Белорусский Государственный Университет

Информатики и Радиоэлектроники

Кафедра ЭВМ

Отчет по лабораторной работе № 2

Тема: «Прерывания. Таймеры»

Выполнили: Проверил:

Шеменков В.В

ст. гр. 950503

Зарубо Д. Ю

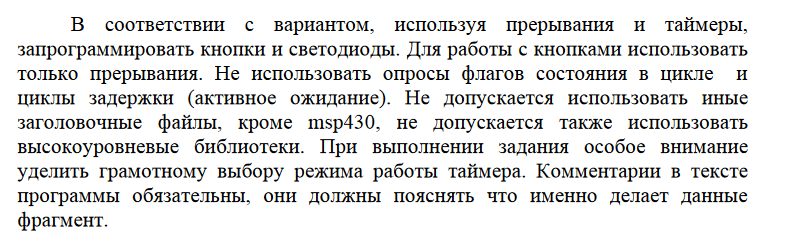
Ященко В.П

Минск 2022

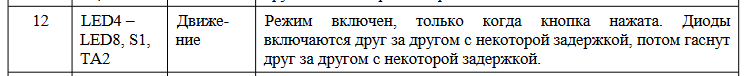
1. **ЦЕЛЬ РАБОТЫ**

Ознакомиться с работой подсистемы прерываний и таймерами

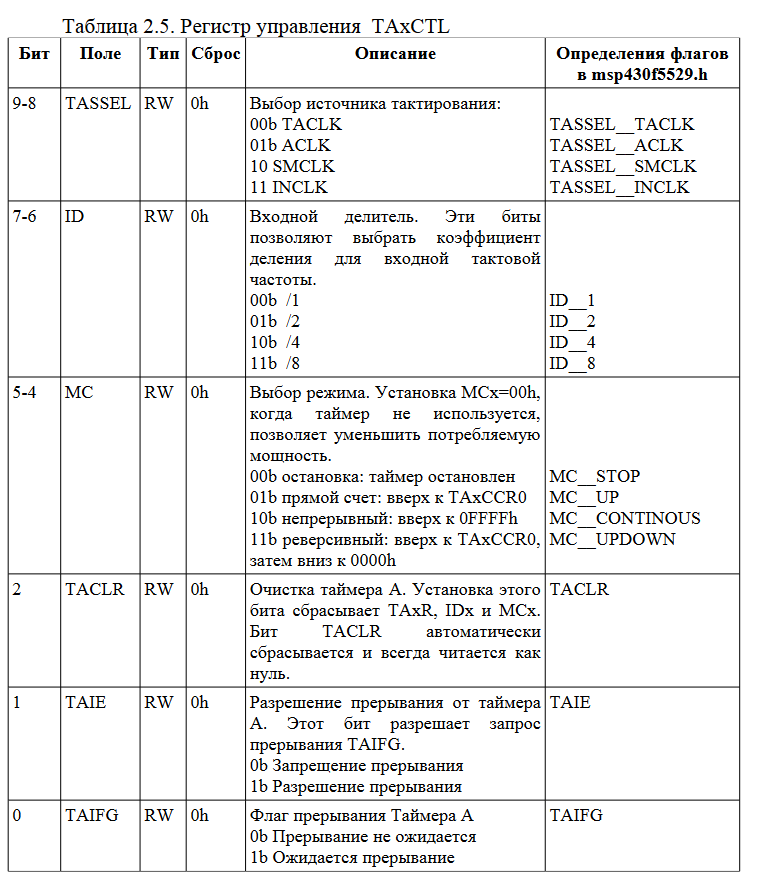
микроконтроллера MSP430F5529. Написать программу, используя таймеры и прерывания в соответствии с заданием варианта

