

FlowPaw: A Build Your Own Gizmo Kit

David Hayward
samples the latest
educational
innovation

Ever since the Raspberry Pi was launched, we've seen something of a revival in schools with regards to how IT/computing and technology is being taught. What the Raspberry Pi gave teachers was a cost effective resource to help expand the teaching of basic electronics, programming and the interaction between the two.

Since then, schools have adopted this technology into the new curriculum. Students are happy to get their hands on working models, robotics, flashing LEDs, buzzers and so on, and no doubt the teaching staff are equally pleased with the levels of attention in the classroom.

This is a revival of attitudes not really seen since the late 70s and early 80s, and following this ethos is the FlowPaw, a recently developed electronics and programming kit, which has just finished a funding campaign on Kickstarter.

Made by DSP Robotics, the FlowPaw is a multi-purpose electronics board that can accommodate up to four different expansion modules (known as claws) at once. The modules in question range from an IR proximity sensor, 8x8 LED matrix display, a three-axis accelerometer and buzzer, through to a joystick control, DC motor controller, compass and a Bluetooth component.

With the modules connected, the student hooks the FlowPaw up to a PC via a spare USB port and interacts with the board and modules through a graphical interface called FlowStone. By using FlowStone to connect the various abilities of each of the modules, the student can create a vast number of experiments, games and basic electronic applications that are outputted through the FlowPaw itself.

Running With All Paws

Getting started is simple. You unpack the FlowPaw main board and gently insert the four modules that comes included with the starter kit: the IR proximity sensor, 8x8 LED matrix display, three-axis accelerometer and buzzer. After that you install the FlowStone software and connect the FlowPaw to your PC with the included USB cable.

Once your PC has recognised the hardware, you can launch FlowStone and start to experiment with the newly connected hardware.

FlowStone

FlowStone is a programming language based on graphical components, the components in this case being the modules and hardware of the FlowPaw.

It's a visual language much in the same was as Visual Basic, to some degree,

where you would drop a component onto the work area, then write the code to control it, followed by connecting the component with other lumps of code and components. The result is a continuous flow circuit of interconnected components and snippets of code, which when executed will perform a certain task on the FlowPaw.

FlowStone starts off with a main schematic window, a graph-paper looking setup where you will ultimately drag and drop the various components to. Down the left side of the screen you'll find the toolbox, which houses various components, which are the building blocks of your program and the method with which to communicate and interact with the actual hardware on the FlowPaw.

Above the components are the types of connectors. These have different inputs and outputs based on how they will interact with the components. For example, you could drop the FlowPaw board component on the schematic, followed by a trigger component, text box and boolean component. You can then draw connector lines from each of the components, so that when the boolean component is set to true and the trigger is activated, the firmware number of the FlowPaw will appear in the text box.

When combined, these elements will allow you to control the behaviour of each of the 'claws' of the FlowPaw, as well as any other real-world components you attach to the setup – like robot arms, controllers and such. To better control each component and to tweak the behaviour of any of the above processes you'll need to start using FlowStone modules.

The modules are the programming backbone of FlowStone and are made up of groups of components that perform their own custom definitions depending on the input or output of other components or the current state of anything attached to or interacting with the FlowPaw. For example, it could be a switch component together with values for on or off states, with perhaps an added true or false option to activate the buzzer when the switch is clicked to the on position, and be grouped together into a single module as defined by its own separate FlowStone schematic. With the module created, you can then save it and use it as an individual routine whenever you have a need for it.

Furthermore, there's the ability to program your own custom components using FlowStone's built-in Ruby editor. By

“ The sheer number of variations and components available mean you can create some extremely complex graphical programming routines that will interact with the real-world components attached to the FlowPaw ”

defining your own component and mixing it with other default components, you can develop pretty much anything you want as a self-packaged module. Obviously there's the matter of learning how Ruby works, but that's something that can

come in time and as the student or user expands their knowledge.

It's the same logic that applies to traditional, non-graphical programming. You'll have a number of components and libraries you can call on to produce an

