

# **BILL VALIDATOR**

**(지폐식별기)**

## **RS232 Communication Programmer's Guide**

Version 1.5

(2011. 10. 17)



442-832 B/D507 Hansin 944-2 Ingyedong, Paldal-Gu, Suwon-City Kyungki-Do, Korea

Tel : +82-31-2361061-3. Fax : +82-31-236-1064

--- CONTENTS ---

1. RS232 통신 Format
2. 내부 Register
3. Command 세부 설명
4. 예제 Firmware Source
5. PC Demo Program 설명
6. Data 송·수신 Program시 주의 사항
7. Release Note

## 1. RS232 통신 Format

### 1) 통신 Format :

#### ㄱ) 일반 Head 형 :

- 통신속도 : 9600,n,8,1 또는 4800,n,8,1
- 통신방식 : Full Duplex 또는 Half Duplex

#### ㄴ) 수직형 :

- 통신속도 : 4800,n,8,1
- 통신방식 : Half Duplex

### 2) 통신 Packet : 총 5Byte로 구성

BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5
--------	--------	--------	--------	--------

#### - BYTE 1

'\$' : Packet의 시작을 의미

#### - BYTE 2, BYTE 3, BYTE 4 : Command 세부 설명 참조

#### - BYTE 5

BYTE2, 3, 4의 합 ( CheckSum )

(ex : BYTE 5 = (unsigned char)( BYTE 2 + BYTE 3 + BYTE 4 )

## 2. 지폐식별기 내부 Register

1) CONFIG (Read/Write) : 동작과 관련된 Configuration 표시

BIT 8	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1
-------	-------	-------	-------	-------	-------	-------	-------

- BIT 8, BIT 7 : Reserved
- BIT 6 : Event발생시 TX
  - 0 : Data 요청시에만 지폐식별기가 Data를 송출 (Half Duplex Mode)
  - 1 : 지폐식별기의 동작 상태가 변동 되었을 때  
그 내용을 자동으로 송출 (Full Duplex Mode)  
→ 일반 Head형 Model에 한해서만 가능
- BIT 5 : 지폐인식 후 자동 Stack
  - 0 : 인식 완료 후 입수 또는 반환 명령이 올 때 까지 기다림.  
ESCROW Model(Solenoid적용Model 또는 수직형Model)에 한함
  - 1 : 인식 완료 후 진권일 경우 자동으로 Stack 처리 함.
- BIT 4 : 50,000원 입수 여부 설정
  - 0 : 50,000원 입수 불가
  - 1 : 50,000원 입수 가능
- BIT 3 : 10,000원 입수 여부 설정
  - 0 : 10,000원 입수 불가
  - 1 : 10,000원 입수 가능
- BIT 2 : 5,000원 입수 여부 설정
  - 0 : 5,000원 입수 불가
  - 1 : 5,000원 입수 가능
- BIT 1 : 1,000원 입수 여부 설정
  - 0 : 1,000원 입수 불가
  - 1 : 1,000원 입수 가능

2) BillData (Read only) : 입수된 지폐의 권종을 표시

상수명	Dec	내용
BILL_0	0	입수된 지폐 없음
BILL_1000	1	1,000원 권 입수됨
BILL_5000	5	5,000원 권 입수됨
BILL_10000	10	10,000원 권 입수됨
BILL_50000	50	50,000원 권 입수됨

### 3) ActiveStatus (Read/Write) : 지폐식별기의 현 동작 상태를 표시

No	상수명	Hex	용도	내용
1	RESET_WAIT	0x0	RX	Reset후 Initial 동작중
2	WAIT	0x01	RX	Initial후 대기 중(입수금지상태)
3	STARTWAIT	0x02	RX	입수 가능 상태
4	RECOGNITION_RETURN	0x03	RX	인식작업 중 오류로 인한 반환 작업 중
5	RECOGNITION_WAIT	0x04	RX	인식작업 중
6	RECOGNITION_END	0x05	RX	인식완료 후 대기 중(입수금지상태)
7	RETURN_START	0x06	TX	인식완료 대기상태에서 반환 명령
8	RETURN_WAIT	0x07	RX	반환동작 중
9	RETURN_END	0x08	RX	반환동작 완료 후 대기 중(입수금지상태)
10	STACK_START	0x09	TX	인식완료 대기상태에서 Stack 명령
11	STACK_WAIT	0x0A	RX	Stack 동작 중
12	STACK_END	0x0B	RX	Stack 동작 완료 후 대기 중(입수금지상태)
13	ERROR_WAIT	0x0C	RX	동작 Error로 인한 대기 중(입수금지상태)
14	INSERT_ENABLE	0x0D	TX	입수금지 상태에서 입수 가능 명령
15	INSERT_DISABLE	0x0E	TX	입수가능 상태에서 입수 금지 명령
16	STACK_OPENED	0x10	RX	Stack이 Open된 상태(입수금지상태) → 수직형 Model에 한함
17	ForceStack_WAIT	0x11	RX	강제 입수 동작 중 → 일반Head형에서 지폐걸림시 일정시간후 강제입수 하는 Mode(Dip Switch설정)에 한함.
18	ForceStack_END	0x12	RX	강제 입수 완료 후 대기 중(입수금지 상태)

### 4) ErrorCode (Read only)

: ActiveStatus가 'ERROR\_WAIT' 일때 Error내용을 표시

상수명	Dec	내용
ERROR_NO	0	Error없음
ERROR_START	1	Start Sensor 관련 Error
ERROR_SHUTTER	2	Shutter Sensor 관련 Error
ERROR_END	3	End Sensor 관련 Error
ERROR_MOTOR	4	이송 Motor 관련 Error
ERROR_STACK	5	Stack Motor 관련 Error
ERROR_DETECT_STEAL	9	불손한 의도가 시도되어 1분간 입수금지
ERROR_SENSOR_OPTIC1	11	인식 Sensor1 관련 Error
ERROR_SENSOR_OPTIC2	12	인식 Sensor2 관련 Error
ERROR_SENSOR_OPTIC3	13	인식 Sensor3 관련 Error
ERROR_SENSOR_OPTIC4	14	인식 Sensor4 관련 Error
ERROR_SENSOR_OPTIC5	15	인식 Sensor5 관련 Error
ERROR_SENSOR_OPTIC6	16	인식 Sensor6 관련 Error
ERROR_SENSOR_OPTIC7	17	인식 Sensor7 관련 Error
ERROR_SENSOR_OPTIC8	18	인식 Sensor8 관련 Error

### 3. Command 세부 설명

#### 1) Get 명령

항목	TX			응답			내용	설 명
	BYTE2	BYTE3	BYTE4	BYTE2	BYTE3	BYTE 4		
ActiveStatus 읽기	'G'	'A'	'?'	'g'	'a'	<i>ActiveStatus</i>	내부 Register 설명 참조	현재 지폐식별기의 동작 상태를 확인합니다.
BillData 읽기	'G'	'B'	'?'	'g'	'b'	<i>BillData</i>		입수된 지폐의 권종을 표시합니다. 주1) 인식완료상태 이후에 유효합니다. 주2) Stack완료상태에서 입수가능상태로 설정을 변경하면 BillData는 0으로 Clear함.
CONFIG 읽기	'G'	'C'	'?'	'g'	'c'	<i>CONFIG</i>		식별기 설정 상태를 확인합니다.
ErrorCode 읽기	'G'	'E'	'?'	'g'	'e'	<i>ErrorCode</i>		Error상태시 Error의 원인을 표시합니다.

#### 2) Set 명령

항목	TX			응답			설 명	비고
	BYTE2	BYTE3	BYTE4	BYTE2	BYTE3	BYTE4		
입수 가능 설정 하기	'S'	'A'	0x0D (INSERT_ENABLE)	'0'	'K'	'a'	입수가능으로 설정완료	
				'N'	'G'	'a'	입수가능으로 설정 할수 없음.	
0x0E (INSERT_DISABLE)			'0'	'K'	'a'	입수금지로 설정완료		
			'N'	'G'	'a'	입수금지로 설정 할수 없음.		
지폐 반환 하기			0x06 (RETURN_START)	'0'	'K'	'a'	인식완료상태에서 반환동작 Start	ESCROW Model에 한함. (수직형Model포함)
				'N'	'G'	'a'	반환동작을 할 수 없음	
지폐 입수 하기			0x09 (STACK_START)	'0'	'K'	'a'	인식완료상태에서 Stack동작 Start	ESCROW Model에 한함. (수직형Model포함)
				'N'	'G'	'a'	Stack동작을 할수 없음	
CONFIG 설정 하기	'S'	'C'	CONFIG	'0'	'K'	'c'	Config 설정 완료	
				'N'	'G'	'c'	Config 설정을 할수 없음	

3) Connection 명령 : 지폐식별기와 통신 상태를 Check하기 위해 사용합니다.

항목	TX			응답			설 명	비고
	BYTE2	BYTE3	BYTE4	BYTE2	BYTE3	BYTE4		
Connection	'H'	'i'	'?'	'm'	'e'	'!'	정상적 통신 여부를 확인합니다.	

4) Event TX 수신 : CONFIG Register의 Bit60이 Set되었을 경우 유효합니다.

항목	RX			TX			설 명	비고
	BYTE2	BYTE3	BYTE4	BYTE2	BYTE3	BYTE4		
Event TX	'E'	'S'	Active Status Reg.	'e'	's'	Active Status Reg.	식별기의 동작상태 변경시 송신되며 수신후 응답을 하지 않으면, 식별기에서는 400msec 간격으로 계속 송신을 합니다.	

5) 인식기의 F/W Veriosn 정보 :

항목	RX			TX			설 명	비고
	BYTE2	BYTE3	BYTE4	BYTE2	BYTE3	BYTE4		
Version Check	'G'	'V'	'?'	'v'	Major Version	Minor Version	Byte3 : Major Version Byte4 : Minor Version	

## 4. 예제 Firmware Source

- . Device = PIC16F690
- . Compiler = Hi-Tech C

```
#include "pic.h"

#define BufferSize 20

#define RESET_WAIT          0
#define WAIT                1
#define STARTWAIT          2
#define RECOGNITION_RETURN  3
#define RECOGNITION_WAIT    4
#define RECOGNITION_END     5
#define RETURN_START        6
#define RETURN_WAIT         7
#define RETURN_END          8
#define STACK_START         9
#define STACK_WAIT          10
#define STACK_END           11
#define ERROR_WAIT          12
#define INSERT_ENABLE       13
#define INSERT_DISABLE      14
#define STACK_OPENED        16
#define ForceStack_WAIT     17
#define ForceStack_END      18

unsigned char bank2 RS232RXBuffer[BufferSize];
unsigned char bank2 RS232TXBuffer[BufferSize];
unsigned char bank2 RS232_RBuffer[6];
unsigned char bank2 RS232RXWRPtr, RS232TXWRPtr, RS232TXRDPtr, RS232RXRDPtr;
unsigned char bank2 OldTXData1, OldTXData2, OldTXData3;
unsigned char RS23Mode, Time_RS232Receive, Config, ActiveStatus, BillData, ErrorCode;
bit bRS232BoardRXIn, bWaitCode, bRS232Initial;

//*****//
//          Interrupter Service          //
//*****//

//--Interrupter Service Timer 0--//
void ISR_Timer0(void)
{
    Time_RS232Receive++;          //8msec Interval
}

//--Interrupter Service UART RX--//
void ISR_RX(void)
{
    RS232RXBuffer[RS232RXWRPtr] = RCREG;
    if(++RS232RXWRPtr >= BufferSizeRX) RS232RXWRPtr = 0;
}

//--Interrupter Service UART TX---//
void ISR_TX(void)
{
    if(RS232TXWRPtr != RS232TXRDPtr){
        TXREG = RS232TXBuffer[RS232TXRDPtr++];
        if(RS232TXRDPtr >= BufferSizeTX) RS232TXRDPtr=0;
    }
    else{
        if(TRMT) TXEN=0;          // tx buffer empty
    }
}
```



```

//*****//
//          Main Routine          //
//*****//

void main(void)
{
    ResetInitial();    // H/W Port & System Reg. Initial
    Uart_Setup();      // rs232 setup

    while(1){
        RS232_Service();
    }
}

//*****//
//          Serial Sub-Routine    //
//*****//

void RS232_TXByte(unsigned char data)
{
    RS232_TXBuffer[RS232_TXWRPtr++] = data;
    if(RS232_TXWRPtr >= BufferSizeTX) RS232_TXWRPtr = 0;
}

void Serial_TXStart(unsigned char data1, unsigned char data2, unsigned char data3)
{
    RS232_TXByte('$');
    RS232_TXByte(data1);
    RS232_TXByte(data2);
    RS232_TXByte(data3);
    RS232_TXByte( (unsigned char)(data1+data2+data3) );
    if(TXEN==0) TXIF=0;
    TXEN=1;

    OldTXData1 = data1;          // 응답이 없을 경우 재 송신을 위해 송신 Data저장
    OldTXData2 = data2;
    OldTXData3 = data3;
    bWaitCode = 1;
    Time_RS232Receive = 0;
}

void RS232Receive(void)
{
    unsigned char out;

    out = 0;
    do{
        if( RS232_RXRDPtr != RS232_RXWRPtr ){
            RS232_RBuffer[0] = RS232_RBuffer[1];
            RS232_RBuffer[1] = RS232_RBuffer[2];
            RS232_RBuffer[2] = RS232_RBuffer[3];
            RS232_RBuffer[3] = RS232_RBuffer[4];
            RS232_RBuffer[4] = RS232_RXBuffer[RS232_RXRDPtr];
            RS232_RXRDPtr = (RS232_RXRDPtr+1) % BufferSize;

            if( (RS232_RBuffer[0] == '$') && (RS232_RBuffer[4] ==
                ((unsigned char)(RS232_RBuffer[1]+RS232_RBuffer[2]+RS232_RBuffer[3])) ) ){
                out = 1;
                bRS232BoardRXIn = 1;
            }
        }
        else out = 1;
    }while(out == 0);
}

```

```

void RS232_Service(void)
{
    if( bWaitCode != 0 ){
        if( Time_RS232Receive > 50 ){ //400msec check하여 미 응답시 재 송신처리
            Serial_TXStart(OldTXData1, OldTXData2, OldTXData3);
            Time_RS232Receive = 0;
        }
    }

    RS232Receive();
    if( bRS232BoardRXIn ){
        bRS232BoardRXIn = 0;
        if( (RS232_RBuffer[1] == 'm') && (RS232_RBuffer[2] == 'e') ){
            if(!bRS232Initial){ // 지폐식별기 Initial 완료 여부
                Serial_TXStart('S', 'A', INSERT_DISABLE); // 입수 금지로 설정
            }
        }
        else{
            if( (RS232_RBuffer[1] == 'E') && (RS232_RBuffer[2] == 'S') ){ //event rx receive
                ActiveStatus = RS232_RBuffer[3];
                Serial_TXStart('e', 's', ActiveStatus); //ACK
                bWaitCode = 0; //응답필요 없음
            }
            else{
                if( RS232_RBuffer[1] == 'g' ){ //Get명령에 대한 응답Service
                    bWaitCode = 0; //응답 도착-> Clear
                    switch( RS232_RBuffer[2] ){
                        case 'a':
                            ActiveStatus = RS232_RBuffer[3];
                            switch(ActiveStatus){
                                case WAIT:
                                    Serial_TXStart('S', 'A', INSERT_ENABLE);
                                    break;
                                case STACK_END:
                                    Serial_TXStart('G', 'B', '?'); //지폐확인
                                    break;
                                case RECOGNITION_END: // 인식완료후 Stack
                                    Serial_TXStart('S', 'A', STACK_START);
                                    break;
                                case RETURN_END:
                                case RECOGNITION_RETURN:
                                    break;
                                case STARTWAIT:
                                case RESET_WAIT:
                                case STACK_WAIT:
                                case RETURN_WAIT:
                                case RECOGNITION_WAIT:
                                case ERROR_WAIT:
                                    break;
                            }
                        break;
                        case 'b':
                            BillData = RS232_RBuffer[3];

                            // insert Service Code
                            Serial_TXStart('S', 'A', INSERT_ENABLE);
                            break;
                        case 'c':
                            Config = RS232_RBuffer[3];

                            // insert Service Code
                            break;
                        case 'e':
                            ErrorCode = RS232_RBuffer[3];

                            // insert Service Code
                            break;
                    }
                }
            }
        }
    }
}

```

```

else{
    if( (RS232_RBuffer[1] == '0') && (RS232_RBuffer[2] == 'K') ){
        //set명령에 대한 응답
        if( (RS232_RBuffer[3] == 'c') || (RS232_RBuffer[3] == 'a') ){
            bWaitCode = 0;           //응답 도착-> Clear
            if(!bRS232Initial){
                // Config Reg. Set
                Config = 0x27; // Full Duplex, 3권종 입수가능
                Serial_TXStart('S', 'C', Config);

                // 지폐식별기 Initial End
                bRS232Initial = 1;
            }

            // insert Service Code
        }
    }
}
else{
    if( (RS232_RBuffer[1] == 'N') && (RS232_RBuffer[2] == 'G') ){
        //set명령에 대한 응답
        if( (RS232_RBuffer[3] == 'c') ||
            (RS232_RBuffer[3] == 'a') ){
            bWaitCode = 0;           //응답 도착-> Clear

            // insert Service Code
        }
    }
}
}
}
}
}
}
}
}
}
}
}

```

## 5. PC Demo Program 설명

**통신 설정**

COM Port: COM4    9600 bps    RS232 Port Change    종료

**지폐식별기**

Connection !!!    Version Check

**지폐 인식기 Config**

- ☒ Event발생시 Data TX
- ☒ 지폐인식후 자동 Stack
- ☐ 50,000원 입수 가능
- ☒ 10,000원 입수 가능
- ☒ 5,000원 입수 가능
- ☒ 1,000원 입수 가능

Read Config    Set Config

**지폐 인식기 상태**

입수된 지폐는  
1,000원권 입니다.  
: Stack 관련 Error

Read Status    지폐종류 확인    오류 내용

**입수된 지폐 종류**

입수가능    입수    반환

**RS232 Message**

TX  
0x24,0x47,0x45,0x3f,0xcb ( \$ G E ? ? )

RX  
0x24,0x67,0x62,0x1,0xca ( \$ g b r ? )

천원권    2  
오천원권    0  
만원권    0  
오만원권    0

Manual

### 1) 통신 설정 창

- PC환경에 맞게 COM Port를 설정합니다.
- 지폐식별기의 통신속도에 맞게 속도를 설정합니다.
- "RS232 Port Change"를 Click하여 Port를 Open합니다.

### 2) RS232 Message 창

- 실제로 지폐 식별기 통신시 TX 및 RX의 Packet Data를 표시합니다.
- 표시되는 Data는 ASCII이며 괄호안에 표시되는 것이 16진수 값입니다.

### 3) 지폐 식별기 창

- 각 버튼을 Click하여 동작을 확인 하십시오

## 6. Data 송·수신 Program시 주의 사항

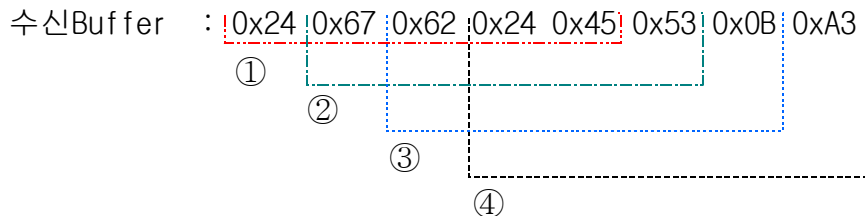
### 1) Data 송신 시

: GET, SET 또는 Connection 명령을 송신 후,  
일정 시간(500msec)내에 응답이 없으면 명령을 재 송신하여 주십시오.  
또한 재 송신에도 응답이 없으면 재송신을 2회 추가 송신하여 응답을  
확인하여 주십시오. 만약 3회 재송신 후에도 응답이 없으면  
"지폐인식기 Error"로 처리하여 적절한 조치를 취하여 주십시오

### 2) Data 수신 시

- 모든 Data통신은 항상 5Byte입니다. 따라서 수신된 Data가 5Byte 이상  
이더라도 5byte씩 수신Data를 확인하여 적절한 조치를 취하여 주십시오.

적용 예)



Serial Data 수신처리

- ①항 CheckSum 오류 (Byte2,3,4의 합이 Byte5와 같아야 함)
- 수신Data 오류로 다음 수신 Data 확인
- ②항 STX 오류 (첫번째 Byte는 '\$' 이어야 함)
- 수신Data 오류로 다음 수신 Data 확인
- ③항 STX 오류 (첫번째 Byte는 '\$' 이어야 함)
- 수신Data 오류로 다음 수신 Data 확인
- ④항 Data 수신 OK ( \$ES(0x0b) )
- Data송신 : 0x24, 0x65, 0x73, 0x0B, 0xE3 ( \$es(0x0b) )
- 다음 수신 Data 확인

- 본 통신 Protocol은 command에 대해서는 항상 응답이 있어야 합니다.  
따라서 Event TX에 의한 Data 수신시 반드시 해당 응답을 해 주셔야 합니다.

- 내부 Register의 Data 수신시

그 Data값이 각 Register의 Data 범위를 초과시에는  
Data 수신 오류로 처리하도록 합니다.

Register명	최소값	최대값
CONFIG	0	0xff
BillData	0	0x32
ActiveStatus	0	0x10
ErrorCode	0	0x12

## 7. Release Note

### - Ver 1.5

- (1) 일반 Head 형에서 지폐걸림시 일정시간 후 강제입수 처리하는 모드에서 강제입수 후 현재 상태를 파악 할 수 있는 통신 Protocol 추가.  
→ ActiveStatus의 내용 추가 : 표 2-3)의 17번, 18번 항목 추가
- (2) 배포용 PC DEMO Program변경(Ver 2.1) 및 PC Demo Source 변경  
→ 강제입수 관련 통신 Protocol 및 출력 Message 추가
- (3) 적용 Firm-Ware Version
  - (ㄱ) 지폐 인식 Main Board 4권종(Ver 4.6), 3권종(Ver 4.5)
  - (ㄴ) RS232 Interface Board : Ver 1.1 이상

### - Ver 1.4

- (1) 수직형 Model 추가에 따른 사양 추가
- (2) 배포용 PC DEMO Program변경 : Ver 2.0  
→ 수직형Model에서는 Event TX 설정이 불가함에 따라 기존 PC Program의 Auto 기능이 정상동작 하지 않는 것을 수정함.

### - Ver 1.3

- (1) "6. Data 송·수신 Program시 주의 사항" 추가  
→ 통신 Program 개발자들의 공통적인 질문사항이나 오류들을 정리하여 추가함.

### - Ver 1.2

- (1) 2-4) 표 수정  
Error Code 추가 : Code 9번  
→ 내용 "ERROR\_DETECT\_STEAL:불손한 의도가 시도되어 1분간 입수금지"
- (2) 배포용 PC DEMO Program변경  
→ Error발생시 Error내용 확인을 위한 Button 추가  
→ Auto Mode 동작상태에서 Stack Error(Stack Full등)시 입수된 금액을 표시하도록 수정  
→ Auto Mode 동작상태에서 Error 내용을 표시하도록 수정
- (3) 적용 Firm-Ware Version
  - (ㄱ) 지폐 인식 Main Board :
    - 4권종 : Ver 2.7 이상
    - 3권종 : Ver 2.6 이상
  - (ㄴ) RS232 Interface Board : Ver 1.0 이상
- (4) 3-5) 표 추가  
→ Firmware Version정보 확인하기 위한 Command 표기 누락

### - Ver 1.1

- 3-4)표 수정, 송수신 Data BYTE4의 내용을 CONFIG Reg.에서 ActiveStatus Reg. 변경수정

### - Ver 1.0 : First Version