

Microcontroladores e Microprocessadores

Prática 01: Oscilador, I/O e Interrupção

Ana Keylla da Fonseca Sousa¹
Marcela Barbosa Batista²
Natália Alves de Araújo³

¹IFCE – Instituto Federal de Educação, Ciência e Tecnologia do Ceará
CEP 60.040-531 Fortaleza (CE)
anakeyllasousa@yahoo.com.br

²IFCE – Instituto Federal de Educação, Ciência e Tecnologia do Ceará
CEP 60.040-531 Fortaleza (CE)
marcela_barbosa2801@hotmail.com

³IFCE – Instituto Federal de Educação, Ciência e Tecnologia do Ceará
CEP 60.040-531 Fortaleza (CE)
natialves635@hotmail.com

Resumo:

De acordo com a disciplina de microcontroladores e microprocessadores do curso de Engenharia de Computação do IFCE, iremos descrever nesse artigo o sistema de oscilação, e suas fontes de clock, portas I/O e Interrupção do PIC18F2550. Essa descrição será necessária para o desenvolvimento da prática 1, usando para a simulação o software PROTEUS-ISIS e para desenvolvimento do código o software MPLAB versão 8.80 e compilador C18 versão 3.4 da Microchip.

Palavra-chave: microcontrolador, PIC18F2550, interrupções, oscilador, I/O, simulações.

1. Introdução

Um **microcontrolador** (também denominado **MCU**) é um computador num chip, contendo um processador, memória e periféricos de I/O. É um microprocessador que pode ser programado para funções específicas, em contraste com outros microprocessadores de propósito geral, eles são embarcados no interior de algum outro dispositivo para que possam controlar as funções ou ações do dispositivo. O microcontrolador integra elementos adicionais em sua estrutura interna, como memórias RAM, ROM e EEPROM, dispositivos periféricos como conversores ADC, conversores DAC em alguns casos; e, interfaces I/O, clocks e osciladores.

No segundo tópico iremos abordar o funcionamento do oscilador no projeto, como as diferentes funções do clock funcionam e interferem no funcionamento do Microcontrolador, inclusive como configurá-los. No terceiro tópico mostraremos o funcionamento, a configuração e como utilizar as portas I/O no projeto. No quarto tópico veremos Interrupção, em específico, as utilizadas no projeto: TIMER2, INT0, e RB e como elas colaboram para o funcionamento do circuito. A demonstração, montagem e compilação será no quinto tópico, junto com os conhecimentos extras que utilizamos para concluir o projeto com sucesso. No sexto

tópico, nossas considerações finais e a conclusão de aprendizado.

2. Oscilador:

Um oscilador, é um circuito que produz um sinal eletrônico repetitivo, frequentemente uma onda senoidal ou uma onda quadrada, sem a necessidade de aplicação de um sinal externo.

O oscilador do Microcontrolador PIC 18F2550 é composto, pela seguinte estrutura:

Quatro modos de

cristal, incluindo alta precisão PLL para USB

- Dois modos externos de clock, até 48 MHz

- Bloco de oscilador interno:

- 8 frequências selecionáveis pelo usuário, a partir de 31 kHz a 8 MHz.

- User-tunable para compensar o desvio de frequência

- Oscilador Secundária usando Timer1 de 32kHz

- Opções do oscilador duplos permitem ao microcontrolador e Módulo USB a funcionar em velocidades de clock diferentes

- Monitor de Clock Fail-Safe:

- Permite o desligamento

seguro se qualquer clock pára.

Para simulação do projeto usaremos o oscilador implementado pelo software PROTEUS – ISIS. Para o caso de uma montagem do circuito em protoboard usaremos o oscilador interno do Microcontrolador.

Oscilador Interno

O oscilador interno do Pic 18F2550 gera dois diferentes sinais de clock, sendo a saída principal (INTOSC) uma fonte de clock de 8 MHz, que pode ser usado para conduzir diretamente o clock do dispositivo. Ele também dirige o postscaler INTOSC que pode proporcionar uma faixa de frequências de clock que variam de 31 kHz a 4 MHz. A saída do INTOSC é ativada quando uma frequência de clock está selecionada entre 125 kHz a 8 MHz.

