

ABLEpick Application Program Interface (API)

User's Manual

**VERSION : 10
DATE : May. 1, 2012**

(C) Copyright of ATOP Tech. Inc., 2005, All Rights Reserved

Atop Technologies, Inc.

This document may contain Trade Secrets Information which is proprietary to ATOP Technologies Inc. The software describe in this document are furnished under a license agreement or nondisclosure agreement. The software and/or information may be used or copied only in accordance with the terms of the agreement. It is against the law to copy the software and/or information on any medium except as specifically allowed in the license or nondisclosure agreement. The purchaser may make one copy of the software for backup purposes. No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchasers personal use, without the express written permission of ATOP technologies Inc.

Information is subject to change without notice. ATOP Technologies Inc. makes no warranties, either expressed or implied with respect to the software herein described as to its quality, performance, including, without limitations to, its merchantability or its fitness for any particular purpose.

ATOP Technologies Inc. accepts no responsibility or liability for any damages or loss of business or revenue due to the use of this document.

All trademarks, registered trademarks and trade names mentioned in this document are the sole property of their respective holders.

Application Program Interface(API) for ABLEPick pick-to-light system

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.

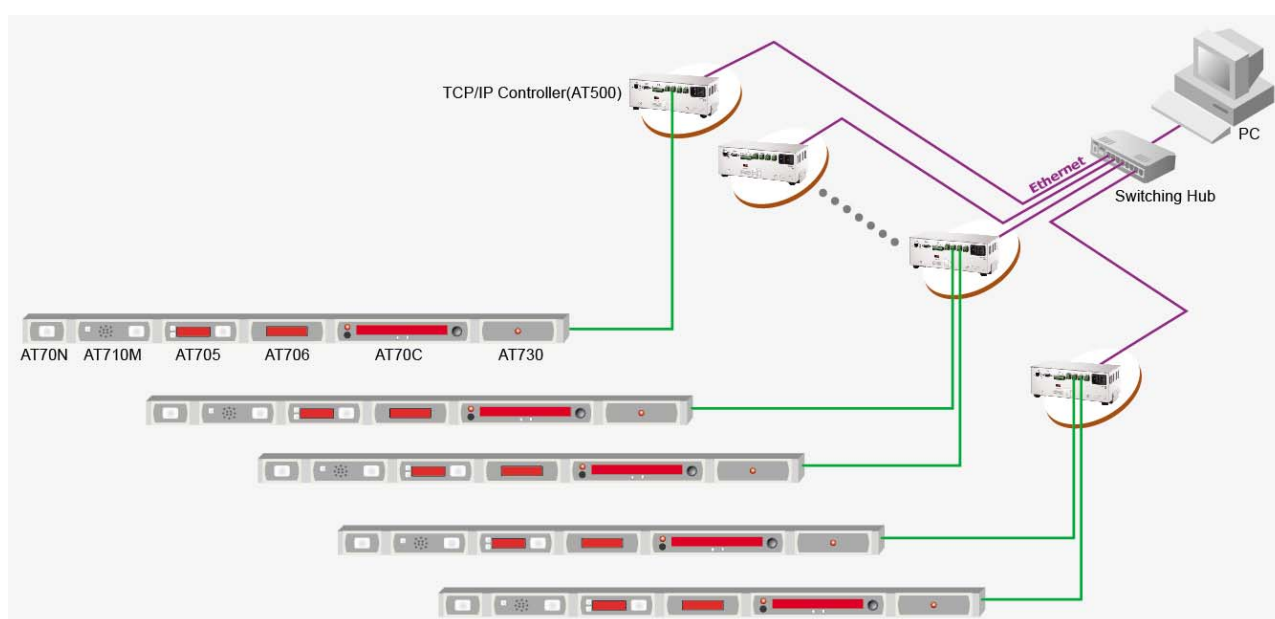


















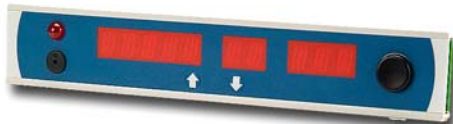



Figure ABLEPICK Architecture








The basic ABLEPick system includes a personal computer, a controller, pick tags, bay indicators, order displays 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	<p>TCP/IP controller</p> <p>A Maximum of 120 light module can be connected</p> <p>10/100 Mbps Ethernet TCP/IP protocol</p>	

AT400	Compact TCP/IP controller A maximum of 30 light modules 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 Confirm button 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 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	

**

ABLEPick FUNCTION LIBRARY

FUNCTION DESCRIPTION:

The table listed below describes the meaning of parameters in API function call.

Type	Parameter	Description
(N)	Gateway_ID	Address of Gateway.
(N)	Node_addr	Minus(-) address means picking tag is connected to Port1. Plus(+) address means picking tag is connected to Port2. If Node_Address is equal to 252 or -252 , it implies in "Broadcast mode", that means all devices on this Port accept the command.
(N)	DOT	Set decimal points on picking tag. 00H -> Do not display any decimal point. 01H -> Show decimal point on 1st digit. For instance, 123456. 02H -> Show decimal point on 2nd digit. For instance, 12345.6 04H -> Show decimal point on 3rd digit. For instance, 1234.56 08H -> Show decimal point on 4th digit. For instance, 123.456 10H -> Show decimal point on 5th digit. For instance, 12.3456 20H -> Show decimal point on 6th digit. For instance, 1.23456 To show multiple decimal points is to OR the above hex-codes. For instance, 03H =01H OR 02H -> Show decimal points on 1st and 2nd digits. e.g. "12345.6." 3FH =01H OR 02H OR 04H OR 08H OR 10H OR 20H -> Show decimal points on all digits. e.g. "1.2.3.4.5.6."

PS:AT500 TCP/IP controller has only one port, it is Port #1 .(ATOP has the other TCP/IP controller which has two or one ports). So node address should be always minus. AT400 also has port 1 to connect to lights.

TCP/IP controller's polling range

Polling range define the communication scope between TCP/IP controller and its connected device. If devices' node address is less than polling range value, then the devices can communicate with controller well. On the contrary, if device's node address is out of TCP/IP controller's polling range, then it will lose the communication with controller. If the device has LED display, then it will show up "FF" when it lose the communication with controllers.

AT500's default polling range is 250, AT400's default polling range is 30. After the installation ready, all the devices' node address also be 'assigned. It would be better to re-configure TCP/IP controller's polling range to be equal to the maximum node address which it connected to. That could let the whole system have the best communication performance.

By this way, we would suggest to assign the contiguous address number beginning from 001 to all the devices.,

Initialization

Re-configure the text file of "IPINDEX" to specify the TCP/IP controller's information. IPINDEX is a list of TCP/IP controller's IP address which DAPAPI.DLL could connect to. In other words, before activating the function calls, you need to tell DAPAPI.DLL all the TCP/IP controller's IP address which the program would like to control.

Function AB_API_Open()

Description : Startup and initialize the ABLEPick DAPAPI.DLL API function calls. The function will load the gateway configuration file: "ipindex" into memory. The file is at the same directory as application program. All gateway TCP/IP connection information is defined in this file. The file contents 3 fields: Gateway ID, TCP port, IP address. The purpose is to simplify to identify a gateway by gateway ID. The first field is the TCP/IP controller's ID (from 1, 2, 3.....). The second one is the TCP port, which is fixed, can not be changed. The third is the TCP/IP controller's IP address

Example for file: "ipindex" :

1	4660	10.0.0.15
2	4660	10.0.0.16
3	4660	10.0.0.17

Parameters : None

Return Value :

>0 -> OK
 <=0 -> error

Example : If AB_API_Open() <= 0
 WAIT WINDOW " ABLEPICK API open failed"
 endif

Function AB_API_Close()

Description : Shutdown the ABLEPICK API.

