

## Лабораторная работа №1

### Цель работы:

Знакомство со средой разработки.

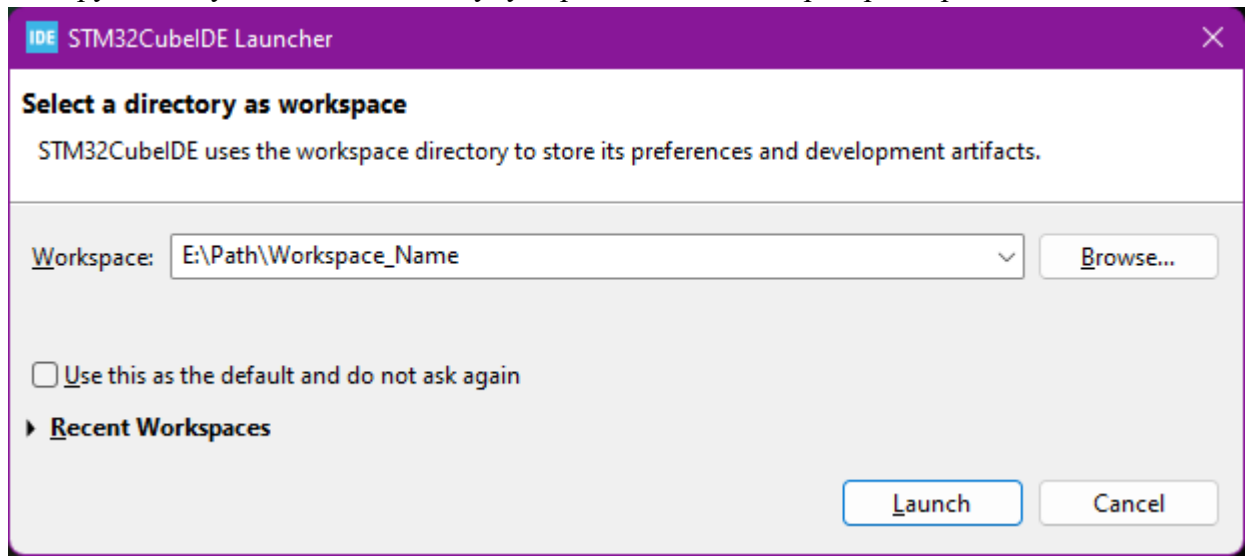
### Программное обеспечение:

STM32CubeIDE.

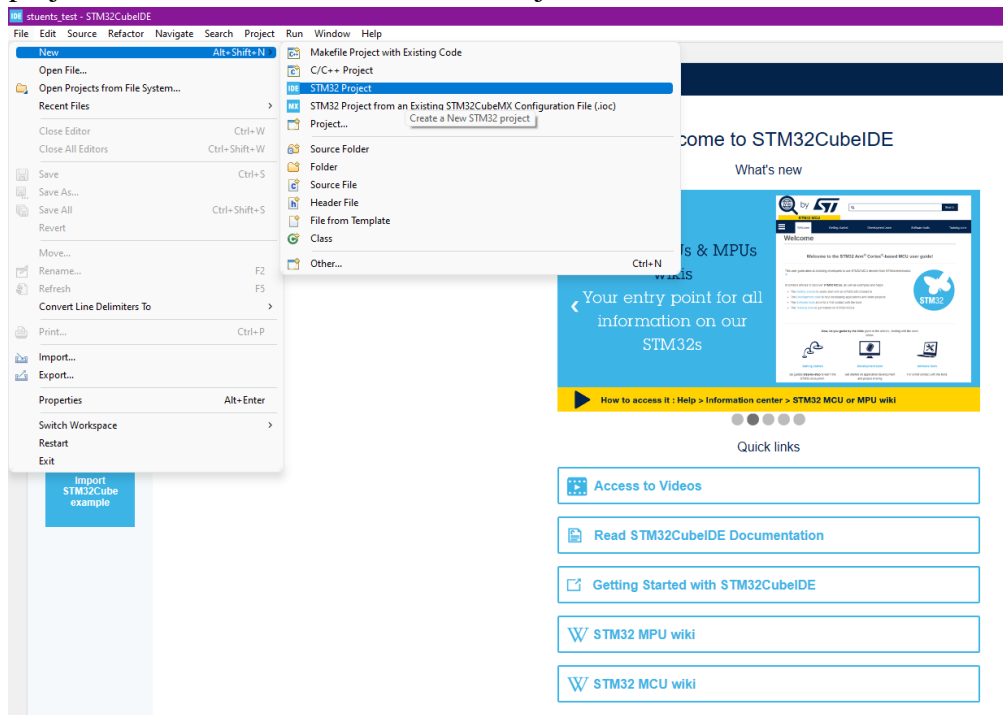
### Порядок выполнения работы:

*Часть I. Создание проекта при помощи генератора кода.*

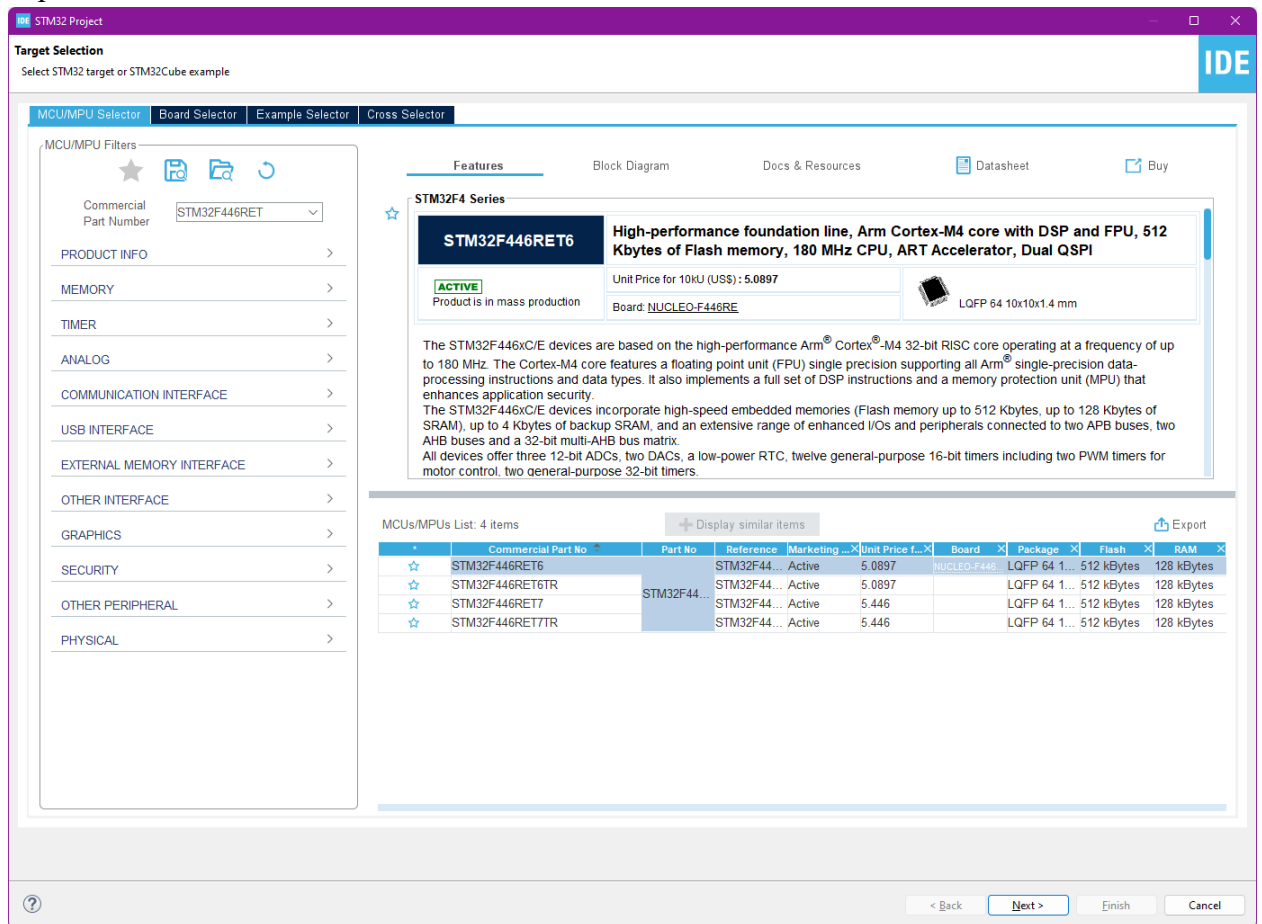
1. Создайте рабочую папку для проекта. Запустите STM32CubeIDE, в открывшемся окне задайте путь к вашей рабочей папке. В пути к рабочей папке и названии проекта не должно быть русских букв. В этой папке будут храниться все лабораторные работы.



