# Do-more Programming Examples

**In This Chapter...**

Do-more Programming Examples Overview . . . . . .9-2
Load and Run a Pulse Profile . . . . . .9-3
Dynamic Positioning/Positioning Plus . . . . . .9-4
Dynamic Velocity . . . . . .9-5
Velocity Mode . . . . . .9-6
Run to Limit Mode . . . . . .9-7
Run to Position Mode . . . . . .9-8
System Functions Examples Overview . . . . . .9-9
Simulating Retentive Counter . . . . . .9-9
Reading CTRIO Internal Registers . . . . . .9-10

# Do-more Programming Examples Overview

***NOTE:*** *The programming examples on the following pages are provided "as is" without a guarantee of any kind. This Chapter is provided by our technical support group to assist others. We do not guarantee the examples are suitable for a particular application, nor do we assume any responsibility for them in your application.*

The Do-more Designer programming examples provided on the following pages are simple examples that are intended to assist you in the basics of loading and running various output pulse profiles. The examples are complete enough to load a profile, process the command and load the Parameter registers necessary to execute the profile. Two System Functions examples are also provided.

**Load and Run a Pulse Profile example:**
You will need to have a Trapezoid, S-Curve, Symmetrical S-Curve, Home Search or Free Form profile configured using the Configure I/O dialog.

**Dynamic Positioning/Positioning Plus Profile example:**
You will need to have a Dynamic Positioning/Positioning Plus profile configured as Table File Number 1 using the Configure I/O dialog.

**Dynamic Velocity Profile example:**
You will need to have a Dynamic Velocity profile configured as Table File Number 1 using the Configure I/O dialog.

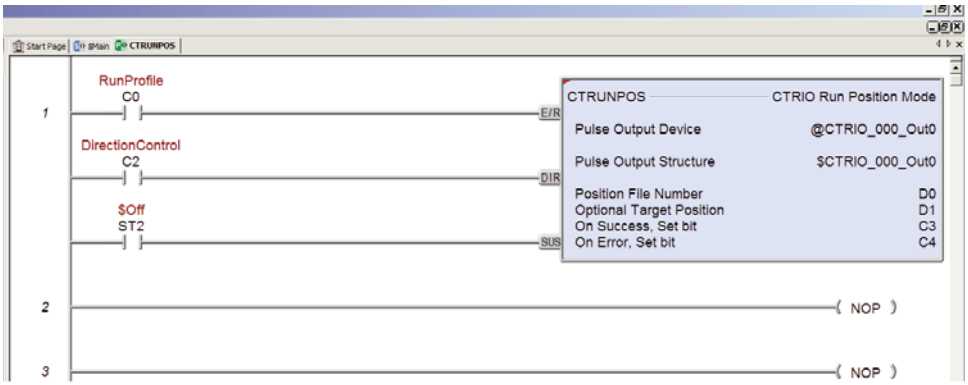
**Velocity Mode, Run to Limit Mode and Run to Position Mode examples:**
No CTRIO Pulse Profile Tables are necessary to execute these profiles, but the Outputs need to be configured for Step/Direction or CW/CCW using the Configure I/O dialog. All parameters are stored in memory as shown in the examples. For Run to Position Mode, Ch1 Inputs A & B must be configured for Quad Counter in the Configure I/O dialog. The position from that encoder is used to stop the move.

**Simulating Retentive Counter example:**
This Systems Functions example uses the Write to Intelligent (WT) instruction to write the current count stored in the PLC's retentive memory to the CTRIO's current count register on a power cycle or a RUN-STOP-RUN PLC mode change. No permissive bits are required to be turned on in this example.

**Reading CTRIO Internal Registers example:**
This 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. You must turn C0 on to initialize the settings to perform the Read routine.

# Load and Run a Pulse Profile

Make sure D0 of the instruction (Position File Number) has the correct file number, choose the direction with C2 and turn on C0 to begin the move. (Optional Target Position is ignored by the instruction when used with a fixed profile. This field is the equivalent of Parameter 3 for other profiles in DirectLOGIC).