Parameters : None

Return Value :

0 -> OK
 -1 -> error

Example : If AB_API_Close() < 0
 WAIT WINDOW " ABLEPICK API close failed"
 endif

Function AB_GW_Open(INTEGER Gateway_ID)

Description : To build up a TCP connection with the specified TCP/IP controller. Normally, the program usually will Function AB_GW_Status to check about the connection status. Please refer the description of AB_GW_Status()

Parameters :

Gateway_ID
 0 -> current Gateway.
 >0 -> the specified Gateway.

Function AB_GW_Close(INTEGER Gateway_ID)

Description : Disconnect the TCP connection with the specified TCP/IP controller.

Parameters :

Gateway_ID
 0 -> current Gateway.
 >0 -> the specified Gateway.

Function AB_GW_Status(INTEGER Gateway_ID)

Description : Get the status of current Gateway or specified Gateway.

Parameters :

Gateway_ID
 0 -> current Gateway.
 >0 -> the specified Gateway.

Return Value :

7 -> means Gateway is normal.
-1 or 6 -> means Gateway fails to work.

Example : if AB_GW_Status(2) >=0
 WAIT WINDOW "Gateway 2 works normally at present"
else
 WAIT WINDOW "Gateway 2 fails to work at present"
endif

Function AB_GW_AllStatus (status As Byte)

Description : Get the TCP/IP connection status of all Gateways.

Parameters :

Output:

status -> A buffer for storing status of all gateways. Its data type is byte array. Each byte is stand for the status of each gateway sequentially. For example: status[0] is the status of the 1-st gateway, status[1] is the 2-nd, ...,etc. The value of status is as following:
0: TCP connection is closed
6: TCP connection is waiting to connect to the gateway
7: TCP connection is connected to the gateway

Return Value :

>= 0: Total count of gateways

Example :

Dim status[200] as byte

Count = AB_GW_Status(status[0])
msgbox "Total count of gateways=" + count

(2)Receive message from Gateway

Function AB_GW_RcvMsg(INTEGER Gateway_ID, DAPCCB rcv_ccb)

Description : Receive message from a Gateway. This function is a general message receiving function which need to decode the command protocol on the rcv_ccb parameters.

Parameters :

Gateway_ID

-1 -> any Gateway.

0 -> current Gateway.

>0 -> the specified Gateway.

rcv_ccb: response message

Return Value :

>0 -> means there is a message collected by specified Gateway.

0 -> means there is no message.

Example : Gateway_ID = 1

If AB_GW_RcvMsg(Gateway_ID, rcv_ccb) > 0

....

else

WAIT WINDOW " Presently there is no message"

endif

....

Function AB_Tag_RcvMsg (ByRef gatewayID As Long, ByRef tag_addr As Long, ByRef subcmd As Long, ByRef msg_type As Long, ByRef data As Byte, ByRef data_cnt As Long)

Description : Receive message from a specified field device.

Parameters :

Input: (sending by reference/address)

GatewayID : the designated gateway ID for receiving message

= 0: To receive message from any gateway.

> 0 : To receive message from the specified gateway. (Each Gateway is identified by an unique Gateway ID, value=1~255).

tag_addr: the designated node address for receiving message

= 0: receive from any node

= others: To receive message from the specified field device. (Each field device is identified by an unique ID, value=1~200).

Subcmd: the designated message command for receiving

= -1:receive any data,

= 0-255:designated message command

data: buffer for receiving message

data_cnt= length of receiving buffer

Output:

Gateway ID : the gatewayID where the data from.

tag_addr: the node address where data come from. < 0 : from port 1, > 0 from port 2.

subcmd: the 1-st message command (to distinguish message)

=-1:no used,

= 0-255: message command

msgtype: the 2-nd message command (only for some message, such as cmd = 64H)

=-1:no used,

= 0-255:message command

data: Received message

Return Value :

- >0 -> length of the received message.
- 0 -> No message being received.
- 1 ->: error

Example :

```
Dim data(200)
gatewayID = 0
node = 0
subcmd = -1
msgtype = -1
datacnt = 200
Ret = AB_Tag_RcvMsg(@gatewayID,@node, @subcmd, @msgtype, data(0), @datacnt)&& receive data
from any gateway message
```

If Ret > 0 && receiving one message from gatewayID

- The node value is the node address of the received message from specified "gatewayID"
- Node < 0 : from port 1
- Node > 0 : from port 2
- The sub-command of the received message is "subcmd" (please refer the below list)
- If the sub-command is some specified command, such as 64H, then "msgtype" will have a special meaning described as below.

If Ret = 0 && there is no message received.

Example :

```
Ret = AB_Tag_RcvMsg(2,@node, @subcmd, @msgtype, data(0), @datacnt) &&receive data from specified
gateway ID = 2
```

If ret > 0 : receiving one message from gateway ID = 2

If ret = 0 : no message received.

If ret < 0 : it might be there is no gateway ID = 2

Example :

```
Ret = AB_Tag_RcvMsg(2,-3, @subcmd, @msgtype, data(0), @datacnt) &&receive data from specified
gateway ID = 2, port 1 and a specified node = 3
```

If ret > 0 : receiving one message from gateway ID = 2, port 1 and node address = 3

If ret = 0 : no message received.

If ret < 0 : it might be there is no gateway ID = 2 or no node address = 3

Example :

```
Ret = AB_Tag_RcvMsg(@gatewayID, @node, 6, @msgtype, data(0), @datacnt) &&receive data from any
gateway ID, but just sub-command = 06 message
```

If ret > 0 : receiving one message with sub-command = 06H

If ret = 0 : no message received.

PS : Sub-command (subcmd) list

There are several kinds of sub-command received, as below:

- (1) 06H -> "confirmation key is pressed.
- (2) 07H -> shortage situation is happened.
- (3) 09H -> Device's status after sending diagnosis requirement. (AB_GW_TagDiag)
- (4) 0AH -> Time out, either the node address is not existed or the device's communication is fail.
- (5) 0CH -> Un-executed command.
- (6) 0DH -> 'button lock' , Tag's some button keeps pushed when message is sent to it at the same time. That would cause the sending message to disappear.
- (7) 0FH-> device is under 'stock mode', then it will return 0FH instead of 06H.
- (8) 24H-> When cycle edit function is enabled, then it will return message with 24H instead 06H.(please refer the function of AB_TAG_CycleEdit).
- (9) 64H -> see Function AB_Tag_RcvMsg0 to identify what kind of **message type** it is, as below:
- (10) FCH-> power on ready. After device power on, then it will return one message with sub-command FCH back.

PS : Message type (msg_type) : Get message type when the receiving sub-command is 64h

Return Value :

- = 0, it means it is the key code return, in other words could be for the any key return function.
- = 1, it means tag is busy. It is the situation that the message queued in tag not be polled back by TCP/IP controller, and at same time TCP/IP control still sends one message to it. Then the latter sending message will be drop off and return one message with sub-command = 64H and data0 is 1 to inform this situation happened.
- = 2, This message is only for AT503A, AT502V and AT506-3W-123, when their buffer is full and the first message has been replaced
- = 3 , The last message in buffer has been displayed on. Just for AT506-3W-123.
- = 4 , The return message from the blue function key on the model of AT706-24-3K, AT706-3K or AT705-RFID

PS : “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 Function AB_TAG_Mode). 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. Due to there is still another kind of message with sub-command 64h. so check the message type is necessary. If the message type = 0, then the below function AB_GE_RcvButton can know which buttons be push down to cause the key code return.

Function AB_GW_RcvButton(Byref data As Byte, ByRef data_cnt As Long)

Description : Get the button type when the value gotten by AB_Tag_RcvMsg () and the sub-command = 64H

Parameters : data : the same “data” value received from AB_Tag_RcvMsg()

 Data_cnt : the same “data_cnt” value received from AB_Tag_RcvMsg()

Return Value :

- 1 : Only CONFIRM button
- 2 : Only UP button
- 3 : Only DOWN button
- 4 : First UP, then CONFIRM
- 5 : First DOWN , then CONFIRM
- 6 : First UP and DOWN , then CONFIRM
- 7 : First DOWN , then UP
- 8 : First CONFIRM , then UP
- 9 : First DOWN and CONFIRM , then UP
- 10 : First UP , then DOWN
- 11 : First CONFIRM , then DOWN
- 12 : First UP and CONFIRM , then DOWN

Function AB_GW_TagDiag(INTEGER Gateway_ID, INTEGER Port_ID)

Description : Request the status of devices which are connected to a port of specified Gateway

Parameters : Gateway_ID

 Gateway_ID

 0 -> current Gateway.

