

Instruction of Level 3 Interface

Content

1	Summarize	1
1.1	Definition and application.....	1
1.2	Files Introduction.....	1
1.3	Functions Introduction.....	1
2	Process of Functions Call	3
3	Functions Detail	6
3.1	LV3_InitializeSystem.....	6
3.2	LV3_SetAuthCode	6
3.3	LV3_ReadCardID	8
3.4	LV3_IssueChiefCard	8
3.5	LV3_IssueBlankCard	9
3.6	LV3_IssueGuestCard.....	9
3.7	LV3_SetGuestCardExtData.....	12
3.8	LV3_ReadRoomStructure.....	15
3.9	LV3_GuestCardByName.....	15
3.10	LV3_SetGuestCardExtByName	16
3.11	LV3_ReadCardInfo	16
3.12	LV3_ReadCard	18
3.13	LV3_WriteData	18
3.14	LV3_ReadData	19
3.15	LV3_WriteKey	19
3.16	LV3_CheckIssuer	20
3.17	LV3_CompareRegCode.....	20
3.18	LV3_FreeRoomStructure	20
3.19	LV3_FreeSystem	20
4	Return Code Table	21
5	FAQ	25
5.1	About Unlock	25

5.2 About Programming..... 25

5.3 About Return Code..... 26

1 Summarize

1.1 Definition and application

This DLL (Dynamic Link Library) is available when issue Chief Card or Guest Cards with encoder, and this two cards can unlock the doors which is install in LEVEL 3 Hotel Lock System.

1.2 Files Introduction

Files:

- ✚ Five *.dll files: LevelLock.dll, lv3.dll, EMUSB.dll, RCUSB.dll, RFUSB.dll(All in folder “LEVEL3_EN_Build”);
- ✚ One user manual, it’s this manual;
- ✚ Several Demos that show how to use this DLL(Folders name with “Source” at the end are sample source code of calling this DLL in different development platform, and the *.exe files that after compiled are in folder “LEVEL3_EN_Build”);
- ✚ One “RoomStructure.lvr” file and one “Sample data in RoomStructure.lvr.txt” file(For details of these two files see “3.8 LV_ReadRoomStructure”);

1.3 Functions Introduction

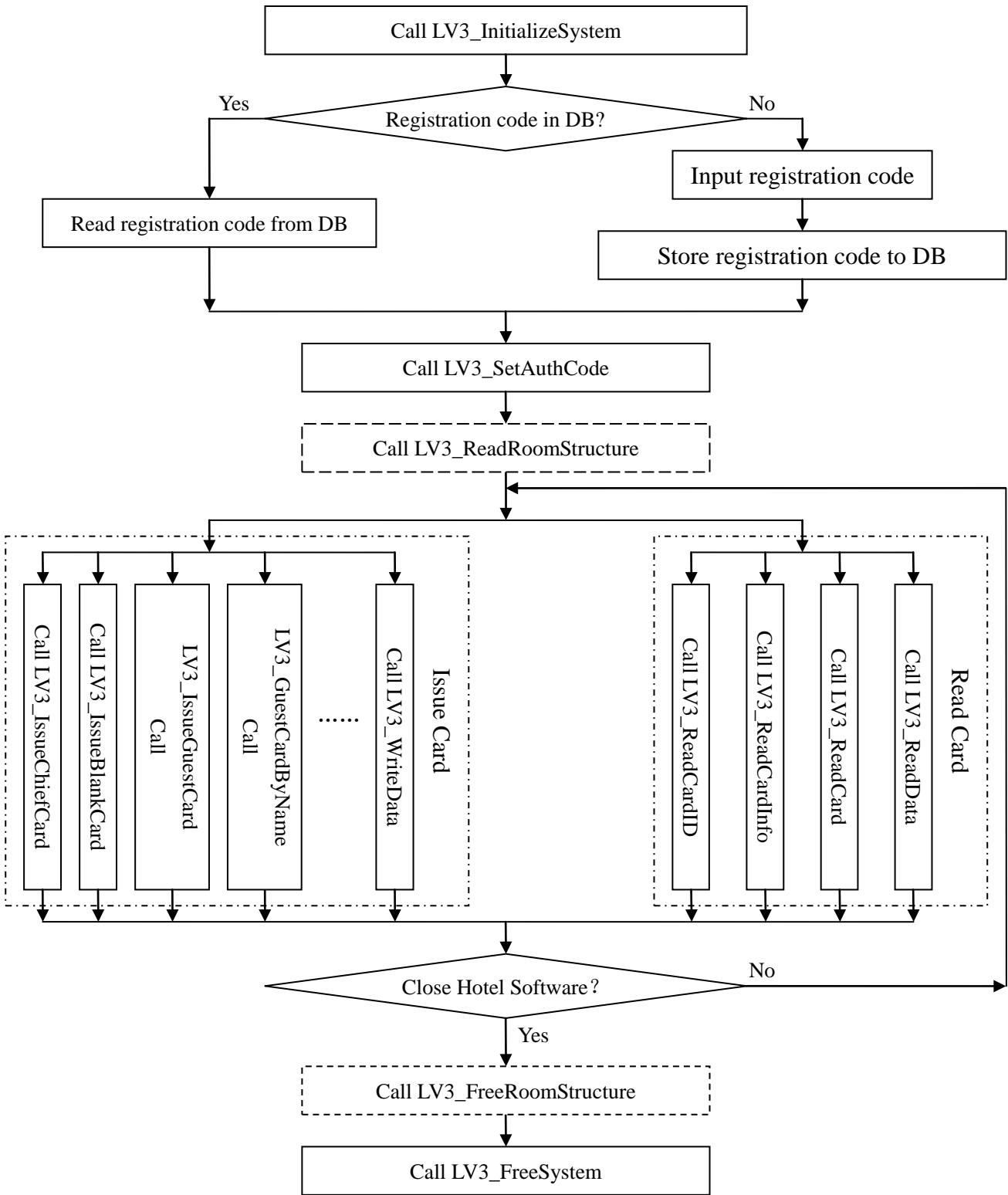
The functions include in this DLL are list as table 1-1. Each parameter and return is base as data type in Delphi2007.

