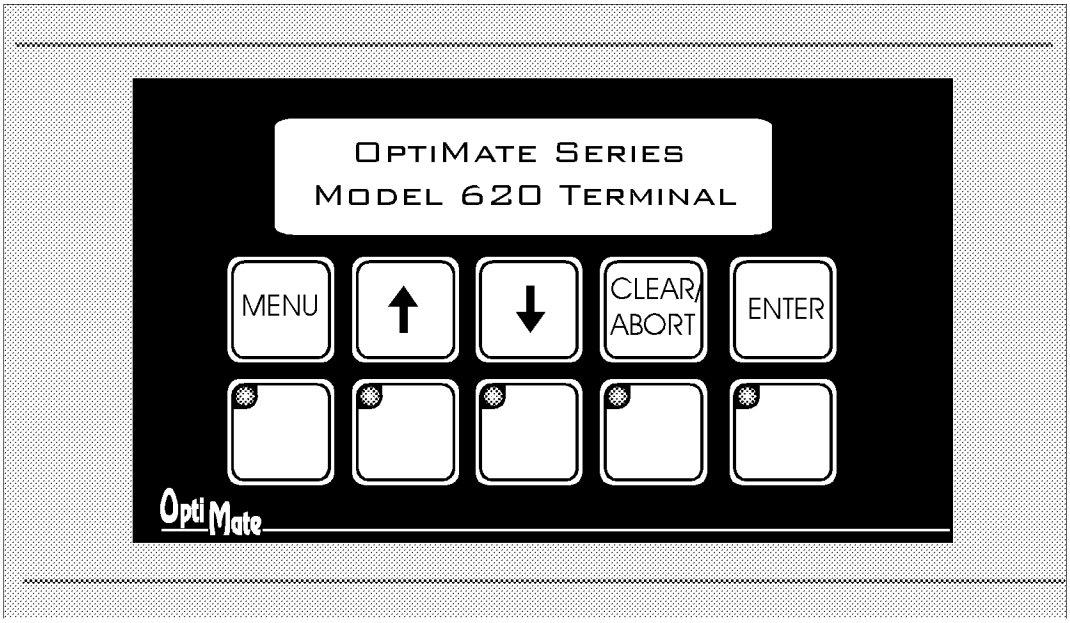



# OP-620 Operator Panel

The OP-620 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.

OP-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, 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 OP-620 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 OP-620 takes care of the rest.

**Applications**

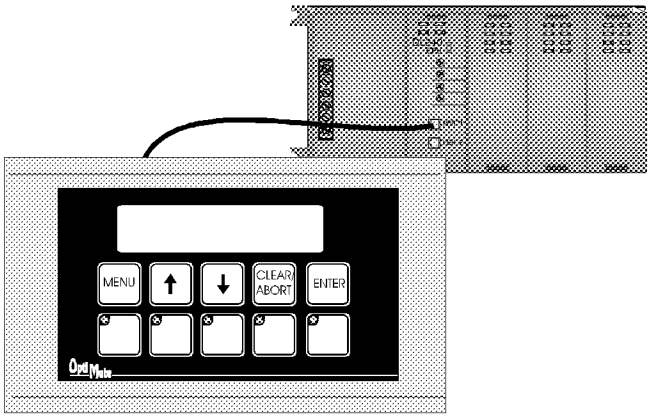
- Machine control
- Process control
- Security systems
- HVAC
- Plant monitoring/control
- PLC applications

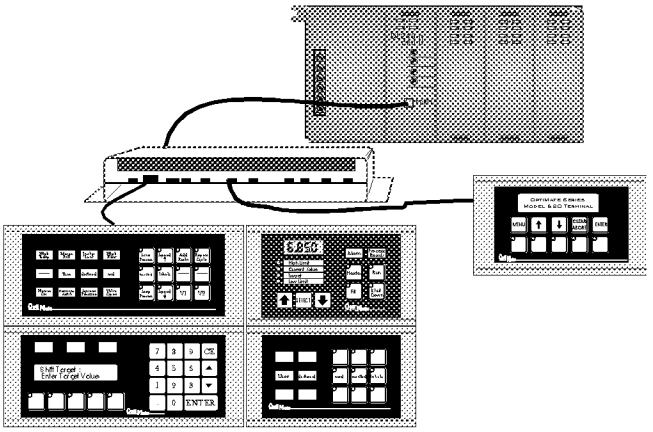
**Features**

- 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
- 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 OP-620 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 OP-620 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.

## PLC Multi Panels

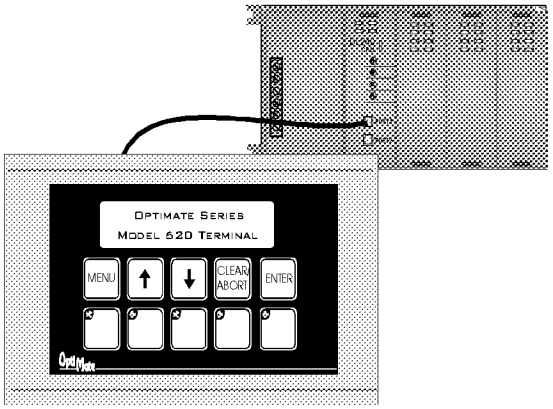
Larger systems involving operator panels and I/O can be successfully addressed using OptiMate panels. These applications utilize the OP-9001 Communications Master to transfer data between the PLC and the individual OptiMate modules. OptiMate panels can be located together to form custom panels or they can be distributed anywhere within 4000 feet.

The OP-9001 Communications Master provides a transparent interface between the PLC and a group of OptiMate panels. The communication interface between modules requires only four wires.

System configuration is simple using OP-WINEDIT software that runs on any IBM PC compatible computer.

This modular approach to custom applications provides a nearly limitless number of possibilities.

# 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 OP-620 Operator Panel uses a bank of 9 contiguous PLC registers. The register set definition is shown in the table below.

**OP-620 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) |
| M+4 | Bottom line data |
| M+5 | Bottom line data 2 (for long BCD) |
| 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 or 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 or 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 "unlock" 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 OP-WINEDIT configuration software, up to 160 predefined messages can be entered and stored in the OP-620. 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 OP-620 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.

#### *Displaying Fixed Point Numbers*

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 these type numbers, 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 OP-620 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 OP-620 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 OP-620 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 OP-WINEDIT software.

Floating point numbers can only be used with the PLC*Direct* DL250, DL350 and the DL450 CPUs since they are the only compatible 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.

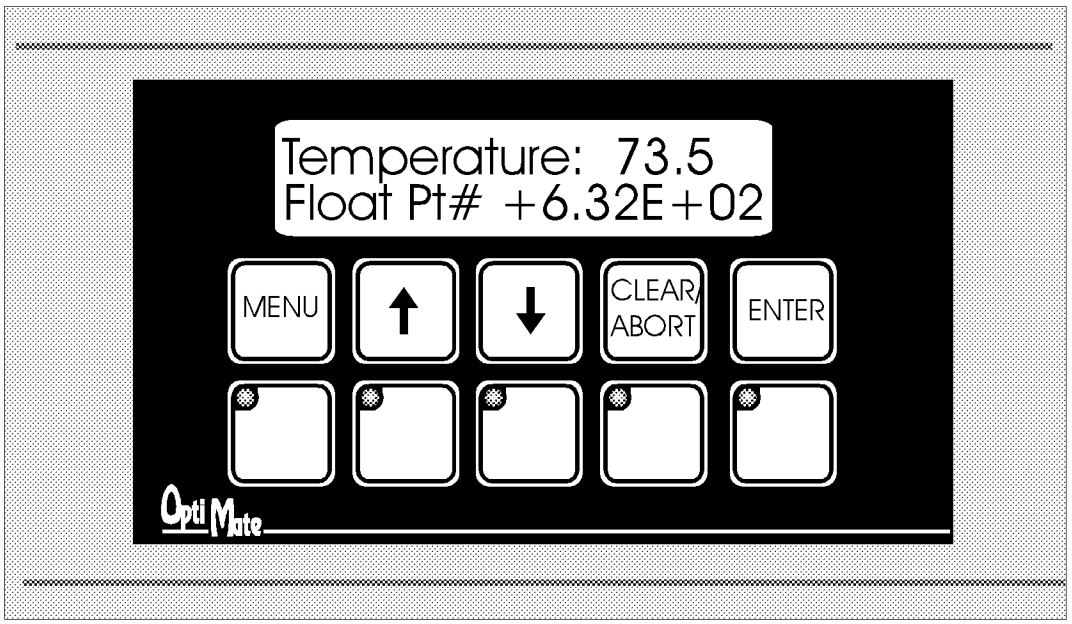
An IEEE 32-bit floating point number has a range of -3.402823E+38 to +3.402823E+38. The OP-620 will be able to display any number within that range. The panel always uses the format ±X.XXE±XX to display the numbers.

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 OP-620 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 | OP-620 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 "Floating Point" option for the data format.

Suppose you wanted to configure message #58 to display a floating point number. In the OP-WINEDIT software you would select OP-620 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.*

With the exception of floating point, 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 OP-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 limited to within the limits defined through the configuration editor. Data also will be limited to 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 OP-620 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 OP-620 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.

### *Menu Tree Operation*

The OP-620 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 OP-620 allows up to four levels of menu. Each menu selection can be either a function or the next lower menu level. 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. If the ENTER key is pressed while a function "Process Part Type 1" is displayed, the associated function number will be writen to X+6 and the FS flag will be set within the PLC.

The top level menu selection "Manual Control" is a sub-menu. If it is selected via the ENTER key, the second menu level will appear. In this case, each item on the second level is a function. If selected, each will place it's function number in X+6 and also set the FS flag.

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. In this branch, the items on the first and second levels are sub-menu items, the items on the third level are functions.

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. The OP-620 will be "locked up" until the PLC program decodes the function number and clears the ME flag.** Once the ME flag is cleared, the panel will return the LCD display to the control of your program.

Once you have completed your function, you may return to the same point in the menu by setting ME and MR. To require the user to start the menu selection process from the beginning, don't set MR, only set ME.

The menu enable is totally under PLC program control. If ME is set, the menu operation is enabled. If for any reason you need to take control of the display back from the menu, just clear the ME bit. The panel will leave the menu tree and display the message numbers in X and X+1.

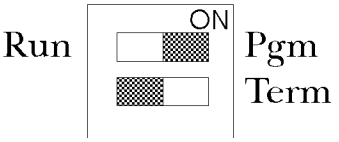
***Each menu tree message takes up one of the 160 available total messages.***

# Configuration

Configuration of the OP-620 Terminal or system of OptiMate modules is performed via an IBM PC compatible computer with the Windows operating system. Optimation supplies OP-WINEDIT software that will allow you to select module configuration, system configuration and PLC protocol definition.

If the OP-620 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 OP-WINEDIT.

> Note : When configuring, always remember to set Run/Pgm to Pgm (towards the "ON"). Always wait for the module to reset( approximately 2 seconds) before configuring the module.




Specific configuration of the OP-620 begins with defining the block of PLC register data 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 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.

# Examples of Use with a PLC*Direct* PLC

## Register Usage

The OptiMate OP-WINEDIT software allows you to configure a module to use a block of registers at a starting value that you define. For a PLC*Direct* DL105, DL205, DL350 or DL405 PLC the recommended memory to use is the general purpose data words starting at V2000 and V4000. For the 305 family, the recommended memory is the registers beginning at R400. Any block of registers within the data word range can be used.

The first seven PLC registers in the block used by the OP-620 panel are used for numeric information. As such they are ideally suited for the general purpose data registers (V2000 and V4000 area for the DL105/DL205/DL350/DL405 and R400 range for the 305). The last two registers use individual bits for control and status. These registers are better suited for the control relay register range of memory. The solution to this minor conflict is to define the base register address in general purpose data register memory and place a rung in your PLC program to copy the last two registers to/from control relay registers.

The following table lists the control relay register addresses for the various PLC Direct PLCs.

| PLC Direct CPU | Control Relay Register address assignment |
|---|---|
| | |
| DL130 | V40600-V40617 |
| DL230 | V40600-V40617 |
| DL240 | V40600-V40617 |
| DL250 | V40600-V40617 |
| DL330 | R016-R037 |
| DL330P | R016-R017 and R020-R027 |
| DL340 | R016-R037 and R100-R106 |
| DL350 | V40600-V40617 |
| DL430 | V40600-V40635 |
| DL440 | V40600-V40677 |
| DL450 | V40600-V40777 |

## Memory Mapping

The examples on the following pages use an OP-620 connected to a PLC Direct DL105/DL205/DL350/DL405 series PLC. The OP-620 is configured for a base address of V2000. The program rung on the right should be placed in the program to copy the status register to V40600 (VC0) and copy from V40601 (VC20) to the control register. This is necessary because the OP-620 writes to the status register and reads from the control register.

With this rung placed in the PLC program, the status and control bits will be control relays. The register association is shown in the figure below.

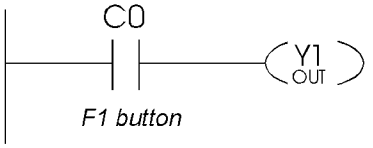
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | bit | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSB | | | | | | | | | | | | | | | LSB | | |
| Register V40600 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | | Status Register |
| V40601 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | | Control Register |

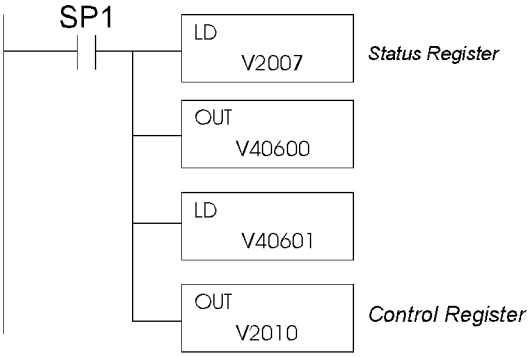
This will result in the following control relay association for the status and control registers.