 >0 -> the specified Gateway.

 Port_ID

 1 -> Port 1

 2 -> Port 2

If this request is ok, you can receive the data by AB_GW_RcvMsg() or AB_Tag_RcvMsg().

By function of AB_GW_RcvMsg()

The 7th byte of received data is sub-command = 09H

The 8th byte of received data is max tag node

The 12th ~ 43th byte (status data): a data string with 32 bytes long store the devices’ status, 8 bits of each byte stand for status of 8 devices on the same port of Gateway. Therefore 32 bytes can

show up to 256 devices status

By function of AB_Tag_RcvMsg()

The 4th ~ 35th byte (status data) : a data string with 32 bytes long store the devices' status, 8 bits of each byte stand for status of 8 devices on the same port of Gateway. Therefore 32 bytes can show up to 256 devices status

Bit 0 of 1st byte of status data stands for status of Node 1
 Bit 1 of 1st byte of status data stands for status of Node 2
 Bit 2 of 1st byte of status data stands for status of Node 3

...

Bit 7 of 1st byte of status data stands for status of Node 8
 Bit 0 of 2nd byte of status data stands for status of Node 9

...

Bit 7 of 2nd byte of status data stands for status of Node 16

...

Bit 6 of 25th byte of status data stands for status of Node 199
 Bit 7 of 25th byte of status data stands for status of Node 200

...

Bit N of Mth byte of status data stands for status of Node (M-1)*8 + N+1

Bit = 0 means device located at the specified address is normal .
 Bit = 1 means no device or device on the specified address fails to work.

Example :

```
&& Request the status of devices which are connected to Gateway=2, Port=1.
Dim data(200), gateway =0, node = 0, subcmd = -1, msgtype = -1
Status = AB_GW_TagDiag(2,1)
if AB_AB_RcvMsg(gateway,node,subcmd,msgtype,data(0), 200 ) > 0

    if SubCmd == 9
        WAIT WINDOW "receive status data"
    Endif
    if SubCmd == 7
        WAIT WINDOW "Push shortage key"
    endif
    ...
endif
```

we can dump the received data string into the "0"and"1" data string

```
00100110 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00001011 11111111
11111111 11111111 11111111 11111111 11111111 11111111
11111111 11111111 11111111 11111111 11111111 11111111
11111111 11111111 11111111 11111111 11111111 11111111
11111111 11111111 11111111 11111111 11111111 11111111
11111111 11111111
```

** Listing as above means at most 38 devices are connected to Gateway=2, Port=1. and device with address 37 fails or non-exist.

(3) Send message to picking tag

Function AB_LB_DspNum(INTEGER Gateway_ID, INTEGER Node_addr, LONG Disp_INT, BYTE Dot, INTEGER Interval)

Description : Show numeral on LED digits panel of picking tag.

Parameters :

Node_Addr: address, <0:port1, >0:port 2
 Disp_INT: display data
 Dot: digit points, bit1:1-st digit point, bit2:2-nd digit point
 Interval
 0: normal display
 -1: blinking with default frequency
 >0: set blinking frequency (by msec)
 -2: turn off digit
 -3: turn off digits and lamp

Example :

```
AB_LB_DspNum(2,-5,100,0,0) && Send numeral "100" to tag and twinkle at a frequency
    2 : Gateway ID
        -5: Send to picking tag of address 5 on Port 1
        100: Number shown on picking tag
        0: Do not display decimal point
        0: Do not twinkle.
AB_LB_DspNum(3, 26,385,2,300) && Send numeral "38.5" to picking tag
    3 : Gateway ID
        26: Address of picking tag on Port 2.
        385: Number displayed on picking tag's panel.
        2: Display decimal point on second digit.
        300: Digits twinkle at a period 300 msec.
AB_LB_DspNum(1, -252,888888,63,250) && Make all tags on Port1 show 8.8.8.8.8. and twinkle at a
    frequency.
    1 : Gateway ID
        -252: Broadcast mode, all devices accept command.
        888888: Number will be displayed on picking tags
        63: =Hex(3FH) display all decimal points.
        250: Digits twinkle at a period 250 msec.
```

Function AB_LB_DspStr(INTEGER Gateway_ID, INTEGER Node_Addr, STRING Disp_STR, BYTE Dot, INTEGER Interval)

Description : Show data string on LED digits panel of picking tag.

Parameters :

Node_Addr: tag address, <0:port1, >0:port 2
 Disp_STR: display data
 Dot: digit points, bit1:1-st digit point, bit2:2-nd digit point ...
 Interval
 0: normal display
 -1: blinking with default frequency
 >0: set blinking frequency (by msec)
 -2: turn off digit
 -3: turn off digits and lamp

Example :

```
AB_LB_DspStr(3, -5,"CH-10",0,0) && Show data string CH-10 on picking tag.
    3: Gateway ID
        -5: Send to picking tag of address 5 on Port 1
        CH-10: Data string shown on picking tag
        0: Do not display decimal point
        0: Do not twinkle.
AB_LB_DspStr(1, -252,"960123",8,300) && Show data string 960.123 on all picking tags -252:
Broadcasting
    1: Gateway ID
```

960123: Show data string 960123 on tag
 Hex(08H) Show decimal point on 3rd digit.
 300: LED digits twinkle at a period 300 msec

Function **AB_LB_DspAddr**(INTEGER Gateway_ID, INTEGER Node_addr)

Description : Picking tag shows its address on LED digits panel.

Parameters :

Node_Addr: tag address, <0:port1, >0:port 2

Example :

AB_LB_DspAddr(2, -16) && Make picking tag on Gateway 2, Port 1, address 16 show its address .
 AB_LB_DspAddr(2, 16) && Make the picking tag on Gateway 2, Port 2, Address 16 show its address.
 AB_LB_DspAddr(2, -252) && Make all picking tags on Gateway 2, Port 1 show their addresses.

Function **AB_LED_Dsp**(INTEGER Gateway_ID, INTEGER Node_Addr, BYTE Lamp_STA, INTEGER Interval)

Description : Control state of Lamp .

Parameters :

Node_Addr: tag address, <0:port1, >0:port 2

Lamp_STA

0: LED off

1: LED on

2: LED blinking

Interval:

0: normal display

>0: blinking frequency (by msec)

Example :

AB_LED_Dsp(3, -5, 1, 0) && Turn on lamp of tag which is located on Gateway 3 、 Port 1 and address is 5.
 AB_LED_Dsp(3, 17, 0, 0) && Turn off lamp of tag which located on Gateway 3 、 Port 2 and address is 17.
 .AB_LED_Dsp(3, -252, 2, 300) && All lamps located at Gateway 3 、 Port 1 twinkle with period 300msec.

Function **AB_BUZ_On**(INTEGER Gateway_ID, INTEGER Node_Addr, BYTE Buzzer_Type)

Description : Control buzzer "on" or "off".

Parameters :

Node_Addr: tag address, <0:port1, >0:port 2

Buzzer_Type: 1:buzzer on, 0:buzzer off

Example :

.AB_BUZ_On(3, -5, 1) && Turn on the buzzer which is on Gateway 3 、 Port 1 and address is 5 .
 .AB_BUZ_On(3, 17, 0) && Turn off the buzzer which is on Gateway 3 、 Port 2 and address is 17.
 .AB_BUZ_On(3, -252, 1) && Turn on all buzzers which are on Gateway 3 、 Port 1

Function **AB_TAG_Reset**(INTEGER Gateway_ID, INTEGER Node_addr)

Description : Make picking tag reset itself.

