## **Device programming. Raspberry and Arduino**

## **Introduction**

Raspberry Pi is an amazing minicomputer, and I would love to use it in some projects. There is just one tiny problem. I have little to no experience with Python. I used to do some Python coding a few years ago but only the basics. I’m more experienced in C++, specifically writing programs for Arduino. Wouldn’t it be nice if there was some way for those of us to utilize our Arduino programming skills on Raspberry Pi? Luckily, there is!

In this article, you will learn how to run sketches written for Arduino on Raspberry Pi! To achieve this, we will use RasPiArduino framework. That will allow us to compile the Arduino code into binaries which can run on Raspberry Pi. But before we can do that, we have to prepare a few things, both in the Arduino IDE and on Raspberry Pi.

## **Equipment**

Hardware

* Raspberry Pi 3 Model B

Software

* Arduino IDE
* RasPiArduino framework – <https://github.com/me-no-dev/RasPiArduino>
* MobaXterm – <https://mobaxterm.mobatek.net/>

## How to Run Arduino Sketches: Arduino IDE Setup

1. Open Arduino installation folder. On Windows systems, this will most likely be “C:\Program Files\Arduino” on 32-bit system or “C:\Program Files(x86)\Arduino” on 64-bit system.

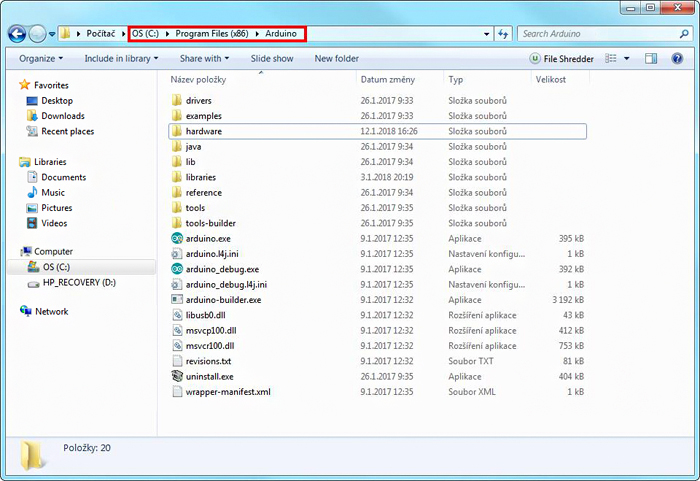


Figure 1. Arduino installation folder

2. Open the folder “hardware” and create a new folder named “RaspberryPi” in this folder.

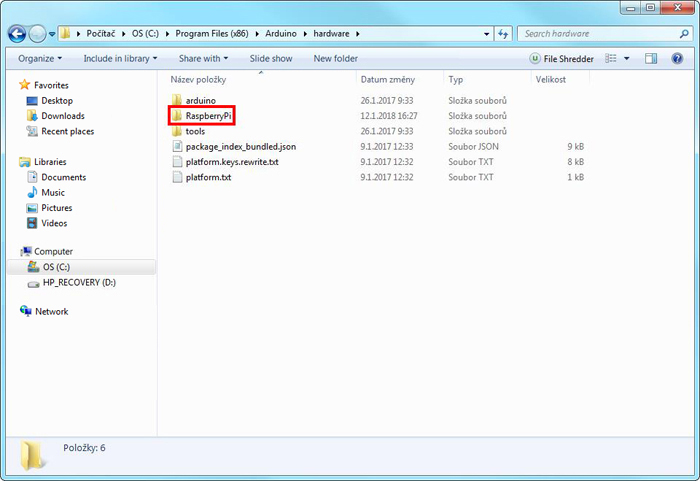


Figure 2. The newly created folder

3. Go to <https://github.com/me-no-dev/RasPiArduino> and clone the repository into a new folder “piduino” inside “RaspberryPi”. If you don’t want to clone the repository, you can download the repository content as a zip file and unzip it into the “piduino” folder.

