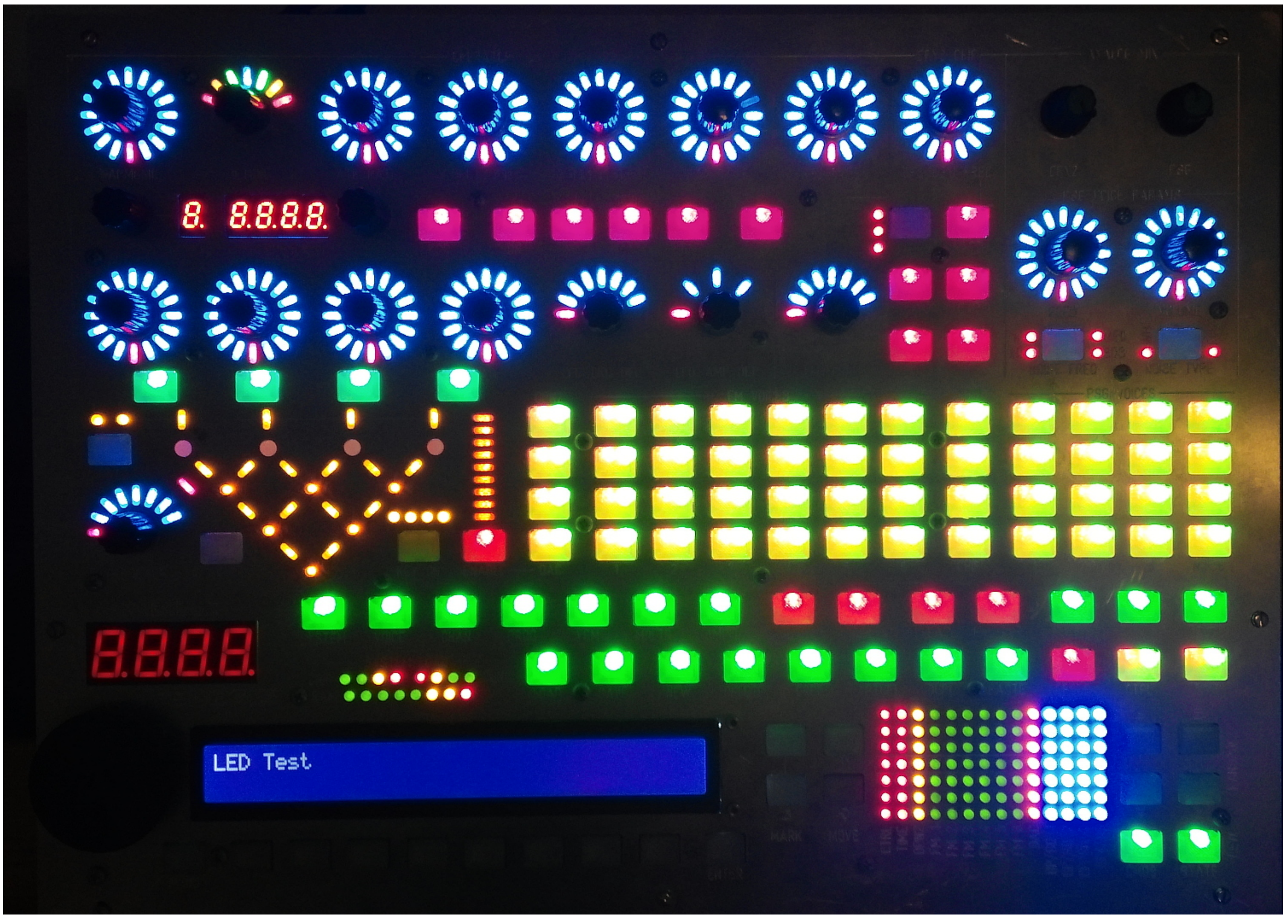
MIDIbox Quad Genesis: Front Panel



To build the front panel of MIDIbox Quad Genesis, you will need (details on each part below):

- An aluminum front panel **OR** 3D Printed Front Panel (See Below)
- A PCB to hold all the buttons, LEDs, encoders, etc., and of course all these electronic components themselves
- A bunch of 3D printed transparent button caps
- Some 3D printed transparent LED pipes for the FM widget
- A 2×40 standard MIDIbox-compliant character LCD screen
- Knob caps for the encoders, including a datawheel-style cap for the datawheel encoder
- The dual-gang and single-gang potentiometers (one each per synth, not per module!) for the Genesis module volume controls, and their appropriate knob caps (see MBHP_GENESIS)
- A whole bunch of M3 or 4-40 screws, nuts, washers, and standoffs
- Standard MIDIbox 10-pin and 16-pin IDC cables, for connecting the front panel PCB to the core's J8/9 and for connecting the LCD to the core's J15A

Dimensions

The aluminum (or acrylic) front panel of MIDIbox Quad Genesis is 15“ x 11” (38.1 x 27.9 cm), and the front panel PCB itself is 13.85“ x 9.85” (35.18 x 25.02 cm), leaving a 0.57“ (1.45 cm) border around the edge. The aluminum I used was 1/16” (0.062“ or 1.57 mm); this may be a consideration if you use acrylic which is much thicker (2mm should probably be fine but 3mm might be a problem, with the 3D printed buttons - **Update** - Smithy has provided modified files for 3D printed buttons and light pipes to fit 3mm panels below.).

The spacing between the underside of the front panel and the upper side of the PCB is about 0.27” (7 mm), which is originally defined by the encoders and the 3D printed buttons are made to match this. I used 1/4“ nylon spacers/standoffs from eBay plus a flat washer (4-40 or M3) to achieve this spacing. I used 0.75” 4-40 screws to put together the front panel, in which case the depth of the assembly was 0.75“ with the electrolytic capacitors and pin header / IDC connector sticking out past this. The panel and PCB can be fastened together with 4-40 or M3 screws.

Aluminum Front Panel

There are two designs available here: my original, and a modified design with the LED pipes in the LED rings having been replaced by 3mm LEDs just sticking through holes in the aluminum.

It took me over 2 hours just to insert the tiny LED pipes in the slits in one single front panel with the first design, which is why I recommend the second. Since they're 3D printed (see below), they have a small range of sizes, and at that scale some are too big and some are too small. I still left them in place for the FM widget and the DAC VU meter; if you don't want those either for some reason, it shouldn't be hard to replace them with holes.

All the LED holes and screw mounting holes are 1/8” (3.17 mm), which should give 3mm LEDs a little play (some are in practice slightly wider than 3mm, and some aren't actually 3mm at all), and are also big enough for both M3 and 4-40“ screws. On my own panel, I had to drill out some of the holes a bit for the yellow and green LEDs sticking through the front panel; I recommend you buy the LEDs you're going to put through the holes first and measure them before you have the front panel manufactured.

It may be possible to make the front panel out of acrylic, a-la MIDIbox SEQ V4. Since the front panel PCB is bolted to it in many places, it will probably be sturdy enough. If it's not, there's a row of screw holes slightly below halfway down the panel, and maybe you can have those screws go into an aluminum bar spanning the width of the front panel and attaching to the sides of your case, to provide extra support in the middle.

Downloads

(You will need to be logged in to your MIDIbox Forum account to access these downloads)

mbqg_fp_original.fpd

mbqg_fp_modledrings.fpd

3D Printed Front Panel

Smithy has provided an STL file for a 2mm 3D Printed Front Panel with 1mm Raised Lettering as a more budget friendly option. It will require a printer with a bed side size typically of 400 x 400mm or larger. This is currently untested and unprinted, and may require a resolution / layer height setting of 0.1mm to be accurate. More info to come when I have designed and 3D printed and a Rear Panel.