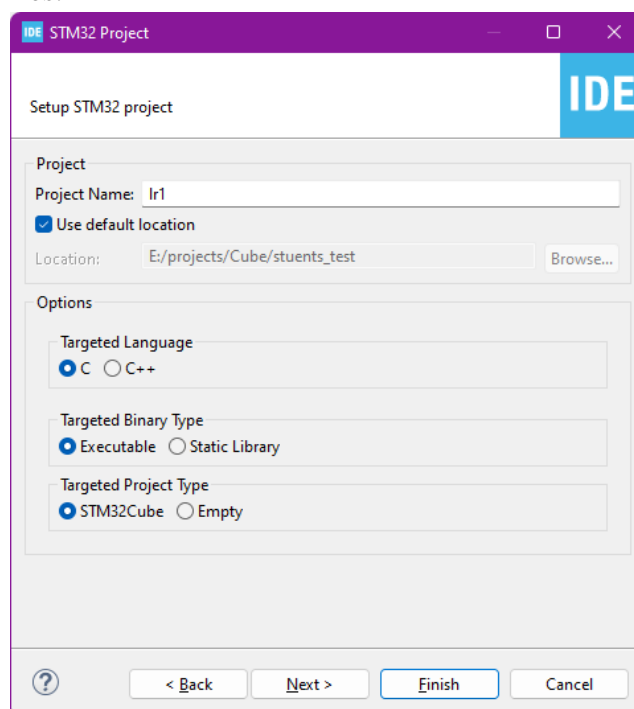
2. В появившемся окне необходимо создать проект. Либо нажмите «Start new STM32 project», либо «File->New->STM32 Project».



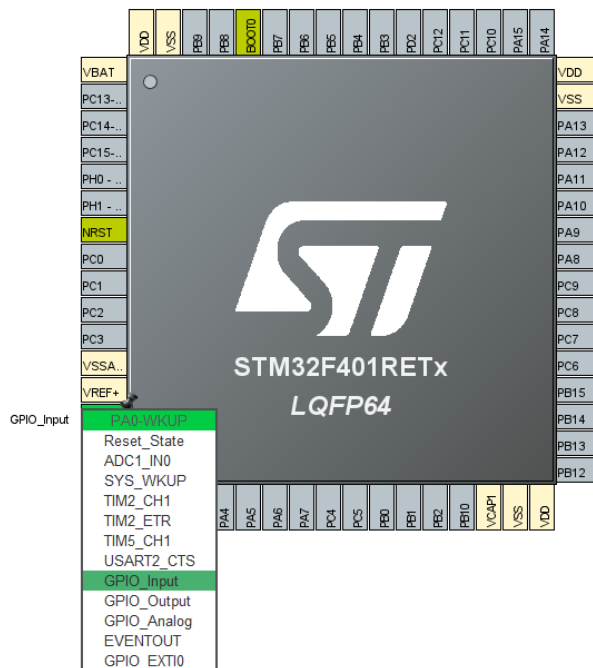
3. После открытия окна выбора конечного устройства, в поле «Part Number» необходимо ввести наименование микроконтроллера – STM32F446RET. Из списка необходимо выбрать вариант - STM32F446RET6.



4. В следующем окне необходимо ввести название проекта, выберете язык – C, типы выходного файла – Executable, тип проекта – STM32Cube; нажмите Finish. На сообщение необходимо ответить Yes.



5. В открывшемся окне появится схематичное изображение микроконтроллера. Для того что бы задать режим работы пина, необходимо нажать на него и выбрать режим.



Задайте для пина PB5 режим – GPIO\_Output.

6. Сгенерируйте код, для этого нажмите Project->Generate Code на панели инструментов. Откроется новая вкладка с кодом программы – “main.c”. Если “main.c” не открылся, то его необходимо открыть вручную, он находится в Project Explorer по пути Core->Src->main.c.

