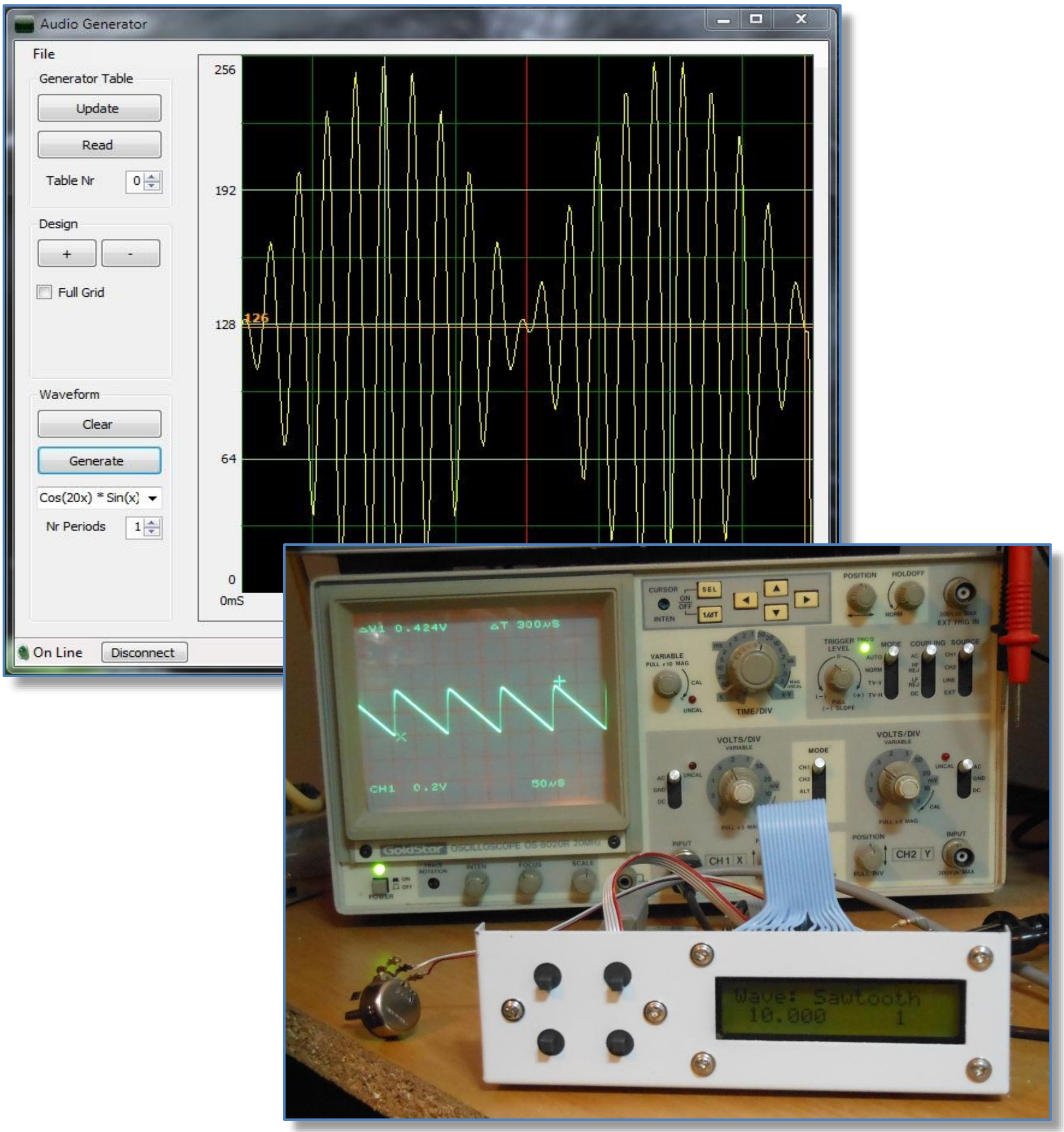
Fox Delta
Amateur Radio
Projects & Kits

ARBITRARY AUDIO GENERATOR

by i2mkm and i2tzk



January, 2013

Index

1 Project genesis

Digital Direct Synthesizer (DDS) is, in my opinion, one of the most successfully example of digital and analog technology integration.

I played with DDS since Analog Devices presented the first device of his AD98xx series, a number of my projects are based on this component but I never had time and resources to develop my own DDS from scratch.

The "Arbitrary Audio Generator" project, finally, gave to me this opportunity.

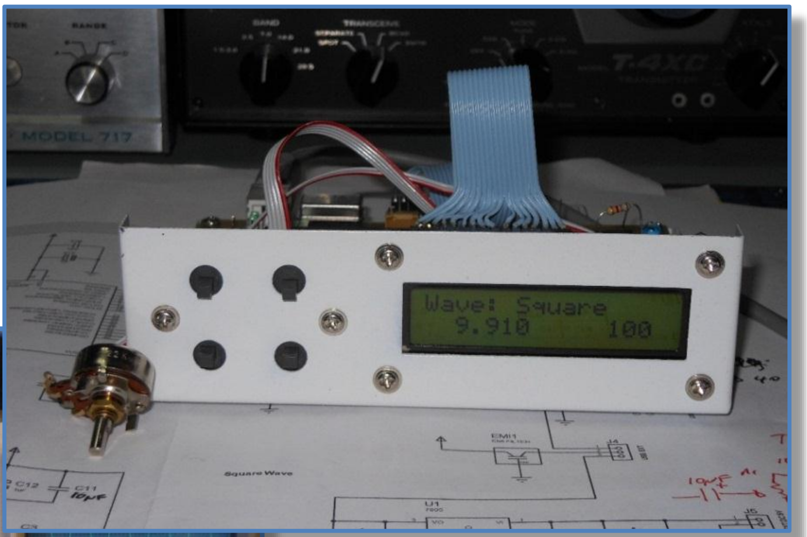
With the involvement of Mirko, ik2mkm who did a great job designing the analog section of the circuit and implementing the first prototype, the Arbitrary Audio Generator is now a good working, cheap instrument dedicated to all Ham Radio fellows.

Thanks also to Dinesh, vu2fd for proposing a very easy and economic kit to assemble this generator.

Tony, I2TZK

AUDIO GENERATOR main features :

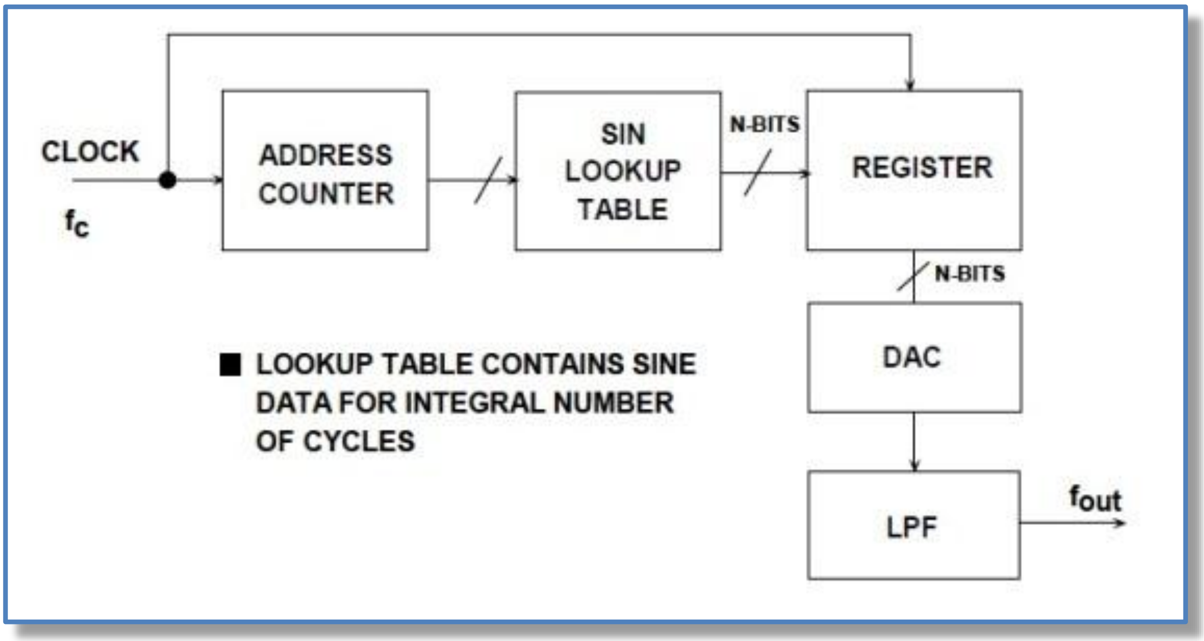
- Smart and very cheap hardware implementation
- Free firmware and software for the Ham Radio Community
- Generate standard sine, sawtooth and square wave from 10Hz to 50KHz
- Generate two tones for cat-eye SSB test
- Generate AFSK tones (2125/2295 Hz)
- Generate white and pink noise
- Generate arbitrary waveform
- Special PC program allows to design custom waveforms and upload the generator memory



2 Audio Generator project's notes

The "Audio Generator" project implements a Digital Direct Synthesizer (DDS), a type of frequency synthesizer used for creating arbitrary waveforms from a single, fixed-frequency reference clock.

In its simplest form this particular circuit configuration can be implemented from a precision clock reference, an address counter, a programmable memory, and a D/A converter.



Please, for more detailed notes, refer to "A Technical Tutorial on Digital Signal Synthesis", Analog Devices and to the very reach literature on this argument available from the net.

We choose a mid-range Microchip 18F4550 pic to implement the main blocks of the DDS process, f/w takes care to run a 32 bits architecture counter and to host the lookup table in his memory. A very tight loop reading and outputting the table values allows to generate very good signals from 10Hz up to 50KHz.

The address counter stores the current phase value of the generated waveform. The rate at which the register (phase accumulator) is updated and his increment determine the frequency of the output waveform. By setting the increment value to different values, we are able to produce different frequencies.

The current phase accumulator value is used to perform a lookup operation in the table to determine the next output value. The lookup table contains one cycle of the waveform to be generated, 256 sample points which represent the waveform.

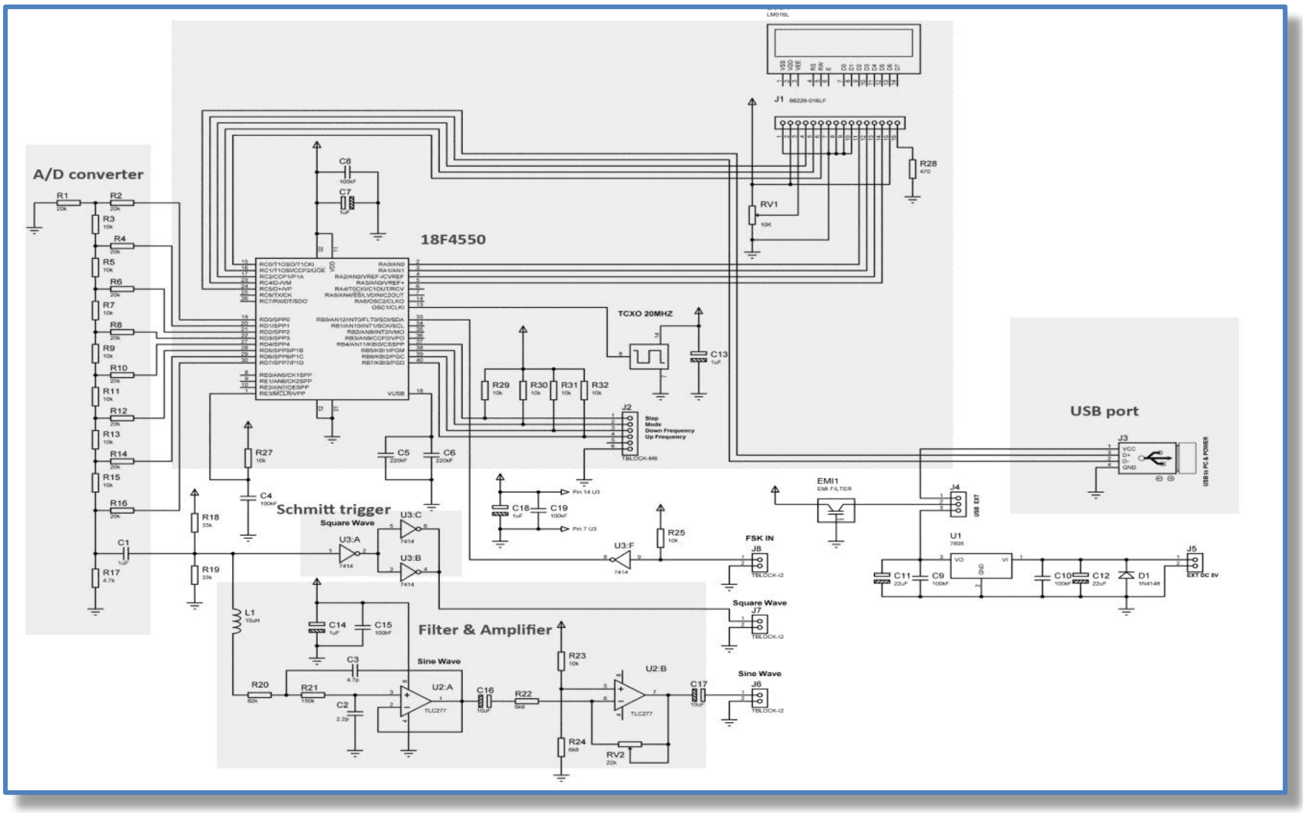
Standard lookup tables are present for generate sine, sawtooth and square wave while a **customizable table** can be designed and uploaded to the pic memory by a pc program (Windows) specially written for this purpose.

