



**CTRIO
High-Speed Counter
Module**

Manual Number: HX-CTRIO-M

⚡ WARNING ⚡

Thank you for purchasing automation equipment from **AutomationDirect.com**[®], doing business as, **AutomationDirect**. We want your new automation equipment to operate safely. Anyone who installs or uses this equipment should read this publication (and any other relevant publications) before installing or operating the equipment.

To minimize the risk of potential safety problems, you should follow all applicable local and national codes that regulate the installation and operation of your equipment. These codes vary from area to area and usually change with time. It is your responsibility to determine which codes should be followed, and to verify that the equipment, installation, and operation is in compliance with the latest revision of these codes.

At a minimum, you should follow all applicable sections of the National Fire Code, National Electrical Code, and the codes of the National Electrical Manufacturer's Association (NEMA). There may be local regulatory or government offices that can also help determine which codes and standards are necessary for safe installation and operation.

Equipment damage or serious injury to personnel can result from the failure to follow all applicable codes and standards. We do not guarantee the products described in this publication are suitable for your particular application, nor do we assume any responsibility for your product design, installation, or operation.

Our products are not fault-tolerant and are not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the product could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). **AutomationDirect** specifically disclaims any expressed or implied warranty of fitness for High Risk Activities.

For additional warranty and safety information, see the Terms and Conditions section of our catalog. If you have any questions concerning the installation or operation of this equipment, or if you need additional information, please call us at 770-844-4200.

This publication is based on information that was available at the time it was printed. At **AutomationDirect** we constantly strive to improve our products and services, so we reserve the right to make changes to the products and/or publications at any time without notice and without any obligation. This publication may also discuss features that may not be available in certain revisions of the product.

Trademarks

This publication may contain references to products produced and/or offered by other companies. The product and company names may be trademarked and are the sole property of their respective owners. **AutomationDirect** disclaims any proprietary interest in the marks and names of others.

**Copyright 2018, AutomationDirect.com[®] Incorporated
All Rights Reserved**

No part of this manual shall be copied, reproduced, or transmitted in any way without the prior, written consent of **AutomationDirect.com[®] Incorporated**. **AutomationDirect** retains the exclusive rights to all information included in this document.

⚡ ADVERTENCIA ⚡

Gracias por comprar equipo de automatización de **Automationdirect.com**[®]. Deseamos que su nuevo equipo de automatización opere de manera segura. Cualquier persona que instale o use este equipo debe leer esta publicación (y cualquier otra publicación pertinente) antes de instalar u operar el equipo.

Para reducir al mínimo el riesgo debido a problemas de seguridad, debe seguir todos los códigos de seguridad locales o nacionales aplicables que regulan la instalación y operación de su equipo. Estos códigos varían de área en área y usualmente cambian con el tiempo. Es su responsabilidad determinar cuales códigos deben ser seguidos y verificar que el equipo, instalación y operación estén en cumplimiento con la revisión más reciente de estos códigos.

Como mínimo, debe seguir las secciones aplicables del Código Nacional de Incendio, Código Nacional Eléctrico, y los códigos de (NEMA) la Asociación Nacional de Fabricantes Eléctricos de USA. Puede haber oficinas de normas locales o del gobierno que pueden ayudar a determinar cuales códigos y normas son necesarios para una instalación y operación segura.

Si no se siguen todos los códigos y normas aplicables, puede resultar en daños al equipo o lesiones serias a personas. No garantizamos los productos descritos en esta publicación para ser adecuados para su aplicación en particular, ni asumimos ninguna responsabilidad por el diseño de su producto, la instalación u operación.

Nuestros productos no son tolerantes a fallas y no han sido diseñados, fabricados o intencionados para uso o reventa como equipo de control en línea en ambientes peligrosos que requieren una ejecución sin fallas, tales como operación en instalaciones nucleares, sistemas de navegación aérea, o de comunicación, control de tráfico aéreo, máquinas de soporte de vida o sistemas de armamentos en las cuales la falla del producto puede resultar directamente en muerte, heridas personales, o daños físicos o ambientales severos (“Actividades de Alto Riesgo”). **Automationdirect.com** específicamente rechaza cualquier garantía ya sea expresada o implicada para actividades de alto riesgo.

Para información adicional acerca de garantía e información de seguridad, vea la sección de Términos y Condiciones de nuestro catálogo. Si tiene alguna pregunta sobre instalación u operación de este equipo, o si necesita información adicional, por favor llámenos al número 770-844-4200 en Estados Unidos.

Esta publicación está basada en la información disponible al momento de impresión. En **Automationdirect.com** nos esforzamos constantemente para mejorar nuestros productos y servicios, así que nos reservamos el derecho de hacer cambios al producto y/o a las publicaciones en cualquier momento sin notificación y sin ninguna obligación. Esta publicación también puede discutir características que no estén disponibles en ciertas revisiones del producto.

Marcas Registradas

Esta publicación puede contener referencias a productos producidos y/u ofrecidos por otras compañías. Los nombres de las compañías y productos pueden tener marcas registradas y son propiedad única de sus respectivos dueños. **Automationdirect.com**, renuncia cualquier interés propietario en las marcas y nombres de otros.

PROPIEDAD LITERARIA 2018, **AUTOMATIONDIRECT.COM**[®] INCORPORATED

Todos los derechos reservados

No se permite copiar, reproducir, o transmitir de ninguna forma ninguna parte de este manual sin previo consentimiento por escrito de **Automationdirect.com**[®] Incorporated. **Automationdirect.com** retiene los derechos exclusivos a toda la información incluida en este documento. Los usuarios de este equipo pueden copiar este documento solamente para instalar, configurar y mantener el equipo correspondiente. También las instituciones de enseñanza pueden usar este manual para propósitos educativos.

⚡ AVERTISSEMENT ⚡

Nous vous remercions d'avoir acheté l'équipement d'automatisation de **Automationdirect.com**[®], en faisant des affaires comme, **AutomationDirect**. Nous tenons à ce que votre nouvel équipement d'automatisation fonctionne en toute sécurité. Toute personne qui installe ou utilise cet équipement doit lire la présente publication (et toutes les autres publications pertinentes) avant de l'installer ou de l'utiliser.

Afin de réduire au minimum le risque d'éventuels problèmes de sécurité, vous devez respecter tous les codes locaux et nationaux applicables régissant l'installation et le fonctionnement de votre équipement. Ces codes diffèrent d'une région à l'autre et, habituellement, évoluent au fil du temps. Il vous incombe de déterminer les codes à respecter et de vous assurer que l'équipement, l'installation et le fonctionnement sont conformes aux exigences de la version la plus récente de ces codes.

Vous devez, à tout le moins, respecter toutes les sections applicables du Code national de prévention des incendies, du Code national de l'électricité et des codes de la National Electrical Manufacturer's Association (NEMA). Des organismes de réglementation ou des services gouvernementaux locaux peuvent également vous aider à déterminer les codes ainsi que les normes à respecter pour assurer une installation et un fonctionnement sûrs.

L'omission de respecter la totalité des codes et des normes applicables peut entraîner des dommages à l'équipement ou causer de graves blessures au personnel. Nous ne garantissons pas que les produits décrits dans cette publication conviennent à votre application particulière et nous n'assumons aucune responsabilité à l'égard de la conception, de l'installation ou du fonctionnement de votre produit.

Nos produits ne sont pas insensibles aux défaillances et ne sont ni conçus ni fabriqués pour l'utilisation ou la revente en tant qu'équipement de commande en ligne dans des environnements dangereux nécessitant une sécurité absolue, par exemple, l'exploitation d'installations nucléaires, les systèmes de navigation aérienne ou de communication, le contrôle de la circulation aérienne, les équipements de survie ou les systèmes d'armes, pour lesquels la défaillance du produit peut provoquer la mort, des blessures corporelles ou de graves dommages matériels ou environnementaux («activités à risque élevé»). La société **AutomationDirect** nie toute garantie expresse ou implicite d'aptitude à l'emploi en ce qui a trait aux activités à risque élevé.

Pour des renseignements additionnels touchant la garantie et la sécurité, veuillez consulter la section Modalités et conditions de notre documentation. Si vous avez des questions au sujet de l'installation ou du fonctionnement de cet équipement, ou encore si vous avez besoin de renseignements supplémentaires, n'hésitez pas à nous téléphoner au 770-844-4200.

Cette publication s'appuie sur l'information qui était disponible au moment de l'impression. À la société **AutomationDirect**, nous nous efforçons constamment d'améliorer nos produits et services. C'est pourquoi nous nous réservons le droit d'apporter des modifications aux produits ou aux publications en tout temps, sans préavis ni quelque obligation que ce soit. La présente publication peut aussi porter sur des caractéristiques susceptibles de ne pas être offertes dans certaines versions révisées du produit.

Marques de commerce

La présente publication peut contenir des références à des produits fabriqués ou offerts par d'autres entreprises. Les désignations des produits et des entreprises peuvent être des marques de commerce et appartiennent exclusivement à leurs propriétaires respectifs. **AutomationDirect** nie tout intérêt dans les autres marques et désignations.

Copyright 2018, **Automationdirect.com**[®] Incorporated
Tous droits réservés

Nulle partie de ce manuel ne doit être copiée, reproduite ou transmise de quelque façon que ce soit sans le consentement préalable écrit de la société **Automationdirect.com**[®] Incorporated. **AutomationDirect** conserve les droits exclusifs à l'égard de tous les renseignements contenus dans le présent document.

CTRIO HIGH-SPEED COUNTER USER MANUAL



Please include the Manual Number and the Manual Issue, both shown below, when communicating with Technical Support regarding this publication.

Manual Number: HX-CTRIO-M
Issue: Third Edition, Rev. B
Issue Date: 6/19

Publication History		
Issue	Date	Description of Changes
First Edition	9/01	Original
Rev. A	10/01	Corrections
Rev. B	8/02	Corrections
Second Edition	2/03	Added T1H-CTRIO and H4-CTRIO. Updated for CTRIO/Workbench version 2.
Rev. A		Added H0-CTRIO and flowcharts.
Rev. B	10/03	Corrections
Rev. C	03/11	Made corrections and updated manual.
Rev. D	2/13	Updated manual with new H0-CTRIO2 and H2-CTRIO2 information. Added Do-more PLC series data and examples. Updated CTRIO Workbench section with new pulse profiles available. Made minor corrections throughout manual.
Rev. E	09/13	Added Do-more structure table to Chapter 6. Added Appendix A, Appendix B and Appendix C which contain flowcharts for various instructions Made minor corrections throughout manual.
Third Edition	06/16	Completely reworked and reorganized manual.
Rev. A	11/18	Revised and added material to Appendix B.
Rev. B	6/19	Revised and added material to Chapter 9, pp 28-34.

TABLE OF CONTENTS



Chapter 1 - Introduction to the CTRIO & CTRIO2 Modules

Introduction.....	1-2
Conventions Used.....	1-3
CTRIO and CTRIO2 Module Overview	1-4
Support Systems for the CTRIO(2) Modules	1-6
CTRIO(2) Specifications.....	1-7
H0-CTRIO(2) LED Indicators.....	1-10
H2-CTRIO(2) LED Indicators.....	1-11
H4-CTRIO LED Indicators.....	1-12
T1H-CTRIO LED Indicators.....	1-13
Overview, How it Works as Part of the Control System	1-14
CTRIO(2) Module Work Flow Diagram.....	1-17

Chapter 2 - Getting Started, Basics and Examples

Overview	2-2
Basic Motion Functions, Summary of Examples	2-2
Detailed Example: Configure and Test a Quadrature Input	2-7
Detailed Example: Configure and Test a Pulse Output w/Trapezoidal Profile	2-16
Flow Chart Example: Configure and Test a Pulse Output w/Trapezoidal Profile	2-27

Chapter 3 - Installation and Field Wiring

Installing the H0-CTRIO(2) Module	3-2
Setting H0-CTRIO(2) Jumpers	3-3

H0- CTRIO(2) Quadrature Encoder Wiring Example	3-5
H0-CTRIO(2) TTL Quadrature Encoder Field Wiring	3-6
H0- CTRIO(2) TTL Input Wiring	3-7
H0- CTRIO(2) Output Wiring Schematic.....	3-8
H0-CTRIO(2) Stepper/Servo Drive Wiring Example.....	3-9
Solid State Input Device Wiring to the H0-CTRIO(2) Module.....	3-10
Installing the H2-CTRIO(2) Module	3-11
Setting H2-CTRIO(2) Jumpers	3-12
Wiring the H2-CTRIO(2) Module	3-13
H2- CTRIO(2) Quadrature Encoder Wiring Example.....	3-14
H2-CTRIO(2) TTL Quadrature Encoder Field Wiring	3-15
H2-CTRIO(2) TTL Input Wiring	3-16
H2- CTRIO(2) Output Wiring Schematic.....	3-17
H2-CTRIO(2) Stepper/Servo Drive Wiring Example.....	3-18
Solid State Input Device Wiring to the H2-CTRIO(2) Module.....	3-19
Installing the H4-CTRIO.....	3-20
Wiring the H4-CTRIO Module	3-21
H4-CTRIO Quadrature Encoder Wiring Example.....	3-22
H4-CTRIO TTL Quadrature Encoder Field Wiring.....	3-23
H4-CTRIO TTL Input Wiring	3-24
H4-CTRIO Output Wiring Schematic.....	3-25
H4-CTRIO Stepper/Servo Drive Wiring Example.....	3-26
Solid State Input Device Wiring to the H4-CTRIO Module.....	3-27
Installing the T1H-CTRIO.....	3-28
Wiring the T1H-CTRIO Module	3-29
T1H-CTRIO Quadrature Encoder Wiring Example.....	3-31
T1H-CTRIO TTL Quadrature Encoder Field Wiring.....	3-32
T1H-CTRIO TTL Input Wiring	3-33
T1H-CTRIO Output Wiring Schematic	3-34

T1H-CTRIO Stepper/Servo Drive Wiring Example.....	3-35
Solid State Input Device Wiring to T1H-CTRIO Module	3-36
Chapter 4 - CTRIO Workbench, Overview	
Configuring a CTRIO Module for Do-more CPUs.....	4-2
What is CTRIO Workbench?.....	4-2
Getting Started with CTRIO Workbench.....	4-3
Module Modes of Operation.....	4-6
Chapter 5 - CTRIO Workbench, Configuring Inputs	
Configure I/O Dialog Overview	5-2
Input Function Selections.....	5-3
Counter Function.....	5-4
Pulse Catch	5-6
Edge Timer.....	5-7
Dual Edge Timer	5-8
Reset FN1 and Reset FN2 (Hard Resets for Counters Only)	5-10
Soft Resets	5-10
Capture FN1.....	5-11
Inhibit FN1	5-11
Limit Out.....	5-11
Introduction to the Scaling Wizard	5-12
Chapter 6 - CTRIO Workbench Configuring Outputs	
Configure I/O Dialog Overview	6-2
Output Function Selections.....	6-3
Raw Output.....	6-4
Discrete Outputs.....	6-5
Pulse Outputs.....	6-11

Chapter 7 - CTRIO Workbench, I/O Map

I/O Map Dialog.....	7-2
---------------------	-----

Chapter 8 - CTRIO Workbench, Monitor I/O

Using the Monitor I/O Dialog.....	8-2
Monitor I/O Error Codes	8-7

Chapter 9 - Output Functions

Runtime Changes to CTRIO Configured Preset Tables (DL PLCs)	9-3
Pulse Output Profiles (DL PLCs).....	9-6
Trapezoid Profile	9-7
S-Curve Profile	9-8
Symmetrical S-Curve Profile.....	9-9
Home Search Profile.....	9-10
Free Form Profile	9-13
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-14
DirectLOGIC Programming Examples Overview	9-20
Trapezoid with Limits Profile	9-21
Trapezoid with Limits (CTRIO2) Profile.....	9-22
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-25
Trapezoid Plus (CTRIO2) Profile	9-29
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-31
Load and Run a Trapezoid Plus Profile	9-34
Dynamic Positioning Plus and Dynamic Positioning Profiles.....	9-35
Dynamic Positioning Plus (CTRIO2) Profile	9-36
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-37
Dynamic Velocity Profile	9-43
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-44
Velocity Mode	9-49
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-50

Run to Limit Mode	9-55
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-56
Run to Position Mode	9-62
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-63
Run to Position Mode with DirectSOFT IBox Instructions	9-68
Raw Output.....	9-70
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-71

Chapter 10 - Input Functions

Input Memory Mapping for Counter Data Transfer	10-2
Input Memory Map for Scaled Counter Data	10-5
Input Memory Map for Capture Count Data Transfers	10-9
Input Memory Map for Edge Timer and Dual Edge Timer.....	10-12
Input Memory Map for Pulse Catch Data Transfers	10-25

Chapter 11 - Runtime Table Functions

Introduction to Runtime Table Functions.....	11-2
Preset Tables and Programmable Limit Switch (PLS) Tables	11-4
Load Preset Table	11-4
Load Table	11-8
Clear Preset Table	11-11
Create Preset Table (Initialize Table)	11-14
Add Entry to Preset Table	11-17
CTRIO2 - <i>Direct</i> LOGIC Using IBoxes Flowcharts.....	11-18
CTRIO2 - <i>Direct</i> LOGIC PLC Flowcharts	11-19
Edit Preset Table Entry	11-20
Edit Preset Table Entry and Reload.....	11-23
Write File to ROM.....	11-26
Create Preset Table on Reset (Initialize Table on Reset).....	11-29
Update Level (Edit Level Response).....	11-32

Appendix A - Memory Mapping

Input Memory Map for Data Transfers from CTRIO(2) to DL CPUs.....	A-2
Output Memory Map for Data Transfers from DL CPUs to CTRIO(2).....	A-4
Addressing Conventions (with V-memory Examples for <i>Direct</i> LOGIC PLCs).....	A-7
Input Function Status/Control Bits and Parameters	A-8

Appendix B - System Functions

System Functions.....	B-2
Write All Registers (IBoxes).....	B-3
Write All Registers (DL-PLC)	B-4
Write One Register (IBoxes)	B-5
Write One Register (DL-PLC)	B-6
Read All Registers (IBoxes).....	B-7
Read All Registers (DL-PLC)	B-8
Read One Register (IBoxes)	B-9
Read One Register (DL-PLC)	B-10
Read Error Code (IBoxes).....	B-11
Read Error Code (DL-PLC).....	B-12
System Functions Examples Overview	B-13
Single Channel Simulating Retentive Quad Counter	B-14
Dual Channel Simulating Retentive Quad Counters	B-17
Reading CTRIO Internal Registers.....	B-20

INTRODUCTION TO THE CTRIO & CTRIO2 MODULES



In This Chapter...

Introduction.....	1-2
Conventions Used.....	1-3
CTRIO and CTRIO2 Module Overview	1-4
Support Systems for the CTRIO(2) Modules	1-6
CTRIO(2) Specifications.....	1-7
CH0-CTRIO(2) LED Indicators	1-10
H2-CTRIO(2) LED Indicators.....	1-11
H4-CTRIO LED Indicators.....	1-12
T1H-CTRIO LED Indicators.....	1-13
Overview, How it Works as Part of the Control System.....	1-14
CTRIO(2) Module Work Flow Diagram.....	1-17

Introduction

The Purpose of this Manual

This manual is intended as a help for the user to install, program, and maintain the CTRIO(2) module in their system. This manual contains important information for personnel who will install the CTRIO(2) high-speed counter module as well as for the PLC programmer. This manual will provide all the information needed for the novice and seasoned automation professional alike to start and keep your system up and running.

Online Help Files and Other Documentation

Regardless of the platform you are using, the programming software needed for the CTRIO(2) modules is available as a download from our website.

<http://www.aboutplcs.com/>

Each programming software includes searchable online help topics covering all aspects of the software, instruction set, module setup, and communications.

Technical Support

We strive to make our manuals the best in the industry. We rely on your feedback to let us know if we are reaching our goal. If you cannot find the solution to your particular application, or, if for any reason you need technical assistance, please call us at:

1-770-844-4200

Our technical support group will work with you to answer your questions. They are available Monday through Friday from 9:00 A.M. to 6:00 P.M. Eastern Time. We also encourage you to visit our web site where you can find information about our company and specific technical information about a wide array of our products, **www.automationdirect.com**.

Conventions Used



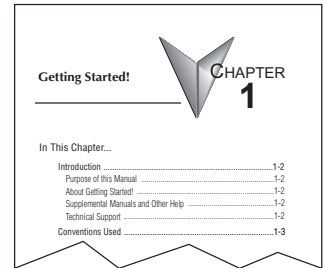
NOTE: When you see the “note pad” icon in the left-hand margin, the paragraph to its immediate right will be a special note. Notes represent information that may make your work quicker or more efficient. The word **NOTE** in boldface will mark the beginning of the text.



WARNING! When you see the “exclamation point” icon in the left-hand margin, the paragraph to its immediate right will be a warning. This information could prevent injury, loss of property, or even death in extreme cases. Any warning in this manual should be regarded as critical information that should be read in its entirety. The word **WARNING** in boldface will mark the beginning of the text.

Key Topics for Each Chapter

The beginning of each chapter will list the key topics that can be found in that chapter.



Icons

CTRIO(2) modules are available for several hardware platforms, including DL05/06, DL205, DL405 and Terminator. Among these four hardware platforms, there are a variety of CPUs that can occupy the CPU slot, including, Do-more, *Direct*LOGIC or WinPLC. There are also several communication interface modules that could occupy the CPU slot, including EBCs, Modbus, DeviceNet or Profibus.

Throughout this manual, a set of icons (on right) is used to designate which hardware platform a topic applies to, based on what module is in the CPU slot.

- DM**
- DL**
- Win**
- NI**

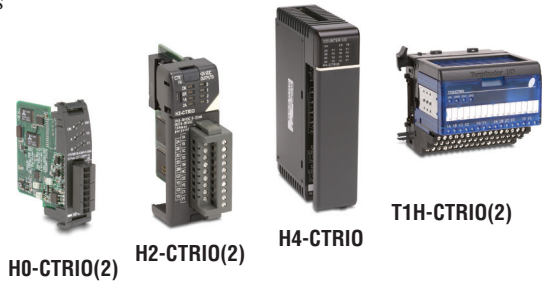
Icon Legend:

- DM Do-more icon-** Topic is applicable when CTRIO(2) is used with a Do-more CPU or in a slave on a Do-more CPU’s Ethernet I/O network.
- DL DirectLOGIC icon-** Topic is applicable when CTRIO(2) is used with a DirectLOGIC CPU.
- Win WinPLC icon-** Topic is applicable when CTRIO(2) is used with a WinPLC CPU.
- NI Network Interface icon-** Topic is applicable when CTRIO(2) is used with any of the network interfaces: EBC (see **Do-more icon** instead if EBC is a slave on Do-more Ethernet I/O network) DeviceNet, Profibus or Modbus.

CTRIO and CTRIO2 Module Overview

A CTRIO(2) module is a programmable motion co-processor capable of accepting a variety of encoder or discrete sensor inputs, accepting commands from the CPU, natively executing simple control algorithms, and generating a variety of pulse-type motion control signals or discrete actuator outputs. A CTRIO can be used for a wide variety of basic motion tasks. Most commonly, a CTRIO module is used to:

- Track an encoder
- Calculate rate from an encoder
- Execute homing routines
- Generate simple motion profiles
- Send pulse train control signals to a stepper or servo amplifier
- Precisely fire a discrete output based on the position read from an encoder



NOTE: The T1H-CTRIO(2) is only supported by the T1H-EBC, T1H-EBC100 and T1H-PBC.

Available Functions

The various functions available in a CTRIO(2) module are enumerated in the following text. There are however, limitations and dependencies to keep in mind.

- An H0-CTRIO(2) module has 4 input points on its single channel. The H2-CTRIO(2) modules have 8 input points split between two channels. Although each channel of a CTRIO(2) module has four input points, a channel may have only two counters/quad counters defined. The counter/quad counter inputs appear as Function 1 and Function 2 in CTRIO Workbench.
- Some functions rely on another function being enabled. For example, scaling cannot be selected until other inputs have been configured as a counter or quad counter. The dependencies are enforced by CTRIO Workbench software to ensure that an invalid configuration cannot be created.

If a function cannot be found in CTRIO Workbench, try enabling the function(s) on which it might depend.

Inputs:

Input types accepted:

- Quadrature encoder with AB or ABZ
- Counter (tachometer)
- Discrete (photo eye, limit switch, Z pulse, etc.)

Functions available for discrete inputs:

- Pulse catch (high-speed discrete input with programmable filter)
- Timing: edge timer (period), dual edge timer (time difference of two inputs)
- Reset counts (Z input from encoder)
- Capture counts (copy counts of one of the counter inputs to a register)
- Inhibit counting (freeze one of the counter inputs)
- Limit for pulse output functions (CTRIO2, home switch on an axis)

Scaling of timing functions or encoder inputs:

- Rate scaling (allows the CTRIO(2) to provide encoder data to the PLC in engineering units)
- Position scaling (allows the CTRIO(2) to provide encoder data to the PLC in engineering units)

Outputs:

Assign the output points:

- Stepper control: Step/Direction or CW/CCW
- Discrete

Pulse profiles for stepper outputs to follow:

- Dynamic Position Plus, Trapezoid Plus, Trapezoid with Limits (homing) (CTRIO2 only and CTRIO, Workbench v2.2.0 or later required)
- Trapezoid, S-curve, Symmetrical S-curve, Dynamic Position, Dynamic Velocity
- Home Search, Free Form

Associate output functions with inputs

- Programmable Limit Switch or 'PLS' (CTRIO2 only and CTRIO Workbench v2.2.0 or later required)
- Preset tables.

Unsuitable Applications

There are some applications the CTRIO(2) specifications appear to support that are not feasible. Common applications a CTRIO(2) cannot readily handle are listed below. If in doubt regarding your application, please contact Automationdirect Tech Support for assistance.

Closed loop control: CTRIO2 modules are capable of very basic closed loop control. However, they do not have full functionality expected of a typical dedicated closed loop controller. Most notably, position or velocity errors are not reported and there is no built-in error alarming.

- CTRIO modules do not support any closed loop control. Trying to use the CPU to close the control loop will produce unacceptable results due to excessive latency.

Coordinated motion: Some CTRIO(2) modules have enough outputs to control multiple axes, but there is no internal mechanism to coordinate them. Axes can move simultaneously, but not with coordination.

Follower: CTRIO(2) modules cannot support follower applications natively. Trying to use the CPU to close the control loop will produce unacceptable results due to excessive latency.

Precise registration: There is no means for precisely timing the start of a motion profile. Motion profiles are initiated by the controller in the base, so scan time latency of the controller is always a factor.

Absolute encoders: A CTRIO(2) module cannot read an absolute encoder.

Mechanical contacts as counter or encoder inputs: Reliable readings are not possible using mechanical contacts. The bounce of mechanical contacts will cause the CTRIO(2) to see more edges than intended.

Direct connection to TTL, line driver or differential encoders: A CTRIO(2) module cannot accept these low voltage inputs directly. These signals need to be level shifted as shown in Chapter 3: *Installation and Field Wiring*.

Support Systems for the CTRIO(2) Modules

The CTRIO(2) modules are compatible with several CPU-slot interfaces. Consideration must be given to the firmware versions of the CPU-slot interfaces to assure their compatibility with the CTRIO(2) (See Chapter 3 for CPU/CTRIO compatibility listings). Multiple CTRIO(2) modules can reside in the same base provided that the backplane power budget is adequate.

Support Systems for the H0-CTRIO(2):

- *Direct*LOGIC 05/06 PLC systems

Support Systems for the H2-CTRIO(2):

- *Direct*LOGIC 205 PLC systems (D2-240, D2-250-1, D2-260, D2-262)
- DL205 WinPLC systems (H2-WPLCx-xx)
- PC-based control strategies using the H2-EBC(100) interface module
- Hx-ERM networks using the H2-EBC(100) interface module
- Profibus systems using the H2-PBC slave interface module
- Do-more PLC systems (H2-DM1, H2-DM1E); See Do-more Designer help file.
- Do-more PLC Ethernet I/O network using H2-EBC100; See Do-more Designer help file.

Support Systems for the H4-CTRIO:

- *Direct*LOGIC 405 PLC systems (D4-450 OR D4-454 only)
- PC-based control strategies using the H4-EBC interface module
- Hx-ERM networks using the H4-EBC interface module

Support Systems for the T1H-CTRIO:

- PC-based control strategies using the T1H-EBC interface module
- Profibus systems using the T1H-PBC slave interface module
- Hx-ERM networks using the T1H-EBC interface module
- Do-more PLC Ethernet I/O network using T1H-EBC100

CTRIO(2) Specifications

The tables following show general and specific information associated with CTRIO modules.

General						
Module	H0-CTRIO	H0-CTRIO2	H2-CTRIO	H2-CTRIO2	H4-CTRIO	T1H-CTRIO
Module Type	Intelligent					
Modules Per Base	Limited only by power consumption					
I/O Points Used	None, I/O map directly in PLC memory (V-memory for <i>Direct</i> LOGIC PLCs and Data structures for Do-more PLCs) or PC control access					
Field Wiring Connector	Standard removable terminal block					
Internal Power Consumption	250mA at +5V from Base Power Supply (All I/O in ON State at Max Voltage/Current)	400mA Max at +5V from Base Power Supply (All I/O in ON State at Max Voltage/Current)	400mA Max at +5V from Base Power Supply (All I/O in ON State at Max Voltage/Current)	275mA Max at +5V from Base Power Supply (All I/O in ON State at Max Voltage/Current)	400mA Max at +5V from Base Power Supply (All I/O in ON State at Max Voltage/Current)	400mA Max at +5V from Base Power Supply (All I/O in ON State at Max Voltage/Current)
Operating Environment	32°F to 140°F (0°C to 60°C), Humidity (non-condensing) 5% to 95%					
Manufacturer	Host Automation Products, LLC					
Isolation	2500V I/O to Logic, 1000V among Input Channels and All Outputs	1500V I/O to Logic, 1000V among Input Channels and All Outputs	2500V I/O to Logic, 1000V among Input Channels and All Outputs	1500V I/O to Logic, 1000V among Input Channels and All Outputs	2500V I/O to Logic, 1000V among Input Channels and All Outputs	2500V I/O to Logic, 1000V among Input Channels and All Outputs

CTRIO(2) Specifications, cont'd

CTRIO(2) Input Specifications						
Module	H0-CTRIO	H0-CTRIO2	H2-CTRIO	H2-CTRIO2	H4-CTRIO	T1H-CTRIO
Inputs	4 pts. sink/ source 100kHz Max	4 pts. sink/ source 250kHz Max	8 pts. sink/source 100kHz Max	8 pts. sink/source 250kHz Max	8 pts. sink/source 100kHz Max	
Minimum Pulse Width	5 μ sec	0.5 μ sec	5 μ sec	0.5 μ sec	5 μ sec	
Input Voltage Range	9-30 VDC					
Maximum Voltage	30VDC					
Input Voltage Protection	Zener Clamped at 33VDC					
Rated Input Current	8mA typical, 12mA maximum					
Minimum ON Voltage	9.0 VDC					
Maximum OFF Voltage	2.0 VDC					
Minimum ON Current	5.0 mA (9VDC required to guarantee ON state)					
Maximum OFF Current	2.0 mA					
OFF to ON Response	Less than 3 μ sec	Less than 0.5 μ sec	Less than 3 μ sec	Less than 0.5 μ sec	Less than 3 μ sec	
ON to OFF Response						
Counter/Timer	2, (2 per single 4 input channel); supports 1 quadrature counter max.		4, (2 per each 4 input channel group); supports 2 quadrature counters max.			
Resource Options	1X, 2X, or 4X Quadrature, Up or Down Counter, Edge Timer, Dual Edge Timer, Input Pulse Catch, Reset, Inhibit, Capture					
Timer Range/ Resolution	4.2 billion (32 bits); 1 μ sec (70 minutes)					
Counter Range	\pm 2.1 billion (32 bits or 31 bits + sign bit)					

CTRIO(2) Output Resources						
Module	H0-CTRIO	H0-CTRIO2	H2-CTRIO	H2-CTRIO2	H4-CTRIO	T1H-CTRIO
Pulse output / Discrete outputs	Pulse outputs: 1 channel (2 outputs per single channel) Discrete outputs: 2 pts.		Pulse outputs: 2 channels (2 outputs per each channel) Discrete outputs: 4 pts.			
Resource Options	Pulse Outputs: pulse/direction or cw/ccw; Profiles: Trapezoid, S-Curve, Symmetrical S-Curve, Dynamic Positioning, Dynamic Velocity, Home Search, Free Form, Dynamic Positioning Plus (CTRIO2 only), Trapezoid Plus (CTRIO2 only), Trapezoid w/Limits (CTRIO2 only), Velocity Mode, Run to Limit Mode, Run to Position Mode Discrete Outputs: configurable for set, reset, pulse on, pulse off, toggle, reset count functions (assigned to respond to Timer/Count input functions). Raw Mode: Direct access to discrete outputs from user application program					
Target Position Range	\pm 2.1 billion (32 bits or 31 bits + sign bit)					

CTRIO(2) Specifications, cont'd

CTRIO(2) Output Specifications						
Module	H0-CTRIO	H0-CTRIO2	H2-CTRIO	H2-CTRIO2	H4-CTRIO	T1H-CTRIO
Outputs	2 pts, isolated, either both current sourcing or both current sinking FET Outputs: open drain and source with floating gate drive		4 pts, independently isolated, current sourcing or sinking FET Outputs: open drain and source with floating gate drive			
Pulse Output Control Range	20Hz to 25kHz	20Hz to 250kHz	20Hz to 25kHz	20Hz to 250kHz	20Hz to 25kHz	
Voltage Range	5VDC – 36VDC					
Maximum Voltage	36VDC					
Output clamp voltage	60VDC					
Maximum load current	1.0A	0.5 A at 23°C 0.33 A at 60°C	1.0 A	1.0 A at 23°C, 0.5 A at 60°C	1.0 A	
Maximum load voltage	36VDC	33VDC	36VDC	36VDC	36VDC	
Maximum leakage current	100µA					
Inrush current	5A for 20ms	1A for 10ms	5A for 20ms	2A for 10ms	5A for 20ms	
OFF to ON response	less than 3 µsec			less than 1 µsec	less than 3 µsec	
ON to OFF Response	less than 3 µsec					
ON state V drop	≤ 0.3 V			≤ 0.45 V	≤ 0.3 V	
External power supply	For loop power only, not required for internal module function*					
Overcurrent protection	15A max	15A max; Self resetting overcurrent protection	15A max			
Thermal shutdown	T _{junction} = 150°C					
Overtemperature reset	T _{junction} = 130°C					
Duty cycle range	1% to 99% in 1% increments (default = 50%)	0.1% to 99.9% in 0.1% increments	1% to 99% in 1% increments (default = 50%)	0.1% to 99.9% in 0.1% increments	1% to 99% in 1% increments (default = 50%)	
Configurable Presets a) single b) multiple	a) each output can be assigned one preset, or b) each output can be assigned one table of presets, one table can contain max. 128 presets, max. predefined tables = 255					

H0-CTRIO(2) LED Indicators

H0-CTRIO(2) LED Descriptions	
OK	Module OK
ERR	User Program Error
A	Ch1 F1 Resource State
B	Ch1 F2 Resource State
Y0 - Y1	Output Status



H0-CTRIO(2) LED Diagnostics		
LED Status		Status Description
OK	ERR	
ON	OFF	RUN Mode
ON	ON	Hardware Failure
Blinking	Blinking	Boot Mode - Used for Field OS Upgrades
Blinking	OFF	Program Mode
OFF	Blinking	Module Self-diagnostic Failure
OFF	ON	Module Error Due to Watchdog Timeout
OFF	OFF	No Power to Module

H0-CTRIO(2) LED Run Diagnostics Definitions	
A	Blinks when Channel 1 Function 1 is counting or timing
B	Blinks when Channel 1 Function 2 is counting or timing
Y0-Y1	Follow actual output state; ON = output is passing current

H2-CTRIO(2) LED Indicators

H2-CTRIO(2) LED Descriptions	
OK	Module OK
ER	User Program Error
1A	Channel 1 Status
2A	Channel 2 Status
0-3	Output Status



H2-CTRIO(2) LED Diagnostics		
LED Status		Status Description
OK	ER	
ON	OFF	RUN Mode
ON	ON	Hardware Failure (H2-CTRIO)
		Not Used (H2-CTRIO2)
Blinking	Blinking	Boot Mode - Used for Field OS Upgrades
Blinking	OFF	Program Mode
OFF	Blinking	Module Self-diagnostic Failure (Blinks may be coded by counts)
OFF	ON	Module Error Due to Watchdog Timeout
OFF	OFF	No Power to Module

H2-CTRIO(2) LED Diagnostics Definitions	
1A /2A	
Blinking 7 times per second	Input is Configured as Counter and is Changing
Following State of Input	Input is not Configured as Counter
0-3	
Follow actual output state; ON = output is passing current	

H4-CTRIO LED Indicators

H4-CTRIO LED Descriptions	
OK	Module OK
ER	User Program Error
TB	Removed Terminal Block
1A–1D	Ch1A - Ch1D Input Status
2A–2D	Ch2A - Ch2D Input Status
(Ch1) F1–F2	Ch1 Resource State
(Ch2) F1–F2	Ch2 Resource State
Y0–Y3	Output Status



H4-CTRIO LED Diagnostics		
LED Status		Status Description
OK	ER	
ON	OFF	RUN Mode
Blinking	Blinking	Boot Mode - Used for Field OS Upgrades
Blinking	OFF	Program Mode
OFF	Blinking	Module Self-diagnostic Failure
OFF	ON	Module Error Due to Watchdog Timeout
OFF	OFF	No Power to Module
TB		User Terminal Block is not Properly Installed

H4-CTRIO LED Diagnostics Definition	
1A–1D	Follow actual input state / Ch1
2A–2D	Follow actual input state / Ch2
(Ch1) F1	Blinks when Channel 1 Function 1 is counting or timing
(Ch1) F2	Blinks when Channel 1 Function 2 is counting or timing
(Ch2) F1	Blinks when Channel 2 Function 1 is counting or timing
(Ch2) F2	Blinks when Channel 2 Function 2 is counting or timing
Y0–Y3	Follow actual output state; ON = output is passing current



NOTE: Due to the multiplexed design of the DL405 LED matrix, OFF state LEDs may appear to blink ON slightly. This is to be expected and does not necessarily indicate a transient condition of the function corresponding to the LED.

T1H-CTRIO LED Indicators

T1H-CTRIO LED Descriptions	
OK	Module OK
ERR	User Program Error
CH1	Channel 1 Status
CH2	Channel 2 Status
1A-1D	Channel 1 A-D Input Status
2A-2D	Channel 2 A-D Input Status
Y0-Y3	Output Status



T1H-CTRIO LED Diagnostics		
LED Status		Status Description
OK	ER	
ON	OFF	RUN Mode
ON	ON	Hardware Failure
Blinking	Blinking	Boot Mode - Used for Field OS Upgrades
Blinking	OFF	Program Mode
OFF	Blinking	Module Self-diagnostic Failure
OFF	ON	Module Error Due to Watchdog Timeout
OFF	OFF	No Power to Module

T1H-CTRIO LED Diagnostics Definitions	
CH1	Blinks when Channel 1 Function 1 is counting or timing
CH2	Blinks when Channel 2 Function 1 is counting or timing
1A-1D	Follow actual input state / Ch1
2A-2D	Follow actual input state / Ch2
Y0-Y3	Follow actual output state; ON = output is passing current

Overview, How it Works as Part of the Control System

Basic Operation

A CTRIO(2) is an intelligent co-processor module. It has to be configured using CTRIO Workbench before it can do anything. It has its own scan time and can be in either program or run mode.

Being an Intelligent Module means that the CTRIO(2) controls its own writes to and reads from CPU memory. The CPU does not directly write to or read from the CTRIO(2). The CTRIO(2) will write to and read from the addresses designated in I/O Map. Understanding this relationship is helpful to understanding the timing (interlocking) requirements for performing operations that require the PLC to make changes to the CTRIO(2) from ladder logic.

Understand that the CPU can only make requests to the CTRIO(2) by turning on specific bits and placing appropriate data in its own memory. The CTRIO(2) reads the data from those addresses in the CPU and then acts on it. The CTRIO(2) provides feedback to the CPU by writing to other CPU memory.



NOTE: A CTRIO(2) is an Intelligent Module. It directly writes to and reads from CPU memory. The CPU cannot directly write to or read from CTRIO(2) memory. The CTRIO(2) writes to and reads from the addresses designated in I/O Map.

After being configured by CTRIO Workbench, a CTRIO(2) module is ready to be placed in run mode. Basic input functions of the CTRIO(2) run automatically, such as reporting counts on an input channel. Output functions are initiated by the controller (the PLC CPU). The controller uses Command Codes to control the output functions of the CTRIO(2), such as executing a pulse profile on an output, or loading or editing a preset table. IBox instructions are macros that use Command Codes. Command Codes and the instructions on their use are found later in this manual.

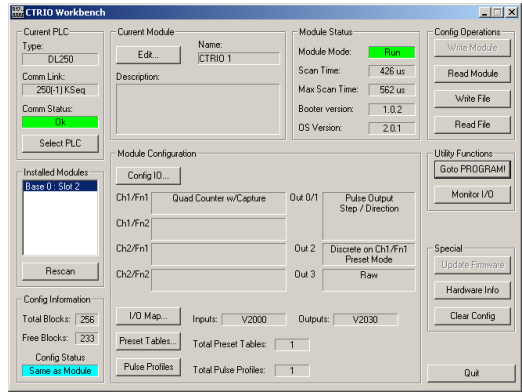
CTRIO Workbench

CTRIO Workbench is the utility used to configure the many functions available for a CTRIO(2) module.

CTRIO Workbench communicates with a CTRIO(2) module through the base controller to configure the CTRIO(2). The configuration is stored in the CTRIO(2) and also stored in a file on your computer. Configuring the CTRIO(2) is a separate process from programming the base controller.

The CTRIO Workbench configuration contains these basic parameters:

- Assignments of the input and output points
- Scaling of inputs (optional)
- Preset Table setup (optional)
- Programmable Limit Switch setup (CTRIO(2), optional)
- Mapping to CPU memory



NOTE: The CTRIO(2) must be configured in program mode, and the memory must be mapped prior to program execution.

Command Codes

Command Codes are the instructions available to the CPU to tell the CTRIO what to do. A list of the Command Codes with a brief description of their function are shown in the table below.

Command Code and Parameter Definitions

Command	Code (Hex/BCD)	Word Parameter 1 (decimal)	Word Parameter 2	DWord Parameter 3
Load Table from ROM	10	Trapezoid or S-curve Symmetrical S-Curve Home Search File Number	-	-
Load Table from ROM	10	Dynamic Positioning File Number	-	Target Position (decimal)
Load Table from ROM	10	Dynamic Velocity File Number	-	Target Velocity (decimal)
Velocity Mode	20	Run Frequency (CTRIO: 20Hz - 25KHz CTRIO2: 20Hz - 64KHz)	Duty Cycle (0 to 99)* (decimal)	Number of Pulses (BCD/Hex)
Run to Limit Mode	21	Run Frequency (CTRIO: 20Hz - 25KHz CTRIO2: 20Hz - 64KHz)	Edge & Duty Cycle (0 to 99)* (Hex/BCD)	-
Run to Position Mode	22	Run Frequency (CTRIO: 20Hz - 25KHz CTRIO2: 20Hz - 64KHz)	Compare Function & Duty Cycle (0 to 99)* (Hex/BCD)	Desired Input Function Value (decimal)

* A value of 0 will generate a duty cycle of 50%

Those fields separated by an “&” indicate a code with different definitions for each byte (high byte and low byte). For example, to enter the Pulse Output to Limit command, set the high byte of the Word Parameter 2 to the edge you wish to terminate the output pulses (see definition following), and set the low byte to the desired duty cycle.

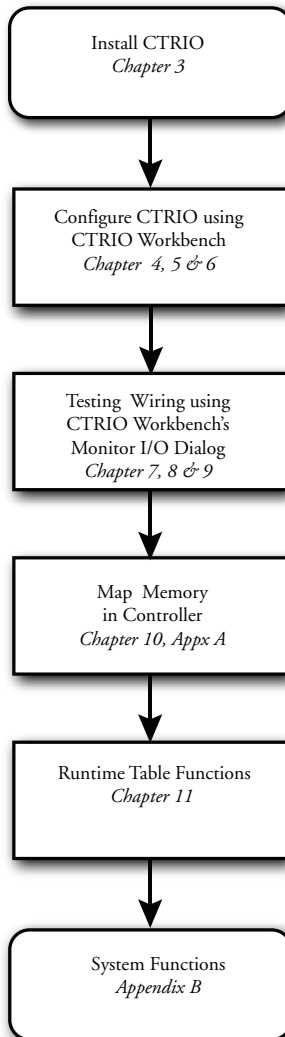
In order to process a command, first the program must load the Command Code and required DWord, Word, and bit parameters. Then the program should turn ON the Process Command bit and look for the CTRIO(2) module to acknowledge the command with the Command Complete bit. Finally, the program should reset the Process Command bit and set the Enable Output bit when appropriate. If the Command Error bit is received, the CTRIO(2) module was unable to process the command due to an illegal value in either the Command Code or parameter files.

DWord and Word values for pulse outputs are unsigned integers (Parameter 3 on some profiles can be signed).

CTRIO(2) Module Work Flow Diagram

The following workflow diagram shows the steps needed, with their associated chapters in this manual, to install and setup a CTRIO(2) module into your system.

DirectLOGIC, WinPLC or EBC



GETTING STARTED, BASICS AND EXAMPLES



In This Chapter...

Overview	2-2
Basic Motion Functions, Summary of Examples	2-2
Detailed Example: Configure and Test a Quadrature Input	2-7
Detailed Example: Configure and Test a Pulse Output w/Trapezoidal Profile.....	2-16
Flow Chart Example: Configure and Test a Pulse Output w/Trapezoidal Profile	2-27

Overview

This chapter is intended for the newcomer and includes brief descriptions of how to implement some common motion control solutions using CTRIO(2). The descriptions should give the newcomer a good understanding of what basic steps are required to implement the function. With this general understanding, specifics on each step can be sought out later in the manual.

Later in the chapter, two detailed examples walk the user through what is required to implement two of the most common functions, reading a quadrature encoder and generating a trapezoidal profile.

Basic Motion Functions, Summary of Examples

Get Position Using an Encoder

To read the position of an encoder, follow these basic steps in CTRIO Workbench:

- 1: Config I/O - Configure the appropriate inputs to recognize the encoder.
- 2: Optionally set up position scaling if desired.
- 3: Use I/O Map to map the CTRIO(2) to the controller's memory and print the I/O Map Report.
- 4: Download the configuration to the CTRIO(2) module and put the CTRIO(2) in Run.
- 5: Finally, go online with the controller to check that the encoder counts are appearing in the mapped address.

Get Rate Using an Encoder

To read the rate of an encoder, follow these basic steps in CTRIO Workbench:

- 1: Select Config I/O under Module Configuration to Configure the appropriate inputs to recognize the encoder.
- 2: Set scaling to rate - Scaling Wizard ruler button:
- 3: Choose the conversion parameters.
- 4: Use the Rate Scaling Calculator to verify the chosen settings.
- 5: Use I/O Map to map the CTRIO(2) to the controller's memory and print the I/O Map Report.
- 6: Download the configuration to the CTRIO(2) module and put the CTRIO(2) in Run.
- 7: Go online with the controller to check that encoder rate is appearing in the mapped address.

Measure Timing Between Pulse Edges

To measure the time between edges of a pulse in CTRIO Workbench:

- 1: Select Config I/O under Module Configuration to configure an input as Edge Timer, selecting the appropriate options (free-run is suggested for testing since it does not require interaction from the controller to function).
- 2: Optionally, set up scaling if desired.

- 3: Select I/O Map to map the CTRIO(2) to the controller memory and print the I/O Map Report.
- 4: Download the configuration to the CTRIO(2) module and put the CTRIO(2) in Run.
- 5: Go online with the controller to check that pulse measurements are appearing in the mapped address.

Output Position Pulses

Several options are available for generating pulses for controlling drives, steppers, servos, etc.. Some options are only available on a CTRIO2 module, as noted. Those only available on CTRIO2 tend to be more versatile and are preferred. Shaded cells highlight the advantaged attributes.

Options	Position Source	Change target position on the fly	Accel / Decel ramps	Maximum pulse rate	Encoder Feedback Possible	Requires CTRIO2
Dynamic Positioning Plus	From controller	Yes	Specify separately	250kHz	Yes	Yes
Trapezoid Plus	From controller	Yes	Specify separately	250kHz	Yes	Yes
Dynamic Positioning	From controller	No	One setting for both	65kHz ¹	No	No
S-Curve	Hard-coded in pulse profile	No	Specify separately	65kHz ¹	No	No
Symmetrical S-Curve	Hard-coded in pulse profile	No	One setting for both	65kHz ¹	No	No
Trapezoid	Hard-coded in pulse profile	No	Specify separately	65kHz ¹	No	No
Free Form	Hard-coded in pulse profile	No	None	65kHz ¹	No	No

The following example uses Dynamic Positioning Plus on a CTRIO2. To use Dynamic

¹The older CTRIO outputs can only achieve 25kHz maximum. CTRIO2 outputs are limited to 65kHz when using these profiles.

Positioning Plus to send output pulses to an amplifier (without encoder feedback), follow these basic steps in CTRIO Workbench:

- 1: Select Config I/O under Module Configuration to configure the outputs to provide pulses appropriate for the amplifier.
- 2: Select Pulse Profiles at bottom of dialog box.
- 3: Optionally, give the profile a name.
- 4: Select Dynamic Positioning Plus and choose the Frequency Settings appropriate for the motor and system.
- 5: Note the File Number assigned.
- 6: Use I/O Map to map the CTRIO(2) to the controller's memory and print the I/O Map Report.
- 7: Download the configuration to the CTRIO(2) module and put the CTRIO(2) in Run.

Home an Output

There are several available options for finding home on an output. If using a CTRIO2, *Trapezoid with Limits* profile will nearly always be the best option. The following example uses Trapezoid with Limits profile on a CTRIO2.

The table below shows a comparison of the available methods of homing an output with a CTRIO(2) module. Shaded cells highlight the advantaged attributes.

To home an output with one or more limit switches, follow these basic steps in CTRIO Workbench.

Profile	Accel and Decel Ramps	Creep to second limit or position	Change velocity manually on the fly	Multiple limits or triggers on a single home search	Single input can act as multiple triggers	Profile defined in CTRIO Workbench	Requires CTRIO2
Trapezoid with Limits	Yes	Yes	No	Yes	Yes	Yes	Yes
Home Search*	No	No*	No	Yes	No	Yes	No
Run to Limit	No	No	Yes	No	No	No	No

*Home Search allows you to select 2nd limit or different speed.

- 1: Select Config I/O to configure the outputs to provide pulses appropriate for the chosen amplifier.
- 2: Configure one or more inputs as *Limit Out n* where *n* is the output configured above.
- 3: Select Pulse Profiles at bottom of dialog box.
- 4: Select Add to create a new profile, or select an existing named profile.
- 5: Optionally, give the profile a name (defaults to “File 1-” upon closing dialog box).
- 6: Select Trapezoid profile with Limits.
- 7: Configure the Decel and Stop Triggers.
- 8: Choose the Frequency Settings appropriate for the motor and system.
- 9: Note the File Number assigned.
- 10: Use I/O Map to map the CTRIO(2) to the controller’s memory and print the I/O Map Report.
- 11: Download the configuration to the CTRIO(2) module and put the CTRIO(2) in Run.



NOTE: The following steps are generic. Details are available in the manual.

In the controller:

- 1: Command the CTRIO2 to load the Trapezoid with Limits profile (by file number) for the appropriate output.
- 2: Specify the direction (CW/CCW) to seek home.
- 3: Set the appropriate CTRIO2 Enable Output bit.
- 4: The controller can monitor status bits for visibility into the CTRIO's progress.

Output PWM Pulses¹

To generate PWM outputs, follow these basic steps in CTRIO Workbench.

- 1: Config I/O - Configure an output for Step/*Direction*, the step output will have the PWM signal.
- 2: Use I/O Map to map the CTRIO(2) to the controller's memory and print the I/O Map Report.
- 3: Download the configuration to the CTRIO(2) module and put the CTRIO(2) in Run.
- 4: In the controller - (**Note:** These steps are generic. Details are available in the *DirectLOGIC* manual).
- 5: Set the command code for Velocity Mode.
- 6: Set Parameter 1 (Frequency).
- 7: Set Parameter 2 (Duty Cycle) from 1 to 991.
- 8: Set Parameter 3 (Number of Output Pulses) to FFFF FFFF for unlimited.
- 9: Set the appropriate CTRIO2 Enable Output bit to start pulses.
- 10: To stop pulses, reset the appropriate Enable Output bit.
- 11: To change the Duty Cycle, need source.

Programmable Limit Switch or Preset Table

To control one of the CTRIO(2) outputs with a PLS or Preset Table that monitors an encoder input, follow these basic steps in CTRIO Workbench.

- 1: Config I/O - Configure the appropriate inputs to recognize the encoder, noting the Channel and Function numbers assigned.
- 2: Optionally set up scaling if desired.
- 3: Configure an output as Discrete on Ch_/Fn_, using the Channel and Function numbers from the encoder input.
- 4: Discrete Tables...

¹ The adjustable range for the PWM is 1-99(%) BCD. Entering a value of 0 is equivalent to entering 50(%) BCD. To stop pulses, reset the appropriate Enable Output bit.

- 5: Add Preset Table or Add PLS Table²
- 6: Optionally give the table a name.
- 7: Configure the table for the desired behavior.
- 8: Scales will be available if scaling was defined for the input.
- 9: Note the File Number assigned.
- 10: Use I/O Map to map the CTRIO(2) to the controller's memory and print the I/O Map Report.
- 11: Download the configuration to the CTRIO(2) module and put the CTRIO(2) in Run.



NOTE: *The following steps are generic. Details are available in the manual.*

In the controller:

- 1: Command the CTRIO2 to load the Table by its file number
- 2: Set the appropriate CTRIO(2) Enable Output bit
- 3: The controller can monitor status bits for visibility into the CTRIO's progress

Output Discrete On/Off from Ladder

The output points on a CTRIO(2) can be turned on and off from ladder, called Raw control. Keep in mind that they will not react as quickly as the outputs of a standard discrete output module since there is also the scan time of the CTRIO(2) that can add latency. To simply turn the output of a CTRIO(2) on or off from ladder, follow these basic steps in CTRIO Workbench:

- 1: Config I/O - Configure any outputs to be controlled as Raw.
- 2: Use I/O Map to map the CTRIO(2) to the controller's memory and print the I/O Map Report.
- 3: Download the configuration to the CTRIO(2) module and put the CTRIO(2) in Run.



NOTE: *These steps are generic. Details are available later in the manual.*

In the controller:

- 1: Set the appropriate CTRIO(2) Enable Output bit.
- 2: The controller can monitor status bits for visibility into the CTRIO's progress.

² PLS Tables are only available when using CTRIO2 hardware.

Detailed Example: Configure and Test a Quadrature Input

This example walks through the steps required to get the counts from a quadrature encoder connected to a CTRIO(2) to appear in the CPU memory of a *DirectLOGIC* PLC. The example uses DL06 hardware, but is applicable to DL05 and DL205 hardware as well.

The Basic Steps

- 1: Gather and connect the hardware (not covered here).
- 2: Launch CTRIO Workbench.
- 3: Use Config I/O to configure the appropriate inputs to recognize the encoder.
- 4: Use I/O Map to map the CTRIO(2) to the controller's memory and print the I/O Map Report.
- 5: Write the configuration to the CTRIO(2) module and put the CTRIO(2) in Run.
- 6: Use Monitor I/O to verify the encoder movement is being interpreted by the CTRIO(2) properly.
- 7: Launch *DirectSOFT* and use Data View to check that position data is appearing in the mapped addresses.

Equipment Needed

DirectLOGIC 06 base and H0-CTRIO2 installed in slot 4¹. A quadrature encoder properly powered and connected to the H0-CTRIO2.

Launch CTRIO Workbench

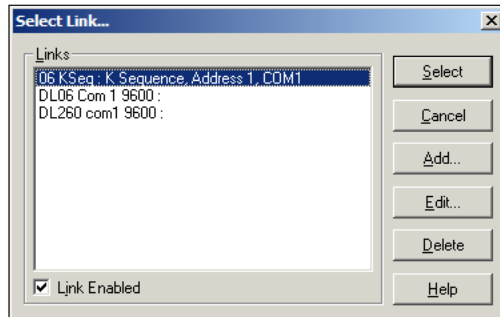
When using CTRIO Workbench with a *DirectLOGIC* CPU and the latest version of *DirectSOFT*, the best way to launch it is from DSLaunch. DSLaunch is installed when installing *DirectSOFT*. Open DSLaunch.exe.



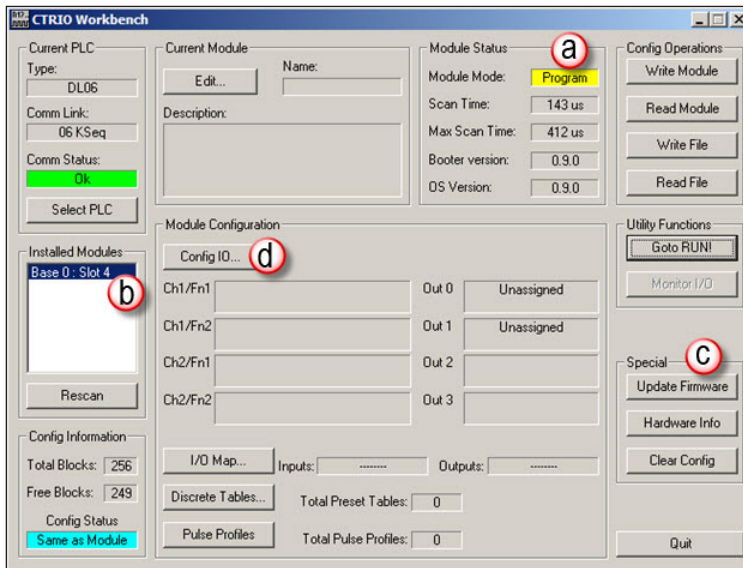
Click “CTRIO WB 2 – *DirectLOGIC* PLC” to open CTRIO Workbench. The Select

1. If using *DirectLOGIC* 205 hardware, the CTRIO(2) cannot be installed directly next to the CPU slot.

Link... dialog box appears. Select the appropriate PLC communication link to use. If it does not appear in the list, check power and the communications cable or, create a new link by clicking on 'Add'.



Once connected, CTRIO Workbench will look like the one below. The H0-CTRIO2 shown is new and has no configuration. It is in Program mode (a). It is installed in slot 4 (b). Now is a good time to check the firmware version and make sure it's current (c).



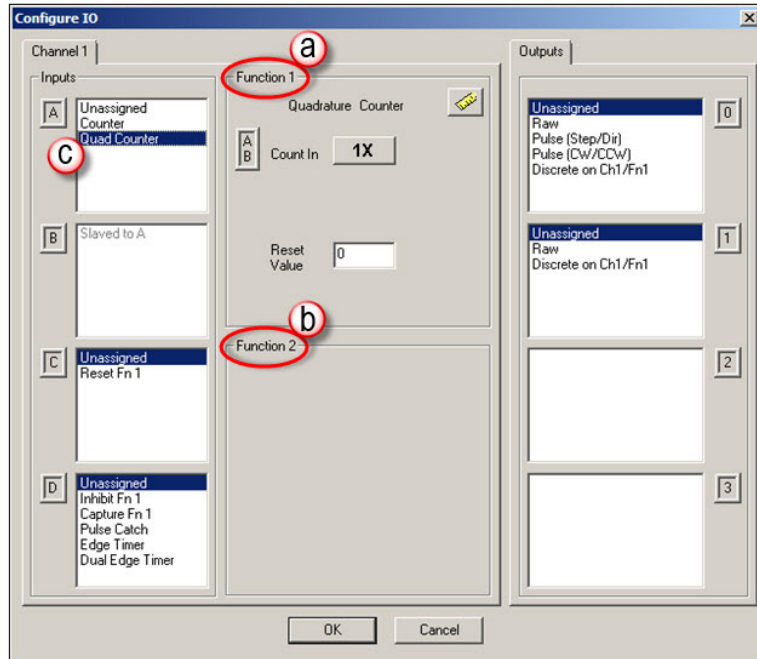
Config I/O

For this example, it is only necessary to configure the inputs of the H0-CTRIO2 to recognize a quadrature encoder. Click Config I/O (d) command box.

The Configure I/O panel is divided into two sections. One section is for Channel 1 Inputs (A-D) and the other section is for the Outputs (0-3). If working with an H2-CTRIO2, there would also be a tab for Channel 2.

Within Channel 1, note there are Function 1 (a) and Function 2 (b). Each Channel on a CTRIO(2) module may have up to two Functions assigned.

On Channel 1, select Quad Counter (c). This tells the CTRIO(2) to expect quadrature signals on Channel 1 Inputs A and automatically, B.



NOTE: Many other defaults change when a selection is made:

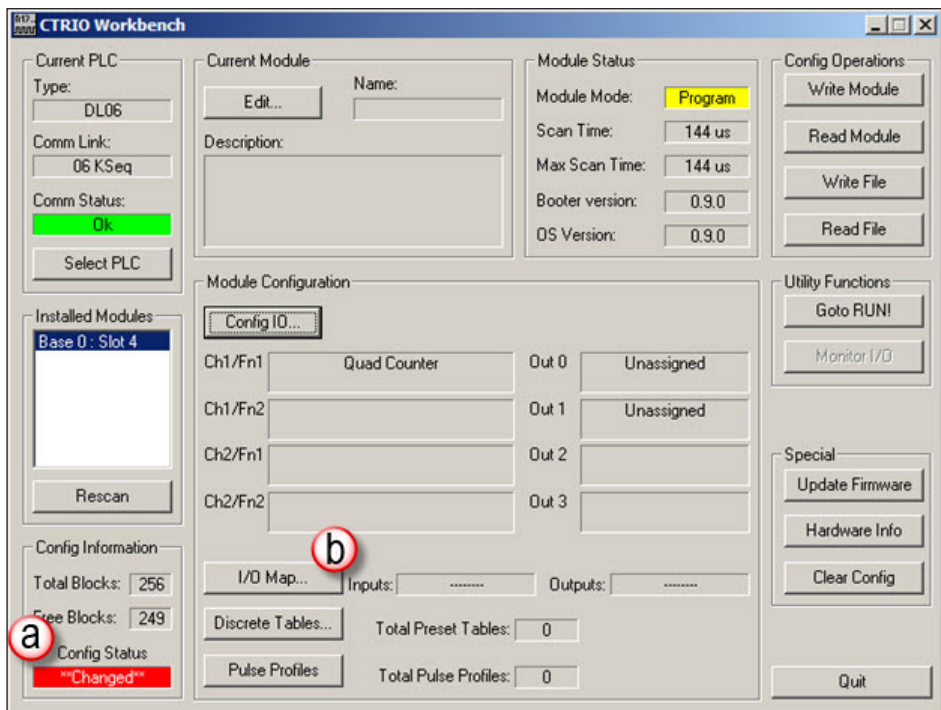
- The option for Input B to be a simple counter disappears and it is assigned as Slave to A and can no longer be directly changed.
- Some options for Input C disappear and a new option appears.
- Channel 1 Function 1 area displays Quadrature Counter and offers multiple options that apply to that counter input.
- New options for the Outputs appear that reference Ch1/ Fn1, the quadrature input function created.



NOTE: This automatic reconfiguring of available options is an important feature of CTRIO Workbench. The primary benefit is that it prevents the user from selecting options that will not work together. It is not possible to create an invalid configuration. However, keep this feature in mind when going about configuring a CTRIO(2). If a function cannot be found, it's likely that some dependency has not yet been enabled, or a feature that has been enabled is consuming an exclusive resource the desired feature also requires.

Click OK to keep the changes and go back to the CTRIO Workbench home screen.

On the home screen, note the indication in the lower left (a), Config Status ****Changed****. This indicates that the configuration in Workbench does not match what was read from the module. Before the new configuration is written to the module, there is one other thing that must be done. Click I/O Map (b). This brings up the I/O Map dialog box.



I/O Map

Depending on the configuration, the CTRIO(2) will have some number of variables to pass back and forth to the CPU. The data does not fit in the normal structure that most I/O modules use. Instead, the CTRIO(2) is able to write to and read from any CPU memory specified. In this panel, map the variables into CPU memory.



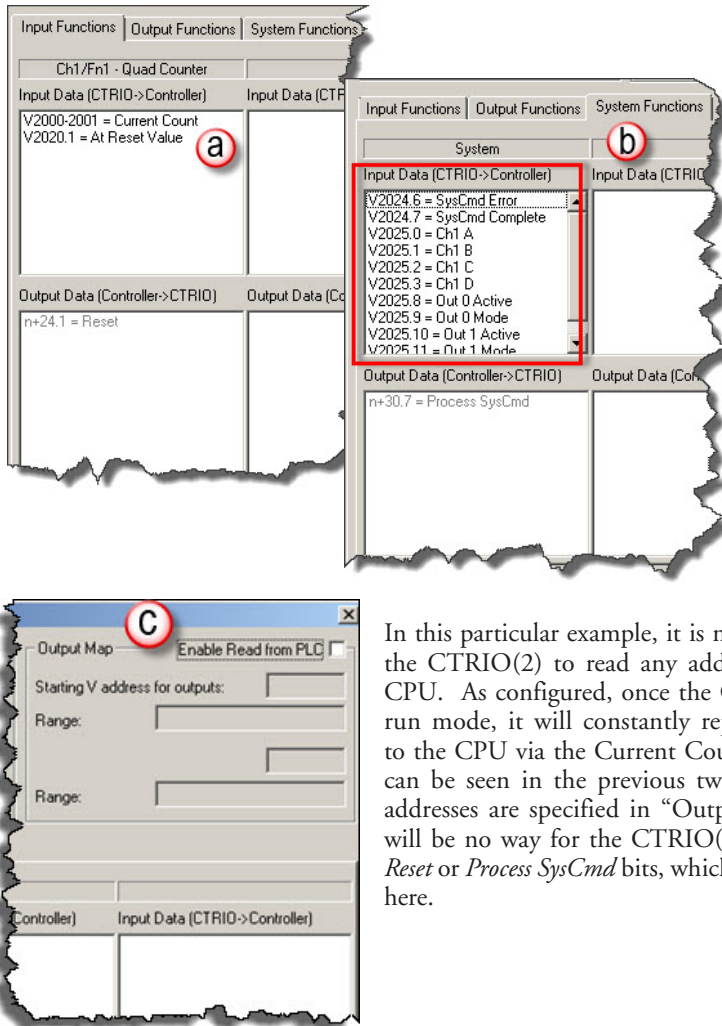
Be careful to map memory addresses that will not be used for any other purpose.

In Map Display Mode (a), select “PLC - Mapped Addresses (2 ranges)” mode.

In the Input Map box (b) point the CTRIO(2) where to write input data to the PLC memory. Click “Enable Write to PLC” and type in an address for “Starting V address for inputs”. V2000 is used here as the starting address.

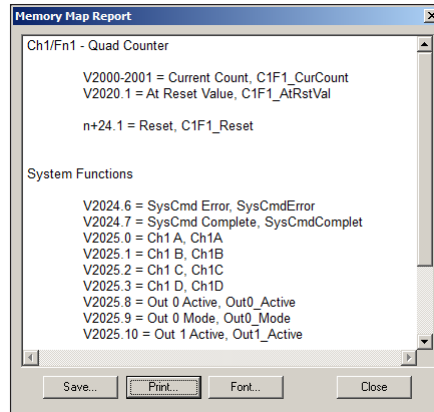
Note how entering the address changes the display under (a) Input Function tab, “Ch1/Fn1 – Quad Counter Input Data”. The two variables displayed are Current Count (32 bit integer that gets mapped to V2000-2001) and AT Reset Value (a bit that gets mapped to the second bit of V2020, or V2020.1). Clicking on the (b) “System Functions” tab reveals ten more status bit variables that were also mapped. Each of these variables provides status information to the CPU.

The “Output Map”(c) section tells the CTRIO(2) where to read variables out of the CPU and to be able to reset the counter from the ladder logic program.



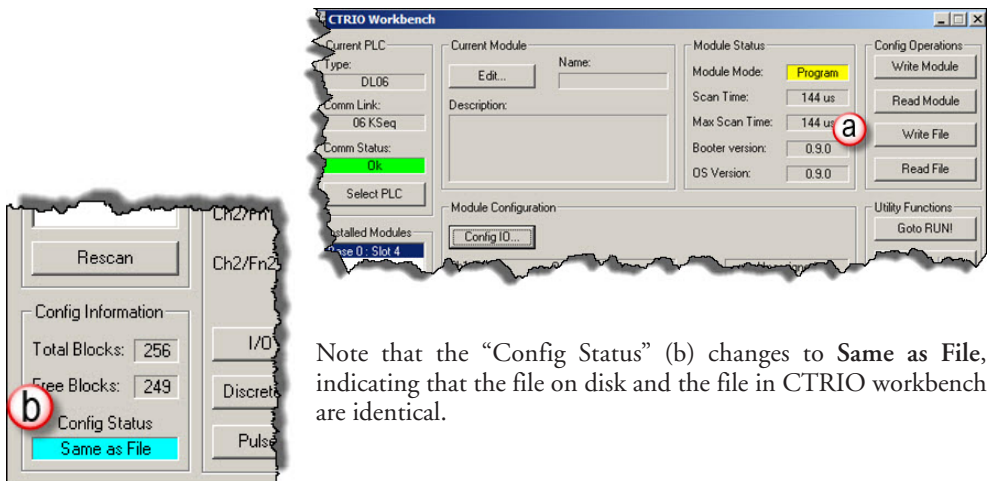
In this particular example, it is not necessary for the CTRIO(2) to read any addresses from the CPU. As configured, once the CTRIO(2) is in run mode, it will constantly report the counts to the CPU via the Current Count variable. As can be seen in the previous two images, if no addresses are specified in “Output Map”, there will be no way for the CTRIO(2) to get to the *Reset* or *Process SysCmd* bits, which are not needed here.

Once the I/O Map is set up, it is best to **print the “Memory Map Report”**. Click on the “Report...” button to access the dialog below. Click on “Print”. Creating this hard copy provides a critical reference to have in hand while writing the control program. The report can also be saved to disk as a “.txt” file by clicking on “Save” and giving it a unique name. When finished, close this dialog then click OK on the I/O Map dialog to return to the CTRIO Workbench home screen.



Download the Configuration

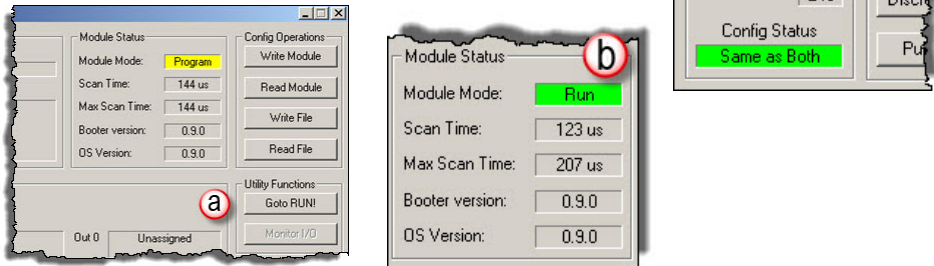
Now that the configuration has been defined, press “Write File” (a) to save a copy of the configuration to disk. It will save as a “.cwb” file that should be kept together with the other project files and documents.



Note that the “Config Status” (b) changes to Same as File, indicating that the file on disk and the file in CTRIO workbench are identical.

Once saved, click “Write Module” to write the file to the CTRIO(2) module. The Config Status changes to **Same as Both**, indicating that the file on disk, the file in CTRIO workbench and the file in the CTRIO(2) module are identical.

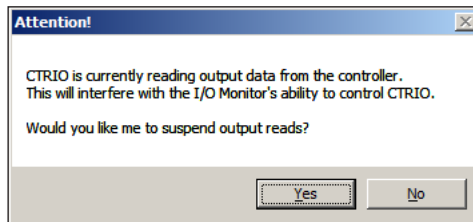
At this point, the CTRIO(2) module is still in Program mode, doing nothing. Click “Goto RUN!” (a) to have the module begin execution. The module will begin interpreting input pulses as a quadrature input and writing the resulting counts to V2000-2001. Note (b) “Module Mode” changes to **Run**.



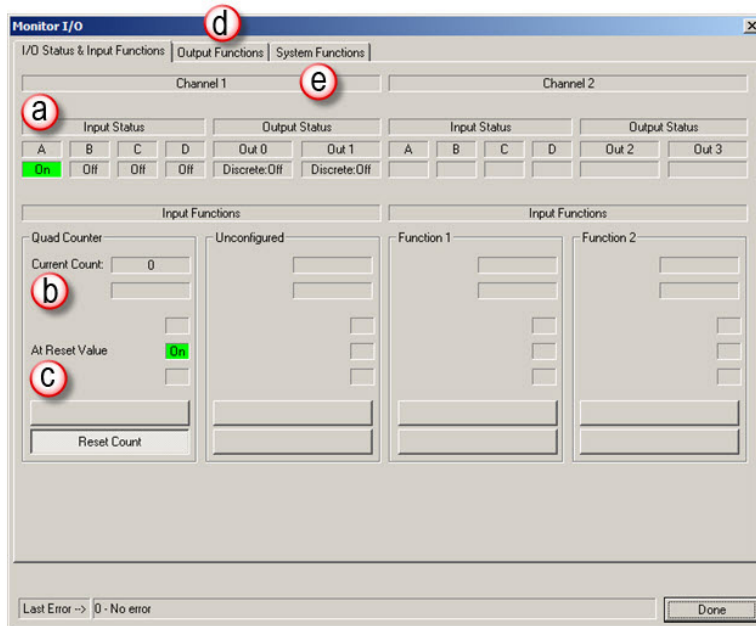
NOTE: It is only necessary to manually switch the CTRIO(2) into Run or Program during testing. During normal operation, the CTRIO(2) follows changes in the mode of the CPU. If the CPU goes to Stop, the CTRIO(2) goes to Program. If the CPU goes to Run, the CTRIO(2) goes to Run.

Monitor I/O

Before jumping over to using the CPU to work with the CTRIO(2) module, there is a very convenient way to verify everything is working as expected from within CTRIO Workbench. Click Monitor I/O. The following prompt may appear, asking whether the system should suspend output reads. Output reads are what the CTRIO(2) does to know what the CPU is telling it to do. Suspending them just means that the CTRIO(2) should temporarily stop taking commands from the CPU. The CTRIO(2) will be controlled from the Monitor I/O panel, so answer Yes.



The CTRIO(2) stops reading addresses from the CPU and Monitor I/O opens.



This dialog box displays a variety of important I/O information at a glance. The raw state of each discrete input point is indicated under **Input Status** (a). This is very handy for quickly checking the encoder wiring.

Give the encoder a twist to verify that Current Counts (b) follows the movement and At Reset Value (c) goes low. If no inputs ever appear On, check the encoder wiring carefully. If the text Quad Counter and Current Count do not appear, make sure the configuration was sent to the CTRIO(2) module.

The other tabs, not used in this example, have some very useful tools for monitoring and testing Output Functions (d) as well as System Functions (e).

Detailed Example: Configure and Test a Pulse Output with a Trapezoidal Profile

This example walks through the steps required to use a *DirectLOGIC* PLC to generate a trapezoidal profile on an output channel of a CTRIO(2). The example uses DL06 hardware, but is applicable to DL05, DL205 and DL405 hardware as well.

The Basic Steps

- 1: Gather and connect the hardware (not covered here).
- 2: Launch CTRIO Workbench.
- 3: Use Config I/O to configure the appropriate outputs to generate pulses appropriate for the amplifier in use.
- 4: Use I/O Map to map the CTRIO(2) to the controller's memory and print the I/O Map Report.
- 5: Use Pulse Profile Tables to create a Trapezoid Plus Pulse Profile and select its Frequency Settings.
- 6: Download the configuration to the CTRIO(2) module and put the CTRIO(2) in Run.
- 7: Use Monitor I/O to manually generate an output profile to make sure the configuration and hardware connections are correct.
- 8: Launch *DirectSOFT* and use Data View to manually generate an output profile.

Equipment Needed

DirectLOGIC 06 base and H0-CTRIO2 installed in slot 2¹. A stepper amplifier and motor, properly powered and connected to the H0-CTRIO2. Alternately, the output activity can be monitored in Data View or in Monitor I/O in CTRIO Workbench.

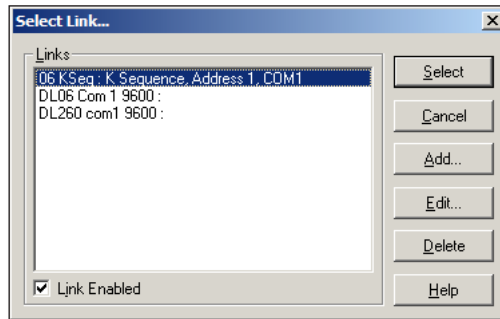
Launch CTRIO Workbench



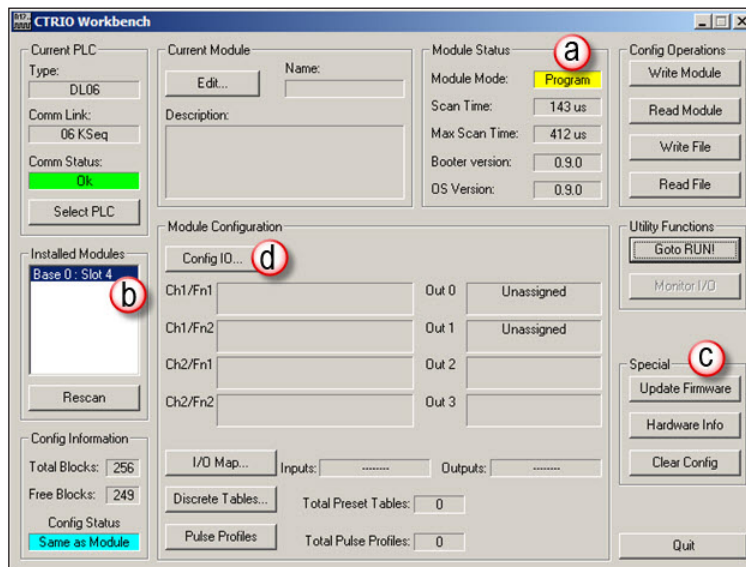
¹If using *DirectLOGIC* 205 hardware, the CTRIO(2) cannot be installed directly next to the CPU slot.

