A few years ago I needed a function generator for my home laboratory. In my job I had worked with some expensive commercial models and initially I had planned to buy one of these. However, none of them really was what I wanted – too complex for simple use – and so I decided to design and build my own. It turned out to be DDS (direct digital synthesis) based but that was not the only component selection issue I ran into. Here are a few more.

My components, your components?

After researching various techniques I settled on a Direct Digital Synthesis (DDS) based architecture for my project. DDS employs a digital oscillator with quartz crystal precision to accurately generate sinewaves up to very high frequencies. For the microcontroller I chose one from Analog Devices. Although well known for their Digital Signal Processor (DSP) families, this may not be the first MCU manufacturer that comes to mind. They do have a nice family though of 32-bit ARM controllers supported by something I like a lot: comprehensive documentation. Unlike other MCU manufacturers that sometimes need more than one thousand pages to elucidate their devices, Analog Devices manages to fit a complete description of a complex microcontroller into a document of slightly more than a hundred pages. Analog Devices also specializes in DDS chips, operational amplifiers and other analog support chips and so made for a great one-stop shop for this project. Their friendly sampling service quickly got me started without forking out a lot of money.

The MCU I selected was the ADuC7024B-STZ62, a member of the Precision Analog Microcontroller family in a 64-pin package, containing an ARM7TDMI core running at 44-MHz clock speed. These MCUs are called analog because they feature analog inputs and outputs (yup, ADC and DAC) and an analog comparator. Besides, they sport PWM, timers and standard serial ports (SPI, UART, I²C). Our micro has 8 KB of RAM and 62 KB of flash memory that can be programmed in-circuit over a serial port. Note that some types use I²C for flash memory programming, so make sure you get the specified type and not one with a slightly different part number.

The DDS chip for the project is the popular AD9834. The maximum frequency of the external oscillator is 75 MHz, allowing a maximum output frequency of 37.5 MHz (half the clock frequency). The downside of such a fast clock signal is a frequency resolution of just 0.28 Hz owing to the chip’s 28-bit integer frequency divider. That doesn’t sound like a big deal, but it does equate to almost 0.6% at 20 Hz. To improve this, the clock frequency can be lowered, albeit at the expense of the maximum achievable output frequency. At 1 MHz for instance the resolution becomes 0.004 Hz, which corresponds to an error of 0.005% at 20 Hz, but the maximum frequency is then down to a measly 500 kHz. In this project the DDS clock frequency is 75 MHz to ensure a relatively clean signal up to 10 MHz; resolution was traded against range.

The user interface is an important part of any instrument and so I spend a lot of attention on this. I added a graphic display and a bunch of pushbuttons to allow easy navigation through clear menus and options. Two multiturn potentiometers are used for quickly adjusting the output signal’s amplitude and DC offset. For the display I used a cheap cellphone replacement display, the Nokia 6100, that’s well-documented by the open-source community on the internet. Unfortunately
this display exists in two variants (Epson and Philips) that are not fully compatible. Even more inconveniently, in most cases you cannot specify the type you want when you buy it. To work around this problem I wrote two versions of the software to support both types. Because looks are important too, when the electronics were completed I built the generator into a nicely milled standard aluminum enclosure from Hammond. The result is a very stylish instrument that has won a preeminent place on my electronics workbench.

Specifications

- Direct Digital Synthesis (DDS) with analog front-end
- Frequency range: 1 – 10 MHz
- Frequency resolution: 0.28 Hz
- Output: 0 – 15 Vpp
- THD+N (100 kΩ load, B > 500 kHz):
  - 1 V, 1 kHz: 0.12% (0.09% for B = 22 kHz)
  - 5 V, 1 kHz: 0.1% (0.09% for B = 22 kHz)
  - 1 V, 10 kHz: 0.1% (0.09% for B = 80 kHz)
  - 5 V, 10 kHz: 0.09% (0.08% for B = 80 kHz)
  - 1 V, 100 kHz: 0.1%
  - 5 V, 100 kHz: 0.08%
- S/N (referred to 1 V): 72 dB
- Maximum output (10 MΩ load):
  - Sine: 16 Vpp
  - Triangle: 16 Vpp
  - Square: 18 Vpp
- DC offset voltage range: -10 to +10 V
- Output impedance: 50 Ω
- Duty cycle (square wave): 1 – 99%
- Rise and fall time (80%, square wave): 100 ns
- Sweep mode
- Power consumption: 3 VA

An FFT plot up to 130 kHz of a 1-V, 1-kHz sinusoidal output signal (fundamental suppressed). The THD+N is mainly caused by the harmonics of the signal. Some AC line related (50 Hz) components are visible but are close to the noise floor.

