



Application Note #5482

Interfacing to a Galil controller with a Unitronics Jazz PLC

It is the goal of all motion control applications to be as autonomous as possible. However, it is sometimes necessary to require an input from a user to coordinate motion with other events. For instance, an input may be necessary to indicate to the machine that a particular part has been placed into position by the user. For this type of input a standard contact switch would suffice and can be wired directly into the motion controller. If the input from the user was expanded to not only tell the machine that the part has been placed into position, but also to provide a numerical input to indicate the dimensions of the part, a Human Machine Interface (HMI) may be necessary. Many HMIs provide a screen which can be used to display prompts to the user, and preprogrammed buttons to send certain commands to the motion controller. If continuous communication is required to the motion controller, or large amounts of data must be shared between the motion controller and the user, a custom software front-end program may be built to provide an adequate interface. Although this is the most flexible and capable option, it also requires the addition of a PC into the system, and development of software front ends can quickly become complex and expensive.

For applications that require more interface flexibility than what can be offered by an HMI, and do not need a custom front end software package, the Programmable Logic Control (PLC) can provide an interface solution. PLC's can offer higher processing capabilities, more IO points, and other communication protocols at a cost affective price.

PLC's range in price, capability and complexity. One low cost PLC that can fit into and provide an adequate interface for many applications is the Unitronics Jazz PLC. This PLC offers an array of additional IO to coordinate motion with external events and devices. This coordination and processing is done completely divorced from the controller. The Jazz PLC also offers a 2 line programmable digital display and keypad for easy input. A complete description of the Jazz PLC can be found on the Unitronics web site at: <http://www.unitronics.com/Series.aspx?page=Jazz>

By taking advantage of Galil's standard 8-bit ASCII communication protocol, connection between the Jazz PLC and any Galil controller is as simple as assembling contacts within a ladder program. An example of how to establish communication between a Galil controller and a Jazz PLC is provided below. Also a brief program which issues a relative move (Galil PR command) to a controller, through the PLC based on an operator input is provided.

Step 1: Connecting Hardware

The Jazz PLC offers serial communication through various add-on modules. The proper hardware components to communicate to a Galil controller and configuration are as follows:

- a. PLC: Unitronics Jazz PLC
- b. Add on Module: MJ20-RS
- c. RJ11 Communication Cord: MJ20-CB200
- d. "PC Adaptor" (actually RJ11 to Serial converter): MJ10-22-CS25

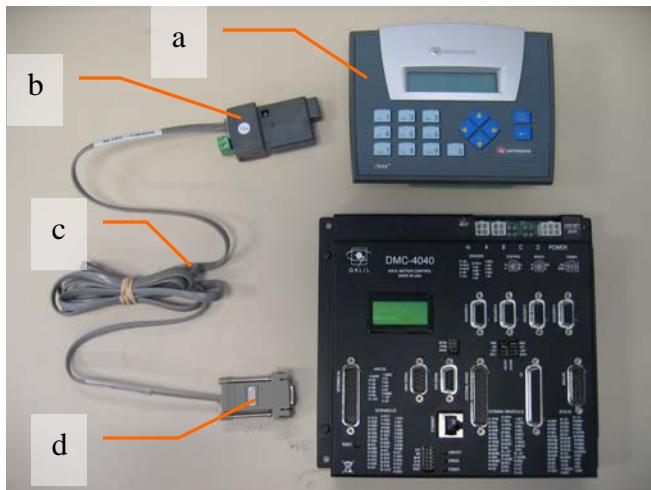


Figure 1.a: Components needed for communication, labeled as indicated in above paragraph.



Figure 1.b: The Unitronics Add on Mocule plugged into the Jazz PLC.

Because the Unitronics PC adaptor has different pinouts than the Galil serial port, a custom RS-232 plug must be created to connect the two devices. A wiring schematic of how to create this custom plug can be seen below in Figure 2. If a custom plug is not feasible please contact Galil motion control for customization options that will allow a direct connection from the Unitronics PC adaptor to the Galil main serial port.

