Application Example Programs

This appendix is designed to illustrate various instructions described previously in this manual. Application example programs include:

- paper drilling machine using most of the instructions.
- time driven sequencer using TON and SQO instructions.
- event driven sequencer using SQC and SQO instructions.
- on/off circuit using basic, program flow, and application specific instructions.
- interfacing with enhanced bar code decoders over DH-485.

Because of the variety of uses for this information, the user of and those responsible for applying this information must satisfy themselves as to the acceptability of each application and use of the program. In no event will Rockwell Automation be responsible or liable for indirect or consequential damages resulting from the use of application of this information.

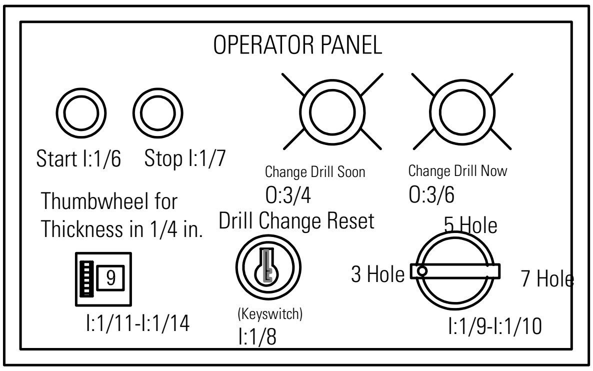
The illustrations, chart, and examples shown in this appendix are intended solely to illustrate the principles of the controller and some of the methods used to apply them. Particularly because of the many requirements associated with any particular installation, Allen-Bradley Company cannot assume responsibility or liability for actual use based upon the illustrative uses and applications.

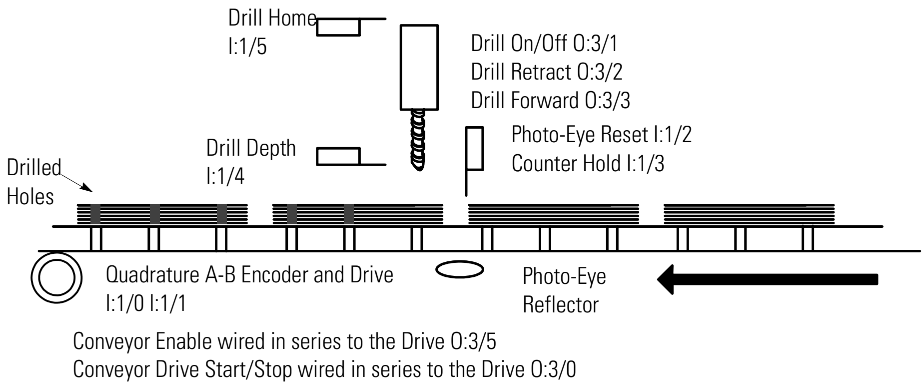
Paper Drilling Machine Application Example

For a detailed explanation of:

- XIC, XIO, OTE, RES, OTU, OTL, and OSR instructions, see chapter 2.
- EQU and GEQ instructions, see chapter 3.
- CLR, ADD, and SUB instructions, see chapter 4.
- MOV and FRD instructions, see chapter 5.
- JSR and RET instructions, see chapter 6.
- INT instruction, see chapter 11
- SQO instruction, see chapter 7.

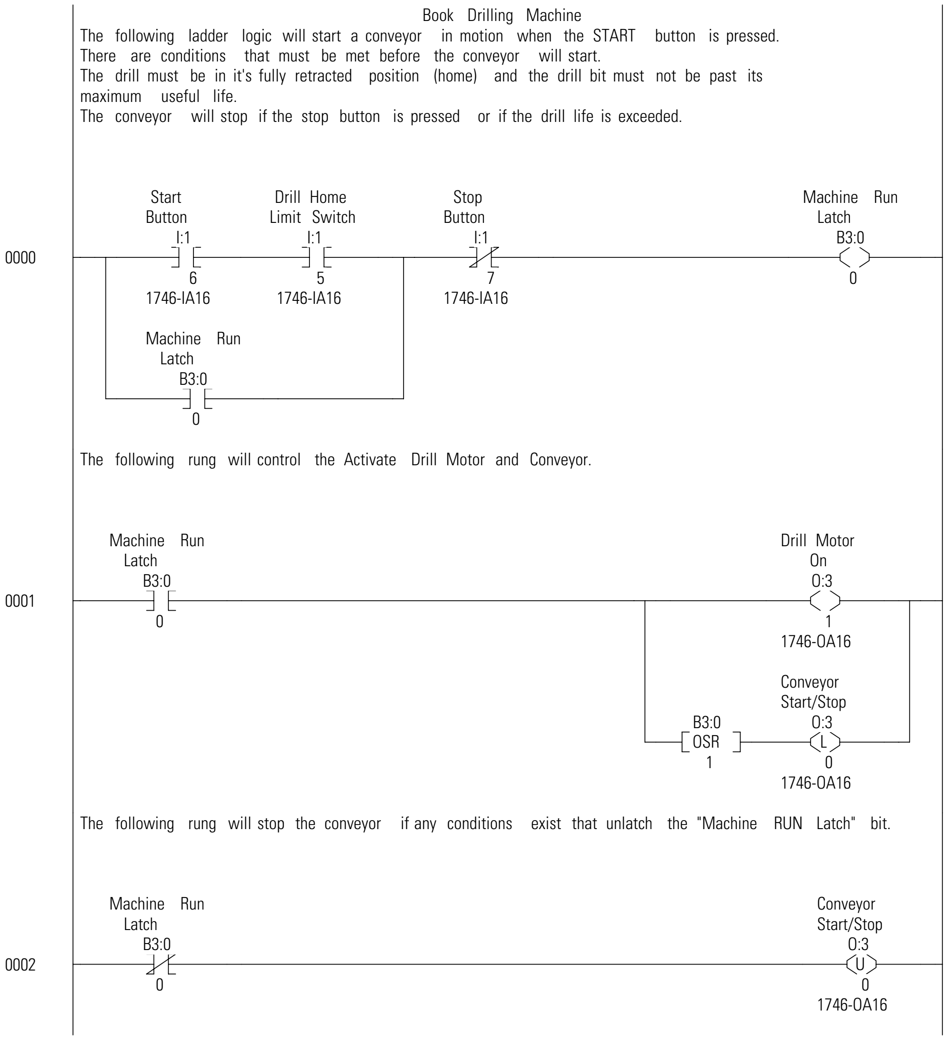
This machine can drill 3 different hole patterns into bound manuals. The program tracks drill wear and signals the operator that the bit needs replacement. The machine shuts down if the signal is ignored by the operator.

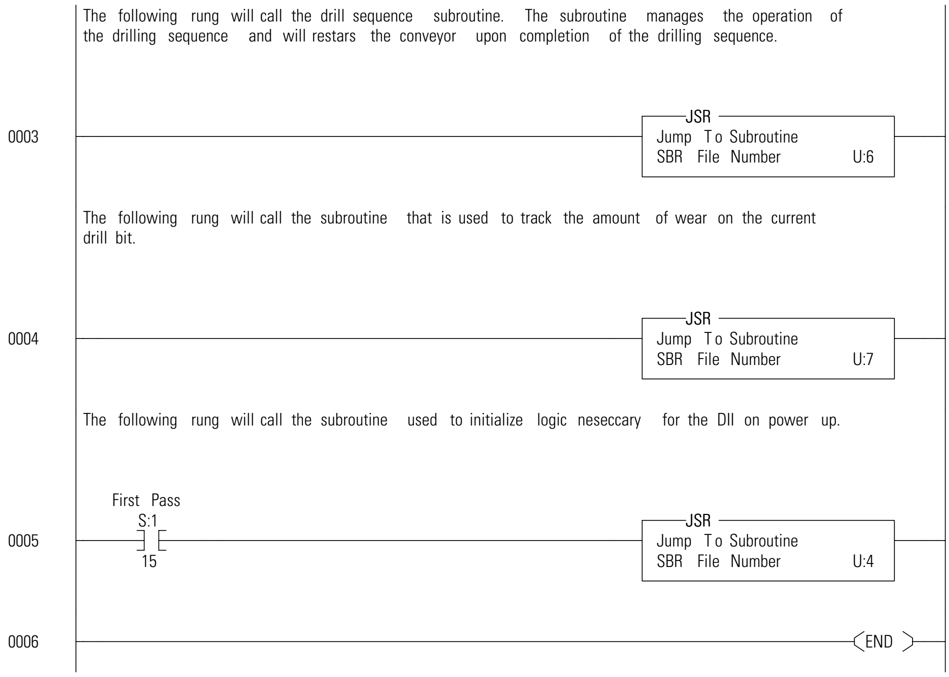

OPERATOR PANEL
Start I:1/6
Stop I:1/7
Change Drill Soon
O:3/4
Change Drill Now
O:3/6
Thumbwheel for
Thickness in 1/4 in.
Drill Change Reset
5 Hole
3 Hole
7 Hole
9
I:1/11-I:1/14
(Keyswitch)
I:1/8
I:1/9-I:1/10


