



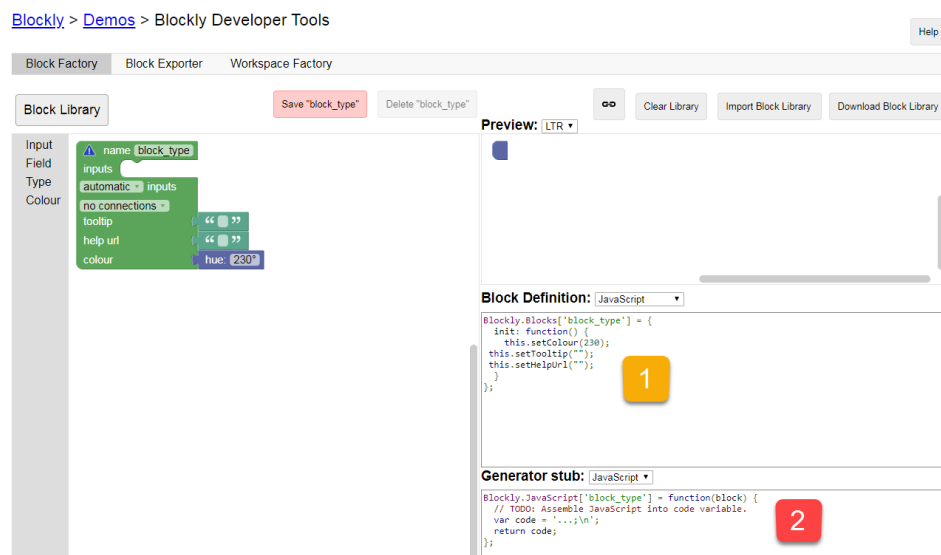
blockcoding.click


– A Guide for implementing new Blocks

The backend of blockcoding.click is implemented in JavaScript and is based the Google Blockly Framework. This is brilliant, because with that you can use the *Blockly Development Tools* from Google which will assist you in creating your new Blocks.

1. Open the Google *Blockly Development Tools* at

<https://blockly-demo.appspot.com/static/demos/blockfactory/index.html>



2. Create your Block Layout in the *Block Factory* by defining inputs, connections, tooltip, etc. and create a permanent link by using this symbol  to save it.
3. In two boxes on the right side of the screen you will see the generated code for your block. Now we need to integrate this code into blockcoding.click:

- a. View the code in JavaScript and copy the code from the *Block Definition* Box. 1
- b. Open your blockcoding.click program folder and navigate to the path `/blockly/blocks/arduino`. Now you can choose a JavaScript File from the related block category and open it in an editor or you can create a new file on your own and open that one. If you create a new file, the basic content needs to be added like in the other files and there will be an additional step to add it into the index.html file, which will be described later. As soon as you opened the file, paste the content from the *Block Definition* Box at the end of the file and save it.
- c. Now go back to the *Block Factory* Website and copy the JavaScript Code from the *Generator stub* Box 2. Open your blockcoding.click program folder and navigate to the path `/blockly/generators/arduino`. Similar to step 3b you also have the chance here to add the code to an already existing file from a block category or you are free to create a new file. Use the similar file name like you used for the *Block Definition* file at 3b. As soon as you opened the file with an editor, paste the content from the *Generator stub* Box at the end of the file. Important: Now replace the term "Javascript" with the term "Arduino" everywhere in this block!

Now you can start to implement the code that should be generated when this block is used within a block program. Within blockcoding.click you have following options to choose the location for the generated code:

Using the ...

- return value: will place the code to the current position where the block is located in the program.
- Function Blockly.Arduino.addInclude: Will add a new include line at the beginning of the program
- Function Blockly.Arduino.addSetup: Will add the code to the setup function of the program
- Function Blockly.Arduino.addVariable: Will create a global variable at the beginning of the program.
- Function Blockly.Arduino.addFunction: Will create a new function at the beginning of the program

For detailed information on how to code this section, please consult the files from existing blocks or use the documentation of blockcoding.click on the Internet <https://github.com/pati5000/Ardublockly-ESP/wiki> as well as the documentation of Ardublockly <https://github.com/carlosperate/ardublockly/wiki/Create-Blocks>

4. The next step is to add the implemented Blocks to the Toolbox of blockcoding.click and attach them to a Block Category:
 - a. Open your blockcoding.click program folder and open the file
/ardublockly/ardublockly_toolbox.js in an editor
 - b. You can see tagged categories and the blocks that are defined inside these categories. Add a new line for your block with the same syntax as the existing ones. If you want, you can also create a new block category at this point.
5. The next step is to evaluate the content of the *index.html* File
 - a. It is important to ensure that the uncompressed version of Ardublockly is loaded by the *index.html*. Only with the uncompressed version all the single JavaScript files from the implemented blocks will be loaded. Execute the following steps to ensure this:
 - i. Open your blockcoding.click program folder and open the file
/ardublockly/index.html in an editor

- ii. Change the lines 17 and 18 where you find the `blockly_compressed.js` and the `arduino_compressed.js` into comments to deactivate them.

```
15 <!-- Ardublockly - These three files contain the compress version -->
16 <!--script src="../blockly/blockly_compressed.js"></script>
17 <script src="../blockly/blocks_compressed.js"></script>
18 <script src="../blockly/arduino_compressed.js"></script-->
19 <!-- To use the uncompressed version comment out the above and comment in the ones below -->
20 <script src="../blockly/blockly_uncompressed.js"></script>
21 <script src="../blockly/javascript_compressed.js"></script>
22
23 <script src="../blockly/blocks/logic.js"></script>
24 <script src="../blockly/blocks/loops.js"></script>
```

- iii. Then remove the comment tags for the uncompressed files starting at line 20 – they need to be active. Now you can save the file
- b. OPTIONAL – Only In case you decided to create a new file in step 3b and/or 3c: These new files need to be added to the `index.html` file to let blockcoding.click know that they need to be loaded like all the other JavaScript files.
 - i. Open your blockcoding.click program folder and open the file
`/ardublockly/index.html` in an editor
 - ii. Add a new line into the `<head>` part to include the Block file from step 3b:
`<script src="../blockly/blocks/arduino/[new_file_name].js"></script>`
 - iii. Add a new line into the `<head>` part to include the Generator file from step 3c:
`<script src="../blockly/generators/arduino/[new_file_name].js"></script>`
 6. CONGRATULATIONS! You created a new block for blockcoding.click. Please reload the Ardublockly page now and see if everything works fine and test the new block.
 7. If everything works fine, it would be nice if you participate in the whole project and push your new blocks to the Github Repository <https://github.com/pati5000/Ardublockly-ESP>.