

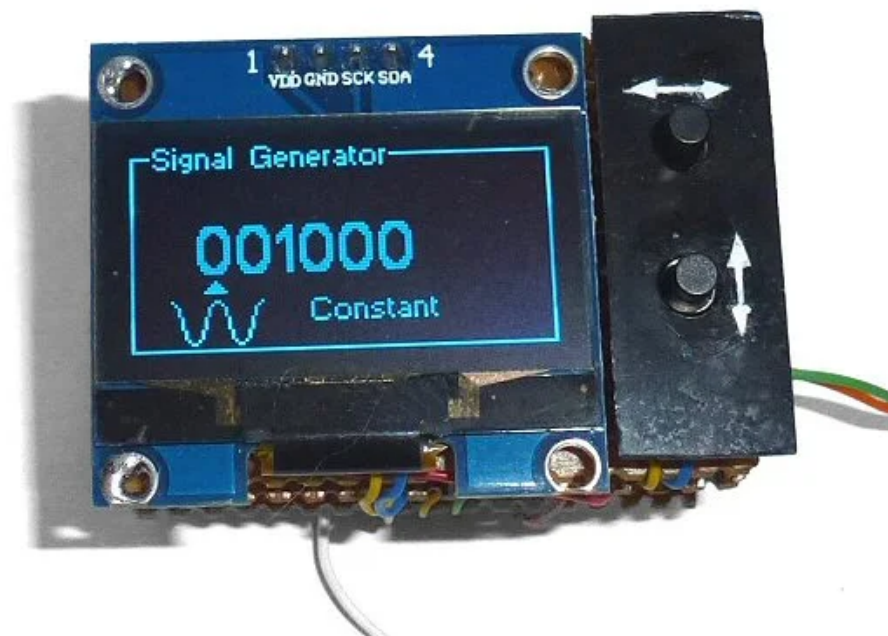
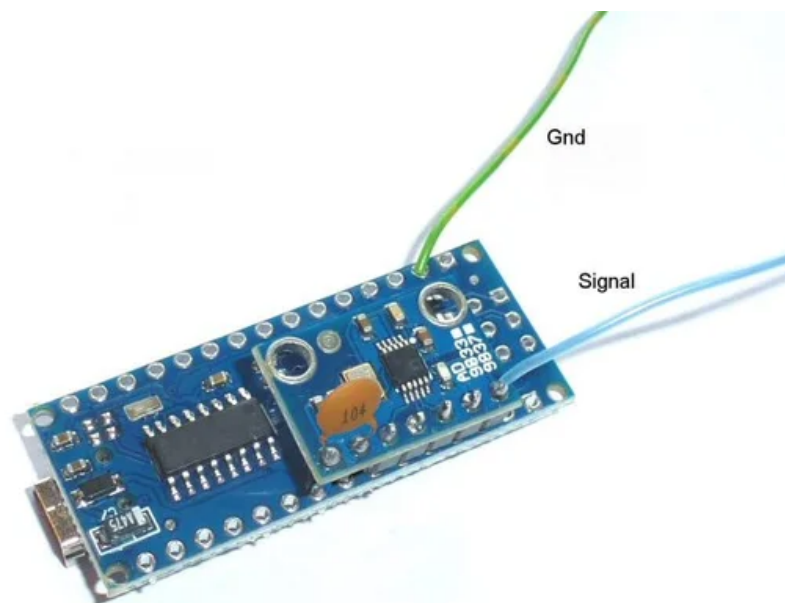
AUTODESK
Instructables

Signal Generator AD9833

By [Peter Balch](#) in [CircuitsElectronics](#)



