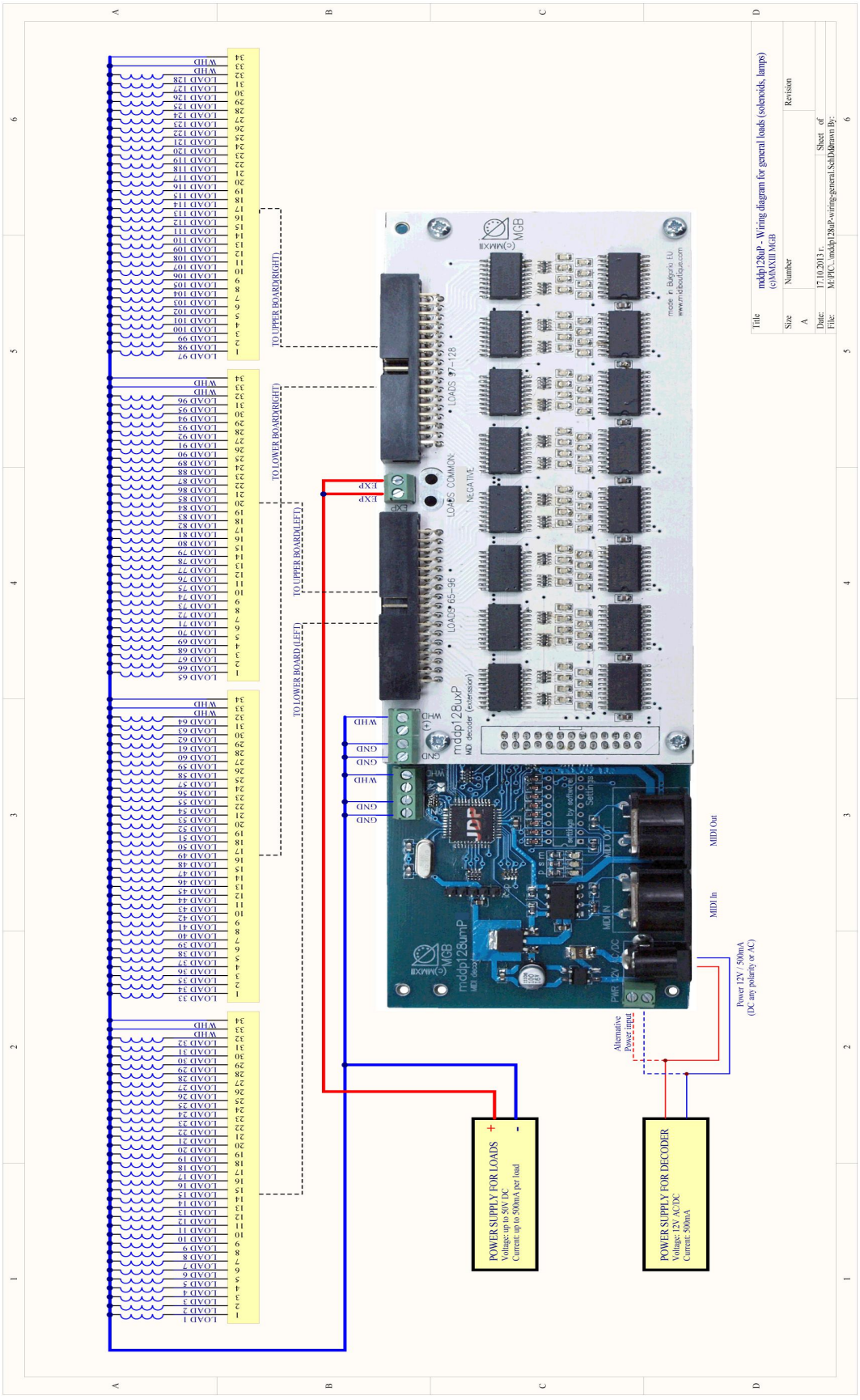
***Appendix B*. mddp128uN** wiring diagram for LEDs



