

User Manual

Some important terms

Sketch - A program created using the Engduino IDE

Module - A block of code representing some hardware on the board or some programming flow control

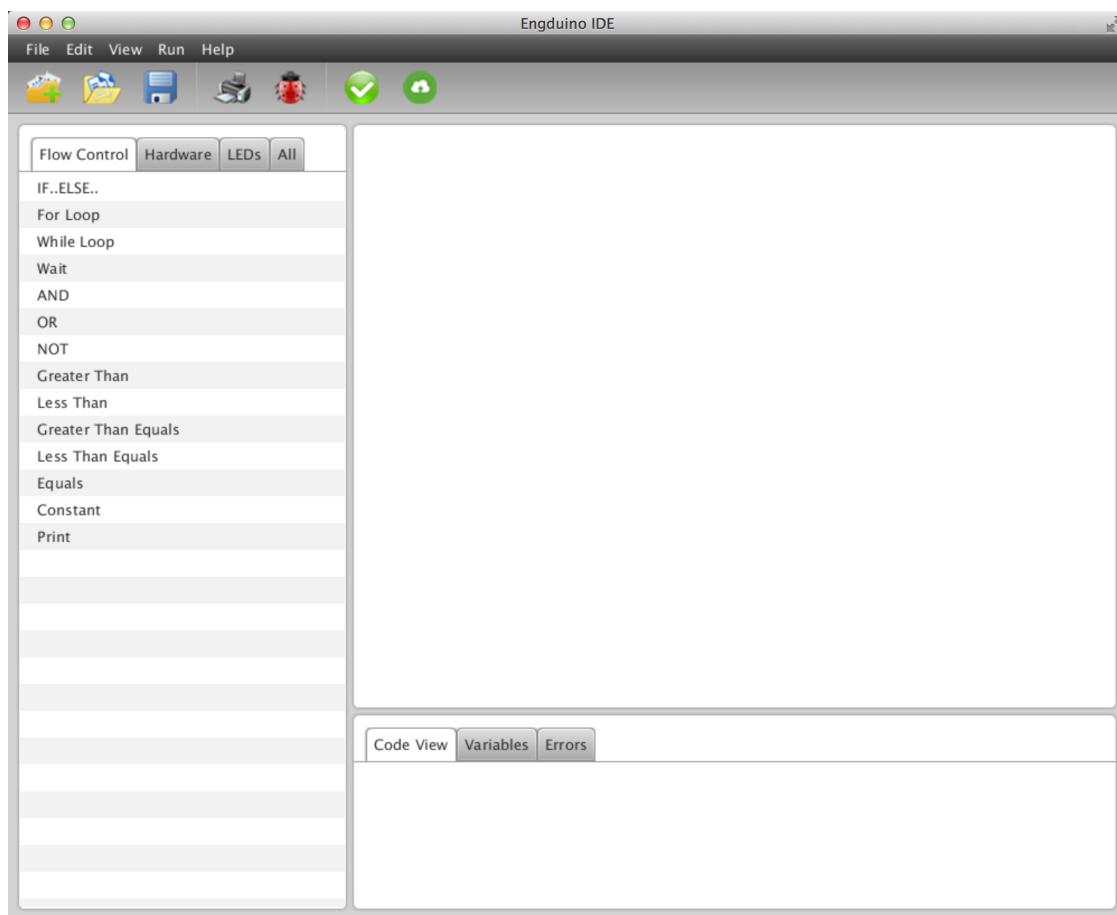
XML - A format in which the sketches are saved and parsed back to make sketches

Table of Contents :

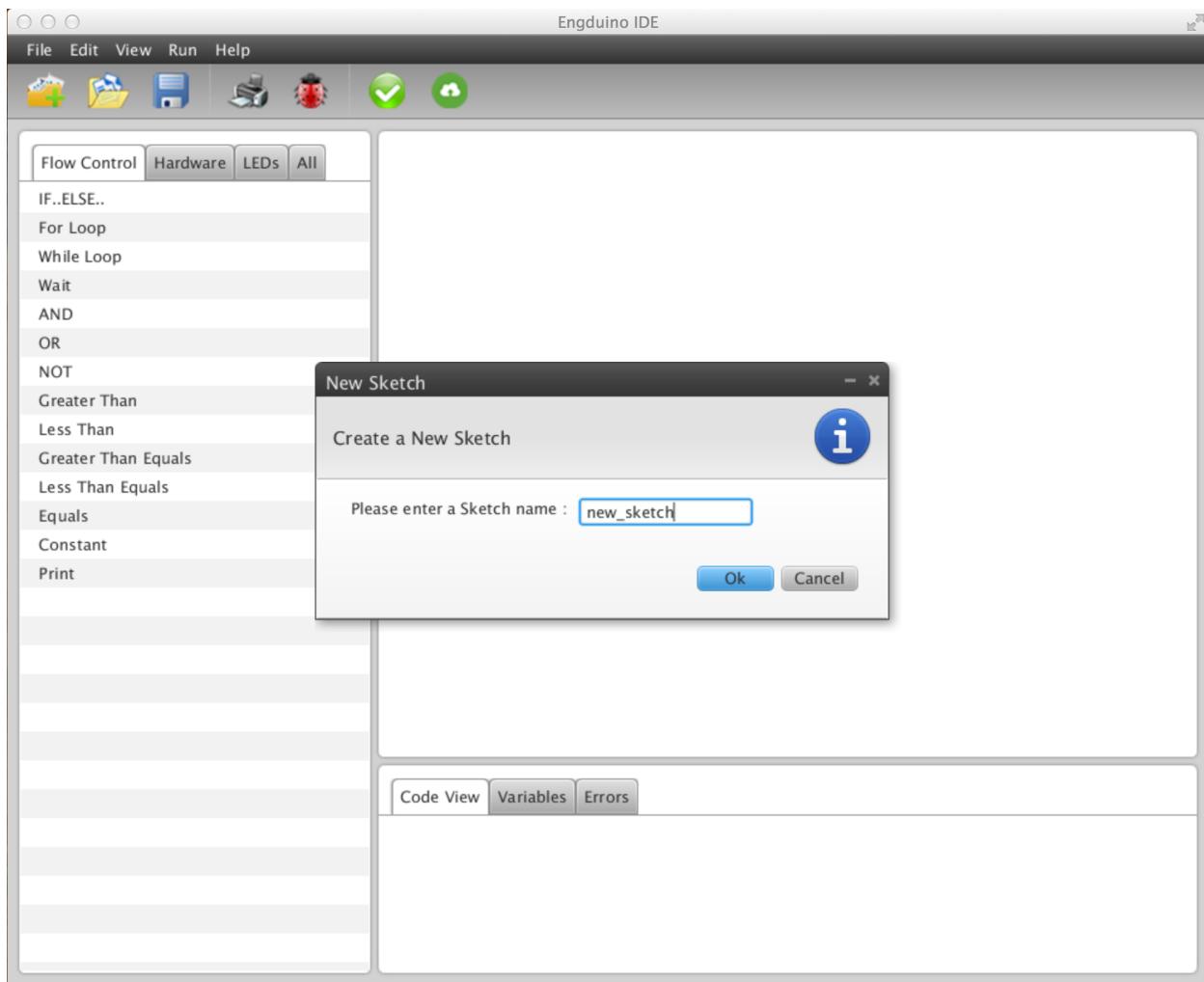
- How to create a Sketch ?
- How to create programs?
- How to save a sketch ?
- How to open a sketch ?
- How to save a sketch as module for reuse of code and creating larger program?
- How to connect modules of different type together ?

How to create a new Sketch ?

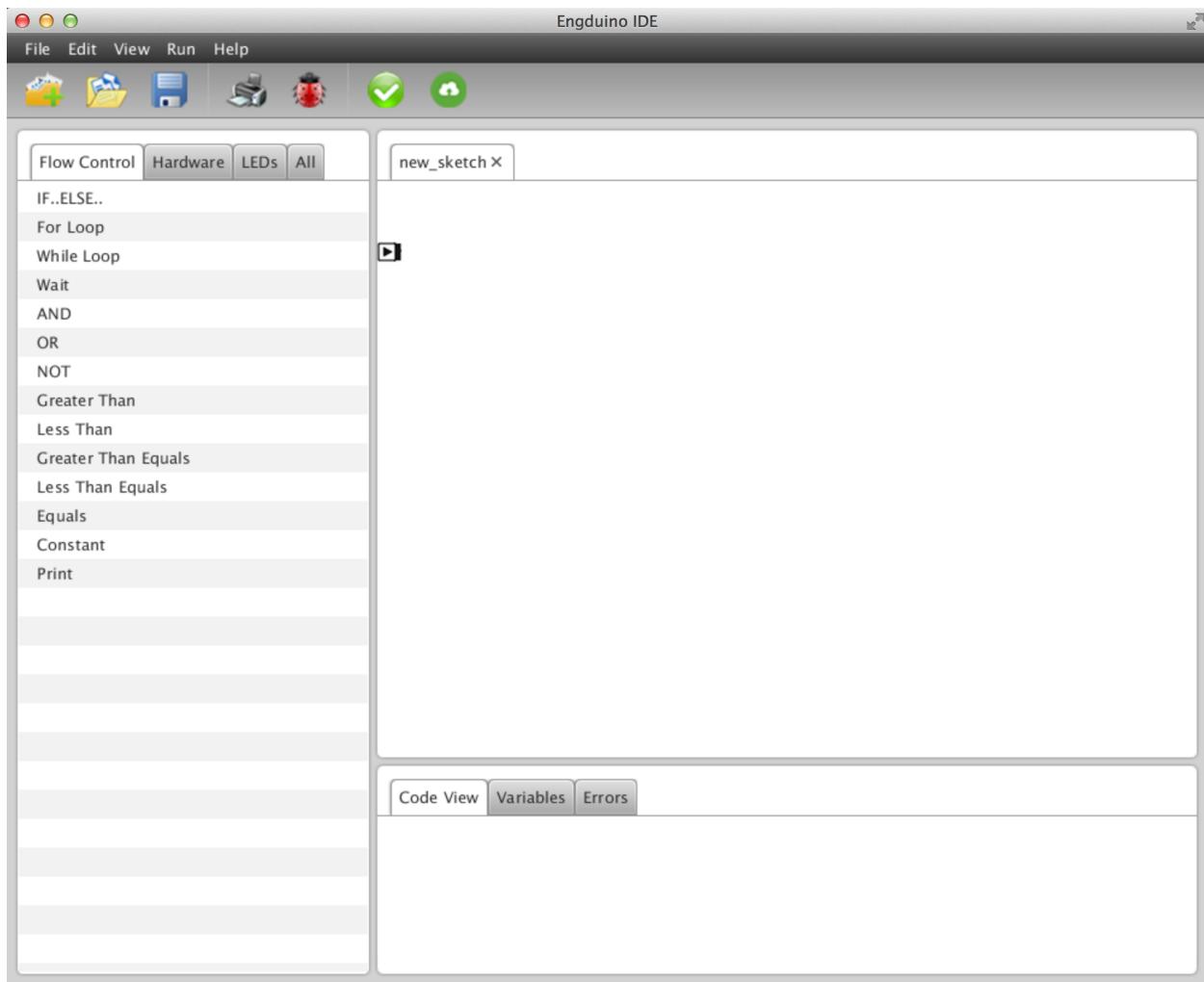
Click the new sketch button in the top left corner or select 'New Sketch' from the file menu item.



