

**D4-HSC**  
**High Speed Counter**  
Manual Number D4-HSC-M

Shaded area represents the die-cut covers from  
Imaging Technologies, 5-98



# WARNING

Thank you for purchasing automation equipment from **Automationdirect.com**<sup>™</sup>. We want your new **DirectLOGIC**<sup>™</sup> 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.*

If you have any questions concerning the installation or operation of this equipment, or if you need additional information, please call us at 1-770-844-4200 or FAX 1-770-886-3199.

This publication is based on information that was available at the time it was printed. At **Automationdirect.com**<sup>™</sup> 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

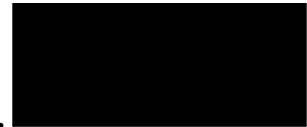
This publication may contain references to products produced and/or offered by other companies. The product and company names may be trademarked and are the sole property of their respective owners. **Automationdirect.com**<sup>™</sup> disclaims any proprietary interest in the marks and names of others.

Stage is a trademark of Koyo Electronics Industries Co., LTD. Think & Do Software is a trademark of Think & Do Software, Inc. Texas Instruments is a registered trademark of Texas Instruments, Inc. TI, TIWAY, Series 305, Series 405, TI305, and TI405 are trademarks of Texas Instruments, Inc. Siemens and SIMATIC are registered trademarks of Siemens, AG. GE is a registered trademark of General Electric Corporation. Series One is a registered trademark of GE Fanuc Automation North America, Inc. MODBUS is a registered trademark of Gould, Inc. IBM is a registered trademark of International Business Machines. MS-DOS and Microsoft are registered trademarks of Microsoft Corporation. Windows and Windows NT are trademarks of Microsoft Corporation. OPTOMUX and PAMUX are trademarks of OPTO 22.

**Copyright 1999, Automationdirect.com**<sup>™</sup> 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**<sup>™</sup> Incorporated. **Automationdirect.com**<sup>™</sup> retains the exclusive rights to all information included in this document.

# Manual Revisions



*Refer to this history in all correspondence and/or discussion about this manual.*

Title: DL405 High Speed Counter Module User Manual, Rev A

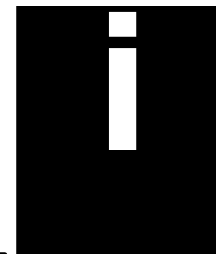
Manual Number: D4-HSC-M

Issue	Date	Effective Pages	Description of Changes
Original	2/95	Cover/Copyright Contents Manual History 1-1 - 1-12 2-1 - 2-7 3-1 - 3-9 4-1 - 4-12 5-1 - 5-7 6-1 - 6-5 7-1 - 7-10 A-1 - A-3 B-1 - B-7 Index-1 - Index-3	Original Issue
Rev A	6/98		Minor corrections and downsizing



# Table of Contents

---



## Chapter One: Getting Started

<b>Introduction</b> .....	<b>1-2</b>
The Purpose of this Manual .....	1-2
Who Should Read this Manual .....	1-2
Where to Begin .....	1-2
Supplemental Manuals .....	1-2
How this Manual is Organized .....	1-3
Technical Assistance .....	1-3
<b>HSC Features</b> .....	<b>1-4</b>
What is a High Speed Counter? .....	1-4
Who Needs a High Speed Counter? .....	1-4
Types of Counting .....	1-4
Standard Counting .....	1-4
Quadrature Counting .....	1-5
Output Control .....	1-5
Sampling .....	1-6
Home Search .....	1-6
<b>How Does the HSC Work With the CPU?</b> .....	<b>1-7</b>
Setup Performed Via Shared Memory .....	1-7
<b>Physical Characteristics &amp; Specifications</b> .....	<b>1-8</b>
LED Assignments .....	1-8
<b>Overview of HSC Inputs and Outputs</b> .....	<b>1-10</b>
Y Data Type Equivalents for Some Functions .....	1-10
<b>X Input and Y Output Assignments</b> .....	<b>1-11</b>
<b>Putting It All Together</b> .....	<b>1-12</b>
Five Steps for Using the HSC .....	1-12
The Next Chapter .....	1-12

## Chapter 2: Installation & Wiring

<b>How to Install the D4-HSC</b> .....	<b>2-2</b>
<b>Connecting the Wiring</b> .....	<b>2-3</b>
Wiring the D4-HSC Terminal Block .....	2-3
Wiring Guidelines .....	2-3
<b>Connecting the Power Supply</b> .....	<b>2-4</b>
Using an External Power Supply .....	2-4
Using the Internal DL-405 System Power Supply .....	2-4

---



Count Input Wiring Diagram .....	2-5
Control Input Wiring Diagram .....	2-6
Control Output Wiring Diagram .....	2-7

## Chapter 3: Understanding the Operation

The Operating Basics .....	3-2
Assigning Your Data Types .....	3-3
Automatic Configuration .....	3-3
Using the X and Y Assignment Table .....	3-4
I/O Assignment Table .....	3-5
Reading and Writing Shared Memory .....	3-6
Contents and Data Flow .....	3-6
The Two-Step Process for Writing Data to Shared Memory .....	3-7
Reading Data From Shared Memory .....	3-7
Understanding How Numbers are Stored In Shared Memory .....	3-8
Why is it important? .....	3-8
Allocating the 4 Bytes of Memory .....	3-8
Dealing With the Negative Numbers .....	3-9
Placing a Negative Number into Shared Memory .....	3-9
The Next Chapter .....	3-9

## Chapter Four: Setting Up and Controlling the Counting

Introduction to Using <i>DirectSOFT</i> .....	4-2
Selecting the Counting Mode .....	4-3
Determining Which Mode to Use .....	4-3
Ladder Logic for Determining the Counting Mode .....	4-3
Selecting the Counting Direction .....	4-4
How Ym+13 and Ym+14 Together Determine Counting Direction .....	4-4
Ladder Logic to Select Counting Direction .....	4-4
Selecting the Counting Resolution .....	4-5
Choose from 3 Resolution Settings .....	4-6
Why Change the Counting Resolution? .....	4-6
Example RLL: 2x Resolution .....	4-6
Example RLL: 4x Resolution .....	4-6
Default Setting .....	4-6
Specifying an Offset .....	4-7
What is an Offset? .....	4-7
External Method .....	4-7
Internal Method .....	4-7
Specifying a Preset .....	4-8
What is a Preset? .....	4-8
How Does the Preset Affect the Outputs? .....	4-8
Loading the Preset Into Shared Memory .....	4-8
Checking the Status of Preset Relative to Current Count .....	4-8

---



<b>Starting and Resetting the Current Count</b> .....	<b>4-9</b>
Starting the Counter .....	4-9
Automatically Resetting the Counter .....	4-9
Internal Reset .....	4-9
External Reset Using RST .....	4-9
Summary of Reset Count Relays .....	4-9
External Reset Using INZ .....	4-10
<b>Latching or Inhibiting the Current Count</b> .....	<b>4-11</b>
What Does Latching Do? .....	4-11
How Do You Trigger the Latching Process? .....	4-11
Sample RLL for Latching .....	4-11
What Is Meant By Inhibiting the Count? .....	4-11
How Do You Inhibit the Count? .....	4-11
Sample RLL for Inhibiting the Count .....	4-11
Summary of Latch and Inhibiting Output Relays .....	4-11
<b>Monitoring Overflow and Resetting Flags</b> .....	<b>4-12</b>
What is a Counting Overflow? .....	4-12
Status Flag for Overflow .....	4-12
Tracking Overflows .....	4-12
Summary of Input and Output Relays for Overflow and Flag Reset .....	4-12
 <b>Chapter Five: Controlling the Outputs</b>	
<b>Introduction</b> .....	<b>5-2</b>
The 4 Control Outputs .....	5-2
All the Output Signals Look the Same .....	5-2
The 4 Output Relays for Turning ON Each Control Output .....	5-2
Using the Internal Relays to Control the Outputs is Optional .....	5-2
<b>Manual Output Control</b> .....	<b>5-3</b>
Using Ym+6 and Ym+23 to turn direction outputs ON or OFF .....	5-3
<b>Automatic Output Control</b> .....	<b>5-3</b>
Also Called HSC RUN .....	5-3
How Do You Invoke HSC RUN? .....	5-3
Example of Using Ym+3 to Activate HSC RUN .....	5-3
<b>Using the Direction Outputs</b> .....	<b>5-4</b>
What is a Direction Output? .....	5-4
How the HSC Knows When to Turn ON CW or CCW .....	5-4
<b>Timing Diagrams for HSC RUN</b> .....	<b>5-5</b>
Current Count < Preset .....	5-5
Current Count = Preset .....	5-5
Current Count > Preset .....	5-5
<b>Using the Speed Outputs with the Direction Outputs</b> .....	<b>5-6</b>
What are the Speed Outputs? .....	5-6
2 Ways to Initiate OUT1 and OUT2 .....	5-6
Using HSC RUN to Initiate OUT1 .....	5-6
How OUT2 is Initiated in HSC RUN .....	5-6
Monitoring Speed Output Status .....	5-6
Figure A: Example of Outputs (Decel=1000, Preset=3000, Current Count >Preset) .....	5-6
Figure B: Example of Outputs (Decel=1000, Preset=3000, Current Count<Preset) .....	5-6

