



The FlowStone of Wisdom

More than a visual programming tool with a pretty (inter)face

By **Clemens Valens**
(Elektor.LABS)

Visual programming implies creating programs by manipulating functions graphically rather than by typing in text. In many visual programming languages (VPLs) screen objects (often boxes) represent functions interconnected by arrows, lines or arcs representing data relations. A well-known VPL is LabVIEW, but there exist many more. One of them is FlowStone, a VPL with a twist.

Flowstone, a sheet-like deposit of calcite or other carbonate minerals, is formed where water flows down the walls or along the floors of a cave. In FlowStone a program is formed by stacking layers of functions through which data flows. You may be able to formulate it a little better, but you get the idea of the analogy.

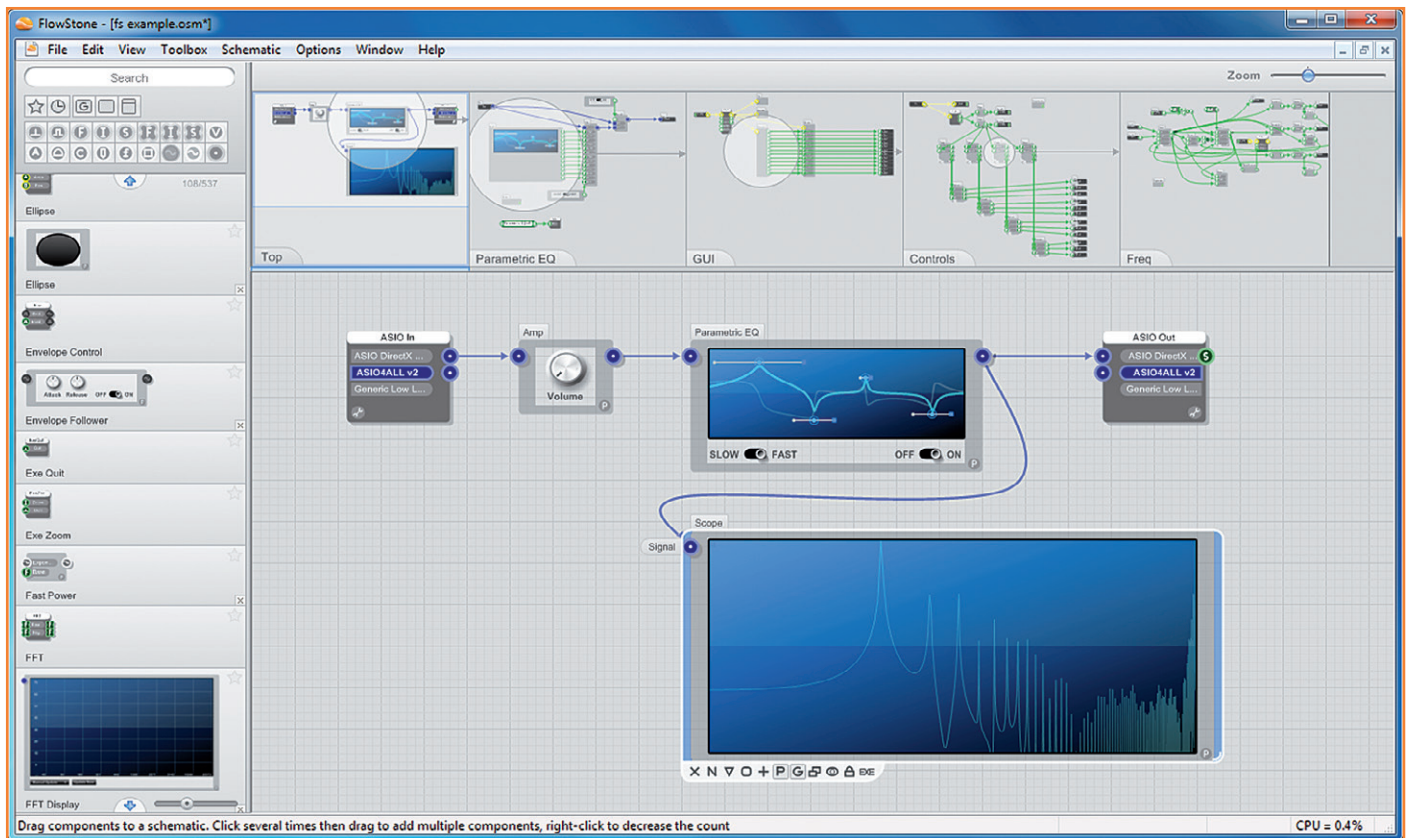
It all started some ten years ago with SynthMaker, an audio programming application that lets its user create virtual instruments, effects and controller plugins without the need to write a program. These instruments and effects can be used for making music using for instance Virtual Studio Technology (VST) compatible recording software. After a few years SynthMaker got a little sister called FlowStone. The baby sister turned out to be very demanding and she started to absorb her elder twin sister, the feat completed when she turned version 3 in November 2012. The medically inclined might speak of a case of *Vanishing Twin Syndrome* (VTS).

In FlowStone a program is drawn on a 1024 x 1024 grid called the Schematic. Function objects, or

components as they are called, are placed on the schematic. Components have connectors for receiving and/or sending data. The connections between the components — the links — are represented by lines that usually run from an output connector (start) to an input connector (end). Several types of connector are available for different types of data, each identified by a unique symbol and color. Schematics quickly become complex and so sub-circuits can be turned into modules for use as components in a schematic.

