



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
DEPARTAMENTO DE ELETRÔNICA E BIOMÉDICA
CURSO TÉCNICO DE ELETRÔNICA

PROJETO FORNO ELÉTRICO

João Monteiro Delveaux Silva
Joaomdelveaux@gmail.com

Marco Antônio
damascenomarcoantonio2017@gmail.com

Matheus Aurélio Alves
Alves-matheus1@hotmail.com

Alunos do curso de eletrônica no CEFET-MG
Turma: ELT3C

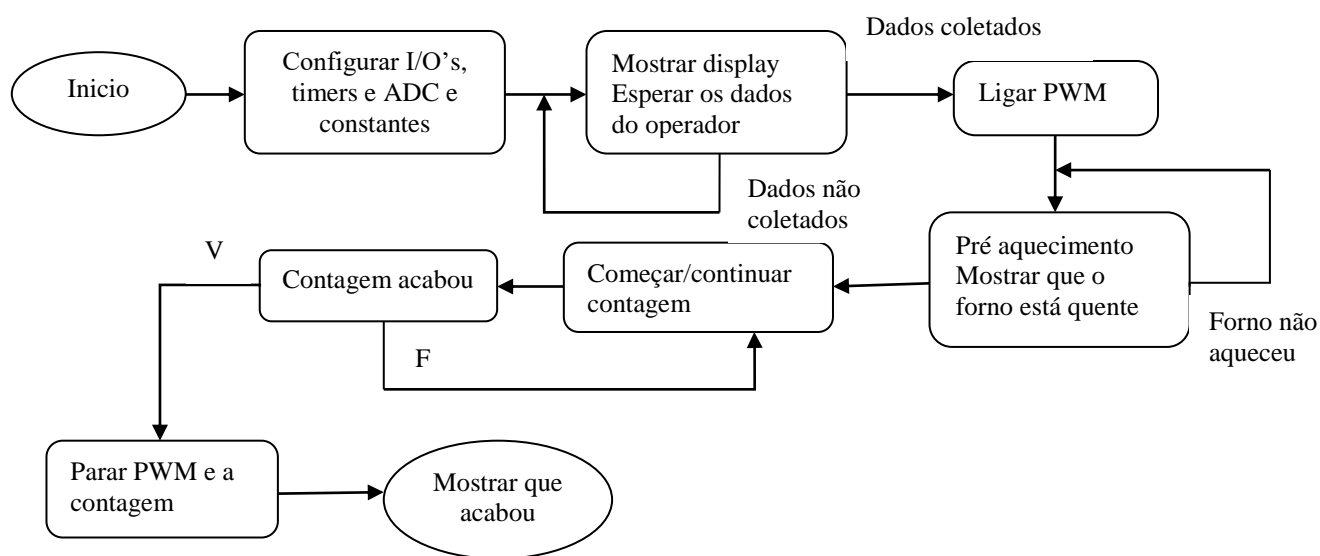


CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
DEPARTAMENTO DE ELETRÔNICA E BIOMÉDICA
CURSO TÉCNICO DE ELETRÔNICA

1) Introdução

O projeto se propõe a ser um controle de um fogão, controlando o tempo de cozimento do alimento e a temperatura desejada pelo usuário, funcionará da seguinte maneira:

O usuário irá introduzir, através de um teclado matricial numérico mecânico, um determinado valor de tempo e uma determinada temperatura, se a temperatura for maior do que a temperatura máxima que o forno consegue fornecer ele irá colocar a temperatura máxima. Em um display de 16*2 caracteres irá ser mostrado o tempo, em minutos, na primeira linha e a temperatura na segunda linha. Sendo que a temperatura será decrementada uma unidade a cada minuto. O programa usado nesse projeto se encontra na seção de Anexos do documento, um escope de como ele funciona será apresentado a seguir.





CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
DEPARTAMENTO DE ELETRÔNICA E BIOMÉDICA
CURSO TÉCNICO DE ELETRÔNICA

Desenvolvimento

A primeira parte do trabalho que foi desenvolvida foi a contagem de segundos e em seguida foi decidido o que seria mostrado no display LCD e como isso aconteceria.

As sub-rotinas "hora" e "mostrardisplay" foram então criadas para atender as necessidades do projeto que eram: oferecer uma contagem regressiva de minutos que poderia se auto-corrigir caso algum valor caísse abaixo do esperado, por exemplo, quando temos que a unidade dos minutos vale 0 e foi decrementada um minuto, nesse caso a unidade irá assumir o valor 9 e a dezena de minutos irá decrementar uma unidade. Além de limitar o valor máximo de temperatura que poderia ser aplicado ao forno e mostrar de forma correta e clara os vetores de tempo restante e temperatura desejada.

