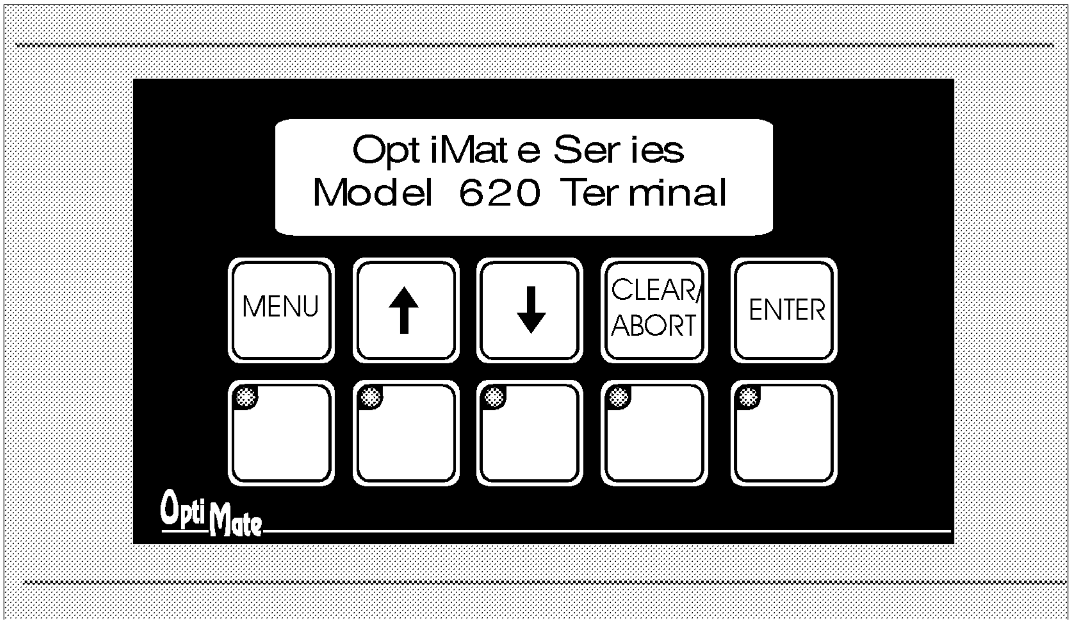


KM620 Operator Panel

The KM620 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 2 line by 20 character LCD display, arrow adjustment data entry, five function keys and menu tree capability.

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

Function keys 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 KM620 Operator Panel is part of Optimation's OptiMate® series. Each OptiMate panel is designed to connect to a microprocessor or 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 microprocessor system, simple communications over either an RS232 or RS422 communication cable allows the microprocessor to directly control the LCD display as well as read numeric data entry and function key status.

When used with a PLC, operation is transparent to the user. Terminal functions tie directly into your PLC ladder logic program. The OM620 takes care of the rest.

Applications

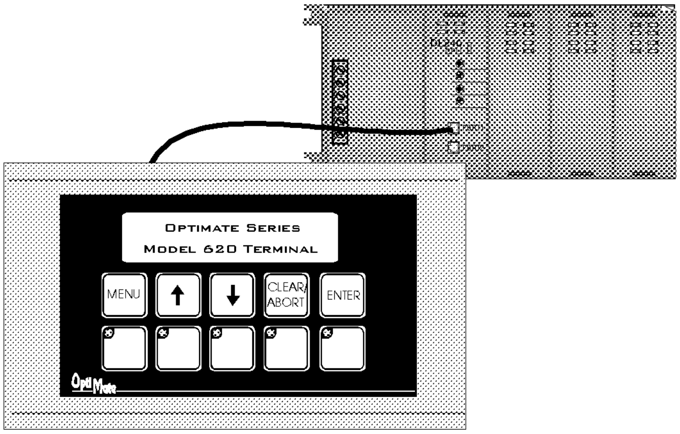
- Machine control
- Process control
- Security systems
- HVAC
- Plant monitoring/control
- PLC applications
- Microprocessor applications

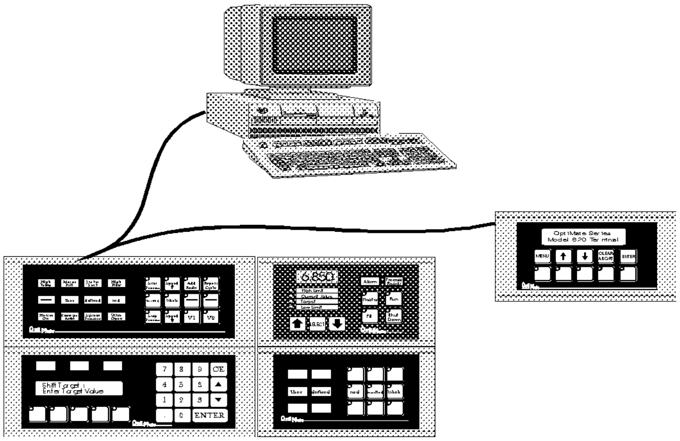
Features

- 2 line x 20 character LCD
- Numeric arrow adjustment
- 5 User-defined function keys
- Menu tree capability
- PLC compatible
- RS232/RS422 communications
- Stand alone operation capable
- Multimodule 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 KM620 operator terminal uses a bank of PLC registers. Complete Operator interface is performed with 9 PLC registers for display message selection, data entry and function key interface. The KM620 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 communicate most major PLC protocols. Configuration for particular PLC protocols and interconnect cabling is covered in the following pages.

Microprocessor Based Systems

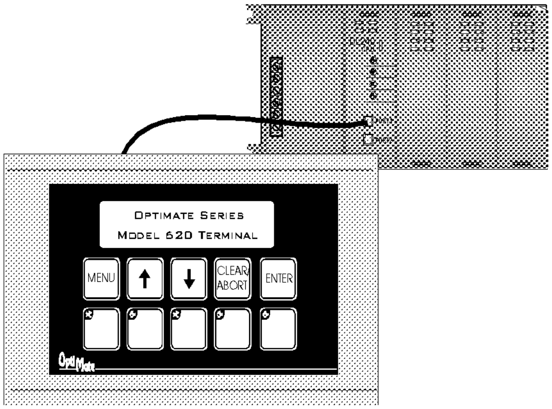
OptiMate panels can interface directly to most computers or microcontrollers. The modules communicate over either RS422 or RS232 serial communications. All that is required to interface OptiMate panels is a serial port and the ability to transmit and receive Hex numbers. The OptiMate Hex protocol, detailed 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 KM620, these commands are messages for display and data initialization as well as function button status and data entry retrieval.

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.

MSB															LSB
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

PLC Register

The KM620 Operator Panel uses a bank of 9 contiguous PLC registers. The register set definition is shown in the table below.

KM620 Panel PLC Register Map

PLC Register	Register Function
M+0 (first register of bank)	Top line message selection
M+1	Bottom Line message selection
M+2	Top line data
M+3	Top line data 2 (for long BCD and floating-point)
M+4	Bottom line data
M+5	Bottom line data 2 (for long BCD and floating-point)
M+6	Function selection
M+7	Status register
M+8	Control register

Register Definition

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

- Register M - 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 bottom line of the LCD display.
- Register M+2 - Numeric data associated with top line display (described in more detail in following paragraphs)
- Register M+3 - For long BCD and floating-point data only. Numeric value is the four most significant BCD digits of data for the top line.
- Register M+4 - Numeric data associated with bottom line of display.
- Register M+5 - For long BCD and floating-point data only. Value is the four most significant BCD digits of data for the bottom line.
- Register M+6 - The function number selected through the menu tree.
- Register M+7 - Status register

MSB															LSB
				↓	↑	ENT	MA	AB	FS	DA	F5	F4	F3	F2	F1

Status Register

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

> DA - Data available. Associated with data entry. Set to 1 when new data has been entered.

> FS - Function Selected. Indicates that a function has been selected through use of the menu tree.

> ENT - ENTER button is active. (Not necessarily Data Entry - see DA bit).

> AB - Abort key active.

> MA - Menu operation is currently active.

> Up Arrow - Up arrow key is active.

> Down Arrow - Down arrow key is active.

- Register X+8 - Control bits

MSB															LSB
												BD	DAK	MR	ME

Control Register

> ME - Menu Enable. Must be set for menu operation to be enabled. (Normally, your control program would clear this bit when a function is in process).

> MR - Menu Return. Used at the end of a function. If set, will cause the menu to return to the same point in the menu tree as when the function started. If not set, function complete will not return directly into menu.

> DAK - Data acknowledge. This bit is used for repeat data entry into the same message. Setting to 1 acknowledges data entered from the keyboard as accepted by the PLC program. The 620 Terminal will clear the data on the screen message, clear the DA bit and allow new data entry. This bit must be cleared after DA (status register) is cleared.

> 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 KM620. These messages are 20 characters long and can include a field for the display and/or entry of numeric data.

Any predefined message can be displayed on either the top 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 .." and message #22 as "white fleeced lamb". If we wanted to put these two lines on the top and bottom lines of the display respectively, we would simply need to put the number 16 in register M and 22 in register M+1.

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. This field may be either a display data field or a data entry field. 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 bottom line of the display, you would place `36` in register M+1 and `465` in register M+4. The bottom line of the display would then read "# widgets sold: 465".

Displaying Data with a Decimal Point

The KM620 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+2, 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 this type of number, register M+2 is used for numeric value for the top line and register M+4 is used for the bottom line.

Displaying "Double" Numbers

The KM620 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 KM620 will use data in the register pair M+2 and M+3 for the top line. Likewise, M+4 and M+5 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+2 or M+4) will contain the 4 least significant digits of the number. The second register in the sequence (M+3 or M+5) contains the data for the 4 most significant digits of the "BCD double" number.

If the data displayed on the top line of the panel is 92345678, the top line data registers will contain the following: (shown in BCD/Hex format)

BCD Double Data	PLC Register
M+2	5678
M+3	9234

Displaying Floating-Point Numbers

The KM620 has the capability to display Floating-Point (or Real) numbers if you select the option "Float" when the display message is being defined in the OM-WINEDIT software.

Floating-point numbers can only be used with CPUs that 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.

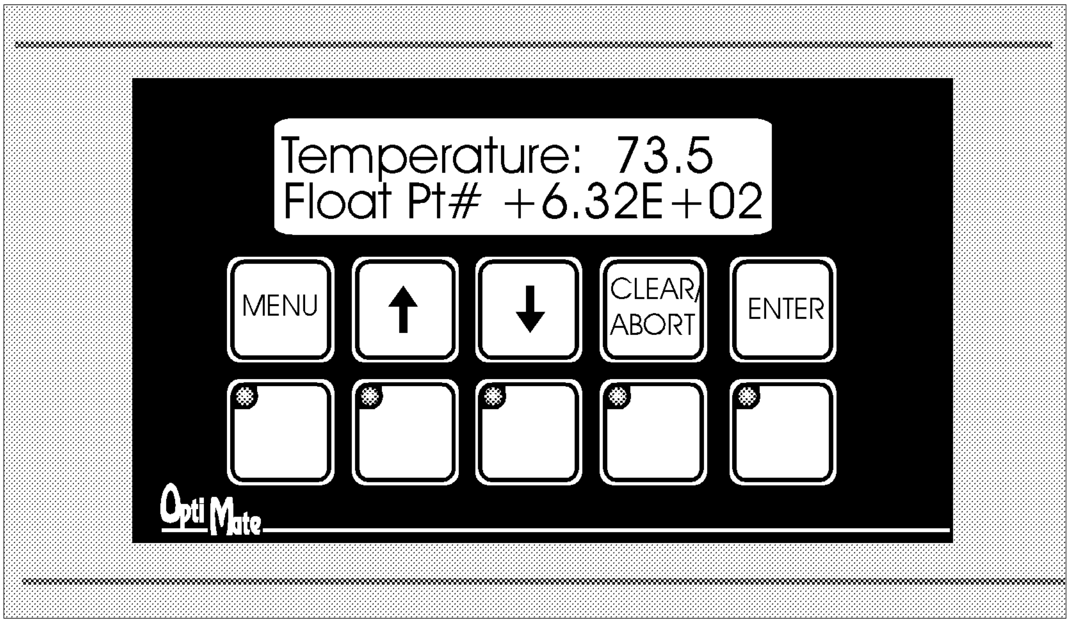
An IEEE 32-bit floating-point number has a range of -3.402823E+38 to +3.402823E+38. The KM620 will be able to display any number within that range. The panel always uses the format ±X.XXE±XX to display the numbers.

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 KM620 does not "round" the numbers up or down, instead it truncates the remaining digits. 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	KM620 Display
12301.789	+1.23E+04
123.96783	+1.23E+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 KM620 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 or bottom line data registers and load the appropriate message number into the corresponding top or bottom line message



selection register. If the number 632.15 is to be displayed in message #58, it will be displayed as the following:
"Float Pt # +6.32E+02".

Numeric Data Entry

Numeric data can be entered through the adjustment arrow keys. To do so, the message must be marked for data entry via the configuration editor.

> ***Note : Only one data entry message may be active at a time. If data entry is selected simultaneously for both lines, unexpected operation may occur.***

All of the numeric features described for data display apply to numeric data entry. This includes the definition of the caret symbol "^" place holder within the message as well as "double" number types.

Data Entry/Adjustment with the Arrow Keys

A message can be set up for arrow adjustment through the OM-WINEDIT software. To do so, define the message with a numeric field. Select arrow adjustment. If there are limits, select and enter minimum and maximum values.

When the data screen is required in the program, place the number of the preconfigured message in the selection register and its current value in the associated data register. The digits marked by carets will initially display the current value.

