SERIES E2000 PROTOCOL

Revision 1.03

Feb 23, 2004

CONTENT

Version Information

Revision 1.03 support reading and writing product S/N. This information is helpful for maintenance. Refer to "Commands" for detail.
Revision 1.02 adds DECOUNTER control. The DECOUNTER control share the control code F0H with the TEMPERATURE control. Refer to "Control Characters" for detail.
Revision 1.01 adds TEMPERATURE control. Only those signs with temperature sensors can display temperature correctly. Refer to "Control Characters" for detail.

COM Port Settings

8 data bits, 1 stop bit, no parity.
Baud rate: 1200, 2400, 4800, 9600(Default), 14400, 19200, 28800, 57600

Frame Format

All data must transmit in frame format. Maximum 1000 bytes data can transmit in one frame. Data larger then 1000 bytes must transmit in two or more frames.

SYNC	LENGTH	HEAD	DATA	VERIFY
(5 bytes)	(2 bytes)	(8 bytes)	(1~1000 bytes)	(2 bytes)

SYNC

Frame synchronization string, must be '$WRL>'.

4 command chasacter goes between head & data

LENGTH

Total size of HEAD field and DATA field. Low byte first.

HEAD

TYPE	ADDRRESS	NUMBER	RESERVED
(1 byte)	(1 byte)	(1 byte)	(5 bytes)

TYPE

Sign type, must be 41H for series E2000.

ADDRESS

Serial address. Value from 1 to 255. A frame with zero address will be receipt by all sign, which is called broadcast mode.

NUMBER

Frame number. During communication conversation, NUMBER of the first frame should be ignored; other frame number should be checked to see if the frame is repeated. A repeated frame number equal to the frame number previous received. When a repeated frame received, answer the frame again and discard it.

RESERVED

Reserved for future use. Must be zero.

DATA

The actual data being transmit in frame.

VERIFY

Sum up all bytes in HEAD and DATA field and then perform a NOT operation, use the result to verify the frame data. When the VERIFY field is not equal to the calculated result, the whole frame should be discarded.

Data Transferring

Data is a stream of bytes. They must be packet into one or more frames to ensure correct transferring. The actual data is always sequence of commands, which tell the receiver what to do.

Broadcast Mode

Set the ADDRESS field to zero, the frame will be transmit in broadcast mode and will be received by all signs. In broadcast mode, the receiver does not need to answer the frame. Otherwise, the transmitter needs to delay 100ms after transferring every frame.

Point-to-Point Mode

Transmitter

Send out one frame.

Abandon all data has been received.

Wait for answer code. The answer code is one byte FAH, no frame format. If no answer code has been received in 500ms, resend the frame.

If the frame has been send for three times but still no answer, the transmitter failed. Otherwise, increase the frame number and finish frame transmitting.

Receiver

After receive one frame correctly (The VERIFY field is correct, and the TYPE and ADDRESS fields are meet), answer the frame with one byte FAH.

If the communication conversation has not start, mark the begging of communication conversation and initialize the frame counter with the NUMBER field. Otherwise check repeated frame by compare the NUMBER field with the frame counter.

Handle the data and increase frame counter when the frame is not repeated or the first frame in conversation.

Commands

PC software transmits files, schedule to the sign or setup the sign by issue one or more write commands. It can also read the files and schedule from the sign by issue a read command. You can transmit several write command and one read command in one frame. The read command must be the last command in the frame. All data after a read command will be ignored.

After receiving a read command, the sign will response with one or more frames of relative write command. PC software become receiver and must answer these frames and handle the write command.

