JARLTECH

*ISO 9002 Certified*
*Lead with technology*
*Win customers with service*

# Customer Pole Display
# MODEL 8003

## OPERATION MANUAL

Every effort has been made to ensure that the information in this manual is accurate. Jarltech is not responsible for printing or clerical errors.

Jarltech International Inc.
3F, No. 1, Lane 538, Chung Cheng Road, Hsin Tien,
Taipei, Taiwan, R.O.C.

This equipment has been tested and found to comply with the limits for Class A digital device. Pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and if not installed and used in accordance with the instructions may cause harmful interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on. The user is encouraged to try correct interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help. This booklet is available from the U.S. government Printing Office, Washington, DC 20402, Stock NO.004-000-00345-4.

**CAUTION:** Any changes of modifications not expressly approved by the grantee of this device could void the user's authority to operate the equipment.

Operation is subject to the following two conditions:

(1) This device may not cause harmful interference.

(2) This device must accept any interference received including interference that may cause undesired operation.

# *Contents*

# 1 Before You Install

This manual describes functions and usage of the Jarltech Model 8003 customer pole display.

The 8003 is a 2x20 alphanumeric customer pole display designed for retail and industrial environments. Its outstanding features include high quality LCD with backlight, large sized characters(12.7mm), RS232 interface, easy to use and powerful programming features.

## Step 1: Turn Off Your Computer

By shutting off your computer, you will prevent any accidental damage to the pole display and computer.

## Step 2: Review Packing List

Please ensure that your pole display shipment is complete.

Model 8003 includes:

- 1 pce 8003 pole display
- 1 pce operation manual
- 11 pce +12V DC power plate with internal power cable (GC-POS-POWER)
- 1 pce DC cable (GC-RCA-DC)
  1 pce Y-cable (GC-8003P)

# 2 Installing Model 8003

This chapter describes the procedures for installing the 8003 pole display by using RS232C interface.

## Step 1: Turn off your computer

If you have not already done so, turn off your computer to avoid any accidental damage to the pole display and computer.

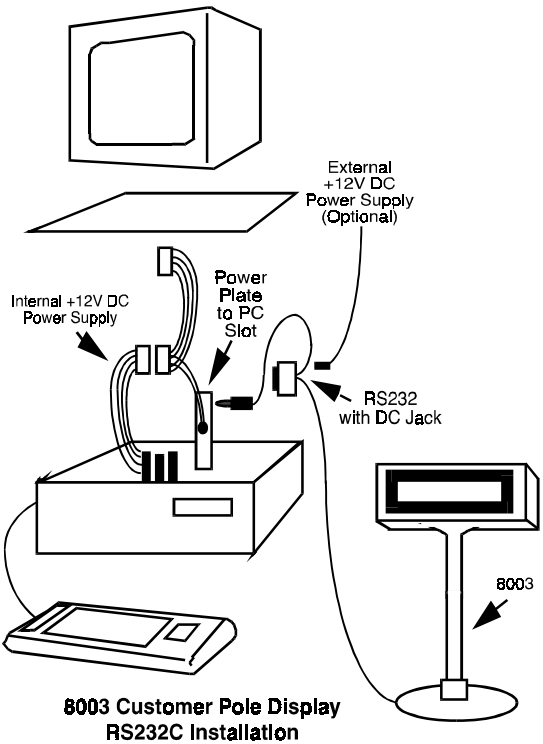
## Step 2: Decide on power access

The RS232 connection requires power +12V DC. This may be provided through an internal connection in your computer or through an external connection to a 110/220V adapter.

The components for an internal connection are provided. If you are using an external connection, be sure that your adaptor confirms with the specifications listed in Appendix I, then go to step 4.

## Step 3: Using internal power source

Refer to the installation diagram on the following page. Remove the access cover to your computer. Mount the +12V DC power plate on an available expansion slot in the back of your computer. Attach the 4-pin male connector to the open female connector of the same type in your computer. Alternatively, an internal power source may be available already if the 9-pin RS232 port on your computer or terminal matches the 8003 pin assignment (see Specifications in Appendix I).




**8003 Customer Pole Display**
**RS232C Installation**

## Step 4: Connect to your computer

Connect the 9-pin female RS232 connector (DB9F with DC jack) to the male equivalent (DB9M) RS232 port on your computer or terminal. Provide power to the DC jack on the DB9F connector using either a cable connection to the +12V DC power plate or an external adapter.

## Step 5: Turn on your computer

Turn on your computer. It should boot up normally. The pole display will show a self-diagnostic status and then the display will be blank.

## Step 6: Turn to Chapter 3

You are now ready for operation, please refer to Chapter 3 for programming to meet the specific requirement of your application environment.

# 3 Programming Commands

## Introduction

The basic function of the 8003 display is comparable to the display programming by your software should be as easy. You just have to open the COM-port on which the display has been connected by you. Then, you just send the character you want the 8003 to display directly to the COM interface. Please use the following RS232 parameters:

***9600 Baud, No Parity, 8 Data Bits, 1 Stop bit***

In Qbasic, you would initialize the interface as follows:

***OPEN "COMx: 9600, N, 8, DS0" FOR OUTPUT AS #1***

(x=number of the COM port you are using for the display) And you would print something to the display using the PRINT command:

***PRINT#1, "Hello World!"***

In the end, you can close the interface:

***CLOSE #1***

In other programming languages, the commands for serial output shall be different, but they will work in a similar way. For some compilers, you will need an extra toolbox, that offers you RS232 routines. Please refer to your compiliers/interpreters manual for more details.

***Example:***
***OPEN "COM1: 9600, N, 8, 1, DS0" FOR OUTPUT AS #1***
***PRINT #1, "Hello World!"***
***CLOSE #1***

## Programming using DOS routines

You can also generate a display output using the simple DOS routines.

***Example:***
***MODE COM1: 9600, N, 8, 1***
***ECHO Hello! >COM1:***

## Control characters and special functions

*For special display functions, there are some commands which will be explained in this chapter. Some of the commands consist of one ASCII-CTRL-code, others are command strings, introduced by ESC.*

*If a command needs additional parameters, please do not forget to use ASCII format for the parameter. That means, if the parameter is 0(zero), then you have to transmit the ASCII code "0" (=CHR$(48) in Basic; 48 is the decimal position of the "0" character in the ASCII code table). But please consider that only ONE byte is allowed for each parameter. That is why you cannot transmit two-digit numbers. In this case, just add the number you want to transmit as parameter 48 and transmit the corresponding character. For example, if you want to transmit the parameter 11, you have to send CHR$(11+48)=CHR$(59)=";". Attention: For some other commands, only BYTE values are allowed as parameter. For those, you directly send the corresponding character code without adding 48 (e.g. CHR$(11) for 11). For details, please refer to the individual command code descriptions.*

**Example: Set the cursor to the last position in the display area**

**WRONG:**

```
PRINT#1,CHR$(27)+"=";          :REM command ESC =
PRINT #1, "19" + "1"           :REM parameter column
                               19, line 1
```

**CORRECT:**

```
PRINT#1,CHR$(27)+"=";";
PRINT#1,CHR$(48+19)+"1"        :REM or
                                   CHR$(48+19)+CHR$(48+1)
```

Below is a list of command sequences for user to design an interface to the JARLTECH 8003 customer pole display. Please note that pole display is default with **9600 bps baud rate, no parity, 8 data bits, 1 stop bit.**

## Command codes explanation (control sequences)

| COMMANDS | FUNCTION | DESCRIPTION |
|---|---|---|
| | **Wrap mode** | |
| CTRL A | Turn on wrap mode<br>Code: 001 | This allows the text displayed to the screen to wrap to the next line when the cursor position exceeds the right handside boundary. If autoscroll is also on and the cursor is on the bottom line, the screen will scroll up one row. |
| CTRL B | Turn off wrap mode<br>Code: 002 | When the cursor position meets the right hand side boundary, the cursor will not continue. If any further characters are received then they will over write the last character at the right handside. |

| COMMANDS | FUNCTION | DESCRIPTION |
|---|---|---|
| | **Cursor Move** | |
| CTRL H | Move cursor left one column<br>Code: 008 | This is simply the BACK SPACE function, though characters are not deleted as you back space over them. When you reach the beginning of a line, the cursor will wrap to end of the previous line until cursor = 0,0 is met. |
| CTRL J | Scroll (line feed)<br>Code: 010 | This is the LINE FEED function. It will move the cursor down one line. It will always scroll the screen if at the bottom. |
| CTRL V | Move cursor down one row<br>Code: 022 | This is an alternative LINE FEED function that will not scroll the screen up one row when at the bottom line. |
| CTRL K | Move cursor up one row<br>Code: 011 | This control sequence will move the cursor up one row. if it is at the top of the screen, it will wrap to the bottom line, the cursor's horizontal location stays the same. |
| CTRL L | Move cursor right one column<br>Code: 012 | This is RIGHT ARROW function. It will move the cursor right by one character cell. If it is at the end of a line, the cursor will wrap to the next line until the bottom righthand side is met. |

| COMMANDS | FUNCTION | DESCRIPTION |
|---|---|---|
| CTRL M | Move cursor to column 0<br>Code: 013 | This is CARRIAGE RETURN function which returns the cursor's horizontal location to the first position, on the same line. |

***NOTE:*** ***In BASIC, after a PRINT#-command, a CR is always sent to the display if you do not add an ";" to the end of the command. The CR command is normally used for the line switching.***

```
Example:
PRINT #1, “First line!”
PRINT #1, CHR$(10);
PRINT #1, “Second line!”
```

| | | |
|---|---|---|
| CTRL ^ | Cursor home<br>Code: 030 | This function will return the cursor position to 0,0. |

| **Clear unprotected sells** | | |
|---|---|---|
| CTRL Z<br>or<br>ESC: | Home cursor, clear unprotected cells to nulls<br>Code: 026<br>Code: 027, 058 | This function will clear all the unprotected characters to blank character cells and returns the cursor to 0,0. |

| **Reset** | | |
|---|---|---|
| CTRL \ | Reset display<br>Code: 028 | This function will execute a software reset which will initialize the entire pole display. The power up test will begin as if power was just switched on. |