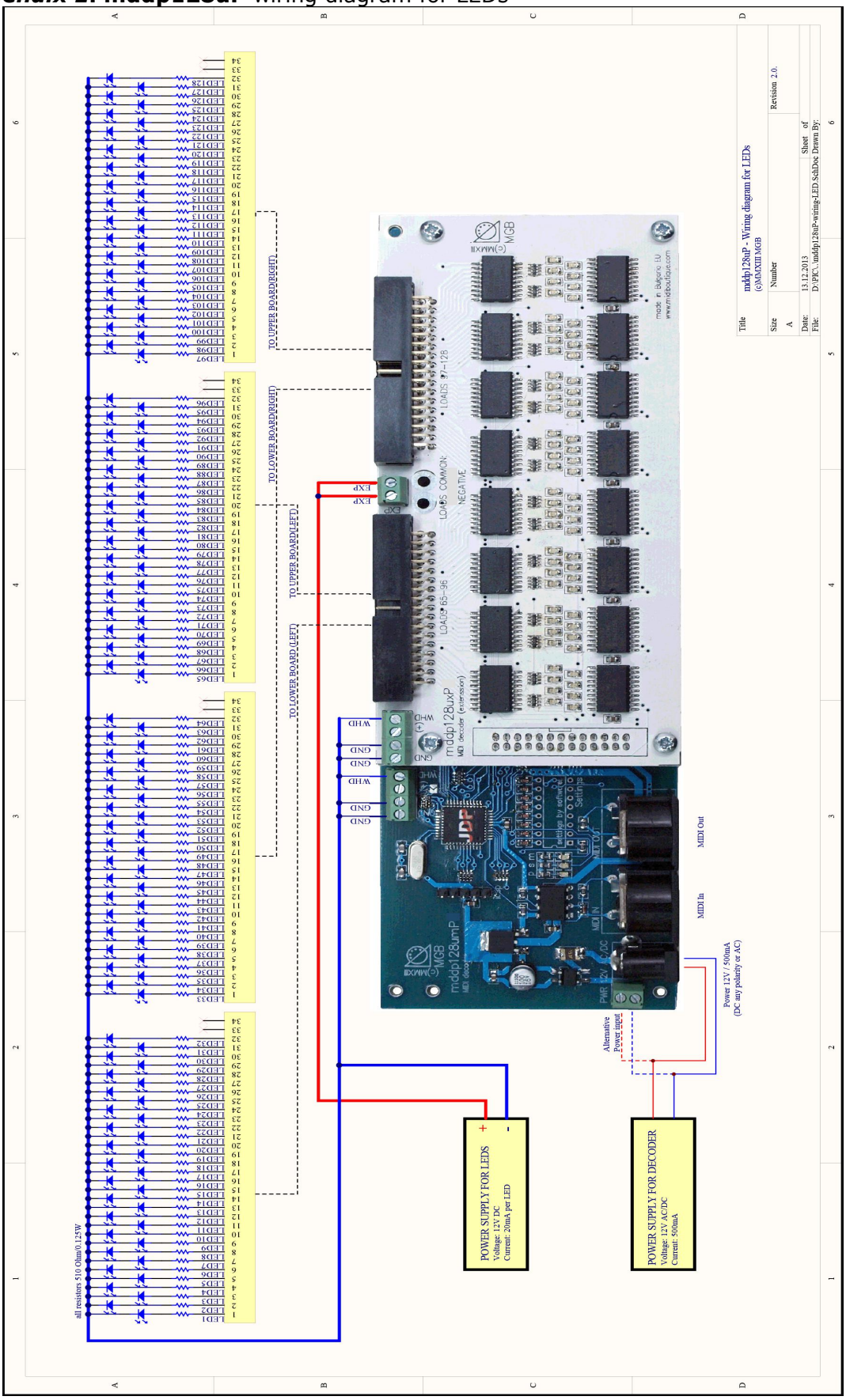
Appendix C. **mddp128uN** wiring diagram for SAM – operated switches/draw stops



