Runtime Table Functions

Chapter 11

In This Chapter...

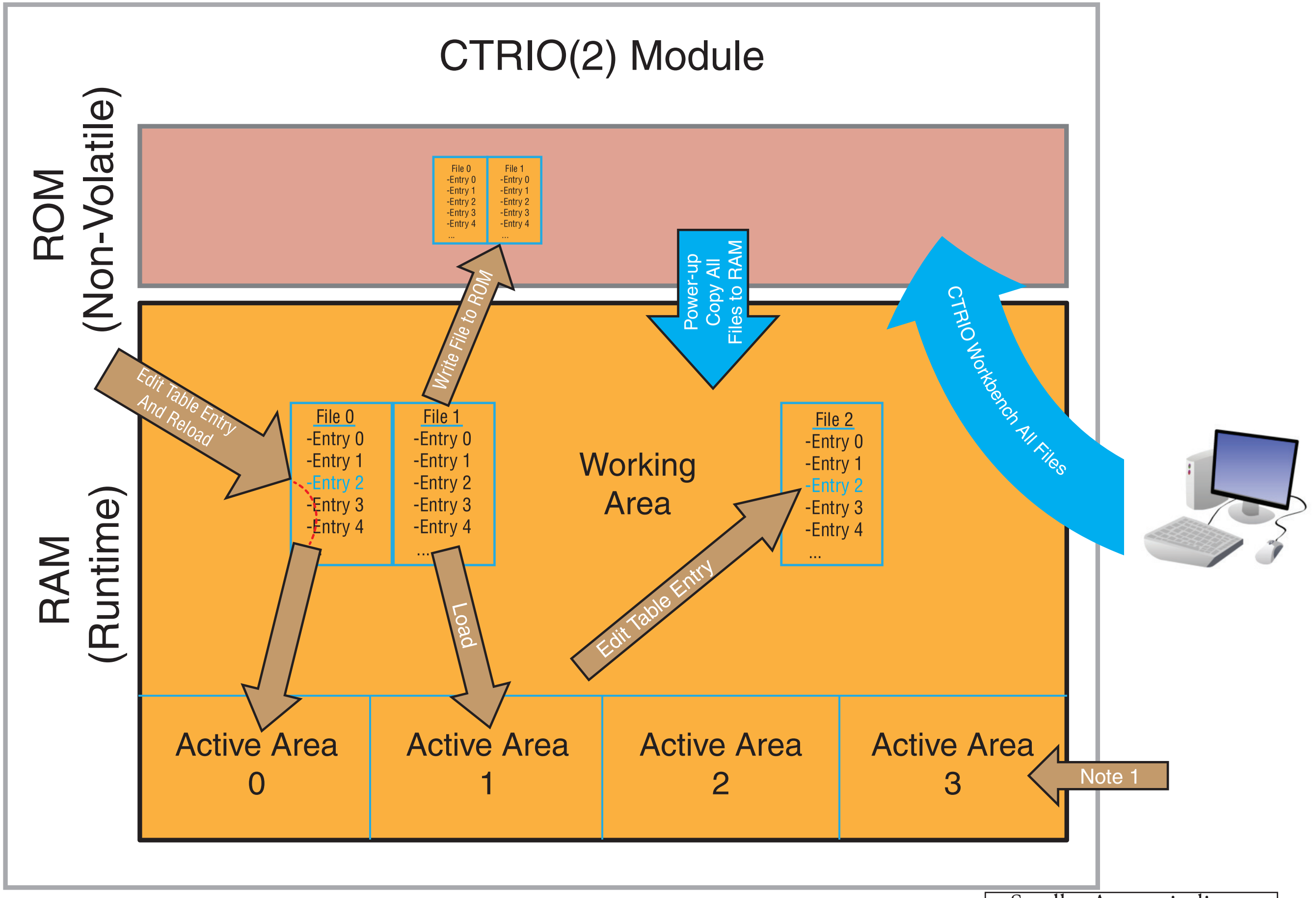
Introduction to Runtime Table Functions

The following drawing depicts the ways in which Files are moved and manipulated within a CTRIO(2).

- Preset and PLS (Programmable Limit Switches) Tables such as Pulse Profiles, exist as Files in the CTRIO(2) module's memory.
- There are two areas of memory in a CTRIO(2) module:
- ROM – this is the non-volatile (Flash) memory where files are stored to survive a power cycle
- RAM – this is the area of memory used at runtime, subdivided into:
- Working Area - All Tables and Profiles are copied from ROM to RAM Working Area at power up
- Active Areas - Each output has a dedicated area in RAM where its current Table or Profile is stored
- CTRIO Workbench only reads configuration data from ROM
- CTRIO Workbench only writes all configuration data to ROM and RAM, overwriting all contents of both areas
- All files are copied from ROM to RAM at power up
- Load function copies one File to one Active Area
- Write File to ROM function copies one File from RAM to ROM
- Clear Preset Table sets an Active Area back to its default state

NOTE: Various functions write or modify an Active Area based on parameters provided by the CPU.

Warning: Cannot move from an Active Area into the Working Area. Only files in the Working Area can be saved to ROM. Therefore, changes made using the function codes that directly target Files in an Active Area cannot be saved to ROM.



Note 1:

- Initialize Table
- Initialize Table on Reset
- Velocity Mode
- Run to Limit Mode
- Run to Position Mode
- Edit Level Response
- Add Table Entry
- Clear Table

Smaller Arrows indicate operations initiated programmatically.

Preset Tables and Programmable Limit Switch (PLS) Tables

Preset Tables and Programmable Limit Switch (PLS) Tables are handled programmatically using the following functions:

- Load Table: Moves the Table File from RAM into the specified Active Area for an Output Function
- Clear a Preset Table: Remove all entries from the Preset Table in RAM (providing a blank slate to build a new Table programmatically)
- Create a Preset Table: Creates a new Preset Table in RAM with one entry
- Create Preset Table On Reset:
- Add Entry to Preset Table: Add an entry to the end of the Preset Table in RAM
- Edit Preset Table: Modify an entry in the Preset Table in RAM
- Write File to ROM: Copies one File from RAM to ROM
- Edit Preset Table Entry and Reload: Modify an entry in the Preset Table in RAM and make it active
- Update Level (move up a couple notches?):

Load Preset Table

Description

Load Table loads the File from RAM Working Area into Active Area, making it the active Table for that Output. The Table can be a Preset Table or Programmable Limit Switch (PLS). The previous contents of the RAM Active Area are overwritten.

Prerequisites

- An Input configured for counting
- An Output configured to Discrete on Ch*/Fn*
- A Table created in CTRIO Workbench is loaded into the CTRIO(2)

Steps

Each step requires targeting the correct address for the desired Output. Check the Memory Map from your CTRIO(2) configuration. The following steps relate directly to the flow charts that follow.

- Set the Command Code to 0x10.
- Set Parameter 1 to the desired Table number in decimal.
- Set ProcessCmd ON to tell the CTRIO(2) to read in the CommandCode and Parameter1.

Load Preset Table, continued

The CTRIO(2) module begins the process of loading the Table. When it's done, it sets the CmdComplete bit and will set or reset the CmdError bit accordingly.

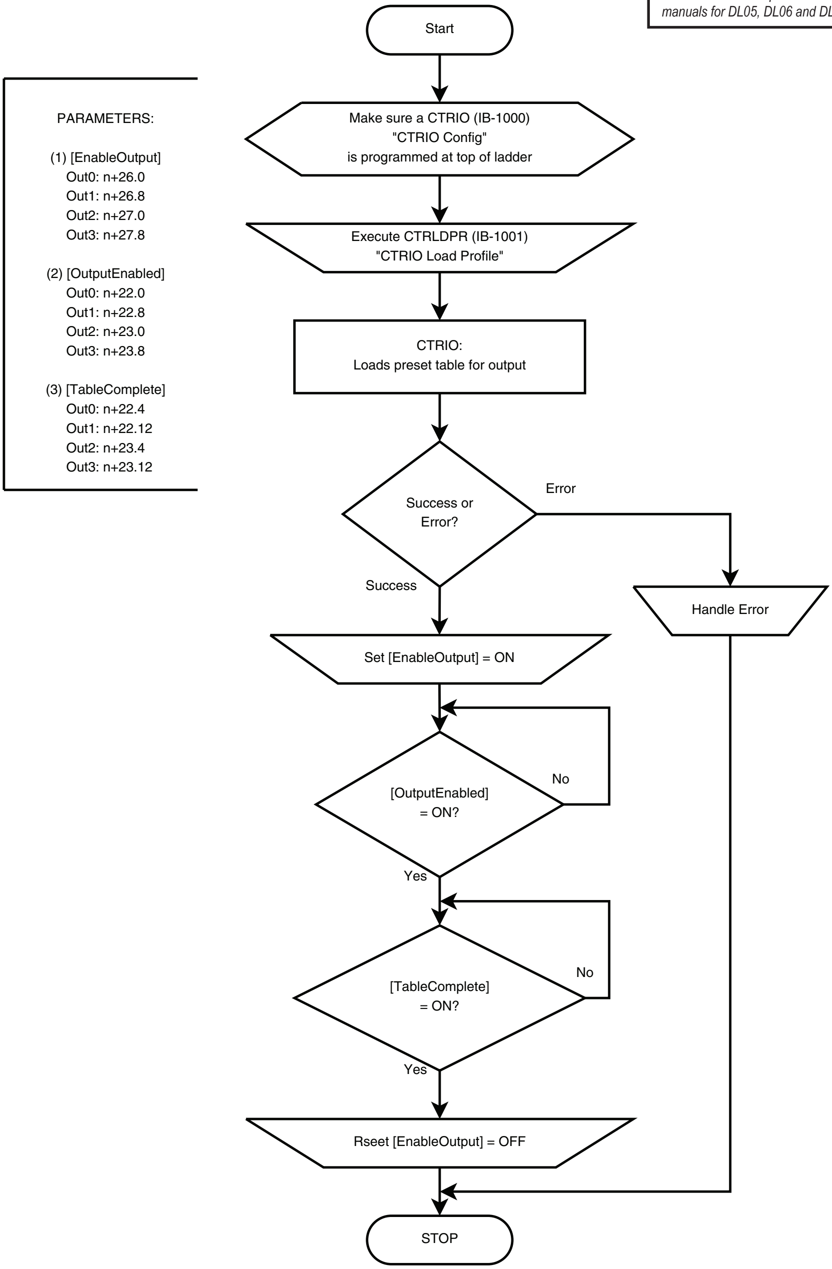
Once CmdComplete is seen high, your code should check whether the CmdError is ON.