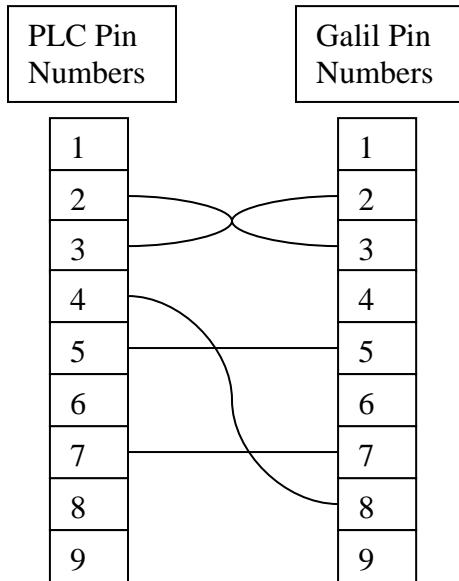


Figure 2: PLC to Galil pinouts and wiring schematic.

Any additional IO can be connected to the Jazz at this point or any point hereafter if needed by the system.

Step 2: Download Software

Software will be needed to program and communicate to the Jazz PLC. This can be done by downloading the free Unitronics U90 Ladder interface software off of their web site at: <http://www.unitronics.com/Content.aspx?Page=Downloads&CatId=3>. Software will also be needed to configure the Galil. The particular software package needed depends on the controller model, however all of Galil's software can be downloaded off of the web site at: <http://www.galilmc.com/support/download.html>

Step 3: Configure the PLC.

Communication from the Jazz PLC to a Galil controller is done via RS-232 by sending and receiving ASCII strings. The configuration parameters to communicate to the Galil controller are entered into the PLC by assigning integer values to different ‘System Integers.’ A description of the input parameters and configuration of these system integers is available under the U90 Help menu “Change COM Port Parameters.” To communicate to a Galil controller the following parameters must be set as listed in Table 1, and programmed in the U90 Ladder interface as in Figure 3:

Parameter	Value	System Integer
Baud Rate*	9600	141
Data Bits	8	142
Parity	None	143
Hardware Flow Control	1 (Flow Control)	144
Timeout	50 (ms)	145
Stop Bits	1	146

Table 1: Galil RS-232 communication protocol and appropriate Jazz system integers.

