

BIBLIOTECA LIBTOPL



EXEMPLO DE USO

Tutorial de utilização da Biblioteca LibTopL para interface com teclados de satisfação USB com protocolo WYMA.

Rev.0 – 03/05/2017 – Revisão inicial

Índice

1. Introdução	2
2. Requisitos para iniciar	2
3. Instalando	3
4. Software aplicativo exemplo	6
5. Biblioteca LibTopL	6
6. Código exemplo em C#.NET	9
7. Registrando os arquivos typeLib (tlb), no Windows.....	9
8. Criando os componentes no Delphi.....	11
9. Criando um projeto no Delphi	13

1. Introdução

A Biblioteca de Comunicação LibTopL é um arquivo de extensão de aplicativo (DLL) com diversas funções prontas, que embutem e simplificam a interface com dispositivos com protocolo WYMA que precisem enviar comandos para outro sistema.

A biblioteca foi concebida usando a tecnologia Microsoft .NET 4, portanto, para sua utilização será necessário possuir no mínimo a mesma versão, além das ferramentas de desenvolvimento. O framework pode ser adquirido no link abaixo:

<https://www.microsoft.com/pt-br/download/details.aspx?id=24872>

Para este projeto, utilizou-se o Visual Studio 2015, que pode ser instalado gratuitamente a partir do site da Microsoft, disponível no link:

<http://www.visualstudio.com/downloads/download-visual-studio-vs>

Selecione uma versão com linguagem C#.

2. Requisitos para iniciar

Antes de iniciar o projeto, verifique se recebeu todos os arquivos necessários. Você deve ter os seguintes arquivos:

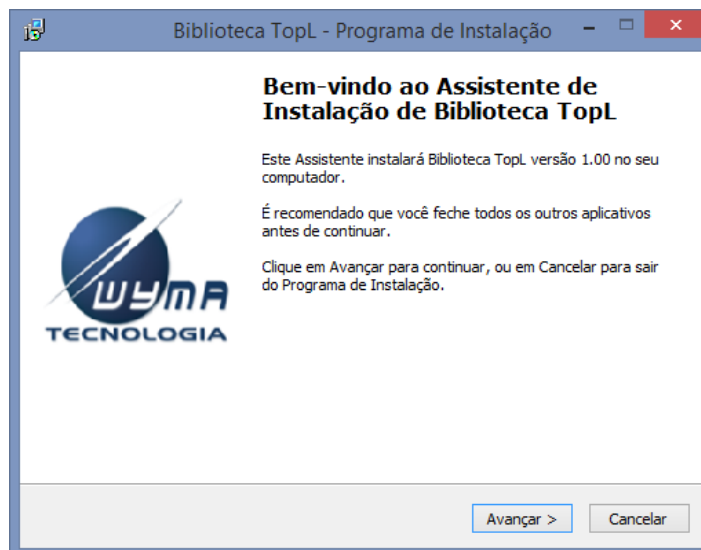
1-setup.exe = Instalador dos arquivos de código em C#.NET para testar a comunicação, aplicativo exemplo e arquivos de biblioteca “DLL” e “TLB”.

2-LibTopL – Exemplo de uso.pdf = este manual de utilização

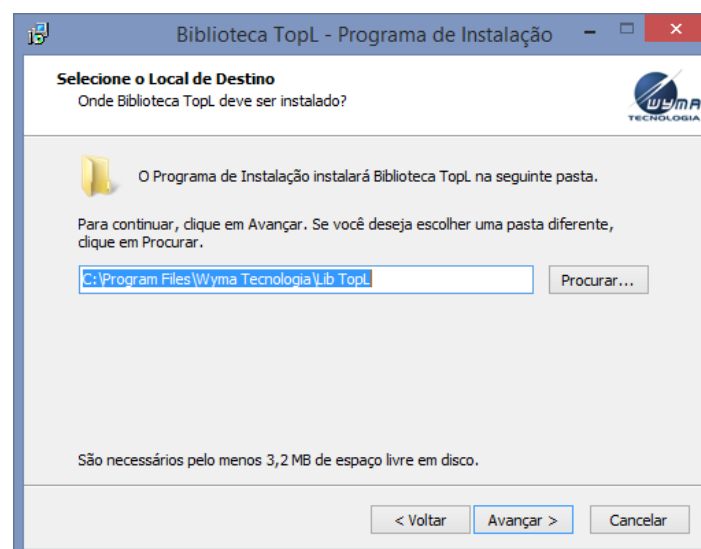
Também é necessário que o dispositivo WYMA esteja ligado, e conectado à rede. Também deve ter o endereço IP configurado (consulte o manual produto).