Please see the following guide to achieve a dual colour panel with a different colour for the lettering and for the panel: Forum Link

Download

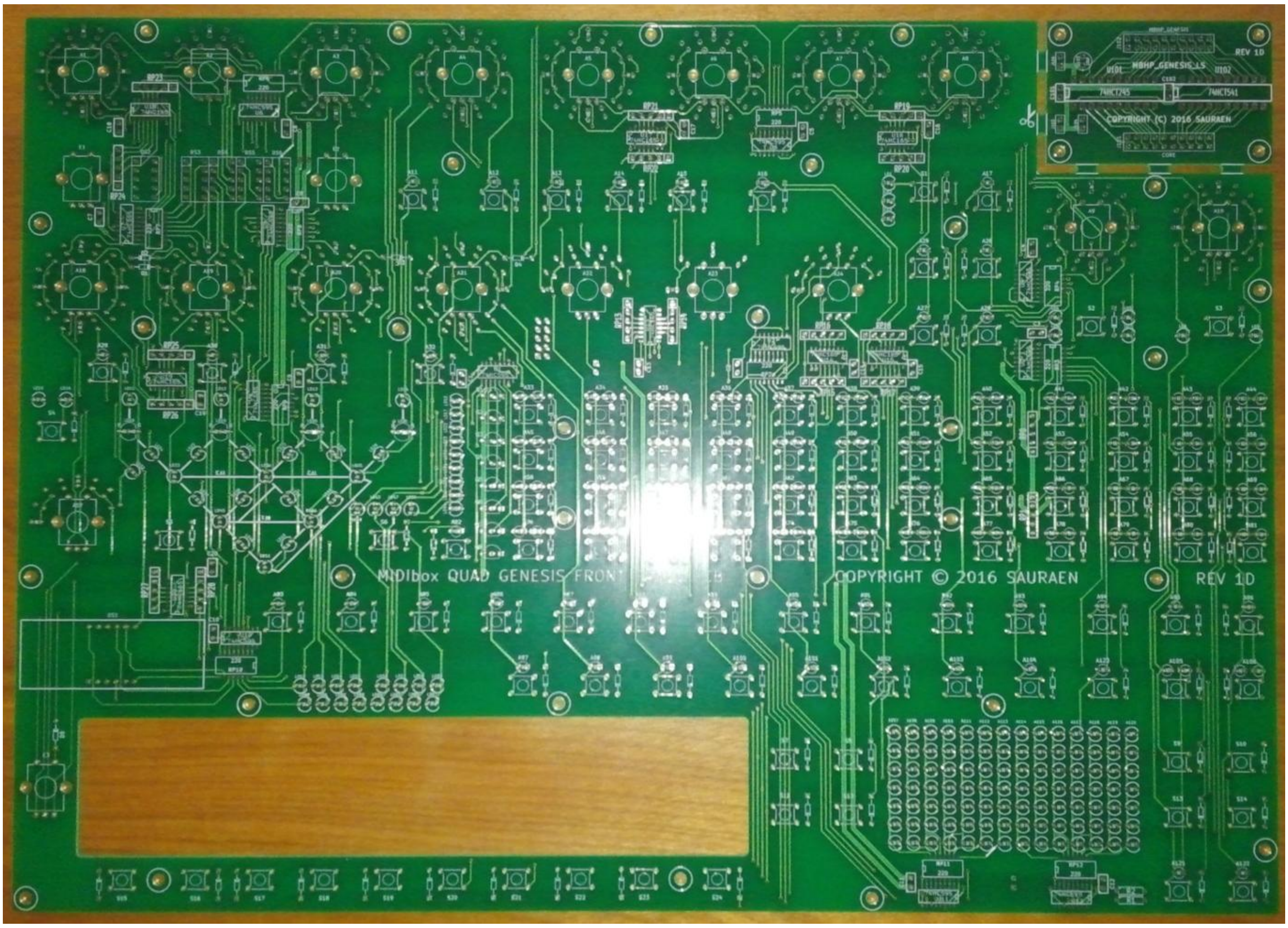
mbqg_fp_modledrings_Holes_Moved_ready_to_print.zip

MBQG_FP PCB

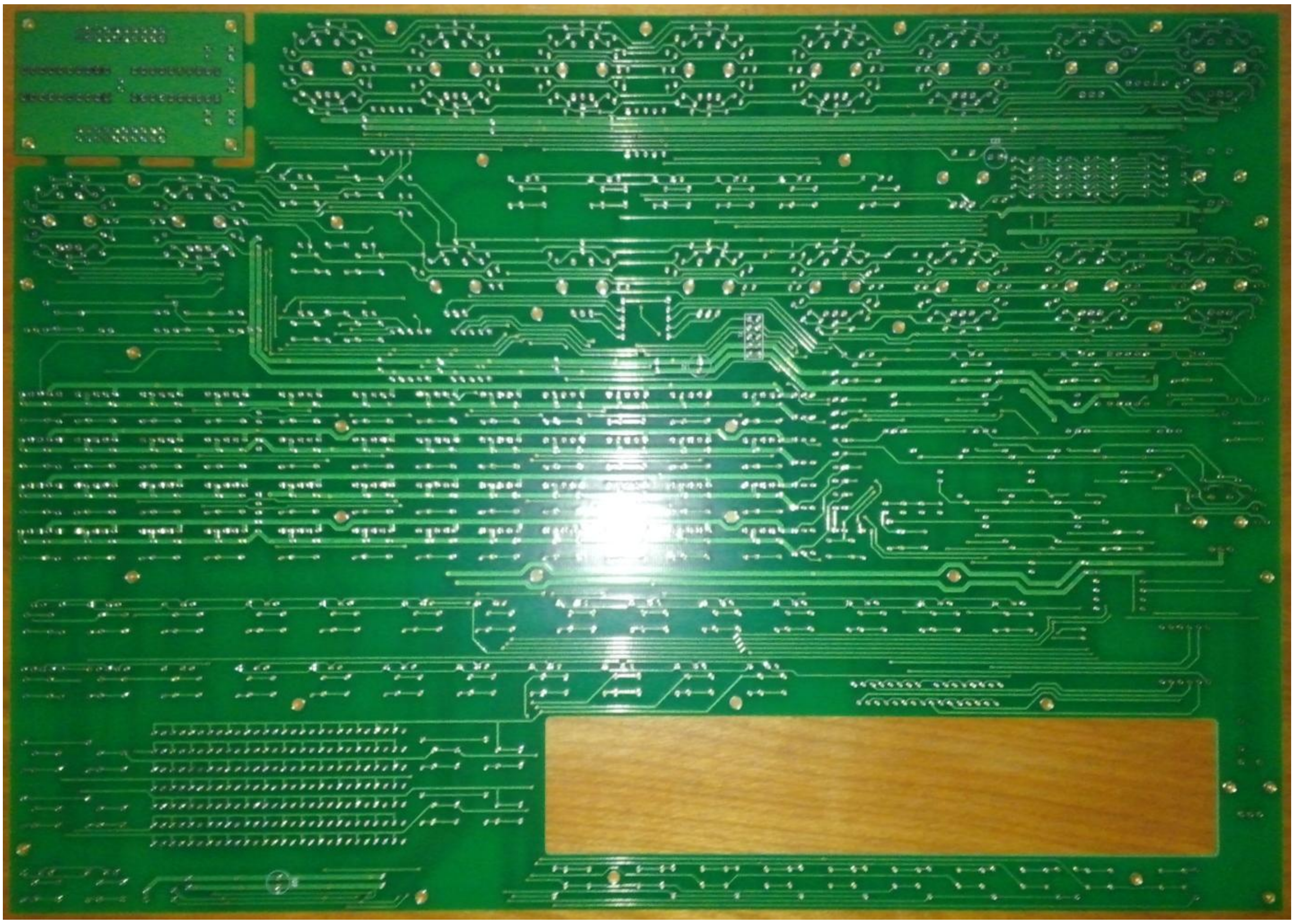
A custom front panel PCB is purchaseable directly from Sauraen for $50 plus at-cost shipping. It includes a free detachable MBHP_GENESIS_LS PCB in the space where the volume knobs will be on the front panel. If you are interested in purchasing one, contact Sauraen by private message at the MIDIbox Forums or post in the MIDIbox Quad Genesis thread there.

