System Functions

In This Appendix...

System Functions

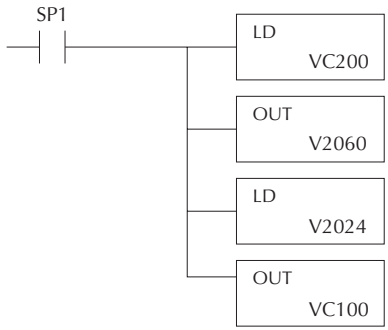
System Functions Commands are primarily used to read from and write to the CTRIO(2) module's internal registers. ☑ DL

The CTRIO(2) module's internal current count register can be read from or written to if the input is configured for a Counter or Quadrature Counter. Timer values are not accessible. ☑ Win ☑ NI

The CTRIO(2) module's internal current output pulse count can be read from or written to only if the pulse output is running Dynamic Velocity or Dynamic Positioning profiles.

Steps	Name	PLC Control Outputs Base Addr = V2030 (Bit-of-Word)	PLC Status Inputs Base Addr = V2000 (Bit-of-Word)	PLC Control Outputs Base Addr = V2030 (Control Relay) D2-240	PLC Status Inputs Base Addr = V2000 (Control Relay) D2-240	Action
1	Command Code	User Specified to use with RD/WT Instruction		User Specified to use with RD/WT Instruction		1 Hex: Read All Registers 2 Hex: Write All Registers 4 Hex: Write One Register 5 Hex: Write Reset Value
2	System Command Error		V2024.6		C106	ON if Command or Parameters are invalid
3	System Command Complete		V2024.7		C107	When ON, command has been accepted, clear Process Command bit
6	Process Command	V2060.7		C207		Turn ON Command Complete status bit is returned

***NOTE 1**:The D2-240 CPU does not support bit-of-word addressing. The status and control bits must be mapped to control relay words. An example of mapping code is shown on right.*



***NOTE 2:** For example, **Direct**SOFT uses B2020.1 in the ladder code to indicate that you are addressing the second bit of V-memory register V2020. The "B" prefix indicates bit-of-word addressing.*

Write All Registers (IBoxes)

PARAMETERS:

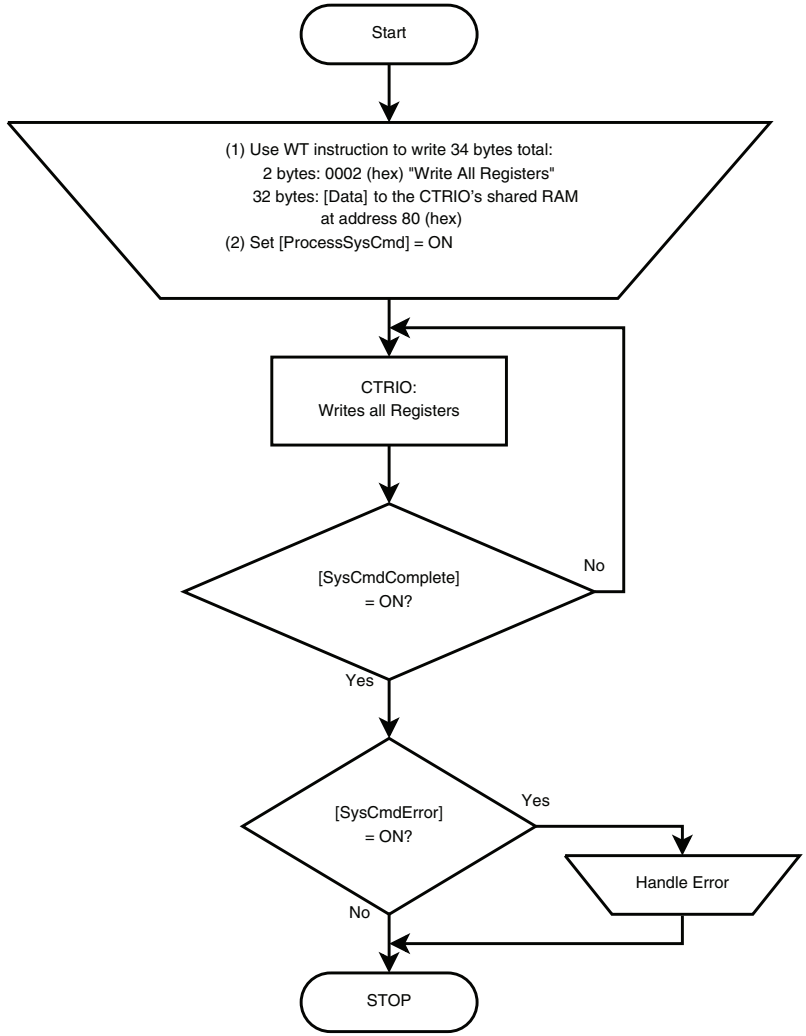
(1) [ProcessSysCmd]
n+30.7

(2) [SysCmdComplete]
n+24.7

(3) [SysCmdError]
n+24.6

KEY:

[Data]:
Bytes 1-4: Ch1/Fn1
Bytes 5-8: Ch1/Fn2
Bytes 9-12: Ch2/Fn1
Bytes 13-16: Ch2/Fn2
Bytes 17-20: Output0
Bytes 21-24: Output1
Bytes 25-28: Output2
Bytes 29-32: Output3



Write All Registers (DL-PLC)

PARAMETERS:

