// --- Ligações entre PIC e LCD ---

sbit LCD\_RS at RE2\_bit; // PINO 2 DO PORTD INTERLIGADO AO RS DO DISPLAY

sbit LCD\_EN at RE1\_bit; // PINO 3 DO PORTD INTERLIGADO AO EN DO DISPLAY

sbit LCD\_D7 at RD7\_bit; // PINO 7 DO PORTD INTERLIGADO AO D7 DO DISPLAY

sbit LCD\_D6 at RD6\_bit; // PINO 6 DO PORTD INTERLIGADO AO D6 DO DISPLAY

sbit LCD\_D5 at RD5\_bit; // PINO 5 DO PORTD INTERLIGADO AO D5 DO DISPLAY

sbit LCD\_D4 at RD4\_bit; // PINO 4 DO PORTD INTERLIGADO AO D4 DO DISPLAY

// Selecionando direção de fluxo de dados dos pinos utilizados para a comunicação com display LCD

sbit LCD\_RS\_Direction at TRISE2\_bit; // SETA DIREÇAO DO FLUXO DE DADOS DO PINO 2 DO PORTD

sbit LCD\_EN\_Direction at TRISE1\_bit; // SETA DIREÇAO DO FLUXO DE DADOS DO PINO 3 DO PORTD

sbit LCD\_D7\_Direction at TRISD7\_bit; // SETA DIREÇAO DO FLUXO DE DADOS DO PINO 7 DO PORTD

sbit LCD\_D6\_Direction at TRISD6\_bit; // SETA DIREÇAO DO FLUXO DE DADOS DO PINO 6 DO PORTD

sbit LCD\_D5\_Direction at TRISD5\_bit; // SETA DIREÇAO DO FLUXO DE DADOS DO PINO 5 DO PORTD

sbit LCD\_D4\_Direction at TRISD4\_bit; // SETA DIREÇAO DO FLUXO DE DADOS DO PINO 4 DO PORTD

char ucRead; // Variavel para armazenar o dado lido.

unsigned int uiContador0; // Variavel de armazenamento do contador.

unsigned int uiContador1; // Variavel de armazenamento do contador.

unsigned int uiValor0; // Variavel auxiliar para exibido do contador.

unsigned int uiValor1; // Variavel auxiliar para exibido do contador.

unsigned char ucStatus\_inc0; // Variavel de travamento do incremento.

unsigned char ucStatus\_inc1; // Variavel de travamento do incremento.

int controle\_x; // Variavel que faz o controle do envio de notificação da falta de remédios.

char uiContador00[7]; // Variavel de armazenamento do contador.

char uiContador11[7]; // Variavel de armazenamento do contador.

void interrupt(){

INTCON.INT0IF = 0; // Limpa o flag sinalizador de interrupção externa INT0

controle\_x =0; // Retorna o controle de notificação para 0 pois a reposição foi executada.

while(PORTB.RB3==1){

if((PORTB.RB1==0)&&(ucStatus\_inc0==0)){ // Incrementa o remedio 1 somente uma vez quando a tecla for pressionada.

ucStatus\_inc0=1; // Variavel de travamento do incremento.

uiContador0++; // Esse operador aritmetico(++) realiza o mesmo que variavel = variavel + 1.

if(uiContador0>99){ // Define o valor maximo a ser mostrado no displays como 99.

uiContador0=99;

}

}

if((PORTB.RB1==1)&&(ucStatus\_inc0==1)){ // Volta a disponibilizar a opção de incremento quando a tecla for solta.

ucStatus\_inc0=0;

}

if((PORTB.RB2==0)&&(ucStatus\_inc1==0)){ // Incrementa o remedio 2 somente uma vez quando a tecla for pressionada.

ucStatus\_inc1=1; // Variavel de travamento do incremento.

uiContador1++; // Esse operador aritmetico(++) realiza o mesmo que variavel = variavel + 1.

if(uiContador1>99){ // Define o valor maximo a ser mostrado no displays como 99.

uiContador1=99;

}

}

if((PORTB.RB2==1)&&(ucStatus\_inc1==1)){ // Volta a disponibilizar a opção de incremento quando a tecla for solta.

ucStatus\_inc1=0;

}

}

}

void main(){

unsigned char ucMask[]= {0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};

/\* Variavel do tipo vetor que armazena os valores correspondentes ao numero

para ser mostrado no display de 7 segmentos \*/

ADCON1 = 0x0f; //Configura os pinos do PORTB como digitais

Lcd\_Init(); //Inicializa modulo LCD

Lcd\_Cmd(\_LCD\_CURSOR\_OFF); //Apaga cursor

Lcd\_Cmd(\_LCD\_CLEAR); //Limpa display

TRISB.RB0=1; // RB0 congifurado como entrada.

