



**IF2202-02B**  
**INTERFACE BOARD**  
**TECHNICAL REFERENCE**

U00034193251

**Seiko Instruments Inc.**

## Revision List for the IF2202-02B Interface Board Technical Reference

Document Number: U00034193251

Date: April 2000

Page	Line	Type of revision	Before change	After change
		<b>Revised</b>	<b>U00034193250</b>	<b>U00034193251</b>
Cover	-	Modification	<b>U00034193250</b>	<b>U00034193251</b>
Copyright	-	Added	-	Second Edition May 2000
1-4	-	Modified	Figure 1-4 System Configuration <u>PRITER</u>	Figure 1-4 System Configuration <u>PRINTER</u>
4-2	-	Modified	Figure 4-1 IF2202-02B Circuit Block Diagram STT0 to 3	Figure 4-1 IF2202-02B Circuit Block Diagram ST0 to 3
8-47	-	Modified	DC2+ '~' + n You can adjust print density by setting the energy applied to the head to a value between 65% and 135% of the rated energy. The range of n is between 65 and 135.	DC2+ '~' + n Although n can be set between 65 and 135, print density should be adjusted by setting the energy applied to the head to a value between 65% and 130% of the rated energy.
10-1	-	Modified	10.1 GENERAL CHARACTERISTICS Reset voltage: Specification MIN <u>4.2</u> , TYP <u>4.3</u> , MAX <u>4.4</u>	10.1 GENERAL CHARACTERISTICS Reset voltage: Specification MIN <u>4.0</u> , TYP <u>4.1</u> , MAX <u>4.2</u>
B-2	-	Modified	Figure B-1 Circuit Diagram for the IF2202-02B The circuit of the IC7 was changed.	Figure B-1 Circuit Diagram for the IF2202-02B The circuit of the IC7 was changed.

## **IF2202-02B TECHNICAL REFERENCE**

**Document Number U00034193251**

First Edition      February 2000  
Second Edition    May 2000

Copyright © 2000 by Seiko Instruments Inc.  
All rights reserved.

Seiko Instruments Inc. (SII) has prepared this manual for use by SII personnel, licensees, and customers. The information contained herein is the property of SII and shall not be reproduced in whole or in part without the prior written approval of SII.

SII reserves the right to make changes without notice to the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including but not limited to typographical, arithmetic, or listing errors.

SII is a trademark of Seiko Instruments Inc.

Centronics is a registered trademark of Centronics Data Computer Corporation.

ESC/POS is a trademark of EPSON.

IBM is a registered trademark of IBM corp.

## **PREFACE**

This reference manual describes the specifications, functions, and operating procedures for the IF2202-02B Interface Board.

The IF2202-02B is an interface board for the LTP2000 series printer mechanism.

This reference manual also describes the print operation of the LTP2000 series printer mechanism. Read this reference manual thoroughly before using the IF2202-02B.

Fully investigate the industrial property of the sample circuits described in this manual before using.

## TABLE OF CONTENTS

Section	Page
<b>CHAPTER 1 OVERVIEW</b>	
1.1 PRODUCT DESCRIPTION.....	1-1
1.2 PRECAUTIONS .....	1-2
1.3 TERMINOLOGY.....	1-4
<b>CHAPTER 2 GENERAL SPECIFICATIONS</b>	
<b>CHAPTER 3 CONNECTOR AND TERMINAL FUNCTIONS</b>	
3.1 POWER SUPPLY CONNECTOR (CN1) .....	3-1
3.2 THERMAL HEAD CONNECTOR (CN2).....	3-2
3.3 AUTO-CUTTER CONNECTOR (CN3) .....	3-3
3.4 MOTOR CONNECTOR (CN4).....	3-3
3.5 DETECTOR CONNECTOR (CN5) .....	3-4
3.6 HOST DEVICE CONNECOR (CN6) .....	3-5
<b>CAHTER 4 BASIC OPERATION</b>	
4.1 BASIC STRUCTURE .....	4-1
4.2 BASIC PRINT OPERATION .....	4-3
4.2.1 Initialization.....	4-3
4.2.2 Basic Operation for Printing One Line.....	4-4
4.3 POST-INITIALIZATION TIMING .....	4-5
<b>CHAPTER 5 OPERATION PRINCIPLES</b>	
5.1 DATA INPUT/OUTPUT .....	5-1
5.1.1 Parallel Input/Output.....	5-1
5.1.2 Serial Input/Output.....	5-3
5.2 THERMAL HEAD CONTROL .....	5-6
5.2.1 Data Transfer to the Thermal Head .....	5-6
5.2.2 Thermal Head Drive Operation .....	5-6
5.2.3 Head control circuit.....	5-8

Section	Page
5.2.4 Adjustment of Applied Energy .....	5-9
5.2.5 2-ply Thermal Paper Print.....	5-9
5.3 PRINTER MECHANISM CONTROL.....	5-10
5.3.1 Paper Feed Step Motor Drive.....	5-10
5.3.2 The Timing for Driving the Head and Motor .....	5-11
5.3.3 Status Detection .....	5-12
5.3.4 Mechanism Status Output .....	5-13
5.4 AUTO-CUTTER FUNCTION.....	5-14
5.4.1 Auto-cutter Operation .....	5-14
5.4.2 Auto-cutter Initialization .....	5-14
5.4.3 Unlocking of Movable Blade .....	5-14
5.4.4 Auto-cutter Control Timing .....	5-16

## CHAPTER 6 SETUP

6.1 SETTING FUNCTION SWITCHES.....	6-1
6.2 CONNECTING TO THE PRINTER MECHANISM.....	6-6
6.3 CONNECTING TO THE POWER SUPPLY .....	6-8
6.3.1 Selecting the Power Supply.....	6-8
6.3.2 Turning Power On/Off .....	6-8
6.4 CONNECTING TO THE HOST DEVICE .....	6-9
6.4.1 Connecting for Parallel Input/Output .....	6-9
6.4.2 Connecting for Serial Input/Output .....	6-10
6.4.3 Connecting to Switches and LEDs .....	6-11
6.5 OPERATING .....	6-13
6.5.1 Loading Paper .....	6-13
6.5.2 Paper Feed and Backfeed.....	6-14
6.5.3 Test Print Procedure .....	6-14
6.5.4 HEX Dump Print Procedure .....	6-16

## CHAPTER 7 ERROR PROCESSING

7.1 HARDWARE ERRORS.....	7-1
7.2 OPERATION ERRORS.....	7-2

## CHAPTER 8 SOFTWARE SPECIFICATIONS

8.1 CHARACTER SET SELECTION .....	8-1
8.2 FUNCTION CODES AND CHARACTER CODES.....	8-1
8.2.1 Kanji specification/Cancel Mode.....	8-1
8.2.2 Shift Kanji Mode .....	8-2

Section	Page
8.3 MEMORY .....	8-3
8.4 FUNCTION CODE DESCRIPTION.....	8-7
8.4.1 Line/Paper Feed .....	8-7
8.4.2 Format Setting.....	8-9
8.4.3 Character Modification .....	8-13
8.4.4 Character Set .....	8-18
8.4.5 Character Definition.....	8-21
8.4.6 Ruler .....	8-31
8.4.7 Bit image.....	8-36
8.4.8 Stamp .....	8-38
8.4.9 Character String .....	8-43
8.4.10 Print Method .....	8-46
8.4.11 Response Function .....	8-48
8.4.12 Auxiliary Function .....	8-54
8.4.13 Bar Codes.....	8-60

**CHAPTER 9  
CHARACTER CODES**

**CHAPTER 10  
ELECTRICAL CHARACTERISTICS**

10.1 GENERAL CHARACTERISTICS .....	10-1
10.2 INPUT / OUTPUT SIGNAL CONDITIONS.....	10-2
10.3 CONNECTOR LIST .....	10-2

**APPENDIX A  
DIMENSIONS**

**APPENDIX B  
CIRCUIT DIAGRAM**

**APPENDIX C  
COMMAND INDEX**

## TABLES

Table	Page
2-1 General Specifications .....	2-2
3-1 CN1 Terminal Assignments .....	3-1
3-2 CN2 Terminal Assignments .....	3-2
3-3 CN3 Terminal Assignments .....	3-3
3-4 CN4 Terminal Assignments .....	3-3
3-5 CN5 Terminal Assignments .....	3-4
3-6 CN6 Terminal Assignments .....	3-5
5-1 Printer Status Signals.....	5-13
6-1 Function switches.....	6-2
6-2 Setting Function Switches (Printer model) .....	6-3
6-3 Setting Function Switches (2-ply thermal paper mode) .....	6-3
6-4 Setting Function Switches (Paper) .....	6-3
6-5 Setting Function Switches (Print density).....	6-4
6-6 Setting Function Switches (Print speed) .....	6-4
6-7 Setting Function Switches (Thermal head error output method) .....	6-4
6-8 Setting Function Switches (Data input method) .....	6-5
6-9 Setting Function Switches (Data control method) .....	6-5
6-10 Setting Function Switches (Thermal head division method) .....	6-6
6-11 Setting Function Switches (Number of dots for dynamic division) .....	6-6
6-12 Current Consumption When Driving with Fixed Division .....	6-8
6-13 Maximum Activated Current.....	6-8
6-14 Parallel Input/Output Terminals (CN6).....	6-9
6-15 Serial Input/Output Terminals (CN6).....	6-10
6-16 Switch and Error Terminals (CN6) .....	6-11
7-1 Hardware Error Codes .....	7-1
8-1 Memory Area after Initialization.....	8-3
8-2 Number of Bytes of Memory Control Information.....	8-4
8-3 Commands for Allocating or Freeing Memory Area .....	8-5
8-4 Range of Corresponding Output Codes.....	8-48



## FIGURES

Figure	Page
1-1 System Configuration.....	1-4
4-1 IF2202-02B Circuit Block Diagram.....	4-2
4-2 Post-Reset Operation Timing.....	4-5
5-1 Timing for Receiving Parallel Data.....	5-2
5-2 Serial Input/Output Signal Timing Chart.....	5-3
5-3 Dynamic Division of LTP2442 When All Dots are Activated.....	5-7
5-4 The Timing for Driving the Head and the Motor.....	5-11
5-5 Returning the Movable Blade to Inside the Unit.....	5-15
5-6 Auto-cutter Control Timing.....	5-16
6-1 Position of the Setup Terminals.....	6-1
6-2 Parallel Input/Output Circuit Example.....	6-10
6-3 RS-232C Level Convert Circuit Example.....	6-11
6-4 Switch and Display Circuit Example.....	6-12
6-5 Test Print Sample (LTP2442).....	6-15
6-6 Hex Dump Print Sample (LTP2442).....	6-17
8-1 Program Sample 1.....	8-12
8-2 Print Sample 1.....	8-13
8-3 Program Sample 2.....	8-17
8-4 Print Sample 2.....	8-17
8-5 Program Sample 3.....	8-20
8-6 Print Sample 3.....	8-20
8-7 Program Sample 4.....	8-29
8-8 Print Sample 4.....	8-30
8-9 Program Sample 5.....	8-35
8-10 Print Sample 5.....	8-35
8-11 Program Sample 6.....	8-37
8-12 Print Sample 6.....	8-37
8-13 Program sample 7.....	8-42
8-14 Print Sample 7.....	8-43
8-15 Program Sample 8.....	8-45
8-16 Print Sample 8.....	8-46
8-17 Program Sample 9.....	8-52
8-18 Print Sample 9.....	8-53
8-19 Program Sample 10.....	8-58
8-20 Print Sample 10.....	8-59
8-21 Program Sample 11.....	8-63
8-22 Print Sample 11.....	8-64

Figure	Page
9-1 Extended Graphics Character Set .....	9-1
A-1 Dimensions .....	A-1
B-1 Circuit Diagram for IF2202-02B .....	B-2

## CHAPTER 1

### OVERVIEW

#### 1.1 PRODUCT DESCRIPTION

The IF2202-02B is an interface used with LTP2000 series printers for printing data sent from a host device.

The IF2202-02B processes the data input from a host device and also outputs internal test patterns.

The features are described below:

- The IF2202-02B is compatible with both parallel and serial data inputs and can print characters and bit images.
- The IF2202-02B supports an auto-loading function, so paper exchange is simple and hassle-free.
- The IF2202-02B supports autocutter control function and can cut a paper automatically.

## 1.2 PRECAUTIONS

### <Power Supply Precautions>

- Use a power supply voltage within the range specified in **Section 10.1, General Specifications**.
- When switching the power supply ON, turn on  $V_{CC}$  and  $V_P$  simultaneously, or turn on  $V_{CC}$  first, and then  $V_P$ .
- When switching the power supply OFF, turn off  $V_{CC}$  and  $V_P$  simultaneously, or turn off  $V_P$  first, and then  $V_{CC}$ .
- Use two power supplies (a  $V_{CC}=5$  V line and a  $V_P=24$  V line) and both GND should be common.
- Use a power supply which provides stable voltage output even when the current fluctuates.
- The IF2202-02B incorporates a reset circuit to prevent malfunction of the circuit due to a reduction in voltage.

### <Connecting Precautions>

- Be sure that terminals are not shorted.
- Be sure to turn the power off before you connect or disconnect any cables to or from the IF2202-02B.
- Make the connecting cable (CN6) to the host device as short as possible.
- When connecting to a power supply (CN1), use all power supply terminals ( $V_{CC}$ ,  $V_P$ , and GND).
- Since the signal terminal of the connecting cable (CN6) to the host device is CMOS input / output, configure the connecting circuit so as not to cause latch-up: when a voltage of  $V_{CC} + 0.3$  V or more or GND-0.3 V or less is on the signal terminal, the IC is shorted by the flow of overcurrent in the IC. Latch-up will happen in the following cases:
  - The output signal voltage of the power supply on the host device side and  $V_{CC}$  are different.
  - The power-on of the interface and the host device are not executed simultaneously.
  - GND impedance is high: the power cable is too long or too thin, or the number of the cables is too few.

For the connection method in order to avoid latch-up, see **Section 6.4, Connecting to the Host Device**.

### <Handling Precautions>

- Do not install the IF2202-02B in a location that is subject to intensive static electricity, high vibration, electromagnetic fields, corrosive gas, rain, fog, or direct sunlight.
- Use the following SII-recommended thermal paper for printing with a LTP2000 series printer mechanisms.

Normal thermal paper:      TF50KS-E2C from Nippon Paper Co., Ltd.  
   PD160R-N from Oji Paper Co.,Ltd.  
   TW80KKS from Nippon Paper Co., Ltd.  
   AFP235 from Mitsubishi Paper Co.,Ltd.  
   TL69KS-R2 from Nippon Paper Co., Ltd.  
   TL69KS-HW74 from Nippon Paper Co., Ltd.  
   TC98KS-T1 from Nippon Paper Co., Ltd.

- Never modify the IF2202-02B.

### 1.3 TERMINOLOGY

- Dot** A character or graphic image is composed of picture elements called dots. A dot corresponds to one of the thermal head heat elements.
- Dot line** A line of dots arranged in the direction perpendicular to the paper feed direction. The dot line corresponds to the line of the thermal head heat elements.
- Character spacing** The space between two characters in the direction perpendicular to the paper feed direction.
- Line spacing** The space between two adjacent characters in the direction parallel to the paper feed direction.
- Input buffer** The memory inside the interface for storing the data (characters and commands) received from the host device. The command is executed at the time when fetching from the input buffer. The characters are fetched from the input buffer and stored in the line buffer. The capacity of the input buffer is 252 bytes.
- Line buffer** The memory for storing one line of character data fetched from the input buffer. When the input buffer is filled with character data or the printing conditions are made valid through a command, the data in the input buffer is printed. The system configuration is shown in **Figure 1-1**.

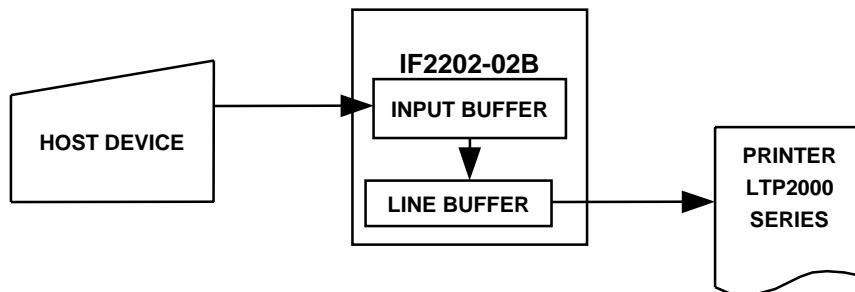


Figure 1-1 System Configuration

- One line full print** The printing executed when the data in the line buffer exceeds one line. The overflowing data is stored at the beginning of the next line.
- Font** The form of a character. A character is composed of dots in a group. The user can define a font using the downloaded font, etc.
- Auto-loading function** When inserting paper in the printer after paper is used up, the paper feed motor is driven automatically to easily replace the paper.

**Font size**

The size of a character. There are two sizes of font, 24-dot and 16-dot font family.

	<u>16 dots</u>	<u>24 dots</u>
Extended graphics character	16×8	24×12
Downloaded character	16×8	24×12
User-defined character	16×16	24×24





**CHAPTER 2**  
**GENERAL SPECIFICATIONS**

**Table 2-1** lists the general specifications of the IF2202-02B.

Table 2-1 General Specifications

Item	Specifications		
	LTP2242	LTP2342	LTP2442
Character type	Extended graphics character set Downloaded character Option font User-defined character		
Character configuration	<u>16 dots</u> 16×8	<u>24 dots</u> 24×12 <sup>*1</sup>	
Extended graphics character set	16×8	24×12 <sup>*1</sup>	
Downloaded character	16×8	24×12 <sup>*1</sup>	
User-defined character	16×16	24×24 <sup>*1</sup>	
Input control method	Parallel (modified Centronics) Serial (TTL level)		
Total dot number	432 dots	576 dots	864 dots
Printable dot number	432 dots	576 dots	832 dots
Common activatable dot number	432 dots	576 dots	720 dots
Number of characters per line (24 dot font size, character spacing:4 dots)	27	36	52
Dot resolution	8 dots/mm		
Paper feed pitch	0.125 mm		
Line spacing	16 dots*1		
Character spacing	4 dots*1		
Maximum print speed (mm / second) (Condition 1)	90	75	
Operating voltage range V <sub>CC</sub> V <sub>P</sub>	5 ±0.5 V 24 V ±2.4 V		
Current consumption (I <sub>CC</sub> ) (Condition 2) Stand by Printing	Max.50 mA Max.170 mA		
Operating temperature	0 to 50 °C		
Storage temperature	-20 to 60 °C		
Dimensions	90 mm (W)×100 mm (D)×22 mm (H)		
Weight	approx. 70g		

\*1 The default value can be changed through commands.

Condition 1: 25°C, fixed division, 24-dot family normal character,  
line spacing : 16 dots, character spacing : 4 dots, and within 36 characters in one line.

Condition 2: V<sub>CC</sub>=5 V, 25°C, no error  
When input/output terminal is not connected.

## CHAPTER 3

### CONNECTOR AND TERMINAL FUNCTIONS

The IF2202-02B, the LTP2000 series printer, auto-cutter and the host device are connected using five connectors.

In the tables for each terminal, the negative logic signals have “!” before the signal name. The input/output direction indicates the direction of a signal as viewed from the IF2202-02B.

#### 3.1 POWER SUPPLY CONNECTOR (CN1)

CN1 is connected to the power supply for the IF2202-02B and has 6 pins. CN1 terminal assignments are listed in **Table 3-1**.

Table 3-1 CN1 Terminal Assignments<sup>\*1</sup>

Terminal No.	Signal Name	I/O <sup>*2</sup>	Function
1	Vcc	I	Circuit power (5 V)
2	GND	—	GND
3	GND	—	GND
4	GND	—	GND
5	Vp	I	Printer power (24V)
6	Vp	I	Printer power (24V)

<sup>\*1</sup> Conformed socket: : 171216-1 (AWG20 to 24), AMP

<sup>\*2</sup> I-input, O-output, — -not applicable

### 3.2 THERMAL HEAD CONNECTOR (CN2)

CN2 is connected to the thermal head and has 16 pins.  
 CN2 terminal assignments are listed in **Table 3-2**.

Table 3-2 CN2 Terminal Assignments<sup>\*1</sup>

Terminal No.	Signal Name	I/O <sup>*2</sup>	Function
1	Vp	O	Head common power
2	Vp	O	Head common power
3	TH	I	Thermistor signal
4	GND	—	GND
5	!DST1	O	1st block · head strobe signal
6	GND	—	GND
7	GND	—	GND
8	GND	—	GND
9	GND	—	GND
10	Vdd	O	Head logic power (5 V)
11	!DST2	O	2nd block · head strobe signal
12	!LATCH	O	Head data latch signal
13	CLOCK	O	Head data clock signal
14	DATAIN	O	Head data signal (serial)
15	Vp	O	Head common power
16	Vp	O	Head common power

<sup>\*1</sup> Connector used: 00-8283-1612-00-000, Elco

<sup>\*2</sup> I- input, O- output; — - not applicable

### 3.3 AUTO-CUTTER CONNECTOR (CN3)

CN3 is connected to the auto-cutter attached to the LTP2000 series printers and has 4 pins. CN3 terminal assignments are listed in **Table 3-3**.

Table 3-3 CN3 Terminal Assignments<sup>\*1</sup>

Terminal No.	Signal Name	I/O <sup>*2</sup>	Function
1	CMOT+	O	Auto-cutter motor drive signal (+)
2	CMOT-	O	Auto-cutter motor drive signal (-)
3	GND	—	GND
4	!SENS	I	Home position detector signal

<sup>\*1</sup> Connector used : 5483-04AX, Molex

<sup>\*2</sup> I-input, O-output, — -not applicable

### 3.4 MOTOR CONNECTOR (CN4)

CN4 is connected to the motor of the LTP2000 series printers and has 4 pins. CN4 terminal assignments are listed in **Table 3-4**.

Table 3-4 CN4 Terminal Assignments<sup>\*1</sup>

Terminal No.	Signal Name	I/O <sup>*2</sup>	Function
1	!A	O	Motor drive signal
2	B	O	Motor drive signal
3	A	O	Motor drive signal
4	!B	O	Motor drive signal

<sup>\*1</sup> Connector used : 53324-0412, Molex

<sup>\*2</sup> I-input, O-output, — -not applicable

### 3.5 DETECTOR CONNECTOR (CN5)

CN5 is connected to the paper empty detector and the head up detector, and has 5 pins. CN5 terminal assignments are listed in **Table 3-5**.

Table 3-5 CN5 Terminal Assignments<sup>\*1</sup>

Terminal No.	Signal Name	I/O <sup>*2</sup>	Function
1	V <sub>PS</sub>	O	Power supply for the paper empty detector
2	PS	I	Paper empty detector signal
3	GND	—	Paper empty detector GND
4	GND	—	Head up detector GND
5	HS	I	Head up detector signal

<sup>\*1</sup> Connector used : 53324-0510, Molex

<sup>\*2</sup> I-input, O-output, — -not applicable

### 3.6 HOST DEVICE CONNECOR (CN6)

CN6 carries the following signals:

- V<sub>cc</sub> power supply and GND
- Parallel input / output signals
- Serial input / output signals
- Error output signals
- Feed, backfeed and reset input signals

CN6 terminal assignments are listed in **Table 3-6**.

Table 3-6 CN6 Terminal Assignments<sup>\*1</sup>

Terminal No.	Signal Name	I/O <sup>*3</sup>	Function
1	GND	—	GND
2	!STROBE	I	Parallel strobe signal
3	DATA0	I	Parallel data signal
4	DATA1	I	Parallel data signal
5	DATA2	I	Parallel data signal
6	DATA3	I	Parallel data signal
7	DATA4	I	Parallel data signal
8	DATA5	I	Parallel data signal
9	DATA6	I	Parallel data signal
10	DATA7	I	Parallel data signal
11	!ACK	O	Parallel acknowledge signal
12	BUSY	O	Parallel busy signal
13	V <sub>cc</sub>	O	Circuit power <sup>*2</sup>
14	PE	O	Paper empty signal
15	!ERR	O	Error signal
16	ST0	O	Printer status signal
17	ST1	O	Printer status signal
18	ST2	O	Printer status signal
19	!INIT	I	Reset signal
20	!FEED	I	Feed switch signal
21	!BFEEED	I	Back feed signal
22	V <sub>cc</sub>	O	Circuit power <sup>*2</sup>
23	TxD	O	Serial data output signal (TTL level)
24	RxD	I	Serial data input signal (TTL level)
25	SBUSY	O	Serial busy signal (TTL level)
26	GND	—	GND

<sup>\*1</sup> Conformed sockets: XG4M-2630-T (1.27 Pitch flat cable, AWG28), OMRON  
 XG5M-2632-N (AWG24), OMRON  
 XG5M-2635-N (AWG26 to 28), OMRON

<sup>\*2</sup> This terminal is prepared for connecting an external LED drive circuit or a RS-232C level convert circuit.

The circuit power must be supplied from the CN1.

<sup>\*3</sup> I-input, O-output, — -not applicable





## CHAPTER 4

### BASIC OPERATION

#### 4.1 BASIC STRUCTURE

The IF2202-02B consists of a single circuit board. The circuit block diagram for the IF2202-02B is illustrated in **Figure 4-1**.

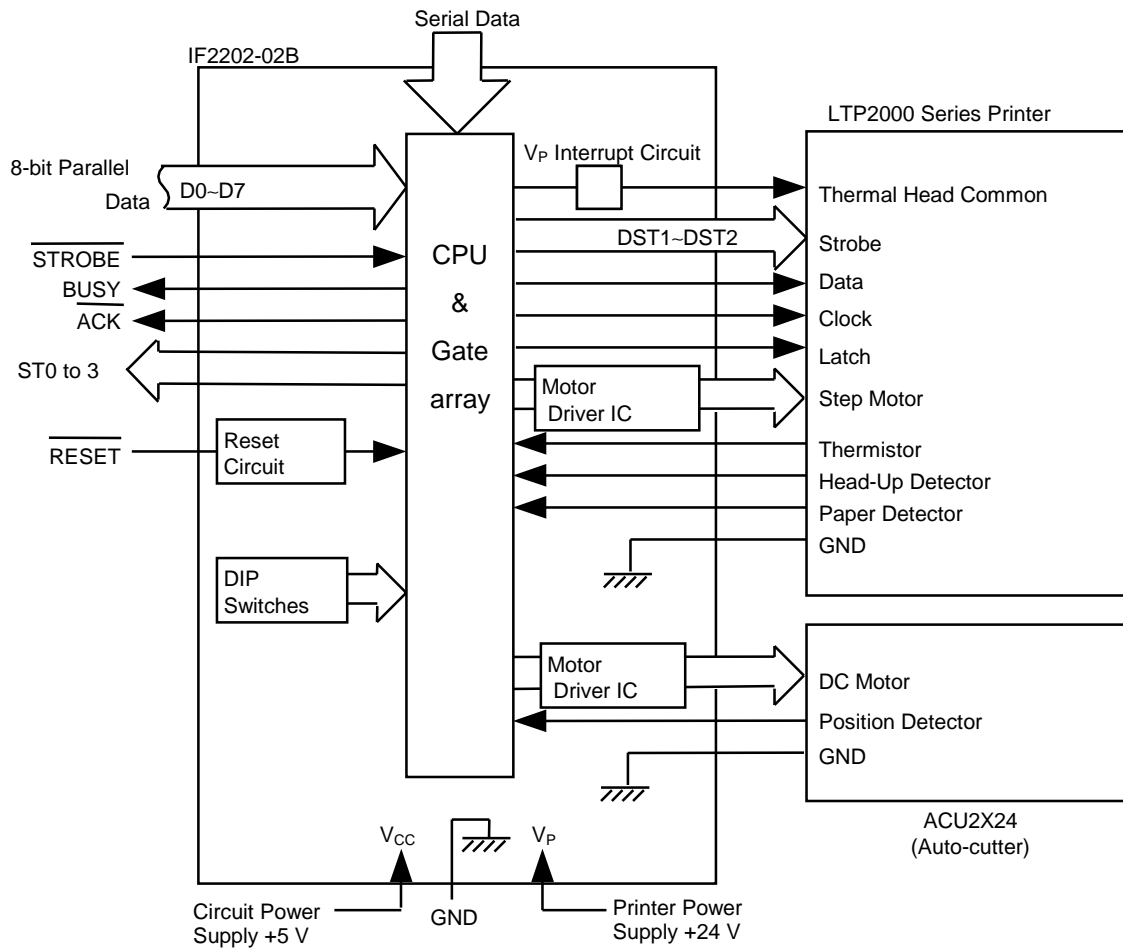


Figure 4-1 IF2202-02B Circuit Block Diagram

The data processing block is a multichip and consists of a CPU and a gate array. It also includes a DIP switch pattern, a reset circuit, a head resistance sensing circuit, a  $V_p$  interrupt circuit, and a motor control circuit.

## 4.2 BASIC PRINT OPERATION

### 4.2.1 Initialization

The IF2202-02B is initialized through power-on (power-on reset) and hardware reset. According to the functions selected using the function switches, the IF2202-02B checks the peripheral circuit, clears the memory, measures head resistance, and senses  $V_P$  voltage.

When an error occurs in the peripheral circuits, the function switch settings, or the thermal head during initialization, serial output of error codes is performed (see **Section 7.1, HARDWARE ERRORS**).

When  $V_p$  is OFF, or when  $V_p$  is under 10 V at initialization, the initialization is not complete until  $V_p$  is in the permitted range (10 V to 27 V). During these times, data input cannot be performed and error codes are not output.

When turning off  $V_p$  following  $V_P$  power-on and initialization, a  $V_p$  error may occur (see **CHAPTER 7, ERROR MODE**).

#### NOTE

Though the  $V_p$  voltage range is 21.6 V to 26.4 V, the range for the IF2202-02B is 10 V to 27 V in consideration for the detection error and momentary fluctuation.

#### **4.2.2 Basic Operation for Printing One Line**

- (1) The data sent from the host device is stored in the input buffer.  
Since the storage to the input buffer is processed using interrupts, it is possible to receive data during printing.
- (2) The data is fetched from the input buffer. If it is character data, step (3) is executed .  
If it is not character data but a command, the function specified by the command is executed.
- (3) The character data is stored in the line buffer.  
When the character data exceeds one line or fetches the print command, printing starts.
- (4) The character codes of the line buffer are changed into the image data of a dot line.
- (5) The image data is transferred to the head.
- (6) The motor and the head are driven.
- (7) The IF2202-02B repeats steps (4) to (6) until the printing of one line has been complete.
- (8) The IF2202-02B feeds the paper one line space and returns to step (2).

### 4.3 POST-INITIALIZATION TIMING

Figure 4-2 shows the operation timing immediately after the IF2202-02B is initialized.

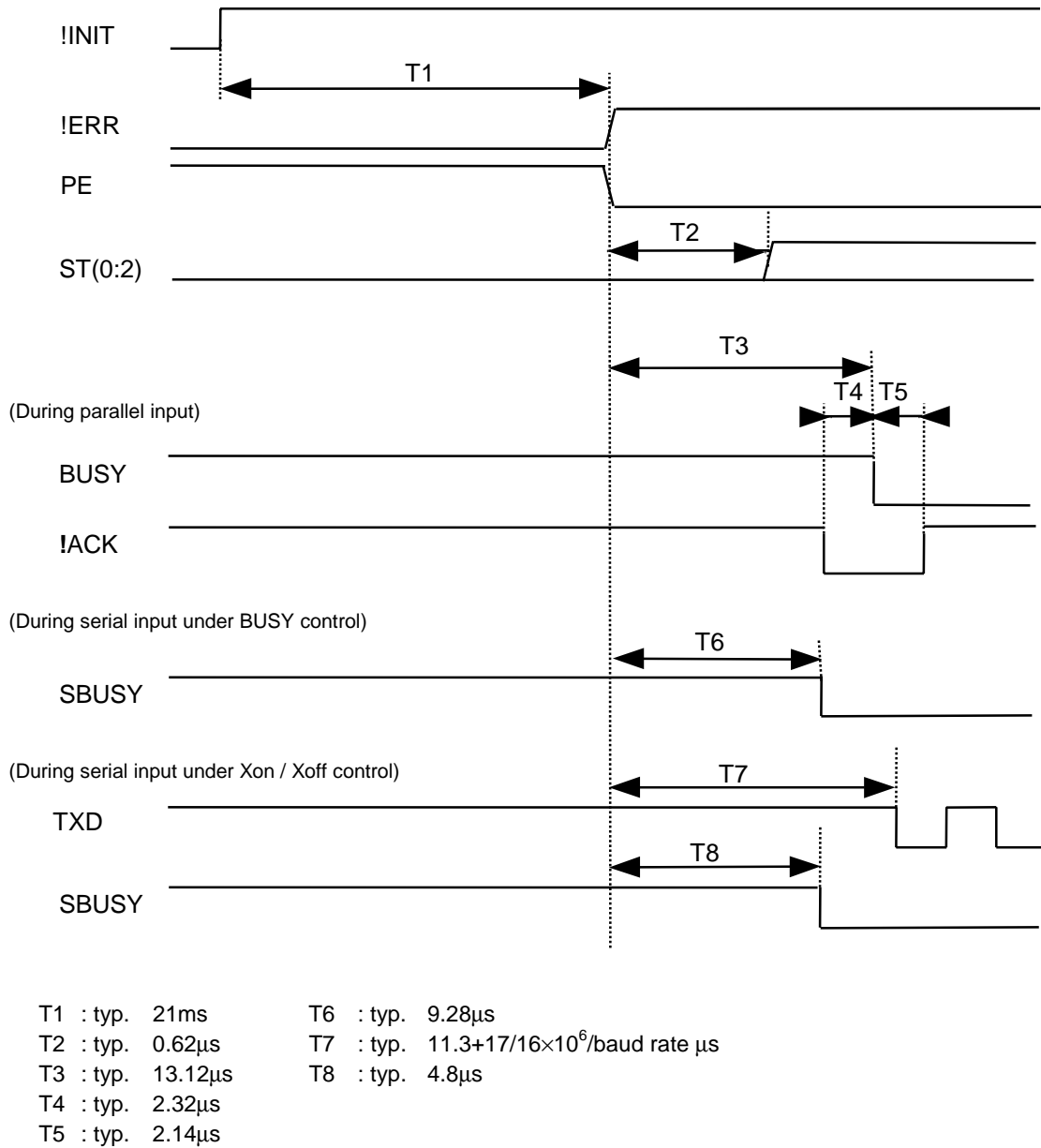


Figure 4-2 Post-Reset Operation Timing

- **During parallel input**

The SBUSY and TXD signals remain high and do not change. If there is an error at the time when initialization is complete, the BUSY signal will be changed to low and the !ACK signal will also be changed to low when the error conditions are released.

- **During serial input under BUSY control**

The BUSY, !ACK, and TXD signals remain high and do not change. If there is an error at the time when initialization is complete, the SBUSY signal will be changed to low when the error conditions are cleared.

- **During serial input under  $X_{on}/X_{off}$  control**

The BUSY and !ACK signals remain high and do not change. If there is no error at the time when initialization is complete,  $X_{on}$  will be output. If an error occurs after initialization,  $X_{off}$  will be output.

## CHAPTER 5

### OPERATION PRINCIPLES

#### 5.1 DATA INPUT/OUTPUT

##### 5.1.1 Parallel Input / Output

###### (1) Timing for receiving parallel data

The data input/output between the host device and the IF2202-02B is controlled as shown in **Figure 5-1**. The IF2202-02B continues to receive data during print operation.

The busy and acknowledge signal are output with every byte. If 252 bytes have been stored in the input buffer, the busy status continues until the amount of data stored in the input buffer becomes 251 bytes or less.

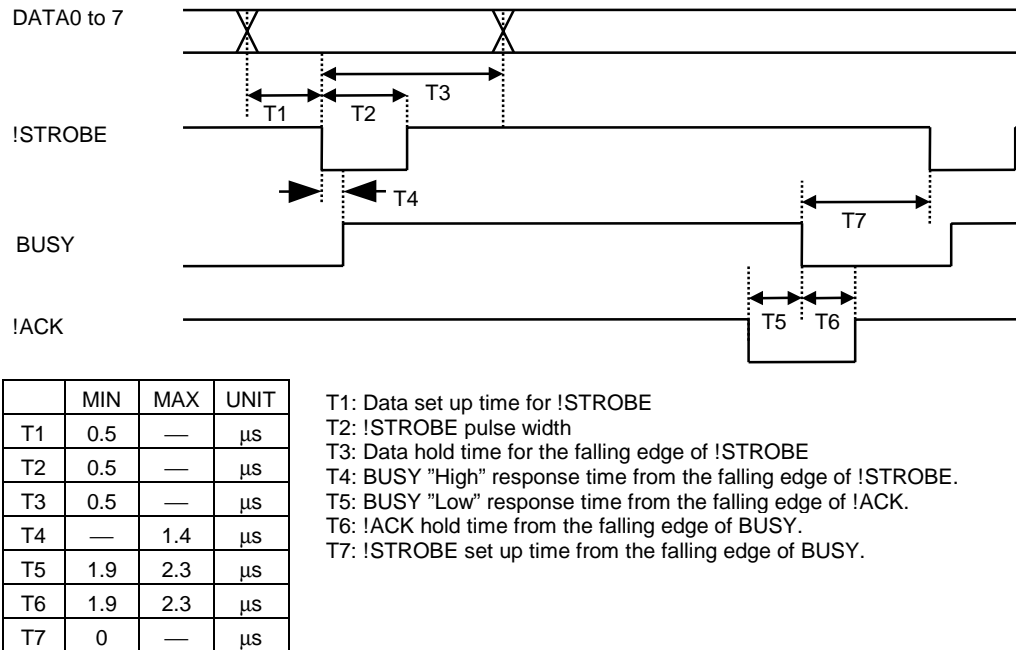


Figure 5-1 Timing for Receiving Parallel Data

## (2) Parallel input/output port

Strobe (!STROBE): Input Pin, CN6-2

- Triggers reading of 8-bit parallel data (DATA0 to DATA7).
- This signal is ignored when the BUSY signal is high.
- The !STROBE signal is normally high. The data on the data lines (DATA0 to DATA7) are placed into the buffer at the transition of the signal to low.

Data (DATA0 to DATA7): Input Pin, CN6-3 through 10

- Carries 8-bit parallel signals from the host device.
- When High is 1 and Low is 0, the data on these lines are placed by the !STROBE signal.



Busy (BUSY): Output Pin, CN6-12

- The BUSY signal indicates that the IF2202-02B is ready to receive data.
- When the BUSY signal is low, data can be input. This signal remains high from the transition of the !STROBE signal to low until the time when the data input processing is complete to indicate that the IF2202-02B is unable to receive data.

Acknowledge (!ACK): Output Pin, CN6-11

- The !ACK signal indicates that data reception is complete.
- The !ACK signal is normally high. A pulse approximately 10 $\mu$ s wide is generated when one-byte of data has been input.
- The IF2202-02B generates this signal to indicate to the host device that it can receive further input data.

## 5.1.2 Serial Input/Output

### (1) Timing for receiving serial data

The data input/output between the computer and the IF2202-02B is controlled as shown in **Figure 5-2**.

The IF2202-02B continues to receive data during print operation.

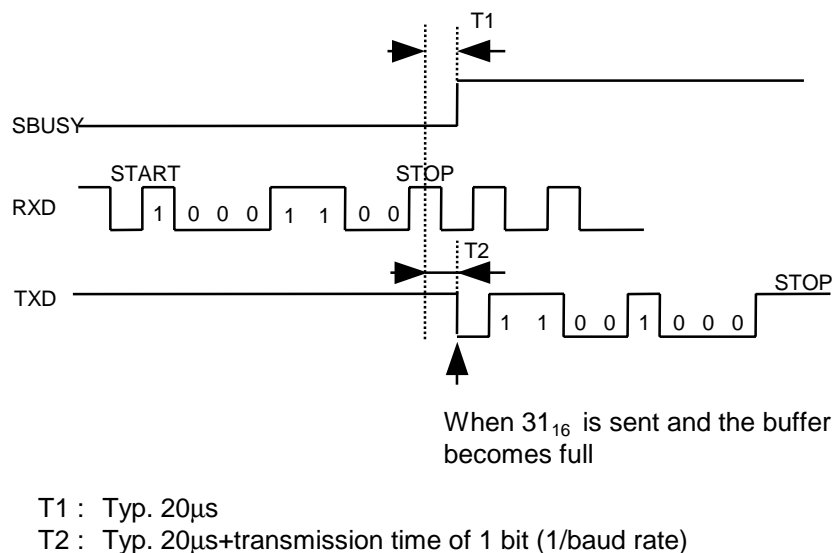


Figure 5-2 Serial Input/Output Signal Timing Chart

- **If BUSY control is selected:**

When 236 bytes or more of data are stored in the input buffer, the SBUSY signal (CN6-25) switches to high and requires the host device to temporarily stop sending data. When the amount of data stored in the input buffer becomes 235 bytes or less, the SBUSY signal switches to low and requires the host device to resume data transfer. Up to 16 bytes of input data are guaranteed after the SBUSY signal has become high.

When an error occurs, the SBUSY signal switches to high. Data input is prohibited until the error is canceled.

- **If Xon/Xoff control is selected:**

When 188 bytes or more of data are stored in the input buffer,  $X_{off}$  ( $13_{16}$ ) is output through the TxD terminal (CN6-23) and requires the host device to temporarily stop sending data. When the amount of data stored in the input buffer becomes 124 bytes or less,  $X_{on}$  ( $11_{16}$ ) is output and requires the host device to resume data transfer. Up to 64 bytes of the input data are guaranteed after  $X_{off}$  has been output. When an error occurs, the  $X_{off}$  signal is output to prohibit the data input. When the error is canceled,  $X_{on}$  is output.

## (2) Serial data transfer

The IF2202-02B transfers various data other than the  $X_{on}$  and  $X_{off}$  codes which are output when controlling  $X_{on}$  and  $X_{off}$ .

Regardless of whether the input mode is serial or parallel, the following data are transferred through the TXD terminal:

- (a) error codes when a hardware error occurs at initialization
- (b) when (DC2+'v') the Vp voltage response is executed
- (c) when (DC2+'e') the error status response is set and an error occurs
- (d) when (DC2+'q') the execution response request is executed
- (e) when (DC2+'r') the remaining RAM capacity response is executed
- (f) when (DC2+'s') the expanded RAM check sum response is executed
- (g) when (DC2+'h') the thermal head response is executed

At the selection of serial input, data (b) through (g) are transferred according to the transmission conditions which are set using the function switches. However, as for hardware error code (a), regardless of whether the selection is of serial input, the transmission conditions for the hardware error code are the same as those of parallel input.

Moreover, when transferring data, data control by SBUSY and Xon/Xoff is not executed, and the data is freely transferred.

All of the transmission conditions of serial data transfer for sending hardware error codes are fixed at the selection of parallel input as follows:

Baud rate: 2400 bps  
Parity: none  
Bit length: 8 bits  
Stop bit 1 bit

For the hardware error codes, see **Section 7.1, HARDWARE ERRORS**.

### (3) Serial input/output port

Since the IF2202-02B serial input/output signals (RXD, SBUSY, and TXD) are at the TTL level, a level converter must be used when using RS-232C signal. For examples of level converters, see **Section 6.4.2, Connecting for Serial Input/Output.**

Serial data output (TXD) : Output pin, CN6-23

- When Xon/Xoff control is selected, the Xon/Xoff signal is output.
- Data is output according to the transmission conditions which are set by the function switches.
- All response data is output.

Serial data input (RXD) : Input pin, CN6-24

- Data input port
- Data is input from the host device according to the transmission conditions which are set using the function switches.

Serial busy (SBUSY) : Output pin, CN6-25

- Indicates whether or not the IF2202-02B is ready to receive data.
- When the SBUSY signal is low, data can be input.
- When Xon/Xoff control is selected, SBUSY is always low.

### (4) Setting the transmission conditions

The IF2202-02B can set the serial input/output conditions from the following choices. However, stop bit and data bit length are fixed and can not be changed.

Baud rate: 2400 bps, 4800 bps, 9600 bps, 19200 bps, 38400 bps  
Parity: nothing, odd number, even number  
Data control: Busy, Xon/Xoff  
Data bit length: 8 bits (fixed)  
Stop bit length: 1 bit (fixed)

For the setting method for the transmission conditions, see **Section 6.1, SETTING FUNCTION SWITCHES.**

### (5) Error processing when receiving serial data

The IF2202-02B receives and checks serial data according to the transmission conditions.

When the IF2202-02B has received one byte of data without errors, the data is stored in the input buffer. If there are any errors, the following data is stored in the input buffer according to the type of error.

<u>Error type</u>	<u>Error code</u>
Parity error	(!: 21 <sub>16</sub> )
Framing error	(?: 3F <sub>16</sub> )

If the input data cannot be printed correctly and instead “!” or “?” is printed, the transmission conditions between the host device and the IF2202-02B most likely does not match. If this happens, adjust the conditions so that they match.

## 5.2 THERMAL HEAD CONTROL

### 5.2.1 Data Transfer to the Thermal Head

The IF2202-02B transfers one dot line of data at 3.125 Mbps, synchronized with the CLOCK signal. The data is transferred in order to the shift register inside the thermal head from the left (when facing the paper feed direction).

The transferred data is then transferred by the head latch signal to the latch register inside the thermal head. Turning on the head strobe signal (DST1 and 2) initiates printing of one dot line of data on the thermal paper.

### 5.2.2 Thermal Head Drive Operation

Generally, when the line thermal head is operating, the line is divided into several blocks which are activated one after another in succession.

For the line thermal head of the LTP2000 series printer mechanisms, the line is divided into 2 blocks called physical blocks. A strobe signal (DST1 and DST2) is allocated to each physical block to activate it. To drive the head, physical blocks are activated in groups. The group of physical blocks is called a logical block.

For the IF2202-02B, either dynamic division or fixed division can be selected as the method of division for logical blocks. This selection is made through a function switch or a command.

For the selection by the commands, see **Section 8.4.10, Print Method**.

For the selection by the function switches, see **Section 6.1, SETTING FUNCTION SWITCHES**.

#### (1) Dynamic division

The IF2202-02B counts the number of dots activated in each physical block and groups the physical blocks into logical blocks to print a single dot line so that the number does not exceed the specified maximum number of activated dots. The IF2202-02B can set a maximum activated dots in 72 dots unit and determines logical blocks each time it prints a single dot line.

In dynamic division, since the order of the printing blocks and printing speed are changed in each dot line according to the content of the print data, print quality may be lower than that in fixed division. Use care that unevenness of paper feed amount and print disorder may be caused due to print image. If print quality is regarded as important, printing in fixed division is recommended.

The maximum number of activated dots in the initialization status is specified using the function switches. 72 through 288 dots can, however, be set using the command for setting the number of dynamic division dots (DC2+'%'+n).

When the maximum number of activated dots is 144 dots, and all of the dots are driven, as shown in **Figure 5-3**, the IF2202-02B activates the two logical blocks groups of physical blocks in order, so that the number does not exceed 144 dots.

#### NOTE

When the maximum number of activated dots is set to less than 144 dots through dynamic division command or function pins, print disorder and light printing may occurred in low temperature.

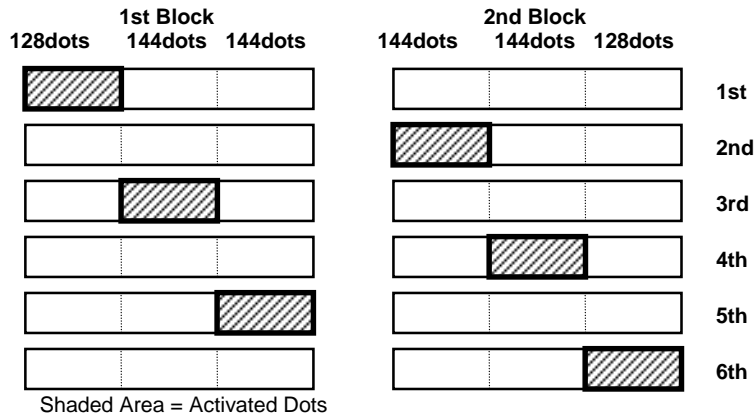


Figure 5-3 Dynamic Division When All Dots are Activated for LTP2442  
(Maximum Number of Activated Dots=144)

The advantage of dynamic division is that the maximum number of activated dots can be specified according to the capacity of the power supply used. The printer can be operated with a small capacity power supply when a small number is specified for the maximum number of activated dots. However, this process may increase the number of divisions and may lower print speed for printing at a high print ratio. On the other hand, printing speed becomes faster when the maximum number of activated dots is larger, requiring, however, a larger power supply.

**(2) Fixed division**

In fixed division, the logical blocks (a group of physical blocks simultaneously activated) are determined beforehand. Fixed division always activates physical blocks in the same order, thereby allowing higher print quality.

When a common activated dot number exceeds 720 dots in LTP2442, activated dots are divided and driven quasily. (ie. 1st: 720 dots, 2nd: remaining dots exceeded 720 dots)

In fixed division, a large power supply is needed because a large number of dots is activated at the same time.

### **5.2.3 Head control circuit**

#### **(1) Resistance measurement circuit**

The IF2202-02B has a function for measuring the resistance of the thermal head connected to the LTP2000 series printers.

The IF2202-02B measures the resistance of the thermal head and detects the errors at initialization.

Based on the measurement, the IF2202-02B determines how much energy is to be applied. No adjustment is needed for replacing the LTP2000 series printers to get the best printing.

#### **(2) Vp interrupt circuit**

This circuit interrupts Vp to prevent electrolytic corrosion of the thermal head and to enable the detection of the resistance mentioned above.

Electrolytic corrosion may significantly shorten the life of the thermal head. Thermal paper ordinarily contains electrolytic material to prevent sheets from sticking to each other due to static electricity. If there is too much of this electrolytic material, high temperatures and humidity cause the material to ionize, resulting in electrolytic corrosion of the thermal head.

The IF2202-02B turns the relay off and breaks the Vp applied to the thermal head during waiting status to prevent electrolysis corrosion of the thermal head.

If the print conditions are met after inputting the data is completed, the IF2202-02B turns the relay on and start printing. If the print conditions are not met for 5 seconds after printing is completed, the IF2202-02B turns the relay off.

#### **(3) Head temperature measurement circuit**

The thermistor is mounted on the LTP2000 series printers to measure the temperature of the thermal head.

The measures the temperature of the thermal head by receiving the signal of the thermistors.

The IF2202-02B determines the energy to be applied to the head and also checks for head temperature errors.

If the temperature of the thermal head is  $-10^{\circ}\text{C}$  or lower, or  $80^{\circ}\text{C}$  or higher, the IF2202-02B stops drive and puts the printer in head temperature error status. If the temperature of the thermal head is returned to from  $-5^{\circ}\text{C}$  to  $75^{\circ}\text{C}$ , the printer goes to printable status.

Although the temperature of the head is monitored by the CPU, the hardware also can stop thermal head output by sending a strobe signal from the comparator, such as in the case when the CPU loses operational control.

#### 5.2.4 Adjustment of Applied Energy

The IF2202-02B automatically controls the energy applied to the thermal head according to the resistance, temperature of the thermal head and the Vp voltage.

Print density differs depending on the thermal paper used.

Print density can be changed using the function switch or the command to compensate for the difference in print density due to the paper used. Applied energy can be adjusted to 100% or 130% of the rated energy by the function switches (See **Section 6.1, SETTING FUNCTION SWITCHES**). Applied energy can be adjusted from 65% to 135% of the rated energy by the print density specify command (DC2 + '~'+n) (See **Section 8.4.10, PRINTING METHOD**).

See the Technical Reference of the LTP2000 series printer mechanisms for the setting of the applied energy on the recommended paper.

#### 5.2.5 2-ply Thermal Paper Print

A 2-ply thermal paper is a paper which is stucked a low thermal sensitive thermal paper on a high thermal sensitive thermal paper and can be printed on two sheets upper and lower at one printing.

A 2-ply thermal paper needs higher energy to print than a normal thermal paper because a 2-ply thermal paper is composed of two layers.

Therefore, a print control for the 2-ply thermal paper differs from one for the normal thermal paper.

Print modes for normal papers or 2-ply thermal papers can be changed through function switch settings.

For the selection by the function switches, see **Section 6.1, SETTING FUNCTION SWITCHES**.

### **5.3 PRINTER MECHANISM CONTROL**

#### **5.3.1 Paper Feed Step Motor Drive**

The paper feed motor of the LTP2000 series printers is a four-phase, bipolar type stepping motor. Paper is fed for a single dot line by 2-2 phase excitation, 1 step.

The drive speeds of the motor are 720 pps for LTP2242 and 600 pps for LTP2342 and LTP2442. Acceleration control is executed to prevent the printing pitch from being too small and assure the needed paper feed at the start. Acceleration is automatically executed to the drive speed, then normal-speed operation starts.

When the drive time of the head is longer than one step of the motor, the motor is driven after the driving of the head is complete. Therefore, the motor drive speed decreases during printing according to the division method of the head and the head temperature.

#### **NOTE**

If excessive load is applied to the paper during paper feed and printing, the printer may step out and cannot return to the normal status. The paper supply load to the printer must be 50 gf or less.



### 5.3.2 The Timing for Driving the Head and Motor

Printing is executed by driving the motor while driving the head. The timing for driving the head in two division and the timing for driving the motor are shown in **Figure 5-4**.

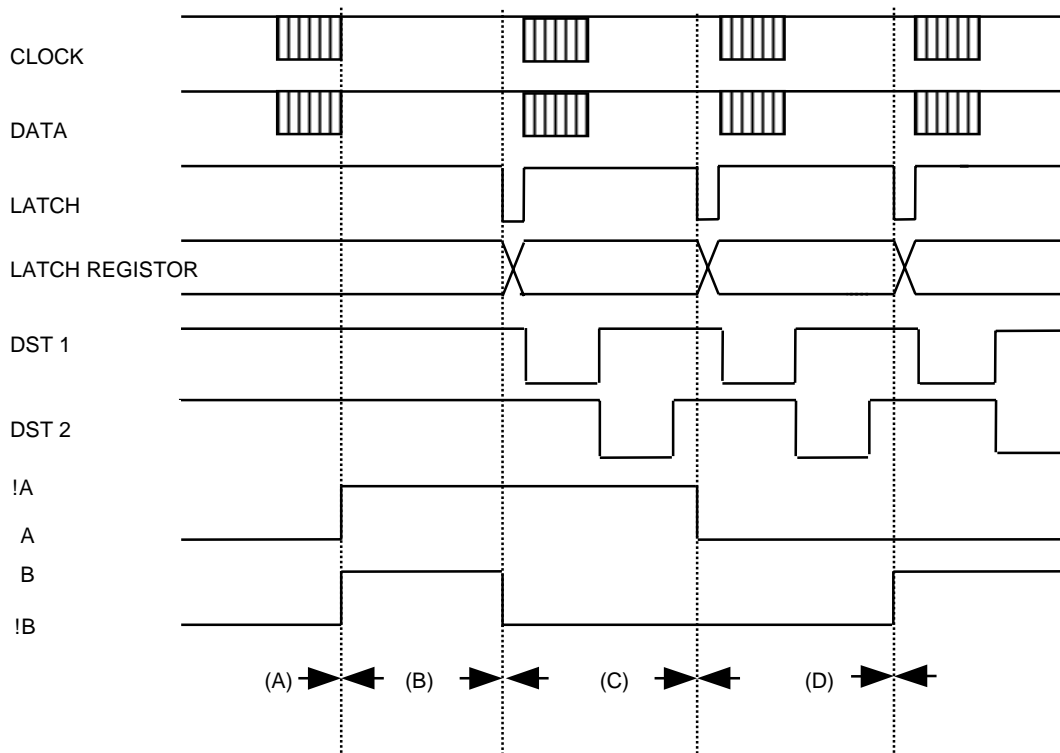


Figure 5-4 The Timing for Driving the Head and the Motor

- (A) Pause state: The IF2202-02B transfers the data to the head to be printed in the first dot line.
- (B) Start step output: The IF2202-02B generates the same phase as the one which was generated just before the printer had paused.
- (C) First step output: The IF2202-02B latches the data transferred in step (A) and prints by driving DST1 and DST2 signals.
- (D) Second step output: The IF2202-02B latches the data transferred in step (C) and prints by driving DST1 and DST2 signals.

The IF2202-02B executes the remaining steps in the same manner: activates the head synchronized with the motor driving, to transfer the data to the head with every step.

### 5.3.3 Status Detection

#### (1) Paper Empty Detection

The IF2202-02B uses the input signal from the photo-interrupter mounted in the printer mechanism to check for paper.

The IF2202-02B determines the paper existence by the PS terminal (CN5: pin 2).

When the paper empty is detected, the IF2202-02B stops mechanism driving and enters the out-of-paper error status after the current line is printed. If the paper is installed when the printer is in out-of-paper error status and head down status, the printer executes the auto-loading and turns to the waiting status. Set the !FEED terminal (CN6: pin 20) to "High" from "Low" or down the thermal head after lifting it to return the print ready status. The printer resumes printing from the next line at this time.

The printer does not execute the auto-loading when the printer is in the out-of-paper error status and the head up status. The printer returns to the print ready status after approx. one sec. when the paper is installed on the printer and then put the thermal head down.

See **Section 5.3.4, Mechanism Status Output** about the error signals for the out-of-paper error and the waiting status.

See **Section 6.5.1, Loading paper** about the auto-loading.

#### (2) Head-up detection

The IF2202-02B uses the signal from the mechanical switch mounted in the printer mechanism to check whether the thermal head is up or down.

The head-up is detected through the HS terminal (CN5: pin 5). When the printer is in the head-up status, the printer stops mechanism driving and turns to the head-up error status after completion of the current line printing. When the head returns to the down position, the IF2202-02B resumes printing about one second later from the next line.

See **Section 5.3.4, Mechanism Status Output** about the error signals for the head-up error.

### 5.3.4 Mechanism Status Output

The following five signals show the status of the printer mechanism:

PE	(CN6: pin 14)
!ERR	(CN6: pin 15)
ST0	(CN6: pin 16)
ST1	(CN6: pin 17)
ST2	(CN6: pin 18)

See **Chapter 7, ERROR PROCESSING** for the details of each error status and how to clear it. This section explains printer status.

- During Printing : The printer mechanism is in operation.  
The operation means both printing and command paper feed status.  
The paper feed by !FEED signal (CN6: pin 20) does not turn to this status.
- Print Ready status : The printer is in normal ready status for data input and printing.

**Table 5-1** shows each signal level for each printer status.

Table 5-1 Printer Status Signals

Printer Status	PE	!ERR	ST0	ST1	ST2
Hardware error	Low	Low	Low	Low	Low
Head-up error	Low <sup>*1</sup>	Low	High	Low	Low
Vp voltage error	Low <sup>*1</sup>	Low	Low	High	Low
Auto-cutter error	Low <sup>*1</sup>	Low	High	High	Low
Thermal head temperature error	Low <sup>*1</sup>	Low	Low	Low	High
Return-Waiting status	Low	Low	High	Low	High
Out-of-paper error	High	Low	High	Low	High
During printing	Low	High	Low	High	High
Print-ready status	Low	High	High	High	High

When more than one operation error (operation errors: head-up error, Vp voltage error, auto-cutter error, thermal head temperature error and out-of-paper error) occur, only the topmost error listed in **Table 4-2** is displayed.

For example, when head-up error and Vp voltage error occur, only head-up error is displayed. When Vp voltage error and thermal head temperature error occur, only Vp voltage error is displayed.

<sup>\*1</sup>: When out-of-paper error occurs with other errors, the PE signal turns to “High”.

## 5.4 AUTO-CUTTER FUNCTION

### 5.4.1 Auto-cutter Operation

The IF2202-02B has a function to control the auto-cutter ACU2\*24 for the LTP2000 series printers. The IF2202-02B connected with auto-cutter ACU2\*24 can execute both Partial cutting (one point uncut) and Full cutting (three points uncut). The cutting is executed through the full cut or partial cut command (Auxiliary Function “ESC+i” and “ESC+m”). Read the “**ACU2000 series Auto-cutter Unit Technical Reference**” about the details of the auto-cutter ACU2000 series.

### 5.4.2 Auto-cutter Initialization

The auto-cutter is initialized either through power on or through reset by hardware or software. When the movable blade of the cutter is outside the paper-through window, the cutter-drive motor automatically rotates, returning the movable blade to the home position.

#### NOTE

Error status is not indicated if the movable blade is locked at the home position at initialization.

### 5.4.3 Unlocking of Movable Blade

If the movable blade is locked due to a paper jam or other error and the cutter does not operate, unlock the movable blade by following the procedure below:

#### 1. When the movable blade is at the home position

The IF2202-02B has a function by which the drive motor for the movable blade is rotated in the reverse direction to automatically return the locked blade to the home position. Remove the cause of the lock (such as paper jam) after turning off the power, then turn on the power again.

#### 2. When the movable blade is not at the home position

Turn off the power.

Remove the cause of the lock and reset the IF2202-02B.

Check that the movable blade has been returned to the home position by initialization.

In case the movable blade has not been returned to the home position, turn off the power, insert a Philips screwdriver as shown in **Figure 5-5** and rotate the cross recess on the tip of the worm-gear on the motor shaft to return the movable blade from the paper-through window to inside the unit.

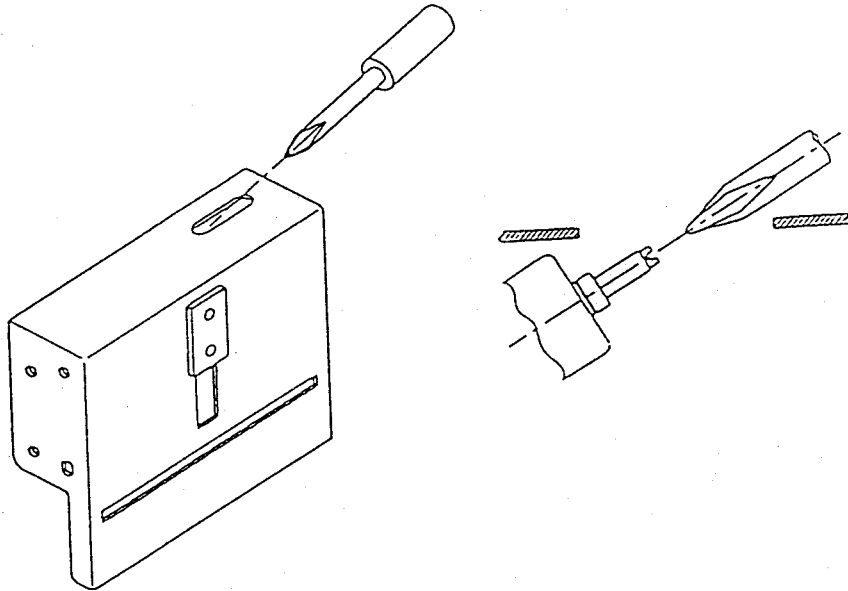
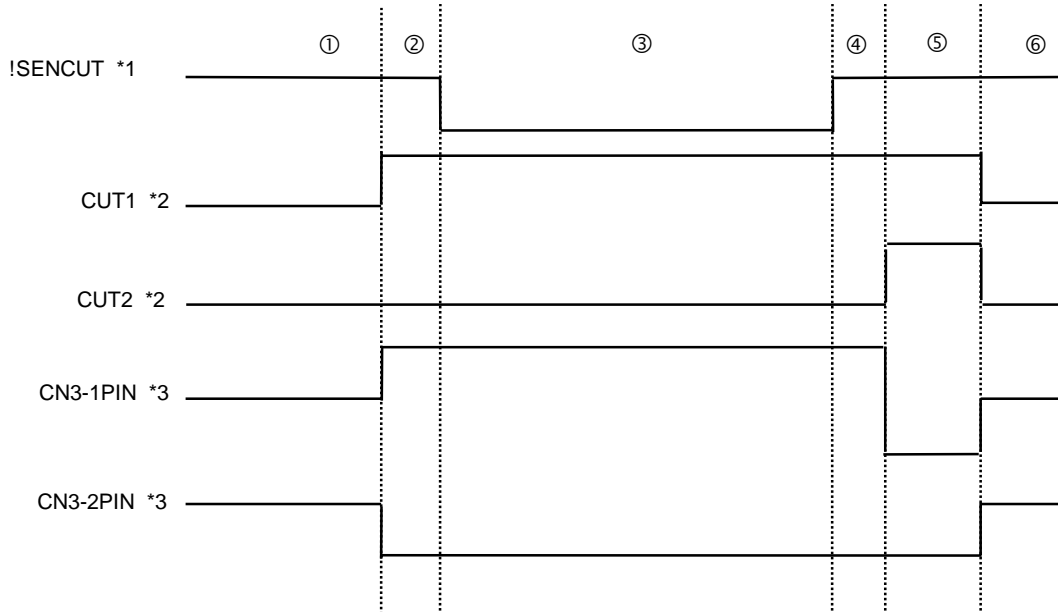


Figure 5-5 Returning the Movable Blade to Inside the Unit

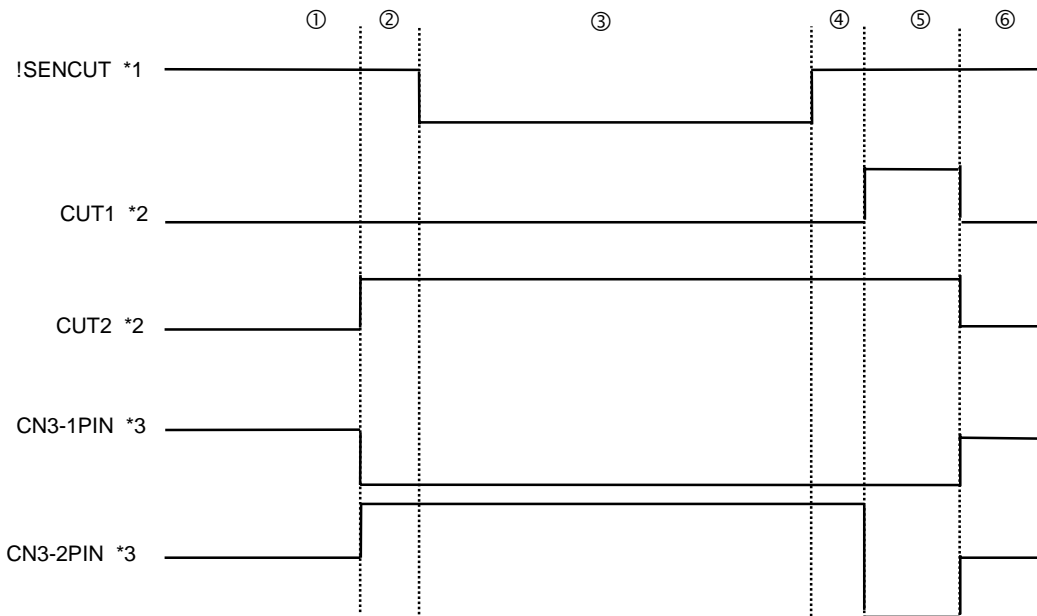
### 5.4.4 Auto-cutter Control Timing

The control timing when ACU2000 series auto-cutter is connected to the IF2202-02B is shown in **Figure 5-6**.

#### Full cut



#### Partial cut



- ① Stand-by
- ② Drive the movable blade
- ③ Cut the paper
- ④ Home position detect
- ⑤ Brake the drive motor
- ⑥ Stand-by

\*1: CPU input signal \*2: CPU output signal \*3: Cutter driver output signal

Figure 5-6 Auto-cutter Control Timing



Mount a DIP switch or connect the jumper wires by soldering for setting function switches.

All jumper wires are opened at shipping.

when mounting the DIP switch, mount 8-bit DIP switches onto the DIP1 and DIP2 locations marked on the board respectively.

Hereafter, a short circuit and an open circuit, by DIP switches settings and soldering jumper wires, are shown "ON" and "OFF" respectively.

The settings of DIP1 and DIP2 are read in at initialization just after powering on and resetting the IF2202-02B. Therefore, turn the power of IF2202-02B from off to on or input the reset signal when the settings are changed.

The function switches for the IF2202-02B consist of sixteen bits from DIP1-1 to DIP2-8. In each bit, the function is assigned as shown in **Table 6-1**.

Table 6-1 Function switches

Terminal Name	Function	OFF	ON
DIP1 to 3	Printer model selection	See <b>Table 6-2</b>	
DIP1-4	2-ply thermal paper mode selection	Normal	2-ply
DIP1-5	Thermal paper selection	Normal	PD160R-N
DIP1-6*	Print density selection	100 %	130 %
DIP1-7	Print speed selection	High	Low
DIP1-8	Thermal head error output unit selection	1 block unit	1 dot unit
DIP2-1 to 4	Data input method selection	See <b>Table 6-8</b>	
DIP2-5	Data control method selection	Busy	Xon/ Xoff
DIP2-6*	Thermal head division method selection	Dynamic division	Fixed division
DIP2-7 and 8*	Number of dots selection for dynamic division drive	See <b>Table 6-11</b>	

\*  
DIP1-6 for print density, DIP2-6 for thermal head drive method, and DIP2-7 and 8 for number of dots for dynamic division drive settings define default status of the printer and can be changed through commands.



- **Printer model selection**

The DIP1-1 to 3 are terminals to determine the printer model to be connected to the IF2202-02B.

Do not use the “inhibit” settings in **Table 6-2**.  
 “Inhibit” settings are settings for future expansion.

Always set function switches to usable settings.

Table 6-2 Setting Function Switches (Printer model)

Printer model	DIP1-3	DIP1-2	DIP1-1
Inhibit	ON : OFF	ON : ON	ON : ON
LTP2442	OFF	ON	OFF
LTP2342	OFF	OFF	ON
LTP2242	OFF	OFF	OFF

- **2-ply thermal paper mode selection**

The DIP1-4 is a terminal to determine the paper to be used.  
 See **Section 5.2.5, 2-ply Thermal Paper Print** for 2-ply thermal papers.

Table 6-3 Setting Function Switches (2-ply thermal paper mode)

Terminal Name	Functions	OFF	ON
DIP1-4	2-ply thermal paper mode selection	Normal	2-ply

- **Thermal paper selection**

Selects a thermal paper for print.  
 Normal thermal paper is TF50KS-E2C made by Nippon Paper Industry.  
 A higher energy than that of the normal thermal paper is applied to PD160R-N (Oji Paper).

**NOTE**

When both DIP-1-4 and DIP1-5 are set to ON, priority is given to DIP1-5, therefore, the thermal paper that is being used is detected as PD160R-N thermal paper.

Table 6-4 Setting Function Switches (Paper)

Pin Name	Functions	OFF	ON
DIP1-5	Thermal paper selection	Normal	PD160R-N

- **Print density selection**

Selects a default print density.

The print density can be changed from 65% to 135% through a print density command. (See **Section 8.4.10, Print Method**, DC2+'~'+n).

Table 6-5 Setting Function Switches (Print density)

Pin Name	Functions	OFF	ON
DIP1-6	Print density selection	100%	130%

- **Print speed selection**

Selects a print speed.

The printer operates at the maximum print speed in the “High” setting.

The print speed is changed due to a print ratio and thermal head drive methods.

The printer operates at 50 mm/s (400pps) in the “Low” setting.

The low speed print can get a better print quality than high speed prints one.

We recommend the low speed print when the print quality is more important than print speed.

Table 6-6 Setting Function Switches (Print speed)

Pin Name	Functions	OFF	ON
DIP1-7	Print speed selection	High	Low

- **Thermal head error output unit selection**

Selects the detection unit of the thermal head open/short error at the initialization directly after powering on and reset from 1 physical block or 1 dot.

The block unit detection for normal character printing and 1 dot unit detection for bar code printing are recommended.

Table 6-7 Setting Function Switches (Thermal head error output unit)

Pin Name	Functions	OFF	ON
DIP1-8	Thermal head error output unit selection	1 block	1 dot

- **Data input method selection**

The data input method is selected through the setting of the DIP2-1 to 4 as follows:

The data input method is set only through this setting.

When selecting serial input method, data control method should be selected through the setting of the DIP2-5.

When transferring at 38,400bps, a host computer needs to transfer data with  $\pm 2\%$  accuracy.

When selecting parallel method, the setting of the DIP2-5 is ignored.

Table 6-8 Setting Function Switches (Data input method)

Input method	Parity bit	Baud rate	DIP2-4	DIP2-3	DIP2-2	DIP2-1
Serial	None	2400 bps	ON	ON	ON	ON
Serial	None	4800 bps	ON	ON	ON	OFF
Serial	None	9600 bps	ON	ON	OFF	ON
Serial	None	19200 bps	ON	ON	OFF	OFF
Serial	None	38400 bps	ON	OFF	ON	ON
Serial	Odd	2400 bps	ON	OFF	ON	OFF
Serial	Odd	4800 bps	ON	OFF	OFF	ON
Serial	Odd	9600 bps	ON	OFF	OFF	OFF
Serial	Odd	19200 bps	OFF	ON	ON	ON
Serial	Odd	38400 bps	OFF	ON	ON	OFF
Serial	Even	2400 bps	OFF	ON	OFF	ON
Serial	Even	4800 bps	OFF	ON	OFF	OFF
Serial	Even	9600 bps	OFF	OFF	ON	ON
Serial	Even	19200 bps	OFF	OFF	ON	OFF
Serial	Even	38400 bps	OFF	OFF	OFF	ON
Parallel	—	—	OFF	OFF	OFF	OFF

- **Data control method selection**

Selects a data control method for serial data input.

Table 6-9 Setting Function Switches (Data control method)

Pin Name	Functions	OFF	ON
DIP2-5	Data control method selection	Busy control	Xon/Xoff

- **Thermal head division method selection**

Selects the thermal head division method from dynamic or fixed through the setting of the DIP2-6. When the dynamic is selected, the maximum number of the drive dot can be set by the DIP2-7.8. Thermal head division method can be changed through a command.

Table 6-10 Setting Function Switches (Thermal head division method)

Pin Name	Functions	OFF	ON
DIP2-6	Thermal head division method selection	Dynamic	Fixed

- **Number of dots selection for dynamic division drive**

When selected the dynamic division, the maximum number of the drive dot can be set in the range from 72 to 432 dots.

The setting of the dynamic division and the number of the drive dots can be changed through a command.

Table 6-11 Setting Function Switches (Number of dots for dynamic division)

Functions		DIP2-8	DIP2-7
Dynamic	72 dots	ON	ON
Dynamic	144 dots	ON	OFF
Dynamic	288 dots	OFF	ON
Dynamic	432 dots	OFF	OFF

## 6.2 CONNECTING TO THE PRINTER MECHANISM

To connect the IF2202-02B to an LTP2000 series printer mechanisms, the following three cables are connected. When the auto-cutter is used, four cables are connected.

(1) Thermal head cable (supplied as an accessory for the IF2202-02B)

Plug the cable connectors into both the connector on the thermal head and the CN2 connector on the IF2202-02B.

(2) Motor cable

Plug the connector on the motor cable into the CN4 connector.

(3) Paper empty and Head up detectors cable

Plug the connector on the paper empty and head-up detectors cable into the CN5 connector.

(4) Auto-cutter cable (Only when the auto-cutter is used.)

Plug the connector on the auto-cutter cable into the CN3 connector.

### **Precaution When Connecting to the GND of the Interface Board**

To prevent the electrolytic etching of the thermal head, connect the FG of the printer mechanism to the GND of the interface board so that the conductive portion on the FG line of the printer mechanism and the FG line are the same potential as that of the reference potential (GND) of this interface board.

### 6.3 CONNECTING TO THE POWER SUPPLY

#### 6.3.1 Selecting the Power Supply

The IF2202-02B requires two power sources:  $V_P$  (24 V) for driving the thermal head and motor, and  $V_{CC}$  (5 V) for logic electronics. The power supplies must satisfy the following conditions.

$V_{CC}$  (for logic electronics):  $5 \pm 0.5$  V, 0.3 A or more

$V_P$  (for thermal head and motor):  $24 \pm 2.4$  V

The maximum current consumption of  $V_P$  varies according to the used voltage and print method. Select the power supply according to the used voltage and print method.

- **When the LTP2000 series printer is driven with fixed division:**

Table 6-12 Current Consumption When Driving with Fixed Division

LTP2242	LTP2342	LTP2442
6.8 A max.	9.0 A max.	11.1 A max.

- **When the LTP2000 series printer is driven with dynamic division:**

The maximum current consumption  $V_P$  is calculated with a maximum activated current of one dot on the thermal head shown in **Table 6-13** and a maximum activated current of the motor from the following equation:

$$I_P (\text{max.}) = N \times I_{\text{dot}} + I_m \quad (\text{A})$$

N: Maximum number of activated dots.

I dot: Maximum activated current for one dot.

I<sub>m</sub>: Maximum activated current for the motor.

Table 6-13 Maximum Activated Current

Maximum activated current for one dot	Maximum activated current for motor	
	LTP2242	LTP2342 LTP2442
14.7 mA	360 mA	500 mA

#### 6.3.2 Turning Power On/Off

Power must be turned on in proper order. Turn on both  $V_{CC}$  and  $V_P$  simultaneously, or turn on  $V_{CC}$  first and then  $V_P$ . Otherwise, the IF2202-02B may be damaged.

To turn the power off, first turn off  $V_P$  and then  $V_{CC}$  or turn off both  $V_{CC}$  and  $V_P$  simultaneously.

## 6.4 CONNECTING TO THE HOST DEVICE

### 6.4.1 Connecting for Parallel Input/Output

For the parallel input/output, CN6 terminals 2 to 12, 14, 15 and GND are used.

Table 6-14 Parallel Input/Output Terminals (CN6)

Terminal No.	Signal Name	I/O	Function
2	!STROBE	I	Parallel strobe signal
3	DATA0	I	Parallel data signal
4	DATA1	I	Parallel data signal
5	DATA2	I	Parallel data signal
6	DATA3	I	Parallel data signal
7	DATA4	I	Parallel data signal
8	DATA5	I	Parallel data signal
9	DATA6	I	Parallel data signal
10	DATA7	I	Parallel data signal
11	!ACK	O	Parallel, Acknowledge signal
12	BUSY	O	Parallel, Busy signal
14	PE	O	Paper empty signal
15	!ERR	O	Error signal

“High” is 1  
“Low” is 0

Handshaking is performed using !STROBE, DATA0 to 7 and BUSY, or !ACK. The !STROBE and DATA0 to 7 signals are CMOS input/output.

The !ACK, BUSY, PE, and !ERR signals are open drain output which was pulled up through the resistance (2.2 k $\Omega$ ).

Push the protection circuit separately in accordance with the connection with the host device.

For the latch-up phenomenon, see **Section 1.2, PRECAUTIONS**.

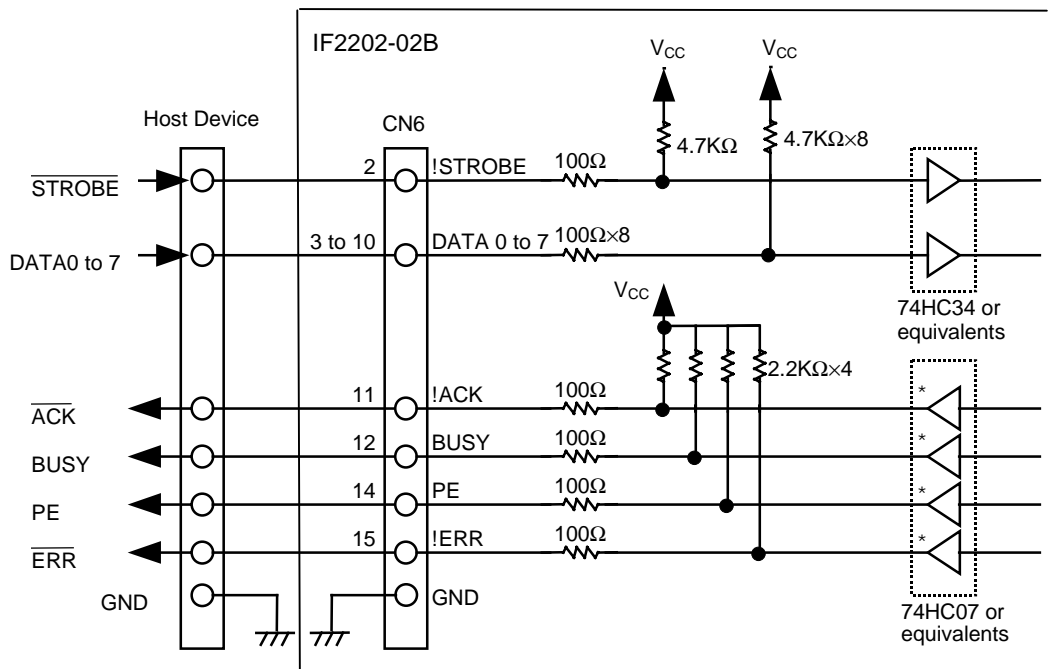


Figure 6-2 Parallel Input/Output Circuit Example

### 6.4.2 Connecting for Serial Input/Output

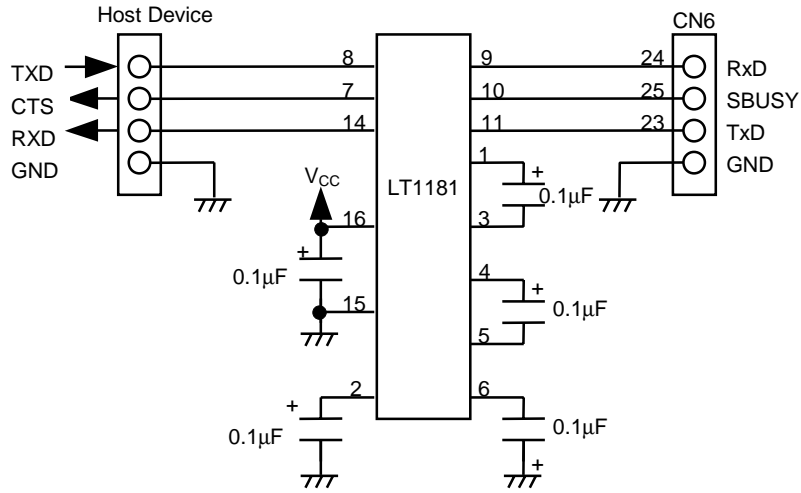
For the serial input/output, CN6 terminals 23 to 25 and GND are used.

Table 6-15 Serial Input/Output Terminals (CN6)

Terminal No.	Signal Name	I/O	Function
23	TxD	O	Serial data output signal
24	RxD	I	Serial data input signal
25	SBUSY	O	Serial busy signal

Handshaking is performed using SBUSY, or the Xon/Xoff signal which is output from TxD and RxD. These signals are TTL level. When using with RS-232C, level conversion is required. The level convert circuit example is shown in **Figure 6-3**.





IC used: LT1181 (LINEAR TECHNOLOGY)

Figure 6-3 RS-232C Level Convert Circuit Example

### 6.4.3 Connecting to Switches and LEDs

To connect to the reset, feed and back feed switches, CN6 terminals 19, 20, and 21 are used. To connect to the LEDs for displaying the printer status, CN6 terminals 14 to 18 are used.

Table 6-16 Switch and Error Terminals (CN6)

Terminal No.	Signal name	I/O	Function
14	PE	O	Paper empty signal
15	!ERR	O	Error signal
16	ST0	O	Printer status 0 signal
17	ST1	O	Printer status 1 signal
18	ST2	O	Printer status 2 signal
19	!INIT	I	Reset signal
20	!FEED	I	Feed signal
21	!BFEEED	I	Back feed signal

The !INIT, !FEED and !BFEEED signals are pulled up by  $V_{cc}$  at a resistance of 10 k $\Omega$ . When connecting the switch externally, it is possible to operate the IF2202-02B only if the switch are connected between the signals and GND, as shown in **Figure 6-4**.

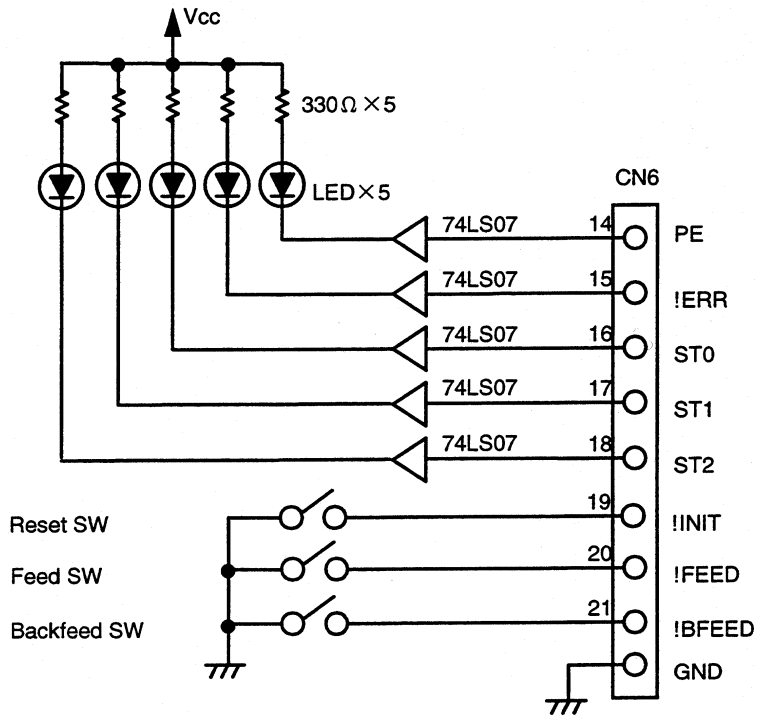


Figure 6-4 Switch and Display Circuit Example

## **6.5 OPERATING**

### **6.5.1 Loading Paper**

There are two ways to load the paper in the LTP2000 series printers: automatically loading paper in the head-down status, and manually loading paper in the head-up status.

#### **(1) Automatically loading paper**

The IF2202-02B has an auto-loading function for replacing paper easily. Auto-loading function loads paper automatically when paper is inserted at the paper empty status and the head is down. When the head is up, the auto-loading function is not performed. When the paper empty error is set to be invalid using the paper empty select command (DC2 + 'p' +0), auto-loading is not performed.

The amount of paper sent in auto-loading is 480 dot lines (approx. 6 cm). In this case, the driving speed is 150 pps.

The operation of auto-loading is as follows:

Cut the edge of the paper so that the corners of the paper are square (90°).

1. The LTP4242 must have no paper.  
If there is paper in the LTP2000 series printers, remove the paper by feeding the paper either in the forward direction or the reverse direction.
2. Insert the paper into the paper inlet with the head in the down position until the paper stops, then the paper load automatically.
3. After auto-loading is complete, when the !FEED terminal is set to high from low or the head position goes up then down again, then printing is ready to start.  
When there is data to print, printing restarts from the next line.

#### **(2) Manually loading paper**

When the head is "up", it is possible to load paper manually.

Cut the edge of the paper so that the corners of the paper are square (90°) to the inserting direction.

1. The head must be in the "up" position.
2. Insert the paper into the paper entrance until the paper comes out of the paper exit.
3. Make sure that the paper is straight in the printer, then put the head in the "down" position.  
The print-available status returns about one second later.  
If there is data to print, the IF2202-02B resumes printing from the next line.

### 6.5.2 Paper Feed and Backfeed

IF2202-02B has a function to feed the paper to the forward or backward directions through external signals.

When the !FEED signal (CN6-20) is set to low, paper is fed in the forward direction.

When the !BFEED signal (CN6-21) is set to low, paper is fed in the backward direction.

The !FEED signal is used when canceling the return-waiting status after canceling a paper empty error, a  $V_p$  voltage error, or an auto-cutter error.

When the !FEED signal is set to high from low, the return-waiting status is canceled. In this case, the paper is not fed.

When the !FEED signal or !BFEED is set to low, the IF2202-02B operates as follows:

1. The IF2202-02B feeds paper one line space in the forward or backward direction.
2. After feeding the first line, operation is suspended for 0.5 second.
3. After this, the IF2202-02B continues to feed paper one line at a time until the signal input returns to high.

The amount of paper feed for one line is the amount of line spacing plus the character height which is set at that time. The amount of paper feed just after initialization is 40 dot lines (5 mm).

### 6.5.3 Test Print Procedure

The IF2202-02B has a test print function. The current settings of the function switches and all available characters are printed when a test print is performed. A test print sample is shown in **Figure 6-6**.

Through this function, you can simply confirm the current settings of the function switches and whether or not the IF2202-02B and the LTP2000 series printers are connected correctly.

Test print is performed by powering ON or releasing the reset while remaining !FEED signal (CN6-20 and 21) in "Low".

Test print is available even while lid open error status.

Test print procedure is as follows:

Set up the LTP2000 series printers, the IF2202-02B, and the power supply.

Turn the power ( $V_{cc}$  and  $V_p$ ) OFF while setting up.

#### (1) Test printing by turning the power on

Connect the !FEED signal to GND, and then turn on the power. The test print will start.

#### NOTE

When turning the power on, turn on  $V_{cc}$  and  $V_p$  simultaneously, or  $V_{cc}$  first and then  $V_p$ .



#### 6.5.4 HEX Dump Print Procedure

The IF2202-02B has a hex dump function which allows data to be printed in hexadecimal code. A hex dump sample is shown in **Figure 6-6**.

Through this function, you can confirm simply whether or not the data sent from the host device is received correctly or the correct data from the host device has been sent.

A hex dump print is performed by turning the power on or canceling the reset while the !BFEEED (CN6-21) signal is low.

The hex dump print procedure is as follows:

- (1) A hex dump print by turning the power on.

Set the !BFEEED signal to low, and turn on the power. A hex dump print will then start.

When turning the power on, turn either  $V_{CC}$  and  $V_P$  simultaneously, or  $V_{CC}$  first and then  $V_P$ .

- (2) A hex dump print by resetting.

Turn on the power, turn on  $V_{CC}$  and  $V_P$  (simultaneously, or  $V_{CC}$  first and then  $V_P$ ).

Set the !INIT signal to low. Set the !BFEEED signal to low, and the !INIT signal must be high. The hex dump will then start.

When "[HEX DUMP MODE]" is printed, return the !BFEEED signal to high.

The data after the !BFEEED signals returned to high is printed all in hexadecimal code.

The data is printed eight bytes each for LTP2242 and LTP2342 and sixteen bytes each for LTP2442.

Data less than eight bytes is printed when the !FEED signal (CN6-20) for the printer mechanism executing a hex dump print is switched to low.

```
[ HEX DUMP MODE ]
2A 20 50 41 52 41 4C 4C 45 4C 20 49 2F 46 20 53
41 4D 50 4C 45 2D 39 0D 0A 12 7E 46 50 72 69 6E
74 20 44 65 6E 73 69 74 79 20 37 30 20 25 0D 0A
12 7E 50 50 72 69 6E 74 20 44 65 6E 73 69 74 79
20 38 30 20 25 0D 0A 12 7E 5A 50 72 69 6E 74 20
44 65 6E 73 69 74 79 20 39 30 20 25 0D 0A 12 7E
64 50 72 69 6E 74 20 44 65 6E 73 69 74 79 20 31
30 30 20 25 0D 0A 12 7E 6E 50 72 69 6E 74 20 44
65 6E 73 69 74 79 20 31 31 30 20 25 0D 0A 12 7E
78 50 72 69 6E 74 20 44 65 6E 73 69 74 79 20 31
32 30 20 25 0D 0A 12 7E 82 50 72 69 6E 74 20 44
65 6E 73 69 74 79 20 31 33 30 20 25 0D 0A 12 7E
```

Figure 6-6 Hex Dump Print Sample (LTP2442)





## CHAPTER 7

### ERROR PROCESSING

Errors are classified into two types. The hardware errors are detected at the power ON or at the initialization just after the reset. The operation errors occur during operation. When an error occurs, data input is prohibited.

#### 7.1 HARDWARE ERRORS

If a hardware error occur, a hardware error code is sent from TxD (CN 6-23) and no other operation is performed. See **Section 5.1.2, Serial Input / Output**.

Hardware errors and the corresponding error codes are listed in **Table 7-1**. One byte of error code corresponding to each error cause is sent approximately every one second.

Table 7-1 Hardware Error Codes

Cause	Code
Peripheral circuit error	02
Head short error	03
Head open error	04
Printer model selection error	04

The causes of the errors and the solution for each error are listed below.

- **Peripheral circuit error**

When the CPU diagnosed a circuit, an error was detected.

→Non recoverable error.

- **Head short error**

a) the average resistance of the thermal head is 1000  $\Omega$  or lower, or

b) the dot resistance is 20  $\Omega$  or lower.

This means that the printer mechanism is damaged, or not correctly connected to the IF2202-02B and the printer.  
→Non recoverable error

- **Head open error**

**(1) When the head error output method is set to the block-unit:**

If the DIP1-8 is set to "High" :

- a) the average resistance of the thermal head is 3000  $\Omega$  or higher, or
- b) all dots of a certain physical block have become open

This means that the printer mechanism is damaged, or not correctly connected to the IF2202-02B and the printer.  
→ Non recoverable error.

**(2) When the head error output method is set to the dot-unit:**

If the DIP1-8 is set to "Low" :

- a) the average resistance of the thermal head is 3000  $\Omega$  or higher, or
- b) one dot or more of a certain physical block has become open

This means that the printer mechanism is damaged, or not connected to the IF2202-02B and the printer.  
→ Non recoverable error.

- **Printer model selection error**

The settings through function switch DIP1-1 to 3 is not correct for the printer used.  
→Set the function switch correctly and reset the IF2202-02B.

## 7.2 OPERATION ERRORS

Operation errors include: out-of-paper error, head-up error,  $V_p$  power supply error, thermal head temperature error, and auto-cutter error.

The type of operation errors can be judged from the status output from the PE, !ERR, and ST0 to 2 (CN6-14 to 18) signals. They can be output as an error code through the TxD terminal (CN4-23) using the error status response command (see **Section 8.4.11, Response Function**, DC2+ 'e' +1). (Excluding the auto-cutter error code)

Even when the operation error is in process, paper can be fed by using the !FEED terminal (CN6-20). On the other hand, since the data is buffered, printing can be resumed once the causes of the errors are removed and the errors are canceled. For information on !FEED and !BFEED signals, see **Section 6.5.2, Paper Feed and Back feed**.

When the out-of-paper error, the  $V_p$  voltage error, and the auto-cutter error occur, after canceling the error, the IF2202-02B goes to the return-waiting status until printing is ready to resume. The return-waiting status is set in order to avoid the following.

- No paper is in the head even if paper is inserted and the sensor finds there is paper inside.
- Instantaneous reduction of the  $V_p$  voltage by printing occurs and as a result, the  $V_p$  voltage error is repeated when printing.
- The paper is not cut even after the auto-cutter function is operated.

The causes of each operation error, the solution for each error, and the methods for returning from the return-waiting status to the print ready status are listed below.

- **Out-of-paper error**

No paper

→Load paper.

There are two methods to load the paper: automatic loading and manual loading.

See **Section 6.5.1, Loading Paper.**

The IF2202-02B goes to return-waiting status if auto-loading is used. If paper is set manually, the IF2202-02B goes to head-up error status.

- **Head-up error**  
The head is “up”.  
→Lower the head.  
The IF2202-02B goes to print ready status in about a second.
- **Vp error**  
Vp is out of the allowable range (10 V to 27 V).  
→The IF2202-02B goes to return-waiting status when Vp becomes within the allowable range.
- **Head temperature error**  
The temperature of the thermal head is - 10 °C or lower, or 80 °C or higher.  
→If it ranges from - 5 °C to 75 °C , the chip set goes to print-ready status.
- **Auto-cutter error**  
There are three kinds of auto-cutter errors:
  1. An error occurs during operation of the cutter; the movable blade is in the home position.  
→The IF2202-02B goes to the return-waiting status immediately after generating the auto-cutter error.
  2. An error occurs during operation of the cutter; the movable blade is not in the home position (the movable blade is locked).  
→The IF2202-02B occurs the auto-cutter error. The IF2202-02B goes to return-waiting status when the cause of the locking of the movable blade (such as paper jamming) is removed and the movable blade is returned to the home position (see **Section 5.4.3, Unlocking of Movable Blade**).
  3. The auto-cutter control command (ESC+ ‘m’, ESC+ ‘i’ ) is executed without connecting the auto-cutter.  
→The IF2202-02B goes to the return-waiting status immediately after generating the auto-cutter error.
- **Return-waiting status**  
The printer will be in this status immediately after auto-loading and after removing a Vp voltage error or an auto-cutter error. Data input is prohibited, as when in error status. The methods for returning to the print ready status are follows:
  - ① Turn the !FEED signal (CN6-20) to “High” from “Low”.
  - ② Down the head from “up” status.

#### NOTE

The  $V_p$  voltage allowable range is programmed to be from 10 V to 27 V in consideration of detection errors and instantaneous changes.



## CHAPTER 8

### SOFTWARE SPECIFICATIONS

#### 8.1 CHARACTER SET SELECTION

The character set which can be printed using the IF2202-02B is only an extended graphic character set. See **CHAPTER 9, CHARACTER CODES** for details.

#### 8.2 FUNCTION CODES AND CHARACTER CODES

The character code range and user-defined character codes when the Kanji specification/cancel mode is selected are different from those when the shift Kanji mode is selected with the Kanji Code System Select command.

##### 8.2.1 Kanji Specification/Cancel Mode

When the Kanji specification/cancel mode is selected with the Kanji Code System Select command, Kanji characters can be printed using 2-byte character codes after Kanji mode is selected with the Kanji Mode Specify command.

The command functions as 1-byte character code or the first or second byte of 2-byte character code. If a command is input in 2-byte character codes, the next data is always processed as the first byte. However, the next data is 1-byte character code for FS+ "." and ESC+"H". If a command is input as the second byte of 2-byte character code, the first byte is ignored.

(1) 1-byte character codes

- 00<sub>16</sub> - 1F<sub>16</sub>: The codes listed below are processed as commands.  
The other codes are ignored.  
0A<sub>16</sub> (LF), 0D<sub>16</sub>(CR), 0E<sub>16</sub>(SO), 12<sub>16</sub>(DC2), 13<sub>16</sub>(DC3), 14<sub>16</sub>(DC4),  
18<sub>16</sub>(CAN), 1B<sub>16</sub>(ESC), 1C<sub>16</sub>(FS)
  
- 20<sub>16</sub> - 7E<sub>16</sub>: Character code
- 7F<sub>16</sub>: Ignored
- 80<sub>16</sub> - FE<sub>16</sub>: Character code
- FF<sub>16</sub>: Ignored

(2) 2-byte character codes

User-defined characters are specified with 2-byte character codes.

- First byte

77<sub>16</sub>: User-defined character area

The other codes are ignored unless they are commands.

The next data to be received is processed as the first byte.

- Second byte

21<sub>16</sub> - 7E<sub>16</sub>: Processed as the second byte of a user-defined character.

The other codes are ignored unless they are commands.

The next data to be received is processed as the first byte.

### 8.2.2 Shift Kanji Mode

If the shift Kanji mode is selected with the Kanji Code System Select command, Kanji characters can be printed with 2-byte character codes without entering the Kanji Mode Specify command.

The command functions as 1-byte character code or the second byte of 2-byte character code.

If a command is input for the second byte of 2-byte character code, the first byte is ignored.

(1) 1-byte character codes

00<sub>16</sub> - 1F<sub>16</sub>: The codes listed below are processed as commands.

The other codes are ignored because they are 1 byte.

0A<sub>16</sub> (LF), 0D<sub>16</sub> (CR), 0E<sub>16</sub> (SO), 12<sub>16</sub> (DC2), 13<sub>16</sub> (DC3), 14<sub>16</sub> (DC4), 18<sub>16</sub> (CAN), 1B<sub>16</sub> (ESC), 1C<sub>16</sub> (FS)

20<sub>16</sub> - 7E<sub>16</sub>: Character code

7F<sub>16</sub>: Ignored

80<sub>16</sub> - FE<sub>16</sub>: See **(2)**.

FF<sub>16</sub>: Ignored

(2) 2-byte character codes

User-defined characters are specified with 2-byte character codes.

- First byte

81<sub>16</sub> - 9F<sub>16</sub>: The first and second bytes are ignored.

E0<sub>16</sub> - EB<sub>16</sub>: The first and second bytes are ignored.

EC<sub>16</sub>: User-defined character area

ED<sub>16</sub> - EF<sub>16</sub>: The first and second bytes are ignored.

The other codes 80<sub>16</sub> - FE<sub>16</sub> are processed as 1-byte character.

- Second byte

40<sub>16</sub> - 7E<sub>16</sub>: Processed as the second byte of a user-defined character.

80<sub>16</sub> - 9E<sub>16</sub>: Processed as the second byte of a user-defined character.

The first and second bytes of the other codes are ignored unless they are commands.

### 8.3 MEMORY

When using the optional font, the character string function, and the stamp function, it is necessary to know the remaining amount of memory beforehand.

#### (1) Memory area

The IF2202-02B sets memory area active or inactive in order to change the memory (RAM) uses and its capacity.

Setting the memory area active means that the memory is divided into the specified capacity (the number of bytes) so as to be exclusively used for a certain function.

Setting the memory area inactive means that the division of the memory for a certain function's exclusive use is abolished so that the other functions become usable.

Normally it is unnecessary for the user to be aware of this. However, when using the optional font, the character string function, and the stamp function, the user must not exceed the remaining memory area.

The 32K-byte memory (32,768 bytes RAM) is mounted on the IF2202-02B. After initialization, the memory area of the bytes listed in **Table 8-1** is set active.

Table 8-1 Memory Area after Initialization

Use	Bytes	Available memory area
System control information (LTP2242, LTP2342)	2,000	disable
System control information (LTP2442)	2,761	disable
Stamp directory	776	disable
Character string directory	776	disable
User-defined character	9,784	enable
Downloaded character	14,568	enable
Optional font	0	enable
Character string function	0	enable
Stamp function	0	enable

The system control information, the character string directory, and the stamp directory are the areas used internally. Therefore, they can not be set inactive.

For user-defined characters and downloaded characters, the memory area is set active beforehand at initialization.

This area can be set inactive using commands, and its empty area increased for the optional font, the character string function, and the stamp function.

When defining and registering the optional font, the character string, and the stamp function, it is necessary to calculate the remaining memory area constantly. Be aware that all of the data which exceeds the remaining memory area is ignored.

For the optional font definition, the stamp registering, and the character string registering, the area is automatically set active when the respective command is input. These commands operate as follows:

- **When the parameter of the command is outside the range:**  
The IF2202-02B ignores the part outside and processes the remaining data as character codes.
- **When the command is normal but memory area is not set active (Memory shortage):**  
All of the data, including the successive data, is ignored.
- **When the command is normal and memory area is set active:**  
The data is registered.  
If the serial signal can be input, the remaining memory is confirmed using the remaining memory response command (See **Section 8.4.11, Response Function, DC2+'r'**).

## (2) Memory control information

After setting the area active, the memory control information is added at the front of the area set active. The number of bytes for memory control information differs depending on each function. The number of bytes for memory control information of each function is listed in **Table 8-2**.

Table 8-2 Number of Bytes of Memory Control Information

Function	Number of Bytes
User-defined character	8
Downloaded character	8
Option font	12
Character string (1 pattern)	10
Stamp (1 pattern)	11



**[Example]**

For a user-defined character, the 24 × 24 dot character takes 72 bytes and the 16 × 16 character takes 32 bytes. Therefore, there can be up to 94 characters, with 8 bytes of memory control information, as shown in the following:

$$(72+32) \times 94 +8=9784 \text{ bytes}$$

For the character strings and the stamps, allocate area for each pattern. If a character string of 30 bytes is registered, 10 bytes of memory control information are added. The total capacity is as follows:

$$30+10=40 \text{ bytes}$$

When calculating the remaining memory, include the number of bytes for memory control information. Since one memory area is limited to 65,535 bytes, the maximum number of bytes of the character string, the stamp, and the option font which can be registered, is as follows:

Character string	=65535 - 10	=65525 bytes
Stamp	=65535 - 11	=65524 bytes
Option font	=65535 - 12	=65523 bytes

**(3) Precautions for the character string and the stamp**

The character string function can not include the commands for allocating or freeing (up) the memory area listed in **Table 8-3**. When selecting the stamp, the following commands are not usable.

Table 8-3 Commands for Allocating or Freeing Memory Area

DC2+'P'	Option font define
DC2+'Q'	Option font clear
DC2+'T'	Stamp define
DC2+'U'	Stamp clear
DC2+'M'	Character string define
DC2+'N'	Character string clear
DC2+'D'	Downloaded character area operation
DC2+'G'	User-defined character area operation
DC2+'R'	External RAM initialization

If the commands listed in **Table 8-3** are input during selection of the stamp, the stamp selection is canceled. When the commands listed in **Table 8-3** are in the selected character string, the command is ignored and the character string selection is canceled. For the commands listed in **Table 8-3** contained in the character string, DC2 and the next byte are ignored.

#### **(4) Initialization of memory**

The IF2202-02B does not erase the content of the memory even if a reset signal or a reset command is input. A password written in the system control information in the memory is checked at initialization immediately after canceling the reset.

If the password coincides, the IF2202-02B determines that the data in the memory is valid. Downloaded characters, user-defined characters, option fonts, character string function, and stamp function are then available.

However, reset cancels all the character and function selections. It is necessary to specify them again using the selection commands.

If the password does not coincide, the IF2202-02B determines that the data in the memory is broken, and initializes memory area.

The IF2202-02B erases the content of the memory and initializes the memory area also when the external RAM initialization command is executed.

## 8.4 FUNCTION CODE DESCRIPTION

This chapter describes the commands in each function.

① ESC + 'J' + n		Print and Feed Forward
② Code	$1B_{16} + 4A_{16} + n$ , or $27_{10} + 74_{10} + n$	
③ Function	Feeds the paper forward. n can take on a value from 0 to $FF_{16}$ (0 to 255 dot line).	
④ Operation	If the line buffer contains print data, the IF2202-02B prints that data, feeds the paper by the specified line spacing, then feeds n dot lines forward. If the line buffer contains no data, the printer feeds the paper n dot lines forward.	

- ① Indicates the name of the command.
- ② Indicates both hexadecimal and decimal codes.
- ③ Indicates the functions of the commands.
- ④ Explains detailed printer operation when inputting commands.  
The program sample is coded in GW-BASIC (MS-DOS version).

### 8.4.1 Line/Paper Feed

#### LF Line Feed

Code	$0A_{16}$ or $10_{10}$
Function	Provides the same function as <b>CR</b> . Any <b>LF</b> code received immediately following a <b>CR</b> is ignored.

#### CR Carriage Return

Code	$0D_{16}$ or $13_{10}$
Function	Prints the contents of the line buffer in the currently selected print mode, and then feeds paper a predefined distance.

**Operation** Line spacing is initialized to 16 dots when power is turned on or when reset is performed.

The IF2202-02B performs only the line feed operation when the line buffer is empty.

The amount of paper feed for one line is the amount of line spacing plus the character height which is set at that time. Depending on font size (DC2+'F'+n) and double-height (ESC+'w'+n), the character height changes as follows:

	<u>16-dot size</u>	<u>24-dot size (Default)</u>
Normal character	16	24
Double height	32	48

Any **LF** code received immediately following a **CR** is ignored.

#### ESC+' J '+n

#### Print and Feed Forward

**Code**  $1B_{16} + 4A_{16} + n$ , or  $27_{10} + 74_{10} + n$

**Function** Feeds the paper forward.

n can take on a value from 0 to  $FF_{16}$  (0 to 255 dot line).

**Operation** If the line buffer contains print data, the IF2202-02B prints that data, feeds the paper by the specified line spacing then feeds n-dot lines forward.

If the line buffer contains no data, the IF2202-02B feeds the paper n dot lines forward.

#### ESC+' j '+n

#### Print and Feed Backward

**Code**  $1B_{16} + 6A_{16} + n$ , or  $27_{10} + 106_{10} + n$

**Function** Feeds the paper backwards.

n can take on a value from 0 to  $FF_{16}$  (0 to 255 dot line).

**Operation** If the line buffer contains print data, the IF2202-02B prints that data, feeds the paper by the specified line spacing, then feeds n-dot lines backwards.

If the line buffer contains no data, the IF2202-02B feeds the paper n-dot lines backward. At this time, ruler line and stamp are not printed out.

## 8.4.2 Format Setting

### ESC+' 2 '

### 16-dot Line Spacing

Code	$1B_{16} + 32_{16}$ , or $27_{10} + 50_{10}$
Function	Sets line spacing to 16 dots. This is the default setting.

### ESC+' 0 '

### 4-dot Line Spacing

Code	$1B_{16} + 30_{16}$ , or $27_{10} + 48_{10}$
Function	Sets line spacing to 4 dots.

### ESC+' A '+n

### ESC+' 3 '+n

### n-dot Line Spacing

Code	$1B_{16} + 41_{16} + n$ , or $27_{10} + 65_{10} + n$ $1B_{16} + 33_{16} + n$ , or $27_{10} + 51_{10} + n$
Function	Sets line spacing to n dots. n can take on a value from 0 to $FF_{16}$ (0 to 255 dot line).

### ESC+SP+n

### Character Spacing

Code	$1B_{16} + 20_{16} + n$ , or $27_{10} + 32_{10} + n$
Function	Sets the character spacing to n dots. The least significant 7 bits are valid for n. (0 to 127)
Operation	The specified amount of space is appended to the right side of each character. The character spacing on the left side is automatically set to zero.  n is initialized to 4 dots by default.

**ESC+' s '+nl+nr****Left/Right Character Spacing**

**Code**  $1B_{16} + 73_{16} + nl + nr$ , or  $27_{10} + 115_{10} + nl + nr$

**Function** Sets the character spacing on the left and right sides of each character to nl and nr dots, respectively.

The least significant 7 bits of nl and nr are valid.

nl : Left Character Spacing (0 to 127)  
nr : Right Character Spacing (0 to 127)

**Operation** nl specifies the character spacing on the left and nr specifies the character spacing on the right.

nl is initialized to 0 and nr to 4 dots by default.

**ESC+' U '+n****Inversion (Flip)**

**Code**  $1B_{16} + 55_{16} + n$ , or  $27_{10} + 85_{10} + n$

**Function** Turns on / off inverse print by line.

Only the least significant bit is valid for n.

n=0 : inverse print off (default)  
n=1 : inverse print on (upside down)

Inverse print is specified or canceled line by line.  
The last mode selected in each line becomes effective.

**Operation** In the inverse print mode, each character is printed upside down and the sequence of characters is printed backwards in a line as well.

The character sent first is printed at the right margin and subsequent characters are printed from right to left.

The inverse print mode is applied to character and underline, but not to bit image, ruler, or stamp.

Code	$12_{16} + 59_{16} + n$ , or $18_{10} + 89_{10} + n$
------	---

Function	Rotates characters.
----------	---------------------

Only the lower 2 bits are valid in n.

n = 0: Non rotation (default)  
n = 1: Right 90° rotation  
n = 2: Left 90° rotation  
n = 3: Ignored.

Operation	When you set a character rotation through this command, characters hereinafter are rotated.
-----------	---

Non-rotated and rotated characters are able to exist in a same line.

When other character modification is set, the modified characters are rotated. Underlines are not rotated.

The following characters can be rotated through this command:

- Expanded graphic characters
- User-defined characters
- Downloaded characters

The option fonts are not rotated.

4 dots of the right edge of the expanded graphic characters fonts composed with 24 x 16 dots can not be printed.

Form the 24 dot downloaded characters under 12 dots because only 24 x 12 dots is printed.

```

100 OPEN  "LPT1:BIN" FOR OUTPUT AS #1
110 PRINT #1, "CR+LF"
120 PRINT #1, "Line Feed";CHR$(&HA);
130 PRINT #1, "Carriage Return";CHR$(&HD);
140 PRINT #1,CHR$(&H1B); "J";CHR$(20);
150 PRINT #1, "ESC+J"
160 PRINT #1,CHR$(&H1B); "j";CHR$(40);
170 PRINT #1, "==== ESC + j"
180 FOR I=0 TO 32 STEP 8
190   PRINT #1,CHR$(&H1B); "3";CHR$(I);
200   PRINT #1, "ESC+3+n - ";I;"dot-line space"
210 NEXT I
220 PRINT #1,CHR$(&H1B); "0";
230 PRINT #1, "ESC+0 --- 4 dot-line space"
240 PRINT #1,CHR$(&H1B); "2";
250 PRINT #1, "ESC+2 --- 16 dot-line space"
260 PRINT #1, "-----"
270 PRINT #1, "Character space"
280 FOR I=0 TO 24
290   PRINT #1,CHR$(&H1B); " ";CHR$(I);
300   PRINT #1,CHR$(&H41+I);
310 NEXT I
320 PRINT #1, "  "
330 FOR I=0 TO 24
340   PRINT #1,CHR$(&H1B); "s" ;CHR$(I);CHR$(I);
350   PRINT #1,CHR$(&H41+I);
360 NEXT I
370 PRINT #1,CHR$(&H1B); " " ;CHR$(4)
380 PRINT #1,CHR$(&H1B); "U1" ;" Inverted Print"
390 PRINT #1,CHR$(&H1B); "U0" ;" Normal Print"
400 PRINT #1,CHR$(&H12); "Y1" ;" Rotation Right"
410 PRINT #1,CHR$(&H12); "Y2" ;" Rotation Left"
420 PRINT #1,CHR$(&H12); "Y0" ;" Normal Print"
430 CLOSE #1
440 END

```

Figure 8-1 Program Sample 1



```

CR+LF
Line Feed
Carriage Return

ESC+3+j ESC + j
ESC+3+n ----- 0 dot-line space
ESC+3+n ----- 8 dot-line space
ESC+3+n ----- 16 dot-line space
ESC+3+n ----- 24 dot-line space
ESC+3+n ----- 32 dot-line space

ESC+0 ----- 4 dot-line space
ESC+2 ----- 16 dot-line space
-----
Character space
ABCDEFGHIJKLMN O P Q R S T U V W X Y Z
ABCDEFGHIJ K L M N O P Q R S T U V W X
Y Z
Inverted Print

Normal Print
Normal Print

```

Figure 8-2 Print Sample 1

### 8.4.3 Character Modification

ESC+'-' +n		Underline
Code	1B <sub>16</sub> + 2D <sub>16</sub> + n, or 27 <sub>10</sub> + 45 <sub>10</sub> + n	
Function	Turns on/off underline mode.	
	Only the least significant three bits are valid for n.	
	n = 0: Turn off underline mode(default)	
	n =1 to 7: Set the underline thickness to n dots	
Operation	If a line contains a character with underline turned on, line spacing is increased by the number of the underline dots.	
	Underlining can be turned on and off many times in a line. The thickness of the underline, however, must be constant within the line. Consequently, the underline thickness that was specified last takes effect.	

**SO****Double-Width (with automatic reset ) ON**

Code	$0E_{16}$ or $14_{10}$
------	------------------------

Function	Turns on double-width mode.
----------	-----------------------------

Operation	Prints double-width characters until a <b>DC4</b> , <b>LF</b> , <b>CR</b> , <b>CAN</b> , or an <b>ESC+' W '+0</b> sequence is received or one line is fully printed.
-----------	--

Character spacing is also doubled (up to 254 dots).

A line may contain normal size, double-width, and/or double-height characters.

The IF2202-02B enters double-width/double-height mode if double-height mode is also active.

**DC4****Double-Width (with automatic reset) OFF**

Code	$14_{16}$ or $20_{10}$
------	------------------------

Function	Resets double-width mode with automatic reset.
----------	--

The IF2202-02B is placed in this mode when initialized.

**ESC+' W '+n****Double-Width**

Code	$1B_{16} + 57_{16} + n$ , or $27_{10} + 87_{10} + n$
------	--

Function	Turns on/off double-width mode. Only the least significant bit is valid for n.
----------	---

n = 0: Turns double-width mode off. (default)  
n = 1: Turns double-width mode on.

Operation	Prints double width characters. Character spacing is also doubled (up to 254 dots).
-----------	--

A line may contain normal size, double-width, and/or double-height characters.

The IF2202-02B enters the double-width/double-height mode if the double-height mode is also selected.

**ESC+ ' w ' +n****Double-Height**

**Code**  $1B_{16} + 77_{16} + n$ , or  $27_{10} + 119_{10} + n$

**Function** Turns on/off double-height mode.

Only the least significant bit is valid for n.

n = 0: Turns double-height mode off. (default)  
n = 1: Turns double-height mode on.

**Operation** Prints double-height characters.  
A line may contain normal size, double width, and/or double height characters.  
The IF2202-02B enters the double-width/double-height mode if the double-width mode is also selected.

**ESC+ ' l ' +n****Reverse**

**Code**  $1B_{16} + 49_{16} + n$ , or  $27_{10} + 73_{10} + n$

**Function** Turns on/off reverse mode.

Only the least significant bit is valid for n.

n = 0: resets the reverse mode (prints black characters on a white background).  
(default)  
n = 1: prints white characters on a black background.

**Operation** The space between characters is printed as black.  
The space between lines remains unprinted, and underline is printed as black even in the reverse mode.

**ESC+ ' E'****Bold Print ON****Code**

1B<sub>16</sub> + 45<sub>16</sub>, or 27<sub>10</sub> + 69<sub>10</sub>

**Function**

Turns on the bold print mode.

The bold print mode bolds the width direction.

**Operation**

When inputting this command, the following data are printed in bold type.

When turning on the reverse mode, this is ignored.

This can be used in mix with the other character modifications.

The right side edge of the printable area (432th dot for LTP2242, 57th dot for LTP2342, 832th dot for LTP2442) is not be bolded.

**ESC+ ' F'****Bold Print OFF****Code**

1B<sub>16</sub> + 46<sub>16</sub>, or 27<sub>10</sub> + 70<sub>10</sub>

**Function**

Turns off the bold print mode.

**Operation**

When inputting this command, the bold print mode is turned to off.

```

100 OPEN "LPT1:BIN" FOR OUTPUT AS #1
110 FOR I=7 TO 0 STEP -1
120   PRINT #1,CHR$(&H1B); "-" ;CHR$(I);
130   PRINT #1, "Under line ";I; "dot"
140 NEXT I
150 PRINT #1,CHR$(&HE); "DoubleWidth"
160 PRINT #1," Normal Print"
170 PRINT #1," Normal" ;
180 PRINT #1,CHR$(&H1B); "W1" ;"+DoubleWidth";
190 PRINT #1,CHR$(&H1B); "w1" ;"+DoubleHeight";
200 PRINT #1,CHR$(&H1B); "E" ;"+BoldPrint";
210 PRINT #1,CHR$(&H1B); "I1" ;"+Reverse";
220 PRINT #1,CHR$(&H1B); "W0" ;"-DoubleWidth";
230 PRINT #1,CHR$(&H1B); "w0" ;"-DoubleHeight";
240 PRINT #1,CHR$(&H1B); "I0" ;"-Reverse";
250 PRINT #1,CHR$(&H1B); "F" ;"-BoldPrint";
260 CLOSE #1
270 END

```

Figure 8-3 Program Sample 2

```

Under line 7 dot
Under line 6 dot
Under line 5 dot
Under line 4 dot
Under line 3 dot
Under line 2 dot
Under line 1 dot
Under line 0 dot
DoubleWidth by 50
Normal Print

Normal+DoubleWidth+DoubleHeight+BoldPrint+Reverse-DoubleWidth
DoubleHeight-Reverse-BoldPrint

```

Figure 8-4 Print Sample 2

#### 8.4.4 Character Set

**DC2+' F '+n**

Font Size Selection

Code

$12_{16} + 46_{16} + n$ , or  $18_{10} + 70_{10} + n$

Function

Selects one of 16-dot size and 24-dot size in character font.  
Only the least significant bit is valid for n.

n=0 : 16-dot size  
n=1 : 24-dot size (default)

Operation

The sizes of each character font are as follows:

	<u>16-dot size</u>	<u>24-dot size</u>
Extended graphics character set	16×8	24×12 <sup>*1</sup>
Downloaded character	16×8	24×12 <sup>*1</sup>
User-defined character	16×16	24×24

The font size of downloaded characters and user-defined characters becomes the font size specified by this command.

Be careful while defining downloaded characters and user-defined characters because the size of data varies depending on the font size.

\*1: The downloaded character and some graphics characters in the graphics character set consist of 24 × 16 dot size. However, the character font size may limit the actual printing operation.

Therefore, "24 × 12" dot size is recommended for full printing of any characters.  
For more information, see **Section 8.4.5 (1), Downloaded Character**.

**ESC+' t '+n**

Character Set Select

Code

$1B_{16}+74_{16}+n$ , or  $27_{10}+116_{10}+n$

Function

The printer ignores this command.

**FS+'C'+n****Kanji Code System Select**

Code	$1C_{16}+43_{16} +n$ or $28_{10}+67_{10}+n$
Function	Selects the Kanji specification/cancel mode or shift Kanji mode for the Kanji code system. Only the least significant bit of n is valid. n=0: Kanji specification/cancel mode (default) n=1: Shift Kanji mode
Operation	When n=0, the Kanji specification/cancel mode becomes effective.

To print the Kanji size characters, enter the Kanji mode using the Kanji Mode Specify command and then specify the Kanji size characters with the 2-byte character codes.

To print standard size characters, after canceling the Kanji mode using the Kanji Mode Clear command, specify the standard size characters with the 1-byte character codes. In this mode, the 1st byte of the user-defined characters is  $77_{16}$ .

When n=1, the shift Kanji mode becomes effective.

In the shift Kanji mode, by combining the character code of the specific range and the character code which is input next, the Kanji size character is specified with the 2-byte character code. In this mode, the 1st byte of the user-defined characters is  $EC_{16}$ . Character spaces of Kanji size characters are double the set value.

Also, in the shift Kanji mode, the Kanji Mode Specify/Clear command is ignored. For the details of the code, see 8.2 FUNCTION CODES AND CHARACTER CODES. In the IF2202-02B, if the user-defined character memory area is freed up, this command is ignored.

**ESC+' K '****FS+' & '****Kanji Mode Specify**

Code	$1B_{16}+4B_{16}$ , or $27_{10}+75_{10}$ $1C_{16}+26_{16}$ or $28_{10}+38_{10}$
Function	Specifies the Kanji mode. User-defined characters can be printed out by entering 2 bytes of the character codes $7721_{16}$ to $777E_{16}$ .

ESC+ ' H '

FS+ ' . '

Kanji Mode Clear

Code	1B <sub>16</sub> +48 <sub>16</sub> , or 27 <sub>10</sub> +72 <sub>10</sub> 1C <sub>16</sub> +2E <sub>16</sub> , or 28 <sub>10</sub> +46 <sub>10</sub>
Function	Clears the Kanji mode.

```
100 OPEN  "LPT1:BIN " FOR OUTPUT AS #1
110  PRINT #1,CHR$(&H12); "F";  CHR$(0)
120  PRINT #1, "16 DOT FONT"
130  PRINT #1, "#$%&01234567ABCDEFGHabcdefgh"
140  PRINT #1,CHR$(&H12); "F";CHR$(1);
150  PRINT #1, "24 DOT FONT"
160  PRINT #1, "#$%&01234567ABCDEFGHabcdefgh"
165 CLOSE #1
170 END
```

Figure 8-5 Program Sample 3

```
16 DOT FONT
#$%&01234567ABCDEFGHabcdefgh
24 DOT FONT
#$%&01234567ABCDEFGHabcdefgh
```

Figure 8-6 Print Sample 3



## 8.4.5 Character Definition

### (1) Downloaded Character

Downloaded character is a user defined character which can be allocated to character codes between  $20_{16}$  and  $FE_{16}$  (excluding  $7F_{16}$ ).

The size of the downloaded character is classified as follows, according to the size specified by the Font Size Select function code (DC2+'F'+n).

24-dot font size:  $24 \times 16$  dots  
16-dot font size:  $16 \times 8$  dots

Although the 24-dot font size is defined as  $24 \times 16$  dots, there are some restrictions when printed out.

If the character space on the right is specified as  $n=4$  dots or more, the complete character is printed out in  $24 \times 16$  dot size. On the other hand, if the character space is specified as  $n=3$  dots or less, 12 dots plus  $n$  dots (character space defined) are printed. Therefore, the complete character is not printed.

To print the character completely, the font size needs to be set at less than 12 dots, regardless of character space on the right.

When the font size is switched, both the 24-dot font size and 16-dot font size need to be defined.

Specify the downloaded character by using the downloaded character select function code and enter the defined character code to print out the downloaded character.

**Code**  $1B_{16} + 26_{16} + s + e + \text{Font Data}$ , or  
 $27_{10} + 38_{10} + s + e + \text{Font Data}$

**Function** Defines downloaded characters.

Specify the range of downloaded characters with s and e.

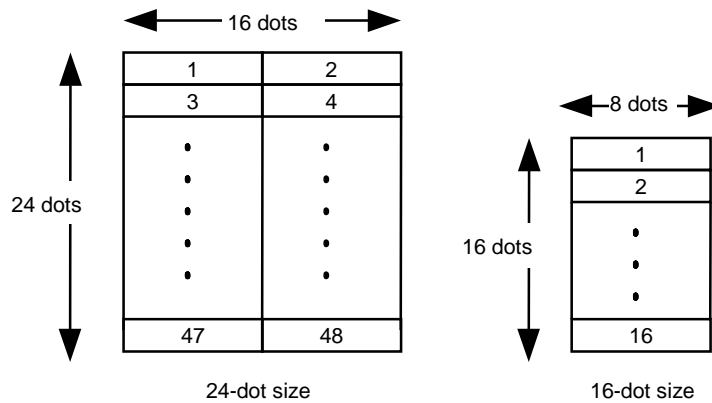
- s: Definition start character code
- e: Definition end character code
- $20_{16} \leq s \leq e \leq FE_{16}$ , where  $s \neq 7F_{16}$ ,  $e \neq 7F_{16}$

**Operation** If a range different from that of “s” or “e” above is specified, the invalid range is ignored and the rest of the data is processed as normal character codes. Font data comes after s and e.

A 24-dot size character is composed of 48 bytes and each dot line is represented by 2 bytes.

A 16-dot size character is composed of 16 bytes and each dot line is represented by 1 byte.

Enter the data as described below :



Enter the font data corresponded to the character codes specified between “s” and “e” sequentially. If the range between “s” and “e” contains  $7F_{16}$ , enter 48 byte of  $0_{16}$  at 24-dot size or 16 bytes of  $0_{16}$  at 16-dot size instead of  $7F_{16}$ . The character code of  $7F_{16}$  is ignored and cannot be printed out.

The correspondence between each bit of the font patterns can be changed by ESC + '=' + n (Image LSB/MSB select). LSB is set to the leftmost in the default setting.

If the downloaded character area is set inactive using DC2+'D'+0 (downloaded character area operation), ESC+'&' is ignored and the other data are processed as normal character data.

Downloaded character pattern is cleared by external RAM initialization command.

**ESC+ '%' + n****Downloaded Character Select**

Code	$1B_{16} + 25_{16} + n$ , or $27_{10} + 37_{10} + n$
------	--

Function	Select downloaded character set.
----------	----------------------------------

Only the least significant bit is valid for n.

n=0: Downloaded character deselect (default)

n=1: Downloaded character select

Operation	Once n=1 is selected, downloaded characters allocated to the character codes are printed out.
-----------	---

Option fonts or internal character set are printed out for undefined character codes.

**DC2+' D ' + n****Downloaded Character Area Operation**

Code	$12_{16} + 44_{16} + n$ , or $18_{10} + 68_{10} + n$
------	--

Function	Sets downloaded character area active or inactive. This function code allows line buffer to be cleared.
----------	---

Only the least significant bit (LSB) is valid for n.

n=0: Sets downloaded character area inactive.

n=1: Sets downloaded character area active (default).

Operation	Once n=0 is selected, downloaded characters are cleared and downloaded character set is deselected. The following function codes are ignored.
-----------	---

ESC+& Downloaded character define

ESC+% Downloaded character select

If downloaded character area is set inactive when character strings or stamps are being selected, the selected character strings and stamps are deselected automatically.

Downloaded character area occupies 14,568 bytes. When downloaded character area is set inactive, the remaining memory area increases by 14,568 bytes and can be used for character strings or stamps, that is, more than 14,568 bytes memory area is necessary to set downloaded character area active again. If the remaining memory area is not enough, downloaded character area cannot be set active and DC2+'D'+1 is ignored.

## (2) User-Defined Characters

User-defined characters are Kanji size characters that can be defined by the user. The range of codes are as follows according to the code system selected with the Kanji Code System Select command.

If the Kanji specification/cancel mode is specified :  $7721_{16} - 777E_{16}$

If the shift Kanji mode is specified :  $EC40_{16} - EC9E_{16}$  (except  $EC7F_{16}$ )

The size of definable characters is as follows according to the size specified with the font size select command (DC2+'F'+n).

If a 24-dot font is selected:  $24 \times 24$  dots

If a 16-dot font is selected:  $16 \times 16$  dots

To change the font size, a 24-dot character and a 16-dot character must be defined separately even if they have the same character code.

To print user-defined characters, a registered user-defined character code must be input in the same way as for the other Kanji and Kanji size characters.

The default is the space of a Kanji size character. If an unregistered user-defined character code is input, the space of a Kanji size character is printed.

### ESC+'+'+k1+k2+Font data

#### FS+'2'+k1+k2+Font data

#### User-Defined Character Define

Code	$1B_{16}+2B_{16}+ k1+k2+Font\ data\ or,$ $27_{10}+43_{10}+ k1+k2+Font\ data$ $1C_{16}+32_{16}+ k1+k2+Font\ data\ or,$ $28_{10}+50_{10}+ k1+k2+Font\ data$
------	--

Function	Define a user-defined character.
----------	----------------------------------

k1 and k2 specify the character code of the user-defined character.

k1: First byte

k2: Second byte

Operation	k1 and k2 specify 94 user-defined characters.
-----------	---

The ranges of k1 and k2 change according to the Kanji code system selected with the Kanji Code System Select command.

See **User-defined characters**.

If k1 or k2 is out of the range of the user-defined characters, the wrong data is ignored and the subsequent data is processed as normal character code.

After k1 and k2, user-defined character font data must be input.

The number of font data items per character is as follows:

24-dot font data =  $3 \times 24 = 72$  bytes

16-dot font data =  $2 \times 16 = 32$  bytes



### (3) Option Font

Option font is a user-defined font which you can edit its size, and pattern.

This is similar to downloaded character, except that you can define only  $16 \times 8$  dot or  $24 \times 16$  dot size for downloaded character.

You can define minimum  $1 \times 8$  dot to maximum  $127 \times 127$  dot character for option fonts.

While you can define downloaded characters in the separate area (ex.  $30_{16} - 39_{16}$  and  $41_{16} - 46_{16}$ ), you can define option fonts only in a consecutive area. If you redefine option fonts, the option fonts defined before are cleared and new option fonts are entered.

Option fonts can be printed out as 1 byte character code, in the same way as built-in fonts and downloaded characters.

The entered character codes are examined in the following order and the corresponding characters are printed out.

- 1) If downloaded character is selected and the character is defined in the downloaded character area, downloaded character is printed out.
- 2) In case other than 1), if option font is selected and the character is defined in the option font area, option font is printed out.
- 3) In case other than 1) and 2), built-in font is printed out.

Option font can be enlarged horizontally and vertically, and reversed in the same way as normal character.

However, option font cannot be rotated.

Option font is aligned to bottom and printed out in the same way as normal character.

#### DC2+ ' P ' + s + e + x + y + Font Data

#### Option Font Define

Code	$12_{16} + 50_{16} + s + e + x + y + \text{Font Data}$ , or $18_{10} + 80_{10} + s + e + x + y + \text{Font Data}$
------	---

Function	Defines option fonts. Specifies the range of option fonts with s and e.
----------	--

S: Definition start character code  
E: Definition end character code  
 $20_{16} \leq s \leq e \leq FE_{16}$ , where  $s \neq 7F_{16}$ ,  $e \neq 7F_{16}$

Specifies the number of dots to the horizontal direction and the vertical direction of option fonts with x and y.

x: Number of dots to the horizontal direction (8 to 127)  
y: Number of dots to the vertical direction (1 to 127)

**Operation**

This function code allows the line buffer to be cleared. If you specify the different range than above to s or e, the invalid range is ignored. The rest of the data is processed as normal character codes.

You can define option fonts only in the consecutive area specified with s and e. If you redefine option fonts, the option fonts defined before are cleared and new option fonts are entered.

You can specify the number of dots to the horizontal direction with x and to the vertical direction with y. In each case, the value between 8 and 127 for x, and 1 and 127 for y must be specified as the number of dots. If the number of dots is out of range, the rest of the data is processed as normal character code.

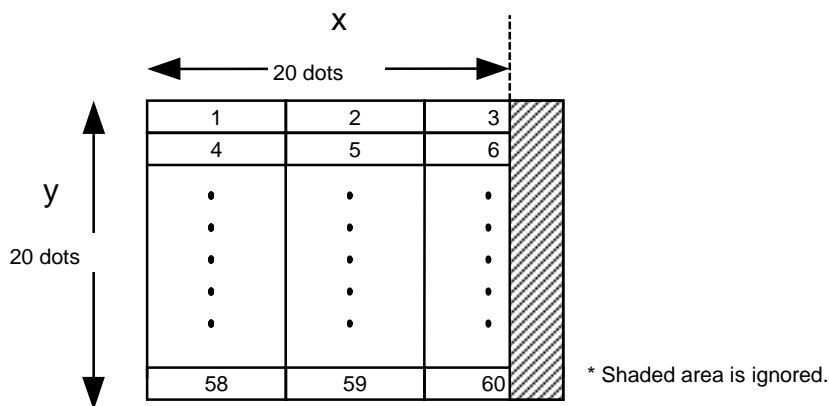
Create font data as follows:

You need  $\text{INT}((x+7)/8)$  bytes for data to the horizontal direction for a character.

Since there are y dot lines in the vertical direction, data for a character is expressed by the following equation.

$$\text{Data for a character} = \text{INT}((x+7)/8) \times y \text{ bytes}$$

Enter the font data as described below (x=20, y=20):



Enter the data for the character codes in order, from character code s through character code e.

The correspondence between each bit of the font pattern can be changed by ESC + '=' + n (Image LSB/MSB select). LSB is set to the leftmost in the default setting.

If x is not a multiple of 8, the rightmost bit is ignored. For example, when 20 dots are substituted into x, the rightmost 4 dots are ignored as shown above. If the range between s and e contains  $7F_{16}$ , enter one character of  $0_{16}$  in the  $7F_{16}$  area.  $7F_{16}$  is ignored and cannot be printed out.

Total font data is expressed by the following equation.

$$\text{Total font data} = \text{data for a character} \times (e-s+1) \text{ bytes}$$

Since 12 bytes memory control data is added to option font, total font data +12 bytes memory is used.

If the remaining memory is less than the total font data+12 bytes, all the data is ignored.

For more information on remaining memory, see **Section 8.3, Memory**.

The amount of all font data plus 12 has to be 65, 535 bytes or less.

If character strings or stamps are being selected, the selected character strings and stamps are suspended.

Option font pattern is cleared by external RAM initialization command.

DC2+' O ' +n

Option Font Select/Deselect

Code

$12_{16} + 4F_{16} + n$ , or  $18_{10} + 79_{10} + n$

Function

Selects option font.

Only the least significant bit is valid for n.

n=0: Deselects option font (default).

n=1: Selects option font.

Operation

If option fonts are defined and selected, the codes specified as option fonts in the following character codes are printed out in option fonts.

DC2+' Q '

Option Font Clear

Code

$12_{16} + 51_{16}$ , or  $18_{10} + 81_{10} + n$

Function

Clears option fonts and sets the memory area inactive for the option fonts.

Operation

The remaining memory increases by the memory area set inactive.

If character strings or stamps are being selected, the selected character strings and stamps are deselected automatically.

This function code allows line buffer to be cleared.



```

100 OPEN "LPT1:BIN" FOR OUTPUT AS #1
110 GJS=CHR$(&H77)+CHR$(&H21)
120 PRINT #1,CHR$(&H12);"F0"; '16 dots font
130 PRINT #1,CHR$(&H1B);"&CC"; 'C
140 RESTORE 380 : N=16 : GOSUB 350
150 PRINT #1,CHR$(&H1B);"+";GJ$; '7721
160 RESTORE 400 : N=32 : GOSUB 350
170 PRINT #1,CHR$(&H12);"F1"; '24 dots font
180 PRINT #1,CHR$(&H1B);"&DD"; 'D
190 RESTORE 400 : N=48 : GOSUB 350
200 PRINT #1,CHR$(&H1B);"+";GJ$; '7721
210 RESTORE 440 : N=72 : GOSUB 350
220 PRINT #1,CHR$(&H12);"PBB";CHR$(16);CHR$(8);'B
230 RESTORE 500 : N=16 : GOSUB 350
240 PRINT #1,CHR$(&H1B);"%1";
250 PRINT #1,CHR$(&H12);"O1";
260 FOR I=0 TO 7
270 PRINT #1,CHR$(&H1B);"W";CHR$(I);
280 PRINT #1,CHR$(&H1B);"w";CHR$(I);
290 PRINT #1,CHR$(&H1B);"I";CHR$(I\2);
300 PRINT #1,CHR$(&H12);"F";CHR$(I\4);
310 PRINT #1,"ABCDE";
320 PRINT #1,CHR$(&H1B);"K"GJ$;CHR$(&H1B);"H"
330 NEXT I
335 CLOSE #1
340 END
350 FOR I=0 TO N-1
360 READ D$: PRINT #1,CHR$(VAL("&h"+D$));
370 NEXT I : RETURN
380 '16x8 Font
390 DATA 18,18,3C,3C,7E,7E,DB,99,18,18,18,18,18,18,18,18
400 '24x12 Font
410 DATA 60,00,60,00,F0,00,90,00,98,01,08,01,0C,03,04,02
420 DATA 06,06,02,04,9F,0F,9F,0F,90,00,90,00,90,00,90,00
430 DATA 90,00,90,00,90,00,90,00,90,00,90,00,F0,00,F0,00
440 '24x24 Font
450 DATA 00,18,00,00,3C,00,00,7E,00,00,E7,00,80,C3,01,C0
460 DATA 81,03,E0,00,07,70,00,0C,B8,E7,1B,FC,24,3F,1E,24
470 DATA 78,03,24,C0,00,24,00,00,24,00,00,24,00,00,24,00
480 DATA 00,24,00,00,24,00,00,24,00,00,24,00,00,24,00,00
490 DATA 24,00,00,3C,00,00,3C,00
500 '8x16 Font
510 DATA 08,10,0C,30,0E,70,FF,FF,FF,FF,0E,70,0C,30,08,10

```

Figure 8-7 Program Sample 4

A ← ↑ DE 田 ↕  
 A ↔ ↑ DE 田 ↕  
~~A ← ↑ DE 田 ↕~~  
 A ← ↑ DE 田 ↕  
 A ← C ↑ E 田 ↕  
 A ↔ C ↑ E 田 ↕  
~~A ← C ↑ E 田 ↕~~  
**A ← C ↑ E 田 ↕**

Figure 8-8 Print Sample 4

### 8.4.6 Ruler

For the line thermal printer to print vertical lines continuously while printing characters. The IF2202-02B is provided with two ruler line buffers (called A and B) which have the capacity to accommodate one dot line of data and can print two types of ruler line images as selected by the user. When printing characters while the ruler is on, the dot corresponding to the bit which is 1 in the selected ruler buffer is printed out in black. This produces continuous vertical lines.

Ruler line can be printed out using the following procedure.

1. Select ruler line buffer A or B.

DC3+'A' Ruler Line Buffer A  
DC3+'B' Ruler Line Buffer B

2. Clear the selected ruler buffer and select the appropriate ruler data function.

DC3+'C' Ruler Line Buffer Clear  
DC3+'V' Ruler Line Image  
DC3+'D' Define Ruler Line by Dot  
DC3+'L' Define Ruler Line by Line  
DC3+'F' Define Ruler Line with Repeating Pattern  
(Select overlapping mode for ruler and character.)  
ESC+'#' Overlapping Mode Select

3. Turn on the ruler line.

DC3+'+' Ruler Line ON

4. When printing characters, the ruler line and characters are overlapped and printed.

The ruler line can be printed on the characters and spaces. However, when feeding paper backward using ESC+'j', feeding forward through !FEED and feeding backward through !BFEEED, no printing is performed.

If you select stamp when the ruler line is on, the stamp is printed prior to the others. After the stamp is printed, the ruler line is turned on again.

#### DC3+' A '

#### Ruler Line Buffer A

Code	13 <sub>16</sub> + 41 <sub>16</sub> , or 19 <sub>10</sub> + 65 <sub>10</sub>
Function	Selects ruler line buffer A (default) .
Operation	Subsequently, the IF2202-02B writes ruler line data into ruler line buffer A and prints the image from ruler line buffer A when ruler line printing is turned on.

**DC3+' B '****Ruler Line Buffer B**

Code	$13_{16} + 42_{16}$ , or $19_{10} + 66_{10}$
Function	Selects ruler line buffer B.
Operation	Subsequently, the IF2202-02B writes ruler line data into ruler line buffer B and prints the image from ruler line buffer B when ruler line printing is turned on.

**DC3+' V ' + Image Data****Ruler Line Image**

Code	$13_{16} + 56_{16} + \text{Image Data}$ , or $19_{10} + 86_{10} + \text{Image Data}$
Function	Writes image data into the selected ruler line buffer. Image data is corresponding to the number of bytes of one dot line.
Operation	The image data consists of one dot line as with the bit image. Number of bytes in one dot line: LTP2242 : 54 LTP2342 : 72 LTP2442 : 104  The correspondence between each bit of the image data can be changed by ESC + '=' + n (Image LSB/MSB select). LSB is set to the leftmost in the default setting.

**DC3+' D ' +nl+nh****Define Ruler Line by Dot**

Code	$13_{16} + 44_{16} + nl + nh$ , or $19_{10} + 68_{10} + nl + nh$
Function	Sets to "1" the dot in position of the selected ruler line buffer.  nl represents the least significant byte ( $0 \leq nl \leq FF_{16}$ ) and nh the most significant byte ( $0 \leq nh \leq 3_{16}$ , i.e., the least significant two bits are valid).  Dot position= $nh \times 256 + nl$ $0 \leq nhnl \leq 3FF_{16}$
Operation	Dot position means the position with the leftmost dot as 0.

**DC3+ 'L' +ml+mh+nl+nh****Define Ruler Line by Line**

Code	13 <sub>16</sub> +4C <sub>16</sub> +ml+mh+nl+nh, or 19 <sub>10</sub> +76 <sub>10</sub> +ml+mh+nl+nh
Function	Sets to "1" a line of dots from the mhl dot to the nhnl dot of the selected ruler line buffer.  ml and nl represent the least significant byte (0 ≤ ml or nl ≤ FF <sub>16</sub> ) and mh and nh the most significant byte (0 ≤ mh or nh ≤ 3 <sub>16</sub> , i.e., the least significant two bits are valid).  Dot position = mh × 256 + ml, or nh × 256 + nl
Operation	Dot position means the position with the leftmost dot as 0.

**DC3+ 'F' +n1+n2****Define Ruler Line with Repeating Pattern**

Code	13 <sub>16</sub> + 46 <sub>16</sub> + n1 + n2, or 19 <sub>10</sub> + 70 <sub>10</sub> + n1 + n2
Function	Fills the selected ruler line buffer with 2 bytes of data represented by n1n2. n1 and n2 are data 0 to FF <sub>16</sub> .
Operation	Fills the ruler line buffer with the image of 16 dots represented by n1n2 (n1 is 8 dots from the left and n2 is 8 dots from the right, viewed from the paper feed direction). The correspondence between each bit of n1 and n2 can be changed by ESC + '=' + n (Image LSB/MSB select) . LSB is set to the leftmost in the default setting.

**DC3+ ' +'****Ruler Line ON**

Code	13 <sub>16</sub> + 2B <sub>16</sub> , or 19 <sub>10</sub> + 43 <sub>10</sub>
Function	Turns on ruler line printing.
Operation	The IF2202-02B prints ruler lines while printing characters, spacing lines, or feeding paper as specified by a <b>CR</b> or <b>LF</b> code or an <b>ESC+ 'J'</b> sequence. The dots which the bit of the selected ruler line buffer is 1 are printed out. The overlapping mode can be set either to OR or XOR by overlapping mode select (ESC + '#'+ n).

**DC3+ ' - '****Ruler Line OFF**

Code	$13_{16} + 2D_{16}$ , or $19_{10} + 45_{10}$
Function	Turns off ruler line printing.
Operation	The IF2202-02B prints no ruler lines.

**DC3+ ' P '****Print One Dot Line after Printing Line Buffer Data**

Code	$13_{16} + 50_{16}$ , or $19_{10} + 80_{10}$
Function	Prints one dot line from the selected ruler line buffer when ruler line printing is on.
Operation	The IF2202-02B prints one dot line of ruler line after printing any data left in the line buffer and feeding the paper one line space. (The ruler line is printed out on the characters or on the line spaces.) If ruler line printing is off, the IF2202-02B feeds paper one dot line.

**DC3+ ' C '****Ruler Line Buffer Clear**

Code	$13_{16} + 43_{16}$ , or $19_{10} + 67_{10}$
Function	Clears all bits in the selected ruler line buffer to zero.

**DC3+ ( ' )****Continuous Ruler Line Control Code Input**

Code	$13_{16} + 28_{16}$ , or $19_{10} + 40_{10}$
Function	After this code is received, the IF2202-02B accepts any ruler line control codes without a leading <b>DC3</b> code until a ")" code is received.
Operation	The IF2202-02B ignores any codes other than the <b>DC3</b> extended function codes. For example, a sequence of Ruler Line Buffer A command, Ruler Line ON command, Print One Dot Line command and Ruler Line OFF command can be shortened as follows. DC3 + '(A+P-)' ( $13_{16} + 28_{16} + 41_{16} + 2B_{16} + 50_{16} + 2D_{16} + 29_{16}$ )

```

100 OPEN "LPT1:BIN  " FOR OUTPUT AS #1
110 PRINT #1,CHR$(&H13); "A";
120 PRINT #1,CHR$(&H13); "V";
130 PRINT #1,STRING$(52,CHR$(&H55));
140 FOR I=1 TO 52 : PRINT #1,CHR$(I); : NEXT I
150 PRINT #1,CHR$(&H13); "+";
160 PRINT #1, "RULED LINE OR"
170 PRINT #1,CHR$(&H1B); "#1";
180 PRINT #1, "RULED LINE XOR"
190 PRINT #1,CHR$(&H13); "(" ;
200 FOR I=0 TO 200
210   X=INT(SQR(40000!-I^2)):X1=415-X:X2=415+X
220   PRINT #1, "C";
230   PRINT #1, "L";CHR$(X1 MOD 256);CHR$(X1\256);CHR$(160);CHR$(1);
240   PRINT #1, "D";CHR$(X2 MOD 256);CHR$(X2\256);
250   PRINT #1, "P";
250 NEXT I
270 PRINT #1, "ACF";CHR$(&HFF);CHR$(&HFF);
280 PRINT #1, "BCL";CHR$(0);CHR$(0);CHR$(3);CHR$(0);
290 PRINT #1, "L";CHR$(60);CHR$(3);CHR$(63);CHR$(3);
300 PRINT #1, "D";CHR$(94);CHR$(0);
310 PRINT #1, "D";CHR$(158);CHR$(0); "-"
320 PRINT #1,CHR$(&H13); "(+APPPPB)" ;
330 PRINT #1, " RULEDLINEPRINT"
340 PRINT #1,CHR$(&H13); "(APPB)" ;
350 PRINT #1,CHR$(&HE); "ABCDEFGH IJ"
360 PRINT #1,CHR$(&H13); "(APPPP-)" ;
370 CLOSE #1
380 END

```

Figure 8-9 Program Sample 5

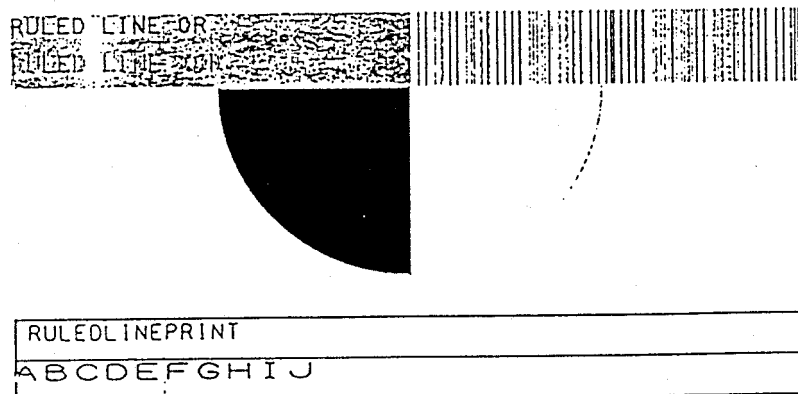


Figure 8-10 Print Sample 5

## 8.4.7 Bit image

ESC+ ' V ' +nl+nh+Image Data

Bit Image Specify

**Code**  $1B_{16} + 56_{16} + nl + nh + \text{Image Data}$ , or  
 $27_{10} + 86_{10} + nl + hn + \text{Image Data}$

**Function** Prints bit image.

nl, nh: number of dot lines in the vertical direction.

nl represents the least significant byte and nh represents the most significant byte.

Number of dot lines =  $nh \times 256 + nl$

$0 \leq nhnl \leq FFFF_{16}$  (65535)

Ignored when  $nl = nh = 0$ .

**Operation** Enter the image data following nl and nh.  
 The amount of image data is as follows:

Amount of image data =  $(nh \times 256 + nl) \times m$  bytes

m: number of bytes per dot line

The number of bytes per dot line varies with the printer mechanism as summarized below.

LTP2242: 54 bytes

LTP2342: 72 bytes

LTP2442: 104 bytes

The data following nl and nh is printed out entirely as image data.

Enter the data from the leftmost of the top dot line to the bottom dot line, sequentially.

The correspondence between each bit of the image data can be changed using

ESC+ '=' + n (Image LSB/MSB select). LSB is set to the leftmost in the default setting.

If you specify bit image without entering CR or LF after inputting characters for a range not exceeding one line, the characters of that line and the bit image are overlapped when printed out. At this time, all modifications to characters are valid.

Note that the characters are not normally printed out when overlapped unless a value more than the character height is specified by nhnl.

You can not overlap bit images with stamps or ruler lines. If you specify bit images, the stamp or the ruler line are suspended. After the bit image is complete, the print-out of stamps or ruler line restarts.



```

100 OPEN  "LPT1:BIN" FOR OUTPUT AS #1
110 PRINT #1, "Left= LSB"
120 PRINT #1,CHR$(&H1B);"V" ;CHR$(80);CHR$(0);
130 FOR I=1 TO 80
140   PRINT #1,STRING$(104,CHR$(I));
150 NEXT I
160 PRINT #1,CHR$(&H1B); "=1";
170 PRINT #1, "Left = MSB"
180 PRINT #1,CHR$(&H1B); "V";CHR$(80);CHR$(0);
190 FOR I=1 TO 80
200   PRINT #1,STRING$(104,CHR$(I));
210 NEXT I
220 PRINT #1, "OVER LAP PRINTING" ;
230 PRINT #1,CHR$(&H1B); "V" ;CHR$(40);CHR$(0);
240 FOR I=1 TO 20
250   PRINT #1,STRING$(40,CHR$(&H55));STRING$(64,CHR$(0));
260   PRINT #1,STRING$(40,CHR$(&HAA));STRING$(64,CHR$(0));
270 NEXT I
280 PRINT #1,CHR$(&H1B); "w1";
290 PRINT #1, "OVER LAP PRINTING";
300 PRINT #1,CHR$(&H1B); "V" ;CHR$(80);CHR$(0);
310 FOR I=1 TO 20
320   PRINT #1,STRING$(104,CHR$(&H11));STRING$(104,CHR$(0));
330   PRINT #1,STRING$(104,CHR$(&H44));STRING$(104,CHR$(0));
340 NEXT I
345 CLOSE #1
350 END

```

Figure 8-11 Program Sample 6

```

Left = LSB
.....
Left = MSB
.....
OVER LAP PRINTING
OVER LAP PRINTING

```

Figure 8-12 Print Sample 6

#### 8.4.8 Stamp

Stamp function allows you to overlap characters with predefined bit images and to print out them simultaneously. You can define bit image in a rectangular area of any size.

Since defined stamps are stored in the memory, you do not have to enter a large amount of data when you wish to print out the same bit image repeatedly.

Stamp is defined as following.

Create bit image in the rectangular area which is specified by a number of bytes in the horizontal direction and a number of dot lines in the vertical direction. You can specify up to 127 bytes (1016 dots) in the horizontal direction and up to 2047 dot lines in the vertical direction. You can register up to 256 stamps and print out the stamps with stamp select function code.

Stamp select allows you to specify the registered stamp number and printing position and print out the stamps overlapping with printed character or character spaces. Specify the printing position in the horizontal direction by 8 dots. Stamp print function code allows you to print out stamps without overlapping.

You cannot print out more than one stamp simultaneously.

You can set the stamp overlapping mode to OR or XOR by overlapping mode select (ESC+'#'+n).

If you select a stamp when ruler line printing is on, stamps are printed out and ruled lines are not printed out. When print out of stamps has completed, print out of ruler lines restarts. Note that you can print out stamps without turning on ruler line.

While the selected stamp is being printed, you should not enter the function codes such as setting memory area active or inactive (**Table 8-3**).

If such codes are included, printing out of the stamps is aborted. For more information, see **Section 8.3, Memory**.

## Code

$12_{16} + 54_{16} + n + x + yl + yh + \text{Stamp Data}$ , or  
 $18_{10} + 84_{10} + n + x + yl + yh + \text{Stamp Data}$

## Function

Defines stamp image.

n: Stamp number (0 to  $FF_{16}$ )  
 x: Number of bytes in the horizontal direction (1 to  $7F_{16}$ )  
 yl, yh: Number of dot lines in the vertical direction  
 Number of dot lines= $yh \times 256 + yl$   
 $1 \leq yhyl \leq 7FF_{16}$

## Operation

Specify the stamp numbers between  $0_{16}$  and  $FF_{16}$  (0 to 255) to n. The stamp number is any number. Stamps are selected with this number.

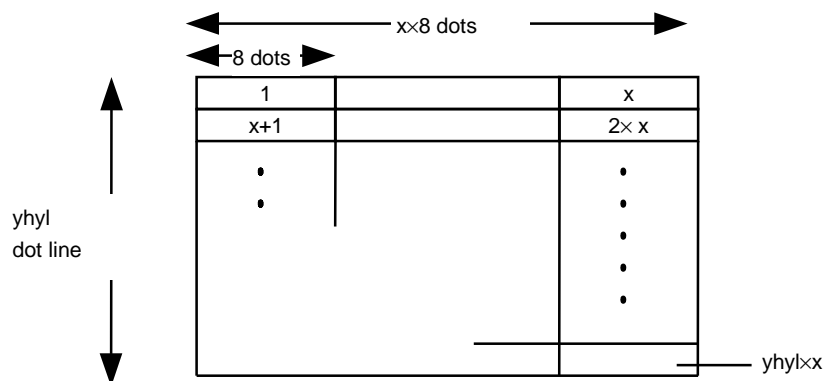
If you specify the defined stamp number, the predefined stamp is cleared and the memory area of the predefined stamp is freed, then the new stamp is defined.

x specifies the number of bytes of stamp image in the horizontal direction. The least significant 7bits of x are valid and the values between  $1_{16}$  and  $7F_{16}$  (1 to 127) are specified. Thus you can specify up to  $127 \times 8 = 1016$  dots by 8 dots to the horizontal direction.

yl and yh specify the number of dot lines of stamp image in the vertical direction. yl represents least significant byte and yh represents most significant byte. The least significant 3 bits of yh are valid. You can specify up to  $7FF_{16}$  (2047) dot lines with yl and yh.

If an invalid value beyond the range is specified for x, yl, or yh, the command data is ignored up to that specification and the following data is processed as character codes.

Enter the stamp data in the horizontal direction by 8 dots as follows:



The correspondence between each bit of the pattern can be changed by ESC + '=' + n (Image LSB/MSB select). LSB is set to the leftmost in the default setting.

Total stamp data is expressed by the following equation.

Total stamp data= $x \times (yh \times 256 + yl)$  bytes

Since 11 bytes of memory control data is added to stamp data, total stamp data +11 bytes memory is used.

If the remaining memory is less than total stamp data+11 bytes, all the data is ignored.

See **Section 8.3, MEMORY**, concerning remaining memory amount.

Total stamp data plus 11 must be 65,535 bytes or less.

If character string is selected, the selected character string is cleared automatically.

The stamp pattern defined using this command is cleared by the external RAM initialization command.

#### DC2+' S '+n +x

#### Stamp Select

##### Code

$12_{16} + 53_{16} + n + x$ , or  $18_{10} + 83_{10} + n + x$

##### Function

Selects stamp and specifies the printing position to the horizontal direction.

n: Stamp number (0 to FF<sub>16</sub>)

x: Printing position in the horizontal direction (by eight dots) (0 to FF<sub>16</sub>)

##### Operation

Specify the stamp numbers between 0<sub>16</sub> and FF<sub>16</sub> to n.

x specifies the printing position to the horizontal direction. You can specify the printing position by 8 dots. The stamp is printed at the x × 8th dot position from the left edge of the printable area.

If stamp is too large or parts of stamp area are out of the print area, those parts are not printed out.

If stamp has not been specified, the data until n is ignored and x is processed as a character.

More than one stamp cannot be printed out simultaneously.

The stamp selection is canceled automatically after printing the selected stamp completely.

Total character string data + 10 must be 65,535 bytes or less.

If you enter this function code before the printing of the selected stamp is completed, the stamp selection is canceled and the newly selected stamp is selected.

While the selected stamp is being printed, you should not enter the function codes listed in **Table 8-3**. If function codes are entered at this time, stamp printing is aborted.

#### DC2+' V '

#### Stamp Print Out

##### Code

$12_{16} + 56_{16}$ , or  $18_{10} + 86_{10}$

##### Function

Prints out all the selected stamps.

##### Operation

If stamp is not selected, this function code is ignored.

If stamp is being selected, the stamp is printed out to the bottom edge.

**DC2+' W '****Stamp Abort**

Code	$12_{16} + 57_{16}$ , or $18_{10} + 87_{10}$
Function	Aborts the print out of the selected stamp.
Operation	If stamp is not selected, this function code is ignored. If stamp is being selected, stamp printing is aborted.

**DC2+' U '+ n****Stamp Clear**

Code	$12_{16} + 55_{16}$ , or $18_{10} + 85_{10}$
Function	Clears the stamp specified with n (stamp number) and sets the memory area inactive.
Operation	The remaining memory area increases by the memory area freed. If stamp is selected, the selected stamp is cleared automatically.

```

100 OPEN  "LPT1:BIN " FOR OUTPUT AS #1
110 S$=STRING$(20,CHR$(&HFF))+STRING$(6,CHR$(0))+STRING$(20,CHR$(&H55))
120 PRINT #1,CHR$(&H12);"T";CHR$(10);CHR$(20);CHR$(160);CHR$(0);
130 FOR I=1 TO 20 : FOR J=0 TO 7
140   PRINT #1,MID$(S$,I+J,20);
150 NEXT J,I
160 PRINT #1,CHR$(&H12); "S";CHR$(10);CHR$(0);
170 GOSUB 290
180 PRINT #1,CHR$(&H12); "S";CHR$(10);CHR$(51);
190 GOSUB 290
200 PRINT #1,CHR$(&H12); "S";CHR$(10);CHR$(10);
210 PRINT #1, "          STAMP PRINT"
220 PRINT #1,CHR$(&H12); "W";
230 PRINT #1, "          STOP STAMP"
240 PRINT #1,CHR$(&H12); "U";CHR$(10);
250 PRINT #1, "STAMP DELETED"
260 PRINT #1,CHR$(&H12); "S";CHR$(10);CHR$(0);
270 PRINT #1,CHR$(&H12); "V"
275 CLOSE #1
280 END
290 ` WELCOME
300 PRINT #1, "          WELCOME to"
310 PRINT #1,CHR$(&H1B); "w1";CHR$(&HE);"    STAMP"
320 PRINT #1,CHR$(&H1B); "w0";CHR$(&HE);"    World"
330 PRINT #1,CHR$(&H1B); "W0";CHR$(&H12); "V";
340 RETURN

```

Figure 8-13 Program sample 7

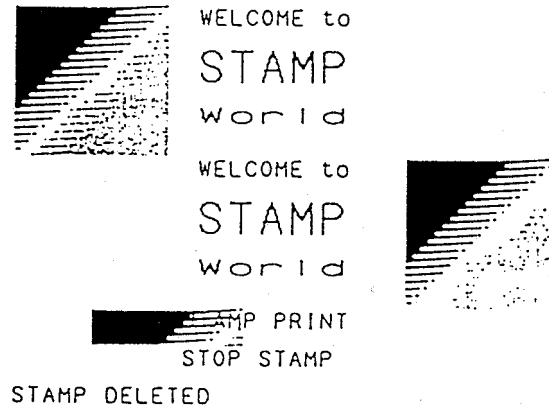


Figure 8-14 Print Sample 7

#### 8.4.9 Character String

The character string function code registers any character string, function code or other data into the memory and prints them with the character string select function code.

Since the registered character string is stored in the memory, you do not have to enter a large amount of data when you wish to print out the same character string or function code repeatedly.

Normally the data entered through a parallel or serial interface is stored in the input buffer and retrieved sequentially from the buffer to be analyzed or printed out.

If the character string is selected with this function code, the data is entered from the specified character string buffer instead of the input buffer. Data is fetched out from the input buffer again after the process of the character string is completed.

You should not enter the function codes such as setting memory area active or inactive (**Table 8-3**) into the character string.

If such codes are included, print out of the character string is aborted. For more information, see **Section 8.3, Memory**.

You can enter any function code excluding the codes listed in **Table 8-3**.

You can execute the character string select function codes by registering it in the character string. In this case, the process of the character string selected before is completed when this function code is processed.

Note that you should not register the select function code into the character string which selects the character string itself or selects each other. This results in an infinite loop that you cannot stop unless you reset the system or turn off the power.

**DC2+ ' M ' + n + dl + dh+ Character String Data**

**Character String Define**

Code	12 <sub>16</sub> + 4D <sub>16</sub> + n +dl +dh + Character String Data, or 18 <sub>10</sub> + 77 <sub>10</sub> + n +dl +dh + Character String Data
Function	Defines character string. n : Character sting number (0 to FF <sub>16</sub> ) dl, dh : Number of bytes for character string data Number of data bytes=dh×256+dl 0 ≤ dhdl ≤ FFFF <sub>16</sub>
Operation	Specify a character string number between 0 <sub>16</sub> and FF <sub>16</sub> to n. Character string number is any number and character string is selected with this number. If you specify a character string number for which a character string has already been defined, the previously defined character string is cleared, the memory area of the character string is freed, and the new character string is defined for the specified number. dl and dh specify a number of bytes of the character string data.  The function codes listed in <b>Table 8-3</b> must not be included in the character string. Since 10 bytes of memory control data is added to character string data, total character string data plus 10 bytes memory is used. If the remaining memory is less than total character string data+10 bytes, all data is ignored. For more information on remaining memory, see <b>Section 8.3, Memory</b> . Total character string data plus 10 must be 65,535 bytes or less. If stamp is being selected, the selected stamp is cleared automatically. The character string defined by using this command is cleared by the External RAM initialization command.

**DC2+ ' L ' +n**

**Character String Select and Execute**

Code	12 <sub>16</sub> + 4C <sub>16</sub> + n , or 18 <sub>10</sub> + 76 <sub>10</sub> + n
Function	Selects and executes the process of the character string. n : Character string number (0 to FF <sub>16</sub> )
Operation	Specifies a character string number between 0 <sub>16</sub> and FF <sub>16</sub> to n. If character string is not allocated to the specified character string number, this function code is ignored.  The data is analyzed and processed from the head of the character string sequentially. If the function codes listed in <b>Table 8-3</b> are included in the character string, the process of the character string is aborted when any of the function codes appears.



Code	$12_{16} + 4E_{16}+n$ , or $18_{10} + 78_{10}+n$
Function	Clears the character string specified with n (character string number) and sets the memory area inactive. n : Character string number (0 to FF <sub>16</sub> )
Operation	The remaining memory area increases by the memory area freed. If stamp is selected, it is deselected automatically.

```

100 OPEN "LPT1:BIN" FOR OUTPUT AS #1
110 L=8+4+104*32
120 PRINT #1,CHR$(&H12); "M";CHR$(0);CHR$(L MOD 256);CHR$(L\256);
130 PRINT #1," String 0";           `8 bytes
140 PRINT #1,CHR$(&H1B); "V";CHR$(32);CHR$(0);   `4 bytes
150 FOR I=1 TO 8                       `104*32 bytes
160   PRINT #1,STRING$(208,CHR$(&H33));
170   PRINT #1,STRING$(208,CHR$(&HCC));
180 NEXT I
190 L=22
200 PRINT #1,CHR$(&H12); "MA";CHR$(L MOD 256);CHR$(L\256);
210 PRINT #1,CHR$(&H1B); "W1";
220 PRINT #1,CHR$(&H1B); "I1";
230 PRINT #1, "STRING No.65"+CHR$(&HD);
240 PRINT #1,CHR$(&H1B); "I0";
250 `
260 PRINT #1,CHR$(&H12); "L";CHR$(0);
270 PRINT #1,CHR$(&H12); "LA";
280 PRINT #1, "This is ";CHR$(&H12); "L";CHR$(0);
290 PRINT #1,CHR$(&H1B); "W0";
300 PRINT #1,CHR$(&H1B); "w0";
310 PRINT #1,CHR$(&H12); "NA";
320 PRINT #1, "DELETED No.65 STRING"
330 PRINT #1,CHR$(&H12); "LA";
340 PRINT #1,CHR$(&H12); "L";CHR$(0);
350 CLOSE #1
360 END

```

Figure 8-15 Program Sample 8

```

String 0
STRING No. 65
This is String 0
DELETED No. 65 STRING
String 0

```

Figure 8-16 Print Sample 8

#### 8.4.10 Print Method

DC2+ ' > ' + n

Fixed Division Select

Code

$12_{16} + 3E_{16} + n$ , or  
 $18_{10} + 62_{10} + n$

Function

Sets the head driving method to fixed division mode.  
n is ignored.

Operation

When you input this command to the IF2202-02B, the fixed division drive mode is set.  
For more information on fixed division method, see **Section 5.2.2 (2) , Fixed division.**

If you print out horizontal lines or perform dense printing in fixed division mode, a very large electric current flows (approximately 11A at  $V_p=24V$ , printed at 720 dots).

When you are using a small capacity power supply, you may get sparse printing or a voltage range error.

The default status can be determined using DIP 2 (pin 6).

**DC2+ ' % ' + n****Dynamic Division, Number of Dots Specify**

Code	$12_{16} + 25_{16} + n$ , or $18_{10} + 37_{10} + n$
Function	Sets the head driving method to dynamic division mode and specifies the maximum number of activated dots. n: Specifies maximum number of activated dots to $n \times 8$ n<9: n=9 (9×8=72 dots) n>36 <sub>16</sub> (54): n=36 <sub>16</sub> (54×8=432 dots)
Operation	The range of n is between 9 and 54. The maximum number of activated dots is represented by $n \times 8$ . For more information on dynamic division method see <b>Section 5.2.2 (1), Dynamic division</b> . When you are using a small capacity power supply and specify a large number of activated dots, you may get sparse printing or voltage range error. Make sure that a larger electric current than the capacity of the power supply does not flow. The default status can be determined using DIP 2 (pin 6 through 8).

**DC2+ ' ~ ' + n****Print Density Specify**

Code	$12_{16} + 7E_{16} + n$ , or $18_{10} + 126_{10} + n$
Function	Sets print density. n: $65 \leq n \leq 135$
Operation	Although n can be set between 65 and 135, print density should be adjusted by setting the energy applied to the head to a value between 65% and 130% of the rated energy. If you specify less than 65 or more than 135 to n, this function code is ignored.  The default status can be selected 100% or 130% using DIP 1 (pin 6).

### 8.4.11 Response Function

The IF2202-02B can output the status of the interface or the printer through a serial port.

**Table 8-4** shows the output information and the range of the corresponding output codes.

Table 8-4 Range of Corresponding Output Codes

Information	Range of response Code
Hardware error code	02 <sub>16</sub> - 04 <sub>16</sub>
Xon, Xoff	11 <sub>16</sub> , 13 <sub>16</sub> (Only when Xon/Xoff control is selected)
Error status response	60 <sub>16</sub> - 7F <sub>16</sub>
Execute response request	50 <sub>16</sub> - 5F <sub>16</sub>
Vp voltage response	2E <sub>16</sub> , 30 <sub>16</sub> - 39 <sub>16</sub> ( , 0 - 9)
Remaining memory response	30 <sub>16</sub> - 39 <sub>16</sub> , 41 <sub>16</sub> - 46 <sub>16</sub> (0 - 9, A - F)
External RAM check sum response	30 <sub>16</sub> - 39 <sub>16</sub> , 41 <sub>16</sub> - 46 <sub>16</sub> (0 - 9, A - F)

For more information on hardware error codes, see **Section 7.1, Hardware error**.

Hardware error code, Xon/Xoff, error status response, and execute response request are represented by 1 byte codes. External RAM check sum response and V<sub>p</sub> voltage response are represented by 4 byte codes. Remaining memory response is represented by 6 byte codes.

Response codes are converted to serial signals and output through the TxD (CN6 : pin 23) terminal.

In serial input mode, these statuses (other than hardware error codes) are responded in the transmission conditions specified by function switches. For more information, see **Section 6.1, SETTING FUNCTION SWITCHES**.