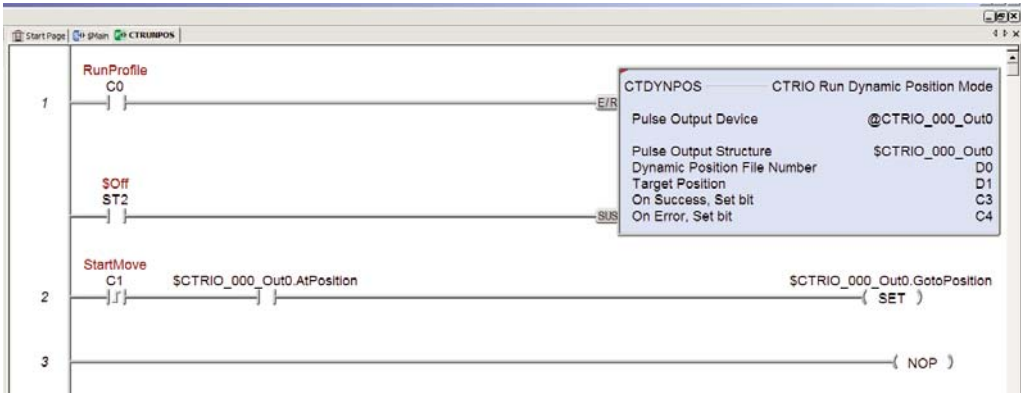



For detailed information on the CTRUNPOS – CTRIO Run Position Mode instruction, see Do-more Designer help file topic DMD0529.

# Dynamic Positioning/Positioning Plus

Make sure D0 of the instruction (Dynamic Position File Number) has the correct file number. Set the target position in D1 and turn on C0 to load the profile and zero out the position register, $CTRIO_000_Out0.OutputPosition. Turn on C1 to start the move. Pulses are generated to move toward the position specified in D1. The structure member, .AtPosition, turns on when the move completes.

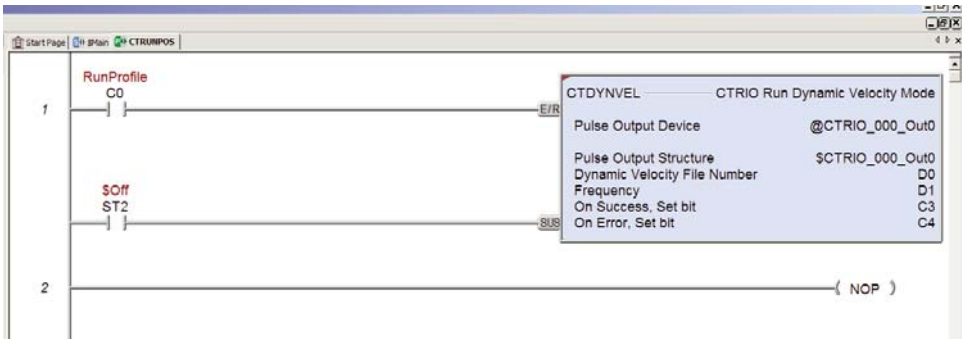
After a move completes, load the next position into D1 and turn on C1 to start another move.




For detailed information on the CTDYNPOS – CTRIO Run Dynamic Position Mode instruction, see Do-more Designer help file topic DMD0516.

# Dynamic Velocity

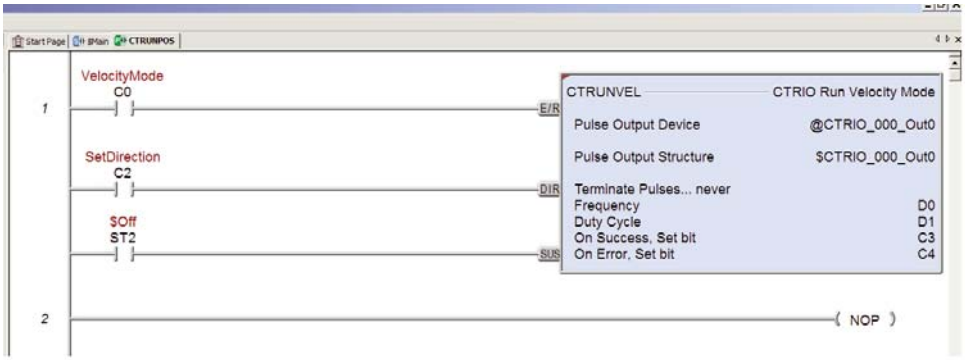
Make sure D0 of the instruction (Dynamic Velocity File Number) has the correct file number. Set the target velocity in D1 and turn on C0 to load the profile. The sign of the value in D1 determines the pulse output direction. The velocity can be changed 'on the fly' by placing a new value in D1.




