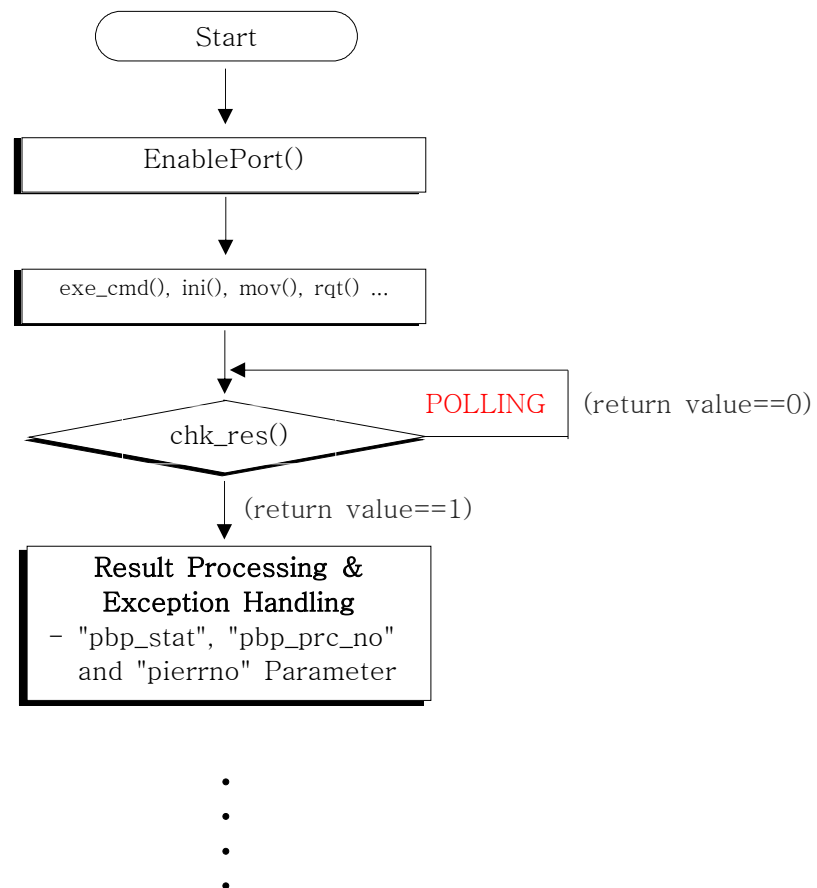


\$. "kyt\_6xxx.dll" Library

- "kyt\_6xxx.dll" Library is for supporting KYT-3XXX and KYT-6XXX terminal.
- Library for Windows NT4.0/2000/XP and Windows 98/Millennium
- Notice: BYTE==unsigned char, UINT==unsigned int

Flow Chart



=====

@. EnablePort()

- Set the serial port to communicate with the terminal.

\* PROTOTYPE

- BOOL EnablePort(char\* port, BYTE size, BYTE parity, BYTE stopbit, DWORD baudrate, BYTE control)

\* PARAMETERS(Refer to "struct \_DCB" in the MSDN.)

- port : ex) "COM1", "COM2", ...
- size : Number of bits/byte, 4-8.
- parity : 0-4=None,Odd,Even,Mark,Space.
- stopbit : 0,1,2 = 1, 1.5, 2.
- baudrate : Baudrate at which running.
- control : 0,1,2,3 = None, XOn/XOff, RTS/CTS, Both.

\* RETURN

- Normal: 1
  - Error : 0
- =====

@. DisablePort()

- Close the serial port to communicate with the terminal.

\* PROTOTYPE

- BOOL DisablePort()

\* PARAMETERS : void

- \* RETURN :
- Normal: 1
- Error : 0

=====

@. exe\_cmd()

- Transmit the command at the terminal

\* PROTOTYPE

- int exe\_cmd(BYTE \*pbp\_cmd, BYTE \*pbp\_dat, UINT piu\_dat\_len)

\* PARAMETERS : Refer to the spec.

- pbp\_cmd: The Pointer of the buffer that 3Byte instruction("Code", Cm" and "Pm")is filled.
- pbp\_dat: The Pointer of the buffer that Data(Data field in the Command structure) of the command is filled.
- piu\_dat\_len: The length of Data.

\* RETURN

- 0 : Parameter Error
- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* DETAIL

- It retry 3 time to the automatic when a communication obstacle happens.
- This function can execute all command that is written on the Spec except "R40" and "R41").
- "M/S Read, M/S Write" command(Refer to "Command Sets List") use this function necessarily.

=====

@. chk\_res()

- Check finishing the execution about the request command. and read the result values about the request command.

\* PROTOTYPE

- int chk\_res(BYTE \*pbp\_stat, BYTE \*pbp\_res,  
UINT \*pipu\_res\_len, BYTE \* pbp\_prc\_no, int \*pi\_errno)

\* PARAMETERS

- pbp\_stat: The Pointer of the buffer to take the status value(1BYTE) of the terminal.
- pbp\_res: The Pointer of the buffer to take the result value of the request command
- res\_len: The length of pbp\_res
- pbp\_prc\_no: The variable for save the sequence number of the request command to process.(1Byte)
- pi\_errno:
  - . Normal: 0
  - . Error : etc(Refer Error Code List.)

\* RETURN

- 0: Don't finish the processing.
- 1: Finish the processing.

\* DETAIL

- If(pi\_errno!=0): You must handle that error
- If(pi\_errno==2000): Negative Response save Negative Code(2Byte) in pbp\_res variable.

=====

@. exe\_stop()

- Stop process about request command.

\* PROTOTYPE

- void exe\_stop()

\* PARAMETERS : void

\* RETURN : void

\* DETAIL

- If you must stop the execution because of the time is delayed.  
Call exe\_stop() function.

=====

@. call\_src\_ver()

- Display source version of the dll program into the Message Box

\* PROTOTYPE

- void call\_src\_ver()

\* PARAMETERS : void

\* RETURN : void

=====

@. ini()

- Execute "Initialize" command(Refer to "Command Sets List")

\* PROTOTYPE

- int ini(BYTE pb\_pm)

\* PARAMETERS :

- pb\_pm:
  - . if(pb\_pm==0x30): Initialize after forward direction Card discharge
  - . if(pb\_pm==0x31): Initialize after reverse direction Card discharge

```

        . if(pb_pm==0x32): Initialize after transfer to "STAND-BY" mode
* RETURN
    - 0      : Parameter Error
    - 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether
               it is the result about request when you read the result.)
=====

```

```

@. rqt()
    - Execute "Request" command(Refer to "Command Sets List")

```

```

* PROTOTYPE
    - int rqt(BYTE pb_pm)
* PARAMETERS :
    - pb_pm:
        . If(pb_pm==0x30): Read present Card location
        . If(pb_pm==0x31): Read F/W Version of unit
* RETURN
    - 0      : Parameter Error
    - 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether
               it is the result about request when you read the result.)
=====

```

```

@. mov()
    - Execute "Moving" command(Refer to "Command Sets List")

```

```

* PROTOTYPE

```

```

    - int mov(BYTE pb_pm)
* PARAMETERS :
    - pb_pm:
        . If(pb_pm==0x30): Card Forward Direction Discharge
        . If(pb_pm==0x31): Card Reverse Direction Discharge(Capture)
        . If(pb_pm==0x32): Transfer to "STAND-BY" mode
        . If(pb_pm==0x33): Card movement(FRONT ->REAR)
        . If(pb_pm==0x34): Card ejection to the front
        . If(pb_pm==0x41): IC Card Accept & Contact
* RETURN
    - 0      : Parameter Error
    - 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether
                it is the result about request when you read the result.)

```

=====

```

@. ic_card_ctl()
    - Execute "IC Card Control" command(Refer to "Command Sets List")

```

```

* PROTOTYPE
    - int ic_card_ctl(BYTE pb_pm, BYTE *pbp_dat, UINT piu_dat_len)
* PARAMETERS :
    - pb_pm:
        . If(pb_pm==0x35): IC Direct Control
        . If(pb_pm==0x38): IC Card Reset
    - pbp_dat:
        . If(pb_pm==0x35): The Pointer of the buffer that Data(Data field in the Command structure) of the command is filled.

```

```

        . If(pb_pm==0x38): NULL
-   piu_dat_len:
        . If(pb_pm==0x35): Length of Data.
        . If(pb_pm==0x38): 0
* RETURN
-   0      : Parameter Error
-   1 ~ 255: The Sequence number about that request(It is utilized to confirm whether
              it is the result about request when you read the result.)