## Chapter Six: Special Features

<b>Planning a Home Search Application</b> .....	<b>6-2</b>
Requirements for Home Search .....	6-2
Example of Home Search Application .....	6-2
<b>Using Sampling</b> .....	<b>6-4</b>
Why Use Sampling? .....	6-4
Enabling and Monitoring the Sampling .....	6-4
How to Calculate the Timebase .....	6-5
What Happens If You Want to Enter a Value with Decimal Points? .....	6-5
Summary of Input and Output Relays for Sampling .....	6-5

## Chapter Seven: Program Applications

<b>A Quick Checkout of the Module</b> .....	<b>7-1</b>
What It Does .....	7-2
How It Works .....	7-2
Things You Need for the Example .....	7-2
Wiring Diagram for the Example .....	7-2
RLL Program for Quick Checkout of the Module .....	7-3
<b>Application No.1: Drilling Operation</b> .....	<b>7-4</b>
Ladder Logic for Drilling Operation .....	7-5
<b>Application No. 2: Cut-to-Length Operation</b> .....	<b>7-8</b>
Ladder Logic for Cut-to-Length Operation .....	7-9

## Appendix A: Introduction to Motor Drives and Encoders

<b>What is a Drive and How Does It Connect to the HSC?</b> .....	<b>A-2</b>
<b>What is an Encoder and How Does It Connect to the HSC?</b> .....	<b>A-3</b>
How the Incremental Encoder Works .....	A-3
What is the Z-Marker? .....	A-3

## Appendix B: Summary of Tables and Charts Presented in the Text

<b>Specifications</b> .....	<b>B-2</b>
<b>X and Y Assignment Table</b> .....	<b>B-4</b>
<b>Shared Memory Table</b> .....	<b>B-5</b>
<b>Table for Determining Count Direction</b> .....	<b>B-6</b>
<b>Counting Resolution Table (Quadrature Only)</b> .....	<b>B-7</b>

# Getting Started

---

## In This Chapter. . . .

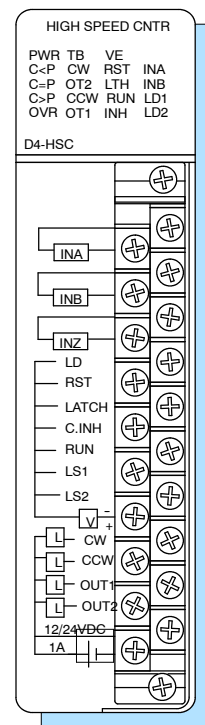
- Introduction
  - HSC Features
  - How Does the HSC Work With the CPU?
  - Physical Characteristics & Specifications
  - Overview of HSC Inputs and Outputs
  - X Input and Y Output Assignments
  - Putting It All Together
-



## Introduction

### The Purpose of this Manual

Thank you for purchasing the High Speed Counter module for the DL405. This manual shows you how to install, program, and maintain the equipment. It also helps you understand the module's operating characteristics. Since we constantly try to improve our product line, we occasionally issue addenda that document new features and changes to the products. If an addendum is included with this manual, please read it to see which areas of the manual or product have changed.



### Who Should Read this Manual

If you understand PLC systems, this manual will provide all the information you need to get and keep your High Speed Counter module up and running. We will use examples and explanations to clarify our meaning and perhaps help you brush up on specific features used in the DL405 system. This manual is not intended to be a generic PLC training manual, but rather a user reference manual for the DL405 High Speed Counter Module.

### Where to Begin

If you are in a hurry and already understand the basics of high speed counters and basic motion control, you may only want to skim this chapter, and move on to Chapter 2, Installation and Wiring.

Be sure to keep this manual handy for reference when you run into questions. If you are a new DL405 customer, we suggest you read this manual completely so you can fully understand the high speed counter module's configurations and the procedures used. We believe you will be pleasantly surprised with how much you can accomplish with PLC**Direct**<sup>™</sup> products.

### Supplemental Manuals

Depending on the products you have purchased, there may be other manuals necessary for your application. You will want to supplement this manual with any other manuals written for other products. We suggest:

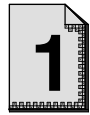
- D4-USER-M (the D4-405 User Manual)
- DA-DSOFT-M (the **Direct**SOFT User Manual)

### Technical Assistance

If you have questions that are not answered by this manual, our Technical Support Team is glad to help. They are available from 9:00AM until 6:00PM Eastern Time Monday through Friday at 800-633-0405.

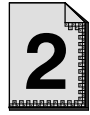
## Chapters

Below is a table showing a summary of contents provided within each section of this manual. The manual is organized into the following seven chapters:



### Getting Started

includes a brief description of the high speed counter module, common applications for high speed counters, and an overview of the steps necessary to setup and operate the high speed counter.



### Installation and Wiring

shows you step-by-step how to install and wire the HSC. Includes wiring diagrams.



### Understanding Operation

a must for understanding the rest of the manual. It covers the shared memory concept, the assignment of data types for the HSC, and how values are stored.



### Setting Up and Controlling the Count

covers offsets and presets, which are needed for many applications. It provides the programming tools needed to make full use of the counting capability. It includes count inhibiting, count latching and overflow flags.



### Controlling the Outputs

this chapter introduces you to the HSC's four different control outputs. It shows you how HSC RUN uses your preset information and current count to trigger the outputs in an ordered format. It also covers "manual" operation of the outputs *without* using HSC RUN.



### Special Features

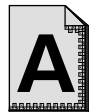
covers two special features that have been built into the HSC. You will learn about sampling and home search capabilities.



### Applications

shows you how to write programs that will provide possible solutions for some common applications you might encounter.

## Appendices



### Introduction to Motor Drives and Shaft Encoders

although **PLCDirect™** does not provide encoders or motor drives, we have included a brief overview explaining some of the more common encoders and electronic drives that may be used with the D4-HSC High Speed Counter.



### Introduction to Motor Drives and Shaft Encoders

this includes the X and Y data type assignment chart and an address map for the seven shared memory parameters.

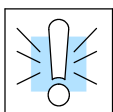
## Other Resources

You can also check our online resources for the latest product support information:

- **Internet** - the address of our Web site is <http://www.plcdirect.com>
- **Bulletin Board Service (BBS)** - call (770) 844-4209



The "note pad" icon in the left-hand margin indicates the paragraph to its immediate right will be a **special note**.



The "exclamation mark" icon in the left-hand margin indicates the paragraph to its immediate right will be a **warning** or **caution**. These are very important because the information may help you prevent serious personal injury or equipment damage.

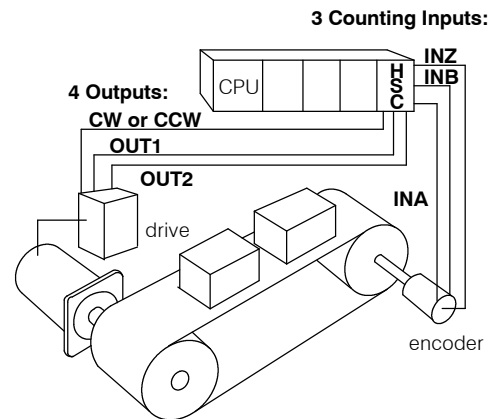
## HSC Features

### What is a High Speed Counter?

Literally, high speed counters count fast! The D4-HSC high speed counter has one channel for counting pulses from sensors, encoders, switches, and so on, at rates up to 100 kHz (50% duty cycle). It is designed to make your job simpler. The HSC has its own microprocessor that asynchronously counts and accumulates the high speed pulses. This means the main CPU of the DL405 is free to do the other important tasks. It can simply check the accumulated count when it needs to do so.

### Who Needs a High Speed Counter?

If you have an application that needs to count pulses rapidly, then you are a prime candidate for an HSC. The D4-HSC also has 4 outputs that can be used for controlling motor speed and direction. There is one special requirement. The variable speed motors or motor drives that are used must be capable of changing speed when receiving a voltage input between 10.2 VDC and 26.4 VDC. Many digital drives being offered today offer programmable input capability, precisely for this sort of application.



**NOTE:** The motor control capability should not be confused with a pulse output capability such as used with stepper motors. The D4-HSC outputs a voltage level dependent on an external power supply and **does not have pulse output capability**. You should check the specs of your drive or motor carefully to make certain that the specifications of this module match your application requirements.

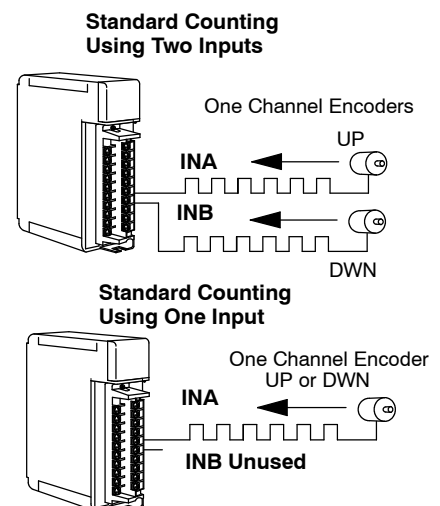
### Types of Counting

The D4-HSC can do **standard** UP and DOWN counting or it can do **quadrature** counting. These are software selectable as two different modes.

