

TUTORIAL

FLOWSTONE



 DSP Robotics

Version 1.0

C O N T E N T S

CHAPTER 1	1 Introduction	3
	ABOUT THIS GUIDE.....	4
	THIS TUTORIAL.....	5
	PROJECT OUTLINE.....	5
	WHAT'S COVERED.....	5
	SOURCE FILES.....	6
CHAPTER 2	2 The Tutorial	7
	THE ENVIRONMENT.....	8
	TOOLBOX.....	9
	NAVIGATOR.....	10
	SCHEMATIC WINDOW.....	10
	STEP 1 - THE COUNTER.....	12
	STEP 2 – CONTROLLING COUNTER SPEED.....	14
	STEP 3 – SPLITTING THE COUNT & BUSES.....	16
	STEP 4 – CREATING THE LEDs.....	19
	STEP 5 – ARRANGING THE GUI.....	21
	STEP 6 – FINALIZING THE GUI.....	22
	STEP 7 – MAKE AN EXECUTABLE APP.....	23
	PROJECT EXPANSION.....	27

1

Introduction

ABOUT THIS GUIDE

About This Guide

This guide takes you through the process of creating an example schematic using FlowStone.

For information about individual components and modules that ship with the software, see the Component Reference and User Guide guide. These documents can be found on the Manuals section of our web site at:

<http://www.dsrobotics.com/manualsarea.php>

Additional information and articles about the software can be found at:

<http://www.dsrobotics.com/support>

If you have any comments about this guide please email them to info@dsrobotics.com.

This Tutorial

In this tutorial we are going to take you through a step by step guide on how we built the Knight Rider LED example FlowStone project that is included with the FlowStone install.

Project Outline

This project displays 8 LEDs on a custom GUI (Graphical User Interface) that switch on and off in sequence. The speed of the LED sequence is to be controlled by a graphical slider. Finally a stand alone EXE application (app) is to be made and run.

What's Covered

During the course of the tutorial we are going to cover:

- IDE Layout (Integrated Development Environment)
- Components (The building blocks of FlowStone)
- Modules (Groups of components)
- Links (Connecting Modules and Components)
- Ticks and Triggers (Creating events)
- Busses (Multiple links in one wire)
- Wireless Links (No Wires)
- Data Types (Different types of numbers and strings etc.)
- Synchronized Modules (Modules that maintain the same structure when any one is edited)
- Creating a GUI (Graphical User Interface)
- Creating an app. (Making an EXE files that runs independently of FlowStone)

Source Files

For each step in the tutorial we've created a schematic (.fsm) file. These are stored in your FlowStone Install folder in a sub folder called "tutorial".

Usually this folder is located here:

<C:\Program Files\DSP Robotics\FlowStone>

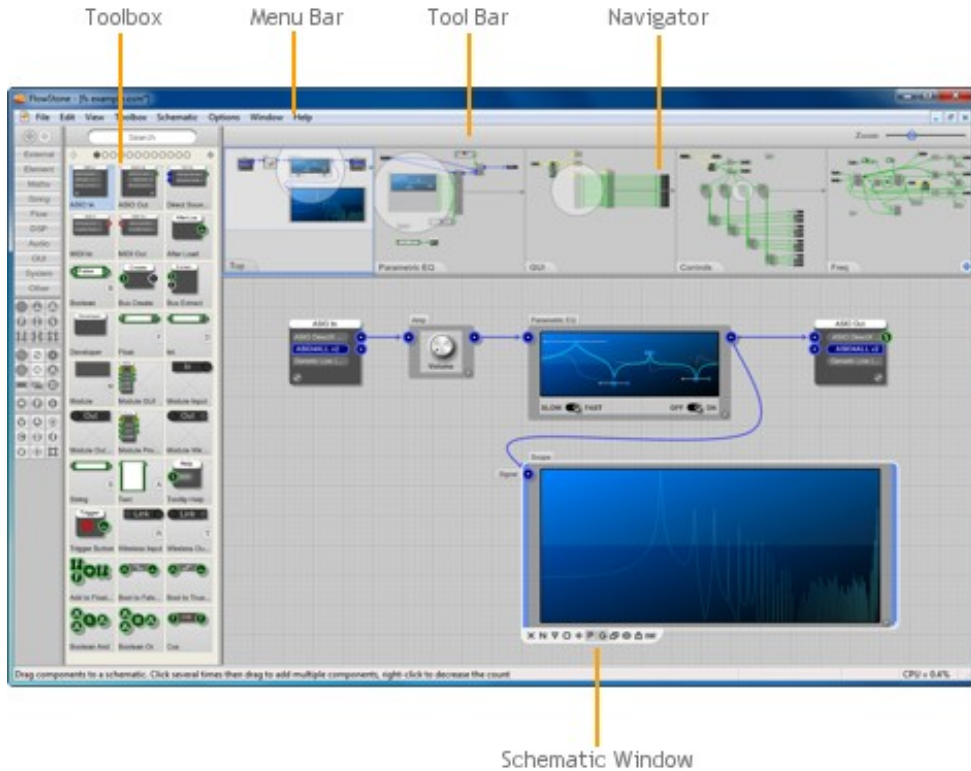
You also access these files easily from the Help menu under the Help | Tutorial sub menu by selecting Show Schematic Files.

2 The Tutorial

CREATING THE 'KNIGHT RIDER' SCHEMATIC

The Environment

Before we start we'll take a quick look over the FlowStone user interface. Open FlowStone and familiarize yourself with the IDE (Integrated Development Environment).



To start a new project, select New from the File Menu - a new blank schematic will be created.

Toolbox

Running down the left-hand side of the application is the **Toolbox**. The main part of the toolbox is the Component Browser. This shows every component and its name. You can scroll through the components by clicking on the page buttons.