As the operator presses the up or down arrow key, the numeric value will increment or decrement respectively. As it is adjusted, the value will be continuously updated in the PLC data register. **When adjustment is complete, the operator will press the ENTER button. When this happens the data available (DA) status bit will be set. The DA bit will remain set until a new message number is placed in the message selection register (M or M+1) or the DAK bit is set.**

Adjustment of data will be confined to the limits defined through the configuration editor. Data also will be limited by the number of digits defined by carets in the message.

Example of Arrow Adjustment of Numeric Data

Suppose that your automatic banana peeler has a peel rate that can be adjusted between 1 and 50 bananas per second. With the configuration editor you define message 15 as "Set Peel Rate:^^.^". You would also select arrow adjustment and range limits of 10 to 500 (in tenths).

When this message is used, your PLC program would put 15 in register M (or M+1) and the current peel rate value in M+2 (or M+4). If the current peel rate was 5.7 bananas per second, the display would read "Set Peel Rate: 5.7". Pressing the arrow keys would adjust the value up or down while continuously writing the value to data register M+2 (or M+4). When the adjustment is complete, the operator would press ENTER. The KM620 will then set DA. DA will remain set until a new message is selected via M (or M+1) or the DAK bit is set.

Function Buttons

The KM620 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 (M+7) 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.

Menu Tree Operation

The KM620 terminal is designed to allow you to create a menu tree for function selection. The menu tree allows for interactive selection of a required function operation from a "menu" or list of options.

The KM620 allows up to four levels of menu. Each menu selection can be either a function or the path to the next lower menu level (sub-menu). With this type of "tree" arrangement, you can construct an application menu that goes from general to specific. You can also place frequently used menu selections on the top layer and infrequent selections on lower layers.

The graphic below is an example of a typical menu tree.

```
Process Part type 1
Process Part type 2
Manual Control
    Turn Pump On
    Turn Pump Off
    Turn Heater On
    Turn Heater Off
    Turn Mixer On
    Turn Mixer Off
Setpoint Entry
    Tank Level Setpoints
        Set Minimum Level
        Set Maximum Level
        Set Low Alarm Level
        Set High Alarm Level
    Temperature Settings
        Temperature Setpoint
        Set Low Alarm Temp
        Set High Alarm Temp
```

Sample Menu Tree

The example shown has three menu layers. The top level has four selections. When "Process Part type 1" is selected, the associated function number will be placed in X+6 and the FS flag set. "Manual Control" is a sub-menu which will bring up the next menu. When selected, each item under "Manual Control" is a function which will place a function number in X+6 and set the FS bit. The "Setpoint Entry" selection from the top level menu will bring up a second level menu of two items. Each second level item will in turn bring up another level. The items at the bottom level will perform actual function selections.

When a function is selected through the menu tree, the PLC program should start the function process. **Once a function is selected, the menu selection is locked onto the display until the PLC program clears the ME flag.** When the PLC program clears the ME flag, control of the display will return to your program. When the function is complete, you may return to the same point in the menu by setting the ME and MR flags. To require the user to start the menu selection process from the beginning ("Process Part type 1" in this case), don't set the MR flag.

The menu enable is totally under PLC program control. If for any reason while in the menu you need to take control of the display from the menu to the program, just clear the ME bit.

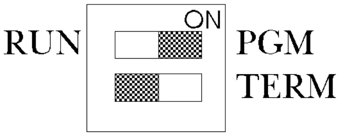
Each menu tree message takes up one of the 160 available total messages.

Configuration

Configuration of the KM620 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 KM620 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 the RUN/PGM DIP switch to PGM (towards the "ON"). Always wait for the module to reset (approximately 2 seconds) before downloading the configuration to the module.



Specific configuration of the KM620 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.
- If the message has a field for numeric arrow adjustment, select arrow adjustment. Select and enter minimum and maximum values if they apply.
- If applicable, select the data type for the message.

Examples of Use with a Keyence PLC

Register Usage

The OptiMate OM-WINEDIT software allows you to configure a module to use a block of 9 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 it does not conflict with any devices that may be using registers in the data memory area.

The first seven PLC registers in the block used by the KM620 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

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.

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

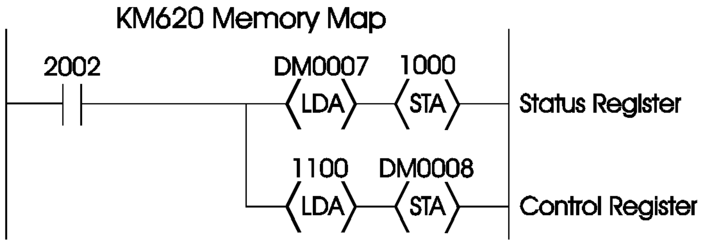
This will result in the following internal relay association for the status and control registers.

Status Register (M+7)			Control Register (M+8)	
Bit	Relay		Bit	Relay
F1	1000		ME	1100
F2	1001		MR	1101
F3	1002		DAK	1102
F4	1003		BD	1103
F5	1004			
DA	1005			
FS	1006			
AB	1007			
MA	1008			
ENTER	1009			
Up arrow	1010			
Down arrow	1011			

Memory Mapping Example

The program rung on the right shows a typical way of mapping the control and status registers to/from the internal relays of a Keyence PLC. In this example, the KM620 is configured for a base register address of DM0000. **The program rung should be placed in the program to copy the status register DM0007 (M+7) to internal relays beginning with 1000 and copy from the internal relays beginning with 1100 to the control register DM0008 (M+8).** 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 depends 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.

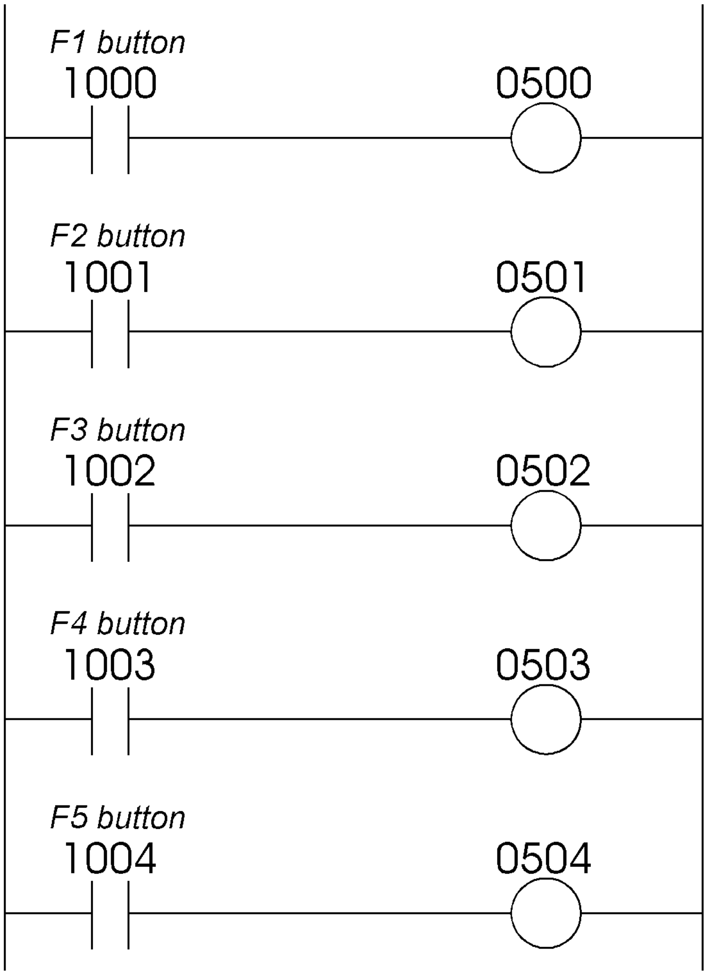


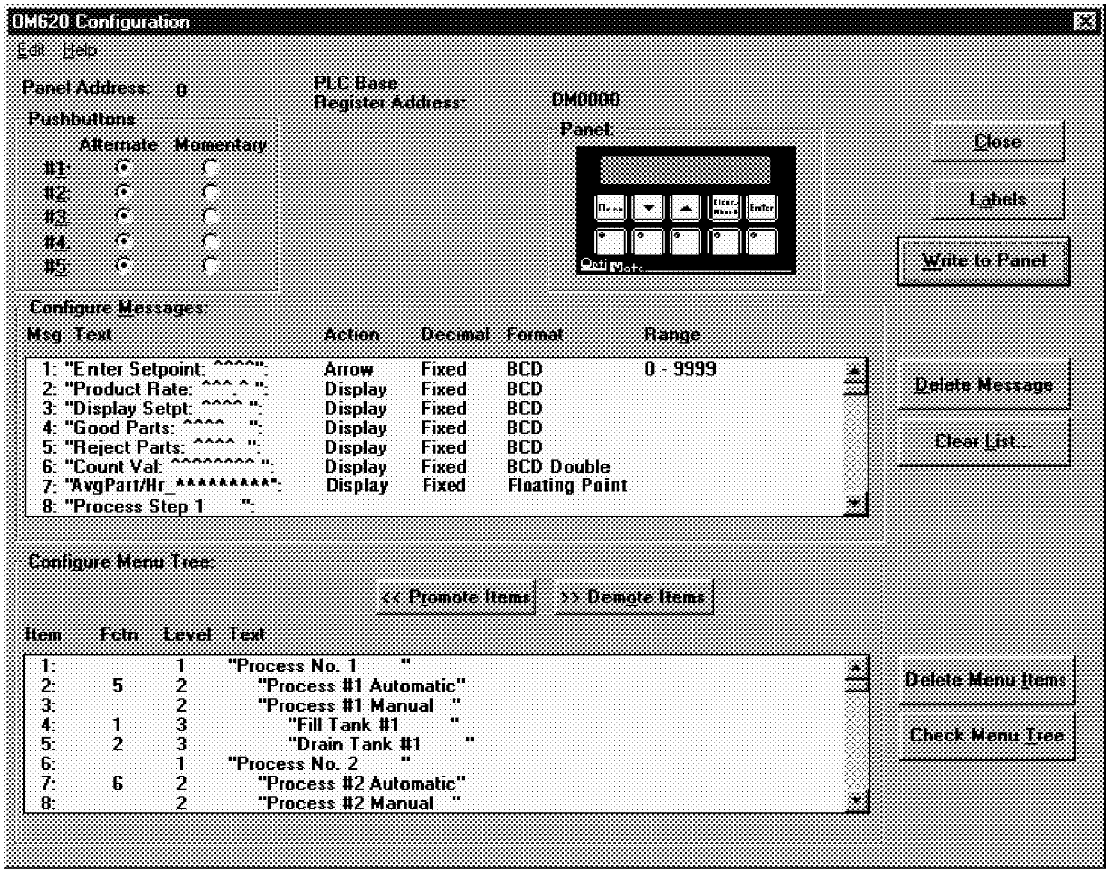
Programming Examples

The examples on the following pages use a KM620 connected to a Keyence PLC. The KM620 is configured for a base 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.

Using a Function Button

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





Panel Address: 0

PLC Base Register Address: DM0000

Pushbuttons

	Alternate	Momentary
#1:	●	○
#2:	●	○
#3:	●	○
#4:	●	○
#5:	●	○

Configure Messages:

Msg Text	Action	Decimal	Format	Range
1: "Enter Setpoint: ^^^^":	Arrow	Fixed	BCD	0 - 9999
2: "Product Rate: ^^^.^ ":	Display	Fixed	BCD	
3: "Display Setpt: ^^^^ ":	Display	Fixed	BCD	
4: "Good Parts: ^^^^ ":	Display	Fixed	BCD	
5: "Reject Parts: ^^^^ ":	Display	Fixed	BCD	
6: "Count Val: ^^^^^^^^^ ":	Display	Fixed	BCD Double	
7: "AvgPart/Hr_*********":	Display	Fixed	Floating Point	
8: "Process Step 1 ":				

Configure Menu Tree:

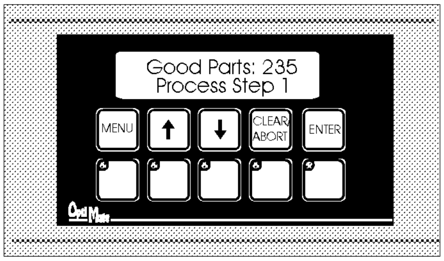
Item	Fctn	Level	Text
1:		1	"Process No. 1 "
2:	5	2	"Process #1 Automatic"
3:		2	"Process #1 Manual "
4:	1	3	"Fill Tank #1 "
5:	2	3	"Drain Tank #1 "
6:		1	"Process No. 2 "
7:	6	2	"Process #2 Automatic"
8:		2	"Process #2 Manual "

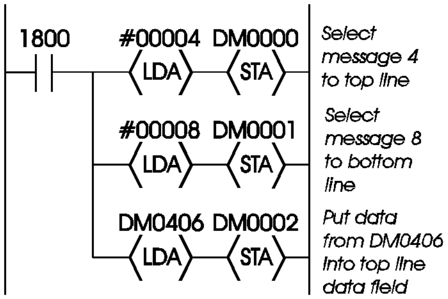
Displaying Messages on the LCD Display

Messages of various types can be configured via OM-WINEDIT and downloaded to the KM620. The message definitions shown in the figure above will be used in all of the examples that follow.

The example on the right shows a couple of messages being displayed on the LCD display. This example will work for BCD or Binary data. The top line uses data display message 4. The data for the data field is coming from DM0406. The bottom line is text message 8.

Keyence PLCs do not have an easy way of manipulating BCD Double or IEEE 32-bit floating-point numbers. Examples on the next page show how to display these types of numbers *if* the data is in the proper format within the PLC.





