**Тема: Аналіз та синтез систем.**

; DECLARATION SYMBOLES

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;off\_time DS.B 1

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

; Ended DECLARATION SYMBOLES ZONE

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

BYTES

segment byte 'ram0'

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

; VARIABLES DECLARATION ZONE

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Variable for SPI data send (MASTER MODE)

SPI\_data ds.b 1

var ds.b 1

delay\_h ds.b 1

delay\_l ds.b 1

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

; Ended DECLARATION VARIABLES ZONE

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

WORDS

segment byte 'rom'

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

; CONSTANTES DECLARATION ZONE

;

; Value unit digit (Voltage goes from 0 up to 5 Volts) table1

table1 dc.b 0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1

dc.b 1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2

dc.b 2,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3

dc.b 3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,5

; Value dot unit digit (Voltage goes from 0.1 up to 0.9 Volts) table2

table2 dc.b 0,1,2,3,4,4,5,6,7,8,8,9,0,1,1,2

dc.b 3,4,5,5,6,7,8,8,9,0,1,2,2,3,4,5

dc.b 5,6,7,8,9,9,0,1,2,3,3,4,5,6,6,7

dc.b 8,9,0,0,1,2,3,3,4,5,6,7,7,8,9,0

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

; Ended CONSTANTES DECLARATION ZONE

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

; SUBROUTINE PROGRAM DECLARATION ZONE

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

init\_ST7:

clr MCCSR ; normal mode

ret

init\_ports:

ld A,#%10011011 ; Init port A with this value

ld PADDR,A ; init port A (PWM0 is in PA2)

ld PAOR,A

ld A,#%01110100

bset PBDR,#2 ; Set PB2 pin in level logic “1” load data into SPI

ld PBDDR,A ; init port B SPI clock (SCK = PB1 pin )

ld PBOR,A ; MOSI is PB3 and PB0 is for analog input

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

; Ended Subroutine program declaration ZONE;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

; SUBROUTINE NAME: init\_IO

; DESCRIPTION:

;

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

init\_IO:

ld A,#%10011011

ld PADDR,A

ld PAOR,A

ld A,#%01110100

bset PBDR,#2 ; Set the PB2 pin in level logic “1” load data into SPI

ld PBDDR,A

ld PBOR,A

ret

; - Program for ADC: Initializing the ADCSR register

select\_CH:

ld A,#%00000000 ; The CH0=CH1=CH2 = 0 means AIN0 (PB0)

; is selecting because the potentiometer (analog input

; signal) is connecting to this PB0 pin, this I/O must be

; configuring in High impedance

ld ADCCSR,A ; configuring the ADCCSR register with %00000000

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

; GET DATA FROM ADC

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; - Launch the conversion of the input analog signal

process\_adc:

bset ADCCSR,#5 ; Launch the ADC for one conversion

cont:

ld A,ADCCSR ; load the content of the ADCCSR in the “A” register

and A,#$80 ; the “And” allows verifying the end conversion of ADC

jreq cont ; if not, this means the bit « OEC » is not yet equal 1

; This bit7 “OEC” of the ADCCSR register

bres ADCCSR,#5 ; Stop the ADC by setting the bit 5 of the ADCCSR

; bit5 = ADON in the ADCCSR register

; begin reading the conversion result of conversion

ld A,ADCDRH ; Load the MSB of the ADC in the A register

; “var” is a variable reserved at the begin in RAM0

ld var,A ; var content now the ADC conversion result

; mean the conversion result is storing in the “var”

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

; INIT SPI

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

init\_SPI:

ld A,#$03 ;INIT SPISR

ld SPISR,A

ld A,#%01011100 ;INIT SPICR

ld SPICR,A

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

; INIT SPI

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

delay:

push x

push y

ld x, delay\_l

ld y, delay\_h

dec\_2:

dec x

JRNE dec\_2

ld x,delay\_l

dec y

JRNE dec\_2

pop y

pop x

ret

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

; MAIN PROGRAM

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

main:

rsp

ld a, #%01111100

ld $04, a

ld $05, a

ld a, #%00110000

ld $03, a ; stop motor 1

ld a, #%00011111

ld $01, a

ld $02, a

ld a, #%00000000

ld $00, a ; stop motor 2

;=======================================================

clr $80

LD A,#%00010000 ; завантажити маску джерела таймеру

LD $0D,A ; джерело таймеру CNTR1 від так тової частоти

LD A,#%00001111 ;1000

LD $10,A ; ATR1 high

LD $1b,A ; DCR2 high (pa4)

