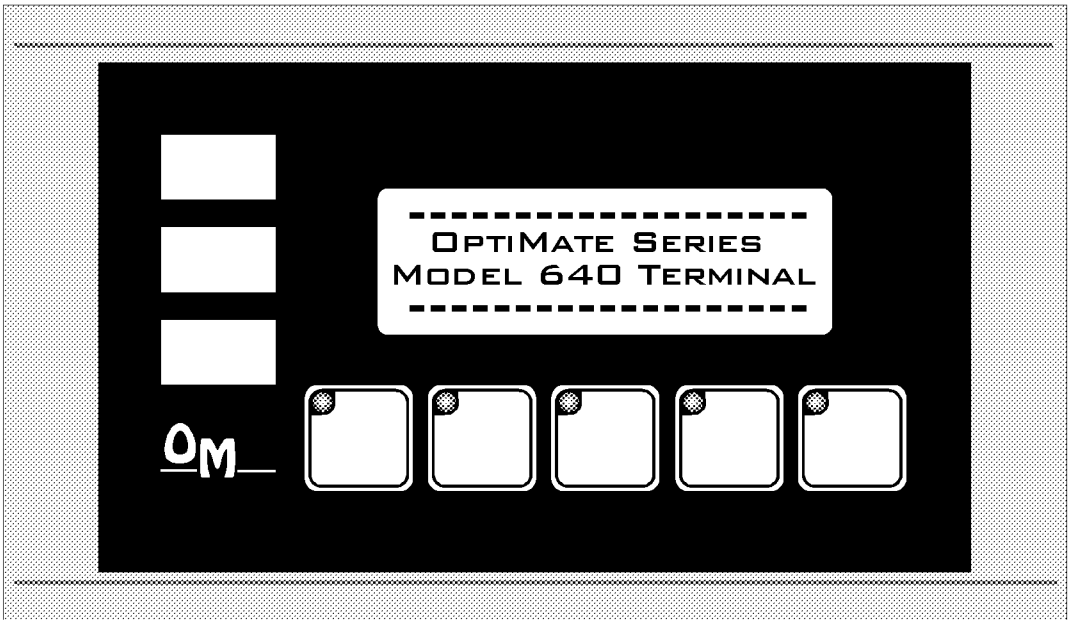


KM640 Operator Panel

The KM640 Operator Panel is a low cost/high performance man/machine interface with a broad range of operator-input and display capabilities. The panel includes a 4 line by 20 character LCD display, five function keys, 3 LED light bar indicator lamps, and the capability to display text messages, BCD, BCD Double, Binary and Floating-Point numbers.

OM-WINEDIT configuration software allows you to predefine up to 160 messages. These messages can be later selected for display by your PLC program to display status and variable data.

Function keys, indicator lamps and a company logo can be custom labeled by the user with plastic inserts. The inserts can be custom legended with text and/or graphics, and slipped into a protective pocket behind the faceplate.

The KM640 Operator Panel is part of Optimation's OptiMate® series. Each OptiMate panel is designed to connect to most PLCs with a single cable connection. OptiMate panels can be used individually, or together with any combination of other OptiMate panels.

When used with a PLC, operation is transparent to the user. Terminal functions tie directly into your PLC ladder logic program. The KM640 takes care of the rest.

Applications

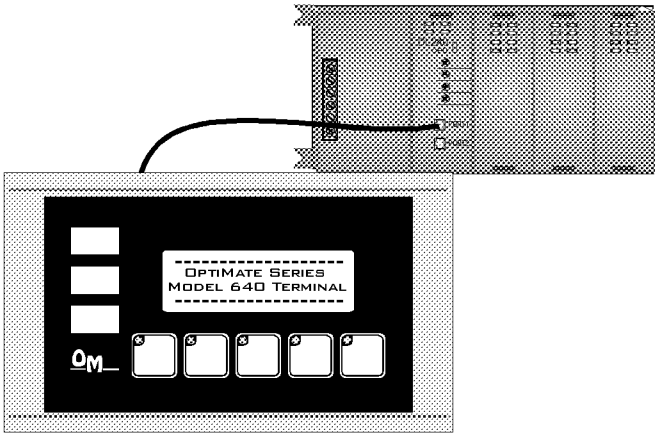
- Machine control
- Process control
- Security systems
- HVAC
- Plant monitoring/control
- PLC applications
- Microprocessor applications

Features

- 4 line x 20 character LCD
- 5 User-defined function keys
- 3 LED light bars
- User-legendable light bars and function keys
- PLC compatible
- RS232/RS422 communications
- Stand alone operation capable
- Multipanel operation capable

Contents

Configuration Options

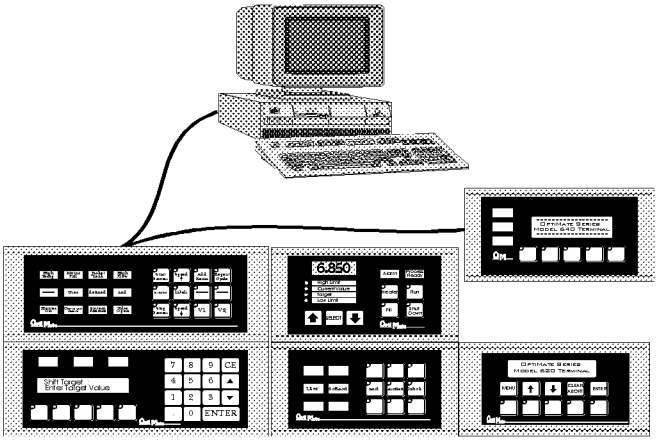


PLC Stand Alone

OptiMate panels plug directly into most PLCs. A simple cable connection allows you to interface and control the OptiMate panel via PLC data registers and ladder logic.

The KM640 Operator Terminal uses a bank of PLC registers. Complete operator interface is performed with 14 PLC registers for display message selection, indicator lamp control and function key interface. The KM640 continuously accesses these PLC registers and performs operations under ladder logic control on a real-time basis.

PLCs are slave devices on their standard communications ports. This means that a panel attached to the standard port must control the transfer of information by reading and writing the PLC registers. OptiMate panels will perform this communications for most major PLC protocols. Configuration for particular PLC protocols and interconnect cabling is covered in the following pages.



Microprocessor Based Systems

OptiMate panels can interface directly to most computers or microcontrollers. The modules communicate over either RS422 or RS232 serial communications. All that is required to interface OptiMate panels is either an RS422 or RS232 serial port and the ability to send and receive Hex numbers. The OptiMate Hex protocol, detailed in this document, allows the user to directly control panel operation and retrieve operator inputs.

Since each panel has its own unique address, up to 31 modules can be interfaced on one communications cable.

In a microprocessor-based system, the host microprocessor is the system master. The OptiMate modules are slave devices that respond to commands from the host. In the case of the KM640, these commands are messages for message display and data as well as function button status and lamp bar control.

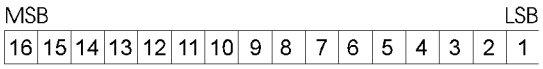
Communications over RS422 allows placement of modules anywhere within a 4000 foot cable distance. Panels can be grouped together to form a larger panel. Panels can be grouped in several clusters all on the same communications cable.

Use with a PLC

Memory Mapping

Memory mapping is a technique that "maps" the memory of an OptiMate panel into the registers of the programmable controller. By knowing where the data of a specific OptiMate panel is mapped, this data can be moved, changed or monitored using ladder logic.

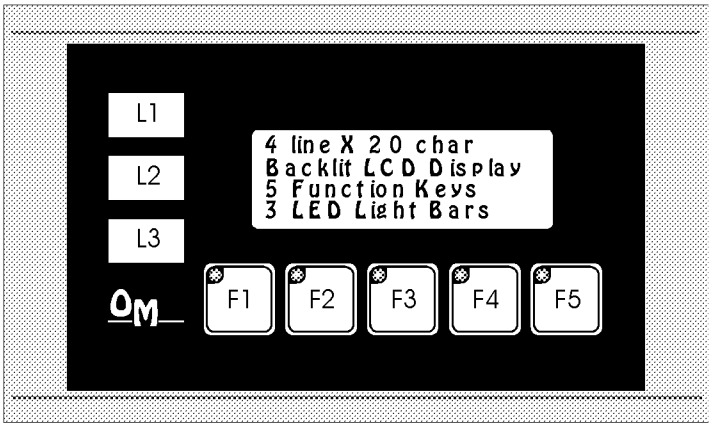
The term PLC register is used for describing the area of memory within the programmable controller that can be used for data storage. PLC registers are sometimes known as data registers or internal registers.



PLC Register

The KM640 Operator Panel uses a bank of 14 contiguous PLC registers. The register set definition is shown in the table below.

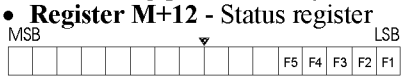
KM640 Panel PLC Register Map	
PLC Register	**Register Function**
M+0 (first register of bank)	Top line message selection
M+1	Second line message selection
M+2	Third line message selection
M+3	Bottom Line message selection
M+4	Top line data
M+5	Top line data 2 (for long BCD and floating-point)
M+6	Second line data
M+7	Second line data 2 (for long BCD and floating-point)
M+8	Third line data
M+9	Third line data 2 (for long BCD and floating-point)
M+10	Bottom line data
M+11	Bottom line data 2 (for long BCD and floating-point)
M+12	Status register
M+13	Control register



Register Definition

The following describes the function of the registers shown in the table.

- **Register M+0** - When a number from 1 - 160 is placed in this register, the predefined message associated with that number will be displayed on the top line of the LCD display.
- **Register M+1** - When a number from 1-160 is placed in this register, the predefined message associated with that number will be displayed on the second line of the LCD display.
- **Register M+2** - When a number from 1-160 is placed in this register, the predefined message associated with that number will be displayed on the third line of the LCD display.
- **Register M+3** - When a number from 1-160 is placed in this register, the predefined message associated with that number will be displayed on the bottom line of the LCD display.
- **Register M+4** - Numeric data associated with top line display (described in more detail in following paragraphs)
- **Register M+5** - For long BCD and floating-point data only.
- **Register M+6** - Numeric data associated with second line of display.
- **Register M+7** - For long BCD and floating-point data only.
- **Register M+8** - Numeric data associated with the third line of display.
- **Register M+9** - For long BCD and floating-point data only.
- **Register M+10** - Numeric data associated with bottom line of display.
- **Register M+11** - For long BCD and floating-point data only.
- **Register M+12** - Status register

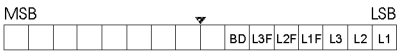


Status Register

> F1-F5 are status of the five function keys. Set to 1 when the button is active.

- **Register M+13** - Control bits

> L1-L3 - on/off control bits



Control Register

for each of the three indicator lamps. Set to 1 to turn the lamp on.

> L1F-L3F - Flash control bits for each indicator lamp. To flash, set the lamp on and the lamp flash bits to 1.