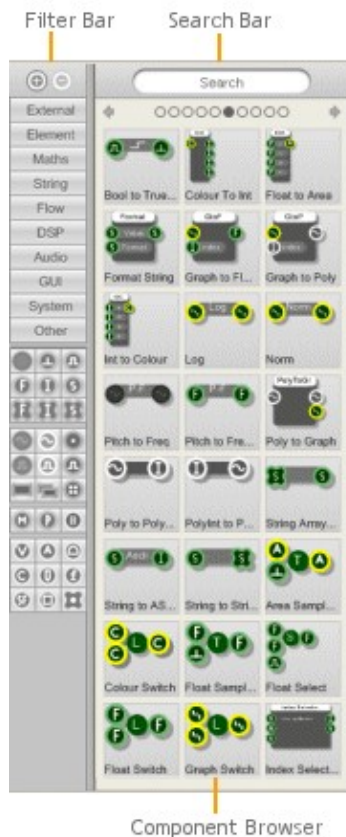


You can also scroll by clicking on the arrows to the left and right of the page buttons or by pressing the PGUP and PGDN keys.

Typing in the search box will make the browser only show components containing the search text.

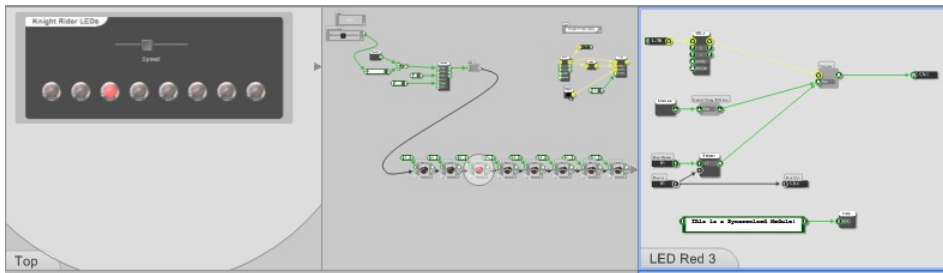
On the far left side is the Filter Bar (starts External, Element, Math etc.) This also filters the browser making it only show components that match the group or data types you select.

Have a click around and see how the browser changes.



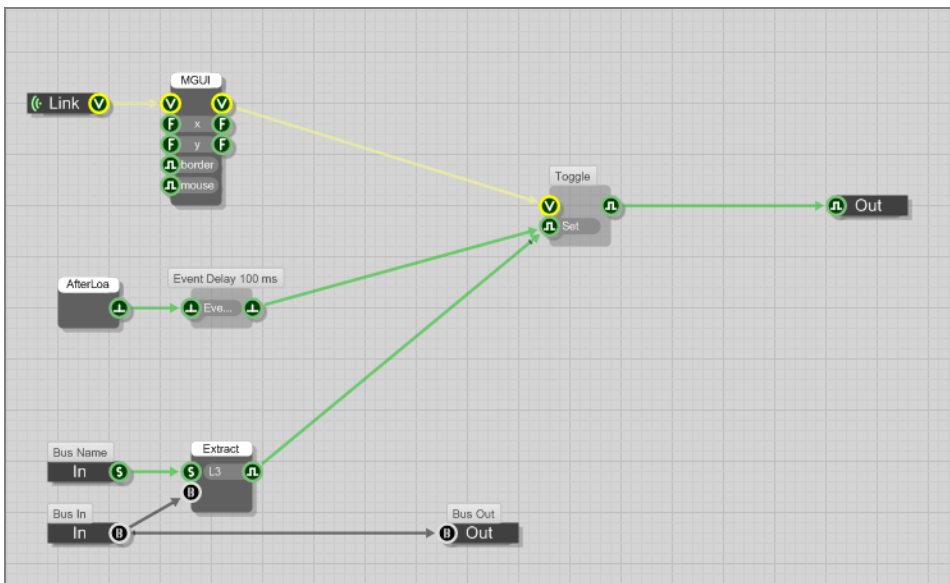
Navigator

Across the top of the window is the **Navigator**. As its name suggests, this is used for navigating through your schematic. You'll find this invaluable once you start delving deeper inside modules.



Schematic Window

Underneath the Schematic Navigator is the **Schematic Window**. This is where you create your program by dragging components in from the Component Library and connecting them with links or wires.



Step 1 - The Counter

(Tutorial1.fsm)

In order to select each of our 8 LEDs we are going to use a simple up/down counter that counts from 0-7 (8 steps) and 7-0 , controlled by a timing tick.

Type 'counter' into the search bar on the Toolbox. You will see several components displayed that are relevant to counters. If you hover your mouse over each component you will get a text bubble popping up with a further explanation of what that component does.

We want to use the 'Counter Advanced' for this project. Left click on this component and drag it to the Schematic window.

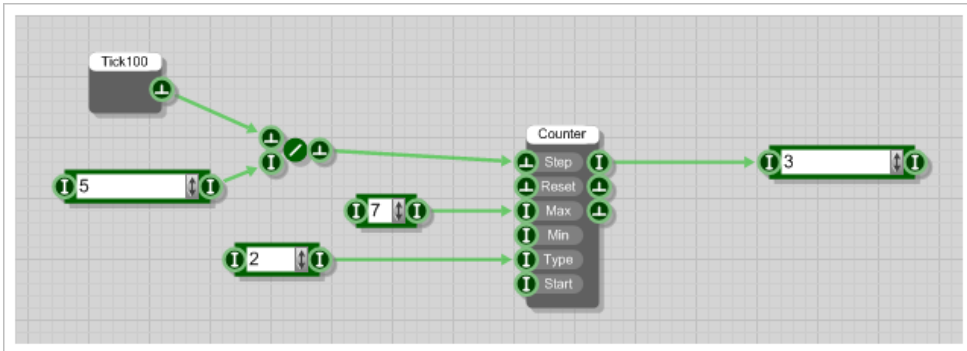
Next type 'tick' into the search bar and drag the 'Ticker 100' into your project. This component outputs a timing tick 100 times a second and the output type is Trigger (Indicated by the upside-down T symbol). This is a little too fast for our project so we need to slow it down. To do this we use a 'Trigger Div' (Type Trigger Div into the search box), add one of these to your project and connect the output of the 'Tick 100' to the Trigger input of the 'Trigger Div' (same Symbol). To do this Click and drag with the left mouse from the output of the 'Tick 100' to the trigger input of the 'Trigger Div' , a stretchy wire will follow the mouse until a connection is made.

