IS - 3400 V3.0 RFID Reader

ISO 14443-A
ISO 14443-B
ISO 15693
Mifare Classic
Mifare UltraLight
Mifare Plus
Mifare NTAG
ICode SLIX1, ICode SLIX2
Encryption AES-128Bit, 3DES

날짜	버전	내용
2012.02.29	V1.0	V 1.0 Release
2012.10.20	V1.4	V 1.4 Release
2017.06.12	V3.0	V 3.0 Release
2017.10.02	V3.1	V 3.1 Release
2018.09.18	V3.2	V 3.2 Release

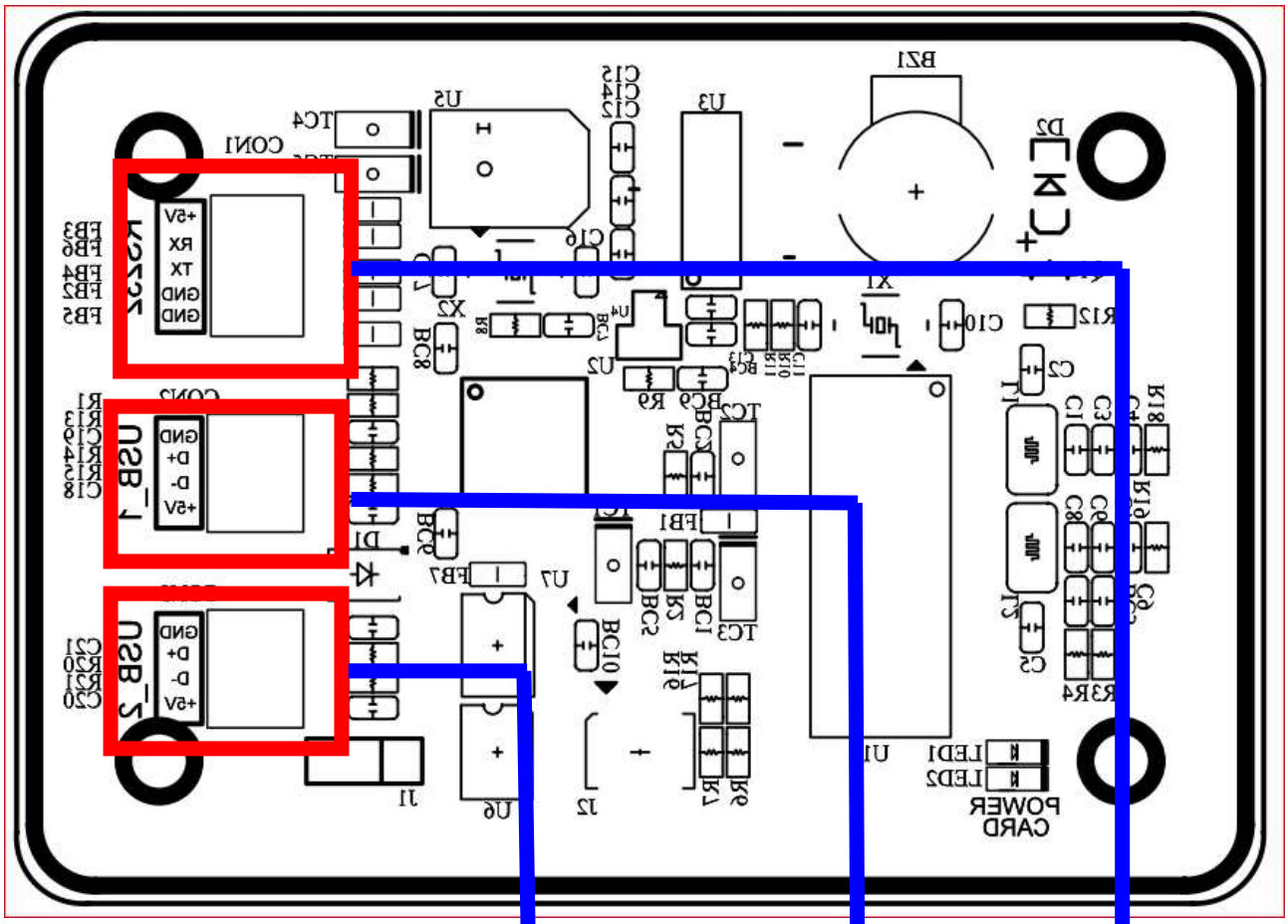
- 목 차 -

1. Specification

RF Frequency	13.56MHz
Power Supply	4.5 to 5.5V DC Operation
Supply Current	40mA @ 5V
Dimensions	70 x 50 x 6 mm
