

БГУИР

Кафедра ЭВМ

Отчет по лабораторной работе № 4  
Аналоговый ввод-вывод. АЦП. Компаратор. Потенциометр. Сенсорная  
клавиатура

Выполнил:  
студент группы 050503 Липский Г.В.

Проверил:  
Шеменков В.В.

Минск 2023

## **1 ЦЕЛЬ РАБОТЫ**

Цель: Изучить принципы работы с аналоговыми сигналами на базе микроконтроллера MSP430F5529.

Задача: Написать программу с использованием АЦП, компаратора, потенциометра и сенсорных элементов в соответствии с заданием.

## 2 ИСХОДНЫЕ ДАННЫЕ

1. В соответствии с вариантом написать программу, которая непрерывно сравнивает сигнал на указанных выводах и в зависимости от того, где уровень выше, включает тот или иной светодиод. Для тач-панели использовать соответствующие кнопкам светодиоды, для потенциометра — LED3, для иных сигналов — LED1. Не допускается использовать иные заголовочные файлы, кроме `msp430`, а также использовать высокоуровневые библиотеки.

2. В отчет по выполнению работы включить исходный текст программы с обязательными комментариями. Комментарии в тексте программы обязательны, они должны пояснять что именно делает данный фрагмент. Описать подробнее используемый метод сравнения сигналов, изобразить соответствующую схему соединения и диаграммы работы. Привести объяснение полученным результатам.

### 3 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Микроконтроллер MSP430F5529 содержит 16-канальный АЦП ADC12A, обладающий следующими возможностями:

- максимальная скорость преобразования — более 200 тысяч выборок / с;
- выборка и сохранение с программируемым периодом выборки;
- запуск преобразования программно или от таймера;
- программно конфигурируемый внутренний генератор опорного напряжения (1.5, 2.0, 2.5 В);
- программный выбор внешнего или внутреннего источника опорного напряжения;
- 12 отдельно конфигурируемых внешних входных каналов;
- каналы для внутреннего датчика температуры, V<sub>сс</sub> и внешних опорных напряжений;
- независимый для каждого канала опорный источник, как положительного, так и отрицательного напряжения;
- конфигурируемый источник тактового сигнала;
- четыре режима преобразования: одноканальный, повторно-одноканальный, последовательный и повторно-последовательный;
- ядро АЦП и ИОН могут выключаться независимо друг от друга;
- быстрое декодирование 18 источников прерываний АЦП;
- сохранение 16 результатов;
- ядро АЦП автоматически отключается, если не идет цикл измерения.

Цифровое значение измеряемой величины вычисляется по формуле:

$$N_{ADC} = 4095 * (V_{IN} - V_{R-}) / (V_{R+} - V_{R-}),$$

где V<sub>IN</sub> – измеряемый (входной) сигнал, V<sub>R-</sub> и V<sub>R+</sub> источники опорного напряжения.



В импульсном режиме преобразования ( $ADC12SHP = 1$ ) сигнал  $SHI$  определяет начало выборки, биты  $ADC12SHT0x$  и  $ADC12SHT1x$  регистра  $ADC12CTL0$  задают длительность времени выборки:

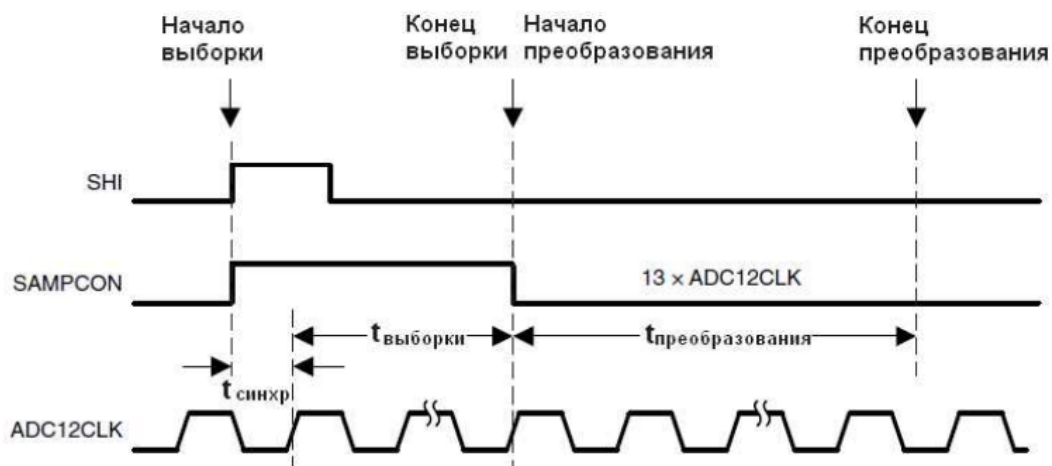


Рис. 4.3 Импульсный режим

Таблица 4.1. Регистры АЦП

Регистр	Адрес	Назначение
ADC12CTL0	0700h	Регистры управления
ADC12CTL1	0702h	
ADC12CTL2	0704h	
ADC12IFG	070Ah	Флаги прерываний
ADC12IE	070Ch	Разрешение прерываний
ADC12IV	070Eh	Вектор прерываний
ADC12MCTL0... ADC12MCTL15	0710h... 071Fh	Управление памятью
ADC12MEM0... ADC12MEM15	0720h 073Dh	Память

Таблица 4.2 Поля регистра управления АЦП ADC12CTL0

Биты	Поле	Назначение	Определение флагов в msp430f5529.h
12-15	ADC12SHT1x	Количество циклов ADC12CLK (длительность выборки) для ADC12MEM8 ... ADC12MEM15	ADC12SHT1_0 ... ADC12SHT1_15
8-11	ADC12SHT0x	Количество циклов ADC12CLK (длительность выборки) для ADC12MEM0 ... ADC12MEM7 - 4 / 8 / 16 / 32 / 64 / 96 / 128 / 192 / 256 / 384 / 512 / 768 / 1024	ADC12SHT0_0 ... ADC12SHT0_15
7	ADC12MSC	Для режима последовательности и непрерывного: 0 – для запуска требуется фронт SHI сигнала, 1 – первый по фронту SHI, потом автоматически по окончании цикла начинается новый	ADC12MSC
6	ADC12REF2_5V	Напряжение опорного генератора (ADC12REFON должен быть установлен) 0 – 1.5, 1 – 2.5 В	ADC12REF2_5V
5	ADC12REFON	Включение опорного генератора	ADC12REFON
4	ADC12ON	Включение АЦП	ADC12ON
3	ADC12OVIE	Разрешение прерывания OV	ADC12OVIE
2	ADC12TOVIE	Разрешение прерывания TOV	ADC12TOVIE
1	ADC12ENC	Разрешение измерения	ADC12ENC
0	ADC12SC	Программный запуск выборки и преобразования. Сбрасывается автоматически	ADC12SC