Next we want to tell the 'Trigger Div' how much to divide by, to do this we need to connect an Integer (Whole Numbers, 0,1,2,3, -1,-2,-3 etc) input value to the 'I' – or Integer Input. You can search for Integer in the toolbox or simply click once on the schematic window where you would like the component to appear and press 'D' for the keyboard short cut. An Int (Integer) edit box will appear, then connect the 'I' output to the 'I' input of the 'Trigger Div'. To give it a value type say '5' into the Int box.

Now connect the output of the Trigger Div to the Step input on the Counter Component.

For our counter we don't need to set a value to every input on the module as the default values will do in most cases. The only two inputs we need to set are the 'Max' input and the 'Type' input. Add two more Integer boxes and set the Max to 7 and the Type to 2 (Up/Down).

All done! To test our counter add a further Int Box to the 'I' output on the Counter Module and you should see the count value going up and down.



Our counter all wired up!

Well done, you've now created your first working FlowStone program! There is no compiling necessary with FlowStone, all projects run in real time all of the time so your program is now running!

Step 2 – Controlling Counter Speed

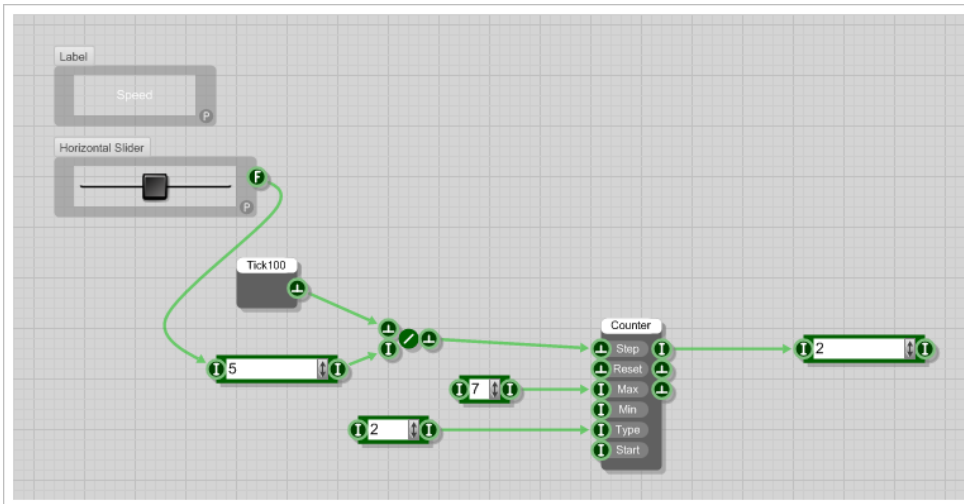
(Tutorial2.fsm)

To be able to dynamically control the speed of our counter we are going to add a graphical slider to the project. To do this, type 'Slider' into the Toolbox Search Bar and drag the Horizontal Slider Module into your project. You will see that the output type of the slider component is 'F' or Float (Meaning Floating Point numbers : 0.01, 10.75, -5.23 etc.).

By default the output of our Slider is between 0 and 1 in thousands of little steps. To test this connect a Float component to the output of the Slider (press 'F' or drag a Float component in from the toolbox). Moving the slider you can now see the output value.

For our project we need an output value of between 1 and 10, to edit this value on the Slider Component click on the small 'P' button on the bottom-right side of the Slider module. This will open the Properties for the Slider module. Change the Max value to '10' and the Min Value to '1', leave the rest as it is. Now close the Properties by clicking on the 'X' button again on the bottom right. Moving the slider will now output values from 1 to 10.

Now Connect the output of the Slider to the Int box connected to the Trigger Div. Doing this automatically converts the Float output of the Slider into Integers (Whole Numbers) in the Int Box. This can be seen now when you move the Slider the values in the Int box are 1,2,3,4, etc.



Slider Control of the counter speed

If you move the slider you will see the speed of our counter now changes with the slider value. There's just one final thing we need to add, and that's a Label. This will be used to tell user what the Slider is controlling i.e. Speed.

CHAPTER 2

To add a Label, type Label in the toolbox search bar and drag in a Label Module. To change the label text, open the properties panel for the label (it's the small 'P' button again). Change the text to say 'Speed'. You can also change the colour here, so change it to White by clicking on the Colour well. The physical position of the graphical components doesn't matter at this stage as they can be arranged later when we make our GUI (Graphical User Interface).

Step 3 – Splitting the Count & Busses

(Tutorial3.fsm)

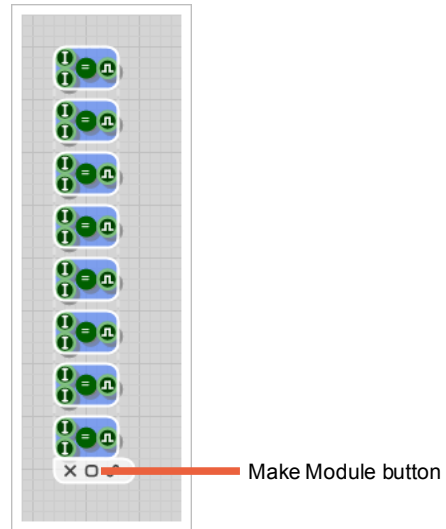
We now need to split our count into 8 separate pulses. To do this we can just use an Int Equals Module to test what value the counter is at. In fact we are going to need eight of these to test for our eight values.

