


```

    TEMPO1
ENDC

    ORG 0x0000
    GOTO INICIO

    ORG 0x0004
    RETFIE

DELAY_MS
    MOVWF TEMPO1
DELAY_MSB
    MOVLW .250
    MOVWF TEMPO0
DELAY_MSA
    NOP
    DECFSZ TEMPO0,F
    GOTO DELAY_MSA

    DECFSZ TEMPO1,F
    GOTO DELAY_MSB
    RETURN

INICIO
    BANCO1
    MOVLW B'00000001'
    MOVWF TRISA

    MOVLW B'00000000'
    MOVWF TRISD

    MOVLW B'00000000'
    MOVWF TRISE

    MOVLW B'10001111'
    MOVWF ADCON1

    MOVLW B'10000111'
    MOVWF OPTION_REG

    BANCO0
    MOVLW B'00000000'
    MOVWF INTCON

    MOVLW B'01000001'
    MOVWF ADCON0

    CLRF PORTD
    CLRF PORTE
    CLRF TEMPO0
    CLRF TEMPO1

    MOVLW .1
    CALL DELAY_MS

LOOP
    BSF ADCON0,GO_DONE

CONV_PRONTO
    BTFSC ADCON0,GO_DONE
    GOTO CONV_PRONTO
    MOVF ADRESH,W
    MOVWF PORTE
    BANCO1

```

```

    MOVF ADRESL,W
    BANCO0
    MOVWF PORTD
    MOVLW .1
    CALL DELAY_MS
    GOTO LOOP
    END

```

No programa, inicialmente realiza-se a configuração dos registradores que controlam a conversão A/D.

O registrador ADCON0 é configurado da seguinte forma:

- ADCS1 = 0 , ADCS0 = 1 - juntamente com o ADCS2 do ADCON1, configuram a velocidade de conversão para $F_{osc}/8$;
- CHS2 = 0 , CHS1 = 0 e CHS0 = 0 - selecionam o canal 0 (AN0) como o canal de entrada do conversor;
- ADON = 1 - liga o conversor.

O ADCON1 é configurado da seguinte forma:

- ADFM = 1 - ajusta o formato da conversão A/D à direita, ou seja, os oito bits menos significativos da conversão ficam no ADRESL e os dois mais significativos no ADRESH;
- ADCS2 - ajusta a velocidade de conversão;
- PCFG3 = PCFG2 = PCFG1 = PCFG0 = 1 - configuram o pino AN0 como analógico, AN3 e AN2 como entrada de referência da tensão (A tensões de referência são os limites no qual o conversor deve operar) e os demais pinos como I/O digital.

No rótulo LOOP, o bit GO_DONE do ADCON0 é setado, informando que a conversão deve ser iniciada. Posteriormente, este bit é testado até que a conversão termine, ou seja, até que o bit seja limpo. Então, o valor da conversão armazenado nos registradores ADRESH e ADRESL é movido para os PORTE e PORTD, respectivamente.

Para exemplificar, a Figura 2 ilustra uma leitura, em 10 bits, de um valor de tensão aplicado na entrada do conversor A/D do PIC. A partir da Equação 3, pode-se obter o valor correspondente no formato analógico.

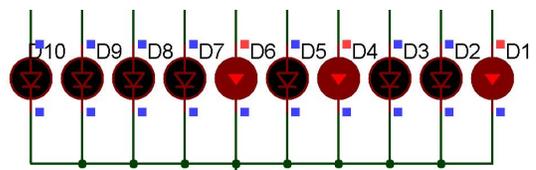


Figura 2. Leitura dos LEDs no Multímetro.

Sabendo que as tensões de referências aplicadas nos pinos AN3 e AN2, V_{ref+} e V_{ref-} são 5 e 0 volts, respectivamente, e o valor no formato digital apresentado nos LEDs, conforme Figura 2, é $V_d = 0000101001 = 41$, então, como mostrado Equação 4, tem-se:

$$V_{ad} = \frac{5 - 0}{1024} \times 41 = 0,2V \quad (4)$$

V. RELATÓRIO - PARTE I

Altere o programa e o esquema elétrico apresentado anteriormente para que o valor da tensão aplicada na entrada do conversor A/D seja apresentado em um display LCD. A apresentação dos valores deve ser semelhante ao ilustrado na Figura 3.

