

## 2.3 Software installation

We will need software listed below:

1. **Arduino IDE** is basic “development environment”
2. **RobDuino** library for easier programming
3. **Ardublockly** is needed for introduction to programming
  - Python is needed for running Ardublockly
4. VSC in PlatformIO proper IDE include:
  - auto-completion,
  - error marking (e.g. forgotten ";"),
  - auto-detect USB port,
  - function information

### 2.3.1 Arduino IDE

1. Go to Arduino web page Arduino->Software->**Download**.
2. Download **Arduino IDE 1.8.9** choose **Windows Install...**
3. ... click **JUST DOWNLOAD**.
4. run **arduino-1.8.9.exe** and follow the instructions.
5. ... don't forget to install also 3rd party drivers (for Chinese version of Arduino UNO controller)...
6. if you do forget... Try this **Russian drivers** from **page**.

#### 2.3.1.1 Getting started

1. Run **Arduino IDE**
2. Connect Arduino Uno controller to USB port.  
**Arduino Uno**
3. Open simple basic program:  
**files -> examples -> 01.basics -> blink**

```

1  void setup() {
2      pinMode(LED_BUILTIN, OUTPUT);
3  }
4
5  void loop() {
6      digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the
          voltage level)
7      delay(1000);                        // wait for a second
8      digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the
          voltage LOW
9      delay(1000);                        // wait for a second
10 }

```

4. Make this settings in **Tools** menu ->

1. **Board:** Arduino/Genuino Uno
2. **Port:** COM3 or similar

5. Run :

**Upload** to transere the program to Arduino UNO controller.

6. If everything is OK you will get this message:

```

1  Done uploading.
2  Sketch uses 970 bytes (3%) of program storage space. Maximum is 32256
   bytes.
3  Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes
   for local variables. Maximum is 2048 bytes.

```

9. Optional this preferences are suggested:

**File -> Preferences:**

1. **Editor Language:** English
2. **Editor font size:** 20
3. **Show verbose output during:** ☐ compiling ☒ upload
4. ☒ **Display linenumbers**
5. ☒ **Enable code folding**

### 2.3.2 RobDuino

RobDuino is Arduino library which include some usefull functions for driving motors and on-board key usage...

#### 2.3.2.1 RobDuino Library Installation

1. Download zip file:
  - [RobDuino-master.zip](#)
2. rename RobDuino-master.zip in:
  - **RobDuino.zip**
3. run Arduino IDE
4. choose:
  - [Sketch -> Include Library -> Add .ZIP Library...](#)
5. find
  - [.../Download/RobDuino.zip](#)
  - [\[OK\]](#)

### 2.3.3 Ardublockly

[Ardublockly](#) is [graphical programming environment](#) for programming Arduino controllers. A demo version of the program is also available [on-line](#).

Note: For actual programming you will need Arduino IDE installed.

Note: For running Ardublockly you will need to install Python program.

#### 2.3.3.1 Python Installation

1. You will have to install [Python 3.7](#) or grater. First [Download](#) the newest version of Python.
2. Run installation file and set this settings:
  1. [\[x\] Add Python to PATH](#) in
  2. choose [Clasic Instalation](#)

#### 2.3.3.2 Ardublockly Installation

3. From [github.com/.../ardublockly](#) download **zip** file by clicking **Clone or download** and choosen [Download ZIP file](#).
4. Extract [ardublockly-master.zip](#) to dirrectory of your choice e.g. [C:\\Program Files\(x86\)](#)
5. That is it! Installation is complete.

### 2.3.3.2.1 Running Ardublockly

6. Find this file `C:\\Program Files(x86)\\ardublockly-master` and double-click on `start.py`. Python program should run and you should see:
  1. terminal window with some code running...
  2. and a new window should appear in your Internet Browser. If this is will not happend try to run `start.py` with right mouse button and `Start program with` then choose `Python 3.7`.

### 2.3.3.3 Settings

7. Click `menu` and choose `Settings`:
  1. `Compiler Location`: `C:\\Program Files (x86)\\Arduino\\arduino_debug.exe`
  2. `Arduino Board`: `Uno`
  3. `Com port`: `COM3 or appropriate one`
  4. Click `[RETURN]`.

### 2.3.4 VSC in PlatformIO

Note: For programming Arduino controllers you will need Arduino IDE installed.

Download installation file:

1. run `VSCodeUserSetup-ia32-1.49.3.exe` installation file.
2. run VSC program and click `Extensions`
3. search for `PlatformIO IDE` and
4. run `Install`.
5. restart VSC or click `Reload now`.

#### 2.3.4.1 Getting Started Write basic program `Blink`:

1. plug in Arduino Uno.
2. open `PlatformIO - Home Page`:
  - in left icon bar find `PlatformIO`
    - `QUICK ACCESS -> PIO Home -> Open`
3. choose + `New Project`
4. Setup:

- **Name:** ime\_projekta
- **Board:** Arduino UNO
- **Framework:** Arduino Framework

5. click **Finish**

6. Find directory **src** (e.g. **source code**), where you can find main program code in file **main.cpp**

7. Copy-Paste this example:

```
1  #include <Arduino.h>
2  void setup() {
3      pinMode(13, OUTPUT);
4  }
5
6  void loop() {
7      digitalWrite(13,HIGH);
8      delay(500);
9      digitalWrite(13,LOW);
10     delay(500);
11 }
```

8. Run **Build** and **Upload**.