



**DCS & Labelling Worldwide**

# SATO Smart Keyboard Programming manual



## SATO Group of Companies

<p><b>BARCODE SATO INTERNATIONAL PTE LTD</b></p> <p>438A Alexandra Road #05-01/ 02, Alexandra Technopark, Singapore 119967 Tel: +65-6271-2122 Fax: +65-6271-2151 Website: www.barcodesato.com Email: sales@sato-int.com</p>	<p><b>SATO EUROPE NV</b></p> <p>Leuvensesteenweg 369, 1932 Sint-Stevens-Woluwe, Brussels, Belgium Tel: +32 (0)-2-788-80-00 Fax: +32 (0)-2-788-80-80 Website: www.sato-europe.com Email: info@sato-europe.com</p>
<p><b>SATO UK LTD</b></p> <p>Valley Road, Harwich, Essex England Co12 4RR, United Kingdom Tel: +44-1255-240000 Fax: +44-1255-240111 Website: www.satouk.com Email: enquiries@satouk.com</p>	<p><b>SATO DEUTSCHLAND GMBH</b></p> <p>Schaberweg 28, 61348 Bad Homburg, Germany Tel: +49 (0)-6-1726-8180 Fax: +49 (0)-6-1726-818-199 Website: www.sato-deutschland.de Email: info@sato-deutschland.de</p>
<p><b>SATO POLSKA SP Z O.O.</b></p> <p>UI Okolna 2, 50-422 Wroclaw Poland Tel: +48-71-335-23-20 Fax: +48-71-335-23-25 Website: www.sato-europe.com Email: info@sato-europe.com</p>	<p><b>SATO AMERICA INC.</b></p> <p>10350 Nations Ford Road Suite A, Charlotte, NC 28273 Tel: +1-704-644-1650 Fax: +1-704-644-1662 Website: www.satoamerica.com Email: satosales@satoamerica.com</p>
<p><b>SATO SHANGHAI CO, LTD</b></p> <p>11 Floor, D, Pudong South Road 1111, Pudong New Area, Shanghai, China 200120 Tel: +86 (0)-21-58307080 Fax: +86 (0)-21-58307978 Website: www.barcodesato.com Email: sales@satochina.com</p>	<p><b>BAR CODE SATO ELECTRONICS (S) PTE LTD</b></p> <p>438A Alexandra Road #05-01/02, Alexandra Technopark, Singapore 119967 Tel: +65-6271-5300 Fax: +65-6273-6011 Website: www.barcodesato.com www.satosingapore.com Email: sales@satosingapore.com</p>
<p><b>SATO BAR CODE &amp; LABELLING SDN BHD</b></p> <p>Suite B-08-11&amp;12, Block B Plaza Mont' Kiara, No.2, Jalan Kiara Mont' Kiara, 50480 Kuala Lumpur, Malaysia Tel: +60-3-6203-5950 Fax: +60-3-6203-1209 Website: www.barcodesato.com Email: sales@satosbl.po.my</p>	<p><b>BARCODE SATO (THAILAND) CO. LTD</b></p> <p>370/8 Supattra Building, 5th Floor, Rama9 Road, Bangkapi, Huay Kwang Bangkok 10320, Thailand Tel: +662-719-7780-3 Fax: +662-719-7784 Website: www.barcodesato.com Email: sales@satothailand.co.th</p>

- The information contained herein is subject to change without notice.
- Reproduction of this manual either in part or its entirety is forbidden.
- Note that the manufacturer assumes no responsibility for any injury or loss incurred while using this manual.

**CONTENT**

DOCUMENT CONVENTIONS .....5

FEATURES OF K012.....6

SPECIFICATIONS .....6

COMMUNICATION INTERFACE .....7

SAFETY REGULATION .....8

CHECK-LIST.....8

OPTIONS.....8

KEYBOARD SETUP .....9

KEYBOARD OPERATION .....9

POWER-ON UTILITIES .....9

AUTO EXECUTION .....9

KEYBOARD INITIALIZATION .....9

KEYBOARD CONFIGURATION SETUP .....10

KEYBOARD BIOS UPDATE.....12

ON-LINE EDITING UTILITY .....12

EXTENDED FILE MANAGER .....13

EXECUTE DOWNLOADED BASIC PROGRAM.....14

USING K012 .....15

EDIT A PROGRAM .....15

DOWNLOAD A PROGRAM.....15

EXECUTE A PROGRAM.....16

VARIABLES, CONSTANTS, FUNCTIONS, OPERATORS, AND EXPRESSIONS .....17

VARIABLES .....17

LONG INTEGER DATA TYPE VARIABLES .....18

FLOAT DATA TYPE VARIABLES .....18

DOUBLE FLOAT DATA TYPE VARIABLES .....18

STRING VARIABLES .....18

SYSTEM VARIABLES .....18

FUNCTIONS .....19

NUMERIC FUNCTIONS.....19

STRING FUNCTIONS.....19

OPERATORS.....19

ARITHMETIC OPERATORS .....19

STRING OPERATORS .....19

RELATIONAL OPERATORS .....19

EXPRESSIONS .....20

ARITHMETIC EXPRESSIONS.....	20
STRING EXPRESSIONS.....	20
RELATIONAL EXPRESSIONS.....	20
COMMANDS AND STATEMENTS.....	21
TSKL COMMANDS.....	22
<ESC>!R.....	22
<ESC>!W.....	23
SOUND.....	24
CLS.....	25
CLEAR.....	26
CURSOR.....	27
LOCATE.....	28
PRINT...[USING].....	29
INPUT.....	30
OUT.....	31
FOUT.....	32
OUT USING.....	33
DOWNLOAD.....	34
EOP.....	35
OPEN.....	36
CLOSE.....	37
KILL.....	38
SEEK.....	39
READ.....	40
WRITE.....	41
GOSUB...RETURN.....	42
LOOP STATEMENTS.....	43
DO...[EXITDO]...LOOP.....	43
FOR...[EXITFOR]...NEXT.....	45
WHILE...WEND.....	46
GOTO.....	47
PROGRAM-CONTROL STATEMENTS.....	48
IF...THEN...ELSE.....	48
END.....	50
DIM.....	51
FREE.....	52
REM.....	53
TSKL FUNCTIONS.....	54

---

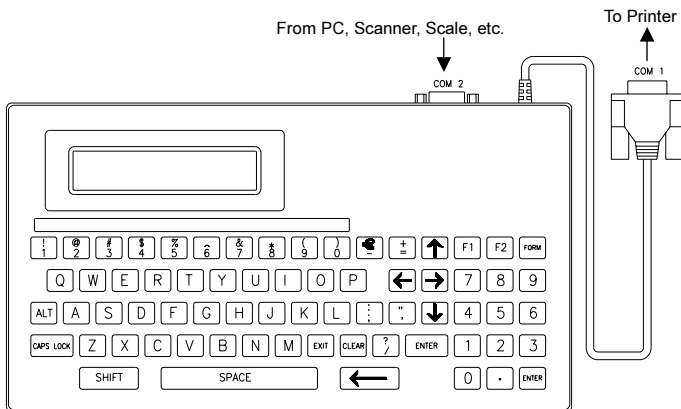
POS()	54
INKEY()	55
INP\$()	56
EOF()	57
LOF()	58
ABS()	59
ASC()	60
LEN()	61
FRE()	62
INT()	63
VAL()	64
RND()	65
CHR\$()	66
FREAD\$()	67
LEFT\$()	68
RIGHT\$()	69
MID\$()	70
STR\$()	71
SPC\$()	72
SYSTEM VARIABLES	73
YEAR	73
MONTH	74
DATE	75
HOUR	76
MINUTE	77
IDNUMBER\$	79
APPENDIX A - LIMITATIONS OF TSKL	80
APPENDIX B - ERROR MESSAGES	81
APPENDIX C - RS-232 PIN CONFIGURATIONS	84
OPERATION GUIDE	87

## Document Conventions

This manual uses the following typographic conventions

Examples of convention	Description
< >	Angle brackets, enclose mnemonic representations of ASCII control characters.
[ ]	Square brackets, the data within square brackets is optional.
<b>Ctrl</b>	Bold letters represent a key on the keyboard.
<b>Ctrl-C</b>	Two keys with a dash, means to press them simultaneously.
<i>Italic</i>	Italic letters represent explanations given in the context.
<b><i>Bold Italic</i></b>	Note. Important information.

