Санкт-Петербургский Государственный Политехнический Университет

Институт Компьютерных Наук и Технологий

ОТЧЕТ

По лабораторной работе №1

Тема:

**Создание нового проекта в Keil µVision5**

Выполнил: Нгуен Куок Дат  
группы 3530201/10001

Преподаватель: Вербова Н.М.

# Цель и постановка задачи

## 1. Цель работы:

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

## 2. Постановка задачи.

2.1. Установить среду разработки KEIL с необходимыми библиотеками.

2.2. Создать проект в среде разработки и создать главный исполняющий файл.

2.3. На языке Cи проверить программу для мигания 1 светодиода на выходе PG7

2.4. Реализовать программу для мигания светодиодов отладочной платы в последовательном порядке.

# Выполнение задания

1. Согласно методическим указаниям, я установил среду разработки KEIL Vision5 и все необходимые библиотеки.

2. Я создал проект в среде, затем создал исполняемый файл BlinkyLed.c

3. Проверил согласно методическим указаниям программу, представленную в работе.

4. Дополнил пример кода своей реализацией работы с другими светодиодами.

Код программы:

/\*------------------------------------------------------------

\* Name: Blinky.c

\* Purpose: LED PG7 Flasher for MCBSTM32F200

\*----------------------------------------------------------\*/

/\*------------------------------------------------------------

Main function

\*----------------------------------------------------------\*/

#define RCCAHB1\_ENR (\*(unsigned long\*)0x40023830)

#define RCCAHB1\_PORTG 0x40

#define GPIOG\_MODER (\*(unsigned long\*)0x40021800)

#define GPIOH\_MODER (\*(unsigned long\*)0x40021C00)

#define GPIOI\_MODER (\*(unsigned long\*)0x40022000)

#define GPIOG\_ODR \*(unsigned long\*)(0x40021814)

#define GPIOH\_ODR \*(unsigned long\*)(0x40021C14)

#define GPIOI\_ODR \*(unsigned long\*)(0x40022014)

unsigned long int j;

void delay(){

for( j=0; j<200000 ;j++ ){} //Delay

}

int main ()

{

//-------Declaration of type of variables---------------------

int i; //counter for get ready delay

//counter for blinky delay

//-------Initialization of variables--------------------------

i=0;

//-------Main cycle of algorithm------------------------------

RCCAHB1\_ENR |= 0x40; //Enable port G clocking

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

RCCAHB1\_ENR |= 0x80;//Enable port H clocking

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

RCCAHB1\_ENR |= 0x100;//Enable port I clocking

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

GPIOG\_MODER = (GPIOG\_MODER & ~(1ul<<15)) | (1ul<<14); //Set PG7 as GPO

GPIOG\_MODER = (GPIOG\_MODER &~(1ul<<13)) | (1ul<<12); //Set PG6 as GPO

GPIOG\_MODER = (GPIOG\_MODER & ~(1ul<<17)) | (1ul<<16); //Set PG8 as GPO

GPIOH\_MODER = (GPIOH\_MODER & ~(1ul<<5)) | (1ul<<4); //Set PH2 as GPO

GPIOH\_MODER = (GPIOH\_MODER & ~(1ul<<7)) | (1ul<<6); //Set PH3 as GPO

GPIOH\_MODER = (GPIOH\_MODER & ~(1ul<<13)) | (1ul<<12); //Set PH6 as GPO

GPIOH\_MODER = (GPIOH\_MODER & ~(1ul<<15)) | (1ul<<14); //Set PH7 as GPO

GPIOI\_MODER = (GPIOI\_MODER & ~(1ul<<21)) | (1ul<<20); //Set PI10 as GPO

for(i=0; i<4; i++){}

while(1)

{

//PH3

GPIOH\_ODR |= (1ul<<3); //Turn LED ON!

delay();

//PH6

GPIOH\_ODR |= (1ul<<6); //Turn LED ON!

delay();

//PH7

GPIOH\_ODR |= (1ul<<7); //Turn LED ON!

delay();

//PI10

GPIOI\_ODR |= (1ul<<10); //Turn LED ON!

delay();

//PG6

GPIOG\_ODR |= (1ul<<6); //Turn LED ON!

delay();

//PG7

GPIOG\_ODR |= (1ul<<7); //Turn LED ON!

delay();

//PG7

GPIOG\_ODR |= (1ul<<7); //Turn LED ON!

delay();

//PG8

GPIOG\_ODR |= (1ul<<8); //Turn LED ON!

delay();

//PH2

GPIOH\_ODR |= (1ul<<2); //Turn LED ON!

delay();

//PH3

GPIOH\_ODR &= ~(1ul<<3); //Turn LED OFF!

delay();

//PH6

GPIOH\_ODR &= ~(1ul<<6); //Turn LED OFF!

delay();

//PH7

GPIOH\_ODR &= ~(1ul<<7); //Turn LED OFF!

delay();

//PI10

GPIOI\_ODR &= ~(1ul<<10); //Turn LED OFF!

delay();

//PG6

GPIOG\_ODR &= ~(1ul<<6); //Turn LED OFF!

delay();

//PG7

GPIOG\_ODR &= ~(1ul<<7); //Turn LED OFF!

delay();

//PG7

GPIOG\_ODR &= ~(1ul<<7); //Turn LED OFF!

delay();

//PG8

GPIOG\_ODR &= ~(1ul<<8); //Turn LED OFF!

delay();

//PH2

GPIOH\_ODR &= ~(1ul<<2); //Turn LED OFF!

delay();

}

}

# Вывод

Вывод:  
Выполнив все задания работы, можно сделать следующие выводы:

На отладочной плате для каждого вывода используется свои регистры с отведенными адресами, при работе с ними мы оперируем битами и битовыми операциями. Для оптимизации работы программы можно объединять битовые операции с 1 регистром вместе, но так возможна ухудшение читабельности кода. Но в рамках ограниченных ресурсов памяти, следует делать более компактные программы и более точно оперировать ресурсами микроконтроллера.

# 

## 