

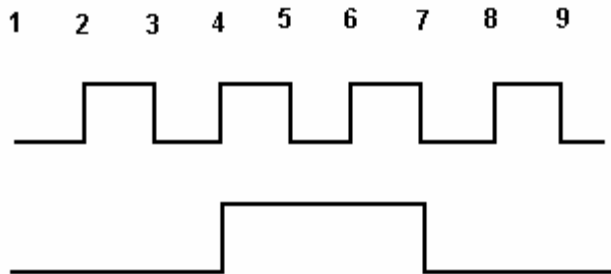
Aprendendo a programar com o BASIC Step M8

Neste artigo nós vamos escrever um programa que fará o BASIC Step M8 gerar duas frequências diferentes simultaneamente. Você deve escrever este programa e fazê-lo funcionar. Ele será muito útil para testar o programa do próximo artigo.

Sendo assim, não se preocupe se não entender o programa totalmente agora, nesta fase de aprendizagem. É realmente muita coisa para quem está iniciando, mas nós faremos o programa e você poderá usá-lo como auxílio para o próximo artigo.

A idéia por trás deste programa é aprender a usar o timer/contador (chamado TIMER0) do processador para uma base de tempo. Usando esta base de tempo, o nosso programa irá variar as duas frequências de saída de acordo com uma chave DIP-switch. Isto vai permitir que o usuário modifique qualquer uma das duas saídas independentemente apenas mudando a posição das chaves.

O TIMER0 é um contador livre. Ele é um hardware no chip do BASIC Step M8 que incrementa continuamente, em tempo real, sem qualquer intervenção do programa. Através do programa nós podemos ler este contador a qualquer momento. Para ajudar a entender a idéia, a figura a seguir mostra a forma de onda do contador e duas saídas, onde uma é $\frac{1}{2}$ da frequência do timer e a outra $\frac{1}{6}$ da frequência. Os números no alto da figura são os estados do TIMER0 e as formas de onda as duas saídas.



Para usar o TIMER0, nós devemos utilizar o comando TIMER0 da linguagem BASIC para ligá-lo. O comando terá a seguinte forma: `TIMER0 ON N`, onde N pode ser 1, 8, 64, 256 ou 1024. N é chamado de prescaler, ele diminui a frequência do oscilador interno dividindo a por N. Então se N for 64 e como o BASIC Step M8 roda a 8MHz, o TIMER0 será incrementado a uma frequência de 8MHz dividido por 64, ou 125KHz.

Outro comando que iremos usar será o PAUSE. Ele simplesmente para a execução do programa por um número de milissegundos especificado após a palavra PAUSE. Em nosso

Conhecendo o BASIC Step M8 – parte 2

caso iremos colocar um comando PAUSE 50 no início do programa para que as tensões nas chaves se estabilizem antes de lê-las.

Uma outra idéia nova introduzida neste programa é o comando OUTPORT para uma porta de entrada. A porta D é a nossa porta de saída para as duas frequências geradas e a porta B é a porta onde iremos ler as chaves. Quando nós escrevemos 1 em qualquer bit de uma porta de entrada, nós ligamos um resistor interno ao chip do pino correspondente ao +5V. Fazendo isto para toda a porta B, a chave de 8 posições pode ser lida sem nenhum outro componente.

As variáveis L1 e L2 em nosso programa são variáveis byte e são usadas para armazenar os valores para cada frequência de saída.

A variável SFTCNT é byte e controla a escala das frequências. Ela é lida apenas quando ligamos a alimentação e se mantém fixa até a alimentação ser desligada. Antes de ligarmos o processador, devemos colocar as chaves nas posições desejadas para que sejam lidas e armazenadas em SFTCNT.

Sendo F a frequência de clock do processador e B o número setado nas chaves, a frequência máxima de qualquer saída será:

$$F_{max} = F / 2 / (B + 1)$$

E a frequência mínima:

$$F_{min} = F / 32 / (B + 1)$$

Alguns outros comandos novos usados neste programa são INCR, IF, END IF, WHILE e WEND. Estes comandos serão explicados em detalhes em artigos futuros, mas se quiser conhecê-los melhor agora, veja o arquivo de ajuda no Compilador.