7. В данном режиме программирования контроллера, пользователю необходимо писать код исключительно в областях, которые имеют следующие начало и конец:

```
/* USER CODE BEGIN ### */

/* USER CODE END ### */
```

Где ### - наименование блока пользовательского кода созданное генератором.

8. Написание первой программы:

В область с названием: `/* USER CODE BEGIN WHILE */`, после строк

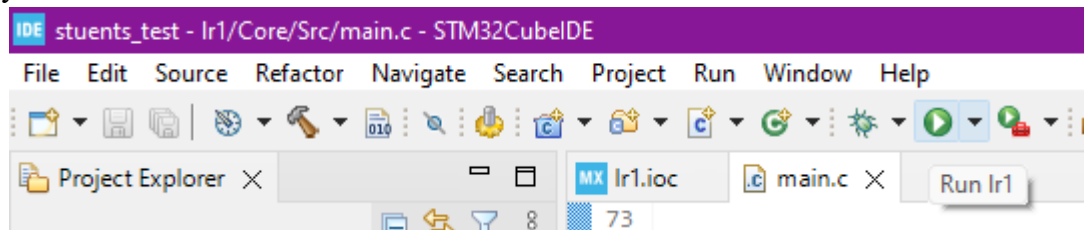
```
while (1)
{
```

Добавьте следующий код:

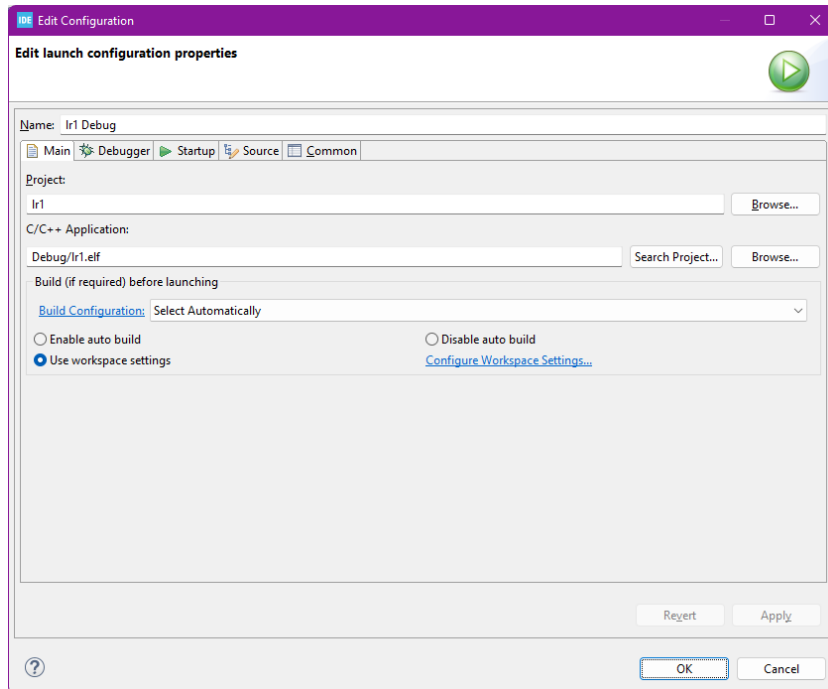
```
    HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_5);
    HAL_Delay(1000);
```

9. После написания кода нажмите Project->Build all для компиляции кода. Если в программе нет ошибок то в Build Console будет надпись: “Build Finished”.

10. Для программирования контроллера достаточно нажать кнопку Run на панель инструментов:



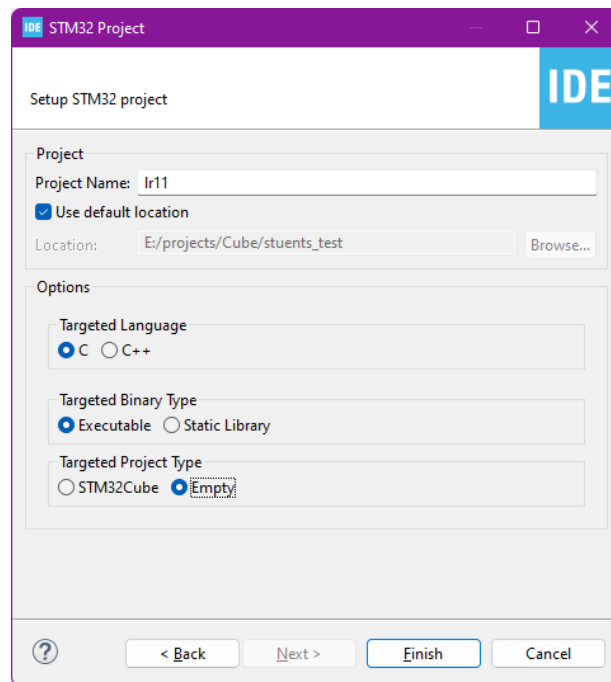
При первом запуске откроется окно конфигурации запуска, в котором необходимо выбрать (если автоматически не указан) исполняемый файл программы с расширением *.elf*.