> BD - Buzzer disable. When this bit is set to 1, the buzzer that beeps every time a button is pressed will be disabled (i.e. not beep).

Operational Overview

Displaying Messages on the LCD Display

Through the OM-WINEDIT configuration software, up to 160 predefined messages can be entered and stored in the KM640. These messages are 20 characters long and can include a field for the display of numeric data.

Any predefined message can be displayed on either the top, second, third, or bottom line. The messages entered via the configuration editor are numbered 1 through 160. To display a particular predefined message on the display, simply place that message's number in the message-selection register.

For example, lets assume that we have defined message #16 as "Mary had a little .." , message #22 as "white fleeced lamb", message #25 as "Everywhere Mary went", and message #26 as "the lamb went also". If we wanted to put these four messages on the top, second, third and bottom lines of the display respectively, we would simply need to put the number 16 in register M, the number 22 in register M+1, the number 25 in register M+2 and the number 26 in register M+3.

If any number other than 1 to 160 is placed in a message selection register, the associated line will not change.

Placing Numeric Data in the Display

Certain predefined messages may incorporate a numeric data field. One numeric field per line is allowed. Messages that contain data are entered through the configuration editor with a caret symbol "^" as a place holder for each numeric digit.

An example of the use of numeric data is the message "#Widgets sold: ^^^^". Assume that this is message # 36 entered through the configuration editor. Also assume that a total of 465 widgets have been sold today. To display the current number of widgets sold on the second line of the display, you would place `36` in register M+1 and `465` in register M+6. The second line of the display would then read "#Widgets sold: 465".

Displaying Data with a Decimal Point

The KM640 panel allows you to display fixed-point numbers. Fixed point numbers are numeric values that have a known decimal point placement and are simply handled as integer values within the PLC program. The only time you use an actual decimal point is for display to the operator. An example of a fixed point number is a program that uses temperature as a control variable. Within the program, all temperatures are scaled in tenths of a degree. The values are integer. A temperature of 73.5 degrees would be 735 in a data register. For the convenience of the operator, you would want the display to include the decimal.

Fixed-point numbers are handled by simply placing a decimal point or period in the message field during configuration. In other words, the message " Temperature : ^^^.^" would be entered during configuration (say message 47). If 47 were placed in register M and the value 735 in register M+4, the display would read "Temperature : 73.5" on the top line.

Displaying BCD and Binary Numbers

Normally, numeric values to be displayed are values contained in one 16-bit register. One 16 bit register will handle values between 0 and 65535 (in binary format), or 0 to 9999 (in BCD format). For these types of numbers, register M+4 is used for numeric value for the top line, register M+6 is used for the second line, register M+8 is used for the third line and register M+10 is used for the bottom line.

Displaying "Double" Numbers

The KM640 will handle larger numeric numbers. If you select the option "BCD double" when the display message is being defined, your display will handle numbers between 0 and 99,999,999. The KM640 will use data in the register pair M+4 and M+5 for the top line. Likewise, M+6 and M+7 are used for the second line, M+8 and M+9 are used for the third line and M+10 and M+11 are used for the bottom line. The data must be in BCD format.

When placing a "BCD double" number in the display registers, the first register numerically in the sequence of two registers (M+4, M+6, M+8 or M+10) will contain the 4 least significant digits of the number. The second register in the sequence (M+5, M+7, M+9 or M+11) contains the data for the 4 most significant digits of the "BCD double" number.

If the data displayed on the third line of the panel is 46378345, the third line data registers will contain the following: (shown in BCD/Hex format)

BCD Double Data	PLC Register
M+8	8345
M+9	4637

Displaying Floating-Point Numbers

The KM640 has the capability to display Floating-Point (or Real) numbers if you select the option "Floating-Point" when the display message is being defined in the OM-WINEDIT software.

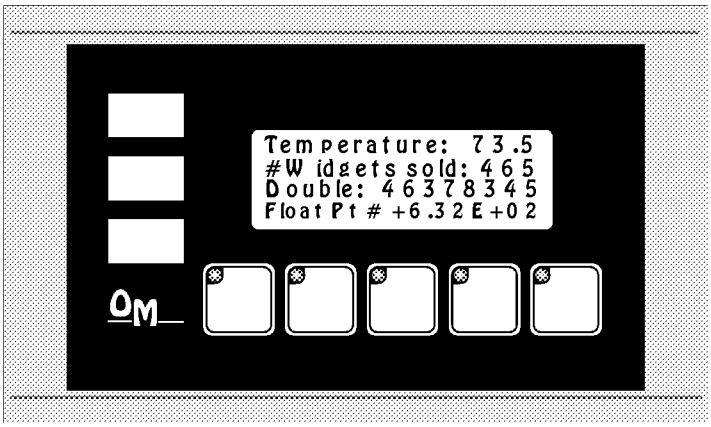
Floating-Point numbers can only be used with CPUs or a microprocessor based system using the OptiMate Hex Protocol if they support the IEEE 32-bit floating-point number format. The floating-point numbers are stored in the IEEE 32-bit floating-point format within the PLC. They always occupy 2 16-bit register locations regardless of the size of the number. Refer to the PLC manufacturer's programming documentation for more information on the IEEE 32-bit floating-point number format.

An IEEE 32-bit floating-point number has a range of -3.402823E+38 to +3.402823E+38. The KM640 will be able to display any number within that range. The panel always uses the format ±X.XXE±XX display the numbers.

The panel does not have the ability to display all the significant digits of a floating-point number, it only displays the first 3 significant digits. The KM640 does not "round" the numbers up or down, instead it truncates the remaining digits so you always see the true number. The two examples in the table below show the data contained in the PLC registers and the value displayed on the panel in its format. Notice how the data is truncated, not rounded.

PLC Registers	KM640 Display
12301.789	+1.23E+04
123.96783	+1.24E+02

The configuration of a floating-point number message is similar to any other message. First you select the message number, then you type in the text using 9 caret symbols "^" as a place holder for each of the 9 floating-point number symbols. Next, select the "Float" option for the data format.



Suppose you wanted to configure message #58 to display a floating-point number. In the OM-WINEDIT software you would select KM640 as module type. Then to configure message #58 simply select it with the mouse and type in a message in the following manner "Float Pt # ^^^^^^^^^" . Also, select floating-point as the message format. To display a number, simply move it into either the top, second, third or bottom line data registers and load the appropriate message number into the corresponding top, second, third or bottom line message selection register. If the number 632.15 is to be displayed in message #58 on the bottom line, it will be displayed as the following: "Float Pt # +6.32E+02".

Function Buttons

The KM640 contains five user definable pushbuttons. These pushbuttons can be custom labeled and used for any purpose.

The pushbuttons can be individually configured as either alternate-action or momentary pushbuttons. Alternate-action buttons alternate state each time they are pressed. Momentary buttons are active only while they are being pressed.

The status register holds the current state of each of the five pushbuttons. In a typical PLC application, these pushbuttons would be mapped to control contacts for easy ladder-logic interface.

Indicator Lamps

Three general purpose indicator lamps are located to the left of the LCD Display. These lamps can be custom labeled and used for any purpose.

There are two control bits for each lamp in the control register. One bit controls whether the lamp is ON or OFF. The other bit controls whether the lamp is flashing or steady. The lamp must be turned ON in order for the corresponding flash bit to flash the lamp ON and OFF.

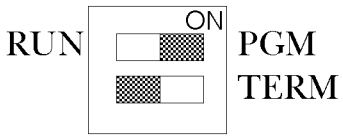
In a typical PLC application, these bits would be mapped to control coils for easy ladder-logic interface.

Configuration

Configuration of the KM640 Terminal or system of OptiMate modules is performed via an IBM PC compatible computer with the Windows operating system. Optimation supplies OM-WINEDIT software that will allow you to select module configuration, system configuration and PLC protocol definition.

If the KM640 is to be operated stand alone with a PLC, the configuration selection must be made to select the proper PLC protocol information. If it is part of a multi-panel system, each panel must be configured under the multi-panel selection in OM-WINEDIT.

Note : When configuring, always remember to set RUN/PGM to PGM (towards the "ON") before trying to download to the module.



Specific configuration of the KM640 begins with defining the block of PLC registers to be used. Next, each of the function buttons must be configured for either momentary or alternate action operation. Then each of the messages used by the PLC program must be defined.

Message definition is very straightforward and easily accomplished. All that is necessary is the following sequence.

- Select the message number to enter.
- Type the message. Up to 20 characters are allowed. Any unused characters will be filled with blanks. One numeric field per message may be defined with caret `^` characters. One decimal point or colon may be placed within the field.

Examples of Use with a Keyence PLC

Register Usage

The OptiMate OM-WINEDIT software allows you to configure the KM640 to use a block of 14 contiguous registers at a starting value that you define. For a Keyence PLC the recommended memory to use is the general purpose data memory starting at DM0000. Any block of registers within the data memory range can be used as long as the block does not conflict with anything that may be using registers in the data memory area.

The first twelve PLC registers in the block used by the KM640 panel are used for numeric information. As such they are ideally suited for the general purpose data memory registers DM0000 - DM0699. The last two registers use individual bits for control and status. These registers are better suited for the internal relay register range of memory. The solution to this minor conflict is to define the base register address in the data memory area and place a rung in your PLC program to copy the last two registers to/from internal relay registers. See the example at the bottom of the page.

The following table lists the internal relay register addresses for the various Keyence PLCs.

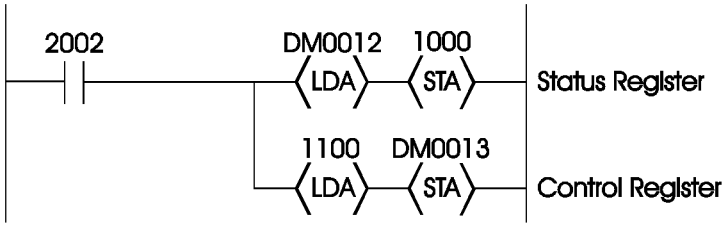
Keyence CPU	Internal Relay Register address assignment
KV10/16	1000 - 1915
KV24/40/80	1000 - 1915, 3000 - 6915
KV300	1000 - 1915, 3000 - 6915

Memory-Mapping Example

The program rung below shows a typical way of mapping the control and status registers to/from the internal relays of a Keyence PLC. In this example, the KM640 is configured for a base register address of DM0000. **The program rung should be placed in the program to copy the status register DM0012 (M+12) to internal relays beginning with 1000 and copy from the internal relays beginning with 1100 to the control register DM0013 (M+13).** The 2002 relay is ALWAYS ON in Keyence PLCs. This is placed in the rung to ensure that it happens every scan.

The location of the status and control registers depend on the base register address (the location of register M). In this case, the base register address is DM0000. Any valid internal relays can be used in the memory map. If you do not use the relays shown in the example, adjust each lamp, button, etc., to correspond to the relays that you define.

