

<b>Doc No</b>	<b>KHX-21XX SPECIFICATIONS</b>	<b>REV</b>	<b>PAGE</b>	<b>DATE</b>
			G	1 OF 26

# **Compact Card Dispenser**

<b>Doc No</b>	<b>KHX-21XX SPECIFICATIONS</b>	<b>REV</b>	<b>PAGE</b>	<b>DATE</b>
		G	2 OF 26	2012. 10. 24.

● **REVISION HISTORY**

<b>CHECK</b>	<b>DATE</b>	<b>DESCRIPTION</b>	<b>REV.</b>	<b>PAGE</b>
1	2009. 01. 19	First Edition	A	16
2	2009. 07. 16	Mechanical drawing is modified. Card issuing length is changed.	B	16
3	2009.12.24	TTL SPEC is added.	C	20
4	2010.01.07	TTL Interface connector is modified.	D	24
5	2010.02.06	TTL BACK SENSOR signal is modified.	E	24
6	2012.02.13	Add the FRONT SENSOR CHECK ISSUE(0x55) command	F	25
7	2012.10.24	Add the Shutter type	G	15

<b>Doc No</b>	<b>KHX-21XX SPECIFICATIONS</b>	<b>REV</b>	<b>PAGE</b>	<b>DATE</b>
		G	3 OF 26	2012. 10. 24.

# **C O N T E N T S**

1. Overview
2. Features
3. System Block Diagram
4. Environmental Requirements
5. Specification
6. DC Power Connector
7. Interface
8. Technical Drawing
9. RS232C Interface
10. Card Status

Doc No	KHX-21XX SPECIFICATIONS	REV	PAGE	DATE
		G	4 OF 26	2012. 10. 24.

## 1. Overview

All the processes and operations of KHT-21XX are monitored by its intelligent Microprocessor, which makes itself self-recover function from faulty running.

There are 2 types of Interface for KHX-21XX Series, TTL Interface and RS-232C Interface, which can be integrated as User requires .

## 2. Features

2.1. Card thickness dispensable can be adjusted easily.

2.2. RS232C Interface

A. Baud Rate: changeable (9,600 BPS ↔ 19,200BPS)

B. Can change position of card (one way direction allowed)

C. With self-diagnosis function.

D. Easy to control

2.3. TTL Interface

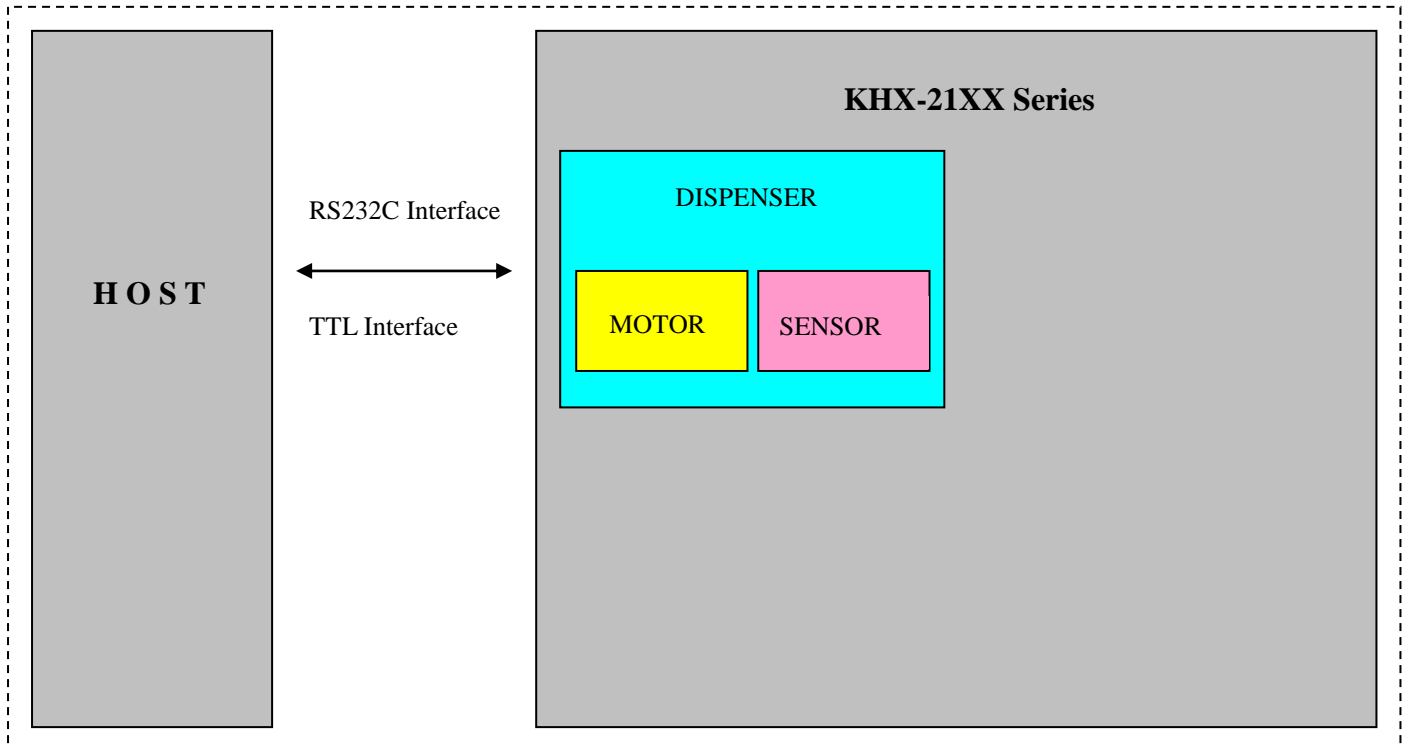
2.4. KHX-21XX series are applied and integrated to following products and systems;

- Prepaid card vending machine
- ID card issuing machine
- Parking card vending machine
- Payphone card vending machine
- Automatic card issuing machine
- Ticketing vending machine
- And more

2.5. Intelligent monitoring all the process cards empty and card low level warning function with its own microprocessor

<b>Doc No</b>	<b>KHX-21XX SPECIFICATIONS</b>	<b>REV</b>	<b>PAGE</b>	<b>DATE</b>
		G	5 OF 26	2012. 10. 24.

### 3. System Block Diagram

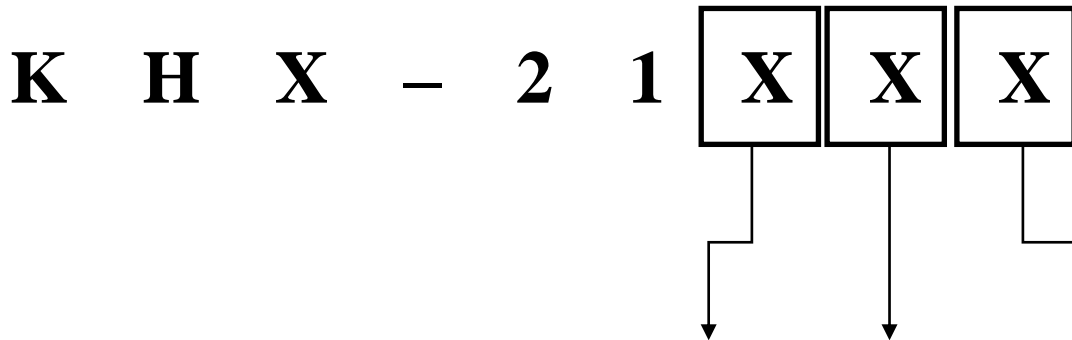


### 5. Specifications.

#### 5.1. Model.

MODEL	KHX-211X	KHX-212X	KHX-213X	KHX-214X
Dimensions (W x L x H) mm	Refer to Page 14			
Card Dispensing Time (Sec)	0.8	0.8	0.8	0.8
Max. Card Loading Capacity	100 PCS	200 PCS	300 PCS	500 PCS
	In case of 0.76 mm card			
Card Material	P.V.C			
Max. Card Width, Length	ISO 7810			
Max. Card Thickness	0.76~1.0 mm			
Environmental Requirements	1. Operating Temperature and Humidity: 0~40°C, 0~95% RH 2. Conservation Temperature and Humidity: -20~70°C, 0~95% RH			
Power Consumption	1. DC Motor Driver: Output Current 1.5A per channel. 2. Input voltage: DC 24V Only (DC±5%, Min. 2.5A) 3. Standby: 44 mA(+5%).			

<b>Doc No</b>	<b>KHX-21XX SPECIFICATIONS</b>	<b>REV</b>	<b>PAGE</b>	<b>DATE</b>
		G	6 OF 26	2012. 10. 24.

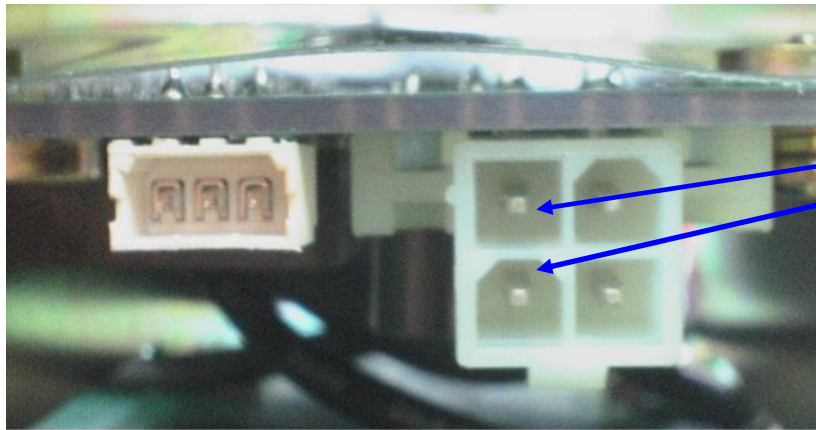


INTERFACE	FUNCTION	TYPE	CAPACITY	THICKNESS	OPTION
T: RS-232C L:TTL	2: Dispenser	1: Single Stacker	1: 100 PCS 2: 200 PCS 3: 300 PCS 4: 500 PCS	4: 0.76T 5: 0.84T 6: 1.00T	A: With Bezel B: Without Bezel C: With Bezel & DC Power Jack type (RS-232C ONLY) D: Without Bezel & DC Power Jack type (RS-232C ONLY) E: With Shutter F: With Shutter & DC Power Jack type (RS-232C ONLY)

<b>Doc No</b>	<b>KHX-21XX SPECIFICATIONS</b>	<b>REV</b>	<b>PAGE</b>	<b>DATE</b>
		G	7 OF 26	2012. 10. 24.

## 6. DC Power Connector(RS-232C ONLY)

### 6.1. Connector number: J6

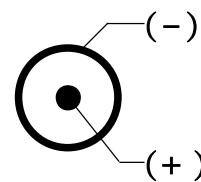


Pin NO.	Signal Name	Direction
1	GND	Input
2	Not use	
3	+24V	
4	Not use	

### 6.2. Connector number: J7 (Option)

-DC Power Jack

Pin NO.	Signal Name	Direction
1	GND	Input
2	GND	
3	DC +24V	



<b>Doc No</b>	<b>KHX-21XX SPECIFICATIONS</b>	<b>REV</b>	<b>PAGE</b>	<b>DATE</b>
		G	8 OF 26	2012. 10. 24.

## 7. Interface

### 7.1 RS232C type model

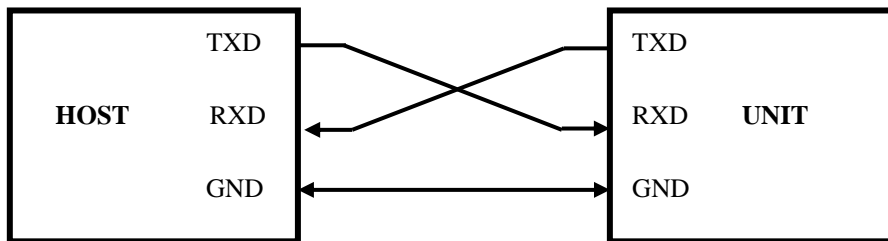
. Interface connector:

When use the KHX-2100's com-cable, connect to twist cable.

When use the user's com-cable, connect to as bellows table

. Connector number: J1

. Connector signal table



Pin No.	Index	KHX-21XX	HOST	Dsub-9	Remark
1	RXD	←		3	Receive
2	TXD	→		2	Transmit
3	GND	↔		5	S.G

. Communication Method

- Asynchronous, Half duplex.
- Baud Rate: 9600, 19200BPS (Default: 9600BPS)
- Data Length: 8Bits
- Parity: None
- Stop Bit: 1Bit



<b>Doc No</b>	<b>KHX-21XX SPECIFICATIONS</b>	<b>REV</b>	<b>PAGE</b>	<b>DATE</b>
		G	9 OF 26	2012. 10. 24.

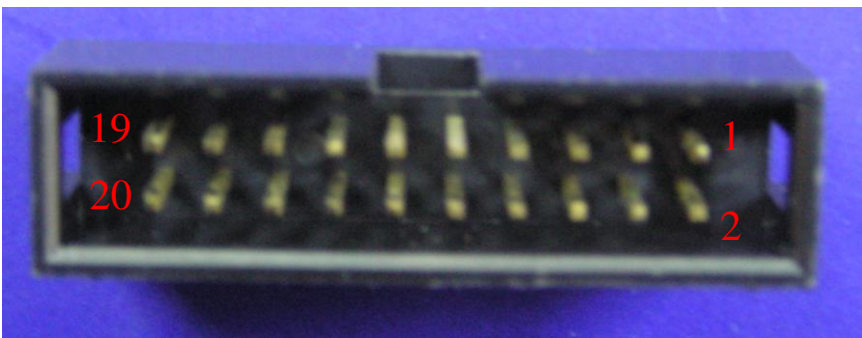
## 7.2 TTL type model

Interface connector:

- Part number: HIF3F-20PA-2.54DS, Manufacture: HIROSE

Connector number: J1

Connector signal table



No.	Signal Name	Input /Output	Function	Configuration	Remark
1	VCC	Input			+24V
2	VCC	Input			+24V
3	D_MOT_ENA	Input			DISPENSER MOTOR
4	D_MOT_A	Input			
5	D_MOT_B	Input			
6	F_MOT_ENA	Input			FEEDING MOTOR
7	F_MOT_A	Input			
8	F_MOT_B	Input			
9	SENSOR_1	Output	Detected by sensor	Active High	FEED1 SENSOR
10	-				
11	-				
12	SENSOR_4	Output	Detected by sensor	Active High	EMPTY SENSOR
13	SENSOR_5	Output	Detected by sensor	Active Low	WARNING SENSOR
14	SENSOR_6	Output	Detected by sensor	Active High	BACK SENSOR
15	SENSOR_7	Output	Detected by sensor	Active High	FRONT SENSOR
16	-				
17	-				
18	-				
19	GND	Input			
20					

<b>Doc No</b>	<b>KHX-21XX SPECIFICATIONS</b>	<b>REV</b>	<b>PAGE</b>	<b>DATE</b>
		G	10 OF 26	2012. 10. 24.

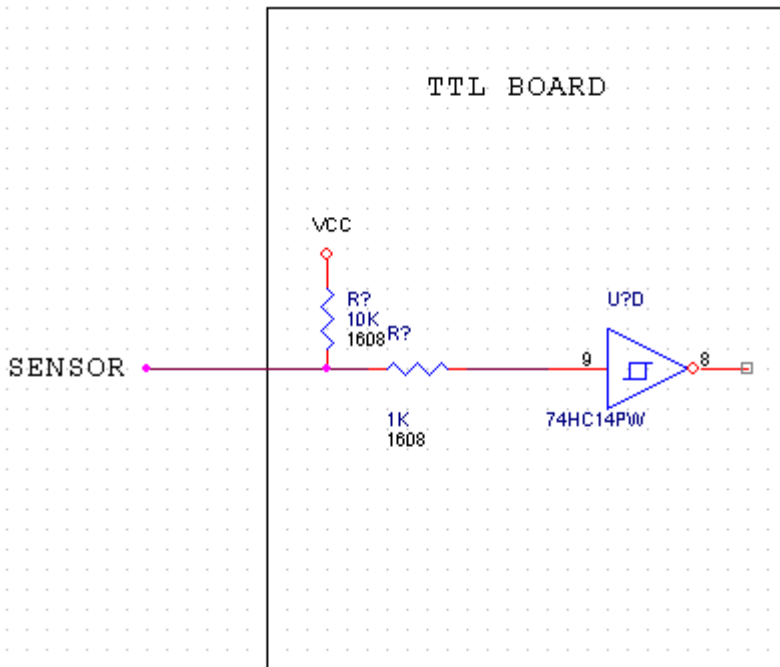
\* D.C Motor Control Table

INPUTS			FUNCTION
MOTOR_ENA	MOTOR_A	MOTOR_B	
H	L	H	Motor Regular Direction(LEFT)
H	H	L	Motor Reverse Direction(RIGHT)
H	MOTOR_A = MOTOR_B		Fast Motor Stop
L	X	X	Feed Running Motor Stop

H: HIGH

L: LOW

X: Don't Care

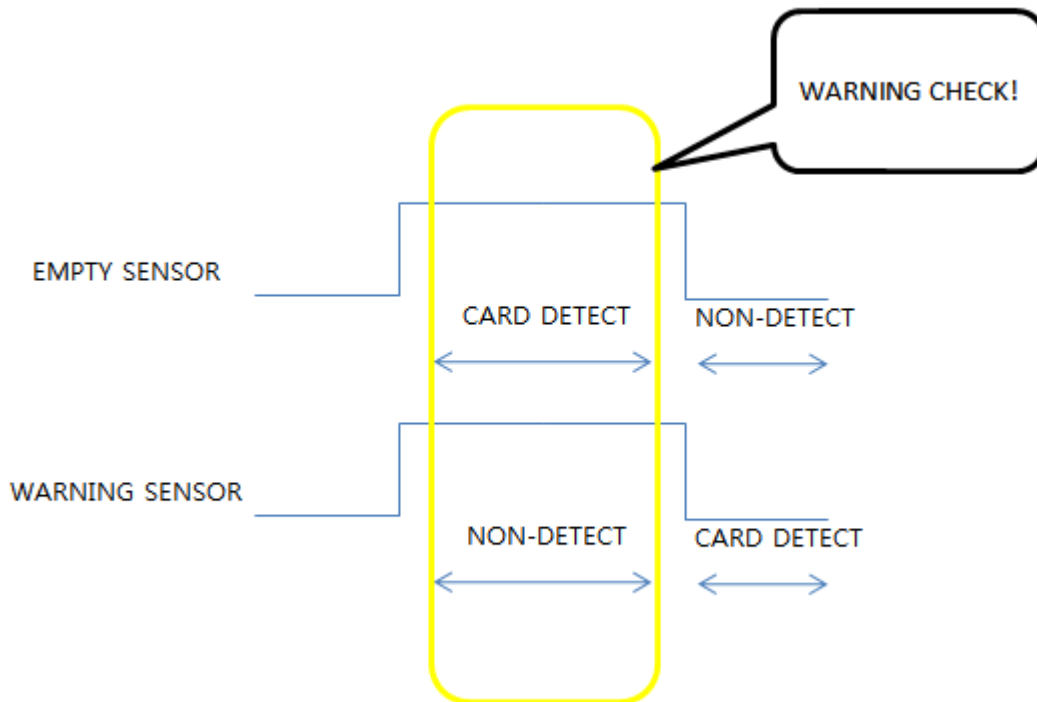


\* Card detected signal (FEED1 SENSOR, FRONT SENSOR, EMPTY SENSOR, BACK SENSOR)



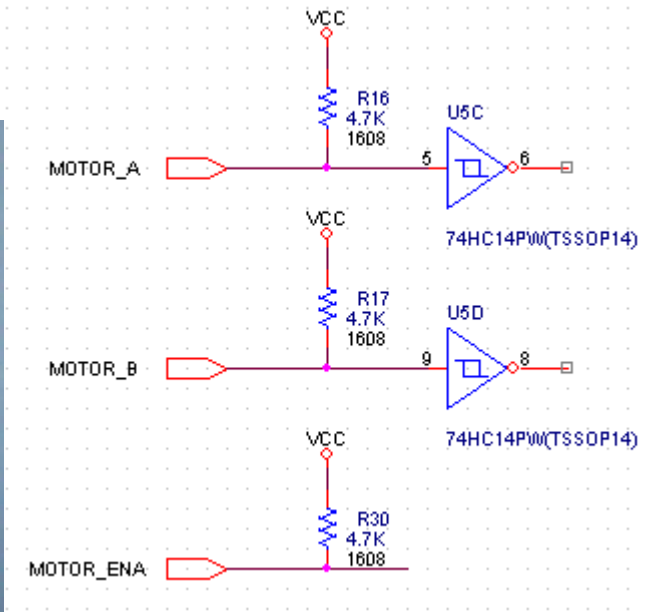
<b>Doc No</b>	<b>KHX-21XX SPECIFICATIONS</b>	<b>REV</b>	<b>PAGE</b>	<b>DATE</b>
		G	11 OF 26	2012. 10. 24.

\* Card detected signal (WARNING SENSOR)

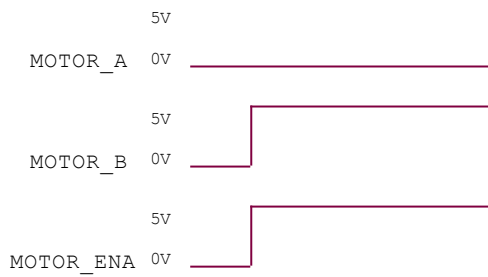


Doc No	KHX-21XX SPECIFICATIONS	REV	PAGE	DATE
		G	12 OF 26	2012. 10. 24.

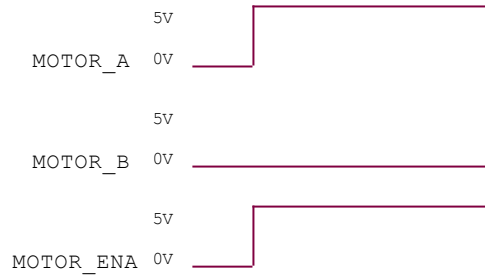
★ MOTOR



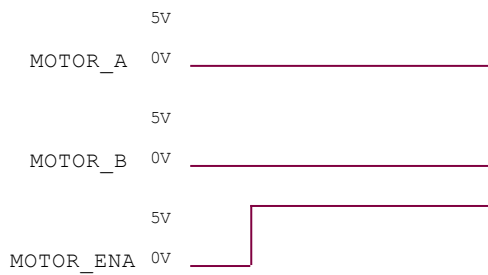
\* Motor Regular Direction Signal(LEFT)



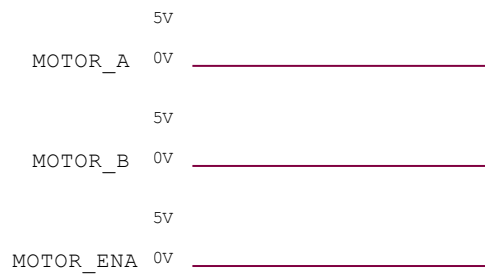
\* Motor Reverse Direction Signal(RIGHT)



\* Fast Motor Stop Signal

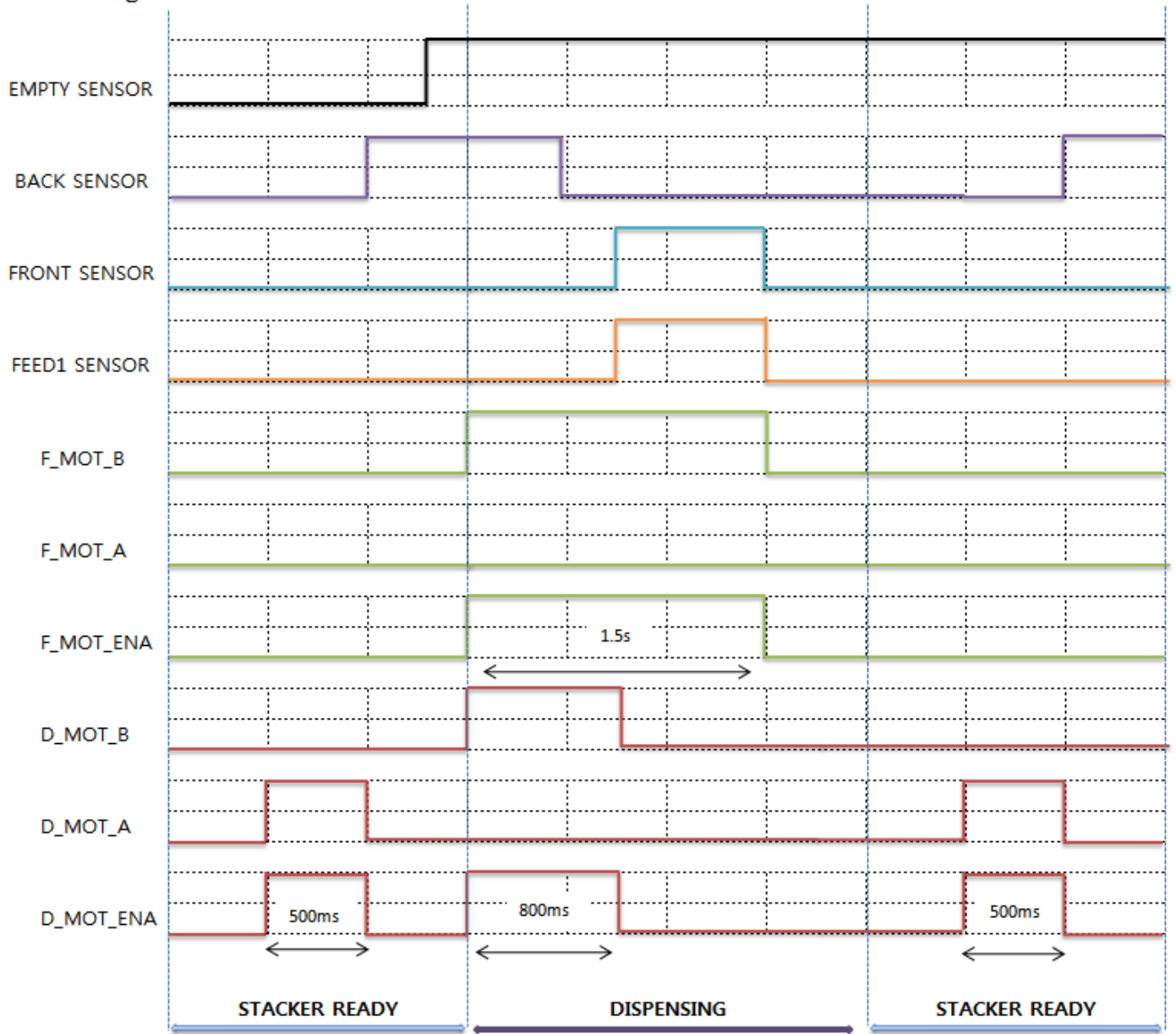


\* Feed Running Motor Stop Signal



Doc No	KHX-21XX SPECIFICATIONS	REV	PAGE	DATE
		G	13 OF 26	2012. 10. 24.

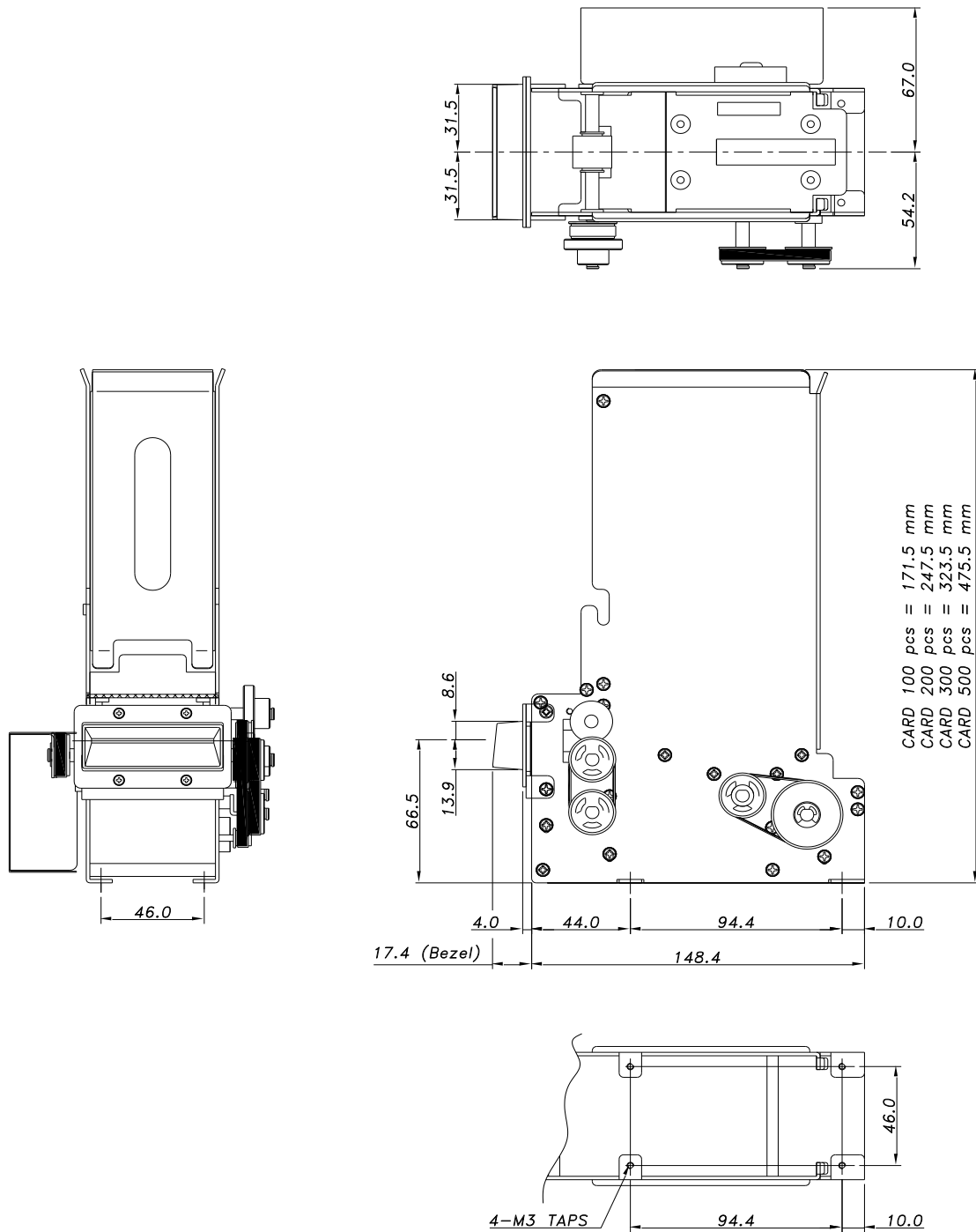
**\*Timing Chart**



Doc No	KHX-21XX SPECIFICATIONS	REV	PAGE	DATE
		G	14 OF 26	2012. 10. 24.

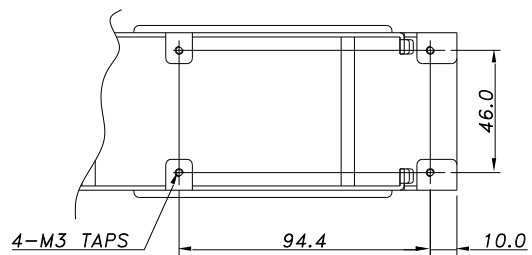
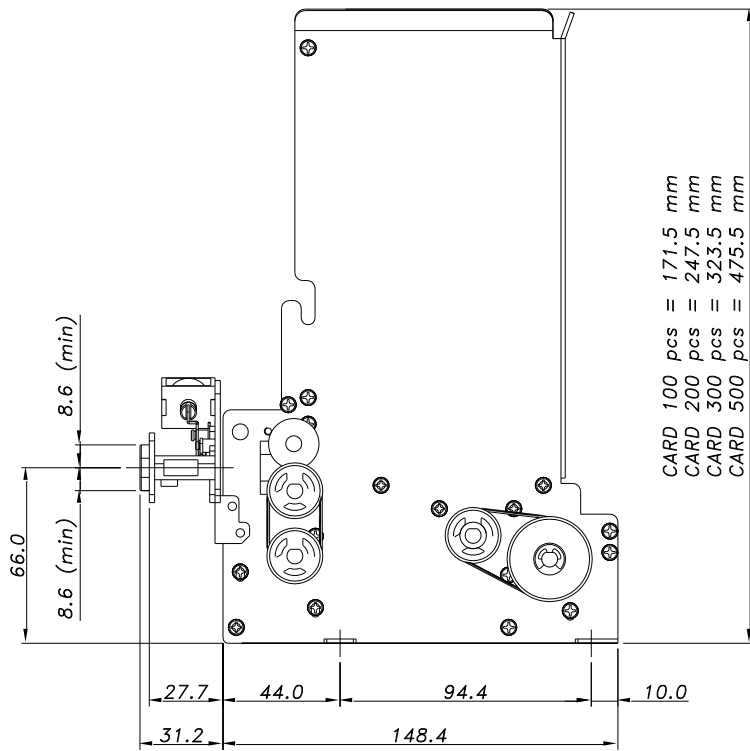
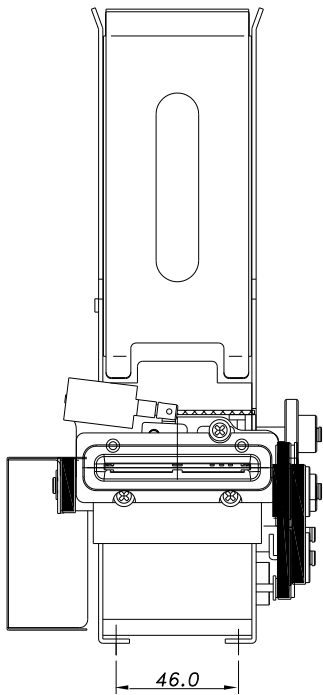
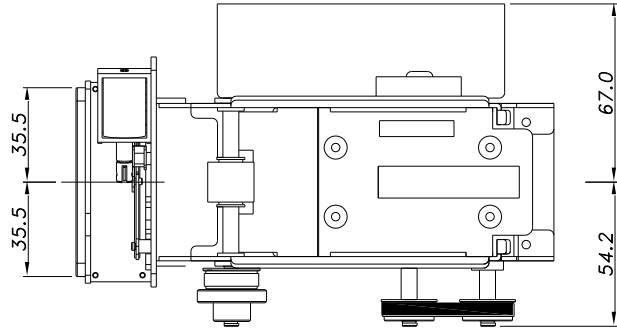
## 8. Technical Drawing

### <Bezel Type>



Doc No	KHX-21XX SPECIFICATIONS	REV	PAGE	DATE
		G	15 OF 26	2012. 10. 24.

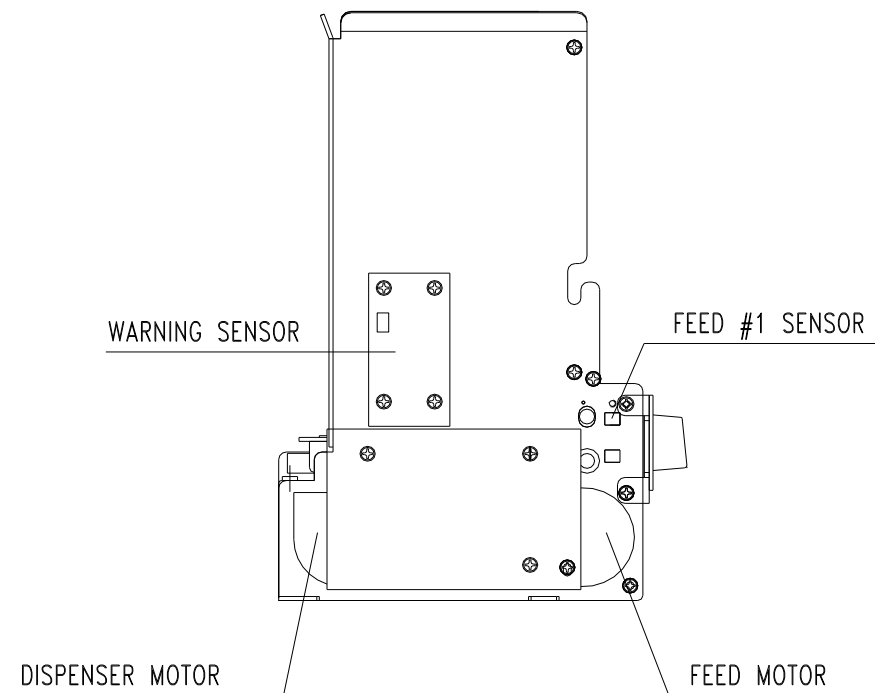
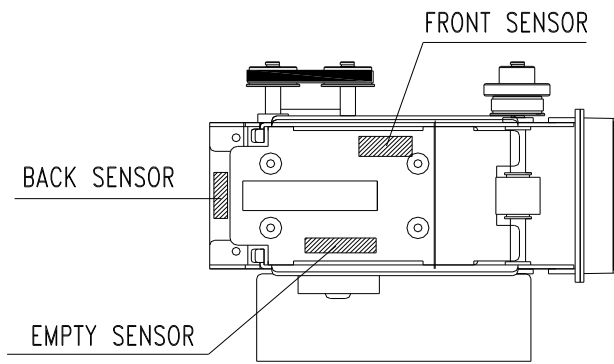
<Shutter Type>



<b>Doc No</b>	<b>KHX-21XX SPECIFICATIONS</b>	<b>REV</b>	<b>PAGE</b>	<b>DATE</b>
		G	16 OF 26	2012. 10. 24.

<Sensor & Motor locations>

Connector No.	Remark
J3	Feed1 Sensor
J4	-
J5	-
J8	Empty Sensor
J13	Back Sensor
J14	Front Sensor
J9	Warning Sensor
J11	Feed Motor
J2	Dispenser Motor





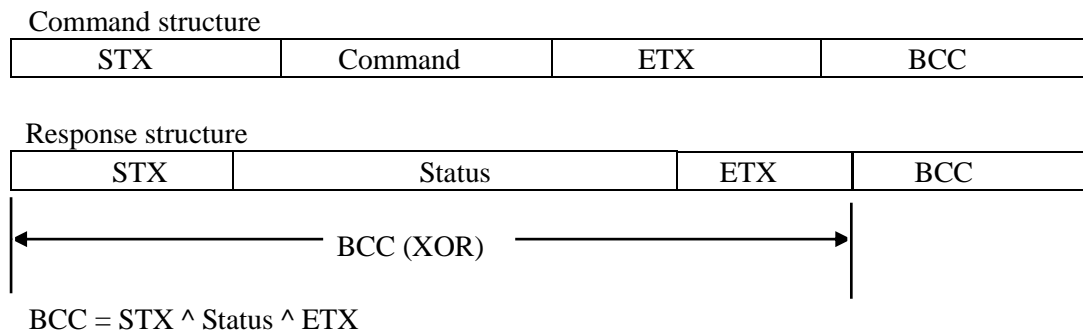
<b>Doc No</b>	<b>KHX-21XX SPECIFICATIONS</b>	<b>REV</b>	<b>PAGE</b>	<b>DATE</b>
		G	17 OF 26	2012. 10. 24.

## 9. RS232C Interface

### 9.1. Control Characters

Character	Hex Value	Description
STX	02	Start of Text Character
ETX	03	End of Text Character
EOT	04	End of Text Character
ENQ	05	Enquiry Character
ACK	06	Acknowledge Character
NAK	15	Negative Acknowledge Character
CAN	18	Cancel Character

### 9.2. Frame Format

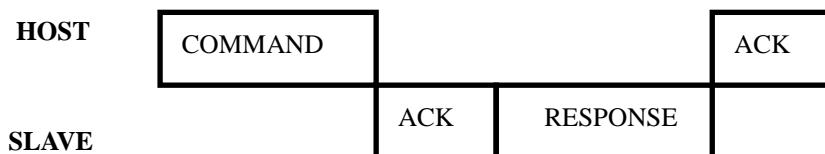


### 9.3. Communication Protocol Sequence

CASE 1)

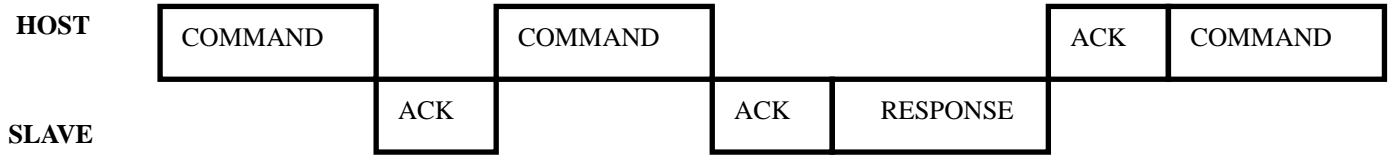


CASE 2) Request Command (0x31)



<b>Doc No</b>	<b>KHX-21XX SPECIFICATIONS</b>	<b>REV</b>	<b>PAGE</b>	<b>DATE</b>
		G	18 OF 26	2012. 10. 24.

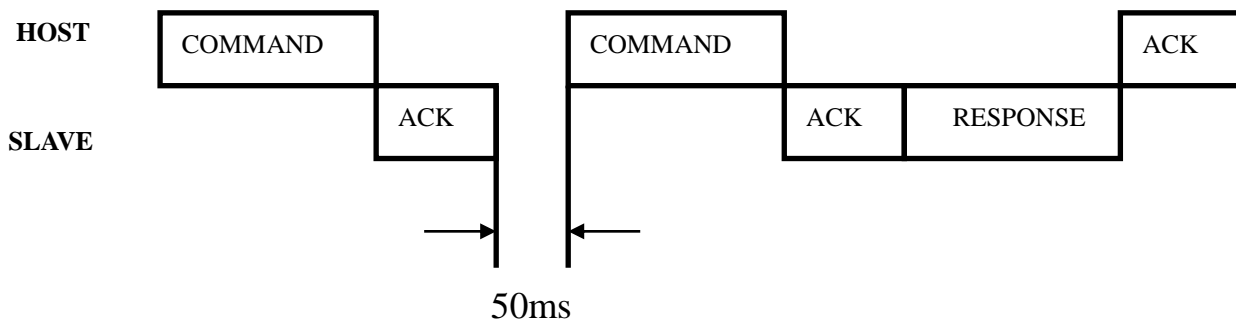
CASE 2-1)



CASE 2-2)



CASE 2-3)



Cf.) To change Baud Rate, send command 50ms after receiving ACK.

<b>Doc No</b>	<b>KHX-21XX SPECIFICATIONS</b>	<b>REV</b>	<b>PAGE</b>	<b>DATE</b>
		G	19 OF 26	2012. 10. 24.

### 10.1. Command Sets List

<b>Item</b>	<b>Command</b>	<b>Description</b>	<b>Note</b>
Clear	0x30	Error Clear	
Request	0x31	Status Request	
Issue	0x40	Issue	
FRONT SENSOR CHECK ISSUE	0x55	Issue(When the there is no card at the FEED#1 SENSOR)	
Issuing Length Set	0xF0	Card Drop(Default)	
	0xF1	Card Issuing Length Setting	Ref.) Page.20
	0xF2		
	0xF3		
	0xF4		
Baud Rate Set	0x50	9600 BPS Setting	Default
	0x51	19200 BPS Setting	

<b>Doc No</b>	<b>KHX-21XX SPECIFICATIONS</b>	<b>REV</b>	<b>PAGE</b>	<b>DATE</b>
		G	20 OF 26	2012. 10. 24.

## 10.2. Command Details

### 10.2.1. Clear

: Clear Motor Jam bit of Status Request Command Response

#### ☞ Command Packet

STX	Command(0x30)	ETX	BCC
-----	---------------	-----	-----

### 10.2.2. Status Request

: Host's Request for status of dispenser

#### ☞ Command Packet

STX	Command(0x31)	ETX	BCC
-----	---------------	-----	-----

#### ☞ Response Packet

STX	Status	ETX	BCC
-----	--------	-----	-----

#### ☞ Status Data Format (Status) – Ref.) Page.14

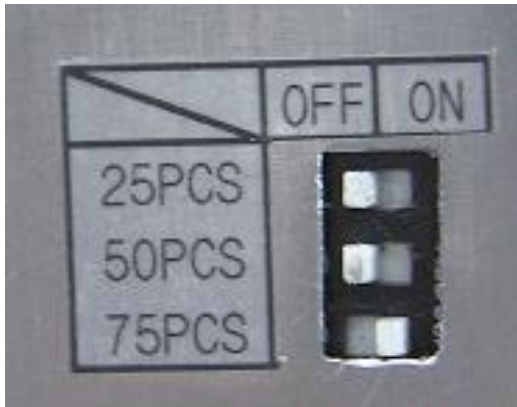
<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>0</b>
1	0	0	0	0	0	0	0

Data	Status 1	Remark
0x80	Good	Normal
0xC0	Busy	Ready
0xA0	Back Sensor Detection	-
0x90	Motor Fail or Card JAM	-
0x88	Front Sensor Detection	-
0x84	Finish Sensor	Finish Sensor detect Card
0x82	Warning Sensor	Warning Sensor detect Card
0x81	Card Empty	Dispenser Card Empty

<b>Doc No</b>	<b>KHX-21XX SPECIFICATIONS</b>	<b>REV</b>	<b>PAGE</b>	<b>DATE</b>
		G	21 OF 26	2012. 10. 24.

Stacker Status	Detail
'Stacker Good'	Good.
'Card Warning'	A few Card in the stacker
'Stacker Empty'	No cards in the stacker

### -Warning Sensor Setting



Set Value	25pcs	50pcs	75pcs
Set to 25pcs cards	ON	OFF	OFF
Set to 50pcs cards	OFF	ON	OFF
Set to 75pcs cards	OFF	OFF	ON

#### 10.2.3. Issue

: Dispense the card. And completely eject it from the feeder module.

☞ Command Packet

STX	Command(0x40)	ETX	BCC
-----	---------------	-----	-----

#### 10.2.4. FRONT SENSOR CHECK ISSUE

: Dispense the card. And completely eject it from the feeder module when the there is no card at the FEED #1 SENSOR.

☞ Command Packet

STX	Command(0x55)	ETX	BCC
-----	---------------	-----	-----

#### 10.2.5. Baud Rate Set

: Baud Rate Setting (After ACK receive, next Command should be transmitted after 50ms)  
Case)

☞ Command Packet (9600BPS)

STX	Command(0x50)	ETX	BCC
-----	---------------	-----	-----

Doc No	KHX-21XX SPECIFICATIONS	REV	PAGE	DATE
		G	22 OF 26	2012. 10. 24.

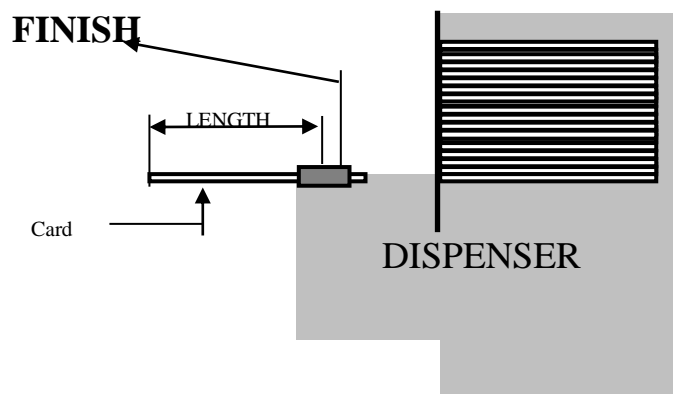
#### 10.2.6. Status of the card jam and the way to deal with the card jam

##### - Dispenser Jam

: Can't use all command except Status Request commands

(All jam is canceled and you can use these command, if Clear command is executed.)

#### 10.2.7. Card Issuing Length Set. Commands (Ref. Page.18)



Commands fix card length before dispensing a card.

- This is reserved as a fixed command and will be in operation until before power-off.
- If this command is not used, 0xF0 is set to be Default.
- To change this command while in operation, send a command after operation stops.
- This command is operable from 0xF0 to 0xF4.

. 0xF0: Card Drop (Default).

. 0xF1: When issued, a card stops 3 mm away from finish sensor.

. 0xF2: When issued, a card stops 36mm away from finish sensor. (-5 mm, + 1.5 mm)

. 0xF3: When issued, a card stops 54mm away from finish sensor. (-8 mm, + 1.5 mm)

. 0xF4: When issued, a card stops 64mm away from finish sensor. (-10 mm, + 1.5 mm)

Doc No	KHX-21XX SPECIFICATIONS	REV	PAGE	DATE
		G	23 OF 26	2012. 10. 24.

TTL EXAMPLE)

```

void stacker_ready(void)
{
    data byte sencnt = 0;
    MOT1_BW
    timer_clear(0);
    timer_count(0,50);           // 0.5 seconds
    while(counter_clear(0) != 0){
        busy();
        if(BACK_SEN == 0){
            if(++sencnt >= 20){
                MOT1_BREAK
                break;
            }
        }
        else{
            sencnt = 0;
        }
    }
    timer_clear(0);
    timer_count(0,5);           //0.05 secods
    while(counter_clear(0) != 0){
        busy();
    }
    MOT1_STOP
    timer_clear(0);
}

void feeding(void)
{
    data byte sencnt = 0;
    MOT1_FW
    MOT2_FW
    timer_clear(0);
    timer_count(0,80);
    while(counter_clear(0) != 0){
        busy();
        if(FINISH_SEN == 1 || FRONT_SEN == 1){
            if(++sencnt >= 20){
                break;
            }
        }
        else{
            sencnt = 0;
        }
    }
    MOT1_STOP
    timer_clear(0);
}

```

Doc No	KHX-21XX SPECIFICATIONS	REV	PAGE	DATE
		G	24 OF 26	2012. 10. 24.

```

void feeding_retry(void)
{
    MOT2_BW
    timer_clear(0);
    timer_count(0,20);
    while(counter_clear(0) != 0){
        busy();
    }
}

void feeding2(void)
{
    MOT2_FW
    timer_clear(0);
    timer_count(0,100);
    while(counter_clear(0) != 0){
        busy();
        if(Stopflag == 1) return;
        if(FINISH_SEN == 0){MOT2_STOP return;}
        if(FINISH_SEN == 0){break;}
    }
    timer_clear(0);

    timer_count(0,100);
    while(counter_clear(0) != 0){
        busy();
        if(Stopflag == 1) return;
        if( FINISH_SEN == 0){ MOT2_STOP break;}
    }
    timer_clear(0);
}

void issue_wait_mode(WaitTimerValue)
{
    if(WaitTimerValue != 0){
        timer_clear(0);
        timer_count(0,WaitTimerValue);
        MOT2_FW
        while(counter_clear(0) != 0){
            if(Stopflag == 1) return;
            busy();
        }
    }
    MOT2_BREAK
    timer_clear(0);
    timer_count(0,5);
    while(counter_clear(0) != 0){
        if(Stopflag == 1) return;
        busy();
    }
    MOT2_STOP
}

```



Doc No	KHX-21XX SPECIFICATIONS	REV	PAGE	DATE
		G	25 OF 26	2012. 10. 24.

```

void dispensing(void)
{
    char i;
    Twocardflag = 0;
    Stackerflag = 1;
    Stopflag = 0;
    if(Dispjamflag == 1) return;
    if(BACK_SEN == 1 && EMPTY_SEN == 1 && Feedflag == 0){
        stacker_ready();
    }
    if( FINISH_SEN == 1){ Twocardflag = 1;}

    if(EMPTY_SEN == 1 && FINISH_SEN == 0 && Twocardflag == 0 && Feedflag == 0){
        feeding();
        if(FINISH_SEN == 0){
            for(i = 0; i<3; i++){
                MOT2_STOP
                stacker_ready();
                if(FINISH_SEN == 0) feeding_retry();
                else if(FINISH_SEN == 1){ break;}
                feeding();
            }
            if(FINISH_SEN == 0){Dispjamflag = 1; MOT1_STOP MOT2_STOP return;}
        }
        if(Standbyflag == 0){
            Stackerflag = 0;
            if(FINISH_SEN == 1){
                timer_clear(0);
                timer_count(0,200);
                while(counter_clear(0) != 0){
                    busy();
                    if(Stopflag == 1) return;
                    if(FINISH_SEN == 1){break;}
                    if(FINISH_SEN == 0){Dispjamflag = 1; MOT1_STOP MOT2_STOP return;}
                }
                timer_clear(0);
                if(FINISH_SEN == 0 ){Dispjamflag = 1; MOT1_STOP MOT2_STOP return;}
            }
            if(Waitflag == 0) feeding2();
            else{
                issue_wait_mode(WaitTimerValue);
                return;
            }
            if(Stopflag == 1){MOT2_STOP return;}
            if( FINISH_SEN == 0){ MOT2_STOP return;}
            else if(FINISH_SEN == 1){Dispjamflag = 1; MOT2_STOP return;}
        }
    }
}

```

Doc No	KHX-21XX SPECIFICATIONS	REV	PAGE	DATE
		G	26 OF 26	2012. 10. 24.

```

else if((FINISH_SEN == 1)){
  Stackerflag = 0;
  if( FINISH_SEN == 0) return;
  if(Standbyflag == 0 && Exitflag == 0){
    if(FINISH_SEN == 1){
      if(Waitflag == 0) feeding2();
      else{
        feeding2();
        if(FINISH_SEN == 1){Dispjamflag = 1; MOT1_STOP MOT2_STOP return;}
        if(EMPTY_SEN == 0){MOT1_STOP MOT2_STOP return;}
        feeding();
        if(FINISH_SEN == 0){
          for(i = 0; i<3; i++){
            MOT2_STOP
            stacker_ready();
            feeding();
            if(FINISH_SEN == 1){ break;}
          }
          if(FINISH_SEN == 0){Dispjamflag = 1; MOT1_STOP MOT2_STOP return;}
        }
        if(Standbyflag == 0){
          Stackerflag = 0;
          if(FINISH_SEN == 1){
            timer_clear(0);
            timer_count(0,200);
            while(counter_clear(0) != 0){
              busy();
              if(Stopflag == 1) return;
              if(FINISH_SEN == 1){break;}
              if(FINISH_SEN == 0){Dispjamflag = 1; MOT1_STOP MOT2_STOP return;}
            }
            timer_clear(0);
            if(FINISH_SEN == 0 ){Dispjamflag = 1; MOT1_STOP MOT2_STOP return;}
          }
          if(Waitflag == 0) feeding2();
          else{
            issue_wait_mode(WaitTimerValue);
            return;
          }
          if(Stopflag == 1) return;
          if( FINISH_SEN == 0){ MOT2_STOP return;}
          else if(FINISH_SEN == 1){Dispjamflag = 1; MOT2_STOP return;}
        }
        return;
      }
    }
    if(Stopflag == 1){MOT2_STOP return;}
    if( FINISH_SEN == 1){Dispjamflag = 1; MOT1_STOP MOT2_STOP return;}
  }
}
}
}
MOT1_STOP
MOT2_STOP
}

```