Drill Home
I:1/5
Drill On/Off O:3/1
Drill Retract O:3/2
Drill Forward O:3/3
Drill Depth
I:1/4
Photo-Eye Reset I:1/2
Counter Hold I:1/3
Drilled
Holes
Quadrature A-B Encoder and Drive
I:1/0 I:1/1
Photo-Eye
Reflector
Conveyor Enable wired in series to the Drive O:3/5
Conveyor Drive Start/Stop wired in series to the Drive O:3/0

Paper Drilling Machine Operation Overview

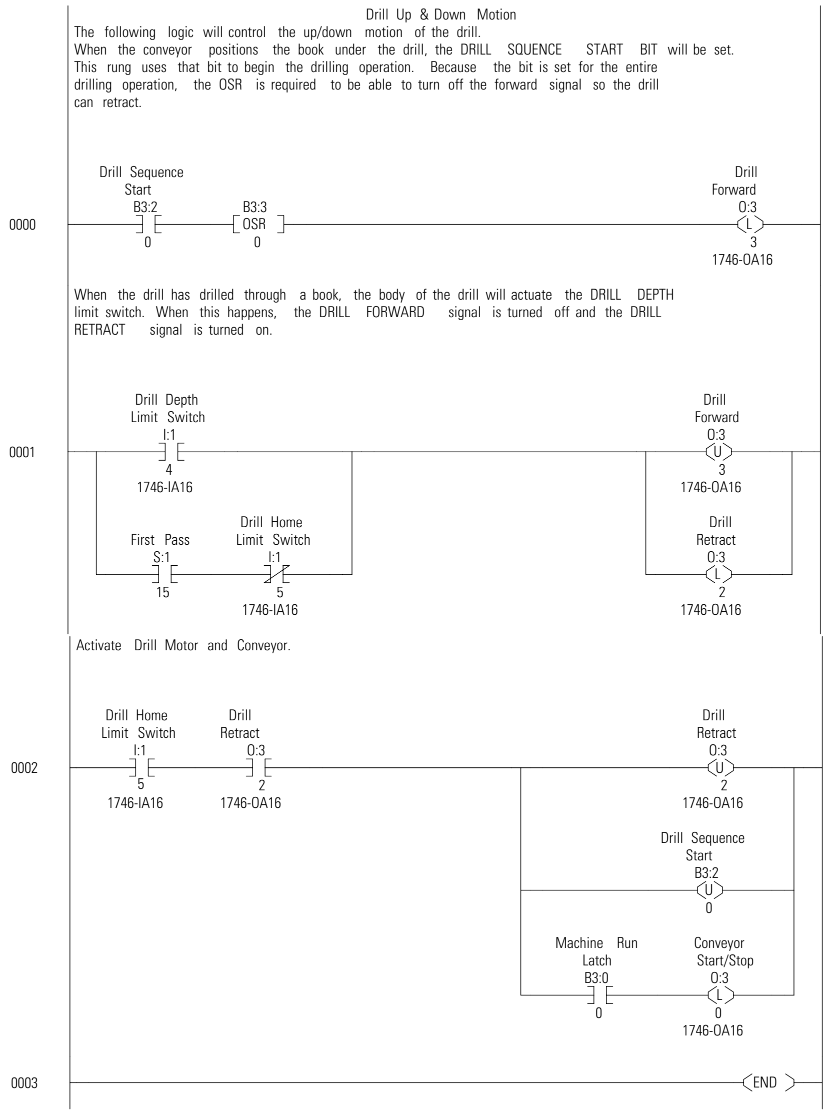
Undrilled books are placed onto a conveyor taking them to a single spindle drill. Each book moves down the conveyor until it reaches the first drilling position. The conveyor stops moving and the drill lowers and drills the first hole. The drill then retracts and the conveyor moves the same book to the second drilling position. The drilling process is repeated until there are the desired holes per book.




The following rung will call the drill sequence subroutine. The subroutine manages the operation of the drilling sequence and will restars the conveyor upon completion of the drilling sequence.
0003
JSR
Jump To Subroutine
SBR File Number U:6
The following rung will call the subroutine that is used to track the amount of wear on the current drill bit.
0004
JSR
Jump To Subroutine
SBR File Number U:7
The following rung will call the subroutine used to initialize logic neseccary for the DII on power up.
First Pass
S:1
15
0005
JSR
Jump To Subroutine
SBR File Number U:4
0006
END

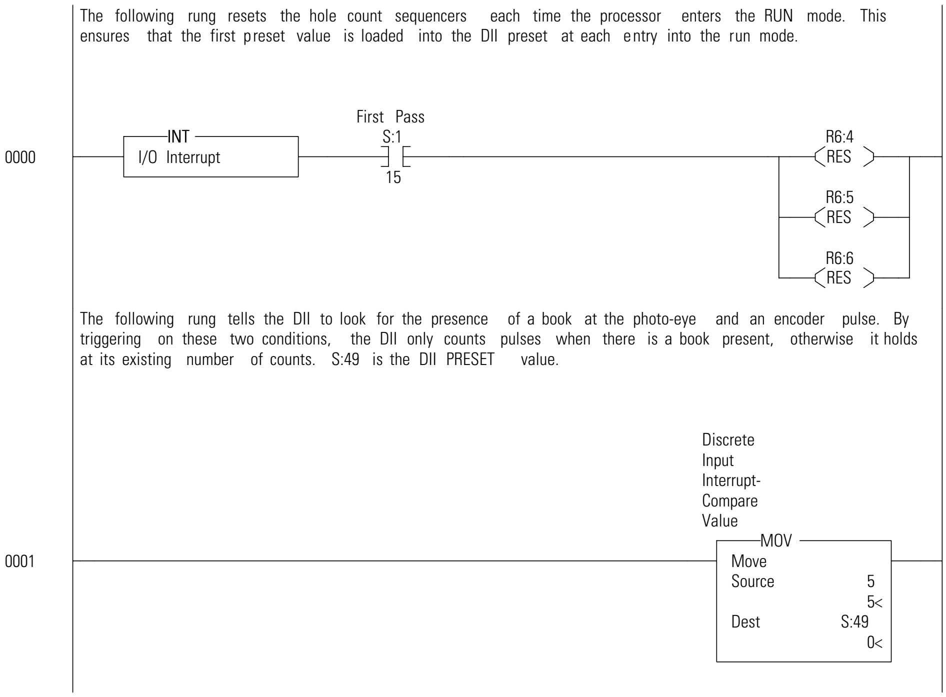
Drill Mechanism Operation

When the operator presses the start button, the drill motor turns on. After the book is in the first drilling position, the conveyor subroutine sets a drill sequence start bit, and the drill moves toward the book. When the drill has drilled through the book, the drill body hits a limit switch and causes the drill to retract up out of the book. When the drill body is fully retracted, the drill body hits another limit switch indicating that it is in the home position. Hitting the second limit switch unlatches the drill sequence start bit and causes the conveyor to move the book to the next drilling position.

