MICROPROCESSADORES

ANO LECTIVO: 2023/2024

CURSO: ENG. ELETROTÉCNICA E DE COMPUTADORES

REGIME: D TURMA: PL3

MINIPROJECTO PROPOSTA 9 - VERSÃO C

* + FOSC=8 MHz
  + Baud=9600 bps
  + MPX4250 pressure sensor
  + Bipolar stepper motor (MOTOR-BISTEPPER)
  + Display TFT gráfico c/ resolução 240x320 pixels c/ driver ILI9341
  + LM017L 32x2 alphanumeric LCD
  + 25LC256 256 kbit I2C serial EEPROM

AUTORES:

|  |  |
| --- | --- |
| NOME: Diogo Cravo | NOME: Paulo Sousa |
| Nº ALUNO: 2222030 | Nº ALUNO: 2222031 |
|  |  |
|  |  |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Resumo do Projeto**

|  |
| --- |
| Neste projeto utilizamos um led e um buzzer para o alarme e um botão para desativar o alarme  Motor de passo – controlo do grau de abertura da válvula  Sessor de pressão – Obtém o valor da pressão  LED 0.5 Hz – Timer 0  Buzzer 2 Hz – Timer2 (ECCP) BUZZER  Timer1 – obter valor do adc  Botão interrupção externa – INT0 (desativar alarme)  Eusart1 – comunicação serie  SPI - display  ADC – sensor de pressão  **Alarme**  Quando o valor baixa dos 30kPa - o alarme **liga** (pressão baixa)  Quando o valor passa dos 120kPa - o alarme **liga** (pressão alta)  Estes valores de limite podem ser alterados  Interface serie: (**usart**)  Controlo **automático** da válvula (motor de passo)  • 0º de abertura - 0% (válvula totalmente fechada) – pressão inferior a 30kPa;  • 90º de abertura - 25% - pressão entre 30kPa e 60kPa;  • 180º de abertura - 50% - pressão entre 60kPa e 90kPa;  • 270º de abertura - 75% - pressão entre 90kPa e 120kPa  • 360º de abertura - 100% (válvula totalmente aberta) – pressão superior a 120kPa;  Controlo **manual** da válvula (motor de passo)  O grau de abertura da válvula será definido em função da percentagem do ângulo máximo  (0%, 25%, 50%, 75% e 100%).  Poder alterar entre controlo automático e manual;  Alteração dos valores críticos limite de pressão, mínimo e máximo, para alarme  Visualizar o valor atual da pressão na câmara;  Visualizar o valor atual do grau de abertura de válvula de admissão de ar;  Display mostra a pressão, os graus da válvula e se o alarme for ativado mostra a causa do alarme (baixa/alta pressão) |

**Explicação dos Registos**

|  |
| --- |
| 1.**Interrupcao** **INT0 (desativar alarme)**    ●RCONbits.IPEN = 1; /\* Activa prioridades nas interrupções \*/  /\*Funcoes baixa e alta prioridade  ●INTCONbits.GIEH = 1;  ●INTCONbits.GIEL = 1;  (INT0 é sempre alta perioridade)  ●INTCON INTCONbits.INT0IF = 0; //Flag da interrupcao  ●INTCONbits.INT0IE = 1; //Activação da interrupção  ●INTCON2bits.INTEDG0 = 0; //Flanco descendente |

|  |
| --- |
| 2. **Timer 0**  **Cálculos:**  O sinal luminoso deverá ser ligado com a intermitência de 2 Hz  ● F=2Hz, com DC=50%  ● TH=TL=500/2 = 250ms  ● Para CONTInicial = 0  ● Logo, **Prescaler = 8**  ● Depois, calculamos o CONTInicial  **Registos:**  ● TMR1 = 15536; //Inicialização do temporizador a 15536  ● T1CONbits.TMR1CS= 0; //Seleção de Relógio Interno (Fosc/4)  ● T1CONbits.T1CKPS= 0b10; //Prescaler a 4  ● T1CONbits.T1OSCEN = 0; //Oscilador secundário desligado  ● T1CONbits.T1SYNC = 0; // Relógio sincronizado com oscilador externo  ● T1CONbits.RD16 = 1; //Lê 16 bits  ● T1CONbits.TMR1ON = 1; //Inicia a contagem  //Configuração da interrupção do Timer 1  ● PIE1bits.TMR1IE = 1; //Autoriza a Interrupção  ● PIR1bits.TMR1IF = 0; // Apaga a flag de Interrupção  ● IPR1bits.TMR1IP = 0; // Interrupção de Baixa prioridade |