If it is ON, the CTRIO(2) module failed to load the Table. Write code to handle the error and finish here.

- If CmdError is OFF, reset ProcessCmd and set EnableOutput ON.
- The CTRIO(2) module begins the process of enabling the Table. Once it is enabled, the CTRIO(2) module sets the OutputEnabled bit ON. The Output is now under control of the Table.
- To tell the CTRIO(2) to stop controlling the Output from the Table, reset EnableOutput to OFF.

Load Preset Table & Run (IBoxes)

NOTE: *An example program using these IBoxes can be found in Chapter 5 of the DirectLOGIC user manuals for DL05, DL06 and DL205.*



Load Preset Table & Run (DL-PLC)

PARAMETERS:

(1) [CommandCode]:
Out0: n+10
Out1: n+13
Out2: n+16
Out3: n+21

(2) [Parameter1]
Out0: n+11
Out1: n+14
Out2: n+17
Out3: n+22

(3) [ProcessCmd]
Out0: n+26.7
Out1: n+26.15
Out2: n+27.7
Out3: n+27.15

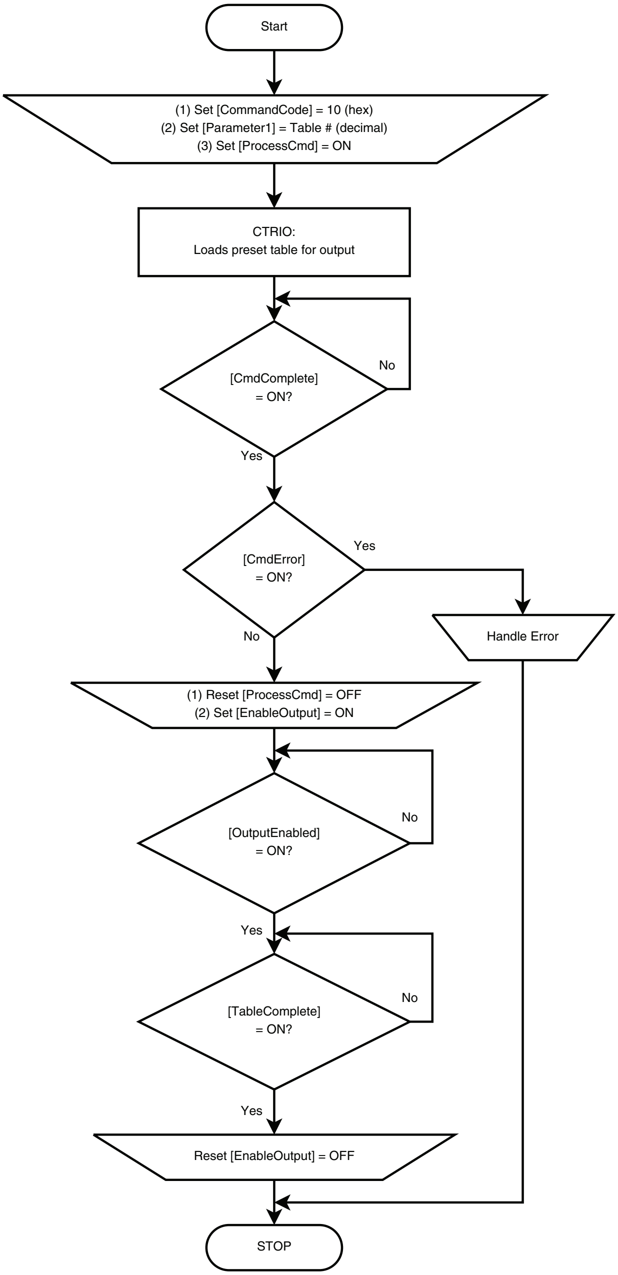
(4) [CmdComplete]
Out0: n+22.7
Out1: n+22.15
Out2: n+23.7
Out3: n+23.15

(5) [CmdError]
Out0: n+22.6
Out1: n+22.14
Out2: n+23.6
Out3: n+23.14

(6) [EnableOutput]
Out0: n+26.0
Out1: n+26.8
Out2: n+27.0
Out3: n+27.8

(7) [OutputEnabled]
Out0: n+22.0
Out1: n+22.8
Out2: n+23.0
Out3: n+23.8

(8) [TableComplete]
Out0: n+22.4
Out1: n+22.12
Out2: n+23.4
Out3: n+23.12


Start
(1) Set [CommandCode] = 10 (hex)
(2) Set [Parameter1] = Table # (decimal)
(3) Set [ProcessCmd] = ON
CTRIO:
Loads preset table for output
[CmdComplete]
= ON?
No
Yes
[CmdError]
= ON?
Yes
No
Handle Error
(1) Reset [ProcessCmd] = OFF
(2) Set [EnableOutput] = ON
[OutputEnabled]
= ON?
No
Yes
[TableComplete]
= ON?
No
Yes
Reset [EnableOutput] = OFF
STOP

Load Table

Description

Load Table loads the File from RAM Working Area into Active Area, making it the active Table for that Output. The Table can be a Preset Table or Programmable Limit Switch (PLS). The previous contents of the RAM Active Area are overwritten.

Prerequisites

- An Input configured for counting
- An Output configured to Discrete on Ch*/Fn*
- A Table created in CTRIO Workbench is loaded into the CTRIO(2) or was built programmatically.

Steps

Each step requires targeting the correct address for the desired Output. Check the Memory Map from your CTRIO(2) configuration. The following steps relate directly to the flow charts that follow.

- Set the Command Code to 0x10.
- Set Parameter 1 to the desired Table number in decimal.
- Set ProcessCmd ON to initiate the CTRIO(2) to read in the CommandCode and Parameter1.

The CTRIO(2) module begins the process of loading the Table. When it's done, it sets the CmdComplete bit and will set or reset the CmdError bit accordingly.

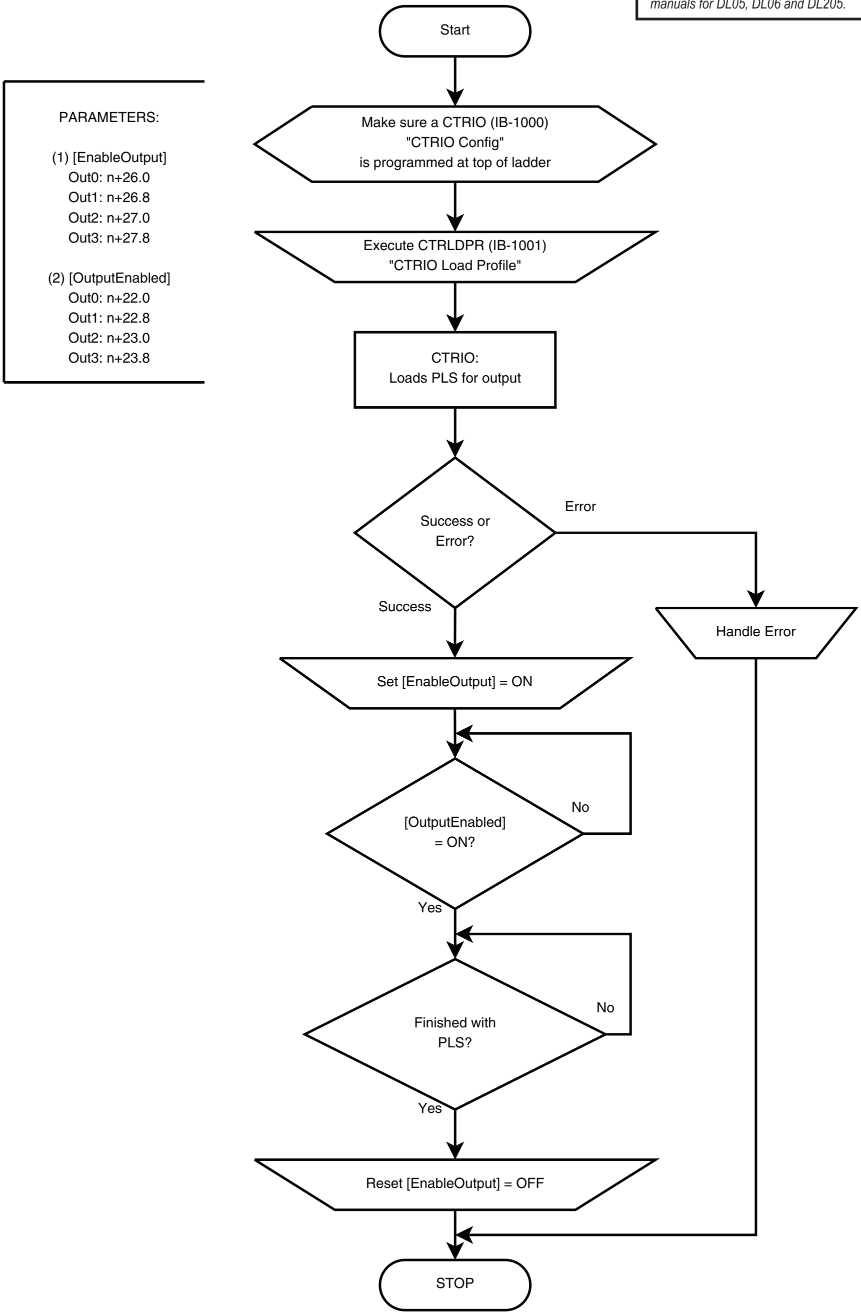
Once CmdComplete is seen high, your code should check the status of CmdError:

If it is ON, the CTRIO(2) module failed to load the Table. Write code to handle the error and finish here.

