

**PT630**

Application Note



Document number: 3506000180

# TABLE OF CONTENTS

Chapter 1	Introduction.....	1
1.1	JobGen Plus.....	1
1.2	Why C .....	1
1.3	Contents of the Demo Disk.....	2
Chapter 2	Notes to Develop Terminal Application.....	3
2.1	Programming Tips.....	3
2.2	Typical Program Flow .....	4
Chapter 3	C Function Library for PT630 programming.....	5
3.1	Display Font Functions .....	5
3.1.1	Select Large Font.....	5
3.1.2	Get Font Type .....	5
3.1.3	Set User-Defined Font for All Characters.....	5
3.1.4	Get Font Data for All Characters.....	5
3.1.5	Set User-Defined Font for One Character.....	5
3.1.6	Get Font Data for One Characters.....	5
3.2	Kermit Server Function.....	5
3.3	LCD Control Functions.....	6
3.3.1	Clear LCD Screen .....	6
3.3.2	Enable/Disable Screen Scroll.....	6
3.3.3	Set Cursor Position.....	6
3.3.4	Get Cursor Position.....	6
3.3.5	Display 16x16 Bitmap .....	6
3.3.6	Set Cursor On/Off.....	6
3.3.7	Display 16*16 Bitmap at Current Cursor Position .....	6
3.4	System Functions .....	6
3.4.1	Terminate Program.....	6
3.4.2	Read Keypad (wait if no key) and Write to LCD.....	6
3.4.3	Write LCD.....	6
3.4.4	Read RS232 (wait if no character).....	7
3.4.5	Write RS232.....	7
3.4.6	Direct Console I/O.....	7
3.4.7	Read Keypad (wait if no key).....	7
3.4.8	Write Character String to LCD.....	7
3.4.9	Buffered Keypad Input .....	7
3.4.10	Check Keypad Status.....	7
3.4.11	LCD Backlight ON/OFF Control.....	7
3.4.12	Buzzer ON/OFF Control.....	7
3.4.13	Beeper Volume .....	7
3.4.14	Enable/Disable RS232 Port.....	8
3.4.15	Enable/Disable Key or Key Function.....	8
3.4.16	Set Keypad Language.....	8
3.4.17	Check Main Battery Status.....	8
3.4.18	Check Backup Battery Status.....	8
3.4.19	Set Good-read LED .....	8
3.4.20	Set Laser Scanner Trigger Mode.....	8
3.4.21	Enable/Disable Double Verification When Read Bar Code Label .....	9
3.4.22	Check Laser Scanner .....	9
3.4.23	Get Scanner Port Status .....	9

3.4.24	Get MULTIPOINT Address .....	9
3.4.25	Set Communication Parameters .....	9
3.4.26	Control Flow .....	9
3.4.27	Set Communication Protocol .....	9
3.4.28	Get MULTIPOINT Address .....	9
3.4.29	Set MULTIPOINT Address .....	10
3.4.30	Set File-Uploading Message ON/OFF .....	10
3.4.31	Set Terminal ID .....	10
3.4.32	Set ONLINE/LOCAL in Terminal Mode .....	10
3.4.33	Set ECHO ON/OFF in Terminal Mode .....	10
3.4.34	Set AUTOLF ON/OFF in Terminal Mode .....	10
3.4.35	Set CHARACTER/BLOCK in Terminal Mode .....	10
3.4.36	Define LINE Character in Terminal Mode .....	10
3.4.37	Define Page Character in Terminal Mode .....	10
3.4.38	Set User-defined Key-map .....	11
3.4.39	Get System Key-map .....	11
3.4.40	Enable/Disable Decoding of a Barcode Symbology .....	11
3.4.41	Get Decoding Status of a Barcode Symbology .....	11
3.4.42	BarCode symbology Settings .....	11
3.4.43	Set Interrupt Vector .....	13
3.4.44	Get System Date .....	13
3.4.45	Set System Date .....	13
3.4.46	Get System Time .....	13
3.4.47	Set System Time .....	13
3.4.48	Set Alarm Date .....	13
3.4.49	Set Alarm Time .....	13
3.4.50	Get DOS and Firmware Version Number .....	14
3.4.51	Get Interrupt Vector .....	14
3.4.52	Get Free Disk Cluster .....	14
3.4.53	Create File with Handle .....	14
3.4.54	Open File with Handle .....	14
3.4.55	Close File with Handle .....	14
3.4.56	Read File with Handle .....	14
3.4.57	Write File with Handle .....	15
3.4.58	Delete File .....	15
3.4.59	Move File Pointer .....	15
3.4.60	Allocate Memory .....	15
3.4.61	Free Allocated Memory .....	15
3.4.62	End Program .....	15
3.4.63	Read Data from Scanner Port .....	15
3.4.64	Set Scanner Port .....	15
3.4.65	Rename File .....	16
3.4.66	Create New File .....	16
3.4.67	Receive Data from RS232 Port in MULTIPOINT Protocol .....	16
3.4.68	Send Data to RS232 Port in MULTIPOINT Protocol (Buffered Output) .....	16
3.4.69	Check RS232 Output Buffer Status in MULTIPOINT Protocol .....	16
3.4.70	Power Management Function .....	16
3.6	Beeper Control Function .....	16
3.7	RS232 Control Functions ( INT 33H ) .....	17
3.7.1	Set Communication Parameters .....	17
3.7.2	Get Character from RS232 Port .....	17
3.7.3	Send Character to RS232 Port .....	17

3.7.4	Enable RS-232 Port .....	17
3.7.5	Disable RS-232 Port .....	17
3.7.6	Set RTS/DTR Signal of RS232 Port .....	17
3.7.7	7. Get CTS Signal Status of RS232 Port .....	18
Chapter 4	DLL function call for VC++ .....	19
Chapter 5	32 Bits DLL for Visual Basic .....	36
Chapter 6	32 Bits DLL for Delphi .....	59

## Chapter 1 Introduction

This Application Note Guide demonstrates some application examples that will help users to develop quickly and efficiently for their applications. Since the system of PT630 is an open architecture, you are able to write a program run on PT630 to fully control its input/output facilities or have a complicate accumulation, data validation or table look up features for your application program. The PT630 provides an MS-DOS compatible platform that allows the user to write his application program by the following programming tools.

- JobGen Plus, a Windows base program generator that requiring the minimum programming skills. Please contact your local distributor or dealer for more detailed information.
- Microsoft C version 5.x, 6.x, 7.x or 8.x for MSDOS application
- Microsoft Visual C/C++ V1.xx for MSDOS application
- Borland C or Turbo C for MSDOS application
- Macro Assembler for MSDOS application

### 1.1 JobGen Plus

JOB GENERator Plus (JobGen Plus) is a Windows-based program generator that gives the user an ease-of-use and comprehensive terminal application developing environment. Through the use of JobGen Plus, user can design an application program for a data collection terminal by simply drawing the data collection sequence on paper, and thus, requires minimum programming skills. The Communication Manager in JobGen Plus, enables users to upload and download files between a PC and a data collection terminal. JobGen Plus currently supports the PT630.

The main features of JobGen Plus are listed below:

- Handy design interface .
- Define system configuration of the PT630.
- Define data fields and process flow.
- Download a Jobgen Plus application to the PT630 with lookup data files.
- Upload the collected data files from PT630 to PC.
- Print specification of a Jobgen Plus application for documentation.
- Building C compiler to interface with C language.
- Application simulation on PC.

### 1.2 Why C

If you are familiar with Microsoft C or other equivalent C programming language, you may adopt one of them as your programming tool to develop your application program and download it into the PT630. Or if the JobGen Plus can not meet your requirement, the C language is recommended to be adopted, due to the size of its executable object program is compact. The executable file generated by C compiler can be downloaded to and run on PT630.

- 1) The DOS-compatible system of PT630 supports part of MS/DOS system and uses different BIOS calls for its hardware driver. Some functions in the run-time library of your C programming tool kit

are not supported by PT630. Please refer to *PT630 Programming Reference Manual* for more detailed information to write C program to comply PT630's system.

- 2) Select the small or large memory model to compile your program, according to its memory requirement. Using parameter /AS or /AL or /AH to compile your program, according to its memory requirement. Remember that the available memory of your PT630 terminal is limited.

## 1.3 Contains of the Demo Disk

### *PT630 function library C*

- 1) TrmLIB.C/OBJ/H C library source/object/header files PT630
- 2) Trmtest.C/EXE Test program and source code for using C library

### *Communication Utility for Windows*

- 1) 32BIT\dll32 32 bits DLL library for MS-Windows 95/NT
- 32BIT\multi32 32bits Communication Utility for MS-Windows 95/NT
- 32BIT\sample VC++ (v5.0) Sample program for using 32bits DLL
- 3) \Delphi Delphi Sample program for using 32bits DLL
- 4) \VB Visual Basic Sample program for using 32bits DLL

## Chapter 2 Notes to Develop Terminal Application

Following notes will help you to develop application programs more efficiently and reliably.

### 2.1 Programming Tips

#### Program structure

- 1) Make sure your program size as compact as it can; remove the features that may be not necessary or required; analyze and modularize your application.
- 2) The PT630 uses one processor to handle all input/output devices and decoding service. The PT630 can only serve one input or output or decoding signal event at a time, that means, the system will disable other facilities until complete the service. Therefore, to prevent from occupying the processor too long to interfere with other devices' activities. It is recommended to perform the I/O service only when they are required. For instance, the prompting message shown on terminal's LCD display, should not keep rewriting the message while the system is expecting for data or other events occurred from keyboard or communication port or scanner port.

#### Communication environment

- 1) Decide which communication protocol to be adopted, multi-point, or none protocol. Generally, if the application will be used in a network communication environment, you should adopt the multi-point protocol. If the application will be used in a point-to-point communication environment, you may use built-in Kermit communication server to upload/download file to/from terminal. If the terminal is attached with a serial printer or other serial device rather than host computer/PC, you may select none protocol mode and simply use XON/XOFF to handshake with the external serial device.
- 2) If you plan to install the terminal in a network communication environment. You should prepare a terminal address management table to decide which address ID will be designated to the terminal and its corresponded downloaded application program and data files. In order to download/upload and poll data to/from the connected terminals.

#### Compatibility

- 1) The PT630 doesn't support PC BIOS function call, you should not use them in your program.

#### Choosing the development tool

- 1) Choose the JobGen Plus to generate the terminal application without programming effort or developing the application by use of PC programming language. It is depended on the processing complication of the program. If the purpose of the program is simply collecting data from input device without too complex accumulation or file accessing, the JobGen Plus is recommended to be adopted as your development tool, such as Inventory data collection program on the terminal.
- 2) Programs compiled by Microsoft C, Borland C and Microsoft Assembly (8088 instruction set) can be executed on the PT630.

## Optimize the memory

- 1) Decide the terminal's memory size for retaining the downloaded program and data file, and for executable program area.

## Optimize decoding performance

- 1) To make the system to decode the barcode symbologies in the most efficient performance, it is better to only enable those barcode symbologies will be used in your application. And disable the rest of barcode symbologies.

## 2.2 Typical Program Flow

Following instructions is a typical program flow for data collection application, you may refer the steps below to organize your program prior to writing it.

- STEP 1. Open/create the necessary files for read/write, depended on the usage purpose.
- STEP 2. Secure your application, recommended.
- STEP 3. Change the keypad mapping table, if you have your own keypad layout or definition.
- STEP 4. Setup communication parameters.
- STEP 5. Enable the communication port and designate the protocol used.
- STEP 6. Enable the devices which will be used for data collection or signal controlling.
- STEP 7. Disable the barcode symbologies which will not be used. (the system default is all enabled)
- STEP 8. Prompt the message on terminal LCD display.
- STEP 9. Execute INT22 to force PT630 go to Standby mode for saving PT630's battery power, it will be waked up to Active mode when key-pressing, scanning or receiving data from R232.
- STEP 10. Scanning the available devices for data input or signal occurred.
- STEP 11. Processing and validating the data or signal.
- STEP 12. Store input data into RAM disk.
- STEP 13. Go back to step 8 for next processing, or exit program.
- STEP 14. Restore the system default keypad mapping table, if you have changed it in step 3.
- STEP 15. Exit program.