Table 1-1 Functions List

Function	Return	Description
LV3_InitializeSystem(Smallint)	SmallInt	Choose the type of card and initialize system
LV3_SetAuthCode(Pointer, Smallint, Smallint, Smallint, Smallint, Smallint, Smallint)	SmallInt	Set Registration Code and Sectors
LV3_ReadRoomStructure()	SmallInt	Read Room Structure
LV3_CheckIssuer()	SmallInt	Check the encoder

LV3_CompareRegCode(Pointer, Pointer)	SmallInt	Compare Registration Code
LV3_IssueChiefCard()	SmallInt	Issue a Chief Card
LV3_IssueBlankCard()	SmallInt	Issue a Blank Card
LV3_IssueGuestCard(Pointer, Pointer, Pointer, Pointer, Pointer, Pointer, Pointer, Smallint)	SmallInt	Issue a Guest Card
LV3_SetGuestCardExtData(Pointer, Pointer, Pointer, Pointer, Smallint)	SmallInt	Set Guest Card extend room data
LV3_GuestCardByName(Pointer, Pointer, Pointer, Pointer, Pointer, Pointer, Pointer, Smallint)	SmallInt	Issue a Guest Card by room name
LV3_SetGuestCardExtByName(Pointer, Pointer, Pointer, Pointer, Smallint)	SmallInt	Set Guest Card extend room data by room name
LV3_WriteData(Pointer, Pointer, SmallInt)	SmallInt	Write 48 bytes data in card
LV3_WriteKey(Pointer, Pointer, SmallInt)	SmallInt	Revise Write Read Key
LV3_ReadCard(Pointer, Pointer)	SmallInt	Read card data
LV3_ReadCardInfo(Smallint, Pointer, Pointer)	SmallInt	Read card information
LV3_ReadCardID(Pointer)	SmallInt	Read card ID
LV3_ReadData(Pointer, Pointer, SmallInt)	SmallInt	Read 48 bytes data
LV3_BeepOk()	SmallInt	Sound of success
LV3_BeepFailure()	SmallInt	Sound of failure
LV3_BeepAlarm()	SmallInt	Sound of alarm
LV3_FreeSystem()	SmallInt	Release system
LV3_FreeRoomStructure()	SmallInt	Release room structure

2 Process of Functions Call



Picture 2-1 Process of Functions Call

Regarding to the picture displays above, process of functions call in DLL can be divided into three steps:

Step 1:

Initialization. This step includes [Call LV3_InitializeSystem] and [Call LV3_SetAuthCode], if need to issue a Guest Card by calling function LV3_GuestCardByName and set Guest Card extend room data by calling function LV3_SetGuestCardExtByName, this step will also includes[Call LV3_ReadRoomStructure]. Registration code is required before issuing or reading cards, therefore it is suggesting store it into DataBase.

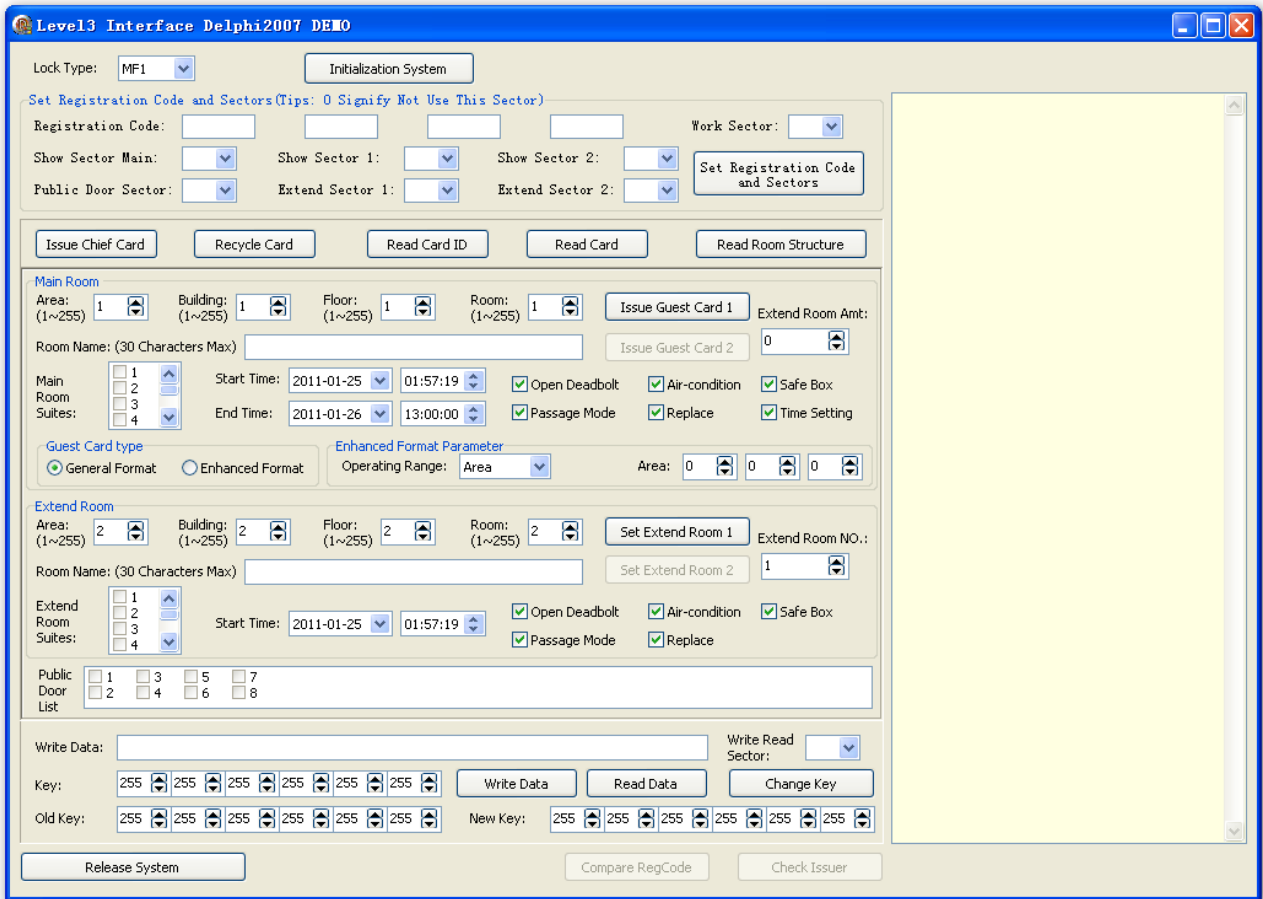
Step 2:

Issue card and read card. This step includes several issue card or read card functions. Once runs step 1, the issue card and read card operation can be done much time before run step 3.

Step 3:

Close. This step only includes [Call LV3_FreeSystem]. Once run this step, issue card or read card function is unavailable, if we must issue card and read card again, please restart from step 1.

Picture 2-2 is a Delphi2007 demo GUI.



Picture 2-2 Delphi2007 demo GUI

3 Functions Detail

The returns of every functions in this DLL are a return code, return 0 signify the process is success, others signification of return code, please refer to chapter 4 “Return Code Table”.

3.1 LV3_InitializeSystem

Description: Selecting specified type of card and initialize system. Before issue or read card, this function needs to be called once.

Declare: LV3_InitializeSystem(LockType : smallint):smallint;

Argument:

LockType: Its data type is smallint, different value of LockType signify different card types.

Details can be referred to table 3-1.

Table 3-1 Card Types List

Value of LockType	Card Type
4	RF57
5	MF1
6	S70
7	MF0
8	S20
9	RF80

3.2 LV3_SetAuthCode

Description: Setting registration code and sectors. Only the setting is correct can cards unlock door.

Declare: LV3_SetAuthCode(RegCode : pointer; WorkSector, ShowSectorMain, ShowExtSector1, ShowExtSector2, PubDoorSector, ExtendSector1, ExtendSector2 : smallint):smallint;

Argument:

RegCode: This is a 16 characters string, and store registration code (consist of 4 bits registration code from each 4 sections) which input by user.

WorkSector: Its data type is smallint, different value of WorkSector signify use different sector of card as work sector. Work sector must be used. See Table 3-2 about the sectors of different card types.

ShowSectorMain: Its data type is smallint, different value of ShowSectorMain signify use different sector of card as show sector main, 0 signify not use show sector main. See Table 3-2 about the sectors of different card types.

ShowExtSector1: ShowExtSector1 is similar to ShowSectorMain, the only difference is that show sector main has to be set before set show sector 1.

ShowExtSector2: ShowExtSector2 is similar to ShowExtSector1, the only difference is that show sector 1 has to be set before set show sector 2.

PubDoorSector: Its data type is smallint, different value of PubDoorSector signify use different sector of card as public door sector, 0 signify not use public door sector. See Table 3-2 about the public doors amount and sectors of different card types.

ExtendSector1: Its data type is smallint, different value of ExtendSector1 signify use different sector of card as Guest Card extend sector 1, 0 signify not use Guest Card extend sector 1.

ExtendSector2: ExtendSector2 is similar to ExtendSector1, the only difference is that Guest Card extend sector 1 has to be set before set Guest Card extend sector 2.

Picture 3-2 Public Doors and Sectors of Different Card Types

Card Type	Sectors	Public Doors	
		Not Use Public Door Sector	Use Public Door Sector
RF57	1	1~8	
RF50	1~15	1~120	1~480
S70	1~39	1~120	1~480
MF0	1	1~120	
S20	1~4	1~120	1~480
RF80	1	1~8	

Note:

1. Work Sector, Show Sector, Public Door Sector, Guest Card Extend Sector must use different sector.
2. Only Work Sector can use sector 1.
3. Show Sector Main and Guest Card Extend Sector 1 have to be set before set Show Sector 1. Show Sector 1 and Guest Card Extend Sector 2 have to be set before set Show Sector 2. Guest Card Extend Sector 1 has to be set before set Guest Card Extend Sector 2
4. No Extend Sector is set, Show Sector 1 or Show Sector 1 can not be set.
5. You must set Show Sector 1 when both Extend Sector 1 and Show Sector Main are set. You must set Show Sector 2 when both Extend Sector 2 and Show Sector Main are set.
6. Sector of card need to be initialized before use as Work Sector. Our company usually initialized sector 1 and sector 11, if you need us to initialize other sector, please contact us.
7. Sector of card once has been initialized it can not be used as other sectors any more.
8. Sector of card once has been used as Show Sector, Public Door Sector or Extend Sector it can not be used as other sectors any more.
9. Guest Card can open 3 additional guest rooms for each Extend Sector is added.

Tips: Registration code is required before issuing or reading cards. Therefore, it is suggested to save it in file or database after successful registration.

3.3 LV3_ReadCardID

Description: Reading card ID of the card which put on the issuer.

Declare: LV3_ReadCardID(CardID : pointer):smallint;

Argument:

CardID: This is a 16 characters string which will be stored the card ID.

3.4 LV3_IssueChiefCard

Description: Issuing a Chief Card. This Chief Card is able to unlock all doors 1 hour earlier or later than current time, it means Chief Card issue at 13:00 is valid only between 12:00-14:00.