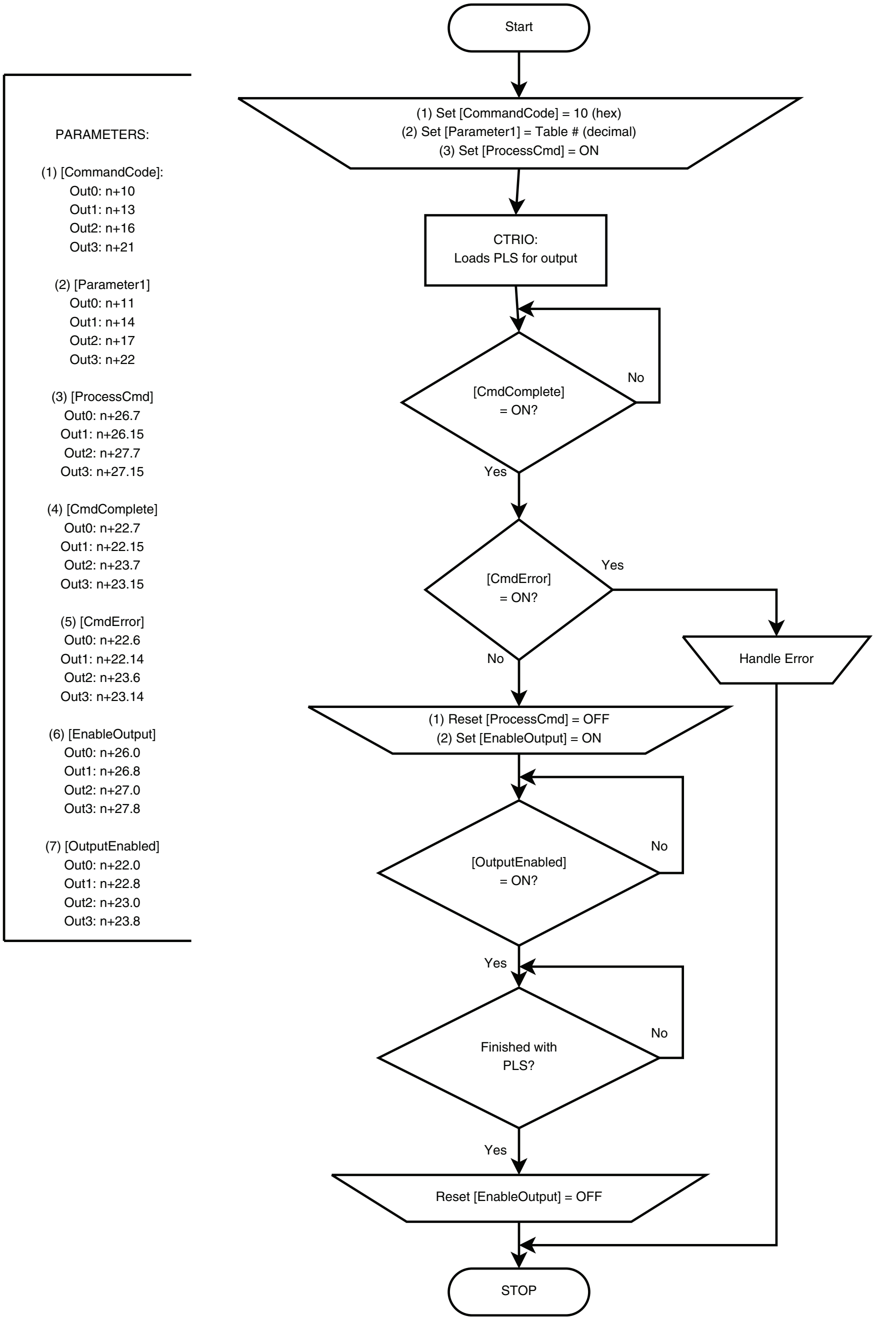
- If CmdError is OFF, reset ProcessCmd and set EnableOutput ON.
- The CTRIO(2) module begins the process of enabling the Table. Once it is enabled, the CTRIO(2) module sets the OutputEnabled bit ON. The Output is now under control of the Table.
- To tell the CTRIO(2) to stop controlling the Output from the Table, reset EnableOutput to OFF.

Load PLS & Run (IBoxes)

NOTE: *An example program using these IBoxes can be found in Chapter 5 of the DirectLOGIC user manuals for DL05, DL06 and DL205.*



Load PLS & Run
(DL-PLC)



Clear Preset Table

Description

Clear Table returns the Active Area for an Output to its default state, blank. Using it is not required as any function can be disabled, and loading any new function overwrites the previous contents. It might be used to make absolutely sure an Output cannot be turned ON.

Prerequisites

None

Steps

Each step requires targeting the correct address for the desired Output. Check the Memory Map from your CTRIO(2) configuration. The following steps relate directly to the flow charts that follow.

- Set the Command Code to 0x11.
- Set ProcessCmd ON to tell the CTRIO(2) to read in the CommandCode.

The CTRIO(2) module begins the process of clearing the Active Area. When it's done, it sets the CmdComplete bit and will set or reset the CmdError bit accordingly.

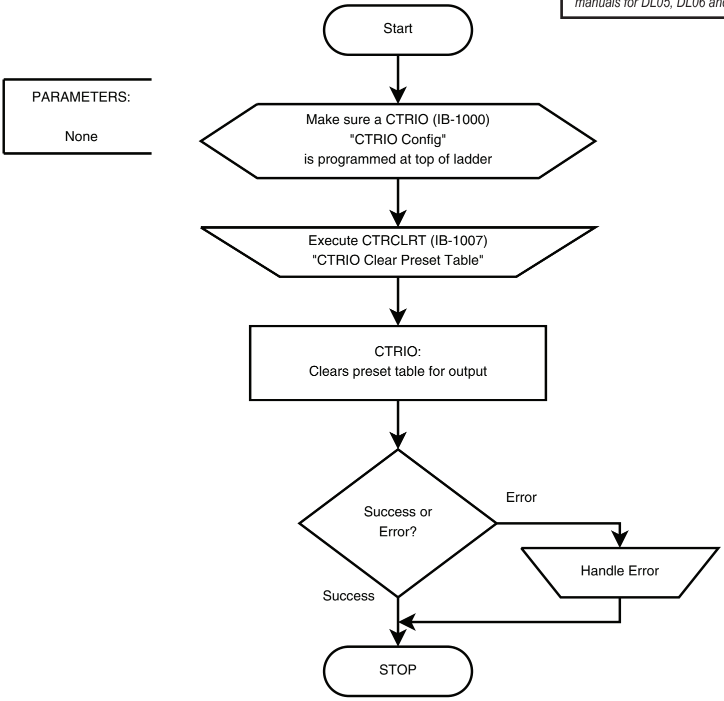
Once CmdComplete is seen high, your code should check whether the CmdError is ON.

If CmdError is ON, the CTRIO(2) module failed to clear the Active Area. Write code to handle the error and finish here.

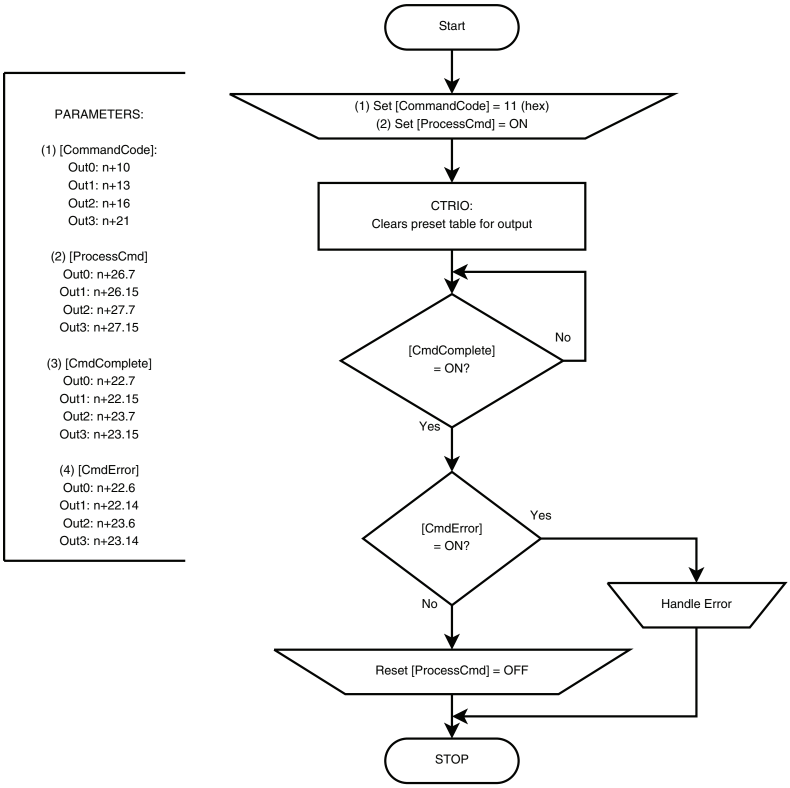
- If CmdError is OFF, the Active Area is clear.
- Reset ProcessCmd OFF.

Clear Preset Table (IBoxes)

NOTE: *An example program using these IBoxes can be found in Chapter 5 of the DirectLOGIC user manuals for DL05, DL06 and DL205.*


Start
PARAMETERS:
None
Make sure a CTRIO (IB-1000)
"CTRIO Config"
is programmed at top of ladder
Execute CTRCLRT (IB-1007)
"CTRIO Clear Preset Table"
CTRIO:
Clears preset table for output
Success or
Error?
Error
Handle Error
Success
STOP

Clear Preset Table
(DL-PLC)



Create Preset Table (Initialize Table)

Description

Initialize Table creates a Table with one entry in the Active Area for an Output. The single entry is defined in Parameters 1-3. The new Table immediately becomes active.

Prerequisites

- An Input configured for counting
- An Output configured to Discrete on Ch*/Fn*

Steps

Each step requires targeting the correct address for the desired Output. Check the Memory Map from your CTRIO(2) configuration. The following steps relate directly to the flow charts that follow.

- Set the Command Code to 0x12.
- Write a value to Parameter1 to define the entry type.

Parameter1	Function
0	Set
1	Reset
2	Pulse ON
3	Pulse OFF
4	Toggle OFF
5	Reset Count

If Parameter1 is either Pulse function (2 or 3), write a value to Parameter2 to specify Pulse Time in milliseconds. Otherwise, Parameter2 is ignored.

- Write a value to Parameter3 to specify the counts where the function should occur.
- Set ProcessCmd ON to tell the CTRIO(2) to read in the CommandCode.

