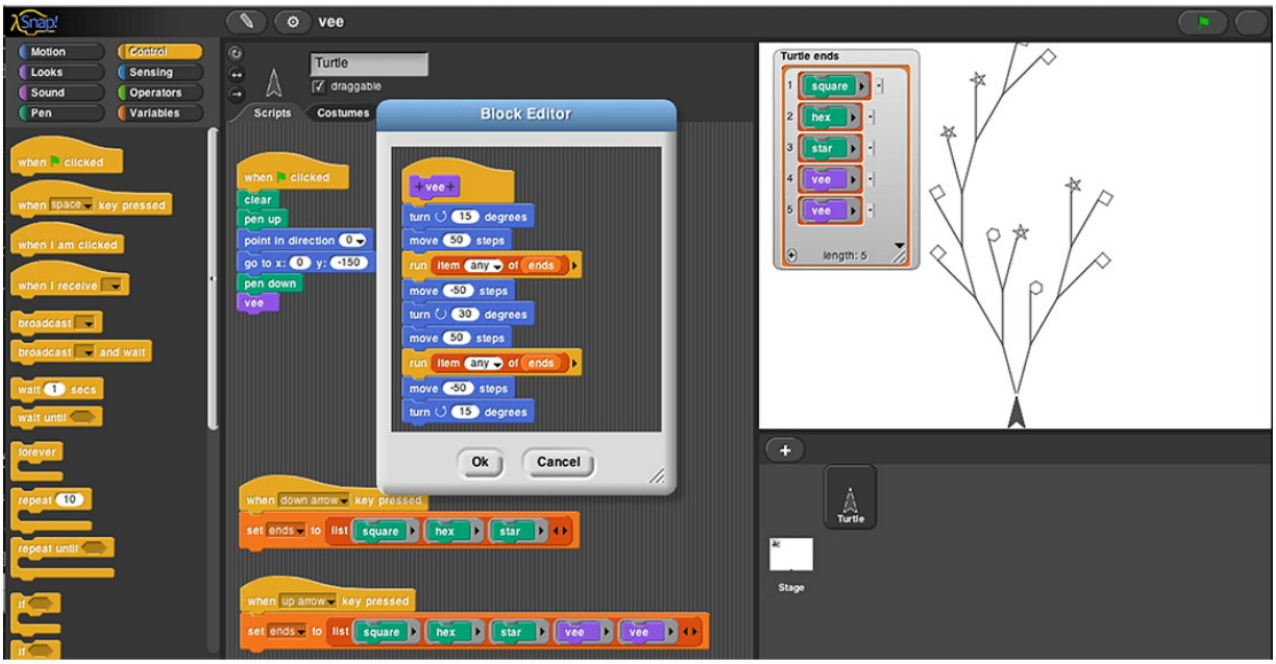
First steps with "Snap4Arduino"
Create your "Snap" for eduArdu



release 1.0/20.12.2018

I. Part - Preparations:

1. What is "Snap4Arduino" ?

- This is modified version of Snap! - visual programming language, which allows communication with almost all Arduino based boards.

- Interractive software, which beginners in programming and Arduino can use to create their first program without knowledge for particular programming language. By using small blocks the programming with Snap4Arduino is like to solve puzzle.

2. What do we need to work with Snap4Arduino ?

- Instlled ArduinoIDE
- Installed Snap4Arduino
- Copy of eduArdu Github repository (not obligatory)

2.1 Installing ArduinoIDE

- Download ArduinoIDE from:
https://www.arduino.cc/en/Main/Software
- Choose the right installer according your OS.
- Install Arduino IDE.

2.2 Installing Snap4Arduino

- Go to http://snap4arduino.rocks/
- From the menu bar click on "**Download**" button
- Choose the right insttaller for your OS.
- Launch Snap4Arduino installer or extract Snap4Arduino archive.

