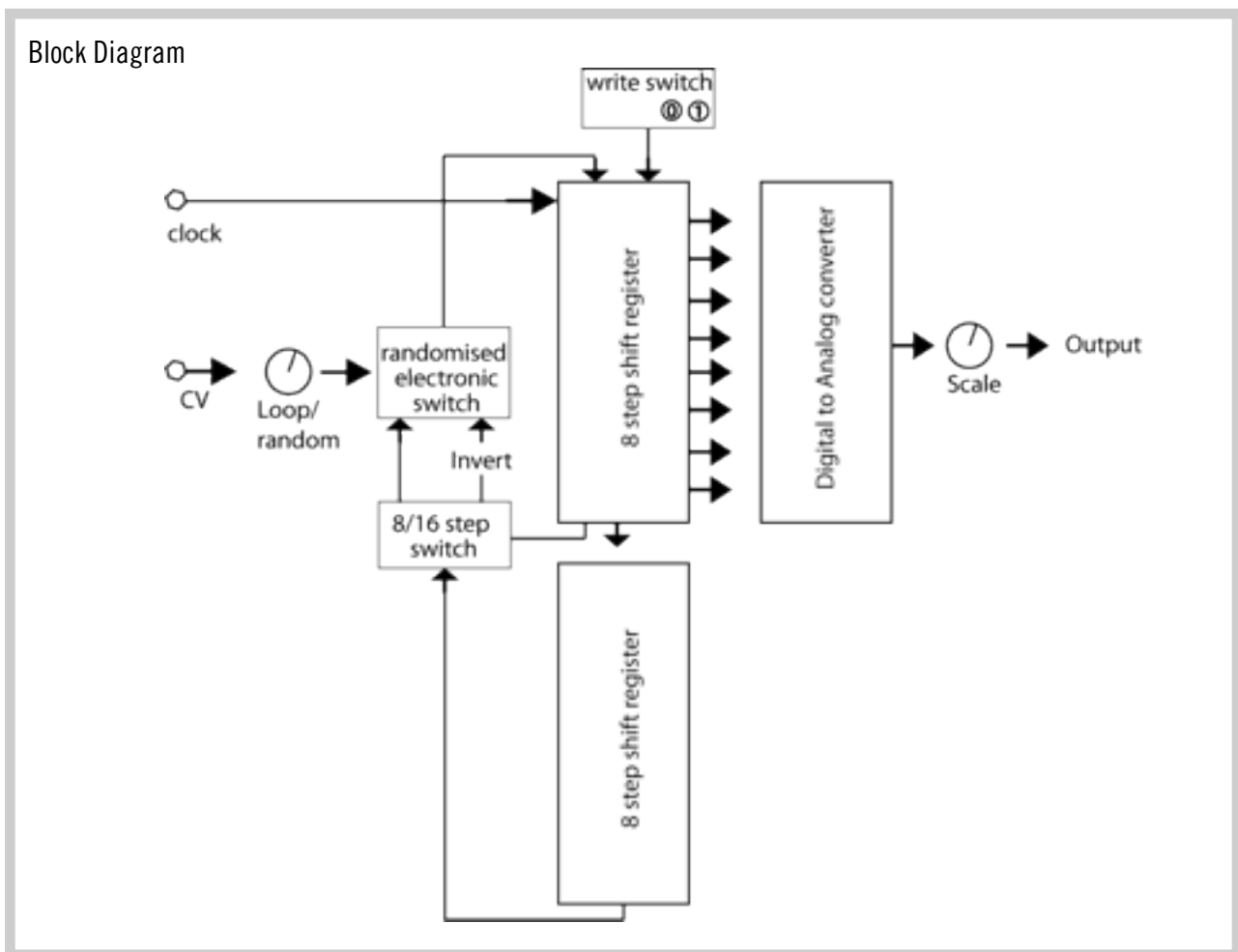


Random Sequencer Documentation

PCB v2
May 2012
by Tom Whitwell

The Random Sequencer is a circuit that produces random control voltages. These can be completely random, or they can be locked into loops that repeat every 8, 16 or 32 steps.

It was inspired by the long history of shift register pseudorandom synth circuits, including the Triadex Muse, Buchla 266 Source of Uncertainty, Ken Stone's gated comparator and Grant Richter's Noising. This circuit includes a transistor noise generator, so should create true random binary sequences. However, the structure of the sequencer and the nature of the digital-to-analog converter used seem to give a particular character - tones that rise and fall in a particular way.



How the module works

This is a binary sequencer, based around a 16 bit memory circuit called a shift register.

NB: You cannot program this sequencer to play specific tunes. You cannot save sequences. You can never go back to a sequence that has changed.

It's designed as a sequencer that you can steer in one direction or another, not one that you can program precisely.

Each clock pulse moves the binary sequence one step through the 16 bit memory. Step 1 is copied to step 2,

and so on. What happens to step 16? It's copied to step 1, but may be transformed along the way.