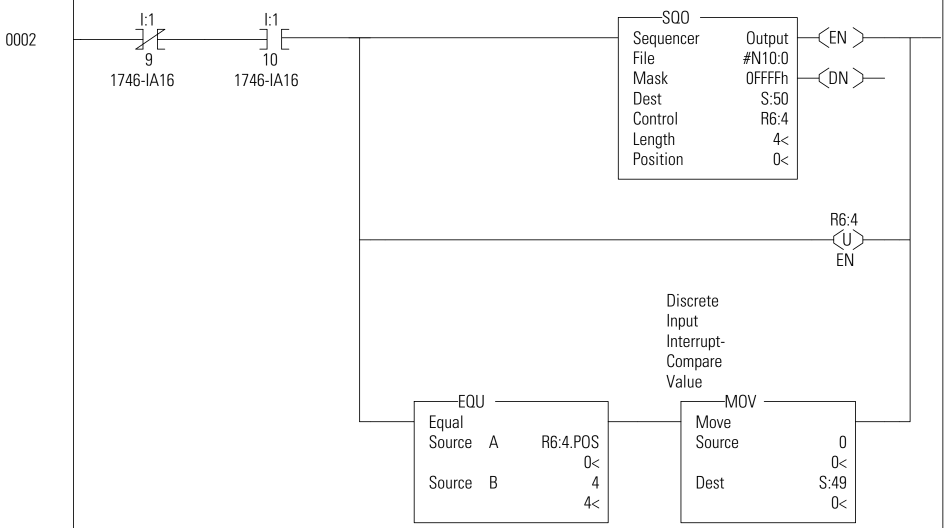


Conveyor Operation

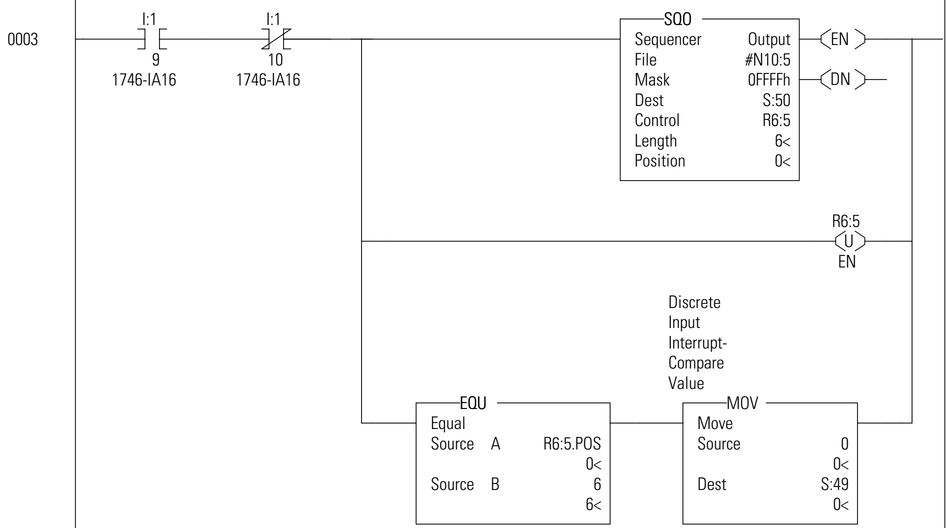
When the start button is pressed, the conveyor moves the books forward. As the first book moves close to the drill, the book trips a photo-eye sensor. This tells the machine where the leading edge of the book is. Based on the position of the selector switch, the conveyor moves the book until it reaches the first drilling position. The drill sequence start bit is set and the first hole is drilled. The drill sequence start bit is now unlatched and the conveyor moves the same book to the second drilling position. The drilling process is then repeated until there are the desired holes per book. The machine then looks for another book to break the photo-eye beam and the process is repeated. The operator can change the number of drilled holes by changing the selector switch.



The following rung will keep track of the hole number that is being drilled and loads the next correct DII preset based on the hole count. This rung is only active when the "HOLE SELECTOR" switch is in the "3-HOLE" position. The sequencer uses step 0 as a null step upon reset. It uses the last step as a "go forever" in anticipation of the "end of manual". Moving a 0 into S:49 tells the DII to trigger an interrupt when the trailing edge of the current book is detected.

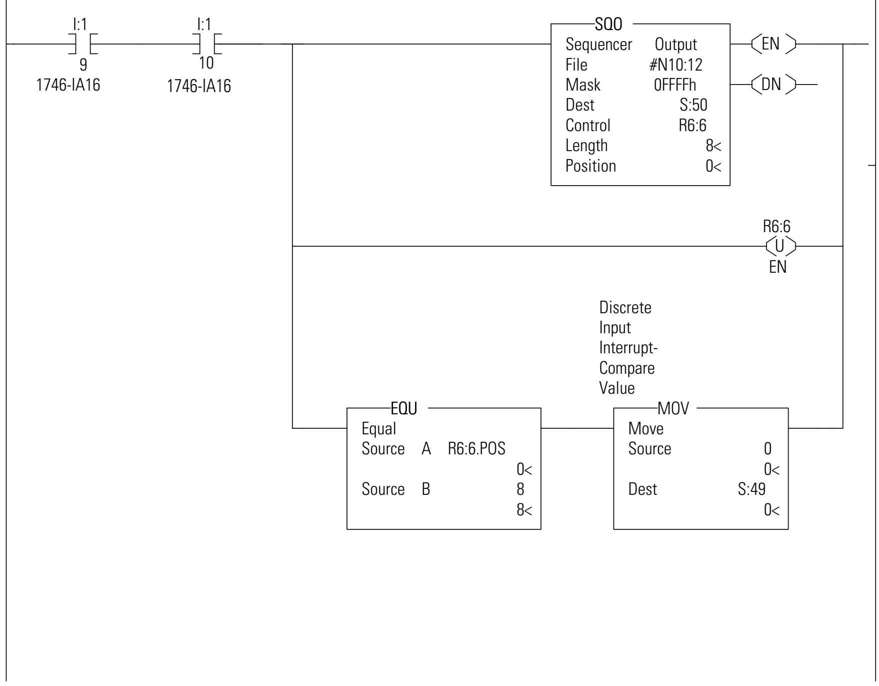


The following rung is identical to the previous rung except that it is only active when the "HOLE SELECTOR" switch is in the "5-HOLE" position.



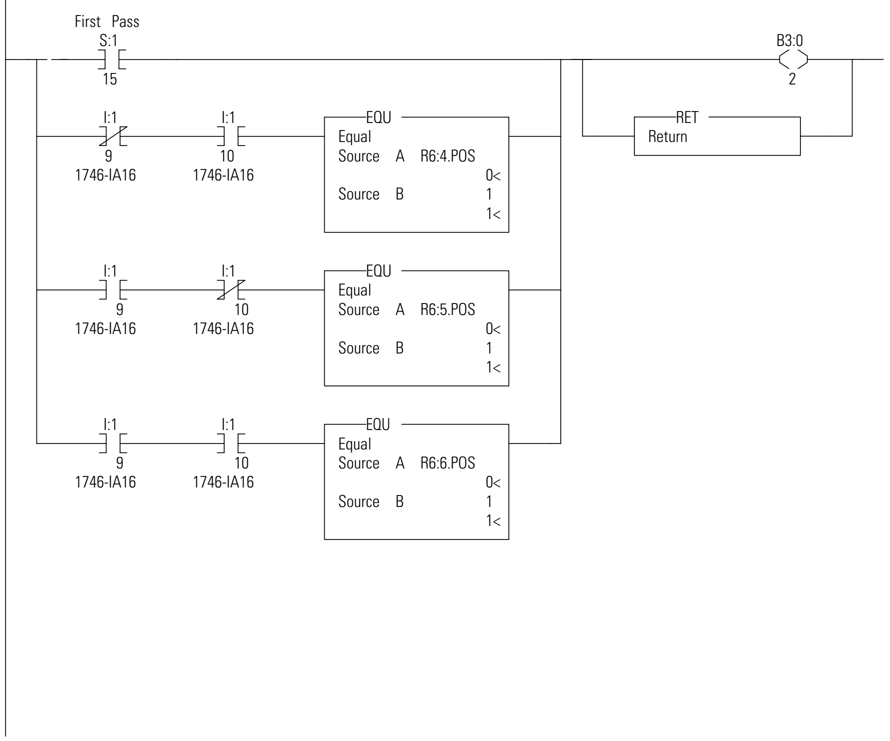
The following rung is identical to the previous two rungs except that it is only active when the "HOLE SELECTOR" switch is in the "7-HOLE" position.

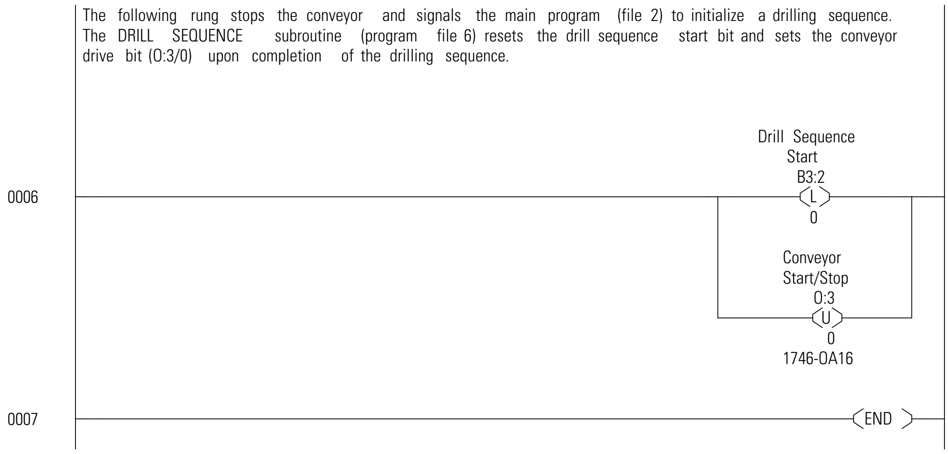
0004



If the processor is in this subroutine either for initialization or due to sensing the trailing edge of a manual, just return and skip the logic that stops the conveyor and starts the drill sequence.