KM640 Memory Map



With the memory mapping rung shown at the bottom of the page placed in the PLC program, the status and control bits will correspond to internal relays. The register association is shown in the figure below.

Register	15 MSB	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0 LSB	bit
1000	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Status Register
1100	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Control Register

This will result in the following internal relay association for the status and control registers.

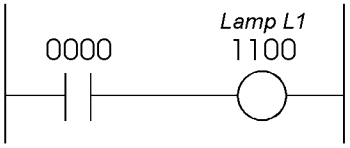
Status Register (M+12)			Control Register (M+13)	
bit	relay		bit	relay
F1	1000		L1	1100
F2	1001		L2	1101
F3	1002		L3	1102
F4	1003		L1F	1103
F5	1004		L2F	1104
			L3F	1105
			BD	1106

Programming Examples

The examples on the following pages use a KM640 connected to a Keyence PLC. The KM640 is configured for a base register address of DM0000. The following examples correspond to the memory mapping program rung shown on the previous page. That rung should be placed into the top of the PLC program.

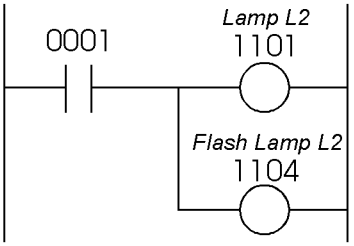
Lighting a Lamp

Lighting a lamp simply requires activating the internal relay associated with the lamp. The example on the right will light the first lamp when input 0000 is active (Remember to place the register copy rung described previously in the program).



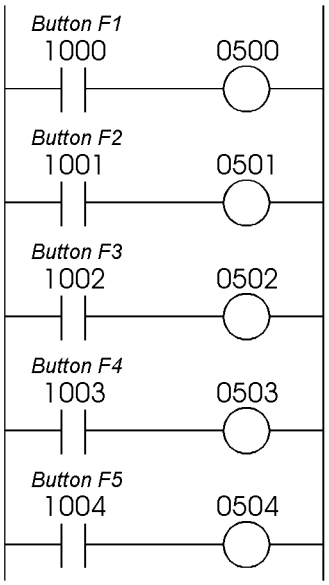
Flashing a Lamp

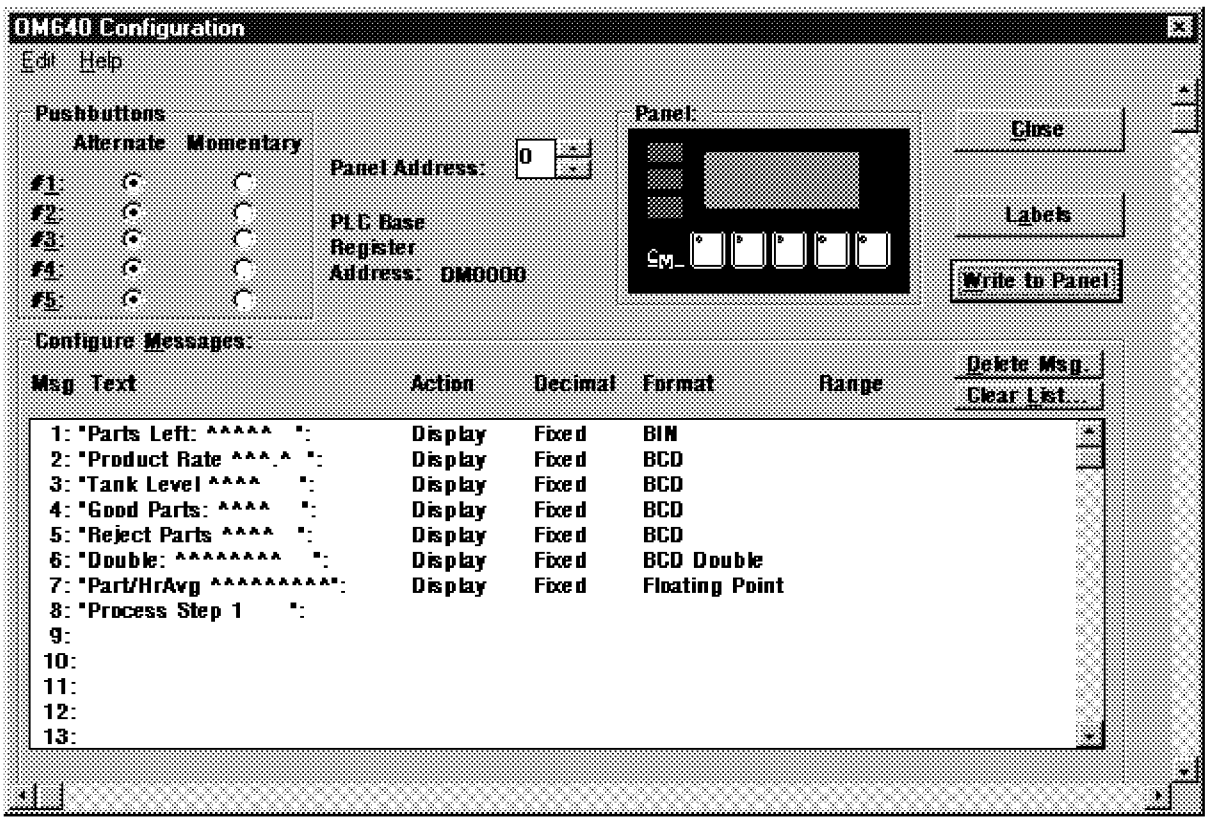
Flashing a lamp simply requires activating the lamp's associated relay (to turn the lamp ON) and the lamp's associated flash relay. The example on the right will flash the second lamp when input 0001 is active.



Using a Function Button

The five function buttons below the LCD display will appear as internal relay coils in your program (assuming the register copy rung described previously is in your program). The example on the right turns ON output 0500 when button F1 is active. It also turns ON output 0501 when button F2 is active, output 0502 when button F3 is active, output 0503 when button F4 is active, and 0504 when button F5 is active.

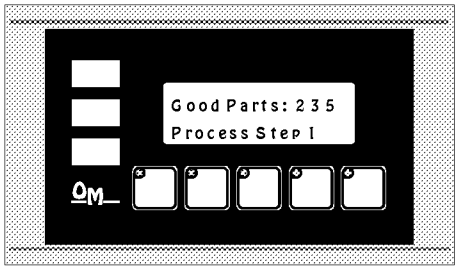




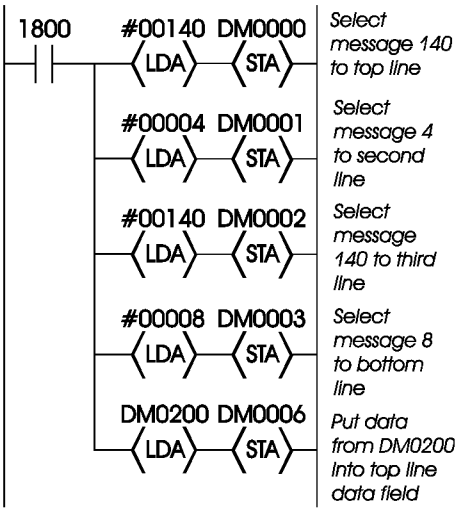
Displaying Messages on the LCD Display

Messages of various types can be configured via OM-WINEDIT and downloaded to the KM640. The message definitions shown in the figure above will be used in all of the examples that follow. **Also, assume that message #140 consists of all blanks " ".**

The example on the right shows messages being displayed on the LCD display as long as 1800 is active. The second line displays BCD message #4. The data for the second line data field is coming from DM0200. The bottom line displays text message #8. The top and third lines display message #140, which has been configured as a blank text message.

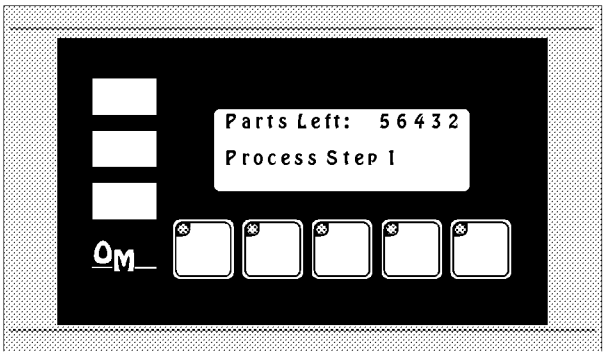


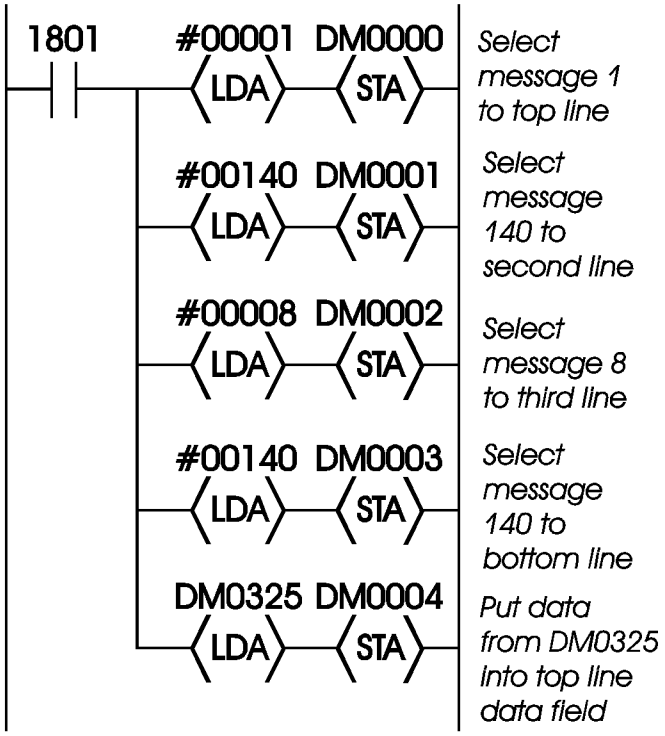
DM0000	Top line message selection
DM0001	Second line message selection
DM0002	Third line message selection
DM0003	Bottom line message selection
DM0004	Top line data
DM0005	Top line data 2 (for long BCD & floating point)
DM0006	Second line data
DM0007	Second line data 2 (for long BCD & floating point)
DM0010	Third line data
DM0011	Third line data 2 (for long BCD & floating point)
DM0012	Bottom line data
DM0013	Bottom line data 2 (for long BCD & floating point)



Displaying Binary Numbers

The example shown on the right is similar to the last example. The primary difference is that it places a Binary number on the top line of the display. The top line uses data display message #1, which has been configured as a Binary display message. The data for the top line data field is coming from DM0325. The third line displays text message #8. The second and bottom lines display message #140 which has been configured as a blank text message.

