|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

Факультет: «Специальное машиностроение»

Кафедра: «Робототехнические системы и мехатроника»

**Лабораторная работа № 1**

по курсу «Теория автоматического управления»

Вариант 8

Выполнил: Ионин Даниил

Группа: СМ11-61Б

Проверил(а):

Москва, 2024 г.

ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 2](#_Toc167680229)

[программный код 3](#_Toc167680230)

программный код

|  |
| --- |
| ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ;\* Designer Mechtcherjakowa R.I.  ;\* Version: 1.0  ;\* Date 08.04.2010  ;\* Title: ADC\_proj.asm  ;\* Device ATmega16  ;\* Clock frequency:Частота кв.резонатора 8 mHz  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ; учебная  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ;Программа осуществляет измерение напряжения,снимаемого с 2-х переменных резисторов,  ;подключенных к цепи AREF=2,56B(вывод 32)и вывод измеренного значения на семисегм. индикаторы  ;измеряемые напряжения подаются на вх PA0 (ADC0) и PA3 (ADC3)  ;выбор канала, выводимого на индикаторы осуществляется нажатием кнопки "Выбор канала",  ;подключенной к выводу PD3  ;данные с АЦП усредняются путем суммирования (Val\_N\_ADC=1,2,4,8) раз и нахождения  ;среднего значения, используя для деления сдвиг вправо  ;измеренное значение Vin=(ADC\*2,56)/1024=ADC/400. Используем только целые числа для  ;вывода на дисплей, т.е. Vin\*100. таким образом измеренное значение равно ADC/4  ;увеличение в 100 раз при выводе на дисплей компенсируем выводом точки у первой зн.цифры  ;семисегментные индикаторы подключены к PC0-PC7(выходы),  ;PC0-a,PC1-b,PC2-c,PC3-d,PC4-e,PC5-f,PC6-g,PC7-h  ;сигналы выбора индикатора PB3-номер канала АЦП,PB2 - единицы,PB1-десятые,PB0-сотые (выходы)  ;===================================================================================  ;ЗАДАНИЕ: подключить резисторы к цепи VCC и изменить программные настройки и рез.измер-й  ;продемонстрировать умение выполнять операции с битами  ;доработать программу на выполнение одного преобразования  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  .include "m16def.inc"  ; присоединение файла описаний;  .list ;включение листинга  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ; Register Variables  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  .def temp\_L =R16  .def temp\_H =R17  .def Number =R18  .def Hundreds =R19;  .def Tens =R20  .def Ones =R21  .def Disp\_Numb =R22  .def Disp\_Count =R23  .def Time =R24;счетчик переполн.Т0(1024\*255/8000000\*Xotc=1cek(Xotc=30)  .def ADC\_h =R3  .def ADC\_l =R4  .def cou\_ADC =R25  ;-------------------  .def byte\_fl =R9  ;-------------------  .equ F\_iz\_kan =0;флаг изменения канала:0 2-й канал.1 6-й канал  .equ F\_end\_ADC =1;флаг завершения (Val\_N\_ADC) преобразований АЦП  .equ Disab\_Key =2;бит запрещения опроса кнопки "Выбор канала"  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ; Constants  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  .equ Val\_dispCount=50  .equ Val\_N\_ADC =4;4;колич-во преобразований АЦП(1,2,4,8.16..)  .equ VAL\_time =30;30 переполнений соответствуют 1 сек  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  .cseg  .org $0000  rjmp Init  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  .org INT0addr;=$002 ;External Interrupt0 Vector Address  reti;  ;----------------  .org INT1addr;=$004 ;External Interrupt1 Vector Address  reti  .org OC2addr; =$006 ;Output Compare2 Interrupt Vector Address  reti  .org OVF2addr;=$008 ;Overflow2 Interrupt Vector Address  reti  .org ICP1addr;=$00A ;Input Capture1 Interrupt Vector Address  reti  .org OC1Aaddr;=$00C ;Output Compare1A Interrupt Vector Address  reti  .org OC1Baddr;=$00E ;Output Compare1B Interrupt Vector Address  reti  .org OVF1addr;=$010 ;Overflow1 Interrupt Vector Address  reti  .org OVF0addr;=$012 ;Overflow0 Interrupt Vector Address  rjmp time\_d\_k ;  .org SPIaddr; =$014 ;SPI Interrupt Vector Address  reti  .org URXCaddr;=$016 ;UART Receive Complete Interrupt Vector Address  reti  .org UDREaddr;=$018 ;UART Data Register Empty Interrupt Vector Address  reti  .org UTXCaddr; =$01A ;UART Transmit Complete Interrupt Vector Address  reti  .org ADCCaddr; =$01C ;ADC Interrupt Vector Address  rjmp IN\_ADC  .org ERDYaddr; =$01E ;EEPROM Interrupt Vector Address  reti  .org ACIaddr; =$020 ;Analog Comparator Interrupt Vector Address  reti  .org TWIaddr; =$022 ;Irq. vector address for Two-Wire Interface  reti  .org INT2addr; =$024 ;External Interrupt2 Vector Address  reti  .org OC0addr; =$026 ;Output Compare0 Interrupt Vector Address  reti  .org SPMRaddr; =$028 ;Store Program Memory Ready Interrupt Vector Address  reti  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Init:  ldi temp\_L,LOW(RAMEND);выбор вершины стека  out SPL, temp\_L;Указатель стека  ldi temp\_L,HIGH(RAMEND)  out SPH,temp\_L  ; ------Инициализация портов В/B  ;  ldi temp\_L,0b11111111;(PB1-PB7)-выходы,  out DDRB,temp\_L  ldi temp\_L,0b00001000;выбраны PB3-номер байта (1-й индикатор)  out PORTB,temp\_L  ;  ldi temp\_L,0b00000000;PD3-кнопка "Выбор канала" (вход)  out DDRD,temp\_L  ldi temp\_L,0b00001000;Вкл подтяжка на кнопке PD3  out PORTD,temp\_L;  ;  ldi temp\_L,0b11111111;(PС1-PС7)-выходы  out DDRC,temp\_L  ;Analog-to-digital  ldi temp\_L,0x00  out DDRA,temp\_L; входы АЦП  ;  ;(Internal 2.56V Voltage Refer. with external capacitor at AREF pin)REFS1,REFS0=1  ;0-й канал ADC0  ldi temp\_L,(1<<REFS1)|(1<<REFS0);  out ADMUX,temp\_L  ;  ldi temp\_L,(1<<ADPS2)|(1<<ADPS1);  out ADCSR,temp\_L  ;  clr byte\_fl  clr cou\_ADC  ;  clr ADC\_h;  clr ADC\_l  ;  clr Disp\_Numb  clr Hundreds;включение индикаторов 0 знач.  clr Tens  clr Ones  ldi Number,0;0-й канал  ldi Disp\_Count,Val\_dispCount  sei  ;  ;;==================================================  ;начало цикла программы  ;==================================================  Start: rcall start\_ADC  ;  wait\_ADC: rcall Display  sbrs byte\_fl,F\_end\_ADC  rjmp wait\_ADC  rcall out\_ADC  sbrc Byte\_fl,Disab\_Key;  rjmp Start; wait\_ADC  sbis PinD,3;Проверка нажатия кнопки  rcall izm\_Nkan;;по нажатию кнопки инвертировать флаг изменения канала  rjmp Start; wait\_ADC  ;==================================================  ; конец цикла программы  ;==================================================  ; Подпрограмма смены канала