### Standard Counting

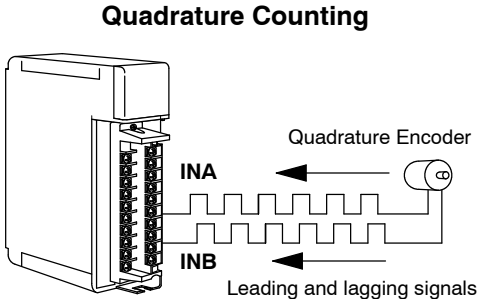
With standard counting you can use the two counting input signals (**INA** and **INB**) of the D4-HSC. One input is used for counting UP and the other used for counting DOWN. You can't use both inputs for the same direction of counting.

You could be using only one of the inputs if desired. In this case, the other input terminal should be left unwired. You control the direction of counting by the manner in which you set a certain bit in your control program (shown later).



### Quadrature Counting

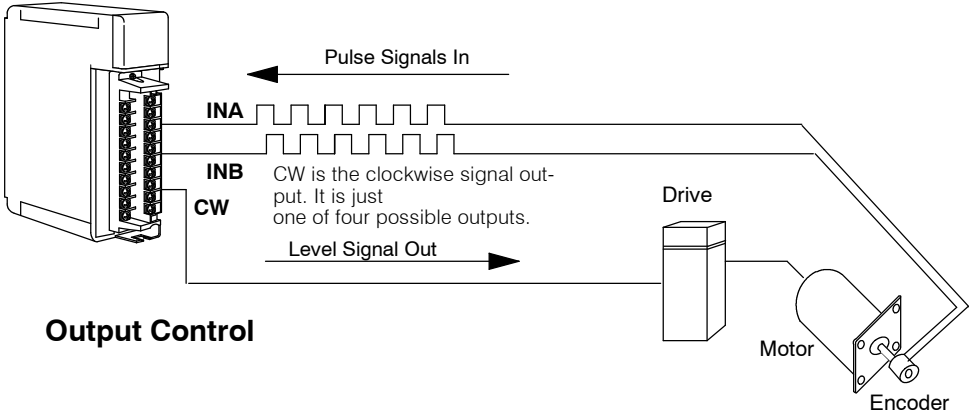
With quadrature counting, you must use both signals (**INA** and **INB**). Both input terminals are connected to the same field device, capable of outputting two square wave signals, each being offset 90 degrees. Quadrature counting is often preferred to standard counting because it can sense direction. Quadrature inputs are also more noise immune. With quadrature counting, the direction (UP or DOWN counting) is determined by whether the signal being received at INA leads or lags the signal received at INB. The D4-HSC looks at the signals coming in and compares them. It then determines which is leading and which is lagging. **NOTE:** We have not shown the optional use of the Z-output signal (connected to INZ of the HSC) that comes standard on most quadrature encoders. The use of this input option will be discussed when we cover resetting the counter externally and the automatic home search feature.



With a rotary encoder, the leading and lagging signal is determined by which direction the shaft is turning. This is how quadrature counting is able to sense direction.

### Output Control

A typical application for the D4-HSC might be that of having a quadrature type shaft encoder connected to your motor with the HSC counting and accumulating the pulses from the encoder as the motor rotates. The HSC knows which way the motor shaft is turning because it knows which of the two signals being received is leading and lagging. You can write ladder logic to change either the motor's speed or direction. (We'll show you how to do that a little later!) The D4-HSC provides output signals that can be used to change speed or direction. Usually these signals are connected to an electronic drive or motor controller rather than the motor itself. Some motors are "smart" with built-in logic circuitry for initiating speed and direction changes. In these exceptional cases, the HSC can be connected directly to the motor to initiate the changes. We leave it up to you to specify the motor control and thus dictate the load side of the application. **Make sure you check the specs of your motor or motor controller and that you are sure they match up with the specs of D4-HSC before making any connections.**



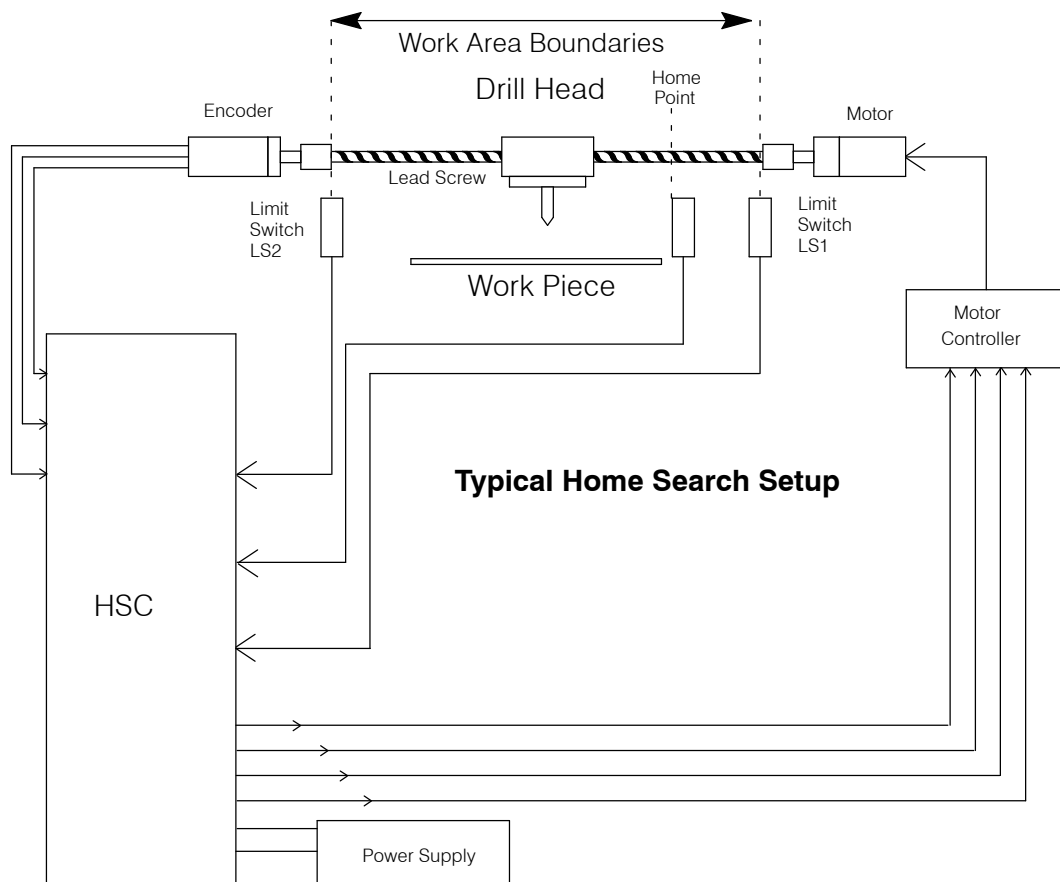
## Sampling

The HSC can also do sampling over time. That is, you can use simple ladder logic instructions to indicate the time period for a sampling. When you invoke the sampling feature of the HSC, it will keep track of the counted pulses for the time period you have specified and store the total for later RLL retrieval. This is a great feature for determining frequency of incoming pulses. It's as simple as specifying a time base, say 3 seconds, and then counting the pulses for that period. If the HSC sees 6000 pulses during that time span, then you know that you have an incoming pulse rate of 2 kHz ( $6000/3=2000$ ). There are many other uses for the sampling feature--frequency counting is just one example.

## Home Search

Many applications require a known starting position for a given work cycle, called "home point". This is the point to which the moving piece of apparatus doing the work (i.e. welder, drill, saw, glue gun, etc.) is returned at the end of each work cycle.

The HSC has an automatic feature that is designed to help find and return to a home position at the end of each work cycle. You could, of course, write your own home search RLL. However, the D4-HSC relieves you of that task. Since the algorithm associated with the automatic home search routine assumes a certain type of configuration, you will have to make sure that the components and position of each meets certain specified criteria. Then, you activate the home search using a Y-output relay. The HSC takes over from there. We'll show you how to do this later in this manual.



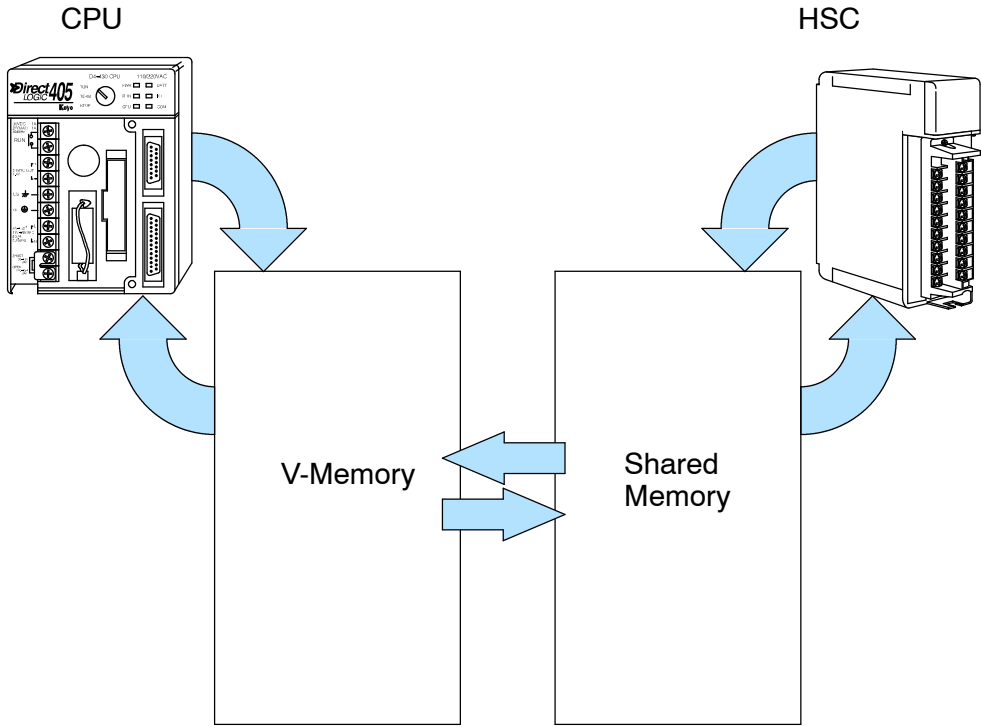
# How Does the HSC Work With the CPU?

