C-Power1200 Multi-window control communication protocol

Version V3.3

Recension Log:

Version	Date	Changes	Executor
V1.0	2009.9.1	The first version	
V1.1	2009.12.28	Increase the protocol of select play stored program and set variable value	
V1.2	2010.2.3	Increase the introduction of network data package	
V1.3	2010.2.24	Increase the introduction of cancel select play stored program	
V1.4	2010.5.20	Add protocol for setting user variable	
V1.5	2010.6.22	Add protocol for setting global display zone	
V1.6	2010.10.9	Add protocol for stopwatch timer	
V1.7	2011.5.14	Add protocol for template box play	

1、General agreement of communication

Data packet is used to communicate between the PC and the controller, in order to enhance the reliability of data, expanding capabilities to deal with images and other data.

Communication process：

a).PC send a data packet to the controller；

b).After the controller received the data packet, analysis and processing of data packet， and then return a data packet to PC；

c). PC receives the data packets returned from the control card, and analysis the received data packets to determine whether communication is successful.

Serial setting:

Baud rate: 115200, 57600, 38400, etc. selected by the selected tool.

Format string: "115200, 8, N, 1", you can change the baud rate value 115200 according to what you set to the controller.

Packet data checksum

Communication process using the packet data checksum to check the correctness of data transmission, Checksum calculations we should pay attention：Data checksum is cumulative each byte of data, use the 16bit (2 bytes) unsigned number to represent, so when the data validation and more than 0xFFFF, the checksum, and retain only 16bit value. For example, 0xFFFA + 0x09 = 0x0003.

2、Data packet format

2.1 RS232/RS485 data packet format：

2.1.1 The data packet format of RS232/RS485 sending：

0xa5 0x68 0x32 ID 7B FF LL LH PO TP CC 。。。。。。SH SL 0xae

Data	Value	Length(Byte)	Description
Start code	0xa5	1	The start of a packet
Packet type	0x68	1	Recognition of this type of packet
Card type	0x32	1	Fixed Type Code
Card ID	0x01~0xFE 0XFF	1	Control card ID, the screen No, valid values are as follows: 1 ~ 254: the specified card ID 0XFF: that group address, unconditionally receiving data
Protocol code	0X7B	1	Recognition of this type of potocol
Additional information/ confirmation mark	FF	1	The meaning of bytes in the packet is sent, "Additional Information", is a packet plus instructions, and now only use the lowest: bit 0: whether to return a confirmation, 1 to return and 0 not to return bit1 ~ bi7: reserved, set to 0
Packed data length LL LH	0x0000~0xffff	2	Two bytes, the length of the "CC ……" part content . Lower byte in the former
Packet number PO	0x00~0x255	1	When the packet sequence number is equal to when the last packet sequence number, indicating that this is the last one package.
Last packet number TP	0x00~0x255	1	The total number of packages minus 1.
Packet data	CC ……	Variable-length	Command sub-code and data
Packet data checksum SH SL	0x0000~0xffff	2	Two bytes, checksum 。 Lower byte in the former。The sum of each byte from " Packet type " to “ Packet data” content
End code	0xae	1	The end of a packet（Package tail）

2.1.2 The data packet format of the control card returned to RS232/RS485 sender：

0xa5 0x68 0x32 ID 7B FF LL LH PO TP CC 。。。。。。 SH SL 0xae

Data	**Value**	**Length(Byte)**	**Description**
Start code	0xa5	1	The start of a packet
Packet type	**0xE8**	1	Recognition of this type of packet 0xE8 = (0x68 | 0x80), for the app 3.2 or below return 0x68, app 3.3 or above return 0xe8 to same as other protocol(such as "set time" protocol), so you can ignore the highest bit (0x80), then it works for all app version
Card type	0x32	1	Fixed Type Code
Card ID	0x01~0xFE 0XFF	1	Control card ID, the screen No, valid values are as follows: 1 ~ 254: the specified card ID 0XFF: that group address, unconditionally receiving data
Protocol code	0X7B	1	Recognition of this type of protocol.
Return value	RR	1	RR = 0x00: that successful; RR = 0x01 ~ 0xFF: that the failure error code. (0x01: checksum error) (0x02: packet sequence error) (Other: to be confirmed) In addition, a certain period of time does not receive the returned data packet, said communication failures.
Packed data length LL LH	0x0000~0xffff	2	Two bytes, the length of the "CC 。。。。。。" part content . Lower byte in the former
Packet number PO	0x00~0x255	1	When the packet sequence number is equal to when the last packet sequence number, indicating that this is the last one package.
Last packet number TP	0x00~0x255	1	The total number of packages minus 1.
Packet data	CC 。。。。。。	Variable-length	Command sub-code and data
Packet data checksum SH SL	0x0000~0xffff	2	Two bytes, checksum 。 Lower byte in the former。 The sum of each byte from " Packet type " to " Packet data" content 。
End code	0xae	1	The end of a packet（Package tail）