0005

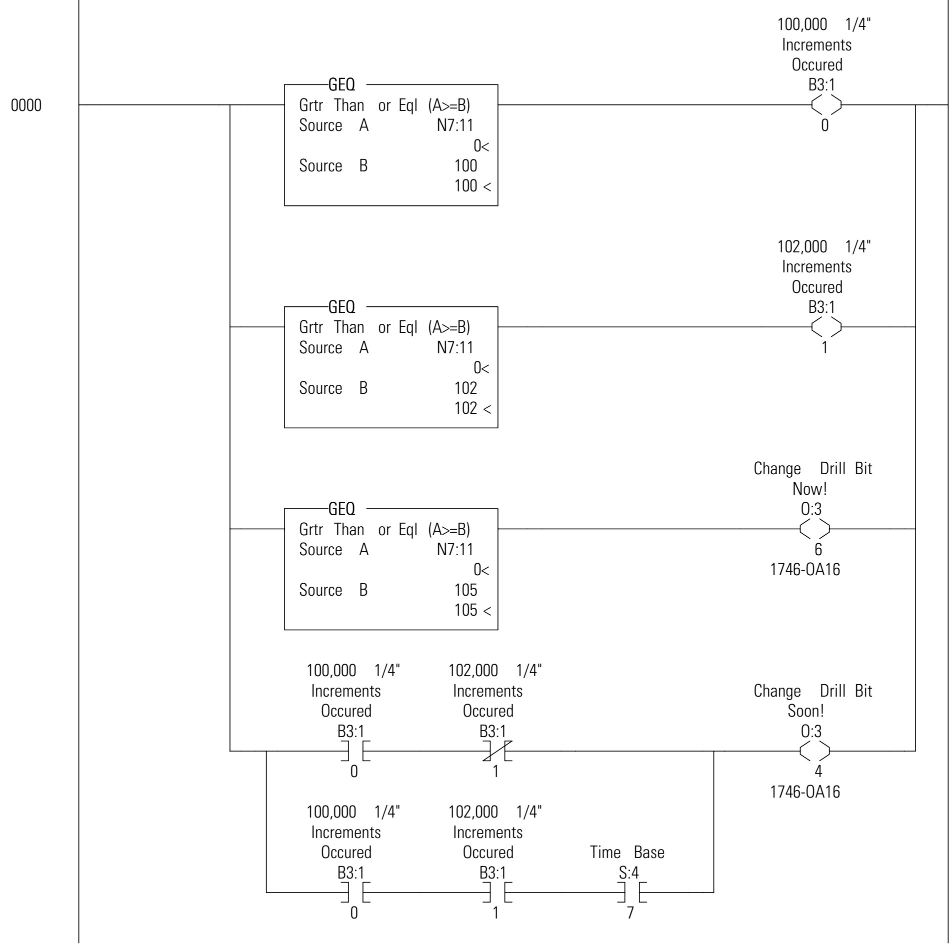


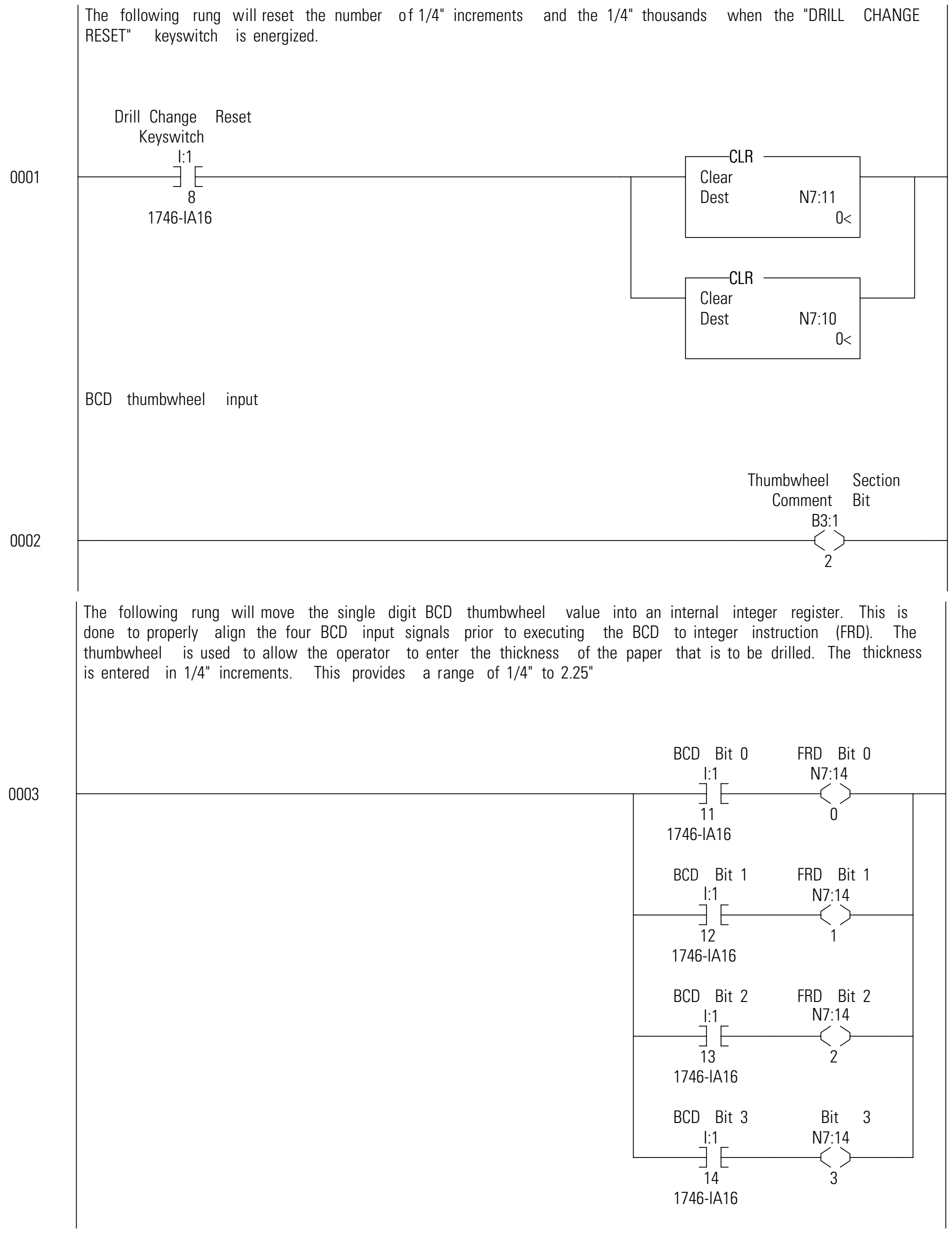

The following rung stops the conveyor and signals the main program (file 2) to initialize a drilling sequence. The DRILL SEQUENCE subroutine (program file 6) resets the drill sequence start bit and sets the conveyor drive bit (O:3/0) upon completion of the drilling sequence.
0006
Drill Sequence
Start
B3:2
L
0
Conveyor
Start/Stop
O:3
U
0
1746-OA16
0007
END

Drill Calculation and Warning

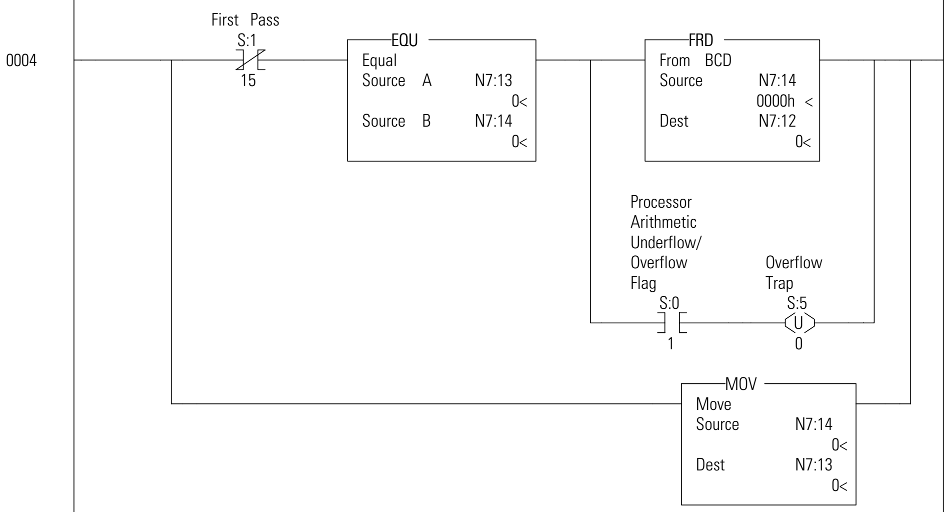
The program tracks the number of holes drilled and the number of inches of material that have been drilled through using a thumbwheel. The thumbwheel is set to the thickness of the book per 1/4 inch. (If the book is 1 1/2 inches thick, the operator would set the thumbwheel to 6.) When 25,000 inches have been drilled, the Change Drill Soon pilot light turns on. When 25,500 inches have been drilled, the Change Drill Soon pilot light flashes. When 26,000 inches have been drilled, the Change Drill Now pilot light turns on and the machine turns off. The operator changes drill bits and then resets the internal drill wear counter by turning the Drill Change Reset keyswitch.

The following rung examines the number of 1/4" thousands that have accumulated over the life of the current drill bit. If the bit has drilled between 100,000 and 101,999 1/4" increments of paper, then the "CHANGE DRILL" light will illuminate steady. When the value is between 102,000 and 103,999, then the "CHANGE DRILL" light will flash at a 1.28 second rate. When the value reaches 105,000, then the "CHANGE DRILL" light will flash, and the "CHAGE DRILL NOW" light will illuminate.




