

## Laboratório 6

# ESTRUTURAS TEMPORIZADAS

### 6.1 Objetivos

Estudar e investigar os recursos de temporização com *timers* de tempo real em estruturas de sequenciamento de tarefas, tipo SEQUENCE e de *Loops* tipo WHILE e FOR.

### 6.2 Generalidades

Para propósitos de controle e em muitas atividades de supervisão de sistemas é essencial a disponibilidade de coordenação de rotinas e tarefas sendo realizadas em tempo real. Tempo real neste conceito significa realizar uma tarefa em períodos exatos de tempo contados em segundos.

A geração de tempo real normalmente é feita através de um *Timer (hardware)* ou contador digital de 32 Bits, o qual recebe um *clock* de uma unidade estável de tempo, como por exemplo um Cristal ou outro dispositivo de frequência estável.

No sistema em uso no Laboratório existem duas fontes de *clock* que podem ser utilizadas. Uma delas é o *clock* do *Host* ou seja, do próprio microcomputador e a outra fonte é disponível da própria placa o *LabAcquisition*.

Quando os *timers* ou contadores não estão disponíveis, alguma placa adicional deve ser incorporada para se gerar o *clock* necessário. Além dos contadores e *timers* é necessário providenciar algum meio de se programar estes componentes de modo a se especificar os desejados tempos de operação. Normalmente a operação dos *timers/counters* produzem interrupções a nível de *hardware* ou de *software* em forma de *loops* de espera e então coordenam a execução e rotinas de interrupção e sequenciamento.

No LabVIEW são disponíveis várias maneiras de se estabelecer algum tipo de sequenciamento temporizado em tempo real através de sub-VIs disponíveis em *Pop-Menus*. Neste menu são disponibilizadas funções básicas que podem ser utilizadas para executar ou implementar funções de temporização mais elaboradas. As funções básicas de temporização possibilitam ainda utilização ou geração de variáveis de data contendo Dia/Mês/Ano incluindo hora/minuto/segundo e mesmo conversão destas grandezas entre formatação numérica e caracteres.

O acesso aos recursos de temporização é obtido no Pop-Menu da janela DIAGRAMA FUNCTIONS-PROGRAMMING *Timing*. As opções de interesse neste Sub-Menu são: *Tick Count (ms)*, *Wait (ms)* e *Wait Until Next (ms) Multiple*. Estas três formas básicas são vistas na primeira linha da figura 7.1. Na linha seguinte se encontram os elementos tipo sub-VIs para manipulação de variáveis tipo data.

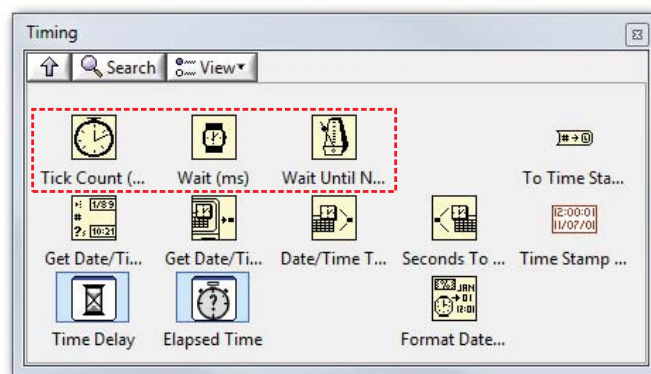
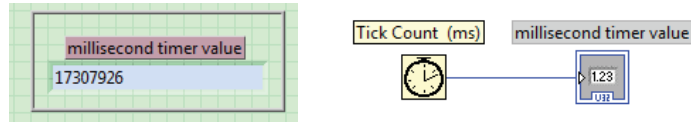


Figura 6.1: Recursos para Temporização do Menu *Functions*

### 6.2.1 Opção *Tick Count (ms)*

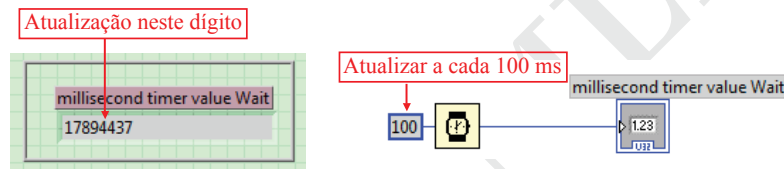
Este elemento ou sub-VI fornece a leitura da contagem do *Timer* interno do microcomputador ou *Host*. Não há como inicializar este elemento, pois a leitura corresponde ao valor de contagem atual desde que o *Host* foi ligado. Como *Timer* interno é de 32 Bits, o valor deste elemento só ira indicar zero ao contar de  $(2^{32} - 1)$  para 0. Portanto deve-se tomar cuidado ao utilizar este elemento.

A indicação numérica fornecida na saída deste elemento revela o total de milisegundos e portanto é representado por um valor numérico grande ou seja, com muitos dígitos.