Type 'Int Equals' into the toolbox search bar but this time click on the module in the toolbox 8 times in succession. You will see a small number increment on the module. This means that now when you drag the component in to your project it will create eight instances of that component in one go.

In order to keep your schematic window clean and structured we are going to make our first module. What this means is that we are going to package up some parts into a nice clean single module component.

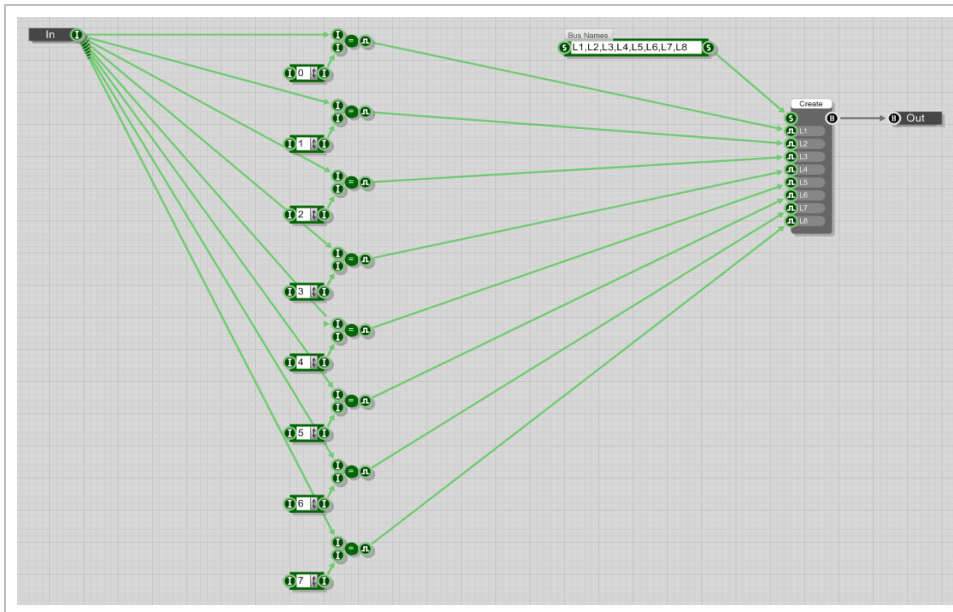
To do this, using the left mouse click and drag to select all eight of our Int Equals components. Now Click on the Make Module button on the action panel at the bottom of the selection rectangle (it's the middle button as shown opposite). This will now create a single module containing your components.

Double Click on this module and you will step inside and see your Int Equals components. You can step out again by double clicking on free space on the schematic window, or right clicking and selecting 'Move to Parent' or by clicking on the parent module in the Navigator.



So we now need to add some inputs and outputs to our module. Whilst still inside your module search for 'Input' (or press 'I' on the Keyboard for the short cut) and add a single Input component. Repeat for the Output (or press 'O' to use the Keyboard shortcut). Connect the Input component to the top Int input on each of the Int Equals components and add an Int box to each of the other inputs.

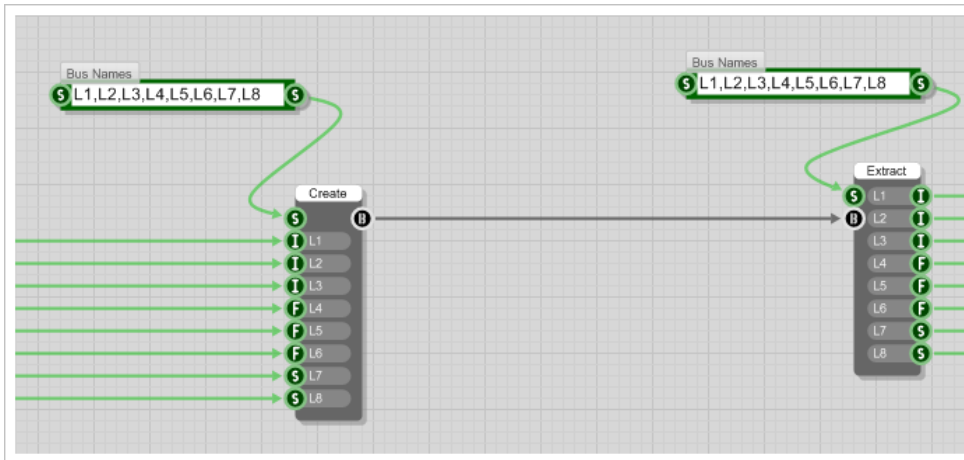
Set the values of these Int Boxes to 0,1,2,3,4,5,6,7 to detect our eight different values from the counter. The output of the Int Equals component is a Boolean (Bool) value (0 or 1, or True or False). We now have eight separate output lines that we can use to control the LEDs.



Count value split into 8 pulses then added to a bus

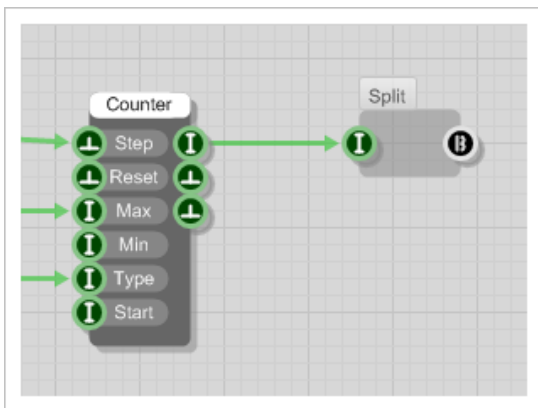
In order to simplify our design further we are now going to use our first Bus. Busses in FlowStone are single links that contain multiple links inside them. These can be created and extracted using the 'Bus Create' and 'Bus Extract' Components. For our project we need to add a Bus Create inside our current new module.