## Chapter 3 C Function Library for PT630 programming

### 3.1 Display Font Functions

#### 3.1.1 Select Large Font

void TL\_font(int status)

Status : 0 - Large font -- 4 lines \* 16 columns display

1 - Small font -- 8 lines \* 20 columns display

#### 3.1.2 Get Font Type

int TL\_get\_font\_type()

return : 0 - Large font

1 - Small font

#### 3.1.3 Set User-Defined Font for All Characters

void TL\_change\_all\_ASCII\_font(int type,unsigned char \*str)

type : 0 -- Large

1 -- Small

str : font buffer

#### 3.1.4 Get Font Data for All Characters

void TL\_get\_all\_ASCII\_font(int type,unsigned char \*str)

type : 0 -- Large

1 -- Small

str : font buffer

#### 3.1.5 Set User-Defined Font for One Character

void TL\_change\_one\_ASCII\_font(int type,int ascii\_code,unsigned char \*str)

type : 0 -- Large

1 -- Small

ascii\_code : ASCII code

str : font buffer

#### 3.1.6 Get Font Data for One Characters

void TL\_get\_one\_ASCII\_font(int type,int ascii\_code,unsigned char \*str)

type : 0 -- Large

1 -- Small

ascii\_code : ASCII code

str : font buffer

### 3.2 Kermit Server Function

void TC\_kermit\_mode()

## 3.3 LCD Control Functions

### 3.3.1 Clear LCD Screen

void TL\_clrscr()

### 3.3.2 Enable/Disable Screen Scroll

void TL\_scroll(int status)

status : 0 -- Disable

1 -- Enable

### 3.3.3 Set Cursor Position

void TL\_gotoxy(int x,int y)

x : Row

y : Column

### 3.3.4 Get Cursor Position

void TL\_getxy(int \*x,int \*y)

x : Row

y : Column

### 3.3.5 Display 16x16 Bitmap

void TL\_display\_16x16\_location(int x,int y,unsigned char \*str)

X : Row

Y : Column

Str : bitmap buffer

### 3.3.6 Set Cursor On/Off

void TL\_cursor\_type(int status)

status : 0 -- off

1 -- on

### 3.3.7 Display 16\*16 Bitmap at Current Cursor Position

void TL\_display\_16x16(unsigned char \*str)

str : bitmap buffer

## 3.4 System Functions

### 3.4.1 Terminate Program

void TS\_exit\_program()

### 3.4.2 Read Keypad (wait if no key) and Write to LCD

unsigned char TS\_stdin()

return : ASCII character

### 3.4.3 Write LCD

void TS\_stdout(unsigned char ch)

ch : Output ASCII character

#### **3.4.4 Read RS232 (wait if no character)**

unsigned char TS\_stdaux\_in()

return : ASCII character

#### **3.4.5 Write RS232**

void TS\_stdaux\_out(unsigned char ch)

ch : Output ASCII character

#### **3.4.6 Direct Console I/O**

unsigned char TS\_stdin\_out(unsigned char ch)

ch :     0 – 254         ; write ASCII character in DL to LCD  
       255         ; read ASCII character from keypad

#### **3.4.7 Read Keypad (wait if no key)**

unsigned char TS\_stdin\_noecho()

Return : 0 – 255 ASCII character

#### **3.4.8 Write Character String to LCD**

void TS\_stdout\_string(unsigned char \*str)

str : pointer to string buffer

#### **3.4.9 Buffered Keypad Input**

void TS\_stdin\_string(unsigned char \*str)

str : string buffer

#### **3.4.10 Check Keypad Status**

unsigned char TS\_kbhit()

return : 0 -- no keys were pressed

       -1 -- key was pressed and input character is Ready

#### **3.4.11 LCD Backlight ON/OFF Control**

void TL\_backlight(int status)

status : 0 -- OFF

       1 -- ON

#### **3.4.12 Buzzer ON/OFF Control**

void TD\_buzzer(int status)

status : 0 -- OFF

       1 -- ON

#### **3.4.13 Beeper Volume**

void TD\_beeper\_vol(int status)

status : 0 -- low

       1 -- medium

       2 -- high

### 3.4.14 Enable/Disable RS232 Port

void TD\_serial(int status)

status : 0 -- disable

1 -- enable

### 3.4.15 Enable/Disable Key or Key Function

void TD\_keylock(int type,int status)

type : 0 -- All keys

1 -- Supervisor mode

2 -- Cold start

3 -- Warm start

4 -- User mode (press [CMD] for 2 seconds)

5 -- ALPHA key

status 0 -- Disable

1 -- Enable

### 3.4.16 Set Keypad Language

void TD\_key\_language(int status)

status : 0 -- English

1 -- Sweden / Finland

2 -- Danish

3 -- Spanish

4 -- French

5 -- German

6 -- Italian

### 3.4.17 Check Main Battery Status

int TS\_battery()

return : 0 -- Normal

1 -- Battery low

### 3.4.18 Check Backup Battery Status

int TS\_lithium\_battery()

return : 0 -- Normal

1 -- Battery low

### 3.4.19 Set Good-read LED

void TS\_bar\_good\_read(int status)

status : 0 -- Set the Good-read LED (green light) OFF

1 -- Set the Good-read LED (green light) ON

2 -- Set the Good-read LED controlled by system

### 3.4.20 Set Laser Scanner Trigger Mode

void TD\_flash\_trigger(int status)

return : 0 -- Normal

1 -- Flash

### 3.4.21 Enable/Disable Double Verification When Read Bar Code Label

```
void TD_double_verify_bar(int type)
type : 0 -- Disable
       1 -- Enable double verification
```

### 3.4.22 Check Laser Scanner

```
int TD_check_scanner()
return : 0 -- Has no built-in or clip on laser scanner
        1 -- Has built-in or clip on laser scanner
```

### 3.4.23 Get Scanner Port Status

```
int TD_get_bar_type()
return : 0 -- Scanner port is disabled
        1 -- Scanner port is enabled
```

### 3.4.24 Get MULTIPOINT Address

```
char TC_get_address()
Return : Address -- 'A' - 'Y' or '0' - '6'
```

### 3.4.25 Set Communication Parameters

```
int TC_232_parameter(long baud,int parity,int stop,int data)
baud : 150 ~ 57600
parity: 0 -- None
        1 -- Odd
        2 -- Even
stop bit: 1 -- one stop bit
          2 -- two stop bit
data bit: 7 -- 7 data bits
          8 -- 8 data bits
```

### 3.4.26 Control Flow

```
void TC_flow_ctrl(int status)
status : 0 -- None
        1 -- XON/XOFF
        2 -- CTS/RTS
```

### 3.4.27 Set Communication Protocol

```
void TC_protocol(int status)
status : 2 -- Set to MULTIPOINT protocol
        3 -- Set to NONE protocol
```

### 3.4.28 Get MULTIPOINT Address

```
char TC_get_address()
return : Address -- 'A' - 'Y' or '0' - '6'
```

### 3.4.29 Set MULTIPOINT Address

int TC\_set\_address(char status)  
status : Address -- ASCII character 'A' - 'Y' or '0' - '6'  
return: 0 -- successful  
1 -- if failed

### 3.4.30 Set File-Uploading Message ON/OFF

void TD\_upload\_message(int status)  
status : 0 -- Not display message while uploading file  
1 -- Display message while uploading file

### 3.4.31 Set Terminal ID

void TT\_id(unsigned char \*str)  
str : pointer to buffer of 8 characters ID string

### 3.4.32 Set ONLINE/LOCAL in Terminal Mode

void TT\_online\_local(int status)  
status : 0 -- online  
1 -- LOCAL

### 3.4.33 Set ECHO ON/OFF in Terminal Mode

void TT\_echo(int status)  
status : 0 -- ON  
1 -- OFF

### 3.4.34 Set AUTOLF ON/OFF in Terminal Mode

void TT\_auto\_LF(int status)  
status : 0 -- on  
1 -- off

### 3.4.35 Set CHARACTER/BLOCK in Terminal Mode

void TT\_mode(int status,int status1)  
status : 0 -- CHARACTER mode  
1 -- BLOCK mode  
status1: 0 -- block = LINE  
1 -- block = PAGE  
2 -- block = LINE or PAGE

### 3.4.36 Define LINE Character in Terminal Mode

void TT\_line\_terminal(unsigned char status)  
status: 0 - 255 -- ASCII character

### 3.4.37 Define Page Character in Terminal Mode

void TT\_page\_terminal(unsigned char status)  
status: 0 - 255 -- ASCII character

**3.4.38 Set User-defined Key-map**

void TD\_key\_map(unsigned char \*str)

str : pointer to Key-map buffer with 5\*24 characters corresponding to keypad in 5 input modes

**3.4.39 Get System Key-map**

void TD\_get\_key\_map(unsigned char \*str)

str : pointer to 120 bytes buffer

**3.4.40 Enable/Disable Decoding of a Barcode Symbology**

void TD\_set\_decode\_status(int type,int status)

type : 0 -- All supported bar code symbologies

1 -- Code 39

2 -- I 2 of 5

3 -- Codabar

4 -- EAN/UPC

5 -- Code 128

6 -- EAN 128

status: 1 -- Enable decoding of barcode symbology

0 -- Disable decoding of barcode symbology

**3.4.41 Get Decoding Status of a Barcode Symbology**

int TD\_get\_decode\_bar(int type)

type : 0 -- All supported bar code symbologies

1 -- Code 39

2 -- I 2 of 5

3 -- Codabar

4 -- EAN/UPC

5 -- Code 128

6 -- EAN 128

return: 1 -- Enable decoding of barcode symbology

0 -- Disable decoding of barcode symbology

**3.4.42 BarCode symbology Settings**

void TD\_decoder\_setting(int decoder\_type,int setting)

type: 1 -- Code 39

2 -- I 2 of 5

3 -- Codabar

4 -- EAN/UPC

5 -- Code 128

6 -- EAN 128

setting: **Code 39**

bit0= 0/1 -- disable/enable Code 39 decoding

bit1= 0/1 -- disable/enable Check Digit verification

bit2= 0/1 -- no-send/send Check Digit

bit3= 0/1 -- no-send/send Start/Stop characters

bit4= 0/1 -- Full ASCII OFF/ON

**I 2 of 5**

bit0= 0/1 -- disable/enable Interleaved 2 of 5 decoding

bit1= 0/1 -- disable/enable Check Digit verification

bit2 = 0/1 -- no-send/send Check Digit

#### **Codabar Settings**

bit0= 0/1 -- disable/enable Codebar decoding

bit1= 0/1 -- disable/enable Check Digit verification

bit2= 0/1 -- no-send/send Check Digit

bit3= 0/1 -- no-send/send Start/Stop characters

#### **Code 128 Setting**

Bit0= 0/1 -- disable/enable Code 128 decoding

#### **EAN 128 Setting**

Bit0= 0/1 -- disable/enable EAN 128 decoding

#### **Code 93 Setting**

bit0= 0/1 -- disable/enable Code 93 decoding

#### **UPC-A Settings**

bit0= 0/1 -- disable/enable UPC-A decoding

bit2= 0/1 -- no-send/send Check Digit

bit3= 0/1 -- no-send/send Leading Digit

#### **UPC-E Settings**

bit0= 0/1 -- disable/enable UPC-E decoding

bit2= 0 1 -- no-send/send Check Digit

bit3= 0/1 -- no-send/send Leading Digit

bit4= 0/1 -- disable/enable Zero Expansion

#### **EAN-13 Settings**

bit0= 0/1 -- disable/enable EAN-13 decoding

bit2= 0/1 -- no-send/send Check Digit

bit3= 0/1 -- no-send/send Leading Digit

#### **EAN-8 Settings**

bit0= 0/1 -- disable/enable UPC-E decoding

bit2= 0/1 -- no-send/send Check Digit

#### **User Code 1 Setting (TRIOPTIC Code)**

bit0= 0/1 -- disable/enable User Code 1 decoding

#### **User Code 2 Setting (TOSHIBA code)**

bit0= 0/1 -- disable/enable User Code 2 decoding

### 3.4.43 Set Interrupt Vector

```
void TS_set_interrupt_vector(int vect,unsigned int ds,unsigned int dx)
```

vect: interrupt number

ds: Segment pointer

dx: offset pointer

### 3.4.44 Get System Date

```
void TS_get_date(int *year,int *month,int *day,int *week)
```

year : 1980 - 2079

month : 1 - 12

day: 1 - 31

weekday: 0 - 6

### 3.4.45 Set System Date

```
int TS_set_date(int year,int month,int day)
```

year : 1980 - 2079

month : 1 - 12

day: 1 - 31

### 3.4.46 Get System Time

```
void TS_get_time(int *hour,int *minute,int *second,int *mini_sec)
```

hour :0 - 23

minute :0 - 59

second :0 - 59

### 3.4.47 Set System Time

```
int TS_set_time(int hour,int minute,int second)
```

hour :0 - 23

minute :0 - 59

second :0 - 59

### 3.4.48 Set Alarm Date

```
int TS_alarm_date(int status,int year,int month,int day)
```

status: 0 -- disable alarm

1 -- enable alarm everyday

2 -- enable alarm by date

year : 1980 - 2079

month : 1 - 12

day: 1 - 31

### 3.4.49 Set Alarm Time

```
int TS_alarm_time(int hour,int minute,int second)
```

hour :0 - 23

minute :0 - 59

second :0 - 59

### 3.4.50 Get DOS and Firmware Version Number

```
int TS_version1(int *ver,int *firm,int *oem)
ver :    DOS version no
firm:    Firmware version no.
oem:    OEM firmware version code (=0 for standard version)
```

### 3.4.51 Get Interrupt Vector

```
void TS_get_interrupt_vector(int vect,unsigned int *es,unsigned int *bx)
vect:    interrupt number
ds:     Segment pointer
dx:     offset pointer
```

### 3.4.52 Get Free Disk Cluster

```
long TS_free_disk(int drive)
drive:    drive -- always 0
return:   disk size
```

### 3.4.53 Create File with Handle

```
int TS_create_file(char *fn)
fn:       File name
return:   -1 -- fail to open file
          1~ ? -- file handle
```

### 3.4.54 Open File with Handle

```
int TS_open_file(char *fn,int mode)
fn:       File name
mode:    0 -- read only
          1 -- write only
          2 -- read and write
return:   -1 -- fail to open file
          1~ ? -- file handle
```

### 3.4.55 Close File with Handle

```
void TS_close_file(int hdl)
hdl:     File handle
```

### 3.4.56 Read File with Handle

```
int TS_read_file(int hdl,int cnt,char *str)
hdl:     File handle
cnt:     Byte counter
str:     string buffer
```

### 3.4.57 Write File with Handle

```
int TS_write_file(int hdl,int cnt,char *str)
hdl:    File handle
cnt:    Byte counter
str:    string buffer
```

### 3.4.58 Delete File

```
int TS_delete_file(char *fn)
fn:     File name
return: 1 -- OK
        -1-- fail
```

### 3.4.59 Move File Pointer

```
long TS_seek_file(int hdl,int type,long loc)
hdl:    File handle
type:   0 -- from beginning
        1 -- from current
        2 -- from end
long:   -1 -- seek error
        0 ~ ? -- file position
```

### 3.4.60 Allocate Memory

```
int TS_alloc_mem(unsigned int size,unsigned char far *str,int *free)
size:   segment address of block allocated
str:    memory pointer
free:   largest available memory block in paragraphs
```

### 3.4.61 Free Allocated Memory

```
int TS_free_mem(unsigned char far *str)
str:    memory pointer
return: 1 -- OK
        -1 -- fail to free memory
```

### 3.4.62 End Program

```
void TS_end_program(int ret_code)
return: program-defined return value
```

### 3.4.63 Read Data from Scanner Port

```
int TD_get_bar1(unsigned char *str,int wait,int *type,int *dir)
str:    string buffer
wait:   1 -- waiting for input if there is no data
        0 -- not wait
dir:    0 -- scan direction from left to right
        1 -- scan direction from right to left
```

### 3.4.64 Set Scanner Port

```
void TD_set_bar(int status)
```

status: 0 -- disable scanner port  
1 -- enable scanner

### **3.4.65 Rename File**

int TS\_rename\_file(char \*inf,char far \*outf)  
inf: old file name  
outf: new file name  
return: -1 -- fail to rename  
other -- OK

### **3.4.66 Create New File**

int TS\_create\_new\_file(char \*fn)  
fn: file name  
return: -1 -- fail to create file  
0~? -- file handle

### **3.4.67 Receive Data from RS232 Port in MULTIPOINT Protocol**

int TC\_str\_I(unsigned char \*str,int wait)  
str: string buffer for input data  
wait: waiting until data input when there is no data  
return: 0 -- input successful  
1 -- no data

### **3.4.68 Send Data to RS232 Port in MULTIPOINT Protocol (Buffered Output)**

int TC\_str\_O(unsigned char \*str,int wait)  
str: string buffer  
wait: 1 -- waiting until buffer is no busy  
0 -- not wait  
return: 0 -- input successful  
1 -- no data

### **3.4.69 Check RS232 Output Buffer Status in MULTIPOINT Protocol**

int TC\_ready(int wait)  
wait: 1 -- wait for ready, if output buffer is not empty  
0 -- not wait  
return: 0 -- output buffer is empty  
1 -- output buffer has data

### **3.4.70 Power Management Function**

void TS\_power()

## **3.6 Beeper Control Function**

void TD\_beep\_new(int fz,int tm)  
fz: Frequency 0 -- 200 Hz  
1 -- 400 Hz

	2 -- 600	
	3 -- 800	
	4 -- 1K	
	5 -- 2K	
	6 -- 2.5K	
	7 -- 3K	
	8 -- 5K	
tm:Time Duration(ms)		0 -- 10
	1 -- 50	
	2 -- 100	
	3 -- 200	
	4 -- 500	
	5 -- 800	
	6 -- 1K	
	7 -- 1.5K	
	8 -- 2K	

## 3.7 RS232 Control Functions ( INT 33H )

### 3.7.1 Set Communication Parameters

int TC\_232\_parameter(long baud,int parity,int stop,int data)

baud : 150 ~ 57600

parity: 0 -- None

1 -- Odd

2 -- Even

stop bit: 1 -- one stop bit

2 -- two stop bit

data bit: 7 -- 7 data bits

8 -- 8 data bits

### 3.7.2 Get Character from RS232 Port

unsigned char TC\_232\_char\_I()

return: 0~254 -- character

255 -- No data

### 3.7.3 Send Character to RS232 Port

void TC\_232\_char\_O(unsigned char ch)

ch: output character 0~254

### 3.7.4 Enable RS-232 Port

void TC\_232\_enable()

### 3.7.5 Disable RS-232 Port

void TC\_232\_disable()

### 3.7.6 Set RTS/DTR Signal of RS232 Port

void TC\_232\_RTS(int rts)

rts: 0 -- enable

1 -- disable

void TC\_232\_DTR(int dtr)

dtr: 0 -- enable

1 -- disable

### **3.7.7 7. Get CTS Signal Status of RS232 Port**

int TC\_232\_CTS()

return: 0 -- High voltage on signal

1 -- Low voltage on signal

## Chapter 4 DLL function call for VC++

There are two directory in this disk and each directory have three sub-directory

Dir \16BIT

DLL16	16bit DLL , include and library
MULTICOM	multicom 16bit version
SAMPLE	VC sample program for DLL (VC v1.5)

Dir \32BIT

DLL32	32bit DLL , include and library
MULTICOM	multicom 32bit version
SAMPLE	VC sample program for DLL (VC 5.0)

- **BOOL HM\_OpenCommPort(UINT port, int nComBaud ,int nComDataBit, int nComParity, int nComStopBit)**

Function Description:

Open communication COM port

Parameter:

port	:COM port	: 1-n for COM1 - COMn
nComBaud	:Baud Rate	0-8 for 110,300,600,1200,2400,4800,9600,19200,38400
nComDataBit	:Byte size	: 4-8
nComStopBit	:Stop bit length	: 1/2
nComParity	:parity check	0-NO PARITY 1-ODD PARITY 2-EVEN PARITY

Returns : TRUE-COM port open successfully  
FALSE-COM port open failure

- **void HM\_CloseCommPort(UINT port)**

Function Description:

Close communication COM port

Parameter:

port	:COM port	1-n for COM1 - COMn
------	-----------	---------------------

- **void HM\_time\_delay(int time\_period)**

Function description

Delay specify time period

Parameter:

time\_period : the number of milliseconds that have elapsed

- **void HM\_set\_power\_saving\_status(BOOL status);**

function description

For the portable terminal (700/870) , user must execute power saving command before and after each ESC command. This function will set whether each ESC command do power saving setting before and after each command or not.

For example , if status is set TRUE , each ESC command will do follow step

1. Remote set power saving off, if OK do next step.  
If fail return 9 then. User can execute HM\_get\_power\_saving\_return() to get error code
2. If step 1 OK, do the ESC command you call
3. Remote set power saving on, User can execute HM\_get\_power\_saving\_return() to get return code of power saving on setting

If status is set False , each ESC command only execute step 2

(POLL command is not include , use must execute power saving command by himself).

Parameter

status : True / False

- **BOOL HM\_get\_power\_saving\_status();**

Function description

Get Power saving status

Parameter

no

Return

True/False

- **int HM\_get\_power\_saving\_return();**

Function description

When HM\_set\_power\_saving\_status(TRUE). System will keep the return code of Power saving on/off setting in one variable. Use can get this return code by this function

Return:

0,3,4,5,6 (Please return value table)

- **long HM\_get\_time\_out();**

Function description

Get current time-out time setting

Return:

mini-second (Default is 400L = 0.4 second)

- **void HM\_set\_time\_out(long delay\_time);**

Function description

Set current time-out period

Parameter

delay\_time :mini-second

- **void HM\_set\_ESC\_delay(int delay\_time)**

Function description

After each ESC command executed. MULTIDLL will delay this time period default time is 10 milliseconds for on-line terminal or 30 for protable terminal

Parameter:

time\_delay :the number of milliseconds to be delay

- **int HM\_get\_ESC\_delay()**

Function description

Get ESC delay time (milliseconds)

Return

the number of milliseconds to be delay

- **void HM\_set\_NAK\_delay(int delay\_time)**

Function description

When NAK response is got , system will delay this time period before do re-try. The default value is 100 for portable terminal or 30 for on-line

Parameter

time\_delay :the number of milliseconds to be delay

- **int HM\_get\_NAK\_delay()**

Function description

Get NAK delay time

Return:

the number of milliseconds to be delay

- **int HM\_warm\_start(UINT port, char address)**

Function description

Remote warm-start terminal

Parameter

port :COM port

1-n for COM1 - COMn

address :Terminal address range form 'A'-'Y', '0'-'6'

Returns

0,3,4,5 (Please return value table)

- **int HM\_autoboot(UINT port, char address, char FAR\* fn)**

Function description

Set file name that terminal will auto-execute after power on

Parameter

port :COM port

1-n for COM1 - COMn

address :Terminal address range form 'A'-'Y', '0'-'6'

fn :File name (without file extention)

Returns

0,3,4,5,6,0x30,0x31 (Please return value table)

- **int HM\_bar\_tab(UINT port, char address, char FAR\* str)**

Function description

Set terminal's bar code symbology

Parameter

port :COM port

1-n for COM1 - COMn

address :Terminal address range form 'A'-'Y', '0'-'6'

str :for PT870

byte 0 : Code 39 E: enable D: disable F: full

byte 1 : I 2 of 5 E: enable D: disable

byte 2 : Codar bar E: enable D: disable

byte 3 : EAN/UPC E: enable D: disable

byte 4 : Code 128 E: enable D: disable



byte 1 : Stop bit : 1/2  
 byte 2 : Bit size : 7/8  
 byte 3 : Parity : N-No Parity, O-Odd parity, E-Even parity  
 byte 4 : Protocol : F-No protocol, M-Multi protocol  
 byte 5 : Address : Terminal address range form 'A'-'Y', '0'-'6'  
 byte 6,7: Time out : 00-99  
 for MR320  
 byte 0-7 same with MR350  
 byte 8 : always F  
 for PT700/870/860  
 byte 0-3 same with MR350  
 byte 4 : Flow control : F-none, C-CTS/RTS, X-Xon/Xoff  
 byte 5 : Protocol : F-none, M-multi, N-Null, A-ACK/NAK  
 byte 6 : Address : Terminal address range form 'A'-'Y', '0'-'6'  
 byte 7-9 : always SPACE

Returns

0,2,3,4,5,6 (Please return value table)

- **int HM\_date\_time(UINT port, char address, char FAR\* str)**

Function description

Set terminal's date and time

Parameter

port :COM port  
           1-n for COM1 - COMn  
 address :Terminal address range form 'A'-'Y', '0'-'6'  
 str :byte 0-3 year ex. "1996"  
       byte 4-5 month ex. "02"  
       byte 6-7 day ex. "31"  
       byte 8-9 hour ex. "12"  
       byte 10-11 month ex. "59"  
       byte 12-13 second ex. "59"

Returns

0,2,3,4,5,6 (Please return value table)

- **int HM\_delete(UINT port, char address, char FAR\* file\_name)**

Function description

Delete terminal file

Parameter

port :COM port  
           1-n for COM1 - COMn  
 address :Terminal address range form 'A'-'Y', '0'-'6'  
 file\_name :File name

Returns

0,3,4,5,0x30,0x31

- **int HM\_dev\_tab(UINT port, char address, char FAR\* str)**

Function description

Set terminal's communication parameter

Parameter

```

port      :COM port
           1-n for COM1 - COMn
address   :Terminal address range form 'A'-'Y', '0'-'6'
str       :Please see terminal's menu
for MR350/360
    byte 0 : Scanner    : N-Enable F-Disable
           byte 1 : Badge      : B-Barcode M-Magnetic D-Disable
           byte 2 : BackLight  : N-On F-Off
           byte 3 : Buzzer     : N-On F-Off
           byte 4 : Keylock   : N-Unlock K-Lock P-Partial
           byte 5 : Volume    : 0-Low 5-Medium 9-High
for PT700/870/860
    byte 0 : Scanner    : N-Enable F-Disable
           byte 1 : BackLight  : N-On F-Off
           byte 2 : Buzzer     : N-On F-Off
           byte 4 : Volume    : 0-Low 5-Medium 9-High
           byte 5 : Serial    : N-On F-Off
           byte 6 : Auto-Off   : 0-9 minute
           byte 7 : Always space
for MR380
    byte 0 : External     : P-Pen A-Auto M-MSR D-Disable
           byte 1 : Internal  : B-Barcode M-Magnetic D-Disable
           byte 2 : BackLight  : N-On F-Off
           byte 3 : Buzzer     : N-On F-Off
           byte 4 : Keylock   : N-Unlock K-Lock P-Partial
           byte 5 : Volume    : 0-Low 5-Medium 9-High

```

Returns

0,2,3,4,5,6 (Please return value table)

- **int HM\_dir(UINT port, char address, char FAR\* str);**

Function description

Get terminal's directory

Parameter

```

port      :COM port
           1-n for COM1 - COMn
address   :Terminal address range form 'A'-'Y', '0'-'6'
str       :Buffer for directory

```

Returns

0,3,4,5,6 (Please return value table)

- **int HM\_download(UINT port, char address, char FAR\* sour\_name, char FAR\* dest\_name);**

Function description

Download specified file to terminal(This function will not exit until download complete . If user want to show bytes count or packet count , please use HM\_download1 and HM\_download2)

Parameter

```

port      :COM port
           1-n for COM1 - COMn
address   :Terminal address range form 'A'-'Y', '0'-'6'
sour_name :File name in PC to be download

```

dest\_name :Assign new file name in terminal

Returns

0,2,3,4,5,6,0x30 (Please return value table)

- **int HM\_download1(UINT port, char address, char FAR\* sour\_name, char FAR\* dest\_name, long FAR\* lTotalLength);**

Function description

Download specified file to terminal (just check file and send first packet). This function must combine with HM\_download2.

Parameter

port :COM port

1-n for COM1 - COMn

address :Terminal address range form 'A'-'Y', '0'-'6'

sour\_name :File name in PC to be download

dest\_name :Assign new file name in terminal

lTotalLength :size of specified file

Returns

0,2,3,4,5,6,0x30 (Please return value table)

Example:

```
long lTotalLength;
if (HM_download1(1,'A',"ABC.DAT","DEF.DAT",&lTotalLength) == 0)
{
    do {
        ret = HM_download2(&lByteCounter,&lPacketCounter);
        Invalidate(FALSE);
    } while (ret == 1);
}
```

- **int HM\_download2(long FAR\* lByteCount, long FAR\* lPacketCount)**

Function description

Download : Send one packet to terminal (must used with HM\_download1())

Parameter

lByteCount :Total bytes that had been sent to terminal

lPacketCount :Total packet that had been sent to terminal

Returns

0,1,2,3,4,5,6 (Please return value table)

- **int HM\_exec(UINT port, char address, char FAR\* file\_name)**

Function description

Remote execute program

Parameter

port :COM port

1-n for COM1 - COMn

address :Terminal address range form 'A'-'Y', '0'-'6'

file\_name :File name to be executed (without file extension)

Returns

0,3,4,5,6,0x30,0x31 (Please return value table)

- **int HM\_exec\_file(UINT port, char address, char FAR\* fn)**

## Function description

Check the program name that currently running on terminal

## Parameter

port :COM port  
1-n for COM1 - COMn  
address :Terminal address range form 'A'-'Y', '0'-'6'  
file\_name :store file name that is running on terminal

## Returns

0,3,4,5,6,0x35 (Please return value table)

- **int HM\_exe\_size(UINT port, char address, int size)**

## Function description

Change size of executing area

## Parameter

port :COM port  
1-n for COM1 - COMn  
address :Terminal address range form 'A'-'Y', '0'-'6'  
size :RAM size (KB)

## Returns

0,3,4,5,6,0x40,0x41 (Please return value table)

- **int HM\_keypad(UINT port, char address, int status);**

## Function description

Set keypad lock status

## Parameter

port :COM port  
1-n for COM1 - COMn  
address :Terminal address range form 'A'-'Y', '0'-'6'  
status :0-Lock 1-Unlock 2-Partial lock

## Returns

0,3,4,5,6,0x30,0x31 (Please return value table)

- **int HM\_password(UINT port, char address, char FAR\* str)**

## Function description

Remote change supervisor mode password

## Parameter

port :COM port  
1-n for COM1 - COMn  
address :Terminal address range form 'A'-'Y', '0'-'6'  
str :Password

## Returns

0,2,3,4,5,6 (Please return value table)

- **int HM\_power\_saving\_off(UINT port, char address)**

## Function description

Turn off terminal's power saving mode

## Parameter

port :COM port

1-n for COM1 - COMn  
 address :Terminal address range form 'A'-'Y', '0'-'6'  
 Returns  
 0,3,4,5 (Please return value table)

- **int HM\_power\_saving\_on(UINT port, char address)**

Function description

Turn on terminal's power saving mode

Parameter

port :COM port

1-n for COM1 - COMn  
 address :Terminal address range form 'A'-'Y', '0'-'6'

Returns

0,3,4,5 (Please return value table)

- **int HM\_poll(UINT port, char address, unsigned char FAR\* data\_buf, int FAR\* data\_len)**

Function description

Poll data from terminal

Parameter

port :COM port

1-n for COM1 - COMn  
 address :Terminal address range form 'A'-'Y', '0'-'6'  
 data\_buf :store polled data  
 data\_len :length of poll data

Returns

0,3,4,5,7 (Please return value table)

- **int HM\_ram\_size(UINT port, char address, int FAR\* t\_size, int FAR\* e\_size, int FAR\* f\_size, int FAR\* et\_size, int FAR\* ef\_size);**

Function description

Get terminal's memory size

Parameter

port :COM port

1-n for COM1 - COMn  
 address :Terminal address range form 'A'-'Y', '0'-'6'  
 t\_size :Total memory size  
 e\_size :Execution area size  
 f\_size :Free RAM disk size  
 et\_size :Total EMS RAM disk size (If EMS exist)  
 ef\_size :Free EMS RAM disk size (If EMS exist)

Returns

0,3,4,5,6 (Please return value table)

- **int HM\_send\_data(UINT port, char address, char FAR\* data\_buf)**

Function description

Send application data to terminal

Parameter

port :COM port

1-n for COM1 - COMn

address :Terminal address range form 'A'-'Y', '0'-'6'  
 data\_buf :Application data

Returns

0,3,4,5 (Please return value table)

● **int HM\_sendback(UINT port, char address, char FAR\* in\_buf, char FAR\* out\_buf)**

Function description

Send and receive check

Parameter

port :COM port  
           1-n for COM1 - COMn  
 address :Terminal address range form 'A'-'Y', '0'-'6'  
 in\_buf :data to be sent  
 out\_buf :received data

Returns

0,3,4,5 (Please return value table)

● **int HM\_set\_address(UINT port, char addr, char n\_addr);**

Function description

Change terminal address

Parameter

port :COM port  
           1-n for COM1 - COMn  
 address :Terminal address range form 'A'-'Y', '0'-'6'  
 n\_addr :new terminal address

Returns

0,2,3,4,5 (Please return value table)

● **int HM\_term\_tab(UINT port, char address, char FAR\* str)**

Function description

Set terminal's communication parameter

Parameter

port :COM port  
           1-n for COM1 - COMn  
 address :Terminal address range form 'A'-'Y', '0'-'6'  
 str :Please see terminal's menu  
           byte 0-7 : Terminal ID  
           byte 8 : On-line :R-Remote L-Local  
           byte 9 : Echo : N-On F-Off  
           byte 10 : AutoLF : N-On F-Off  
           byte 11 : Mode : C-Character mode B-Block mode  
           byte 12 : LinePage : L-Line P-Page B-Both  
           byte 13 : Line terminalor : ASCII  
           byte 14 : Page terminalor : ASCII

Returns

0,2,3,4,5,6 (Please return value table)

● **int HM\_upload(UINT port, char address, char FAR\* sour\_name, char FAR\* dest\_name)**

Function description



- **int HM\_upload\_mode(UINT port, char address)**  
Function description  
Check whether terminal have been upload or not  
Parameter  
port :COM port  
1-n for COM1 - COMn  
address :Terminal address range form 'A'-'Y', '0'-'6'  
Returns  
0,3,4,5,0x50 (Please return value table)
- **int HM\_ups(UINT port, char address)**  
Function description  
Send UPS off  
Parameter  
port :COM port  
1-n for COM1 - COMn  
address :Terminal address range form 'A'-'Y', '0'-'6'  
Returns  
0,3,4,5,6 (Please return value table)
- **int HM\_320\_field\_port(UINT port, char address, char FAR\* data\_buf)**  
Function description  
MR320 field port setting  
Parameter  
data\_buf :Please see MR320 menu  
Returns  
0,3,4,5,6 (Please return value table)
- **int HM\_320\_min\_len(UINT port, char address, char FAR\* data\_buf)**  
Function description  
MR320 field port setting  
Parameter  
data\_buf :Please see MR320 menu  
Returns  
0,2,3,4,5,6 (Please return value table)
- **int HM\_320\_record(UINT port, char address, char FAR\* data\_buf)**  
Function description  
MR320 field port setting  
Parameter  
data\_buf :Please see MR320 menu  
Returns  
0,2,3,4,5,6 (Please return value table)
- **int HM\_file\_exist(UINT port, char address, char FAR\* fn, long FAR\* len);**  
Function description  
Check whether specified file is exist in terminal  
Parameter  
port :COM port

address :Terminal address range form 'A'-'Y', '0'-'6'  
 f\_n :File name to be checked  
 len :File length

Returns

0,3,4,5,6,0x30 (Please return value table)

*Add new return code 0x36 -- When file is existed on flash ROM*

- **int HM\_run(UINT port, char address, char FAR\* data\_buf)**

Function description

Remote execute program (same with HM\_exec());M

Parameter

port :COM port

1-n for COM1 - COMn

address :Terminal address range form 'A'-'Y', '0'-'6'

file\_name : File name to be executed (without file extension), *if program name is append with "/ROM", it will run the program on Flash ROM*

Returns

0,3,4,5,6,0x30,0x31 (Please return value table)

*Add new return code 0x36 -- When file is existed on flash ROM*

- **int HM\_dir\_ROM(UINT port, char address, char \* str);**

Function description

Get terminal's directory from flash ROM

Parameter

port :COM port

1-n for COM1 - COMn

address :Terminal address range form 'A'-'Y', '0'-'6'

str :Buffer for directory

returns :0,3,4,5,6 (Please return value table)

9 for HM\_set\_power\_saving\_status(TRUE)

- **int HM\_cancel\_download(UINT port, char address);**

Function description

Cancel download

Parameter

port :COM port

1-n for COM1 - COMn

address :Terminal address range form 'A'-'Y', '0'-'6'

returns :0,3,4,5,6 (Please return value table)

9 for HM\_set\_power\_saving\_status(TRUE)

- **int HM\_cancel\_upload(UINT port, char address);**

Function description

Cancel upload

Parameter:

port :COM port

address : 1-n for COM1 - COMn  
 : Terminal address range form 'A'-'Y', '0'-'6'  
 returns : 0,3,4,5,6 (Please return value table)  
 9 for HM\_set\_power\_saving\_status(TRUE)

- **int HM\_flash\_size(UINT port, char address, int \*t\_size, int \*f\_size);**

Function description

Get memory size of flash ROM

Parameter

port : COM port

1-n for COM1 - COMn

address : Terminal address range form 'A'-'Y', '0'-'6'

t\_size : Total flash ROM size

f\_size : Free flash ROM disk size

returns : 0,3,4,5,6 (Please return value table)

9 for HM\_set\_power\_saving\_status(TRUE)

- **int HM\_firmware\_version(UINT port, char address, char \*str);**

Function description

Get terminal's firmware version

Parameter

port : COM port

1-n for COM1 - COMn

address : Terminal address range form 'A'-'Y', '0'-'6'

str : Terminal type and version no.

returns : 0,3,4,5,6 (Please return value table)

9 for HM\_set\_power\_saving\_status(TRUE)

- **unsigned char HC\_getch(UINT port);**

Function description

Get one byte (character) from COM port

Parameter [input]

**port** : **UINT** : Serial COM port

1-n for COM1 - COMn

Returns : **unsigned char**

ASCII 0 ~ 254

ASCII 255 = no character input

- **void HC\_putch(UINT port, unsigned char cc);**

Function description

Output one byte (character) to COM port

Parameter [input]

**port** : **UINT** : Serial COM port

1-n for COM1 - COMn

**cc** : **unsigned char** : output character

ASCII 0 ~ 254

Returns : No

- **int HC\_modem\_cmd(UINT port, unsigned char \*str, long wait);**

Function description

Send string (modem command) to COM port and then wait for response of modem. This command will automatically append CR to end of string.

Parameter : [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**str** : **char \*** :Output string

**wait** : **longint** : Maximum Waiting time (milliseconds)

Returns : **integer** : return code. (Please check table "Modem function return code")

- **int HC\_modem\_cmd\_mode(UINT port, long wait);**

Function description

Force modem return to command mode (send "+++")

Parameter : [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**wait** : **long** : Maximum Waiting time (milliseconds)

Returns : **integer** : return code. (Please check table "Modem function return code")

- **void HM\_set\_upload\_file\_save\_mode(int mode)**

Function Description:

Setup how to save file uploaded from terminal

Parameter:

**mode** : save mode (Default is 0)  
0 -- Search for an available file(not exist)  
1 -- Over-write target file  
2 -- Append to target file

- **int HM\_get\_upload\_file\_save\_mode()**

Function Description:

Get mode of how to save file uploaded from terminal

Parameter: no

Return : save mode (Default is 0)  
0 -- Search for an available file(not exist)  
1 -- Over-write target file  
2 -- Append to target file

- **int HM\_version()**

Function Description:

Get DLL version no.

Returns : Version No.

- **Communication Return code**

Return Code	Meaning

0	Successfully
0x01	Unsuccessful or successfully download/upload one packed
0x02	Illegal command
0x03	Host get NAK
0x04	Time out
0x05	Get unknown response
0x06	Return string BBC error
0x07	No data to be enquired
0x08	User press ESC to break
0x09	Set Power saving off fail
0x10	Power failure
0x11	Processor exception occurred
0x12	Reset by host
0x13	User reset
0x14	Application reset
0x15	Cold start
0x16	Warm start
0x20	Terminal in diagnostic state
0x21	Terminal already start
0x22	Invalid number of parameter
0x30	File doesn't exist
0x31	Illegal file name
0x32	File already exist ,upload file name to be changed
0x33	File already exist ,system can't change file name
0x34	File open error
0x35	No program is executed
0x40	RAM disk isn't empty
0x41	Out of memory size
0x50	Upload mode

- **Modem function return code**

<b>Return code</b>	<b>Meaning</b>
0	OK
1	CONNECT
2	RING
3	NO CARRIER
4	ERROR
6	NO DIALTONE
7	BUSY
8	NO ANSWER
24	DELAYED
32	BLACKLISTED
33	FAX
35	DATA
40	CARRIER

200	Timeout
-----	---------

## Chapter 5 32 Bits DLL for Visual Basic

### File description

#### Dir \VB

<b>MULTIDLL.DLL</b>	<b>DLL(This file is same with \32bit\dll32\multidll.dll)</b>
<b>MULTI.BAS</b>	<b>Communication Function declare module (Please insert this file to your application as Module)</b>

#### Dir \VB\Sample

<b>Project1.exe</b>	<b>Test program (VB 4.0)</b>
<b>Project1.MAK</b>	<b>Make file</b>
<b>FORM1.FRM</b>	<b>form1 source</b>
<b>Multi.bas</b>	<b>Declare module</b>

\* Please copy MULTIDLL.DLL to SYSTEM sub-directory of Windows. If user want to place this file to other directory, please modify declare of MULTI.BAS to full path.