As sub-rotinas do teclado, inclusive a de interrupção da porta 2 foram reutilizadas do programa do cronômetro feito no 1º Bimestre desse ano com algumas pequenas mudanças, essas mudanças foram: a tecla anula volta uma casa a ser alterada no vetor e a tecla Enter inicia a contagem regressiva do vetor tempo. Quando o vetor tempo chega a 0 minutos a sub-rotina "Stop" é chamada e ela faz com que o LED indicativo pisque mais rápido além de que um novo vetor aparece no display dizendo que o tempo acabou.

Entrando na parte que engloba a matéria desse bimestre, foram usados 2 blocos de contagem do timer A, eles foram configurados no main e um deles realiza a contagem de 1 segundo e possui um vetor de interrupção para indicar ao programa que um segundo se passou, enquanto o outro bloco serve para realizar o PWM cuja saída é o P1.6, o valor de contagem desse bloco é alterado de acordo com a temperatura real do forno e será explicado mais à frente.

Outra função que foi desenvolvida foi a conversão analógica-digital realizada pelo devido circuito presente no MSP430F1611, ela foi de extrema importância para o projeto pois ela fornecerá o valor da temperatura real do forno no mesmo instante em que for chamada, com isso pode-se comparar o valor da temperatura desejada com a real do forno e decidir se o mesmo deve continuar esquentando ou apenas continuar ligado sem deixar a comida esfriar muito, esse controle é feito pelo ciclo de trabalho do PWM, alterando o valor de TACCR1.

Por fim, para um entendimento mais completo do programa, há uma variável chamada "dígito" que se repete várias vezes durante o programa e serve para indicar qual casa dos vetores será alterada e qual vetor que será alterado também, além de fazer com que a próxima casa a ser alterada pisque como forma de indicação a quem está colocando os dados no MSP saiba em que casa está mexendo.



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
DEPARTAMENTO DE ELETRÔNICA E BIOMÉDICA
CURSO TÉCNICO DE ELETRÔNICA

2) Embasamento Teórico

Esse projeto usou como base para as linhas de programa e maior entendimento do assunto o guia do usuário de uma das famílias de microcontroladores da Texas [1], além de seu datasheet para a confirmação das portas do chip e suas conexões internas [2]. Para a criação do projeto foram usados dois recursos adicionais em relação ao projeto do primeiro bimestre, esses recursos foram: o ADC e o PWM. Abaixo é descrito como esses métodos funcionam.

ADC: O ADC (analog to digital converter) é um recurso utilizado para se converter um valor elétrico de 0 a 5V em um número binário de 10 bits, ou seja, um número que varia de 0 (0000000000) a 1023 (1111111111), com tal recurso é possível converter o valor de tensão do sensor de temperatura em uma variável que pode ser trabalhada pelo programa. O gráfico 3.1 apresenta essa relação entre os valores de tensão de 0 a 5 V com os valores de 0 a 1023 no caso de um conversor A/D de 4 bits.

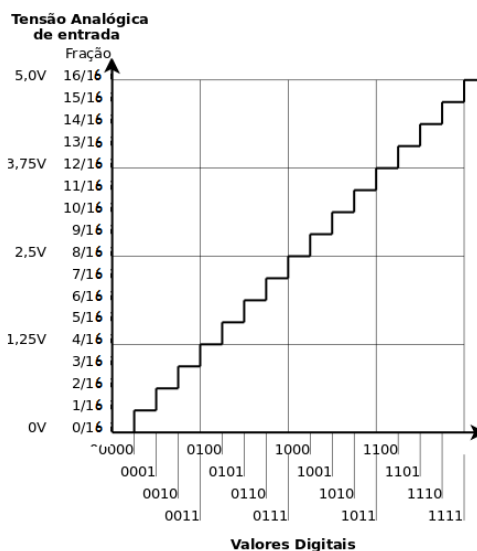


Gráfico 3.1: Conversor A/D de 4 bits

▪ **PWM:** A PWM é um recurso utilizado para se gerar uma saída gradada, isto é, uma saída que pode variar a sua tensão média de acordo com o que foi programado para ela, funciona da seguinte maneira, determina-se um período total e, posteriormente, o período em que a PWM ficará ativa em relação ao período total, com isso se estabelece o seu ciclo de trabalho. A figura 3.1 apresenta essa relação.