To create the eight inputs we need to add a Sting component ('S' Keyboard Shortcut) and edit the string value to represent our eight inputs, in this case we have used L1,L2,L3,L4,L5,L6,L7,L8 to represent the eight LED lines. Connect this string to the string input on the Create Bus component, and the eight inputs will appear on the component. Connect the eight Bool outs from the Int Equals components to these eight inputs. Finally we need to add an 'Output' component (Shortcut 'o') to give our module an output. Connect this to the Bus output of the Bus Create.



Example of a Bus (many wires in one)

Step back out of the module and you will see the simple module you have just created. Connect the Int out of the Counter to the Int In of our new Module and we are done.



Our first Module 'Split'

Step 4 – Creating the LEDs

(Tutorial4.fsm)

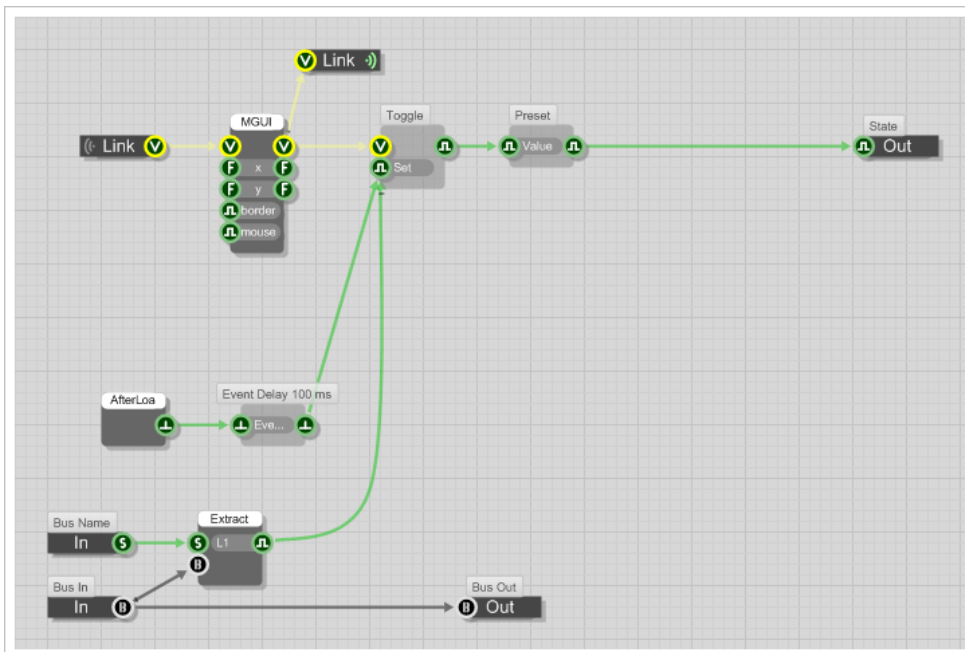
To create the LEDs start by dragging in an LED Red Module.

Now double click and enter inside the LED module. What we need to do is add some extra inputs to allow us to connect to our BUS. Firstly add a Bus Extract Module and connect an Input to the String input, you can name this 'Bus Name' by clicking on the 'N' button once the component is selected.

Now add another Input and call this 'Bus In' and an output called 'Bus Out'. Connect the Bus In to the Bus Out. Now step back out of this module and make a String called 'L1' and connect this to the string Input of your LED module. Go back inside the LED Module and see that now that your Bus Extract has a single output called 'L1'. Connect this to the Set input on the Toggle Module and you can delete the other unused Input also connected to the Set Input if you like.



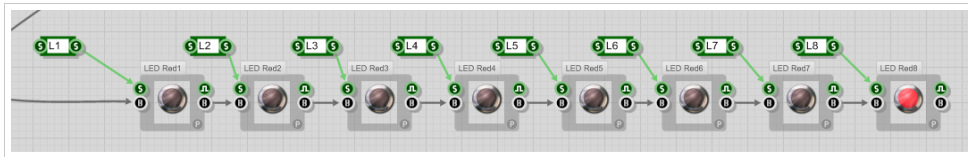
LED Red Module



Our Modified LED Module

So we now have a single LED with a String Input called L1. Connect the Bus from the splitter Module to the Bus input of the LED. In order to make 7 more of these, select the LED module and the String by dragging with the mouse. Now right Click and choose Copy. Then right click on some free space and choose Paste Synchronized, this will make an identical copy of the Module that is Synchronized i.e. any changes to any of the synchronized modules are reflected in all of the Synced Modules.

Use this method of Synchronizing modules if you are going to have multiples of the same module that you might edit in the future, so that you only have to edit one module to change them all. Now edit the names of each LED module to be LED Red1, LED Red2 – LED Red8 etc. and connect the bus input to the outputs so that they are all connected. You should now see all of the eight LEDs flashing in sequence!



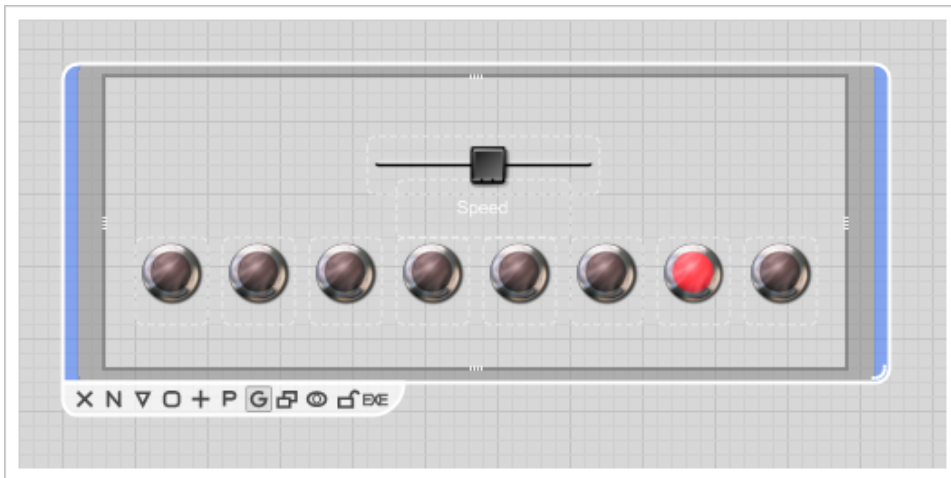
Our 8 Synchronized LEDs

Step 5 – Arranging the GUI

(Tutorial5.fsm)

Now our design is fundamentally working we can turn our attentions to the GUI (Graphical User Interface). The first step is to make a module out of our whole design. To do this select the entire design (Ctrl + A) and then click on the Make Module button on the action panel at the bottom left of the selection rectangle.