A module can have a graphical front panel with knobs and buttons and other controls. Once the application is ready, it can be exported as a stand-alone (native) PC application that runs without FlowStone or a virtual machine.

The FlowStone user interface is pretty slick and a lot of effort has gone into making components easily accessible and making navigation through the design fast. The component library — toolbox — can be searched in many ways using the component browser's filters, so if you really cannot



find the component you are looking for, then it probably does not exist.

The schematic grid is quite large and a special navigation window is available for quickly scrolling and panning through your design. It also features a thumbnail zoom view providing an overview of the design together with detailed views of some modules. The life of the experienced FlowStone user is made even easier by tons of shortcuts. Furthermore, a detailed user manual is available for download.

Looking through the toolbox you will notice a lot of audio functions and signal processing components, but more generic functions like text handling or line drawing are available too. Highly interesting also are the external hardware components that make interfacing to for instance a Wiimote (the remote control of a Wii game console), an X10 network (a popular home automation protocol) or Phidgets (low-cost USB building blocks for sensing and control) very easy. Special FlowStone hardware exists too, notably the FlowBoard DAQ, a board sporting eight analog

inputs, sixteen digital inputs and sixteen digital outputs. FlowStone now also supports communication through HID devices (see inset).

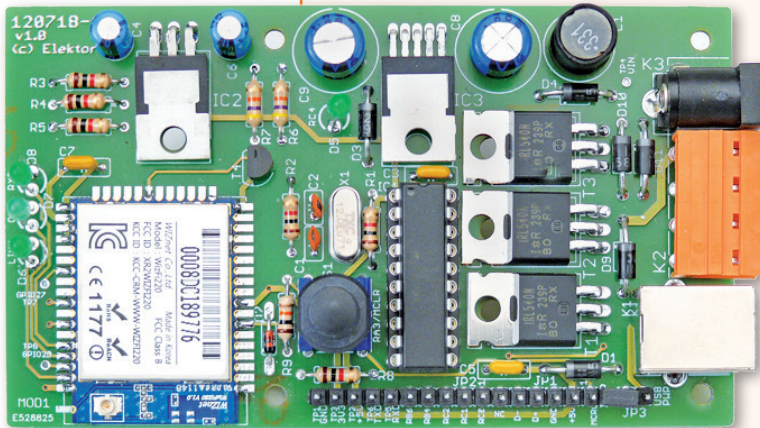
Creating a design in FlowStone is very easy (I did not say *working* design); you just drag and drop components from the toolbox onto the schematic (other ways of placing components are available too). If you place a component close to another with compatible inputs and/or outputs, the connecting links can be drawn for you, speeding up your work. When you select a component in the schematic a short help text is shown for its connectors, making it easier to understand their purpose.

Manually connecting components always starts at an output and ends at an input. Usually the outputs are on the right side of a component and the inputs on the left. When dragging a link to a component, only the connectors compatible with the data type transported by the link will be accessible.

Links do not always have to be drawn, they can

Figure 1. A FlowStone schematic showing simple audio processing using a parametric EQ and a display to plot the FFT of the resulting signal.

Control your own hardware



The latest version of FlowStone is capable of communicating with HID devices, meaning you can develop your own HID board and control it from FlowStone. The Elektor Wi-Fi Controller Board [2] for instance is such a compatible board. Because it has a bootloader it is a perfect platform for developing FlowStone controlled applications. Create some sort of music application like a synthesizer or a complicated sound processing tool and use the Wi-Fi Controller Board to create accompanying light effects or make a robot dance. Or use FlowStone for your hi-tech scientific data acquisition and processing application and control the physical process with the Wi-Fi Controller Board. I have given it a try and it works fine. To give you a head start my C18 project can be downloaded (for free, of course) from [3].

also be wireless. Such links are similar to the net labels found in schematic capture programs with one subtle difference: wireless outputs can only send to modules on a lower level in the hierarchy.

FlowStone recognizes more than 30 data types, divided in three categories: streams, events and triggered. Streams are continuously flowing data streams like audio or video samples. Triggered types and events are signals caused by events. The difference between these two categories is that triggered types only signal that something has changed whereas events (can) carry data. Also, events are scheduled, meaning that they only occur at times specified by the programmer. Interestingly, streams come in two flavors, monophonic (mono) and polyphonic (poly), and the way they behave is quite different. As the user manuals states: *"Poly is only used for audio applications where sound signals are generated from MIDI notes. If you're not generating audio in this way then you can ignore Poly completely."*

Even though FlowStone is a graphical programming language, it is easy to write (part of) a program in the traditional way using the Ruby programming language. This of course violates the graphical programming paradigm, but it is a logical option as some functions may be easy to draw whereas others may be more quickly implemented by writing an algorithm.

Unfortunately, space restrictions for this article do not allow an in-depth review of FlowStone and all it has to offer, which is plenty. If you want to play around with the tool yourself, I suggest you download the free demo version from [1]. If you happen to own a copy of the latest version of the digital audio workstation (DAW) FLStudio then you already have FlowStone as it is part of the package (older versions of FLStudio shipped with SynthMaker).

(130064)

Internet Links

- [1] Flowstone main page: www.dsprobotics.com/flowstone.html
- [2] Wi-Fi Controller Board: www.elektor.com/120718
- [3] Downloads for this article: www.elektor.com/130064