Select message 4 to top line

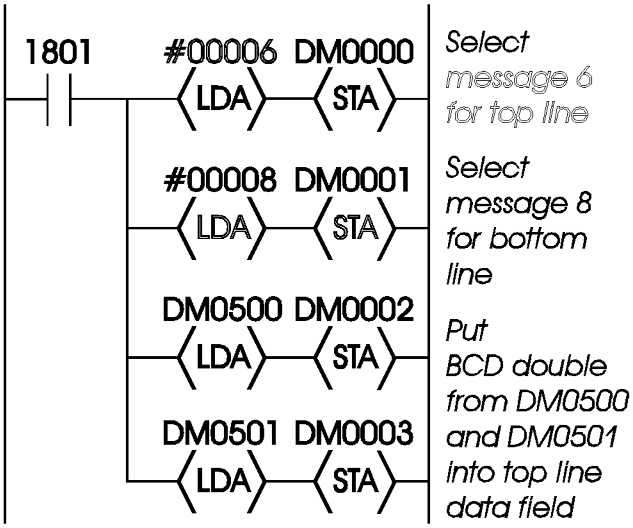
Select message 8 to bottom line

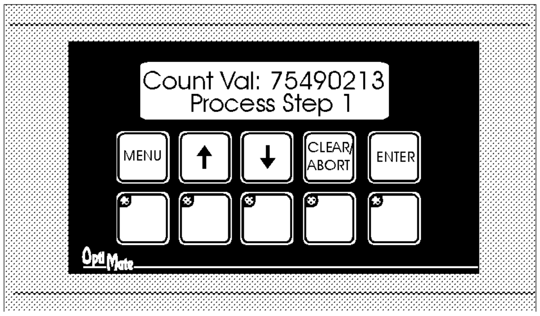
Put data from DM0406 into top line data field

DM0000	Top line message selection
DM0001	Bottom line message selection
DM0002	Top line data
DM0003	Top line data 2 (for long BCD)
DM0004	Bottom line data
DM0005	Bottom line data 2 (for long BCD)
DM0006	Function Selection
DM0007	Status register
DM0008	Control register

Displaying BCD Double Numbers

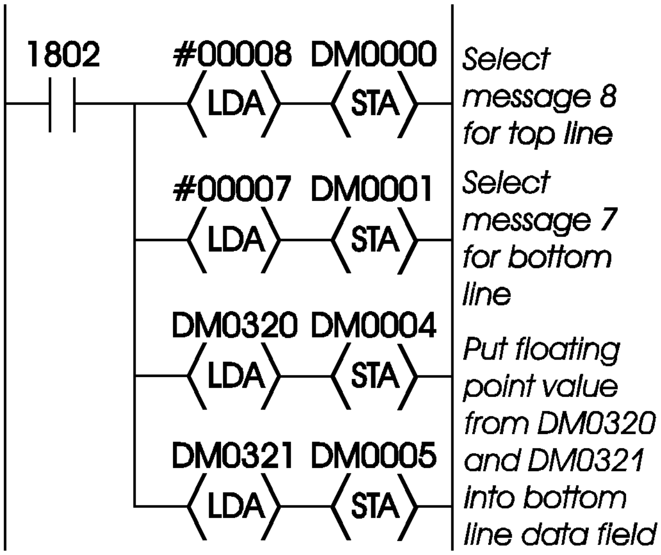
The example shown on the right is similar to the last example. The primary difference is that it uses a BCD double number in the top line display. The top line uses data display message 6, which has been configured as a BCD double display. The data for the data field is coming from DM0500 and DM0501. The bottom line is text message 8. DM0500 contains the least significant word while DM0501 contains the most significant word of the BCD double value.

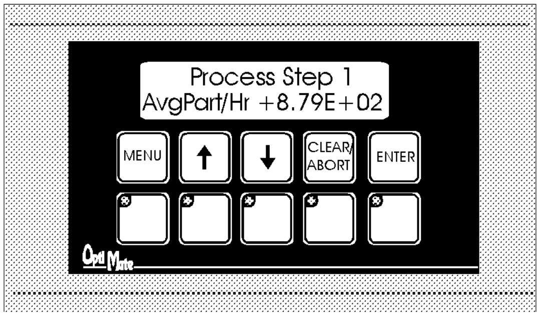

1801
#00006 DM0000
LDA STA
Select message 6 for top line
#00008 DM0001
LDA STA
Select message 8 for bottom line
DM0500 DM0002
LDA STA
Put BCD double from DM0500 and DM0501 into top line data field
DM0501 DM0003
LDA STA



Displaying Floating-Point Numbers

The example shown on the right is similar to the previous example. The primary difference is that it displays a floating-point number. The bottom 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 16-bit registers. 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 DM0320 and DM0321 to the bottom line display registers DM0004 and DM0005. The top line is text message 8.


1802
#00008 DM0000
LDA STA
Select message 8 for top line
#00007 DM0001
LDA STA
Select message 7 for bottom line
DM0320 DM0004
LDA STA
Put floating point value from DM0320 and DM0321 into bottom line data field
DM0321 DM0005
LDA STA

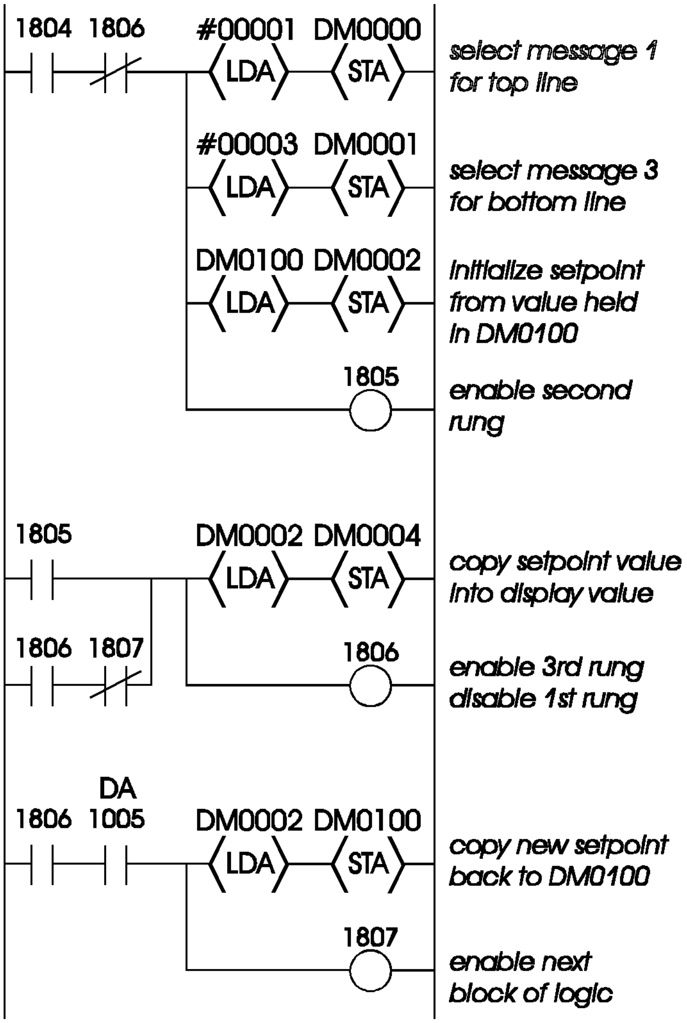


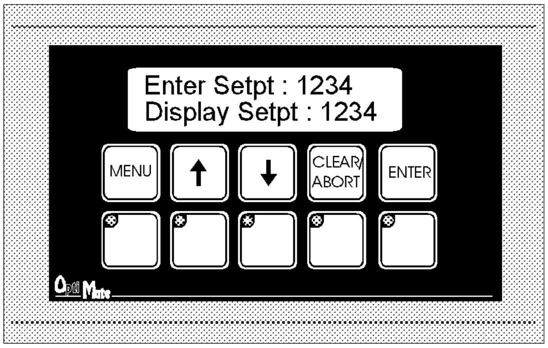
Arrow Adjustment of Setpoint Data

The figure to the right illustrates arrow adjustment of setpoint data. The process is enabled when 1804 is energized. The first program rung places the “Enter Setpoint” and “Display Setpt” messages on the top and bottom lines of the LCD. It also initializes the setpoint value by copying the value from DM0100 to the top line data register, DM0002, and it also enables the second rung.

As long as it is enabled, the second rung continually copies the setpoint value on the top line to the display message on the bottom line. It also disables the first rung and latches itself on until unlatched by the next rung.

The third rung waits until the data-available flag is set (1005), then copies the setpoint value back to DM0100. It also unlatches the second rung and, by activating 1807, enables the next block of logic (whatever that might be) in the program.


1804
1806
#00001
DM0000
LDA
STA
select message 1 for top line
#00003
DM0001
LDA
STA
select message 3 for bottom line
DM0100
DM0002
LDA
STA
initialize setpoint from value held in DM0100
1805
enable second rung
1805
DM0002
DM0004
LDA
STA
copy setpoint value into display value
1806
1807
1806
enable 3rd rung disable 1st rung
DA
1806
1005
DM0002
DM0100
LDA
STA
copy new setpoint back to DM0100
1807
enable next block of logic

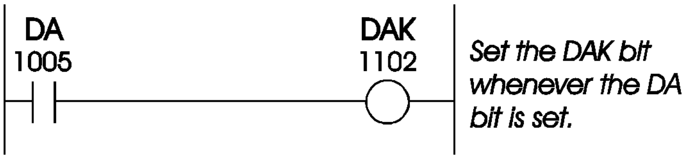


Acknowledging Data Entry

When the ENTER key is pressed during an arrow adjust message, the panel will set the data available (DA) bit in the status register. The arrow adjust message data will lock up (i.e. can’t be changed by the user) until the DA bit is cleared.

The DA bit can be cleared by two means. The first is to change the arrow adjust message to a message with a different number. The other method used to clear the DA bit is to set the data acknowledge (DAK) bit.

If you want to use the DAK bit to clear the DA bit, the program example on the right shows how to implement that procedure. **This rung should be placed at the end of the program to ensure that it will be checked every scan.** The rung will remain active as long as the DA bit is active. When the module sees that the DAK bit is set, it will write to the status register and clear the DA bit. This will normally happen in less than a second. When the DA bit clears, the panel will unlock the arrow adjust message and the user can change it if necessary.



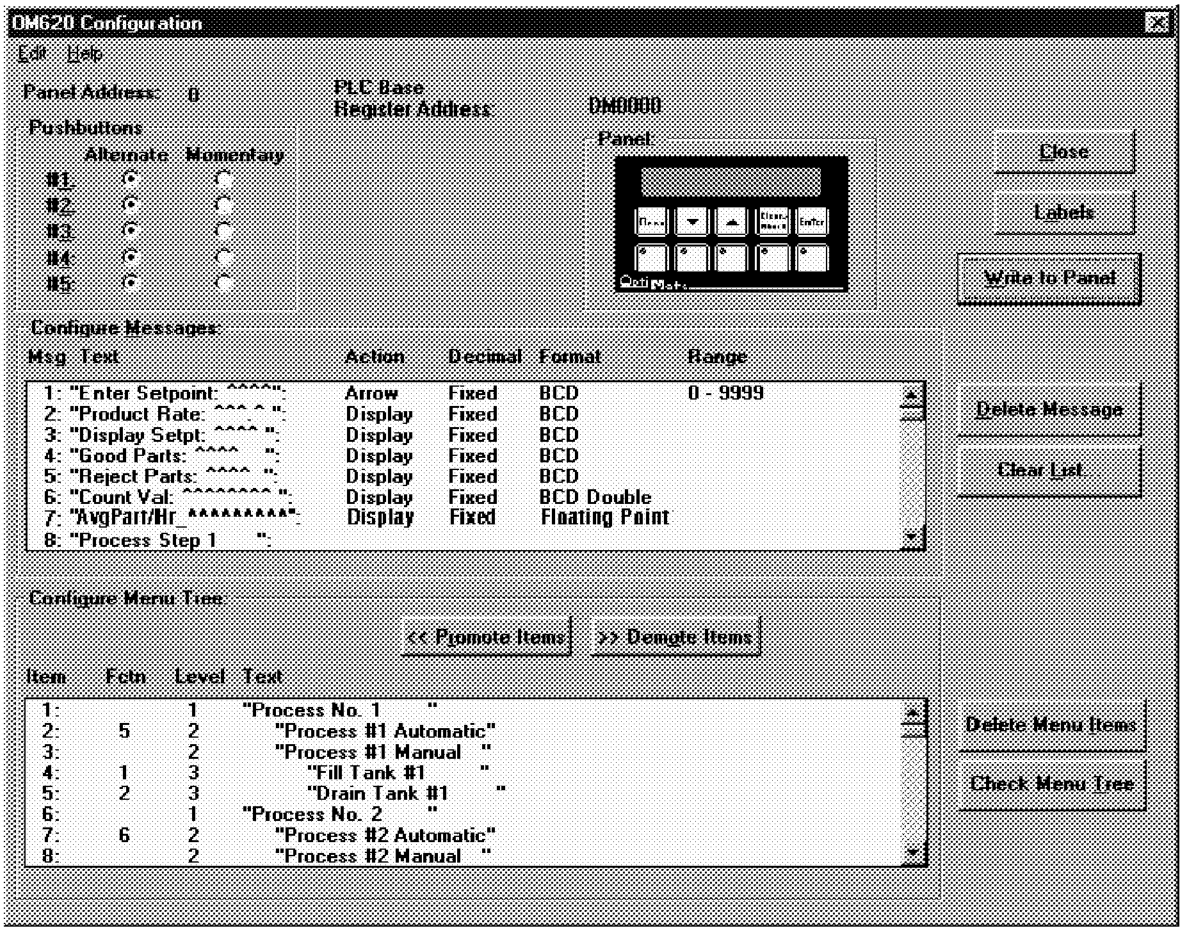
Using a Menu Tree

The KM620 allows you to pre-define and use a layered menu tree for function selection. The operation of this menu tree is taken care of entirely by the KM620 panel. The only requirement that the user has is to enable or disable the menu operation, and branch to the appropriate function logic when a function selection is made.

