Blue Bamboo
P25 Development Guide

Firmware version: 1.0.58 (or higher)

Document version: V 3.4.1

Document date: 2011-09-22

Products: P25 / P25-M / P25i / P25i-M

Headquarters
Blue Bamboo Transaction Systems (Shanghai) Co., Ltd.
6 Floor, Huishan Technology Park, No. 3333 Changning Road,
Shanghai, P.R. China 200336
Tel: +86 21 5219 0333
Fax:+86 21 5219 0603
www.bluebamboo.com

Table of Contents

1 Introduction

1.1 Purpose

This document is a guide for the basic application development of the P25 Printer product family.

1.2 Scope

This document consists of the communication options and software protocols for communication with the P25.

This document is intended for use by engineers who will develop applications based on the P25 printer. **This document does not provide detailed hardware and software specifications for the P25.** The P25 User Guide contains information on the hardware specifications for the P25, and the P25 Developer's User Guide contains information on the software specifications of the P25 Developer program.

2 ESC commands

Support command list: Customers can click page number directly to view detail command information.

2.1 Print Commands

The P25 printer supports the following commands for printing characters and feeding paper:

Command	
LF, CR	Print and feed line
ESC J	Print and feed paper
ESC d	Print and feed *n* lines

LF

[Name]	Print and line feed	
[Format]	ASCII	LF
	HEX	0A
	Decimal	10
[Description]	Print the data in the print buffer and feed one line based on the current line spacing.	
[Note]	Set the print position to the beginning of the line.	
[Reference]	**ESC 2, ESC 3**	

CR Same as LF, ref LF

ESC J

[Name]	Print and feed paper
[Format]	ASCII ESC J n HEX 1B 4A n Decimal 27 74 n
[Range]	0 ≤ n ≤ 255
[Description]	Print the data in the print buffer and feed paper [n x (vertical or horizontal motion unit)] inches.

ESC d

[Name]	Print and feed *n* lines
[Format]	ASCII ESC d n HEX 1B 64 n Decimal 27 100 n
[Range]	0 ≤ n ≤ 255
[Description]	Print the data in the print buffer and feed *n* lines.
[Note]	1) This command defines the print starting position to the beginning of the line. 2) This command does not affect the line spacing set by ESC 2 or ESC 3.
[Reference]	**ESC 2, ESC 3**

GS k
d1...dn

[Name] Print bar code(one dimension)

[Format]

ASCII	GS	k	m	n	d1 … dn
HEX	1D	6B	m	n	d1 … dn
Decimal	29	107	m	n	d1 … dn

[Range]
0 ≤ m ≤ 4 , m = 0x49
if m=2 then n=0x0d (ean-13)
if m=3 then n=0x08 (ean-8)
if m=0 then n=0x0c (upc-a)
if m=1 then n=0x08 (upc-e)
if m=0x49 then n variable (code128) (since version 1.0.34)

[Description] Print bar code
Eg:
1d 6b 02 0d 36 39 30 31 32 33 34 35 36 37 38 39 32 ; ean-13
1d 6b 03 08 36 39 30 31 32 33 34 31; ean-8
1d 6b 00 0c 30 30 31 32 33 34 35 36 37 38 39 35; upc-a
1d 6b 01 08 30 30 31 32 33 34 35 37; upc-e”
1d 6b 49 03 41 49 4d; code128

[Note]

[Reference] **ESC 2, ESC 3**

[Name] Print pdf417 bar code(two dimension)

[Format]

HEX
Expand 3 times, 3 data symbols per row
1D 6B 10 col(2 Bytes) row(2 Bytes) len(2 Bytes) c1 … cn
Or
Expand 2 times, 7 data symbols per row
1D 6B 11 col(2 Bytes) row(2 Bytes) len(2 Bytes) c1 … cn

[Range]
0 ≤ col ≤ 3(1D 6B 10 …),
0 ≤ col ≤ 7(1D 6B 11 …),
data words number per row, 0 means auto select.
0 ≤ row ≤ 90, row number, 0 means auto select.
0 ≤ len ≤ 500, characters number.
Col, row, len all big endian, that is 3 should be 0x00 0x03.

[Description] Print pdf417 bar code
Eg:
"\x1d\x6b\x10\x00\x00\x00\x00\x00\x1f""Hello, world! A PDF417 example."

[Note]

[Reference]

2.2 Line Space Commands

The P25 printer supports the following commands for setting line space. These commands can only be used for specifying line space, not actually feeding paper. The line space set through these commands affects the results of **LF** and **ESC d** and paper feeding by using **FEED**.

Command

ESC 2	Select default line space
ESC 3	Set line space

ESC 2

[Name]	Select default line space
[Format]	ASCII ESC 2 HEX 1B 32 Decimal 27 50
[Description]	Select 1/7 inch line (approx. 3.75mm) space
[Reference]	**ESC 3**

ESC 3

[Name]	Set line space
[Format]	ASCII ESC HEX 1B 33 n Decimal 27 51 n
[Range]	0 ≤ n ≤ 255
[Description]	Set the line space to [n x vertical or horizontal motion until] inches.
[Note]	1) The horizontal and vertical motion unit is specified by GS P. Changing the horizontal or vertical motion unit does not affect the current line space. 2) The GS P command can change the horizontal (and vertical) motion unit. However, the value set by this command cannot be less than the minimum vertical movement amount set by GS P, and it must be in even units of the minimum vertical movement amount.
[Reference]	**ESC 2, GS P**

2.3 Character Commands

The printer supports the following commands for setting character font and size:

Command	
ESC K	Select ACP or UTF8 coding
ESC R	Select an international character set
ESC !	Select print mode
ESC-	Turn underline mode on/off
GS !	Select character size
GS B	Turn white/black reverse printing mode on/off

ESC K

[Name]	Select ACP or UTF8 coding
[Format]	ASCII
	HEX 4B n
	Decimal 75 n
[Range]	0x30, 0x31
[Description]	Select ACP or UTF8 coding
0x30	ACP coding
0x31	UTF8 coding, deprecated, ref ESC R command.
[Default]	ACP coding
[Note]	This command should be placed previous than any normal printing characters in a line.

ESC R

[Name] Select an international character set.

[Format] ASCII

HEX 52 n

Decimal 82 n

[Range] 0 , 30,65

[Description] Select an international character set *n* from the following table.

n(hex)	Character set
0	(ISO/IEC 8859-15)Latin Character (include English, France, Germany, Spain…)
30	Simplified Chinese
65	UTF8(Since P25 Version S1.0.41)

[Default] n = 0

ESC !

[Name] Select print mode.

[Format]

ASCII	ESC	!	n
HEX	1B	21	n
Decimal	27	33	n

[Range] 0 ≤ n ≤ 255

[Description] Select print mode(s) using n as follows.

Bit	Off/On	Value	Function
0		0	Select 32 dot font (24 char/per line) eg. 1B 21 00
	--	1	Select 24 dot font (36 char/per line) eg. 1B 21 01
1	--	--	undefined
2	--	--	undefined
3	--	--	undefined
4	off	0	Double-height mode not selected
	on	1	Double-height mode selected
5	off	0	Double-width mode not selected
	on	1	Double-width mode selected
6	--	--	undefined
7	off	0	Underline mode not selected
	on	1	Underline mode selected

[Note]

1. When both the double-height and double-width modes are selected, quadruple size characters will be printed.
2. The printer can underline all characters, but cannot underline the space set by **HT**.
3. The thickness of the underline is that selected by ESC –, regardless of the character size.
4. ESC – can also turn on or off underline mode. However, the setting of the last received command is effective.

5. GS ! can also select character size. However, the setting of the last received command is effective.

6 This command should be placed previous than any normal printing characters in a line.

[Reference] **ESC -, GS !**

ESC –

[Name]	Turn underline mode on/off			
[Format]	ASCII	ESC	-	n
	HEX	1B	2D	n
	Decimal	27	45	n
[Range]	0 ≤ n ≤2 (or 48 ≤ n ≤ 50)			
[Description]	Turns underline mode on or off, based on the following values of n;			

n	Function
0(or 48)	Turns off underline mode
1(or 49)	Turns on underline mode (1 dot thick)
2(or 50)	Turns on underline mode (2 dots thick)

[Note]

1 The printer can underline all characters (including right-side character spacing), except for the space set by HT.

2 The printer cannot underline white/black inverted characters.

3 When underline mode is turned off by setting the value “n” to 0 or 48, the following data is not underlined, and the underline thickness set before turning off does not change. The default underline thickness is 1 dot.

4 Changing the character size does not affect the current underline thickness.

5 Underline mode can also be turned on or off by using ESC !. However, the last received command is effective.

[Default] n = 0

[Reference] **ESC !**

GS !

[Name] Select character size

[Format]

ASCII	GS	!	n
HEX	1D	21	n
Decimal	29	33	n

[Range] 0 ≤ n ≤ 255

[Description] Select the character height using bits 0 to 3, and select the character width using bits 4 to 7, as follows;

Bit7-Bit4(Hex)	Character width	Bit3-Bit0(Hex)	Character height
0	1(normal height)	0	1(normal width)
1	2(double height)	1	2(double width)
2	3	2	3
3	4	3	4
4	5	4	5
5	6	5	6
6	7	6	7
7	8	7	8

[Note]

1 This command is effective for all characters.

2 If n is outside of the defined range, this command is ignored.

3 The ESC ! command can also turn double width and double height modes on or off.

4 This command should be placed previous than any normal printing characters in a line.

[Default] n = 00(HEX)

[Reference] **ESC !**

GS B

[Name] Turn white/black reverse printing mode On/Off.

[Format]

ASCII	GS	B	n
HEX	1D	42	n
Decimal	29	66	n

[Range] 0 ≤ n ≤ 255

[Description] Turn white/black reverse printing mode On/Off.

1 When LSB is 0, white/black reverse printing mode is turned off.

2 When LSB is 1, white/black reverse printing mode is turned on.

[Note]

1 Only the lowest bit of *n* is valid.

2 This command is available for built-in and user-defined characters.

3 When white/black reverse printing mode is on, it also applies to character space set by ESC SP.

4 This command does not affect the space between lines.

5 White/black reverse mode has higher priority than underline mode. Even if underline mode is on, it is disabled (but not canceled) when white/black reverse is on.

2.4 Keypad Button Commands

The printer supports the following commands for enabling and disabling the keypad buttons.

Command	
ESC c 5	Enable/disable keypad buttons

ESC c 5

[Name]	Enable/disable keypad buttons				
[Format]	ASCII	ESC	c	5	n
	HEX	1B	63	35	n
	Decimal	27	97	53	n
[Range]	0 ≤ n ≤ 255				
[Description]	Enable or disable the keypad buttons.				
	When the LSB is 0, it enables the keypad button				
	When the LSB is 1, it disables the keypad button				
[Note]	If keypad buttons are disabled, all buttons in panel are disabled (including POWER button, in other words, user could not press key to power off P25, but user could use command to power off P25).				

2.5 Print Position Commands

The printer supports the following commands for setting the print position.

Command	
ESC $	Set absolute print position
ESC a	Select justification
HT	Horizontal tab
ESC D	Set horizontal tab positions
GS L	Set left margin

ESC $

[Name] Set absolute print position

[Format]

ASCII	ESC	$	nL	nH
HEX	1B	24	nL	nH
Decimal	27	36	nL	nH

[Range] 0 ≤ nL ≤ 255, 0 ≤ nH ≤ 255

[Description] Set the distance from the beginning of the line to the position where subsequent characters are to be printed.

Note

1. The distance between the beginning of the line to the print position is [(nL + nH x 256) x (vertical or horizontal motion unit)] inches.
2. Setting outside the specified printable area is ignored.
3. The horizontal and vertical motion unit are specified by GS P.
4. The GS P command can change the horizontal and vertical motion unit. However, the value cannot be less than the minimum horizontal movement amount, and it must be in even units of the minimum horizontal movement amount.
5. This command should be placed previous than any normal printing characters in a line.

[Reference] ESC\, GS$, GS\, GS P

ESC a

[Name] Select justification

[Format]

ASCII	ESC	a	n
HEX	1B	61	n
Decimal	27	97	n

[Range] 0≤ n ≤ 2 , 48 ≤ n ≤ 50

[Description] Aligns all the data in one line to the specified position, *n* selects the type of justification as follows

[Note] This command should be placed previous than any normal printing characters in a line.

N	Justification
0, 48	Left justification
1, 49	Center justification
2, 50	Right justification

[Note]

1 The command is enabled only when processed at the beginning of the line.

2 This command formats the justification in the printing area.

3 This command justifies the space area according to HT, ESC $ or ESC \ [Default] n = 0

HT

[Name]	Horizontal Tab	
[Format]	ASCII	HT
	HEX	09
	Decimal	9
[Description]	Moves the print position to the next horizontal tab position.	

[Note]

1 This command is ignored unless the next horizontal tab position has been set.

2 If the next horizontal tab position exceeds the printing area, the printer sets the printing position to [Printing area width + 1]

3 Horizontal tab positions are set with ESC D.

4 If this command is received when the printing position is at [Printing area width + 1], the printer executes print buffer-full printing of the current line and horizontal tab processing from the beginning of the next line.

5 The default setting of the horizontal tab position for the paper roll is every 0 characters.

[Reference] ESC D

ESC D

[Name]	Set horizontal tab positions.				
[Format]	ASCII	ESC	D	n1...nk	NUL
	HEX	1B	44	n1…nk	00
	Decimal	27	68	n1…nk	0
[Range]	0 ≤ k ≤ 32				
[Description]	Set horizontal tab position				

[Note]

1 *n* specifies the column number for setting a horizontal tab position from the beginning of the line.

2 *k* indicates the total number of horizontal tab positions to be set.

3 The horizontal tab position is stored as a value of [character width x n] measured from the beginning of the line. The character width includes the right-side character spacing, and double-width characters are set as twice the width of normal characters.

4	This command cancels the previous horizontal tab settings.
5	If you set n=8, the print position is moved to column 9 by sending HT.
6	Up to 32 tab positions (k=32) can be set. Data exceeding 32 tab positions is processed as normal data.
7	Transmit [n]k in ascending order and place an NUL code 0 at the end.
8	When [n]k is less than or equal to the preceding value [n]k-1, tab setting is finished and the following data is processed as normal data.
9	ESC D NUL cancels all horizontal tab positions.
10	The previously specified horizontal tab positions do not change, even if the character width changes.
[Default]	The default tab positions are at intervals of 0 characters.
[Reference]	HT

GS L

[Name]	Set left margin.
[Format]	ASCII GS L nL nH HEX 1D 4C nL nH Decimal 29 76 nL nH
[Range]	0 ≤ nL ≤ 255, 0 ≤ nH ≤ 255
[Description]	Set the left margin using nL and nH.
[Note]	1. The left margin is set to [(nL+nHx256)] x (horizontal motion unit) inches. 2. This command is effective only processed at the beginning of the line. 3. If the setting exceeds the printable area, the maximum value of the printable area is set. 4. The horizontal and vertical motion units are specified by GS P. Changing the horizontal and vertical motion unit does not affect the current left margin. 5. The horizontal motion unit (x) is used for calculating the left margin. The calculated result is truncated to the minimum value of the mechanical pitch. 6. This command should be placed previous than any normal printing characters in a line. 7. Printing content should less than one line.
[Default]	nL = 0, nH = 0
[Reference]	GS P

2.6 Bit-Image Commands

The printer supports the following bit-image command:

Command

ESC * Print bit image vertical mode
ESC X Print bit image horizon mode
ESC f Print downloaded bit image
ESC * m nL nH d1 dk

[Name] Print bit-image vertical mode

[Format]

ASCII	ESC	*	m	nL nH	d1…dk
HEX	1B	2A	m	nL nH	d1…dk
Decimal	27	42	m	nL nH	d1…d

[Range]
m = 0,1,32,33
0 ≤_nL ≤ 255
0 ≤ nH ≤ 3
0 ≤ d ≤ 255

[Description] Select a bit-image mode using *m* for the number of dots specified by nL and nH, as follows:

m	mode	Vertical Direction				Horizontal Direction			
		Number of Dots		Dot Density		Dot Density		Number of Data	
0	8 dot		single	8	60	DPI	90	DPI	nL+nH x256
1	8	dot	double	8	60	DPI	180	DPI	nL+nH x256
32	24 dot		single	24	180DPI	90	DPI	(nL+nHx256)x 3	
33	24	dot	double	24	180	DPI	180	DPI	(nL+nHx256)x3

[Note]

1 If the values of *m* are out of the specified range, nL and data following are processed as normal data.
2 The nL and nH indicate the number of dots of the bit image in the horizontal direction.
3 The number of dots is calculated by nL + nH x 256.
4 If the bit-image data input exceeds the number of dots to be printed on a line, the excess data is ignored.

5 *d* indicates the bit-image data. Set a corresponding bit to 1 to print a dot or to 0 to not print a dot.

6 If the width of the printing area set by GS L is less than the width required by the data sent with the ESC * command, the following will be performed on the line in question (but the printing cannot exceed the maximum printable area).The width of the printing area is extended to the right to accommodate the amount of data. If step does not provide sufficient width for the data, the left margin is reduced to accommodate the data.

7 After printing a bit image, the printer returns to normal data processing mode.

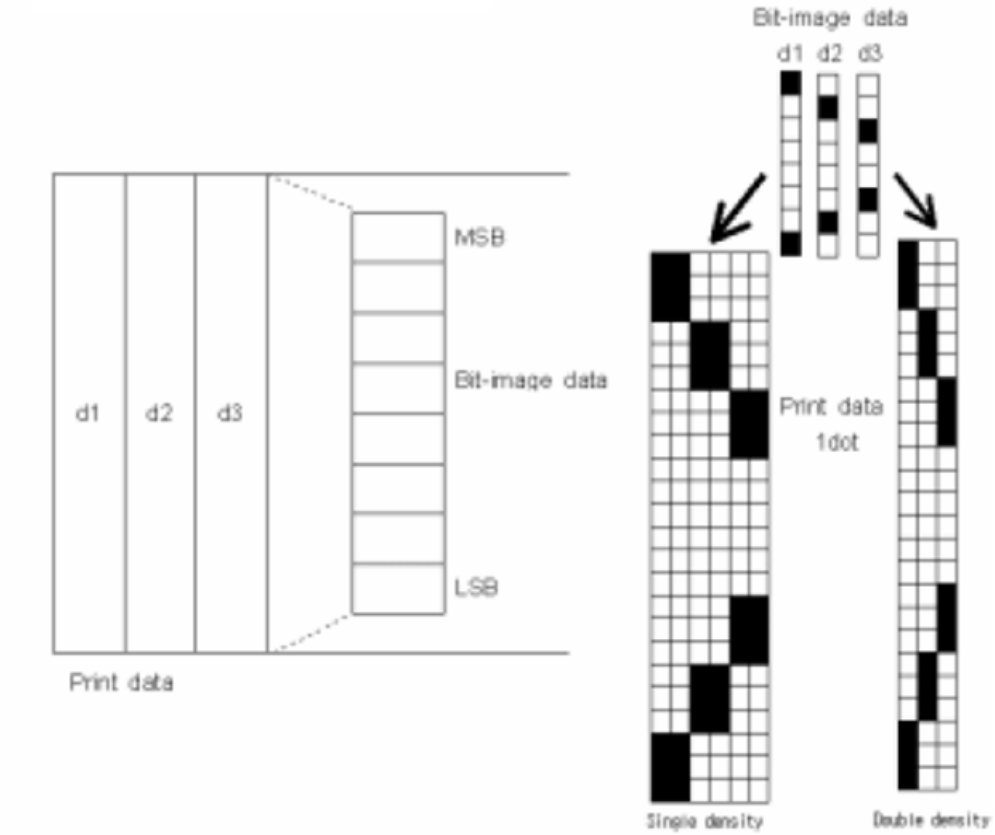
8 This command is not affected by print modes (emphasized, double strike, underline, character size or white/black reverse printing), except upside-down printing mode.

9 The relationship between the image data and the dots to be printed is as follows:

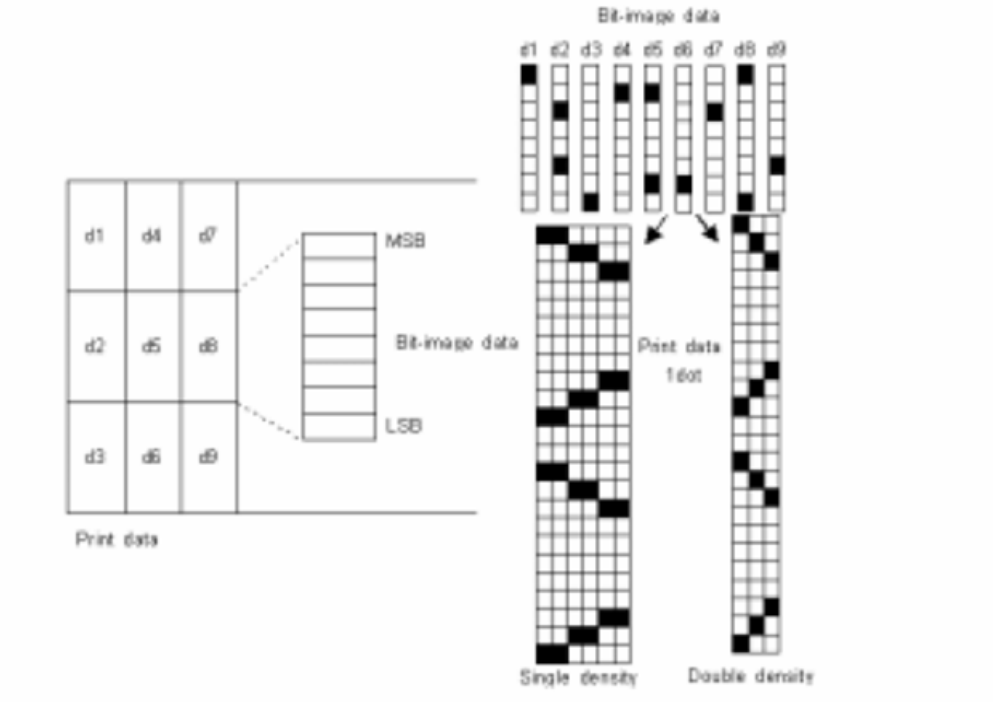
10 [d1…dk] is the data.

11 Deprecated, better use horizon mode.

_ When 8-dot bit image is selected



_ When 24-dot image is selected



ESC X

X 1 x y d1…dk
X 4 x y d1…dk

[Name] Print bit-image horizon mode

[Format]

ASCII	ESC	X	1	x	y	d1...dk
	ESC	X	4	x	y	d1...dk
HEX	1B	58	31	x	y	d1…dk
	1B	58	34	x	y	d1…dk
Decimal	27	88	49	x	y	d1…dk
	27	88	52	x	y	d1…dk

[Description] ESC X 1 x y d1 ... d(x*y) print bit image using x*8 dots in the horizontal direction and y dots in the vertical direction.

ESC X 4 x y d1 ... d(x*y) will double both its horizon and vertical size

- Horizontal direction dots = (x * 8)dots
- Vertical direction dots = (y)dots

GS v

GS v 0 m XL XH YL YH d1…dk

[Name] Print bit-image horizon mode

[Format]

ASCII	GS	v	0	m	XL	XH	YL	YH	d1…dk
HEX	1D	76	30	m	XL	XH	YL	YH	d1…dk

[Description] print bit image using x*8 dots in the horizontal direction and y dots in the vertical direction.

$k = (x_L + x_H \cdot 256) \cdot (y_L + y_H \cdot 256)$

m Mode Scaling for horizontal Scaling for vertical

0, 48 Normal · 1 · 1
1, 49 Double-width · 2 · 1
2, 50 Double-height · 1 · 2
3, 51 Quadruple · 2 · 2

- Horizontal direction dots = (x * 8)dots
- Vertical direction dots = (y)dots

ESC f

[Name] Print downloaded bit-image.

[Format]

ASCII	ESC	f	n
HEX	1B	66	n
Decimal	27	102	n

[Range] n = 0, 1, 48, 49

[Description] ESC f prints a downloaded bit image specified by *n* as follows:
Print a downloaded bit image1 when n = 0 or n = 48,
Print a downloaded bit image2 when n = 1 or n = 49.

[Reference] ESC L, ESC FF

2.7 Miscellaneous Function Commands

The P25 printer supports the following miscellaneous function commands:

Command

GS P Set horizontal and vertical motion units

GS P x y

[Name] Set horizontal and vertical motion units.

[Format]

ASCII	GS	P	x	y
HEX	1D	50	x	y
Decimal	29	80	x	y

[Range] 0 ≤ x ≤ 255, 0 ≤ y ≤ 255

[Description] Sets the horizontal and vertical motion units to ~25.4/x mm(1/x inch) and ~25.4/y mm(1/y inch), respectively. When x and y are set to 0, the default setting of each value is used.

Note

The horizontal direction is perpendicular to the paper feed direction and the vertical direction is the paper feed direction.

The following commands use x or y, regardless of character rotation (upside-down).
Command using x : ESC $, GS L
Command using y : ESC 3, ESC J

The command does not affect the previously specified values.

The calculated result from combining this command with others is truncated to the minimum value of the mechanical pitch.

In this printer, the minimum value of the mechanical pitch is 0.125mm.

[Default] x = 203, y = 203

[Reference] **ESC $, ESC 3, ESC J, GS L**

2.8 Control Device Commands

ESC =

[Name] Select peripheral

[Format]

ASCII	ESC	=	n
HEX	1B	3D	n
Decimal	27	61	n

[Range] $1 \le n \le 3$

[Description] Set *n* to choose the receiving device to which the host will send data.

[Note]

[Default] n=1

ESC |

[Name] Pause *n* seconds

[Format]

ASCII	ESC	|	n
HEX	1B	7C	n
Decimal	27	124	n

[Range] $0 < n < \tilde{5}$

[Description] Set *n* to let the printing process suspend.

GS | 0

[Name] Set the time to sleep

[Format]

ASCII	GS	|	0	n
HEX	1D	7C	00	n
Decimal	29	124	0	n

[Range] 2<=n<=59, 0xFF

[Description] Set enter sleep mode time, unit is minute, if no operation occurs, P25 will enter sleep mode after this time, set to 0xFF means disable this feature(never enter sleep mode), default setting is 5 minutes.

This command is **Deprecated** since version 1.0.25. P25 will enter and quit Sleep mode periodically in every several milliseconds automatically to save power. The user can not set sleep the time to sleep for the printer.

GS | 1

[Name] Set the time to power off

[Format]

ASCII	GS	|	1	n
HEX	1D	7C	01	n
Decimal	29	124	1	n

[Range] 2<=n<=59(0x3B), 0xFF(Version<S1.0.25)

2<=n<=240(0xF0),0xFF(Version>=S1.0.25)

[Description] Set enter power off mode time, unit is minute, after P25 enter sleep mode(ref above cmd), if no operation occurs, P25 will auto power off after this time, set to 0xFF means disable this feature(never auto power off), default setting is 15 minutes.

GS H

[Name] Set the time to immediately power off

[Format] ASCII GS H n
HEX 1D 48 n
Decimal 29 72 n

[Range] 0<=n<=59

[Description] Let the P25 immediately power off after this time, unit is second, P25 will immediately power off after this time.

GS t

[Name] Control BlueTooth discovery mode

[Format] HEX 1D 74 n

[Range] 0<=n<=2

[Description] 1b 74 00 Forever discover
1b 74 01 Mode key (BT discover would enable when mode key is pressed or in 90 seconds since mode key just released)
1b 74 02 One Connect (BT discovery enable when P25 first power on, and disabled after first BlueTooth connect)

This command only support those P25 equipt with new version BlueTooth module, set result save to flash, customer could display it by self test printing. If P25 equipt with old BT module, self print would display “Inctrl BT”, its act like “One Connect” mode of new BT module.

In “mode key” mode, if P25 in indiscoverable state, press mode key would cause blue LED flash, if P25 already in discoverable state, press mode key would not cause blue LED flash. If P25 entered sleep mode, press mode key would not awake P25, and cannot change P25 to discoverable state.

GS {

[Name] Set default font (only set to flash, font would not active immediately)

[Format] HEX 1D 7B n

[Range] 0x00 <= n <= 0x01

[Description] 1d 7B 00 Set default font to 32 dot font (this is factory setting)
1d 7B 01 Set default font to 24 dot font

GS (E

[Name] Set UART 1(serial port or USB port) Baud Rate

[Format] ASCII GS (E pL pH fn a d1…dk
HEX 1D 28 45 pL pH 0B 01 d1…dk

[Range] 3<=(pL+pH*256)<=8
fn=0x0B
a=0x01
0x30<=d<=0x39
1<=k<=6

[Description] Set the UART1 Baud Rate.
UART1(Serial port or USB port) default Baud Rate is 115200

UART0 (BT port) default Baud Rate is 38400, and it cannot be changed.

If set baudrate to 0, then back to default baudrate. (Since S1.0.42)

The default Baud rate in the bootloader firmware is 115200, and it can not be changed. These commands are only used to adjust the Baud rate in the application of the P25. Furthermore, the change will be affect only after you restart the printer.

Valid Bade Rate: 9600, 19200, 38400, 57600, 115200

Eg:

```
1d 28 45 03 00 0b 01 30                   //set to default Baud rate
1d 28 45 06 00 0b 01 39 36 30 30          //set to 9600 Baud rate
1d 28 45 07 00 0b 01 31 39 32 30 30       //set to 19200 Baud rate
1d 28 45 07 00 0b 01 33 38 34 30 30       //set to 38400 Baud rate
1d 28 45 07 00 0b 01 35 37 36 30 30       //set to 57600 Baud rate
1d 28 45 08 00 0b 01 31 31 35 32 30 30 //set to 115200 Baud rate
```

2.9 ESC/GS Commands with Respond Info

GS I

[Name] Transmit printer ID

[Format]

ASCII	GS	I	n
HEX	1D	49	n
Decimal	29	73	n

[Range] Ref following Description field

[Description]

Hex Decimal n Printer ID Specification

41 65 Firmware version
42 66 Manufacturer name “BlueBamboo”
43 67 Printer model
44 68 Serial No of the printer
50 80 Hardware version of the printer

Each printer’s information is composed of [header to NUL].

Send data	**Hex**	**Decimal**	**Data**
Header	5FH	95	1 byte
Printer information	…	…	0 to 80 bytes
NUL	00H	0	1 byte

[Note]

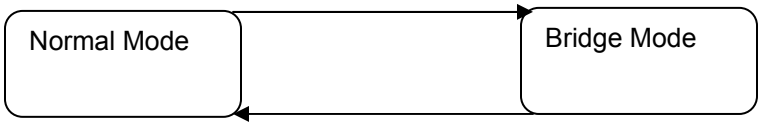
[Default]

2.10 Enter/Quit Bridge Mode Commands

These commands are used to switch the P25 printer between normal mode and bridge mode.

P25 printer has two working modes, one is normal mode, and another is bridge mode. In the normal mode, P25 can accept and print data both in Blue Bamboo frame protocol and plain text protocol. When the printer is in the bridge mode, P25 can receive and print data in Blue Bamboo frame protocol as same as in the normal mode. However, if the P25 receive the data in other format, instead of printing it out, it will act as a bridge, to exchange these data between USB port (serial port) and Bluetooth port which connects with other device.
Power on the P25 printer, it is in the normal mode by default. You can send the printer the commands of ENTER_BRIDGE_MODE to enter bridge mode, and send the commands of QUIT_BRIDGE_MODE to quit bridge mode.

The commands of ENTER_BRIDGE_MODE and QUIT_BRIDGE_MODE should be in Blue Bamboo frame protocol.



The commands of ENTER_BRIDGE_MODE or QUIT BRDGE MODE should be wrapped in C0...C1 frame, and must be sent to the printer together because the P25 need to differentiate these commands from other data that need to be exchanged between two ports. When the P25 is in bridge mode, after receiving some data from one port, P25 will judge whether it is in Blue Bamboo frame protocol or not, if yes, P25 would act. Otherwise, it will send all these data to another port.

ESC w

[Name]	Enter/Quit Bridge mode Commands			
[Format]	ASCII	ESC	w	n
	HEX	1B	77	n
	Decimal	27	119	n
[Range]				
[Description]	1B 77 30 QUIT_BRIDGE_MODE 1B 77 31 ENTER_BRIDGE_MODE			
[Note]	This command should be only in Blue Bamboo frame protocol, not in plain text protocol.			
[Default]				

2.11 Realtime Commands

OVERVIEW

Character Data and Normal Commands
The printer stores data sent from the host computer in the receive buffer temporarily, and then the printer interprets the data and classifies them into commands or character data sequentially. If the data from the receive buffer is a normal command, the printer processes the command corresponding to its function; for example, if the data interpreted is ESC 3, the printer changes a setting value for the line spacing, and if it is LF, the printer prints the data in the print buffer and feeds the paper one line.

If the data from the receive buffer is character data, the printer reads the appropriate font data from the resident character generator and print it.

Real-time Commands
The printer stores data sent from the host computer in the receive buffer, interprets the data, and processes the commands corresponding to their function one line at a time (plain text mode). The real-time commands are the commands that consist of a DLE extension, such as DLE EOT or DLE ENQ. They are processed immediately.

DLE DC4 (*fn* = 7)

[Name] Transmit specified status in real time
ASCIIDLE DC4 fn m
Hex 10 14 fn m
Decimal 16 20 fn m

[Range] fn = 7
m = 5

[Description] Transmits specified status in real-time as follows.

[Notes]
- ■ This is a real-time command
- ■ The construction of battery status depends on printer model.
- ■ When you use this command, obey the following rules.
- • After the host PC transmits the function data, the printer will send response data or status data back to the PC. Do not transmit more data from the PC until the response data or status data are received from the printer.
- • When operating with a serial interface, be sure to configure operation so that the host computer uses the printer only when it is READY.

The transmitted battery status from this printer is constructed by [Header ~ NUL] as shown in the following table.

Transmitted data	Hex	Decimal	Amount of data
Header	37H	55	1 byte
Identifier	45H	69	1 byte
Power source (*1)	31H	49	1 byte
Battery remaining amount (*2)	30H-33H	48-51	1 byte
NUL	00H	0	1 byte

(*1) “Power source” must be 31H, indicates is power by battery.
(*2) “Battery remaining amount” is as indicated in the following table.

Battery remaining amount		Information
Hex	Decimal	
30H	48	Battery remaining amount: H level
31H	49	Battery remaining amount: M level
32H	50	Battery remaining amount: L level
33H	51	Battery remaining amount: S level

• When the battery remaining amount is “L level,“ we recommend replacing or charging the battery.
• When the battery remaining amount is “S level,“ the printer terminates printing.

3. Introduction of Communication Protocol

The P25 printer supports two types of communication, plain text protocol and framing protocol, Plain text protocol is much simpler however with limited features. The framing protocol supports more features and is recommended.

Those two protocols can be automatically interpreted on the fly by the P25.

3.1 Plain Text Protocol

Plain text protocol is very simple, for example, if you want to print the string “abc”, you can just send the string “abc” to the serial port, and then attach a CR. The P25 will store every character received from the buffer, and after meeting a ‘CR’, it will print all the characters in buffer. The user can also send ESC commands by using the plain text protocol. In fact, the user can connect a terminal like “Hyper terminal” in Windows to the P25, and type something to make the printer print.

If using the plain text protocol to print text, no data will be sent from the P25, so the user will have to wait until the P25 finishes printing, and the waiting time depends on the contents that are printed. If the user wants to get an active response and have more time to send additional data, it is better to use the framing protocol than use plain text protocol.

If P25 receive data in Plain Text Protocol, P25 would not respond any result except it receives the ESC/GS Commands with respond info(GS I n command),

3.2 Framing Protocol

Communication Frame Structure

SOF (Start Of Frame)	**TOF** (Type Of Frame)	**DATA**	**EOF** (End Of Frame)
1 Byte	1 Byte	Variable	1Byte

Type of Frame	**Value**	**Field**
ACK	0x06	X
NACK	0x15	X
ENQ	0x05	X
DATA	‘D’ (0x44)	O
FLASH Erase	‘E’ (0x45)	O
Download Mode	‘F’ (0x46)	O
Printer Status Response	‘S’ (0x53)	O
Printer Status Inquiry	‘Q’ (0x51)	X
EOT	0x04	X
ETX	0x03	X

- ✧ The items marked ‘X’ must be present during the entire communication process.
  The items marked ‘O’ must be present during certain parts of the process.
- ✧ During transmission, if C0H, C1H, and/or 7DH are shown in the DATA field, 7DH should be inserted before the data and the data should be XORed with 20H and sent.
- ✧ During reception, if 7DH is encountered, 7DH should be ignored and the next byte should be XORed with 20H and stored.

3.3 Process of Getting the Printer Status

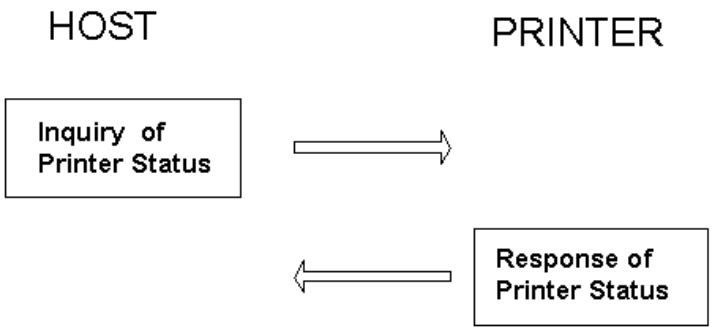


Figure 1: Process of getting the printer status

It is recommended that the host send the same inquiry up to 5 times with 400ms time interval in case of no response from the printer.

3.4 Printing or Downloading Data

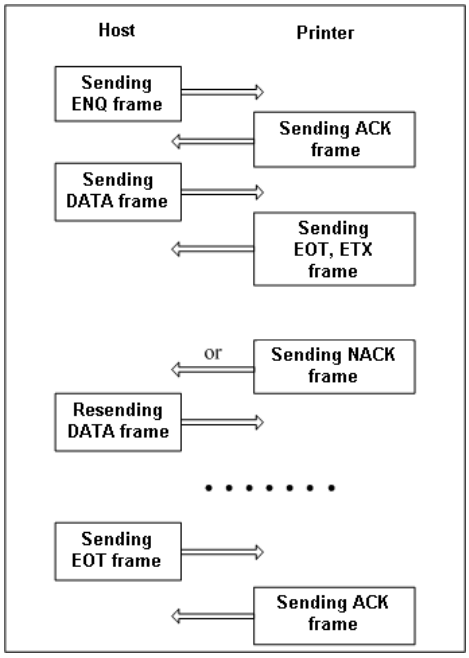


Figure 2: Process of printing data

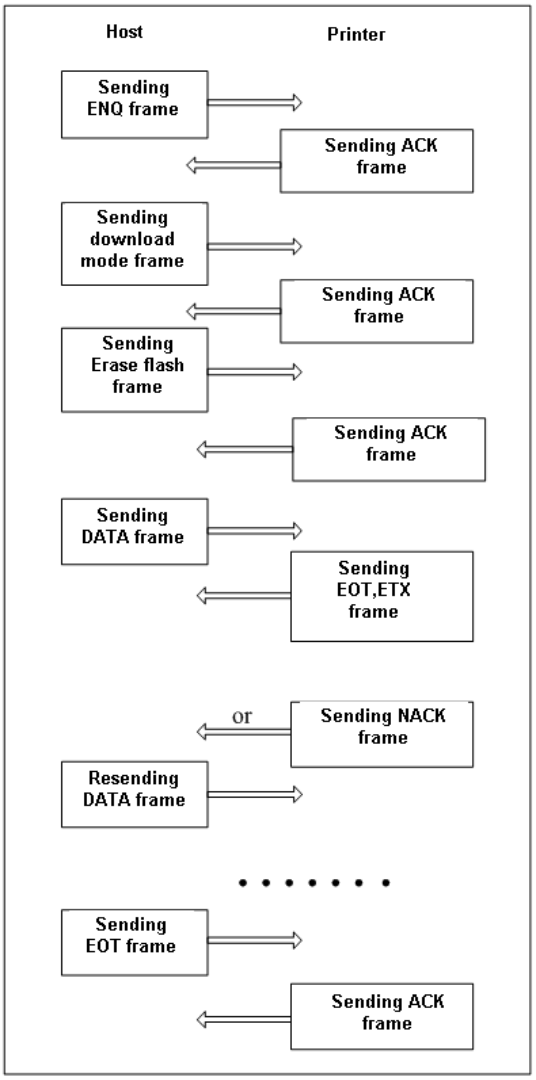


Figure 3: Process of software update

Note: The “Sending ENQ frame” is optional, it just let app know whether print is working normally, customer could print data without sending this frame (that is only sending data frame).

In normal operation, after receiving the print data frame, the printer sends EOT, prints out the data, and sends ETX; then it will wait for the next frame.

✧ Conditions of NACK frame issued

1) Different checksum value
2) No EOF (End of frame) received in maximum frame length time
3) No predefined number in Data Length field

When using those ESC/GS Commands with respond info (GS I n command), respond info package would insert between EOT package and ACK package.

3.4.1 Format of Printing or Downloading Data Frame

C0H (SOF)	'D' (0x44) (TOF)	DATA ID Number	DATA Length	DATA	CHECK SUM	C1H (EOF)
		1 Byte	4 Bytes	Data	2 Bytes	

- ✧ When printing data or downloading data to the printer, the communication protocol should comply with the upper communication structure.
- ✧ Data ID number: '0'~'9' (0x30~0x39). Every time the host sends a new print data frame, it increases this number.
- ✧ Data Length: "0001" ~ "3000". Each number must be an ASCII code, the highest byte of Data Length should be transmitted first, P25 does not support printing more than 3000 characters per frame.
- ✧ Checksum: 2 bytes. The first byte is the result of XORing the even parts of data in the Print Data field and the second byte is from doing the same to the odd numbers.

[Example] If "SAMPLE TEST" is in Print Data field, the data length will be "0011 (0x30 0x30 0x31 0x31)" and the first byte of checksum will be the result of XOR of S, M, L, space, E, and T and the second byte that of A, P, E, T, and S.

- ✧ It is recommended that the host goes back to the initial stage in case it receives neither ETO nor NACK from the printer one second after it has sent the print data frame.
- ✧ Before downloading, enter the **DOWNLOAD** mode first by pressing **FEED**, and then turn on the printer, which is the only way to enter the DOWNLOAD mode.
- ✧ However, in order to download anything to FLASH memory, you should comply with the FLASH space distribution and storage format; for details please see the Flash Memory Distribution Map.

3.4.2 Format of Flash Erase Frame

C0H (SOF)	'E'(0x45) (TOF)	DATA ID Number	DATA Length	DATA	CHECK SUM	C1H (EOF)

- ✧ DATA Length is equal to 8 in this command, because there are just 8 bytes in the DATA field for this command.
- ✧ The DATA field contains two parts, the "Start address" and "File size", both of which occupy 4 bytes.

  "Start address" is used to define the start address for erasure of FLASH memory, and the command will erase information from this address to the end of the whole chip. The "File size" section defines the byte number you want to erase from FLASH memory from the start address.

Please see Figure 4.

Start address				File size			
Byte3	Byte2	Byte1	Byte0	Byte3	Byte2	Byte1	Byte0

Figure 4: DATA Field Description

Each byte in the DATA field should be in HEX code; the highest byte should be transmitted first in the same way DATA Length is transmitted.

3.4.3 Format of Download Frame

C0H (SOF)	'F'(0x46) (TOF)	C1H (EOF)

✧ This frame just indicates that the following frame is an erasing or a writing flash frame.

3.4.4 Format of ENQ Frame

C0H(SOF)	0x05(TOF)	C1H(EOF)

✧ It is recommended that the host send the same ENQ frame up to 10 times with 400ms time interval pauses in case there is no response from the printer.

3.4.5 Format of Frame of request to MSR

C0H (SOF)	'H'(0x48) (TOF)	DATA ID Number	DATA Length	DATA	CHECK SUM	C1H (EOF)

✧ The DATA Length is fixed to 2

✧ The DATA field just is the time to wait to pull the MSR. The first byte is the MSB.

- the time valid value: 0 (DATA: "00")~ 30(DATA: "30") seconds
- other time value: pull the MSR forever

✧ The other fields is same as section 3.2.1

✧ eg.

Index	0	1	2	3	4	5	6	7	8	9	10	11
data	C0	48	30	30	30	30	32	32	30	32	30	C1

0. Frame head
1. TOF
2. DATA ID
3-6. DATA Length(its value fix to 2)
7,8. Wait time (this example set to 20 seconds)
9,10. CRC
11. Frame tail

3.4.6 MSR Response Frame Format

DATA in clear text

C0H (SOF)	'H'(0x48) (TOF)	DATA ID Number	DATA Length	DATA	CHECK SUM	C1H (EOF)

The 'DATA' field contains the three tracks of MSR data, the format of each track is:

Track No (1 bytes)	Track Data Length (4 bytes)	Track Data

1. The Track No: 0x31 for track 1, 0x32 for track 2 and 0x33 for track 3;
2. The Track Data Length's format is same as the DATA Length field;
3. If there are more than one track, track data should be filled in sequence.

✧ The format of other fields is same as the section 3.2.1

✧ Eg. C0 48 30 30 30 34 34 31 30 30 30 39 25 31 48 49 4A 4B 4C 4D 3F 32 30 30 30 39 3B 32 35 36 37 38 39 30 3F 33 30 30 31 31 3B 33 34 34 35 35 36 36 37 38 3F 7F 6E C1, This frame has 3 tracks, Track1: <%1HIJKLM?>, Track2: <;2567890?>, Track3: <;344556678?>

DATA encrypted

C0H (SOF)	(0x90) (TOF)	DATA ID Number	DATA Length	DATA	CHECK SUM	C1H (EOF)

Note: 0x90 means CT_ENCRYPT_DUKPT, which means DATA is encrypted by DUKPT. The Field DATA Length is length of next field data. The length is the encrypted data's length, not plain text's length.

The format of the “DATA” is:

FIELD		LEN Byte	DESCRIPTION
Version		1	Specify the data encryption protocol version. It should be 0x02
Key ID (RFU)		1	Specify which key is used to encrypt data. P25 supports only one key right now. Its value should be 0x00.
Alg ID		1	Specify the type of data encryption and key management method 0x01 : Fixed DES 0x02 : Fixed 3DES 0x03 : Single DES DUKPT 0x04 : 3DES DUKPT 0x05-0xFF: RFU (Reserved for Future Use)
KSN (optional)		10	Key Serial Number, if DUKPT is applied
Encrypted Track Data	Encrypted Track1 Data Length	1	The length should be 0 if there is no data in this track. It is should be in multiple of cipher block size (8 bytes in this case). 1. The data to be encrypted is the track data(s) in clear text. 2. The track data(s) should be padded to multiple of cipher block size (8 bytes in this case). 3. The padding string is comprised of bytes equal to the padding string length (if data is already multiple of block size, a new block with all bytes equal to block size needs to be padded) Eg. The data to be encrypted in the example of above section is 45 bytes long (1 cmd+44 track data) 31 30 30 30 39 25 31 48 49 4A 4B 4C 4D 3F 32 30 30 30 39 3B 32 35 36 37 38 39 30 3F 33 30 30 31 31 3B 33 34 34 35 35 36 36 37 38 3F; 3 bytes need to be padded to make it 48 bytes long, the padding bytes are all ‘03’: 31 30 30 30 39 25 31 48 49 4A 4B 4C 4D 3F 32 30 30 30 39 3B 32 35 36 37 38 39 30 3F 33 30 30 31 31 3B 33 34 34 35 35 36 36 37 38 3F 03 03 03. If the MSR data is 48 bytes long which is multiple of 8 already, a new block with 8 ‘08’ should be padded: 31 30 30 30 39 25 31 48 49 4A 4B 4C 4D 3F 32 30 30 30 39 3B 32 35 36 37 38 39 30 3F 33 30 30 31 33 3B 33 34 34 35 35 36 36 37 38 39 39 39 39 3F 08 08 08 08 08 08 08 08
	Encrypted Track1 Data		
	Encrypted Track2 Data Length	1	
	Encrypted Track2 Data		
	Encrypted Track3 Data Length	1	
	Encrypted Track3 Data		
Card’s Last 4 Digits (Optional)		4	
Cardholder’s First and Last name (Optional)		26	
Card Expiration Date (Optional)		4	

3.4.7 ACK Frame Format

C0H(SOF)	0x06(TOF)	C1H(EOF)

3.4.8 NACK Frame Format

C0H(SOF)	0x15(TOF)	C1H(EOF)

3.4.9 ETX Frame Format

C0H(SOF)	0x03(TOF)	Data ID No.	C1H(EOF)

✧ Printer will send this frame after it finishes all of the requested printing.

3.4.10 Format of EOT Frame

C0H(SOF)	0x04(TOF)	C1H(EOF)

✧ Printer will send this frame after it receives the print data frame successfully.

- **IMPORTANT!!**

In every frame coming from the printer, 1 byte of null is preceded by SOF, EOF is followed by CR and LF.
For example, the actual data of ETX frame is 0x00, 0xC0, 0x03, 0xC1, 0x0D, and 0x0A. The host can ignore these prefixes and suffixes.

00	C0H	'F' (0x46)	C1H	0D	0A

3.4.11 Format of Set DUKPT KSN and initial key (Request)

If customer need encrypt MSR data with DUKPT algorism, they need first set DUKPT KSN and initial key to P25.

C0H (SOF)	(0x9B) (TOF)	DATA ID Number	DATA Length	Version	Algor	Reserved	KSN	IKEY	CRC	C1H (EOF)

Eg:

C0 9B 30 30 30 32 39 01 04 00 FF FF 31 32 33 34 35 A0 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 58 CC C1

Explain to last Eg:

Eg Data Content	Length	Explain
C0	1	frame begin
9B	1	Set initial key request
30	1	Frame ID
30 30 32 39	4	Len
01	1	Version
04	1	Algor
00	1	Reserved
FF FF 31 32 33 34 35 A0 00 00	10	KSN
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	16	IKEY
58 CC	2	CRC
C1	1	frame end

3.4.12 Format of Set DUKPT KSN and initial key (Response)

This Data is respond from P25 to program like Device Manager.

C0H (SOF)	(0x9C)	DATA ID Number	DATA Length	Version	Algor	Reserved	Result	CRC	C1H (EOF)

Eg:

C0 9C 36 30 30 30 34 01 04 00 00 01 04 C1

Explain to last Eg:

Eg Data Content	Length	Explain
C0	1	frame begin
9C	1	Set initial key response
36	1	Frame ID
30 30 30 34	4	Len
01	1	Version
04	1	Algor
00	1	Reserved
00	1	Result(00_OK, else fail)
01 04	2	CRC
C1	1	frame end

3.4.13 P25 MK (Master key)/SK (Session Key) Encrypt

The purpose to set MK and SK is to ensure the printing commands in P25 safer than before. The user can set MK/SK according to their request.

1. Encrypt Algorism

MK/SK only use 3Des algorism, Make message padding to multiple of 8 bytes first then using pkcs#5, for eg:
data： FF FF FF FF FF FF FF FF
after padding： FF FF FF FF FF FF FF FF 07 07 07 07 07 07 07 07
data： FF FF FF FF FF FF FF FF
after padding： FF FF FF FF FF FF FF FF 08 08 08 08 08 08 08 08

If the key length is 48(master key), first encrypt with the beginning 24 bytes key then encrypt with the rest 24 bytes key.

2. Reference application example:

We provide P25EncryptApp as the example which can be compiled and run under cygwin environment.

Compile:

1. Install cygwin develop enviroment (www.cygwin.com)
2. make

Before running p25EncryptApp, the user can edit P25EncryptApp.ini as you needed.

2. Download master key

Master key download use clear text and length should be 48 bytes. It will be saved in P25 printer permanently once it downloaded successfully. Each P25 printer can load 16 master keys, which has their own key ID separately.

oper:
Use command "./P25EncryptApp set_master_key_down"
---

6 [00000004] IRP_MJ_WRITE Length: 0061
Data: C0 8B 36 30 30 35 31 01 01 01 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 78 90 12 34 56 08 E1 C1

7 [00000033] IRP_MJ_READ Length: 0014
Data: C0 8C 36 30 30 30 34 01 01 01 00 00 01 C1

REQUEST:

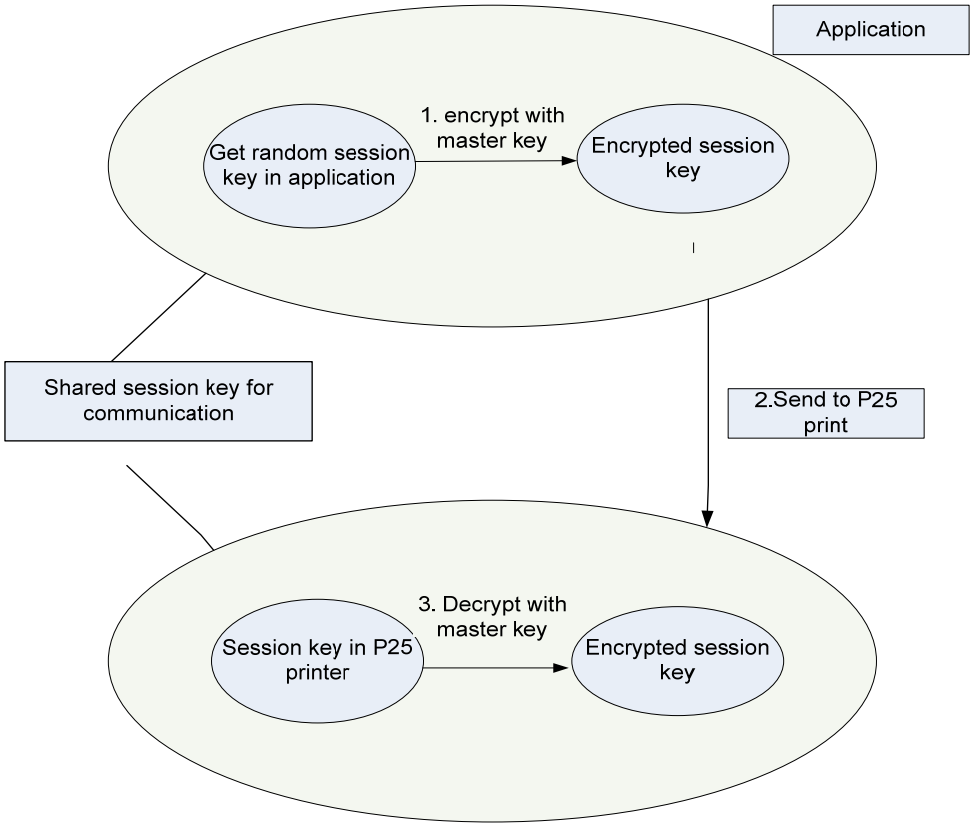
Eg Data Content	Length	Explain
C0	1	start of frame
8B	1	CMD_SET_MASTER_KEY_DOWN_REQ
36	1	data id
30 30 35 31	4	data len
01	1	Version
01	1	Algor, ENCRYPT_ALG_3DES
01	1	key id
12 34....34 56	48	master key
08 E1	2	CRC
C1	1	frame end

RESPONSE:

Eg Data Content	Length	Explain
C0	1	start of frame
8C	1	CMD_SET_MASTER_KEY_DOWN_RESPONSE
36	1	data id
30 30 30 34	4	data len
01	1	Version
01	1	Algor, ENCRYPT_ALG_3DES
01	1	key id
00	1	result
08 E1	2	CRC
C1	1	frame end

3. Sync Session key

Main procedures of syncing session key is shown as the below diagram.



The customer can encrypt the card information by setting session key which length is 24 bytes in P25 printer. Session key only in ram, if customer begin new session, better renew it, Sync session key:

Use command "./P25EncryptApp sync_session_key_down"

---

6 [00000004] IRP_MJ_WRITE Length: 0045

Data: C0 80 36 30 30 33 35 01 01 01 20 87 8F 8F D7 59 3D FA A8 89 1E 6D 41 2F 36 4F
4C A8 48 87 3A FF 63 36 CB 70 B8 04 1B B8 B6 2A 2F 06 C1

7 [00000014] IRP_MJ_READ Length: 0014

Data: C0 86 36 30 30 30 34 01 01 00 00 01 01 C1

REQUEST:

<table>
<tr><th>Eg Data Content</th><th>Length</th><th>Explain</th></tr>
<tr><td>C0</td><td>1</td><td>start of frame</td></tr>
<tr><td>80</td><td>1</td><td>CT_ENCRYPT_WITH_MASTER</td></tr>
<tr><td>36</td><td>1</td><td>data id</td></tr>
<tr><td>30 30 33 35</td><td>4</td><td>data len</td></tr>
<tr><td>01</td><td>1</td><td>Version</td></tr>
<tr><td>01</td><td>1</td><td>key id</td></tr>
<tr><td>01</td><td>1</td><td>Algor, ENCRYPT_ALG_3DES</td></tr>
<tr><td>20 87...B6 2A</td><td>32</td><td>ENCRYPTED_WITH_MASTER_KEY(CMD_SET_SESSION_KEY,SESSION_KEY,CRC,PADDING)
after decrypt, it would be:
<table>
<tr><th>Eg data content</th><th>Length</th><th>Explain</th></tr>
<tr><td>85</td><td>1</td><td>CMD_SYNC_SESSION_KEY_DOWN_REQ</td></tr>
<tr><td>??...??</td><td>24</td><td>24 bytes session_key</td></tr>
</table></td></tr>
<tr><td>2F 06</td><td>2</td><td>CRC</td></tr>
<tr><td>C1</td><td>1</td><td>end of frame</td></tr>
</table>

RESPONSE:

Eg Data Content	Length	Explain
C0	1	start of frame
8C	1	CMD_SET_MASTER_KEY_DOWN_RESPONSE
36	1	data id
30 30 30 34	4	data len
01	1	Version
01	1	Algor, ENCRYPT_ALG_3DES
01	1	key id
00	1	result
08 E1	2	CRC
C1	1	frame end

4. Print encrypted information by session key

The following operation is a combined operation. Sync session key for the encrypted information first then print it out through P25 printer. This paragraph only explains the format of printing encrypted information.

Sync session key and print string "abc"
use command "./P25EncryptApp sync_session_prn_abc"

---

6 [00000004] IRP_MJ_WRITE Length: 0046
Data: C0 80 36 30 30 33 35 01 01 01 75 6C 2B BF E1 D7 3F 07 7D E1 94 B2 88 14 CA 19 27 4C A8 48 87 3A FF 63 36 CB 70 B8 04 1B B8 B6 2A F2 7C C1

7 [00000013] IRP_MJ_READ Length: 0014
Data: C0 86 36 30 30 30 34 01 01 00 00 01 01 C1
Ref sync session key

8 [00000014] IRP_MJ_WRITE Length: 0021
Data: C0 81 37 30 30 31 31 01 01 01 D3 36 57 EA 37 94 B5 EA A2 07 C1

Request to print "abc" in encrypt format

<table>
<tr><th>Eg Data Content</th><th>Length</th><th colspan="3">Explain</th></tr>
<tr><td>C0</td><td>1</td><td colspan="3">start of frame</td></tr>
<tr><td>81</td><td>1</td><td colspan="3">CT_ENCRYPT_WITH_SESSION</td></tr>
<tr><td>37</td><td>1</td><td colspan="3">data id</td></tr>
<tr><td>30 30 31 31</td><td>4</td><td colspan="3">data len</td></tr>
<tr><td>01</td><td>1</td><td colspan="3">Version</td></tr>
<tr><td>01</td><td>1</td><td colspan="3">key id</td></tr>
<tr><td>01</td><td>1</td><td colspan="3">Algor, ENCRYPT_ALG_3DES</td></tr>
<tr><td>D3 36...B5 EA</td><td>Multiple of 8</td><td colspan="3">ENCRYPTED_WITH_SESSION_KEY(CMD_PRN_REQ_WILL_ENCRYPT,PRINT_CONTENT,CRC,PADDING)
after decrypt, it would be:
<table>
<tr><th>Eg Data Content</th><th>Length</th><th>Explain</th></tr>
<tr><td>87</td><td>1</td><td>CMD_PRN_REQ_WILL_ENCRYPT</td></tr>
<tr><td>??...??</td><td>Variable</td><td>print content</td></tr>
</table></td></tr>
<tr><td>A2 07</td><td>2</td><td colspan="3">CRC</td></tr>
<tr><td>C1</td><td>1</td><td colspan="3">end of frame</td></tr>
</table>

9 [00000047] IRP_MJ_READ Length: 0021
Data: C0 81 37 30 30 31 31 01 01 01 08 36 14 A0 E9 B9 52 25 0A A6 C1

<table>
<tr><th>Eg Data Content</th><th>Length</th><th>Explain</th></tr>
<tr><td>C0</td><td>1</td><td>start of frame</td></tr>
<tr><td>81</td><td>1</td><td>CT_ENCRYPT_WITH_SESSION</td></tr>
<tr><td>37</td><td>1</td><td>data id</td></tr>
<tr><td>30 30 31 31</td><td>4</td><td>data len</td></tr>
<tr><td>01</td><td>1</td><td>Version</td></tr>
<tr><td>01</td><td>1</td><td>key id</td></tr>
<tr><td>01</td><td>1</td><td>Algor, ENCRYPT_ALG_3DES</td></tr>
<tr><td>08 36...52 25</td><td>Multiple of 8</td><td>ENCRYPTED_WITH_SESSION_KEY(CMD_PRINT_RESPONSE, PRN_FRAME_ACK,CRC,PADDING)
after decrypt, it would be:
other content, means error occure.
<table>
<tr><th>Eg Data Content</th><th>Length</th><th>Explain</th></tr>
<tr><td>88</td><td>1</td><td>CMD_PRINT_RESPONSE</td></tr>
<tr><td>06</td><td>1</td><td>PRN_FRAME_ACK</td></tr>
</table></td></tr>
<tr><td>0A A6</td><td>2</td><td>CRC</td></tr>
<tr><td>C1</td><td>1</td><td>end of frame</td></tr>
</table>

5. Get encrypted MSR information by session key

The following operation is a combined operation, Firstly,sync session key then read MSR information. This paragraph only explains the format of reading MSR informaiton.

Sync session key and read msr info
Use command "./P25EncryptApp sync_session_msr"
---
6 [00000004] IRP_MJ_WRITE Length: 0046,
Data: C0 80 36 30 30 33 35 01 01 01 6E 52 7D 5D 18 75 CC BE 3D AA F9 C7 C2 9D 0C 26 73
4C A8 48 87 3A FF 63 36 CB 70 B8 04 1B B8 B6 2A FF 8C C1
7 [00000014] IRP_MJ_READ Length: 0014
Data: C0 86 36 30 30 30 34 01 01 00 00 01 01 C1
Ref sync session key

8 [00000014] IRP_MJ_WRITE Length: 0003, Data: C0 05 C1

9 [00000023] IRP_MJ_READ Length: 0006, Data: 00 C0 06 C1 0D 0A

Just confirm P25 works OK, or following send msr command may timeout for long time, it would confuse user, but this command is option

10 [00000024] IRP_MJ_WRITE Length: 0021

Data: C0 81 37 30 30 31 31 01 01 01 11 9D F1 BA 87 39 15 CA D4 73 C1

Request to read MSR data

<table>
<tr><td>Eg Data Content</td><td>Length</td><td>Explain</td></tr>
<tr><td>C0</td><td>1</td><td>start of frame</td></tr>
<tr><td>81</td><td>1</td><td>CT_ENCRYPT_WITH_SESSION</td></tr>
<tr><td>37</td><td>1</td><td>data id</td></tr>
<tr><td>30 30 31 31</td><td>4</td><td>data len</td></tr>
<tr><td>01</td><td>1</td><td>Version</td></tr>
<tr><td>01</td><td>1</td><td>key id</td></tr>
<tr><td>01</td><td>1</td><td>Algor, ENCRYPT_ALG_3DES</td></tr>
<tr><td>11 9D...15 CA</td><td>Multiple of 8</td><td>ENCRYPTED_WITH_MASTER_KEY(CMD_MSR_REQ,timout,CRC,PADDING)after decrypt, it would be:
<table>
<tr><td>Eg Data Content</td><td>Length</td><td>Explain</td></tr>
<tr><td>89</td><td>1</td><td>CMD_MSR_REQ</td></tr>
<tr><td>??...??</td><td>2</td><td>timeout, eg "32 30" means msr timeout would be 20 second</td></tr>
</table></td></tr>
<tr><td>D4 73</td><td>2</td><td>CRC</td></tr>
<tr><td>C1</td><td>1</td><td>end of frame</td></tr>
</table>

11 [00000435] IRP_MJ_READ Length: 0136

Data: C0 81 37 30 31 32 33 01 01 01 FD 78 C8 99 42 BA 7F 2B E8 5D E5 89 EB 4A 77 11 27 4A E7 61 8D 7D E0 B0 56 4C 9E 97 D6 A3 8A 8D 6B 5C C4 A9 C2 A5 A3 95 89 B2 A2 C7 05 14 93 9E 7B B0 A0 43 F6 DD 80 68 05 74 A2 50 F7 D3 A0 57 2D 32 69 43 7D E0 B7 36 C3 82 05 01 3A 70 86 50 A8 47 A3 BF C6 B7 6B 73 A1 84 0C 77 93 27 93 69 5F E0 D1 7D E0 04 45 23 9A 14 26 54 9B B4 77 65 1C 84 D3 49 E9 78 2F 4A 94 E7 60 F6 DB C1

<table>
<tr><td>Eg Data Content</td><td>Length</td><td colspan="3">Explain</td></tr>
<tr><td>C0</td><td>1</td><td colspan="3">start of frame</td></tr>
<tr><td>81</td><td>1</td><td colspan="3">CT_ENCRYPT_WITH_SESSION</td></tr>
<tr><td>37</td><td>1</td><td colspan="3">data id</td></tr>
<tr><td>30 31 32 33</td><td>4</td><td colspan="3">data len</td></tr>
<tr><td>01</td><td>1</td><td colspan="3">Version</td></tr>
<tr><td>01</td><td>1</td><td colspan="3">key id</td></tr>
<tr><td>01</td><td>1</td><td colspan="3">Algor, ENCRYPT_ALG_3DES</td></tr>
<tr><td>FD 78...E7 60</td><td>Multiple of 8</td><td colspan="3">ENCRYPTED_WITH_MASTER_KEY(CMD_MSR_RESPONSE,msr_info,CRC,PADDING)after decrypt, it would be:
other content, means error occure.
<table>
<tr><td>Eg Data Content</td><td>Length</td><td>Explain</td></tr>
<tr><td>8a</td><td>1</td><td>CMD_MSR_RESPONSE</td></tr>
<tr><td>??..??</td><td>Variable</td><td>msr info, ref plain text msr info.</td></tr>
</table></td></tr>
<tr><td>F6 DB</td><td>2</td><td colspan="3">CRC</td></tr>
<tr><td>C1</td><td>1</td><td colspan="3">end of frame</td></tr>
</table>

6. Attention

When define protocol format, because carelessness, in those MK/SK commands format, those encrypt format and those not encrypt format's ALGOR and keyid order reversed.
In Encrypted Package: Keyid, ALGOR
In Non-Encrypted Package: ALGOR, Keyid (include SET_MASTER_KEY_REQUEST, SET_MASTER_KEY_RESPONSE, SYNC SESSION_KEY_RESPONSE)

3.5 Some Example of Communication Command

These examples are pasted from the free software AccessPort, if you have questions on what serial data is at the serial port, you could use this software to monitor it, you can download it for free from http://www.sudt.com/en/download.htm.

Latter data may have some data not in C0..C1 frame, like ‘00’ ‘0D 0A’, it do not appear in protocol describe, it is no meaning, just ignore it.

3.5.1 Query Status

C0 51 C1 is deprecated, it return fixed result.
C0 53 C1 could be used to query status.
1 [00006329] IRP_MJ_WRITE Length: 0003, Data: C0 53 C1
2 [00006342] IRP_MJ_READ Length: 0008, Data: 00 C0 53 00 C1 C1 0D 0A
3 [00007332] IRP_MJ_WRITE Length: 0003, Data: C0 53 C1
4 [00007346] IRP_MJ_READ Length: 0008, Data: 00 C0 53 01 C1 C1 0D 0A

Note: Step 1. Query P25 Status
Step 2. P25 report no error, The 00 Byte indicates no error
Step 4. P25 report no paper, The 01 Byte indicates no paper or cover open
Error Indicate Byte Meaning:
00 ---------- No error
01 ---------- No paper or cover open
04 ---------- Printer header's temperature too high
08 ---------- Low battery

3.5.2 Print

1 [00001440]IRP_MJ_WRITE Length: 0013, Data: C0 44 30 30 30 30 33 61 62 63 02 62 C1
2 [00001444]IRP_MJ_READ Length: 0006, Data: 00 C0 04 C1 0D 0A
3 [00001459]IRP_MJ_READ Length: 0007, Data: 00 C0 03 30 C1 0D 0A

Note: Step 1. Send Print Command
Step 2. P25 report received command OK
Step 3. P25 report print finished

Print protocol:

Mobile phone		P25M
Write	ENQ(C0 05 C1)	Read
Read	ACK(C0 06 C1)	Write
Write	Print Data(C0 44 30 30 30 31 38 48 65 6C 6C 6F 2C 20 42 6C 75 65 42 61 6D 62 6F 6F 21 0E 73 C1)	Read
Read	EOT(C0 04 C1)	Write
Read	ETX(C0 03 30 C1)	Write
Write	EOT(C0 04 C1)	Read
Read	ACK(C0 C1)	Write

a) print 1D barcode
Barcode content: 6901234567892
Frame content: c0 44 32 30 30 31 37 1d 6b 02 0d 36 39 30 31 32
33 34 35 36 37 38 39 32 23 66 c1
Description:
C0: start tag
44: print command
32: frame sequence
30 30 31 37: 17 bytes data
1d 6b 02 0d: barcode format
36..32: barcode data
23: odd check
66: even check
C1: end tag

b) print 2D barcode
Barcode content: Hello, world! A PDF417 example.
Frame content:
c0 44 33 30 30 34 30 1d 6b 10 00 00 00 00 00 1f
48 65 6c 6c 6f 2c 20 77 6f 72 6c 64 21 20 41 20
50 44 46 34 31 37 20 65 78 61 6d 70 6c 65 2e 00 33 c1
Description:
C0: start tag
44: print command
33: frame sequence
30 30 34 30: 40 bytes data
1d 6b 10 00 00 00 00 00 1f: 2D barcode format
48..2e: barcode data
00: odd check
33: even check
C1: end tag

c) print English text:
print content: Welcome to Bluebamboo

c1) small font size, underline:
Frame Content:
c0 44 33 30 30 33 33 1b 4b 31 1b 21 41 1b 2d 01
1d 42 00 57 65 6c 63 6f 6d 65 20 74 6f 20 62 6c
75 65 62 61 6d 62 6f 6f 53 72 c1
Description:
C0: start tag
44: print command
33: frame sequence
30 30 33 33: 33 bytes data
1b 4b 31: UTF-8
1b 21 41: small font size
1b 2d 01: underline on
1d 42 00: reverse off
57..6f: text data
53: odd check
72: even check
C1: end tag

c2) double small font size:

Frame Content:

c0 44 34 30 30 33 33 1b 4b 31 1b 21 19 1b 2d 00
1d 42 00 57 65 6c 63 6f 6d 65 20 74 6f 20 62 6c
75 65 62 61 6d 62 6f 6f 52 2a c1

Description:

C0: start tag
44: print command
34: frame sequence
30 30 33 33: 33 bytes data
1b 4b 31: UTF-8
1b 21 19: double width double height based on small font size
1b 2d 00: underline off
1d 42 00: reverse off
57..6f: text content
52: odd check
2a: even check
C1: end tag

c3) big font size, reverse:

Frame Content:

c0 44 35 30 30 33 33 1b 4b 31 1b 21 00 1b 2d 00
1d 42 01 57 65 6c 63 6f 6d 65 20 74 6f 20 62 6c
75 65 62 61 6d 62 6f 6f 52 32 c1

Description:

C0: start tag
44: print command
35: frame sequence
30 30 33 33: 33 bytes data
1b 4b 31: UTF-8
1b 21 00: big font size
1b 2d 00: underline off
1d 42 01: reverse on
57..6f: text data
52: odd check
32: even check
C1: end tag

c4) double big font size:

Frame Content:

c0 44 36 30 30 33 33 1b 4b 31 1b 21 18 1b 2d 00
1d 42 00 57 65 6c 63 6f 6d 65 20 74 6f 20 62 6c
75 65 62 61 6d 62 6f 6f 52 2b c1

Description:

C0: start tag
44: print command
36: frame sequence
30 30 33 33: 33 bytes data
1b 4b 31: UTF-8
1b 21 18: double height double width based on big font size
1b 2d 00: underline off
1d 42 00: reverse off
57..6f: text data
52: odd check
2b: even check
C1: end tag

d) print Chinese text (use UTF-8 which is often used in phone)

print content:欢迎使用蓝竹打印机\n

d1) small font size:

Frame Content:

c0 44 37 30 30 34 31 1b 4b 31 1b 21 01 1b 52 30
ef bb bf e6 ac a2 e8 bf 8e e4 bd bf e7 94 a8 e8
93 9d e7 ab b9 e6 89 93 e5 8d b0 e6 9c ba 0d 0a
df eb c1

Description:

C0: start tag
44: print command
37: frame sequence
30 30 34 31: 41 bytes data
1b 4b 31: UTF-8
1b 21 01: small font
1b 52 30: Chinese language
ef..0a: text content
df: odd check
eb: even check
C1: end tag

d2) double small font size:

Frame Content:

c0 44 39 30 30 34 31 1b 4b 31 1b 21 19 1b 52 30
ef bb bf e6 ac a2 e8 bf 8e e4 bd bf e7 94 a8 e8
93 9d e7 ab b9 e6 89 93 e5 8d b0 e6 9c ba 0d 0a
df f3 c1

Description:
C0: start tag
44: print command
39: frame sequence
30 30 34 31: 41 bytes data
1b 4b 31: UTF-8
1b 21 19: double width double height based on small font size
1b 52 30: Chinese language
ef..0a: text data
df: odd check
f3: even check
C1: end tag

d3) big font size:
Frame Content:
c0 44 38 30 30 34 31 1b 4b 31 1b 21 00 1b 52 30
ef bb bf e6 ac a2 e8 bf 8e e4 bd bf e7 94 a8 e8
93 9d e7 ab b9 e6 89 93 e5 8d b0 e6 9c ba 0d 0a
df ea c1
Description:
C0: start tag
44: print command
38: frame sequence
30 30 34 31: 41 bytes data
1b 4b 31: UTF-8
1b 21 00: big font
1b 52 30: Chinese language
ef..0a: text data
df: odd check
ea: even check
C1: end tag

d4) double big font size:
Frame Content:
c0 44 30 30 30 34 31 1b 4b 31 1b 21 18 1b 52 30
ef bb bf e6 ac a2 e8 bf 8e e4 bd bf e7 94 a8 e8
93 9d e7 ab b9 e6 89 93 e5 8d b0 e6 9c ba 0d 0a
df f2 c1
Description:
C0: start tag
44: print command
37: frame sequence
30 30 34 31: 41 bytes data
1b 4b 31: UTF-8
1b 21 18: double width double height based on big font size
1b 52 30: Chinese language
ef..0a: text data
df: odd check
f2: even check
C1: end tag

e) Print Image

print Image Data:

width* height: 30 * 30 pixels

Frame content:

c0 44 32 30 31 32 35 1b 58 31 04 1e 00 00 00 00
00 00 00 00 1f ff ff e0 1c 00 00 20 18 00 00 00
18 00 00 00 18 00 00 00 18 00 00 00 18 00 00 00
18 00 00 00 19 08 22 00 19 fc 3f 00 19 f8 1f 00
1b ff ff 00 1b ff ff 00 18 00 00 00 18 00 00 00
18 00 00 20 18 00 00 60 1c 00 00 60 1c 00 00 20
1c 00 00 20 1c 00 00 20 1c 00 00 20 1c 00 00 20
1c 00 00 20 1c 00 00 20 1f ff ff e0 00 00 00 00
00 00 00 00 18 5b c1

Description:

C0: start tag
44: print command
32: frame sequence
30 31 32 35: 125 bytes data
04: image mode((30/8+1)because 1 byte means 8-bit white-bla eightge pixel)
1e: heigth (30 pixel)
00..00: image data
18: odd check
5b: even check
C1: end tag

Note: If you want to add space at left margin, you should add some bytes to the left of every pixel line, 0ne bit means one pixel
For example:
Image Data:

1B 58 31 07 2C □----------command itself, every dotline has 0x07 Bytes, total 0x2c dot lines.
00 00 07 E0 00 00 00 □----------fist dotline data
00 00 78 1F 00 00 00 □----------second dotline data
00 01 80 C0 C0 00 00
00 06 11 09 30 00 00

add space at left margin, That data may be:

1B 58 31 09 2C □----------command itself, every dotline has 0x09 Bytes, total 0x2c dot lines.
00 00 00 00 07 E0 00 00 00 □----------fist dotline data
00 00 00 00 78 1F 00 00 00 □----------second dotline data
00 00 00 01 80 C0 C0 00 00
00 00 00 06 11 09 30 00 00

It would be 16pixels margin at left

3.5.3 MSR Operation

1 [00027590] IRP_MJ_WRITE Length: 0012, Data: C0 48 36 30 30 30 32 32 30 32 30 C1
2 [00027590] IRP_MJ_READ Length: 0006, Data: 00 C0 04 C1 0D 0A
3 [00027589] IRP_MJ_READ Length: 0006, Data: 00 C0 15 C1 0D 0A

Note: 1. MSR request, set Timeout to 20 second 2. P25 immediatly back EOT
3. After 20 seconds, no card swipt, P25 back error

1 [00046059] IRP_MJ_WRITE Length: 0012, Data: C0 48 30 30 30 30 32 32 30 32 30 C1
2 [00046059] IRP_MJ_READ Length: 0006, Data: 00 C0 04 C1 0D 0A
3 [00046027] IRP_MJ_READ Length: 0168
Data: 00 C0 48 39 30 31 35 35 32 30 30 33 39 3B 36 32 32 32 30 30 31 30 30 31 31 30 34 38 31 33
36 37 31 3D 30 36 31 30 31 32 30 35 33 36 39 39 39 31 38 37 38 3F 33 30 31 30 36 3B 39 39
36 32 32 32 30 30 31 30 30 31 31 30 34 38 31 33 36 37 31 3D 31 35 36 30 30 30 30 30 30 30 30
30 30 30 30 30 30 30 30 30 30 33 35 33 36 39 39 39 32 31 36 30 30 30 30 30 36 31 30 30 30 30
30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 3D 30 30 30 30 30 30 30 30 30 30 30 30
3D 30 30 30 30 30 30 30 30 3F 08 35 C1 0D 0A

Note: 1. MSR request, set Timeout seconds
2. P25 immediatly back EOT
3. After swipt card, P25 back data

1) Send MSR Command

Frame content: C0 48 36 30 30 30 32 32 30 32 30 C1

Description:
C0: start tag
48: read msr command
36: frame sequence
30 30 30 32: 2 bytes data
32 30: 2 spaces
32: oddck

30: even check
C1: end tag

2) Receive MSR Data:

Frame content:

C0 48 30 30 30 34 34 31 30 30 30 39 25 31 48 49 4A 4B 4C 4D 3F 32 30 30 30 39 3B 32
35 36 37 38 39 30 3F 33 30 30 31 31 3B 33 34 34 35 35 36 36 37 38 3F 7F 6E C1

Description:

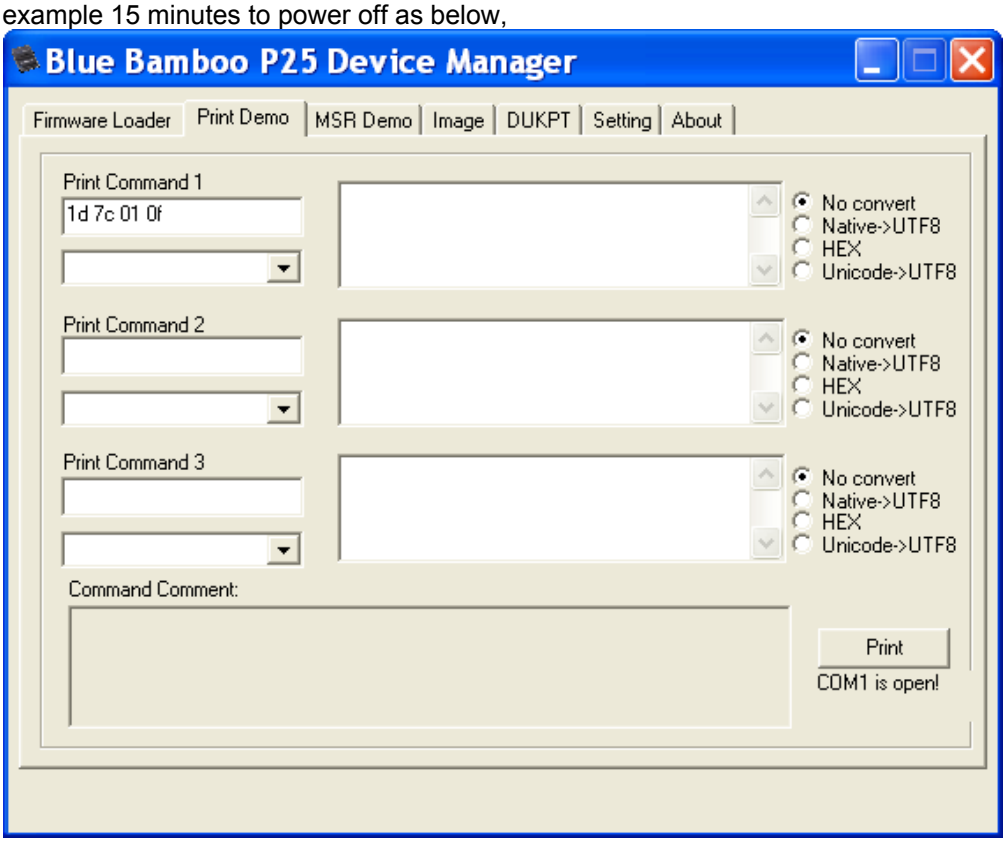
C0: start tag
48: read msr command
30: frame sequence
30 30 34 34: 44 bytes
31 30 30 30 39: Track 1, 9 bytes
25 31 48 49 4A 4B 4C 4D 3F: track1 data, <%1HIJKLM?>
32 30 30 30 39: Track 2, 9 bytes
3B 32 35 36 37 38 39 30 3F: track2 data, <;2567890?>
33 30 30 31 31: Track 3, 11 bytes
3B 33 34 34 35 35 36 36 37 38 3F: track3 data, <;344556678?>
7F:odd check
6E: even check
C1:end tag

3.5.4 Set the time to power off

GS | 1 n

[Name]	Set the time to power off				
[Format]	ASCII	GS	|	1	n
	HEX	1D	7C	01	n
	Decimal	29	124	1	n
[Range]	2<=n<=59, 0xFF				
[Description]	Set enter power off mode time, unit is minute, after P25 enter sleep mode(ref above cmd), if no operation occurs, P25 will auto power off after this time, set to 0xFF means disable this feature(never auto power off), default setting is 15 minutes.				

For example 15 minutes to power off as below,



3.6 P25i/P25i-M Communication Protocol (Applies only to P25 Made for iPhone version)

P25i/P25i-M is a special Printer that can working with iPhone/iPod Touch through serial cable. It has two operations, Print receipt and read MSR read

protocolString "com.bluebamboo.p25i", protocolIndex 1

Default bundleSeedIDString is "YT79N447RD", in fact, we would custom it to our customer's App ID that is assigned by Apple.

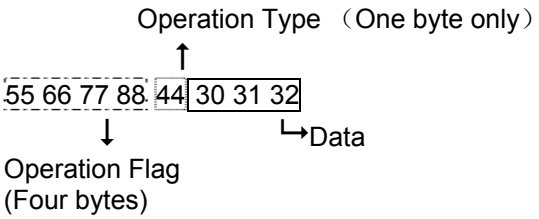
3.6.1 Operation Commands Format

There are four types of data in the frame: Operation flag, Operation Type, Parameters and Data.
•Operation Flag: Operation Flag is a constant four-byte array. For example:"55 66 77 88".
•Operation Type: There are four kinds of operation type: 44-print, 03-print result, 48-MSR, 84-MSR read result.
•Parameters and Data: Parameters and Data are operational frame, they would be changed by different operation types

3.6.1.1 Print command

Direction: iPhone--to-->P25i/P25i-M printer(s):
One print command consists of Operation Flag, Operation Type and Data.

For example:

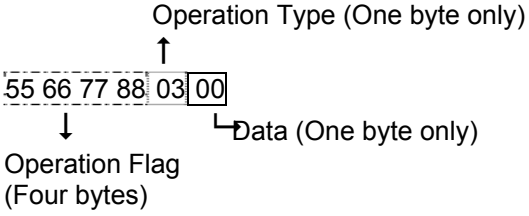


The printer will print "123" on the receipt according to above print command. In the Command,"55 66 77 88" is Operation Flag, "44" is print command and "30 31 32" is the Data that would be printed.

3.6.1.2 Print Result Command

Direction: P25i/P25i-M printer(s)--to-->iPhone
If the print command has been already processed, P25i/P25i-M printer(s) will send a print response message to iPhone with process result.

For example:

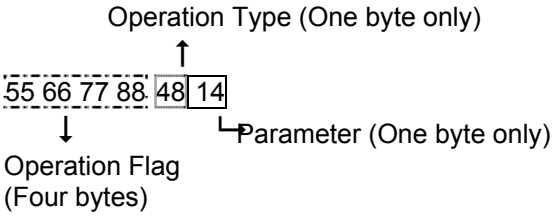


This example Command means printing is successful. In the Command “55 66 77 88” is Operation flag, “03” is the print result command and “00” is the Data that indicates the printing is successful. If the Data is not “00” that means error in printing.

3.6.1.3 MSR Read Command

Direction: iPhone--to-->P25i/P25i-M printer(s).
Send Commands to P25i/P25i-M that application needs MSR data

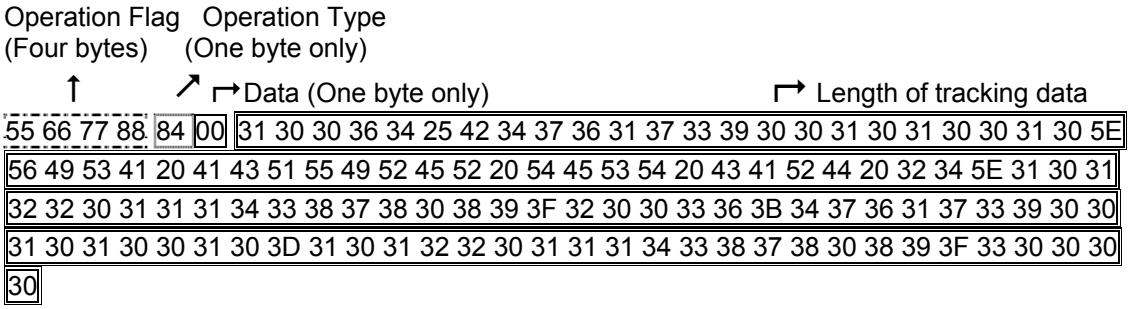
For example:



The above example means MSR read is required with 20 seconds timeout. In the Command “55 66 77 88” is Operation flag, “48” is MSR read command and “14” is timeout parameter that indicates how long(seconds) P25i/P25i-M need(s) to wait when MSR Read is processing.

3.6.1.4 MSR Result Command

Direction: P25i/P25i-M printer(s)--to-->iPhone
If MSR read has been already processed, P25 will send a print response message to iPhone with process result.
For example:



The above example means MSR Read operation is succeed. In the Command “55 66 77 88” is Operation flag,“84” is MSR result command and “00” is Data that indicates MSR Read operation is succeed. The following “0x 31” is the tracking number. After that the next four bytes indicate the length of current track. “31 30 30 36 34” means tracking one has 64 bytes data and the tracking data could be empty.

3.6.2 Operation Commands Data Dump

Following is dump data of total communication process in serial line, those data using Apple package protocol, List those data just help customer to understand, most of packages are generate automatically by iPhone app.

IRP_MJ_WRITE means iPhone--to-->P25i/P25i-M
IRP_MJ_READ means P25i/P25i-M--to-->iPhone

Following data has time info, unit is PC's tick, about 18 ticks/sec.

3.6.2.1 Print Operation

Print Command
Print a string "012"

OpenDataSessionForProtocol:
6 [00000004] IRP_MJ_WRITE Length: 0011
Data: FF 55 07 00 3F 00 01 00 00 01 B8
DevACK:
7 [00000013] IRP_MJ_READ Length: 0010
Data: FF 55 06 00 41 00 01 00 3F 79

iPodDataTransfer: print "012"
8 [00000013] IRP_MJ_WRITE Length: 0018
Data: FF 55 0E 00 43 00 02 00 00 55 66 77 88 44 30 31 32 1C
DevACK: means P25 accept last command
9 [00000023] IRP_MJ_READ Length: 0010
Data: FF 55 06 00 41 00 02 00 43 74
DevDataTransfer: print finished OK
10 [00000045] IRP_MJ_READ Length: 0016
Data: FF 55 0C 00 42 00 01 00 00 55 66 77 88 03 00 F4
ACK:
11 [00000045] IRP_MJ_WRITE Length: 0010
Data: FF 55 06 00 02 00 01 00 42 B5
CloseDataSession:
12 [00000058] IRP_MJ_WRITE Length: 0010
Data: FF 55 06 00 40 00 03 00 00 B7
DevACK:
13 [00000068] IRP_MJ_READ Length: 0010
Data: FF 55 06 00 41 00 03 00 40 76

3.6.2.2 MSR Operation

OpenDataSessionForProtocol:
6 [00000003] IRP_MJ_WRITE Length: 0011
Data: FF 55 07 00 3F 00 01 00 00 01 B8
DevACK:
7 [00000013] IRP_MJ_READ Length: 0010
Data: FF 55 06 00 41 00 01 00 3F 79
iPodDataTransfer: msr timeout 20 seconds
8 [00000013] IRP_MJ_WRITE Length: 0016
Data: FF 55 0C 00 43 00 02 00 00 55 66 77 88 48 14 99
DevACK: P25 accept last command
9 [00000022] IRP_MJ_READ Length: 0010
Data: FF 55 06 00 41 00 02 00 43 74
DevDataTransfer: after swipe card, msr result
10 [00000287] IRP_MJ_READ Length: 0131
Data: FF 55 7F 00 42 00 01 00 00 55 66 77 88 84 00 31 30 30 36 34 25 42 34 37 36 31 37
33 39 30 30 31 30 31 30 30 31 30 5E 56 49 53 41 20 41 43 51 55 49 52 45 52 20 54
45 53 54 20 43 41 52 44 20 32 34 5E 31 30 31 32 32 30 31 31 31 34 33 38 37 38 30
38 39 3F 32 30 30 33 36 3B 34 37 36 31 37 33 39 30 30 31 30 31 30 30 31 30 3D 31
30 31 32 32 30 31 31 31 34 33 38 37 38 30 38 39 3F 33 30 30 30 30 0F

ACK:
11 [00000287] IRP_MJ_WRITE Length: 0010
Data: FF 55 06 00 02 00 02 00 42 B4

CloseDataSession:
12 [00000300] IRP_MJ_WRITE Length: 0010
Data: FF 55 06 00 40 00 03 00 00 B7
DevACK:
13 [00000310] IRP_MJ_READ Length: 0010
Data: FF 55 06 00 41 00 03 00 40 76

4 P25 Reference Code

4.1 Java for Blackberry

This kit of files offers source code that BLUE BAMBOO has provided for customer reference. Developers can copy and modify this code to accelerate their development. Please see the Software license agreement at the end of this text which applies to all Blue Bamboo software products. When using this software, you are agreeing to the Software license agreement.

For questions or comments, please contact support@bluebamboo.com.

REVISION HISTORY

Jan 23rd, 2008 release, version 1.0.0

Summary of included files:

Deployed: Includes compiled jar and jad files.
Res: Includes logo and other resource
Src/com: Includes program source code.
Src/net: Includes Bluetooth control source code.

Instructions

1. This is a demo source code for P25 on J2ME platform.

2. This demo can be compiled in any Java development environment.

3. The mobile phone Java requirement is MIDP 2.0 and CLDC 1.0.

4.2 Java for Motorola

REVISION HISTORY

December 10, 2007 release, version 1.0.1

Summary of included files:

P25_Console_Demo: Includes printing source code
P25_Console_Msr: Includes sourced code to access the MSR reader
P25_Print_Demo.bat: REM this file should be run in the directory that contain execute file P25_Console_Demo
P25_Print_Demo.txt: Please input print content or ESC command in this file.
P25_Console_Demo.sln: Visual C++ project file

Instructions

1. This Source Code consists of: two code samples located in folders'P25_Console_Demo' and 'P25_Console_Msr', P25_Print_Demo.bat and P25_Print_Demo.txt files described above.

2. This demo should be compiled in Visual C++ Studio, users of other development tools can print string and test ESC command.

4.3 Windows Mobile 5.x and 6.x

REVISION HISTORY

January 7th, 2008 release, version 0.1.0

Summary of included files:

SerialApi.cs
P25.cs

Instructions

1. This Source Code consists of: 2 source code sample file for Window Mobile. These source code are written in C#.

2. This demo should be compiled in Visual Studio 2005, the source code include 4 basic functions. Open port, Transfer data, sending data and close connection. Developer can copy related source code.

4.4 Windows CE 4.2

4.4.1 Introduction

This'code has been extended from Blue Bamboo's Developer Network as a reference for customer use. Blue Bamboo is not the author of this code and takes no liability in warranting/supporting it. Please review the Blue Bamboo Software License Agreement packaged with this source code.

The document and associated sample application illustrates the use of a client-side Windows CE–based Bluetooth port emulator, device discovery, and client-side SDP queries.. Upon start of the application an icon appears in the tray on the DSVIIxx. When the icon is clicked, a dialog box appears. From this dialog the user can discover devices, then select a device from the list. In the process, it runs inquiries, name queries and SDP queries. After a Bluetooth device is discovered with serial port profile (SPP) capabilities, the pairing and authentication process is initiated. If those processes complete successfully, a Bluetooth printer profile appears under the printer program. When a new connection is established through this profile, the target printer can be selected from the application.

4.4.2 Setting up a Serial Port profile (SPP) Connection

We demonstrate Bluetooth printing using the SPP profile, this is the only one Windows CE natively supports. The basic steps to set up a Bluetooth SPP connection are:

• **Inquiry**: Find nearby devices in range. This will result with all access points responding with their address.

• **Paging**: The device will invoke a baseband procedure called paging. This results in synchronization of the device with the access point, in terms of its clock offset and phase in the frequency hop, among other required initializations.

• **Link establishment**: The LMP will now establish a link with the access point. As the application in this case is printing, an SPP link will be used. Various setup steps will be carried out as described below.

• **Service Discovery**: The LMP will use the SDP (Service Discovery Protocol) to discover what services are available from the access point, in particular whether SPP access or access to the relevant host is possible from this access point or not. Let us assume that the service is available, otherwise, the application cannot proceed further. The information regarding the other services offered at the access point may be presented to the user.

• **L2CAP channel**: With information obtained from SDP, the device will create an L2CAPchannel to the access point. This may be used by the application or another protocol like RFCOMM may be run over it.

• **Security**: If the access point restricts its access to a particular set of users or otherwise offers secure mode communications to people having some prior registration with it, then at this stage, the access point will send a security request for “pairing”. This will be successful if the user knows the correct PIN code to access the service. Note that the PIN is not transmitted over the wireless channel but another key generated from it is used, so that the PIN is difficult to compromise. Encryption will be invoked if secure mode is used.

• **SPP**: COM Port Emulation is used after a virtual COM port is created. (*The reference code uses COM7: as the default virtual COM port*)

4.4.3 Needed Components for this Demonstration

The following is a list of the needed components to duplicate this demonstration:

• Host Device Running P25 with Bluetooth module supporting SPP profile
• Reference Code (virtual COM port mapping of SPP profile)
• Blue Bamboo P25 Printer

4.4.4 Procedure

Using the Reference Code:

1. When the application is started it first appears minimized in the tray.
    a. If the application was previously run and a device bonded to the virtual port, the connection will automatically reestablish.
2. Open up the dialog by clicking icon in tray.
    a. If the virtual port connection is established, there will be a message displayed in the lower left of the dialog indicating the connected device address, the virtual com port and whether the connection is running or stopped.
    b. If there is no virtual port connection established the message will state “No Bonding”
3. To establish a connection to a device through the virtual com port:
    a. Click “Scan Device”
    b. Highlight the desired device from list.
    c. Click “Select Device”; Dialog will close if connected.
    d. Open dialog to verify bonding.
    e. Click “Cancel” to close dialog.

NOTE: to be able to print from within a Windows CE application (eg: Wordpad), the WinCE PCL printer driver must be included/built into the platform (it is included as of WinCE Version 4.20-1.80.)

4.5 Software License Agreement

This is a legal agreement between you (either as an individual or as an authorized representative of your employer) and BLUE BAMBOO. It concerns your rights to use this file and any accompanying written materials (the P25 Reference Code). In consideration for BLUE BAMBOO allowing you to access the P25 Reference Code, you are agreeing to be bound by the terms of this Agreement. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, DO NOT INSTALL THE P25 Reference Code. If you change your mind later, stop using the P25 Reference Code and delete all copies of the P25 Reference Code in your possession or control. Any copies of the P25 Reference Code that you have already distributed, where permitted, and do not destroy will continue to be governed by this Agreement. Your prior use will also continue to be governed by this Agreement.

4.5.1 Definitions

(i) " P25 Reference Code" means all files including source code, .C,.h, files, and documentation contained within the installation package.

(ii)"DERIVATIVE WORKS" shall mean a work which is based on one or more preexisting works, such as a revision, modification, translation, abridgment, condensation, expansion, collection, compilation or any other form in which such preexisting works may be recast, transformed or adapted.

4.5.2 License

For as long as LICENSEE complies with its obligations under this Agreement, BLUE BAMBOO hereby grants to LICENSEE for the sole purposes of incorporating the corresponding P25 Reference Code with BLUE BAMBOO's transaction terminal devices, a personal, worldwide, non-exclusive, nontransferable (except as set forth below), paid-up, royalty-free license and right:

(i) to use, execute, perform and copy the P25 Reference Code, and

(ii)to prepare DERIVATIVE WORKS using the P25 Reference Code. For as long as licensee complies with its obligations under this Agreement, BLUE BAMBOO hereby grants to LICENSEE for the sole purposes of distributing LICENSEE's products developed with BLUE BAMBOO's P25 Reference Code, a personal, worldwide, non-exclusive, paid-up, royalty-free license and right:

(i) to distribute DERIVATIVE WORKS

(ii)to sublicense to others the right to use DERIVATIVE WORKS This license is not a sale of any interest in the P25 Reference Code's Copyright or any copy thereof. LICENSEE agrees not to remove or destroy any Copyright notices, proprietary markings or confidential legends placed upon, contained within or associated with the P25 Reference Code.

4.5.3 Non-disclosure

LICENSEE hereby agrees (i) to hold Blue Bamboo Confidential Information (the REFERENCE CODE and all other BLUE BAMBOO information marked with "Blue Bamboo CONFIDENTIAL" or similar legend) furnished hereunder in confidence. These confidential obligations shall not apply to information that: (a) is in or enters the public domain without breach of this Agreement by LICENSEE; or (b) is disclosed without restrictions by a third party to LICENSEE; or (c) is independently developed by LICENSEE; or (d) is already known to LICENSEE at the time of disclosure under this Agreement as shall be proved by contemporaneous written records; or (e) is disclosed by BLUE BAMBOO to any other person or entity without similar restrictions, or (f) is required to be disclosed by law, governmental agency or court order, except where such disclosure is disclosure of P25 Reference Code as Restricted Rights under Section 4.

4.5.4 U.S. Government Disclosure

LICENSOR will allow LICENSEE to disclose the P25 Reference Code SOURCE CODE to those employees of the U.S. Government that require access to the P25 Reference Code SOURCE CODE to approve the LICENSEE 'S Products; LICENSOR understands that the U.S. Government is legally restricted in its ability to distribute contractor proprietary information. LICENSEE will maintain a legally binding contract with the U.S. Government to enforce the confidentiality of the P25 Reference Code and require the U.S. Government not to disclose the P25 Reference Code SOURCE CODE to any and all persons; and will treat the P25 Reference Code SOURCE CODE as Restricted Rights as stated in the Defense Federal Acquisition Regulation Supplement and under DFARS clause #252.227-7014.

4.5.5 Limitation of Liability

This Code IS PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT SHALL BLUE BAMBOO BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, LOSS OF INFORMATION AND, INSOFAR AS IS LEGALLY POSSIBLE, CONSEQUENTIAL OR INCIDENTAL DAMAGES) ARISING OUT OF THE USE OF OR INABILITY TO USE THE P25 Reference Code, EVEN IF BLUE BAMBOO HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BLUE BAMBOO SHALL NOT BE RESPONSIBLE FOR ANY ERRORS THAT MAY APPEAR IN THE P25 Reference Code OR DERIVATIVE WORKS NOR SHALL BLUE BAMBOO BE UNDER ANY OBLIGATION TO SUPPORT OR UPDATE THE P25 Reference Code. BLUE BAMBOO RETAINS THE RIGHT TO MAKE CHANGES TO ITS P25 Reference Code AT ANY TIME, WITHOUT NOTICE TO LICENSEE.

4.5.6 Maintenance

BLUE BAMBOO shall be under no obligation to assist Licensee to use, execute, perform or copy the P25 Reference Code, or to provide support to Licensee of the P25 Reference Code, or to provide maintenance, correction, modification, enhancement, or upgrades to the P25 Reference Code. BLUE BAMBOO may however provide such support, maintenance, correction, modification, enhancement or upgrades in its sole discretion, and, if provided, such support, maintenance, correction, modification, enhancement, or upgrades shall be considered part of the P25 Reference Code and shall be subject to all terms and conditions of this Agreement.

4.5.7 Termination

This Agreement and the licenses granted by BLUE BAMBOO herein shall terminate immediately without notice from BLUE BAMBOO or by a judicial resolution, if LICENSEE fails to comply with any provisions of this Agreement. Following termination of this Agreement, LICENSEE covenants that it shall only use DERIVATIVE WORK(S) of the P25 Reference Code to incorporate BLUE BAMBOO's transaction terminal products with Licensee's products. Any other use of DERIVATIVE WORK(S) of the P25 Reference Code is strictly prohibited.

4.5.8 Taxes

The amount of any taxes of any kind, if any is applicable to this transaction or to the use or possession of the P25 Reference Code covered hereby shall be paid by LICENSEE. The parties agree that the tangible portion of the property delivered and to be delivered by BLUE BAMBOO to LICENSEE or by LICENSEE to BLUE BAMBOO is valued at one hundred dollars ($100.00).

4.5.9 Governing Law

This Agreement shall be governed and construed in accordance with the laws of the Hong Kong S.A.R. as applied to contracts entered into in the Hong Kong S.A.R by a corporation based in Hong Kong S.A.R.

4.5.10 Damage Limitation

INDEPENDENTLY OF ANY OTHER REMEDY LIMITATION HEREOF AND NOTWITHSTANDING ANY ESSENTIAL PURPOSE OF ANY SUCH LIMITED REMEDY, IT IS AGREED THAT IN NO EVENT SHALL BLUE BAMBOO BE LIABLE FOR SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND UNDER THIS AGREEMENT.

4.5.11 Transferability

LICENSEE may not divide its interests under this Agreement and transfer same to different parties; however, LICENSEE may assign and transfer all its rights under this Agreement solely to a party controlling, controlled by or under common control with LICENSEE, a successor in interest in the event of a merger, consolidation or sale of substantially all of the LICENSEE'S assets or stock, or to a successor to the portion of its business that designs or markets products incorporating the BLUE BAMBOO P25 Reference Code.

4.5.12 Intellectual Property Ownership

BLUE BAMBOO shall retain title to and ownership of all proprietary rights with respect to the P25 Reference Code and copies thereof. LICENSEE shall retain title to and ownership of all proprietary rights in DERIVATIVE WORKS of the P25 Reference Code made by or for LICENSEE, and all copies thereof, excluding the portions owned by BLUE BAMBOO made reference to under the previous sentence. Except as expressly provided herein, BLUE BAMBOO does not grant any express or implied right to Licensee under BLUE BAMBOO's patents, Copyrights, trademarks, or trade secrets.

4.5.13 Survival Clauses

The following provisions shall survive expiration or termination of this Agreement for any reason: Section 3 (Non-disclosure), Section 5 (Limitation of Liability), Section 10 (Damage Limitation), Section 12 (Intellectual Property Ownership).

4.5.14 Entire Agreement

This Agreement and its Exhibits contain the entire Agreement and understanding between the parties with respect to the subject matter hereof and merges and supersedes all prior agreements, understandings and representations. No addition or modification to the Agreement is valid unless made in writing and signed by both parties hereto. To the extent that this Agreement contains a provision that contradicts any provision in the Exhibits hereto, this Agreement shall control. This Agreement shall be construed and controlled by the law of the Hong Kong S.A.R. In the event of any dispute hereunder, the parties agree to jurisdiction in the Hong Kong S.A.R.

5 Revision History

Revision 1.0 (June 01, 2007)	Initial Release.
Revision 1.1 (February 18, 2008)	Add Print Commands in Section 2.1 Add Communication protocol in Section 3.
Revision 1.2 (June 16, 2008)	Add P25 Reference Code and the Software License Agreement.
Revision 1.3 (August, 2008)	Add commands of P25.
Revision 1.4 (September, 2008)	Add data encryption in Section 3.4.6
Revision 1.5 (October, 2008)	Add additional clear text data elements in Section 3.4.6
Revision 1.6 (October, 2008)	Separate encrypted track data in Section 3.4.6
Revision 1.7 (October 31, 2008)	Dukpt add clear text info like last 4 digit card
Revision 1.8 (November 3, 2008)	Merge dukpt set key 3.4.9 from previous version
Revision 1.9 (November 4, 2008)	Add BlueTooth discovery control command
Revision 2.0 (Feb 27, 2009)	Add print pdf417 bar code command
Revision 2.1 (Mar 11, 2009)	Add set default font command
Revision 2.2 (Aug 28, 2009)	Delete the word “DUKPT” from the title of “DATA in clear text-DUKPT” and “DATA encrypted-DUKPT” in section 3.4.6
Revision 2.3 (Oct 01, 2009)	Add the examples in section 3.5
Revision 2.4 (Dec 04, 2009)	Deprecate the sleep mode, modify Sleep power off command, add section 2.9 ESC/GS Commands with Respond Info.
Revision 2.5 (Dec 23, 2009)	Add section 2.10 Enter/Quit Bridge Mode Commands
Revision 2.6 (Jan 12, 2010)	Add the Set UART 1(serial port or USB port) Baud Rate commands in section 2.8
Revision 2.7 (Jan 26, 2010)	Delete section 3.6 and examples for sleep mode setting Update section 2.3 Character Commands
Revision 2.8 (June.1,2010)	Add P25 MK/SK encryption
Revision 2.9 (June 24, 2010)	Add P25i/P25i-M Communication Protocol
Revision 3.0 (July 30, 2010)	Update the format
Revision 3.1 (September 13, 2010)	Change the print command in Section 2.1 “Print Commands”
Revision 3.2 (October 2, 2010)	Add section 2.11 Realtime Commands
Revision 3.2.1 (April 21, 2011)	Add commands GS t n, GS { n, ESC |n, ESC w n in page 6.
Revision 3.3 (July 13, 2011)	Update some commands and add new command GS v 0
Revision 3.4 (September 20, 2011)	Correction.
Revison 3.4.1 (September 22, 2011)	Correction.