Photos

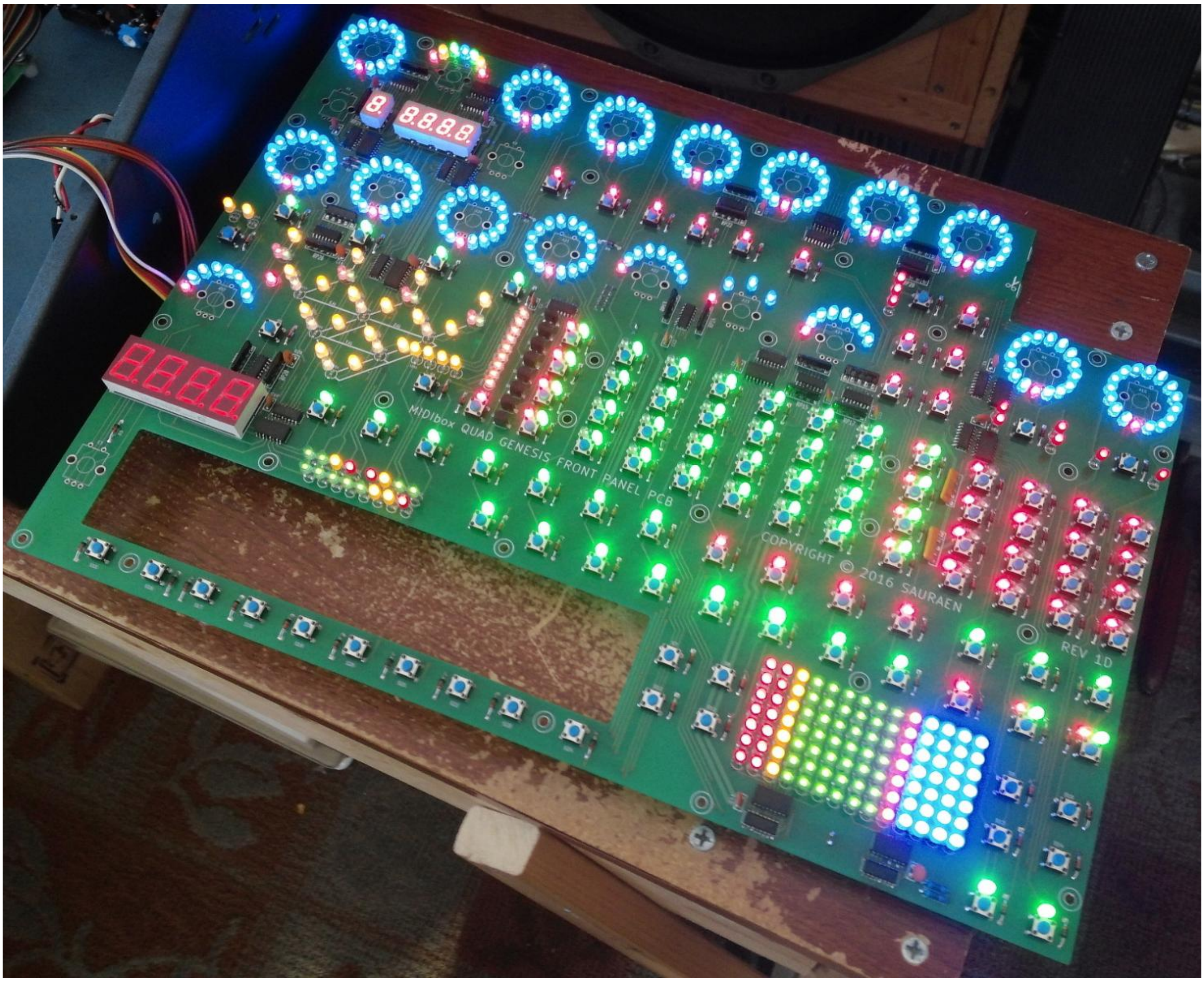
Front:



Back:



Populated and with almost all LEDs lit:



Parts List for MBQG_FP PCB

This is the set of parts I used to build the two MIDIbox Quad Genesis units I constructed. It can also serve as guidelines for if you're building your own board from scratch on veroboard. Here are some notes and caveats:

- As explained above in the section of the aluminum front panel, I used LED pipes for the LED rings, to give them a distinctive oblong shape. To save money on the milling costs and labor on inserting hundreds of tiny plastic pieces into the front panel, most of which are either too small or too big, I recommend making the LED rings just by having 3mm LEDs sticking up through the aluminum. In this case, the LEDs I used below will be too bright to look at directly–they were used to illuminate the LED pipes. So please consider changing to different LEDs.
- Please note that for the LEDs which are under illuminated buttons or under the remaining LED pipes, you will need LEDs which are very bright like the ones below! Cheap 1mcd LEDs will be completely invisible! You need at least 1000mcd; the brightnesses I used are listed below. If you are going to the trouble of making this thing, don't skimp on the LEDs!
- Also don't skimp on the encoders! For MIDIbox FM V2.0/V2.1 I used cheap encoders from China to save money, and many of them barely work anymore!
- Of course, the color of all the LEDs is completely up to personal preference, though you should always use the same two colors beneath all bi-color buttons. Please plan out carefully how

many of each type of LED you will need based on your color preferences and which LEDs will be sticking through the front panel versus lighting up buttons or light pipes. Trust me, you don't want to use regular LEDs to try to light up the plastic, it'll look dim and poor; and you don't want to use the high-brightness LEDs sticking through the front panel, they will blind you!
- On the PCB itself, the individual buttons, LEDs, and diodes are usually not marked. To simplify PCB design in KiCad, “assemblies” were created with e.g. an encoder and 16 LEDs, or a button, two LEDs, and a diode. This is why they're marked “A21” or whatever. All the diodes are the same 1N914 or 1N4148 (or actually any through-hole small signal diode), and each one has a cathode marking, so there should never be any confusion. With the exception of the LEDs in the LED rings, every LED on the board has an actual diode symbol marked, so there should be no confusion about which direction they go (square pad is anode is longer lead). In the LED rings, the four quadrant LEDs' directions are marked, and the rest go in the corresponding direction, following around the circle (anode always on inside). All the tact switches are also the same–just make sure each one is pushed in fully before soldering!