3. Instalando

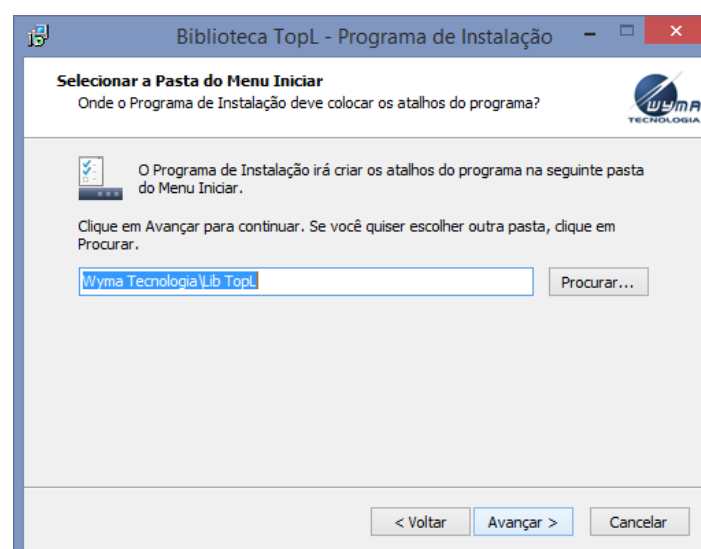
Localize o arquivo “**setup.exe**” e com dois cliques abra o instalador. Clique no botão “**Avançar**” para iniciar a instalação.



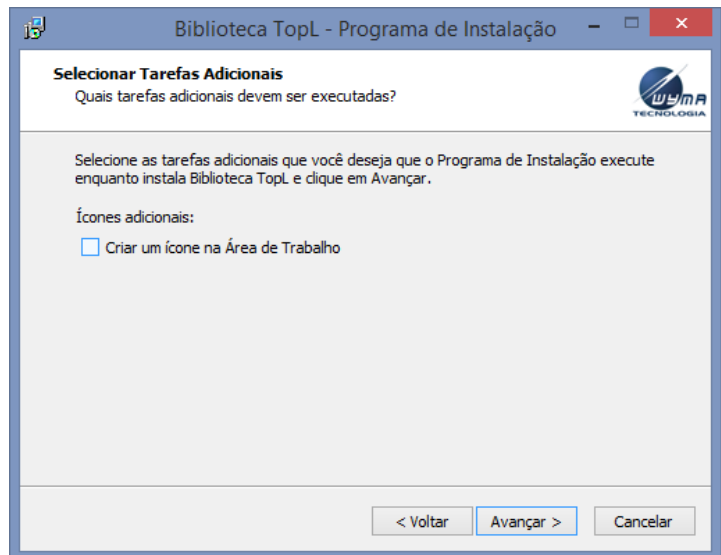
Sugerimos deixar a mesma pasta de instalação que será usada como referência neste documento.