O bit INTSRC (OSCTUNE <7>=1) é responsável pelo acionamento de uma frequência de 31, 25 KHz. A outra fonte usada pelo oscilador é o INTRC, que é ativada através do bit (OSCTUNE<7> = 0), que seleciona a uma frequência de 31KHz.

A combinação dos bits

IRCF:IRCF0(OSCCON<6:4>), mostrada na tabela abaixo, fornecem as frequências, onde ambas as fontes utilizam – se dessa seleção de bits.

OSCCON<6:4>	Configuração
111	8MHz
110	4MHz
101	2MHz
100	1MHz
011	500KHz
010	250KHz
001	125KHz
000	31KHz

3. Portas I/O

Uma das características mais importantes do microcontrolador é um número de I/O saída pinos utilizados para a ligação com periféricos. Em ordem de operação dos pinos dos pode corresponder organização de 8 bits interno, todos eles são, semelhante aos registradores, agrupadas em três portas (no microcontrolador PIC18F2550) chamados denotados por A, B e C. Todos eles têm várias características em comum:

- Por razões práticas, muitos pinos I/O tem duas ou três funções. Se um pino é usado como qualquer outra função, pode não ser usado como um propósito geral pino de I/O, e
- Cada porta tem o seu "satélite", ou seja, o correspondente registrador TRIS: TRISA, TRISB, TRISC etc, que determina o desempenho, mas não o conteúdo dos bits da porta.

Em compensação alguns bits do registrador de TRIS (bit = 0), o pino de porta correspondente está configurado como saída. Do mesmo modo, definindo alguns bits do registrador de TRIS (bit = 1), o pino de porta correspondente está configurado como entrada.

As portas também possuem mais dois outros registros, que guardam os status dos pinos:

- PORT registrador (lê os níveis sobre os pinos do dispositivo)
- LAT registrador (fecho da saída)

O registrador de dados Trava (LATA) é útil para readmodify-write de gravação sobre o valor impulsionado pela I / O pinos.

4. Interrupções

Interrupção é o sinal enviado para o micro para marcar o evento que requer atenção imediata. Interrupção é "pedir" o processador parar para executar o programa atual e para "fazer tempo" para executar um código especial.

De fato, o método de interrupção define a opção para transferir a informação gerada pelo sistema internos ou externos do microcontrolador. Uma vez que o sistema tenha terminado a tarefa imposta, o processador será notificado para que ele pode acessar e receber informações a serem utilizadas.

O "pedido" para o microcontrolador para "libertar-se" para executar a interrupção pode vir de várias fontes:

- Dispositivos externos de hardware. Exemplo comum é pressionar a tecla no teclado, o que provoca a interrupção de teclado para enviar ao microcontrolador para ler a informação da tecla pressionada.
- O processador pode enviar interrupções para si próprio como um resultado da execução do programa, para comunicar um erro no código. Por exemplo, a divisão por vontade 0 faz com que uma interrupção.
- No sistema multi-processador, os processadores podem enviar para cada outras interrupções, como forma de comunicar.

O microcontrolador PIC18F2550 possui múltiplas fontes de interrupção e uma função prioritária de interrupção que permite que cada fonte de interrupção a ser atribuído nível alta prioridade ou um nível de baixa prioridade. A alta prioridade vetor de interrupção está em 000008h e a baixa prioridade vetor de interrupção está em 000018h. De alta prioridade de interrupção eventos irá interromper todas as interrupções de baixa prioridade que poderá estar em progresso.

Existem dez registros, que são usados para controlar interromper o funcionamento. Esses registros são:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1,PIR2
- PIE1,PIE2
- IPR1,IPR2

Cada fonte de interrupção tem três bits para controlar sua operação. As funções destes bits são:

- bit sinalizador para indicar que um evento de interrupção ocorreu
- Habilitar bit que permite a execução do programa ramificar para o endereço de vetor de interrupção, quando o bit sinalizador está definido

- bit Prioridade para selecionar prioridade alta ou baixa prioridade

O recurso de prioridade de interrupção está habilitada, definindo o Bit IPEN (RCON <7>). Quando prioridade de interrupção é ativada, existem dois bits que permitem interrupções globalmente. Definir o bit GIEH (INTCON <7>) permite que todos os interrupções que têm o conjunto de bits de prioridade (alta prioridade). Definir o bit GIEL (INTCON <6>) permite que todos os interrupções que tem o bit prioridade limpo (prioridade baixa). Quando a bandeira de interrupção, bit de habilitar e apropriado enable interrupção bit globais são definidas, a interrupção será vetor imediatamente para tratar ou 000008h ou 000018h, dependendo da configuração de bits de prioridade. Interrupções individuais pode ser desabilitado através do seu correspondente bits.

