



Aula 10

Microcontroladores: Programação em C

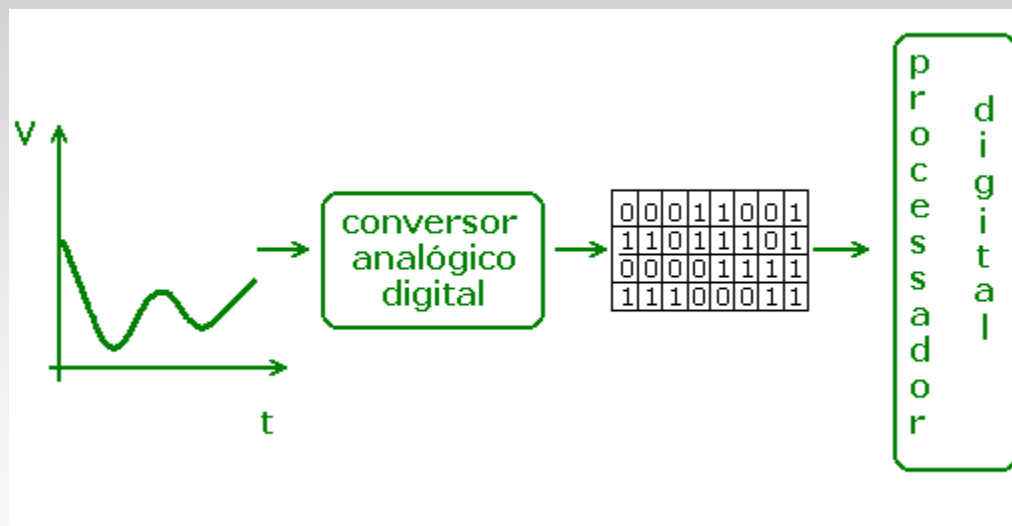
Prof. Tecg^o Flávio Murilo

Eletroeletrônica – Microcontroladores– Módulo IV



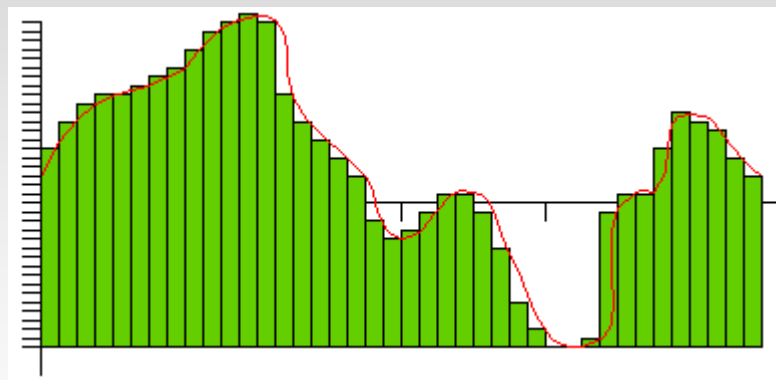


- Conversores analógico digitais (conversores A/D ou ADC) convertem um sinal analógico em um equivalente digital, comumente com resoluções de 8 bits variando valores entre 0 (00000000) e 255 (11111111), 10 bits variando valores entre 0 (0000000000) e 1023 (1111111111), etc.





- Essa conversão é feita por base na leitura de tensão e de corrente numa porta do microcontrolador e dependendo da resolução utilizada, cada nível de tensão tem um valor binário equivalente. Por exemplo: Se for feita a leitura de um sinal analógico que varia de 0 a 5V a uma resolução de 8 bits (variação de valores de 0 a 255) então teremos que a cada $(5/255)V$ teremos um incremento no valor digital.





- Sendo assim, $5/255$ é igual a aproximadamente 0.0196 . Então a cada $0.0196V$ teremos um incremento no valor digital. Esse processo é chamado de discretização, ou digitalização.

