Serial Number :_____
Option:_____

OPERATOR'S MANUAL
MODEL AP15/AP15-1
PC-ALARM PANEL

Micro Seven, Inc.
P.O. Box 5597
Beaverton, OR
97006
U.S.A.
phone: 503-693-6982
fax: 503-693-9742
www.microseveninc.com

All requests for repairs should be directed to the factory.

This instrument is warranted against defective workmanship and materials for a period of six months. There will be no warranty when the instrument is misused, or when the factory seal on the instrument is broken.

Printed in U.S.A.

SECTION 1

INTRODUCTION

Micro Seven Model AP15, PC-Alarm Panel, transmits Ademco Contact-ID alarm messages to alarm receivers by starting commands sent at RS232C interface in PC. PC sends ASCII equivalent of alarm message to AP15 without check-sum. PC also sends telephone number to AP15. Since the alarm messages, telephone numbers, and AP15 Control Register for storing programming parameters are stored in non-volatile memory in AP15, a single ASCII command at RS232C interface starts alarm reporting process. After AP15 dials a telephone number for a receiver, the receiver answers the call and transmits the handshake tones. Then AP15 transmits alarm message in DTMF tone. If the receiver receives the alarm message correctly, it transmits kiss-off tone. Then AP15 hangs up the call, and it sends status messages to PC. If the alarm reporting is not successful, AP15 transmits a different status message to PC. Transmission of alarm messages is repeated for four times if AP15 does not receive kiss-off tones. The whole sequence, which involves re-dialing of telephone number, may be repeated for three times. AP15 contains audio monitor internally to hear dialing process, handshake signals, DTMF tones for alarm messages, and kiss-off tones. AC/DC adapter and interfacing cable between PC and AP15 is provided. 220V AC/DC adapter is optional. There are several AP15 registers that are programmable by using RS232C interface.
AP15 is compatible with the Ademco ® Contact ID Protocol for Alarm System Communications.

AP15-1 is compatible with AP15 mechanically and electrically. Ap15-1 additionally contains email capability that sends emails automatically on Internet when alarm messages are sent over telephone lines. Dynamic line library software is provided to generate Ademco digital alarm messages over telephone lines as well as generating decoded alarm messages over Internet.

SECTION2

SPECIFICATIONS

DTMF dialing signal power: 0 dBm total for two frequency components, 100 ms on, 100ms off

Pulse dialing: Break period: 60 ms, make period: 40 ms, inter-digit wait time: 1 sec

Alarm message signal power: -12 dBm (factory default), 0 dBm with Control Register bit 7 on

Alarm message DTMF tone timing: burst ON time: 50 ms (factory default), burst OFF time: 50 ms (factory default), programmable by DTMF register

Handshake tone and kiss-off tone signal detection: -26dBm minimum (factory default), -38dBm with x4 gain by Control Register bit 6 off

Front panel controls:

Power switch
Power-On Indicator: green LED display

Rear panel controls:

DC power input, DC12V, 800 ma
RS232C interface connector, 9-pin D-sub, female
RJ11 connector

Audio monitor: monitoring receiving signal (handshake tone and kiss-off tone) and transmitted signal (dialing signal and alarm message), 0.1 watts maximum with internal speaker

RS232C interface: 9600 baud nominal (may be changed to 1200 baud by control register 2), no parity, and 8-bit character

There are no RS232C data communication handshakes available for AP15 and PC. The hardware and software handshakes involving with CTS, RTS, XON, and XOFF are not required for AP15 because AP15 does not require any delay between character inputs and AP15 transmits few characters for messages normally. However, if the AP15 Control Register bit 5 is cleared to enable nonvolatile memory operation for storing telephone numbers and alarm messages, minimum 10 ms delay is required between characters inputs for telephone number and alarm message entry.

Telephone number entry: D<telephone number-maximum fifteen digits><CR>
for example D5035551212<CR> for dialing 5035551212. Note <CR> is a carriage return. It is stored in EEROM (non-volatile memory) in AP15.
Warning: When the non-volatile memory is selected by clearing the Control Register bit 5, time delay of minimum 10 ms is required between characters in a command.

