МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра «Информационная безопасность систем и технологий»

Отчет

по лабораторной работе №1

на тему «Порты ввода-вывода микроконтроллера семейства STM32. Средства отладки и симуляции»

Дисциплина: ПМК

Группа: 21ПИ1

Выполнил: Гусев Д. А.

Количество баллов:

Дата сдачи:

Принял: Хворостухин С. П .

2024

1 Цель работы: Ознакомиться со средой разработки Keil uVision: встроеннымисредствами разработки, отладки и симуляции; освоить программные средства работы с портами ввода-вывода микроконтроллера STM32.

2 Задание на лабораторную работу.

2.1 Получить вариант задания у преподавателя.

2.2 Создать проект в среде Keil uVision5. Создать папку проекта, задать имя проекта и сохранить его в папку.

2.3 Выполнить настройку режима отладки для проекта.

2.4 Создать файлы программы.

2.5 Разработать файл сценария. Сигнальная функция должна обеспечивать изменение состояния входного порта через заданные интервалы времени. Длительность интервалов выбрать самостоятельно.

2.6 азработать программу согласно варианту задания. Программа должна быть организована в виде бесконечного цикла. Должны быть реализованы функции инициализации портов ввода-вывода, формирования очередного бита псевдослучайной последовательности, опроса состояния входного порта, установки состояния выходного порта. Опрос состояния входного порта должен выполняться с задержкой. Для реализации задержки использовать задержку, реализуемую через цикл с большим числом итераций. Длительность задержки (количество итераций в цикле) выбрать самостоятельно.

2.7 Выполнить построение проекта - кнопка «Build». При каждом изменении программных функций необходимо выполнять перестроение проекта - кнопка «Rebuild».

2.8 Выполнить симуляцию разработанной.

3 Выполнение лабораторную работы:

3.1 Был получен 8 вариант задания. Вариант представлен на рисунке 1.

Номер варианта	Полином	Входной порт	Выходной порт
8	$x^{17} + x^3 + 1$	PB7	PA5

Рисунок 1 — Вриант задания

3.2 Был запущен Keil uVision5, был выбран пункт меню «Project - New uVision Project» (рис.2).