## The Keyboard



## Features of K012

- 68 keys with keypad
- LCD screen (20 characters × 2 lines)
- Additional RS-232 port (COM 2) for another input device
- Maximum of 50 files can be stored in memory
- Upload or download files through both serial ports
- Real Time Clock (Y2K compatible)
- Built-in Euro logo (ASCII 176, 177)
- Floating point calculation
- Password security locking
- Auto execution function

## Specifications

### ● Keyboard Unit

<b>Size</b>	261mm(L)×142.2mm(W)×31mm(H)
<b>Weight</b>	440g
<b>LCD</b>	20 characters × 2 lines
<b>Max Current</b>	5V, 250mA
<b>External Power</b>	5V, 150mA
<b>Operating Temperature</b>	40°F~104°F (5°C~40°C)
<b>Storage Temperature</b>	-4°F~140°F (-20°C~60°C)

### ● Memory

<b>FLASH</b>	512 KB for system, 1 MB for application
<b>SRAM</b>	128 KB for system, 128KB for application



## Communication Interface

The available communication parameters for both serial ports are listed below:

Bit rate: 2400, 4800, 9600, 19200 bps  
 Parity check: none, even or odd  
 Data bits: 7 or 8  
 Stop bit(s): 1 or 2

### COM1 Port

Serial interface, COM1, is a 9-pin, male, D-style subminiature connector with cable. The pin assignments are shown below:

Pin	Configuration
1	Power input 5 volts, 250mA
2	RxD
3	TxD
4	DTR
5	Ground
6	DSR
7	RTS
8	CTS
9	Connected with Pin1 internally

### COM2 Port

COM2 is a 9-pin, female, D-style subminiature connector with pin assignments as shown as below:

Pin	Configuration
1	Power input 5 volts, 250mA
2	TxD
3	RxD
4	DSR
5	Ground
6	DTR
7	CTS
8	RTS
9	Connect with Pin1 internally

## Safety Regulation

FCC Class A

CE EMC

## Check-List

Verify the contents of the container according to the list below, if any parts are missing, please contact your local representative.

- K012 keyboard unit
- K012 User's Manual
- 25 pin to 9 pin RS-232 converter
- 2 screws for SI thread

## Options

- External power set
  - AC adapter  
Input: 110V AC or 220V AC  
Output: 5V DC
  - RS-232 cable with power adapter jack

## Keyboard Setup

Please turn off printer power *prior* to connecting the K012 COM1 port to the printer. The COM2 port of the keyboard is used for downloading files from a PC or may be connected to other input devices such as a bar code scanner or scale.

## Keyboard Operation

### ● Power-on Utilities

#### ■ Auto Execution

At Power ON, the K012 will automatically execute a program without pressing any key, if one of the downloaded programs is named "AUTO.BAS".

Press **ALT-F1**, while applying power, to disable the auto execution function. This should be done prior to keyboard configuration setup, on-line editing, initialization and extended file manager utilities, *if an auto execution file is downloaded in the keyboard.*

#### ■ Keyboard Initialization

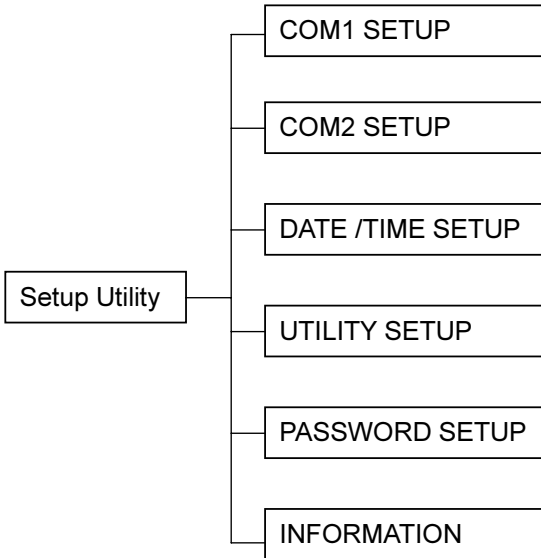
Pressing **ALT-CLEAR** while turning on the printer power will initialize the keyboard. All files stored in the SRAM will be deleted. Code page and country codes are set to default values of 437 and 001 respectively. Both serial port communications parameters are set to 9600 bps, no parity check, 8 data bits and 1 stop bit.

Items	Default Parameters
COM1 Port	9600,N,8,1,Xon/Xoff
COM2 Port	9600,N,8,1,DSR/DTR
Country Code	001
Code Page	437
Key Sound	On
ID Number*	T.S.C.
Password*	None

**Note: The items with asterisk (\*) can't be initialized.**

● **Keyboard Configuration Setup**

Keyboard configurations can be changed by pressing the **ALT-F1** keys. The left and right arrow keys are used to select different parameters. Press **ENTER** to enable the selected parameter. Press **EXIT** to return to the top of the menu.



■ **COM1, COM2 SETUP:**

With COM1, COM2 SETUP, users can select different communication parameter as well as handshaking to download or upload data.

■ **DATE / TIME SETUP:**

The built-in Real Time Clock is compatible with year 2000. The leap year timing is automatic. The available setting of year is from 1950 to 2049.

■ **INFORMATION:**

The information lists the available memory (SRAM) in system and the extended Flash memory status. If the extended memory is installed, it shows 8M. Otherwise “None” is shown on the display.

■ **UTILITY SETUP:**

The code page, country code and key sound ON/OFF is set in UTILITY SETUP.

The available code pages and country code are listed as below:

➤ **Code pages**

- 437: United States
- 850: Multilingual
- 852: Slavic
- 860: Portuguese
- 863: Canadian/French
- 865: Nordic

➤ **Country codes**

- 001: USA
- 002: Canadian French
- 003: Spanish (Latin America)
- 031: Dutch
- 032: Belgian
- 033: French (France)
- 034: Spanish (Spain)
- 036: Hungarian
- 038: Yugoslavian
- 039: Italian
- 041: Switzerland
- 042: Slovak
- 044: United Kingdom
- 045: Danish
- 046: Swedish
- 047: Norwegian
- 048: Polish
- 049: Germany
- 055: Brazil
- 061: English (International)
- 351: Portuguese
- 358: Finnish

There is a beep after each keystroke. It can be disabled by setting the key sound OFF.

■ **PASSWORD SETUP:**

K012 supports password security. With password locked on, the download/upload programs, on-line editing, deleting files, firmware upgrade and extended file

manager will be disabled. The maximum number of password characters is 14, which are not case sensitive.