## Setup Performed Via Shared Memory

The D4-HSC is an intelligent module that has its own microprocessor and memory. The microprocessor operates asynchronously to the DL405 CPU. Its memory area is called "shared memory", because both the DL405 CPU and the HSC can read and write to this area. In fact, that's how you handle important items like-- telling the counter your preset value. In this case, you store parameters first in your DL405 CPU's V-memory area; and then you transfer them to the shared memory area. Then, the HSC can read and use the information. The HSC microprocessor cannot read information directly from the DL405's V-memory area. Likewise, it cannot write information directly into the CPU's V-memory area. This is why the two-step process is always necessary.

The diagram below shows you the basic concept. Chapter 4 will cover the subject in depth.

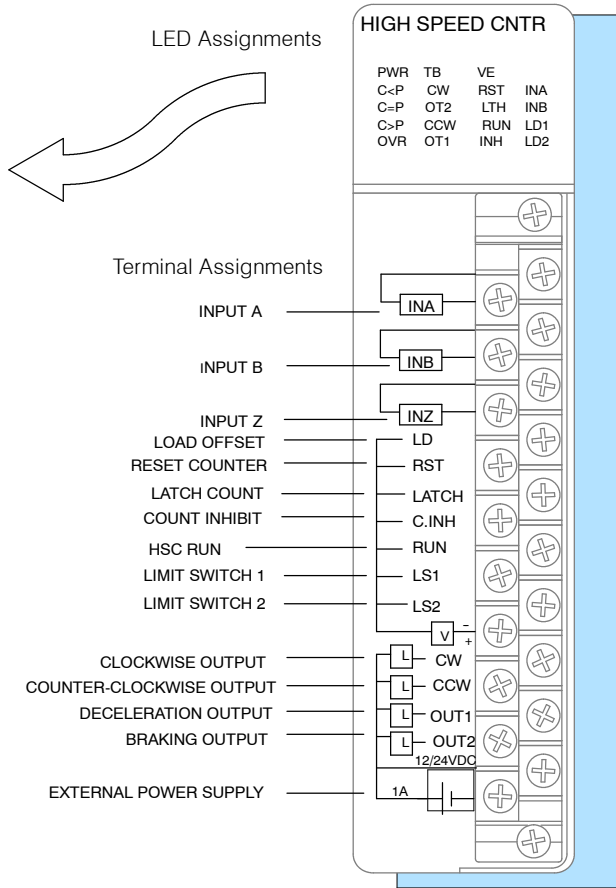
**The CPU and the HSC do not communicate directly. They do so by exchanging information to and from the shared memory area, via the CPU's V-Memory. Your ladder logic decides what information and when it is read and written between the two memory areas.**



# Physical Characteristics & Specifications

## LED Assignments

Label	Function
PWR	5V POWER ON
C<P	CURRENT COUNT LESS THAN PRESET
C=P	CURRENT COUNT EQUAL TO PRESET
C>P	CURRENT COUNT MORE THAN PRESET
OVR	COUNT OVERFLOW
TB	LOOSE OR MISSING TERMINAL BLOCK
CW	CLOCKWISE OUTPUT ENERGIZED
OUT2	BRAKE (OUT2) OUTPUT ENERGIZED
CCW	COUNTER CLOCKWISE OUTPUT ENERGIZED
OUT1	DECELERATION (OUT1) OUTPUT ENERGIZED
VE	EXTERNAL POWER SUPPLY FOR OUTPUTS FAILED
RST	SIGNAL APPLIED TO RESET INPUT (External Only)
LTH	SIGNAL APPLIED TO LATCH INPUT (External Only)
RUN	SIGNAL APPLIED TO RUN INPUT (External Only)
INH	SIGNAL APPLIED TO INHIBIT CNT INPUT (External Only)
INA	SIGNAL APPLIED TO INA INPUT
INB	SIGNAL APPLIED TO INB INPUT
LD1	SIGNAL APPLIED TO INZ INPUT
LD2	SIGNAL APPLIED TO LD INPUT



General Specification	Rating or Requirement
405 CPU Firmware Requirements	Any PLCDirect CPU or other vendor's 405 CPU (Version 1.6 or later.)
Slot for Installation	Can be installed in any CPU or expansion base. Cannot be installed in a remote base.
Maximum No. HSC's per CPU	8
No. of I/O points required	Consumes 16 X-inputs and 32 Y-outputs
Intelligence Source	Has its own microprocessor (operates asynchronously to the DL405 CPU)
Internal Power Consumption	300 mA maximum at 5VDC
Field Wiring Connector	Removable terminal type
Count Signal Level	4.75VDC to 30VDC less than 10mA
Maximum Count Speed	100 kHz (50% duty cycle)
Minimum Input Pulse Width for Counting	5 μs (either state)
Count Input Signal Types	Standard (UP/DOWN) or quadrature (phase differential)
Count Range	-8,388,608 to +8,388,607
Count Direction	UP or DOWN (software selectable or hardwired)
CPU Scan Time Increase per HSC in base	4.2 to 5.5 ms

Counting Inputs (INA, INB, INZ)	Rating or Requirement
Input Voltage Range	4.75VDC to 30VDC
Maximum Input Current	10 mA
ON Voltage	4.75VDC
ON Current	5mA
OFF Voltage	2.0VDC
OFF Current	1.6mA
OFF to ON Delay	1.2 $\mu$ s at 5VDC 0.8 $\mu$ s at 12VDC 0.5 $\mu$ s at 24VDC
ON to OFF Delay	1.0 $\mu$ s at 5VDC 1.2 $\mu$ s at 12VDC 2.5 $\mu$ s at 24VDC

Control Inputs (LD, LATCH, RST, CINH, RUN, LS1 LS2)	Rating or Requirement
Input Voltage Range	10.2VDC-26.4VDC
Maximum Input Current	10mA
ON Voltage	10.2VDC
ON Current	5mA (LD and LATCH); $\geq$ 4.8mA (RST,CINH,RUN,LS1 and LS2)
OFF Voltage	4.6VDC (LD and LATCH); $\leq$ 5.6VDC (RST,CINH,RUN,LS1 and LS2)
OFF Current	1.6mA (LD and LATCH); $\leq$ 2mA (RST,CINH,RUN,LS1 and LS2)
OFF to ON Delay	75 $\mu$ s at 12VDC (LD and LATCH) 82.5 $\mu$ s at 12VDC (RST,CINH,RUN,LS1 and LS2) 30 $\mu$ s at 24VDC (LD and LATCH) 37.5 $\mu$ s at 24VDC (RST,CINH,RUN,LS1 and LS2)
ON to OFF Delay	240 $\mu$ s at 12VDC (LD and LATCH) 105 $\mu$ s at 12VDC (RST,CINH,RUN LS1 and LS2) 260 $\mu$ s at 24VDC (LD and LATCH) 105 $\mu$ s at 24VDC (RST,CINH,RUN LS1 and LS2)

Control Outputs (CW,CCW,OUT1,OUT2)	Rating or Requirement
Output Power Source	External 10.2VDC-26.4VDC, 1A
Output Type	Open Collector
Maximum Output Current	100 mA per point
Output ON Voltage Drop	1.5VDC
Output OFF Leakage Current	100 $\mu$ A
Output OFF to ON Delay	22.5 $\mu$ s at 12VDC 21 $\mu$ s at 24VDC
Output ON to OFF Delay	210 $\mu$ s at 12VDC 270 $\mu$ s at 24VDC
Built-In Protection	Shut off when output driver IC=175 $^{\circ}$ C (Recovers at 150 $^{\circ}$ C) Shut off when short ( $\leq$ 500mA) is detected (Recovers when short is removed)
Input-pulse-reaching-Preset to internal-signal-reaching-Output 1 Time Delay	110 $\mu$ s



## Overview of HSC Inputs and Outputs

### Counting Inputs

**INA**-Depending on mode chosen, this is either a standard UP/DOWN counter input, or one of the quadrature counter inputs.

**INB**-Depending on mode chosen, this is either a standard UP/DOWN counter input, or one of the quadrature counter inputs.

**INZ**-This input can be used to *help you find home position* for positioning control. It can also be used as an external means of resetting the counter.

### Control Inputs

**LD**-If you want to *use an offset number with your counting*, a rising edge signal at this terminal will copy the offset value into the current count.

**RST**-A high (ON) signal at this terminal *resets the counter to zero and it remains there until there is a transition to a low signal (OFF)*.

**LATCH** - You may want to *store the current count*. The rising edge of a signal at this terminal will store the current count in shared RAM. Counting continues with no interruption.

**C.INH**- You may want to temporarily ignore the count inputs coming in on INA and INB. A high (ON) signal at this terminal will *inhibit the counting* to accomplish this need. Current count is suspended until a transition to a low (OFF) signal is seen.

