САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. ПЕТРА ВЕЛИКОГО

Институт компьютерных наук и технологий
Высшая школа искусственного интеллекта
Направление 3.02.01 Математика и Компьютерные науки

Отчёт по дисциплине Программирование микроконтроллеров.

Лабораторная работа №1.

Работу выполнила:
Гусева С.А.
студент группы 3530201/10001
Проверила:
Вербова Н. М.

Санкт-Петербург - 2023 г.

Тема:
Создание нового проекта в Keil µVision5.

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 включает и выключает светодиод.

Алгоритм программы:

Сперва запускается цикл из четырех итераций, чтобы подготовить микроконтроллер к работе.

Управление тактированием периферийных блоков: происходит включение тактирования периферийных блоков с помощью регистра периферии RCC. В регистр устанавливаются биты, соответствующие номеру блока (номера всех блоков представлены на рис.1).

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reser-ved	OTGHS ULPIEN	OTGHSEN	ETHMACPTPEN	ETHMACRXEN	ETHMACTXEN	ETHMACEN	Reserved		DMA2EN	DMA1EN	Reserved		BKPSRAMEN	Reserved	
	rw	rw	rw	rw	rw	rw			rw	rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CRCEN	Reserved			GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
			rw				rw	rw	rw	rw	rw	rw	rw	rw	rw

Рис.1

Управление режимом работы блока GPIO: управление режимами работы разрядов портов ввода/вывода общего назначения выполняется с помощью регистра режимов. По адресу конкретного разряда, который определяется по таблице (часть таблицы приведена на рис.3), устанавливаются биты, определяющие режим работы. На каждую лампочку в блоке отводится два бита, таким образом крайние «правые» два бита отвечают за состояние нулевой лампочки, следующие два бита за состояние первой и тд (рис.2).

```
*(unsigned long*)(0x40021800) = (*(unsigned long*)(0x40021800) & (~0x00002000)) | (0x00001000); //Set PG6 as General purpose output
```

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

рис.2

0x4002 3C00 - 0x4002 3FFF	Flash interface register	AHB1	See Flash programming manual
0x4002 3800 - 0x4002 3BFF	RCC		*Section 5.3.24: RCC register map on page 135*
0x4002 3000 - 0x4002 33FF	CRC		*Section 3.4.4: CRC register map on page 61*
0x4002 2000 - 0x4002 23FF	GPIOI		*Section 6.4.11: GPIO register map on page 156*
0x4002 1C00 - 0x4002 1FFF	GPIOH		
0x4002 1800 - 0x4002 1BFF	GPIOG		
0x4002 1400 - 0x4002 17FF	GPIOF		
0x4002 1000 - 0x4002 13FF	GPIOE		
0X4002 0C00 - 0x4002 0FFF	GPIOD		
0x4002 0800 - 0x4002 0BFF	GPIOC		
0x4002 0400 - 0x4002 07FF	GPIOB		
0x4002 0000 - 0x4002 03FF	GPIOA		

Рис.3

Установка требуемых значений на выводе МК: запись «0» или «1» в биты 0-15 приводят к соответствующему изменению состояния выводов порта. Для того чтобы установить определенное значение на выходе одного или нескольких выводов МК и не изменить состояния остальных, будем использовать булевы операции специально предназначенные для работы с отдельными битами так, как это уже показывалось выше. Например: для записи «1» в бит PG7 в регистр следует записать 0x80. Определение номера бита, который необходимо изменить происходит по таблице (рис.4).

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Рис.4

Для того чтобы диоды загорались и гасли последовательно и непрерывно в цикле while (аргумент которого всегда истинен) происходят следующие действия: сначала «1» устанавливается в бит регистра соответствующий левому диоду на микроконтроллере, далее проходит цикл в один миллион итераций, чтобы выдержать паузу и диод не погас сразу, далее в тот же бит устанавливается «0», а в бит регистра соответствующий следующему диоду устанавливается «1», аналогичные действия продолжаются до последнего диода, после чего с помощью цикла на два миллиона итераций выдерживается пауза. Далее эти действия повторяются.