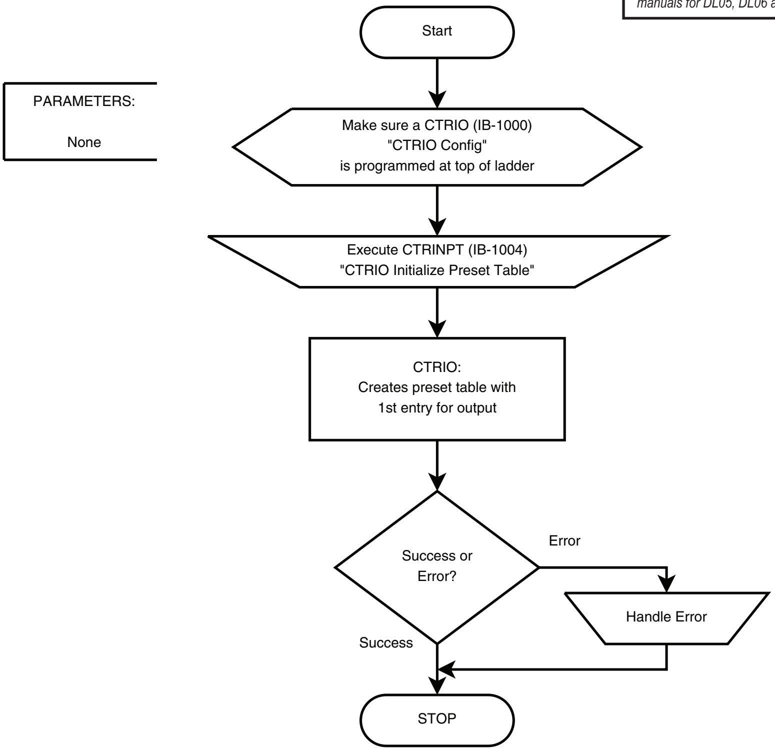
The CTRIO(2) module begins the process of creating the Table. When it’s done, it sets the CmdComplete bit and will set or reset the CmdError bit accordingly.

Once CmdComplete is seen high, your code should check the status of CmdError:

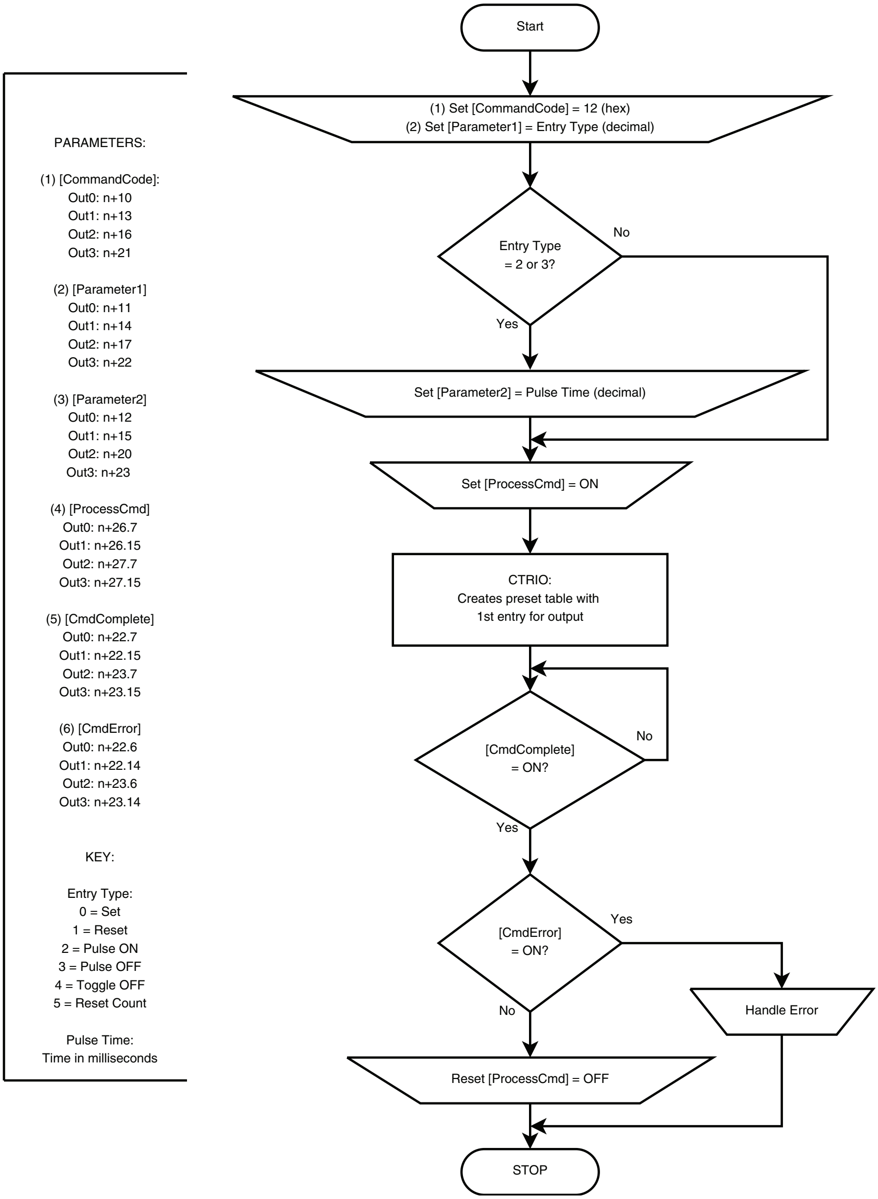
- If CmdError is ON, the CTRIO(2) module failed to create the Table. Write code to handle the error and finish here.
- If CmdError is OFF, the Table was created.
- Reset ProcessCmd OFF.

Create Preset Table (IBoxes)

NOTE: *An example program using these IBoxes can be found in Chapter 5 of the DirectLOGIC user manuals for DL05, DL06 and DL205.*


Start
PARAMETERS:
None
Make sure a CTRIO (IB-1000) "CTRIO Config" is programmed at top of ladder
Execute CTRINPT (IB-1004) "CTRIO Initialize Preset Table"
CTRIO: Creates preset table with 1st entry for output
Success or Error?
Error
Handle Error
Success
STOP

Create Preset Table (DL-PLC)



Add Entry to Preset Table

Description

- Add Entry to Preset Table adds a new entry to the end of a Table in an Active Area. The entry is defined in Parameters 1-3.

Prerequisites

- An Input configured for counting
- An Output configured to Discrete on Ch*/Fn*
- A Preset Table loaded into an Active Area

Steps

Each step requires targeting the correct address for the desired Output. Check the Memory Map from your CTRIO(2) configuration. The following steps relate directly to the flow charts that follow.

- Set the Command Code to 0x13.
- Write a value to Parameter1 to define the entry type.

Parameter1	Function
0	Set
1	Reset
2	Pulse ON
3	Pulse OFF
4	Toggle OFF
5	Reset Count

If Parameter1 is either Pulse function (2 or 3), write a value to Parameter2 to specify Pulse Time in milliseconds. Otherwise, Parameter2 is ignored.

Write a value to Parameter3 to specify the counts where the function should occur.

Set ProcessCmd ON to tell the CTRIO(2) to read in the CommandCode.

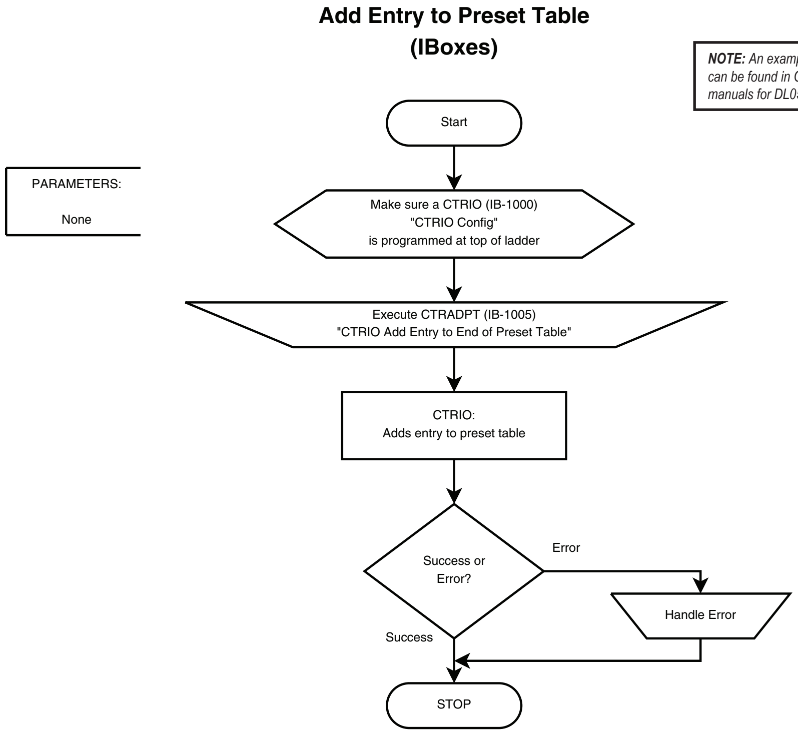
The CTRIO(2) module begins the process of adding the Table entry. When it's done, it sets the CmdComplete bit and will set or reset the CmdError bit accordingly.

Once CmdComplete is seen high, your code should check the status of CmdError:

- If CmdError is ON, the CTRIO(2) module failed to add the Table entry. Write code to handle the error and finish here.
- If CmdError is OFF, the updated Table is ready for use.
- Reset ProcessCmd OFF.