Passo da resolução	Tensão lida	Equivalente binário
0	$0 * 0.0196 = 0V$	00000000
1	$1 * 0.0196 = 0.0196V$	00000001
2	$2 * 0.0196 = 0.0392V$	00000010
...
150	$150 * 0.0196 = 2,94V$	10010110
...
255	$255 * 0.0196 \approx 5V$	11111111





- Após ativar (setar) o canal ADC, usando o comando `set_adc_channel(canal);` (sendo o valor do canal de 0 a 7, em um PIC com 8 canais), estes valores resultantes da variação pode ser tratados como variáveis através do comando de leitura `v=read_adc();` (a variável v recebe o valor da conversão AD).
- Se o valor da variável for usado para setar o ciclo ativo (duty cycle) em um controle PWM, há que se fazer a relação entre a resolução do canal ADC e a carga (período) do Timer2.
- **Exemplo:** Se a resolução do ADC for de 8 bits (0 a 255) e a carga do timer2 variar de 0 a 149, então temos:

Valor ADC	Valor PWM	Tensão equivalente
0	0	0V
255	149	5V





- Logo, podemos estabelecer a seguinte relação:

$$\Rightarrow \frac{Res(ADC)}{Res(PWM)} = \frac{5V/255}{5V/149} = \frac{0,01960784313725V}{0,03355704697986V} = 0,58431372549015$$

- Em outras palavras, podemos dizer que se a cada “passo” na resolução do ADC (1 em uma variação de 0 a 255) temos 0,019607...V e a cada “passo” na resolução do PWM (1 em uma variação de 0 a 149) temos 0,033557...V, então a variável precisa ser tratada por um “fator de conversão” de 0,584313...
- Para facilitar trabalhar com PWM e ADC em conjunto, basta trabalhar com a resolução de ambos iguais, pois se $Res(ADC) = Res(PWM)$, então $Res(ADC)/Res(PWM) = 1$. Dessa forma, é só usar o valor lido para uma variável diretamente.





- Logo, podemos estabelecer a seguinte relação:

$$\Rightarrow \frac{Res(ADC)}{Res(PWM)} = \frac{5V/255}{5V/149} = \frac{0,01960784313725V}{0,03355704697986V} = 0,58431372549015$$

- Em outras palavras, podemos dizer que se a cada “passo” na resolução do ADC (1 em uma variação de 0 a 255) temos 0,019607...V e a cada “passo” na resolução do PWM (1 em uma variação de 0 a 149) temos 0,033557...V, então a variável precisa ser tratada por um “fator de conversão” de 0,584313...
- Para facilitar trabalhar com PWM e ADC em conjunto, basta trabalhar com a resolução de ambos iguais, pois se $Res(ADC) = Res(PWM)$, então $Res(ADC)/Res(PWM) = 1$. Dessa forma, é só usar o valor lido para uma variável diretamente.



