Wyliodrin – Program Raspberry PI from the browser using a visual language or your favorite programming language

**Imagine making your own radio or creating disco lights using a Raspberry Pi, which you control from browser. Wyliodrin gives you this possibility. The user can write, modify and run programs in real time, no matter where the board is located.**

We all faced the hustle and bustle of building our first Raspberry Pi based devices: installing the software and repeatedly plug and unplug the board from the device until it finally does what is is supposed to. There is always the SSH connection option, but what if we don’t have the required permissions?

With Wyliodrin, users can develop applications from the browser. Simply install the Raspberry Pi on the gadget you want to build, sign in to Wyliodirn and program it from your favourite browser. This way, programming becomes accessible, regardless of the board’s location, without the need to connect directly to it. The code is stored on the Wyliodrin servers, so users can access it from anywhere. Even if there appears some storage error on the Raspberry Pi, the user will replace the SD Card and he will have access to all of his programs (as they are stored online).

Many people have asked where Wyliodrin comes from. The answer is old galic, where “wylio” stands for looking our for and “trin” means to handle. As “wyliotrin” sounded a bit difficult, we adapted it to “wyliodrin”. This is how the name summarizes what the platform does: it controls and monitors the boards.

**What languages can you use?**

One of our goals with Wyliodrin is to bring students closer to engineering, to building things. With the appearance of the Raspberry Pi, building electronic gadgets has become way easier. One of the major problems encountered in the embedded field is the programming language. The most used is C. Don’t get us wrong, it is very good for small projects on microcontrollers. But what if you want to do more? The Raspberry Pi is much more complex and can accomplish way more sophisticated tasks than a microcontroller. C language makes it rather hard to access online resources for instance. Users need to know about sockets and low level programming. We think there is a need for a high level language that would have to power of C. With Wyliodrin, we have integrated the C/C++ GPIO libraries in several programming languages, so that you are able to program in language that you know. Among the available languages we have Pascal, C#, Javascript, Python, Shell, Objective-C, PHP, Perl.

This is very good for education. This way teaching electronics in school becomes easier. Students can use their known language to build things.

Starting a project from scratch can be difficult. I always used to take examples and modify them in order to learn. Wyliodrin allows you to start a new project using an existing example. This way you can see it works and you can start modifying it.

**How do you use Wyliodrin?**

Wyliodrin is currently available for the Raspberry Pi. It provides an online IDE, accessible on [www.wyliodrin.com](http://www.wyliodrin.com). All you need to do is to sign up, download the boards software from Wyliodrin and write it on an SD card and start building projects.

Wyliodrin software on the SD card is the latest Raspbian Linux withWwyliodrin’s server and development libraries installed. You might use the image directly, or you may download it’s source and build it on your own linux distribution. You may find it at <https://github.com/Wyliodrin>.

Let’s say we want to create our first project using Wyliodrin. We will make a LED blink.

Firstly we will access Wyliodrin's website, [www.wyliodrin.com](http://www.wyliodrin.com) (picture 1), where we can log in using one of our Facebook, GitHub or Gmail accounts. Now, we are on the projects’ page (picture 2), where we can create a project and add the board we want to develop on. Before doing that, we can make a virtual tour or we can study the tutorials from projects.wyliodrin.com/wiki.

Click on the “Add board” button to add a new board. You will have to name and choose it’s type. Next, you will be asked about the networking setup. In order to use Wyliodrin, you will need to connect the board to the Internet so it can connect to Wyliodrin’s servers. If you have an ethernet connection, plugin in the network cable and it should work. If you have WiFi, you will need to attach a WiFi dongle into the Raspberry Pi and write the network’s SSID and password on the website.

Once it is setup, Wyliodrin will display you some instructions how to connect the board.

1. You need a 4 GB SD card, preferably a class 10 for better performance
2. Download the Wyliodrin SD card image from <https://projects.wyliodrin.com/images/raspberrypi>
3. Write it on the SD card. You may do this differently depending on your system and it is done the same way you did with the official Raspberry Pi image. On Windows, you can use Win32DiskImager (<http://sourceforge.net/projects/win32diskimager>), on a Mac PiWriter (<http://sourceforge.net/projects/piwriter/>) and on Linux dd. For more information, please look at our turorial <https://www.wyliodrin.com/wiki/boards_setup/raspberrypi>.
4. Once you have written the image, you need to copy a configuration file on it. This file is specific to your board. It is a JSON containing your board’s identification and network setup. You can download it from your Wyliodrin account from the boards popup menu (picture 3). Copy the downloaded file on the root of the SD card and name it wyliodrin.json.
5. Power up the board and wait for it to appear online.

For the electronics setup, we need the following objects:

* 1 breadboard
* 1 LED
* 1 resistor (220 Ohm)
* 2 jumper wires
* the Raspberry Pi model B.

The schematics is shown in picture 4.

You have to wire the components, keeping in mind that each pin on the board acts like a software programmed power source. On Wyliodrin's tutorial about Raspberry Pi's pins you can see how to connect the components to the Raspberry Pi.The device needs to be connected to the internet using a wire or WIFI.

The next step is to implement the project, so we go back to the projects’ page. Select “Create new project” and give it a name, a description and choose a programming language: Visual Programming, Shell Script, C, C++, C#, Java, Javascript, Objective-C, Pascal, Perl, PHP or Python.

The “Main” file was implicitly created together with the project. There, you can write the code in the chosen programming language and then we will run it on the board. In picture 5 you have the code for making a led blink using C++. By choosing C++ -> LED Blink C++ when creating a new project, you can have and use the code provided by Wyliodrin. You only have to “Run” the program and to select the board in order to make LED blink. All this time, the board had no connection to the computer you wrote the program on.