Таблица 4.3 Поля регистров управления АЦП ADC12CTL1-2

Регистр	Биты	Поле	Назначение	Определение флагов в msp430f5529.h
ADC12CTL1	12-15	ADC12CSTARTADDx	Стартовый адрес записи результата	ADC12CSTARTADD_0 ... ADC12CSTARTADD_15
	10-11	ADC12SHSx	Источник сигнала запуска	ADC12SHS_0 ... ADC12SHS_3
	9	ADC12SHP	Выбор источника сигнала SAMPCON (расширенный или импульсный режим)	ADC12SHP
	8	ADC12ISSH	Инвертирование сигнала запуска	ADC12ISSH
	5-7	ADC12DIVx	Делитель тактовой частоты	ADC12DIV_0 ... ADC12DIV_7
	3-4	ADC12SSELx	Выбор тактового сигнала	ADC12SSEL_0 ... ADC12SSEL_3
	1-2	ADC12CONSEQx	Режим	ADC12CONSEQ_0 ... ADC12CONSEQ_3
	0	ADC12BUSY	Индикатор активного режима	ADC12BUSY
ADC12CTL2	8	ADC12PDIV	Деление тактовой частоты на 4	ADC12PDIV
	7	ADC12TCOFF	Отключение температурного датчика	ADC12TCOFF
	4-5	ADC12RES	Точность (8, 10, 12 бит)	ADC12RES_0 ... ADC12RES_3
	3	ADC12DF	Результат - знаковый	ADC12DF
	2	ADC12SR	Буфер работает на частоте 0: 200K , 1: 50K выборок / с	ADC12SR
	1	ADC12REFOUT	Выход опорного напряжения	ADC12REFOUT
	0	ADC12REFBURST	Режим работы буферного усилителя опорного напряжения: постоянный (0) или автоматический (1)	ADC12REFBURST



Таблица 4.4 Поля иных регистров АЦП ADC12

Регистр	Биты	Поле	Назначение	Определение флагов в msp430f5529.h
ADC12MEMx	0-15		Результат измерения	
ADC12MCTLx	7	ADC12EOS	Маркер конца последовательности	ADC12EOS
	4-6	ADC12SREFx	Выбор пары опорных напряжений VR+, VR-	ADC12SREF_0 ... ADC12SREF_7
	0-3	ADC12INCHx	Выбор входного канала	ADC12INCH_0 ... ADC12INCH_15
ADC12IE			Разрешение прерывания по соответствующему флагу	ADC12IE0 ... ADC12IE15
ADC12IFG			Флаг запроса на прерывание	ADC12IFG0 ... ADC12IFG15
ADC12IV			Вектор запросов на прерывания	ADC12IV_NONE, ADC12IV_ADC12OVIFG, ADC12IV_ADC12TOVIFG, ADC12IV_ADC12IFG0 ... ADC12IV_ADC12IFG15

Для получения максимальной частоты измерений используется бит  $ADC12MSC = 1$  (Multiple Sample and Convert) при режиме  $CONSEQx > 0$  и использовании таймера выборки. В этом случае первый фронт SHI запускает первое измерение, после измерения автоматически запускается следующее. «Лишние» фронты SHI игнорируются, пока не окончится цикл измерения.

Использование встроенного температурного датчика: помимо выбора номера канала (1010), устанавливается либо  $ADC12REFON = 1$  (при  $REFMSTR = 0$ ), либо  $REFON = 1$  (при  $REFMSTR = 1$ ), при этом необходимо использовать период выборки больше 30 мкс.

## 4.2 Компаратор

Компаратор в составе MSP430F5529 обладает следующими возможностями: прямое и инверсное сравнение; программное подключение RC-фильтра на выходе; выход подключается ко входу таймера А; программный выбор каналов (из 16 возможных); использование прерываний; программируемый генератор опорного напряжения.

Структура компаратор представлена на рис. 4.4, а принцип работы фильтрации на выходе — на рис. 4.5.