```

=====

```

@. retry_set()
-   Execute "Retry Set" command(Refer to "Command Sets List")

```

```

* PROTOTYPE
-   int retry_set(BYTE pb_pm)
* PARAMETERS :
-   pb_pm: Retry number.
* RETURN      : The Sequence number(pbp_prc_no) about that request.
-   It is utilized to confirm whether it is the result about request
    when you read the result.

```

=====

```

@. clean()
-   Head Cleaning

```

```

* PROTOTYPE

```



```
    - int clean()
* PARAMETERS :
* RETURN
    - 0      : Parameter Error
    - 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether
               it is the result about request when you read the result.)
```

=====

```
@. buzz_ctl()
    - Execute "Buzzer Control" command(Refer to "Command Sets List")
```

```
* PROTOTYPE
    - int buzz_ctl(BYTE pb_pm)
* PARAMETERS :
    - pb_pm:
        . if(pb_pm==0x31): Good Buzzer On
        . if(pb_pm==0x32): Error Buzzer On
```

```
* RETURN
    - 0      : Parameter Error
    - 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether
               it is the result about request when you read the result.)
```

=====

```
@. setting()
    - Execute "setting" command(Refer to "Command Sets List")
```

\* PROTOTYPE

- int setting(BYTE pb\_pm, BYTE pb\_data)

\* PARAMETERS :

- pb\_pm:

- . If(pb\_pm==0x30): Card Wait Time Set
- . If(pb\_pm==0x31): Series(RTS, CTS) Setting
- . If(pb\_pm==0x32): Series(RTS, CTS) Cancellation
- . If(pb\_pm==0x35): Baud Rate Setting

- pb\_data:

- . If(pb\_pm==0x30): Card Wait Time(second)
- . If(pb\_pm==0x31): 0
- . If(pb\_pm==0x32): 0
- . If(pb\_pm==0x35):
  - '0' - 9600 BPS(Default)
  - '1' - 19200 BPS
  - '2' - 38400 BPS
  - '3' - 57600 BPS

\* RETURN

- 0 : Parameter Error
- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

=====



@. mi\_pwr\_on()

- Emit the carrier wave from the antenna.

(CMD: R40)

\* PROTOTYPE

- int mi\_pwr\_on()

\* PARAMETERS : void

\* RETURN

- 0 : Parameter Error
- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_pwr\_off()

- Stop emitting the carrier wave

(CMD: R41)

\* PROTOTYPE

- int mi\_pwr\_off()

\* PARAMETERS : void

\* RETURN

- 0 : Parameter Error
- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_det()

- Confirm whether the antenna detect the RF card.

(CMD: R11)

\* PROTOTYPE

- int mi\_det()

\* PARAMETERS : void

\* RETURN

- 0 : Parameter Error

- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_get\_block()

- Get the Sector and Block Area.

(CMD: R10)

\* PROTOTYPE

- int mi\_get\_block()

\* PARAMETERS : void

\* RETURN

- 0 : Parameter Error

- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_get\_card\_serial()

- Read the card serial number.

(CMD: R14)

\* PROTOTYPE

- int mi\_get\_card\_serial()

\* PARAMETERS : void

\* RETURN

- 0 : Parameter Error

- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_rd()

- Read the data of the specified Block(16 Byte).

(CMD: R20)

\* PROTOTYPE

- int mi\_rd()

\* PARAMETERS : void

\* RETURN

- 0 : Parameter Error

- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_rd\_bal()

- Read the balance of the specified Block

(CMD: R21)

\* PROTOTYPE

- int mi\_rd\_bal()

\* PARAMETERS : void

\* RETURN

- 0 : Parameter Error

- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_set\_block()

- Set the Sector and Block Area.

(CMD: R12)

\* PROTOTYPE

- int mi\_set\_block(BYTE pb\_sct, BYTE pb\_blk)

\* PARAMETERS :

- pb\_sct: Sector(0x00~0x0F).

- pb\_blk: Block(0x00~0x02).

