

Universidade de São Paulo-USP
Escola de Engenharia de São Carlos-EESC
SEL0629 - Aplicações de Microprocessadores I
Relatório Prática V

Vinicius William da Silva - Número USP: 11233842
Prof. Marcelo Andrade da Costa Vieira

São Carlos, 2023

Em seguida, é apresentado na figura 2 o esquemático da esteira.

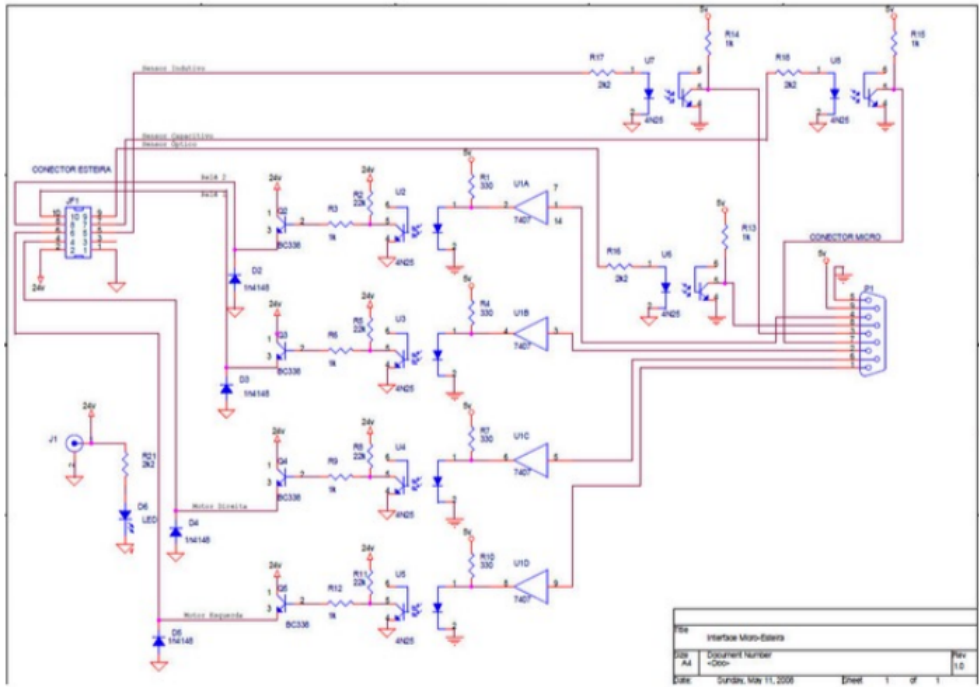


Figura 2: Esquemático da esteira.

3 Cálculos

Para o cálculo do tempo de atuação do cilindro pneumático e também do retorno da esteira ao estado inicial, era necessário, utilizando o *TIMER0*, gerar interrupções em intervalos de 3,45 s, 2,10 s e 0,20 s. O exemplo do cálculo realizado para o para a interrupção de 3,45 s está disposto a seguir

$$T = (2^n - TMR) \cdot CM \cdot PS)$$

Para o valor de tempo de interrupção do TIMER ser de aproximadamente 1 segundo optou-se pelos valores:

$$T = (2^{16} - 11630) \cdot (4/(8 \cdot 10^6)) \cdot 128$$

$$T = 3,449984 \text{ s}$$

Convertendo 11630 para hexadecimal temos o valor 0X2D6E, assim atribuímos ao *timer* TMR0H o valor 0X2D e ao TMR0L o valor 0X6E

Os valores hexadecimais encontrados para as interrupções de 2,10 s e 0,20 s são respectivamente 0X7FD3 e 0XF3CB.

4 Código de Simulação

```
#define TH_3_45 0x2D
#define TL_3_45 0x6E
#define TH_2_10 0x7F
#define TL_2_10 0xD3
#define TH_0_20 0xF3
#define TL_0_20 0xCB

sbit LCD_RS at RB4_bit;
sbit LCD_EN at RB5_bit;
sbit LCD_D4 at RB0_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D7 at RB3_bit;
sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;

// End LCD module connections

int madeira_counter = 0;           // Contador de peas de madeira
```

```
int metal_counter = 0;           // Contador de peas de metal
char current_material[16];       // String do material atual
int current_material_id = 1;     // Id do material (1 mad, 0
    met)
char total_materials[16];        // String de contagem dos
    objetos
int flag_optico = 0;             // Flag do sensor ptico
int flag_capacitivo = 0;        // Flag do sensor capacitivo
int flag_indutivo = 0;          // Flag do sensor indutivo
int flag_passa = 0;             // Flag para liberar objeto
int flag_cilindro = 0;          // Ativao do cilindro
int flag_restart = 0;           // Recomear a esteira

void return_cilinder()
{
    // Recolhe o cilindro
    latb.f5 = 0;
    latb.f4 = 1;
}

void knock_down()
{
    // Ativa o cilindro
    latb.f5 = 1;
    latb.f4 = 0;
}

void turn_left()
{
    // Esquerda
```

```
    latb.f7 = 0;
    latb.f6 = 1;
}

void restart()
{
    // Recomear o processo

    // Retorna motor
    return_cilinder();

    // Esquerda
    turn_left();

    // Zera as flags
    flag_optico = 0;
    flag_capacitivo = 0;
    flag_indutivo = 0;
    flag_passa = 0;
    flag_cilindro = 0;
    flag_restart = 0;

    // Desativa timer
    TOCON = 0;
}

void turn_right()
{
    // Direita
    latb.f7 = 1;
    latb.f6 = 0;
```

```

}

void write_to_lcd()
{
    // Escreve no LCD
    if (current_material_id)
        sprintf(current_material, "PassaMadeira");
    else
        sprintf(current_material, "PassaMetal");

    sprintf(total_materials, "Mad:%dMet:%d",
madeira_counter, metal_counter);

    INTCON.f7 = 0;          // Desliga interrupções
    trisb = 0;              // Output
    Lcd_Cmd(_LCD_CLEAR);    // Limpa Display
    Lcd_Out(1,1,current_material);
    Lcd_Out(2,1,total_materials);
    trisb = 0b00001111;     // Define entrada e saída

    // Reseta bits de interrupção
    INTCON = 0;
    INTCON3 = 0;

    // Religa interrupção
    INTCON = 0b11110000;
    INTCON3 = 0b00011000;
}

// Interrupções
void Interrupcoes() iv 0x0008 ics ICS_AUTO