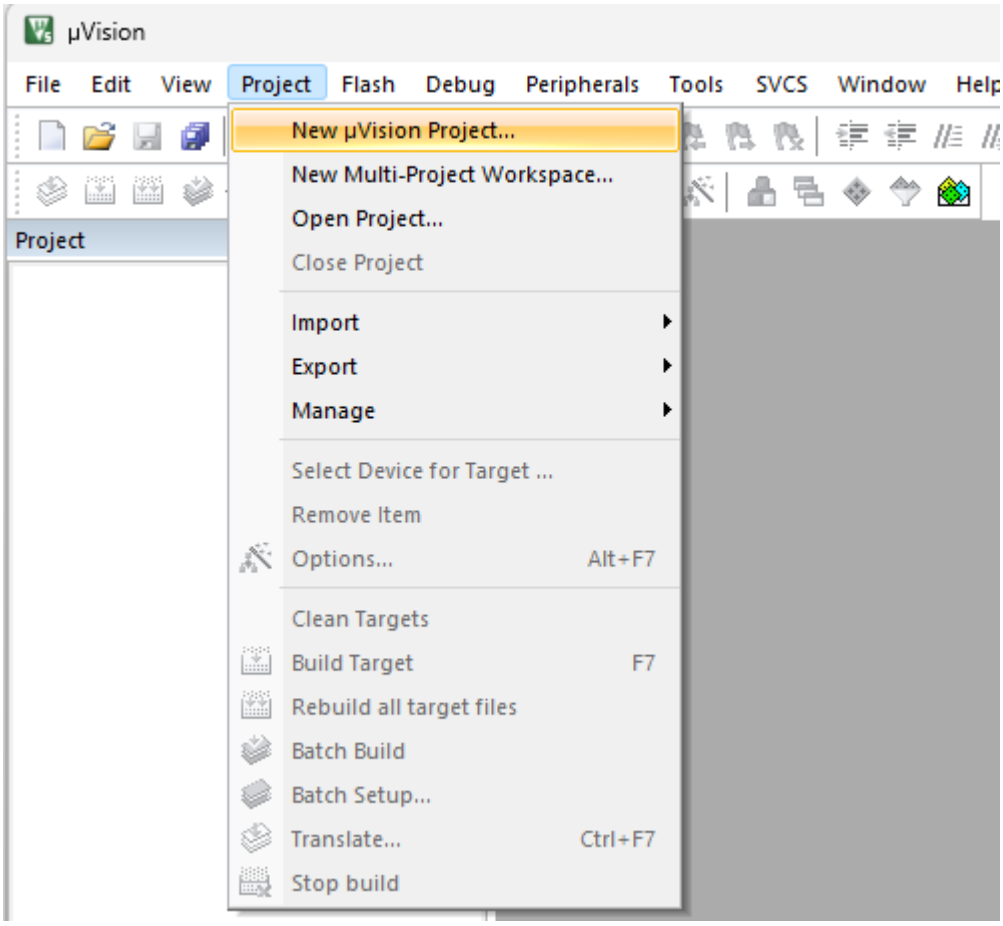

Рисунок 2 — Создание проекта

Была создана папка проекта, задано имя проекта. Была выбрана модель микроконтроллера для использования в проекте: STM32F103RB (рис.3).