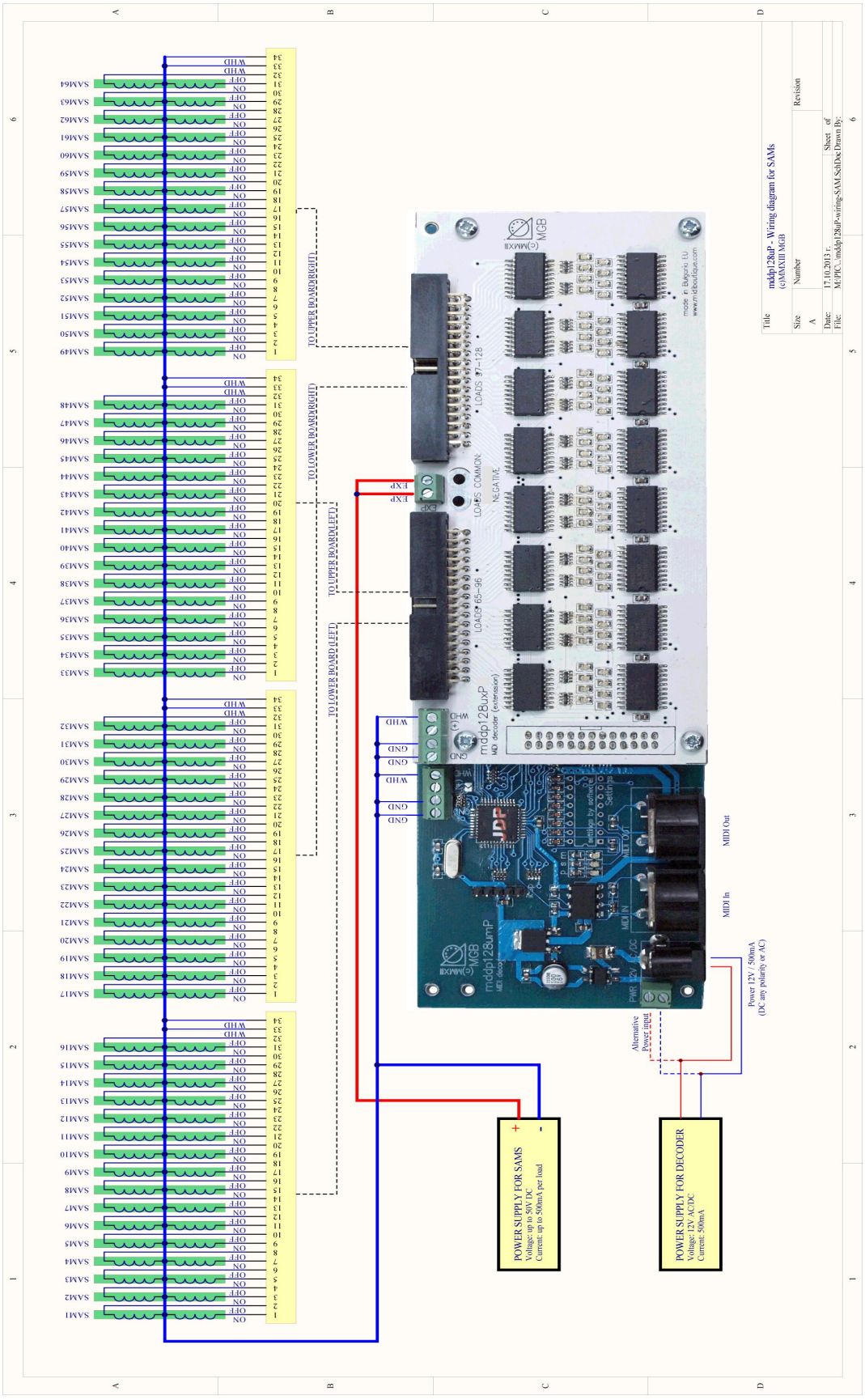
***Appendix D*. mddp128uP** wiring diagram for general loads: solenoids, lamps etc.