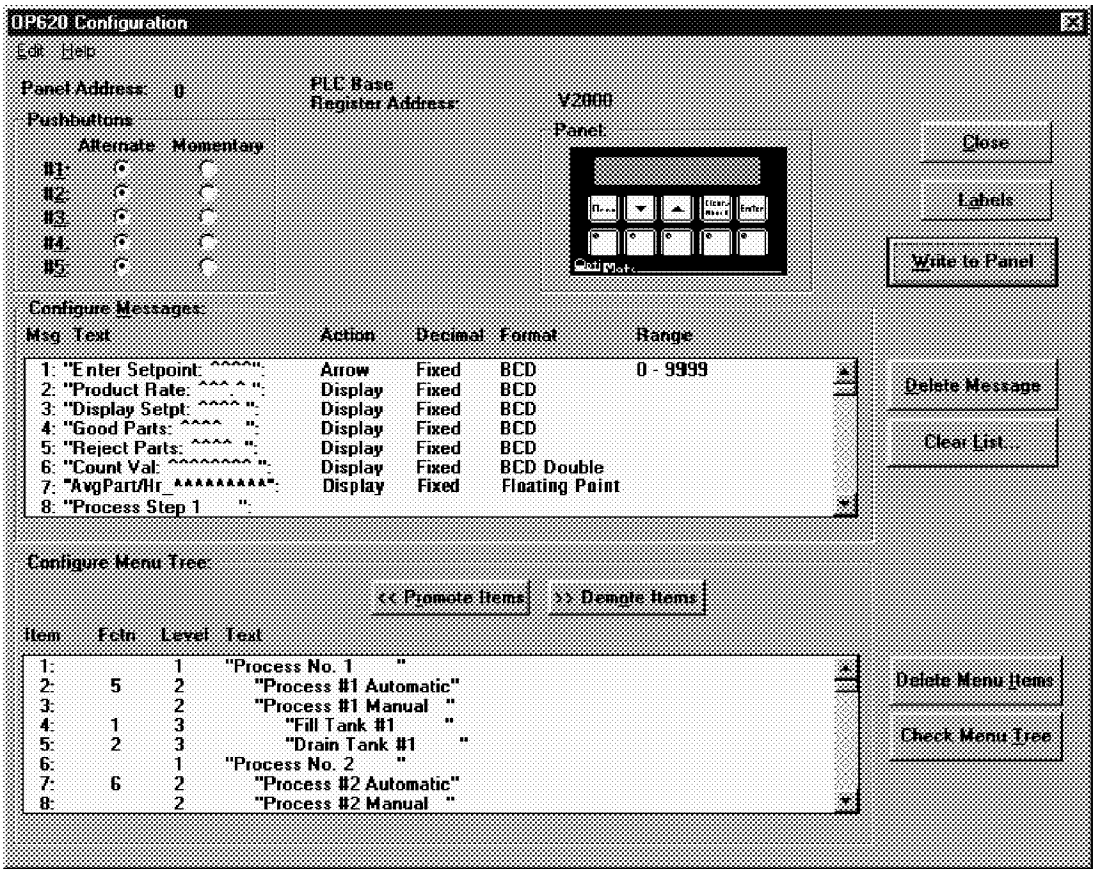
| Status Register | | | Control Register | |
|---|---|---|---|---|
| bit | relay | | bit | relay |
| | | | | |
| F1 | C0 | | ME | C20 |
| F2 | C1 | | MR | C21 |
| F3 | C2 | | DAK | C22 |
| F4 | C3 | | BD | C23 |
| F5 | C4 | | | |
| DA | C5 | | | |
| FS | C6 | | | |
| AB | C7 | | | |
| MA | C10 | | | |
| ENTER | C11 | | | |
| Up arrow | C12 | | | |
| Down arrow | C13 | | | |

## Using a Function Button

The five function buttons below the LCD display will appear as control relay coils in your program (assuming the memory mapping rung described previously is in your program). The following example turns on output Y1 when button F1 is active.



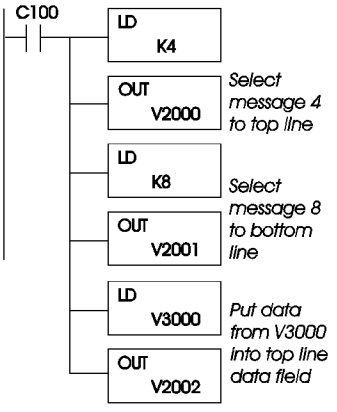



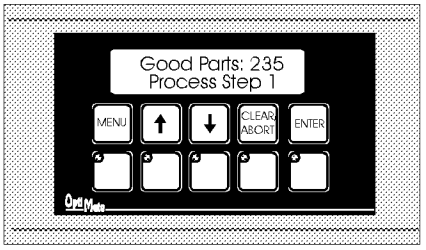



# Displaying Messages on the LCD Display

Messages of various types can be configured via OP-WINEDIT and downloaded to the OP-620. The message definitions shown in the figure above will be used in all of the examples that follow.

The following example 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 V3000. The bottom line is text message 8.

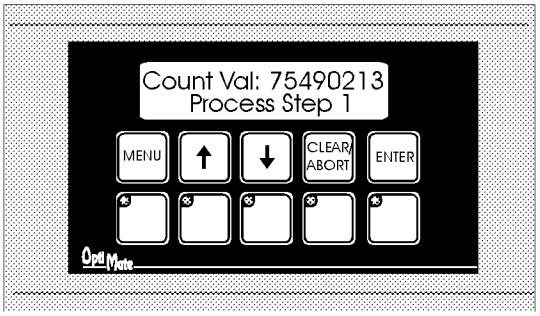


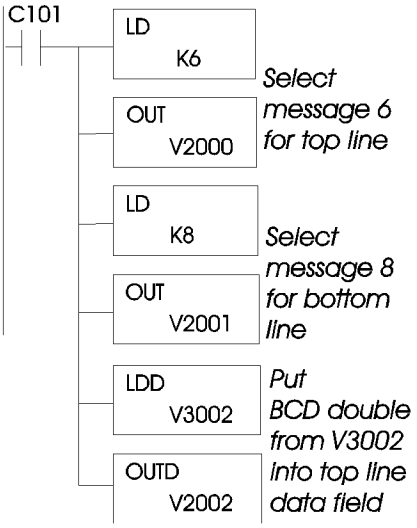




| | |
|---|---|
| V2000 | Top line message selection |
| V2001 | Bottom line message selection |
| V2002 | Top line data |
| V2003 | Top line data 2 (for long BCD) |
| V2004 | Bottom line data |
| V2005 | Bottom line data 2 (for long BCD) |
| V2006 | Function Selection |
| V2007 | Status register |
| V2010 | Control register |

## Displaying long BCD Numbers

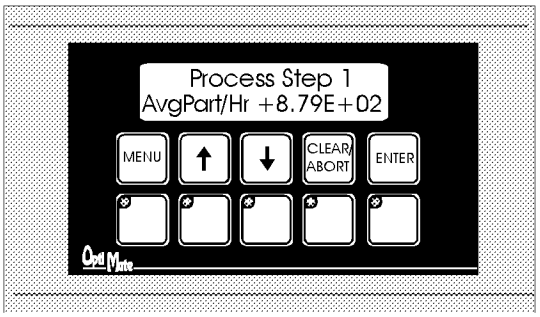
The example shown on the right is similar to the last example. The primary difference is that it uses a BCD double number in the top line display. The top line uses data display message 6, which has been configured as a BCD double display. The data for the data field is coming from V3002 (&V3003). The bottom line is text message 8.

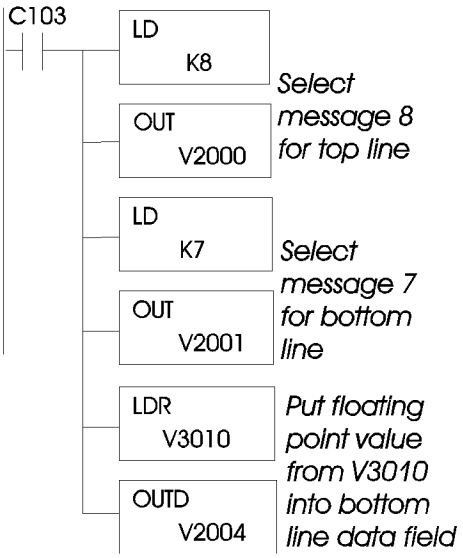



## Displaying Floating Point Numbers

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 V3010 and V3011 using the LDR (load real number) instruction to the bottom line display registers V2004 and V2005. The top line is text message 8.



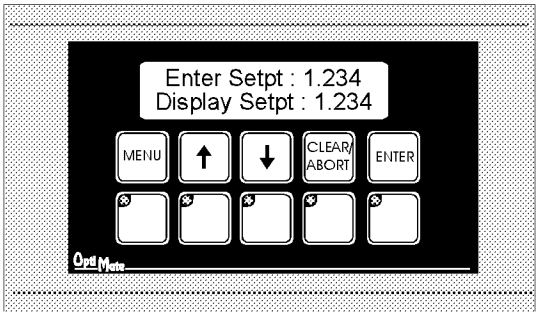



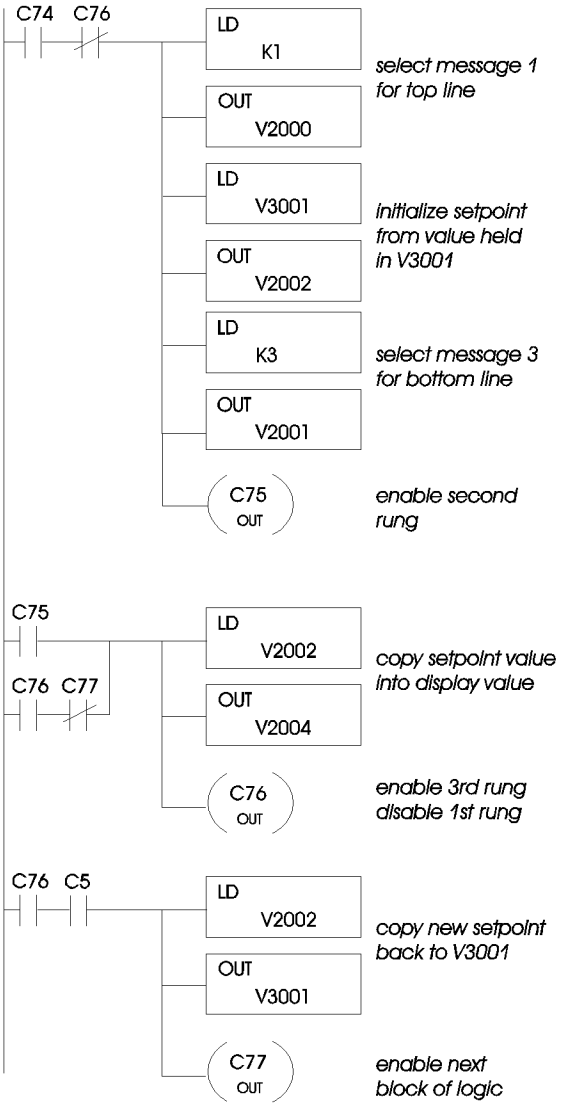
## Arrow Adjustment of Setpoint Data

The figure to the right illustrates arrow adjustment of setpoint data (and a whole lot more). The process is enabled when C74 is energized. The first program rung places the "Enter Setpoint" and "Display Setpt" messages in the top and bottom lines, initializes the setpoint value from the value in V3001 and enables the second rung.

The second rung continually copies the setpoint value to the display value so long as it is enabled. It latches itself until unlatched by the next rung while disabling the first rung.

The third rung waits until the data available flag is set (C5), then copies the setpoint back to V3001. It also unlatches the second rung and, by activating C77, enables the next block of logic (whatever that might be) in the program.



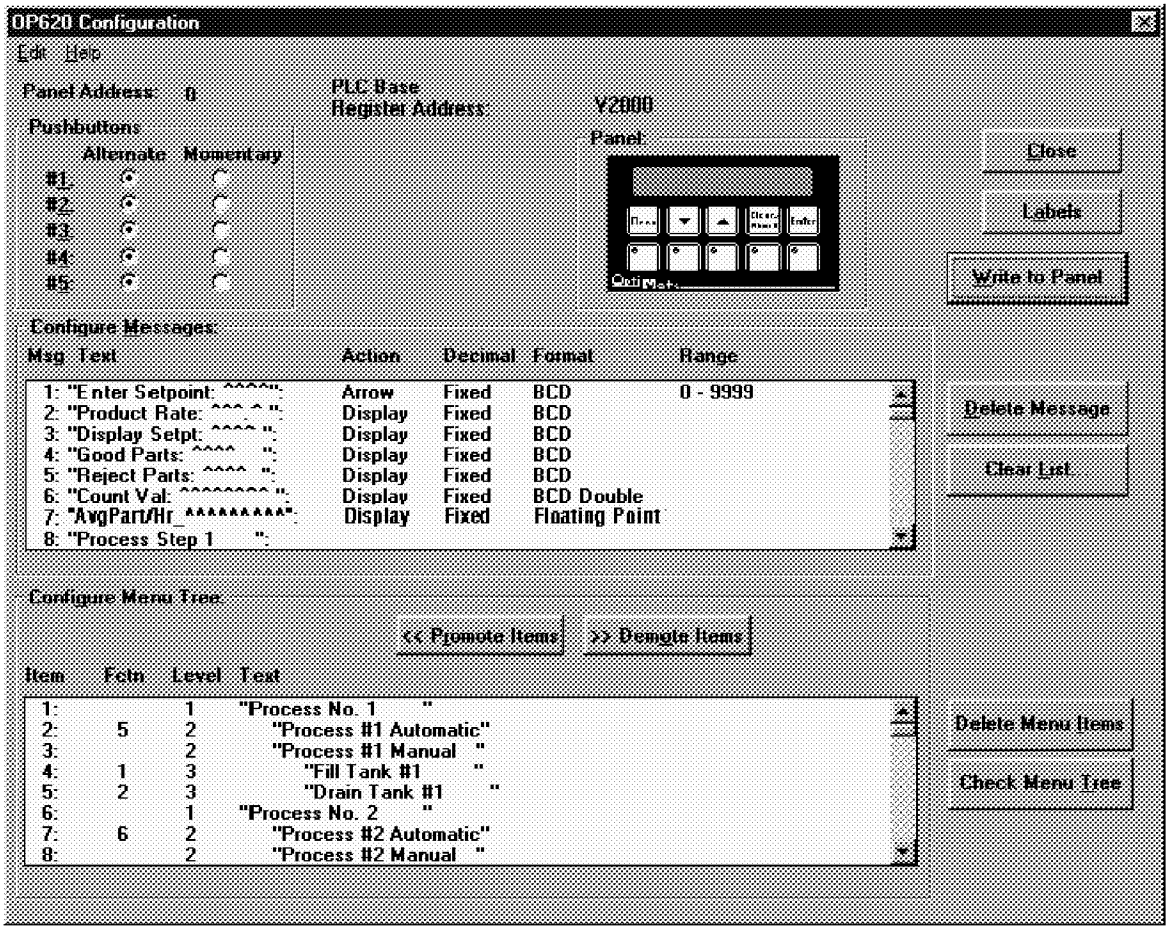
## Using a Menu Tree

The OP-620 allows you to predefine and use a layered menu tree for function selection. The operation of this menu tree is taken care of entirely by the OP-620 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.