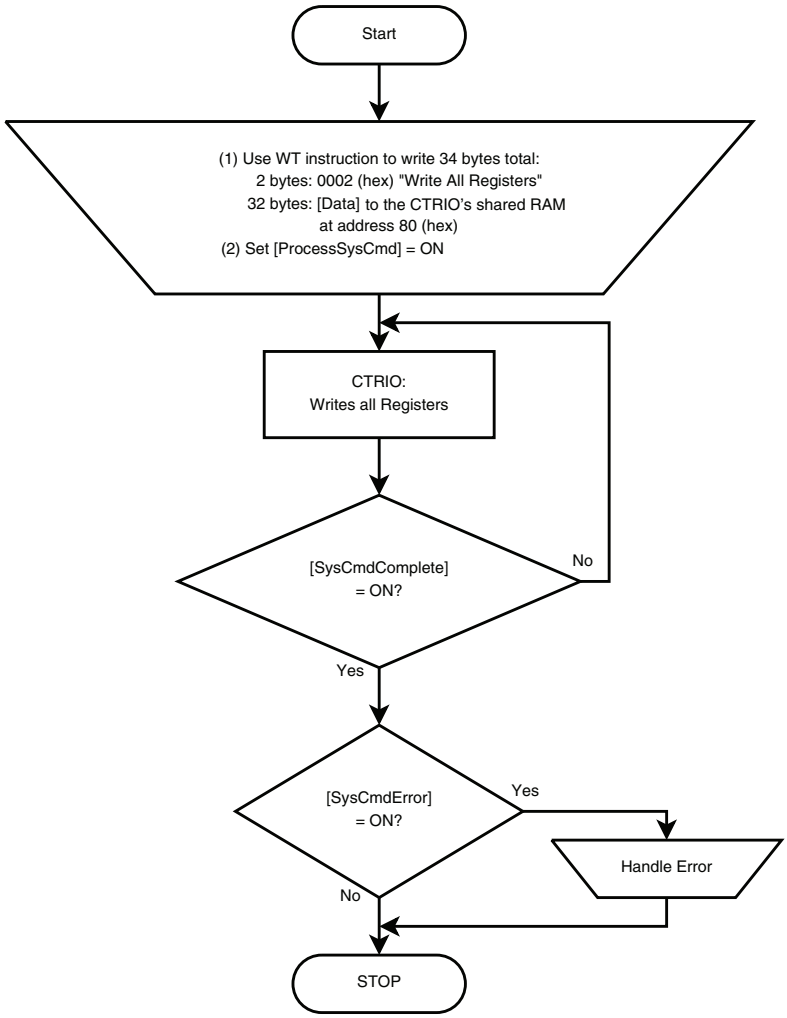
(1) [ProcessSysCmd]
n+30.7

(2) [SysCmdComplete]
n+24.7

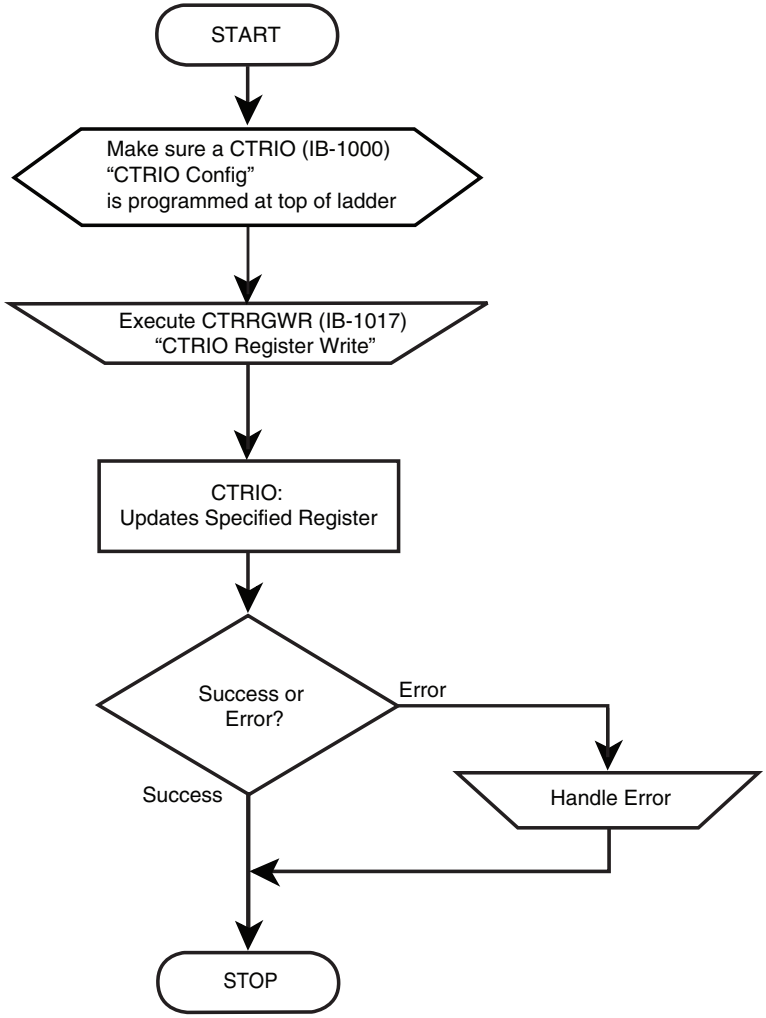
(3) [SysCmdError]
n+24.6

KEY:

[Data]:
Bytes 1-4: Ch1/Fn1
Bytes 5-8: Ch1/Fn2
Bytes 9-12: Ch2/Fn1
Bytes 13-16: Ch2/Fn2
Bytes 17-20: Output0
Bytes 21-24: Output1
Bytes 25-28: Output2
Bytes 29-32: Output3



Write One Register (IBoxes)



Write One Register (DL-PLC)

PARAMETERS:

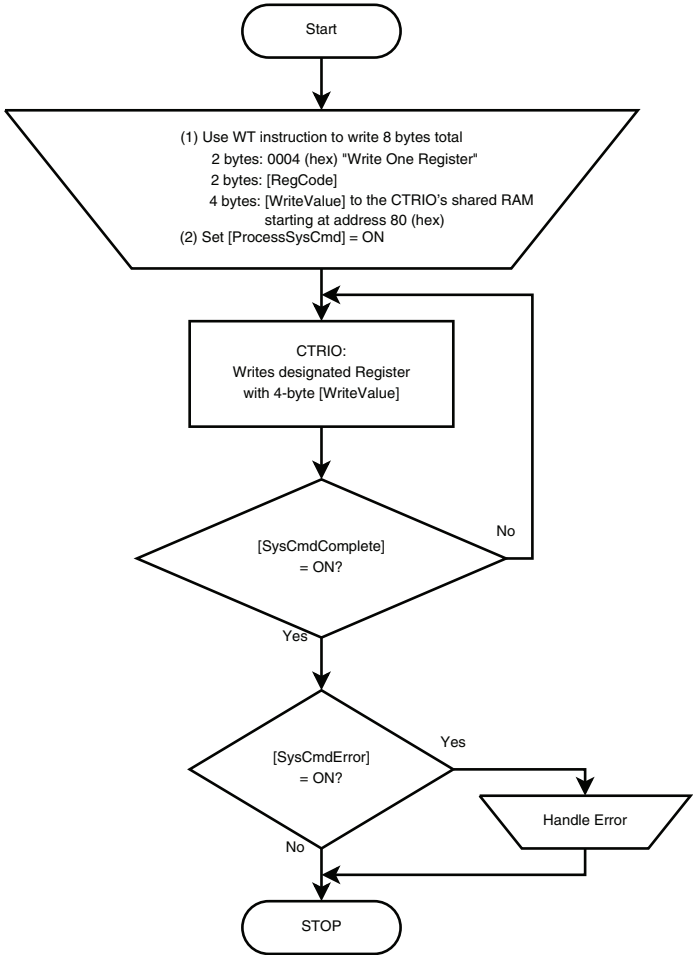
(1) [ProcessSysCmd]
n+30.7

(2) [SysCmdComplete]
n+24.7

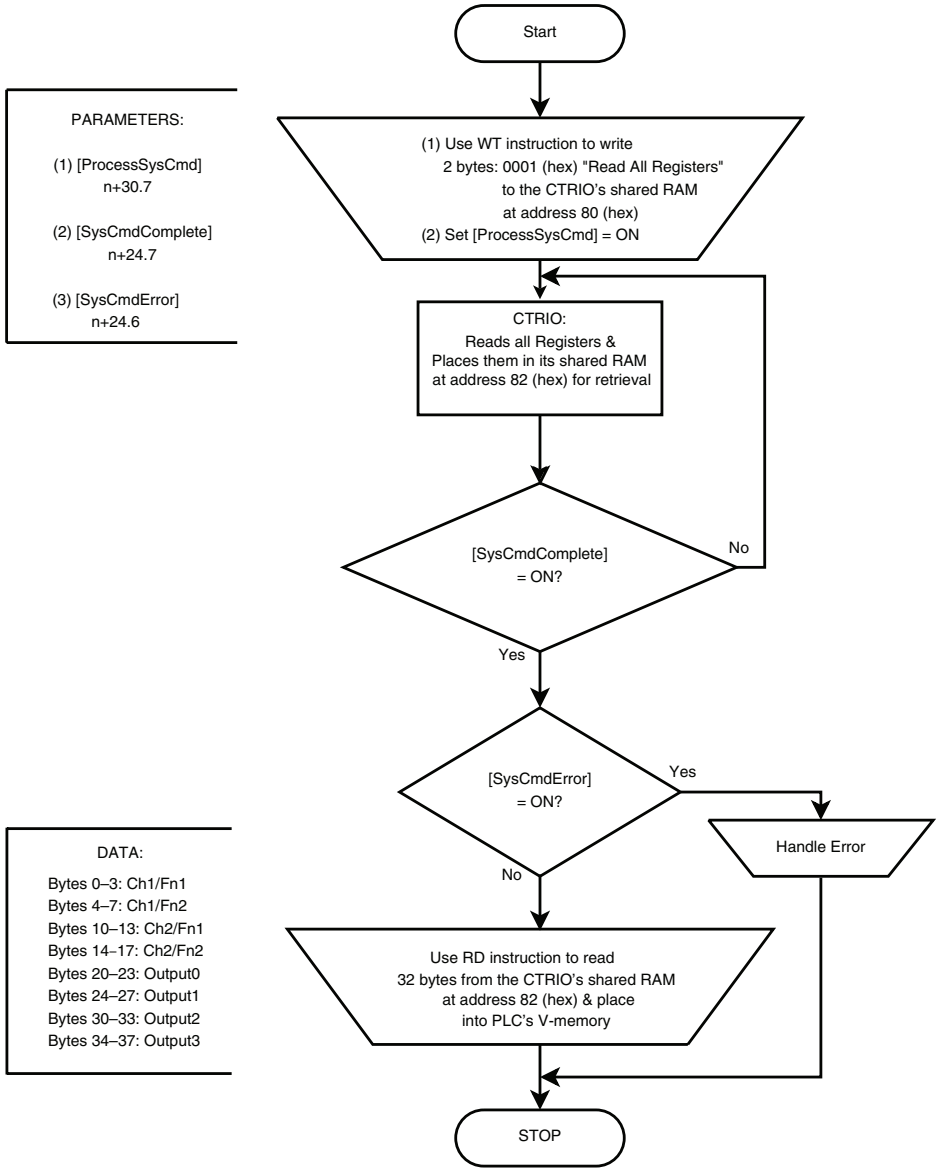
(3) [SysCmdError]
n+24.6

KEY:

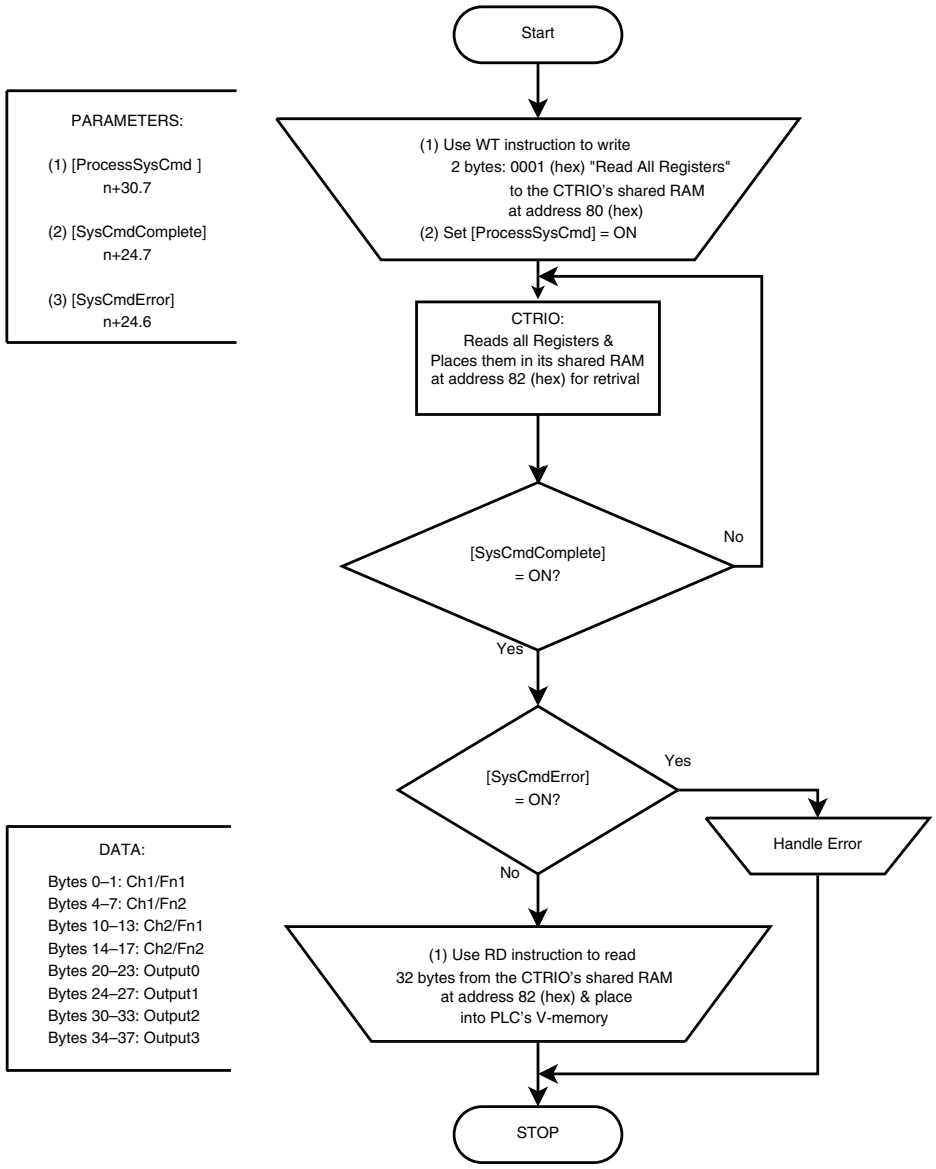
[RegCode]
(hex value):
0000 = Ch1/Fn1
0001 = Ch1/Fn2
0002 = Ch2/Fn1
0003 = Ch2/Fn2
0004 = Output0
0005 = Output1
0006 = Output2
0007 = Output3
0008 = Ch1/Fn1 Reset
0009 = Ch1/Fn2 Reset
000A = Ch2/Fn1 Reset
000B = Ch2/Fn2 Reset
000C = Ch1A Filter
000D = Ch1B Filter
000E = Ch1C Filter
000F = Ch1D Filter
0010 = Ch2A Filter
0011 = Ch2B Filter
0012 = Ch2C Filter
0013 = Ch2D Filter



