# Mục tiêu:

* Hiểu và sử dụng được các ngoại vi UART, I2C, SPI
* Hiểu cách giao tiếp với RTC, EEPROM

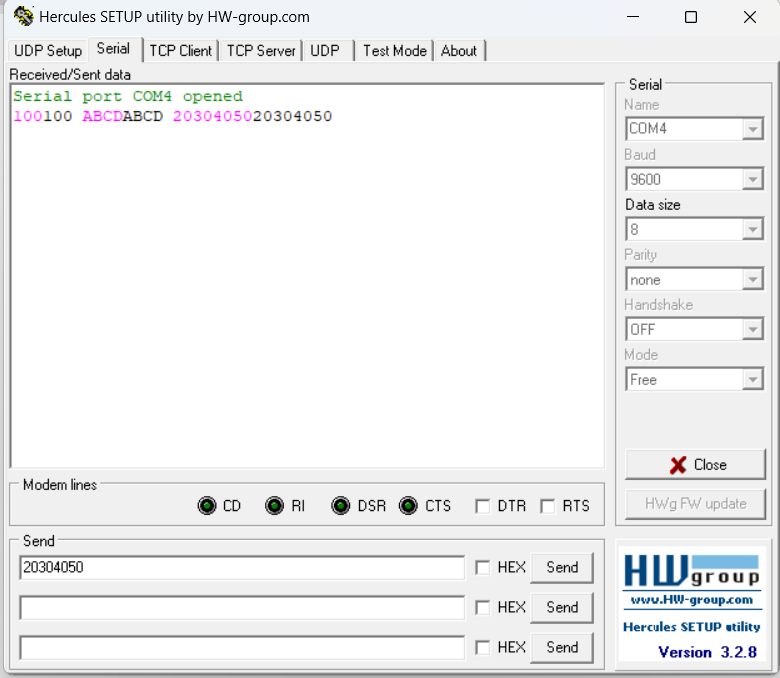
# Tham khảo:

* Tài liệu hướng dẫn thí nghiệm, chương 7, 9, 11
* Atmel-2505-Setup-and-Use-of-AVR-Timers\_ApplicationNote\_AVR130.pdf

# Bài 1

1. Kết nối chân TxD và RxD của UART0 vào vào tín hiệu UART\_TxD0 và UART\_RxD0 trên header J85 ở khối UART.
2. Kết nối dây USB-Serial vào kit thí nghiệm
3. Setup chương trình Hercules với baudrate 9600, 8 bit data, no parity, 1 stop, no handshake.
4. Sử dụng các ví dụ mẫu trong tài liệu thí nghiệm, viết chương trình khởi động UART0 với các thông số như trên, chờ nhận một byte từ UART0 và phát ngược lại UART0.
5. Dùng Hercules truyền một ký tự xuống kit và quan sát các dữ liệu nhận được để kiểm tra hoạt động chương trình.

(Lưu ý: tần số xung clock cho CPU trên kit thí nghiệm là 8Mhz)



.DEF DATA\_RX = R18

.DEF DATA\_TX = R19

.ORG 0

RJMP MAIN

.ORG $40

MAIN:

LDI R16, HIGH(RAMEND)

OUT SPH, R16

LDI R16, LOW(RAMEND)

OUT SPL, R16

CALL USART\_INIT

CLR DATA\_RX

CLR DATA\_TX

START:

CALL USART\_RECEIVER\_CHAR

MOV DATA\_TX, DATA\_RX

CALL USART\_SEND\_CHAR

RJMP START

USART\_INIT:

LDI R16, (1<<RXEN0) | (1<<TXEN0) ;Enable transmitter and receiver

STS UCSR0B, R16

LDI R16, $00

STS UBRR0H, R16

LDI R16, 51 ;Set baudrate to 9600bps with 8Mhz clock

STS UBRR0L, R16

LDI R16, (1<<UCSZ01) | (1<<UCSZ00) ;8 bit data, no parity, 1 stop bit

STS UCSR0C, R16

RET

USART\_SEND\_CHAR:

PUSH R17

WAIT\_USART\_SEND:

LDS R17, UCSR0A

SBRS R17, UDRE0 ;bit UDRE0 = 1 then continue to send char

RJMP WAIT\_USART\_SEND

STS UDR0, DATA\_TX ;Write new data

POP R17

RET

USART\_RECEIVER\_CHAR:

PUSH R17

WAIT\_USART\_RECEIVE:

LDS R17, UCSR0A

SBRS R17, RXC0 ;bit RXC0 = 1 then continue to receive char

RJMP WAIT\_USART\_RECEIVE

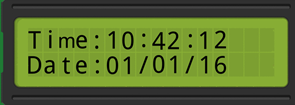
LDS DATA\_RX, UDR0 ;Read new data

POP R17

RET

# Bài 2

1. Kết nối các tín hiệu SDA và SCL của AVR vào các tín hiệu tương ứng trên module RTC. Kết nối 1 chân port vào tín hiệu MFP. Kết nối LCD 16x2 vào 1 port của AVR
2. Viết chương trình con khởi động RTC với thời gian hiện hành, cấu hình xung MFP tần số 1Hz. Sau đó cứ mỗi cạnh lên của MFP, đọc các giá trị ngày tháng năm giờ phút giây của RTC và cập nhật lên LCD



1. Biên dịch chương trình và quan sát LCD để kiểm tra chương trình.

; SCL PC0, SDA PC1

.DEF REG\_FLAG=R19

.DEF COUNT=R20

.DEF NUM\_MAX=R21

.DEF NUM\_MIN=R22

.DEF POS\_CRS=R23

.EQU LCD=PORTA ;KHAI BAO PORTA

.EQU LCD\_DR=DDRA

.EQU CONT=PORTD ; KHAI BAO PORTD

.EQU CONT\_DR=DDRD

.EQU CONT\_IN=PIND

.EQU SW\_FLG=0

.EQU RS=0 ;bit RS

.EQU RW=1 ;bit RW

.EQU E=2 ;bit E

.EQU SCL=0

.EQU SDA=0

.EQU SW1=0

.EQU SW2=1

.EQU STO=7

.EQU VBATEN=3

.EQU NULL=$00

.EQU CTL\_BYTE=0B11011110

.EQU RTC\_BUF=0X200

.ORG 0x00

RJMP MAIN

.ORG 0X40

MAIN: LDI R16,HIGH(RAMEND)