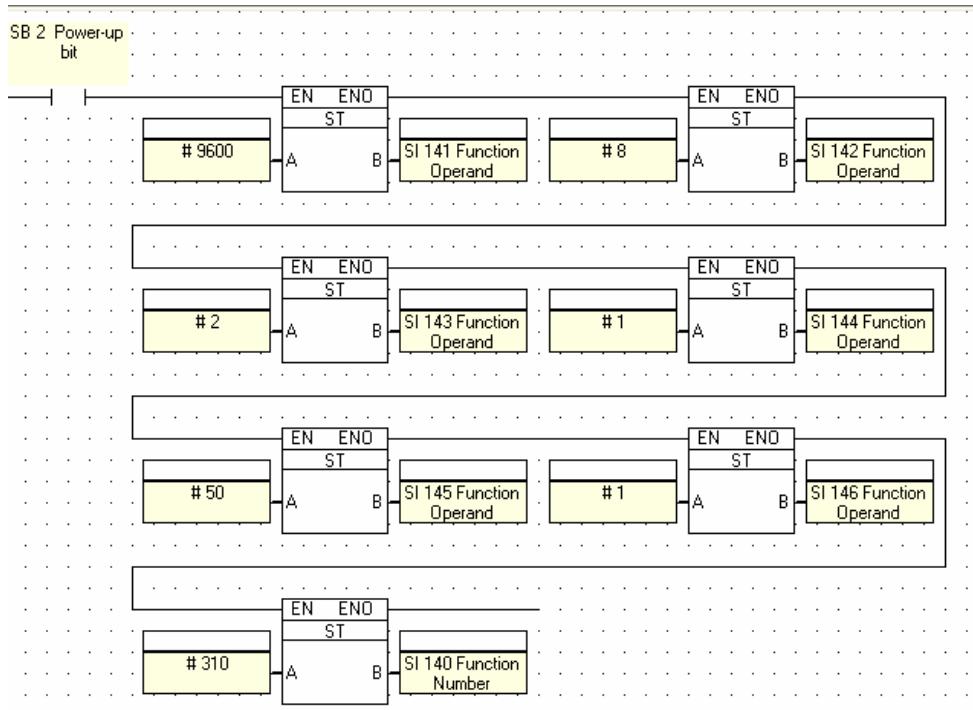


Figure 3: Screen shot of Ladder configuration to set Galil RS-232 communication protocol. Note that the rung starts with System Bit (SB) 2, which activates only on power up, and ends with System Integer (SI 140) being set to 310, which stores the com parameters and indicates that communication has been set.

Step 4: Develop Program and Implement

Now that communication has been established, you can develop a program to interface with the Galil controller. An example program which will serve to establish communication and initiate a simple move to a Galil controller is available for download off of the Galil web site at: <http://www.galilmc.com/ftp/pub/appnotes/note5482/>. This program can be opened with the U90 software and downloaded to the PLC. Additionally a description of the program development for reference can be found in the appendix of this note.

Systems that involve motion control, often require operator interfaces. Depending on how the operator interfaces with the system, will dictate the type of interface that should be implemented. These operator interfaces range from standard buttons and switches, to PC front end software programs. PLC's can accommodate an intermediate level of operator interface, and offer external IO processing, additional IO, different communication protocols, keypads, screens and more. One cost effective PLC solution is the Unitronics Jazz PLC, which easily interfaces with Galil motion controllers. Establishing this connection is made possible by Galil's standard 8 bit ASCII communication protocol, RS-232 port and easy programming language. Using a Galil motion controller with a Unitronics Jazz PLC can accommodate most motion control applications needing an operator interface. For more information or assistance with setting up communication to a Galil motion controller to a Unitronics Jazz PLC, please contact Galil's applications department with the information provided on the following link: <http://www.galilmc.com/contact/index.html>

Appendix 1

This appendix will review the development of the example program provided by Galil. This program will establish communication between the Unitronics PLC and the Galil motion controller and issue a relative move of a user defined distance.

Step 1:

Open a new ladder project in the U90 Software. You will be prompted to indicate the specific PLC and IO module(s) you are using. Do so and click okay. A blank screen should appear like the one below in Figure 4.