Parameters :

Node_Addr: tag address, <0:port1, >0:port 2

Example :

AB_TAG_Reset(2, -5) && rest a picking tag on Gateway 2 、 Port 1 、 Address 5
 .AB_TAG_Reset(2, 252) && Reset all picking tags on Gateway 2 、 Port 2.

Function **AB_GW_SetPollRang** (INTEGER Gateway_ID, INTEGER max_node)

Description : Set polling range of each port on Gateway.

Parameters :

max_node: (from 1 to 200), set Polling Range from 1 to max_node

>0: means port2.

<0: means Port1.

Example :

AB_GW_SetPollRange(2,-120) && Set maximum polling address as 120 on Port 1 of Gateway 2.

AB_GW_SetPollRange(3, 100) && Set maximum polling address as 100 on Port 2 of current Gateway 3.

Function AB_LB_SetMode(INTEGER Gateway_ID, INTEGER Node_Addr, BYTE pick_mode)

Description : Set picking tag in "picking mode" or "stock mode".

Parameters :

Node_Addr: tag address, <0:port1, >0:port 2

pick_mode= 0:picking mode, 1:stock mode

Example :

AB_LB_SetMode(3,-252, 1) && Set all picking tags on Gateway 3 · Port 1 in "stock mode".

***Please note that when device has been configured to stock mode, then its return message's sub-command would be 0FH instead 06H**

Function AB_LB_Simulate(INTEGER Gateway_ID, INTEGER Node_Addr, BYTE simulate_mode)

Description : Make picking tag simulate "push confirmation key" or "push shortage key".

Parameters :

Node_Addr: tag address, <0:port1, >0:port 2

simulate_mode= 1:push confirmation mode, 2:push shortage mode

Example .:

AB_LB_Simulate(2, -252, 1) && Make all picking tag on Gateway2, Port 1 send back simulation message of "push confirmation key".

Function AB_LB_SetLock(INTEGER Gateway_ID, INTEGER Node_Addr, BYTE Lock_State, BYTE Lock_Key)

Description : Enable/Disable all keys on picking tags.

Parameters :

Node_Addr: tag address, <0:port1, >0:port 2

Lock_State= 0: unlock key, 1:lock key

Lock_Key= 1: confirm key, 2:shortage key

Example :

AB_LB_SetLock(1,-252,1,2) && Disable shortage keys of all picking tags which are connected to Gateway1, Port 1.

AB_LB_SetLock(1,-252,0,3) && Enable all keys on all picking tags which are connected to Gateway1 , Port 1.

AB_LB_SetLock(1,-252,1,1) && Disable confirm keys of all picking tags which are connected to Gateway 1 · Port 1.

Function AB_LED_Status (INTEGER Gateway_ID, INTEGER Node_Addr, BYTE Lamp_Color, BYTE Lamp_STA)

Description : Control the color and state of Lamp on picking tag

Parameters :

Node_Addr: tag address, <0:port1, >0:port 2

Lamp_color:

0: RED 3: BLUE (for AT7XX series)

1: GREEN 4: PINK (for AT7XX series)

2: AMBER 5: CYAN (for AT7XX series)

Lamp_STA:

0: LED Off

1: LED on

2~5: set blinking frequency (2 → 2 sec, 3 → 1 sec, 4 → 0.5 sec, 5 → 0.25 sec)

Example :

AB_LED_Status(1,-252,0,4) && Let all the tag's display blinking green light on Gateway1, Port 1.

Function AB_TAG_CountDigit (INTEGER Gateway_ID, INTEGER Node_Addr, BYTE Digit)

Description : To setup the counting digits when the tag is changed to "stock mode"

Parameters :

Node_Addr: tag address, <0:port1, >0:port2

Digit: the counting digit (1~5)

Example :

AB_LB_SetMode(2,-252, 1) && Set all tags on Gateway 2 Port 1 in "stock mode".

AB_TAG_CountDigit(2,-252,3) && Set all tags' count digits to 3 digits on Gateway 2 port 1.

Function AB_TAG_ButtonDelay (INTEGER Gateway_ID, INTEGER Node_Addr, BYTE DelayTime)

Description : Define the time interval between two useful confirmation at the same button. This function can be used only when the "key code return" bit is enabled (see the function AB_TAG_mode). In the normal situation, only when the LED display show something on it, and push the confirmation button, then it will return one message to AP. But if you enable the "key code return" bit, even though there is nothing shown on the LED display, and push the confirmation button, then it will return one message to AP, too. Under this situation, this return message's sub-command will be 64H different from 06H.

Parameters :

Node_Addr: tag address, <0:port1, >0:port2

DelayTime: (1~5) The default value is 0.8 sec.

Example :

AB_TAG_ButtonDelay(1,-252,2) && Set all tags' button on Gateway 1 port 1. to delay 1.6 sec.

Function AB_Tag_ChgAddr(INTEGER Gateway_ID, INTEGER old_Addr, INTEGER new_Addr)

Description : To configure device's new node address

Parameters :

old_Addr & new_Addr : tag address, <0:port1, >0:port2

Example :

AB_Tag_ChgAddr(2,-3,-8) && Configure device with node address 3 on gateway 2 into new address 8.

(4) How to configure the up-count & down-count button

Function AB_TAG_mode (INTEGER Gateway_ID, INTEGER Node_Addr, INTEGER Save_mode, BYTE mode_type)

Description : Configure the button's behavior of picking tag.

Parameters :

Node_Addr: tag address, <0:port1, >0:port2

Save_mode: the setting value will be saved into the EEPROM or nor.

1 -> save into the EEPROM

0 -> do not save into the EEPROM

Mode_type: there is one byte to record the configuration of the behavior of buttons

- (1) **Default mode_type** value = 115 (bit 0 + bit 1 + bit 4 + bit 5 + bit 6 = 1+2+16+32+64) : The down-count button is the shortage button, when the LED display turn on, push the shortage button(down-count), then it will be off, then one shortage message with value = 0 will be rerun back with sub-command = 07H.

Bit	Description	Enabled	disabled
0	Confirmation button	•	
1	Shortage (down-count) button	•	
2	Edit button(up-count) button		•
3	Key code return		•
4	Communication time-out display	•	
5	Node address configuration	•	
6	Redisplay	•	
7	0	-	-

- (2) If you would like the tag can behave the partially picking function (up-count & down-count button could be used to adjust the quantity), the value will be **117**, enabled the edit button and disable the shortage button (bit 0 + bit 2 + bit 4 + bit 5 + bit 6 = 1+4+16+32+64)

Bit	Description	Enabled	disabled
0	Confirmation button	•	
1	Shortage (down-count) button		•
2	Edit button(up-count) button	•	
3	Key code return		•
4	Communication time-out display	•	
5	Node address configuration	•	
6	Redisplay	•	
7	0	-	-

- (3) If you would like to enable the "key code return" function, then the code value will be 123 for normal mode, and 125 for the partially picking mode.

Example :

AB_TAG_mode(2, 252, 1, 117) && Set all tags' mode value on Gateway 2 port 2 to 117 and save it into the EEPROM. to enable the up-count & down-count button to adjust quantity.

(5) Send message to 12-digits Alphanumeric display(AT50C/AT70C)

Function AB_AHA_DspStr (INTEGER Gateway_ID, INTEGER Node_Addr, PCHAR Disp_Str, BYTE BeConfirm, BYTE DigitSta)

Description : Show data string on LED digits panel of Alphanumeric display

Parameters :

Node_Addr: address, <0:port1, >0:port 2

Disp_STR: display data

BeConfirm:

1 → means the display message on LED display can be confirmed by pushing the confirmation button

0 → means the display message on LED display can not be confirmed by pushing the confirmation button

DigitSta:

1: 12-digits LED display on.

2~5: set blinking frequency (slow ~ quick)

Example :

AB_AHA_DspStr(2,-25,"1234567890",1,1) && Display "1234567890" on the alphanumeric display and this message can be confirmed by pushing the confirmation button.