**Note: If the password is lost, please contact the Customer Service Department of your distributor or reseller.**

K012 provides another password namely ID Number, which can be verified in programs. The allowable maximum number of ID characters is 49. The ID number is not case sensitive. The default ID number is T.S.C.



## ● Keyboard BIOS Update

Press **ALT-SHIFT** to enter the update BIOS utility.

Users can upgrade the BIOS by copying the firmware through the specified serial port to the keyboard in the DOS environment.

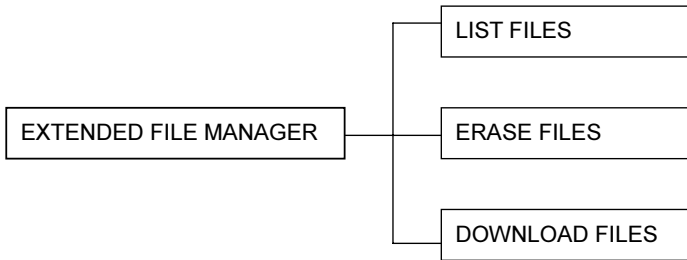
If the check sum of the BIOS is not correct after being upgraded, "UPDATE FAIL" will show on the LCD screen. Turn off the power and enter Upgrade mode again. If the upgrade process is successful, K012 will boot automatically.

## ● On-Line Editing Utility

The On-line Editing utility can be accessed by pressing **ALT-F2**. This feature is useful for modifying programs in the field. The available maximum editing characters are 79 characters x 250 lines. The  and  keys are used to scroll the files displayed on LCD screen. **Enter** is used to select the program for editing. **ALT-CLEAR** is used to delete the files stored in keyboard memory. **F1** used to upload the files through the specified serial port of keyboard to the connected device if the cursor is stopped at selected file. Select "New File" and press **Enter** to begin editing a new file. Permissible filenames are up to 8 characters with 3 characters for the extension. Press **F2** to save the file when editing is completed.

## ● Extended File Manager

Extended File Manager can be accessed by pressing **ALT-D**. Files stored in flash memory are read-only. File listing, deleting, and the download utility must be operated through Extended File Manager. The download and delete files process will be different from files stored in standard memory. The operations of Extended File Manager are described below:



### ■ LIST FILES:

All the files stored in Flash memory will be listed. The BASIC files stored in Flash memory will also be listed in the BASIC program file list by pressing **FORM**.

### ■ ERASE FILES:

This utility delete all the files stored in Flash memory. The “KILL” command can’t delete files stored in Flash memory.

### ■ DOWNLOAD FILES:

Enter this utility before downloading files to Flash memory. Download files either by DOS command or by K012 Windows software. Press the **EXIT** key to exit the download utility.

#### **Note:**

- 1. ALL files will be deleted when you invoke the DOWNLOAD FILES utility. Copy all the files again although only ONE file is appended.**
- 2. With password locked on, the above utilities will be disabled.**

- **Execute downloaded BASIC program**

Press **FORM** to list all the BASIC files stored in the K012 smart keyboard. The UP and Down arrow keys are used to scroll files displayed in LCD screen. K012 also supports a hot key feature to find the program. For example: If the program is named DEMO.BAS, press **FORM** and then press **D**. The K012 scrolls the files to the first one that begins with D automatically. Execute the selected program by pressing **ENTER**.



## Using K012

By the end of this section, you will be able to

1. Edit a TSKL program file on a PC.
2. Download the program file to the K012.
3. Execute the program.

## Edit A Program

To edit a program file, you need a plain text editor, for example: DOS Editor, Windows Notepad or the enclosed K012 Windows software.

**The following is an example of editing with a general text editor.**

1. Open a new text file.
2. Add the header at the first line of the file as listed below:

```
DOWNLOAD "MYPROG.BAS"
```

*Where "MYPROG.BAS" is the file name to be stored in keyboard memory.*

3. Edit the program contents.
4. Insert a line of command at the end of program as below:

```
EOP
```

5. Save this file.

The advantages of using enclosed K012 Windows utility software is that users don't have to add header and end of program commands to the program file.

## Download A Program

Users can download the program either by DOS commands or via the enclosed K012 Windows software.

Connect the PC and K012 together with the keyboard COM2 port.

Connect the K012 COM1 cable to the printer.

### ■ Download the program with DOS commands:

Enter the commands as below:

```
C:\> MODE COM2 96,N,8,1 (sets communications for COM2 of your PC)
```

```
C:\> COPY MYPROG.BAS /B COM2
```

Where MYPROG.BAS is the name of program file, COM2 is the serial port of your PC.

- **Download the program by K012 Windows software**

Save the program file.

Select *Utilities* | *Download File* to download the program to the keyboard.

**Note: 1. When downloading programs (\*.BAS), please select BAS file extension in the Download file dialog box.**

**2. The file name of downloaded file**

A sharp short beep means the file has been downloaded to keyboard memory.

## Execute A Program

Reset the K012 by pressing **ALT-EXIT**.

Press **FORM** to list program files.

Press **↑** and **↓** to select a program to execute.

If you want to interrupt the program, press **ALT-EXIT**.

## Variables, Constants, Functions, Operators, and Expressions

The information in this chapter will help you to learn about variables, constants, functions, operators, and expressions in TSKL (Taiwan Semiconductor keyboard Language).

Variables and constants are manipulated by operators to form expressions.

### Variables

Variables are placeholders used to store values. They have names and data types. The data type of a variable determines how the bits representing those values are stored in the computer's memory. When you declare a variable, you can also supply a data type for it. All variables have a data type that determines what kind of data they can store.

The variable name in TSKL can vary from one to ten characters. The first character must be a letter or an underscore with subsequent characters being letters, numbers, or underscore. There are two categories of variables in TSKL: numeric data type and string data type.

TSKL supplies several numeric data types: long integer, float and double float.

The maximum numbers of variable available in one program are listed below:

- 200 long integer and float data type variables
- 100 double float data type variables
- 50 string data type variables

**The range of data types in the system are listed below.**

Data Type	Identifier	Byte	Range	Significant digits
Long integer	%	4	-2147483648 to 2147483647	10
Float	none	4	-9999999 to 9999999	7
Double float	#	8	-9999999999999999 to 9999999999999999	15
String	\$	256	254 characters	N/A

### Long Integer Data Type Variables

The “%” identifier is used to declare a long integer variable by placing the “%” at the end of variable name. For example, A% and SUM% are integer variables.

### Float Data Type Variables

The default data type in TSKL is float data type. If no identifier is placed at the end of variable name, the variable will be treated as float data type in the system. For example, A and B are float variables. The precision of float data type is to 6 digits.

### Double Float Data Type Variables

The “#” identifier is used to declare a double float variable by placing the “#” at the end of variable name. For example, A# and SUM# are floating point variables. The precision of double float data type is to 15 digits.

### String Variables

The “\$” identifier is used to declare a string variable by placing the “\$” at the end of variable name. For example, A\$ and TITLE\$ are string variables. Each string variable can store 254 bytes of data.

### System Variables

System variables are the data maintained by the K012. For example, Real Time Clock.

All the system variables are listed below:

- YEAR
- MONTH
- DATE
- HOUR
- MINUTE
- SECOND
- IDNUMBER\$

## Functions

Functions are built-in procedures or subroutines used to evaluate, make calculations on, or transform data.