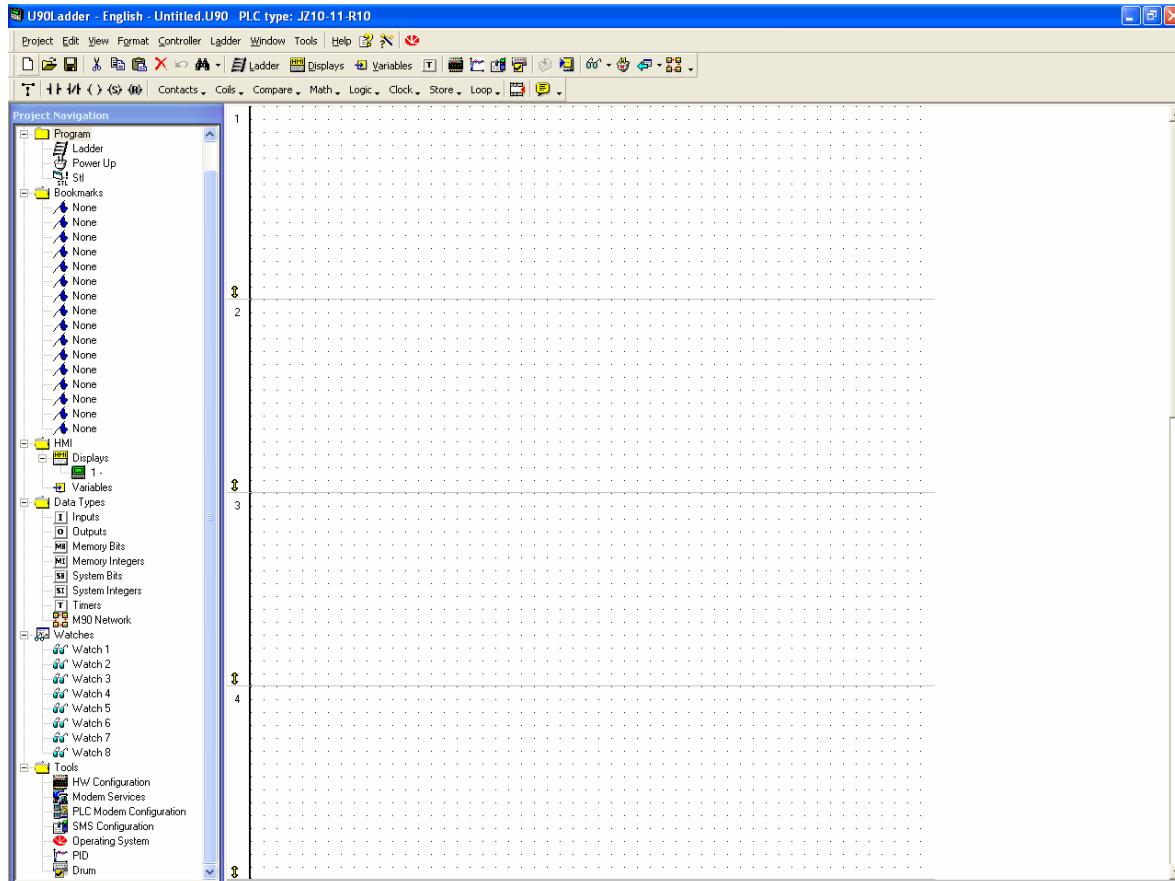


Figure 4: Screen shot of new ladder project screen

Step 2:

Select and insert all of the buttons necessary to establish communication based on the communication parameters of your system and system integers (SI) settings needed by the PLC. See Figure 5 for a complete illustration of the assembled communication buttons.

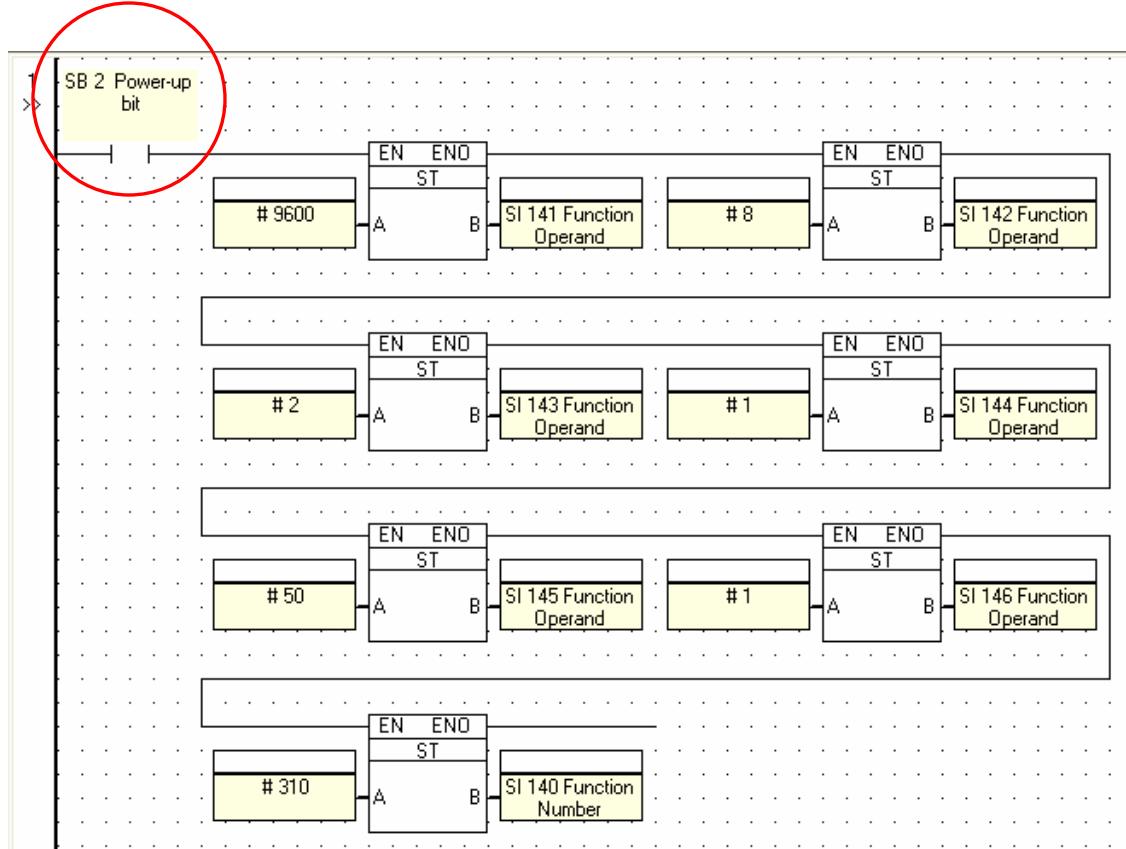


Figure 5: Assembled communication buttons, with Power-up bit SB2 highlighted.

You will notice that the direct contact, highlighted in the above illustration, is triggered by system bit (SB) 2. SB2 is the ‘power-up’ bit of the PLC. This means this series of objects will only execute once, on power up. The communication parameters are set by linking the ‘Store Direct’ objects (found under the store menu) and storing the integers into the proper SI locations. For a complete description of these and other system integers, click on ‘System Integers’ in the Project Navigation tree located in the ‘Data Types’ folder.

3) At this point communication to the controller through the PLC can be established. However there is no program to provide an action after connection. Provided as a simple example program, the following will illustrate how to set up a user prompted input through the 2 line display, entering values through the keypad and ultimately issuing a PR move to the controller.

The first step to generating this program is to set up a system variable. The system variable will store the input from the user as to the distance of the relative move. Under the project navigation tree, if the ‘Variables’ tab is clicked, which is found in the ‘HMI’ folder, a new variable screen will appear in the main window. Select the ‘Integer (Numeric value)’ radio button under ‘Variable Type’ and set up all options similar to the ones shown in the following figure, Figure 6.

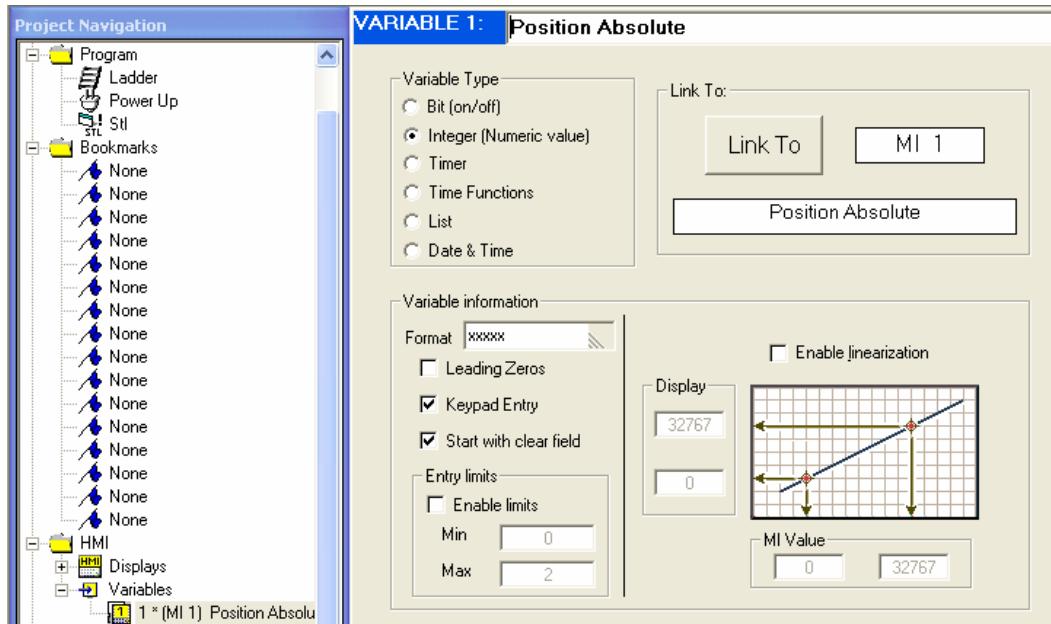


Figure 6: Screen shot of the variable set up screen and necessary parameters.