A dialog box will appear, enter the name of the sketch you want and click OK.

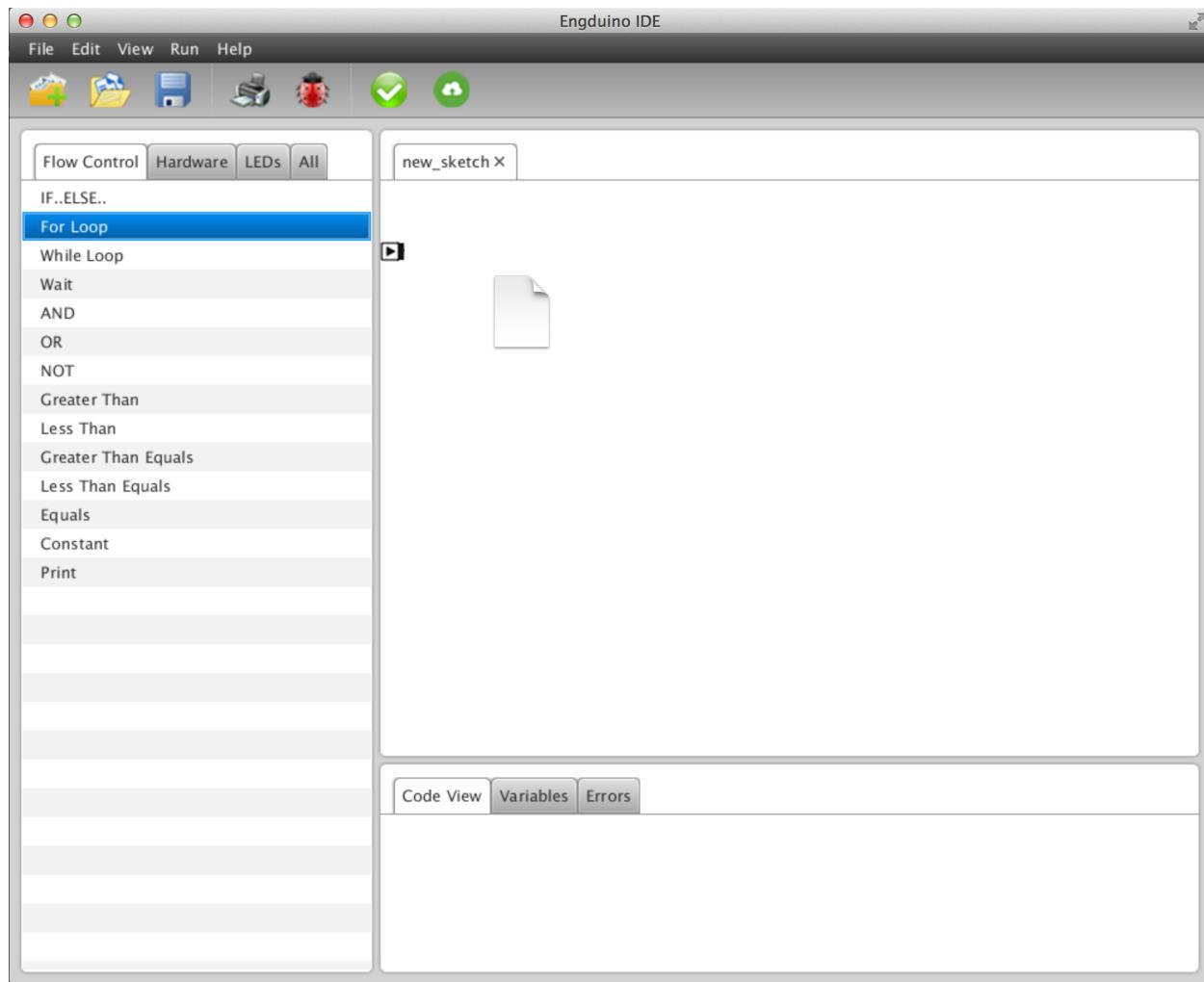


A blank sketch will be created for you write programs and play with.

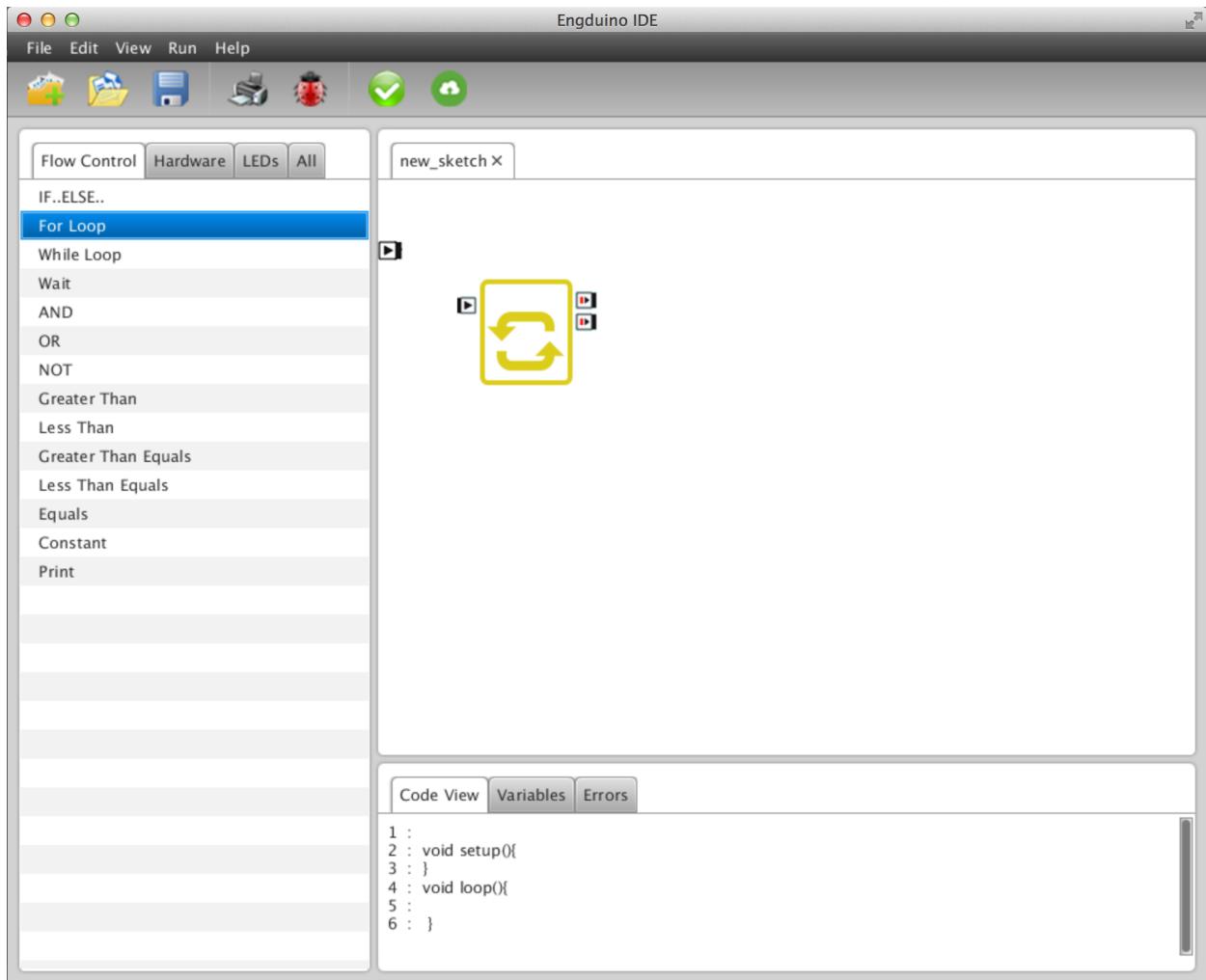


Adding Modules and Creating Programs

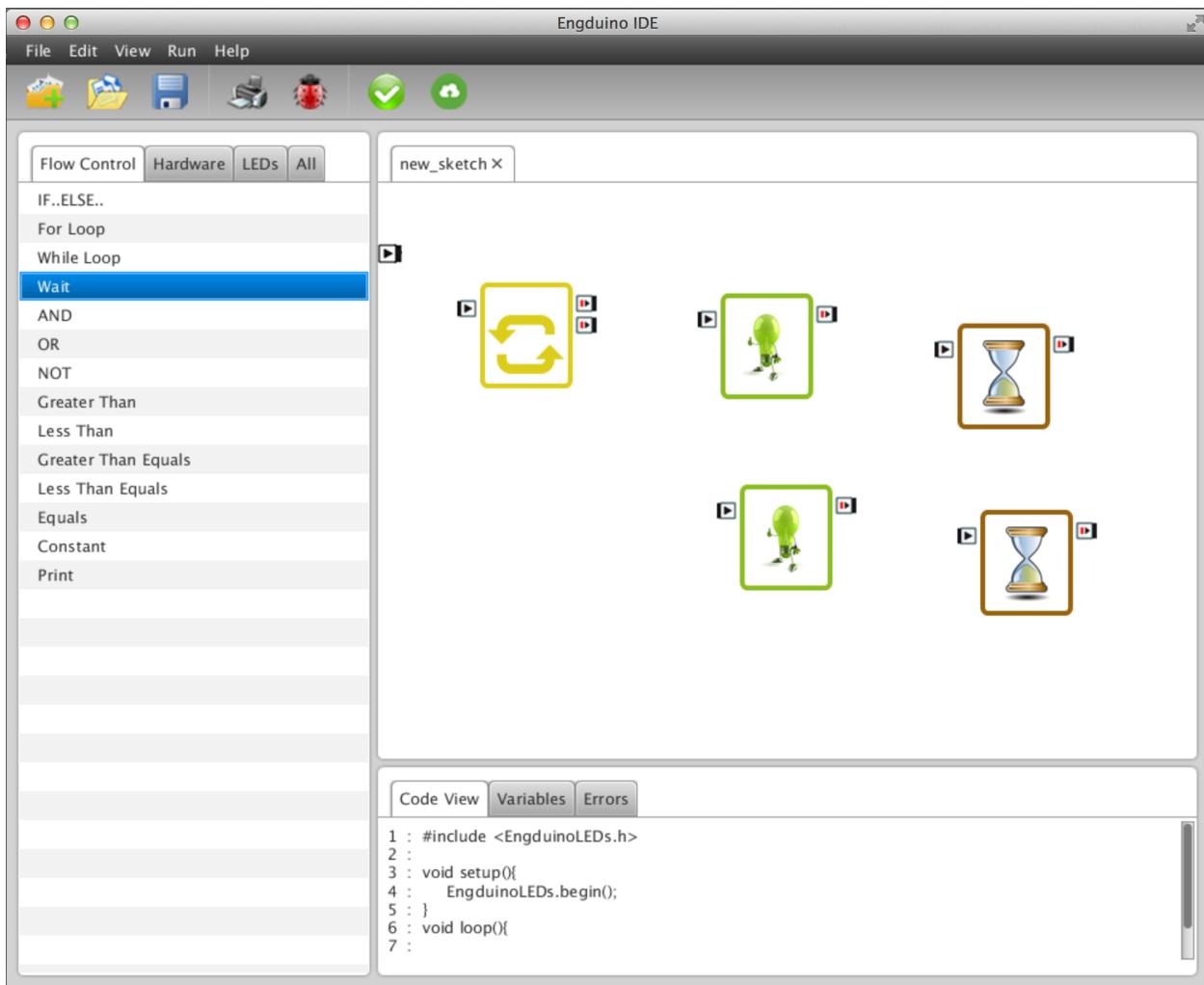
Step 1 : Drag an element in the sketch window until it changes to a file symbol