The following rung will reset the number of 1/4" increments and the 1/4" thousands when the "DRILL CHANGE RESET" keyswitch is energized.
0001
Drill Change Reset
Keyswitch
I:1
8
1746-IA16
CLR
Clear
Dest N7:11
0<
CLR
Clear
Dest N7:10
0<
BCD thumbwheel input
0002
Thumbwheel Section
Comment Bit
B3:1
2
The following rung will move the single digit BCD thumbwheel value into an internal integer register. This is done to properly align the four BCD input signals prior to executing the BCD to integer instruction (FRD). The thumbwheel is used to allow the operator to enter the thickness of the paper that is to be drilled. The thickness is entered in 1/4" increments. This provides a range of 1/4" to 2.25"
0003
BCD Bit 0
I:1
11
1746-IA16
FRD Bit 0
N7:14
0
BCD Bit 1
I:1
12
1746-IA16
FRD Bit 1
N7:14
1
BCD Bit 2
I:1
13
1746-IA16
FRD Bit 2
N7:14
2
BCD Bit 3
I:1
14
1746-IA16
Bit 3
N7:14
3

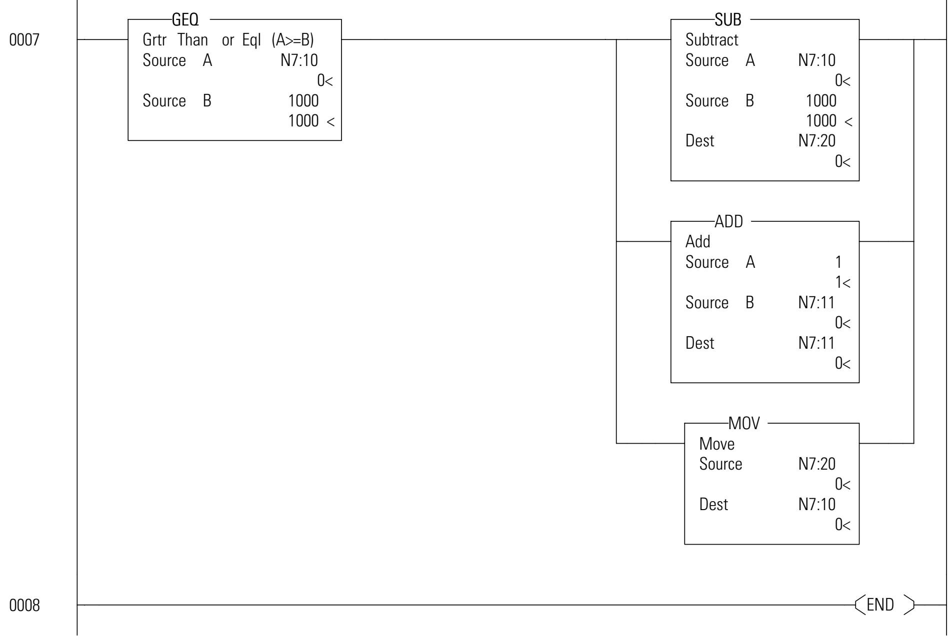
The following rung will convert the BCD thumbwheel value from BCD to integer. This is done because the processor operates upon integer values. This rung also "debounces" the thumbwheel to ensure that conversion only occurs on valid BCD values. Note that invalid BCD values can occur while the operator is changing the BCD thumbwheel. This is due to input filter propagation delay differences between the 4 input circuits that provide the BCD input value.



The following rung will ensure that the operator cannot select a paper thickness of 0. If this were allowed, the drill bit life calculation could be defeated resulting in poor quality holes due to a dull drill bit. Therefore the minimum paper thickness that will be used to calculate drill bit wear is ¼".

The following rung will keep a running total of how many inches of paper have been drilled with the current drill bit. Every time a hole is drilled, add the thickness (in ¼"s) to the running total (kept in ¼"s). The same OSR is necessary because the ADD will execute every scan that the rung is true, and the drill body would actuate the DRILL DEPTH limit switch for more than 1 program scan. Integer N7:12 is the integer-converted value of the BCD thumbwheel on inputs I:3/11 - I:3/14.

The following rung will keep track of the number of counts that are past 1,000.



Time Driven Sequencer Application Example

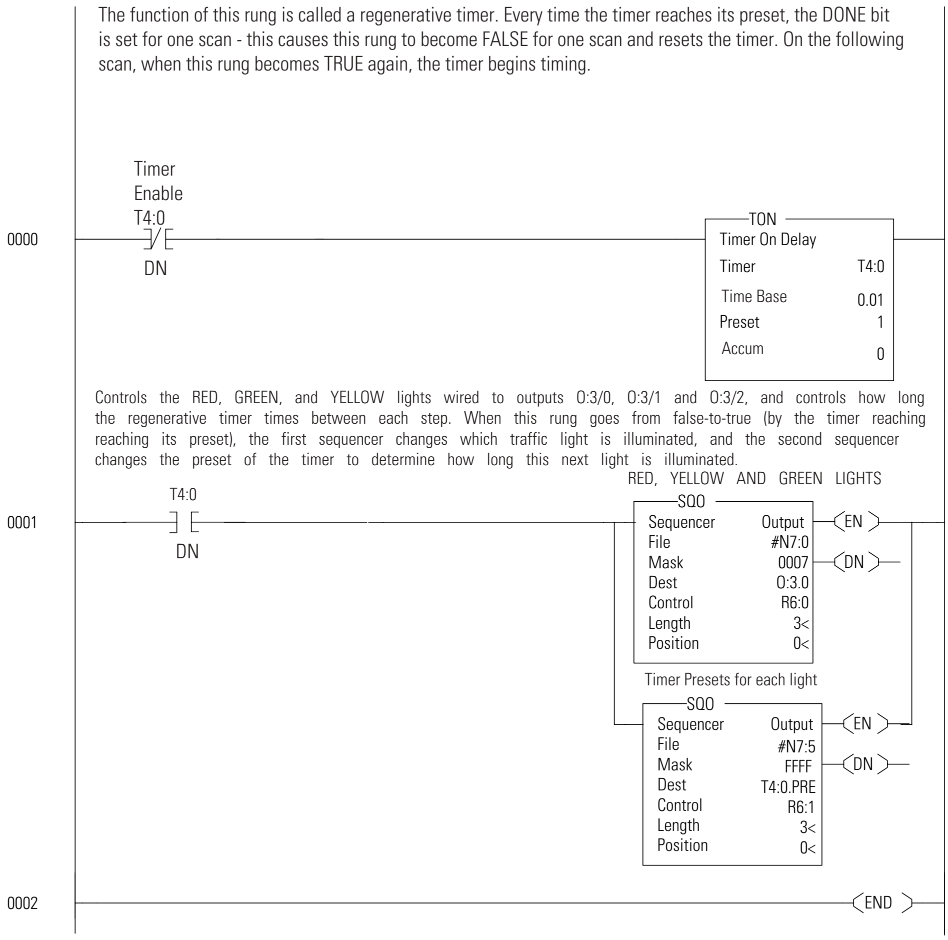
The following application example illustrates the use of the TON and SQO instructions in a traffic signal at an intersection. The timing requirements are:

- Red light - 30 seconds.
- Yellow light - 15 seconds.
- Green light - 60 seconds.

The timer, when it reaches its preset, steps the sequencer that in turn controls which traffic signal is illuminated. For a detailed explanation of:

- XIC, XIO, and TON instructions, see chapter 2.
- SQO and SQC instructions, see chapter 7.

