ABLEPick Pick-to-light

Communication Protocol

User Manual

V 2.4

Updated at: 2012/04/25

by

ATOP Technologies, Inc.

Tel: +886-3-5508137

Fax: +886-3-5508131

E-mail: Tabelin@atop.com.tw

CONTENTS

1. Introduction

ABLEPick is an advanced paperless picking system providing an innovative, streamlined and cost-effective Pick-to-light solution to simplify the order fulfillment process in warehouse or distribution center. ABLEPick use a state-of-the-art and light-directed technology to maximize the picking productivity, speed and accuracy in different picking operation.

ABLEPick provides considerate tools for pick-to-light applications. Data from tag sites can be handled by on-site standalone controller, then be packaged in TCP/IP frame and sent to remote host computer. The tags network is feasible for diverse field data collection.

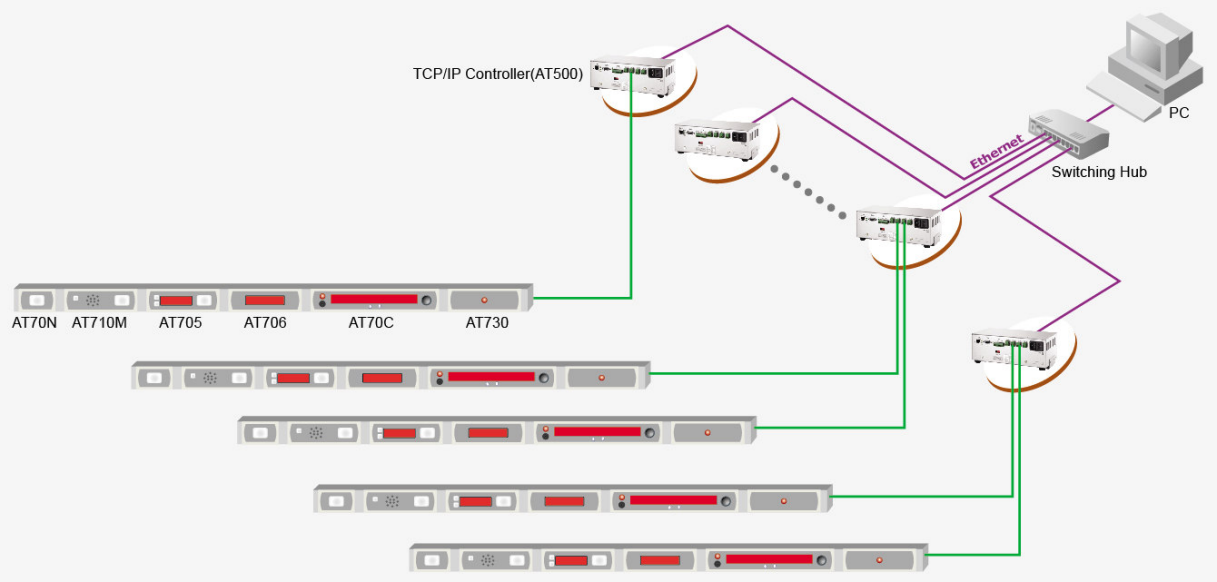
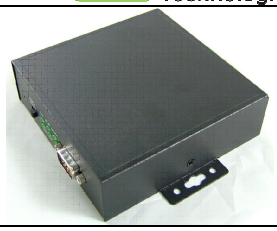


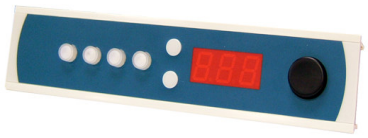
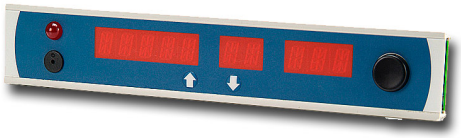
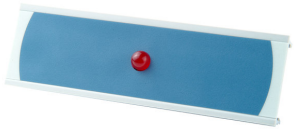
Figure ABLEPICK Architecture

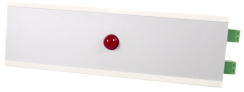
The basic ABLEPick system includes a personal computer, a controller, pick tags, bay indicators, order displayers and completion indicators. It is designed as compact, reliable, versatile, easily maintained and installed module to satisfy diversified applications. The picking system computer acquires the picking data from warehouse management system via on-line or off-line medium, and sends to picking tags. Pickers pick up the items by quantity shown on the tags and return picking status back to the system.

ATOP provides a series of ABLEPICK devices to meet your application requirement.

Model no.	Function Description	Outlook
AT500	TCP/IP controller A Maximum of 120 light module can be connected 10/100 Mbps Ethernet TCP/IP protocol	
AT500N	Two-wired magnetic bus translator A Maximum of 120 light module.	

AT400	Compact TCP/IP controller A Maximum of 30 light module can be connected 10/100 Mbps Ethernet TCP/IP protocol	
AT505	5-digit Picking Tag 1 confirm button + 2 function keys 3 colored LED light	
AT502	2-digit Picking Tag 1 confirm button + 2 function keys 3 colored LED light	
AT50N	Economic picking tag 1 confirm button + 2 function keys 3 colored LED light	
AT506	Order Display 6-digit display	
AT50C	12-digit Alphanumerical display 1 confirm button + 2 function keys. Red LED light + Buzzer	
AT510	Completion Indicator Confirm button Green LED light + buzzer	
AT510M	Melody Completion Indicator Confirm button Green LED light + speaker 12 melody songs 16 volume degree	
AT511	Bay indicator Yellow LED light	
AT504A	4-digit directional picking tags 1 confirm button + 2 function keys 3 colored LED light 2 directional arrows, up arrow is red, the down arrow is green	
AT503A	3-digit directional picking tags 1 confirm button + 2 function keys 3 colored LED light 2 directional arrows, up arrow is red, the	

	down arrow is green	
AT503-4K	3-digit picking tags 1 confirm button + 2 function keys 4 lightable buttons: Red, Green, Yellow Blue	
AT502V	2-digit vertical & directional picking tags 1 confirm button + 2 function keys 3 colored LED light 2 directional arrows, right arrow is red, the left arrow is green	
AT506-3W-123	6-digit, 3 separated windows picking tag 1 confirm button + 2 function keys 3 colored LED light 3 separated windows(1+2+3)	
AT506-2W-33	6-digit, 2 separated windows picking tag 1 confirm button + 2 function keys 3 colored LED light 2 separated windows (3+3)	
AT50A-3W-523	10-digit alphanumerical display, 3 separated windows (5+2+3) 1 confirmation button + 2 Function Keys LED Indicator + Buzzer	
AT530	RS232/485 converter 9-pin female RS232 connector	
AT520	RS232 & digital I/O field interface 4 Digital Input 4 Digital Output with relay One 9-pin female RS232 connector	
AT705	5-digit Picking Tag Large illuminated button with 6 colored LED light. 2 Function Keys	

AT703	3-digit Picking Tag Large illuminated button with 6 colored LED light 2 Function Keys	
AT702	2-digit Picking Tag Large illuminated button with 6 colored LED light 2 Function Keys	
AT70N	Large illuminated button with 6 colored LED light	
AT706	Order Display 6-digit display	
AT70C	12-digit Alphanumerical display 1 confirm button + 2 function keys. Red LED light + Buzzer	
AT710M	Melody Completion Indicator Confirm button Green LED light + speaker 12 melody songs 16 volume degree	
AT730	RS232/485 converter 9-pin female RS232 connector	
AT706-24-3K	6-digit Picking Tag with 2 color LED First 2 digits by Green, 4 digits by Red Large illuminated button with 6 colored LED light 2 up/down Function Keys 1 blue function key	
AT540	Address installation tool Portable with 12V battery.	

2. PICK-TO-LIGHT OPERATION

2.1 Work flow of pick-to-light system: Below simply describe the basic operation

2.1.1 Data entry

Picking list files can be downloaded to the LAN server or PC from WMS/MIS host. These files will be read and merged into pick-to-light picking list database.

2.1.2 Control and communication

Pick-to-light software will monitor picking flow and offer real-time information on the screen.

2.1.3 Light up

Different models of picking tags light up to indicate what order, which location and how many pieces to be picked.

2.1.4 Pick and confirm

The picker picks the quantity as tag shows or modifies the quantity directly from tag, then push the button to confirm this action.

2.1.5 Complete

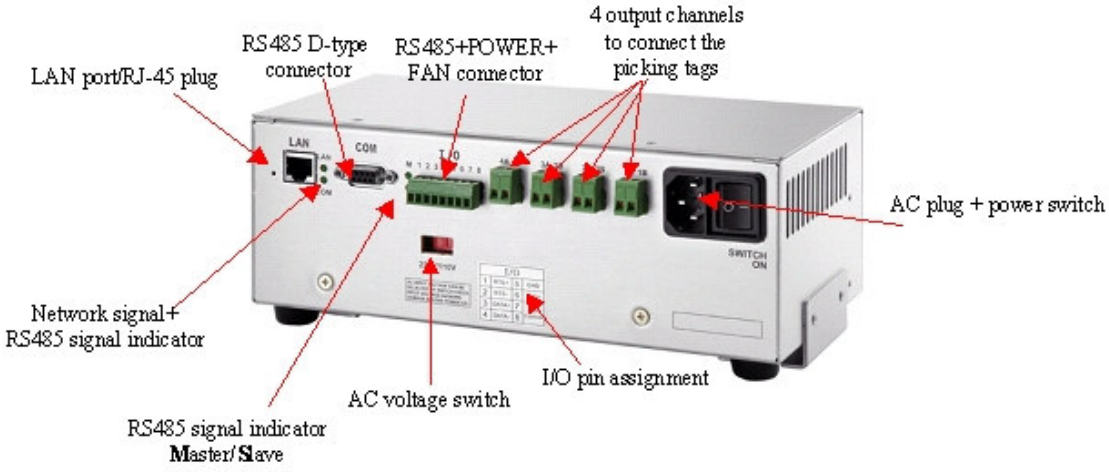
Completion indicator will light up and buzz after all jobs in a zone are done. Push the button to confirm this action and move this order to the next available zone.

3. ABLEPick Hardware Introduction

3.1 TCP/IP controller

3.1.1 AT500-Standard TCP/IP controller.

TCP/IP controller is a data transmission medium between picking control PC and all the picking devices. Which is Ethernet architecture product, following up the standard TCP/IP communication protocol to communicate with the host PC. AT500 has only PORT #1 (MESSAGE_TYPE = 60H) with 4 output channels to connect to the picking devices, each channel can connect to maximum 30 devices. So one TCP/IP controller can connect to maximum 120 pcs of picking devices.



3.1.1.1 AT500's IP configuration

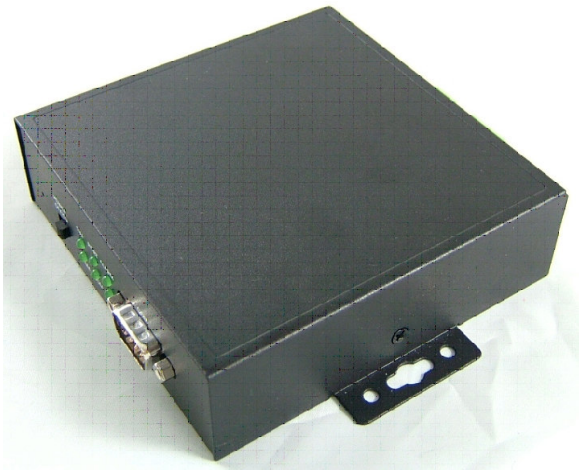
AT500's default IP address is "10.0.50.100" and sub-mask is "255.255.0.0". You can use ATOP's tool "MONITOR.exe" to know and re-configure each AT500's IP address.

3.1.1.2 Connection to TCP/IP controller (AT500)

Since AT500 has no DHCP function, so its host control PC/NB need to assign one IP address which have to be within the same domain as the AT500. Then you can create one TCP connection to connect it. AT500's TCP port is 4660.

3.1.2 AT400-Compact TCP/IP controller.

AT400 is another kind of TCP/IP controller of ABLEPick Pick-to-light system. AT400 has two serial ports design, Port #1 (MESSAGE_TYPE = 60H) is the one output channel to connect to the picking devices. which with compact capacity can be up to maximum 30 pcs of picking devices. The Port 2 (MESSAGE_TYPE = 61H) is a RS232 serial port, which can be used to connect to device with RS232 interface, such as power scanner.



AT400 can be powered by two ways, one is using the standard power adapter with specific DC12V/5A/ 60W, by plug in DC jack. The other way is powered by external power supply or battery with DC 12V via the 3-pin terminal block

AT400's default IP address is "10.0.50.100" and sub-mask is "255.255.0.0". You can use ATOP's tool "MONITOR.exe" to know and re-configure each AT400's IP address.

3.1.2.1 Abnormal voltage protection function

Basically, AT400 has the similar functionality as the standard TCP/IP controller(AT500), but it has been designed in an additional function to implement the abnormal voltage protection , either in the hardware or by return message via firmware function to inform application software. That is a powerful function on the power control especially when using an external power supply, like battery…etc.

Hardware

DC voltage	Description
Voltage high	>14.8V ; Voltage over, AT400 will shut off <14.8V ;If the voltage go down, theAT400 will re-power on.

Voltage low	<10V : if provided voltage is lower than 10V, AT400 will shut off If voltage goes back more than 10V, AT400 will re-power on.

Software

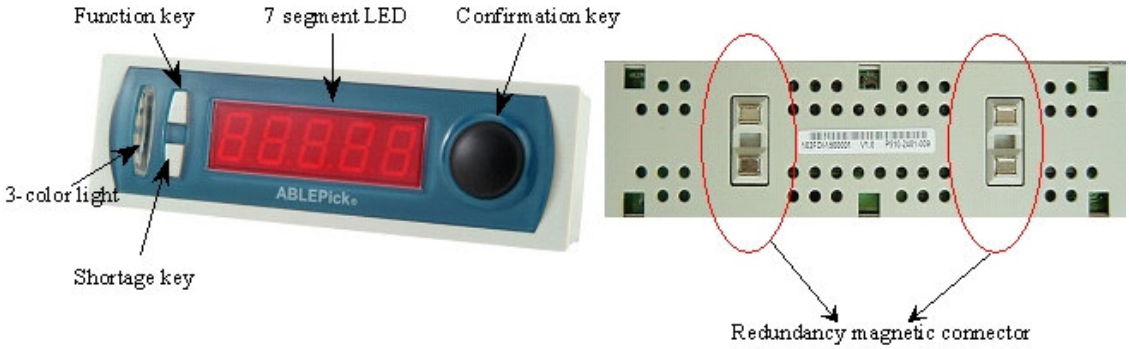
DC voltage	Description
Voltage high	When voltage is more than 14.3V, AT400 firmware will send message to inform this “voltage high” status, then ALARM LED indictor also will be blinking..
Voltage low.	When voltage is down lower 11.3V, AT400 firmware will send message to inform this “voltage low” status, then ALARM LED indictor also will be blinking.