Visit www.elektor-labs.com/150210 for more figures and plots.
The circuit

Figure 1 shows the schematic of the DDS function generator. The signal generating part is at the top, the lower part shows the microcontroller and the user interface.

The passive parts around DDS chip IC3 are as recommended by the manufacturer. Its output is filtered by RC network R16/C18 before being amplified by IC5.A. The reason for such a simple filter instead of a higher-order one was simplicity. Also, as we will see later on, a high-order filter was not really necessary. The output signal from IC5.A follows two paths. The upper path is used for sinusoidal signals, while the lower path is used for square waves.

Figure 1. The signal generator’s main circuit board contains everything except the power supply.
adds this DC voltage to the output signal. The output signal is fed to one of the microcontroller’s analog inputs (ADC0) after scaling and removal of the DC offset (IC2). The purpose is to monitor the level of the output signal as indicated on the display. Refer to [2] for measurements.

The circuitry around microcontroller IC1 is quite self-explanatory: eight pushbuttons, a graphic LCD, a number of decoupling capacitors and some headers, that’s all there is to it. The MCU runs from its internal oscillator so no external quartz crystal is required. Note the JTAG connector I used for programming and debugging the software. The serial port is also accessible for folks forced to use the MCU’s bootloader to program it. JP1 lets you make the MCU start in bootloader mode.

The power supply (Figure 3) is a classic design built around LM317s and an LM337. Together they produce a stable ±15 V needed for the analog output stage, and a clean +3.3 V for the rest of the circuit. Contrary to the main PCB that’s almost completely populated with surface-mount devices (SMDs), the power supply was built from through-hole components from the Elektor.Labs Preferred Parts (ELPP) list.

A single transformer is used with two secondary windings of 15 V / 5 VA each. 10 VA or so is enough to power the DDS

---

**Figure 2.** One period of a 1.00-MHz square wave (50% duty cycle) at almost maximum output with potentiometer P2 a 10-turn, 1-kΩ type.

**Figure 3.** The power supply for the signal generator consists of a bunch of LM3x7 voltage regulators. Some hocus-pocus was applied to reduce power consumption in the 3.3 V part.
generator. To minimize power loss in the 3.3 V regulator a separate rectifier (D5/D6) with a relatively small filter capacitor is used (C17/C18). The high ripple voltage that remains reduces power loss in regulator IC3 somewhat. Schottky diodes are used in positions D1 and D2. At maximum load the voltage drop across ‘normal’ diodes together with the ripple on C5 cause the input voltage of IC1 to come perilously close to its minimum value. At 0.5 A the forward voltage of the Schottky diodes is less than 0.45 V whereas a classic 1N4007 would drop almost twice as much.

Printed circuit boards were designed for both the main circuit and the power supply. The main board was given special attention in order to keep the output signal away from any interference produced by high-speed digital control signals. Most parts by far are SMD but assembling the board should not pose problems as long as you wear strong glasses.

### The software

The source code was written in C using the well-known µVision integrated development environment from Keil. A real-time operating system (RTOS) was not used to keep things simple. The code consists of only a few files. Some of them are header files containing definitions and function prototypes, the others contain the functions itself. Two of them are to support the two different graphic LCDs, the file ‘main.c’ con-

### Component List Generator Board

#### Resistors

- All 1%, 0.125 W, SMD 0805
- R1 = 220Ω, 1%, 0.75W, SMD 2010
- R2, R22, R23, R37 = 1kΩ
- R3 = 1kΩ, 1%, 0.1W, SMD 0603
- R5, R6, R7, R8, R9, R10, R11, R12, R30, R31, R36 = 10kΩ
- R13 = 6.8kΩ
- R14, R15 = 2000Ω
- R16, R19, R25 = 910Ω
- R17, R20 = 1.2kΩ
- R18, R26 = 2.0kΩ
- R21, R24 = 499Ω
- R27 = 220Ω
- R28 = 100Ω
- R29 = 10kΩ, 1%, 0.1W, SMD 0603
- R32, R33 = 100Ω, 1%, 0.75W, SMD 2010
- P1 = 10kΩ, 2W, 10-turn potentiometer
- P2 = 1kΩ, 2W, 10-turn potentiometer