Description	Mouser P/N	Quantity
Encoders; 0.1” board mount with switch; bushing; 24 PPR, 24 detent; etc. If you want to save a couple dollars, you can use the ones without the switch for all the encoders except the datawheel and the four Operator Level encoders–those are the only ones for which the switch is wired up.	652-PEC12R3220FS0024	21
Tact switches; standard, 5mm high	611-PTS645SM502	115
Diodes for button matrix	583-1N4148-T	120
Red LEDs for lighting buttons/caps (4500mcd (!))	604-WP710A10SRC/J4	87
Orange LEDs for lighting buttons/caps (2700mcd (!))	604-WP710A10SEC	9
Yellow LEDs for lighting buttons/caps (3200mcd (!))	859-LTL17KYV3JS	18
Green LEDs for lighting buttons/caps (7200mcd (!))	859-LTL17KTGX3KS	76
Blue LEDs for lighting buttons/caps (1500mcd (!))	859-LTL17KTBS3KS	220
Red LEDs for panel indication (i.e. sticking through holes in front panel) (Tinted, Diffused)	604-WP710A10SRD/D	26
Orange LEDs for panel indication (Tinted, Diffused)	696-SSL-LX3044SOD	7
Yellow LEDs for panel indication (Tinted, Diffused) (For some reason I ended up using these only for the yellow LEDs in the commands display and the load meter, not for the crosspoints in the FM widget; it may have been because I wanted clear-looking LEDs in the FM widget to go with all the other clear/white things there, or because these LEDs are closer to yellow than to amber, and the other LEDs in the FM widget were more amber)	710-151031YS06000	23
Green LEDs for panel indication (Tinted, Clear–I couldn't get appropriate brightness diffused ones)	859-LTL1CHJGTNN	51
Blue LEDs for panel indication (NOT tinted, Diffused–the picture lies, they're white diffused but light up blue; and actually these are a bit too bright to use for sticking through the front panel. You could fudge something with the resistors supplying these columns to try to reduce the brightness, but I might recommend looking for alternative LEDs.)	593-VAOL-3LSBY1	28
Red/Green 5mm LEDs for FM widget operator nodes (3 wire common ANODE)	696-LLX5099SRSGCCA	4

Description	Mouser P/N	Quantity
Large LED display	859-LTC-5623HR	1
Small LED displays	859-LSHD-7501	5
Row drive NMOSFETs (2n7000s aren't strong enough)	689-VN3205N3-G	8
74HC595 shift registers, SOIC-16	511-M74HC595YRM13TR	12
74HC165 shift registers, SOIC-16	863-MC74HC165ADR2G	8
220 ohm x 8 resistor packs, SOIC-16	652-4816P-1LF-220	11
10k ohm x 4 resistor packs, SIL-5 (RP13–RP28, not RP0 and RP1)	858-L051S103LF	16
1k ohm x 4 resistor packs, SIL-5 (these are just RP0 and RP1)	652-4605X-1LF-1K	2
Electrolytic caps (470 uF)	647-UVR1C471MPD	3
Ceramic/film caps (0.1 uF)	Buy by the 100 from eBay	~20
10k dual gang audio taper panel mount pot (OPN2 volume)	313-1240F-10K	1
10k audio taper panel mount pot	858-P160KNPC15A10K	1
5×2 pin header	Buy in larger size (preferably from eBay) and snap off	1

Don't forget: the 2×40 character LCD, the pin header for the LCD, the ribbon cable and IDC connectors for the LCD and front panel, knob caps for the regular encoders, datawheel knob cap for the datawheel, knob caps for the volume pots.

Schematic and Reference Designators

Schematic image (unfortunately, MIDIbox Gallery scaled down the original image, and offsite documentation is frowned upon):

For Color fields, as discussed above, you may change the colors to whatever you want, but it is recommended to keep the colors consistent (e.g. make everything “green” below be the same color, and everything “red” below be a different color). For the Red/Green buttons below, the Red LED is always on the left and the Green LED is always on the right.

Ref. Des.	Type	Color	Description
A1	Encoder + 16-LED ring	Blue	FM operator parameter “Harmonic” (“FMult”)
A2	Encoder + 7-LED ring	Rainbow	FM operator parameter “Detune”
A3	Encoder + 16-LED ring	Blue	FM operator parameter “Atk Rate”
A4	Encoder + 16-LED ring	Blue	FM operator parameter “Dec1 Rate”
A5	Encoder + 16-LED ring	Blue	FM operator parameter “Dec1 Level”
A6	Encoder + 16-LED ring	Blue	FM operator parameter “Dec2 Rate”
A7	Encoder + 16-LED ring	Blue	FM operator parameter “Rel Rate”
A8	Encoder + 16-LED ring	Blue	OPN2 parameter “Ch3 CSM Freq”
A9	Encoder + 16-LED ring	Blue	PSG voice parameter “Freq”
A10	Encoder + 16-LED ring	Blue	PSG voice parameter “Volume”
A11	Button + LED	Red	FM operator parameter “KSR” (Key Scale Rate)
A12	Button + LED	Red	FM operator SSG-EG parameter “On”
A13	Button + LED	Red	FM operator SSG-EG parameter “Init”

Ref. Des.	Type	Color	Description
A14	Button + LED	Red	FM operator SSG-EG parameter “Toggle”
A15	Button + LED	Red	FM operator SSG-EG parameter “Hold”
A16	Button + LED	Red	FM operator parameter “LFO AM” (LFO → operator amplitude modulation toggle)
A17	Button + LED	Red	OPN2 parameter “Ch3 CSM Fast” (originally Timer A fast, now Timer A enable)
A18–A21	Encoder + 16-LED ring	Blue	FM voice parameter “Oper 1 Level” – “Oper 4 Level”
A22	Encoder + 8-LED ring	Blue	FM voice parameter “LFO-Freq Depth” (LFO → frequency modulation depth)
A23	Encoder + 4-LED ring	Blue	FM voice parameter “LFO-Amp Depth” (LFO → amplitude modulation depth)
A24	Encoder + 8-LED ring	Blue	OPN2 parameter “LFO Freq”
A25	Button + LED	Red	OPN2 parameter “Ugly” (now-famous test bit 0x21:4)
A26	Button + LED	Red	OPN2 parameter “DAC Override” (test bit 0x2C:5)
A27	Button + LED	Red	OPN2 parameter “LFO Enable”
A28	Button + LED	Red	OPN2 parameter “EG Enable” (invert of test bit 0x21:5)
A29–A32	Button + LED	Green	Operator Selection 1–4
A33–A44	Button + 2 LEDs	Red/Green	Genesis 1 Voice Selection: DAC, FM voices 1–6, OPN2 globals, PSG voices 1-3, noise
A45–A56	Button + 2 LEDs	Red/Green	Genesis 2 Voice Selection: DAC, FM voices 1–6, OPN2 globals, PSG voices 1-3, noise
A57	Encoder + 8-LED ring	Blue	OPN2 voice parameter “Feedback”
A58–A69	Button + 2 LEDs	Red/Green	Genesis 3 Voice Selection: DAC, FM voices 1–6, OPN2 globals, PSG voices 1-3, noise
A70–A81	Button + 2 LEDs	Red/Green	Genesis 4 Voice Selection: DAC, FM voices 1–6, OPN2 globals, PSG voices 1-3, noise
A82	Button + LED	Red	OPN2 parameter “DAC Enable”
A83	Button + LED	Green	System Mode
A84	Button + LED	Green	Voice Mode
A85	Button + LED	Green	Channel Mode
A86	Button + LED	Green	Program Mode
A87	Button + LED	Green	VGM Editor Mode
A88	Button + LED	Green	Modulator Mode
A89	Button + LED	Green	Sample Mode
A90	Button + LED	Red	Mute
A91	Button + LED	Red	Solo
A92	Button + LED	Red	Release
A93	Button + LED	Red	Panel Override
A94	Button + LED	Green	Restart
A95	Button + LED	Green	Play
A96	Button + LED	Green	Reset
A97	Button + LED	Green	Load
A98	Button + LED	Green	Save
A99	Button + LED	Green	New
A100	Button + LED	Green	Delete
A101	Button + LED	Green	Crop