Alarm message entry: S<fifteen digit long alarm message><CR>
i.e. S123418313101015<CR>
Note: AP15 prepares check sums, so do not enter sixteen digits. When sixteen digit-long alarm

message including check sum is entered, checksum, which is prepared by AP15, will become a wrong one. It is stored in EEROM (non-volatile memory) in AP15 when the Control Register bit 5 is off.

Warning: When the non-volatile memory is selected by clearing the Control Register bit 5, time delay of minimum 10 ms is required between characters in a command.

Start alarm reporting: G
Start alarm reporting process by dialing the telephone number, receive handshake tones, transmit alarm messages, receive kiss-off tones and hang up the line if AP15 Control Register is programmed.
A carriage return is not needed.

Hang-up command or Abort command: A
Turn off off-hook relay in AP15 or abort alarm reporting process. A carriage return is not needed.

Repeat alarm message command: B
It transmits alarm message when AP15 is off-hook condition. A carriage return is not needed.

Reading telephone-number command: T
AP15 transmits programmed telephone numbers to PC. A carriage return is not needed.

Reading alarm-message command: U
AP15 transmits programmed alarm message in sixteen digit-long including check sum to PC. A carriage return is not needed.

Status messages from AP15 to PC

DIALED # BUSY
MESSAGE SUCCESS
MESSAGE FAILED
CALL FAILED

Telephone interface: use of mechanical relay, optical-coupler, and telephone coupling transformer to obtain high voltage isolation and high on-hook impedance

Reference: Digital Communication Standard-SIA DC-05-1999.09, Ademco Contact ID Protocol for Alarm System Communications

AP15 REGISTERS: Caution: We recommend factory default values for all AP15 registers.
AP15 registers are written by a command format of "M m1 m2 n1 n2" for EEROM/data memory operation or "W m1 m2 n1 n2" for data memory operations. EEROM/data memory operation saves copy of data memory in EEROM for restoring data memory contents on power-up. The “m1 m2” is address, and “n1 n2” is the data. There is a character “C” is returned to PC after each EEROM operation. The "W m1 m2 n1 n2" operation writes new value in data memory only and not in EEROM. The content of EEROM is not changed.

1. DTMF Register
DTMF Register operation writes new value of on/off timing of DTMF generation of alarm messages.

Each increment is 10 ms, and the factory default condition is 5 for 50ms on and off timing. Ademco specification limits 50ms minimum and 60 ms maximum. The command format is:
MA5 n1 n2 for EEROM/data memory operation
Or
WA5 n1 n2 for data memory operation.
Where n1 and n2 forms increment of 10 ms. For example, MA506 or WA506 selects 60 ms on/off times.
The content of DTMF Register may be done by "RA5", and AP15 outputs "06".

2. Tone Detect register

Tone Detect register contains threshold value for detecting handshake and kiss-off tones for poor line conditions or for long-distance telephone calls.
The threshold value for long-distance telephone line condition when Control Register bit 6 is off is programmable by the following command:
MA7 n1 n2 for EEROM/data memory operation, and WA7 n1 n2 for data memory operation, and when smaller hexadecimal value provides lower threshold value and higher sensitivity. The factory default value is 20 hexadecimal value. The content of Tone Detect register may be read by RA7.

3. Control Register

It is written by the following command:

MAC n1 n2, where n1 n2 form one byte in hexadecimal notation or WAC n1 n2.

Note: "MACB0" programs the factory default conditions.

Status of the control register is performed by entering "RAC", and AP15 generates "B0".

Bit 7: "1" selects higher signal power output (0 dBm) for alarm message, and "0" selects lower signal power output (-12 dBm). The factory default is "0" to select –12dBm.

Bit 6: "1" disables input signal amplification (x4) for detecting weak handshake and kiss-off tones, and "0" enables input signal amplification. The factory default is "1" for selecting normal input signal .

Bit 5: "1" disables storing telephone number in non-volatile memory for telephone number input command, and "0" is for storing telephone number in non-volatile memory. The factory default is "0" for enabling non-volatile memory operation.