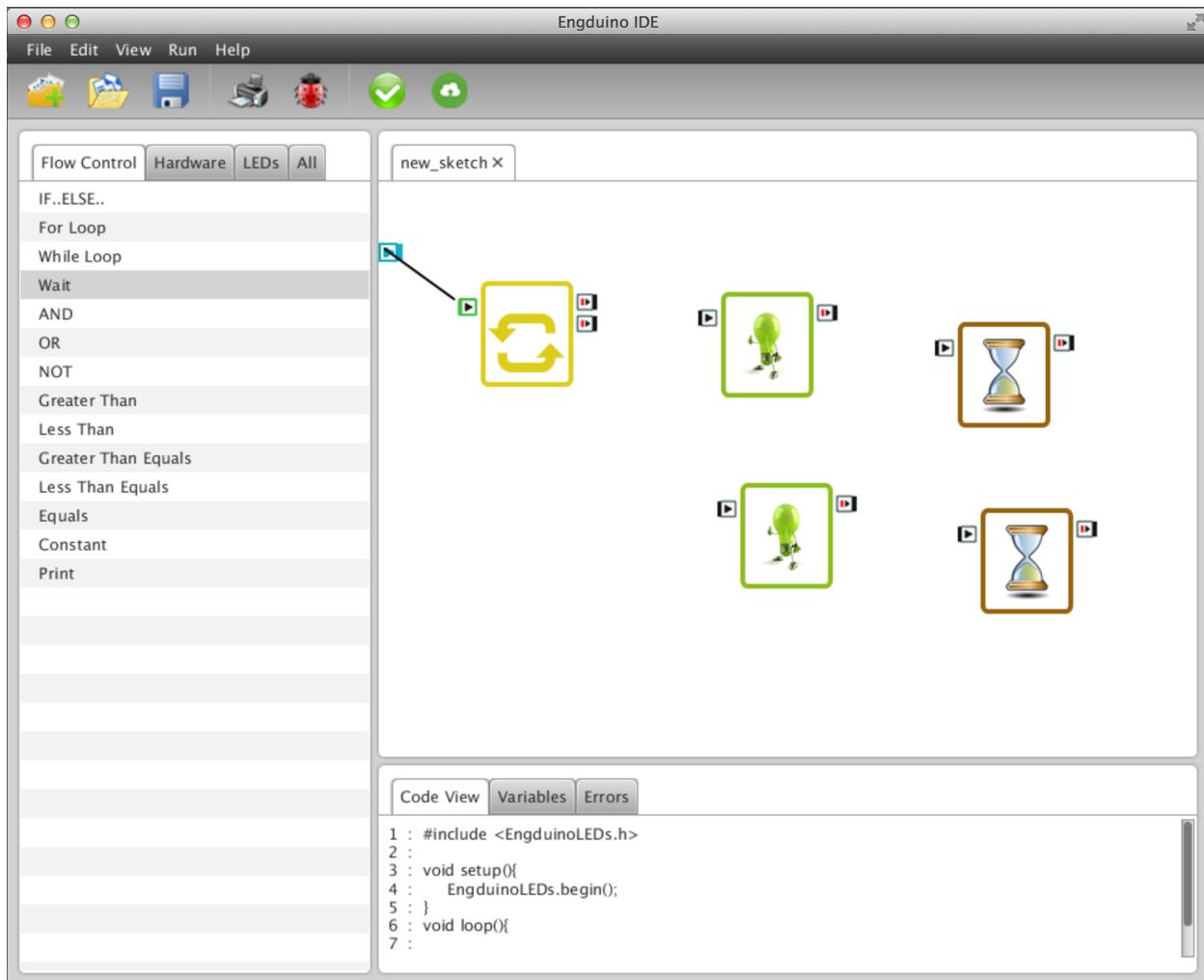
Step 2 : Drop the module where you want the module to be and icon for that module will appear



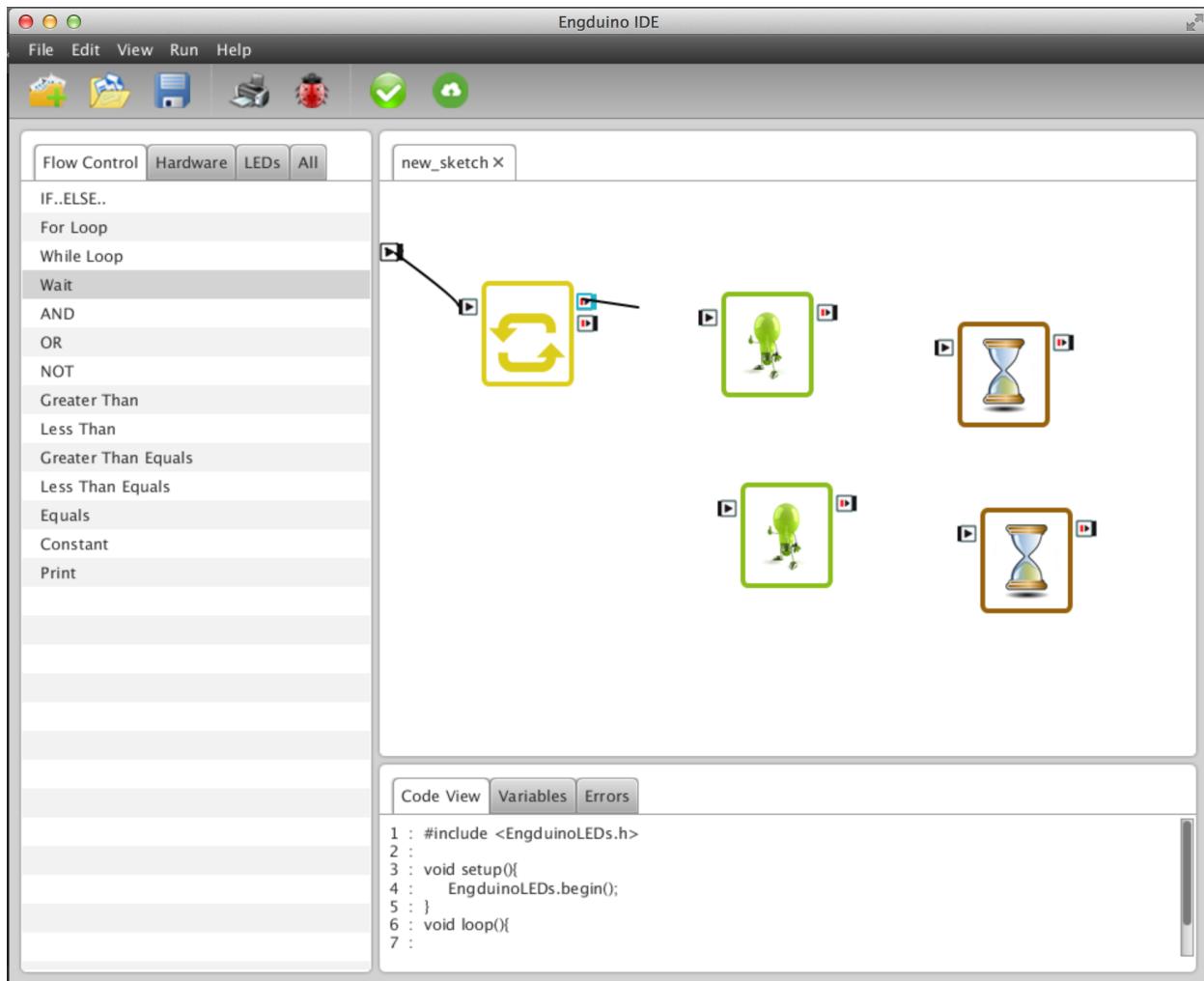
Step 3 : Repeat the process for all the modules you want to use in your sketch