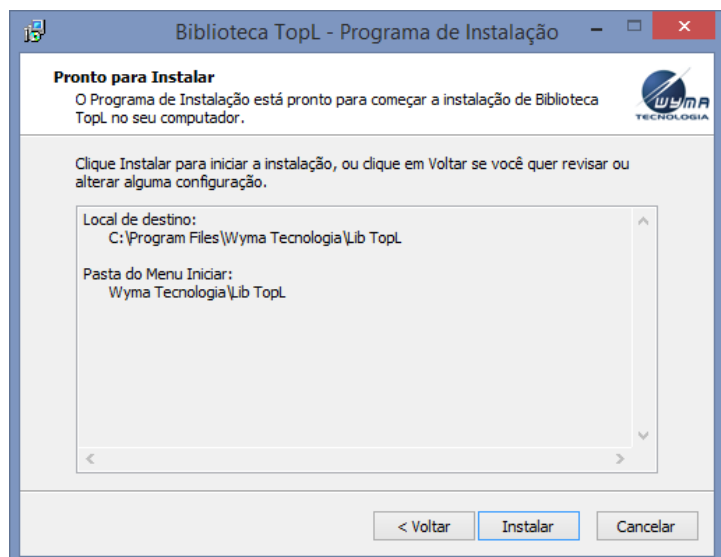
Esta é a pasta de atalhos para o aplicativo de teste.



Se desejar criar um atalho na área de trabalho marque a opção ao lado.



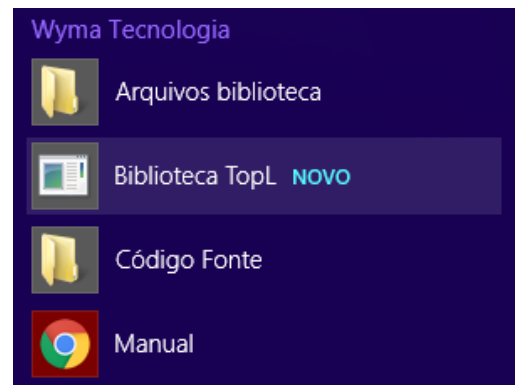
Este é o resumo das opções escolhidas para a instalação, clique em instalar para finalizar.



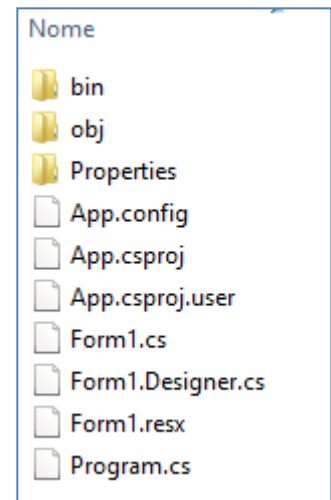
Marque a opção “Executar” para iniciar o aplicativo assim que concluir a instalação.



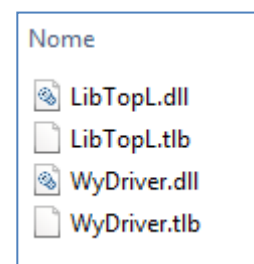
Após a instalação alguns atalhos serão criados dentro da pasta “Wyma Tecnologia” o nome do atalho para o aplicativo será “Lib TopL”. Note que existe um atalho para a pasta “**Código fonte**” com o código de um projeto exemplo e também um atalho para uma pasta “**Arquivos biblioteca**”, onde estão os arquivos dll e tlb.



Clique no atalho “**Código Fonte**” para abrir a pasta com os arquivos do projeto.



Clique no atalho “**Arquivos biblioteca**” para abrir a pasta com os arquivos “dll” e “tlb”.



4. Software aplicativo exemplo

O software aplicativo exemplo deve ser usado para testar a interface entre o computador com Windows XP ou superior e os dispositivos WYMA.

Depois de instalado o software no computador conforme explicado anteriormente, será necessário conectar o teclado de satisfação na porta USB do computador.



5. Biblioteca LibTopL

Propriedades:

- bool _isPlugged** – Informa o status de conexão do teclado na USB (R).
- string _status** – Informa o status de comunicação com o teclado (R).
- Double _voltage** – Informa a voltagem do teclado conectado, caso o teclado tenha bateria (R).
- string _dateTime** – Informa a data e hora do teclado conectado (R).
- bool _newInfo** – Informa que recebeu informações de um teclado que se conectou na porta USB (RW). Após ler os dados configure o valor “false”.
- int _numResults** – Informa o número de resultados disponíveis no teclado para upload (R).
- byte _model** – Informa o modelo do teclado conectado (R).
- byte _version** – Informa a versão do teclado conectado (R).
- bool _newResult** – Informa que chegou um resultado de pesquisa de opinião (RW). Após ler os dados configure o valor “false”.
- ushort _resultTicket** – Resultado da votação: Número do ticket relacionado ao voto (R).
- char _resultVote** – Resultado da votação: Nota do cliente (R).
- string _resultDateTime** – Resultado da votação: Hora e data do voto (R).
- ushort _resultPeriod** – Resultado da votação: Período de tempo da votação (R).