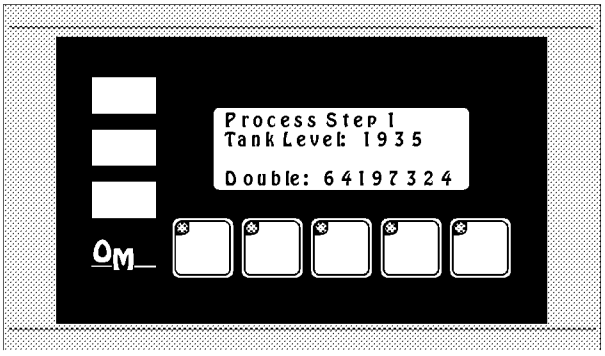


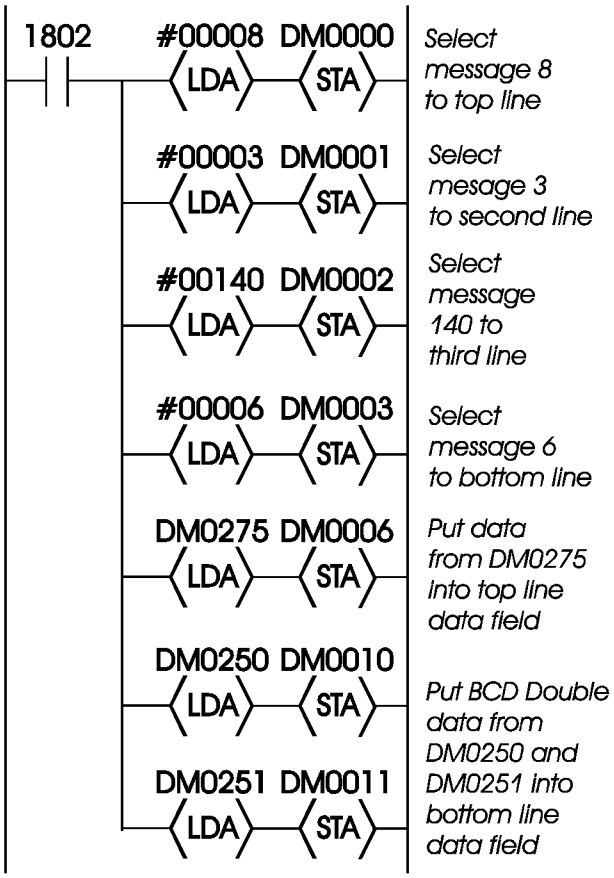


Displaying BCD Double Numbers

Keyence PLCs do not have a means of easily handling BCD double numbers, however, the KM640 has the capability to display them. For the panel to display BCD double numbers, the number has to be in the proper format within the PLC registers. The following paragraph describes the process in more detail.

The example shown on the right places a BCD double number on the bottom line of the display. The bottom line uses data display message #6, which has been configured as a BCD double display message. The data for the bottom line data field is coming from DM0250 and DM0251. The top line displays text message #8. The second line displays BCD message #3. The data for second line BCD message comes from register DM0275. The third line uses message #140 which has been configured as a blank text message.

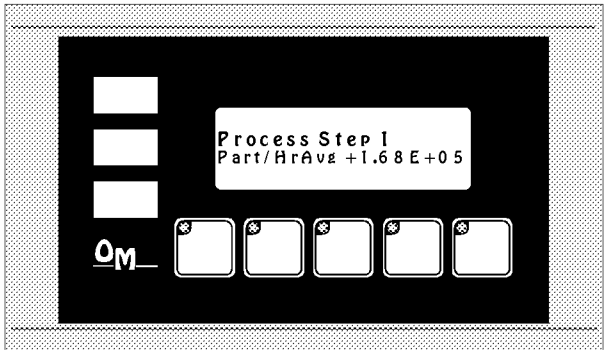


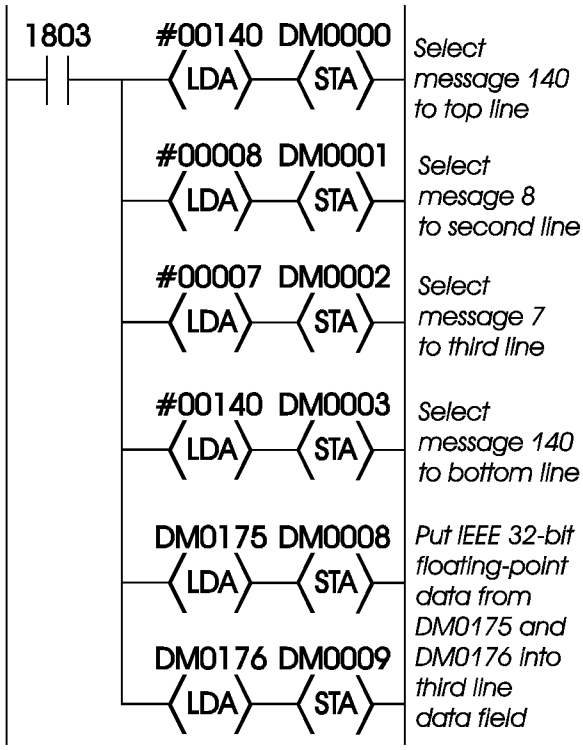


Displaying Floating-Point Numbers

Keyence PLCs do not have a means of easily handling floating-point numbers, however, the KM640 has the capability to display them. For the panel to display floating-point numbers, the number has to be in the IEEE 32-bit floating-point format within the PLC registers. The following paragraph describes the process in more detail.

The example shown on the right gets its value from two PLC registers. The third line uses data display message #7, which has been configured as a floating-point display message. Since the data is a floating-point number, it uses two consecutive 16-bit registers containing data in the IEEE 32-bit floating-point format. The two registers have to be looked at together, not individually, for you to be able to make any sense of the data. In this example, the data is loaded from DM0175 and DM0176 to the third line display registers DM0008 and DM0009. The top and bottom lines use message #140 which has been configured as a blank text message. The second line uses message #8, a text message.





Examples of Use with a Modicon PLC

Register Usage

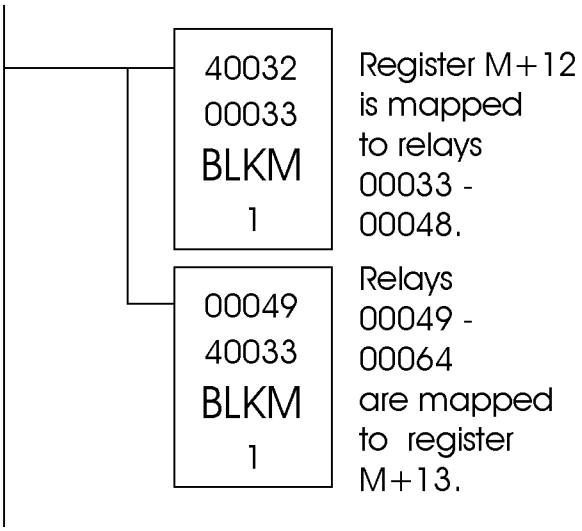
The OptiMate OM-WINEDIT software allows you to configure the KM640 to use a block of 14 contiguous registers at a starting value that you define. For a Modicon PLC the recommended memory to use is the **4x** memory area. Any block of registers between 40020 and 40617 can be used as long as the block does not conflict with anything that may be already using registers in the KM640's planned memory area.

The first twelve PLC registers in the block used by the KM640 panel are used for numeric information. As such they are ideally suited for the data memory registers 40020-40617. The last two registers use individual bits for control and status. These registers are better suited for the 0x relay register range of memory. The solution to this minor conflict is to define the base register address in the 4x memory area and place a rung in your PLC program to copy the last two registers to/from the 0x relay registers. See the example below.

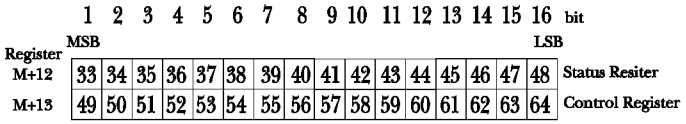
Memory-Mapping Example

The program rung below shows a typical way of mapping the control and status registers to/from the 0x relays of a Modicon PLC. In this example, the KM640 is configured for a base address of 40020. **The program rung should be placed in the program to copy the status register 40032 (M+12) to 0x relays beginning with 00033 and copy from the 0x relays beginning with 00049 to the control register 40033 (M+13).** This rung should be placed at the top of the program and it should be active every scan.

The location of the status and control registers depend on the base register address (the location of register M). In this case, the base register address is 40001. Any valid internal relays can be used in the memory map. If you do not use the relays shown in the example, adjust each lamp, button, etc., to correspond to the relays that you define.



With the memory-mapping rung shown at the bottom of the page placed in the PLC program, the status and control bits will correspond to relays in the 0x memory. The register association is shown in the figure below.



This will result in the following relay association for the status and control registers.

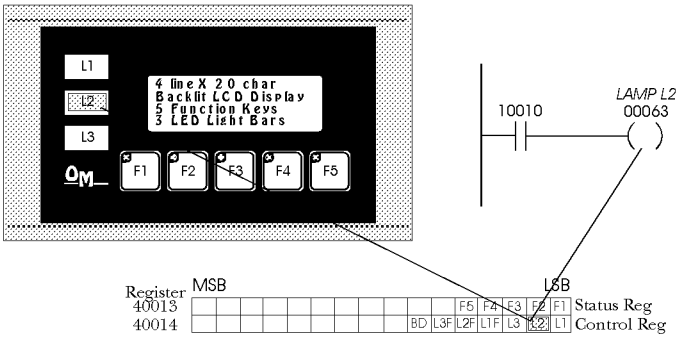
Status Register (M+12)			Control Register (M+13)	
bit	relay		bit	relay
F1	00048		L1	00064
F2	00047		L2	00063
F3	00046		L3	00062
F4	00045		L1F	00061
F5	00044		L2F	00060
			L3F	00059
			BD	00058

Programming Examples

The examples on the following pages use a KM640 connected to a Modicon PLC. The KM640 is configured for a base address of 40020. The following examples correspond to the memory mapping program rung shown on the previous page. That rung should be placed into the top of the PLC program.

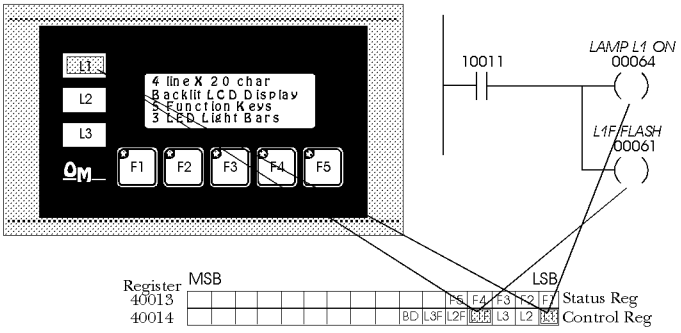
Lighting a Lamp

Lighting a lamp simply requires activating the relay associated with the lamp. The example on the right will light the second lamp (L2) when input 10010 is active.