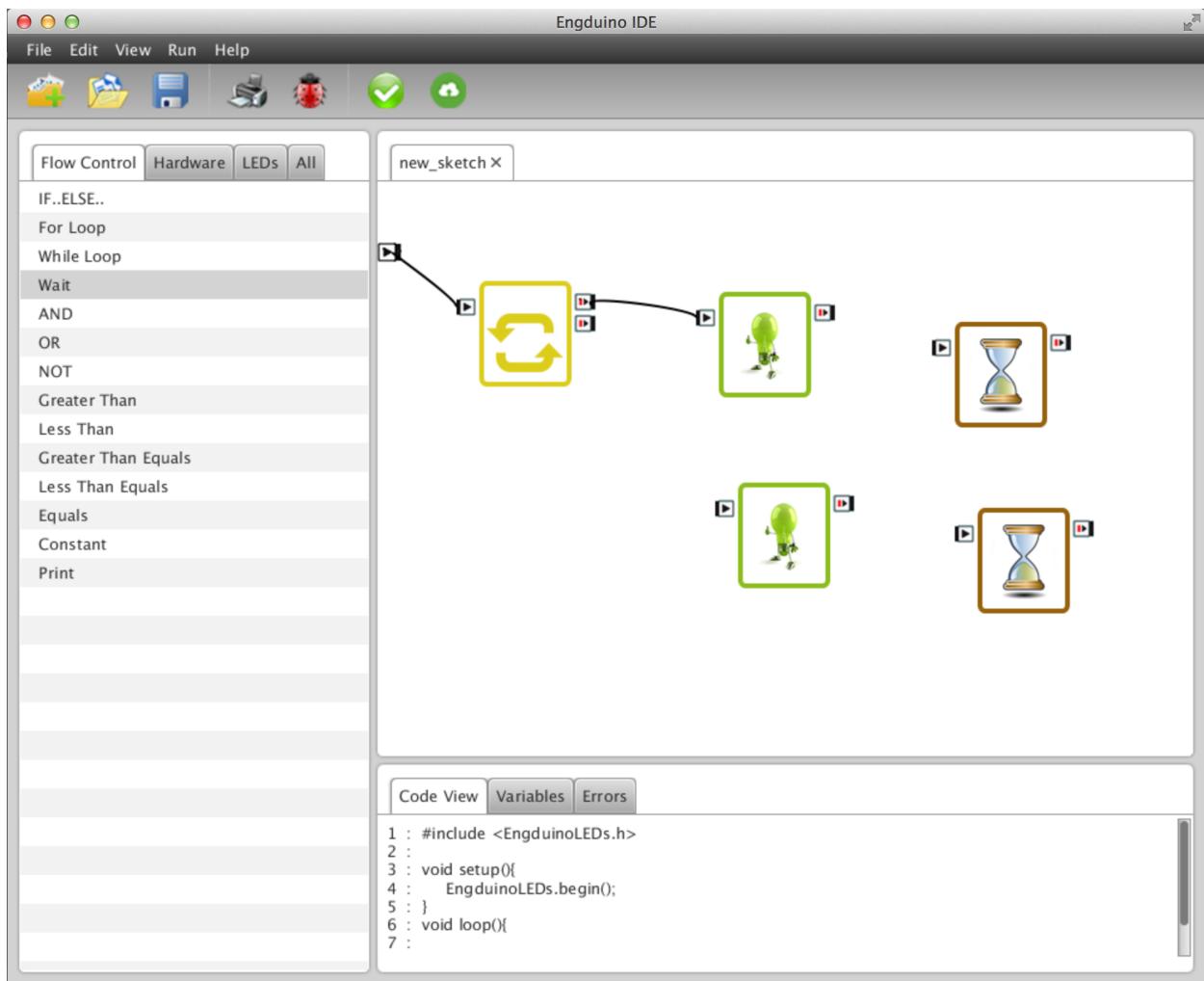


Step 4 : Connect the starting module with the main Input marker (colored blue) y dragging and dropping a connection wire to the input marker of the flow control. This will determine the start of your program

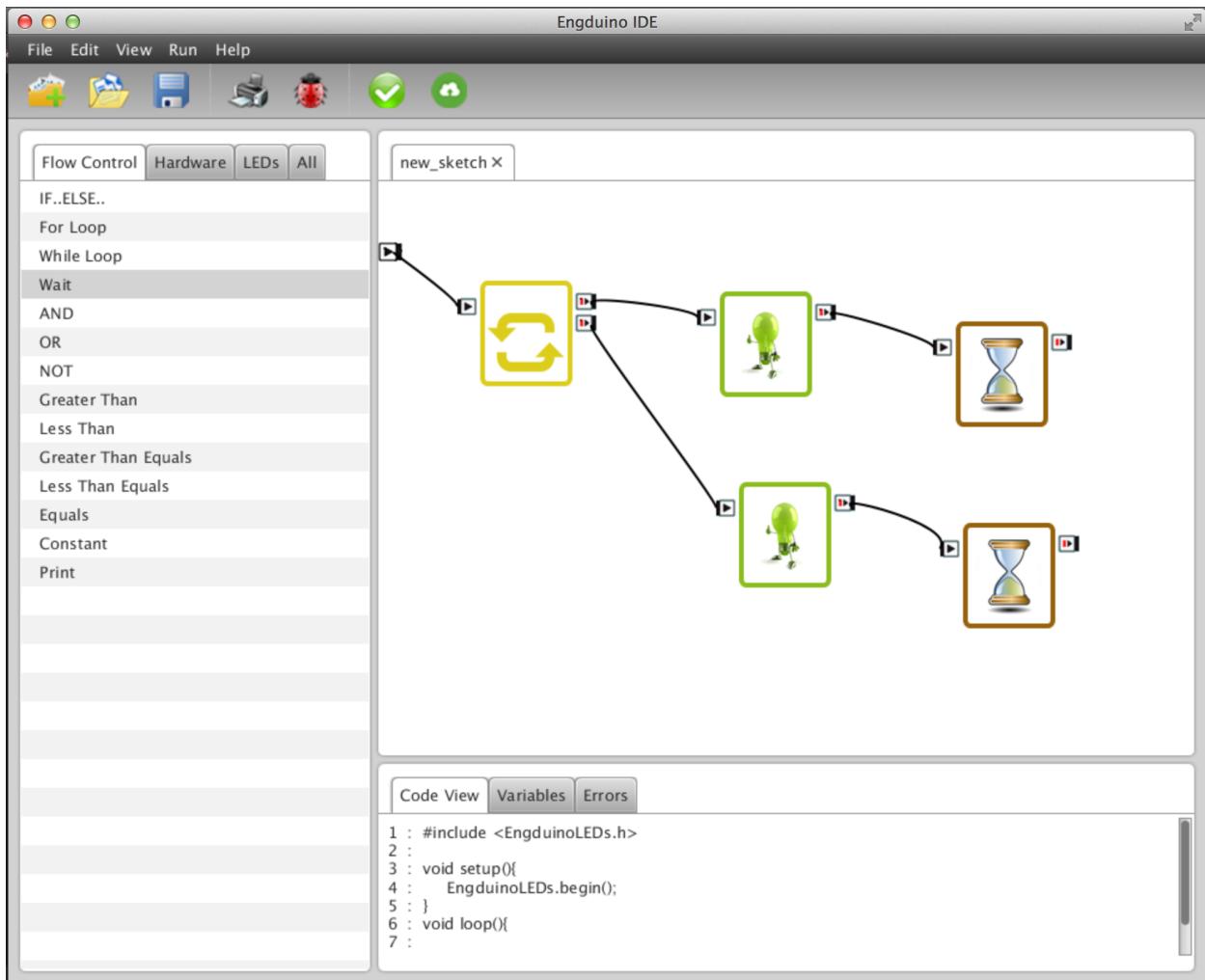


Step 5 : Connect other modules with the for loop by dragging and dropping a connection wire from the output marker of the for loop into the LED module. For more information on how to connect the modules please see the connecting module section below