Introduction: Signal Generator AD9833



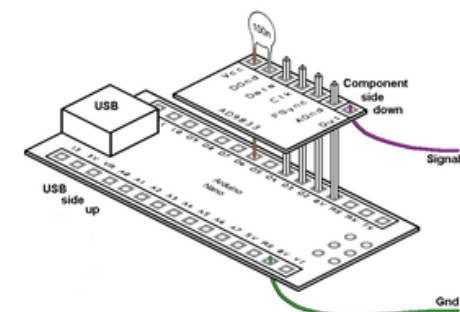
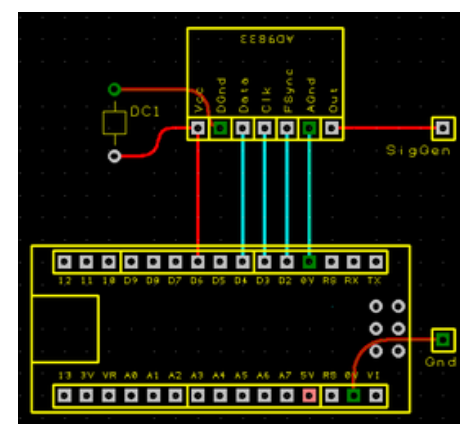
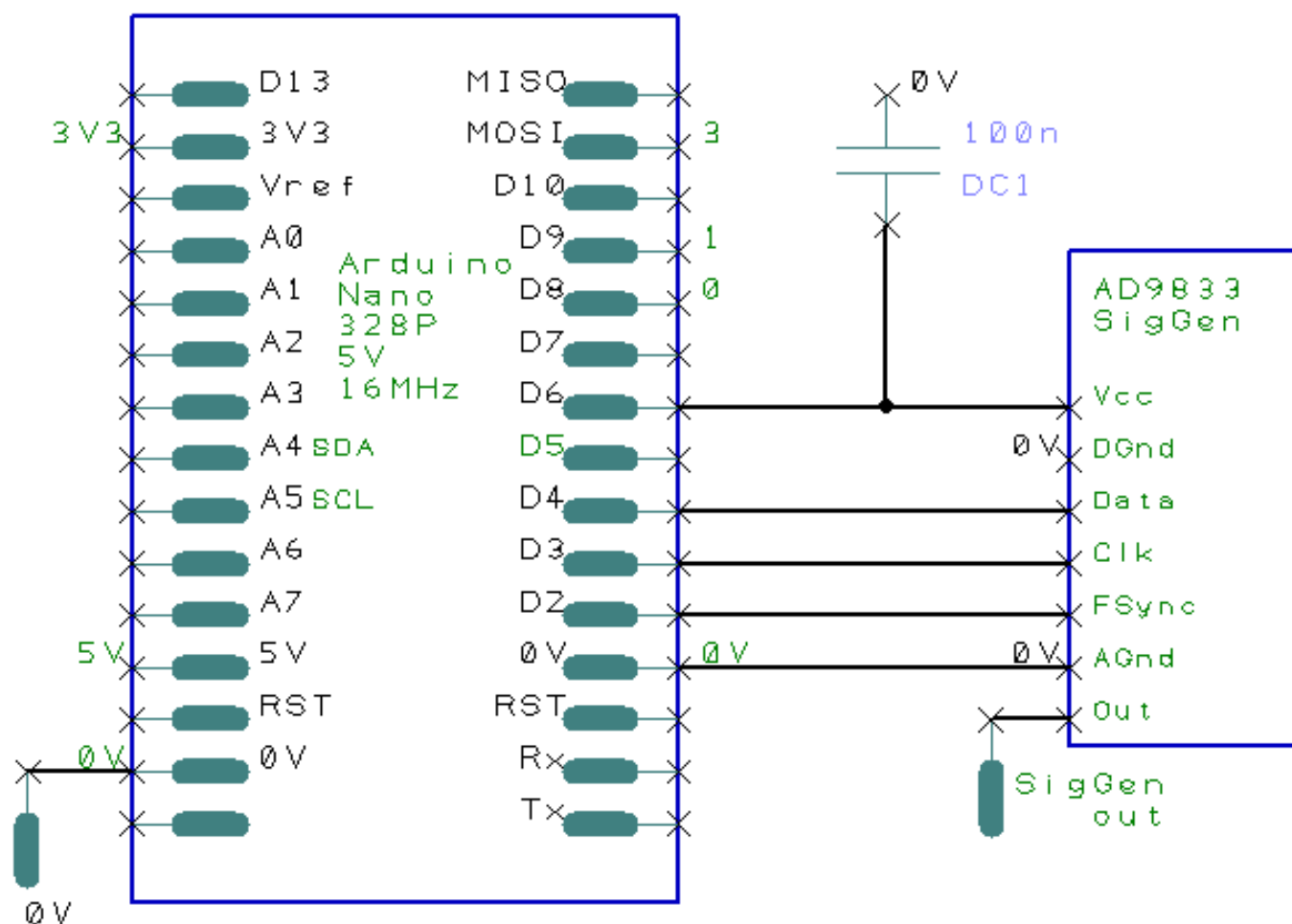
A signal generator is a very useful piece of test gear. This one uses an AD9833 module and an Arduino Nano - that's all, not even a PCB. You can optionally add an OLED display. The AD9833 can generate sine, triangle and square waves from 0.1 Hz to 12.5 MHz - the software in this project is limited to 1Hz to 100kHz.