Bit 4: "1" disables detection of busy tone, and "0" enables detection of busy tone. Note; The busy-tone detection when enabled is between time period for 2.5 seconds after the end of dialing. The factory is "1" for disabling busy-tone detection.

Bit 3: "1" selects rotary-pulse dialing, and "0" selects DTMF dialing of telephone numbers. The factory default is "0" for DTMF dialing.

Bit 2: "1" disables re-dialing of telephone numbers when "MESSAGE FAILED" is generated. "0" enables re-dialing of telephone number for maximum times. The factory default is "0" for re-dialing telephone numbers when "MESSAGE FAILED" is displayed.

Bit 1: "1" keeps telephone line off-hook after successful delivery of alarm messages. "0" enables hanging up the line after successful of alarm messages. The factory default is "0" for hanging up the line.

Bit 0: "1" disables re-dialing telephone numbers when message "CALL FAILED" is displayed. "0" enables re-dialing telephone numbers. The factory default is "0" for re-dialing telephone numbers.

4. Redial Register
The number of times that AP15 repeats dialing when there is a “CALL FAILED” or “DIALED # BUSY” message is output to PC is written by the following command:
MA3 n1 n2, where n1 n2 form one byte in hexadecimal notation or WAC n1 n2. The factory default is three times.

5. Message Length Register
The alarm message length for AP15 is normally sixteen digits as defined in SIA DC-05-1999.09, Ademco Contact ID Protocol for Alarm System Communications, but it may be changed for fewer length by the following command:
MF0 n1 n2

For example, "MF010" programs the factory-default condition. The command "MF00F" programs fifteen-digit alarm message.

The Message Length Register is in EEROM.

6. AP15 control register 2 (address location AE, Factory default condition is obtained by MAE05.**)**

Bit 0 of AP15 control register 2 enables CTS handshake between AP15 and PC. In null condition, CTS is asserted to allow PC to transmit characters. During alarm message transmission, CTS is de-asserted to stop transmission of characters to AP15. As soon as transmission of alarm message is completed or aborted because of error conditions, CTS is asserted. The factory default condition is CTS-enabled.

Bit 1 of AP15 control register 2 selects baud-rate. When it is cleared, 9600 baud is selected. When it is set, 1200 baud is selected. Factory default is cleared.
Caution: changing this bit affects baud-rate.

Bit 2 of AP15 control register 2 enables clearing CTS signal when “MESSAGE SUCCESS” or telephone line is disconnected or AP15 goes on-hook. When the bit 2 is off, CTS signal is cleared when the telephone line is disconnected or AP15 goes on-hook. Factory default is set.

SECTION3

OPERATION

3.1 First-time operation

Connect AC adapter to an appropriate AC power source, and connect power plug to AP15. Push power switch on AP15 front panel. You will see green power indicator turned on. Connect RS232C interface cable between PC and AP15. Connect telephone cord to RJ11 connector on rear-panel of AP15, and connect the other end to an inside line of PBX. In your PC, go to Hyper-terminal with communication set in 9600 baud. Local echoing of input character may be desirable to view key board entries.

3.2 Programming telephone number

If a telephone number for dialing alarm receiver is 9-503-555-1212, input the following:
D9,5035551212<CR>
Note: “CR” is a return or carriage return on PC-key board. And a comma, “,” , provides 2.5 second delay that may be necessary to reach outside line in PBX equipment after dialing "9".

To verify the above telephone number entry, enter the following:
T
Then AP15 outputs 9,5035551212.

You will also see the front-panel LED, “DATACOMM” is turned on momentarily when each character is output to AP15. Telephone numbers are stored in EEROM (non-volatile memory) and data memory when Control Register bit 5 is off as a factory default condition.

When Control Register bit 5 is on, programmed telephone numbers and alarm messages are stored only in data memory and not in EEROM. And verification of telephone numbers is done by the same command “T”.
Note: During power-on initialization of AP15, data in EEROM is brought to data memory.