Declare: LV3_IssueChiefCard():smallint;

Tips: Chief Card is able to unlock all doors while the setting of registration code and sectors is correct, so by using Chief Card to unlock the door, it can help to verify whether the setting of registration code and the sectors are correct or not.

3.5 LV3_IssueBlankCard

Description: Issuing a Blank Card.

Declare: LV3_IssueBlankCard():smallint;

3.6 LV3_IssueGuestCard

Description: Issuing a Guest Card. Each room is defined by Area ID, Building ID, Floor ID and Room ID.

Declare: LV3_IssueGuestCard(RoomPara, Suite, PubDoor, OptionNO, StartTime, EndTime, OpPara : pointer; ExtRoomAmt : smallint):smallint;

Argument:

RoomPara: This is a 4 bytes array pointer, the array stores room parameter. From the first byte to the third byte, it signifies Area ID, Building ID, Floor ID and Room ID. The value range of them is from 1 to 255.

Suite: This is an 8 bytes array pointer, the array stores suites that this Guest Card is able to unlock or not, 0 signify unable, 1 signify enabled.

PubDoor: This is a dynamic smallint array pointer, the array stores all public doors that this Guest Card is able to unlock, the last smallint must be 0 to signify no more public door.

OptionNO: This is a 3 bytes array pointer, the array stores Option NO that will be valid under Enhanced Format.

StartTime: This is a 12 characters string, it stores start time. The format of StartTime is: yyyyMMddHHmm. yyyy signify year, MM signify month, dd signify day, HH signify hour, mm signify minute.

EndTime: This is a 12 characters string, it stores end time. Its format is similar with StartTime.

OpPara: This is an 8 bytes array pointer, the array stores the open parameters of Guest Card.

Those parameters are: Open the dead bolt is enabled or not(1: enabled; 0: unable); Air-condition is enabled or not(0: enabled; 2: unable); Safe box is enabled or not(4: enabled; 0: unable); Set the lock under passage mode after unlock or not(8: set; 0: not set); Replace other Guest Card or not(16: replace; 0: not replace); Time Setting (Set the time in lock with start time of Guest Card after unlock. 128: set; 0: not set); The Operating Range(16: Area; 32: Building; 48: Floor; Room; 64: Suite); Guest Card type(128: Enhanced Format; 0: General Format).

When the Guest Card type is Enhanced Format (OpPara[7]=128), Guest Card is able to unlock a lot of rooms by using argument Operating Range (OpPara[6]), RoomPara and OptionNO. For example, Operating Range is floor (OpPara[6]=48), Building ID is 3 (RoomPara[1]=3) and OptionNO is 2, 4 and 6. Then this Guest Card is able to unlock all rooms that Building ID is 3 and Floor ID is 2, 4 or 6. If OptionNO is 0, 2, 7 (or 0, 7, 2), this Guest Card is able to unlock all rooms that Building ID is 3 and Floor ID is 2, 3, 4, 5, 6 or 7.

ExtRoomAmt: Its data type is smallint, it signifies the amount of Guest Card extend room.

Sample:

Supposing the configuration of a Guest Card is: Area ID is 1; Building ID is 2; Floor ID is 3; Room ID is 4; is able to unlock suite 2, 3, 8; is able to unlock public door 2, 6, 18, 72, 113, 246, 325, 477; check-in time is 2010-9-22 13:30; check-out time is 2010-9-24 11:30; open the dead bolt is enabled; air-condition is enabled; safe box is unable; set the lock under passage mode; replace other Guest Card; without time setting; Enhanced Format; also be able to unlock all rooms that Building ID is 2 and Floor ID is 4, 5, 6, 7 or 8. Here are some sample codes to issue this Guest Card:

Define variable:

RoomPara : array[0..3] of byte;

Suite : array[0..7] of byte;

PubDoor : array of smallint;

OptionNO : array[0..2] of byte;

StartTime, EndTime : array[0..11] of char;

OpPara : array[0..7] of byte;

Assigning value:

RoomPara[0]:=1; // Area ID is 1

RoomPara[1]:=2; // Building ID is 2

```
RoomPara[2]:=3;      // Floor ID is 3
RoomPara[2]:=4;      // Room ID is 4
for i := 0 to 7 do Suite[i]:=0;
Suite[1]:=1;         // Unlock suite 2
Suite[2]:=1;         // Unlock suite 3
Suite[7]:=1;         // Unlock suite 8
Setlength(PubDoor,9); // Unlock 8 public doors
PubDoor[0]:=2; PubDoor[1]:=6; PubDoor[2]:=18; PubDoor[3]:=72;
PubDoor[4]:=113; PubDoor[5]:=246; PubDoor[6]:=325; PubDoor[7]:=477;
PubDoor[8]:=0;       // 0 signify no more public door
StartTime:='201009221330'; // Check-in time is 2010-9-22 13:30
EndTime:='201009241130'; // Check-out time is 2010-9-24 11:30
OpPara[0]:=1;        // Open the dead bolt is enabled
OpPara[1]:=0;        // ** Air-condition is enabled **
OpPara[2]:=0;        // Safe box is unable
OpPara[3]:=8;        // Set the lock under passage mode
OpPara[4]:=16;       // Replace other Guest Card
OpPara[5]:=0;        // Without time setting
OpPara[6]:=48;       // Operating Range is floor
OpPara[7]:=128;     // Enhanced Format
OptionNO[0]:=0;      // Unlock all rooms that Building ID is 2 and Floor ID is 4, 5, 6, 7 or 8
OptionNO[1]:=4;
OptionNO[2]:=8;
At last, call function:
LV3_IssueGuestCard(@RoomPara, @Suite, @PubDoor[0], @OptionNO, @StartTime,
@EndTime, @OpPara, 0);
```

Then, this Guest Card can unlock the door with above conditions.