2.3. Cloning eduArdu Github repository

- Go to https://github.com/OLIMEX/eduArdu

- Clone the repository via GIT or click on
"Clone or download" -> Download ZIP

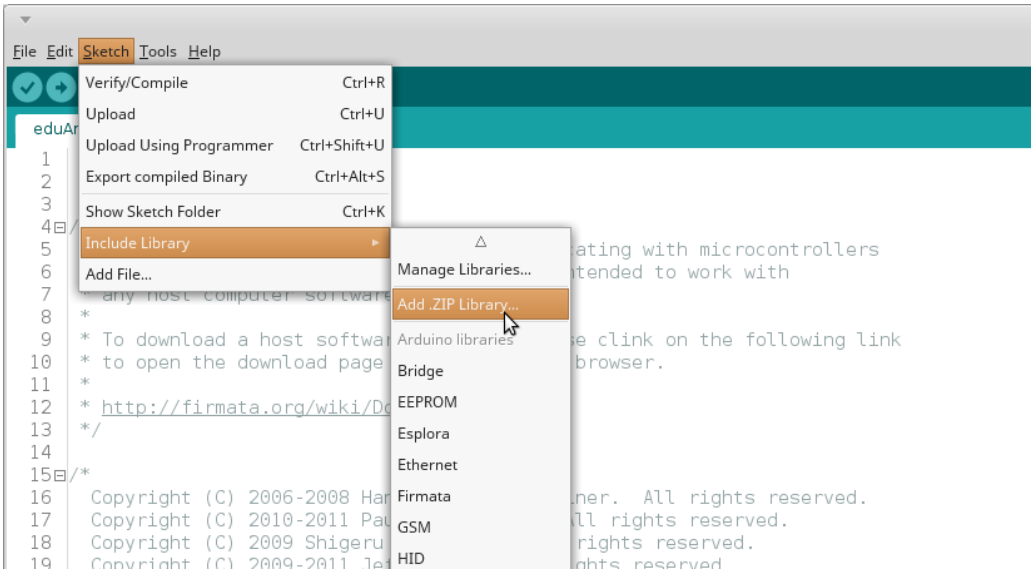
2.4. How to load eduArdu Firmata?

- Go to your local copy of the already cloned repository or extract already downloaded ZIP file. Then go to the directory:
/SOFTWARE/Snap4Arduino_eduArdu/eduArdu_Firmata/

- Open with ArduinoIDE the sketch file "**eduArdu_Firmata.ino**".

- Load all libraries sfrom **/SOFTWARE/libraries/**

To do so, click on the "**Sketch**" menu; after that go to "**Include Library**" sub-menu; choose "**Add .ZIP library**"



In the dialogue window point the full path to the library you want to include. This step is repeated for every library in "**/libraries/**"

