**Лабораторная работа №1**

**Измерение аналоговых сигналов**

**Цель:**

1. Получить знания о способах измерения параметров аналоговых сигналов и первичную обработку результатов измерения;

2. Получить навыки построения системы измерения аналоговых сигналов.

**Задание:**

1. Реализовать измерение трех аналоговых сигналов. В диапазоне от 0 до 5 вольт с периодом дискретизации 1 с

2. Необходимо провести усреднение этих сигналов

3. Усредненные сигналы необходимо сравнить с аварийным пределом 4 вольта и при выходе за аварийный предел производить индикацию

4. Подготовить для каждого сигнала данные для индикации на трехразрядных семисегментных индикаторах.

**Теория:**

Основной задачей измерений является установление числового значения параметров, характеризующих свойства физической величины. В статистических измерениях входное воздействие рассматривается как реализация случайного процесса, вероятностная характеристика которого измеряется. При измерении физической величины необходимо указать, до какого момента времени, относится результат измерения. В основе формализованного описания измерительной процедуры лежит уравнение измерений, устанавливает связь результата измерения с входным воздействием и выполняемыми преобразованиями. Наличие уравнения измерения позволяет создать необходимое программное обеспечение, а также провести метрологический анализ измерительной процедуры и результатов измерения. При проведении измерений используются три вида измерительных преобразований - аналоговые, аналого-цифровые и цифровые. Аналоговое преобразование обеспечивает необходимый диапазон изменения измеряемой величины, аналого-цифровое - переход к числового представления, цифровое - дополнительную обработку сигнала. Основой работы цифровых измерительных приборов является превращение непрерывной (аналоговой) величины в цифровой код. Методы аналого-цифрового преобразования в измерениях сводятся к представлению мгновенных значений входного воздействия в фиксированные моменты времени соответствующей кодовой комбинацией (числом). Физическую основу аналого-цифрового преобразования составляет стробирования и уравнение с фиксированными опорными уровнями. Наибольшее распространение получили аналого-цифровые преобразователи (АЦП) поразрядного кодирования, последовательного счета, следящего уравновешивания и другие.

**Код программы:**

.include "m16def.inc"

.def temp = R16

.equ CTC\_VALUE = 31250

.def CHANNEL = R17

.def tempH = R18

.def CH0L = R20

.def CH0H = R21

.def CH1L = R22

.def CH1H = R23

.def CH2L = R24

.def CH2H = R25

.def Ans = R15

.def AnsU = R14

.def AnsX = R13

.macro CH\_N ; @0-CHNL | @1-CHNH

add @0,temp

adc @1,tempH ; Add with Carry

lsr @1

ror @0 ; Rotate Right Through Carry

rjmp CHANGE\_CH

.endm

.macro CMP\_CH\_N ; @0-CHNL | @1-CHNH | @2-TURN\_OFF\_0 | @3-TURN\_ON\_0 | @4-CMPR\_L0 | @5-CMPR\_H0 | @6-END\_CH0 | @7- Port Number (0/1/2)

cpi @1,HIGH(800) ;

brsh @5

rjmp @2

@5:

cpi @1,HIGH(800) ;

breq @4

@3:

sbi PORTB,@7 ;turn on LED

rjmp @6

@4:

cpi @0,LOW(800) ;

brsh @3

@2:

cbi PORTB,@7 ;turn off LED

@6:

.endm

jmp RESET ; Reset Handler

jmp EXT\_INT0 ; IRQ0 Handler

jmp EXT\_INT1 ; IRQ1 Handler

jmp TIM2\_COMP ; Timer2 Compare Handler

jmp TIM2\_OVF ; Timer2 Overflow Handler

jmp TIM1\_CAPT ; Timer1 Capture Handler

jmp TIM1\_COMPA ; Timer1 CompareA Handler

jmp TIM1\_COMPB ; Timer1 CompareB Handler

jmp TIM1\_OVF ; Timer1 Overflow Handler

jmp TIM0\_OVF ; Timer0 Overflow Handler

jmp SPI\_STC ; SPI Transfer Complete Handler

jmp USART\_RXC ; USART RX Complete Handler

jmp USART\_UDRE ; UDR Empty Handler

jmp USART\_TXC ; USART TX Complete Handler

jmp ADC\_C ; ADC Conversion Complete Handler

jmp EE\_RDY ; EEPROM Ready Handler

jmp ANA\_COMP ; Analog Comparator Handler

jmp TWSI ; Two-wire Serial Interface Handler

jmp EXT\_INT2 ; IRQ2 Handler

jmp TIM0\_COMP ; Timer0 Compare Handler

jmp SPM\_RDY ; Store Program Memory Ready Handler

;

RESET:

ldi temp,high(RAMEND) ; Main program start

out SPH,temp ; Set Stack Pointer to top of RAM

ldi temp,low(RAMEND)

out SPL,temp

; Define pull-ups and set outputs high

; Define directions for port pins

ldi temp,(0<<PB0)|(0<<PB1)|(0<<PB2)

out PORTB,temp

ldi temp,(1<<DDB0)|(1<<DDB1)|(1<<DDB2)

out DDRB,temp

; Define ADC

ldi temp,(0<<MUX4)|(0<<MUX3)|(0<<MUX2)|(0<<MUX1)|(0<<MUX0)|(0<<ADLAR)|(0<<REFS1)|(0<<REFS0)|(1<<DDB2)

out ADMUX,temp

ldi temp,(1<<ADEN)|(1<<ADSC)|(0<<ADATE)|(1<<ADIE)|(1<<ADPS2)|(1<<ADPS1)|(0<<ADPS0)