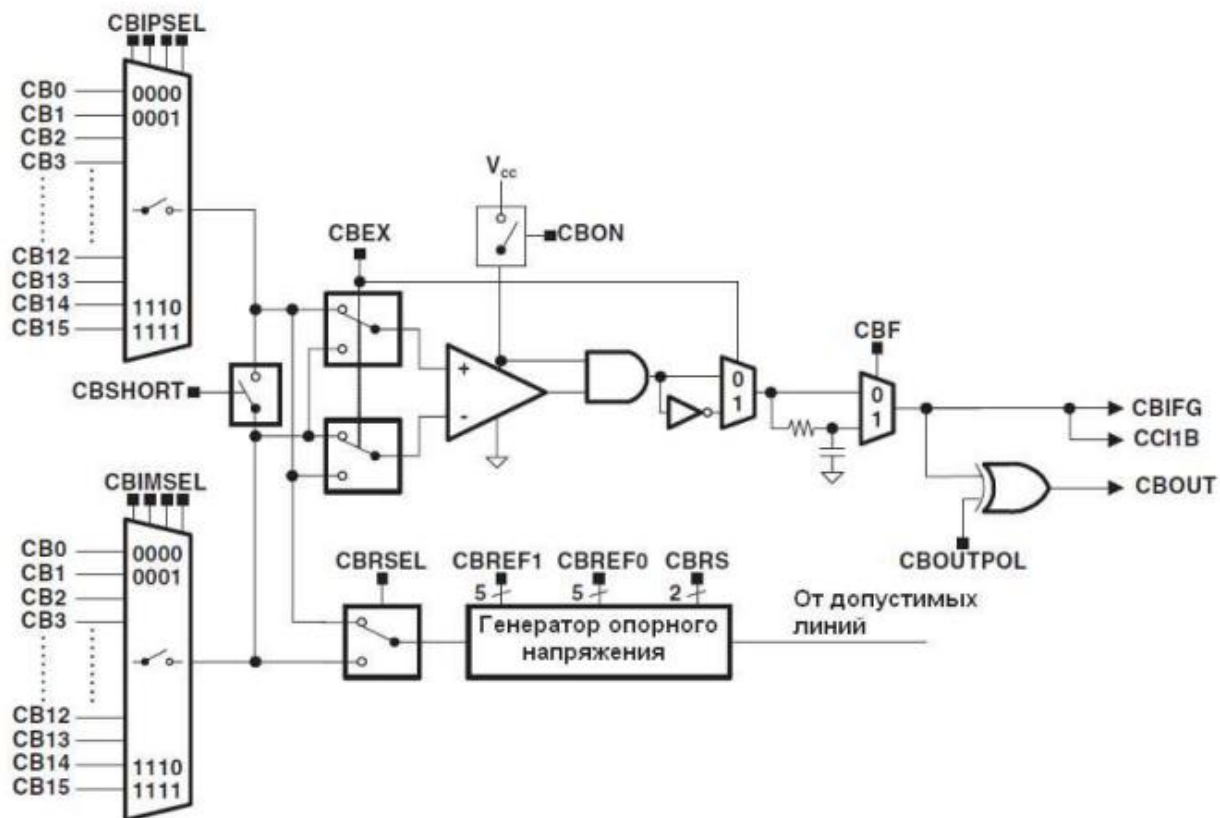


Рис. 4.4 Структура компаратора

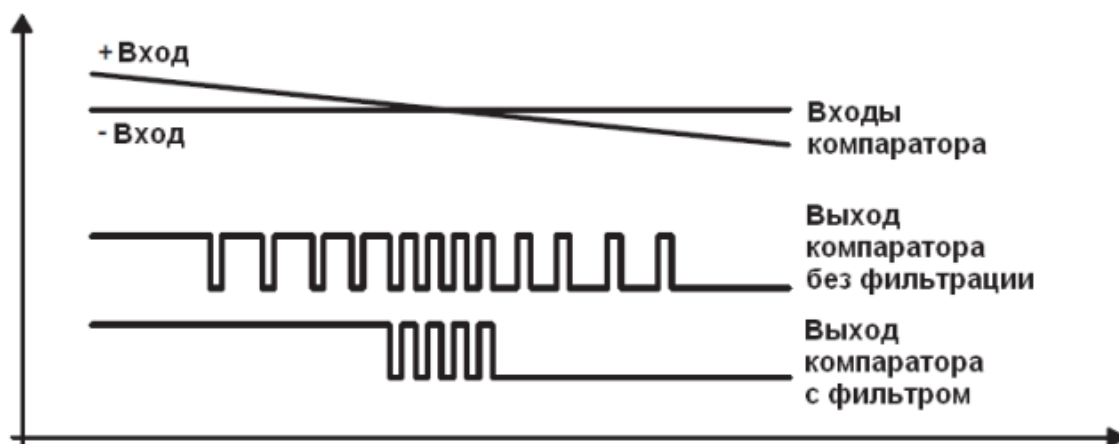


Рис. 4.5 Использование фильтра на выходе компаратора

Регистры управления компаратором CBCTL0 ... CBCTL3 имеют адреса 08C0h, 0802h, 0804h, 0806h, регистр управления прерываниями CBINT – 080Ch, регистр флагов прерываний CBIV – 080Eh. Состав и назначение полей регистров компаратора приведены в таблице:

Таблица 4.5 Поля регистров компаратора

Регистр	Биты	Поле	Назначение	Определение флагов в <a href="#">msp430f5529.h</a>
CBCTL0	15	CBIMEN	Разрешение входного канала на V-	CBIMEN
	8-11	CBIMSEL	Выбор входного канала V-	CBIMSEL_0 ... CBIMSEL_15
	7	CBIPEN	Разрешение входного канала на V+	CBIPEN
	0-3	CBIPSEL	Выбор входного канала V+	CBIPSEL_0 ... CBIPSEL_15
CBCTL1	12	CBMRVS	0 – выход компаратора управляет выбором между VREF0 и VREF1 1 – управляет CBMRVL	CBMRVS
	11	CBMRVL	Выбор VREFx	CBMRVL
	10	CBON	Включение компаратора	CBON
	8-9	CBPWRMD	Режим питания	CBPWRMD_0 ... CBPWRMD_3
	6-7	CBFDLY	Величина задержки фильтра (0.6, 1.0, 1.8, 3.4 мкс)	CBFDLY_0 ... CBFDLY_3
	5	CBEX	Инверсный режим	CBEX
	4	CBSHORT	Закорачивание входных каналов	CBSHORT
	3	CBIES	Выбор прерывания по фронту или спаду	CBIES

Окончание табл. 4.5

Регистр	Биты	Поле	Назначение	Определение флагов в msp430f5529.h