The best way to launch CTRIO Workbench is from DSLaunch. DSLaunch is installed as part of a *DirectSOFT* install. Open DSLaunch.exe.

Click *CTRIO WB 2 – DirectLOGIC PLC* to open CTRIO Workbench. The *Select Link...* dialog appears asking which PLC communication link to use. Select the appropriate PLC. If it does not appear in the list, check power and the communications cable; or, create a new link.



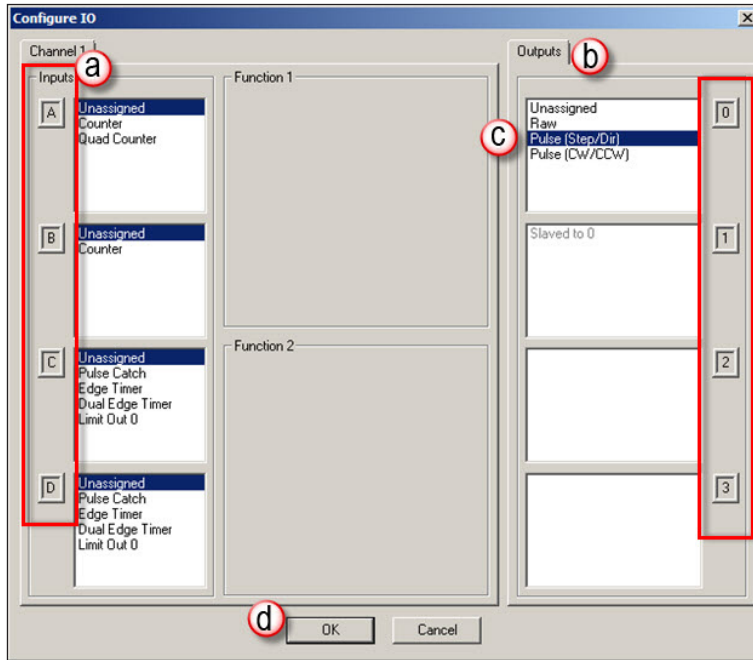
Once connected, CTRIO Workbench will look like the one below. The H0-CTRIO2 shown is new and has no configuration. It is in Program mode (a). It is installed in slot 4 (b). Now is a good time to check the firmware version and make sure it's current (c).



Config I/O

For this example, it is only necessary to configure the outputs of the H0-CTRIO2 to generate pulses for the connected stepper amplifier. Click “Config I/O”(d) command box.

The Configure I/O panel is divided into two sections. One section (a) is for Channel 1 Inputs (A-D) and the other section (b) is for the Outputs (0-3). If working with an H2-CTRIO2, there would also be a tab for Channel 2. Within Channel 1, note there are Function 1 and Function 2. Each Channel on a CTRIO(2) module may have up to two Functions assigned.



On Output 1 (c), select the appropriate format for the step amplifier. The example uses Pulse (Step/Dir), (reference here to Output 0). This tells the CTRIO(2) that Output 0 will be used as the step signal. Automatically, Output 1 is assigned as the Direction signal.

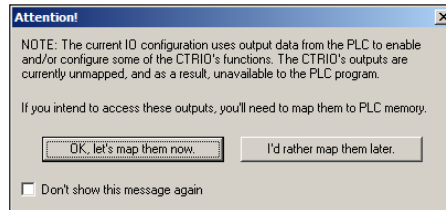
NOTE: Many other defaults change when a selection is made:

- Options for Output 1 to be Raw disappears and it is assigned as Slaved to 0 and can no longer be directly changed.
- An option is added to Channel 1 Inputs C and D for them to be Limit Out 0.

NOTE: This automatic reconfiguring of available options is an important feature of CTRIO Workbench. The primary benefit is that it prevents the user from selecting options that will not work together. It is not possible to create an invalid configuration. However, keep this feature in mind when going about configuring a CTRIO(2). If a function cannot be found, it's likely that some dependency has not yet been enabled, or a feature that has been enabled is consuming an exclusive resource the desired feature also requires.

Click "OK" (d) to keep the changes and go back to the CTRIO Workbench home screen.

A dialog box appears warning that I/O has not yet been mapped to PLC memory. This warning only occurs if mapping the memory is required. Click “OK, let’s map them now” to open the I/O Map dialog box (also accessible by clicking on “I/O Map” from the CTRIO Workbench home screen).

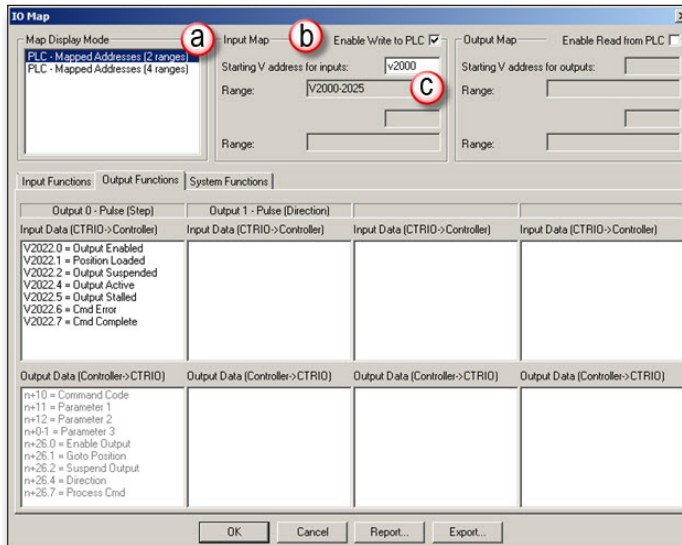


I/O Map

Depending on the configuration, the CTRIO(2) will have some number of variables to pass back and forth to the CPU. The data does not fit in the normal structure that most I/O modules use. Instead, the CTRIO(2) is able to write to and read from any CPU memory specified. In this panel, map the variables into CPU memory.



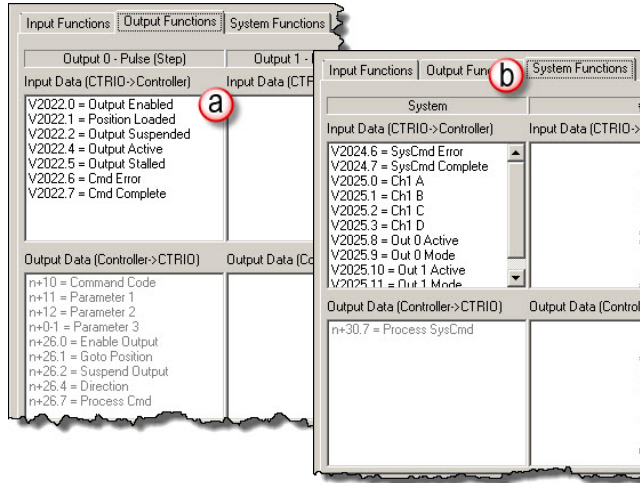
Be careful to map dedicated memory addresses that will only be used for this purpose.



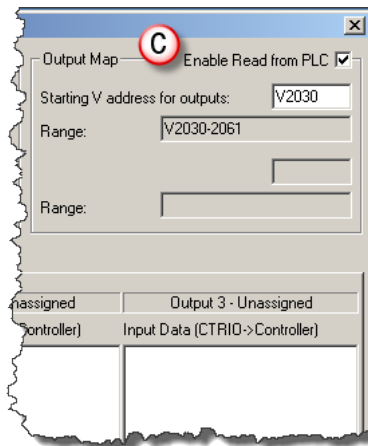
In the Map Display Mode box (a), select “PLC - Mapped Addresses (2 ranges)” mode.

Input Map section tells the CTRIO(2) where to write input data into the PLC. Click “Enable Write to PLC” (b) check box and type in an address below (c) for “Starting V address for inputs”. V2000 is used here.

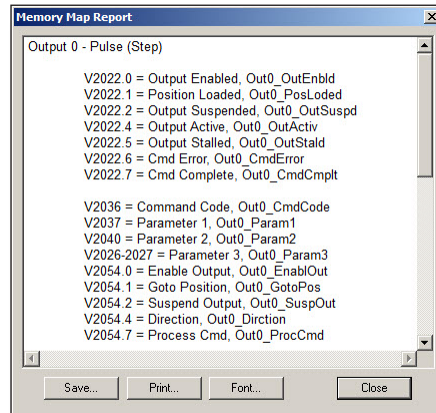
Note how entering the address changes the display under “Output Functions”, “Output 0 – Pulse (Step)” Input Data (a). Each variable is a status bit the CTRIO(2) module will write to the CPU. Clicking on the “System Functions” (b) tab reveals ten more status bit variables that were also mapped to addresses in the V2000-2025 range.



Output Map tells the CTRIO(2) where to read variables out of the CPU. Click “Enable Read from PLC” (c) and type in an address for Starting V address for outputs. The next available consecutive address, V2030, is used here. Note the Range field indicates V2030-2061 of the CPU will be used.



Once the I/O Map is set up, it is best to **print the “Memory Map Report”**. Click on the “Report...” button to access the dialog below. Click on “Print”. Creating this hard copy provides a critical reference to have in hand while writing the control program. The report can also be saved to disk as a “.txt” file by clicking on “Save” and giving it a unique name. When finished, close this dialog then click OK on the I/O Map dialog to return to the CTRIO Workbench home screen.

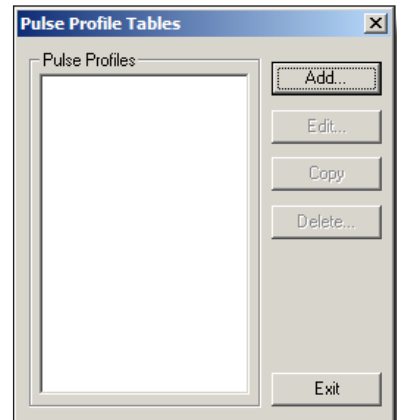


Pulse Profile Tables

For the CTRIO(2) module to generate a Trapezoidal Profile, the characteristics of the profile must be provided. As is the case with most controllers¹, the characteristics of the profile are defined as part of the CTRIO configuration. Multiple profiles may be added to the configuration to address different motion needs.

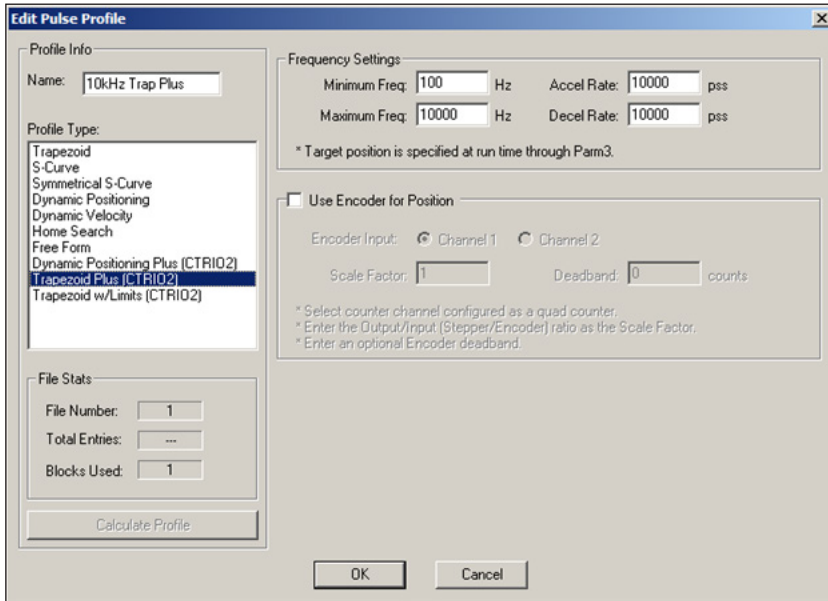
Using addresses covered in I/O Map, the controller will tell the CTRIO(2) to load one of its stored profiles. The CTRIO(2) will respond when complete. Then the CPU provides the other parameters of the move and tells the CTRIO(2) to start. During the move, the CTRIO(2) notifies the CPU of its status.

From the CTRIO Workbench home screen, press “Pulse Profiles” to access the Pulse Profile Tables panel. Click “Add...” to get to the Edit Pulse Profile panel.



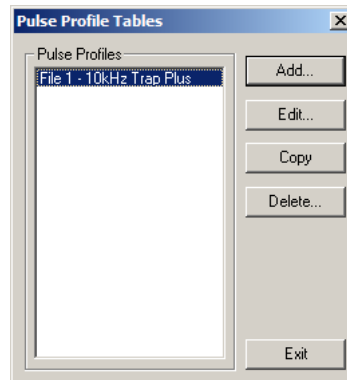
¹Using a CTRIO2 with a Do-more CPU allows use of a ladder instruction, CTAXCFG, where the characteristics of the profile are defined.

Edit Pulse Profiles offers a list of profile types on the left. Note the last three profiles are only valid for use with a CTRIO2. Select Trapezoid Plus (CTRIO2) and optionally give the profile a name; *10kHz Trap Plus* is used here. In the “Frequency Settings” section, enter appropriate frequency range for the application. The default values were left in place for the example. Press “OK” to return to **Pulse Profile Tables** box.



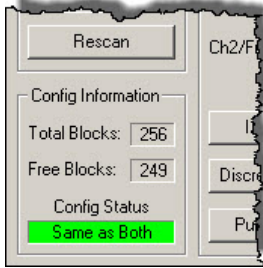
The **Pulse Profiles Tables** newly defined profile appears as *File 1 – 10kHz Trap Plus*. The assignment as File 1 is significant, as profiles will be specified in ladder by file number alone. Also, creating a PLS or Preset Table definition also results in creation of a File. Files are not just for defining Pulse Profiles.

Click “Exit” to return to the CTRIO Workbench home screen.



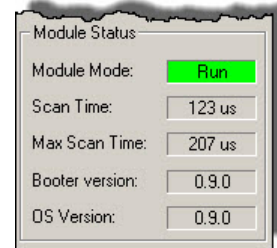
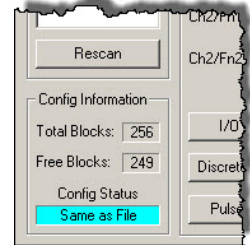
Download the Configuration

Now that the configuration has been defined, click “Write File” to save a copy of the configuration to disk. It will save as a “.CWB” file that should be kept with the other files and documents for the project. Note that the Config Status changes to Same as File, indicating that the disk file and the CTRIO workbench file are identical.



Once saved, click “Write Module” to write the file to the CTRIO(2) module. Note that the Config Status changes to Same as Both, indicating that the file on disk, the file in CTRIO workbench and the file in the CTRIO(2) module are identical.

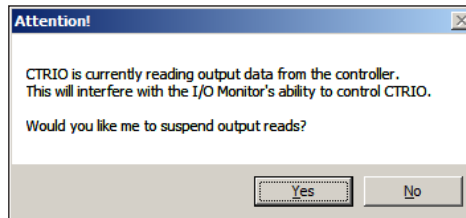
At this point, the CTRIO(2) module is still in Program mode. Click “Goto RUN!” to initiate program execution. The module will begin reading addresses from the CPU (refer to I/O map) to determine what action, if any, to take. Note that Module Mode changes to Run.



NOTE: It is only necessary to manually switch the CTRIO(2) into Run or Program during testing. During normal operation, the CTRIO(2) follows changes in the mode of the CPU. If the CPU goes to Stop, the CTRIO(2) goes to Program. If the CPU goes to Run, the CTRIO(2) goes to Run.

Monitor I/O

Before moving over to using the CPU to work with the CTRIO(2) module, there is a very convenient way to verify everything is working as expected from within CTRIO Workbench. Click “Monitor I/O” on Workbench Panel. The following prompt may appear, asking whether the system should suspend output reads. Output reads are what the CTRIO(2) does

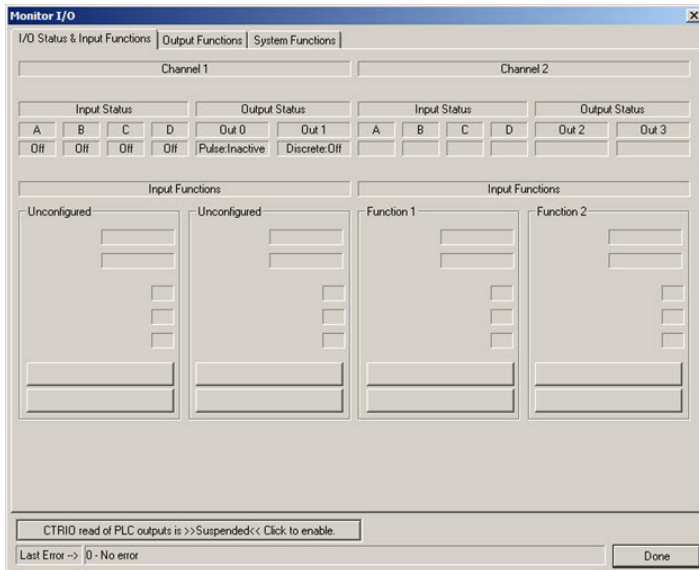
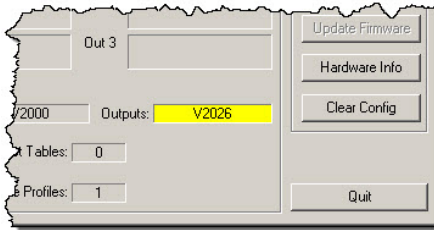


to know what the CPU is telling it to do. Suspending them just means that the CTRIO(2) should temporarily stop taking commands from the CPU. The CTRIO(2) is controlled from the Monitor I/O panel, answer Yes.

The CTRIO(2) stops reading addresses from the CPU and the Monitor I/O panel opens.



Notice the long button at the bottom of Monitor I/O panel, “CTRIO read of PLC outputs is >>Suspended<< Click to enable”. When suspended, the Outputs address will be highlighted in yellow (This is true if you exit Monitor I/O without re-enabling reads).

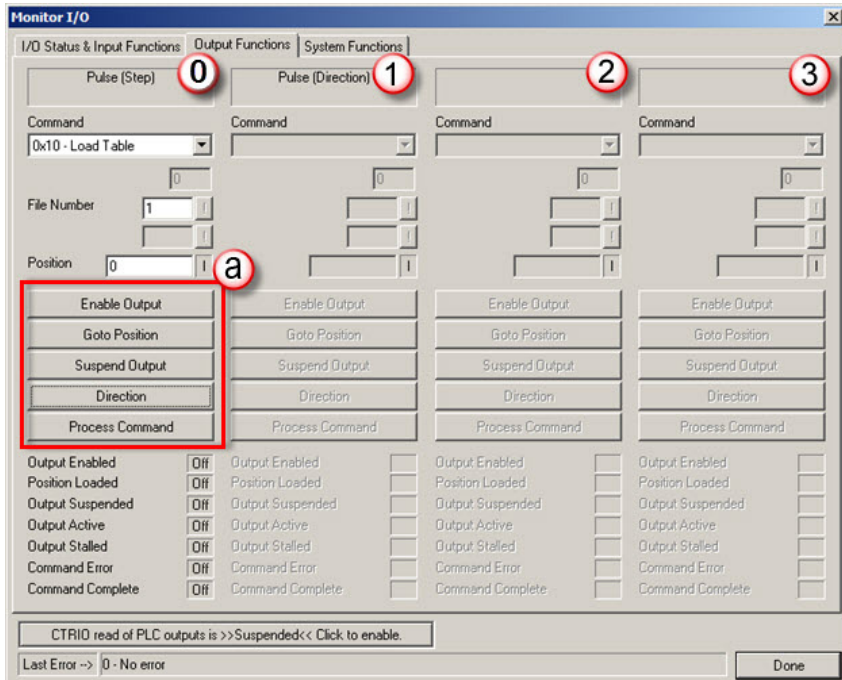


Another indication that the output reads have been suspended can be seen on the Monitor I/O panel.

At first glance, not much is going on here. Most visible buttons are grayed out. This is because the default display tab is **I/O Status and Input Functions**. There are no Input Functions defined, for this example. The only relevant section of this display, for the purpose of this example, is **Output Status - Out 0 and Out 1**. The two fields show the current status of the **Pulse Output** as Inactive and the **Direction (Discrete Output)** as Off.

Click **Output Functions** tab.

The display for Output Functions is divided into four columns representing the 4 discrete output points (0, 1, 2, 3) of the hardware. The first two points (0,1) are assigned as Pulse (Step/Dir) as indicated at the top of the first two columns (Outputs 0 & 1). The only available buttons appear under Output 0 (a) since Output 1 is slaved to Output 0. The fields and buttons under Output 0 will be used to execute a Trapezoid Plus move.



Run Trapezoid w/Limits or Trapezoid PLUS (DL-PLC)

PARAMETERS:

(1) [CommandCode]
Out0-1: n-10
Out2-3: n-16

(2) [Parameter1]
Out0-1: n-11
Out2-3: n-17

(3) [Parameter2]
Out0-1: n-0-1
Out2-3: n-4-6

(4) [ProcessCmd]
Out0-1: n-26.7
Out2-3: n-27.7

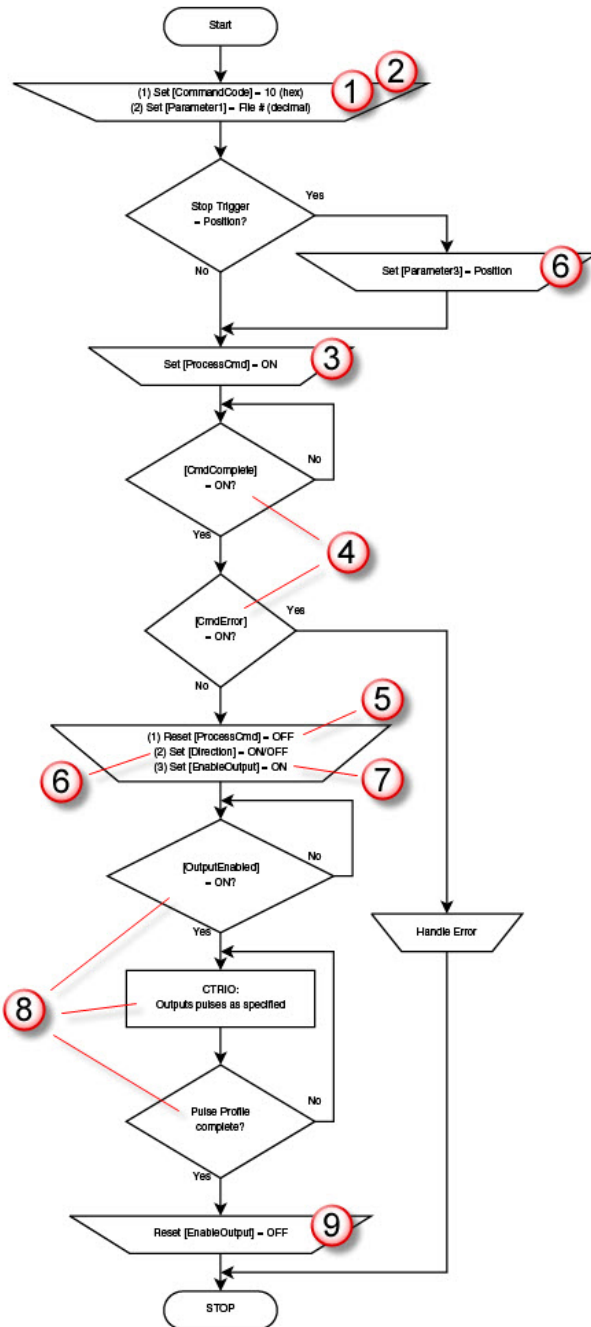
(5) [CmdComplete]
Out0-1: n-22.7
Out2-3: n-23.7

(6) [CmdError]
Out0-1: n-22.6
Out2-3: n-23.6

(7) [Direction]
Out0-1: n-26.4
Out2-3: n-27.4

(8) [EnableOutput]
Out0-1: n-26.0
Out2-3: n-27.0

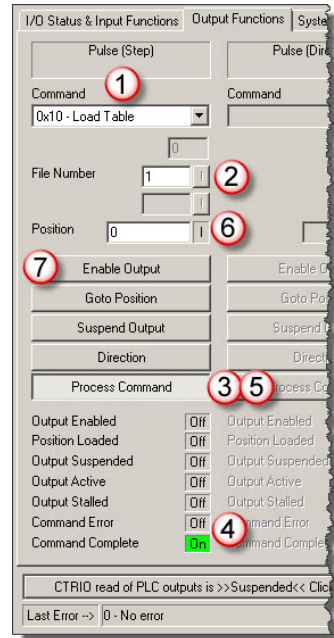
(9) [OutputEnabled]
Out0-1: n-22.0
Out2-3: n-23.0



Flow Chart Example: Configure and Test a Pulse Output with a Trapezoidal Profile

The following steps are noted on the graphic below as well as the flow chart on the previous page.

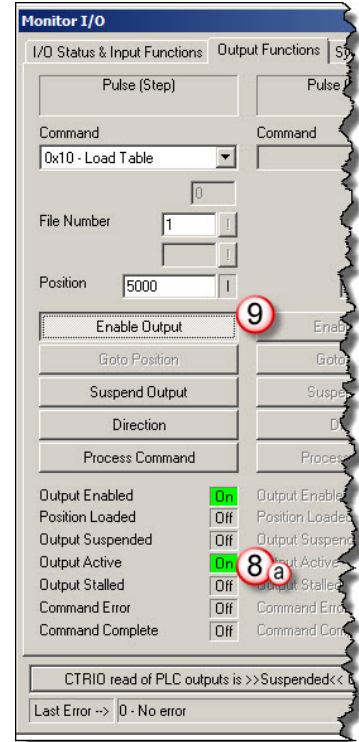
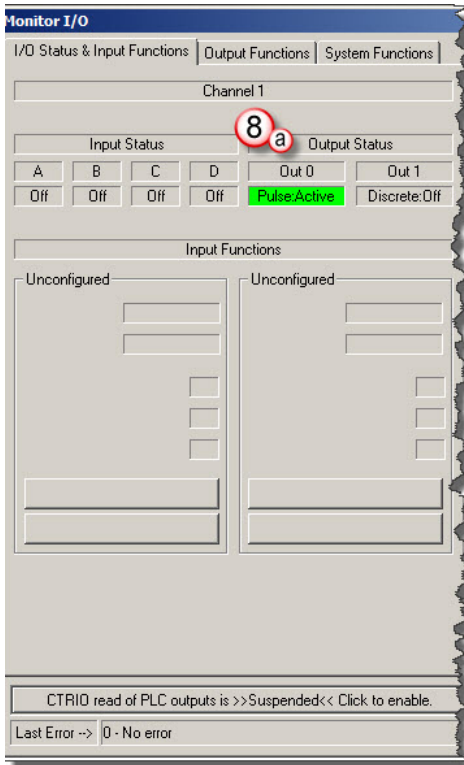
- 1: For **Command**, select **0x10 – Load Table**.
Command “0x10” is used for multiple functions: Load Preset Table, Load PLS or Load a Pulse Profile. We are loading a pulse profile in this example.
- 2: For **File Number**, select 1. File 1 is the Trapezoid Plus Pulse Profile configuration that was entered earlier. This is Parameter 1 in the flow chart and in the I/O mapping.
- 3: Press **Process Command** to tell the CTRIO(2) to act on the variables selected in steps 1 - 2.
- 4: The CTRIO(2) acts on the **Process Command** and reports its result as ‘success’ by setting the **Command Complete** status bit. This means the CTRIO(2) has successfully loaded File 1. If the CTRIO(2) had been unable to load the file for any reason, the **Command Error** status bit would have come on as well.



NOTE: Loading a table is not instantaneous within the CTRIO(2). What the CTRIO(2) just did here took enough time that several CPU scans would lapse before Command Complete status bit turned On. Ladder code needs to be written to account for the asynchronous nature of interactions with the CTRIO(2). Stage is well suited for writing sequential asynchronous code.

- 5: Press **Process Command** again to release (reset) it. The CTRIO(2) clears the **Command Complete** status bit.
- 6: For **Position**, type in the distance (total number of pulses) desired for the move. This is Parameter 3 in the flow chart and in the I/O mapping. The sign of this parameter sets the direction of the move. A positive value will keep Output 1 (Dir) off (With this profile, the **Direction** button (bit) is not used. Some other profiles use the **Direction** bit instead of the sign.).
- 7: Press **Enable Output** to start the move.

- 8: The CTRIO(2):
 - a. Turns On the Output Active and Output Enabled status bits
 - b. Generates the appropriate pulse train on Output 0 and Output 1
 - c. Turns Off the Output Active bit when the move completes



- 9: Press the Enable Output button again to release (reset) it. The CTRIO(2) turns Off the Output Enabled status bit.
- 10: To initiate another move, repeat steps 6 – 9.

Ladder Example

A detailed ladder example is provided in *Chapter 9: Output Functions* later in this manual.

INSTALLATION AND FIELD WIRING



CHAPTER 3

In This Chapter...

Installing the H0-CTRIO(2) Module	3-2
Setting H0-CTRIO(2) Jumpers	3-3
H0- CTRIO(2) Quadrature Encoder Wiring Example	3-5
H0-CTRIO(2) TTL Quadrature Encoder Field Wiring	3-6
H0- CTRIO(2) TTL Input Wiring	3-7
H0- CTRIO(2) Output Wiring Schematic.....	3-8
H0-CTRIO(2) Stepper/Servo Drive Wiring Example	3-9
Solid State Input Device Wiring to the H0-CTRIO(2) Module	3-10
Installing the H2-CTRIO(2) Module	3-11
Setting H2-CTRIO(2) Jumpers	3-12
Wiring the H2-CTRIO(2) Module	3-13
H2- CTRIO(2) Quadrature Encoder Wiring Example	3-14
H2-CTRIO(2) TTL Quadrature Encoder Field Wiring	3-15
H2-CTRIO(2) TTL Input Wiring	3-16
H2- CTRIO(2) Output Wiring Schematic.....	3-17
H2-CTRIO(2) Stepper/Servo Drive Wiring Example	3-18
Solid State Input Device Wiring to the H2-CTRIO(2) Module	3-19
Installing the H4-CTRIO.....	3-20
Wiring the H4-CTRIO Module	3-21
H4-CTRIO Quadrature Encoder Wiring Example	3-22
H4-CTRIO TTL Quadrature Encoder Field Wiring	3-23

H4-CTRIO TTL Input Wiring	3-24
H4-CTRIO Output Wiring Schematic.....	3-25
H4-CTRIO Stepper/Servo Drive Wiring Example.....	3-26
Solid State Input Device Wiring to the H4-CTRIO Module	3-27
Installing the T1H-CTRIO.....	3-28
Wiring the T1H-CTRIO Module	3-29
T1H-CTRIO Quadrature Encoder Wiring Example	3-31
T1H-CTRIO TTL Quadrature Encoder Field Wiring	3-32
T1H-CTRIO TTL Input Wiring	3-33
T1H-CTRIO Output Wiring Schematic.....	3-34
T1H-CTRIO Stepper/Servo Drive Wiring Example.....	3-35
Solid State Input Device Wiring to T1H-CTRIO Module	3-36

Installing the H0-CTRIO(2) Module

The H0-CTRIO(2) module is compatible with *Direct*LOGIC DL05 and DL06 PLCs. Consideration must be given to the firmware versions of the PLCs to assure their compatibility with the H0-CTRIO(2) (see chart below).



SPECIAL NOTE: For applications requiring multiple CTRIO modules, *Direct*LOGIC CPUs, and dynamic access (in ladder logic) to CTRIO data, we recommend using the D2-250-1, D2-260 or DL06 CPU. These CPUs support Bit-of-Word addressing, 32 bit math instructions and have adequate memory for multiple CTRIO applications.

The H0-CTRIO(2) module plugs into any option card slot in the DL05 and DL06 base.

For installation instructions, refer to the DL05 or DL06 User Manual (D0-USER-M or D0-06USER-M).

PLC CPU	Firmware	<i>Direct</i> SOFT
H0-CTRIO(2)		
DL05	v. 4.60 or later	v. 4.0, Build 16 or later
DL06	v. 1.40 or later	v. 4.0, Build 16 or later

At first power-up the CTRIO(2) module, the OK LED will be blinking. The blinking LED indicates that the module is in program mode.

CPU and CTRIO Compatibility Chart

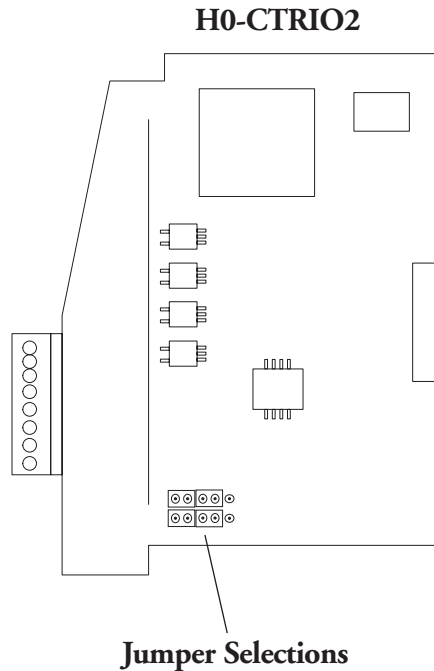
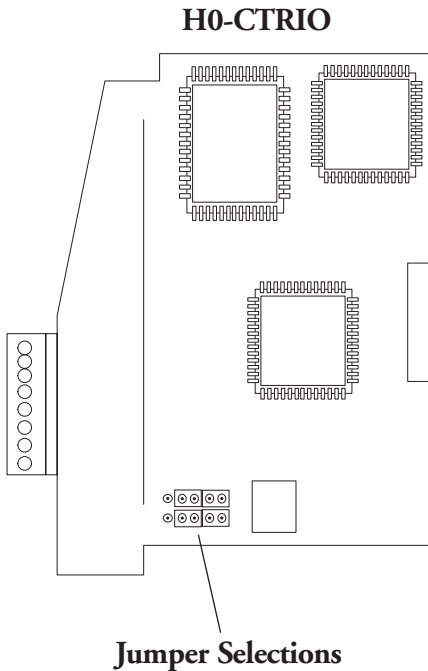
Updated firmware versions can be downloaded from our web site at www.automationdirect.com.



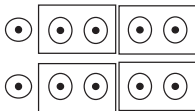
NOTE: CTRIO Workbench Version 2.2.0 or later is required for the H0-CTRIO2.

Setting H0-CTRIO(2) Jumpers

The module internal jumpers must be set to the **High Common** position for high side switching (sourcing) outputs or to the **Low Common** position for low side switching (sinking) outputs. The sink/source jumper selection sets both outputs to the same option. Source operation is the factory default setting.

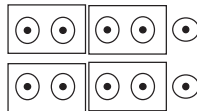


Sourcing Outputs



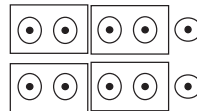
High Common position for switching the high side of a DC load.

Sinking Outputs



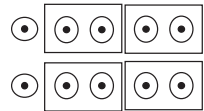
Low Common position for switching the low side of a DC load.

Sourcing Outputs



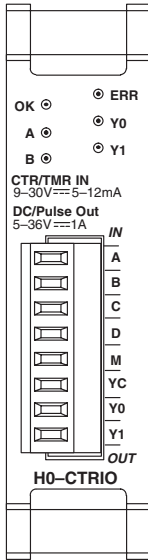
High Common position for switching the high side of a DC load.

Sinking Outputs



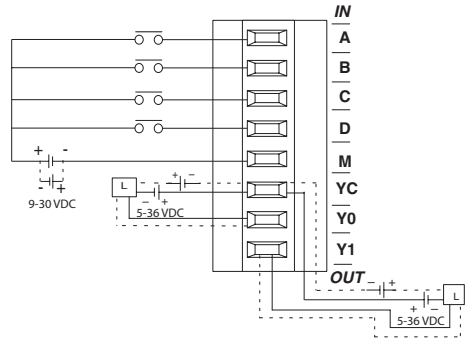
Low Common position for switching the low side of a DC load.

Wiring the H0-CTRIO(2) Module



The H0-CTRIO(2) module has one input channel, consisting of 4 optically isolated input points (A–D on common M). The inputs can be wired to either sink or source current.

The module has 2 optically isolated output points (Y0–Y1 on common YC). The outputs can be wired to either sink or source current, but the sink/source jumper selection sets both outputs to the same option. Sourcing outputs must be wired so positive current flows into the YC terminal and then out of the Yn terminal. Sinking outputs must be wired so positive current flows into Yn terminal and then out of the YC terminal (see the diagram to the right and the schematic on page 3-9). Source operation is the factory default setting for the outputs.



The module is configured, using CTRIO Workbench, to accommodate the user’s application. The function of each input (counting, timing, reset, etc.) and output (pulse output, discrete output, etc.) is defined in the configuration of the module.

Refer to Chapters 5 and 6 to determine what input and output configurations are possible.



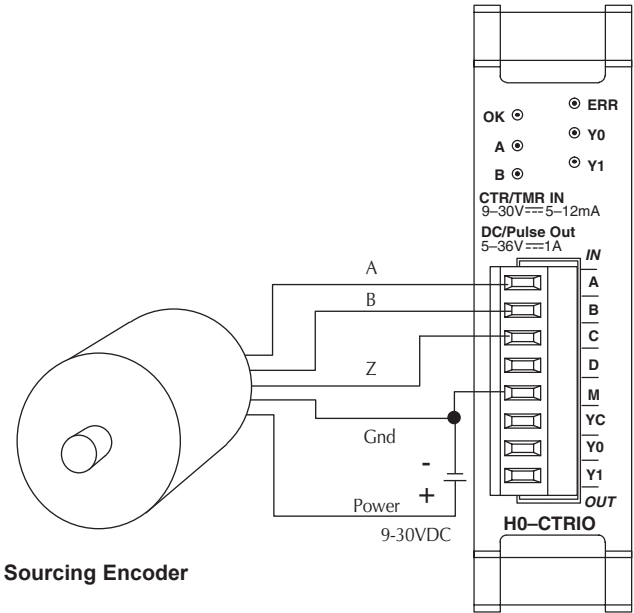
NOTE: Field device wiring must be compatible with the module configuration.

See the notes below for further details about power source considerations, circuit polarities, and field devices. Also, refer to the specifications in Chapter 1 on pages 1-7 and 1-8 for more information.

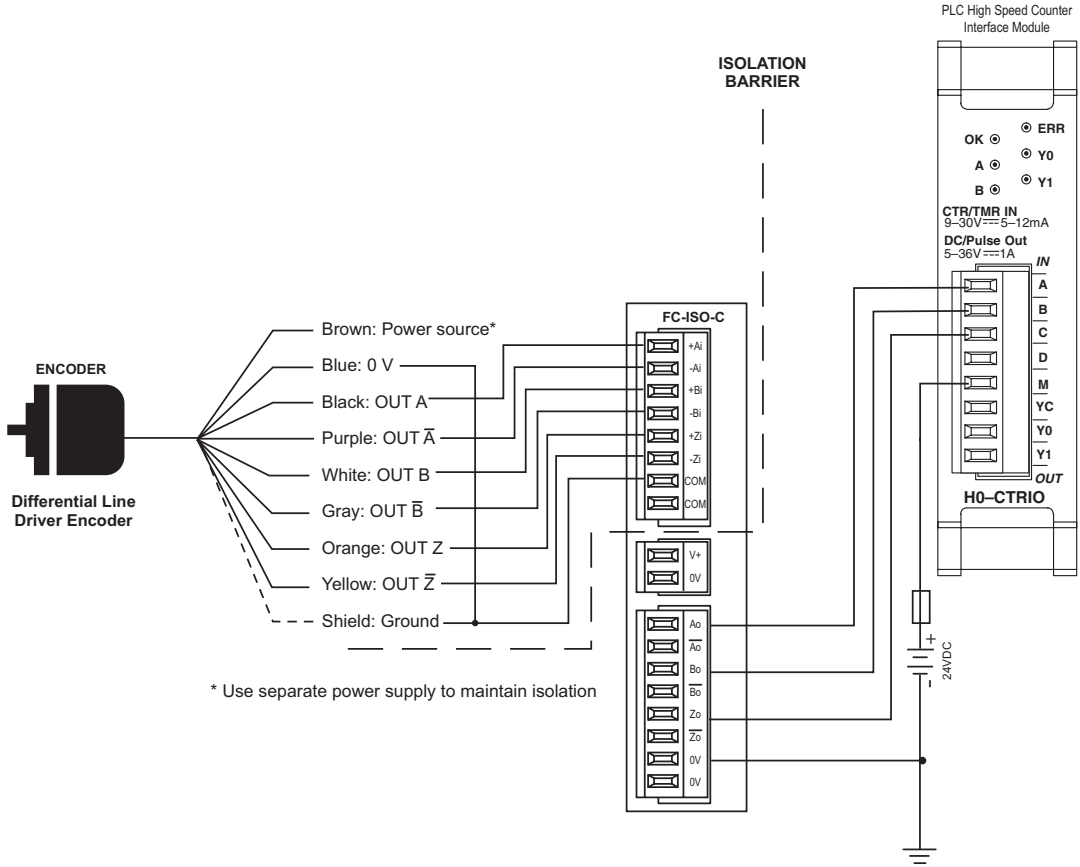
NOTES:

- 1: Inputs (A, B, C, and D) require user-provided 9–30VDC power sources. Terminal M is the common for the inputs. Maximum current consumption is 12mA per input point.
- 2: Polarity of the input power sources (shown above) can be reversed. Consideration must be given, however, to the polarity of the field device. Many field devices are designed for only one polarity and can be damaged if power wiring is reversed.
- 3: The maximum allowable current per output circuit is 1A.

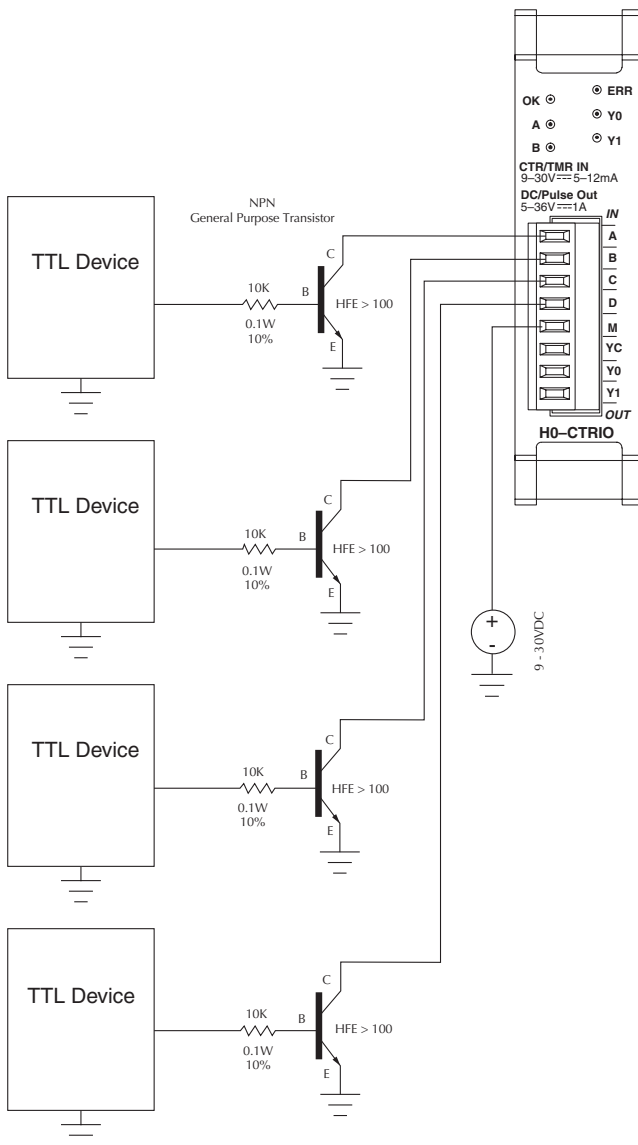
H0- CTRIO(2) Quadrature Encoder Wiring Example



H0-CTRIO(2) TTL Quadrature Encoder Field Wiring

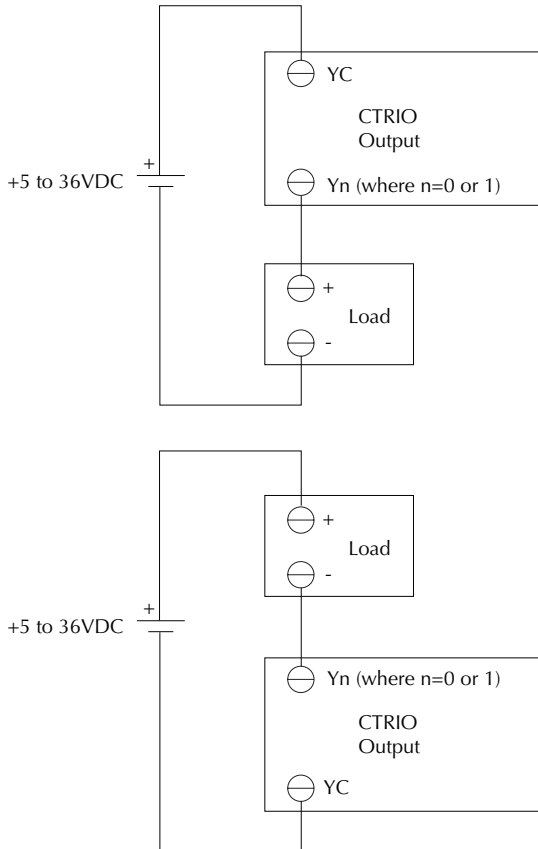


H0- CTRIO(2) TTL Input Wiring



H0- CTRIO(2) Output Wiring Schematic

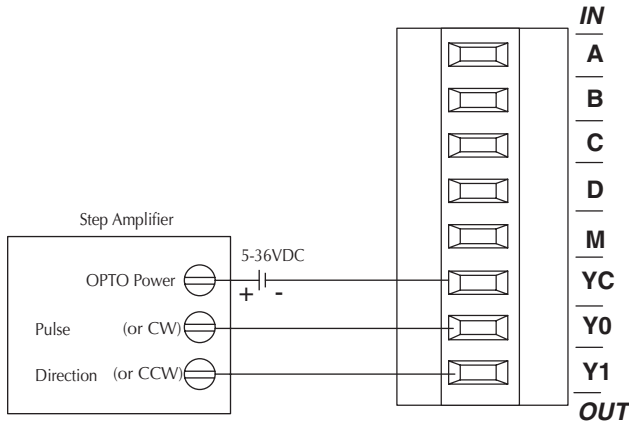
See page 3-3 for locating and setting the jumpers



H0-CTRIO(2) Stepper/Servo Drive Wiring Example



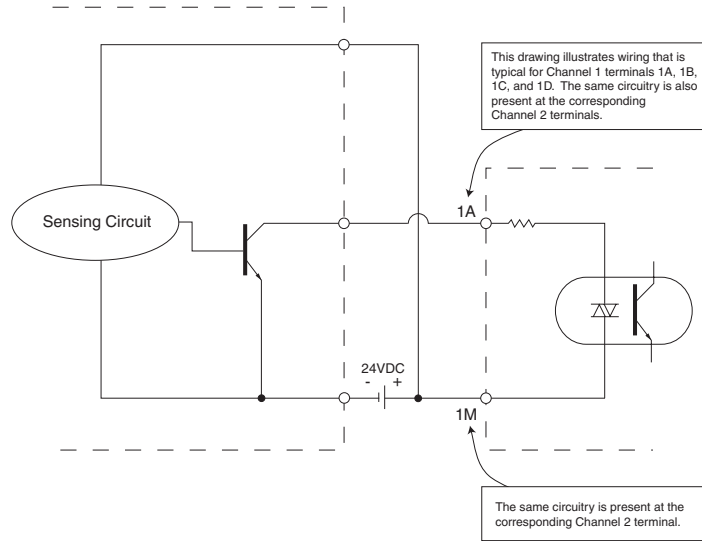
NOTE: Sinking output connection shown.



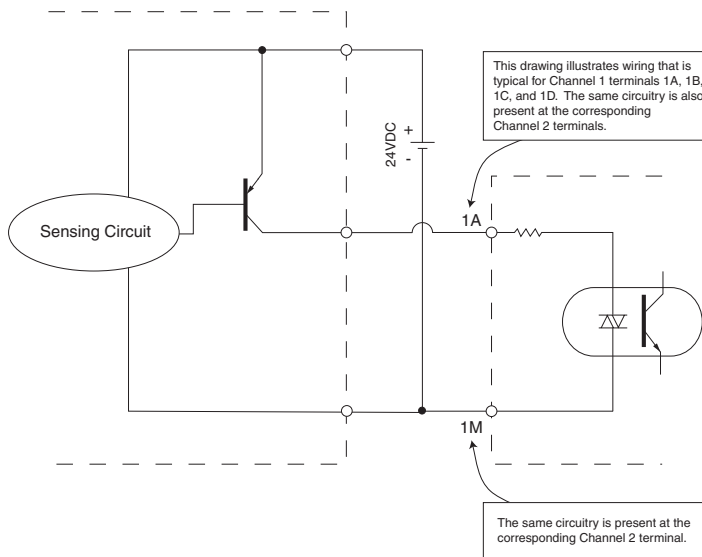
See page 3-3 for locating and setting the jumpers

Solid State Input Device Wiring to the H0-CTRIO(2) Module

NPN Field Device



PNP Field Device



Installing the H2-CTRIO(2) Module

The H2-CTRIO(2) module is compatible with Do-more CPUs and several DL205 CPU/slot interface devices. Consideration must be given to the firmware version of the CPU to assure their compatibility with the H2-CTRIO(2) (see chart below).

The H2-CTRIO(2) module plugs into any I/O slot of any Do-more or *Direct*LOGIC 205 base except slot 0 when using a *Direct*LOGIC PLC. Slot 0 is also not allowed if using the H2-CTRIO and a WinPLC or H2-PBC controller. However, slot 0 is available for the H2-CTRIO(2) module when using the H2-EBC interface devices (Slot 0 is the I/O slot adjacent to the CPU).



NOTE: The H2-CTRIO(2) cannot be used in DL205 local expansion bases or in Serial Remote I/O bases.

For installation instructions, refer to the:

- DL205 User Manual (D2-USER-M) if using a DirectLOGIC PLC
- DL205 Installation and I/O Manual (D2-INST-M) if using a WinPLC, EBC or PBC module.
- Do-more H2 series PLC Hardware User Manual (H2-DM-M) if using a Do-more PLC

At first power-up of the CTRIO(2) module, the OK LED will be blinking. The blinking LED indicates that the module is in program mode.

CPU and CTRIO Compatibility Chart

CPU-slot Device	Firmware	Hardware	<i>Direct</i> SOFT	Slot Restrictions
H2-CTRIO(2)				
D2-240	v. 3.22 or later	Any	v. 3.0C, Build 71 or later	any I/O slot except 0
D2-250	v. 1.56 or later	Any	v. 3.0C, Build 71 or later	any I/O slot except 0
D2-250-1	v. 3.5 or later	Any	v. 3.0C, Build 71 or later	any I/O slot except 0
D2-260	v. 1.2 or later	Any	v. 4.0 or later	any I/O slot except 0
H2-WinPLC	Any	xK or later	N/A	any I/O slot except 0
H2-EBC	v. 2.1.357 or later	Any	N/A	prior to Rev 9A any I/O slot except 0; Rev 9A or later any I/O slot
H2-PBC	Any	Any	N/A	prior to Rev 4A any I/O slot except 0; Rev 4A or later any I/O slot
Do-more	Any	Any	N/A	No Restrictions



NOTE: CTRIO Workbench version 2.2.0 or later is required for H2-CTRIO2. However, with a Do-more CPU, CTRIO Workbench is not used at all. Instead, the Do-more Designer Module Configuration is used to configure the H2-CTRIO2.

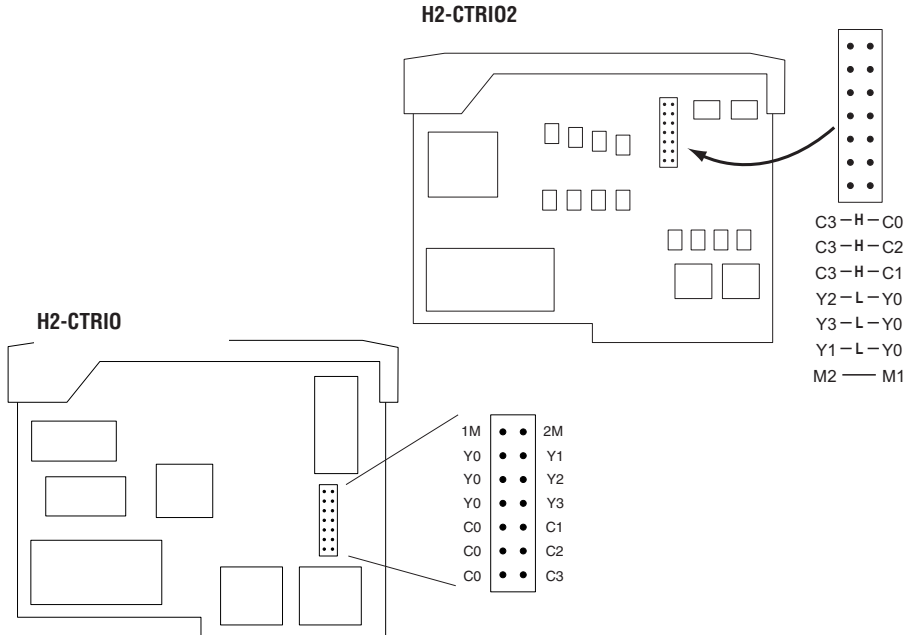
Updated firmware versions can be downloaded from our web site at www.automationdirect.com.

Setting H2-CTRIO(2) Jumpers



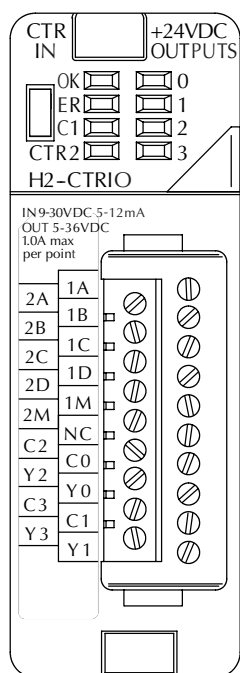
SPECIAL NOTE: For applications requiring multiple CTRIO modules, DirectLOGIC CPUs, and dynamic access (in ladder logic) to CTRIO data, we recommend using the D2-250-1 or D2-260 CPU. These CPUs support Bit-of-Word addressing, 32 bit math instructions and have adequate memory for multiple CTRIO applications.

Jumpers are provided to connect input commons or outputs/output commons. Use of these jumpers is not necessary to set up the CTRIO(2) module. The jumpers are provided solely for convenience in wiring.

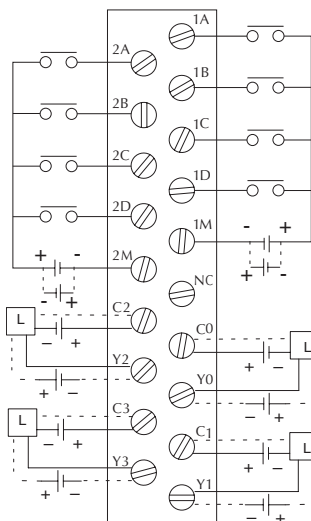


H2-CTRIO(2) Jumper Functions		
H2-CTRIO	H2-CTRIO2	Function
1M-2M		Install jumper to internally connect the input commons 1M and 2M in order to reduce wiring if appropriate.
Y0-Y1		Install jumper(s) to internally connect Y0 to other Y terminals in order to reduce wiring if appropriate. Connect wire at Y0.
Y0-Y2		
Y0-Y3		
C0-C1		Install jumper(s) to internally connect C0 to other C terminals in order to reduce wiring if appropriate. Connect wire at C0.
C0-C2		
C0-C3		
	C3-C0	Install jumper(s) to internally connect C3 to other C terminals in order to reduce wiring if appropriate. Connect wire at C3.
	C3-C1	
	C3-C2	

Wiring the H2-CTRIO(2) Module



The H2-CTRIO(2) module has two independent input channels, each consisting of 4 optically isolated input points (1A–1D on common 1M and 2A–2D on common 2M). The inputs can be wired to either sink or source current. The module has 4 optically isolated output points (pts. Y0–Y3 with isolated commons C0–C3, respectively). The outputs must be wired so positive current flows into C(n) terminal and then out of the Y(n) terminal (see diagram



at left and the schematic on page 3-17). Remember that the internal jumpers can be used to connect the input commons or outputs/output commons together.

The module is configured, using CTRIO Workbench, to accommodate the user's application. The function of each input (counting, timing, reset, etc.) and output (pulse output, discrete output, etc.) is defined in the configuration of the module.

Refer to Chapters 5 and 6 to determine what input and output configurations are possible.

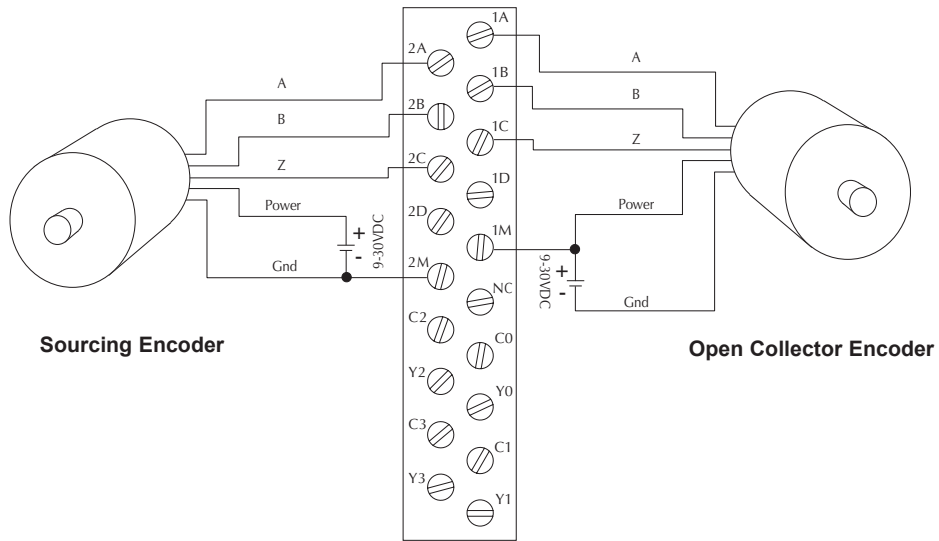
NOTE: Field device wiring must be compatible with the module configuration.

See the notes below for further details about power source considerations, circuit polarities, and field devices.

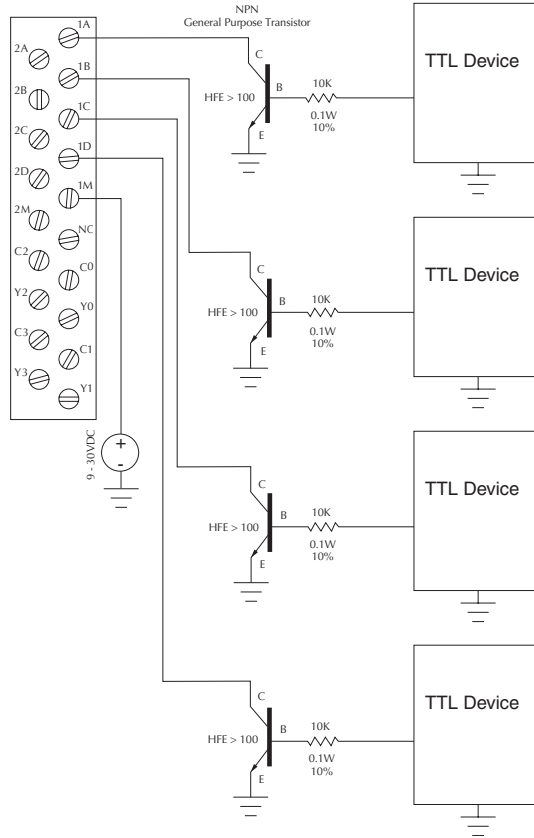
NOTES:

- 1: Inputs (1A, 1B, 1C, 1D and 2A, 2B, 2C, 2D) require user-provided 9-30VDC power sources. Terminals 1M and 2M are the commons for Channel 1 and Channel 2 inputs. Maximum current consumption is 12mA per input point.
- 2: Polarity of the input power sources (shown above) can be reversed. Consideration must be given, however, to the polarity of the field device. Many field devices are designed for only one polarity and can be damaged if power wiring is reversed.
- 3: Outputs have one polarity only (as shown above) and are powered by user-provided 5-36VDC power sources. The maximum allowable current per output circuit is 1A for the H2-CTRIO and 1A at 23°C or 0.5A at 60°C for the H2-CTRIO2.

H2- CTRIO(2) Quadrature Encoder Wiring Example

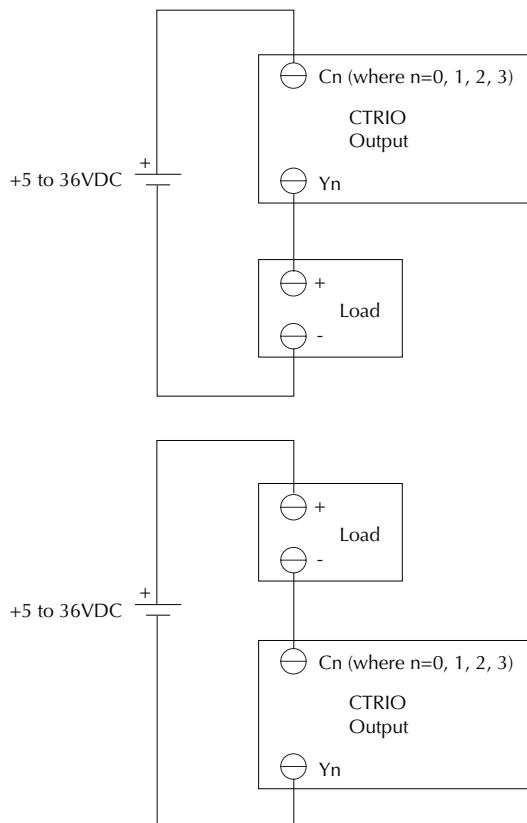


H2-CTRIO(2) TTL Input Wiring



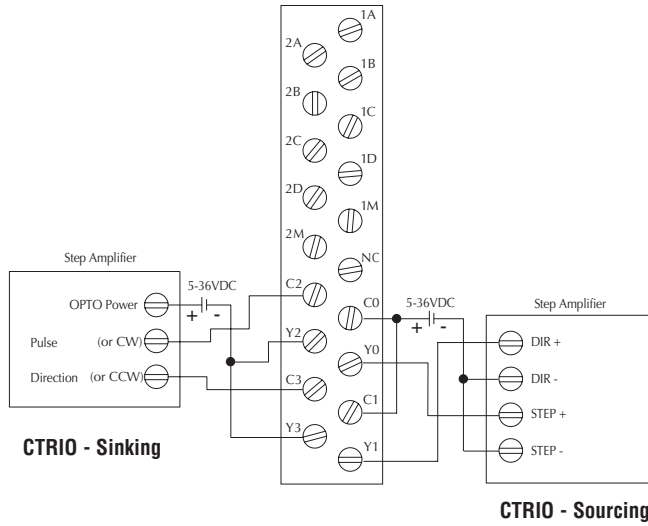
H2- CTRIO(2) Output Wiring Schematic

The CTRIO outputs are individually isolated DC switches that can be used to break the high or the low side of a DC load.



NOTE: The outputs must be wired so positive current flows into C(n) terminal and then out of the Y(n) terminal.

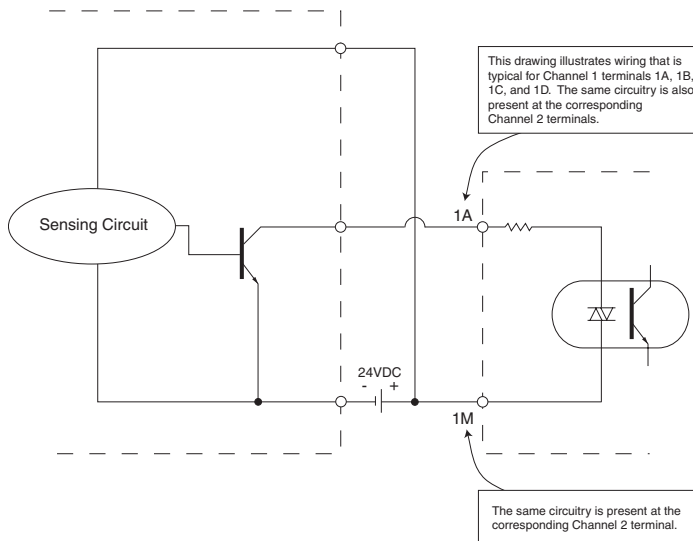
H2-CTRIO(2) Stepper/Servo Drive Wiring Example



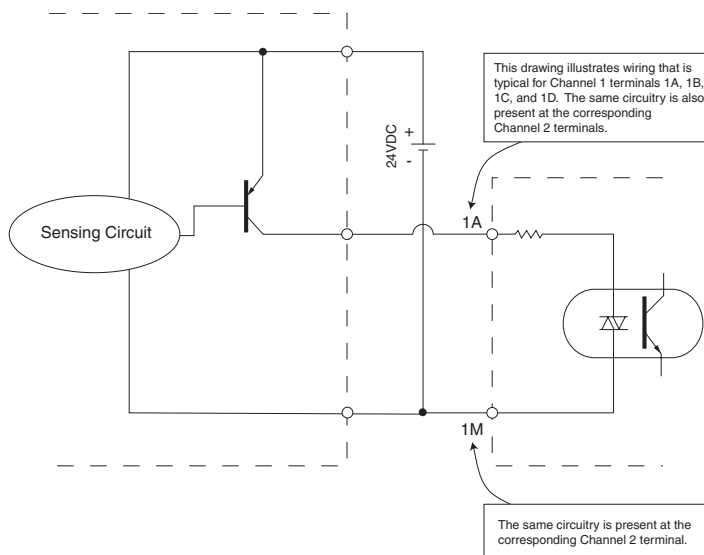
This example assumes that the Step Amplifier interface to be optocoupler LEDs (common anodes at the “OPTO Power” terminal) with internal current limiting resistors. This is a standard method; consult the step amplifier documentation to ensure that this method is applicable.

Solid State Input Device Wiring to the H2-CTRIO(2) Module

NPN Field Device



PNP Field Device



Installing the H4-CTRIO

The H4-CTRIO module is compatible with two DL405 CPU/slot interface devices. Consideration must be given to the firmware versions of the CPU-slot interfaces to assure their compatibility with the H4-CTRIO, (see chart below).

The H4-CTRIO module plugs into any I/O slot of any *Direct*LOGIC 405 base. H4-EBCs support the use of the H4-CTRIO in DL405 local expansion bases. The H4-CTRIO cannot be used in Serial Remote I/O bases.

For installation instructions, refer to the:

- DL405 User Manual (D4-USER-M) if using a *Direct*LOGIC PLC
- DL405 Installation and I/O Manual (D4-INST-M) if using an H4-EBC interface

At first power-up of the CTRIO module, the OK LED will be blinking. The blinking LED indicates that the module is in program mode.

CPU and CTRIO Compatibility Chart

CPU-slot Device	Firmware*	Hardware	<i>Direct</i> SOFT
D4-450	April 2000 or earlier: H8 (CISC) v. 2.00 SH (RISC) v. 1.500 May 2000 or later: H8 (CISC) v. 2.00 SH (RISC) v. 2.500	Any	v. 4.0, Build 16 or later
H4-EBC	2.1.328 or later	v. 4F or later	N/A

*Updated firmware versions can be downloaded from our web site at www.automationdirect.com

Wiring the H4-CTRIO Module

The H4-CTRIO module has two independent input channels, each consisting of 4 optically isolated input points (1A–1D on common 1M and 2A–2D on common 2M). The inputs can be wired to either sink or source current. The module has 4 optically isolated output points (Y0–Y3 on isolated commons C0–C3, respectively). The outputs must be wired so that positive current flows into Cn terminal and then out of the Yn terminal (See the diagram below and the schematic on page 3-25).

The module is configured, using CTRIO Workbench, to accommodate the user's application. The function of each input (counting, timing, reset, etc.) and output (pulse output, discrete output, etc.) is defined in the configuration of the module.

Refer to Chapters 5 and 6 to determine possible input and output configurations.

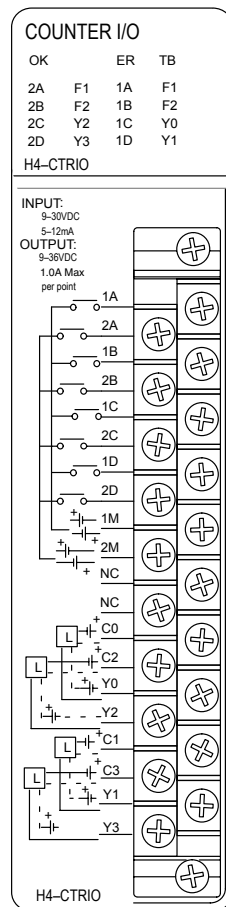
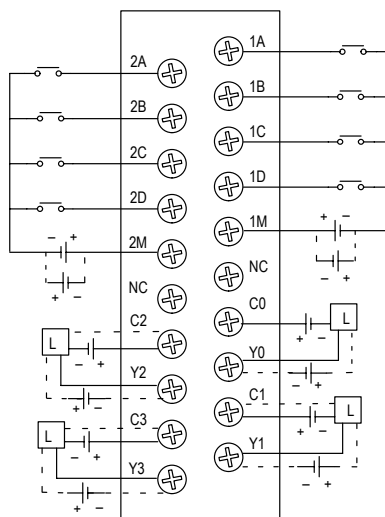


NOTE: Field device wiring must be compatible with the module configuration.

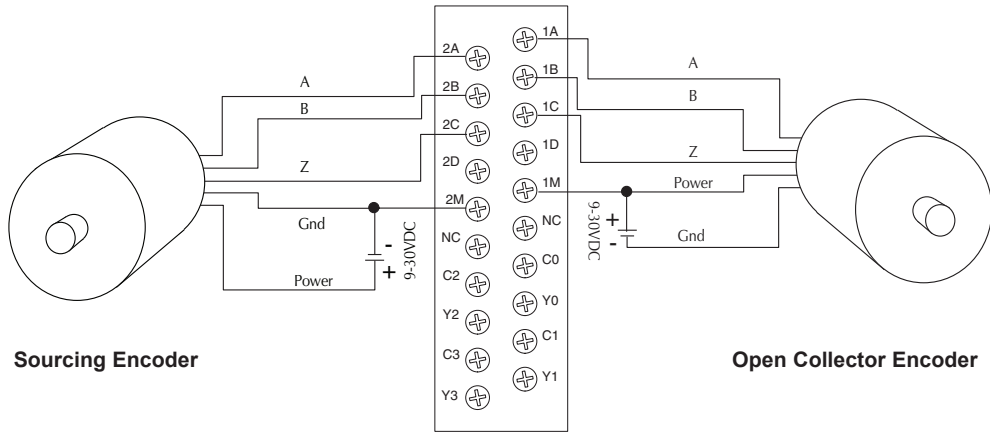
See the notes below for further details about power source considerations, circuit polarities, and field devices. Also, refer to the specifications in Chapter 1 on pages 1-7 and 1-8 for more information.

NOTES:

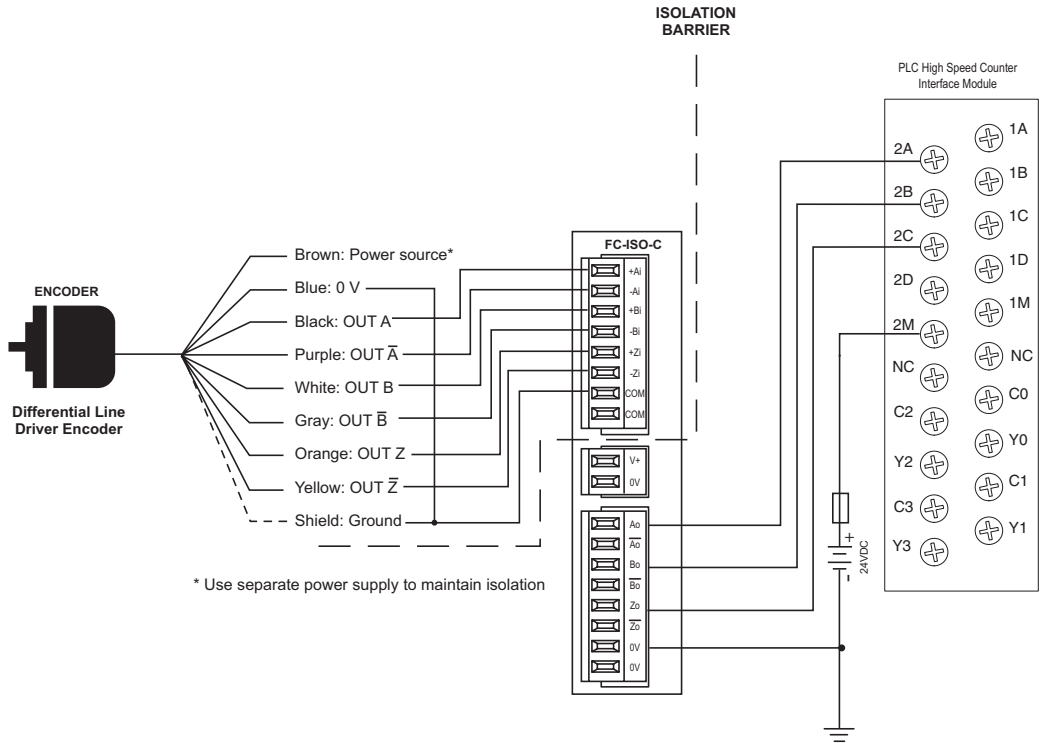
- 1: Inputs (1A, 1B, 1C, 1D and 2A, 2B, 2C, 2D) require user-provided 9–30VDC power sources. Terminals 1M and 2M are the commons for Channel 1 and Channel 2 inputs. Maximum current consumption is 12mA per input point.
- 2: Polarity of the input power sources (shown above) can be reversed. Consideration must be given, however, to the polarity of the field device. Many field devices are designed for only one polarity and can be damaged if power wiring is reversed.
- 3: Outputs have one polarity only (as shown above) and are powered by user-provided 5–36 VDC power sources. The maximum allowable current per output circuit is 1A.



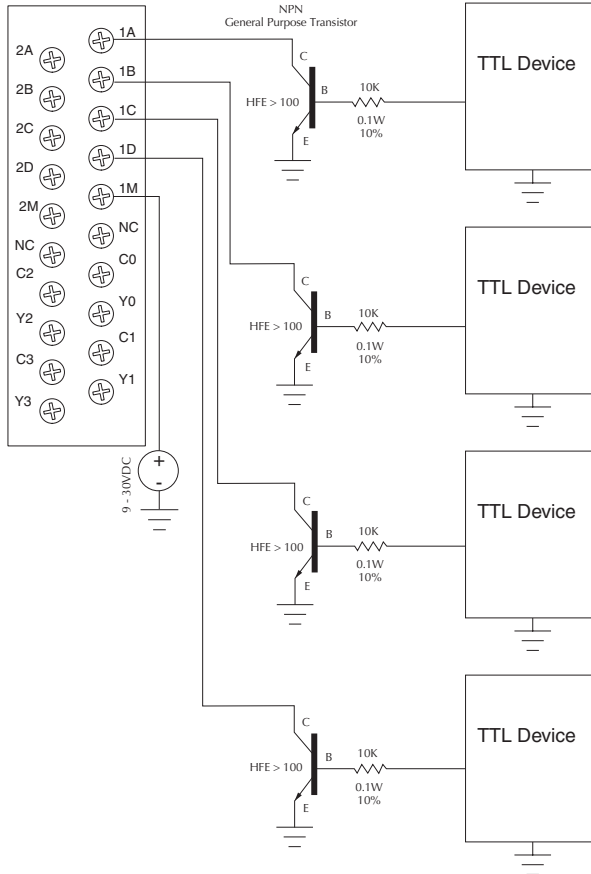
H4-CTRIO Quadrature Encoder Wiring Example



H4-CTRIO TTL Quadrature Encoder Field Wiring

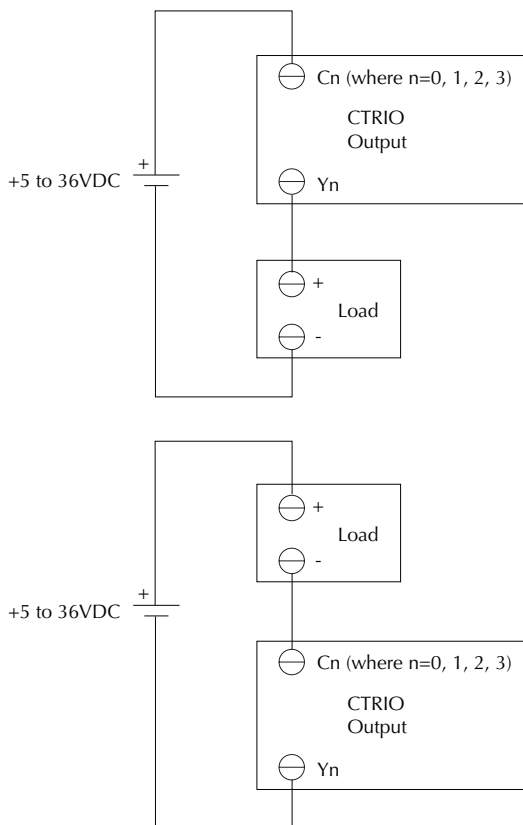


H4-CTRIO TTL Input Wiring



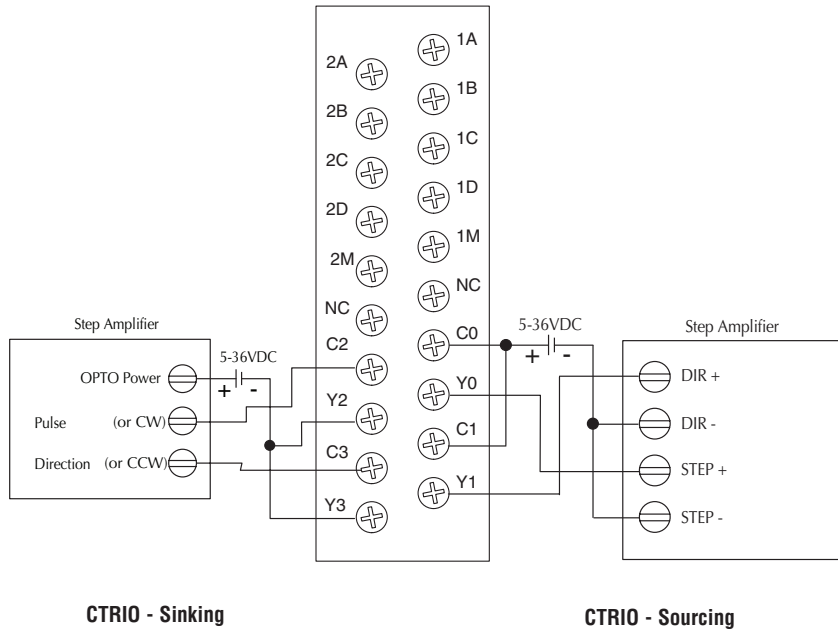
H4-CTRIO Output Wiring Schematic

The CTRIO outputs are individually isolated DC switches that can be used to break the high or the low side of a DC load.



NOTE: The outputs must be wired so that positive current flows into C_n terminal and then out of the Y_n terminal.

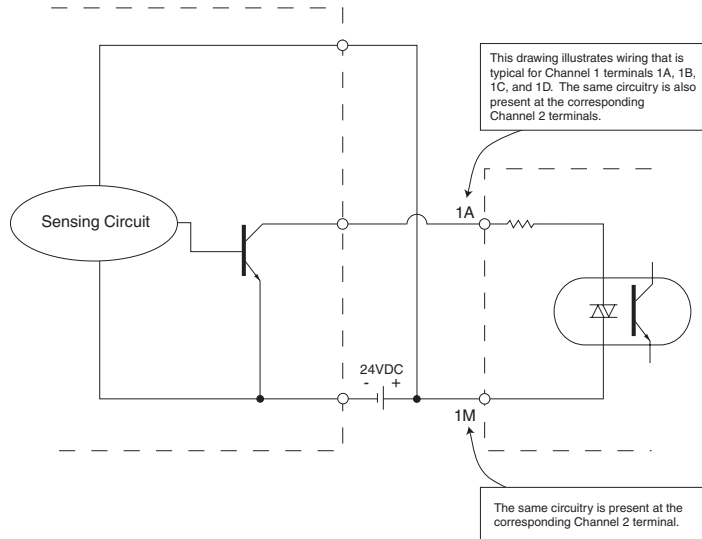
H4-CTRIO Stepper/Servo Drive Wiring Example



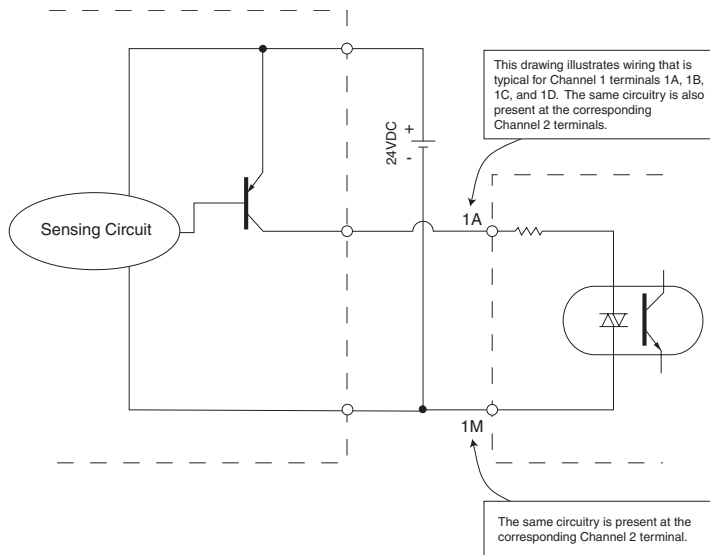
This example assumes that the Step Amplifier interface to be optocoupler LEDs (common anodes at the “OPTO Power” terminal) with internal current limiting resistors. This is a standard method; consult the step amplifier documentation to ensure that this method is applicable.

Solid State Input Device Wiring to the H4-CTRIO Module

NPN Field Device



PNP Field Device



Installing the T1H-CTRIO

The T1H-CTRIO module is compatible with several Terminator I/O Network interface devices. Consideration must be given to the firmware versions of the Network interfaces to assure their compatibility with the T1H-CTRIO, (see chart below).



NOTE: *The T1H-CTRIO is only supported by the T1H-EBC, T1H-EBC100 and T1H-PBC.*

The T1H-CTRIO module plugs into any valid I/O slot in a Terminator I/O system. The T1H-CTRIO cannot be used with the Network interfaces: T1K-RSSS, T1K-MODBUS and T1K-DEVNETS.

For installation instructions, refer to the Terminator I/O Installation and I/O Manual (T1K-INST-M)

CPU and CTRIO Compatibility Chart

CPU-slot Device	Firmware*	Hardware
T1H-EBC	v. 1.0.444 or later	v. 2I or later
T1H-EBC100	Any	Any
T1H-PBC	v. 1.1.10 or later	v. 2D or later

*Updated firmware versions can be downloaded from our web site at www.automationdirect.com

At first power-up of the CTRIO module, the OK LED will be blinking. The blinking LED indicates that the module is in program mode.

Wiring the T1H-CTRIO Module

The T1H-CTRIO module has two independent input channels, each consisting of 4 optically isolated input points (1A–1D on common 1M and 2A–2D on common 2M). The inputs can be wired to either sink or source current.

The module has 4 optically isolated output points (Y0–Y3 on isolated commons C0–C3, respectively). The outputs must be wired so that positive current flows into Cn terminal and then out of the Yn terminal, (see the diagram on the following page and the schematic on page 3-34).

The module is configured, using CTRIO Workbench, to accommodate the user’s application. The function of each input (counting, timing, reset, etc.) and output (pulse output, discrete output, etc.) is defined in the configuration of the module.

Refer to Chapters 5 and 6 to determine what input and output configurations are possible.



NOTE: Field device wiring must be compatible with the module configuration.

See the notes below for further details about power source considerations, circuit polarities, and field devices. Also, refer to the specifications in Chapter 1 on pages 1-7 and 1-8 for more information.

Apply the labels that come with the I/O module to the I/O base terminals to properly identify the base terminal points.

Input / Output Channels

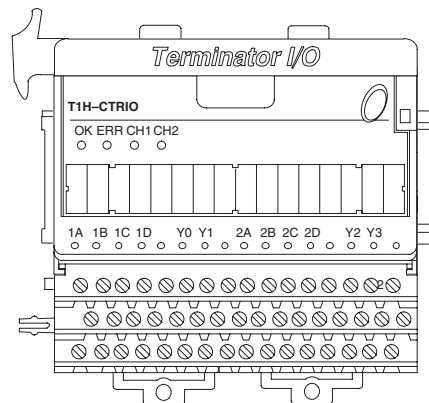
⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
1A	1B	1C	1D	Y0	Y1	Y1	Y1	2A	2B	2C	2D	Y2	Y3
□	□	□	□	□	□	□	□	□	□	□	□	□	□

Channel Commons

⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
1M	1M	1M	1M	C0	C0	C1	C1	2M	2M	2M	2M	C2	C3
□	□	□	□	□	□	□	□	□	□	□	□	□	□

User Bus Terminals (no internal connection to CTRIO)

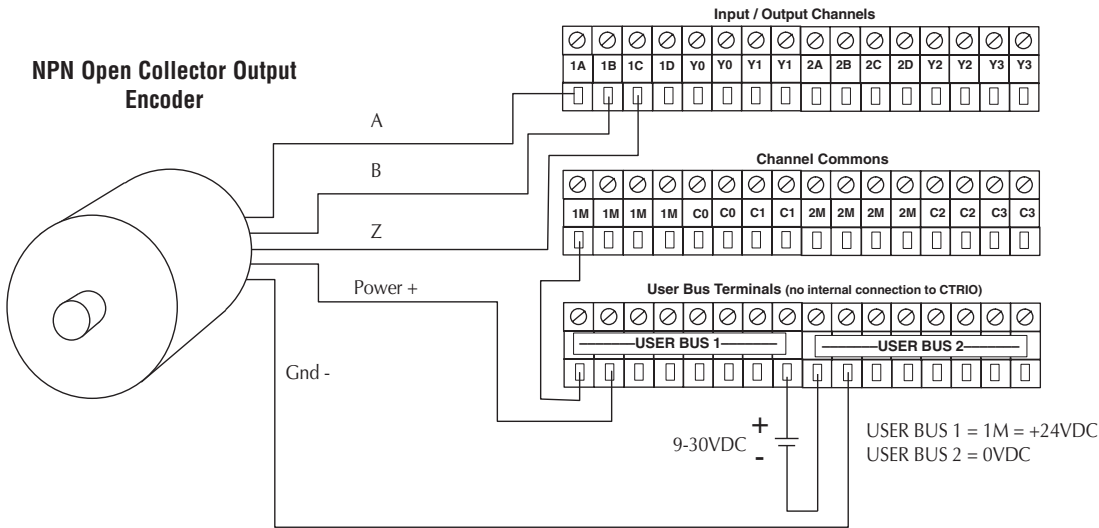
⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗	⊗
USER BUS 1				USER BUS 2									
□	□	□	□	□	□	□	□	□	□	□	□	□	□



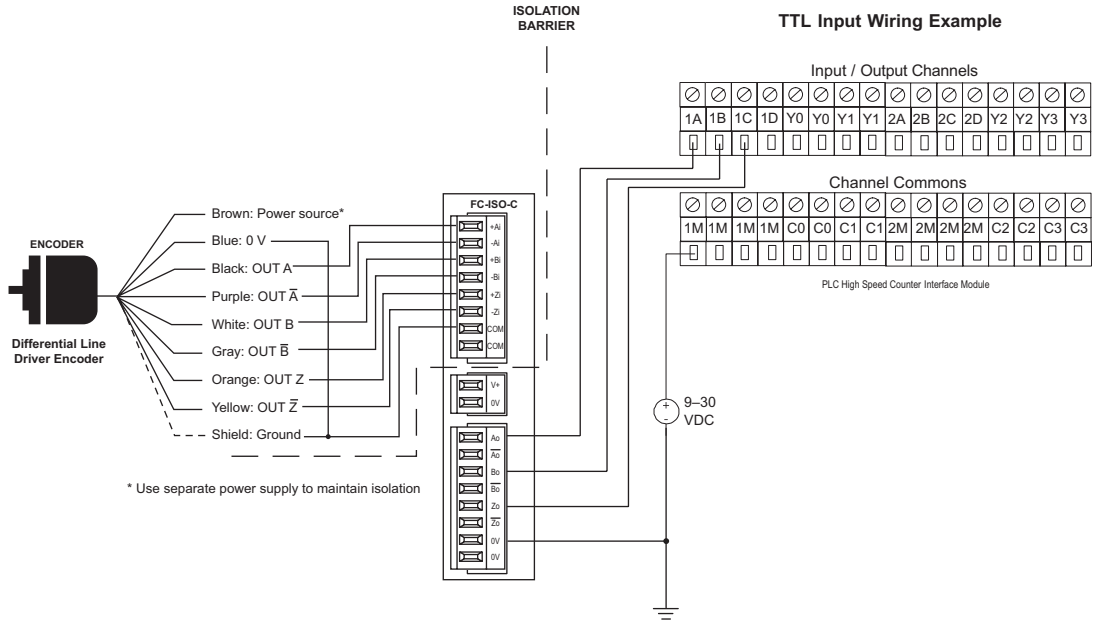
NOTES:

- 1: Inputs (1A, 1B, 1C, 1D and 2A, 2B, 2C, 2D) require user-provided 9–30VDC power sources. Terminals 1M and 2M are the commons for Channel 1 and Channel 2 inputs. Maximum current consumption is 12mA per input point.
- 2: Polarity of the input power sources can be reversed. Consideration must be given, however, to the polarity of the field device. Many field devices are designed for only one polarity and can be damaged if power wiring is reversed.
- 3: Outputs have one polarity only and are powered by user-provided 5–36VDC power sources. The maximum allowable current per output circuit is 1A.
- 4: User Bus 1 and User Bus 2 are each an independent 8 wiring terminal bus. They can be used for additional power rail connections.

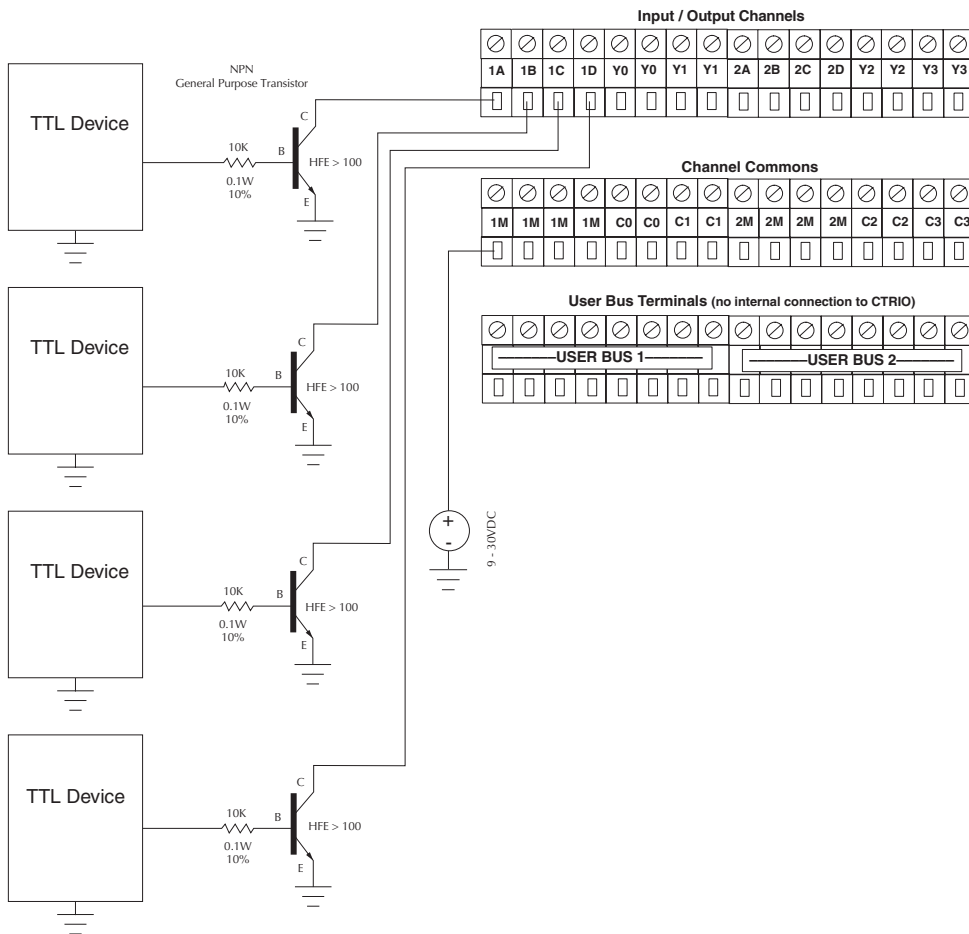
T1H-CTRIO Quadrature Encoder Wiring Example



T1H-CTRIO TTL Quadrature Encoder Field Wiring

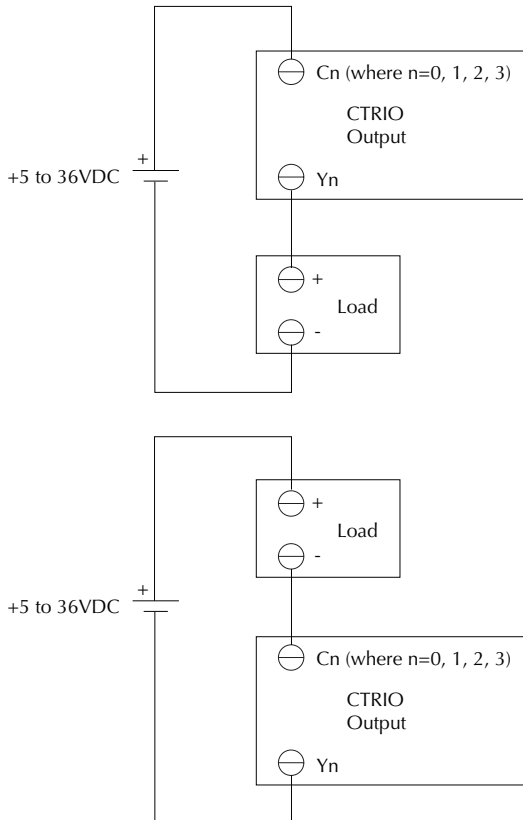


T1H-CTRIO TTL Input Wiring



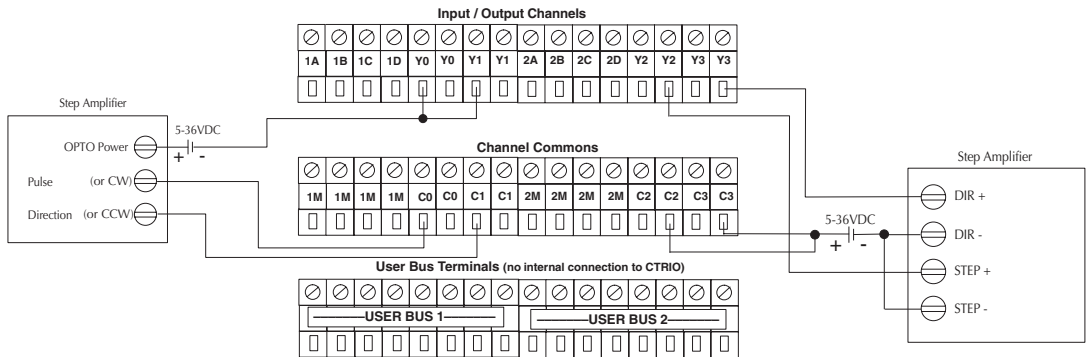
T1H-CTRIO Output Wiring Schematic

The CTRIO outputs are individually isolated DC switches that can be used to break the high or the low side of a DC load.



NOTE: The outputs must be wired so that positive current flows into C_n terminal and then out of the Y_n terminal.

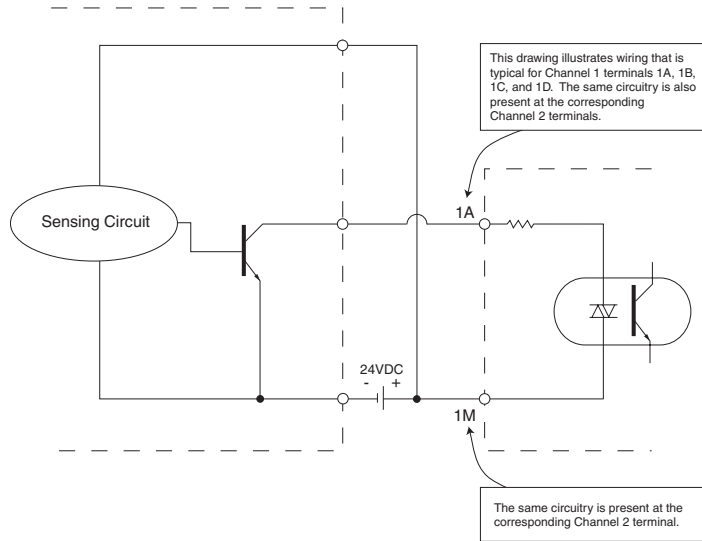
T1H-CTRIO Stepper/Servo Drive Wiring Example



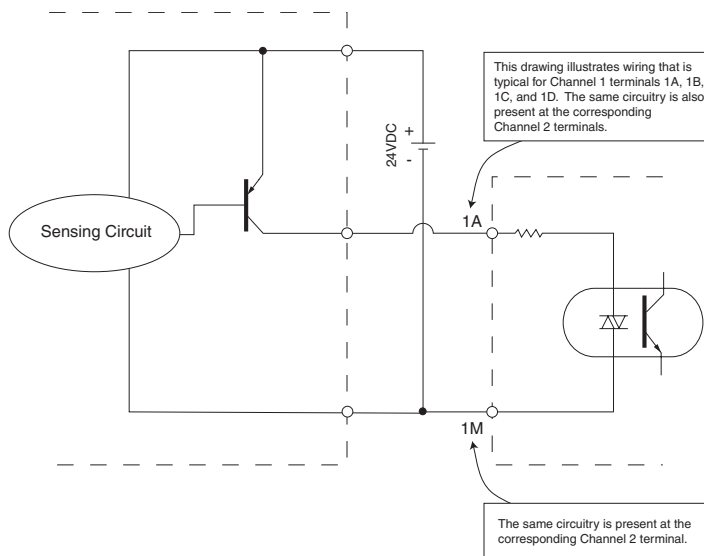
This example assumes that the Step Amplifier interface to be optocoupler LEDs (common anodes at the “OPTO Power” terminal) with internal current limiting resistors. This is a standard method; consult the step amplifier documentation to ensure that this method is applicable.

Solid State Input Device Wiring to T1H-CTRIO Module

NPN Field Device



PNP Field Device



CTRIO WORKBENCH, OVERVIEW



In This Chapter...

Configuring a CTRIO Module for Do-more CPUs.....	4-2
What is CTRIO Workbench?	4-2
Getting Started with CTRIO Workbench	4-3
Module Modes of Operation.....	4-6

Configuring a CTRIO Module for Do-more CPUs

NOTE: The functionality of CTRIO Workbench is integrated into the Do-more CPU software. Check Do-more documentation for help with CTRIO application.

What is CTRIO Workbench?

CTRIO Workbench is the software utility used to configure the CTRIO(2) module's inputs and outputs. Workbench provides a built-in scaling function to configure signals to desired engineering units, switch between the CTRIO(2) Program mode and Run mode, monitor I/O status and functions, and have diagnostic control of module functions.



NOTE: CTRIO Workbench Version 2.2.0 or later is required for the Hx-CTRIO2. Download the latest version of the CTRIO Workbench utility at no charge from the Host Engineering Web site: www.hosteng.com.

Installing CTRIO Workbench

The CTRIO Workbench utility installs directly from its executable file. Double click on the Setup.exe icon. The install shield will step through the installation process. The utility install` defaults into C:\HAPTtools directory. Find shortcuts to CTRIO Workbench from the Windows Start Menu under All Programs>AutomationDirect Tools.

Getting Started with CTRIO Workbench

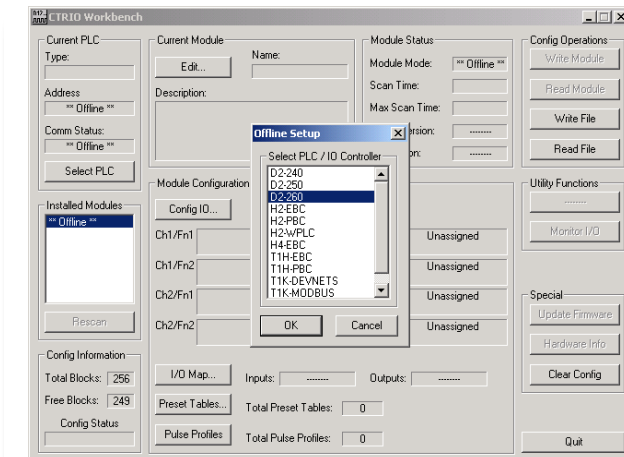
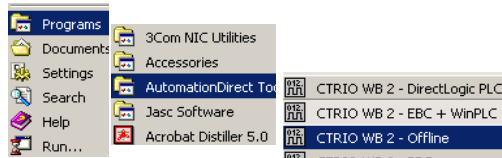
Several paths are available to start CTRIO Workbench. All users will find CTRIO Workbench at Start>Programs>AutomationDirect Tools>CTRIO Workbench. *DirectSOFT* users will find CTRIO Workbench in the Utilities section of the DSLaunch Window.

Offline CTRIO Configuration

A complete CTRIO configuration file (.cwb) can be created Offline.

To launch the CTRIO Workbench 2 Offline version, go to Start>Programs>AutomationDirect Tools>CTRIO WB2 - Offline.

In the Workbench Offline window, as shown below, click on the Select PLC button. Select desired PLC or interface device.



The only limitations in the Offline version are that Monitor I/O is not enabled and connecting to the CTRIO(2) from the Offline utility is not possible. Save the configuration file to disk and connect to the CTRIO(2) using the appropriate Workbench support version, then write the file to the CTRIO(2).

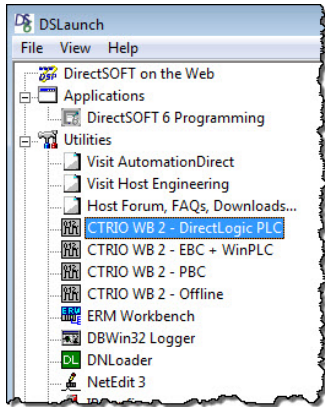
Online CTRIO Configuration

To configure the CTRIO(2) module Online, a CTRIO(2) must be installed in the PLC base or Terminator I/O system, and the system power must be on. The PC communicates with the CTRIO(2) module through the PLC or interface device port.

DirectSOFT

Connect the PC to the CPU, DCM or ECOM module. If linked to a CPU through *DirectSOFT*, CTRIO Workbench will start via the existing link. If disconnected from the PLC and open CTRIO Workbench, a prompt appears to establish a link to the CTRIO(2) module.

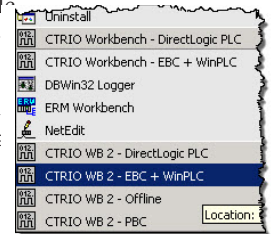
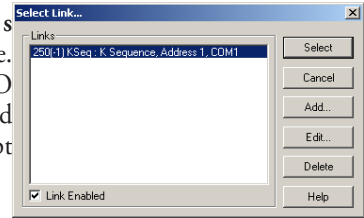
WinPLC, EBC and PLC>ERM>EBC Users



DirectSOFT users will find CTRIO Workbench in the DSLaunch Window's Utilities section.

Connect to the PC with the RJ45 Ethernet port on the WinPLC or EBC interface device directly or via hub, switch, etc. (Connect to the ST-style fiber optic port on the Hx-EBC-units).

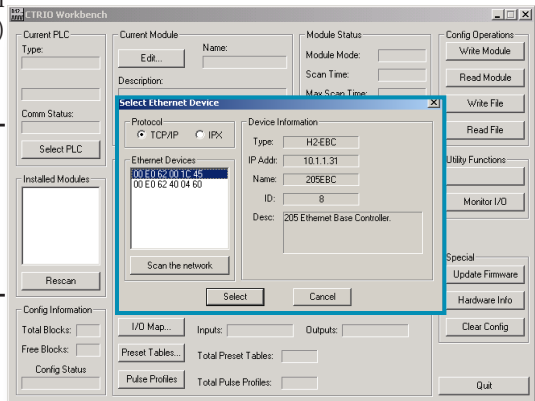
Access the WinPLC and EBC support version



Start>Programs>AutomationDirect Tools>CTRIO WB2 - EBC + WinPLC or select CTRIO WB2 - EBC + WinPLC in the *DirectSOFT* Launch Window Utilities menu.

A prompt will appear to establish an Ethernet link to the CTRIO(2) module.

NOTE: WinPLCs will need to be given an IP address before connecting with Workbench. EBCs will need to have an address selected by DIP Switch or via NetEdit before connecting with Workbench.

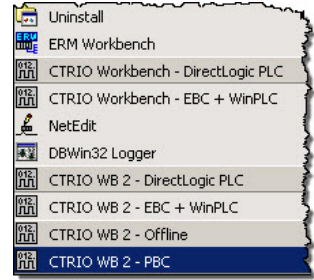


PBC Users

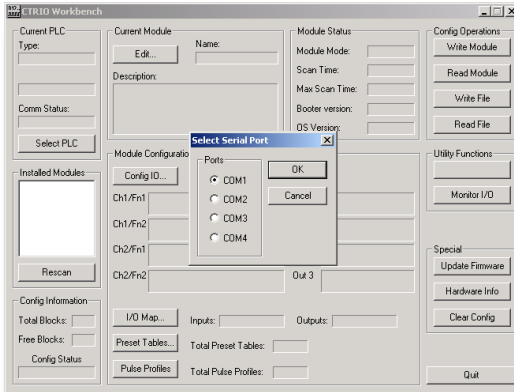
Connect PC to the RJ12 serial port on the PBC interface device.

Access the PBC support version at Start>Programs>AutomationDirect Tools>CTRIO WB2 - PBC or select CTRIO WB2 - PBC in the *DirectSOFT* Launch Window Utilities menu.

A prompt appears to establish a serial link to the CTRIO(2) module.

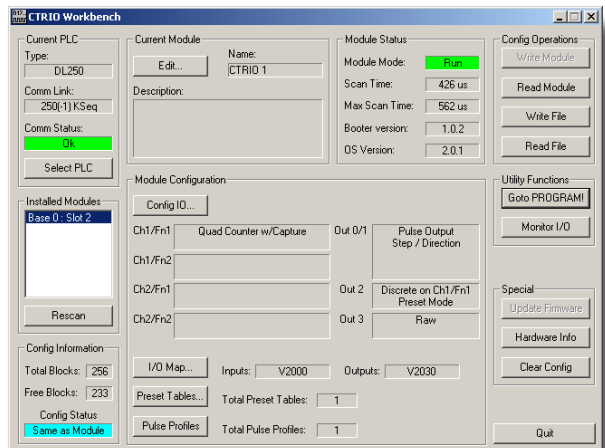


Select the PC serial port Workbench will use to connect to the CTRIO(2) module.



Successful On-line Connection

Once connected to the CTRIO(2) module, the main window of CTRIO Workbench is enabled. Here, select the CTRIO(2) module to configure by clicking on its slot number in the “Installed Modules” box. In Workbench’s Configuration dialog, after successfully configuring the module, toggle the CTRIO(2) module between Program Mode and Run Mode and enter the Monitor I/O dialog.

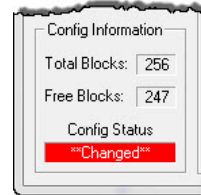


Module Modes of Operation

On the CTRIO Workbench main window, a single button toggles between Run Mode and Program Mode. The Module Mode indicator will show which mode the module is in. Make configuration changes in either Run Mode or Program Mode. To save the configuration to the module, click “Write Module”, which is only active in Program Mode.



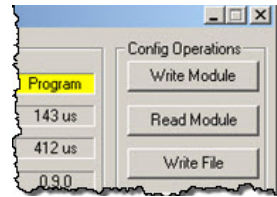
In the lower left corner of the main Workbench dialog, is the Config Status indicator. If the current configuration is different from the CTRIO(2) and different from any saved files, the indicator will display the word “Changed”. If the current configuration has been written to the module or a file, the message will read “Same as Module”, “Same as File”, or “Same as Both”.



Program Mode - Configuring the CTRIO(2) Module

After the configuration is created or changed in CTRIO Workbench, it must be written to the CTRIO(2) module. This is accomplished by returning to the main CTRIO Workbench window and clicking on “Write Module”.

If the configuration was created using Workbench Offline version, connect the PC to the CTRIO(2) module through the CPU/controller and write the configuration to the module.



NOTE: Entering program mode takes the CTRIO(2) module offline. Input pulses are not read or processed in Program mode, and all outputs are disabled. CPUs will hold last value in memory while the CTRIO(2) is in Program Mode.

Run Mode - Start Processing I/O Pulses with the CTRIO(2) Module

Selecting Run Mode causes the CTRIO(2) module to begin processing pulses based on the I/O configuration created.

In Run mode the CTRIO Workbench utility also allows monitoring to verify the proper operation of inputs and outputs. Using Monitor I/O dialog, the count change, reset, etc. are displayed. Monitor I/O is very useful for debugging and commissioning a new system. See Chapter 8 “CTRIO Workbench, Monitor I/O” for more information.

The CTRIO mode follows the CPU mode. If the CPU is placed in Run Mode, the CTRIO(2) module will also enter Run Mode (see note below). If the CPU is placed in STOP or PROGRAM Mode, the CTRIO(2) will enter Program Mode. The CTRIO(2) also responds to mode changes made in Workbench and can be placed in Run Mode while the CPU is in Stop or Program Mode. The CTRIO(2) module responds to the most recent change whether performed in Workbench or from the CPU.

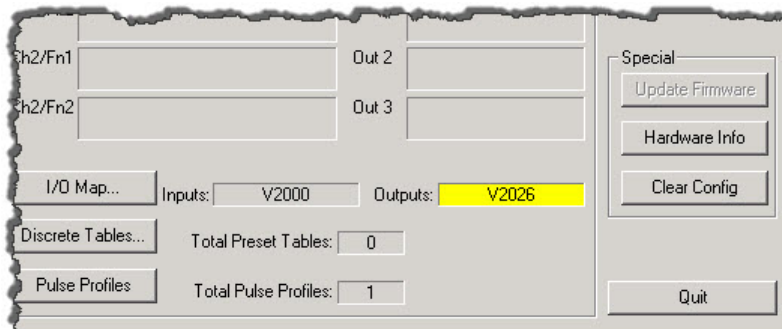


NOTE: The CTRIO(2) module will not enter Run Mode if it does not have a valid configuration.

Suspend (Output Reads) – In Monitor I/O, Work Independently from the Controller

Another mode of operation is used in Monitor I/O. Suspended has the CTRIO(2) executing its code, but ignoring commands initiated in ladder logic. Instead, commands are taken only from Monitor I/O. Using Suspend prevents interference from the CPU while manually controlling the CTRIO module. CTRIO Workbench offers to ‘Suspend output reads’ when Monitor I/O is opened.

Upon closing Monitor I/O, CTRIO Workbench offers to ‘Enable output reads’. If left Suspended, the Outputs address field will be highlighted yellow, as seen below.



CTRIO WORKBENCH, CONFIGURING INPUTS



In This Chapter...

Configure I/O Dialog Overview	5-2
Input Function Selections.....	5-3
Counter Function.....	5-4
Pulse Catch	5-6
Edge Timer.....	5-7
Dual Edge Timer	5-8
Reset FN1 and Reset FN2 (Hard Resets for Counters Only).....	5-10
Soft Resets	5-10
Capture FN1.....	5-11
Inhibit FN1	5-11
Limit Out.....	5-11
Introduction to the Scaling Wizard	5-12

Configure I/O Dialog Overview

- DL
- Win
- NI

The **Configure I/O** dialog box is where input and output functions are assigned to the module. The choice of input and output function determine which options are available.

The input and output function boxes allow selections for supported functions. The configuration software disallows any unsupported selections.

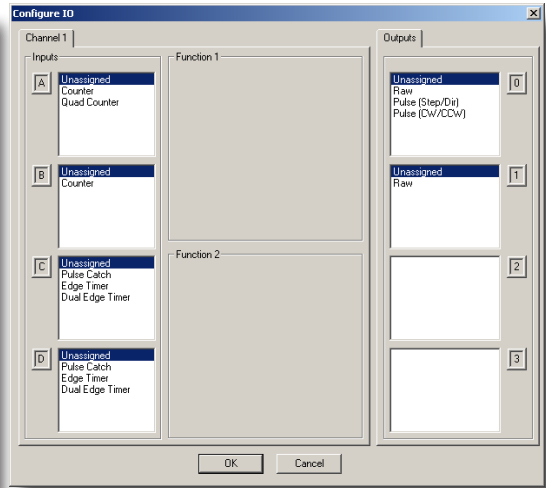
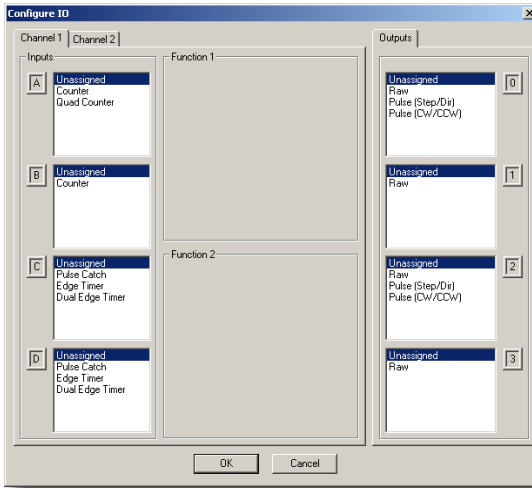
For *DirectLOGIC* users, click on the “**Config IO...**” button to arrive at a dialog box shown below. Notice that the window has a tab for each input Channel. The H0-CTRIO(2) only has one input channel.



NOTE: The *Configure IO* dialog box can be accessed without being in *PROGRAM* mode, however *PROGRAM* mode must be selected to write the configuration to the CTRIO(2) module.

H2, H4, T1H-CTRIO, H2-CTRIO2
Configure I/O Dialog

H0-CTRIO(2)
Configure I/O Dialog



The input options are listed by function. Four boxes labeled A, B, C, and D correspond to the input terminals on the face of the module (1A-1D or 2A-2D; A-D for the H0-CTRIO(2)). The Output functions are listed as 0, 1, 2, and 3. The numbers correspond to the markings beside the module’s output terminals (Y0-Y3; Y0-Y1 for the H0-CTRIO(2)).

- DL
- Win
- NI

Be sure to write the changes to the module from Workbench when the configuration is complete. For example, click on “Counter” in the “A” box, then OK to return to the main Workbench window. Then at the main window, click “**Write Module**” to save the configuration to the module, (The module will need to be in Program Mode to perform the **Write Module** operation). Without performing the **Write Module** operation (or a **Write File** operation) any configuration changes will be lost upon quitting Workbench. This applies to all changes to the module configuration.

Input Function Selections

The module supports five primary input functions:

- **Counter:** Supports up or down counting using single-ended encoders (or (other single-ended pulse sources) as inputs.
- **Quadrature Counter:** Supports quadrature counting using quadrature encoders as inputs.
- **Pulse Catch:** Allows a very short duration pulse to be qualified and lengthened to a time period long enough to guarantee that it is seen by the CPU scan.
- **Edge Timer:** Measures the time between edges on an input. This function is available on inputs C and D.
- **Dual Edge Timer:** Designed to measure from one incoming signal pulse edge to another incoming signal pulse edge. The user selects whether to measure between rising edges, falling edges, etc.



Making Selections

Each of the primary functions uses one or two input terminals for making connections to field devices (plus a common). Combinations of the listed functions are possible. The **Config I/O** dialog box disallows any unsupported configurations.

As a selection is made at Input box A, the assigned function is displayed under the Function 1 area. For Input box B the selected function is displayed under the Function 2 area (Selections in box B will be unavailable when Quad Counter is selected in box A). For input function boxes C and D, as selections are made, these are displayed as modifiers to their respective functions (A or B).

As an example, click on **Counter** from box A, the relevant parameters are displayed under Function 1 header (Up or Down Counter, Count In, Reset Value and a Ruler icon). Choosing a “modifier” from box C or D will add another parameter to the respective function area.



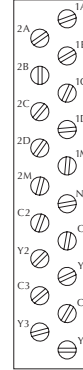
NOTE: Notice that the selection choices in the **Outputs** panel change based on selections made in the **Inputs** panel, congruently as selections are made in **Outputs** panel, available **Inputs** selections are changed.

The following pages describe the function selections available in the Config I/O dialog box.

Counter Function

- ✓ DL
- ✓ Win
- ✓ NI

The CTRIO(2) module supports up or down counting using single-ended encoders (or other single-ended pulse sources) as inputs. Encoders, proximity sensors, etc., can be connected to Input A and/or Input B on either channel or both channels. The C and D Inputs are available to modify the A and B Inputs such as Reset, Inhibit, or Capture. These functions are explained later in this chapter. The CTRIO(2) discrete output(s) can be assigned to the Counter function using the Preset Tables dialog box. Refer to Chapter 6, *Creating and Using the Output Preset Tables* (6-8), for details.

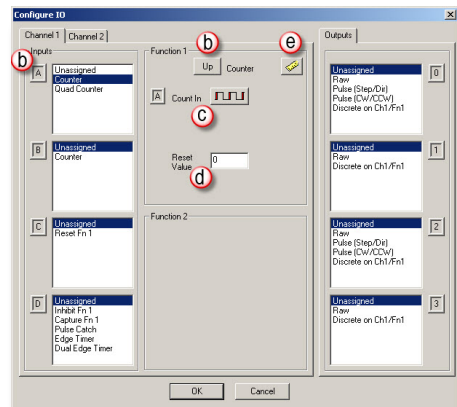


NOTE: To insure proper operation, the field device wiring and the configuration must be compatible. For wiring information see Chapter 3 “Installation and Field Wiring”.

The module’s four input terminals are represented by A, B, C, and D boxes on the left side of this dialog box. If wiring the counter input to terminal 1A, select the Channel 1 tab near the top of this window and click “Counter” (a) in box A.

At this point, four decisions are required to configure the input at 1A.

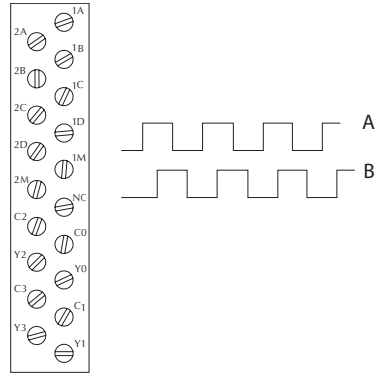
1. Select count Up or count Down in **Function 1** panel. Click the button labeled **Up** (or **Down**) (b) to view the opposite count value.
2. Each input pulse is counted. Designate whether the count is to register on the rising edge of the pulse, the falling edge, or both. The button with the graphical representation of a pulse (c) toggles between these options.
3. The Reset value is assigned by clicking in the data input field (d) and typing a value. This value is for hardwired resets. When the hardwired reset is activated, the count value returns to the reset value.
4. The last remaining decision to be made is about scaling. Clicking on the button (e) with the ruler symbol opens the Scaling Wizard. The Scaling Wizard is intelligent in that it offers scaling options that are appropriate for the input selections. The scaling wizard will be discussed later in this chapter.



Quadrature Counter

- DL
- Win
- NI

The CTRIO(2) module supports quadrature counting using quadrature encoders as inputs. Connect the encoder to input A and input B on either channel. A second quadrature encoder can be connected to the other channel. The C and D inputs are available to control the quadrature input counting. The C and D inputs can be used for Reset, Inhibit, or Capture. These functions are explained later in this chapter. The CTRIO(2) discrete output(s) can be assigned to the Quad Counter function using the Preset Tables dialog box. Refer to Chapter 6 *Creating and Using the Output Preset Tables* (6-8), for details.

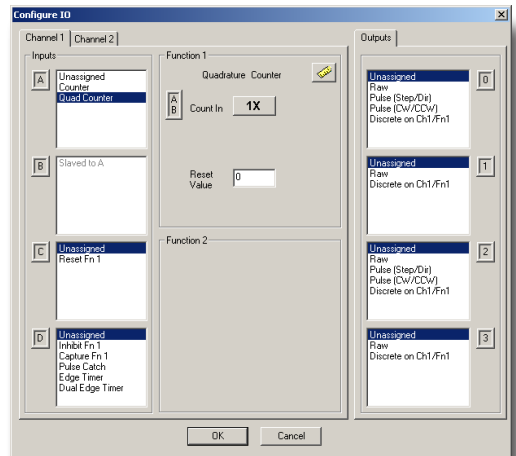


NOTE: To insure proper operation, the field device wiring and the configuration must be compatible. For wiring information see Chapter 3 “Installation and Field Wiring” topic.

Notice the A, B, C, and D boxes down the left side of this dialog box panel represent the module’s four input terminals. If wiring quadrature counter inputs to terminal 1A and 1B, select **Channel 1** tab near the top of this window and click “Quad Counter” in box A. Notice that input B options are now grayed out and slaved to input A.

At this point, there are three decisions to make regarding quadrature input:

1. A multiplier can be applied to the quadrature input to increase its resolution. Select “1X”, “2X”, or “4X.” (1X = pulses processed on leading edge of input A, 2X = pulses are processed on both edges of input A, 4X = pulses processed on both edges of input A and both edges of input B).
2. The “Reset Value” is assigned by clicking in the data input field and typing in a value. When the count is reset, using any of the reset methods, the count value returns to the Reset Value. The reset options are described in more detail later in this chapter.
3. The last remaining decision to be made is about scaling. Clicking the button with the ruler symbol starts the Scaling Wizard. The Scaling Wizard is intelligent in that it offers only those scaling options that are appropriate for the input selections. The scaling wizard is discussed in greater detail later in this chapter.



Pulse Catch

- ✓ DL
- ✓ Win
- ✓ NI

The Pulse Catch function allows a very short duration pulse to be qualified and lengthened to a time period long enough to guarantee that it is seen by the CPU scan. CPU scans necessarily vary with the length and complexity of the user's program. A scan frequency of several milliseconds is possible. On a conventional discrete input module, if a pulse is shorter than one scan time, there is a chance the CPU will not see the pulse.

The CTRIO(2) module's Pulse Catch function sees the fast incoming signal and holds its status in a status bit until the CPU can see it. A discrete output(s) can also be tied to follow the Pulse Catch input.

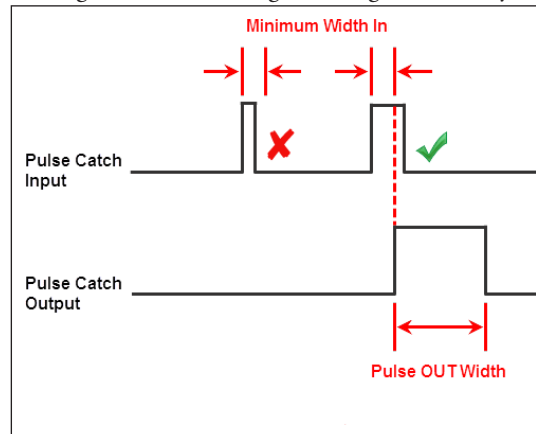
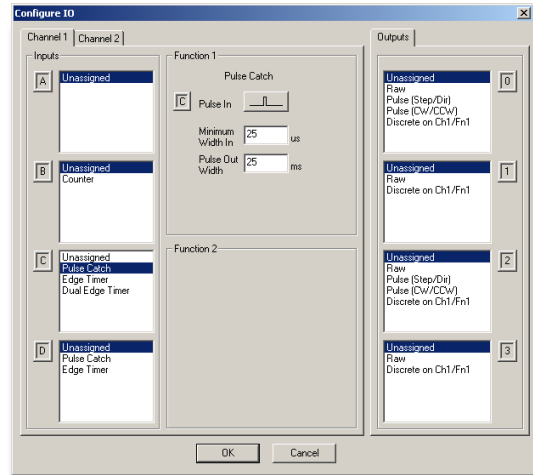


NOTE: To insure proper operation, the field device wiring and the configuration must be compatible. For wiring information see Chapter 3 "Installation and Field Wiring".

Notice that the module's four input terminals are represented by the A, B, C, and D boxes along the left side of this dialog box. If wiring an input to terminal 1C, select the **Channel 1** tab near the top of this window and click Pulse Catch in box C.

Three selections must be made in conjunction with the Pulse Catch option.

1. First, a decision must be made whether to look for the rising edge of the pulse or the falling edge of the pulse. This is a critical decision. Careful attention should be paid to the type of output the field device generates. If the signal voltage is normally low, but a short duration pulse sends the signal to the ON state, use rising edge as trigger OFF, and falling edge as trigger ON for the reverse situation.
2. The second decision is the minimum pulse width to capture. Transients below this width will not be recorded. Set this value by typing the desired value in the "Minimum Width In" field.
3. The final decision to be made is the length of pulse the CTRIO(2) module should send in response to the input pulse. Make this setting by typing in the desired value in the "Pulse Out Width" field.



Edge Timer

- DL
- Win
- NI



The Edge Timer measures the time between edges on an input. This function is available on inputs C and D. The CTRIO(2) discrete output(s) can be assigned to the timer function using the Preset Tables dialog box. Refer to Chapter 6 *Creating and Using the Output Preset Tables* (6-8) for details.

A scaling option is available with the Edge Timer to convert the time measured to a calculated rate, like flow or velocity.

NOTE: To insure proper operation, the field device wiring and the configuration must be compatible. For wiring information see Chapter 3 “Installation and Field Wiring”.

Notice that the module’s four input terminals are represented by the A, B, C, and D boxes down the left side of this dialog box. When wiring an input to terminal 1C, select the Channel 1 tab near the top of this window and click Edge Timer in box C.

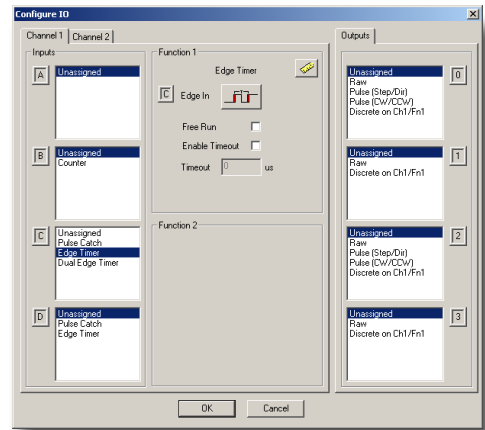
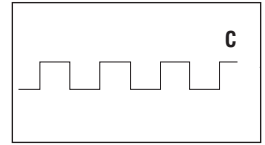
At this point, four decisions must be made regarding the input at 1C.

1. First, designate which edges to use.
There are four options:

- Rising edge to rising edge
- Falling edge to falling edge
- Rising edge to falling edge
- Falling edge to rising edge







2. The “Free Run” option is assigned by clicking in the appropriate box. If the application calls for velocity measurements to be taken at the commencement of some event, do not use Free Run. If the application calls for velocity measurement on a continuous (moving average) basis, choose Free Run.
3. The “Enable Timeout” option is assigned by clicking in the appropriate box and specifying a Timeout period. Once the timer is enabled, the Timeout Bit is set if the time that it takes the CTRIO(2) to see the configured input edge exceeds the specified Timeout Period. Also, if the time before the CTRIO(2) sees the next configured edge exceeds the specified Timeout Period, the Timeout bit is set. More information about the Timeout function can be found in Chapter 6.
4. The last remaining decision to be made is about scaling. Clicking the button with the tape measure symbol starts the Scaling Wizard. The Scaling Wizard is intelligent in that it offers scaling options that are appropriate for the chosen input selections. This will be discussed later in this chapter.



Dual Edge Timer

- DL
- Win
- NI

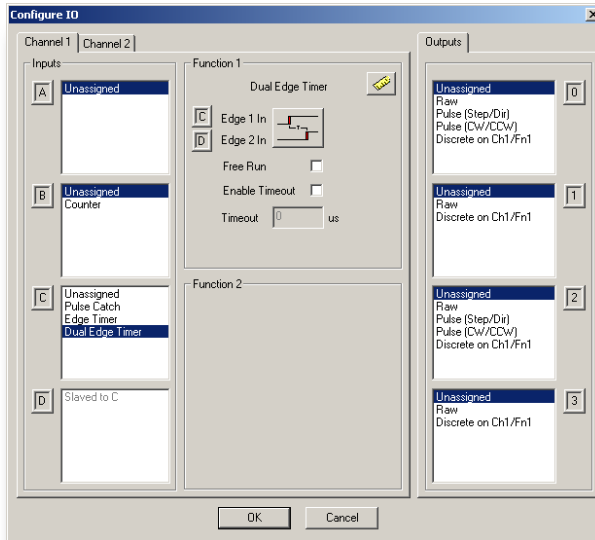
The Dual Edge Timer is designed to measure from a pulse edge on one incoming signal to a pulse edge on another incoming signal. The user selects whether to measure between rising edges, falling edges, etc. The choices are summarized in the tables below. The CTRIO(2) discrete output(s) can be assigned to the Dual Edge Timer function using the Preset Tables dialog box. Refer to *Creating and Using the Output Preset Tables* section (6-8), in Chapter 6 for details. A scaling option is available with the Dual Edge Timer to convert the time measured to a calculated rate, like flow or velocity.

Dual Edge Timer at Function 1		Dual Edge Timer at Function 2
Rising edge of C to rising edge of D		Rising edge of D to rising edge of C
Rising edge of C to falling edge of D		Rising edge of D to falling edge of C
Falling edge of C to rising edge of D		Falling edge of D to rising edge of C
Falling edge of C to falling edge of D		Falling edge of D to falling edge of C



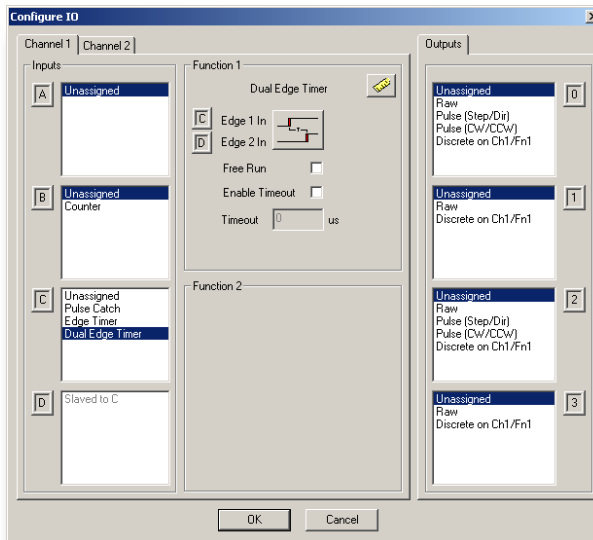
NOTE: To insure proper operation, the field device wiring and the configuration must be compatible. For wiring information see Chapter 3 “Installation and Field Wiring”.

Notice the four boxes, A, B, C, and D, along the left side of this dialog panel represent input terminals on the module. If wiring inputs to terminals 1C and 1D, select the **Channel 1** tab near the top of this window and click “Dual Edge Timer” in box C or D.



At this point, four decisions must be made regarding the input at 1C or 1D.

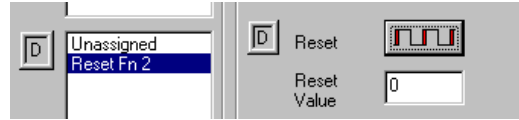
1. First, designate the pulse edges to measure between.
2. If the application calls for velocity measurement on a continuous basis, choose Free Run. The “Free Run” option is assigned by clicking in the appropriate box. If the application calls for velocity measurements to be taken at the commencement of some event, do not use Free Run.
3. The “Enable Timeout” option is assigned by clicking in the appropriate box and specifying a Timeout period. Once the timer is enabled, the Timeout Bit is set if the time that it takes the CTRIO(2) to see the configured input edge exceeds the specified Timeout Period. Also, if the time elapsed before the CTRIO(2) sees the next configured edge exceeds the specified Timeout Period, the Timeout bit is set. More information about the Timeout function can be found in chapter 6.
4. The last remaining decision to be made is about scaling. Clicking the tape measure icon with opens the Scaling Wizard. The Scaling Wizard is intelligent in that it offers scaling options that are appropriate for selected input selections. The scaling wizard is discussed later in this chapter.



Reset FN1 and Reset FN2 (Hard Resets for Counters Only)

- DL
- Win
- NI

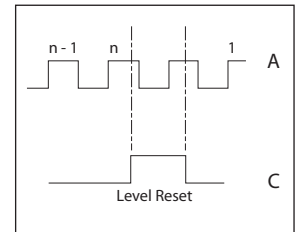
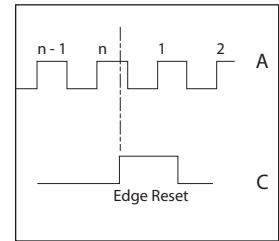
Reset FN1 is available only if a Counter or Quad Counter is selected as the primary function. For example, if either counter function is chosen on terminal 1A, (single-ended or quadrature) terminal 1C may be selected for a hard reset signal. Other options are available on terminal 1D such as Capture and Inhibit (see next page).



Reset FN2 is available if terminal 1B is chosen for a counter input. Reset FN2 will reset the counter connected to terminal 1B.

Two distinct types of hard resets are available. One is an edge reset. The other is a level reset. The **Edge Reset** sets the current count to zero on the specified edge (rising or falling) of the reset pulse (see upper example). The **Level Reset** resets the count to zero (as long as the reset pulse is held high (or low) depending on configuration. When the reset pulse disappears, the count resumes (see lower example).

If the Reset options are not available in the **Configure I/O** dialog box, then the selected input functions do not use the reset modifier.



NOTE: Reset FN1 and Reset FN2 represent hard-wired inputs to terminal C or D. An appropriate field device must be connected to the designated terminal to perform the reset function.

Soft Resets

Soft resets are available by turning on the appropriate control bit in the control program (Counters only) or by using the **Reset Count** function within a Discrete Output Preset Table configuration (Counters/Timers). Counter control bit resets are always level resets, meaning they hold the count at zero until the reset bit is turned off.

Capture FN1

- DL
- Win
- NI

Capture FN1 is available only if a Counter or Quad Counter is selected as the primary function. For example, if either counter function on terminal 1A is selected, then the option of using terminal 1D for a capture signal is presented.

Capture FN1 “snapshots” the current count into the 2nd DWord register (Parameter 2). The Capture feature is available with a single-ended Counter on input A or a Quad Counter on inputs A and B.



NOTE: Capture FN1 represents a hard-wired input to terminal D. An appropriate field device must be connected to the designated terminal to perform the capture function.

Inhibit FN1

Inhibit FN1 is available only if a Counter or Quad Counter is selected as the primary function. For example, if either counter function is chosen on terminal 1A, then an option of using terminal 1D for an inhibit signal is presented.

The Inhibit FN1 signal prevents the CTRIO(2) from counting pulses. The Inhibit feature is available with the “A” Counter or Quad Counter on each channel.



NOTE: Inhibit FN1 represents a hard-wired input to terminal D. An appropriate field device must be connected to the designated terminal to perform the inhibit function.

Limit Out

Limit Out is used for monitoring a discrete input for position feedback while homing an output. If using a CTRIO2 module, **Trapezoid with Limits** will nearly always be a better profile option than using a Home Search profile to find the home position. Refer to **Home an Output** earlier in this manual for a comparison of the available options.

Limit Out on terminals 1C and 1D is only available if an output is configured as **Pulse**.

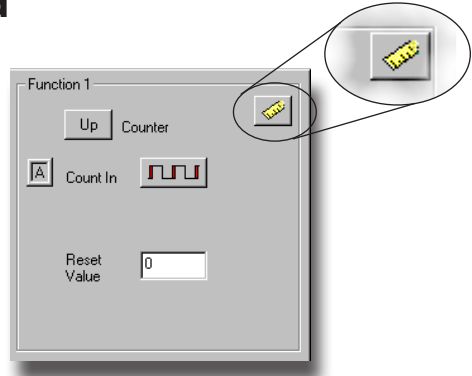
Introduction to the Scaling Wizard

- DL
- Win
- NI

Scaling raw signals to engineering units is accomplished using the Scaling Wizard. Start the Scaling Wizard by clicking the ruler button on the Configure IO dialog box. This button appears only after one of the Counter or Timer functions is selected.

The Scaling Wizard options are different for the Counter functions as compared with the Timer functions. “Position” and “Rate” scaling are available when a Counter function is selected. “Interval” scaling is available when selecting a Timing function.

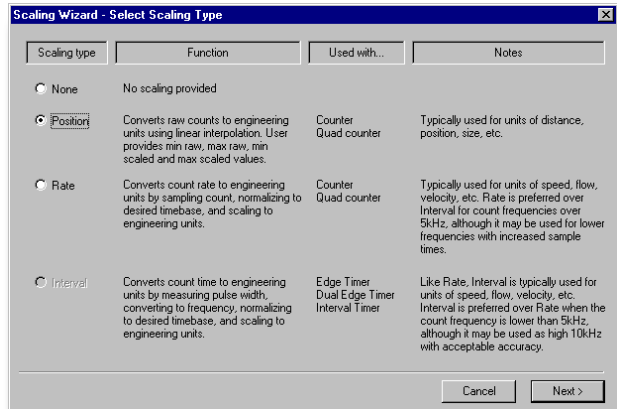
The following text steps through the dialogs used for each scaling type. Substitute appropriate values to set up scaling for the application.



Scaling Wizard Examples for Counter Functions

On the Scaling Wizard select the appropriate scaling type:

- None, Position, or Rate. No scaling is accomplished if the None button is selected.
- Position scaling is appropriate for measuring distance, position, or size.
- Rate scaling is appropriate for velocity, RPM, flow, or similar rate based measurements.



Read the Notes and other information on this panel before leaving this window.



NOTE: If the input pulse rate is always slower than 5kHz, consider using Interval scaling on a Timer input instead of Rate scaling on a Counter input.

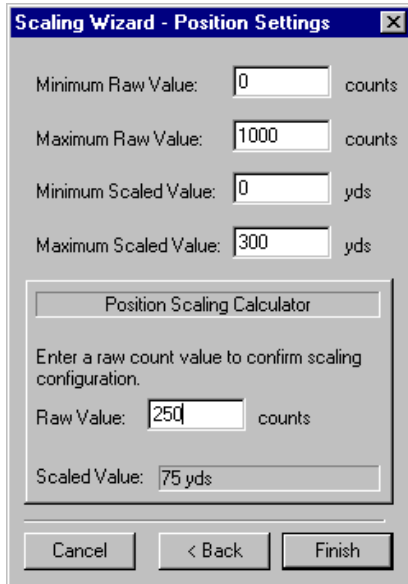
Position Scaling (Counter)

To select Position Scaling, click the radio button beside the word **Position**. Click **Next** to move to the Output Settings dialog box.

Notice on the Output Settings dialog box the field for engineering units. Enter an appropriate value for Position Scaling, for example yards, feet, meters, cubic inches, etc. Seven data types are available including BCD (to make values more easily used by *DirectLOGIC* PLCs).

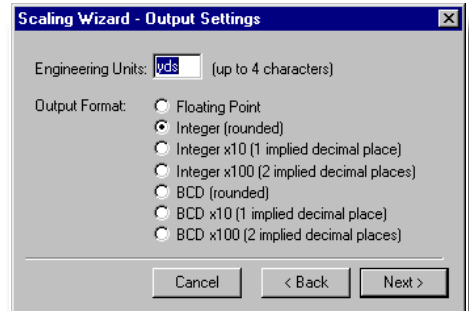
Click **Next**, to open the **Scaling Wizard Position Settings** dialog box. It is here that the span of raw counts that equates to a span of engineering units is entered.

This window contains a calculator to double check the meaning of selected Position Settings. Enter a value into the **Raw Value** field to see the equivalent value in engineering units.



The **Scaling Wizard - Position Settings** dialog box contains the following fields and controls:

- Minimum Raw Value: counts
- Maximum Raw Value: counts
- Minimum Scaled Value: yds
- Maximum Scaled Value: yds
- Position Scaling Calculator** section:
 - Enter a raw count value to confirm scaling configuration.
 - Raw Value: counts
 - Scaled Value: yds
- Buttons: **Cancel**, **< Back**, **Finish**



The **Scaling Wizard - Output Settings** dialog box contains the following fields and controls:

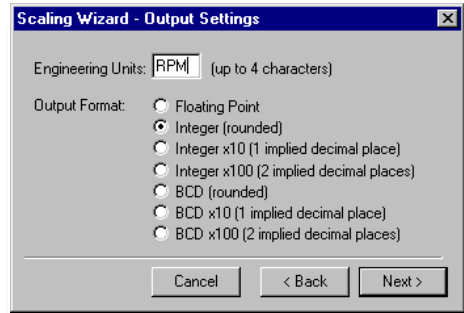
- Engineering Units: (up to 4 characters)
- Output Format:
 - Floating Point
 - Integer (rounded)
 - Integer x10 (1 implied decimal place)
 - Integer x100 (2 implied decimal places)
 - BCD (rounded)
 - BCD x10 (1 implied decimal place)
 - BCD x100 (2 implied decimal places)
- Buttons: **Cancel**, **< Back**, **Next >**

Rate Scaling (Counter)

To select Rate Scaling, click the radio button beside the word Rate. Click **Next** to move to the **Scaling Wizard - Output Settings** dialog box.



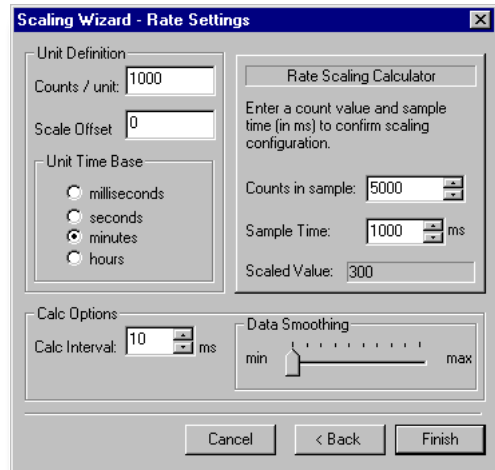
On the Output Settings dialog panel, notice the field for engineering units. Enter an appropriate value for Rate Scaling, RPM, fps, flow, etc. Seven data types are available including BCD (to make values more easily used by *Direct*LOGIC PLCs).



Click **Next**, to open the **Scaling Wizard - Rate Settings** dialog box. On this panel enter the counts per unit of time and the time base. A scale offset is also provided to adjust the result by a constant amount.

This window contains a calculator to double check the Rate Settings. Enter a value into the Raw Value field to see the equivalent value in engineering units.

As an example, with a 1000 pulse/revolution encoder used to measure RPM (of the encoder shaft), enter 1000 for the **Counts/unit** and “minutes” as the Time Base. A check using the calculator (over a sample time of 1000 ms = 1 second) reveals that 5000 counts equals 300RPM.



5000 counts/1000 counts per rev = 5 revolutions;

5 revolutions/1 sec x 60 sec= 300 RPM

Data Smoothing allows rolling averages to be taken to calculate a value.

Min = 1 sample, max = 25 samples in the rolling average.

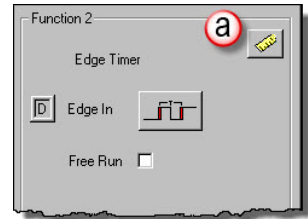
Calc Interval – This tells the CTRIO how often to calculate the rate. Choose a Calc Interval larger than the slowest rate at which pulses will usually arrive.



NOTE: If the input pulse rate is always slower than 5kHz, consider using Interval scaling on a Timer input instead of Rate scaling on a Counter input.

Using the Scaling Wizard with Timer Functions

Scaling raw signals to engineering units is accomplished using the Scaling Wizard. Start the Scaling Wizard by clicking the “ruler button” (a) on the Configure IO dialog panel. This button appears only after selecting one of the Timer functions.

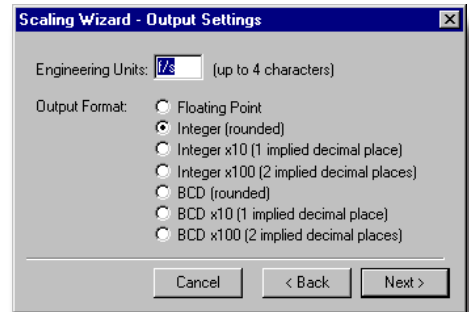


Interval Scaling (Timer)

To select Interval Scaling, click the radio button beside the word Interval. Now, click Next to move to the Scaling Wizard - Output Settings dialog box.



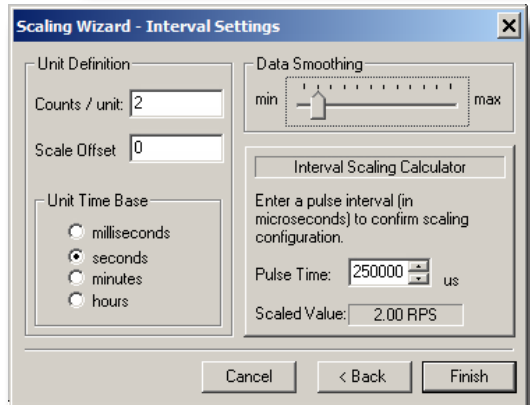
On the Output Settings dialog box notice the field for engineering units. Enter an appropriate value for Interval Scaling, RPM, fps, flow, etc. Seven data types are available including BCD (to make values more easily used by *DirectLOGIC* PLCs). Click Next, to open the Scaling Wizard Interval Settings dialog box.



In the counts/unit field, enter how many pulses should pass in one unit of time (in the time base of the engineering units) for the engineering units to equal 1 unit of measure. Then, select the unit time base (of the engineering units).

For example, use interval scaling to determine the revolutions per second (RPS) of a rotating object with pickups every 180°. At 1 RPS, two pulses will pass in 1s (seconds is time base of engineering units), so enter ‘2’ in counts/ unit and select seconds for Unit Time Base.

To verify, enter the time between pulses in the calculator. If the device were rotating at 2 RPS, a pulse would occur every 250,000µs.



Data Smoothing - This is a slider adjustment that has 10 positions from min to max. Moving the slider toward max increases the number of samples that are averaged together. This provides a smoothing affect in the rate scale value, i.e., so that it doesn’t jump around. The positions from min to max represent 1, 2, 3, 5, 7, 10, 13, 17, 21 and 25 samples.

CTRIO WORKBENCH

CONFIGURING OUTPUTS



In This Chapter...

Configure I/O Dialog Overview	6-2
Output Function Selections	6-3
Raw Output.....	6-4
Discrete Outputs.....	6-5
Pulse Outputs.....	6-11

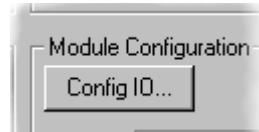
Configure I/O Dialog Overview

- DL**
- Win**
- NI**

The **Configure I/O** dialog box (pictured below) is where input and output functions are assigned to the module. The choice of input and output functions determines which options are available.

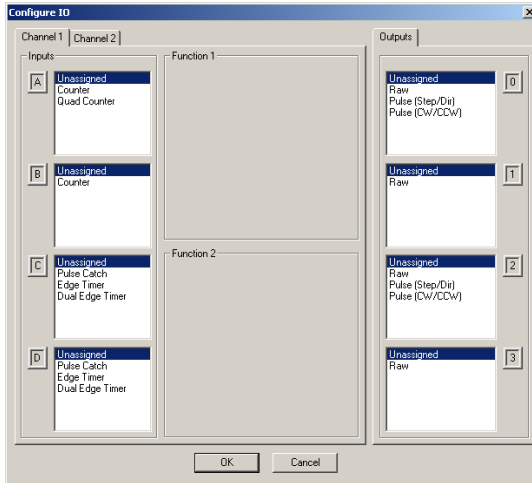
The input and output function boxes prompt you with selections for supported functions. The configuration software allows only supported selections.

For **DirectLOGIC** users, click on the “**Config IO...**” button to arrive at a dialog box shown below. Notice that the window has a tab for each input Channel. Remember that the **H0-CTRIO(2)** only has one input channel (left graphic).

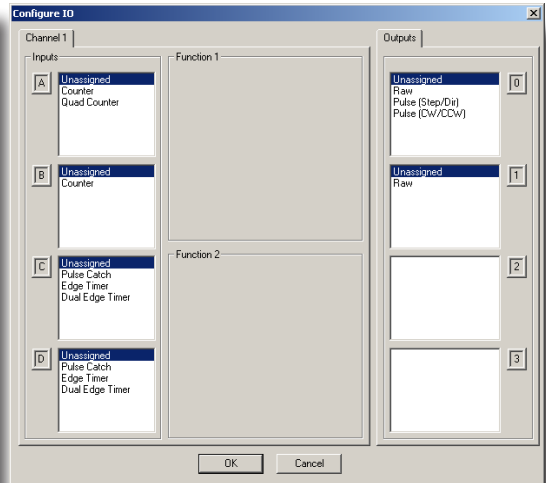


NOTE: You do not have to be in **PROGRAM** mode to enter the **Configure IO** dialog box, however you must be in **PROGRAM** mode to write the configuration to the **CTRIO(2)** module.

H2, H4, T1H-CTRIO, H2-CTRIO2
Configure I/O Dialog



H0-CTRIO(2)
Configure I/O Dialog



The input options are listed by function. Four boxes labeled A, B, C, and D correspond to the input terminals on the face of the module (1A-1D or 2A-2D; A-D for the **H0-CTRIO(2)**). The Output functions are listed as 0, 1, 2, and 3. These numbers correspond to the markings on the module output terminals [(Y0-Y3)(C0-C3); Y0-Y1 for the **H0-CTRIO(2)**].

- DL**
- Win**
- NI**

Be sure to write the changes to the module from **Workbench** when the configuration is complete. For example, you might click on **Pulse(step/dir)** in the “0” box, then **OK** to return to the main **Workbench** window. Once you arrive back at the main window, you must click **Write Module** to save your selection to the module. The module will need to be in **Program Mode** to perform the **Write Module** operation. If you do not perform the **Write Module** operation (or a **Write File** operation) any configuration changes you made will be lost upon quitting **Workbench**. This applies to all changes to the module configuration.

Output Function Selections

Supported Functions

The module supports four output functions (five if CTRIO2):

- Raw: The control program has simple ON/OFF control of the output.
- Pulse (Step/Direction): Use for motor control. The control program sends profiles or calls stored profiles that define the movements.
- Pulse (CW/CCW): Use for motor control. The control program sends profiles or calls stored profiles that define the movements.
- Discrete Ch(1,2)/Fn(1,2): CTRIO(2) has precision ON/OFF control of the output, determined by:
 - a. Preset Table: ON/OFF as events of a preset table occur. Preset table monitors a counter input. Presets occur in order only.
 - b. Programmable Limit Switch (PLS): ON/OFF depending on current region in the table. PLS monitors a counter input.
 - c. Pulse Catch: Precise ON/OFF of an output as an input pulses ON or OFF.
 - d. Level Mode: Only available on a counter input that is scaled to a rate. Similar to a PLS, but only offers one threshold. Also offers a deadband.
- (CTRIO2) Discrete on Pulse Out (0/1; 2/3) has precision ON/OFF control of the output, determined by:
 - a. Preset Table: ON/OFF as events of the preset table occur. Preset table monitors a pulse output position. Presets occur in order only.
 - b. Programmable Limit Switch (PLS): ON/OFF depending on current region in the table. PLS monitors a pulse output position.

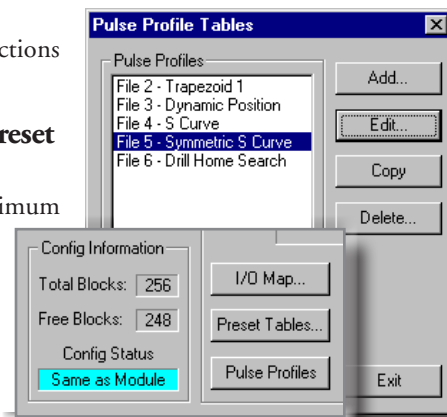


Each function uses one or two output terminals for making connections to field devices (plus a common). Combinations of the listed functions are possible. The configuration dialog allows only supported configurations.

The following pages describe the function selections available in the **Config I/O Outputs** panel.

CTRIO Memory Usage: Pulse Profiles and Preset Tables

CTRIO configuration software can create a maximum of 255 predefined Pulse Profiles. Pulse Profiles available is 255 minus the number of predefined Preset Tables. Pulse Profiles and Preset Tables are saved as File 1 through File 255. The module has 256 Total Blocks of memory allocated for Pulse Profiles and Preset Tables usage.



Raw Output

- DL
- Win
- NI



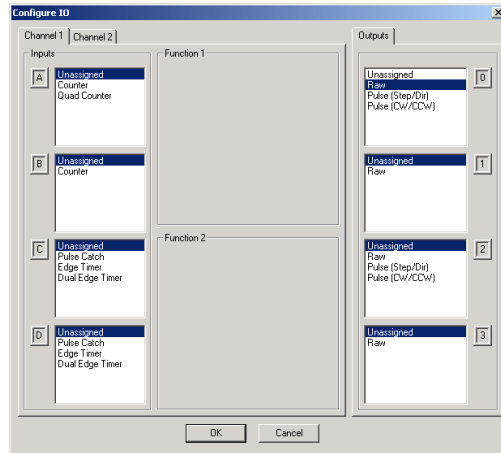
The CTRIO(2) module supports a Raw output mode that allows the CPU/controller program to have direct access to the module output points. Each output can be configured for Raw output mode and each will have a unique control bit.

NOTE: To insure proper operation, the field device wiring and the configuration must be compatible. For wiring information see Chapter 3.



Refer to (A-6) *Output Control Bit Definitions (Raw Mode)* in Appendix A for Raw output control bit addressing.

The module's output terminals are represented by the 0, 1, 2, and 3 boxes (0 and 1 for the H0-CTRIO(2)) along the right side of this dialog box.



Discrete Outputs

- DL
- Win
- NI

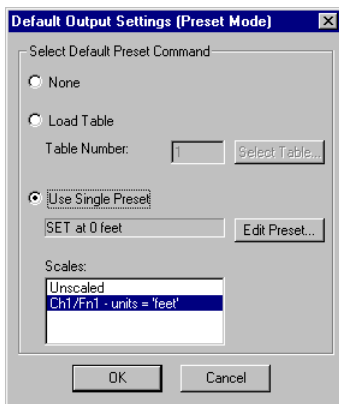
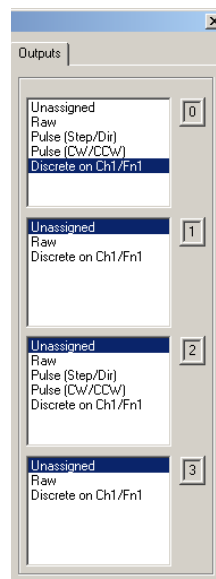
The H0-CTRIO(2) modules have two discrete outputs (Y0-Y1), all other CTRIO(2) modules have four discrete outputs numbered Y0-Y3 (C0-C3). When the Discrete option is selected, a variety of functions can be assigned to the output: Single Preset, PLS, Preset Table, Level Mode or Pulse Catch. Each function is described on the following pages.

Creating and Using a Single Preset

The outputs can respond to presets assigned by the user in the **Configure IO** dialog.

The four outputs can all be assigned to one function, or they can be grouped within functions and within channels in any manner selected by the user.

To assign single output presets, begin by selecting the Output on the **Configure IO** dialog. The outputs are identified based on terminal number. In the example to the right, output terminal “0” is designated for a discrete output, i.e. Discrete on Ch1/Fn1 counter.



Once the output selection is made, a **Preset** button appears on the **Configure IO** dialog. The button is labeled as shown.

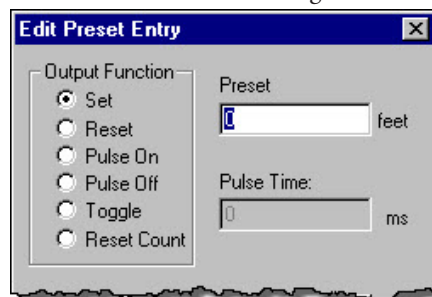


The leading numeral represents the number of the output terminal. Clicking on the **Preset** button causes the **Default Output Settings** dialog to pop up (on left). Default settings are loaded on power-up.

On the **Default Output Settings** dialog, select “**Use Single Preset**”. We will discuss Preset Tables later in this chapter. Now, click **Edit Preset** to arrive at the **Edit Preset Entry** dialog.

Six output functions are available (as shown in the figure below). Set the **Preset** value in engineering units if the signal has been scaled or in raw count if the signal has not been scaled. Refer to Chapter 5 for a discussion on scaling. **Pulse On** and **Pulse Off** require a **Pulse Time** setting. The **Pulse Time** is set in ms (1 sec = 1000 ms).

Output Function Definitions	
Set	Writes output ON (maintained)
Reset	Writes output OFF
Pulse On	Writes output ON for specified time
Pulse Off	Writes output OFF for specified time
Toggle	Changes state of output
Reset Count	Resets the count to Reset Value



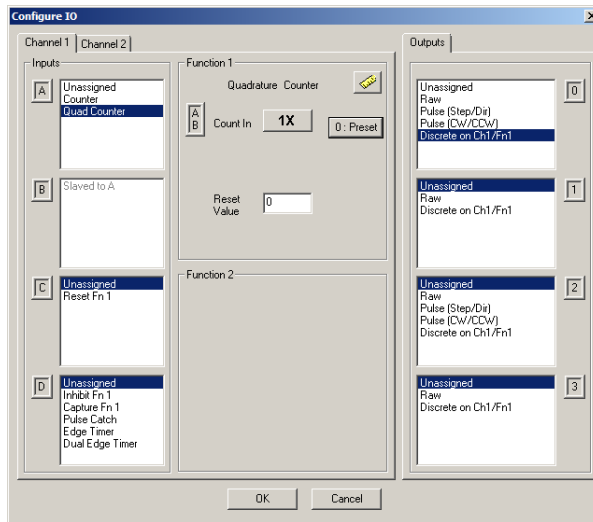
Creating and Using the Programmable Limit Switch (CTRIO2 only)

A Programmable Limit Switch (PLS) is a discrete output table used to turn an output ON or OFF at multiple points across an input's range.

- ✓ DL
- ✓ Win
- ✓ NI

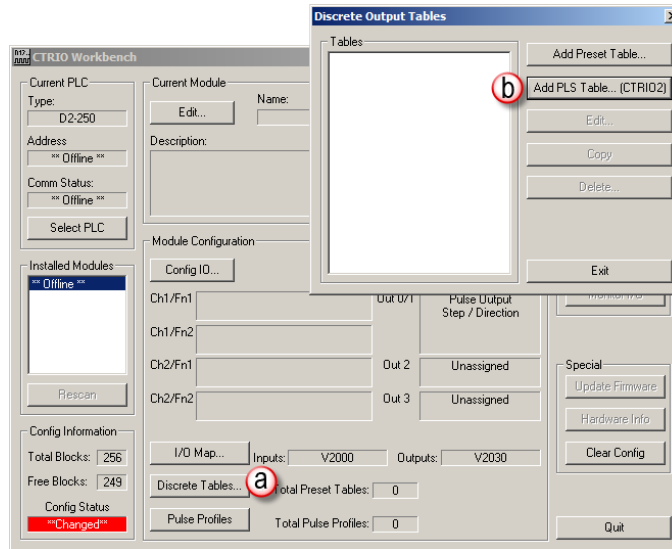
The PLS function is only available in the CTRIO2 modules. A PLS table must be created in the CTRIO2 module configuration. Once created, it can be edited using the configuration application.

To use a PLS table, an output must be configured as a discrete output paired with a counter or timer input function. For example, Channel 1 inputs A and B could be configured for quadrature counting, which would be Channel1 Function1. Output 0 could be configured as Discrete on Ch1/Fn1. When a PLS table is loaded for Output 0, the CTRIO2 module will monitor the input value for Ch1/Fn1 (Channel1 quad counter) and Output 0 will be the output turned ON and OFF by the table.

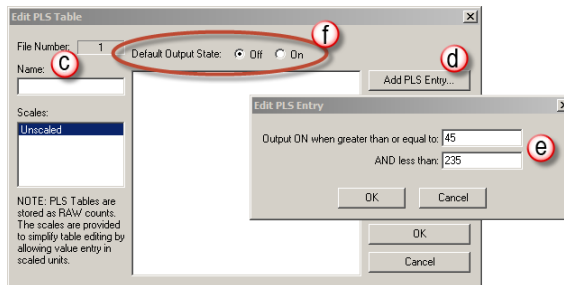


NOTE: Since a CTRIO2 Output's function is fixed in the configuration, an output cannot be changed programmatically to reference a different input function. This specific change requires a configuration change and project transfer.

To create a PLS table, click the “Discrete Tables...” button (a) then click the “Add PLS Table... (CTRIO2)” button (b).



Enter a table name in the text box (c), then click OK. Use the buttons on the right to build and modify a PLS table (d). If an input channel has scaling, it will show in the Scales list, and entries can optionally be entered in scaled units after selecting the appropriate entry in the Scales list (e).



The table entries are very simple. Select a default value for the discrete output; the output will be in this state if the input is not within a defined range (f). Add an entry for each range of input values where the output should be in the opposite state.



NOTE: Unlike a preset table, events in a PLS table can occur in any order, even simultaneously.

Creating and Using the Output Preset Tables

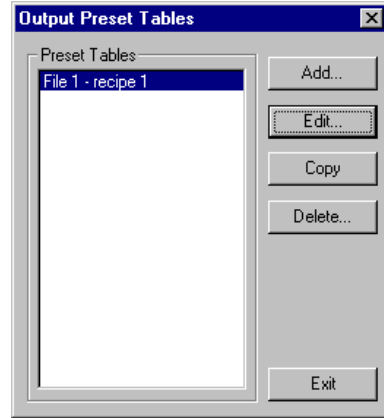
Preset tables can be used on an input configured for a:

- Timer, not scaled
- Quad/counter, not scaled
- Counter, scaled to position.

Click the “Preset Tables...” button on the main Workbench dialog. This will open the **Output Preset Tables** dialog. To create a new table, click the “Add...” (or “Edit...”)

Preset Tables...

button. This will open the Edit Preset Table dialog. To create a new table, click the “Add...” (or “Edit...”)



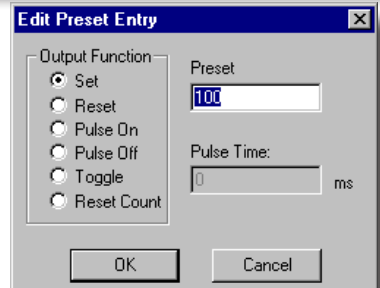
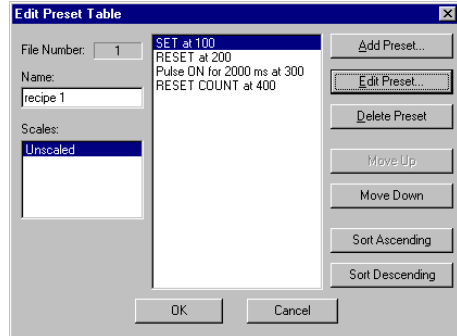
NOTE: The preset tables work similar to an event drum, not a programmable limit switch. For example, in the Edit Preset Table dialog below, the output is SET at count 100. Once the output is SET, if the count drops below 100, the output will not go OFF, it will remain SET. Once a step is complete, the focus is on the next step and that step only.



On the **Edit Preset Entry** dialog, select one of the six Output Functions. Set the preset value in engineering units if the signal has been scaled. Set the preset value in raw count if the signal has not been scaled (Scaling is discussed in Chapter 5, 5-9). Pulse On and Pulse Off require a Pulse Time setting. During the Pulse Output Time the table will be paused at the current table entry. The next table entry will not be evaluated until after the Pulse Output Time expires.

The Pulse Time is set in ms (1,000 ms = 1 sec). For the description of the Output Functions see page 6-3.

To set a particular table as the default table, use the Default Output Settings dialog described on page 6-5.



Using the Discrete Outputs in Level Mode

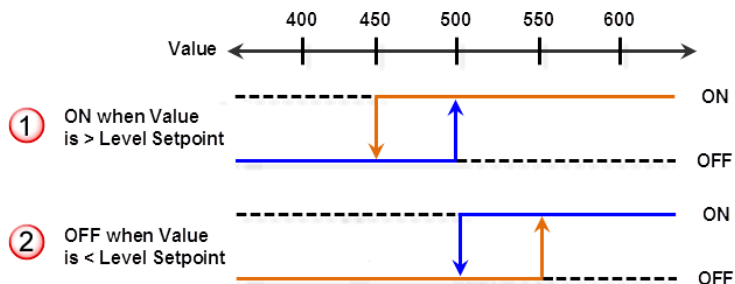
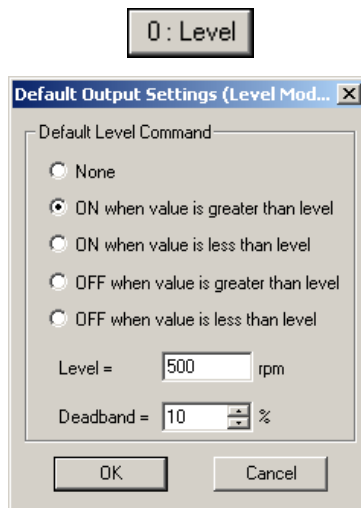
If a Counter or Timer function is scaled to produce a rate, alarm level settings can be used to trigger discrete outputs at values predetermined by the user.

Click the **Level** button on the Configure IO dialog. This will open the **Default Output Settings (Level Mode)** dialog box. The alarm level is set within this panel.

Also, a deadband percentage (in tenths of a percent) can be set to prevent the output from changing too frequently (chattering) near the Rate Level threshold.

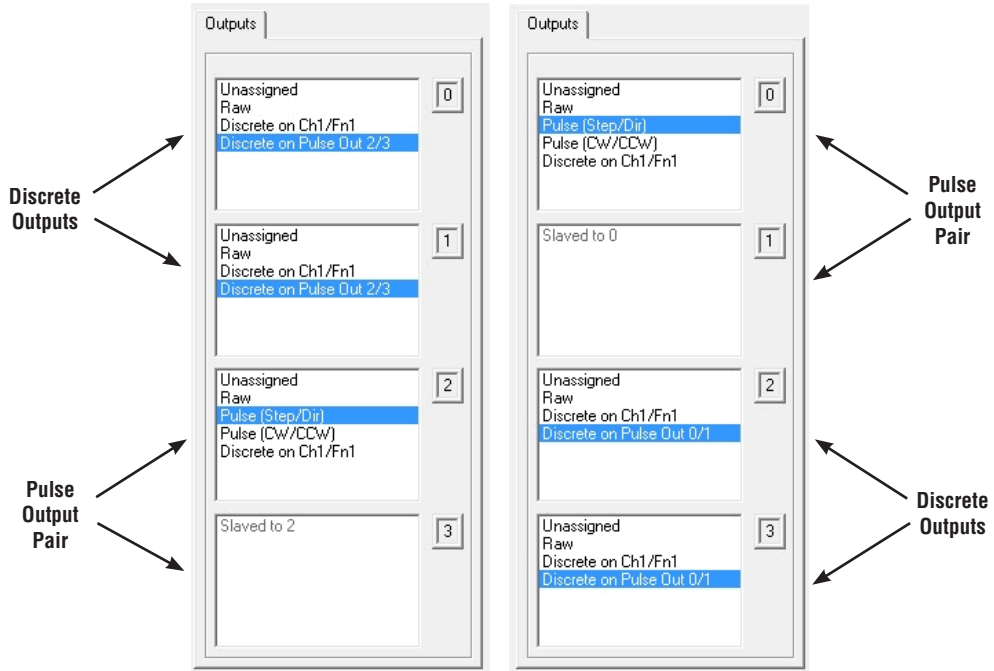
“ON when value is greater than level” example: Consider a Discrete Output set to turn ON when greater than 500 rpm and deadband is set to 10%. The output will turn ON when the level rises above 500 rpm. Due to the 10% deadband, the level must fall below 450 rpm to turn back OFF. See example 1 below.

“OFF when value is less than level” example: Consider a Discrete Output set to turn OFF when less than 500 rpm and a deadband set to 10%. When the level drops below 500 rpm, the output turns OFF. Due to the 10% deadband, the level must rise above 550 rpm for the output to turn back ON. See example 2 below.



Using the Discrete Output For Pulse Output Function

Discrete on Pulse Out “x/x” is available on any CTRIO output, but only if there is an output pair (i.e. Out0/Out1 or Out2/Out3) configured for Pulse Output. Discrete on Pulse Out “x/x” allows the output to be used in conjunction with the other pulse output pair and a discrete output table. In other words, the Discrete Output state is determined by the Pulse Output position value in a preset table. This works just like the Discrete Output On Chx/Fnx function mentioned previously with the exception that the Preset or Level buttons are not supported.



Pulse Outputs

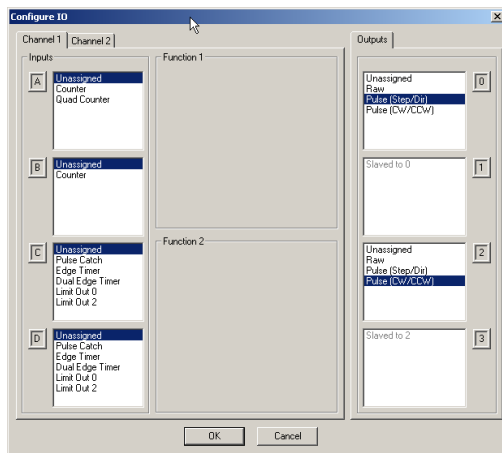
The CTRIO module offers two axes of motion control (Y0 and Y1 as an axis and/or Y2 and Y3 as an axis). The H0-CTRIO(2) has one axis of motion control (Y0 and Y1). The outputs can be configured for CW/CCW, or step and direction operation. The outputs respond to profiles defined by the user and called by the user control program. The following pulse profiles are supported:



- Trapezoid
- S-Curve
- Symmetrical S-Curve
- Dynamic Positioning
- Dynamic Velocity
- Home Search
- Free Form
- Dynamic Positioning Plus (CTRIO2)
- Trapezoid Plus (CTRIO2)
- Trapezoid w/ Limits (CTRIO2)
- Velocity Mode¹
- Run to Limit Mode¹
- Run to Position Mode¹



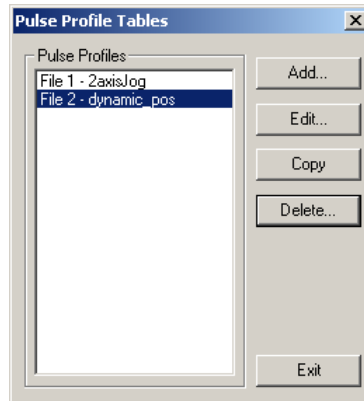
NOTE 1: There are three additional pulse profiles that are available but are not created using the Pulse Profile Tables. These profiles: Velocity Mode, Run to Limit Mode and Run to Position Mode, are discussed at the end of this chapter. For details on setting up and using the Pulse Outputs see Chapter 9 “Output Functions”.



Creating Pulse Profiles

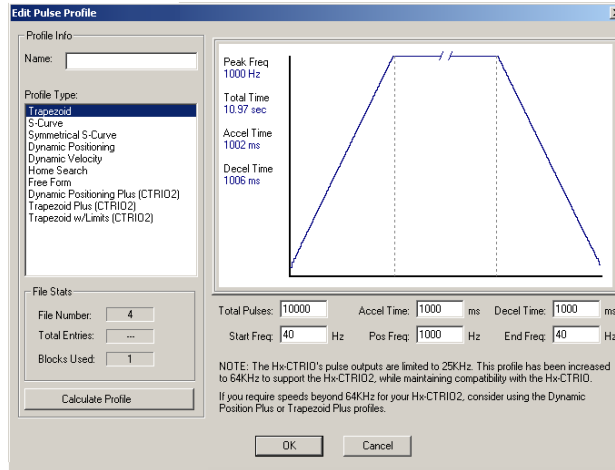
Click the **Pulse Profiles** button on the main Workbench dialog box. This will open the **Pulse Profile Tables** dialog. To create a new profile, click **Add** (or **Edit**).

Pulse Profiles



This will open the **Edit Pulse Profile** dialog.

On the **Edit Pulse Profile** dialog, select one of the ten **Profile Types**.



This dialog is used to name and define the pulse profile parameters. The various parameter fields contain typical default values however, any valid parameter entries can be entered.

CTRIO WORKBENCH, I/O MAP



In This Chapter...

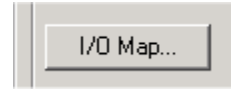
I/O Map Dialog.....	7-2
I/O Map with DirectLOGIC PLC (2 ranges mode).....	7-2
I/O Map with DirectLOGIC PLC (4 ranges mode).....	7-3
I/O Map with Direct LOGIC PLC with CTRIO(2) in ERM/EBC Network.....	7-4
I/O Map with EBC/WinPLC	7-4
I/O Map with an H2-PBC or T1H-PBC Profibus DP Controller.....	7-5
Printing a Memory Map Report	7-6
Exporting to Direct SOFT	7-6

I/O Map Dialog

- ✓ DL
- ✓ Win
- ✓ NI

The I/O Map dialog panel is used to allocate base controller memory to be used with CTRIO(2) modules. Each module will require unique memory ranges for exchanging data with the base controller. All of the *DirectLOGIC* ladder examples use the “2 ranges” mode with the Input Map starting at V2000 and the Output Map starting at V2030.

The I/O Map dialog is accessible from the main Workbench dialog. On the main Workbench dialog, click the button labeled **I/O Map**.



The I/O Map dialog shows the controller I/O memory used by the CTRIO(2) module, divided into three groups: Input Functions, Output Functions and System Functions. Just below the Map Display Mode field, you will see tabs labeled Input Functions, Output Functions and System Functions.

Click on the **Input Functions** tab or **Output Functions** tab to display the CTRIO(2) module’s assigned input or output functions (quad counter, pulse catch, pulse out, discrete out, etc.). For each input and/or output function assigned, the I/O Map dialog displays the **Input Data (CTRIO > Controller)** addresses and **Output Data (Controller > CTRIO)** addresses based on the Map Display Mode and the starting I/O addresses specified (See Chapter 10 for the memory map addresses and offset address tables).

Click on the **System Functions** tab to display the System Functions addressing. The command bits are used when reading from and writing to the CTRIO(2)’s internal registers. The other bits can be used to monitor the status of each individual I/O point on the module.

I/O Map with DirectLOGIC PLC (2 ranges mode)

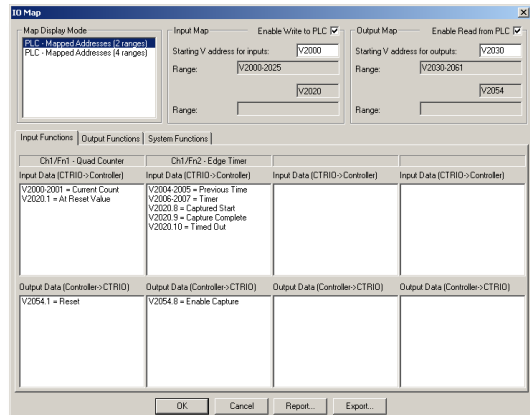


NOTE: Do not use “2 ranges” mode with a D2-240 CPU.

When using the CTRIO(2) module with a *DirectLOGIC* PLC in “2 ranges” mode, enter

- ✓ DL
- ✗ Win
- ✗ NI

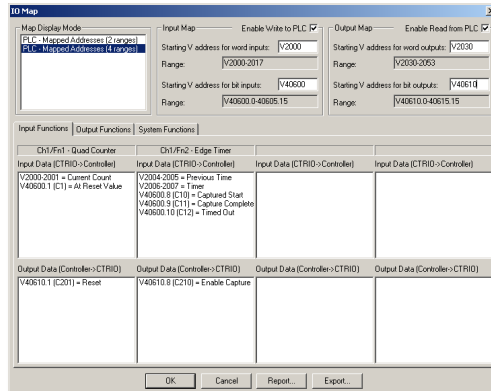
the starting V memory location for the inputs and outputs in the appropriate fields at the top of the I/O Map dialog. In the I/O Map dialog shown here, note that the Input, Output and Systems Functions addresses shown are in word and bit-of-word formats. Thus, word and bit-of-word addressing will need to be used in the ladder logic program to address the CTRIO(2)’s control and status words/bits.



I/O Map with DirectLOGIC PLC (4 ranges mode)



When using the CTRIO(2) module with a *DirectLOGIC* PLC in “4 ranges” mode, enter the starting V-memory location for the *word* inputs and outputs and the starting V-memory location for the *bit* inputs and outputs. Control relays (V40600 range) would usually be used for bit control. In the I/O Map dialog below, note that Input, Output and Systems Functions addresses shown are in word and Control Relay formats. Thus, word and Control Relay addressing will need to be used in the ladder logic program to address the CTRIO(2) control and status words/bits. Remember that the CTRIO(2) module will consume the address ranges listed in all four range fields.



I/O Map with *Direct*LOGIC PLC with CTRIO(2) in ERM/EBC Network

- DL
- Win
- NI

When using the CTRIO(2) module with a *Direct*LOGIC PLC in an ERM/EBC network, first configure the ERM network using the ERM Workbench utility. Then, from ERM Workbench, enter the CTRIO(2) module Starting input and output V-Map addresses into CTRIO Workbench I/O Map starting V-memory location for the *bit* inputs and outputs.



NOTE: If there is an 8-pt. discrete I/O module preceding the CTRIO module in the EBC base, enter the appropriate starting V-memory bit I/O address in CTRIO Workbench with a V40xxx.8 address as shown in the I/O Map example below (bottom right). This corrects the word offset created by the 8-pt. discrete I/O module. In the example below, note that V40416 Hi(8-15) is the starting ERM Workbench CTRIO input V-Map location due to the 8-pt. discrete input module preceding the CTRIO module.

The screenshot shows the ERM Workbench I/O Map window. A table lists I/O points for various modules. A callout 'See Note Above' points to the 'Slave 1/Slot 0' row. The 'I/O Map' dialog box is open, showing 'Map Display Mode' set to 'EBC - Mapped to PLC (through ERM)'. The 'Input Map' section shows 'Starting V address for bit inputs' set to 'v40416.8'. The 'Output Map' section shows 'Starting V address for bit outputs' set to 'v40515'. A note in the table indicates 'Ethernet Address[00 E0 62 00 19 AD] on IPX:'.

I/O Module	I/O Points	PLC Start	PLC End	V-map	Notes
(reserved)	Slave Status Bits	X300	X317	V40414	
	ERM Status Word	X320	X337	V40415	
	Disable Slave Command Bits	Y300	Y317	V40514	
Slave 1	H2:EBC				
Slave 1/Slot 0	8 Discrete Input	X340	X347	V40416 Lo(0-7)	
Slave 1/Slot 1	36 Discrete Input	X350	X367	V40416 Hi(8-15)/V40424 Lo(0-7)	
	96 Discrete Output	Y320	Y357	V40515/V40522	
	12 Word Output	V2100	V2113		16-bit Binary;
	8 Double Word Input	V2000	V2017		32-bit Binary;
	4 Double Word empty				
Slave 1/Slot 2	16 Doc				
Slave 1/Slot 3					

I/O Map with EBC/WinPLC

- DL
- Win
- NI

When using the CTRIO(2) module in an EBC/WinPLC system (non PLC system), the addressing will be shown as Native EBC/WinPLC addresses or if using Think&Do, the addressing can be shown as Native Think&Do addresses. Just click on the desired mode in the Map Display Mode field. The 8-pt module offset described in the note above does not apply to EBC/WinPLC or EBC/Think&Do systems.

The screenshot shows the I/O Map dialog box with 'Map Display Mode' set to 'EBC/WinPLC - Native Addresses'. The 'Input Map' section shows 'Starting V address for word inputs' set to 'v2000' and 'Starting V address for bit inputs' set to 'v40416.8'. The 'Output Map' section shows 'Starting V address for word outputs' set to 'v21000' and 'Starting V address for bit outputs' set to 'v40515'.

I/O Map with an H2-PBC or T1H-PBC Profibus DP Controller

When using the CTRIO(2) module with an H2-PBC or T1H-PBC native Profibus addressing will be displayed in the I/O Map as shown below. For the T1H-PBC, the first two output bytes of memory are automatically reserved for the Hot Swap base-rescan feature. The H2-PBC does not support the Hot Swap feature.



H2-PBC I/O Map

T1H-PBC I/O Map

Note that output bytes 0-1 are reserved for the Hot Swap base rescan feature.

Creating an Offline File for H2-CTRIO(2)/T1H-CTRIO Module

For the T1H-PBC, enter a 2 byte Output Offset to accommodate memory used by the Hot Swap base-rescan feature. This does not apply to an H2-PBC system. The example shown below assumes the T1H-CTRIO module is the first module in the system.

The CTRIO(2) modules consume 44 bytes of input memory and 52 bytes of output memory. The maximum amount of I/O memory per H2/T1H-PBC station is 244 input bytes and 242 output bytes. You may need to refer to the Profibus User Manuals (H2-PBC-M / T1H-PBC-M) for information on bytes used by discrete and/or analog I/O modules in order to determine the appropriate Starting input and output byte offset addresses for the CTRIO(2) module.

For the T1H-PBC, enter a 2 byte Output Offset to accommodate memory used by the Hot Swap base-rescan feature.

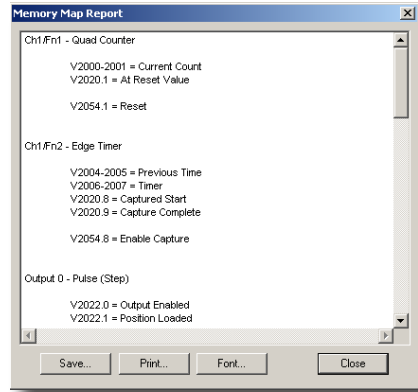
Printing a Memory Map Report

- DL
- Win
- NI

You can print an I/O Memory Map Report from the I/O Map dialog or save as a (.txt) file. Click on the **Report** button located near the bottom of the I/O Map dialog to display the **Memory Map Report** dialog.



The addresses listed in the Memory Map Report are a combination of the Input Functions, Output Functions and Systems Functions addresses shown in the I/O Map dialog. It is very convenient to have a printed list of the CPU/controller I/O memory used by the CTRIO(2) module while writing the control program.



Exporting to DirectSOFT

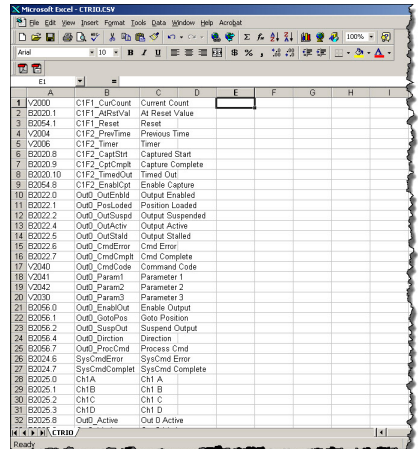
You can export a (.csv) file containing addressing and nicknames used in the I/O Map dialog. Click on the **Export** button located near the bottom of the I/O Map dialog to display the **Export to DirectSOFT** dialog shown below on the left.



- DL
- Win
- NI

The (.csv) file (shown below on right) contains a combination of the Input Functions, Output Functions and Systems Functions addressing and nicknames shown in the I/O Map dialog. This file can be imported into your *DirectSOFT* ladder logic program (*DirectSOFT*>File>Import>Element Documentation).

If you have more than one CTRIO(2) module in a system and intend to create a (.csv) file for more than one module, use the Add Prefix or Add Suffix option to distinguish one module nickname from the others. For example, add prefix or suffix "S1" to identify the CTRIO(2) module nicknames in slot 1.



CTRIO WORKBENCH, MONITOR I/O



In This Chapter...

Using the Monitor I/O Dialog.....	8-2
Monitor I/O Error Codes	8-7

Using the Monitor I/O Dialog



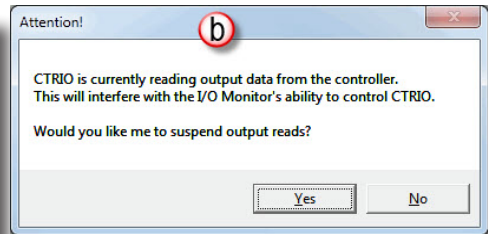
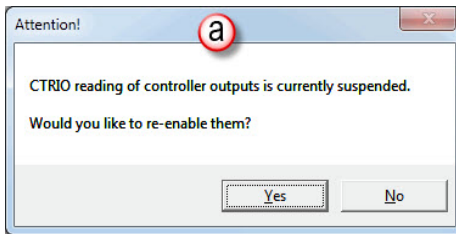
NOTE: It is highly recommended to simulate your CTRIO Counter, Timer or Pulse Output Profile, etc., application using Monitor I/O before attempting to control the module from your controller program. Monitor I/O is extremely useful for debugging and the commissioning of a new system. Monitor I/O allows you to confirm proper configuration of the module, as well as field wiring and external device operation

The Monitor I/O dialog is accessible from the main Workbench dialog when the module is in Run Mode. On the main Workbench dialog, click the button labeled **Monitor I/O**.

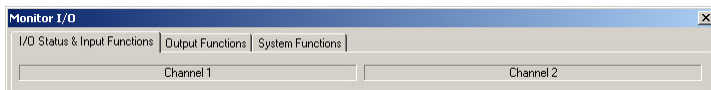


- DL
- Win
- NI

After clicking on the **Monitor I/O** button, the dialog below (a) will appear if you have mapped the I/O in the CTRIO(2) module to the controller. Here you have the ability to suspend CTRIO(2) module reads from the CPU/controller. Doing so will allow Monitor I/O to control the CTRIO(2) module without any control program intervention. With the output reads suspended, the Monitor I/O dialog allows you to simulate program control; for example, enabling a timer, resetting a counter, running a pulse profile or turning on an output configured for Raw mode, etc. When exiting Monitor I/O, you will be prompted (b) to re-enable the controller output reads.



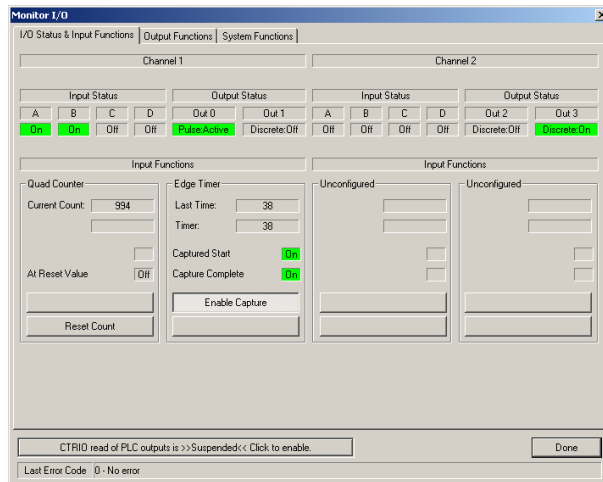
The Monitor I/O dialog is divided into three functional areas: **I/O Status & Input Functions**, **Output Functions** and **System Functions**. Just below the Windows title bar, you will see tabs to switch between the three functions, shown below. The functions are described on the pages that follow.



I/O Status & Input Functions

I/O Status & Input Functions dialog includes all Input Function DWord Parameters (raw count/time, scaled count/time, etc.) and status bits passed from the CTRIO(2) module to the CPU (Capture Starting, Complete bits, etc.). The control bits that would be passed from the CPU to the CTRIO(2) are also included (Function enable bits, etc.).

The current status of each configured input and output is shown just below the **Input Status** and **Output Status** columns.



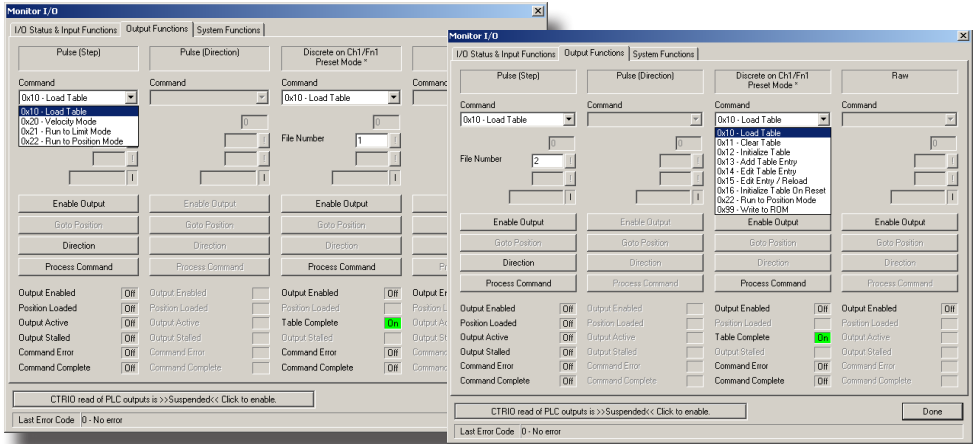
In the dialog panel above, the **Current Count** for Ch1/Fn1 **Quad Counter** is 994. The **Reset Count** button can be used to reset the count to the configured **Reset Value**.

For Ch1/Fn2, the **Edge Timer** is captured at 38us. The **Enable Capture** bit must be on prior to when the configured edge input occurs.

Note that **Output Status - Out 0** and **Out 3** - are **ON**. **Out 0** is configured for pulse output and **Out 3** is configured for a Raw discrete output. These outputs can be controlled from the Output Functions window.

Output Functions

The Monitor I/O Output Functions dialog includes all Output Function Word and DWord Parameters (file number, duty cycle, target position, etc.) and status bits passed from the CTRIO(2) module to the CPU (Output Enabled, Command Complete, etc.). The control bits that would be passed from the CPU to the CTRIO(2) are also included (Enable Output, Go to Position, Direction, etc).



In the example above, Outputs 0 and 1 are configured for Pulse step and direction, Output 2 is configured to Preset mode assigned to Ch1/Fn1 (quad counter) and Output 3 is configured as Raw mode.

In the screen capture on the left, notice the pull down menu. The menus are context sensitive. They will change to display values that are appropriate to the CTRIO(2) module's configuration. Here you have access to all pulse profile commands. Command 0x10 will allow you to load any configured Pulse Profiles (Trapezoidal, S-Curve, Dynamic Positioning, etc.). In the screen capture on the right, you'll see we have selected Pulse Profile number 2 for this example.

To run a configured Pulse Profile, follow these steps:

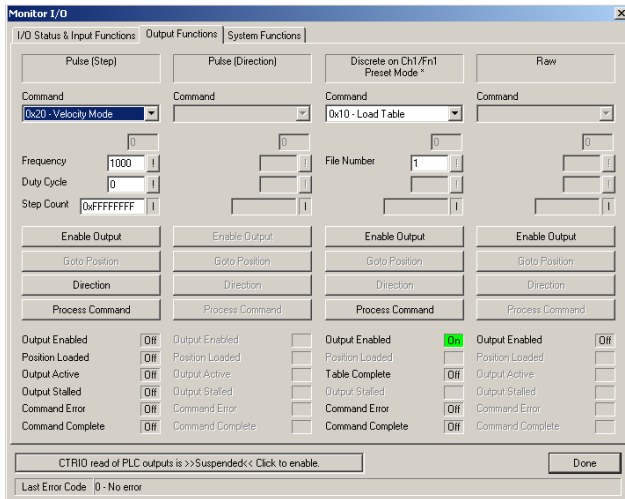
- 1: Select **Command** code 10 (0x10).
- 2: Enter the desired **Pulse Profile Number** in the **File Number** field.
- 3: Click the **Process Command** button and confirm the **Command Complete** bit is **ON**. If the **Command Error** is **ON**, an explanation of the error will appear on the dialog status line. Then turn the **Process Command** button **OFF**.
- 4: Select the **Direction**; leaving the **Direction** button **OFF** selects forward, clicking the button **ON** selects the reverse direction.
- 5: Click on **Enable Output** to run the Pulse Profile. The **Output Enabled** and **Output Active** indicators will turn **ON**. When the profile is complete, the **Output Active** indicator will turn **OFF**.

Turning OFF the **Enable Output** during the profile run will terminate the pulse output. To run the profile again, turn OFF the **Enable Output** and then re-enable it. In the screen capture on the right on the previous page, notice the drop down menu. Here you have access to all of the Preset Table Commands. The Load Table Command (0x10) will allow you to load any configured Preset Tables. In the screen capture on the left, you'll see we have selected Preset Table number 1 for this example. Remember that Output 2 is assigned to Input Function Ch1/Fn1, which is configured as a Quad Counter Input.

To load a configured Preset Table for Output 2 to use based on Ch1/Fn1's count, follow the steps below:

- 1: Select **Command** code 10 (0x10).
- 2: Enter the desired Preset Table Number in the File Number field.
- 3: Click the **Process Command** button and confirm the **Command Complete** bit is ON. If the **Command Error** is ON, an explanation of the error will appear on the dialog status line. Then turn the **Process Command** button OFF.
- 4: Click on the **Enable Output** to allow the output to operate based on the Preset Table and current status of Ch1/Fn1 quad counter input.

As the encoder count on Ch1/Fn1 changes, the output 2 turns ON and OFF based on the entries in Preset Table number 1. Turning the **Enable Output** OFF while the Preset Table is being executed will disable the output.



Pulse Output Command Codes 0x20, 0x21 and 0x22

Velocity mode (0x20) is shown in the example above. Based on which command is selected, different parameter fields, status bits and control bits will apply. No matter which one is selected, be sure to fill in the parameter fields with valid entries (refer to Chapter 9), and then Process the Command.

System Functions

The Systems Functions dialog allows you read from or write to the current input count and the current output pulse count under the following conditions:

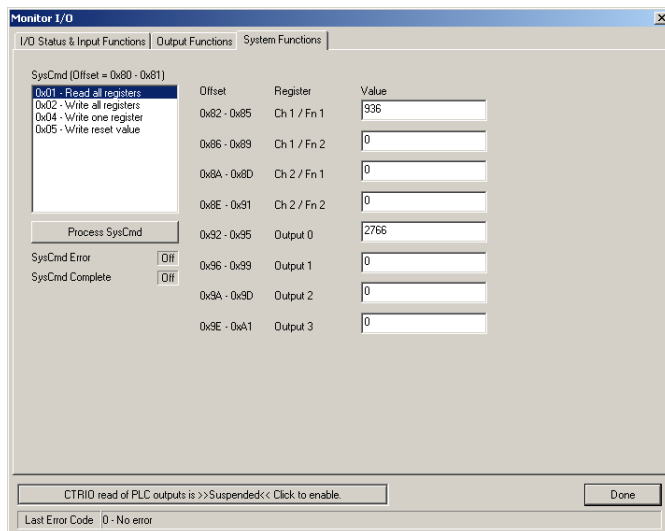
- The current input count can be read from or written to if the input is configured for a Counter or Quad Counter. Timer values are not accessible.
- The current output pulse count can be read from or written to only if the pulse output is running Dynamic Velocity or Dynamic Positioning profiles.

DirectLogic Users

The reading from and writing to the CTRIO(2) module's internal registers is accomplished using the DirectLOGIC Read from Intelligent module (RD) and Write to Intelligent module (WT) instructions, respectively. Refer to Appendix B for Systems Functions ladder logic examples.

EBC, WinPLC, PBC, DEVNETS, MODBUS Users -

The Systems Functions dialog is available for use when connected to these interface devices, however, there is currently no way for the user control program to read from or write to the CTRIO(2) module internal registers.

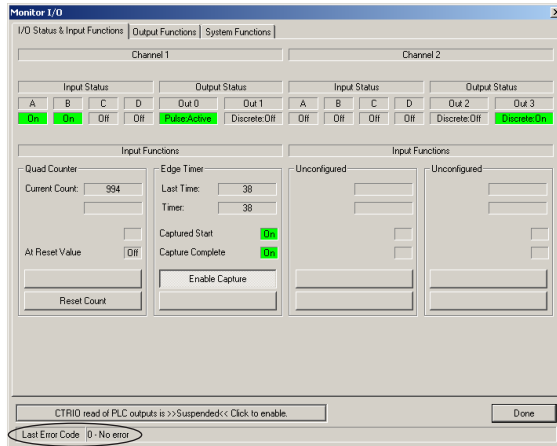


Monitor I/O Error Codes

The appropriate error code listed below will be displayed on the Monitor I/O Status Bar when an error occurs.

- DL
- Win
- NI

Error Code	Description
0	No error
100	Specified command code is unknown or unsupported
101	File number not found in file system
102	File type is incorrect for specified output function
103	Profile type is unknown
104	Specified input is not configured as a limit on this output
105	Specified limit input edge is out of range
106	Specified input function is unconfigured or invalid
107	Specified input function number is out of range
108	Specified preset function is invalid
109	Preset table is full
110	Specified table entry number is out of range
111	Specified register number is out of range
112	Specified register is in unconfigured input or output



Status bar

OUTPUT FUNCTIONS



CHAPTER 9

In This Chapter...

Runtime Changes to CTRIO Configured Preset Tables (DL PLCs)	9-2
Pulse Output Profiles (DL PLCs).....	9-5
Trapezoid Profile	9-6
S-Curve Profile	9-7
Symmetrical S-Curve Profile.....	9-8
Home Search Profile.....	9-9
Free Form Profile	9-12
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-13
DirectLOGIC Programming Examples Overview	9-19
Trapezoid with Limits Profile	9-20
Trapezoid with Limits (CTRIO2) Profile.....	9-21
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-24
Trapezoid Plus (CTRIO2) Profile.....	9-28
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-30
Load and Run a Trapezoid Plus Profile	9-33
Dynamic Positioning Plus and Dynamic Positioning Profiles.....	9-34
Dynamic Positioning Plus (CTRIO2) Profile.....	9-35
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-36
Dynamic Velocity Profile.....	9-42
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-43
Velocity Mode	9-48
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-49

Run to Limit Mode	9-54
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-55
Run to Position Mode	9-61
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-62
Run to Position Mode with DirectSOFT IBox Instructions	9-67
Raw Output.....	9-69
Pulse Output Status/Control Bits and Command Codes (DL PLCs)	9-70

Runtime Changes to CTRIO Configured Preset Tables (DL PLCs)



Presets and preset tables can be set up entirely within CTRIO Workbench so that no program control is necessary to assign discrete Preset Tables to CTRIO Input Functions.

You can make runtime edits to presets/preset tables from your control program. To make a runtime change, a series of commands must be executed which will pass new values to a preset table or call a different pre-configured table.

Command Codes are passed to the CTRIO(2) module to effect the required edit. Each Command Code has its own syntax, and all Command Codes must be presented in a particular sequence:

- The command code and associated parameters must be loaded into the appropriate memory locations.
- A Process Command instruction must be passed to the CTRIO(2) module.
- A Command Complete signal must be received and the Command Error bit must stay at zero.
- Finally, the Enable Output instruction must be passed to the CTRIO(2) module.

Some changes require a combination of Command Codes so those changes must follow the steps above for each Command Code processed (Output Control and Status Offsets are listed in order of Output 0 - Output 3).

Control Bit (Transfers from CPU to CTRIO(2))	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets DirectLOGIC PLCs
Enable Output	32, 40, 48, 56	26.0, 26.8, 27.0, 27.8
Process Command	39, 47, 55, 63	26.7, 26.15, 27.7, 27.15

Status Bit (Transfers from CTRIO(2) to CPU)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets DirectLOGIC PLCs
Command Error	38, 46, 54, 62	22.6, 22.14, 23.6, 23.14
Command Complete	39, 47, 55, 63	22.7, 22.15, 23.7, 23.15

In order to process a command, first the program must load the Command and Required Word and DWord Parameters. Then the program should turn ON the Process Command bit and look for the CTRIO(2) module to acknowledge the command with the Command Complete bit. Finally the program should reset the Process Command bit and set the Enable Output bit when appropriate. If the Command Error bit is received, the CTRIO(2) module was unable to process the command due to an illegal value in either the Command Code or Parameter fields.

Word Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
Command Code	0, 6	10, 16
Word Parameter 1	1, 7	11, 17
Word Parameter 2	2, 8	12, 20
DWord Parameter 3	0, 2	0, 4

Command	See Topic in Manual	Code (hex)	Parameter 1 (Word)	Parameter 2 (Word)	Parameter 3 (Word)
<i>Direct</i> LOGIC n+10			<i>Direct</i> LOGIC n+11	<i>Direct</i> LOGIC n+12	<i>Direct</i> LOGIC n+0/n+1
Load File	Load Table, Load a Pulse Profile (Trapezoid, S-Curve, Symmetrical S-Curve, Home Search, Free Form)	10	File No. (decimal)	N/A	N/A
Load File	Trapezoid with Limits	10	File No. (decimal)	N/A	Stop Position, Optional (decimal)
Load File	Trapezoid Plus	10	File No. (decimal)	N/A	Target Position (decimal)
Load File	Dynamic Positioning, Dynamic Positioning Plus	10	File No. (decimal)	N/A	Target Position (decimal)
Load File	Dynamic Velocity	10	File No. (decimal)	N/A	Target Velocity (decimal)
Clear Preset Table	Clear Preset Table	11	N/A	N/A	N/A
Create Preset Table	Create Preset Table	12	Entry Type (decimal)	Pulse Time ¹	N/A
Add Table Entry ⁷	Add Entry to Preset Table	13	Entry Type (decimal)	Pulse Time ¹	Preset Count ⁵ (decimal)
Edit Table Entry ⁷	Edit Preset Table Entry	File No. & ² 14	Entry Number ³ & ² Entry Type ⁴ (hex)	Pulse Time ¹ (decimal)	Preset Count ⁵ (decimal)
Edit and Reload ⁷	Edit Preset Table Entry and Reload	File No. & ² 15	Entry Number ³ & ² Entry Type ⁴ (hex)	Pulse Time ¹ (decimal)	Preset Count ⁵ (decimal)
Initialize Table on Reset ⁷	Create Preset Table on Reset	16	Entry Type (decimal)	Pulse Time ¹ (decimal)	N/A
Pulse at Velocity	Velocity Mode	20	Frequency (decimal)	Duty Cycle (decimal)	Number of Pulses

¹ If appropriate for Entry Type (in ms).

² Field entries separated by an "&" are to be loaded in the high byte and low byte of that word. See example on page 9-15.

³ Entry numbers are defined below.

⁴ Entry types are defined below.

⁵ Follows format of Input DWord Parameter 1.

⁶ Flash ROM is rated for 100,000 writes.

⁷ Counter/Quad Counter Reset must be ON to perform the Edit.

Command	See Topic in Manual	Code (hex)	Parameter 1 (Word)	Parameter 2 (Word)	Parameter 3 (Word)
<i>Direct</i> LOGIC n+10			<i>Direct</i> LOGIC n+11	<i>Direct</i> LOGIC n+12	<i>Direct</i> LOGIC n+0/n+1
Run to Limit	Run to Limit	21	Frequency (decimal)	Input /Edge & ² Duty Cycle (0 - 99)* (Hex/BCD)	N/A
Run to Position ⁷	Run to Position	22	Frequency (decimal)	Function (hex) & Duty Cycle	Target Position (decimal)
Edit Level Response ⁷	Update Level	30	Function (hex)	Deadband (decimal)	Scaled Level (decimal)
Write RAM to ROM ⁷	Write File to ROM	99 ⁶	N/A	N/A	N/A

* A value of 0 will generate a duty cycle of 50%.

¹ If appropriate for Entry Type (in ms).

² Field entries separated by an "&" are to be loaded in the high byte and low byte of that word. See example on page 9-15.

³ Entry numbers are defined below.

⁴ Entry types are defined below.

⁵ Follows format of Input DWord Parameter 1.

⁶ Flash ROM is rated for 100,000 writes.

⁷ Counter/Quad Counter Reset must be ON to perform the Edit.

Entry Number for Edit Table Entry Commands

The Entry Number refers to the position of the preset in the table sequence. The first preset is Entry Number 0, the second preset is 1 and so forth.

Entry Type for Edit Table Entry Commands

The Entry Type is defined according to the table below.

Entry Type	Code	Notes
Write Output ON (Set)	0	-
Write Output OFF (Reset)	1	-
Pulse Output ON	2	-
Pulse Output OFF	3	-
Toggle Output	4	-
Reset Function	5	Edits preset that resets count

Pulse Output Profiles (DL PLCs)

The two charts below summarize the capabilities and requirements of the various Pulse Output Profiles. Information on each is found in this chapter.

Options	Position Source	Change target position on the fly	Accel/decel ramps	Maximum pulse rate	Encoder Feedback Possible	Requires CTRIO2	See Page(s)
Dynamic Positioning Plus	From controller	Yes	Specify separately	250kHz	Yes	Yes	9-34
Trapezoid Plus							9-28
Dynamic Positioning	Hard-coded in pulse profile	No	One setting for both	65kHz ¹	No	No	9-34
S-Curve			Specify separately				9-7
Symmetrical S-Curve			One setting for both				9-8
Trapezoid			Specify separately				9-6
Free Form			None				9-12

¹The older CTRIO outputs can only achieve 25kHz maximum. CTRIO2 outputs are limited to 65kHz when using these profiles.

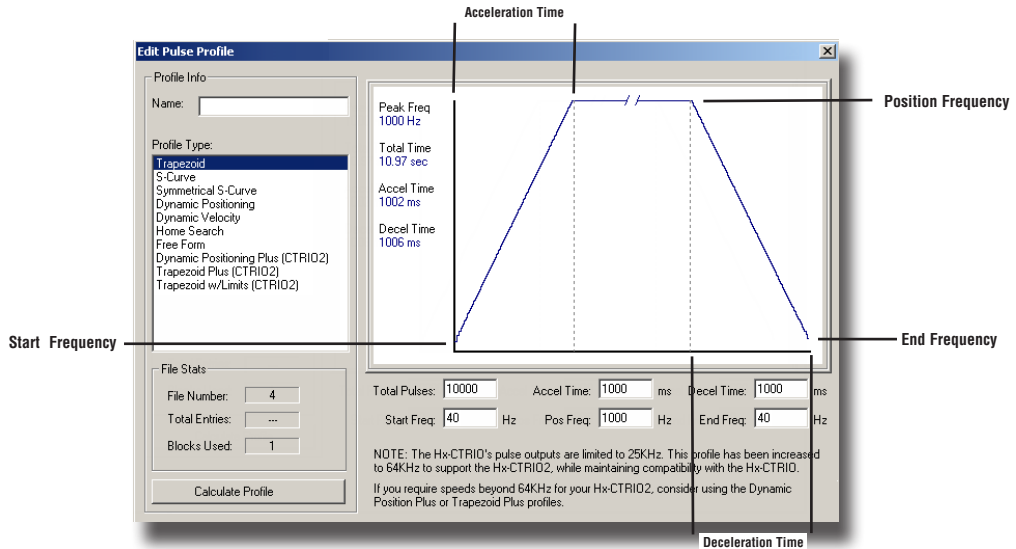
Profile	Accel and Decel Ramps	Creep to second limit or position	Change velocity manually on the fly	Multiple limits or triggers on a single home search	Single input can act as multiple triggers	Profile defined in CTRIO Workbench	Requires CTRIO2	See Page(s)
Trapezoid with Limits	Yes	Yes	No	Yes	Yes	Yes	Yes	9-21
Home Search	No	No		No	No		No	No
Run to Limit			Yes	No		9-54		

Fixed, Predefined Profiles

Trapezoid, S-Curve, Symmetrical S-Curve, Home Search and Free Form profiles are the most basic. The parameters for these profiles are defined in CTRIO Workbench and cannot be changed programmatically.

Trapezoid Profile

The Trapezoid profile changes the velocity in a linear fashion from the specified Start Frequency until the specified target Position Frequency is reached. During deceleration, the velocity changes in a linear fashion from the specified Position Frequency until the specified End Frequency and Total Pulses is reached.

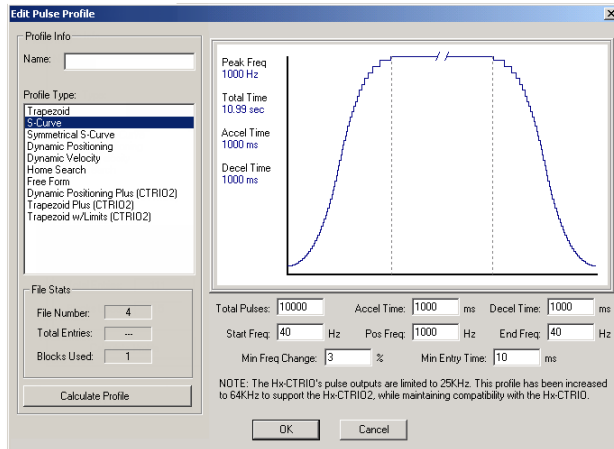


- **Total Pulses:** The total number of output pulses that will be generated during the Trapezoid profile.
- **Accel Time:** The amount of time required for the Start Frequency to ramp up to the Position Frequency.
- **Decel Time:** The amount of time required for the Position Frequency to ramp down to the End Frequency.
- **Start Freq:** The frequency at which the Trapezoid profile will begin.
- **Pos Freq:** The maximum steady state frequency the profile can attain during a move.
- **End Freq:** The lowest Frequency of movement during deceleration. The Frequency drops to zero upon reaching this value.

S-Curve Profile

The S-Curve profile can be used for applications that are sensitive to sudden changes in the rate of acceleration.

- DL
- Win
- NI

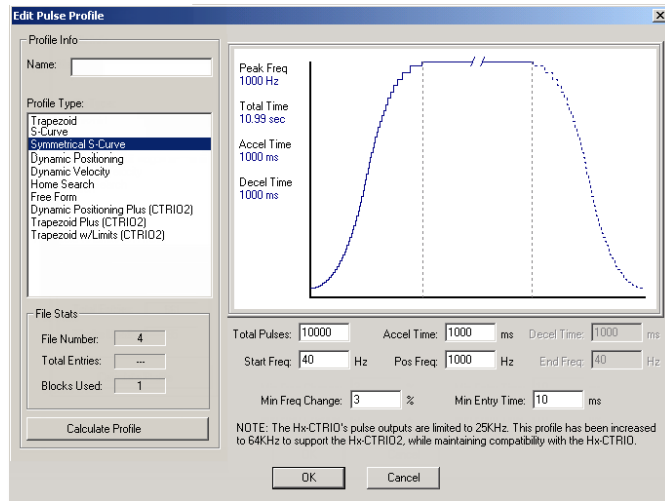


- **Total Pulses:** The total number of output pulses that will be generated during the S-Curve profile.
- **Accel Time:** The amount of time required for the Start Frequency to ramp up the Position Frequency.
- **Decel Time:** The amount of time required for the Position Frequency to ramp down to the End Frequency.
- **Start Freq:** The frequency at which the S-Curve profile will begin.
- **Pos Freq:** The maximum steady state frequency the profile can attain during a move.
- **End Freq:** The lowest Frequency of movement during deceleration. The Frequency drops to zero upon reaching this value.
- **Min Freq Change:** The amount of calculated frequency change that must take place before stepping to the next frequency.
- **Min Entry Time:** The amount of time spent in each step.

Symmetrical S-Curve Profile

The Symmetrical S-Curve profile can also be used for applications that are sensitive to sudden changes in the rate of acceleration.

The S-Curve and Symmetrical S-Curve profiles differ in that the Symmetrical S-Curve has symmetrical acceleration and deceleration profiles. The Decel Time and End Frequency are determined by the Accel Time and Start Frequency. The Symmetrical S-Curve uses less memory than the S-Curve profile.



- **Total Pulses:** The total number of output pulses that will be generated during the Symmetrical S-Curve profile.
- **Accel Time:** The amount of time required for the Start Frequency to ramp up the Position Frequency. This also represents the deceleration time.
- **Start Freq:** The frequency at which the Symmetrical S-Curve profile will begin. This also represents the End Frequency.
- **Pos Freq:** The maximum steady state frequency the profile can attain during a move.
- **Min Freq Change:** The amount of calculated frequency change that must take place before stepping to the next frequency
- **Min Entry Time:** The amount of time spent in each step.

Home Search Profile

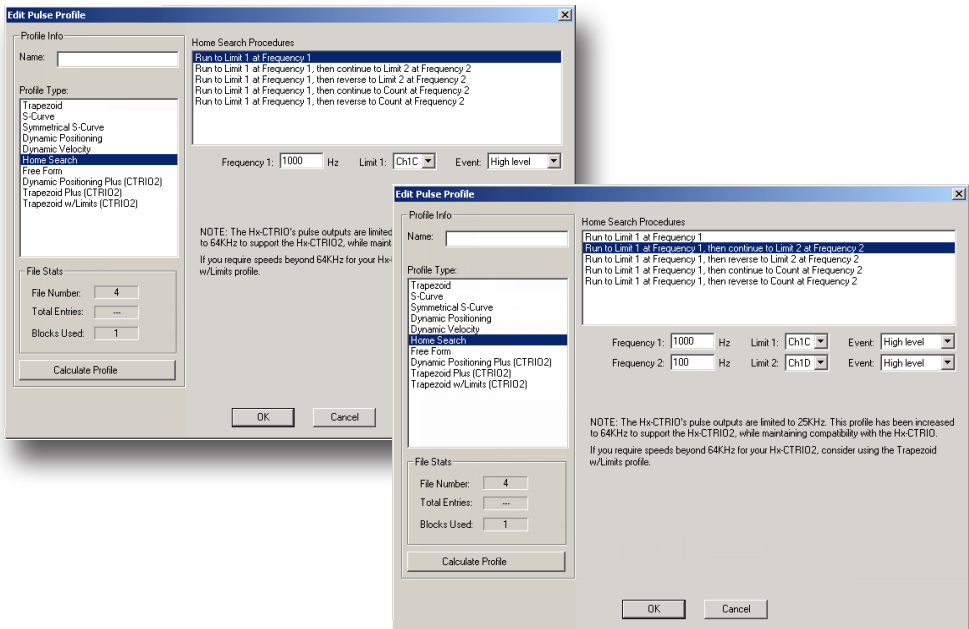
- DL
- Win
- NI

The Home Search profile is used to “*find the home position*”, which is usually a reference point to which the object being moved can return upon command at any time during or after the execution of a positioning profile.

There are several Home Search routines to choose from, all with the option to designate whether you want Limit 1 and/or Limit 2 (a CTRIO(2) discrete input) to register on the rising edge, falling edge, high level or low level signal. Limit 1 and Limit 2 can be the opposite edges of the same physical CTRIO(2) input.



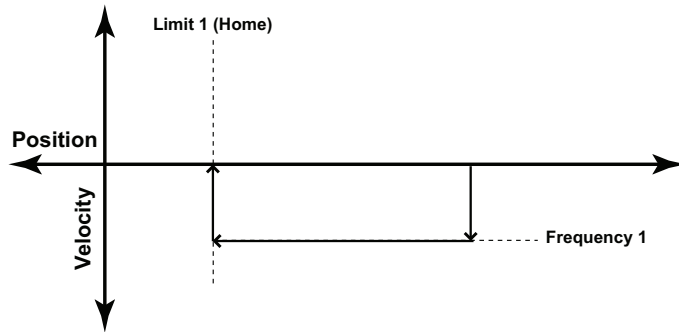
NOTE: The Home Search profile requires that CTRIO(2) inputs C and/or D are configured for Limit Out 0 or Limit 2. This is done using the Configure IO dialog.



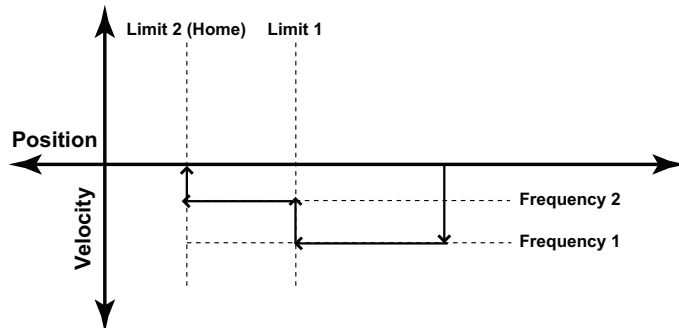
- **Frequency 1:** The frequency at which the Home Search will begin.
- **Limit 1:** Home Search Frequency 1 will run to CTRIO(2) input Limit 1 and stop unless Frequency 2 is enabled.
- **Frequency 2:** (if enabled) Once Limit 1 is reached, the pulse output will continue at Frequency 2 until CTRIO(2) Limit 2 is reached or pulse Count is reached at Frequency 2.
- **Limit 2:** (if enabled) Home Search Frequency 2 will run to CTRIO(2) input Limit 2 and stop.
- **Count:** (if enabled) The number of output pulse counts generated at Frequency 2 before terminating.
- **Event:** Specifies the condition that will terminate this leg of the Home Search.

Home Search Routines

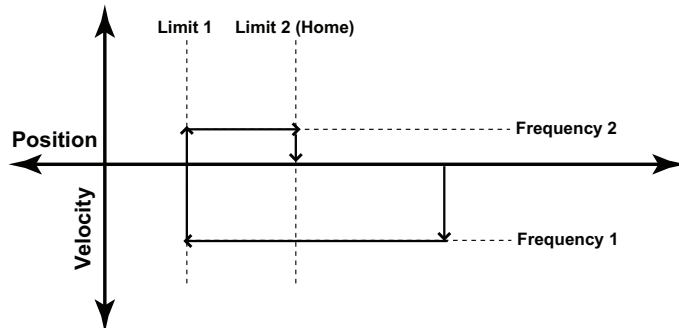
1: Run to Limit 1 at Frequency 1.



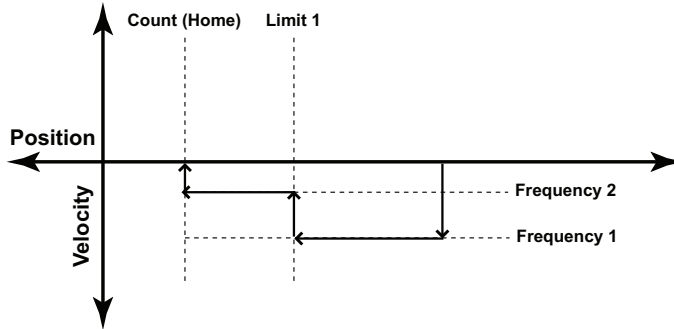
2: Run to Limit 1 at Frequency 1, then continue to Limit 2 at Frequency 2.



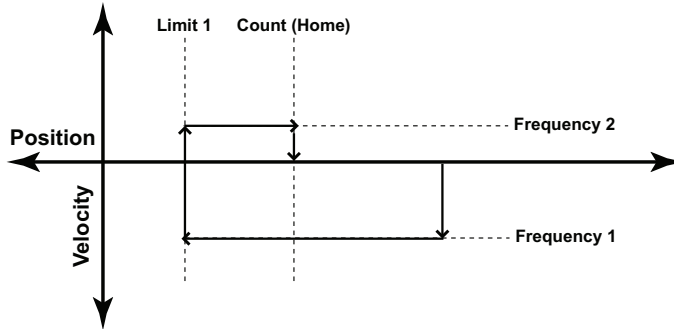
3: Run to Limit 1 at Frequency 1, then reverse to Limit 2 at Frequency 2.



4: Run to Limit 1 at Frequency 1, then continue to Count at Frequency 2.

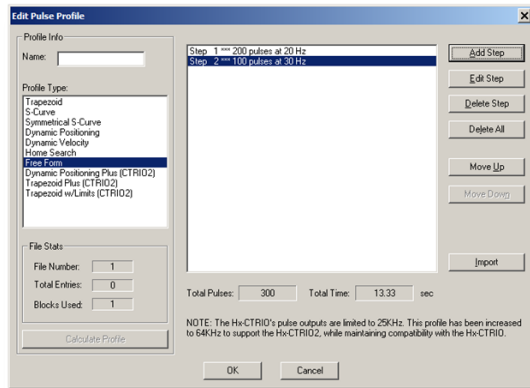


5: Run to Limit 1 at Frequency 1, then reverse to Count at Frequency 2.



Free Form Profile

The Free Form profile allows for stepping between output frequencies with no acceleration or deceleration ramps. Profiles, up to 256 steps, can be imported from a CSV file.



Total Pulses: Provided by the utility, the total number of output pulses that will be generated during the profile.

Total Time: The total time required for the profile to run to completion.

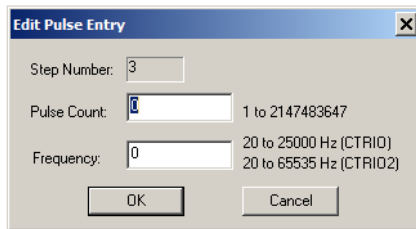
Import: Opens a dialog that allows importing a CSV file. Importing the CSV replaces the existing entries. The contents of a CSV file used to create the profile seen above would appear like this:

200,20

100,30

Add or Edit Step: Will invoke the Edit Pulse Entry dialog above. This window will allow you to modify or create pulse entries.

- Pulse Count: The number of pulses to generate for this step.
- Frequency: The frequency of pulses during this step.



Pulse Output Status/Control Bits and Command Codes (DL PLCs)

- DL
- Win
- NI

Output Status Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Status Bit CTRIO(2) to CPU	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Input Start (octal)
Output Enabled	32, 48	22.0, 23.0
Position Loaded	33, 49	22.1, 23.1
Output Suspended	34, 50	22.2, 23.2
Output Active	36, 52	22.4, 23.4
Output Stalled	37, 53	22.5, 23.5
Command Error	38, 54	22.6, 23.6
Command Complete	39, 55	22.7, 23.7

Output Control Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Control Bit CPU to CTRIO(2)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)	Read as:
Enable Output	32, 48	26.0, 27.0	Level
Go to Position	33, 49	26.1, 27.1	Rising Edge
Suspend Output	34, 50	26.2, 27.2	Level
Direction	36, 52	26.4, 27.4	Level
Process Command	39, 55	26.7, 27.7	Rising Edge

Output Control (D)Words (Pulse Output)

Word Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
Command Code	0, 6	10, 16
Word Parameter 1	1, 7	11, 17
Word Parameter 2	2, 8	12, 20

DWord Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
DWord Parameter 3	0, 2	0, 4

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Trapezoid, S-Curve, Symmetrical S-Curve, Home Search, Free Form Profiles

- DL
- Win
- NI

For predefined Trapezoid, S-Curve, Symmetrical S-Curve, Home Search and Free Form profiles, the program must prepare the Load Table command (see table on next page), by selecting Command Code = 0010 Hex/BCD and setting Word Parameter 1 to the File number of the profile (example: File 1 Trapezoid 1). Then the program sets the process **Command** bit and watches for the **Command Complete** bit. Then the program will clear the process **Command** bit and set the **Direction** bit (if necessary) and finally the **Enable Output** bit to start the output pulses.

Clearing the **Enable Output** bit will always suspend pulsing and reset any profile in progress to its beginning. Once complete, the profile remains loaded and can be restarted by clearing the **Enable Output**, changing the **Direction** bit (if desired), and again setting the **Enable Output**.

For the Home Search routine, a CTRIO(2) input must be assigned to Limit by the CTRIO Workbench *Configure I/O* dialog.



NOTE: For a Home Search Profile, if you are at the home position and the Home Search profile is initiated, there will not be any pulse outputs.

A Pulse Profile programming example, that loads and runs a pulse profile implementing the bit/(D)Word addressing suggested in the table on the page 9-16, is discussed in pages following.

Command Code and Parameter Definitions

Command	Code (Hex/BCD)	Word Parameter 1 (decimal)	Word Parameter 2	DWord Parameter 3
Load File from ROM	10	File Number Trapezoid or S-curve Symmetrical S-Curve Home Search	N/A	N/A
Load File from ROM	10	File Number Dynamic Positioning		Target Position (decimal)
Load File from ROM	10	File Number Dynamic Velocity		Target Velocity (decimal)
Velocity Mode	20	Run Frequency (CTRIO: 20Hz - 25KHz CTRIO2: 20Hz - 64KHz)	Duty Cycle (0 to 99)* (decimal)	Number of Pulses (decimal)
Run to Limit Mode	21	Run Frequency (CTRIO: 20Hz - 25KHz CTRIO2: 20Hz - 64KHz)	Input / Edge & Duty Cycle (0 to 99)* (Hex/BCD)	N/A
Run to Position Mode	22	Run Frequency (CTRIO: 20Hz - 25KHz CTRIO2: 20Hz - 64KHz)	Compare Function & Duty Cycle (0 to 99)* (Hex/BCD)	Desired Input Function Value (decimal)

* A value of 0 will generate a duty cycle of 50%

Where the fields of the Parameters in the table above are separated by an “&” character, this indicates a code with different definitions for each byte (high byte and low byte). For example, to enter the **Pulse Output to Limit** command, set the high byte of the *Word Parameter 2* to the Edge used to terminate the output pulses (see definition following), and set the low byte to the desired duty cycle.

In order to process a command, first the program must load the **Command Code** and required DWord, Word, and bit parameters (DWord and Word values for pulse outputs are unsigned integers). Then the program will turn ON the process **Command** bit and look for the CTRIO(2) module to acknowledge the command with the **Command Complete** bit. Finally, the program will reset the process **Command** bit and set the **Enable Output** bit when appropriate. If the **Command Error** bit is received, the CTRIO(2) module was unable to process the command due to an illegal value in either the **Command Code** or parameter files.

Running a Trapezoid, S-Curve, Symmetrical S-Curve Profile, Home Search or Free Form Profile on CTRIO(2) Y0 & Y1

- DL
- Win
- NI

The required basic steps for a DL-PLC to run a Trapezoid, S-Curve, Symmetrical S-Curve profile, Home Search or Free Form profile are outlined in this table. The steps coordinate with the flow chart steps on page 9-21 (left hand side). Pulse Output Status/Control Bits are shown on the following page.

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	V2040	N/A	V2040	N/A	Set to 10 (Load Stored Profile)
2	Parameter 1	V2041		V2041		File # of stored profile, determined by user
	Parameter 3	V2031 - V2030		V2031 - V2030		Target Position: User Defined (DWord) Target Position is only used with Trapezoid Plus and Trapezoid with Limits
3	Process Command	V2056.7		C227		Turn ON until Command Complete status bit is returned (see step 4)
4	Command Complete Status	N/A	V2022.7	N/A	C127	When ON, Profile is now loaded, clear Process Command bit (step 3)
5	Command Error		V2022.6		C126	ON if Command or Parameters are invalid
6	Set Direction	V2056.4	N/A	C224	N/A	Set ON or OFF for Direction of Rotation
7	Enable Output	V2056.0		C220		Turn ON to start pulses
8	Output Enable Status	N/A	V2022.0	N/A	C120	When ON, module is confirming Enable Output
9	Output Active Status		V2022.4		C124	When ON, module is pulsing, OFF with Enable Status ON = profile has completed
10	Disable Output	V2056.0	N/A	C220	N/A	Turn OFF when pulse status is OFF and Enable Status is ON
11	Suspend Output	V2056.2		C222		Turn ON to "pause" output pulses without resetting pulse count
12	Output Suspended	N/A	V2022.2	N/A	C122	ON when out pulse train has been suspended

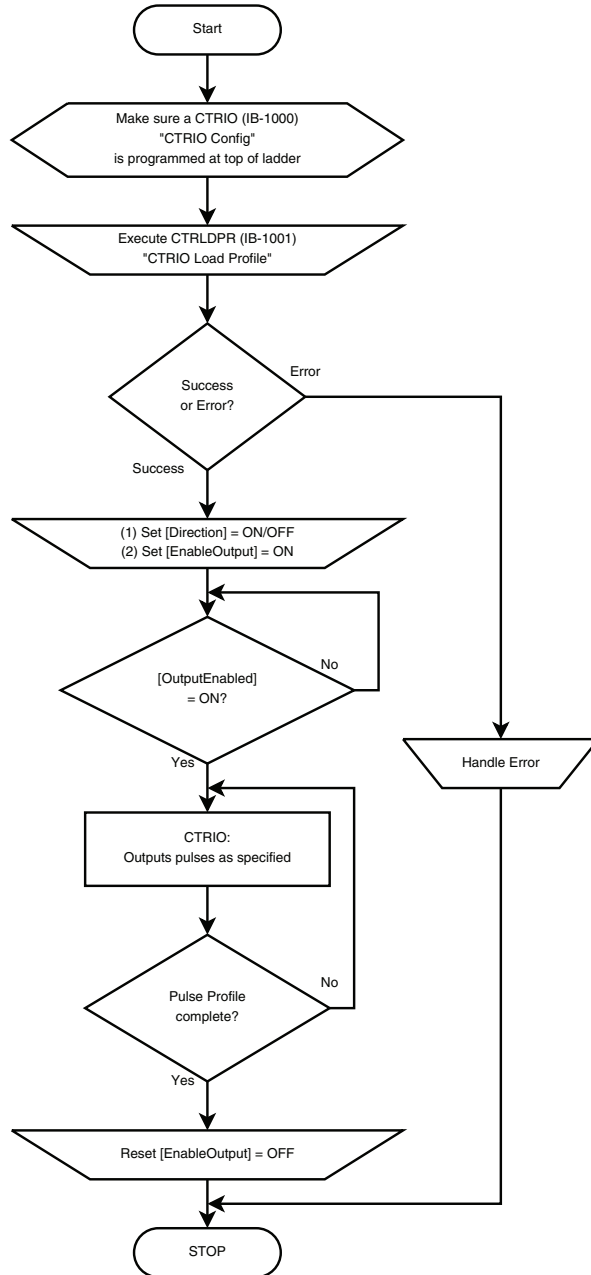
**Run Pulse Profile
(Trapezoid, S-Curve, Symmetrical S-Curve,
Home Search, Free Form)
(IBoxes)**

PARAMETERS:

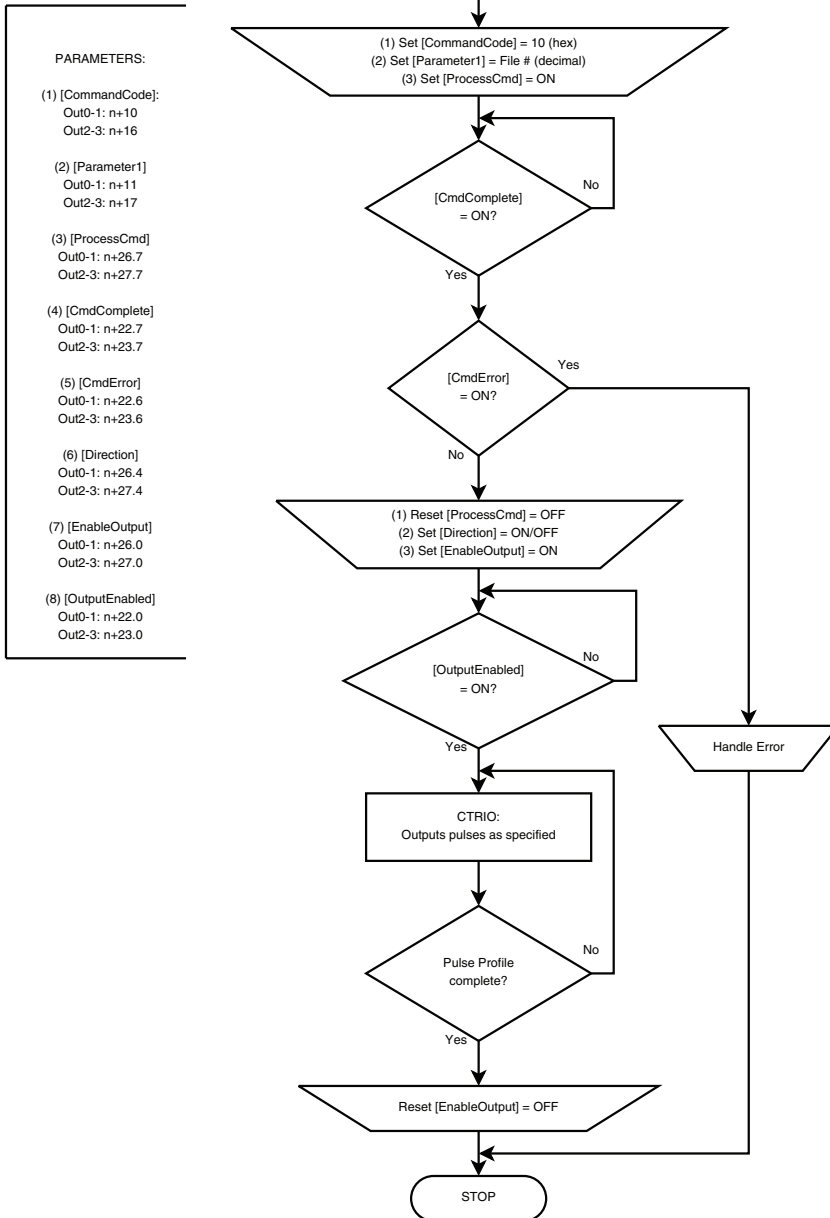
(1) [Direction]
Out0-1: n+26.4
Out2-3: n+27.4

(2) [EnableOutput]
Out0-1: n+26.0
Out2-3: n+27.0

(3) [OutputEnabled]
Out0-1: n+22.0
Out2-3: n+23.0



**Run Pulse Profile
(Trapezoid, S-Curve, Symmetrical S-Curve,
Home Search, Free Form)
(DL-PLC)**



DirectLOGIC Programming Examples Overview



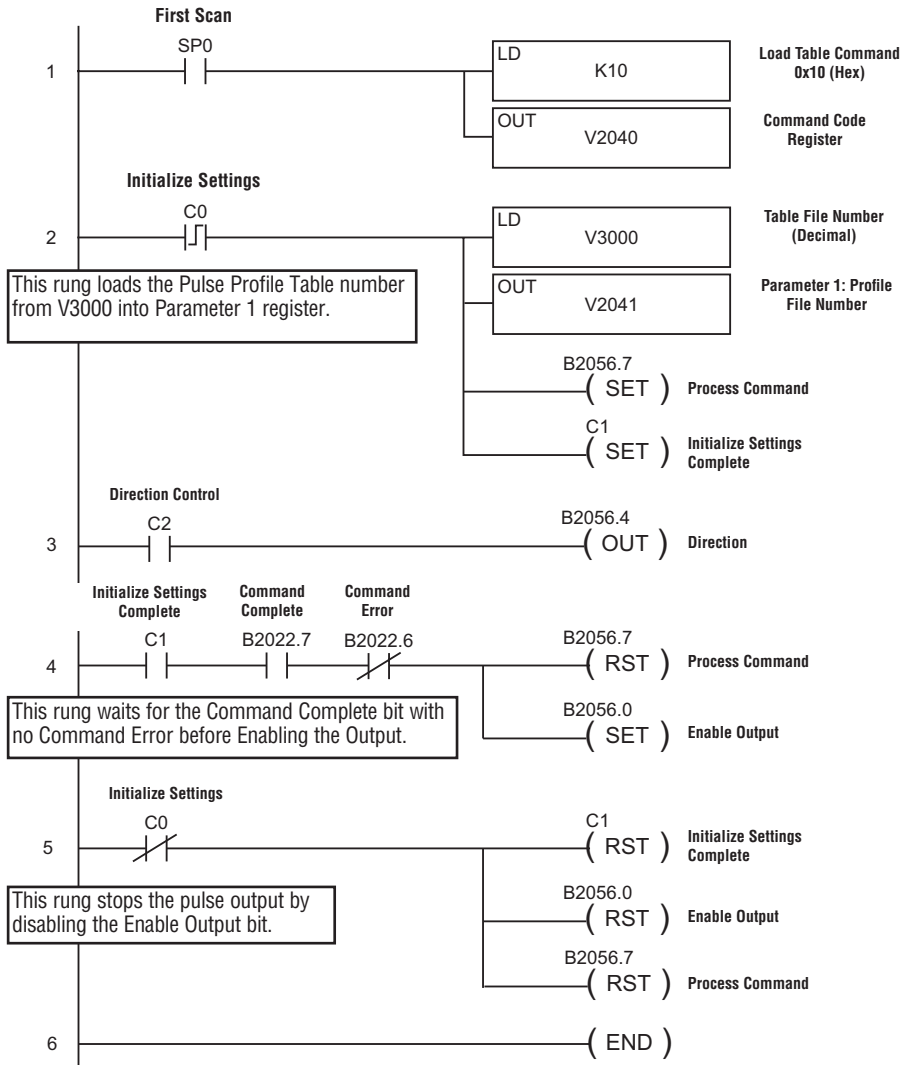
NOTE: The *DirectSOFT* programming examples provided on the following pages are simple examples that are intended to assist you in the basics of loading and running various output pulse profiles. The examples are complete enough to load a profile, process the command and load the Parameter registers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

Load and Run a Pulse Profile

- DL
- Win
- NI

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. You will also need to have the appropriate Pulse Profile Table File Number (decimal) stored in V3000 for this example. You must turn C0 on to load and run the pulse profile. C2 controls the pulse output direction.