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
DEPARTAMENTO DE ELETRÔNICA E BIOMÉDICA
CURSO TÉCNICO DE ELETRÔNICA

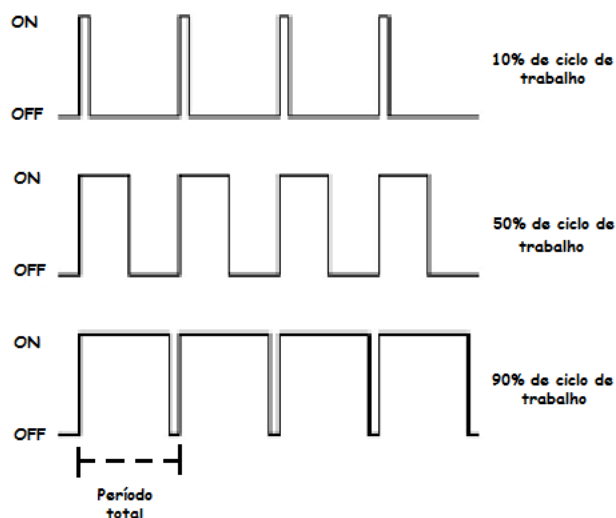


Figura 3.1: Exemplo de PWM

3) Conclusões

O trabalho foi bastante útil, não só por incentivar o trabalho em equipe, mas também pelo contato direto a novas aplicações e recursos do MSP430, como novas funções para os blocos de Timer, e principalmente o conversor A/D que é base para a comunicação da CPU com o mundo externo. Sendo todas as atividades fundamentais para a aprendizagem e fixação desses novos conteúdos e nos servindo como base para a construção de novos projetos, ampliando nossos horizontes, não só para o MSP430, mas também para diversos outros microcontroladores que usam dos mesmos princípios.



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
DEPARTAMENTO DE ELETRÔNICA E BIOMÉDICA
CURSO TÉCNICO DE ELETRÔNICA

4) ANEXO 1: Programa utilizado no projeto

```
#include "msp430f1611.h"
void Config_LCD();
void CLR_LCD();
void envia_dado();
void envia_comando();
void mostrardisplay();
void config_teclado();
void stop();          // É a função que irá sinalizar que o processo terminou
void salvaLetra();
void LerTemp();
unsigned char rd_tec();
unsigned char tecla;
unsigned int Temp;
int digito;           // Irá indicar qual posição do display será alterada
int sensor;
int segundo;
int fornoquente;

char minutos[]={ '1','2','0',' ','M','i','n','u','t','o','s' }; //Vetor que irá armazenar o tempo restante
char temperatura[]={ '2','0','6',' ','G','r','a','u','s' }; //Vetor que irá armazenar a temperatura desejada
char fim []= { 'A','C','A','B','O','U','!','!','!' }; //Vetor que irá aparecer quando acabar o processo

void mostrardisplay(void) // Essa função é responsável por mostrar o tempo restante na linha de cima do display
                          // na linha de cima do display e a temperatura escolhida na linha de baixo do display
{
    P5OUT=0x83;          // Centraliza a primeira linha
    envia_comando();
    for(int i=0;i<11;i++) // Escreve o vetor do tempo na primeira linha
        {P5OUT=minutos[i];envia_dado();}
    P5OUT=0xC3;          // Centraliza a segunda linha
    envia_comando();
    for(int i=0;i<9;i++) // Escreve o vetor da temperatura na segunda linha
        {P5OUT=temperatura[i];envia_dado();}
}

void hora (void)
{
    if (segundo==0)       // Checa se passou 1 minuto
        {segundo=60;minutos[2]--;}
    if(minutos[2]<'0')     // Checa se passou 10 minutos
        {minutos[2]='9';minutos[1]--;}
    if(minutos[0]!='0')   // Checa se é possível diminuir o valor da casa de centenas de minutos
        {
            if(minutos[1]<'0') // Checa se faltam 99 minutos para o fim
                {minutos[1]='9';minutos[0]--;}
        }
    if ((minutos[0]=='0')&&(minutos[1]=='0')&&(minutos[2]=='0')) // Checa se o tempo acabou
        {stop();}
    if((temperatura[0]>='2')&&(temperatura[1]>='5')) // Checa se a temperatura desejada está acima do
    máximo permitido
    {
        temperatura[0]='2';
        temperatura[1]='5';
    }
}
```