Time Driven Sequencer Ladder Program

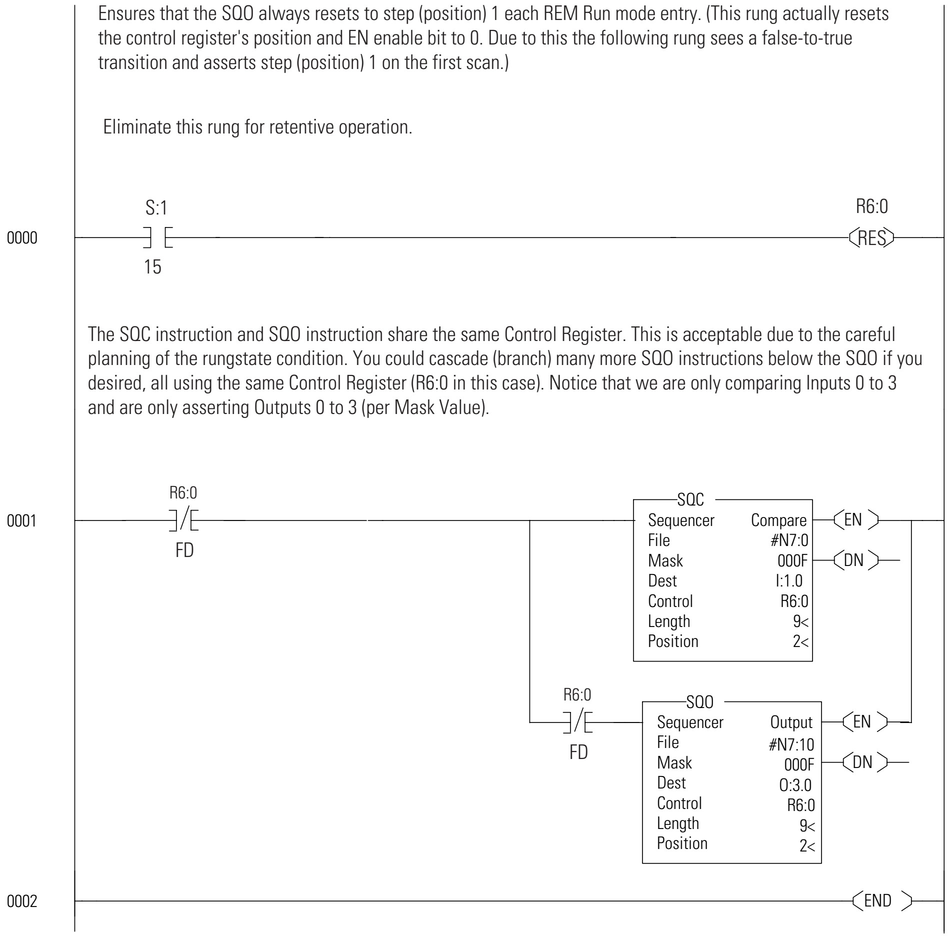


Event Driven Sequencer Application Example

The following application example illustrates how the FD (found) bit on an SQC instruction can be used to advance as SQO to the next step (position). This application program is used when a specific order of events is required to occur repeatedly. By using this combination, you can eliminate using the XIO, XIC, and other instructions. For a detailed explanation of:

- XIC, XIO, and RES instructions, see chapter 2.
- SQO and SQC instructions, see chapter 7.

Event Driven Sequencer Ladder Program

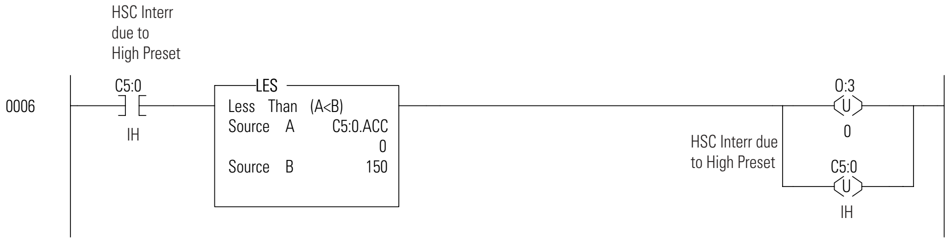


The following table displays the FILE DATA for both sequencers. The SQC compare data starts at N7:9. While the SQO output data starts at N7:10 and ends at N7:19. Please note that the step 0 of the SQO is never active. The reset rung combined with the rung logic of the sequencers guarantees that the sequencers always start at step 1. Both sequencers also roll over to step 1. Roll Over to step 1 is integral to all sequencer instructions.

Table G.1 SQC Compare Data

Addresses	Data (Radix = Decimal)									
N7:0	0	1	2	3	4	5	6	7	8	9
N7:10	0	0	1	2	3	4	5	6	7	8

If the high-speed counter reached its high preset of 350 (indicates that the holding area reached maximum capacity), it would energize O:0/0, shutting down the filling operation. Before re-starting the filler, allow the packer to empty the holding area until it is about 1/3 full.

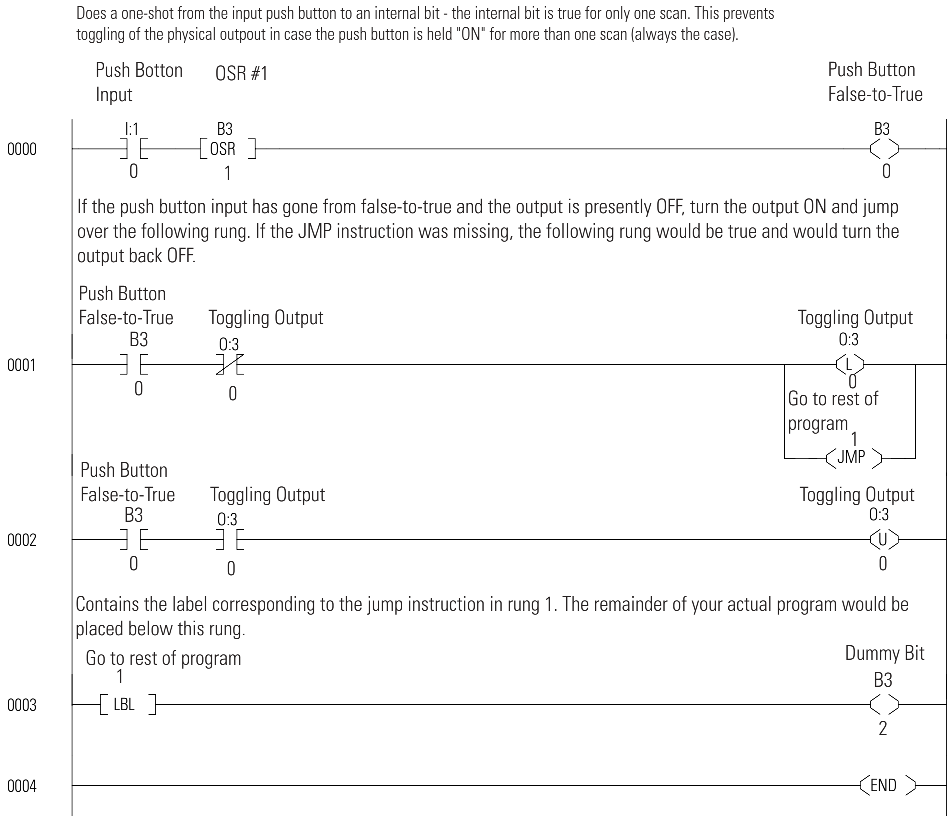


On/Off Circuit Application Example

The following application example illustrates how to use an input to toggle an output either on or off. For a detailed explanation of:

- XIC, XIO, OTE, OTU, OTL, and OSR instructions, see chapter 2.
- JMP and LBL instructions, see chapter 6.

On/Off Circuit Ladder Program



Interfacing with Enhanced Bar Code Decoders Over DH-485 Network Using the MSG Instruction

The purpose of this section is to illustrate how to interface Allen-Bradley Enhanced Bar Code Decoders to SLC 5/03 and higher processors via the DH-485 network. Enhanced Bar Code Decoders act only as slave devices on this network. This means that these decoders cannot initiate the transfer of data to a host device, such as the SLC 5/03 (or higher) processor on DH-485. The SLC processor must initiate commands to a decoder and poll that decoder for the reply to those commands.

Processor and Decoder Operation

The Enhanced Bar Code Decoder (catalog number 2755-DS/DD, Series B), when used as a node on a DH-485 network can act as a slave only. This means that the decoder may not initiate communications to any other node on the network. Therefore, in order for a device to get bar code data from an Enhanced Bar Code Decoder on a DH-485 network, that device must send a read command and then poll the decoder for the reply with data.

The only devices capable of polling a slave device on DH-485 are the SLC 5/03 and higher processors. For the SLC 5/03 processors (1747-OS302, FRN10 or later), polling can be done via channels 0 and 1. For the SLC 5/04 processors (1747-OS401, FRN7 or later), channel 0 supports this capability.

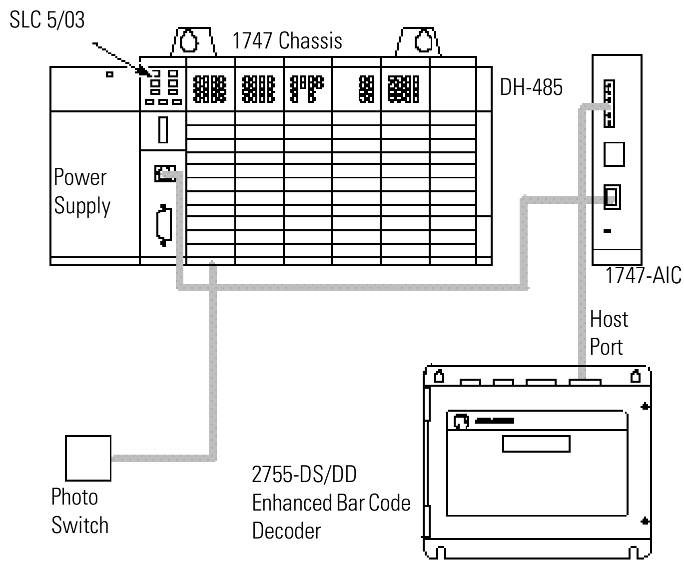
There are many ways to trigger bar code decoders to read a bar code label when a label is present.

- A package detect switch wired to both an SLC input module and the bar code decoder
- A package detect switch wired only to an SLC input and an SLC output then used to trigger the decoder
- Via a software trigger command from the SLC processor

For this example, the software trigger is used. However, the basic principal is the same for all trigger modes.

