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 |
|  |  |
|  |  |

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

|  |
| --- |
| **Explicação dos Registos**  1. **Timer 0 Cálculos:**  O sinal luminoso deverá ser ligado com a intermitência de 2 Hz  ● Logo, f =2\*2 = 4 Hz  ● T = 1/f = 1/4 = 0.25s;  FCLK = FOSC /4 = 8\*10^6/4 = 2\*10^6  Para CONTInicial = 0  ● PRE = (TTMR0IF \* FCLK) / (2^16 - CONTInicial)  ● PRE = (0.25 \* 2\*10^6) / (65536) = 7.63  Logo, **Prescaler = 8**  Depois, calculamos o Valor Inicial CONTInicial  ● CONTInicial = 2^16 - [(TTMR0IF\* FCLK) / PRE]  ● CONTInicial = 2^16 - [(0.25\*2\*10^6) / 8] = 3036  **Registos Timer 0:**  ● 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  2. **Timer 1 Cálculos:**  ● T = 1/f = 1/4 = 1\*10^-3s;  FCLK = FOSC /4 = 8\*10^6/4 = 2\*10^6  Para CONTInicial = 0  ● PRE = (TTMR0IF \* FCLK) / (2^16 - CONTInicial)  ● PRE = (0.25 \* 2\*10^6) / (65536) = 7.63  Logo, **Prescaler = 8**  Depois, calculamos o Valor Inicial CONTInicial  ● CONTInicial = 2^16 - [(TTMR0IF\* FCLK) / PRE]  ● CONTInicial = 2^16 - [(0.25\*2\*10^6) / 8] = 3036  **Registos Timer1:**  ● 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 2 Cálculos:**    T = 0.025s  Fosc = 8M  X=(PRExPOST)  ●Para encontrar X: Definir PR2 como 255  X = (TxFosc)/(4x(PR2+1)) = (0.025x4M)/(4x256)=97.7 => 112  ●Definir por exemplo:  PrescalerxPoscaler = 112 = (16x7)  Prescaler = 16  Poscaler = 7  PR2 = (TxFosc/4)/((PRExPOST)-1) = (0.025x4M/4)/((16x7)-1) = 225,2 => 225  Ttotal = 0.025xcount  Ttotal = 1s  Count =1/0.025 = 40  **Registos** **Timer 2**:  ●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; |
| 4.**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 |
| 5.**Eusart**  ●**BRG16 = 0 (8 BITS) Velocidade Baixa**  Fosc = 8Mhz, Baud = 9600, 8-bit  Baud Rate Desejado = Fosc/(64((SPBRGH:SPBRG) + 1))  Resolvendo para SPBRGH:SPBRG  X=((8\*10^6/9600)/64)-1 = 12.02 = 12  Baud Rate Calculado = 11\*10^6/64(0+1) = 171875  Erro = ((Calculo Baud Rate – Baud Rate Desejado)/Baud Rate Desejado)\*100  = (171875-115200)\ 115200)\*100 = 49.2%  ●**BRG16 = 0 (8 BITS) Velocidade Alta**  Baud Rate Desejado = Fosc/(16((SPBRGH:SPBRG) + 1))  Resolvendo para SPBRGH:SPBRG  X=((11\*10^6/115200)/16)-1 = 4.96 = 5  Baud Rate Calculado = 11\*10^6/16(5+1) = 9615.4 = 114583,3 = 114583  Erro = ((Calculo Baud Rate – Baud Rate Desejado)/Baud Rate Desejado)\*100  = (114583 -115200 \ 115200)\*100 = -0.54%  ●**BRG16 = 1 (16 BITS) Velocidade Baixa**  Baud Rate Desejado = Fosc/(16((SPBRGH:SPBRG) + 1))  Resolvendo para SPBRGH:SPBRG  X=((11\*10^6/115200)/16)-1 = 4.96 = 5  Baud Rate Calculado = 11\*10^6/16(5+1) = 114583.3 = 114583  Erro = ((Calculo Baud Rate – Baud Rate Desejado)/Baud Rate Desejado)\*100  = (114583 -115200)\ 115200)\*100 = -0.54%  ●**BRG16 = 1 (16 BITS) Velocidade Alta**  Baud Rate Desejado = Fosc/(4((SPBRGH:SPBRG) + 1))  Resolvendo para SPBRGH:SPBRG  X=((11\*10^6/115200)/4)-1 = 22.87 = 23  Baud Rate Calculado = 11\*10^6/4(23+1) = 114583.3 = 114583  Erro = ((Calculo Baud Rate – Baud Rate Desejado)/Baud Rate Desejado)\*100  = (114583-115200)\ 115200)\*100 = -0.54%  6.**ADC Cálculos:**  0.263658V -> 0kPa  4.87084V -> 250kPa  **y=mx+b**  m=(y2-y1)/(x2-x1) = (250 - 0)/(4.87084 - 0.263658) = 54.263105  b=y2-m\*x1 = 0 - 54.263105 \* 0.263658 = -14.306917  **Comandos utilizados para obter o valor da pressão:**  MAX\_VREF = 5  MAX\_ADC\_VALUE = 1023  adc.voltage = adc.result \* MAX\_VREF / MAX\_ADC\_VALUE;  mpx4250.current\_data = m \* adc.voltage + b; |
| 7.**SPI** |
|  |