3.3 Programming alarm message

If an alarm message is sixteen digit-long, 123418113101015 and a checksum, input the following:

S123418113101015<CR>
Note: “CR” is a return or carriage return on PC-key board. Do not enter a checksum. AP15 prepares check sum by itself.
To verify alarm message entry, input the following:
U
Then AP15 outputs 1234181131010158. Note: 8 at the end is a check sum that AP15 has produced.

If sixteen character alarm command is entered, i.e., S1234181131010158<CR>, AP15 prepares a wrong checksum for an alarm message. Generating wrong checksum may be useful in testing AP15 for repeating generation of alarm message for four times.

Alarm messages are stored in EEROM (non-volatile memory) and data memory when Control Register bit 5 is off as a factory default condition.

When Control Register bit 5 is on, telephone numbers and alarm messages are stored only in data memory and not in EEROM. And verification of alarm message is done by the same command “U”.
Note: During power-on initialization of AP15, data in EEROM is brought to data memory.

3.4 Start alarm message transmission

To start sending alarm message, input the following:

G

You will hear a relay is turned on, front-panel LED is turned on, and dialing telephone numbers in DTMF tones. If the dialed number is alarm receiver, handshake tones with 1400 Hz and 2300 Hz is heard. Then AP15 transmits alarm message very quickly. After AP15 completes transmission of alarm message, the alarm receiver transmits kiss-off tone (1400Hz) for short period. After the end of kiss-off tone, AP15 hangs up the line. The front-panel LED is turned off.

The following message is output to PC:
MESSAGE SUCCESS

3.5 Testing called-line busy

When dialed number is busy, the following message is displayed:
DIALED # BUSY
Note: Dialed-line busy condition is only checked during 2.5 seconds after the end of dialing because of handshake tone must be checked.
Note: The busy-tone checking may be disabled when the Control Register bit 4 on.

3.6 Testing no loop current, no dial tone, or dialing wrong number for an alarm receiver

When there is no loop current, no dial tone received or dialing wrong number, the following message is output:
CALL FAILED
When re-dialing is enabled with Control Register bit 2 off, AP15 tries three times as a factory default condition.

3.7 Poor telephone line or long-distance telephone calls

Under poor telephone line or long-distance calls, the following sequence may be necessary:

3.7.1 Adjustment of DTMF signal power for alarm messages

Signal power of alarm message is increased from normal –12 dBm to 0 dBm when the Control Register bit 7 is on. The factory default condition is –12 dBm for signal power with the Control Register bit 7 off.

3. 7.2 Sensitivity adjustment for detecting handshake and kiss-off tone

When handshake and kiss-off tones are difficult for AP15 to recognize because of poor telephone line condition or long-distance call, tone input is amplified by four times when the Control Register bit 6 is off.

When the Control Register bit 6 is off, tone detection threshold value is determined by value in Tone Detect Register. The factory default of the Tone Detect register is hexadecimal 20. Smaller value makes detection much sensitive.

3.8 Multiple alarm message transmission

When several types alarm messages need to be transmitted without hanging up the line, set bit 1 on of the Control Register. After successful delivery of alarm message, it remains off-hook without hanging up the line. Also setting bit 5 of Control Register prevents alarm message stored into EEROM area, then input the following new alarm message:
S123418313101015<CR>
Type the following command to deliver the above message:
B
When the alarm message delivery is successful, “MESSAGE SUCCESS” is output from AP15 to PC.
To hang up the call, input the following:
A

3.9 testing MESSAGE FAILED operation

An alarm message with incorrect checksum is prepared by entering extra character as follows:
S1234183131310158<CR>
G
AP15 dials a telephone number stored in AP15, and sends the above alarm message. But an alarm receiver does not generate kiss-off tone because the alarm message contains an incorrect checksum. AP15 sends four times because it does not receive kiss-off tones. The AP15 sends the following:
MESSAGE FAILED
AP15 hangs up the call, and re-dials the telephone number four times, but AP15 will not receive kiss-off tone because the alarm message contains an incorrect checksum. AP15 sends "MESSAGE FAILED" messages three times.

When the Control register bit 2 is on, AP15 does not redial the telephone number. AP15 sends "MESSAGE FAILED" message once.

