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 №4 по дисциплине

ПРОГРАММИРОВАНИЕ МИКРОКОНТРОЛЛЕРОВ

на тему:

Использование таймера для формирования заданного временного интервала и аппаратного прерывания для перевода микроконтроллера в режим пониженного энергопотребления

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 методами формирования заданных интервалов времени и перевода микроконтроллера в режим пониженного энергопотребления. Закрепить навыки работы с осциллографом и оценочной платой MCBSTM32F200 в качестве измерительного генератора.

**Постановка задачи**

Используя библиотеки Keil μVision5, разработать программу для микроконтроллера (МК) STM32F200, которая при помощи таймера формирует периодическое попеременное включение и выключение светодиодов PG6 и PG7 с заданными временными характеристиками (периодом следования переключений и/или длительностью фаз этого периода). При нажатии на кнопку “WAKEUP” программа должна переводить МК в спящий режим, а при ее отпускании пробуждать МК (См. Рисунок 1).

Diagram, schematic

Description automatically generated

Рисунок 1

Схема блоков кнопок и светодиодов

**Алгоритм программы**

Сперва стандартно подключаются несколько библиотек. Стандартно переопределяется функция delay().

Переопределяется функция обработки прерываний от стандартного таймера TIM и DAC (ЦАП – цифрово аналоговый преобразователь). Обработчик прерывания от таймера должен сбросить бит TIM\_SR\_UIF регистра TIMx->SR путём записи в бит 0. Иначе, после возврата из прерывания, будет снова вызван его обработчик. Проверяем, было ли прерывание от таймера в условии. Если да, то сбрасываем бит TIM\_SR\_UIF. Светодиод G8 на время загорается, затем гаснет.

Переопределяется прерывание для нулевой линии. При прерывании по нажатию кнопки WAKEUP светодиод G6 на время загорается, затем гаснет. И сбрасывается флаг прерывания.

В основном теле программы настраиваем выводы PG6 и PG7 на вывод цифровых данных аналогично предыдущим лабораторным работам. Переводим МК в состояние DeepSleep, чтобы избежать дребезг сигнала. Резервируем две линии под прерывания (сконфигурировали маскирующие биты), настраиваем линию прерывания на возрастающий и падающий фронт. Прикрепляем pa0 к зарезервированной линии.

Устанавливаем приоритет прерывания с помощью регистров приоритета прерывания NVIC\_IPR (см. файл CD00228163.pdf). Для линии использовалась функция “void NVIC\_SetPriority(IRQn\_Type IRQn, uint32\_t priority))” из строки 1464 файла core\_cm3.h. В данной функции на первом месте стоит тип прерывания, то есть его позиция в таблице прерываний, а на втором непосредственно приоритет. И активируем прерывание. Для этого необходимо воспользоваться справочным руководством по программированию (см. файл CD00228163.pdf) активация обработки определенного вектора прерывания осуществляется с помощью регистров NVIC\_ISER. Для линии использовалась функция “void NVIC\_EnableIRQ(IRQn\_Type IRQn)” из строки 1382 файла core\_cm3.h.

Разрешаем таймеру генерацию прерываний (по умолчанию, после сброса бит TIM\_CR1\_URS сброшен в 0 и прерывание генерируется как при переполнении счётчика, так и при установке бита TIM\_EGR\_UG).

Запускаем цикл в котором поочередно запускаются и гаснут светодиоды G6 и G7.

**Полученные результаты**

В результате работы с осциллографом были получены следующие результаты:

A screenshot of a computer

Description automatically generated with medium confidence

Программное измерение периода, частоты сигнала, ширины положительного и отрицательного уровней (состояние “включено”/состояние “выключено”), коэффициента заполнения периода (скважности) представлено на изображении ниже.

Graphical user interface

Description automatically generated

**Вывод**

Были освоены основные методы формирования заданных интервалов времени и перевода микроконтроллера в режим пониженного энергопотребления. Закреплены навыки работы с осциллографом и оценочной платой MCBSTM32F200 в качестве измерительного генератора.

**Приложение**

#include "stm32f2xx.h" // Device header

#include "core\_cm3.h"

#include "core\_cm0.h"

void delay ()

{

unsigned long i;

i=0;

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

}

void TIM\_DAC\_IRQHandler()

{

if(TIM6->SR&TIM\_SR\_UIF)

{

TIM6->SR&=~TIM\_SR\_UIF;

GPIOG->ODR |= 1ul<<8;

delay ();

GPIOG->ODR &= ~(1ul<<8);

}

}

void EXTI0\_IRQHandler(void)

{

GPIOG->ODR |= 1ul<<6;

delay ();

GPIOG->ODR &= ~1ul<<6;

delay ();

EXTI->PR|=EXTI\_PR\_PR0;

}

int main ()

{

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

RCC->APB2ENR|= 1ul<<14; //SYSCFGEN

SCB->SCR |= 1ul<<2;

GPIOG->MODER = (GPIOG->MODER & ~(1ul<<13)) | 1ul<<12;//PG6

GPIOG->MODER = (GPIOG->MODER & ~(1ul<<15)) | 1ul<<14;//PG7

GPIOA->MODER = (GPIOA->MODER & ~(1ul<<1)) & ~(1ul);//PA0

EXTI->IMR|=EXTI\_IMR\_MR0;

EXTI->RTSR|= EXTI\_RTSR\_TR0; //Rise Signal

EXTI->FTSR|= EXTI\_FTSR\_TR0; //Fall Signal

SYSCFG->EXTICR[0] |= SYSCFG\_EXTICR1\_EXTI0\_PA;

NVIC\_SetPriority(6,5);

NVIC\_EnableIRQ(6);

RCC->APB1ENR|=RCC\_APB1ENR\_TIM6EN;

TIM6->DIER|=TIM\_DIER\_UIE;

TIM6->EGR|=TIM\_EGR\_UG;

for (;;)

{

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

delay ();

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

GPIOG->ODR |= 1ul<<6;

delay ();

GPIOG->ODR &= ~1ul<<6;

}

}