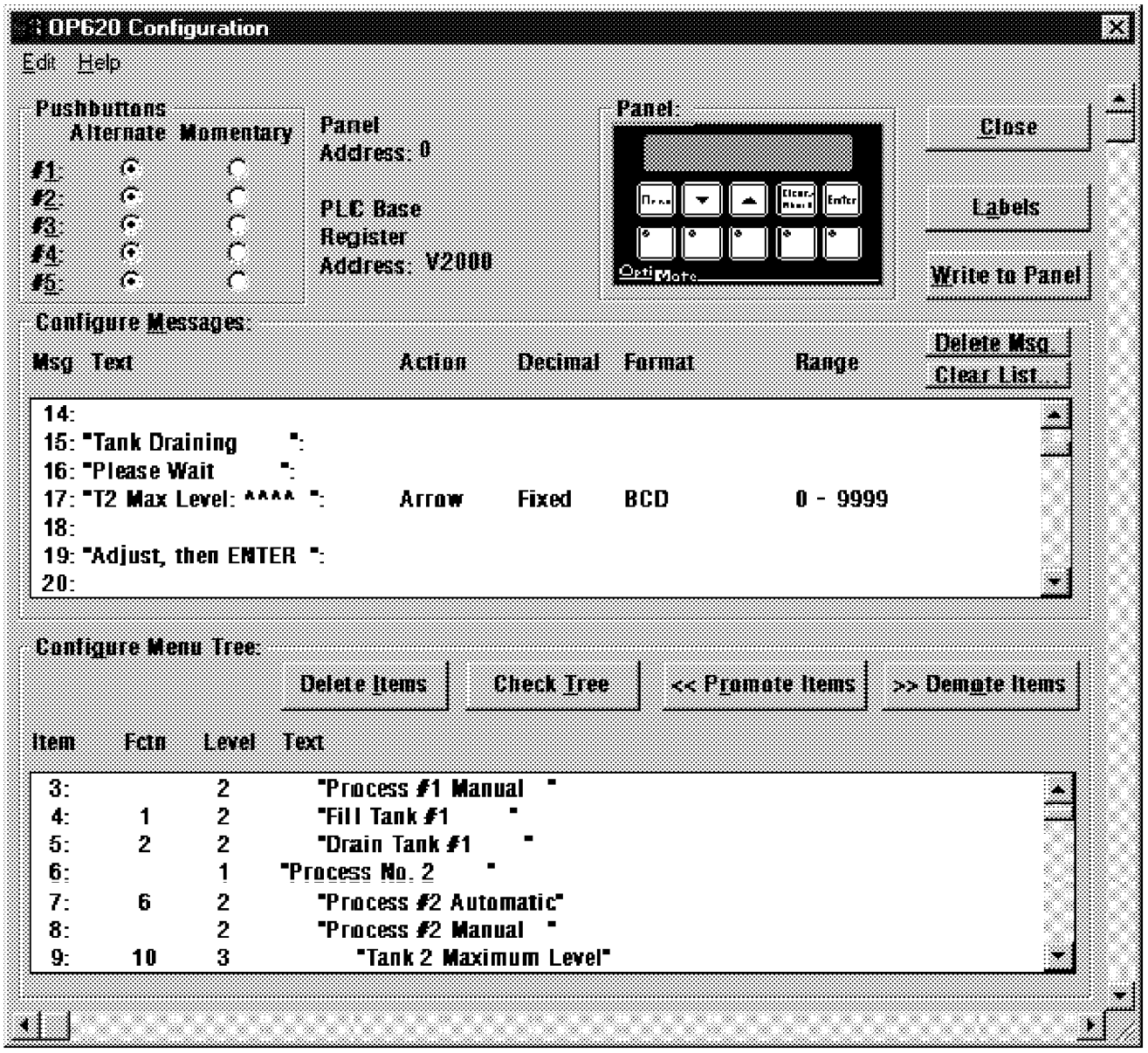
The OP-WINEDIT Editor screen shown below displays the menu structure,including function associations, on the lower section of the display. For example, if the operator selects "Process #1 Automatic" from the menu tree, function number 5 would be placed in the function select register. If "Process No.2" 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 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.



## A Menu Tree Example

The following pages show examples using a Menu Tree. They show how to determine when a function has been selected and how to decode a function number. The examples also show how to implement a function with text display messages and how to implement a function containing arrow adjustment of a setpoint. The base register address for the following examples is V2000. The examples use the messages and the menu tree shown in the figure below.

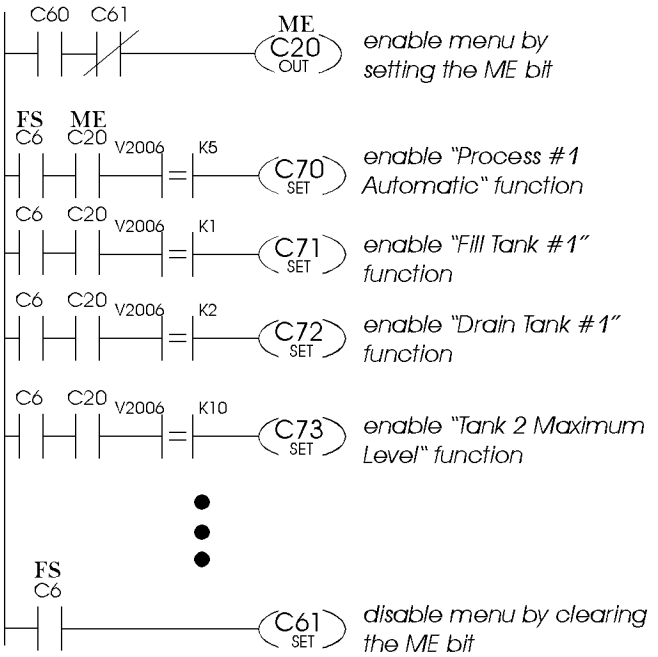


## Decoding a Function Number

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 when C60 is active.

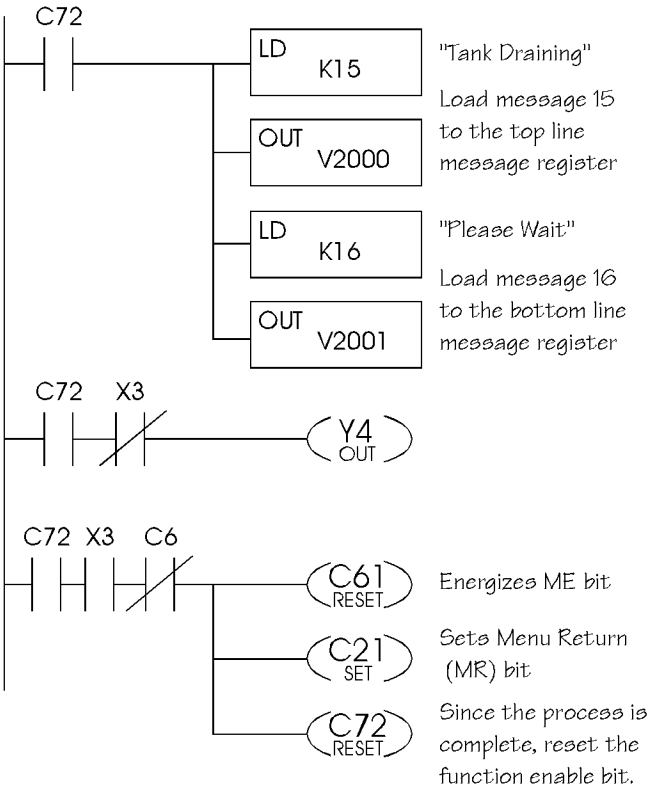
> Note : Enabling the menu tree does not automatically put the terminal into the menu. Once enabled, the terminal will bring up the menu tree when the operator presses "MENU". Until then, the display is under PLC program control.

The second rung sets the appropriate function enable bit when a selection is selected from the OP-620. Function selection will activate the FS flag (which, based on our register copy shown earlier, will be C6). The value held in the function register V2006 (M+6) is compared to a constant value. When one of the comparisons becomes true, that rung will set the appropriate enable relay (C70,C71, etc.). The last rung shown will disable the Menu Enable (ME) bit by setting C61. Until the ME bit is reset, the Menu may appear to "lock up" when a function is selected. Note that the logic shown interlocks the function number comparison (setting C70, C71, etc.) with the function select and menu enable flags to ensure that only one function is enabled each time a function is selected.



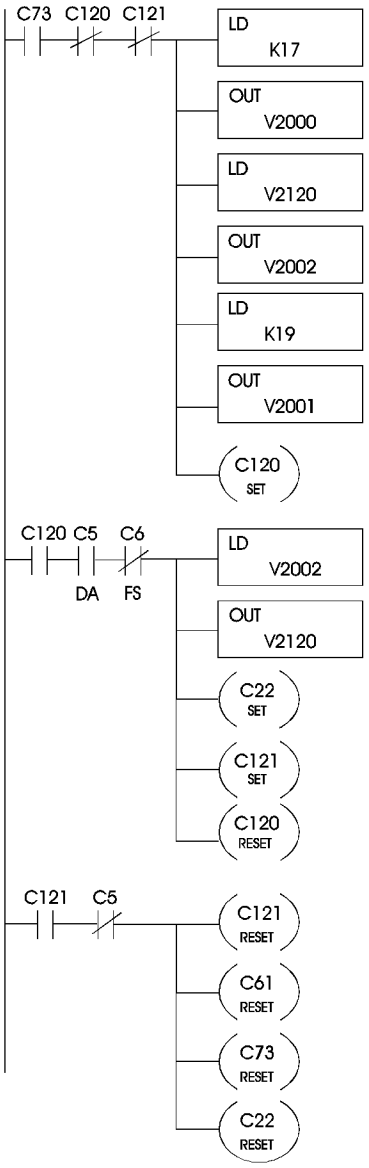

## Implementing a Menu Function

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 just shown, this would result in control relay C72 being set. The first rung of the "Drain Tank #1" function, shown below, will put messages into the top and bottom lines of the display. The second rung energizes output Y4 to open the drain until level sensor input X3 senses that it is empty. The third rung re-enables the menu and sets it to return back to the "Drain Tank #1" selection, via the MR (menu return) bit, when the tank is empty. The third rung also disables the function. By putting the function select relay (C6) in series, we force the program to wait until the function select has been cleared before re-enabling the menu.



# Implementing Data Entry using a Menu Function

The logic shown below is another typical example of how a setpoint function might be implemented. Suppose the selection for "Tank 2 Maximum Level" was made from the menu tree. From the logic on the previous page, this will result in C73 being set. Accordingly, the first rung shown below selects the appropriate setpoint message for the top line and a prompt message for the bottom line. It also initializes the setpoint value for arrow adjustment. The first rung disables itself and enables the second rung. When the setpoint data is entered (after the FS flag is cleared) the DA bit will be set. Then the second rung will copy the setpoint value back to its working location, set the DAK bit, and pass control to the third rung. The third rung waits until DA bit has been cleared by the OP-620, then clears the DAK and re-enable the menu. If the menu return flag is also set, the OP-620 will return to the same point in the menu.




"T2 Max Level : ^^^^"

Load message 17 to the top line message register.

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

"Adjust then ENTER"

Load message 19 to the bottom line message 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.

## Example Program using an OP-620

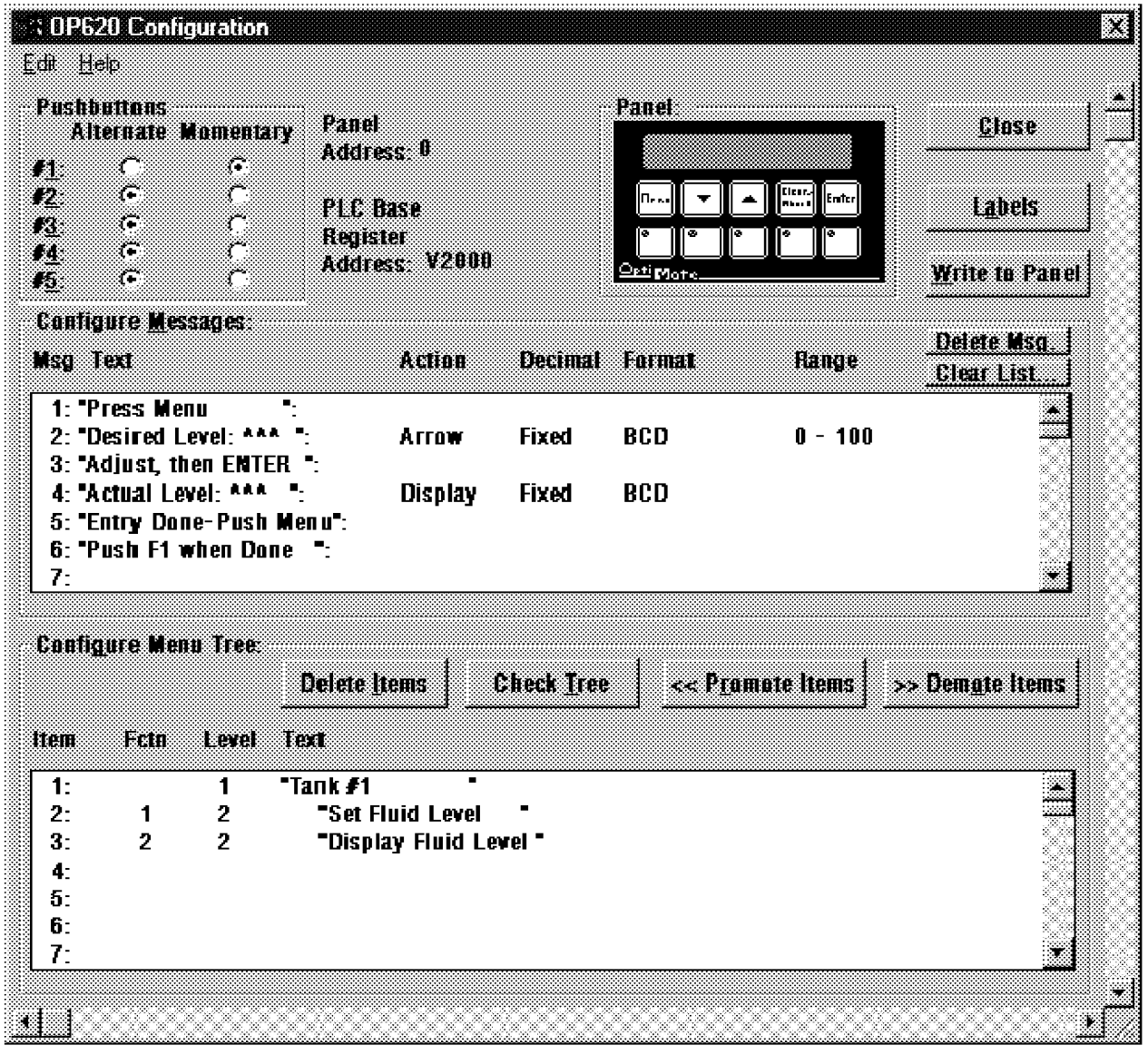
The example program shown in the following 4 pages describes how to implement a program using an OP-620. The program uses the menu tree, an arrow adjust message, a data display message and text messages.

It is a complete program which will run on a DL105, DL205, DL350 or DL405 PLC. The program uses a two level deep menu tree with two menu functions. The first menu level contains only a single sub-menu message. The second menu level contains two messages, both of which are functions.

The OP-620 configuration for this example is shown in the figure below. The base register address is V2000. The first pushbutton (F1) is configured for momentary action and the rest for alternate. The first pushbutton is the only one used for this example.

The program implements the following tasks:

- Initializing the display
- Mapping the status and control registers to/from control relay registers
- Setting and Resetting the Menu Enable bit (ME)
- Decoding a function number
- Implementing a function containing an arrow adjust message
- Implementing a function containing a data display message
- Displaying text messages
- Using a function key (pushbutton)
- Using the CLEAR/ABORT button to escape from a menu or function
- Resetting the Data Available (DA) bit using the Data Acknowledge (DAK) bit



### *Example Program*

*Rung 1*

This rung only happens on the first program scan. It places the initial messages onto the OP-620's LCD display. It loads message 1 "Press Menu" to the top line of the display. Message 7, which is blank, is loaded to blank the bottom line of the display. The control relay C60 is set to energize the menu enable (ME) bit via rung 3.

*Rung 2*

This rung is the Memory Mapping rung. It should be active every scan. Since it is easier to treat individual bits (pushbuttons, DA, ME, DAK, etc.) as coils and contacts within the program, a memory mapping scheme is needed to mimic the status and control registers. This allows the individual bits of the status and control registers to be treated as coils and contacts.

The OP-620 writes data to the status register (V2007) so it's data can be copied to a control relay register such as VC0 (V40600). Therefore, the program can look at the corresponding bits within register VC0 to determine if a button has been pressed, DA is set, etc.

The OP-620 reads from the control register (V2010). Therefore, a control relay register such as VC20 (V40601) can be designated to mimic the panel's control register. The corresponding bits in VC20 are updated in the PLC program, then that register is mapped (copied) to the panel's control register. The panel will read the updated control register and act accordingly.