System Set Up

In this example, a photo switch is located such that when it detects a product is in position for the bar code scanner to read a bar code label on the product, a discrete input to the SLC 5/03 processor is energized.



The 5/03 ladder program then initiates a MSG Write to the decoder to trigger the decoder to start scanning for a valid bar code label. When the decoder is scanning for a valid bar code label, it operates as shown below.

Table G.2 Result of Scanning a Valid Bar code

Result of Scan	Bar Code Decoder Response	Processor Response
Good Read	turns on its "Good Read" onboard output wired to the SLC processor	When one of these two inputs to the SLC are turned on, the SLC will initiate a "MSG Read" to the decoder to get the label data or no-read message data.
No-Read	turns on its "No-Read" onboard output wired to the SLC processor	

In this case, the good read output is turned on as soon as a valid read occurs, and the no-read output is turned on after the decoder has attempted to read a label for a specified amount of time and could not.

The amount of time the decoder attempts to read a label is variable and is called the "No-Read Timer". For this example it is assumed that the product is moving by the scanner and if the label is not read in 2 seconds, it is not read at all. Therefore, the "No-Read Timer" parameter in the bar code decoder is set to 2 seconds. Refer to the DS/DD Series Enhanced Bar Code Decoders (Bulletin 2755) User's Manual, publication 2755-833, for details concerning the configuration of your Allen-Bradley Enhanced Bar Code Decoder.

Operating Sequence

With the bar code decoder configured as previously described, the following series of event take place when a product with a good bar code label breaks the photo switch and this input to the SLC goes from false-to-true. The SLC 5/03 ladder program logic to make it happen is also included. Please note that, as previously stated, a bar code/SLC system may be configured in a variety of ways.

Messages sent by the SLC processor to the Enhanced Bar Code Decoder must be programmed as shown by the example ladder program on page H-29. If this logic is not followed, the communication between the two DH-485 devices could become out of sequence, resulting in no data transfers between the decode and the SLC processor. To correct such a problem, cycle power to the decoder.

Sequence of Events

The photo switch input to the SLC goes from false-to-true

The SLC processors send a "trigger" command to the decoder via a "MSG Write" command.

The decoder immediately replies to the SLC that it has properly received the command.

The reply sets the MSG DN bit and clears the MSG DA bit in the SLC processor.

The SLC processor then waits a specified amount of time and re-enables the same MSG instruction which now sends the "poll" to the decoder.

The decoder then responds to this "poll" with its reply to the "trigger" command. The SLC processor then sets the MSG DN and MSG DA bits.

The decoder then reads the bar code label and turns on the "good read" output wired to the SLC processor.

When this "good read" input to the SLC goes from false-to-true, the SLC sends a "MSG Read" to the decoder.

The decoder immediately replies to the SLC that it has properly received the command.

The reply sets the MSG DN bit and clears the MSG DA bit in the SLC processor.

The decoder then processes the command and formulates a reply to the "MSG Read" command with the decoded label data included in this reply, and places into its output buffer awaiting a "poll".

The SLC processor waits a specified amount of time, then re-enables the same "MSG Read" instruction which now sends the "poll" to the decoder.

The decoder then responds to the "poll" command with its reply to the "MSG Read" command which includes the bar code label data.

The SLC processor then sets the MSG DN and MSG DA bits to signify its receipt of the data.

TIP These events are described in more detail by the comments listed within the example ladder program page G-21.

Optimizing MSG Time-out

If the time delay between sending a command to an Enhanced Bar Code Decoder and polling for the reply is not long enough, the MSG instruction will time-out (MSG TO bit = 1) each time it is enabled from that point forward. To re-synchronize the SLC processor and the decoder, you need to cycle power on the decoder to clear its buffer.

There are ways of clearing the buffers in the decoder, such as sending a Clear Buffers command or a Reset command to the decoder. However, the best way to handle this issue is to never let it happen. Optimizing the time delay between sending the initial command and polling for the reply is the best way to accomplish this. The delay must be long enough so the decoder has enough time to formulate a reply to the command and short enough to not impact the throughput of the application.

Example MSG Instruction Configuration

The example SLC 5/03 and SLC 5/04 ladder program demonstrates how to send commands to an Enhanced Bar Code Decoder, and then after a time delay, pol for a reply. The internal set up screen parameters for the two MSG instructions in the example ladder program are shown below, along with the necessary Enhanced Bar Code Decoder configuration parameters.

Table G.3 Message Configuration

	MSG #1	MSG #2
Type	peer-to-peer	peer-to-peer
Read/write	write	read
Target device	485CIF	485CIF
Local/remote	local	local
Control block	N7:0	N7:20
Channel	1	1
Target node	2	2
Our source file address	N7:15	N7:40
Target CIF offset(1)	0	0
Message length in elements	1	10
Message time-out (seconds)	5	5

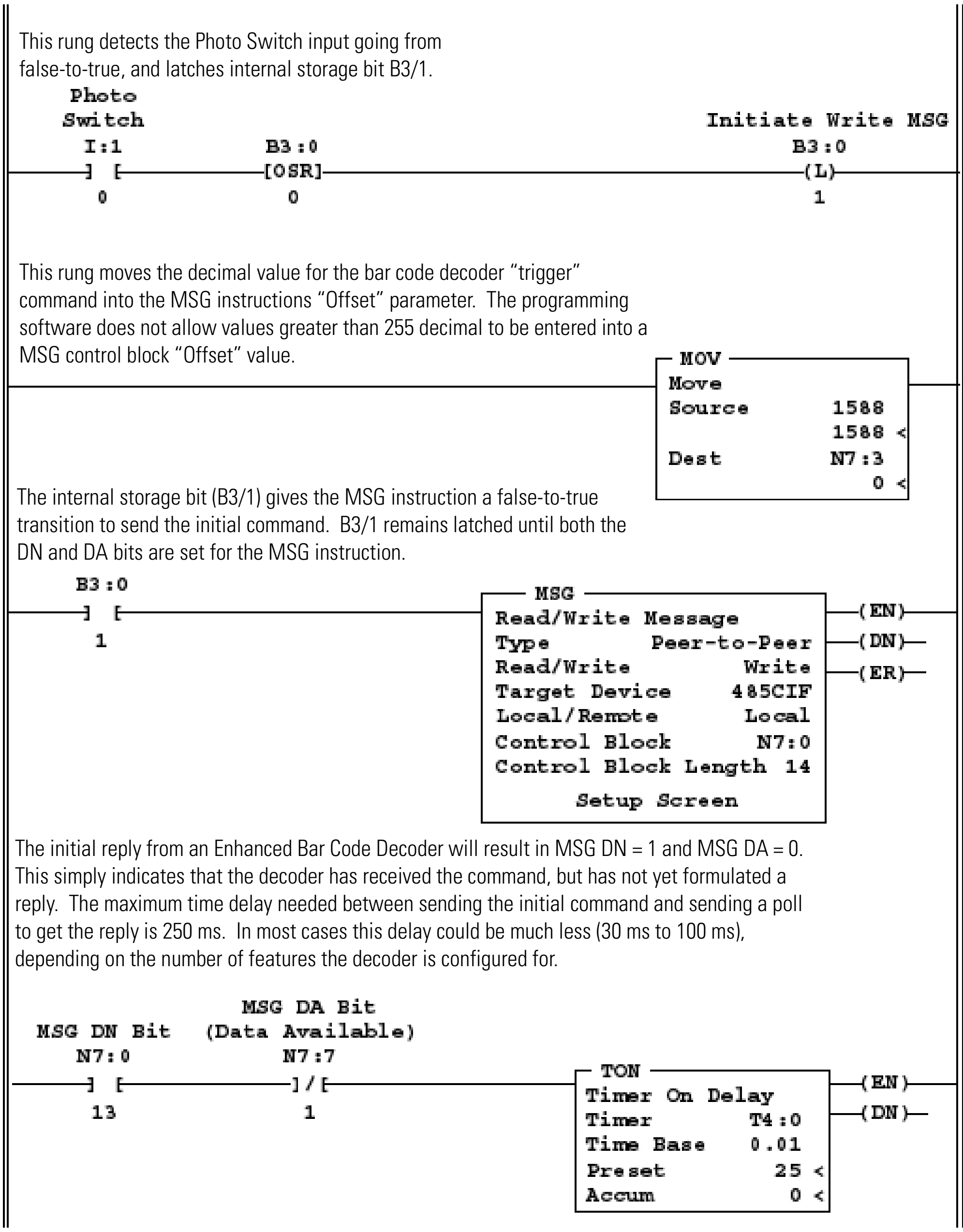
(1) The Target CIF Offset when working with Enhanced Bar Code Decoders as slaves on DH-485 must contain a value greater than 255. However, 255 is the largest value SLC programming software allows you to enter into this parameter in a MSG instruction. Therefore, use an unconditioned rung with a MOV instruction to move the proper value into the Target CIF Offset field. The example ladder program in this section demonstrates this. Note that 1586 decimal in a “MSG Write” is the value which results in a properly configured Enhanced Bar Code Decoder to initiate the ”trigger” function. A value of 256 in a “MSG Read” requests a specified number of words of data from the bar code decoder. In this example, we are reading 10 words or 20 characters (bytes).