\* RETURN

- 0 : Parameter Error

- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_rd\_key\_type()

- Read the key type (A type or B type)  
(CMD: R13)

\* PROTOTYPE

- int mi\_rd\_key\_type()

\* PARAMETERS : void

\* RETURN

- 0 : Parameter Error
- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_sel\_key\_type()

- Select the key type (A type or B type)  
(CMD: R15)

\* PROTOTYPE

- int mi\_sel\_key\_type(BYTE pb\_index)

\* PARAMETERS

- pb\_index  
0: Key A  
1: Key B

\* RETURN

- 0 : Parameter Error



- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_cng\_card\_key()

- Change the RF card key value of the specified sector.  
(CMD: R30, R31)

\* PROTOTYPE

- int mi\_cng\_card\_key(BYTE pb\_sct, BYTE pb\_access, BYTE \*pbp\_dat)

\* PARAMETERS

- pb\_sct: Sector(0x00 ~ 0x0F)
- pb\_access
  - 0: Access Condition does not exist in "pbp\_dat" buffer.
  - 1: Access Condition exist in "pbp\_dat" buffer.
- pbp\_dat: The Pointer of the buffer that KEY A, Access Condition code and KEY B was filled.  
(Refer to KYT-3xxx SPEC)

\* RETURN

- 0 : Parameter Error
- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_cng\_tm\_key()

- Change the terminal key value of the specified sector.

(CMD: R32)

\* PROTOTYPE

- int mi\_cng\_tm\_key(BYTE pb\_sct, BYTE \*pbp\_dat)

\* PARAMETERS

- pb\_sct: Sector(0x00 ~ 0x0F)
- pbp\_dat: The Pointer of the buffer that KEY A and KEY B was filled.  
(12 BYTE, Refer to KYT-3xxx SPEC)

\* RETURN

- 0 : Parameter Error
- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_wt\_val()

- Write the Balance Value at the specified block in the RF card.

(CMD: R22)

\* PROTOTYPE

- int mi\_wt\_val(ULONG plu\_value)

\* PARAMETERS

- plu\_value: Balance Value to write in the specified block.

\* RETURN

- 0 : Parameter Error
- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_wt\_hex()

- Write data string at the specified block in the RF card.  
(CMD: R23)

\* PROTOTYPE

- int mi\_wt\_hex(BYTE \*pbp\_dat)

\* PARAMETERS

- pbp\_dat: The pointer of the buffer that the data string is filled(16Byte).

\* RETURN

- 0 : Parameter Error
- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_bal\_inc()

- Increase the RF Card's Balance as much as "plu\_value" value.  
(CMD: R24)

\* PROTOTYPE

- int mi\_bal\_inc(ULONG plu\_value)

\* PARAMETERS

- plu\_value: the value to Increase in the specified block.

\* RETURN

- 0 : Parameter Error

- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_bal\_dec()

- Decrease the RF Card's Balance as much as "plu\_value" value.  
(CMD: R25)

\* PROTOTYPE

- int mi\_bal\_dec(ULONG plu\_value)

\* PARAMETERS

- plu\_value: the value to Decrease in the specified block.

\* RETURN

- 0 : Parameter Error
- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_rd\_direct\_key()

- Read the data of the specified Block(By directly using the secret key).  
(CMD: R2A)

\* PROTOTYPE

- int mi\_rd\_direct\_key(BYTE pb\_key\_typ, BYTE pb\_sct, BYTE pb\_blk, BYTE \*pbp\_key\_val)

\* PARAMETERS

- pb\_key\_typ: Key type.  
     0: Key A  
     1: Key B
- pb\_sct: Sector(0x00 ~ 0x0F)
- pb\_blk: Block(0x00~0x02).
- pbp\_key\_val: The Pointer of the buffer that KEY A or KEY B was filled(6BYTE).

\* RETURN

- 0 : Parameter Error
- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_rd\_bal\_direct\_key()

- Read the balance of the specified Block(By directly using the secret key).  
     (CMD: R2B)

\* PROTOTYPE

- int mi\_rd\_bal\_direct\_key(BYTE pb\_key\_typ, BYTE pb\_sct, BYTE pb\_blk, BYTE \*pbp\_key\_val)

\* PARAMETERS

- pb\_key\_typ: Key type.  
     0: Key A  
     1: Key B
- pb\_sct: Sector(0x00 ~ 0x0F)
- pb\_blk: Block(0x00~0x02).
- pbp\_key\_val: The Pointer of the buffer that KEY A or KEY B was filled(6BYTE).

**\* RETURN**

- 0 : Parameter Error
- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

**\* REMARKS**

- Call chk\_res() function for check the result value.