At any moment, the first 8 of the memory positions are being read by the digital to analog converter.
00000000 = 0 = minimum voltage, around 0v
11111111 = 256 = maximum voltage, around 10v
10000011 = 131 = around 5v

With the main knob at 5pm and the length switch UP, a 16 bit sequence will loop forever.

With the main knob at 5pm and the length switch DOWN the 16 bit sequence is split in half, with the 8th step linked to the 1st step.

Things get interesting when the main knob is not at 5pm. The knob controls a randomised switch connecting the end of the loop to its start.

In a binary system, there are obviously just two options. It can take the last bit and accurately write it into the first bit. Or it can write inaccurately, flipping a 0 to a 1, or a 1 to a 0.

The knob (and the associated CV input) determine how likely each bit is to be flipped:

At 3pm, maybe 1 bit in 10 will be flipped. The loop will change gradually.

At 12 noon, 5 bits in 10 will be flipped. The loop is filled with random data that never repeats.

At 7pm, every bit will be flipped, and the loop becomes a mobius loop.

In effect, the sequence doubles in length. With the length switch UP, 16 steps will play, then those 16 steps will be repeated, inverted. NB: This is a binary inversion, not a melodic or CV inversion. The effect on the output voltage will be hard to predict, but the loop will repeat after 32 steps, not 16.

How the circuit works

The Clock input goes through an Op-Amp Comparator to boost weak clocks. R9 and R3 are a voltage divider to set the trigger voltage.

The two 4015 chips are dual 4-bit shift registers, so there are 4 pins that receive the clock signals.

The output of each shift register is connected to the input of the next via a 10K resistor with a 1n cap to ground, to delay the signal. (This fix is one of many elements drawn from Ken Stone's gated comparator circuit).

Eight shift register outputs are connected to the LEDs, the optional Expansion port and the DAC inputs. The LEDs are not buffered (Ken Stone uses transistor buffers in his Gated Comparator, which is probably a good idea, but it seems to work without them).

The DAC is connected according to the datasheet, driving one gate of the TL074, with a simple attenuator pot on the output.

The noise circuit is a basic reverse-bias transistor noise circuit, boosted by inverting op amp stages with capacitors to remove the offset. The front panel noise output is not independently buffered, so it's possible using this output could have an impact on the rest of the circuit in some situations.

The switch part of the circuit works like this: White noise flows into a comparator, with the reference

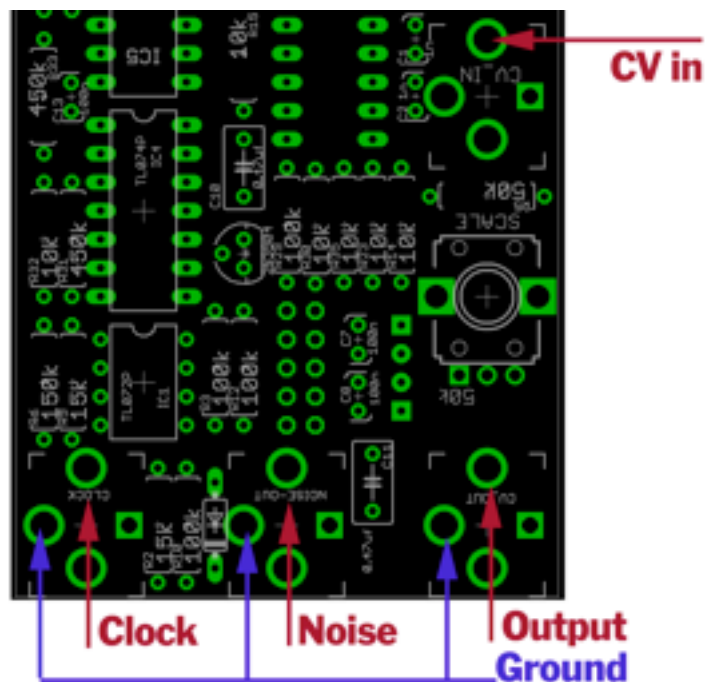
voltage set by a combination of the main knob and the CV input (summed by IC1A). Using a transistor inverter, two gates of the 4016 are set up as a SPDT switch, routing the end of the shift register loop either straight back to the start of the loop, or back via another transistor inverter.

Bill of Materials

Most of the components are straightforward off-the-shelf parts, but the parallel PCB construction means some components are very specific.

Sockets: The board is designed for 3.5mm vertical jacks from <http://erthenvar.com/> model PJ-301b. If you're not using these sockets, the lug nearest the top of the board carries the signal.

The DAC0800 is also known as a DAC-08. This part of the circuit could be replaced by R-2R resistor ladder, as seen in Ken Stone's Gated Comparator.



The LED Bar Graph module could easily be replaced on this PCB by individual LEDs, which might be easier to mount in a drilled panel - put the shorter LED legs towards the edge of the board. The LED current protection resistors can be adjusted according to the LEDs used.

