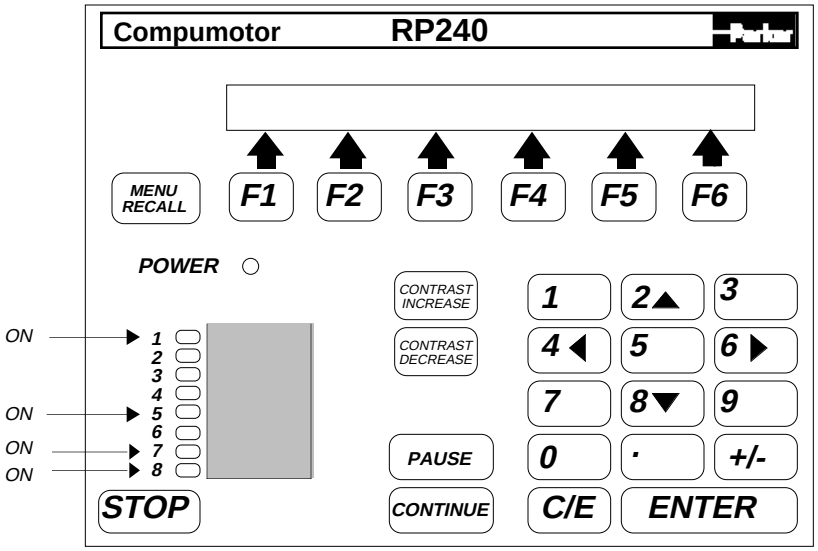
# Discussion of Extended X Language Commands

This section discusses the commands described previously in the command reference.

## Prompting an Operator or Displaying Information

In many motion control applications, the most important requirement is the operator interface. Presenting information to an operator in a desired format is often difficult at best. The RP240 has two visual indicators to help present information to the operator. The simplest indicator is the 8 LEDs on the panel. These LEDs; can be turned on or off with the **DLED** command. The LEDs can be used in conjunction with the outputs to show the state of an output, or they can be used to show status, such as motor moving, specific sequence in progress, etc.

IF a **DLED1ØØØ1Ø11** command is issued, the LEDs shown below would be illuminated. These eight LEDs can be labeled, using the slide-in card provided, to represent cycle status, output status, etc.




The other indicator is the two line, 40-character LCD display. This display can be controlled with specific Extended X Language commands. The Position Cursor (**DPC**) command allows the user to program the location of the cursor on the LCD display. The Display Text Data (**DTXT**) command on RP240 Display allows the user to place text, beginning at the current cursor location, on the LCD display.

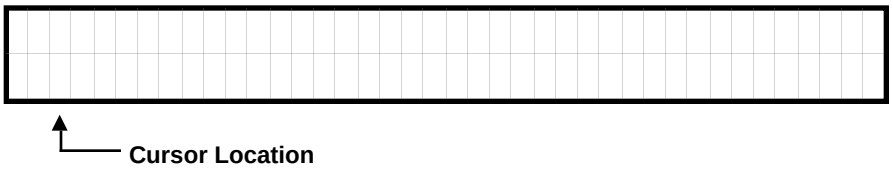
# Extended X Language Command Programming Example

A user wants his operator to see the message **ENTER THE CYCLE COUNT**. He wants this message placed on line two, starting after two spaces in from the left. He also wants the user to be able to enter the cycle count three spaces after the message. Below are the steps required to accomplish this.

## Step 1

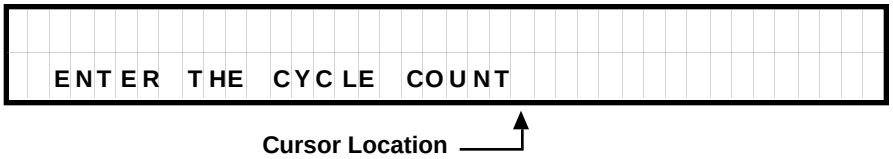
Issue the **DPC2Ø2** command

☞ Helpful Hint:
The cursor does not appear on the display. The cursor is displayed when the VARn=NUM command is used.



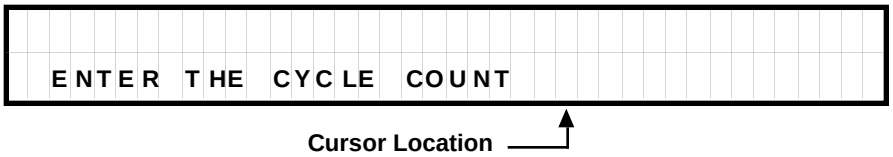

## Step 2

Issue the **DTXTENTER_THE_CYCLE_COUNT** command




## Step 3

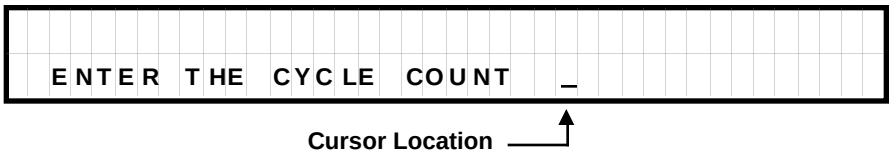
Issue the **DPC225** command




Now that we have provided the operator with the prompting message, how do we obtain the information? The Read and Enable Numeric Keypad (**VARn=NUM**) or Read and Enable Function Keys (**VARn=FUN**) commands provide the answer. The **VARn=NUM** command will enable the numeric keypad and allow the operator to enter information. The numbers, as entered, will be displayed at the current cursor location. Once the ENTER key is pressed, the number will be transmitted from the RP240 to the Model 500, SX, or ZX.