OUT SPH,R16

LDI R16,LOW(RAMEND)

OUT SPL,R16

LDI R16,0XFF

OUT LCD\_DR,R16

LDI R16,0X00

OUT LCD,R16

CBI CONT\_DR,SW1 ; SW1 INP

SBI CONT,SW1 ; RESISTOR

CBI CONT\_DR,SW2 ; SW2 INP

SBI CONT,SW2 ; RESISTOR

LDI R16,250 ; DELAY 25MS

RCALL DELAY\_US ; CTC DELAY 100US\*R16

LDI R16,250

RCALL DELAY\_US

CBI LCD,RS ; RS=0 WRITE

LDI R17,$30 ; LAN 1 ,RS=RW=E=0

RCALL OUT\_LCD4 ; CTC WRITE LCD

LDI R16,42 ; DELAY 4.2MS

RCALL DELAY\_US

CBI LCD,RS

LDI R17,$30 ;mã lenh=$30 lAn 2

RCALL OUT\_LCD4

LDI R16,2 ;delay 200Us

RCALL DELAY\_US

CBI LCD,RS

LDI R17,$30 ;mã lenh=$30 LAN 3

RCALL OUT\_LCD4

LDI R16,1

RCALL DELAY\_US

CBI LCD,RS

LDI R17,$20

RCALL OUT\_LCD4

LDI R18,$28 ;FUNCTION SET 2 DONG FONT 5X8, MODE 4 BIT

LDI R19,$01 ;CLEAR DISPLAY

LDI R20,$0C ;DISPLAY ON, CON TRO OFF

LDI R21,$06 ;ENTRY MODE SET DICH PHAI CON TRO, DDRAM TANG 1 D/C

; KHI NHAP KY TU, MAN HINH KHOONG DICH

RCALL INIT\_LCD4 ;CTC KHOI DONG LCD 4 BIT

RCALL TWI\_INIT

START:

LDI REG\_FLAG,0 ;XOA CAC CO BAO

LDI R16,1 ;CHO 100US

RCALL DELAY\_US

CBI LCD,RS ; RS=0 GHI LENH

LDI R17,$01 ; XOA MAN HINH

RCALL OUT\_LCD

LDI R16,20 ; CHO 2 MS SAU LENH CLEAR DISPLAY

RCALL DELAY\_US

LDI R17,$80 ;CON TRO BAT DAU O DAU DONG 1

RCALL CURS\_POS ;XUAT LENH RA LCD

LDI ZH,HIGH(MSG1<<1) ; Z TRO DIA CHI DAU BANG MSG1 TRONG FLASH ROM

LDI ZL,LOW(MSG1<<1)

RCALL MSG\_DISP ;GHI MSG1 RA LCD

LDI R17,$C0

RCALL CURS\_POS

LDI ZH,HIGH(MSG2<<1) ; Z TRO DAU BANG TRA MSG2

LDI ZL,LOW(MSG2<<1)

RCALL MSG\_DISP ; GHI MSG2 RA LCD

;------------------------------------------------------------------

;DAT BIT SQWEN=1, RS2:0=000 CHO DAO DONG 1HZ; XUAT RA CHAN MFP

;------------------------------------------------------------------

RCALL TWI\_START

LDI R17,(CTL\_BYTE|0X00)

RCALL TWI\_WRITE

LDI R17,0X07

RCALL TWI\_WRITE

LDI R17,0B01000000

RCALL TWI\_WRITE

RCALL TWI\_STOP

;-----------------------------------------------------------------

;DOC CAC THANH GHI 0X00-0X06 RTC

;----------------------------------

START1:

LDI XH,HIGH(RTC\_BUF); X TRO DAU BUFFER RTC

LDI XL,LOW(RTC\_BUF)

LDI COUNT,7

RCALL TWI\_START ; PHAT XUNG START

LDI R17,(CTL\_BYTE|0X00) ; TRUY XUAT GHI RTC\_TCCR

RCALL TWI\_WRITE ; GHI RTC+W

LDI R17,0X00 ; DIA CHI THNAH GHI 0X00

RCALL TWI\_WRITE

RCALL TWI\_START ;PHAT XUNG START

LDI R17,(CTL\_BYTE|0X01) ;TRUY XUAT DOC RTC\_TCCR

RCALL TWI\_WRITE ;GHI RTC+R

RTC\_RD:

RCALL TWI\_READ

ST X+,R17

DEC COUNT

BRNE RTC\_RD

RCALL TWI\_NAK

RCALL TWI\_STOP

;-----------------------------------------------------------------

;HIEN THI THU GIO:PHUT:GIAY

;-----------------------------------------------------------------

START2:

LDI R17,$0C

CBI LCD,RS

LDI R16,1

RCALL DELAY\_US

RCALL OUT\_LCD

LDI XH,HIGH(RTC\_BUF+3) ; X TRO BUFFER RTC THU

LDI XL,LOW(RTC\_BUF+3)

LDI R17,$84 ;CON TRO BAT DAU O DONG 1 VI TRI THU

RCALL CURS\_POS ; XUAT LENH RA LCD

LD R17,X ; LAY DATA THU

ANDI R17,0X07

LDI R18,0X30 ;CHUYEN SSANG MA ASCII

ADD R17,R18

SBI LCD,RS

LDI R16,1 ;DELAY 100US

RCALL DELAY\_US

RCALL OUT\_LCD ;HIEN THI RA LCD

LDI R17,0X20 ;MA DAU TRONG

SBI LCD,RS

LDI R16,1

RCALL DELAY\_US ;CHO 100US

RCALL OUT\_LCD

LDI COUNT,3

LDI R17,$86

RCALL CURS\_POS

DISP\_NXT1:

LD R17,-X ;LAY DATA

CPI COUNT,1 ;DATA=SEC

BRNE D\_NXT ;KHAC, HIEN THI TIEP

CBR R17,(1<<STO) ;XOA BIT ST

D\_NXT:

RCALL NUM\_DISP

DEC COUNT

BREQ QUIT1

LDI R17,':'

SBI LCD,RS

LDI R16,1 ;CHO 100US

RCALL DELAY\_US

RCALL OUT\_LCD ;HIEN THI RA LCD

RJMP DISP\_NXT1

;---------------------------------------------

;HIEN THI NGAY/THANG/NAM

;---------------------------------------------

QUIT1:

LDI XH,HIGH(RTC\_BUF+4) ; X TRO BUFFER RTC NGAY

LDI XL,LOW(RTC\_BUF+4)

LDI COUNT,3

LDI R17,$C6 ;CON TRO BAT DAU O DONG 2 VI TRI NGAY

RCALL CURS\_POS ;XUAT LENH RA LCD

DISP\_NXT2:

LD R17,X+

RCALL NUM\_DISP

DEC COUNT

BREQ SW\_CHK

LDI R17,'/'

SBI LCD,RS

LDI R16,1 ;CHO 100US

RCALL DELAY\_US

RCALL OUT\_LCD ;HIEN THI RA LCD

RJMP DISP\_NXT2

;-----------------------------------

;DAT LAI RTC

;------------------------------

SW\_CHK:

RCALL GET\_SW ; DOC SW CHO SW NHAN

SBRS REG\_FLAG,SW\_FLG

RJMP START1

CPI R17,1

BRNE SW\_CHK

LDI R17,$0E ; HIEN THI CON TRO

CBI LCD,RS

LDI R16,1 ; CHO 100US

RCALL DELAY\_US

RCALL OUT\_LCD ;XUAT LENH RA LCD

RTC\_SET:

CPI COUNT,0 ; CAI DAT THU

BRNE HR\_CHK ; KHAC, KIEM TRA GIO

LDI XH,HIGH(RTC\_BUF+3) ; X TRO BUFFER RTC THU

LDI XL,LOW(RTC\_BUF+3)

LDI NUM\_MAX,7

LDI NUM\_MIN,1

LDI POS\_CRS,$84 ;DAT CON TRO VI TRI THU

RCALL SET\_NUM ;DAT VA HIEN THU

LD R17,X

SBR R17,(1<<VBATEN) ;CHO PHEP NGUON BACKUP

ST X,R17

RJMP RTC\_SET

HR\_CHK:

CPI COUNT,1 ; CAI DAT GIO

BRNE MI\_CHK ; KHAC, KIEM TR APHUT

LDI XH,HIGH(RTC\_BUF+2) ; X TRO BUFFER RTC GIO

LDI XL,LOW(RTC\_BUF+2)

LDI NUM\_MAX,0X23

LDI NUM\_MIN,0

LDI POS\_CRS,$86 ;DAT CON TRO VI TRI GIO

RCALL SET\_NUM

RJMP RTC\_SET

MI\_CHK:

CPI COUNT,2

BRNE SEC\_CHK ; CAI DAT PHUT

LDI XH,HIGH(RTC\_BUF+1) ;KHAC, KIEM TRA GIAY

LDI XL,LOW(RTC\_BUF+1) ;X TRO BUFFER RTC PHUT

LDI NUM\_MAX,0X59

LDI NUM\_MIN,0

LDI POS\_CRS,$89 ;DAT CON TRO VI TRI PHUT

RCALL SET\_NUM

RJMP RTC\_SET

SEC\_CHK:

CPI COUNT,3 ;CAI DAT GIAY

BRNE DAT\_CHK ;KIEM TRA NGAY

LDI XH,HIGH(RTC\_BUF) ;X TRO BUFFER RTC GIAY

LDI XL,LOW(RTC\_BUF)

LDI NUM\_MAX,0X59

LDI NUM\_MIN,0

LDI POS\_CRS,$8C ; DAT CON TRO VI TRI GIAY

RCALL SET\_NUM

LD R17,X

SBR R17,(1<<STO)

ST X,R17

RJMP RTC\_SET

DAT\_CHK:

CPI COUNT,4 ;CAI DAT NGAY

BRNE MO\_CHK ;KHAC, KIEM TRA THANG

LDI XH,HIGH(RTC\_BUF+4) ;X TRO BUFFER RTC NGAY

LDI XL,LOW(RTC\_BUF+4)

LDI NUM\_MAX,0X31

LDI NUM\_MIN,1

LDI POS\_CRS,$C6

RCALL SET\_NUM

RJMP RTC\_SET

MO\_CHK:

CPI COUNT,5 ;CAI DAT THANG

BRNE YEA\_CHK ;KHAC, KIEM TRA NAM

LDI XH,HIGH(RTC\_BUF+5) ;X TRO BUFFER RTC THANG

LDI XL,LOW(RTC\_BUF+5)

LDI NUM\_MAX,0X12

LDI NUM\_MIN,1

LDI POS\_CRS,$C9

RCALL SET\_NUM

YEA\_CHK:

CPI COUNT,6 ; CAI DAT NAM

BRNE EXIT\_CHK ; KHAC, THOAT

LDI XH,HIGH(RTC\_BUF+6) ;X TRO BUFFER RTC NAM

LDI XL,LOW(RTC\_BUF+6)

LDI NUM\_MAX,0X99

LDI NUM\_MIN,1

LDI POS\_CRS,$CC ;CON TRO VI TRI NAM

RCALL SET\_NUM

RJMP RTC\_SET

;-----------------------------------

;LUU CAC GIA TRI CAI DAT VAO RTCC

;-----------------------------------

EXIT\_CHK:

LDI COUNT,7 ;LUU VAO RTCC

LDI XH,HIGH(RTC\_BUF); X TRO BUFFER RTC

LDI XL,LOW(RTC\_BUF)

RCALL TWI\_START ;PHAT XUNG START

LDI R17,(CTL\_BYTE|0X00) ;TRUY XUAT GHI RTC

RCALL TWI\_WRITE ;GHI RTC+W

LDI R17,0X00 ;DIA CHI THANH GHI GIAY

RCALL TWI\_WRITE ;GHI DIA CHI TCCR

WR\_RTC:

LD R17,X+

RCALL TWI\_WRITE ;GHI TCCR

DEC COUNT

BRNE WR\_RTC

RCALL TWI\_STOP

RJMP START1

;------------------------------------

;GET\_SW DOC TRANG THAI SW1,SW2 CO CHONG RUNG

;TRA VE MA SW1=1 HOAC MA SW2=2 VA CO SW\_FLG=1 NEU CO SW NHAN

;TRA VE CO SW\_FLG=1NEU KHOGN CO SW NHAN

;SU DUNG R16,R17, CO SW\_FLG THUOC THANH GHI FLAG\_REG

;------------------------------------

GET\_SW:

CBR REG\_FLAG,(1<<SW\_FLG) ;XOA CO BAO NHAN SW

BACK0:

LDI R16,50 ;KIEM TRA SW NHAN 50 LAN LIEN TUC

WAIT0:

IN R17,CONT\_IN

ANDI R17,(1<<SW1)|(1<<SW2) ;CHE BIT SW1,SW2

CPI R17,(1<<SW1)|(1<<SW2) ;KIEM TRA SW NHAN?

BREQ EXIT\_SW ;KHONG NHAN THOAT

DEC R16 ;CO NHAN TIEP TUC

BRNE WAIT0

PUSH R17 ; CAT MA SW

BACK1:

LDI R16,50 ;KIEM TRA SW NHA 50 LAN LIEN TUC

WAIT1:

IN R17,CONT\_IN

ANDI R17,(1<<SW1)|(1<<SW2)

CPI R17,(1<<SW1)|(1<<SW2)

BRNE BACK1 ;CHO NHA SW

DEC R16

BRNE WAIT1

POP R17 ;PHUC HOI MA SW

CPI R17,(1<<SW2) ;SW1=0 NHA, =1 LA KHONG NHAN

BRNE SW2\_CODE

LDI R17,1

RJMP SET\_FLG

SW2\_CODE:

CPI R17,(1<<SW1) ;SW2=0 NHAN, =1 LA KHONG NHAN

BRNE EXIT\_SW ;KHONG PHAI THOAT

LDI R17,2 ;GAN GIA TRI MA SW2

SET\_FLG:

SBR REG\_FLAG,(1<<SW\_FLG) ; DAT CO BAO NHAN SW

EXIT\_SW:

RET

;---------------------------------

;SET\_NUM CAI DAT CAC GIA TRI THOI GIAN CHON QUA BIEN COUNT

;NHAN/NHA SW1 THOAT

;NHAN/NHA SW2 CAI DAT GIA TRI

;SU DUNG R17,R18, CTC CURS\_POS, GET\_SW

;---------------------------------

SET\_NUM:

MOV R17,POS\_CRS ;DAT CON TRO DUNG VI TRI CAI DAT

RCALL CURS\_POS

SW\_CHK1:

RCALL GET\_SW ;DOC SW

SBRS REG\_FLAG,SW\_FLG ;CO SW NHAN

RJMP SW\_CHK1 ;CHO NHAN SW

CPI R17,1 ;SW1 NHAN?

BREQ EXIT\_NUM ; DUNG, THOAT

CPI R17,2 ;SW2 NHAN

BRNE SW\_CHK1 ;KHAC, DOC LAI SW

LD R17,X ; NAP GIA TRI CAI DAT

CPI COUNT,3 ; CAI DAT GIAY?

BRNE DAY\_CHK ;KHAC, KIEM TRA NGAY

CBR R17,(1<<STO) ;DUNG, XOA BIT ST

RJMP PRESET ;TIEN HANH DAT

DAY\_CHK:

CPI COUNT,0 ;CAI DAT NGAY?

BRNE PRESET ;KHAC, TIEN HANH DAT

ANDI R17,0X07 ;DOC LAY DATA NGAY

PRESET:

INC R17 ;TANG GIA TRI THEM 1

MOV R18,R17 ;CAT GIA TRI DAT

ANDI R17,$0F ;CHE LAY 4 BIT THAP

CPI R17,$0A ;GIA TRI <10

BRCS NON\_CR ;DUNG, KHONG TRAN

LDI R17,$06 ;HIEU DINH BCD

ADD R18,R17

NON\_CR:

MOV R17,R18 ;TRA SO BCD DAT VE R17

CP R17,NUM\_MAX ;SO SANH GIOI HAN MAX

BRCS DISP ;NHO HON, HIEN THI

BREQ DISP ;BANG, HIEN THI

MOV R17,NUM\_MIN ;LON HON, GIA VE GIOI HAN MIN

DISP:

ST X,R17 ;CAT SO BCD VAO BUFFER

RCALL NUM\_DISP ;HIEN THI SO BCD DAT

RJMP SET\_NUM ;TIEP TUC DAT

EXIT\_NUM:

INC COUNT ;TANG BIEN DEM VI TRI CAI DAT

RET

;--------------------------------------

;------------------------------------

NUM\_DISP:

PUSH R17 ;CAT DATA

SWAP R17 ;HOAN VI TRI 4 BIT CAO <-> THAP

ANDI R17,0X0F ;CHE LAY BCD CAO

ORI R17,0X30 ;CHUYEN SANG MA ASCII

SBI LCD,RS

LDI R16,1 ;CHO 100US

RCALL DELAY\_US

RCALL OUT\_LCD ;HIEN THI GIA TRI

POP R17 ;PHUC HOI DATA

ANDI R17,0X0F ;CHE LAY BCD THAP

ORI R17,0X30 ;CHUYEN SANG MA ASCII

SBI LCD,RS

LDI R16,1 ;CHO 100US

RCALL DELAY\_US

RCALL OUT\_LCD ; HIEN THI GIA TRI

RET

;------------------------------------------------------

;MSG\_DISP

;----------------------------

MSG\_DISP:

LPM R17,Z+ ;LAY MA ASCII KY TU TU FLASH ROM

CPI R17,NULL ;KIEM TRA KY TU KET THUC

BREQ EXIT\_MSG ;KY TU NULL THOAT

LDI R16,1 ;CHO 100US

RCALL DELAY\_US

SBI LCD,RS ;RS=1 GHI DATA HIEN THI LCD

RCALL OUT\_LCD ;GHI MA ASCII KY TU RA LCD

RJMP MSG\_DISP

EXIT\_MSG:

RET

;------------------------------------------------

;CURS\_POS

;------------------------------------------------

CURS\_POS:

LDI R16,1 ; CHO 100US

RCALL DELAY\_US

CBI LCD,RS ; RS=0 GHI LENH

RCALL OUT\_LCD

RET

;-----------------------------------------------------------------

;INIT\_LCD4 kh?i ??ng LCD ghi 4 byte mã l?nh theogiaoti?p4bit;Function set:R18=$28 2 dòng font 5x8 giao ti?p 4bit

;Clear display:R19=$01 xóa màn hình

;Display on/off LCDrol:R20=$0C màn hình on,contr?off;Entry mode set:R21=$06 d?ch ph?i con tr? ,?/c DDRAMt?ng1khighi data

;RS=bit0=0,RW=bit1=0

;----------------------------------------------------------------

INIT\_LCD4:

CBI LCD,RS ;RS=0: ghi lenh

MOV R17,R18 ;R18=Function set

RCALL OUT\_LCD ;ghi 1 byte dataraLCD

MOV R17,R19 ;R19=Clear display

RCALL OUT\_LCD

LDI R16,20 ;ch? 2ms sau lenhClear display

RCALL DELAY\_US

MOV R17,R20 ;R20=Display LCDrol on/off

RCALL OUT\_LCD

MOV R17,R21 ;R21=Entry mode set

RCALL OUT\_LCD

RET

;--------------------------------------------------

;OUT\_LCD4 ghi mã lenh/data ra LCD

;Input: R17 chia mã lenh/data 4 bit cao

;--------------------------------------------------

OUT\_LCD4:

OUT LCD,R17

SBI LCD,E

CBI LCD,E

RET

;------------------------------------------------------

;OUT\_LCD4\_2 ghi 1 byte mã l?nh/data ra LCD

;chia làm 2 l?n ghi 4bit

;Input: R17 ch?a mã l?nh/data,R16

;bit RS=0/1:l?nh/data,bit RW=0:ghi

;S? d?ng ctc OUT\_LCD4

;------------------------------------------------------

OUT\_LCD:

LDI R16,1

RCALL DELAY\_US

IN R16,LCD

ANDI R16,(1<<RS)

PUSH R16

PUSH R17

ANDI R17,$F0

OR R17,R16

RCALL OUT\_LCD4

LDI R16,1

RCALL DELAY\_US

POP R17

POP R16

SWAP R17

ANDI R17,$F0

OR R17,R16

RCALL OUT\_LCD4

RET

;------------------------------------------------------

;DELAY\_US t?o th?i gian tr? =R16x100?s(Fosc=8Mhz)

;Input:R16 h? s? nhân th?i gian tr? 1 ??n 255

;-------------------------------------------------------

DELAY\_US:

MOV R15,R16 ;1MC n?p data choR15

LDI R16,200 ;1MC s? d?ng R16

L1: MOV R14,R16 ;1MC n?p data choR14

L2: DEC R14 ;1MC

NOP ;1MC

BRNE L2 ;2/1MC

DEC R15 ;1MC

BRNE L1 ;2/1MC

RET ;4MC

;-----------------------------------------

;TWI\_INIT KHOI DONG CONG TWI

;DAT TOC DO TRUYEN=100KHZ

;------------------------------------------

TWI\_INIT:

LDI R17,8 ;TOC SO TRUYEN SCL=100KHZ

STS TWBR,R17

LDI R17,1

STS TWSR,R17 ;HE SO DAT TRUOC=4

LDI R17,(1<<TWEN) ;CHO PHEP TWI

STS TWCR,R17

RET

;---------------------------------------------------------

TWI\_START:

LDI R17,(1<<TWEN)|(1<<TWSTA)|(1<<TWINT)

STS TWCR,R17

WAIT\_STA:

LDS R17,TWCR

SBRS R17,TWINT

RJMP WAIT\_STA

RET

;----------------------------------------------------------

TWI\_WRITE:

STS TWDR,R17

LDI R17,(1<<TWEN)|(1<<TWINT)

STS TWCR,R17

WAIT\_WR:

LDS R17,TWCR

SBRS R17,TWINT

RJMP WAIT\_WR

RET

;----------------------------------------------------------

TWI\_READ:

LDI R17,(1<<TWEN)|(1<<TWINT)|(1<<TWEA)

STS TWCR,R17

WAIT\_RD:

LDS R17,TWCR

SBRS R17,TWINT

RJMP WAIT\_RD

LDS R17,TWDR

RET

;-----------------------------------------------------------

TWI\_NAK:

LDI R17,(1<<TWEN)|(1<<TWINT)

STS TWCR,R17

WAIT\_NAK:

LDS R17,TWCR

SBRS R17,TWINT

RJMP WAIT\_NAK

RET

;------------------------------------------------------------

TWI\_STOP:

LDI R17,(1<<TWEN)|(1<<TWSTO)|(1<<TWINT)