Ref. Des.	Type	Color	Description
A102	Button + LED	Green	Capture
A103	Button + LED	Green	Duplicate
A104	Button + LED	Green	Paste
A105	Button + LED	Red/Green	Ctrl
A106	Button + LED	Red/Green	Time
A107	7-LED Column	Red	VGM Commands Display: Ctrl
A108	7-LED Column	Red	VGM Commands Display: Time
A109	7-LED Column	Yellow	VGM Commands Display: OPN2
A110	7-LED Column	Green	VGM Commands Display: FM 1
A111	7-LED Column	Green	VGM Commands Display: FM 2
A112	7-LED Column	Green	VGM Commands Display: FM 3
A113	7-LED Column	Green	VGM Commands Display: FM 4
A114	7-LED Column	Green	VGM Commands Display: FM 5
A115	7-LED Column	Green	VGM Commands Display: FM 6
A116	7-LED Column	Orange	VGM Commands Display: DAC
A117	7-LED Column	Blue	VGM Commands Display: OP/SQ 1
A118	7-LED Column	Blue	VGM Commands Display: OP/SQ 2
A119	7-LED Column	Blue	VGM Commands Display: OP/SQ 3
A120	7-LED Column	Blue	VGM Commands Display: OP4/NS
A121	Button + LED	Green	Commands View
A122	Button + LED	Green	State View
A123	Button + LED	Red	Group
C1–C20	Capacitor	N/A	0.1uF ceramic or film capacitors for chips
C21–C23	Capacitor	N/A	470uF electrolytic capacitors for power rails, reverse mounted
D1–D5	Diode	N/A	Diodes for encoder buttons; all normal button diodes included in Assembly or Switch
DS1	LED Display	Red	Main Display
DS2	LED Display	Red	Octave
DS3–DS6	LED Display	Red	Frequency
E1	Encoder	N/A	OPN2 voice parameter “Octave”
E2	Encoder	N/A	OPN2 voice parameter “Frequency”
E3	Encoder	N/A	Datawheel
LD1–LD3	LED	Red	OPN2 Ch3 Mode
LD4–LD7	LED	Red	PSG Noise Freq
LD8–LD9	LED	Red	PSG Noise Type
LD10, LD16	LED	Yellow	FM Voice Output
LD11–LD14, LD24–LD29, LD37–LD40, LD48–LD49	LED	Yellow	FM Widget Path
LD15, LD17, LD22, LD23, LD30, LD31, LD36, LD42, LD44	LED	Orange	DAC VU Meter
LD18–LD21	Bicolor LED	Red/Green	FM Widget Operator Node

Ref. Des.	Type	Color	Description
LD33–LD35, LD45–LD46, LD51	LED	Yellow	FM Widget Path Node
LD32	LED	Red	FM Widget Feedback
LD41, LD43, LD47, LD50	LED	Yellow	Key On
LD52–LD53, LD58-LD59, LD60-LD64,	LED	Green	System Load Meter
LD54, LD57, LD65-LD66,	LED	Yellow	System Load Meter
LD55-LD56, LD67	LED	Red	System Load Meter
R1–R2	Resistors	N/A	2.2k terminating resistors for clock and latch lines
RP0–RP1	Resistor Packs	N/A	1k row driver pull-ups (to reduce ghosting)
RP2–RP12	Resistor Packs	N/A	220 ohm LED column current limiters
RP13–RP28	Resistor Packs	N/A	10k button column pull-ups
S1	Button	N/A	OPN2 Ch3 Mode
S2	Button	N/A	PSG Noise Freq
S3	Button	N/A	PSG Noise Type
S4	Button	N/A	FM Voice Output
S5	Button	N/A	Algorithm
S6	Button	N/A	Key On
S7	Button	N/A	Mark Beginning
S8	Button	N/A	Move Up
S9	Button	N/A	Up One Command
S10	Button	N/A	Up One State
S11	Button	N/A	Mark End
S12	Button	N/A	Move Down
S13	Button	N/A	Down One Command
S14	Button	N/A	Down One State
S15	Button	N/A	Menu
S16–S23	Button	N/A	Softkeys
S24	Button	N/A	Enter
TF1	Triforce	Red	Triforce of Power
TF2	Triforce	Blue	Triforce of Wisdom
TF3	Triforce	Green	Triforce of Courage
U1–U12	IC	N/A	74HC595 Output Shift Register, SMD
U13–U20	IC	N/A	74HC165 Input Shift Register, SMD

Custom PCB / Veroboard Info

This is the mapping for the MBQG_FP board itself–if you have the commercially produced PCB from Sauraen, you don't need this info because it's already in the copper!

MIDIbox Quad Genesis supports its front panel controls in any valid MIDIbox button-LED matrix configuration, with the following restrictions:

DOUT Restrictions

- The same BLM must be used for buttons and LEDs, and it must have 8 rows.
- LED display digits must be common anode, connected with their anode to a DOUT shift register pin, and their cathodes to the row lines in the order 0-7 = A-B-C-D-E-F-G-DP.

DIN Restrictions

- The same BLM must be used for buttons and LEDs, and it must have 8 rows.
- Encoders must have their common pin grounded, and their two switching pins directly connected to two DIN shift register pins (not in the matrix).

For now, the mapping is hard-coded in frontpanel.c, but eventually (if there is interest), it will be read from a text-based configuration file on the SD card upon startup. If you match the matrix maps below, you won't need to edit this at all.

DOUT Matrix Map

Syntax:

- X = no item
- B = button
- EB = encoder button (push)
- L = LED
- R = Red
- G = Green
- G# = Genesis #
- O# = LED Ring Segment # (counted from bottom, clockwise)
- LW# = LED, FM Widget, Reference Designator # (since there is no good way to label the individual widget segments)

All LED display digits are wired, from rows 0 to 7: A-B-C-D-E-F-G-DP

VGM Commands Matrix is wired, from top to bottom, 0-1-2-3-4-5-6 (row 7 not used for any columns)

Counting the individual segments in the LED displays, there are 638 LEDs on the front panel.

SR	IDX	BIT	0	1	2	3	4	5	6	7
U1	1	0	ROW0 DRIVER (ACTIVE HIGH)							
		1	ROW1 DRIVER (ACTIVE HIGH)							
		2	ROW2 DRIVER (ACTIVE HIGH)							
		3	ROW3 DRIVER (ACTIVE HIGH)							
		4	ROW4 DRIVER (ACTIVE HIGH)							
		5	ROW5 DRIVER (ACTIVE HIGH)							
		6	ROW6 DRIVER (ACTIVE HIGH)							
		7	ROW7 DRIVER (ACTIVE HIGH)							