CTRIO2 - DirectLOGIC Using IBoxes Flowcharts

***NOTE:** An example program using these IBoxes can be found in Chapter 5 of the DirectLOGIC user manuals for DL05, DL06 and DL205.*


Add Entry to Preset Table
(IBoxes)
PARAMETERS:
None
Start
Make sure a CTRIO (IB-1000)
"CTRIO Config"
is programmed at top of ladder
Execute CTRADPT (IB-1005)
"CTRIO Add Entry to End of Preset Table"
CTRIO:
Adds entry to preset table
Success or
Error?
Error
Handle Error
Success
STOP

CTRIO2 - DirectLOGIC PLC Flowcharts

Add Entry to Preset Table (DL-PLC)

PARAMETERS:

(1) [CommandCode]:
Out0: n+10
Out1: n+13
Out2: n+16
Out3: n+21

(2) [Parameter1]
Out0: n+11
Out1: n+14
Out2: n+17
Out3: n+22

(3) [Parameter2]
Out0: n+12
Out1: n+15
Out2: n+20
Out3: n+23

(4) [Parameter3]
Out0: n+0-1
Out1: n+2-3
Out2: n+4-5
Out3: n+6-7

(5) [ProcessCmd]
Out0: n+26.7
Out1: n+26.15
Out2: n+27.7
Out3: n+27.15

(6) [CmdComplete]
Out0: n+22.7
Out1: n+22.15
Out2: n+23.7
Out3: n+23.15

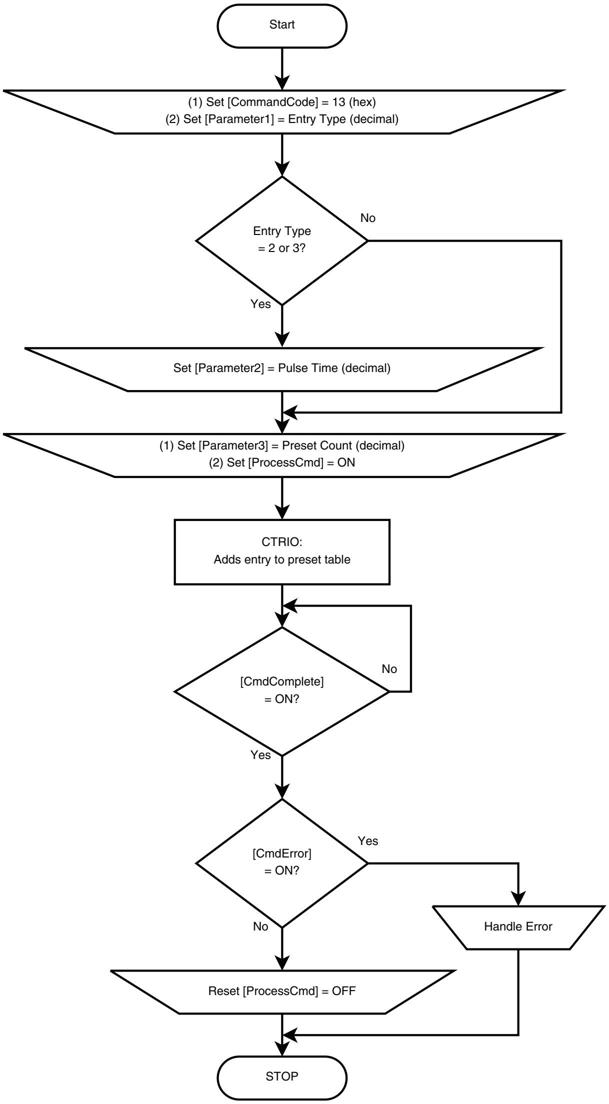
(7) [CmdError]
Out0: n+22.6
Out1: n+22.14
Out2: n+23.6
Out3: n+23.14

KEY:

Entry Type:
0 = Set
1 = Reset
2 = Pulse ON
3 = Pulse OFF
4 = Toggle OFF
5 = Reset Count

Pulse Time:
Time in milliseconds

Preset Count:
Raw count (DW)


Start
(1) Set [CommandCode] = 13 (hex)
(2) Set [Parameter1] = Entry Type (decimal)
Entry Type
= 2 or 3?
No
Yes
Set [Parameter2] = Pulse Time (decimal)
(1) Set [Parameter3] = Preset Count (decimal)
(2) Set [ProcessCmd] = ON
CTRIO:
Adds entry to preset table
[CmdComplete]
= ON?
No
Yes
[CmdError]
= ON?
Yes
No
Handle Error
Reset [ProcessCmd] = OFF
STOP

Edit Preset Table Entry

Description

Edit Preset Table Entry is used to overwrite one Preset Table Entry in RAM Working Area. The new table does not automatically get Loaded and become ready for use. To make the new Preset Table active, execute a Load Preset Table. (With CTRIO2, if the table will immediately be loaded to RAM, use Edit Preset Table Entry and Reload instead. It saves code and executes more quickly.)

Command Code Low Byte	Function
14	Edit Preset Table Entry
Parameter1 High Byte	**Entry#**
00	Not Used
01	File #1
02	File #2
...	...

Prerequisites

- An Input configured for counting
- An Output configured to Discrete on Ch*/Fn*
- A Table created in CTRIO Workbench is loaded into the CTRIO(2) or was built programmatically

Steps

Each step requires targeting the correct address for the desired Output. Check the Memory Map from your CTRIO(2) configuration. The following steps relate directly to the flow charts that follow.

Write a value to the Command Code high byte for the file number of the table to edit and the low byte to 0x14. For example, 0x0214 would be for working on file 2, Edit Preset Table Entry.

Write a value to Parameter1 high byte for the entry number and to the low byte to define the entry type. For example, 0x0201 would change entry 3 on table 1 (file 1).

If Parameter1 is either Pulse function (2 or 3), write a value to Parameter2 to specify Pulse Time in milliseconds. Otherwise, Parameter2 is ignored.

Write a value to Parameter3 to specify the counts where the function should occur.

Set ProcessCmd ON to tell the CTRIO(2) to read in the CommandCode.

The CTRIO(2) module begins the process of editing the Table entry. When it's done, it sets the CmdComplete bit and will set or reset the CmdError bit accordingly.

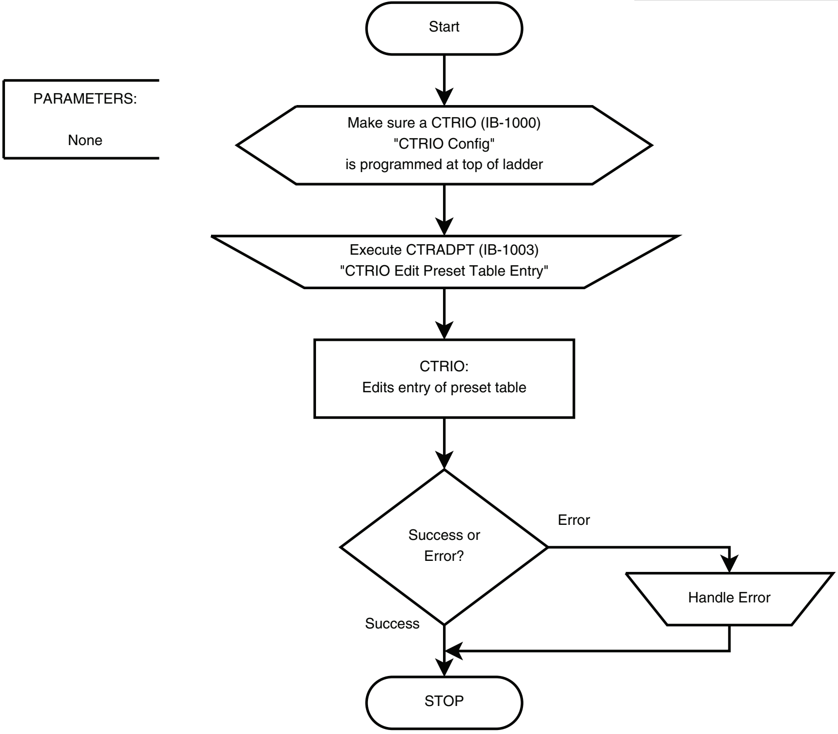
Once CmdComplete is seen high, your code should check the status of CmdError:

Parameter1 Low Byte	Function
00	Set
01	Reset
02	Pulse ON
03	Pulse OFF
04	Toggle OFF
05	Reset Count
Parameter1 High Byte	**Entry#**
00	Entry 1
01	Entry 2
02	Entry 3
...	...

- If CmdError is ON, the CTRIO(2) module failed to edit the Table entry. Write code to handle the error and finish here.
- If CmdError is OFF, the updated Table is ready for use.
- Reset ProcessCmd OFF.

Edit Preset Table Entry (IBoxes)

***NOTE:** An example program using these IBoxes can be found in Chapter 5 of the DirectLOGIC user manuals for DL05, DL06 and DL205.*


Start
PARAMETERS:
None
Make sure a CTRIO (IB-1000)
"CTRIO Config"
is programmed at top of ladder
Execute CTRADPT (IB-1003)
"CTRIO Edit Preset Table Entry"
CTRIO:
Edits entry of preset table
Success or
Error?
Error
Handle Error
Success
STOP

Edit Preset Table Entry
(DL-PLC)

PARAMETERS:

(1) [Table#-CmdCode]:
Out0: n+10
Out1: n+13
Out2: n+16
Out3: n+21

(2) [Entry#-Type]
Out0: n+11
Out1: n+14
Out2: n+17
Out3: n+22

(3) [Parameter2]
Out0: n+12
Out1: n+15
Out2: n+20
Out3: n+23

(4) [Parameter3]
Out0: n+0-1
Out1: n+2-3
Out2: n+4-5
Out3: n+6-7

(5) [ProcessCmd]
Out0: n+26.7
Out1: n+26.15
Out2: n+27.7
Out3: n+27.15

(6) [CmdComplete]
Out0: n+22.7
Out1: n+22.15
Out2: n+23.7
Out3: n+23.15

(7) [CmdError]
Out0: n+22.6
Out1: n+22.14
Out2: n+23.6
Out3: n+23.14

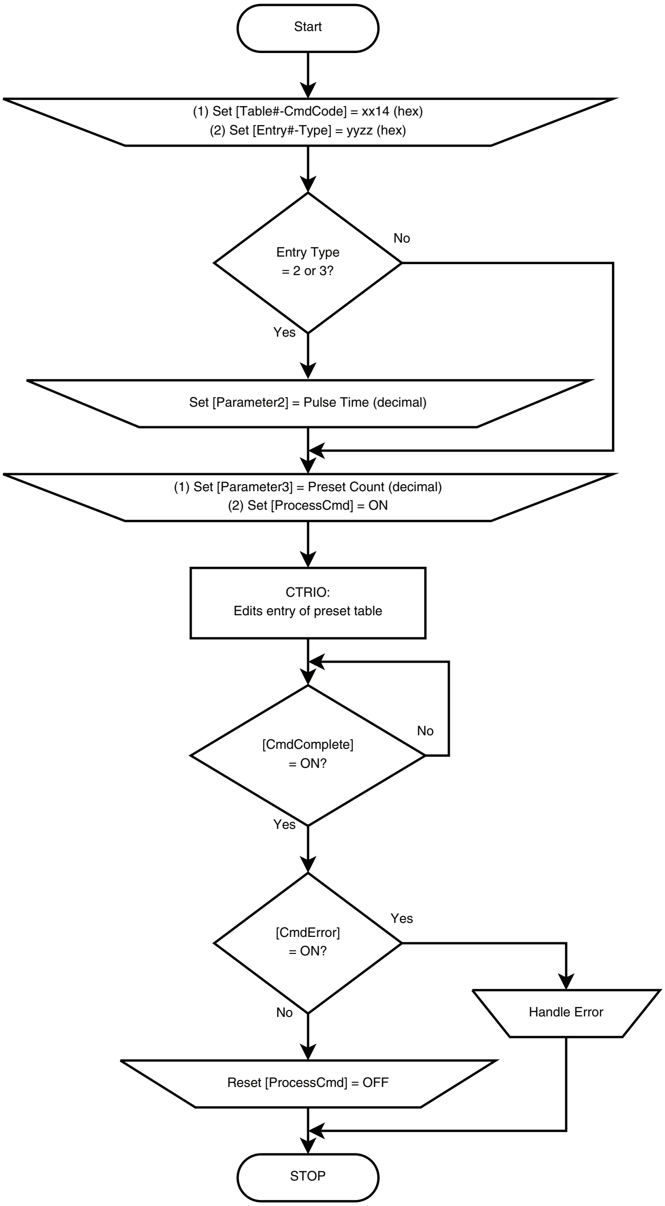
KEY:

xx14:
xx = Table #

yyzz:
yy = Entry #
zz
0 = Set
1 = Reset
2 = Pulse ON
3 = Pulse OFF
4 = Toggle OFF
5 = Reset Count