STS TWCR,R17

RET

;------------------------------------------------------------

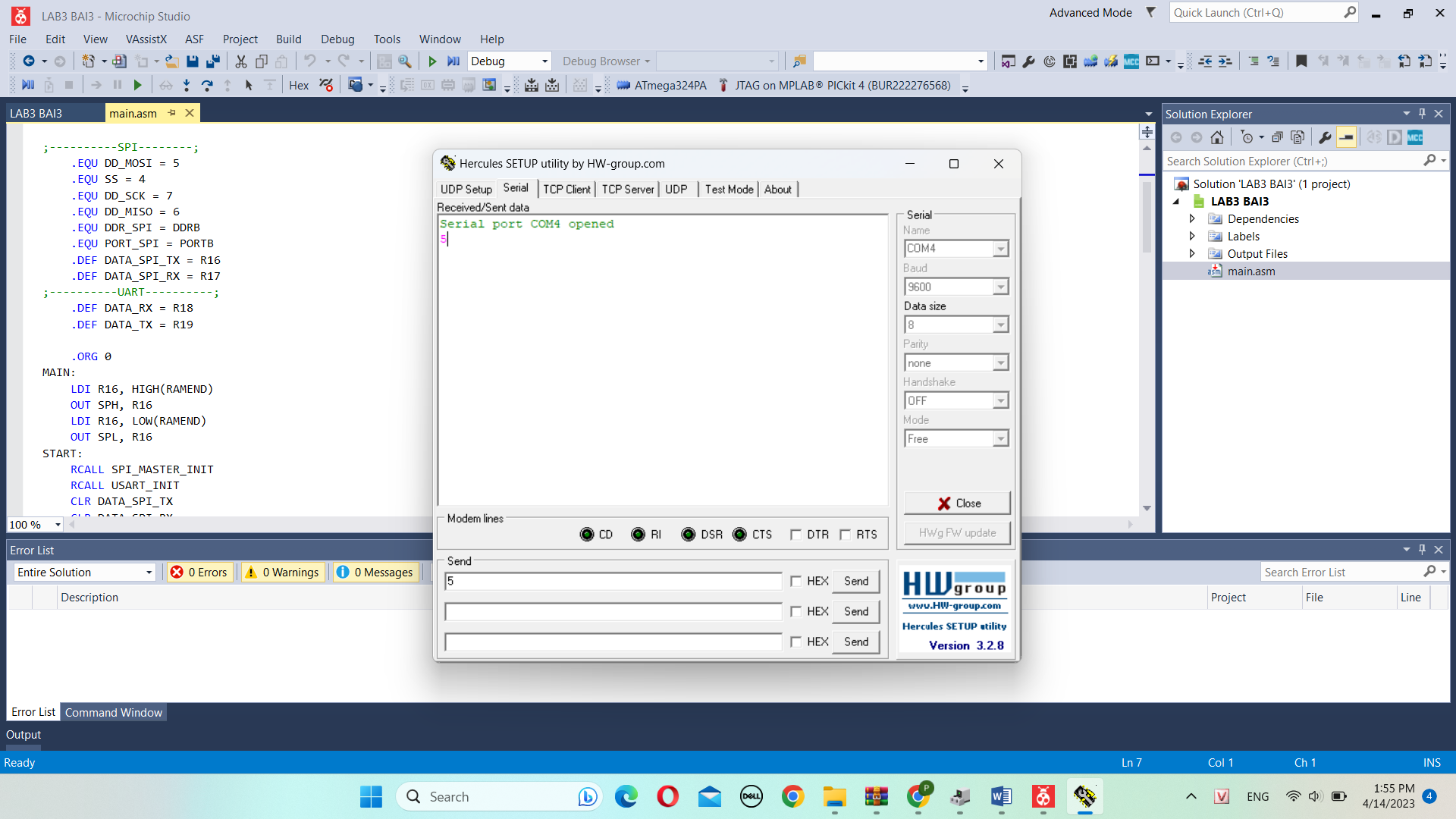
.ORG 0X200

MSG1: .DB"TIME: ",$00

MSG2: .DB"DATE: ",$00

# Bài 3

1. Kết nối các tín hiệu MOSI, SCK của port SPI từ AVR đến tín hiệu SDI và CLK của khối thanh ghi dịch. Kết nối 2 chân port khác vào tín hiệu nCLR và LATCH. Kết nối ngõ ra của thanh ghi dịch vào Bar LED
2. Kết nối các tín hiệu UART như ở bài 1.
3. Viết chương trình nhận 1 giá trị từ UART và xuất ra Bar Led sử dụng SPI.



;----------SPI--------;

.EQU DD\_MOSI = 5

.EQU SS = 4

.EQU DD\_SCK = 7

.EQU DD\_MISO = 6

.EQU DDR\_SPI = DDRB

.EQU PORT\_SPI = PORTB

.DEF DATA\_SPI\_TX = R16

.DEF DATA\_SPI\_RX = R17

;----------UART----------;

.DEF DATA\_RX = R18

.DEF DATA\_TX = R19

.ORG 0

MAIN:

LDI R16, HIGH(RAMEND)

OUT SPH, R16

LDI R16, LOW(RAMEND)

OUT SPL, R16

START:

RCALL SPI\_MASTER\_INIT

RCALL USART\_INIT

CLR DATA\_SPI\_TX

CLR DATA\_SPI\_RX

CLR DATA\_TX

CLR DATA\_RX

SBI PORT\_SPI, SS ;STOP TRANSMIT SPI

AGAIN:

RCALL USART\_RECEIVER\_CHAR

MOV DATA\_SPI\_TX, DATA\_RX

;TRANSMIT DATA

CBI PORT\_SPI, SS

RCALL SPI\_MASTER\_TRANSMIT

SBI PORT\_SPI, SS

RJMP AGAIN

SPI\_MASTER\_INIT:

;CONFIG I/O SPI

;MOSI, SCK, SS -> OUT

;MISO ->IN

LDI R17, (1<<DD\_MOSI)|(1<<DD\_SCK)|(1<<SS)

OUT DDR\_SPI, R17

SBI PORT\_SPI, DD\_MISO ;PULL UP

;SPI ENABLE, MASTER, Fosc/16, MODE 0

LDI R17, (1 << SPE0) | (1 << MSTR0) | (1 << SPR00)

OUT SPCR0, R17

RET

SPI\_MASTER\_TRANSMIT:

PUSH R17

OUT SPDR0, DATA\_SPI\_TX ;TRANSMIT DATA FROM MASTER TO SLAVE, OUT TO PIN MOSI

WAIT\_MASTER\_TRANSMIT:

IN R17, SPSR0

SBRS R17, SPIF0

RJMP WAIT\_MASTER\_TRANSMIT

IN DATA\_SPI\_RX, SPDR0 ;IN FROM PIN MISO

POP R17

RET

USART\_INIT:

LDI R16, (1<<RXEN0) | (1<<TXEN0) ;Enable transmitter and receiver

STS UCSR0B, R16

LDI R16, $00

STS UBRR0H, R16

LDI R16, 51 ;Set baudrate to 9600bps with 8Mhz clock

STS UBRR0L, R16

LDI R16, (1<<UCSZ01) | (1<<UCSZ00) ;8 bit data, no parity, 1 stop bit

STS UCSR0C, R16

RET

USART\_RECEIVER\_CHAR:

PUSH R17

WAIT\_USART\_RECEIVE:

LDS R17, UCSR0A

SBRS R17, RXC0 ;bit RXC0 = 1 then continue to receive char

RJMP WAIT\_USART\_RECEIVE

LDS DATA\_RX, UDR0 ;Read new data

POP R17

RET

# Bài 4

1. Kết nối các tín hiệu MOSI, MISO, SCK của port SPI từ AVR các tín hiệu tương ứng trên header J80. Kết nối 1 chân port khác vào tín hiệu nCS.
2. Kết nối các tín hiệu UART như ở bài 1.
3. Kết nối 1 port vào Bar LED.
4. Viết chương trình đếm số ký tự nhận được từ UART và xuất ra Bar Led, cứ mỗi lần có 1 byte nhận được, số đếm tăng lên 1 và được ghi vào EEPROM. Khi vi xử lý mất điện và có lại, số đếm được đọc ra từ EEPROM và lấy làm giá trị bắt đầu.

.EQU SPI\_PORT=PORTB

.EQU SPI\_PORT\_DDR=DDRB

.EQU SS=4

.EQU MOSI=5

.EQU MISO=6

.EQU SCK=7

.EQU WIP=0 ;kư hi?u bit báo b?n b? nh?

.EQU PE=$42 ;mă l?nh xóa 1 trang b? nh?

.EQU WREN=$06 ;mă l?nh cho phép ghi b? nh?

.EQU WRDI=$04 ;mă l?nh xóa cho phép ghi b? nh?

.EQU RDSR=$05 ;mă l?nh ??c tr?ng thái b? nh?

.EQU WRSR=$01 ;mă l?nh ghi tr?ng thái b? nh?

.EQU SPI\_RD=$03 ;mă l?nh ??c b? nh?

.EQU SPI\_WR=$02 ;mă l?nh ghi b? nh?

.EQU MEM\_BYTE3=0X00 ;??a ch? b? nh? byte 3 bit 23-16

.EQU MEM\_BYTE2=0X01 ;??a ch? b? nh? byte 2 bit 15-8

.EQU MEM\_BYTE1=0X00 ;??a ch? b? nh? byte 1 bit 7-0

.DEF COUNT\_REG=R20

.ORG 0

LDI R16,HIGH(RAMEND)

OUT SPH,R16

LDI R16,LOW(RAMEND)

OUT SPL,R16

RCALL USART\_INIT

RCALL SPI\_INIT

LDI R16,$FF

OUT DDRA,R16

LDI COUNT\_REG,0

CBI PORTB,SS ;Put EEPROM to active state

SBI PORTB,SS

LOOP:

RCALL EEPROM\_READ

RCALL UART\_RECEIVE

RCALL UART\_TRANS

INC COUNT\_REG ;TANG BIEN DEM

OUT PORTA,COUNT\_REG ;XUAT RA BARLED

RCALL EEPROM\_ERASE

RCALL EEPROM\_WRITE

RJMP LOOP

;??C S? ? ??A CHI 0X001000 TRONG EEPROM RA THANH GHI R20

EEPROM\_READ:

LDI R16,SPI\_RD ;mă l?nh ghi b? nh?

CBI PORTB,SS ;cho phép truy?n SPI

RCALL SPI\_TRANS ;truy?n SPI

LDI R16,MEM\_BYTE3 ;n?p ??a ch? byte 3 b? nh?

RCALL SPI\_TRANS ;truy?n SPI

LDI R16,MEM\_BYTE2 ;n?p ??a ch? byte 2 b? nh?

RCALL SPI\_TRANS ;truy?n SPI

LDI R16,MEM\_BYTE1 ;n?p ??a ch? byte 1 b? nh?

RCALL SPI\_TRANS ;truy?n SPI

LDI R16,$FF ;dump data ?? nh?n

RCALL SPI\_TRANS ;truy?n SPI

SBI PORTB,SS ;k?t thúc truy?n SPI

MOV R20,R17 ;d? li?u c?t vào R20

RET

;XÓA TRANG C?A EEPROM

EEPROM\_ERASE:

LDI R16,WREN ;mă l?nh cho phép ghi b? nh?

CBI SPI\_PORT,SS ;cho phép truy?n SPI

RCALL SPI\_TRANS ;truy?n SPI

SBI PORTB,SS ;SS=1 ch?t l?nh cho phép ghi

LDI R16,PE ;mă l?nh xóa trang b? nh?

CBI PORTB,SS ;cho phép truy?n SPI

RCALL SPI\_TRANS ;truy?n SPI

LDI R16,MEM\_BYTE3 ;n?p ??a ch? byte 3 b? nh?

RCALL SPI\_TRANS ;truy?n SPI

LDI R16,MEM\_BYTE2 ;n?p ??a ch? byte 2 b? nh?

RCALL SPI\_TRANS ;truy?n SPI

LDI R16,MEM\_BYTE1 ;n?p ??a ch? byte 1 b? nh?

RCALL SPI\_TRANS ;truy?n SPI

SBI SPI\_PORT,SS ;k?t thúc truy?n SPI

WR\_FIN: ;??c thanh ghi STATUS ki?m tra bit WIP=0 báo b? nh? r?i

LDI R16,RDSR ;mă l?nh ??c thanh ghi STATUS

CBI SPI\_PORT,SS ;cho phép truy?n SPI

RCALL SPI\_TRANS ;truy?n SPI

SBRC R17,WIP ;bit WIP=0 b? nh? r?i

RJMP WR\_FIN ;bit WIP=1 b? nh? b?n,ch? ti?p

SBI SPI\_PORT,SS ;k?t thúc truy?n SPI

RET

;GHI S? TRONG THANH GHI R19 RA ??A CHI 0X001000 TRONG EEPROM

EEPROM\_WRITE:

LDI R16,WREN ;mă l?nh cho phép ghi b? nh?

CBI SPI\_PORT,SS ;cho phép truy?n SPI

RCALL SPI\_TRANS ;truy?n SPI

SBI PORTB,SS ;SS=1 ch?t l?nh cho phép ghi

LDI R16,SPI\_WR ;mă l?nh ghi b? nh?