Note:

1. When more than one Guest Card need to unlock the same door and the value of argument *ReplaceState* is “True”, some Guest Cards may replace by others. “Replace” means that:

the Guest Card with later start time can replace all the Guest Card with earlier start time, and the Guest Card which has been replace is unable to unlock the door anymore. For example, Guest Card A and B is able to unlock the same door, but the start time of Guest Card A is 2010-9-22 **13:31**, and the start time of Guest Card B is 2010-9-22 **13:30**. After Guest Card A unlock the door, Guest Card B is unable to unlock this door anymore. So, in order to issue more than one available Guest Card for a room, please guarantee they have the same start time.

2. When assigning value, the length of string *StartTime* and *EndTime* must be fixed, otherwise error occurs. For example, if start time is 2010-9-1 3:01, string *StartTime* should be assigned as '201009010301', a 12 characters string, the '0' can not be omitted. String *EndTime* should be assigned value as the same way.
3. When assigning value, start time must earlier than end time.
4. Cause of timing error may between lock and PC, in order to avoid this error and guarantee Guest Card can unlock the door normally, start time should be set 30 minutes advance, and end time should be set 30 minutes delay.
5. The last smallint of array *PubDoor* should be set as 0 to signify no more public door. So when there is no need to unlock public door, array *PubDoor* should be set *PubDoor[0]:=0*. Delphi programmer please noted that: When calling function *LV3_IssueGuestCard*, the argument is *@PubDoor[0]* and not *@PubDoor*.
6. Operating Range (OpPara[6]) and OptionNO are valid only under Enhanced Format.
7. The value of ExtRoomAmt must less than or equal to the real amount of Guest Card extend room (Each time call function *LV3_SetGuestCardExtData* or *LV3_SetGuestCardExtByName*, one Guest Card extend room will be set), please see next chapter.

3.7 LV3_SetGuestCardExtData

Description: Set Guest Card extend room data. Each time call this function, one Guest Card extend room will be set. The max amount of Guest Card extend room is 3 or 6, it's decided by the arguments of calling function *LV3_SetAuthCode*. Each room is defined by Area ID, Building ID,

Floor ID and Room ID.

Declare: LV3_SetGuestCardExtData(RoomPara, Suite, StartTime, OpPara : Pointer;
SequenceNO : smallint):smallint;

Argument:

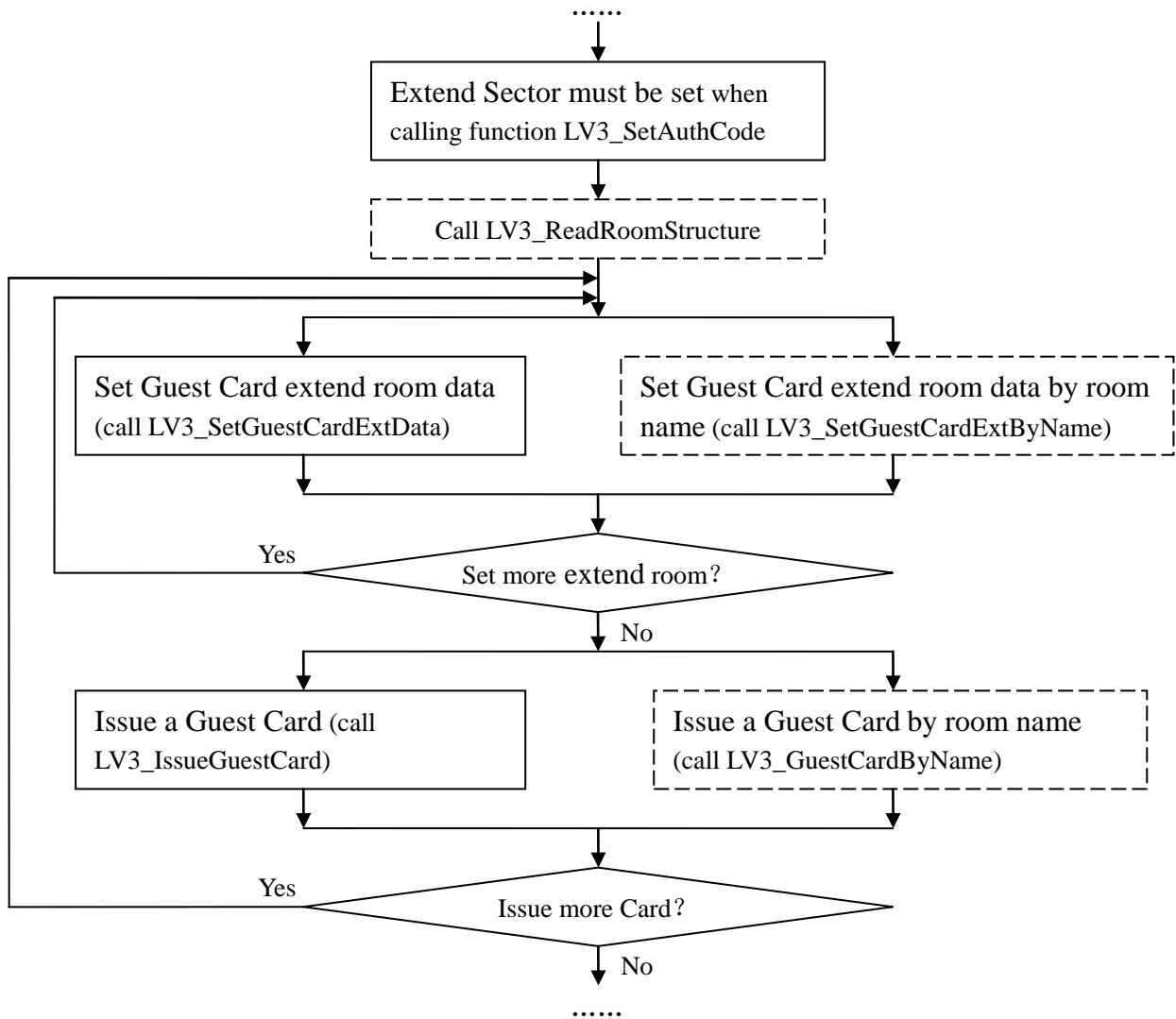
Arguments RoomPara, Suite, StartTime are similar to “3.6 LV3_IssueGuestCard”.