"Packet number", "Last packet number" in the return package was re-calculated by the number of packets returned。

2.1.3 RS232/RS485 packet data transcoding description：

The following process is done sending and receiving low-level functions, If you write your own PC side of the sending and receiving programs, you must implementation as below conventions. Use the without transcoding code to calculation checksum.

Send：
Between start code and end code, if there is 0xA5, 0xAA or 0xAE, it should be converted to two code：

0xa5 ➔ 0xaa 0x05。The purpose is to avoid the same with the start character 0xa5
0xae ➔ 0xaa 0x0e。The purpose is to avoid the same with the end of the symbol 0xae。
0xaa ➔ 0xaa 0x0a。The purpose is to avoid the same with the escape character 0xaa。

Receive：

Received symbol 0xa5, said that the beginning of a packet
Received symbol 0xae, said that the end of a packet

When PC receive data from controller, if there is 0xA5, 0xAA or 0xAE, it should convert two code to one code, specifically for

0xaa 0x05 ➔ 0xa5
0xaa 0x0e ➔ 0xae
0xaa 0x0a ➔ 0xaa

2.2 Network data packet format

2.2.1 The data packet format of network sending

Data	Value	Length(Byte)	Description
ID Code	0x00000000 ~ 0xffffffff	4	Control card ID, high byte in the former。 Need to set to the same value on the card.
Network data length	0x0000 ~ 0xffff	2	The byte length that from “Packet type” to “Packet data checksum”.
Reservation	0x0000	2	Reservations. Fill 0
Packet type	0x68	1	Recognition of this type of packet
Card type	0x32	1	Fixed Type Code
Card ID	0x01~0xFE 0XFF	1	Control card ID, the screen No, valid values are as follows: 1 ~ 254: the specified card ID 0XFF: that group address, unconditionally receiving data
Protocol code	0X7B	1	Recognition of this type of protocol.

Additional information/ confirmation mark	FF	1	The meaning of bytes in the packet is sent, "Additional Information", is a packet plus instructions, and now only use the lowest: bit 0: whether to return a confirmation, 1 to return and 0 not to return bit1 ~ bi7: reserved, set to 0
Packed data length LL LH	0x0000˜0xffff	2	Two bytes, the length of the "CC 。。。。。。" part content . Lower byte in the former
Packet number PO	0x00~0x255	1	When the packet sequence number is equal to when the last packet sequence number, indicating that this is the last one package.
Last packet number TP	0x00~0x255	1	The total number of packages minus 1.
Packet data	CC 。。。。。。	Variable-length	Command sub-code and data
Packet data checksum SH SL	0x0000~0xffff	2	Two bytes, checksum 。 Lower byte in the former。 The sum of each byte from " Packet type " to “ Packet data” content 。

The network packet data does not need to do transcoding processing.

2.2.2 The data packet format of the control card returned to network sender

Data	Value	Length(Byte)	Description
ID Code	0x00000000 ~ 0xffffffff	4	Control card ID, high byte in the former。 Need to set to the same value on the card.
Network data length	0x0000 ~ 0xffff	2	The byte length that from “Packet type” to “Packet data checksum”.
Reservation	0x0000	2	Reservations. Fill 0
Packet type	0xE8	1	Recognition of this type of packet 0xE8 = (0x68 | 0x80), for the app 3.2 or below return 0x68, app 3.3 or above return 0xe8 to same as other protocol(such as “set time” protocol), so you can ignore the highest bit (0x80), then it works for all app version
Card type	0x32	1	Fixed Type Code
Card ID	0x01~0xFE 0XFF	1	Control card ID, the screen No, valid values are as follows: 1 ~ 254: the specified card ID 0XFF: that group address, unconditionally receiving data
Protocol code	0X7B	1	Recognition of this type of protocol.
Return value	RR	1	RR = 0x00: that successful; RR = 0x01 ~ 0xFF: that the failure error code.