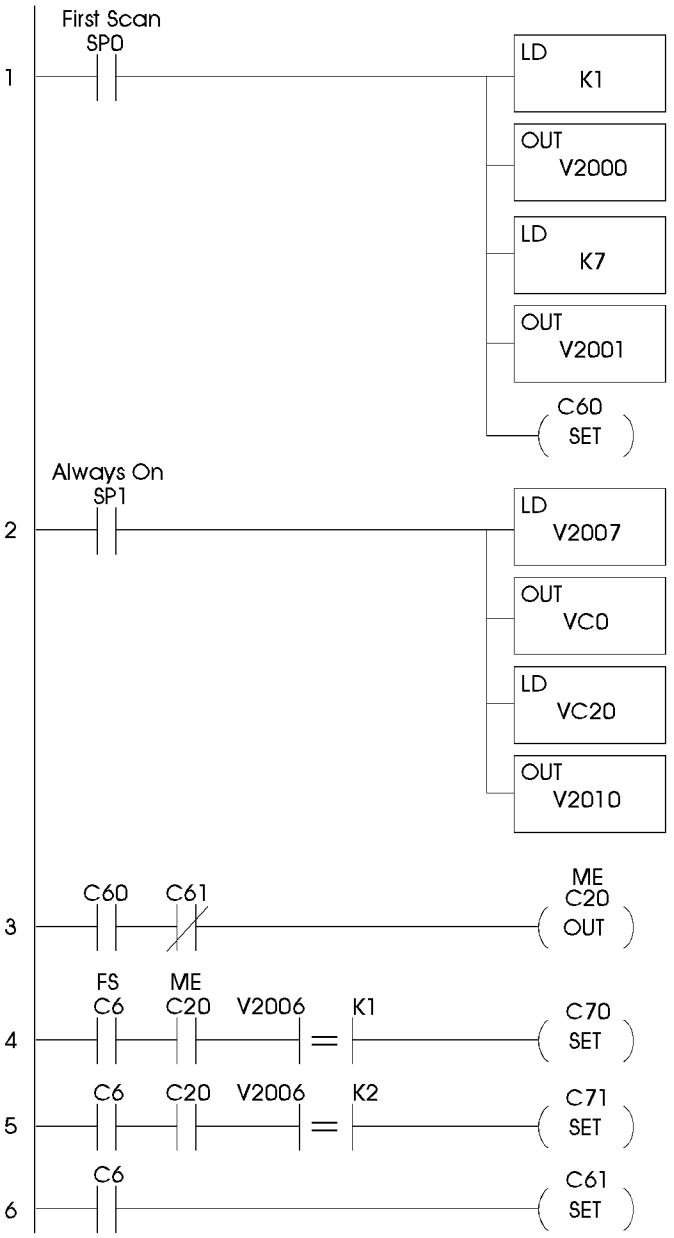
*Rung 3*

This rung sets the menu enable (ME) bit . This bit allows the user to enter the menu tree when the MENU button is pressed.

*Rungs 4 and 5*

Rungs 4 and 5 are used to determine which function was selected by the OP-620's user. Notice that both the function select (FS) and the menu enable (ME) bits are placed in the rung to ensure that a function has been selected before it determines which function the user selected. After it determines that the FS and ME bits are set, the function number is decoded to determine which function the user seleted. To decode the function number, selection register (V2006) is compared with a constant number. The constant number (K1, K2, etc. ) represents the desired function. Once the function number has been determined, a coil is set. This coil is used only for this particular function.

Rung 4 represents function 1 and rung 5 represents function 2. These rungs can be duplicated for all the function numbers within the program. Later in the program, a rung needs to be added for each bit that starts a function process (C70, C71, etc.). When that particular bit has been set, the program knows which function was entered and then the program can implement that function using whatever logic is necessary. Two examples are described within this program.




*Rung 6*

This rung disables the menu enable (ME) bit via C61. It checks for the function select (FS) bit. When the FS bit is set, the program knows that a function has been selected. If the function number has been decoded correctly as shown in the previous two rungs, then the panel needs to exit the menu so the program can implement the function.

The only way to leave the menu is to disable the ME bit. If the ME bit is not disabled after a function is selected, the OP-620's display will "lock up." When the ME bit is reset, the panel will leave the menu mode and the display will return to PLC program control.

The program is continued on the next page.

### *Example Program*
*(Continued from previous page)*

#### *Rung 7*

This rung is the first rung in the logic for function 1, defined in the menu tree as "Set Fluid Level". When C70 is enabled and C100 is not, the function begins. The second predefined message, "Desired Level: ^^^" is loaded into the top line (V2000) of the display. Since message 2 is an arrow adjust message, the initial starting point for the adjustment has to be loaded into the top line data register, V2002. In this case, the data which is stored in V2210, is copied from V2210 to V2002.

Next, message 3 ,"Adjust, then ENTER", is loaded into the bottom line (V2001) of the display. Then, C100 is set which enables the next rung. A rung of this type which loads arrow adjustment messages should only execute one time within the function, therefore, C70 is reset to disable the current rung.

#### *Rung 8*

When the user is finished entering the data, the user will press the ENTER key as prompted by the bottom line message. The OP-620 will set the data available (DA) bit, C5. When the DA bit is set, rung 8 will copy the newly entered data from the top line data register, V2002, to the data's storage register,V2210. Next, the program sets C101 to enable the next rung and resets C100 to disable the current rung.

Notice that the function select (FS) bit, C6, is in series with C100 and the DA bit. This ensures that the panel has cleared the function select flag before allowing the program to accept the new data.

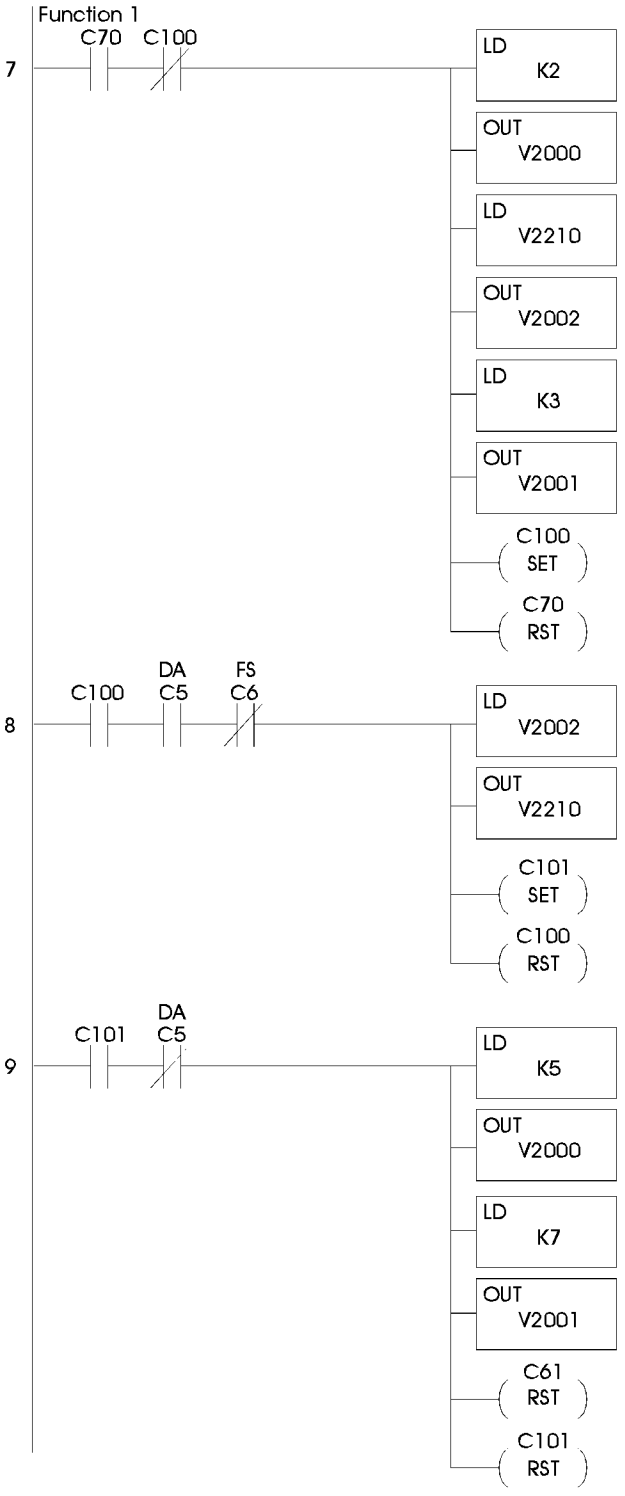
#### *Rung 9*

Rung 9 completes function 1. It waits until the OP-620 clears the data available (DA) bit, C5, before prompting the user for the next step.

When the panel clears the DA bit, two new messages are loaded to the display. Message 5, "Entry Done-Push Menu" is loaded to the top line of the display. Message 7, a blank message, is loaded to the bottom line of the display. Next, C61 is reset to re-enable the menu enable (ME) bit via rung 3. Finally, C101 is reset to disable the current rung.

After this rung is completed, the OP-620 will display the "Entry Done-Push Menu" message until the menu button is pressed.

The DA bit can be cleared two different ways. One is to replace the data entry message with another message, the other is to set the data acknowledge (DAK) bit, C22. In this example, the latter was chosen. Rung 14 explains that process in more detail.




The program is continued on the next page.

***Example Program***
*(Continued from previous page)*

*Rung 10*

This rung is the first rung in the logic for function 2, defined in the menu tree as "Display Fluid Level". When C71 is enabled and C102 is not, the function begins. The fourth predefined message, "Actual Level: ^^^", is loaded into the top line (V2000) of the display. Since message 2 is a data display message, the initial display value has to be loaded into the top line data register, V2002. In this case, the initial value, which is stored in V2211, is copied from V2211 to V2002.

Next, message 6 ,"Push F1 when Done", is loaded into the bottom line (V2001) of the display. Then, C102 is set to enable the next rung. Last, C71 is reset to disable the current rung.

*Rung 11*

Rung 11 updates the data field of the display message on the top line. The OP-620's user can view the data as it changes within V2211. As long as this rung is active, the top line data register, V2002, will be updated with the current data, stored in V2211.

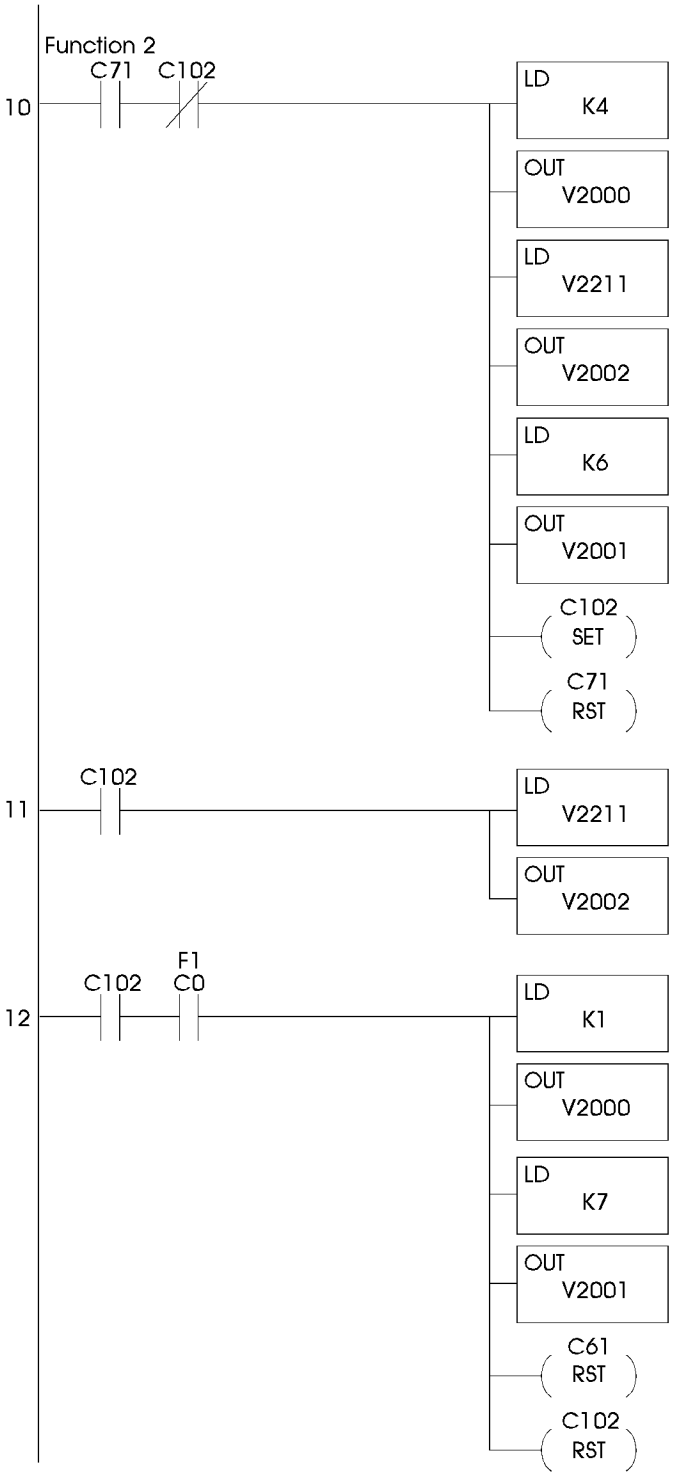
*Rung 12*

Message 6 on the bottom line prompts the user to "Push F1 when Done". When the OP-620's user wishes to stop viewing the data, they should press the first pushbutton, F1. This will complete function 2.

When F1, a momentary action button, is pressed, a new set of messages are loaded into the display registers . Message 1, "Press Menu", is loaded into the top line and message 7, a blank message, is loaded into the bottom line of the display. Next, C61 is reset to re-enable the menu enable (ME) bit via rung 3. Finally, C102 is reset to disable the current and previous rungs.

After this rung is completed, the OP-620 will display the "Press Menu" message until the menu button is pressed.

The program is continued on the next page.



***Example Program***
*(Continued from previous page)*

*Rung 13*

This rung is an abort rung. If the OP-620's user is in the menu or in the middle of a function and they realize that they need to exit without making any changes, then they can use the CLEAR/ABORT button for that purpose. If the panel is in the menu tree when the CLEAR/ABORT button is pressed, the panel will exit the menu and default back to the messages that are in the top and bottom line message registers. If the user is in a function, this rung can be used to reset different bits within the program that pertain to any given function which could be active at the time, C70, C71, C100, etc.

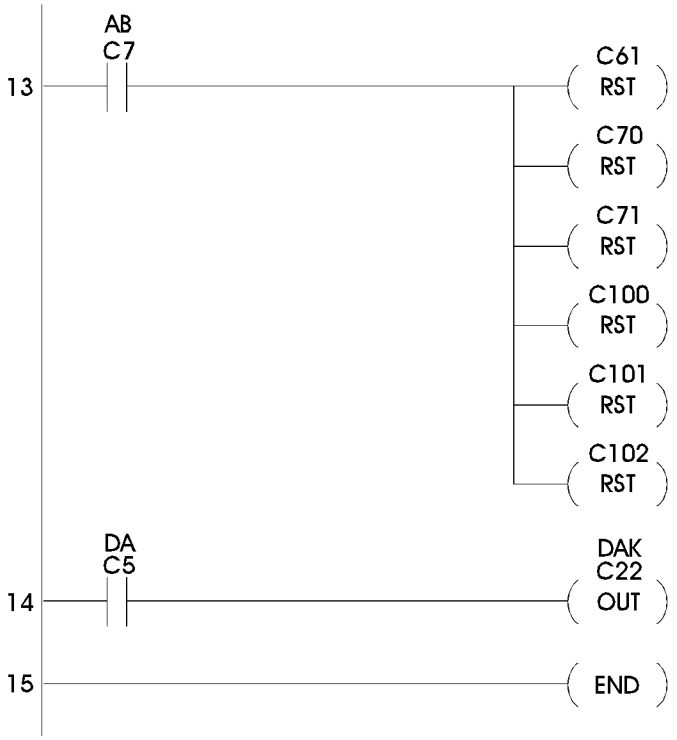
This rung can also be OR'd with other conditions, such as machine jams, emergency conditions, etc. that might warrant exiting the menu (by clearing the menu enable bit) or disabling a function. Messages can also be loaded to the display to describe the problem to the user if so desired.

