void savesetup(void) // menu save\_setup

{

fxx=1.2;

// rom\_fxx=fxx; // запис в ROM значення частоти

// tp= TMAX;

}

// eeprom float rom\_fxx; // частота з ROM

volatile float fxx=22.2; // опорна частота- можна встановити від 1... 29999 Гц

-----------------------------------------------------------------------------------------------------------------------

// функція меню - set speed до 999 ms

unsigned int set\_speed(void)

{

unsigned char x[3] = { 7, 8, 9 }, i = 0; // позиції символів число з 3 значень

char time\_chars[3][2] = {"^", "^", "^"}; // нижній курсор - стрілка чотири позиції

lcd\_gotoxy(x[0],1);

lcd\_puts(time\_chars[0]);

if (speed>=1000) {speed=999;}; // перше "^" - курсор нижній

A0 = speed; V0=A0/1000;

value[0]=(unsigned char)(A0/1000); // старший-перший символ зліва

A1=A0-V0\*1000; V1= (A1/100);

value[1]=(unsigned char)(A1/100); // другий символ зліва

A2=A1-V1\*100; V2=A2/10;

value[2]=(unsigned char)(A2/10); // третій символ зліва

while (1) {

PREV\_PINF = PINF;

sprintf(buffer, "%u%u%u", value[0], value[1], value[2]); // три символи для виводу

lcd\_gotoxy(x[0],0);

lcd\_puts(buffer);

if (get\_key\_status(SELECT\_PLUS\_BTN)) // select+ UP --- set speed -

if (!get\_prev\_key\_status(SELECT\_PLUS\_BTN) || (gSelectPlusCounter == CNTREPEAT))

{

if (gSelectPlusCounter == CNTREPEAT)

delay\_ms(80);

switch (i)

{

case 0: // ліва\_старша=нульова позиція числа page значення=( 0..9)

if (value[0] < 9)

value[0]++;

else value[0] = 0;

break;

case 1: // друга позиція зліва числа page значення =( 0..9)

if (value[1] < 9)

value[1]++;

else value[1] = 0;

break;

case 2: // третя позиція зліва числа page значення =( 0..9)

if (value[i] < 9)

value[i]++;

else value[i] = 0;

break;

}

}

//-end select+ UP ---set speed ()--

if (get\_key\_status(SELECT\_MINUS\_BTN)) // select- DOWN--- set speed --

if (!get\_prev\_key\_status(SELECT\_MINUS\_BTN) || (gSelectMinusCounter == CNTREPEAT))

{

if (gSelectMinusCounter == CNTREPEAT)

delay\_ms(80);

switch (i)

{

case 0:

if (value[0] > 0)

value[0]--;

else value[0] = 2;

break;

case 1:

if (value[1] > 0)

value[1]--;

else value[1] = 9;

break;

case 2:

if (value[i] > 0)

value[i]--;

else value[i] = 9;

break;

}

}

//-end select- DOWN--- set speed ()-

if (get\_key\_status(MENU\_ENTER\_BTN)) // enter--- set speed ()-

if (!get\_prev\_key\_status(MENU\_ENTER\_BTN))

{

if (i==2) // кількість символів числа =3

{

speed = (unsigned int)(value[0])\*100 ;

speed = speed +(unsigned int)(value[1])\*10;

speed = speed +(unsigned int)(value[2]);

if (speed > =1000)

speed =999;

lcd\_gotoxy(x[i],1); // затерти курсор

lcd\_putchar(' ');

return(speed); // сформоване значення

}

lcd\_gotoxy(x[i],1);

lcd\_putchar(' ');

lcd\_gotoxy(x[++i],1);

lcd\_puts(time\_chars[i]);

}

//-end enter--- set speed

if (get\_key\_status(ESC\_BTN)) // exit --set\_page()-

{

PREV\_PINF = PINF;

return;

}

}

}

//--end exit --- set speed

-----------------------------------------------------------------

unsigned int Dred, Dgreen, Dblue; // duty = red green blue

unsigned int ICR3, RedPwm, GreenPwm, BluePwm ;

***// Timer 3 overflow interrupt service routine***

**interrupt [TIM3\_OVF] void timer3\_ovf\_isr(void)**

**{**

**OCR3AH=redpwmh; OCR3AL=redpwml; *// RED***

**OCR3BH=greenpwmh; OCR3BL=greenpwml; *// GREEN***

**OCR3CH=bluepwmh; OCR3CL=bluepwml; *// BLUE***

**}**

unsigned int **Red(**unsigned int **dutyr) *// функція для PWM - red***

**{**

**RedPwm = (ICR3/100)\* dutyr;**

if (RedPwm==0) RedPwm=1;

redpwml=(char)(RedPwm); // Dred% pwm RED - low byte

redpwmh=(char)((RedPwm)>>8); // Dred% pwm RED - high byte

return;

}

unsigned int **Green (**unsigned int **dutyg)** *// функція для PWM - Green*

{

**GreenPwm = (ICR3/100)\*** **dutyg;**

if (GreenPwm==0) GreenPwm=1;

greenpwml=(char)(GreenPwm);

greenpwmh=(char)((GreenPwm)>>8); *// Dgreen% pwm GREEN - high byte*

return;

}

unsigned int **Blue (**unsigned int **dutyb)** *// функція для PWM - Blue*

**{**

**BluePwm = (ICR3/100)\* dutyb;**

if (BluePwm==0) BluePwm=1;

bluepwml=(char)(BluePwm);

bluepwmh=(char)((BluePwm)>>8); *// Dblue% pwm BLUE - high byte*

return;

}

**void test (**unsigned int **speedx);**

**{**

**unsigned char R=0, G=0, B=0;**

**for (R=0; R<100; R++) *// red від 0 до 100 %***

**{**

**Red(R);**

**delay\_ms(speedx);**

**};**

**for (R=99; R>0; R--) *// red від 100% до 1%***

**{**

**Red(R);**

**delay\_ms(speedx);**

**};**

**for (G=0; G<100; G++) *// Green* *від 0 до 100 %***

**{**

**Green (G);**

**delay\_ms(speedx);**

**};**

**for (G=99; G>0; G--) *// Green* *від 100% до 1%***

**{**

**Green (G);**

**delay\_ms(speedx);**

**};**

**for (B=0; B<100; B++) *// Red Green* *Blue* *від 0 до 100 %***

**{**

**Red(B);**

**Green (B);**

**Blue (B);**

**delay\_ms(speedx);**

**};**

**for (B=99; B>0; B--) *//* Red *=100% Green* -*Blue* *від 100% до 1%***

**{**

**Green (B);**

**Blue (B);**

**delay\_ms(speedx);**

**};**

**}**

**----------------------------------------------------------------------**

**eeprom float rom\_fxx; // частота з ROM**

**fxx=rom\_fxx; // прочитати опорну частоту з памяті в змінну fxx**

**if (fxx> 30000.1) fxx=29999.0;**

**// fxx=99,9;**

**fxcalculator(); // функція для обчислення значень регістрів TCCR3B, ICR3H, ICR3L**

**//ініціалізація кнопок menu на порті F**

**void savesetup(void) // функція menu save\_setup**

**{**

**// fxx=1.2;**

**rom\_fxx=fxx; // запис в ROM значення частоти**

**// tp= TMAX;**

**}**