## Command codes explanation (escape sequences)

| COMMANDS | FUNCTION | DESCRIPTION |
|---|---|---|
| | **Character attribute: cell protected** | |
| ESC & | Set protected mode<br>Code: 027, 038 | All characters sent to the display after this command are displayed in protected so that they cannot be overwritten by other unprotected characters. This function is useful especially for scrolling, if you want some characters not to be fixed and not influenced by the scrolling function. This has a kind of "window effect".<br>You can also use this function to use a simple delete command to clear all character except the protected ones very fast. |
| ESC' | Clear protected mode<br>Code: 027, 039 | This function simply returns the character attribute to unprotected. |
| ESCV | Protect cursor column<br>Code: 027, 086 | This function will set all characters, vertically at the current cursor "X" position, to the protected attribute. |

**EXAMPLE:**
This is useful when you have a series of lists and the beginning selections are the same. e.g :

(1) : option 1
(2) : option 2
(3) : option 3
(4) : option 4

In this example, you would protect the "(x) :"" fields of columns 0, 1, 2, & 3, by moving the cursor to each column and using ESC V.

| COMMANDS | FUNCTION | DESCRIPTION |
|---|---|---|

**Automatic Scrolling lines**

With the following commands, you can define up to 8 scrolling strings. Using another command, you can then start and stop them in a certain display line. Your PC does not have to care about this. The display does the scrolling on its own until it receives the stop command.

| | | |
|---|---|---|
| ESC( | Program a message for the scrolling lines<br>Code: 027, 040 | This function allows the programmer to down load 1 of 8 messages for lines that you are going to scroll.<br>These messages are 255 bytes long or can be terminated by carriage return, ENTER [010,013]. |

**PARAMETER FORMAT:**
ESC(<BLOCK><MESSAGE>
**RANGES:**
BLOCK :"1" - "8" (049 - 056)
**MESSAGE:**
Any text string terminated by 010,013
**EXAMPLE:**
PRINT #1, CHR$(27)+"< 1 This is scrolled"
**ATTENTION:**
Please do not forget to use ASCII format for the parameters, so that for the text number, only the codes 049-056 are allowed. Do not send 001-008!

| | | |
|---|---|---|
| ESC) | Start a line scrolling<br>Code: 027, 041 | This function starts one of the total number of lines, being (1 or 2), scrolling horizontally. You may specify the direction, speed and message. |

| COMMANDS | FUNCTION | DESCRIPTION |
|---|---|---|
| **PARAMETERFORMAT:**<br>ESC)<LINE><DIRECTION><SPEED><BLOCK_NO><br>**RANGES:**<br>LINE : "0"" (048) = first<br>"1"" (049) = second<br>DIRECTION : "0" = right<br>"1" = left<br>SPEED : 0 to 16 (048-064).<br>BLOCK_NO : "1" to "8" (049-056)<br>**EXAMPLE:**<br>PRINT #1, CHR$(27)+")0041" or PRINT #1, CHR$(27)+CHR$(41)+CHR$(48)+CHR$(48)+CHR(52)+CHR$(49)<br>This commands start the scrolling in the first line from the left to the right with the speed 4 using the scrolling text no.1 (as defined in the example above) | | |
| ESC% | Stop a line from scrolling<br>Code: 027, 037 | This function will stop one of the LCD rows from scrolling its message. For further information on scrolling message, please refer to the explanation on :<br>ESC ( : Program a message for the scrolling lines.<br>ESC ) : Start a line scrolling. |
| **PARAMETERFORMAT:**<br>ESC%<LINE><br>**RANGES:**<br>LINE : "0" (048)=first line<br>"1" (049)=second line | | |
| **Clear & Fill** | | |
| ESC*<br>or<br>ESC+ | Turn off protect mode, cursor home (clear all)<br>Code: 027, 042<br>Code: 027, 043 | These 2 command codes clear the entire screen, regardless of the protected or unprotected attributes, to unprotected cells and returns the cursor to 0,0. |

| COMMANDS | FUNCTION | DESCRIPTION |
|---|---|---|

**EXAMPLE:**

```
PRINT #1,  "Press any key."
   INPUT a$, : REM Wait for key press.
  PRINT #1,  CHR$(27)+ "*" + "Thanks"
```

| | | |
|---|---|---|
| ESC, | Clear screen to protected spaces (home cursor)<br>Code: 027, 044 | This special function clears the whole display (no matter if protected or not) and fills the display with protected space. This disables the display until you clear the protection using ESC*/ ESC+. |
| ESC. | Replace all unprotected cells with the specified character<br>Code: 027, 046 | This function replace all character cells that are unprotected with the character specified. For example, if the entire display has unprotected cells, you could flood fill it with the '*'' or '$'....etc. |

**PARAMETER FORMAT:**
ESC.<CHARACTER>
**RANGES:**
CHARACTER: 00-255 (000-255).
**EXAMPLE:**

```
PRINT #1, CHR$(27)+".*": REM Fills the display with "*"
```

| | | |
|---|---|---|
| CTRL Z or ESC: | Home cursor, clear unprotected cells to nulls<br>Code: 026<br>Code: 027, 058 | This function will clear all the unprotected characters to blank character cells and returns the cursor to 0,0. |
| ESC! | Clear unprotected characters to spaces<br>Code: 027, 033 | This function is the same as CTRL Z & ESC : , except that the cursor position is not changed |

| COMMANDS | FUNCTION | DESCRIPTION |
|---|---|---|
| ESC ; | Home cursor, clear unprotected cells to spaces<br>Code: 027, 059 | This function is the same as CTRL Z except that with CTRL Z the attribute stays as unprotected. With this function, if protected mode is on then the attribute becomes protected. |
| ESC y | Clear display to spaces<br>Code: 027, 121 | This function will clear the entire screen to spaces, but will not change the character attributes that are associated to each character. |
| ESC Y | Clear from cursor to end of the display<br>Code: 027, 089 | This function is the same as ESC T, except that the screen will be cleared to the bottom right most boundary, end of screen |
| ESC t | Clear current line to spaces<br>Code: 027, 116 | This function follows the same rules as the ESC y, except that instead of clearing the entire screen. This function only clears the current character line. |
| ESC T | Clear from cursor to the end of the line<br>Code: 027, 084 | This function will clear all unprotected characters to spaces, on the current line, from the current cursor " X" to the end. If the protected mode is on then protected characters will also be cleared to spaces. |

| COMMANDS | FUNCTION | DESCRIPTION |
|---|---|---|
| ESCR | Delete an entire line<br>Code: 027, 082 | This function will delete the current line at cursor "Y". All data below this line will move up and the last line will be blank. |
| | **Line Scroll** | |
| ESCE | Insert line of space characters<br>Code: 027, 069 | This function will insert a line of space characters at the current vertical position. Data on this line and underneath will scroll downward. |
| ESCj | Move cursor up one line (scroll if at top)<br>Code: 027, 106 | This function will move the cursor up 1 line, if it is at the top of the screen will scroll down all the lines down, the bottom line will be lost and the top line will become black. |
| ESCO | Turn autoscroll on<br>Code: 027, 079 | This function enables autoscrolling, which simply means that when the bottom right most boundary is met, the screen will scroll up when the next printable character is recieved. |
| ESCN | Turn autoscroll off<br>Code: 027, 078 | This function will turn off the autoscroll mode. |

| COMMANDS | FUNCTION | DESCRIPTION |
|---|---|---|
| | **Set cursor position** | |
| ESC=XY | Move cursor to X,Y<br>Code: 027,061 | It will address the cursor to an X, Y location on display. |

**PARAMETER FORMAT :**

ESC = <COLUMN X> <ROW Y>

**RANGES :**

COLUMN X : "0"-"19"' (048-067).

ROW Y : "0"-"1"' (048-049).

**EXAMPLE :**

PRINT #1, CHR$(27)+"=51";

Set the cursor to the second line, the 5th cell.

**ATTENTION :**

For all cursor move commands, please make sure that your PRINT-command does not send a CR as terminator, which will also change the cursor position.

| | | |
|---|---|---|
| | **RS232 setting** | |
| ESC4 | Set baud rate<br>Code: 027,052 | This function simply changes the baud rate. |

**PARAMETER FORMAT :**

ESC 4 <BAUD>

**BAUD RANGES :** 30h - 3Ch

***Note : The BAUD "8" ~ "<" only available for JP-8003A***

| BAUD | ACTUAL RATE | BAUD | ACTUAL RATE |
|---|---|---|---|
| 0 | 9600 | 8 | 14400 |
| 1 | 300 | 9 | 28800 |
| 2 | 600 | : | 38400 |
| 3 | 1200 | ; | 57600 |
| 4 | 2400 | < | 115200 |
| 5 | 4800 | | |
| 6 | 9600 | | |
| 7 | 19200 | | |

*NOTE :*

*Please use "ESC 8" command to save this value into EEprom and it will be the default value next time 8003 boots up.*

| | | |
|---|---|---|
| ESC6 | Turn on echo RS232 data received<br>Code: 027,054 | These 2 functions allow you to enable and disable RS232 data loop back mode. In other words, |

| COMMANDS | FUNCTION | DESCRIPTION |
|---|---|---|
| ESC 7 | Turn off echo RS232 data received Code: 027, 055 | what you transmit you'll receive, but the LCD will still decode the data stream as normal. |
| ESC 5 | Request display type Code: 027, 053 | This function request the LCD display type, so that it is easy when powering up your Point of sale package, you may configure formats according to the returned string. RETURNS : Default value is "LX2" |
| | **Printer functions** | |
| ESC_P | Enable printer, disable display Code: 027, 095, 080 | If you have connected both a serial printer and 8003 display on the same COM port, you can use this command to start the access to the printer. After power on, only the display is active. If you send ESC_P, the printer mode will be activated, all data sent to this COM-port will be printed and not displayed. |
| ESC_D | Disable printer, enbale display Code: 027, 095, 068 | This command quits the printer mode and return to display mode. The following data will be displayed, not printed. |

