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 №1 по дисциплине

ПРОГРАММИРОВАНИЕ МИКРОКОНТРОЛЛЕРОВ

на тему:

Создание нового проекта 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**

Ознакомится с основными приемами работы с документацией при составлении программ для микроконтроллеров.

**Постановка задачи**

Создать новый проект в Keil μVision5 и разработать программу для микроконтроллера (МК) STM32F200, которая попарное включает и выключает диоды на сверху вниз.

**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).

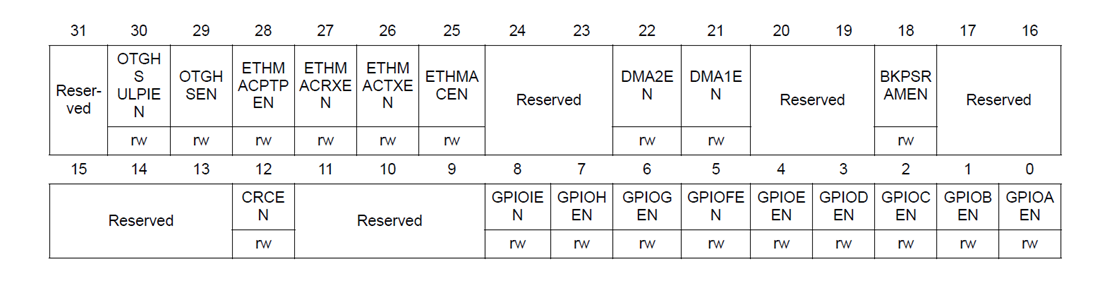


Рисунок 1

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).

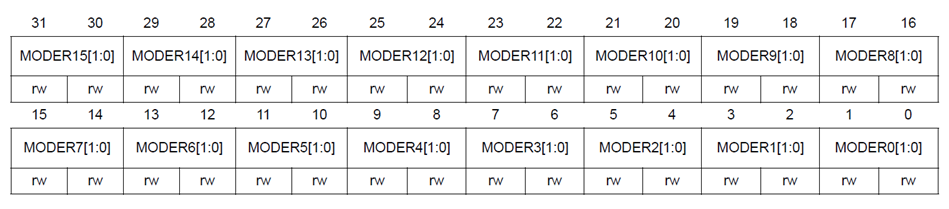


Рисунок 3

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).

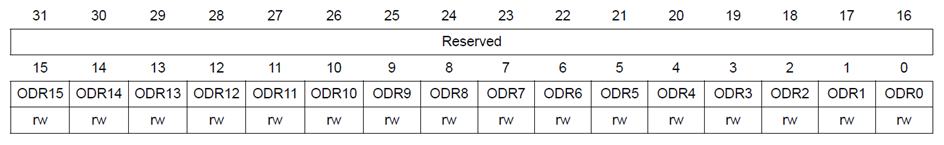


Рисунок 4

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, а промежуток между переключениями составляет примерно 0.5 секунды.

**Вывод**

В результате выполнения лабораторной работы был разработан проект в интегрированной среде разработки Keil µVision, а также реализована программа на языке программирования C (см. Приложение). Данная программа обеспечивает последовательное попарное включение и выключение светодиодов на микроконтроллере STM32F200 сверху вниз с определенным временным интервалом между переключениями. Важно подчеркнуть, что работа с адресами реализована без применения макросов, что хотя и обеспечивает функциональность, но и не является оптимальным подходом с точки зрения читаемости и поддержки программного кода.

**Приложение**

int main (){

int i;

unsigned long int j;

i=0;

j=0;

\*(unsigned long\*)(0x40023830) |= 0x40; //port G clocking

\*(unsigned long\*)(0x40023830) |= 0x80; //port H clocking

\*(unsigned long\*)(0x40023830) |= 0x100; //port I clocking

for(i=0; i<4; i++){} //small delay for GPIOG get ready

\*(unsigned long\*)(0x40021C00) = (\*(unsigned long\*)(0x40021C00) & (~0x00000080)) | (0x00000040); //Set PH3

\*(unsigned long\*)(0x40021C00) = (\*(unsigned long\*)(0x40021C00) & (~0x00002000)) | (0x00001000); //Set PH6

\*(unsigned long\*)(0x40021800) = (\*(unsigned long\*)(0x40021800) & (~0x00002000)) | (0x00001000); //Set PG6

\*(unsigned long\*)(0x40021C00) = (\*(unsigned long\*)(0x40021C00) & (~0x00008000)) | (0x00004000); //Set PH7

\*(unsigned long\*)(0x40022000) = (\*(unsigned long\*)(0x40022000) & (~0x00200000)) | (0x00100000); //Set PI10

\*(unsigned long\*)(0x40021800) = (\*(unsigned long\*)(0x40021800) & (~0x00008000)) | (0x00004000); //Set PG7

\*(unsigned long\*)(0x40021800) = (\*(unsigned long\*)(0x40021800) & (~0x00020000)) | (0x00010000); //Set PG8

\*(unsigned long\*)(0x40021C00) = (\*(unsigned long\*)(0x40021C00) & (~0x00000020)) | (0x00000010); //Set PH2

while(1){

\*(unsigned long\*)(0x40021C14) |= 0x08; //PH3 on

\*(unsigned long\*)(0x40021C14) |= 0x40; //TPH6 on

for( j=0; j<1000000 ;j++ ){} //Delay

\*(unsigned long\*)(0x40021C14) &= ~0x08; //PH3 off

\*(unsigned long\*)(0x40021C14) &= ~0x40; //PH6 off

for( j=0; j<1000000 ;j++ ){} //Delay

\*(unsigned long\*)(0x40021C14) |= 0x80; //PH7 on

\*(unsigned long\*)(0x40022014) |= 0x400; //PI10 on

for( j=0; j<1000000 ;j++ ){} //Delay

\*(unsigned long\*)(0x40021C14) &= ~0x80; //PH7 off

\*(unsigned long\*)(0x40022014) &= ~0x400; //PI10 off

for( j=0; j<1000000 ;j++ ){} //Delay

\*(unsigned long\*)(0x40021814) |= 0x40; //PG6 on

\*(unsigned long\*)(0x40021814) |= 0x80; //PG7 on

for( j=0; j<1000000 ;j++ ){} //Delay

\*(unsigned long\*)(0x40021814) &= ~0x40; //PG6 off

\*(unsigned long\*)(0x40021814) &= ~0x80; //PG7 off

for( j=0; j<1000000 ;j++ ){} //Delay

\*(unsigned long\*)(0x40021814) |= 0x100; //PG8 on

\*(unsigned long\*)(0x40021C14) |= 0x04; //PH2 on

for( j=0; j<1000000 ;j++ ){} //Delay

\*(unsigned long\*)(0x40021814) &= ~0x100; //PG8 off

\*(unsigned long\*)(0x40021C14) &= ~0x04; //PH2 off

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

}

}