In parallel input mode, the transmission conditions are fixed in the following and cannot be changed.

Baud Rate : 2400 bps  
 Parity : none  
 Data length : 8 bits  
 Stop bit : 1 bit

#### DC2+ ' r '

#### Remaining Memory Response

Code	12 <sub>16</sub> + 72 <sub>16</sub> , or 18 <sub>10</sub> + 114 <sub>10</sub>
Function	Outputs the remaining memory (RAM) capacity as a serial signal.
Operation	The unused memory area is represented by hexadecimal 6 bytes. In case a 32 K- byte external RAM is connected to the IF2202-02B and the remaining memory just after power-on is 4,864 bytes(1300 <sub>16</sub> ), the transmission data is 30 <sub>16</sub> , 30 <sub>16</sub> , 31 <sub>16</sub> , 33 <sub>16</sub> , 30 <sub>16</sub> and 30 <sub>16</sub> in the order starting from the 1st byte. Consequently, 6-byte data is sent.

Code	$12_{16} + 65_{16} + n$ , or $18_{10} + 101_{10} + n$
Function	<p>Selects whether the error status is returned or not when an error occurs.</p> <p>Only the least significant bit (LSB) is valid for n.</p> <p>n=0: No response (default)  n=1: Responds with error status</p>
Operation	<p>If you specify n as 1, when an error occurs or the status of the error is changed, the error status at this time is output as 1 byte code and subsequent 1 byte error codes which indicate the cause are output as serial signals through the TxD (CN6-23) terminal. Each bit of a error status represents the following errors.</p> <p>bit 7: 0  bit 6: 1  bit 5: 1  bit 4: Auto-cutter error  bit 3: <math>V_p</math> voltage error  bit 2: Head temperature error  bit 1: Head-up  bit 0: Out-of-paper</p> <p>Since upper 3 bits are fixed, the error status is represented by the value between <math>60_{16}</math> and <math>7F_{16}</math>.  An error status is output whenever the status of the error is changed. Therefore, <math>60_{16}</math> is output when the cause of the error disappears.</p> <p>Even if you specify the out-of-paper error is invalid with the out-of-paper select function command (DC2+'p'+0), the out-of-paper error status is output when out-of-paper is detected.</p>

**DC2+ ' q ' + n****Execute Response Request**

Code	12 <sub>16</sub> + 71 <sub>16</sub> + n, or 18 <sub>10</sub> + 113 <sub>10</sub> + n
Function	Returns the specified code when the command is processed. 'n' specifies the code to be returned. Only the last four bits are valid.
Operation	Because the interface is equipped with a 252 byte input buffer, command and data input and execution and printing are not performed simultaneously. Therefore, it is impossible to externally confirm which commands have been executed and which data has been printed. By inserting this command at the end of a command and/or print sequence, however, it is possible to confirm which commands have been executed because when the response to this command is returned it means all preceding commands and printing have been executed.

Specify the response code with 'n'. Only the last four bits of 'n' are valid. The code transmitted is the logical sum of the last 4 bits of 'n', which you specified, and 50<sub>16</sub>, which is the code from 50<sub>16</sub> to 5F<sub>16</sub>.

**DC2+ ' v '****V<sub>p</sub> Voltage Response**

Code	12 <sub>16</sub> + 76 <sub>16</sub> , or 18 <sub>10</sub> + 118 <sub>10</sub>
Function	Measures the V <sub>p</sub> voltage (power voltage for head and motor).
Operation	The accuracy of the measured voltage is within ±0.5V. The voltage is represented by hexadecimal 4-byte characters consisting of a 2-digit integer, decimal point and first decimal place figure.

Example:      24 V    : '24.0' (32<sub>16</sub>, 34<sub>16</sub>, 2E<sub>16</sub>, 30<sub>16</sub>)  
                  23.5 V : '23.5' (32<sub>16</sub>, 33<sub>16</sub>, 2E<sub>16</sub>, 35<sub>16</sub>)

The detected voltage is an average of the voltages measured every 5 ms. Note that the voltage may be lowered instantaneously while printing.  
If the voltage is out of the V<sub>p</sub> voltage tolerance (10 V to 27 V), an error occurs and you cannot enter any function code and data.

**DC2+ ' s '****External RAM Check Sum Response**

Code	12 <sub>16</sub> + 73 <sub>16</sub> , or 18 <sub>10</sub> + 115 <sub>10</sub>
Function	Outputs check sum of memory (RAM) by serial signal.
Operation	Check sum of memory is calculated and represented by hexadecimal 4-byte characters. For instance, when the result of check sum is 12AB <sub>16</sub> , the transmission data is 31 <sub>16</sub> , 32 <sub>16</sub> , 41 <sub>16</sub> and 42 <sub>16</sub> in the order starting from the first byte. Consequently, 4 byte data is sent.

Code	$12_{16} + 68_{16} + n$ , or $18_{10} + 104_{10} + n$
Function	Returns a dot information of a thermal head open error.
Operation	A dot information of a thermal head open error is transferred by a fixed byte length.

A byte length differs according to a printer model.

An each bit to be transferred is corresponded to an each heat element of the thermal head.

"1" shows a open error dot. "0" shows a normal dot.

Data is transferred from the LSB of the last byte by serial transfer.

- Number of bytes and models

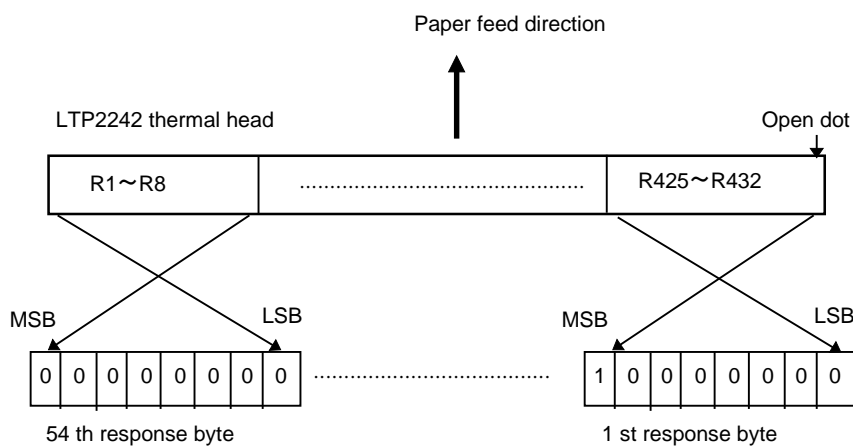
LTP2242 : 54 bytes

LTP2342 : 72 bytes

LTP2442 : 104 bytes

(example) An open error occurred, in R432 (last dot of a 2 inch thermal head.)

- Bytes transferred and heat elements



54 bytes data are transferred from the 1st response byte in the order of  $80_{16}$ ,  $00_{16}$ ,  $00_{16}$ , .....,  $00_{16}$ .

```

100 OPEN "LPT1:BIN" FOR OUTPUT AS #1.
110 CHR$(&H12);">1";"Fixed Division"
120 CHR$(&H12);"%";CHR$(8);"Dynamic Division 64 dots"
130 FOR I=70 TO 130 STEP 10
140   CHR$(&H12);"~";CHR$(I);"Print Density";I;"%"
150 NEXT I
160 CHR$(&H12);"~";CHR$(100);
170 OPEN "COM:N81NN" FOR INPUT AS #1
180 CHR$(&H12);"r";
190 "Rest Memory      : ";INPUT$(6,#1);"h Byte"
200 CHR$(&H12);"s";
210 "Memory Check Sum : ";INPUT$(4,#1);"h"
220 CHR$(&H12);"v";
230 "Vp voltage       : ";INPUT$(4,#1);"V"
240 CHR$(&H12);"qP";
250 IF INPUT$(1,#1)="P" THEN PRINT "OK"
260 LPRINT CHR$(&H12);"h";
270 FOR I=1 TO 104
280   D$=INPUT$(1,#1)
290   LPRINT USING"###@";(105-I);": ";
300   LPRINT RIGHT$("0"+HEX$(ASC(D$)),2);" ";
310   IF (I MOD 7)=0 THEN LPRINT CHR$(&HA);
320 NEXT I
330 LPRINT CHR$(&HA);
340 LPRINT CHR$(&H12);"e1";
350 PRINT "Indicate error when an error occurs."
360 PRINT "When you want to stop program, depress Ctrl + Break key."
370 *LOOP
380 E=ASC(INPUT$(1,#1))
390 IF (E AND 1)<>0 THEN PRINT "Paper empty error ! "
400 IF (E AND 2)<>0 THEN PRINT "Head Up error! "
410 IF (E AND 4)<>0 THEN PRINT "Head temperature error ! "
420 IF (E AND 8)<>0 THEN PRINT "VP voltage error ! "
430 IF E=&H60 THEN PRINT "No error. "
440 GOTO *LOOP

```

Figure 8-17 Program Sample 9



Fixed Division  
 Dynamic Division 64 dots  
 Print Density 80 %  
 Print Density 90 %  
 Print Density 100 %  
 Print Density 110 %  
 Print Density 120 %  
 Print Density 130 %  
 Rest Memory : 001300h Byte  
 Memory Check Sum : 1240h  
 Vp voltage : 24.0V  
 OK

104:00	103:00	102:00	101:00	100:00	99:00	98:00
97:00	96:00	95:00	94:00	93:00	92:00	91:00
90:00	89:00	88:00	87:00	86:00	85:00	84:00
83:00	82:00	81:00	80:00	79:00	78:00	77:00
76:00	75:00	74:00	73:00	72:00	71:00	70:00
69:00	68:00	67:00	66:00	65:00	64:00	63:00
62:00	61:00	60:00	59:00	58:00	57:00	56:00
55:00	54:00	53:00	52:00	51:00	50:00	49:00
48:00	47:00	46:00	45:00	44:00	43:00	42:00
41:00	40:00	39:00	38:00	37:00	36:00	35:00
34:00	33:00	32:00	31:00	30:00	29:00	28:00
27:00	26:00	25:00	24:00	23:00	22:00	21:00
20:00	19:00	18:00	17:00	16:00	15:00	14:00
13:00	12:00	11:00	10:00	9:00	8:00	7:00
6:00	5:00	4:00	3:00	2:00	1:00	

Figure 8-18 Print Sample 9

### 8.4.12 Auxiliary Function

**ESC+ '#'+ n**

**Overlapping Mode Select**

**Code**  $1B_{16} + 23_{16} + n$ , or  
 $27_{10} + 35_{10} + n$

**Function** Sets the overlapping mode of character with bit image, ruled line or stamp to OR or XOR.

Only the least significant bit is valid for n.  
n=0: OR overlapping mode (default)  
n=1: XOR overlapping mode

**Operation** The image such as bit image, ruled line and stamp can be printed out overlapped with characters.  
If you specify OR overlapping mode, either the part where dots exist in either image or character is print out in black.  
If you specify XOR overlapping mode, the part where image is overlapped with character is printed out in white and the non-overlapped part is print out in black.  
The last mode selected in the line becomes effective.

**ESC+' '+ n**

**Image LSB/MSB Select**

**Code**  $1B_{16} + 3D_{16} + n$ , or  
 $27_{10} + 61_{10} + n$

**Function** Sets the leftmost position of the print image data to LSB or MSB.

Only the least significant bit is valid for n.  
n=0: Sets LSB to the leftmost position (default)  
n=1: Sets MSB to the leftmost position

**Operation** This function code is valid when you specify the following function codes.

DC2+'P'	Option Font Define
DC2+'T'	Stamp Define
DC3+'V'	Ruled Line Image Define
DC3+'F'	Ruled Line Pattern Fill
ESC+'V'	Bit Image Specify
ESC+'+', FS+'2'	User defined character Define
ESC+'&'	Downloaded character Define

If these function codes are input with setting LSB to the leftmost position, the font or pattern data is defined or printed out so that the least significant bit comes at the left of print-out.

If these function codes are input with setting MSB to the leftmost position, the font or pattern data is defined or printed out so that the most significant bit comes at the left of print-out.

**DC2+' p '+ n**

**Out-of-paper Select**

- Code**  $12_{16} + 70_{16} + n$ , or  
 $18_{10} + 112_{10} + n$
- Function** Specifies whether or not the error processing operation is performed when an out-of-paper error occurs.
- Only the least significant bit is valid for n.  
 n=0 : Out-of-paper error is invalid.  
 n=1 : Out-of-paper error is valid (default).
- Operation** If out-of-paper error processing is activated, the system stops entering data or printing and goes to the error mode if it detects out-of-paper while printing or feeding. If out-of-paper error processing is not activated (n=0), the system prints when data is entered, even if it detects out-of-paper. At this time, the PE signal (CN6-14) becomes "high", when it detects out-of-paper.

**DC2+ ' m '+ s + nl + nh**

**Mark Position Detect**

- Code**  $12_{16} + 6D_{16} + s + nl + nh$ , or  
 $18_{10} + 109_{10} + s + nl + nh$
- Function** Feeds paper until a mark is detected.  
 s : sets the direction of paper feed, and the conditions of stoppage.  
 The least significant 2 bits are valid.  
 nl, nh : maximum number of dot lines for paper feed  
 nl represents the least significant byte.  
 nh represents the most significant byte.  
 Maximum dot lines= $nh \times 256 + nl$   
 $0 \leq nhnl \leq FFFF_{16}$  (65535)
- Operation** The out-of-paper sensor built in the printer mechanism is used to detect a mark.
- Deactivate out-of-paper error processing using the out-of-paper select function code (DC2+'p'+0).
- "s" specifies the direction of paper feed and the conditions of stoppage. The least significant 2 bits of "s" are valid. Each bit is defined as follows:
- | <b>Bit</b> | <b>Function</b>         | <b>0</b>                   | <b>1</b>                    |
|------------|-------------------------|----------------------------|-----------------------------|
| 1          | Direction of paper feed | Forward                    | Reverse                     |
| 0 (LSB)    | Conditions of stop      | Paper loaded ("Low" level) | Out-of-paper ("High" level) |
- The system stops if the conditions of stoppage specified by "s" are met or paper feed for the maximum number of dot lines specified with nhnl is complete.

**CAN****Cancel**

Code	18 <sub>16</sub> , or 24 <sub>10</sub>
Function	Cancels the contents of the line buffer that have been received before this code.
Operation	The double-width mode with automatic reset (SO), if on, is reset. The other mode settings remain unchanged.

**ESC+'@'****Reset**

Code	1B <sub>16</sub> + 40 <sub>16</sub> , or 27 <sub>10</sub> + 64 <sub>10</sub>
Function	Initializes the IF2202-02B.
Operation	The IF2202-02B clears the contents of the line and input buffers.

The PE(CN6-14) signal becomes high, and !ERR(CN6-15) and all ST0 to 2(CN6-16 to18) signals become low while the reset is processing. After input of the reset code, you must confirm that the PE(CN6-14) signal becomes high, !ERR(CN6-15) and all ST0 to 2 (CN6-16 to18) signals become low and inputting the data is ready before inputting the next data. Be careful that all data which has been input before reset processing disappears after reset code input.

**DC2+'R'****External RAM Initialization**

Code	12 <sub>16</sub> + 52 <sub>16</sub> , or 18 <sub>10</sub> + 82 <sub>10</sub>
Function	Initializes memory (RAM).
Operation	The IF2202-02B clears all user-defined, downloaded characters, option fonts , stamps, and character strings and initializes RAM.

After initialization, memory areas indicated in **Table 8-1** are allocated.

See **Section 8.3, MEMORY**.

The memory cannot be initialized by a reset signal or software reset command.

**ESC+'i'**

## Full cut

Code	$1B_{16} + 69_{16}$ , or $27_{10} + 105_{10}$
Function	Cut the paper fully.
Operation	During cutting, printing and paper feeding is stopped. This command is valid only when an auto-cutter is connected. If this command is sent without auto-cutter connection, the IF2202-02B enters the return-waiting status after generating the auto-cutter error. The interval between each cutting operation must be two seconds or more.

**ESC+'m'**

## Partial cut

Code	$1B_{16} + 6D_{16}$ , or $27_{10} + 109_{10}$
Function	Cut the paper partially.
Operation	During cutting, printing and paper feeding is stopped. This command is valid only when an auto-cutter is connected. If this command is sent without auto-cutter connection, the IF2202-02B enters the return-waiting status after generating the auto-cutter error. The interval between each cutting operation must be two seconds or more.

```

100 OPEN "LPT1:BIN" FOR OUTPUT AS #1
110 PRINT #1,CHR$(&H1B);"l1"; "ABCDEFGG";
120 PRINT #1,CHR$(&H12); "M";CHR$(1);CHR$(10);CHR$(0);
130 PRINT #1, "ABCDEFGHIIJ";
140 PRINT #1,CHR$(&H1B); "@";
150 FOR I=0 TO 999 : NEXT I           `waiting reset
160 PRINT #1"RESET"
170 PRINT #1,CHR$(&H12); "L";CHR$(1)
180 PRINT #1,CHR$(&H12); "R";
190 PRINT #1, "External RAM Data Delete";
200 PRINT #1,CHR$(&H12); "L";CHR$(1)
210 PRINT #1,"ABCDEFGG";
220 PRINT #1,CHR$(&H18); "CANCELED";
230 PRINT #1,CHR$(&H1B); "=0";
240 PRINT #1,CHR$(&H13);"(AF";CHR$(&H3);CHR$(&HC0); ")" ;
250 PRINT #1,CHR$(&H1B); "=1" ;
260 PRINT #1,CHR$(&H13); "(BF";CHR$(&H3);CHR$(&HC0); "+)";
270 PRINT #1,CHR$(&H13); "A";
280 PRINT #1,CHR$(&H1B); "#0";"Left=LSB, Overlap print with OR."
290 PRINT #1,CHR$(&H13); "B";
300 PRINT #1,CHR$(&H1B); "#1";"Left=MSB, Overlap print with
XOR."
310 PRINT #1,CHR$(&H13); "-";
320 PRINT " Cut the paper with 5..10 cm away from paper
insertion slot."
330 PRINT " Depress any key to feed paper, message will be
printed near the paper end."
340 I$=INPUT$(1)
350 PRINT #1,CHR$(&H12); "p0" ;
360 PRINT #1,CHR$(&H12); "m" ;CHR$(1);CHR$(0);CHR$(8);
370 PRINT #1,CHR$(&H1B); "j" ;CHR$(10);
380 PRINT #1, "Paper End detected"
390 PRINT #1, "Paper END!"
400 PRINT #1,CHR$(&H12);"p1" ;
410 CLOSE #1
420 END

```

Figure 8-19 Program Sample 10

RESET  
ABCDEFGHIJ  
External RAM Data Delete

CANCELED Left L5B Overlap print PRI  
Left L5B Overlap print XOP

Paper End detected  
Paper End!

Figure 8-20 Print Sample 10

### 8.4.13 Bar Codes

Data is converted to a bar code. 9 types of bar codes are supported.  
 Refer to the specifications of each bar code.  
 Setting values differ according to bar code reader used.

<b>GS+ 'k' + n + DATA</b>	<b>Bar Code Print Select</b>
---------------------------	------------------------------

<b>Code</b>	$1D_{16} + 6B_{16} + n + DATA$ or $29_{10} + 107_{10} + n + DATA$
-------------	--

<b>Function</b>	Convert data to a bar code according to the specified code and print it. Only the least significant 4 bits of n are valid.
-----------------	---

- n=0:    UPC A
- n=1:    UPC E
- n=2:    JAN 13
- n=3:    JAN 8
- n=4:    CODE 39
- n=5:    ITF
- n=6:    CODABAR
- n=7:    CODE 93
- n=8:    CODE 128

If n is out of range, this command is ignored.

<b>Operation</b>	If input data does not meet the bar code system, no bar code is printed. If the specified bar code width and the quiet zone width exceed the printable area, the command is ignored.
------------------	--

Each bar code data is input as follows:

- UPC A:    Input one of '0' to '9' 11 times, then input  $0_{10}$ .
- UPC E:    Input one of '0' to '9' 11 times, then input  $0_{10}$ .
- JAN 13:   Input one of '0' to '9' 12 times, then input  $0_{10}$ .
- JAN 8:    Input one of '0' to '9' 7 times, then input  $0_{10}$ .
- CODE 39:   Input one of ' ', '\$', '%', '+', '-', '.', '/', '0' to '9', 'A' to 'Z' a given number of times, then input  $0_{10}$ .
- ITF:       Input one of '0' to '9' a given even-number of times, then input  $0_{10}$ .
- CODABAR:   Input a start character 'A' to 'D', one of '0' to '9', '+', '-', '.', '/', ':', '\$', '-', a given number of times, a stop character 'A' to 'D', then input  $0_{10}$ .
- CODE 93:   Input  $0_{10}$  to  $46_{10}$  a given number of times, and input data  $47_{10}$  or larger.
- CODE 128:   Input a start code  $103_{10}$  to  $105_{10}$ ,  $0_{10}$  to  $102_{10}$  a given number of times, then input data  $103_{10}$  or larger.



**GS+ ' w ' + n1 + n2****Narrow / Wide width**

**Code**  $1D_{16} + 77_{16} + n1 + n2$ , or  
 $29_{10} + 119_{10} + n1 + n2$

**Function** Sets narrow and wide width of a bar code with number of dots.

n1=0 to 2: Narrow width  
 n2=0 to 3: Wide width

The least significant 2 bits are valid in n1 and n2.

Narrow Width

n1	Number of dots
0	2
1	3*
2	4

Wide Width

n1	n2			
	0	1	2	3
0	5	6	6	6
1	7	8	9*	9
2	9	10	11	12

\* Shows a default.

**GS+ ' P ' + n****Bar Code Print Position**

**Code**  $1D_{16} + 50_{16} + n$ , or  
 $29_{10} + 80_{10} + n$

**Function** Sets a print position of a bar code.

n=0 : Left (default)  
 n=1 : Center  
 n=2 : Right

The least significant 2 bits are valid in "n".

This command is ignored when n is out of range.

**GS+ 'h'+ n****Bar Code Height****Code** $1D_{16} + 68_{16} + s + n$ , or  
 $29_{10} + 104_{10} + n$ **Function**

Sets height of the bar code.

n=1 to 255 : Height of the bar code (Unit : dot line)  
n=162 (default)

**GS+ 'H'+ n****Bar Code Character Print Position****Code** $1D_{16} + 48_{16} + n$ , or  
 $29_{10} + 72_{10} + n$ **Function**

Sets a print position of a bar code character.

n=0 : No print (default)  
n=1 : Top of the bar code  
n=2 : Bottom of the bar code  
n=3 : Top and bottom of the bar code

Only the least significant 2 bits are valid in "n".

**GS+ 'f'+ n****Bar Code Character Font Size****Code** $1D_{16} + 66_{16} + n$ , or  
 $29_{10} + 102_{10} + n$ **Function**

Selects 16 dots or 24 dots for a bar code character.

n=0 : 24×12 (default)  
n=1 : 16×8

Only the LSB is valid in "n".

```

100 OPEN "LPT1:BIN" FOR OUTPUT AS #1
110 PRINT #1,CHR$(&H1D);"h";CHR$(8*10);          'Bar code height 10mm
120 PRINT #1,CHR$(&H1D);"H";CHR$(2);            'String position
130 FOR I=0 TO 6  'Null terminate Bar Code command
140 PRINT #1,CHR$(&H1D);"P";CHR$(I MOD 3);      'Print position
150 PRINT #1,CHR$(&H1D);"w";CHR$(I MOD 3);CHR$(1); 'Narrow / Wide
160 PRINT #1,CHR$(&H1D);"f";CHR$(I MOD 2);      'Character size
170 READ CODE$,DAT$
180 PRINT #1,"*****";CODE$;"*****"
190 PRINT #1,CHR$(&H1D);"k";CHR$(I);DAT$;CHR$(0) 'Print bar code
200 NEXT I
210 FOR I=7 TO 8  ' Value data Bar Code command
220 PRINT #1,CHR$(&H1D);"P";CHR$(I MOD 3);      'Print position
230 PRINT #1,CHR$(&H1D);"w";CHR$(I MOD 3);CHR$(1); 'Narrow / Wide
240 PRINT #1,CHR$(&H1D);"f";CHR$(I MOD 2);      'Character size
250 READ CODE$
260 PRINT #1,"*****";CODE$;"*****"
270 PRINT #1,CHR$(&H1D);"k";CHR$(I);            'Print bar code
280 D=0
290 WHILE D<255
300 READ D : PRINT #1,CHR$(D);
310 WEND
320 LPRINT
330 NEXT I
340 END
350 ' Bar Code Data
360 DATA "UPC A", "12345678901"
370 DATA "UPC E", "01245000002"
380 DATA "JAN 13", "491234567890"
390 DATA "JAN 8", "4912345"
400 DATA "CODE39", "CODE-39"
410 DATA "ITF", "12345678"
420 DATA "CODABAR", "A12:34/56$B"
430 DATA "CODE93", 12,24,13,14,36,9,3,255
440 DATA "CODE128",104,35,79,68,69,13,17,18,24,255

```

Figure 8-21 Program Sample 11

\*\*\*\*\*UPC A\*\*\*\*\*



\*\*\*\*\*UPC E\*\*\*\*\*



\*\*\*\*\*JAN 13\*\*\*\*\*



\*\*\*\*\*JAN 8\*\*\*\*\*



\*\*\*\*\*CODE39\*\*\*\*\*



\*\*\*\*\*ITF\*\*\*\*\*



\*\*\*\*\*CODABAR\*\*\*\*\*



\*\*\*\*\*CODE93\*\*\*\*\*



\*\*\*\*\*CODE128\*\*\*\*\*



Figure 8-22 Print Sample 11

## CHAPTER 9

### CHARACTER CODES

The character codes for the IF2202-02B are listed in **Figure 9-1**.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
80	Ç	ü	é	â	ä	à	ç	ê	ë	è	ï	í	î	ï	Ä	Å
90	É	æ	œ	ô	ö	ò	ù	û	ÿ	ÿ	ö	ü	ç	£	¥	℞
A0	á	í	ó	ú	ñ	Ñ	á	Q	ó	~	~	½	¼	i	《	》
B0	☐	☒	☓		†	‡	¶	§		¶	¶	¶	¶	¶	¶	¶
C0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
D0	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
E0	α	β	Γ	κ	Σ	σ	μ	τ	ϕ	θ	Ω	δ	∞	φ	Ε	Π
F0	≡	±	λ	κ	∫	∫	÷	≈	°	.	-	∫	n	z	.	

Figure 9-1 Extended Graphics Character Set



**CHAPTER 10**  
**ELECTRICAL CHARACTERISTICS**

**10.1 GENERAL CHARACTERISTICS**

Item	Symbol	Condition	Specification			Unit
			MIN	TYP	MAX	
Circuit power voltage	V <sub>cc</sub>		4.5	5.0	5.5	V
Reset voltage	V <sub>R</sub>	V <sub>cc</sub> : high→low	4.0	4.1	4.2	V
Printer mechanism voltage	V <sub>p</sub>		21.6	24.0	26.4	V
Circuit current consumption* <sup>1</sup>	I <sub>cc1</sub>	Standby			50	mA
	I <sub>cc2</sub>	printing			170	mA
Printer mechanism current consumption	I <sub>p1</sub>	Standby			30	mA
	I <sub>p2</sub>	Auto loading (LTP2242)			360	mA
	I <sub>p2</sub>	Auto loading (LTP2342, LTP2242)			500	mA
	I <sub>p3</sub>	Paper feed			360	mA

\*1 V<sub>cc</sub>=5 V, 25°C, no error  
When no input/output terminal is connected.