3.2 Picking Tag

3.2.1 5-digit picking tag (AT505 & AT705)

AT505 has 5-digit 7-segment LED display, 3 buttons and 3-color LED light.

AT705 has 5-digit 7-segment LED display, one large illuminated button with 6-color LED light.



AT705

3.2.1.1 LED light.

Among ABLEPick all picking devices, AT5XX series have 3-color LED light design, AT7XX series has 6-color LED light design. The light’s color could be configured via the software control. AT5XX’s 3 colors are RED, GREEN, and AMBER individually, AT7XX’s 6 colors are RED, GREEN, AMBER, BLUE, PURPLE and INDIGO individually. The default one is RED. The configured light color can’t be stored in the EEPROM for AT5XX series, so the color will be back to default when re-power on. But it can be stored in EEPROM for AT7XX series..

3.2.1.2 3 Buttons.

Most of the ABLEPick picking devices has 3 buttons design, and their functions are the same except the AT50C.& AT70C

Right side : Confirmation button, which is to push to confirm picking successfully.

Left side : Two small buttons, up-count(function key) and down-count (shortage) buttons. Push shortage (down-count) button can acknowledge the shortage situation or by adjusting up-count/down-count button to present partial picking process. Both different behaviors could be configured by software.

There are two modes in picking operation: ***Pick mode*** and ***Stock mode.***

(a) ***Pick mode*** is for picking time. Picker can only push confirmation button to confirm picking successfully or push short button to indicate out of stock.

(b) ***Stock mode*** is for stock counting. Picker can push up-count/down-count buttons to adjust the stock quantity shown on the tag and then push confirmation button to confirm this stock count.

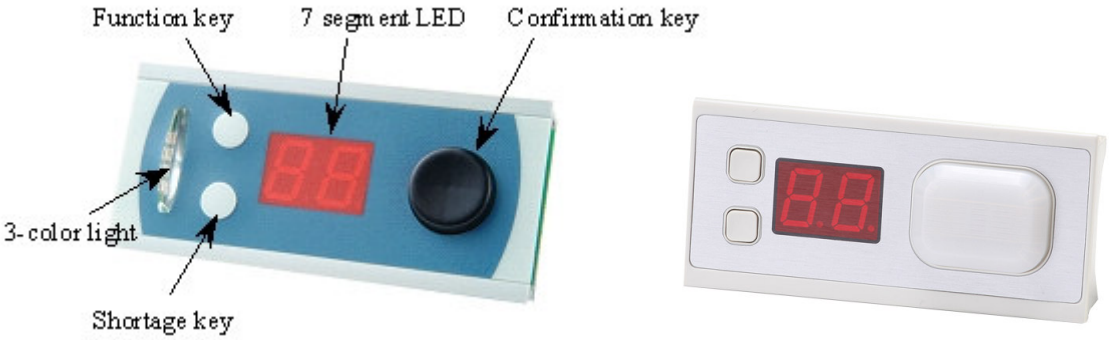
3.2.2 3-digit picking tag (AT703)

AT702 has 3-digit 7-segment LED display, one large illuminated button with 6-color LED

3.2.3 2-digit picking tag (AT502 & AT702)

AT502 has 2-digit 7-segment LED display, 3 buttons and 3-color LED lights.

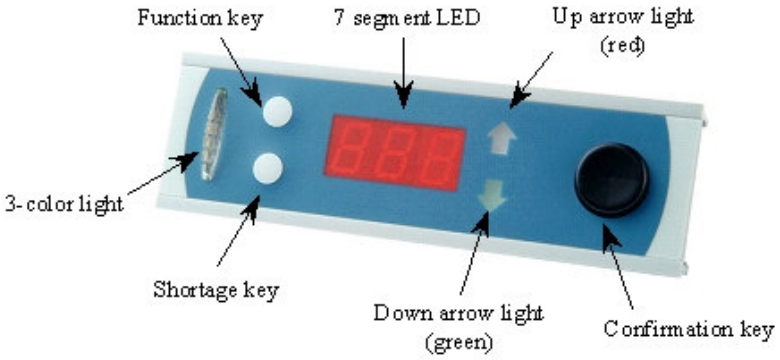
AT702 has 5-digit 7-segment LED display, one large illuminated button with 6-color LED



3.2.4 3-digit directional picking tag(AT503A)

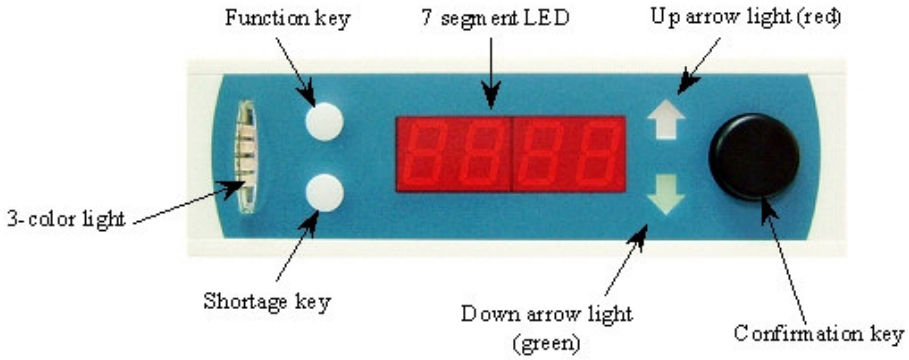
Besides 3-digit 7-segment LED display, 3 buttons and 3-color LED lights, AT503A has two arrows. One Up arrow with RED light and one Down arrow with GREEN light. Most of the features and behaviors are similar to AT505, only one difference is AT503A has two message buffers.(AT505 has only one). In other words, you can send two messages to

AT530 at the same time. The first message will be shown on the LED display, but second one will be queued in the buffer. Once the first message has been confirmed, the second one will pop out to show automatically. That is why it has two arrows to distinguish.

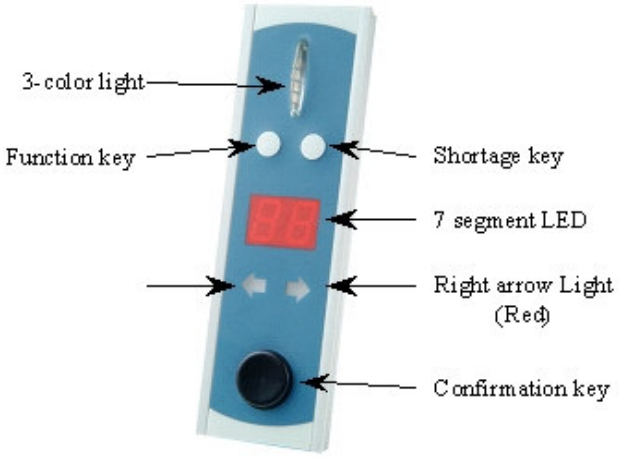


3.2.5 4-digit directional picking tag(AT504A)

AT504A is almost the same as AT503A, except the numbers of digits.



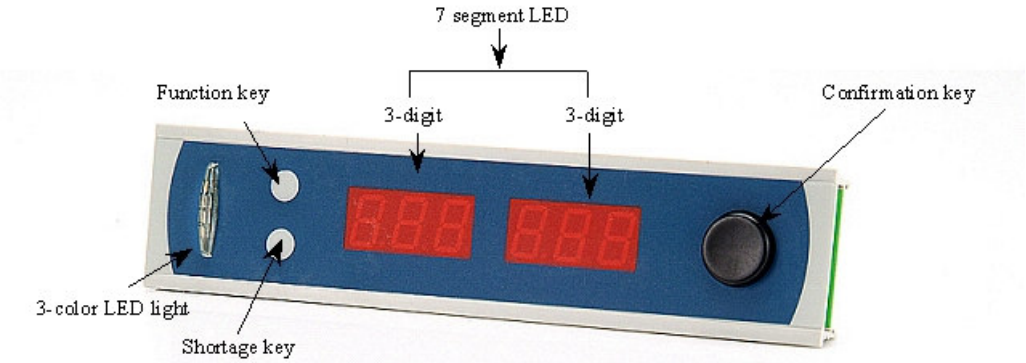
3.2.6 2-digit vertical directional picking tag(AT502V)



AT502V is a vertical design picking tag, which is similar to AT503A. Besides 2-digit 7-segment LED display, 3 buttons and 3-color LED lights, AT502V also has two arrows. One Right arrow with RED light and one Left arrow with GREEN light. Moreover, it also has two message buffers same as AT503A.

3.2.7 6-digit, 2 separated windows picking tag(AT506-2W-33)

Besides 3 buttons and 3-color LED lights, AT506-2W-33 has 6 digits 7-segment LED which is divided into two separated windows, each has 3 digits individually. Normally, AT506-2W-33 can used to show two different units of number on these windows, such as "case" on the first window and "piece" o the second window.

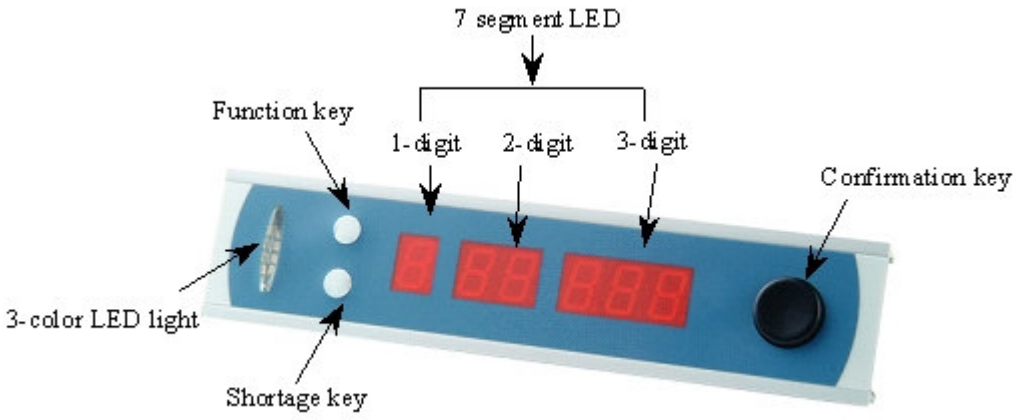


3.2.8 6-digit, 3 separated windows picking tag(AT506-3W-123)

AT506-3W-123 has the similar design as AT506-2W-33, whose 6 digits 7-segment LED display are separated into 3 separated windows, one digit, two digits and three digits individually. More than that, AT506-3W-123 also has the similar feature as AT503A, it has 10 messages buffers. It means you can send 10 messages into AT506-3W123 in advance, first message will be shown on LED, the others will be pop out to show sequentially by pushing the confirmation buttons.

Normally, one AT506-3W-123 can used to control multiple locations, especially for the slowing moving items. The 3-separated windows design are for this purpose. The first and second window can be used to indicate the specific location by rack's row and column information.

Moreover, AT506-3W-123's firmware also has special design to return message to inform software designer when its buffer is full or is empty.

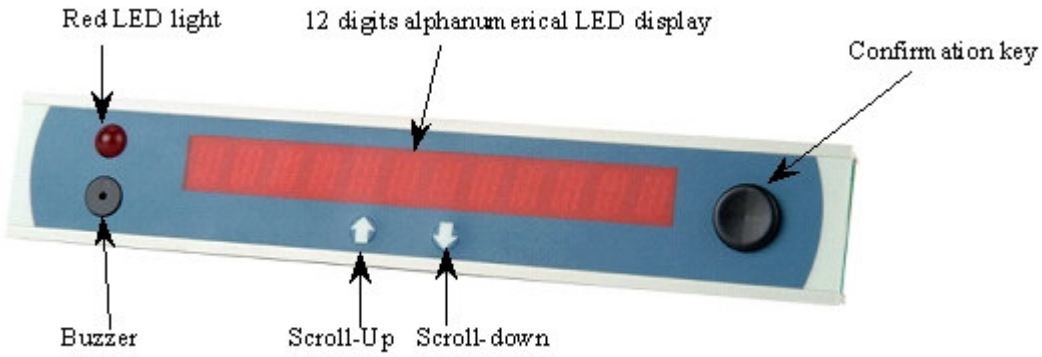


3.2.9 12-digit alphanumerical display(AT50C & AT70C)

AT50C/AT70C is an alphanumerical display with 12-digit LED, one RED LED indicator, one buzzer, one confirmation button and scroll-up & scroll down button. AT50C/AT70C could show up full alphabets and numerical characters, so normally it used to the message displayer.

AT50C/AT70C has 6 message buffers, and you can use “scroll-up” & “scroll-down” button to select and change the message.

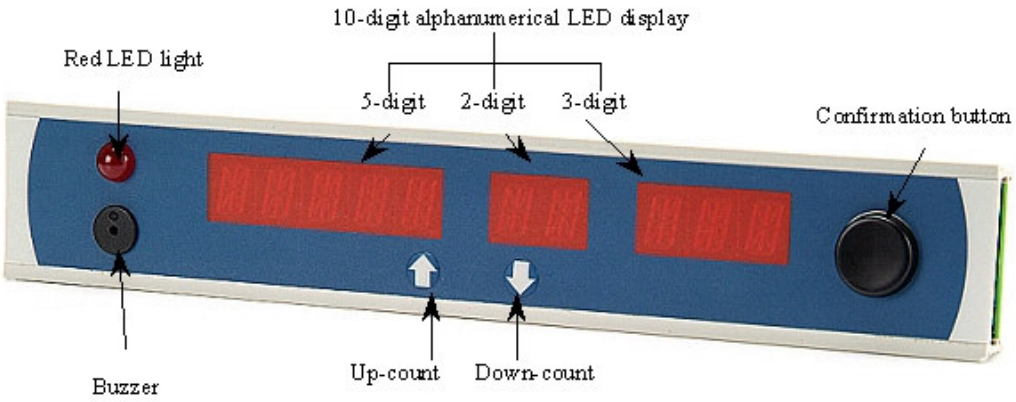
Moreover, the buzzer, LED light, confirmation button of AT50C/AT70C all can be controlled independently by software accordingly.