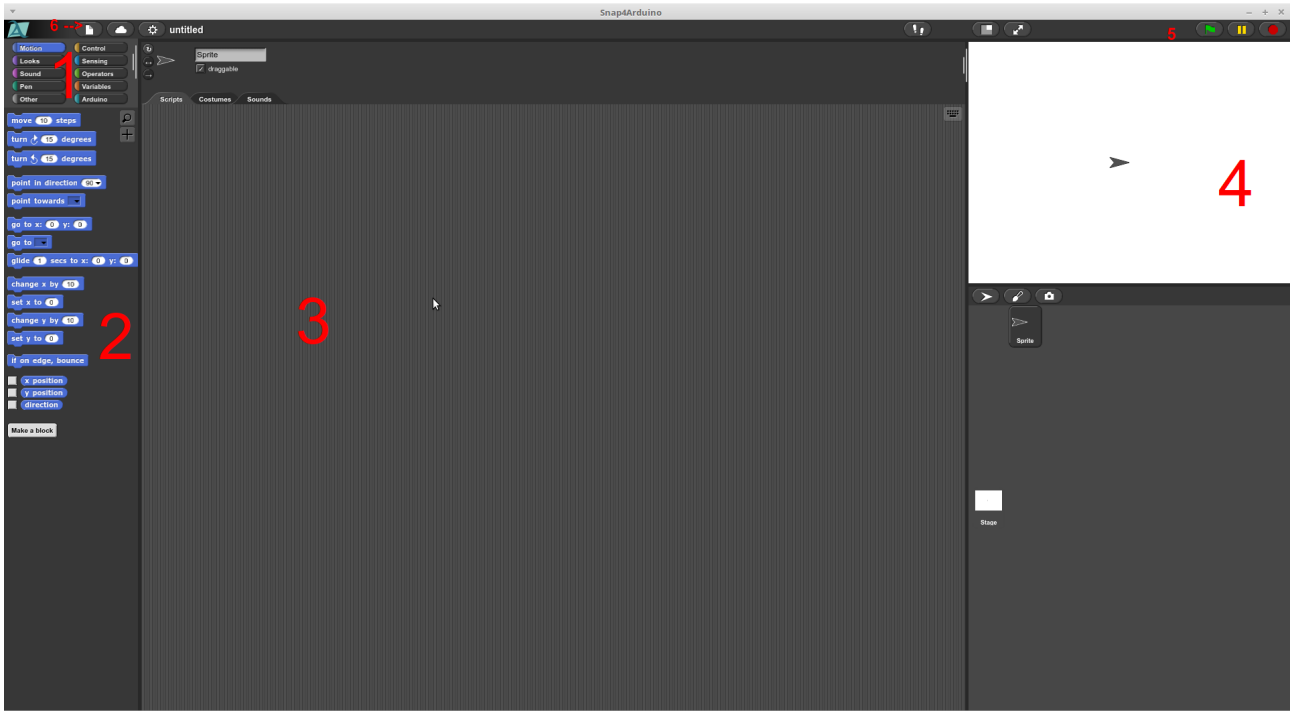
2.5. After loading all the libraries:

Once included there is no need to include again the same library.

II. Part - Intro to Snap4Arduino

Launch Snap4Arduino application.

You should see interface like this:



Short review of interface regions:

1. Block categories. Every category contains similar block types. Block categories are grouped according the block's functionality. There are several categories:
- **Motion** - Blocks used to moove Sprites
- **Looks** - pop-up balloons with information/text
- **Sound** - contains blocks for sound playback from PC
- **Pen** - Controls the pen (look label 4).
- **Control** - functions for start/stop, delay, if/else conditions, loops and etc. With those blocks you shall start and stop your program.
- **Operators** - Contains different operators for arythmetic calculations. Also this category contains blocks which can execute JavaScript code.
- **Variables** - Blocks for creation and controling variables.
- **Arduino** - This category contain all block which are related to Arduino. You can control motors, sensors; read/write data and etc.
- **Other** - This category is empty by default, but we can use it in future for our own purposes.

2. Block types.

There are 3 block types.

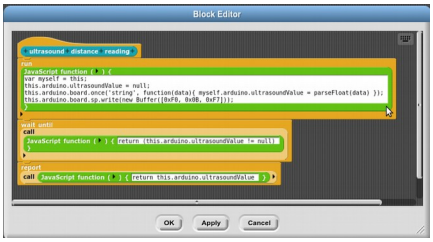
Command simply takes an action and continues execution of the program .
Reporter similar with Command blocks. These blocks also return results, which can be used as input to other blocks. Can not be used alone, they should always be put into body of another block.
Predicate very similar to **Reporter blocks**. These blocks return only True/False results.

Blocks can be used standalone. If you drag a block and leave it in Pane 3, you can activate the block by clicking on it. You can also connect different blocks, arranging them one under another, making application.