AB_AHA_DspStr(2,-25,"NO CONFIRM",0,4) && Display blinking "NO CONFIRM" on the alphanumeric display and this message can not be confirmed by pushing the confirmation button.

Function AB_AHA_ClrDsp (INTEGER Gateway_ID, INTEGER Node_Addr)

Description : Clear data string stored in the buffer of LED digits panel of Alphanumeric display

Parameters :

Node_Addr: address, <0:port1, >0:port 2

Example :

AB_AHA_DspStr(2,-25) && Clear all the message data stored in the buffer of alphanumeric display.

Function AB_AHA_ReDsp (INTEGER Gateway_ID, INTEGER Node_Addr)

Description : Show the last display data string on LED digits panel of Alphanumeric display

Parameters :

Node_Addr: address, <0:port1, >0:port 2

Example :

AB_AHA_ReDsp(1,-10) && Redisplay the last message on node 10.

Function AB_AHA_LED_Dsp (INTEGER Gateway_ID, INTEGER Node_Addr, BYTE Lamp_Type, INTEGER Lamp_STA)

Description :To control the LED indicator of Alphanumeric Display ON or blinking with different period .

Parameters :

Node_Addr: address, <0:port1, >0:port 2

Lamp_Type: define the which LED to work.

0: LED 1

1: LED 2 (reserved)

2: Two LEDs

Lamp_STA

0: LED OFF

1: LED ON

2~5: set blinking frequency (slow ~ quick)

Example :

AB_AHA_LED_Dsp(2,-10,0,1) && Light on the light of alphanumeric display on node 10 gateway 2 port 1.

AB_AHA_LED_Dsp(2,-18,0,4) && Blinking the light of alphanumeric display on node 18 gateway 2 port 1.

Function AB_AHA_BUZ_On (INTEGER Gateway_ID, INTEGER Node_Addr, INTEGER Buz_STA)

Description : Control the buzzer on the Alphanumeric Display "on" or "off".

Parameters :

Node_Addr: address, <0:port1, >0:port 2

Buz_STA:

0: buzzer OFF

1~5: buzzer ON, and set buzzer frequency (slow ~ quick)

Example :

AB_AHA_BUZ_On(2,-10,1) && Light on the Buzzer of alphanumeric display on node 10 gateway 2 port 1.

Function AB_AHA_DEMO_Control (INTEGER Gateway_ID, INTEGER Node_Addr, INTEGER Con_STA)

Description : **Only for model AT50C-20**, Control the demonstration on the Alphanumeric Display status. This function needs to send once using AB_AHA_DspStr to sending message in order to ask the AT50C-20 to start demonstration.

Parameters :

Node_Addr: address, <0:port1, >0:port 2

Con_STA:

0: STOP the demonstration

1: START the demonstration

2: PAUSE the demonstration

Example : For demonstration purpose on **AT50C-20**

AB_AHA_DspStr(1,-1,"DISPLAY-1",1,1) && sending "DISPLAY-1" to buffer on gateway ID=1, node =1

AB_AHA_DspStr(1,-1,"DISPLAY-2",1,1)

AB_AHA_DspStr(1,-1,"DISPLAY-3",1,1)

AB_AHA_DspStr(1,-1,"DISPLAY-4",1,1)

AB_AHA_DEMO_Control(1,-1,1) && asking AT50C-20 to start to do the demonstration.

AB_AHA_DEMO_Control(1,-1,0) && asking AT50C-20 to STOP to do the demonstration.

AB_AHA_DEMO_Control(1,-1,2) && asking AT50C-20 to PAUSE to do the demonstration.

Function AB_AHA_DEMO_Interval (INTEGER Gateway_ID, INTEGER Node_Addr, INTEGER Interval)

Description : **Only for model AT50C-20**, Control the demonstration time interval between each message.

Parameters :

Node_Addr: address, <0:port1, >0:port 2

Interval: Interval time = 0.5 sec * Interval

Example : For demonstration purpose on **AT50C-20**

AB_AHA_DEMO_Interval(1,-1,1) && Time interval is 0.5 sec for demonstration.

AB_AHA_DEMO_Interval(1,-1,2) && Time interval is 1 sec for demonstration..

(6) Send message to 3-digit directional and 2-digit vertical & directional pick tag(AT503A&AT502V)

Function AB_AV_LB_DspNum(INTEGER Gateway_ID, INTEGER Node_addr, LONG Disp_INT, BYTE Arrow, BYTE Dot, INTEGER Interval)

Description : Show numeral message and assign the proper arrow on AT503A and AT502V.

Parameters :

Node_Addr: address, <0:port1, >0:port 2

Disp_INT: display data

Arrow :

0: all arrows are off.

1: right arrow is on for AT502V, and up arrow is on for AT503A

2: left arrow is on for AT502V, and down arrow is on for AT503A

3: both arrows are on for AT502V and AT503A.

>3: all arrows are off.

Dot: digit points, bit1:1-st digit point, bit2:2-nd digit point

Interval

0: normal display

-1: blinking with default frequency

>0: set blinking frequency (by msec)

-2: turn off digit

-3: turn off digits and lamp

Example :

AB_AV_LB_DspNum(2-5,100,1,0,0) && Send numeral "100" to pick tag and assign up/right arrow on.

2: Gateway ID

-5: Send to picking tag of address 5 on Port 1

100: Number shown on picking tag

1: right/up arrow is on

0: Do not display decimal point

0: Do not twinkle.

AB_AV_LB_DspNum(3, 26,385,2,2,300) && Send numeral "38.5" to picking tag and assign left/down arrow is on.

3: Gateway ID

26: Address of picking tag on Port 2.

385: Number displayed on picking tag's panel.

2: left/down arrow is on

2: Display decimal point on second digit.

300: Digits twinkle at a period 300 msec.

AB_AV_LB_DspNum(4,-252,888888,3,63,250) && Make all tags on Port1 show 8.8.8.8.8. and twinkle at a frequency.

4: Gateway ID

-252: Broadcast mode, all devices accept command.

888888: Number will be displayed on picking tags

3: both arrows are off.

63: =Hex(3FH) display all decimal points.

250: Digits twinkle at a period 250 msec.

(7) Send message to 6-digit,3 separated windows pick tag(AT506-3W-123)

Function AB_3W_LB_DspNum(INTEGER Gateway_ID, INTEGER Node_addr, STRING Row, STRING Col, LONG Disp_INT, BYTE Dot, INTEGER Interval)

Description : Show numeral message and on AT506-3W-123.

Parameters :

Node_Addr: address, <0:port1, >0:port 2
 Row : the row information of the display data.(one digit)
 Col : the column information of the display data.(two digits)
 Disp_INT: display data
 Dot: digit points, bit1:1-st digit point, bit2:2-nd digit point
 Interval
 0: normal display
 -1: blinking with default frequency
 >0: set blinking frequency (by msec)
 -2: turn off digit
 -3: turn off digits and lamp

Example :

AB_3W_LB_DspNum(2, -8,"1", "03", 12,0,0) && Send numeral "12" to location with Row is "1" and "Column" is "03".

2: Gateway ID
 -8: Send to picking tag of address 5 on Port 1
 "1" the Row information of location.
 "03" : the column information of location
 12: Number shown on picking tag
 0: Do not display decimal point
 0: Do not twinkle.

AB_3W_LB_DspNum(4,16,"4","12",385,2,300) && Send numeral "38.5" to picking tag and twinkle at a frequency.

4: Gateway ID
 16: Address of picking tag on Port 2.
 "4" the Row information of location
 "12": the column information of location
 385: Number displayed on picking tag's panel.
 2: Display decimal point on second digit.
 300: Digits twinkle at a period 300 msec.

(8) Send message to Melody completion indicator (AT510M/AT710M)

Function AB_Melody_On(INTEGER Gateway_ID, INTEGER Node_Addr, BYTE Song, BYTE Buzzer_Type)

Description : Control assign the melody song on AT510M.