#### Capacitors

- Default: SMD 0603
- C1, C3, C4, C12, C13, C14, C16, C17, C19, C20, C26, C27, C28, C29, C31 = 100nF, 50V, X7R

#### Semiconductors

- IC1 = ADUC7024BSTZ62, LQFP-64, programmed*
- IC2 = AD8032ARZ, SOIC-8
- IC3 = AD9834BRUZ, TSSOP-20
- IC4 = FXO-HC736R-75, 7 x 5 mm
- IC5, IC6 = AD8042ARZ, SOIC-8
- IC7 = ADG779BKSG-REEL7, 6-Lead SC-70
- IC8 = AD811ANZ, DIP-8
- IC9 = OP1177ARZ, SOIC-8
- IC10, IC11 = ADR1581ARTZ-REEL7, SOT-23-3

#### Other

- K1 = Socket, 0.5 mm, 1.5 mm stack, 10-way,
- DF23C-10DS-0.5V(51), Hirose (HRS)
- K2, JP1, JP2 = 1x2 pin header, vertical, 0.1” pitch
- K3 = 2x3 pin header, vertical, 0.1” pitch
- K4 = 1x4 pin header, vertical, 0.1” pitch
- K5 = BNC 50 Ω, Straight Bulkhead Jack, Panel mount
- S1-S8 = 6 mm tactile switch, actuator length 4.9mm, 24V/0.05A, SPST-NO
- 8-way DIP socket for IC8,
- Graphic B/W replacement LCD for Nokia 6100
- * Not available ready programmed from www.elektor.com

### Miscellaneous

- Aluminum enclosure, Hammond type 1455T1601, 165 x 160 x 51.5mm
- Optional: EMI/EMC filter, inlet, IEC, 250VAC / 4A
- 2 knobs, black, 16mm, 0.25 in. shaft diam.
- PCB 150210-1 v1.11 (www.elektor.com)

* Not available ready programmed from www.elektor.com

---

**Figure 4a. Component overlays of the generator PCB (double-sided SMD populated).**
tains all the signal generator code. The file ‘init.s’ is written in assembly language and contains the functions to initialize the microcontroller. This file was written from scratch and is not part of the development environment.

The LCD driver is partly based on open source code found on the Internet and partly written by myself. Since the display is not connected to a hardware SPI port, the communication protocol is emulated and bit-banged on GPIO pins. The functions WriteLcdCommand and WriteLcdData take care of this. Most of the display driver coding effort went in creating two fonts, a large and a small one. The keyboard is handled by polling in the main endless loop.

To keep code size small and IDE-independent, an attempt was made to avoid using floating-point arithmetic and math libraries.

All source files have been compiled into so-called ARM thumb code (16-bit code). Not so much reducing the size of the hex file, but because the flash memory of the microcontroller is 16-bit wide, making 16-bit code faster to execute. The total size of the executable is about 8 KB plus another 20 KB to hold the splash screen that appears at power up. No tricks were needed to fit all this in the MCU because its flash memory is large enough for the purpose. The software archive for the project can be downloaded FOC from [1].

Building it

Two PCBs have been designed to hold all the parts; they are printed in Figure 4 along with the component lists. As mentioned, assembling the main PCB is not too difficult as long as you have good eyes and/or a magnifying glass. Note that the pushbuttons and the LCD must be mounted on the bottom side of the PCB. The LCD requires some special attention as its flexible flat cable must be detached from its plastic support (Figure 5) so that it can be folded over the PCB and reach K1 (Figure 6).

The two potentiometers should stick through the PCB with the shaft protruding from the bottom side. At Elektor.Labs

Figure 5. On the left, the display as used in some mobile phones of a well-known Scandinavian brand (ain’t Ikea), on the right the same display adapted for our generator.

Figure 4b. Component overlay of the power supply PCB (single-sided TH populated).