```

```

{
    if(INTCON.INT0IF)
    {
        // Sensor ptico
        flag_optico = 1;
        INTCN.INT0IF = 0;
    }

    if(INTCON3.INT1IF)
    {
        // Sensor Capacitivo
        flag_capacitivo = 1;
        INTCN3.INT1IF = 0;
    }

    if(INTCON3.INT2IF)
    {
        // Sensor Indutivo
        flag_indutivo = 1;
        INTCN3.INT2IF = 0;
    }

    if(INTCON.TMROIF)
    {
        // Timer 0
        if(flag_cilindro)
        {
            if(flag_passa == 1)
            {
                // Caso deva derrubar objeto
                knock_down();
            }
        }
    }
}

```



```

        TMROH = TH_0_20; // Recarrega o valor no TIMER
0
        TMROL = TL_0_20;
        delay_ms(100);
        return_cilinder();
    }
    else
    {
        // Aguarda objeto cair da esteira
        TMROH = TH_2_10; // Recarrega o valor no TIMER
0
        TMROL = TL_2_10;
    }

    flag_cilindro = 0;
}
else
{
    flag_restart= 1;
}

    INTCON.TMROIF = 0;
}
}

void main()
{
    anse1b = 0;           // Portas digitais
    trisb = 0b00001111;  // Input e Output

    anse1c = 0;           // Portas digitais

```

```
trisc = 255;          // Input

Lcd_Init();           // Inicializa LCD
Lcd_Cmd(_LCD_CURSOR_OFF); // Cursor off
Lcd_Cmd(_LCD_CLEAR);  // Limpa Display

write_to_lcd();

// Liga interrupcoes na portaB
// com descida de borda
INTCON = 0b11110000;
INTCON2.INTEDG0 = 0;
INTCON2.INTEDG1 = 0;
INTCON2.INTEDG2 = 0;
INTCON3 = 0b00011000;

// Reinicia a esteira inicialmente
restart();

// Loop
while(1)
{
    if(portc.rc0 == 0)
    {
        // Escolhe madeira
        current_material_id = 1;
        write_to_lcd();
        delay_ms(300);
    }

    if(portc.rc1 == 0)
```

```

{
    // Escolhe metal
    current_material_id = 0;
    write_to_lcd();
    delay_ms(300);
}

if(flag_optico)
{
    // Vira a esteira
    turn_right();
    flag_optico = 0;
}

if(flag_capacitivo)
{
    // Identificou objeto
    if(current_material_id == flag_indutivo)
        flag_passa = 1; // Derrubar
    else
        flag_passa = 0; // Passar

    if(flag_indutivo == 0)
        madeira_counter++;
    else
        metal_counter++;

    write_to_lcd(); // Atualiza LCD
    TOCON = 0b10000110; // Inicia TIMER0 com prescale

128    TMROH = TH_3_45; // Recarrega o valor no TIMER 0

```

```

        TMROL = TL_3_45;
        flag_cilindro = 1; // Pede possivel ativao do
cilindro
        flag_capacitivo = 0;
    }

    if (flag_restart)
    {
        // Reinicia esteira
        restart();
    }
}
}

```

5 Resultados e Discussão

Após carregar o código do programa no PIC18F45k22, o *display* LCD exibe na primeira linha o material que será aceito pela esteira, e na segunda, a quantidade de materiais para cada tipo que já passaram pelo sensor capacitivo e pelo indutivo. A esteira inicia com o movimento para a esquerda e só muda de sentido quando algum objeto é detectado pelo sensor óptico. Quando um material é detectado pelos sensores capacitivo e indutivo, a depender de qual material está sendo aceito naquele momento, o pistão pneumático é acionado ou não, fazendo com que se rejeite ou aceite o objeto.

O programa foi carregado com sucesso no PIC18F45k22, realizando as tarefas corretamente de acordo como desejado no código, considerando as condições descritas na seção 6. Ao colocar um material e escolher o tipo que será aceito pela esteira, observa-se que o comportamento da mesma é realizado como previsto no código. Caso o material posto seja aceito, este chega ao final e altera-se o sentido de movimento da esteira. Caso o material seja rejeitado, o pistão é acionado no tempo correto para derrubar o objeto

e a esteira move-se para o lado oposto, permitindo que mais materiais sejam detectados.

6 Conclusão

Pode-se concluir que a prática foi um sucesso, em que conseguimos alcançar o objetivo esperado com determinada precisão. Foi possível compreender o uso das interrupções na porta B assim como uma melhor compreensão do funcionamento do TIMER0. Além disso, foi interessante ver a aplicação do microprocessador e do Kit PIC Genios em uma verdadeira aplicação.