Полученные результаты:

```
int main ()
{
//-------Declaration of type of variables----------------------
int i; //counter for get ready delay
unsigned long int j; //counter for blinky delay
//-------Initialization of variables-------------------------
i=0;
j=0;
//-------Main cycle of algorithm-----------------------------

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

*(unsigned long*)(0x40023830) |= 0x40; //Enable port G clocking
*(unsigned long*)(0x40023830) |= 0x80; //Enable port H clocking
*(unsigned long*)(0x40023830) |= 0x100; //Enable port I clocking

 *(unsigned long*)(0x40021800) = (*(unsigned long*)(0x40021800) & (~0x00002000)) | (0x00001000); //Set PG6 as General purpose output
 *(unsigned long*)(0x40021800) = (*(unsigned long*)(0x40021800) & (~0x00008000)) | (0x00004000); //Set PG7 as General purpose output
 *(unsigned long*)(0x40021800) = (*(unsigned long*)(0x40021800) & (~0x00020000)) | (0x00010000); //Set PG8 as General purpose output

 *(unsigned long*)(0x40021C00) = (*(unsigned long*)(0x40021C00) & (~0x00000020)) | (0x00000010); //Set PH2 as General purpose output
 *(unsigned long*)(0x40021C00) = (*(unsigned long*)(0x40021C00) & (~0x00000080)) | (0x00000040); //Set PH3 as General purpose output
 *(unsigned long*)(0x40021C00) = (*(unsigned long*)(0x40021C00) & (~0x00002000)) | (0x00001000); //Set PH6 as General purpose output!
 *(unsigned long*)(0x40021C00) = (*(unsigned long*)(0x40021C00) & (~0x00008000)) | (0x00004000); //Set PH7 as General purpose output!

*(unsigned long*)(0x40022000) = (*(unsigned long*)(0x40022000) & (~0x00200000)) | (0x00100000); //Set PI10 as General purpose output!

while(1)
 {
*(unsigned long*)(0x40021C14) |= 0x04; //Turn LED ON! PH2
for( j=0; j<1000000 ;j++ ){} //Delay
*(unsigned long*)(0x40021C14) &= ~0x04; //Turn LED OFF PH2

*(unsigned long*)(0x40021814) |= 0x100; //Turn LED ON! PG8
for( j=0; j<1000000 ;j++ ){} //Delay
```

```
*(unsigned long*)(0x40021814) &= ~0x100; //Turn LED OFF PG8

*(unsigned long*)(0x40021814) |= 0x80; //Turn LED ON! PG7
for( j=0; j<1000000 ;j++ ){} //Delay
*(unsigned long*)(0x40021814) &= ~0x80; //Turn LED OFF PG7

*(unsigned long*)(0x40021814) |= 0x40; //Turn LED ON! PG6
for( j=0; j<1000000 ;j++ ){} //Delay
*(unsigned long*)(0x40021814) &= ~0x40; //Turn LED OFF PG6

*(unsigned long*)(0x40022014) |= 0x400; //Turn LED ON! PI10
for( j=0; j<1000000 ;j++ ){} //Delay
*(unsigned long*)(0x40022014) &= ~0x400; //Turn LED OFF PI10

*(unsigned long*)(0x40021C14) |= 0x80; //Turn LED ON! PH7
for( j=0; j<1000000 ;j++ ){} //Delay
*(unsigned long*)(0x40021C14) &= ~0x80; //Turn LED OFF PH7

*(unsigned long*)(0x40021C14) |= 0x40; //Turn LED ON! PH6
for( j=0; j<1000000 ;j++ ){} //Delay
*(unsigned long*)(0x40021C14) &= ~0x40; //Turn LED OFF PH6

*(unsigned long*)(0x40021C14) |= 0x08; //Turn LED ON! PH3
for( j=0; j<1000000 ;j++ ){} //Delay
*(unsigned long*)(0x40021C14) &= ~0x08; //Turn LED OFF PH3

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

Выводы по работе с анализом реализованной программы:

В результате работы был **с**оздан проект в среде Keil µVision и написана программа на языке C++. Программа реализует последовательное переключение диодов на микроконтроллере STM32F200 слева направо, с промежутком между переключениями примерно равным 0,5 секунды. Работа с адресами проходила без использования макросов, все адреса выражаются в виде чисел, что является не самым удобным вариантом реализации.