---

**Component List**

**Power Supply**

<table>
<thead>
<tr>
<th>Resistors (1%, 0.6 W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1, R3 = 180Ω</td>
</tr>
<tr>
<td>R2, R4 = 2.0kΩ</td>
</tr>
<tr>
<td>R5 = 240Ω</td>
</tr>
<tr>
<td>R6 = 390Ω</td>
</tr>
<tr>
<td>R7 = 10kΩ, 5%, 0.25W</td>
</tr>
<tr>
<td><strong>Capacitors</strong></td>
</tr>
<tr>
<td>C1,C2,C3,C4,C15,C16 = 10nF 50V, Y5V, 0.2” pitch</td>
</tr>
<tr>
<td>C5 = 1000µF 50V, 5 or 7.5 mm pitch, 16mm diam.</td>
</tr>
<tr>
<td>C6 = 470µF 50V, 5 or 7.5mm pitch, 13mm diam.</td>
</tr>
<tr>
<td>C7,C8,C13,C14,C18,C21 = 100nF 50V, X7R, 0.2” pitch</td>
</tr>
<tr>
<td>C9,C10,C11,C12,C19,C20 = 10µF 50V, 2mm pitch, 6.3mm diam. max.</td>
</tr>
<tr>
<td>C17 = 47µF 50V, 2.5mm or 3.5mm pitch, 8mm diam. max.</td>
</tr>
<tr>
<td><strong>Semiconductors</strong></td>
</tr>
<tr>
<td>D1,D2 = STPS2L60, DO-41 case</td>
</tr>
<tr>
<td>D3,D4,D5,D6 = 1N4007, DO-41 case</td>
</tr>
<tr>
<td>IC1,IC3 = LM317, TO-220 case</td>
</tr>
<tr>
<td>IC2 = LM337, TO-220 case</td>
</tr>
<tr>
<td>LED1 = LED, green, 3mm</td>
</tr>
<tr>
<td><strong>Miscellaneous</strong></td>
</tr>
<tr>
<td>K1 = 2-way PCB screw terminal block, 0.3” pitch, 500V</td>
</tr>
<tr>
<td>K2 = 4-way (2x2) PCB screw terminal block, 0.2” pitch, 250V</td>
</tr>
<tr>
<td>TR1 = 2x115V prim./2x15V sec., 10VA, e.g. Block FL 10/15</td>
</tr>
<tr>
<td>F1 = fuse, 100 mA (230 VAC line) or 200mA (115 VAC line); slow blow, 250V, 20 x 5mm</td>
</tr>
<tr>
<td>F2, F3 = fuse, 315mA, slow blow, 250V, 20x5mm</td>
</tr>
<tr>
<td>Fuse holder for F1, F2, F3, 20 x 5mm, 500V, 10A</td>
</tr>
<tr>
<td>Covers for F1, F2, F3 fuse holders, 20 x 5mm</td>
</tr>
<tr>
<td>JP1 = Jumper wire</td>
</tr>
<tr>
<td>PCB 150210-2 v1.1 (<a href="http://www.elektor.com">www.elektor.com</a>)</td>
</tr>
</tbody>
</table>
we used homebrew rubber ‘disks’ (cut from an old bicycle inner tire) to prevent the potentiometers from slipping while adjusting them. 10 mm for the disk’s hole diameter is fine, and the outside diameter is preferably a little under 22 mm. In the case of P2 the rubber disk also prevents damage to a copper track on the top side next to the hole for P2.

The power supply PCB should not pose any problems. We went for a transformer with two primary windings to support both 115 VAC and 230 VAC grid voltages. A jumper wire has to in place at JP1 (the middle one of three dotted lines) if you are on 230 VAC. Two jumper wires are necessary for 115 VAC (the two outer dotted lines). Do not install all three jumper wires! Also don’t forget to fit the correct primary fuse: 100 mA(T) for 230 VAC and 200 mA(T) for 115 VAC, where (T) is time delay.

The three regulators require a heatsink which we made out of a single 2-mm thick aluminum strip (Figure 7). Don’t forget the electrical insulation (mica washer and plastic bush) for the three regulators. The M3 screw should be about 6 mm long. A metal washer between the head of the screw and the plastic bush is advised. Often the plastic bush is a bit too long so cut it to the proper length (with a hobby knife) before mounting the regulators on the heatsink.

If you decide to use the same enclosure as we did (Figure 8) then you can download a mechanical drawing from the Elektor website [1] showing milling details of the front and back panels.

BNC socket K5 is isolated and should be mounted on the front panel. After fixing the main PCB to the front panel (we glued the screws to the back of the panel), connect the BNC with short wires to the PCB.