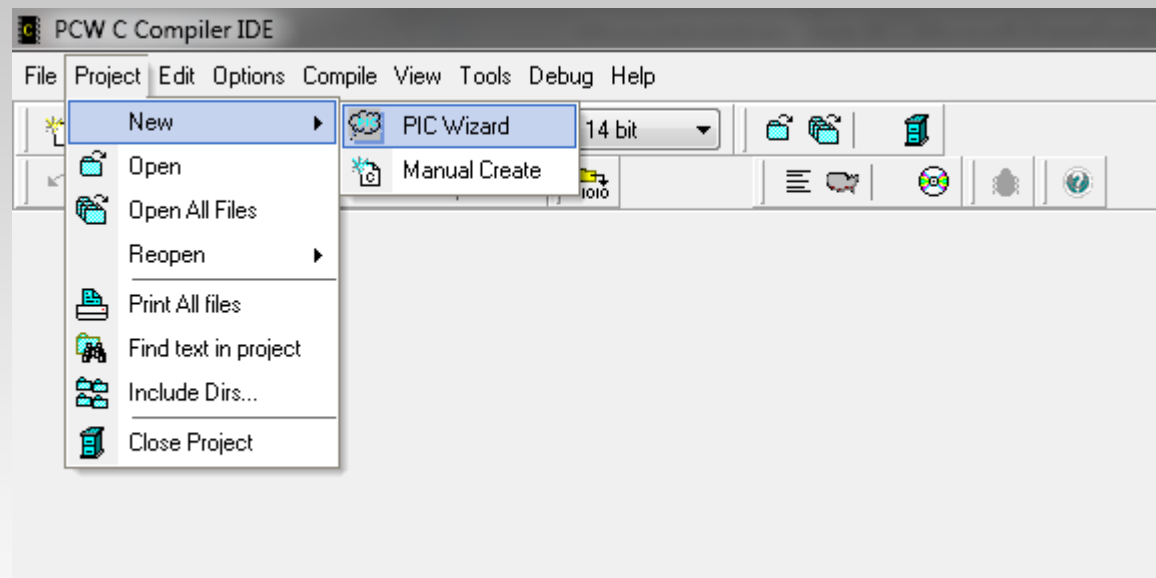


- Algumas características relativas a Conversão Analógica Digital devem ser levadas em consideração, são elas:
 - Canais se conversão são portas (pinos) de entrada para conversão AD.
 - Os módulos fazem a conversão do valor lido do canal para um equivalente binário.
 - Alguns microcontroladores possuem 8 canais para conversão (16F877A), outros 10 canais (18F2550) e alguns nenhum canal (16F628A).
 - Embora alguns microcontroladores tenham vários canais, todos eles possuem apenas um módulo de conversão.
 - Não se pode utilizar mais de um canal simultaneamente num mesmo microcontrolador, dessa forma, é necessário desabilitar um para poder habilitar outro.



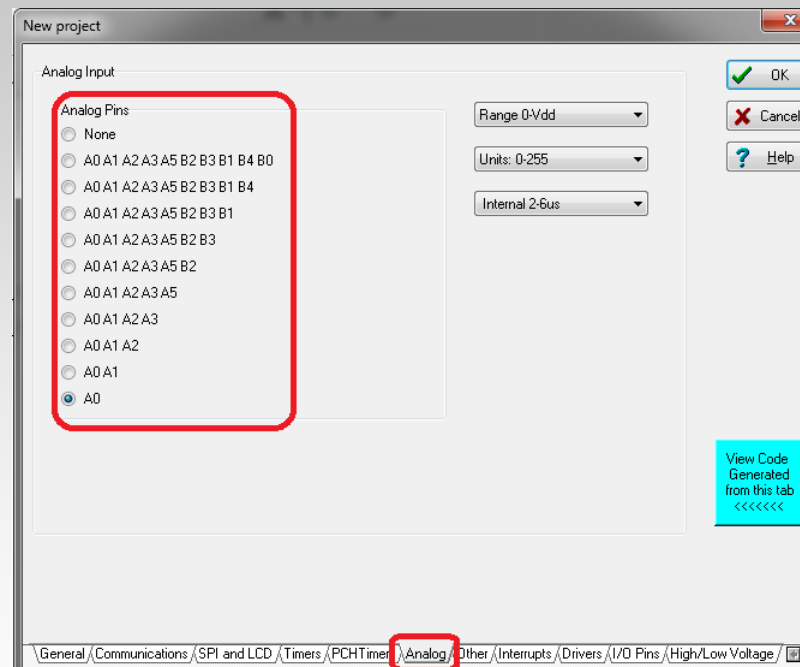


- Crie um novo projeto pelo PIC Wizard.