LD A,#%00010000 ; завантажити маску вибору ШИМ на PA3

LD $12,A ; вибір ШИМ

LD A,#%10000000

LD $11,A ; ATR1 low (0..255)

LD A,#%10000001 ; значення = 1

LD $1C,A ; DCR2 low (pa4)

LD A,#1 ; завантажити маску дозволу зміни ширини

LD $21,A ; дати дозвіл на зміну ширини імпульсу

;============================================================================

;jp MOTOR2

MOTOR1:

;============================================================================

call Read\_Signal

cp A, #10 ;Еталонне значення

;============================================================================

;==============================our kroks============================

ld a, #10

CALL VVERH1

ld a, #10

CALL VLIVO1

ld a, #10

CALL VNIZ1

ld a, #10

CALL VVERH1

ld a, #10

CALL VVERH1

ld a, #10

CALL VNIZ1

ld a, #10

CALL VPRAVO1

ld a, #10

CALL VVERH1

ld a, #10

CALL VNIZ1

ld a, #10

CALL VNIZ1

jp MOTOR1

;=============================================

;==============================================

VLIVO1:

push a

ld a, #%01110100 ;song

ld $03, a

call PAUSE\_S

ld a, #%00100100

ld $03, a

call PAUSE\_S

ld a, #%01100000 ;song

ld $03, a

call PAUSE\_S

ld a, #%00000000

ld $03, a

call PAUSE\_S

ld a, #%00010000

ld $03, a

call PAUSE\_S

ld a, #%00011000

ld $03, a

call PAUSE\_S

ld a, #%00111000

ld $03, a

call PAUSE\_S

ld a, #%00111100

ld $03, a

call PAUSE\_S

pop a

dec a

jrne VLIVO1

ld a, #%00110000

ld $03, a ; stop motor 1

RET

;==============================================

VPRAVO1:

push a

;8

ld a, #%00111100

ld $03, a

call PAUSE\_S

;7

ld a, #%00111000

ld $03, a

call PAUSE\_S

;6

ld a, #%00011000

ld $03, a

call PAUSE\_S

;5

ld a, #%00010000

ld $03, a

call PAUSE\_S

;4

ld a, #%00000000

ld $03, a

call PAUSE\_S

;3

ld a, #%00100000

ld $03, a

call PAUSE\_S

;2

ld a, #%00100100

ld $03, a

call PAUSE\_S

;1

ld a, #%00110100

ld $03, a

call PAUSE\_S

pop a

dec a

jrne VPRAVO1

ld a, #%00110000

ld $03, a ; stop motor 1

RET

;==============================================

VNIZ1:

push A

ld a, #%00000001 ;1

ld $00, a

call PAUSE\_S

ld a, #%00000011 ;1-2

ld $00, a

call PAUSE\_S

ld a, #%00000010 ;2

ld $00, a

call PAUSE\_S

ld a, #%00000110 ;2-3

ld $00, a

call PAUSE\_S

ld a, #%00000100 ;3

ld $00, a

call PAUSE\_S

ld a, #%00001100 ;3-4

ld $00, a

call PAUSE\_S

ld a, #%00001000 ;4

ld $00, a

call PAUSE\_S

ld a, #%00001001 ;4-1

ld $00, a

call PAUSE\_S

pop A

dec A

jrne VNIZ1

ld A,#%00000000

ld $00,A ; stop motor 2

RET

;==============================================

VVERH1:

push A

ld a, #%00001000 ;4

ld $00, a

call PAUSE\_S

ld a, #%00001100 ;3-4

ld $00, a

call PAUSE\_S

ld a, #%00000100 ;3

ld $00, a

call PAUSE\_S

ld a, #%00000110 ;2-3

ld $00, a

call PAUSE\_S

ld a, #%00000010 ;2

ld $00, a

call PAUSE\_S

ld a, #%00000011 ;1-2

ld $00, a

call PAUSE\_S

ld a, #%00000001 ;1

ld $00, a

call PAUSE\_S

ld a, #%00001001 ;4-1

ld $00, a

call PAUSE\_S

pop A

dec A

jrne VVERH1

ld A,#%00000000

ld $00,A ; stop motor 2

RET

;=============================================

DIA1:

push A

ld a, #%00000001 ;1

ld $00, a

;8

ld a, #%00111100

ld $03, a

call PAUSE\_S

ld a, #%00000011 ;1-2

ld $00, a

;7

ld a, #%00111000

ld $03, a

call PAUSE\_S

ld a, #%00000010 ;2

ld $00, a

;6

ld a, #%00011000

ld $03, a

call PAUSE\_S

ld a, #%00000110 ;2-3

ld $00, a

;5

ld a, #%00010000

ld $03, a

call PAUSE\_S

ld a, #%00000100 ;3

ld $00, a

;4

ld a, #%00000000

ld $03, a

call PAUSE\_S

ld a, #%00001100 ;3-4

ld $00, a

;3

ld a, #%00100000

ld $03, a

call PAUSE\_S

ld a, #%00001000 ;4

ld $00, a

;2

ld a, #%00100100

ld $03, a

call PAUSE\_S

ld a, #%00001001 ;4-1

ld $00, a

;1

ld a, #%00110100

ld $03, a

call PAUSE\_S

pop A

dec A

jrne DIA1

ld a, #%00110000

ld $03, a ; stop motor 1

ld A,#%00000000

ld $00,A ; stop motor 2

RET

;==============================================

DIA2:

push a

ld a, #%01110100 ;song

ld $03, a

ld a, #%00001000 ;4

ld $00, a

call PAUSE\_S

ld a, #%00100100

ld $03, a

ld a, #%00001100 ;3-4

ld $00, a

call PAUSE\_S

ld a, #%01100000 ;song

ld $03, a

ld a, #%00000100 ;3

ld $00, a

call PAUSE\_S

ld a, #%00000000

ld $03, a

ld a, #%00000110 ;2-3

ld $00, a

call PAUSE\_S

ld a, #%00010000

ld $03, a

ld a, #%00000010 ;2

ld $00, a

call PAUSE\_S

ld a, #%00011000

ld $03, a

ld a, #%00000011 ;1-2

ld $00, a

call PAUSE\_S

ld a, #%00111000

ld $03, a

ld a, #%00000001 ;1

ld $00, a

call PAUSE\_S

ld a, #%00111100

ld $03, a

ld a, #%00001001 ;4-1

ld $00, a

call PAUSE\_S

pop a

dec a

jrne DIA2

ld a, #%00110000

ld $03, a ; stop motor 1

ld A,#%00000000

ld $00,A ; stop motor 2

RET

;==============================================

;==============================================

PAUSE\_S:

ld a, #6

l\_pause:

push a

ld a, #255

l\_pa:

bset $2e, #6

dec a

jrne l\_pa

pop a

dec a

jrne l\_pause

ret

Read\_Signal:

LD A,#%01100001 ; Завантажити в A маску настройки АЦП для

; PB0

LD $34,A ; Записати її в регістр ADCCSR, що дає дозвіл ан

;алогового вводу

; з ніжки PB0.

M1:

BTJF $34,#7,M1 ; Переписати біт EOC в CF і перейти на M1,

;якщо він нульовий.

LD A,$35 ; Зчитати в регістр A отримане значення.

ret

dummy\_rt:

IRET ; Empty Procedure : Mean return to Main program.

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

; VECTORS D’INTERRUPTION DECLARATION

;

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

segment 'vectit'

DC.W dummy\_rt ; Adresse FFE0-FFE1h

SPI\_it DC.W dummy\_rt ; Adresse FFE2-FFE3h

lt\_RTC1\_it DC.W dummy\_rt ; Adresse FFE4-FFE5h

lt\_IC\_it DC.W dummy\_rt ; Adresse FFE6-FFE7h

at\_timerover\_it DC.W dummy\_rt ; Adresse FFE8-FFE9h

at\_timerOC\_it DC.W dummy\_rt ; Adresse FFEA-FFEBh

AVD\_it DC.W dummy\_rt ; Adresse FFEC-FFEDh

DC.W dummy\_rt ; Adresse FFEE-FFEFh

lt\_RTC2\_it DC.W dummy\_rt ; Adresse FFF0-FFF1h

ext3\_it DC.W dummy\_rt ; Adresse FFF2-FFF3h

ext2\_it DC.W dummy\_rt ; Adresse FFF4-FFF5h

ext1\_it DC.W dummy\_rt ; Adresse FFF6-FFF7h

ext0\_it DC.W dummy\_rt ; Adresse FFF8-FFF9h

AWU\_it DC.W dummy\_rt ; Adresse FFFA-FFFBh

softit DC.W dummy\_rt ; Adresse FFFC-FFFDh

reset DC.W main ; Adresse FFFE-FFFFh

END

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*