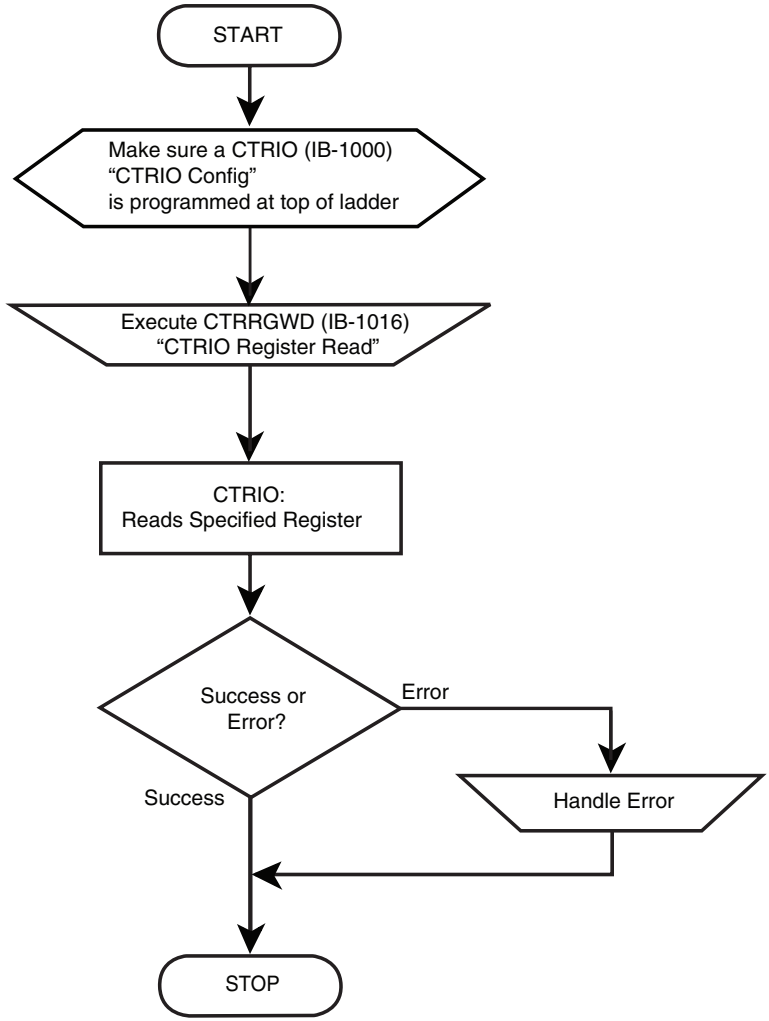
Read All Registers (IBoxes)



Read All Registers (DL-PLC)



Read One Register (IBoxes)



Read One Register (DL-PLC)

PARAMETERS:

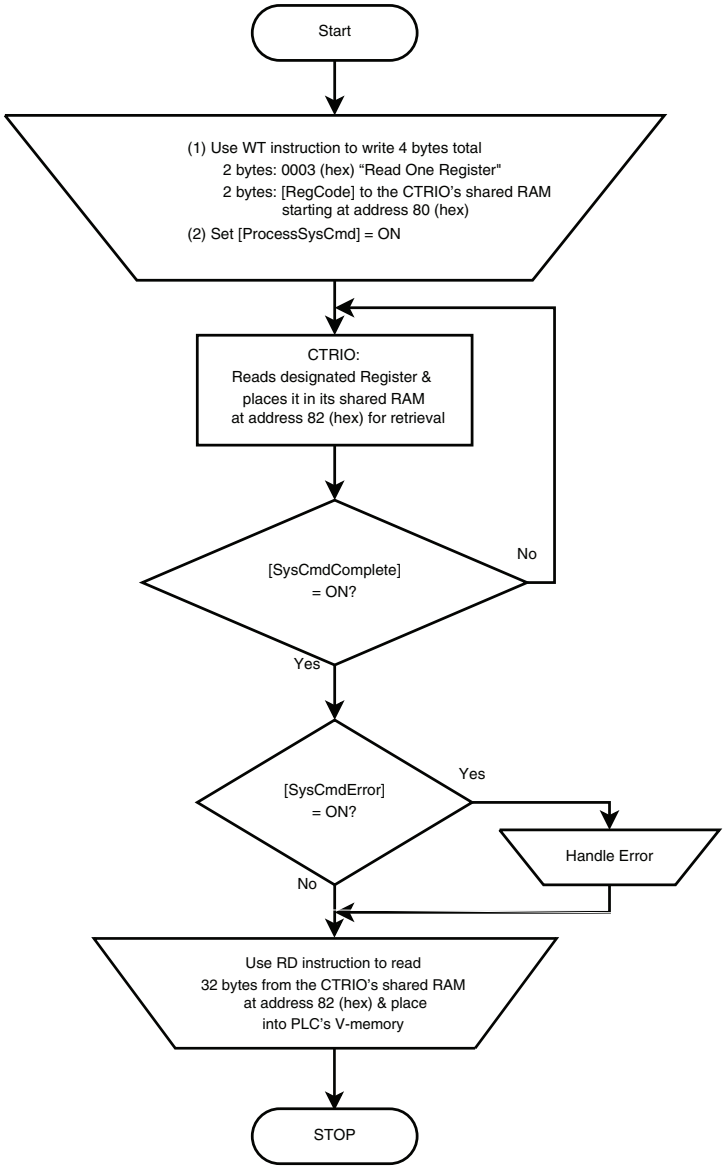
(1) [ProcessSysCmd]
n+30.7

(2) [SysCmdComplete]
n+24.7

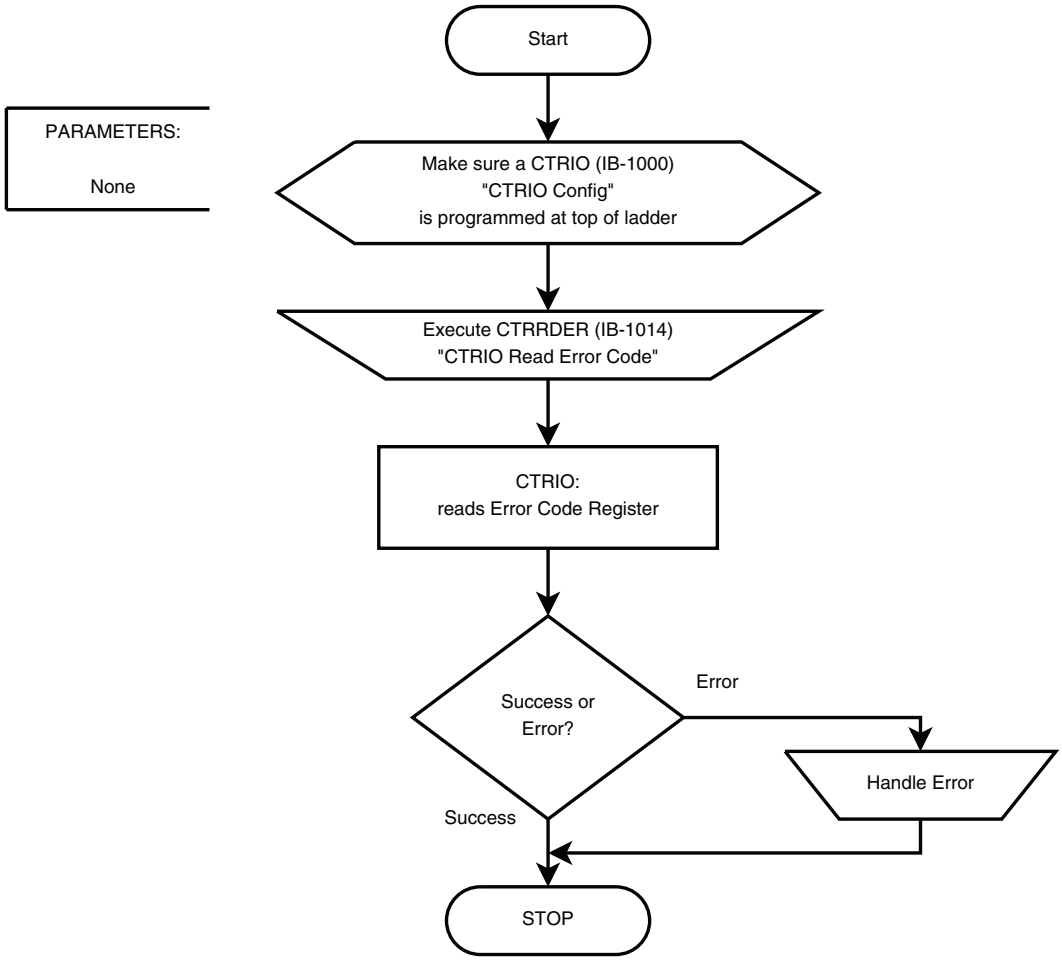
(3) [SysCmdError]
n+24.6

KEY:

[RegCode]
(hex value):
0000 = Ch1/Fn1
0001 = Ch1/Fn2
0002 = Ch2/Fn1
0003 = Ch2/Fn2
0004 = Output0
0005 = Output1
0006 = Output2
0007 = Output3
0008 = Ch1/Fn1 Reset
0009 = Ch1/Fn2 Reset
000A = Ch2/Fn1 Reset
000B = Ch2/Fn2 Reset
000C = Ch1A Filter
000D = Ch1B Filter
000E = Ch1C Filter
000F = Ch1D Filter
0010 = Ch2A Filter
0011 = Ch2B Filter
0012 = Ch2C Filter
0013 = Ch2D Filter



Read Error Code (IBoxes)



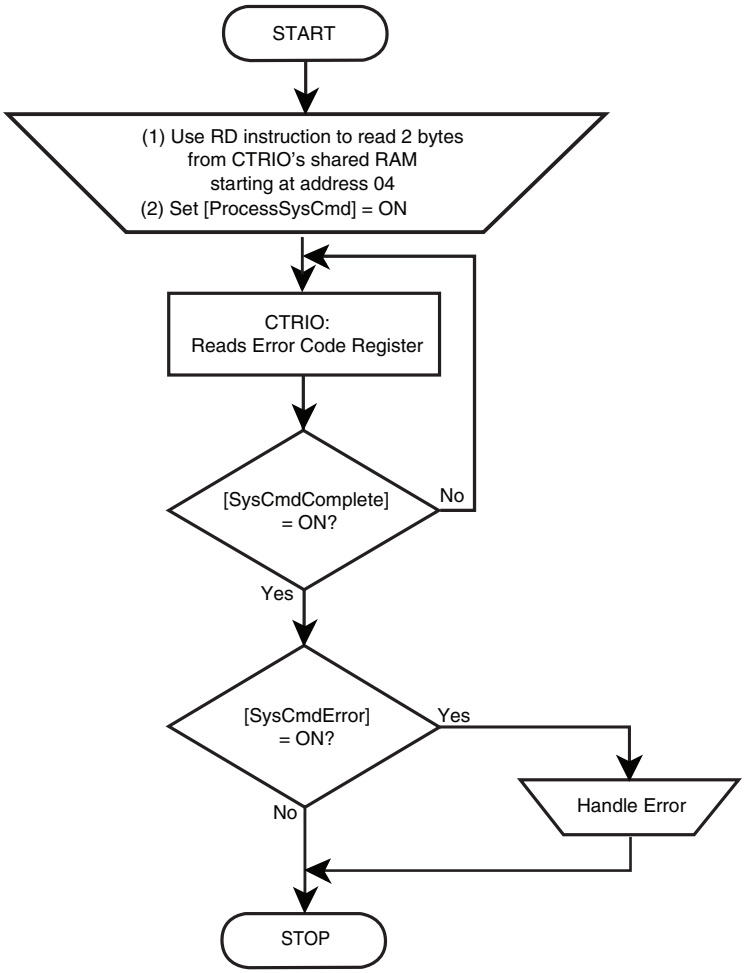
Read Error Code (DL-PLC)

PARAMETERS:

(1) [ProcessSysCmd]:
n + 30.7

(2) [SysCmdComplete]:
n + 24.7

(3) [SysCmdError]:
n + 24.6



System Functions Examples Overview

NOTE: *System Functions are supported only when the CTRIO module is installed in the same base as the DirectLOGIC CPU.*

The Systems Functions examples on the following pages use the ***Direct***LOGIC Write to Intelligent Module (WT) and/or Read from Intelligent Module (RD) instructions to write to or read from the CTRIO's internal registers.

Reading From CTRIO Internal Memory

Reading the CTRIO's internal memory consists of several steps. Step one is using the WT instruction to send a Systems Function's command to the CTRIO telling it to put its internal register values into the CTRIO's "shared RAM". Step two is processing the request for the internal register values using the Process Command bit. Step three is using the RD instruction to read the values from the CTRIO's "shared RAM" memory into PLC V-memory.

Steps 1 and 2: WT instruction and Process Command
PLC V-memory ==> CTRIO's Shared RAM
CTRIO's Shared RAM ==> Process Command to internal processor
CTRIO's Shared RAM <== Internal data values

Step 3: RD instruction
PLC V-memory <== CTRIO's Shared RAM

Writing to CTRIO Internal Memory

Writing to the CTRIO's internal registers is basically a two step process. Step one is using the WT instruction to send a System Function command and the desired data values to the CTRIO's "Shared RAM". Step two is using the Process Command bit to tell the CTRIO to process the command and data values that are in the CTRIO's Shared RAM. This moves the data values from the Shared RAM into the CTRIO's internal registers.

Steps 1 and 2: WT instruction (command and data) and Process Command Bit:
PLC V-memory ==> CTRIO Shared RAM
CTRIO Shared RAM ==> Process Command to internal processor
CTRIO Shared RAM ==> internal data registers

NOTE: *This function is not available when the CTRIO module is installed in an EBC expansion base.*

Single Channel Simulating Retentive Quad Counter

This example program will simulate a retentive count register in the CTRIO. It will store the current count from the CTRIO in the PLC's retentive memory, then on a powerup, it will write the stored count back into the CTRIOs current count register.

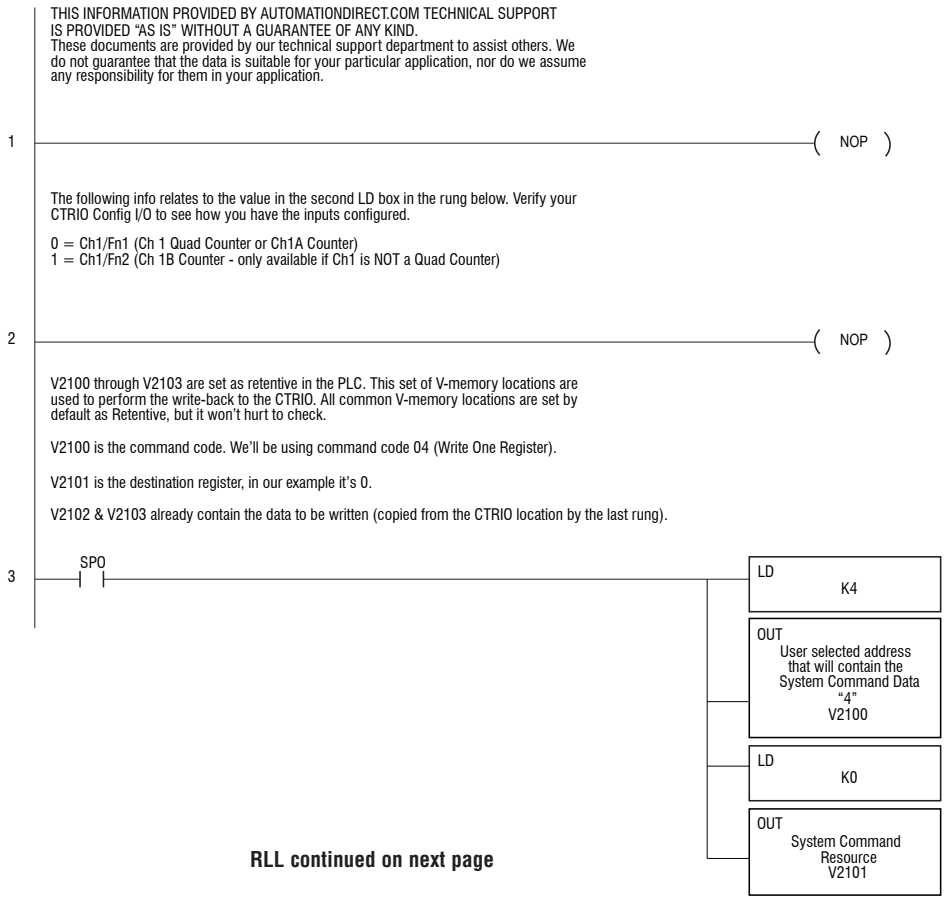
The example assumes the use of standard I/O mapping based off V2000 for Inputs, and V2030 for Outputs. Adjust according to your CTRIO Config I/O map.

V2100-2103 is just an address range, it can be altered by the user if desired. But all the associated addresses in the locations need to be altered to match.

The raw count from the CTRIO is part of the standard published I/O data from the CTRIO. In this case the scaled value is a DWORD at V2000, the raw count is a DWORD at V2002.

The CTRIO only accepts the raw count as the retentive value. On every scan after T0 completes, the raw count at V2002 is copied to V2102. (V2000 needs to be copied if Scaling is not used; see last rungs).

Example RLL for Single Channel Simulation of Retentive Quad Counter



RLL continued on next page

RLL continued from previous page

A WT command is use to write the data to the CTRIO. Details are in Chapter 5 for WT setup. CTRIO is in Slot 2, so the first LD loads a constant of .... slot 3 would be K3, etc.

A total of 8 bytes (4 byte command at V2100 7 V2101, 4 byte value at V2102 & V2103). So the second LD loads a constant of 8.

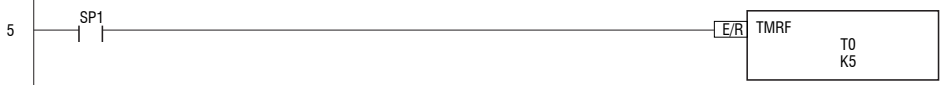
Offset 80 (hex) is the beginning of the command frame in the module's shared RAM. So the third LD loads a constant of 80.

The offset of 80 hex is specific to the CTRIO, just use that value.

The WT uses the K8 and loads 8 bytes of data from V2100 to the CTRIO.

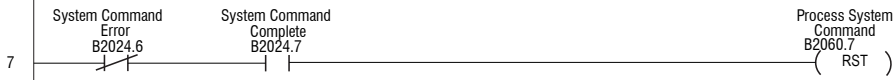
In PGM-to-RUN situations, this timer allows the CTRIO enough time to initialize before you ask it to restore the current count register.

In power up situations, this timer isn't necessary as the CTRIO will be initialized before the CPU gets to RUN mode.



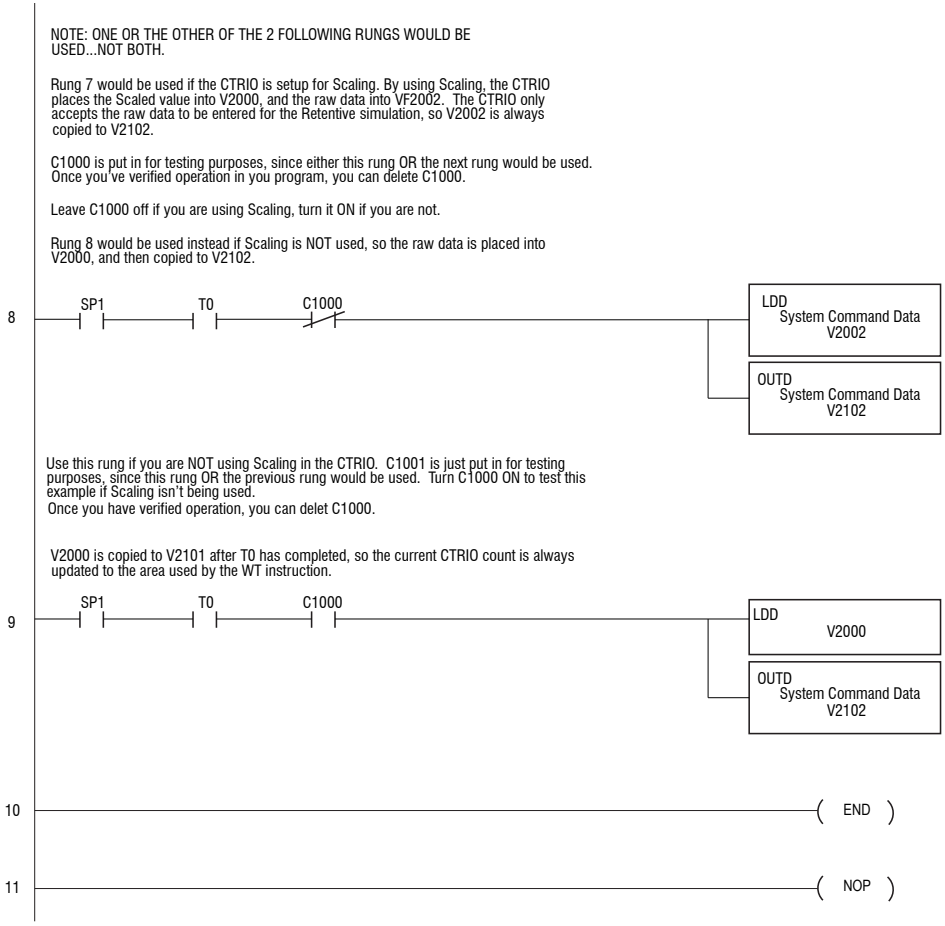
What you did on rung 5 was to write a command to the CTRIO. Now turning on the ProcessSysCmd bit will cause the CTRIO to perform the command you just entered.

Once you get the feedback bits for SysCommandError and SysCommandComplete, you can reset the ProcessSysCmd bit.



RLL continued on next page

RLL continued from previous page



Dual Channel Simulating Retentive Quad Counters

This example program will simulate a retentive count register in the CTRIO for both channels. It will store the current count from the CTRIO in the PLC's retentive memory, then on a powerup, it will write the stored count back into the CTRIOs current count register.

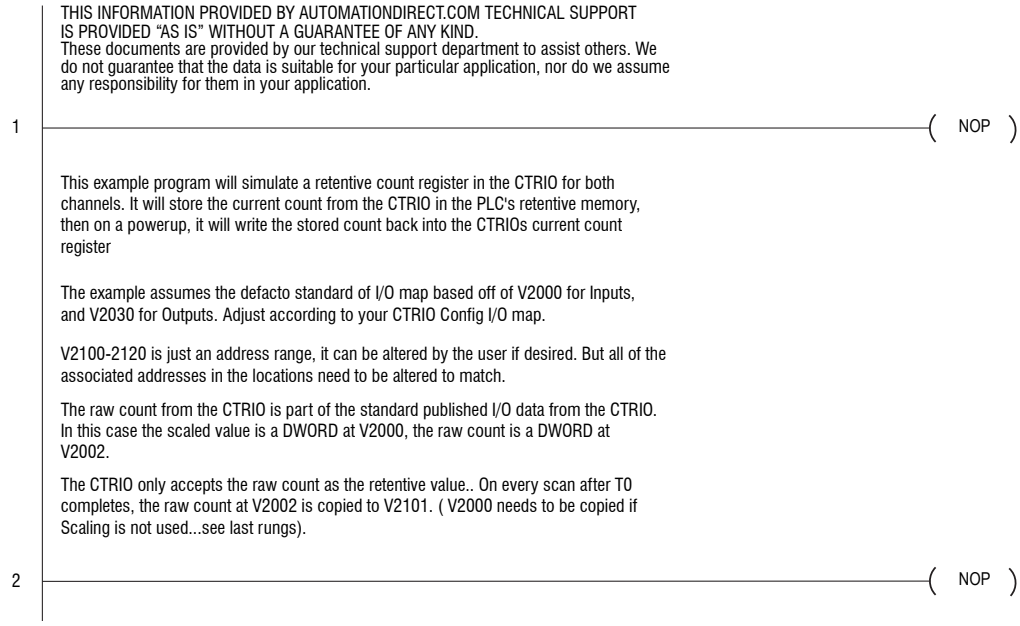
The example assumes the use of standard I/O mapping based off V2000 for Inputs, and V2030 for Outputs. Adjust according to your CTRIO Config I/O map.

V2100-2120 is just an address range, it can be altered by the user if desired. But all the associated addresses in the locations need to be altered to match.

The raw count from the CTRIO is part of the standard published I/O data from the CTRIO. In this case the scaled value is a DWORD at V2000, the raw count is a DWORD at V2002.

The CTRIO only accepts the raw count as the retentive value. On every scan after T0 completes, the raw count at V2002 is copied to V2101. (V2000 needs to be copied if Scaling is not used... see last rungs).

Example RLL for Dual Channel Simulating a Retentive Quad Counter



RLL continued on following page

RLL continued from previous page

2 — NOP

This section below is setup to make BOTH quadrature channels of the CTRIO be retentive on power cycle. Mapping for the CTRIO needs to start at V2000 for the inputs and V2030 for the outputs. CTRIOs can only be made retentive if they are located in the local base with the CPU.

This first rung writes a hex 2 to the command word, This is the command for 'write all'.

3 — SPO

- LD K2
- OUT User selected address that will contain the SysCmdCod value of "2" V2100

This rung writes values for:
LD K2 - the CTRIO slot # (slot 2 in this case)
LD K34 - the number of bytes (34)
LD K80 -the beginning offset (always k80) to shared RAM locations for the CTRIO.

V2100-V2120 range is changeable, just make sure to change ALL the references that are contained in this range. Using "Replace" feature of DirectSoft is recommended.

In lower rungs, the current CTRIO counts are copied to the V2100 range, so they can be retrieved after power-up or PGM>Run transition.

4 — SPO

- LD K2
- LD K34
- LD K80
- WT User selected address that will contain the SysCmdCod value of "2" V2100