We will use the menu-tree definition shown below for our example.

The OM-WINEDIT Configuration Editor screen shown below displays the menu structure, including function associations, on the lower section of the display. For example, if the operator selects "Process #1 Automatic" from the menu tree, function number 5 would be placed in the function select register (M+6). If "Process No.2" is selected, no function number is selected; the terminal display will go to the next lower layer of the menu - "Process #2 Automatic".

Any menu item that has lower level menu items below it will, when selected, branch to the next level. If the "MENU" button is pressed, the terminal will back the menu tree up to the next higher level (towards the trunk). Arrow keys will, when pressed, step the panel through selection items on the same level of the same branch. The lowest level items on any branch will be function selections.



OM620 Configuration

Panel Address: 0

PLC Base Register Address: DM0000

Pushbuttons

	Alternate	Momentary
#1	●	○
#2	●	○
#3	●	○
#4	●	○
#5	●	○

Configure Messages:

Msg	Text	Action	Decimal	Format	Range
1:	"Enter Setpoint: ^^^^":	Arrow	Fixed	BCD	0 - 9999
2:	"Product Rate: ^^^.^ ":	Display	Fixed	BCD	
3:	"Display Setpt: ^^^^ ":	Display	Fixed	BCD	
4:	"Good Parts: ^^^^ ":	Display	Fixed	BCD	
5:	"Reject Parts: ^^^^ ":	Display	Fixed	BCD	
6:	"Count Val: ^^^^^^^^ ":	Display	Fixed	BCD Double	
7:	"AvgPart/Hr_*********":	Display	Fixed	Floating Point	
8:	"Process Step 1 ":				

Configure Menu Tree:

Item	Fctn	Level	Text
1:		1	"Process No. 1 "
2:	5	2	"Process #1 Automatic"
3:		2	"Process #1 Manual "
4:	1	3	"Fill Tank #1 "
5:	2	3	"Drain Tank #1 "
6:		1	"Process No. 2 "
7:	6	2	"Process #2 Automatic"
8:		2	"Process #2 Manual "

A Menu Tree Example

The program shown below illustrates menu tree function selection using the menu tree shown on the previous page. The first rung enables the menu tree by setting the ME bit when 1810 is active.

> **Note : Enabling the menu tree does not automatically put the terminal into the menu. Once enabled, the terminal will bring up the menu tree when the operator presses "MENU". Until then, the display is under PLC program control.**

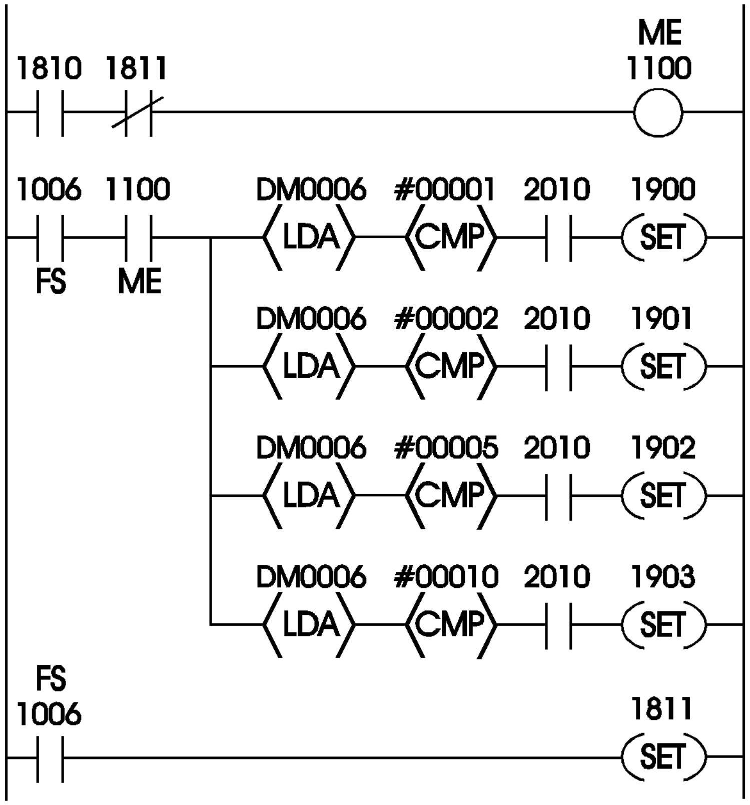
Decoding a Function Number

The second rung shown below sets the appropriate function enable bit when a selection is selected from the KM620. Function selection will activate the FS flag (which, based on our register copy shown earlier, will be 1006). The value held in the function register (M+6) is to be compared to function numbers defined in the menu tree. When a match is found, the appropriate enable relay (1900, 1901, etc.) that you choose for that function should be set.

The last rung shown will disable the Menu Enable (ME) bit by setting 1811. When this occurs, the panel will exit the menu tree, then LCD control will return to your program.

Note that the logic shown interlocks the function select enable bits (setting 1900, 1901, etc.) with the function select and menu enable flags (1006, 1100) to ensure that only one function is enabled each time a function is selected.

See the next two pages for implementation of functions.



Enable menu by setting the ME bit

Enable "Fill Tank #1" function

Enable "Drain Tank #1" function

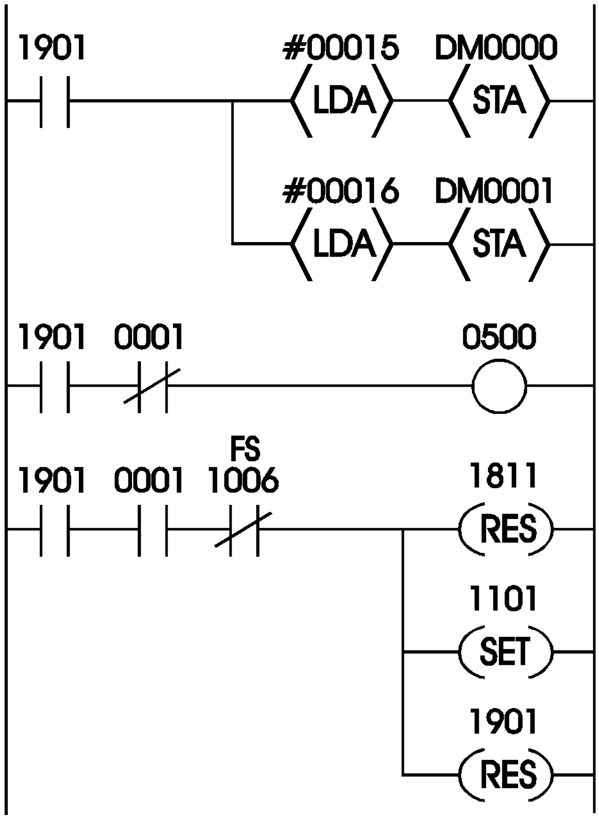
Enable "Process #1 Automatic" function

Enable "Tank 2 Maximum Level " function

Disable menu by clearing the ME bit

Implementing a Menu Function that utilizes the Message Display Capability

The program logic shown below illustrates how a typical function might be implemented. Suppose the selection was made by the operator to “Drain Tank #1”. According to the logic shown on the previous page, the selection of “Drain Tank #1” will place the function number 2 in register DM0006 which in turn would result in internal relay 1901 being set. The first rung of the “Drain Tank #1” function, shown below, will put messages 15 (”Tank Draining”) and 16 (”Please Wait”) into the top and bottom lines of the display. The second rung energizes output 0500 to open the drain until level sensor input 0001 senses that it is empty. When the tank is empty, the third rung re-enables the menu and sets it to return back to the “Drain Tank #1” selection by setting the menu return (MR) bit. The third rung also disables the function. By putting the function select relay (C1006) in series, we force the program to wait until the function select has been cleared before re-enabling the menu.



Load message 15 "Tank Draining" to the top line message register

Load message 16 "Please Wait" to the bottom line message register

Drains tank until sensor senses that it is empty

Energizes ME bit

Sets Menu Return (MR) bit

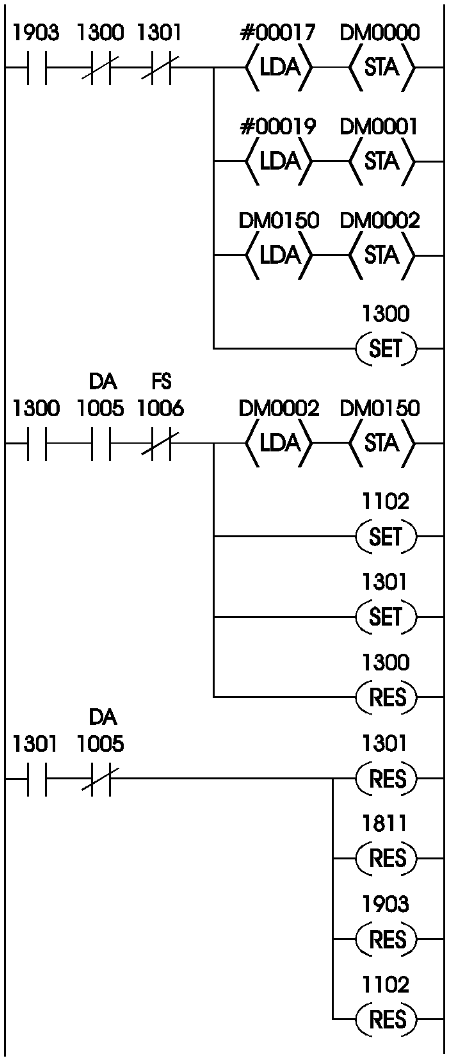
Since the process is complete, reset the function enable bit.

Implementing a Menu Function that uses an Arrow Adjustment Message

The logic shown below is a typical example of how a setpoint function might be implemented. Suppose the menu tree selection for "Tank 2 Maximum Level" was made. From the logic shown in ***Decoding a Function Number*** , the selection of "Tank 2 Maximum Level" will place function number 10 in register DM0006, which in turn will result in relay 1903 being set. Accordingly, the first rung shown below selects setpoint message 17 (T2 Max Level: ^^^^") for the top line and message 19 ("Adjust then ENTER") for the bottom line. The first rung also initializes the setpoint value for arrow adjustment, disables itself, and enables the second rung.

When the setpoint data is entered (after the FS flag is cleared), the second rung will copy the setpoint value back to its holding register (DM0150), acknowledge the DA bit, disable itself and enable the third rung.

The third rung waits until DA bit is clear, then clears the DAK bit and re-enables the menu. If the menu return (MR) flag is also set, the KM620 will return to the same point in the menu.



Load message 17
"T2 Max Level : ^ ^ ^ ^" to the
top line message register.

Load message 19 "Adjust then ENTER"
to the bottom line message register.

Initial value for arrow adjust message.
Copied from T2 Max Level holding
register to top line data register.

Disable current rung and enable
next rung.

When data entry complete (DA bit set),
copy new value from top line data register
to holding register for T2 Max Level.

Set DAK (Data Acknowledge) bit so
panel will clear DA bit.

Enable next rung which will wait for the
panel to clear the DA bit.

Disable current rung

Since the DA bit has been cleared,
disable current rung.

Energizes the ME bit to
Re-Enable the Menu

Since the function is complete,
clear the bit that enabled the function.

Clear the DAK bit.

Examples of Use with a Modicon PLC

Register Usage

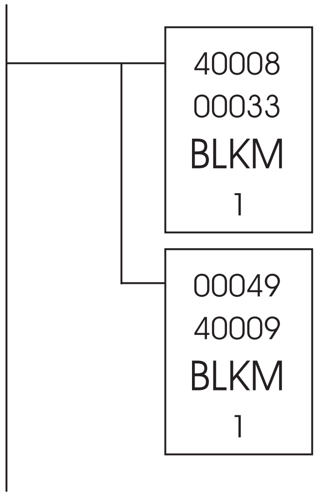
The OptiMate OM-WINEDIT software allows you to configure a module to use a block of 9 contiguous registers at a starting value that you define. For a Modicon PLC the recommended memory to use is the **4x** data memory area. Any block of registers between 40001 and 40617 can be used as long as the block does not conflict with anything that may be using those registers.

The first seven PLC registers in the block used by the KM620 panel are used for numeric information. As such they are ideally suited for the general purpose data memory registers 40001 - 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 data 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 KM620 is configured for a base address of 40001. **The program rung should be placed in the program to copy the status register 40008 (M+7) to the 0x relays beginning with 00033 and copy from the 0x relays beginning with 00049 to the control register 40009 (M+8).** 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 depends 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.



Register M+7 is mapped to relays 00033 - 00048.

Relays 00049 - 00064 are mapped to register M+8.