Functions used in TSKL can be grouped into numeric functions or string functions. For more information, please refer to **TSKL Functions**.

### Numeric Functions

Numeric functions include integral functions and floating point functions. For example, INT(), ASC().

### String Functions

String functions include string conversion and string manipulation. For example, RIGHT\$(), STR\$().

## Operators

The operators used in TSKL can be grouped into numeric operators, string operators and relational operators.

### Arithmetic Operators

Arithmetic operators: '+', '-', '\*', '/'.

### String Operators

String operators: '+'.  
(Note: The original text only lists '+', which is unusual for string operators in most programming languages. It may be a typo or specific to the TSKL environment.)

### Relational Operators

Relational operators: '>', '=', '<', '<>', '>=', '<='

## Expressions

Operators, constants, and variables are the constituents of expressions. An expression in TSKL is any valid combination of these pieces. There are three kinds of expressions in TSKL.

### Arithmetic Expressions

Arithmetic expressions can be integral expressions or floating point expressions, depending on the calculation value. In integral expressions, floating point operands will be converted to integer, and vice versa.

### String Expressions

There is only one operator in string expression, '+', that is, add a string to another string.

### Relational Expressions

The relational expressions are used to determine the relationship of one quantity to another. The result is true if the value is non-zero, otherwise, it is false.

## Commands and Statements

Commands instruct K012 to work accordingly. Sometimes, commands followed by one or several parameters, For example, INPUT A\$. For more information refer to **TSKL Commands**.

A statement is composed of one command or several commands, For example, IF...THEN...ELSE.

## TSKL Commands

### <ESC>!R

#### Description

This command is used to reset the keyboard. The keyboard will search for an AUTO.BAS auto-execution program as the first after reset.

#### Syntax

<ESC>!R

#### Remarks

<ESC> is ASCII 27 escape character.

#### Example

N/A



**<ESC>!W****Description**

This command is used to upgrade the firmware. The keyboard will enter BIOS upgrade mode after receiving this command and disable the execution of any AUTO.BAS program.

**Syntax**

<ESC>!W

**Remarks**

N/A

**Example**

N/A

## SOUND

### Description

Turn the speaker on at the specified frequency and interval.

### Syntax

SOUND frequency, interval

Where

Frequency: 0~15

Interval: 0~65535 msec,

### Remarks

N/A

### Example

```
FOR F=0 TO 15
  FOR I=0 TO 65535
    SOUND F,I
    FOR J=1 TO 10
      NEXT J
    NEXT I
  NEXT F
```

**CLS****Description**

Clears the LCD display.

**Syntax**

CLS

**Remarks**

CLS clears the LCD and places the cursor in the upper left corner. (At position 0,0)

**Example**

CLS

## **CLEAR**

### **Description**

Remove all the declared variables, arrays, and opened files in the program from memory.

### **Syntax**

CLEAR

### **Remarks**

N/A

### **Example**

CLEAR

## **CURSOR**

### **Description**

Selects cursor appearance.

### **Syntax**

CURSOR mode

### **Remarks**

The available modes for the cursor are listed below:

0: Hides the cursor.

1: Block with blinking cursor.

2: Fixed Underline cursor.

3: Block blinking and underline fixed cursor. (Default)

### **Example**

CURSOR 1

## LOCATE

### Description

Position cursor in LCD display.

### Syntax

LOCATE x, y

### Remarks

LOCATE moves the cursor to the given position in LCD display. If the coordinates are invalid, the command will be ignored.

The permissible value of the x parameter is from 0 to 79.

The permissible values of the y parameter are 0 and 1. Where 0 and 1 are the first and the second lines respectively of the LCD display.

### Example

LOCATE 1, 1

## PRINT...[USING]

### Description

The PRINT command is used to output the numeric data and strings to the LCD display. TSKL also supports formatted output to LCD display with the PRINT USING command.

The maximum number of specified formats for the PRINT USING command is 15.

### Syntax

PRINT list of expression [;]

Specify the numeric data output format:

```
PRINT USING "###.##",A
```

Specify the string data output format:

```
PRINT USING "\ \",A$
```

### Remarks

A blank line is displayed on the LCD if there is no expression after the PRINT command. The PRINT USING command sends (hex) 0D 0A (carriage return and line feed) at the end of the expression if no ";" (semicolon) is at the end of the expression. The "," (comma) is used to separate the values by 8 spaces. There are no spaces between the two values if the ";" (semicolon) is used between the two expressions. If ";" (semicolon) is the last character of PRINT statement, the cursor will stop in this line.

The "#" (pound sign) and "\" (back slash) are used to specify the output format of numeric data and string data respectively. If the actual value is greater than the specified format, a "%" (percent) sign will be placed at the end of value.

### Example

```
B=123.45
```

```
PRINT 12+3;ASC("A")
```

```
PRINT USING "###.##",B;
```

## INPUT

### Description

Input numeric data or strings from the keypad or K012 COM2 port and assign them to variables.

### Syntax

INPUT prompt; variables

### Remarks

Prompt must be a string constant. The variable list contains the variable names to be assigned.

The input value will pass to the variable if **ENTER** is pressed. Other input devices such as a bar code scanner or scale may be connected to the COM2 port of the K012 to get data instead of keying in data on the keyboard.

### Example

```
INPUT "ENTER ITEM NAME",ITEMS  
INPUT PRICE
```



## **OUT**

### **Description**

Output a data stream from the K012 serial port.

### **Syntax**

OUT port; list of expressions[:]

### **Remarks**

Port specifies the serial port to send data. It can be 0 (K012 COM1) or 1 (K012 COM2). The List of Expressions consists of string or numeric expressions separated by semicolons. The OUT command sends 0D0A (carriage return and line feed) at the end of the expression except if the semicolon is used as the terminating character. Comma (",") also sends 0D0A to the serial port.

### **Example**

OUT 0;A;A\$

## FOUT

### Description

Output a specified file from the K012 serial port.

### Syntax

FOUT Port, FileHandle, FileSize

### Remarks

Port specifies the serial port to send data. It can be 0 (K012 COM1) or 1 (K012 COM2).

The number of FileHandle is between 0 and 14. FileSize is expressed in bytes.

### Example

```
OPEN "DATA1" FOR INPUT AS #1
```

```
FOUT 0,1,1024
```

## **OUT USING**

### **Description**

Output a formatted data stream from the K012 serial port.

### **Syntax**

OUT port USING "format",list of expressions[:]

### **Remarks**

Port specifies the K012 serial port to send data. It can be 0 (K012 COM1) or (K012 COM2) port.

List of expressions consists of string or numeric expressions separated by semicolons. The OUT USING command sends 0D0A (carriage return and line feed) at the end of the expression except if the semicolon (";") is used as a terminator. Comma (",") also sends 0D0A to the serial port.

### **Example**

```
OUT 0 USING "####.##" ,A  
OUT 1 USING "\    \",A$
```

## DOWNLOAD

### Description

The “DOWNLOAD” keyword is the identifier to save files into K012 memory. The two types of file that can be downloaded into keyboard memory are program files and data files.

### Syntax

**Program file:**

```
DOWNLOAD “FILENAME.BAS”
```

```
    File contents...
```

```
EOP
```

**Data file:**

```
DOWNLOAD “FILENAME”, FILESIZE,<DATA FILE>
```

### Remarks

Filename may be up to 8 characters with a 3-character extension.