Quando o bit IPEN é apagada (estado padrão), o recurso de prioridade de interrupção é desativado. O modo de compatibilidade, os bits de prioridade de interrupção para cada fonte não têm nenhum efeito.

INTCON <6> é o bit PEIE que ativa / desativa todas as fontes de periféricos de interrupção.

INTCON <7> é o bit GIE que ativa / desativa todos interromper fontes. Todos ramo interrupções para tratar 000008h em modo de compatibilidade.

Para eventos externos de interrupção, como os pinos INTx ou PORTB interrupção mudança de entrada, a latência de interrupção vai ser de três a quatro ciclos de instrução. A exata latência é o mesmo para um ou dois-ciclo instruções.

Flags individuais de interrupção são definidas independentemente do status de seu correspondente bit de habilitação ou o bit GIE.

Para a seguinte prática em questão, foram utilizados 3 sistemas de interrupção :

- **Interrupção externa INT0 (high priority);**
- **Interrupção externa RB (low priority)**
- **Interrupção TIMER2 (low priority)**

Essas interrupções foram configuradas de acordo com as tabelas abaixo.

INT0		
Enable bit	INT0IE (INTCON<4>)	1 – Habilita interrupção externa 0-Desabilita interrupção externa
Flag bit	INT0IF (INTCON<1>)	1 – Ocorreu interrupção externa 0- Não ocorreu interrupção externa
Priority bit	===	Não suporta nível de prioridade
Edge	INEDG0 (INTCON2<6>)	1- Interrupção na subida da borda 0- Interrupção na descida da borda

RB		
Enable bit	RBIE (INTCON<3>)	1-Habilita interrupção por mudança

		de estado nos pinos RB4:RB7 0-Desabilita interrupção por mudança de estado dos pinos RB4:RB7
Flag bit	RBIE (INTCON<0>)	1-Pelo menos um dos pinos RB4:RB7 mudou de estados 0-Nenhum dos pinos RB4:RB7 mudou de estado
Priority bit	RBIP (INTCON2<0>)	1-High priority 0-Low priority

TIMER2		
Enable bit	TMR2IE (PIE1<1>)	1-Habilita interrupção por combinação do TMR2 e PR2 0-Desabilita interrupção por combinação do TMR2 e PR2
Flag bit	TMR2IF (PIR1<1>)	1-O valor do registro TMR2 é igual ao valor do PR2 0-O valor do registro do TMR2 é diferente do PR2
Priority bit	TMR2IP (IPR1<1>)	1-High priority 0-Low

5. Prática 1 – Simulação software PROTEUS-ISIS

De acordo com o que foi solicitado na prática 1, compilamos com sucesso código-fonte do projeto, logo após identificamos e carregamos o *hex do projeto no software PROTEUS-ISIS e verificamos o acionamento de todos os componentes corretamente, a fim de testar o funcionamento do firmware e a interface gráfica de simulação.

Teste Oscilador

No PROTEUS-ISIS, modificamos o campo *Processor Clock Frequency* para os valores:

- 4 MHz;
- 8 MHz;
- 16 MHz;
- 48 MHz;

Para observarmos o tempo com que o LED1 pisca, anotando os resultados.

Simulação PROTEUS-ISIS	
Frequência	Tempo observado para LED1 piscar
4MHz	2s
8MHz	1s
16MHz	0,5s
48MHz	0,168s

Eventos em cadeia

Inicialmente modificamos o circuito para um similar a figura 1 mostrada na descrição da prática, em seguida salvamos o arquivo com o nome de “PISCA LED- parte 2”, como pedido.