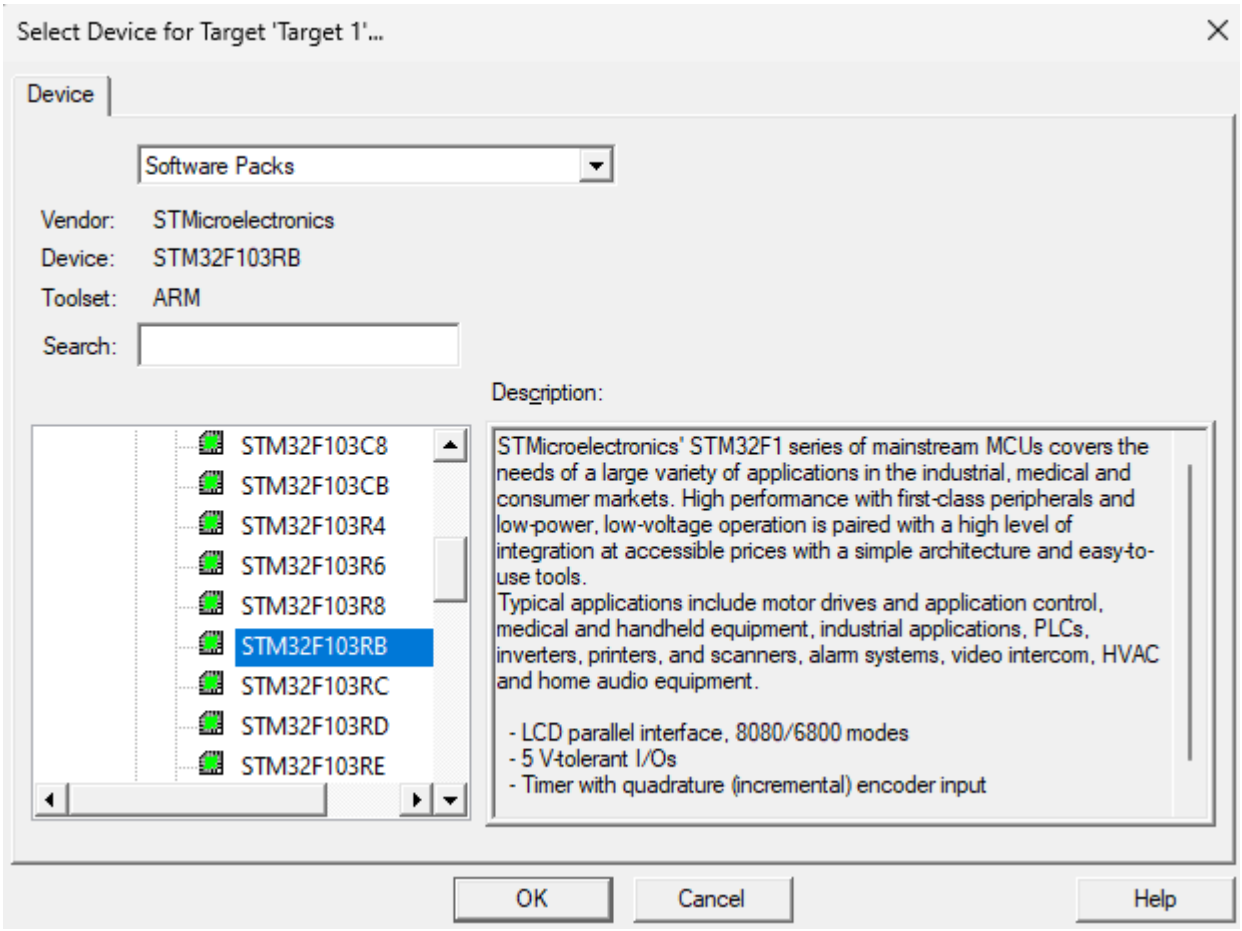

Рисунок 3 — Выбор модели микроконтроллера

Были выбраны программные компоненты: CMSIS/Core, Device/Startup, Device/StdPeriph Drivers/Framework, Device/StdPeriph Drivers/GPIO, Device/StdPeriph Drivers/RCC (рис.4).

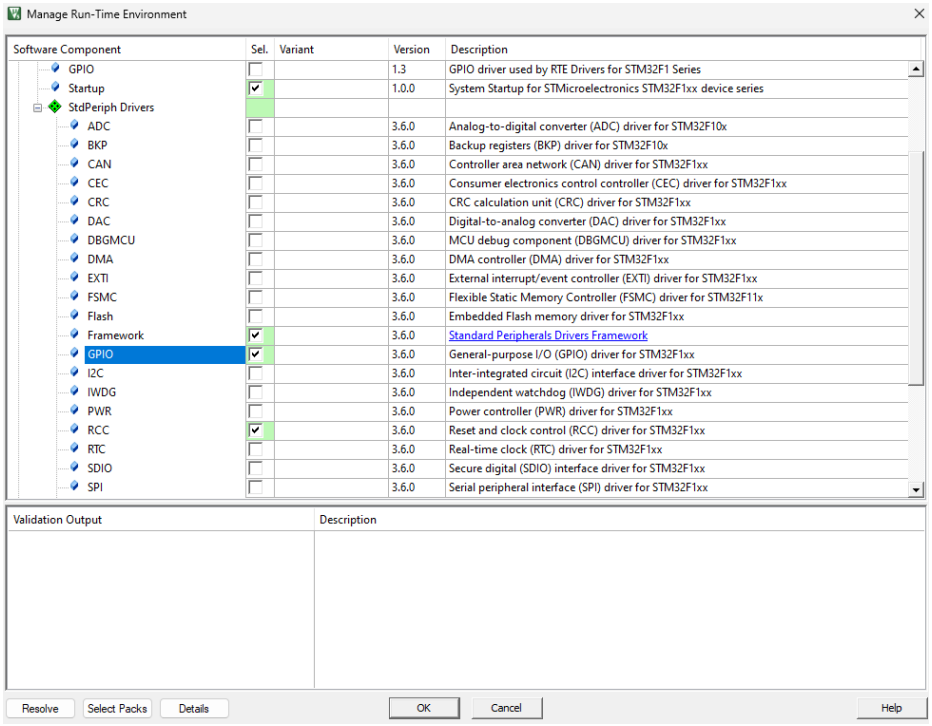

Рисунок 4 — выбор программных компонентов

3.3 Была выполнена настройка режима отладки для проекта. Необходимо сформировать в папке проекта файл MAP.ini (рис.5).