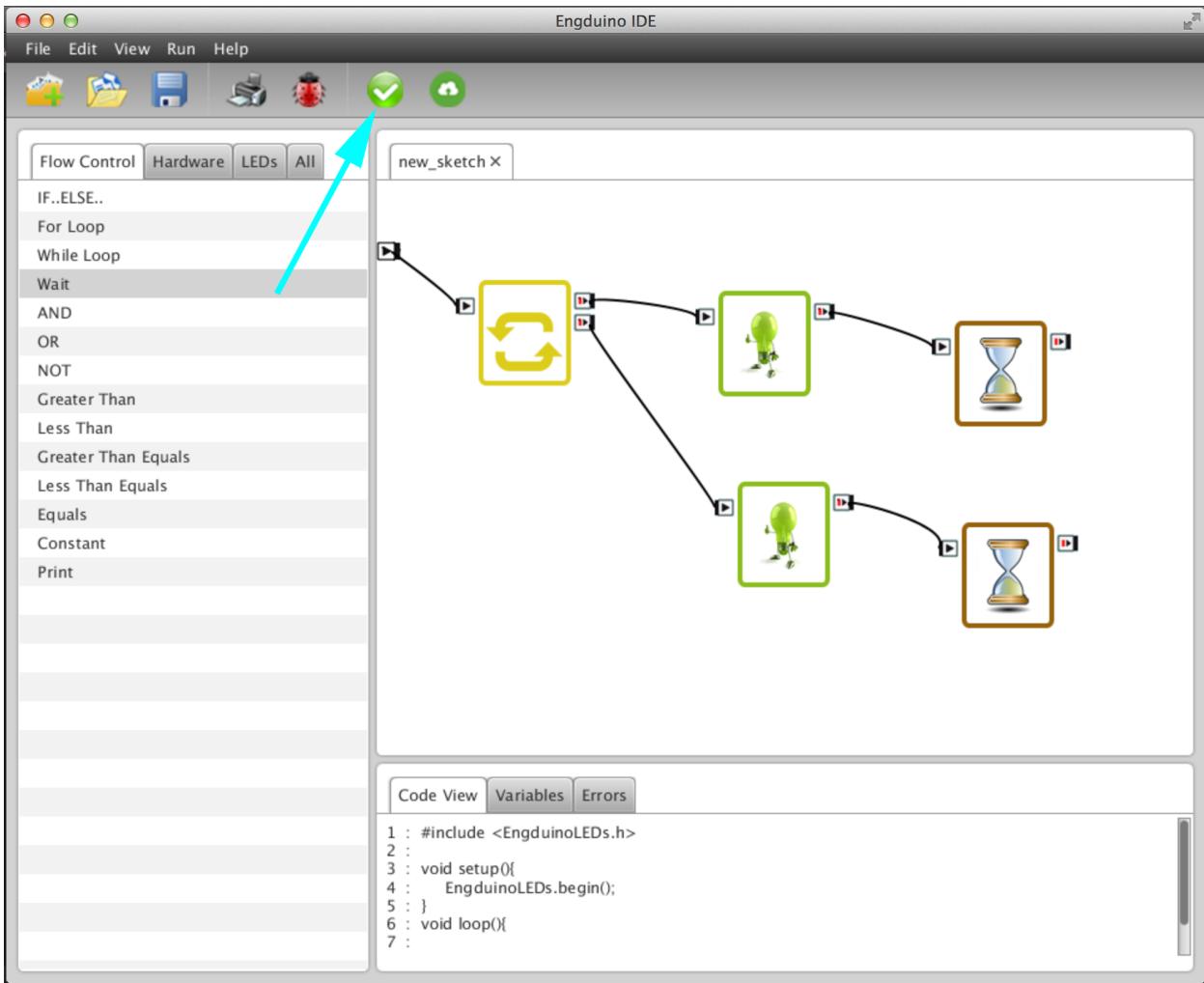




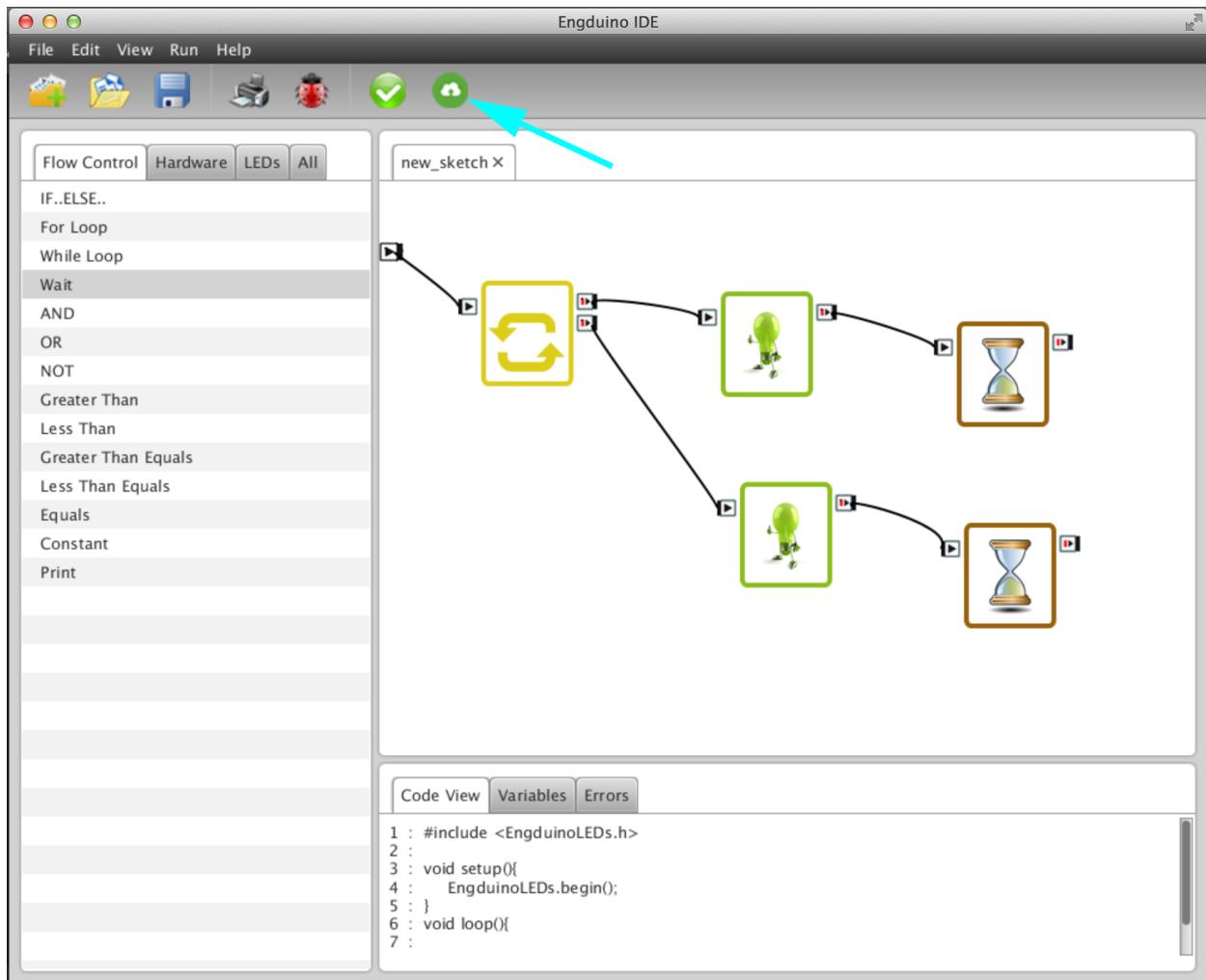
This is how your connections should look like after you have connected the modules together



Step 6 : Press the verify code button, this will generate the code for the connected modules and will be shown in the code view window at the bottom. You can check the code to see how the connections work to generate the code and to check whether the code generated for your particular sketch is correct or not

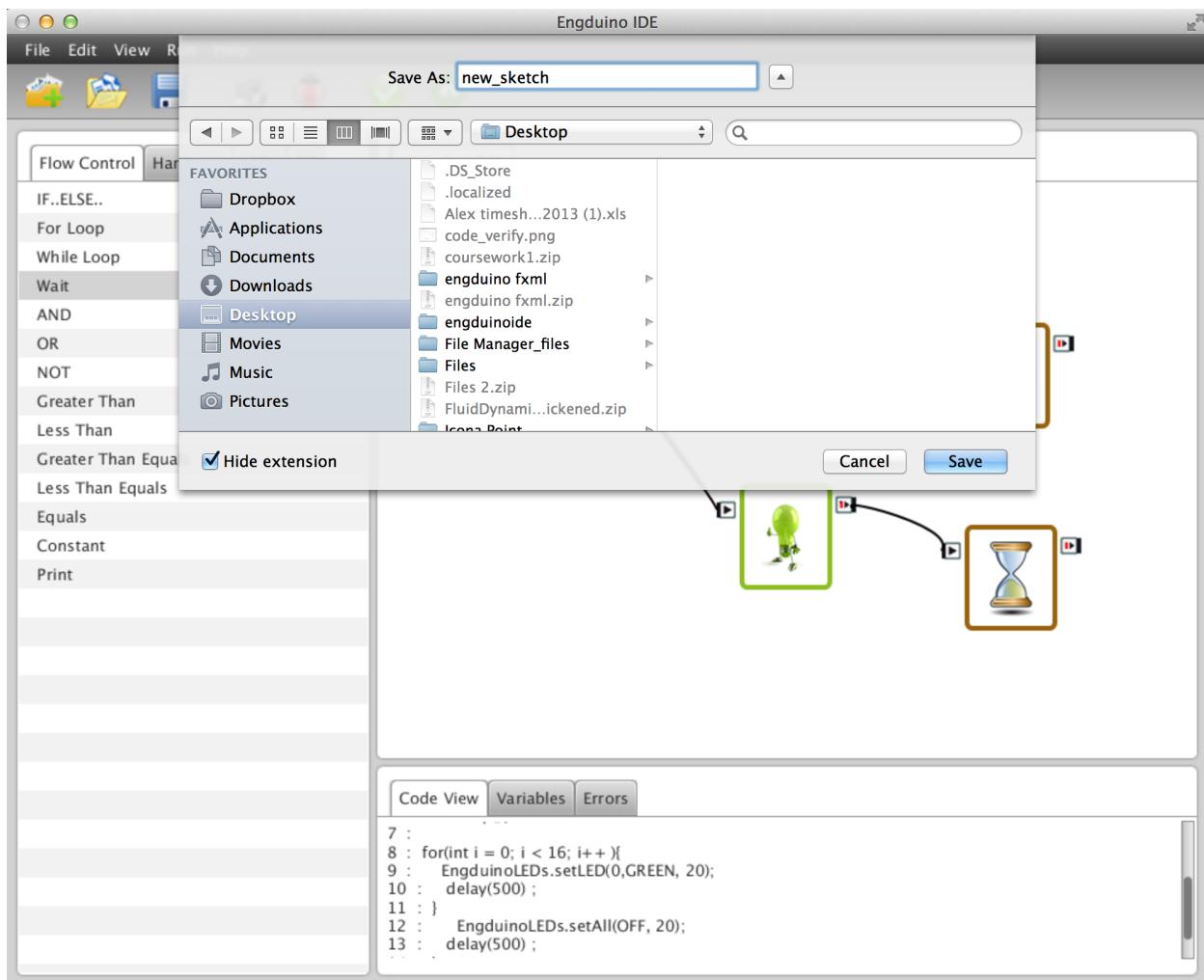


Step 7 : Now simply connect your engduino to the USB port and press the compile and upload button. This will compile your code and upload it to the engduino

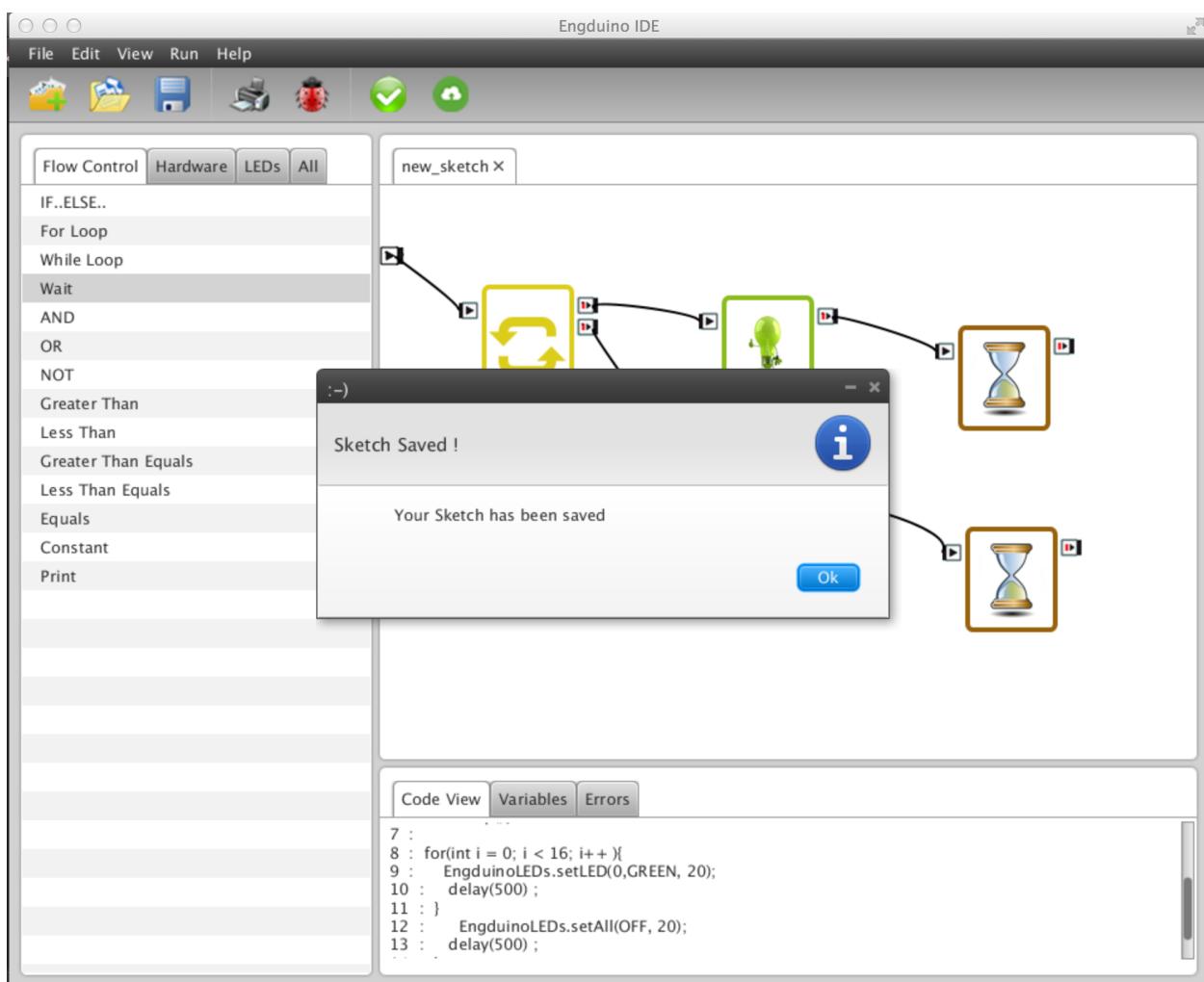


Saving a Sketch :

Step 1 : Select a sketch to save, click on the save bottom in the tools menu or select Save sketch from the file menu. A File location window will appear for you to select the destination of the saved file. Select the appropriate directory and click save button.