SR	IDX	BIT	0	1	2	3	4	5	6	7
U2	2	0	LR_G1_DAC	LG_G1_DAC	LR_G2_DAC	LG_G2_DAC	LR_G3_DAC	LG_G3_DAC	LR_G4_DAC	LG_G4_DAC
		1	LR_G1_V1	LG_G1_V1	LR_G2_V1	LG_G2_V1	LR_G3_V1	LG_G3_V1	LR_G4_V1	LG_G4_V1
		2	LR_G1_V2	LG_G1_V2	LR_G2_V2	LG_G2_V2	LR_G3_V2	LG_G3_V2	LR_G4_V2	LG_G4_V2
		3	LR_G1_V3	LG_G1_V3	LR_G2_V3	LG_G2_V3	LR_G3_V3	LG_G3_V3	LR_G4_V3	LG_G4_V3
		4	LR_G1_V4	LG_G1_V4	LR_G2_V4	LG_G2_V4	LR_G3_V4	LG_G3_V4	LR_G4_V4	LG_G4_V4
		5	LR_G1_V5	LG_G1_V5	LR_G2_V5	LG_G2_V5	LR_G3_V5	LG_G3_V5	LR_G4_V5	LG_G4_V5
		6	X	X	X	X	LFOFREQ_O4	LFOFREQ_O5	LFOFREQ_O6	LFOFREQ_O7
		7	X	X	X	X	LFOFREQ_O11	LFOFREQ_O10	LFOFREQ_O9	LFOFREQ_O8
U3	3	0	LR_G1_V6	LG_G1_V6	LR_G2_V6	LG_G2_V6	LR_G3_V6	LG_G3_V6	LR_G4_V6	LG_G4_V6
		1	LR_G1_OPN2	LG_G1_OPN2	LR_G2_OPN2	LG_G2_OPN2	LR_G3_OPN2	LG_G3_OPN2	LR_G4_OPN2	LG_G4_OPN2
		2	LG_CTRL	X	LR_CTRL	X	LG_TIME	L_CMDS	LR_TIME	L_STATE
		3	L_GROUP	L_MUTE	L_RESTART	L_SOLO	L_PLAY	L_RELEASE	L_RESET	L_PNLOVR
		4	LR_G1_S1	LG_G1_S1	LR_G2_S1	LG_G2_S1	LR_G3_S1	LG_G3_S1	LR_G4_S1	LG_G4_S1
		5	LR_G1_S2	LG_G1_S2	LR_G2_S2	LG_G2_S2	LR_G3_S2	LG_G3_S2	LR_G4_S2	LG_G4_S2
		6	LR_G1_S3	LG_G1_S3	LR_G2_S3	LG_G2_S3	LR_G3_S3	LG_G3_S3	LR_G4_S3	LG_G4_S3
		7	LR_G1_NOISE	LG_G1_NOISE	LR_G2_NOISE	LG_G2_NOISE	LR_G3_NOISE	LG_G3_NOISE	LR_G4_NOISE	LG_G4_NOISE
U4	4	0	X	X	X	X	X	L_CH3NORM	L_CH34FREQ	L_CH3CSM
		1	X	X	L_EG	L_LFO	L_DACOVR	L_UGLY	L_CH3FAST	X
		2	L_NFMED	L_NFHI	L_KSR	L_SSGON	L_SSGINIT	L_SSGTGL	L_SSGHOLD	L_LFOAM
		3	L_NFSQ3	L_NFLO	L_NMPLS	L_NMWHT	X	X	X	X
		4	PSGVOL_O15	PSGVOL_O14	PSGVOL_O13	PSGVOL_O12	PSGVOL_O11	PSGVOL_O10	PSGVOL_O9	PSGVOL_O8
		5	PSGVOL_O0	PSGVOL_O1	PSGVOL_O2	PSGVOL_O3	PSGVOL_O4	PSGVOL_O5	PSGVOL_O6	PSGVOL_O7
		6	PSGFREQ_O15	PSGFREQ_O14	PSGFREQ_O13	PSGFREQ_O12	PSGFREQ_O11	PSGFREQ_O10	PSGFREQ_O9	PSGFREQ_O8
		7	PSGFREQ_O0	PSGFREQ_O1	PSGFREQ_O2	PSGFREQ_O3	PSGFREQ_O4	PSGFREQ_O5	PSGFREQ_O6	PSGFREQ_O7
U5	5	0	RELRATE_O15	RELRATE_O14	RELRATE_O13	RELRATE_O12	RELRATE_O11	RELRATE_O10	RELRATE_O9	RELRATE_O8
		1	RELRATE_O0	RELRATE_O1	RELRATE_O2	RELRATE_O3	RELRATE_O4	RELRATE_O5	RELRATE_O6	RELRATE_O7
		2	CSMFREQ_O15	CSMFREQ_O14	CSMFREQ_O13	CSMFREQ_O12	CSMFREQ_O11	CSMFREQ_O10	CSMFREQ_O9	CSMFREQ_O8
		3	CSMFREQ_O0	CSMFREQ_O1	CSMFREQ_O2	CSMFREQ_O3	CSMFREQ_O4	CSMFREQ_O5	CSMFREQ_O6	CSMFREQ_O7
		4	DECLVL_O15	DECLVL_O14	DECLVL_O13	DECLVL_O12	DECLVL_O11	DECLVL_O10	DECLVL_O9	DECLVL_O8
		5	DECLVL_O0	DECLVL_O1	DECLVL_O2	DECLVL_O3	DECLVL_O4	DECLVL_O5	DECLVL_O6	DECLVL_O7
		6	DEC2R_O15	DEC2R_O14	DEC2R_O13	DEC2R_O12	DEC2R_O11	DEC2R_O10	DEC2R_O9	DEC2R_O8
		7	DEC2R_O0	DEC2R_O1	DEC2R_O2	DEC2R_O3	DEC2R_O4	DEC2R_O5	DEC2R_O6	DEC2R_O7
U6	6	0	DEC1R_O15	DEC1R_O14	DEC1R_O13	DEC1R_O12	DEC1R_O11	DEC1R_O10	DEC1R_O9	DEC1R_O8
		1	DEC1R_O0	DEC1R_O1	DEC1R_O2	DEC1R_O3	DEC1R_O4	DEC1R_O5	DEC1R_O6	DEC1R_O7
		2	ATTACK_O15	ATTACK_O14	ATTACK_O13	ATTACK_O12	ATTACK_O11	ATTACK_O10	ATTACK_O9	ATTACK_O8
		3	ATTACK_O0	ATTACK_O1	ATTACK_O2	ATTACK_O3	ATTACK_O4	ATTACK_O5	ATTACK_O6	ATTACK_O7
		4	X	X	X	X	DETUNE_O11	DETUNE_O10	DETUNE_O9	DETUNE_O8
		5	X	X	X	X	X	DETUNE_O5	DETUNE_O6	DETUNE_O7
		6	HARM_O15	HARM_O14	HARM_O13	HARM_O12	HARM_O11	HARM_O10	HARM_O9	HARM_O8
		7	HARM_O0	HARM_O1	HARM_O2	HARM_O3	HARM_O4	HARM_O5	HARM_O6	HARM_O7
U7	7	0	OP1LVL_O0	OP1LVL_O1	OP1LVL_O2	OP1LVL_O3	OP1LVL_O4	OP1LVL_O5	OP1LVL_O6	OP1LVL_O7
		1	OP1LVL_O15	OP1LVL_O14	OP1LVL_O13	OP1LVL_O12	OP1LVL_O11	OP1LVL_O10	OP1LVL_O9	OP1LVL_O8
		2	OP2LVL_O0	OP2LVL_O1	OP2LVL_O2	OP2LVL_O3	OP2LVL_O4	OP2LVL_O5	OP2LVL_O6	OP2LVL_O7
		3	OP2LVL_O15	OP2LVL_O14	OP2LVL_O13	OP2LVL_O12	OP2LVL_O11	OP2LVL_O10	OP2LVL_O9	OP2LVL_O8
		4	FREQUENCY DIGIT 3 COMMON ANODE							
		5	FREQUENCY DIGIT 2 COMMON ANODE							
		6	FREQUENCY DIGIT 1 COMMON ANODE							
		7	OCTAVE DIGIT COMMON ANODE							
U8	8	0	OP3LVL_O0	OP3LVL_O1	OP3LVL_O2	OP3LVL_O3	OP3LVL_O4	OP3LVL_O5	OP3LVL_O6	OP3LVL_O7
		1	OP3LVL_O15	OP3LVL_O14	OP3LVL_O13	OP3LVL_O12	OP3LVL_O11	OP3LVL_O10	OP3LVL_O9	OP3LVL_O8
		2	OP4LVL_O0	OP4LVL_O1	OP4LVL_O2	OP4LVL_O3	OP4LVL_O4	OP4LVL_O5	OP4LVL_O6	OP4LVL_O7
		3	OP4LVL_O15	OP4LVL_O14	OP4LVL_O13	OP4LVL_O12	OP4LVL_O11	OP4LVL_O10	OP4LVL_O9	OP4LVL_O8
		4	X	X	X	X	LFOFDEP_O4	LFOFDEP_O5	LFOFDEP_O6	LFOFDEP_O7
		5	X	X	X	X	LFOFDEP_O11	LFOFDEP_O10	LFOFDEP_O9	LFOFDEP_O8
		6	X	X	X	X	LFOADEP_O4	LFOADEP_O10	LFOADEP_O6	LFOADEP_O8
		7	FREQUENCY DIGIT 4 COMMON ANODE							