OpPara: This is a 5 bytes array pointer, the array is the first five bytes of OpPara in “3.6 LV3_IssueGuestCard”, they are dead bolt, air-condition, safe box, passage mode, replace.

SequenceNO: Its data type is smallint, it signifies the sequence number of Guest Card extend room.

Process of issue a Guest Card that is able to unlock Guest Card extend room:

- 1) When calling function LV3_SetAuthCode, Extend Sector must be set.
- 2) Set several Guest Card extend rooms (call function LV3_SetGuestCardExtData or LV3_SetGuestCardExtByName several time).
- 3) Issue a Guest Card (call function LV3_IssueGuestCard or LV3_GuestCardByName).



Picture 3-1 Issue Process

Note:

1. When calling function LV3_SetAuthCode, if no Extend Sector has been set, then Guest Card extend room data can not be set; if only set Extend Sector 1, then the maximum value of SequenceNO is 3; if both Extend Sector 1 and Extend Sector 2 are set, then the maximum value of SequenceNO is 6. If SequenceNO is bigger than the maximum value, then return 39 when setting Guest Card extend room data.
2. When issuing a Guest Card, if the argument ExtRoomAmt is n, then Guest Card extend room data must be set n time (the sequence number of Guest Card extend room from 1 to n must be set), otherwise return 12 when issuing a Guest Card.
3. Other notes please see “3.6 LV3_IssueGuestCard”.

3.8 LV3_ReadRoomStructure

Description: Reading room structure information from file “RoomStructure.lvr” (For details of this file see “Tips”). Before issue a Guest Card by room name or set extend room data by room name, room structure should be read successfully once.

Declare: LV3_ReadRoomStructure():smallint;

Tips:

1. The process of read room structure are as follow:
 - 1) Run Level 3 Hotel Lock System;
 - 2) Settings-> Hotel Structure;
 - 3) After room structure setting finish, click [Export Room Structure], choose export path and click [ok];
 - 4) Put file “RoomStructure.lvr” into the same file directory of “LevelLock.dll”;

Then, call function LV3_ReadRoomStructure can read room structure from “RoomStructure.lvr”.

2. There is a file “RoomStructure.lvr” in folder “LEVEL3_EN_Build”, and the room structure of this file is as file “Sample data in RoomStructure.lvr.txt” show (Also in folder “LEVEL3_EN_Build”), programmer can do some test through these two files.

Note:

1. File “RoomStructure.lvr” should be put into the same directory of “LevelLock.dll”.

3.9 LV3_GuestCardByName

Description: Issuing a Guest Card by room name.

Declare: LV3_GuestCardByName(RoomName, Suite, PubDoor, OptionNO, StartTime, EndTime, OpPara : Pointer; ExtRoomAmt : smallint):smallint;

Argument:

RoomName: This is a 30 characters string, it stores the room name.

Other arguments are similar to “3.6 LV3_IssueGuestCard”.

Note:

Besides notes mention on chapter “3.6 LV3_IssueGuestCard” please note that:

1. Issuing Guest Card by calling this function, function LV3_ReadRoomStructure should be called successfully once.
2. Cause of the max length of room name in Level 3 Hotel Lock System is 30 characters, argument RoomName can not be set more than 30 characters.

3.10 LV3_SetGuestCardExtByName

Description: Set Guest Card extend room data by room name.

Declare: LV3_SetGuestCardExtByName(RoomName, Suite, StartTime, OpPara : Pointer; SequenceNO : smallint):smallint;

Arguments and notes are similar to “3.7 LV3_SetGuestCardExtData“and “3.9 LV3_GuestCardByName”.

3.11 LV3_ReadCardInfo

Description: Reading card information and save it to a string according to the type of language.

Declare: LV3_ReadCardInfo(LanguageType : smallint ; CardType , CardInfo : pointer):smallint;

Argument:

LanguageType: Its data type is smallint, it signifies language types, 0 signify English.

CardType: This is a byte pointer, after call this function, its store code of card function which can be seen in table 3-3.

CardInfo: This is a 5000 characters string, it stores card information.

Table 3-3 Function Types of Cards List

Code	Function Types of Cards
0	Authorized Card
1	Clock Card

2	Install Card
3	Spare Card
4	Data Card
5	Check Out Card
6	Stop Card
7	Lost Card
8	Meeting Card
9	Emergent Card
10	Master Card(Chief Card)
11	Gate Card
12	Area Card
13	Building Card
14	Floor Card
15	Guest Card
255	Blank Card

Sample:

Define variable:

CardInfo:array[0..4999] of char;

CardType: byte;

After calling function *LV3_ReadCardInfo(0,@CardType, @CardInfo)*, *CardType* stores the code of card function type, *CardInfo* stores English information from the card.

Note:

1. The setting of sectors when calling function *LV3_SetAuthCode* before reading card and issuing card must be same. For example, call *LV3_SetAuthCode(@RegCode, 11, 0, 0, 0, 2, 0)* before issuing card, but before reading this card, the function call is *LV3_SetAuthCode(@RegCode, 11, 0, 0, 0, 4, 0)*, then this card can't read correctly(return 17 or read the false information).

3.12 LV3_ReadCard

Description: Reading card data.

Declare: LV3_ReadCard(CardType : pointer; CardInfo : pointer):smallint;

Argument:

Argument CardInfo between this function and LV3_ReadCardInfo is different. CardInfo in this function is digital while in LV3_ReadCardInfo is a long string.

