



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Компьютерные системы и сети»

Задание к лабораторной работе №1
по курсу «Основы микропроцессорной техники»

Хохлов С.А.

«Сборка и симуляция проектов для микроконтроллера Atmega328p»

2025 г.

СОДЕРЖАНИЕ

1	Сборка и симуляция проектов для микроконтроллера Atmega328p	3
1.1	Цель и задачи лабораторной работы	3
1.2	Задание	4
1.2.1	Установка программных инструментов	4
1.2.2	Создание проекта в VSCode/PlatformIO	4
1.2.3	Сборка проекта	9
1.2.4	Настройка Wokwi	9
1.2.5	Запуск моделирования	11
1.3	Вопросы	12
1.4	Самостоятельное задание	13
2	Отчет к лабораторной работе	14

1 Сборка и симуляция проектов для микроконтроллера Atmega328p

1.1 Цель и задачи лабораторной работы

Цель

Освоить основы низкоуровневого программирования микроконтроллера, научиться собирать единый проект из кода на C, C++ (Arduino) и ассемблере, понять устройство памяти и механизм линковки.

Задачи

1. Настроить среду VSCode+PlatformIO+Wokwi и собрать многофайловый проект (Arduino-скетч + C-модуль + ASM-модуль).
2. Реализовать простую функцию на ассемблере и вызвать её из кода на C/C++.
3. Исследовать работу регистров ввода-вывода (пример — переключение светодиода через PINB).
4. Сравнить задержки `delay()`, `_delay_ms()` и собственной обёртки по точности и принципу работы.
5. Измерить время выполнения кода с помощью логического анализатора и проанализировать полученные данные.

1.2 Задание

1.2.1 Установка программных инструментов

Установите на рабочем компьютере следующие программные инструменты:

1. VSCode (<https://code.visualstudio.com/>)
2. Wokwi extension для VSCode (<https://marketplace.visualstudio.com/items?itemName=Wokwi.wokwi-vscode>)
3. Platformio extension для VSCode (<https://marketplace.visualstudio.com/items?itemName=platformio.platformio-ide>)
4. VaporView для VSCode (<https://marketplace.visualstudio.com/items?itemName=lramseyer.vaporview>)

1.2.2 Создание проекта в VSCode/PlatformIO

Откройте в VSCode вкладку “PIO Home” (F1->PlatformIO Home) и создайте проект для платы Arduino UNO.

Добавьте в проект следующие файлы и код:

blink.ino

```
#include "externs.h"
#include <Arduino.h>

#define LED_PIN 13  // встроенный светодиод (PB5)

void setup() {

    // Настроим вывод стандартно через Arduino API - для наглядности
    pinMode(LED_PIN, OUTPUT);
    Serial.begin(115200);
    Serial.println("Multi-file demo: Arduino + C + ASM");
}

void loop() {
    Serial.println("toggle from ASM");
    asm_toggle_led();    // быстрое переключение, реализовано в asm
    c_delay_ms(200);     // задержка, реализована в C
}
```

c_funcs.c

```
#include <avr/io.h>
#include <util/delay.h>
#include "externs.h"

// Простая обёртка над util/delay.h
void c_delay_ms(unsigned int ms) {
    // Для учебной цели используем _delay_ms в небольших числах
    while (ms--) {
        _delay_ms(1);
    }
}
```

asm_funcs.S

```
#define PINB_ADDR 0x03
```

```
.section .text
```

```
.global asm_toggle_led
```

```
.type asm_toggle_led, @function
```

```
asm_toggle_led:
```

```
    ; На AVR запись единицы в PINx: бит
```

```
    ;                               - переключает соответствующий PORTx
```

```
    ; Для Arduino UNO встроенный LED на PB5 -> используем PINB,5
```

```
    sbi PINB_ADDR, 5
```

```
    ret
```

externs.h

```
#ifndef EXTERNS_H
#define EXTERNS_H

#ifdef __cplusplus
extern "C" {
#endif

void c_delay_ms(unsigned int ms);           // реализовано в c_funcs.c
void asm_toggle_led(void);                  // реализовано в asm_funcs.S

#ifdef __cplusplus
}
#endif

#endif // EXTERNS_H
```