SECTION 4

EXTRA FEATURE FOR MODEL AP15

Warning: the alarm-panel polling command is not standard feature for AP15, and it is not guarantied to work.

4.1 Alarm-panel polling command

When AP15 receives “C”, it performs the “double-dialing” operation, which goes off-hook and dials a number. Then it hangs up, waits for few seconds, goes off-hook and dials the same number. If the called line transmits 2200Hz-answer tone to AP15, AP15 transmits “POLLING SUCCESS” to PC. When AP15 does not receive the 2200Hz-answer tone, it transmits “CALL FAILED”. AP15 does not re-dial the number.

The time between the end of the first dialing and hanging-up is programmed by the following command:
MAA(n1)(n2)
Note: n1 and n2 forms one byte in hexadecimal notation, and each count is 2.5 seconds. The minimum notation is 01. For example, MAA03 selects five seconds delay. The factory default value of n1 n2 is 07, and the delay is fifteen seconds. There is a character output of “C” from AP15.

The time between hanging-up the phone line and second dialing is programmed by the following command:
MA4(n1)(n2)
Note: n1 and n2 forms one byte in hexadecimal notation, and each count is 2.5 seconds. The minimum notation is 01. For example, MA403 selects five seconds delay. The factory default value of n1 n2 is 04, and the delay is seven seconds. There is a character output of “C” from AP15.

SECTION 5

AP15-1

5.1 Software for AP15-1:

AP15 or AP15-1 may be operated by using ASCII commands that are described in the previous section including “G” for starting sending alarm messages and “A” for hanging up telephone lines, but there are no ASCII commands for sending emails automatically.

The provided Dynamic Link Library, ap15.dll, may be inserted in customers’ software in C++ or Visual Basic for dialing telephone numbers, sending Ademco-compatible digital alarm messages, and sending emails to any email addresses. The function in ap15.dll includes telephone number for dialing, digital alarm message, RS232 communication port number, local SMTP server name, to-addresses, from-address, sender name, subject field, and email messages.

5.2 Installation

Create a directory “ap15” in your computer. If you already have a working directory, use the same directory.

Open ap15 directory or your working directory.

Using a word processing software like Notepad, Create a file “liability” with no file extention which contains a word “agree” with no carriage return which you agree that Micro Seven, Inc. does not provide any liability for you using AP15-1 over PBX or public telephone systems and also for sending emails over Internet.

Copy ap15.dll into the same directory. Your application software must reside in the same directory.

5.3 Overhead code for ap15.dll

Using ap15.dll may require the following overhead code in C++ in your application:

```
HINSTANCE gLibap15 = NULL;

gLibap15=LoadLibrary("AP15.DLL");

typedef int (*AP15PP)(  char telnumber[] //telephone number
                      , char alarmmessage[] //alarm message for 15 digit long
                      , char commport[] //communication port number
                      , char szSmtpServerName[] //smtp server name on Internet
                      , char  szToAddr[] //to-email address
                      , char szFromAddr[] //from-email address
                      , char szUsername[] //smtp user account name
```

, char szPassword[] //password

, char szSender[] //sender name

, char szSubject[] //subject field

, char szBody[] //email message

, LPSTR bufpointer //output message

, int command

, int callingmode

);

AP15PP ap15;

ap15=(AP15PP)GetProcAddress(gLibap15,"ap15");

//note: , char szToAddr[]="" //for no-email generation

5.4 Output parameter and input parameters (arguments) for ap15.dll

The structure of ap15 is as follows:

int ap15(//return 0 for idle, 1 for successful alarm message delivery operation, 2 for failed operation
char telnumber[] //string for telephone number
, char alarmmessage[] //string for alarm message for 15 digit long
, char commport[] //RS232 communication port number
, char szSmtpServerName[] //local smtp server name, non-SSL type
, char szToAddr[] //to-email address, use commas "," to separate addresses for sending emails to multiple addresses,
//i.e. //anyone@anyplace.com,anyman@anyplace.com,---"
, char szFromAddr[] //from-email address
, char szUsername[] //smtp user account name
, char szPassword[] //smtp password
, char szSender[] //sender name
, char szSubject[] //subject field
, char szBody[] //email message
, LPSTR bufpointer //display buffer pointer
, int command //command character, C for hang-up, N for status word read, A for hanging up telephone line, S for all others
, int callingmode //0 for sending text in szBody[], 1 for sending decoded message of alarmmessage[].