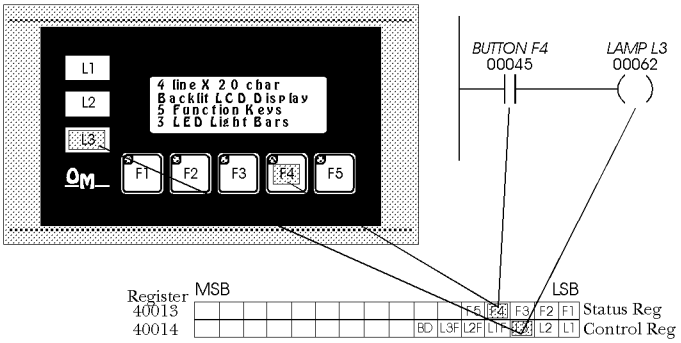
Flashing a Lamp

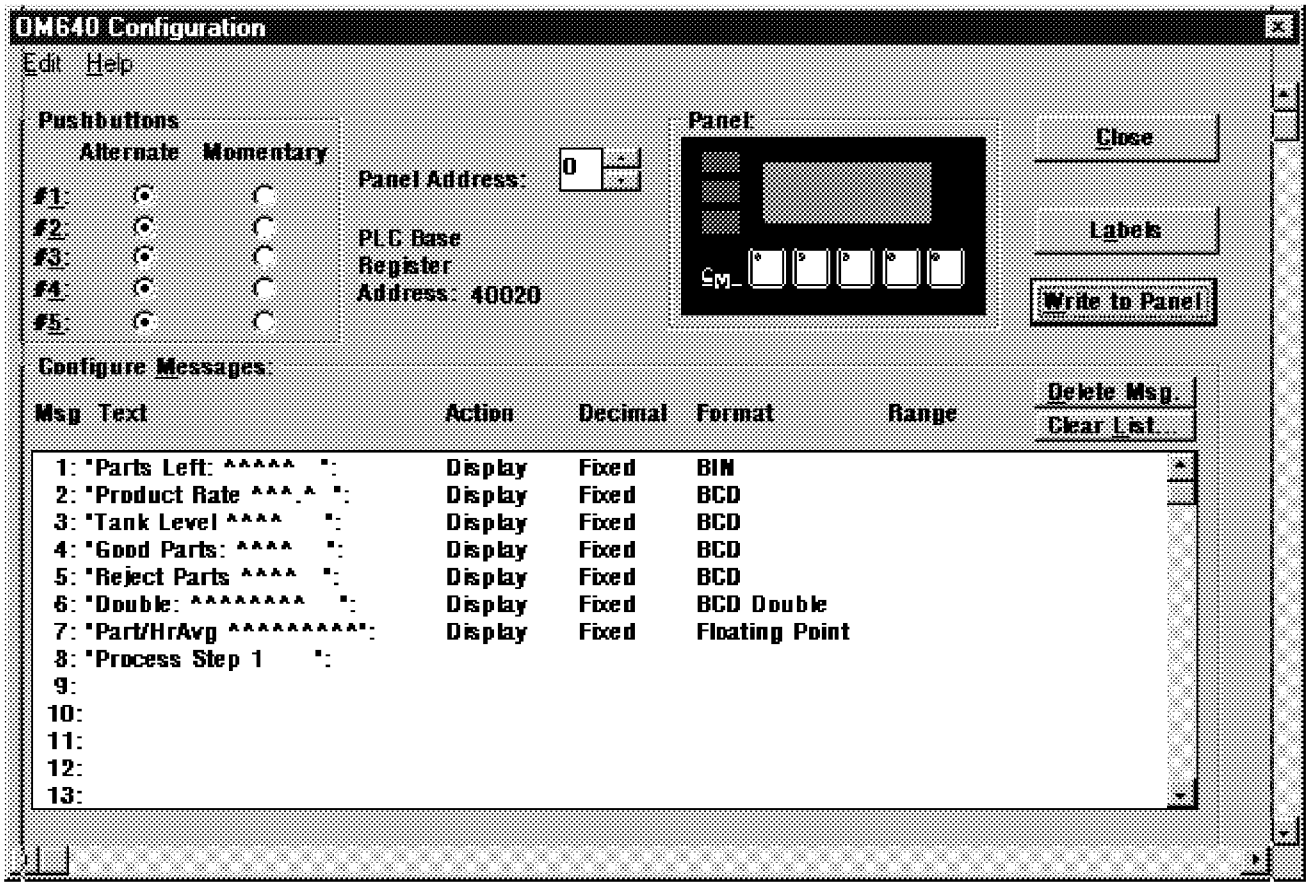
Flashing a lamp simply requires two things:
(1) Activating the relay to turn the lamp ON
(2) Activating the associated flash relay.
The example on the right will flash the first lamp (L1) when input 10011 is active.



Using a Function Button

Assuming the memory-mapping rung shown on the previous page is in your program, the five function buttons below the LCD display will appear as relay coils in your program. The example on the right turns ON lamp 3 (00062) when button F4 (00045) is active.

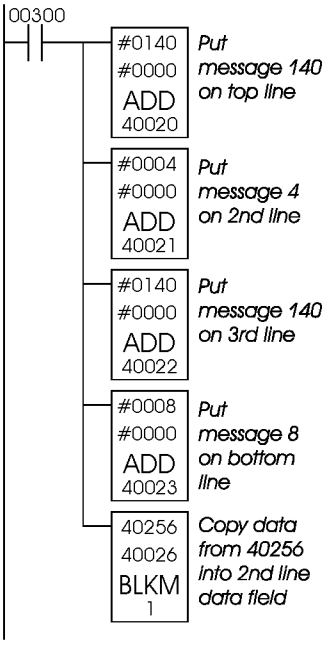


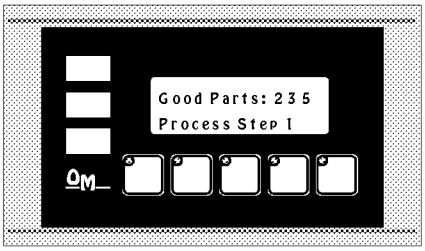


Displaying Messages on the LCD Display

Messages of various types can be configured via OM-WINEDIT and downloaded to the KM640. The message definitions shown in the figure above will be used in all of the examples that follow. **Also assume that message #140 consists of all blank characters, " ".**

The example on the right shows a couple of messages being displayed on the LCD display as long as 00300 is active. The second line displays BCD message #4. The data for the second line data field is coming from register 40256. The bottom line displays text message #8. The top and third lines display message #140, which has been configured as a blank text message.

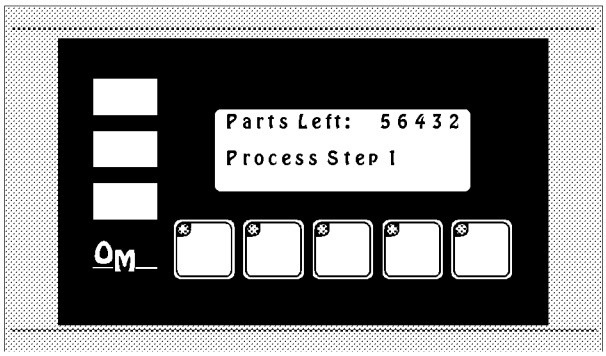


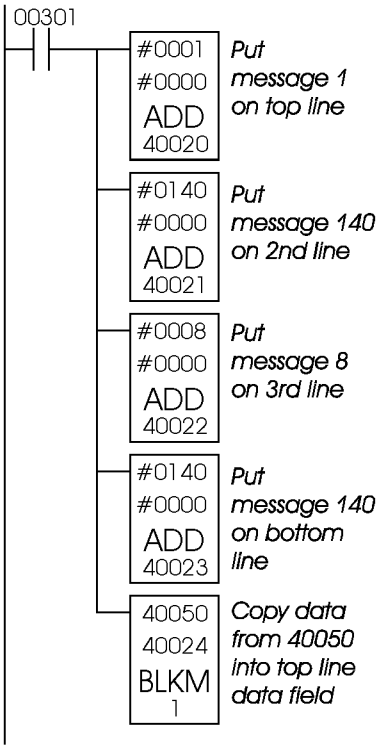


40020	Top line message selection
40021	Second line message selection
40022	Third line message selection
40023	Bottom line message selection
40024	Top line data
40025	Top line data 2 (for long BCD & floating point)
40026	Second line data
40027	Second line data 2 (for long BCD & floating point)
40028	Third line data
40029	Third line data 2 (for long BCD & floating point)
40030	Bottom line data
40031	Bottom line data 2 (for long BCD & floating point)

Displaying Binary Numbers

The example shown on the right is similar to the last example. The primary difference is that it displays a Binary number in the top line of the display. The top line displays data display message #1, which has been configured as a Binary display message. The data for the top line data field is copied from register 40050 to the top line display register (M+4) 40024. The third line displays text message #8. The second and bottom lines display message #140 which has been configured as a blank text message.

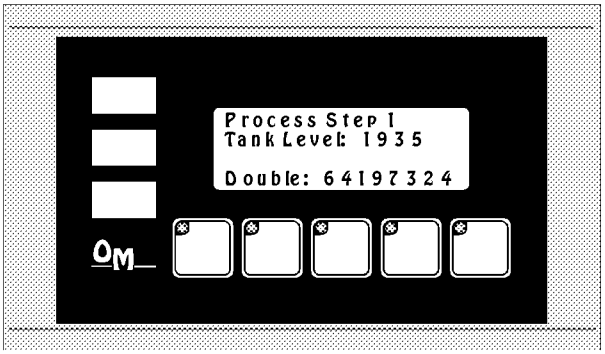


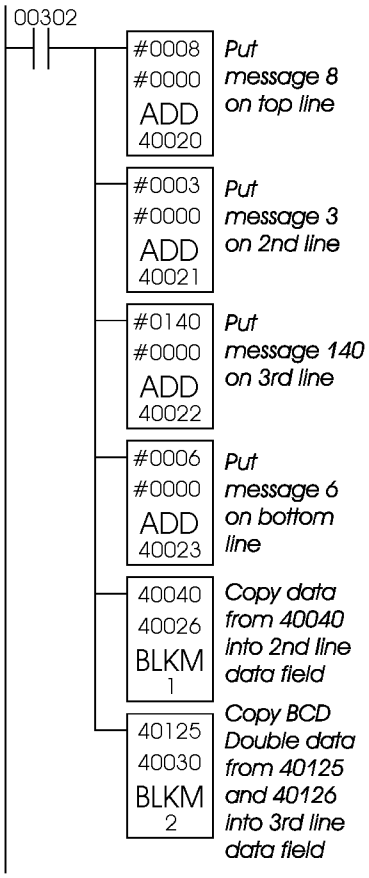


Displaying BCD Double Numbers

Long BCD numbers (up to 8 digits) can be displayed by the KM640. This number format is a standard capability for some PLCs, however, Modicon PLCs do not have an easy means of handling BCD double numbers. If the number is in the proper format in the KM640's data registers, it can display the number. Due to the limitations of the Modicon PLCs, this capability will not be commonly implemented.