There have been other Instructables using an Arduino and an AD9833, [here](#) and [here](#). This is simpler and can be used as a sweep generator. Sweep generators help test the frequency response of filters, amplifiers and so on. Unlike the other Instructables designs, t

this does not include an amplifier or amplitude control but you could add them if you wanted.

Step 1: Simplest Signal Generator

SigGen with no display



For the simplest Signal Generator, you just solder the AD9833 module onto the back of the Arduino Nano. No PCB is needed.

The AD9833 module I chose is similar to [this one](#). I'm not saying that's the best or cheapest supplier but you should buy one that looks like that photo (or the photo above).

The connections between the modules are:

- grounds connected together
- D2 = FSync
- D3 = Clk
- D4 = Data
- D6 = Vcc of AD9833

The AD9833 is powered from data pin D6 of the Arduino - the Arduino can supply sufficient current. I've added a 100nF decoupling capacitor because I thought I "ought" to but I couldn't see any difference - there is already a decoupling capacitor on the AD9833 module board.

If you were being fancy, you might worry about "analogue ground" vs "digital ground" but if you were being fancy, you'd be spending more than £4.

The simplest Signal Generator is controlled and powered over a USB lead from a PC. The USB emulates a serial port running at 115200bps (8-bits, no parity). The commands are:

- '0'..'9': shift digit into "min" frequency array
- 'S': set AD9833 frequency and produce sine wave
- 'T': set frequency and produce triangle wave
- 'Q': set frequency and produce square wave
- 'R': reset the AD9833
- 'M': copy "min" frequency array into "max" array
- 'G': sweep from "min" to "max" over 1 second
- 'H': sweep from "min" to "max" over 5 seconds
- 'I': sweep from "min" to "max" over 20 seconds

The Arduino program contains two 6-character arrays "min" and "max". If you transmit a digit then it is shifted into the "min" array. If you send an 'S' then the "min" array characters are converted into a longint frequency and sent to the AD9833. So sending the string

002500S

will set the AD9833 output to a 2500Hz sine wave. You must always send all 6 digits. The minimum frequency is 000001 and the maximum frequency is 999999.

If you send an 'M' then the "min" array is copied into the "max" array. If you send an 'H' then the AD9833 repeatedly outputs a gradually increasing frequency over 5 seconds. It starts at "min" frequency and 5 seconds later is at "max" frequency. So