5.4 Output parameter and input parameters (arguments) for ap15.dll

5.4.1 char telnumber[]

character string for ap15-1 to dial.

Example in C++ programming:

char telnumber[]="5035551212';

5.4.2 alarmmessage[]

15-digit long character string for digital alarm message. Do not include a checksum character.

Example in C++ programming:

*char alarmmessage[]="*123418113101015";

5.4.3 commport[]

RS232 communication port number in capital letters and column character.

Example in C++ programming:

char commport[]="COM1:";

5.4.4 szSmtpServerName[]

Local smtp server name and it requires non-SSL type server.

Example in C++ programming:

char szSmtpServerName[]*="smtp.localhost.com*";

5.4.5 szToAddr[]

To-email address, use commas to separate addresses for sending emails to unlimited number of addresses. Local smtp server name and it requires non-SSL type server.

Example in C++ programming:

char szToAddr[]="person1@anybusiness.com,person2@anycompany.com";

Caution: char szToAddr[]=""; //for no email generation

5.4.6 szFromAddr[]

From-email address.

Example in C++ programming:

char szFromAddr[]="person1@anybusiness1.com";

5.4.7 szUsername[]

Your smtp user account name

Example in C++ programming:

char szUsername[]="myname@myemail.com";

5.4.8 szPassword[]

Your smtp password.

Example in C++ programming:

char szPassword[]="anypassword";

5.4.9 szSender[]

Sender's name.

Example in C++ programming:

char szSender[]="John Smith <john.smith@smith.mmm>";

5.4.10 szSubject[]

Subject field.

Example in C++ programming:

char szSubject[]="PC-Alarm AP15-1 reporting";

5.4.11 szBody[]

Email message when the callingmode is 1.

Example in C++ programming:

char szBody[]="Reporting burglar at plant 1 zone 1";

5.4.12 *LPSTR bufpointer*

Memory pointer for displaying purposes (future expansion).

5.4.13 *int command*

Single ASCII character, C for hang-up telephone line after delivering digital alarm message, N for status word read, A for hanging up telephone line, and S for all others

5.4.13.1 *int command C or S*

Example in C++ programming:

The statement below dials the telephone number, send the alarmmessage and does not hang up the telephone line, and sends emails if szToAddr contains email address. Multiple alarm messages may be sent. If the character 'C' is replaced with 'S', it hangs up the telephone line after successful transmission of the alarm message.

Int c;

ap15(telnumber, alarmmessage, commport, szSmtpServerName, szToAddr, szFromAddr

, szUsername, szPassword, szSender, szSubject, szBody, bufpointer, 'C' ,callingmode);

5.4.13.2 *int command A*

The command below hangs up the telephone. No messages transmitted.

ap15(telnumber, alarmmessage, commport, szSmtpServerName, szToAddr, szFromAddr

, szUsername, szPassword, szSender, szSubject, szBody, bufpointer, 'A' ,callingmode);

5.4.14 callingmode

If integer callingmode contains 0, text in character string szBody[] is transmitted in the email. If the integer callingmode contains 1, decoded message of alarmmessage is transmitted in the email.

Create a directory called "customer" and create a text file "1234.txt" in the "customer" directory containing the following:

Abc corporation
1234 NE Broadway
Anycity, Anystate
Zip code
Phone number555-555-5555
Fax 555-555-5555
Email address
Name John Smith

Then C++ statement including the following codes produces the email messages as shown below:

5.5 AP15 status word and multiple alarm message transmission

The following ap15 command with command character of 'N' inquires ap15 status. "0" indicates idling/busy condition, "1" indicates that the alarm command is transmitted and received confirmation is received from distant alarm receiver, and "2" indicates that there is failure in transmission of alarm message. There are no transmissions of alarm messages nor emails with command character of 'N'.