*Rung 14*

This rung sets the data acknowledge (DAK) bit whenever the data available (DA) bit is set by the OP-620. It will stay active until the panel sees that the DAK has been set. At that time, the panel will reset the DA bit which will deactivate the rung. If you choose to reset the DA bit in this manner, this rung should be placed at the end of the program where it can be checked every scan.

*Rung 15*

This rung marks the end of the program.



# Examples of Use with an Allen-Bradley PLC

## Interfacing to A-B Memory

OptiMate panels interface to Allen-Bradley SLC 5/03, SLC 5/04 and Micrologix PLCs via integer file type N. The 5/03 and 5/04 have file type N7 as standard. Other "N" type files can be created. The Micrologix has a fixed file type N7. Please refer to Allen-Bradley documentation for information on setting up and using "N" type files.

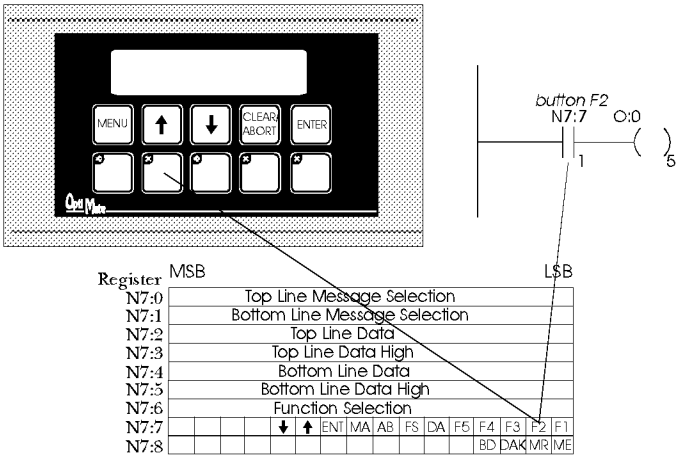
**Note: When using an OP-620 with an A-B PLC, always ensure that at least 9 words of memory are allocated to ensure proper communications.**

All of the examples that follow assume that the OP-620 module has been configured, through the OP-WINEDIT Editor, for a file type 7 and base register address 0. With this configuration, the status and control registers will be at N7:7 and N7:8 respectively. The following is a table relating status and control register bits to their N7 locations.

| Status Register | | | Control Register | |
|---|---|---|---|---|
| bit | location | | bit | location |
| | | | | |
| F1 | N7:7/0 | | ME | N7:8/0 |
| F2 | N7:7/1 | | MR | N7:8/1 |
| F3 | N7:7/2 | | DAK | N7:8/2 |
| F4 | N7:7/3 | | BD | N7:8/3 |
| F5 | N7:7/4 | | | |
| DA | N7:7/5 | | | |
| FS | N7:7/6 | | | |
| AB | N7:7/7 | | | |
| MA | N7:7/8 | | | |
| ENTER | N7:7/9 | | | |
| Up arrow | N7:7/10 | | | |
| Down arrow | N7:7/11 | | | |

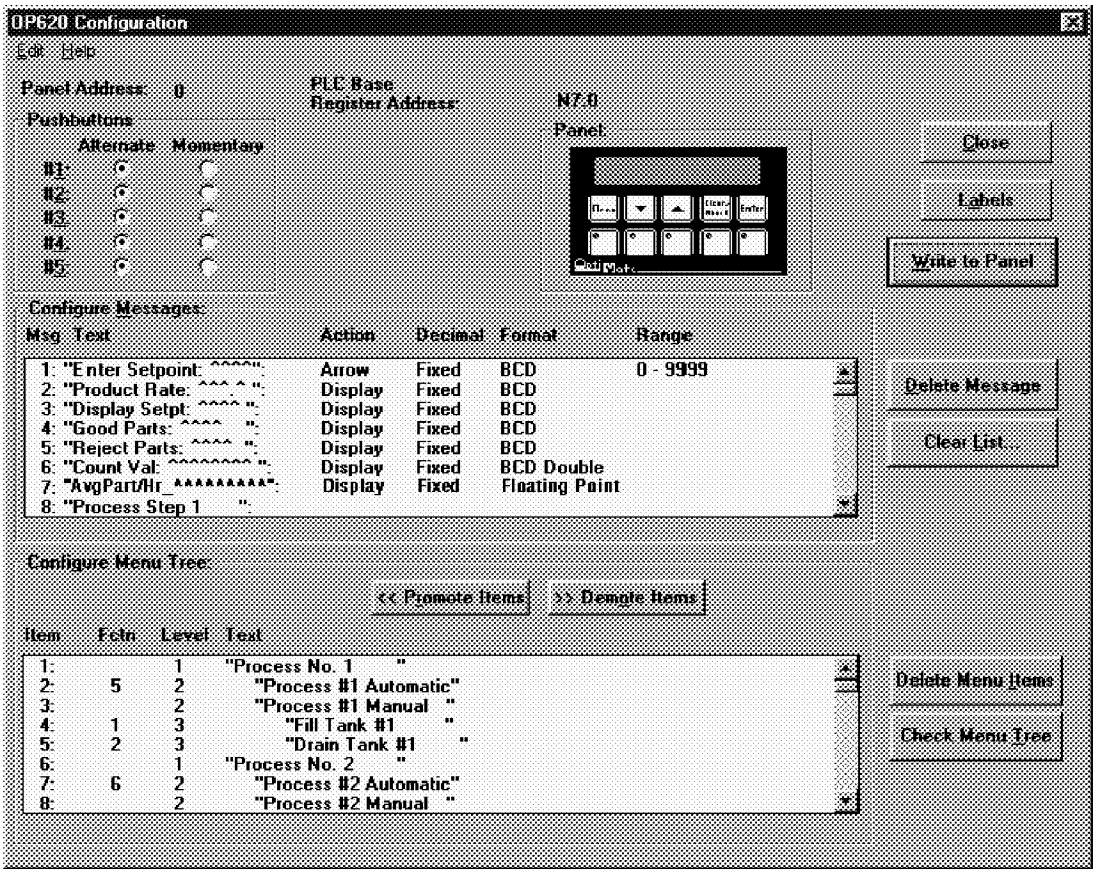
## Using a Function Button

The five function buttons below the LCD display will appear as control relay coils in your program. The following example turns on output O:0/5 when button F2 is active.




## Displaying Floating Point Numbers

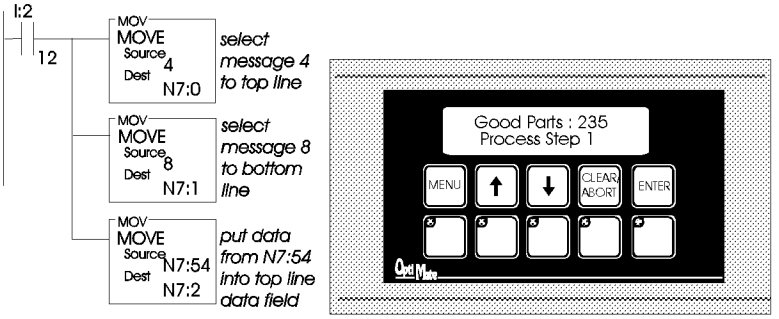
Floating point numbers can be displayed by the OP-620. This number format is a standard capability for PLC Direct DL250, DL350 and DL450 PLCs. However, the A-B SLC PLCs do not have a means of handling floating point numbers. Due to the limitations of the SLC, this capability will not be commonly implemented with A-B PLCs.




# Displaying Messages on the LCD Display

Messages of various types can be configured via OP-WINEDIT and downloaded to the OP-620. 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 N7:54. The bottom line is text message 8.

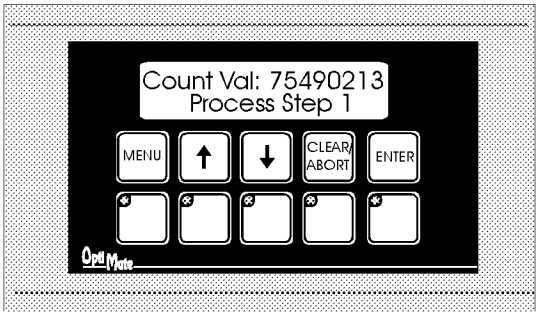


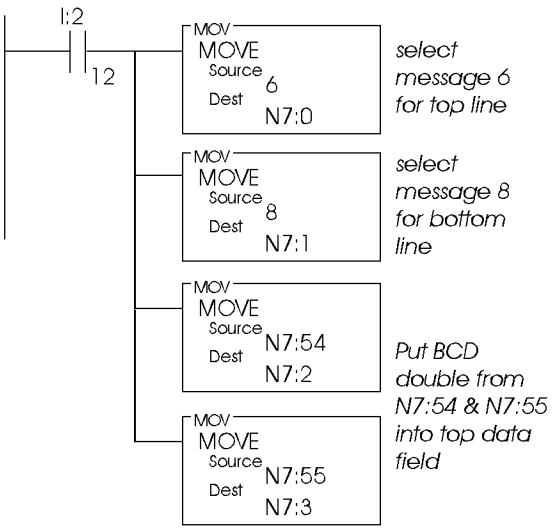

| Register | MSB | | | | | | | | | | | | | | | LSB |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| N7:0 | Top Line Message Selection | | | | | | | | | | | | | | | |
| N7:1 | Bottom Line Message Selection | | | | | | | | | | | | | | | |
| N7:2 | Top Line Data | | | | | | | | | | | | | | | |
| N7:3 | Top Line Data High | | | | | | | | | | | | | | | |
| N7:4 | Bottom Line Data | | | | | | | | | | | | | | | |
| N7:5 | Bottom Line Data High | | | | | | | | | | | | | | | |
| N7:6 | Function Selection | | | | | | | | | | | | | | | |
| N7:7 | | | | | ↓ | ↑ | ENT | MA | AB | FS | DA | F5 | F4 | F3 | F2 | F1 |
| N7:8 | | | | | | | | | | | | | BD | DAK | MR | ME |

## Displaying long BCD Numbers

Long (up to 8 digit) BCD numbers can be displayed by the OP-620. This number format is a standard capability for PLC Direct PLCs. However, the A-B SLC PLCs do not have an easy means of handling long BCD numbers. Due to the limitations of the SLC, this capability will not be commonly implemented.

The example in the next column illustrates the method for displaying large numbers. Registers N7:54 and N7:55 contain an 8 digit BCD number, with the most significant 4 digits in N7:55.



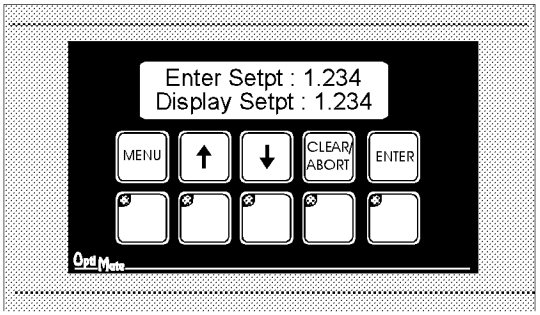



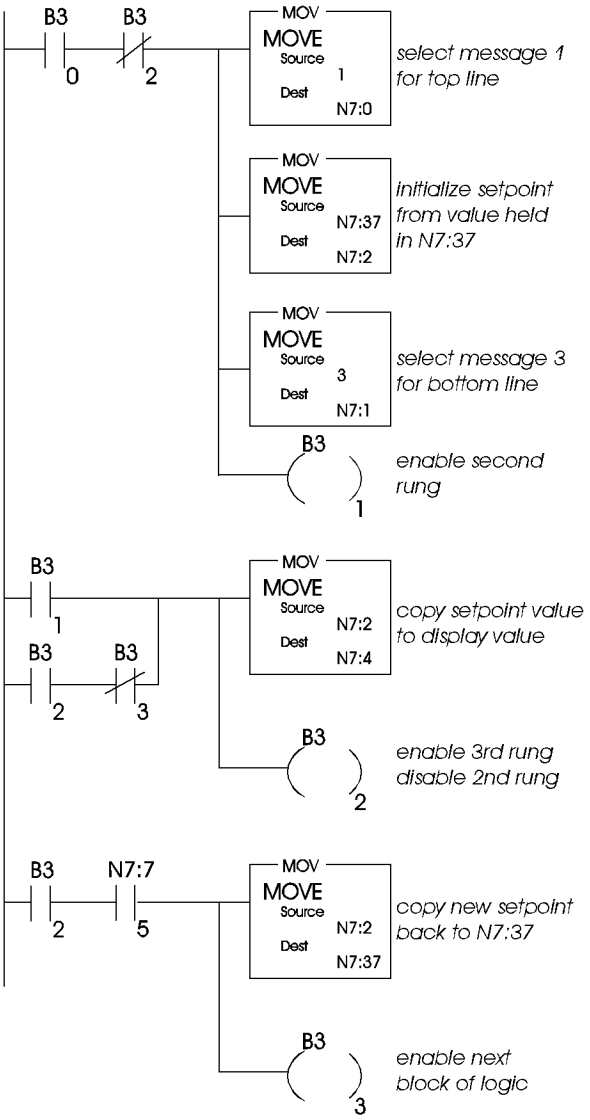

## Arrow Adjustment of Setpoint Data

The figure to the right illustrates arrow adjustment of setpoint data (and a whole lot more). The process is enabled when B3/0 is energized. The first program rung places the "Enter Setpoint" and "Display Setpt" messages in the top and bottom lines, initializes the setpoint value from the value in N7:37 and enables the second rung.

The second rung continually copies the setpoint value to the display value so long as it is enabled. It latches itself until unlatched by the next rung while disabling the first rung.

The third rung waits until the data available flag is set (N7:7/5), then copies the setpoint back to N7:37. It also unlatches the second rung and, by activating B3/3, enables the next block of logic (whatever that might be) in the program.