Because to produce different frequencies from 10Hz to 50KHz at 1Hz step we are incrementing by more than one the phase accumulator (or in other word we are up-sampling or down-sampling the waveform stored in the lookup table) this results in missing samples. Hence some **embedded jitter** is present.
Only frequencies equal to the clock frequency divided by the waveform length and its sub multiples are not sampled and therefore they do not suffer from this problem.
Generally speaking, skipping some samples doesn't represent a problem when generating a sine or similar waveform, the following filter stage absorbs most of this aliasing.

Waveform jitter is an issue generating an arbitrary function specially if the designed output is composed by a pulses burst. For this reason we decided to implement two algorithms processing the custom lookup table, obtaining a satisfactory work around to the jitter problem.

First algorithm **CUSTOM_1** manages the sampling rate the same way as done for the standard tables: fixed clock and variable phase accumulator. The second one **CUSTOM_2** is always full reading the table and different frequencies are obtained varying the sampling rate, this way the waveform is always drawn by 256 samples, **no aliasing or jitter**.

Of course, reading and outputting 256 samples whit a midrange pic device, implies some limitation to the max frequency generated but this guaranties the accuracy of the produced signal. Being the sampling rate 1uS, the max frequency is limited to a 256uS period.



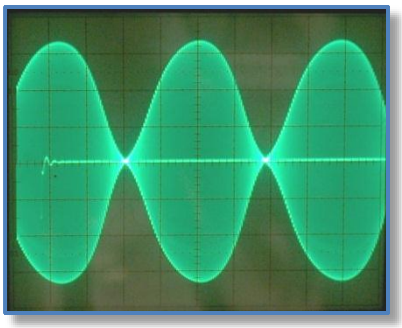
To keep low the hardware cost, a ladder, made up of resistor pairs, makes the A/D converter, we recommend to select 1% tolerance resistors for these components.

A low-pass filter and a buffer amplifier bring the signal to the output connector.

Because we want also to output a square wave or a custom burst at digital TTL level, we added a Schimtt trigger stage to properly adjust the signal level.

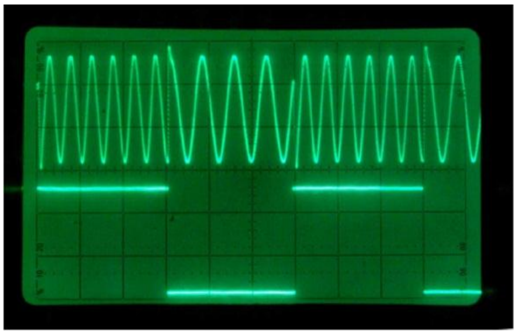
Having a very flexible hardware design we decided to include in the "Arbitrary Audio Generator" project some additional functions useful for testing the Ham radio equipment.

- A **two tones generator** has been implemented, two non-harmonically related sine waves of equal amplitude at 400Hz/2600Hz or 800Hz/1000Hz are generated.

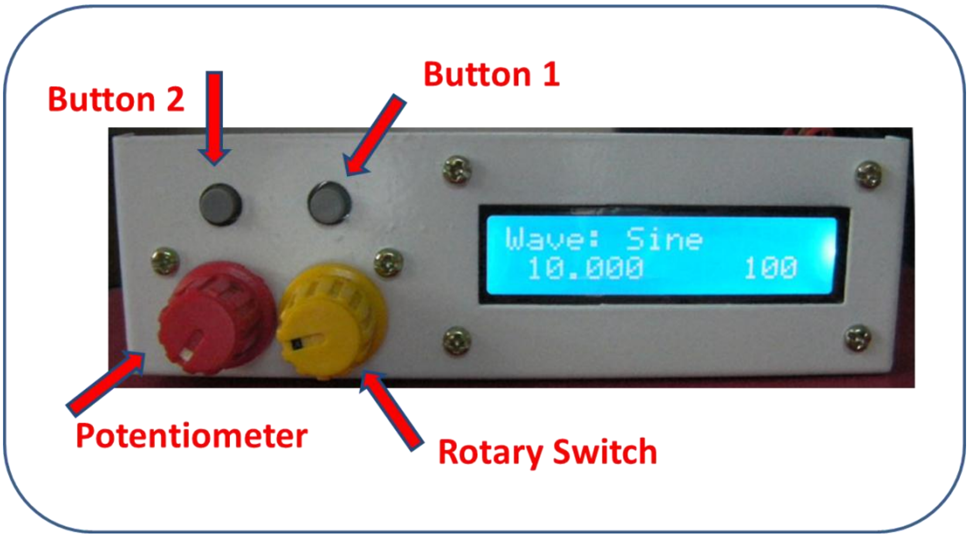


- A **noise generator** producing WHITE and PINK noise. The "White Noise" generator implements the Linear Congruential Generator (LCG) algorithm. The "Pink Noise is generated by putting the white noise through a "pinking" filter which removes more energy as the frequency increases. The algorithm implemented applies to the white noise generator a weighted sum of first order filters to approximate a 10dB/decade filter (Paul Kellet's "economy" method).

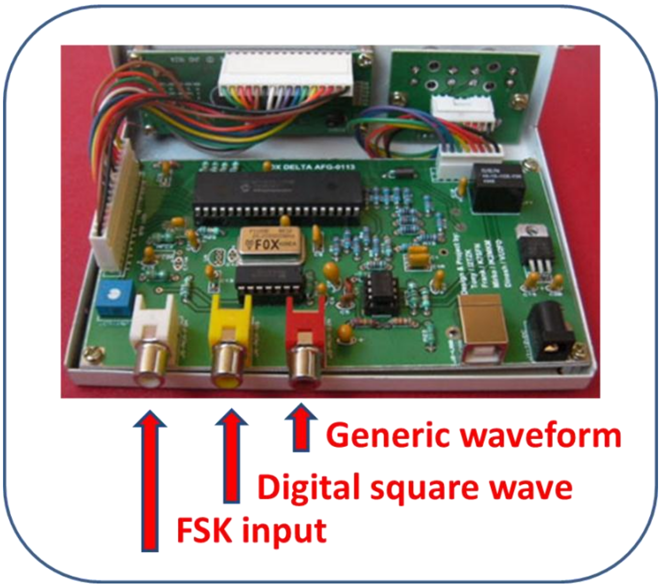
- **AFSK tones**. FSK signal is obtained switching between two continuously running oscillators at 2125Hz and 2295Hz, this method assure to maintain the phase coherency of each single tone allowing optimum decoding of weak signals.



3 Front panel

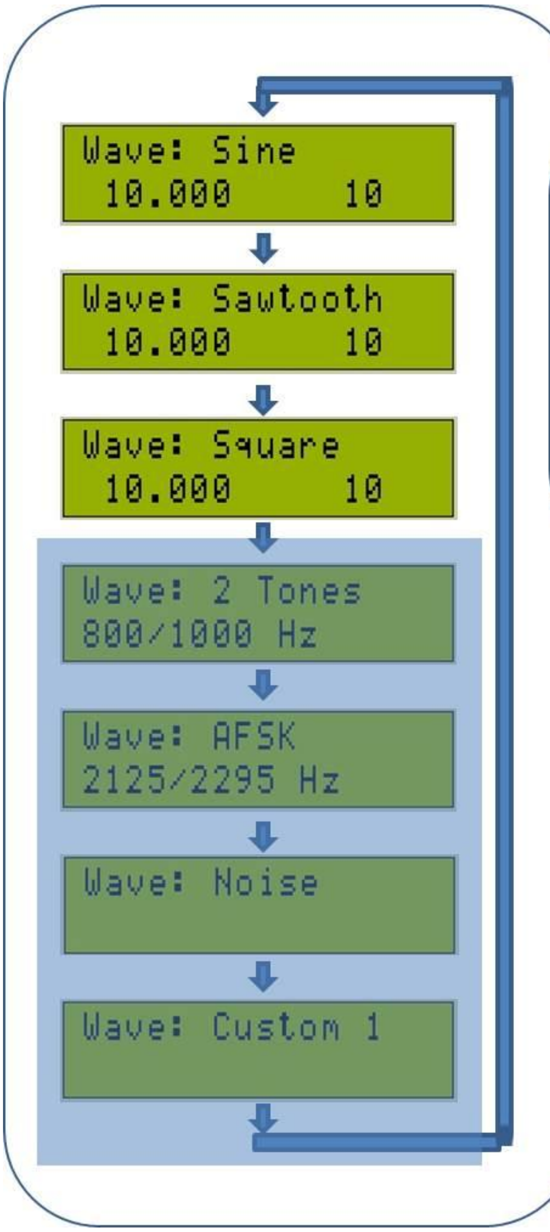


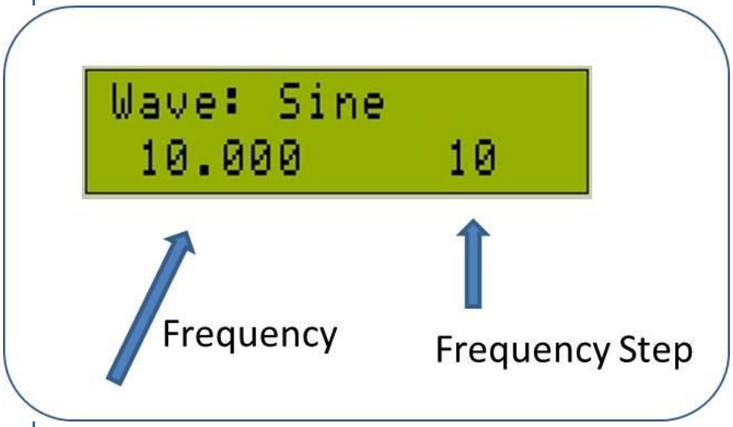
- **Button 1** : select output waveform type
  Push this button during power on to activate the bootloader
- **Button 2**: select step
  Push this button during power on to activate the PC communication
- **Rotary switch**: increment/decrement frequency or select the output type
- **Potentiometer**: adjust the output voltage level



4 Sine, sawtooth and square wave

Push **Button 1** to go through the different available waveforms selecting the one you want the Audio Generator outputs:



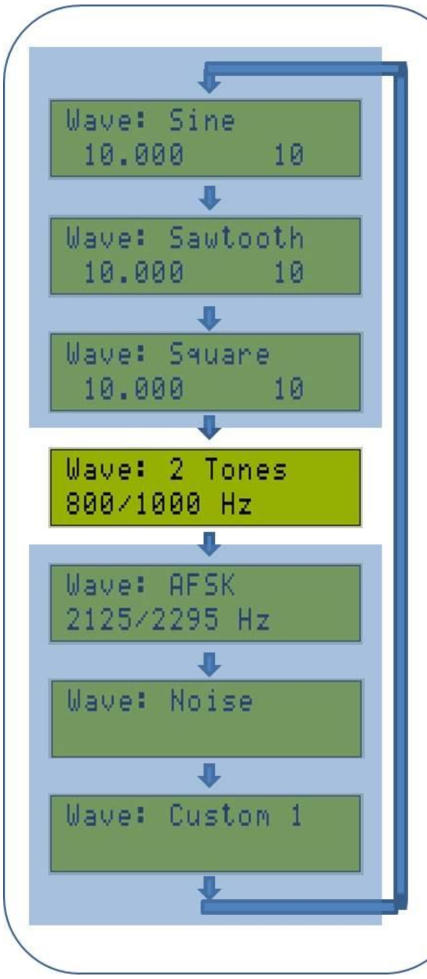


Turn the **rotary switch** to increase / decrease the frequency, push the **button 2** to select the frequency step.