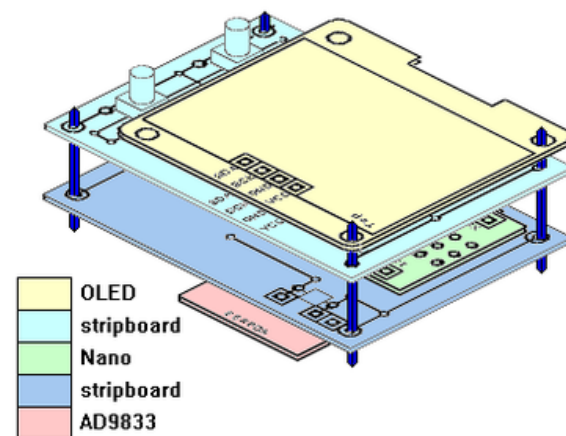
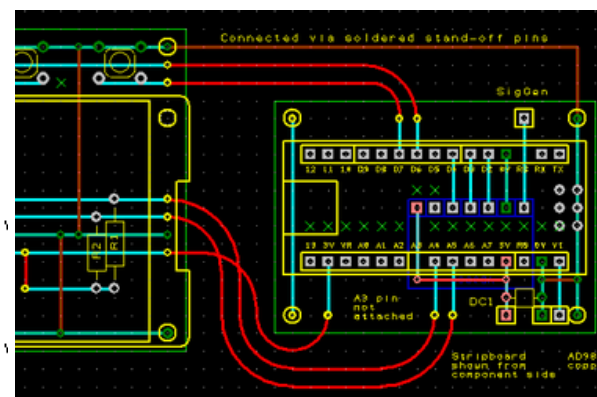
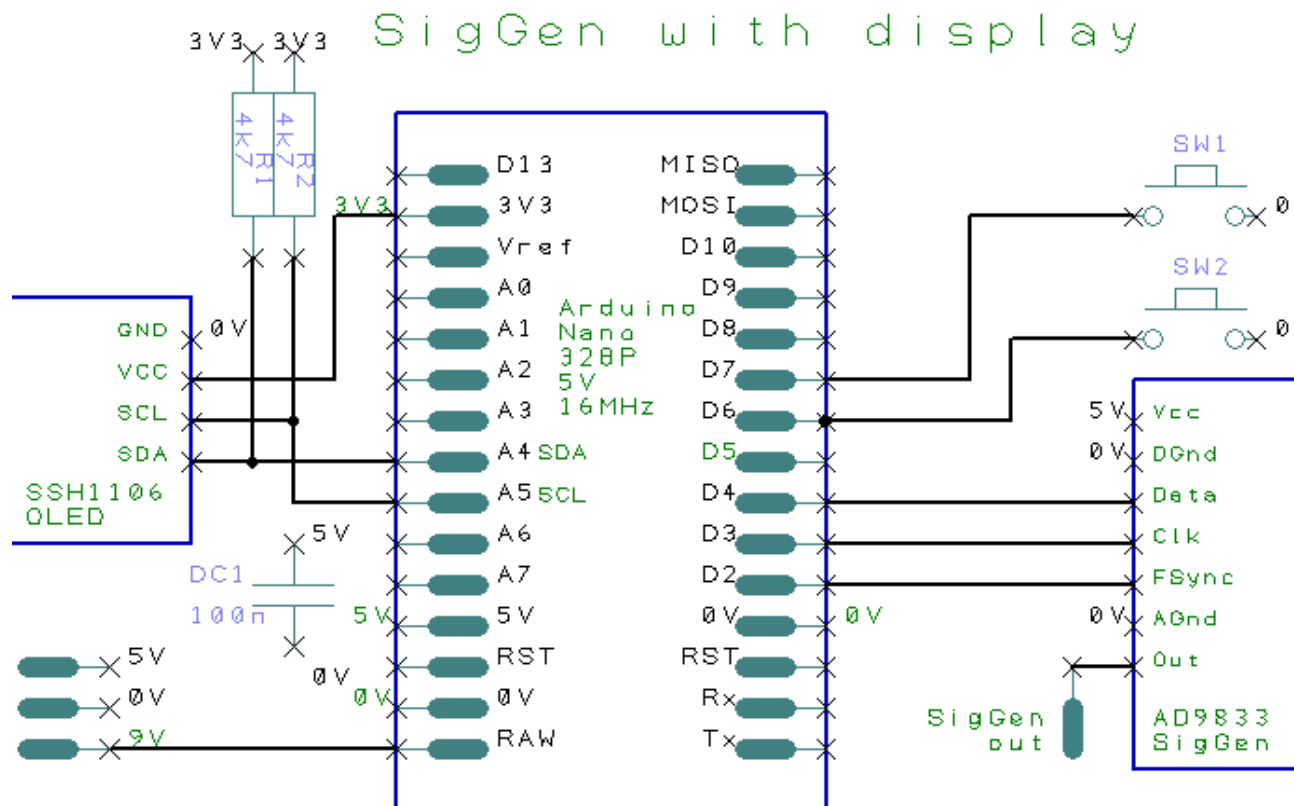
020000M000100SH

sweeps from 100Hz to 20kHz. The frequency change is logarithmic so after 1 second the frequency will be 288Hz, after 2 seconds 833Hz then 2402, 6931 and 20000. The frequency is changed every milliSecond.