SR	IDX	BIT	0	1	2	3	4	5	6	7
U9	9	0	X	X	X	X	L_DACB9	L_DACENAB	LW46	X
		1	LW45	LW48	LW51	LW49	L_KON1	L_KON2	L_KON3	L_KON4
		2	FEEDBACK_O11	FEEDBACK_O10	FEEDBACK_O9	FEEDBACK_O8	L_DACB5	L_DACB6	L_DACB7	L_DACB8
		3	FEEDBACK_O4	FEEDBACK_O5	FEEDBACK_6	FEEDBACK_7	L_DACB1	L_DACB2	L_DACB3	L_DACB4
		4	LW32	LW33	LW37	LW38	LW34	LW39	LW40	LW35
		5	L_OUTL	L_OUTR	LW24	LW25	LW26	LW27	LW28	LW29
		6	LR_OP1	LG_OP1	LR_OP2	LG_OP2	LR_OP3	LG_OP3	LR_OP4	LG_OP4
		7	L_SELO1	L_CARRO1	L_SELO2	L_CARRO2	L_SELO3	L_CARRO3	L_SELO4	L_CARRO4
U10	10	0	MAIN DIGIT 4 COMMON ANODE							
		1	MAIN DIGIT 3 COMMON ANODE							
		2	MAIN DIGIT 2 COMMON ANODE							
		3	MAIN DIGIT 1 COMMON ANODE							
		4	L_RAM0	L_RAM1	L_RAM2	L_RAM3	L_RAM4	L_RAM5	L_RAM6	L_RAM7
		5	L_CHIP0	L_CHIP1	L_CHIP2	L_CHIP3	L_CARD3	L_CARD2	L_CARD1	L_CARD0
		6	L_LOAD	L_CROP	L_SAVE	L_CAPTURE	L_NEW	L_DUPL	L_DELETE	L_PASTE
		7	L_PROG	X	L_VGM	L_SYSTEM	L_MDLTR	L_VOICE	L_SAMPLE	L_CHAN
U11	11	0	VGM COMMANDS MATRIX COLUMN 7: FM 4							
		1	VGM COMMANDS MATRIX COLUMN 6: FM 3							
		2	VGM COMMANDS MATRIX COLUMN 5: FM 2							
		3	VGM COMMANDS MATRIX COLUMN 4: FM 1							
		4	VGM COMMANDS MATRIX COLUMN 3: OPN2							
		5	VGM COMMANDS MATRIX COLUMN 2: TIME							
		6	VGM COMMANDS MATRIX COLUMN 1: CTRL							
		7	X							
U12	12	0	X							
		1	VGM COMMANDS MATRIX COLUMN 14: OP4/NS							
		2	VGM COMMANDS MATRIX COLUMN 13: OP/SQ 3							
		3	VGM COMMANDS MATRIX COLUMN 12: OP/SQ 2							
		4	VGM COMMANDS MATRIX COLUMN 11: OP/SQ 1							
		5	VGM COMMANDS MATRIX COLUMN 10: DAC							
		6	VGM COMMANDS MATRIX COLUMN 9: FM 6							
		7	VGM COMMANDS MATRIX COLUMN 8: FM 5							

DIN Matrix Map

SR	IDX	BIT	0	1	2	3	4	5	6	7
U13	1	0	OP4LVL ENCODER A							
		1	OP4LVL ENCODER B							
		2	LFOFDEP ENCODER B							
		3	LFOFDEP ENCODER A							
		4	LFOADEP ENCODER B							
		5	LFOADEP ENCODER A							
		6	LFOFREQ ENCODER A							
		7	LFOFREQ ENCODER B							
U14	2	0	B_MENU	B_G1_V2	X	B_G2_V2	X	B_G3_V2	B_ENTER	B_G4_V2
		1	B_SOFT1	B_G1_V3	B_SOFT2	B_G2_V3	B_SOFT3	B_G3_V3	B_SOFT4	B_G4_V3
		2	B_SOFT5	B_G1_V4	B_SOFT6	B_G2_V4	B_SOFT7	B_G3_V4	B_SOFT8	B_G4_V4
		3	B_MARKST	B_G1_V5	B_MOVEUP	B_G2_V5	B_MARKEND	B_G3_V5	B_MOVEDN	B_G4_V5
		4	X	B_G1_V1	X	B_G2_V1	X	B_G3_V1	X	B_G4_V1
		5	X	B_G1_DAC	X	B_G2_DAC	X	B_G3_DAC	X	B_G4_DAC
		6	B_SELO1	B_OUT	B_SELO2	B_ALG	B_SELO3	B_DACEN	B_SELO4	B_KON
		7	X	X	X	X	EB_OP1LVL	EB_OP2LVL	EB_OP3LVL	EB_OP4LVL

SR	IDX	BIT	0	1	2	3	4	5	6	7
U15	3	0	X	B_G1_V6	X	B_G2_V6	X	B_G3_V6	X	B_G4_V6
		1	X	B_G1_OPN2	X	B_G2_OPN2	X	B_G3_OPN2	X	B_G4_OPN2
		2	X	B_G1_S1	X	B_G2_S1	X	B_G3_S1	X	B_G4_S1
		3	X	B_G1_S2	X	B_G2_S2	X	B_G3_S2	X	B_G4_S2
		4	X	B_G1_S3	X	B_G2_S3	X	B_G3_S3	X	B_G4_S3
		5	X	B_G1_NOISE	X	B_G2_NOISE	X	B_G3_NOISE	X	B_G4_NOISE
		6	B_NSTYPE	B_NSFREQ	B_EG	B_LFO	B_DACOVR	B_UGLY	B_FAST	B_CH3MODE
		7	X	X	B_KSR	B_SSGON	B_SSGINIT	B_SSGTGL	B_SSGHOLD	B_LFOAM
U16	4	0	CSMFREQ ENCODER A							
		1	CSMFREQ ENCODER B							
		2	RELRATE ENCODER A							
		3	RELRATE ENCODER B							
		4	PSGFREQ ENCODER A							
		5	PSGFREQ ENCODER B							
		6	PSGVOL ENCODER A							
		7	PSGVOL ENCODER B							
U17	5	0	DEC2R ENCODER B							
		1	DEC2R ENCODER A							
		2	DECLVL ENCODER B							
		3	DECLVL ENCODER A							
		4	DEC1R ENCODER B							
		5	DEC1R ENCODER A							
		6	FREQ ENCODER B							
		7	FREQ ENCODER A							
U18	6	0	ATTACK ENCODER B							
		1	ATTACK ENCODER A							
		2	DETUNE ENCODER B							
		3	DETUNE ENCODER A							
		4	HARM ENCODER B							
		5	HARM ENCODER A							
		6	OCTAVE ENCODER B							
		7	OCTAVE ENCODER A							
U19	7	0	OP2LVL ENCODER B							
		1	OP2LVL ENCODER A							
		2	OP3LVL ENCODER B							
		3	OP3LVL ENCODER A							
		4	OP1LVL ENCODER B							
		5	OP1LVL ENCODER A							
		6	FEEDBACK ENCODER B							
		7	FEEDBACK ENCODER A							
U20	8	0	DATAWHEEL ENCODER B							
		1	DATAWHEEL ENCODER A							
		2	X							
		3	X							
		4	B_CTRL	B_CMDDN	B_CMDUP	B_STATEDN	B_TIME	B_CMDS	B_STATEUP	B_STATE
		5	B_LOAD	B_CROP	B_SAVE	B_CAPTURE	B_NEW	B_DUPL	B_DELETE	B_PASTE
		6	B_GROUP	B_MUTE	B_RESTART	B_SOLO	B_PLAY	B_RELEASE	B_RESET	B_PNLOVR
		7	B_PROG	EB_DATAWHL	B_VGM	B_SYSTEM	B_MDLTR	B_VOICE	B_SAMPLE	B_CHAN

3D Printed Items

The buttons on MBQG_FP consist simply of a 5mm tact switch and up to 2 3mm LEDs on the front panel board, and a 3D printed, transparent plastic button cap which sits loosely on top. It is held in by the front panel, its flanges, and the button and LEDs underneath.