The extension for program files must be BAS.

Data files can be any format of file. 0D 0A is used as separator of each data for a text data file.

The FILESIZE parameter is expressed in bytes.

### Example

**Program file:**

```
DOWNLOAD “DEMO.BAS”
```

```
    CLS
```

```
    PRINT “This is a test”
```

```
EOP
```

**Data file:**

```
DOWNLOAD “DEMO.DAT”,10,0123456789
```

## EOP

### Description

End of program. This keyword must be placed at the end of a program file if the DOWNLOAD keyword is used in the program.

### Syntax

EOP

### Remarks

N/A

### Example

**Program file:**

```
DOWNLOAD "DEMO.BAS"  
CLS  
PRINT "This is a test"  
EOP
```

## OPEN

### Description

To establish file handles for file access. Up to 15 files can be accessed in one program.

### Syntax

OPEN "filename" FOR mode AS #FileHandle.

### Remarks

Filename is the name of the file.

Mode specifies the file operation mode. It can be:

INPUT: Position to the beginning of the file and this file is "read only". If the file does not exist, the "File not found" error is displayed on LCD display.

OUTPUT: Position to the beginning of the file, and this file is "write only". If the file does not exist, a new file is created.

APPEND: Append characters to the end of file.

FileHandle is a constant number or expression result, between 0 and 14.

**Remember to close the file handle when the file is no longer used.**

### Example

```
OPEN "DATA1" FOR INPUT AS #1
```

```
OPEN "DATA2" FOR OUTPUT AS #2
```

**CLOSE****Description**

To clear file handles.

**Syntax**

CLOSE #FileHandle

**Remarks**

FileHandle must be a constant number, with which the file is opened.

**Example**

CLOSE #1

## KILL

### Description

To delete file(s) in keyboard memory (SRAM only).

### Syntax

KILL "filename"

KILL "\*.\*"

KILL "\*\*"

### Remarks

Filename can be any file in the memory.

Wild card (\*) supports all files.

The file must be closed before deleting.

### Example

KILL "DEMO.BAS"

**Note:** KILL cannot be used to delete files in Flash memory. For this you must use the K012 Extended File Manager utility from the keyboard.



## SEEK

### Description

Reposition a file pointer in a specified file buffer.

### Syntax

SEEK #FileHandle, offset.

### Remarks

Offset is the number from the beginning of file to the new position.

### Example

```
OPEN "DATA" FOR INPUT AS #1  
SEEK #1,12
```

## READ

### Description

Read data from a data file and assign them to variables.

### Syntax

READ #FileHandle; list of variables.

### Remarks

FileHandle specifies the file to read data from.

The variables store the data read from the FileHandle.

### Example

```
OPEN "DATA" FOR INPUT AS #1  
READ #1;A;A$
```

## WRITE

### Description

Write data to an opened file.

### Syntax

WRITE #FileHandle; list of expressions.

### Remarks

FileHandle specifies the file to write data to.

The variables are used to write data to opened data file.

### Example

```
OPEN "DATA2" FOR OUTPUT AS #1  
WRITE #1; "DATA"
```

## GOSUB...RETURN

### Description

To branch to and return from a subroutine.

### Syntax

```
GOSUB label
    Statement block1
Label:
    Statement block2
RETURN
```

### Remarks

Label is a tag to mark a specified position in the program. The available maximum label name is 20 characters. A return statement will cause the program return to the statement following the GOSUB statement.

The total number of GOSUB...RETURN statements cannot exceed 40 in one program.

### Example

```
PRINT "MAIN ROUTINE"
GOSUB SUB1
    PRINT "MAIN ROUTINE"
END
SUB1:
    PRINT "SUBROUTINE"
RETURN
```

The execution result should be as follows:

```
OK
MAIN ROUTINE
SUBROUTINE
MAIN ROUTINE
```

## Loop Statements

Loop statements allow a program to execute one or more lines of code repetitively.

The loop statements that TSKL supports include:

- **DO...LOOP**
- **FOR...NEXT**
- **WHILE...WEND**

## DO...[EXITDO]...LOOP

### Description

Use a **DO** loop to execute a block of statements an indefinite number of times. There are several variations of **DO...LOOP** statement, but each evaluates a numeric condition to determine whether or not to continue execution.

### Syntax

```
DO {WHILE | LOOP}
    Statements
    {EXITDO}
LOOP
```

*Or*

```
DO
    Statements
    {EXITDO}
LOOP {WHILE | LOOP}
```

*Or*

```
DO
    Statements
    {EXITDO}
LOOP
```

## Remarks

The total number of DO...LOOP statement in one program cannot exceed 40.

The maximum number of IF...THEN...ELSE, FOR...NEXT, WHILE...WEND, and DO...LOOP statements available in one program is 40.

## Example

```
A=1
DO WHILE A<10
  A=A+1
LOOP
PRINT A
```

## FOR...[EXITFOR]...NEXT

### Description

Executes a series of instructions by a specified number of times in a loop.  
(As opposed to DO ...LOOP which executes until a condition is met)

### Syntax

```
FOR variable= I TO J [STEP K]
    Statements
NEXT variable
```

### Remarks

I, J, K are numeric expressions

I: the initial value of the counter

J: the final value of the counter

K: The increment of the counter. If the K parameter is ignored, the default increment is 1.

The maximum numbers of IF...THEN...ELSE, FOR...NEXT, WHILE...WEND, and DO...LOOP statements available in one program is 40.

### Example

The following sample program prints out the sum of numbers between 1 and 10:

```
SUM=0
FOR I=1 TO 10
SUM=SUM+I
NEXT I
PRINT SUM
```

## WHILE...WEND

### Description

To execute a series of statements in a loop until the given condition is **false**.

### Syntax

```
WHILE expression
    statements
WEND
```

### Remarks

If the expression is true the program will be executed until the WEND statement is encountered, then return to the WHILE statement to check again. After encountering a false condition, the program will branch to the statement following the WEND.

The total numbers of WHILE...WEND statement in one program can not exceed 40.

The maximum numbers of IF...THEN...ELSE, FOR...NEXT, WHILE...WEND, and DO...LOOP available in one program is up to 40.

### Example

```
A=10
WHILE A
    A=A-1
    PRINT "12345678"
WEND
```



## GOTO

### Description

Branches from the program to a specified block of statements.

### Syntax

GOTO label

### Remarks

Label is a tag to mark a specified position in the program.

The Label name is limited to a maximum of 20 characters.

The total number of GOTO statement in a program cannot exceed 200.

### Example

```
INIT:
SUM=0
INDEX=1
INPUT "S/N:"; SN$(INDEX)
IF SN$="" THEN GOTO INIT
```

## Program-Control Statements

The program-control statements are the essence of any computer language because they govern the flow of program execution. Program-control statements may be separated into two categories:

- IF...THEN
- IF...THEN...ELSE

### IF...THEN...ELSE

#### Description

Use an **IF...THEN** block to execute one or more statements conditionally. You can use either a single-line syntax or multiple-line “block” syntax:

#### Syntax

IF condition THEN statement

***Notice that the single-line form of IF...THEN does not use an ENDIF statement.***

*Or*

```
IF condition THEN
    Statements
ENDIF
```

*Or*

```
IF condition THEN
    Statements
ELSE
    Statements
ENDIF
```

*Or*

```
IF condition1 THEN
    Statement block 1
ELSEIF condition2 THEN
    Statement block 2
ELSE
    Statement block n
ENDIF
```

### **Remarks**

If the result of the expression is nonzero, the statement following THEN will be executed. If the result of the expression is zero, and a statement following the ELSE is present, it will be executed. Otherwise the next line of statement is executed.

