

Projeto de Sistemas Microcontrolados

Aula 10 - Roteiro Prático nº 3

Apresentação

Nesta aula, você estudará dois blocos muito importantes no controle de sinais analógicos e na geração de sinais modulados por largura de pulsos do microcontrolador 18F45k20, que são, respectivamente, o módulo Conversor A/D e o módulo CCP1. A linguagem usada na programação envolverá o uso de funções e diretivas do compilador C18 (ou opcionalmente do CCS), especialmente definidas para esses módulos e para o tratamento de conceitos anteriormente já assimilados envolvendo o tratamento de interrupções de programa e as características de portas de E/S do PICKit 3.

Objetivos

Ao final das atividades previstas para esta aula, você será capaz de:

- Descrever o funcionamento do módulo Conversor A/D, dos módulos CCP (Capture, Compare e PWM) e dos seus relacionamentos com o sistema de interrupção de programa do microcontrolador 18F45k20.
- Mostrar como o Timer 2 é usado no controle das funções pertinentes ao módulo CCP2.
- Escrever, depurar, gravar e testar programa em C que permita o tratamento do Conversor A/D e do módulo CCP1 do 18F45k20.

Desenvolvendo atividades e aplicativos em C para o PICKit 3 – nível avançado

Estamos concluindo nosso curso. Nesta última aula de atividades práticas, você terá a oportunidade de escrever programas em C que possibilitam testar dois dos principais módulos funcionais dos microcontroladores PIC: o módulo Conversor A/D e o módulo CCP (ou o ECCP). Com essas atividades você se sentirá apto a programar qualquer módulo funcional presente em qualquer PIC e de qualquer família.

Mas uma vez terá que manipular o tratamento adequado de interrupções de programa e temporizações precisas associadas a aplicações que interagem com o ambiente, recebendo informações dele e atuando nele.

Ao concluir esta aula, a aplicação desenvolvida pode ser entendida como sendo um sistema que monitora, dentre muitas outras, por exemplo:

- 1. a temperatura de um ambiente ou equipamento qualquer e que atua em um sistema de refrigeração, aumentando ou diminuindo sua velocidade de giro;**
- 2. a redução ou o aumento de rotação de um motor e que atua no seu sistema de controle, mantendo-o em uma velocidade constante.**

Enfim, que os conhecimentos adquiridos neste curso possam lhe ser úteis no desenvolvimento de sistemas e aplicações que possam ajudar no seu futuro e no futuro da nossa gente. Boa sorte!

Atividade 01

A partir do estudo sobre o funcionamento do Conversor A/D, do módulo CCP1 e do Timer 2 do 18F45k20, responda às questões a seguir.

- 1. Quais as principais funções do Timer 2 e como é sua estrutura funcional? Quais os registros e respectivos bits envolvidos em sua programação?**

- 2. Como é feito o cálculo da frequência de interrupção do Timer 2?**
- 3. Explique o funcionamento do CCP1 no modo PWM, evidenciando os registros e bits envolvidos em sua programação.**
- 4. Como pode esse módulo CCP1 ser usado para medir frequência de um sinal de entrada?**
- 5. É possível usar o módulo ECCP redirecionando a saída PWM para o pino 7 do PORTD? Se sim, como isso pode ser feito?**
- 6. Explique a operação do Conversor A/D do PIC 18F45K20 evidenciando os registro e bits envolvidos em sua programação, bem como os tempos envolvidos numa operação de conversão analógica para digital.**
- 7. Como usando apenas um conversor A/D, múltiplas entradas analógicas podem ser amostradas? Represente isso através de um diagrama de blocos simplificado.**

Conversão Analógico-Digital: PIC18F45K20

O microcontrolador PIC18F45K20 contém apenas um conversor A/D, porém existe um multiplexador (seletor) capaz de comutar a chave para ler valores de 13 diferentes canais. Ou seja, é possível realizar uma leitura analógica a partir dos pinos RA0 (canal AN0), RA1 (canal AN1), RA2 (canal AN2), RA3 (canal AN3), RA5 (canal AN4), RE0 (canal AN5), RE1 (canal AN6), RE2 (canal AN7), RB2 (canal AN8), RB3 (canal AN9), RB1 (canal AN10), RB4 (canal AN11) e RB0 (canal AN12).

Conforme realizado em outros roteiros, é necessário configurar os bits de alguns registradores para converter valores analógicos em digitais.

Passos para configurar o ADC:

- 1. Configurar o valor do oscilador interno.o OSCCONbits.IRCF = 0b110.**
 - Registrador OSCON: bits<6:4> IRCF.
- 2. Configurar a fonte de clock e o tempo de aquisição da conversão.**
 - Registrador ADCON2:

REGISTER 19-3: ADCON2: A/D CONTROL REGISTER 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0

- Bits<2:0> ADCS: Seleção do clock de conversão.
- Bits<5:3> ACQT: Configuração do tempo de aquisição.
- Bit 7 - ADFM: Formato do valor digital armazenado, após conversão, nos registradores ADRESH e ADRESL.

3. Configurar os valores de tensões de referência para a conversão.

- Registrador ADCON0:
 - Bits<5:4> - VCFG: Tensões de referência: Vss e Vdd.

4. Configurar o pino como analógico.

- Registradores ANSEL e ANSELH.
 - Bit 0 ANSEL: Configurar o pino RA0 (canal AN0) como analógico.
 - Bit 1 ANSEL: Configurar o pino RA1 (canal AN1) como analógico.
 - ...
 - Bit 4 ANSELH: Configurar o pino RB0 (canal AN12) como analógico.