char **_resultEnd** – Resultado da votação: Tipo de encerramento da pesquisa
 ‘N’=Normal / ‘T’=Tempo esgotado (R).

byte **_resultUser** – Resultado da votação: Número do usuário relacionado ao
 voto (R).

Métodos:

Nome:

Connect

Descrição:

Realiza a conexão USB com o dispositivo.

Cabeçalho:

```
bool Connect()
```

Parâmetros: Nenhum

Retorno: 1 = Indica que a conexão foi bem sucedida

Nome:

Disconnect

Descrição:

Realiza a desconexão do USB com o dispositivo.

Cabeçalho:

```
void Disconnect()
```

Parâmetros: Nenhum

Retorno: Nenhum

Nome:

StartListening

Descrição:

Começa a receber os pacotes USB.

Cabeçalho:

```
void StartListening()
```

Parâmetros: Nenhum

Retorno: Nenhum

Nome:

StopListening

Descrição:

Interrompe o recebimento de pacotes USB.

Cabeçalho:

```
void StopListening()
```

Parâmetros: Nenhum

Retorno: Nenhum

Nome:

StatusSurvey

Descrição:

Envia comando para o teclado de opinião para mudar o status da pesquisa.

Cabeçalho:

bool StatusSurvey (byte, bool)

Parâmetros: byte userNum = Número do usuário

bool statusSurvey = (0)desabilita pesquisa (1) Habilita pesquisa

Retorno: 1 = Comunicação bem sucedida

Nome:

SetDateTime

Descrição:

Configura o horário do dispositivo.

Cabeçalho:

bool SetDateTime (string)

Parâmetros: string dateTime = Data e Hora para atualizar

Retorno: 1 = Comunicação bem sucedida

Nome:

GetDateTime

Descrição:

Recupera o horário do dispositivo.

Cabeçalho:

bool GetDateTime()

Parâmetros: Nenhum

Retorno: 1 = Comunicação bem sucedida

Nome:

GetVoltage

Descrição:

Recupera a tensão do dispositivo.

Cabeçalho:

bool GetVoltage ()

Parâmetros: Nenhum

Retorno: 1 = Comunicação bem sucedida

Nome:

GetResult

Descrição:

Recupera os resultados da pesquisa.

Cabeçalho:

bool GetResult ()

Parâmetros: Nenhum

Retorno: 1 = Comunicação bem sucedida

6. Código exemplo em C#.NET

Iniciaremos a análise do código abrindo o Visual Studio e clicando na sequência de menus FILE → OPEN PROJECT, abra o projeto instalado em "C:\Arquivos de programas\Wyma Tecnologia\Lib TopL\source\App".

Com o projeto aberto localize "**Solution Explorer**" e expanda o projeto "App" e em seguida expanda a pasta "**References**" note que a dll "**libTopL**" aparece nas referências do projeto.

Agora clique em **Form1.cs** e vamos abrir o código para ver como é feito o uso da DLL.

Nossa DLL e variável para identificar se o teclado esta conectado, estão no começo do código:

```
ITopL TOPL = null;  
private bool usbConnected = false;
```

Note que criamos uma nova instância do objeto na chamada do construtor da classe, através da função Connect_TOPL():

```
TOPL = new LibTopL.TopL();
```

Um timer é habilitado no construtor para verificar o status da comunicação USB com o teclado, este timer verifica as "flags", que indicam quando ocorrem os eventos no teclado.

```
timer1_Tick (object sender, EventArgs e)
```

7. Registrando o arquivo typeLib (tlb), no Windows

Para usar a DLL em sistemas não microsoft .net, é necessário registrar os arquivos TLB que são distribuídos na instalação junto com os arquivos DLL, verifique na pasta de atalhos do programa o atalho para "**LibTopL → Arquivos biblioteca**", este é um atalho para a pasta onde são copiadas as DLLs e TLBs necessários para o funcionamento do software.