3.2.10 10-digit, 3 separated windows alphanumerical picking tag(AT50A-3W-523)

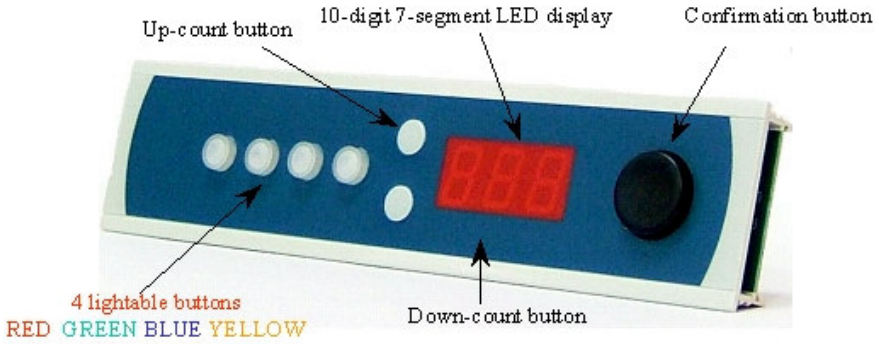
AT50A-3W-523 has the similar features and behaviors as the AT506-3W-123, which has 3 windows, 5-digit, 2-digit and 3-digit individually. One AT50A-3W-523 can control multiple locations. The first and second windows are used to display the location’s or item’s information, third window is use to show the order quantity. Since there are more digits on the first and second windows than AT506-3W-123, and the digit is alphanumerical LED, so AT50A-3W-523 can display more detail item or location’s message to instruct the operator.

AT50A-3W-523’s power on procedure, node address configuration procedure and self-testing procedure is the same as AT50C.



3.2.11 3-digit, 4 lightable buttons picking tag(AT503-4K)

AT503-4K is a new powerful device. Besides 3-digit 7-segment LED display and 3 buttons, it has 4 lightable buttons, which is RED, GREEN, BLUE and YELLOW individually.



AT503-4K has 4 buffers design, in other words, it can store maximum 4 data strings within tag in advance. And these 4 buffers will be corresponding to the 4 lightable buttons individually. Buffer 1 to **RED** light, Buffer 2 to **GREEN**, Buffer 3 to **BLUE** and Buffer 4 to **YELLOW**,

-

Different functionalities for AT503-4K :

AT503-4K has 4 special modes, detail description as below :

(1) Function 1 : Push on demand

(a) It needs to assign the specific buffer when sending data to AT503-4K. We will define a byte in communication protocol to be this specific buffer.

(b) The 3-digit LED display will still keep to be off when sending data to device, just will light on its specific lightable button.

(c) When pressing one of the solid lighting button, its specific buffer's data will show on the 3-digit LED display, meantime, the lighting button will become blinking.

(d) At this moment, you could use the up/down button to adjust the quantity on display.

(e) Press the confirmation button to return this confirmed message. Meantime, both of the blinking button and the quantity display will be off at the same time. And its specific

buffer will be eliminated. Then you can repeat to press the other lighting button to do the picking process.

(f) However, before pressing the confirmation button, it allows to press the other lighting button to change to the other color's process instead of the original one. At this moment, the original blinking one will become solid lighting, and the pressed button will become blinking, and the LED display will change to show up the new one's data.

(g) If the buffer has data existed already, and sending data the same one buffer again. Then the original existed data will be replaced by the new one.

(h) Redisplay : the lightable button which has been confirmed off has the redisplay function. You can push the off button, no released, then it can show up its specific button's original value.

(i) When pushing the lightable button and LED display will show up its specific quantity, then you can push the button again, the lightable button will rollback to last status, and the LED display will be off.

(j) According to the above description, the active light is blinking and the standby light is solid on. However, actually the light status for these two situation could be configured, such as it can be reverse.

(2) Function 2 : Auto retrieve data from buffer sequentially.

(a) It needs to assign the specific buffer when sending data. We will define a byte in communication protocol to be this specific buffer.

(b) The 3-digit LED display will show up the first assign buffer's value, and its specific lightable button will be also on. However, the second, or third…sending data, only will light on its specific lightable button.

(c) At this moment, you could use the up/down button to adjust the quantity on display.

(d) Press the confirmation button to return this confirmed message. Meantime, both of the blinking button and the quantity display will be off at the same time. And its specific buffer will be eliminated. Then the next assigned buffer's value will be show up on LED display and lightable button by its auto-retrieve function from buffer sequentially..

(e) However, before pressing the confirmation button, it also allows to press the other lighting button to change to the other color's process instead of the original one. At this moment, the original blinking one will become solid lighting, and the pressed button will become blinking, and the LED display will change to show up the new one's data.

(f) If the buffer has data existed already, and sending data the same one buffer again. Then the original existed data will be replaced by the new one.

(g) Redisplay : the lightable button which has been confirmed off has the redisplay function. You can push the off button, no released, then it can show up its specific button's original value.

(h) According to the above description, the active light is blinking and the standby light is solid on. However, actually the light status for these two situation could be configured, such as it can be reverse.

(3) Function 3: Control the 4 lightable buttons individually.

(a) Use the original command (like AT505) to send data to AT503-4K, and only the 3-digit LED display will show up the value. At this moment, the AT503-4K only has one buffer.

(b) You can send command to control the 4 lightable buttons individually, just like function keys. And you can configure the light to be on by different status, such as solid on, slow blinking, fast blinking….etc.

(4) Function 4: Auto-demo

(a) By one special command, you can send data to AT503-4K by assigning its specific buffer, then AT503-4K can repeatedly display each buffer's data sequentially.

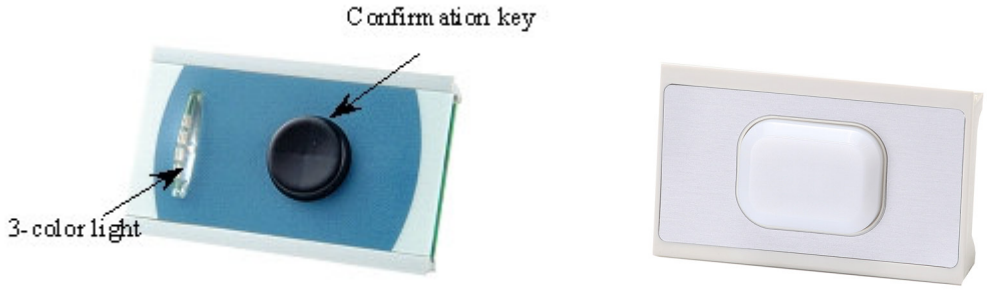
(b) You also can define the time interval for the display switching time from buffer to the next buffer.

3.2.12 Economic picking tag(AT50N & AT70N)

AT50N only has one confirmation button and one 3-color LED light.

AT70N has only one illuminated button with 6-color LED light.

Normally, it can be applied to the picking process with fixed picking quantity for each order and each item, such as the kitting system in the assembly line.

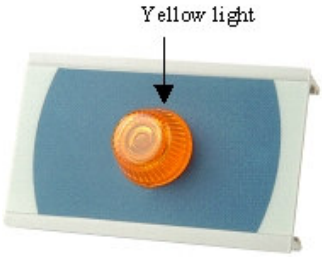


3.2.13 6-digit order display(AT506 & AT706)

AT506/AT706 only has 6-digit 7-segment LED display, which is usually used to be the message display, such as to show the order number or customer code during the picking process.

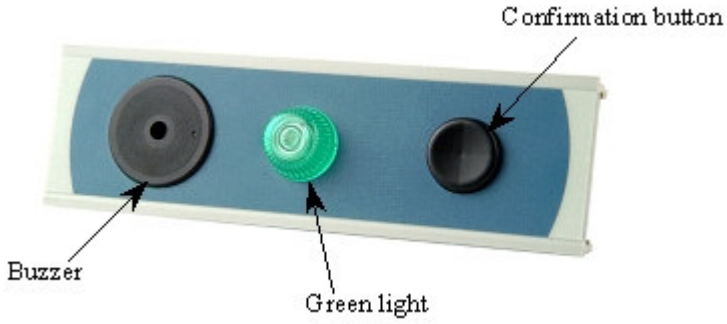
3.2.14 Bay indicator(AT511)

AT511 only has one YELLOW color light, which is usually used to be the indicator to help operator to implement or improve the productivity during picking process.



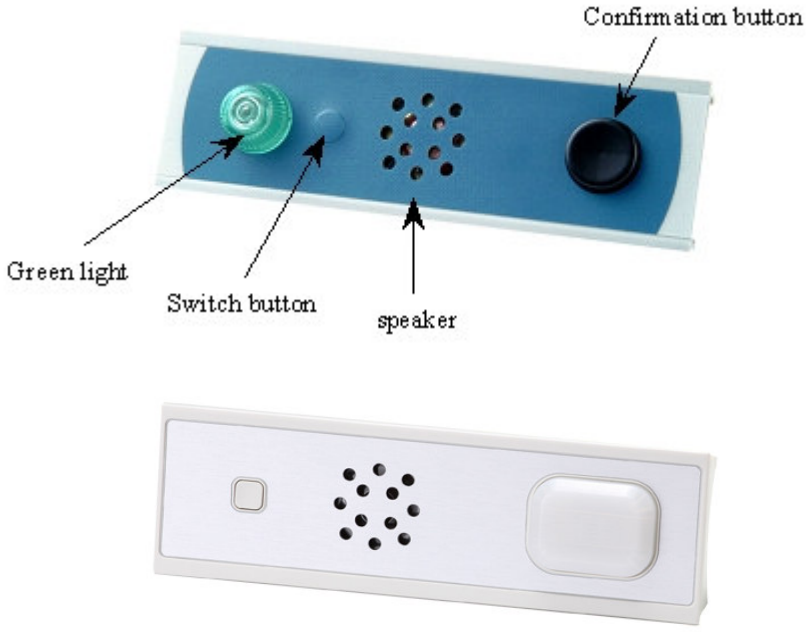
3.2.15 Completion indicator(AT510)

AT510 only has one GREEN color light, one buzzer and one confirmation button. By its “sound” and “light” features, it can use to be the indictor to attract operator’s ears or eyes attention during some picking stage.



3.2.16 Melody completion indicator(AT510M & AT710M)

AT510M/AT710M is one kind of indicator like AT510, which use melody song to be instead of buzzer. This kind of design just considers the human factor concern. There are 12 songs built in it. You can use the button on it to perform the song selection and also could use the button to change the volume of sound.



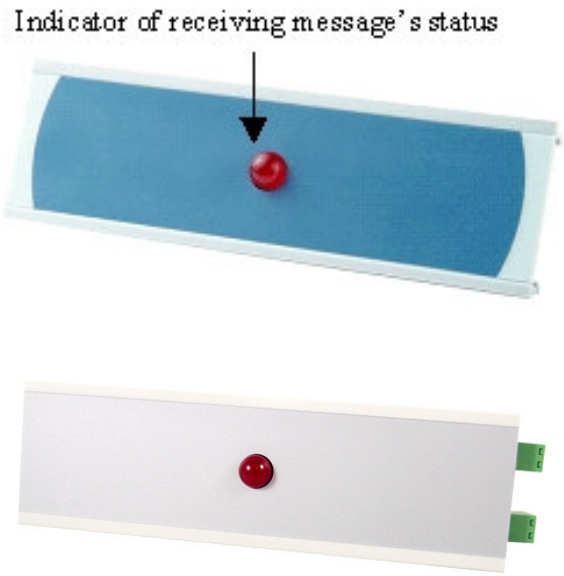
Push the left small button, not released and then push the confirmation button to change to different song. There are twelve songs, names are as below :

(1) Jingle bells
(2) Carmen
(3) Happy Chinese new year
(4) Edelweiss
(5) Going home
(6) PAPALA
(7) Classical
(8) Listen to the rhythm of the falling rain
(9) Rock and roll
(10) Happy birthday
(11) Do Re Me

(12)Strauss

3.2.17 RS232 converter(AT530/AT530L & AT730)

AT530./AT730 is used to convert message from RS232 interface into ATOP's proprietary protocol, such as Barcode scanner with RS232 interface….etc. The only one red light is used to be the indicator of receiving message status. When message is sent into the device from RS232, the led will be on and when this message is gotten back by the controller, then the led will be off.



3.2.17 6-digit, 2-color display pick tag(AT706-24-3K)

AT706-24-3K has 6-digit 7-segment LED display with 2 colors, the first 2 digits are GREEN, the last 4 digits are RED color. Besides the two up and down functions, AT706-24-3K has another blue function key, which can return message any time when it has been push down. AT706-24-3K has the fully compatible function as the other pick tags, like AT705, AT703.

AT706-24-3K has 10 messages buffers. It means software can send 10 messages into AT706-24-3K in advance, first message will be shown on LED, the others will be pop out to show sequentially by pushing the confirmation buttons.

ATOP also has another similar model called AT706-3K, whose 6 digits 7-segment LED are all RED color, and have one display buffer.

4. ABLEPick TCP/IP communication Protocol

4.1 Communication Control Block (CCB) format

Communication data format (CCB: Communication Control Block) between PC and TCP/IP controller is as below:

CCB LEN (L) =	Block length (8 Bytes + DATA Length) 60H : port 1 , (61H : port 2)
CCB LEN (H) =	
MESSAGE TYPE = 60H	
Reserved	
Reserved	
Reserved	
SUB-COMMAND	Description as below
SUB-NODE = {01H ~ C8H}	Specify tag address (up to 200), 0FC:broadcasting
DATA	Command message

PS : For AT500 ,TCP/IP controller, only port 1 is available. So Message Type will be always = 60H.

4.2 Sub-command List

Sub-command	*Data Direction* ***PC***√TCP/IP controller	*Description*
00H	→	Showing alphanumeric on tag (*note* ←)
01H	→	Turning off all alphanumeric on tag/clear all display in buffer
02H	→	LED indicator ON
03H	→	LED indicator OFF
04H	→	Buzzer ON
05H	→	Buzzer OFF
06H	←	Returning confirmation message from push-button
07H	←	Return shortage message from push-button
08H	→	Setting maximal field device address
09H	→	Reading status of the connected field devices
09H	←	Returning status of the connected field devices
0AH	←	Field device timeout
0BH	→	Query device malfunction
0BH	←	Returning device malfunction
0CH	←	Device being unable to execute the command
0DH	←	Returning Buttons locked message (only active after sending message from the host to device)
FCH	←	Device resetting OK
0FH	←	Returning quantity under stock mode
10H	→	Flashing alphanumeric on tag
11H	→	Flashing LED (Lamp) indicator
12H	→	Setting flash timer
13H	→	Showing tag address
14H	→	Resetting device
15H	→	Disabling shortage push-button
16H	→	Enabling shortage push-button
17H	→	Emulating confirmation push-button

18H	→	Emulating shortage push-button
19H	→	Switching to stock mode
1AH	→	Switching to picking mode
1BH	→	Disabling confirmation push-button
1CH	→	Enabling confirmation and shortage buttons
1EH	→	Setting valid digits for counting
1FH	→	Setting tag's Multiple configuration
3AH	→	Setting the device node address
64H	←	Return from the device
FCH	→	Get device's detail information
FCH	←	Return device's detail information (same as reset OK)
FAH	→	Get device's F/W model information
FAH	←	Return device's F/W model information
30H	→	Display data with the illuminant buttons for AT503-4K
30H	←	Return message while sending by 30H for AT503-4K
31H	→	Controlling of the 4 illuminant buttons solely for AT5034K
31H	←	Return message of 4 illuminant buttons from AT503-4K
32H	→	AT503-4K auto-demonstrating function
24H	←	When enable cycle edit bit in sub-command=1F, Data[0] = 0A
FBH	→	Advanced way to configure device's node address
F7H	→	AT400 alarm status control
F7H	←	AT400 alarm status feedback

Note:

(1) → means sending message from Host to device via TCP/IP controllers.

(2) ← means return message from Device to Host via TCP/IP controllers

(3) Symbols to be able to display on device's 7-segment LED display refer to below **"Display code for Tags"**

Display code for Tags

ASCII code	7 segment display
30H	0
31H	1
32H	2
33H	3
34H	4
35H	5
36H	6
37H	7
38H	8
39H	9
41H	A
62H	b
43H	C
64H	d
45H	E
46H	F
63H	c
48H	H
4CH	L
20H	(space)

ASCII code	7 segment display
6EH	n
6FH	o
68H	h
72H	r
75H	u
69H	i
79H	y
74H	t
71H	q
47H	G
53H	S
6CH	l
4FH	O
5BH	[
5DH	]
50H	P
55H	U
2DH	-

4.3 Send Command to Tags (PC→TCP/IP controller)

All the different kinds of behaviors on each light module could be controlled from host

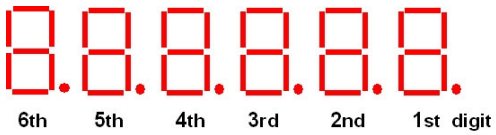
PC/NB to send specified sub-command via TCP/IP controller to the devices, such as turn show up message on 7-segment LED display, turn on/off LED light, turn on/off buzzer, change light color…etc.

4.3.1 Showing alphanumerical characters on 7-segment LED display

SUB-COMMAND = **00H**

CCB	Description
CCB LEN(L)= 0FH	Length of CCB = 15 bytes
CCB LEN(H) =00H	
MESSAGE TYPE = 60H	60H: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	00H
SUB-NODE	0x01~0xFA (address) or 0xFC (Broadcasting)
Data 0	6th digit in LED
Data 1	5th digit in LED
Data 2	4th digit in LED
Data 3	3rd digit in LED
Data 4	2nd digit in LED
Data 5	1st digit in LED
Data 6	Dot position

DATA[0~5] = The alphanumerical characters will display on the 7-segment LED display



DATA[0] = 6th digit
DATA[1] = 5th digit
:
DATA[5] = 1st digit
DATA[6] = decimal points

Each decimal points are mapping to each bit of DATA[6].
i.e. DATA[6] = 01H → Showing the 1st decimal point
02H → Showing the 2nd decimal point
03H → Showing the 1st & 2nd decimal point ats the same time
04H → Showing the 3rd decimal point
07H → Showing the 1st, 2nd & 3rd decimal points at the same time
……
3FH → Showing all of the decimal points

4.3.2 Turn off 7-segment LED dispisplay or clear all message buffers.

SUB-COMMAND = **01H**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	

Reserved	
Reserved	
SUB-COMMAND = **01H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

Sub-command = 01H also can be used to eliminate the display buffers, such as for AT503A, AT502V, AT50C/AT70C,AT506-3W-123, AT506-2W-33A, AT50A-3W-523…etc. And if send to the other models, like AT505/AT705, AT703, AT502/AT702, AT506-2W-33…etc, then the message for redisplay is also reset, in other words, redisplay won't work after 01H..

4.3.3 Turn LED indicator ON

SUB-COMMAND = **02H**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **02H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

Turn on all the LED light, for AT50C/AT70C, there is a little difference, please refer 4.5.1.2.

4.3.4 Turn LED indicator OFF

SUB-COMMAND = **03H**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **03H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

4.3.5 Turn Buzzer ON

SUB-COMMAND = **04H**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **04H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

Turn on the buzzer of AT510 and Melody of AT510M/AT710M, not for AT50C/AT70C's

buzzer.

4.3.6 Turn Buzzer OFF

SUB-COMMAND = **05H**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **05H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

Turn off the buzzer of AT510 and Melody of AT510M/AT710M, not for AT50C/AT70C's buzzer.

4.3.7 Flashing the display on 7-segment LED display

SUB-COMMAND = **10H**

This sub-command can force the 7-segment LED display to be blinking. And its flashing time interval is according to the flash time interval setup of sub-command = 12H.

CCB LEN(L)= 0FH	Length of CCB = 15 bytes
CCB LEN(H) =00H	
MESSAGE TYPE = 60H	60H: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND = 10H	
SUB-NODE	0x01~0xFA (address) or 0xFC (Broadcasting)
Data 0	6th digit in LED
Data 1	5th digit in LED
Data 2	4th digit in LED
Data 3	3rd digit in LED
Data 4	2nd digit in LED
Data 5	1st digit in LED
Data 6	Dot position

DATA[0~5] = The alphanumerical characters will display on the 7-segment LED display

DATA[0] = 6th digit
DATA[1] = 5th digit
:
DATA[5] = 1st digit
DATA[6] = decimal points

4.3.8 Flashing the LED indicator

SUB-COMMAND = **11H**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	

MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **11 H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

This sub-command can force the LED indicator to be blinking. And its flashing time interval is according to the flash time interval setup of sub-command = 12H.

4.3.9 Configure the flashing time interval

SUB-COMMAND = **12H**

CCB LEN (L) = 0AH	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **12 H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
DATA[0] = 00H	Reserved
DATA[1] = Time interval value	00H ~F0H

There are 4 blinking levels
DATA[1] = 00H ~ 01H : the blinking time interval by 0.25 sec
02H ~ 03H : the blinking time interval by 0.5 sec
04H ~ 07H : the blinking time interval by 1 sec
08H ~ F0H : the blinking time interval by 0.25 sec

4.3.10 Showing tag's node address on display

SUB-COMMAND = **13H**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **13 H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

4.3.11 Disable the shortage/down-count button

SUB-COMMAND = **15H**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	

SUB-COMMAND = **15 H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

4.3.12 Enable the shortage/down-count button

SUB-COMMAND = **16H**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **16 H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

4.3.13 Emulating push confirmation button

SUB-COMMAND = **17H**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **17 H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

4.3.14 Emulating push shortage/down-count button

SUB-COMMAND = **18H**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **18 H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

4.3.15 Switching to stock mode

SUB-COMMAND = **19H**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	

SUB-COMMAND = **19 H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

Note: The default mode is "pick mode", tag switches pick mode when it receive the changing mode command (SUB-COMMAND = 1AH). Additionally, it will return back to "pick mode" when the tag is reset or power off. However, AT503A, AT502V, AT506-3W-123, AT50C/AT70C, AT50A-3W-523 do not have stock mode

4.3.16 Switching to picking mode

SUB-COMMAND = **1AH**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **1A H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

Note: The default mode is "picking mode", tag switches to stock mode when it receive the changing mode command (SUB-COMMAND = 19H) and it will return back to picking mode when the tag is reset or power off.

4.3.17 Disable the confirmation button

SUB-COMMAND = **1BH**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **1B H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

4.3.18 Enable confirmation button

SUB-COMMAND = **1CH**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **1C H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

4.3.19 Set available number of digits for up/down counting

Specify the available numbers of digits from the right side on LED display could be

adjusted by pressing up-count or down-count button to allow user to change quantity

SUB-COMMAND = **1EH**

CCB LEN (L) = 09H	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **1E H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
DATA[0] = high byte value of timer	

DATA[0] = Setting the number of digit is valid to adjust quantity: (Range: 01H ~ 06H)
01H: Only the 1st digit from right side on LED display is valid for adjusting
02H: Only 1st, 2nd digits from right side on LED display is valid for adjusting
:
06H: 1st to 6th digits are valid for adjusting

4.3.20 Set Tag's specific configuration

SUB-COMMAND = **1FH**

4.3.20.1 Configure the LED light's color

CCB LEN (L) = 0AH	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **1F H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
DATA[0] = 00H	
DATA[1] = LED light's color	

DATA[1] = LED Color
LED Color = 00H, **Red**
LED Color = 01H, **Green**
LED Color = 02H, **Orange**
LED Color = 03H, LED light Off

For new AT7XX series, big illuminant button(AT705,AT703,AT702, AT70N)

CCB LEN (L) = 0AH	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **1F H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
DATA[0] = 00H	

DATA[1] = LED light's color

DATA[1] = LED Color

LED Color = 00H,	**Red**
LED Color = 01H,	**Green**
LED Color = 02H,	**Orange**
LED Color = 03H,	**Blue**
LED Color = 04H,	**Pink**
LED Color = 05H,	**Cyan**

DATA[2] = 55H ------→ Store the color configuration into EEPROM

Warning : EEPROM has the life cycle for writing. Thus we don't recommend to save every new configuration into EEPROM each time.

4.3.20.2 Set valid digits for up/down counting

CCB LEN (L) = 0AH	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **1F H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
DATA[0] = 01H	
DATA[1] = Valid digits configuration	

DATA[1] = Setting Value of Valid Digits: The value of binary type represents to the digit of tag. (Range: 00H ~ 1FH)

00H = none digit valid for counting.
01H = 1st digit valid for counting, others not.
02H = 2nd digit valid for counting, others not.
03H = 1st,2nd digits valid for counting. others not.
:
0FH = 1st,2nd,3rd,4th digits valid for counting.
1FH = 1st to 5th digits valid for counting.

4.3.20.3 Disable/Enable tag mode configuration (Type 1)

CCB LEN (L) = 0BH	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **1F H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
DATA[0] = 02H	**permanent, save to EEPROM**
DATA[1] = setting value	
DATA[2] = 55H	

DATA[1] = Setting Value of Function: The value of binary type represents to the individual function Enable/Disable.

Bit = 1: Enable, Bit = 0: Disable

Bit0 = Confirmation push-button
Bit1 = Shortage push-button
Bit2 = Allowed to change the quantity by push-button
Bit3 = Key code return while push tag's button
Bit4 = Time-out showing on 7-segments.
Tag has not gotten the polling message for 10 seconds
Bit5 = Allowed to change the Node Address by push-button
Bit6 = Redisplay enable/disable
Showing the last quantity number on 7-segment again when the down-count button keep to be pressed, no released.
Bit7 = no used

Note : Key code return : Normally, when there is data shown on the 7-segment LED display, then push the tag's button, and there is one message back. If there is nothing shown on it, even pushing the button, there is still no response. But our tags can be re-configured to enable the key code return function (see bit 3 above). When this function is enabled, even there is nothing shown on LED display, and pushing any kind of button. Then there will have one message sent back with sub-command = 64h.

The default tag configuration mode value is 73H, or 115 in decimal which can be saw on the power on procedure, it means :

Bit0: Confirmation Button,	**Enable**/Disable (1/0).
Bit1: Shortage Button,	**Enable**/Disable (1/0)
Bit2: up-count / down-count	Enable/**Disable** (1/0).
Bit3: Key code return while press button	Enable/**Disable**(1/0).
Bit4: Communication time-out "FF" display	**Enable**/disable (1/0).
Bit5: Scroll-up Key	**Enable**/Disable(1/0)
Bit6: Redisplay	**Enable**/Disable(1/0)

Above means the down-count will behavior to be as the shortage button.

If The tag mode value reconfigure to 75H, or 117 in decimal, it means :

Bit0: Confirmation Button,	**Enable**/Disable (1/0).
Bit1: Shortage Button,	Enable/**Disable** (1/0).
Bit2: upcount / downcount	**Enable**/Disable (1/0).
Bit3: Key code return while press button	Enable/**Disable**(1/0).
Bit4: Communication time-out "FF" display	**Enable**/disable (1/0).
Bit5: Scroll-up Key	**Enable**/Disable(1/0)
Bit6: Redisplay	**Enable**/Disable(1/0)

Then, the two small buttons will behavior to be as the down-count and up-count button for adjusting the quantity, and the tag could present partial picking behavior.

Warning : EEPROM has the life cycle for writing. Thus we don't recommend to save every new configuration into EEPROM each time.

4.3.20.3 Disable/Enable tag mode configuration (Type 2)

CCB LEN (L) = 0AH	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	

Reserved	
Reserved	
SUB-COMMAND = **1F H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
DATA[0] = 03H	**Temperately, without save to EEPROM**)
DATA[1] = setting value	

DATA[1] = Setting Value of Function: details as above.4.3.20.2

4.3.20.4 Set LED blinking time interval

CCB LEN (L) = 0AH	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **1F H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
DATA[0] = 04H	
DATA[1] = Blinking Period Value	

DATA[1] = Blinking Period Value (00H ~ 05H)

00H	LED Off
01H	LED On
02H	2 sec blinking
03H	1 sec blinking
04H	0.5 sec blinking
05H	0.25 sec blinking

4.3.20.5 Set digits blinking time interval (digit by digit)

CCB LEN (L) = 0CH	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **1F H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
DATA[0] = 05H	
DATA[1] = time interval	
DATA[2] = time interval	
DATA[3] = time interval	

DATA[1~3] = Blinking Period

Blinking Period: each 4 bits represent one digit blinking period value.

DATA[1] bit4 ~ bit7 = 6^{th} digit
DATA[1] bit0 ~ bit3 = 5^{th} digit
DATA[2] bit4 ~ bit7 = 4^{th} digit
DATA[2] bit0 ~ bit3 = 3^{rd} digit
DATA[3] bit4 ~ bit7 = 2^{nd} digit
DATA[3] bit0 ~ bit3 = 1^{st} digit

00	LED Off
01	LED On
02	2 sec blinking
03	1 sec blinking
04	0.5 sec blinking
05	0.25 sec blinking

For AT7XX series (V1.1),

CCB LEN (L) = 0CH	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **1F H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
DATA[0] = 05H	
DATA[1] = 00H	
DATA[2] = time interval	
DATA[3] = time interval	
DATA[4] = time interval	

DATA[2~4] = Blinking Period

Blinking Period: each 4 bits represent one digit blinking period value.

DATA[2] bit4 ~ bit7 = 6th digit
DATA[2] bit0 ~ bit3 = 5th digit
DATA[3] bit4 ~ bit7 = 4th digit
DATA[3] bit0 ~ bit3 = 3rd digit
DATA[4] bit4 ~ bit7 = 2nd digit
DATA[4] bit0 ~ bit3 = 1st digit

00	LED Off
01	LED On
02	2 sec blinking
03	1 sec blinking
04	0.5 sec blinking
05	0.25 sec blinking

For example :
DATA[2] = 52H : means 6th digit blink by 0.25 sec interval, 5th digit by 2 sec.
DATA[3] = 34H : means 4th digit blink by 1 sec interval, 3rd digit by 0.5 sec.
DATA[4] = 10H : means 2nd digit blink by solid on, 1st digit is off

4.3.20.6 Set Digits brightness

CCB LEN (L) = 0AH	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **1F H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

DATA[0] = 06H
DATA[1] = time interval

DATA[1] = *Digit Brightness*

Digit Brightness: Setting each digit brightness while it change to be dark with blinking.

Bit = 1: brightness 50 %, bit = 0: light off

Bit0: represents digit0
Bit1: represents digit1
Bit2: represents digit2
Bit3: represents digit3
Bit4: represents digit4
Bit5: represents digit5
Bit6: no used
Bit7: no used

4.3.20.7 Configure Special function

For configuring some special function

CCB LEN (L) = 0AH	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **1F H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
DATA[0] = 0AH	
DATA[1] = Configuration	

DATA[1] =

Bit0: selecting Picking mode	1-stock/0-picking
Bit1: stock mode quick completion	enable/disable
Bit2: Completion mark ”-------“ display	enable/disable
Bit3: device self-testing function	enable/dispable
Bit4: Cycle edit function	enable/disable
Bit5: no used	
Bit6: no used	
Bit7: no used	

Note (1) : Bit 0, selecting pick mode.
configure the light to be stock mode or pick mode, same as sub-command 19H & 1AH.

Note (2) : Bit 1 : stock mode quick completion.
When configure device to be "stock mode", after sending number on display, user needs to follow left to right sequence, digit by digit to adjust the quantity and confirm back, even quantity no need to adjust, this procedure also needs to go through to confirm. "Stock mode quick completion" can skip this procedure to confirm quickly if there is no any change on digits under stock mode, just push the button to confirm back. Under this configuration, if user needs to adjust the quantity, just push the down-count/up-count button instead, then it will enter the original procedure to adjust each digit

Note (3) : Bit 2 : Completion mark “-----“ display

In general mode, when device be pushed the confirmation button, the display will be off, nothing left on it. Enable this bit, it will display "-----" instead of off after confirmation.

Note (4) : Bit 3 : device self-testing function
.Enable/disable device's self-testing function, described in ABLEPick user manual.3.2.1.5

Note (5) : Bit 4 : Cycle edit function
In general mode, the maximum value you can adjust by up-count/down-count button is always equal to the original sending value, no more than it. Enable this bit, user can adjust the quantity to be more than the original sending value.
This function only can be applied on AT505 V1.2 & AT705 above, when it is enabled, the LED light & display will be blinking all the way.
This bit is valid only when the tag mode's bit 2 (up-count/down-count) is enabled.
When this bit has been enabled, the sub-command of return message is 24H instead of 06H

4.3.**21 Get device's F/W model information**

Asking the device to return its F/W model number.

SUB-COMMAND = **FAH**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **FA H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

Note : The function only can be applied on AT505 V1.2 above

Some models share the same F/W, then the return F/W model name will be the same, for example, AT705, AT703, AT702, AT70N, AT706 share the same F/W version, then the return model name is always "AT705" instead.

4.3.22 Get device's detail configured information

Get device's detail information

SUB-COMMAND = **FCH**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **FC H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

4.3.23 Configure the TCP/IP controller's polling range

SUB-COMMAND = **08H**

TCP/IP always sends a polling message communicate with devices. The polling sequence is from device address 1 to device address 'N'. This command is used to set the maximal polling address 'N'. You can use this command to set the maximal tag node address to the TCP/IP controller's best polling range to save the polling time of a TCP/IP controller.

CCB LEN (L) = 0BH	Data length of this block
CCB LEN (H) = 00H	(11 Bytes)
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **08H**	
SUB-NODE ={01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
DATA[0] = any value	Reserved bytes
DATA[1] = any value	
DATA[2] = any value	

DATA[0~2]: for compatibility reason with older version format

4.3.23 Getting all the device's status under one TCP/IP controller

SUB-COMMAND = **09H**

This command could use to ask TCP/IP controller to report all its connected device's status. Once TCP/IP controller receives this sub-command, it will return a message with same sub-command = 09H to list out all the devices status.

CCB LEN (L) = 07H	Data length of this block
CCB LEN (H) = 00H	(7 Bytes)
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **09H**	

4.3.24 Asking device to reset

SUB-COMMAND = **14H**

You can use this command to force a specific device to reset. The host may receive a resetting OK message (SUB-COMMAND = FCH)

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	(8 Bytes)
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **14H**	

SUB-NODE={01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)

4.3.25 Configure the device's node address

SUB-COMMAND = **3AH**

Some field devices support software command to configure its node address.

CCB LEN (L) = 0DH	Data length of this block
CCB LEN (H) = 00H	(13 Bytes)
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **3AH**	
SUB-NODE={01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
DATA[0] = 40H	
DATA[1] = 1BH	
DATA[2] = 1BH	
DATA[3] = 10H	
DATA[2] = {01H ~ FAH}	New Node Address

Note: 1. SUB-NODE = FCH : Broadcast to all field devices
2. New Node Address: the new device address (for device), from 1 to 250 (01H ~ FAH)
3. In case you do not know the device address and you want to set a new one. You can do the following steps:
Step1: Keep only this device on the field bus. Please make sure it is the only one.
Step2: Use this command and make the old device address to FCH (broadcast), then you can set the device to a new address.

4.4 Receive message from Tags (PC←TCP/IP controller)

Note: The practical expected receiving block length refer to the first 2 bytes of block (Block Length, It's important for programming).

4.4.1 Returning confirmation message from push-button

Return the confirmation message back when tag confirmation push-button pressed or received the confirmation emulating command.

SUB-COMMAND = **06H**

CCB	Description
CCB LEN(L)= 0FH	Length of CCB in byte (14 bytes)
CCB LEN(H) = 0x00	
MESSAGE TYPE = 60H	60H: port1 , 61H : port 2
Reserved	
Reserved	
Reserved	
SUB-COMMAND = 06H	
SUB-NODE	0x01~0xFA (address) or 0xFC (Broadcasting)
Data 0	6th digit in LED
Data 1	5th digit in LED
Data 2	4th digit in LED
Data 3	3rd digit in LED

Data 4	2nd digit in LED
Data 5	1st digit in LED
Data 6	Dot position

DATA[0~6] = *Current Showing Alphanumerics*

AT500 has only PORT #1, its MESSAGE_TYPE is always 60H. AT400 has two port2, besides the port 1 for picking devices. The RS232 serial port is PORT #2, MESSAGE_TYPE will be 61H.

SUB-COMMAND = **24H**

If the cycle edit bit is enabled (please refer 4.3.20.7 Configure Special function), then the return confirmation message sub-command will be 24H instead of 06H.

4.4.2 Returning shortage message from push-button

Return the shortage message back when tag shortage push-button pressed or received the shortage emulating command.

SUB-COMMAND = **07H**

CCB	Description
CCB LEN(L)= 0FH	Length of CCB in byte (14 bytes)
CCB LEN(H) = 0x00	
MESSAGE TYPE = 60H	60H: port1 ; , 61H : port 2
Reserved	
Reserved	
Reserved	
SUB-COMMAND = 07H	
SUB-NODE	0x01~0xFA (address) or 0xFC (Broadcasting)
Data 0	6th digit in LED
Data 1	5th digit in LED
Data 2	4th digit in LED
Data 3	3rd digit in LED
Data 4	2nd digit in LED
Data 5	1st digit in LED
Data 6	Dot position

DATA[0~6] = *Current Showing Alphanumerics*

4.4.3 Returning device timeout

SUB-COMMAND = **0AH**

When the Host sends a message to the field device through a TCP/IP controller, the TCP/IP controller will wait for an acknowledge message from the field device. If the TCP/IP controller does not gets an acknowledge message in a timeout period, then the TCP/IP controller will send this message to the HOST.

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	(8 Bytes)
MESSAGE TYPE =60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **0AH**	

SUB-NODE = {01H ~ FAH}	Device address

4.4.4 Returning a device malfunction

Return a message back to host when the device is malfunction

SUB-COMMAND = **0BH**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	(8 Bytes)
MESSAGE TYPE =60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **0BH**	
SUB-NODE = {01H ~ FAH}	Device address

4.4.5 Returning an illegal command

Return a message back to host when TCP/IP controller receive an illegal sub-command sent from host.

SUB-COMMAND = **0CH**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	(8 Bytes)
MESSAGE TYPE =60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **0CH**	
SUB-NODE = {01H ~ FAH}	Device address

4.4.6 Returning push-button locked message

When the tag receive command message from host PC and the tag's push-button is on the status during the time pressed and not release yet, the locked message will be prompted

SUB-COMMAND = **0DH**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	(8 Bytes)
MESSAGE TYPE =60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **0DH**	
SUB-NODE = {01H ~ FAH}	Device address

4.4.7 Returning a device resetting OK

When a device received the Resetting request command (SUB-COMMAND = 14H) and reset successfully, it returned this message to the Host. If the Host does not get this message after sending the Resetting request command in a timeout period, then the device has failed to reset.

SUB-COMMAND = **0EH**

CCB LEN (L) = 08H	Data length of this block
CCB LEN (H) = 00H	(8 Bytes)
MESSAGE TYPE =60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **0EH**	
SUB-NODE = {01H ~ FAH}	Device address

Note : 0EH is only for ATOP's old generation pick-to-light hardware, for the AT5XX, and AT7XX, when reset OK or power on OK, the return message will be FCH instead, no more 0EH

4.4.8 Returning quantity under stock mode

When the device is under stock mode, its return message will be 0F in stead of 06H or 07H.

SUB-COMMAND = **0FH**

CCB	Description
CCB LEN(L)= 0FH	Length of CCB in byte (14 bytes)
CCB LEN(H) = 0x00	
MESSAGE TYPE = 60H	60H: port1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = 0FH	
SUB-NODE	0x01~0xFA (address) or 0xFC (Broadcasting)
Data 0	6th digit in LED
Data 1	5th digit in LED
Data 2	4th digit in LED
Data 3	3rd digit in LED
Data 4	2nd digit in LED
Data 5	1st digit in LED
Data 6	Dot position

DATA[0~6] = *Current Showing Alphanumerics*

4.4.9 Returning all devices' communication status

Please refer the 4.3.23, when host send a message to request TCP/IP controller to report all its connected device's status, then one return message with sub-command = 09H will be generated.

SUB-COMMAND = **09H**

You can use this return CCB to identify field devices and the number that are connected.

CCB LEN (L) = 2BH	Data length of this block
CCB LEN (H) = 00H	(43 Bytes)
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **09H**	
SUB-NODE	Maximum Node Address

DATA[0]
DATA[1]
:
DATA[34]

DATA[0~2]: reserved for compatibility reason with older version format
DATA[3~34]: *Field Device Status*

Field Device Status:

Bit 0 of BYTE 0 -> status of SUB-NODE 1
Bit 1 of BYTE 0 -> status of SUB-NODE 2
Bit 2 of BYTE 0 -> status of SUB-NODE 3
.
.
Bit 7 of BYTE 0 -> status of SUB-NODE 8
Bit 0 of BYTE 1 -> status of SUB-NODE 9
.
.
Bit 7 of BYTE 1 -> status of SUB-NODE 16
.
.
Bit 6 of BYTE 24 -> status of SUB-NODE 199
Bit 7 of BYTE 24 -> status of SUB-NODE 200

Note:
Bit value = 0 : normal (connected)
Bit value = 1 : abnormal (unconnected)

4.4.10 Special return code.

SUB-COMMAND = **64H**

CCB LEN (L) = 0AH	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **64 H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
DATA[0] =	
DATA[1] =	

DATA[0] = 00H
The return message when the Keycode return function is enabled.
Data(1)= 16H "confirmatin key " pressed
43H "shotage button/down-count button" pressed
25H "function button/up-count button" pressed.

Actually, the return code number is not only the above 3 values (16H, 25H and 43H), users can push the 3 buttons by different combinations to get different return code. Within each combination, the last released button is different, then the value will be different, even on the same buttons.