4) Next, set up the SMS message which will send the ASCII string out of the serial port to the controller. This can be done by clicking ‘SMS Configuration’ in the Project Navigation tree in the ‘Tools’ folder. The following window will appear as in Figure 7.

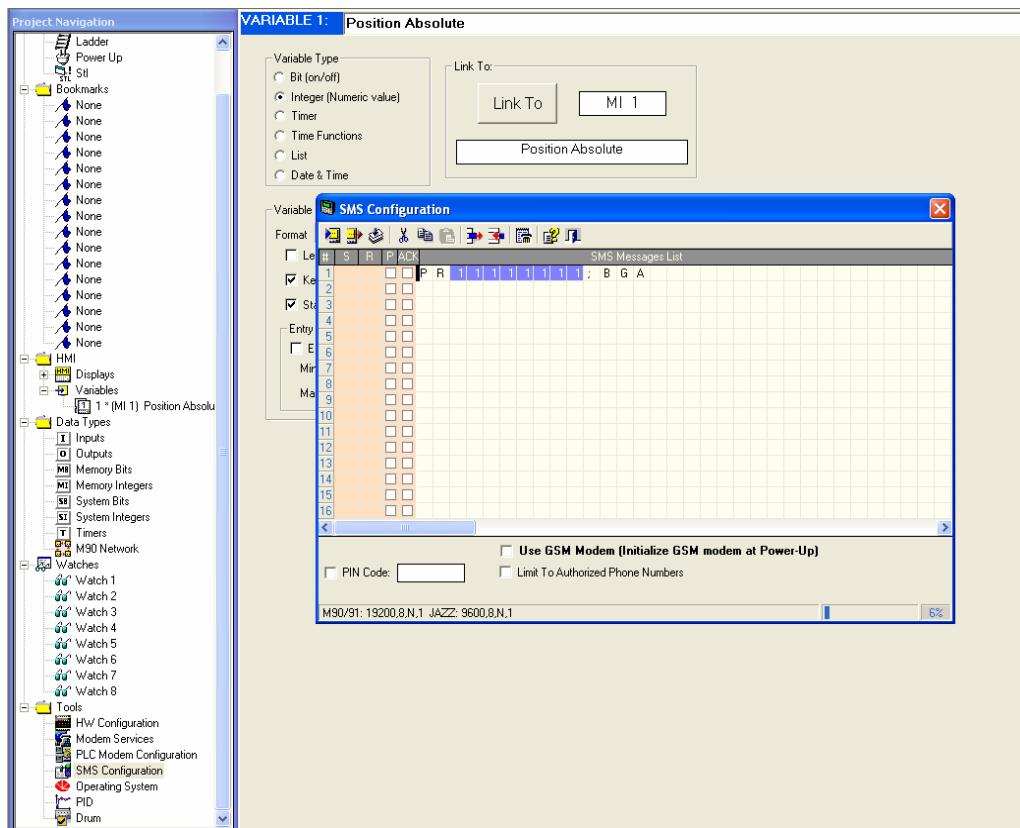


Figure 7: Screen shot of the SMS configuration window, and the built message.

This window will allow you to set up the message that will be sent when prompted by the PLC. The text string seen below will send the ASCII string; “PR”, followed by the variable “1”, then a new line prompt with a “;” then the begin command, “BGA”.