out ADCSRA,temp

; Define Timer (sampling time 1s)

ldi temp,high(CTC\_VALUE)

out OCR1AH,temp

ldi temp,low(CTC\_VALUE)

out OCR1AL,temp

ldi temp,(0<<WGM11)|(0<<WGM10)

out TCCR1A,temp

ldi temp,(0<<WGM13)|(1<<WGM12)|(1<<CS12)|(0<<CS11)|(0<<CS10)

out TCCR1B,temp

; start from 0 channel

ldi CHANNEL,0

clr CH0L

clr CH0H

clr CH1L

clr CH1H

clr CH2L

clr CH2H

sei ; Enable interrupts

LOOP:

rjmp LOOP

ADC\_C: ; ADC Conversion Complete Handler

in temp,ADCL; from ADC to Core

in tempH,ADCH

cpi CHANNEL,0

breq CH0

cpi CHANNEL,1

breq CH1

cpi CHANNEL,2

breq CH2

CHANGE\_CH:

inc CHANNEL

cpi CHANNEL,3

breq END\_ADC

out ADMUX,CHANNEL

ldi temp,(1<<ADEN)|(1<<ADSC)|(0<<ADATE)|(1<<ADIE)|(1<<ADPS2)|(1<<ADPS1)|(0<<ADPS0)

out ADCSRA,temp

END\_ADC:

reti

CH0:

CH\_N CH0L, CH0H

//add CH0L,temp

//adc CH0H,tempH ; Add with Carry

//lsr CH0H

//ror CH0L ; Rotate Right Through Carry

//rjmp CHANGE\_CH

CH1:

CH\_N CH1L, CH1H

//add CH1L,temp

//adc CH1H,tempH ; Add with Carry

//lsr CH1H

//ror CH1L ; Rotate Right Through Carry

//rjmp CHANGE\_CH

CH2:

CH\_N CH2L, CH2H

//add CH2L,temp

//adc CH2H,tempH ; Add with Carry

//lsr CH2H

//ror CH2L ; Rotate Right Through Carry

//rjmp CHANGE\_CH

TIM1\_COMPA: ; Timer1 CompareA Handler

ldi CHANNEL, 0

out ADMUX,CHANNEL

ldi temp,(1<<ADEN)|(1<<ADSC)|(0<<ADATE)|(1<<ADIE)|(1<<ADPS2)|(1<<ADPS1)|(0<<ADPS0)

out ADCSRA,temp ;start ADC after 3 ch , start next measure cycle

CMP\_CH\_N CH0L, CH0H, TURN\_OFF\_0, TURN\_ON\_0, CMPR\_L0, CMPR\_H0, END\_CH0, 0

CMP\_CH\_N CH1L, CH1H, TURN\_OFF\_1, TURN\_ON\_1, CMPR\_L1, CMPR\_H1, END\_CH1, 1

CMP\_CH\_N CH2L, CH2H, TURN\_OFF\_2, TURN\_ON\_2, CMPR\_L2, CMPR\_H2, END\_CH2, 2

clr Ans; clear var

clr AnsU

clr AnsX

mov temp,CH0L

mov tempH,CH0H

rcall TO\_BCD

sts (CH0\_IND+0),Ans

sts (CH0\_IND+1),AnsU

sts (CH0\_IND+2),AnsX

mov temp,CH1L

mov tempH,CH1H

rcall TO\_BCD

sts (CH1\_IND+0),Ans

sts (CH1\_IND+1),AnsU

sts (CH1\_IND+2),AnsX

mov temp,CH2L

mov tempH,CH2H

rcall TO\_BCD

sts (CH2\_IND+0),Ans

sts (CH2\_IND+1),AnsU

sts (CH2\_IND+2),AnsX

reti

EXT\_INT0: ; IRQ0 Handler

EXT\_INT1: ; IRQ1 Handler

TIM2\_COMP: ; Timer2 Compare Handler

TIM2\_OVF: ; Timer2 Overflow Handler

TIM1\_CAPT: ; Timer1 Capture Handler

TIM1\_COMPB: ; Timer1 CompareB Handler

TIM1\_OVF: ; Timer1 Overflow Handler

TIM0\_OVF: ; Timer0 Overflow Handler

SPI\_STC: ; SPI Transfer Complete Handler

USART\_RXC: ; USART RX Complete Handler

USART\_UDRE: ; UDR Empty Handler

USART\_TXC: ; USART TX Complete Handler

EE\_RDY: ; EEPROM Ready Handler

ANA\_COMP: ; Analog Comparator Handler

TWSI: ; Two-wire Serial Interface Handler

EXT\_INT2: ; IRQ2 Handler

TIM0\_COMP: ; Timer0 Compare Handler

SPM\_RDY:; Store Program Memory Ready Handler

reti

To\_BCD: //binary code to decimal digits

lvl200:

cpi tempH,0

brne Sub200

cpi temp,200

brlo lvl20

Sub200:

subi temp,200

sbci tempH,0

inc Ans

rjmp lvl200

lvl20:

cpi temp,20

brlo lvl2

subi temp,20

inc AnsU

rjmp lvl20

lvl2:

lsr temp; shif temp ==> /2

mov AnsX,temp

ret

.dseg

CH0\_IND:

.byte 3

CH1\_IND:

.byte 3

CH2\_IND:

.byte 3

**Вывод.** На лабораторной работе мы реализовали измерение трех аналоговых сигналов, провели усреднение этих сигналов и сравнили с аварийным пределом 4 вольта.