You now have a very small module with nothing showing. To make the graphical parts visible select the module and then click on the 'G' button at the bottom to enable the module GUI. Now if you pull the bottom right corner of the module to enlarge the size you will see some of the graphical components that we put inside. Since we copied all of the LEDs they are actually all on top of each other.



Our basic GUI – Note 'G' is selected and the padlock is open

To move the graphical components around you need to unlock the graphics. This is done by clicking on the lock symbol again on the action panel at the bottom of the selection. Once unlocked you can drag the components around. This is where the naming of the eight LEDs comes in handy as they all look the same so you can use the names to identify them to put them in sequence.

Step 6 – Finalizing the GUI

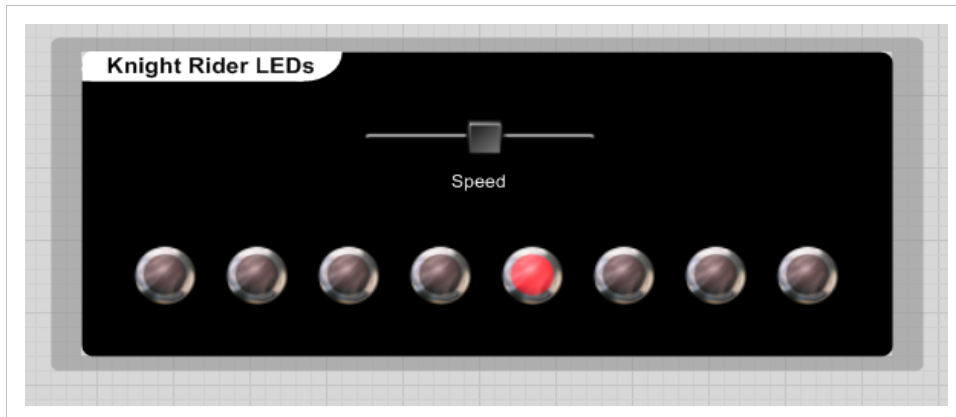
(TutorialFinal.fsm)

Now we have a basic GUI, we can add some final details to enhance the appearance and give it a custom feel. To do this we're going to add the 'Module GUI' (MGUI) component inside our top level module. This does exactly the same thing as the 'G' button that we used previously in that it gives the module a GUI. However, by using a component we get access to the drawing surface (provided by the View connectors) so that we can apply low-level drawing functions.

We'll paint the background black. Add an 'Area' component and a 'Filled Round Rectangle' (FRRect). Connect the 'V' (View) out of the MGUI to the 'V' in of the FRRect and also to the 'V' in of the 'Area' component. This calculates the area of the module GUI. Next connect the 'A' (area) out of the 'Area' module to the 'A' in on the FRRect component.

We now need to give our background a Colour, to do this we can use the 'Colour' component which has a colour palette to select or make custom colours. Connect the 'C' (Colour) out of the colour component to the 'C' in on the FRRect. We can also specify a rounded corner for our background, so add a 'Float' input to the Corner input and specify a value of 1.5.

Finally we need to add our first Wireless Link to filter down the graphical hierarchy to the other graphical components (using the 'G' button method previously this was automatically created behind the scenes). To do this we add a "Wireless Output" component and connect it to the 'V' output of the MGUI component. If you want to use wireless links, they simply connect points together with the same name and type running down the module hierarchy always. We also added a graphical label for fun that you can use in your own projects.



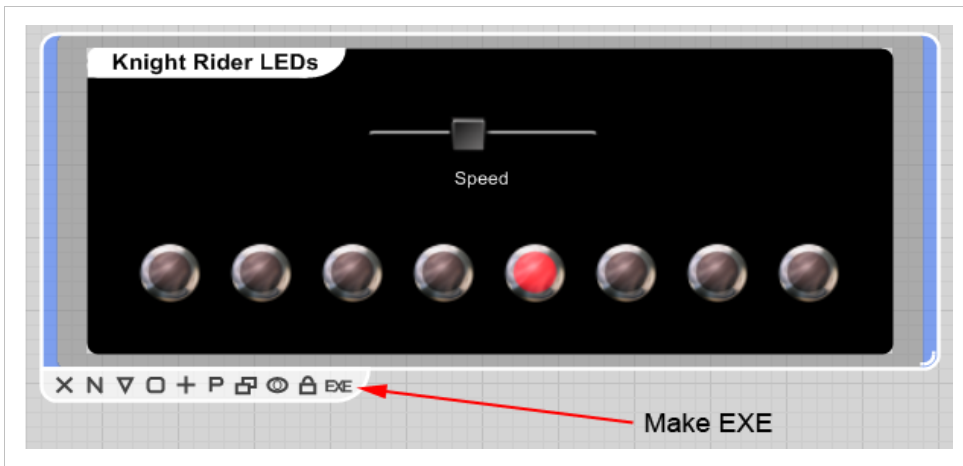
Our finished GUI

Step 7 – Make an Executable App

(Led.exe)