преобразования  ;==================================================  izm\_Nkan:  set  bld Byte\_fl,Disab\_Key; бит запрещ. опроса кнопки "Выбор канала" от дребезга    clt  sbrs byte\_fl,F\_iz\_kan  set  bld byte\_fl,F\_iz\_kan  ;изменить № канала АЦП  in temp\_L,ADMUX  andi temp\_L,0b11100000  ch\_mux\_ADC: sbrs byte\_fl,F\_iz\_kan  rjmp kanN2  ldi Number,3;3-й канал  ori temp\_L,(1<<MUX0)|(1<<MUX1)  ex\_c\_mux: out ADMUX,temp\_L  ldi temp\_L,(1<<CS02)|(1<<CS00);частота TCNT0 Clk/1024,(CS02,CS01,CS00) - 101  out TCCR0,temp\_L  ldi temp\_L,(1<<TOIE0);TOIE0-Timer/Counter0 Overflow Interrupt Enable  out TIMSK,temp\_L; разрешить прерывание Tov0  ret  ;---------------  kanN2: ldi Number,0;0-й канал  rjmp ex\_c\_mux  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ; Подпрограмма запуска преобразования АЦП  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  start\_ADC: ldi temp\_L,(1<<ADEN)|(1<<ADIE)|(1<<ADPS2)|(1<<ADPS1);ч-та преобр.64(125кГц)  out ADCSR,temp\_L  ldi temp\_L,(1<<ADEN)|(1<<ADIE)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADSC);  out ADCSR,temp\_L  ret  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ;Подпрограмма out\_ADC изменение данных выводимых на дисплей  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ;данные с АЦП усредняются путем суммирования (Val\_N\_ADC=1,2,4,8.16) раз и нахождения  ;среднего значения, используя для деления сдвиг вправо  ;измеренное значение Vin=(ADC\*2,54)/1024=ADC/400. Используем только целые числа для  ;вывода на дисплей, т.е. Vin\*100. таким образом измеренное значение равно ADC/4  ;увеличение в 100 раз при выводе на дисплей компенсируем выводом точки у первой  ; зн.цифры  ;кол-во сдвигов вправо для нах-я средн.значения  out\_ADC: ldi temp\_L,Val\_N\_ADC;  cpi temp\_L,8; 8 накоплений  brne ch\_4izm  ldi temp\_L,3  rjmp sh\_ADC  ch\_4izm: cpi temp\_L,4; 4 накоплений  brne ch\_2izm  ldi temp\_L,2  rjmp sh\_ADC  ch\_2izm: cpi temp\_L,2; 2 накоплений  brne ch\_1izm  ldi temp\_L,1  rjmp sh\_ADC  ch\_1izm: cpi temp\_L,1  breq norm\_ADC  rjmp ex\_out  sh\_ADC: lsr ADC\_h  ror ADC\_l  dec temp\_L  cpi temp\_L,0x00  brne sh\_ADC;в ADC\_h,ADC\_l среднее значение  ;  norm\_ADC: lsr ADC\_h  ror ADC\_l  lsr ADC\_h  ror ADC\_l; в ADC\_l значение,увеличенное в 100 раз  ;  mov temp\_L,ADC\_l  mov temp\_H,ADC\_h  clr ADC\_h  clr ADC\_l  rcall digitConvert  clt  bld byte\_fl,F\_end\_ADC  ex\_out: ret  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ; Подпрограмма Display работы с дисплеем (динам. индикация)  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Display: dec Disp\_Count  brne ex\_displ  ;  ldi Disp\_Count,Val\_dispCount  ;  inc Disp\_Numb  cpi Disp\_Numb,4  brne Out\_disp  clr Disp\_Numb  ;  Out\_disp:  ldi ZL,18; ;указатель на Number  ldi ZH,00  add ZL,Disp\_Numb  ld temp\_L,Z  cpi Disp\_Numb,1;позиция Hundreds  breq sym\_toch  ; преобразуем в семисегментный код  ldi ZL,low(TABLE\*2);загружаем адрес начала  ldi ZH,high(TABLE\*2);таблицы в памяти программ (\*2 - для байтовой  out\_date: add ZL,temp\_L ;адресации)  lpm temp\_L,Z ;читаем семисегментный код значения ;  out PortC,temp\_L; передаем на индикатор  ;  in temp\_L,PINB  lsr temp\_L  brcc PC+2  ldi temp\_L,0b00001000; в начало (Number)  out PORTB,temp\_L  ;  ex\_displ: ret  ;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  sym\_toch: ldi ZL,low(TABLE1\*2);загружаем адрес начала  ldi ZH,high(TABLE1\*2);таблицы в памяти программ (\*2 - для байтовой адрес.  