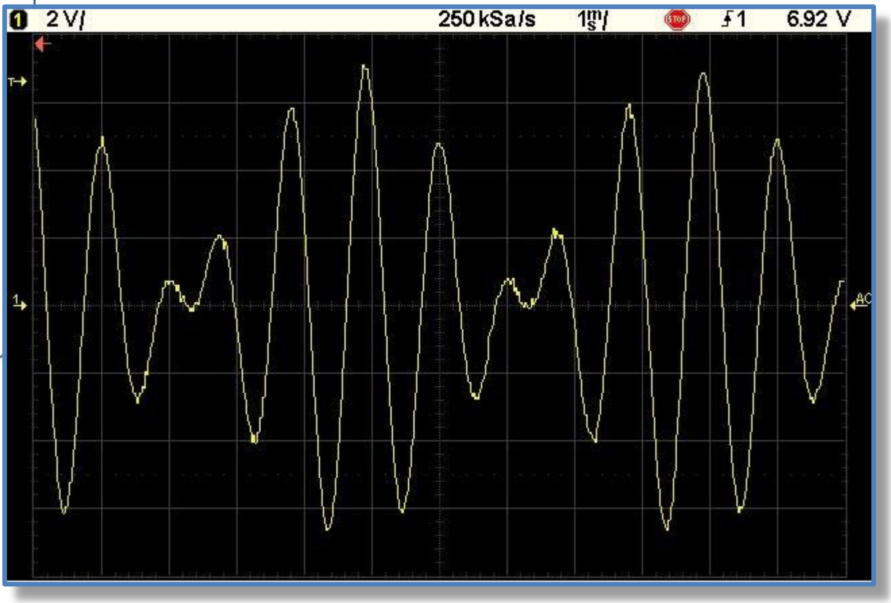
5 Two Tones transmitter test

As the major mode of operation on HF is SSB, the two tone testing is the simplest way to check the transmitter output for splatters and unwanted harmonic content.

With a two tone generator, a dummy load and an oscilloscope, a transmitter can be really put through its paces. Two tone testing consists of applying two clean **non-harmonically related sine waves** of equal amplitude to the audio input of an SSB transmitter. The sine waves are typically 400Hz/2600Hz (or 800Hz/1000Hz) about 300Hz from either end of the audio pass band. The result, in a properly adjusted transmitter, is an RF output that varies from zero to maximum at a rate determined by the difference in frequency between the two audio inputs. Consequently, overdrive (which causes splatter), non-linearity, instability and a host of other problems are easily visible on the oscilloscope.

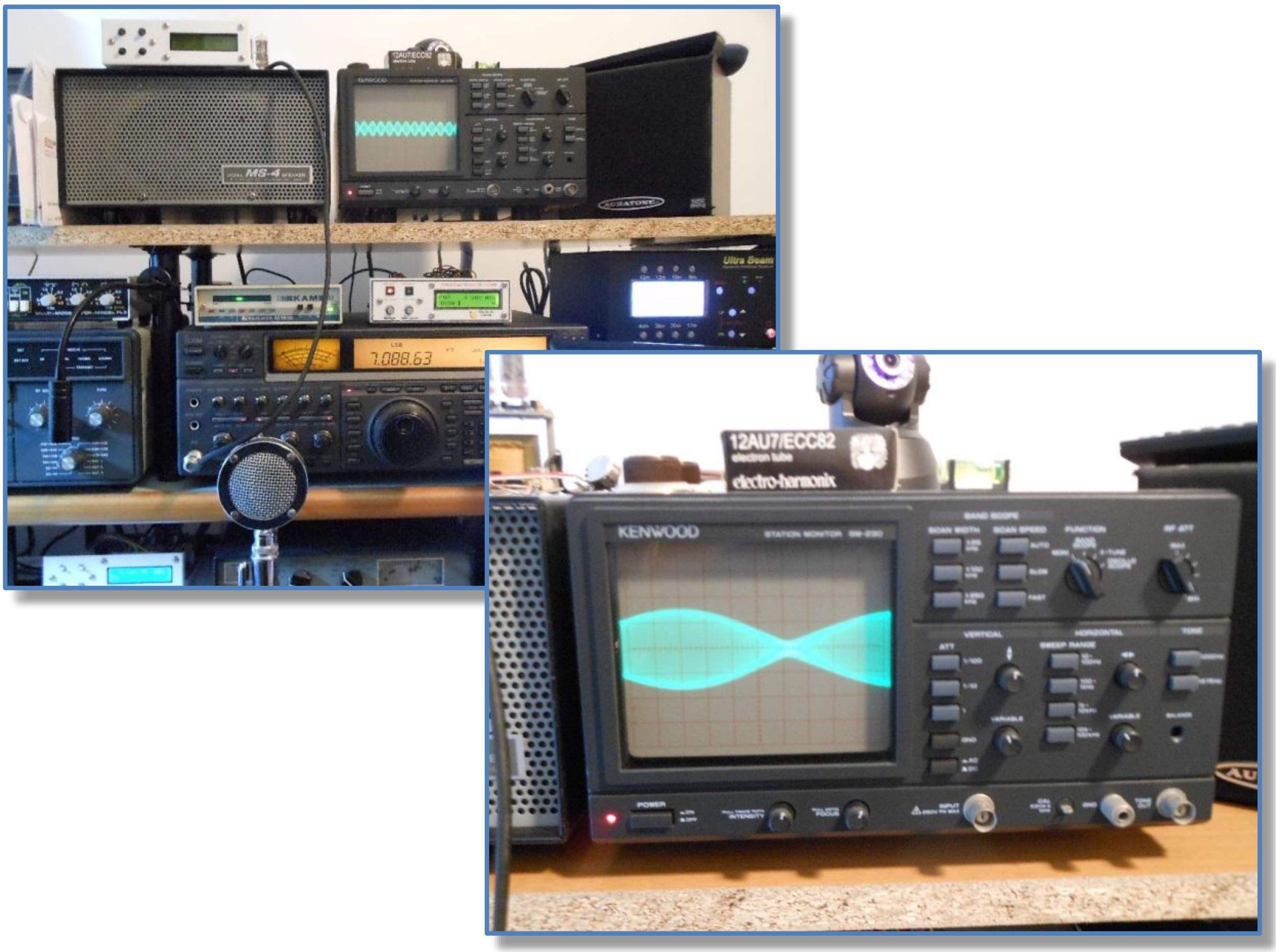


Default generated tones are the couple 400Hz/2600Hz, use the rotary switch to select 800Hz/1000Hz.



Audio Generator output
(400Hz - 2600Hz)

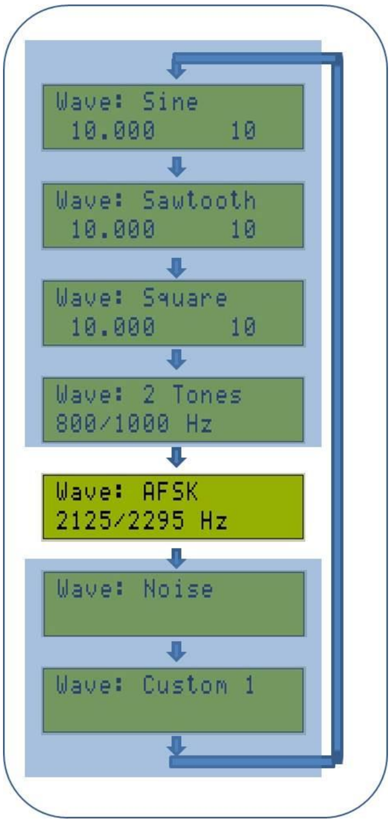
Mirko, IK2MKM, Two Tones transmitter test set-up.



6 Generating AFSK Tones

Voltage applied to the FSK input controls the tone frequency:

- FSK in Low (0v) -> 2.125 Hz (space)
- FSK in High (5v) -> 2.295 Hz (mark)

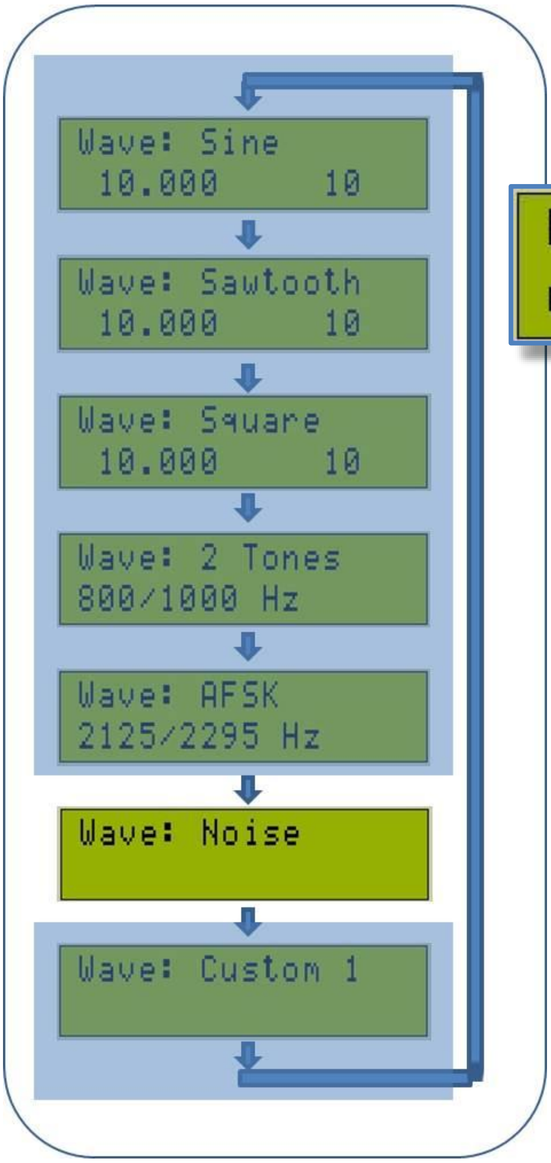


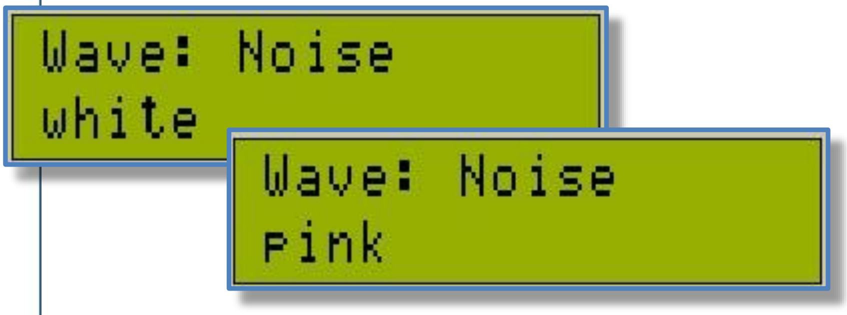
FSK signal is obtained switching between two continuously running oscillators at 2.125Hz and 2.295Hz, this method assures to maintain the phase coherency of each single tone allowing optimum decoding of weak signals.