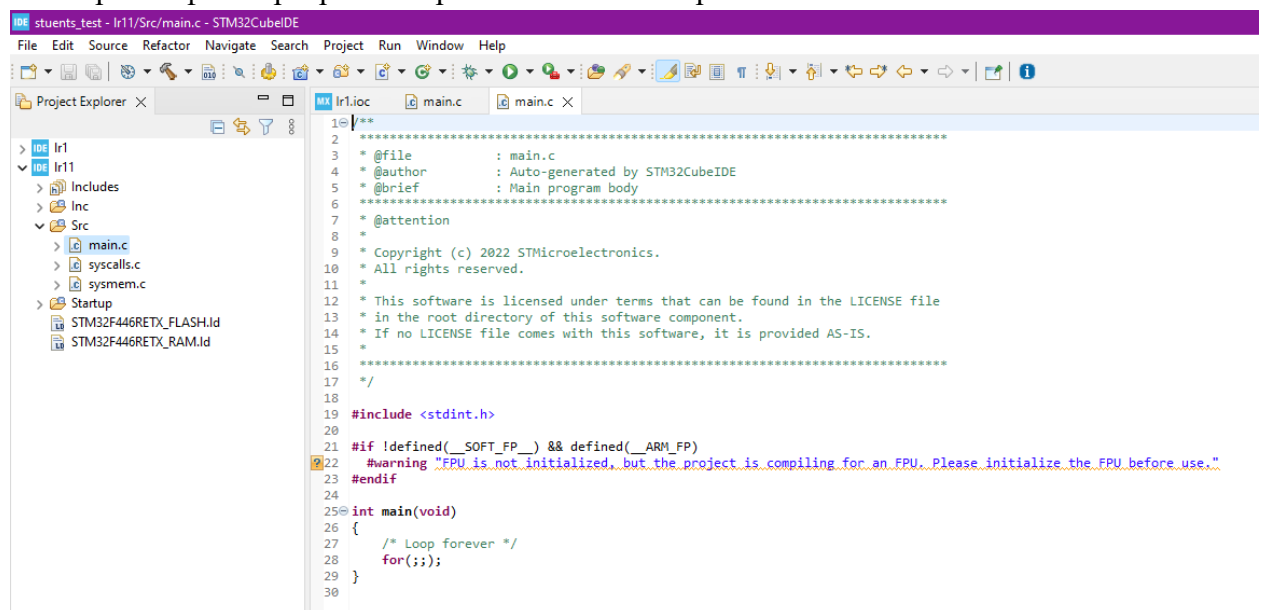
После выполнения всех действий на плате должен начать изменять свой цвет светодиод.