This function will be used only when there is special need, programmer can see the Delphi2007 source code in folder “Read_Card_DEMO_Source” when this function needs to be call.

3.13 LV3_WriteData

Description: Writing 48 bytes or chars data into the specified sector of card. Only when the sectors of card are more then 1, this function is valid. See Table 3-2 about the sectors of different card types.

Declare: LV3_WriteData(WData, WKey : pointer; SectorToWrite : smallint):smallint;

Argument:

WData: This is a 48 characters string, it stores the data which will be written into the specified sector of card.

WKey: This is a 6 bytes array pointer, the array stores the key of the specified sector of card. This key must be correct, otherwise data in WData can not be written into the card. This key is consisting of 6 bytes, range of each byte is from 0 to 255. The default key of sectors is 6 times of 255.

SectorToWrite: Its data type is smallint, different value of SectorToWrite signify write data in different sector of card. See Table 3-2 about the sectors of different card types.

Note:

1. Write Read Sector, Work Sector, Show Sector, Public Door Sector, Guest Card Extend Sector must not in the same sector.
2. If the specified sector of card has been initialized or it has been used as Show Sector, Public Door Sector or Guest Card Extend Sector it can not be use as Write Read Sector.

3.14 LV3_ReadData

Description: Reading the data that is written by calling function LV3_WriteData. Only when the sectors of card are more than 1, this function is valid. See Table 3-2 about the sectors of different card types.

Declare: LV3_ReadData(RData, RKey : pointer; SectorToRead : smallint):smallint;

Argument:

RData: This is a 48 characters string, it stores the data which read from the specified sector.

RKey: This is a 6 bytes array pointer, the array stores the key of the specified sector of card. This key must be correct, otherwise the data can not be read from the card. Default: 6 times of 255.

SectorToRead: Its data type is smallint, different value of SectorToRead signify read data from different sector of card. See Table 3-2 about the sectors of different card types.

Notes are similar to “3.13 LV3_WriteData”.

3.15 LV3_WriteKey

Description: Revising the key of the specified sector of card.

Declare: LV3_WriteKey(OldKey, NewKey : pointer; SectorToWrite : smallint):smallint;

Argument:

OldKey: This is a 6 bytes array pointer, the array stores the old key of the specified sector of card. This key must be correct, otherwise the key can not be revised successfully. Default: 6 times of 255.

NewKey: This is a 6 bytes array pointer, the array stores the new key.

SectorToWrite: Its data type is smallint, different value of SectorToWrite signify revise the key of different sector of card. See Table 3-2 about the sectors of different card types.

Note:

Besides notes mention on chapter “3.13 LV3_WriteData” please note that:

1. While calling this function to revise key, please note that: If the key of sectors is not 6 times of 255, this sector can not be used as Show Sector or Public Door Sector.

3.16 LV3_CheckIssuer

Description: Check the encoder connected or not, return 0 signify the encoder has connected.

Declare: LV3_CheckIssuer():smallint;

3.17 LV3_CompareRegCode

Description: Compare the registration code, check the authorization codes are same or not, return 0 signify same.

Declare: LV3_CompareRegCode(OldRegCode, NewRegCode: Pointer):smallint;

Argument:

OldRegCode: This is a 16 characters string, and store old registration code.

NewRegCode: This is a 16 characters string, and store new registration code.

3.18 LV3_FreeRoomStructure

Description: If there is no need to issue Guest Card by calling function LV3_GuestCardSingle or LV3_GuestCardDouble, call this function to release room structure. If never call function LV3_ReadRoomStructure to read room structure, this function is not necessary.

Declare: LV3_FreeRoomStructure():smallint;

3.19 LV3_FreeSystem

Description: Releasing system resource.

Declare: LV3_FreeSystem():smallint;

4 Return Code Table

Error code	Description
0	Success
Error caused by Issuer	
80	Error in request for the card reader
81	Fail to read the card code!
82	Fail to choose card!
83	Fail to verify the key for card data.
84	Fail to write data into the card!
86	Fail to read data from the card!
89	No card exists
90	Card reader can not be detected.
91	Invalid card
92	Incorrect card type
93	Read-write error.
94	The communication port still close.
95	End of reading data card.(finish receiving data)
98	Other error
99	The port have been took up
100	Communication error
Error caused by dongle	
101	Dongle authority is not allowed to operate this function. Please redo it
102	Fail to write data to dongle!
103	Fail to verify the password of dongle!
104	This dongle has been locked up and can't be used any more!
105	Dongle parameter Error!
106	Error in operating this dongle!
107	Fail to send data to this dongle!

108	Fail to receive data from this dongle!
109	Fail to encrypt the dongle!
110	Fail to decrypt the dongle!
111	Error to read data from the dongle!
112	Dongle is running!
113	Dongle is not running!
114	Error in the dongle password!
115	Can not find the dongle!
116	Fail to run the dongle!
117	Verifying time of dongle password is 0.
118	Error in format of the dongle password!
119	Data written into the dongle is too long!
120	Error in dongle password!
Other error	
126	Some system file is missing.
127	Fail to save file.
128	Fail to read file.
148	The version of backup file does not match this application!
149	Structure of the backup file is not completed or error in verifying error!
150	Software types do not match.
151	The name has already existed!
153	The data is not found or it is not the only existence.
155	This data has associated data!
156	Cannot find file(picture)in the specified path !
164	Dongle does not match the software.
165	The system is expired.
166	Please insert an initialized card from LEVEL.
167	Registration code will be out of use soon, please contact the dealer to renew it!
168	Issuing time parameter is out of the valid period of the software!