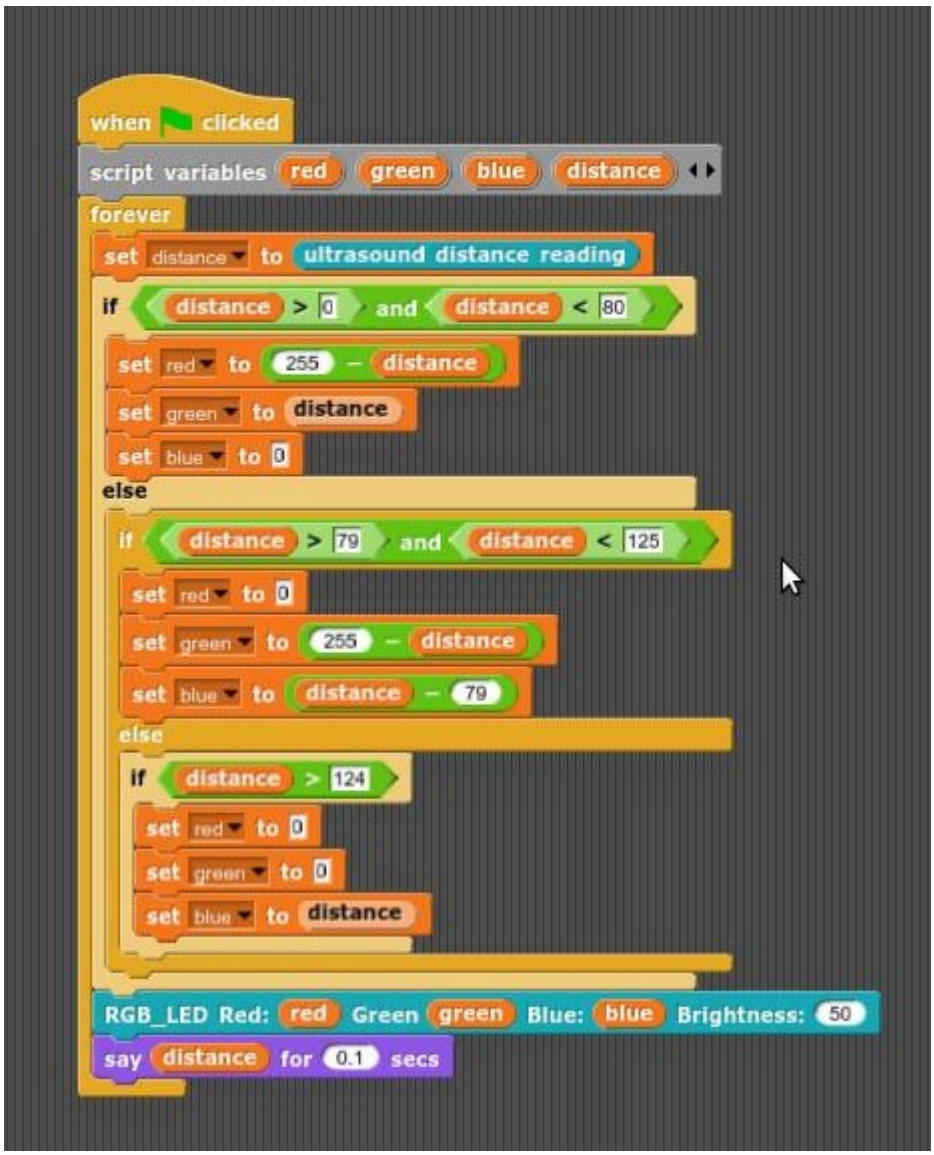
Command can be put between other blocks.

Every block or array of block can be replicated. That means, there's no need to make every time same block combination, as we can simply replicate an existing one.

Complex blocks or those created by us can be edited. Simply click with the right button on the block and choose the menu "edit":



3. Code pane - We drop the blocks that will be used in that pane. Most dialogue windowses would appear in that pane.



4. Screen - used for drawing and displaying our Sprites and animations. Block types like "Motion", "Looks", "Sensing", "Pen" woul moove the cursor on that pane; create images and animations; moove sprites.