rjmp out\_date  ;==================================================  ;------- Таблица перекодировки символов  TABLE: .db 0b00111111,0b00000110; коды "0","1"  .db 0b01011011,0b01001111; коды "2","3"  .db 0b01100110,0b01101101;;коды "4","5"  .db 0b01111101,0b00000111;;коды "6","7"  .db 0b01111111,0b01101111;;коды "8","9"  ;==================================================  ;------- Таблица перекодировки символов c точкой, отделяющей целые числа от дробных  TABLE1: .db 0b10111111,0b10000110; коды "0","1"  .db 0b11011011,0b11001111; коды "2","3"  .db 0b11100110,0b11101101;;коды "4","5"  .db 0b11111101,0b10000111;;коды "6","7"  .db 0b11111111,0b11101111;;коды "8","9"  ;==================================================  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ;Конвертирует двоичное 1-e байтное число в двоично-десятичный код  ;===================================================  digitConvert:  clr Hundreds  clr Tens  clr Ones  ;  FindHundreds:  subi temp\_L,100  ; sbci temp\_H,0;  brcs FindTens  inc Hundreds  rjmp FindHundreds  ;  FindTens:  subi temp\_L,-100  subi temp\_L,10  brcs FindOnes  inc Tens  rjmp FindTens+1  ;  FindOnes:  subi temp\_L,-10  mov Ones,temp\_L  ret  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ;Subroutine interrupt ADC  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  IN\_ADC: push temp\_L  push temp\_H  in temp\_L,SREG  push temp\_L  ;  rd\_ADC: in temp\_L,ADCL  in temp\_H,ADCH  add ADC\_l,temp\_L  adc ADC\_h,temp\_H  inc cou\_ADC  cpi cou\_ADC,Val\_N\_ADC;колич-во преобразований АЦП  breq end\_ADC  ldi temp\_L,(1<<ADEN)|(1<<ADIE)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADSC);;start convers.  out ADCSR,temp\_L  ex\_INADC: pop temp\_L  out SREG,temp\_L  pop temp\_H  pop temp\_L  reti  ;-------------------------  end\_ADC: clr cou\_ADC  set  bld byte\_fl,F\_end\_ADC  rjmp ex\_INADC  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ;Subroutine interrupt Overflow 0  ;Подпрограмма обработки прерывания переполнения таймера Т0  ;Т0 - 8 разр таймер. Counter0 оverflow соотвествуетвремени 256\*1024/8000000=0,033с  ;за 1cек =(1024/8000000)\*256\*N\_отсчетов(переполнений)=30 переполнений  ;(предотвращение срабатываний от дребезга контактов кнопки "Просмотр")  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  time\_d\_k: push temp\_L  in temp\_L,SREG  push temp\_L  ;  inc Time  cpi Time,VAL\_time  brne ex\_timDK  clt  bld Byte\_fl,Disab\_Key; бит разр. опроса кнопки "Просмотр" от дребезга  ldi temp\_L,0x00;(1<<CS02)|(1<<CS00);No clock source,(CS02,CS01,CS00) - 000  out TCCR0,temp\_L  in temp\_L,TIMSK  clt  bld temp\_L,TOIE0 ;TOIE0-Timer/Counter0 Overflow Interrupt Enable  out TIMSK,temp\_L; запрещение прерывание Tov0  clr Time  ;  ex\_timDK: pop temp\_L  out SREG,temp\_L  pop temp\_L  reti  ;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |