ZEBEX INDUSTRIES INC.

Data Collector Series

WinTaskGen (Version 1.50)

User's Manual

ZEBEX INDUSTRIES INC.

WWW.ZEBEX.COM

Revision History

Changes to the original manual are listed below:

Version	Date	Description of Version
1.0	September. 20, 2010	Initial release
1.5	August. 26, 2011	1. Add changing (UI)Language 2. Add the Timestamp LONGTIME, SHORTTIME, and Format Time in the Input Mode with the Free Task Editor 3. Add Show Macro Reference in the Macro Editor with the Procedure Editor
1.6	Nov.28,2011	Indicates the start point of the loop of collecting data in free task

Important Notice

No warranty of any kind is made in regard to this material, including, but not limited to, implied warranties of merchantability or fitness for a particular purpose. We are not liable for any errors contained herein or incidental or consequential damages in connection with furnishing, performance or use of this material.

We reserve the right to make changes in product design without reservation and without notification. The material in this guide is for information only and is subject to change without notice.

Guidance for Printing
This manual is in A5 size. Please double check your printer setting before printing it.

Table of Contents

Introduction

WinTaskGen is a powerful software platform for supporting Data Collector series products. It is under an open system frame and provides great possibilities for extension and development.

The new 5.0 version has a redesigned user interface that makes WinTaskGen more user-friendly. If you have used WinTaskGen in the past, you will immediately notice that the menu has been completely refactored and now features three distinct areas: Procedure Task Editor, Free Task Editor, and Remote Settings. By combining related Tasks into each of these panels, using WinTaskGen becomes faster and more intuitive.

To create programs to exactly fit your work, simply select the commands from the menu and link the Tasks to the databases you desire. You can use the "Procedure Task Editor" to create sequence of programs (Procedures) for your inventory needs or use the "Free Task Editor" to create simple commands for data collecting purpose. Then use WinTaskGen's communication functions to download Tasks to the Data Collector unit.

In addition, the "Remote Settings" provides a quick, convenient means to configure your data collectors and the barcode settings.

The software supports different baud rates for up- and downloading data. The highest supported baud rate is 115200 bps. When uploading data to a PC, you can choose whether to delete data from the Data Collector unit or not. Also you can add a variety of different separator symbols into the uploaded data.

Features

- Great ability to develop and extend Tasks under an open system frame.
- Supports Windows XP, Vista, and 7.
- Supports real-time device and barcode settings.
- Supports pop-up dialogs for creating command Procedures and Macros.
- Supports numerical operations such as +, -, *, / and compare operations.
- Supports: Uploading data forms, Downloading FreeTask and Task linking (compiling).

System Requirement

WinTaskGen requires a PC with the following specifications:

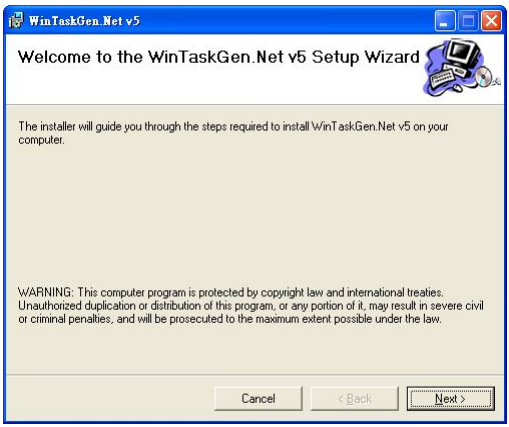
- Windows XP Service Pack 2 or above
- Microsoft Internet Explorer 5.01 or later
- Microsoft Framework 2.0
- Disk Space Requirements: 280 MB (x86), 610 MB (x64)

Installation

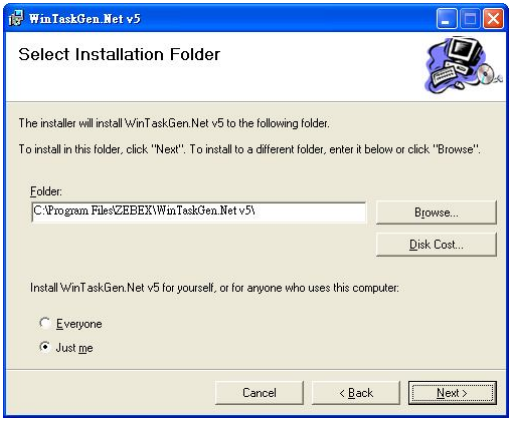
Follow the instructions below to install WinTaskGen on your computer.

1. Find and click the "Setup" file on your computer to begin installation.

2. Follow the on screen instructions and click "Next" when prompted to continue installation.

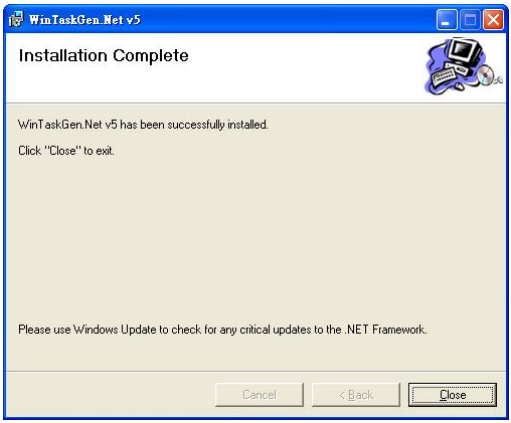


3. Click ”Browse” to select the location your want to install the program in and click “Next.”



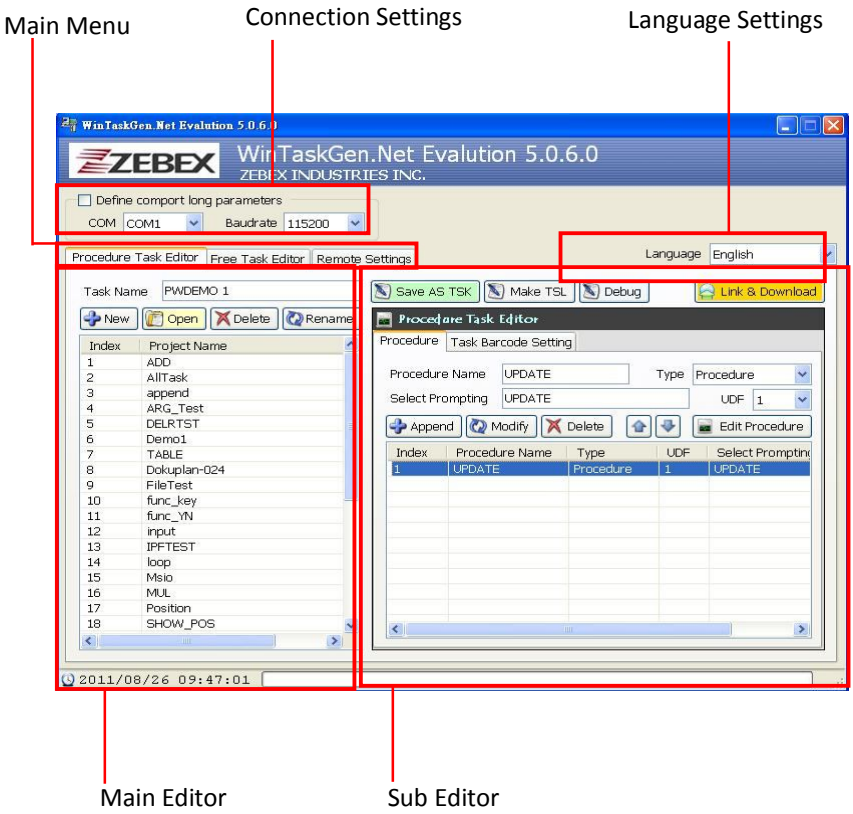
4. Continue to follow the on screen instructions and click “Next” to continue installation.

5. Click “Close” when the installation is done.



Overview

Refer to the following illustrations to familiarize yourself with the user interface.



Description	Function
Main Menu	Used to access all the functions in the program
Connection Settings	Device settings used for connection
Language Settings	Used to set UI language
Main Editor	Used to manage or view Tasks or device information
Sub Editor	Used to manage Procedure or Free Tasks or device settings and to download or save Tasks

Getting Started

Read this section to learn how to start using WinTaskGen. This section covers basic functions such as writing a simple Procedure and configuring device settings.

See later chapters for advanced functions.

Starting WinTaskGen

To start WinTaskGen:

1. From Windows "Start Menu", "Programs", then select "WinTaskGen."

2. The WinTaskGen main screen appears on the desktop.

Changing Language

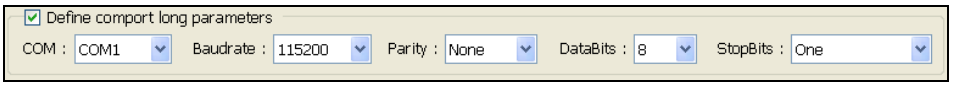
Default is English. In the Language folder of the WinTaskGen installed, users can create their languages - .LNG files, Japanese.LNG for instance. And then user can choose and change different language in the above Combo box.

About the format of the language file, user can refer to the English.LNG format.

Connecting Data Collector

To connect a data collector:

1. Check "Define comport long parameters" checkbox on top if necessary. Extra options appear as shown below when you check this option.



2. Select the options from the dropdown menu. When you define communication protocols, you must set the same value with the device.

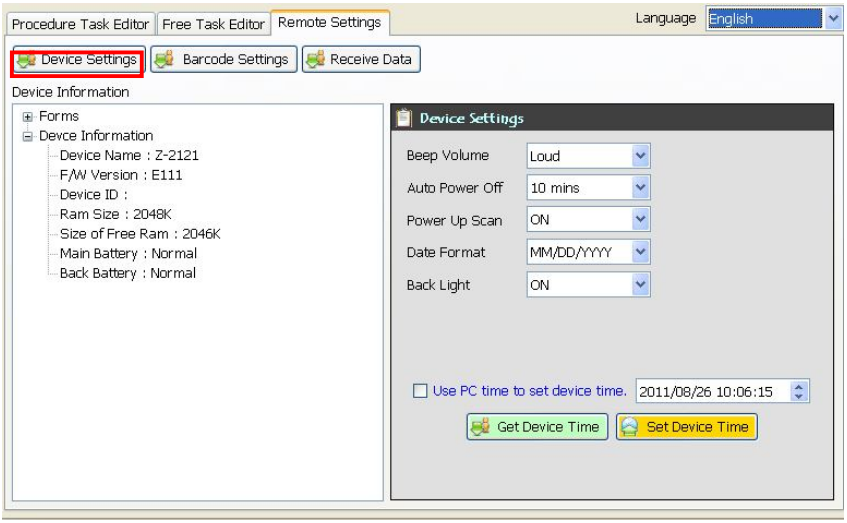
- COM options: Select the COM port the device is connected to.
- Baudrate options: 4800,9600,19200,38400,57600,115200.
- Parity options: Even, Mark, None, Odd, Space.
- DataBits options: 8,7.
- StopBits options: None, One, OnePointFive, Two.

3. After the parameters are set, the device is automatically detected.

Setting up the Device

To setup the device:

1. Select the "Remote Settings" tab on the Main Menu. Click the "Device Settings" to obtain the device status.



2. Select the options from the dropdown menus in "Device Settings."

- Set beep volume options: Quiet, Low, Medium, Loud.
- Set auto power off options: Disable, 10mins, 15mins, 20mins, 30 mins.
- Set power up scan options: OFF/ON.
- Set date format options: MM/DD/YY, DD/MM/YY, MM/DD/YYYY, DD/MM/YYYY.
- Set back light options: OFF/ON.
- Follow PC Time: Check this option to obtain time from your PC or adjust the time manually using the up and down arrow.
- Get Device Time: Obtain device now date and time.Set Device Time: Set up device date and time.

Creating Tasks

Before using a data collector product, you need at least one Task in the data collector unit (See Table 1). The function of "Task Editor" is to create Tasks for your data collector unit.

There are two kinds of files that can be executed in data collector products. One is Free Task (TSF) and the other is Procedure Task (TSK). A Free Task (TSF) is a simple Task. It is easy to edit, but has less functions. A Procedure Task (TSK), on the other hand, can make use of the full functions of data collector products. A Task is made up of Procedure units. A data collector can contain 1 Free Task, or 1 Procedure Task with up to 4 or 8 Procedures depending on model; each Procedure can contain several Macros.