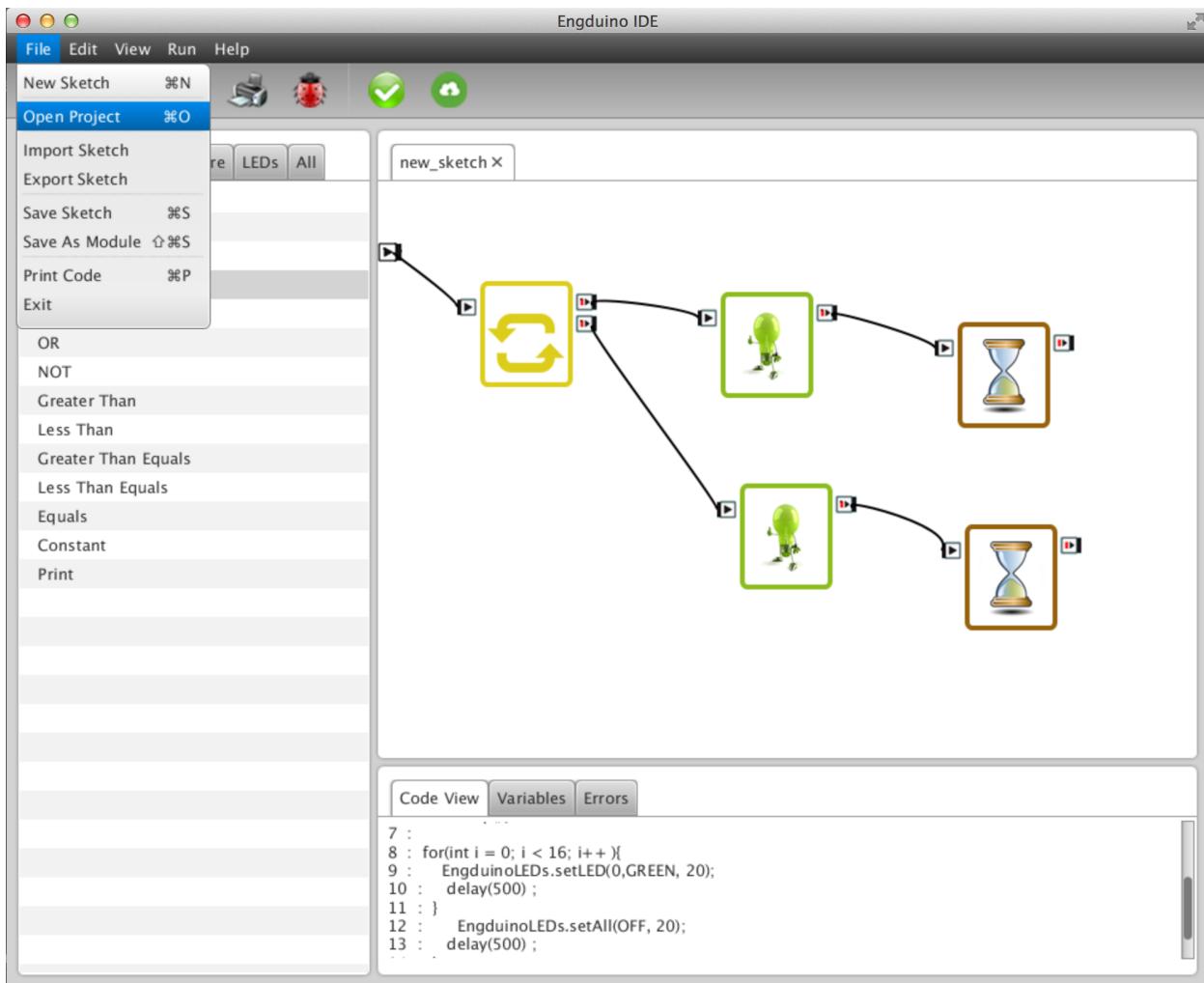


Step 2 : You will get a message that your sketch has been saved successfully.

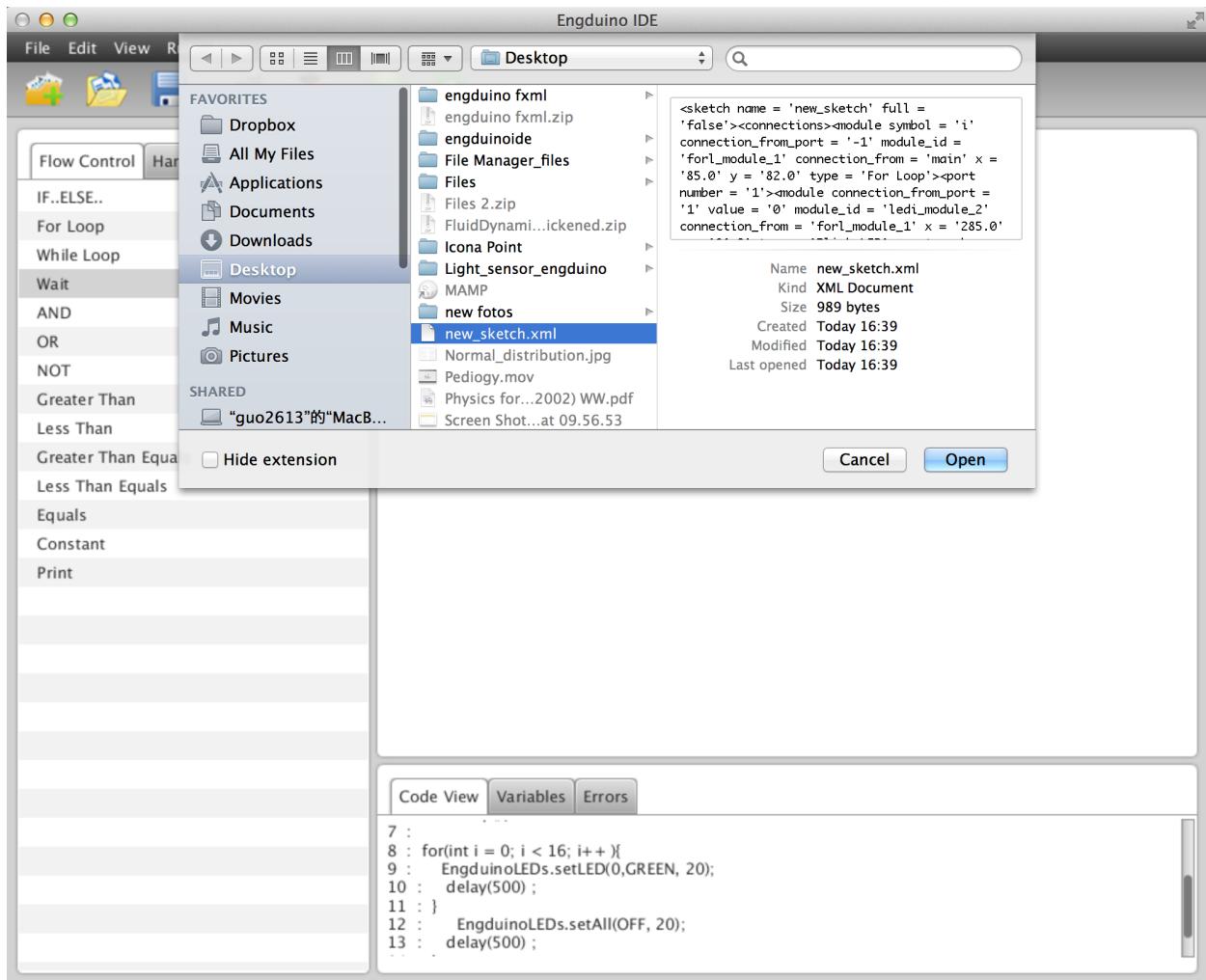


Opening a Sketch :