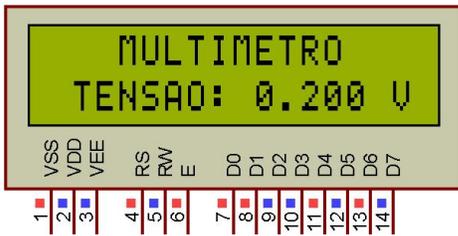


Figura 3. Multímetro com Display LCD.

VI. TERMÔMETRO DIGITAL EM C

Um simples termômetro digital pode ser implementado utilizando o conversor A/D do PIC. Para isso, utiliza-se um circuito com esquema elétrico semelhante ao da Figura 4.

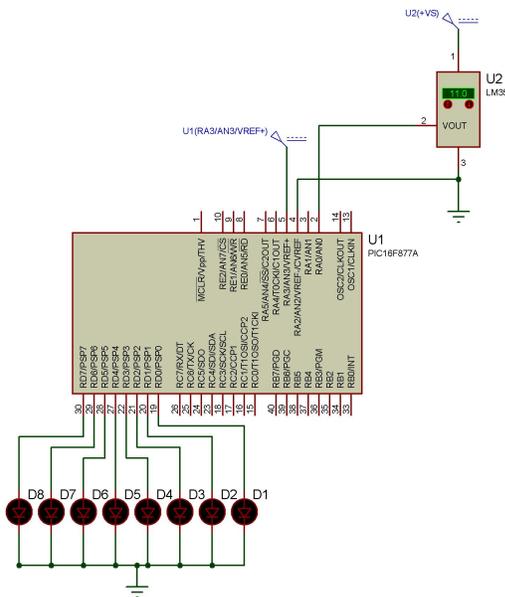


Figura 4. Esquema Elétrico do Termômetro .

Neste circuito, o sensor de temperatura LM35 é conectado ao pino de entrada, AN0, do conversor A/D do PIC16F877A e as tensões de referência são aplicadas aos pinos AN3 e AN2.

O termômetro efetuará a medida de temperatura na faixa de 0 a 150 °C, que corresponde à uma tensão na saída do sensor de 0 a 1.5 V, pois o sensor fornece 10 mV para cada grau Celsius. Devido a este fato, as tensões de referência devem ser $V_{ref+} = 1.5 \text{ V}$ e $V_{ref-} = 0$, aplicadas ao pino AN3 e AN2, respectivamente.

Os valores da conversão no formato digital são apresentados em oito LEDs conectados ao PORTD do microcontrolador.

O programa, escrito em C, responsável por executar tal função é apresentado abaixo:

```
#include <16f877a.h>
#device ADC=8;
#fuses xt, nowdt, noprotect, put, brownout
#use delay(clock=4000000)

void Inicializa(void);

void main() {
    Inicializa();
    while(TRUE)
    {
        int dado;
        dado = read_adc();
        output_d(dado);
        delay_ms(1);
    }
}

void Inicializa(void)
{
    delay_ms(200);
    Setup_ADC(ADC_CLOCK_DIV_8);
    Setup_ADC_Ports(RA0_ANALOG_RA3_RA2_REF);
    Set_ADC_Channel(0);
}
```

O código é iniciado com as instruções reproduzidas abaixo:

```
#include <16f877a.h>
#device ADC=8;
#fuses xt, nowdt, noprotect, put, brownout
#use delay(clock=4000000)

void Inicializa(void);
```

Neste trecho, é realizada a inclusão do ficheiro com as características do PIC16F877A, microcontrolador utilizado no projeto, e, também, é efetuada a configuração do conversor A/D em oito bits de resolução. Posteriormente, os fusíveis ou opções de gravação são ajustados da seguinte forma:

- Oscilador externo tipo cristal (xt);
- Power-up timer (put) ligado;
- Código de Proteção (noprotect) desligado;
- WatchDog Timer (nowdt) desligado.

```
void main() {
    Inicializa();
    while(TRUE)
    {
        int dado;
        dado = read_adc();
        output_d(dado);
        delay_ms(1);
    }
}
```

Na função *main* é executada a função *Inicializa()* responsável por realizar a configuração inicial do PIC. Na função *main*, o comando *while(true)* cria um laço onde é declarada a variável local *dado* do tipo *int* no qual é armazenado o valor da conversão obtido da instrução *read_adc()*. Posteriormente,

o valor da conversão e enviado ao PORTD através da instrução `output_d(dado)`.