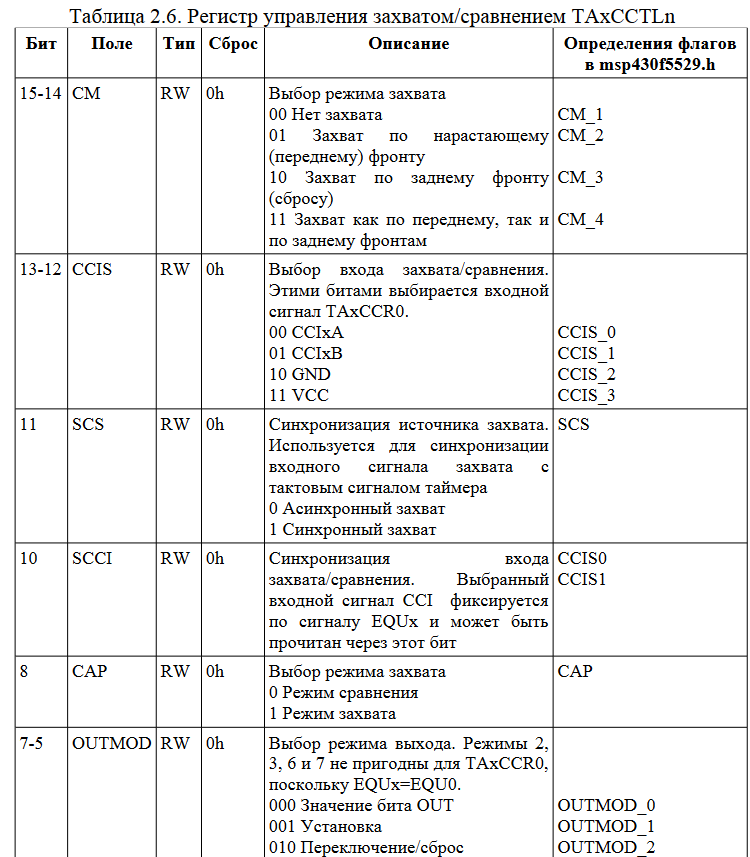
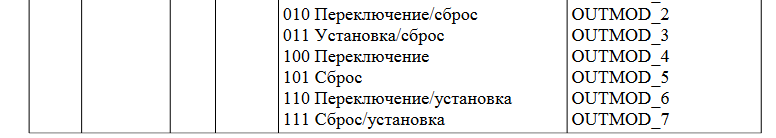
1. **ИСХОДНЫЕ ДАННЫЕ К РАБОТЕ**

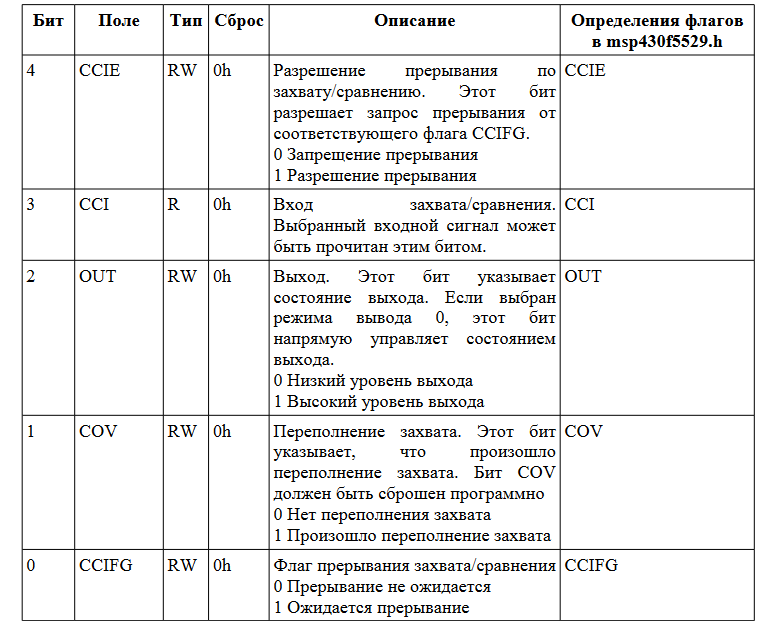
Вариант 12



1. **ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

****

****

****

**4 ВЫПОЛНЕНИЕ РАБОТЫ**

Код программмы:  
**#include** <msp430.h>

**volatile** **long** **int** sysMillis = 0;

// debounce (times)

**volatile** **int** DEBOUCE\_TRESHOLD = 2;

// Led blink poll period

**volatile** **int** ledBlinkPollPeriod = 500;

// button poll period

**volatile** **int** buttonPollPeriod = 5;

// Leds off poll period

**volatile** **char** ledOffPollPeriod = 100;

// Is button pressed or released poll period

**volatile** **char** pressReleasePollPeriod = 50;

**int** leds[5] = { BIT1, BIT2, BIT3, BIT4, BIT5 };

**volatile** **char** ledsPrevStates[5] = { };

**volatile** **int** ledIndex = 0;

**volatile** **char** ledsWereOn = 0;

**volatile** **char** buttonState, button\_debounce\_counter;

**volatile** **char** buttonPrevState;

**volatile** **char** isButtonPressed = 0;

**volatile** **char** isButtonReleased = 1;

**void** **setupPins**()

{

// S1 input

P1DIR &= ~BIT7;