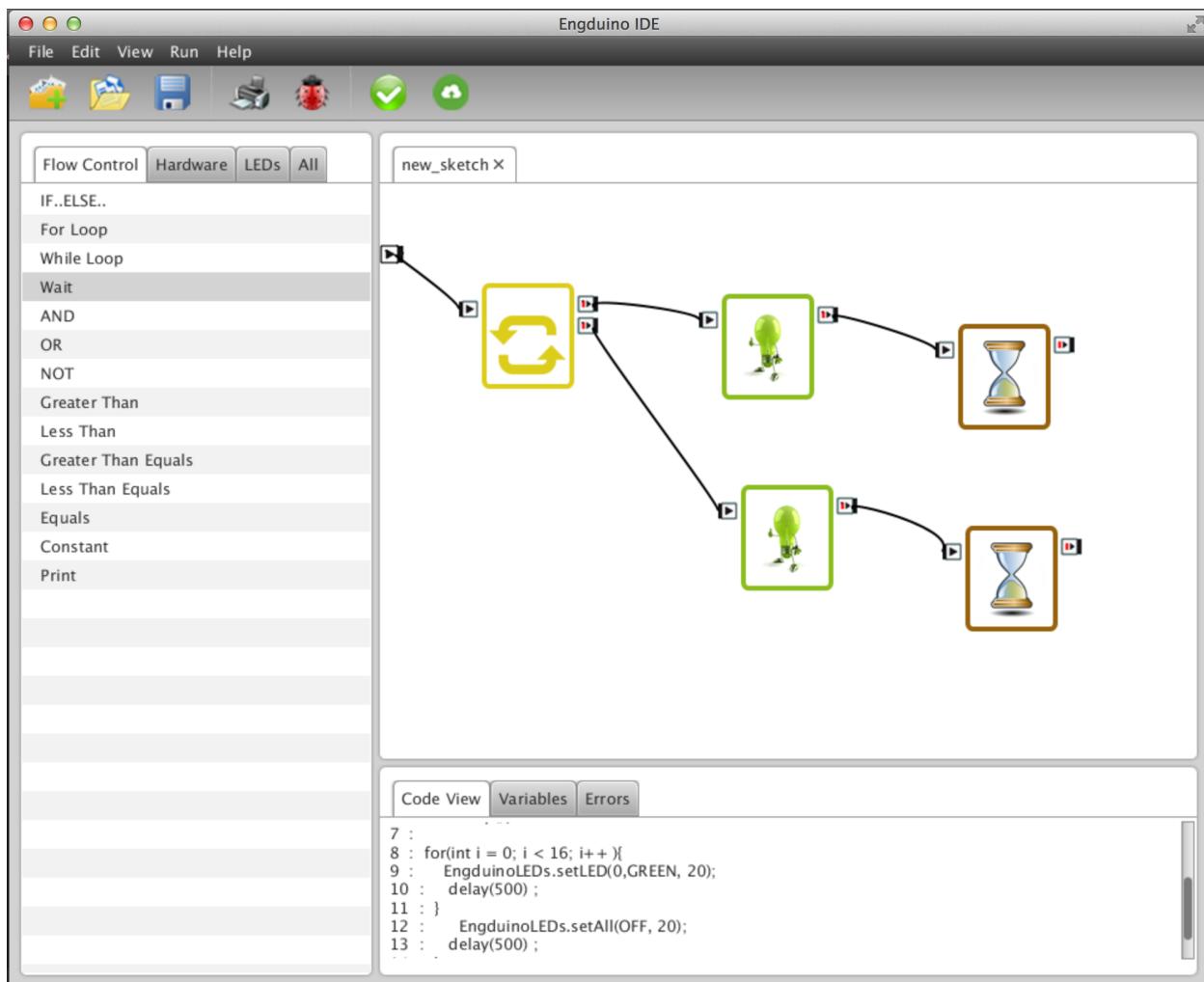
Step 1 : Select open from the File menu or click the open button on the tools menu.



Step 2 : A file chooser window will appear for you to locate and select the file. Once you have located the file , click the open button.



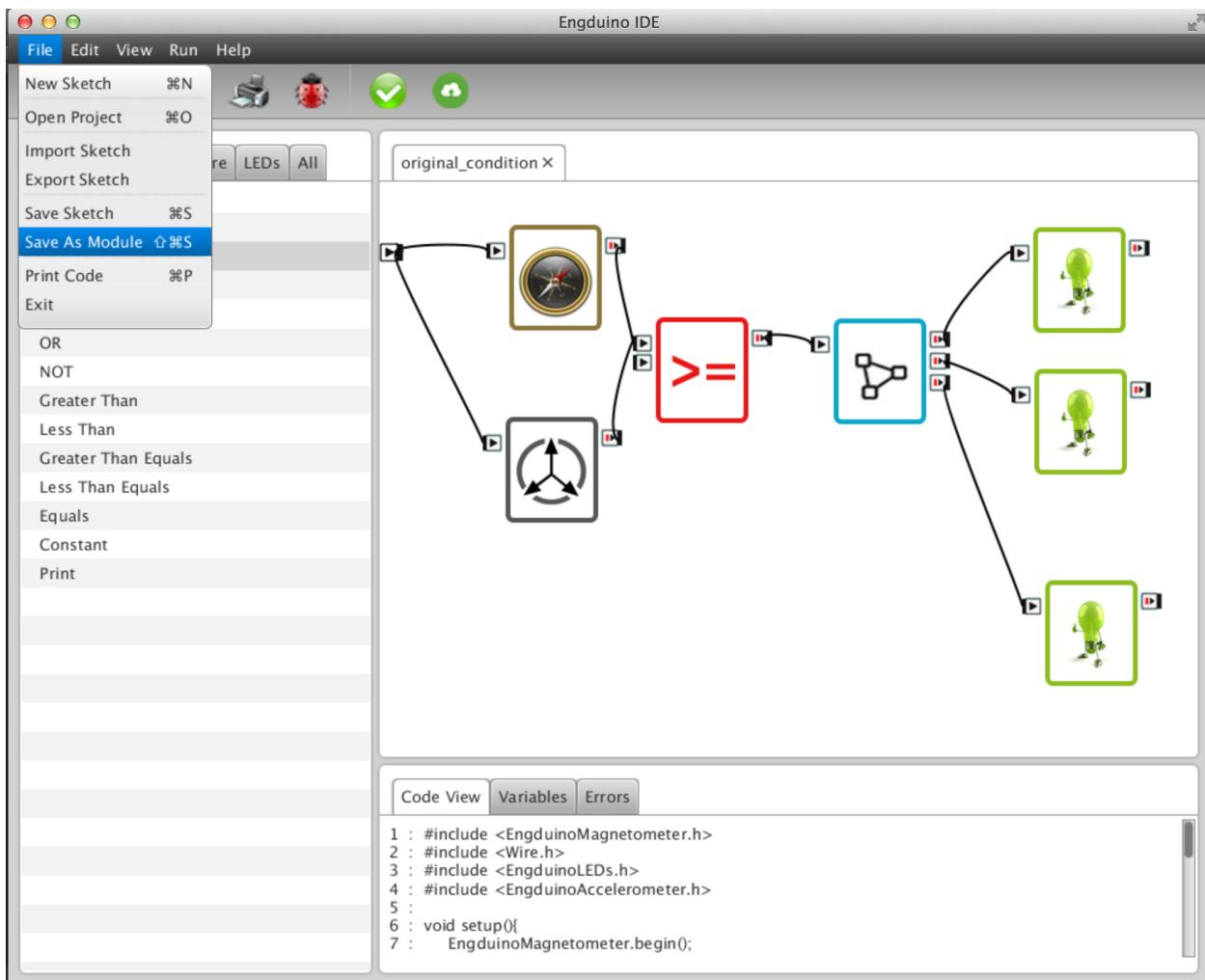
Step 3 : Your sketch will be parsed and all the module and their connections will be created.



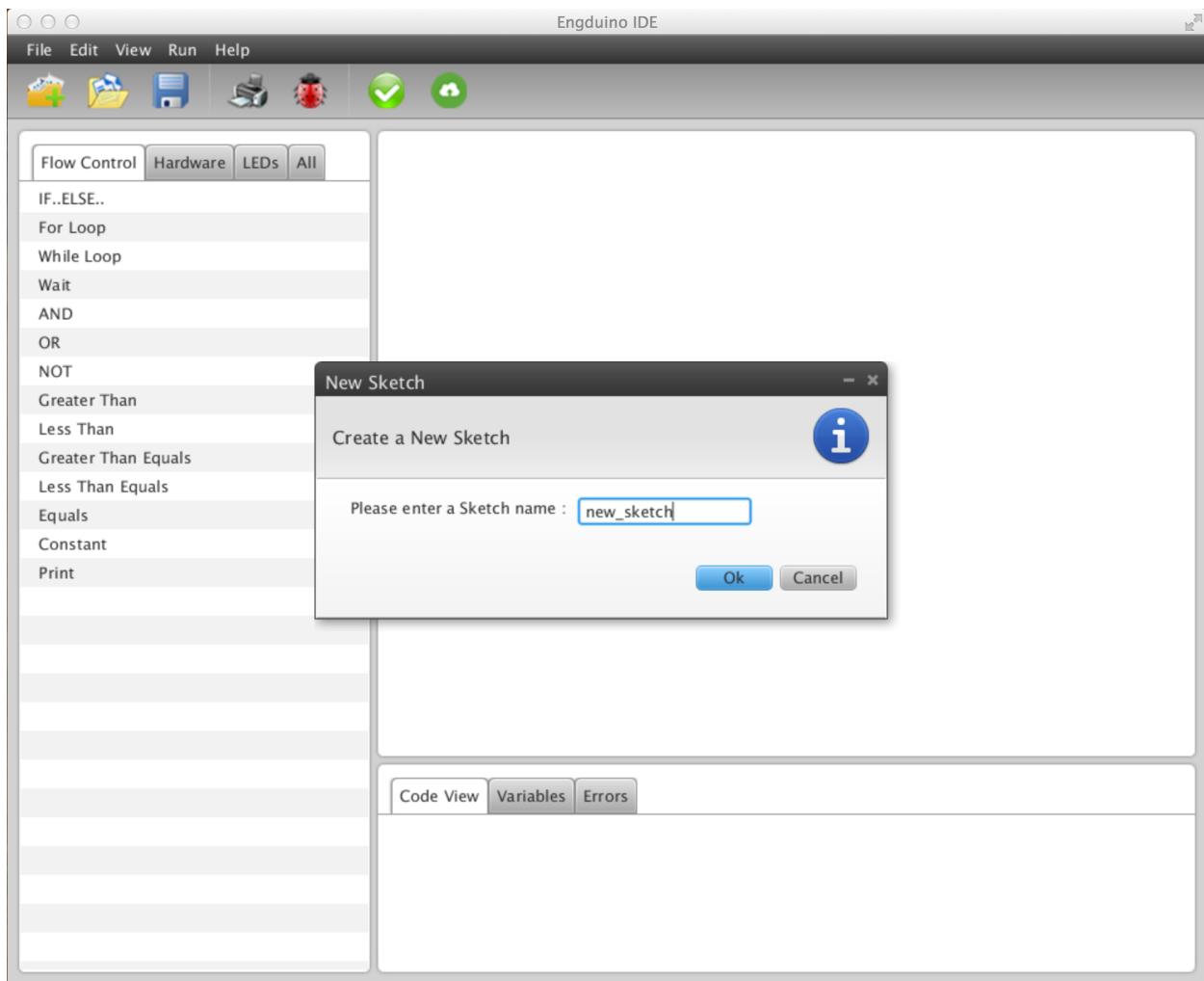
How to save sketches as custom modules for later reuse

Creating bigger program can often be tedious so we have added another powerful feature which allows the user to save the sketch as module and to import that module in different sketches. This will allow the users to reuse code and easily scale their programs and write bigger and complex programs.