For example, push up-count + confirmation buttons together at the same time:
> release both up-count + confirmation together -> return code value is 04H

> release up-count first, then confirmation -> return code is 24H
> release confirmation first, then up-count -> return code 14H,

Consider the real application and reasonable design, we only define above three return codes of confirmation, up-count and down-count buttons for used implementation..

DATA[0] = 01H
Tag busy. It is the situation that the message queued in tag not be polled back, and meantime host AP still sending message to it. Then the latter sending message will be drop off and return one message with sub-command = 64H and data0 is 01H to inform.

DATA[0] = 02H
This message is only for AT503A, AT502V and AT506-3W-123, when their buffer is full and the first message has been replaced.

DATA[0] = 03H
The last message in buffer has been displayed on. Just for AT506-3W-123.

DATA[0] = 04H
The return message from the blue function key on the model of AT706-24-3K, AT706-3K or AT705-RFID.

4.4.11 Return device's F/W model string.

Receiving the device's F/W model name when sending Sub-command = FAH..

SUB-COMMAND = **FAH**

Block Length = 8 + 8 = 16Bytes
Data[0~7] : 8-byte model string

Note : The function only can be applied on AT505 V1.2 above and AT7XX series(excludes AT70C& AT730). However, some models share the same F/W, thus the return F/W model name will be the same, for example, AT705, AT703, AT702, AT70N, AT706 share the same F/W version, then the return model name is always "AT705" instead.

4.4.12 Return device's detail information

Receiving the device's detail information when sending Sub-command = FCH..

SUB-COMMAND = **FCH**

Block Length =13 Bytes

Different models will have different value on the Data bytes

AT5XX

Data[0] : Product ID code

02H	AT505 / AT502 / AT50N / AT506 / AT511 / AT510 / AT510M /AT705/AT703/AT702/AT70N/AT706/AT710M AT506-2W-33
03H	AT50C/AT70C
04H	AT530/AT730

05H	AT503A / AT502V / AT506-3W-123 / AT506-2W-24 / AT506-2W-33A / AT504-2W-22A / AT50A-3W-523
06H	AT503-4K

Data[1] : F/W version
Data[2] : tag's mode code, default = 73H
Data[3] : the configuration result of data[1], defined in sub-command = 1FH, data[0]=0AH, please refer the page32 of **4.3.20.7 Configure Special function**

Data[4] : delay time : such as 02H means "delay time" = 2 *0.4 sec = 0.8 sec

For example : Product ID code = 04H, AT530
Data[1] : F/W version
Data[2] : buffer size, default = 32H(50bytes)
Data[3] : block interval time : 0AH (10 msec)
Data[4] : 08H (default)

AT7XX

Data[0] : Product ID code

02H	AT505 / AT502 / AT50N / AT506 / AT511 / AT510 / AT510M /AT705/AT703/AT702/AT70N/AT706/AT710M AT506-2W-33
03H	AT50C/AT70C
04H	AT530/AT730
05H	AT503A / AT502V / AT506-3W-123 / AT506-2W-24 / AT506-2W-33A / AT504-2W-22A / AT50A-3W-523
06H	AT503-4K

Data[1] : F/W version
Data[2] : tag's mode code, default = 73H
Data[3] : the configuration result of data[1], defined in sub-command = 1FH, data[0]=0AH, please refer the page32 of **4.3.20.7 Configure Special function**

Data[4] : delay time : such as 02H means "delay time" = 2 *0.4 sec = 0.8 sec
Data[5] : LED color status
Data[6] :reserved (default = 02H)

For example : Product ID code = 02H, AT705
Data[1] : F/W version
Data[2] : tag mode = 73H
Data[3] : 02H (default)
Data[4] : 08H(default)
Data[5] : 00H(RED color)
Data[6] : 02H(reserved)

4.5 Specific protocol for some light models

4.5.1 Protocol for AT50C/AT70C

The CCB(communication Control Block) from Host PC to AT50C/AT70C is as below :

CCB	Description
CCB LEN(L)= 15H	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 60H	60H: port1 of controller,
Reserved	

Reserved	
Reserved	
SUB-COMMAND	
SUB-NODE	0x01~0xFA(address of AT50C/ AT70C) or 0xFC (Broadcasting)
Data 0	Special definition for AT50C/AT70C
Data 1	1st digit display for AT50C/AT70C
Data 2	2nd digit display for AT50C/AT70C
Data 3	3rd digit display for AT50C/AT70C
Data 4	4th digit display for AT50C/AT70C
Data 5	5th digit display for AT50C/AT70C
Data 6'	6th digit display for AT50C/AT70C
Data 7	7th digit display for AT50C/AT70C
……	…..
Data 12'	12th digit display for AT50C/AT70C

Data[0]: Bit assignment as below

Bit 0~2: 12 digits LED display control
Bit 3 : 0
Bit 4 : Confirmation key control (enable/disable)
Bit 5,6,7: 0

Bit							
7	6	5	4	3	2	1	0

0 0 0-------12 digits LED display OFF
0 0 1-------12 digits LED display ON
0 1 0-------12 digits LED display, 2 second blinking
0 1 1--------12 digits LED display, 1 second blinking
1 0 0--------12 digits LED display, 0.5 second blinking
1 0 1 ------- 12 digits LED display, 0.25 second blinking

0----------Disable confirmation key
1----------Enable confirmation key

Data[1]….Data[12]: message display

Totally 13 data bytes in each buffer, its display format is controlled by 1st byte Data(0). AT50C/AT70C have max. 6 buffers for application.. Data stored in stack buffer according to receiving order. Each buffer has its own display format respectively.. By scroll-up and scroll-down key, buffer data can be scrolled up and scrolled down. Pressing “confirmation key”, displayed data will be uploaded with sub-command 06H if bit4 of data(0) is enabled. In case disabled, no message sent back.

EX: To show Test-1234567 on LED display, located in address 5, port1 and indicator with 1 second blinking , contents of CCB should be :

CCB LEN(L)= 15H
CCB LEN(H) = 0x00
MESSAGE TYPE = 60H
Reserved
Reserved
Reserved
SUB-COMMAND= 0x00
SUB-NODE= 0x05
Data 0=0x13

Data 1='T'
Data 2='e'
Data 3='s'
Data 4='t'
Data 5='-'
Data 6'='1'
Data 7='2'
Data 8='3'
Data 9='4'
Data 10='5'
Data 11='6'
Data 12='7'

4.5.1.1 Turn off message display or clear buffer.

SUB-COMMAND = **01H**
Block Length = 8 Bytes

This function is the same as 4.3.2.

4.5.1.2 LED indicator ON

SUB-COMMAND = **02H**

CCB LEN (L) = 0AH	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **02H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
Data(0)	
Data(1)	LED status

LED indicator can be controlled as "ON" only or blinking with different period.
Data(0)= 00H

Data(1)= 00H	Indicator OFF
01H	Indicator ON
02H	2 second blinking
03H	1 second blinking
04H	0.5 second blinking
05H	0.25 second blinking

4.5.1.3 LED indicator OFF

SUB-COMMAND = **03H**

This function is the same as 4.3.4

4.5.1.4 Turn Buzzer ON

SUB-COMMAND = **04H**

CCB LEN (L) = 09H	Data length of this block
CCB LEN (H) = 00H	

MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **04H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
Data(0)	Buzzer status
Data(1)	Number of beep

The buzzer can be turned "ON" with different period.

Data(0)= 00H buzzer OFF
01H buzzer ON
02H ON with 2 second period
03H ON with 1 second period
04H ON with 0.5 second period
05H ON with 0.25 second period

Data(1)= 01H ~ FFH, can decide the number of beep sound y above period

4.5.1.5 Turn Buzzer OFF

SUB-COMMAND = **05H**

This function is the same as 4.3.6

4.5.1.6 Receiving from AT50C/AT70C

The message received from AT50C/AT70C or TCP/IP controller as below:

CCB	Description
CCB LEN(L)= 14H	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 60H	60H: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	
SUB-NODE	0x01~0xFA (address of AT50C/AT70C)
Data 0	1st digit display
Data 1	2nd digit display
Data 2	3rd digit display
Data 3	4th digit display
Data 4	5th digit display
Data 5'	6th digit display
Data 6	7th digit display
……	
Data 11'	12th digit display

The receiving sub-command, please refer 4.4

4.5.2 Protocol for AT504/AT503A/AT502V

The CCB(communication Control Block) from Host PC to AT503A/AT502V is as below :

CCB	Description

CCB LEN(L)= 0FH	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 60H	60H: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	
SUB-NODE	0x01~0xFA (address of AT504A, AT503A, AT502V) or 0xFC (Broadcasting)
Data 0	Arrow status
Data 1	Arrow status
Data 2	1st digit for AT504A
Data 3	2nd digit for AT504A 1st digit for AT503A
Data 4	3rd digit for AT504A 2nd digit for AT503A 1st digit for AT502V
Data 5'	4th digit for AT504A 3rd digit for AT503A 2nd digit for AT502V
Data 6	Dot position

Both arrows on AT504A, AT503A and AT502V used the 6th and 5th digit of the 7-segment LED display to control. In order to let arrows show brightly, sending "8" to 6th and 5th digit will be the good choice.

Data [0] : "8:" -> light on up arrow (AT504A and AT503A), right arrow (AT502V)
Data [1] : "8" -> light on down arrow(AT504A and AT503A), left arrow(AT502V)
Data [2] ~ Data [5] : The display message for AT504A, AT503A and AT502V.
Data [6] : dot position.

AT503A/AT502V has two message buffers to display, one related feature for the return message is to inform user the buffer is full.

Buffer full: The maximum display buffer size for AT503A and AT502V is 2 records. When the buffer is full and still send 3rd data to it, then the original first data will be dropped out automatically and meanwhile AT503A or AT502V will send one message back to inform this situation.

Sub-command = 64H , *please refer 4.4.10.*

Data(0) = 02H , Mean the AT503A or AT502V's buffer is full

4.5.3 Protocol for AT506-3W-123

The CCB(communication Control Block) from Host PC to AT506-3W-123.

CCB	Description
CCB LEN(L)= 0FH	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 60H	60H: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	
SUB-NODE	0x01~0xFA or 0xFC (Broadcasting)

Data 0	1st digit LED display
Data 1	2nd digit LED display
Data 2	3rd digit LED display
Data 3	4th digit LED display
Data 4	5th digit LED display
Data 5	6th digit LED display
Dats 6	Dot position

AT506-3W-123 has ten FIFO message buffers, the related features for the return message is to inform user the buffer is full or buffer empty, as below :

1) Buffer full: The maximum display buffer size is 10 records. When the buffer of AT506-3W-123 is full and still send 11th data to it, then the original first data will be dropped out automatically and meanwhile AT506-3W-123 will send one message back to inform this situation.

Sub-command = 64H, *please refer 4.4.10.*
Data(0) = 02H, Mean the AT506-3W-123's buffer is full

2) Buffer empty: When the stored messages in buffer keep to be confirmed back till the last one is confirmed back, at this moment, AT506-3W-123 will send one message back to inform this "buffer empty" situation.

Sub-command = 64H, *please refer 4.4.10.*
Data(0) = 03H, Mean the AT506-3W-123's buffer is empty

4.5.4 Protocol for AT530/AT730

The RS-232 communication mode of RS232 Converter is fixed to be as follows :

Baud Rate	9600 bps
Parity	None
Data Length	8 bits
Stop Bits	1 bit

4.5.4.1 Configure the data receiving block time interval.

AT530 uses this block time interval to distinguish two different consecutive received data strings.

SUB-COMMAND = **3AH**

CCB	Description
CCB LEN(L)= 0DH	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 60H	60H: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND = 3AH	
SUB-NODE	0x01~0xFA or 0xFC (Broadcasting)
Data 0 = 40H	
Data 1 = 1BH	
Data 2 = 1B H	
Data 3 = 35H	
Dats 4 = time interval	

DATA[4] = 2~255 ms (ex: 0AH means 10 msec and it is the default value)

4.5.4.2 Enable the flow control function

SUB-COMMAND = **3AH**

CCB	Description
CCB LEN(L)= 0CH	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 60H	60H: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND = 3AH	
SUB-NODE	0x01~0xFA or 0xFC (Broadcasting)
Data 0 = 40H	
Data 1 = 1BH	
Data 2 = 1BH	
Dats 3 = 36H	

4.5.4.3 Disable the flow control function

SUB-COMMAND = **3AH**

CCB	Description
CCB LEN(L)= 0CH	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 60H	60H: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND = 3AH	
SUB-NODE	0x01~0xFA or 0xFC (Broadcasting)
Data 0 = 40H	
Data 1 = 1BH	
Data 2 = 1BH	
Dats 3 = 37H	

4.5.4.4 Send message to RS-232 port

SUB-COMMAND = **28H**

Block Length = 8 Bytes + DATA[0~ N] = (8 + N + 1) Bytes (Note: N<50)

DATA[0~N] = *Message being sent to RS-232 Port*

Example: sending the “HELLO123” + ***CR***(Carriage-Return) + ***CF***(Line-Feed) message through RS232 Convertor via RS-232.
Block Length = 8 + 10 = 18 Bytes
DATA[0~7] = “HELLO123”
DATA[8] = 0DH
DATA[9] = 0AH

4.5.5 Protocol for AT510M/AT710M

You can use the same sub-command of turn buzzer on/off (refer to 4.3.5 and 4.3.6) to control the melody on/off of AT510M/AT710M.

4.5.5.1 Selection of AT510M/AT710M's songs

CCB LEN (L) = 0AH	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **1F H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
DATA[0] = 0BH	
DATA[1] = 00h~0BH	(12 songs)

4.5.5.2 Adjust the volume of AT510M/AT710M's speaker

CCB LEN (L) = 0AH	Data length of this block
CCB LEN (H) = 00H	
MESSAGE TYPE = 60H	60H : port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = **1F H**	
SUB-NODE = {01H ~ FAH}	0x01~0xFA (address) or 0xFC (Broadcasting)
DATA[0] = 0CH	
DATA[1] = 00h~0FH	The volume of sound 16 different levels, 00H(min), 0FH(max)

4.5.6 Protocol for AT503-4K

AT503-4K has 3-digit LED display and 4 lightable buttons. The basic functions are compatible with the other models. But there are some special functions AT503-4K.

4.5.6.1 Display data with the illuminant buttons

The following describes the sub-command and Data string CCB(communication Control Block) to AT503-4K.