RF Protocol	ISO14443-A/B, ISO15693 Mifare Classic, Mifare UltraLight, Mifare Plus, Mifare NTAG, ICODE SLIX1, ICODE SLIX 2
Host Interface	RS232, TTL232, USB To Serial(FTDI USB Chip) USB HID Keboard
Antennna	50-ohm Internal antenna
RF Power	150mW @ 5V
Read Range	50mm internal ant
Anticollision	Support(1tags)

2. IS-3400 V3.0 구성

2.1 Connect 구분



USB To Serial
1. Vcc
2. D-
3. D+
4. Gnd

HID KeyBoard
1. Vcc
2. D-
3. D+
4. Gnd

UART RS232
1. Vcc
2. Rx
3. Tx
4. Gnd
5. Gnd

2.2 USB Driver

(1) HID USB KeyBoard

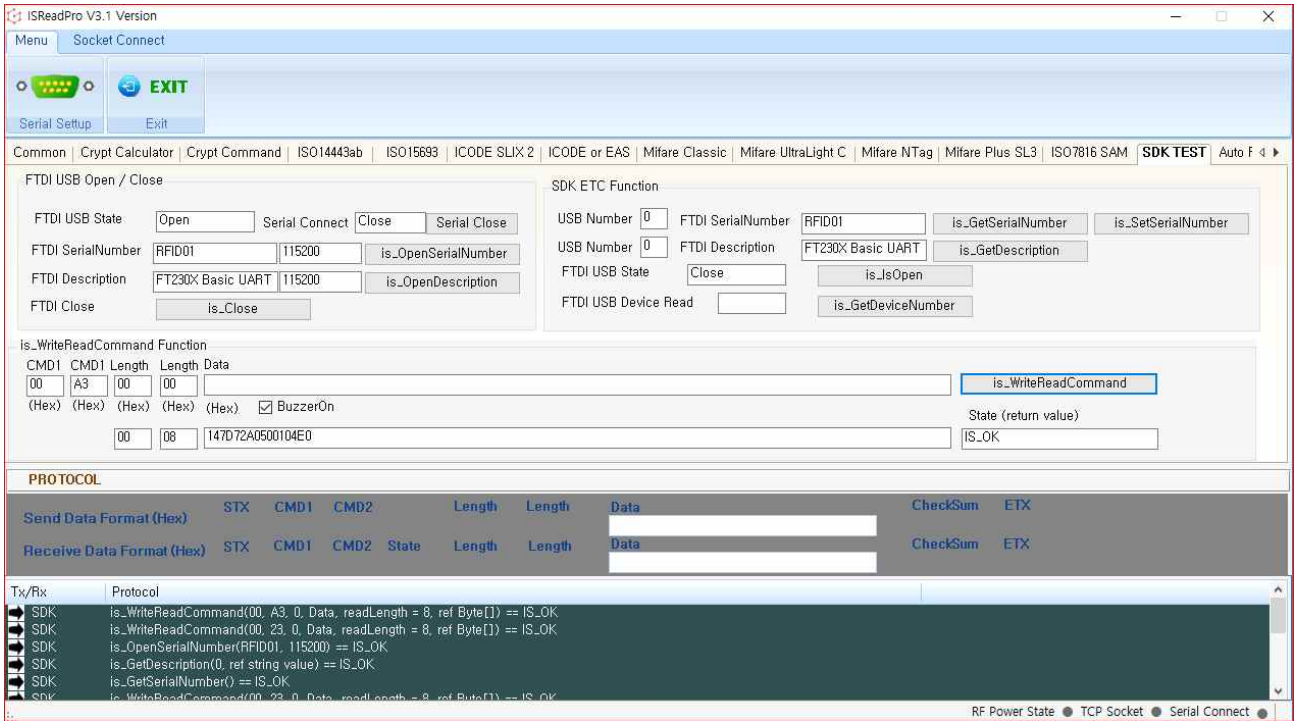
- Driver 설치가 필요 없이 자동으로 인식 됩니다.

(2) USB To Serial Driver

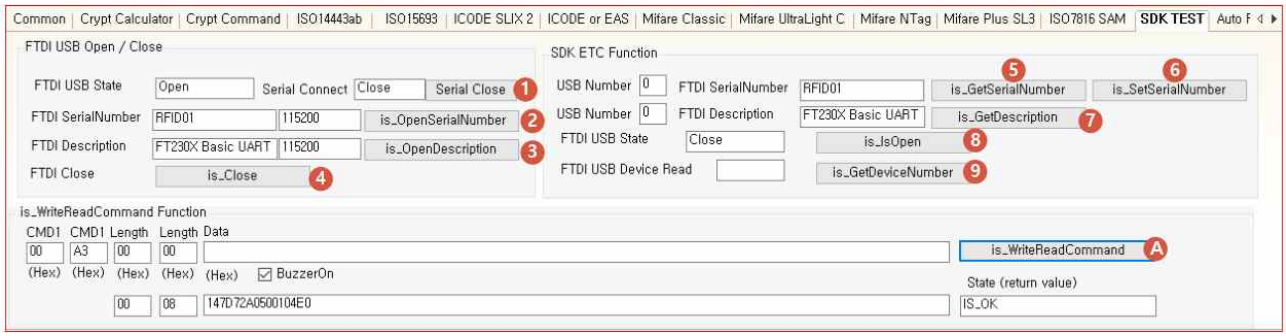
- USB Chip : FTDI230x
- 다운로드 사이트

  http://www.ftdichip.com/Drivers/VCP.htm

3. SDK TEST 탭


ISReadPro V3.1 Version
Menu
Socket Connect
Serial Setup
EXIT
Exit
Common | Crypt Calculator | Crypt Command | ISO14443ab | ISO15693 | ICODE SLIX 2 | ICODE or EAS | Mifare Classic | Mifare UltraLight C | Mifare NTag | Mifare Plus SL3 | ISO7816 SAM | SDK TEST | Auto F
FTDI USB Open / Close
FTDI USB State
Open
Serial Connect
Close
Serial Close
FTDI SerialNumber
RFID01
115200
is_OpenSerialNumber
FTDI Description
FT230X Basic UART
115200
is_OpenDescription
FTDI Close
is_Close
SDK ETC Function
USB Number
0
FTDI SerialNumber
RFID01
is_GetSerialNumber
is_SetSerialNumber
USB Number
0
FTDI Description
FT230X Basic UART
is_GetDescription
FTDI USB State
Close
is_IsOpen
FTDI USB Device Read
is_GetDeviceNumber
is_WriteReadCommand Function
CMD1 CMD1 Length Length Data
00 A3 00 00
(Hex) (Hex) (Hex) (Hex) (Hex)
BuzzerOn
is_WriteReadCommand
State (return value)
00 08
147D72A0500104E0
IS_OK
PROTOCOL
Send Data Format (Hex) STX CMD1 CMD2 Length Length Data CheckSum ETX
Receive Data Format (Hex) STX CMD1 CMD2 State Length Length Data CheckSum ETX
Tx/Rx
Protocol
SDK is_WriteReadCommand(00, A3, 0, Data, readLength = 8, ref Byte[]) == IS_OK
SDK is_WriteReadCommand(00, 23, 0, Data, readLength = 8, ref Byte[]) == IS_OK
SDK is_OpenSerialNumber(RFID01, 115200) == IS_OK
SDK is_GetDescription(0, ref string value) == IS_OK
SDK is_GetSerialNumber() == IS_OK
RF Power State
TCP Socket
Serial Connect

3.1 기본 SDK Funtion 사용법



① Serial이 ISReaderPro 3.0으로 연결 되어 있으면 SDK로 연결이 안됩니다. Serial 연결을 Close 하고 테스트 하시면 됩니다.

② is_OpenSerialNumber 함수를 테스트 합니다. FTDI SerialNumber를 ⑤에서 확인 후 SerialNumber와 통신 속도를 입력 후 연결을 시도 하세요.

③ is_OpenDescription 함수를 테스트 합니다. FTDI Description를 ⑦에서 확인 후 통신 속도를 입력 후 연결을 시도 하세요.

④ is_Close 함수 테스트, 통신 연결을 Close 합니다.

⑤ is_GetSerialNumber 함수 테스트, FTDI SerialNumber를 확인 할수 있습니다. 통신 연결이 되어 있으면 읽어 올수 없습니다. USB Number는 FTDI USB To Serial 이 연결 되어 있는 번호를 나타 냅니다.
현재 FTDI 사의 USB To Serial 칩이 몇 개 연결 되어 있는지는 ⑨ 통해 확인 할수 있습니다.
예) 2개가 연결 되어 있으면 USB Number는 0, 1입니다.

⑦ is_GetDescription 함수를 테스트, FTDI Description을 확인 할수 있습니다.

⑧ is_IsOpen 현재 연결 상태를 확인 할수 있습니다.

⑨ is_GetDeviceNumber 함수를 테스트, 현재 FTDI USB to Serial 칩으로 연결 되어 있는 개수를 읽어 옵니다.

Ⓐ is_WriteReadCommand 함수 테스트, 인자에 시리얼 통신에서 사용한 값과 동일하게 입력 하고 사용 가능 합니다. 명령 실행을 요청 하고, 응답 할때 까지 대기 하여 응답 데이터를 읽어 옵니다.
응답 시간을 짧게 사용 해야 한다면 is_WriteCommand 명령어 사용 후 스레드로 is_ReadExCommand 함수를 이용 하여 읽으시면 됩니다.

4. SDK Function

(1) is_OpenSerialNumber

- FTDI SerialNumber로 USB 포트를 오픈 연결 합니다.

```
int is_OpenSerialNumber
(
    IS_HANDLE *ftHandle,
    char *serialNumberString,
    long BaudRate
);
```

Arguments :

ftHandle : 핸들 변수

SerialNumber, : 연결할 FTDI SerialNumber

BaudRate : 연결할 BaudRate

BaudRate
4800,
9600
19200
38400
57600
115200
230400

Return Value

성공시 IS_OK = 0을 리턴 합니다.

Example

```
IS_HANDLE ftHandle = 0;
char SerialNumber[] = "RFID01"
if (is_OpenSerialNumber(&ftHandle, SerialNumber, 115200) != IS_OK)
{
  printf("USB To Serial 와통신연결실패₩n");
  return -1;
}
```

(2) is_OpenDescription

FTDI Description으로 USB 포트를 오픈 연결 합니다.

```
IS_STATUS is_OpenDescription
(
    IS_HANDLE *ftHandle,
    char *descriptionString,
    long BaudRate
)
```

Arguments :

ftHandle : 핸들 변수

Description, : 연결할 FTDI Description

Description 데이터는 "FT230X Basic UART" 입니다.

BaudRate : 연결할 BaudRate

BaudRate
4800,
9600
19200
38400
57600
115200
230400

Return Value

성공시 IS_OK = 0을 리턴 합니다.

Example

```
IS_HANDLE ftHandle = 0;
char Description [] = "FT230X Basic UART"
if (is_OpenDescription(&ftHandle, Description , 115200) != IS_OK)
{
  printf("USB To Serial 와통신연결실패₩n");
  return -1;
}
```

(3) is_Close

FTDI SerialNumber로 USB 포트를 오픈 연결 합니다.

```
IS_STATUS is_Close
(
   IS_HANDLE ftHandle
)
```

Arguments :

ftHandle : 핸들 변수

Return Value

성공시 IS_OK = 0을 리턴 합니다.

Example

```
if (is_Close(ftHandle))  == IS_OK)
{
    printf("연결을 닫습니다. “);
}
```

(4) is_GetDeviceNumber

FTDI USB To Serial 개수를 읽어 옵니다.

```
IS_STATUS is_GetDeviceNumber
(
  short *deviceNumber
)
```

Arguments :

deviceNumber : FTDI USB To Serial 연결된 개수 받아 옵니다.

Return Value

성공시 IS_OK = 0을 리턴 합니다.

Example

```
short usbnumber = 0;
if (is_GetDeviceNumber(&usbnumber) == IS_OK)
{
        printf("FTDI USB To Serial 연결된개수: %d₩n", usbnumber);
}
```

(5) is_GetDescription

FTDI USB To Serial Description 값을 읽어 옵니다.

```
IS_STATUS is_GetDescription
(
  long    usb_device_number,
  char *descriptionString
)
```

Arguments :

usb_device_number: FTDI USB To Serial 연결되어 있는 번호 0, 1, 2, 형태로 됩니다.
첫 번째 있는 FTDI USB To Serial 은 0 이 됩니다.

DescriptionReadBuffer : Description 읽을 버퍼
IS-3400 V3.0 버전은 "FT230X Basic UART" 입니다.

Return Value

성공시 IS_OK = 0을 리턴 합니다.

Example

```
char str[100];
if (is_GetDescription(0, str) == IS_OK)
{
   printf(" Description Name = %s", str);
}
```

(6) is_GetSerialNumber

FTDI USB To Serial SerialNumber 값을 읽어 옵니다.

```
IS_STATUS is_GetSerialNumber
(
    long    usb_device_number,
    char *serialNumberString
)
```

Arguments :

usb_device_number: FTDI USB To Serial 연결되어 있는 번호 0, 1, 2, 형태로 됩니다.
첫 번째 있는 FTDI USB To Serial 은 0 이 됩니다.

SerialNumberReadBuffer: FTDI USB to Serial이 램덤하게 들어 있는 값을 읽을 버퍼

Return Value

성공시 IS_OK = 0을 리턴 합니다.

Example

```
char readSerialNumber[100];
if (is_GetSerialNumber(0, readSerialNumber) == IS_OK)
{
        printf(" FTDI SerialNumber :  %s ₩n", readSerialNumber);
}
```

(7) is_GetTimeOut

FTDI USB To Serial Open완료 후 몇 타이 아웃 설정 시간을 확인 할수 있습니다..

```
IS_STATUS   is_GetTimeOut
(
   unsigned long *readTimeOut_milliseconds,
   unsigned long *writeTimeOut_milliseconds
)
```

Arguments :

readTimeOut_milliseconds : 데이터를 받을 때 까지 대기 시간

writeTimeOut_milliseconds : 명령을 보내고 응답 할 때 까지 대기 시간

Return Value

성공시 IS_OK = 0을 리턴 합니다.

Example

```
unsigned long readTimeOut = 0, writeTimeOut = 0;

if (is_GetTimeOut(ftHandle, &readTimeOut, &writeTimeOut) == IS_OK)
{
   printf("Read Timeout : %d₩n", readTimeOut);
   printf("Write Timeout : %d₩n", writeTimeOut);
}
```

(8) is_SetTimeOut

FTDI USB To Serial Open완료 후 몇 타이 아웃을 설정 할 수 있습니다.

```
IS_STATUS  is_SetTimeOut
(
   unsigned long readTimeOut_milliseconds,
   unsigned long writeTimeOut_milliseconds
)
```

Arguments :

readTimeOut_milliseconds : 데이터를 받을 때 까지 대기 시간

writeTimeOut_milliseconds : 명령을 보내고 응답 할 때 까지 대기 시간

Return Value

성공시 IS_OK = 0을 리턴 합니다.

참조 : is_OpenSerialNumber, is_OpenDescription 함수로 연결 후 사용 가능 합니다.
is_WriteCommand, is_WriteReadCommand, is_ReadCommand, is_ReadExCommand
응답 대기 시간, 전송대기 시간을 설정 할수 있습니다.
기본 값 500mS로 설정 되어 있습니다.

Example

```
readTimeOut = 700;
writeTimeOut = 300;
if (is_SetTimeOut(ftHandle, readTimeOut, writeTimeOut) == IS_OK)
{
   printf("설정완료₩n");
}
```

(9) is_GetCOMPort

FTDI USB To Serial Open 후 사용 가능 하며, 현재 Open한 ComPort를 읽어 옵니다.

```
IS_STATUS   is_GetCOMPort
(
    IS_HANDLE ftHandle,
    unsigned   long *portNumber
)
```

Arguments :

ftHandle : 핸들 변수

portNumber: ComPort 번호를 읽어 옵니다.

Return Value

성공시 IS_OK = 0을 리턴 합니다.

Example

```
unsigned long portNumber;
if (is_GetCOMPort(ftHandle, &portNumber) == IS_OK)
{
    printf("COM Port : %d₩n", portNumber);
}
```

(10) is_GetCOMPort_NoConnect

FTDI USB To Serial Open 후 사용 가능 하며, 현재 Open한 ComPort를 읽어 옵니다.

```
IS_STATUS   is_GetCOMPort_NoConnect
(
    long usb_device_number,
    unsigned   long *portNumber
)
```

Arguments :

usb_device_number: FTDI USB To Serial 연결되어 있는 번호 0, 1, 2, 형태로 됩니다.
첫 번째 있는 FTDI USB To Serial 은 0 이 됩니다.

portNumber: ComPort Nubmer Value.

Return Value

성공시 IS_OK = 0을 리턴 합니다.

Example

```
unsigned long portNumber;

if (is_GetCOMPort_NoConnect(0, &portNumber) == IS_OK)
{
    printf("COM Port : %d₩n", portNumber);
}
```

(11) is_WriteReadCommand

RFID 리더기의 명령어를 전송 하고 데이터를 읽어 오는 함수입니다.

```
IS_STATUS   is_WriteReadCommand
(
   IS_HANDLE ftHandle,
   unsigned char cmd1,
   unsigned char cmd2,
   unsigned short writeLength,
   unsigned char *writeData,
   unsigned short *readLength,
   unsigned char *readData
)
```

Arguments :

ftHandle : 핸들 변수
cmd1 : 상위 명령어
cmd2 : 하위 명령어
writeLength : 명령어 데이터 길이
writeData : 명령어 데이터
readLength : 읽어온 데이터 길이
readData : 읽어올 데이터 버퍼

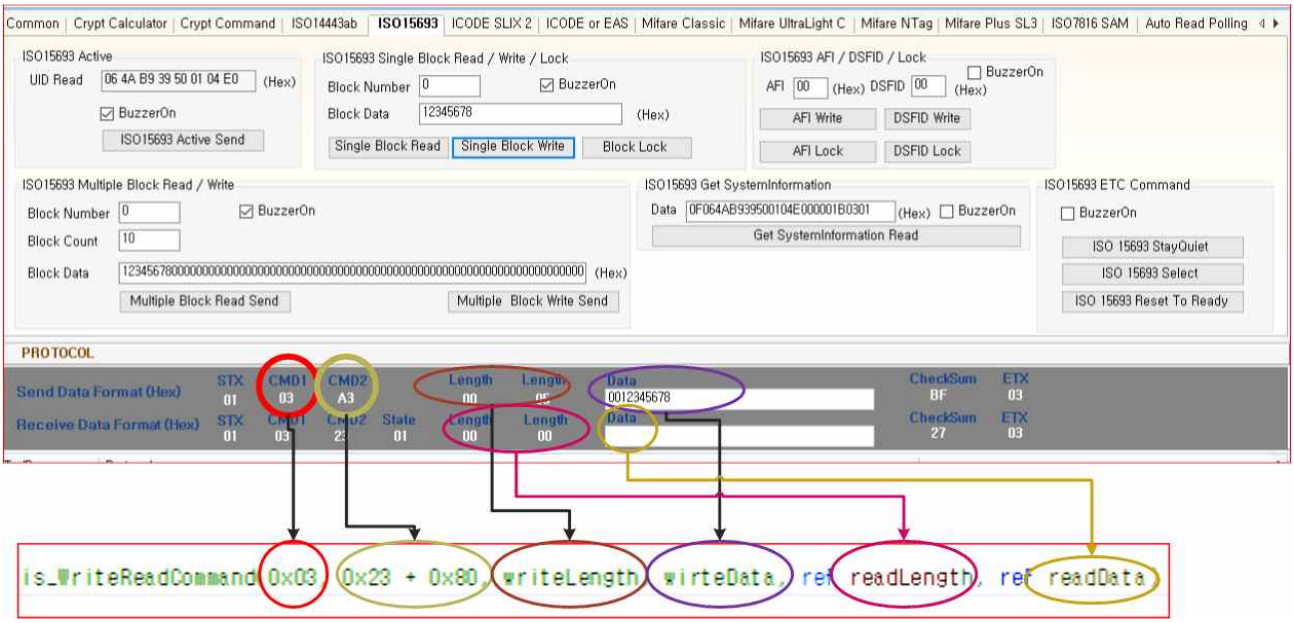
Return Value

성공시 IS_OK = 0을 리턴 합니다.

Example

```
if (is_WriteReadCommand(ftHandle, CM1_COMMON, CMD2_COMMON_ALL_UID_READ +
                    BUZZER_ON, writeLength, wirteData, &readLength, readData) == IS_OK)
{
    int i;
    printf("UID : ");
    for (i = 0; i < readLength; i++)
    {
        printf("%x ", readData[i]);
    }
    printf("₩n");
}
```

(12) is_WriteReadCommand 명령어 구조



(13) is_WriteCommand

RFID 리더기의 명령어를 전송 하는 함수입니다.

```
IS_STATUS    is_WriteCommand
(
     IS_HANDLE ftHandle,
     unsigned char cmd1,
     unsigned char cmd2,
     unsigned short writeLength,
     unsigned char *writeData
)
```

Arguments :

ftHandle : 핸들 변수
cmd1 : 상위 명령어
cmd2 : 하위 명령어
writeLength : 명령어 데이터 길이
writeData : 명령어 데이터

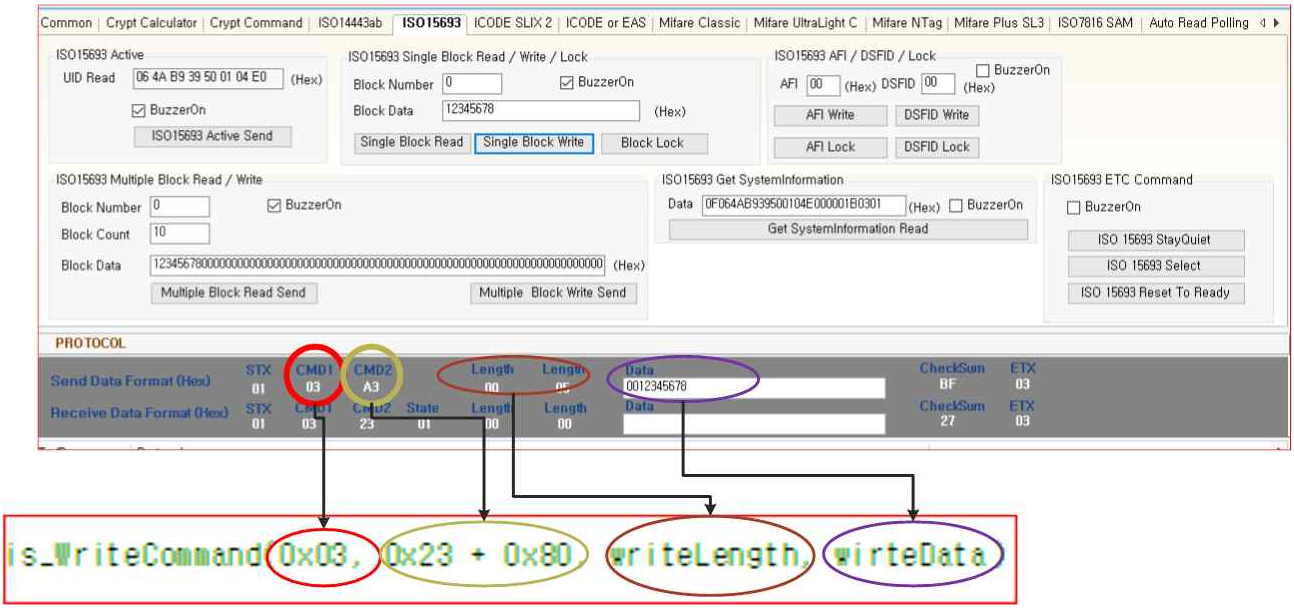
Return Value

성공시 IS_OK = 0을 리턴 합니다.

Example

```
if (is_WriteCommand(ftHandle, CM1_COMMON, CMD2_COMMON_ALL_UID_READ +
                                        BUZZER_ON, writeLength, wirteData) == IS_OK)
{
      if (is_ReadCommand(ftHandle, &readLength, readData) == IS_OK)
      {
            int i;
            printf("UID : ");
            for (i = 0; i < readLength; i++)
            {
                printf("%x ", readData[i]);
            }
            printf("₩n");
      }
}
```

(14) is_WriteCommand 명령어 구조



(15) is_ReadCommand

RFID 리더기의 데이터를 읽어 오는 함수입니다.

```
IS_STATUS   is_ReadCommand
(
    IS_HANDLE ftHandle,
    unsigned short *readLength,
    unsigned char *readData
)
```