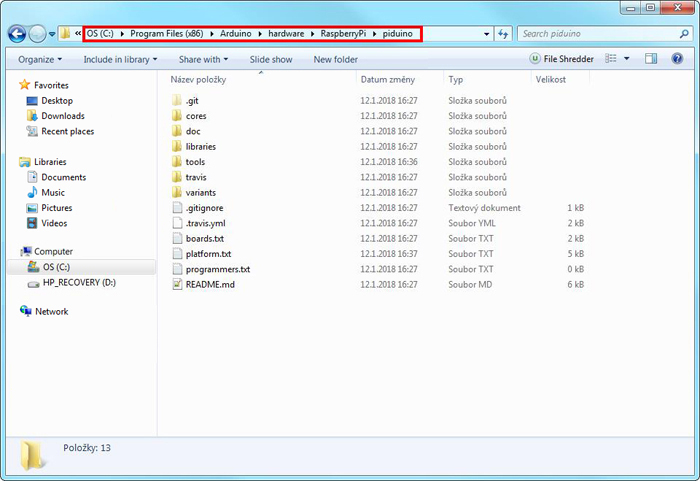


Figure 3. The content of “piduino” folder

4. Next, you have to download GNU Toolchain from <http://gnutoolchains.com/raspberry/>. You have to download the version that supports the version of your Raspberry OS. Since you probably use the latest Raspbian Stretch, you will want to download GCC 6.3.0. Once the file is downloaded, run it. Leave the installation path at “C:\SysGCC\raspberry”, check you accept the license agreement and click the button “Install”. After a while, the installation should complete successfully.

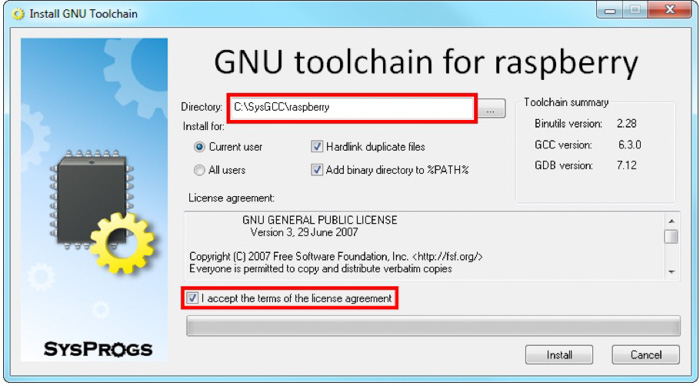


Figure 4. GNU Toolchain installer

5. Inside the Arduino installation directory, go to the “piduino“ folder we created in step 3. Open the file “platform.txt” and change the line number 5 from  
  
runtime.tools.toolchain.path={runtime.platform.path}/tools/arm-linux-gnueabihf  
  
to  
  
runtime.tools.toolchain.path=C:/SysGCC/Raspberry  
  
This will point the compiler to the directory that contains the toolchain we installed in step 4.



Figure 5. Contents of platform.txt

6. Restart Arduino IDE and open a new sketch. In the “Tools” menu, under the “Board” option, there should now be a new board called “RaspberryPI B+/2”. To make sure everything is installed correctly, select the Raspberry Pi board, copy the code below and compile the sketch. The compilation might take longer than for Arduino boards and generate some warnings, but as long as it says “Done compiling” at the end, it should be fine.

void setup() {

Console.println("This is the SETUP function");

}

void loop() {

Console.println("This is the LOOP function");

delay(1000);

}

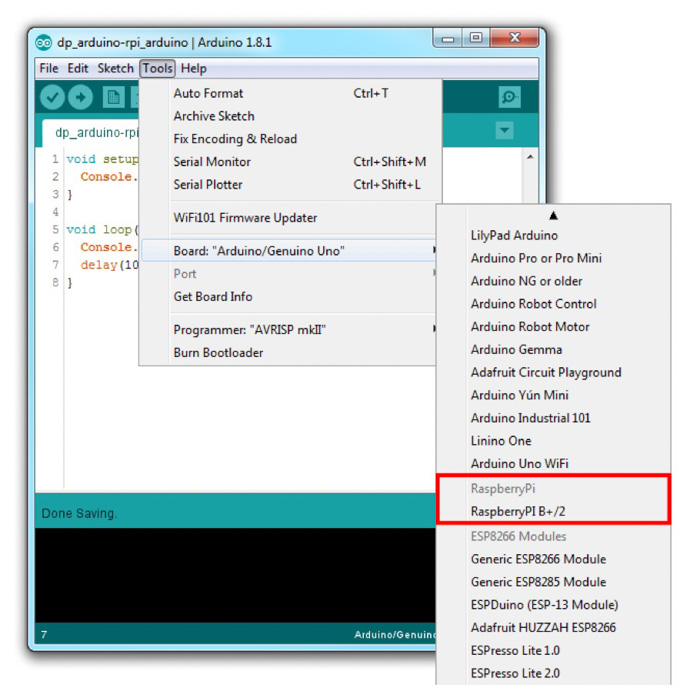


Figure 6. Raspberry Pi board in Arduino IDE