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

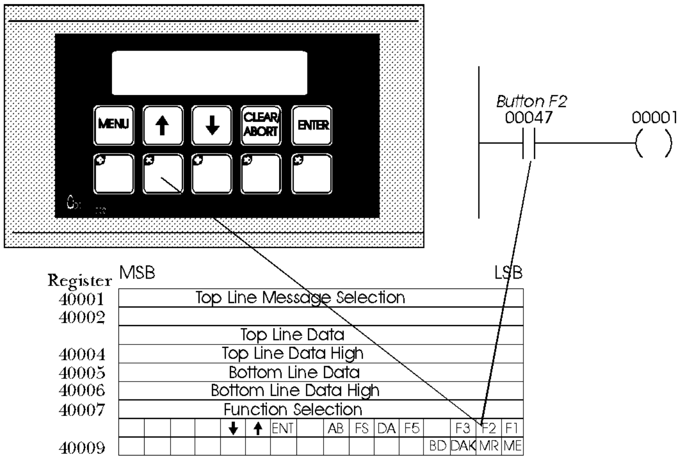
Register	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	bit
	MSB															LSB	
M+7	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	Status Resiter
M+8	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	Control Register

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

Status Register (M+7)			Control Register (M+8)	
Bit	Relay		Bit	Relay
F1	00048		ME	00064
F2	00047		MR	00063
F3	00046		DAK	00062
F4	00045		BD	00061
F5	00044			
DA	00043			
FS	00042			
AB	00041			
MA	00040			
ENTER	00039			
Up arrow	00038			
Down arrow	00037			

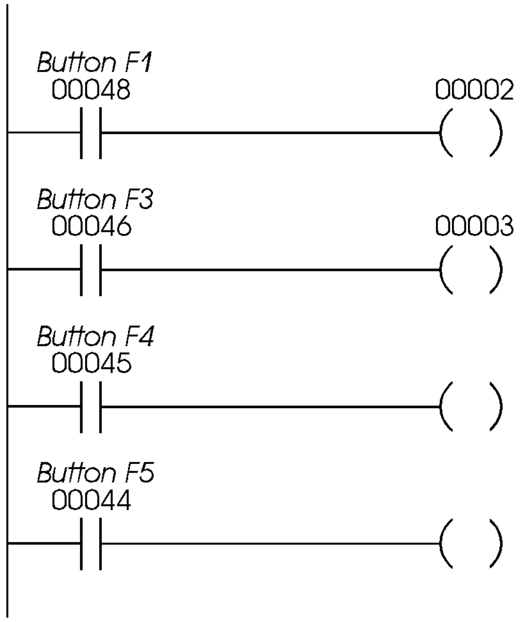
Using a Function Button

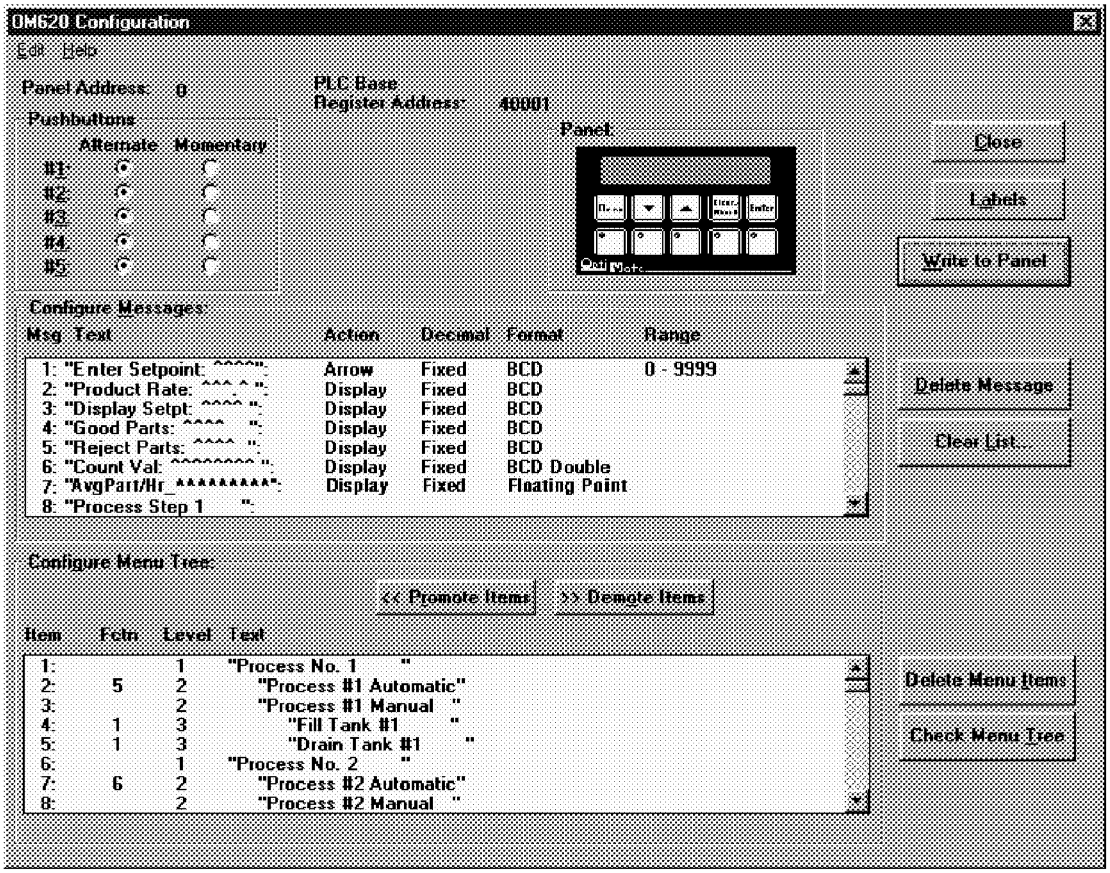
The five function buttons below the LCD display will appear as relay coils in your program (assuming the register copy rung described on the previous page is in your program). The following example turns ON output 00001 when button F2 is active.



Register	MSB															LSB
40001	Top Line Message Selection															
40002																
	Top Line Data															
40004	Top Line Data High															
40005	Bottom Line Data															
40006	Bottom Line Data High															
40007	Function Selection															
					↓	↑	ENT		AB	FS	DA	F5		F3	F2	F1
40009													BD	DAK	MR	ME

The following example shows the other four pushbuttons being used in a program. When active, button F1 turns ON output 00002, button F3 turns ON 00003, button F4 turns ON 00004 and button F5 turns ON output 00005.





OM620 Configuration

Edit Help

Panel Address: 0

PLC Base Register Address: 40001

Pushbuttons

	Alternate	Momentary
#1:	●	○
#2:	●	○
#3:	●	○
#4:	●	○
#5:	●	○

Panel

Configure Messages:

Msg	Text	Action	Decimal	Format	Range
1:	"Enter Setpoint: ^^^^":	Arrow	Fixed	BCD	0 - 9999
2:	"Product Rate: ^^^.^ ":	Display	Fixed	BCD	
3:	"Display Setpt: ^^^^ ":	Display	Fixed	BCD	
4:	"Good Parts: ^^^^ ":	Display	Fixed	BCD	
5:	"Reject Parts: ^^^^ ":	Display	Fixed	BCD	
6:	"Count Val: ^^^^^^^^ ":	Display	Fixed	BCD Double	
7:	"AvgPart/Hr_*********":	Display	Fixed	Floating Point	
8:	"Process Step 1 ":				

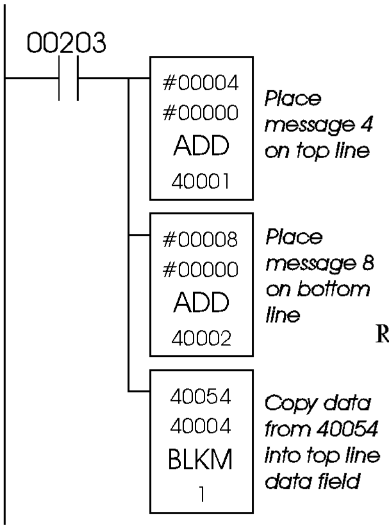
Configure Menu Tree:

Item	Fctn	Level	Text
1:		1	"Process No. 1 "
2:	5	2	"Process #1 Automatic"
3:		2	"Process #1 Manual "
4:	1	3	"Fill Tank #1 "
5:	1	3	"Drain Tank #1 "
6:		1	"Process No. 2 "
7:	6	2	"Process #2 Automatic"
8:		2	"Process #2 Manual "

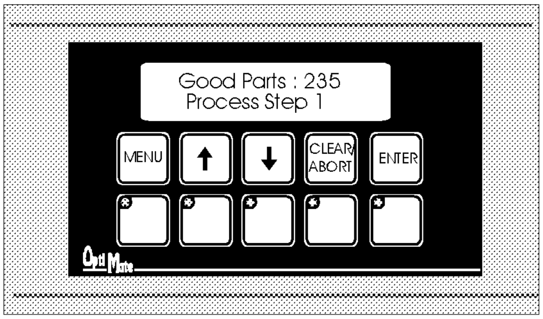
Displaying Messages on the LCD Display

Messages of various types can be configured via OM-WINEDIT and downloaded to the KM620. The message definitions shown in the figure above will be used in all of the examples that follow.

The example on the right shows a couple of messages being displayed to the LCD display. The top line uses data display message 4. The data for the data field is coming from 40054. The bottom line is text message 8.



00203
#00004
#00000
ADD
40001
Place message 4 on top line
#00008
#00000
ADD
40002
Place message 8 on bottom line
40054
40004
BLKM
1
Copy data from 40054 into top line data field

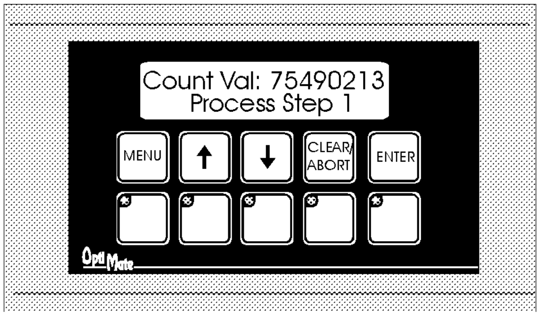


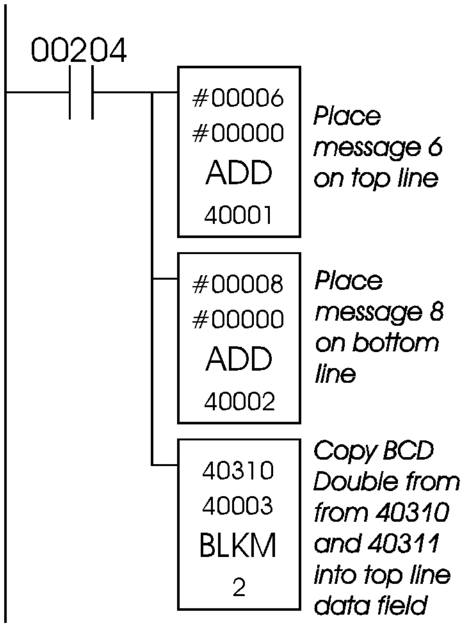
Register	MSB															LSB
40001	Top Line Message Selection															
40002	Bottom Line Message Selection															
40003	Top Line Data															
40004	Top Line Data High															
40005	Bottom Line Data															
40006	Bottom Line Data High															
40007	Function Selection															
40008					↓	↑	ENT	MA	AB	FS	DA	F5	F4	F3	F2	F1
40009													BD	DAK	MR	ME

Displaying BCD Double Numbers

Long (up to 8 digit) BCD numbers can be displayed by the KM620. This number format is a standard capability for some PLCs. However, the Modicon PLCs do not have an easy means of handling long BCD numbers. Due to the limitations of the Modicon, this capability will not normally be implemented, but, if a valid BCD double number is placed in the data registers for a BCD double message, the panel will display it.

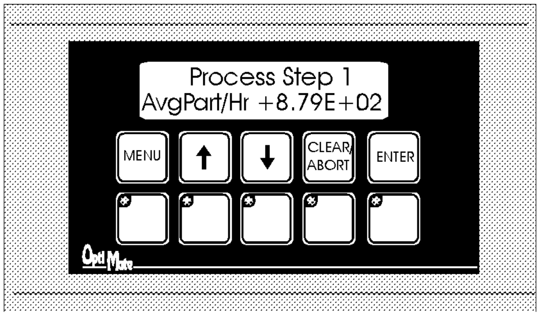
The example on the right illustrates the method for displaying BCD Double numbers. The top line displays message 6, a BCD Double message. Registers 40310 and 40311 contain an 8 digit BCD number, with the most significant word in 40311. The BCD Double value from registers 40310 and 40311 is copied to the top line data registers 40003 and 40004 using the BLKM command. The bottom line contains message 8, a text message.

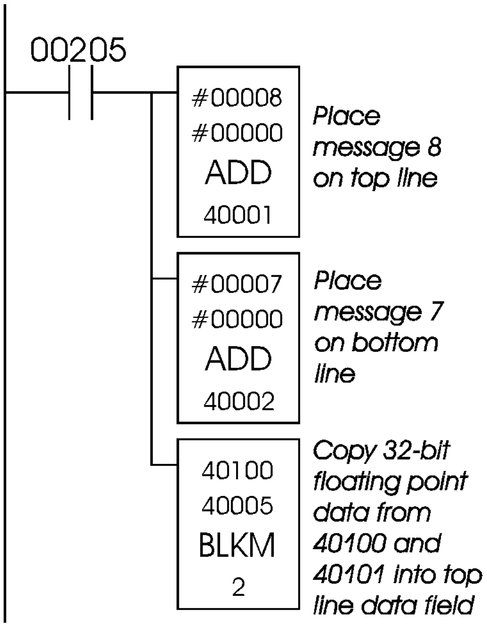



00204
#00006
#00000
ADD
40001
Place message 6 on top line
#00008
#00000
ADD
40002
Place message 8 on bottom line
40310
40003
BLKM
2
Copy BCD Double from from 40310 and 40311 into top line data field

Displaying Floating-Point Numbers

The example shown on the right is similar to the previous example. The primary difference is that it displays a floating-point number. The bottom 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 16-bit registers. 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 40100 and 40101 to the bottom line display registers 40005 and 40006. The top line is text message 8.