MBQG_FP also uses a number of LED pipes, which are 3D printed, transparent pieces which are held in the slots of the aluminum front panel by friction, and which are intended to cause the slot to light up uniformly when lit underneath by an LED. The original version used LED pipes for all the LED rings, but since this was extremely tedious, I recommend only using the LED pipes for the FM Widget and the DAC VU meter.

Things You Should Know If Having These Printed

- You need 115 buttons and 26 LED pipes, assuming you're not using LED pipes for all the LED rings. However, especially if the 3D printer (machine and/or person) is inexperienced, you will need lots of extras. For the LED pipes, because they are friction-fit, buy at least double the number you will need; and for the buttons, at least 20 extra or so.
- The total cost to have these items printed should be roughly $20-$30 plus shipping. If someone is trying to charge you more than that, either there's an issue (like they got the scale wrong and think you want giant-sized parts), or they're trying to gyp you. **If you try to get these printed by a commercial service (e.g. Ponoko), they will charge a minimum of $1 per part and the result will be prohibitively expensive!** Have them printed by a real person instead, either a friend with a 3D printer or possibly someone from the RepRap forums.

Things You Need to Tell the Person Printing Them

- The scale in the STL files is 1 INCH, not 1 mm. The buttons should be about 1/2“ or 12mm wide. (This is not intended as an engineering dimension, just as a sanity check–use the actual dimensions in the file.)
- **Update:** Smithy has added additional STL files in metric scale below.
- Either ABS or PLA transparent filament should work. I had the guy who made them for me do tests with both, and I liked the light-up appearance of the ABS better but the “off” appearance of the PLA better. But they were pretty similar.
- I had the guy who made them for me do tests with different fill rates, and if I remember correctly around 10% fill worked best. 100% fill would be needlessly expensive and would dim the LEDs.
- The buttons need to be printed with the flanges and the cutout down, and the beveled rectangular surface up. The LED pipes need to be printed with the flat surface down and the crown-shaped end up.

Downloads

(You must be logged in with your MIDIbox Forum account to access the downloads)

mbqg_fp_buttons.stl

mbqg_fp_ledpipes.stl

Additional files provided by Smithy:

Buttons in metric scale for 2mm Front Panel

LED pipes in metric scale for 2mm Front Panel

Buttons in metric scale for 3mm Front Panel

LED pipes in metric scale for 3mm Front Panel

Assembly Guide

Once you have your front panel PCB, your aluminum front panel, and all the other parts, here's some tips for the assembly process.

1. Begin by soldering all the SMD shift registers and resistor packs. The pads aren't very big (if I had known, I would have made them bigger…), and I had a lot of issues with bad connections because a leg which looked soldered wasn't actually attached. If you have access to solder paste and a heat gun, use that. If not, apply solder to two corner pads on the board, place the chip on and melt those two legs into the solder to connect them. Adjust the position so that all the pins line up correctly. Then solder the rest of the legs, making sure to press down firmly with the iron on each leg just before applying the solder, in hopes that the leg will conduct heat to the copper and then when you apply the solder it will flow across the two.
2. Solder the diodes, the two discrete resistors, the through-hole resistor packs, and the ceramic capacitors.
3. Solder all the tactile switches. Only solder two pins of each switch to start, and **make absolutely sure** every single button is seated fully before soldering the remaining two pins. If the switch isn't seated fully, the 3D printed button will bind and it may prevent the front panel from going together well.
4. Solder all the LEDs which are under buttons or LED pipes. Make sure each is also fully seated, and ideally, pointing as close to straight up as possible. Again, the panel won't go together right if any are sticking out, and buttons may bind if the LEDs under them are a little tilted.
5. For each region of LEDs which stick through the front panel: Insert the LEDs into the board. Stick a couple screws in the area up through from under the board (backwards). Drop a 1/4” nylon standoff/spacer and a thin washer on top of each, and then slide the front panel down onto the screws. Secure them with a nut. Now flip over the assembly and drop/push each LED into the hole in the aluminum. Hold up the assembly and make sure all the LEDs are pushed in uniformly. Solder the LEDs.
6. For the above step, if these are LEDs around an encoder, instead of securing the front panel with screws while soldering, solder in the encoder before you begin the process, and then use the encoder (with its nut) to hold the front panel to the aluminum while pushing in and soldering the LEDs. Make sure of course that the encoder is seated fully! If the encoder's mounting tabs have trouble going through the mounting holes, bend them a bit with pliers–they will eventually fit smoothly. Make sure to use plenty of solder (and plenty of heat) on the mounting tabs.
7. For the LED displays, do a similar process to with the LEDs to ensure they fit correctly in the front panel and are flush with its surface. I used a small piece of wood to hold the displays flush with the surface while soldering them–“pushed in all the way” will not be flush with the surface, it will be below the surface.

8. Make sure to mount the electrolytic capacitors behind the front panel, i.e. on the opposite side from all the other components.

Final Assembly

Once you have finished the board and tested it on its own, it's time to put it all together.

1. Insert all the LED pipes in the aluminum. May take elbow grease as the aluminum “cuts” plastic off the sides of them to fit.
2. Insert long (~5cm) M3 or 4-40 screws through all the holes in the aluminum, in the correct direction. Sit the front panel on your table with the screws sticking up (back side up).
3. Lay all the 3D printed buttons in the appropriate cutouts, making sure to get the LED cutout pointing in the correct direction.
4. Add a thin flat washer and a 1/4“ nylon standoff/spacer to each screw.
5. Slip the front panel PCB down over all the screws. The encoders will hold it up against the table through the holes.
6. Loosely cap each screw with a nut.
7. Pick up the aluminum so that the front panel PCB slips down into all the holes. Press the PCB where necessary and ensure that the LEDs are in their holes and everything is tightly together.
8. Loosely tighten all the nuts.
9. For each screw, loosen and remove the nut, then take another screw and put its tip against the tip of the screw sticking through the assembly. Slide the two screws down through the assembly, so that you have now exchanged the original long screw pointing correctly with a screw pointing backwards (out the front of the panel). Then, take your screw of final length, push it against the tip of the backwards screw, and slide them through, so the screw of final length is in its final position. Add a small lockwasher and the nut, and tighten. The purpose of this is to exchange the long screw for a shorter screw without risking the washer and spacer getting dislodged.
10. Make sure none of the nuts you use are big enough that they touch any of the soldered connections, especially the ones in the middle of the voice selection buttons.
11. Once all the screws are in place, flip over the panel and add the washer and nut to each encoder.
12. Finally, hand-tighten all the screws between the screw head in front and the nut in back.