CBCTL1	2	CBF	Выходной фильтр	CBF
	1	CBOUTPOL	Полярность выхода	CBOUTPOL
	0	CBOUT	Выход компаратора	CBOUT
CBCTL2	15	CBREFACC	Точность	CBREFACC
	13-14	CBREFL	Опорное напряжение (отключено, 1.5, 2, 2.5 В)	CBREFL_0 ... CBREFL_3
	8-12	CBREF1	Устанавливает напряжение на выходе резисторного делителя	CBREF1_0 ... CBREF1_31
	6-7	CBRS	Источник опорного напряжения	CBRS_0 ... CBRS_3
	5	CBRSEL	Коммутация опорного напряжения к + и - входам	
	0-4	CBREF0	Как и CBREF1	CBREF0_0 ... CBREF0_31
CBCTL3	0-15	CBPD	Отключение входного буфера для каждого вывода	CBPD0 ... CBPD15
CBINT	9	CBIE	Выход разрешения прерывания по инверсному выходу	CBIE
	8	CBIE	Выход разрешения прерывания	CBIE
	1	CBIFG	Флаг инверсного прерывания	CBIFG
	0	CBIFG	Флаг прерывания	CBIFG
			Вектор прерываний	CBIV_NONE, CBIV_CBIFG, CBIV_CBIFG

### 4.3 Потенциометр

Схема подключения потенциометра на макете MSP-EXP430F5529 приведена на рис. 4.6. Сигнал с потенциометра подан на вывод A5 контроллера. Он соединен с соответствующими каналами (CB5, A5) на входах компаратора и АЦП. Поскольку данный вывод разделен с цифровым I/O (P6.5), его необходимо переключить в режим периферийных устройств на ввод данных. Порт P8.0 необходимо использовать в цифровом режиме на выход, подав на него высокий уровень, чтобы подать на резистор разность потенциалов.