The example program on the following page loads and executes a Pulse Profile that was created using CTRIO Workbench Pulse Profiles dialog. This example can be used for Trapezoid, S-Curve, Symmetrical S-Curve, Home Search and Free Form profiles (Home Search requires that CTRIO inputs C and/or D are configured for Limit Out 0 and/or Limit Out 2). The Pulse Profile number is stored in V3000 for this example. Turning on C0 will load and run the pulse profile.



Trapezoid with Limits Profile

Trapezoid with Limits is used for homing an axis by stopping a movement based on limit inputs tied to one or two input channels of the CTIO2. Several combinations of input events are available to trigger the deceleration to the creep frequency and to trigger the actual stop. One of the possible stop triggers is position. If used, indicate the desired stop position in Parameter 3.

Trapezoid with Limits (CTRIO2) Profile

- DL
- Win
- NI

Trapezoid with Limits (CTRIO2) profile is only available when using a CTRIO2 module. Trapezoid with Limits is a homing routine, typically used to find a hard reference point for the axis. There are other routines to home an axis, but this is typically the one to use. The profile is a homing routine similar to Home Search but has five additional features:

1. The profile is trapezoidal (has linear accel and decel rates), allowing for faster homing routines without stalling the stepper.
2. The profile can be asymmetrical. (Accel and Decel rates are separate)
3. Encoder Feedback can be added, useful for correcting excessive lash or slippage in the system.
4. Allows output rates up to 250kHz.
5. When Stop Trigger is position, it is a variable (parameter 3 in the output memory map), set by the base controller.



NOTE: When using an input channel as the Stop Trigger with this profile, the Stop Trigger must be beyond the first limit in the same direction. The output cannot change direction to reach the second limit. Use Home Search if the output must Reverse to Limit 2.

This profile offers several routines using a Decel Trigger and Stop Trigger. Specify the Decel Trigger as the rising or falling edge of a CTRIO2 input. The Stop Trigger can be a position relative to the Decel Trigger, the opposite edge of the Decel Trigger or the rising or falling edge of another CTRIO2 input.



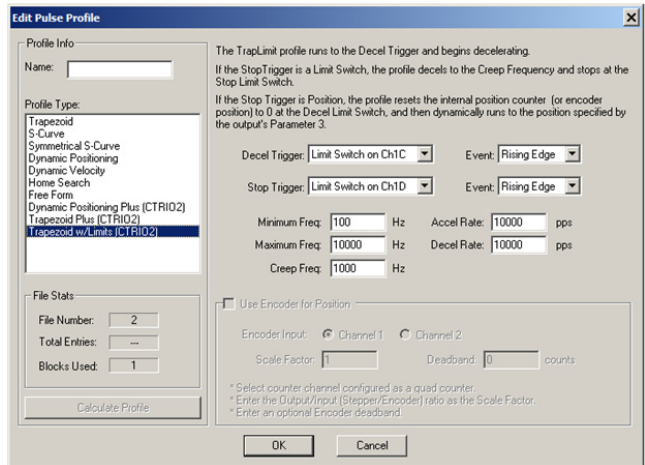
NOTE: This profile requires that CTRIO2 inputs C and/or D are configured for Limit Out X, where X is the output channel being used.

Decel Trigger: The CTRIO2 input used as the Decel Trigger. Hitting this limit during the move will either reset counts to 0 (if Stop Trigger is position) or initiate deceleration to the Creep Frequency (if Stop Trigger is a CTRIO2 input).

Event: The edge of the limit switch used as the Decel Trigger.

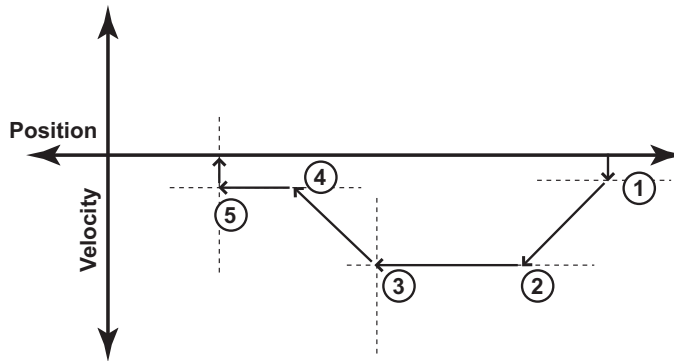
Stop Trigger: The CTRIO2 input or position that stops the output. If Position is selected, Parameter 3 in the output memory map defines the target position. This is relative to 0, which is clocked in when the Decel Trigger Event condition is met. Position can be positive or negative.

Event: The edge of the limit switch used as the Stop Trigger.



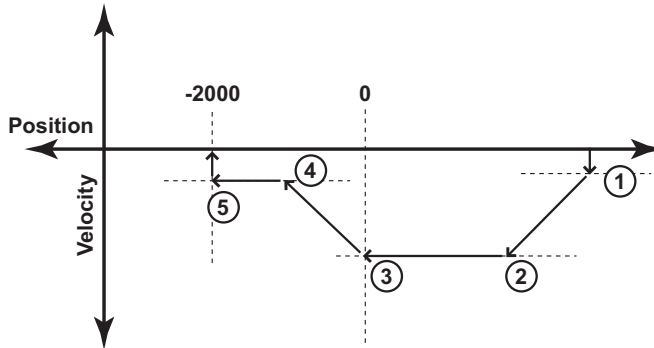
- **Minimum Freq:** The frequency at which the profile will begin and end.
- **Maximum Freq:** The maximum steady state frequency the profile can attain during a move.
- **Accel Rate:** The rate at which the output frequency will increase during the acceleration period.
- **Decel Rate:** The rate at which the output frequency will decrease when Decel Trigger is reached.
- **Creep Freq:** The (slower) rate to use between the Decel Trigger and the Stop Trigger.
- **Encoder Input:** Select the channel where the encoder is connected.
- **Scale Factor:** This is the output to input resolution (stepper/encoder) ratio. In other words, if the stepper motor being used is a 1000 ppr (pulses per revolution) and the encoder is 800 ppr, then the scale factor would be $1000/800 = 1.25$.
- **Deadband:** This is the number of position counts away from the target position that causes no action. This can reduce “hunting” or “ringing” as the profile attempts to get to the target position. Be sure to include a deadband when the encoder has a higher resolution than the stepper.

Example: Trapezoid with Limits using a limit switch for the Stop Trigger.



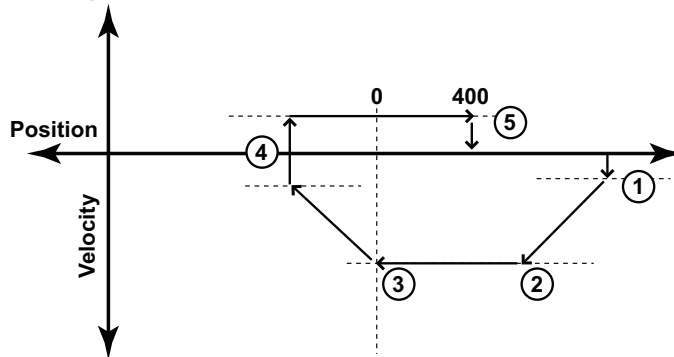
- 1: Move starts at the Minimum Freq and accelerates at Accel Rate.
- 2: Acceleration ends at Maximum Freq and move continues at Maximum Freq.
- 3: When the Decel Trigger is reached, the position register is zeroed out and the output begins to slow down at Decel Rate.
- 4: Deceleration ends at Creep Freq and the move continues towards the Stop Trigger, a limit switch.
- 5: When the Stop Trigger is reached, the output stops.

Example: Trapezoid with Limits using a position for the Stop Trigger. The Stop Trigger position does not require change of direction.



- 1: Move starts at the Minimum Freq and accelerates at Accel Rate.
- 2: Acceleration ends at Maximum Freq and move continues at that rate.
- 3: When the Decel Trigger is reached, the position register is zeroed out and the output begins to slow down at Decel Rate.
- 4: Deceleration ends at Creep Freq and the move continues towards the Stop Trigger, a position relative to the Decel Trigger. The Stop Trigger is the position -2000 in this example.
- 5: When the Stop Trigger is reached, the output stops.

Example: Trapezoid with Limits using a position for the Stop Trigger. The Stop Trigger position requires change of direction.



- 1: Move starts at the Minimum Freq and accelerates at Accel Rate.
- 2: Acceleration ends at Maximum Freq and move continues at that rate.
- 3: When the Decel Trigger is reached, the position register is zeroed out and the output begins to slow down at Decel Rate.
- 4: Deceleration ends at Minimum Freq and the move changes direction to move back towards the Stop Trigger, a position relative to the Decel Trigger.
- 5: When the Stop Trigger is reached (position 400 in this example), the output stops.

Pulse Output Status/Control Bits and Command Codes (DL PLCs)



Output Status Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Status Bit CTRIO(2) to CPU	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Input Start (octal)
Output Enabled	32, 48	22.0, 23.0
Position Loaded	33, 49	22.1, 23.1
Output Suspended	34, 50	22.2, 23.2
Output Active	36, 52	22.4, 23.4
Output Stalled	37, 53	22.5, 23.5
Command Error	38, 54	22.6, 23.6
Command Complete	39, 55	22.7, 23.7

Output Control Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Control Bit CPU to CTRIO(2)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)	Read as:
Enable Output	32, 48	26.0, 27.0	Level
Go to Position	33, 49	26.1, 27.1	Rising Edge
Suspend Output	34, 50	26.2, 27.2	Level
Direction	36, 52	26.4, 27.4	Level
Process Command	39, 55	26.7, 27.7	Rising Edge

Output Control (D)Words (Pulse Output)

Word Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
Command Code	0, 6	10, 16
Word Parameter 1	1, 7	11, 17
Word Parameter 2	2, 8	12, 20

DWord Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
DWord Parameter 3	0, 2	0, 4

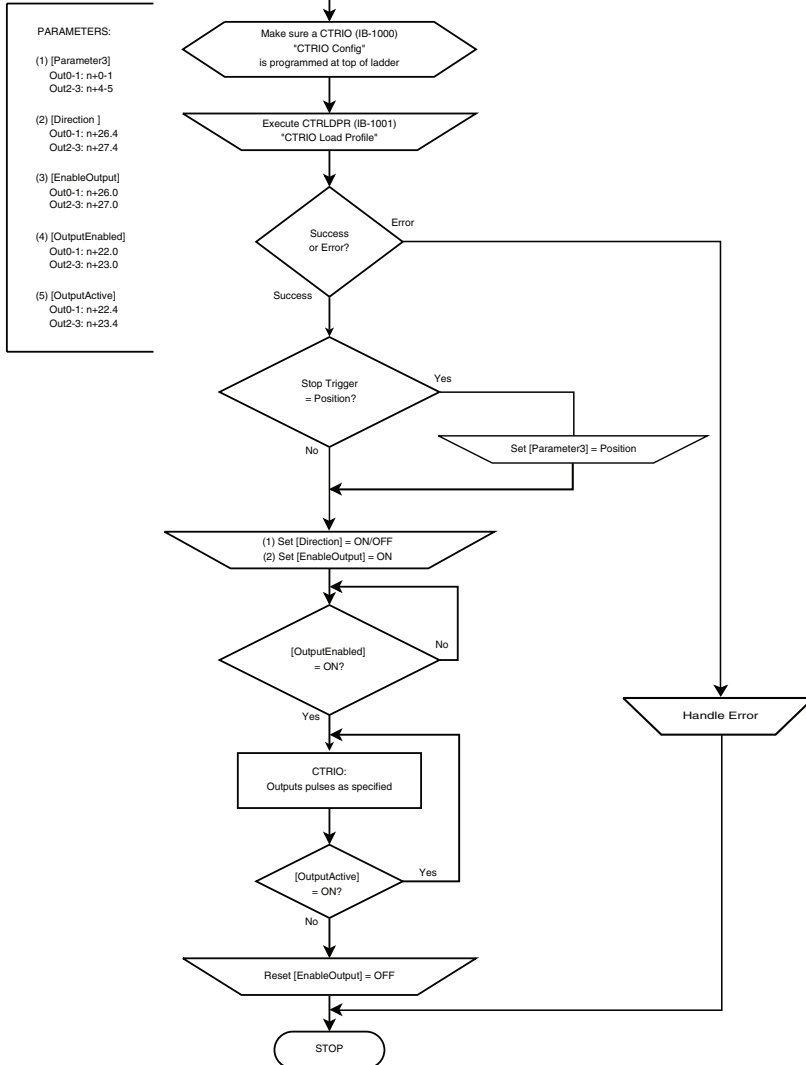
Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Load and Run a Trapezoid with Limits Profile

Example: You will need to have a Trapezoid Plus or Trapezoid with Limits profile configured using the Configure I/O dialog. You will also need to have the appropriate Pulse Profile Table File Number (decimal) stored in V3000 for this example. You must turn C0 on to load and run the pulse profile.

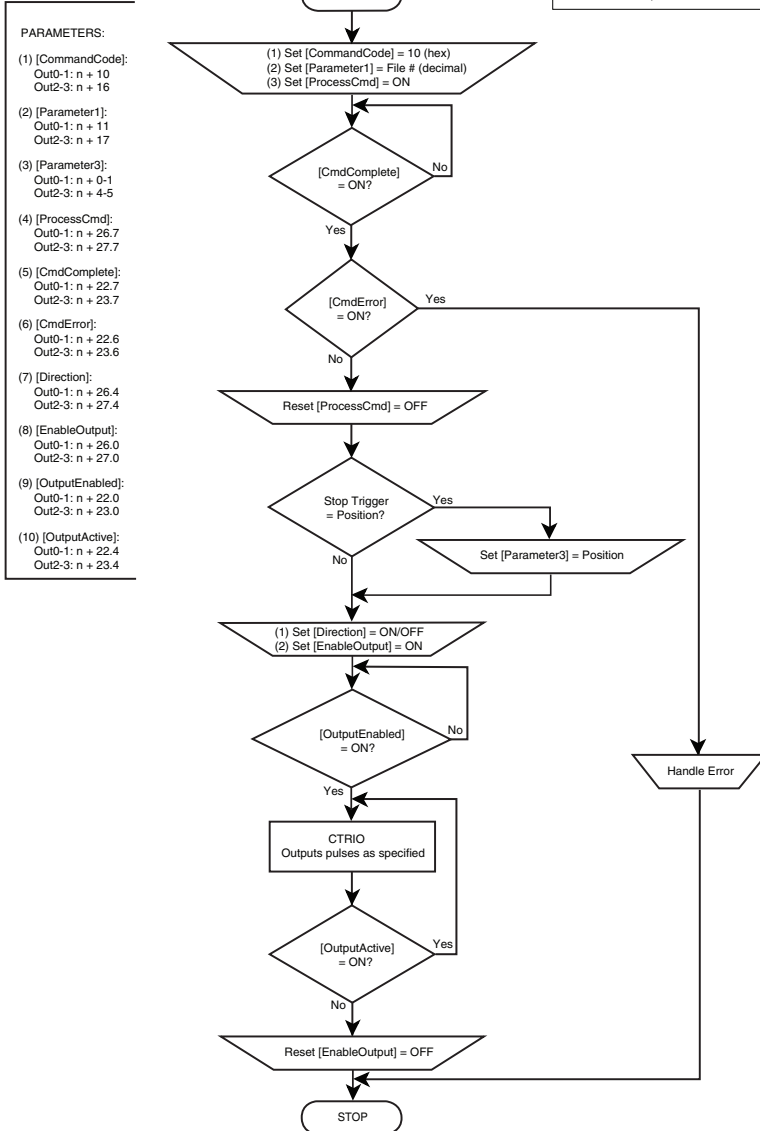
Run Trapezoid With Limits (IBoxes)

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



Run Trapezoid With Limits
(DL-PLC)

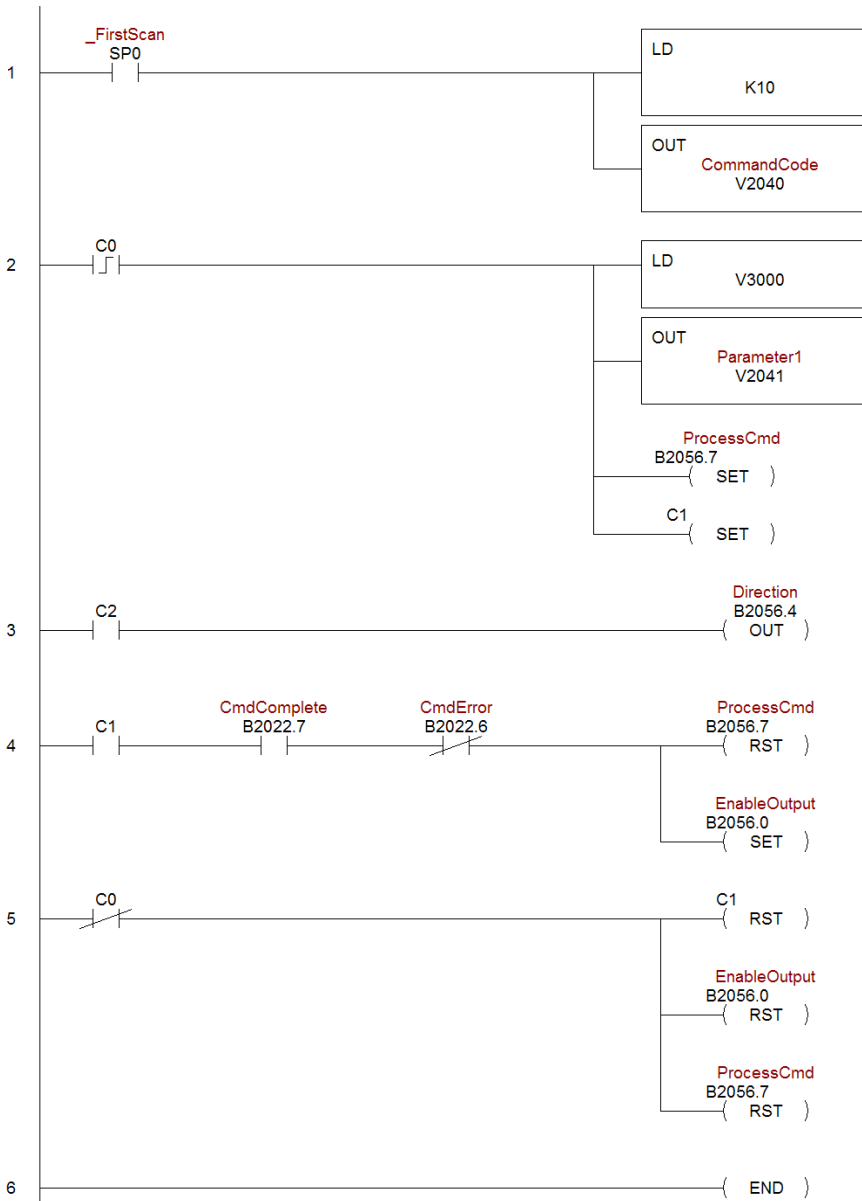
NOTE: An example program using these IBoxes can be found in the Appendices of the DirectLOGIC user manuals for DL05, DL06 and DL205.



Chapter 9: Output Functions

- DL
- Win
- NI

The following example program loads and executes a Trapezoid with Limits Profile that was created using CTRIO Workbench Pulse Profiles dialog. These profiles can only be used with a CTRIO2 module. The Pulse Profile number is stored in V3000. Turning on C0 will load the pulse profile and target position, and run the profile.



Trapezoid Plus (CTRIO2) Profile

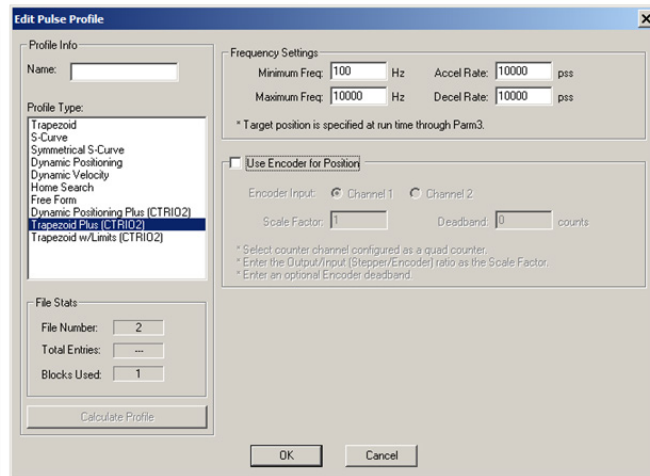
Trapezoid Plus

Trapezoid Plus is used for executing trapezoidal moves to a position value. The desired position is specified in Parameter 3. The sign of the value of the Target Position (Parameter 3) determines the direction of the pulse train output. This profile allows for different acceleration and deceleration rates. Also, Trapezoid Plus will accept a new target position to be entered while the output is in motion. The proper sequence of events for using this profile can be found in the previous flow charts and those that follow.



The Trapezoid Plus (CTRIO2) profile is only available when using a CTRIO2 module. See Trapezoid description for a general description of this profile. The profile resembles Trapezoid Profile, but has four additional features:

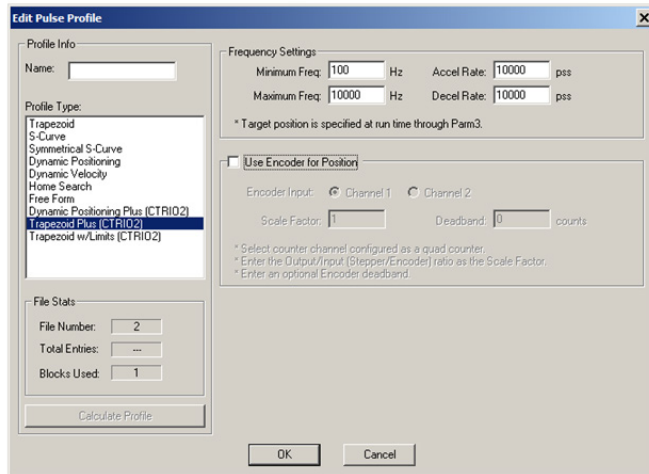
- 1: The target position is a variable (Parameter 3 in the output memory map) set by the base controller, instead of a constant specified in the profile.
- 2: The profile can be asymmetrical (Accel and Decel rates are separate).
- 3: Encoder Feedback can be added, useful for correcting excessive lash or slippage in the system.



- 4: Allows output rates up to 250kHz.

Trapezoid Plus, cont'd

When **Use Encoder for Position** is enabled, the target position is specified in units of the encoder. Encoder feedback determines when deceleration of a move should begin and determines when the move should stop. If **Scale Factor** is set incorrectly, the output could overshoot the target position, or start decelerating too soon.



- **Minimum Freq:** The frequency at which the profile will begin and end.
- **Maximum Freq:** The maximum steady state frequency the profile can attain during a move.
- **Accel Rate:** The rate at which the output frequency will increase during the acceleration period.
- **Decel Rate:** The rate at which the output frequency will decrease during the deceleration period.
- **Encoder Input:** Select the channel where the encoder is connected. The inputs for the encoder (A&B) must be configured for Quad Counter.
- **Scale Factor:** This is the output to input resolution (stepper/encoder) ratio. In other words, if the stepper motor being used is a 1000 ppr (pulses per revolution) and the encoder is 800 ppr, then the scale factor would be $1000/800 = 1.25$.
- **Deadband:** This is the number of position counts away from the target position that causes no action. This can reduce “hunting” or “ringing” as the profile attempts to get to the target position. Be sure to include a deadband when the encoder has a higher resolution than the stepper.

Pulse Output Status/Control Bits and Command Codes (DL PLCs)

Output Status Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

- DL
- Win
- NI

Status Bit CTRIO(2) to CPU	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Input Start (octal)
Output Enabled	32, 48	22.0, 23.0
Position Loaded	33, 49	22.1, 23.1
Output Suspended	34, 50	22.2, 23.2
Output Active	36, 52	22.4, 23.4
Output Stalled	37, 53	22.5, 23.5
Command Error	38, 54	22.6, 23.6
Command Complete	39, 55	22.7, 23.7

Output Control Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Control Bit CPU to CTRIO(2)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)	Read as:
Enable Output	32, 48	26.0, 27.0	Level
Go to Position	33, 49	26.1, 27.1	Rising Edge
Suspend Output	34, 50	26.2, 27.2	Level
Direction*	36, 52	26.4, 27.4	Level
Process Command	39, 55	26.7, 27.7	Rising Edge

* Not used in ladder. Controlled by the sign of the Target Position (Parameter 3) value.

Output Control (D)Words (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Word Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
Command Code	0, 6	10, 16
Word Parameter 1	1, 7	11, 17
Word Parameter 2	2, 8	12, 20

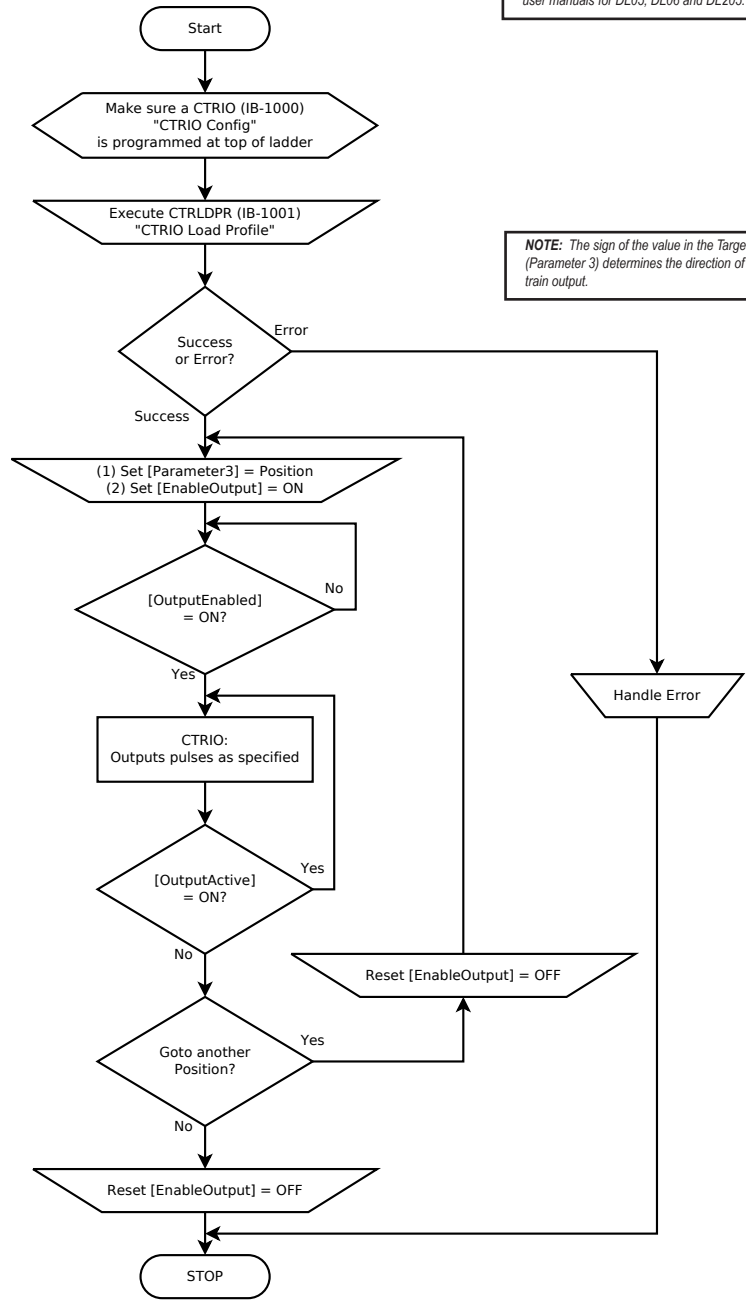
DWord Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
DWord Parameter 3	0, 2	0, 4

Run Trapezoid PLUS (IBoxes)

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

- PARAMETERS:**
- (1) [Parameter3]
Out0-1: n+0-1
Out2-3: n+4-5
 - (2) [EnableOutput]
Out0-1: n+26.0
Out2-3: n+27.0
 - (3) [OutputEnabled]
Out0-1: n+22.0
Out2-3: n+23.0
 - (4) [OutputActive]
Out0-1: n+22.4
Out2-3: n+23.4

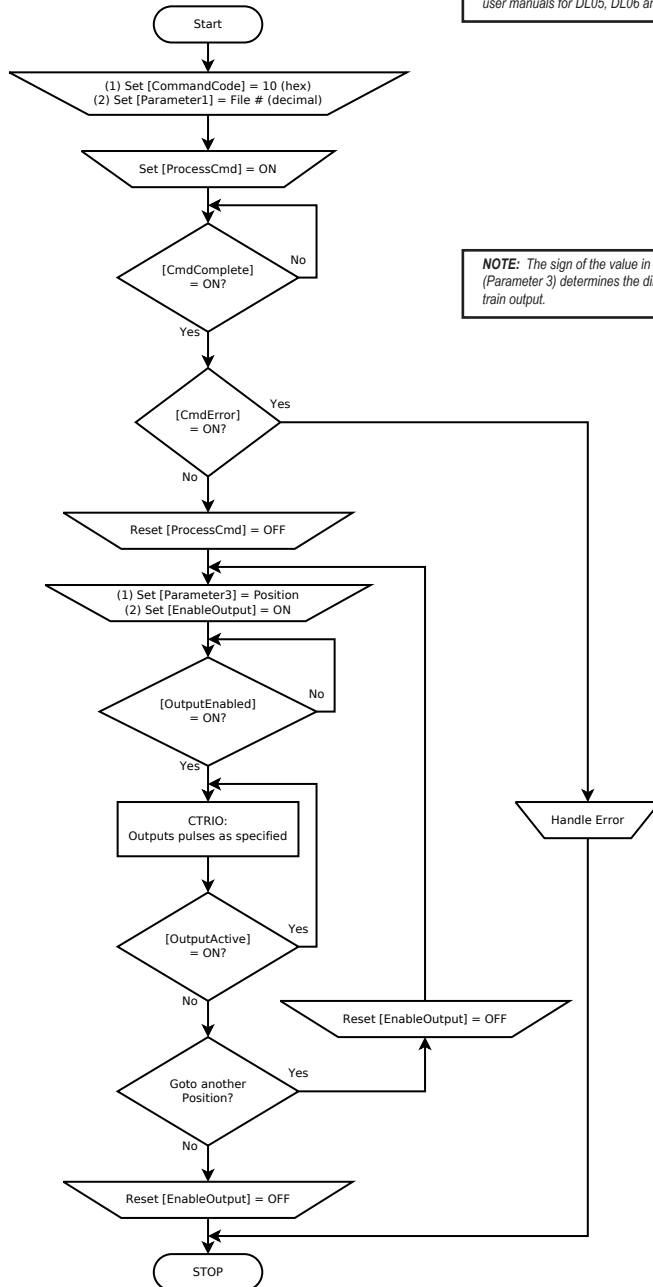
NOTE: The sign of the value in the Target Position (Parameter 3) determines the direction of the pulse train output.



**Run Trapezoid PLUS
(DL-PLC)**

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

- PARAMETERS:**
- (1) [CommandCode]:
Out0-1: n+10
Out2-3: n+16
 - (2) [Parameter1]
Out0-1: n+11
Out2-3: n+17
 - (3) [Parameter3]
Out0-1: n+0-1
Out2-3: n+4-5
 - (4) [ProcessCmd]
Out0-1: n+26.7
Out2-3: n+27.7
 - (5) [CmdComplete]
Out0-1: n+22.7
Out2-3: n+23.7
 - (6) [CmdError]
Out0-1: n+22.6
Out2-3: n+23.6
 - (7) [EnableOutput]
Out0-1: n+26.0
Out2-3: n+27.0
 - (8) [OutputEnabled]
Out0-1: n+22.0
Out2-3: n+23.0
 - (9) [OutputActive]
Out0-1: n+22.4
Out2-3: n+23.4

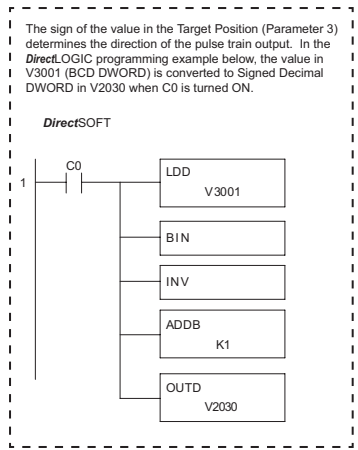
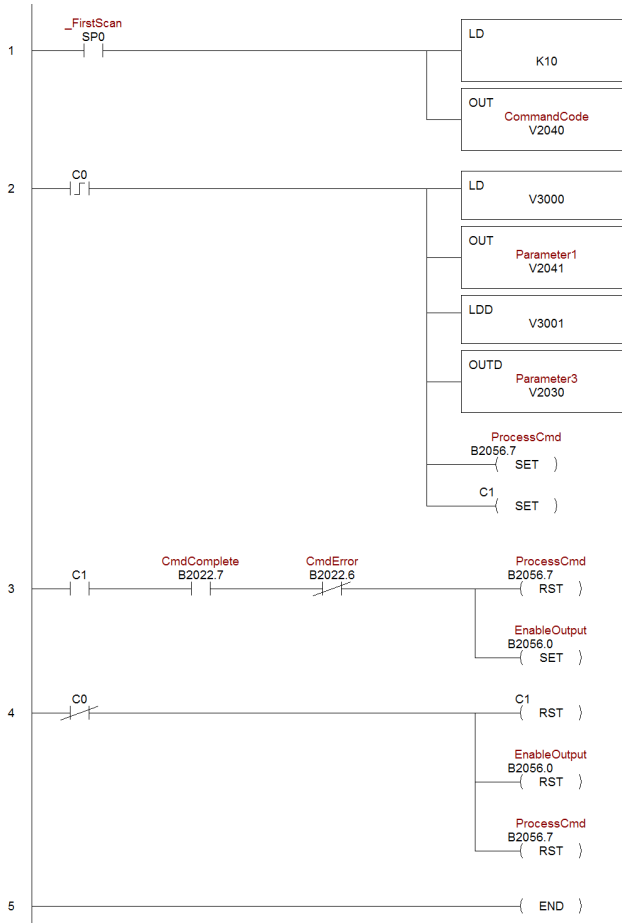


NOTE: The sign of the value in the Target Position (Parameter 3) determines the direction of the pulse train output.

Load and Run a Trapezoid Plus Profile

The following example program loads and executes a Trapezoid Plus or Trapezoid with Limits Profile that was created using CTRIO Workbench Pulse Profiles dialog. These profiles can only be used with a CTRIO2 module. The Pulse Profile number is stored in V3000 and the Target Position is stored in V3001-V3002. Turning on C0 will load the pulse profile and target position, and run the profile.

- DL
- Win
- NI



Dynamic Positioning Plus and Dynamic Positioning Profiles

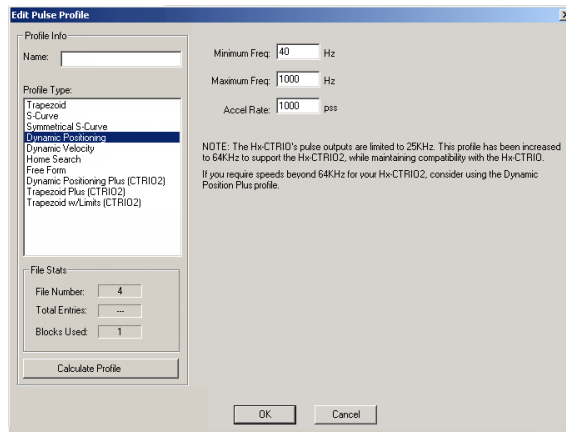
- DL
- Win
- NI

Dynamic Positioning Plus and Dynamic Positioning profiles are very similar and are covered together in this section. The table below summarizes their differences.

Options	Position Source	Change target position on the fly	Accel/decel ramps	Maximum pulse rate	Encoder Feedback Possible	Requires CTRIO2
Dynamic Positioning Plus	From controller	Yes	Specify separately	250kHz	Yes	Yes
Dynamic Positioning	From controller	No	One setting for both	65kHz ¹	No	No

¹The older CTRIO outputs can only achieve 25kHz maximum. CTRIO2 outputs are limited to 65kHz when using these profiles.

The Dynamic Positioning profile is a trapezoidal profile with identical acceleration/deceleration rates and identical starting/stopping frequencies. The maximum target frequency is specified. The target position (# of output pulses) is located in a memory register in the CPU/controller. Once the position is reached, the output is disabled and a new target position can be specified in the memory register.



- **Minimum Freq:** The frequency at which the profile will begin and stop.
- **Maximum Freq:** The maximum steady state frequency the profile can attain during a move.
- **Accel Rate:** The rate at which the output will ramp from the Minimum Frequency to the Maximum Frequency. This sets the deceleration rate as well.

Dynamic Positioning Plus (CTRIO2) Profile

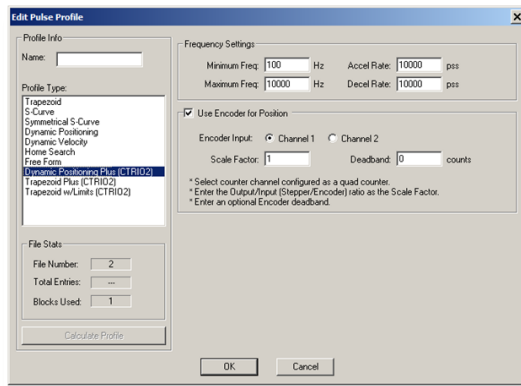
- DL
- Win
- NI

The Dynamic Positioning Plus (CTRIO2) profile is only available when using a CTRIO2 module. The profile resembles a Dynamic Positioning Profile, but adds three features:

- 1: The profile can be asymmetrical (Accel and Decel Rates are separate).
- 2: Encoder Feedback can be added, useful for correcting excessive lash or slippage in the system.
- 3: Allows output rates up to 250kHz.

When **Use Encoder for Position** is enabled, the target position is specified in units of the encoder. Encoder feedback determines when deceleration of a move should begin and determines when the move should stop.

See Dynamic Positioning Profile for a general description of this profile.



- **Minimum Freq:** The frequency at which the profile will begin and end.
- **Maximum Freq:** The maximum steady state frequency the profile can attain during a move.
- **Accel Rate:** The rate at which the output frequency will increase during the acceleration period.
- **Decel Rate:** The rate at which the output frequency will decrease during the deceleration period.
- **Encoder Input:** Select the channel where the encoder is connected.
- **Scale Factor:** This is the output to input resolution (stepper/encoder) ratio. In other words, if the stepper motor being used is a 1000 ppr (pulses per revolution) and the encoder is 800 ppr, then the scale factor would be $1000/800 = 1.25$.
- **Deadband:** This is the number of position counts away from the target position that causes no action. This can reduce “hunting” or “ringing” as the profile attempts to get to the target position. Be sure to include a deadband when the encoder has a higher resolution than the stepper.

Pulse Output Status/Control Bits and Command Codes (DL PLCs)

- DL
- Win
- NI

Output Status Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Status Bit CTRIO(2) to CPU	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Input Start (octal)
Output Enabled	32, 48	22.0, 23.0
Position Loaded	33, 49	22.1, 23.1
Output Suspended	34, 50	22.2, 23.2
Output Active	36, 52	22.4, 23.4
Output Stalled	37, 53	22.5, 23.5
Command Error	38, 54	22.6, 23.6
Command Complete	39, 55	22.7, 23.7

Output Control Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Control Bit CPU to CTRIO(2)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)	Read as:
Enable Output	32, 48	26.0, 27.0	Level
Go to Position	33, 49	26.1, 27.1	Rising Edge
Suspend Output	34, 50	26.2, 27.2	Level
Direction	36, 52	26.4, 27.4	Level
Process Command	39, 55	26.7, 27.7	Rising Edge

Output Control (D)Words (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Word Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
Command Code	0, 6	10, 16
Word Parameter 1	1, 7	11, 17
Word Parameter 2	2, 8	12, 20

DWord Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
DWord Parameter 3	0, 2	0, 4

Dynamic Positioning/Dynamic Positioning Plus, (cont'd)

- DL
- Win
- NI

For Dynamic Positioning/Positioning Plus only, the motion limits of Min Frequency, Max Frequency, and Acceleration rate come from the CTRIO Workbench Profile. After loading a Dynamic Positioning/Positioning Plus Profile, setting the **Enable Output** causes the CTRIO(2) module to assume a position of 0 pulses. The program should write the target position in DWord Parameter 3, and set the **Go to Position** bit. This will cause the CTRIO(2) module to set both Pulses Active and the **New Position Loaded** bit, then begin to output pulses. The number of pulses and direction are determined by the CTRIO(2) module based on the difference between the current location and the specified target location.

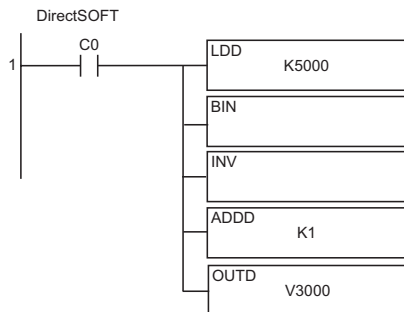
The program can monitor the state of the **Pulses Active** bit and the **New Position Loaded** bit to

Position Loaded Status Bit V40622.1 or C441	Pulses Active Status Bit V40622.0 or C440	CTRIO(2) Pulse Output State
0	0	Idle
1	1	Go To Position Acknowledged, Pulsing
0	1	Still Pulsing, Go To Position Control Bit is OFF
1	0	Go To Position Acknowledged, Position Attained

determine when the new position has been attained. The **New Position Loaded** status bit will always follow the state of the Load/Seek New Position control bit. This status bit should be used to signal the program that the CTRIO(2) module has received the new state of the control bit. You do not have to wait on the CTRIO(2) module to complete a move that is in progress before loading the next target location. After the GoTo Position is acknowledged, the program can load the next position into the DWord Parameter 3. When **Pulses Active Status** goes to 0, then setting the **GoTo Position** control bit will again start the output toward the new position. The CTRIO(2) module moves to the new position relative to its previous position as long as the **Enable Output** control bit remains set. Clearing the **Enable Output** bit will disable output pulsing and reset the current position to 0.

The following pages show a *Direct*LOGIC programming example that executes a Dynamic Positioning/Positioning Plus pulse profile using the bit/(D)word addressing in the table on page 9-39.

The sign of the value in the Target Position register (Parameter 3) determines the direction of the pulse train output. In the *Direct*LOGIC programming example to the right, BCD 5000 is converted to decimal -5000 when C0 is turned ON. You could load (LD) a V memory location instead of using a constant as shown in the example.



Dynamic Positioning or Dynamic Positioning Plus Using the CTRIO(2) Y0 and Y1

The required basic steps for a DL-PLC to run Dynamic Positioning or Dynamic Position Plus are outlined in this table. The steps describe the sequence and coordinate with the ladder code sequence on page 9-41.



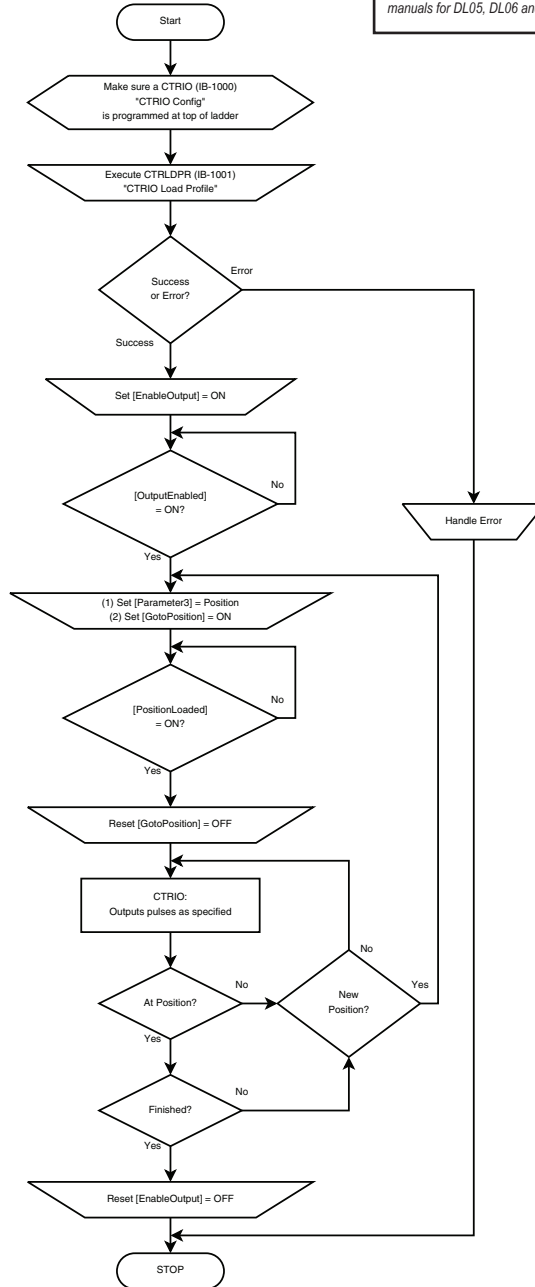
Steps	Name	PLC Control Outputs Base Addr = V2030 (Bit-of-Word)	PLC Status Inputs Base Addr = V2000 (Bit-of-Word)	PLC Control Outputs Base Addr = V2030 (Control Relay) D2-240	PLC Status Inputs Base Addr = V2000 (Control Relay) D2-240	Action
1	Command Code	V2040	N/A	V2040	N/A	Set to 10 (Load Stored Profile)
2	Parameter 1	V2041		V2041		File # of desired Dynamic Positioning Profile
3	Process Command	V2056.7		C227		Turn ON until Command Complete status bit is returned (see step 4)
4	Command Complete Status	N/A	V2022.7	N/A	C127	When ON, Profile is now loaded, clear Process Command bit (step 3)
5	Command Error		V2022.6			C126
6	Enable Output	V2056.0	N/A	C220	N/A	Turn ON to assume 0 position, Turn OFF to disable pulses and zero position
7	Output Enable Status	N/A	V2022.0	N/A	C127	When ON, pulses are now enabled and last position is retained
8	Parameter 3	V2031 / V2030	N/A	V2031 / V2030	N/A	Target position: User defined (DWord)
9	Go To Position	V2056.1		C221		Starts pulses with direction to obtain the new position relative to previous position.
10	Position Loaded Status	N/A	V2022.1	N/A	C121	When ON, Go To position is acknowledged
11	Output Active Status		V2022.4			C124
12	Go To Position	V2056.1	N/A	C221	N/A	Turn OFF to be ready to load a new position
13	Suspend Output	V2056.2		C222		Turn ON to "pause" output pulses without resetting pulse count
14	Output Suspend	N/A	V2022.2	N/A	C122	ON when out pulse train has been suspended

Repeat steps 7-10 to seek the next position.

Dynamic Position or Dynamic Position PLUS (IBoxes)

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

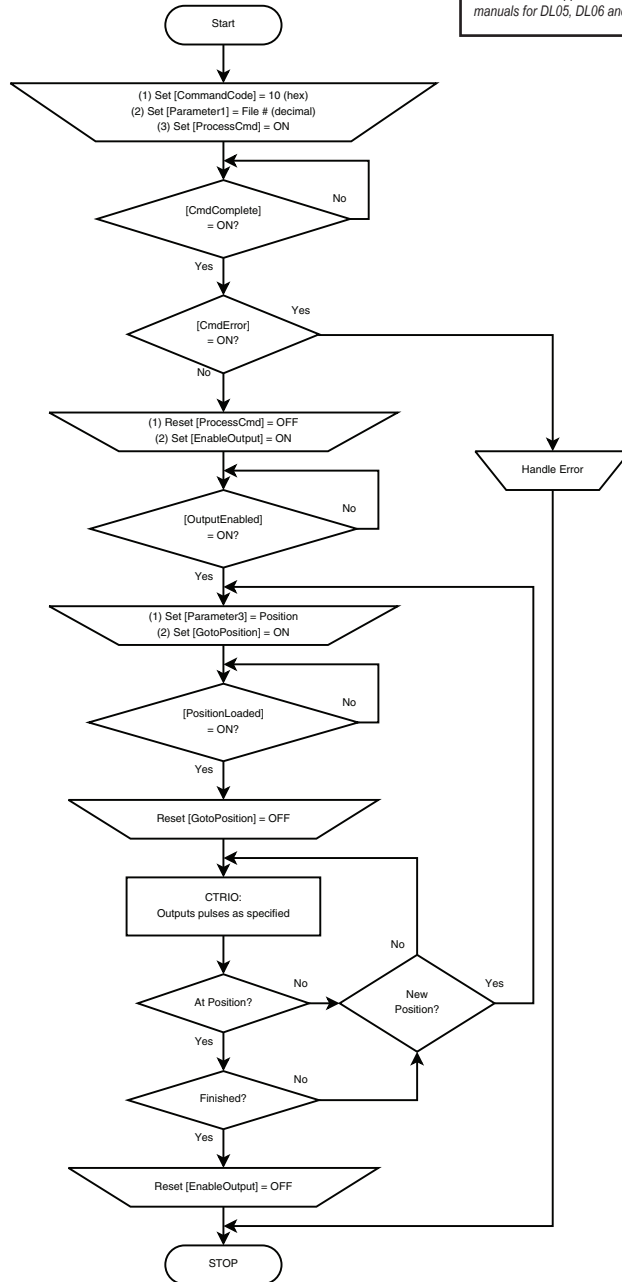
- PARAMETERS:**
- (1) [EnableOutput]
Out0-1: n+26.0
Out2-3: n+27.0
 - (2) [OutputEnabled]
Out0-1: n+22.0
Out2-3: n+23.0
 - (3) [Parameter3]
Out0-1: n+0.1
Out2-3: n+4.5
 - (4) [GotoPosition]
Out0-1: n+26.1
Out2-3: n+27.1
 - (5) [PositionLoaded]
Out0-1: n+22.1
Out2-3: n+23.1



**Dynamic Position or Dynamic Position PLUS
(DL-PLC)**

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

- PARAMETERS:**
- (1) [CommandCode]
Out0-1: n+10
Out2-3: n+16
 - (2) [Parameter1]
Out0-1: n+11
Out2-3: n+17
 - (3) [ProcessCmd]
Out0-1: n+26.7
Out2-3: n+27.7
 - (4) [CmdComplete]
Out0-1: n+22.7
Out2-3: n+23.7
 - (5) [CmdError]
Out0-1: n+22.6
Out2-3: n+23.6
 - (6) [EnableOutput]
Out0-1: n+26.0
Out2-3: n+27.0
 - (7) [OutputEnabled]
Out0-1: n+22.0
Out2-3: n+23.0
 - (8) [Parameter3]
Out0-1: n+0-1
Out2-3: n+4-5
 - (9) [GotoPosition]
Out0-1: n+26.1
Out2-3: n+27.1
 - (10) [PositionLoaded]
Out0-1: n+22.1
Out2-3: n+23.1

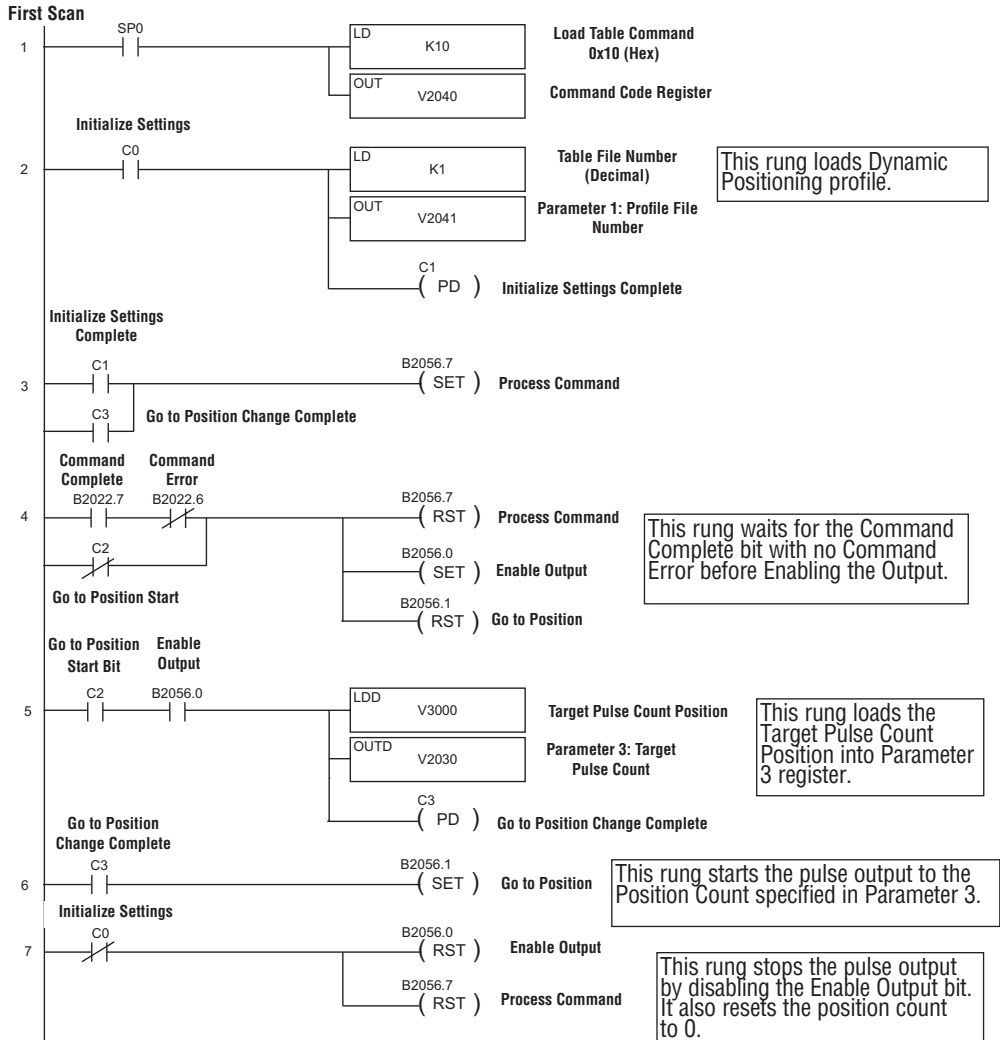


Dynamic Positioning/Positioning Plus Example

- DL
- Win
- NI

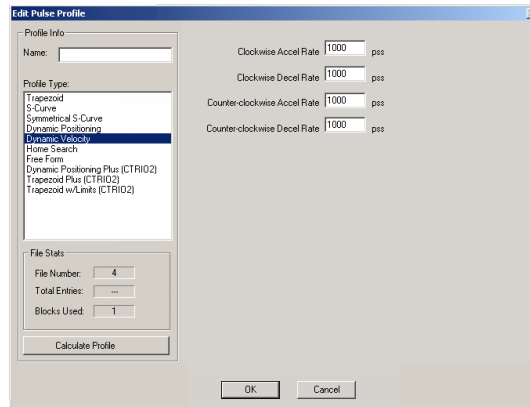
The following example program runs a Dynamic Positioning/Positioning Plus pulse profile. Turn on C0 to load the profile number and process the command. The first move starts at position count = 0. Turning on C2 will start the pulse output to position specified in Parameter 3.

The output will move in whichever direction is appropriate to reach the position specified in Parameter 3. To make additional moves, wait for the current move to complete, load a new value into the Target Pulse Count register and set the Go to Position bit. Subsequent moves are still referenced to the same 0 location as the first move. Clearing the Enable Output bit will disable output pulsing and reset the current position to 0.



Dynamic Velocity Profile

The Dynamic Velocity profile is a trapezoidal profile with the directional acceleration and deceleration rates specified. The target velocity is located in DL, Win, and NI.



- **Clockwise Accel Rate:** The clockwise rate at which the output will ramp up from 0Hz to the target velocity that is specified in the CPU/controller memory register.
- **Clockwise Decel Rate:** The clockwise rate at which the output will ramp down from the target velocity that is specified in the CPU/controller memory register to 0Hz.
- **Counter-Clockwise Accel Rate:** The counter-clockwise rate at which the output will ramp up from 0Hz to the target velocity that is specified in the CPU/controller memory register.
- **Counter-Clockwise Decel Rate:** The counter-clockwise rate at which the output will ramp down from the target velocity that is specified in the CPU/controller memory register to 0Hz.

Dynamic Velocity Profile

For the **Example** that follows you will need to have a Dynamic Velocity profile configured as Table File Number 1 using the Configure I/O dialog box. You will also need to have the appropriate Target Velocity (signed decimal) stored in V3000 for this example. You must turn C0 on to initialize the settings and enable the output.

Pulse Output Status/Control Bits and Command Codes (DL PLCs)

Output Status Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

- DL
- Win
- NI

Status Bit CTRIO(2) to CPU	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Input Start (octal)
Output Enabled	32, 48	22.0, 23.0
Position Loaded	33, 49	22.1, 23.1
Output Suspended	34, 50	22.2, 23.2
Output Active	36, 52	22.4, 23.4
Output Stalled	37, 53	22.5, 23.5
Command Error	38, 54	22.6, 23.6
Command Complete	39, 55	22.7, 23.7

Output Control Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Output Control (D)Words (Pulse Output)

Control Bit CPU to CTRIO(2)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)	Read as:
Enable Output	32, 48	26.0, 27.0	Level
Go to Position	33, 49	26.1, 27.1	Rising Edge
Suspend Output	34, 50	26.2, 27.2	Level
Direction	36, 52	26.4, 27.4	Level
Process Command	39, 55	26.7, 27.7	Rising Edge

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Word Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
Command Code	0, 6	10, 16
Word Parameter 1	1, 7	11, 17
Word Parameter 2	2, 8	12, 20

DWord Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
DWord Parameter 3	0, 2	0, 4

Dynamic Velocity (cont'd)

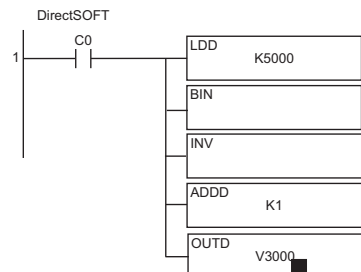
- ☑ DL
- ☑ Win
- ☑ NI

For Dynamic Velocity, the motion limits of clockwise acceleration and deceleration, and counter clockwise acceleration and deceleration come from the CTRIO Workbench Profile. The target velocity is stored in a register in the CPU/controller. The program needs to prepare the Load Table command by selecting Command Code = 0010 Hex/BCD, set Word Parameter 1 to the File number of the profile (example: File 3 Dynamic Velocity 1) and set Word Parameter 3 to the desired target velocity. Then the program can set the Process Command bit and watch for the Command Complete bit. Then the program should clear the Process Command bit. Set the Enable Output bit to start the output pulses. The velocity can be changed “on the fly” by entering a different value into the target velocity register and setting the Process Command bit. The velocity will ramp up/down to the new target velocity at the specified Accel/Decel rates. Clearing the Enable Output bit will always suspend pulsing. The following pages show a programming example that executes a Dynamic Velocity pulse profile using the bit/(D)word addressing in the table below.

Dynamic Velocity using the CTRIO(2) Y0 and Y1

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	V2040	N/A	V2040	N/A	Set to 10 (Load Stored Profile)
2	Parameter 1	V2041		V2041		File # containing CW accel/decel and CCW accel/decel
3	Process Command	V2056.7		C227		Turn ON until Command Complete status bit is returned
4	Command Complete Status	N/A	V2022.7	N/A	C127	When ON, Profile is now loaded, clear Process Command bit
5	Command Error		V2022.6			C126
6	Enable Output	V2056.0	N/A	C220	N/A	Turn ON to ramp to target velocity, Turn OFF to disable pulses.
7	Parameter 3	V2031 / V2030	N/A	V2031 / V2030		Target velocity: User defined (DWord)
8	Output Active Status	N/A	V2022.4	N/A	C124	When ON, module is pulsing
9	Suspend Output	V2056.2	N/A	C222	N/A	Turn ON to “pause” output pulses without resetting pulse count
10	Output Suspended	N/A	V2022.2	N/A	C122	ON when out pulse train has been suspended

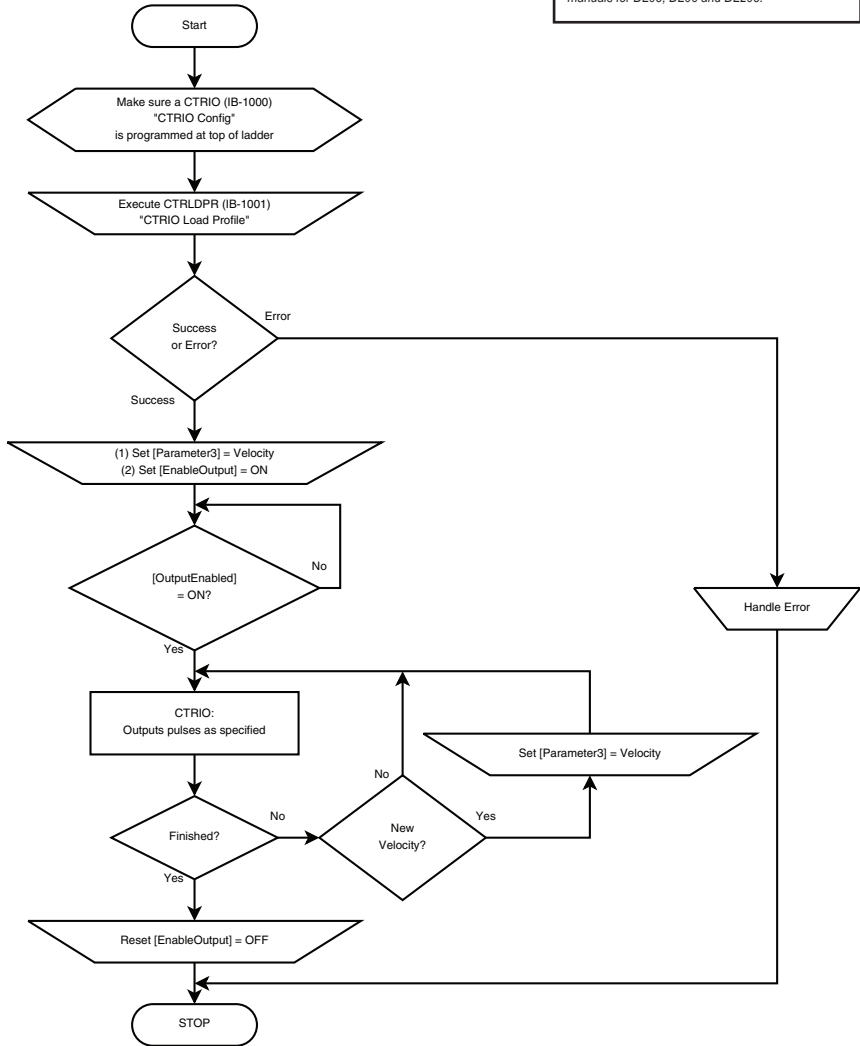
The sign of the value in the Target Velocity register (Parameter 3) determines the direction of the pulse train output. In the DirectLOGIC programming example to the right, BCD 5000 is converted to decimal -5000 when C0 is turned ON. You could load (LD) a V memory location instead of using a constant as shown in the example.



Dynamic Velocity (IBoxes)

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

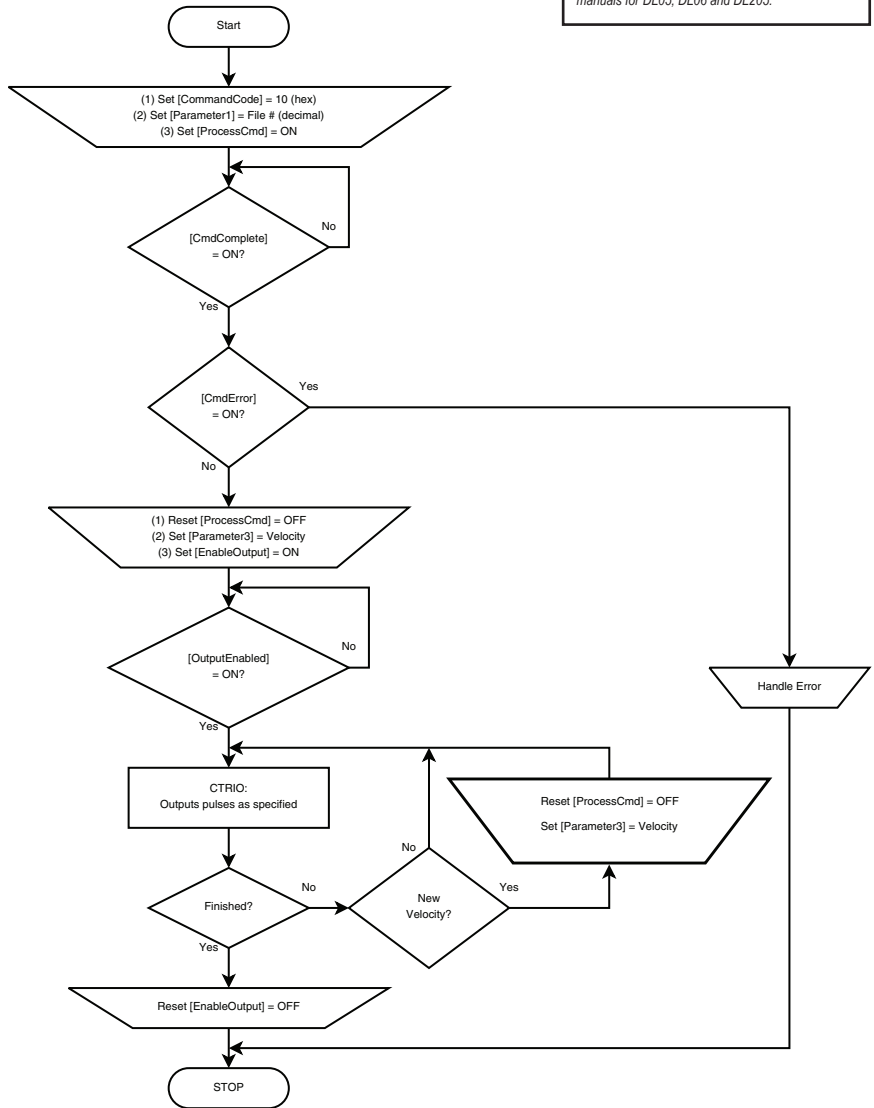
- PARAMETERS:
- (1) [Parameter3]
Out0-1: n+0-1
Out2-3: n+4-5
 - (2) [EnableOutput]
Out0-1: n+26.0
Out2-3: n+27.0
 - (3) [OutputEnabled]
Out0-1: n+22.0
Out2-3: n+23.0



**Dynamic Velocity
(DL-PLC)**

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

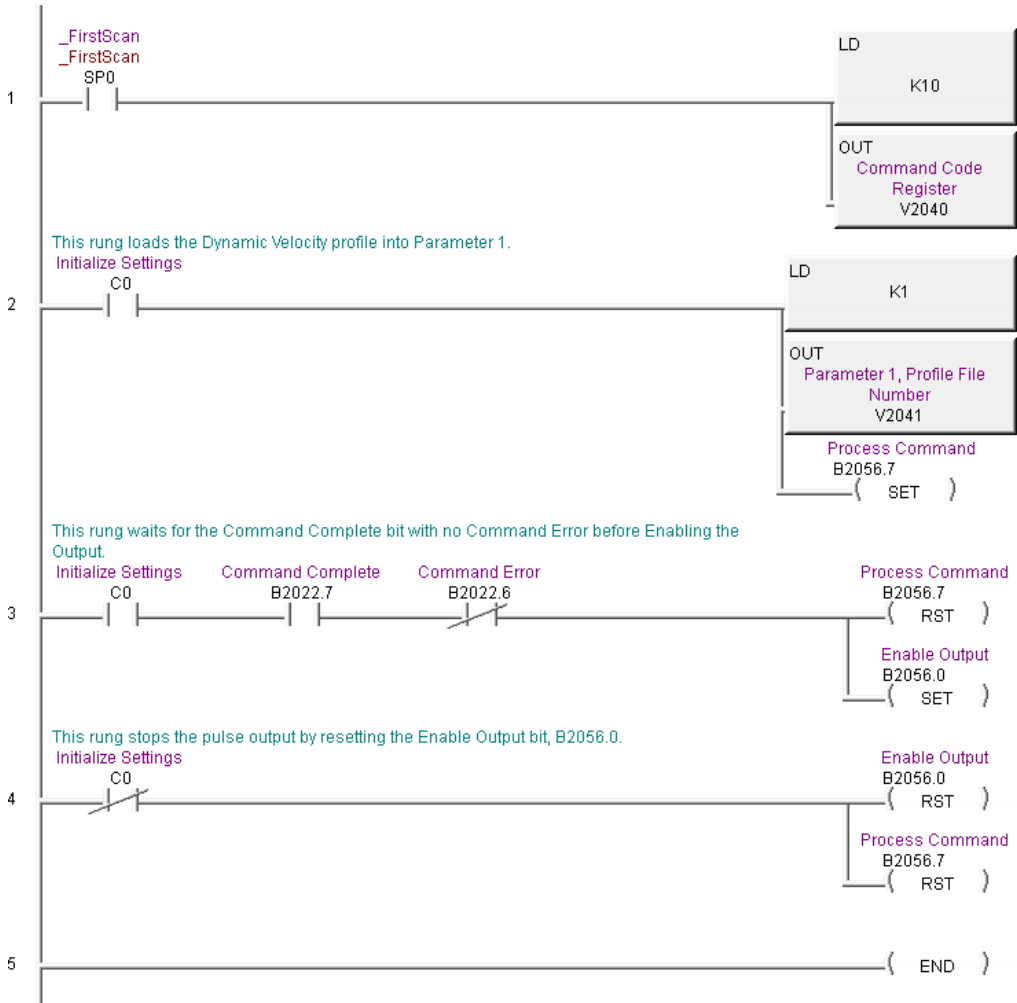
- PARAMETERS:**
- (1) [CommandCode]
Out0-1: n+10
Out2-3: n+16
 - (2) [Parameter1]
Out0-1: n+11
Out2-3: n+17
 - (3) [ProcessCmd]
Out0-1: n+26.7
Out2-3: n+27.7
 - (4) [CmdComplete]
Out0-1: n+22.7
Out2-3: n+23.7
 - (5) [CmdError]
Out0-1: n+22.6
Out2-3: n+23.6
 - (6) [Parameter3]
Out0-1: n+0-1
Out2-3: n+4-5
 - (7) [EnableOutput]
Out0-1: n+26.0
Out2-3: n+27.0
 - (8) [OutputEnabled]
Out0-1: n+22.0
Out2-3: n+23.0



Dynamic Velocity Example

- DL
- Win
- NI

The following example program executes a Dynamic Velocity pulse profile. Turn C0 on to load the profile number, process the command and enable the output. The Target Velocity needs to be specified in V2030. The velocity can be changed “on the fly” by entering a different value into V2030. The sign of the value in the target velocity register controls the pulse output direction. Clearing the Output Enable bit will always suspend pulsing.



Velocity Mode



Velocity Mode is used to set the output to a frequency and duty cycle.

NOTE: *Velocity Mode controls the pulse outputs directly from the CPU/controller program. No CTRIO Workbench Pulse Profile is required for this mode.*

- DL
- Win
- NI

Velocity Mode is not defined or created using the Pulse Profiles Table dialog in CTRIO Workbench. However, the output must be configured for Pulse (Step/Direction) or Pulse (CW/CCW) using the Configure IO dialog. The profile parameters are specified in CPU/controller memory registers. There are no acceleration ramps with Velocity Mode. The output will step directly to the specified frequency and to the specified duty cycle.

The user specifies the target frequency, pulse train duty cycle and the step count. Once initiated, the output will begin pulsing at the target frequency and continue until the step count is reached. There is no ramping to the target frequency. With a step count of 0xFFFFFFFF, the pulse output will continue indefinitely until the control program disables the output.

Velocity Mode command (Command = 0020 Hex/BCD) allows a specified number of pulse output counts or the number of Pulses can be set to "FFFFFFFF" in Hex for unlimited pulse counts. Leaving the **Duty Cycle** set to 0 achieves the default (50%), otherwise it can be set in 1% increments by writing this value from 1 to 99 decimal. After this command is processed, the **Run Frequency** and **Duty Cycle** fields can be adjusted by direct access. In order to change directions from **Pulse Output** in "Velocity" mode, the **Enable Output** bit must first be cleared (which stops the Pulse Outputs). Then after the new **Direction** bit is written, the **Enable Output** bit can be set to resume pulsing. Refer to the following table for a *Direct*LOGIC programming example that executes a Velocity Mode pulse profile using bit/(D)word addressing.

Pulse Output Status/Control Bits and Command Codes (DL PLCs)

Output Status Bit Definitions (Pulse Output)

- DL
- Win
- NI

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Status Bit CTRIO(2) to CPU	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Input Start (octal)
Output Enabled	32, 48	22.0, 23.0
Position Loaded	33, 49	22.1, 23.1
Output Suspended	34, 50	22.2, 23.2
Output Active	36, 52	22.4, 23.4
Output Stalled	37, 53	22.5, 23.5
Command Error	38, 54	22.6, 23.6
Command Complete	39, 55	22.7, 23.7

Output Control Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Control Bit CPU to CTRIO(2)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)	Read as:
Enable Output	32, 48	26.0, 27.0	Level
Go to Position	33, 49	26.1, 27.1	Rising Edge
Suspend Output	34, 50	26.2, 27.2	Level
Direction	36, 52	26.4, 27.4	Level
Process Command	39, 55	26.7, 27.7	Rising Edge

Output Control (D)Words (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Word Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
Command Code	0, 6	10, 16
Word Parameter 1	1, 7	11, 17
Word Parameter 2	2, 8	12, 20

DWord Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
DWord Parameter 3	0, 2	0, 4

Velocity Mode Control on CTRIO(2) Y0 and Y1

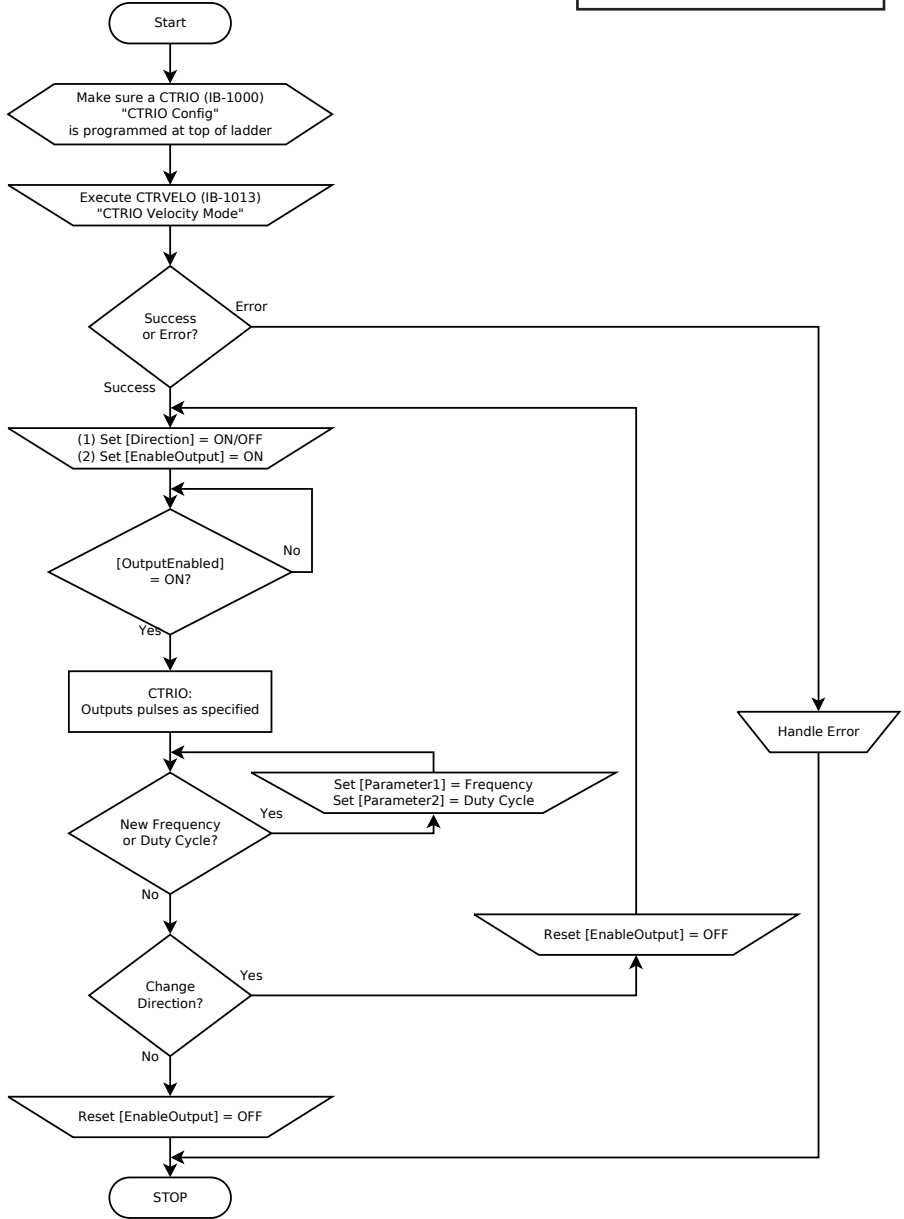
Steps	Name	PLC Control Outputs Base Addr = V2030 (Bit-of-Word)	PLC Status Inputs Base Addr = V2000 (Bit-of-Word)	PLC Control Outputs Base Addr = V2030 (Control Relay) D2-240	PLC Status Inputs Base Addr = V2000 (Control Relay) D2-240	Action
1	Command Code	V2040	N/A	V2040	N/A	Set to 20 Hex (Pulse at Velocity)
2	Parameter 1	V2041		V2041		Run frequency, decimal CTRIO: 20Hz-25000Hz CTRIO2: 20Hz-64000Hz
3	Parameter 2	V2042		V2042		Duty cycle (1-99) (can leave 0 for 50%) decimal
4	Parameter 3	V2031 / V2030		V2031 / V2030		Number of pulses (DWord); set to FFFFFFFF for no limit, Hex
5	Set Direction	V2056.4		C224		Set ON or OFF for Direction of Rotation
6	Process Command	V2056.7		C227		Turn ON Command Complete status bit is returned (see step 7)
7	Command Status	N/A	V2022.7	N/A	C127	When ON, command has been accepted, clear Process Command bit (step 6)
8	Command Error		V2022.6		C126	ON if Command or Parameters are invalid
9	Enable Output	V2056.0	N/A	C220	N/A	Turn ON to start pulses
10	Disable Output	V2056.0		C220		Turn OFF to start pulses
11	Suspend Output	V2056.2		C222		Turn ON to "pause" output pulses without resetting pulse count
12	Output Suspended	N/A	V2022.2	N/A	C122	ON when out pulse train has been suspended

While Velocity Mode Control is running, **Run Frequency** (step 2) and **Duty Cycle** (step 3) may be actively changed simply by writing a new Parameter value. Since no Accel/Decel parameters are specified in this profile, the output change is a step response.

