МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

Институт компьютерных наук и технологий

Высшая школа искусственного интеллекта

Отчёт по лабораторной работе №2 по дисциплине

ПРОГРАММИРОВАНИЕ МИКРОКОНТРОЛЛЕРОВ

на тему:

Использование библиотек в Keil μVision5

Работу выполнил:

студент группы 5130201/20001

Тугай В.В.

Преподаватель:

Вербова Н. М.

Санкт-Петербург

2024

# Содержание

1. Цель
2. Постановка задачи
3. Алгоритм программы
4. Полученные результаты
5. Вывод
6. Приложение

**Цель**

Ознакомление с основными приемами работы с документацией при составлении программ для микроконтроллеров. Приобретение навыков работы с осциллографом и оценочной платой MCBSTM32F200 в качестве измерительного генератора.

**Постановка задачи**

Используя библиотеки Keil μVision5, разработать программу для микроконтроллера (МК) STM32F200, которая включает и выключает светодиод.

**Алгоритм программы**

Сперва запускается процесс, состоящий из четырех итераций, направленных на подготовку микроконтроллера к функционированию. Управление тактированием периферийных блоков осуществляется путем активации соответствующего тактирования через регистр периферии RCC. Необходимые биты в этом регистре устанавливаются в соответствии с номерами блоков (См. Рисунок 1).

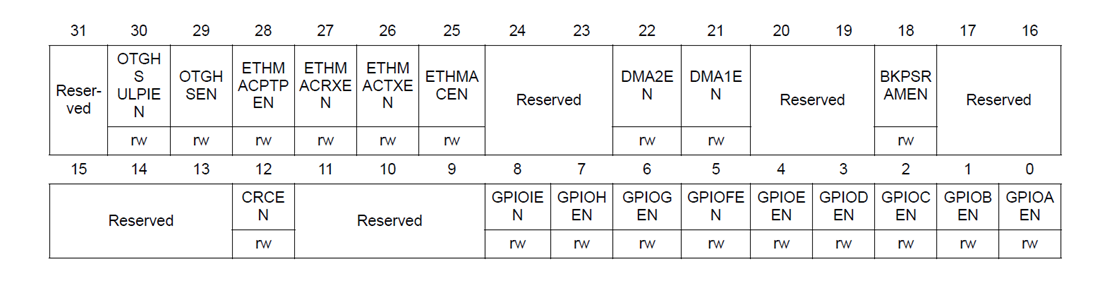


Рисунок 1

Управление режимом работы GPIO блока: изменение режимов работы разрядов портов ввода/вывода общего назначения осуществляется с использованием регистра режимов. По адресу конкретного разряда, определяемого по таблице (см. Рисунок 2), устанавливаются соответствующие биты, задающие режим работы.

A screenshot of a computer

Description automatically generated

Рисунок 2

Для каждой лампочки в блоке выделяется по два бита, таким образом, крайние "правые" два бита отвечают за состояние нулевой лампочки, следующие два за первую и так далее (см. Рисунок 3).

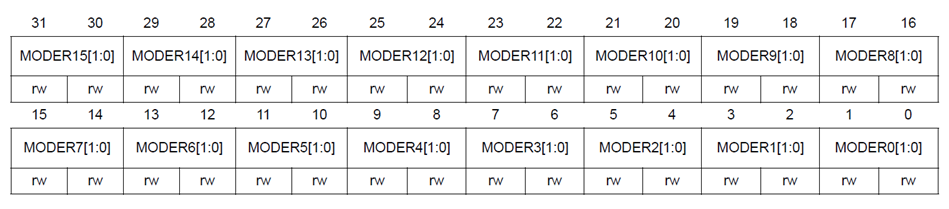


Рисунок 3

Установка необходимых значений на выводе микроконтроллера (МК) происходит путем записи "0" или "1" в соответствующие биты с 0 по 15, что приводит к соответствующему изменению состояния выводов порта. Для установки конкретного значения на одном или нескольких выводах МК без изменения состояний остальных используются булевы операции, специально предназначенные для работы с отдельными битами, как уже показано выше. Например, чтобы записать "1" в бит PG7, необходимо записать значение 0х80 в соответствующий регистр. Номер бита, который необходимо изменить, определяется по таблице (см. Рисунок 4).

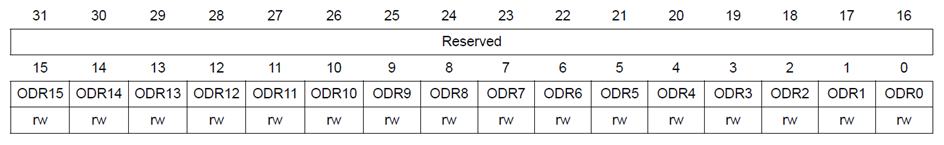


Рисунок 4

Для достижения эффекта последовательного и непрерывного включения и выключения диодов в цикле while (условие которого всегда верно) выполняются следующие шаги: сначала устанавливается "1" в бит регистра, соответствующий левому диоду на микроконтроллере. Затем происходит ожидание в течение одного миллиона итераций, чтобы обеспечить паузу и предотвратить мгновенное выключение диода. После этого "0" устанавливается в тот же бит регистра, а в бит регистра, соответствующий следующему диоду, устанавливается "1". Подобные операции выполняются для всех диодов до последнего. Затем следует пауза в два миллиона итераций с помощью соответствующего цикла, и новая итерация цикла while.

**Полученные результаты**

Реализованная на языке C программа (см. Приложение) успешно выполняет поставленную задачу.

**Вывод**

В результате выполнения лабораторной работы был создан проект в среде Keil µVision, написана программа на языке C++ и проведен анализ сигнала с помощью асциллографа. Программа реализует последовательное включение и выключение диода G7 на микроконтроллере STM32F200. Была получена характеристика частоты сигнала, размах, период, время нарастания и спада. Было выяснено, что время нарастания сигнала примерно в два раза меньше времени спада сигнала.

**Приложение**

#include "stm32f2xx.h" // Device header

void delay ()

{

unsigned long i; // Counter for blinky delay

i=0;

for(i=0; i<2000000; i++){} // Delay

}

int main ()

{

RCC->AHB1ENR |= 1ul<<6; // Enable port G clocking

GPIOG->MODER = (GPIOG->MODER & ~1ul<<15) | 1ul<<14;

for (;;)

{

GPIOG->ODR |= 1ul<<7;

delay ();

GPIOG->ODR &= ~1ul<<7;

delay ();

}

}