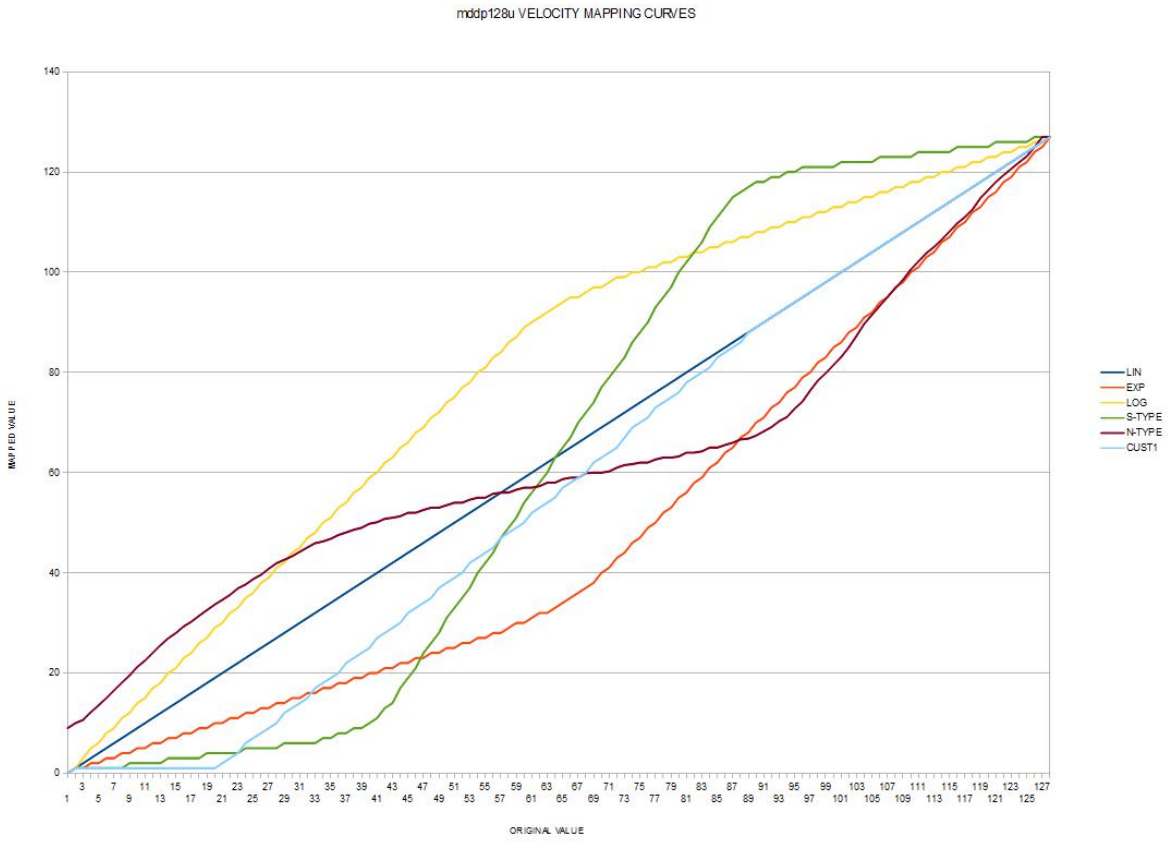
***Appendix E*. mddp128uP** wiring diagram for LEDs



***Appendix F*. mddp128uP** wiring diagram for SAM – operated switches/draw stops



***Appendix G*. mddp128u** velocity mapping curves



***Appendix H*.** SysEx configuration message format description (all numbers shown in Hexadecimal format)

F0 - SysEx start
00 - mandatory
21 - MIDI Boutique ID
7F - MIDI Boutique ID
00 - Device ID MSB (mddp128u ID)
16 - Device ID LSB (mddp128u ID)
00 - Device sub-ID (message ID = 00 – Program Settings v.2016)
p1 - Parameter number MSB (00..1F)
p2 - Parameter number LSB (00..7F)
v1 - Parameter value MSB (00..01)
v2 - Parameter number LSB (00..7F)
F7 - SysEx end