**EXAMPLE:**

```
PRINT #1, CHR$(27) + "*";
PRINT #1, "Display";
PRINT #1, CHR$(27) + "_P"
PRINT #1, "Printer is active." + CHR$(13)
          + CHR$(13) + CHR$(13) + CHR$(13)
PRINT #1, CHR$(27) + "_D"
PRINT #1, "again."
```

| COMMANDS | FUNCTION | DESCRIPTION |
|---|---|---|
| | **LCD functions** | |
| ESCG | Set brilliance to level<br>Code: 027, 071 | This function will set the brilliance level to a value between 00 (OFF) and 32 (FULL ON). |

**PARAMETER FORMAT:**
ESCG<LEVEL>
**RANGES:**
LEVEL : 00 - 32 (048 - 080)
Note:
Please use "ESC 8" command to save this value into EEprom and it will be the default value next time 8003 boots up.

| | | |
|---|---|---|
| ESC 0 | Fade brilliance<br>(at rate * 8.88ms)<br>Code: 027, 048 | This function will fade brilliance from current level to level (?) at a rate * 8.88ms.<br>The dimming time of the brightness is between 0 sec. and 2.26 secs. |

**PARAMETER FORMAT:**
ESC 0 <RATE> <LEVEL VALUE>
**RANGES:**
RATE : 00 - 255 (000-255): 0 sec-2.265 sec
LEVEL VALUE : 0-32 (048 - 080)

| | | |
|---|---|---|
| ESC 1 | Increase brilliance<br>(at rate * 8.88ms)<br>Code: 027, 059 | This function will increase the brilliance from the current level until the level user specify.<br>User may make the rate of increase, time until each level of brightness, between 0 sec. and 2.26 secs. |

**PARAMETER FORMAT:**
ESC 0 <RATE> <LEVEL VALUE>
**RANGES:**
RATE : 00 - 255 (000-255): 0 sec-2.265 sec
LEVEL VALUE : 0-32 (048 - 080)

| COMMANDS | FUNCTION | DESCRIPTION |
|---|---|---|
| ESC^ | Adjust contrast to level<br>Code: 027, 094 | This function will set the contrast level to a value between 00 (OFF) and 32 (FULL ON). The contrast is character darkness ! |
| Note: For JP-8003P only | | |

**PARAMETER FORMAT:**
ESC^<LEVEL>
**RANGES:**
LEVEL : 0 - 32 (048 - 080)

| | | |
|---|---|---|
| ESC2 | Fade contrast down (at rate * 8.88mS)<br>Code: 027, 050 | This function will dim the contrast from the current level until it is at the level (?) you specify. You may make the rate of dimming, time until each level of character darkness, between 0 sec. and 2.26 secs. |
| Note: For JP-8003P only | | |

**PARAMETER FORMAT:**
ESC2<RATE><LEVEL VALUE>
**RANGES:**
RATE : 00 - 255 (000-255): 0 sec-2.265 sec
LEVEL VALUE : 0-32 (048 - 080)

| | | |
|---|---|---|
| ESC3 | Increase contrast up (at rate * 8.88mS)<br>Code: 027, 051 | This function will increase the contrast from the current level until it is at the level (?) you specify. You may make the rate of increase, time until each level of character darkness, between 0 sec. and 2.26 secs. |
| Note: For JP-8003P only | | |

**PARAMETER FORMAT:**
ESC3<RATE><LEVEL VALUE>
**RANGES:**
RATE : 00 - 255 (000-255): 0 sec-2.265 sec
LEVEL VALUE : 0-32 (048 - 080)

| COMMANDS | FUNCTION | DESCRIPTION |
|---|---|---|
| ESC 8 | Save brilliance & contrast to eeprom Code: 027, 056 | This function allows you to store the current brilliance, level of brightness, contrast, level of character darkness, to an on board EEPROM. |
| | **Misc. cursor and LCD functions** | |
| ESC` CTRL A | Turn cursor off Code: 027,096,001 | This function is same as ESC W, self explanatory |
| ESC` CTRL B | Turn cursor on Code: 027,096,002 | Self explanatory |
| ESC` CTRL C | Turn cursor blink on Code: 027,096,003 | Self explanatory |
| ESC` CTRL D | Turn cursor blink off Code: 027,096,004 | Self explanatory |
| ESC` CTRL G | Turn LCD screen off Code: 027,096,007 | Self explanatory |
| ESC` CTRL H | Turn LCD screen on Code: 027,096,008 | Self explanatory |
| | **User definable characters** | |
| You can program up to 8 user-defined characters. You can use this for foreign characters or for a little graphics. First, you define character matrix. Then you activate the special character mode. To the display the characters, then you use ASCII 001 to 008(CTRL-A etc.). As the last step, you return from special character mode to the normal mode. | | |
| ESC` CTRL I | Program special character Code: 027,096,009 | This function allows the programmer to define 1 of 8 user definable characters. These characters are for those |

| COMMANDS | FUNCTION | DESCRIPTION |
|---|---|---|
| | | whom wish to use variations of characters. There are only 8 characters limited. |

**PARAMETER FORMAT:**
ESC ` CTRL I <*CHARACTER*> <8 * *CHARACTER DATA*>

**RANGES:**

CHARACTER : "1" - "8" (049 -056)
CHARACTER DATA : 0 - 255 (000-255).
This is the definition of the character.