Command	Meaning	Property	Parameter
41H('A')	Set Serial Address	Write	Serial Address
42H('B')	Set Baud Rate	Write	Baud Rate
43H('C')	Change Baud Rate (Auto recover when conversation end)	Write	Baud Rate
44H('D')	Write Product S/N	Write	Product S/N
45H('E')	Read Product S/N	Read	None
61H('a')	Transmit Text File	Write	Text File
62H('b')	Transmit Graphics File	Write	Graphics File
63H('c')	Transmit Schedule	Write	Schedule
64H('d')	Set Date and Time	Write	Date and Time
65H('e')	Clear Memory	Write	None
66H('f')	Clear Schedule	Write	None
67H('g')	Clear Graphics	Write	None
68H('h')	Clear Texts	Write	None
69H('i')	Require Memory Info	Read	None
6AH('j')	Set Default File	Write	File Name
6BH('k')	Require Text Directory	Read	None
6CH('l')	Require Graphics Directory	Read	None
6DH('m')	Require Text File	Read	File Name
6EH('n')	Require Graphics File	Read	File Name
6FH('o')	Require Schedule	Read	None
70H('p')	Transmit Text Directory (from PC only)	Write	File Directory
71H('q')	Transmit Graphics Directory (from PC only)	Write	File Directory
72H('r')	Report Memory Info (from PC only)	Write	Memory Info
73H('s')	Display Testing (End communication conversation)	Write	None
74H('t')	Turn OFF the sign (End communication conversation)	Write	None
75H('u')	Turn ON the sign (End communication conversation)	Write	None
76H('v')	Reserved		
77H('w')	Reserved		
78H('x')	Communication Status ON/OFF (No status display when OFF, default is ON)	Write	None
79H('y')	Get Default File Name	Read	None
7AH('z')	End Communication Conversation	Write	None

Command Parameters

I. Serial Address

Value: 1 to 255

Size: 1 byte

II. Baud Rate

Value: 0 = 1200, 1 = 2400, 2 = 4800, 3 = 9600, 4 = 14400, 5 = 19200, 6 = 28800, 7 = 57600

Size: 1 byte

III. Date and Time

SECOND	MINUTE	HOUR	DAY	MONTH	YEAR

SECOND 1byte, binary 0-59

MINUTE 1 byte, binary 0-59

HOUR 1byte, binary 0-23

DAY 1 byte, binary 1-31

MONTH 1 byte, binary 1-12

YEAR 1 byte, binary 0-99 (year 2000 to year 2099)

Size: 6 bytes

IV. File Name

Maximum 8 characters make up of numbers, uppercase letters or "!#$%&'()-.@_". Fill with zero when less then 8 characters.

Size: 8 bytes

V. Text File

HEADER				CONTENT			VERIFY
Type	Name	Size	Time	EFH	CHARACTERS	EFH	

1□ HEADER

File Type, 1 byte, 0 = Text

File Name, 8 bytes, refer to "File Name"

File Size, 2 bytes, total size of HEADER, CONTENT and VERIFY

File Time, 6 bytes, refer to "Date and Time"

2□ CONTENT

Make up of text characters and control characters. Should always begin and end with EFH.

20H-7FH ASCII character

80H-EEH International character

EFH mark the beginning and ending of text content

F0H TEMPERATURE control character

F1H mark the beginning and ending of METHOD control character

F2H mark the beginning and ending of FONT control character

F3H mark the beginning and ending of COLOR control character

F4H mark the beginning and ending of SPEED control character

F5H mark the beginning and ending of PAUSE control character

F6H mark the beginning and ending of TIME control character

F7H reserved for future control character

F8H mark the beginning and ending of GRAPHICS control character

F9H reserved for future control character

FAH mark the beginning and ending of SPECIAL control character

FBH	mark the beginning and ending of WINDOW control character
FCH	mark the beginning and ending of VSPACE control character
FDH	mark the beginning and ending of INCLUDE control character
FEH	reserved for future control character
FFH	reserved for future control character

Refer to "Text Characters" and "Control Characters" for detail.

3□ VERIFY, 2 bytes, sum up all bytes in HEADER and CONTENT, and then perform NOT operation.

Size: equal to "File Size" in HEADER

VI. Graphics File

HEADER				CONTENT				VERIFY
Type	Name	Size	Time	Width	Height	Reserved	Data	

1□ HEADER

File Type, 1 byte, 1 = Graphics

File Name, 8 bytes, refer to "File Name"

File Size, 2 bytes, total size of HEADER, CONTENT and VERIFY

File Time, 6 bytes, refer to "Date and Time"

2□ CONTENT

Graphics Width: 1 byte, limited to the screen width

Graphics Height: 1 byte, limited to the screen height

Reserved: 1 byte, must be zero

Graphics Data: refer to "The Graphics Data"

3□ VERIFY, 2 bytes, sum up all bytes in HEADER and CONTENT, and then perform NOT operation.

Size: equal to "File Size" in HEADER

VII. Schedule

1□ Task Count, 1 byte, 1-30

2□ The First Task

Task Property, 1 byte, Bit1..Bit0: Priority, 00 = LOW; 01 = NORMAL; 10 = HIGH; Bit7: 1=Enabled, 0=Disabled. Bit6..Bit2: Reserved.

Task Start Date: one byte binary day, month and two bytes binary year.

Task End Date: see "Task Start Date"

Task Start Time: one byte binary, minute and hour.

Task End Time: see "Task Start Time"

Weekly Setting: 1 byte, bit3..bit0 = start day of week; bit7..bit4 = end day of week.

Text to Run: 8 bytes, see "File Name"

3□ Other Tasks

Size: (Task Count)*22+1

VIII. File Directory

1□ File Count: 1 byte, 1-250

2□ Header of the first file

File Type, 1 byte, 0=Text; 1=Graphics

File Name, 8 bytes, see "File Name"

File Size, 2 bytes

File Time, 6 bytes, see "Date and Time"

3□ Headers of other files

Size: (File Count)×17+1

IX. Memory Info

1□ Used text size, 2 bytes

2□ Free text size, 2 bytes

3□ Used graphics size, 2 bytes

4□ Free graphics size, 2 bytes

Size: 8 bytes

X. Product S/N

Zero ended string. Maximum 64 characters.

Text Characters

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	¢	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{	|	}		
80	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
90	É	æ	Æ	ô	ö	ò	û	ù	ÿ	ö	Ü	¢	£	¥	₧	ƒ
A0	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¡	«	»	∅	ć	Ć
B0	č	Č	đ	Đ	š	ž	Ž	ß	Š	ß	Á	À	Ã	ã	Ð	ſ
C0	ð	ő														

TEMPERATURE control

F0H	Temperature Index	F0H

Temperature Index:

0 Centigrade temperature, such as 20°C

1 Fahrenheit temperature, such as 68°F

2 Kelvin temperature, such as 293K

NOTE: the code for symbol (°) is A9H

DECOUNTER control

F0H	Decounter Index	Date and Time	F0H

Decounter Index:

10H count in days

11H count in hours

12H count in minutes

13H count in seconds

Date and Time: refer to "**Command Parameter □. Date and Time**"

NOTE: the DECOUNTER control share the control code F0H with the TEMPERATURE control. The Decounter Index starts from 10H. The decounter can display number within 65535, a larger number will be displayed as '-----'. 0 will be displayed when the time passed.

METHOD control

F1H	Method Index	F1H

Method Index:

0 [AUTO] Automatic display (Default), randomly selects one of available display modes