CCB	Description
CCB LEN(L)= length of CCB	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 0x60	0x60: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	30H
SUB-NODE	0x01~0xFA (address) or 0xFC (Broadcasting)
Data 0	define buffer index and its specific light's status
Data 1	All keys' status of AT503-4K
Data 2	5th digit in LED
Data 3	4th digit in LED
Data 4	3rd digit in LED
Data 5	2nd digit in LED

Data 6	1st digit in LED
Dot	Dot position

When sending message to AT503-4K by 30H, then AT503-4K will have 4 buffers to store the data string. Meantime, each buffer will map to one specific illuminant button, as below:

1st buffer -> Red light
2nd buffer -> Green light
3rd buffer -> Blue light
4th buffer -> Yellow light

You have to assign the buffer index you would like to put, in other words, which button you would like to light on by defining the specific bit value on the data 0, as below:

1) **Data [0]:** Define the buffer index of sending data and its specific illuminant button's status. Bit assignment of Byte is as below.

Bit 0 : LED status
Bit 1 : LED status
Bit 2 : LED status
Bit 3 : Auto retrieve data from buffer and display or not
Bit 4 : Buffer index
Bit 5 : Buffer index
Bit 6 : reserved
Bit 7 : reserved

Bit assignment:

7	6	5	4	3	2	1	0	
					0	0	0--------	delete this buffer data.
					0	0	1--------	Light is ON
					0	1	0--------	Light is blinking by 2 second interval
					0	1	1--------	Light is blinking by 1 second interval
					1	0	0--------	Light is blinking by 0.5 second interval
					1	0	1--------	Light is blinking by 0.25 second interval
				0	----------			By pressing illuminant button to retrieve.
				1	----------			Auto retrieve data from buffer and display
			0	0	----------			assign to 1st button
			0	1	----------			assign to 2nd button
			1	0	----------			assign to 3rd button
			1	1	----------			assign to 4th button

PS (1): Bit 3 will configure the sending data is going to be displayed automatically on LED display (bit 3= 1) or just light on the illuminant button only (bit 3= 0) and need to press the illuminant button to show up its specific data.

PS (2): When define Bit 3 = 0, when you press the button to display its specific quantity on LED display. However, you also could press the same button again to cancel the last job to rollback to standby status.

Example :

Data 0 = 09H → sending data to button 1 by solid on light with auto display.
Data 0 = 19H → sending data to button 2 by solid on light with auto display.
Data 0 = 29H → sending data to button 3 by solid on light with auto display.
Data 0 = 39H → sending data to button 4 by solid on light with auto display.

Data 0 = 01H → sending data to button 1 by solid on light with none-auto display.
Data 0 = 11H → sending data to button 2 by solid on light with none-auto display.
Data 0 = 21H → sending data to button 3 by solid on light with none-auto display.
Data 0 = 31H → sending data to button 4 by solid on light with none-auto display.

2) **Data [1]:** Define all AT503-4K's key's status (enable/disable) when sending one message, which includes confirmation button, up-count button, down-count button and 4 illuminant buttons.

Bit 0 : Confirmation button (reserved)
Bit 1 : Up-count key (reserved)
Bit 2 : Down-count key (reserved)
Bit 3 : 4 illuminant buttons to be confirmed button or not(reserved).
Bit 4 : Define up-count whether can be increased quantity more than sending value.
Bit 5 : reserved
Bit 6 : Blinking time interval for standby light
Bit 7 : Blinking time interval for standby light

Bit assignment:

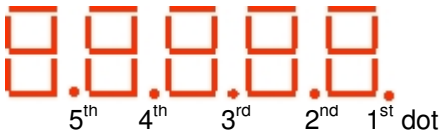
7	6	5	4	3	2	1	0

0 ----------Disable increasing quantity function on up-count button
1 ----------Enable increasing quantity function on up-count button

0 0 ----------Blinking type 1(25% ON, 75% OFF)
0 1 ----------Blinking type 2(50% ON, 50% OFF)
1 0 ----------Blinking type 3(75% ON, 25% OFF)
1 1 ----------Blinking type 4(100% ON, 0% OFF)

3) **Data [2] ~ Data [6]:** We reserve maximum 5 digits could be displayed on AT503-4K's LED display.

4) **Data 7 (Dot) :** Decimal dot position.



Bit assignment:

7	6	5	4	3	2	1	0	
0	0	0	0	1	----------	1st dot is on		
0	0	0	1	0	----------	2nd dot is on		
0	0	1	0	0	----------	3rd dot is on		
0	1	0	0	0	----------	4th dot is on		
1	0	0	0	0	----------	5th dot is on		

4.5.6.2 Return message while sending by 30H

CCB	Description
CCB LEN(L)= length of CCB	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 0x60	0x60: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	30H
SUB-NODE	0x01~0xFA (address) or 0xFC (Broadcasting)
Data 0	Data buffer index
Data 1	5th digit in LED
Data 2	4th digit in LED

Data 3	3rd digit in LED
Data 4	2nd digit in LED
Data 5	1st digit in LED
Data 6	Dot position

When sending message to AT503-4K by 30H, and push the confirmation button, then AT503-4K will return the message with sub-command = 30H with buffer index and the data string:

Data 0 = 00H -> 1st buffer -> Red light
Data 0 = 01H -> 2nd buffer -> Green light
Data 0 = 02H -> 3rd buffer -> Blue light
Data 0 = 03 H -> 4th buffer -> Yellow light

4.5.6.3 Controlling of the 4 illuminant buttons solely by 31H

CCB	Description
CCB LEN(L)= length of CCB	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 0x60	0x60: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	31H
SUB-NODE	0x01~0xFA (address) or 0xFC (Broadcasting)
Data 0	Mask of the 4 lightable buttons(keys)
Data 1	Active status of the 4 keys
Data 2	Light mode of key 1& key 2
Data 3	Light mode of key 3& key 4

1) **Data [0]:** Setup mask of the illuminant buttons. In other words, to define whether each illuminant button could be controlled validly or not.

Bit 0 : 1st button's validation : = 1 (invalid, mask 1st button), = 0 (valid)
Bit 1 : 2nd button's validation : = 1 (invalid, mask 2nd button), = 0 (valid)
Bit 2 : 3rd button's validation : = 1 (invalid, mask 3rd button), = 0 (valid)
Bit 3 : 4th button's validation : = 1 (invalid, mask 4th button), = 0 (valid)
Bit 4 : reserved
Bit 5 : reserved
Bit 6 : reserved
Bit 7 : reserved

When the button is defined to valid, then the following configuration to each buttons will be accepted, otherwise if the specific button is masked, the new definition of the keys in following bytes will be ignored.

Example :
Data 0 = 0EH → 1st button is valid, the other 3 buttons are masked.
Data 0 = 0DH → 2nd button is valid, the other 3 buttons are masked.
Data 0 = 0BH → 3rd button is valid, the other 3 buttons are masked.
Data 0 = 07H → 4th button is valid, the other 3 buttons are masked.
Data 0 = 00H → 4 buttons are valid.

2) **Data [1]:** Define if each valid illuminant button (no mask) could be confirmed or not.

Bit 0 : = 1 (1st button can be confirmed), = 0 (CAN NOT)
Bit 1 : = 1 (2nd button can be confirmed), = 0 (CAN NOT)

Bit 2 : = 1 (3rd button can be confirmed), = 0 (CAN NOT)
Bit 3 : = 1 (4th button can be confirmed), = 0 (CAN NOT)
Bit 4 : reserved
Bit 5 : reserved
Bit 6 : reserved
Bit 7 : reserved

Example :
Data 0 = 00H → 4 buttons are valid.
Data 1 = 0FH → 4 buttons are all can be confirmed.

3) **Data [2]:** Define the light mode of the button 1 & 2 when it is valid by Data 0's configuration.

Bit 0 : = Light status of button 1
Bit 1 : = Light status of button 1
Bit 2 : = Light status of button 1
Bit 3 : = reserved
Bit 4 : = Light status of button 2
Bit 5 : = Light status of button 2
Bit 6 : = Light status of button 2
Bit 7 : = reserved

Bit assignment:

7	6	5	4	3	2	1	0

0 0 0-------- Light is OFF(button 1)
0 0 1-------- Light is ON (button 1)
0 1 0-------- Light is blinking by 2 second interval(button 1)
0 1 1-------- Light is blinking by 1 second interval (button 1)
1 0 0-------- Light is blinking by 0.5 second interval (button 1)
1 0 1--------Light is blinking by 0.25 second interval (button 1)

0 0 0---------- Light is OFF(button 2)
0 0 1---------- Light is ON(button 2)
0 1 0---------- Light is blinking by 2 second interval (button 2)
0 1 1---------- Light is blinking by 1 second interval (button 2)
1 0 0---------- Light is blinking by 0.5 second interval (button 2)
1 0 1----------Light is blinking by 0.25 second interval (button 2)

Example :
Data 2 = 51H →1st button is solid and 2nd button is blinking quickly.

4) **Data [3]:** Configure the light mode of the button 3 & 4 when it is valid by Data 0's configuration..

Bit 0 : = Light status of button 3
Bit 1 : = Light status of button 3
Bit 2 : = Light status of button 3
Bit 3 : = reserved
Bit 4 : = Light status of button 4
Bit 5 : = Light status of button 4
Bit 6 : = Light status of button 4
Bit 7 : reserved

Bit assignment:

7	6	5	4	3	2	1	0

0 0 0-------- Light is OFF(button 3)
0 0 1-------- Light is ON (button 3)

0	1	0-------- Light is blinking by 2 second interval (button 3)
0	1	1-------- Light is blinking by 1 second interval (button 3)
1	0	0-------- Light is blinking by 0.5 second interval (button 3)
1	0	1--------Light is blinking by 0.25 second interval (button 3)
0	0	0---------- Light is OFF(button 4)
0	0	1---------- Light is ON(button 4)
0	1	0---------- Light is blinking by 2 second interval (button 4)
0	1	1---------- Light is blinking by 1 second interval (button 4)
1	0	0---------- Light is blinking by 0.5 second interval (button 4)
1	0	1----------Light is blinking by 0.25 second interval (button 4)

Example :

Data 3 = 11H →3rd button is solid on and 4th button is solid on, too.

4.5.6.4 Return message while sending by 31H

CCB	Description
CCB LEN(L)= length of CCB	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 0x60	0x60: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	64H
SUB-NODE	0x01~0xFA (address) or 0xFC (Broadcasting)
Data 0	Message type = 04H
Data 1	Button index

When sending message to AT503-4K by 31H, and push the illuminant buttons, then AT503-4K will return the message with sub-command = 64H and Data 0 = 04H and Data 1 to notify the button index

Data 1 = 00H -> 1st button -> Red light
Data 1 = 01H -> 2nd button -> Green light
Data 1 = 02H -> 3rd button -> Blue light
Data 1 = 03 H -> 4th button -> Yellow light

4.5.6.5 Clear all the display and buffer

CCB	Description
CCB LEN(L)= length of CCB	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 0x60	0x60: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	01H
SUB-NODE	0x01~0xFA (address) or 0xFC (Broadcasting)

This command will either clear all the sending data into buffers, or clear all the buffers when all the message have been confirmed back. Before buffer is clear, you can use the 4 illuminant buttons to enable the re-display function.

4.5.6.6 Function keys return when buffer is clear

CCB	Description
CCB LEN(L)= length of CCB	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 0x60	0x60: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	64H
SUB-NODE	0x01~0xFA (address) or 0xFC (Broadcasting)
Data 0	Message type = 05H
Data 1	Button index

When key_code return is enabled, and AT503-4K does not display anything both on illuminant buttons and LED display, the these 4 buttons can be function keys. **But before this function is enabled, it needs to send sub-command = 01H to clear the buffers firstly, otherwise it will conflict with "re-display function".**
When push it, it will return message back with sub-command = 64H and Data 0 = 05H and Data 1 to notify the button index

Data 1 = 00H -> 1st button -> Red light
Data 1 = 01H -> 2nd button -> Green light
Data 1 = 02H -> 3rd button -> Blue light
Data 1 = 03 H -> 4th button -> Yellow light

4.5.6.7 Message return when buffer is empty

CCB	Description
CCB LEN(L)= length of CCB	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 0x60	0x60: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	64H
SUB-NODE	0x01~0xFA (address) or 0xFC (Broadcasting)
Data 0	Message type = 03H

Each time, when all AT503-4K's data in buffer have been confirmed back, then it will return a buffer empty message back to notify this kind of status.

4.5.6.8 Define the valid time duration on two contiguous pressing on the buttons.

CCB	Description
CCB LEN(L)= length of CCB	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 0x60	0x60: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	1FH
SUB-NODE	0x01~0xFA (address) or 0xFC (Broadcasting)
Data 0	Message type = 08H

Data 1	Delay time (default=02H, 00H ~ 0FH)

This function will define the valid & successful pressing on the button on two contiguous pressing. The default value of Data1 = 02H, And the actual delay time will be 0.4sec * Data 1, in other words, the default delay time is 0.8 sec for a valid action on the button between two contiguous pressing.

4.5.6.9 Tag mode configuration

Please refer the 4.3.20.3 & 4.3.20.4

4.5.6.10 AT503-4K auto-demonstrating function

CCB	Description
CCB LEN(L)= length of CCB	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 0x60	0x60: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	**32H**
SUB-NODE	0x01~0xFA (address) or 0xFC (Broadcasting)
Data 0	define buffer index and its specific light's status
Data 1	Display swapping time interval
Data 2	5th digit in LED
Data 3	4th digit in LED
Data 4	3rd digit in LED
Data 5	2nd digit in LED
Data 6	1st digit in LED
Dot	Dot position

1) **Data 0**: Define the buffer index of sending data and its specific illuminant button's status. Bit assignment of Byte is as below.

Bit 0 : LED status
Bit 1 : LED status
Bit 2 : LED status
Bit 3 : reserved
Bit 4 : Buffer index
Bit 5 : Buffer index
Bit 6 : reserved
Bit 7 : reserved

Bit assignment:

7	6	5	4	3	2	1	0	
					0	0	0--------	delete the this buffer data.
					0	0	1--------	Light is ON
					0	1	0--------	Light is blinking by 2 second interval
					0	1	1--------	Light is blinking by 1 second interval
					1	0	0--------	Light is blinking by 0.5 second interval
					1	0	1--------	Light is blinking by 0.25 second interval
0	0	----------						assign to 1st button
0	1	----------						assign to 2nd button
1	0	----------						assign to 3rd button

1 1---------- assign to 4th button

5) **Data 1:** Define the automatically display swapping time interval .