Parameter number is 14-bit value split in two 7-bit values carried by two separate bytes.
Parameter value is 8-bit value slit in one -1bit and one 7-bit value, carried by two separate bytes.
The parameter numbers, value ranges and default factory values are shown in *Table 2.*

Table 2. Parameters numbers, ranges and default values

Parameter name	Parameter number (HEX)	Value range (HEX)	Factory value (HEX)	Comment
Mode	$0	$0..$5	$0:Standard	$0 means Standard $1 means SAM $2 means VP $3 means VPWM $4 means PWF $5 means SCN *$FF (DEC 255) will force the unit to reset to factory defaults*
MIDI Channel	$1	$0..$F	$0:Channel#1	MIDI channel for all outputs. $0 means Channel#1 $1 means Channel#2 .. $F means Channel#16
MIDI Note	$2	$0..$7F	$0	Starting note, associated with output#1
Curve	$3	$0..$5	$0:Linear	$0 means LIN $1 means EXP $2 means LOG $3 means S-type $4 means N-type $5 means C1 (custom)
TP1	$4	$0..$7F DEC 0..127	$20 DEC 32	
TP2	$5	$1..$A DEC 1..10	$4 DEC 4	
PP1	$6	$0..$7F DEC 0..127	$7F DEC 127	
PP2	$7	$0..$7F DEC 0..127	$3F DEC 63	
RP1	$8	$1..$7 DEC 1..7	$5 DEC 5	
Spare	$9	HEX $0..$F DEC 0..15	16/32/64/96 or 128 per real outputs number	value $0 means 8, value $1 means 16, ... value $F means 128 outputs
Outputs	$A	HEX $0..$F DEC 0..15	**$F** (128 outputs)	value $0 means 8, value $1 means 16, ... value $F means 128 outputs
Out#1 MIDI Channel	$100	$0..$F	$0:Channel#1	MIDI Channel for output#1
Out#1 MIDI	$101	$0..$7F	$0	MIDI Note for output#1

Note				
Out#2 MIDI Channel	$102	$0..$7F	$0	MIDI Channel for output#2
Out#2 MIDI Note	$103	$0..$7F	$1	MIDI Note for output#2
...	...	...	...	...
Out#128 MIDI Channel	$1FE	$0..$F	$1	MIDI Channel for output#128
Out#128 MIDI Note	$1FF	$0..$7F	$7F	MIDI Note for output#128

***Appendix I*. mddp128uset.exe** operation instructions

mddp128uset (v.2016) is simple Windows-based software application, intended for configuring **mddp128u (firmware version 2016)** units manufactured by MIDI Boutique.

Following are instructions for **mddp128uset** use:

I. Standard Users:

Each of **mddp128u** parameters is programmed by separate *SysEx* message. The entire set of parameters can be uploaded at once for each mode by single button click.

Alternatively, advanced users can program parameters one-by-one, regardless the mode (described in part IV. below).

Following is short description of steps that should be taken to change device's settings in standard case:

1. Connect **mddp128u** MIDI input to PC MIDI output. Special cable or USB-MIDI interface might be necessary in case that PC has no hardware MIDI port, terminated with DIN5 connector;
2. Power up **mddp128u** unit;
3. Run **mddp128uset** application and select the PC physical MIDI output port from "MIDI Out Port" drop-down list. Make sure to UN-check "Advanced user interface" check box This will simplify the use of the software;
4. Choose the correct number of outputs per your particular mddp128 board from "Outputs" drop-down list. Choosing wrong number may cause the unit to respond a bit slower (if bigger number has been chosen) or part of outputs to become non-responsive (if smaller number has been chosen). The unit cannot be damaged by choosing wrong number, it will only affect its current function.
5. Choose desired mode from "Mode" drop-down list. This unit supports 6 modes, described above. Part of controls below will be shaded (disabled) automatically per mode. Only parameters related to currently selected mode will remain allowed.
6. Choose values for all parameters that appear enabled. The allowed parameter combinations are different for each of modes;
7. Click on "Send All" button. When done, it will flash shortly in yellow.