For detailed information on the CTDYNVEL – CTRIO Run Dynamic Velocity Mode instruction, see Do-more Designer help file topic DMD0517.

# Velocity Mode

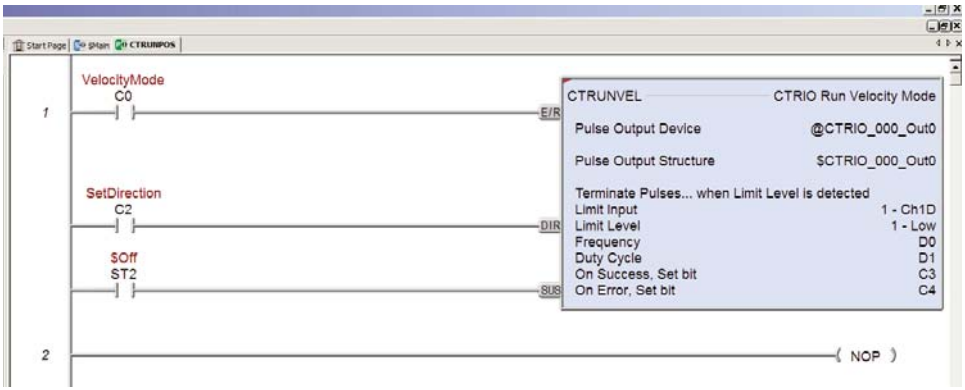
Load the desired frequency into D0 and select the desired direction using C2. Turn on C0 to start the output. The velocity can be changed 'on the fly' by placing a new value in D1. The direction is latched in when the instruction is enabled. To change direction, the instruction must first be disabled (C0 OFF).




For detailed information on the CTRUNVEL – CTRIO Run Velocity Mode instruction, see Do-more Designer help file topic DMD0530.

# Run to Limit Mode

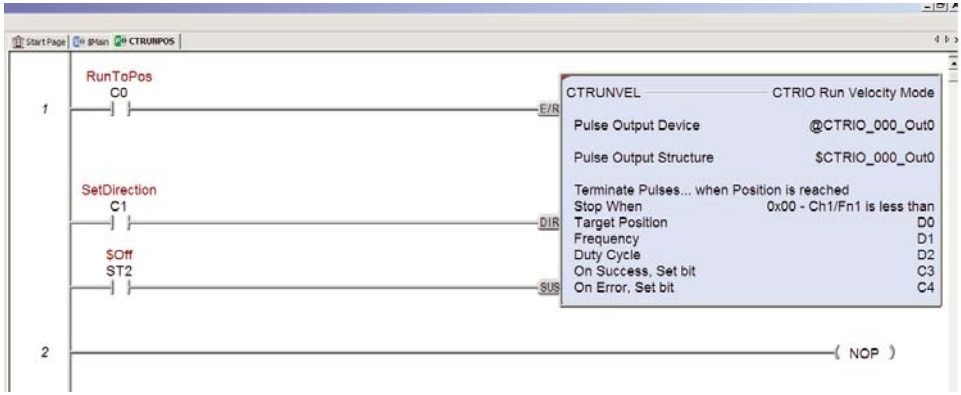
Choose the "Terminate Pulses when Limit Level is detected" option when selecting the CTRUNVEL instruction shown below. Load the desired frequency into D0 and select the desired direction using C2. Turn on C0 to start the output. When Ch1D meets the limit condition, ('Low' in this example) the move stops.




For detailed information on the CTRUNVEL – CTRIO Run Velocity Mode instruction, see Do-more Designer help file topic DMD0530.

## Run to Position Mode

This is the equivalent ladder in Do-more Designer to the ladder shown on the previous two pages. Choose the "Terminate Pulses when Position is reached" option and select the desired stop conditions when programming the CTRUNVEL instruction shown below. As configured by default, the move completes when the counts on an encoder connected to Ch1 are less than the value in D0. Load the desired position into D0, the desired frequency into D1 and select the direction using C2. Turn on C0 to start the move.




For detailed information on the CTRUNVEL – CTRIO Run Velocity Mode instruction, see Do-more Designer help file topic DMD0530.

