МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ

ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

кафедра «Вычислительная техника»

**Лабораторная работа №3 (вариант 10)**

по дисциплине: «Микропроцессорные системы»

Выполнил:

студент 4 курса, гр. ИВТАПбд-41

Снежкин В. В.

Проверил:

к.т.н, доцент кафедры ВТ

Игонин А.Г.

г. Ульяновск, 2017

Содержание

Содержание………………………………………………………………………..2

Общее задание…………………………………………………………………….3

Ход работы...............................................................................................................3

Исходный код..........................................................................................................4

Вывод……………………………………………………………………………...6

Список использованной литературы……………………………………………6

**Общее задание**

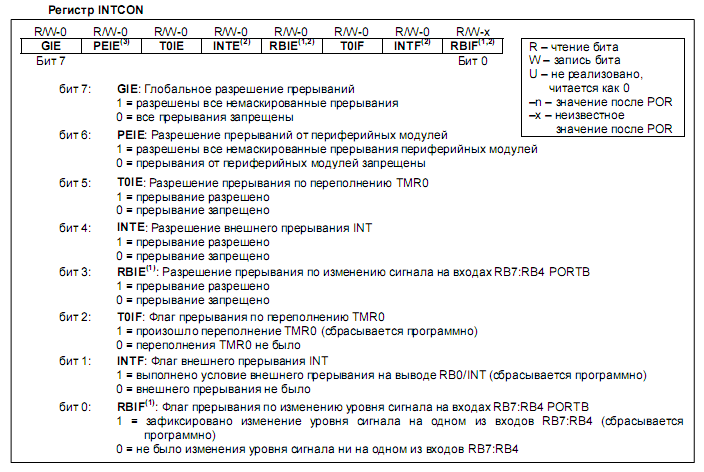
Моделирование прерываний с сохранением состояния на микроконтроллере PIC16.

**Ход работы**

Данная работа выполнялась на основе предыдущей лабораторной работы, где описаны характеристики микроконтроллера и среда моделирования.

В выбранном микроконтроллере pic 16f628a имеется 10 возможных источников прерываний:

* внешний источник прерываний INT
* изменение уровня сигнала на цифровых входах RB4:7
* переполнение таймера TMR0
* переполнение таймера TMR1
* совпадение TMR2 и PR2
* завершение записи в EEPROM
* изменение выходного уровня компаратора
* получение\завершение отправки данных через USART
* прерывания от модуля CCP



Прерывание по каждому источнику можно как разрешить, так и запретить индивидуально, изменяя соответствующие биты в регистрах **INTCON** и **PIE1**.

Для разрешения прерываний, управляемых регистром PIE1 необходимо разрешить прерывания от периферии битом **PEIE** в регистре INTCON.  
После выбора необходимых источников прерываний необходимо глобально разрешить прерывания битом **GIE** в регистре INTCON.

Для каждого прерывания имеется еще один бит в регистре **INTCON** или **PIR1** — флаг прерывания.  
При срабатывании прерывания соответствующему флагу присваивается значение 1, по которому можно легко определить какое из прерываний сработало.

Сбрасывать флаги прерываний необходимо вручную после входа в обработчик прерываний, иначе при нескольких источниках разобрать кто конкретно его вызвал будет невозможно.

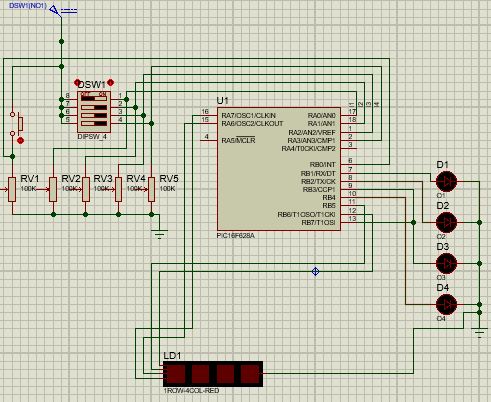
Мною были выбраны два прерывания:

1. по входу INT
2. по изменению сигнала на 7 контакте порта B

Включение прерываний происходит установкой единицы в переменные INTE и RBIE для первого и второго прерывания соответственно. А флагами для них являются INTF и RBIF.

В микроконтроллерах PIC16 используется упрощённая система прерываний. При возникновении любого из прерываний вызывается специальная функция, объявленная как *void interrupt func\_name(),* в этой функции программист должен сам определить тип прерывания и обработать его.

Прерывание INT управляется кнопкой, при его обработке выводится на лампы LD1 4 последних бита счетчика на 1 секунду. А прерывание RBI возникает при изменении сигнала на контакте RB3 и происходит моргание светодиода D2.



**Исходный код**

#include <xc.h>

#include <stdlib.h>

#include <stdio.h>

#include "config.h"

#include <stdbool.h>

#include <time.h>

int cnt = 0;

void interrupt isr(){

if(INTF){

INTF = 0; // reset interrupt flag/

RB6 = cnt & 1;

RB5 = (cnt >> 1) & 1;

RA6 = (cnt >> 2) & 1;

RA7 = (cnt >> 3) & 1;

RB7 = 1;

\_\_delay\_ms(1000);

RB7 = 0;

}

if(RBIF){

if(RB7){

for(int i = 0; i < 20; ++i){

RB2 = ~RB2;

\_\_delay\_ms(100);

}

}

RBIF = 0;

}

}

void main(void) {

TRISA = 0b00001111;

TRISB = 0B10000001;

PORTA = 0;

PORTB = 0;

INTF = 0; //reset the external interrupt flag

INTEDG = 1; //interrupt on the rising edge

INTE = 1; //enable the external interrupt

GIE = 1; //set the Global Interrupt Enable

RBIE = 1;

RBIF = 0;

while(1)

{

\_\_delay\_ms(500);

++cnt;

bool I1 = RA0;

bool I2 = RA1;

bool I3 = RA2;

bool I4 = RA3;

bool T1 = (cnt % 3 == 0);

bool T2 = (cnt % 7 == 0);

bool T3 = (cnt % 10 == 0);

bool O1 = (!I1 && !I2 && !I3) || (I2 && I4);

bool O2 = (I1 && T1 || I2 && T2);

bool O3 = (O1 && T3);

bool O4 = (O1 && O2);

RB1 = O1;

RB2 = O2;

RB3 = O3;

RB4 = O4;

}

return;

}

**Вывод**

В ходе выполнения данной лабораторной работы, были получены навыки использования прерываний в микроконтроллерах PIC16.

**Список использованной литературы**

1. <http://alex-exe.ru/radio/microcontrollers/int-pic16f628a/>
2. <https://habrahabr.ru/post/98147/>