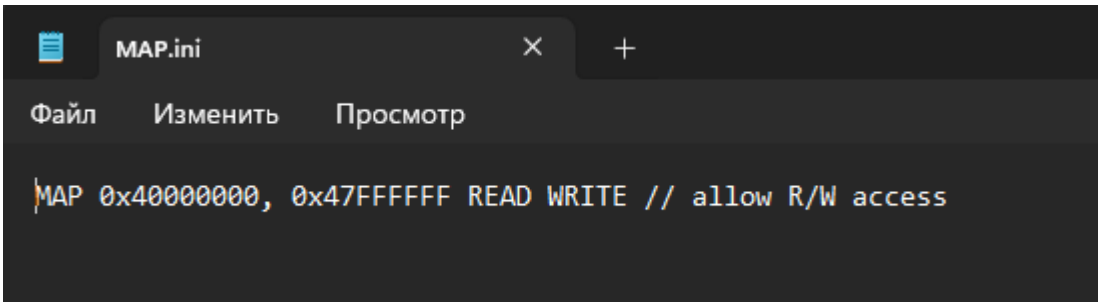

Рисунок 5 — Настройка файла map.ini

Была открыта вкладка «Option for Target…» (рис.6), было выполнено переключение на вкладку «Debug», установить переключатель «UseSimulator» для включения симуляции в режиме отладки. В поле «DialogDLL» записать DARMSTM.DLL, в поле «Parameter» -pSTM32F103RB. Установить путь к файлу MAP.ini в поле «InitializationFile». После чего завершить настройку вкладки «Debug» (рис.7).

