

Notes to the program hyperterm

(Dec/8/05)

Generalities

The program **hyperterm** is written in Visual Basic 6.0, Visual Basic.NET, Visual C++.NET 2003. It is based on the Microsoft Comm Control 6.0, which handles all the communication details.

The corresponding file is located at Windows/system32/MSCOMM32.OCX

It is invoked according to the following example:

```
MSComm1.Output = "out string"
```

Documentation on how to use the MSComm is available at:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/comm98/dt_vbobjComm_P.asp

<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/comm98/html/vbobjcomm.asp>

Visual Basic 6.0

The source code in Visual Basic 6.0 will be improved and maintained. Some entry-level versions of Visual Basic, like Visual Basic Learning Edition, do not allow the use of some licensable controls and will not work with MSComm 6.0.

Visual Basic Professional and Enterprise do work with MSComm 6.0

To use MSComm perform the following steps from the main window of the compiler:

- ◆ file Windows/system32/MSCOMM32.OCX must exist.
- ◆ Open Project menu; select components.
- ◆ Check the checkbox next to "Microsoft Comm Control 6.0". The control will appear in the toolbox.
- ◆ Drag the control into the program form.

Visual Basic .NET

The source code in Visual Basic .NET is for example only and it will not be improved or maintained.

To use MSComm perform the following steps from the main window of the compiler:

- ◆ File Windows/system32/MSCOMM32.OCX must exist.
- ◆ On "Tools" menu, select "Customize Toolbox"
- ◆ Check the checkbox next to "Microsoft Comm Control 6.0". The control will appear in the toolbox.
- ◆ Drag the control into the program form.

Visual C++ .NET 2003

To use MSComm perform the following steps from the main window of the compiler:

- ◆ File Windows/system32/MSCOMM32.OCX must exist.
- ◆ On “Tools” menu, select “Add/Remove toolbox items”.
- ◆ Select tab “Com Components”.
- ◆ Select component “Microsoft Communication control version 6.0” corresponding to the file MSCOMM32.OCX. The control will appear in the Toolbox.
- ◆ Drag the control in the program form

The source code in **Visual C++.NET** uses the 2003 version of the language. The main file is **hyterm_vc.sln**, which is the solution file to be loaded by the compiler.

The code is contained in file: **form1.h**

All other files are automatically generated by the compiler.

Visual C++. NET programs can be written in managed or unmanaged coding style.

This particular program is written in ‘managed code’ style; therefore, its syntax may look unfamiliar to the classic C++ programmer (the classic C++ code corresponds to what Microsoft calls “unmanaged code” in Visual C++. NET).

How to use the GUI

The program displays a GUI (Graphic User Interface) with several buttons. Pressing a button the following action will be taken:

- **Exit.** This button exits the application. It should always be used to exit, as it turns off the serial port before exiting.
- **Reset.** It generates the Roboteq code **%rrrrrr** which resets the controller.
- **RC.** It generates the Roboteq codes **TCR - ^00 00 - %rrrrrr** which put the controller in ICM=0 (Input Control Mode 0) or Radio Control mode. (Note: **TCR** = Ten Carriage Returns).
- **RS232 no watchdog.** It generates the Roboteq codes **TCR - ^00 01 - %rrrrrr** which put the controller in ICM=1 or serial port mode and watchdog disabled.
- **RS2323 watchdog.** It generates the Roboteq codes **TCR - ^00 02 - %rrrrrr** which put the controller in ICM=2 or serial port mode and watchdog enabled.
- **Analog.** It generates the Roboteq code **^00 03** which puts the controller in ICM=3 or joystick mode.
- **TX.** Pushing it will transmit any command previously typed in its textbox (the program adds the required carriage return at the end). It will also show any command transmitted to the controller which is generated pushing any command button.
- **RX.** Pushing it will show any message waiting in the receive buffer. Activating the timer, the receive buffer will be sampled at regular time intervals. In particular allows to observe the character **W** issued by the watchdog, and the log string issued by the controller in ICM 0 and 3.

- **Ten Carriage Returns.** Generates a sequence of ten carriage returns **TCR** , which is the Roboteq code that forces the controller to listen, regardless of its ICM value.
- **Query buttons Power, Amps, Analog ins, Temperature, Batteries.** They generate the query codes **?X** where X is the letter corresponding to each query. The controller must be in ICM 1 or 2 to be able to reply. Push the **RX** button to visualize the reply.