			(0x01: checksum error) (0x02: packet sequence error) (Other: to be confirmed) In addition, a certain period of time does not receive the returned data packet, said communication failures.
Packed data length LL LH	0x0000˜0xffff	2	Two bytes, the length of the "CC 。。。。。。" part content . Lower byte in the former
Packet number PO	0x00~0x255	1	When the packet sequence number is equal to when the last packet sequence number, indicating that this is the last one package.
Last packet number TP	0x00~0x255	1	The total number of packages minus 1.
Packet data	CC 。。。。。。	Variable-length	Command sub-code and data
Packet data checksum SH SL	0x0000~0xffff	2	Two bytes, checksum 。 Lower byte in the former。 The sum of each byte from " Packet type " to " Packet data" content 。

"Packet number", "Last packet number" in the return package was re-calculated by the number of packets returned。

The network packet data does not need to do transcoding processing.。

2.3 Command sub-code and data：CC。。。。。。

CC： A sub-byte instruction code, specifying the meaning of the data。

。。。。。。： Data content for different sub-code instructions, there are different elements。

If the data needs to be divided into several packages, command sub-code only in the first data packet appears, the other only contains the data content of data packets。

2.2.1 Command sub-code includes：

Commad sub-code(CC)	**meanings**
0x01	Division of display window (area)
0x02	To send text data to a specified window
0x03	To send image data to the specified window
0x04	Static text data sent to the specified window
0x05	To send clock data to the specified window
0x06	Exit show to return to play within the program
0x07	Save / clear the data
0x08	Select play stored program (single-byte)
0x09	Select play t stored program (double-byte)
0x0a	Set variable value
0x0b	Select play single stored program, and set the

	variable value

2.2.2 The specific definition of command sub-code and data：

Division of display window：CC=0x01:

Data Items	Value	Length(byte)	Description
CC	0x01	1	Note This command is divided into display window (area)
Window Number	0x01~0x08	1	The window should be divided into the number of valid value 1 ~ 8.
Window 1 X-coordinate XH XL	0x0000~0xffff	2	Window x-coordinate, high byte in the former
Window 1 Y-coordinate YH YL	0x0000~0xffff	2	Window y-coordinate, high byte in the former
The width of the window 1 WH WL	0x0000~0xffff	2	The width of the window, high byte in the former
The height of the window 1 HH HL	0x0000~0xffff	2	The height of the window, high byte in the former
……			
Window N X-coordinate XH XL	0x0000~0xffff	2	Window x-coordinate, high byte in the former
Window N Y-coordinate YH YL	0x0000~0xffff	2	Window y-coordinate, high byte in the former
The width of the window N WH WL	0x0000~0xffff	2	The width of the window, high byte in the former
The height of the window N HH HL	0x0000~0xffff	2	The height of the window, high byte in the former

- Based on the above definition, requires 8 bytes for each window’s location and size, then divided into N windows, data on a total of 2 +8 * N bytes

Send text data to a specified window：CC=0x02:

Data Items	Value	Length(byte)	Description

CC	0x02	1	Description This is a text data packet
Window No	0x00~0x07	1	The window sequence number, valid values 0 ~ 7.
Mode	1	1	0x00: Draw 0x01: Open from left 0x02: Open from right 0x03: Open from center(Horizontal) 0x04: Open from center(Vertical) 0x05: Shutter 0x06: Move to left 0x07: Move to right 0x08: Move up 0x09: Move down 0x0A: Scroll up 0x0B: Scroll to left 0x0C: Scroll to right
Alignment	0～2	1	0: Left-aligned 1: Horizontal center 2: Right-aligned
Speed	1～100	1	The smaller the value, the faster
Stay time	0x0000~0xffff	2	High byte in the former. Units of seconds.
String		Variable-length	Every three bytes to represent a character. First bytes: Said the color and font size: High 4bits (1-7) said the colors (red, green, yellow blue, purple, cyan, white) , low 4bits (= 0 that 8 text; = 2 indicates that 16-point text; = 3 indicated that 24-point text; = 4 indicated that 32 point text). Second bytes: English letters, symbols, is 0. Third bytes: English letters, symbols, its ASCII code.