For example user want to place it in c:\test

change

```
HM_set_time_out Lib "multidll" (ByVal elay_time As Long)
```

to

```
HM_set_time_out Lib "c:\test\multidll" (ByVal elay_time As Long)
```

- **Function HM\_OpenCommPort(port, baud , databit, parity , stopbit)**

Function Description:

Open communication COM port

Parameter [input]

**port : ByVal As long** :Serial COM port  
1-n for COM1 - COMn

**baud : ByVal As long** :Baud Rate  
0-9 for 110,300,600,1200,2400,4800,9600, 19200,38400,57600

**databit : ByVal As long** :Byte size  
4-8

**parity : ByVal As long** :parity check  
0 for NO PARITY  
1 for ODD PARITY  
2 for EVEN PARITY

**stopbit : ByVal As long** :Stop bit length  
1/2

Returns: **As Boolean**

TRUE for serial COM port open successfully

FALSE for serial COM port open failure

- **Sub HM\_CloseCommPort(port)**

Function Description:

Close communication COM port

Parameter: [input]

**port : ByVal As long** :Serial COM port  
1-n for COM1 - COMn

Return : No

- **Sub HM\_time\_delay(time\_period)**

Function description

Delay specify time period

Parameter: [input]

**time\_period : ByVal As long** : the number of milliseconds that have elapsed

- **Sub HM\_set\_power\_saving\_status(status)**

Function description:

For the portable terminal (700/870) , user must execute power saving command before and after each ESC command. This function will set whether each ESC command do power saving setting before and after each command or not.

For example , if status is set TRUE , each ESC command will do follow step

1. Remote set power saving off, if OK do next step.  
If fail return 9 then. User can execute HM\_get\_power\_saving\_return() to get error code
2. If step 1 OK, do the ESC command you call
3. Remote set power saving on, User can execute HM\_get\_power\_saving\_return() to get return code of power saving on setting  
If status is set False , each ESC command only execute step 2 (POLL command is not include , use must execute power saving command by himself).

Parameter [input]

**status :**        **ByVal As Boolean**  
True for remote setting power saving  
False for NOT remote setting power saving

- **Function HM\_get\_power\_saving\_status()**

Function description

Get Power saving status (please see HM\_set\_power\_saving\_status)

Parameter : no

Return : **As Boolean**  
True/False

- **Function HM\_get\_power\_saving\_return()**

Function description

When HM\_set\_power\_saving\_status(TRUE). System will keep the return code of remote power saving setting in one variable. Use can get this return code by this function

Parameter : No

Return: **As long**  
0,3,4,5,6 (Please see return value table)

- **Function HM\_get\_time\_out()**

Function description

Get current time-out time setting

Return: **As Long**  
mini-second (Default is 400 = 0.4 second)

- **Sub HM\_set\_time\_out(delay\_time)**

Function description

Set current time-out period

Parameter [input]

**delay\_time: ByVal As Long** :mini-second, Default is 400 = 0.4 second

- **Sub HM\_set\_ESC\_delay(delay\_time)**

Function description

After each ESC command executed. MULTI.DLL will delay this time period default time is 10 milliseconds for on-line terminal or 30 for portable terminal

Parameter: [input]

**time\_delay : ByVal As long** :the number of milliseconds to be delay

- **Function HM\_get\_ESC\_delay()**

Function description

Get ESC delay time (milliseconds)

Return : **As long** :the number of milliseconds to be delay

- **Sub HM\_set\_NAK\_delay(delay\_time)**

Function description

When NAK response is got ,system will delay this time period before do re-try. The default value is 100 for portable terminal or 30 for on-line

Parameter [input]

**time\_delay: ByVal As long** :the number of milliseconds to be delay

- **Function HM\_get\_NAK\_delay()**

Function description

Get NAK delay time

Return: **As long** :the number of milliseconds to be delay

- **Function HM\_warm\_start(ByVal port As long, Address)**

Function description

Remote warm-start terminal

Parameter [input]

**port :ByVal As long** :Serial COM port  
1-n for COM1 - COMn

**address :ByVal As Byte** :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

Returns : **As long**

0,3,4,5 (Please see return value table)

- **Function HM\_autoboot(port ,address , fn)**

Function description

Set file name that terminal will auto-execute after power on

Parameter [input]

**port :ByVal As long** :Serial COM port  
1-n for COM1 - COMn

**address :ByVal As Byte** :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

**fn : ByVal As String** :File name (without file extention)

Returns : **As long**

0,3,4,5,6,0x30,0x31 (Please see return value table)

- **Function HM\_bar\_tab(port , address, str)**

Function description

Set terminal's bar code symbology

Parameter [input]

**port :ByVal As long** :Serial COM port  
1-n for COM1 - COMn

**address :ByVal As Byte** :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

**str :ByVal As String** :for PT870

byte 0 : Code 39 E: enable D: disable F: full

byte 1 : I 2 of 5 E: enable D: disable

byte 2 : Codar bar E: enable D: disable

byte 3 : EAN/UPC E: enable D: disable

byte 4 : Code 128 E: enable D: disable

for MR350/360/380

byte 0 : Code 39 N: enable D: disable F: full

byte 1 : I 2 of 5 N: enable F: disable

byte 2 : Codar bar N: enable F: disable

byte 3 : EAN/UPC N: enable F: disable

byte 4 : Code 128 N: enable F: disable

for PT700

byte 0 : Code 39 E: enable D: disable F: full

byte 1 : I 2 of 5 E: enable D: disable

byte 2 : Codar bar E: enable D: disable

byte 3 : EAN/UPC	E: enable D: disable
byte 4 : Code 128	E: enable D: disable
byte 5 : EAN 128	E: enable D: disable
byte 6 : Code 93	E: enable D: disable
byte 7 : Trioptic	E: enable D: disable

Returns : **As long**

0,2,3,4,5,6 (Please see return value table)

- **Function HM\_buzzer(port , address , status)**

Function description

Set terminal's buzzer volume

Parameter [input]

**port** :ByVal As long :Serial COM port  
1-n for COM1 - COMn

**address** :ByVal As Byte :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

**status** :ByVal As long :  
0 for Low  
5 for medium  
9 for High

Returns : As long  
0,3,4,5,6 (Please see return value table)

- **Function HM\_cold\_start(port , address)**

Function description

Remote cold start terminal

Parameter [input]

**port** :ByVal As long :Serial COM port  
1-n for COM1 - COMn

**address** :ByVal As Byte :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

Returns : As long  
0,3,4,5 (Please see return value table)

- **Function HM\_comm\_tab(port ,address, str)**

Function description

Set terminal's communication parameter

Parameter [input]

**port** :ByVal As long :Serial COM port  
1-n for COM1 - COMn

**address** :ByVal As Byte :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

**str** **ByVal As String** :Please see terminal's menu  
for MR350/360/380

byte 0	: Baud rate	: 1-9 for 110,300,600,1200,2400...38400
byte 1	: Stop bit	: 1/2
byte 2	: Bit size	: 7/8
byte 3	: Parity	: N-No Parity, O-Odd parity, E-Even parity
byte 4	: Protocol	: F-No protocol, M-Multi protocol
byte 5	: Address	: Terminal address range form 'A'-'Y', '0'-'6'
byte 6,7	: Time out	: 00-99

for MR320  
byte 0-7 same with MR350

byte 8 : always F  
 for PT700/870/860  
 byte 0-3 same with MR350  
 byte 4 : Flow control : F-none, C-CTS/RTS, X-Xon/Xoff  
 byte 5 : Protocol : F-none, M-multi, N-Null, A-ACK/NAK  
 byte 6 : Address : Terminal address range form 'A'-'Y', '0'-'6'  
 byte 7-9 : always SPACE

Returns : **As long**  
 0,2,3,4,5,6 (Please see return value table)

- **Function HM\_date\_time(port, address, str)**

Function description

Set terminal's date and time

Parameter [input]

**port** : **ByVal As long** : Serial COM port  
 1-n for COM1 - COMn  
**address** : **ByVal As Byte** : Terminal address range form 'A'-'Y', '0'-'6'  
 65-89, 48-54  
**str** : **ByVal As String** :  
 byte 0-3 year ex. "1996"  
 byte 4-5 month ex. "02"  
 byte 6-7 day ex. "31"  
 byte 8-9 hour ex. "12"  
 byte 10-11 month ex. "59"  
 byte 12-13 second ex. "59"

Returns : **As long**  
 0,2,3,4,5,6 (Please see return value table)

- **Function HM\_delete( port, address, file\_name)**

Function description

Delete terminal's file

Parameter [input]

**port** : **ByVal As long** : Serial COM port  
 1-n for COM1 - COMn  
**address** : **ByVal As Byte** : Terminal address range form 'A'-'Y', '0'-'6'  
 65-89, 48-54  
**file\_name**: **ByVal As String** :File name

Returns : **As long**  
 0,3,4,5,0x30,0x31

- **Function HM\_dev\_tab( port, address, str)**

Function description

Set terminal's communication parameter

Parameter [input]

**port** : **ByVal As long** : Serial COM port  
 1-n for COM1 - COMn  
**address** : **ByVal As Byte** : Terminal address range form 'A'-'Y', '0'-'6'  
 65-89, 48-54

**str** : **ByVal As String** : Please see terminal's menu  
 for MR350/360  
     byte 0 : Scanner : N-Enable F-Disable  
           byte 1 : Badge : B-Barcode M-Magnetic D-Disable  
           byte 2 : BackLight : N-On F-Off  
           byte 3 : Buzzer : N-On F-Off  
           byte 4 : Keylock : N-Unlock K-Lock P-Partial  
           byte 5 : Volume : 0-Low 5-Medium 9-High  
 for PT700/870/860  
     byte 0 : Scanner : N-Enable F-Disable  
     byte 1 : BackLight : N-On F-Off  
     byte 2 : Buzzer : N-On F-Off  
     byte 4 : Volume : 0-Low 5-Medium 9-High  
     byte 5 : Serial : N-On F-Off  
     byte 6 : Auto-Off : 0-9 minute  
     byte 7 : Always space  
 for MR380  
     byte 0 : External : P-Pen A-Auto M-MSR D-Disable  
     byte 1 : Internal : B-Barcode M-Magnetic D-Disable  
     byte 2 : BackLight : N-On F-Off  
     byte 3 : Buzzer : N-On F-Off  
     byte 4 : Keylock : N-Unlock K-Lock P-Partial  
     byte 5 : Volume : 0-Low 5-Medium 9-High

Returns : **As long**  
 0,2,3,4,5,6 (Please see return value table)

- **Function HM\_dir( port, address, dirstr)**

Function description

Get terminal's directory

Parameter [input]

**port** : **ByVal As long** : Serial COM port  
 1-n for COM1 - COMn

**address** : **ByVal As Byte** : Terminal address range form 'A'-'Y', '0'-'6'  
 65-89, 48-54

Parameter [output]

**str** : **ByRef As String** : Buffer for directory

Returns : **As long**  
 0,3,4,5,6 (Please see return value table)

- **Function HM\_download( port, address, sour\_fn, dest\_fn)**

Function description

Download specified file to terminal (This function will not exit until download complete . If user want to show bytes count or packet count , please use HM\_download1 and HM\_download2)

Parameter [input]

**port** : **ByVal As long** : Serial COM port  
 1-n for COM1 - COMn

**address** : **ByVal As Byte** : Terminal address range form 'A'-'Y', '0'-'6'  
 65-89, 48-54

**sour\_fn** :ByVal As String :File name in PC to be download  
**dest\_fn** :ByVal As String :Assign new file name in terminal  
 Returns : As long  
 0,2,3,4,5,6,0x30 (Please see return value table)

- **Function HM\_download1( port, address , sour\_fn, dest\_fn, lSize)**

Function description

Download specified file to terminal (just check file and send first packet). This function must be used to combine with HM\_download2.

Parameter [input]

**port** :ByVal As long :Serial COM port  
 1-n for COM1 - COMn  
**address** :ByVal As Byte :Terminal address range form 'A'-'Y', '0'-'6'  
 65-89, 48-54  
**sour\_fn** :ByVal As String :File name in PC to be download  
**dest\_fn** :ByVal As String :Assign new file name in terminal  
**lSize** :ByRef As Long :return file's sizes (bytes)

Returns : As long  
 0,2,3,4,5,6,0x30 (Please see return value table)

Example:

```

if HM_OpenCommPort(2, 6, 8, 0, 1) = True then
  state = HM_download1(2, 68, sour_name, dest_name, FSize)
  If state = 0 Then
    state = 1
    Do While state = 1
      state = HM_download2(lByteCounter, lPacketCounter)
      [your code to show information]
    Loop
  End If
End If
HM_CloseCommPort (2)
  
```

- **Function HM\_download2( lByteCount , lPacketCount)**

Function description

Download : Send one packet to terminal (must used with HM\_download1())

Parameter [output]

**lByteCount** : ByRef As Long : Total bytes that had been sent to terminal  
**lPacketCount** : ByRef As Long : Total packet that had been sent to terminal

Returns : As long  
 0,1,2,3,4,5,6 (Please see return value table)

- **Function HM\_exec( port, address, fname)**

Function description

Remote execute program

Parameter[input]

**port** :ByVal As long :Serial COM port  
 1-n for COM1 - COMn  
**address** :ByVal As Byte :Terminal address range form 'A'-'Y', '0'-'6'

65-89, 48-54

**fname : ByVal As String** :File name to be executed (without file extension)

Returns :As long

0,3,4,5,6,0x30,0x31 (Please see return value table)

- **Function HM\_exec\_file(port , address, fname)**

Function description

Check the program name that currently running on terminal

Parameter [input]

**port :ByVal As long** :Serial COM port  
1-n for COM1 - COMn**address :ByVal As Byte** :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

Parameter [output]

**fname : ByVal As String** :store file name that is running on terminal

Returns :As long

0,3,4,5,6,0x35 (Please see return value table)