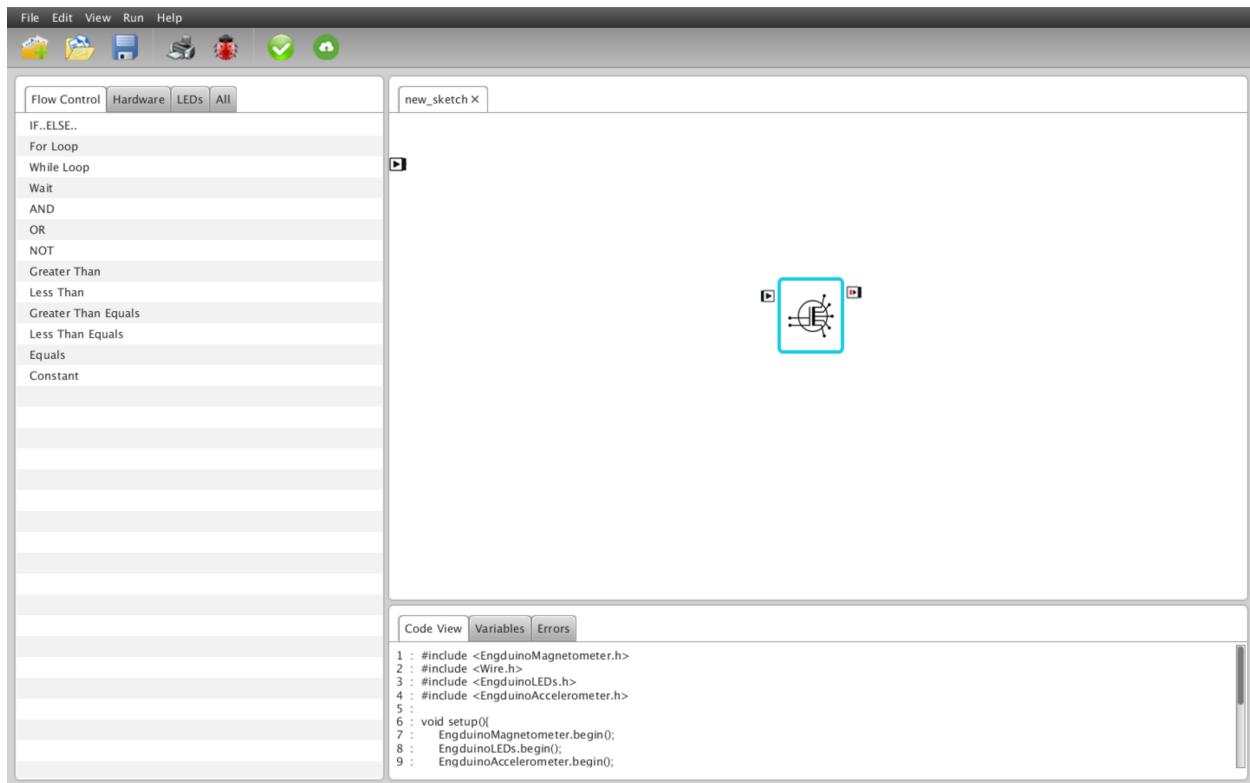
Step 1 : Create a new program and select 'Save as module' from file menu



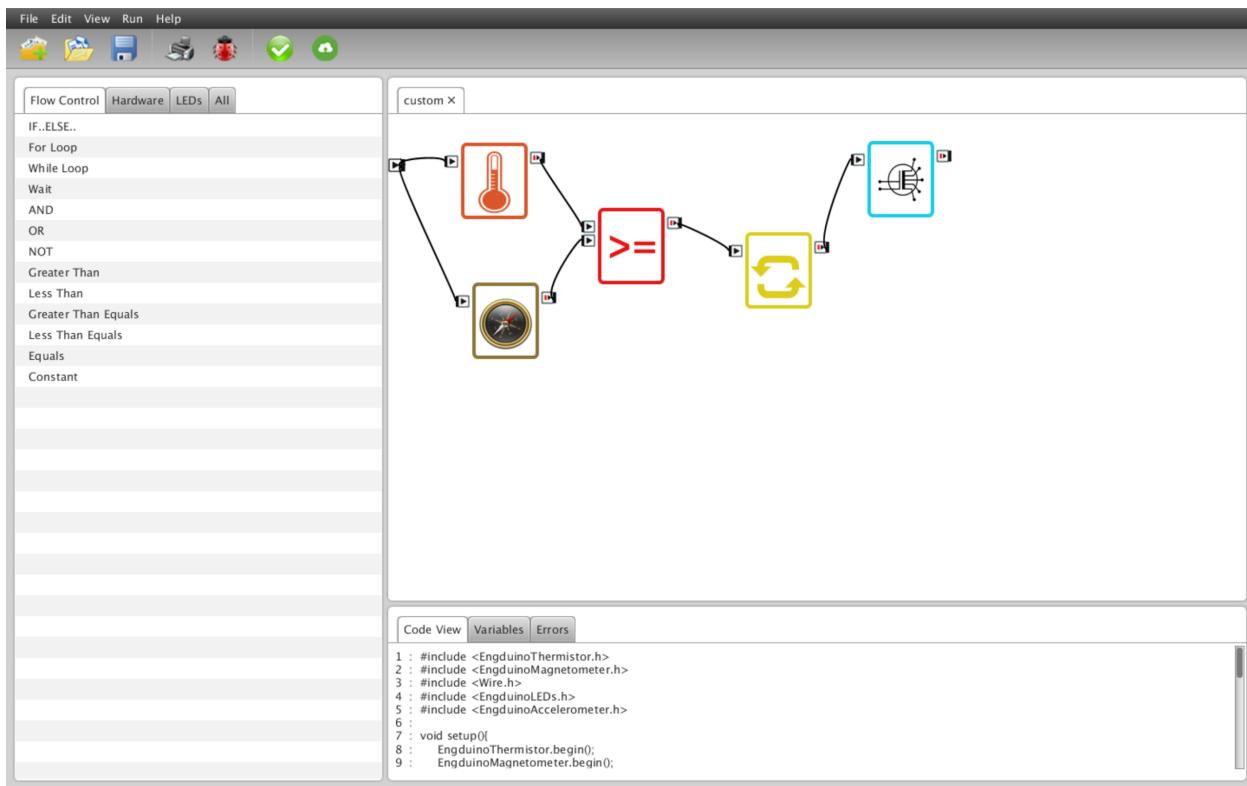
Step 2 : After your sketch will be saved as custom module, create a new sketch to use the custom module. After creating a new sketch select 'Import Sketch' from file menu and select the sketch you saved as module.



Step 3 : The custom module will appear in the sketch, by double clicking you can go inside the module and see the program and customize it

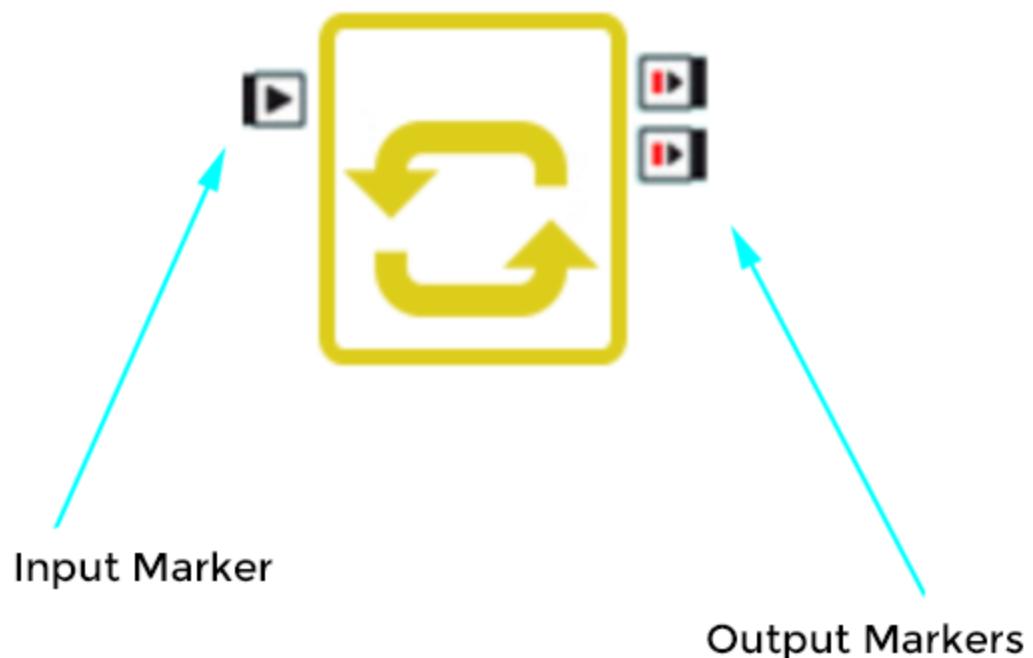


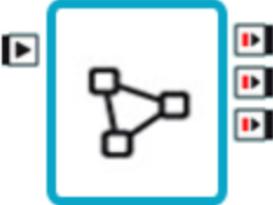
Step 4 : Create new programs which use the code from the custom sketch.



How to connect Programming flow control modules :

Only one module can be connected to an input or an output port of the marker.



Name	Image	How to connect
ForLoop		<p>A for Loop has two output markers and one input marker. It is suppose to execute instructions inside the loop body for a specific numbers of times and then exists the loop and executes whatever follows outside the loop. The first output marker i.e the upper one, represents what goes inside the loop and therefore all the instructions that you want should be connected one after the other to this port like demonstrated in the example above.</p> <p>The second port will be connected to the modules which are suppose to execute after the loop terminates.</p>
Condition		<p>The condition has three output ports, it takes a boolean value as input and check if it is true if it is then it will execute the 'chain' of modules connected to the first port.</p> <p>If the boolean value is false then the chain modules connected to the second port will be executed and at the end the chain connected to third port is executed.</p>
While Loop		<p>While loop takes a boolean input and if it is true then execute it until the condition is false so whatever is connected to the output port will be executed while the boolean condition is true.</p>
Logical Operator		<p>The logical operators such as OR, AND take two values as input and gives out one boolean value based on the type of operator.</p>

Hardware Values		Hardware or constant values can be used to give inputs to operators in the sketches which can then be used by the programming flow control to decide the execution path of the program.
Inequality Operators		The inequality and equality operators such as \geq , \leq , $>$, $<$, \neq take two values as input and give one boolean value based on the type of operator. This value can be fed to loops and conditions to decide the execution path of the program.