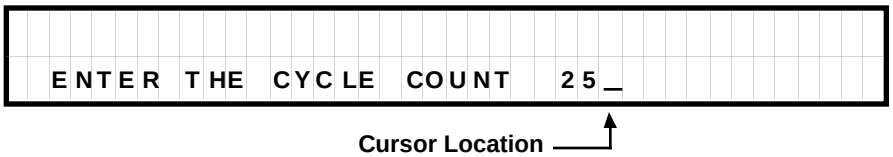
## Step 4

Issue the **VAR1=NUM** command




## Step 5

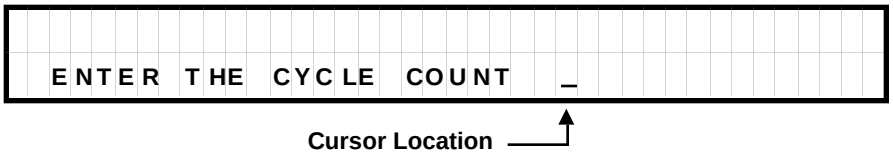
Press a 2, followed by a 5



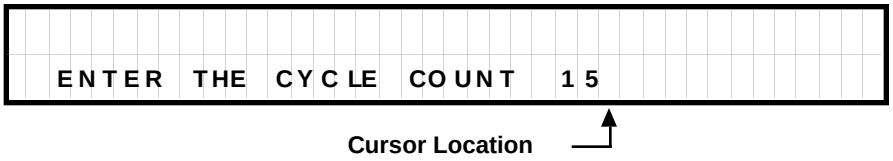

*If the wrong value is entered, press the C/E key and re-enter the value.*

## Step 6

Press the C/E key

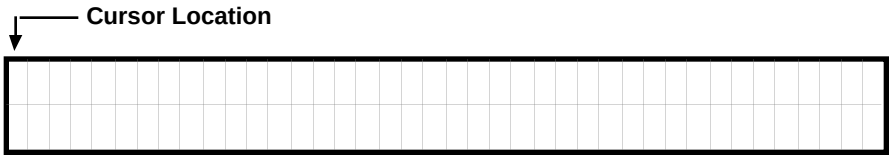


Step 7 Press a 1, followed by a 5, followed by an ENTER

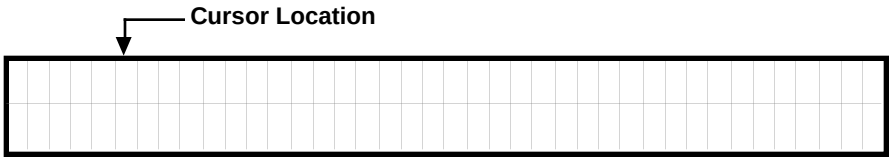



After the ENTER key is pressed, the RP240 will transmit the value 15 to the Model 500, SX, or ZX. This value will be stored in variable 1. Use the **VARn=FUN** command to enter information based on function key input. Steps 8 through 14 illustrate this capability.

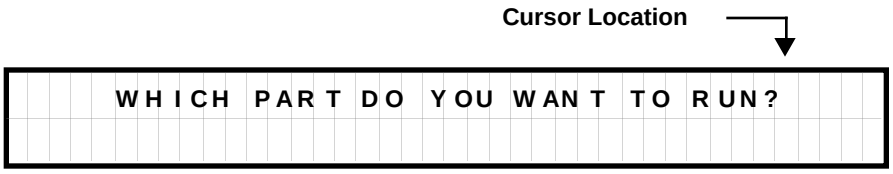
Step 8 **DCLRØ** command is issued.



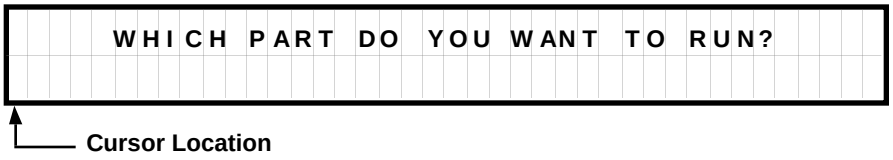

Step 9 Issue the **DPC1Ø5** command



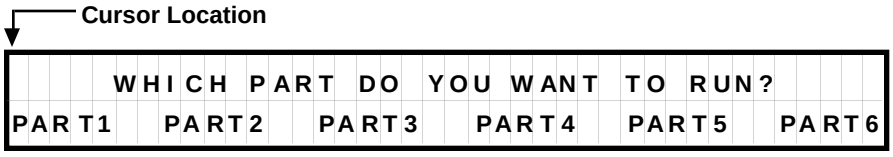

Step 10 Issue the **DTXTWHICH_PART_DO_YOU_WANT_TO_RUN**? command




Step 11 Issue the **DPC2ØØ** command




Step 12 Issue the **DTXTPART1__PART2__PART3__PART4__PART5__PART5__PART6** command. This text serves as an operator menu for the function keys.




Step 13 Issue the **VAR2=FUN** command

The **VARn=FUN** command enables the function (F1 - F6), and the MENU RECALL keys. When any of these keys are pressed, the number corresponding to the function key will be transmitted from the RP240 to the 500, SX, or ZX. Function keys 1—6 will return the values 1—6. The MENU RECALL key will return a zero.

Step 14 Press F1

After the F1 key is pressed, the RP240 will transmit the value 1 to the Model 500, SX, or ZX. This value will be stored in variable 2.

## Processing Information

Variables can be used in conditional statements like **IF**, **WHILE**, and **REPEAT**, or as data values in **D**, **V**, **A**, **AD**, **L**, or **T** commands. In step 7 above, the value 15 was entered stored in variable 1 to be used as the cycle count. The following is a command example:

| Command | Description |
|---|---|
| **D2400** | Set distance to 2400 steps |
| **V1** | Set velocity to 1 rps |
| **L(VAR1)** | Loop the number of times as specified by variable 1 |
| **G** | Initiate motion |
| **T1** | Wait one second |
| **N** | End the loop |

In step 14, the operator pressed the **F1** key. This value was stored in variable 2, and was to be used to select PART1. The following is a command example:

| Command | Description |
|---|---|
| **IF(VAR2=1) XG11 NIF** | Branch to sequence 11 if F1 is pressed |
| **IF(VAR2=2) XG12 NIF** | Branch to sequence 12 if F2 is pressed |
| **IF(VAR2=3) XG13 NIF** | Branch to sequence 13 if F3 is pressed |
| **IF(VAR2=4) XG14 NIF** | Branch to sequence 14 if F4 is pressed |
| **IF(VAR2=5) XG15 NIF** | Branch to sequence 15 if F5 is pressed |
| **IF(VAR2=6) XG16 NIF** | Branch to sequence 16 if F6 is pressed |

For further information on the additional commands used in the examples above, refer to the appropriate user guide, or software reference guide for the Model 500, SX, or ZX.

## Enabling STOP and PAUSE Keys

In addition to the function keys and numeric keypad, there are three other keys. The STOP key, and the PAUSE and CONTINUE keys, must be enabled before they can be used.

Typically, if an application uses the STOP key, the key will be enabled (**DSTP1**) in the power-up sequence (sequence #100). However, the STOP key can be enabled and disabled in any sequence, and at any time. The STOP key issues a Kill (**K**) command to the 500, SX, or ZX, immediately halting motion. ***There is no controlled deceleration when using the STOP key.***

The PAUSE and CONTINUE keys can be enabled in any sequence. Before these keys will function as expected, you must enable the commandSave Command Buffer on Stop (**SSH1**) and Resume Execution Enable (**SSL1**) commands. The **DCNT** command enables the PAUSE and CONTINUE keys, and the commands **SSH1** and **SSL1** cause the PAUSE and CONTINUE keys to function in different ways.

If **SSH** is disabled (**SSHØ**), the PAUSE key will function as a controlled stop. The motor will decelerate at the last programmed deceleration rate (**AD**) and the program buffer will be dumped. The program will be exited.

If you issue **SSH1** and **SSLØ**, the PAUSE key will function as a controlled stop. After motion is halted, the CONTINUE key will not resume motion. Instead, it will resume command processing with the command directly following the command that was stopped with the PAUSE key.

☞ Helpful Hint:
Both the SSH and SSL commands should be placed in the power-up sequence (Sequence #100).

If **SSH1** and **SSL1** are issued, the PAUSE key will function as a controlled stop. After motion is halted, command processing can be continued by pressing the CONTINUE key. If motion was in progress when the PAUSE key was pressed, that motion will be resumed. This is the recommended way to use the PAUSE and CONTINUE keys.

## Sample Program

This section provides an example of an RP240 application program. Refer to the specific indexer's Software Reference Guide for a more detailed description of the commands.

### Power-Up Sequence

The power-up sequence (Sequence #100) is used to initialize the SX, 500, or ZX to a state compatible with the RP240. The power-up sequence is also used to initialize variables.

```
XE100                              Erase sequence #100
XD100                              Begin definition of sequence #100
LD0                                Enable limits (only if connected, otherwise LD3)
SSH1                               Save command buffer on stop
SSN0                               Disable Message mode
DSTP1                              Enable STOP key
VAR20=5                            Variable 20 will be used for the high jog velocity
VAR21=1                            Variable 21 will be used for the low jog velocity
XG1                                Branch to sequence #1
XT                                 End definition of sequence #100
```

### Sequence #1

Sequence #1 provides the main menu for a demonstration program. Control is transferred to another sequence based on a function key input.

```
XE1                                Erase sequence #1
XD1                                Begin definition of sequence #1
DCLR0                              Clear all lines of the RP240 display
DPC106                             Position cursor at row 1, column 6
DTXTCOMPUMOTOR_RP240_DEMO_PROGRAM
DPC200                             Position cursor at row 2, column 0
DTXTACCESS
DPC235                             Position cursor at row 2, column 35
DTXTEXIT
VAR1=FUN                           Wait for function key and place value in variable 1
IF(VAR1=1)                         If function key 1 is pressed
    XG2                            Branch to sequence 2
NIF                                End If
IF(VAR1=6)                         If function key 6 is pressed
    DCLR0                          Clear all lines of the RP240 display
    SSA0                           Enable echo
    SSI0                           Enable interactive mode
    HALT                           Halt program execution
NIF                                End If
DPC220                             Position cursor at row 2, column 20
DTXTWRONG_BUTTON!
T1                                 Time delay of 1 second
XG1                                Branch to sequence #1
XT                                 End definition of sequence #1
```

### Sequence #2

Sequence #2 prompts for a code number, and then transfers control to sequence #3. If the code number is incorrect, control is passed back to sequence #1.

```
XE2                                Erase sequence #2
XD2                                Begin definition of sequence #2
DCLR0                              Clear all lines of the RP240 display
DPC100                             Position cursor at row 1, column 0
DTXTENTER_CODE_NUMBER_xxxx
DPC118                             Position cursor at row 1, column 18
VAR1=NUM                           Wait for numeric keypad input and place in variable 1
IF(VAR1<1234_OR_VAR1>1234)         If variable 1 is less than 1234 or greater than 1234
    DPC220                         Position cursor at row 2, column 20
    DTXTWRONG_NUMBER
    T1                             Time delay of 1 second
    XG1                            Branch to sequence #1
ELSE                               Else part of If statement (Variable 1 equals 1234)
    XG3                            Branch to sequence #3
NIF                                End If
XT                                 End definition of sequence #2
```

## Sequence #3

Sequence #3 asks for a selection via the function keys. Control is passed to a sequence based upon the function key pressed.

```
XE3                                 Erase sequence #3
XD3                                 Begin definition of sequence #3
DCLRØ                               Clear all lines of the RP240 display
DPC1ØØ                              Position cursor at row 1, column 1
DTXTJOG___I/O____TEACH__MAKE___FEED____EXIT
DPC2Ø1                              Position cursor at row 2, column 1
DTXTAXES__TEST___MODE___MOVE_TO_LENGTH
VAR1=FUN                            Wait for function key and place value in variable 1
IF(VAR1=1) XR4 NIF                  If function key 1 was pressed, gosub to sequence #4
IF(VAR1=2) XR5 NIF                  If function key 2 was pressed, gosub to sequence #5
IF(VAR1=3) XR6 NIF                  If function key 3 was pressed, gosub to sequence #6
IF(VAR1=4) XR7 NIF                  If function key 4 was pressed, gosub to sequence #7
IF(VAR1=5) XR8 NIF                  If function key 5 was pressed, gosub to sequence #8
IF(VAR1=6) XG1 NIF                  If function key 6 was pressed, branch to sequence #1
XG3                                 Branch to sequence #3
XT                                  End definition of sequence #3
```

## Sequence #4

Sequence #4 simulates jogging with CW or CCW jog options, and either low or high velocity. The default jog velocities are stored in variables 20 and 21, which were assigned in power-up sequences (sequence #100).

```
XE1                                 Erase sequence #4
XD1                                 Begin definition of sequence #4
A75 AD75                            Set acceleration and deceleration to 75 rev/sec/sec
VAR1=Ø VAR2=Ø                       Initialize variables 1 and 2 to zero
MC                                  Enable continuous mode
MPP                                 Enable position profiling mode
REPEAT                              Repeat all the commands until the UNTIL condition is true
    VAR3=Ø                          Set variable 3 equal to zero
    DCLRØ                           Clear all lines of the RP240 display
    DPC1Ø2                          Position the cursor at row 1, column 2
    DTXTCW_____CW____CCW____CCW__________STOP
    DPC2Ø1                          Position the cursor at row 2, column 1
    DTXTLOW____HIGH___LOW___HIGH_________MOTION
    VAR1=FUN                        Wait for function key and place value in variable 1
    WHILE(VAR2<VAR1_OR_VAR2>VAR1_AND_VAR3=0)
        VØ                          Set velocity to zero rev/sec
        TØ.3                        Wait for 0.3 seconds
        IF(VAR1=1)                  If function key 1 was pressed
            H+ V(VAR21) G           Set direction, velocity, and initiate motion
        NIF                         End If
        IF(VAR1=2)                  If function key 2 was pressed
            H+ V(VAR2Ø) G           Set direction, velocity, and initiate motion
        NIF                         End If
        IF(VAR1=3)                  If function key 3 was pressed
            H- V(VAR21) G           Set direction, velocity, and initiate motion
        NIF                         End If
        IF(VAR1=4)                  If function key 4 was pressed
            H+ V(VAR2Ø) G           Set direction, velocity, and initiate motion
        NIF                         End If
        IF(VAR1=5)                  If function key 5 was pressed
            H+ V(VAR21) G           Set direction, velocity, and initiate motion
        NIF                         End If
        IF(VAR1=1)                  If function key 1 was pressed
            DCLRØ                   Clear all lines of the RP240 display
            DPC212                  Position the cursor at row 2, column 12
            DTXTINVALID_SELECTION
            T1                      Wait for 1 second
            XG4                     Branch to sequence #4
        NIF                         End If
        VAR3=1                      Set variable 3 to one
    NWHILE                          End WHILE loop
    VAR2=VAR1                       Set variable 2 equal to variable 1
UNTIL(VAR1=6)                       Exit REPEAT UNTIL loop when function key 6 is pressed
VØ                                  Set velocity to zero
NG                                  Exit position profiling (MPP) mode
MN                                  Mode normal (exit mode continuous)
VAR1=Ø                              Set variable 1 equal to zero.
XT                                  End definition of sequence #4
```