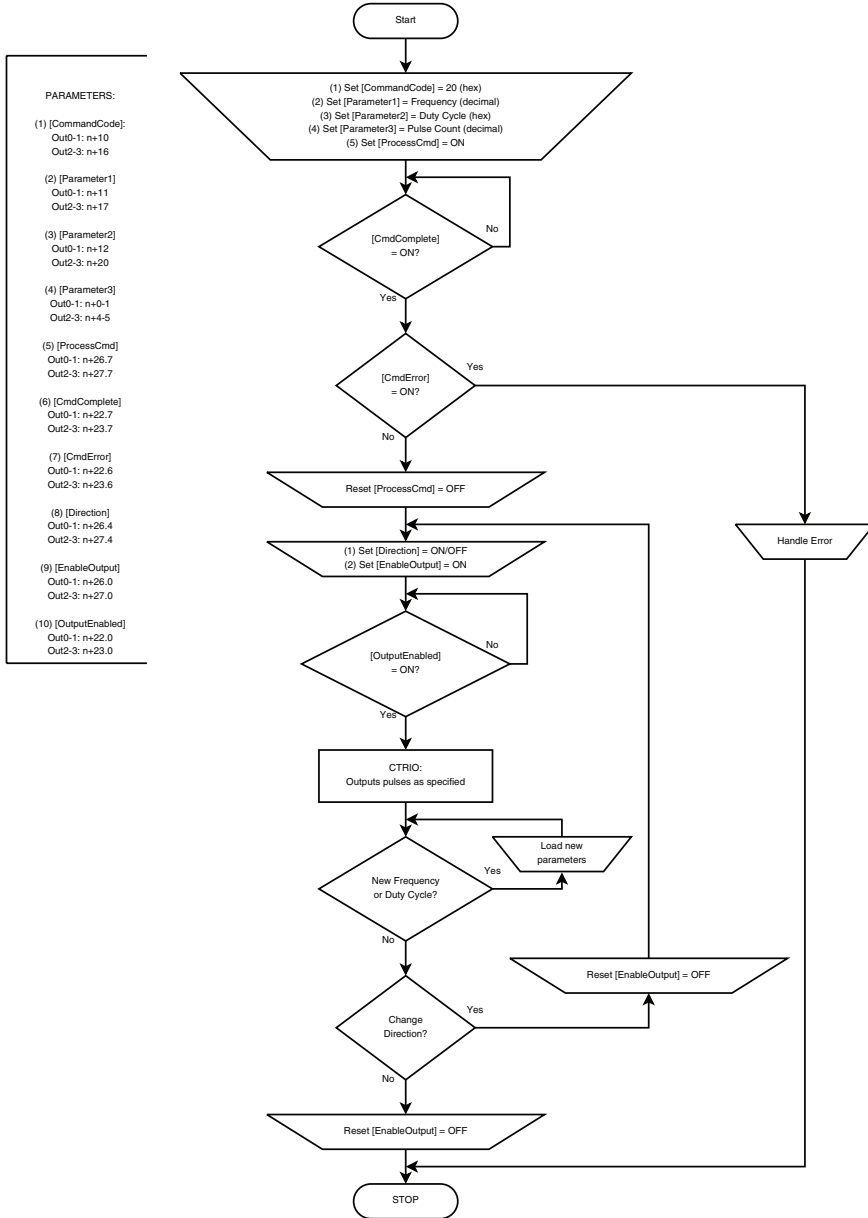
Velocity Mode (IBoxes)

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

- PARAMETERS:**
- (1) [Direction]
Out0-1: n+26.4
Out2-3: n+27.4
 - (2) [EnableOutput]
Out0-1: n+26.0
Out2-3: n+27.0
 - (3) [OutputEnabled]
Out0-1: n+22.0
Out2-3: n+23.0
 - (4) [Parameter1]
Out0-1: n+11
Out2-3: n+17
 - (5) [Parameter2]
Out0-1: n+12
Out2-3: n+20



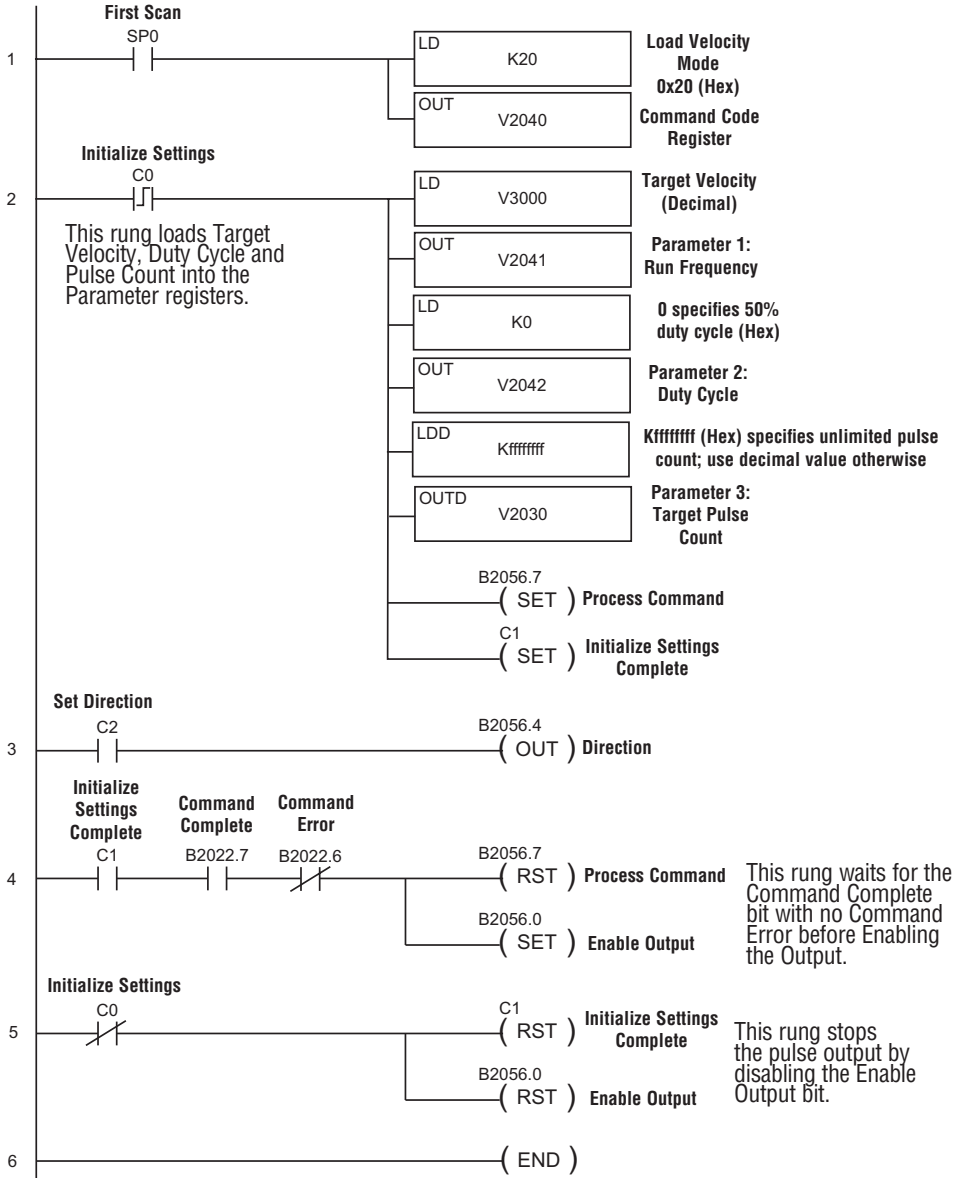
**Velocity Mode
(DL-PLC)**



Velocity Mode Example

- DL
- Win
- NI

The following example program loads and executes a Velocity Mode pulse profile. For Parameter 3, a specific number of pulse output counts can be specified or if set to “ffffff” Hex, the pulse output will remain ON at the specified Target Velocity until the output is disabled. Once running, velocity and duty cycle can be changed on the fly by loading new values into V2041 and V2042 respectively.



Run to Limit Mode

Run to Limit is used to make a simple move to a limit input on the CTRIO(2) module. If using a CTRIO2 module, consider using Trapezoid with Limits instead.

- DL
- Win
- NI

Run to Limit Mode is not defined or created using the Pulse Profiles Table dialog in CTRIO Workbench. However, an output must be configured for Pulse (Step/Direction) or Pulse (CW/CCW) using the Configure IO dialog. Also, an input must be configured as a limit for the Output channel to be used. The profile parameters are specified in CPU/controller memory registers. There are no acceleration ramps with Run to Limit Mode. The output will step directly to the specified frequency and duty cycle.

The user specifies the frequency, pulse train duty cycle and which edge of which limit input to stop at. When the Limit is reached the pulse output is disabled.



NOTE: Run to Limit Mode controls the pulse outputs directly from the CPU/controller program. No CTRIO Workbench Pulse Profile is required for this mode.

Set Word Parameter 1 to the desired Frequency. Set Word Parameter 2 Low Byte to the Duty Cycle and the High Byte to the appropriate Input and signal Edge. Leaving the Duty Cycle set to 0 achieves the default (50%), otherwise it can be set in 1% increments by writing this value from 1 to 99 Hex/BCD.

Pulse Output Status/Control Bits and Command Codes (DL PLCs)

- DL
- Win
- NI

Output Status Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Status Bit CTRIO(2) to CPU	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Input Start (octal)
Output Enabled	32, 48	22.0, 23.0
Position Loaded	33, 49	22.1, 23.1
Output Suspended	34, 50	22.2, 23.2
Output Active	36, 52	22.4, 23.4
Output Stalled	37, 53	22.5, 23.5
Command Error	38, 54	22.6, 23.6
Command Complete	39, 55	22.7, 23.7

Output Control Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Control Bit CPU to CTRIO(2)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)	Read as:
Enable Output	32, 48	26.0, 27.0	Level
Go to Position	33, 49	26.1, 27.1	Rising Edge
Suspend Output	34, 50	26.2, 27.2	Level
Direction	36, 52	26.4, 27.4	Level
Process Command	39, 55	26.7, 27.7	Rising Edge

Output Control (D)Words (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Word Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
Command Code	0, 6	10, 16
Word Parameter 1	1, 7	11, 17
Word Parameter 2	2, 8	12, 20

DWord Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
DWord Parameter 3	0, 2	0, 4

Run to Limit Mode, cont'd

Parameter 2

Word Parameter 2 defines three elements of the Run to Limit routine. Bits 13 and 12 determine which edge(s) to terminate Output Pulses and Bits 9 and 8 determine which CTRIO(2) module input terminal to use for the limit. The low byte specifies the duty cycle.

Edge to Seek	Parameter 2 Bits 13 & 12	Parameter 2 Bits 9 & 8	Parameter 2 (Hex) (Duty cycle at 50%)
Rising Edge Ch1 C	00	00	0000
Falling Edge Ch1 C	01	00	1000
Both Edge Ch1 C	10	00	2000
Rising Edge Ch1 D	00	01	0100
Falling Edge Ch1 D	01	01	1100
Both Edge Ch1 D	10	01	2100
Rising Edge Ch2 C	00	10	0200
Falling Edge Ch2 C	01	10	1200
Both Edge Ch2 C	10	10	2200
Rising Edge Ch2 D	00	11	0300
Falling Edge Ch2 D	01	11	1300
Both Edge Ch2 D	10	11	2300

Edge(s)	Bits 15..12	CTRIO Input	Bits 11..8
Rising	0000, 0Hex	Ch 1 C	0000, 0Hex
Falling	0001, 1Hex	Ch 1 D	0001, 1Hex
Both	0010, 2Hex	Ch 2 C	0010, 2Hex
		Ch 2 D	0011, 3Hex

Run at Velocity on CTRIO(2) Y0 and Y1 until Discrete Input Limit

Dynamic Velocity Example

Example: You will need to have a Dynamic Velocity profile configured as Table File Number 1 using the Configure I/O dialog. You will also need to have the appropriate Target Velocity (signed decimal) stored in V3000 for this example. You must turn C0 on to initialize the settings and enable the output.

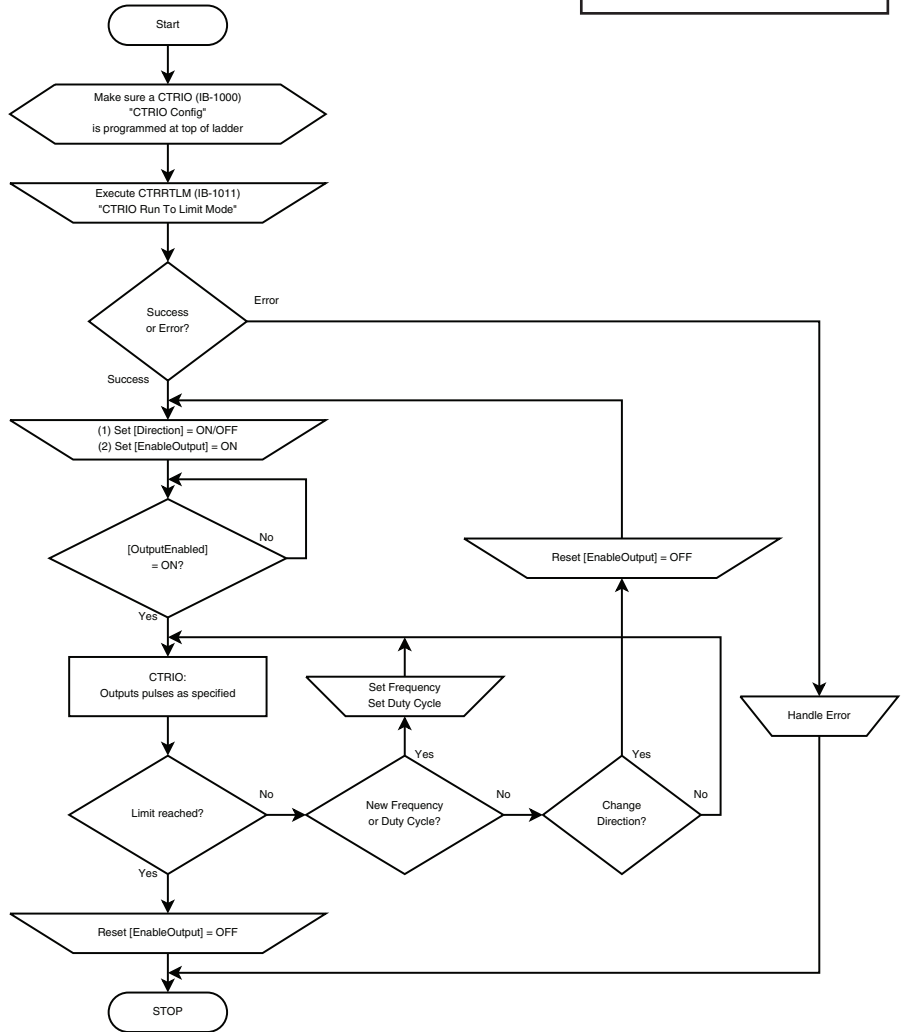


Steps	Name	PLC Control Outputs Base Addr = V2030 (Bit-of-Word)	PLC Status Inputs Base Addr = V2000 (Bit-of-Word)	PLC Control Outputs Base Addr = V2030 (Control Relay) D2-240	PLC Status Inputs Base Addr = V2000 (Control Relay) D2-240	Action
1	Command Code	V2040	N/A	V2040	N/A	Set to 21 Hex (Run to Limit Mode)
2	Parameter 1	V2041		V2041		Set initial run frequency, decimal CTRIO: 20Hz-25000Hz CTRIO2: 20Hz - 64000Hz
3	Parameter 2	V2042		V2042		Select discrete input edge in high byte, low byte = duty cycle (1-99) Example: rising input 1D at Duty = 45%, set this parameter to 212D Hex
4	Set Direction	V2056.4		C224		Set ON or OFF for Direction of Rotation
5	Process Command	V2056.7		C227		Turn ON Command Complete status bit is returned (see step 6)
6	Command Status	N/A	V2022.7	N/A	C127	When ON, command has been accepted, clear Process Command bit (step 5)
7	Command Error		V2022.6			C126
8	Enable Output	V2056.0	N/A	C220	N/A	Turn ON to start pulses
9	Output Active Status	N/A	V2022.4	N/A	C124	ON while pulsing, OFF when limit has stopped pulsing
10	Suspend Output	V2056.2	N/A	C222	N/A	Turn ON to "pause" output pulses without resetting pulse count
11	Output Suspended	N/A	V2022.2	N/A	C122	ON when out pulse train has been suspended

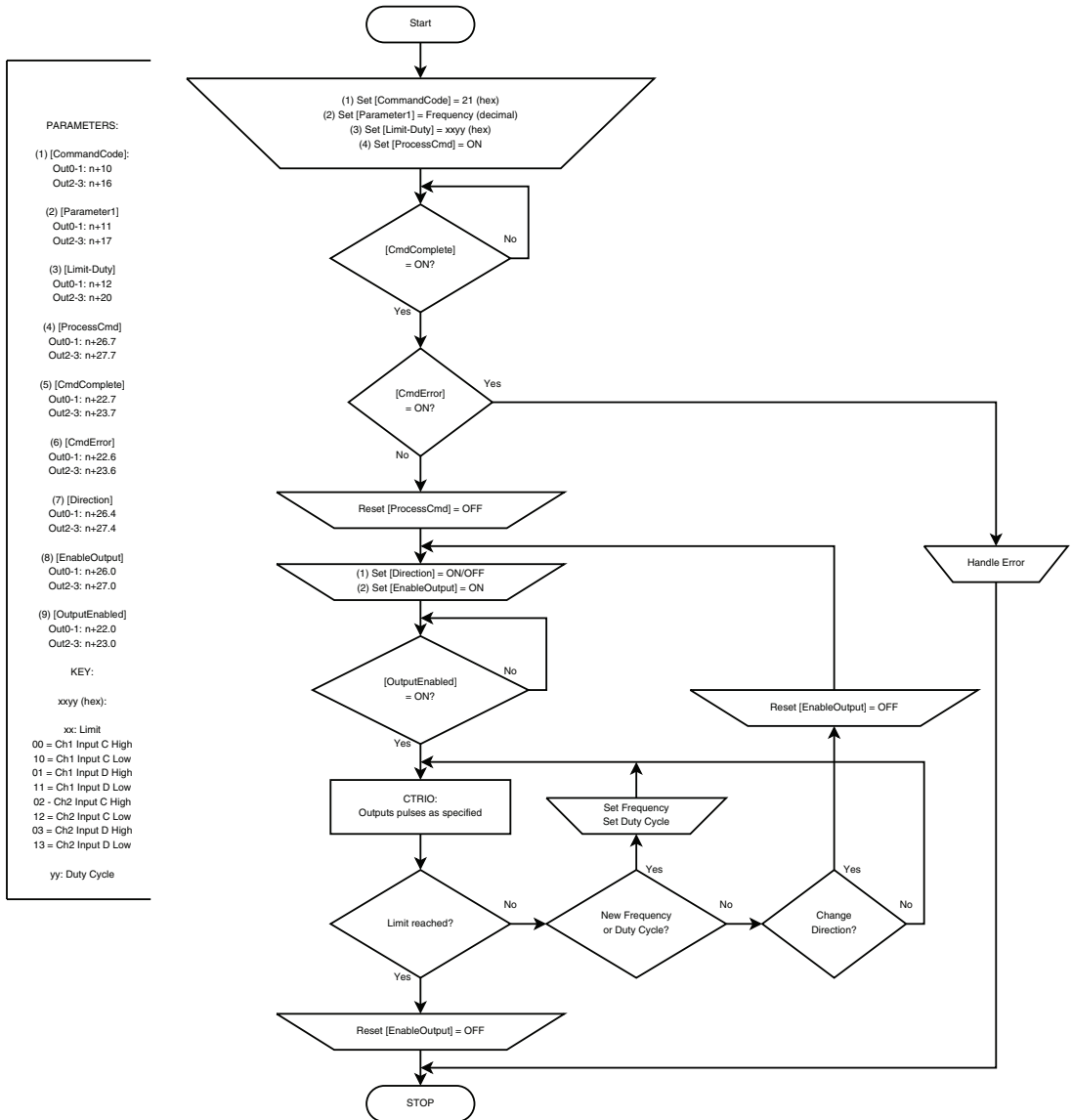
Run To Limit Mode (IBoxes)

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

- PARAMETERS:**
- (1) [Direction]
Out0-1: n+26.4
Out2-3: n+27.4
 - (2) [EnableOutput]
Out0-1: n+26.0
Out2-3: n+27.0
 - (3) [OutputEnabled]
Out0-1: n+22.0
Out2-3: n+23.0



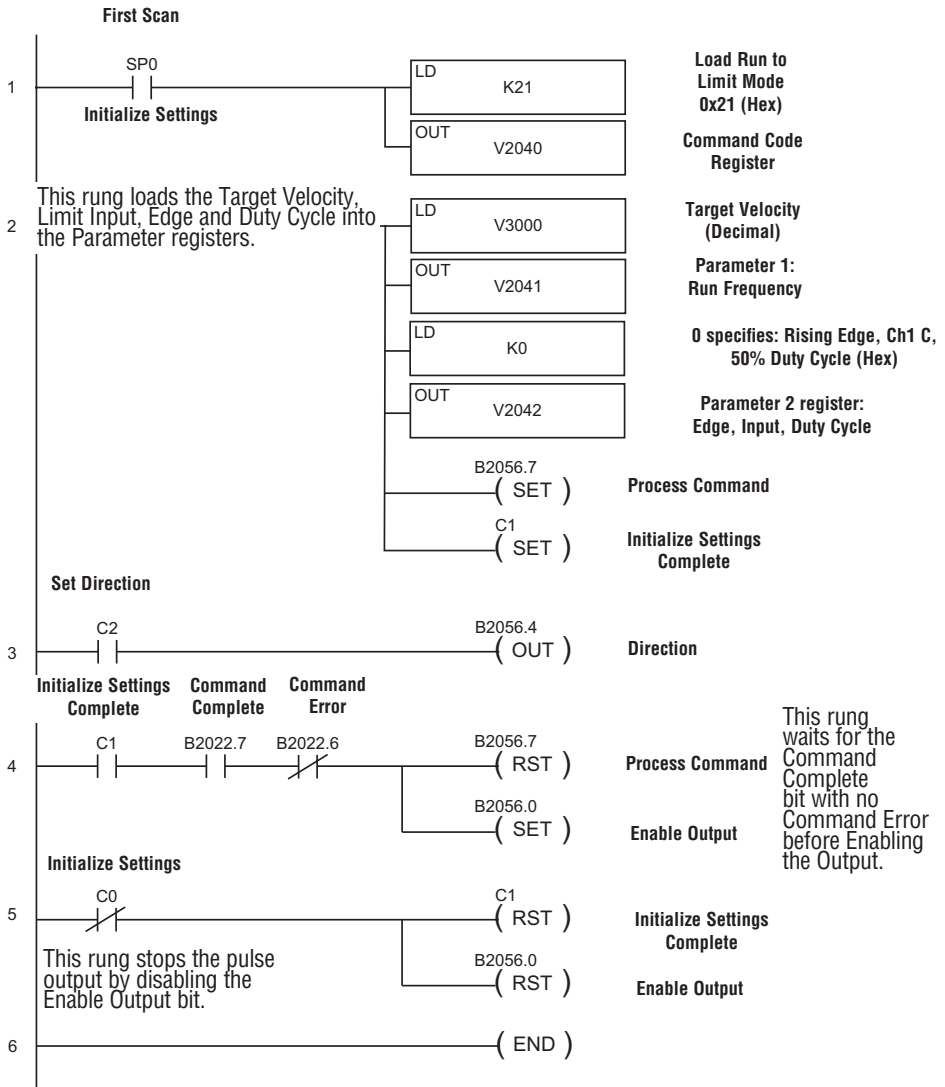
Run To Limit Mode (DL-PLC)



Run to Limit Mode Example

The following example program loads and executes a Run to Limit Mode pulse profile. Turn on C0 to run the profile. CTRIO input C or D must be assigned to Limit for this profile. Once running, velocity and duty cycle can be changed on the fly by loading new values into V2041 and V2042 respectively.

- DL
- Win
- NI



Run to Position Mode

- DL
- Win
- NI

Run to Position is used to make a simple move to a new position. If using a CTRIO2 module, consider using Dynamic Positioning Plus instead. Dynamic Positioning Plus is almost always the better choice.

Run to Position Mode is not defined or created using the Pulse Profiles Table dialog in CTRIO Workbench. However, an output must be configured for Pulse (Step/Direction) or Pulse (CW/CCW) using the Configure IO dialog. Also, an input must be configured as a counter. The profile parameters are specified in CPU/controller memory registers. There are no acceleration ramps with Run to Limit Mode. The output will step directly to the specified frequency and duty cycle. Use caution when choosing the frequency to avoid stalling a stepper.

The user specifies the target frequency, pulse train duty cycle, input function to compare, direction and target position. The current position is obtained from the specified Input Function (i.e. Quadrature counter). When the current position satisfies the comparison specified, the pulse output is disabled. The comparison of the current and target positions can be based on greater than or equal to or less than operators.



NOTE: Run to Position Mode controls the pulse outputs directly from the CPU/controller program. No CTRIO Workbench Pulse Profile is required for this mode.

The Run to Position Mode command (Command = 0022Hex/BCD) allows generating output pulses at a steady frequency until some value has been reached on a counter input. Select the direction using the direction bit. Set Word Parameter 1 to the desired Frequency. Set Word Parameter 2 Low Byte to the Duty Cycle and the High Byte to the Compare Functions as defined below. Leaving the Duty Cycle set to 0 achieves the default (50%), otherwise it can be set in 1% increments by writing this value from 1 to 99 Hex/BCD.

Word Parameter 3 specifies the value that Input Function will compare against.

Pulse Output Status/Control Bits and Command Codes (DL PLCs)

- DL
- Win
- NI

Output Status Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Status Bit CTRIO(2) to CPU	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Input Start (octal)
Output Enabled	32, 48	22.0, 23.0
Position Loaded	33, 49	22.1, 23.1
Output Suspended	34, 50	22.2, 23.2
Output Active	36, 52	22.4, 23.4
Output Stalled	37, 53	22.5, 23.5
Command Error	38, 54	22.6, 23.6
Command Complete	39, 55	22.7, 23.7

Output Control Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Control Bit CPU to CTRIO(2)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)	Read as:
Enable Output	32, 48	26.0, 27.0	Level
Go to Position	33, 49	26.1, 27.1	Rising Edge
Suspend Output	34, 50	26.2, 27.2	Level
Direction	36, 52	26.4, 27.4	Level
Process Command	39, 55	26.7, 27.7	Rising Edge

Output Control (D)Words (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Word Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
Command Code	0, 6	10, 16
Word Parameter 1	1, 7	11, 17
Word Parameter 2	2, 8	12, 20

DWord Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
DWord Parameter 3	0, 2	0, 4

Parameter 2

Word Parameter 2 defines three elements of the Run to Position routine. Bit 12 determines if the specified position is “greater than or equal” or “less than” the current Input Function position value. Bits 9 and 8 determine which Input Function to use for the comparison. The low byte specifies the duty cycle.

Specified Position (Parameter 3) is:	Parameter 2 Bit 12	Parameter 2 Bits 9 & 8	Parameter 2 (Hex) (Duty cycle at 50%)
Less than Ch1/Fn1	0	00	0000
Greater than Ch1/Fn1	1	00	1000
Less than Ch1/Fn2	0	01	0100
Greater than Ch1/Fn2	1	01	1100
Less than Ch2/Fn1	0	10	0200
Greater than Ch2/Fn1	1	10	1200
Less than Ch2/Fn2	0	11	0300
Greater than Ch2/Fn2	1	11	1300

Comparison	Bits 15..12	Input Function	Bits 11..8
Greater Than or Equal	0001, 1Hex	Ch 1 Fn 1	0000, 0Hex
Less Than	0000, 0Hex	Ch 1 Fn 2	0001, 1Hex
		Ch 2 Fn 1	0010, 2Hex
		Ch 2 Fn 2	0011, 3Hex

Run at Velocity on CTRIO(2) until Input Function Value Position

DL

Win

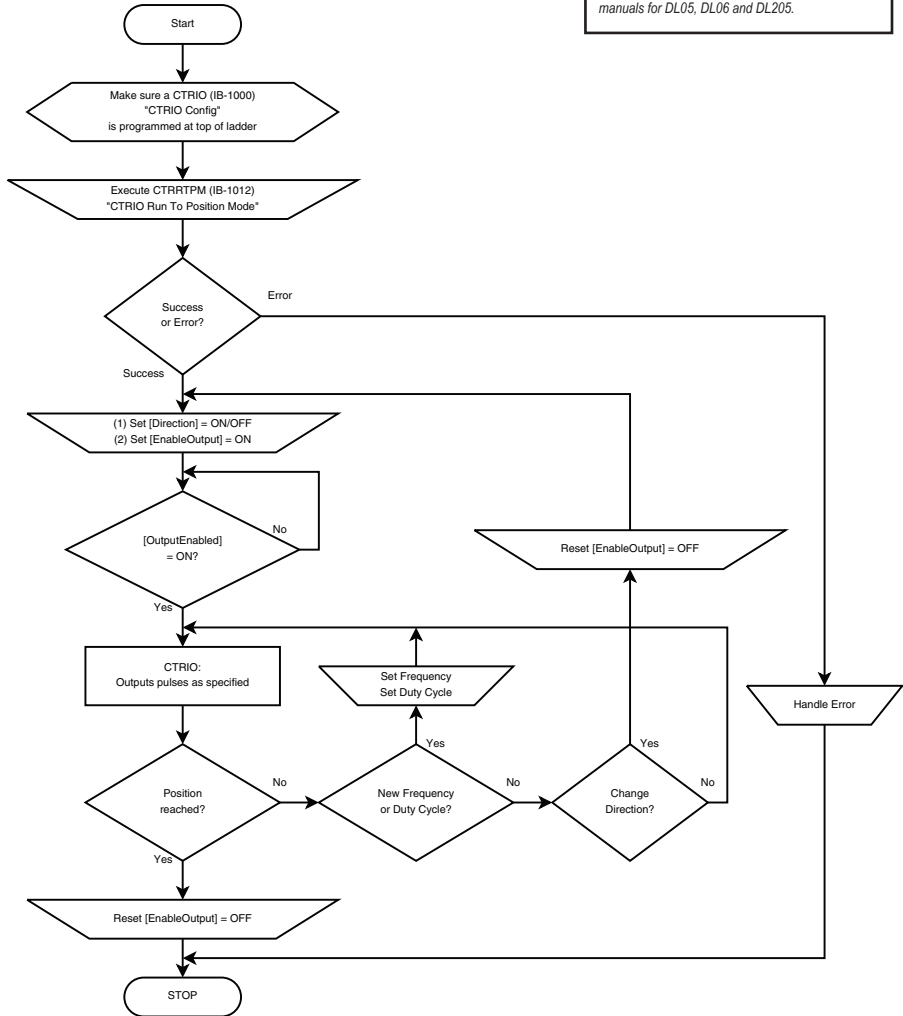
NI

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	V2040	N/A	V2040	N/A	Set to 22 Hex (Pulse at velocity until Function Input Limit)
2	Parameter 1	V2041		V2041		Set initial run frequency, decimal CTRIO: 20Hz-25000Hz CTRIO2: 20Hz - 64000Hz
3	Parameter 2	V2042		V2042		Bits 15-12: Comparison Bits 11-8: Input Function to use Low Byte: Duty cycle (1-99) (can leave 0 for 50%) Hex
4	Parameter 3	V2031 / V2030		V2031 / V2030		Specified position for Input Function DWord to compare against, decimal
5	Set Direction	V2056.4		C224		Set ON or OFF for Direction of Rotation
6	Process Command	V2056.7		C227		Turn ON Command Complete status bit is returned (see step 7)
7	Command Status	N/A	V2022.7	N/A	C127	When ON, command has been accepted, clear Process Command bit (step 6)
8	Command Error		V2022.6		C126	ON if Command or Parameters are invalid
9	Enable Output	V2056.0	N/A	C220	N/A	Turn ON to start pulses
10	Output Active Status	N/A	V2022.4	N/A	C124	ON while pulsing, OFF when position is reached
11	Suspend Output	V2056.2	N/A	C222	N/A	Turn ON to "pause" output pulses without resetting pulse count
12	Output Suspended	N/A	V2022.2	N/A	C122	ON when out pulse train has been suspended

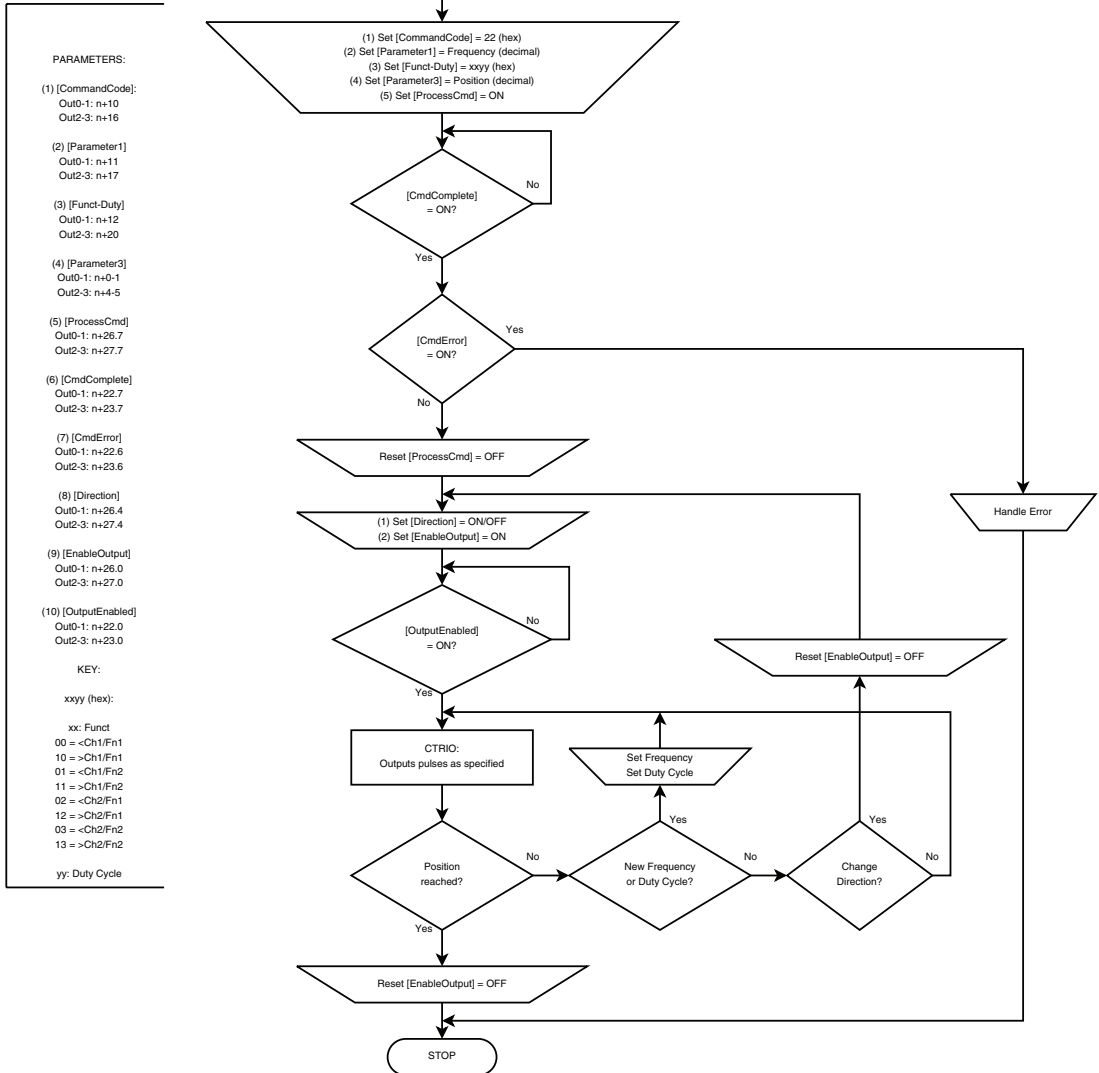
Run To Position Mode (IBoxes)

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

- PARAMETERS:**
- (1) [Direction]
Out0-1: n+26.4
Out2-3: n+27.4
 - (2) [EnableOutput]
Out0-1: n+26.0
Out2-3: n+27.0
 - (3) [OutputEnabled]
Out0-1: n+22.0
Out2-3: n+23.0



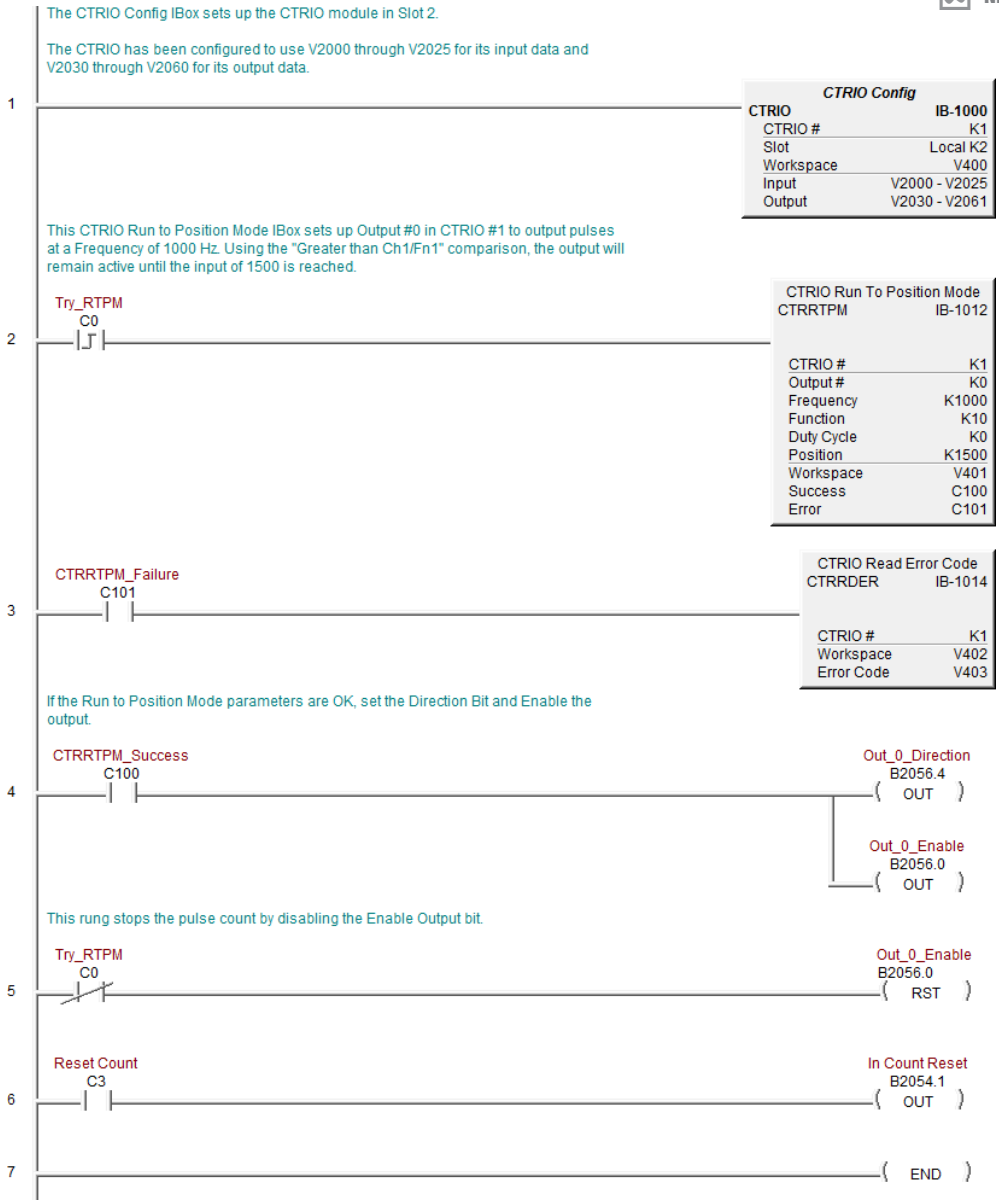
Run To Position Mode
(DL-PLC)



Run to Position Mode with DirectSOFT IBox Instructions

This is the equivalent ladder using IBox instructions in *DirectSOFT* to the ladder shown on the previous page. Turn on C0 to run the pulse profile. Once running, velocity and duty cycle can be changed on the fly by loading new values into V2041 and V2042 respectively.

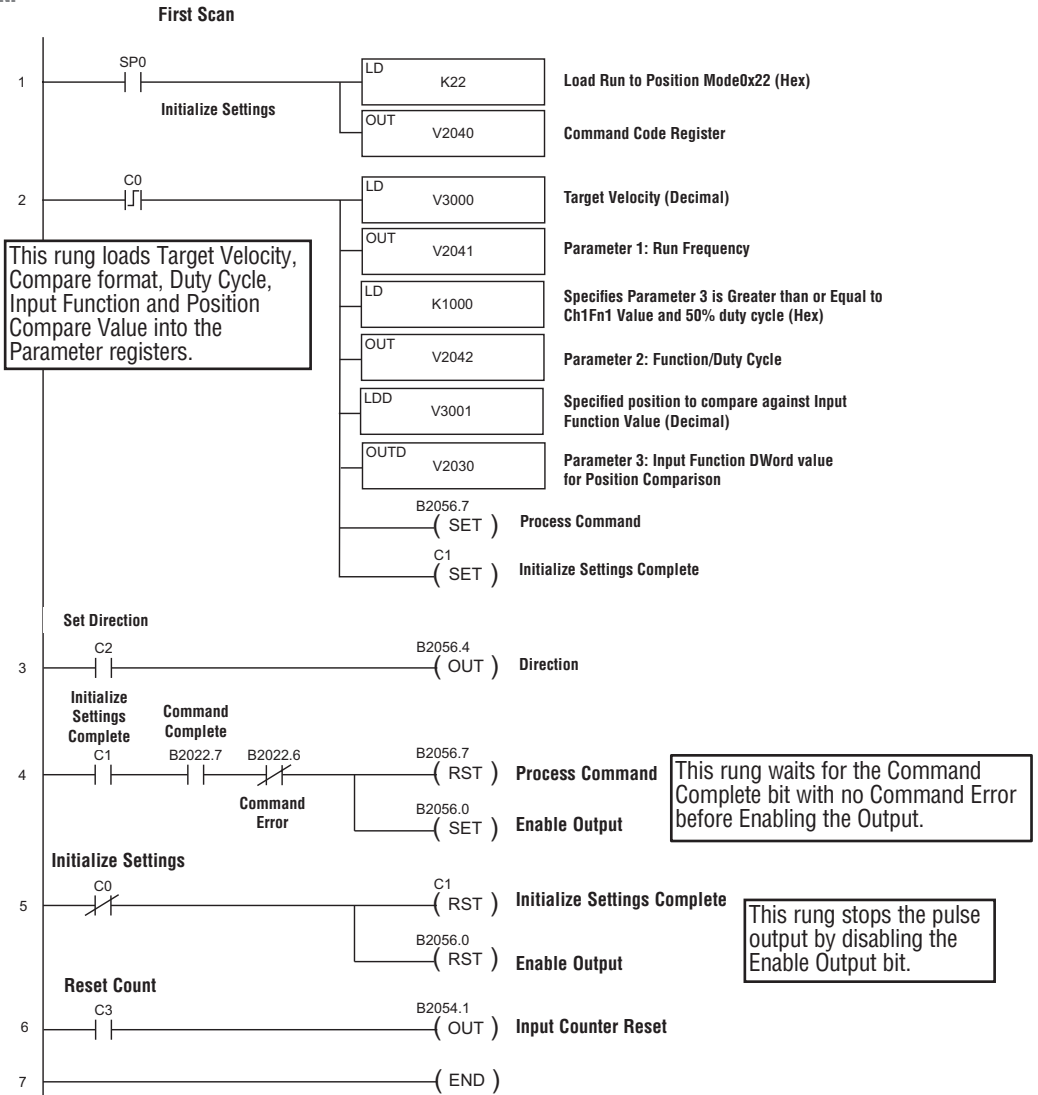
- DL
- Win
- NI



Run to Position Mode Example

The following example program loads and executes a Run to Position Mode pulse profile. Turn on C0 to run the pulse profile. Once running, velocity and duty cycle can be changed on the fly by loading new values into V2041 and V2042 respectively.

- DL
- Win
- NI



Raw Output

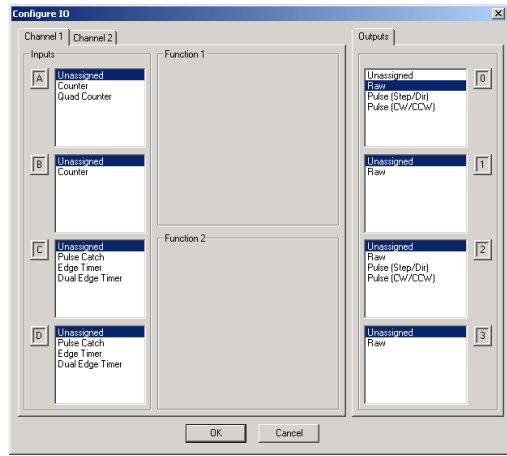
- DL
- Win
- NI

The CTRIO(2) module supports Raw output mode. This mode allows the CPU/controller program to have direct access to the module's output points. Each output can be configured for Raw output mode and each will have a unique control bit.

Refer to *Output Control Bit Definitions (Raw Mode)* in Appendix A for Raw output control bit addressing.



The module output terminals are represented by the 0, 1, 2, and 3 boxes (0 and 1 for the H0-CTRIO(2)) on the Outputs tab at the right side of this Configure IO dialog box.



Pulse Output Status/Control Bits and Command Codes (DL PLCs)

- DL
- Win
- NI

Output Status Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Status Bit CTRIO(2) to CPU	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Input Start (octal)
Output Enabled	32, 48	22.0, 23.0
Position Loaded	33, 49	22.1, 23.1
Output Suspended	34, 50	22.2, 23.2
Output Active	36, 52	22.4, 23.4
Output Stalled	37, 53	22.5, 23.5
Command Error	38, 54	22.6, 23.6
Command Complete	39, 55	22.7, 23.7

Output Control Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Control Bit CPU to CTRIO(2)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)	Read as:
Enable Output	32, 48	26.0/26.8, 27.0/27.8	Level
Go to Position	33, 49	26.1, 27.1	Rising Edge
Suspend Output	34, 50	26.2, 27.2	Level
Direction	36, 52	26.4, 27.4	Level
Process Command	39, 55	26.7, 27.7	Rising Edge

Output Control (D)Words (Pulse Output)

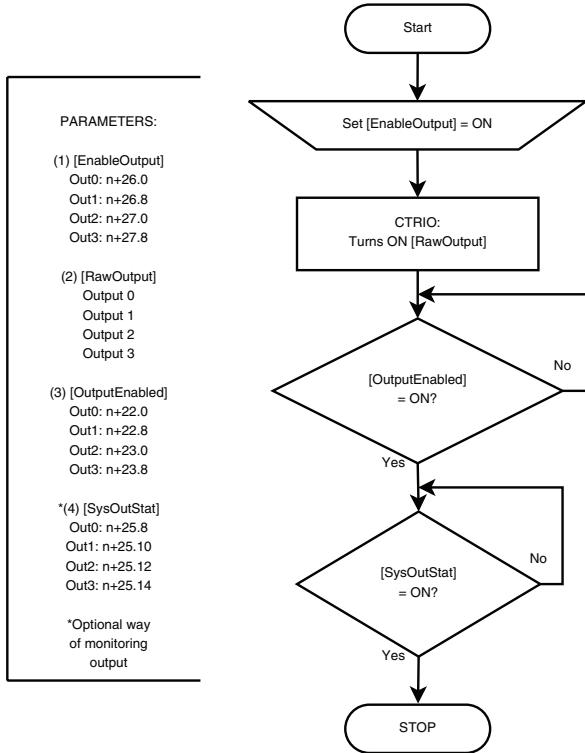
Word Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
Command Code	0, 6	10, 16
Word Parameter 1	1, 7	11, 17
Word Parameter 2	2, 8	12, 20

DWord Control CPU to CTRIO(2)	Word Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
DWord Parameter 3	0, 2	0, 4

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Raw Output (IBoxes)

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



Raw Output (DL-PLC)

PARAMETERS:

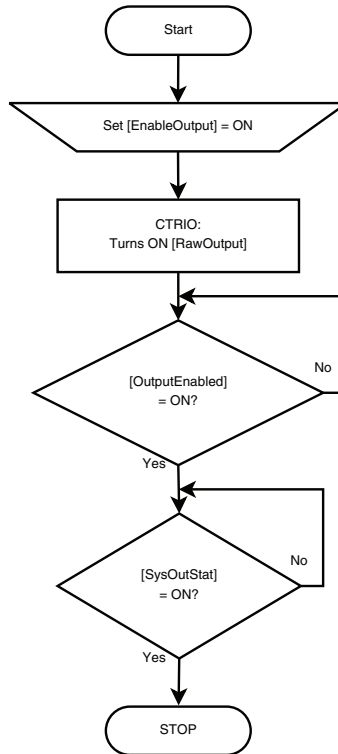
(1) [EnableOutput]
 Out0: n+26.0
 Out1: n+26.8
 Out2: n+27.0
 Out3: n+27.8

(2) [RawOutput]
 Output 0
 Output 1
 Output 2
 Output 3

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

* (4) [SysOutStat]
 Out0: n+25.8
 Out1: n+25.10
 Out2: n+25.12
 Out3: n+25.14

*Optional way
 of monitoring
 output



INPUT FUNCTIONS



CHAPTER 10

In This Chapter...

Input Memory Mapping for Counter Data Transfer	10-2
Input Memory Map for Scaled Counter Data	10-5
Input Memory Map for Capture Count Data Transfers	10-9
Input Memory Map for Edge Timer and Dual Edge Timer	10-12
Inhibit.....	10-22
Input Memory Map for Pulse Catch Data Transfers.....	10-25

- DL
- Win
- NI

Input Memory Mapping for Counter Data Transfer

The Counter function is used to create counts from an attached encoder. To use this function, one input needs to be configured as a Counter or Quad Counter.

The following table shows which memory locations are used for memory transfers from the CTRIO(2) module to the CPU. The starting memory location is defined by the user in the I/O Map within CTRIO Workbench. For the *DirectLOGIC* CPU, memory address offsets are in the second column. When using H2-WinPLC, EBC, PBC, MODBUS, or DEVNETS CPUs in the CPU slot, use the non-PLC offsets in column one.

The proper sequence of events for using this function can be found in the flow charts shown in later pages following.

Data Type and Offset WinPLC, EBC, PBC, DEVNETS, MODBUS	Address for Inputs (<i>DirectLOGIC</i>)	Definition	Format	Bytes
dwX0	n+0	Ch 1/Fn 1 Parameter 1	DWord	4
dwX1	n+2	Ch 1/Fn 1 Parameter 2		
dwX2	n+4	Ch 1/Fn 2 Parameter 1		
dwX3	n+6	Ch 1/Fn 2 Parameter 2		
dwX4	n+10	Ch 2/Fn 1 Parameter 1		
dwX5	n+12	Ch 2/Fn 1 Parameter 2		
dwX6	n+14	Ch 2/Fn 2 Parameter 1		
dwX7	n+16	Ch 2/Fn 2 Parameter 2		
bX0...7 bX8...15	n+20	Ch 1/Fn 1 Status (Low Byte) Ch 1/Fn 2 Status (High Byte)	Word	2
bX16...23 bX24...31	n+21	Ch 2/Fn 1 Status (Low Byte) Ch 2/Fn 2 Status (High Byte)		
bX32...39 bX40...47	n+22	Output 0 Status (Low Byte) Output 1 Status (High Byte)		
bX48...55 bX56...63	n+23	Output 2 Status (Low Byte) Output 3 Status (High Byte)		
bX64...71 bX72...79 bX80...87 bX88...95	n+24	System Functions Read/Write CTRIO(2) Internal Registers (see p. 6-10 for bit definitions)	DWord	4

**44 Total
Bytes**

Example Input Control/Status Bits and Parameter Register Addresses

The following tables provide example addresses based on V2000 selected for the base input address. The Input Functions discussed on the following pages use these example addresses.

Input(n) Parameter Definitions

Parameter values are in Decimal format.

Configured Function from CTRIO Workbench	Parameter 1 Contents DWORD	Parameter 2 Contents DWORD
Non-scaled Counter	Raw Input Value	Not Used
Scaled Counter	Scaled Value (pos. or rate)	Raw Value
Non-scaled Counter with Capture	Raw Value	Captured Value
Scaled Counter with Capture	Scaled Value (pos. or rate)	Scaled Captured Value
Non-scaled Timer	Previous Time (μs)	In Progress Time (μs)
Scaled Timer	Scaled Interval (rate)	Scaled In Progress Time (μs)
Pulse Catch	Not Used	Not Used

Counter & Quadrature Counter

Parameter 1 and 2 values, shown in the table above, will be mapped to V2000 - V2003 in this example. If input D is configured for Count Capture, the Enable Count Capture bit (see table below) must be ON in order for input D to be able to snapshot the current count. The Counter Capture Complete bit is used to indicate the acquisition has occurred. The program will need to turn OFF the Enable Capture and confirm the Capture Complete bit resets before attempting the next count capture. The Reset bit will reset raw and scaled values to the specified reset value. The last captured value, if applicable, will remain at V2054.

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	Description
Parameter 1	N/A	V2001-V2000	N/A	V2001-V2000	Refer to table above
Parameter 2		V2003-V2002		V2003-V2002	Refer to table above
Counter Capture Complete		V2020.0		C160	On when Count Capture is completed (Available only when input D is configured for Capture input)
Enable Count Capture	V2054.0	N/A	C260	N/A	Turn ON to capture Count (Available only when input D is configured for Capture input)
Reset	V2054.1		C261		Turn ON to reset Counter Value to Reset Value
At Reset Value	N/A	V2020.1	N/A	C161	On when Counter is at Reset Value



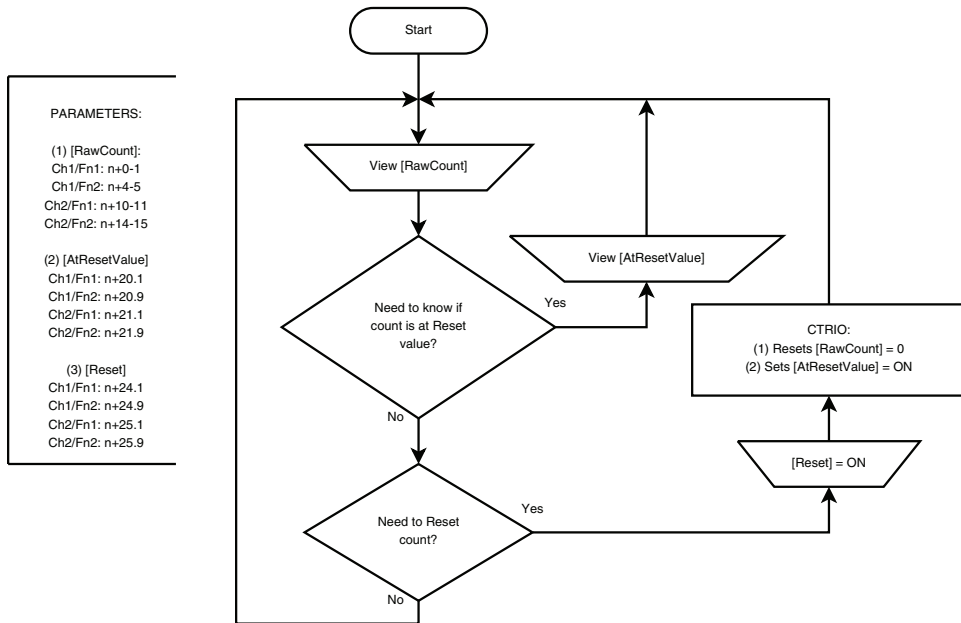
NOTE: For *DirectSOFT* users: the I/O Map dialog displays the exact memory locations in use by the CTRIO(2) module. Within the I/O Map dialog you can print out a report of memory locations in use.

Input Function Status Bit Definitions

Input function offsets are listed in the order of Ch1/Fn1, Ch1/Fn2, Ch2/Fn1, Ch2/Fn2

Ch(x)/Fn(x) Status Bits (transfers from CTRIO(2) to CPU)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets <i>DirectLOGIC</i> PLCs
Count Capture Complete Bit	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8

Counter Function (IBoxes & DL-PLC)



Input Memory Map for Scaled Counter Data

- DL
- Win
- NI

Scaled Counter is simply used to create counts in user-specified engineering units from an attached encoder. To use this function, one input needs to be configured as a Counter or Quad Counter. Then, for that Input functions, select the ruler icon and use the wizard to configure the desired engineering units.

This is the final panel in the Scaling Wizard. Be sure to use the Position Scaling Calculator to validate that the Raw Value (raw counts from the encoder) will generate the desired Scaled Value. Type in the test counts (50 is shown here) and verify the expected engineering units value is displayed in the Scaled Value field. It will be displayed with the Engineering Units characters provided earlier in the Wizard. FPS (feet per second) was used here. The Calculator returned '5FPS' here.

The screenshot shows a dialog box titled "Scaling Wizard - Position Settings". It contains the following fields and controls:

- Minimum Raw Value: counts
- Maximum Raw Value: counts
- Minimum Scaled Value: FPS
- Maximum Scaled Value: FPS
- A section titled "Position Scaling Calculator" containing:
 - Text: "Enter a raw count value to confirm scaling configuration."
 - Raw Value: counts
 - Scaled Value:
- Buttons: "Cancel", "< Back", and "Finish"

The following table shows which memory locations are used for memory transfers from the CTRIO(2) module to the CPU. The starting memory location is defined by the user in the I/O Map within CTRIO Workbench. When configuring the *Direct*LOGIC CPU, use the memory address offsets in the second column. When configuring an H2-WinPLC, EBC, PBC, MODBUS, or DEVNETS placed in the CPU slot, use the non-PLC offsets in column one.

Data Type and Offset WinPLC, EBC, PBC, DEVNETS, MODBUS	Address for Inputs (<i>Direct</i> LOGIC)	Definition	Format	Bytes
dwX0	n+0	Ch 1/Fn 1 Parameter 1	DWord	4
dwX1	n+2	Ch 1/Fn 1 Parameter 2		
dwX2	n+4	Ch 1/Fn 2 Parameter 1		
dwX3	n+6	Ch 1/Fn 2 Parameter 2		
dwX4	n+10	Ch 2/Fn 1 Parameter 1		
dwX5	n+12	Ch 2/Fn 1 Parameter 2		
dwX6	n+14	Ch 2/Fn 2 Parameter 1		
dwX7	n+16	Ch 2/Fn 2 Parameter 2	Word	2
bX0...7 bX8...15	n+20	Ch 1/Fn 1 Status (Low Byte) Ch 1/Fn 2 Status (High Byte)		
bX16...23 bX24...31	n+21	Ch 2/Fn 1 Status (Low Byte) Ch 2/Fn 2 Status (High Byte)		
bX32...39 bX40...47	n+22	Output 0 Status (Low Byte) Output 1 Status (High Byte)		
bX48...55 bX56...63	n+23	Output 2 Status (Low Byte) Output 3 Status (High Byte)		
bX64...71 bX72...79 bX80...87 bX88...95	n+24	System Functions Read/Write CTRIO(2) Internal Registers (see p. 6-10 for bit definitions)	DWord	4

**44 Total
Bytes**

Input(n) Parameter Definitions

Parameter values are in Decimal format.

Configured Function from CTRIO Workbench	Parameter 1 Contents DWORD	Parameter 2 Contents DWORD
Non-scaled Counter	Raw Input Value	Not Used
Scaled Counter	Scaled Value (pos. or rate)	Raw Value
Non-scaled Counter with Capture	Raw Value	Captured Value
Scaled Counter with Capture	Scaled Value (pos. or rate)	Scaled Captured Value
Non-scaled Timer	Previous Time (μ s)	In Progress Time (μ s)
Scaled Timer	Scaled Interval (rate)	Scaled In Progress Time (μ s)
Pulse Catch	Not Used	Not Used

Counter & Quadrature Counter

Parameters 1 and 2 are explained above and will be mapped to V2000 - V2003 in this example. If input D is configured for Count Capture, the Enable Count Capture bit must be ON in order for input D to be able to snapshot the current count. The Counter Capture Complete bit is used to indicate the acquisition has occurred. The program will need to turn OFF the Enable Capture and confirm the Capture Complete bit resets before attempting the next count capture. The Reset bit will reset raw and scaled values to the specified reset value. The last captured value, if applicable, will remain.

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	Description
Parameter 1	N/A	V2001-V2000	N/A	V2001-V2000	Refer to table above
Parameter 2		V2003-V2002		V2003-V2002	Refer to table above
Counter Capture Complete		V2020.0		C160	On when Count Capture is completed (Available only when input D is configured for Capture input)
Enable Count Capture	V2054.0	N/A	C260	N/A	Turn ON to capture Count (Available only when input D is configured for Capture input)
Reset	V2054.1		C261		Turn ON to reset Counter Value to Reset Value
At Reset Value	N/A	V2020.1	N/A	C161	On when Counter is at Reset Value



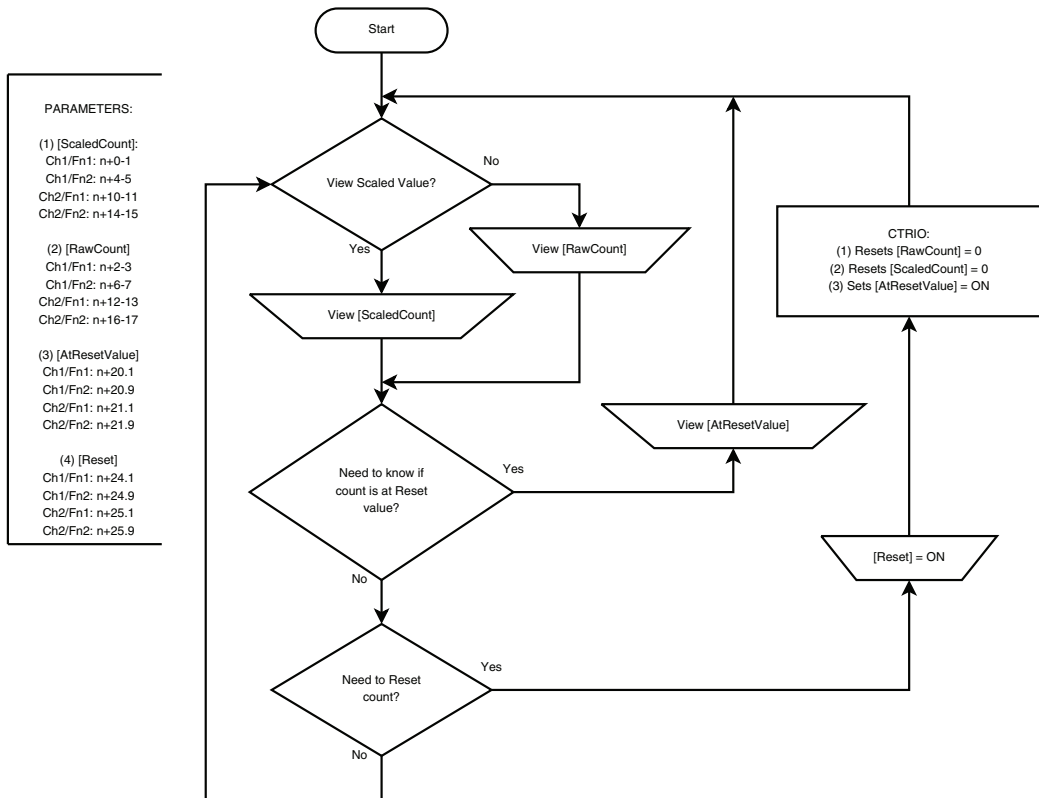
NOTE: For *DirectSOFT* users: the *I/O Map* dialog displays the exact memory locations in use by the *CTR10(2)* module. Within the *I/O Map* dialog you can print out a report of memory locations in use.

Input Function Status Bit Definitions

Input function offsets are listed in the order of Ch1/Fn1, Ch1/Fn2, Ch2/Fn1, Ch2/Fn2

Ch(x)/Fn(x) Status Bits (Transfers from CTRIO(2) to CPU)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets <i>DirectLOGIC</i> PLCs
Count Capture Complete Bit	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8

Counter Function (Scaled) (iBoxes & DL-PLC)



Input Memory Map for Capture Count Data Transfers

- DL
- Win
- NI

This function copies the current value of a counter input into a register when a discrete input trigger condition is met. To use this function, one input must be configured as a Counter or Quad Counter and one as Capture Fn. An input must be set as a counter before Capture Fn will appear in the menu of CTRIO Workbench dialog box.

The following table shows which memory locations are used for memory transfers from the CTRIO(2) module to the CPU. The starting memory location is defined by the user in the I/O Map within CTRIO Workbench. When configuring the *DirectLOGIC* CPU, use the memory address offsets in the second column. When configuring an H2-WinPLC, EBC, PBC, MODBUS, or DEVNETS placed in the CPU slot, use the non-PLC offsets in column one.

The proper sequence of events for using this function can be found in the flow charts on the pages that follow.

Data Type and Offset WinPLC, EBC, PBC, DEVNETS, MODBUS	Address for Inputs (<i>DirectLOGIC</i>)	Definition	Format	Bytes
dwX0	n+0	Ch 1/Fn 1 Parameter 1	DWord	4
dwX1	n+2	Ch 1/Fn 1 Parameter 2		
dwX2	n+4	Ch 1/Fn 2 Parameter 1		
dwX3	n+6	Ch 1/Fn 2 Parameter 2		
dwX4	n+10	Ch 2/Fn 1 Parameter 1		
dwX5	n+12	Ch 2/Fn 1 Parameter 2		
dwX6	n+14	Ch 2/Fn 2 Parameter 1		
dwX7	n+16	Ch 2/Fn 2 Parameter 2		
bX0...7 bX8...15	n+20	Ch 1/Fn 1 Status (Low Byte) Ch 1/Fn 2 Status (High Byte)	Word	2
bX16...23 bX24...31	n+21	Ch 2/Fn 1 Status (Low Byte) Ch 2/Fn 2 Status (High Byte)		
bX32...39 bX40...47	n+22	Output 0 Status (Low Byte) Output 1 Status (High Byte)		
bX48...55 bX56...63	n+23	Output 2 Status (Low Byte) Output 3 Status (High Byte)		
bX64...71 bX72...79 bX80...87 bX88...95	n+24	System Functions Read/Write CTRIO(2) Internal Registers (see p. 6-10 for bit definitions)	DWord	4

44 Total
Bytes

Input(n) Parameter Definitions

Parameter values are in Decimal format.

Configured Function from CTRIO Workbench	Parameter 1 Contents DWORD	Parameter 2 Contents DWORD
Non-scaled Counter	Raw Input Value	Not Used
Scaled Counter	Scaled Value (pos. or rate)	Raw Value
Non-scaled Counter with Capture	Raw Value	Captured Value
Scaled Counter with Capture	Scaled Value (pos. or rate)	Scaled Captured Value
Non-scaled Timer	Previous Time (μ s)	In Progress Time (μ s)
Scaled Timer	Scaled Interval (rate)	Scaled In Progress Time (μ s)
Pulse Catch	Not Used	Not Used



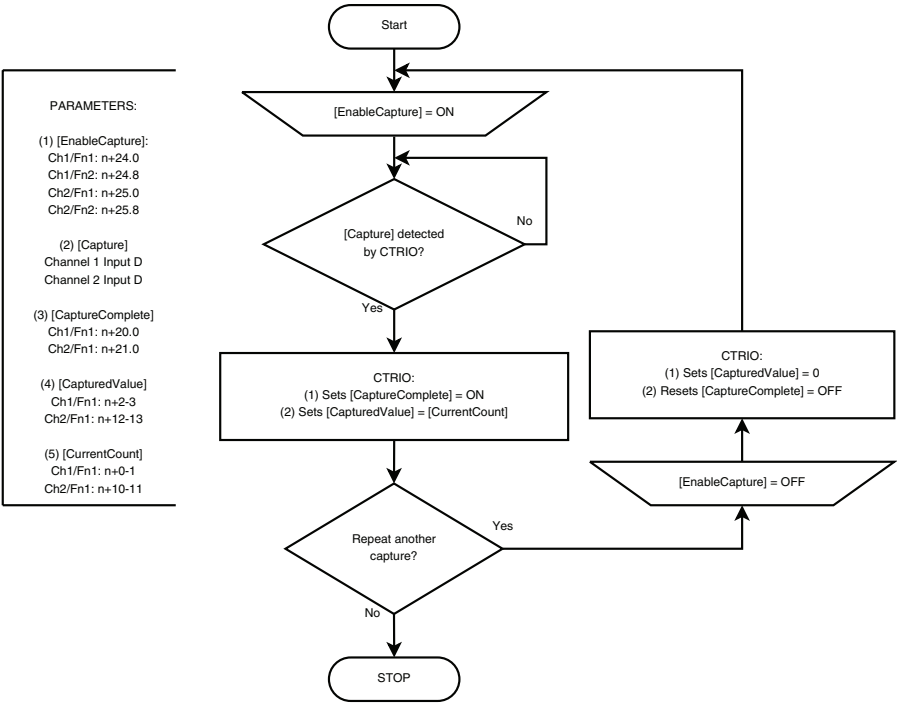
NOTE: For *DirectSOFT* users: the *I/O Map* dialog displays the exact memory locations in use by the CTRIO(2) module. Within the *I/O Map* dialog you can print out a report of memory locations in use.

Input Function Status Bit Definitions

Input function offsets are listed in the order of Ch1/Fn1, Ch1/Fn2, Ch2/Fn1, Ch2/Fn2

Ch(x)/Fn(x) Status Bits (Transfers from CTRIO(2) to CPU)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets <i>DirectLOGIC</i> PLCs
Count Capture Complete Bit	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8
Timer Capture Start	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8
Timer Capture Complete (Timing) OR At Reset Value (Counting)	1, 9, 17, 25	20.1, 20.9, 21.1, 21.9
Timer "Timed Out" Bit	2, 10, 18, 26	20.2, 20.10, 21.2, 21.10
Pulse Catch Output Pulse State	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8
Pulse Catch Start	1, 9, 17, 25	20.1, 20.9, 21.1, 21.9

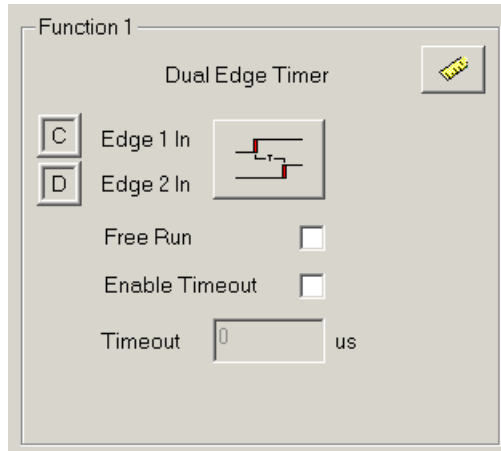
Capture Count Function (IBoxes & DL-PLC)



- DL
- Win
- NI

Input Memory Map for Edge Timer and Dual Edge Timer

Edge Timer and Dual Edge Timer are used to precisely measure the time between two discrete input events. To use one of these functions, configure one input as Edge Timer or configure two inputs as Dual Edge Timer. Then cycle through the various edge options by repeatedly pressing the button with the graphic depiction of the signal edges.



Function one shows the configuration for a Dual Edge Timer that will time from the rising edge of Input C to the rising edge of Input D. Free Run is not selected and Enable Timeout is not selected. There is no scaling. If Scaling were selected, the time result would be in the selected engineering units.

Running an Edge Timer with the default settings takes the least supporting code to operate. The default settings are Free Run disabled and Enable Timeout disabled.

When Free Run is enabled, the timer will start as soon as Edge 1 In is true. With Free Run enabled, the timer will always show the last valid capture value; or, 0 if the last measurement resulted in a timeout.

When Free Run is disabled, the timer will start when Enable Capture and Edge 1 are true. With Free Run disabled, the timer will show the last time captured.

When Enable Timeout is enabled, the timer will timeout if the Edge 2 event does not occur before the timeout timer expires. The timeout timer starts when Enable Capture is true and Edge 1 becomes true. When a timeout occurs, the Timeout and Capture Complete bit are set and the Timer and Last Time values are cleared.

When Enable Timeout is disabled, once Edge 1 event is true, the timer will run until Edge 2 event becomes true.

The behavior resulting from various combinations of the options above can also be found in the flow charts that follow.

The following table shows which memory locations are used for memory transfers from the CTRIO(2) module to the CPU. The starting memory location is defined by the user in the I/O Map within CTRIO Workbench. When configuring the *DirectLOGIC* CPU, use the memory address offsets in the second column. When configuring an H2-WinPLC, EBC, PBC, MODBUS, or DEVNETS placed in the CPU slot, use the non-PLC offsets in column one.

Data Type and Offset WinPLC, EBC, PBC, DEVNETS, MODBUS	Address for Inputs (<i>DirectLOGIC</i>)	Definition	Format	Bytes
dwX0	n+0	Ch 1/Fn 1 Parameter 1	DWord	4
dwX1	n+2	Ch 1/Fn 1 Parameter 2		
dwX2	n+4	Ch 1/Fn 2 Parameter 1		
dwX3	n+6	Ch 1/Fn 2 Parameter 2		
dwX4	n+10	Ch 2/Fn 1 Parameter 1		
dwX5	n+12	Ch 2/Fn 1 Parameter 2		
dwX6	n+14	Ch 2/Fn 2 Parameter 1		
dwX7	n+16	Ch 2/Fn 2 Parameter 2		
bX0...7 bX8...15	n+20	Ch 1/Fn 1 Status (Low Byte) Ch 1/Fn 2 Status (High Byte)	Word	2
bX16...23 bX24...31	n+21	Ch 2/Fn 1 Status (Low Byte) Ch 2/Fn 2 Status (High Byte)		
bX32...39 bX40...47	n+22	Output 0 Status (Low Byte) Output 1 Status (High Byte)		
bX48...55 bX56...63	n+23	Output 2 Status (Low Byte) Output 3 Status (High Byte)		
bX64...71 bX72...79 bX80...87 bX88...95	n+24	System Functions Read/Write CTRIO(2) Internal Registers (see p. 6-10 for bit definitions)	DWord	4

44 Total
Bytes

Input(n) Parameter Definitions

Parameter values are in Decimal format.

Configured Function from CTRIO Workbench	Parameter 1 Contents DWORD	Parameter 2 Contents DWORD
Scaled Counter with Capture	Scaled Value (pos. or rate)	Scaled Captured Value
Non-scaled Timer	Previous Time (μs)	In Progress Time (μs)
Scaled Timer	Scaled Interval (rate)	Scaled In Progress Time (μs)

Input(n) Parameter Edge Timer and Dual Edge Timer

Parameters 1 and 2 are explained and will be mapped to V2000 - V2003 in this example.

Standard Timers:

When the Enable Timer Capture bit is ON and the configured input edge occurs, the CTRIO(2) module will begin timing. The Timer Capture Starting bit will be ON while the timing is in progress and will turn OFF when the next configured input edge occurs and the Timer Capture Complete bit turns ON. The program will need to turn off the Enable Timer Capture bit, and confirm the Timer Capture Starting and Timer Capture Complete bits reset before attempting the next time capture cycle. Turning OFF the Enable Timer Capture bit resets the timers register values to zero.

Free Run Timers:

If the Free Run Timer option was configured, the Enable Timer Capture bit is not available. When the configured input edge occurs, the CTRIO(2) module will begin timing. The Timer Capture Starting bit will be ON while the timing is in progress and will turn OFF when the next configured input edge occurs. When this edge occurs, the Timer “in progress time” register resets to zero. The “previous time” register will always retain the most recent captured time value.

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	PLCStatus Inputs Base Addr = V2000 (Control Relay) D2-240	Description
Parameter 1	N/A	V2001-V2000	N/A	V2001-V2000	Previous Time
Parameter 2		V2003-V2002		V2003-V2002	In Progress Time
Timer Capture Starting		V2020.0		C160	On when Time Capture is in progress
Enable Timer Capture	V2054.0	N/A	C260	N/A	Turn ON to enable Timer Capture Function (Not available when Free Run Timer option is selected)
Timer Capture Complete	N/A	V2020.1	N/A	C161	On when Timing is completed
Timer Timeout Bit		V2020.2		C162	Set when Timeout Occurs

Edge and Dual Edge Timer Timeout Function

The Timer Timeout Function is available for use with standard and Free Run Timers. It is primarily used in Free Run timing of recurring events (rate, velocity calculations, etc.). The specified Timeout Period is in effect once the timer is enabled until receiving the first configured input edge. Then it is in effect until receiving the next edge of the timing input to complete the timing cycle.

Standard Timers:

Once the timer is enabled, the Timeout Bit is set if the time that it takes the CTRIO(2) module to see the configured input edge exceeds the specified Timeout Period. The program will need to turn off the Enable Timer Capture bit, and confirm the Timer Capture Complete bit and Timeout bit resets before attempting the next time capture cycle.