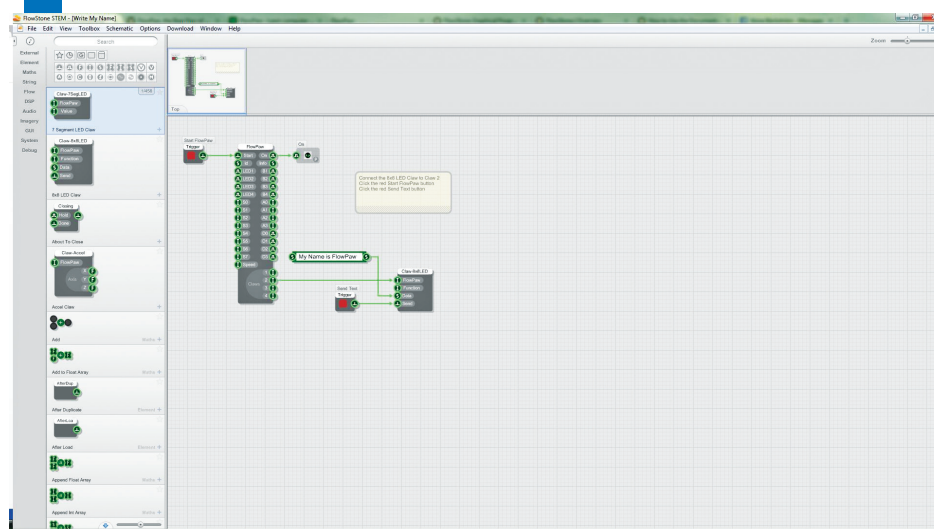
Ruby, Ruby, Ruby, Ruby

To quote DSP Robotics, “Ruby is a general-purpose object-oriented programming language designed from the outset to enhance productivity and at the same time be extremely fun to use.”

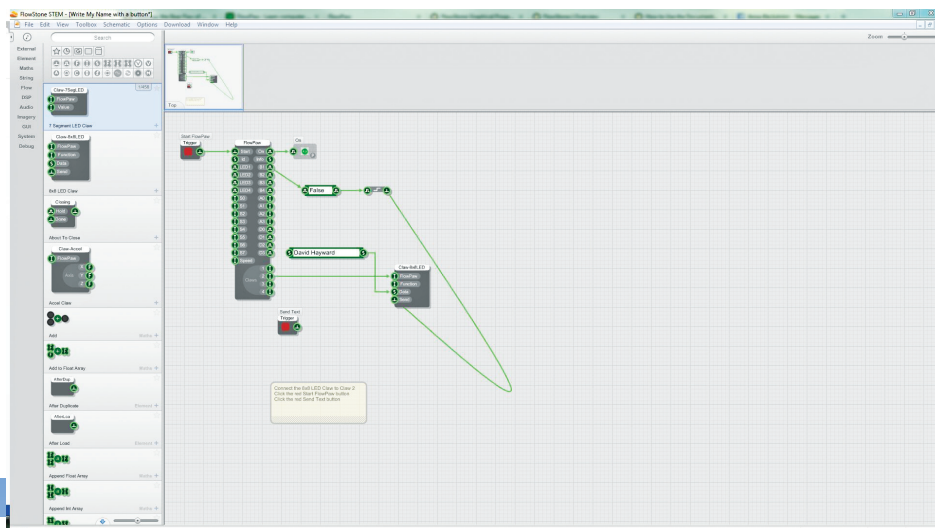
It's actually quite a beautiful balance of Perl, Lisp, Ada and number of other examples all bundled in one reasonably straightforward concept, while still remaining complex enough to tackle everyday coding.

There are naturally many examples of learning how to use Ruby available, but one of the best we've come across is the 'Try Ruby in your browser' example, from the main Ruby development site.

By visiting tryruby.org/levels/1/challenges/0, you can be guided through some simple tasks that will ease you gently into Ruby. If, though, you're already a little more experienced in the likes of Python, then as we said in the main body of the article, you can always hack one of the modules created by the FlowPaw team to see what's going on behind the scenes.



▲ The core 8x8 LED example is simple to understand



▲ The LED example can be hacked to include other elements

effect, which can then be used in different routines to accomplish the same goal. A collision detection routine, as an example.

Building With Flowpaw And Flowstone

The sheer number of variations and components available mean you can create some extremely complex graphical programming routines that will interact with the real-world components attached to the FlowPaw. How complex you want to get is purely up to you or the students using the hardware and software.

Thankfully, though, this level of complexity isn't in your face to begin with. The FlowPaw team has included 25 examples ranging from operating the buzzer by pressing the claw buttons, through to unique versions of *Pong* and *Space Invaders*, where a mixture of the accelerometer claw and the FlowPaw buttons can be used to control your character on-screen.

One of the most effective ways to begin building with the FlowPaw, though, is to hack an already created example. In this case, we'll take a quick look at how the FlowPaw interacts with the 8x8 LED matrix.

By loading up the 'Write my Name' example, we can see that the schematic is made up of six simple components: the FlowPaw board, a power trigger, a power on/off indicator, the 8x8 LED claw control component, a text box and a trigger for the 8x8 LED claw.