If there are block of statements in IF...THEN ...ELSE, ENDIF must be used at the end of the IF...THEN...ELSE statement.

Limitations:

The total number of IF...THEN...ELSE statements in a program can not exceed 40.

The total numbers of IF...THEN...ELSE, FOR...NEXT, WHILE...WEND, and DO...LOOP in a program cannot exceed 40.

### **Example**

```
IF A>0 THEN PRINT "TRUE" ELSE PRINT "FALSE"
IF ANS=1 THEN
    CLS
    PRINT "Error!!!"
    INPUT ANS
ENDIF
```

## END

### Description

Terminates the program execution.

### Syntax

END

### Remarks

END statement may be placed anywhere in a program to terminate the execution.

With the END statement, all variables will be released from memory and file handles closed.

### Example

END

## DIM

### Description

An array is a collection of variables of the same type, referenced by a common name. The DIM statement is used to declare the array variables of integer, float and double float data types. The lowest address corresponds to the first element, and the highest address to the last element. A specific element in an array is accessed by an index.

### Syntax

DIM variable (subscripts)[, variable (subscripts), ...]

### Remarks

The base of an array index always begins from 0. For example, **DIM A(10)** there are, in total, 11 elements of variable A (0 to 10).

Do not declare a duplicate array in the program without executing the FREE statement.

The DIM statement sets all the elements of specified arrays to the following initial values: numbers to zero and strings to null string.

The total number of array elements (no matter what kind of data type) cannot exceed than 200 elements.

The maximum array variable name is 10 characters.

The maximum dimensions of an array variable is 3.

### Example

```
DIM A(20)
```

## FREE

### Description

Eliminates array variables from memory.

### Syntax

FREE dimension variable

### Remarks

Arrays can be re-dimensioned after they are freed, or the memory space previously allocated to the array may be used for other purposes. If an attempt is made to re-dimension an array without first freeing it, an error occurs.

### Example

```
FREE A
```

**REM****Description**

Inserts explanatory remarks in a program.

**Syntax**

REM comments

**Remarks**

REM statements are not executed.

**Example**

```
REM *** this is an example ***
```

## TSKL Functions

### POS()

#### Description

Gets the current position of the cursor in the LCD display.

#### Syntax

POS()

#### Remarks

This function returns the value between 0~79.

#### Example

```
Cur_pos = POS()
```



**INKEY()****Description**

To return the ASCII code of the character read from keyboard. The returned ASCII code will not show on LCD display.

**Syntax**

```
INKEY()
```

**Remarks**

All tasks are pending until a key is pressed.

**Example**

```
A=INKEY()
```

## INP\$()

### Description

To return one byte received from a serial port.

### Syntax

INP\$(expression)

### Remarks

The result of the expression must be numeric. (0 or 1)

0: COM1 (Serial port with cable on K012)

1: COM2 (Serial port mount on K012)

The returned value is a string

### Example

```
A$=A$ + INP$(0)
```

## EOF()

### Description

To return nonzero when the end of a file has been reached, or to return 0 if the end of file (EOF) has not been found.

### Syntax

EOF(file number)

### Remarks

If file pointer points to the end of the file, EOF returns non-zero.

### Example

```
OPEN "DATA" FOR INPUT AS #1
C=0
Repeat:
IF EOF(1)>0 THEN GOTO end_of_file
READ #1;A
C=C+1
GOTO Repeat
End_of_file:
END
```

## LOF()

### Description

To return the size of a file.

### Syntax

LOF(FileHandle)

### Remarks

An integer is returned to indicate the size of file

### Example

```
File_length = LOF(1)
```

**ABS()****Description**

To return the absolute value of the expression.

**Syntax**

ABS(expression)

**Remarks**

The result of the expression must be numeric.

**Example**

X=ABS(5-12)

## ASC()

### Description

To return the value of the ASCII code for the first character of the expression.

### Syntax

ASC(expression)

### Remarks

The result of the expression must be a string.

### Example

X=ASC("A")

**LEN()****Description**

To return the length of the string.

**Syntax**

LEN(string expression)

**Remarks**

The maximum returned string length is 255.

**Example**

X=LEN("A")

## **FRE()**

### **Description**

Return the size of free memory.

### **Syntax**

FRE( )

### **Remarks**

Calculates and returns an integer to indicate free memory size of the K012 in K Bytes.

### **Example**

```
A=FRE()  
PRINT A
```



## INT()

### Description

To truncate an expression to an integral number.

### Syntax

INT(expression)

### Remarks

The result of the expression must be numeric.

### Example

A=INT(5/2)

## VAL()

### Description

To return the numerical value of the string expression.

### Syntax

VAL(expression)

### Remarks

The result of the expression must be a string.

### Example

```
A=VAL("1234")
```

**RND()****Description**

RND returns a number between 0 and 1

**Syntax**

A=RND()

**Remarks**

N/A

**Example**

A=RND()

## CHR\$()

### Description

To return the ASCII character of the numerical expression.

### Syntax

CHR\$(expression)

### Remarks

The result of the expression must be numeric.

### Example

A\$=CHR\$(27)

Used, for example to send an ESCAPE or other control code to a printer.

## FREAD\$()

### Description

To read a number of bytes from specified file.

### Syntax

FREAD\$ (FileHandle, count)

### Remarks

Count is the number of data bytes. Maximum count size is 255.

### Example

```
OPEN "DATA" FOR INPUT AS #1
A$=FREAD$(1,20)
CLOSE #1
END
```

## LEFT\$()

### Description

To return a number of the leftmost characters of the string expression.

### Syntax

LEFT\$(expression, count)

### Remarks

The result of the expression must be a string.

Count is the number of the leftmost characters.

### Example

```
10 A$=LEFT$("123456",3)
```

## RIGHT\$( )

### Description

To return a number of the rightmost characters of the string expression.

### Syntax

RIGHT\$(expression, count)

### Remarks

The result of the expression must be a string.

Count is the number of the rightmost characters.

### Example

A\$=RIGHT\$("123456",3)

## MID\$()

### Description

To return a number of characters beginning from the starting parameter to the specified length.

### Syntax

MID\$(String, Starting, Length)

### Remarks

String: Can be a string or expression

Start: Character position in string at which the part to be taken begins. If start is greater than the number of characters in string, Mid returns a zero-length string ("").

Length: The numbers of characters to be returned.

### Example

```
A$=MID$("123456",3,3)
```



## STR\$()

### Description

Returns a string representation of a number.

### Syntax

STR\$(expression)

### Remarks

The result of the expression must be a numeric.

**Note:** *STR\$() function cannot be included in other functions.*

**Example:** VAL(STR\$("123"))

### Example

A\$=STR\$(12345)

**SPC\$()****Description**

To insert a number of spaces in a string.

**Syntax**

SPC\$(expression)

**Remarks**

The result of the expression must be numeric.

**Example**

```
PRINT "PRICE";SPC$(5);"123.5"
```

## System Variables

### YEAR

#### Description

Sets and returns the year of the Real Time Clock. The YEAR variable may be set from 00 to 99. The returned data is a 4-digit number.

Ex: 1998,2001.