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
DEPARTAMENTO DE ELETRÔNICA E BIOMÉDICA
CURSO TÉCNICO DE ELETRÔNICA

```
void stop (void)          // Subrotina de quando o tempo acabar
{
    TACTL=TASSEL_1+MC_0;    // Para a contagem de segundos
    CLR_LCD();              // Prepara o LCD para a escrita de outro vetor
    P5OUT=0x83;
    envia_comando();
    for(int i=0;i<9;i++)
        {P5OUT=fim[i];envia_dado();}
    while(1)                // Esse while piscará o LED mais rapidamente e mostrará no display que o tempo acabou
    {
        P4OUT^=0x01;
        __delay_cycles(15000);
    }
}

void salvaLetra(int tecla)    // subrotina que altera os vetores que estão no display
{
    switch (tecla)            // Separa as ações devido à casos específicos, como
    {
        case 'E':            // Tecla Enter , significa que o operador acabou de digitar os vetores e pode-se
                                // começar a contagem
            TACTL=TASSEL_1+MC_1;    // do tempo do forno
            break;
        case 'A':            // Tecla Anula, volta uma posição no vetor que está sendo alterado
            digito--;
            break;
        case 'n':            // Esse caso significa que a tecla pressionada não foi identificada
            break;
        default:
            if(digito>5)        // checa se a posição dos vetores a serem alterados não foi estourada
                {digito=0;}
            if (digito>2)        // checa se a casa a ser alterada é do vetor temperatura, caso não seja o vetor a ser
                                // alterado será minutos
                {temperatura[(digito-3)]=tecla;}
            else
            {
                minutos[digito]=tecla;
            }
            digito++;            // Passa para a próxima posição dos vetores
            break;
    }
}

void LerTemp()
{
    ADC12CTL0 |= ADC12SC;    // inicia conversão
    while ((ADC12IFG & BIT0)==0);
    {sensor = ((ADC12MEM0/9)+1);}    // Variável sensor se torna um valor em graus da temperatura real do
    // forno

    Temp=(temperatura[0]-0x30)*100+(temperatura[1]-0x30)*10+(temperatura[2]-0x30); // Carrega o novo valor de
    // temperatura

    if(sensor<Temp)            // Analisa a temperatura do forno e a compara com o preset feito pelo operador
        {TBCCR1=270;}        // Se temperatura do forno é menor do que o preset o ciclo de trabalho do PWM
    // é aprox. 90%
    else
}
```



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
DEPARTAMENTO DE ELETRÔNICA E BIOMÉDICA
CURSO TÉCNICO DE ELETRÔNICA

```
{TBCCR1=30;fornoquente=1;} // Se temperatura do forno é maior do que o preset o ciclo de
trabalho do PWM é aprox. 10%
}

#pragma vector=TIMERA0_VECTOR
__interrupt void TimerA (void)
{
    if (fornoquente==1)
        {segundo--;hora();}
    P4OUT^=0x01;
}

#pragma vector=PORT2_VECTOR // Endereço da interrupção do teclado
__interrupt void Porta2 (void)
{
    tecla=rd_tec(); // Carrega o valor da tecla pressionada no caracter "tecla"
    salvaLetra(tecla); // Subrotina que realiza as operações necessárias para cada tecla
    P2IFG=0;
    __delay_cycles(20000); // Delay que previne repiques e o tempo de pressionamento
}

void main( void )
{
    WDTCTL = WDTPW + WDTHOLD; // Stop watchdog timer to prevent time out reset
    Config_LCD();
    config_teclado();

    P4DIR|=BIT1; // Configura a porta 4 para ser a saída do PWM
    P4SEL=BIT1; // BIT1 da porta 4 ligada ao TimerB1 (Saída do PWM)
    P4DIR|=BIT0; // Saída LED será na porta P2.1
    TACCTL0=CCIE; // Habilita a interrupção do bloco TACCR0 ao chegar em 32768
    TACCR0=32768; // Tempo de 1 segundo
    TBCTL1=OUTMOD_7; // Define o modo de operação dos timers
    TBCTL=TASSEL_1+MC_1; // Aciona Timer B

    P6DIR=0x00; // Escolhe o pino de leitura da temperatura
    P6SEL |= 0x01; // habilita o canal A0 para conversão digital

    ADC12CTL0 = ADC12ON+SHT0_2+ENC; // liga o ADC12, e define tempo de amostragem,além de
    habilitar a conversão
    ADC12CTL1 = SHP; // Usar sampling timer
    ADC12MCTL0 = SREF_2; // Vr+ = VeREF+ externo, ou seja, coloca a referência no GND
    TBCCR0=300;
    digito=0;
    _BIS_SR(GIE); // habilita as interrupções
    segundo=60;
    fornoquente=0;
    while(1)
    {
        mostrardisplay();
        LerTemp();
    }
}
}
```




CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
DEPARTAMENTO DE ELETRÔNICA E BIOMÉDICA
CURSO TÉCNICO DE ELETRÔNICA

5) Referências Bibliográficas

- [1] User's Guide for MSP430x2xx Family. Texas Instruments 2013
- [2] Datasheet for MSP430f249 Family. Texas Instruments 2018