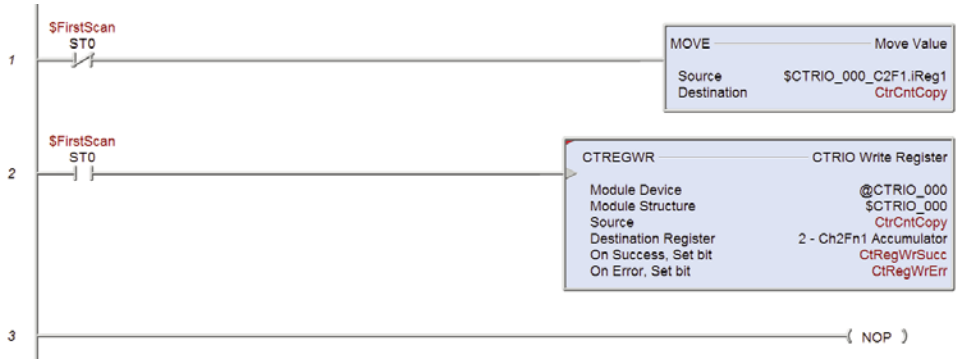
# System Functions Examples Overview

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

The Systems Functions examples on the following pages use the Do-more Designer's CTREGWR – CTRIO Write Register and CTREGRD - CTRIO Read Register instructions to write to or read from the CTRIO's internal registers.

# Simulating Retentive Counter

The following example uses the CTREGWR – CTRIO Write Register instruction to write the current count stored in the PLC's retentive memory to the CTRIO's current count register after a power cycle or a RUN-STOP-RUN PLC mode change.




Rung 1 copies the CTRIO's counts value from channel 2 into a retentive memory location on every scan, except the First Scan.

***NOTE:*** *This example uses the .iReg1 member, which contains integer raw counts when no scaling is applied in the CTRIO module configuration. If scaling WERE applied for this CTRIO function, .iReg1 would contain the scaled value and .iReg2 would contain the raw counts. CTREGWR only writes to the raw counts accumulator, so use .iReg2 if this CTRIO function is scaled in your configuration.*

Rung 2 uses the CTREGWR instruction to write the retained value back into the CTRIO's internal register on the First Scan.

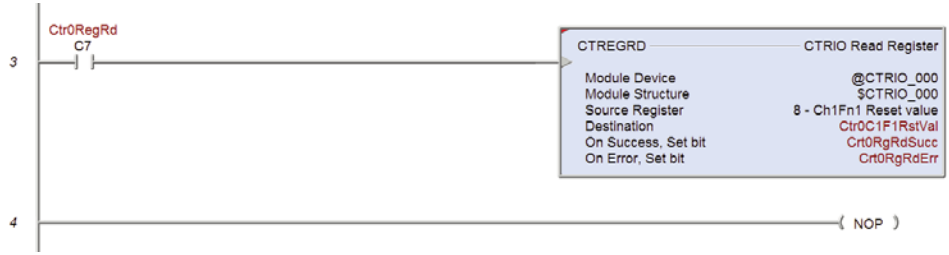
***NOTE:*** *Since CTREGWR is an asynchronous instruction (note the red triangle), execution of any other ladder that references the current count of this CTRIO needs to be held off until this CTREGWR instruction returns its On Success bit, or its On Success JMP to Stage.*

For detailed information on the CTREGWR – CTRIO Write Register instruction, see Do-more Designer help file topic DMD0527.

# Reading CTRIO Internal Registers

For most applications, reading CTRIO Internal Registers using ladder is not necessary. All the data needed from the module is already made available in the Do-more CPU through the CTRIO module structure. For a list of CTRIO module structures, see Help File topic: DMD0313 High Speed I/O Applications.

There are some registers in the CTRIO whose values are not available in structures. Examples are the Reset Value for a resettable counter and Filter Time for inputs. To read these values into ladder, use the CTREGRD - CTRIO Read Register instruction as shown below.



***NOTE::*** *The CTREGRD instruction can also be used to read the accumulator for a counter channel. However, this value is already available in structure elements .iReg1, .iReg2, .fReg1, or .fReg2. The only time CTREGRD would need to be used to read the accumulator is if scaling and capture are enabled and the raw accumulator value is needed in ladder.*

In the sample ladder above, turn on C7 to read in the Channel1 Function1 Reset Value of the CTRIO module in slot 0. The value will be written into address nicknamed Ctr0C1F1RstVal.

For detailed information on the CTREGRD – CTRIO Read Register instruction, see Do-more Designer help file topic DMD0526.