Pulse Time:
Time in milliseconds

Preset Count:
Raw count (DW)


Start
(1) Set [Table#-CmdCode] = xx14 (hex)
(2) Set [Entry#-Type] = yyzz (hex)
Entry Type
= 2 or 3?
No
Yes
Set [Parameter2] = Pulse Time (decimal)
(1) Set [Parameter3] = Preset Count (decimal)
(2) Set [ProcessCmd] = ON
CTRIO:
Edits entry of preset table
[CmdComplete]
= ON?
No
Yes
[CmdError]
= ON?
Yes
No
Handle Error
Reset [ProcessCmd] = OFF
STOP

Edit Preset Table Entry and Reload

Description

Edit Preset Table Entry and Reload combines the functions of Edit Preset Table Entry and Load Preset Table.

Command Code Low Byte	Function
15	Edit Preset Table Entry and Reload
Parameter1 High Byte	**Entry#**
00	Not Used
01	File #1
02	File #2
...	...

Prerequisites

- An Input configured for counting
- An Output configured to Discrete on Ch*/Fn*
- A Table created in CTRIO Workbench is loaded into the CTRIO(2) or was built programmatically

Steps

Each step requires targeting the correct address for the desired Output. Check the Memory Map from your CTRIO(2) configuration. The following steps relate directly to the flow charts that follow.

Write a value to the Command Code high byte for the file number of the table to edit and the low byte to 0x15. For example, 0x0215 would be for working on file 2, Edit Preset Table Entry and Reload.

Write a value to Parameter1 high byte for the entry number and to the low byte to define the entry type. For example, 0x0201 would change entry 3 on table 1 (file 1).

Parameter1 Low Byte	Function
00	Set
01	Reset
02	Pulse ON
03	Pulse OFF
04	Toggle OFF
05	Reset Count
Parameter1 High Byte	**Entry#**
00	Entry 1
01	Entry 2
02	Entry 3
...	...

If Parameter1 is either Pulse function (2 or 3), write a value to Parameter2 to specify Pulse Time in milliseconds. Otherwise, Parameter2 is ignored.

Write a value to Parameter3 to specify the counts where the function should occur.

Set ProcessCmd ON to tell the CTRIO(2) to read in the CommandCode.

The CTRIO(2) module begins the process of editing the Table entry and loading the table. When it's done, it sets the CmdComplete bit and will set or reset the CmdError bit accordingly.

Once CmdComplete is seen high, your code should check the status of CmdError:

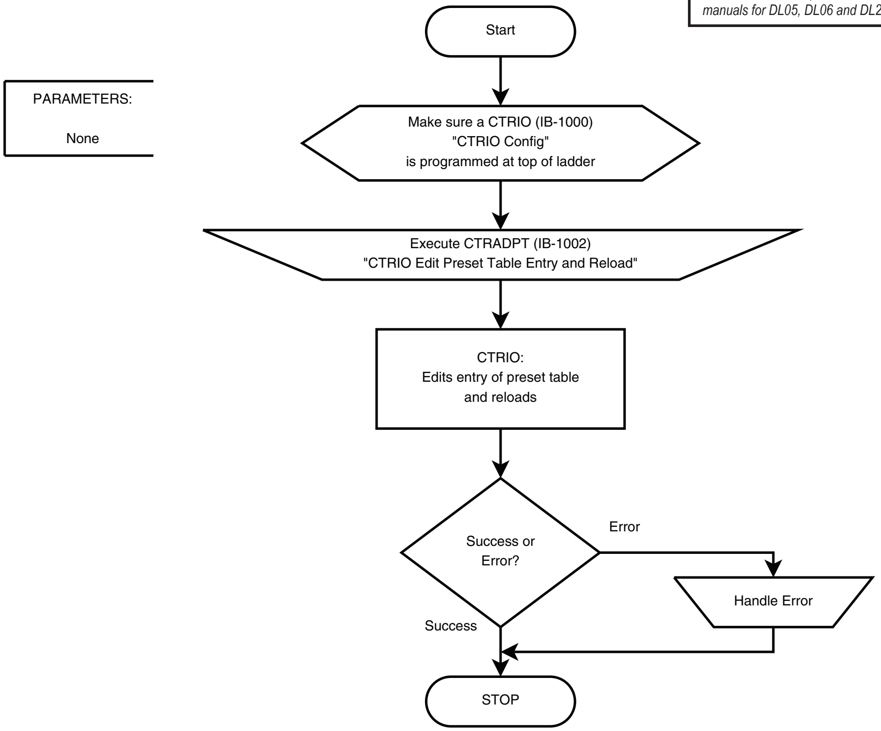
If CmdError is ON, the CTRIO(2) module failed to edit the Table entry. Write code to handle the error and finish here.

If CmdError is OFF, the updated Table is ready for use.

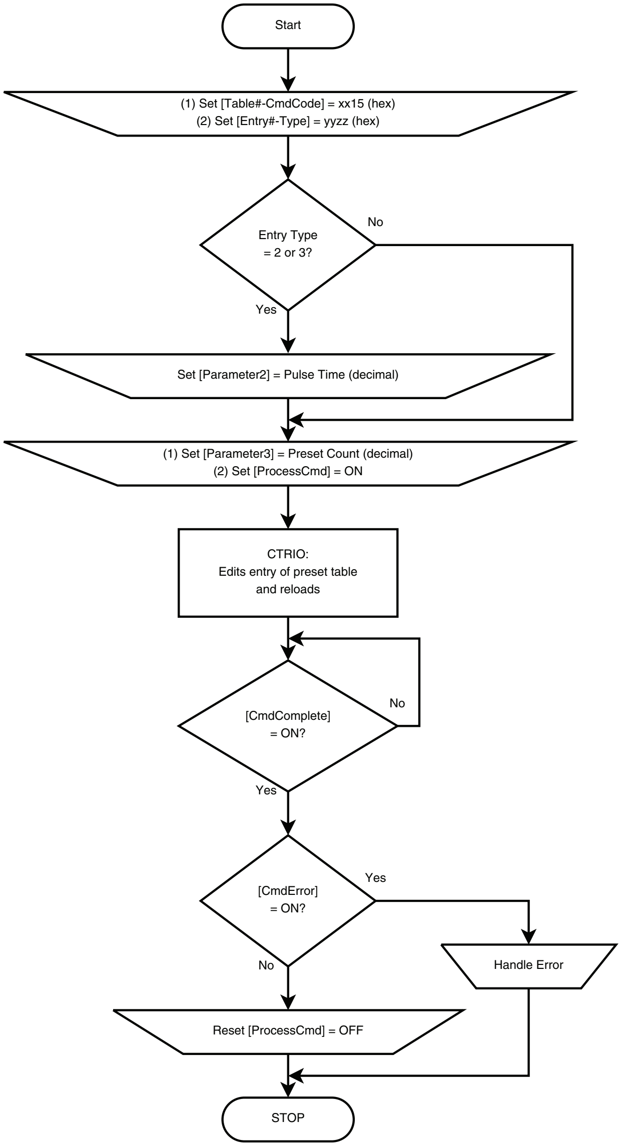
Reset ProcessCmd OFF.

Edit Preset Table Entry & Reload (IBoxes)

NOTE: *An example program using these IBoxes can be found in Chapter 5 of the DirectLOGIC user manuals for DL05, DL06 and DL205.*


PARAMETERS:
None
Start
Make sure a CTRIO (IB-1000) "CTRIO Config" is programmed at top of ladder
Execute CTRADPT (IB-1002) "CTRIO Edit Preset Table Entry and Reload"
CTRIO: Edits entry of preset table and reloads
Success or Error?
Error
Handle Error
Success
STOP

Edit Preset Table Entry & Reload (DL-PLC)


Start
(1) Set [Table#-CmdCode] = xx15 (hex)
(2) Set [Entry#-Type] = yyzz (hex)
Entry Type = 2 or 3?
No
Yes
Set [Parameter2] = Pulse Time (decimal)
(1) Set [Parameter3] = Preset Count (decimal)
(2) Set [ProcessCmd] = ON
CTRIO: Edits entry of preset table and reloads
[CmdComplete] = ON?
No
Yes
[CmdError] = ON?
Yes
No
Handle Error
Reset [ProcessCmd] = OFF
STOP

PARAMETERS:

(1) [Table#-CmdCode]:
Out0: n+10
Out1: n+13
Out2: n+16
Out3: n+21

(2) [Entry#-Type]
Out0: n+11
Out1: n+14
Out2: n+17
Out3: n+22

(3) [Parameter2]
Out0: n+12
Out1: n+15
Out2: n+20
Out3: n+23

(4) [Parameter3]
Out0: n+0-1
Out1: n+2-3
Out2: n+4-5
Out3: n+6-7

(5) [ProcessCmd]
Out0: n+26.7
Out1: n+26.15
Out2: n+27.7
Out3: n+27.15

(6) [CmdComplete]
Out0: n+22.7
Out1: n+22.15
Out2: n+23.7
Out3: n+23.15

(7) [CmdError]
Out0: n+22.6
Out1: n+22.14
Out2: n+23.6
Out3: n+23.14

KEY:

xx15:
xx = Table #

yyzz:
yy = Entry #
zz
0 = Set
1 = Reset
2 = Pulse ON
3 = Pulse OFF
4 = Toggle OFF
5 = Reset Count

Pulse Time:
Time in milliseconds

Preset Count:
Raw count (DW)

Write File to ROM

Description

Write File to ROM saves a Table from RAM Working Area to ROM. Performing this function is only needed when a Table has been modified programmatically and will need to be reloaded after a power cycle. Another option to achieve a similar function would be to always build the table programmatically as part of the control program.