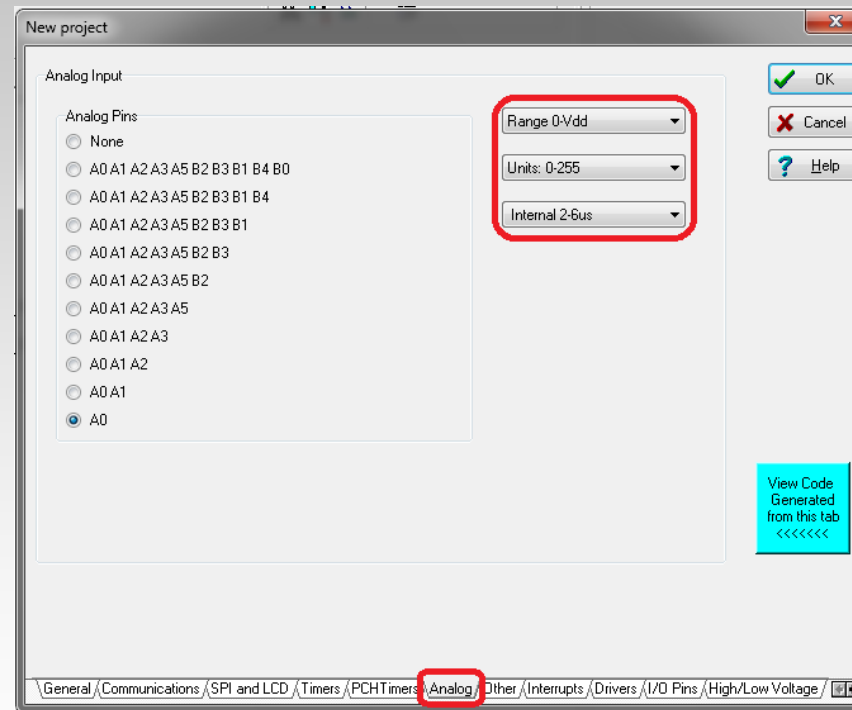


- Em “Device” selecione o PIC utilizado. Desative o “Master Clear”, caso seja necessário e configure o clock. Depois na aba “Analog”, selecione quais canais ADC deseja utilizar.





- Ainda na aba “Analog”, selecione o Range (Variação), a resolução (0 a 155 – resolução de 8 bits, 0 a 1023 – resolução de 10 bits, etc) e o por fim o clock. Feito isso, clique em “OK”.





- As duas primeiras linhas do código gerado são referentes à configuração feita no compilador.

```
void main()
{
    setup_adc_ports(AN0_TO_AN1|USS_UDD);
    setup_adc(ADC_CLOCK_INTERNAL);
    setup_spi(FALSE);
    setup_wdt(WDT_OFF);
    setup_timer_0(RTCC_INTERNAL);
    setup_timer_1(T1_INTERNAL|T1_DIV_BY_1);
    setup_timer_2(T2_DIV_BY_16,149,1);
    setup_timer_3(T3_DISABLED|T3_DIV_BY_1);
    setup_ccp1(CCP_PWM);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);
    setup_low_volt_detect(FALSE);
    setup_oscillator(False);
}
```