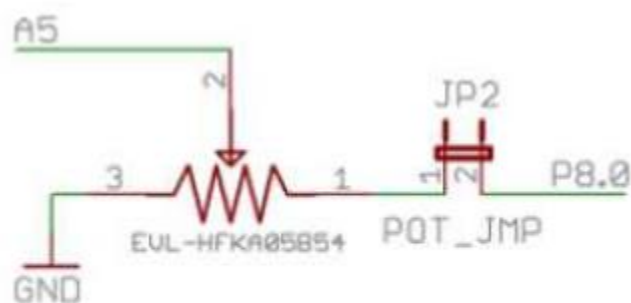


Рис. 4.6 Схема подключения потенциометра

#### 4.4 Сенсорная клавиатура

На плате расположены пять площадок, совмещенных со светодиодными индикаторами. Каждая из таких площадок представляет собой емкостный сенсорный элемент. Сенсор сконструирован таким образом, что его электрическое поле и емкость могут быть изменены внешним токопроводящим объектом, например пальцем:

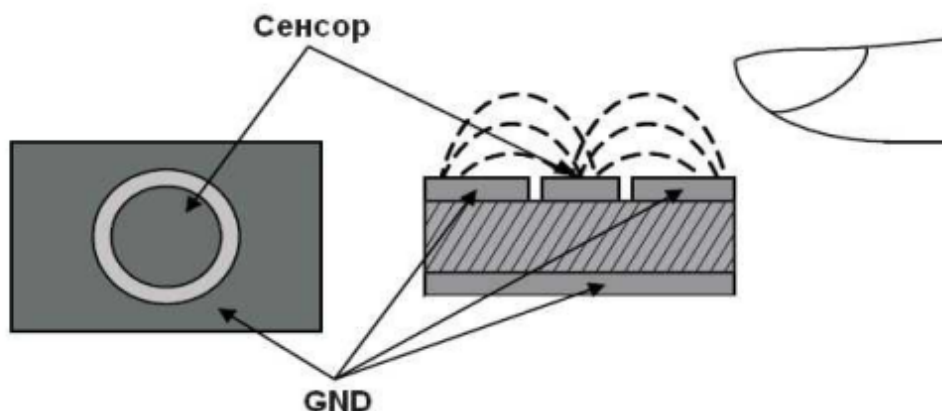


Рис. 4.7 Принцип действия сенсорного элемента

При приближении к сенсору меняется магнитное поле и, следовательно, емкость. Количественная характеристика нажатия получается путем измерения емкости тач-сенсора. Очевидно, что магнитное поле сильно зависит от условий внешней среды, поэтому требуется отслеживание фонового уровня.



Для снижения чувствительности сенсора к шуму используют разные подходы. Например, подавление шума с помощью БИХ-фильтра, либо усреднение времени заряда и времени разряда.

Иногда сенсоры объединяют попарно, что позволяет вдвое снизить количество используемых резисторов. В этом случае, когда один сенсор измеряется, второй заряжается и наоборот:

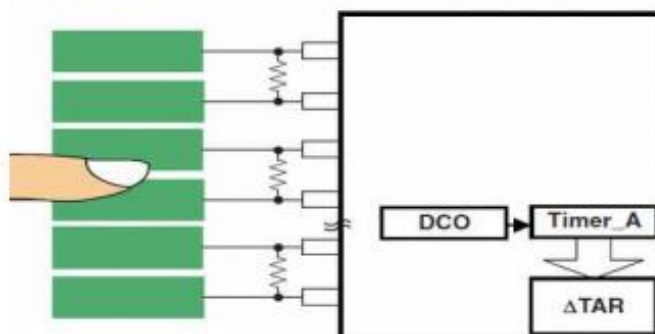


Рис. 4.9 Попарное объединение сенсоров

Другими методами, используемыми для измерений, являются RO-метод (Relaxation Oscillator) и fRO (fast scan Relaxation Oscillator). В случае RO-метода считается количество тактов релаксационного генератора за некоторое время. Генератор включает в себя сенсор и компаратор:

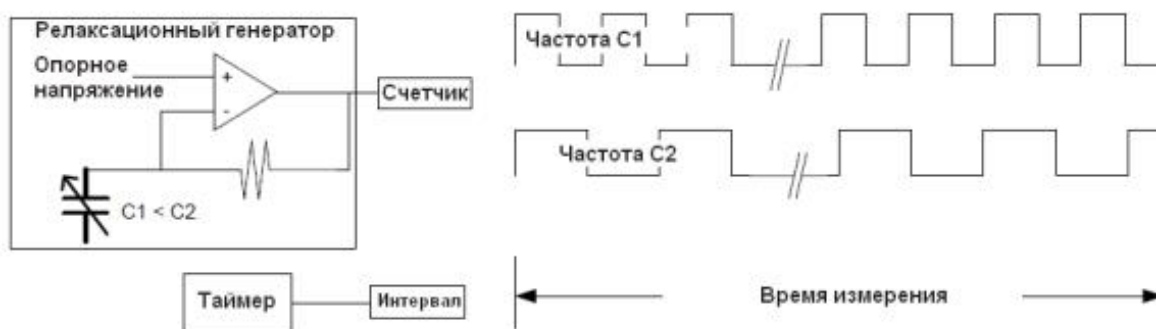


Рис. 4.10 Измерение емкости сенсора. RO-метод

## 4 ВЫПОЛНЕНИЕ РАБОТЫ

### 4.1 Листинг кода

```
#include <msp430.h>

char compareLedMode = 0;
volatile int i;

void led_init()
{
    P1DIR |= BIT0 | BIT1 | BIT3;
    P8DIR |= BIT2;

    P1OUT &= ~BIT0;
    P1OUT &= ~BIT1;
    P1OUT &= ~BIT3;
    P8OUT &= ~BIT2;
}

void button_init()
{
    P1DIR &= ~BIT7;
    P1REN |= BIT7;
    P1OUT |= BIT7;
    P1IES |= BIT7;
    P1IFG &= ~BIT7;
    P1IE |= BIT7;
}

void adc_init()
{
    //enable potentiometer
    ////set potentiometer output as analog input
    P6SEL |= BIT5;
    ////set potentiometer input (max value) to "1"
    P8DIR |= BIT0;
    P8OUT |= BIT0;

    //select start scanning from 0
    ADC12CTL1 &= ~(ADC12CSTARTADD0 | ADC12CSTARTADD1 | ADC12CSTARTADD2 |
ADC12CSTARTADD3);
    //Source select = ADC12SC
    ADC12CTL1 &= ~(ADC12SHS0 | ADC12SHS1);
    //adc clock select (SMCLK)
    ADC12CTL1 |= ADC12SSEL0 | ADC12SSEL1;
    //Pulse mode enable
    ADC12CTL1 |= ADC12SHP;
    //repeated multichannel mode select
    ADC12CTL1 |= ADC12CONSEQ0 | ADC12CONSEQ1;
    //enable auto sampling (but need first SHI rising edge)
    ADC12CTL0 |= ADC12MSC;
    //set adc resolution for 8 bits
    ADC12CTL2 &= ~(BIT5 | BIT4);
    //set sample rate buffer to 50ksp/s
    ADC12CTL2 &= ~ADC12SR;
```

```

// channel 0 config (select potentiometer as source)
ADC12MCTL0 |= ADC12INCH_5;
// channel 1 config (select temp as source)
///set channel 1 as last for adc
ADC12MCTL1 |= ADC12EOS;
///select channel source
ADC12MCTL1 |= ADC12INCH_9;
}