## Sequence #5

Sequence #5 provides input status information and then toggles the eight LEDs of the RP240 on and off.

```
XE5                                 Erase sequence #5
XD5                                 Begin definition of sequence #5
DCLRØ                               Clear all lines of the RP240 display
DPC1ØØ                              Position the cursor at row 1, column 0
DTXTINPUT_STATUS=
DPC2ØØ                              Position the cursor at row 2, column 0
DTXTUPDATE
DPC233                              Position the cursor at row 2, column 33
DTXTOUTPUTS
REPEAT                              Repeat all the commands until the UNTIL condition is true
    VAR1=FUN                        Wait for function key and place value in variable 1
    DPC113                          Position the cursor at row 1, column 13
    IF(IN1) DTXT1 ELSE DTXTØ NIF
    IF(INX1) DTXT1 ELSE DTXTØ NIF
    IF(INXX1) DTXT1 ELSE DTXTØ NIF
    IF(INXXX1) DTXT1 ELSE DTXTØ NIF
    IF(INXXXX1) DTXT1 ELSE DTXTØ NIF
    IF(INXXXXX1) DTXT1 ELSE DTXTØ NIF
    IF(INXXXXXX1) DTXT1 ELSE DTXTØ NIF
    IF(INXXXXXXX1) DTXT1 ELSE DTXTØ NIF
UNTIL(VAR1=6)                       Exit REPEAT UNTIL loop when function key 6 is pressed
DCLRØ                               Clear all lines of the RP240 display
DPC1ØØ                              Position the cursor at row 1, column 0
DTXTTHE_LEDS_BELOW_WILL_CHANGE_STATE
DLED00000001 T1                     Turn on LED 8 and wait 1 second
DLED00000011 T1                     Turn on LEDs 7 and 8 and wait 1 second
DLED00000111 T1                     Turn on LEDs 6, 7 and 8 and wait 1 second
DLED00001111 T1                     Turn on LEDs 5, 6, 7 and 8 and wait 1 second
DLED00011111 T1                     Turn on LEDs 4, 5, 6, 7 and 8 and wait 1 second
DLED00111111 T1                     Turn on LEDs 3, 4, 5, 6, 7 and 8 and wait 1 second
DLED01111111 T1                     Turn on LEDs 2, 3, 4, 5, 6, 7 and 8 and wait 1 second
DLED11111111 T1                     Turn on LEDs 1, 2, 3, 4, 5, 6, 7 and 8 and wait 1 second
VAR1=Ø                              Set variable 1 equal to zero.
XT                                  End definition of sequence #5
```

## Sequence #6

Sequence #6 demonstrates teaching moves that can be executed from sequence #7. This sequence uses the jog routine from sequence #4 to move to the desired locations. The distances taught are assigned to variables using other sequences (13, 14, and 15 in this example). The total number of moves taught is limited by the number of variables and the number of available sequences.

```
XE6                                 Erase sequence #6
XD6                                 Begin definition of sequence #6
MPA                                 Enable absolute positioning mode
MN                                  Enable preset mode
VAR28=13                            Set variable 28 equal to 13 (First sequence to teach)
VAR29=1                             Set variable 29 equal to 1
VAR3Ø=1                             Set variable 30 equal to 1
REPEAT                              Repeat all the commands until the UNTIL condition is true
    DCLRØ                           Clear all lines of the RP240 display
    DPC1ØØ                          Position the cursor at row 1, column 0
    DTXTHOW_MANY_MOVES_TO_TEACH?_x
    DPC2ØØ                          Position the cursor at row 2, column 0
    DTXT(MAXIMUM_OF_THREE_MOVES)
    DPC125                          Position the cursor at row 1, column 25
    VAR3Ø=NUM                       Wait for numeric keypad input and place in variable 30
UNTIL(VAR3Ø>Ø_AND_VAR3Ø<4)          Exit REPEAT UNTIL loop when number entered is 1,2 or 3
L(VAR3Ø)                            Loop the number of times specified in variable 30
    PZ                              Set the current position to absolute zero
    DCLRØ                           Clear all lines of the RP240 display
    DPC1ØØ                          Position the cursor at row 1, column 0
    DTXTPLEASE MAKE MOVE#_
    DVO29,2,Ø,Ø                     Display variable 29
    DPC2ØØ                          Position the cursor at row 2, column 0
    T1                              Wait 1 second
    XR4                             Gosub to sequence #4
    XR(VAR28)                       Gosub to the sequence specified in variable 28
    VAR28=VAR28+1                   Increment variable 28 by 1
    VAR29=VAR29+1                   Increment variable 29 by 1
N                                   End loop
VAR1=Ø                              Set variable 1 equal to zero.
XT                                  End definition of sequence #6
```

## Sequence #7

Sequence #7 performs the moves taught in sequence #6. The distance for each move is based upon the distance that was stored in a variable in sequence #6.

```
XE7                              Erase sequence #7
XD7                              Begin definition of sequence #7
A1ØØ                             Set acceleration to 100 rev/sec/sec
AD1ØØ                            Set deceleration to 100 rev/sec/sec
MPI                              Enable incremental position mode
PZ                               Set the current position to absolute zero
V1Ø                              Set velocity to 10 rev/sec
VAR29=2Ø                         Set variable 29 equal to twenty
L(VAR3Ø)                         Loop the number of times specified by variable 30
    XR(VAR29)                    Gosub to the sequence specified by variable 29
    G                            Initiate motion
    VAR29=VAR29+1                Increment variable 29
N                                End loop
VAR1=Ø                           Set variable 1 equal to zero.
XT                               End definition of sequence #7
```