I've used an eight way 330 ohm resistor network (RN1) for ease of installation, but it should be possible to hack one together - 8 resistors connected to one wire going to ground at the pin connected with a cross (a neat example here: <http://flic.kr/p/65UmMd>).

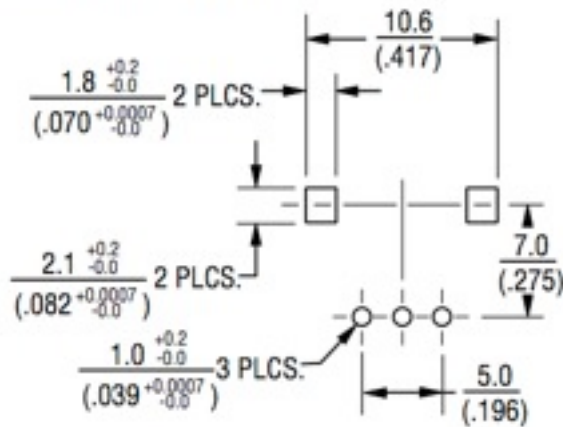
R19 is there to increasing the total resistance of RN1 (Without it, the circuit draws rather high current, but you could use a jumper if you have a higher value of RN1). The Clock LED has it's own current limiting resistor, R13.

I used 1% resistors in many places, because I had them handy. I do not think any of the component values are very critical - nothing needs to be trimmed or calibrated.

Potentiometers: Both pots are voltage dividers, any reasonable value should work. I've used 50k & 100k.

The PCB is designed for 9mm vertical pots from Alpha, Bourns or other manufacturers like this:

RECOMMENDED PCB LAYOUT



I used one with a 25mm D shaft for the main control, and a 12mm knurled for the smaller control.

Switches: These red-bodied solder tag switches are pretty common. Two ranges are Salecom T80-T switches from Rapid or Bitsbox, or E-Switch 100 range from Mouser.

The LENGTH switch is a standard SPDT ON/ON switch.

The WRITE switch MUST be centre off SPDT - and ideally a momentary (ON) OFF (ON).

Expansion Port

The expansion port surfaces the clock pulses, and the 8 shift register outputs that go to the DAC. It is not buffered.



Pinout:

1-8: Data channels from the shift register. 1 = Most significant bit, 8 = Least significant bit

9: Ground

10: Clock

Building the circuit

I've found the 4015 shift register chips to be slightly unpredictable. If the sequence won't loop properly, swap the chip and throw out the old one. I've no idea why this happens, but Navs reported the same thing building his Bitsy. This is why the two 4015 chips are on the back of the board, not hidden behind the panel.

Quirks

Feeding the module from a mulled clock input can sometimes give strange results - clearing a looped sequence. I've no idea why this happens.

Hardware, pots, switches, lights:

1	5+5 pin 0.1 inch Power Header	(Mouser 649-69190-410HLF)
1	10 pin 0.1 inch header	(Mouser 649-67997-210HLF)
1	Length Switch: SPDT ON-NONE-ON	(Mouser 633-M201202)
1	Write Switch: SPDT (ON)-OFF-(ON)	(Mouser 633-M201802)
2	50k ALPHA-9MM vertical Pot	CHANGE, SCALE
(Main 'change' control: 25mm D Shaft Mouser 652-PTV09A-4025FB503 Smaller 'scale' control: 12mm knurled shaft Mouser 317-2090F-50K)		
4	PJ-301B	3.5mm socket (from erthenvar.com)
1	DC10	Kingbright DC10 LED Bar Graph (Mouser 604-DC-10EWA)
3		16 pin DIL socket (.3 inch)
2		14 pin DIL socket (.3 inch)
1		8 pin DIL socket (.3 inch)

Capacitors:

2	0.47uf	Polybox Cap	C10, C11
4	1n	Ceramic Cap	C1, C2, C3, C4
1	10n	Ceramic Cap	C9
2	10uf	Electrolytic Cap	C5, C6
5	100n	Ceramic Cap	C7, C8, C12, C13, C14

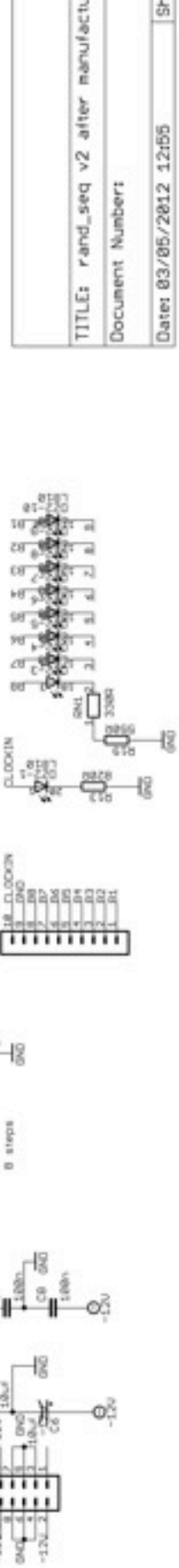
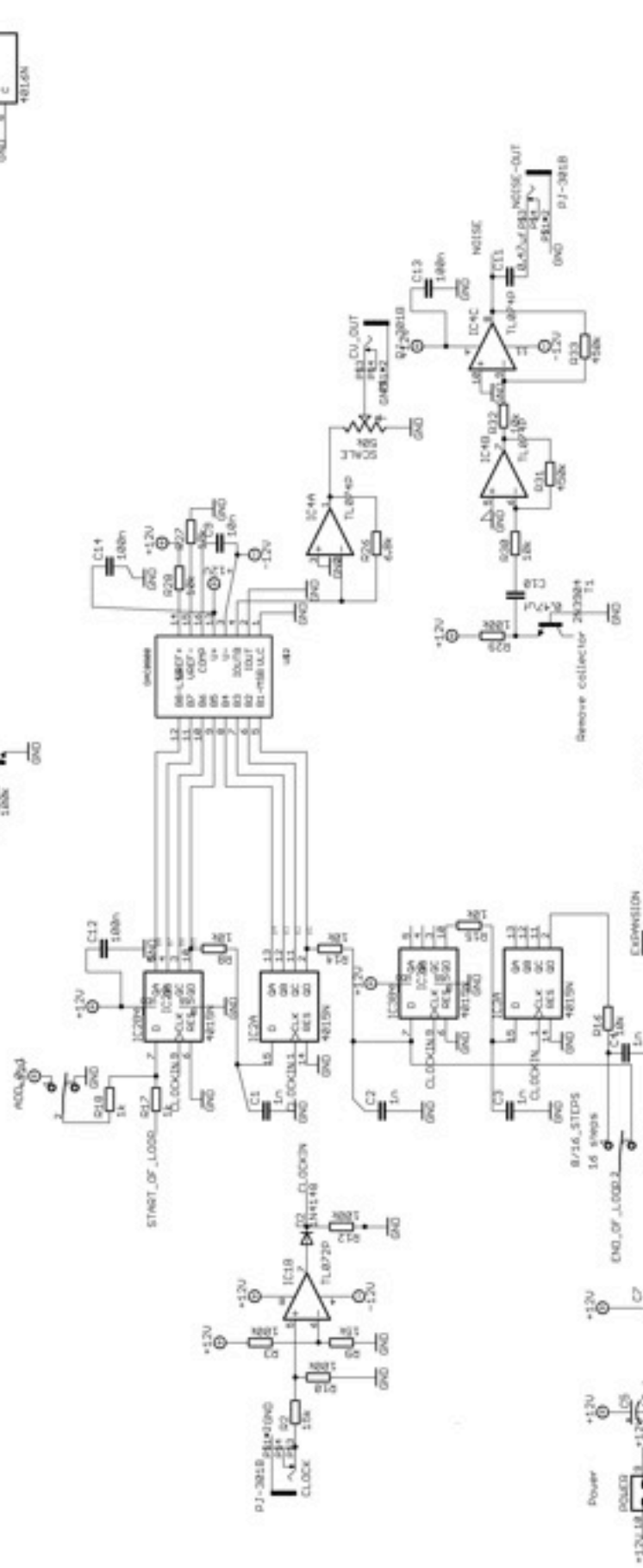
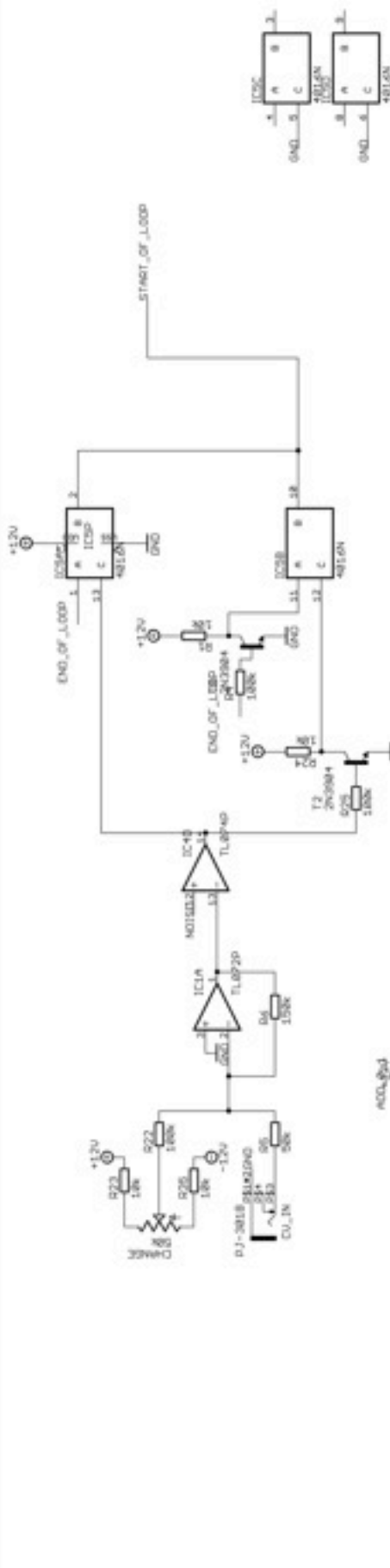
Resistors:

2	1k	Resistor	R17, R18
1	6.8k	Resistor	R26
12	10k	Resistor	R1, R8, R14, R15, R16, R23, R24, R27, R28, R30, R32, R35
2	15k	Resistor	R2, R9
1	51k	Resistor	R5
7	100k	Resistor	R3, R4, R10, R12, R22, R25, R29
1	150k	Resistor	R6
1	330R	8-commoned Resistor Network	RN1
2	470k	Resistor	R31, R33
1	560R	Resistor	R19
1	820R	Resistor	R13

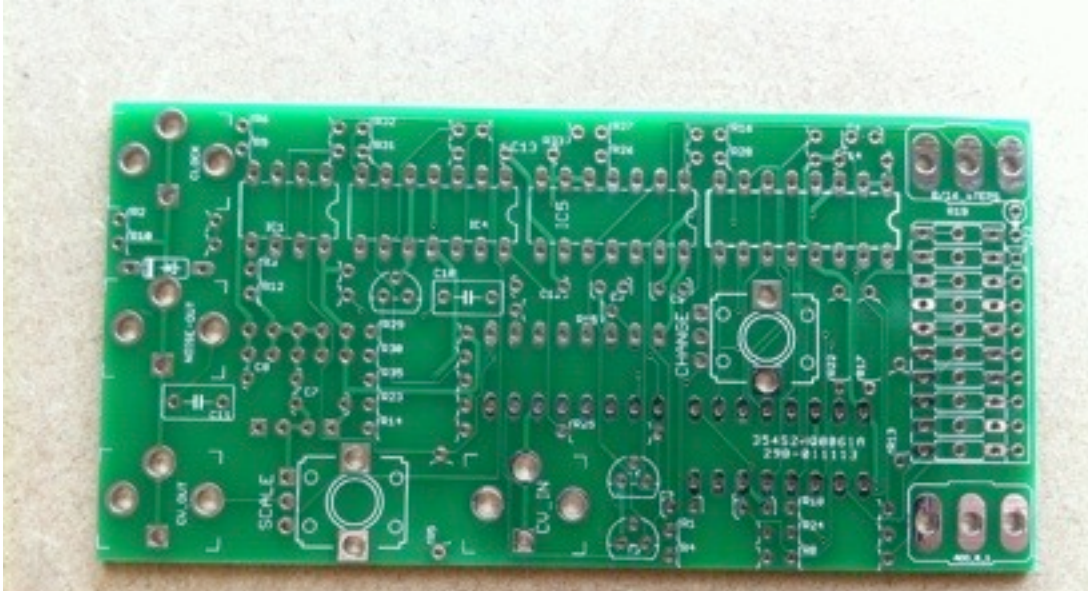
Semiconductors:

1	1N4148	1N4148 Diode	D2
2	4015	Shift Register	IC2, IC3
1	4016	Bilateral Switch	IC5
1	DAC0800	Digital/Analog Conv	U\$2
1	TL072	TL072	IC1
1	TL074	TL074	IC4
3	2N3904	Transistor: 2N3904	T1, T2, T3

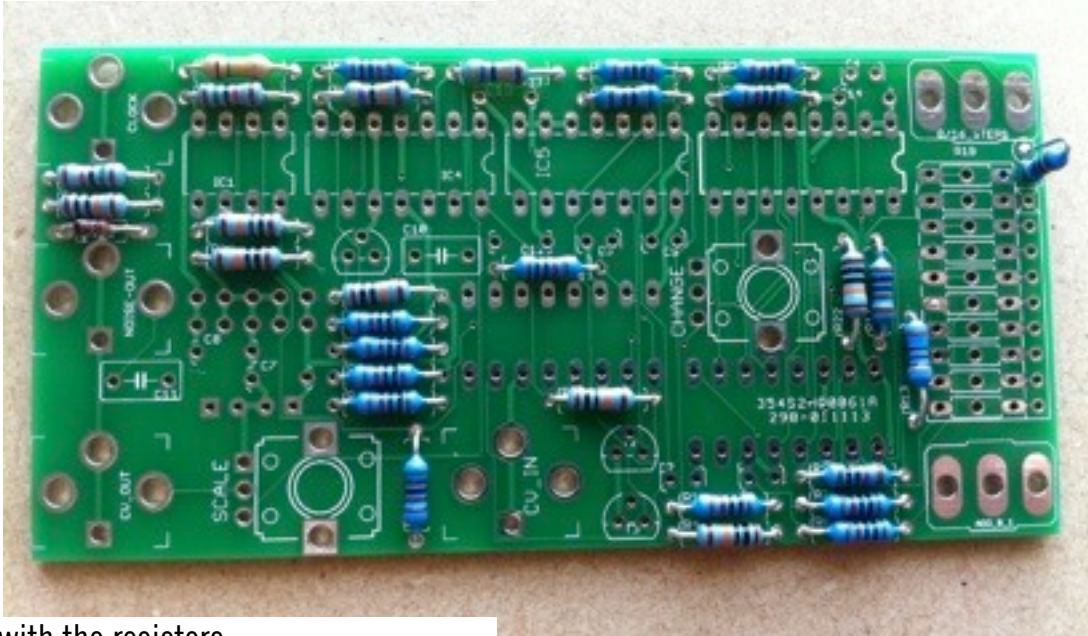
Full Mouser Project: <http://bit.ly/MRGqRF> access id: bc8a4a43bf