Bit 0 : reserved
Bit 1 : reserved
Bit 2 : reserved
Bit 3 : reserved
Bit 4 : reserved
Bit 5 : reserved
Bit 6 : Swapping time interval
Bit 7 : Swapping time interval

Bit assignment:

7	6	5	4	3	2	1	0

0 0 ---------- Time interval 1(0.5 sec)
0 1 ---------- Time interval 2(1 sec)
1 0 ---------- Time interval 3(2 sec)
1 1 ---------- Time interval 4(4 sec)

6) **Data 2 ~ Data 6:** We reserve maximum 5 digits could be displayed on AT503-4K's LED display.

7) **Data 7 (Dot) :** Decimal dot position.

Bit assignment:

7	6	5	4	3	2	1	0	
0	0	0	0	1	---------- 1st dot is on
0	0	0	1	0	---------- 2nd dot is on
0	0	1	0	0	---------- 3rd dot is on
0	1	0	0	0	---------- 4th dot is on
1	0	0	0	0	---------- 5th dot is on

4.5.7 Protocol for AT520

AT520 is a digit input & relay output device base on ABLEPick communication protocol. It has 4 inputs (DI) and 4 outputs (DO) and one RS232 serial port. Users can remote controlling device or monitoring status.

4.5.7.1 Product ID of I/O Devices

I/O TYPE: 0x30 → DI
0x31 → DO
0x34 → DI_AUTO response mode

4.5.7.2 Reading I/O Status：

※ Application Program(AP) sends reading command to TCP/IP controller：

CCB LEN(L) = 08H

CCB LEN(H) = 00H	
MESSAGE TYPE = 60H	Port 1, 61H :port 2
Reserved	
Reserved	
Reserved	
SUB-COMMAND = 3CH	
SUB-NODE = {01H ~ C8H}	Max. device address (up to 200)

※ TCP/IP controller reports the current Digit I/O status when it receive Sub-command = 3CH:

CCB LEN(L) = 0EH	
CCB LEN(H) = 00H	
MESSAGE TYPE = 60H	Port 1, 61H :port 2
Reserved	
Reserved	
Reserved	
SUB-COMMAND = 3CH	
SUB-NODE	0x01~0xC8
Reserved	00H
DI/DO type	
Data #1	ASCII of '0'~'9','A'~'F'
:	ASCII of '0'~'9','A'~'F'
Data #4	ASCII of '0'~'9','A'~'F'

DI/DO Type :
DI : 30H
DO : 31H
DO_AUTO : 34H

Data #1 : DI/DO Channel 5th to 8th
Data #2 : DI/DO Channel 1st to 4th
Data #3 : DI/DO Channel 13th to 16th
Data #4 : DI/DO Channel 9th to 12th

DI type :
Data #1 = high nibble of low byte of DI status
Data #2 = low nibble of low byte of DI status
Data #3 = high nibble of high byte of DI status
Data #4 = low nibble of high byte of DI status

Example: ch16 …… ch9……..ch1
0000 1100 1001 1111 ---→ 1: means ON, 0: means OFF
ASCII '0' 'C' '9' 'F'
means ------> ch 1, 2, 3, 4, 5, 8, 11, 12 is on ch 6, 7, 9, 10, 13, 14, 15, 16 is off

data #1 = 0x39('9')
data #2 = 0x46('F')
data #3 = 0x30('0')
data #4 = 0x43('C')

DO type :
Data #1 = high nibble of low byte of DO status
Data #2 = low nibble of low byte of DO status
Data #3 = high nibble of high byte of DO status
Data #4 = low nibble of high byte of DO status

Example: ch16 ……ch9……..ch1
0011 0100 0001 0010 --→ 1: means ON, 0: means OFF
ASCII '3' '4' '1' '2'

means ------> ch 2, 5, 11, 13, 14 is on ch 1, 3, 4, 6, 7, 8 , 9, 10, 12 , 15, 16 is off

data #1 = 0x31('1')
data #2 = 0x32('2')
data #3 = 0x33('3')
data #4 = 0x34('4')

※ TCP/IP controller reports the current Digit I/O status for DI_AUTO:

CCB LEN(L) = 12H	
CCB LEN(H) = 00H	
MESSAGE TYPE = 60H	Port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = 3CH	
SUB-NODE	0x01~0xC8
Reserved	
I/O type = 34H	34H:DI-AUTO
Data #1 (Now Status)	ASCII of '0'~'9','A'~'F'
:	ASCII of '0'~'9','A'~'F'
Data #4 (Now Status)	ASCII of '0'~'9','A'~'F'
Data #5 (last time Status)	ASCII of '0'~'9','A'~'F'
:	ASCII of '0'~'9','A'~'F'
Data #8 (last time Status)	ASCII of '0'~'9','A'~'F'

DI-AUTO type:

Data #1 ~ Data #4 = the current status for channel 1 to 16
Data #5 ~ Data #8 = last time status for channel 1 to 16

4.5.7.3 Enable the automatic resporting DI status: (DI_AUTO)

CCB LEN(L) = 0x0D	
CCB LEN(H) = 0x00	
MESSAGE TYPE =60H	Port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = 3AH	CONF CMD
SUB-NODE	0x01~0xC8/ 0xFC
40H	
1BH	
1BH	
11H	
'A'	Enable auto mode

4.5.7.4 Disable the automatic resporting DI status (DI_AUTO)

CCB LEN(L) = 0x0D

CCB LEN(H) = 0x00	
MESSAGE TYPE =60H	Port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = 3AH	CONF CMD
SUB-NODE	0x01~0xC8/ 0xFC
40H	
1BH	
1BH	
11H	
'N'	Disable auto mode

4.5.7.5 Write DO Value with mask

CCB

CCB LEN(L) = 0x11	
CCB LEN(H) = 0x00	
MESSAGE TYPE = 60H	Port 1
Reserved	
Reserved	
Reserved	
SUB-COMMAND = 3DH	
SUB-NODE	0x01~0xC8 or 0xFC
Reserved	
Low byte data #1	High nibble of mask value
data #2	Low nibble of mask value
data #3	High nibble of DO value
data #4	Low nibble of DO value
High byte data #5	High nibble of mask value
data #6	Low nibble of mask value
data #7	High nibble of DO value
data #8	Low nibble of DO value

Note: The DO bit value is changed when the corresponding mask bit is '0' ◦

Processing method:

A= original DO value
B= DO value specified by AP
M= Mask value specified by AP
Result = (A and M) or (B and (not M))

Example:

data #1 = 0x30('0'), data #2 = 0x30('0')
data #3 = 0x46('F'), data #4 = 0x46('F')
result = 0xFF -> ch0 ~ch7 all 'ON'

data #5 = 0x46('F'), data #6 = 0x46('F')
data #7 = Any value, data #8 = Any value
result = ch8 ~ ch15 no change

5. Advanced way to configure device's node address

In addition to sub-command 3A which can configure device's new node address, our pick-to-light device provide another easy way by interacting with operator to configure the device's node address. This easy way seems to be just pressing buttons.

There are two modes of this function, one is checking mode, the other one is setup mode.

(1). Node address checking mode : display the specific node address.

CCB	Description
CCB LEN(L)= length of CCB	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 0x60	0x60: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	FBH
SUB-NODE	0xFC (Broadcasting)
Data 0	00H 01H~C8H (specific display node)

This function could ask all the devices to display the specific node address on Data 0. When all devices receive this function, it will have two screens to show up by swapping. First screen is device's original node address displayed by "[node]", ex : [002]. The second screen is the specific node address displayed by "node", ex: 004

[002]→ 004 → [002] → 004 → [002] → 004 ……

However, if Data 0 is 00H, then all the devices will use its original node address to display commutatively, as [002]→ 002 → [002] → 002 → [002] → 002 ……, and two screen's brightness is different.

By pressing device's button (black confirmed button or small white button which on the model without confirmation button), as below:

Then the return CCB is as below:

CCB	Description
CCB LEN(L)= length of CCB	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 0x60	0x60: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	64H
SUB-NODE	Pressed node
Data 0	00H
Data 1	16H

From byte on sub-node, it can know which node ‘s button is pressed.

(2). Node address setup mode : Can be re-configured to the specific node address.

CCB	Description
CCB LEN(L)= length of CCB	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 0x60	0x60: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	FBH
SUB-NODE	0xFC (Broadcasting)
Data 0	00H 01H~C8H (specific display node)
Data 1	= 55H (fixed check byte)
Data 2	= AAH (fixed check byte)

This function could ask all the devices to display the specific node address on Data 0. When all devices receive this function, it will have two screens to display commutatively. First screen is device’s original node address displayed by “[node]”, ex : [002]. The second screen is the specific node address which would like to re-configure, and display by “-node-”, ex : -004-

[002]→ -004- → [002] → -004- → [002] → -004- ……

However, if Data 0 is 00H, then all the devices will use its original node address to display commutatively, as [002]→ -002- → [002] → -002- → [002] → -002- ……, and two screen’s brightness is different.

By pressing device’s button (descried as above), then the device will re-setup its node address to the new specific one as Data 0 and return one message as below :

CCB	Description
CCB LEN(L)= length of CCB	Length of CCB in byte
CCB LEN(H) = 0x00	
MESSAGE TYPE = 0x60	0x60: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	64H
SUB-NODE	Pressed node
Data 0	00H
Data 1	16H

From byte on sub-node, it can know which node ‘s button is pressed and be configured into the device.

PS: When devices receive FB command and enter the node address checking or setup status, then you can stop and cancel this function by sending any other sub-command.

6. AT400 compact TCP/IP controller's alarm status control

AT400 has the full compatible communication commands with AT500. In addition to above control commands, we add on the alarm status control when there is abnormal voltage occurred to AT400, too high or too low.

6.1 Read Alarm status

SUB-COMMAND = F7**H**

CCB	Description
CCB LEN(L)= 09H	Length of CCB = 9 bytes
CCB LEN(H) =00H	
MESSAGE TYPE = 60H	60H: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	F7H
SUB-NODE	00H
Message Type	00H

6.2 Return Alarm status from AT400

SUB-COMMAND = F7**H**

CCB	Description
CCB LEN(L)= 13H	Length of CCB = 19 bytes
CCB LEN(H) =00H	
MESSAGE TYPE = 60H	60H: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	F7H
SUB-NODE	00H
Message Type	00H
Data(0)	New Power Voltage Status
Data(1)	Old Power Voltage Status
Data(2)	Reserved
Data(3)	Reserved
Data(4)	Reserved
Data(5)	Reserved
Data(6)	Reserved
Data(7)	Reserved
Data(8)	Reserved
Data(9)	Reserved

DATA[0] = New Power Voltage Status, 8-BIT: 1->alarm, 0->Idle
Bit 0 → Input Voltage too low
Bit 1 → Input Voltage too high
Bit 2 → reserved
Bit 3 → reserved
Bit 4 → reserved
Bit 5 → reserved
Bit 6 → reserved
Bit 7 → System abnormal

DATA[1] = Old/previous Power Voltage Status, 8-BIT: 1->alarm, 0->Idle

Bit 0 → Input Voltage too low
Bit 1 → Input Voltage too high
Bit 2 → reserved
Bit 3 → reserved
Bit 4 → reserved
Bit 5 → reserved
Bit 6 → reserved
Bit 7 → System abnormal

DATA[2~9] = Reserved

6.3 Enable/Disable the alarm auto-report function

If any alarm status is changed, the AT400 can send the alarm status message automatically (Default: Disable the alarm auto-report function) . The status message format is the same as the return message of the 'Read alarm status' command

SUB-COMMAND = F7**H**

CCB	Description
CCB LEN(L)= 0AH	Length of CCB = 10 bytes
CCB LEN(H) =00H	
MESSAGE TYPE = 60H	60H: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	F7H
SUB-NODE	00H
Message Type	01H : Set/Get the alarm auto-report mode
Data(0)	Mode type

DATA[0] = Set/Get the alarm auto-report mode
00H → Get the alarm auto-report mode
01H → enable the alarm auto-report mode
02H → disable the alarm auto-report mode

6.3.1 Return the alarm auto-report mode

SUB-COMMAND = F7**H**

CCB	Description
CCB LEN(L)= 0AH	Length of CCB = 10 bytes
CCB LEN(H) =00H	
MESSAGE TYPE = 60H	60H: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	F7H
SUB-NODE	00H
Message Type	01H : Get the alarm auto-report mode
Data(0)	Mode type

DATA[0] = Get the alarm auto-report mode
00H → the alarm auto-report mode is disabled
01H → the alarm auto-report mode is enabled

6.4 Enable/Disable the alarm auto-buzzer function

If any alarm event occurred, the AT400 can alarm by buzzer automatically. (Default: Enable the alarm auto-report function)

SUB-COMMAND = F7**H**

CCB	Description
CCB LEN(L)= 0AH	Length of CCB = 10 bytes
CCB LEN(H) =00H	
MESSAGE TYPE = 60H	60H: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	F7H
SUB-NODE	00H
Message Type	02H : Set/Get the alarm auto buzzer mode
Data(0)	Mode type

DATA[0] = Set/Get the alarm auto-report mode
00H → Get the alarm auto-buzzer mode
01H → enable the alarm auto-buzzer mode
02H → disable the alarm auto-buzzer mode

6.4.1 Return the alarm auto-buzzer mode

SUB-COMMAND = F7**H**

CCB	Description
CCB LEN(L)= 0AH	Length of CCB = 10 bytes
CCB LEN(H) =00H	
MESSAGE TYPE = 60H	60H: port1 of controller,
Reserved	
Reserved	
Reserved	
SUB-COMMAND	F7H
SUB-NODE	00H
Message Type	02H : Get the alarm auto-buzzer mode
Data(0)	Mode type

DATA[0] = Get the alarm auto-buzzer mode
00H → the alarm auto-buzzer mode is disabled
01H → the alarm auto-buzzer mode is enabled