

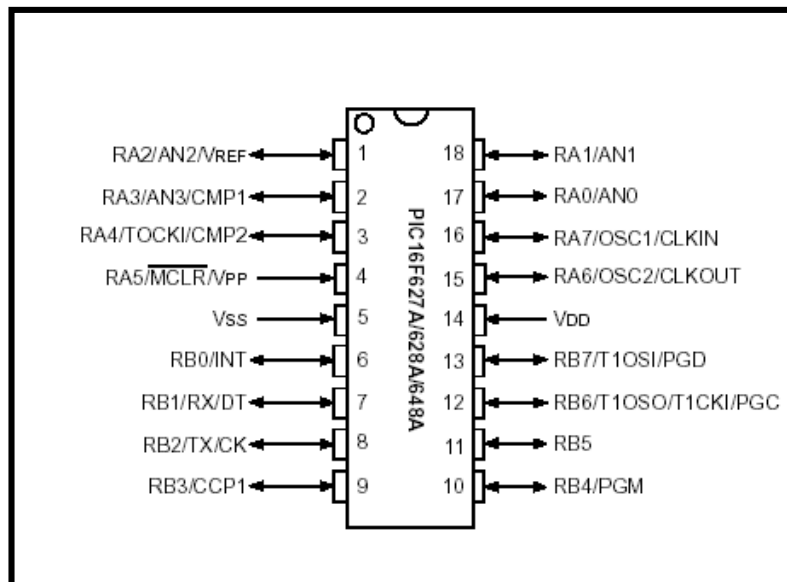


## Seqüencial de Leds utilizando o microcontrolador PIC16F628A

Vitor Amadeu

A Cerne Tecnologia e Treinamento, estará neste mês de agosto, abrindo várias seções mostrando o que é, para que servem e onde são utilizados os microcontroladores. Este evento ocorrerá na própria sede da empresa e todos estão convidados, sendo necessário a inscrição por parte do leitor. O certificado de presença e entrada são totalmente gratuitos. Ligue para (21) 3064-4526 ou envie um e-mail para [cerne@cerne-tec.com.br](mailto:cerne@cerne-tec.com.br) e reserve sua vaga.

Dando prosseguimento a utilização dos microcontroladores no nosso dia-a-dia, mostrarei neste artigo, como é feito um seqüencial de leds com esta tecnologia. O microcontrolador PIC16F628A possui 16 portas de I/O (entrada e saída) usadas para comunicação com o mundo externo. Abaixo está representado a pinagem deste chip:

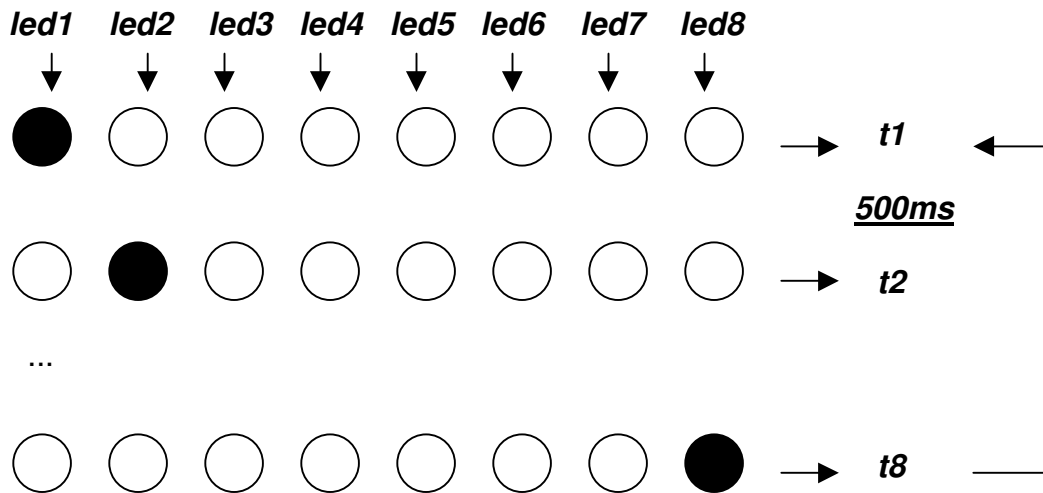


Os pinos de I/O (entrada e saída) são divididos em dois PORTS, sendo estes o PORTA e o PORTB. É através destes pinos que o microcontrolador terá acesso ao mundo externo, podendo controlar um led ou ler o estado de um botão. Verifique que os pinos 1, 2, 3, 4, 15, 16, 17 e 18 sempre começam com RAX, onde X é o número da porta, podendo variar de 0 até 7. Mesma coisa é válida para os pinos 6, 7, 8, 9, 10, 11, 12 e 13, onde os pinos começam com RB. No pino 5 fica ligada a tensão negativa da fonte enquanto no pino 14 a tensão positiva. Este microcontrolador funciona de 3V até 5,5 V. Os pinos indicados pela seta  $\rightarrow$  indicam que o pino tem somente um sentido sendo desta forma unidirecional. Já os pinos indicados pela seta  $\leftrightarrow$  indicam que o pino é bidirecional, podendo ser configurado tanto como entrada ou como saída.

