RAYMON terminal emulation usage

1. Introduction:

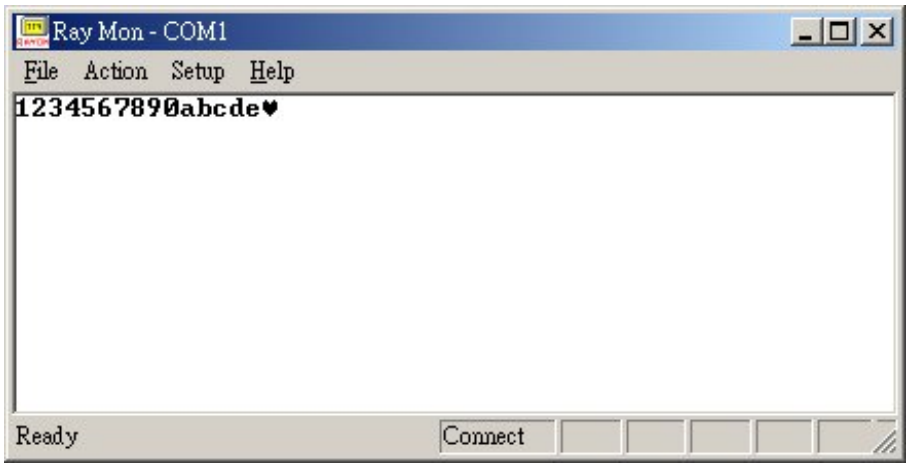
RAYON Technology Co., Ltd. is major in multi-serial port system solution. We always have one question from user. How to know the serial port driver is installed successfully? How to know the COM port can work correctly? Generally we will ask user to use "**Hyperterm**" to test in Windows system. But we may have some Windows system without "**Hyperterm**" available.

Now we offer "**RAYMON**" terminal emulation program for user to test. We offer execution file and source file of "RAYMON". User can use "RAYMON" to test COM port function. User can modify our source file to meet their target application environment.

Following information will describe the function and usage for "RAYMON".

2. Usage:

When we start to run "RAYMON" utility, we can assign one COM port to send and receive data.



In upper line of Window we can set parameter for COM port and start to work.

In "**File**" field we can have three functions to choose.

a) **Send File**: We can use this function to send one file to COM port.

b) **Log File**: We can use this function to save data from COM port to one target file name. When we specify the file name to save, we will start to save data from COM port. We can click "OK" field to stop this process. In current version we will display the data received byte count in "Terminal Mode" only. For other mode we do not display the data received byte count. But we had saved the data from COM port to file.

c) **Exit**: We can use this function to close this utility.

In "**Action**" filed we can have following function to run.

a) **Connect**: We must use this function to start COM port operation. When we need to modify some parameter for COM port, we need to use this function to start COM port operation again.

b) **Disconnect**: We must use this function to stop COM port operation. When we need to modify some parameter for COM port, we need to use this function to stop COM port operation. Or we can not modify any parameter for COM port.

c) **Active RTS**: We can use this function to enable RTS signal in active state.

d) **Clear RTS**: We can use this function to disable RTS signal to inactive state.

e) **Active DTR**: We can use this function to enable DTR signal in active state.

f) **Clear DTR**: We can use this function to disable DTR signal to inactive state.

In "**Setup**" field we can setup parameter for COM port and terminal mode.

a) **PORT** : we can assign target COM port number for this utility.

b) **BAUD RATE** : we can assign target baud rate for this COM port.

c) **DATA BITS** : we can assign data bits protocol for this COM port.

d) **PARITY** : we can assign parity protocol for this COM port.

e) **Flow Control** : We can assign flow control method for this COM port.

f) **Terminal Mode** : We will use this utility in terminal display mode. Each line will display 80 characters. We will process some control code and control sequence for dedicated function.

g) **Monitor Mode** : We will use this utility in Monitor display mode. Each line will display 80 characters. We do not process any control code and control sequence for dedicated function. Every data received will be displayed as displayable character in Window display.

h) **HEX Mode** : We will use this utility in Monitor display mode. But we will display HEX code also. Each line can display 16 characters. The left side is displayable character to show. The right side is HEX code to show. It is due to some control code is not easy to read. So we need to show HEX code to read easily.

i) **Auto Wrap** : When we receive data in 80 character location, we may keep new coming data in 80 character location or first character of next line. To select this function we will display in first character of next line.

j) **New Line** : When we receive CR (0x0d) code, we may move current cursor to first character location of current line or next line. To select this function we will move current cursor to first character location of next line.

k) **Local Echo** : When we strike any key in keyboard, we will send to COM port only or local display also. To select this function we will send data to local display also.