CBI PORTB,SS ;cho phép truy?n SPI

RCALL SPI\_TRANS ;truy?n SPI

LDI R16,MEM\_BYTE3 ;n?p ??a ch? byte 3 b? nh?

RCALL SPI\_TRANS ;truy?n SPI

LDI R16,MEM\_BYTE2 ;n?p ??a ch? byte 2 b? nh?

RCALL SPI\_TRANS ;truy?n SPI

LDI R16,MEM\_BYTE1 ;n?p ??a ch? byte 1 b? nh?

RCALL SPI\_TRANS ;truy?n SPI

MOV R16,COUNT\_REG

RCALL SPI\_TRANS

SBI SPI\_PORT,SS ;k?t thúc truy?n SPI

WR\_FIN0:;??c thanh ghi STATUS ki?m tra bit WIP=0 báo b? nh? r?i

LDI R16,RDSR ;mă l?nh ??c thanh ghi STATUS

CBI SPI\_PORT,SS ;cho phép truy?n SPI

RCALL SPI\_TRANS ;truy?n SPI

SBRC R17,WIP ;bit WIP=0 b? nh? r?i

RJMP WR\_FIN0 ;bit WIP=1 b? nh? b?n,ch? ti?p

SBI SPI\_PORT,SS ;k?t thúc truy?n SPI

RET

SPI\_INIT:

LDI R16,(1<<SS)|(1<<MOSI)|(1<<MISO)|(1<<SCK)

OUT SPI\_PORT\_DDR, R16 ; Output pins

LDI R16,(1<<SPE0)|(1<<MSTR0)|(1<<SPR00)

OUT SPCR0, R16 ; Enable, master, MSB first

LDI R16,(1<<SPI2X0) ; SCK=FOSC/8

STS SPSR0,R16

RET

SPI\_TRANS:

OUT SPDR0,R16 ;ghi data ra SPI

WAIT\_SPI:

IN R18,SPSR0 ;??c c? SPIF0

SBRS R18,SPIF0 ;c? SPIF0=1 truy?n SPI xong

RJMP WAIT\_SPI ;ch? c? SPIF0=1

IN R17,SPDR0

RET

USART\_INIT:

LDI R16,(1<<RXEN0)|(1<<TXEN0) ;Enable Receiver

STS UCSR0B, R16

LDI R16,(1<<UCSZ01)|(1<<UCSZ00) ;8-bit data

STS UCSR0C, R16 ;no parity, 1 stop bit

LDI R16,0x00

STS UBRR0H,R16

LDI R16,51 ;baud rate = 9600 and Fosc = 8 MHz

STS UBRR0L,R16

RET

UART\_RECEIVE:

LDS R17,UCSR0A

SBRS R17,RXC0

RJMP UART\_RECEIVE

LDS R16,UDR0

RET

UART\_TRANS:

LDS R17,UCSR0A

SBRS R17,UDRE0

RJMP UART\_TRANS

STS UDR0,R16

RET

DELAY1MS:

LDI R16,0

OUT TCNT0,R16

LDI R16,$00

OUT TCCR0A,R16

LDI R16,$05 ;0000 0011

OUT TCCR0B,R16

AGAIN: SBIS TIFR0,TOV0

RJMP AGAIN

SBI TIFR0,TOV0

LDI R16,$00

OUT TCCR0B,R16

RET

# Bài 5

1. Kết nối các tín hiệu UART như ở bài 1.
2. Kết nối 1 port vào Bar LED.
3. Viết chương trình đếm số ký tự nhận được từ UART và xuất ra Bar Led, cứ mỗi lần có 1 byte nhận được, số đếm tăng lên 1 và được ghi vào EEPROM nội của AVR. Khi vi xử lý mất điện và có lại, số đếm được đọc ra từ EEPROM nội và lấy làm giá trị bắt đầu.

.DEF COUNTER = R18

.DEF DATA\_UART\_RX = R19

.EQU P\_OUT = PORTA

.EQU DD\_OUT = DDRA

.ORG 0

MAIN:

LDI R16, HIGH(RAMEND)

OUT SPH, R16

LDI R16, LOW(RAMEND)

OUT SPL, R16

CLR COUNTER ;Initial counter = 0

SER R16

OUT DD\_OUT, R16 ;PORTA is out

LDI R20, $01

LDI R21, $00

RCALL USART\_INIT

START:

RCALL WRITE\_TO\_EEPROM

RCALL READ\_FROM\_EEPROM

OUT P\_OUT, COUNTER

RCALL USART\_RECEIVER\_CHAR

INC COUNTER

RJMP START

USART\_INIT:

;9600 BAUD RATE

LDI R16, $00

STS UBRR0H, R16

LDI R16, 51

STS UBRR0L, R16

;Enable reveiver

LDI R16, (1<<RXEN0)

STS UCSR0B, R16

;8 BIT, NO-PARITY, 1 STOP BIT

LDI R16, (1<<UCSZ01)|(1<<UCSZ00)

STS UCSR0C, R16

RET

USART\_RECEIVER\_CHAR:

PUSH R17

WAIT\_UART\_RECEIVE:

LDS R17, UCSR0A

SBRS R17, RXC0

RJMP WAIT\_UART\_RECEIVE

LDS DATA\_UART\_RX, UDR0

POP R17

RET

WRITE\_TO\_EEPROM:

WAIT\_ENABLE\_WRITE:

SBIC EECR, EEPE

RJMP WAIT\_ENABLE\_WRITE ;Wait EEPE = 0

;Load address to EEAR

OUT EEARH, R21

OUT EEARL, R20

OUT EEDR, COUNTER ;Load value to EEDR

SBI EECR, EEMPE

SBI EECR, EEPE

RCALL DELAY\_5MS

RET

READ\_FROM\_EEPROM:

WAIT\_READ:

SBIC EECR, EEPE

RJMP WAIT\_READ ;Wait EEPM = 0

;Load address for EEAR

OUT EEARH, R21

OUT EEARL, R20

SBI EECR, EERE

IN COUNTER, EEDR

RET

DELAY\_5MS:

LDI R20, 5

LP2:

LDI R21, 250

LP1:

NOP

DEC R21

BRNE LP1

DEC R20

BRNE LP2

RET