The loop stops when the Arduino receives another character so be careful not to send the command followed by carriage-return or line-feed. That extra character would terminate the loop. If you're using the Serial Monitor, there's a box at the bottom right that might say for instance "Both NL & CR" which (I think) sends characters after your command. Set it to "No line ending".

You can download the Windows EXE program below which will send the required commands or you could write your own. The Arduino INO file is also here.

Step 2: Add an OLED



If you add an OLED and two buttons, the signal generator can work alone without a PC.

Those of you who have read my [oscilloscope Instructable](#) will recognise the similarity. The AD9833 module can be added to my oscilloscope to produce an "Oscilloscope and Signal Generator in a Matchbox".

The display is a 1.3" OLED running at 3.3V which is controlled by an SH1106 chip via an I2C bus.

Search eBay for 1.3" OLED. I don't want to recommend a particular seller as links quickly go out of date. Choose one that looks like the photo, says "I2C" or "IIC" and has four pins labelled VDD GND SCL SDA. (Some displays seem to have the pins in a different order. Check them. The proper name for the clock of I2C is "SCL" but on eBay the boards can be labelled "SCK" like my one in the photo.)

A fuller description of the OLED library is in my oscilloscope Instructable in Step 8. You should download and install the driver library SimpleSH1106.zip which is in Step 8. (I don't want to upload another copy here and have to maintain two copies.)

The INO file can be downloaded below. The pin numbers used for the OLED are declared around line 70. If you have built my "Oscilloscope and Signal Generator in a Matchbox" and want to test this INO file with it, alternative pin numbers are enabled via a #define.

I've shown a stripboard layout for the circuit. There are two stripboards - one for the Nano and the AD9833 and one for the display. They should form a sandwich. The boards are shown from the component side. Fine flexible wires join the two boards. Attach the boards together with soldered stand-offs. In my diagram, the copper of the stripboard is shown in cyan. Red lines are wire links on the stripboard or flexible wires joining the boards together. I haven't shown the power and "signal" leads.

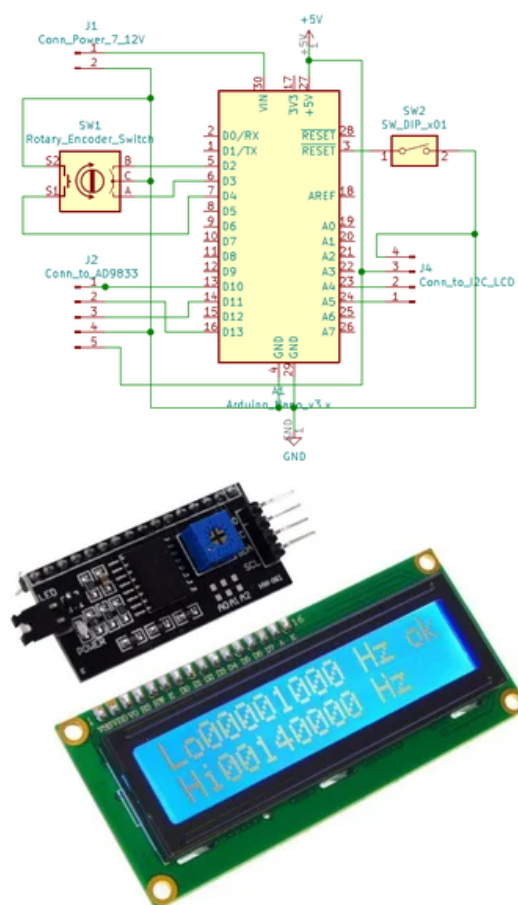
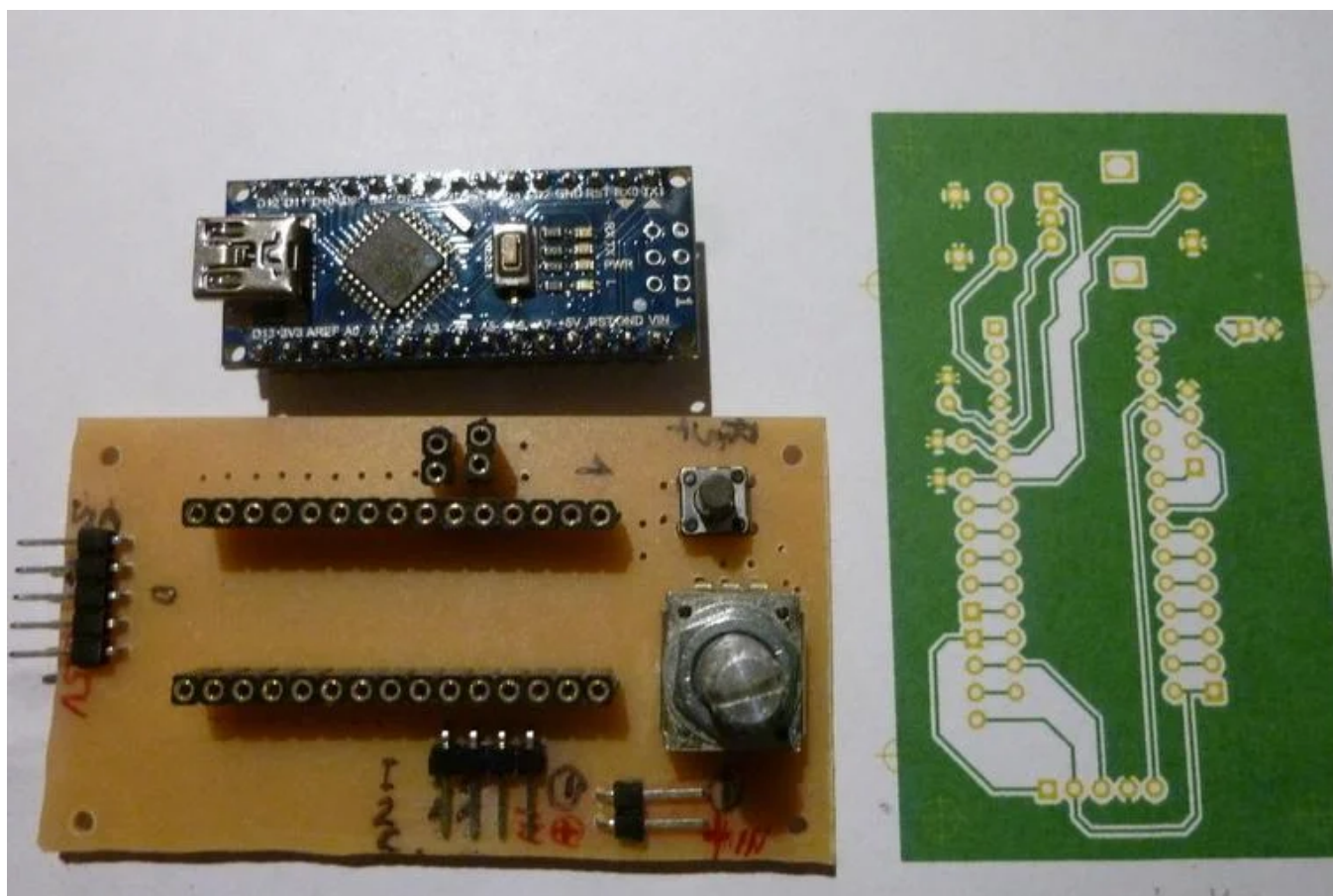
The AD9833 module is soldered on the copper side of the stripboard - on the opposite side from the Nano. Solder pins onto the copper strips then fit the AD9833 onto them and solder it on.