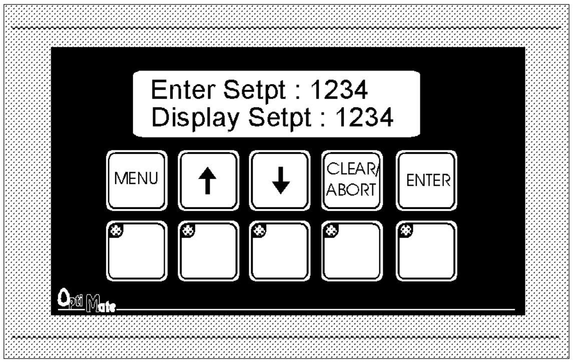

00205
#00008
#00000
ADD
40001
Place message 8 on top line
#00007
#00000
ADD
40002
Place message 7 on bottom line
40100
40005
BLKM
2
Copy 32-bit floating point data from 40100 and 40101 into top line data field

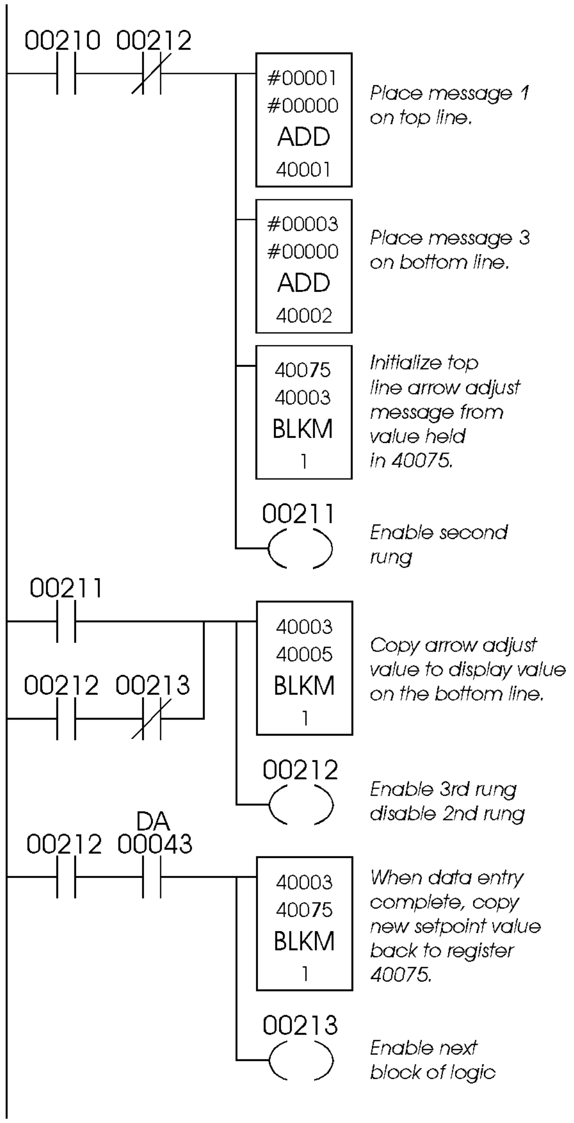
Arrow Adjustment of Setpoint Data

The figure to the right illustrates arrow adjustment of setpoint data. The process is enabled when 00210 is energized. The first program rung places the "Enter Setpoint" and "Display Setpt" messages on the top and bottom lines, initializes the setpoint value to the value in 40075 and also enables the second rung.

As long as it is enabled, the second rung continually copies the setpoint value on the top line to the display message on the bottom line. It also disables the first rung and latches itself on until unlatched by the next rung.

The third rung waits until the data available flag is set (00043), then copies the setpoint back to 40075. It also unlatches the second rung and, by activating 00213, enables the next block of logic (whatever that might be) in the program.





Place message 1 on top line.

Place message 3 on bottom line.

Initialize top line arrow adjust message from value held in 40075.

Enable second rung

Copy arrow adjust value to display value on the bottom line.

Enable 3rd rung disable 2nd rung

When data entry complete, copy new setpoint value back to register 40075.

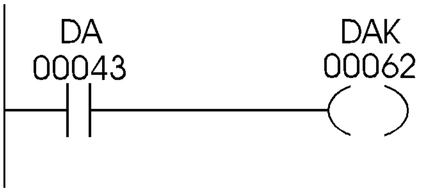
Enable next block of logic

Acknowledging Data Entry

When the ENTER key is pressed during an arrow adjust message, the panel will set the data available (DA) bit in the status register. The arrow adjust message data will lock up (i.e. can't be changed by the user) until the DA bit is cleared.

The DA bit can be cleared by two means. The first is to change the arrow adjust message to a message with a different number. The other method used to clear the DA bit is to set the data acknowledge (DAK) bit.

If you want to use the DAK bit to clear the DA bit, the program example on the right shows how to implement that procedure. **This rung should be placed at the end of the program to ensure that it will be checked every scan.** The rung will remain active as long as the DA bit is active. When the module sees that the DAK bit is set, it will write to the status register and clear the DA bit. This will normally happen in less than a second. When the DA bit clears, the panel will unlock the arrow adjust message and the user can change it if necessary.



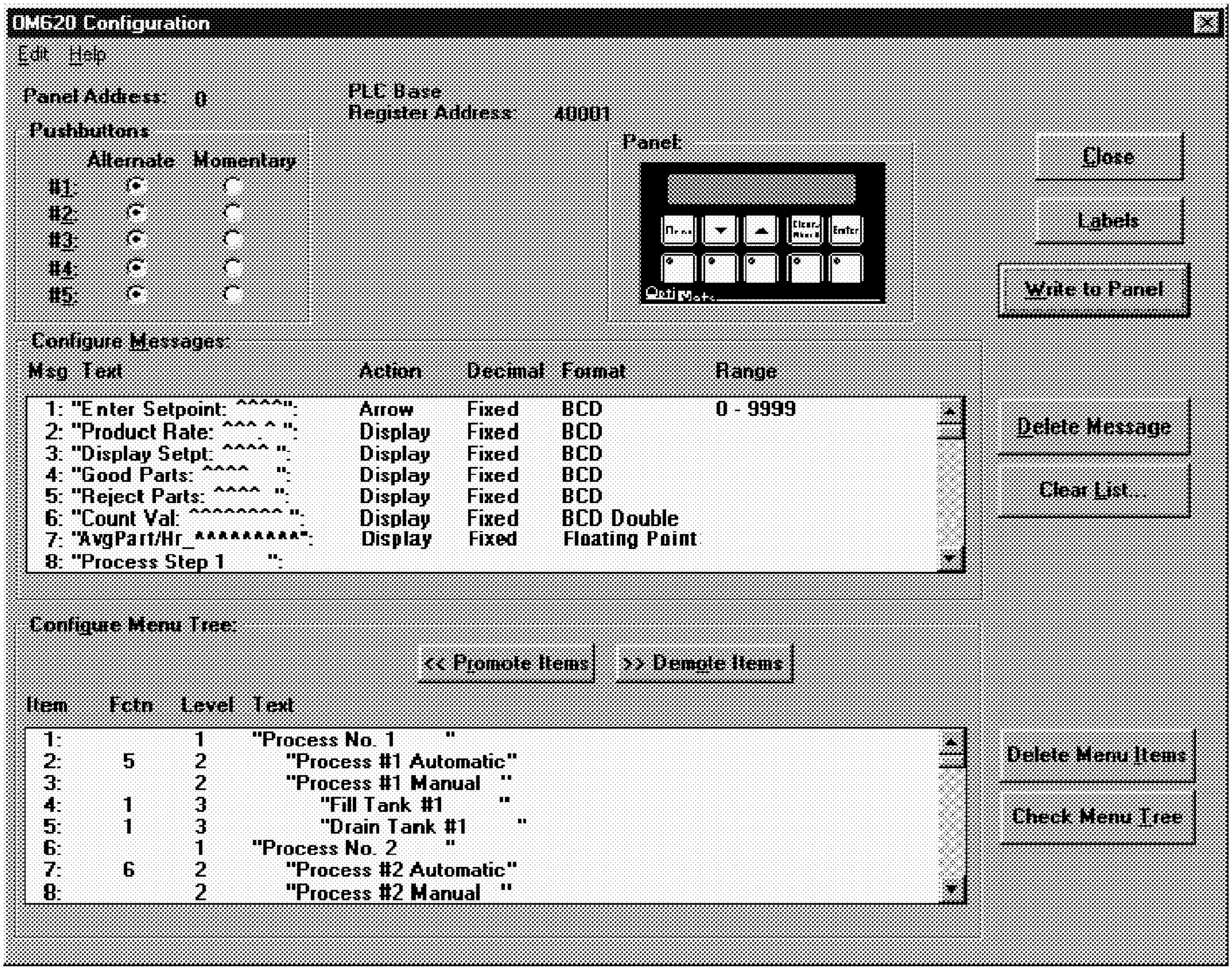
Using a Menu Tree

The KM620 allows you to pre-define and use a layered menu tree for function selection. The operation of this menu tree is taken care of entirely by the KM620 panel. The only requirement that the user has is to enable or disable the menu operation, and branch to the appropriate function logic when a function selection is made.

We will use the menu tree definition shown below for our example.

The OM-WINEDIT Configuration Editor screen shown below displays the menu structure, including function associations, on the lower section of the display. For example, if the operator selects "Process #1 Automatic" from the menu tree, function number 5 would be placed in the function select register (M+6). If "Process No.2" is selected, no function number is selected; the terminal display will go to the next lower layer of the menu - "Process #2 Automatic".

Any menu item that has lower level menu items below it will, when selected, branch to the next level. If the "MENU" button is pressed, the terminal will back the menu tree up to the next higher level (towards the trunk). Arrow keys will, when pressed, step the panel through selection items on the same level of the same branch. The lowest level items on any branch will be function selections.



Panel Address: 0

PLC Base Register Address: 40001

Pushbuttons

	Alternate	Momentary
#1:	●	○
#2:	●	○
#3:	●	○
#4:	●	○
#5:	●	○

Configure Messages:

Msg	Text	Action	Decimal	Format	Range
1:	"Enter Setpoint: ^^^^":	Arrow	Fixed	BCD	0 - 9999
2:	"Product Rate: ^^^.^ ":	Display	Fixed	BCD	
3:	"Display Setpt: ^^^^ ":	Display	Fixed	BCD	
4:	"Good Parts: ^^^^ ":	Display	Fixed	BCD	
5:	"Reject Parts: ^^^^ ":	Display	Fixed	BCD	
6:	"Count Val: ^^^^^^^^ ":	Display	Fixed	BCD Double	
7:	"AvgPart/Hr_^^^^^^^^^":	Display	Fixed	Floating Point	
8:	"Process Step 1 ":				

Configure Menu Tree:

Item	Fctn	Level	Text
1:		1	"Process No. 1 "
2:	5	2	"Process #1 Automatic"
3:		2	"Process #1 Manual "
4:	1	3	"Fill Tank #1 "
5:	1	3	"Drain Tank #1 "
6:		1	"Process No. 2 "
7:	6	2	"Process #2 Automatic"
8:		2	"Process #2 Manual "

A Menu Tree Example

The program shown below illustrates menu tree function selection using the menu tree shown on the previous page. The first rung enables the menu tree by setting the ME bit when 00349 is active.

Note : Enabling the menu tree does not automatically put the terminal into the menu. Once enabled, the terminal will bring up the menu tree when the operator presses "MENU". Until then, the display is under PLC program control.

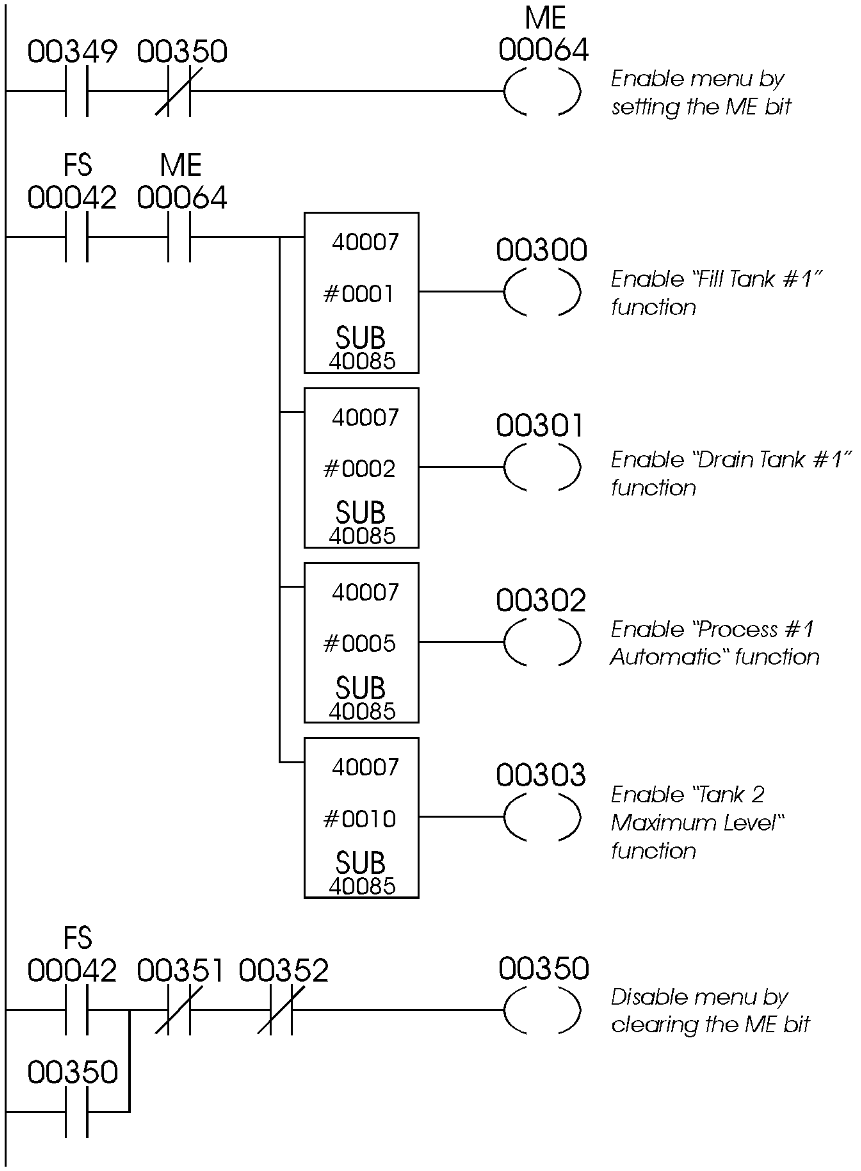
Decoding a Function Number

The second rung shown below sets the appropriate function enable bit when a function is selected in the KM620. Function selection will activate the FS flag (which, based on our register copy shown earlier, will be 00042). The value held in the function register (M+6) is to be compared to function numbers defined in the menu tree. When a match is found, the appropriate enable relay (00300, 00301, etc.) that you choose for that function should be set.