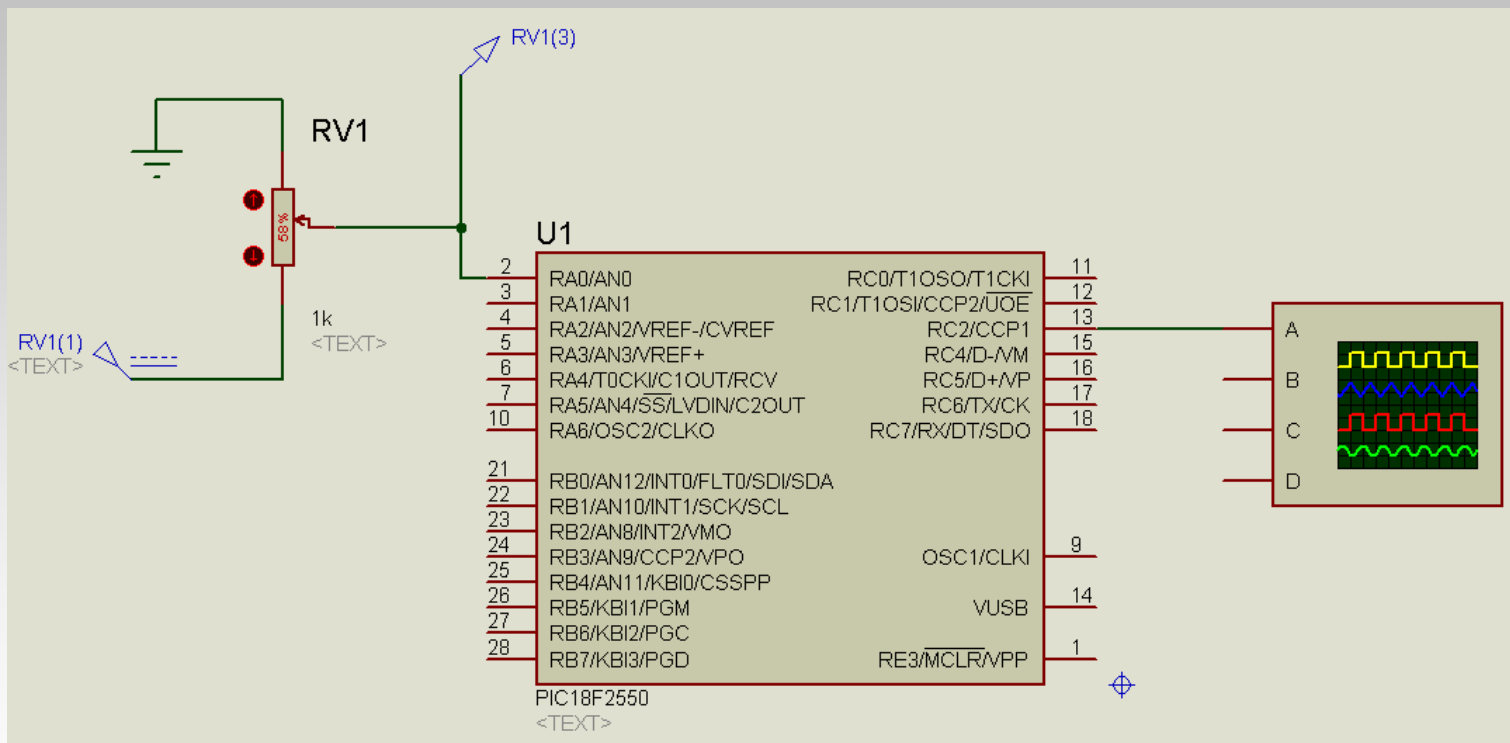


- No Proteus iremos fazer um exemplo usando os seguintes componentes:
 - Microcontrolador PIC 18F550;
 - Potenciômetro (POT-HG);
 - Aterramento (GROUND da aba Terminals Mode);
 - Osciloscópio (OSCILLOSCOPE da aba Virtual Instruments Mode);
 - Fonte CC (DC da aba Generator Mode);
 - Leitor de tensão (Voltage Probe).



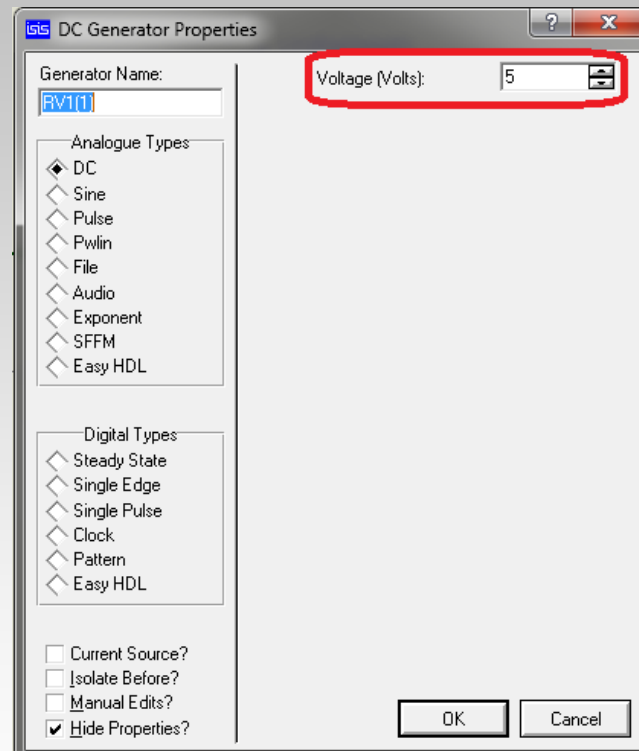


- Monte o circuito como na figura:



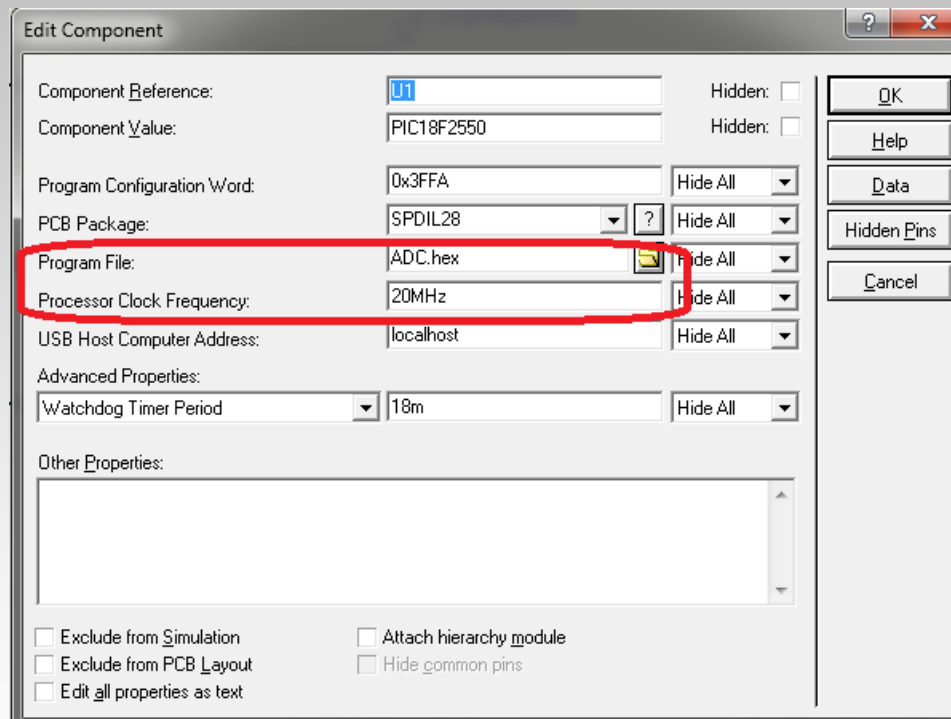


- Configure o DC Generator para fornecer uma tensão de 5V.





- No microcontrolador configure a frequência do clock do processador como 20MHz, escolha o arquivo do programa e simule.





- **Exemplo 01:** Com o circuito montado, faça a modulação da largura de pulso através do controle do potenciômetro. Depois adicione um motor no circuito e observe a variação da velocidade de acordo com o ajuste da resistência.





- Fazer o comentário das linhas de todos os códigos referentes aos exemplos mostrados em sala de aula. Os arquivos que devem ser comentados são os de extensão .c e estão disponíveis para download no site.
 - Para comentar, basta abrir o arquivo .c no compilador e digitar a descrição da linha usando // no início do comentário e no final da linha.
 - Em caso de dúvidas, faça a simulação usando o Proteus. Os arquivos dos circuitos também estão disponíveis para download no site.
 - Lembre de mudar os endereço dos “includes” para a pasta onde está o arquivo que está sendo incluído no código.
 - Lembre também de procurar o arquivo a ser simulado na configuração do microcontrolador no Proteus.
-
- **Prazo para entrega:** 18/04/2013
 - **Site:** www.muriloleal.com.br
 - **E-mail:** murilo@muriloleal.com.br