The FlowPaw board is placed down on the FlowStone main window, and the power trigger and power on/off indicator are placed either side, with connectors

lined up from the Trigger to the FlowPaw board start pin and from the FlowPaw On pin to the on/off indicator.

Next, we drop the 8x8 LED claw to one side of the FlowPaw board and link up the connector for Claw 2 – the physical claw number we've plugged the LED matrix into – to the claw pin labelled FlowPaw. This has established a connection to the claw and the main FlowPaw board.

We can now add a simple text box and draw a connector from its output pin (the right-hand pin) to the data pin on the 8x8 LED claw. And finally, we can drop a trigger component and use a connector to link up the output of the trigger to the send pin on the 8x8 LED claw.

With everything in place and the FlowPaw now connected to the PC, we're able to enter any text we like into the text box, click the trigger to power on the FlowPaw, then click the trigger to send the text (the input data) to the 8x8 LED claw.

The result is whatever you typed into the text box streaming across the 8x8 LED claw. But what if we wanted to expand on that slightly?

Take a true or false boolean component and a bool to true trigger component from the toolbox and place them on the schematic.

Draw a connector from B1 (button 1) on the FlowPaw hardware to the input on the true or false boolean component, then another connector from the output to the input of the bool to true trigger component.

What this does is check if button 1 on the FlowPaw is pressed, and if it is, the value will be true and pass the value through the circuit.

Finally, a connector can be drawn from the bool to component's output pin to the send pin on the 8x8 claw – first removing the trigger component that previously sent the text.

This will then display the text across the 8x8 LED claw only when button 1 on the FlowPaw has been pressed.

That's a reasonably simple example, and naturally it can be expanded to include the output of a game, where it will display 'Game Over' should the value Lives equal zero. Or in terms of hardware, if the IR proximity sensor detects someone near it, then a warning could flash up or a welcome.

The emphasis, however, is on experimentation and exploration of the different components and how they interact and can be programmed to behave and control the real-world hardware in front of the student or user.

Extended Learning

The IT lesson these days does appear to be a far more interesting place to be than it was when we were at school. There was only so much you could do with five BBC Micros and a dot matrix printer that weighed the same as a baby elephant, especially when there were four students to every machine.

Now the lessons could incorporate the likes of the FlowPaw and can be designed to spark the interest in both programming and electronics that we could usually only manage through a local computer club or at home.

There's a huge potential for the FlowPaw and its connecting software for use in schools, for both younger and older students. And while it may not

“ There was only so much you could do with five BBC Micros and a dot matrix printer that weighed the same as a baby elephant ”

FlowPaw, Find Out More

You can find more information on the FlowPaw and FlowStone by visiting the following links:

- The FlowPaw Kickstarter entry – goo.gl/UvTakU
- The FlowPaw Getting Started guide – goo.gl/Rddehp
- The FlowStone introduction – goo.gl/M9wvXI
- A FlowStone overview – goo.gl/KFHvk1

delve into the kind of electronics that you'll find in the workplace or as you get older, it's the building blocks and foundation for wanting to learn more and developing an interest and knowledge of how everything works.

Conclusion

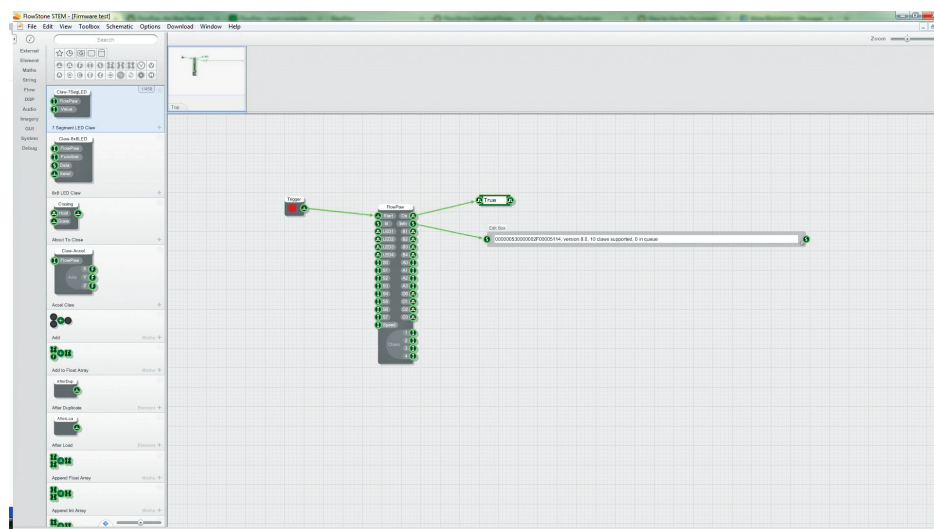
These certainly are interesting times we live in. The wealth of resources available to a student are simply staggering, so when something that concentrates these collective resources into a single, easy-to-use kit comes along, we are in some ways quite envious of the modern student.