1 [SLIDE←] Slide to left

2 [SLIDE→] Slide to right

3 [SLIDE<>] Slide from center

4 [SLIDE><] Slide to center

5 [SLIDE↑]Slide up

6 [SLIDE↓]Slide down

7 [COVER←] Cover to left

8 [COVER→] Cover to right

9 [COVER<>] Cover from center

10 [COVER><] Cover to center

11 [COVER↑] Cover up
12 [COVER↓] Cover down
13 [ROLL←] Roll to left
14 [ROLL→] Roll to right
15 [ROLL<>] Roll from center
16 [ROLL><] Roll to center
17 [ROLL↑] Roll up
18 [ROLL↓] Roll down
19 [INSLID1] Interlaced slide 1
20 [INSLID2] Interlaced slide 2
21 [INSLID3] Interlaced slide 3
22 [INSLID4] Interlaced slide 4
23 [INROLL1] Interlaced roll 1
24 [INROLL2] Interlaced roll 2
25 [INROLL3] Interlaced roll 3
26 [INROLL4] Interlaced roll 4
27 [INROLL5] Interlaced roll 5
28 [INROLL6] Interlaced roll 6
29 [SHUTT1] Shutter mode 1
30 [SHUTT2] Shutter mode 2
31 [SHUTT3] Shutter mode 3
32 [SHUTT4] Shutter mode 4
33 [JUMP] Jump mode
34 [IMMED] Immediate mode
35 [SNOW] Fall like snow
36 [RANDOM] Random points appear
37 [SHOOT] Shoot mode
38 [EXPLODE] Explode mode
39 [TWINKLE] Twinkle mode
40 [FLASH] Flash mode
41 [PACMAN] Pac-man mode
42 [SCROLL] Scroll large continuous messages
43 [ROTATE] Rotate large continuous messages

FONT control

F2H	Font Index	F2H

Font Index (Depend on the font file):

0 [SS5] Sans Serif, size 5
1 [SS7] Sans Serif, size 7 (Default)
2 [SF7] Serif, size 7
3 [SF10] Serif, size 10
4 [SS16] Sans Serif, size 16

5 [SF16] Serif, size 16
6 [TM16] Times New Roman, size 16
7 [AR16] Arial, size 16
8 [SMA] Small Fonts, size 8

COLOR control

F3H	Color Index	F3H

Color Index:

0 [ACL] Auto Color (Default)
1 [RED] Red
2 [GRN] Green
3 [YEL] Yellow
4 [RB1] Rainbow 1
5 [RB2] Rainbow 2
6 [RB3] Rainbow 3
7 [RB4] Rainbow 4
8 [RB5] Rainbow 5
9 [MIX1] Mixture 1
10 [MIX2] Mixture 2
11 [MIX3] Mixture 3
12 [MIX4] Mixture 4
13 [MIX5] Mixture 5
14 [MIX6] Mixture 6
15 [MIX7] Mixture 7
16 [MIX8] Mixture 8

SPEED control

F4H	Speed Index	F4H

Speed Index:

0 to 7 represent [SPEED1] to [SPEED8]. [SPEED1] is the slowest. [SPEED5] is the default speed.

PAUSE control

F5H	Pause Index	F5H

Pause Index:

0 [NO PAU] no pause
1 [PAU 1S] pause 1 second

2	[PAU 2S]	pause 2 seconds
3	[PAU 3S]	pause 3 seconds
4	[PAU 5S]	pause 5 seconds
5	[PAU 10S]	pause 10 seconds
6	[PAU 30S]	pause 30 seconds
7	[PAU 60S]	pause 60 seconds

TIME control

F6H	Time Element	F6H

Time Element:

00H	Time, 24-hour mode, format 23:59
01H	Time, 24-hour mode, format 23:59:59
02H	Time, 12-hour mode, format 11:59 pm
03H	Time, 12-hour mode, format 11:59:59 pm
10H	Date, format 31 Dec 2001
11H	Date, format Dec 31, 2001
12H	Date, format 31/12/2001
13H	Date, format 12/31/2001
14H	Date, format 2001.12.31
20H	Hour, 24-hour mode
21H	Hour, 12-hour mode
30H	Minute
40H	Second
50H	AM/PM
51H	A/P
52H	am/pm
53H	a/p
60H	day of week, format Mon
61H	day of week, format Monday
62H	day of week, format MON
63H	day of week, format MONDAY

70H	day of month
80H	month, format Dec
81H	month, format December
82H	month, format 12
83H	month, format DEC
84H	month, format DECEMBER
90H	year, format 2001
91H	year, format 01

SPECIAL control

FAH	Special Control	FAH

Special Control:

0	[BOLD] or [/BOLD]	Starting or ending of bold font
1	[WIDE] of [/WIDE]	Starting or ending of wide font
2		Reserved
3		New line
4	--PAGE--	New page

WINDOW control

Use WINDOW control to limit the display output to a given area on the screen, and how text align in this area.

FBH	XPosition	YPosition	Width	Height	Align	FBH

Minimum width: 24

Minimum height: 5

Align:

0	left align	top align	
1	center align	top align	
2	right align	top align	
3	left align	middle align	
4	center align	middle align	(default)
5	right align	middle align	
6	left align	bottom align	
7	center align	bottom align	
8	right align	bottom align	

VSPACE control

Use VSPACE control to set the spacing between character lines.

FCH	Spacing	FCH

Zero spacing [VS 0] is the default setting.

INCLUDE control

Use INCLUDE control to "call" other text file. The current font, color and so on, will be saved and reset, then the included text will be run. At last, the font, color and so on, will restore.
The include deep, including plus being included, is limit to 20. Cross including is inhibited.

FCH	Text File Name (8 bytes)	FCH

GRAPHICS control

Use GRAPHICS control to insert graphics in text file.

F8H	Graphics File Name (8 bytes)	F8H

Font File Structure

1. Fonts library name, zero ended string, must be "SDVFONTS"
2. Fonts library version, 1 byte
3. Font Data 1
4. Font Data 2
5. .....
6. Zero marks the end of fonts library.

Font Data Structure

1. Font name, zero ended string, maximum 4 characters
2. Font height, 1 byte
3. Line size of character matrix, 1 byte
4. Start character, 1 byte
5. Character count
6. Characters matrix data (the first byte of the character matrix data is the character's width)

Series E2000's fonts

0	SS5	Sans Serif, 5 pixels height
1	SS7	Sans Serif, 7 pixels height
2	SF7	Serif, 7 pixels height
3	SF10	Serif, 10 pixels height
4	SS16	Sans Serif, 16 pixels height
5	SF16	Serif, 16 pixels height
6	TM16	Times New Roman, 16 pixels height
7	AR16	Arial, 16 pixels height
8	SMA	Small Fonts, 8 pixels height

The Graphics Data

Graphics data is a sequence of pixels. Each pixel occupies 2 bits. The first pixel occupies the most significant two bits (bit7, bit6) of the first byte, and so on...

(0,0) (159,0)

(0,15) (159,15)

The pixels order is: (0,0), (0,1), ... , (0,15), (1,0), (1,1), ... , (159,0), (159,1), ... , (159,15)