For the beginners, who don’t master a programming language, they can add blocks by drag-and-drop and Wyliodrin will write the code for them. Visual Programming used by Wyliodrin is based on Google Blockly, a language like Scratch, but for electronics. Wyliodrin implemented pins, leds and buttons blocks, to let users run several applications.

To make a LED blink using the Raspberry Pi and the visual programming blocks Wyliodrin implemented, we used the blocks in picture 6. There can be seen the resulting code, which is in Python (picture 7).

**Music and an LCD**

The next project that we want to show you is building a music station with an LCD. We will use as sound input an online radio and we will display on an LCD a VU meter and the name of the song that is currently played. For this, we will need to following:

* 1 breadboard
* an LCD screen (16x2 is ours, but it can be a different one)
* 1 potentiometer
* 14 jumper wires
* the Raspberry Pi model B.

The schematics is shown in picture 8. Wiring up an LCD can be done in two ways, we will use the one that requires a smaller number of wires. First of all, you have to power up the LCD. Please be carefull, the pin layout of the LCD may vary from model to model, so take a look at the data sheet. Our LCD is a 5V one with 16 columns and two lines. It’s data sheet can be found on <http://robofun.ro/docs/RC1602B-BIW-CSX.pdf>. You connect the ground (VSS) pins to the ground of the Raspberry Pi, the 5 V pin (VDD) to the 5 V on the Raspberry Pi. The VO pin of the LCD is used to set up the contrast, so connect it to a potentiometer. Also wire the backlight pins (LED - and +) to the ground and 5 V on the Raspberry Pi.

You also need to connect the RS, E, DB4, DB5, DB6 and DB7 to the Raspberry Pi. You can connect them to any of the Raspberry Pi GPIO data.

The R/W pin is connected to the ground, as we will only send data to the LCD.

Let’s make a new project and write the software. Select the Visual programming language, so that you can drag and drop blocks and Wyliodrin will write the Python code for you. Start from an existing project called Music. This way you have the music player set up. Once you have created the project, click it and edit it.

As you can see in picture 9, you already have the blocks for playing music from an online radio. The default online radio is not showing the title of the song, so please change it to another. Click on the block named “Load audio stream from” and write another address. Use any online radio station that you like. Once you have written the address, let’s test if it plays. Press the Run button on the top. Don’t forget to use a loudspeaker for the Raspberry Pi.

Now let’s modify the program so that it displays the song title and then shows a VU meter. First of all, you need to initialize the LCD display. For this, go to the LCD blocks on the left toolbar and drag the Init LCD block and place it between the Play audio stream and repeat blocks. You will see you need to set the pin numbers. In our example, the pins are RS 0, E 2, Data1 3, Data2 12, Data3 13 and Data4 14. They are the same as in the schematics. Another parameter is the number of columns and lines. Our LCD is 16 columns and 2 lines.

First let’s will display the Radio station information. Inside the repeat block, after the delay block, put a Reset Position on LCD block. The LCD works like the normal screen, once you write on it, it moves the cursor ahead. As we want to display the name at the beginning, you need to reset the cursor’s position. After this, place a Print on LCD block. Inside this block, replace the text with the Audio Stream Address Title block found in the Multimedia blocks of the toolbar. This block receives a parameter with the variable of the stream. By default, this is called audio. Change it to music (the same variable used for the other multimedia blocks).

Running the project should result in displaying the station name or song title (depending on what the station broadcasts). Some stations do not provide this information so don’t worry if nothing is written.

The VU meter is a measure of how loud the music is. We want to display this in the form of \* on the LCD. For this, you need to use the Stream level block from the Multimedia blocks. First of all, set a variable called l to the value returned by Stream level. For this, drag the Set variable to block from Variables. Connect to this block the Stream level block. Select the correct stream name (music) and scale it to 10. the scaling will tell the block to return a value between 0 and 10.

Next we need to display the stars. Set the LCD position at the beginning of the second line using a Set LCD position block. We will take a count block, set it’s variable to i and make it count from 1 to 10 by 1. This means that the content of this block will be repeated for every value of i from 1 to 10. The first time it repeats, the value i will have the value 1, the next time 2, and so on up to 10.

Inside this block, we will display on the LCD either a \* either a space using the following rule. If the variable l (that stores the current level) is less than i, then we display a \*, otherwise we display a space. For this we will use a condition block called if. Drag an if block inside the count block. The if block has a star on it. if you click it, a small window will appear and we may set up the block. We need to drag an else block inside the if. Once you have done this, you are ready to write on the screen. Connect to the if the less or equal condition block. You can find it on the Logic block on the toolbar. The condition that we want is i <= l. We need to drag the i and l blocks from the Variables blocks in the toolbar.

The if block has two parts; do and else. The do part will be executed if the condition is true, the else will be executed if the condition is false. In the do part, place a Print in LCD block with the text \*. On the else part, put a Print on LCD block with the text a space.

If you run the program, you will see that the VU meter is displayed but is changes once a second. This is due to the delay block that makes the program wait 1000 milliseconds. Write a lower value for a faster update.

The program is displayed in picture 10.

**Future perspectives**

We work on implementing graphics, which will allow the users to monitor their boards in real-time. We also plan on introducing more blocks for the visual programming language.

We would like to ask you to provides feedback, that is very important for us. Of you would like new features or think that we should differently, please don’t hesitate.

You can read more about the project by accessing the website www.wyliodrin.com, on Facebook and Google Plus pages Wyliodrin and on the Twitter account @wyliodrin.