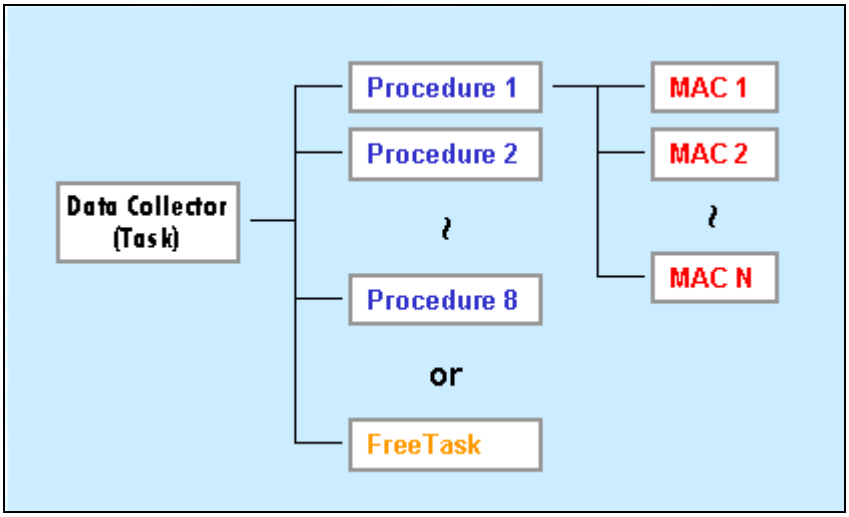


Table 1

Each Procedure has its own custom form. You can use the "UDF" function in "Procedure Task Editor" to appoint the form to the Procedure. Other forms can only be searched (see Table 2). Because the data collector unit can have no more than 4 or 8 Procedures (depending on model), the maximum number of forms is also 4, or 8.

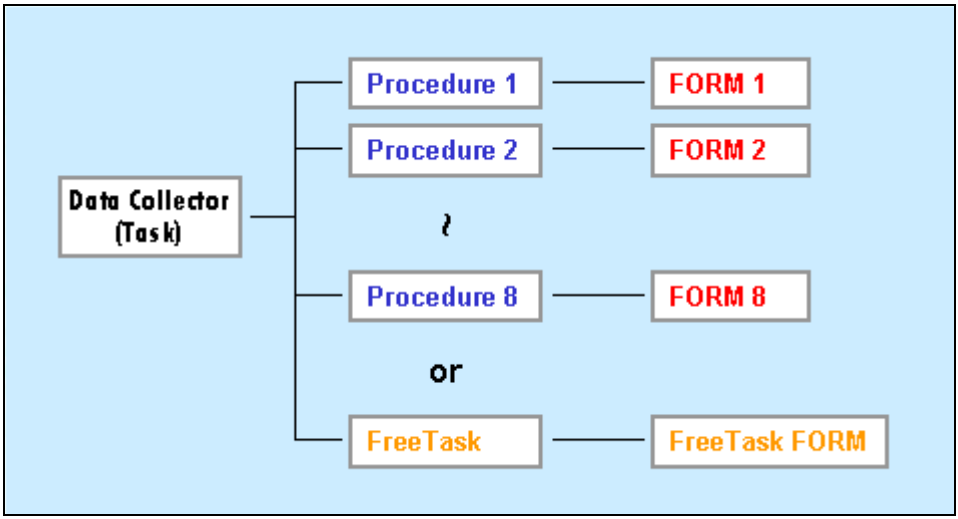


Table 2

All variables can be applied to every Procedure's MAC (See Table 3).

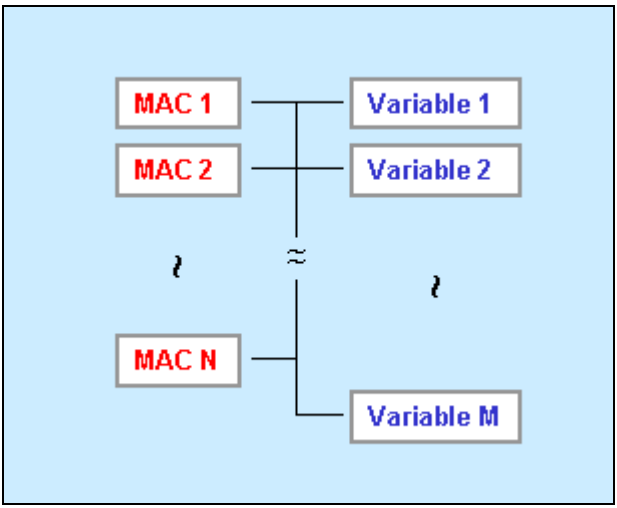


Table 3

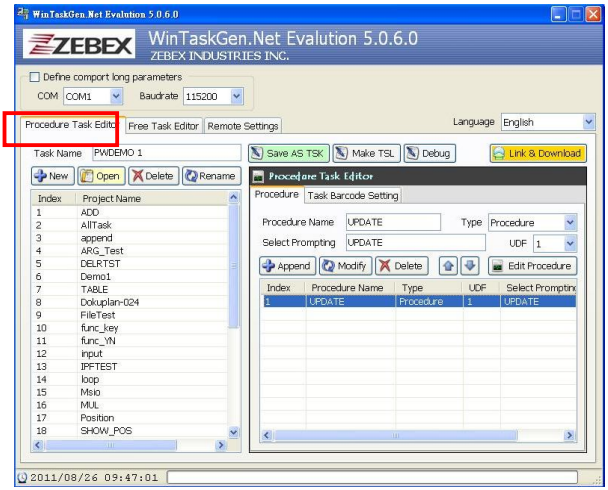
Now we will show you how to create a Task step by step. For working demo Tasks and Free Tasks, please refer to the "Demo" directory on the WinTaskGen CD or contact your dealer.

Creating a Procedure Task

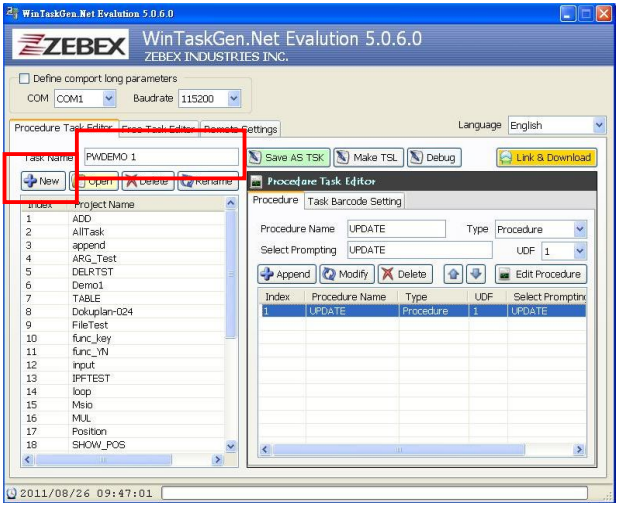
Procedure Tasks are useful for performing repeated or shared Tasks, such as frequently used calculations, text and control manipulation, and database operations.

Creating a New Task:

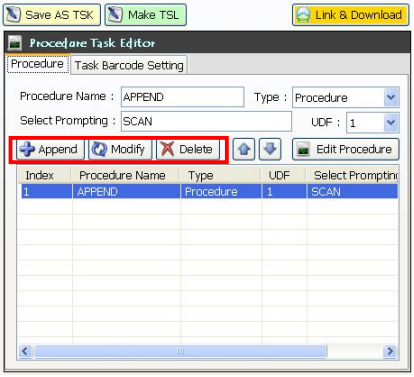
1. Select "Procedure Task Editor" on the "Main Manu."



2. Enter the "Task Name" and click "New" to create a new Task. "Task Name" can be named anything you prefer up to 8 characters. The new Task appears in the "Project Name."



3. After the Task is created, you can use the "Procedure Task Editor" on the right to append, modify, delete, or edit Procedures in the Task.



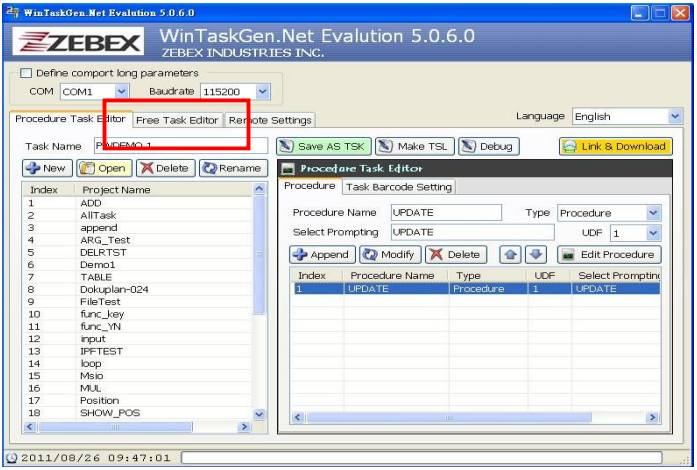
Please see Working with Procedure Tasks for more details.

Creating a Free Task

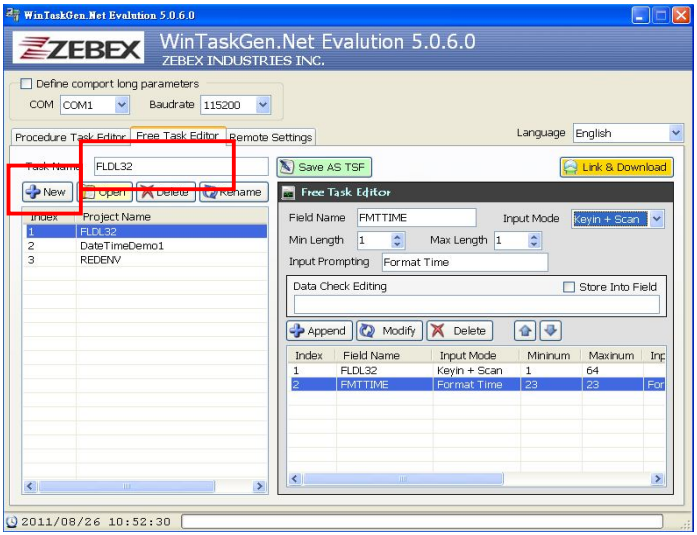
Free Tasks are useful for performing simple Tasks such as inputting and storing data.

Creating a New Task:

1. Select the "Free Task Editor" tab on the Main Menu.



2. Enter the "Task Name" and click "New" to create a new Task. "Field Name" can be named anything you prefer up to 8 characters. The new Task appears in the "Project Name."



3. Enter the "Field Name."

4. Select "Input Mode."

Input Mode	Meaning
Keyin Only	Input by keying data only
Scan Only	Input by scanning data only
Keyin+Scan	Input by keying and scanning data
Long time	Append MMDDYYYYhhmmss timestamp automatically
Short time	Append MMDDhhmm timestamp automatically
Format time	Append customized Date/Time format timestamp automatically

5. Select the "Max Length" and "Min Length."

6. Enter the texts to be displayed in "Input Prompting."

7. Check the ""$" Store into Field" to specify if the format symbols such as "/" should be stored as a part of the data to be input in the field. For example, "123110" input as a "date" will be displayed as 12/31/10 if you check this option.

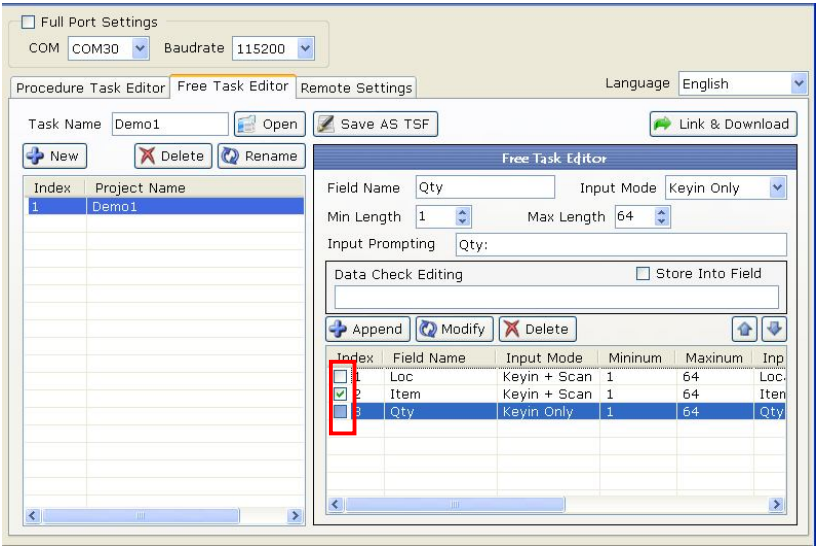
8. Enter "Data check Editing" to prompt how many kinds of character you can use, and what they are. For example, if the data you are going to input in a certain field contains 4 alphabetical characters, of which the first one is an upper case character and the others lower case ones, you input "ulll." Please see the following table for symbols you can use.

Editing	Meaning
n	number 0 ~ 9 and + , -
d	number 0 ~ 9
a	all Alpha characters A ~ Z , a ~ z
u	upper case Alpha characters A ~ Z
l	Lower case Alpha character a ~ z
c	Full ASCII characters
p	printable ASCII characters, ASCII code(32 - 127)
*****	Remove control character

The characters represented by "*", such as the check character(s) in the barcodes, will be deleted when the data is stored. Characters other than the symbols mentioned above (n, d, a, u, l, c, p, *) will be regarded as format symbols.

9. Select "Append", "Modify", or "Delete" in "Free Task Editor."

10. In version 5.07 later, WTG provides the looping control function in free task. User can indicates the start point of the loop for collecting data. As below, for example, user indicates the loop start at "Item". And then the free task will collect loc data first, and then item and its quantity data, and then item and its quantity, and item and its quantity,etc..



Note: If you do not indicate the start point of loop, then the free task will use the first field as the start point as default.

Downloading Tasks

The download function allows the users to download a Task file to a data collector unit.

To download Tasks you created to a data collector:

Changing Language

Default is English. In the Language folder of the WinTaskGen installed, users can create their languages - .LNG files, Japanese.LNG for instance. And then user can choose and change different language in the above Combo box.

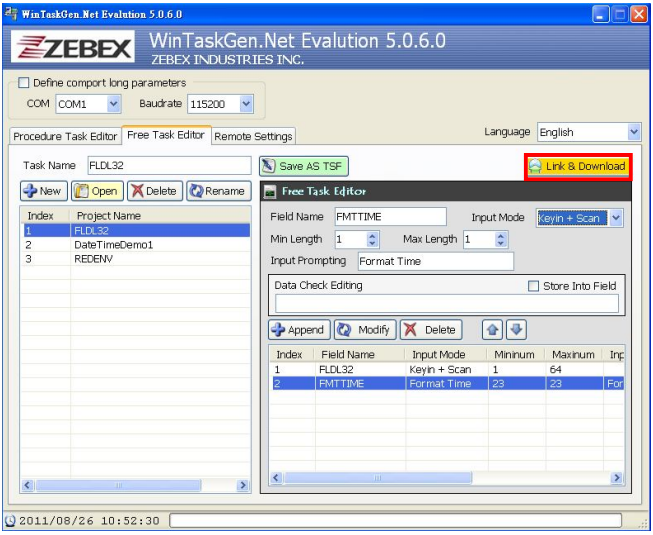
About the format of the language file, user can refer to the English.LNG format.

1. Connecting Data Collector" for more details.

2. Select "Procedure Task Editor" or "Free Task Editor" from the Main Menu.

3. Select the Task you want to download in the "Project Name."

4. Click "Link & Download"



5. Select "Yes" when system message appears to begin download.

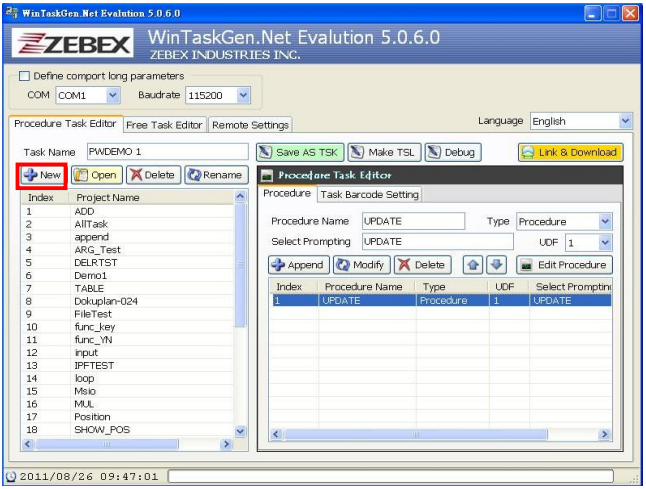
Importing Tasks

You can import existing Tasks to the WinTaskGen.

To import Tasks:

1. Select "Procedure Task Editor" or "Free Task Editor" from the Main Menu.

2. Click "Open" to browse Tasks in the computer.



3. Find and select the Task file you want to import and click ok.

4. Imported Tasks will appear in the "Project Name."

Please see System Variables

***(Asterisk)**

* is a temporary storage register, representing the contents of the current data entry after using INP or INX macro. It can be either numeric or string data type

Examples:

1. When TEMPVAR is a variable, then a procedure could have:

Line No.	Operator	Source Operand	Destination Operand
1	INP	*	
2	ADD	*	TEMPVAR

which will add input data to TEMPVAR.

DATE

Stores date data with the format: MMDDYY (MM-month, DD-date, YY-year).

Examples:

1. When TEMPVAR is a variable, then a procedure could have:

Line No.	Operator	Source Operand	Destination Operand
1	MOVE	DATE	TEMPVAR

which will move DATE data to TEMPVAR.

DATES

Stores date data in the **short** format: MMDD (MM-month, DD-date).

Examples:

1. When TEMPVAR is a variable, then a procedure could have:

Line No.	Operator	Source Operand	Destination Operand
1	MOVE	DATES	TEMPVAR

which will move DATES data to TEMPVAR.

TIME

Stores time data in the format: hhmmss (hh-hour, mm-minute, ss-second).

Examples:

1. When TEMPVAR is a variable, then a procedure could have:

Line No.	Operator	Source Operand	Destination Operand
1	MOVE	TIME	TEMPVAR

TIMES

Stores time data in the **short** format: hhmm (hh-hour, mm-minute).

Examples:

1. When TEMPVAR is a variable, then a procedure could have:

Line No.	Operator	Source Operand	Destination Operand
1	MOVE	TIMES	TEMPVAR

RECORDP

A record pointer pointing at the current record of the selected FORM. It could capture the current record number for other programming purposes.

Examples:

1. When TEMPVAR is a variable, then a procedure could have:

Line No.	Operator	Source Operand	Destination Operand
1	MOVE	RECORDP	TEMPVAR

RECORDL

This moves the record pointer to the end of file. **Examples:**

1. When TEMPVAR is a variable, then a procedure could have:

Line No.	Operator	Source Operand	Destination Operand

1	MOVE	RECORDL	TEMPVAR

Demo Program List in Appendix for the list of demo program you can import.

Receiving Data

You can receive data collected from your data collector unit.

To receive data:

Connect your data collector. See "Changing Language

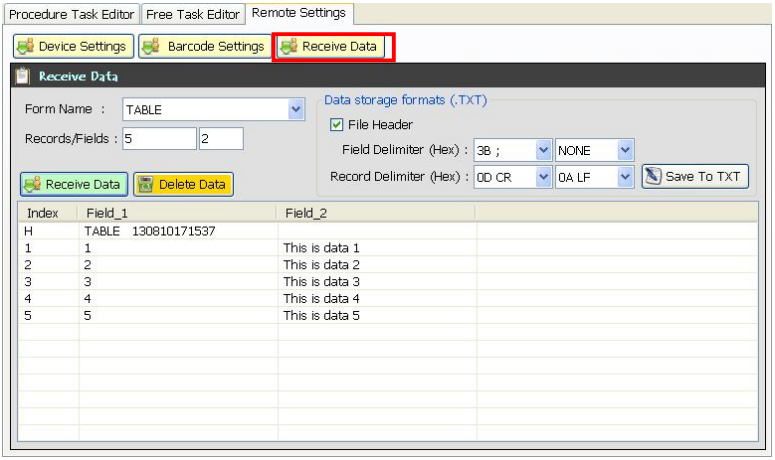
Default is English. In the Language folder of the WinTaskGen installed, users can create their languages - .LNG files, Japanese.LNG for instance. And then user can choose and change different language in the above Combo box.

About the format of the language file, user can refer to the English.LNG format.

1. Connecting Data Collector" for more details.

2. Select "Remote Settings" on the Main Menu.

3. Select "Receive Data" to obtain the device status.



To manage data to be received:

- Select "Form name" to determine what kind of form needs to be received.
- Select "Receive Data" button to get the data, and list view.
- Select “Delete Data” button to clear data in the device.
- Data Storage format (.txt): determine the format of the file to be stored.
  i. File Header: Save data include table header.
  ii. Field delimiter: Users can set the two-byte character to separate fields.
  iii. Record delimiter: Users can set the two-byte character to separate records.

Barcode Settings

Remote Barcode Settings

Learn how to customize the barcode settings for your data collector unit in this section.

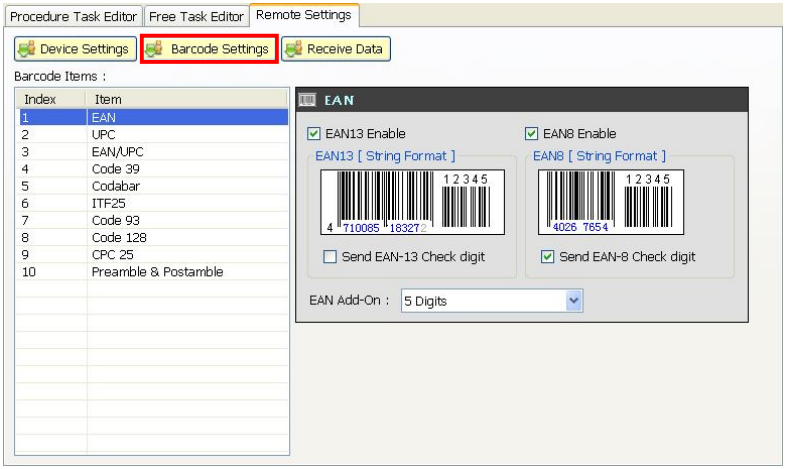
To set up the barcode settings for the unit:

Connect your data collector. See “Changing Language

Default is English. In the Language folder of the WinTaskGen installed, users can create their languages - .LNG files, Japanese.LNG for instance. And then user can choose and change different language in the above Combo box.

About the format of the language file, user can refer to the English.LNG format.

1. Connecting Data Collector” for more details.
2. Select “Remote Settings” on the Main Menu.
3. Select “Barcode Settings” to obtain the barcode parameter.



4. Select the barcode you want to setup in the "Barcode Items."

5. Check and change the options to appropriate values.

6. Settings are communicated to the Data Collector unit immediately.

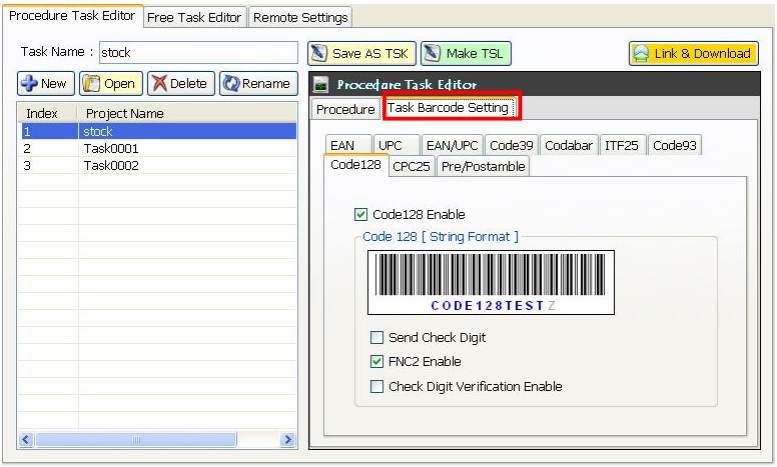
Procedure Barcode Settings

Learn how to customize barcode settings for your Procedure Tasks in this section.

To setup the barcode settings for your Procedure Tasks:

1. Select the "Procedure Task Editor" tab on the Main Menu.

2. Select the "Task Barcode Setting" tab to enter barcode settings.



3. Select the Task for the barcode settings in "Project Name."

4. Select the barcodes you want to setup by selecting the barcode tabs.

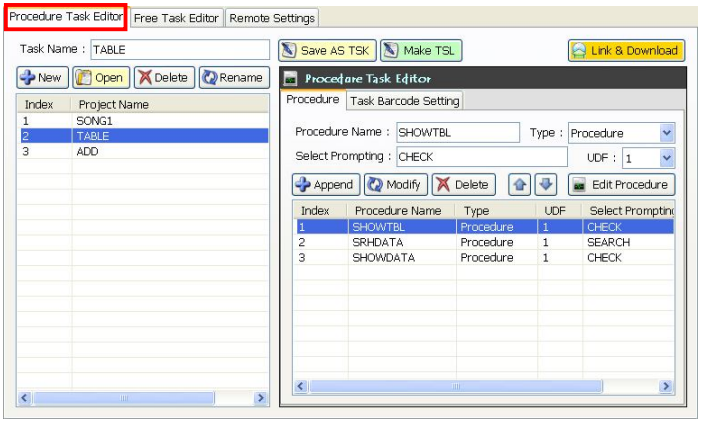
5. Check and change the options to appropriate values.