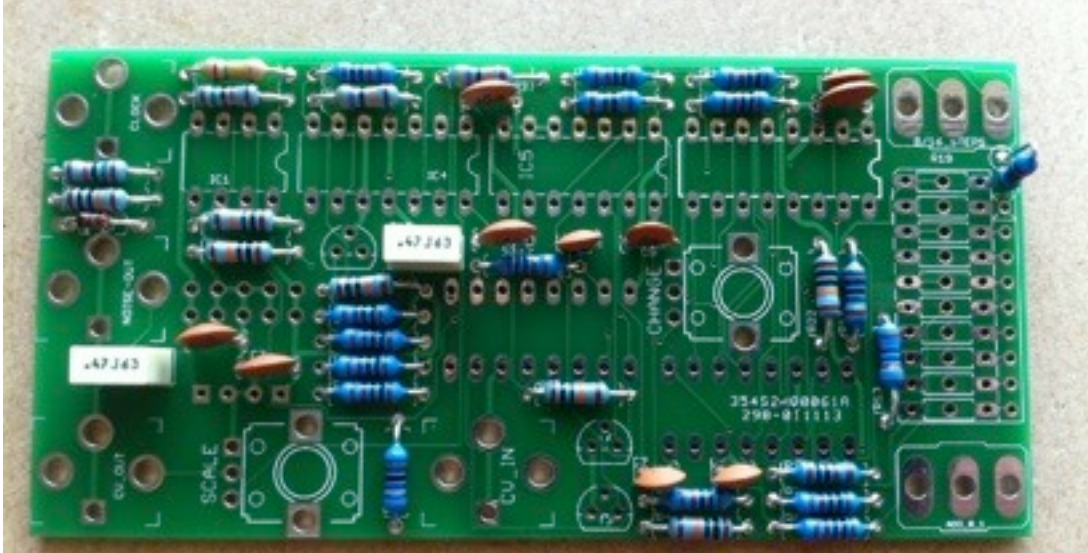
TITLE: rand_seq v2 after manufacture
Document Number
REV: 1
Date: 03/05/2012 12:55
Sheet: 1/1



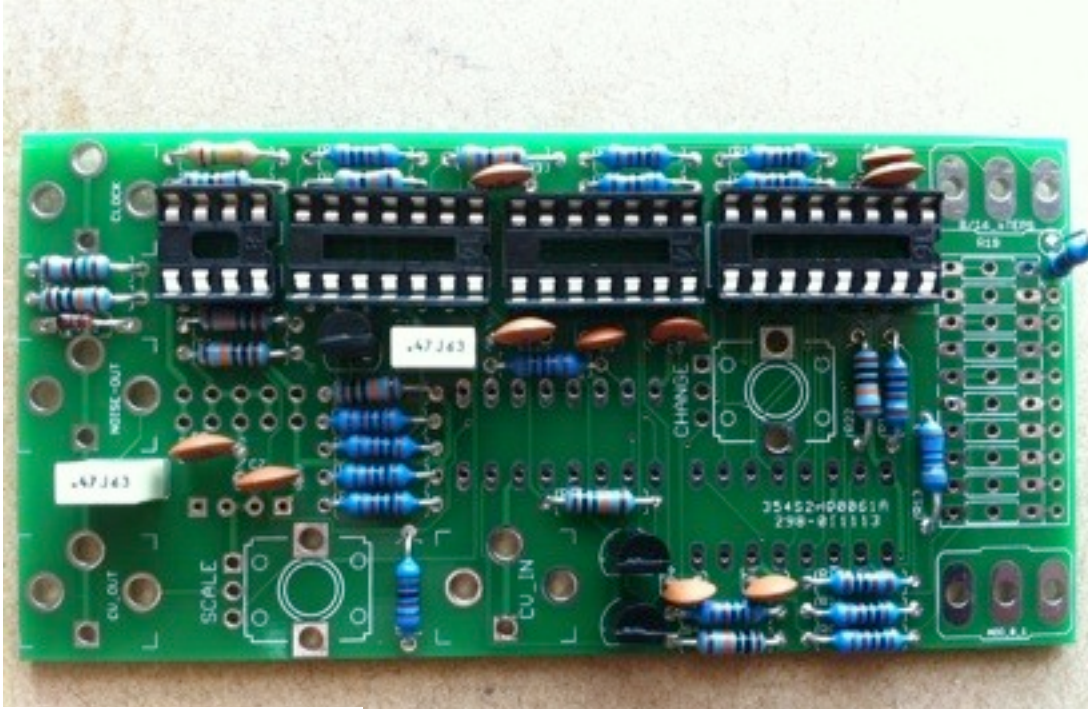
The top of the board...
... where most of the components go. In this picture, the component values are missing from the silkscreen, but they should be there if you make a board from the Gerber files