- **Function HM\_exe\_size(port , address, size)**

Function description

Change size of executing area

Parameter [input]

**port :ByVal As long** :Serial COM port  
1-n for COM1 - COMn**address :ByVal As Byte** :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54**size :ByVal As long** : RAM size (KB)

Returns : As long

0,3,4,5,6,0x40,0x41 (Please see return value table)

- **Function HM\_keypad(Port, Address, status)**

Function description

Set keypad lock status

Parameter [input]

**port :ByVal As long** :Serial COM port  
1-n for COM1 - COMn**address :ByVal As Byte** :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54**status : ByVal As long** :0-Lock 1-Unlock 2-Partial lock

Returns : As long

0,3,4,5,6,0x30,0x31 (Please see return value table)

- **Function HM\_password(Port, Address, str)**

Function description

Remote change supervisor mode password

Parameter [input]

**port :ByVal As long** :Serial COM port  
1-n for COM1 - COMn

**address** : **ByVal As Byte** : Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

**str** : **ByVal As String** : Password (8 bytes maxi.)

Returns : **As long**

0,2,3,4,5,6 (Please see return value table)

- **Function HM\_power\_saving\_off(port, address)**

Function description

Turn off terminal's power saving mode

Parameter [input]

**port** : **ByVal As long** : Serial COM port  
1-n for COM1 - COMn

**address** : **ByVal As Byte** : Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

Returns : **As long**

0,3,4,5 (Please see return value table)

- **Function HM\_power\_saving\_on(Port, Address)**

Function description

Turn on terminal's power saving mode

Parameter [input]

**port** : **ByVal As long** : Serial COM port  
1-n for COM1 - COMn

**address** : **ByVal As Byte** : Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

Returns : **As long**

0,3,4,5 (Please see return value table)

- **Function HM\_poll(Port, address, databuf, data\_len)**

Function description

Poll data from terminal

Parameter [input]

**port** : **ByVal As long** : Serial COM port  
1-n for COM1 - COMn

**address** : **ByVal As Byte** : Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

Parameter [output]

**data\_buf** : **ByRef As String** : store polled data

**data\_len** : **ByRef As long** : length of poll data

Returns : **As long**

0,3,4,5,7 (Please see return value table)

- **Function HM\_ram\_size(Port, Address, t\_size, e\_size, f\_size, et\_size, ef\_size)**

Function description

Get terminal's memory size

Parameter [input]

**port** : **ByVal As long** : Serial COM port  
1-n for COM1 - COMn

**address** :ByVal As Byte :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

Parameter [output]

**t\_size** :ByRef As long :Total memory size  
**e\_size** :ByRef As long :Execution area size  
**f\_size** :ByRef As long :Free RAM disk size  
**et\_size** :ByRef As long :Total EMS RAM disk size (If EMS exist)  
**ef\_size** :ByRef As long :Free EMS RAM disk size (If EMS exist)

Returns : As long

0,3,4,5,6 (Please see return value table)

- **Function HM\_send\_data(Port, Address, data\_buf)**

Function description

Send application data to terminal

Parameter [input]

**port** :ByVal As long :Serial COM port  
1-n for COM1 - COMn  
**address** :ByVal As Byte :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54  
**data\_buf** :ByVal As String :Application data

Returns : As long

0,3,4,5 (Please see return value table)

- **Function HM\_sendback(Port, Address, in\_buf, out\_buf)**

Function description

Send and receive check

Parameter [input]

**port** :ByVal As long :Serial COM port  
1-n for COM1 - COMn  
**address** :ByVal As Byte :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54  
**in\_buf** :ByVal As String :data to be sent

Parameter [output]

**out\_buf** :ByRef As String :received data

Returns : As long

0,3,4,5 (Please see return value table)

- **Function HM\_set\_address(Port, Address, n\_addr)**

Function description

Change terminal address

Parameter [input]

**port** :ByVal As long :Serial COM port  
1-n for COM1 - COMn  
**address** :ByVal As Byte :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54  
**n\_addr** : ByVal As Byte :new terminal address

Returns : As long

0,2,3,4,5 (Please see return value table)

- **Function HM\_term\_tab(Port, Address, str)**

Function description

Set terminal's communication parameter

Parameter [input]

**port** : **ByVal As long** : Serial COM port  
1-n for COM1 - COMn

**address** : **ByVal As Byte** : Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

**str** : **ByVal As String** : Please see terminal's menu

byte 0-7 : Terminal ID

byte 8 : On-line :R-Remote L-Local

byte 9 : Echo : N-On F-Off

byte 10 : AutoLF : N-On F-Off

byte 11 : Mode : C-Character mode B-Block mode

byte 12 : LinePage : L-Line P-Page B-Both

byte 13 : Line terminalor : ASCII

byte 14 : Page terminalor : ASCII

Returns : **As long**

0,2,3,4,5,6 (Please see return value table)

- **Function HM\_upload(Port, Address, sname, dname)**

Function description

Upload specified file from terminal (This function will not exit until whole file is uploaded. So user must wait for a while when file's size is big. If user want to monitor the status of each packet ,please use HM\_upload1 and HM\_upload2)

Parameter [input]

**port** : **ByVal As long** : Serial COM port  
1-n for COM1 - COMn

**address** : **ByVal As Byte** : Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

**sname** : **ByVal As String** : File name in terminal to be upload

Parameter [input/output]

**dname** : **ByRef As String** : Assign new file name in PC (include path), If this new file was exist in PC, system will change it to non-exist file name automatically)

Returns : **As long**

0,3,4,5,0x30,0x33,0x34 (Please see return value table)

- **Function HM\_upload1(Port, Address, sname, dname)**

Function description

Upload specified file from terminal (just check file and send first packet).This function must combine with HM\_upload2.

Parameter [input]

**port** : **ByVal As long** : Serial COM port  
1-n for COM1 - COMn

**address** : **ByVal As Byte** : Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

**sname** : **ByVal As String** :File name in terminal to be upload  
 Parameter [input/output]  
**dname** : **ByRef As String** Assign new file name in PC. If this new file was exist in PC, system will change it to non-exist file name automatically)

Returns : **As long**  
 0,3,4,5,0x30,0x33,0x34 (Please see return value table)

Example :

```
state = HM_upload1(2, 68, sour_name, dest_name)
If state = 0 Then
  state = 1
  Do While state = 1
    state = HM_upload2(IByteCounter, IPacketCounter)
    [your code to show information]
  Loop
End If
```

- **Function HM\_upload2(ByteCnt, PacketCnt)**

Function description

Upload : Receive one packet from terminal (must used with HM\_upload1())

Parameter [output]

**ByteCnt** : **ByRef As Long** :Total bytes that had been received from terminal  
**PacketCnt**: **ByRef As Long** :Total packet that had been received from terminal

Returns : **As long**  
 0,1,2,3,4,5,6 (Please see return value table)

- **Function HM\_upload\_mode(Port, Address)**

Function description

Check whether terminal have been upload or not

Parameter [input]

**port** : **ByVal As long** :Serial COM port  
 1-n for COM1 - COMn  
**address** : **ByVal As Byte** :Terminal address range form 'A'-'Y', '0'-'6'  
 65-89, 48-54

Returns : **As long**  
 0,3,4,5,0x50 (Please see return value table)

- **Function HM\_ups(Port, Address)**

Function description

Send UPS off

Parameter

**port** : **ByVal As long** :Serial COM port  
 1-n for COM1 - COMn  
**address** : **ByVal As Byte** :Terminal address range form 'A'-'Y', '0'-'6'  
 65-89, 48-54

Returns : **As long**  
 0,3,4,5,6 (Please see return value table)

- **Function HM\_kermit ( port, address)**

## Function description

Terminal enter Kermit server mode

## Parameter [input]

**port** :**ByVal As long** :Serial COM port  
1-n for COM1 - COMn

**address** :**ByVal As Byte** :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

Returns : **As long**

0,3,4,5 (Please see return value table)

● **Function HM\_terminal\_id(port, address, str)**

## Function description

Check whether terminal have been upload or not

## Parameter [input]

**port** :**ByVal As long** :Serial COM port  
1-n for COM1 - COMn

**address** :**ByVal As Byte** :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

## Parameter [output]

**str** :**ByRef As String** :terminal id

Returns : **As long**

0,3,4,5 (Please see return value table)

- **Function HM\_320\_field\_port(Port, Address, data\_buf)**

Function description

MR320 field port setting

Parameter [input]

**port** :ByVal As long :Serial COM port  
1-n for COM1 - COMn  
**address** :ByVal As Byte :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54  
**data\_buf** : ByVal As String :Please see MR320 menu

Returns : As long

0,3,4,5,6 (Please see return value table)

- **Function HM\_320\_min\_len(Port, Address, data\_buf)**

Function description

MR320 field port setting

Parameter [input]

**port** :ByVal As long :Serial COM port  
1-n for COM1 - COMn  
**address** :ByVal As Byte :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54  
**data\_buf** : ByVal As String :Please see MR320 menu

Returns : As long

0,2,3,4,5,6 (Please see return value table)

- **Function HM\_320\_record(Port, Address, data\_buf)**

Function description

MR320 field port setting

Parameter [input]

**port** :ByVal As long :Serial COM port  
1-n for COM1 - COMn  
**address** :ByVal As Byte :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54  
**data\_buf** : ByVal As String :Please see MR320 menu

Returns : As long

0,2,3,4,5,6 (Please see return value table)

- **function HM\_file\_exist(port, address, fn, len)**

Function description

Check whether specified file is exist in terminal

Parameter [input]

**port** :ByVal As long :Serial COM port  
1-n for COM1 - COMn  
**address** :ByVal As Byte :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54  
**fn** :ByVal As String :File name to be checked

Parameter [input]

**len** :ByRef As long :Total memory size

Returns : **As long**

0,3,4,5,6,0x30 (Please see return value table)

*Add new return code 0x36 -- When file is existed on flash ROM*

- **Function HM\_run(Port, Address, fname)**

Function description

Remote execute program (same with HM\_exec());

Parameter [input]

**port** :**ByVal As long** :Serial COM port  
1-n for COM1 - COMn

**address** :**ByVal As Byte** :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

**fname** :**ByVal As String** :File name to be executed (without file extension), *if program name is append with "/ROM", it will run the program on Flash ROM*

Returns : **As long**

0,3,4,5,6,0x30,0x31 (Please see return value table)

*Add new return code 0x36 -- When file is existed on flash ROM*

- **Function HM\_dir\_ROM( port, address, dirstr)**

Function description

Get terminal's directory from flash ROM

Parameter [input]

**port** :**ByVal As long** :Serial COM port  
1-n for COM1 - COMn

**address** :**ByVal As Byte** :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

Parameter [output]

**str** **ByRef As String** :Buffer for directory

Returns : **As long**

0,3,4,5,6 (Please see return value table)

9 for HM\_set\_power\_saving\_status(TRUE)

- **Function HM\_cancel\_download(port, address);**

Function description

Cancel download

Parameter [input]

**port** :**ByVal As long** :Serial COM port  
1-n for COM1 - COMn

**address** :**ByVal As Byte** :Terminal address range form 'A'-'Y', '0'-'6'  
65-89, 48-54

Returns :**As long**

0,3,4,5,6 (Please return value table)

9 for HM\_set\_power\_saving\_status(TRUE)

- **Function HM\_cancel\_upload(port, address)**

Function description

Cancel upload

Parameter [input]

**port** :ByVal As long :Serial COM port  
 1-n for COM1 - COMn  
**address** :ByVal As Byte :Terminal address range form 'A'-'Y', '0'-'6'  
 65-89, 48-54  
 Returns :As long  
 0,3,4,5,6 (Please return value table)  
 9 for HM\_set\_power\_saving\_status(TRUE)

- **Function HM\_flash\_size(port, address, t\_size, f\_size);**

Function description

Get memory size of flash ROM

Parameter [input]

**port** :By Val As long :COM port  
 1-n for COM1 - COMn  
**address** :ByVal As Byte :Terminal address range form 'A'-'Y', '0'-'6'  
 65-89, 48-54

Parameter [output]

**t\_size** :ByRef As long :Total ROM size  
**f\_size** :ByRef As long :Free ROM disk size

Returns : As long

0,3,4,5,6 (Please see return value table)  
 9 for HM\_set\_power\_saving\_status(TRUE)

- **Function HM\_firmware\_version(port, address, str);**

Function description

Get terminal's firmware version

Parameter [input]

**port** :ByVal As long :Serial COM port  
 1-n for COM1 - COMn  
**address** :ByVal As Byte :Terminal address range form 'A'-'Y', '0'-'6'  
 65-89, 48-54

Parameter [output]

**str** ByRef As String :Terminal type and version n

Returns : As long

0,3,4,5,6 (Please see return value table)  
 9 for HM\_set\_power\_saving\_status(TRUE)

- **Function HC\_getch(Port)**

Function description

Get one byte (character) from COM port

Parameter [input]

**port** :ByVal As long :Serial COM port  
 1-n for COM1 - COMn

Returns : As Byte

ASCII 0 ~ 254  
 ASCII 255 = no character input

- **Sub HC\_puttch(Port,cc)**

Function description

Output one byte (character) to COM port

Parameter [input]

**port** :ByVal As long :Serial COM port  
1-n for COM1 - COMn

**cc** :ByVal As Byte :output character  
ASCII 0 ~ 254

Returns : No

- **Function HC\_modem\_cmd (Port, str, wait)**

Function description

Send string (modem command) to COM port and then wait for response of modem. This command will automatically append CR to end of string.