Parameters :

Node_Addr: tag address, <0:port1, >0:port 2

Song: 1~12

- (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

Buzzer_Type: 1:buzzer on, 0:buzzer off

Example :

.AB_Melody_On(3,-5,3,1) && Turn on the speaker with song no.3 which is on Gateway 3 、 Port 1 and address is 5 .

.AB_Melody_On(3,17,0) && Turn off the speak which is on Gateway 3 、 Port 2 and address is 17.

.AB_Melody_On(3,-252,10,1) && Turn on all speaker which are on Gateway 3 、 Port 1

Function AB_Melody_Volume(INTEGER Gateway_ID, INTEGER Node_Addr, BYTE Volume)

Description : Rearrange the volume on AT510M.

Parameters :

Node_Addr: tag address, <0:port1, >0:port 2

Volume: 1(Min)~16(Max)

Example :

.AB_Melody_Volume(3, -5,5) && Rearrange the volume of speaker to level 5 which is on Gateway 3 、 Port 1

(9) Send message to 6-digit, 2 separated windows pick tag(AT506-2W-33)

Function AB_2W_LB_DspNum(INTEGER Gateway_ID, INTEGER Node_addr, LONG Case_unit, LONG Piece_unit, BYTE Dot, INTEGER Interval)

Description : Show numeral message on AT506-2W-33.

Parameters :

- Node_Addr: address, <0:port1, >0:port 2
- Case_unit : display data. (three digits)
- Piece_unit: display data.(three digits)
- Dot: digit points, bit1:1-st digit point, bit2:2-nd digit point
- Interval
 - 0: normal display
 - 1: blinking with default frequency
 - >0: set blinking frequency (by msec)
 - 2: turn off digit
 - 3: turn off digits and lamp

Return Value :

- 0-> means Gateway is normal.
- 1 -> means Gateway fails to work.

Example :

AB_2W_LB_DspNum(2,-8,3,12,0,0) && Send numeral "3" and "12" to 2 separated windows pick tag.

- 2: Gateway ID
- 8: Send to picking tag of address 8 on Port 1
- 3 : Number shown on the first window of picking tag
- 12: Number shown on the 2nd window of picking tag
- 0: Do not display decimal point
- 0: Do not twinkle.

AB_2W_LB_DspNum(4, 16,12,385,2,300) && Send numeral "12" and "38.5" to 2 separated windows picking tag and twinkle at a frequency.

- 4: Gateway ID
- 16: Address of picking tag on Port 2.
- "12": the column information of location
- 385: Number displayed on picking tag's panel.
- 2: Display decimal point on second digit.
- 300: Digits twinkle at a period 300 msec.

(10) Send message to 10-digit, 3 separated windows Alphanumerical pick tag(AT50A-3W-523)

Function AB_3W_CP_DspNum(INTEGER Gateway_ID, INTEGER Node_addr, STRING Lot, STRING Loc, LONG Qty, INTEGER Interval)

Description : Show data string and numeral message on AT50A-3W-523.

Parameters :

- Node_Addr: address, <0:port1, >0:port 2
- Lot : the batch information of the display data.(five digits)
- Loc : the location information of the display data.(two digits)
- Qty: display data.(three digits)
- Interval
 - 0: normal display
 - 1: blinking with default frequency
 - >0: set blinking frequency (by msec)
 - 2: turn off digit
 - 3: turn off digits and lamp

Return Value :

- 0-> means Gateway is normal.
- 1 -> means Gateway fails to work.

Example :

```
AB_3W_CP_DspNum(2,-8,"A0003","02",12,0)
2: Gateway ID
-8: Send to picking tag of address 8 on Port 1
"A0003" : batch information shown on the 1st window
"02" : location information shown on the 2nd window
12: Number shown on the 3rd window of picking tag
0: Do not twinkle.

AB_3W_CP_DspNum(4,16, "A0003","02",12,300)
4: Gateway ID
16: Address of picking tag on Port 2.
"A0003" : batch information shown on the 1st window
"02": location information shown on the 2nd window
12: Number shown on the 3rd window of picking tag
300: Digits twinkle at a period 300 msec.
```

(11) Send message to 10-digit, 3 separated windows pick tag (AT50A-3W-334K)

Function AB_3W334_LB_DspNum(INTEGER Gateway_ID, INTEGER Node_addr, LONG Case_Unit, LONG Pack_Unit, LONG Piece_Unit, BYTE Dot1, BYTE Dot2, INTEGER Interval)

Description : Show data string and numeral message on AT50A-3W-334K.

Parameters :

- Node_Addr: address, <0:port1, >0:port 2
- Case_unit : display data. (three digits)
- Pack_unit: display data.(three digits)
- Piece_unit: display data.(four digits)
- Dot1: digit points, bit1:1-st digit point, bit2:2-nd digit point
- Dot2: digit points, bit1:1-st digit point, bit2:2-nd digit point
- Interval
 - 0: normal display
 - 1: blinking with default frequency
 - >0: set blinking frequency (by msec)
 - 2: turn off digit
 - 3: turn off digits and lamp

Return Value :

- 0-> means Gateway is normal.
- 1 -> means Gateway fails to work.

Example :

AB_3W334_LB_DspNum(2,-8,3,4,12,0,0,0)

- 2: Gateway ID
- 8: Send to picking tag of address 8 on Port 1
- 3 : Number shown on the first window of picking tag
- 4 : Number shown on the 2nd window of picking tag
- 12: Number shown on the 3rd window of picking tag
- 0: Do not display decimal point
- 0: Do not display decimal point
- 0: Do not twinkle.

AB_3W334_LB_DspNum(4,16, 1,2,385,2,0,300) && Send numeral "1" ,"2" and "38.5" to 3 separated windows picking tag and twinkle at a frequency.

- 4: Gateway ID
- 16: Address of picking tag on Port 2.
- 1 : Number shown on the first window of picking tag
- 2 : Number shown on the 2nd window of picking tag
- 385: Number shown on the 3rd window of picking tag
- 2: Display decimal point on second digit.
- 0: Do not display decimal point
- 300: Digits twinkle at a period 300 msec.

(12) Send message to 8-digit, 2 separated windows pick tag(AT508-2W-44K)

Function AB_2W44_LB_DspNum(INTEGER Gateway_ID, INTEGER Node_addr, LONG Case_unit, LONG Piece_unit, BYTE Dot, INTEGER Interval)

Description : Show numeral message on AT506-2W-33.

Parameters :

Node_Addr: address, <0:port1, >0:port 2
Case_unit : display data. (four digits)
Piece_unit: display data.(four digits)
Dot: digit points, bit1:1-st digit point, bit2:2-nd digit point
Interval
0: normal display
-1: blinking with default frequency
>0: set blinking frequency (by msec)
-2: turn off digit
-3: turn off digits and lamp

Return Value :

0-> means Gateway is normal.
-1 -> means Gateway fails to work.

Example :

AB_2W44_LB_DspNum(2,-8,3,12,0,0) && Send numeral "3" and "12" to 2 separated windows pick tag.

2: Gateway ID
-8: Send to picking tag of address 8 on Port 1
3 : Number shown on the first window of picking tag
12: Number shown on the 2nd window of picking tag
0: Do not display decimal point
0: Do not twinkle.

AB_2W44_LB_DspNum(4, 16,12,385,2,300) && Send numeral "12" and "38.5" to 2 separated windows picking tag and twinkle at a frequency.

4: Gateway ID
16: Address of picking tag on Port 2.
"12": the column information of location
385: Number displayed on picking tag's panel.
2: Display decimal point on second digit.
300: Digits twinkle at a period 300 msec.

(13) Send message to 6-digit, 2 separated windows pick tag(AT506-2W-24)

Function AB_2W24_LB_DspNum(INTEGER Gateway_ID, INTEGER Node_addr, STRING Lot, LONG Piece_unit, BYTE Dot, INTEGER Interval)

Description : Show numeral message on AT506-2W-33.