Note que além de pino de I/O, um pino pode ter muito mais funções. Vejamos o caso do pino 6, designado como RB0/INT. Este pino tem duas funções, sendo a primeira de I/O, chamada de RB0. A segunda, é que ele também é um pino de interrupção externa. Desta forma, o pino pode funcionar como I/O, ou seja, RB0 ou como entrada de interrupção externa. Quem define de que forma este pino irá funcionar, são os registradores de função especial (SFR).

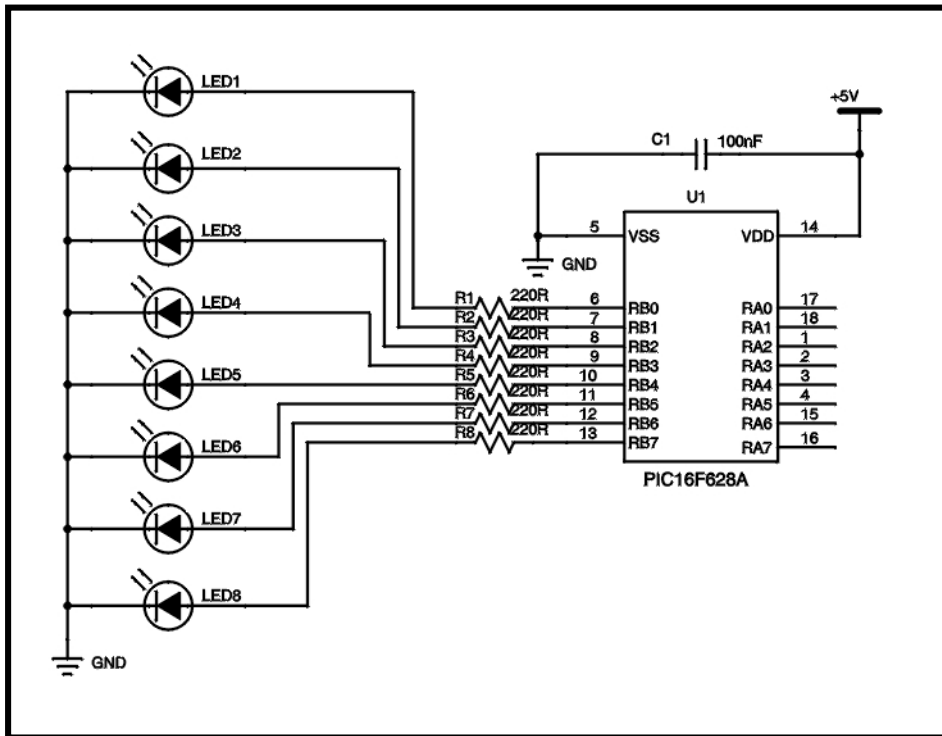
Utilizarei 8 pinos de I/O do microcontrolador para ligar os oito leds. Os leds acenderão em intervalos diferentes, de 500ms em 500 ms.

Teremos a seguinte impressão assim que este projeto estiver funcionando:

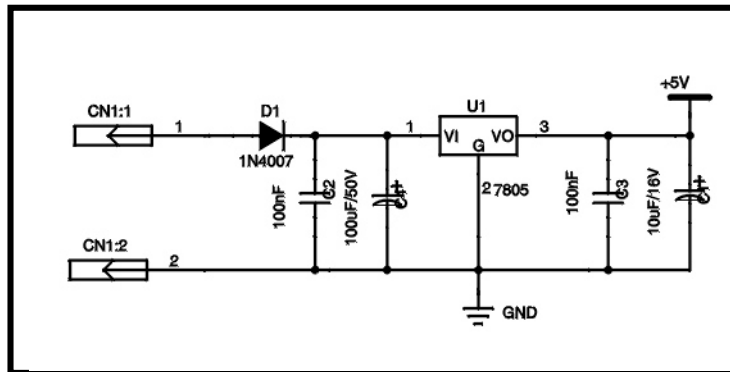


Note que o sistema começa deixando apenas o led1 aceso. Após 500 ms, o led1 apaga e o led2 fica aceso. Este processo ocorre até o led8, quando o processo volta para o primeiro led.