## Sequence #8

Sequence #8 prompts the operator to select the feed length, maximum speed and count of items to be cut.

```
XE8                              Erase sequence #8
XD8                              Begin definition of sequence #8
DCLRØ                            Clear all lines of the RP240 display
DPC1Ø1                           Position the cursor at row 1, column 1
DTXTSELECT_FEED_LENGTH,_MAX_SPEED,_&_COUNT
T2                               Wait for 2 seconds
REPEAT                           Repeat all the commands until the UNTIL condition is true
   DCLRØ                         Clear all lines of the RP240 display
   VAR22=Ø                       Set variable 22 equal to zero (used to exit repeat loop)
   DPC1Ø2                        Position the cursor at row 1, column 2
   DTXTENTER_FEED_LENGTH_IN_INCHES:_(Ø-99.9)
   DPC212                        Position the cursor at row 2, column 12
   DTXTFEED_LENGTH=xx.x
   DPC224                        Position the cursor at row 2, column 24
   VAR2=NUM                      Wait for numeric keypad input and place in variable 2
   IF(VAR2>99.9_OR_VAR2<Ø)       If variable 2 is greater than 99.9 or less than 0
        DCLRØ                    Clear all lines of the RP240 display
        DPC215                   Position the cursor at row 2, column 15
        DTXTOUT_OF_RANGE
        T2                       Wait for two seconds
    ELSE                         Else (variable 2 is greater than 0 and less than 99.9)
        DCLRØ                    Clear all lines of the RP240 display
        DPC112                   Position the cursor at row 1, column 12
        DTXTFEED_LENGTH=
        DPC125                   Position the cursor at row 1, column 25
        DVO2,2,1,Ø               Display variable 2
        DPC2Ø2                   Position the cursor at row 2, column 2
        DTXTYES
        DPC236                   Position the cursor at row 2, column 36
        DTXTNO
        VAR22=FUN                Wait for a function key to be pressed
    NIF                          End If
UNTIL(VAR22=1)                   Exit REPEAT UNTIL loop when variable 22 equals one
REPEAT                           Repeat all the commands until the UNTIL condition is true
   DCLRØ                         Clear all lines of the RP240 display
   VAR22=Ø                       Set variable 22 equal to zero (used to exit repeat loop)
   DPC1Ø1                        Position the cursor at row 1, column 1
   DTXTENTER_MAX_SPEED_(RPM):_(Ø-24ØØRPM)
   DPC215                        Position the cursor at row 2, column 15
   DTXTMAX_SPEED=xxxx
   DPC225                        Position the cursor at row 2, column 25
   VAR3=NUM                      Wait for numeric keypad input and place in variable 3
   IF(VAR3>24ØØ_OR_VAR3<Ø)       If variable 3 is greater than 2400 or less than 0
        DCLRØ                    Clear all lines of the RP240 display
        DPC215                   Position the cursor at row 2, column 15
        DTXTOUT_OF_RANGE
        T2                       Wait for two seconds
    ELSE                         Else (variable 3 is greater than 0 and less than 2400)
        DCLRØ                    Clear all lines of the RP240 display
        DPC114                   Position the cursor at row 1, column 14
        DTXTMAX_SPEED=
        DPC125                   Position the cursor at row 1, column 25
        DVO3,4,2,Ø               Display variable 3
        DPC2Ø2                   Position the cursor at row 2, column 2
```

```
        DTXTYES
        DPC236                  Position the cursor at row 2, column 36
        DTXTNO
        VAR22=FUN               Wait for a function key to be pressed
    NIF                         End If
UNTIL(VAR22=1)                  Exit REPEAT UNTIL loop when variable 22 equals one
REPEAT                          Repeat all the commands until the UNTIL condition is true
    DCLRØ                       Clear all lines of the RP240 display
    VAR22=Ø                     Set variable 22 equal to zero (used to exit repeat loop)
    DPC1Ø3                      Position the cursor at row 1, column 1
    DTXTENTER_TOTAL_NUMBER_OF_CUTS:_(1-1ØØ)
    DPC21Ø                      Position the cursor at row 2, column 10
    DTXTTOTAL_#_OF_CUTS=xxx
    DPC226                      Position the cursor at row 2, column 26
    VAR4=NUM                    Wait for numeric keypad input and place in variable 4
    IF(VAR4>1ØØ_OR_VAR4<1)      If variable 4 is greater than 100 or less than 1
        DCLRØ                   Clear all lines of the RP240 display
        DPC215                  Position the cursor at row 2, column 15
        DTXTOUT_OF_RANGE
        T2                      Wait for two seconds
    ELSE                        Else (variable 4 is greater than 1 and less than 100)
        DCLRØ                   Clear all lines of the RP240 display
        DPC113                  Position the cursor at row 1, column 13
        DTXT#_OF_CUTS=
        DPC124                  Position the cursor at row 1, column 24
        DVO4,3,Ø,Ø              Display variable 3
        DPC2Ø2                  Position the cursor at row 2, column 2
        DTXTYES
        DPC236                  Position the cursor at row 2, column 36
        DTXTNO
        VAR22=FUN               Wait for a function key to be pressed
    NIF                         End If
UNTIL(VAR22=1)                  Exit REPEAT UNTIL loop when variable 22 equals one
REPEAT                          Repeat all the commands until the UNTIL condition is true
    DCLRØ                       Clear all lines of the RP240 display
    DPC2Ø1                      Position the cursor at row 2, column 1
    DTXTSTART
    DPC235                      Position the cursor at row 2, column 35
    DTXTEXIT
    VAR22=FUN                   Wait for a function key to be pressed
    IF(VAR22=1_OR_VAR22=6)      If variable 1 equals 1 or 6 do nothing
    ELSE                        Else, display an error message
        DCLRØ                   Clear all lines of the RP240 display
        DPC212                  Position the cursor at row 2, column 12
        DTXTINVALID_SELECTION
        T2                      Wait two seconds
    NIF                         End If
UNTIL(VAR22=1_OR_VAR22=6)       Exit REPEAT UNTIL loop when variable 22 equals 1 or 6
IF(VAR22=6)                     If variable 22 equals 6
    XG3                         Branch to sequence #3
NIF                             End If
DCLRØ                           Clear all lines of the RP240 display
VAR5=VAR2*25ØØØ                 Variable 5 equals distance, assuming 25000 steps/inch
VAR6=VAR3/6Ø                    Variable 6 equals velocity in RPM
VAR7=VAR4                       Using variable 7 as count down value
LD3                             Enable limits
MN                              Mode normal (preset moves)
A1ØØ                            Set acceleration to 100 rev/sec/sec
AD1ØØ                           Set deceleration to 100 rev/sec/sec
V(VAR6)                         Set velocity to the value specified by variable 6
D(VAR5)                         Set distance to the value specified by variable 5
DPC211                          Position the cursor at row 2, column 11
DTXT#_LEFT_TO_CUT:
L(VAR4)                         Loop the number of times specified by variable 4
    DPC225                      Position the cursor at row 2, column 25
    DVO7,3,Ø,Ø                  Display variable 7
    G                           Initiate motion
    VAR7=VAR7-1                 Decrement variable 7
    T.1                         Wait 0.1 seconds
N                               End loop
DCLRØ                           Clear all lines of the RP240 display
DPC213                          Position the cursor at row 2, column 13
DTXTJOB_COMPLETED!
T1.5                            Set distance to the value specified by variable 5
XG3                             Branch to sequence #3
XT                              End definition of sequence #8
```