Na função `Inicializa()`, é realizada a configuração inicial do PIC através das instruções mostradas abaixo:

- `Setup_ADC(ADC_CLOCK_DIV_8)` - configura a velocidade conversão para $F_{osc}/8$;
- `Setup_ADC_Ports(RA0_ANALOG_RA3_RA2_REF)` - configura o pino AN0(RA0) como analógico e os pinos AN3(RA3) e AN2(RA2) como entrada para as tensões de referência;
- `Set_ADC_Channel(0)` - seleciona o canal 0 (AN0) para a leitura do canal A/D.

```
void Inicializa(void)
{
    delay_ms(200);
    Setup_ADC(ADC_CLOCK_DIV_8);
    Setup_ADC_Ports(RA0_ANALOG_RA3_RA2_REF);
    Set_ADC_Channel(0);
}
```

Para realizar a leitura do valor da temperatura apresentada nos LEDs é utilizada a Equação 2, para oito bits.

A partir da leitura apresentada na Figura 5, pode-se observar que o valor mostrado nos LEDs corresponde ao valor no formato digital $V_d = 00101011 = 43$. Sabendo que a faixa de temperatura a ser convertida é de 0 a 150 °C, então a temperatura pode ser obtida como na Equação 5:

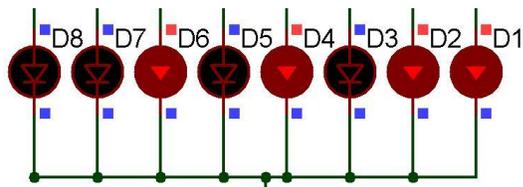


Figura 5. Leitura do LEDs no Termômetro .

$$V_{ad} = \frac{150 - 0}{256} \times 43 = 25.2^{\circ}C \quad (5)$$

VII. RELATÓRIO - PARTE II

Modifique o programa anterior, escrito em C, e o esquema elétrico do circuito no Proteus para que o valor de temperatura, com resolução de 10 bits, seja apresentado em um display LCD e tenha aparência semelhante ao da Figura 6.

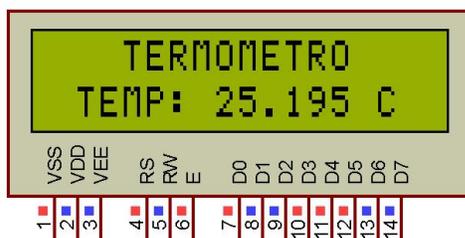


Figura 6. Termômetro com Display LCD.

REFERÊNCIAS

- [1] Souza, Vitor Amadeu, "Projetando com os microcontroladores da família PIC 18: Uma nova percepção", 1ª Ed., São Paulo: Ensino Profissional, 2007.
- [2] Souza, David José de, "Desbravando o PIC: ampliado e atualizado para PIC 16F628A", 6ª Ed., São Paulo: Érica, 2003.
- [3] Pereira, Fábio, "Microcontroladores PIC: Técnicas Avançadas", 6ª Ed., São Paulo: Érica, 2007.
- [4] Apostila de Linguagem C para PIC16F877A com base no CCS - Cerne Tecnologia e Treinamento LTDA.
- [5] Schildt, Herbert, C, Completo e Total, 3ª Ed., São Paulo: Pearson Markron Books, 1997.

APÊNDICE

A - DISPLAY

Hexadecimal	Binário	Descrição
0x38	00111000	Comunicação do módulo com 8 bits
0x06	00000110	Sentido do deslocamento do cursor: para a direita.
0x0C	00001100	Liga display sem cursor.
0x01	00000001	Limpa display com Home cursor.
0x18	00011000	Deslocamento da mensagem sem entrada de caractere: para a esquerda.
0x1C	00011100	Deslocamento da mensagem sem entrada de caractere: para a direita.