**RUN**-Not to be confused with RUN mode of the DL405, a high (ON) signal here will *activate HSC RUN*. A low (OFF) signal will de-activate it.

**LS1 or LS2**-Either or both of these terminals can be connected to limit switches to help find home position, or they can merely be used as discrete inputs.

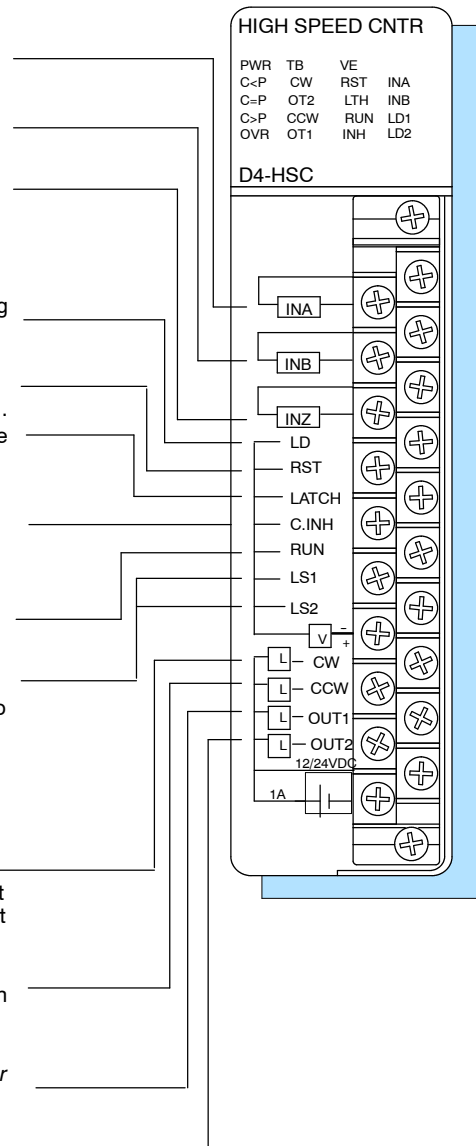
### Control Outputs

**CW**-You connect the output of this terminal to the appropriate terminal of your motor controller for *clockwise motion* when current count is **less than preset** and HSC RUN is ON, or when the output has been turned ON with RLL.

**CCW**-You connect the output of this terminal to the appropriate terminal of your motor controller for *counter-clockwise motion* when current count is **more than preset** and HSC RUN is ON, or when the output has been turned ON with RLL.

**OUT1** - As you approach a target position, you may need to *trigger motor deceleration*. A signal from this output can do that. This is used in the automatic home search algorithm also.

**OUT2** - When you reach your target (current count=preset), you may need to *activate a brake to stop the motor*. A signal from this output can do that.



### Y Data Type Equivalents for Some Functions

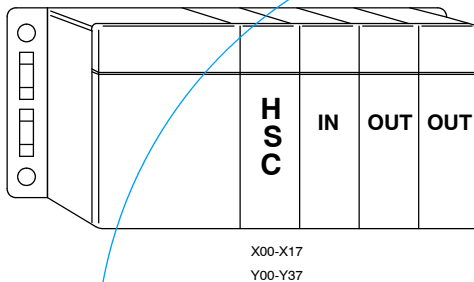
**LD, RST, LATCH, C.INH, and RUN** all have software equivalents built-in to the logic of the D4-HSC. These are Y data types that are discussed on the next page. Thus, you have your choice of either triggering these control inputs *externally* or accomplishing essentially the same task *internally from within your RLL*. They have been shaded to make them easy to spot.

**NOTE:** Use **external inputs** if immediate responses are needed. When a function is activated through RLL (Y-outputs in this case), the function will not activate until the I/O update has been performed. This delay is dependent on your CPU's scanning speed and the size of your program.

## X Input and Y Output Assignments

There are certain X's and Y's reserved by the D4-HSC. By convention, we will be referring to these assignments as  $X_{n+(z)}$  and  $Y_{m+(z)}$  where  $n$  and  $m$  are offset values based on which slot of the CPU base you have placed your HSC. The letter  $z$  will be some octal number that maps the X or Y to a specific input or output function. For example if you have the HSC in slot 0, and are using automatic addressing, then  $m$  and  $n$  will both be equal to zero. In such case, the data type assignments would be as shown below. All of this is explained in great detail in Chapter 4. You will also at that time be given a complete table of the X and Y assignments.

In this example we have placed the HSC in slot 0 of the CPU base. This simplifies X and Y identification because X's and Y's both start at 00.



For example:

$X_{n+6} = X_6 =$  Status of CW output

$Y_{m+3} = Y_3 =$  ON for HSC RUN

X No.	Function
$X_{n+0}$	ON if current count is greater than preset
$X_{n+1}$	ON if current count is equal to preset
$X_{n+2}$	ON if current count is less than preset
$X_{n+3}$	Latched ON if overflow occurs (reset with $Y_{m+1}$ )
$X_{n+4}$	Status of CCW output
$X_{n+5}$	Status of OUT2 (brake) output
$X_{n+6}$	Status of CW output
$X_{n+7}$	Status of OUT1 (deceleration) output
$X_{n+10}$	Status of Limit Switch 2
$X_{n+11}$	Status of Limit Switch 1
$X_{n+12}$	ON if doing a search for home position
$X_{n+13}$	ON if a sampling is being conducted
$X_{n+14}$	NOT USED
$X_{n+15}$	ON for missing terminal block
$X_{n+16}$	ON if external power supply for outputs is missing or OFF

Y No.	Function
$Y_{m+0}$	ON to reset OUT1 and OUT2 when in HSC run
$Y_{m+1}$	ON to reset overflow flag ( $X_{n+3}$ )
$Y_{m+2}$	Rising edge of this signal copies offset value into current count
$Y_{m+3}$	ON for HSC run
$Y_{m+4}$	Used to control CCW when not in HSC RUN or Home Search
$Y_{m+5}$	Used to control OUT2 when not in HSC RUN or Home Search
$Y_{m+6}$	Used to control CW when not in HSC RUN or Home Search
$Y_{m+7}$	Used to control OUT1 when not in HSC RUN or Home Search

## Putting It All Together

Up to this point, we have given you only the very basic information about the HSC module. The six chapters that follow will give you the additional information you need to make full use of the HSC. There are five basic steps for using the HSC.

### Five Steps for Using the HSC

Document **Install the Module and Connect the Wiring** — Chapter Two

Document **Understand How the Module Maps into the I/O Points, and How the Setup Information is Stored** — Chapter Three

Document **Setup the Counting and Control Input Parameters** — Chapter Four

Document **Setup the Control Outputs** — Chapter Five

Document **Setup Any Special Features, Such as Home Search or Sampling** — Chapter Six

### The Next Chapter

The next chapter will walk you through the installation and wiring before moving you on to Step 2.

---

# Installation & Wiring

---

In This Chapter. . . .

- How to install the D4-HSC
  - Connecting the Wiring
  - Connecting the Power Supply
  - Count Input Wiring Diagram
  - Control Input Wiring Diagram
  - Control Output Wiring Diagram
-

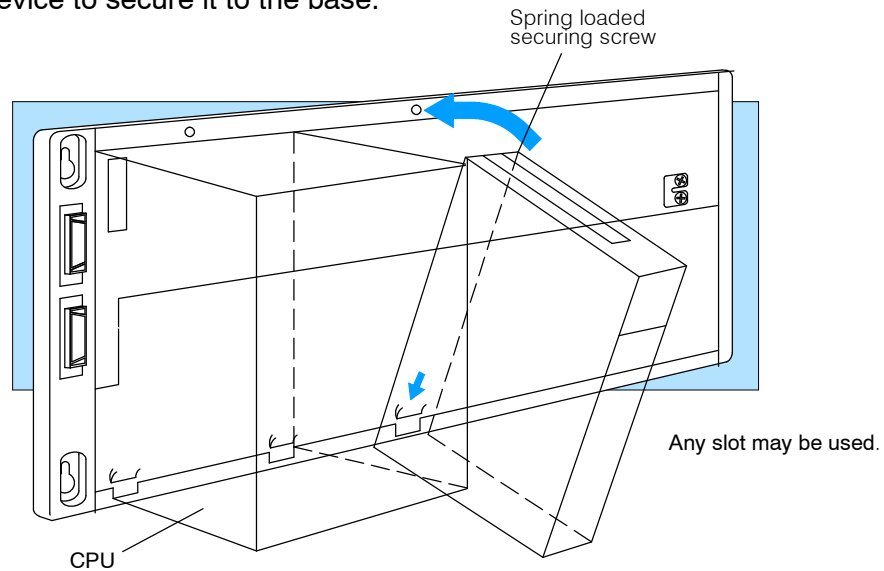
## How to Install the D4-HSC

**WARNING:** To minimize the risk of electrical shock, personal injury, or equipment damage, always disconnect the system power before installing or removing any system component.

The D4-HSC High Speed Counter module can be placed in any slot of the CPU base or an expansion base. It will not, however, work in a remote I/O base.

The following steps show you how to install the module.

1. Notice the D4-HSC has a plastic tab at the bottom and a screw at the top.
2. With the device tilted slightly forward, hook the plastic tab into the notch on the base.
3. Then gently push the top of the module back toward the base until it is firmly installed into the base.
4. Now tighten the screw at the top of the device to secure it to the base.