The graphics data is:

```
unsigned char Dots[640] = {
    0x80,0x00,0x00,0x00,0x20,0x00,0x00,0x00,0x08,0x00,0x00,0x00,0x02,0x00,0x00,0x00,
    0x00,0x80,0x00,0x00,0x00,0x20,0x00,0x00,0x00,0x08,0x00,0x00,0x00,0x02,0x00,0x00,
    0x00,0x00,0x80,0x00,0x00,0x00,0x20,0x00,0x00,0x00,0x08,0x00,0x00,0x00,0x02,0x00,
    0x00,0x00,0x00,0x80,0x00,0x00,0x00,0x20,0x00,0x00,0x00,0x08,0x00,0x00,0x00,0x02,
    0x00,0x00,0x00,0x08,0x00,0x00,0x00,0x20,0x00,0x00,0x00,0x40,0x00,0x00,0x01,0x00,
    0x00,0x00,0x04,0x00,0x00,0x00,0x10,0x00,0x00,0x00,0x40,0x00,0x00,0x01,0x00,0x00,
    0x00,0x04,0x00,0x00,0x00,0x10,0x00,0x00,0x00,0x40,0x00,0x00,0x01,0x00,0x00,0x00,
    0x00,0x40,0x00,0x00,0x00,0x30,0x00,0x00,0x00,0x0C,0x00,0x00,0x00,0x03,0x00,0x00,
    0x00,0x00,0xC0,0x00,0x00,0x00,0x30,0x00,0x00,0x00,0x0C,0x00,0x00,0x00,0x30,0x00,
    0x00,0x00,0xC0,0x00,0x00,0x03,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x00,0x30,0x00,
    0x00,0x00,0x0C,0x00,0x00,0x00,0x03,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x00,0x30,
    0x00,0x00,0x00,0x80,0x00,0x00,0x02,0x00,0x00,0x00,0x08,0x00,0x00,0x00,0x20,0x00,
    0x00,0x00,0x80,0x00,0x00,0x02,0x00,0x00,0x00,0x08,0x00,0x00,0x00,0x20,0x00,0x00,
    0x00,0x80,0x00,0x00,0x00,0x20,0x00,0x00,0x00,0x08,0x00,0x00,0x00,0x02,0x00,0x00,
    0x00,0x08,0x00,0x00,0x00,0x20,0x00,0x00,0x00,0x80,0x00,0x00,0x02,0x00,0x00,0x00,
    0x04,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x00,0x40,0x00,0x00,0x00,0x10,0x00,0x00,
    0x00,0x04,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x00,0x40,0x00,0x00,0x00,0x10,0x00,
```

```
0x00,0x00,0x04,0x00,0x00,0x00,0x10,0x00,0x00,0x00,0x40,0x00,0x00,0x01,0x00,0x00,
0x00,0x00,0x40,0x00,0x00,0x00,0x10,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x01,0x00,
0x00,0x00,0x04,0x00,0x00,0x00,0x10,0x00,0x00,0x00,0x40,0x00,0x00,0x01,0x00,0x00,
0x00,0x04,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x00,0x40,0x00,0x00,0x00,0x10,0x00,
0x00,0x00,0x04,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x00,0x40,0x00,0x00,0x00,0x10,
0x00,0x00,0x00,0x40,0x00,0x00,0x01,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x10,0x00,
0x00,0x00,0x40,0x00,0x00,0x01,0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x10,0x00,0x00,
0x00,0x04,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x00,0x40,0x00,0x00,0x00,0x10,0x00,
0x00,0x00,0x04,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x00,0x40,0x00,0x00,0x00,0x10,
0x00,0x00,0x00,0x04,0x00,0x00,0x00,0x03,0x00,0x00,0x00,0x0C,0x00,0x00,0x00,0x30,
0x00,0x00,0x00,0xC0,0x00,0x00,0x03,0x00,0x00,0x00,0x0C,0x00,0x00,0x00,0x30,0x00,
0x00,0x00,0xC0,0x00,0x00,0x03,0x00,0x00,0x00,0x0C,0x00,0x00,0x00,0x30,0x00,0x00,
0x00,0xC0,0x00,0x00,0x00,0x30,0x00,0x00,0x00,0x0C,0x00,0x00,0x00,0x03,0x00,0x00,
0x00,0x00,0xC0,0x00,0x00,0x00,0x30,0x00,0x00,0x00,0x0C,0x00,0x00,0x00,0x03,0x00,
0x00,0x00,0x0C,0x00,0x00,0x00,0x30,0x00,0x00,0x00,0xC0,0x00,0x00,0x03,0x00,0x00,
0x00,0x0C,0x00,0x00,0x00,0x30,0x00,0x00,0x00,0xC0,0x00,0x00,0x03,0x00,0x00,0x00,
0x0C,0x00,0x00,0x00,0x03,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x00,0x30,0x00,0x00,
0x00,0x0C,0x00,0x00,0x00,0x03,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x00,0x30,0x00,
0x00,0x00,0x0C,0x00,0x00,0x00,0x03,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x02,0x00,
0x00,0x00,0x08,0x00,0x00,0x00,0x20,0x00,0x00,0x00,0x80,0x00,0x00,0x02,0x00,0x00,
0x00,0x08,0x00,0x00,0x00,0x20,0x00,0x00,0x00,0x80,0x00,0x00,0x02,0x00,0x00,0x00,
0x00,0x80,0x00,0x00,0x00,0x20,0x00,0x00,0x00,0x80,0x00,0x00,0x02,0x00,0x00,0x00,
0x08,0x00,0x00,0x00,0x20,0x00,0x00,0x00,0x80,0x00,0x00,0x00,0x20,0x00,0x00,0x00
};
```