O esquema elétrico para este projeto é o seguinte:



Observe que para o PIC funcionar, é necessário ter uma tensão estável de 5VCC. Recomento a montagem da seguinte fonte para alimentação do microcontrolador:



A lista de material para confecção deste projeto é a seguinte:

Para a placa seqüencial:

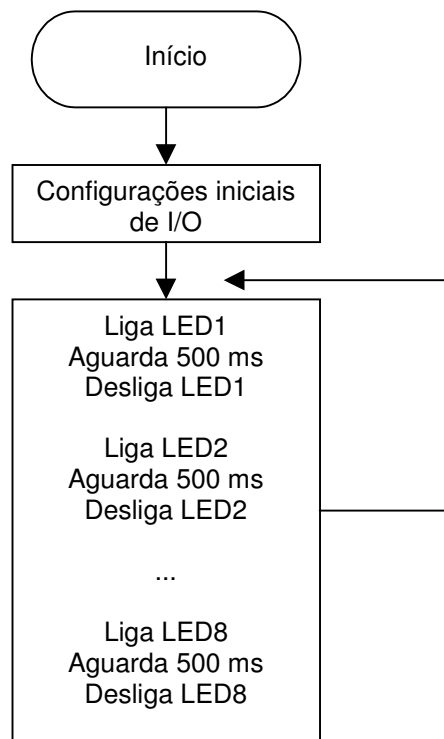
- Microcontrolador PIC16F628A;
- 8 Resistores de 220 R ¼ W;
- 8 Leds de 3 mm ou 5 mm vermelho, amarelo ou verde;
- 1 Capacitor de 100 nF;
- Fios, alicate e etc.

Para a fonte de 5 VCC:

- 01 Conector KRE02;
- 01 Diodo 1N4007;
- 02 Capacitor cerâmico 100 nF;
- 01 Capacitor eletrolítico 100uF/50 V;
- 01 Capacitor eletrolítico 10uF/16 V;
- 01 Regulador de tensão 7805.

Estes circuitos podem ser facilmente comprados na Cerne Tecnologia através do telefone (21) 3064-4526 ou através do site [www.cerne-tec.com.br](http://www.cerne-tec.com.br).

O fluxograma que irá governar o funcionamento deste exemplo é o seguinte:



Novamente, o código para controlar este projeto, foi escrito na linguagem BASIC baseado no compilador mikroBASIC. Este compilador pode ser baixado gratuitamente no site [www.mikroelektronika.co.yu](http://www.mikroelektronika.co.yu).

O código que “roda” dentro do microcontrolador é o seguinte:

```

program sequencial      'nome do programa

symbol led1 = portb.0   'define o pino do led1
symbol led2 = portb.1   'define o pino do led2
symbol led3 = portb.2   'define o pino do led3
symbol led4 = portb.3   'define o pino do led4
symbol led5 = portb.4   'define o pino do led5
symbol led6 = portb.5   'define o pino do led6
symbol led7 = portb.6   'define o pino do led7
symbol led8 = portb.7   'define o pino do led8

main:

    trisb=0                'configura todos os pinos do PORTB como saída

loop_principal:

    led1=1                 'liga o led1
    delay_ms(500)         'aguarda 500 ms
    led1=0                 'desliga o led1

    led2=1                 'liga o led2
    delay_ms(500)         'aguarda 500 ms
    led2=0                 'desliga o led2

    led3=1                 'liga o led3
    delay_ms(500)         'aguarda 500 ms
    led3=0                 'desliga o led3

    led4=1                 'liga o led4
    delay_ms(500)         'aguarda 500 ms
    led4=0                 'desliga o led4

    led5=1                 'liga o led5
    delay_ms(500)         'aguarda 500 ms
    led5=0                 'desliga o led5

    led6=1                 'liga o led6
    delay_ms(500)         'aguarda 500 ms
    led6=0                 'desliga o led6

    led7=1                 'liga o led7
    delay_ms(500)         'aguarda 500 ms
    led7=0                 'desliga o led7

    led8=1                 'liga o led8
    delay_ms(500)         'aguarda 500 ms
    led8=0                 'desliga o led8

    goto loop_principal   'volta para o loop_principal

```

Vamos entender melhor o código que está escrito. Na primeira linha, é informado o nome do programa através do comando ***program nome\_do\_programa***. Logo em seguida, são declarados oito símbolos, especificando um nome para cada led. O registrador *trisb* define a direção do portb. Quando cada bit deste registrador estiver em 1, o respectivo pino estará configurado como entrada e quando estiver em 0, como saída. Como foi atribuído o valor 0 (zero) para todo o registrador *trisb*, o PORTB ficou configurado totalmente como saída. Depois deste ponto, inicia-se o processo de oscilação do led. O comando *led1=1* faz com que o nível lógico alto (5 V) esteja presente na saída. Assim, o led acenderá. Em seguida, a rotina de retardo *delay\_ms(500)* faz com que o programa pare naquele ponto por 500 ms. Logo abaixo, o comando *led1=0* faz com que o mesmo led apague. O fluxo continua assim até o *led8*, quando o comando *goto loop\_principal* faz com que o programa volte a executar tudo novamente, acendendo e apagando o led.

Caso você não tenha um gravador para transferir o código para o microcontrolador, contate a Cerne Tecnologia e Treinamento, pois nós temos um excelente gravador de baixíssimo custo e que grava a maioria dos modelos da linha flash da Microchip.

Pois bem, este foi o segundo exemplo prático com a utilização dos microcontroladores. Espero que tenham gostado. Qualquer dúvida, crítica ou sugestão, não hesitem em me contactar. O meu e-mail de contato é ***vitor@cerne-tec.com.br*** e telefone (21) 3064-4526. É isso aí, até breve!