Parameters :

Node_Addr: address, <0:port1, >0:port 2
 Loc : the location information of the display data.(two digits)
 Qty: display data.(four digits)
 Dot: digit points, bit1:1-st digit point, bit2:2-nd digit point
 Interval
 0: normal display
 -1: blinking with default frequency
 >0: set blinking frequency (by msec)
 -2: turn off digit
 -3: turn off digits and lamp

Return Value :

0-> means Gateway is normal.
 -1 -> means Gateway fails to work.

Example :

AB_2W24_LB_DspNum(2,-8,"3",12,0,0)

2: Gateway ID
 -8: Send to picking tag of address 8 on Port 1
 "3" : location information shown on the first window of picking tag
 12: Number shown on the 2nd window of picking tag
 0: Do not display decimal point
 0: Do not twinkle.

AB_2W24_LB_DspNum(4, 16,"12",385,2,300)

4: Gateway ID
 16: Address of picking tag on Port 2.
 "12": the column information of location
 385: Number displayed on picking tag's panel.
 2: Display decimal point on second digit.
 300: Digits twinkle at a period 300 msec.

(14) Send message to 3-digit, 4-illuminant button pick tag (AT503-4K)

Function AB_4K_DspNum(INTEGER Gateway_ID, INTEGER Node_addr, LONG Qty, BYTE Buf_Idx, BYTE Auto_Rtv, BYTE LED_Sta, BYTE LED_Sta2, BYTE Dot)

Description : Show numerical message on AT503-4K by specific illuminant button.

Parameters :

Node_Addr: address, <0:port1, >0:port 2

Qty: display data.(3 digits)

Buf_Idx: 1 ~ 4 , Small illuminant button index, 1: 1st button(RED), 2: 2nd button(GREEN), 3: 3rd button(BLUE), 4: 4th button (YELLOW)

Auto_Rtv: Auto-retrivel next data from buffer, = 1 : Auto, = 0 :None.

LED_Sta: Active light's status. 1 ~5 (1: solid on, 2~5:Blink) , **0 = delete specific buffer**

LED_Sta2: Standby light's status. 1 ~ 4 (1: solid on, 2~4:Blink)

Dot: digit points, bit1:1-st digit point, bit2:2-nd digit point

Return Value :

0-> means Gateway is normal.

-1 -> means Gateway fails to work.

Example :

AB_4K_DspNum(2,-8,333,**3**,0,5,1,0) // send message to 3rd button(BLUE), standby is solid on, active is blinking.

2: Gateway ID

-8: Send to picking tag of address 8 on Port 1

333 : Number displayed on picking tag's panel

3 : put data into Buffer Index 3, BLUE button.

0: Get data by pushing illuminant button, no display value firstly.

5: active light is blinking.

1: standby lights are solid on

0: do not display decimal point.

AB_4K_DspNum(2,-8,111,**1**,0,4,1,0) // sending 111 to 1st button, standby is solid on , active is blinking.

AB_4K_DspNum(2,-8,222,**2**,0,4,1,0) // sending 222 to 2nd button, standby is solid on , active is blinking.

AB_4K_DspNum(2,-8,333,**3**,0,4,1,0) // sending 333 to 3rd button, standby is solid on , active is blinking.

AB_4K_DspNum(2,-8,444,**4**,0,4,1,0) // sending 444 to 4th button, standby is solid on , active is blinking.

AB_4K_DspNum(2,-8,111,**3**,0,**0**,1,0) // delete 3rd button data, off the light

PS : by AB_LB_DSPNUM(gateway, node, 0,0,-3) to clear all the buffer.

Receiving message

Please refer the receiving function [AB_Tag_RcvMsg](#) (ByRef gatewayID As Long, ByRef tag_addr As Long, ByRef subcmd As Long, ByRef msg_type As Long, ByRef data As Byte, ByRef data_cnt As Long)

SubCmd = 48 (30H).

Msg_type : This parameter is the data buffer index (illuminant button index). 0 is 1st button(RED), 1 is 2nd button(GREEN), 2 is 3rd button(BLUE), and 3 is 4th button(YELLOW).

Function AB_4K_DspLED(INTEGER Gateway_ID, INTEGER Node_addr, BYTE LED_Idx, BYTE BeConfirm, BYTE LED_Sta)

Description : Only control the state of 4-illuminant button of AT503-4K.

Parameters :

Node_Addr: address, <0:port1, >0:port 2

LED_Idx: 1 ~ 4 , Small illuminant button index, 1: 1st button(RED), 2: 2nd button(GREEN), 3: 3rd button(BLUE), 4: 4th button (YELLOW)

BeConfirm: Define Lightable button's Status, = 1 : Can be confirmed, = 0 : not

LED_Sta: Active light's status. 0 ~5 (0: off, 1: solid on, 2~5:Blink)

Return Value :

0-> means Gateway is normal.

-1 -> means Gateway fails to work.

Example :

```
AB_4K_DspLED(2,-8,4,1,5) // turn on 4th button(YELLOW) , let it can be confirmed and by blinking status.
2: Gateway ID
-8: Send to picking tag of address 8 on Port 1
4 : put data intoBuffer Index 4, YELLOW button.
1: the lightable button's cannot be confirmed
5: blinking
```

```
AB_4K_DspLED(2,-1,1,1,1) // turn on 1st button(RED) , let it can be confirmed and by solid on.
AB_4K_DspLED(2,-1,2,1,4) // turn on 2nd button(GREEN) , let it can be confirmed and by blinking
AB_4K_DspLED(2,-1,3,1,1) // turn on 3rd button(BLUE) , let it can be confirmed and by solid on.
AB_4K_DspLED(2,-1,4,0,4) // turn on 4th button(YELLOW) , let it can NOT be confirmed and by blinking.
```

PS : by AB_LB_DSPNUM(gateway, node, 0,0,-3) to clear all the buffer.

Receiving message

Please refer the receiving function `AB_Tag_RcvMsg (ByRef gatewayID As Long, ByRef tag_addr As Long, ByRef subcmd As Long, ByRef msg_type As Long, ByRef data As Byte, ByRef data_cnt As Long)`

SubCmd = 49 (31H).

Msg_type : This parameter is the data buffer index (illuminant button index). 0 is 1st button(RED), 1 is 2nd button(GREEN), 2 is 3rd button(BLUE), and 3 is 4th button(YELLOW).

Function AB_4K_Demo(INTEGER Gateway_ID, INTEGER Node_addr, BYTE Buf_Idx, STRING DISP_STR , BYTE LED_Sta, BYTE Swap_Time, BYTE Dot)

Description : Auto display messages from buffer sequentially on AT503-4K.

Parameters :

Node_Addr: address, <0:port1, >0:port 2
 Buf_Idx: 1 ~ 4 , data string index
 DISP_STR: display data.
 LED_Sta: Active light's status. 0 ~5 (0: off, 1: solid on, 2~5:Blink)
 Swap_Time: the time of data display on pick tag's panel. (1 ~ 4)
 Dot: digit points, bit1:1-st digit point, bit2:2-nd digit point

Return Value :

0-> means Gateway is normal.
 -1 -> means Gateway fails to work.

Example :

```
AB_4K_Demo(2,-2,1," ccc",1,1,0)
2: Gateway ID
-2: Send to picking tag of address 2 on Port 1
"ccc" : String displayed on picking tag's panel
1 : put data intoBuffer Index 1
1: stay time
0: Do not display decimal point.
```

(15) DI/DO operation. (Please refer the AT520 IO communication protocol.doc)

The Digit I/O module all reserve 16 channels in software capacity.

Function AB_DIO_ReadIoStatus (INTEGER Gateway_ID, INTEGER Node_addr,) As Integer

Description : Request to read the I/O status of the specified DI/DO.

Parameters :

Node_Addr: address, <0:port1, >0:port 2

Return Value :

- 0 -> OK.
- 1 ->: Gateway_ID not exist
- 2 ->: TCP not created
- 3 ->: TCP not open
- 4 ->: TCP send error

Example :

AB_DIO_ReadIoStatus(2,-5) ;Request to read the status of DI/DO (node=5) been connected to gateway 2, port 1.