Arguments :

ftHandle : 핸들 변수

readlength : 읽어온 데이터 길이

readData : 읽어올 데이터 버퍼

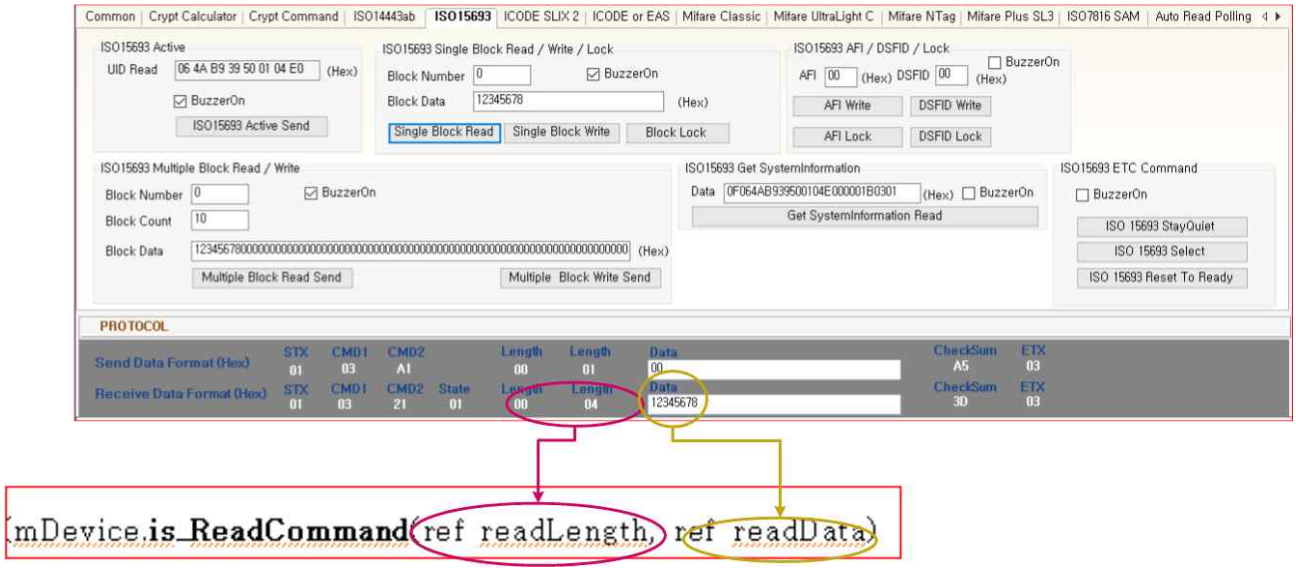
Return Value

성공시 IS_OK = 0을 리턴 합니다.

Example

```
if (is_WriteCommand(ftHandle, CM1_COMMON, CMD2_COMMON_ALL_UID_READ +
                                        BUZZER_ON, writeLength, wirteData) == IS_OK)
{
    if (is_ReadCommand(ftHandle, &readLength, readData) == IS_OK)
    {
        int i;
        printf("UID : ");
        for (i = 0; i < readLength; i++)
        {
            printf("%x ", readData[i]);
        }
        printf("₩n");
    }
}
```

(16) is_ReadCommand 명령어 구조



(17) is_ReadExCommand

RFID 리더기의 데이터를 읽어 오는 함수입니다.

```
IS_STATUS    is_ReadExCommand
(
    IS_HANDLE ftHandle,
    unsigned char *cmd1,
    unsigned char *cmd2,
    unsigned short *length,
    unsigned char *readData
)
```

Arguments :

ftHandle : 핸들 변수

cmd1 : 상위 명령어를 받습니다.

cmd2 : 하위 명령어를 받습니다.

readlength : 읽어온 데이터 길이

readData : 읽어올 데이터 버퍼

Return Value

성공시 IS_OK = 0을 리턴 합니다.

Example

```
if (is_WriteCommand(ftHandle, CM1_COMMON, CMD2_COMMON_ALL_UID_READ +
                                        BUZZER_ON, writeLength, wirteData) == IS_OK)
{
    //스레드작업으로처리할때편리하게사용할수있습니다.
    if (is_ReadExCommand(ftHandle, &cmd1, &cmd2, &readLength, readData) == IS_OK)
    {
        int i;
        printf("Command1 = %x, Command2 =  %x₩n", cmd1, cmd2);
        printf("UID : ");
        for (i = 0; i < readLength; i++)
        {
            printf("%x ", readData[i]);
        }
        printf("₩n");
    }
}
```

(18) is_ReadExCommand 명령어 구조