7 White and Pink noise

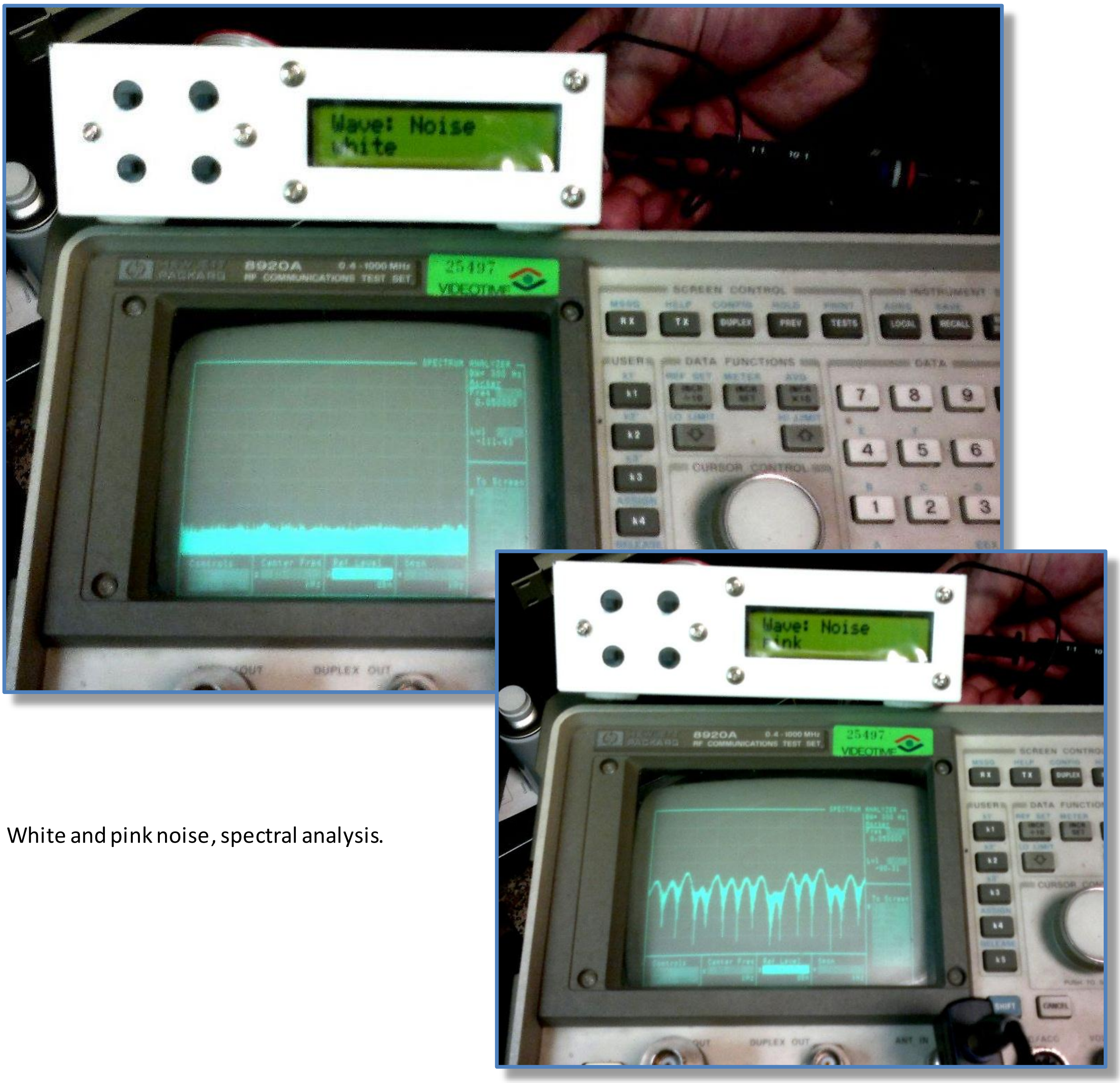
"**White Noise**" is a random signal with a flat power spectral density. In other words, the signal contains equal power within a fixed bandwidth at any center frequency (Wiki).
The "White Noise" generator implements the **Linear Congruential Generator** (LCG) that represents one of the oldest and best-known pseudorandom number generator algorithms.

The "**Pink Noise**" (or 1/f noise or flicker noise) is similar to the white noise except that it contains an equal amount of energy in each octave band, that is to say that the power spectral density is proportional to the reciprocal of the frequency.
Pink noise is generated by putting the white noise through a "pinking" filter which removes more energy as the frequency increases. The algorithm implemented applies to the white noise generator a weighted sum of first order filters to approximate a 10dB/decade filter (Paul Kellet's "economy" method).





Use the rotary switch selects White or Pink noise.

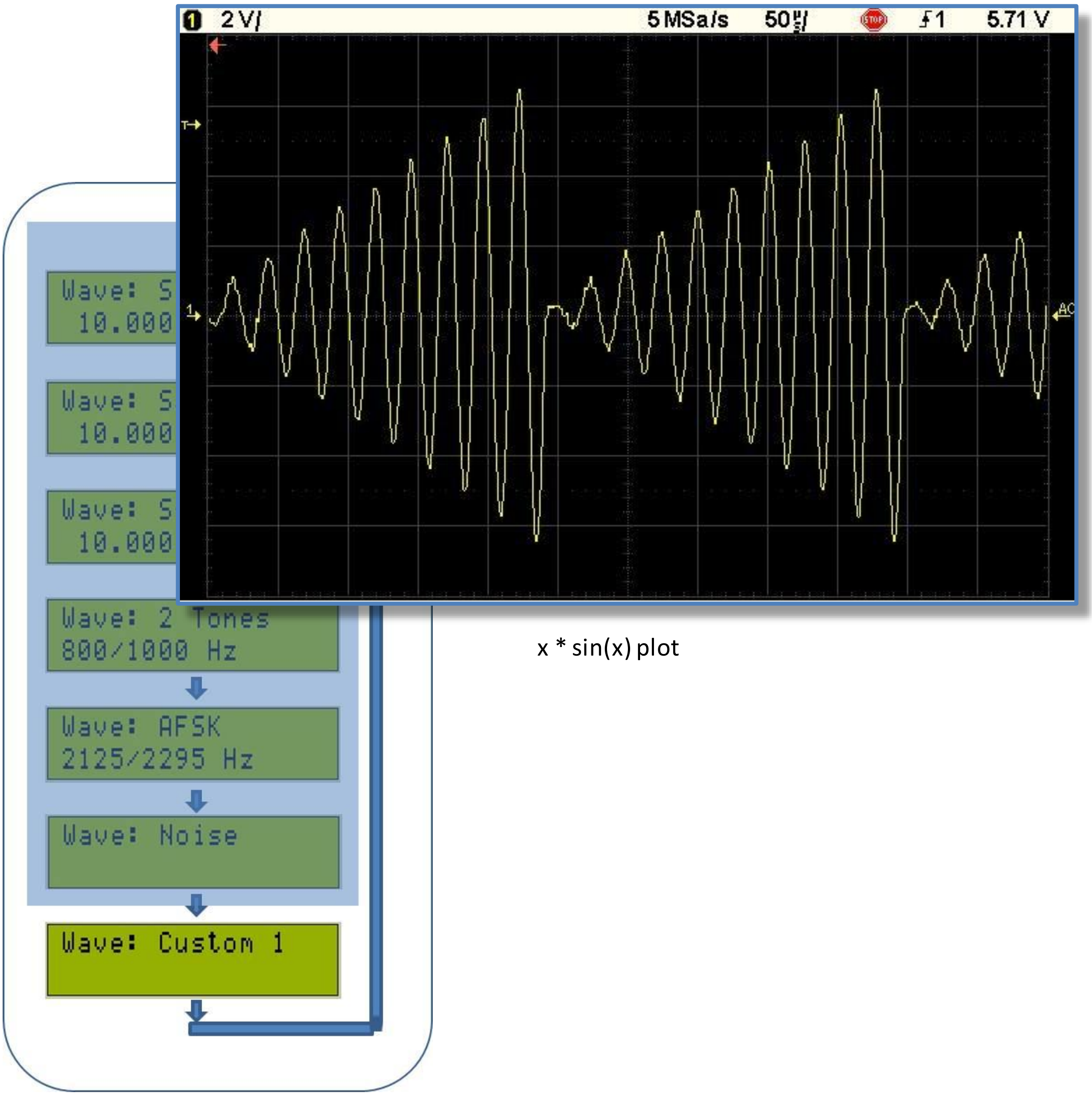

Wave: Noise
White
Wave: Noise
Pink
HEWLETT PACKARD 8920A 0.4 - 1000 MHz RF COMMUNICATIONS TEST SET
25497
VIDEOTIME
SPECTRUM ANALYZER

White and pink noise, spectral analysis.

8 Arbitrary Waveform Generator

“Custom” mode allows to virtually generate any kind of waveform.
This mode uses a data table describing the waveform to be generated.

A special Windows PC program written for this purpose, allows to easily draw the curve, generate the corresponding samples table and upload the table into the pic memory via the USB connection.



x * sin(x) plot

8.1 Connect the Audio Generator to the PC

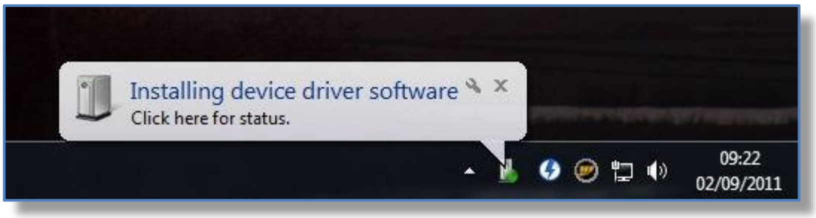
Connect the Arbitrary Audio Generator Unit to the PC using a standard USB cable (printer cable), **while plugin-in the cable press the Button 2.**

USB Enabled

The LCD after the welcome message will show the message “USB Enabled” to indicate that the USB interface has been enabled.

If this is the first time you attach this device, Windows will start to search and install the required drivers. To communicate with Windows, the Audio Generator uses the USB port embedded into the microcontroller PIC18F4550, so **drivers are the standard ones developed by Microchip and Windows Microsoft.**

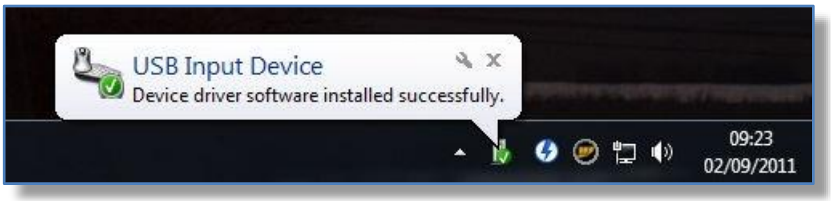
Windows is searching the driver stating that the board is ready to be linked by the PC program.



Installing device driver software
Click here for status.

09:22
02/09/2011

You should listen to the Windows announcement sound that indicates a new device has been detected and the corresponding driver is loading.



USB Input Device
Device driver software installed successfully.