Int x;

//send alarmmessage1 with no hanging up the phone line

ap15(telnumber, alarmmessage1, commport, szSmtpServerName, szToAddr, szFromAddr

, szUsername, szPassword, szSender, szSubject, szBody, bufpointer, 'C' ,callingmode);

//finding out whether ap15 completes transmissions

x=ap15(telnumber, alarmmessage, comport, szSmtpServerName, szToAddr, szFromAddr

, szUsername, szPassword, szSender, szSubject, szBody, bufpointer, 'N' ,callingmode);

if (x==1)

//send alarmmessage2 and hang up phone

x=ap15(telnumber, alarmmessage2, comport, szSmtpServerName, szToAddr, szFromAddr

, szUsername, szPassword, szSender, szSubject, szBody, bufpointer, 'S' ,callingmode);

Received email:

Sun May 10 20 48 14 2009

Alarm message 123418313101015
Acount number 1234
Abc corporation
1234 NE Broadway
Anycity, Anystate
Zip code
Phone number555-555-5555
Fax 555-555-5555
Email address
Name John Smith
Event qualifier Restoral
Event code 131 Perimeter-Burglar Alarm
User/Zone Number Zone 015
Partition number 01

Note: the profile information for customer with account 1234 is included after the account number. The column character is not displayed because of forbidden character in this email. In the first line, numbers between date and year are hour/minute/second without columns. Note that the Perimeter-Burglar Alarm is restored at zone 015, partition number 01.

5.6 Sample C++ programs of multiple alarm message transmissions without emails

//Ap15 dials the telephone number, transmit alarm message1, wait until AP15 for completion and transmit alarm message 2 and hang up the telephone.

```
HINSTANCE gLibap15 = NULL; //DLL overhead code, "ap15.dll" must reside in your directory.
gLibap15=LoadLibrary("AP15.DLL");
typedef int (*AP15PP)(  char telnumber[] //telephone number
                                , char alarmmessage[] //alarm message for 15 digit long
                                , char commport[] //communication port number
                                , char szSmtpServerName[] //smtp server name on Internet
                                , char  szToAddr[] //to-email address
                                , char szFromAddr[] //from-email address
                                , char szrawUsername[] //smtp user account name
                                , char szrawPassword[] //password
                                , char szSender[] //sender name
```

```
                                        , char szSubject[] //subject field
                                        , char szBody[] //email message
                                        , LPSTR bufpointer //output message
                                        , int command
                                        , int callingmode
                                        );

AP15PP ap15;
ap15=(AP15PP)GetProcAddress(gLibap15,"ap15");


int command1='C'; //continuation mode
szToAddr[0]=0; //select no email transmission
char telnumber[]="18005551212";
char alarmmessage1[]="123418313101015";
char commport[]="COM1:"; //RS232 communication port number 1

//function “ap15” returns immediately after sending telephone number, alarm message and
//and start dialing command “G”.
ap15(telnumber,alarmmessage1, commport,
                    0, //szSmtpServerName,
                    szToAddr,
                    0, //szFromAddr,
                    0, //szrawUsername,
                    0, //szrawPassword,
                    0, //szSender,
                    0, //szSubject,
                    0, //szBody,
                    0, //bufpointer,
                    command1,
                    0 ); //callingmode);

Sleep(1000);

for (int n=1;n<130;n++){Sleep(100);
int y=0;
int command2='N'; //polling mode
szToAddr[0]=0; //no email transmission with dummy telephone number
 y=ap15(telnumber,alarmmessage, commport, //dummy alarm message
                    0, szToAddr, //polling only
                    0, 0, 0, 0, 0, 0, 0,
                    command2,
                    0 );

if (y==1){ char alarmmessage2[]="123418313101014";
 //wait here until ap15 completes transmission
telnumber[0]=0; //no dialing
int command3A='S';  //mode that hangs up telephone after done
szToAddr[0]=0; //no email
  ap15(telnumber,alarmmessage2, commport,
                    0, szToAddr,
```

```
0, 0, 0, 0, 0, 0, 0,
command3A,
0 ); n=130;}}
```