Programming

Before the signal generator will work it must be programmed with the right firmware. Because of the two possible LCD configurations, two pieces of firmware are available from our website [1]. Since you can’t tell the exact type of the display by looking at it, it is a matter of trial and error to find the right software. Nota dicky bird from the function generator after programming the MCU? Try the other firmware. No wave either? Uh-oh…

There are two ways of programming the MCU: JTAG or bootloader.

The first option requires a JTAG adapter
and also gives you a debugging interface. Well known JTAG adapters are the J-LINK from Segger and Keil’s ULINK. The standard JTAG interface has 20 pins, but it can work with only six pins too (K3). The second option is over the serial port interface (K2). This is a programming-only option, debugging is not possible over this interface. A suitable cable can be purchased from Analog Devices but it is very easy to make your own with a serial TTL-to-USB cable. A software serial programming tool (ARMWSD.exe) is available for free from the Analog Devices website. First select the appropriate USB serial port and load the hex file. The program will then ask you to “Press Download and pulse Reset on hardware” (Figure 9). That’s why the two jumpers JP1 and JP2 landed on the PCB labeled ‘Download’ and ‘Reset’ respectively. If a jumper is fitted on Download at power-on, the microcontroller will stay in bootloader mode and the display will remain dark, so don’t forget to remove it after programming.

User Manual

<table>
<thead>
<tr>
<th>PB</th>
<th>Function</th>
<th>PB</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Set</td>
<td>S5</td>
<td>–Down</td>
</tr>
<tr>
<td>S2</td>
<td>Right</td>
<td>S6</td>
<td>Mode</td>
</tr>
<tr>
<td>S3</td>
<td>Left</td>
<td>S7</td>
<td>Sweep</td>
</tr>
<tr>
<td>S4</td>
<td>+Up</td>
<td>S8</td>
<td>Calibration</td>
</tr>
</tbody>
</table>

- **Waveform** – Press Mode to toggle between sine, square and triangle.
- **Duty Cycle** – The duty cycle can only be set in square wave mode. Press Mode to activate the square wave output. The duty cycle is indicated on the bottom line of the display. Adjust the duty cycle by pressing +Up and –Down (the digits must not blink).
- **Frequency** – Press Set. A digit starts blinking. Use +Up and –Down to change the value of the blinking digit; use Left and Right to navigate through the digits. When done press Set.
- **Amplitude** – Adjust P2. Note that adjusting the amplitude affects the offset voltage. See [2] for detailed measurements.
- **DC offset** – Adjust P1.
- **Frequency sweep** – Press Sweep to open the sweep menu. The least significant digit of the start frequency is blinking. Use +Up and –Down to change the value of the blinking digit; use Left and Right to navigate through the digits. When done press Set to advance to the next parameter. Set the stop frequency, the sweep time (called “msec”) and the sweep mode (logarithmic or linear). Press Set to start the sweep. This is indicated by “sweep run” in the first line of the sweep menu. Press Set again to stop the sweep (“sweep stop” is displayed in the first line) and new values can be set. Press Sweep to return to the main menu.
- **Contrast** – Press Calibration to open the calibration menu where the LCD contrast can be set. Use Set to navigate to the contrast option, then use +Up and –Down to change the contrast level. Press Calibration to return to the main menu.
- **Calibrate voltage levels** – Connect an oscilloscope to the generator’s output and set the output level to 5 Vpp. Press Calibration to open the calibration menu. Select Measurements to start the calibration procedure. (If entered by mistake, the only way to get out is by switching off the power.) Adjust P1 to set the minimum value of the output signal to 0.00 V, press Set when done. Adjust P1 to change the maximum value of the output signal to 12.00 V. Press Set when done. A message appears to indicate that calibration has been completed. Press Calibration to return to the main menu.
- **Calibrate frequency** – Connect a high-precision frequency counter to the function generator’s output. Press Calibration to open the calibration menu. Select Frequency to start the calibration procedure. (If entered by mistake, the only way to get out is by switching off the power.) Adjust the output frequency to 100,000 Hz by pressing +Up and –Down. Press Set when done. A message appears to indicate that calibration has been completed. Press Calibration to return to the main menu.

Web Links

There you are
Now that you have built and tested this nice DDS function generator you no longer have any excuse for refusing amplifier repair jobs. Graphing a filter’s transfer curve has become child’s play. Riding the waves will come natural to you. Welcome to the wonderful world of well-equipped electronics engineering (WWW-EEE)!