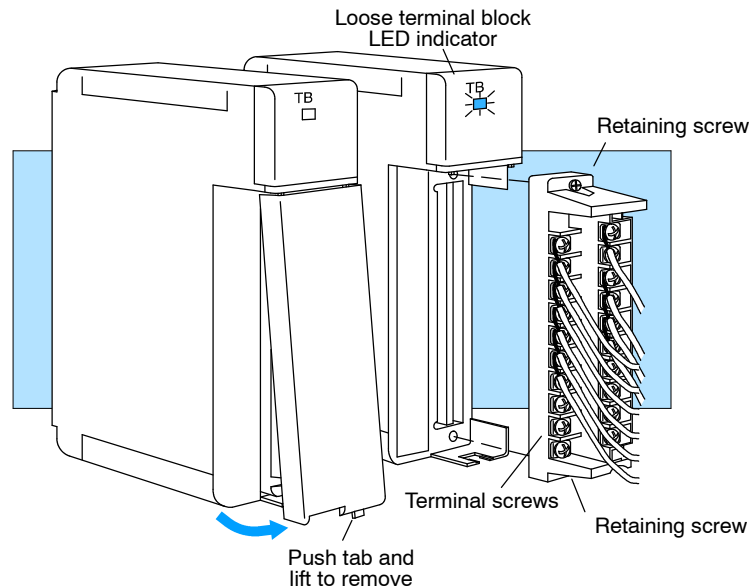
## Connecting the Wiring

### Wiring the D4-HSC Terminal Block

You must first remove the front cover of the module prior to wiring. To remove the cover press the bottom tab of the cover and tilt the cover up to loosen it from the module.

All DL405 I/O module terminal blocks are removable for your convenience. To remove the terminal block loosen the retaining screws and pull the terminal block away from the module. When you return the terminal block to the module make sure the terminal block is tightly seated. Be sure to tighten the retaining screws. You should also verify the loose terminal block LED is off when system power is applied.

**WARNING: Field device power may still be present on the terminal block even though the PLC system is turned off. To minimize the risk of electrical shock, disconnect all field device power *before* you remove the connector.**



**Wiring Guidelines** Use the following guidelines when you connect the wiring:

1. There is a limit to the size of wire the module can accept. The maximum size allowable for the D4-HSC is 14 AWG (Type TFFN or MTW). Other wire sizes may be acceptable--it really depends on the thickness of the insulation. If the insulation is too thick, the cover will not close properly.
2. Always use a continuous length of wire, do not combine wires to attain a needed length.
3. Use the shortest possible wire length.
4. Where possible use wire trays for routing .
5. Avoid running wires near high energy wiring.
6. To minimize voltage drops when wires must run a long distance , consider using multiple wires for the return line.
7. Avoid creating sharp bends in the wires.

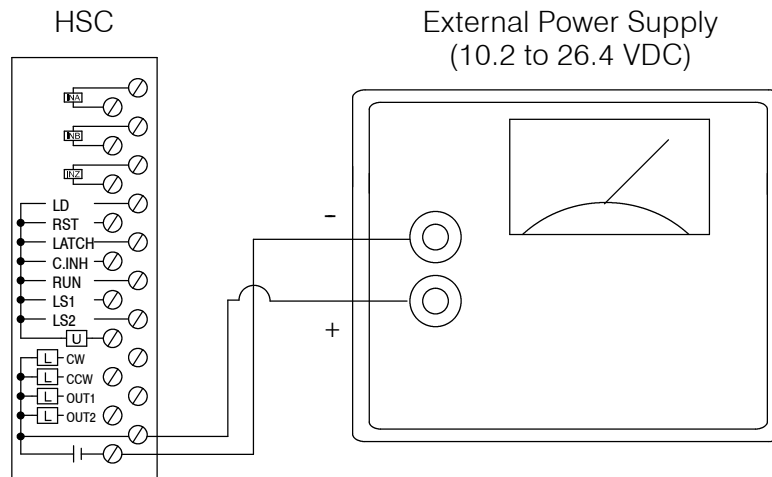
## Connecting the Power Supply

If you plan to use any of the HSC's outputs, in addition to the power supplied over the backplane of the CPU base or expansion base, you must have a power supply connected to the bottom two terminals of the D4-HSC (marked 12/24VDC). *If you do not use an external power supply in your application of the HSC, all functions except the CPU will operate normally but you will have a non-fatal error reported by the CPU.*

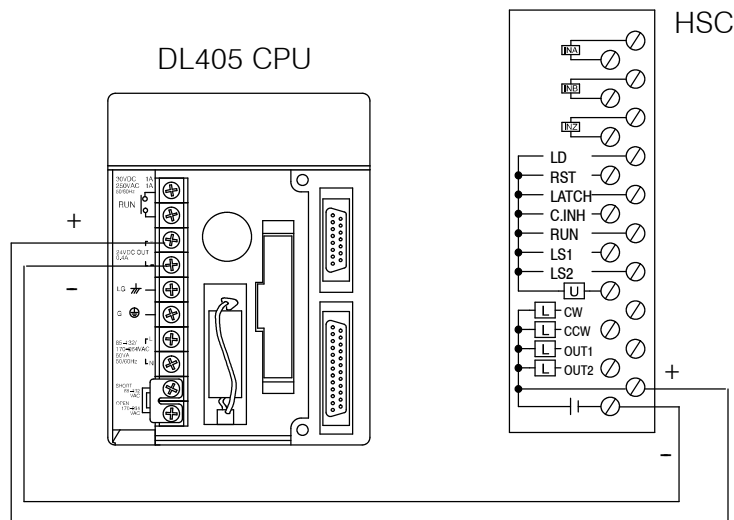
The power supply should be between 10.2 and 26.4 VDC and at least 400 mA (100 mA per output point) or more. We recommend 1 A. Your application will dictate the voltage and current requirements. Do not exceed the specifications shown in the tables of Chapter 1.

The CPU/Power Supply module of the DL-405 system has an *internal* 24VDC, 400mA power supply that can be used if it meets the specifications of your application.

### Using an External Power Supply

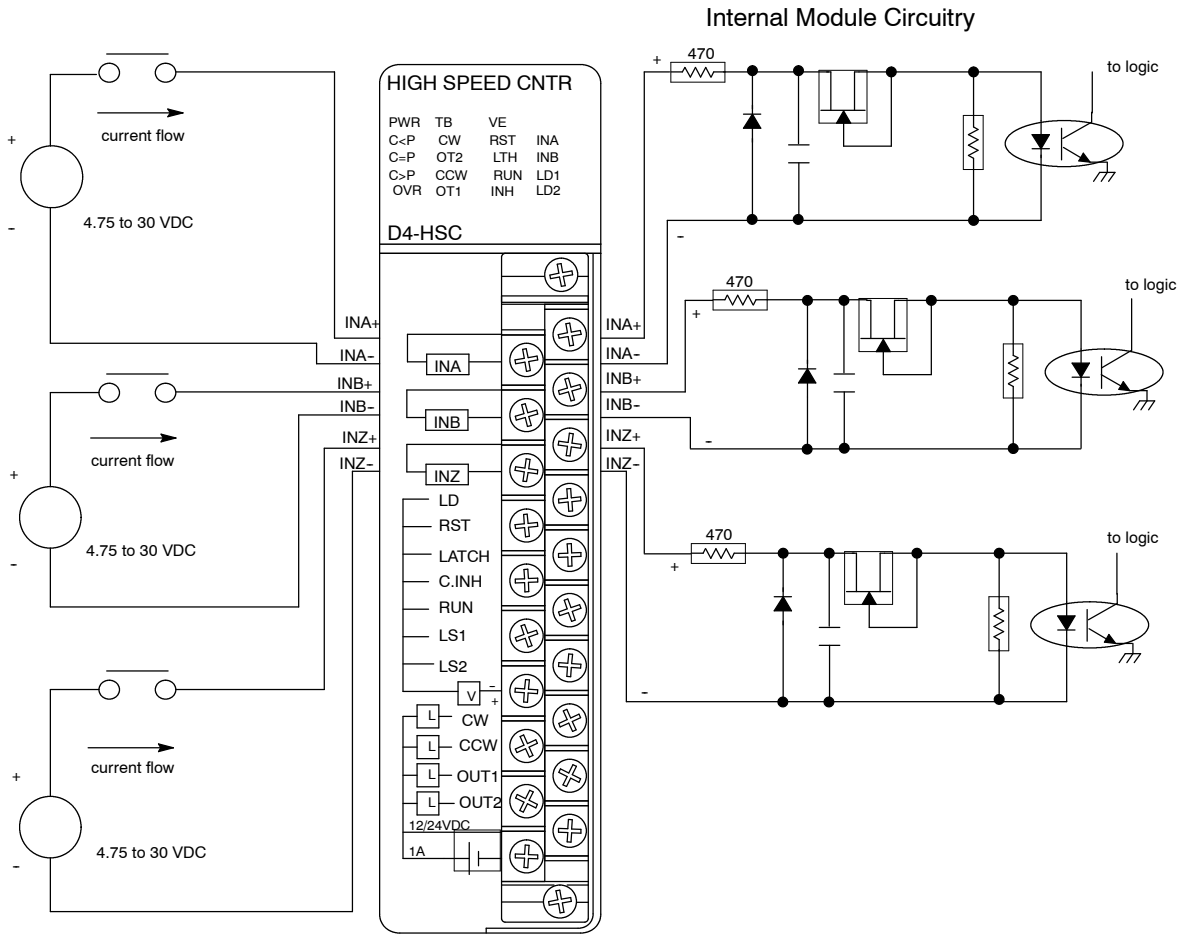


### Using the Internal DL-405 System Power Supply

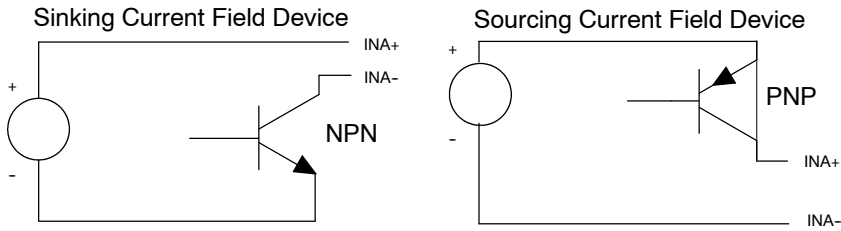


**NOTE: When using the internal power supply, make sure your application does not draw more current than the rated 400mA.**