Send image data to a specified window：CC=0x03:

Data Items	Value	Length(byte)	Description
CC	0x03	1	Description This is a image data packet
Window No	0x00~0x07	1	The window sequence number, valid values 0 ~ 7.
Mode	0x00	1	0x00: Draw
Speed	1	1	The smaller the value, the faster。Now appears that this value is invalid
Stay time	0x0000~0xffff	2	High in the former. Units of seconds.
Image Data	0x01	1	0x01: gif image file format

Format			0x02: gif image file references. 0x03: picture package picture reference. 0x04: simple image format.
Image Display X Position	0x0000~0xffff	2	Began to show the location of X coordinate. Relative upper-left corner the window.
Image Display Y Position	0x0000~0xffff	2	Began to show the location of Y coordinate. Relative upper-left corner the window.
Image Data		Variable-length	According to "image data format" is defined to determine the meaning of the data. Image data format is 0x01: gif image file of the actual data, which contains the image width, height and other information; Image data format is 0x02: the gif image file name stored in the control card. Image data format is 0x03: The image package file name and image number that stored in the controller. The middle separated by spaces. For example, "images.rpk 1" Image data format is 0x04： Simple picture data, see the description format.

Send static text：CC=0x04:

Data Items	Value	Length(byte)	Description
CC	0x04	1	Description of the data packet is static text
Window NO	0x00~0x07	1	Window sequence number, valid values 0 to 7
Data type	1	1	0x01: Simple text data
The level of alignment	0~2	1	0: left Alignment 1: center Alignment 2: right Alignment
Display area X	0x0000~0xffff	2	The X coordinate of upper left corner of the display area。Upper left corner of the window relative
Display area Y	0x0000~0xffff	2	The Y coordinate of upper left corner of the display area。Upper left corner of the window relative
Display area width	0x0000~0xffff	2	The width of display area。High byte in the former.
Display area height	0x0000~0xffff	2	The height of display area。High byte in the former.
Font		1	Bit0~3: font size
Text color R	0~255	1	The red color component
Text color G	0~255	1	The green color component
Text color B	0~255	1	The blue color component
Text		Variable-length	Text string to the end of 0x00.

Send clock：CC=0x05:

Data Items	Value	Length(byte)	Description
CC	0x05	1	Description of the data packet is clock
Window NO	0x00~0x07	1	Window sequence number, valid values 0 to 7
Stay time		2	Stay time in second。High byte in the former
Calendar		1	0: Gregorian calendar date and time 1: Lunar date and time 2: Chinese lunar solar terms 3: Lunar time and date + Solar Terms
Format		1	Format: Format bit 0: when the system (0: 12 hour; 1: 24 hours system) bit 1: Year digit (0: 4; 1: 2) bit 2: Branch (0: single; 1: multi-line) bit 3 ~ 7: reserved (set to 0)
Content		1	By bit to determine the content to display. bit 7: Pointer bit 6: weeks bit 5: seconds bit 4: minute bit 3: hour bit 2: day bit 1: month bit 0: year
Font		1	Bit0~3: font size
Text color R	0~255	1	The red color component
Text color G	0~255	1	The green color component
Text color B	0~255	1	The blue color component
Text		Variable-length	Text string to the end of 0x00.

Exit show and return to play within the program：CC=0x06:

Data Items	Value	Length(byte)	Description
CC	0x06	1	Play programs stored on the card

Save/clear the data：CC=0x07:

Data Items	Value	Length(byte)	Description
CC	0x07	1	The data packet is a request control card to save

			data in the window
Save/clear	0x00/0x01	1	0x00: save data to flash. 0x01: Clear flash data
Reserve	0x00 0x00	2	Reserved for later expansion

Select play stored program (single-byte)：CC=0x08

Data Items	Value	Length(byte)	Description
CC	0x08	1	Description of the data packet is stored program data select play(single-byte)
Options		1	Bit0: Whether to save select play message to flash.　0 not to save，1 save。 Bit1~7: Reserved, set to 0
The number of programs	1~255 or 0	1	The program number that to be selected to play, if the number is 0,the controller will exit the select play state.
The program number table	1~255	Variable-length	The program number list,1 byte for each program. Exceed the number of programs stored program number is ignored