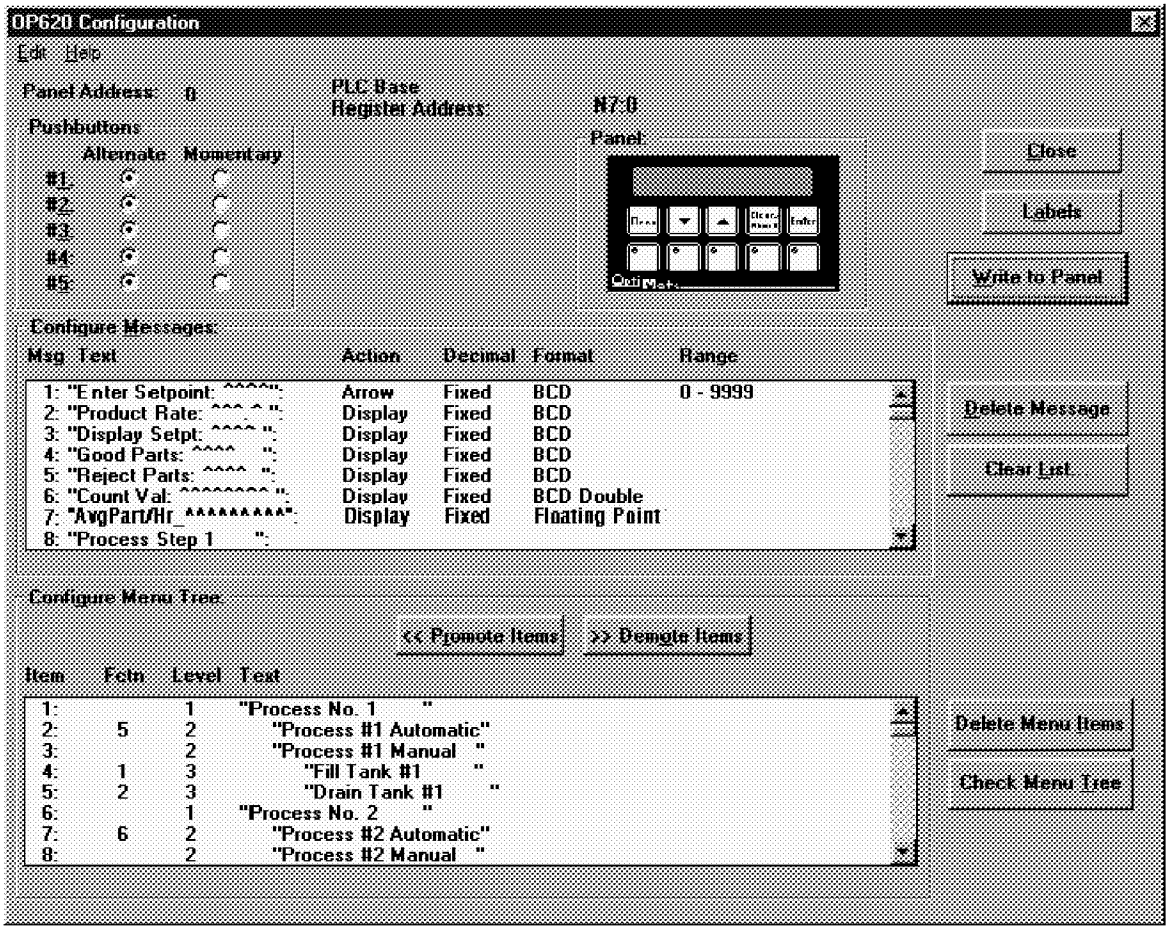
## Using a Menu Tree

The OP-620 allows you to predefine and use a layered menu tree for function selection. The operation of this menu tree is taken care of entirely by the OP-620 terminal. 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.

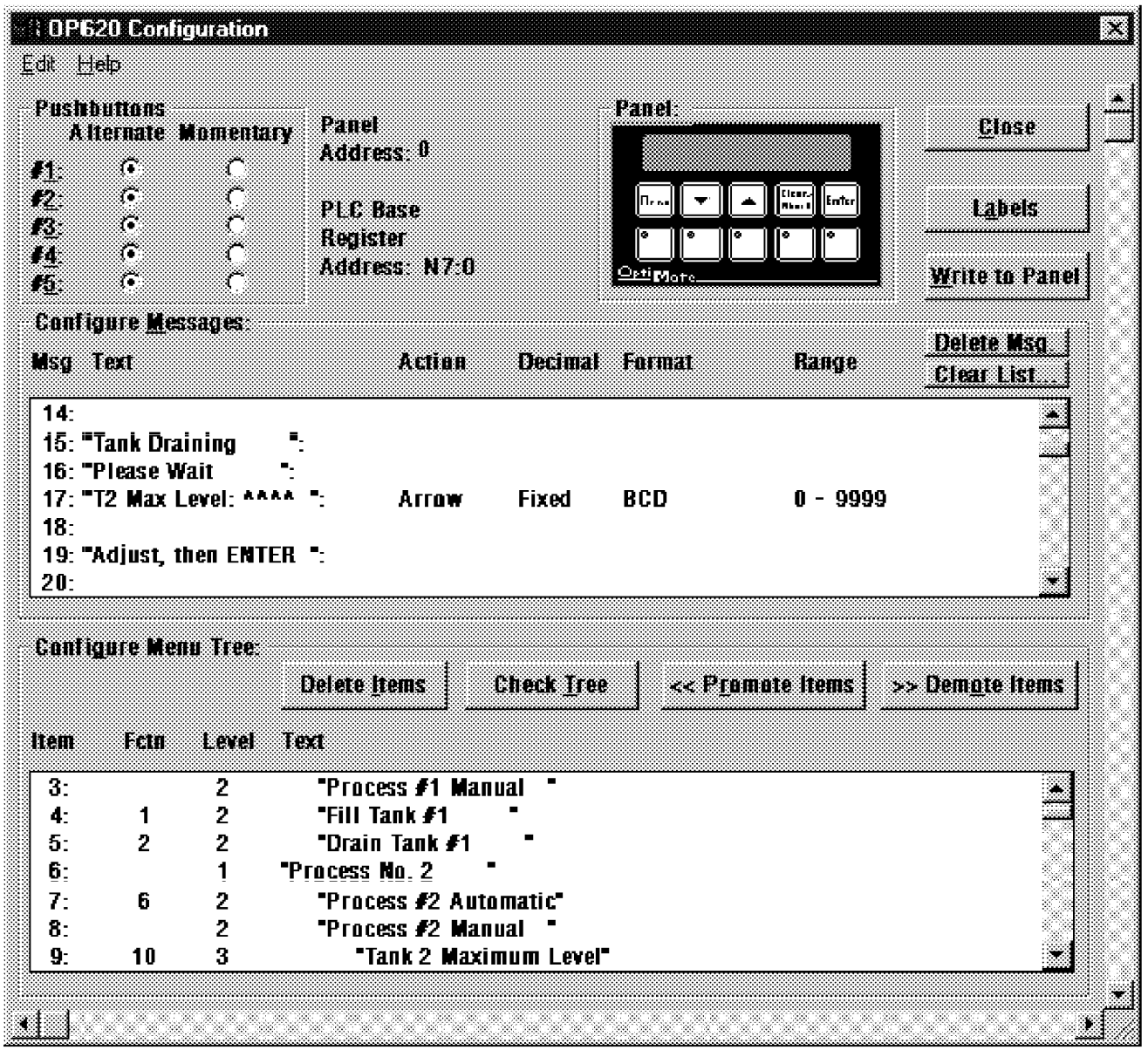
The OP-WINEDIT Configuration Editor screen shown below displays the menu structure, including function associations, in the lower section of the display. For example, if the operator selects "Process #1 Automatic" from the menu tree, function number 5 would be placed in the function select register.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

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 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.



## A Menu Tree Example

The following pages show examples using a Menu Tree. They show how to determine when a function has been selected and how to decode a function number. The examples also show how to implement a function with text display messages and how to implement a function containing arrow adjustment of a setpoint. The base register address for the following examples is N7:0. The examples use the messages and the menu tree shown in the figure below.

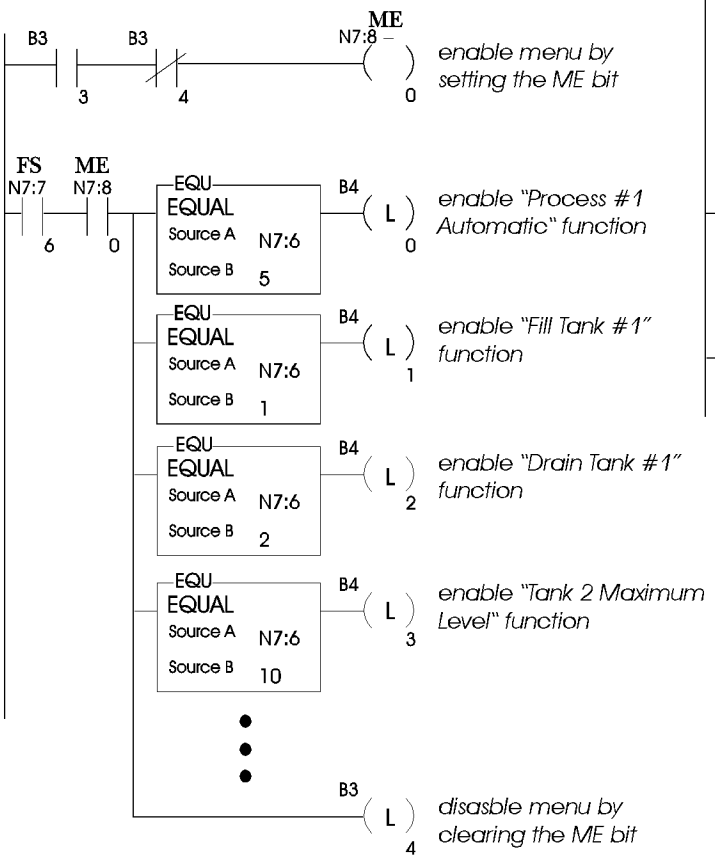


## Decoding a Function Number

The program below illustrates menu tree function selection using the menu tree shown on the previous page. The first rung enables the menu tree when B3/3 is active.

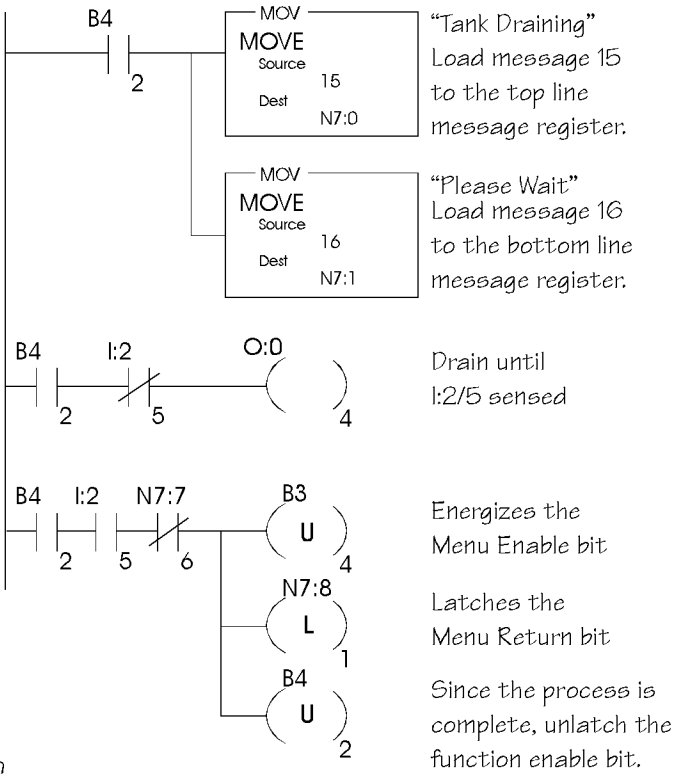
> Note : Enabling the menu tree does not automatically put the terminal into the menu. Once enabled, the terminal will bring up the menu tree when the operator presses "MENU". Until then, the display is under PLC program control.

The second rung sets the appropriate function enable bit when a selection is selected from the OP-620. Function selection will activate the FS flag (N7:7/6). The value comparison to the value held in the function register will latch the appropriate enable relay (B4/0, B4/1, etc.). The last rung shown will also disable the Menu Enable (ME) bit by latching B3/4. Note that the logic shown interlocks the function selects (setting B4/0, B4/1, etc.) with the function select and menu enable flags to ensure that only one function is enabled each time a function is selected.



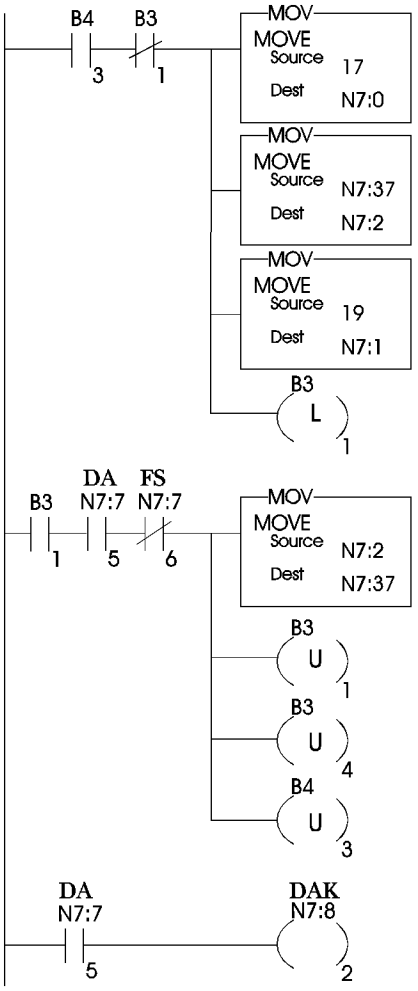

## Implementing a Menu Function

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 just shown, this would result in control relay B4/2 being latched. The first rung of the "Drain Tank #1" function, shown below, will put messages into the top and bottom lines of the display. The second rung energizes output O:0/4 to open the drain until level sensor input I:2/5 senses that it is empty. The third rung re-enables the menu and sets it to return back to the "Drain Tank #1" selection when the tank is empty. The third rung also disables the function. By putting the function select relay (N7:7/6) in series, we force the program to wait until the function select has been cleared before re-enabling the menu.



# Implementng Data Entry using a Menu Function

The logic shown below is another typical example of how a setpoint function might be implemented. Suppose the selection for "Tank 2 Maximum Level" was made. From the logic on the previous page, this will result in B4:3 being latched. Accordingly, the first rung shown below selects the appropriate setpoint message for the top line and a prompt message for the bottom line. It also initializes the setpoint value for arrow adjustment. The first rung 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, unlatch B3:1 to disable the function and unlatch B3:4 to re-enable the menu. If the menu return flag is also latched, the OP-620 will return to the same point in the menu.




"T2 Max Level : ^^^^"
Load message 15 to the top line message register.

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

"Adjust then ENTER"
Load message 16 to the bottom line message 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.

Disable current rung

Energize the ME bit to Re-Enable the Menu

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

Set Data Acknowledge (DAK)

## Example Program using an OP-620

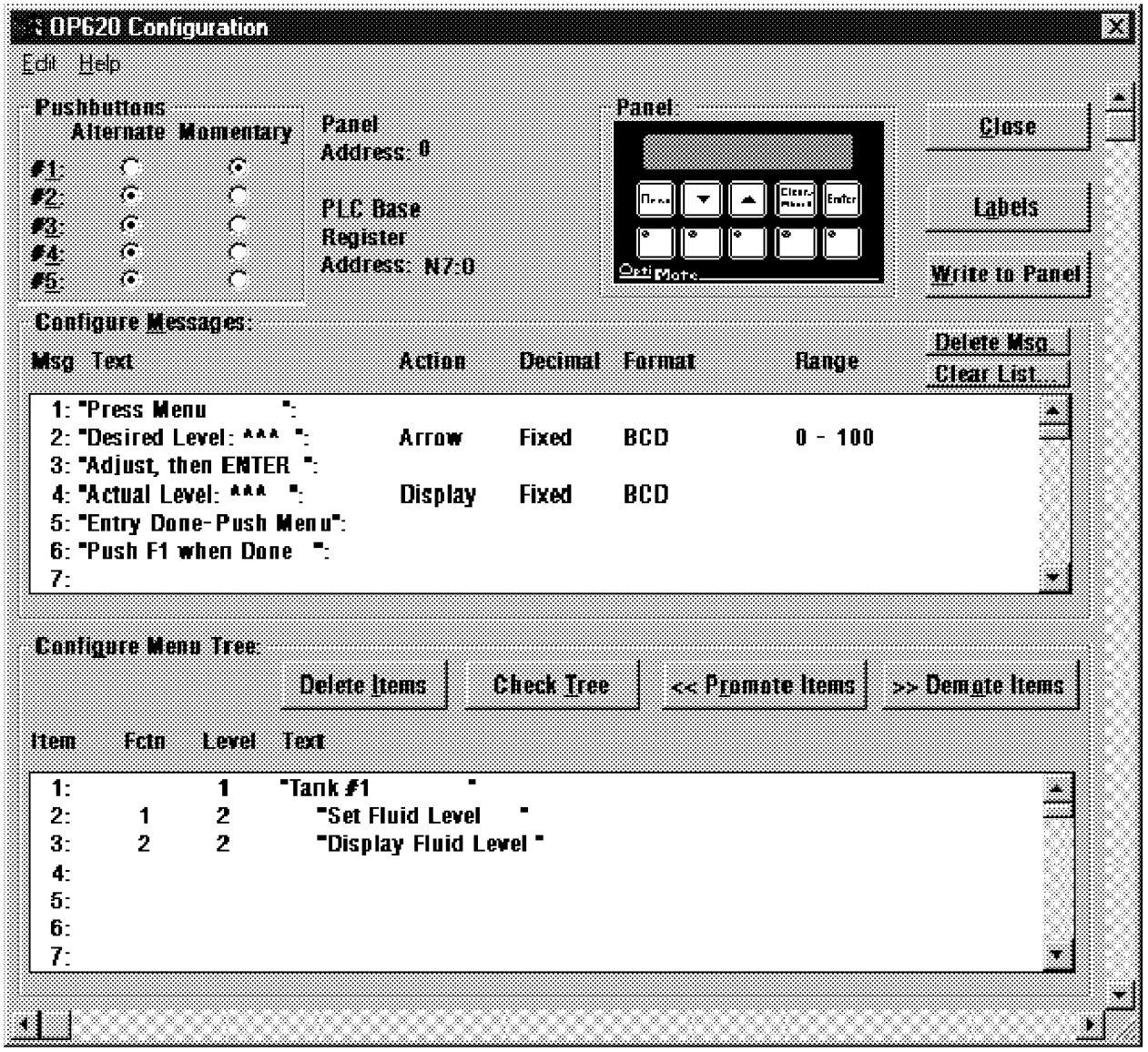
The example program shown in the following 4 pages describes how to implement a program using an OP-620. The program uses the menu tree, an arrow adjust message, a data display message and text messages.