Example Scanner and Decoder Configuration

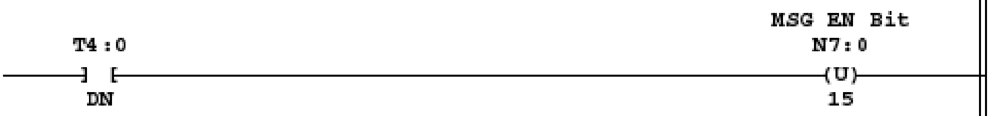
Table G.4 Scanner and Decoder Configuration

Scanner Configuration Parameters **Scanner Control Page**		**2755-DS/DD Series B Enhanced Bar Code Decoder Configuration Parameters** **Host Configurations Page**	
Discrete I/O:	Read Package 25 ms No-Read Package 25 ms	Baud Rate:	19200
Laser Light:	Triggered	Bits/Char:	8 Data 1 Stop
Decode Mode:	Host	Parity:	Even
No-Read Time:	2000 ms	Host Protocol:	DH-485 PCCC-1
Inter Scan Time:	none	Device Address:	2
Capture Count:	2	ACK Char:	none
Symbols/Scan:	1	NAK Char:	none
Symbols/Package:	1	Large Buffer:	No
Match Complete:	1	Send Host Message: Package	Immediately after Valid
		Transmission Check:	none

Example Ladder Program



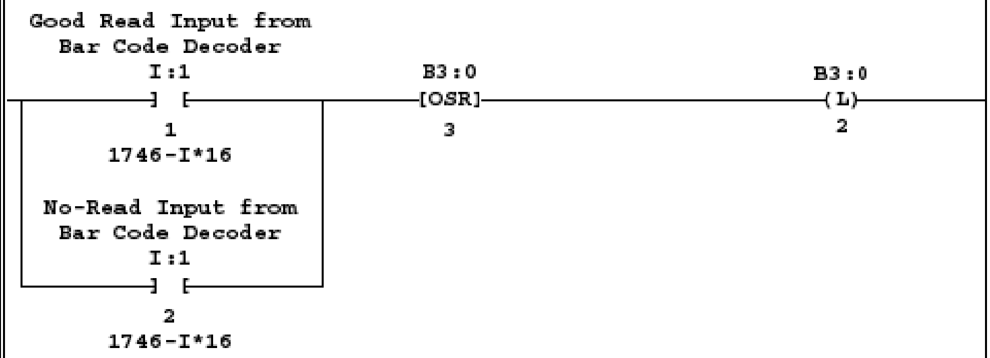
The internal storage bit, B3/1, holds the MSG instruction true until DN and DA are both set, indicating completion of the command sent and reply received sequence. When DN is set and DA is reset, unlatching the MSG EN bit effectively toggles the MSG instruction the same as if the MSG rung were toggled, i.e. rung conditions made false, then true. The MSG instruction is toggled one time after DN and NOT DA plus some time delay, to send a final poll to the decoder to get the MSG reply. When the reply is received, the SLC processor sets DN and DA.



When the SLC processor sets both DN and DA for a MSG instruction, the MSG sequence to an Enhanced Bar Code Decoder is complete. In this case, the decoder has received the “trigger” command and has performed this command. Therefore, unlatch B3/1 at this time to be ready for the next request for “trigger”.

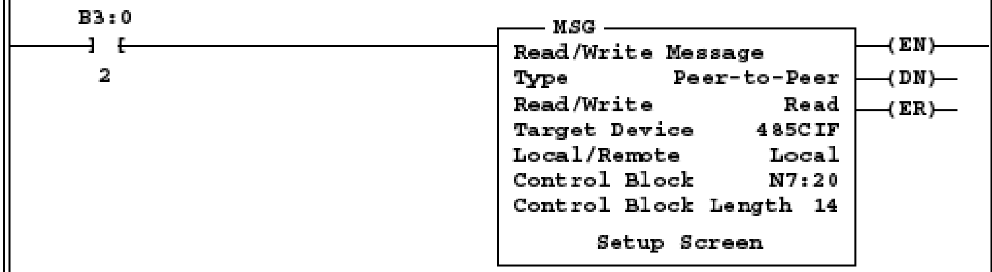
If an error occurs with the MSG instruction, the ER bit is set. If this occurs, the user can either try to resend the same message again by unlatching EN, or at this point, could sound an alarm or route the product down a rework loop or some other similar action. If the latter choice is used, you must also unlatch B3/1 at this time to be ready for the next request for “trigger”.

This rung detects false-to-true transitions of either a good read or a no-read input from the bar code decoder and latches internal storage bit B3/2. B3/2 then, in the next rung, initializes the MSG read command to the decoder. This is done if either a good read or a no-read occurs, because the no-read message configured in the decoder is data as much as actual bar code label data. Therefore, the ladder program must distinguish between data that means no-read as well as actual bar code label data.

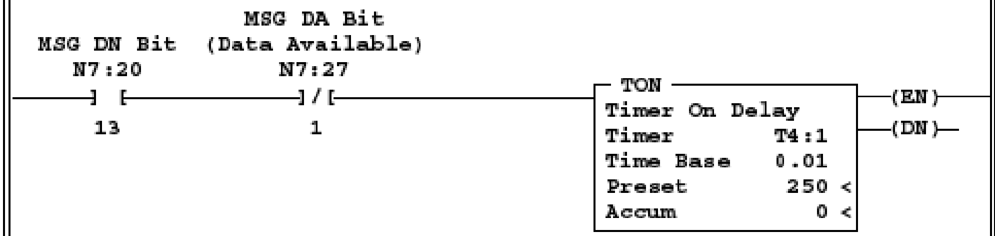


This rung moves the decimal value for the bar code decoder “Read” command into the MSG instruction’s “Offset” parameter. The programming software does not allow values greater than 255 decimal to be entered into a MSG control block “Offset” value.

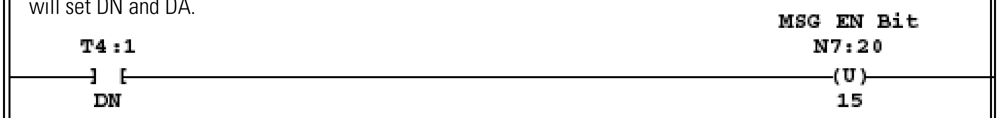
The internal storage bit, B3/2, gives the MSG instruction a false-to-true transition to send the initial command.



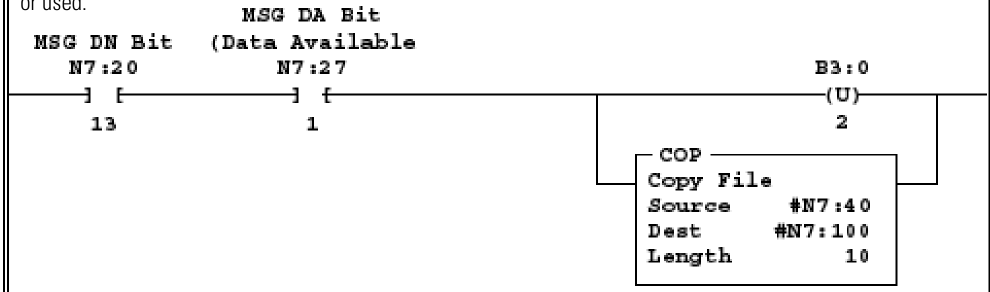
The initial reply from an Enhanced Bar Code Decoder will result in MSG DN = 1 and MSG DA = 0. This indicates that the decoder has received the command, but has not yet formulated a reply. The maximum time delay needed between sending the initial command and sending a poll to get the reply is 250 ms. In most cases this delay could be much less (30 ms to 100 ms), depending on the number of features the decoder is configured for.



The internal storage bit, B3/2, holds the MSG instruction true until DN and DA are both set, indicating completion of the command sent and reply received sequence. When DN is set and DA is reset, unlatching the MSG EN bit effectively toggles the MSG instruction the same as if the MSG rung were toggled, i.e. rung conditions made false, then true. The MSG instruction is toggled one time after DN and NOT DA plus some time delay, to send a final poll to the decoder to get the MSG reply. When the reply is received, the SLC processor will set DN and DA.



When the SLC processor sets both DN and DA for a MSG instruction, the MSG sequence to an Enhanced BAr Code Decoder is complete. In this case, the decoder has received the “Read” command and has formulated a reply to this command. Therefore, unlatch B3/2 at this time to be ready for the next “REad” request. In addition, when DN and DA are both set, this indicates that the data received with the read reply (except “no-read” data) is valid and may be buffered or used.



If an error occurs with the MSG instruction, the ER bit is set. If this occurs, the user can either try to resend the same message again by unlatching EN, or at this point, could sound an alarm or route the product down a rework loop or some other similar action. If the latter choice is used, you must also unlatch B3/2 at this time to be ready for the next “Read” request.