The example shown on the right shows a BCD double number being displayed on the bottom line. The bottom line uses data display message #6, which has been configured as a BCD double display message. The data is copied from 40125 and 40126 to the bottom line data registers. The top line displays text message #8. The second line displays message #3. The data for second line BCD message comes from register 40040. The third line displays message #140 which has been configured as a blank text message.

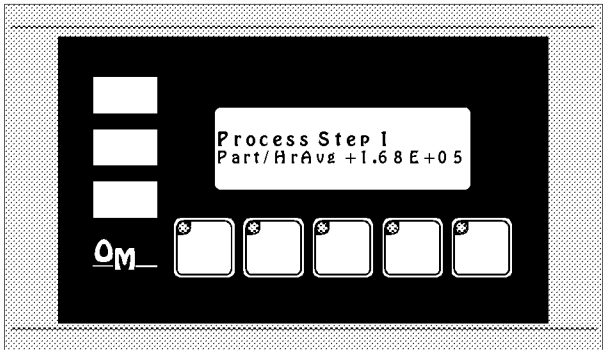


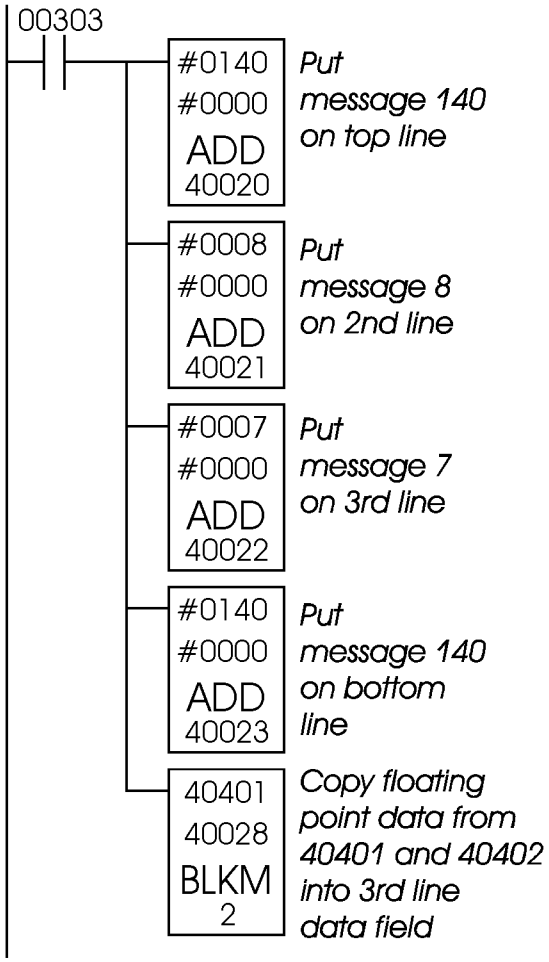


Displaying Floating-Point Numbers

Modicon PLCs have a means of handling floating-point numbers and the KM640 has the capability to display them. For the panel to display floating-point numbers, the number has to be in the IEEE 32-bit floating-point format within the PLC registers. The following paragraph describes the process in more detail.

The example shown on the right gets its value from two PLC registers. The third line displays data display message #7, which has been configured as a floating-point display message. Since the data is a floating-point number, it uses two consecutive 16-bit registers containing data in the IEEE 32-bit floating-point format. The two registers have to be looked at together, not individually, for you to be able to make any sense of the data. In this example, the data is copied from 40401 and 40402 to the third line display registers 40028 and 40029. The top and bottom lines display message #140 which has been configured as a blank text message. The second line displays message #8, a text message.





Use in a Microprocessor-Based System

OptiMate modules can interface a microprocessor-based controller over a serial link. This link can be either RS232 (for point to point) or RS422 (for multidrop or point to point). In either case the microprocessor acts as the master. It can write data to the panel or read data from the panel.

The KM640 uses the OptiMate Hex protocol for fast and easy communications. The OptiMate Hex protocol is defined in subsequent pages.

Module Address

In a microprocessor system, each panel must have its own unique address. You define this address (between 0 and 30) using the configuration editor. The panel will respond to the host only if it is properly addressed.

Communications Protocol

To use an OptiMate panel as a slave device in a microprocessor-based system, the panel must be configured for the OptiMate Hex protocol. The other options that must be set are module address, baud rate, parity and number of stop bits. If parity is set to even or odd, only one stop bit is allowed. Once selected, it must be downloaded to the panel.

Computer-Based Operation

The KM640 protocol for computer based operations is the OptiMate Hex protocol.

All of the basic functionality described for PLC operation is also available to computer or microprocessor-based applications.

The following is a synopsis of the KM640 operation as it relates to computer based applications. In certain cases, more detail is provided under the same topic for PLC operation. The details of messages involved are covered in the protocol documentation which follows.

Displaying Messages on the LCD Display

Under computer-based operation, the KM640 can be instructed to display predefined messages. Selection of a predefined message simply requires transmitting the proper command (0xA1), the message number, the appropriate line, and initial display data.

To display messages using the "Send data display message", the message text (in the ASCII equivalent) and data must be sent along with the command to send the message (0xA6), the line number, displayed data type and initial display data.

Placing Numeric Data in the Display

Any message containing caret `^` place holders will allow either the display of numeric data or numeric data entry.

Numeric data for the numeric data field can be transmitted by the host computer. This is done by transmitting the proper command (0xA7), the line number, and the data in a message field.

Indicator Lights and Function Buttons

The KM640 contains 3 general purpose indicator light bars located to the left of the LCD display. These lights can be custom labeled and used for any purpose. The computer or microcontroller can control the lights individually by turning them ON/OFF and controlling the flash state.

It also contains five user-definable function buttons. These buttons can be custom labeled and used for any purpose.

The buttons can be individually configured as either alternate action or momentary pushbuttons. Alternate action buttons alternate state each time they are pressed. Momentary buttons are active only while they are being pressed.

The General Status/Control message (0xA0) will control the current condition to each indicator light bar and the KM640 will respond with the current state of each of the five buttons.

Note: All OptiMate Hex Protocol messages must be transmitted to the KM640 in the hexadecimal number format.

KM640 OptiMate Hex Protocol

General format

STX Module function ftn_data checksum
address

Where STX = 0x02
Module address = 0 to 30
function = 0xA0 ; General status/control
0xA1 ; Select predefined message display
0xA7 ; Send data for data display message
0xA9 ; Display status request
ftn_data = data specific to the function
checksum = 8 bit sum of all characters after address until checksum

Note : Spaces are shown for readability only. There are no spaces between message fields.
0xXX denotes hexadecimal numbers.

General status/control

STX Module 0xA0 control checksum
Address
where control = control bits
bit 0 : Lamp 1 on (1)/off (0)
bit 1 : Lamp 2 on (1)/off (0)
bit 2 : Lamp 3 on (1)/off (0)
bit 3 : Lamp 1 flash on (1)/off (0)
bit 4 : Lamp 2 flash on (1)/off (0)
bit 5 : Lamp 3 flash on (1)/off (0)
bit 6 : Buzzer Disable on (1)/off (0)

response
STX term_stat display_stat checksum *if message received and processed OK*
where term_stat = terminal status
bits 0 - 4 : Button status for function buttons 1 to 5 respectively
1 = button active, 0 = button inactive
display_stat = display status (= 1 if displayed message is a display data message)
bit 0 : top line display status
bit 1 : second line display status
bit 2 : third line display status
bit 3 : bottom line display status

or
NAK if any errors
Where NAK = 0x15

Select Predefined Message

STX Module 0xA1 line mesg_no data checksum
Address
where line = top (0x00), second (0x01), third (0x02) or bottom (0x03) line
mesg_no = number of the predefined (through the configuration editor) message
(1 - 160, hex integer, i.e. 33 = 0x21)
data = 4 bytes. The format depends on the message type selected
> For an integer type data message, the first two bytes are not used (send as 0's). The Third byte is the high 8 bits of the 16 bit integer data. The fourth byte is the low 8 bits.
> For BCD data type message, the data is sent MSByte through LSByte. For a long BCD Data type message, all 4 bytes (8 digits) are used. For a regular BCD message, only the last two bytes are used.
> For floating-point data format, data sent in IEEE 32-bit floating-point format, MSB first.
- Used as display data for numeric data display message.
- Ignored for all other message types.

response
ACK *if message received and processed OK*
Where ACK = 0x06
or
NAK *if any errors in message*

Send data display message

STX Module 0xA6 line_type text data checksum
Address
where line_type = line number and data type
bits 0,1 = line number
Top line = 0x00
Second line = 0x01
Third line = 0x02
Bottom line = 0x03
bits 2,3 =unused
bits 4,5 = data type
Binary (integer) = 0x00
BCD = 0x01
BCD Double = 0x02
Floating-Point = 0x03
bits 6,7 =unused
text = 20 characters of ASCII text: message including caret characters for numeric display Data placeholders.
Example: STX 0x02 0xA6 0x23 BCD Double ^^^^^^^^ 0x43 0x62 0x78 0x59 0x??(checksum)
will display "BCD Double 43627859 " on the bottom line of the display.
data = 4 bytes. The format depends on the message type selected
> For an integer type data message, the first two bytes are not used (send as 0's). The Third byte is the high 8 bits of the 16 bit integer data. The fourth byte is the low 8 bits.
> For BCD data type message, the data is sent MSByte through LSByte. For a long BCD Data type message, all 4 bytes (8 digits) are used. For a regular BCD message, only the last two bytes are used, the first two bytes are not used (send as 0's).
> For floating-point data format, data sent in IEEE 32-bit floating-point format, MSB first.

response
ACK *if message received and processed OK*
or
NAK *if any errors in message*

Send data for data display message

STX Module 0xA7 line data checksum
Address
where line = top (0x00), second (0x01), third (0x02) or bottom (0x03) line
data = 4 bytes. The format depends on the message type selected
> For an integer type data message, the first two bytes are not used (send as 0's). The Third byte is the high 8 bits of the 16 bit integer data. The fourth byte is the low 8 bits.
> For BCD data type message, the data is sent MSByte through LSByte. For a long BCD Data type message, all 4 bytes (8 digits) are used. For a regular BCD message, only the last two bytes are used.
> For floating-point data format, data sent in IEEE 32-bit floating-point format, MSB first.

response
ACK *if message received and processed OK*
or
NAK *if any errors in message*