| Draft | Binary | Hex.(dez.) | ASCII-Code |
|---|---|---|---|
| ***** | 11111 | $1F (31) | "O"= CHR$(48+31) |
| * * | 10001 | $11 (17) | "A"=CHR$(48+17) |
| * * | 10001 | $11 (17) | "A"=CHR$(48+17) |
| * * | 10001 | $11 (17) | "A"=CHR$(48+17) |
| * * | 10001 | $11 (17) | "A"=CHR$(48+17) |
| * * | 10001 | $11 (17) | "A"=CHR$(48+17) |
| * * | 10001 | $11 (17) | "A"=CHR$(48+17) |
| ***** | 11111 | $1F (31) | "O"= CHR$(48+31) |

**EXAMPLE: (for the special character above)**

```
PRINT#1,CHR$(27)+CHR$(96)+CHR$(9)+"1"+"OAAAAAAO" ':   :REM Define
PRINT#1,CHR$(27)+"H"+CHR$(002)      :REM Turn on special character mode
PRINT #1, CHR$(1)                        :REM  Display this character
PRINT#1,CHR$(27)+"H"+CHR$(003)      :REM Turn off special character mode
```

| | | |
|---|---|---|
| ESC H CTRL B | Turn on special character mode Code: 027,072,002 | The special character mode allows you to use the 8 user definable characters, you also must add DEC 32 or Hex 20 to the character. |
| ESC H CTRL C | Turn off special character mode Code: 027,072,003 | Return the display to normal display mode. |

| COMMANDS | FUNCTION | DESCRIPTION |
|---|---|---|
| | **Macro programming** | |
| You can define up to 32 function blocks (macros) with a length of up to 127 byte. You can exceed the length of 127 bytes if you take care that you do not use the following function block which would overwrite the data of the last one. | | |
| ESC " | Program an executable function block Code: 027,034 | This function allows the programmer to program a sequence of function calls, control or escape sequences and even text. The ability to execute series of functions with a single call is useful for repetitive function formats such as those in the retail industry.<br><br>There are 32 usable blocks all of which, if you desire, may be linked.<br><br>If you exceed the 127 byte size of the blocks, the display will directly link your block to the next consecutive block.<br><br>This is only useful upon power up, because if you have data in the next block, its data will be overwritten. Therefore it is suggested these blocks are unutilized as one of the first steps to use the LCD display.<br><br>To link blocks, simply add a function call, from within a block, to the block that you wish to use. |

| COMMANDS | FUNCTION | DESCRIPTION |
|---|---|---|

**PARAMETER FORMAT:**
ESC "<BLOCK-NO><DATA and/or CTRL/ESC SEQUENCES><ESC EOT>
**RANGES:**
BLOCK-NO : 0 - 31 (048 - 079)
ESC EOT: 027, 004.
**EXAMPLE:**

```
PRINT #1, CHR$(27)+CHR$(34)+'0';
PRINT #1, CHR$(27)+'*'
PRINT #1, CHR$(27)+'Macro#0.';
PRINT #1, CHR$(27)+CHR$(4)
```

| | | |
|---|---|---|
| ESC $ | Pause (for a multiple of 8.88ms)<br>Code: 027,036 | This function allows the programmer to stop the pole display for a period of time, The delay is in multiple of 8.88ms. |

**PARAMETER FORMAT:**
ESC $ <DELAY>
**RANGES:**
DELAY : 0 - 255 (000-255) (0 sec - 2.26 secs).

| | | |
|---|---|---|
| ESC # | Execute a programmed function block<br>Code: 027,035 | This function allows the programmer to call one of the function blocks for execution. |

**PARAMETER FORMAT:**
ESC # <BLOCK-NO>
**RANGES:**
BLOCK-NO : 0 - 31 (048 - 079)
**EXAMPLE :** Starts the Block-NO, that has been defined in the example above.

```
PRINT #1, CHR$(27)+'#0'
```

| | | |
|---|---|---|
| ESC / | Display until terminated by carriage return. | This function will display data received until it gets an ENTER. This is really only used within programmable function block for variable text message. |

**EXAMPLE:**

TOTAL: $ 10.75
TOTAL: $ 13.87
TOTAL: $ 5000.00

In this example you would protected the " TOTAL: $" then use an ESC / within a function block, to get the total account as this operation will be repetitive.

## Additional new commands for JP-8003A only

**ESC 9**: Define String for "request display type" command
**Format: ESC 9 "String"**
**Description:**

1. The length of string: 1~16 charecter
2. User must add ASC (13) as a terminator at the end of string, if the length of this string is less than 16.
3. User can use this command to define the string, which 8003A can echo back to HOST by sending the command ESC 5 (Request display type, see previous manual) , the default string is "LX2" after define the string (1-16 bytes) , user also must send the command "ESC 8" to save data to the EEPROM.

**ESC A:** Define for the boot up message at the 1st line
**Format: ESC A "String"**
**Description:**

1. The length of this string must be fixed at 20 Bytes.
2. Use this command to define the boot up message on the top line of display. Default message is "*** Jarltech Display *" also use command "ESC 8" to save this setting to EEPROM.

**ESC B:** User Define for boot up message at the 2nd line
**Format: ESC B "String"**
**Description:**

1. The length of this string must be fixed at 20 Bytes
2. Use this command to define the boot up message on the 2nd line of display. Default message is: "Model : JP-8003A" also use command "ESC 8" to save data to EEPROM.

**ASC (80H):** Symbol for Europe Dollar
**Description:**

Send code ASC (80H) to JP-8003A, it'll display to screen the Europe Dollar character.This character was saved in the special Character Memory of LCD (position 8). Please don't re-define this special character (8) of "Europe dollar character",if you don't use it.

**ESC D:** Write Default Value to EEPROM

**Description:**

Send this command, all parameters on JP-8003A will back to the default value.

| Default | Value |
|---|---|
| Baud Rate | 9600 |
| Brilliance | Level 32 |
| Contrast | Level 16 |
| Special Character (8) | Europe dollar |
| Display type message | LX2 |
| Welcome message | Line 1= * Jarltech Display *<br>Line 2= Model : JP-8003A |

**ESC 8:** Will save following value to EEPROM

**Description:**

The values are Baud Rate, Contrast, Brilliance, 8 Special Programmed Characters, Display Type Message and 2 lines of Welcome Message.

# A Appendix I

## Specifications

**Display**

- Type: alphanumeric dot matrix LCD with LED backlight
- Text mode: 20 characters x 2 lines
- Character size: 5.9(W) x 12.7mm(H), 5x8 dots
- effective view area: 147mm (W) x 35mm (H)

**Case**

- Dimension: 200mm(L) x 87mm(W) x42mm(H)
- Adjustable angle: 360° swivl and 40° forward/backward
- Pole height: 285mm
- Material: ABS

**Bottom Plate**

- ABS with metal plate. Screw holes available for pole fixed

**Interface**

- RS232C

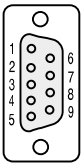
**Power Requirement**

- RS232 interface: +12V DC directly from host or through adaptor from external 110/220V AC source with polarity as follows: 
- Consumption: 430mA

**Programming**

- More than 50 control sequences and escape sequences for powerful programming such as move cursor, cursor home, cursor blink, clear, delete, reset, scrolling, program function block, pause, set protects mode, brilliance, contrast, insert line and special character mode.

**Connector Pin assignments**




**DB9 (Female)**
#2: TX
#3: RX
#5: GND
#7: CTS
#8: RTS
#9: +12V DC
#1, #4, #6: Short together

# A Appendix II

## Sample Program

```
10   'Sample program for Jarltech 8003 series
20   'Copyright(C) Jarltech international INC. 1994-1996
30   '
40   RESET:CLEAR:SCREEN 0:COLOR 7,0:CLS
50   '
60   OPEN "COM1:9600,N,8,1,DS0,CS0" AS #1
70                                    ': Set RS232 Options
80   E$=CHR$(27)
90                                    ': Set E$="[ESC]"
100  '
110  'Set line scrooling stop, Clear Screen, Set cursor off
120  PRINT#1,E$+"%0"+E$+"%1"+E$+";"+E$+"W"
130  '
140  emo program
150  '
160  LOCATE 7,20:PRINT "Testing Jarltech 8003 Series..."
170  LOCATE 8,35:PRINT "[ESC] to Stop..."
180  '
190  'Main  ************************************
200  'Program an Executable Function Block
210  'Syntax: ESC"<BLOCK><DATA and/or CTRL/ESC SEQUENCES>ESC<EOT>
220  'Ranges:
230  '   <BLOCK>: 00-31 (48 DEC - 79 DEC) ®  ASCII: 0 ... O
240  '    <EOT>  : 04 DEC
250  '
260  ED$=E$+CHR$(34)               ': Set ED$= ESC"
270  EN$=E$+CHR$(4)                ': Set EN$= ESC<EOT>
280  '
290  PRINT  #1,ED$+"0"+CHR$(10)+CHR$(10)+CHR$(30)+CHR$(22)
     + "Jarltech 8003 Demo.."+EN$
300  PRINT #1,ED$+"1"+CHR$(10)+CHR$(13)+"Testing Control ..."+EN$
310  PRINT#1,ED$+"2"+CHR$(10)+CHR$(13)+"Command OK!"+EN$
320  PRINT #1,ED$+"3"+CHR$(10)+CHR$(30)+"Testing [ESC] ..."+EN$
330  PRINT#1,ED$+"4"+CHR$(22)+" : Cursor Off/On"+EN$
```

```
340  PRINT#1,ED$+"5"+E$+"`"+CHR$(2)+EN$
350  '
360  DELAY=2                          'Set Delay Time=2 sec
370  '
380  'Execute Programmed Function Block
390  'Syntax: ESC#<BLOCK>
400  'RANGES:
410  '    <BLOCK>: 00-31 (48 DEC - 79 DEC)
420  '
430  FOR I=48 TO 53:PRINT #1,E$+"#"+CHR$(I):GOSUB 710:NEXT
440  PRINT#1,E$+"W"
450  '
460  'Test Fade/Increase Brilliance
470  '
480  PRINT#1,"Fade Brilliance..."
490  PRINT#1,E$+"0"+CHR$(50)+"0"
500  PRINT#1,"Increase Brilliance"
510  PRINT#1,E$+"1"+CHR$(50)+"P"
520  DELAY=12:GOSUB 710
530  '
540  'Test Fade/Increase Contrast
550  '
560  PRINT#1,"Fade Contrast....."
570  PRINT#1,E$+"2"+CHR$(100)+"0"
580  PRINT#1,"Increase Contrast "
590  PRINT#1,E$+"3"+CHR$(100)+"P"
600  DELAY=12:GOSUB 710
610  PRINT #1,E$+";"+E$+"^<"  'Clear Screen & set Contrast to DEC 60
620  '
630  '
640  'Program a Message For The Scrolling Lines & Start a Line Scrolling
650  '
660  PRINT#1,E$+";"+E$+"(1*Nice to see you!*"+CHR$(13)+E$+")0091"
670  PRINT #1,E$+"(2*This is JARLTECH 8003 series DEMO*"+CHR$(13)+E$+")1192"
680  '
690  END ************************************
700  '
710  'timer delay
720  T=INT(TIMER)
730  Y$=INKEY$:IF Y$=CHR$(27)THEN END
740  IF T+DELAY>TIMER THEN 730 ELSE RETURN
750  '
```