Please see Barcode Settings in Appendix for detailed instructions.

Working with Procedure Tasks

Managing Procedure Tasks

1. Select the "Procedure Task Editor" tab. The following screen appears.



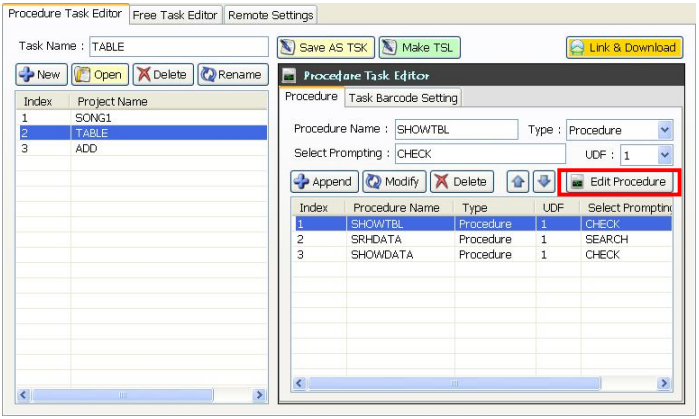
2. To use the "Procedure Task Editor", you can:

- Select "New", "Rename", "Delete" to add, rename, or delete a Task.
- Select "Open" to import Tsk files (*.tsk) into program for Task edit.
- Select "Save as TSK" to export Tsk file (*.tsk).
- Select "Make TSL" to save Tasks as Tsl format for older versions of WinTaskGen.
- Select "Link & Download" to link the program and then download it to the data collector device.
- "Procedure Name": Users can set 8 Procedures within a project.
- "Procedure Type": Select "Procedure" for programmable Procedure. Select "FreeTask" for Not programmable, only data fields inputs.
- "Prompting": Prepare the string displayed in the LCD.
- "UDF": Appoint the data form to the Procedure.
- "Edit Procedure": used to edit "Procedure" type Tasks.
- "Barcode Tab": Set barcode parameters.

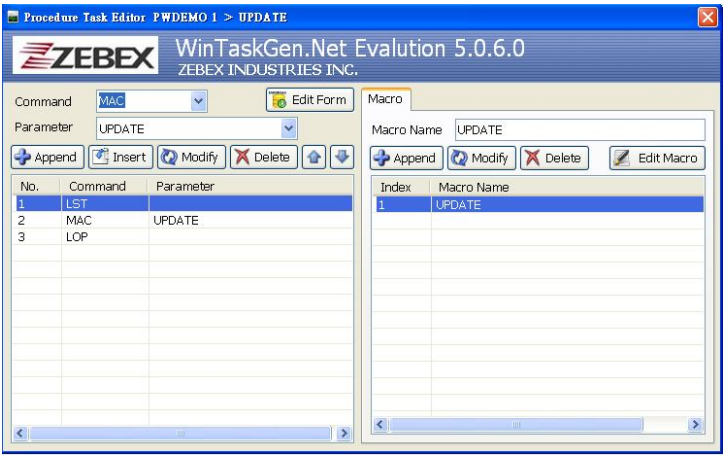
Editing Procedure Tasks

You can add and manage Macros, Forms, and Variables in the Procedures. Follow your own ideas to edit the Procedure Tasks.

1. Select "Edit Procedure" to edit selected Procedure.

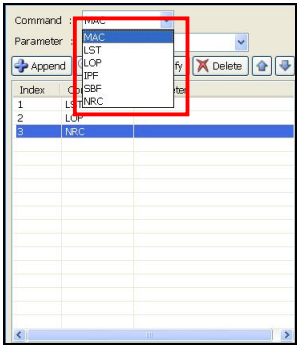


2. The "Procedure Task" screen appears as follows:



- The left side is the Procedure Task Editor and "Edit Form" button to edit Form. The right side is Macro editor.

3. Select "Command" to set the command type.

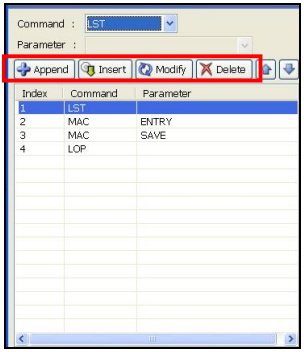


- Select "MAC" to define a Macro batch with name MAC_Name.
- Select "LST" or "LOP" to control process flow put LST at loop start point and put LOP into loop return point (Return to Loop start point).
- Select "IPF" to read data from keypad/scanner to the field you select.
- Select "SBF" to save buffer to field you select.
- Select "NCR" to save current record data and increment record pointer to next record.

Please see Procedure Commands in Appendix for more details

4. Select "Parameter" to select the Macro to be used.

5. Select "Append", "Insert", "Modify", or "Delete" to add, insert, modify, or delete Tasks.

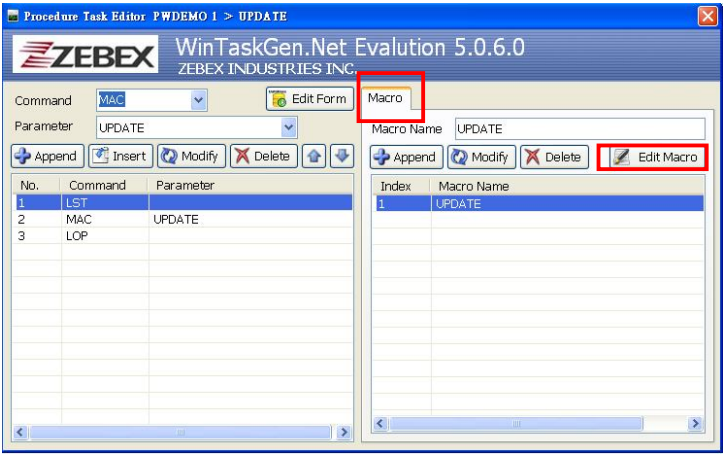


6. Select the "Macro", "Forms" or "Variables" tab.

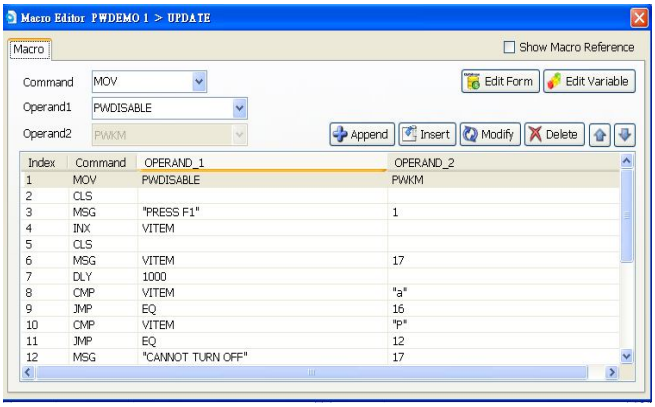
7. Select "Append", "Modify", "Delete", or "Edit" to edit the Macro, Forms, or Variables.

Managing Macro

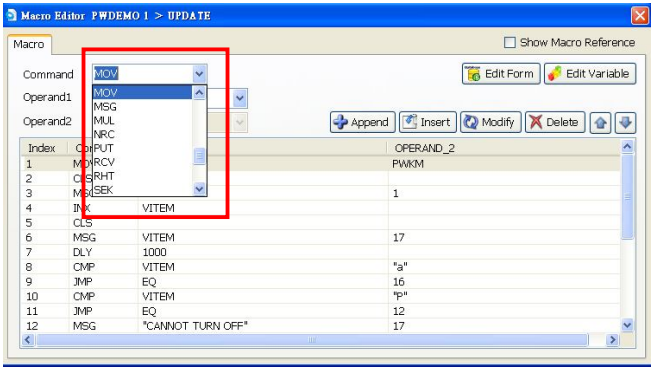
1. Select the "Macro" tab in the "Edit Procedure" screen (see Editing Procedure Tasks) and select "Edit Macro."



2. "Macro Editor" appears when you select "Edit Macro."



3. Select "Command" to select the Macro you want to use.



The list of Macros you can select:

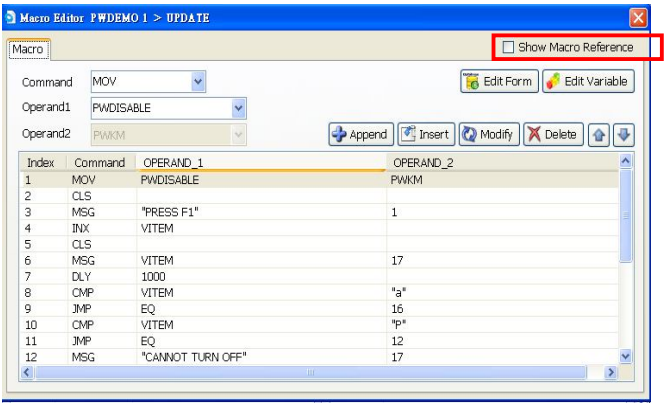
- Select "ADD" to add.
- Select "BEP" to make a sound.
- Select "CDV" to check barcode.
- Select "CKD" to check data format.
- Select "CKT" to check time format.
- Select "CLL" to clear 1 line.
- Select "CLS" to clear screen.
- Select "CMP" to compare.
- Select "DIV" to divide.
- Select "DLF" to delete form.
- Select "DLY" to add time delay.
- Select "END" to end Macro.
- Select "EXT" to exit data collection and go to run Task.
- Select "GET" to read data from a file.
- Select "GTO" to point the current record pointer to a record number in the specified form.
- Select "INP" to input and save.
- Select "INX" to get the input without pressing enter.

- Select "JMP" to jump when condition is met. Set the condition in "Operand1": EQ (equal), NE (not equal), GT (greater than), LT (less than) and UC (unconditional).
- Select "LEN" to store length.
- Select "LFT" to retrieve the leftmost characters of n length.
- Select "MOV" to copy a value and move to another.
- Select "MSG" to display message.
- Select "MUL" to multiply.
- Select "NCR" to save current record data and increment record pointer to next record
- Select "PUT" to write data to a file.
- Select "RCV" to receive a string from the serial port.
- Select "RHT" to retrieve the rightmost sub- string of n length.
- Select "RSV" to reserve for a future MACRO command.
- Select "SEK" to let the current record pointer point to the record number of the selected file.
- Select "SEL" to select the specified file.
- Select "SND" to send specified contents.
- Select "SRH" to search the selected data file.
- Select "STR" to convert data to string type.
- Select "SUB" to subtract.
- Select "VAL" to convert data to numeric type.

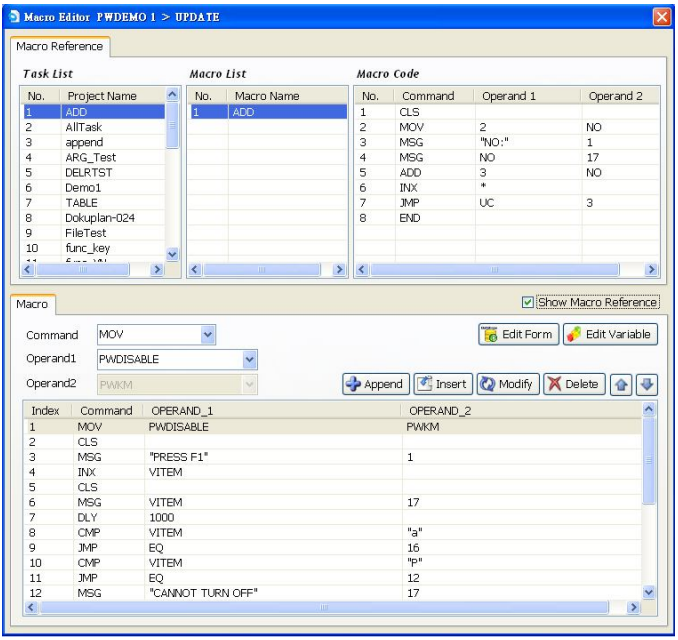
4. Select "Operand1" and "Operand2" for the source and/or destination operand.

5. Select "Append" or "Insert" to add the Macro. You can also select "Modify" or "Delete" to modify or delete existing Macros.