When we setup above feature we need to strike "**OK**" to save or "**EXIT**" to skip. But we must be in "Disconnect" condition to save, or we will show Error message to ask you in "Disconnect" condition to save.

In bottom line of Window display we can see one status bar to show.

a) **Connect** or **Disconnect** status.

b) **DTR** and **RTS** condition. But it will be shown after "Active RTS" "Active DTR" "Clear RTS" "Clear DTR" action.

c) **DCD**, **CTS** and **DSR** condition. It will be shown as COM port input condition upon state changed.

3. Application:

a) RAYMON operation procedure:

After run "RAYMON" we need to enter "**Setup**" field to assign target COM port number, target protocol and target display mode to use.

Then we just need to enter "**Active**" field to start "Connect".

b) **Terminal Mode** application:

Basically it is just like a typewriter to use. You can strike any key in keyboard to send in COM port. You can display any data from COM port.

Each screen we can have 25 lines to display and each line can display 80 characters. We prepare one buffer to save 4 screens. When the data input is over 4 screens, we will remove the first line of data in buffer and display data in last line of buffer (first in first out procedure).

Generally we will display all data received from COM port except following control code and control sequence.

1) CR (0x0d) : move current cursor to first character location of current line or next line.

2) LF (0x0a) : move current cursor to next line.

3) BS (0x08) : move current cursor to previous character location (left one).

4) Esc [A (0x1b 0x5b 0x41) : This control sequence will move current cursor location to previous line (move up).

5) Esc [B (0x1b 0x5b 0x42) : This control sequence will move current cursor location to next line (move down).

6) Esc [C (0x1b 0x5b 0x43) : This control sequence will move current cursor location to next character (move right).

7) Esc [D (0x1b 0x5b 0x44) : This control sequence will move current cursor location to previous character (move left).

So we can use "Terminal mode" as normal console application.

c) **Monitor Mode** application:

Basically it is just like a recorder to use. You can strike any key in keyboard to send in COM port. You can display any data from COM port.

We do not process any control code. In Terminal mode we may process control code and not displayed in screen. So it is not easy to know the real condition for data received. In Monitor mode every data received in COM port will be displayed in screen. So it is easy to know the data received in COM port.

d) **HEX Mode** application:

In Monitor mode we will display any data received in COM port. But we may have different character to display for one code in different Windows environment. So it is not easy for us to know the real received code for one character displayed.

In HEX mode we will display normal character in left side of one line and HEX code in right side of one line. So it is very easy for us to know the binary code received in COM port. Then it is very easy for us to check the data received in dedicated protocol.

e) RTS and DTR setup:

Even though we may need to control the state for DTR and RTS signal. But DTR and RTS state may be modified by some function. So we can use "Active RTS" "Active DTR" "Clear RTS" "Clear DTR" to set the target state of DTR and RTS signal. But we can't confirm the state will always keep in such state.

4. Conclusion:

RAYMON is one simple and flexible utility. Our source file for "RAYMON" will be good reference for COM port application software programmer. In one Windows system we can run "RAYMON" utility anytime. Each "RAYMON" is one terminal emulation window for one target COM port. We can run multiple "RAYMON" to test data transmission between multiple COM ports.

If user needed to know the data transmission condition between two COM ports, then our "**RAYREAL**" utility is good solution. In some "**Poll & Ack**" software environment it is very important to know the timing for data to send "**Poll**" and "**Ack**" received data. If you used "**RAYREAL**" utility with our **iLOG101** box or **ULOG485** box, then you can monitor your "**Poll & Ack**" software environment via IP network or USB connection.