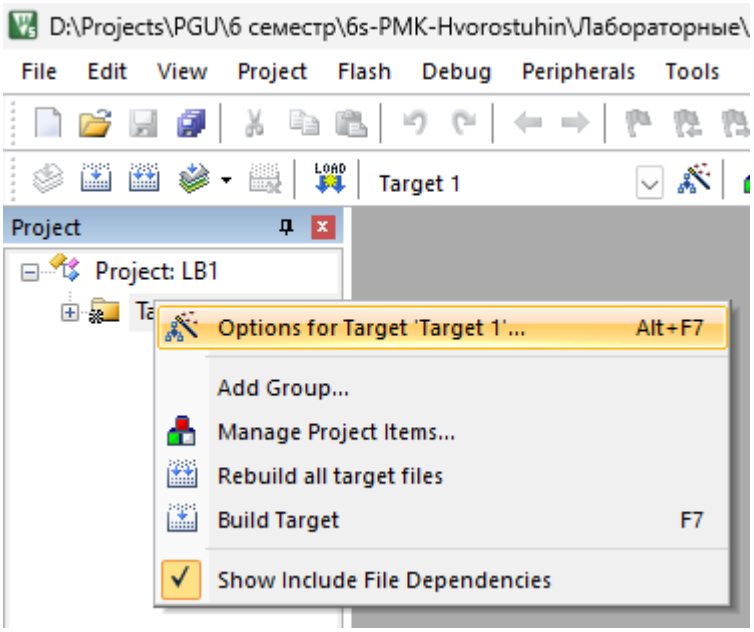

Рисунок 6 — Вкладка Option for Target…

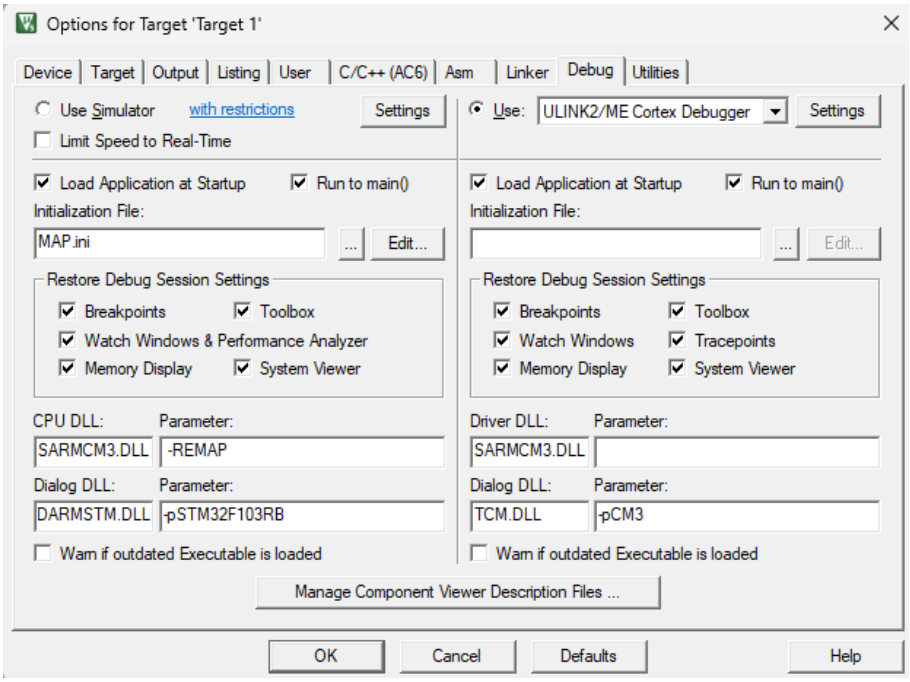

Рисунок 7 — Вкладка Debug

3.4 Были созданы файлы программы. Была создана группа пользовательских файлов «Add Group…», в нее был добавлен файл «main.c» - «Add New Item to Group…» (рис.8). В созданный файл были добавлены директивы включения библиотечных файлов и функция int main(), содержащая бесконечный цикл (рис.9).