|  |
| --- |
| 3. **Timer 1 (ADC)**  **Cálculos:**  ● F=2Hz, com DC=50%  ● TH=TL=500/2 = 250ms  ● Para CONTInicial = 0  ● Logo, **Prescaler = 8**  ● Depois, calculamos o CONTInicial  **Registos:**  ● TMR1 = 15536; //Inicialização do temporizador a 15536  ● T1CONbits.TMR1CS= 0; //Seleção de Relógio Interno (Fosc/4)  ● T1CONbits.T1CKPS= 0b10; //Prescaler a 4  ● T1CONbits.T1OSCEN = 0; //Oscilador secundário desligado  ● T1CONbits.T1SYNC = 0; // Relógio sincronizado com oscilador externo  ● T1CONbits.RD16 = 1; //Lê 16 bits  ● T1CONbits.TMR1ON = 1; //Inicia a contagem  //Configuração da interrupção do Timer 1  ● PIE1bits.TMR1IE = 1; //Autoriza a Interrupção  ● PIR1bits.TMR1IF = 0; //Apaga a flag de Interrupção  ● IPR1bits.TMR1IP = 0; // Interrupção de Baixa prioridade |

|  |
| --- |
| 4. **Timer 2**  **Cálculos:**    O sinal sonoro deverá ser atuado com uma intermitência de 0.5 Hz  ● F=0.5 Hz, com DC=50%  ● TH=TL=500/2 = 250ms  ● X =(PRExPOST)  ● Para encontrar X: Definir PR2 como 255  Como X é um valor muito elevado é necessário reduzir o valor de T e acrescentar uma variável count ao programa.  Se por exemplo, T = 0.05  Ou seja, quando a variável count for igual a 40 ativa o buzzer.  ● Encontrar X com T = 0.05s  Logo, X = => 112  ●Definir por exemplo:  PrescalerxPoscaler = 112 = (16x7)    Prescaler = 16  Poscaler = 7  //O máximo de pr2 e 255  **Registos**:  ●PR2 = 225;  ●TMR2 = 0;  //Postscaler  ●T2CONbits.T2OUTPS3 = 0;  ●T2CONbits.T2OUTPS2 = 1;  ●T2CONbits.T2OUTPS1 = 1;  ●T2CONbits.T2OUTPS0 = 0;  //Prescaler  ●T2CONbits.T2CKPS1 = 1;  ●T2CONbits.T2CKPS0 = 1;  ●PIE1bits.TMR2IE =1; //Autoriza a Interrupção  ●PIR1bits.TMR2IF =0; //Apaga a flag de Interrupção  ●IPR1bits.TMR2IP = 0; //Interrupção de Baixa Prioridade  //Inicia a contagem  ●T2CONbits.TMR2ON = 1; |

|  |
| --- |
| 5.**ADC**  **Converter o valor obtido do ADC em tensão do sensor de pressão**  Para obter a tensão do sensor de pressão utilizamos a seguinte formula:  Sendo que MAX\_VREF = 5 e MAX\_ADC\_VALUE = 1023.  **Converter tensão proveniente do sensor de pressão em um valor correspondente de pressão:**  Depois de ler o datasheet do sensor de pressão MPX4250. Aprendemos que a relação da tensão e do seu valor de pressão tem uma relação linear.  ●Colocamos o sensor de pressão a 0kPa e obtivemos uma tensão de 0.263658V  ●Colocamos o sensor de pressão a 250kPa e obtivemos uma tensão de 4.87084V  ●Logo utilizando a fórmula da reta:  ●Para descobrir o declive da reta utilizamos a seguinte formula:  ●Substituindo os valores obtidos anteriormente:  ●Para calcular o b, utilizamos a seguinte formula:  ●Substituindo os valores obtidos anteriormente: |

|  |
| --- |
| 6.**Eusart**  **Cálculos**:  **Fosc** = 8Mhz, **Baud Rate** = 9600;  **BRG16 = 0 (8 BITS) Velocidade Baixa**    ● Resolvendo para SPBRGH:SPBRG      ● Calcular o Baud Rate    ● Calcular o erro em percentagem  **BRG16 = 0 (8 BITS) Velocidade Alta**    ● Resolvendo para SPBRGH:SPBRG      ● Calcular o Baud Rate    ● Calcular o erro em percentagem  **BRG16 = 1 (16 BITS) Velocidade Baixa**    ● Resolvendo para SPBRGH:SPBRG      ● Calcular o Baud Rate    ● Calcular o erro em percentagem  **BRG16 = 1 (16 BITS) Velocidade Alta**    ● Resolvendo para SPBRGH:SPBRG      ● Calcular o Baud Rate    ● Calcular o erro em percentagem  **Registos**:  //Falta os registos |

|  |
| --- |
| 7.**SPI**  **Registos:**  //Falta os registos |