PR 11111111;BGA

The variable 1 was set up in step 3. By holding the shift key, and moving the cursor with the keyboard arrows, the character blocks will be highlighted. This indicates to the PLC that you are entering a display-specified variable. After the shift key is released, a ‘Select Operand and Address’ window will appear. This allows you to indicate which variable you would like to insert into this space when assembling the SMS message. In this example the variable 1 will be indicated. The number of variable characters blocked in the SMS message window by holding the shift key will be the maximum length of the variable that can be entered by the user. This is illustrated in Figure 8.

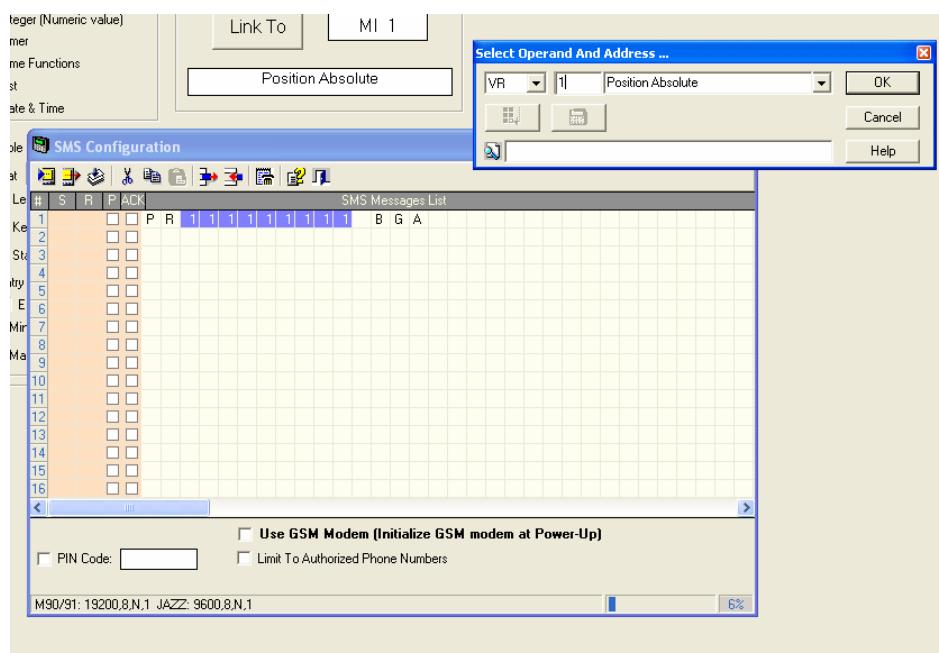


Figure 8: Illustration of the SMS Configuration window and the Select Operand and Address pop-up window that appears after the variable length and location is set.

5) Next, the PLC screen must be set up. Click on ‘Displays’ in the Project Navigation tree in the HMI folder. The configuration screen seen in Figure 9 will appear.