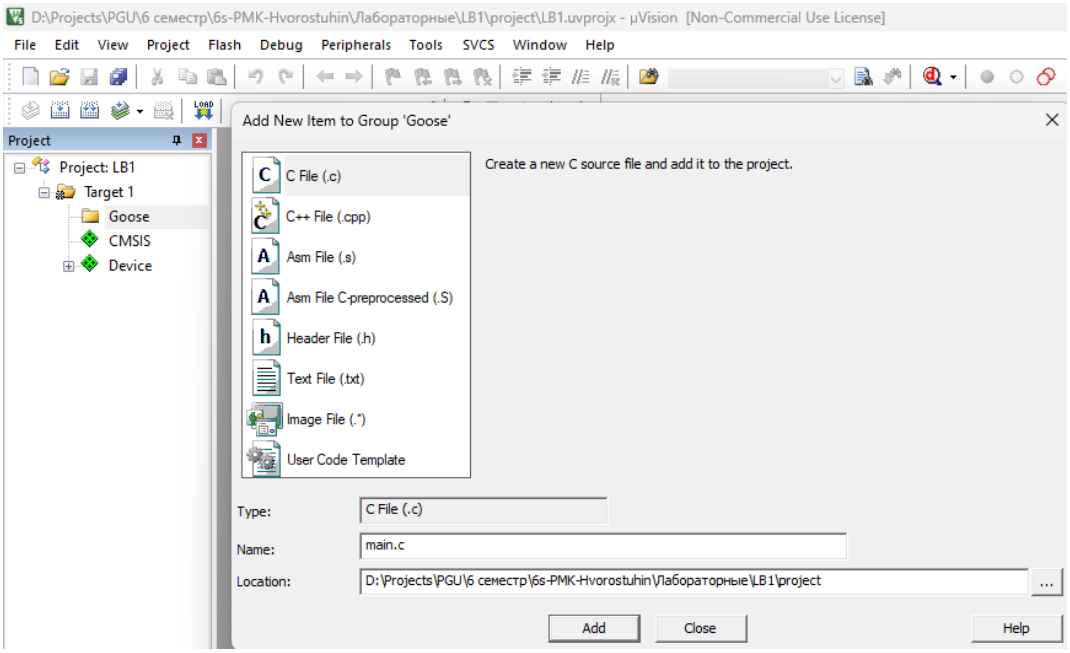

Рисунок 8 — Добавленее файла main.c

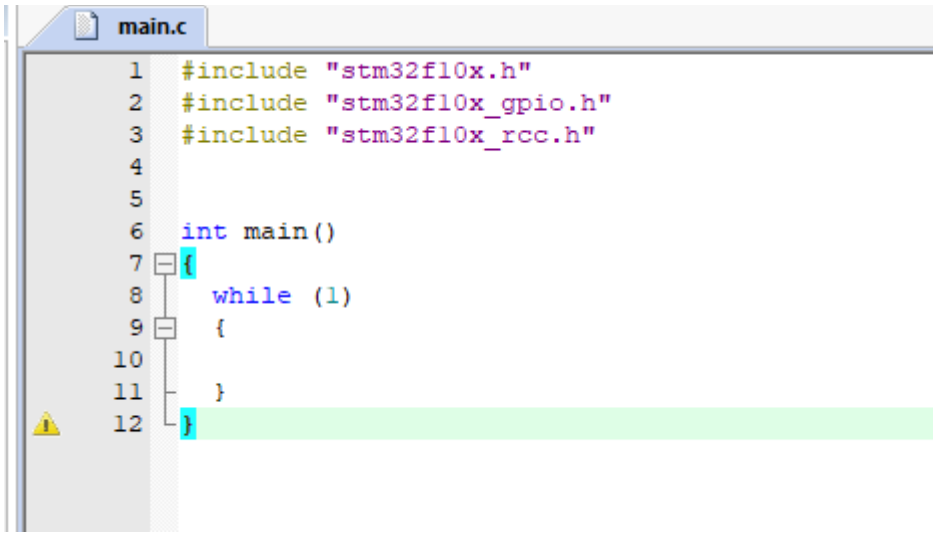

Рисунок 8 — Настройка файла main.c

3.5 Был разработан файл сценария. Код сигнальной функции input представлен ниже.

```
signal void input(void)
{
while(1)
{
        PORTB |= 0x28;
        twatch(1000);
        PORTB &= ~0x28;
```

```
        twatch(1000);
}
}
```

3.6 Была разработана программа согласно варианту задания. Код программы представлен в Приложении А.

3.7 Было выполнено построение проекта - кнопка «Build». Результат представлен на рисунке 9.

```
linking...
Program Size: Code=1636 RO-data=268 RW-data=4 ZI-data=1640
".\Objects\LB1.axf" - 0 Error(s), 13 Warning(s).
Build Time Elapsed:  00:00:00
```

Рисунок 9 — Сборка проекта

3.8 Была выполнена симуляцию разработанной программы. Запущен отладочный режим - кнопка «Start/Stop Debug Session» или сочетание клавиш 15 «Ctrl + F5». После чего добавлено отображение логического анализатора «View-Analysis Windows-Logic Analyzator» и окна просмотра данных «ViewWatch Windows-Watch1». В окно просмотра данных добавлены переменные Seed и GBit для просмотра их графика. Выведено состояние портов задействованных периферийных устройств в окно просмотра для просмотра их тактирования (PORTA и PORTB). Для подключения файла сценария введена в командную строку команда *”include signal.ini”,* а затем *input().* Результат симуляции представлен на рисунке 10.