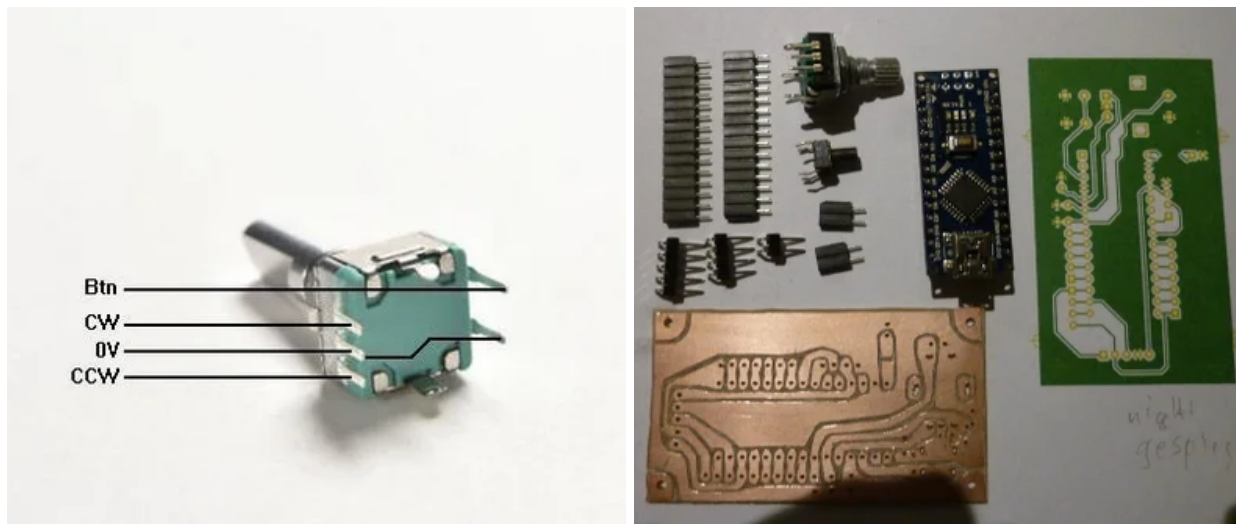
The display shows either a single frequency or the "min" and "max" frequencies.

There are two pushbuttons: a "Horizontal" button to select a digit of the frequencies and a "Vertical" button to change that digit.

I power the signal generator from the circuit I'm developing - I always have 5V available at my workstation.

Step 3: Using a Two-Line LCD





[mausi_mick](#) has redesigned the circuit to use a 2-line LCD and a rotary encoder instead of the OLED and pushbuttons. This Step describes his design. You can see a video of it [here](#).

He has used a LCD1602 with an I2C interface piggyback board (approx £5 total). They are a simple way of connecting a LCD1602 to an Arduino and require only three of the Arduino's pins.

Alternatively, you could use a plain LCD1602 (£2.50) which requires 6 Arduino pins. [Here's an Instructable](#).

He uses a rotary encoder which makes setting the menus much quicker. Turning or clicking the encoder triggers a "Pin-Change-Interrupt".