Once timing has been initiated, if the time before the CTRIO(2) module sees the next configured edge exceeds the specified Timeout Period, the Timeout bit is set. The Timer register values are reset to zero. The program will need to turn off the Enable Timer Capture bit, and confirm the Timer Capture Starting bit, Timer Capture Complete bit and the Timeout bit reset before attempting the next time capture cycle.

Free Run Timers:

The Timeout Bit is set if the time that it takes the CTRIO(2) module to see the configured input edge exceeds the specified Timeout Period. The Timeout bit resets when the next timing cycle begins. The “Previous Time” register value is reset to zero.

Once timing has been initiated, if the time before the CTRIO(2) module sees the next configured edge exceeds the specified Timeout Period, the Timeout bit is set. The Timer register values are reset to zero. The Timeout bit resets when the next timing cycle begins.



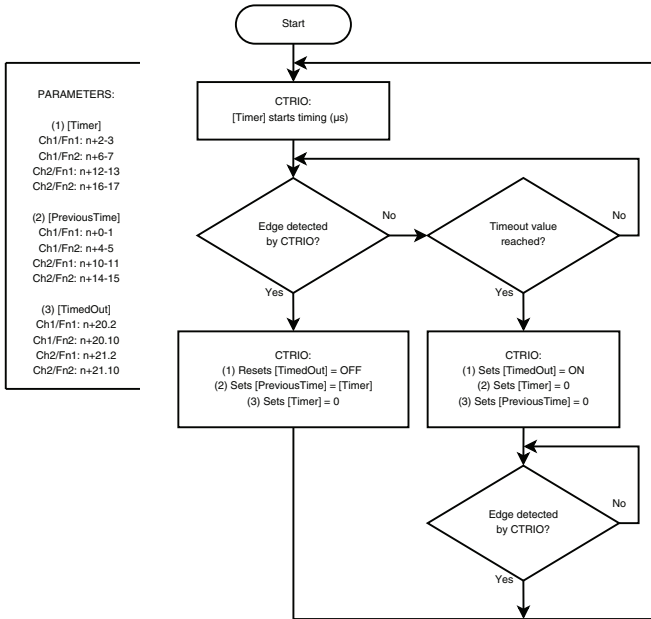
NOTE: For *DirectSOFT* users: the *I/O Map* dialog displays the exact memory locations in use by the CTRIO(2) module. Within the *I/O Map* dialog you can print out a report of memory locations in use.

Input Function Status Bit Definitions

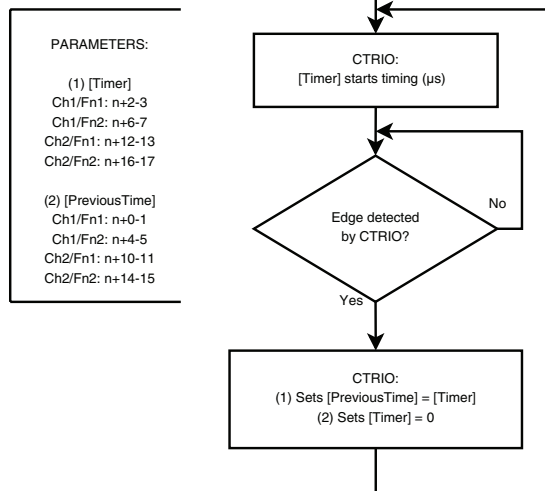
Input function offsets are listed in the order of Ch1/Fn1, Ch1/Fn2, Ch2/Fn1, Ch2/Fn2

Ch(x)/Fn(x) Status Bits (Transfers from CTRIO(2) to CPU)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets <i>DirectLOGIC</i> PLCs
Count Capture Complete Bit	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8
Timer Capture Start	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8
Timer Capture Complete (Timing) OR At Reset Value (Counting)	1, 9, 17, 25	20.1, 20.9, 21.1, 21.9
Timer “Timed Out” Bit	2, 10, 18, 26	20.2, 20.10, 21.2, 21.10
Pulse Catch Output Pulse State	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8
Pulse Catch Start	1, 9, 17, 25	20.1, 20.9, 21.1, 21.9

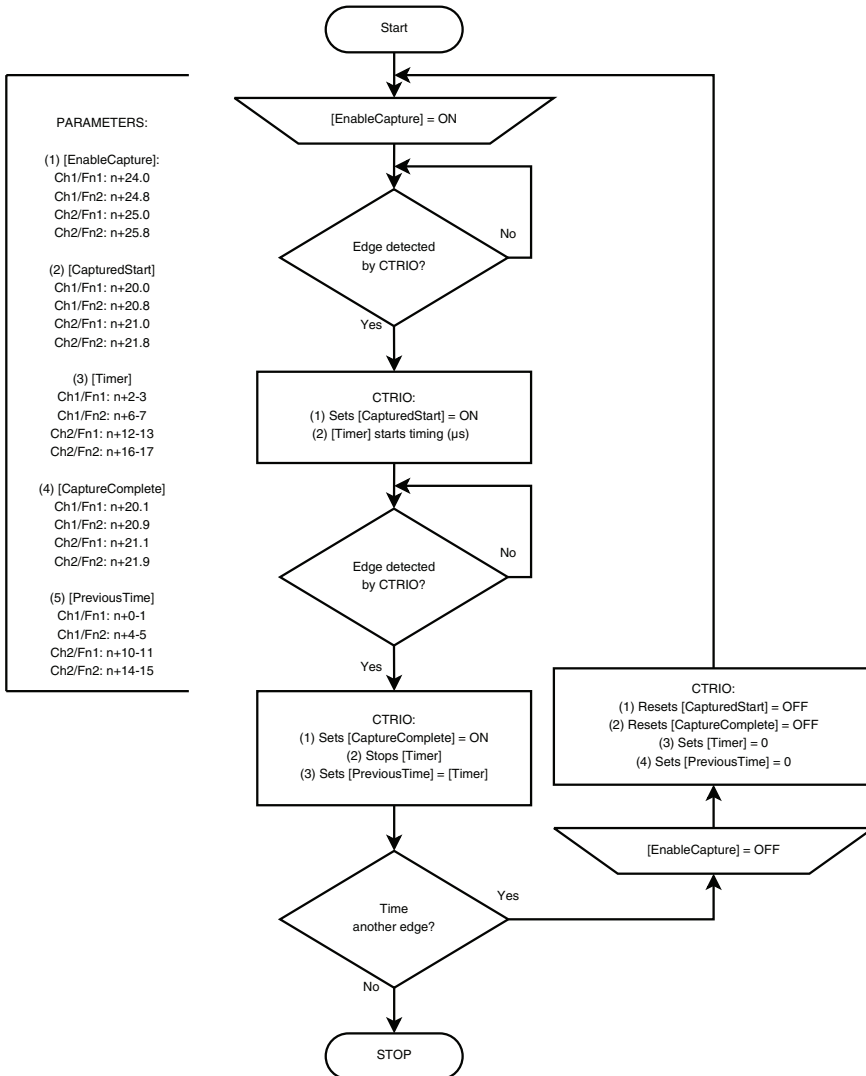
Edge Timer Function (Free Run, Timeout) (IBoxes)



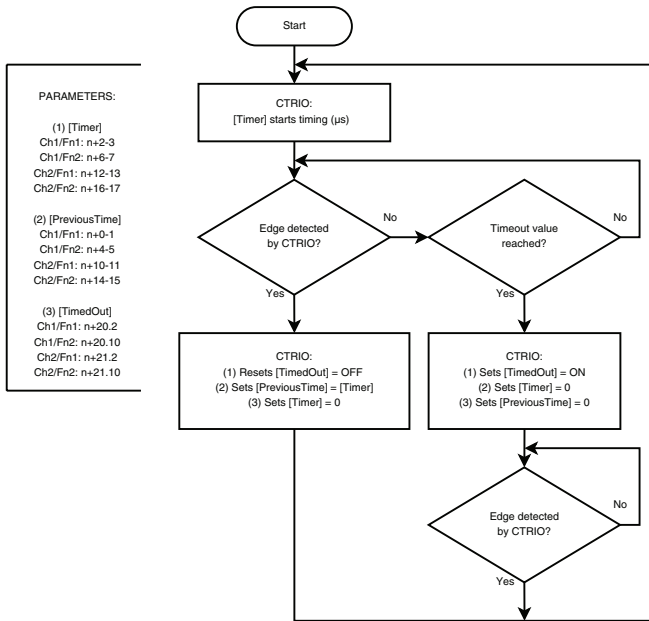
Edge Timer Function (Free Run, No Timeout) (IBoxes)



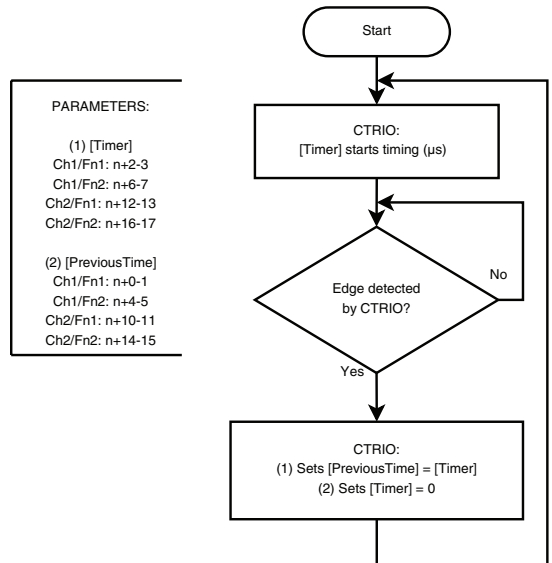
**Edge Timer Function
(Manual, No Timeout)
(IBoxes)**



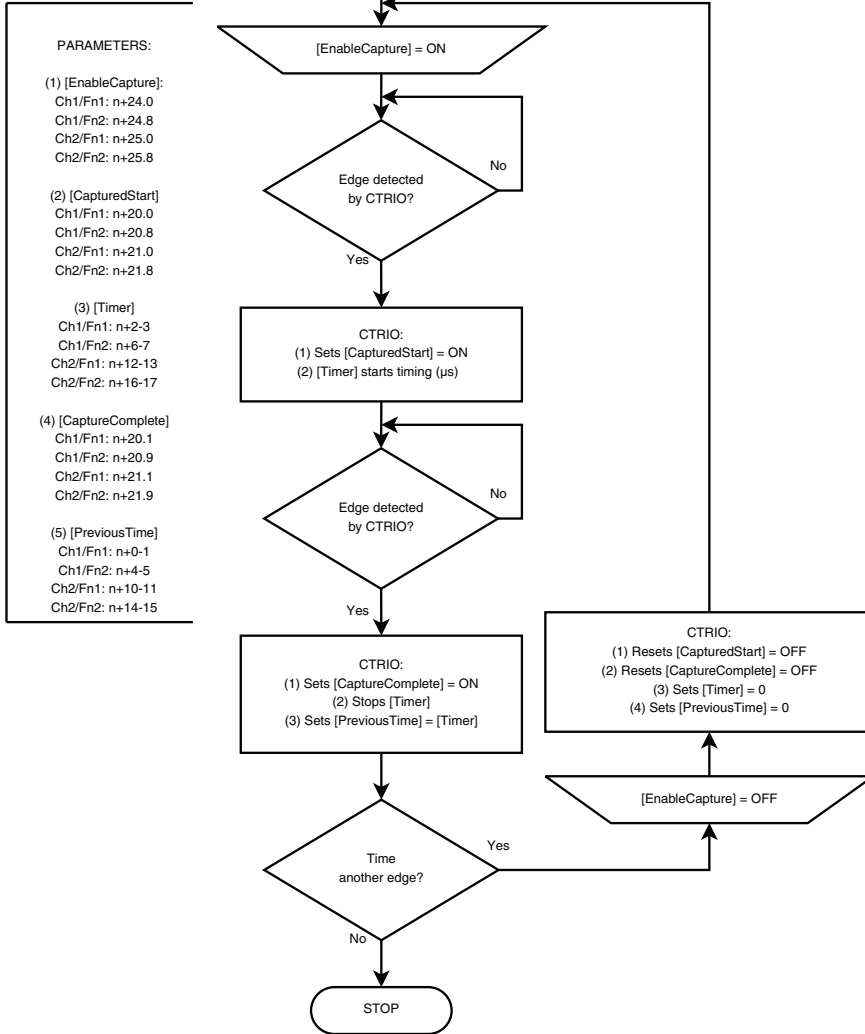
**Edge Timer Function
(Free Run, Timeout)
(DL-PLC)**



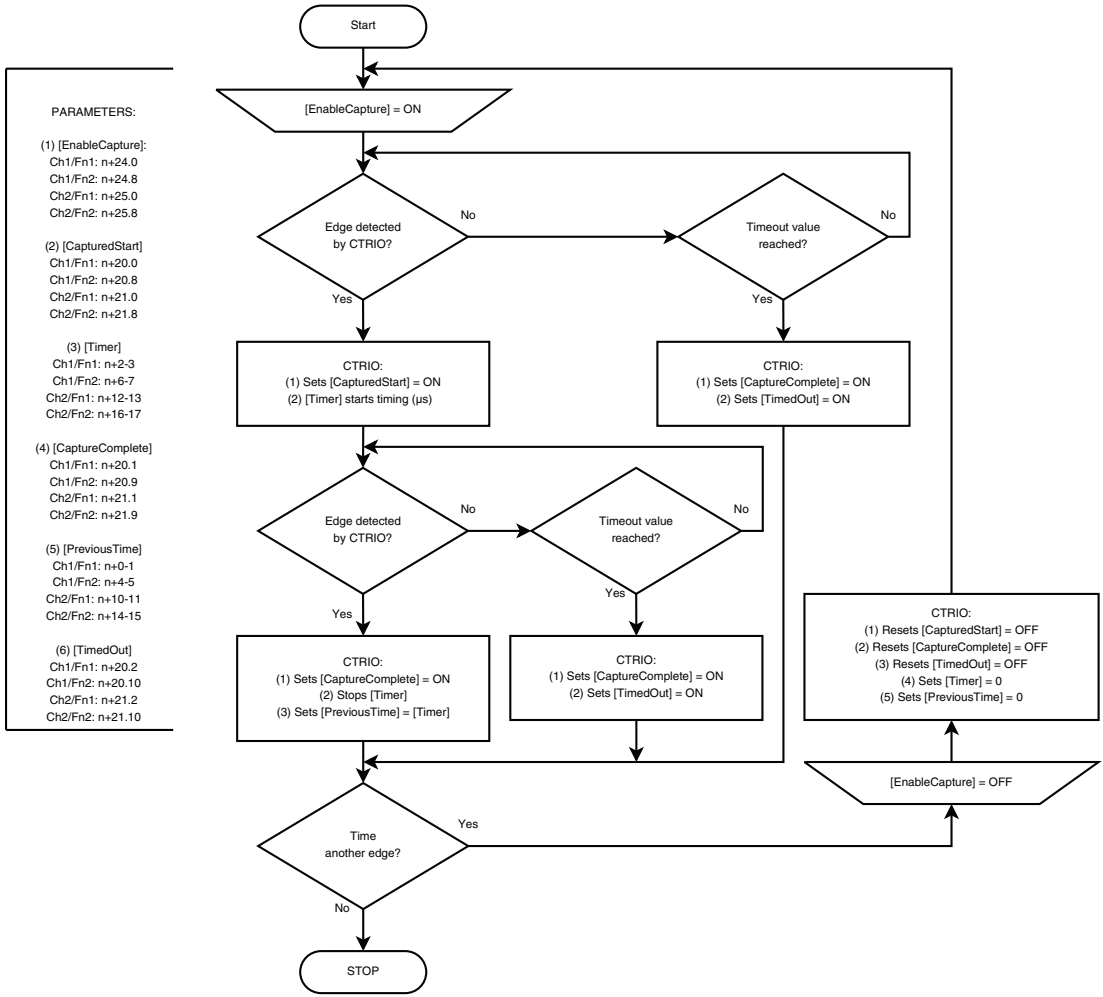
**Edge Timer Function
(Free Run, No Timeout)
(DL-PLC)**



**Edge Timer Function
(Manual, No Timeout)
(DL-PLC)**



**Edge Timer Function
(Manual, Timeout)
(DL-PLC)**



Inhibit

- DL
- Win
- NI

Inhibit allows an input of the CTRIO(2) to pause the counting of Function 1, a Quad Counter or simple Counter. To use this function, Input A must be configured as a Counter or Quad Counter. An input must be set as a counter before Inhibit Fn 1 will appear in the menu in CTRIO Workbench.

The following table shows which memory locations are used for memory transfers from the CTRIO(2) module to the CPU. The starting memory location is defined by the user in the I/O Map within CTRIO Workbench. When configuring the *DirectLOGIC* CPU, use the memory address offsets in the second column. When configuring an H2-WinPLC, EBC, PBC, MODBUS, or DEVNETS placed in the CPU slot, use the non-PLC offsets in column one.

Data Type and Offset WinPLC, EBC, PBC, DEVNETS, MODBUS	Address for Inputs (<i>DirectLOGIC</i>)	Definition	Format	Bytes
dwX0	n+0	Ch 1/Fn 1 Parameter 1	DWord	4
dwX1	n+2	Ch 1/Fn 1 Parameter 2		
dwX2	n+4	Ch 1/Fn 2 Parameter 1		
dwX3	n+6	Ch 1/Fn 2 Parameter 2		
dwX4	n+10	Ch 2/Fn 1 Parameter 1		
dwX5	n+12	Ch 2/Fn 1 Parameter 2		
dwX6	n+14	Ch 2/Fn 2 Parameter 1		
dwX7	n+16	Ch 2/Fn 2 Parameter 2		
bX0...7 bX8...15	n+20	Ch 1/Fn 1 Status (Low Byte) Ch 1/Fn 2 Status (High Byte)	Word	2
bX16...23 bX24...31	n+21	Ch 2/Fn 1 Status (Low Byte) Ch 2/Fn 2 Status (High Byte)		
bX32...39 bX40...47	n+22	Output 0 Status (Low Byte) Output 1 Status (High Byte)		
bX48...55 bX56...63	n+23	Output 2 Status (Low Byte) Output 3 Status (High Byte)		
bX64...71 bX72...79 bX80...87 bX88...95	n+24	System Functions Read/Write CTRIO(2) Internal Registers (see p. 6-10 for bit definitions)	DWord	4

**44 Total
Bytes**

Input (n) Parameter Definitions

Parameter values are in Decimal format.

Configured Function from CTRIO Workbench	Parameter 1 Contents DWORD	Parameter 2 Contents DWORD
Non-scaled Counter	Raw Input Value	Not Used
Scaled Counter	Scaled Value (pos. or rate)	Raw Value
Non-scaled Counter with Capture	Raw Value	Captured Value
Scaled Counter with Capture	Scaled Value (pos. or rate)	Scaled Captured Value
Non-scaled Timer	Previous Time (μ s)	In Progress Time (μ s)
Scaled Timer	Scaled Interval (rate)	Scaled In Progress Time (μ s)
Pulse Catch	Not Used	Not Used



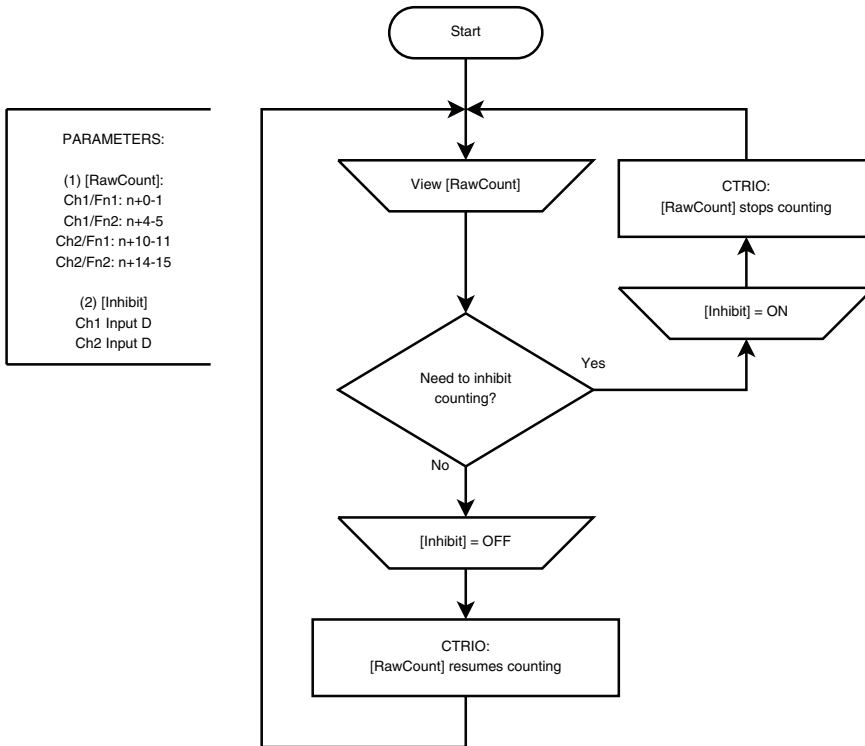
NOTE: For *DirectSOFT* users: the *I/O Map* dialog displays the exact memory locations in use by the CTRIO(2) module. Within the *I/O Map* dialog you can print out a report of memory locations in use.

Input Function Status Bit Definitions

Input function offsets are listed in the order of Ch1/Fn1, Ch1/Fn2, Ch2/Fn1, Ch2/Fn2

Ch(x)/Fn(x) Status Bits (Transfers from CTRIO(2) to CPU)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets <i>DirectLOGIC</i> PLCs
Count Capture Complete Bit	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8
Timer Capture Start	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8
Timer Capture Complete (Timing) OR At Reset Value (Counting)	1, 9, 17, 25	20.1, 20.9, 21.1, 21.9
Timer "Timed Out" Bit	2, 10, 18, 26	20.2, 20.10, 21.2, 21.10
Pulse Catch Output Pulse State	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8
Pulse Catch Start	1, 9, 17, 25	20.1, 20.9, 21.1, 21.9

Inhibit Function (IBoxes & DL-PLC)



Input Memory Map for Pulse Catch Data Transfers



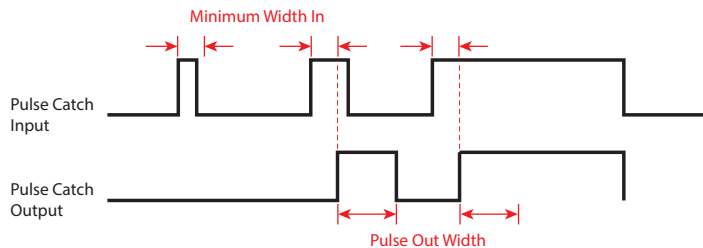
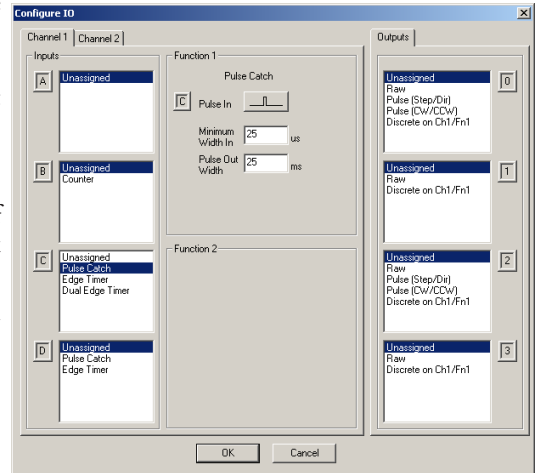
The Pulse Catch function allows a very short duration pulse to be qualified and lengthened to a time period long enough to guarantee that it is seen by the CPU. CPU scans necessarily vary with the length and complexity of the user's program. A scan frequency of several milliseconds is possible. On a conventional discrete input module, if a pulse is shorter than one scan time, there is a chance the CPU will not see the pulse.

The CTRIO(2) module's Pulse Catch function sees the fast incoming signal and holds its status in a status bit until the CPU can see it. A discrete output(s) can also be tied to follow the Pulse Catch.

Notice that the module's four input terminals are represented by the A, B, C, and D boxes on the left side of this dialog. When wiring an input to terminal 1C, select the Channel 1 tab near the top of this window and click Pulse Catch in box C.

Three selections must be made in conjunction with the Pulse Catch option.

1. Choose the rising edge of the pulse or the falling edge of the pulse. This is a critical decision. Careful attention should be paid to the type of output the field device generates. If the signal voltage is normally low, but a short duration pulse sends the signal to the ON state, you will want to trigger off the rising edge, and vice versa.
2. Enter the minimum pulse width to capture. Transients below this width will not be recorded. Set this value by typing the desired value in the "Minimum Width In" field.
3. Enter the minimum length of pulse the CTRIO(2) module should send in response to the input pulse. Make this setting by typing in the desired value in the "Pulse Out Width" field.



The following table shows which memory locations are used for memory transfers from the CTRIO(2) module to the CPU. The starting memory location is defined by the user in the I/O Map within CTRIO Workbench. When you are using the *Direct*LOGIC CPU, use the memory address offsets in the second column. When using an H2-WinPLC, EBC, PBC, MODBUS, or DEVNETS in the CPU slot, use the non-PLC offsets in column one.

Data Type and Offset WinPLC, EBC, PBC, DEVNETS, MODBUS	Address for Inputs (<i>Direct</i> LOGIC)	Definition	Format	Bytes
dwX0	n+0	Ch 1/Fn 1 Parameter 1	DWord	4
dwX1	n+2	Ch 1/Fn 1 Parameter 2		
dwX2	n+4	Ch 1/Fn 2 Parameter 1		
dwX3	n+6	Ch 1/Fn 2 Parameter 2		
dwX4	n+10	Ch 2/Fn 1 Parameter 1		
dwX5	n+12	Ch 2/Fn 1 Parameter 2		
dwX6	n+14	Ch 2/Fn 2 Parameter 1		
dwX7	n+16	Ch 2/Fn 2 Parameter 2		
bX0...7 bX8...15	n+20	Ch 1/Fn 1 Status (Low Byte) Ch 1/Fn 2 Status (High Byte)	Word	2
bX16...23 bX24...31	n+21	Ch 2/Fn 1 Status (Low Byte) Ch 2/Fn 2 Status (High Byte)		
bX32...39 bX40...47	n+22	Output 0 Status (Low Byte) Output 1 Status (High Byte)		
bX48...55 bX56...63	n+23	Output 2 Status (Low Byte) Output 3 Status (High Byte)		
bX64...71 bX72...79 bX80...87 bX88...95	n+24	System Functions Read/Write CTRIO(2) Internal Registers (see p. 6-10 for bit definitions)	DWord	4

**44 Total
Bytes**

Input (n) Parameter Definitions

Parameter values are in Decimal format.

Configured Function from CTRIO Workbench	Parameter 1 Contents DWORD	Parameter 2 Contents DWORD
Non-scaled Counter	Raw Input Value	Not Used
Scaled Counter	Scaled Value (pos. or rate)	Raw Value
Non-scaled Counter with Capture	Raw Value	Captured Value
Scaled Counter with Capture	Scaled Value (pos. or rate)	Scaled Captured Value
Non-scaled Timer	Previous Time (μs)	In Progress Time (μs)
Scaled Timer	Scaled Interval (rate)	Scaled In Progress Time (μs)
Pulse Catch	Not Used	Not Used



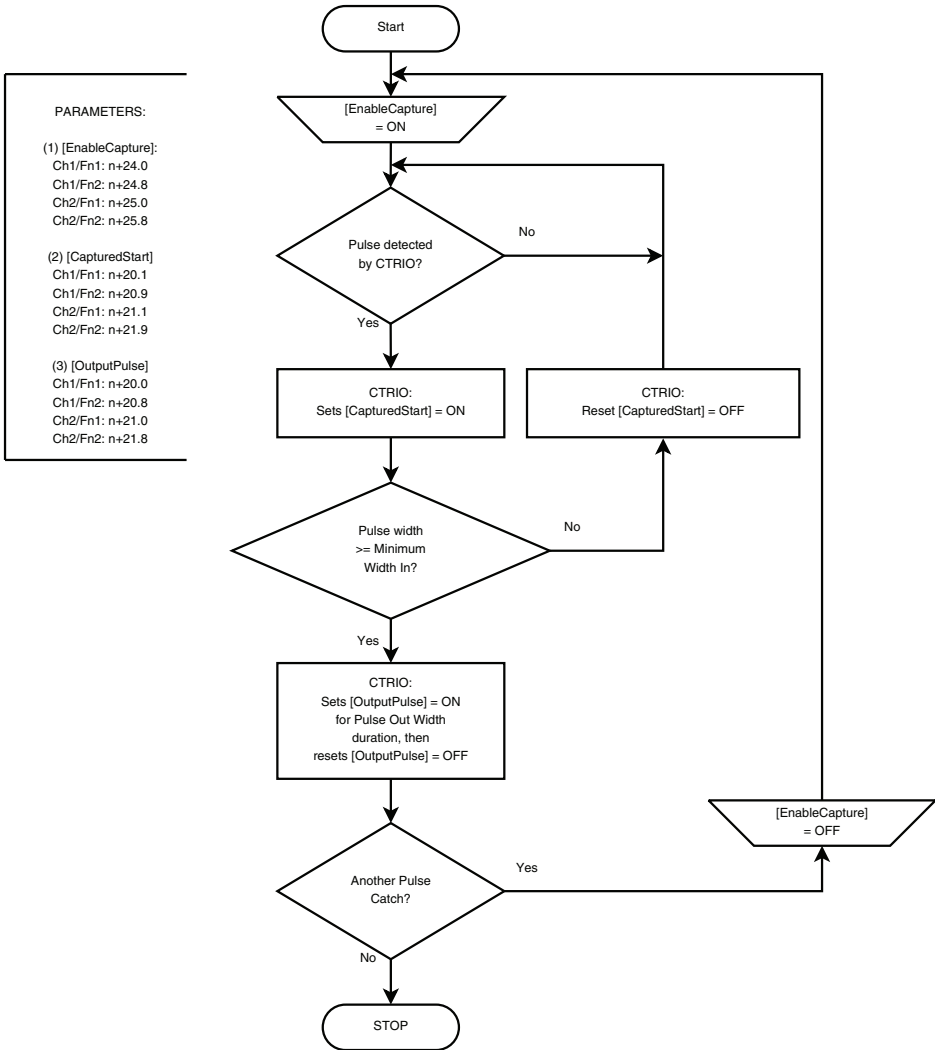
NOTE: For *DirectSOFT* users: the *I/O Map* dialog displays the exact memory locations in use by the *CTRIO(2)* module. Within the *I/O Map* dialog you can print out a report of memory locations in use.

Input Function Status Bit Definitions

Input function offsets are listed in the order of Ch1/Fn1, Ch1/Fn2, Ch2/Fn1, Ch2/Fn2

Ch(x)/Fn(x) Status Bits (Transfers from CTRIO(2) to CPU)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets <i>DirectLOGIC</i> PLCs
Pulse Catch Output Pulse State	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8
Pulse Catch Start	1, 9, 17, 25	20.1, 20.9, 21.1, 21.9

Pulse Catch Function (IBoxes & DL-PLC)



RUNTIME TABLE FUNCTIONS



In This Chapter...

Introduction to Runtime Table Functions.....	11-2
Preset Tables and Programmable Limit Switch (PLS) Tables.....	11-4
Load Preset Table	11-4
Load Table.....	11-8
Clear Preset Table	11-11
Create Preset Table (Initialize Table).....	11-14
Add Entry to Preset Table	11-17
CTRIO2 - <i>Direct</i> LOGIC Using IBoxes Flowcharts	11-18
CTRIO2 - <i>Direct</i> LOGIC PLC Flowcharts	11-19
Edit Preset Table Entry	11-20
Edit Preset Table Entry and Reload.....	11-23
Write File to ROM.....	11-26
Create Preset Table on Reset (Initialize Table on Reset)	11-29
Update Level (Edit Level Response).....	11-32

Introduction to Runtime Table Functions

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

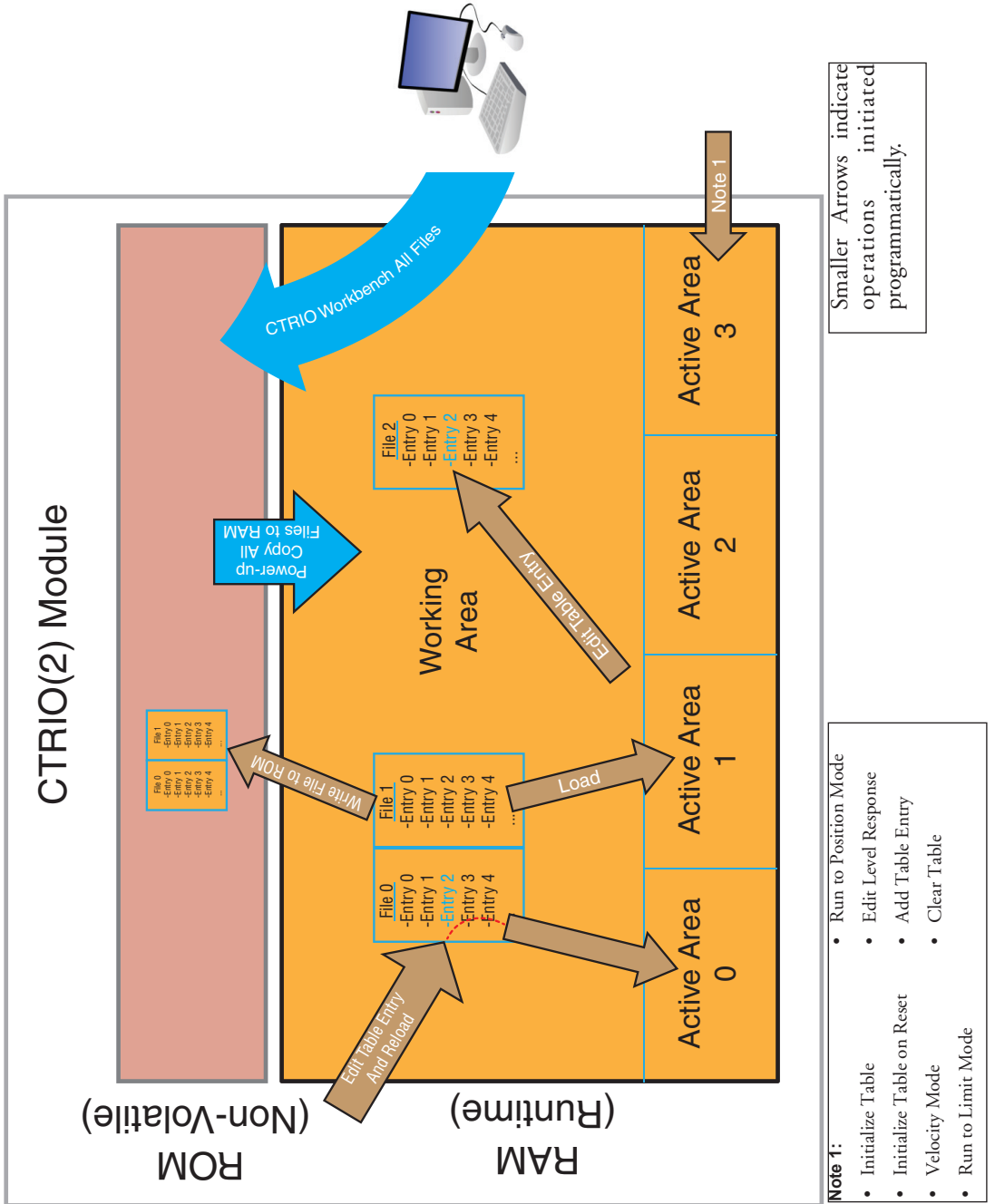
- Preset and PLS (Programmable Limit Switches) Tables such as Pulse Profiles, exist as Files in the CTRIO(2) module's memory.
- There are two areas of memory in a CTRIO(2) module:
 - ◊ ROM – this is the non-volatile (Flash) memory where files are stored to survive a power cycle
 - ◊ RAM – this is the 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.



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.

Load Preset Table, continued

- Set ProcessCmd ON to tell 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 whether the CmdError is ON.

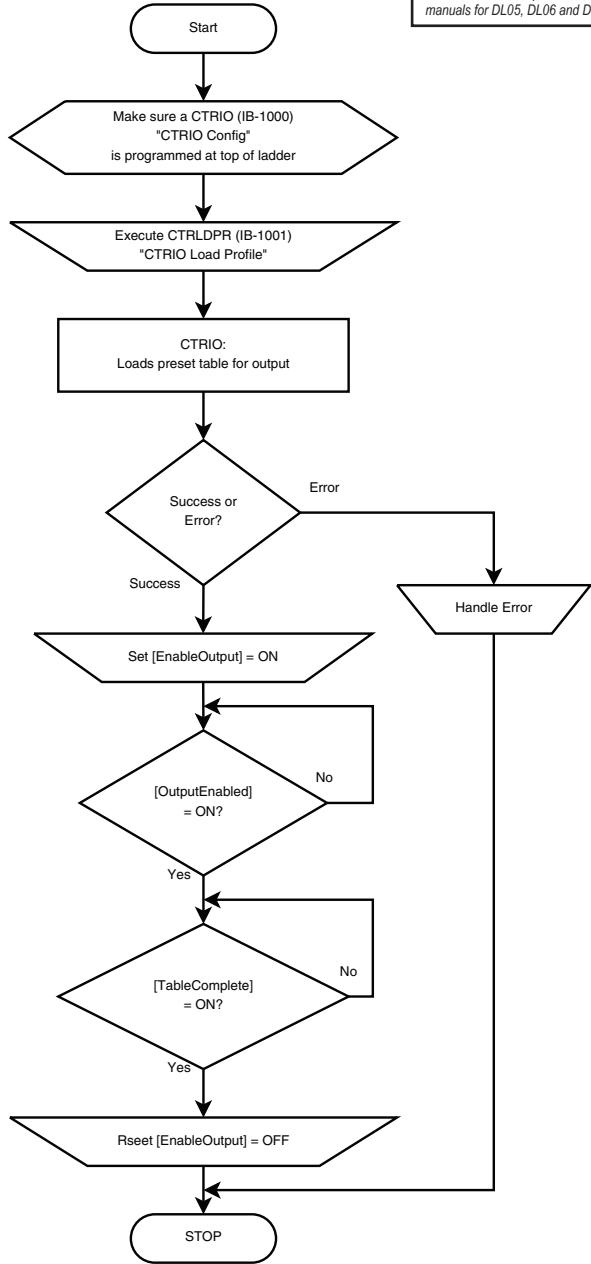
If it is ON, the CTRIO(2) module failed to load the Table. Write code to handle the error and finish here.

- 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.

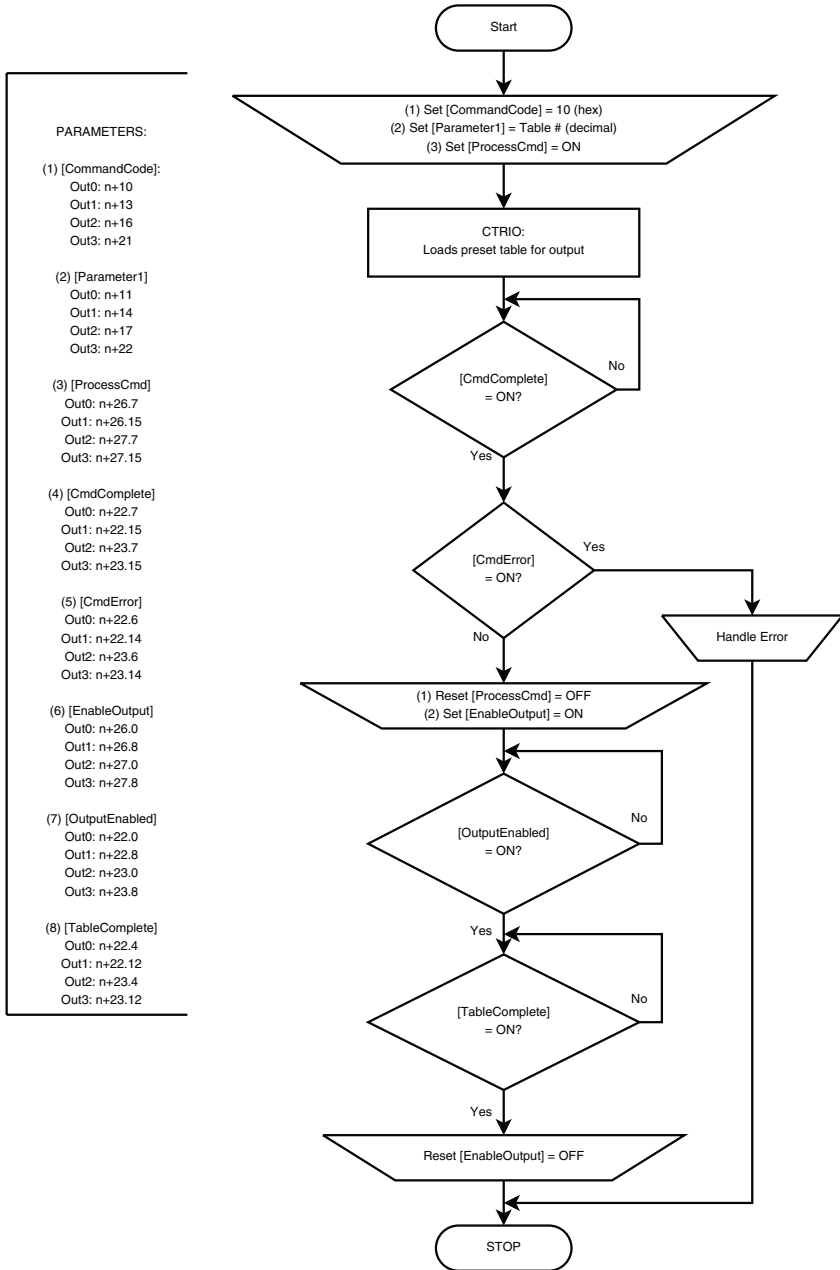
Load Preset Table & Run (IBoxes)

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

- PARAMETERS:**
- (1) [EnableOutput]
 - Out0: n+26.0
 - Out1: n+26.8
 - Out2: n+27.0
 - Out3: n+27.8
 - (2) [OutputEnabled]
 - Out0: n+22.0
 - Out1: n+22.8
 - Out2: n+23.0
 - Out3: n+23.8
 - (3) [TableComplete]
 - Out0: n+22.4
 - Out1: n+22.12
 - Out2: n+23.4
 - Out3: n+23.12



Load Preset Table & Run
(DL-PLC)



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.

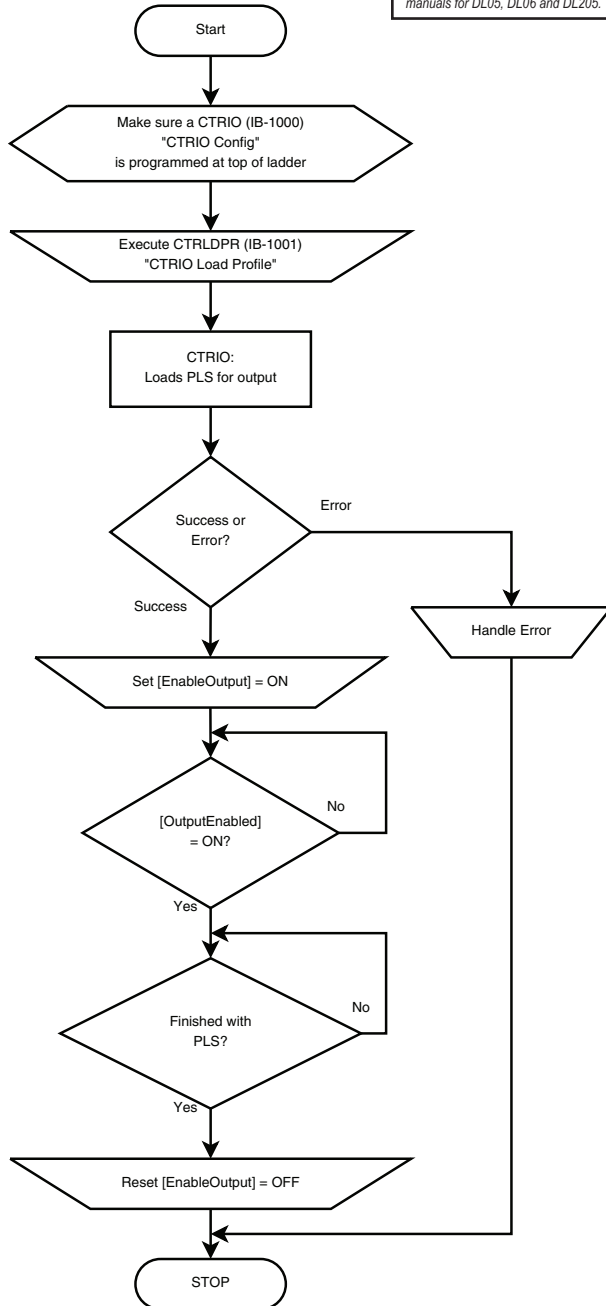
**Load PLS & Run
(IBoxes)**

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

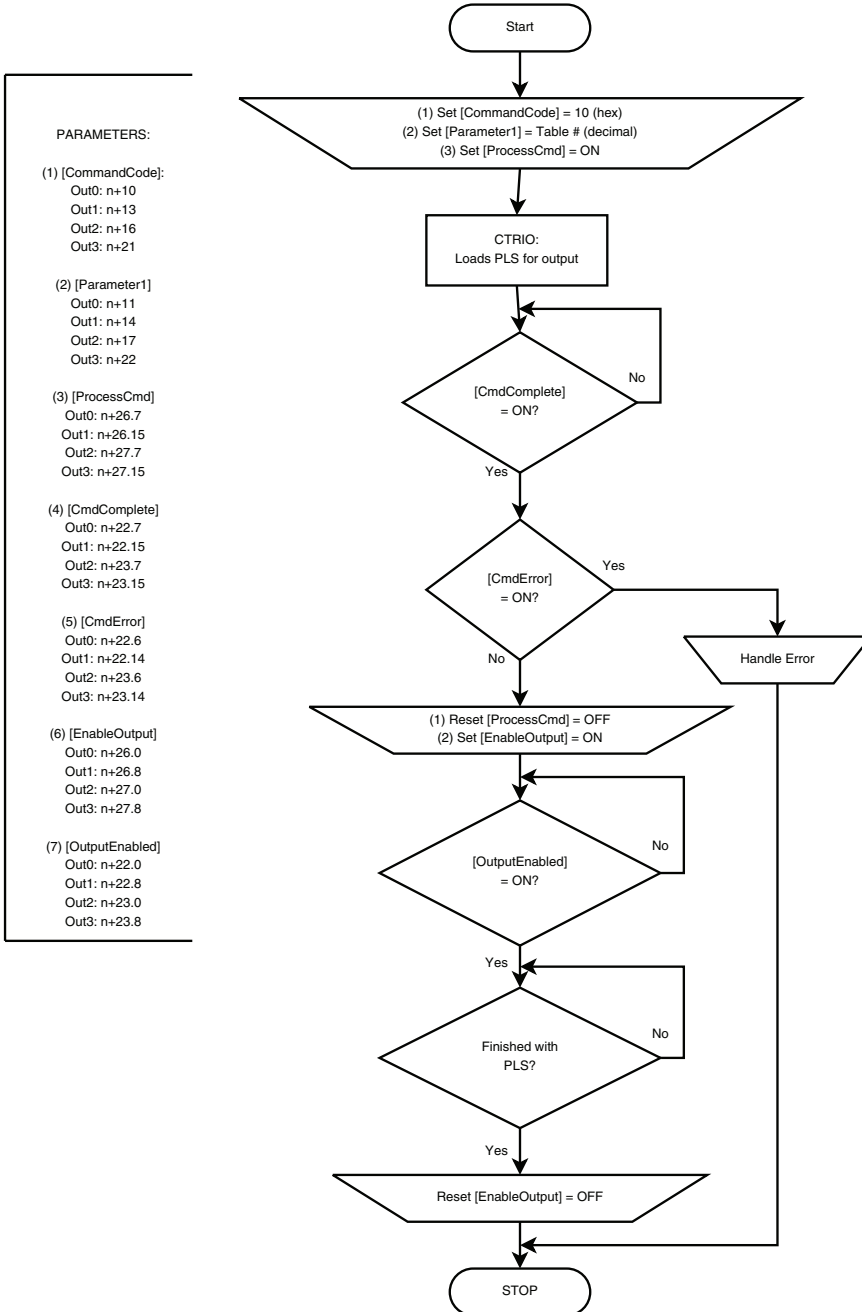
PARAMETERS:

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

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



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.

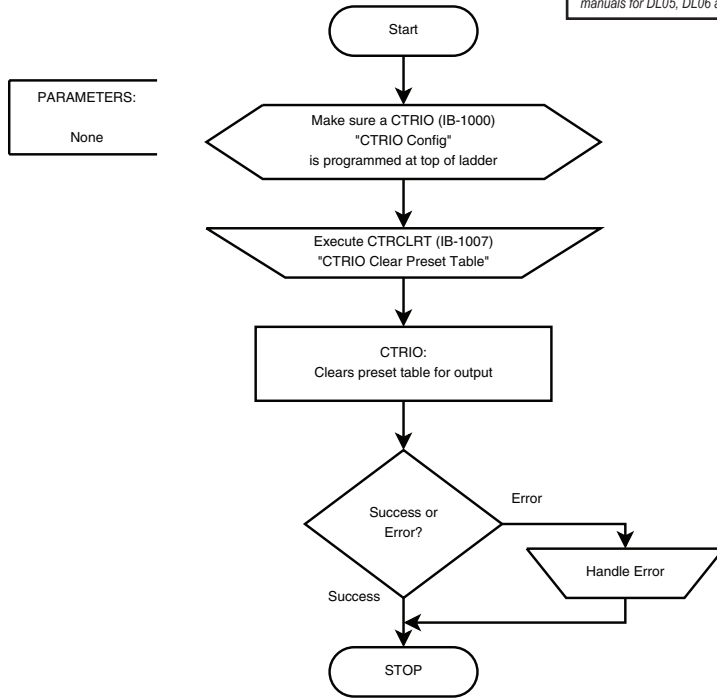
Once CmdComplete is seen high, your code should check whether the CmdError is ON.

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

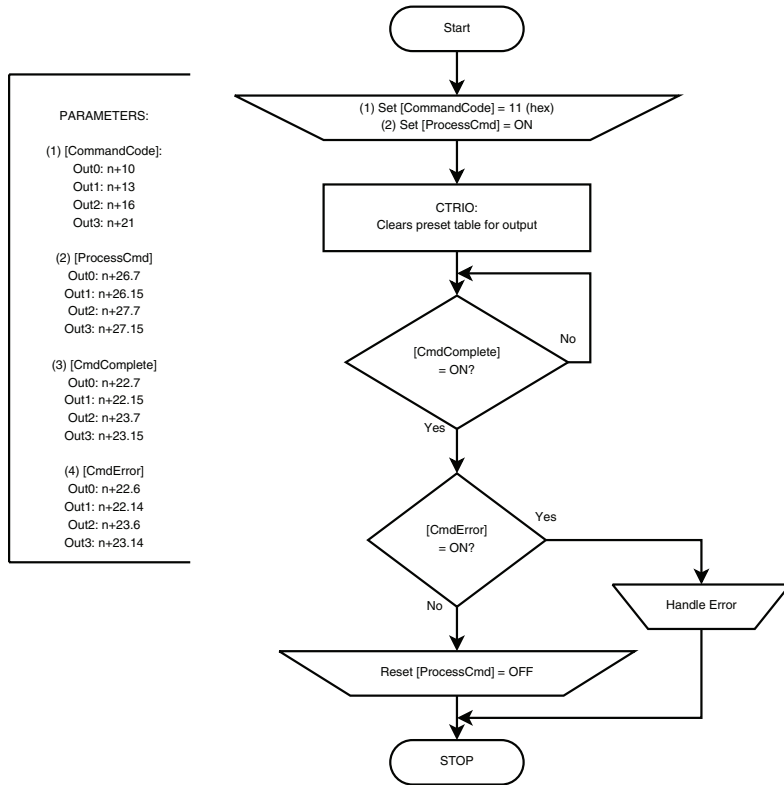
- If CmdError is OFF, the Active Area is clear.
- Reset ProcessCmd OFF.

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.



**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.

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.

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

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.

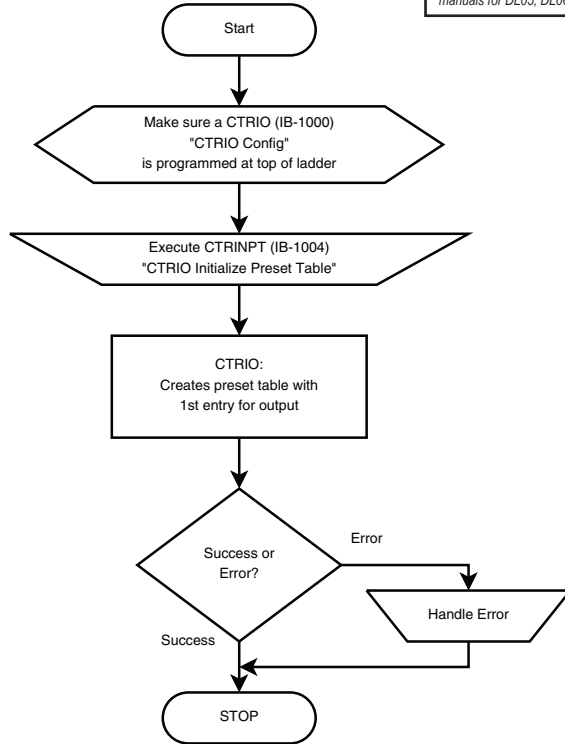
Once CmdComplete is seen high, your code should check the status of CmdError:

- 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.

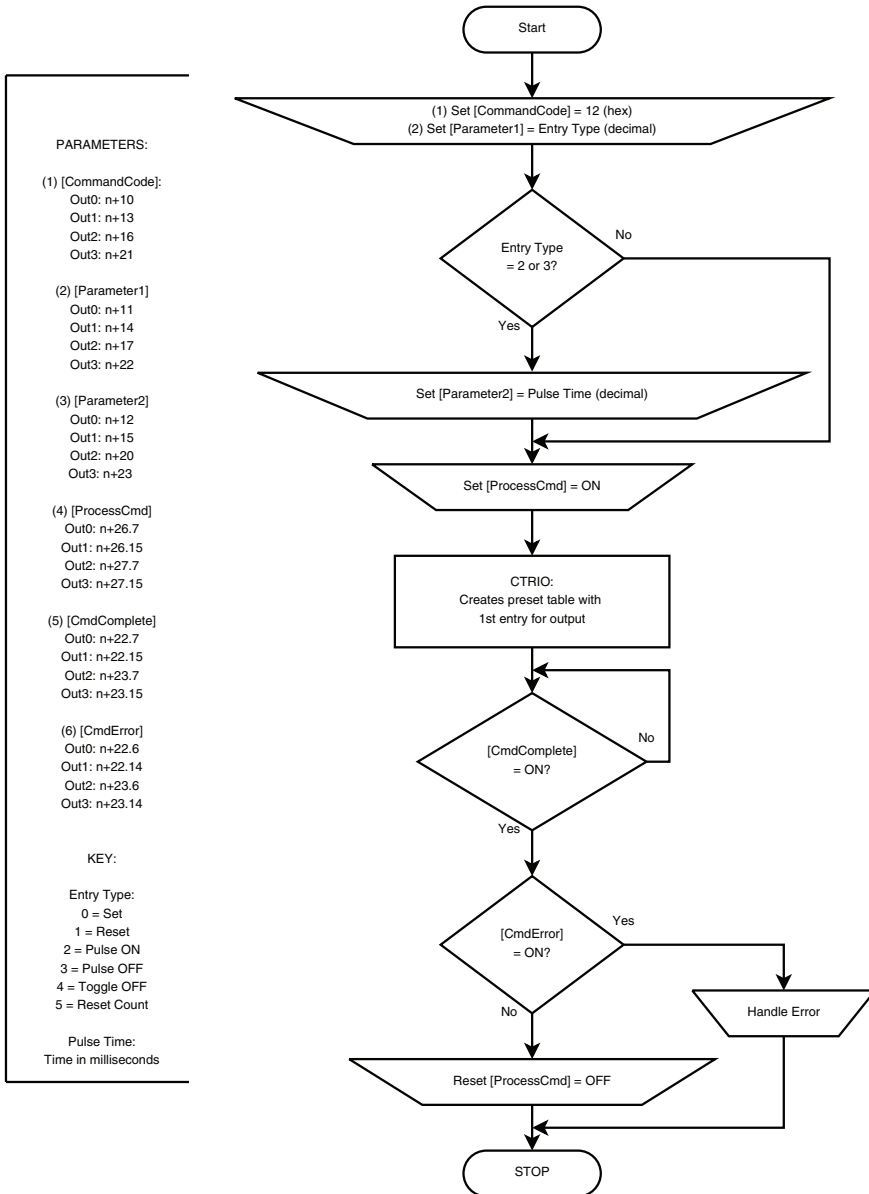
Create Preset Table (IBoxes)

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

PARAMETERS:
None



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.

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.

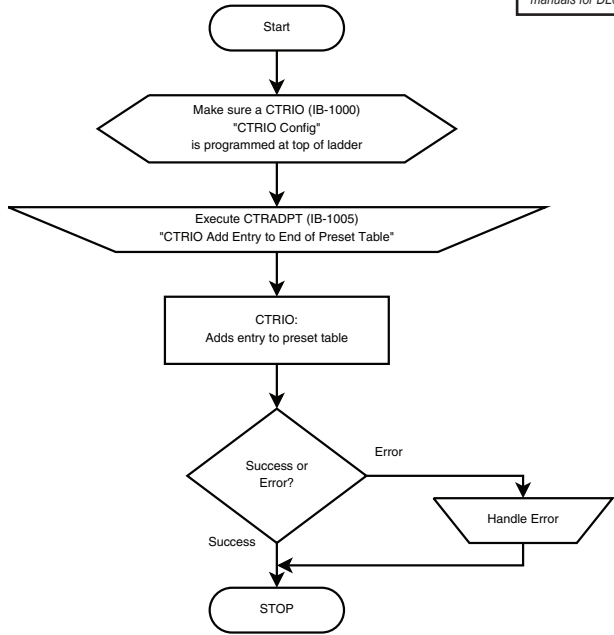
Parameter1	Function
0	Set
1	Reset
2	Pulse ON
3	Pulse OFF
4	Toggle OFF
5	Reset Count

CTRIO2 - DirectLOGIC Using IBoxes Flowcharts

Add Entry to Preset Table (IBoxes)

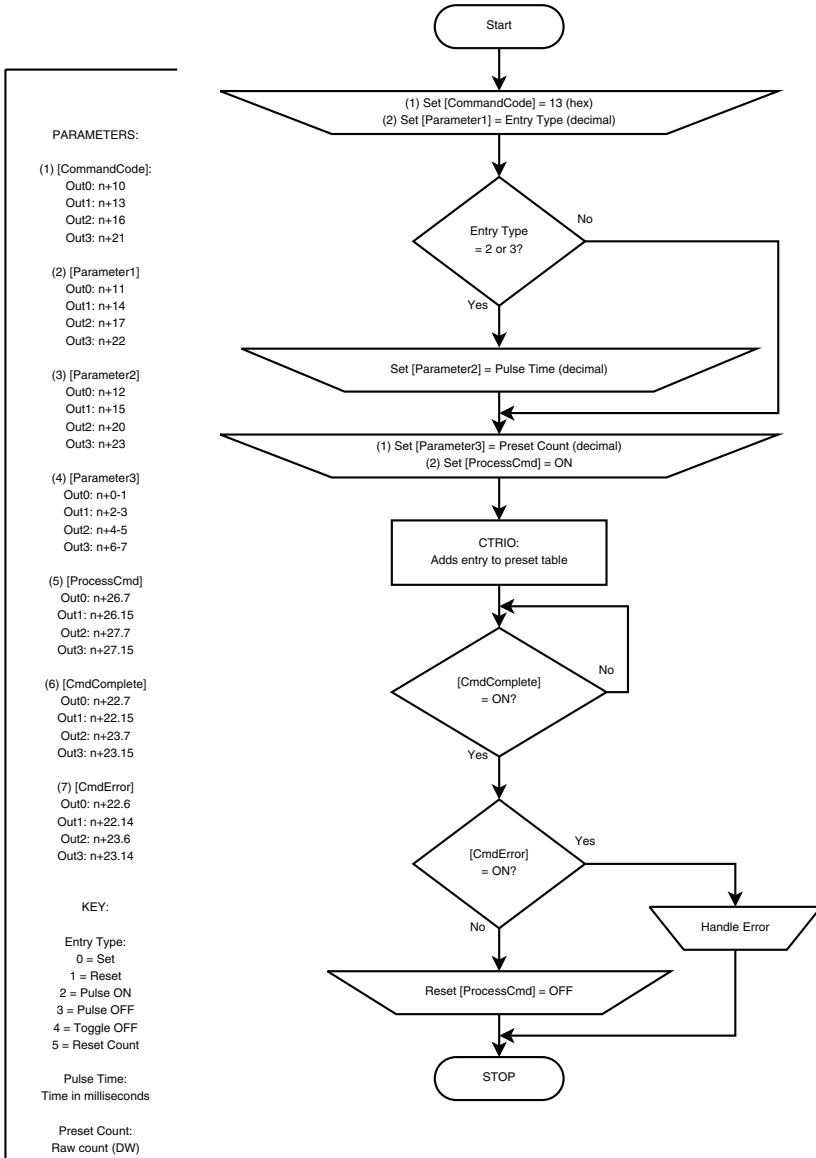
NOTE: An example program using these IBoxes can be found in Chapter 5 of the DirectLOGIC user manuals for DL05, DL06 and DL205.

PARAMETERS:
None



CTRIO2 - DirectLOGIC PLC Flowcharts

Add Entry to Preset Table (DL-PLC)



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:

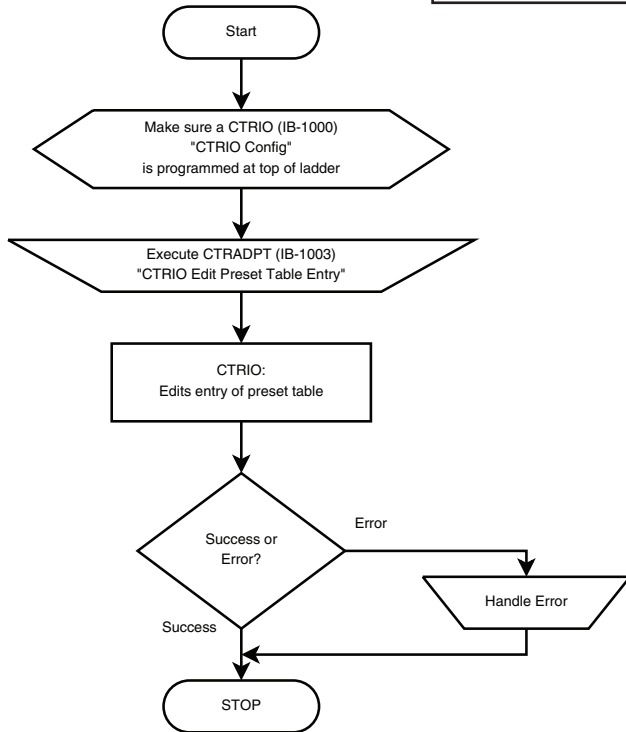
- If CmdError is ON, the CTRIO(2) module failed to edit the Table entry. Write code to handle the error and finish here.
- If CmdError is OFF, the updated Table is ready for use.
- Reset ProcessCmd OFF.

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
...	...

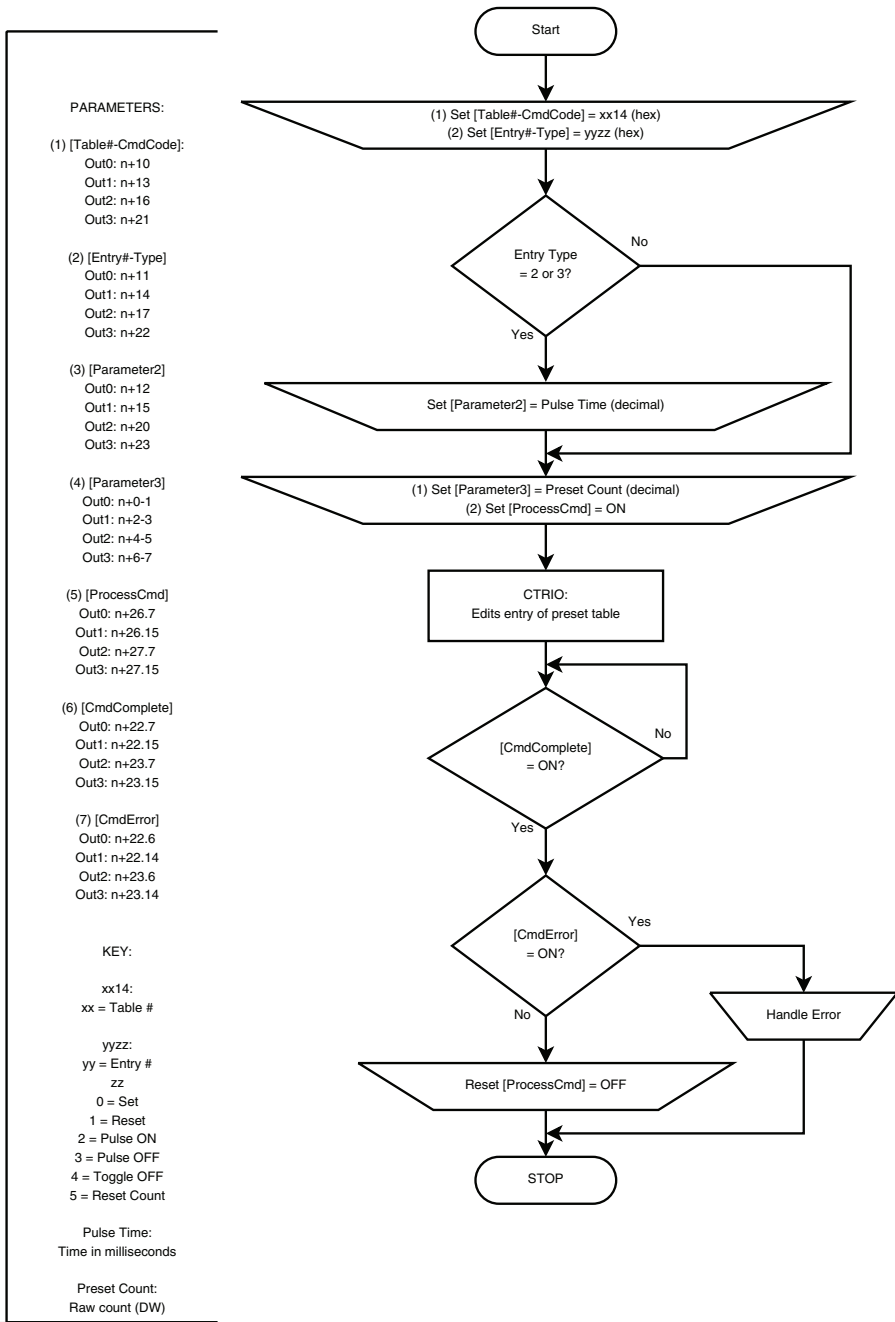
Edit Preset Table Entry (IBoxes)

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

PARAMETERS:
None



Edit Preset Table Entry (DL-PLC)



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.

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

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
...	...

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).

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.

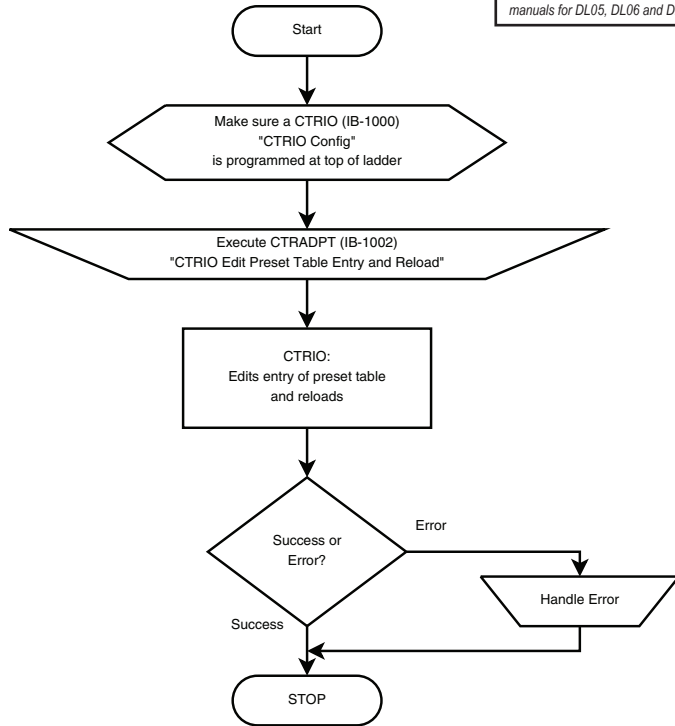
Reset ProcessCmd OFF.

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
...	...

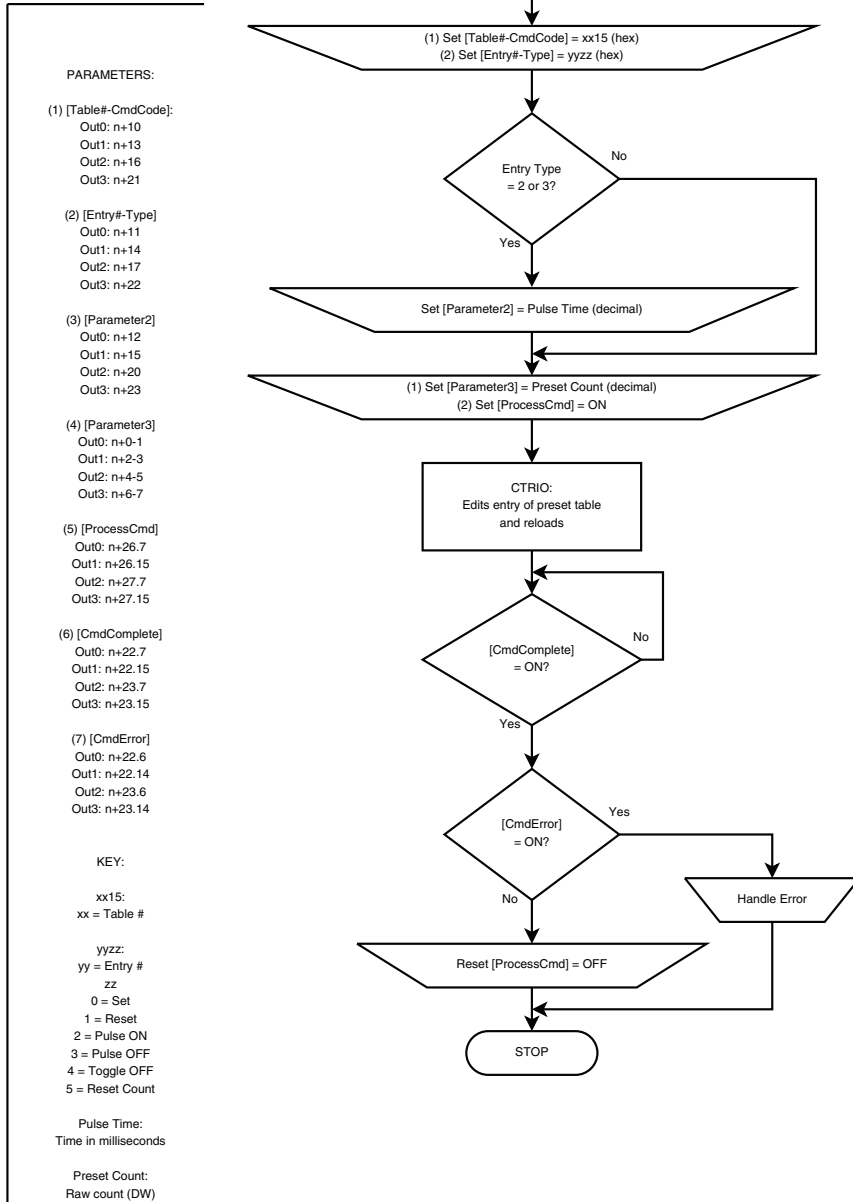
Edit Preset Table Entry & Reload (IBoxes)

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



**Edit Preset Table Entry & Reload
(DL-PLC)**



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.

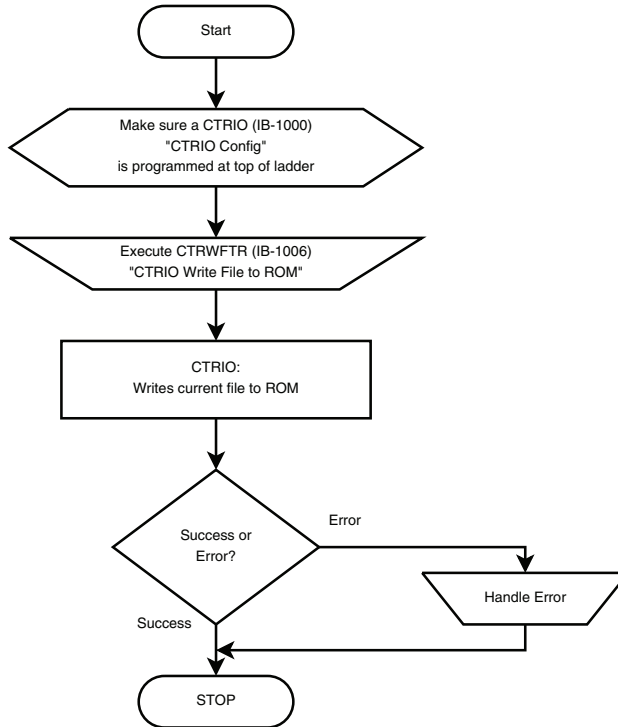
- Set the Command Code to 0x99.
- Set ProcessCmd ON to tell the CTRIO(2) to read in the CommandCode.
- The CTRIO(2) module begins the process of writing the file to ROM.

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

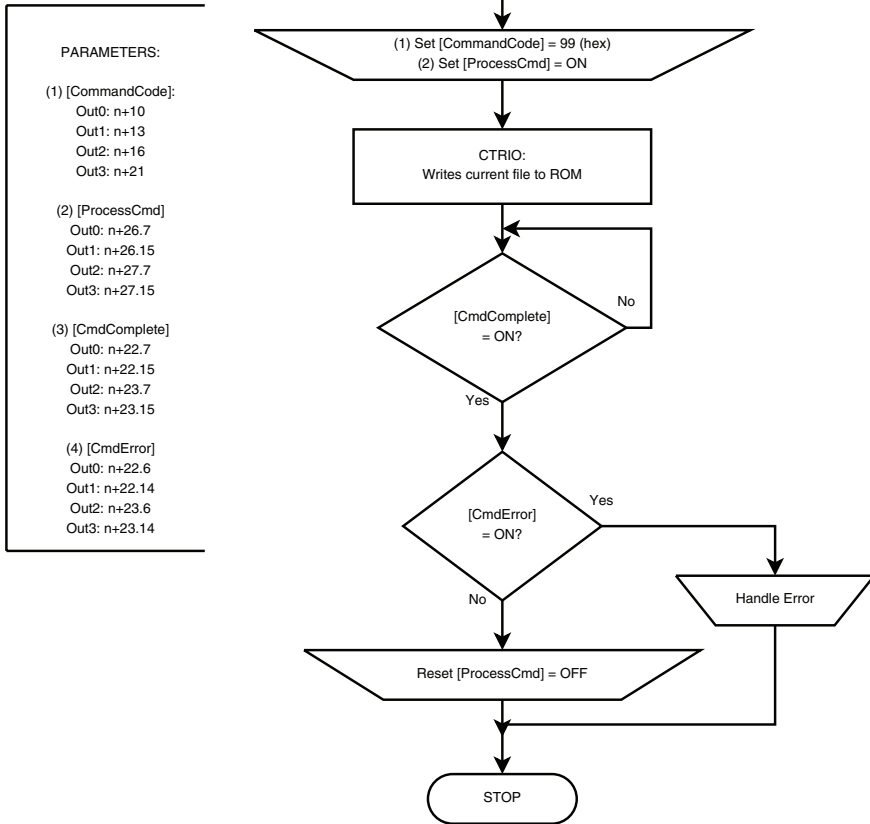
- If CmdError is ON, the CTRIO(2) module failed to write the file. Write code to handle the error and finish here.
- If CmdError is OFF, the file was written to ROM.
- Reset ProcessCmd OFF.

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.



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.

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.

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

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.

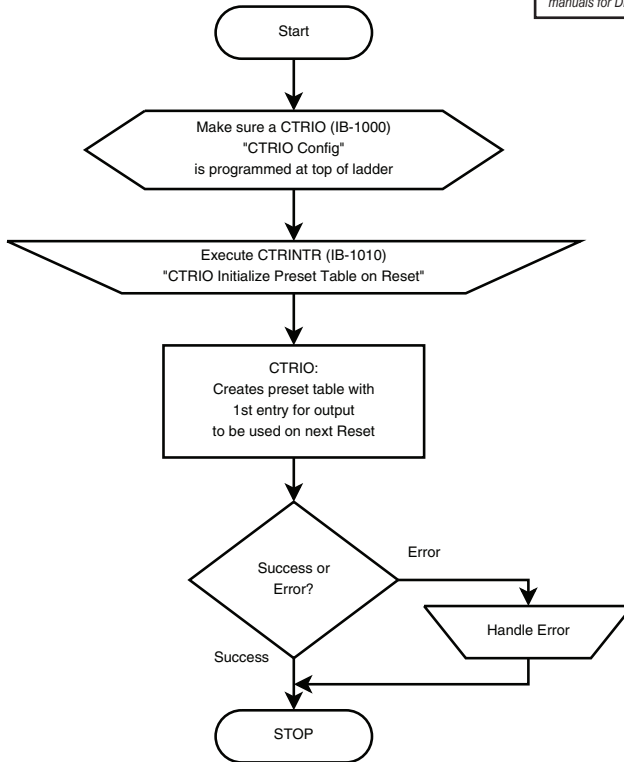
Once CmdComplete is seen high, your code should check the status of CmdError:

- 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.