Tabela I
COMANDOS PARA INICIALIZAÇÃO DO DISPLAY

LCD 16X2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Linha 1	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
Linha 2	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

Tabela II
ENDEREÇOS DE CARACTERES NO DISPLAY.

Pino	Nome	Função	Descrição
1	Vss	Alimentação	Terra ou GND
2	Vdd	Alimentação	VCC ou +5V
3	Vee	V0	Tensão de ajuste de contraste.
4	RS	Register select	1-Dado, 0-Instrução.
5	R/W	Read/Write	1-Leitura, 0-Escrita.
6	E	Chip select	1 ou (1→0) - Habilita, 0 - Desabilita.
7	D0	Dado	Bit 0
8	D1	Dado	Bit 1
9	D2	Dado	Bit 2
10	D3	Dado	Bit 3
11	D4	Dado	Bit 4
12	D5	Dado	Bit 5
13	D6	Dado	Bit 6
14	D7	Dado	Bit 7

Tabela III
PINAGEM DO MÓDULO LCD

B - SENSOR DE TEMPRATURA - LM35

- Calibrado diretamente em °C;
- Linear + 10,0 mV/°C;
- 0,5°C de precisão;
- Alcance de -55°C a 150°C;
- Opera entre 4 a 30 volts;
- Corrente de dreno menor que 60 uA;
- Não linearidade típica de $\pm 1/4^\circ\text{C}$;
- Baixa impedância de saída 0,1 ohm para carga de 1mA.

C - REGISTRADORES

- R/W - leitura/escrita
- R - somente leitura
- U - não implementado
- O valor após o traço é o valor inicial, após reset.

1 - ADCON0

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	-	ADON
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0

Tabela IV
REGISTRADOR ADCON0

- **ADCS1, ADCS0** - A função destes bits é configurar a velocidade de conversão para o módulo de AD.

ADCS2	ADCS1:ADCS0	Velocidade de conversão
0	00	Fosc/2
0	01	Fosc/8
0	10	Fosc/32
0	11	Frc(Clock derivado do oscilador RC interno)
1	00	Fosc/4
1	01	Fosc/16
1	10	Fosc/64
1	11	Frc(Clock do oscilador RC interno)

Tabela V
ADCS2, ADCS1, ADCS0

- **CHS2, CHS1, CHS0** - Seleção de canal de leitura de AD.
000 - Canal 0 (AN0)
001 - Canal 1 (AN1)
010 - Canal 2 (AN2)
011 - Canal 3 (AN3)
100 - Canal 4 (AN4)
101 - Canal 5 (AN5)
110 - Canal 6 (AN6)
111 - Canal 7 (AN7)
- **GO/DONE** - Status da conversão AD:
1 - Conversão AD está sendo realizada;
0 - Conversão AD não está sendo realizada.
- **ADON** - Desliga ou liga o AD:
1 - Conversor AD está ligado;
0 - Conversor AD está desligado.

2 - ADCON1

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
ADFM	ADCS2	-	-	PCFG3	PCFG2	PCFG1	PCFG0
R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0

Tabela VI
REGISTRADOR ADCON1

- **ADFM** - Ajuste do formato do resultado da conversão de AD:
1 - Justificação à direita;
0 - Justificação à esquerda.
- **ADCS2** - Bit de seleção de velocidade de conversão AD (Tabela V)
- **PCFG3:PCFG0** - Bits de controle da configuração do AD.

PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C / R
0000	A	A	A	A	A	A	A	A	VDD	VSS	8 / 0
0001	A	A	A	A	VREF+	A	A	A	AN3	VSS	7 / 1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5 / 0
0011	D	D	D	A	VREF+	A	A	A	AN3	VSS	4 / 1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3 / 0
0101	D	D	D	D	VREF+	D	A	A	AN3	VSS	2 / 1
011x	D	D	D	D	D	D	D	D	—	—	0 / 0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6 / 2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6 / 0
1010	D	D	A	A	VREF+	A	A	A	AN3	VSS	5 / 1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4 / 2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3 / 2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2 / 2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1 / 0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1 / 2

Figura 1. Bits PCFG3:PCFG0

3 - INTCON

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
R/W-0	R/W-X						

Tabela VII
REGISTRADOR INTCON

- **GIE** - Global Interrupt Enable/Habilitação Global de Interrupções - o bit GIE é o controle geral de interrupções no PIC. Ele funciona como a chave principal de controle de interrupções:
 - 0 - Se estiver desligado, não poderá ocorrer nenhuma interrupção no chip (mesmo que alguma interrupção individual esteja habilitada);
 - 1 - Se estiver ligado (e alguma das interrupções individuais também estiver habilitada), poderá ocorrer interrupção.
- **PEIE** - Habilitação das interrupções periféricas:
 - 0 - Interrupções periféricas desligadas;
 - 1 - Interrupções periféricas habilitadas.
- **TOIE** - Habilitação de interrupções de transbordamento do timer 0;
 - 0 - Interrupção de transbordo do timer 0 desligada;
 - 1 - Interrupção de transbordo do timer 0 habilitada.
- **INTE** - Habilitação de interrupção externa (RB0/INT);
 - 0 - Interrupção externa desligada;
 - 1 - Interrupção externa ligada.
- **RBIE** - Habilitação de interrupção por mudança na porta B (pinos RB4 a RB7);
 - 0 - Interrupção por mudança na porta B desligada;
 - 1 - Interrupção por mudança na porta B habilitada.
- **TOIF** - Flag sinalizador de transbordamento do timer 0;
 - 0 - Não houve transbordamento no timer 0 (a contagem dele não superou 255);
 - 1 - Houve transbordamento no timer 0 (a contagem dele superou 255) e se o GIE estiver ligado, assim como o TOIE, será gerada um interrupção neste caso.
- **INTF** - Flag sinalizador de interrupção externa (RB0/INT);
 - 0 - Não houve interrupção externa no pino RB0/INT;
 - 1 - Uma interrupção externa foi acionada pelo pino RB0/INT (dependendo da borda selecionada pelo bit INTEDG do

registrador OPTION). Se o GIE estiver ligado, assim como O INTE, será gerada interrupção neste caso.

- **RBIF** - Flag sinalizador de alteração nos pinos RB4 a RB7;
0 - Não houve alteração nos níveis lógicos dos pinos RB4 a RB7;
1 - Houve alteração em algum dos pinos RB4 a RB7. Se o GIE estiver ligado e o RBIE também, será gerada uma interrupção.

4 - OPTION_REG

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1

Tabela VIII
REGISTRADOR OPTION_REG

- **RBPU** - Ativação dos resistores pull-up internos da porta B:
0 - Resistores de pull-up ativados (apenas para os pinos configurados como entrada);
1 - Resistores de pull-up desativados.
- **INTEDG** - Seleção da borda de ativação da interrupção externa RB0/INT:
0 - Poderá ocorrer interrupção na borda de descida do sinal no pino RB0/INT;
1 - Poderá ocorrer interrupção na borda de subida do sinal no pino RB0/INT;
- **TOCS** - Seleção de entrada de clock para o timer 0:
0 - O clock do timer será proveniente do clock interno do sistema ($F_{osc}/4$);
1 - O clock do timer 0 será obtido externamente pelo pino RA4/T0CKI.
- **TOSE** - Seleção de borda de sensibilidade do clock externo do timer 0;
0 - O timer 0 será incrementado na borda de subida do sinal externo aplicado em RA4/T0CKI;
1 - O timer 0 será incrementado na borda de descida do sinal externo aplicado em RA4/T0CKI.
- **PSA** Este bit permite selecionar se o prescaler está conectado ao timer 0 ou ao watchdog:
0 - O prescaler está conectado ao timer 0;
1 - O prescaler está conectado ao watchdog.
- **PS0, PS1, PS2** - são os bits responsáveis pela configuração do prescaler utilizado com o timer 0 ou com o watchdog.

PS2	PS1	PS0	Timer0(PSA=0)	Watchdog(PSA=1)
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128