171	Error in executing initialization!
172	Error in unknown encrypted device!
173	Unknown error in issuing card!
174	Error in encrypting or decrypting!
Error caused by issue card	
10	Error in system initialization!
11	Have not initialize system yet!
12	Format error in card parameters
13	System does not support this type of card
14	Card-issuing parameter is out of permission
15	System does not support this kind of port for Issuer
16	Can not issue cards which lap over
17	Error in card data verification
18	Current card type is different from the card which the system read
19	Error in card data format
23	This card has been initialized already
24	This card is not suit for this system! (Different Authorization code)
25	Password from this card is not suit for this system! (different time password)
26	This card is the Authorization card for this system. Can not be issued as other kinds of card!
27	Error in recognizing card type!
28	System is not in the state of completing this action!
30	System time is earlier than the replace time!
31	This card is not a blank card. Please recycle it first
32	Error in format of the card code!
33	Error in Registration Code format
34	Dongle language does not match the system language
35	Error in code format in the system
36	Unmatched Authorized Code

37	Authorized Parameter or Authorized Parameter Format Error
38	Card type error
51	Parameters is out of permission
52	Format error in parameters
Error caused by new dongle	
40	Fail to start encryption system
41	Fail to locate encryption system
42	Parameter Format Error
43	Address for the encrypted data is incorrect
44	Communication Error
45	Password of encryption system is incorrect
46	Fail to apply the access to the encryption system
47	Already start up the encryption system
48	Location memory block error
49	Data of Upgrade Package is invalid
50	Synchronous Data Error
Error caused by compare registration code	
202	Error in registration code format
203	Error in new registration code format
204	The two registration code are not from the same authorization code
205	Error in date format of old registration code
206	Error in date format of new registration code
207	The deadline of new registration code is earlier then old one
208	New registration code has been out of use
209	New registration code will be out of use in one week

5 FAQ

5.1 About Unlock

1. Guest Card is unable unlock the door and the Chief Card neither

Possible Cause:

1. The setting of registration code or sectors is incorrect.
2. Data errors when registration code is read from database.
3. The revised time is incorrect.

Solution:

Solution 1: Check the expiration of the registration code.

Solution 2: Re-input registration code, and issue a new Chief Card in DEMO. If the door can be unlocked by this new Chief Card, it means that read in error from database. Please check the process of reading from database and find out problem.

Solution 3: Revise the time in lock.

2. Guest Card is unable unlock the door while Chief Card is enabled

Possible Cause:

1. Area ID, Building ID, Floor ID or Room ID in Guest Card does not match those in the lock.
2. Out of valid period of Guest Card.
3. This Guest Card has been replaced by another card. About replace, please refer to “Note” 1 of chapter “3.6 LV3_IssueGuestCard”.

Solution:

Solution 1: Check 4 IDs of the lock are match or not.

Solution 2: Make sure the time is not out of valid period of Guest Card. Set start time 30 minutes advance and end time 30 minutes delay will help to avoid the timing error.

Solution 3: Find out whether this Guest Card has been replaced.

5.2 About Programming

1. Error occur when calling functions

Possible Cause:

The data types or value range of arguments is not match. For example, the value range of char in VS2008(C#) is 0~65535, while the value range of char in Delphi2007 is 0~255.

Solution:

Match the same value range of both types of data. For example, value range of byte in VS2008(C#), should match value range of char in Delphi2007.

2. The Setting of Public Door is Incorrect**Possible Cause:**

When calling functions of issue Guest Card, the argument use *@PubDoor*. For example:

...

```
Setlength(PubDoor,3);
```

```
PubDoor[0]:=2; PubDoor[1]:=6; PubDoor[2]:=0;
```

```
LV3_IssueGuestCard(@RoomPara, @Suite, @PubDoor, @OptionNO, @StartTime,
```

```
@EndTime, @OpPara, 0);
```

Solution:

```
LV3_IssueGuestCard(@RoomPara, @Suite, @PubDoor[0], @OptionNO, @StartTime,
```

```
@EndTime, @OpPara, 0);
```

5.3 About Return Code

Code	Possible Cause(Solution)
11	The system has not been initialized before set the registration code and sectors; The registration code and sectors have not been set before issue card or read card;
12	The format of time string wrong(See “Note” 2, 3 of “3.6 LV3_IssueGuestCard”); The setting of public door wrong(See “Note” 5 of “3.6 LV3_IssueGuestCard”); The amount of Guest Card extend room when issuing a Guest Card(See “Note” of “3.7 LV3_SetGuestCardExtData”);
14	The Area ID, Building ID, Floor ID, Room ID is out of 1 to 255;
17	Sector of card has not been initialized before use as work sector(Change sector, our company usually initialized sector 1 and sector 11, or contact us);

	The setting of sectors not match when reading card(See “Note” of “3.11 LV3_ReadCardInfo”);
23	Duplication set registration code and sectors;
28	Duplication initialize system;
33	Registration code error(Check the input, or contact us to check the registration code);
38	Used the card type does not support when calling function LV3_InitializeSystem to initialize system(Only support 4 to 9, see table 3-1);
41	The dongle not insert yet; Something wrong with the USB port(Change another port, or change another PC);
47	Others system is using the dongle(Close all programs which may using the dongle);
51	Used the sector that this type of card not have(See Table 3-2);
52	The setting of sectors incorrect (See “Note” of “3.2 LV3_SetAuthCode”);
80	The issuer not insert yet; Something wrong with the USB port(Change another port, or change another PC); The type of issuer does not match the type of card was set when initialize system; The type of issuer does not match the type of card which put on the issuer;
83	The key wrong when write 48 char to card or read 48 byte from card; The old key wrong when change the key; Sector of card has not been initialized before use as work sector(Change sector, our company usually initialized sector 1 and sector 11, or contact us);
148,149	File “RoomStructure.lvr” is incorrect(Export a new one in Level3);
153	The room name not exist;
156	File “RoomStructure.lvr” and “LevelLock.dll” not under the same directory;
165	The registration code is overdue(Contact us); The start time or end time of Guest Card is out of the valid period of the registration code;