### Sequences #13,#14,#15

Sequences #13, #14, and #15 store the current position into variables 6, 7, and 8, respectively.

```
XE13                    Erase sequence #13
XD13                    Begin definition of sequence #13
    VAR6=POS            Variable 6 equals the current position
XT                      End definition of sequence #13
XE14                    Erase sequence #14
XD14                    Begin definition of sequence #14
    VAR7=POS            Variable 7 equals the current position
XT                      End definition of sequence #14
XE15                    Erase sequence #15
XD15                    Begin definition of sequence #15
    VAR8=POS            Variable 8 equals the current position
XT                      End definition of sequence #15
```

### Sequences #20,#21,#22

Sequences #20, #21, and #22 set distance to the value stored in variables 6, 7, and 8, respectively.

```
XE2Ø                    Erase sequence #20
XD2Ø                    Begin definition of sequence #20
    D(VAR6)             Set distance to the value stored in variable 6
XT                      End definition of sequence #20
XE21                    Erase sequence #21
XD21                    Begin definition of sequence #21
    D(VAR7)             Set distance to the value stored in variable 7
XT                      End definition of sequence #21
XE22                    Erase sequence #22
XD22                    Begin definition of sequence #22
    D(VAR8)             Set distance to the value stored in variable 8
XT                      End definition of sequence #22
```

## Editing Sequences

If you wish to edit sequences in your SX, ZX, or Model 500, you must re-establish RS-232C communication ***before*** downloading the new sequences to the controller. Compumotor recommends using the following procedure.

① Remove power from your controller and the RP240.

② Remove connections between the controller and the RP240.

③ Connect Rx, Tx, and Ground between the controller and your RS-232C port.

④ Apply power to your controller.

⑤ Enter the TERMINAL EMULATOR portion of your communication software (preferably Compumotor's X-Ware).

Initially you will not be able to establish communications if you have turned off command echo (**SSA1**) in your power-up sequence.

⑥ If using X-Ware **Device Not Ready** will appear on your screen (if you have disabled the command echo). Press the <**ENTER**> key. Next, press the **Y** key to enter the Terminal Emulator.

⑦ To terminate the power-up sequence, and turn on the command echo as well as the interactive mode, issue the following commands:

**K** <cr>
**SSAØ** <cr>
**SSIØ** <cr>

⑧ Check the controller's status by issuing a **aR** command, where **a** is the device address (**1R** <cr>, for example). The controller should respond with a ***R.** If you do not receive a ***R**, consult the RS-232C Troubleshooting section of your controller user guide.

⑨ Edit the sequence as usual (If using X-Ware, hit the <ESC> key, then use arrow keys to select the Editor menu).

# Daisy Chaining

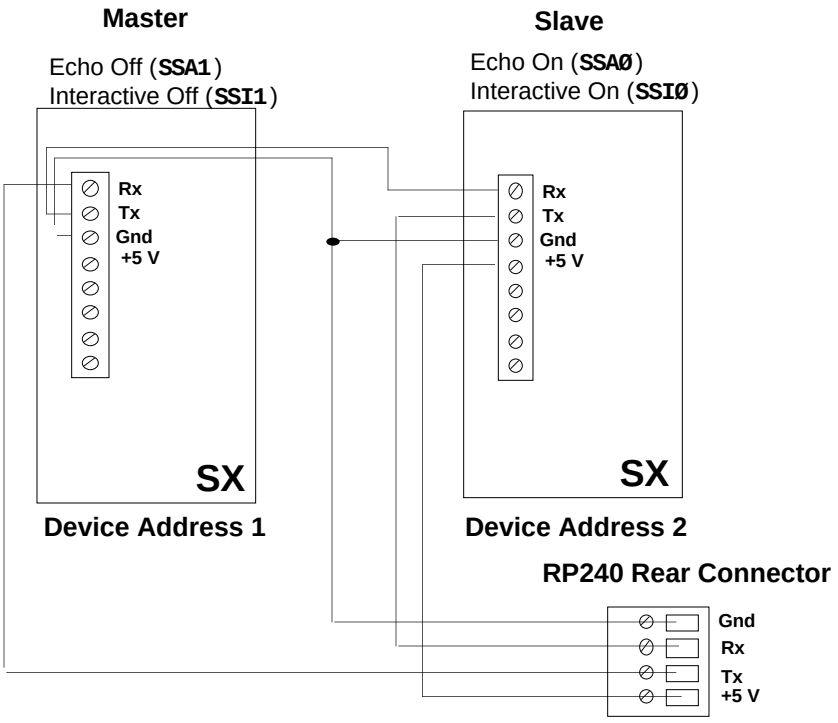
This section describes daisy chaining multiple Extended X language products with an RP240.