Para registrar os arquivos TLB, você precisa abrir o **Prompt de comandos** do Windows em modo **administrador**, clicando o botão direito do mouse sobre o atalho esta opção estará disponível.

Certifique-se de ter instalado o Framework indicado no começo do manual, ou será necessário mudar o path abaixo para o caminho de um framework instalado em seu computador.

Digite o seguinte comando:

```
set path="%path%";C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319
```

A linha de comando acima apenas vai facilitar o uso do aplicativo “**regasm**”, assim não teremos que digitar todo o caminho dele para que o aplicativo funcione, basta digitar “**regasm**”.

Agora vamos mudar o diretório do prompt de comandos para a pasta onde estão os arquivos TLB para executar o comando **regasm**.

Para registrar o WyDriver.TLB execute o comando:

```
regasm /tlb:WyDriver.tlb WyDriver.dll /codebase
```

Para registrar o LibTopL.TLB execute o comando:

```
regasm /tlb:LibTopL.tlb LibTopL.dll /codebase
```

OBS: Caso existir uma biblioteca de revisão anterior já registrada rode antes os mesmos comandos acima trocando o parâmetro “**/codebase**” por “**/u**”. E após desfazer o registro anterior você poderá registrar novamente.

Após estes comandos os componentes estarão prontos para uso.

A tela abaixo exemplifica alguns dos comandos digitados no prompt, note que ao final do registro de cada um dos TLB é indicado se o registro obteve sucesso.

```
Administrador: Linha de comandos
Microsoft Windows [Versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

M:\>c:

C:\>set path="%path%";C:\WINDOWS\Microsoft.NET\Framework\v4.0.30319

C:\>cd programas\wyna tecnologia\lib listen\dll
C:\Programas\wyna tecnologia\Lib Listen\dll>regasm /tlb:WyDriver.tlb WyDriver.dll /u
Microsoft .NET Framework Assembly Registration Utility version 4.0.30319.18408
for Microsoft .NET Framework version 4.0.30319.18408
Copyright (C) Microsoft Corporation. All rights reserved.

Types un-registered successfully
Type library 'C:\Programas\wyna tecnologia\Lib Listen\dll\WyDriver.tlb' un-registered successfully

C:\Programas\wyna tecnologia\Lib Listen\dll>regasm /tlb:WyDriver.tlb WyDriver.dll /codebase
Microsoft .NET Framework Assembly Registration Utility version 4.0.30319.18408
for Microsoft .NET Framework version 4.0.30319.18408
Copyright (C) Microsoft Corporation. All rights reserved.

Types registered successfully
Assembly exported to 'C:\Programas\wyna tecnologia\Lib Listen\dll\WyDriver.tlb',
and the type library was registered successfully

C:\Programas\wyna tecnologia\Lib Listen\dll>regasm /tlb:LibListen.tlb LibListen.dll /codebase
Microsoft .NET Framework Assembly Registration Utility version 4.0.30319.18408
for Microsoft .NET Framework version 4.0.30319.18408
Copyright (C) Microsoft Corporation. All rights reserved.

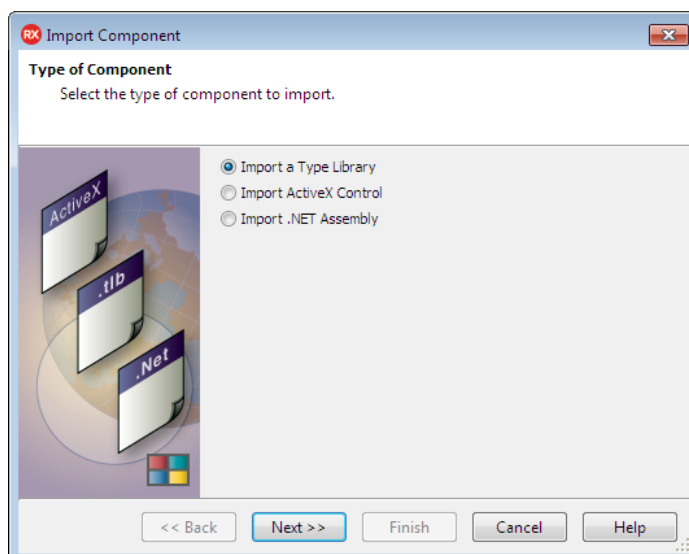
Types registered successfully
Assembly exported to 'C:\Programas\wyna tecnologia\Lib Listen\dll\LibListen.tlb',
and the type library was registered successfully

C:\Programas\wyna tecnologia\Lib Listen\dll>
```