// pull enable

P1REN |= BIT7;

//pull-up

P1OUT |= BIT7;

//led 1 on p1.0

//out

P1DIR |= BIT1;

//off

P1OUT &= ~BIT1;

//led 2 on 1.1

//out

P1DIR |= BIT5;

//off

P1OUT &= ~BIT5;

//led 3 on 1.2

//out

P1DIR |= BIT2;

//off

P1OUT &= ~BIT2;

//led 3 on 1.3

//out

P1DIR |= BIT3;

//off

P1OUT &= ~BIT3;

//led 4 on 1.4

//out

P1DIR |= BIT4;

//off

P1OUT &= ~BIT4;

}

// т.к нет других входных clk

// base MCLK = 1MHz

// период в 1ms = 1000 тиков при предделителе 1,

// значит TAxCCR0 = 500 при предделителе 2 - биты ID1 = 01

// TASSEL\_\_SMCLK 10

// MC\_1 = Управление режимом таймера = 01 - прямой счет до TAxCCR0

// TACLR - начальное обнуление

//

**void** **setupTimers**()

{

TA2CCR0 = 500 - 1;

// Разрешаем прерывание таймера по достижению значения TA1CCR0

TA2CCTL0 = CCIE;

// Настройка режима работы таймера Timer\_A

TA2CTL = TASSEL\_\_SMCLK | ID\_1 | MC\_1 | TACLR;

\_enable\_interrupt();

}

// 1 ms interrupt

**#pragma** vector = TIMER2\_A0\_VECTOR

\_\_interrupt **void** **CCR0\_ISR**(**void**)

{

**static** **long** **int** prevLedBlinkPollEntry = 0;

**static** **long** **int** prevButtonPollEntry = 0;

**static** **long** **int** prevLedOffPollEntry = 0;

**static** **long** **int** prevPressReleasePollPeriod = 0;

sysMillis++;

**if** (sysMillis - prevButtonPollEntry >= buttonPollPeriod)

{

prevButtonPollEntry = sysMillis;

**if** (!(P1IN & BIT7))

{

**if** (button\_debounce\_counter > DEBOUCE\_TRESHOLD)

{

buttonState = 1;

button\_debounce\_counter = 0;

}

**else**

{

button\_debounce\_counter++;

}

}

**else**

{

button\_debounce\_counter = 0;

buttonState = 0;

}

}

**if** (sysMillis - prevLedBlinkPollEntry >= ledBlinkPollPeriod && isButtonPressed)

{

**if** (!ledsWereOn)

{

**char** i;

**for** (i = 0; i < 5; i++)

{

**if** (ledsPrevStates[i])

{

P1OUT |= leds[i];

}

}

ledsWereOn = 1;

}

prevLedBlinkPollEntry = sysMillis;

P1OUT ^= leds[ledIndex];

ledIndex++;

**if** (ledIndex > 4)

ledIndex = 0;

}

**if** (sysMillis - prevPressReleasePollPeriod >= pressReleasePollPeriod)

{

prevPressReleasePollPeriod = sysMillis;

**if** (buttonPrevState != buttonState)

{

**if** (buttonState)

{

isButtonPressed = 1;

isButtonReleased = 0;

}

**if** (!buttonState)

{

isButtonPressed = 0;

isButtonReleased = 1;

}

}

buttonPrevState = buttonState;

}

**if** (sysMillis - prevLedOffPollEntry >= ledOffPollPeriod)

{

**if** (isButtonReleased)

{

**if** (ledsWereOn)

{

**char** i;

**for** (i = 0; i < 5; i++)

{

ledsPrevStates[i] = P1IN & leds[i];

}

ledsWereOn = 0;

}

P1OUT &= ~BIT1;

P1OUT &= ~BIT2;

P1OUT &= ~BIT3;

P1OUT &= ~BIT4;

P1OUT &= ~BIT5;

}

}

**return**;

}

**int** **main**(**void**)

{

WDTCTL = WDTPW | WDTHOLD;

setupPins();

setupTimers();

**while** (1);

}

**ВЫВОД**

В ходе выполнения лабораторной работы была написана программа с использованием таймеров и их прерываний. Для опроса кнопки и взаимодействия со светодиодами было использовано прерывание таймера А.