## *Часть II. Создание пустого проекта.*

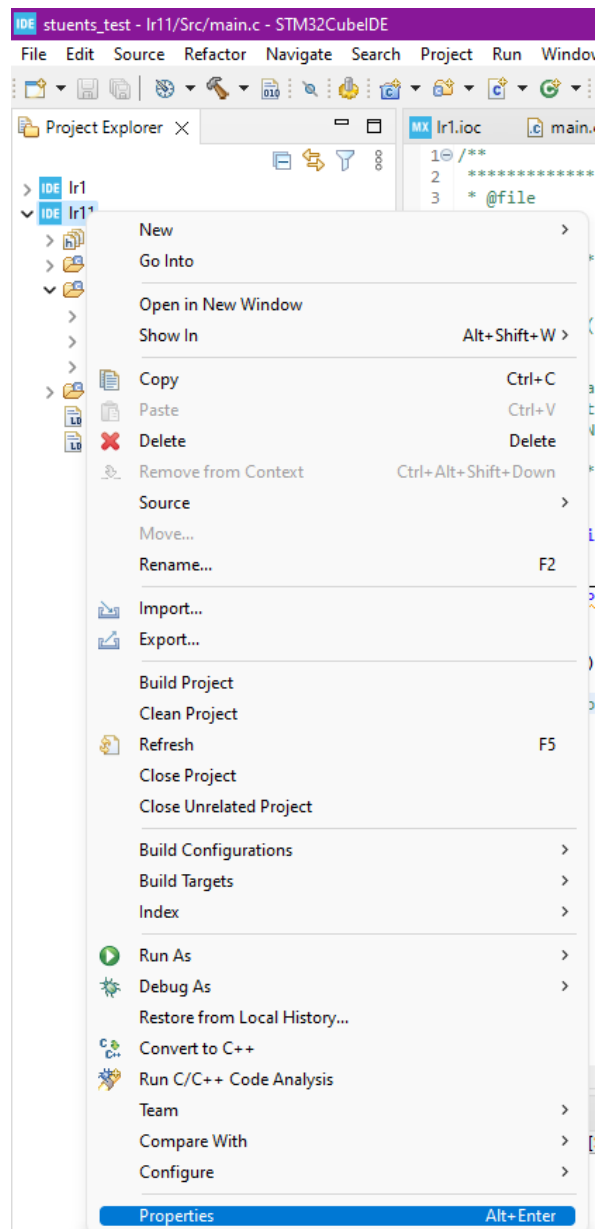
1. Создайте новый проект в рабочем пространстве выполнив пункты 2 и 3 части I.
2. В окне свойств проекта выберите пустой проект – Target Project Type: Empty, нажмите Finish.



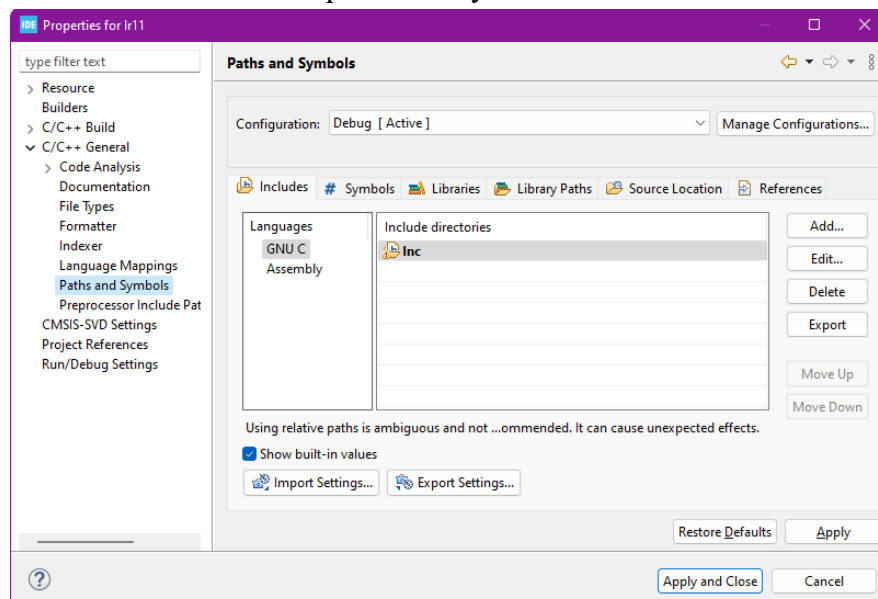
3. Откройте файл программы проекта: Название проекта -> Src -> main.c