Communication Models

1. Put the command and parameter into the buffer
2. Send out the buffer content when the buffer is full
3. Repeat this procedure until all commands and parameters have been sent out or put in buffer
4. Send out the buffer content if buffer is not empty
5. Start the receiving procedure when a read command has been sent

Transmitting Text Files

1. Send command 68H('h') "Clear Texts" when needed
2. Use command 61H('a') "Transmit Text File" to send the first text file. Send out the left buffer content immediately after doing this
3. Repeat sending other text files
4. Use command 7AH('z') "End Communication Conversation" to finish transmitting text files

Reading Text Files

1. Send out command 6DH('m') "Require Text File" with parameter "File Name"
2. The required text file will be returned in command 61H('a') "Transmit Text File"
3. Repeat requiring other text files
4. Use command 7AH('z') "End Communication Conversation" to finish reading text files

Example: Transmitting Large File

Addr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
0000	00	41	00	00	00	00	00	00	00	37	00	00	07	13	13	07	.A.....7.....
0010	02	EF	50	41	52	54	20	41	20	50	41	52	54	20	42	20	..PART.A.PART.B.
0020	50	41	52	54	20	43	20	50	41	52	54	20	44	20	50	41	PART.C.PART.D.PA
0030	52	54	20	45	EF	F1	F4	00	00	00	00	00	00	00	00	00	RT.E...........
0040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

0000H-0036H: text file "A"

We supposed that text file "A" is too large to transmit in one frame. So we packet text file "A" into five frames like this:

Addr	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ASCII
0050	24	57	52	4C	3E	21	00	41	01	00	00	00	00	00	00	61	$WRL>!.A.....a
0060	00	41	00	00	00	00	00	00	00	37	00	00	07	13	13	07	.A.....7.....
0070	02	EF	50	41	52	54	20	41	27	FC	00	00	00	00	00	00	..PART.A'.....
0080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0090	24	57	52	4C	3E	0F	00	41	01	01	00	00	00	00	00	20	$WPL>..A.....
00A0	50	41	52	54	20	42	03	FE	00	00	00	00	00	00	00	00	PART.B.....
00B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00C0	24	57	52	4C	3E	0F	00	41	01	02	00	00	00	00	00	20	$WRL>..A.....
00D0	50	41	52	54	20	43	01	FE	00	00	00	00	00	00	00	00	PART.C.....
00E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00F0	24	57	52	4C	3E	0F	00	41	01	03	00	00	00	00	00	20	$WRL>..A.....
0100	50	41	52	54	20	44	FF	FD	00	00	00	00	00	00	00	00	PART.D.....
0110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
0120	24	57	52	4C	3E	13	00	41	01	04	00	00	00	00	00	20	$WRL>..A.....
0130	50	41	52	54	20	45	EF	F1	F4	7A	AF	FA	00	00	00	00	PART.E...z.....
0140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

0050H-0079H: frame 0

0090H-00A7H: frame 1

00C0H-00D7H: frame 2

00F0H-0107H: frame 3

0120H-013BH: frame 4