More than one SX, Model 500, or ZX can be daisy chained to a single RP240. Daisy chaining up to 16 units is not a trivial programming assignment. When multiple units are daisy chained together, one unit must act as the master, controlling all the slaves down the line. The RP240 must be placed at the end of the daisy chain, where it is a slave. The master unit must control all the messages displayed on the RP240, in addition to processing the input from the RP240. The master unit must also control the motion of all the other slave units. Controlling the motion of the slave units can be accomplished by either pre-programming sequences in the slave units and activating them with I/O from the master, or by sending the motion command across the RS-232 daisy chain via the Quote (**"**) command or the Transmit Data (**TX**) command. In either case, motion between the master and slave units must be coordinated. It is good programming practice to download all desired sequences to each controller *before* wiring the daisy chain.

> ***CAUTION***
>
> When programming two daisy chained units, take caution in using the DSTP and DCNT command. The STOP and PAUSE keys only affect the master unit (i.e., if you issue a K command only the master unit will stop).

To set up the daisy chain, some command statements must be issued. Command Echo (**SSA**) and Interactive mode (**SSI**) must be turned off on the first unit (master) in the daisy chain. Command echo in all the succeeding units (slaves) must be turned on.

☞ Helpful Hint: Daisy Chain Wiring Diagram




## Daisy Chain Sample Program

The following sample program demonstrates how to daisy chain two units and control the second across RS-232C. The program is downloaded to the master (device address 1).

## Power-Up Sequence

The power-up sequence (Sequence #100) is used to initialize the SX, 500, or ZX to a state compatible with the RP240.

| Command | Description |
|---|---|
| **1XE1ØØ** | Erase sequence #100 |
| **1XD1ØØ** | Begin definition of sequence #100 |
| **1LD3** | Disable limits (Use **LDØ** if limits are hard wired) |
| **1SSA1** | Disable command echo |
| **1SSH1** | Save command buffer on STOP |
| **1SSI1** | Disable Interactive mode |
| **1"2E** | Enable communications on axis 2 |
| **1"2LD3** | Disable limits on axis 2 |
| **1"2SSAØ** | Enable command echo on axis 2 |
| **1"2SSIØ** | Enable interactive mode on axis 2 |
| **1"2F** | Disable communications on axis 2 |
| **1DSTPØ** | Disable **STOP** key |
| **1DCNTØ** | Disable **PAUSE** and **CONTINUE** keys |
| **1XG1** | Branch to sequence #1 |
| **1XT** | End definition of sequence #100 |

## Sequence #1

Sequence #1 instructs you to enter the axis to jog. The you are allowed to jog axis one or axis two.

```
1XE1
1XD1
1A1ØØ
1V4
1MC
1"2E
1"2A1ØØ
1"2V2
1"2MC
1"2F
1DCLRØ
1DPC1Ø1
1DTXTWHICH_AXIS_DO_YOU_WANT_TO_JOG?
1DPC2ØØ
1DTXTAXIS_1__AXIS_2
1DPC235
1DTXTEXIT
1L
1VAR1=FUN
1IF(VAR1=1) 1XG2 1NIF
1IF(VAR1=2) 1XG3 1NIF
1IF(VAR1=6) 1DCLRØ 1SSAØ 1SSIØ 1"2E 1HALT 1NIF
1N
1XT
```

## Sequence #2

Sequence #2 jogs axis 1.

```
XE2
XD2
1DCLRØ
1DPC1Ø2
1DTXTJOG___STOP___JOG
1DPC2Ø2
1DTXTCCW__________CW
1DPC235
1DTXTEXIT
1MPP
1L
1VAR1=FUN
1IF(VAR1=1) 1H- 1G 1NIF
1IF(VAR1=2) 1STOP 1NIF
1IF(VAR1=3) 1H+ 1G 1NIF
1IF(VAR1=6) 1STOP 1XG1 1NG 1NIF
1N
1NG
XT
```

## Sequence #3

Sequence #3 jogs axis 2.

```
XE3
XD3
1DCLRØ
1DPC1Ø2
1DTXTJOG___STOP___JOG
1DPC2Ø2
1DTXTCCW__________CW
1DPC235
1DTXTEXIT
1MPP
1L
1VAR1=FUN
1IF(VAR1=1) 1"2E 1"2H- 1"2G 1"2F 1NIF
1IF(VAR1=2) 1"2E 1"2S 1"2F 1NIF
1IF(VAR1=3) 1"2E 1"2H+ 1"2G 1"2F 1NIF
1IF(VAR1=6) 1"2E 1"2S 1"2F 1XG1 1NG 1NIF
1N
1NG
XT
```