Parameter : [input]

**port** :ByVal As long :Serial COM port  
1-n for COM1 - COMn

**str** :ByVal As String :Output string

**wait** :ByVal As long : Maximum Waiting time (milliseconds)

Returns : As long : return code. (Please check table "Modem function return code")

- **Function HC\_modem\_cmd\_mode (port, wait)**

Function description

Force modem return to command mode (send "+++")

Parameter : [input]

**port** :ByVal As long :Serial COM port  
1-n for COM1 - COMn

**wait** :ByVal As long : Maximum Waiting time (milliseconds)

Returns : As long : return code. (Please check table "Modem function return code")

- **Sub HM\_set\_upload\_file\_save\_mode(mode)**

Function Description:

Setup how to save file uploaded from terminal

Parameter [input]:

**mode** :ByVal As long : Save mode (Default is 0)

0 -- Search for an available file(not exist)

1 -- Over-write target file

2 -- Append to target file

- **Function HM\_get\_upload\_file\_save\_mode()**

Function Description:

Get mode of how to save file uploaded from terminal

Parameter [input]: no

Return : As long

0 -- Search for an available file(not exist)

1 -- Over-write target file

2 -- Append to target file

- **Function HM\_version()**

Function Description:

Get DLL version no

Parameter [input]: no

Return : **As long**

- **Communication Return code**

<b>Return Code</b>	<b>Meaning</b>
0	Successfully
0x01	Unsuccessful or successflly download/upload one packed
0x02	Illegal command
0x03	Host get NAK
0x04	Time out
0x05	Get unknown response
0x06	Return string BBC error
0x07	No data to be enquired
0x08	User press ESC to break
0x09	Set Power saving off fail
0x10	Power failure
0x11	Processor exception occurred
0x12	Reset by host
0x13	User reset
0x14	Application reset
0x15	Cold start
0x16	Warm start
0x20	Terminal in diagnostic state
0x21	Terminal already start
0x22	Invalid number of parameter
0x30	File doesn't exist
0x31	Illegal file name
0x32	File already exist ,upload file name to be changed
0x33	File already exist ,system can't change file name
0x34	File open error
0x35	No program is executed
0x40	RAM disk isn't empty
0x41	Out of memory size
0x50	Upload mode

- **Modem function return code**

<b>Return code</b>	<b>Meaning</b>
0	OK
1	CONNECT
2	RING

3	NO CARRIER
4	ERROR
6	NO DIALTONE
7	BUSY
8	NO ANSWER
24	DELAYED
32	BLACKLISTED
33	FAX
35	DATA
40	CARRIER
200	Timeout

## Chapter 6 32 Bits DLL for Delphi

### File description

#### Dir \Delphi

<b>MULTIDLL.DLL</b>	<b>DLL (This file is same with \32bit\dll32\multidll.dll)</b>
<b>MULTIPAS.PAS</b>	<b>Communication Function declare file (Please include this file to your application ex. { \$i multipas.pas })</b>

#### Dir \Delphi\Sample

<b>PROJECT1.EXE</b>	<b>TEST PROGRAM</b>
<b>PROJECT1.DPR</b>	<b>Test APPLICATION (Delphi 2.0)</b>
<b>UNIT1.PAS</b>	<b>form1 source code</b>
<b>MULTIPAS.PAS</b>	<b>same with \delphi\multipas.pas</b>
<b>MULTIDLL.DLL</b>	<b>same with \DLL\multidll.dll</b>
<b>BB.JB2</b>	<b>Test file for download</b>

- **function HM\_OpenCommPort(port; nComBaud; nComDataBit; nComParity; nComStopBit)**

Function Description:

Open communication COM port

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**baud** : **UINT** :Baud Rate  
0-9 for 110,300,600,1200,2400,4800,9600, 19200,38400,57600

**databit** : **integer** :Byte size  
4-8

**parity** : **integer** :parity check  
0 for NO PARITY  
1 for ODD PARITY  
2 for EVEN PARITY

**stopbit** : **integer** :Stop bit length  
1/2

Returns: **BOOL**

TRUE for serial COM port open successfully  
FALSE for serial COM port open failure

- **procedure HM\_CloseCommPort(port)**

Function Description:

Close communication COM port

Parameter: [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

- **procedure HM\_time\_delay(time\_period)**

Function description

Delay specify time period

Parameter: [input]

**time\_period** : **integer** : the number of milliseconds that have elapsed

- **procedure HM\_set\_power\_saving\_status(status)**

function description

For the portable terminal (700/870) , user must execute power saving command before and after each ESC command. This function will set whether each ESC command do power saving setting before and after each command or not.

For example , if status is set TRUE , each ESC command will do follow step

1. Remote set power saving off, if OK do next step.  
If fail return 9 then. User can execute HM\_get\_power\_saving\_return() to get error code
2. If step 1 OK, do the ESC command you call
3. Remote set power saving on, User can execute HM\_get\_power\_saving\_return() to get return code of power saving on setting

If status is set False , each ESC command only execute step 2

(POLL command is not include , use must execute power saving command by himself).

Parameter [input]

**status : BOOL**

True for remote setting power saving

False for NOT remote setting power saving

- **procedure HM\_get\_power\_saving\_status()**

function description

Get Power saving status (please see HM\_set\_power\_saving\_status)

Parameter : no

Return : **BOOL**

True/False

- **function HM\_get\_power\_saving\_return()**

Function description

When HM\_set\_power\_saving\_status(TRUE). System will keep the return code of remote power saving setting in one variable. Use can get this return code by this function

Return : **integer**

0,3,4,5,6 (Please see return value table)

- **function HM\_get\_time\_out()**

Function description

Get current time-out time setting

Return : **longint**

mini-second (Default is 400 = 0.4 second)

- **procedure HM\_set\_time\_out(delay\_time)**

Function description

Set current time-out period

Parameter [input]

**delay\_time** : **longint** :mini-second

Default is 400 = 0.4 second

- **procedure HM\_set\_ESC\_delay(delay\_time)**

Function description

After each ESC command executed. MULTI.DLL will delay this time period default time is 10 milliseconds for on-line terminal or 30 for portable terminal

Parameter: [input]

**time\_delay** : **integer** :the number of milliseconds to be delay

- **function HM\_get\_ESC\_delay()**

Function description

Get ESC delay time (milliseconds)

Return : **integer** :the number of milliseconds to be delay

- **procedure HM\_set\_NAK\_delay(delay\_time)**

Function description

When NAK response is got ,system will delay this time period before do re-try. The default value is 100 for portable terminal or 30 for on-line

Parameter [input]

**time\_delay** :**integer** :the number of milliseconds to be delay

- **function HM\_get\_NAK\_delay()**

Function description

Get NAK delay time

Return : **integer** :the number of milliseconds to be delay

- **function HM\_warm\_start(port; address)**

Function description

Remote warm-start terminal

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

Returns : **integer**

0,3,4,5 (Please see return value table)

- **function HM\_autoboot(port; address; fn)**

## Function description

Set file name that terminal will auto-execute after power on

## Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn  
**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'  
**fn** : **PChar** :File name (without file extention)

Returns : **integer**  
0,3,4,5,6,0x30,0x31 (Please see return value table)

- **function HM\_bar\_tab(port; address; str)**

## Function description

Set terminal's bar code symbology

## Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn  
**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'  
**str** : **PChar** :for PT870

byte 0	: Code 39	E: enable D: disable F: full
byte 1	: I 2 of 5	E: enable D: disable
byte 2	: Codar bar	E: enable D: disable
byte 3	: EAN/UPC	E: enable D: disable
byte 4	: Code 128	E: enable D: disable
for MR350/360/380		
byte 0	: Code 39	N: enable D: disable F: full
byte 1	: I 2 of 5	N: enable F: disable
byte 2	: Codar bar	N: enable F: disable
byte 3	: EAN/UPC	N: enable F: disable
byte 4	: Code 128	N: enable F: disable
for PT700		
byte 0	: Code 39	E: enable D: disable F: full
byte 1	: I 2 of 5	E: enable D: disable
byte 2	: Codar bar	E: enable D: disable
byte 3	: EAN/UPC	E: enable D: disable
byte 4	: Code 128	E: enable D: disable
byte 5	: EAN 128	E: enable D: disable
byte 6	: Code 93	E: enable D: disable
byte 7	: Trioptic	E: enable D: disable

Returns : **integer**  
0,2,3,4,5,6 (Please see return value table)

- **function HM\_buzzer(port ; address; status)**

## Function description

Set terminal's buzzer volume

## Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn  
**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

**status** : **integer** :Sound level

0 for Low

5 for medium

9 for High

Returns : **integer**

0,3,4,5,6 (Please see return value table)

- **function HM\_cold\_start(port ; address)**

Function description

Remote cold start terminal

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

Returns : **integer**

0,3,4,5 (Please see return value table)

- **function HM\_comm\_tab(port ; address ; str)**

Function description

Set terminal's communication parameter

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

**str** : **PChar** :Please see terminal's menu  
for MR350/360/380

byte 0 : Baud rate : 1-9 for 110,300,600,1200,2400...38400

byte 1 : Stop bit : 1/2

byte 2 : Bit size : 7/8

byte 3 : Parity : N-No Parity, O-Odd parity, E-Even parity

byte 4 : Protocol : F-No protocol, M-Multi protocol

byte 5 : Address : Terminal address range form 'A'-'Y', '0'-'6'

byte 6,7: Time out : 00-99

for MR320

byte 0-7 same with MR350

byte 8 : always F

for PT700/870/860

byte 0-3 same with MR350

byte 4 : Flow control : F-none, C-CTS/RTS, X-Xon/Xoff

byte 5 : Protocol : F-none, M-multi, N-Null, A-ACK/NAK

byte 6 : Address : Terminal address range form 'A'-'Y', '0'-'6'

byte 7-9 : always SPACE

Returns : **integer**

0,2,3,4,5,6 (Please see return value table)

- **function HM\_date\_time(port ; address ; str)**

Function description

Set terminal's date and time

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

**str** : **PChar** :date & time string

byte 0-3 year ex. "1996"

byte 4-5 month ex. "02"

byte 6-7 day ex. "31"  
 byte 8-9 hour ex. "12"  
 byte 10-11 month ex. "59"  
 byte 12-13 second ex. "59"

Returns : **integer**

0,2,3,4,5,6 (Please see return value table)

- **function HM\_delete(port ; address ; fname)**

Function description

Delete terminal's file

Parameter [input]

**port** : **UINT** :Serial COM port  
 1-n for COM1 - COMn  
**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'  
**fname** : **PChar** :File name

Returns : **integer**

0,3,4,5,0x30,0x31

- **function HM\_dev\_tab(port ; address ; str)**

Function description

Set terminal's communication parameter

Parameter [input]

**port** : **UINT** :Serial COM port  
 1-n for COM1 - COMn  
**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'  
**str** : **PChar** :Please see terminal's menu

for MR350/360

byte 0 : Scanner : N-Enable F-Disable

byte 1 : Badge : B-Barcode M-Magnetic D-Disable

byte 2 : BackLight : N-On F-Off

byte 3 : Buzzer : N-On F-Off

byte 4 : Keylock : N-Unlock K-Lock P-Partial

byte 5 : Volume : 0-Low 5-Medium 9-High

for PT700/870/860

byte 0 : Scanner : N-Enable F-Disable

byte 1 : BackLight : N-On F-Off

byte 2 : Buzzer : N-On F-Off

byte 4 : Volume : 0-Low 5-Medium 9-High

byte 5 : Serial : N-On F-Off

byte 6 : Auto-Off : 0-9 minute

byte 7 : Always space

for MR380

byte 0 : External : P-Pen A-Auto M-MSR D-Disable

byte 1 : Internal : B-Barcode M-Magnetic D-Disable

byte 2 : BackLight : N-On F-Off

byte 3 : Buzzer : N-On F-Off

byte 4 : Keylock : N-Unlock K-Lock P-Partial

byte 5 : Volume : 0-Low 5-Medium 9-High

Returns : **integer**  
0,2,3,4,5,6 (Please see return value table)

- **function HM\_dir(port ; address ; str)**

Function description

Get terminal's directory

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

Parameter [output]

**str** : **PChar** :Buffer for directory

Returns : **integer**  
0,3,4,5,6 (Please see return value table)

- **function HM\_download(port ; address ; sour\_fn; dest\_fn )**

Function description

Download specified file to terminal (This function will not exit until download complete . If user want to show bytes count or packet count , please use HM\_download1 and HM\_download2)

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

**sour\_fn** : **PChar** :File name in PC to be download

**dest\_fn** : **PChar** :Assign new file name in terminal

Returns : **integer**  
0,2,3,4,5,6,0x30 (Please see return value table)

- **function HM\_download1(port ; address ; sour\_fn; dest\_fn ; lTotalLength)**

Function description

Download specified file to terminal (just check file and send first packet). This function must be used to combine with HM\_download2.