Warning: The Flash memory used in CTRIO can be damaged by an excessive number of writes. It can typically tolerate 100,000 writes.

Prerequisites

- An Input configured for counting
- An Output configured to Discrete on Ch*/Fn*
- A Table created in CTRIO Workbench is loaded into the CTRIO(2) or was built programmatically

Steps

Each step requires targeting the correct address for the desired Output. Check the Memory Map from your CTRIO(2) configuration. The following steps relate directly to the flow charts that follow.

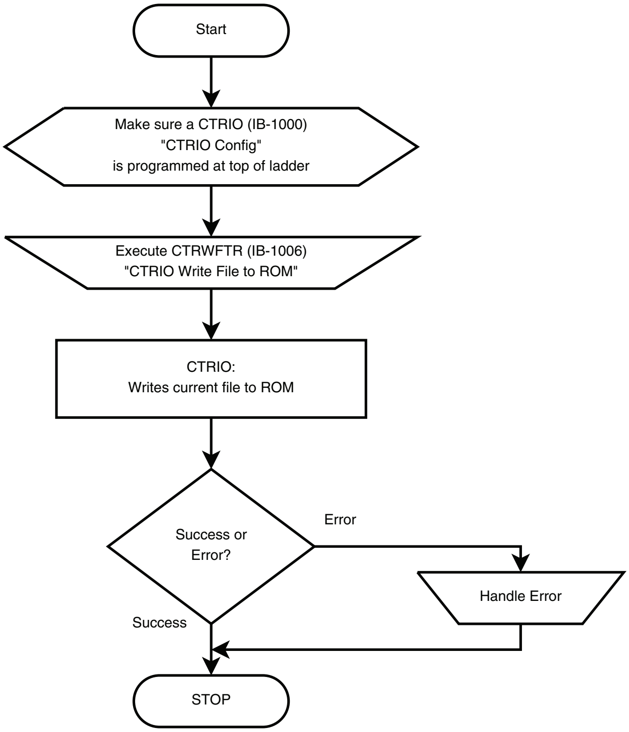
- Set the Command Code to 0x99.
- Set ProcessCmd ON to tell the CTRIO(2) to read in the CommandCode.
- The CTRIO(2) module begins the process of writing the file to ROM.

Once CmdComplete is seen high, your code should check the status of CmdError:

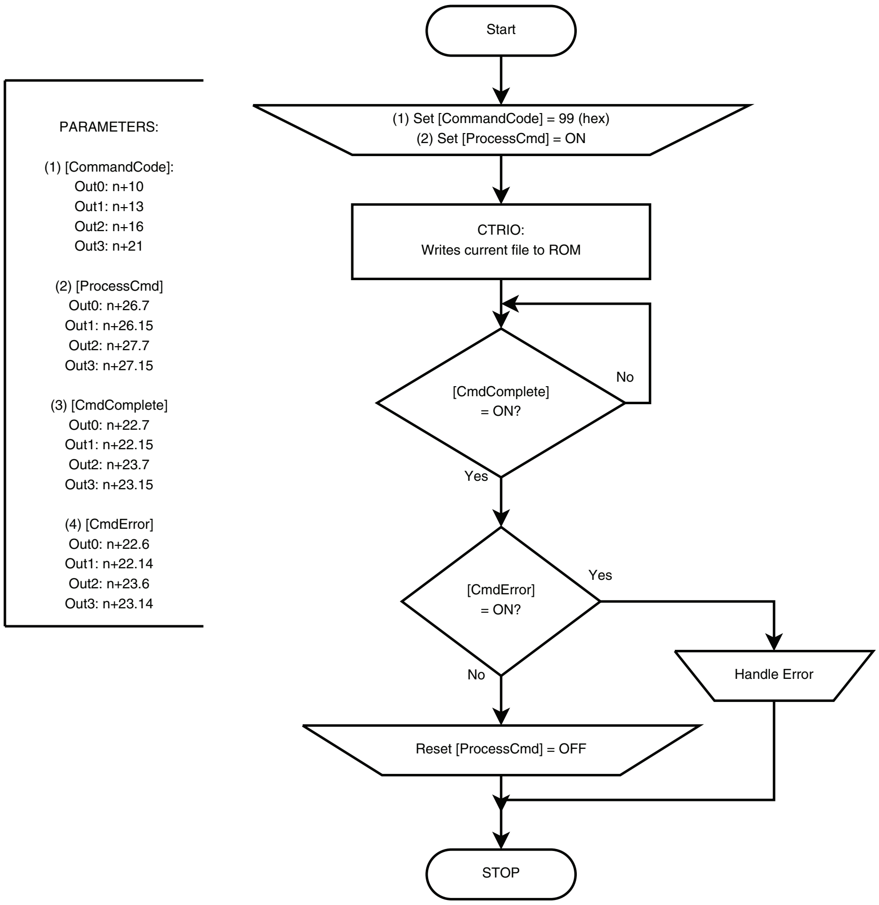
- If CmdError is ON, the CTRIO(2) module failed to write the file. Write code to handle the error and finish here.
- If CmdError is OFF, the file was written to ROM.
- Reset ProcessCmd OFF.

Write File to ROM (IBoxes)

NOTE: An example program using these IBoxes can be found in Chapter 5 of the DirectLOGIC user manuals for DL05, DL06 and DL205.


Start
Make sure a CTRIO (IB-1000) "CTRIO Config" is programmed at top of ladder
Execute CTRWFTR (IB-1006) "CTRIO Write File to ROM"
CTRIO: Writes current file to ROM
Success or Error?
Error
Handle Error
Success
STOP

Write File to ROM (DL-PLC)



Create Preset Table on Reset (Initialize Table on Reset)

Description

Initialize Table creates a Table with one entry in the Active Area for an Output. The lone entry is defined in Parameters 1-3. The new Table becomes active when Count is reset.

Prerequisites

- An Input configured for counting
- An Output configured to Discrete on Ch*/Fn*

Steps

Each step requires targeting the correct address for the desired Output. Check the Memory Map from your CTRIO(2) configuration. The following steps relate directly to the flow charts that follow.

- Set the Command Code to 0x16.
- Write a value to Parameter1 to define the entry type.

Parameter1	Function
0	Set
1	Reset
2	Pulse ON
3	Pulse OFF
4	Toggle OFF
5	Reset Count

If Parameter1 is either Pulse function (2 or 3), write a value to Parameter2 to specify Pulse Time in milliseconds. Otherwise, Parameter2 is ignored.

- Write a value to Parameter3 to specify the counts where the function should occur.
- Set ProcessCmd ON to tell the CTRIO(2) to read in the CommandCode.

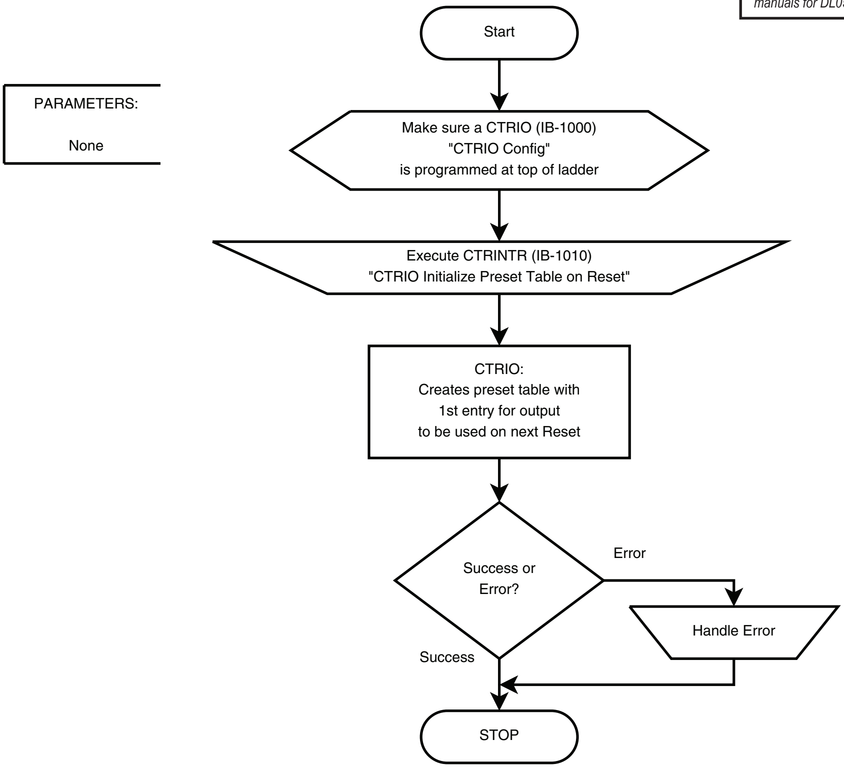
The CTRIO(2) module begins the process of creating the Table. When it's done, it sets the CmdComplete bit and will set or reset the CmdError bit accordingly.