For maximum current consumption of printer mechanism during printing, see **Section 6.3, CONNECTING TO THE POWER SUPPLY.**

## 10.2 INPUT / OUTPUT SIGNAL CONDITIONS

Item	Symbol	Condition	Specification			Unit
			MIN	TYP	MAX	
Input voltage low	$V_{IL}$		0	-	$0.2V_{CC}$	V
Input voltage high	$V_{IH}$		$0.8V_{CC}$	-	$V_{CC}$	V
Output voltage low	$V_{OL}$	$I_{OL} = 1 \text{ mA}$	-	-	0.7	V
Output voltage high	$V_{OH}$	$I_{OH} = -1 \text{ mA}$	4.3	-	-	V

( $V_{CC}=5.0 \text{ V}$ ,  $T_a=0 \text{ to } 50^\circ\text{C}$ )

## 10.3 CONNECTOR LIST

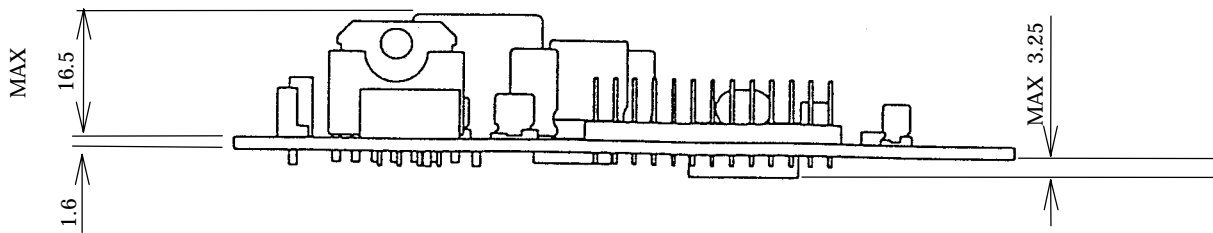
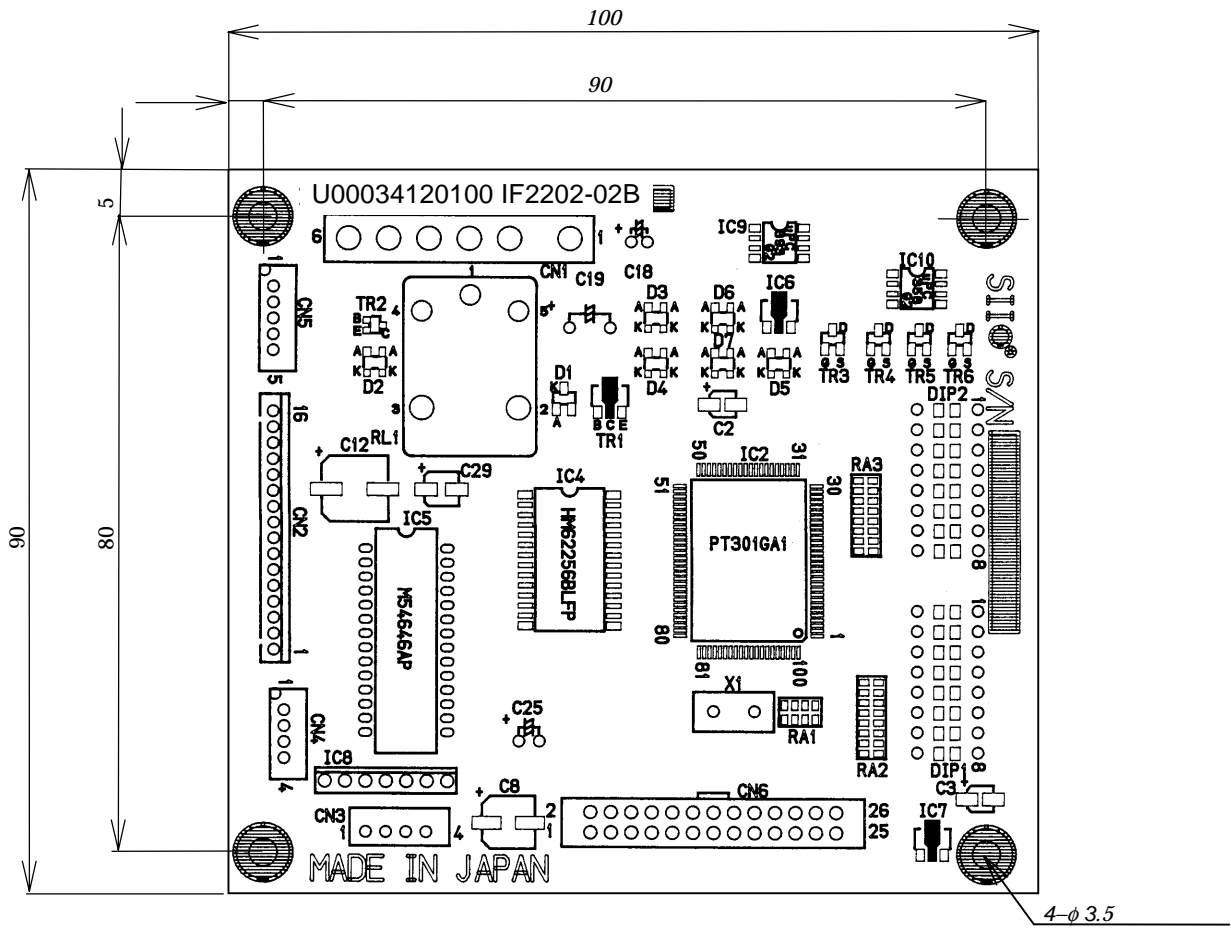
No.	Manufacturer	Model Name
CN1	AMP	171265-1
CN2	Elco.	00-8283-1612-00-000
CN3	Molex. Inc.	5483-04AX
CN4	Molex. Inc.	53324-0412
CN5	Molex. Inc.	53324-0510
CN6	OMRON	XG8W-2631

Applied receptacles to CN6 are as follows:

Manufacturer	Model Name
OMRON	XG4M
OMRON	XG5M-N



APPENDIX A  
DIMENSIONS



[Unit:mm]

Figure A-1 Dimensions



**APPENDIX B**  
**CIRCUIT DIAGRAM**

The circuit diagram for the IF2202-02B is shown in **Figure B-1**.

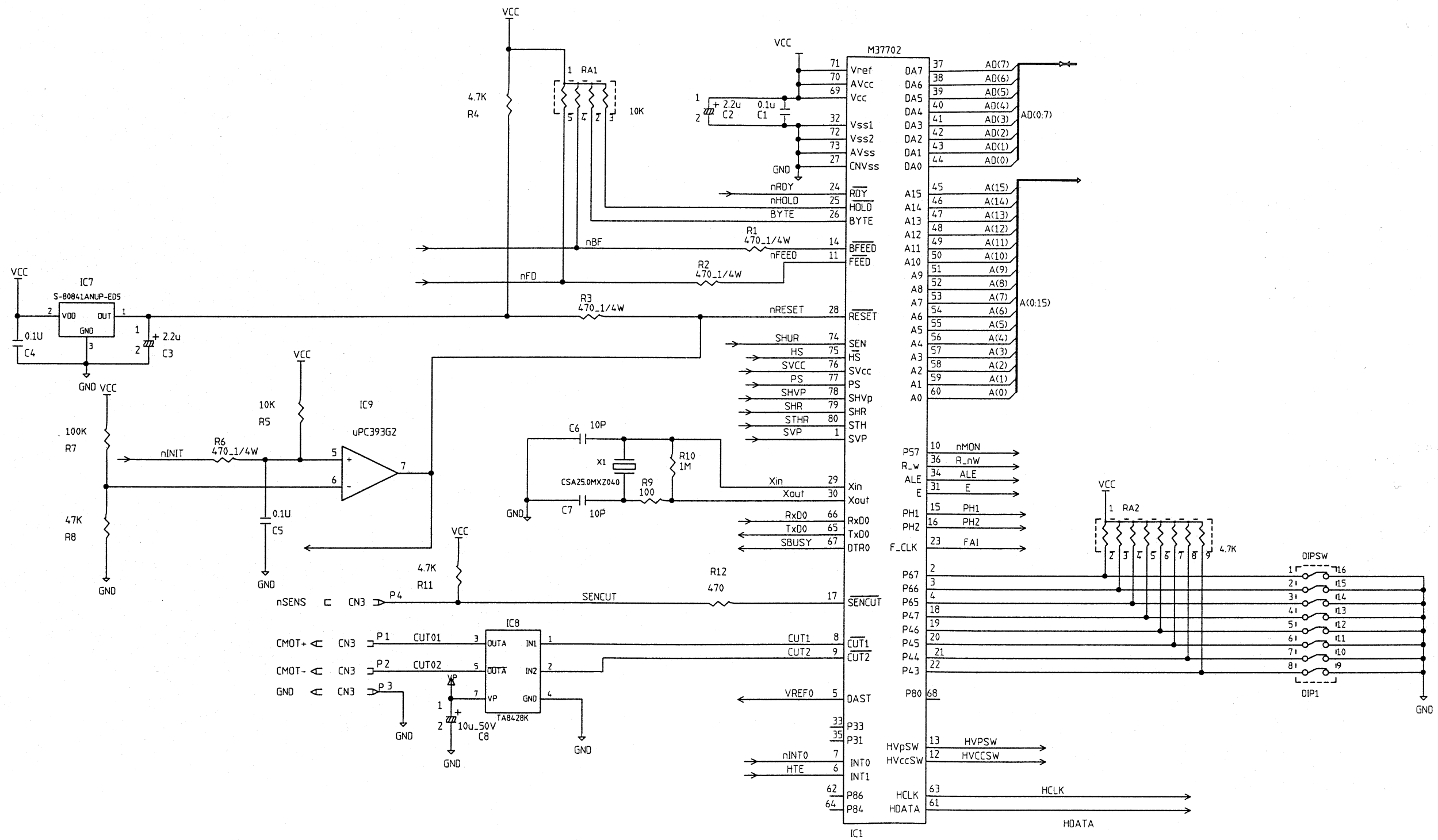


Figure B-1 Circuit Diagram for the IF2202-02B

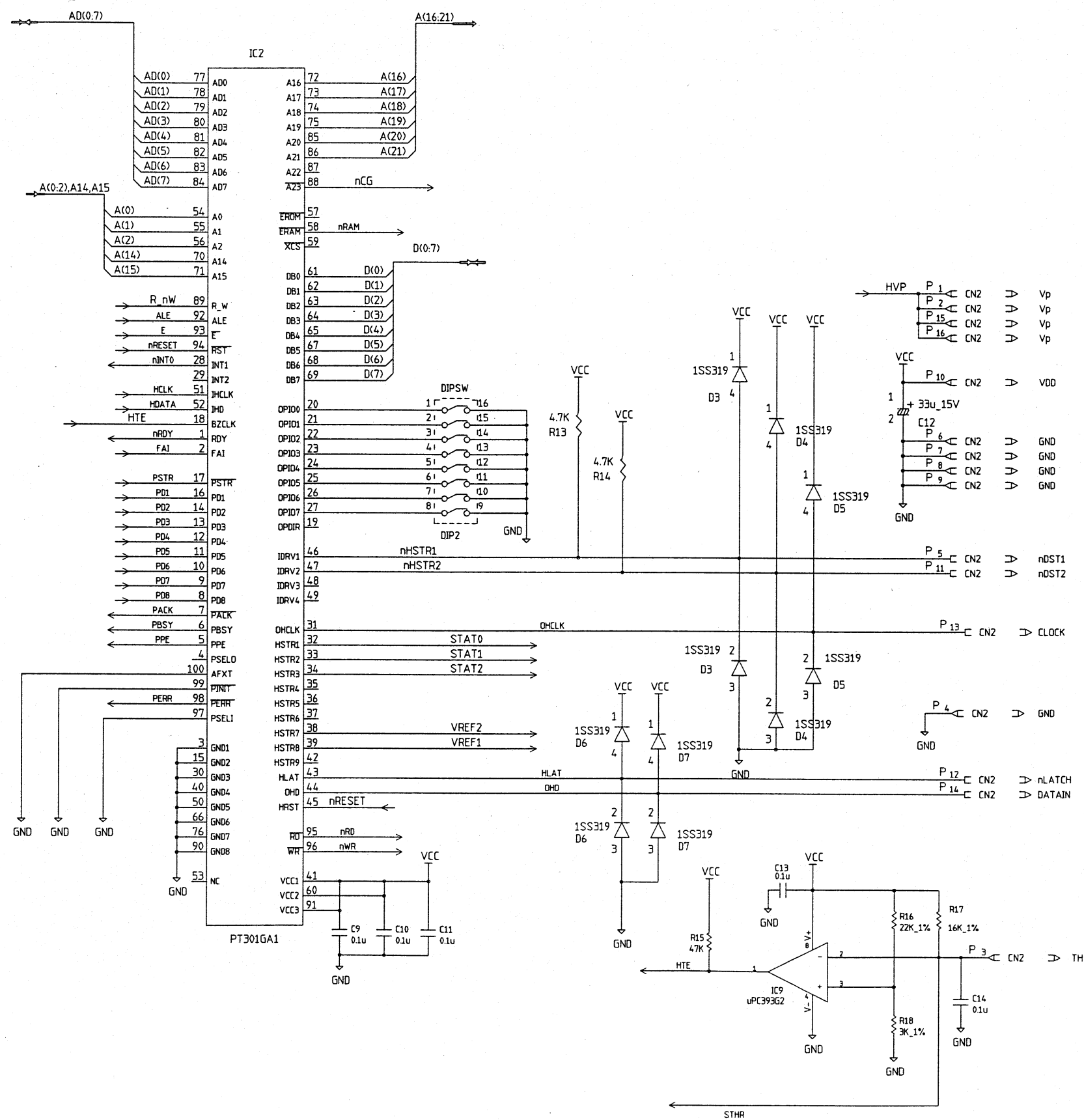


Figure B-1 Circuit Diagram for the IF2202-02B (Continued)

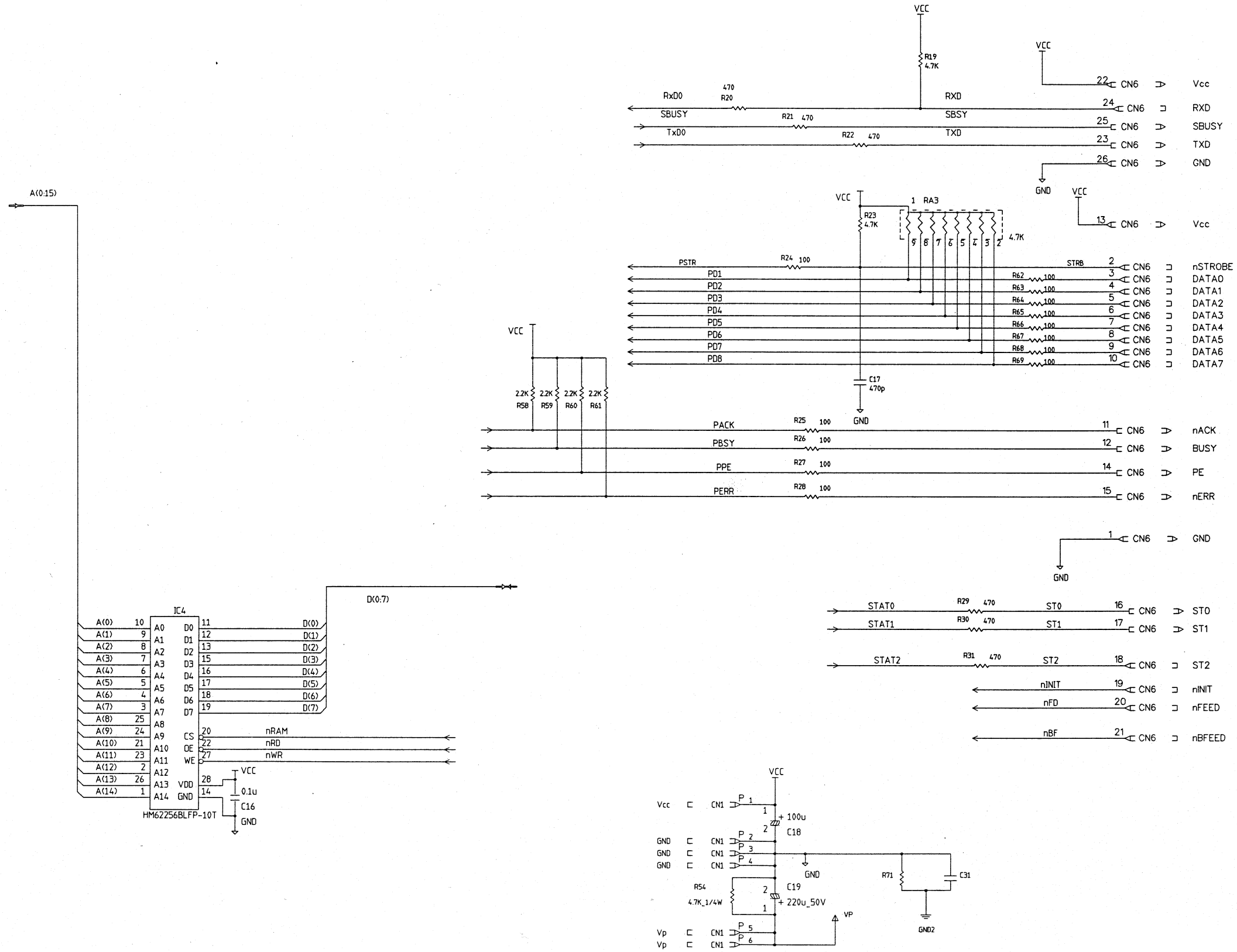


Figure B-1 Circuit Diagram for the IF2202-02B (Continued)

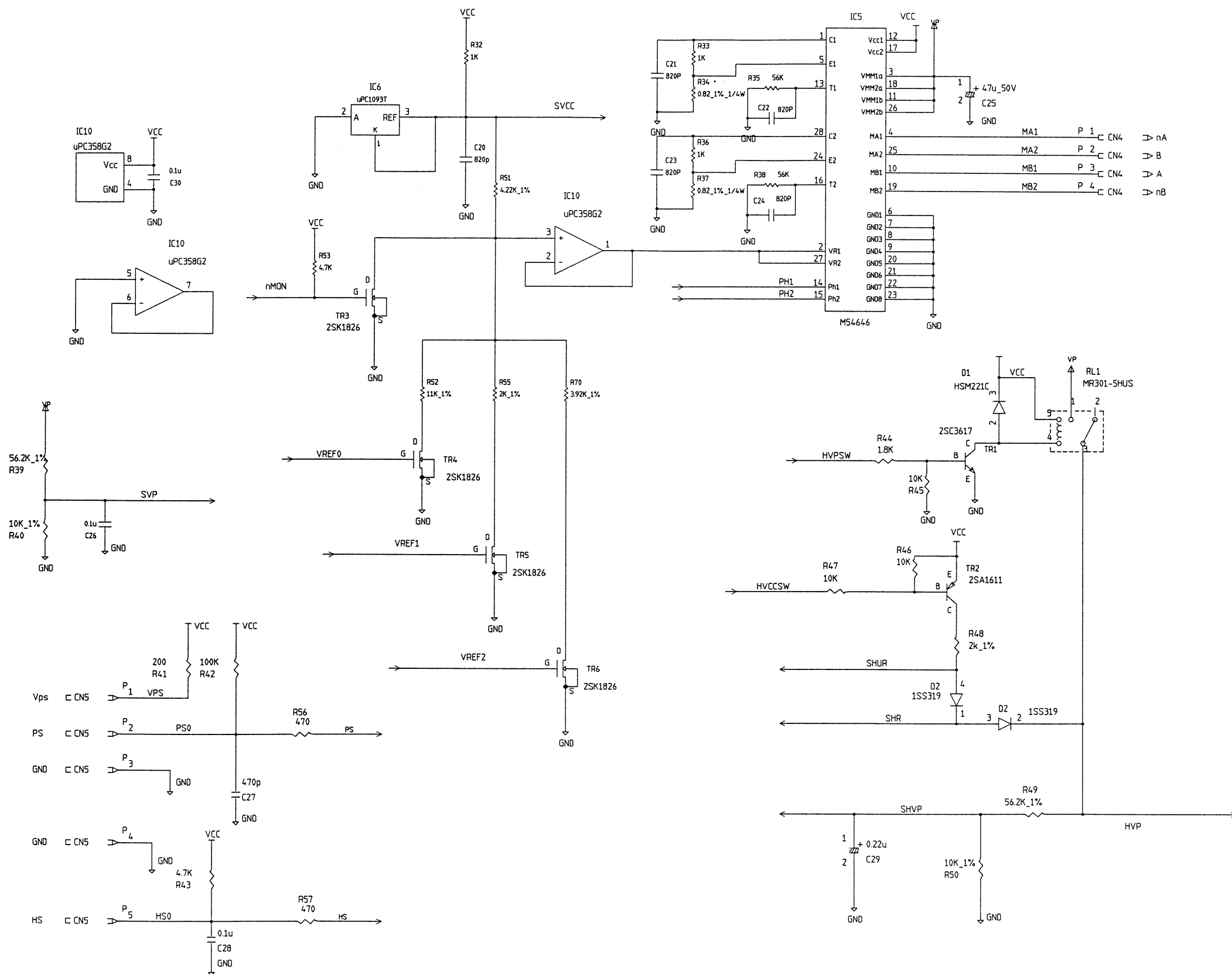


Figure B-1 Circuit Diagram for the IF2202-02B (Continued)





**APPENDIX C**  
**COMMAND INDEX**

<b>Line/Paper Feed (8.4.1)</b> .....		<b>8-7</b>
<b>LF</b>	Line Feed	8-7
<b>CR</b>	Carriage Return	8-7
<b>ESC+ ' J ' +n</b>	Print and Feed Forward	8-8
<b>ESC+ ' j ' +n</b>	Print and Feed Backward	8-8
<b>Format Setting (8.4.2)</b> .....		<b>8-9</b>
<b>ESC+ ' 2 '</b>	16-dot Line Spacing	8-9
<b>ESC+ ' 0 '</b>	4-dot Line Spacing	8-9
<b>ESC+ ' A ' +n or ESC+ ' 3 ' +n</b>	n-dot Line Spacing	8-9
<b>ESC+SP+n</b>	Character Spacing	8-9
<b>ESC+ ' s ' +nl+nr</b>	Left/Right Character Spacing	8-10
<b>ESC+ ' U ' +n</b>	Inversion (Flip)	8-10
<b>DC2+ ' Y ' +n</b>	Character Rotation	8-11
Program Sample 1 (Figure 8-1)		8-12
Print Sample 1 (Figure 8-2)		8-13
<b>Character Modification (8.4.3)</b> .....		<b>8-13</b>
<b>ESC+ ' - ' +n</b>	Underline	8-13
<b>SO</b>	Double-Width (with automatic reset ) ON	8-14
<b>DC4</b>	Double-Width (with automatic reset) OFF	8-14
<b>ESC+ ' W ' +n</b>	Double-Width	8-14
<b>ESC+ ' w ' +n</b>	Double-Height	8-15
<b>ESC+ ' I ' +n</b>	Reverse	8-15
<b>ESC+ ' E ' +n</b>	Bold Print ON	8-16
<b>ESC+ ' F ' +n</b>	Bold Print OFF	8-16
Program Sample 2 (Figure 8-3)		8-17
Print Sample 2 (Figure 8-4)		8-17

<b>Character Set (8.4.4)</b> .....		<b>8-18</b>
DC2+ ' F '+n	Font Size Selection	8-18
ESC+ ' t '+n	Character Set Select	8-18
FS+ ' C '+n	Kanji Code System Select	8-19
ESC+ ' K ', FS+ ' & '	Kanji Mode Specify	8-19
ESC+ ' H ', FS+ ' . '	Kanji Mode Clear	8-20
Program Sample 3 (Figure 8-5)		8-20
Print Sample 3 (Figure 8-6)		8-20

**Character Definition (8.4.5)** ..... **8-21**

ESC+ ' & ' + s + e+ Font Data	Downloaded Character Define	8-22
ESC+ ' % ' + n	Downloaded Character Select	8-23
DC2+ ' D '+n	Downloaded Character Area Operation	8-23
ESC+ ' + ' + k1 + k2+ Font Data		
FS+ ' 2 ' + k1 + k2+ Font Data	User-Defined Character Define	8-24
DC2+ ' G '+n	User Defined Character Area Operation	8-25
DC2+ ' P ' + s + e+ x + y +Font Data	Option Font Define	8-26
DC2+ ' O '+n	Option Font Select/Deselect	8-28
DC2+ ' Q '	Option Font Clear	8-28
Program Sample 4 (Figure 8-7)		8-29
Print Sample 4 (Figure 8-8)		8-30

**Ruler (8.4.6)** ..... **8-31**

DC3+ ' A '	Ruler Line Buffer A	8-31
DC3+ ' B '	Ruler Line Buffer B	8-32
DC3+ ' V ' + Image Data	Ruler Line Image	8-32
DC3+ ' D '+nl+nh	Define Ruler Line by Dot	8-32
DC3+ ' L '+ml+mh+nl+nh	Define Ruler Line by Line	8-33
DC3+ ' F '+n1+n2	Define Ruler Line with Repeating Pattern	8-33
DC3+ ' + '	Ruler Line ON	8-33
DC3+ ' - '	Ruler Line OFF	8-34
DC3+ ' P '	Print One Dot Line after Printing Line Buffer Data	8-34
DC3+ ' C '	Ruler Line Buffer Clear	8-34
DC3+ ' ( '	Continuous Ruler Line Control Code Input	8-34
Program Sample 5 (Figure 8-9)		8-35
Print Sample 5 (Figure 8-10)		8-35

<b>Bit Image (8.4.7)</b> .....		<b>8-36</b>
<b>ESC+ ' V '+nl+nh+Image Data</b>	Bit Image Specify	8-36
Program Sample 6 (Figure 8-11)		8-37
Print Sample 6 (Figure 8-12)		8-37
<b>Stamp (8.4.8)</b> .....		<b>8-38</b>
<b>DC2+ ' T ' + n + x+ yl + yh+ Stamp Data</b>		
	Stamp Define	8-39
<b>DC2+ ' S '+n +x</b>	Stamp Select	8-40
<b>DC2+ ' V '</b>	Stamp Print Out	8-40
<b>DC2+ ' W '</b>	Stamp Abort	8-41
<b>DC2+ ' U ' + n</b>	Stamp Clear	8-41
Program sample 7 (Figure 8-13)		8-42
Print Sample 7 (Figure 8-14)		8-43
<b>Character String (8.4.9)</b> .....		<b>8-43</b>
<b>DC2+ ' M ' + n + dl + dh+ Character String Data</b>		
	Character String Define	8-44
<b>DC2+ ' L '+n</b>	Character String Select and Execute	8-44
<b>DC2+ ' N ' + n</b>	Character String Clear	8-45
Program Sample 8 (Figure 8-15)		8-45
Print Sample 8 (Figure 8-16)		8-46
<b>Print Method (8.4.10)</b> .....		<b>8-46</b>
<b>DC2+ ' &gt; ' + n</b>	Fixed Division Select	8-46
<b>DC2+ ' % ' + n</b>	Dynamic Division, Number of Dots Specify	8-47
<b>DC2+ ' ~ ' + n</b>	Print Density Specify	8-47
<b>Response Function (8.4.11)</b> .....		<b>8-48</b>
<b>DC2+ ' r '</b>	Remaining Memory Response	8-48
<b>DC2+ ' e ' + n</b>	Error Status Response	8-49
<b>DC2+ ' q ' + n</b>	Execute Response Request	8-50
<b>DC2+ ' v '</b>	Vp Voltage Response	8-50
<b>DC2+ ' s '</b>	External RAM Check Sum Response	8-50
<b>DC2+ ' h '</b>	Thermal Head Response	8-51
Program Sample 9 (Figure 8-17)		8-52
Print Sample 9 (Figure 8-18)		8-53

**Auxiliary Function (8.4.12) .....8-54**

<b>ESC+ ' # ' + n</b>	Overlapping Mode Select	8-54
<b>ESC+ ' = ' + n</b>	Image LSB/MSB Select	8-54
<b>DC2+ ' p ' + n</b>	Out-of-paper Select	8-55
<b>DC2+ ' m ' + s + nl + nh</b>	Mark Position Detect	8-55
<b>CAN</b>	Cancel	8-56
<b>ESC+ ' @ ' + n</b>	Reset	8-56
<b>DC2+ ' R ' + n</b>	External RAM Initialization	8-56
<b>ESC+ ' i ' + n</b>	Full cut	8-57
<b>ESC+ ' m ' + n</b>	Partial cut	8-57
Program Sample 10 (Figure 8-19)		8-58
Print Sample 10 (Figure 8-20)		8-59

**Bar Codes (8.4.13) .....8-60**

<b>GS+ ' k ' + n + DATA + NUL</b>	Bar Code Print	8-60
<b>GS+ ' w ' + n1 + n2</b>	Narrow / Wide width	8-61
<b>GS+ ' P ' + n</b>	Bar Code Print Position	8-61
<b>GS+ ' h ' + n</b>	Bar Code Height	8-62
<b>GS+ ' H ' + n</b>	Bar Code Character Print Position	8-62
<b>GS+ ' f ' + n</b>	Bar Code Character Font Size	8-62
Program Sample 11 (Figure 8-21)		8-63
Print Sample 11 (Figure 8-22)		8-64