TRISB.RB1=1; // RB1 congifurado como entrada.

TRISB.RB2=1; // RB2 congifurado como entrada.

TRISB.RB3=1; // RB3 congifurado como entrada.

TRISC.RC1=0; // PORT C configurado como saida.

TRISA.RA2=0; // Define o pino RA2 do PORTA como saida(Seleção Display 1).

TRISA.RA3=0; // Define o pino RA3 do PORTA como saida(Seleção Display 2).

TRISA.RA4=0; // Define o pino RA4 do PORTA como saida(Seleção Display 3).

TRISA.RA5=0; // Define o pino RA5 do PORTA como saida(Seleção Display 4).

INTCON.GIE = 1;

INTCON.INT0IE = 1;

INTCON.INT0IF = 0;

INTCON2.INTEDG0 = 0;

uiContador0=5; // Inicializa a variavel com o valor

uiContador1=6; // Inicializa a variavel com o valor

controle\_x =0; // Variavel que faz o controle do envio da notificação de falta de remédios

ucStatus\_inc0=0;

ucStatus\_inc1=0;

UART1\_Init(9600); // Utiliza bibliotecas do compilador para configuração o Baud rate.

while(1){

// Confere se o caracter 'b' foi recebido, caso tenha ele ativa o buzzer indicando que está na hora de tomar um medicamento.

if(UART1\_Data\_Ready()){ // Verifica se um dado foi recebido no buffer

ucRead = UART1\_Read(); // Le o dado recebido do buffer.

if(ucRead== 'b'){

PORTC.RC1 = 0;

}

// Confere se o caracter '0' foi recebido, caso tenha ele desliga o buzzer e decrementa 1 no valor do contador do remedio 0.

if(ucRead== '0'){

PORTC.RC1 = 1;

uiContador0--;

}

// Confere se o caracter '1' foi recebido, caso tenha ele desliga o buzzer e decrementa 1 no valor do contador do remedio 1.

if(ucRead== '1'){

PORTC.RC1 = 1;

uiContador1--;

}

// Confere se o caracter '2' foi recebido, caso tenha ele desliga o buzzer pois há um remédio atrasado.

if(ucRead== '2'){

PORTC.RC1 = 1;

}

}

uiValor0 = uiContador0; // Coloca o conteudo da variavel do contador na varaivel auxiliar.

uiValor1 = uiContador1; // Coloca o conteudo da variavel do contador na varaivel auxiliar.

IntToStr(uiContador0,uiContador00); //Converte o valor 0 para String para que o mesmo possa ser impresso no LCD.

IntToStr(uiContador1,uiContador11); //Converte o valor 1 para String para que o mesmo possa ser impresso no LCD.

lcd\_out(1,1,"remedio0");

lcd\_out(1,9,"remedio1");

lcd\_out(2,1,uiContador00);

lcd\_out(2,10,uiContador11);

// Confere se está na hora de repor os remédios

if((uiValor0<=3 || uiValor1<=3) && controle\_x ==0){

UART1\_Write('x');

controle\_x=1;

}

// Mostra valor do remedio 0 no Display

PORTD = ucMask[uiValor1%10];

PORTA.RA5 = 1;

Delay\_ms(2);

PORTA.RA5 = 0;

uiValor1/=10;

// Mostra valor da dezena do remedio 0.

PORTD = ucMask[uiValor1%10];

PORTA.RA4 = 1;

Delay\_ms(2);

PORTA.RA4 = 0;

uiValor1/=10;

//Mostra valor do remedio 1 no Display

PORTD = ucMask[uiValor0%10];

PORTA.RA3 = 1;

Delay\_ms(2);

PORTA.RA3 = 0;

uiValor0/=10;

// Mostra valor do dezena do remedio 1.

PORTD = ucMask[uiValor0%10];

PORTA.RA2 = 1;

Delay\_ms(2);

PORTA.RA2 = 0;

uiValor0/=10;

}

}