```

```

void comparator_init()
{
    //Enable Minus & select PAD3 [CB2]
    CBCTL0_H = CBIPEN + CBIPSEL_2;
    //Enable Plus & select PAD1 [CB0]
    CBCTL0_L = CBIPEN + CBIPSEL_0;
    //Set filter delay to 3600ns
    CBCTL1 |= CBFDL_3;
    //Enable output filter
    CBCTL1 |= CBF;
    //Select interrupt rising edge
    CBCTL1 &= ~CBIES;
    //Set comparator power mode to ultra low power
    CBCTL1 |= BIT9;
    CBCTL1 &= ~BIT8;
    //set other pads as input pullup
    P6DIR &= ~(BIT0 | BIT2 | BIT4);
    P6OUT |= BIT0 | BIT2 | BIT4;

    //Disable interrupt
    //CBINT &= ~CBIE;
    //CBINT &= ~CBIIE;
    //Disable comparator
    //CBCTL1 &= ~CBON;

    //Enable comparator
    CBCTL1 |= CBON;
    //Enable interrupt (main & inverted)
    CBINT |= CBIE;
    CBINT |= CBIIE;
    //clear
    CBINT &= ~CBIFG;
    CBINT &= ~CBIIFG;
}

```

```

#pragma vector=PORT1_VECTOR
__interrupt void PORT1_ISR(void)
{
    for(i=0;i<2500;i++);

    if(!(P1IN & BIT7))
    {
        P1IFG &= ~BIT7;

        if(!compareLedMode)
        {
            P1OUT &= ~BIT1;
            P1OUT &= ~BIT3;

```



```

        //Disable interrupt
        CBINT &= ~CBIE;
        CBINT &= ~CBIIE;
        //Disable comparator
        CBCTL1 &= ~CBON;

        //enable ADC
        ADC12CTL0 |= ADC12ON;
        //enable ADC conversion and start calc
        ADC12CTL0 |= ADC12ENC | ADC12SC;
        ///enable channel 1 interrupts
        ADC12IE |= ADC12IE1;
        //clear
        ADC12IFG &= ~ADC12IFG1;
    }
else
{
    P1OUT &= ~BIT0;
    P8OUT &= ~BIT2;

    //disable interrupts
    ADC12IE &= ~ADC12IE1;
    //disable ADC
    ADC12CTL0 &= ~ADC12ON;
    //disable ADC conversion and reset start calc bit
    ADC12CTL0 &= ~(ADC12ENC | ADC12SC);

    //Enable comparator
    CBCTL1 |= CBON;
    //Enable interrupt (main & inverted)
    CBINT |= CBIE;
    CBINT |= CBIIE;
    //clear
    CBINT &= ~CBIFG;
    CBINT &= ~CBIIFG;
}

compareLedMode ^= BIT0;
}
}

/**
 * main.c
 */
int main(void)
{
    WDTCTL = WDTPW | WDTHOLD;    // stop watchdog timer

    led_init();
    button_init();

    adc_init();
    comparator_init();

    _bis_SR_register(GIE + LPM0_bits);
    while(1);

    return;

```

```

}

#pragma vector=COMP_B_VECTOR
__interrupt void comparator_interrupt()
{
    //value = 1 => (+ > -)
    //value = 0 => (+ < -)
    //+ => PAD1
    //- => PAD3

    if(CBCTL1 & CBOUT)
    {
        P1OUT |= BIT1;
        P1OUT &= ~BIT3;
    }
    else{
        P1OUT &= ~BIT1;
        P1OUT |= BIT3;
    }

    //clear
    CBINT &= ~CBIFG;
    CBINT &= ~CBIIFG;
}

#pragma vector=ADC12_VECTOR
__interrupt void adc_interrupt()
{
    short val_U = ADC12MEM1;
    short val_potent = ADC12MEM0;

    if(val_U > val_potent){
        P1OUT |= BIT0;
        P8OUT &= ~BIT2;
    }
    else{
        P1OUT &= ~BIT0;
        P8OUT |= BIT2;
    }

    ADC12IFG &= ~ADC12IFG1;
}

```