09:23
02/09/2011

Driver successfully found and installed

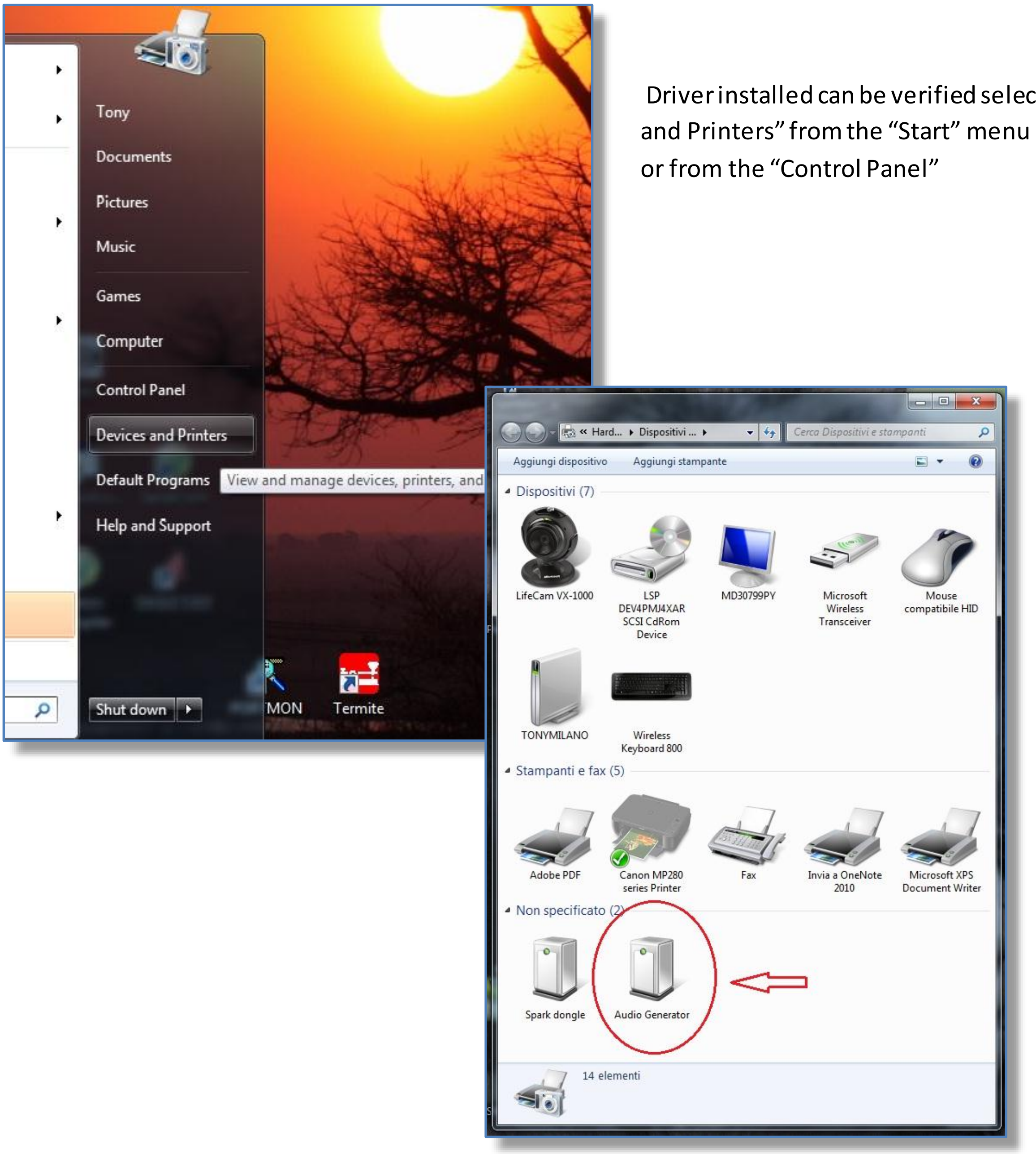
After a while, as soon as Windows has installed the drivers, the LCD will show the message “Waiting…” stating that the Audio Generator is ready to be linked by the Windows PC program.

Waiting...

The above operation can take several seconds depending on the time needed by Windows to find the right drivers.

Usually this happens only once, next time you connect the Audio Generator all needed parameters are already know by Windows.

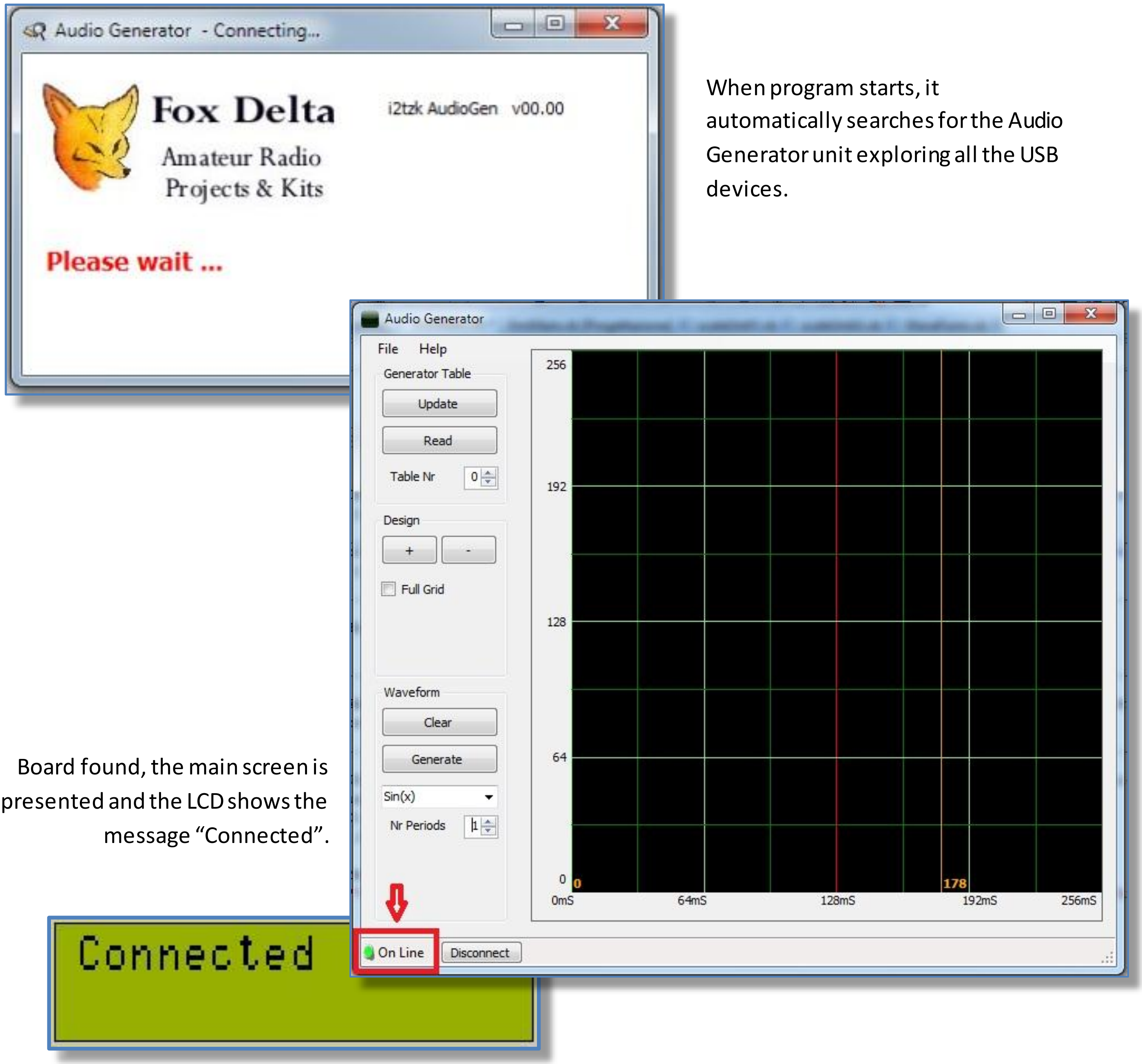
Driver installed can be verified selecting “Device and Printers” from the “Start” menu (Windows 7) or from the “Control Panel”



8.2 Launch the PC program

The software application, Windows based, doesn’t need any installation procedure, just create a new folder and unzip there the file “Audio Generator vx.xx.zip” you’ve downloaded from the server.

To launch the program, navigate to this folder and DoubleClick “Audio Generator.exe”.
For your convenience you can create a link to the desktop right dicking on “Audio Generator.exe” and selecting “Send to Desktop”.



Please wait ...

Connected

When program starts, it automatically searches for the Audio Generator unit exploring all the USB devices.

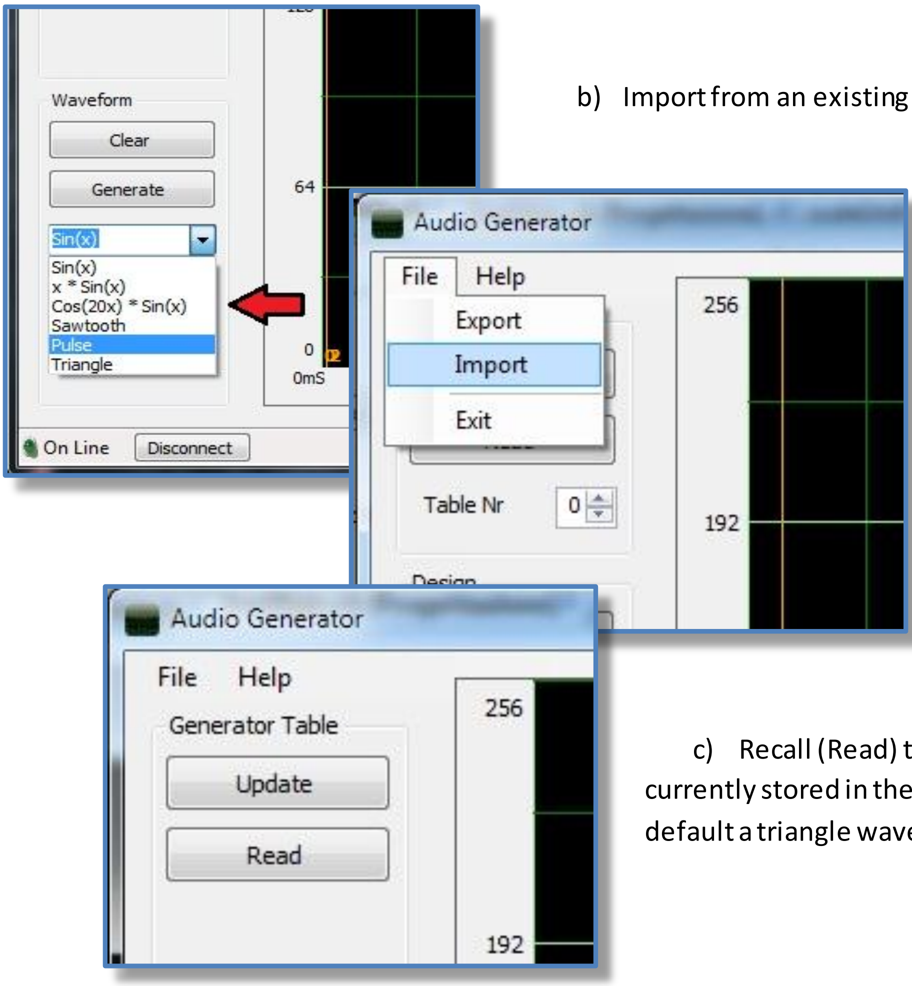
Board found, the main screen is presented and the LCD shows the message “Connected”.

8.3 Design your own waveform

The simple and fast way to design your own waveform is to recall an already existing figure and edit.

Program offers three options:

a) Select one from a predefined algorithms list.



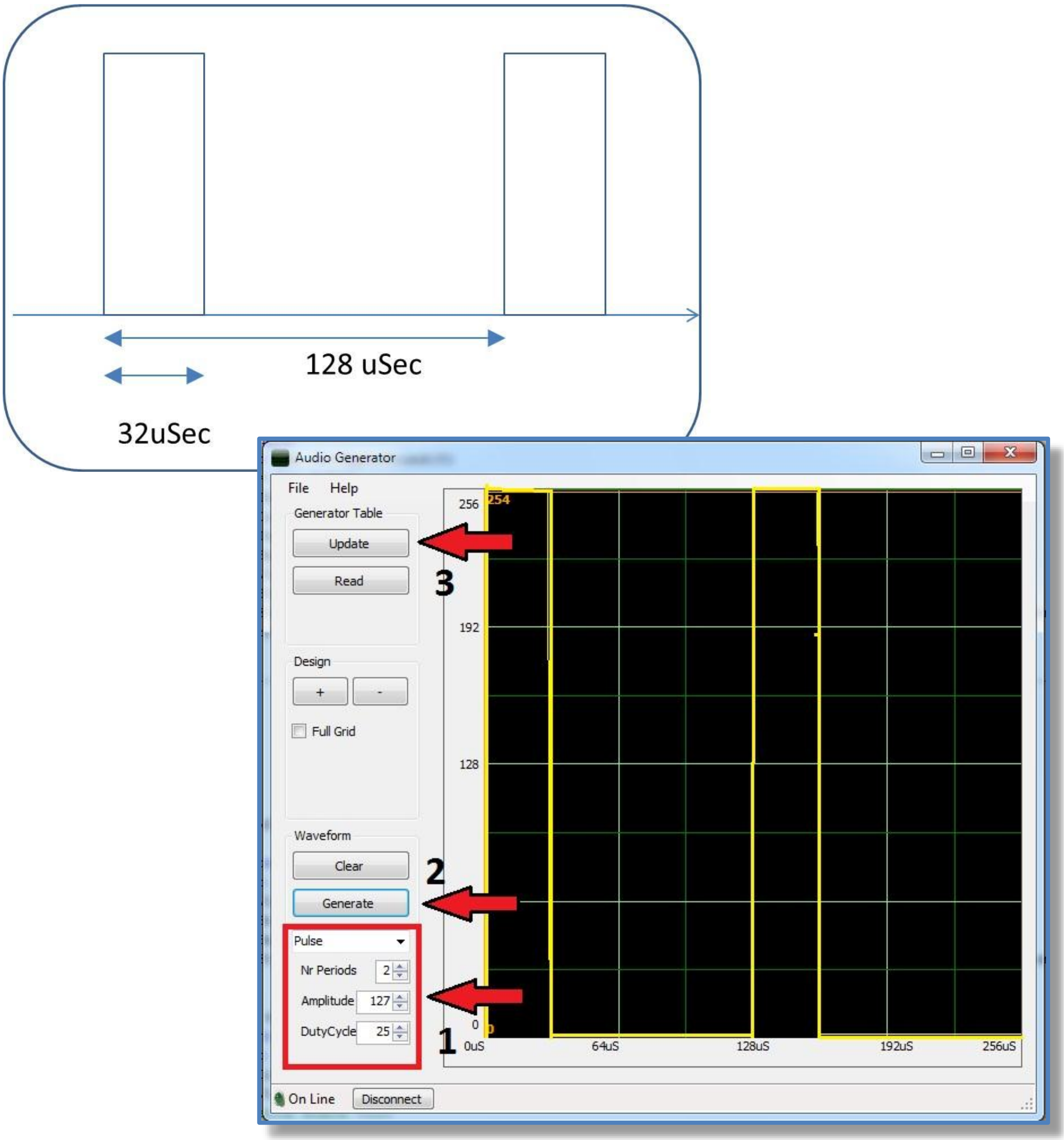
b) Import from an existing "csv" file.

c) Recall (Read) the waveform currently stored in the pic memory (by default a triangle wave).