1.2.3 Сборка проекта

F1->PlatformIO: Build

1.2.4 Настройка Wokwi

1. Создайте проект на Arduino Uno на сайте <https://wokwi.com>:

Start from scratch->Arduino UNO

2. Создайте в проекте в VSCode файл `diagram.json` и перенесите в него содержимое файла с таким же названием из вашего проекта в Wokwi.
3. Добавьте в проект в VSCode файл `wokwi.toml` со следующим содержанием

```
[wokwi]
version = 1
firmware = '..\\.pio\\build\\uno\\firmware.hex'
elf = '..\\.pio\\build\\uno\\firmware.elf'
```

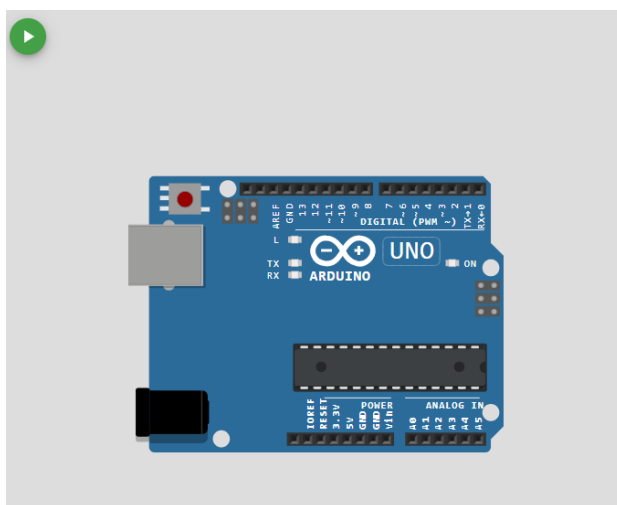


Рисунок 1.1 – Окно симуляции Wokwi в VSCode

1.2.5 Запуск моделирования

Откройте двойным щелчком файл `diagram.json`, должно запускиться расширение Wokwi (см. рисунок 1.1).

Запустите симуляцию зеленой кнопкой, наблюдайте переключение светодиода и вывод в терминал.

1.3 Вопросы

1. Что содержится в файле `wokwi.toml` ?
2. Что выполняет команда `sbi PINB_ADDR, 5`? Для подтверждения ответа приведите цитату из даташита на микроконтроллер.
3. Для чего служит файл `externs.h`?
4. Почему в `c_funcs.c` используют `_delay_ms(1)` в цикле, а не сразу `_delay_ms(ms)`?
5. Как отличаются задержки `delay()`, `_delay_ms()` и собственной обёртки по точности и принципу работы?

1.4 Самостоятельное задание

Сравните реализацию переключения пина `asm_funcs.S` с ардуино-реализацией `digitalWrite()`.

Выполните профилирование обеих функций.

Для этого в редакторе на сайте <https://wokwi.com> добавьте в проект логический анализатор и подключите его к любому цифровому выходу платы ArduinoUno, а затем перенесите код из `diagram.json` в ваш проект VSCode (аналогично тому, как было сделано ранее в п. 1.2.4).

Перепишите код вашей программы таким образом, чтобы до вызова профилируемой функции и после неё вызывалась функция включения/выключения пина, подключенного к логическому анализатору.

После завершения моделирования VSCode предложит сохранить VCD-файл с результатами моделирования.

Сохраните файл, его можно просмотреть с помощью расширения VaporView.

Проанализируйте время выполнения профилируемых функций.

2 Отчет к лабораторной работе

Составьте отчет к лабораторной работе.

Требования:

1. Описание лабораторной работы и основных этапов её выполнения, с соответствующими скриншотами.
2. Ответы на вопросы.
3. Выполнение самостоятельного задания со скриншотами и результатами измерений.
4. Выводы к лабораторной работе.