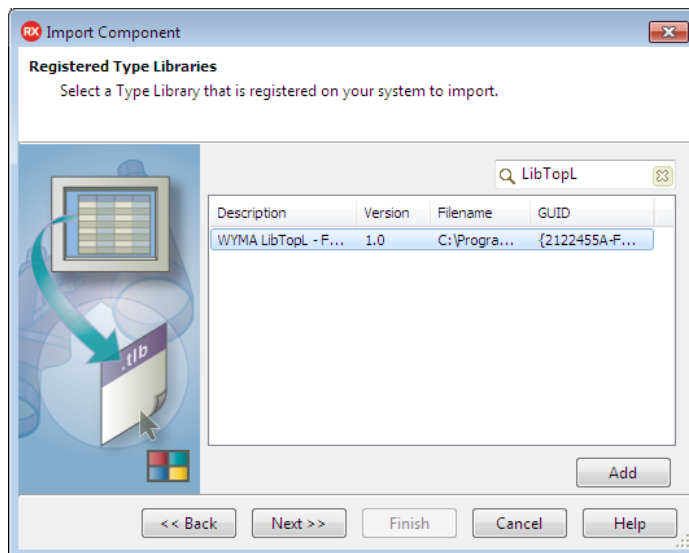
8. Criando os componentes no Delphi

O procedimento para usar a DLL como um componente no Delphi é bem simples, o exemplo abaixo foi feito no RAD Studio 10.1 Berlin.

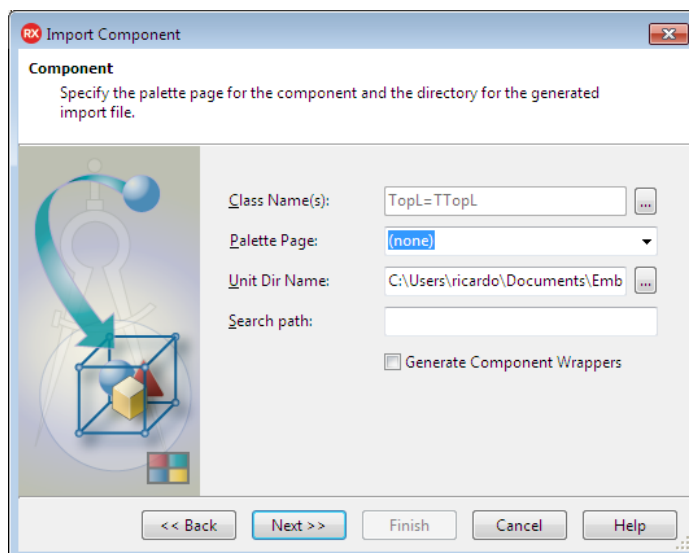
- Para criar os componentes vá até o menu em: **“Component”**, depois em **“Import Component”**.
- Selecione a opção: **“Import a Type Library”** e clique em **“Next”**:



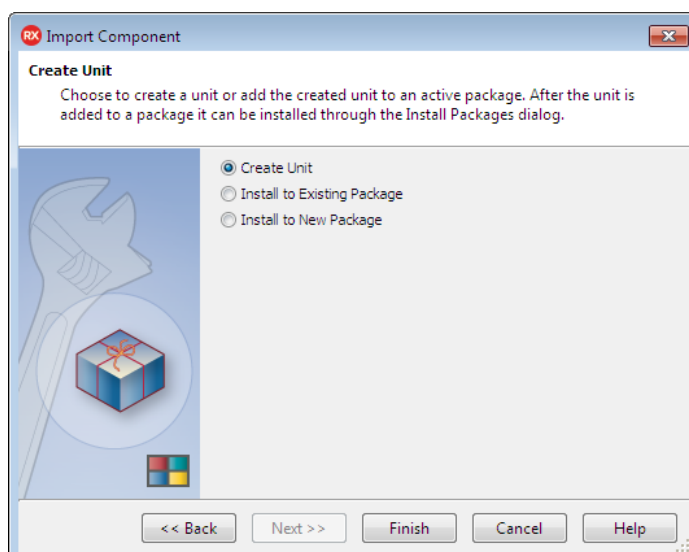
- Digite **“LibTopL.tlb”** na pesquisa escolha o item WYMA LibTopL e clique em **“Next”**:



d) Clique em **“Next”**:



e) Por fim escolha **“Create Unit”** e clique em **“Finish”**:

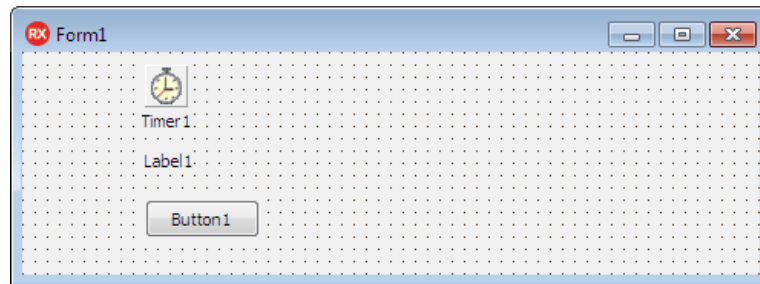


- f) Este componente **“LibTopL.tlb”** depende de outro componente chamado **“WyDriver.tlb”**, é necessário seguir as mesmas etapas acima (passos “a” até “e”), para importar esta outra biblioteca também.

9. Criando um projeto no Delphi

Agora podemos criar um projeto para usarmos estes novos componentes em nosso projeto, siga as etapas abaixo para começar um projeto novo:

- a) Clique em **“File”**, depois em **“New”** e finalmente em **“VCL Forms application – Delphi”** para criar um projeto novo.
- b) Localize a caixa de ferramentas **“Tool Palette”** e digite no campo de busca o texto: **“TLabel”** e acrescente um objeto ao formulário, depois digite **“TButton”** e acrescente um objeto, por fim digite também **“Ttimer”** e acrescente um objeto deste ao projeto conforme abaixo:



- c) Clique duas vezes sobre o formulário para irmos até o código.
- d) Localize mais acima no código a palavra chave **“uses”**. Em **“uses”**, acrescente as bibliotecas: **ActiveX**, **WyDriver_TLB** e **LibTopL_TLB**, estas bibliotecas são necessárias para o nosso projeto.
- e) Declare logo abaixo do TForm1 a nossa DLL:

```
var
  Form1: TForm1;
  dll: ITopL;
```

- f) Agora digite o código abaixo para os controles formulário, timer e button:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  dll.StatusSurvey(1,True);
  label1.Caption:='Enviou comando para o teclado';
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  dll:=CoTopL.Create();
  dll.StartListening();
```

```

if (dll.Connect()) then
begin
    label1.Caption:='Teclado conectado';
end;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
    if (dll._newResult) then
    begin
        dll._newResult:=false;
        label1.Caption:='Recebeu voto';
    end;
end;
end;

```

- g) Para testar com o teclado de satisfação, conecte o teclado e depois rode o software, após o label ser nomeado para “Teclado conectado”, clique no botão para habilitar o teclado para a pesquisa.