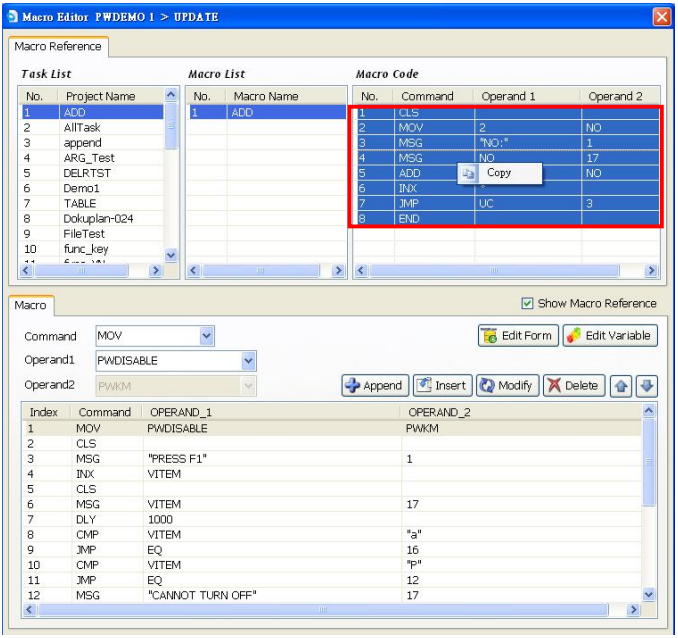
6. Check the "Show Macro Reference" to refer to other Macro program

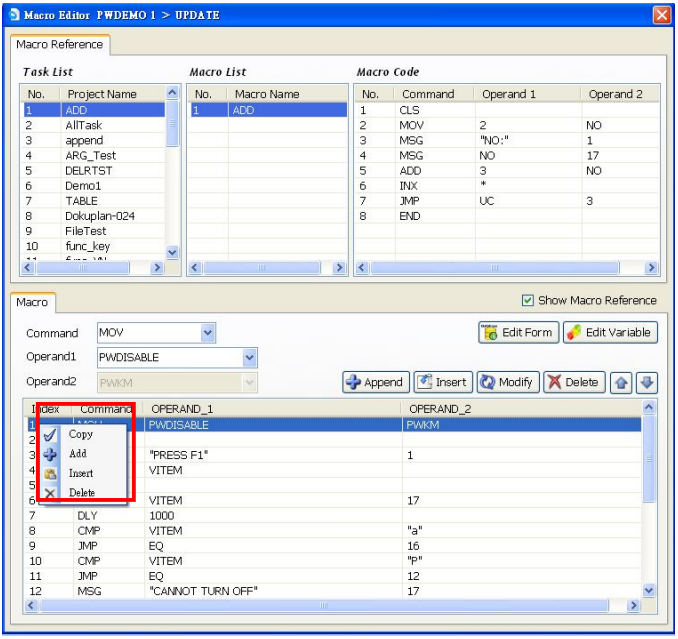


After checking the "Show Macro Reference", the Macro Reference tab shows Task List, Macro List, and Macro Code. Therefore, user can refer to the Macro in any task by indicating it in the Task List and Macro List accordingly.



Besides reading the referred macro code, user can "Copy" these commands in the Macro Code, and Add (appending these copied commands to the end of the program), Insert (appending these copied commands above the current editing Macro command)

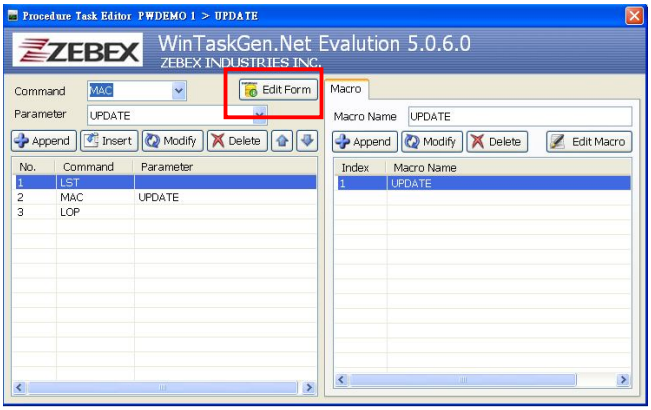

Macro Editor PWDEMO 1 > UPDATE
Macro Reference
Task List
Macro List
Macro Code
Copy
Macro
Show Macro Reference
Command
MOV
Operand1
PWDISABLE
Operand2
PWKM
Edit Form
Edit Variable
Append
Insert
Modify
Delete

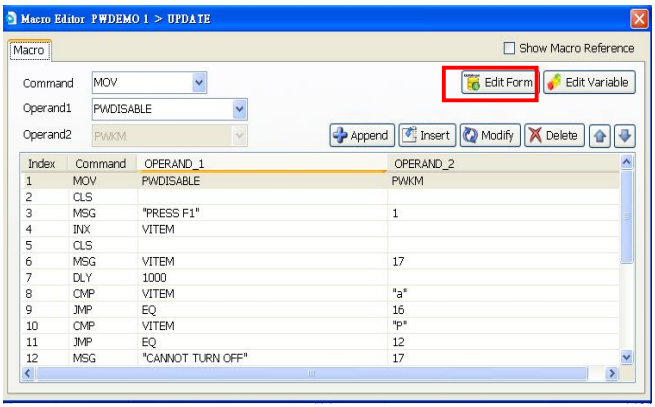

Macro Editor PWDEMO 1 > UPDATE
Macro Reference
Task List
Macro List
Macro Code
Copy
Add
Insert
Delete
Macro
Show Macro Reference
Command
MOV
Operand1
PWDISABLE
Operand2
PWKM
Edit Form
Edit Variable
Append
Insert
Modify
Delete

Please see Macro Commands in Appendix for more details.

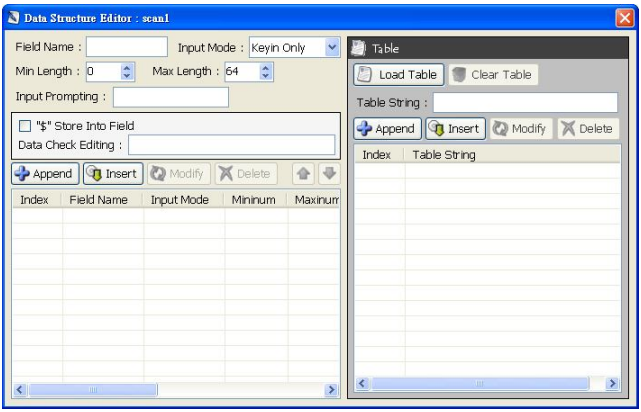
Managing Forms

1. Select "Edit Form" in the "Edit Procedure" screen (see Editing Procedure Tasks) or in the Macro Editor.





2. "Data Structure Editor" appears when you select "Edit Form."



3. Enter the "Field Name."

4. Select input or scan or both for your "Input Mode."

5. Select the "Max Length" and "Min Length."

6. Enter the texts to be displayed in "Input Prompting."

7. Enter "Data check Editing" to prompt how many kinds of character you can use, and what they are. For example, if the data you are going to input in a certain field contains 4 alphabetical characters, of which the first one is an upper case character and the others lower case ones, you input "ulll." Please see the following table for symbols you can use.

Editing	Meaning
n	number 0 ~ 9 and + , -
d	number 0 ~ 9
a	all Alpha characters A ~ Z , a ~ z
u	upper case Alpha characters A ~ Z
l	Lower case Alpha character a ~ z
c	Full ASCII characters
p	printable ASCII characters, ASCII code(32 - 127)
*****	Remove control character

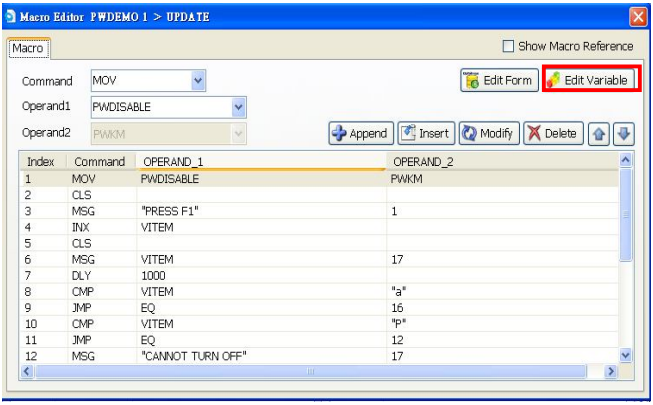
The characters represented by "*", such as the check character(s) in the barcodes, will be deleted when the data is stored. Characters other than the symbols mentioned

above (n, d, a, u, l, c, p, *) will be regarded as format symbols.

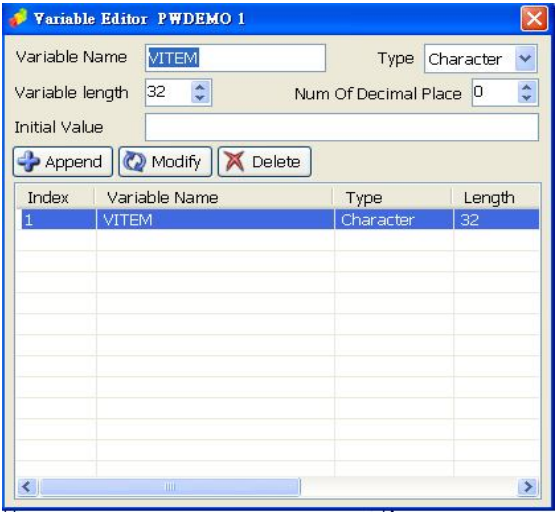
Managing Variables

Variables can be used by Macro to store data temporarily.

1. Select the "Edit Variable" button in the "Macro Editor" screen. (see Editing Procedure Tasks)



2. The "Variable" screen appears.



3. Select the Variable settings.

   - Variable Name: Enter the name for the variable.

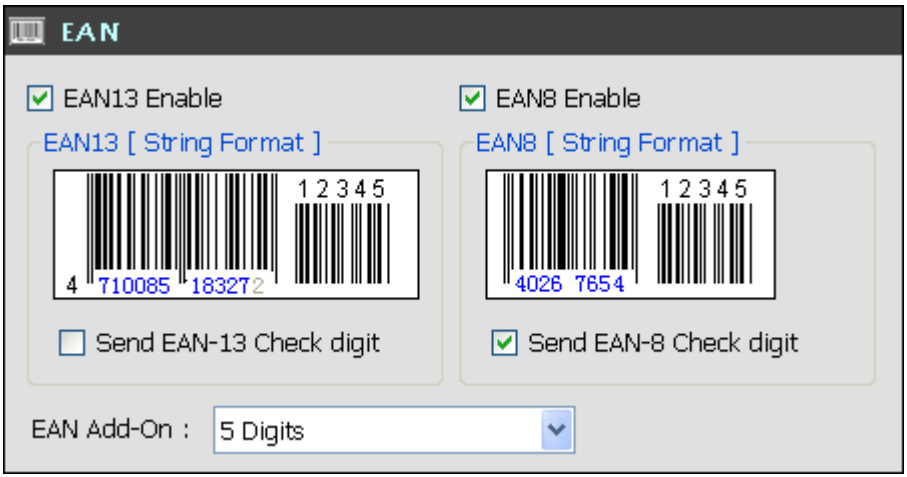
- Variable Type: Select Number or Character.
- Variable Length: Value in the range of 1-32.
- Num of Decimal place: Max Length is 10.

4. Select "Append", "Modify" or "Delete."

Appendix

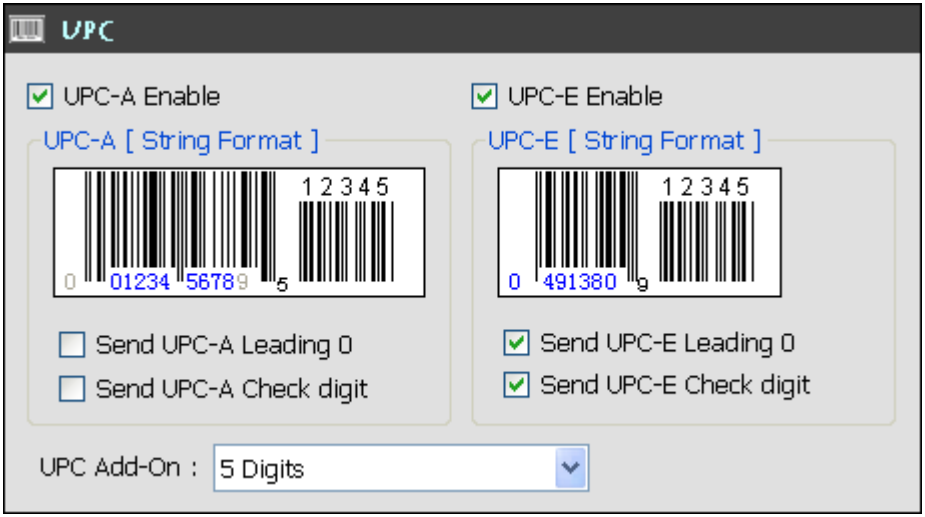
Barcode Settings

EAN



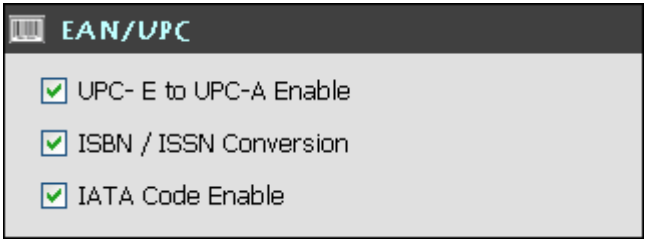
- EAN-8/13 Enable: Enable or disable code EAN-8/13 reading.
- Send EAN-8/13 check digit: Set whether to send `check digit`.
- EAN Add-On options:
- [No Supplement]: None support add-on barcode.
  i. [2 Digits]: Allow 2bytes add-on code.
  ii. [5 Digits]: Allow 5bytes add-on code.
  iii. [2 + 5 Digits]: Allow 2bytes and 5bytes add-on code.

UPC



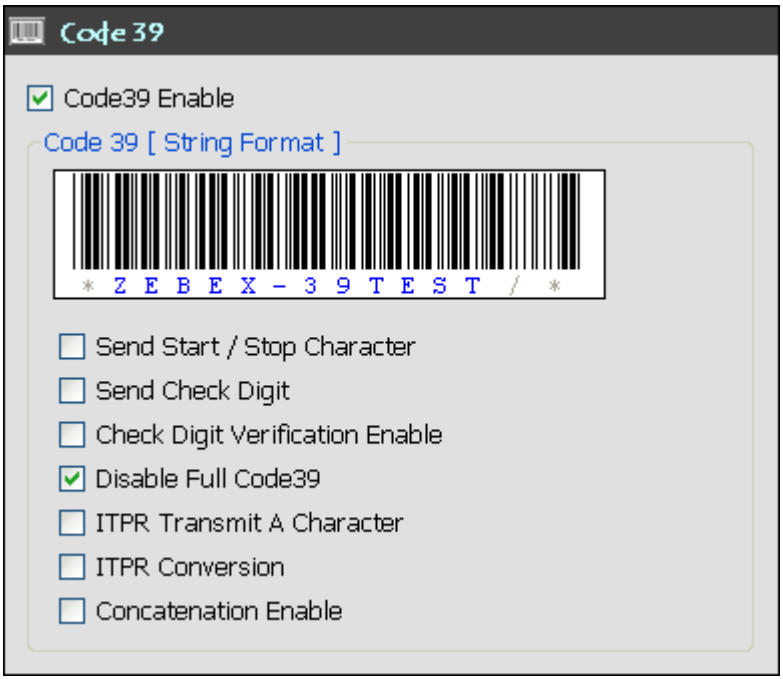
- UPC A/E Enable: Enable or disable UPC A/E code reading.
- Send UPC A/E Leading 0 : Set whether to send "0" String.
- Send UPC A/E check digit: Set whether to send check digit.
- UPC Add-On options:
  - i. [No Supplement]: None support add-on barcode.
  - ii. [2 Digits]: Allow 2bytes add-on code.
  - iii. [5 Digits]: Allow 5bytes add-on code.
  - iv. [2 + 5 Digits]: Allow 2bytes and 5bytes add-on code.

EAN/UPC



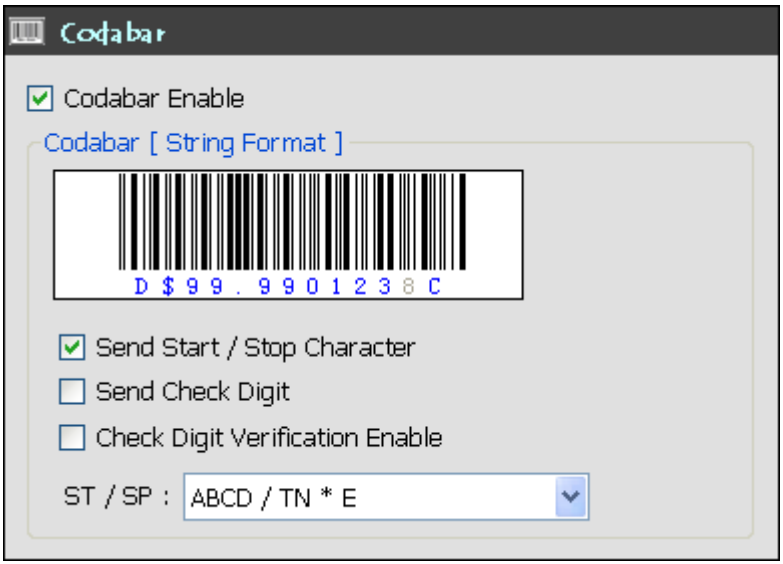
- UPC-E to A Enable: Enable or disable UPC-E to A.
- ISBN/ISSN Conversion: Enable or disable ISBN/ISSN Conversion.
- IATA Code Enable: Enable or disable IATA code reading.

Code39



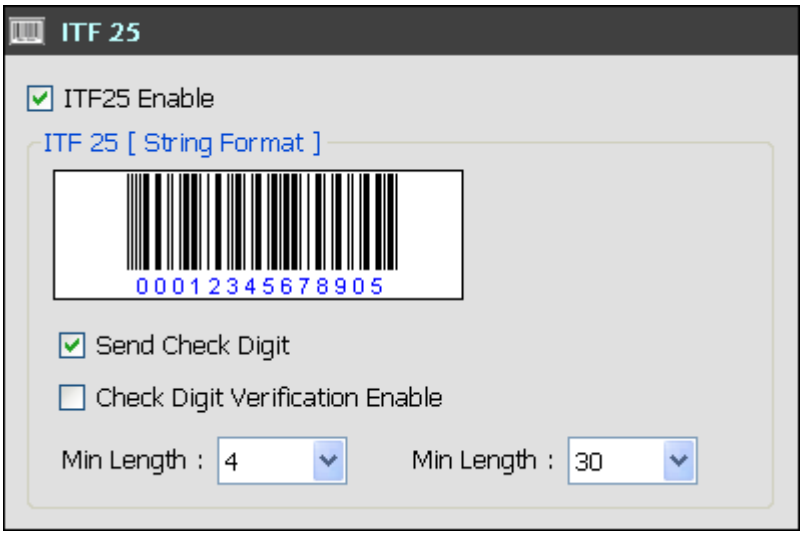
- Code 39 Enable: Enable or disable code UPC A/E code.
- Send Start/Stop Char: Set whether to send the "*"character.
- Send Check Digit: Set whether to send check digit.
- Check Digit Verification Enable: Set whether to enable CDV mode.
- Disable Full Code39: Set whether to disable full code39 mode.
- ITPR Transmit A Character: Set whether to start ITPR transfer mode.
- ITPR Conversion: Set whether to enable ITPR transfer mode.
- Concatenation enable: Set whether to enable Concatenation flag.

Codabar



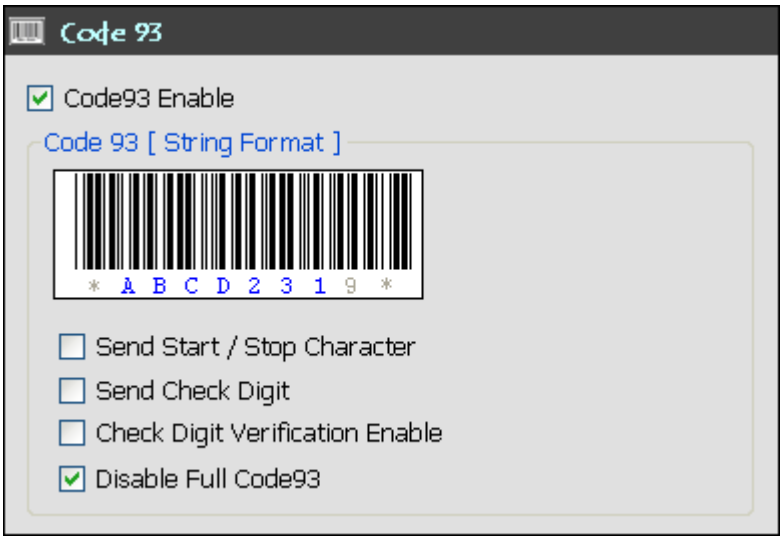
- Codabar Enable: Enable or disable Codabar code.
- Send Start/Stop Char: Set whether to send the "*" character.
- Send Check Digit: Set whether to send check digit.
- Check Digit Verification Enable: Set whether to enable CDV mode.
- ST/SP: select codebar start/stop character.
  - i. ABCD/TN * E.
  - ii. abcd/abcd.
  - iii. abcd/tn * e.
  - iv. DC1,DC2/DC1,DC2 or DC3,DC4/DC3,DC4.
  - v. ABCD/ABCD.

ITF25



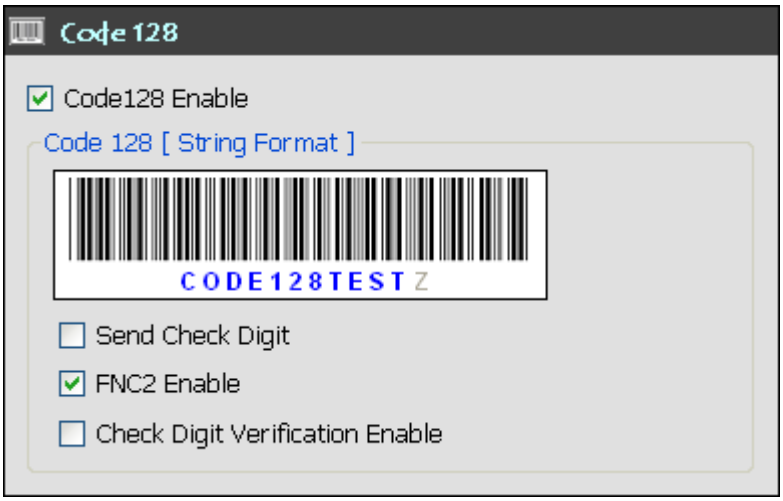
- ITF25 Enable: Enable or disable reading ITF25 code.
- Send Check Digit: To enable or disable sending the check digit string.
- Check Digit Verification Enable: To enable CDV mode.
- Max/Min Length: The value in the range 2-64.

Code93



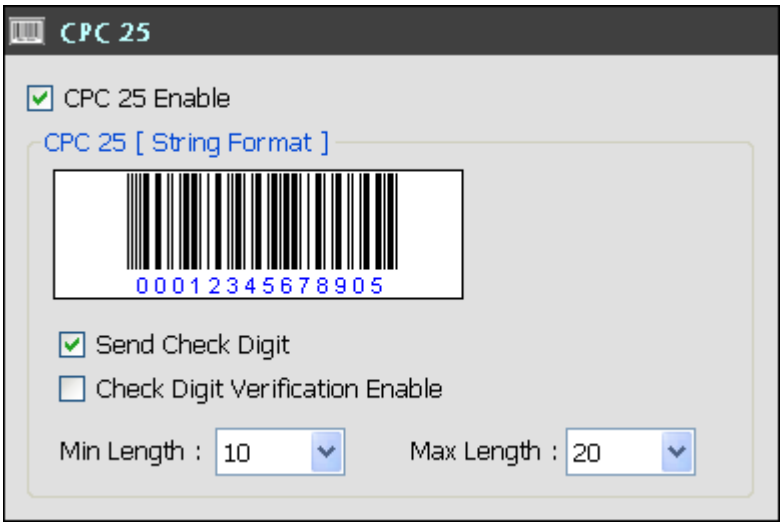
- Code93 Enable: Enable or disable reading Code93 code.
- Send Start/Stop Char: To Set device Whether to allow send the "*" character.
- Send Check Digit: To enable or disable sending the check digit string.
- Check Digit Verification Enable: To enable CDV mode.
- Disable Full Code39: To enable full code39 mode.