# Count Input Wiring Diagram

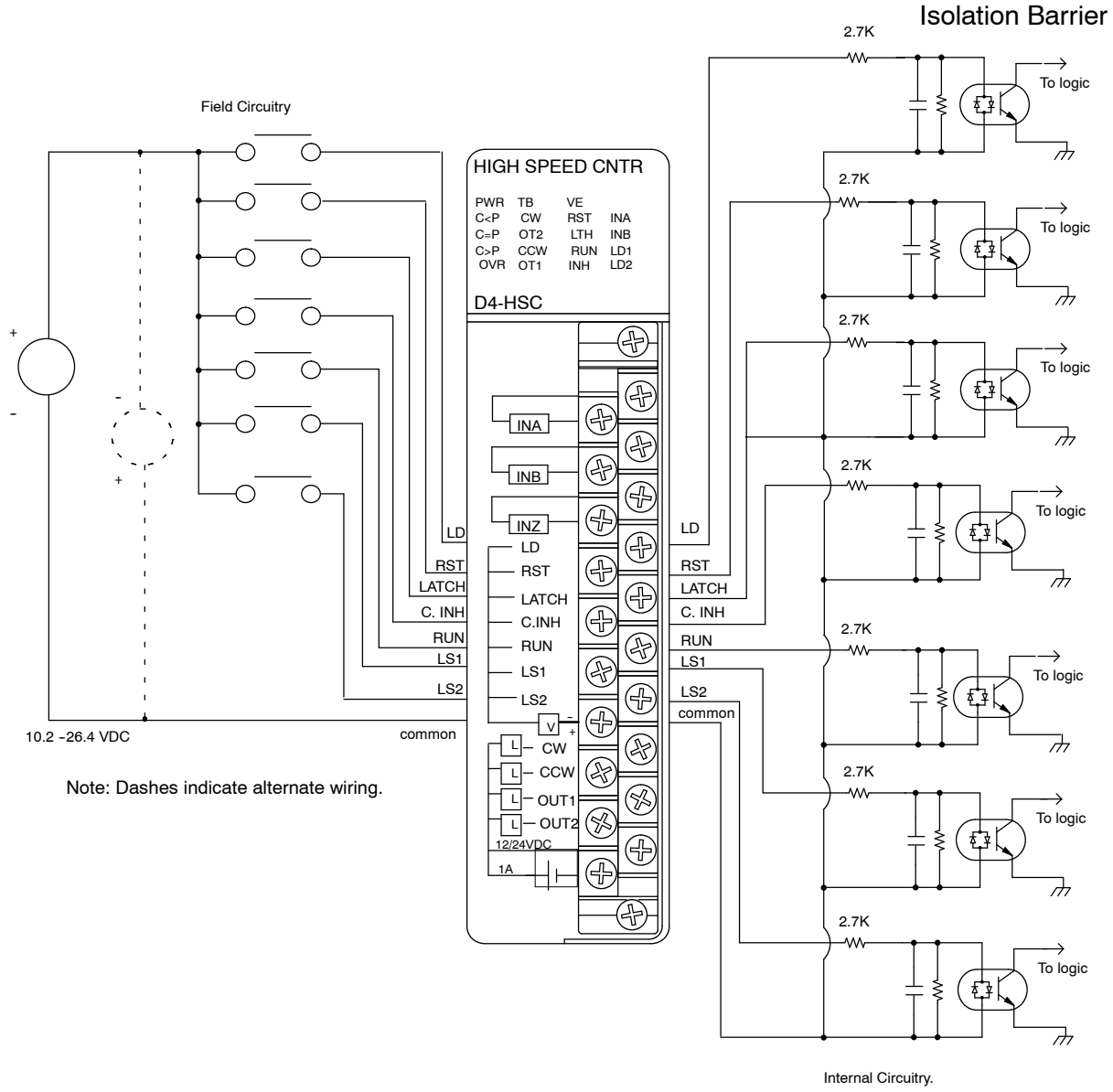


Installation and Wiring

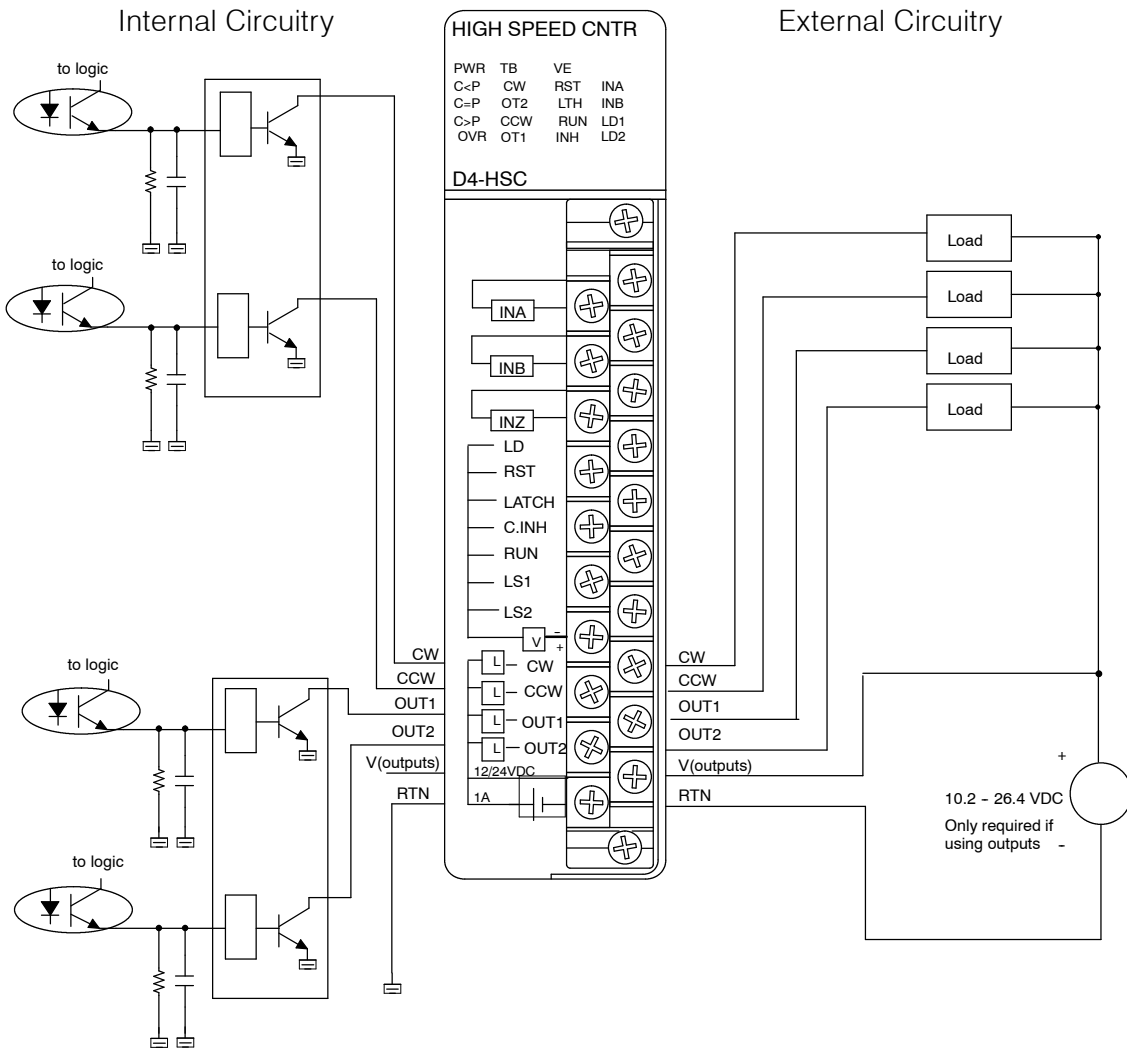




# Control Input Wiring Diagram



# Control Output Wiring Diagram



# Understanding the Operation

---

In This Chapter. . . .

- The Operating Basics
  - Assigning Your Data Types
  - Using the X and Y Assignment Table
  - Reading and Writing Shared Memory
  - Understanding How Numbers are Stored
-

## The Operating Basics

Although most readers of this manual will have had prior experience with the PLC and its various modules, there are some unique features belonging to the HSC, of which you must become familiar. These include:

- How the HSC reserves X (inputs) and Y (outputs).
- The X and Y assignment table of the HSC
- The concept of shared memory and its relation to V-memory
- Reading and writing data to shared memory
- How numbers are stored in shared memory

The pages that follow in this chapter will tell you in detail how all of the above operations are handled.