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

**sour\_fn** : **PChar** :File name in PC to be download

**dest\_fn** : **PChar** :Assign new file name in terminal

**lSize** : **pointer** (integer) :return file's sizes (bytes)

Returns : **integer**  
0,2,3,4,5,6,0x30 (Please see return value table)

Example:

```
fn1 := 'bb.jb2';
```

```
fn2 := 'bb.jb2';
```

```
HM_OpenCommPort(2, 6, 8, 0, 1);
```

```
ret := HM_download1(2,chr(68), PChar(fn1), PChar(fn2), Addr(lTotalLength));
```

```
if (ret=0) then
```

```
begin
```

```

Repeat
  ret := HM_download2(Addr(IByteCounter), Addr(IPacketCounter));
          { print packet counter and bytes counter }
until ret <> 1;
end;
  end;
  HM_CloseCommPort(2);

```

- **function HM\_download2(IByteCount ; IPacketCount)**

Function description

Download : Send one packet to terminal (must used with HM\_download1())

Parameter [output]

**IByteCount** : **pointer** (of integer): Total bytes that had been sent to terminal

**IPacketCount** : **pointer** (of integer): Total packet that had been sent to terminal

Returns : **integer**

0,1,2,3,4,5,6 (Please see return value table)

- **function HM\_exec(port ; address ; fname)**

Function description

Remote execute program

Parameter[input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

**fname** : **PChar** :File name to be executed (without file extension)

Returns : **integer**

0,3,4,5,6,0x30,0x31 (Please see return value table)

- **function HM\_exec\_file(port ; address ; fn)**

Function description

Check the program name that currently running on terminal

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

Parameter [output]

**fn** : **PChar** :store file name that is running on terminal

Returns : **integer**

0,3,4,5,6,0x35 (Please see return value table)

- **function HM\_exe\_size(port ; address ; size)**

Function description

Change size of executing area

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

**size** : **integer** : RAM size (KB)

Returns : **integer**  
 0,3,4,5,6,0x40,0x41 (Please see return value table)

- **function HM\_keypad(port ; address ; status)**

Function description

Set keypad lock status

Parameter [input]

**port** : **UINT** :Serial COM port  
 1-n for COM1 - COMn  
**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'  
**status** : **integer** :0-Lock 1-Unlock 2-Partial lock

Returns : **integer**  
 0,3,4,5,6,0x30,0x31 (Please see return value table)

- **function HM\_password(port ; address ; str)**

Function description

Remote change supervisor mode password

Parameter [input]

**port** : **UINT** :Serial COM port  
 1-n for COM1 - COMn  
**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'  
**str** : **PChar** :Password (8 bytes maxi.)

Returns : **integer**  
 0,2,3,4,5,6 (Please see return value table)

- **function HM\_power\_saving\_off(port ; address )**

Function description

Turn off terminal's power saving mode

Parameter [input]

**port** : **UINT** :Serial COM port  
 1-n for COM1 - COMn  
**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

Returns : **integer**  
 0,3,4,5 (Please see return value table)

- **function HM\_power\_saving\_on(port ; address )**

Function description

Turn on terminal's power saving mode

Parameter [input]

**port** : **UINT** :Serial COM port  
 1-n for COM1 - COMn  
**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

Returns : **integer**  
 0,3,4,5 (Please see return value table)

- **function HM\_poll(port ; address ; data\_buf; data\_len)**

Function description

Poll data from terminal

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn  
**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

Parameter [output]

**data\_buf** : **PChar** :store polled data  
**data\_len** : **pointer** (integer) :length of poll data

Returns : **integer**

0,3,4,5,7 (Please see return value table)

- **function HM\_ram\_size(port ; address; t\_size; e\_size; f\_size; et\_size; ef\_size)**

Function description

Get terminal's memory size

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn  
**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

Parameter [output]

**t\_size** : **pointer** (integer) :Total memory size  
**e\_size** : **pointer** (integer) :Execution area size  
**f\_size** : **pointer** (integer) :Free RAM disk size  
**et\_size** : **pointer** (integer) :Total EMS RAM disk size (If EMS exist)  
**ef\_size** : **pointer** (integer) :Free EMS RAM disk size (If EMS exist)

Returns : **pointer**

0,3,4,5,6 (Please see return value table)

30,0x31 (Please see return value table)

- **function HM\_send\_data(port ; address ; data\_buf)**

Function description

Send application data to terminal

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn  
**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'  
**data\_buf** : **PChar** :Application data

Returns : **integer**

0,3,4,5 (Please see return value table)

- **function HM\_sendback(port ; address; in\_buf; out\_buf)**

Function description

Send and receive check

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn  
**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'  
**in\_buf** : **PChar** :data to be sent

Parameter [output]

**out\_buf** : PChar :received data

Returns : integer

0,3,4,5 (Please see return value table)

- **function HM\_set\_address(port ; address ; n\_addr)**

Function description

Change terminal address

Parameter [input]

**port** : UINT :Serial COM port  
1-n for COM1 - COMn

**address** : char :Terminal address range form 'A'-'Y', '0'-'6'

**n\_addr** : char :new terminal address

Returns : integer

0,2,3,4,5 (Please see return value table)

- **function HM\_term\_tab(port ; address ; str)**

Function description

Set terminal's communication parameter

Parameter [input]

**port** : UINT :Serial COM port  
1-n for COM1 - COMn

**address** : char :Terminal address range form 'A'-'Y', '0'-'6'

**str** : PChar :Please see terminal's menu

byte 0-7 : Terminal ID

byte 8 : On-line :R-Remote L-Local

byte 9 : Echo :N-On F-Off

byte 10 : AutoLF :N-On F-Off

byte 11 : Mode :C-Character mode B-Block mode

byte 12 : LinePage :L-Line P-Page B-Both

byte 13 : Line terminalor : ASCII

byte 14 : Page terminalor : ASCII

Returns : integer

0,2,3,4,5,6 (Please see return value table)

- **function HM\_upload(port ; address ; sname; dname)**

Function description

Upload specified file from terminal (This function will not exit until whole file is uploaded. So user must wait for a while when file's size is big. If user want to monitor the status of each packet ,please use HM\_upload1 and HM\_upload2)

Parameter [input]

**port** : UINT :Serial COM port  
1-n for COM1 - COMn

**address** : char :Terminal address range form 'A'-'Y', '0'-'6'

**sname** : PChar :File name in terminal to be upload

Parameter [input/output]

**dname** : PChar :Assign new file name in PC (include path), If this new file was exist in PC, system will change it to non-exist file name automatically)

Returns : integer

0,3,4,5,0x30,0x33,0x34 (Please see return value table)

- **function HM\_upload1(port ; address ; sname; dest\_name)**

Function description

Upload specified file from terminal (just check file and send first packet). This function must combine with HM\_upload2.

Parameter [input]

**port** : UINT :Serial COM port  
1-n for COM1 - COMn  
**address** : char :Terminal address range form 'A'-'Y', '0'-'6'  
**sname** : PChar :File name in terminal to be upload

Parameter [input/output]

**dname** : Pchar : Assign new file name in PC. If this new file was exist in PC, system will change it to non-exist file name automatically)

Returns : integer

0,3,4,5,0x30,0x33,0x34 (Please see return value table)

Example :

```
fn1 := 'bb.jb2';
fn2 := 'bb.jb2';
HM_OpenCommPort(2, 6, 8, 0, 1);
{ if COM port open successfully }
ret := HM_upload1(2,chr(68), PChar(fn1), PChar(fn2));
if (ret=0) then
begin
Repeat
ret := HM_upload2(Addr(lByteCounter), Addr(lPacketCounter));
{ print packet counter and bytes counter }
until ret <> 1;
end;
end;
HM_CloseCommPort(2);
```

- **function HM\_upload2(ByteCnt; PacketCnt : pointer)**

Function description

Upload : Receive one packet from terminal (must used with HM\_upload1())

Parameter [output]

**ByteCnt** : pointer (integer) :Total bytes that had been received from terminal  
**PacketCnt**: pointer (integer) :Total packet that had been received from terminal

Returns : integer

0,1,2,3,4,5,6 (Please see return value table)

- **function HM\_upload\_mode(port ; address )**

Function description

Check whether terminal have been upload or not

Parameter [input]

**port** : UINT :Serial COM port  
1-n for COM1 - COMn  
**address** : char :Terminal address range form 'A'-'Y', '0'-'6'

Returns : **integer**  
0,3,4,5,0x50 (Please see return value table)

- **function HM\_ups(port ; address )**

Function description

Send UPS off

Parameter

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

Returns : **integer**  
0,3,4,5,6 (Please see return value table)

- **function HM\_kermit(port ; address )**

Function description

Terminal enter Kermit server mode

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

Returns : **integer**  
0,3,4,5 (Please see return value table)

- **function HM\_terminal\_id(port ; address ; str)**

Function description

Check whether terminal have been upload or not

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

Parameter [output]

**str** : **PChar** :terminal id

Returns : **integer**  
0,3,4,5 (Please see return value table)

- **function HM\_320\_field\_port(port ; address r; data\_buf)**

Function description

MR320 field port setting

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

**data\_buf** : **PChar** :Please see MR320 menu

Returns : **integer**  
0,3,4,5,6 (Please see return value table)

- **function HM\_320\_min\_len(port ; address ; data\_buf)**

Function description

MR320 field port setting

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn  
**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'  
**data\_buf** : **PChar** :Please see MR320 menu

Returns : **integer**

0,2,3,4,5,6 (Please see return value table)

- **function HM\_320\_record(port ; address ; data\_buf)**

Function description

MR320 field port setting

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn  
**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'  
**data\_buf** : **PChar** :Please see MR320 menu

Returns : **integer**

0,2,3,4,5,6 (Please see return value table)

- **function HM\_file\_exist(port ; address ; fn; len)**

Function description

Check whether specified file is exist in terminal

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn  
**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'  
**fn** : **PChar** :File name to be checked

Parameter [output]

**len** : **Pointer** :File length, if file exist

Returns : **integer**

0,3,4,5,6,0x30 (Please see return value table)

*Add new return code 0x36 -- When file is existed on flash ROM*

- **function HM\_run(port ; address ; str)**

Function description

Remote execute program (same with HM\_exec());

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn  
**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

**fname** : **PChar** :File name to be executed (without file extension), *if program name is append with "/ROM", it will run the program on Flash ROM*

Returns : **integer**

0,3,4,5,6,0x

*Add new return code 0x36 -- When file is existed on flash ROM*

- **function HM\_dir\_ROM(port; address; str)**

## Function description

Get terminal's directory from flash ROM

## Parameter

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

## Parameter [output]

**str** : **PChar** :Buffer for directory

## Returns :integer

0,3,4,5,6 (Please see return value table)

- **function HM\_cancel\_download(port;address)**

## Function description

Cancel download

## Parameter

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

Returns :0,3,4,5,6 (Please return value table)  
9 for HM\_set\_power\_saving\_status(TRUE)

- **function HM\_cancel\_upload(port; address)**

## Function description

Cancel upload

## Parameter

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

Returns :0,3,4,5,6 (Please return value table)  
9 for HM\_set\_power\_saving\_status(TRUE)

- **function HM\_flash\_size(port; address; t\_size; f\_size);**

## Function description

Get memory size of flash ROM

## Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

## Parameter [output]

**t\_size** : **pointer** (integer) :Total ROM size

**f\_size** : **pointer** (integer) :Free ROM disk size

## Returns : pointer

0,3,4,5,6 (Please see return value table)

- **function HM\_firmware\_version(port; address; str)**

## Function description

Get terminal's firmware version

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**address** : **char** :Terminal address range form 'A'-'Y', '0'-'6'

Parameter [output]

**str** : **PChar** :Terminal type and version no  
returns :0,3,4,5,6 (Please return value table)  
9 for HM\_set\_power\_saving\_status(TRUE)

- **Function HC\_getch(Port)**

Function description

Get one byte (character) from COM port

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

Returns : **char**

ASCII 0 ~ 254

ASCII 255 = no character input

- **procedure HC\_puttch(Port,cc)**

Function description

Output one byte (character) to COM port

Parameter [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**cc** : **char** :output character  
ASCII 0 ~ 254

Returns : No

- **Function HC\_modem\_cmd (Port, str, wait)**

Function description

Send string (modem command) to COM port and then wait for response of modem. This command will automatically append CR to end of string.

Parameter : [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**str** : **PChar** :Output string

**wait** : **longint** : Maximum Waiting time (milliseconds)

Returns : **integer** : return code. (Please check table "Modem function return code")

- **Function HC\_modem\_cmd\_mode (port, wait)**

Function description

Force modem return to command mode (send "+++")

Parameter : [input]

**port** : **UINT** :Serial COM port  
1-n for COM1 - COMn

**wait** : **longint** : Maximum Waiting time (milliseconds)

Returns : **integer** : return code. (Please check table "Modem function return code")

- **procedure HM\_set\_upload\_file\_save\_mode(mode)**

Function description :

Setup how to save file uploaded from terminal

Parameter:[input]

**mode : integer**

0 -- Search for an available file(not exist)

1 -- Over-write target file

2 -- Append to target file

- **function HM\_get\_upload\_file\_save\_mode()**

Function description

Get mode of how to save file uploaded from terminal

Parameter: no

return : **integer**

0 -- Search for an available file(not exist)

1 -- Over-write target file

2 -- Append to target file

- **function HM\_version()**

Function description

Get DLL version

Parameter: no

Return : integer

- **Communication Return code**

<b>Return Code</b>	<b>Meaning</b>
0	Successfully
0x01	Unsuccessful or successfly download/upload one packed
0x02	Illegal command
0x03	Host get NAK
0x04	Time out
0x05	Get unknown response
0x06	Return string BBC error
0x07	No data to be enquired
0x08	User press ESC to break
0x09	Set Power saving off fail
0x10	Power failure
0x11	Processor exception occurred
0x12	Reset by host
0x13	User reset
0x14	Application reset
0x15	Cold start
0x16	Warm start
0x20	Terminal in diagnostic state

0x21	Terminal already start
0x22	Invalid number of parameter
0x30	File doesn't exist
0x31	Illegal file name
0x32	File already exist ,upload file name to be changed
0x33	File already exist ,system can't change file name
0x34	File open error
0x35	No program is executed
0x40	RAM disk isn't empty
0x41	Out of memory size
0x50	Upload mode

● **Modem function return code**

<b>Return code</b>	<b>Meaning</b>
0	OK
1	CONNECT
2	RING
3	NO CARRIER
4	ERROR
6	NO DIALTONE
7	BUSY
8	NO ANSWER
24	DELAYED
32	BLACKLISTED
33	FAX
35	DATA
40	CARRIER
200	Timeout