It is a complete program which will run on an Allen-Bradley SLC 5/03, SLC 5/04 or Micrologix PLC. The program uses a two level deep menu tree with two menu functions. The first menu level contains only a single sub-menu message. The second menu level contains two messages, both of which are functions.

The OP-620 configuration for this example is shown in the figure below. The base register address is N7:0. The first pushbutton (F1) is configured for momentary action and the rest for alternate. The first pushbutton is the only one used for this example.

The program implements the following tasks:

- Initializing the display
- Mapping the status and control registers to/from control relay registers
- Setting and Resetting the Menu Enable bit (ME)
- Decoding a function number
- Implementing a function containing an arrow adjust message
- Implementing a function containing a data display message
- Displaying text messages
- Using a function key (pushbutton)
- Using the CLEAR/ABORT button to escape from a menu or function
- Resetting the Data Available (DA) bit using the Data Acknowledge (DAK) bit



### *Example Program*

*Rung 1*

This rung only happens on the first program scan. It places the initial messages onto the OP-620's LCD display. It loads message 1 "Press Menu" to the top line of the display. Message 7, which is blank, is loaded to blank the bottom line of the display. The bit B3/3 is latched to disable the current rung and to energize the menu enable (ME) bit via rung 3.

*Rung 2*

This rung sets the menu enable (ME) bit as long as B3/3 is latched and B3/4 is not. The ME bit allows the user to enter the menu tree when the MENU button is pressed.

*Rungs 3 and 4*

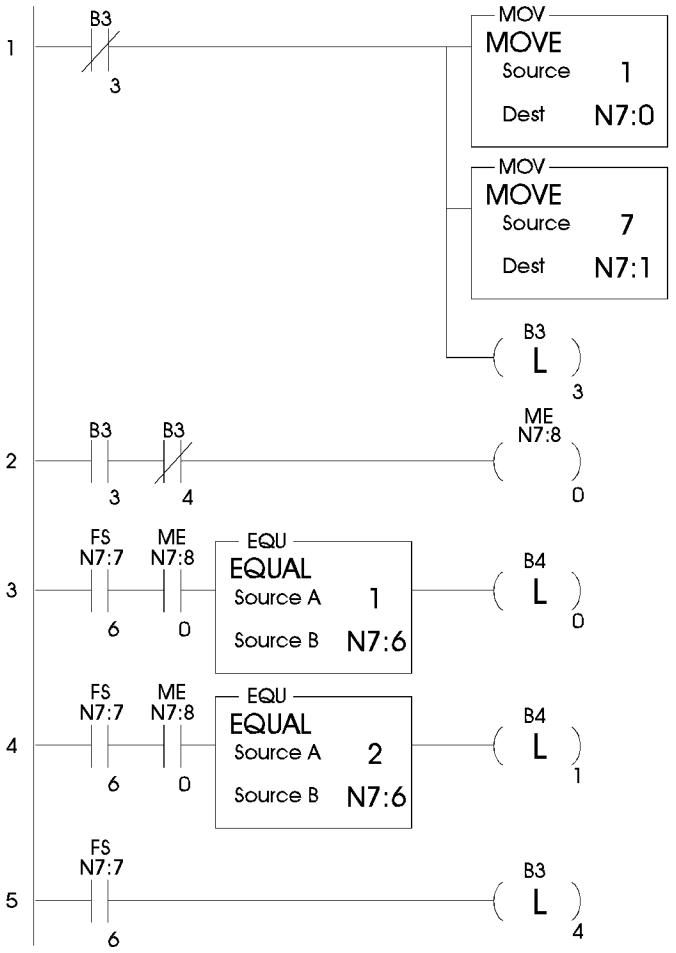
Rungs 3 and 4 are used to determine which function was selected by the OP-620's user. Notice that both the function select (FS) and the menu enable (ME) bits are placed in the rung to ensure that a function has been selected before it determines which function the user selected. After it determines that the FS and ME bits are set, the function number is decoded to determine which function the user selected. To decode the function number, selection register (N7:6) is compared with a constant number. The constant number (1, 2, etc. ) represents the desired function. Once the function number has been determined, a coil is latched. This coil is used only for this particular function.

Rung 3 represents function 1 and rung 4 represents function 2. These rungs should be duplicated for all the function numbers within the program. Later in the program, a rung needs to be added for each bit that starts a function process (B4/0, B4/1, etc.). When that particular bit has been latched, the program knows which function was entered and then the program can implement that function using whatever logic is necessary. Two examples are described within this program.

*Rung 5*

This rung disables the menu enable (ME) bit via B3/4. It checks for the function select (FS) bit. When the FS bit is set, the program knows that a function has been selected. If the function number has been decoded correctly as shown in the previous two rungs, then the panel should exit the menu so the program can implement the function.

The only way to leave the menu is to disable the ME bit. If the ME bit is not disabled after a function is selected, the OP-620's display will "lock up." When the ME bit is reset, the panel will leave the menu mode and the display will return to PLC program control.




The program is continued on the next page.

### *Example Program*
*(Continued from previous page)*

#### *Rung 6*

This rung is the first rung in the logic for function 1, defined in the menu tree as "Set Fluid Level". When B4/0 is enabled and B4/3 is not, the function begins. The second predefined message, "Desired Level: ^^^" is loaded into the top line (N7:0) of the display. Since message 2 is an arrow adjust message, the initial starting point for the adjustment has to be loaded into the top line data register, N7:2. In this case, the data which is stored in N7:20, is copied from N7:20 to N7:2.

Next, message 3 ,"Adjust, then ENTER", is loaded into the bottom line (N7:1) of the display. Then, B4/3 is latched which enables the next rung. A rung of this type which loads arrow adjustment messages should only execute one time within the function, therefore, B4/0 is unlatched to disable the current rung.

#### *Rung 7*

When the user is finished entering the data, the user will press the ENTER key as prompted by the bottom line message. The OP-620 will set the data available (DA) bit, N7:7/5. When the DA bit is set, rung 7 will copy the newly entered data from the top line data register, N7:2, to the data's storage register, N7:20. Next, the program latches B4/4 to enable the next rung and unlatches B4/3 to disable the current rung.

Notice that the function select (FS) bit, N7:7/6, is in series with B4/3 and the DA bit. This ensures that the panel has cleared the function select flag before allowing the program to accept the new data.

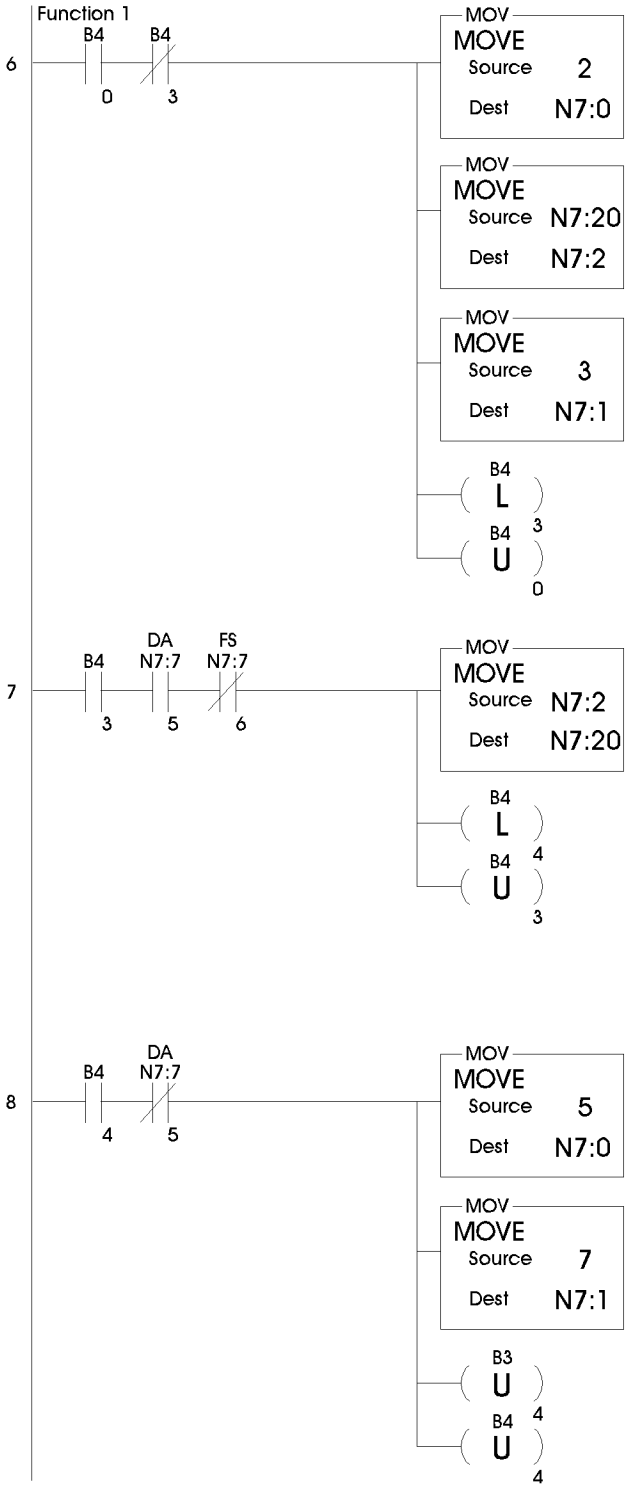
#### *Rung 8*

Rung 8 completes function 1. It waits until the OP-620 clears the data available (DA) bit, N7:7/5, before prompting the user for the next step.

When the panel clears the DA bit, two new messages are loaded to the display. Message 5, "Entry Done-Push Menu" is loaded to the top line of the display. Message 7, a blank message, is loaded to the bottom line of the display. Next, B3/4 is unlatched to re-enable the menu enable (ME) bit via rung 2. Finally, B4/4 is unlatched to disable the current rung.

After this rung is completed, the OP-620 will display the "Entry Done-Push Menu" message until the menu button is pressed.

The DA bit can be cleared two different ways. One is to replace the data entry message with another message, the other is to energize the data acknowledge (DAK) bit, N7:8/2. In this example, the latter was chosen. Rung 13 explains that process in more detail.




The program is continued on the next page.

***Example Program***
*(Continued from previous page)*

*Rung 9*

This rung is the first rung in the logic for function 2, defined in the menu tree as "Display Fluid Level". When B4/1 is enabled and B4/5 is not, the function begins. The fourth predefined message, "Actual Level: ^^^", is loaded into the top line (N7:0) of the display. Since message 2 is a data display message, the initial display value has to be loaded into the top line data register, N7:2. In this case, the initial value, which is stored in N7:21, is copied from N7:21 to N7:2.

Next, message 6 ,"Push F1 when Done", is loaded into the bottom line (N7:1) of the display. Then, B4/5 is latched to enable the next rung. Last, B4/1 is unlatched to disable the current rung.

*Rung 10*

Rung 10 updates the data field of the display message on the top line. The OP-620's user can view the data as it changes within N7:21. As long as this rung is active, the top line data register, N7:2, will be updated with the current data, stored in N7:21.

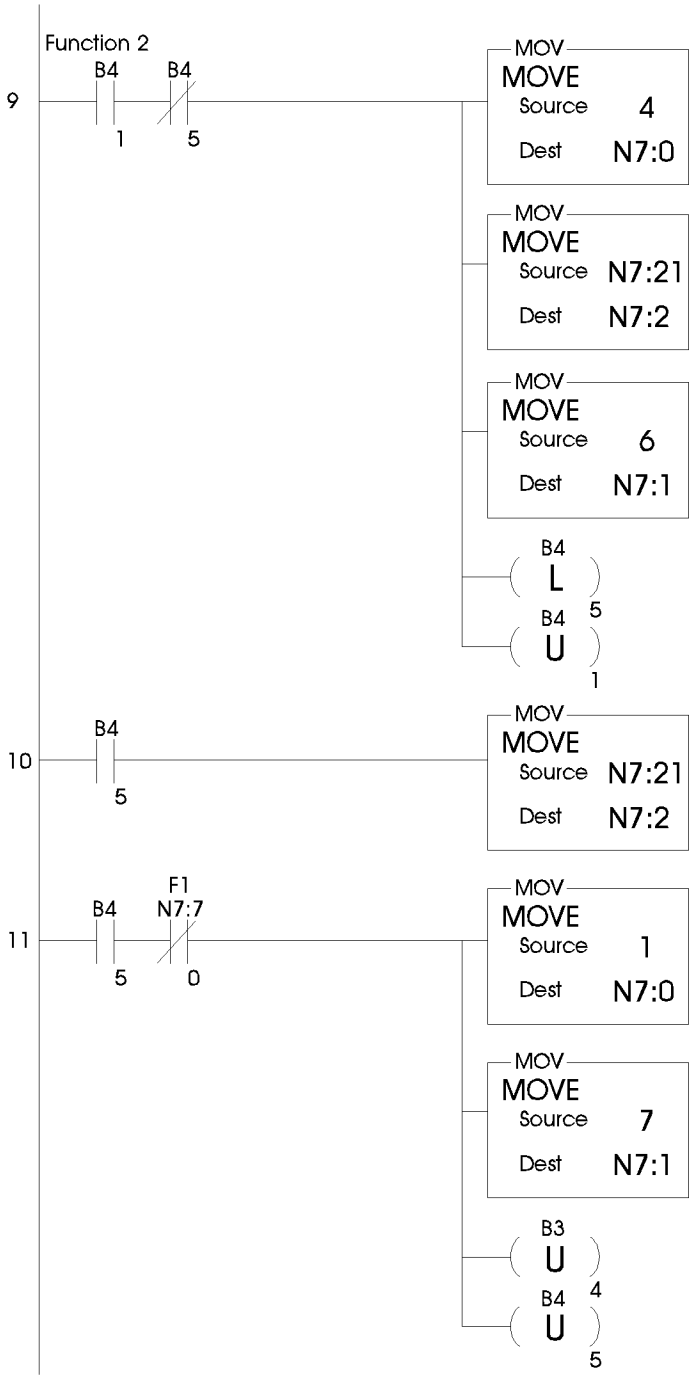
*Rung 11*

Message 6 on the bottom line prompts the user to "Push F1 when Done". When the OP-620's user wishes to stop viewing the data, they should press the first pushbutton, F1. This will complete function 2.