=====

@. mi\_wt\_val\_direct\_key()

- Write the Balance Value at the specified block in the RF card.(By directly using the secret key).  
(CMD: R2C)

**\* PROTOTYPE**

- int mi\_wt\_val\_direct\_key(BYTE pb\_key\_typ, BYTE pb\_sct, BYTE pb\_blk, BYTE \*pbp\_key\_val, ULONG plu\_value)

**\* PARAMETERS**

- pb\_key\_typ: Key type.
  - 0: Key A
  - 1: Key B
- pb\_sct: Sector(0x00 ~ 0x0F)
- pb\_blk: Block(0x00~0x02).
- pbp\_key\_val: The Pointer of the buffer that KEY A or KEY B was filled(6BYTE).
- plu\_value: Balance Value to write in the specified block.

**\* RETURN**

- 0 : Parameter Error
- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

**\* REMARKS**

- Call chk\_res() function for check the result value.

=====

@. mi\_wt\_hex\_direct\_key()

- Write data string at the specified block in the RF card (By directly using the secret key).  
(CMD: R2D)

\* PROTOTYPE

- int mi\_wt\_hex\_direct\_key(BYTE pb\_key\_typ, BYTE pb\_sct, BYTE pb\_blk, BYTE \*pbp\_key\_val, BYTE \*pbp\_dat)

\* PARAMETERS

- pb\_key\_typ: Key type.
  - 0: Key A
  - 1: Key B
- pb\_sct: Sector(0x00 ~ 0x0F)
- pb\_blk: Block(0x00~0x02).
- pbp\_key\_val: The Pointer of the buffer that KEY A or KEY B was filled(6BYTE).
- pbp\_dat: The pointer of the buffer that the data string is filled(16Byte).

\* RETURN

- 0 : Parameter Error
- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_bal\_inc\_direct\_key()

- Increase the RF Card's Balance as much as "plu\_value" value.  
(CMD: R2E)

\* PROTOTYPE

- int mi\_bal\_inc\_direct\_key(BYTE pb\_key\_typ, BYTE pb\_sct, BYTE pb\_blk, BYTE \*pbp\_key\_val, ULONG plu\_value)

\* PARAMETERS

- pb\_key\_typ: Key type.
  - 0: Key A
  - 1: Key B
- pb\_sct: Sector(0x00 ~ 0x0F)
- pb\_blk: Block(0x00~0x02).
- pbp\_key\_val: The Pointer of the buffer that KEY A or KEY B was filled(6BYTE).
- plu\_value: the value to Increase in the specified block.

\* RETURN

- 0 : Parameter Error
- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

@. mi\_bal\_dec\_direct\_key()

- Decrease the RF Card's Balance as much as "plu\_value" value.

(CMD: R2F)

\* PROTOTYPE

- int mi\_bal\_dec\_direct\_key(BYTE pb\_key\_typ, BYTE pb\_sct, BYTE pb\_blk, BYTE \*pbp\_key\_val, ULONG plu\_value)

\* PARAMETERS

- pb\_key\_typ: Key type.
  - 0: Key A
  - 1: Key B
- pb\_sct: Sector(0x00 ~ 0x0F)



- pb\_blk: Block(0x00~0x02).
- pbp\_key\_val: The Pointer of the buffer that KEY A or KEY B was filled(6BYTE).
- plu\_value: the value to Decrease in the specified block.

\* RETURN

- 0 : Parameter Error
- 1 ~ 255: The Sequence number about that request(It is utilized to confirm whether it is the result about request when you read the result.)

\* REMARKS

- Call chk\_res() function for check the result value.

=====

#### \$. Error Code List

1	: No ACK Error
2	: TIMEOUT Error
3	: NAK Error
2000	: Negative Error
102	: Compulsion termination Error
106	: Packet Frame Error
107	: BCC Error

#### \$ Implement the Manager Program.

1.Open and set the serial port by call EnablePort() function in“kyt\_6xxx.dll”.

2.Call exe\_cmd(), ic\_card\_ctl(),... function according to the deed to do.

The Returned value(pbp\_prc\_no) in an each function is the sequence number about that request command.

It is utilized to confirm whether it is the result about the request when you read the result of the request command.

3. For confirm the result about the request. First you ckeck finishing the execution

about the request by `chk_res()` function. if that finished the execution. Process the result values.

4.If you must stop the execution because of the time is delayed. Call `exe_stop()` function.

5.Call `DisablePort()` function if you close the serial port.

6. If you changes the setup of the serial port. call `EnablePort()` function again.

But you must call `DisablePort()` function for close the existing serial port before call

`EnablePort()` function.