Code128



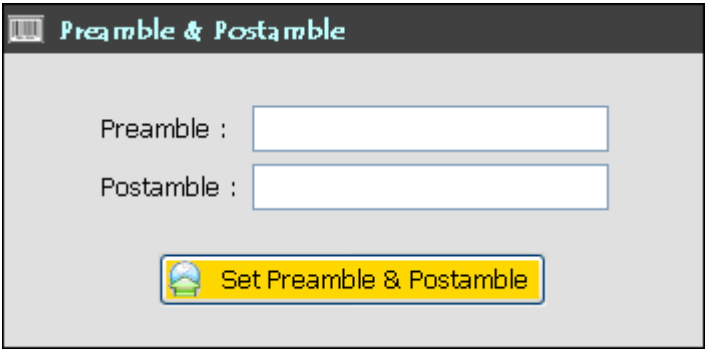
- Code128 Enable: Enable or disable reading Code128 code.
- Send Check Digit: To enable or disable sending the check digit string.
- FNC2 Enable: To enable FNC2 mode.
- Check Digit Verification Enable: To select CDV mode.

CPC25



- CPC25 Enable: Enable or disable reading CPC25 code.
- Send Check Digit: To enable or disable sending the check digit string.
- Check Digit Verification Enable: To select CDV mode.
- Max/Min Length: The value in the range 2-64.

Preamble & Postamble



- Preamble: Prefix.
- Postamble: Suffix.
- Select "Set Preamble & Postamble" to set device parameter.

Procedure Commands

MAC

Syntax:

Command	Name
MAC	MAC_ Name

Description: Define a Macro batch with name MAC_Name.

Examples:

Define the macro name as "MENU" Its function is for a display menu.

Line No.	Command	Name
1	MAC	MENU
2	.	
3	.	
4	LOP	

LST

LOP

Syntax:

Command	Name
LST	
LOP	

Description:

To control process flow put LST at loop start point and put LOP into loop return point (Return to Loop start point).

Examples:

Read customer name from keypad save to data field CUSTOMER.

Line No.	Operator	Source Operand
1	MAC	MENU
2	LST	
3	IPF	CUSTOMER
4	.	
5	NRC	
6	LOP	

IPF

Syntax:

Command	Name
IPF	Field Name

Description: Read data from keypad/scanner to the field CUSTOMER.

Examples:

Read customer name from keypad save to data field CUSTOMER.

Line No.	Operator	Source Operand
1	MAC	MENU
2	LST	
3	IPF	CUSTOMER
4	LOP	

NCR

Syntax:

Operator	Source Operand	Destination Operand
NCR		

Description:

Save current record data and increment record pointer to next record.

Operand Data Type: None.

Examples: If TEMP_F is defined as a Form Field then program will store the number of characters of "TEST" (4) in TEMP_F.

Line No.	Operator	Source Operand	Destination Operand
1	MOV	"TEST"	TEMP_F
2	NCR		

SBF

Syntax:

Command	Name
SBF	

Description: Save buffer (*) to field.

Examples: Read customer name one time for each record.

Line No.	Command	Name
1	MAC	IN_CUST
2	LST	
3	MAC	CUST
4	SBF	CUSTOMER
5	.	
6	LOP	

Define IN_CUST as follows:

Line No.	Operator	Source Operand	Destination Operand
1	INP	CUSTVAR	
2	END		

Define CUST as follows:

Line No.	Operator	Source Operand	Destination Operand
1	MOV	CUSTVAR	*
2	END		

Macro Commands

ADD

Syntax:

Operator	Source Operand	Destination Operand
ADD	OPERAND1	OPERAND2

Description:

Adds the source operand content to the destination operand and stores the result in the destination operand. If the source operand is a string, it will be appended to the destination operand.

Operand Data Type:

OPERAND1

numeric constant, numeric variable, string constant, string variable, DATE string, TIME string, current field content

OPERAND2

numeric variable, string variable

Examples:

1. When TEMPVAR is a variable, then a procedure could have:

Line No.	Operator	Source Operand	Destination Operand
1	INP	*	
2	ADD	*	TEMPVAR

which will add input data to TEMPVAR.

2. If TEMPVAR holds the string "Yourgroup," then

Line No.	Operator	Source Operand	Destination Operand
1	ADD	"Co. LTD."	TEMPVAR

will cause TEMPVAR to become "Yourgroup Co. LTD."

BEP

Syntax:

Operator	Source Operand	Destination Operand
BEP	OPERAND1	OPERAND2

Description:

Drive the Data Collector speaker to make a sound. OPERAND1 holds the duration and OPERAND2 specifies the tone.

Operand Data Type:

OPERAND1

numeric constant, numeric variable,range from 100 (0.1 seconds) to 32767 (32.767 seconds)

OPERAND2

numeric constant, numeric variable range from 100 to 3000 HZ

Examples:

Line No.	Operator	Source Operand	Destination Operand
1	BEP	500	600
2	DLY	800	

Causes Data Collector unit speaker to emit a beep lasting for 0.5 seconds at 600 HZ and delay 0.8 seconds.

CDV

Syntax:

Operator	Source Operand	Destination Operand
CDV	OPERAND1	OPERAND2

Description:

Verify OPERAND1' s data by CDV method. If it matches then the flag is set to EQ, if not, then it is set to NE.

Operand Data Type:

OPERAND1

numeric variable

Examples:

Line No.	Operator	Source Operand	Destination Operand
1	INP	*	
2	CDV	*	
3	JMP	EQ	5
4	JMP	UC	1
5		.	
6		.	
7		.	

If we want to check the current data entry, we can use these macros.

CKD

Syntax:

Operator	Source Operand	Destination Operand
CKD	OPERAND1	

Description:

Check if the OPERAND1 is the date format or not.

If OPERAND1 is in date format, the flag is set to EQ;

If OPERAND1 is not in date format, the flag is set to NE.

Operand Data Type:

OPERAND1

string variable

Examples:

In the verification process we define macros to reject data which is not in the date format- MMDDYY.

Line No.	Operator	Source Operand	Destination Operand
1	INP	*	
2	CKD	*	
3	JMP	EQ	5
4	JMP	UC	1
		.	
		.	

CKT

Syntax:

Operator	Source Operand	Destination Operand
CKT	OPERAND1	

Description:

Check if OPERAND1 is in the time format or not.

If OPERAND1 is in the time format, the flag is set to EQ;

If OPERAND1 is not in the time format, the flag is set to NE.

Operand Data Type:

OPERAND1: string variable

Examples:

In the verification process we define MACROs to reject data which is not in the hhmmss time format.

Line No.	Operator	Source Operand	Destination Operand
1	INP	*	
2	CKT	*	
3	JMP	EQ	5
4	JMP	UC	1
5		.	
6		.	
7		.	

CLL

Syntax:

Operator	Source Operand	Destination Operand
CLL	OPERAND1	

Description:

Clear one line of the LCD display.

Operand Data Type:

OPERAND1

1,2,3,4,5,6 (Depends on number of lines on Data Collector unit screen. Max may be 4, may be 6, etc.)

Examples:

Clear line 1 of the LCD display.

Line No.	Operator	Source Operand	Destination Operand
1	CLL	1	

CLS

Syntax:

Operator	Source Operand	Destination Operand
CLS		

Description:

Clear LCD screen.

Examples:

Clear entire LCD display.

Line No.	Operator	Source Operand	Destination Operand
1	CLS		

CMP

Syntax:

Operator	Source Operand	Destination Operand
CMP	OPERAND1	OPERAND2

Description:

Compare OPERAND1 with OPERAND2, and set the flag;

if OPERAND1 is equal to OPERAND2, the flag is set to **EQ**;

if OPERAND1 is not equal to OPERAND2, the flag is set to **NE**;

if OPERAND1 is greater than OPERAND2, the flag is set to **GT**;

if OPERAND1 is less than OPERAND2, the flag is set to **LT**.

The operand contents are not changed.

Operand Data Type:

OPERAND1

numeric constant, numeric variable, string constant, string variable,DATE string, TIME string

OPERAND2

numeric constant, numeric variable, string constant, string variable

Examples:

In the verification process we define these macros to re-enter data if 0 is entered.

Line No.	Operator	Source Operand	Destination Operand
1	INP	*	
2	CMP	*	0
3	JMP	NE	5
4	JMP	UC	1
5		.	

DIV

Syntax:

Operator	Source Operand	Destination Operand
DIV	OPERAND1	OPERAND2

Description:

Divide OPERAND2 by OPERAND1 and store the result in OPERAND2.

Operand Data Type:

OPERAND1

numeric constant, numeric variable

OPERAND2

numeric variable

Examples:

When TEMPVAR is defined as a numeric variable; then

Line No.	Operator	Source Operand	Destination Operand
1	DIV	10	TEMPVAR

will divide TEMPVAR by 10 and store the result in TEMPVAR.

DLF

Syntax:

Operator	Source Operand	Destination Operand
DLF		

Description:

Delete user form(files) data.

Operand Data Type:

NONE.

Examples:

Clear all data and exit process.

Line No.	Operator	Source Operand	Destination Operand
1	DLF		
2	EXT		

DLY

Syntax:

Operator	Source Operand	Destination Operand
DLY	OPERAND1	

Description:

The delay period depends on the value of OPERAND1.

Operand Data Type:

OPERAND1

numeric constant, numeric variable

range from 0 to 32767 (32.767 seconds)

Examples:

We define the delay macro to allow the user time to read the data.

Line No.	Operator	Source Operand	Destination Operand
1	DLY	1000	

END

Syntax:

Operator	Source Operand	Destination Operand
END		

Description:

End of macro command execution. A process may have several END macros.

Operand Data Type: NONE

Examples:

TEMPVAR is defined as a numeric variable; TEMPVAR will be divided by 100 if it is greater than 1000, but will be divided by 10, if not.

Line No.	Operator	Source Operand	Destination Operand
1	CMP	TEMPVAR	1000
2	JMP	GT	5
3	DIV	10	TEMPVAR
4	END		
5	DIV	100	TEMPVAR
6	END		

EXT

Syntax:

Operator	Source Operand	Destination Operand
EXT		

Description:

Exit data collection and go to **RUN TASK** state.

Operand Data Type: NONE.

GET

Syntax:

Operator	Source Operand	Destination Operand
GET	OPERAND1	OPERAND2

Description:

OPERAND1 holds the field number or name of the current record. This macro gets the field data of the current record and stores it in OPERAND2.

Operand Data Type:

OPERAND1
numeric constant, fieldname

OPERAND2
numeric variable, string variable

Examples:

Where TEMPVAR is defined as a variable; then

Line No.	Operator	Source Operand	Destination Operand
1	SEL	ORDER	
2	SEK	10	
3	GET	3	TEMPVAR

GTO

Syntax:

Operator	Source Operand	Destination Operand
GTO	OPERAND1	

Description:

Point the current record pointer to a record number in the specified form.

Operand Data Type:

OPERAND1

numeric constant, numeric variable

Example:

1. Select the form named ORDER and go to the last record (RECORDP holds last record number).

Line No.	Operator	Source Operand	Destination Operand
1	SEL	ORDER	
2	GTO	10	
3	GET	3	TEMPVAR

INP

Syntax:

Operator	Source Operand	Destination Operand
INP	OPERAND1	

Description:

Read data from the bar code scanner or keypad and store in OPERAND1. The data format depends on the fields format setting. When OPERAND1 is a field variable.

Operand Data Type:

OPERAND1

numeric variable, string variable, Field variable (field name)

Examples:

If TEMPVAR is defined as a numeric variable; then

Line No.	Operator	Source Operand	Destination Operand
1	INP	TEMPVAR	
2	CMP	TEMPVAR	1000
3	JMP	LT	5
4		.	
5		.	

If ITEM is defined as a field name; then

Line No.	Operator	Source Operand	Destination Operand
1	INP	ITEM	
2	MSG	"ITEM:"	1
3	MSG	ITEM	6
4		.	

INX

Syntax:

Operator	Source Operand	Destination Operand
INX	OPERAND1	

Description:

Get one character from the keypad and store in OPERAND1.The input data does not display on the LCD.

Operand Data Type:

OPERAND1 : string variable, numeric variable.

Examples:

If TEMPVAR is defined as a string variable; then

Line No.	Operator	Source Operand	Destination Operand
1	MSG	"(1) 5%/(2) 15%TAX"	49
2	INX	TEMPVAR	
3	CMP	TEMPVAR	"1"
4	JMP	EQ	30
5		Compute 15%TAX	
6		.	
.		.	
.		.	
29	END		
30		Compute 5%TAX	
31		.	
.		.	

JMP

Syntax:

Operator	Source Operand	Destination Operand
JMP	OPERAND1	OPERAND2

Description:

Jumping to OPERAND2 depends on the flag setting from the previous compare.