While the FlowPaw exists as a learning platform for students to get their first taste of electronic and visual programming, it can also be considered an ideal add-on board for those who require slightly more focused control over a particular element of their own projects.

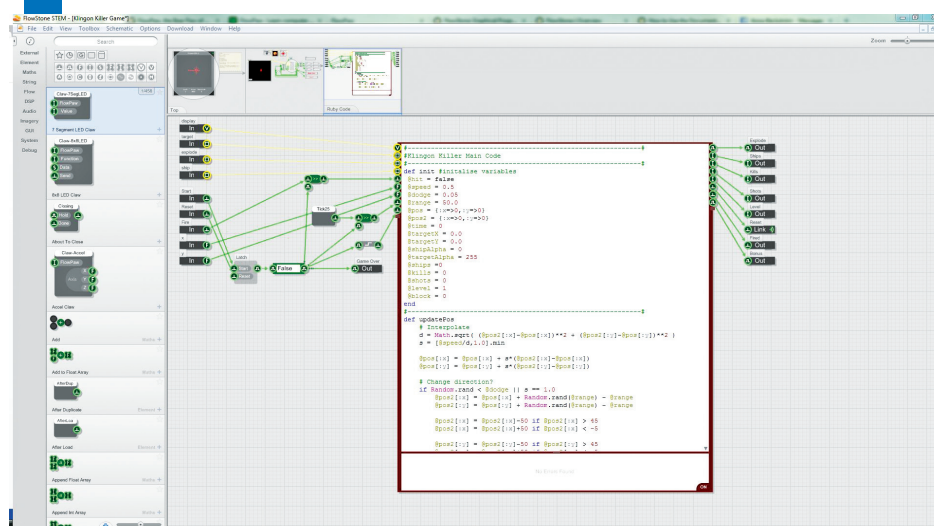
The FlowPaw is capable of hooking up to a Raspberry Pi and interacting with it at the hardware level, which will increase the capacity of both immensely. But on its own there are a number of intriguing possibilities where the increased complexity of the software and hardware could very well prove more effective than a few Python scripts on a Raspberry Pi.

In the end, it's what you put into it that makes using the FlowPaw such an interesting and wonderful concept and design. There's plenty here to keep even the most cynical of students interested for an entire term, not to mention a single lesson or two.

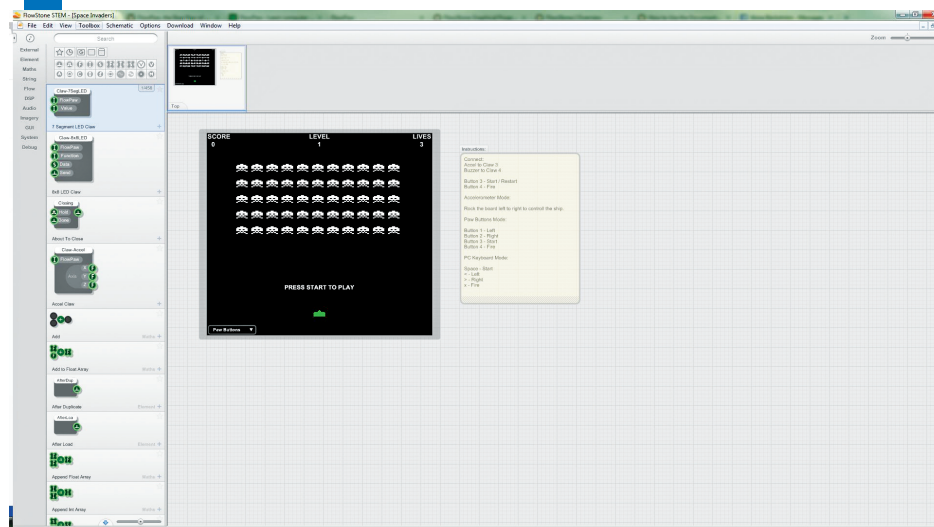
With innovation like this, and with the potential and promise for more claws in the future, there's no excuse now for the next generation to have an understanding of the technology in use around them and perhaps to take us onward and upward. **mm**



▲ The basic FlowPaw firmware test is a good example of how Components and Connections work



▲ You can expand your FlowStone modules to include unique Ruby code for better effect



▲ The examples even come with a FlowPaw enabled Space Invaders game