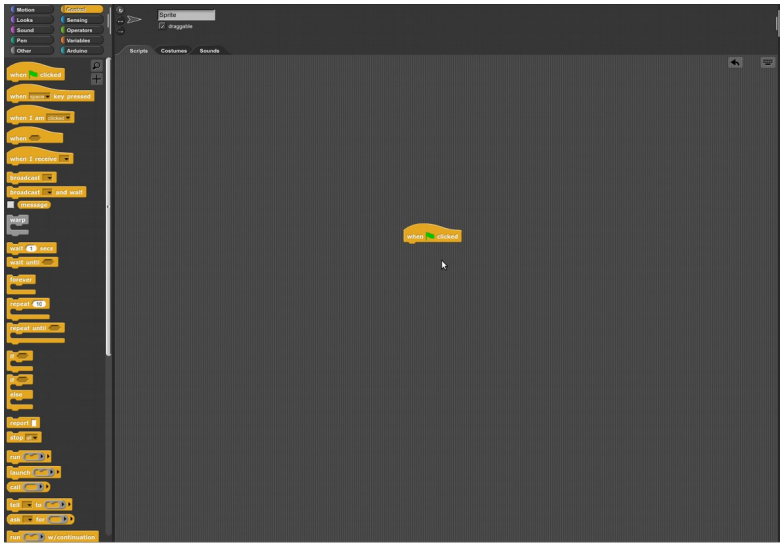
5. Menu

File; Cloud; Settings.

- From **File** menu, we open saved projects; create new project; import library and etc.

- **Cloud** menu allows us to connect to Snap4Arduino cloud and store or load our projects.

- Settings menu - contains different interface settings



Part III - "Hello World"

1. Connect MicroUSB cable to your eduArdu board.
2. Launch Snap4Arduino application
3. Go to Arduino category

If you didnt upload "eduArdu Firmata" sketch, go back to previous steps to do it.

4. Click the button "Connect Arduino".

You shall see message like this:

This means that we already set-up our eduArdu.

Go to **File -> "Import..."**, a dialogue windows must appear. Choose eduArdu.xml file that is in your local copy to eduArdu repository in .../SOFTWARE/Snap4Arduino_eduArdu/

If you followed these steps now you must see many more blocks than before.

Go to "Control" category blocks. Drag and drop this block into the **Code pane**

Go to "Arduino" category and darg "Led Matrix ShowText" block. Drop it under the "When clicked" button

Go to "Control" category again. Drag "forever" block and drop it under "Show Text" in our Code pane. Now your Code pane should look like this:

In **ShowText** textbox enter "Hello World!" text (without quotes).

Go to "Arduino" category. Drag "LED MATRIX ShiftLeft" block and drop it **in** "forever" block. Write "1" into "ShiftLeftt" textbox

Unther"LED MATRIX ShiftLeft" block drag & drop new "wait () secs" block, from "Control" category. In the textbox write "0.1"

Now our program should look like:

When you click on the green flag, the LED MATRIX of eduArdu block should start scrolling "Hello World".

III. Part - Saving a project

We can save our project in two ways:
- Saving our project on our PC
- Saving it in ".xml" file.

Choosing the first option, our project would be saved int DB, and we can not see it as standalone file.

Choosing the ".xml" option all block library, our sketch in the Code pane and everything we have done would be saved. If we later load that xml file we would not have to include libraries, that we used in that project.

How to save our project?

1. Click on "File" menu.
2. Choose "Save As" sub-menu.
3. From the dialogue window click on the "Browser" button.
4. Set project name and click "Save Project"

Hint: We can load later our project faster if we click on File menu, and double click onthe Project Name.

To save your project into ".xml" file:

1. Click on |File" menu.
2. Choose "Export project" sub-menu.
3. From the dialogue window, choose the path where to store the file.
4. Click "Save"

In case you want to contribute or edit this file, you can find it at
https://github.com/OLIMEX/eduArdu

This documents is saved in ODT and PDF files

List of changes:
1.0 - initial version from Kaloyan Nedyalkov – Olimex LTD