Don't forget at the end of your design, to **press "Update" to write the waveform table into the pic memory** or to save it **Exporting** data as csv file.

8.3.1 Example 1

We want to generate a pulse width 32uSec every 128 uSec.



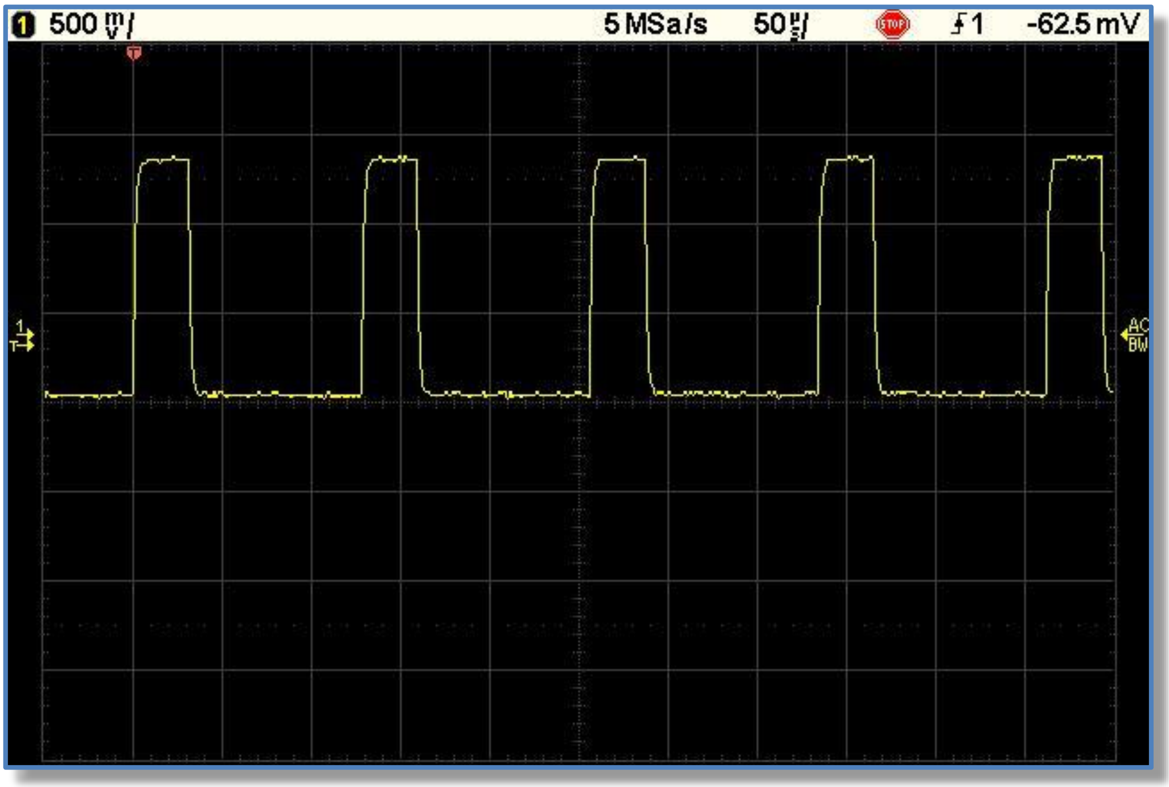
Pulse

Nr Periods 2

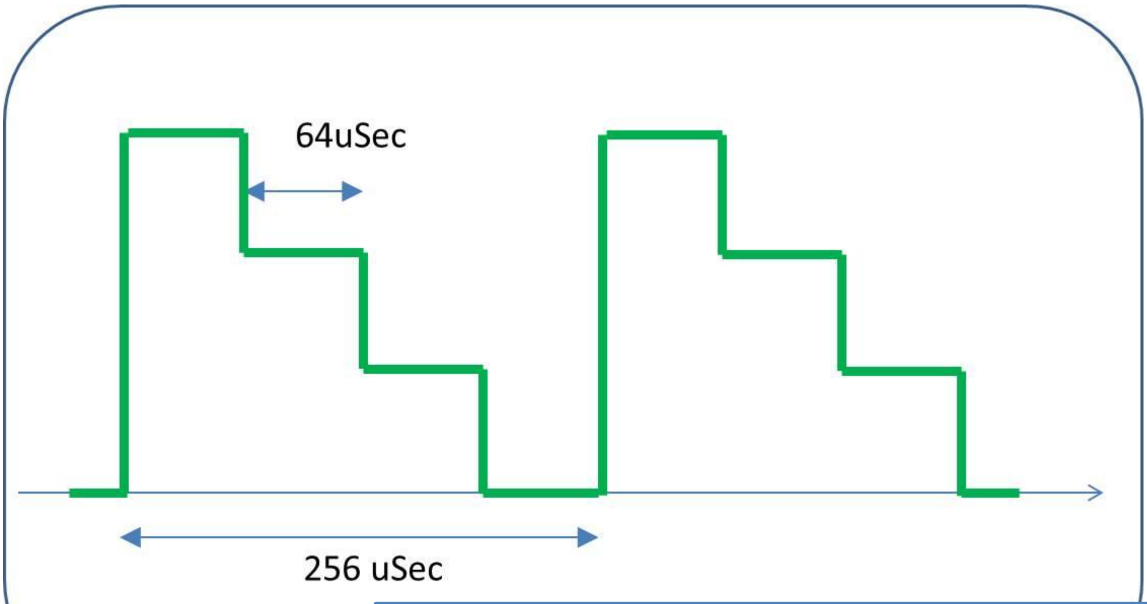
Amplitude 127

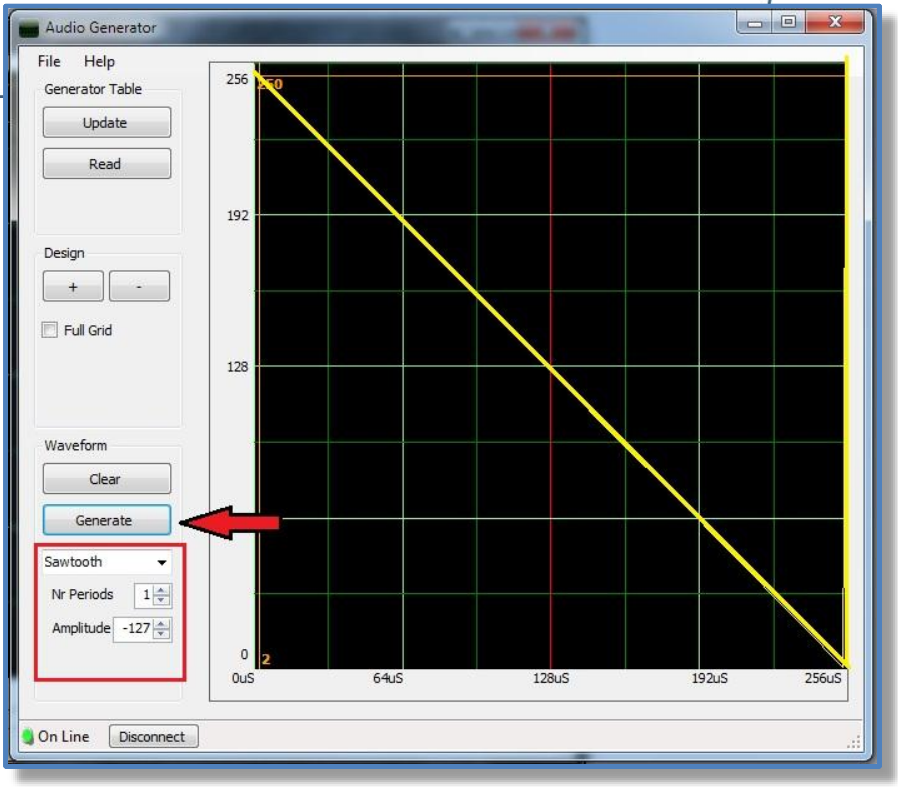
DutyCycle 25

Audio Generator output, DSO display.