Operand Data Type:

OPERAND1

EQ, NE, GT, LT and UC(unconditional jump)

OPERAND2

Line number of macro command, constant

Examples:

If TEMPVAR is defined as a numeric variable; then

Line No.	Operator	Source Operand	Destination Operand
1	CMP	TEMPVAR	10
2	JMP	GT	15
3		.	
.		.	
.		.	
15	JMP	UC	25
19		.	
.			
25		.	
.		.	
30	END		

will compare TEMPVAR with 10, if TEMPVAR is greater than 10,then execute the macro at line number 15 then do an unconditional jump to line number 25, if not execute macros from line numbers 3 to 14.

LEN

Syntax:

Operator	Source Operand	Destination Operand
LEN	OPERAND1	OPERAND2

Description:

Store the length of OPERAND1 in OPERAND2.

Operand Data Type:

OPERAND1

string variable , string constant

OPERAND2

numeric variable

Examples:

If TEMPVAR is defined as a variable; then

Line No.	Operator	Source Operand	Destination Operand
1	LEN	TEST	TEMPVAR

will store the number of characters of "TEST" (4) in TEMPVAR.

LFT

Syntax:

Operator	Source Operand	Destination Operand
LFT	OPERAND1	OPERAND2

Description:

Retrieve the leftmost characters of n length of OPERAND2 and store the sub-string back in OPERAND2, where length n is specified by OPERAND1.

Operand Data Type:

OPERAND1

numeric constant, numeric variable

OPERAND2

string variable

Examples:

TEMPVAR is defined as string and TEMPCNT as numeric variable; then

Line No.	Operator	Source Operand	Destination Operand
1	LEN	TEMPVAR	TEMPCNT
2	SUB	1	TEMPCNT
3	LFT	TEMPCNT	TEMPVAR

will truncate the last character of TEMPVAR.

MOV

Syntax:

Operator	Source Operand	Destination Operand
MOV	OPERAND1	OPERAND2

Description:

Copy the content of OPERAND1 to OPERAND2.

OPERAND1 is not changed, OPERAND2 becomes the same as OPERAND1.

Operand Data Type:

OPERAND1: numeric constant, numeric variable, string constant, string variable, DATE string, TIME string, field name of current append record

OPERAND2: numeric variable, string variable, field variable field name of current append record

Examples:

If we want to add date to DATE _F field, and display on LCD position 1

Line No.	Operator	Source Operand	Destination Operand
1	MOV	DATE	DATE_F
2	MSG	DATE	1
3		.	

MSG

Syntax:

Operator	Source Operand	Destination Operand
MSG	OPERAND1	OPERAND2

Description:

Display the message held in OPERAND1 at LCD position held in OPERAND2.Character '/' used as new line control character.

Operand Data Type:

OPERAND1: numeric variable, string constant, string variable

OPERAND2: numeric constant, numeric variable

Examples:

If TEMPVAR is defined as a variable; then

Line No.	Operator	Source Operand	Destination Operand
1	MOV	TEMPVAR	*
2	MSG	*	17

Display string constant(Task Generator)at LCD position 1.

Line No.	Operator	Source Operand	Destination Operand
1	MSG	"Task Generator"	1

MUL

Syntax:

Operator	Source Operand	Destination Operand
MUL	OPERAND1	OPERAND2

Description:

Multiply OPERAND1 by OPERAND2 and store result in OPERAND2.

Operand Data Type:

OPERAND1

numeric constant, numeric variable

OPERAND2

numeric variable

Examples:

TEMPVAR is defined as numeric variable; then

Line No.	Operator	Source Operand	Destination Operand
1	MUL	10	TEMPVAR

will multiply 10 by TEMPVAR and store result in TEMPVAR.

NCR

Syntax:

Operator	Source Operand	Destination Operand
NCR		

Description:

Save current record data and increment record pointer to next record.

Operand Data Type:

None

Examples:

If TEMP_F is defined as a Form Field; then

Line No.	Operator	Source Operand	Destination Operand
1	MOV	"TEST"	TEMP_F
2	NCR		

will store the number of characters of "TEST" (4) in TEMP_F

PUT

Syntax:

Operator	Source Operand	Destination Operand
PUT	OPERAND1	OPERAND2

Description:

Put OPERAND2' s data to the current field of the current record. The field number is held in OPERAND1.

Operand Data Type:

OPERAND1

numeric constant, field name

OPERAND2

numeric constant, numeric variable, string constant, string variable

Examples:

We want to put the string "MYGROUP" in field 1 of the 10th record of the file "CUSTOM."

Line No.	Operator	Source Operand	Destination Operand
1	SEL	CUSTOM	
2	SEK	10	
3	PUT	1	"MYGROUP"

RCV

Syntax:

Operator	Source Operand	Destination Operand
RCV	OPERAND1	OPERAND2

Description:

Receive a string from the serial port, and store it in OPERAND2 in the period of time specified in OPERAND1.

Operand Data Type:

OPERAND1

Time window in thousandths of a second as a numeric constant, numeric variable range from 0 to 32767 (32.767 seconds)

OPERAND2

string variable

Examples:

If TEMPVAR is defined as a string variable; then

Line No.	Operator	Source Operand	Destination Operand
1	RCV	1000	TEMPVAR

will receive a string from the serial port and store the string in TEMPVAR within 1 second.

RHT

Syntax:

Operator	Source Operand	Destination Operand
RHT	OPERAND1	OPERAND2

Description:

Retrieve the rightmost sub- string of n length of OPERAND2 and store the sub-string back to OPERAND2, where length n is specified by.

Operand Data Type:

OPERAND1

numeric constant, numeric variable

OPERAND2

string variable

Examples:

If TEMPVAR is defined as a string variable and TEMPCNT as a numeric

Line No.	Operator	Source Operand	Destination Operand
1	LEN	TEMPVAR	TEMPCNT
2	SUB	1	TEMPCNT
	RHT	TEMPCNT	TEMPVAR

will truncate off the first character of TEMPVAR.

RSV

Syntax:

Operator	Source Operand	Destination Operand
RSV		

Description:

Reserved for a future MACRO command.

Operand Data Type:

None

SEK

Syntax:

Operator	Source Operand	Destination Operand
SEK	OPERAND1	

Description:

Let the current record pointer point to the record number of the selected file, OPERAND1 specifies the record number, and the file must be selected by "SEL" macro first.

Operand Data Type:

OPERAND1

numeric constant, numeric variable

Examples:

The following macros

Line No.	Operator	Source Operand	Destination Operand
1	SEL	CUSTOM	
2	SEK	10	

will select file named "CUSTOM" and move the record pointer of the selected file to the 10th record.

SEL

Syntax:

Operator	Source Operand	Destination Operand
SEL	OPERAND1	

Description:

Select the file whose name is specified in OPERAND1 The file name must be a FORM name loaded in the Data Collector unit.

Operand Data Type:

OPERAND1

string constant

Examples:

The following macros

Line No.	Operator	Source Operand	Destination Operand
1	SEL	ORDER	

SND

Syntax:

Operator	Source Operand	Destination Operand
SND	OPERAND1	OPERAND2

Description:

Send both the content held by OPERAND1 and a terminator character specified in OPERAND2 from the serial port.

Operand Data Type:

OPERAND1: string constant, string variable

OPERAND2: numeric constant as

0: terminator CR (carriage return)

1: terminator LF (line feed)

2: terminator CRLF (carriage return and line feed)

3: no terminator

Examples:

If TEMPVAR is defined as a string variable; then

Line No.	Operator	Source Operand	Destination Operand
1	SND	TEMPVAR	2

will send the content of TEMPVAR and a terminator "CRLF" from the serial port.

SRH

Syntax:

Operator	Source Operand	Destination Operand
SRH	OPERAND1	OPERAND2

Description:

Search the selected data file, determine whether the same data is held in OPERAND2 or not, OPERAND1 specifies the field number. This macro supports the sequential search of the file.

If the data held in OPERAND2 is found in the file, the compare flag is set to EQ.

if it is not found, the compared flag is set to NE.

Operand Data Type:

OPERAND1: numeric constant, numeric variable

OPERAND2: string constant, string variable

Examples:

Search for the string "HOTWORD" from record number 1 in the file "CUSTOM" field number 1.

Line No.	Operator	Source Operand	Destination Operand
1	SEL	CUSTOM	
2	SEK	1	
3	SRH	1	HOTWORD

STR

Syntax:

Operator	Source Operand	Destination Operand
STR	OPERAND1	OPERAND2

Description:

Convert the data held in OPERAND1 to string type and store it in OPERAND2.

Operand Data Type:

OPERAND1: numeric constant, numeric variable

OPERAND2: string variable

Examples:

TEMPVAR is defined as a string variable; then.

Line No.	Operator	Source Operand	Destination Operand
1	STR	12345	TEMP33

will convert number 12345 to string "12345" and store in TEMP33.

SUB

Syntax:

Operator	Source Operand	Destination Operand
SUB	OPERAND1	OPERAND2

Description:

Subtract OPERAND1 from OPERAND2 and store the result in OPERAND2.

Operand Data Type:

OPERAND1: numeric constant, numeric variable

OPERAND: numeric variable

Examples:

If TEMPVAR is defined as a numeric variable and the content is 20; then

Line No.	Operator	Source Operand	Destination Operand
1	SUB	10	TEMPVAR

will change TEMPVAR value to 10.

VAL

Syntax:

Operator	Source Operand	Destination Operand
VAL	OPERAND1	OPERAND2

Description:

Convert the data held in OPERAND1 to numeric type and store it in OPERAND2.

Operand Data Type:

OPERAND1: string constant, string variable

OPERAND2: numeric variable

Examples:

TEMPVAR is defined a as numeric variable; then

Line No.	Operator	Source Operand	Destination Operand
1	VAL	"12345"	TEMPVAR

will convert the string "12345" to the number 12345 and store it in TEMPVAR.

System Variables

*(Asterisk)

* is a temporary storage register, representing the contents of the current data entry after using INP or INX macro. It can be either numeric or string data type

Examples:

1. When TEMPVAR is a variable, then a procedure could have:

Line No.	Operator	Source Operand	Destination Operand
1	INP	*	
2	ADD	*	TEMPVAR

which will add input data to TEMPVAR.

DATE

Stores date data with the format: MMDDYY (MM-month, DD-date, YY-year).

Examples:

1. When TEMPVAR is a variable, then a procedure could have:

Line No.	Operator	Source Operand	Destination Operand
1	MOVE	DATE	TEMPVAR

which will move DATE data to TEMPVAR.

DATES

Stores date data in the **short** format: MMDD (MM-month, DD-date).

Examples:

1. When TEMPVAR is a variable, then a procedure could have:

Line No.	Operator	Source Operand	Destination Operand
1	MOVE	DATES	TEMPVAR

which will move DATES data to TEMPVAR.

TIME

Stores time data in the format: hhmmss (hh-hour, mm-minute, ss-second).

Examples:

1. When TEMPVAR is a variable, then a procedure could have:

Line No.	Operator	Source Operand	Destination Operand
1	MOVE	TIME	TEMPVAR

TIMES

Stores time data in the **short** format: hhmm (hh-hour, mm-minute).

Examples:

1. When TEMPVAR is a variable, then a procedure could have:

Line No.	Operator	Source Operand	Destination Operand
1	MOVE	TIMES	TEMPVAR

RECORDP

A record pointer pointing at the current record of the selected FORM. It could capture the current record number for other programming purposes.

Examples:

1. When TEMPVAR is a variable, then a procedure could have:

Line No.	Operator	Source Operand	Destination Operand
1	MOVE	RECORDP	TEMPVAR

RECORDL

This moves the record pointer to the end of file. **Examples:**

1. When TEMPVAR is a variable, then a procedure could have:

Line No.	Operator	Source Operand	Destination Operand
1	MOVE	RECORDL	TEMPVAR

Demo Program List

There are several sample task (*.TSK) programs included for instruction. These programs typically reside in "C:\program files\ZEBEX\WinTaskGen.Net.v5\Demo" directory. The XXX.des is the description of XXX.tsk. For example, Loop.des is the program description of Loop.tsk.

Position.tsk

The task explains the screen orientation of the Data Collector unit.

Loop.tsk

The task explains the loop command of LST - LOP.

Input.tsk

The task explains the input command of IPF and INP.

Append.tsk

The task shows the procedure of adding a new record into a database.

Update.tsk

The task explains the definition, query and update of database.

Func_key.tsk

The task explains the echo values of function key when pressing it.