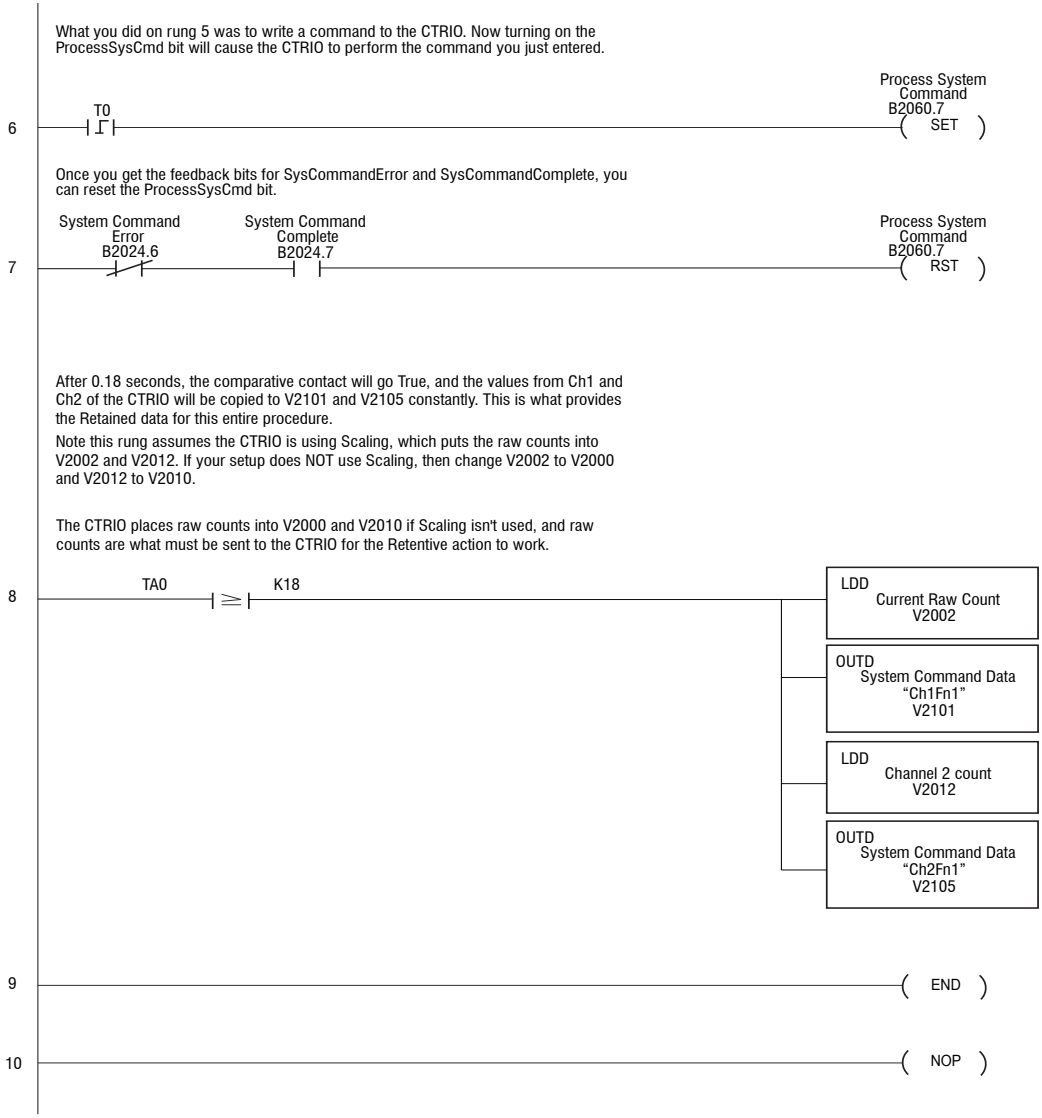
In PGM-to-RUN situations, this timer allows the CTRIO enough time to initialize before you ask it to restore the current count register.

In power up situations, this timer isn't necessary as the CTRIO will be initialized before the CPU gets to RUN mode.

5 — SP1 — E/R — TMRF T0 K5

RLL continued on following page

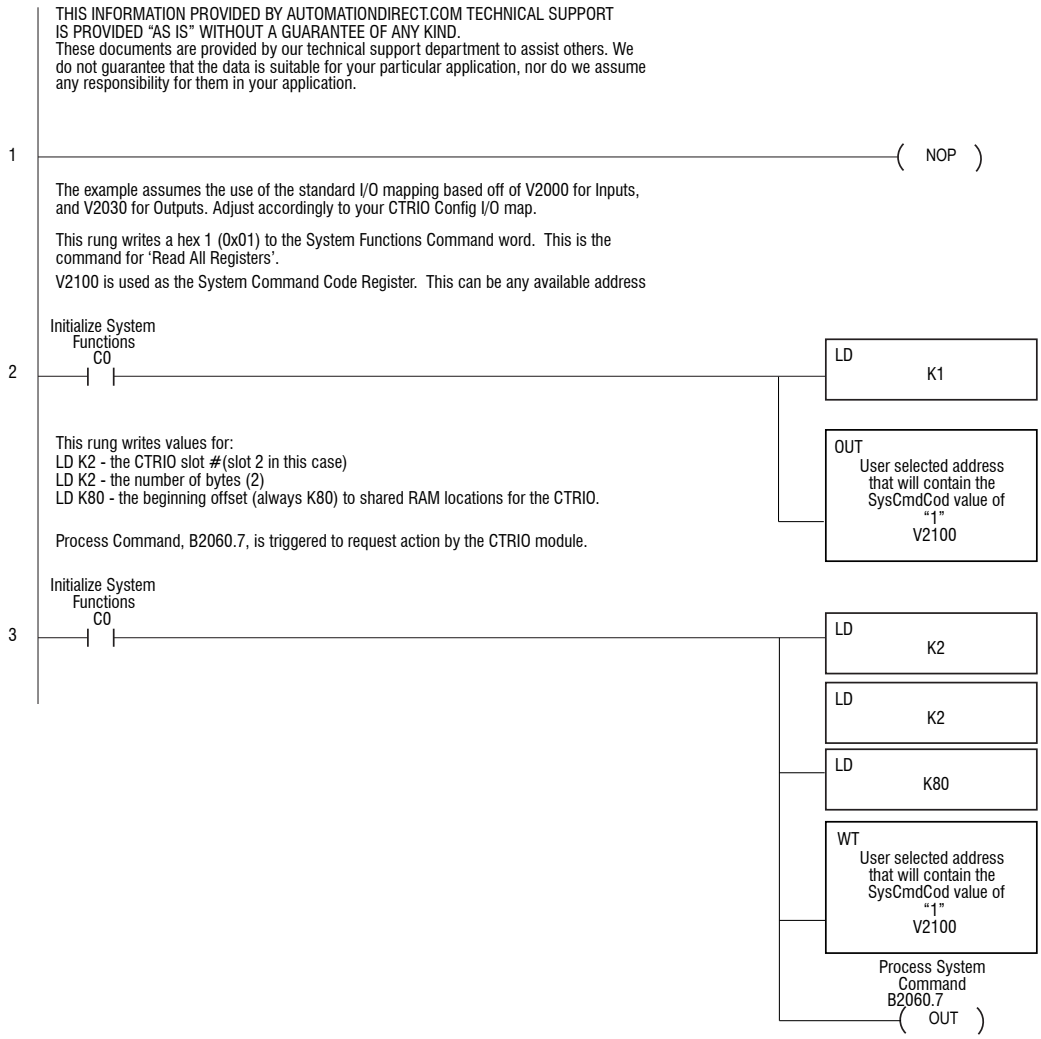
RLL continued from previous page



Reading CTRIO Internal Registers

The following Systems Functions example uses the Write to Intelligent Module (WT) and Read from Intelligent Module (RD) instructions to read all of the CTRIO's internal registers every 900ms and place the data starting at V2200.

Example RLL for Reading CTRIO Internal Registers



RLL continued on next page

RLL continued from previous page