5. Configurar o canal de conversão, habilitar o conversor e iniciar a conversão.

- Registrador ADCON0:

REGISTER 19-1: ADCON0: A/D CONTROL REGISTER 0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
bit 7							bit 0

- Bits<2:0>Bits<5:2> - CHS: Seleção do canal analógico (AN0:AN12).

- **0000 = AN0**
 - **0001 = AN1**
 - **0010 = AN2**
 - ...
 - **1100 = AN12**
- **Bit 1 - GO/DONE: Inicia a conversão.**
 - **Bit 0 - ADON: Habilita o conversor A/D do microcontrolador.**

Além desses passos, é importante lembrar que o registrador TRISx do pino a ser utilizado para receber o sinal analógico deve ser configurado como entrada.

O quadro abaixo apresenta um código exemplo que pode ser utilizado como base para converter um sinal analógico a partir da porta RA0.

```

1 void main() //Função principal
2 {
3     OSCCONbits.IRCF = 0b110; // Configurar oscilador interno para 4MHz
4     ADCON2bits.ADCS = 0b110; // Clock: Fosc/64
5     ADCON2bits.ACQT = 0b101; // Tempo de aquisição: 12*TAD
6     ADCON2bits.ADFM = 1; // Formato do valor digital justificado à direita.
7     ADCON1 = 0b00000000; // ADC Vref = Vdd,Vss
8     ANSEL |= 0b00000001; // Configura RA0 como analógica
9     TRISA |= 0b00000001; // Configura RA0 como entrada
10    ADCON0bits.CHS = 0b0000; // Configura canal AN0 (RA0)
11    ADCON0bits.ADON = 1; // Habilitar conversor AD
12 }
```

Os valores resultantes do circuito de conversão A/D são armazenados nos registradores **ADRESH** e **ADRESL**.

O trecho de código abaixo apresenta as instruções para iniciar a conversão, esperar o tempo de conversão e ler o valor digital.

```

1 void main() //Função principal
2 {
3     ADCON0bits.GO = 1; // Iniciar conversão
4     while (ADCON0bits.GO){}; // Esperar a conversão finalizar
5     int valor = (ADRESH << 8) + ADRESL;
6 }

```

O microcontrolador PIC18F45K20 contém um conversor de 10 bits (bit0 a bit9). A configuração do bit ADFM do registrador ADCON2 como formato do dado justificado à direita (ADFM = 1) armazena os bits resultantes da conversão conforme mostra a tabela a seguir.

ADRESH								ADRES							
.7	.6	.5	.4	.3	.2	.1	.0	.7	.6	.5	.4	.3	.2	.1	.0
X	X	X	X	X	X	bit9	bit8	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

Com isso, é possível utilizar a expressão “(ADRESH << 8) + ADRESL” para representar o valor resultante do conversor em um valor digital inteiro de 0 a 1023.

Atividade 02

1. Escreva um programa usando funções e diretivas do compilador C18 (ou opcionalmente do CCS) para:

- Fazer amostras da tensão presente no potenciômetro RP1 e mostrar, através dos LED1 (menos significativo) até LED6 (mais significativo), a tensão medida no potenciômetro, em binário, com uma aproximação de uma casa decimal. Por exemplo, ao medir uma tensão de 2,34 V, mostrar nos LED o código binário 010 0011 (que será lido como 2,3 V) e, ao medir uma tensão de 3,87 V, mostrar nos LED o código binário 011 1001 (que será lido como 3,9 V).
- De acordo com o valor do potenciômetro, acender o LED7 de acordo com um ciclo de trabalho que varia de um período mínimo a um período máximo de

acendimento e que, ao lê a tensão como sendo de 2,5 V, o tempo de acendimento corresponda a um ciclo de trabalho de 50% (no qual o tempo aceso é o mesmo do tempo apagado). Antes de executar esta atividade, reveja sua resposta à pergunta feita na Atividade 1, item e.

Atividade 03

- 1. Compile, usando o C18 (ou opcionalmente o CCS), o programa escrito na Atividade 2, armazene o executável no PIC do esquemático desenhado no Proteus e simule sua execução.**

Atividade 04

- 1. Grave o executável do programa gerado na Atividade 3 no PIC do *Demo Board* do PICKit 3 e teste o seu funcionamento.**

Resumo

Nesta aula, você estudou os módulos Conversor A/D e CCP1 do 18F45k20, programando-os em linguagem C e testando suas funcionalidades por simulação no Proteus e em execução de tempo real no PICKit 3.

Autoavaliação

1. Descreva o funcionamento do Timer 2 do 18F45k20?
2. Quais os registradores associados ao Conversor A/D do PIC 18F45k20 para tratamento de interrupção num processo de conversão de um sinal analógico para digital?
3. Que módulos podem ser usados para medição de frequência no PIC 18F45k20?
4. Relacione as diretivas utilizadas durante esta aula prática.
5. O que é PWM? Que módulos do PIC 18F45k20 podem ser usados para gerar sinais por PWM?

Referências

PEREIRA, Fábio. **Microcontroladores PIC 18 Detalhado**: Hardware e Software. São Paulo: Érica, 2010.

_____. **Microcontroladores PIC**: Programação em C. Fábio Pereira. São Paulo: Érica, 2005.

SOUZA, David J.; LAVINIA, Nicolas C. **Conectando o PIC**: Explorando recursos avançados. São Paulo: Érica, 2003.