# A Appendix III

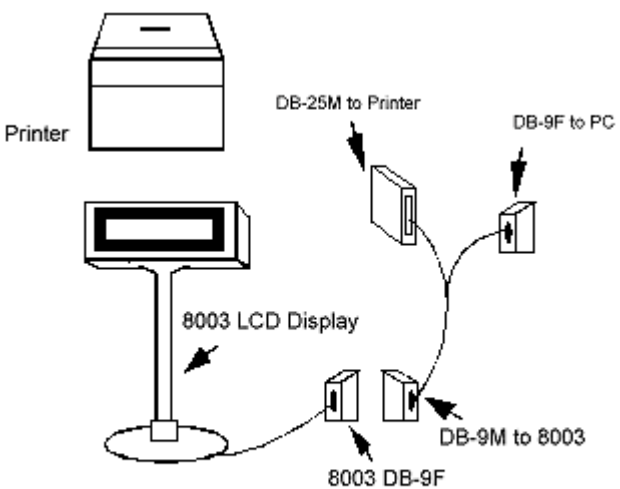
## How to Attach Serial Receipt Printer

### Installations

Please refer to the installation in Chapter 2 of 8003 operation manual. Connect the Y cable (GC-8003P) to serial RS232 port, receipt printer, and 8003 display. (Please refer to figure as below)

***Please make sure the pin out of interface are matched between receipt printer and DB-25M connector.***

***NOTE:***




The Installation of 8003 Display and Receipt Printer

GC-8003P Y cable pin out:

| DB-9F to PC | DB-9M to 8003 | DB-25M to Printer | SIGNAL |
|---|---|---|---|
| 2 | 4 | 2 | RX |
| 3 | 3 | - | - |
| 5 | 5 | 7 | GND,Shield |
| 6 | 8 | - | DSR |
| 9 | 9 | - | Vcc |
| - | 2 | 3 | TX |
| - | 7 | 20 | DTR |

## Command codes

There are output exchange command codes.
ESC_D: To select to print out the data from display.
ESC_P: To select to print out the data from printer.

***NOTE:*** ***Character "D" & "P" must be capital initial.***

## Command code explanation

If you send<ESC> <_> <D> <Hello> then the pole display will display “Hello".
If you send <ESC> <_> <P> <Hello> then the printer will print out “Hello".

***NOTE:*** ***Before sending printer commands, please make sure that you have already connected printer to 8003. Otherwise if you sent <ESC> <_> <P> <Hello> that will make your 8003 data buffer full, and then after you connect printer to 8003 it will work properly again.***

## Sample test program (this program is written by QBASIC)

```
100     Open "COM1:9600,N,8,2,CD,DS,CS" for output AS#1
110     Read A$, B$
120     Rem : [ ESC + "*" + ESC + "+" ]|> clear display & cursor
home
130     Print #1, CHR$(27); CHR$(42); CHR$(27); CHR$(43)
140     Rem : Initialize the display
150     Print #1, CHR$(27); CHR$(95); CHR$(68)
160     Print #1, A$
170     Rem : Initialize the printer
180     Print #1, CHR$(27); CHR$(95); CHR$(80)
190     Print #1, B$
200     CLOSE
210     END
220     DATA ¡§  WELCOME  ¡¨, ¡§  Jarltech 8003  ”
```

***Note:***

***Please refer to the control codes of your printer manual for else printer function.***

# A Appendix IV

## Ultimate Emulation Pole Display (JP-8003U only)

If you are using Jarltech pole display model 8003U which is with Ultimate emulation, please refer to the following commands to switch the function between Jarltech standard and Ultimate emulation.

CTRL + "Z" : from Ultimate version to Jarltech standard version.
CTRL + "q" : from Jarltech standard version to Ultimate version.
CTRL + "_" : from any mode switch to Ultimate version.

**ISSUED:MAR.2000-V5.1**