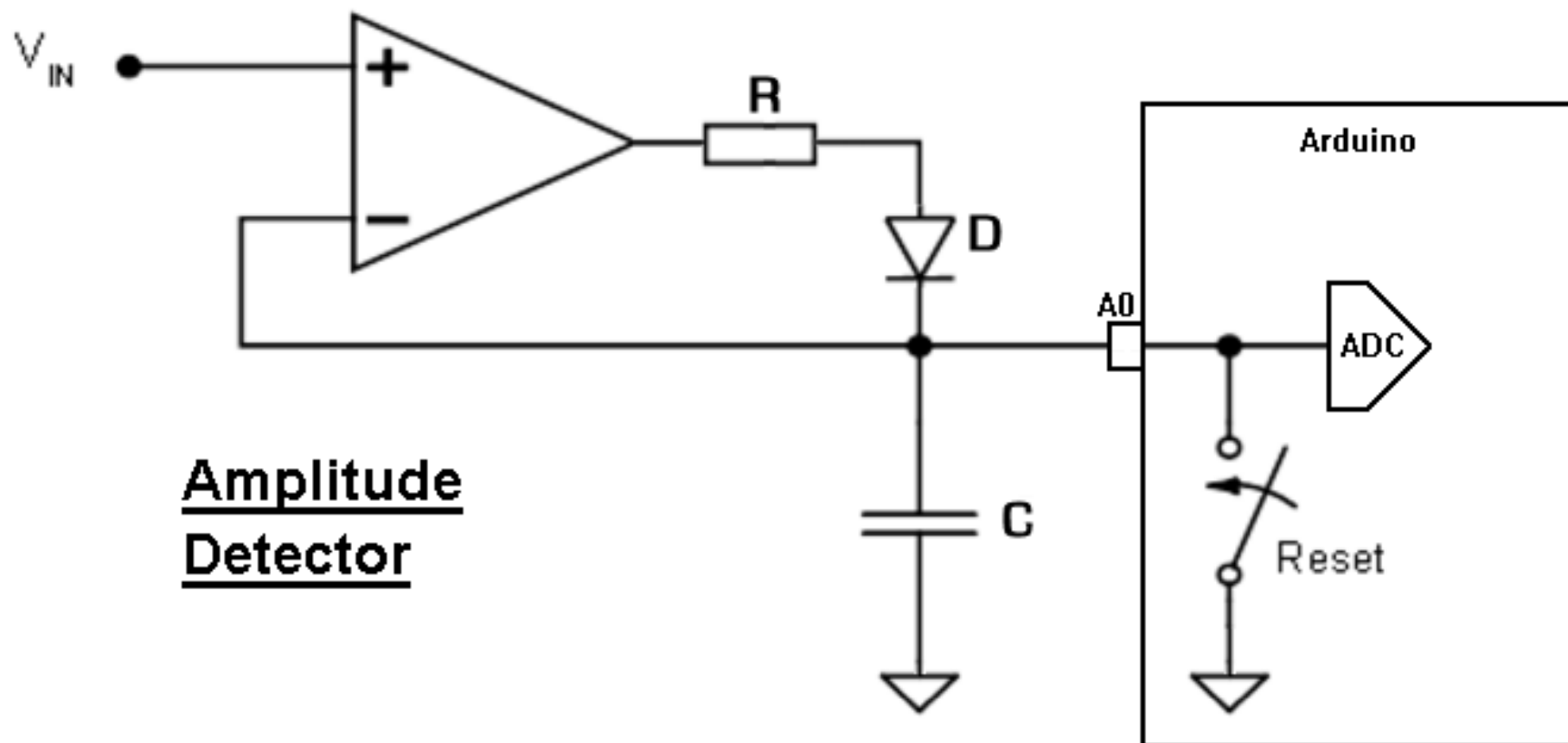
The menu has two new functions:

- Changing frequency (Low and High) from 0 ... 12 MHz
- Changing delay (from 100µs,200µs,500µs,1ms...1s) (sweep-mode).

The signal generator currently is powered from a USB-supply but he intends to change that to a Li-ion battery with a boost converter to give 5V (using a MT3608 SMPS). USB supplies have a lot of noise so the battery will improve the quality of the signal. Of course, a boost converter might also create noise. Personally, I'd use 4 AA cells or two Li cells and no booster.

The sketch can be downloaded below. A KiCAD PCB layout is available of his [GitHub page](#).

Step 4: Future Developments



Could it be battery powered? Yes, just add a 9V PP3 connected to the RAW pin of the Nano. It typically uses 20mA.

Could it be powered by a single lithium cell? I don't see why not. You should connect the OLED Vdd and its pull-up resistor to the 3.7 V battery (I doubt if the 3.3V output of the Arduino would work properly).

A sweep generator is more useful when testing the frequency response of a filter if you can graph amplitude vs frequency. Measuring the amplitude of a signal is tricky - you have to trade off the decay of your envelope detector vs ripple for low frequencies and response time for high frequencies. Having built your amplitude detector, you could feed its output into the ADC of the Arduino of the "Simplest Signal Generator" then send the result, along with the current frequency to the PC.

[This page](#) is a useful starting point or search Google for "envelope detector" or "peak detector". In the suggested circuit above, you would set the signal frequency, wait for it to stabilise, set the Arduino A0 pin to output digital low, wait to discharge C, set A0 to input, wait, then measure with the ADC. Let me know how you get on.