The last rung shown will disable the Menu Enable (ME) bit by setting 00350. When this occurs, the panel will exit the menu tree, then LCD control will return to your program.

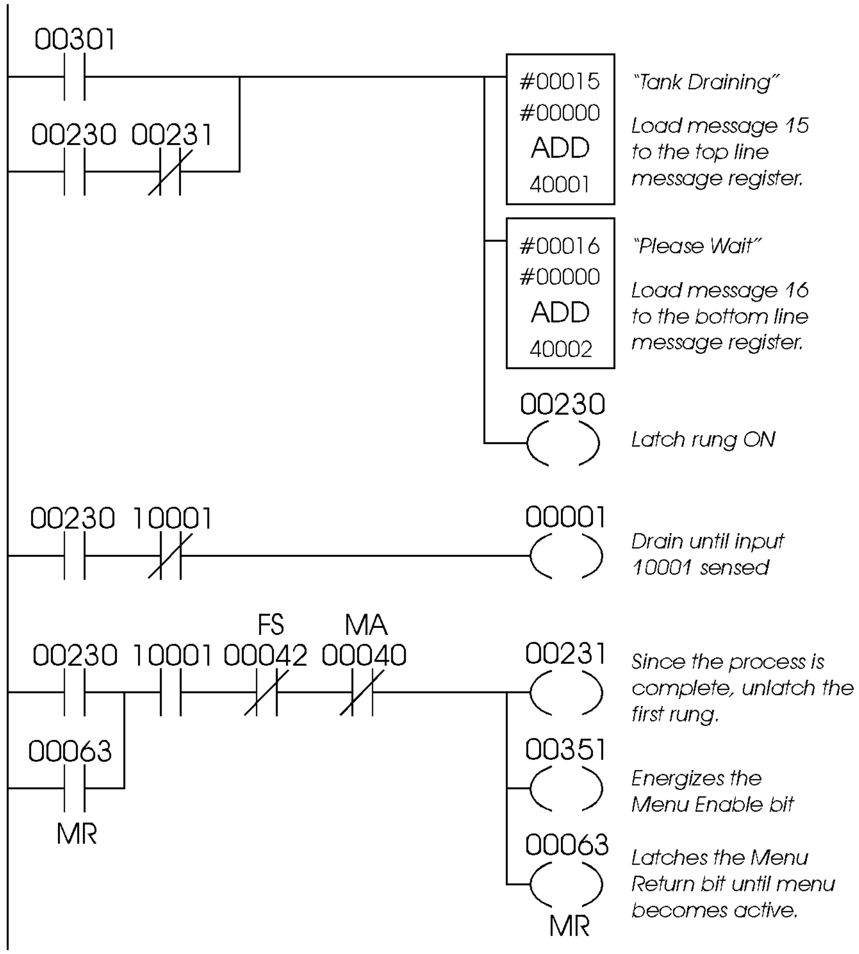
Note that the logic shown interlocks the function select enable bits (setting 00300, 00301, etc.) with the function select and menu enable flags (00042, 00064) to ensure that only one function is enabled each time a function is selected.

See the next two pages for implementation of functions.


00349
00350
ME
00064
Enable menu by setting the ME bit
FS
00042
ME
00064
40007
#0001
SUB
40085
00300
Enable "Fill Tank #1" function
40007
#0002
SUB
40085
00301
Enable "Drain Tank #1" function
40007
#0005
SUB
40085
00302
Enable "Process #1 Automatic" function
40007
#0010
SUB
40085
00303
Enable "Tank 2 Maximum Level" function
FS
00042
00351
00352
00350
Disable menu by clearing the ME bit
00350

Implementing a Menu Function that utilizes the Message Display Capability

The program logic shown below illustrates how a typical function might be implemented. Suppose the selection was made by the operator to "Drain Tank #1". According to the logic shown on the previous page, the selection of "Drain Tank #1" will place the function number 2 in register 40007, which in turn will result in internal relay 00301 being set. The first rung of the "Drain Tank #1" function, shown below, will put messages 15 ("Tank Draining") and 16 ("Please Wait") into the top and bottom lines of the display. The second rung energizes output 00001 to open the drain until level sensor input 10001 senses that it is empty. When the tank is empty, the third rung re-enables the menu and sets it to return back to the "Drain Tank #1" selection by setting the menu return (MR) bit. The third rung also disables the function. By putting the function select relay (00042) in series, we force the program to wait until the function select has been cleared before re-enabling the menu. By placing the MR bit (00063) in parallel with 00230, the MR bit is latched ON until the menu becomes active (MA bit set).



"Tank Draining"

Load message 15 to the top line message register.

"Please Wait"

Load message 16 to the bottom line message register.

Latch rung ON

Drain until input 10001 sensed

Since the process is complete, unlatch the first rung.

Energizes the Menu Enable bit

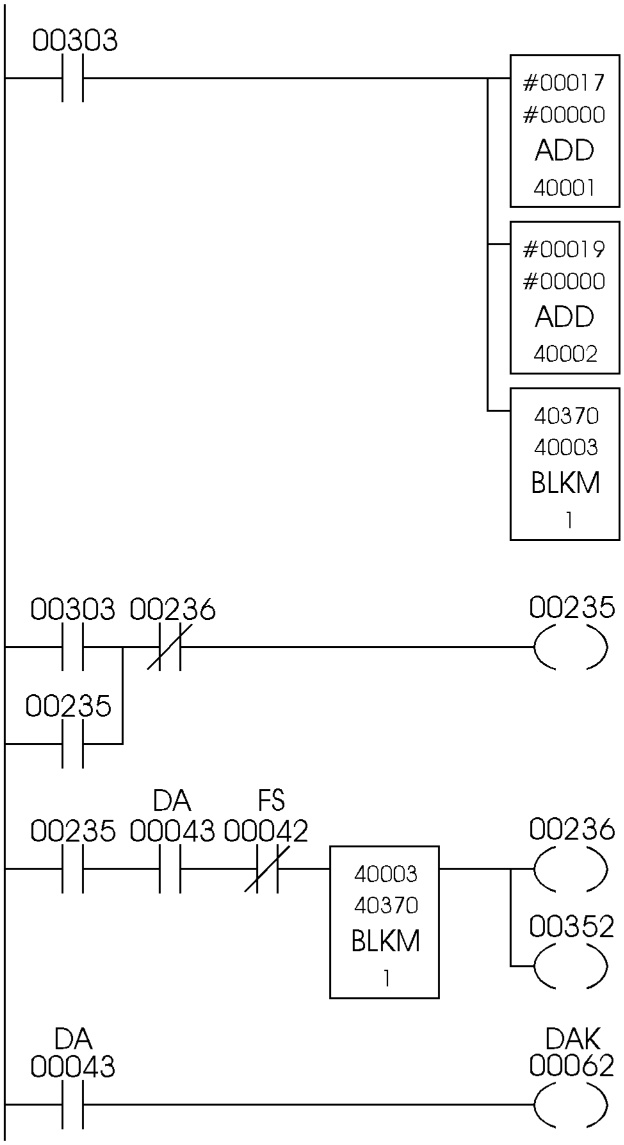
Latches the Menu Return bit until menu becomes active.

Implementing a Menu Function that uses an Arrow Adjustment Message

The logic shown below is a typical example of how a setpoint function might be implemented. Suppose the menu tree selection for “Tank 2 Maximum Level” was made. From the logic shown in ***Decoding a Function Number***, the selection of “Tank 2 Maximum Level” will place function number 10 in register 40007, which in turn will result in relay 00303 being set. Accordingly, the first rung shown below selects setpoint message 17 (T2 Max Level: ^^^^”) for the top line and message 19 (”Adjust then ENTER”) for the bottom line. The first rung also initializes the setpoint value for arrow adjustment. The second rung latches itself ON and also enables the third rung.

When the setpoint data is entered (DA bit set), the third rung will copy the setpoint value back to its holding register (40370) and disable the second rung, which in turn, will also disable the third rung. It will also re-enable the ME bit via 00352.

The last rung waits until DA bit is clear, then clears the DAK bit. If the menu return (MR) flag is also set, the KM620 will return back to the “Tank 2 Maximum Level” selection in the menu tree.



"T2 Max Level : ^ ^ ^ ^"

Load message 15 to the top line message register.

"Adjust then ENTER"

Load message 16 to the bottom line message register.

Initial value for arrow adjust message. Copied from T2 Max Level holding register to top line data register.

Latch ON current rung and enable next rung.

When data entry complete (DA bit set), copy new value from top line data register to holding register for T2 Max Level.

Since function is complete, unlatch previous and current rung..

Energize the ME bit to Re-Enable the Menu

Set Data Acknowledge (DAK) to clear DA bit

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 KM620 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 OM-WINEDIT 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 (note; if parity is set to even or odd, only one stop bit is allowed). Once selected, the configuration must be downloaded to the panel.

Computer-Based Operation

The KM620 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 KM620 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 KM620 can be instructed to display predefined messages. Selection of a predefined message simply requires transmitting a command which selects the particular message.

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. If it is a display data field, it may be continuously updated. If it is a data adjustment message (arrow key adjustment), only the first value transmission will be accepted.

Data Entry/Adjustment with the Arrow Keys

Data entry in the KM620 can be performed via the arrow adjustment keys. This type of data entry is most applicable to cases where you want to have a number adjusted slightly, rather than entered from scratch. This happens many times with setpoints. It is also common when adjusting speeds and rates. When this is done in an operation, it is commonly referred to as a "jog" operation.

A message can be set up for arrow adjustment through the OM-WINEDIT Editor. When the data screen is required in the program, transmit a command for the preconfigured message and its current value. The digits marked by carets will initially display the current value.

As the operator presses the up or down arrow key, the numeric value will increment or decrement respectively. As it is adjusted, the value is continuously available to be read by the host computer. When adjustment is complete, the operator will press the ENTER button. When this happens the data available (DA) status bit will be set. When the computer host sends a DAK or changes the message, the DA status bit will clear.

Adjustment of data will be limited to within the limits defined through the configuration editor. Data also will be limited to the number of digits defined by the carets in the message.

Function Buttons

The KM620 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 status request message will return the current state of each of the five buttons along with other status.

Note: All OptiMate Hex protocol messages are transmitted in hexadecimal number format. The message responses will be transmitted back to the host in the same number format.