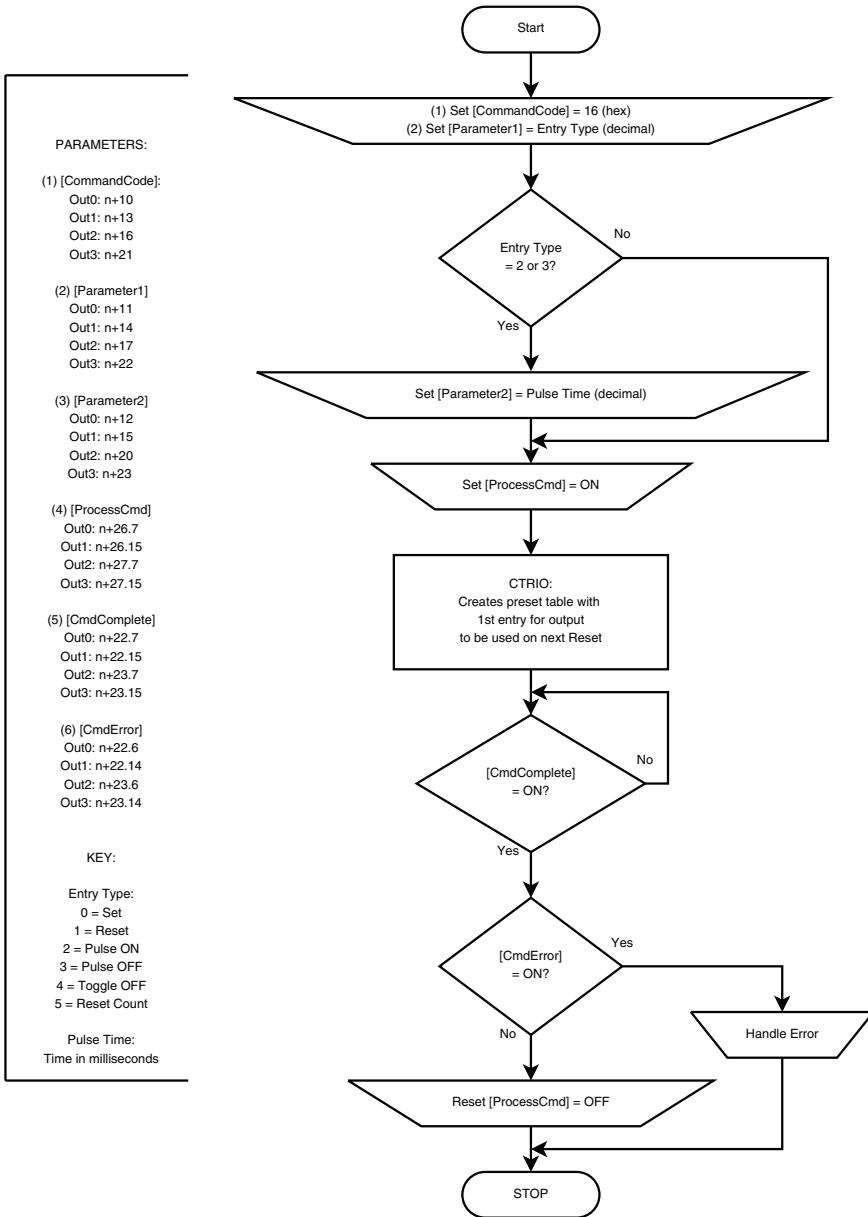
Create Preset Table on Reset (IBoxes)

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



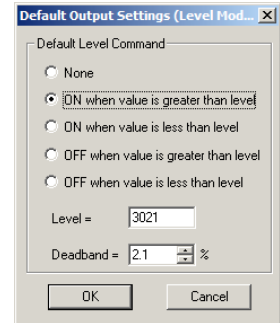
**Create Preset Table on Reset
(DL-PLC)**



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).



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

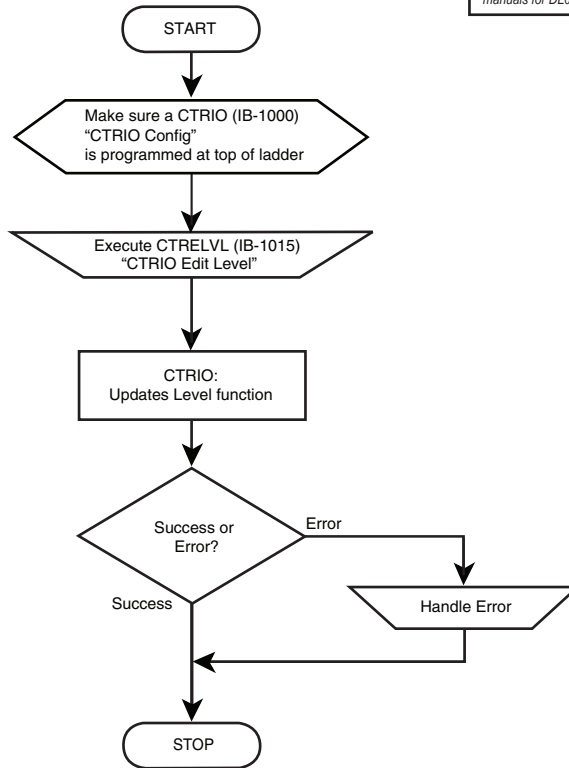
The CTRIO(2) module begins the process of changing the Level Response. When it's done, it sets the CmdComplete bit and will set or reset the CmdError bit accordingly.

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

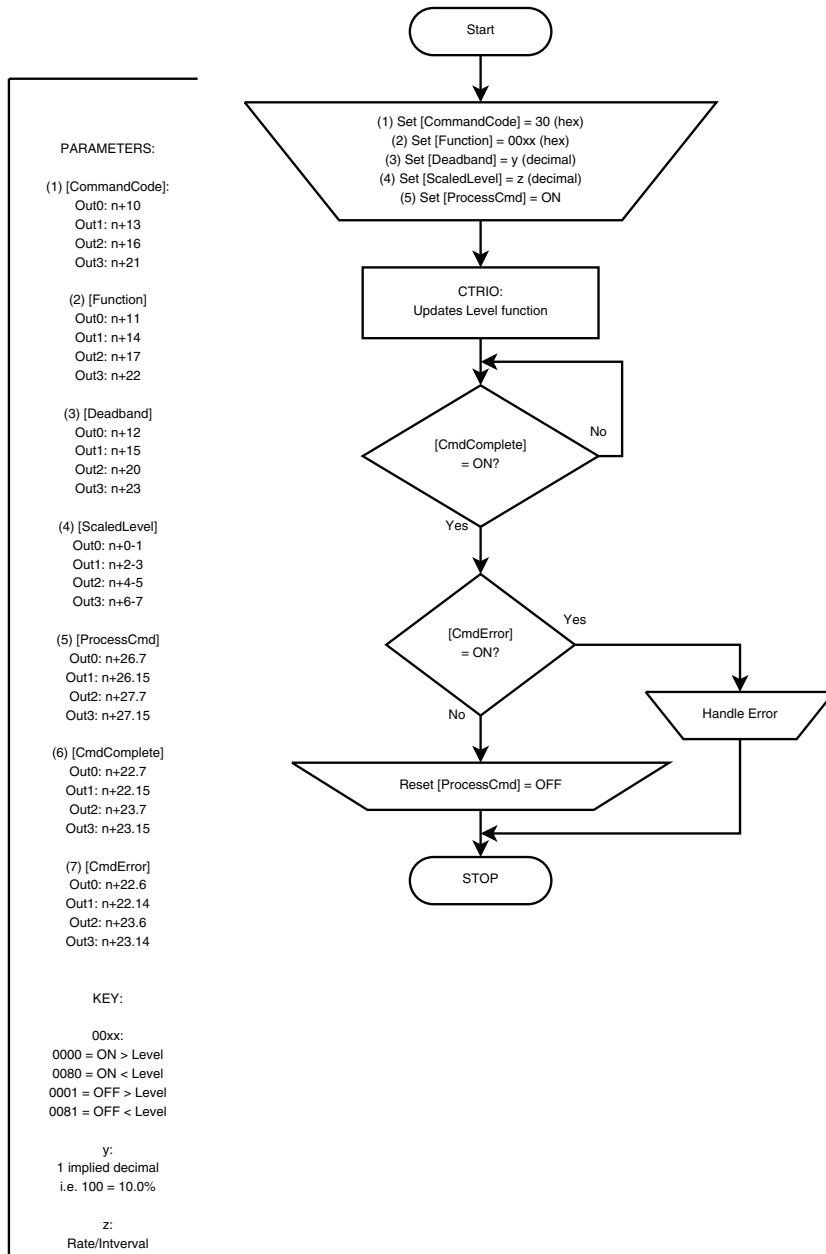
- If CmdError is ON, the CTRIO(2) module failed to update the table entry. Write code to handle the error and finish here.
- If CmdError is OFF, the entry was updated successfully.
- Reset ProcessCmd OFF.

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.



Update Level (DL-PLC)



MEMORY MAPPING



In This Appendix...

Input Memory Map for Data Transfers from CTRIO(2) to DL CPUs.....	A-2
Output Memory Map for Data Transfers from DL CPUs to CTRIO(2).....	A-4
Addressing Conventions (with V-memory Examples for DirectLOGIC PLCs).....	A-7
Input Function Status/Control Bits and Parameters	A-8

Input Memory Map for Data Transfers from CTRIO(2) to DL CPUs

- DL
- Win
- NI

The following table shows which memory locations are used for memory transfers from the CTRIO(2) module to the CPU. The starting memory location is defined by the user in the I/O Map within CTRIO Workbench. If you are using the *DirectLOGIC* CPU, you will use the memory address offsets in the second column. If using an H2-WinPLC, EBC, PBC, MODBUS, or DEVNETS in the CPU slot, you will use the non-PLC offsets in column one.

Data Type and Offset WinPLC, EBC, PBC, DEVNETS, MODBUS	Address for Inputs (<i>DirectLOGIC</i>)	Definition	Format	Bytes
dwX0	n+0	Ch 1/Fn 1 Parameter 1	DWord	4
dwX1	n+2	Ch 1/Fn 1 Parameter 2		
dwX2	n+4	Ch 1/Fn 2 Parameter 1		
dwX3	n+6	Ch 1/Fn 2 Parameter 2		
dwX4	n+10	Ch 2/Fn 1 Parameter 1		
dwX5	n+12	Ch 2/Fn 1 Parameter 2		
dwX6	n+14	Ch 2/Fn 2 Parameter 1		
dwX7	n+16	Ch 2/Fn 2 Parameter 2	Word	2
bX0...7 bX8...15	n+20	Ch 1/Fn 1 Status (Low Byte) Ch 1/Fn 2 Status (High Byte)		
bX16...23 bX24...31	n+21	Ch 2/Fn 1 Status (Low Byte) Ch 2/Fn 2 Status (High Byte)		
bX32...39 bX40...47	n+22	Output 0 Status (Low Byte) Output 1 Status (High Byte)		
bX48...55 bX56...63	n+23	Output 2 Status (Low Byte) Output 3 Status (High Byte)	DWord	4
bX64...71 bX72...79 bX80...87 bX88...95	n+24	System Functions Read/Write CTRIO(2) Internal Registers (see p. 6-10 for bit definitions)		

Input (n) Parameter Definitions

44 Total Bytes

Parameter values are in Decimal format.

Configured Function from CTRIO Workbench	Parameter 1 Contents DWORD	Parameter 2 Contents DWORD
Non-scaled Counter	Raw Input Value	Not Used
Scaled Counter	Scaled Value (pos. or rate)	Raw Value
Non-scaled Counter with Capture	Raw Value	Captured Value
Scaled Counter with Capture	Scaled Value (pos. or rate)	Scaled Captured Value
Non-scaled Timer	Previous Time (μs)	In Progress Time (μs)
Scaled Timer	Scaled Interval (rate)	Scaled In Progress Time (μs)
Pulse Catch	Not Used	Not Used



NOTE: For *DirectSOFT* users: the *I/O Map* dialog displays the exact memory locations in use by the *CTRIO(2)* module. Within the *I/O Map* dialog you can print out a report of memory loctions in use.

Input Function Status Bit Definitions

Input function offsets are listed in the order of Ch1/Fn1, Ch1/Fn2, Ch2/Fn1, Ch2/Fn2

Ch(x)/Fn(x) Status Bits (Transfers from CTRIO(2) to CPU)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets <i>DirectLOGIC</i> PLCs
Count Capture Complete Bit	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8
Timer Capture Start	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8
Timer Capture Complete (Timing) OR At Reset Value (Counting)	1, 9, 17, 25	20.1, 20.9, 21.1, 21.9
Timer "Timed Out" Bit	2, 10, 18, 26	20.2, 20.10, 21.2, 21.10
Pulse Catch Output Pulse State	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8
Pulse Catch Start	1, 9, 17, 25	20.1, 20.9, 21.1, 21.9

Output Status Bit Definitions (for Preset Table Control)

Output Status Offsets are listed in the order of the Output 0 - Output 3.

Output(x) Status Bits (Transfers from CTRIO(2) to CPU)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets <i>DirectLOGIC</i> PLCs
Command Error	38, 46, 54, 62	22.6, 22.14, 23.6, 23.14
Command Complete	39, 47, 55, 63	22.7, 22.15, 23.7, 23.15

Output Status Bit Definitions (Pulse Output)

Output Status Offsets are listed in the order of the Output 0/1, 2/3.

Status Bit CTRIO(2) to CPU	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Input Start (octal)
Output Enabled	32, 48	22.0, 23.0
Position Loaded	33, 49	22.1, 23.1
Output Suspended	34, 50	22.2, 23.2
Output Active	36, 52	22.4, 23.4
Output Stalled	37, 53	22.5, 23.5
Command Error	38, 54	22.6, 23.6
Command Complete	39, 55	22.7, 23.7

Output Memory Map for Data Transfers from DL CPUs to CTRIO(2)

- DL
- Win
- NI

The following table shows which memory locations are used for memory transfers from the CPU module to the CTRIO(2). The starting memory location is defined by the user in the I/O Map within CTRIO Workbench. If you are using a *Direct*LOGIC CPU, you will use the memory address offsets in the second column. If you are using a WinPLC, EBC, PBC, DEVNETS or MODBUS interface, you will use the non-PLC offsets in column one.

Data Type and Offset: WinPLC, EBC, PBC, DEVNETS, MODBUS	Address for Inputs (<i>Direct</i> LOGIC)	Definition	Format	Bytes
dwY0	n+0	Output 0 Parameter 3	DWord	4
dwY1	n+2	Output 1 Parameter 3		
dwY2	n+4	Output 2 Parameter 3		
dwY3	n+6	Output 3 Parameter 3		
wY0	n+10	Output 0 Command	Word	2
wY1	n+11	Output 0 Parameter 1		
wY2	n+12	Output 0 Parameter 2		
wY3	n+13	Output 1 Command		
wY4	n+14	Output 1 Parameter 1		
wY5	n+15	Output 1 Parameter 2		
wY6	n+16	Output 2 Command		
wY7	n+17	Output 2 Parameter 1		
wY8	n+20	Output 2 Parameter 2		
wY9	n+21	Output 3 Command		
wY10	n+22	Output 3 Parameter 1		
wY11	n+23	Output 3 Parameter 2		
bY0...7 bY8...15	n+24	Ch 1/Fn 1 Control (Low Byte) Ch 1/Fn 2 Control (High Byte)	Word	2
bY16...23 bY24...31	n+25	Ch 2/Fn 1 Control (Low Byte) Ch 2/Fn 2 Control (High Byte)		
bY32...39 bY40...47	n+26	Output 0 Control (Low Byte) Output 1 Control (High Byte)		
bY48...55 bY56...63	n+27	Output 2 Control (Low Byte) Output 3 Control (High Byte)		
bY64...71 bY72...79 bY80...87 bY88...95	n+30	System Functions Read/Write CTRIO(2) Internal Registers (see p. 6-10 for bit definitions)	DWord	4

52 Total Bytes

Output (n) Parameter Definitions (Parameters are in decimal format)

Configured Profile from CTRIO Workbench	Parameter 1 Contents WORD	Parameter 2 Contents WORD	Parameter 3 Contents DWORD
Trapezoid /Trapezoid with Limits	File # of stored profile	Not Used	Not Used
S-Curve, Symmetrical S-Curve	File # of stored profile	Not Used	Not Used
Dynamic Positioning /Positioning Plus	File # of stored profile	Not Used	Target Position

Configured Profile from CTRIO Workbench	Parameter 1 Contents WORD	Parameter 2 Contents WORD	Parameter 3 Contents DWORD
Dynamic Velocity	File # of stored profile	Not Used	Target Velocity
Home Search	File # of stored profile	Not Used	Not Used
Trapezoid Plus	File # of stored profile	Not Used	Target Position
Free Form	File # of stored profile	Not Used	Not Used



NOTE: For *DirectSOFT* users: the I/O Map dialog displays the exact memory locations in use by the CTRIO module. Within the I/O Map dialog you can print out a report of memory locations in use.

Output (n) Parameter Definitions (Parameters are in decimal format unless specified)

Profiles Completely Controlled by User Program	Parameter 1 Contents WORD	Parameter 2 Contents WORD	Parameter 3 Contents DWORD
Velocity Mode	Initial Frequency	Duty Cycle	Number of Pulses (Hex)
Run to Limit Mode	Initial Frequency	Input Edge / Duty Cycle(Hex)	Not Used
Run to Position mode	Initial Frequency	Input Function Comparison and Duty Cycle (Hex)	Input Function Comparison Value

Input Function Control Bit Definitions

Input function offsets are listed in the order of Ch1/Fn1, Ch1/Fn2, Ch2/Fn1, Ch2/Fn2

Ch(n)/Fn(n) Control Bits (Transfers from CPU to CTRIO)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets <i>DirectLOGIC</i> PLCs
Enable Count Capture	0, 8, 16, 24	24.0, 24.8, 25.0, 25.8
Enable Timer Capture	0, 8, 16, 24	24.0, 24.8, 25.0, 25.8
Enable Pulse Catch	0, 8, 16, 24	24.0, 24.8, 25.0, 25.8
Reset	1, 9, 17, 25	24.1, 24.9, 25.1, 25.9

Output Control Bit Definitions (for Preset Table Control)

Output Control Offsets are listed in the order of the Output 0 - Output 3.

Output(n) Control Bits (Transfers from CPU to CTRIO(2))	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets <i>DirectLOGIC</i> PLCs
Enable Output	32, 40, 48, 56	26.0, 26.8, 27.0, 27.8
Process Command	39, 47, 55, 63	26.7, 26.15, 27.7, 27.15

Output Control Bit Definitions (Pulse Output)

Pulse output control Offsets are listed in the order of Outputs 0/1, 2/3.

Output Control Bit (Transfers from CPU to CTRIO(2))	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)	Read as:
Enable Output	32, 48	26.0, 27.0	Level
Go to Position	33, 49	26.1, 27.1	Rising Edge
Suspend Output	34, 50	26.2, 27.2	Level
Direction	36, 52	26.4, 27.4	Level
Process Command	39, 55	26.7, 27.7	Rising Edge

Output Control Bit Definitions (Raw Mode)

Output Control Offsets are listed in the order of the Output 0 - Output 3.

Output(n) Control Bits (Transfers from CPU to CTRIO(2))	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets DirectLOGIC PLCs
Enable Output	32, 40, 48, 56	26.0, 26.8, 27.0, 27.8

System Functions Status Bit Definitions

DirectLOGIC Offset (n+24)

Status Bits (Transfers from CTRIO(2) to CPU)	V-memory Offsets <i>DirectLOGIC</i> PLCs
System Command Error	24.6
System Command Complete	24.7
Ch1 A	25.0
Ch1 B	25.1
Ch1 C	25.2
Ch1 D	25.3
Ch2 A	25.4
Ch2 B	25.5
Ch2 C	25.6
Ch2 D	25.7
Out 0 Active	25.8
Out 0 Mode	25.9
Out 1 Active	25.10
Out 1 Mode	25.11
Out 2 Active	25.12
Out 2 Mode	25.13
Out 3 Active	25.14
Out 3 Mode	25.15

System Functions Control Bit Definitions

DirectLOGIC Offset (n+30)

Control Bits (Transfers from CPU to CTRIO(2))	V-memory Offsets <i>DirectLOGIC</i> PLCs
Process System Command	30.7

Addressing Conventions (with V-memory Examples for DirectLOGIC PLCs)

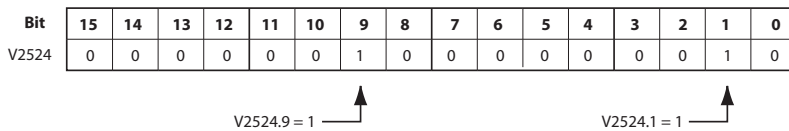
Example for Bit-accessed Data in PLC CPUs

- DL
- Win
- NI

In this example, the V-memory location V2524 contains a value equal to 514 in decimal.

- 514 decimal = 0202 Hex = 0000 0010 0000 0010 binary

The bit V2524.1 refers to the 2nd to the least significant bit (set to 1 in this example). Likewise, V2524.9 refers to bit number 9, the 10th from the least significant bit (also set to 1 in this example).

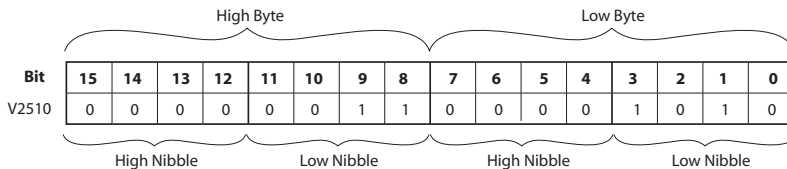


Addressing High and Low Byte of Word Parameters

In the following example, the V-memory location V2510 contains a value equal to 3 (decimal) in the high byte and 10 (decimal) in the low byte.

- 3 decimal = 03 Hex = 0000 0011 binary in the high byte, and
- 10 decimal = 0A Hex = 0000 1010 binary in the low byte.

This example could represent the Command Code “Edit Table Entry.” The value 03 (Hex) would represent the File number in the high byte, and the 0A (Hex) would represent the remainder of the Command Code in the low byte.



Addressing High and Low Word of DWord Parameters

Double Word parameters are addressed in a similar fashion to the high and low bytes of a Word Parameter. For example, a DWord that begins in V2300 consumes both V2300 and V2301. The Low Word is V2300, and the High Word is V2301.

Input Function Status/Control Bits and Parameters

Input Function Status Bit Definitions

Input function offsets are listed in the order of Ch1/Fn1, Ch1/Fn2, Ch2/Fn1, Ch2/Fn2

Ch(x)/Fn(x) Status Bits (Transfers from CTRIO(2) to CPU)	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets <i>DirectLOGIC</i> PLCs
Count Capture Complete Bit	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8
Timer Capture Start	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8
Timer Capture Complete (Timing) OR At Reset Value (Counting)	1, 9, 17, 25	20.1, 20.9, 21.1, 21.9
Timer "Timed Out" Bit	2, 10, 18, 26	20.2, 20.10, 21.2, 21.10
Pulse Catch Output Pulse State	0, 8, 16, 24	20.0, 20.8, 21.0, 21.8
Pulse Catch Start	1, 9, 17, 25	20.1, 20.9, 21.1, 21.9

Input Function Control Bit Definitions

Input function offsets are listed in the order of Ch1/Fn1, Ch1/Fn2, Ch2/Fn1, Ch2/Fn2

Ch(n)/Fn(n) Control Bits (Transfers from CPU to CTRIO(2))	Bit Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets <i>DirectLOGIC</i> PLCs
Enable Count Capture	0, 8, 16, 24	24.0, 24.8, 25.0, 25.8
Enable Timer Capture	0, 8, 16, 24	24.0, 24.8, 25.0, 25.8
Enable Pulse Catch	0, 8, 16, 24	24.0, 24.8, 25.0, 25.8
Reset	1, 9, 17, 25	24.1, 24.9, 25.1, 25.9

Input Function Status DWord Parameters

Input function offsets are listed in the order of Ch1/Fn1, Ch1/Fn2, Ch2/Fn1, Ch2/Fn2 and are in decimal format.

DWord Status CTRIO(2) to CPU	DWord Offsets: WinPLC, EBC, PBC, DEVNETS, MODBUS	V-memory Offsets from Output Start (octal)
DWord Parameter 1	0, 2, 4, 6	0, 4, 10, 14
DWord Parameter 2	1, 3, 5, 7	2, 6, 12, 16

Configured Function from CTRIO Workbench	Parameter 1 Contents DWORD	Parameter 2 Contents DWORD
Non-scaled Counter	Raw Input Value	Not Used
Scaled Counter	Scaled Value (pos. or rate)	Raw Value
Non-scaled Counter with Capture	Raw Value	Captured Value
Scaled Counter with Capture	Scaled Value (pos. or rate)	Captured Value
Non-scaled Timer	Previous Time (μs)	In Progress Time (μs)
Scaled Timer	Scaled Interval (rate)	In Progress Time (μs)
Pulse Catch	Not Used	Not Used

NOTE: If you select the 'discrete on chx/fnx' option for an input channel using pulse catch mode, you will get a message when you exit the I/O config screen noting 'pulse follower mode' or 'Pulse extension mode' for this output channel. This means only that the output will pulse for the specified duration when the input receives a sufficiently long pulse input.



Example Input Control/Status Bits and Parameter Register Addresses

The following tables provide example addresses based on V2000 selected for the base input address and V2030 selected for the base output address. The Input Functions discussed on the following pages use these example addresses.

Status Registers: Example using V2000 as base input address for Input Channel 1 (Status bits and DWords received from CTRIO(2) to CPU)

Name	PLC Example 1: Bit-of-Word (see note 2) DL05, DL06, D2-250-1/260, D4-450	PLC Example 2: Control Relay (see note 1) D2-240	Value
Counter Capture Complete Bit	V2020.0	C160	ON when Capture is completed
Timer Capture Starting	V2020.0	C160	On when Timer Capture begins
Timer Capture Complete (Timing) OR At Reset Value (Counting)	V2020.1	C161	ON when Timer Capture completes
Timer "Timed Out" Bit	V2020.2	C162	On when specified Timer "Time Out" period is exceeded
Pulse Catch Output Pulse State	V2020.0	C160	ON for the specified pulse time if input pulse qualifies as a valid pulse
Pulse Catch Starting	V2020.1	C161	ON when pulse edge occurs
Parameter 1	V2001–V2000	V2001–V2000	Decimal
Parameter 2	V2003–V2002	V2003–V2002	Decimal

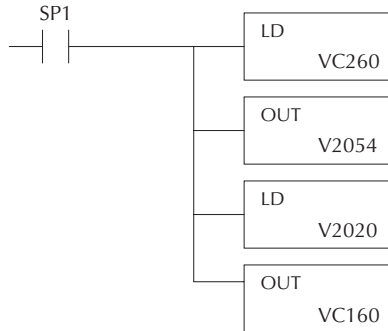
Control Registers: Example using V2030 as base output address for Input Channel 1 (Control bits sent from CPU to CTRIO(2))

Name	PLC Example 1: Bit-of-Word (see note 2) DL05, DL06, D2-250-1/260, D4-450	PLC Example 2: Control Relay (see note 1), D2-240 CPU	Data Format
Enable Counter Capture	V2054.0	C260	Bit
Enable Timer Capture	V2054.0	C260	Bit
Enable Pulse Catch	V2054.0	C260	Bit
Reset	V2054.1	C261	Bit

Memory Mapping Example for D2-240 CPU



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 below.



NOTE 2: For example, **DirectSOFT** 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.

**Status Bits: Example using V2000 as base input address For Output Channel 1
(Status bits received from CTRIO(2) to CPU)**

Name	PLC Example 1: Bit-of-Word (see note 2) DL05, DL06 D2-250-1/260, D4-450	PLC Example 2: Control Relay (see note 1) D2-240	Value
Output Enabled	V2022.0	C120	ON when Enable Output is ON
Position Loaded	V2022.1	C121	Used for Dynamic Positioning
Output Suspended	V2022.2	C122	ON when Output pulse is suspended
Output Active	V2022.4	C124	ON when Output is Pulsing
Output Stalled	V2022.5	C125	CTRIO Output Fault (should never be ON)
Command Error	V2022.6	C126	ON if Command or Parameters are invalid
Command Complete	V2022.7	C127	ON if Module receives Process Command

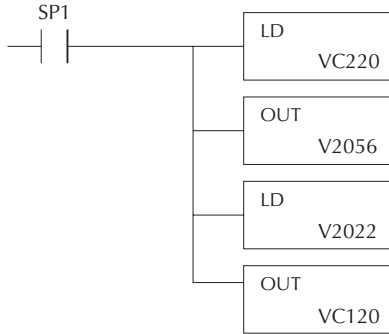
**Control Bits/Registers: Example using V2030 as base output address for Output
Channel 1 (Control DWords, Words, and bits sent from CPU to CTRIO(2))**

Name	PLC Example 1: Bit-of-Word (see note 2), DL-05, DL06 D2-250-1/260, D4-450	PLC Example 2: Control Relay (see note 1), D2-240 CPU
Command Code	V2040	V2040
Parameter 1	V2041	V2041
Parameter 2	V2042	V2042
Parameter 3	V2031– V2030	V2031–V2030
Enable Output	V2056.0	C220
Go to Position	V2056.1	C221
Suspend Output	V2056.2	C222
Direction	V2056.4	C224
Process Command	V2056.7	C227

Memory Mapping Example for D2-240 CPU



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 below.



NOTE 2: For example, **DirectSOFT** uses *B2022.2* in the ladder code to indicate that you are addressing the third bit of V-memory register V2022. The "B" prefix indicates bit-of-word addressing.

SYSTEM FUNCTIONS



In This Appendix...

System Functions	B-2
Write All Registers (IBoxes).....	B-3
Write All Registers (DL-PLC).....	B-4
Write One Register (IBoxes).....	B-5
Write One Register (DL-PLC)	B-6
Read All Registers (IBoxes).....	B-7
Read All Registers (DL-PLC)	B-8
Read One Register (IBoxes)	B-9
Read One Register (DL-PLC)	B-10
Read Error Code (IBoxes).....	B-11
Read Error Code (DL-PLC).....	B-12
System Functions Examples Overview	B-13
Single Channel Simulating Retentive Quad Counter	B-14
Dual Channel Simulating Retentive Quad Counters	B-17
Reading CTRIO Internal Registers.....	B-20

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

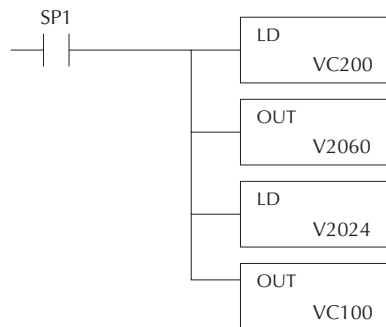
NI

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, **DirectSOFT** 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.

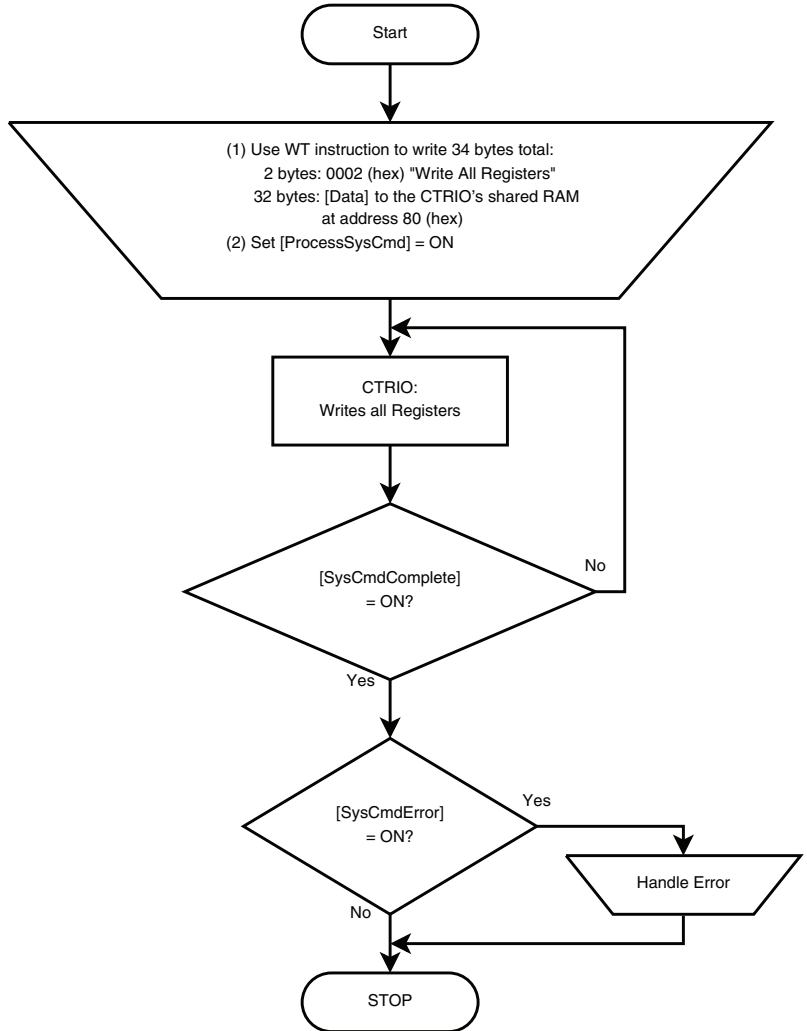
Write All Registers (IBoxes)

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)

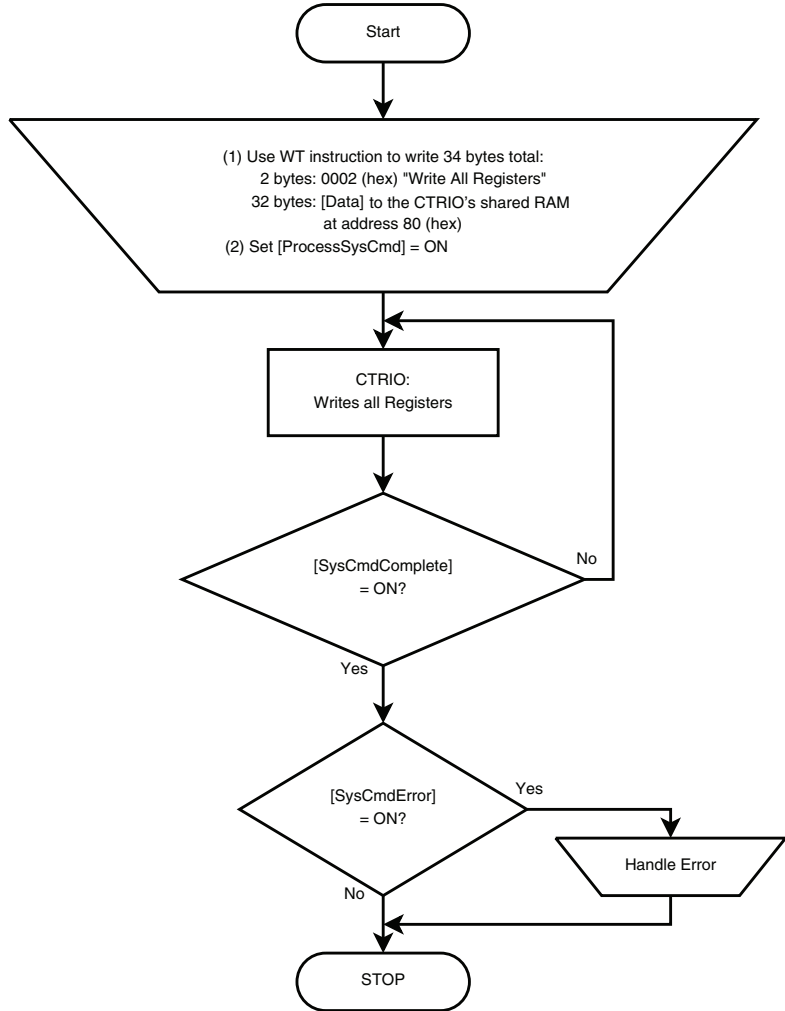
PARAMETERS:

- (1) [ProcessSysCmd]
n+30.7
- (2) [SysCmdComplete]
n+24.7
- (3) [SysCmdError]
n+24.6

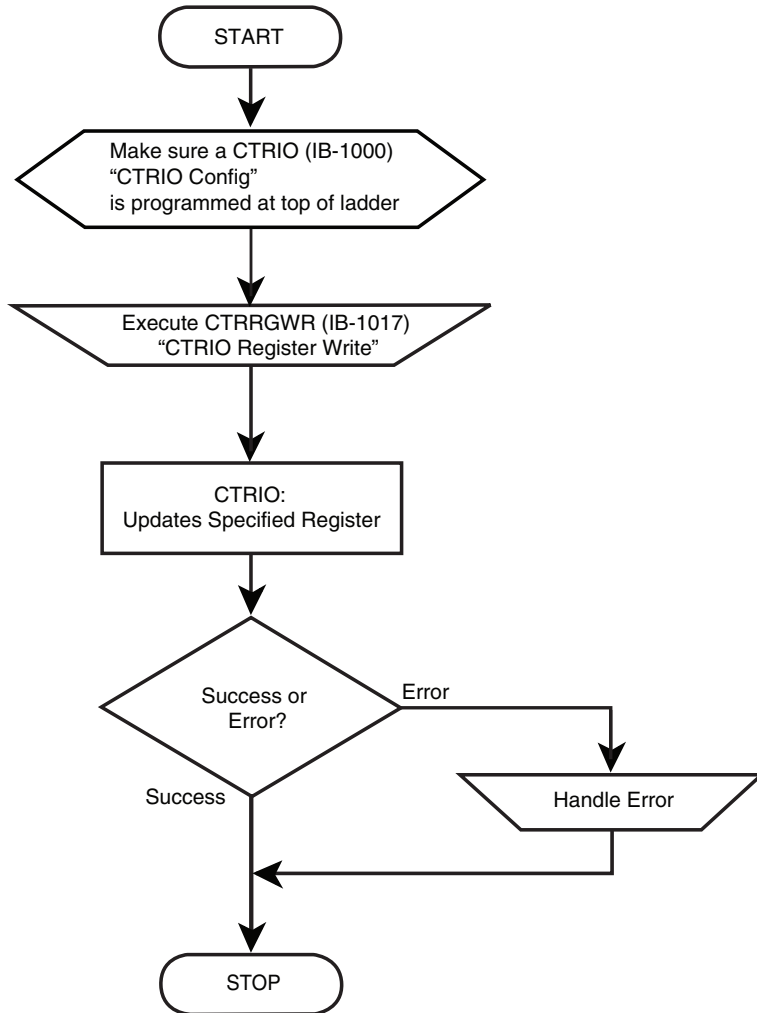
KEY:

[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)

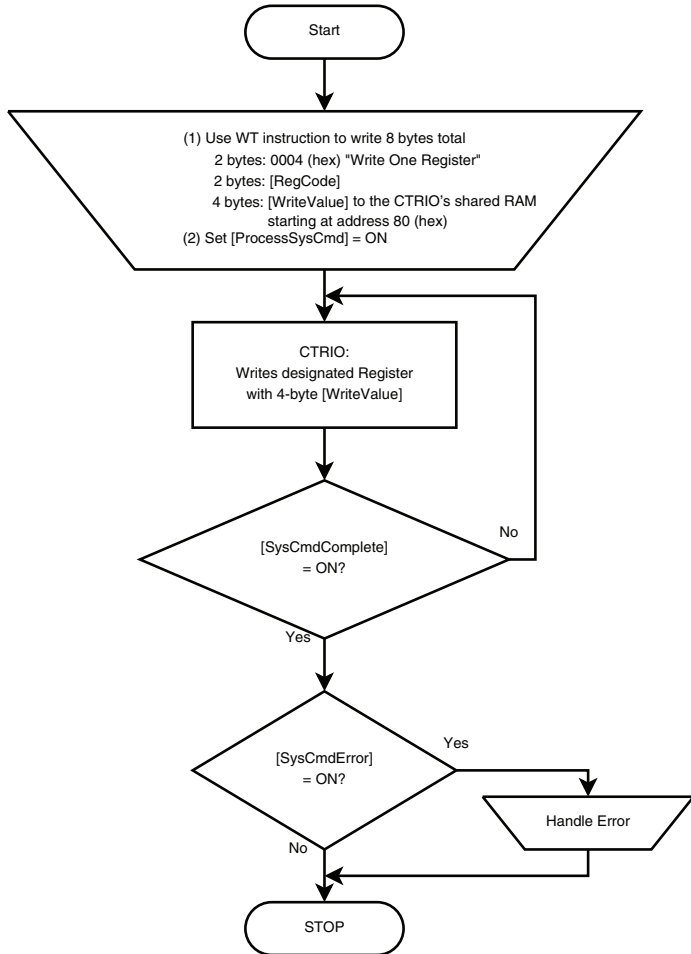
PARAMETERS:

- (1) [ProcessSysCmd] n+30.7
- (2) [SysCmdComplete] n+24.7
- (3) [SysCmdError] n+24.6

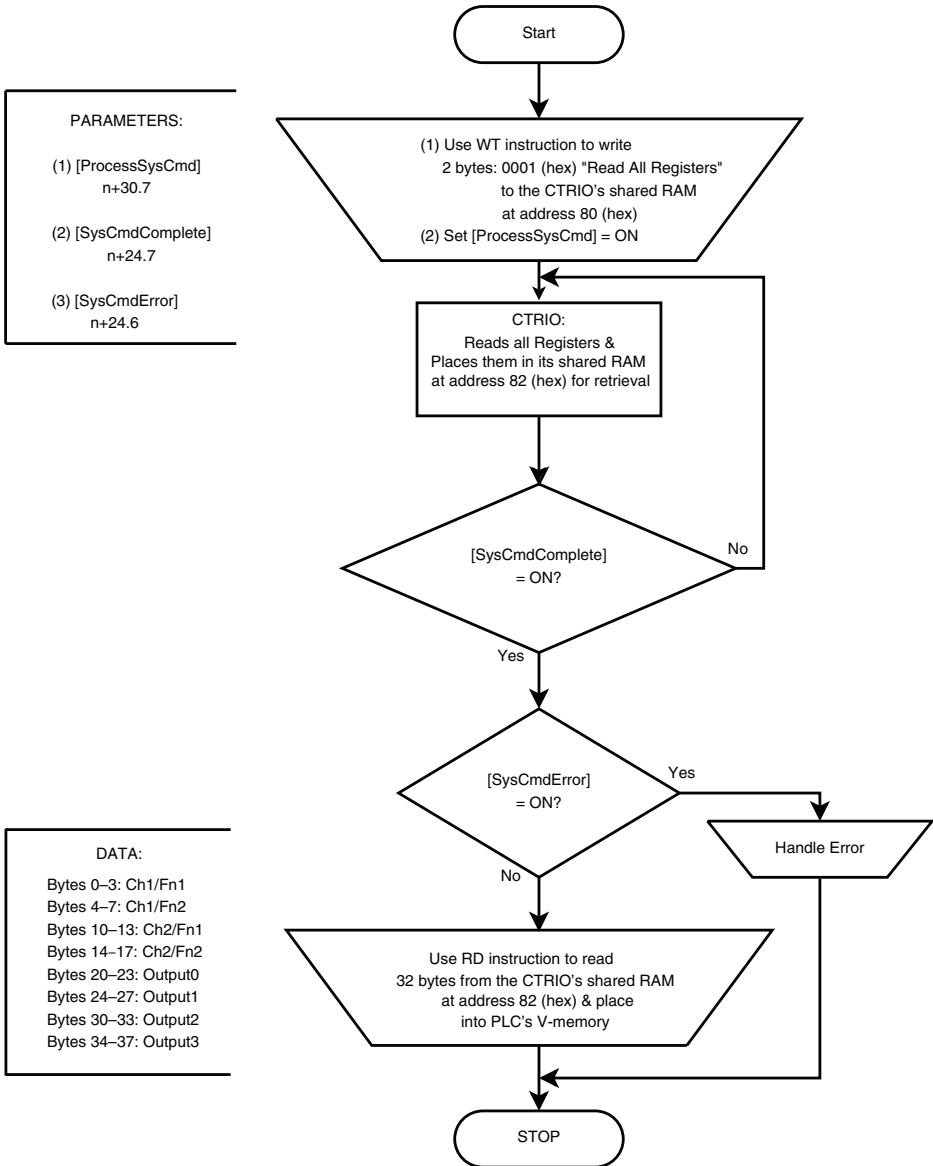
KEY:

[RegCode]
(hex value):

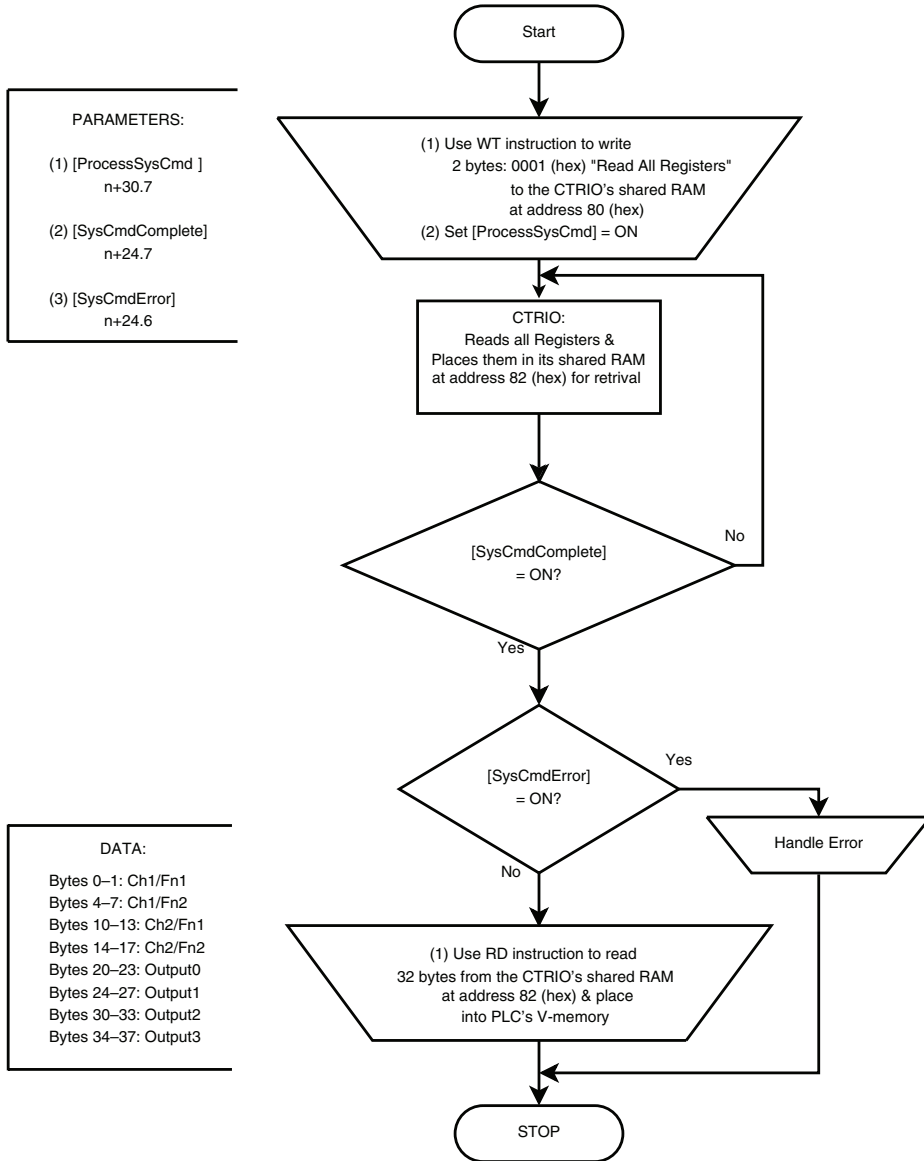
- 0000 = Ch1/Fn1
- 0001 = Ch1/Fn2
- 0002 = Ch2/Fn1
- 0003 = Ch2/Fn2
- 0004 = Output0
- 0005 = Output1
- 0006 = Output2
- 0007 = Output3
- 0008 = Ch1/Fn1 Reset
- 0009 = Ch1/Fn2 Reset
- 000A = Ch2/Fn1 Reset
- 000B = Ch2/Fn2 Reset
- 000C = Ch1A Filter
- 000D = Ch1B Filter
- 000E = Ch1C Filter
- 000F = Ch1D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



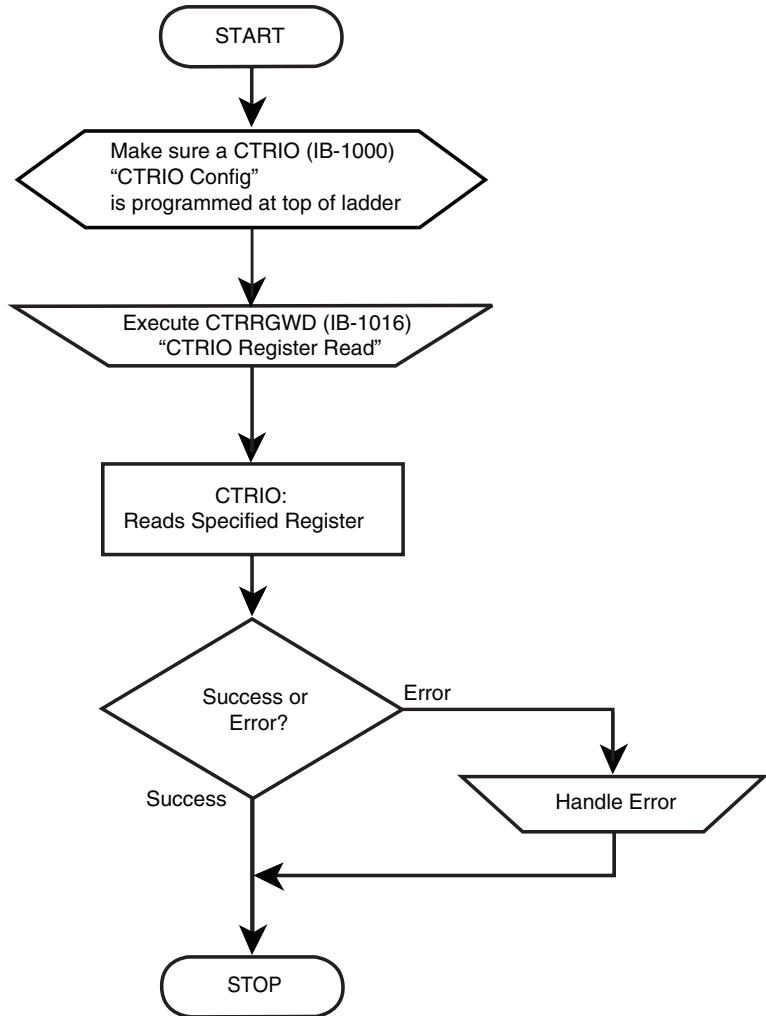
Read All Registers (IBoxes)



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

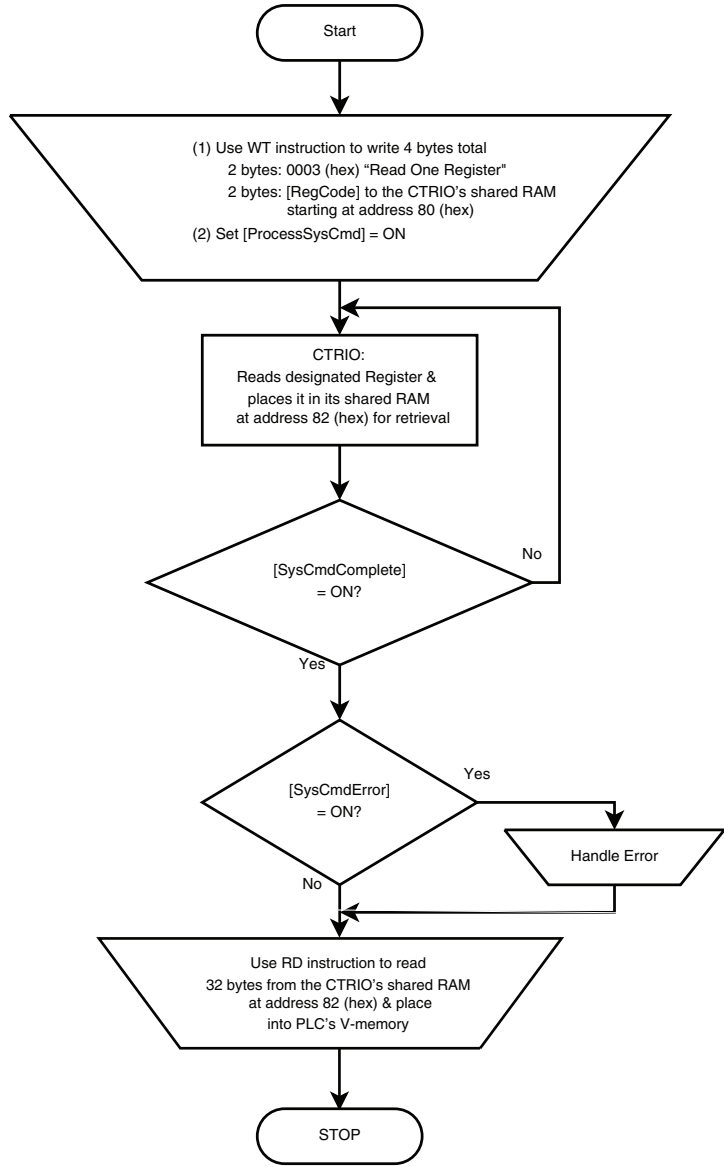
KEY:

[RegCode]
(hex value):

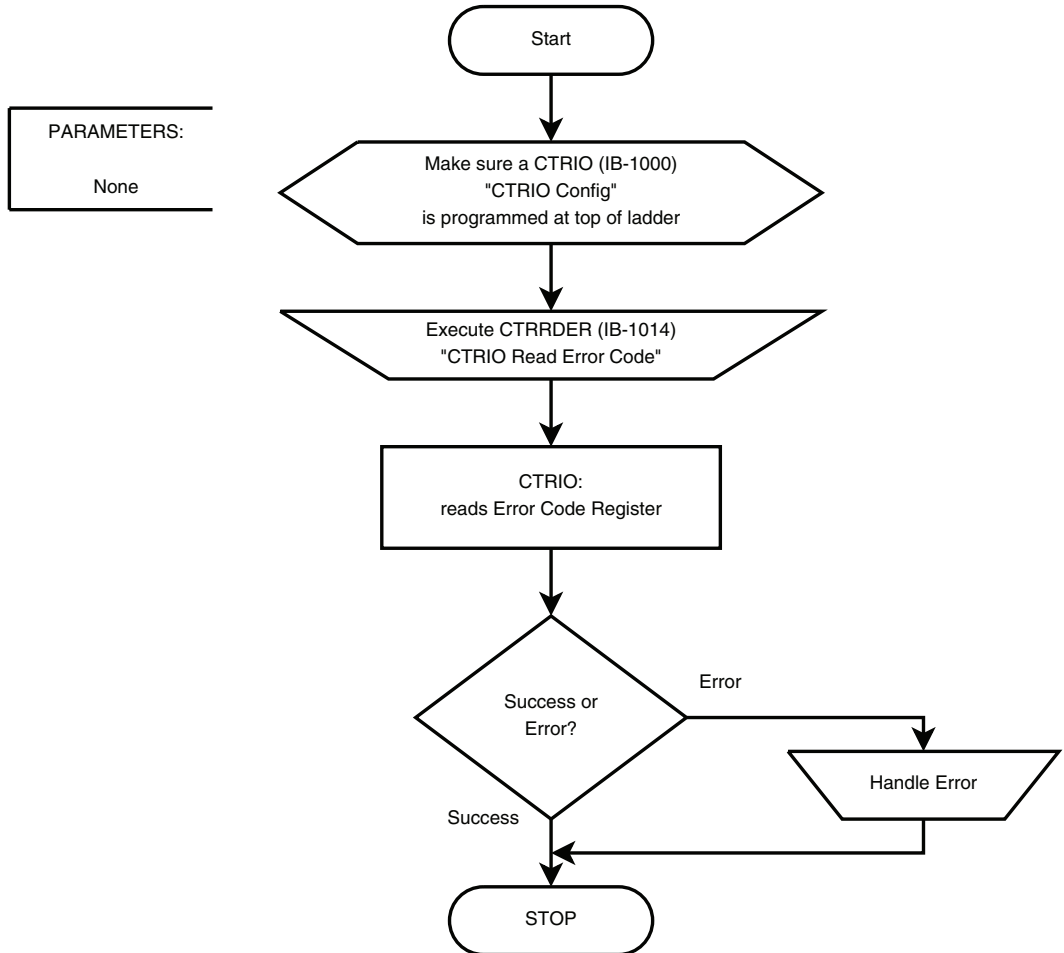
0000 = Ch1/Fn1
0001 = Ch1/Fn2
0002 = Ch2/Fn1
0003 = Ch2/Fn2
0004 = Output0
0005 = Output1
0006 = Output2
0007 = Output3

0008 = Ch1/Fn1 Reset
0009 = Ch1/Fn2 Reset
000A = Ch2/Fn1 Reset
000B = Ch2/Fn2 Reset

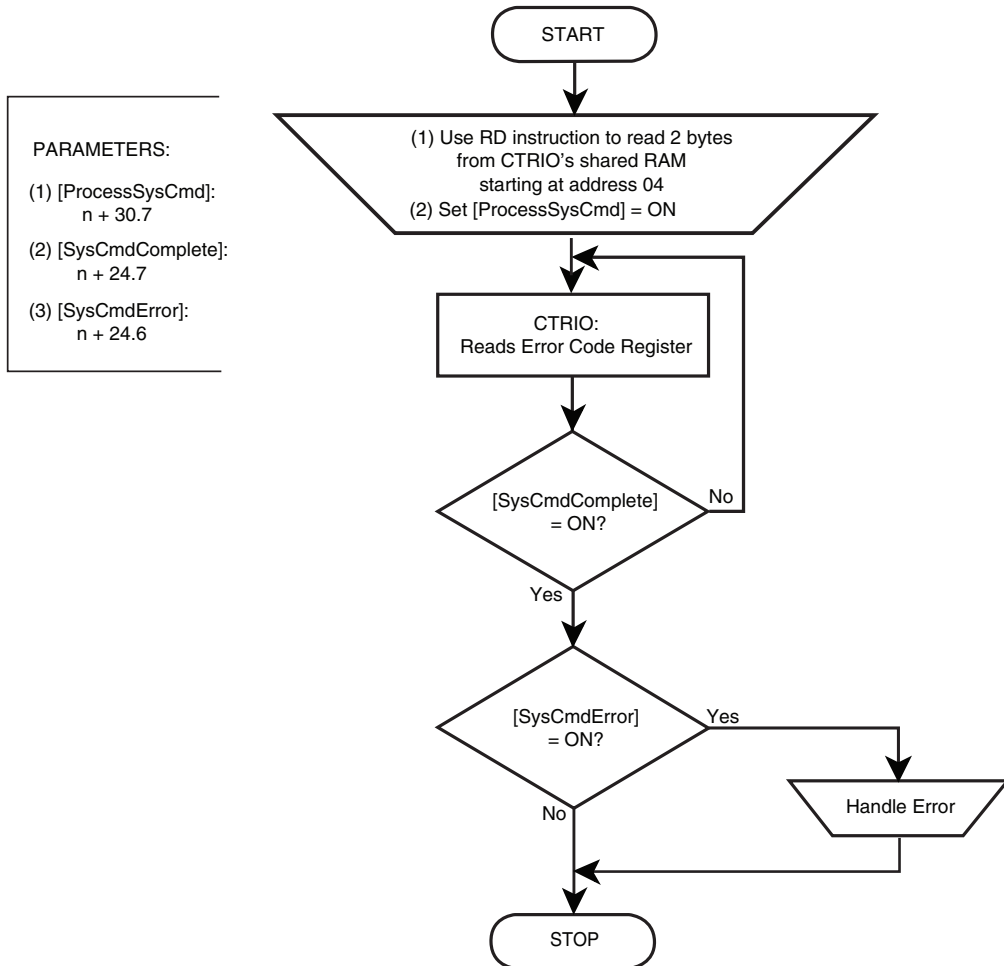
000C = Ch1A Filter
000D = Ch1B Filter
000E = Ch1C Filter
000F = Ch1D Filter
0010 = Ch2A Filter
0011 = Ch2B Filter
0012 = Ch2C Filter
0013 = Ch2D Filter



Read Error Code (IBoxes)



Read Error Code (DL-PLC)



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 CTRIO's current count register.

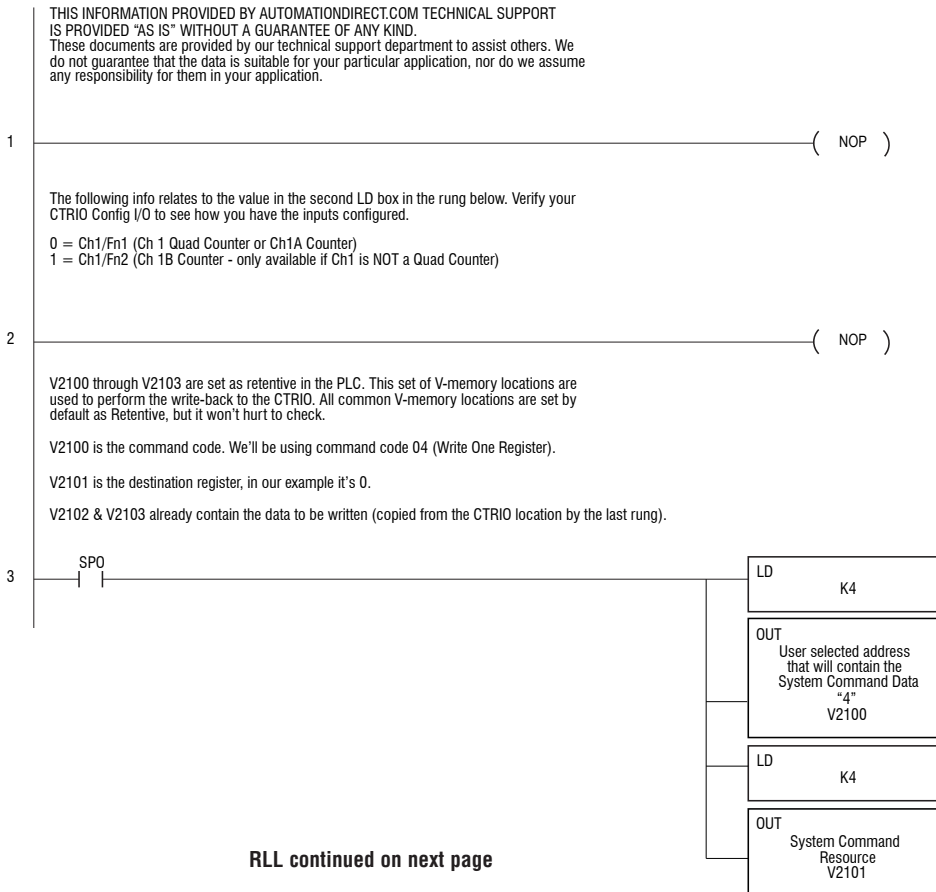
The example assumes the use of standard I/O mapping based off V2000 for Inputs, and V2030 for Outputs. Adjust according to your CTRIO Config I/O map.

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.

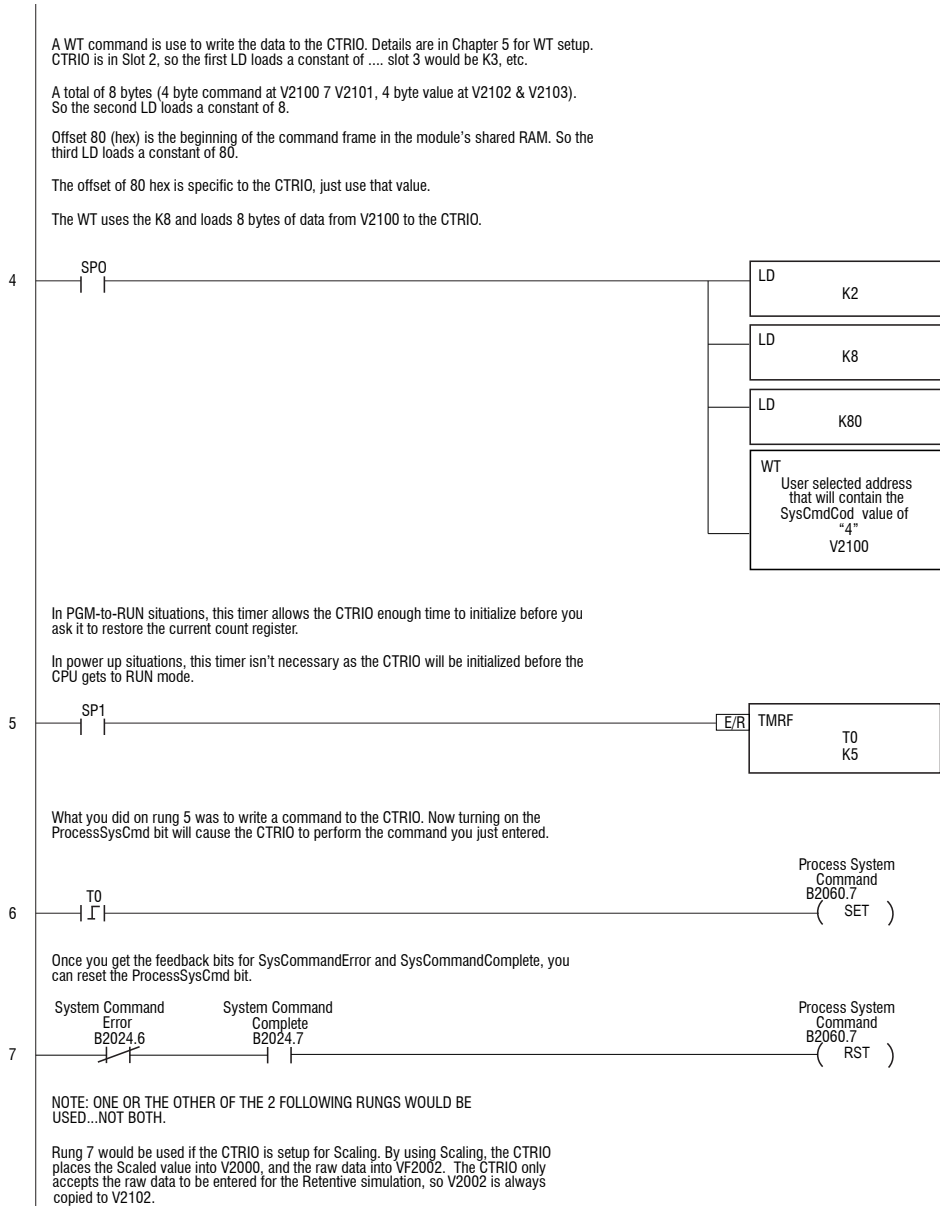
The CTRIO only accepts the raw count as the retentive value. On every scan after T0 completes, the raw count at V2002 is copied to V2102. (V2000 needs to be copied if Scaling is not used; see last rungs).

Example RLL for Single Channel Simulation of Retentive Quad Counter



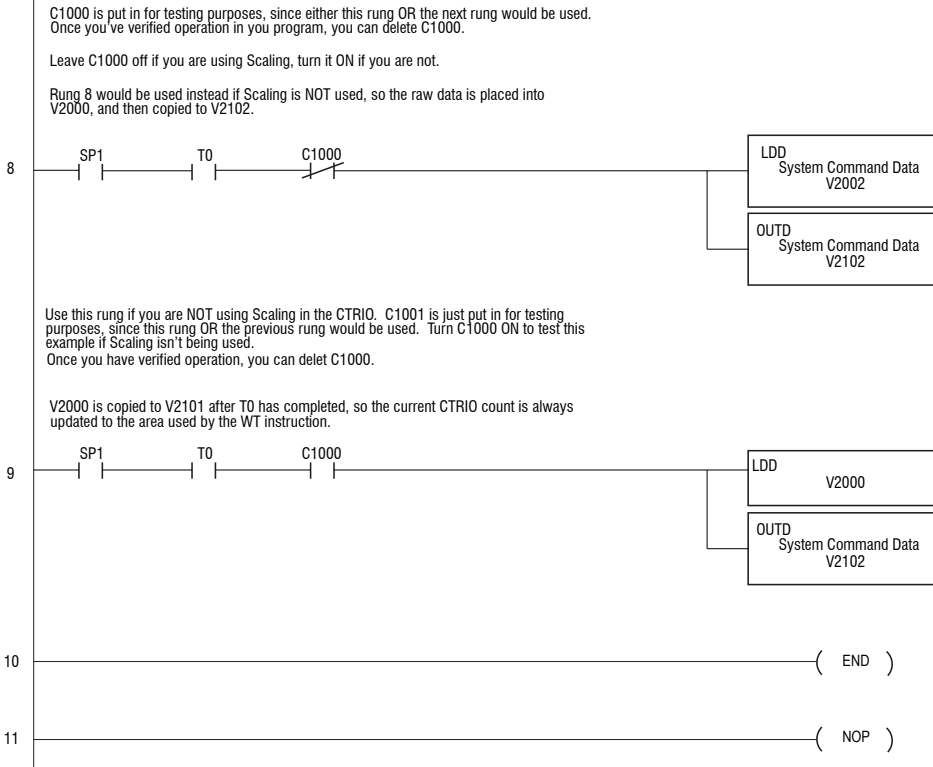
RLL continued on next page

RLL continued from previous page



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

THIS INFORMATION PROVIDED BY AUTOMATIONDIRECT.COM TECHNICAL SUPPORT IS PROVIDED "AS IS" WITHOUT A GUARANTEE OF ANY KIND. These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.

1

(NOP)

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 defacto standard of I/O map based off of 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 of 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).

2

(NOP)

RLL continued on following page

Appendix B: System Functions

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'.



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.



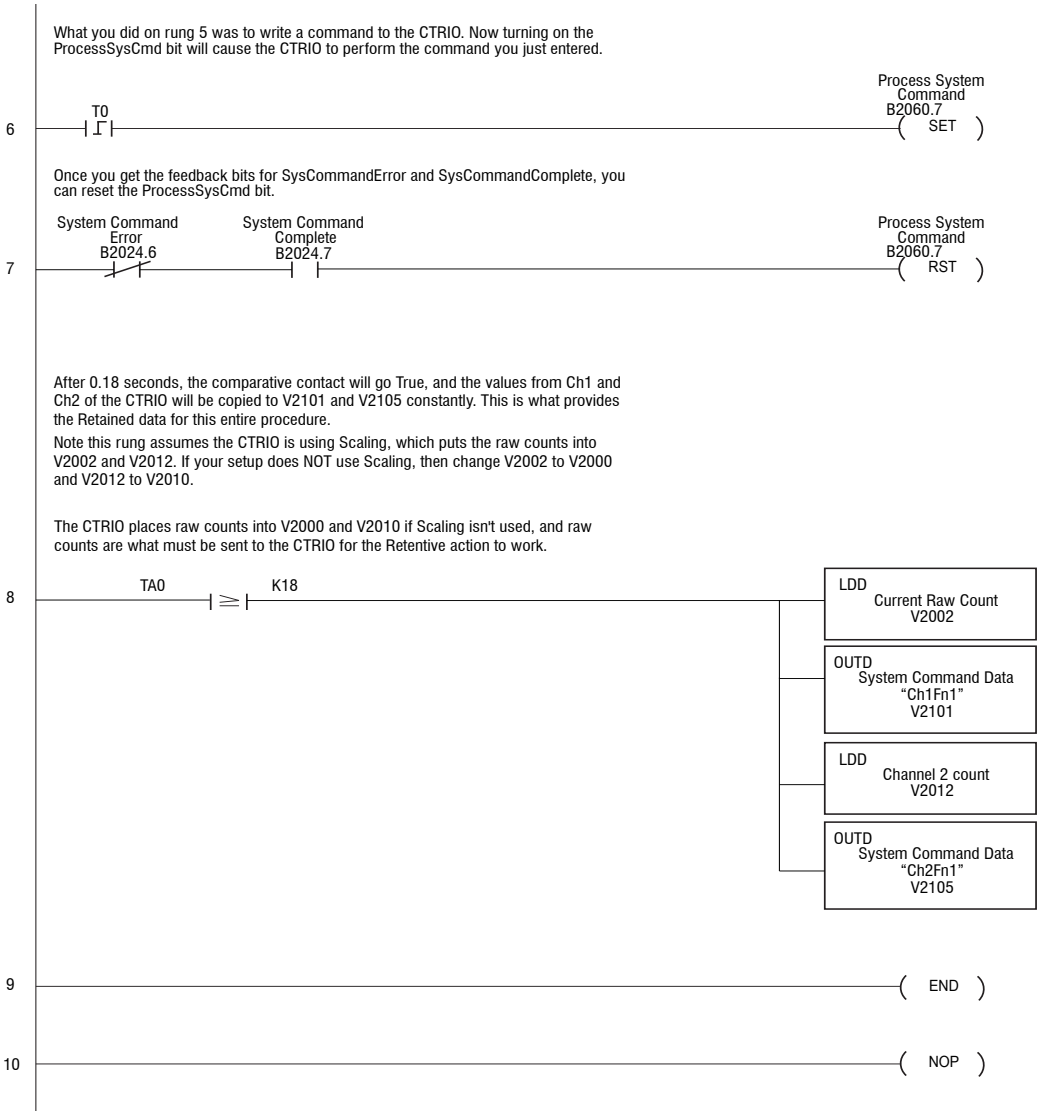
In PGM-to-RUN situations, this timer allows the CTRIO enough time to initialize before you ask it to restore the current count register.

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



RLL continued on following page

RLL continued from previous page

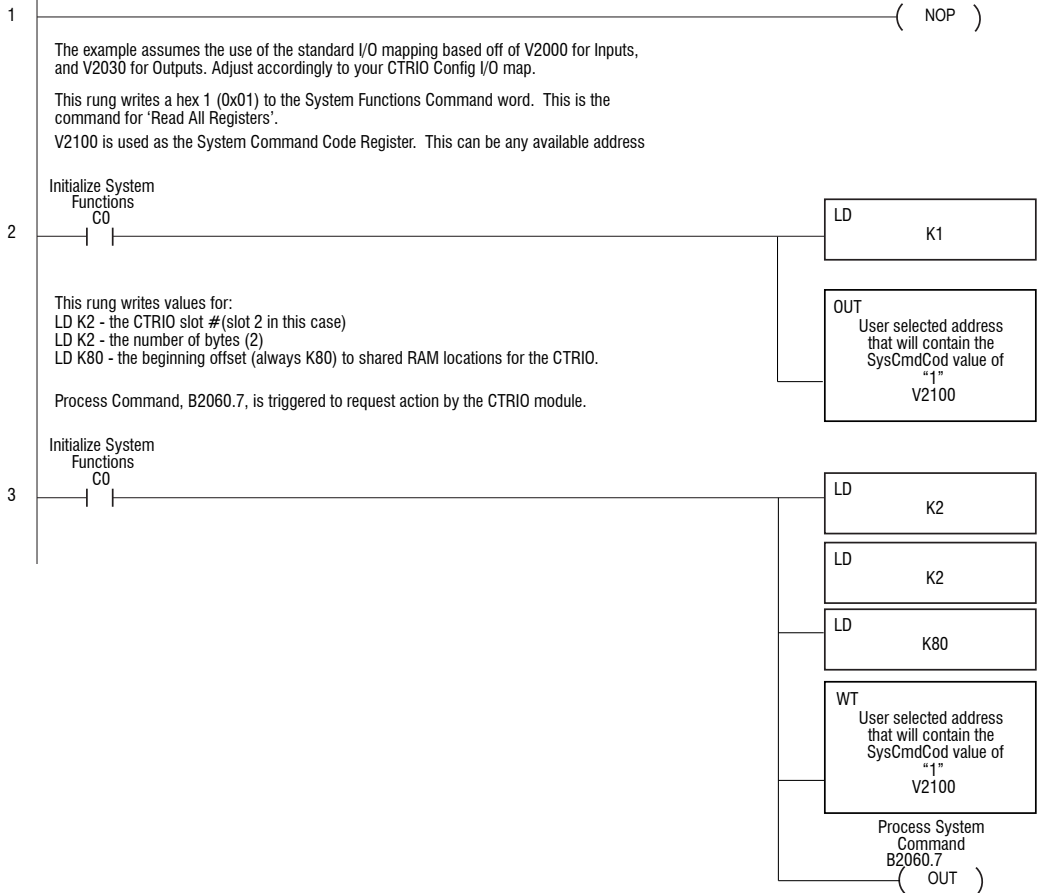


Reading CTRIO Internal Registers

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

THIS INFORMATION PROVIDED BY AUTOMATIONDIRECT.COM TECHNICAL SUPPORT IS PROVIDED "AS IS" WITHOUT A GUARANTEE OF ANY KIND. These documents are provided by our technical support department to assist others. We do not guarantee that the data is suitable for your particular application, nor do we assume any responsibility for them in your application.



RLL continued on next page

RLL continued from previous page

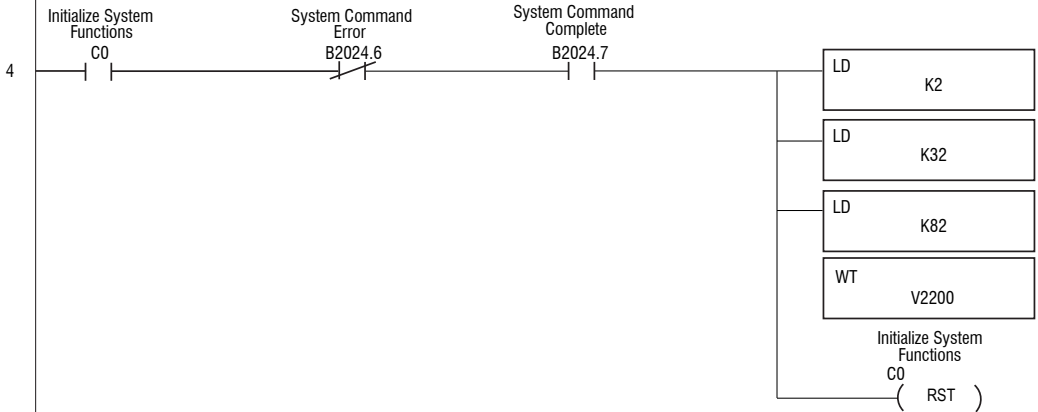
Once feedback is received for SysCommandError and SysCommandComplete after requesting a "Read All Registers" from the CTRIO module, this rung sends a read request of the following:

LD K2 - the CTRIO slot # (slot 2 in this case)

LD K32 - the number of bytes (32)

LD K82 - Offset 82 (Hex) is the beginning of the Input and Output Registers within the Command Frame of the CTRIO.

The received data is stored in a block of memory starting at V2200.



This self-running timer, T0, controls how often the internal registers are Read (RD) from the CTRIO.

This timer will trigger the sequence every 900ms.



The Timer T0 triggers the Read All Registers sequence.