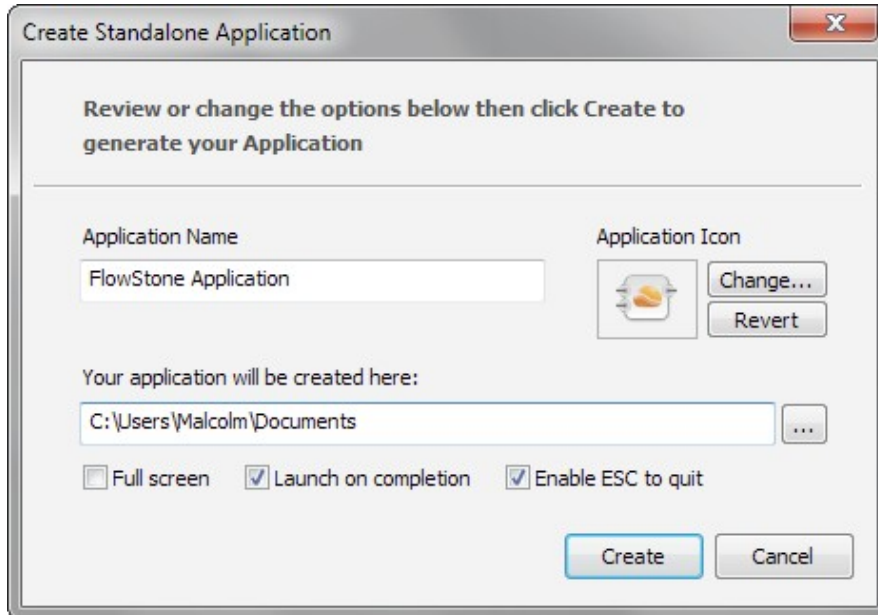
Once you are happy with your program and the GUI is finished you can turn your work into an executable application (app). These are called EXE files and are the heart of any PC system. EXE files are programs that can be run as standalone applications. They can be distributed freely without any reference to FlowStone – they can even be sold!

To make your first EXE all you need to do is select your top level module click and click on the Make EXE button on the action panel at the bottom of the selection.



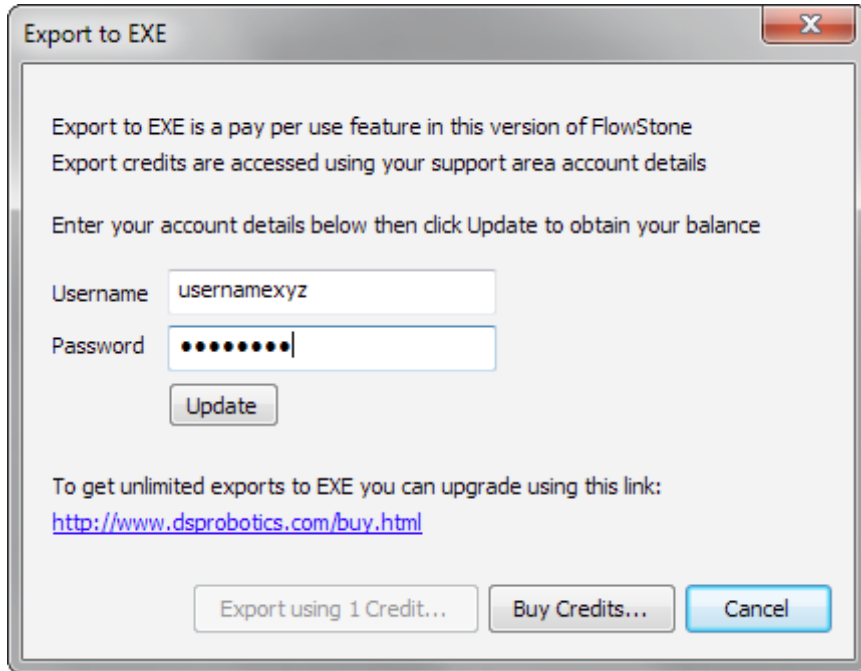
Click the make EXE button

The Create Standalone Application Windows opens and you can name your application, LED say. You can also select a different Icon for your app and choose it's destination folder etc.



Create Standalone Application Window

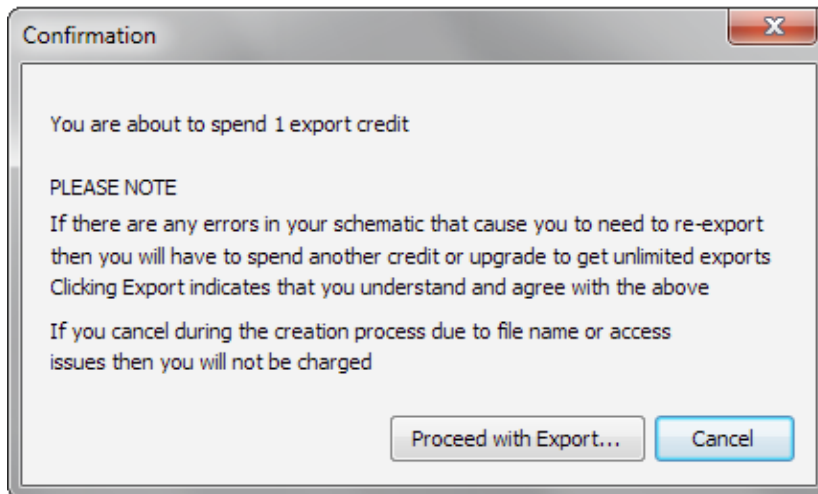
Now Click Create; in FlowStone Enterprise or FlowStone Pro you can make as many EXE's as you like, but in FlowStone FREE there is a restriction. You can still make EXE's from FlowStone FREE but you need to use Credits to do this. With every download of FlowStone FREE we add a couple of Credits to your online account so you can play with this tutorial. If you enter your Username and Password and click Update you will be able to proceed using these free credits.



Export using Pay As You Go Credits

Further Credits can be purchased by using our Pay As You Go (PAYG) system, by clicking on the Buy Credits Button.

To complete making your EXE click on the 'Export Using 1 Credit' button then proceed with export etc.