#### Syntax

YEAR=nn

Where

nn=00~99

A=YEAR

#### Remarks

The Real Time Clock is Year 2000 compatible.

#### Example

```
YEAR$=STR$(YEAR)
```

```
YEAR$=MID$(YEAR$,3,2)
```

## MONTH

### Description

Sets and returns the month of the Real Time Clock. Allowable numbers are 01 to 12.

### Syntax

MONTH=nn

Where

nn=01~12

A=MONTH

### Remarks

N/A

### Example

MONTH=12

**DATE (day)****Description**

Sets and returns the date of the Real Time Clock. Allowable numbers are from 00 to 99.

**Syntax**

DATE=nn

Where

nn=01~31

A=DATE

**Remarks**

The leap year timing is automatic

**Example**

DATE=01

## HOUR

### Description

Sets and returns the hour of the Real Time Clock. Allowable numbers are from 00 to 23.

### Syntax

HOUR=nn

Where

nn=00~23

A=HOUR

### Remarks

N/A

### Example

HOUR=01

## MINUTE

### Description

Sets and returns the minute of the Real Time Clock. Allowable numbers are from 00 to 59.

### Syntax

MINUTE=nn

Where

nn=00~59

A=MINUTE

### Remarks

N/A

### Example

MINUTE=16

## SECOND

### Description

Sets and returns the second of the Real Time Clock. Allowable numbers are from 00 to 59.

### Syntax

SECOND=nn

Where

nn=00~59

A=SECOND

### Remarks

N/A

### Example

MINUTE=16



## IDNUMBER\$

### Description

IDNUMBER\$ is read only at run time. The ID number can be set in the Setup Utility. With password locked ON, users cannot change the ID number. The factory default is: T.S.C.

### Syntax

A\$=IDNUMBER\$

### Remarks

N/A

### Example

```
IF IDNUMBER="T.S.C." THEN
  GOSUB USER1
ENDIF
```

## Appendix A - Limitations of TSKL

- The total numbers of IF...THEN...ELSE, FOR...NEXT, WHILE...WEND and DO...LOOP in a program cannot exceed 40.
- The maximum numbers of GOSUB...RETURN cannot exceed 40 layers.
- The number of labels can't exceed 200.
- 200 long integer and float data type variables
- 100 double float data type variables
- 50 string data type variables
- The maximum number of characters for a variable name is 10.
- The maximum number of characters for a label name is 20.
- The available memory for download is 128 KB (SRAM).
- The number of downloaded files cannot exceed 50 files.
- 15 files can be opened simultaneously in one program.
- Maximum numbers of dimensions for array variables is 3 dimensions.
- The total number of specified formats for PRINT USING and OUT USING statements can't exceed 15.

## Appendix B Error Messages

**(01) Syntax error**

The statement syntax is not correct.

**(02) IF without THEN**

If statement without THEN keyword.

**(03) IF without ENDIF**

If statement without ENDIF keyword.

**(04) ELSEIF without IF**

IF...THEN...ELSE...ELSEIF statement without ELSEIF keyword.

**(05) ELSE without IF**

IF...THEN...ELSE statement without IF keyword.

**(06) ENDIF without IF**

IF...THEN...ELSE statement without IF keyword.

**(07) FOR without NEXT**

FOR...NEXT statement without NEXT keyword.

**(08) NEXT without FOR**

FOR...NEXT statement without FOR keyword.

**(09) EXITFOR without FOR**

FOR...NEXT statement without FOR keyword.

**(10) WHILE without WEND**

WHILE...WEND statement without WEND keyword.

**(11) WEND without WHILE**

WHILE...WEND statement without WHILE keyword.

**(12) DO without LOOP**

DO...LOOP statement without LOOP keyword.

**(13) LOOP without DO**

DO...LOOP statement without DO keyword.

**(14) RETURN without GOSUB**

GOSUB...RETURN without GOSUB keyword

**(15) Mode error**

File operation access error.

**(16) File number error**

Illegal file handle number.

**(17) File not found**

The access file name does not exist in memory.

**(18) Too many labels**

The total numbers of label in the program is exceed than 200.

**(19) Duplicate label**

Declared twice of label name.

**(20) Label not found**

The specified label does not exist in program.

**(21) Illegal function call**

Function does not exist.

**(22) Illegal variable**

The variable name exists illegal letters.

**(23) Variable not defined**

The variable does not assigned value in the program.

**(24) Array not defined**

The accessed array does not exist.

**(25) Out of string space**

The string is longer than 254 bytes.

**(26) Too many variables**

The number of variables is larger than the definition.

**(27) Too many formats**

The numbers of specified format of PRINT USING command and OUT USING command is larger than 15.

**(28) Type mismatch**

The data types of the two variables or returned values are not the same.

**(29) Port error**

The port specified in OUT and FOUT command is other than 0 and 1.

**(30) Stack overflow**

The total numbers of GOSUB...RETURN, WHILE...WEND, DO...LOOP, FOR...NEXT, IF...THEN...ELSE statement is larger than the specified number.

**(31) Division by zero**

The dividend is zero.

**(32) Unknown operator**

The operator is not defined in K012

**(33) Expression too complex**

Arithmetic expression is too complex.

**(35) Duplicate array**

Duplicate declare array variable.

**(36) Out of memory**

There is not enough memory for download program or data file.

**(37) Too many files**

The total number of files stored in memory exceeds that 50.

**(38) Unmatched brace**

One of the right or left parentheses is missing.

## Appendix C - RS-232 Pin Configurations

Host and Printer RS-232 PIN Definition:

### A. RS-232 Serial Interface, 9-Pin D-Type connector pin definitions

Host			Device	Printer		
IBM PC	K012	K012	K012	Axiohm	TSC CLEVER	
COM1	COM1 (Note 1)	COM1 (Note 2)	COM2	COM2	COM2	
<b>Pin 1</b>	/CD ←	+ 5V ←	+ 5V ←	+ 5V ←	+ 5V →	+ 5V →
<b>Pin 2</b>	RXD ←	RXD ←	TXD →	TXD →	RXD ←	TXD →
<b>Pin 3</b>	TXD →	TXD →	RXD ←	RXD ←	TXD →	RXD ←
<b>Pin 4</b>	/DTR →	/DTR →	/DSR ←	/DSR ←	NC	/DSR ←
<b>Pin 5</b>	GND	GND	GND	GND	GND	GND
<b>Pin 6</b>	/DSR ←	/DSR ←	/DTR →	/DTR →	NC	RDY →
<b>Pin 7</b>	/RTS →	/RTS →	/CTS ←	/CTS ←	/RTS →	NC
<b>Pin 8</b>	/CTS ←	/CTS ←	/RTS →	/RTS →	/CTS ←	RDY →
<b>Pin 9</b>	RI ←	+ 5V ←	+ 5V ←	+ 5V ←	+ 5V →	+ 5V →

Note 1: K012 Hardware SW1 & SW2 must ON.

Note 2: K012 Hardware SW1 & SW2 must OFF.

Note 3: <-- symbol emblem INPUT,  
--> symbol emblem OUTPUT.