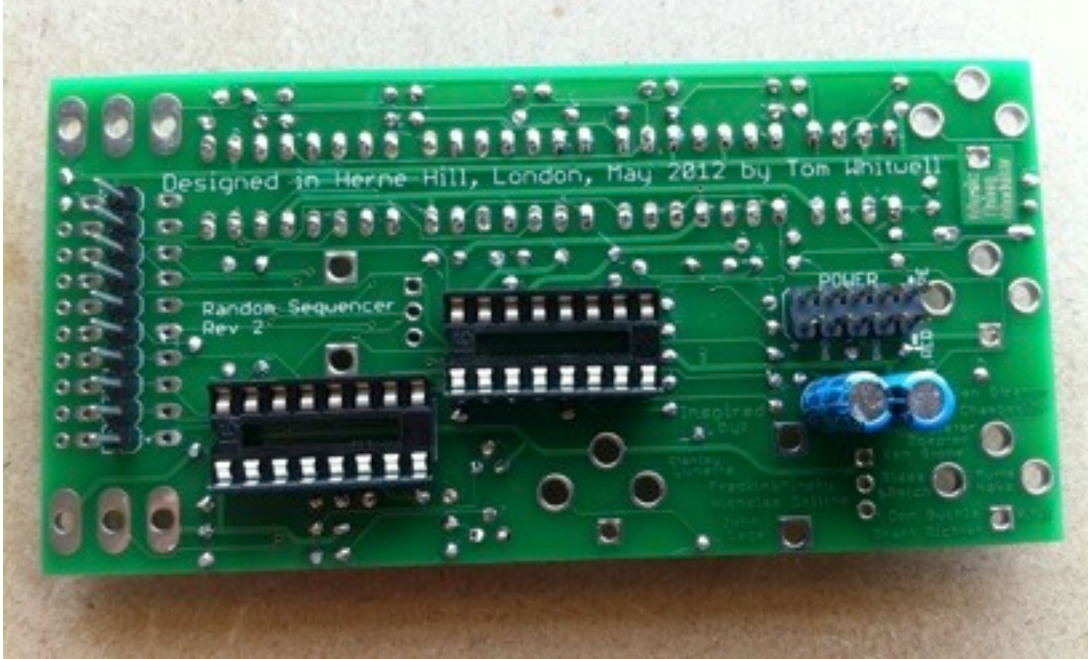
Start with the resistors
I usually check each one with a multimeter, because I can't read the coloured bands



Add the capacitors.
C5 doesn't fit very well



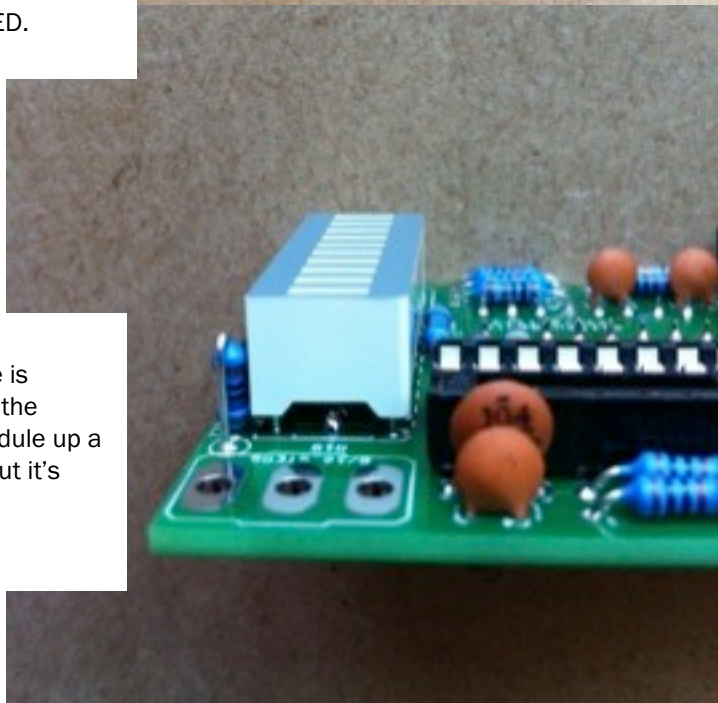
IC Sockets and Transistors on the top of the board