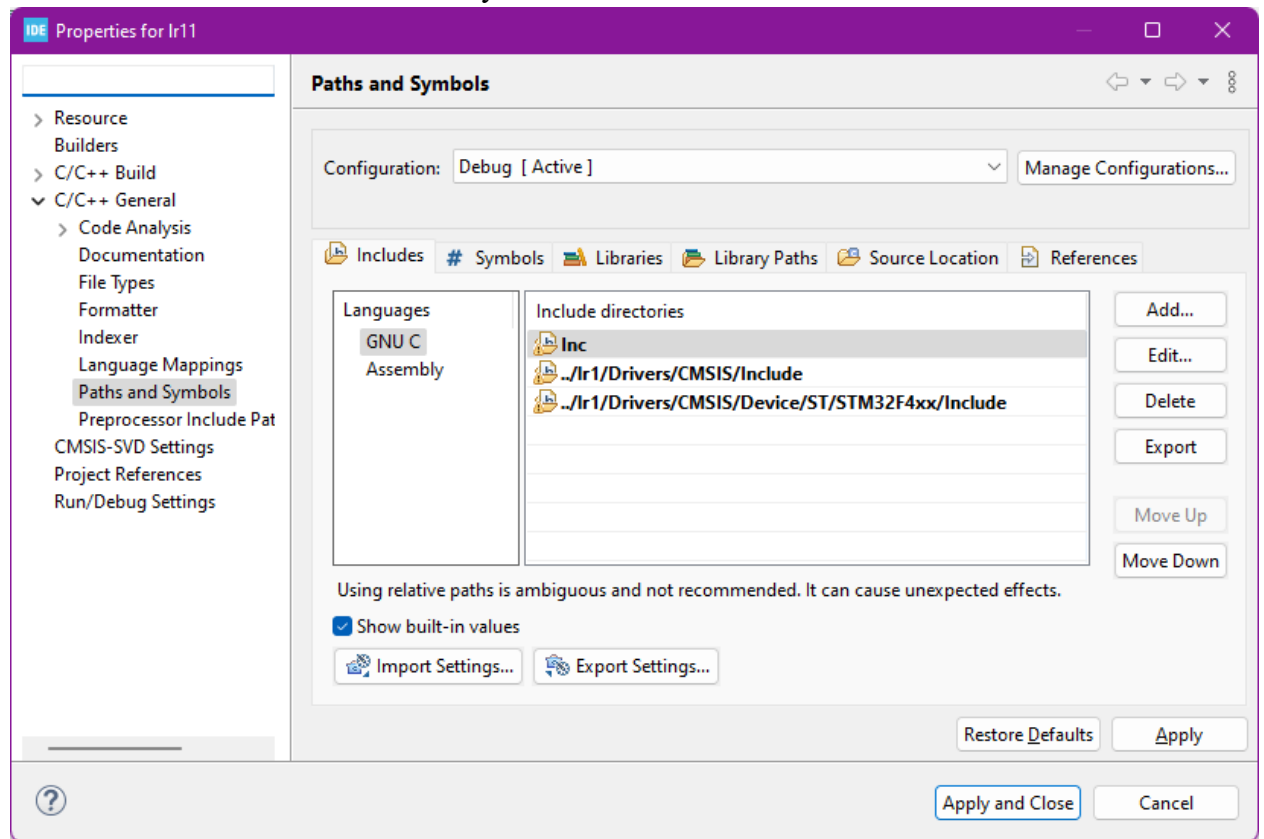
4. В данном варианте работы для взаимодействия с микроконтроллером необходимо вручную подключить библиотеки. Для этого необходимо зайти в свойства текущего проекта: Пр.кнопкой мыши на названии проекта -> Properties.



В открывшемся окне необходимо перейти по пути: C/C++ General -> Paths and Symbols.



Самый оптимальный – добавить библиотеки из проекта с использованием HAL, с использованием относительного пути:



После добавления библиотек нажать – Apply and Close.

##### 5. Написание программы:

Добавьте в main файл текст, чтобы он соответствовал изображению ниже:

```
#include <stdint.h>
#include "stm32f446xx.h"

#if !defined(__SOFT_FP__) && defined(__ARM_FP)
#warning "FPU is not initialized, but the project is compiled with FPU support"
#endif

int main(void)
{
    RCC->AHB1ENR |= RCC_AHB1ENR_GPIOBEN;
    GPIOB->MODER |= 1 << 5*2;

    /* Loop forever */
    for(;;)
    {
        GPIOB->ODR ^= GPIO_ODR_OD5;
        for(uint32_t i = 0; i < 1600000; i++);
    }
}
```

6. Для отправки программы на контроллер повторите пункты 9 и 10 части I.

**Задания**

1. Выполните все действия из части I.
2. Модифицируйте программу так, чтобы вместо пина PB5 использовался PA8.
3. Модифицируйте программу так, чтобы вначале изменялось состояние на PB5, а через задержку на PA8.
4. Выполните все действия из части II.
5. По возможности выполните задачи 2 и 3 для части II.

Все пункты заданий по очереди продемонстрируйте преподавателю.