Once CmdComplete is seen high, your code should check the status of CmdError:

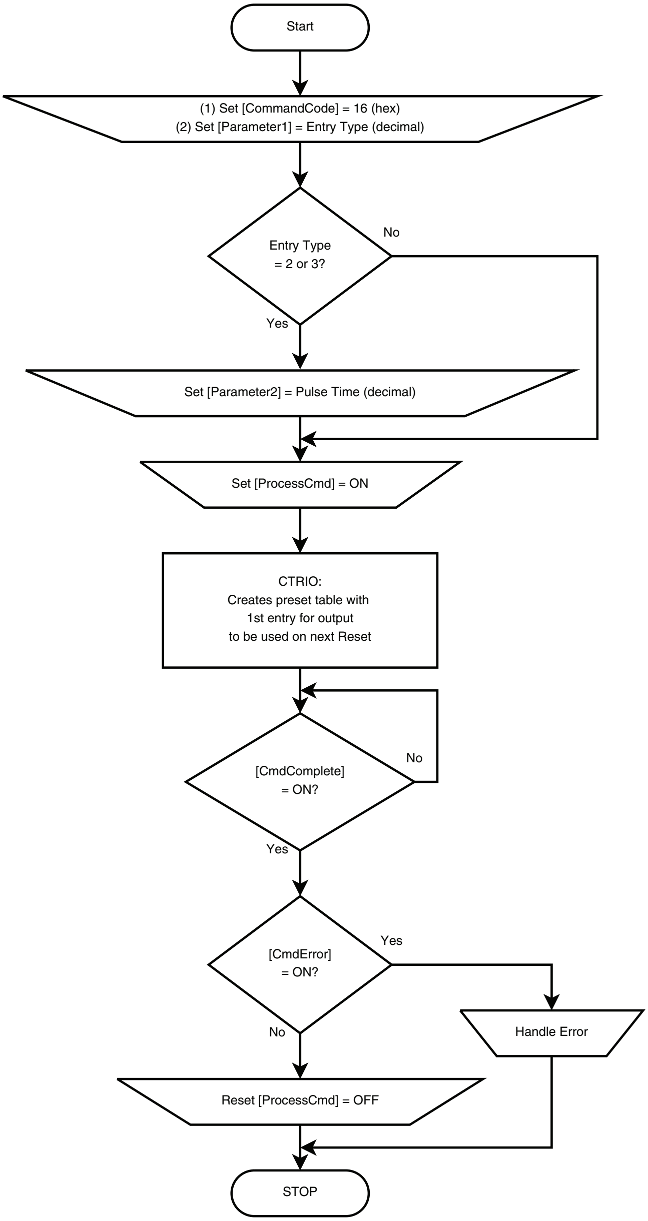
- If CmdError is ON, the CTRIO(2) module failed to create the Table. Write code to handle the error and finish here.
- If CmdError is OFF, the Table is ready for use.
- Reset ProcessCmd OFF.

Create Preset Table on Reset (IBoxes)

NOTE: *An example program using these IBoxes can be found in Chapter 5 of the DirectLOGIC user manuals for DL05, DL06 and DL205.*


PARAMETERS:
None
Start
Make sure a CTRIO (IB-1000) "CTRIO Config" is programmed at top of ladder
Execute CTRINTR (IB-1010) "CTRIO Initialize Preset Table on Reset"
CTRIO: Creates preset table with 1st entry for output to be used on next Reset
Success or Error?
Error
Handle Error
Success
STOP

Create Preset Table on Reset (DL-PLC)



PARAMETERS:

(1) [CommandCode]:
Out0: n+10
Out1: n+13
Out2: n+16
Out3: n+21

(2) [Parameter1]
Out0: n+11
Out1: n+14
Out2: n+17
Out3: n+22

(3) [Parameter2]
Out0: n+12
Out1: n+15
Out2: n+20
Out3: n+23

(4) [ProcessCmd]
Out0: n+26.7
Out1: n+26.15
Out2: n+27.7
Out3: n+27.15

(5) [CmdComplete]
Out0: n+22.7
Out1: n+22.15
Out2: n+23.7
Out3: n+23.15

(6) [CmdError]
Out0: n+22.6
Out1: n+22.14
Out2: n+23.6
Out3: n+23.14

KEY:

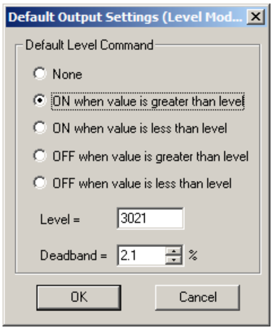
Entry Type:
0 = Set
1 = Reset
2 = Pulse ON
3 = Pulse OFF
4 = Toggle OFF
5 = Reset Count

Pulse Time:
Time in milliseconds

Update Level (Edit Level Response)

Description

When an Input is Scaled to provide Rate and an Output is set to Discrete on Ch*/Fn*, Update Level can be used to alter the response of the Output (to alter the Level Command set up in CTRIO Workbench).



Default Output Settings (Level Mod...

Default Level Command

○ None
● ON when value is greater than level
○ ON when value is less than level
○ OFF when value is greater than level
○ OFF when value is less than level

Level = 3021

Deadband = 2.1 %

Prerequisites

- An Input configured for counting and scaled to Rate
- An Output set to Discrete on Ch*/Fn*

Steps

Each step requires targeting the correct address for the desired Output. Check the Memory Map from your CTRIO(2) configuration. The following steps relate directly to the flow charts that follow.

- Set the Command Code to 0x30.
- Set Parameter 1 to the desired behavior in hex.
- Set Parameter 2 to the desired deadband in decimal, with one implied decimal place. For example, enter 50 for a deadband of 5.0%.
- Set Parameter 3 to the desired scaled level in decimal.
- Set ProcessCmd ON to tell the CTRIO(2) to read in the CommandCode and Parameters 1-3.

Parameter1	Function
00	ON > Level
80	ON < Level
01	OFF > Level
81	OFF < Level

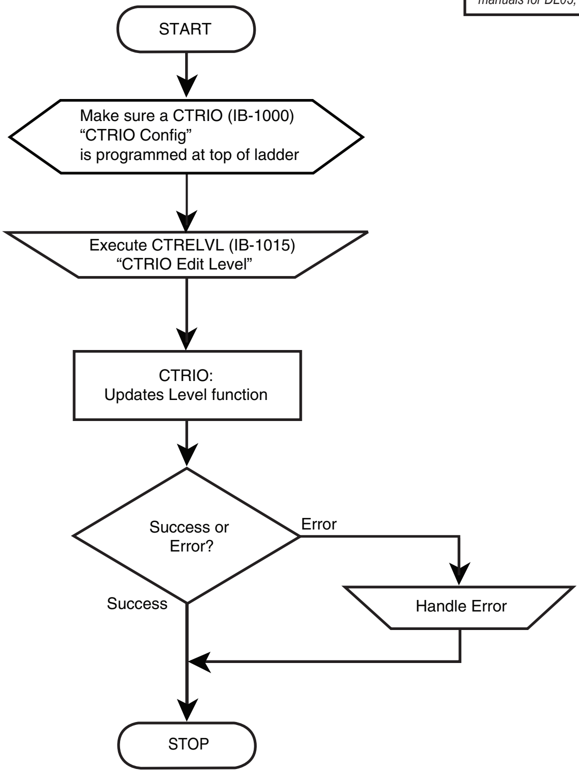
The CTRIO(2) module begins the process of changing the Level Response. When it's done, it sets the CmdComplete bit and will set or reset the CmdError bit accordingly.

Once CmdComplete is seen high, your code should check the status of CmdError:

- If CmdError is ON, the CTRIO(2) module failed to update the table entry. Write code to handle the error and finish here.
- If CmdError is OFF, the entry was updated successfully.
- Reset ProcessCmd OFF.

Update Level (IBoxes)

NOTE: *An example program using these IBoxes can be found in Chapter 5 of the DirectLOGIC user manuals for DL05, DL06 and DL205.*


START
Make sure a CTRIO (IB-1000) "CTRIO Config" is programmed at top of ladder
Execute CTRELVL (IB-1015) "CTRIO Edit Level"
CTRIO: Updates Level function
Success or Error?
Error
Success
Handle Error
STOP

Update Level
(DL-PLC)

PARAMETERS:

(1) [CommandCode]:
Out0: n+10
Out1: n+13
Out2: n+16
Out3: n+21

(2) [Function]
Out0: n+11
Out1: n+14
Out2: n+17
Out3: n+22

(3) [Deadband]
Out0: n+12
Out1: n+15
Out2: n+20
Out3: n+23

(4) [ScaledLevel]
Out0: n+0-1
Out1: n+2-3
Out2: n+4-5
Out3: n+6-7

(5) [ProcessCmd]
Out0: n+26.7
Out1: n+26.15
Out2: n+27.7
Out3: n+27.15

(6) [CmdComplete]
Out0: n+22.7
Out1: n+22.15
Out2: n+23.7
Out3: n+23.15

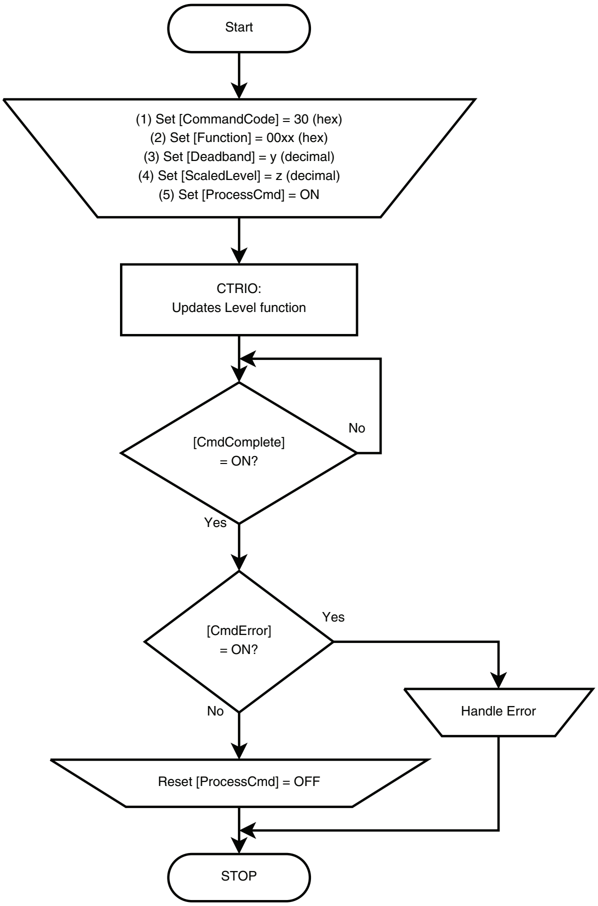
(7) [CmdError]
Out0: n+22.6
Out1: n+22.14
Out2: n+23.6
Out3: n+23.14

KEY:

00xx:
0000 = ON > Level
0080 = ON < Level
0001 = OFF > Level
0081 = OFF < Level

y:
1 implied decimal
i.e. 100 = 10.0%

z:
Rate/Intverval


Start
(1) Set [CommandCode] = 30 (hex)
(2) Set [Function] = 00xx (hex)
(3) Set [Deadband] = y (decimal)
(4) Set [ScaledLevel] = z (decimal)
(5) Set [ProcessCmd] = ON
CTRIO:
Updates Level function
[CmdComplete]
= ON?
No
Yes
[CmdError]
= ON?
Yes
No
Handle Error
Reset [ProcessCmd] = OFF
STOP