No programa a seguir (que chamaremos de FGEN.BAS) nós lemos as 8 chaves e armazenamos em SFTSCN, configuramos o TIMER0 para incrementar a frequência do processador dividido por 64, colocamos valores temporários em L1 e L2 que estarão válidos até a primeira transição de cada frequência de saída. A partir de então as frequências serão definidas pelas posições que as chaves estavam quando o programa iniciou.

No loop principal do programa, nós primeiro decrementamos as variáveis de controle de saída (L1 ou L2) e verificamos se chegaram em 0. Quando chegarem em 0, nós mudamos o pino da saída e carregamos a variável de controle novamente.

O comando SHIFT L2, 4, RIGHT faz com que os 4 bits mais significativos das chaves sejam movidos para as 4 posições menos significativas de L2, fazendo com que as 4 chaves mais significativas controlem a frequência de saída 2 exatamente como as 4 menos significativas controlam a saída 1. O comando SHIFT será explicado em detalhes em artigos futuros mas você pode ler sobre ele no arquivo de ajuda do Compilador.

Conhecendo o BASIC Step M8 – parte 2

Depois de verificarmos se devemos mudar o estado da saída ou não, chegamos ao comando FOR. Este é um comando de repetição que conta de 1 até SFTCNT. Dentro da repetição nós temos um comando WHILE que faz o programa esperar até que o TIMER0 seja incrementado. Então o loop FOR irá parar o programa até ocorrerem SFTCN incrementos do TIMER0.

Após o comando FOR nós colocamos o comando TOGGLE para o pino D,2. Este comando inverte o estado da saída de um pino. Aqui ele é usado somente para teste. Não importa o número que você colocou nas chaves, o pino D,2 terá uma frequência constante de 8MHz dividido por 128.

Aqui está o programa completo.

```
DIRPORT D,OUT
'DIRPORT B,IN
OUTPORT B,&HFF
'=====
'=      B é a porta de entrada. 4 bits controlam a saída 1   =
'=      e 4 a saída 2.                                         =
'=      A saída 3 é timer0 freq / 2.                             =
'=====
EQU "D,0","OUT1"
EQU "D,1","OUT2"
EQU "D,2","OUT3"

PAUSE 50      '..espera estabilizar

INPORT sftcnt,B
INCR sftcnt   '..a frequência é controlada por sftcnt
              '... 1 <= sftcnt <= 256 ....

TIMER0 ON 64  '..TIMER FREQ é 125000 Hz...
              '..Max OUTPUT é 62500 HZ/sftcnt...
              '..Min OUTPUT é 3906 HZ/sftcnt...

L1 = 10      '..divisor inicial para saída 1
L2 = 11      '..divisor inicia para saída 2

'----- Inicio do programa principal -----
MAIN:  DO
        '..inverte a saída 1?....
        DECR L1
        IF L1 = 0 THEN
            INPORT temp,B
            L1 = temp AND &H0F
            INCR L1
            TOGGLE "OUT1"
        END IF

        '..inverte a saída 2?....
        DECR L2
        IF L2 = 0 THEN
            INPORT L2,B
            SHIFT L2,4,RIGHT
            INCR L2
```

Conhecendo o BASIC Step M8 – parte 2

```
    TOGGLE "OUT2"  
END IF  
  
FOR temp = 1 TO sftcnt  
    '...espera até o próximo incremento to TIMER0...  
    TIMER0 READ ctime  
    WHILE ctime = oldtime  
        TIMER0 READ ctime  
    WEND  
    oldtime = ctime  
    TOGGLE "OUT3"      '...saída 3 em TIMER0/2...  
NEXT  
LOOP
```

'----- Fim do programa principal -----'

Você pode executar este programa na placa Super StepLab para ver como ele funciona. Ela possui a chave dip-switch de 8 posições.

No próximo artigo nós iremos precisar de duas frequências diferentes. Seria interessante ter dois processadores, um com este programa controlando o outro com o próximo programa.