B. RS-232 Serial Interface 25-Pin D-Type connector definition

RS-232C	Host			Printer
	IBM PC	PC-8801	ProTool+	Zebra
			(Note 1)	
1 Protective Ground	SHELL	GND	Frame GND	Frame GND
2 Transmitted Data	TXD →	TXD	RXD ←	TXD →
3 Received Data	RXD ←	RXD←	TXD →	RXD ←
4 Request to Send	/RTS →	RTS→	/CTS ←	/RTS →
5 Clear to Send	/CTS ←	/CTS ←	/RTS →	/CTS ←
6 Data Set Ready	/DSR ←	/DSR←	/DTR →	/DSR ←
7 Signal Ground	GND	GND	GND	GND
8 Received Line Signal Detector	/CD ←	DCD	NC	NC
9 (Reserved for Data Set Testing)	NC	NC	+ 5V 0.25A ←	5V 1A →
10 (Reserved for Data Set Testing)	NC	NC	+ 5V 0.25A ←	NC
11 Unassigned	NC	NC	NC	NC
12 Secondary Carrier Detect	NC	NC	NC	NC
13 Secondary Clear to Send	NC	NC	NC	NC
14 Secondary Transmitted Data	NC	NC	+ 5V 0.25A ←	NC
15 Transmission Signal Element Timing	NC	NC	NC	NC
16 Secondary Received Data	NC	NC	NC	NC
17 Receiver Signal Element Timing	NC	RXC←	NC	NC
18 Unassigned	NC	NC	NC	NC
19 Secondary Request to Send	NC	NC	NC	NC
20 Data Terminal Ready	/DTR →	/DTR→	/DSR ←	/DTR →
21 Signal Quality Detector	NC	NC	NC	NC
22 Ring Indicator	RI ←	NC	NC	NC
23 Data Signal Rate Selector	NC	NC	+ 5V 0.25A ←	NC
24 Transmitter Signal Element Timing	NC	TXC→	NC	NC
25 Unassigned	NC	NC	NC	NC

(Continued on next page)

	DATAMAX	Ring	TEC	SATO
1 Protective Ground	CHASSIS	Frame GND	Frame GND	Frame GND
2 Transmitted Data	TXD →	TXD →	RXD ←	TXD →
3 Received Data	RXD ←	RXD ←	TXD →	RXD ←
4 Request to Send	/RTS →	/RTS →	/CTS ←	/RTS →
5 Clear to Send	/CTS ←	/CTS ←	/RTS →	/CTS ←
6 Data Set Ready	NC	/DSR ←	/DTR →	/DSR ←
7 Signal Ground	GND	GND	GND	GND
8 Received Line Signal Detector	NC	NC	NC	NC
9 (Reserved for Data Set Testing)	NC	NC	NC	NC
10 (Reserved for Data Set Testing)	NC	+ 5V	NC	NC
11 (Unassigned)	NC	NC	NC	NC
12 Secondary Carrier Detect	NC	NC	NC	NC
13 Secondary Clear to Send	NC	To Be Left Unused	NC	NC
14 Secondary Transmitted Data	+ 5V 0.1A →	To Be Left Unused	NC	NC
15 Tx Signal Element Timing	NC	NC	NC	NC
16 Secondary Received Data	NC	NC	NC	NC
17 Receiver Signal Element Timing	NC	NC	NC	NC
18 Unassigned	NC	NC	NC	NC
19 Secondary Request to Send	NC	To Be Left Unused	NC	NC
20 Data Terminal Ready	BUSY →	/DTR →	/DSR ←	/DTR →
21 Signal Quality Detector	NC	NC	NC	NC
22 Ring Indicator	NC	NC	NC	NC
23 Data Signal Rate Selector	NC	+ 5V	NC	NC
24 Tx Signal Element Timing	NC	NC	NC	NC
25 Unassigned	NC	To Be Left Unused	NC	NC



## Operation Guide

### Auto Execution Utility

Power on	Execute the "AUTO.BAS" program automatically
----------	----------------------------------------------

### Power-on Utility

Power on with <b>ALT F1</b>	Disable the auto execution utility
Power on with <b>ALT SHIFT</b>	Upgrade keyboard BIOS
Power on with <b>ALT CLEAR</b>	Keyboard initialization
Power on with <b>ALT D</b>	Extended file manager utility

### Keyboard Utilities

<b>FORM</b>	Select BASIC files to execute
<b>ENTER</b>	Execute the selected item
<b>EXIT</b>	Exit sub-menu
← →	Select parameters
↑ ↓	Scroll files or cursor
<b>ALT F1</b>	Keyboard configuration setup
<b>ALT F2</b>	On-line editing utility
<b>ALT F2 F1</b>	Upload files
<b>ALT F2 CLEAR</b>	Delete files
<b>ALT D</b>	Extended file manager
<b>ALT EXIT</b>	Reset keyboard



**INDEX**

<ESC>IR .....	23	Extended file manager .....	14
<ESC>IW .....	24	Features of K012 .....	7
ABS() .....	61	Float Data Type Variables .....	19
Appendix A Limitations of TSKL .....	82	FOR...[EXITFOR]...NEXT .....	47
Appendix B Error Messages .....	83	FOUT .....	33
Appendix C RS-232 Pin Configurations .....	86	FRE() .....	64
Arithmetic Expressions .....	21	FREAD\$() .....	69
Arithmetic Operators .....	20	FREE .....	54
ASC() .....	62	Functions .....	20
Auto Execution .....	10	GOSUB...RETURN .....	44
Check-List .....	9	GOTO .....	49
CHR\$() .....	68	HOUR .....	78
CLEAR .....	27	IDNUMBER\$ .....	81
CLOSE .....	39	IF...THEN...ELSE .....	50
CLS .....	26	INKEY() .....	57
Commands and Statements .....	22	INP\$() .....	58
Communication Interface .....	8	INPUT .....	31
CURSOR .....	28	INT() .....	65
DATE .....	77	Keyboard BIOS Update .....	13
DIM .....	53	Keyboard Configuration Setup .....	11
DO...[EXITDO]...LOOP .....	45	Keyboard Initialization .....	10
Document Conventions .....	5	Keyboard Operation .....	10
Double Float Data Type Variables ..	19	Keyboard Setup .....	10
DOWNLOAD .....	36	KILL .....	40
Download A Program .....	16	LEFT\$() .....	70
Edit A Program .....	16	LEN() .....	63
END .....	52	LOCATE .....	29
EOF() .....	59	LOF() .....	60
EOP .....	37	Long Integer Data Type Variables ..	18
Execute A Program .....	17	Loop Statements .....	45
Execute downloaded BASIC program14		MID\$() .....	72
Expressions .....	21	MINUTE .....	79
		MONTH .....	76

---

Numeric Functions .....	20	SEEK.....	41
On-Line Editing Utility.....	13	SOUND .....	25
OPEN.....	38	SPC\$().....	74
Operation Guide.....	89	Specifications.....	7
Operators .....	20	STR\$().....	73
Options.....	9	String Expressions.....	21
OUT.....	32	String Functions.....	20
OUT USING .....	35	String Operators .....	20
POS() .....	56	String Variables.....	19
Power-on Utilities .....	10	System Variables.....	19, 75
PRINT...[USING] .....	30	TSKL Commands .....	23
Program-Control Statements.....	50	TSKL Functions .....	56
Quick Start.....	6	Using K012.....	16
READ .....	42	VAL().....	66
Relational Expressions.....	21	Variables.....	18
Relational Operators .....	20	Variables, Constants, Functions, Operators, and Expressions .....	18
REM .....	55	WHILE...WEND.....	48
RIGHT\$().....	71	WRITE.....	43
RND() .....	67	YEAR.....	75
Safety Regulation.....	9		
SECOND.....	80		

---

---