Remark : Since AT520 has both DI and DO, so when send AB_DIO_ReadIoStatus to AT520, TCP/IP controller will return two messages to report AT520's DI and DO status individually.

Function AB_DIO_SetDO (INTEGER Gateway_ID, INTEGER Node_addr, INTEGER channel, INTEGER status) As Integer

Description : Set the status of the specified DO.

Parameters :

Node_Addr: address, <0:port1, >0:port 2

channel: specify a channel to be updated.

1 ~ 16, only one channel is updated,

0:all channels are updated

status: 1:on, 0:Off.

Return Value :

- 0 -> OK.
- 1 ->: Gateway_ID not exist
- 2 ->: TCP not created
- 3 ->: TCP not open
- 4 ->: TCP send error

Example :

AB_DIO_SetDo(2,-5,1,1) ;Set the channel 1 of DO (node=5) been connected to gateway 2, port 1 to on.

Function AB_DIO_SetDiRspMode (INTEGER Gateway_ID, INTEGER Node_addr, INTEGER mode) As Integer

Description : Set auto report mode of the specified DI. If the DI is set to auto report mode, it will report DI status whenever DI status is changed *(The default mode is auto report for AT520)*

Parameters :

Node_Addr: address, <0:port1, >0:port 2

mode: 0: none auto report mode

1: auto report mode

Return Value :

- 0 -> OK.
- 1 ->: Gateway_ID not exist
- 2 ->: TCP not created
- 3 ->: TCP not open
- 4 ->: TCP send error

Example :

AB_DIO_SetDiRspMode(2,-1,1) ;Set all DI been connected to gateway 2, port 1 to auto report mode.

Receive DI/DO status

After request the status of specified DI/DO, Using function: AB_Tag_RcvMsg () to receive I/O status from DI/DO. And the receiving message is as following. Please refer ***“AT520 IO communicaiotn protocol.doc”***

(1) Digit I/O for type DI & DO

```
subcmd = 3CH
data() = [0]:Reserved, 00H
          [1]: DI/O Type → '0' (30H):DI, '1' (31H):DO
          [2]: Channel 5-8 Status
          [3]: Channel 1-4 Status
          [4]: Channel 13-16 Status
          [5]: Channel 9-12 Status
```

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

```
data [2] = 0x39('9')
data [3] = 0x46('F')
data [4] = 0x30('0')
data [5] = 0x43('C')
```

For AT520 : there is only Channel 1 to 4. data[2], data[4] & data[5] is always 30H for AT520.

Example: ch16 ch9.....ch1
 0000 0000 0000 0100 ---→ 1: means ON, 0: means OFF
 ASCII '0' '0' '0' '4'
 means -----> channel : only 4 is on , the other channels are off

```
data [2] = 0x30('0')
data [3] = 0x34('4')
data [4] = 0x30('0')
data [5] = 0x30('0')
```

(2) Report for DI-AUTO status

```
subcmd = 3CH
data() = [0]:Reserved, 00H
          [1]:I/O Type → '4' (34H)
          [2]: Current status for channel 5-8
          [3]: Current status for channel 1-4
          [4]: Current status for channel 13-16
          [5]: Current status for channel 9-12
          [6]: Last time status for channel 5-8
          [7]: Last time status for channel 1-4
          [8]: Last time status for channel 13-16
          [9]: Last time status for channel 9-12
```


(16) Cable-less picking tag set up node address automatically.

The new controller for cable-less picking tags provide two kind of mode for configuring the node's address. One is called "checking mode", the other is called "setting mode".

"Checking mode" is mainly to know the node address of some node without LED display to show, such as bay indicator (yellow light), completion indicator (green light). By using this mode, we can show these modules' node address through the AT505 which is on the same port.

"Setting mode" is mainly to setup an assigned node to one module. It can change the node address.

The following will describe the operation procedure of these two modes:

<Checking mode>

1. Activate one port entering the "checking mode". By Function "AB_CLTAG_DspAddr(1, 0)" for port 1 and "AB_CLTAG_DspAddr(2, 0)" for port 2.
2. When one port enter "checking mode", initially all the modules will display exchangeably two screen message on its LED display. The message is its node address, and the two screen message will be [001], 001, or [002], 002, or [003], 003, 001, 002, 003 are the module's node address,
3. You can push one tag's button that you want to know its node address, especial the tag without LED display.
4. Then it will return one CCB socket out, and on the 2nd byte of receiving DATA will be 16H.
5. From the return CCB socket, you can know its sub node address.
6. You can send this received node address by AB_CLTAG_SetAddr(Received Node, 0) to active all module show that node address on its LED display. And it will also be two screen message exchangeably showing, such as [001] , 006 or [002] , 006, or [003] , 006. 006 is that module's node address that you would like to know.

<Setting mode>

1. Activate one port entering the "setting mode". By Function "AB_CLTAG_DspAddr(1, 1)" for port 1 and "AB_CLTAG_DspAddr(2, 1)" for port 2.
2. When one port enter "setting mode", initially all the modules will display exchangeably two screen message on its LED display. The message is its node address, and the two screen message will be [001], -001-, or [002], -002-, or [003], -003-, 001, 002, 003 are the module's node address,
3. You can send one assigned address to that port by AB_CLTAG_SetAddr(assigned Node, 1) to active all module show that node address on its LED display. And it will also be two screen message exchangeably showing, such as [001] , -006- or [002] , -006-, or [003] , -006-. 006 is that assigned node address that you would like to setup.
4. When you push some tag's button, then this assigned node address will be setup into that module's EEPROM to change its original node address.

Function AB_CLTAG_DspAddr(INTEGER Gateway_ID, INTEGER Port, INTEGER mode)

Description : Initially active one port to enter "checking mode " or "setting mode"

Parameters :

Port : define which port of TCP/IP control

mode

=0: checking mode

=1: setting mode

Example :

AB_CLTAG_DspAddr(1,1, 0) && To active Gateway 1, Port 1.to enter node address "checking mode"

Function AB_CLTAG_SetAddr(INTEGER Gateway_ID, INTEGER Node, INTEGER mode)

Description :

Parameters :

mode

=0: checking mode

=1: setting mode

node

In "checking mode", it will be the node address that you would like to know, normally it is the sub-node

address when receiving from the pushed device.

In "setting mode", it will be the assigned address that you would like to setup.

Example :

```
= AB_CLTAG_SetAddr(1,10,1)
```

Under device default mode and enable the up/down count buttons. Both button can be used to adjust the quantity, but the maximum number can be increased is always equal to the original sending value, no more than it. Intend to achieve to adjust the quantity number can be more than the sending value, then it would need to enable the cycle edit function by AB_TAG_CycleEdit as below. When device's cycle edit function is enabled, then its LED light will keep blinking to distinguish.

(new)

Function AB_TAG_CycleEdit(INTEGER Gateway_ID, INTEGER Node_Addr, BYTE on_off)

Description : Enable up-count/down-count button can perform cycle-counting behavior. ***This function must be under the condition which up-count button is enabled. In other words, the bit 2 of parameter mode_type in function AB_TAG_Mode() must be enabled. Please refer the function AB_TAG_Mode() in page 19.***

Parameters :

Node_Addr: tag address, <0:port1, >0:port 2

On_off: 1: Enable, 0:Disable

Example :

```
AB_TAG_Mode(3, -5,117) && enable the edit mode/up-count and down-count button
```

```
.AB_TAG_CycleEdit(3, -5,1) && Enable the digit cycle-counting function on Gateway 3 ` Port 1 and address is 5 .
```

PS1: AB_TAG_CycleEdit() only can be applied in AT505, F/W version 1.2 and above 1.2 or AT7XX series

PS2: The enabled status can not be store into EEPROM, so when device is reset, this configuration will return to default.

PS3: Please note the return sub-command is 36(24H) under this mode.