8.3.2 Example 2

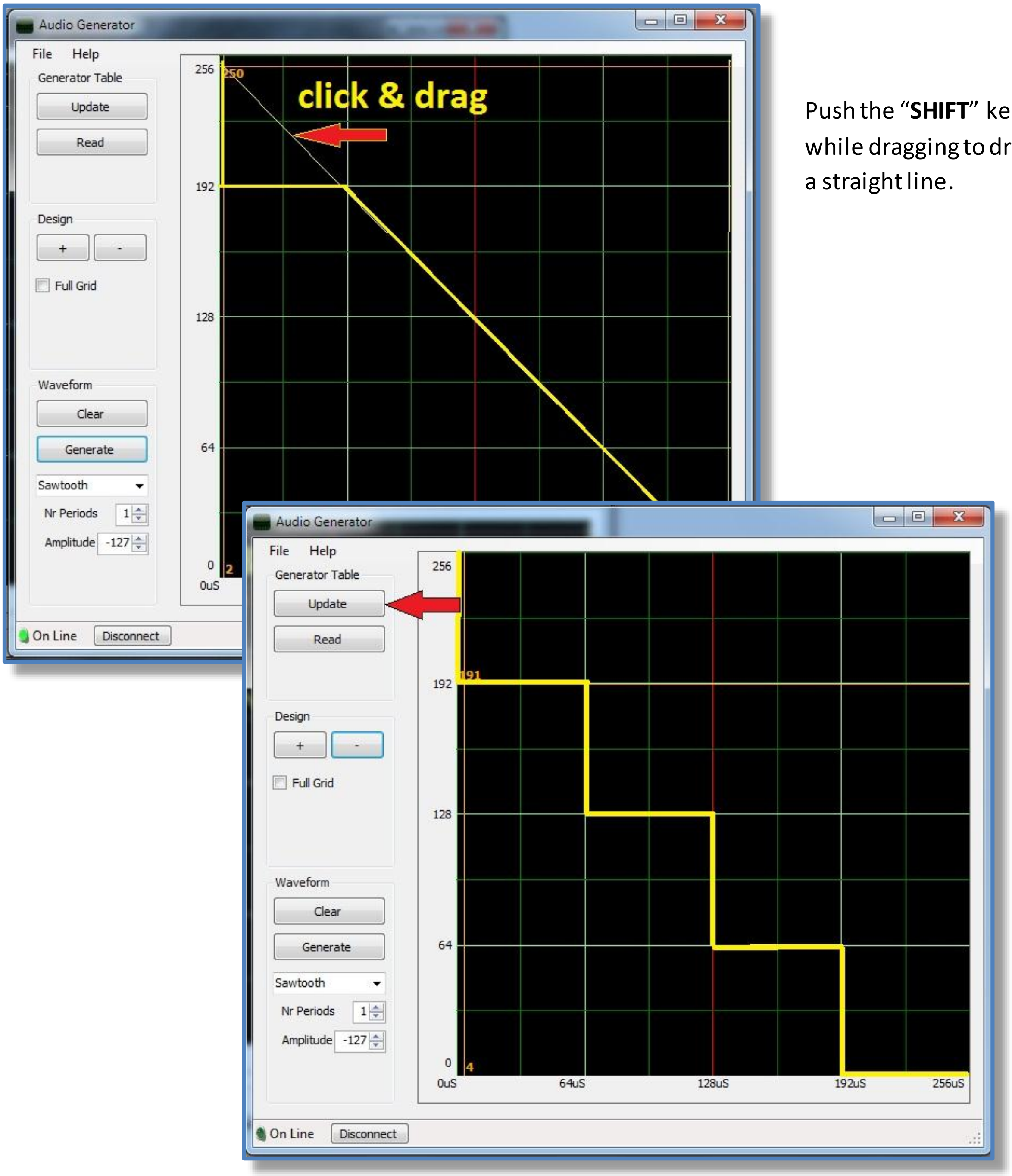




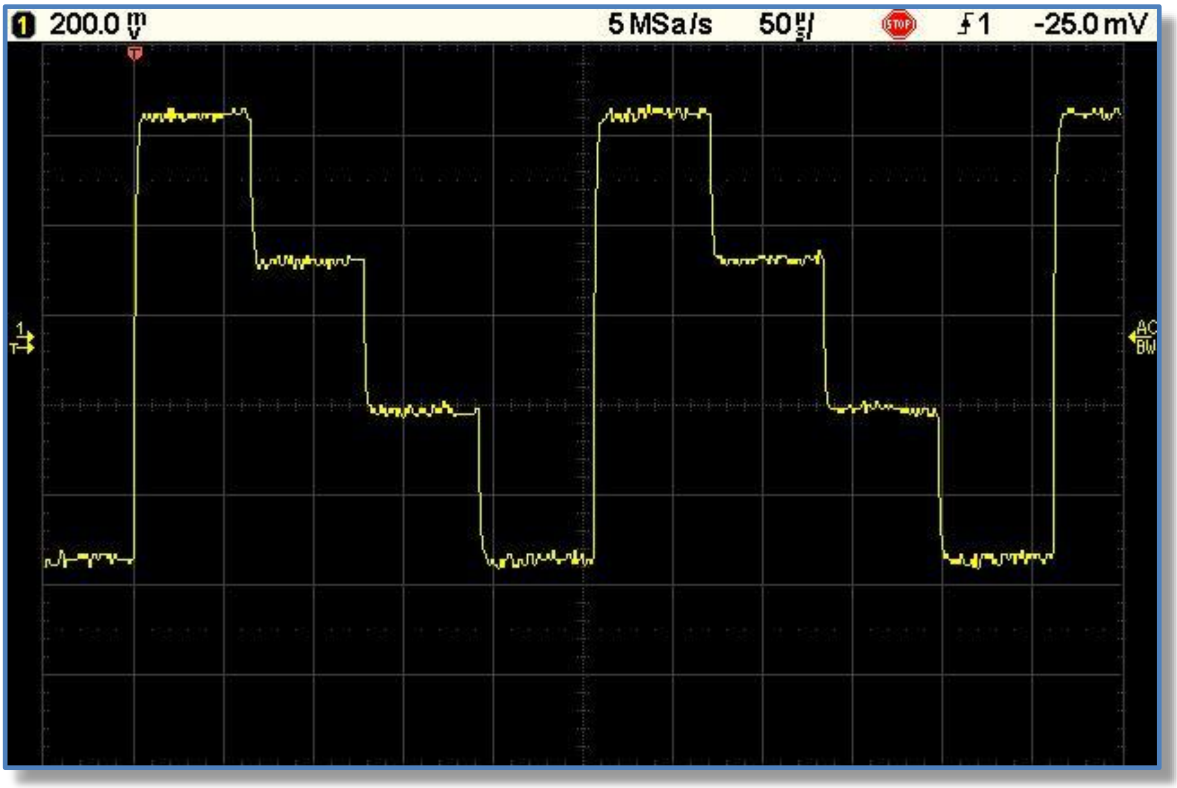
Sawtooth

Nr Periods 1

Amplitude -127



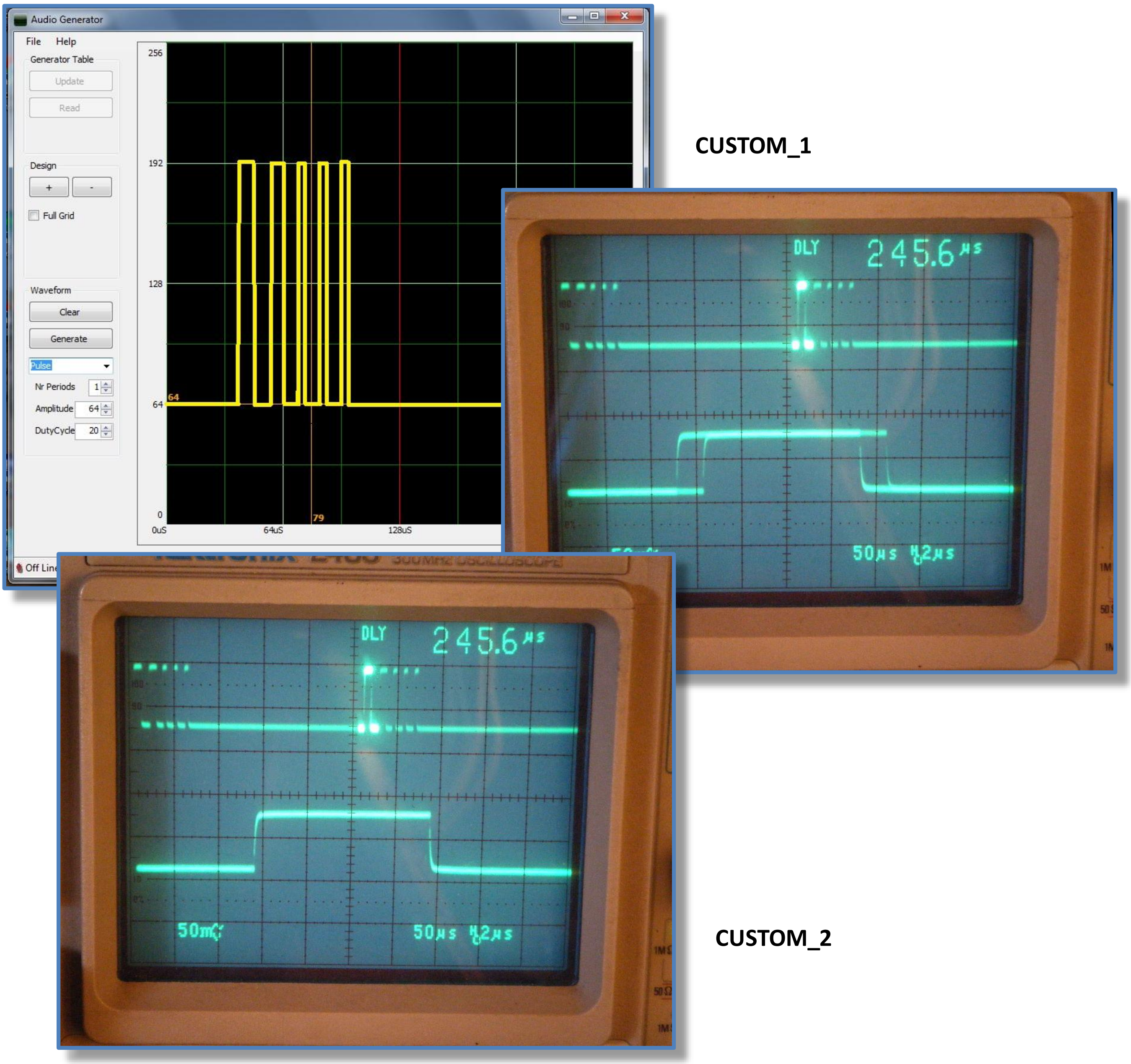
Push the "**SHIFT**" key while dragging to draw a straight line.


200.0 mV
5 MSa/s
50 µs/
STOP
1
-25.0 mV
T
1
AC
BW

8.4 Generating a custom signal

Two different algorithms are available to process the custom waveform table, please refer to the Paragraph “Project’s notes” for more detailes.

- algorithm **CUSTOM_1** manages the sampling rate the same way as done for the standard sine, sawtooth and square table: fixed clock and variable phase accumulator. Some embedded jitter due to the missed samples is possible.
- algorithm **CUSTOM_2** is always full reading the table and different frequencies are obtained varying the clock rate, this way the waveform is always drawn by 256 samples, *no aliasing, no jitter*. Being the sampling rate 1uS, the max frequency is limited to a 256uS period.



Pulse
Nr Periods 1
Amplitude 64
DutyCycle 20

CUSTOM_1

CUSTOM_2

9 Firmware update

The Arbitrary Audio Generator unit is based on Microchip's PIC18F24550 40DIP chip, the component's kit provided by FoxDelta includes the microprocessor already programmed and ready to work.

The firmware implements a special function ("bootloader" provided by Microchip) that allows to update the pic program memory to a new firmware version via the USB port, no external pic programmer is required.

Please refer to: http://www.microchip.com for any detail about the hex code linked to the Audio Generator firmware, any further information and copyright notice.

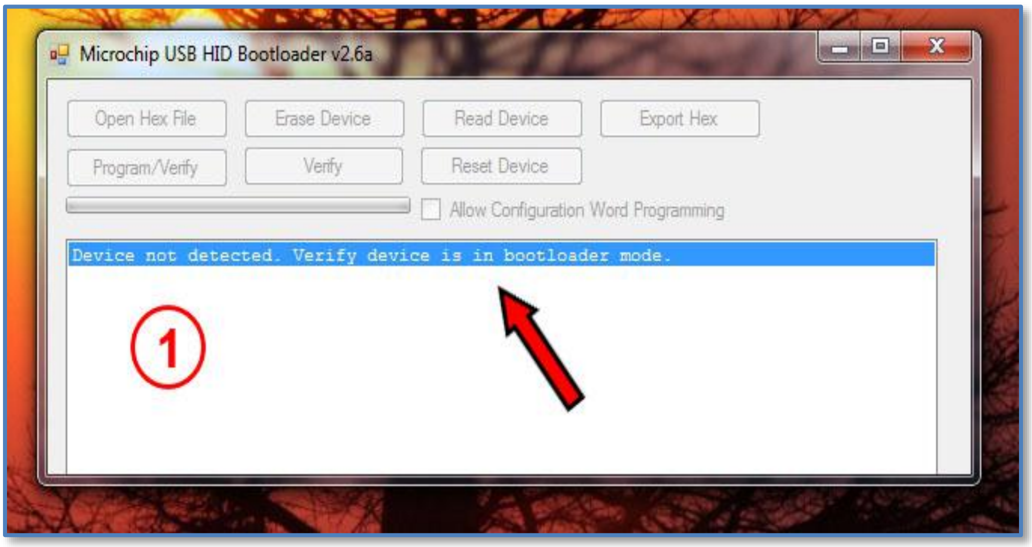
The program "HIDBootLoader.exe" (provided by Microchip) will be used, this is the Windows interface to access the bootloader function and flash the microprocessor memory.

To program from scratch the flash memory of the 18F4550 a pic programmer is needed, please burn the file: "**Audio Generator vx.xx FULL.hex**" included in the new releases

This file already implements the bootloader function.