Figura 6.2: Exemplo do *Tick Count (ms)*

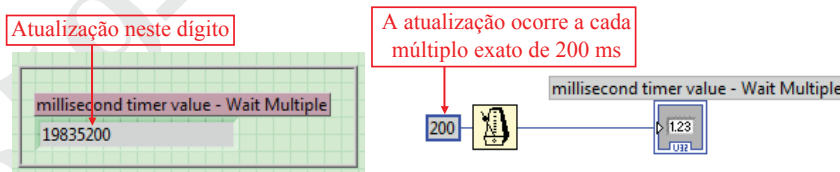
### 6.2.2 Opção *Wait (ms)*

Este elemento usa uma entrada em que se determina um valor em milissegundos até que este elemento atualize sua saída. Desta forma se a entrada for estabelecida em N milissegundos, este elemento conta de N em N milissegundos.

Figura 6.3: Exemplo do *Wait (ms)*

### 6.2.3 Opção *Wait Until Next (ms) Multiple*

Este elemento tem praticamente a mesma função do caso anterior, porém ele indicará e executará uma tarefa ou Loop a cada múltiplo exato do valor conectado à entrada.

Figura 6.4: Exemplo do *Wait Until Next Multiple (ms)*

Em geral este elemento não é o mais indicado para se estabelecer tempo real em um *Loop*, pois a primeira contagem de tempo até o primeiro múltiplo exato desejado poderá ser diferente do intervalo de tempo inicializado. Em ambos os casos, ou seja, no caso de *Wait* e *Wait Multiple* uma escala de tempo com início em zero segundos deve ser obtida utilizando-se o valor da iteração ( $i$ ) do *Loop* FOR ou WHILE, multiplicado pelo valor de espera conectado nas entradas dos elementos *Wait* ou *Wait Multiple*.

### 6.2.4 Opção *Time Delay*

Este elemento opera exatamente como o caso anterior, porém a inicialização é feita na janela de propriedades clicando-se duas-vezes no ícone



Figura 6.5: Exemplo do *Delay Time (ms)*

### 6.2.5 Opções de Data e Manipulação

Este recurso pode ser útil na documentação ou simples indicação de atualização em painel. Um exemplo é indicado a seguir onde se requer apenas a indicação de uma data momentânea.

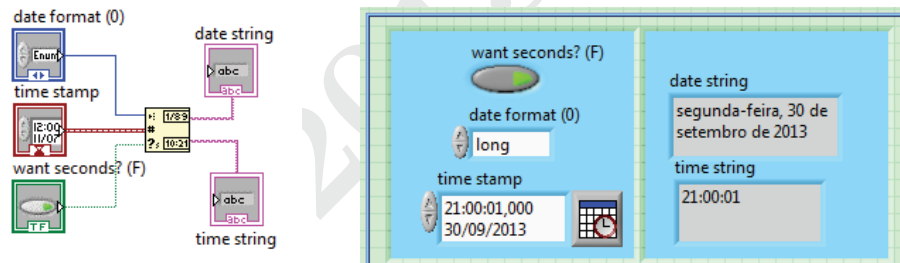


Figura 6.6: Exemplo do recurso de Manipulação de Datas

## 6.3 Atividade de aplicação

Procure executar os VIs com as funções solicitadas a seguir. Procure treinar também a geração e gerenciamento do VI por meio de um painel equivalente dotado entradas, saídas e mensagens *strings*.

- i) - Implemente e teste os recursos de geração e manipulação de datas;

ii) - Teste o elemento *Tick* (ms) , *Wait* (ms) e *Time Delay* sem utilizar *loops*. Execute o VI na forma “Uma Vez” e “Continuamente”.

iii) - Gere um VI com uma estrutura de *Loop WHILE*, insira um elemento *Wait* devidamente inicializado e execute alguma operação. Colete o resultado indexadamente em uma variável.

iv) - Use agora no VI anterior um elemento *Wait Multiple* e troque a forma de *Loop* para FOR e repita usado *Time Delay*.

v) - Incremente o VI de um dos itens anteriores com um painel dotado de uma saída gráfica.

vi) - Experimente inserir um *Loop WHILE* e altere-o para um *Timed Loop* usando fonte de *clock* interna.

#### Desafio Prática 6

vii) - Implemente em um dos VIs dos itens anteriores um procedimento para se gerar 3 ciclos de 60Hz com 20 pontos por ciclo. Cada grupo deverá gerar senoides com amplitude correspondente ao número do grupo.

viii) - Exporte estes pontos em arquivo ou use o recursos do MATLAB-Script. Importe no Matlab os dados gerados no LabVIEW, gere no Simulink o mesmo sinal com 100 pontos e compare no mesmo gráfico os dois resultados.

Salve o VI atual para ser utilizado na aula seguinte.

OBS - OS ITENS ( vii ) E ( viii ) DEVERÃO CONSTAR NO RELATÓRIO 1.  
Indicar VI e procedimento Matlab/Simulink e Resultado