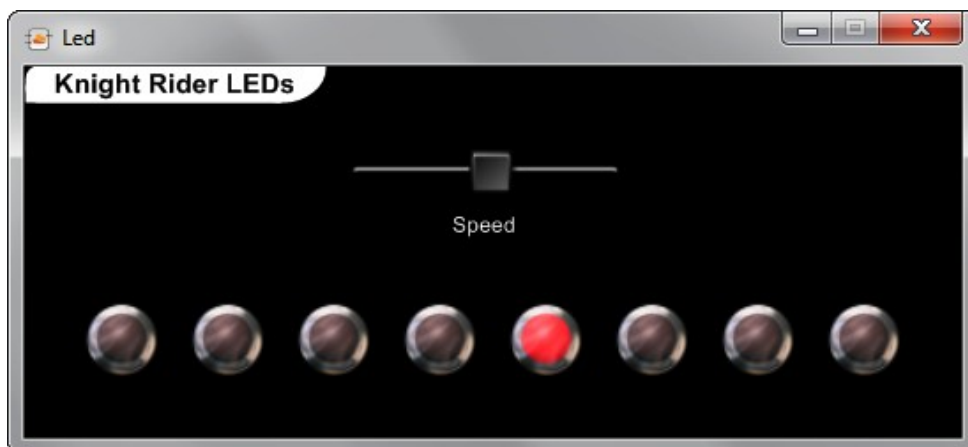


Proceed with Export

That's it, you have created your first application using FlowStone, check in the folder where you exported your EXE file and you will see 'Led.exe' double click on this and the program will run!



The EXE file with your icon can be seen in the Windows Explorer



The running standalone file!

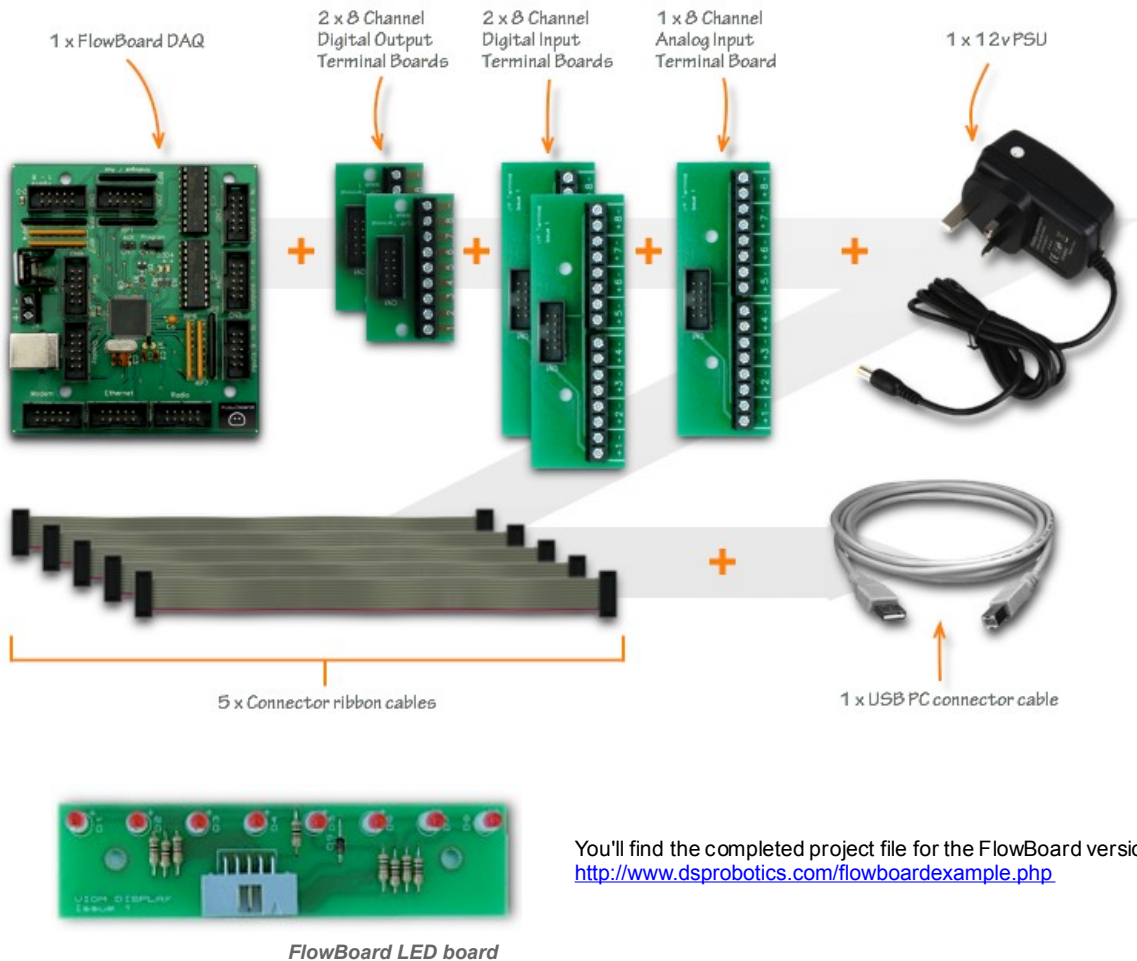
Congratulations you have completed the FlowStone Tutorial!

Project Expansion

If you'd like to expand on this project , why not add a FlowBoard DAQ Pack with an LED board so that you can see the real LED's flashing in sync? The FlowBoard DAQ is a low cost Data Acquisition (DAQ) system developed by DSPRobotics specifically for the use with the FlowStone programming language.

For more information see:

<http://www.dsrobotics.com/flowboard.html>



You'll find the completed project file for the FlowBoard version here:
<http://www.dsrobotics.com/flowboardexample.php>

For more examples of FlowStone projects
that you can learn and copy from go to :

<http://www.dsrobotics.com/examplesarea.php>

or visit:

www.dsrobotics.com