Select play stored program (double-byte)：CC=0x09

Data Items	Value	Length(byte)	Description
CC	0x09	1	Description of the data packet is stored program data select play(double-byte)
Options		1	Bit0: Whether to save select play message to flash.　0 not to save，1 save。 Bit1~7: Reserved, set to 0
The number of programs	1~512 or 0	2	The program number that to be selected to play, the max value is 512, high byte in the former. if the number is 0,the controller will exit the select play state.
The program number table	1~65535	Variable-length	The program number list,2 bytes for each program. Exceed the number of programs stored program number is ignored

Set variable value：CC=0x0a

Data Items	Value	Length(byte)	Description
CC	0x0a	1	Description of the data packet is the data set variable value
Options		1	Bit0: Whether to save all variable value to flash, 0 not to save，1 save。 Bit1: Whether to clear all variable value before save, 0 not to clear，1 clear。

			Bit1~7: Reserved, set to 0
Variable number and allow cross-variable zone	1~100	1	Bit0~6：The variable number Bit7：Whether to allow cross-variable zone setting. 0 is not permitted; 1 is permit Corresponds to a variable number of each variable area size of each variable region is 32 bytes. Multiple continuous variables can be linked to a variable area used, occupied area of the variable number of variables can not be used。 When does not allow cross-variable area, more than 32 bytes of data are discarded；When allow cross-variable area, calculate the length of the data area to use the number of variables.
Variable data length table	n（0～255）	Variable-length	Specified by the order of bytes of data for each variable. The length of variable number and data is (1 + n) bytes.
Variable number and data		Variable-length	The first byte is a variable number, followed by a specified length of variable data

Note：

Valid values for the variable number is 1~100。Number of variables corresponding to each variable area can store 32 bytes of data, a number of continuous variable area can be used together for a variable, the variable area occupied number of variables can not be used。

When variable values are not updated and just save the variable value to the FLASH, it can set the " Variable number **" of the value of 0, set the "** Options **" to save**

Select play single stored program and set the variable value：

CC=0x0b

Data Items	**Value**	**Length(byte)**	**Description**
CC	0x0b	1	Description of the data packet is the data select single program and set variable value
Options		1	Bit0: Whether to save program number to flash 0 not to save，1save。 Bit1: Whether to save all variable value to flash , 0 not to save，1save。 Bit2: Whether to clear all variable value before save, 0 not to clear，1 clear。 Bit3~7: Reserved, set to 0
Program	1~65535	Variable-length	The program number list,2 bytes for each

numbers			program. Exceed the number of programs stored program number is ignored
Variable number and allow cross-variable zone	1~100	1	Bit0~6：The variable number Bit7： Whether to allow cross-variable zone setting. 0 is not permitted; 1 is permit Corresponds to a variable number of each variable area size of each variable region is 32 bytes. Multiple continuous variables can be linked to a variable area used, occupied area of the variable number of variables can not be used。 When does not allow cross-variable area, more than 32 bytes of data are discarded； When allow cross-variable area, calculate the length of the data area to use the number of variables.
Variable data length table	n（0～255）	Variable-length	Specified by the order of bytes of data for each variable. The length of variable number and data is (1 + n) bytes.
Variable number and data		Variable-length	The first byte is a variable number, followed by a specified length of variable data

Note：

Valid values for the variable number is 1~100。Number of variables corresponding to each variable area can store 32 bytes of data, a number of continuous variable area can be used together for a variable, the variable area occupied number of variables can not be used。

When variable values are not updated and just save the variable value to the FLASH, it can set the " Variable number **" of the value of 0, set the "** Options **" to save。**

Set global display zone：CC=0x0c

Data Items	**Value**	**Length(byte)**	**Description**
CC	0x0c	1	Describe the packet is the data which to set the global display zone.
Options		1	Bit0: Whether to save the setup to FLASH 0 not to save，1 save。 Bit1~7: Reserved, set value to 0
ZoneArea count	1~8	1	The count of global display zone which is to be set. Cancel all the zone when zone count is 0
Synchronization		1	Bit0: Synchronous display. 0 not synchronous, 1 synchronous. Bit1~7: Reserved
Retention		2	Set value 0
Zone Definition		Zone Count*16	The specific definition of global display zone

Zone Definition： （16 bytes each item）

The first byte is zone type, available zone types:

Vale	Type
1	Display the variable’s specify text
2	Display the variable’s specify file (.gif)
6	Display hint text of other zone
7	Display stopwatch timer value
other	Reserved

