Laboratory work No. 1

**Goal of research:**

Familiarity with the development environment.

**Software:**

STM32CubeIDE.

**The order of work:**

*Part I: Creating a Project Using a Code Generator.*

1. Create a working folder for the project. Run STM32CubeIDE, in the window that opens, set the path to your working folder. The path to the working folder and the name of the project should not contain Russian letters. All labs will be stored in this folder.

The image looks like text

Auto-generated description

2. In the window that appears, you need to create a project. Either click "Start new STM32 project" or "File->New->STM32 Project".

The image looks like text

Auto-generated description

3. After opening the window for selecting the end device, in the "Part Number" field, you must enter the name of the microcontroller - STM32F446RE. From the list, you must select the option - STM32F446RE.

The image looks like text

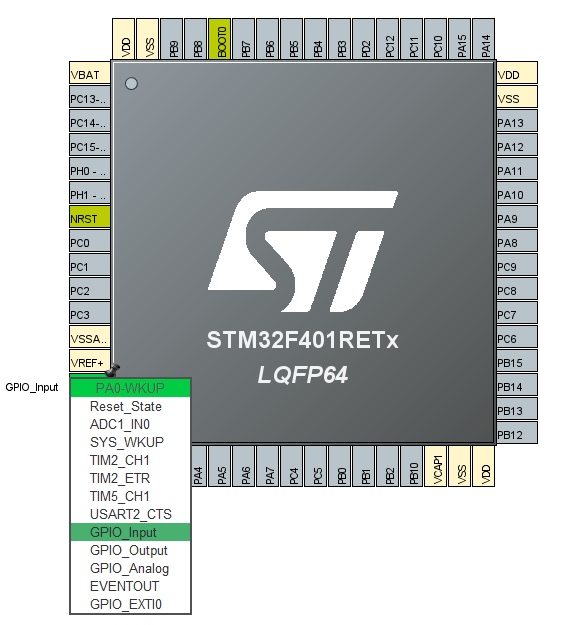
Auto-generated description

4. In the next window, you need to enter the name of the project, select the language - C, the output file types - Executable, the project type - STM32Cube; click Finish. The message must be answered Yes.

The image looks like text

Auto-generated description

5. A schematic image of the microcontroller will appear in the window that opens. In order to set the mode of operation of the pin, you need to click on it and select the mode.



Set the PB5 pin to GPIO\_Output.

6. Generate the code by clicking Project->Generate Code on the toolbar. A new tab will open with the program code - "main.c". If "main.c" does not open, then it must be opened manually, it is located in Project Explorer at Core->Src->main.c.

7. In this controller programming mode, the user needs to write code exclusively in areas that have the following beginning and end:

/\* USER CODE BEGIN ### \*/

/\* USER CODE END ### \*/

Where ### is the name of the user code block created by the generator.

8. Writing the first program:

To the area named: /\* USER CODE BEGIN WHILE \*/, after the lines

**while** (1)

{

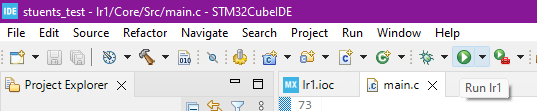
Add the following code:

HAL\_GPIO\_TogglePin(GPIOB, GPIO\_PIN\_5);

HAL\_Delay(1000);

9. After writing the code, click Project->Build all to compile the code. If there are no errors in the program, then in the Build Console there will be an inscription: "Build Finished".

10. To program the controller, just click the Run button on the toolbar:



At the first launch, the launch configuration window will open, in which you must select (if not automatically specified) the executable file of the program with the  *.elf* extension.

The image looks like text

Auto-generated description

After completing all the steps on the board, the LED should start changing its color.

**Task**

1. Follow all the steps in Part I.

2. Modify the program so that PA8 is used instead of PB5 pin.

3. Modify the program so that the state changes to PB5 first, and PA8 through the delay.

Demonstrate all the items of tasks in turn to the teacher.