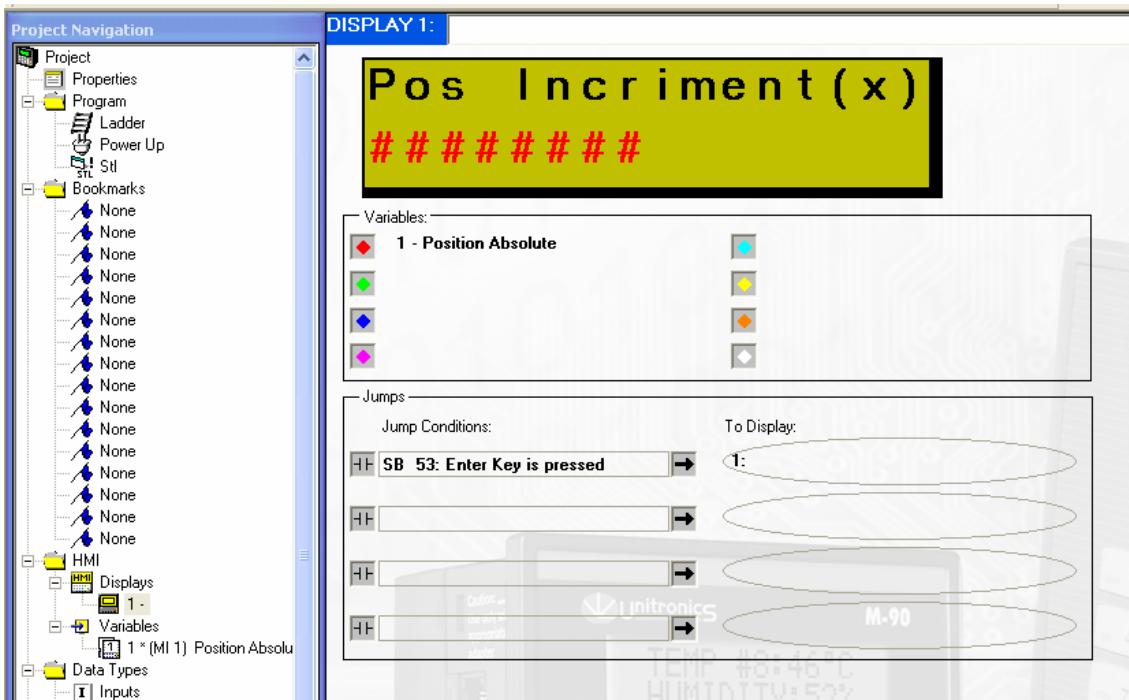


Figure 9: Screen shot of the Unitronics Jazz PLC screen configuration.

The screen available on the JAZZ PLC has 2 rows and 16 columns. For this screen the top row will simply display the text “Pos Increment (x)”. The second row will be the field that the user can enter the relative distance. This is indicated by placing the cursor on the location of the first input character then by holding the shift key and pressing the right arrow. This will assign the variable field. After the shift key is released another ‘Select Operand and Address’ window will appear. Indicate the variable you want to have this input assigned to, and click okay.

- 6) Next return to the main ladder program by clicking on ‘Ladder’ in the Project Navigation tree in the ‘Program’ folder. Insert the objects shown in Figure 10 and assign the proper variables, integers and system integers.

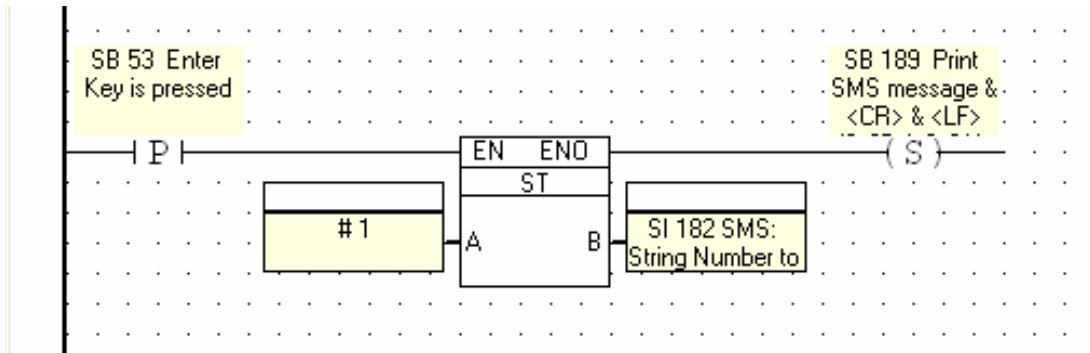


Figure 10: Buttons and objects configured to send the SMS string when the enter key is pressed.

SB 53 is the system bit that activates when the ‘enter’ key on the keypad is activated. In this particular program a ‘Positive Transition Contact’ was used. A Positive Transition

Contact will activate when the button is released. This eliminates the possibility of multiple button presses by the user. By storing the #1 into system integer 182, the PLC will know to assemble the SMS message 1 (first line of the SMS message window). When this function is called the PLC will assemble the message with the variable 1 (the variable set in the display) inserted into the SMS message. Then by setting the coil to system bit (SB) to 189, the PLC will send the assembled message.

7) After this step the program is ready to be downloaded to the PLC. Download the program by connecting the PLC directly to a PC (the custom connection cable is not needed) and pressing the ‘Download’ button in the tool bar. After the download, the PLC can then be connected to the controller through the custom plug and the program executed.