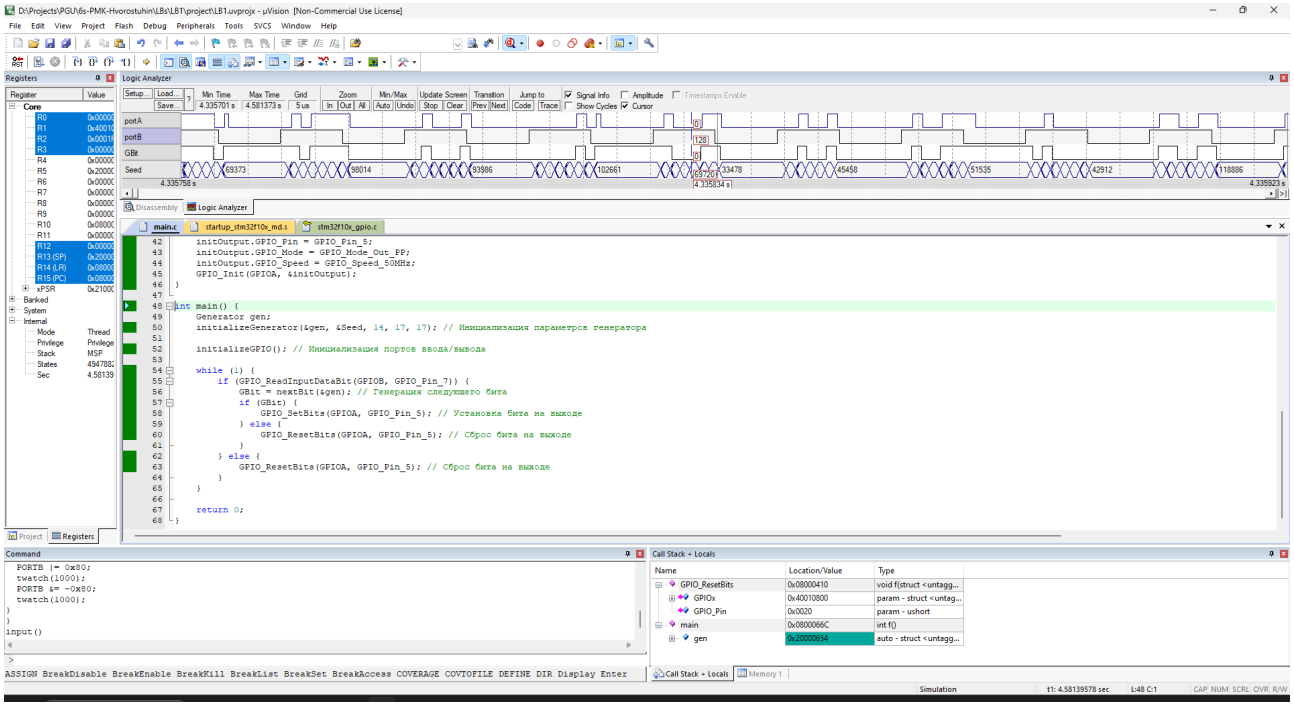

Рисунок 10 - Симуляция

4 Вывод: было выполнено ознакомление со средой разработки Keil uVision: встроеннымисредствами разработки, отладки и симуляции; были освоены программные средства работы с портами ввода-вывода микроконтроллера STM32.

Приложение А

Код main.c

```
#include "stm32f10x.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_rcc.h"

uint32_t Seed = 0xABCDEF; // Переменная для анализатора
uint32_t GBit; // Переменная для анализатора

typedef struct {
    uint32_t *polynomial;  // Указатель на переменную seed
    int id1;
    int id2;
    int length;
} Generator;

void initializeGenerator(Generator *gen, uint32_t *seed, int id1, int
id2, int length) {
    gen->polynomial = seed;
    gen->id1 = id1;
    gen->id2 = id2;
    gen->length = length;
}

uint32_t nextBit(Generator *gen) {
    uint32_t firstBit = ((*gen->polynomial) >> (gen->id1 - 1)) & 0x1;
    uint32_t secondBit = ((*gen->polynomial) >> (gen->id2 - 1)) & 0x1;
    uint32_t sum = firstBit ^ secondBit;
```

```
    *gen->polynomial = ((*gen->polynomial) << 1) | sum;
    *gen->polynomial &= ((1ULL << gen->length) - 1);

    return sum;
}

void initializeGPIO() {
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB | RCC_APB2Periph_GPIOA,
ENABLE);

    GPIO_InitTypeDef initInput;
    initInput.GPIO_Pin = GPIO_Pin_7;
    initInput.GPIO_Mode = GPIO_Mode_IPD;
    GPIO_Init(GPIOB, &initInput);

    GPIO_InitTypeDef initOutput;
    initOutput.GPIO_Pin = GPIO_Pin_5;
    initOutput.GPIO_Mode = GPIO_Mode_Out_PP;
    initOutput.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &initOutput);
}

int main() {
    Generator gen;
    initializeGenerator(&gen, &Seed, 14, 17, 17); // Инициализация
параметров генератора

    initializeGPIO(); // Инициализация портов ввода/вывода
```

```
    while (1) {
        if (GPIO_ReadInputDataBit(GPIOB, GPIO_Pin_7)) {
            GBit = nextBit(&gen); // Генерация следующего бита
            if (GBit) {
                GPIO_SetBits(GPIOA, GPIO_Pin_5); // Установка бита на
выходе
            } else {
                GPIO_ResetBits(GPIOA, GPIO_Pin_5); // Сброс бита на
выходе
            }
        } else {
            GPIO_ResetBits(GPIOA, GPIO_Pin_5); // Сброс бита на выходе
        }
    }

    return 0;
}
```