Display status request

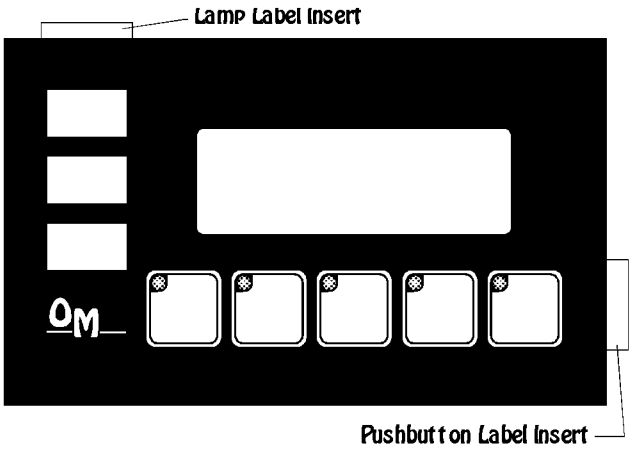
STX Module 0xA9 checksum
Address

response
STX top_line_msg line2_msg line3_msg bot_line_msg checksum *if message received and processed OK*
where top_line_msg = last predefined message selected for top line
line2_msg = last predefined message selected for second line
line3_msg = last predefined message selected for third line
bot_line_msg = last predefined message selected for bottom line

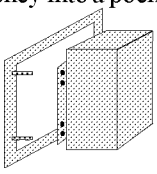
or
NAK *if any errors in message*

Set Up and Interconnect

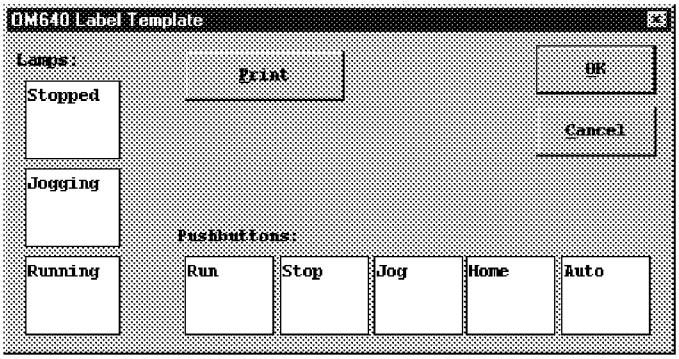
Legending the Indicator Lamps and Function Keys



Legending the KM640 panel is a relatively simple process that basically involves sliding a label transparency into a pocket in the panel overlay. Use the following procedure.



- Remove the bezel from the panel. The bezel snaps to the panel box along the top and bottom edges. Pull the bezel out and over the snaps to remove.
- Create legend transparencies. There are a number of available options for doing so. Patterns are provided on the next to last sheet of this document.
  - > Use the built in label-making capability of the OM-WINEDIT software to create labels. Either print on the transparency directly or print on paper and photocopy onto the transparency. The figure below is a screen from OM-WINEDIT which illustrates the process.



Other options include the following

- > Use a computer graphics program and a laser printer to create the transparency directly. Alternately print on paper and photocopy to a transparency
- > Use press on letters onto a transparency sheet.
- > Use a typewriter, lettering machine or press on letters to letter onto paper, then photocopy.

- Cut along outline. Slide into overlay pocket. Pushbutton legend slides in from the right. Lamp legend slides in from the top.
- Re-attach bezel. Push bezel onto box until the four snaps snap in place.

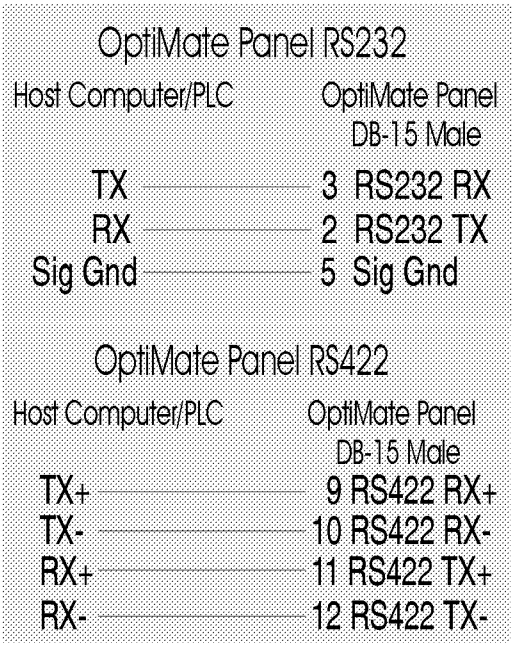
Connection to the System

OptiMate panels are designed for communication connection to system devices. The panel can be connected to a computer or PLC over the serial port (RS232 or RS422).

Connection to a Computer or PLC

Connection of an OptiMate panel to a computer or PLC can be accomplished over either an RS232 or RS422 link. RS232 is limited to one OptiMate module to one computer serial port. RS422 allows up to 31 modules to be connected to one computer port. Since PLCs are slave devices, the RS422 link for a PLC is limited to one OptiMate module.

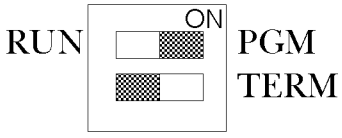
Refer to manufacturer's documentation for PLC or computer serial link connector pinouts. See the figure below for RS232 and RS422 pinouts for OptiMate modules



Interface cables for connection to several different PLCs as well as to IBM PCAT compatible ports are available.

Termination

The termination DIP switch on the back of the panel switches on a terminating resistor. **This terminating resistor does not apply to an RS232 connection, therefore, it should be in the OFF position for RS232.** In an RS422 connected system, the termination DIP switch should be ***ON*** *in the last, and only the last, panel on the cable.*



Power

The KM640 panel will operate on any DC voltage between 8 and 30VDC. Power must be connected to the terminal plug located on the back of the module. Pin 1 of the plug is the 8-30VDC (+) terminal and pin 2 is the 0VDC (-) terminal. Steady state current is listed on the specification page.

There is a brief (0.5 to 2 millisecond) power-on surge to 1.9 amps. This is typical of nearly any type of electronic equipment and is due to the initial charging of power capacitors. This surge is not normally a problem for a commercial power supply.

Configuration

Configuration Selections

OptiMate panels can be configured for the specific application by using the OM-WINEDIT Configuration Editor. The OM-WINEDIT Editor runs on any IBM PC compatible computer with Windows. It allows the user to select the exact functionality to meet application requirements.

For the KM640 module, the following are important configuration parameters.

Microprocessor-Based Systems

Decision	Selection
Single/Multi Module	Choose Single module even if the system will contain several modules. The Multi module selection applies only to systems using a communications master. In computer-based systems, each module is configured independently.
Configuration starting point	First-time configuration, start with defaults for module. Subsequent configurations can utilize disk files you create.
PLC Type	Select OptiMate Hex
Address	Each module must have a unique address.
Protocol	Select appropriate baud rate, 8 data bits, #stop bits & parity. Note that if even or odd parity selected, only 1 stop bit is available.
Messages	Define messages as required for application

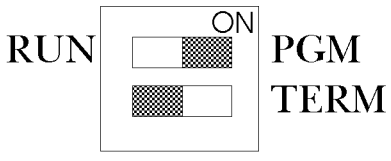
Multi-Module PLC Applications (Uses OM9001 Communications Master ... Not Available with Keyence PLCs)

Decision	Selection
Single/Multi Module	Choose Multi module
PLC Type	This applies to the Communications master. Choose appropriate type
Protocol	This applies to the OM9001 Communications master. Choose PLC address, PLC timeout, appropriate baud rate,# stop bits, # data bits, & parity. Note that if 8 data bits and even or odd parity are selected, only 1 stop bit is available.
Address	Each panel must have a unique address
Panel Protocol	The OM-WINEDIT software will automatically select the OptiMate Hex protocol for communications between the OM9001 and the panel. (This is all transparent to the user)
Buttons	Select momentary or alternate action as required for your application
Messages	Define messages as required for your application.

Single Module PLC-Based Systems

Decision	Selection
Single/ Multi Module	Choose single panel configuration
Configuration starting point	First-time configuration start with defaults for the panel. Subsequent configurations can utilize disk files you create
PLC Type	Select appropriate PLC type
Protocol	Select PLC address, PLC timeout, appropriate baud rate, # data bits, # stop bits & parity. Note that if 8 data bits and even or odd parity selected, only 1 stop bit is available
Buttons	Select momentary or alternate action as required for your application
Messages	Define messages as required for application

Configuration must be downloaded from the IBM PC compatible to each panel. This is done over the serial link. Panel must be selected for "PGM" (DIP switch in back of the module) for it to accept configuration data. After the download to the panel is complete, wait a few seconds before switching the DIP switch from "PGM" to "RUN." The DIP switch must be in the "RUN" position for the module to operate with the selected host.



Configuration using a Keyence PLC

The KM640 requires that certain parameters be set in order for it to communicate with a Keyence PLC. The parameters are PLC type, protocol, baud rate, parity, stop bits and base register address.

If you are using the KV10, KV16, KV24, KV40 or the KV80 PLCs the KM640 should always be set to the following:

KM640 Communication Parameters KV10, KV16, KV24, KV40, KV80, KV300, KV-L2	
Baud rate	9600
Parity	even
Stop Bits	1
Data Bits	8

Communications with a KV300 CPU can be accomplished by two means:

(1) direct connection to the communications port on the CPU

(2) connecting the panel to a KV-L2 Serial Interface Module

Note: If a KV-L2 Serial Interface Module is connected in the PLC system, you must use it to connect to the KM640. If you try to connect the panel to the KV300 CPU's communications port, the panel will not communicate because the CPU will not recognize the protocol.

Direct Connection to the CPU

If you are connected to the KV300 through the CPU communications port, configure the KM640 with the parameters shown in the table above.

Communication through the KV-L2

The KM640 uses the KV mode protocol to communicate with the PLC, therefore, the port that the panel is communicating with should be set for KV mode also.

If you are using a KV-L2, the following dip switch parameters apply:

Port 1		Port 2	
A1	A2	A3	A3
OFF	OFF	OFF	OFF

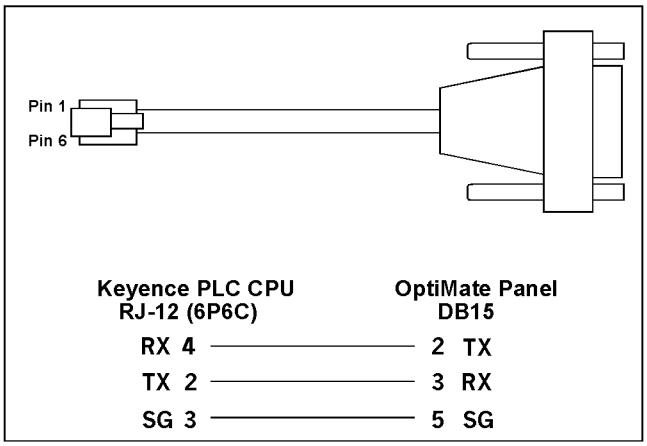
The dip switch settings shown above are dependent on the port that the KM640 is connected to. If the panel is connected to Port 1, then switches A1 and A2 should be OFF. If you have another device on Port 2 , then the switches for Port 2 should be set to the proper protocol for that device.

The KM640 uses the KV Mode protocol to communicate with Keyence PLCs. Regardless of the "B" dip switch settings, the port that the panel is connected to will automatically default to the communication parameters shown in the table above.