KM620 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
                                        0xA5 ; Read operator entered data
                                        0xA7 ; Send data for data display message
                 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 a hexadecimal number.
```

General status/control

```
STX   Module  0xA0  control  checksum
      address
           where   control       bit 0 ; Menu enable
                                 bit 1 ; Menu return
                                 bit 2 ; buzzer disable
                                 bit 3 ; Data acknowledge

response
STX   term_stat   disp_stat   ftn_select   checksum            if message received and processed OK
           where term_stat      = terminal status
                        bits 0-4 = Button status for function buttons 1-5 respectively (1 = active, 0 = inactive)
                        bit 5    : operator entered data available
                        bit 6    : function selected
                        bit 7    : abort key pressed
                 disp_stat       = display status (1=TRUE)
                        bit 0    : Top line display data
                        bit 1    : Top line arrow adjust
                        bit 2    : Bottom line display data
                        bit 3    : Bottom line arrow adjust
                        bit 4    : Enter button
                        bit 5    : Up arrow
                        bit 6    : Down arrow
                        bit 7    : Menu active
or
NAK   if any errors
           Where NAK             =  0x15
```

Select Predefined Message

```
STX   Module  0xA1  line   mesg_no   data   checksum
      Address
           where line          = top (0x00) or bottom (0x01) 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.
                        - Used as display data for numeric data display message.
                        - Used as initial value for arrow adjustment type message (integer portion only).
                        - Ignored for all other message types.
response
ACK        if message received and processed OK
           Where ACK             =  0x06
 or
NAK        if any errors in message
```

Read Operator Entered Data

STX Module Address 0xA5 checksum

response

STX status data checksum *if message receive and processed OK*

where status = 0x00 if ENTER key has not been pressed
= 0x01 if ENTER key has been pressed

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.

Note : This message can be used to read an arrow adjust message value while it is being adjusted.

or

NAK *if any errors in message*

Send data for data display message

STX Module Address 0xA7 line data checksum

where line = top (0x00) or bottom (0x01) 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.

response

ACK *if message received and processed OK*

or

NAK *if any errors in message*

Display status request

STX Module Address 0xA9 checksum

response

STX top_msg bot_msg checksum *if message received and processed OK*

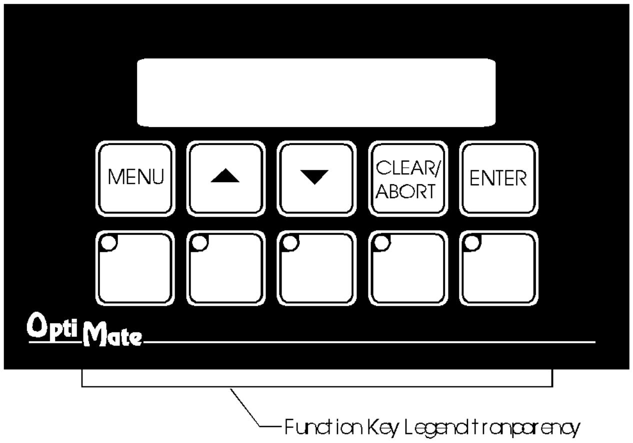
where top_msg = last predefined message selected for top line
bot_msg = last predefined message selected for bottom line

or

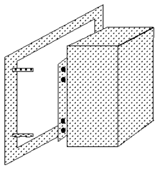
NAK *if any errors in message*

Set Up and Interconnect

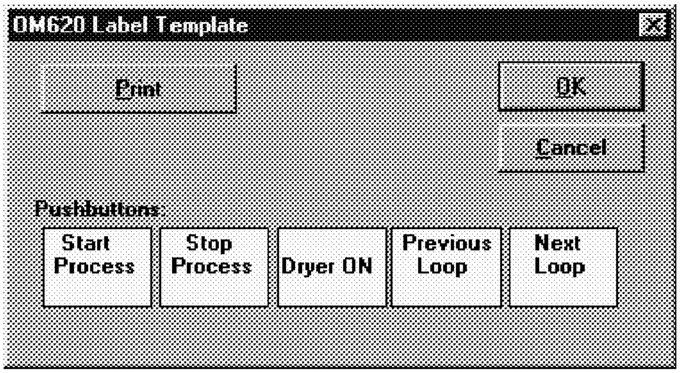
Legending the Function Keys



Legending the KM620 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.



Pushbuttons:

Start Process	Stop Process	Dryer ON	Previous Loop	Next Loop

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 legends slide in from the bottom.
- Re-attach bezel. Push bezel onto box until it snaps together.

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.

OptiMate Panel RS232

Host Computer/PLC	OptiMate Panel DB-15 Male
TX	3 RS232 RX
RX	2 RS232 TX
Sig Gnd	5 Sig Gnd

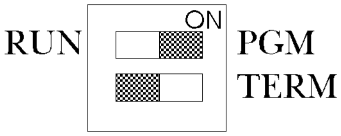
OptiMate Panel RS422

Host Computer/PLC	OptiMate Panel DB-15 Male
TX+	9 RS422 RX+
TX-	10 RS422 RX-
RX+	11 RS422 TX+
RX-	12 RS422 TX-

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 KM620 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 1.5 amps. This is typical of nearly any type of electronic equipment and is due to the initial charging of power capacitors. This surge is normally easily handled by commercial power supplies.

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, Windows 95, 98 or NT. It allows the user to select the exact functionality to meet application requirements.

For the KM620 module, the following are important configuration parameters.

Computer-Based Systems

Decision	Selection
Single/Multi Module	Choose single panel even if the system will contain several panels. The Multi panel selection applies only to systems using an OM9001. In computer based systems, each panel is configured independently. After configuration, multiple panels can be connected together to form a system.
Configuration starting point	First time configuration, start with defaults for panel. Subsequent configurations can utilize disk files you create.
PLC Type	Select OptiMate Hex
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.
Buttons	Select momentary or alternate action as required for your application
Messages	Define messages as required for application

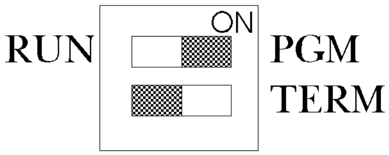
Single Module PLC-Based Systems

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 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

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 appropriate baud rate, # bits, # stop 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.

Configuration must be downloaded from an IBM PC compatible computer to each panel. This is done over the serial link. Panel must be selected for "PGM" (DIP switch in back of the module) for module 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 KM620 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 KM620 should always be set to the following:

KM620 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 KM620. 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 KM620 with the parameters shown in the table above.

Communication through the KV-L2

The KM620 uses the KV mode protocol to communicate with the PLC, therefore, the port that 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

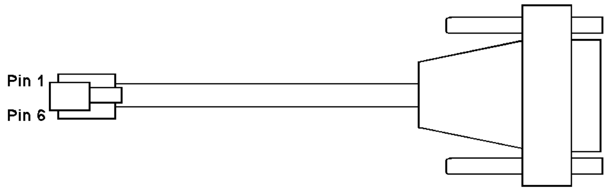
The dip switch settings shown above are dependent on the on the port that the KM620 is connected to. If the panel is connected to Port 1, then switches A1 and A2 should be OFF. If you have another device on Port 2 , then the switches for Port 2 should be set to the proper protocol for that device.

The KM620 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, away have the terminator dip switch OFF on the KV-L2 and on the KM620 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 KM620 to a Keyence PLC CPU. This cable is available from Optimation.



Keyence PLC CPU RJ-12 (6P6C)	OptiMate Panel DB15
RX 4	2 TX
TX 2	3 RX
SG 3	5 SG

The figure below shows the pinouts to connect a KM620 to a KV-L2 Serial Interface Module. The pinouts shown are for RS232.

KV-L2 to OptiMate 600 Series Panels

KV-L2 Port 1 DB25	OptiMate DB15
RX 3	2 TX
TX 2	3 RX
SG 7	5 SG
RS 4	
CS 5	

KV-L2 Port 2 DB25	OptiMate DB15
RD	2 TX
SD	3 RX
SG	5 SG

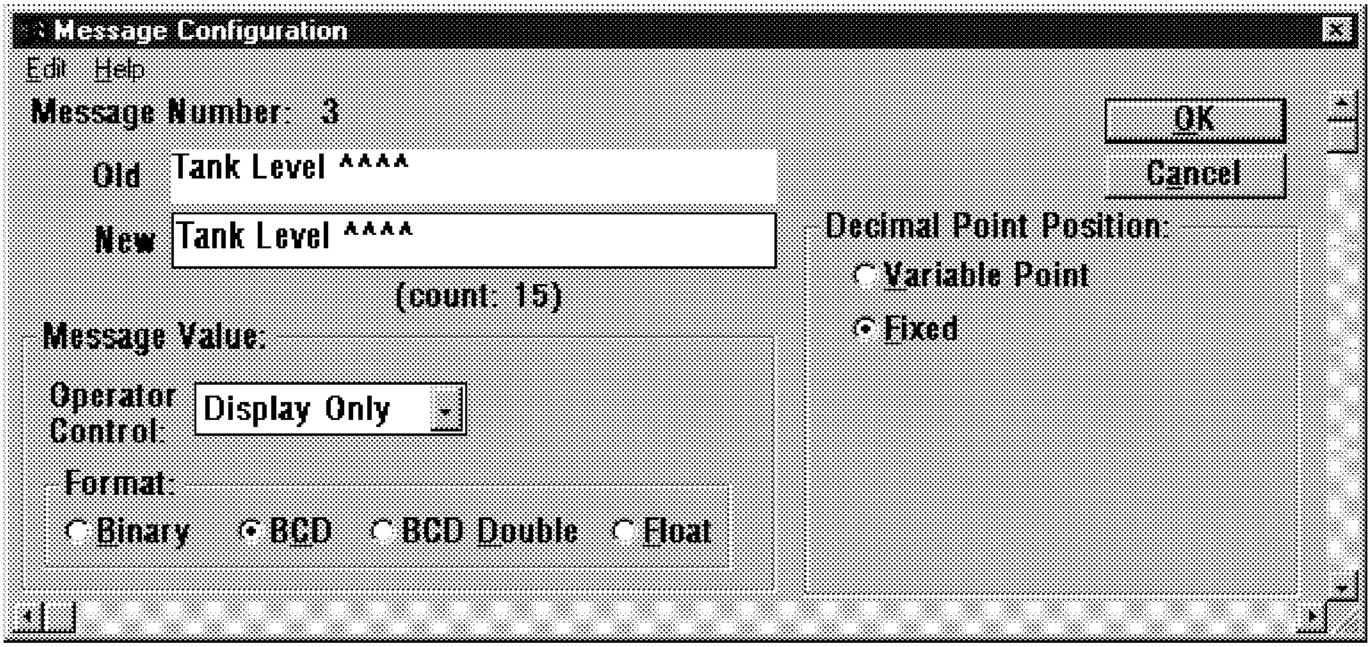
Creating Messages

The figure on right the 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 parameter, including data type, message type, etc., as shown below.

Message #	Text (20Characters Max.)																			
1	E	n	t	e	r		S	e	t	p	o	i	n	t	:		^	^	^	^
2	P	r	o	d	u	c	t		R	a	t	e	:		^	^	^	.	^	
3	S	h	i	f	t		T	a	r	g	e	t	:		^	^	^	.	^	

Example Message Definitions



Message Configuration

Message Number: 3

Old: Tank Level ^^^^

New: Tank Level ^^^^

(count: 15)

Message Value:

Operator Control: Display Only

Format: ○ Binary ● BCD ○ BCD Double ○ Float

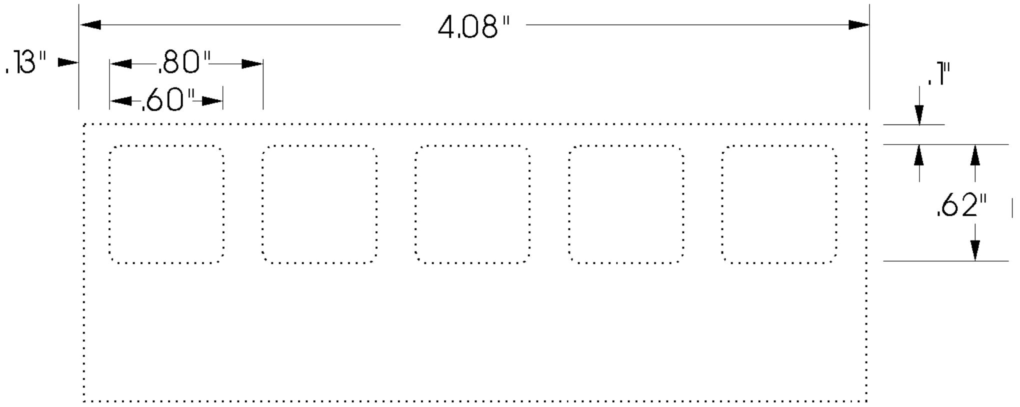
Decimal Point Position: ○ Variable Point ● Fixed

Label and Message Definition Templates

Message #	Text (20Characters Max.)																			

Message Definition Template

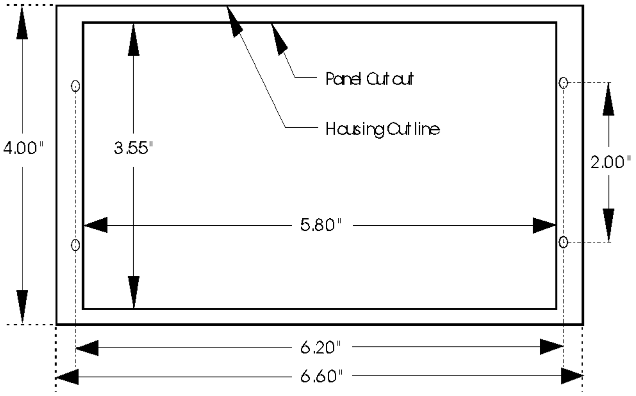
Pushbutton Label Strip Pattern



Specifications

Physical

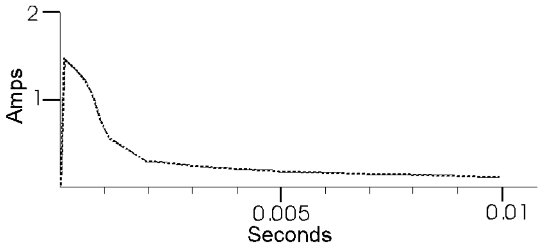
- Recessed Mount Housing 6.6"Lx4.0"H x 1.25"D
- Cutout size for above 3.55"Hx5.80"L
- Panel Fasteners : Four, 6x32 threaded studs, shown above (on ends, symmetrical about center line)
- Weight : 10 ounces
- Colors : Dark gray housing with dark gray panel. Keypad keys; white with dark gray letters. White with user supplied label.
- LCD Display : 2 line x 20 character STN with LED backlight character size : 5.5mm high x 3.2mm wide



Panel Mounting Dimensions

Electrical

- Power : 8 - 30VDC @ 1.8Watts
  150mA @ 12VDC 75mA @ 24VDC
- Power on surge (see figure below)
  1.5A for 2 milliseconds maximum



Power on Surge

- Power connector : Pluggable terminal block, 2 position

Communications

- RS232 and RS422
- 4800 to 19200 baud
- Compatible with most major PLC protocols
- OptiMate Hex protocol for computer based systems
- 15 pin female 'D' shell connector

Communications Failure Operation

Should the module (when not selected for configuration) ever fail to communicate successfully for a period of 12 seconds, 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)
- Arrow adjustment data message (integer or fixed point)

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)