A partir do TIMER2 foi criado um novo evento (CONT_BIN_handle), que recebia o seguinte evento handle:

```
/**
 *@fn PORTC_LEDS_EVENT_handler()
 *@brief PORTC LEDES Event Handler
```

```
*@param \a void
*@return \a void
*/
void PORTC_LEDS_EVENT_handler(void)
{
    //PORTC++;
}
```

Este evento em cadeia gera um contador binário, onde os leds (D5, D4, D3 e D2 – RC0:RC4 respectivamente) formam os números de 4 bits que vão ser representado por led ligado equivalendo a 1 e led desligado valendo 0.

A cada inversão de estado do led D1 o

contador binário representados pelos leds: é incrementado de uma unidade. Os pinos RC4:RC0 são inicializados com 0 , ou seja todos os leds estão desligados. Quando o PORTC++ é executado o valor nos pino RC4:RC0 passa a ser 0001. Quando o valor enviado aos pinos RC4:RC0 forem iguais a 1000, o led D2 deveria acender, porém isso não acontecerá, devido o RC4 e RC5 somente funcionarem como entrada de dados digitais, como o led D2 só irá acender quando houver saída de dado do pino RC4, logo o led D2 permanecerá apagado .

Interrupção de alta prioridade 1

Inicialmente responde-se a pergunta:

A interrupção INT0 pode ser de baixa prioridade? Por quê?

A interrupção INT0 sempre será de alta prioridade, pois de acordo com o fabricante do Microcontrolador 18F2550, Microchip, a interrupção externa INT0 não suporta nível de prioridade.

Em um segundo momento da prática, foi gerado um evento, a partir na interrupção INT0, similar ao código abaixo:

```
/**
 *@fn INT0_EVENT_handler(void)
 *@brief Teste da interrupção externa
 0
 *@param \a void
 *@return \a void
 */
void INT0_EVENT_handler(void)
```

```
{
    LED2 ^= 1;
}
```

Nesse evento, observa-se a cada vez que é gerada a interrupção, mediante ao pressionar o botão 2 (RB7), o estado do led D6 (RA1) terá o seu valor invertido.

Interrupção de alta prioridade 2

Primeiramente responderemos a pergunta: Existe diferença entre utilizar o pino RB7 ou RB4 para executar essa interrupção? E quanto ao Pino RB3?

A interrupção RB sempre acontecerá para a faixa de pinos RB7:RB4 sempre que houver mudança de estado em um dos pinos, logo não é importante se o botão estiver ligado no pino RB4 ou RB7, a interrupção RB irá acontecer. Quanto ao pino RB3 não iria acontecer nenhum tipo de interrupção, pois a interrupção associada a esse pino e a faixa RB0:RB2 é do tipo externa INT0, INT1 ou INT2 e não do tipo RB.

Na segunda parte da prática geramos um evento similar ao descrito abaixo:

```
/**
 * @fn RB_EVENT_handler(unsigned char
 X)
 * @brief Teste da interrupção por
 mudança de borda da porta B
 * @param \a Estado da porta B
 * @return \a void
 */
void RB_EVENT_handler(unsigned char
 X)
{
    X &= 0x80;
    X >>= 7;
    LED2 = !X;
}
```

A função acima recebe um valor 8 bits contido no PORTB no momento da interrupção RB.

Na primeira instrução é feita uma comparação lógica AND bit a bit entre X e 0x80, o resultado é armazenado na própria variável X. Posteriormente apenas um bit do valor de X será selecionado.

Na segunda instrução o número binário

armazenado em X tem seus bits deslocados 7 vezes para a direita.

Na terceira instrução é atribuído ao D6 (ativado pelo pino RB7) a negação de X.

A interrupção RB gerará duas interrupções, uma ao apertarmos o botão e outra assim que deixarmos de pressionar o botão, assim, quando apertamos o botão 1 e mantemos ele pressionado o led D6 permanecerá aceso, no momento em que soltarmos o botão uma nova interrupção ocorrerá e o led apagará.

6. Considerações Finais

Nesse projeto aprendemos a definição e funcionamento do oscilador, TIMER2, sistemas de interrupção de alta e baixa prioridade e ainda uma base sólida de lógica de programação de microcontroladores, devido a implementação da prática em questão.

7. Referências

- Data sheet Microchip - PIC18F2455/2550/4455/4550
- http://www.microcontrollerboard.com/pic_interrupt.html
- http://en.wikipedia.org/wiki/Electronic_oscillator