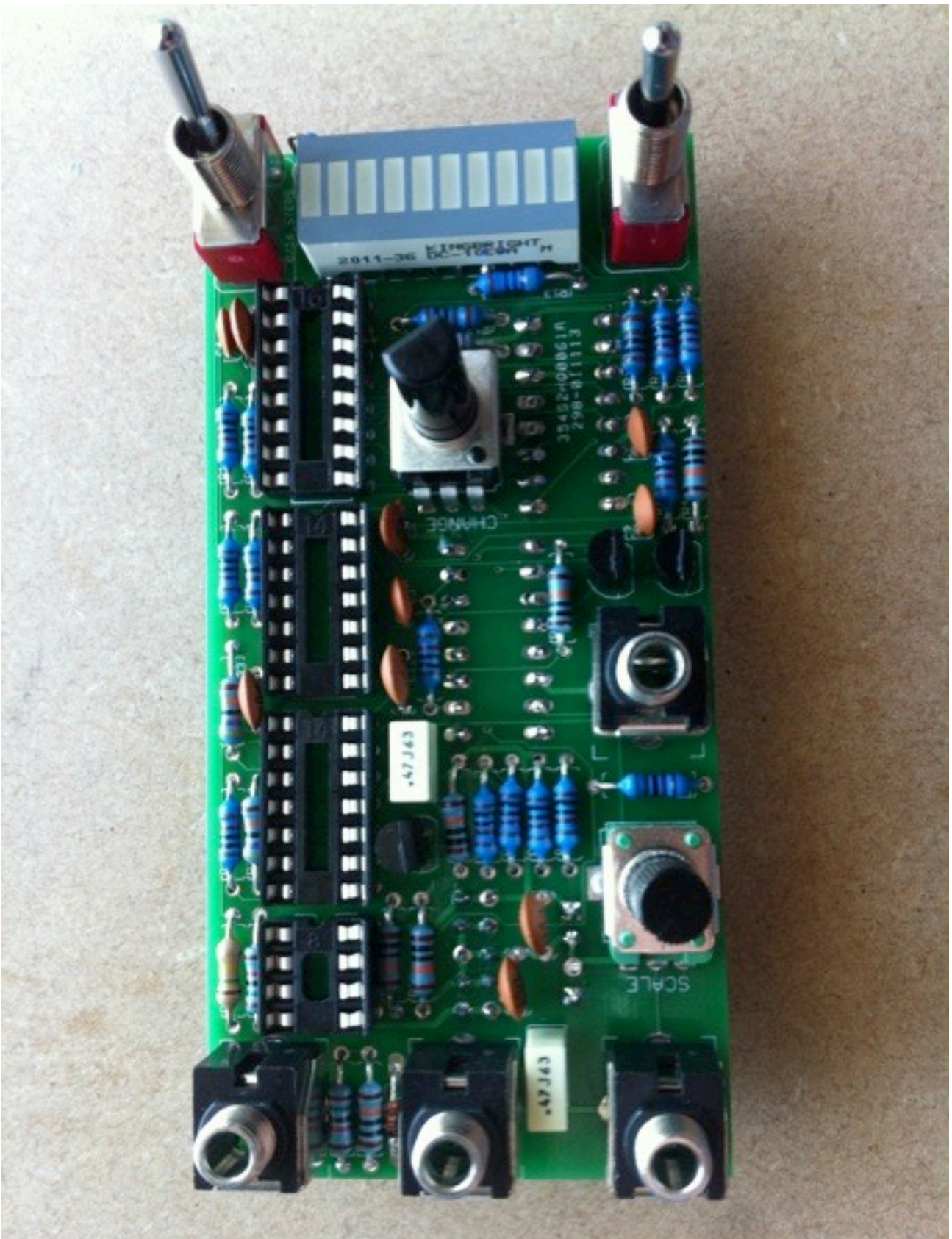


The back of the board
The expansion header is optional -but you won't be able to add it later, as it fits under the LED.

LED Bar Graph

One corner of the LED bar graph module is flattened - make sure you align this with the silkscreen. You might want to lift the module up a little, so it's closer to the acrylic panel, but it's certainly not necessary.





The big bits

Add the switches, pots and sockets. The sockets are fiddly - try to hold them in place, tack some solder on, then add plenty more solder to keep them secure

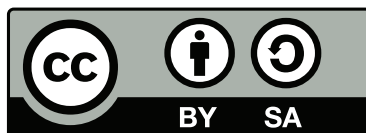
Soldering complete.

At this stage you may want to power up the module and use a multimeter to check the expected +12v, -12v, GND connections to the IC sockets, according to the data sheets.

Once you're happy, insert the ICs. They go, from the LED down: DAC0800, 4016, TL074, TL072, with the two 4015s on the other side. The 4016 and TL074 are rather snug, but they do fit in.

Open Source Hardware

This is an open hardware project. You are free to build this hardware, sell it or make any changes to it, so long as you share your changes and release them under the same license. You should also credit me, Tom Whitwell. All of the original design files (Eagle CAD, Illustrator etc) are shared. These files are licensed under a Creative Commons Attribution Share-Alike license, which allows for both personal and commercial derivative works, as long as they credit me and release your own interpretation under the same license.



This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/3.0/> or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.