The unit is ready for working immediately, without need of restarting. All user-programmable parameters are kept in non-volatile memory and used till next time you decide to change their values.

II. SCN Mode:

This is new mode, introduced with firmware version 2016. It is similar to Standard mode, except for that user can define separate MIDI channel and MIDI Note numbers individually for each of available decoder outputs. Two or more outputs can be set to respond to same Channel/Note combination. This allows much more flexible use of **mddp128u** decoder. When put in SCN mode the unit will ignore all the rest mode settings and will use internal Channel/Note mapping table. This table is user-programmable, i.e. user can define up to 128 sets of Channel/Note, one set per each **mddp128u** output.
Channel/Note Editor can be opened by clicking on "Channel/Note Editor" button only in case CN mode has been selected in "Mode" drop-down list. Using this editor, you can load/edit/save Channel/Note tables for further use. The entire table is uploaded to **mddp128u** at once. There is no option to program it entry-by-entry, even in Advanced user interface. Programming happens when you press "Send All" button on the main form. The Channel/Note table can be saved to external file or loaded from external file. The file extension of these files is .CN. They are in text format and can be open in external text editor. When editing them with text editor, please keep their original format, otherwise they may not be loaded/parsed. Upon loading of **mddp128uset** software, Channel/Note Editor is always initialized to its default state: all entries set to Chanel#1, Note# per output number, reflecting the factory default values for Channel/Note table. It is user responsibility to change the content of the table before upload, by manual change of values or loading the table from external *.CN file.
The table length is always 128 entries, no matter how many outputs are available in particular board. Upon table upload, only first N table entries are uploaded to the board, where N is the number of outputs selected in "outputs" drop-down list.

III. Factory Reset:

In case you want to return the unit to its factory settings, follow next four steps:
1. Connect **mddp128u** MIDI input to PC MIDI output. Special cable or USB-MIDI interface might be necessary in case that PC has no hardware MIDI port, terminated with DIN5 connector;
2. Power up **mddp128u** unit;
3. Run **mddp128uset** application and select the PC physical MIDI output port from "MIDI Out Port" drop-down list;
4. Click on "Factory Reset" button. Confirm.

The unit is ready for working immediately, without need of restarting. The settings are reset to their default factory values (see ***Appendix H***, ***Table 2***).

*Please note that "Outputs" parameter is always reset to **128**.*
*It is important to set this parameter equal to the real number of outputs, available on particular **mddp128u** decoder.*

IV. Advanced Users:

CAUTION: Using of Advanced user interface requires better knowledge on device functions and programming protocol It is not recommended for new users.

Advanced user interface can be activated by checking "Advanced user interface" check box This interface allows user to change parameters individually, regardless currently selected mode. Separate button "Send" appears next to each user-configurable parameter.

Separate parameter programming may be helpful in cases when user needs to update/adjust single parameter (for example TP1) without affecting the values of any other parameter.

Please note that:
1. If you update an parameter that is not related to current mode of **mddp128u**, the change won't affect the current **mddp128u** operation. The change will affect the operation of the unit when you switch it to one of modes using this particular parameter. For example, if the unit is in currently in S (Standard) mode and you change the value of TP1, nothing will change in current operation. The updated TP1 value will be used once you switch the unit to one of modes using TP1: SAM or PWF.
2. If you change only the mode, the parameters related to this mode won't be uploaded and mddp128u will be using their values as they have been programmed before. Parameters related to the new mode can be individually uploaded one-by-one.

List of parameters relevant to each mode:

Standard mode: MIDI Channel, Start Note;
SAM mode: MIDI Channel, Start Note, TP1, TP2;
VP (Velocity-Pulse) mode: MIDI Channel, Start Note, TP2;
VPWM (Velocity-PWM) mode: MIDI Channel, Start Note, RP1;
PWF (PWM attack/sustain) mode: MIDI Channel, Start Note, TP1, TP2, PP1, PP2, RP1.