Type = 1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x00	Type	reserved	x		y		cx		cy		start	end	stay		font	align

Explanation：

Data Name	Data Size (byte)	Description
Type	1	1：Display the variable’s specify text
x	2	Zone start point X. High byte previous
y	2	Zone start point Y. High byte previous
cx	2	Zone width. High byte previous
cy	2	Zone Height. High byte previous
Start	1	Start variable number, valid value 1～100
End	1	End variable number, valid value 1～100
stay	2	The stay time when display each valid variable’s content, the unit is second. High byte previous
font	1	Font size and color Bit0~2: font size(8,12,16,24,32,40,48,56) Bit3: Color invert Bit4: Red value of color Bit5: Green value of color Bit6: Blue value of color Bit7: Reserved
align	1	Text alignment

All “reserved” values need to be set to 0

Type = 2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x00	Type	reserved	x		y		cx		cy		start	end	stay		mode	reserved

Explanation：

Data Name	Data Size (byte)	Description
Type	1	2: Display the variable’s specify file (.gif)
x	2	Zone start point X. High byte previous
y	2	Zone start point Y. High byte previous
cx	2	Zone width. High byte previous

cy	2	Zone Height. High byte previous
Start	1	Start variable number, valid value 1～100
End	1	End variable number, valid value 1～100
stay	2	The stay time when display each valid variable’s content, the unit is second. High byte previous
mode	1	Image draw mode: 0: left top
Reserved	1	

All “reserved” values need to be set to 0

Type = 7

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x00	Type	reserved	x		y		cx		cy		font	format	reserved			

Explanation：

Data Name	Data Size (byte)	Description
Type	1	7: Display stopwatch timer value
x	2	Zone start point X. High byte previous
y	2	Zone start point Y. High byte previous
cx	2	Zone width. High byte previous
cy	2	Zone Height. High byte previous
font	1	Font size and color Bit0~2: font size(8,12,16,24,32,40,48,56) Bit3: Color invert Bit4: Red value of color Bit5: Green value of color Bit6: Blue value of color Bit7: Reserved
format	1	0: “mm:ss” 1: “mm:ss:nn”
Reserved	4	

All “reserved” values need to be set to 0

Push and set the variable value：CC=0x0d

Data Item	Value	Lenght(byte)	Description
CC	0x0d	1	Describe the package is the data which is push and set the variable value
Options		1	Bit0: Whether to save all the variable value to FLASH. 0 not save， 1 save. Bit1: direction。 0 push back,1 push forward Bit 2~3: retention， set to 0 Bit4~7: push count. +1 means the variable

			count of push
Variable area count		1	Bit0~6: The variable count for push 1~100 Bit7: Retention, set 0
Variable data length		1	Specify variable data's byte. The total length of variable number and data is (n+1) byte
Variable number and data		changeable	The first byte is variable number, follow by specify length variable data.

Set Timer：CC=0x0e

Data Item	Value	Lenght(byte)	Description
CC	0x0e	1	Describe the package is the data of set stopwatch
Select Timer		1	Select the Timer by bite. Bit value 1 means the Timer valid Bit0: Timer 1. Bit1: Timer 2. Bit3: Timer 3. Bit4: Timer 4. Bit5: Timer 5. Bit6: Timer 6. Bit7: Timer 7.
Action		1	1：Initialize Timer 2：Reset Timer 3：Startup Timer 4：Pause Timer 5：Save the setup of Timer Other：Retention
property		1	Have different value according to the Action. Check the detail information in the below table.
Value		4	Have different value according to the Action. Check the detail information in the below table.

The description of all Actions and the correspondence property and value

Action	Description	Property	Value
Initialize Timer		Bit0: 0 Time，1 Countdown Bit1: 0 Pause , 1 start immediately Bit2~3: retention Bit4~7: Time count	High byte previous. The initialization value of countdown, measure time by millisecond. The value retention when time, set to 0
Reset Timer		Bit0: 0 Use old value，1 Use new value Bit1: 0 Pause , 1 start	High byte previous. Countdown: Use as a new initialization value

		immediately Bit2~3: Retention	when the property is set to use new value. Ignore when the property is set to use the old value. Time: Retention, set 0.
Start Timer		Reserved, set 0	Retention, set 0
Pause Timer		Retention, set 0	Retention, set 0
Save the setup of Timer		Retention, set 0	Retention, set 0