To update an already programmed pic:

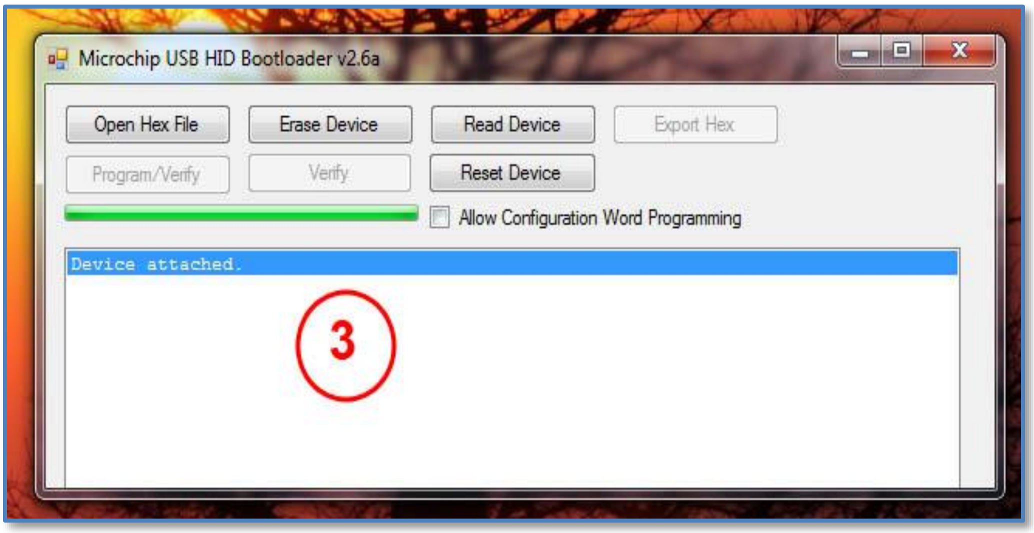
- Create a temporary folder and copy/unzip there the files:
  HIDBootLoader.exe
  Audio Generator vx.xx UPDATE.hex
- Navigate to the folder where the program "HIDBootLoader.exe" has been unzipped and launch it.



```
Device not detected. Verify device is in bootloader mode.
```

All buttons are disabled and the message: "Device not detected….." is presented

- **Press Button 1 while connecting the USB cable Audio Generator < –> PC.**



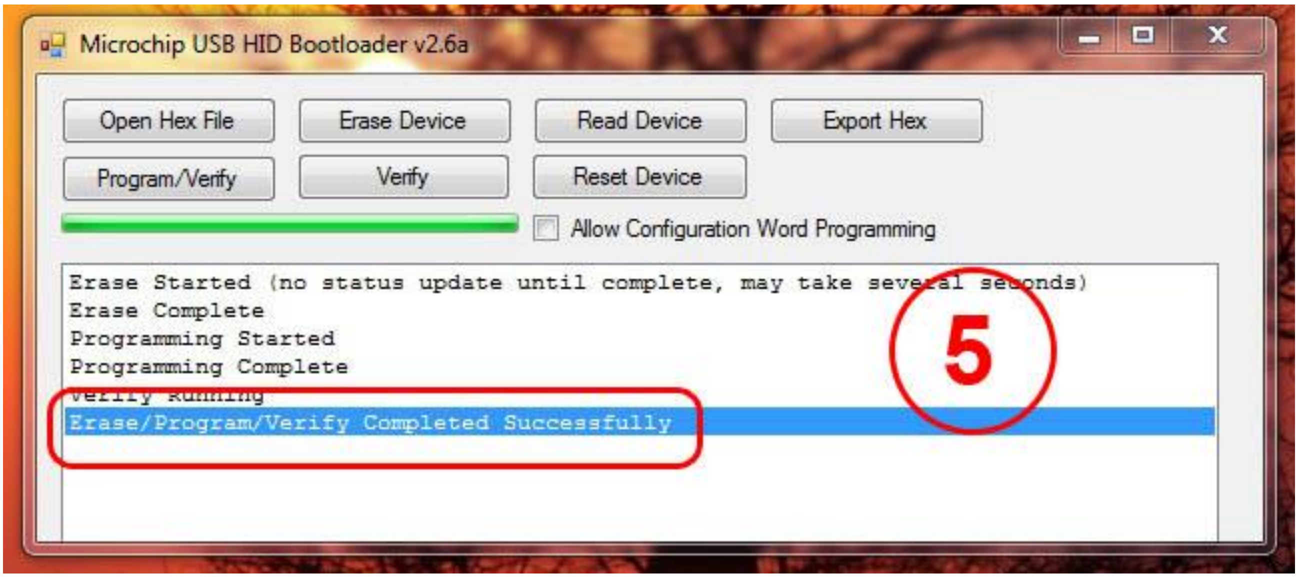
```
Device attached.
```

The green line starts, the Audio Generator board is recognized: "Device attached" and buttons are enabled.

- ➢ Click the button "Open Hex File", navigate to the folder where the new firmware hex file has been unzipped and select it.

Be aware to select the file "Audio Generator vx.xx UPDATE.hex".

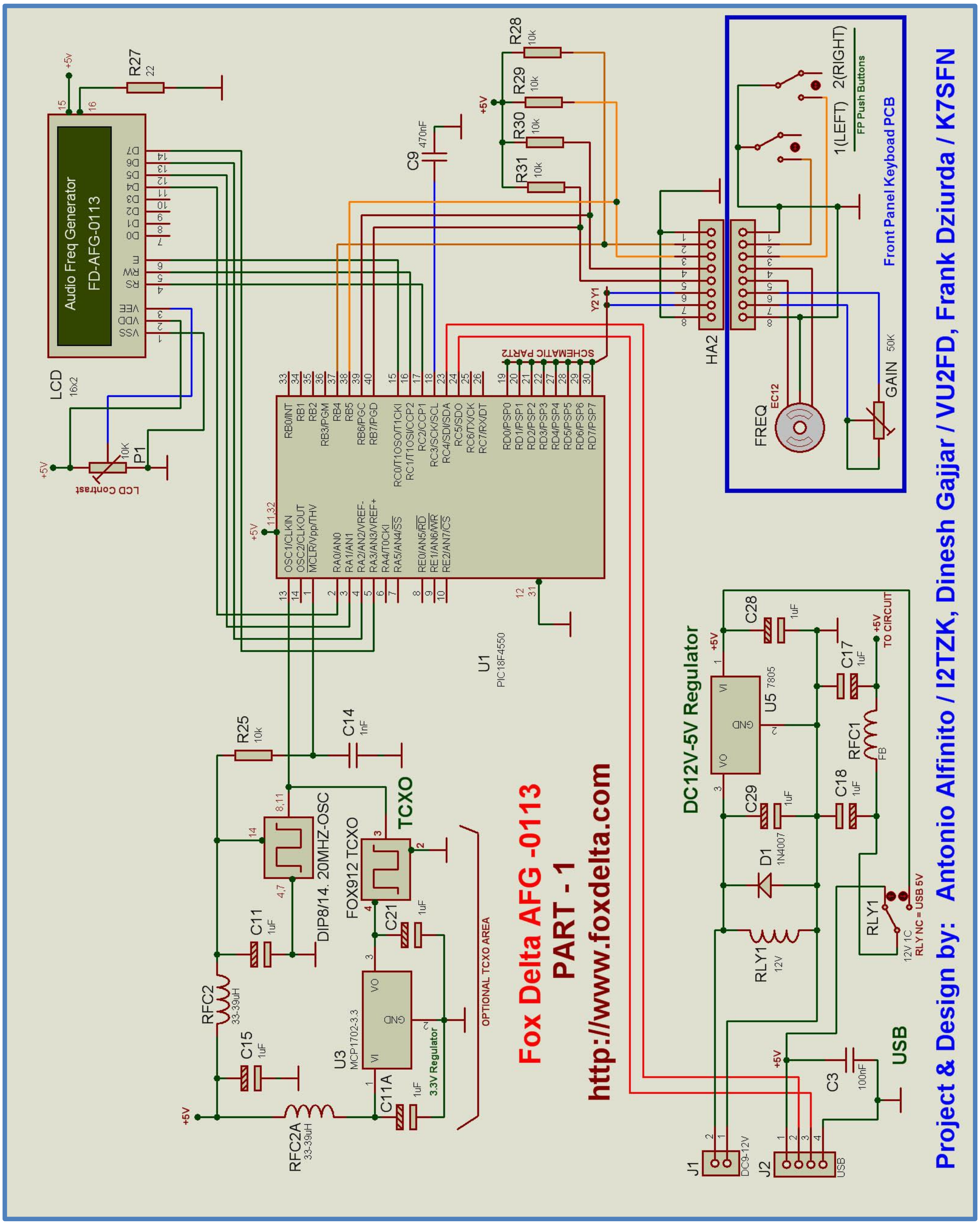
- ➢ Click the button "Program and Verify", **do not remove the USB cable during this step**, wait for the message: "Erase/Program/Verify Completed Successfully"

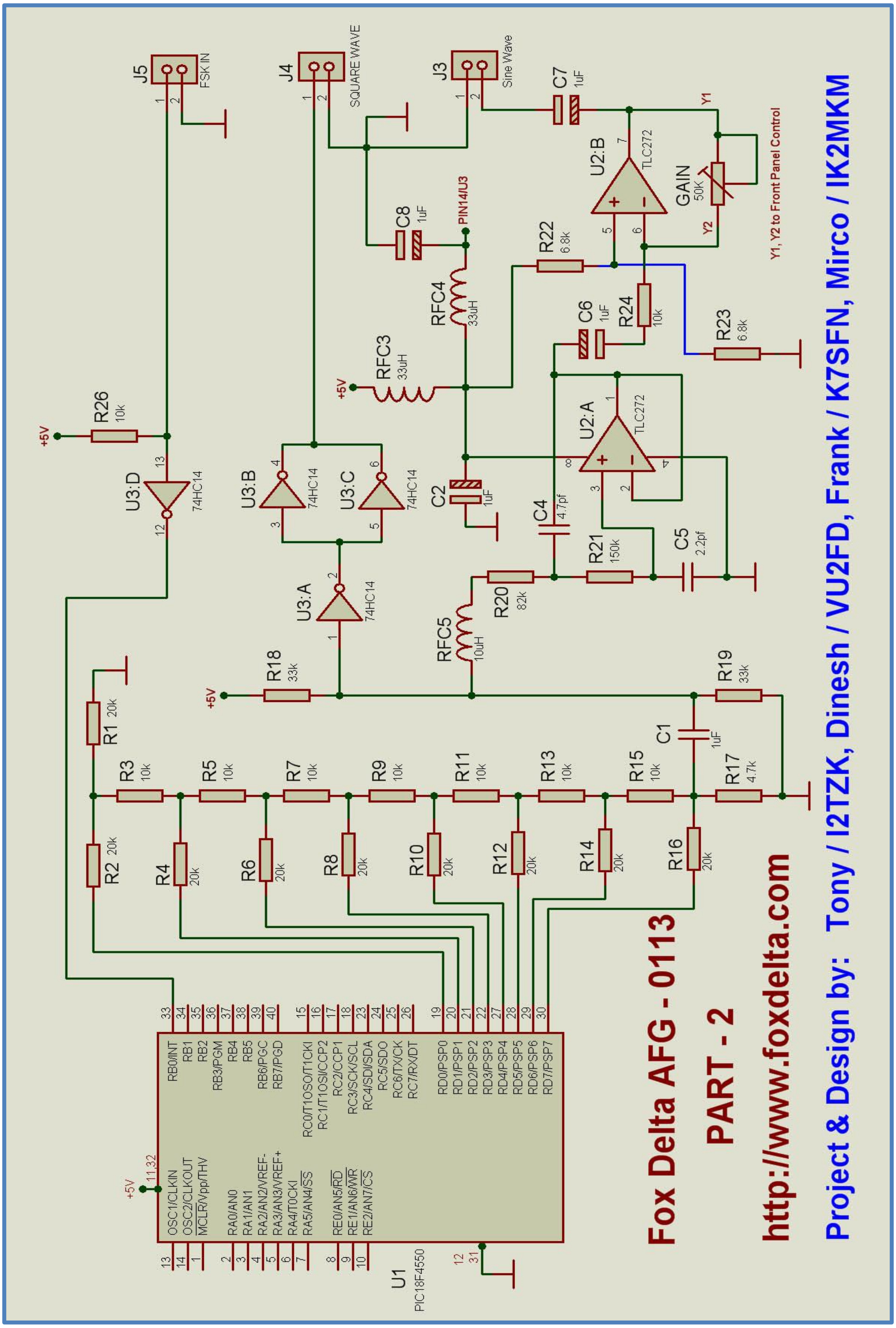


```
Erase Started (no status update until complete, may take several seconds)
Erase Complete
Programming Started
Programming Complete
Verify Running
Erase/Program/Verify Completed Successfully
```

- ➢ Remove the USB cable or reset the Audio Generator
- ➢ Verify that the f/w has been updated

10 Schematics





Fox Delta AFG - 0113

PART - 2

http://www.foxdelta.com

Project & Design by: Tony / I2TZK, Dinesh / VU2FD, Frank / K7SFN, Mirco / IK2MKM