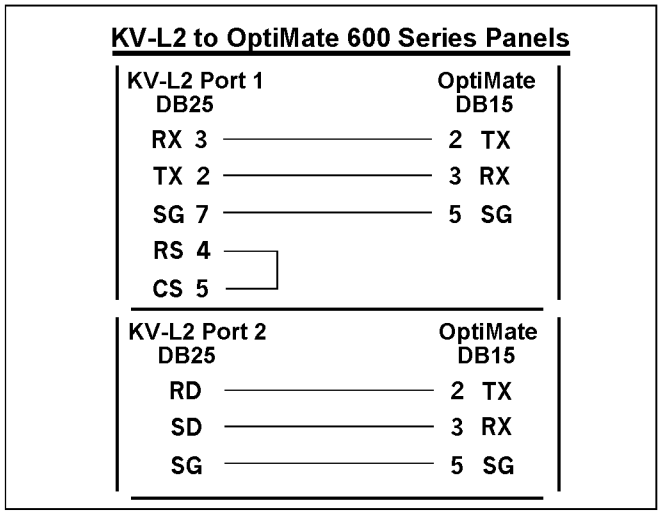
If using Port 2, ensure that the port switch is set either for RS232 or RS422, depending on the cable that you are using. If using RS232, always have the terminator dip switch OFF on the KV-L2 and on the KM640 panel.

Refer to the chapter on the KV-L2 Serial Interface Module in the Keyence User's Manual for more details.

The figure below shows the pinouts to connect a KM640 to a Keyence PLC CPU. This cable is available from Optimation.



The figure below shows the pinouts to connect a KM640 to a KV-L2 Serial Interface Module. The pinouts shown are for RS232.



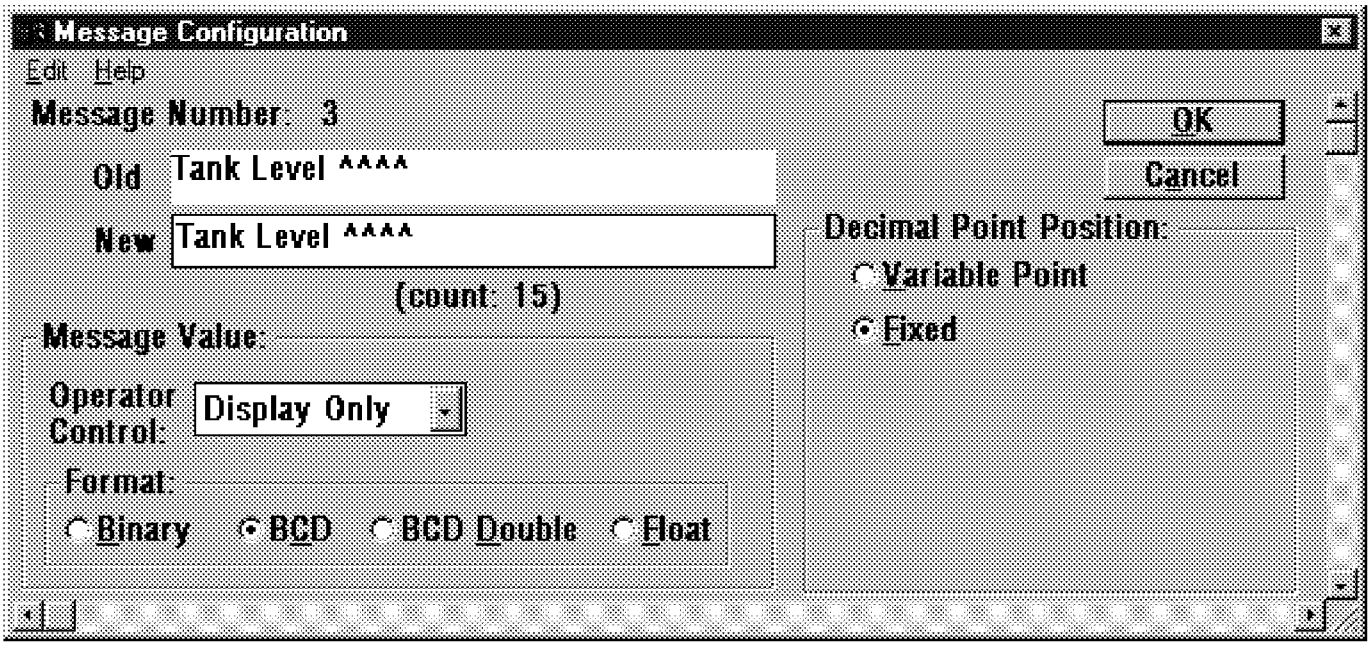
Creating Messages

The figure at right illustrates the process of creating messages for your program to use. The first step involves simply using a message template to define, on paper, each of the messages. We suggest copying the template page and using it to define all of your messages.

The next step is to use the OM-WINEDIT editor to enter the messages as defined. Remember to use carets wherever variable data is to be used. The OM-WINEDIT editor will guide you through other definable parameters, including data type, message type, etc., as shown below.

Message #	Text (20Characters Max.)
1	Enter Setpoint: ^^^^^
2	Product Rate: ^^^.^
3	Shift Target: ^^^.^

Example Message Definitions

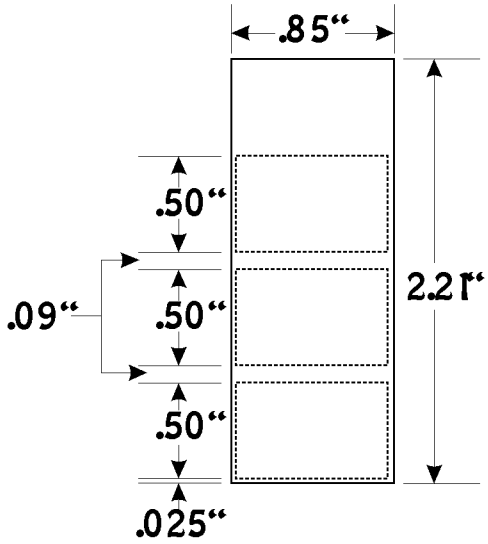


Message Definition Template

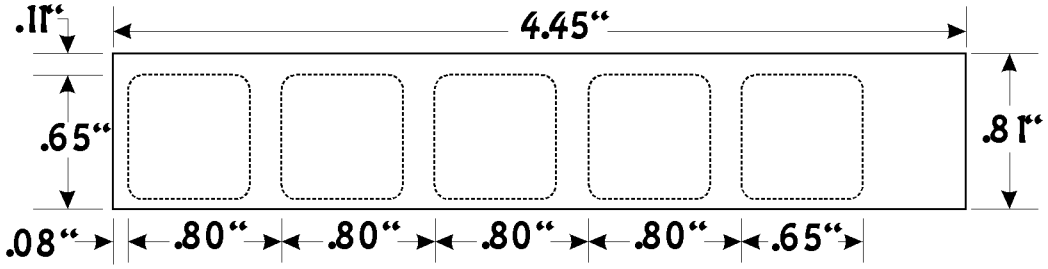
Message #	Text (20Characters Max.)

Message Definition Template

Label Templates



Indicator Lamp Label Strip Pattern

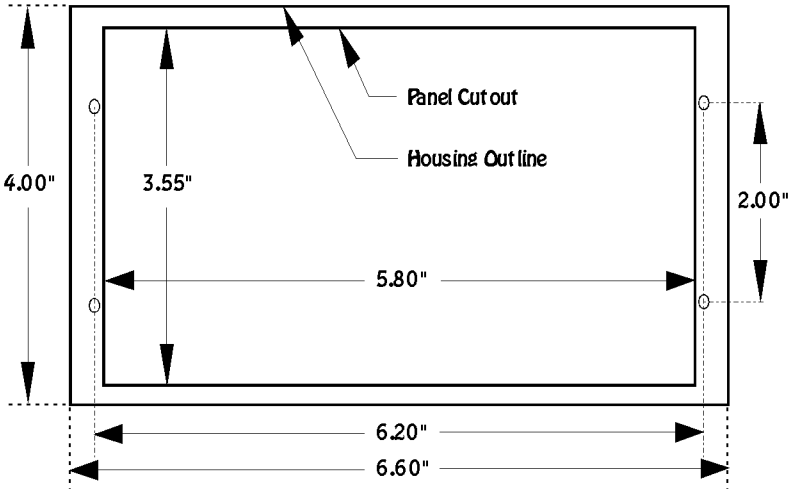


Pushbutton Label Strip Pattern

Specifications

Physical

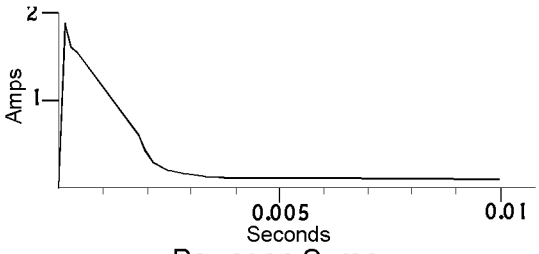
- Recessed Mount Housing 6.6"Lx4.0"H x 1.25"D
- Cutout size for above 3.55"Hx5.80"L
- Panel Fasteners : Four, 6x32 threaded studs, shown above (on ends, symmetrical about center line)
- Weight : 12 ounces
- Colors : Dark gray housing with dark gray panel. Keypad keys; White with user supplied label.
- LCD Display : 4 line x 20 character STN with LED backlight character size : 4.75mm high x 2.95mm wide
- Lamp Colors Available: Red, Yellow, Green
- Lamp Window Size: .7" X .4"



Panel Mounting Dimensions

Electrical

- Power : 8 - 30VDC @ 2.04Watts
  170mA @ 12VDC 85mA @ 24VDC
- Power on surge (see figure below)
  1.9A for 2 milliseconds maximum



Power on Surge

- Power connector : Pluggable terminal block, 2 position

Communications

- RS232 and RS422
- 4800 to 19200 baud
- Compatible with major PLC protocols
- Microprocessor compatible OptiMate Hex protocol
- 15 pin female 'D' shell connector

Communications Failure Operation

Should the panel (when not selected for configuration) ever fail to communicate successfully for a period of 12 seconds, the 3 indicator lamps and the LEDs inset in the corner of the 5 buttons will all simultaneously flash at a rapid rate.

Environmental

- Enclosure - NEMA 4 (when properly installed)
- Temperature - 0 to 50 C
- Humidity - 95% non-condensing

Message Types (160 user-defined messages available)

- General Text message
- Data display message (one data value per line)

Numeric Types & Values

- Integer
- Fixed Point
- BCD (Values between 0 & 9999; with appropriate decimal placement)
- BCD Double (Values between 0 & 99999999 with appropriate decimal placement)
- Binary (Values between 0 & 65535 with appropriate decimal placement)
- Floating-Point (Values between -3.402823E+38 to +3.402823E+38 in the format of ±X.XXE±XX)