When F1, a momentary action button, is pressed, a new set of messages are loaded into the display registers . Message 1, "Press Menu", is loaded into the top line and message 7, a blank message, is loaded into the bottom line of the display. Next, B3/4 is unlatched to re-enable the menu enable (ME) bit via rung 2. Finally, B4/5 is unlatched to disable the current and previous rungs.

After this rung is completed, the OP-620 will display the "Press Menu" message until the menu button is pressed.

The program is continued on the next page.



***Example Program***
*(Continued from previous page)*

*Rung 12*

This rung is an abort rung. If the OP-620's user is in the menu or in the middle of a function and they realize that they need to exit without making any changes, then they can use the CLEAR/ABORT button for that purpose. If the panel is in the menu tree when the CLEAR/ABORT button is pressed, the panel will exit the menu and default back to the messages that are in the top and bottom line message registers. If the user is in a function, this rung can be used to unlatch different bits within the program that pertain to any given function which could be active at the time, B4/0, B4/1, B4/3, etc.

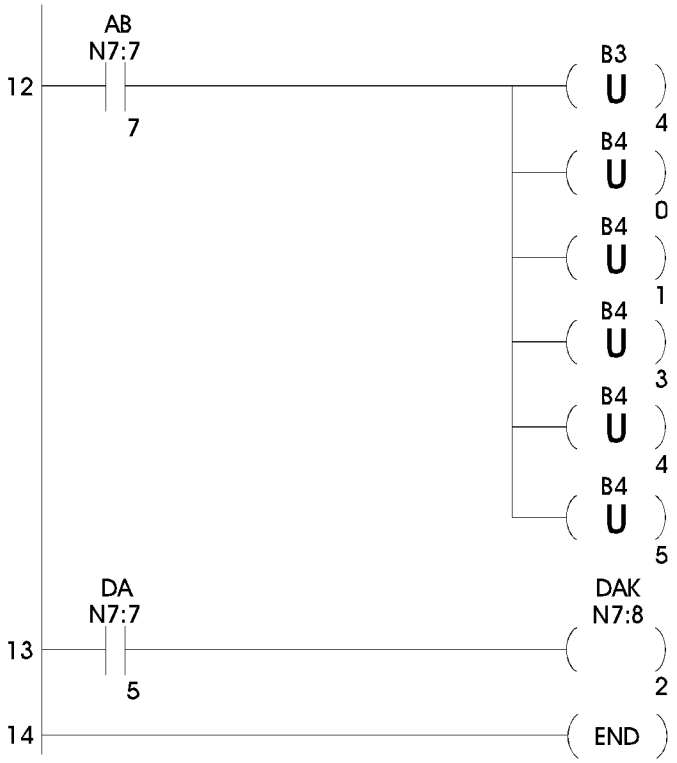
This rung can also be OR'd with other conditions, such as machine jams, emergency conditions, etc. that might warrant exiting the menu (by clearing the menu enable bit) or disabling a function. Messages can also be loaded to the display to describe the problem to the user if so desired.

*Rung 13*

This rung energizes the data acknowledge (DAK) bit whenever the data available (DA) bit is latched by the OP-620. It will stay active until the panel sees that the DAK has been set. At that time, the panel will unlatch the DA bit which will deactivate the rung. If you choose to unlatch the DA bit in this manner, this rung should be placed at the end of the program where it can be checked every scan.

*Rung 14*

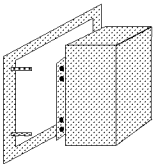
This rung marks the end of the program.



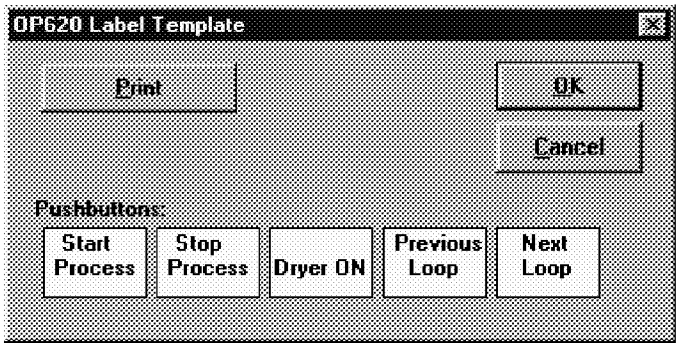
# Set Up and Interconnect

## Legending the Function Keys

Legending the OP-620 panel is a relatively simple process that basically involves sliding a label transparency into a pocket in the panel overlay. Use the following procedure.

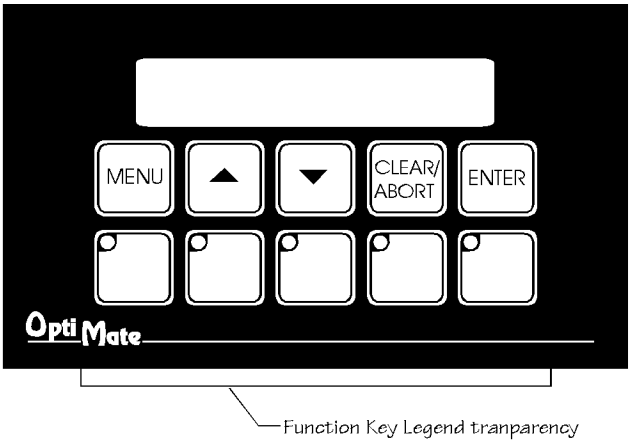


- Remove the bezel from the panel. The bezel snaps to the panel box along the top and bottom edges. Pull the bezel out and over the snaps to remove.
- Create legend transparencies. There are a number of available options for doing so. Patterns are provided on the next to last sheet of this document.
  - > Use the built in label making capability of the OP-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 OP-WINEDIT which illustrates the process.




Other options include the following

- > Uses 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.



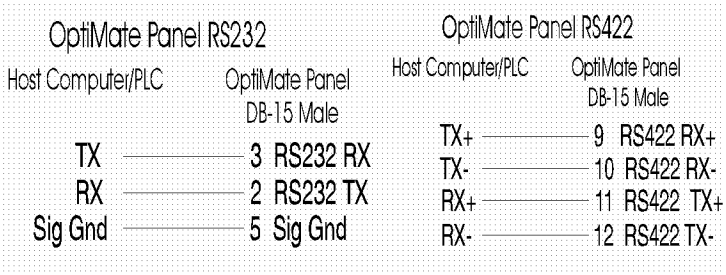
# Connection to the System

OptiMate panels are designed for communication connection to system devices. The panel can be connected to a computer , PLC or communication master 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 RS 232 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.



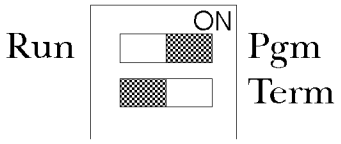

Interface cables for connection to several different PLCs as well as to IBM PCAT compatible ports are available.

### Serial Connection to OP-9001 Communications Master

Connection to an OP-9001 Communications Master over a serial link is via RS422. The Communication Master port connections are reversed from the panel ports to enable direct pin to pin connection. For distances under 50 feet (in a low electrical noise environment), a ribbon cable connection works quite well. For longer distances or in noisy environments, a two pair shielded RS422 cable is recommended. We recommend Belden 9729 or equivalent cable.

### Termination

The termination DIP switch on the back of the panel switches in a terminating resistor. This terminating resistor does not apply to an RS232 connection (and should be in the OFF position for RS232). In an RS422 connected system, such as with the OP-9001 Communication Master, the termination switch should be on in the last, and only the last, panel on the cable.




# Power

OptiMate modules can operate on any voltage between 8 and 30 VDC. Power must be connected to the terminal plug located on the back of the module.

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 not normally a problem for a commercial power supply.

For multi-panel systems using multiple power supplies, we recommend tying the DC commons (grounds) together for each separate power supply. This will ensure that the voltage differential of each power supply DC ground will be at the same level.

# Configuration

## Configuration Selections

OptiMate panels can be configured for the specific application by using the OP-WINEDIT Configuration Editor. The OP-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 OP-620 module, the following are important configuration parameters.

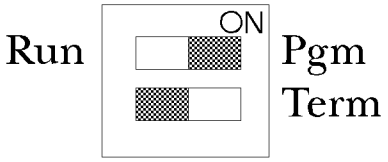
### Single Module PLC Based Systems

| 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 Communications Master)

| Decision | Selection |
|---|---|
| Single/Multi Module | Chose Multi module |
| PLC Type | This applies to the Communications master. Choose appropriate type |
| Protocol | This applies to the OP-9001 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 OP-WINEDIT software will automatically select the OptiMate Hex protocol for communications between the OP-9001 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 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 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. The "Term" switch must be in the OFF position unless the panel is at the end of a cable in an RS422 system.*



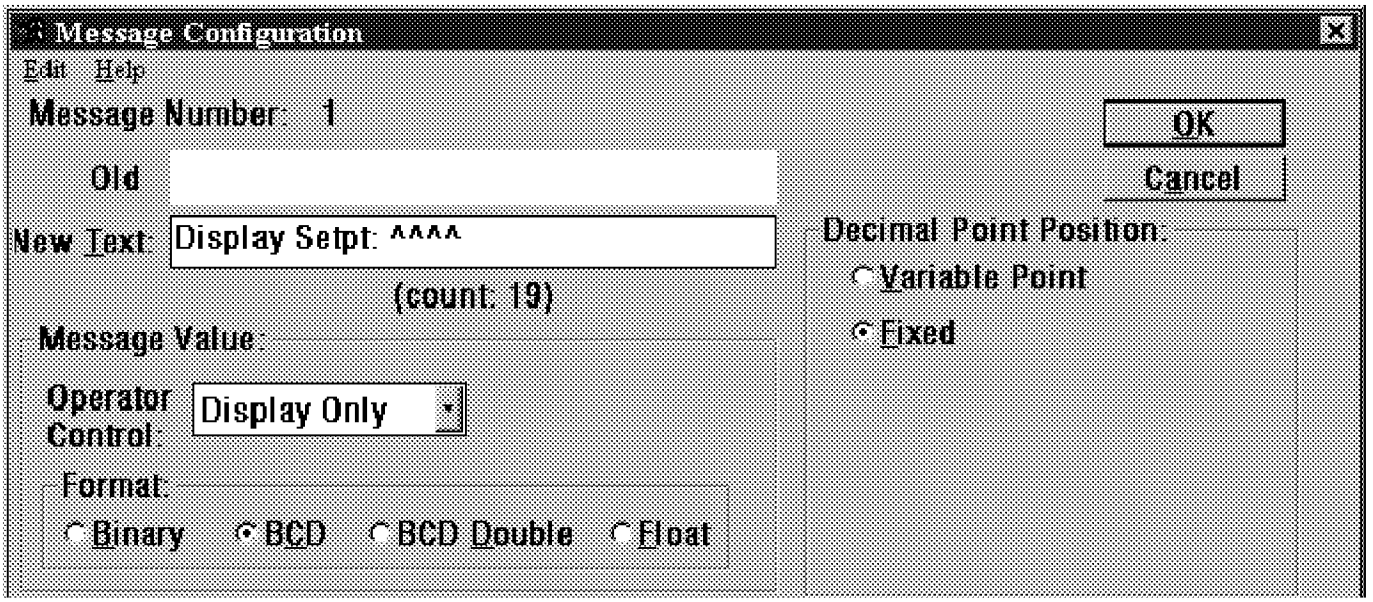
## Creating Messages

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

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

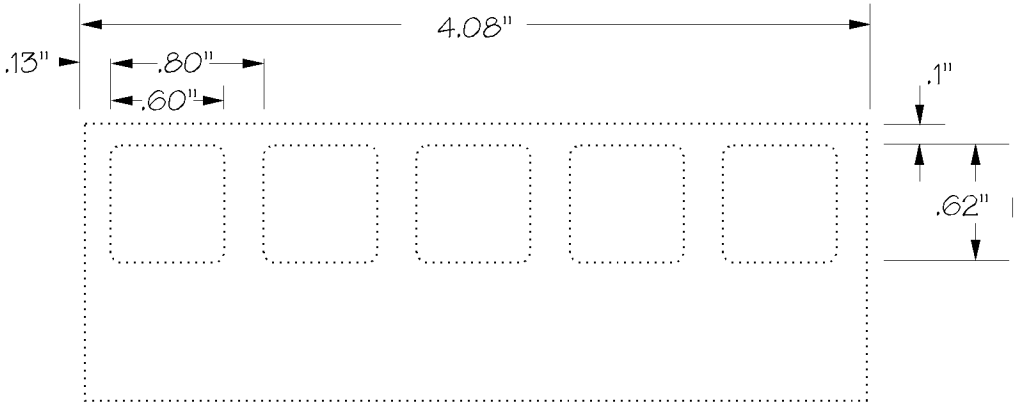
Example Message Definitions



# Label and Message Definition Templates

| Message # | Text (20Characters Max.) |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

Message Definition Template

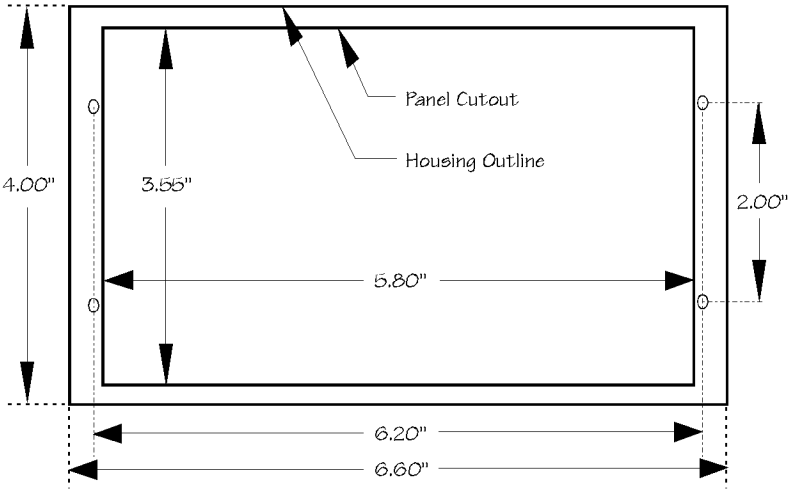



Pushbutton Label Strip Pattern

# Specifications

**Physical**

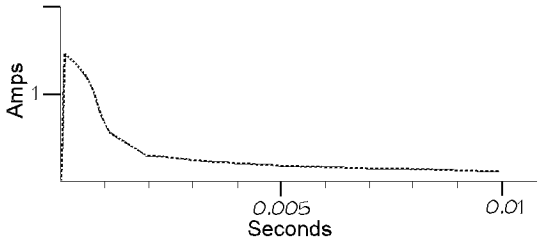
- Recessed Mount Housing 6.6"Lx4.0"H x 1.25"D
- Cutout size for above 3.55"Hx5.80"L
- Panel Fasteners : Four, 6x32 threaded studs, shown above (on ends, symmetrical about center line)
- Weight : 10 ounces
- Colors : Dark gray housing with dark gray panel. Keypad keys; white with dark gray letters. White 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 major PLC protocols
- 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)