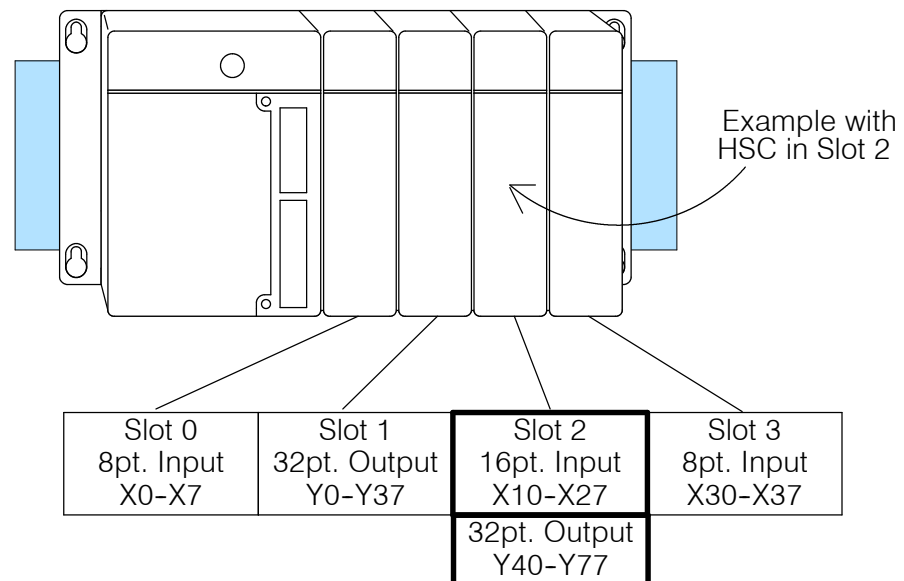
## Assigning Your Data Types

The DL405 family uses the octal (base 8) numbering system to designate I/O points. The letter **X** is always used to indicate inputs and the letter **Y** indicates outputs. Each I/O point also has a number associated with it, i.e. X12, Y32, etc. Assigning letters and numbers to I/O is referred to as “configuring your I/O”. I/O can be configured in one of two ways: **automatically** or **manually**. We will assume throughout this manual that you have used the automatic option. If you plan to manually configure your I/O, then you should read your DL405 User Manual in order to know how to perform the configuration.

### Automatic Configuration

The DL405 CPUs automatically examine any installed I/O modules (including specialty modules) and establish the correct I/O configuration and addressing on power-up. For most applications, you never have to change or adjust the configuration.

The I/O addresses are assigned using octal numbering, starting at X0 and Y0. The addresses are assigned in groups of 8, 16, 32, or 64 depending on the number of points for the I/O module. The following diagram shows the I/O numbering scheme for an example system. Notice that the automatic addressing feature assigns numbers to the I/O of the HSC module, just as it does for other modules.



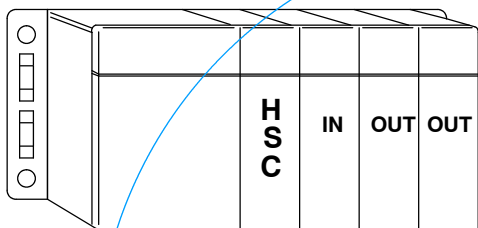
The HSC module will *automatically* consume 16 inputs and 32 outputs. The table on Page 5 of this chapter shows you how the HSC X's and Y's are assigned to specific functions.

**Using the X and Y Assignment Table**

If you are using automatic addressing, the DL405 CPU will look at how many I/O points are in the modules located to the left of the HSC, and automatically start numbering the X's and Y's of the HSC at the next available octal numbers. You need to know these number assignments in advance of writing your RLL, because they play an important role.

On the previous page, we placed the HSC in Slot 2. You could place it in any slot. If you place the HSC module in the first slot of your base, the process for immediately knowing the numbers assigned to your HSC X's and Y's is simple because you can merely substitute 00 for the values of *m* and *n* in the I/O assignment table. In such a case, you would not have to figure out what X's and Y's were committed to modules located to the left of the HSC, because there are none!

In this example we have placed the HSC in slot 0 of the CPU base. This simplifies X and Y identification because X's and Y's both start at 00.



X00-X17  
Y00-Y37

For example:

Xn+6=X6=Status of CW output

Ym+3=Y3=ON for HSC RUN

X No.	Function
Xn+0	ON if current count is greater than preset
Xn+1	ON if current count is equal to preset
Xn+2	ON if current count is less than preset
Xn+3	Latched ON if overflow occurs (reset with Ym+1)
Xn+4	Status of CCW output
Xn+5	Status of OUT2 (brake) output
Xn+6	Status of CW output
Xn+7	Status of OUT1 (deceleration) output
Xn+10	Status of Limit Switch 2
Xn+11	Status of Limit Switch 1
Xn+12	ON if doing a search for home position
Xn+13	ON if a sampling is being conducted
Xn+14	NOT USED
Xn+15	ON for missing terminal block
Xn+16	ON if external power supply for outputs is missing or OFF

Y No.	Function
Ym+0	ON to reset OUT1 and OUT2 when in HSC run
Ym+1	ON to reset overflow flag (Xn+3)
Ym+2	Rising edge of this signal copies offset value into current count
Ym+3	ON for HSC run
Ym+4	Used to control CCW when not in HSC RUN or Home Search
Ym+5	Used to control OUT2 when not in HSC RUN or Home Search
Ym+6	Used to control CW when not in HSC RUN or Home Search
Ym+7	Used to control OUT1 when not in HSC RUN or Home Search

### I/O Assignment Table

The following table provides all of the X and Y assignments for the HSC module. The letter “n” is the starting octal number for the inputs; and the letter “m” is the starting number for the outputs.

The Y outputs that are shaded have an external counterpart. That is, there are connecting terminals and labels on the module, so that field devices can enable these functions. Consequently, you can enable these functions either by setting the respective bits through ladder logic alone, or you can enable them by sending signals from external devices. You can use both methods within a single program. The Y outputs that are shaded in the table have external counterparts (LD=Ym+2, RUN=Ym+3, C.INH=Ym+10, LATCH=Ym+11, RST=Ym+12).

X No.	Function	Y No.	Function
Xn+0	ON if current count is greater than preset	Ym+0	ON to reset OUT1 and OUT2 when in HSC run
Xn+1	ON if current count is equal to preset	Ym+1	ON to reset overflow flag (Xn+3)
Xn+2	ON if current count is less than preset	Ym+2	Rising edge copies offset value into current count
Xn+3	Latched ON if overflow occurs ( reset with Ym+1 )	Ym+3	ON for HSC run
Xn+4	Status of CCW output	Ym+4	Used to control CCW when not in HSC RUN or Home Search
Xn+5	Status of OUT2 (brake) output	Ym+5	Used to control OUT2 when not in HSC RUN or Home Search
Xn+6	Status of CW output	Ym+6	Used to control CW when not in HSC RUN or Home Search
Xn+7	Status of OUT1 (deceleration) output	Ym+7	Used to control OUT1 when not in HSC RUN or Home Search
Xn+10	Status of Limit Switch 2	Ym+10	ON to temporarily suspend counting (inhibit counting)
Xn+11	Status of Limit Switch 1	Ym+11	Rising edge of this signal will latch the current count into shared memory
Xn+12	ON if doing a search for home position	Ym+12	If ON, it resets current count to zero
Xn+13	ON if a sampling is being conducted	Ym+13	OFF=Quadrature mode ON=UP/DOWN mode
Xn+14	NOT USED	Ym+14	Change state to change count direction
Xn+15	ON for loose or missing terminal block	Ym+15	Rising edge of this signal will invoke Home Search. You cannot use this if in HSC RUN.
Xn+16	ON if external power supply for outputs is missing or OFF	Ym+16	ON for x2 count operation (quadrature mode only/Ym+17 must be OFF)
Xn+17	ON if HSC detects an error	Ym+17	ON for x4 count operation (quadrature mode only)
		Ym+20	If OFF it will automatically reset current count to zero when current count=preset. If ON will not reset automatically unless count is at max or minimum.
		Ym+21	Rising edge of this signal will start the sampling feature
		Ym+22	Must be ON to enable external LD function
		Ym+23	ON to reset CW and CCW
		Ym +24	Not Used
		*Ym+25	ON to reset Home Search error.
		*Ym+26	ON to enable reset with INZ.
		**Ym+27	If turned ON before invoking Home Search, OUT2 (brake) will turn ON when Home is found.

\*This manual was written for the latest version of the D4-HSC. If you have an HSC that was purchased from another vendor, it may not support these features.

\*\*Available on HSC modules with production codes 9502 (Feb.'95) or later.

# Reading and Writing Shared Memory

## Contents and Data Flow

We provided a simple description of shared memory in Chapter 1, but it is important to understand it in detail. It resides in the HSC and it is shared by the DL405 CPU and the HSC itself. The table below shows how the memory is allocated. The data flow diagram is *from the perspective of the DL405 CPU*. Notice that it can only write four of the parameters from its V-memory to the shared memory, but it can read all seven parameters from shared memory into V-memory.

Not shown in the data flow chart is the reading and writing capability *from the HSC point of view*. The HSC reads and writes to the shared memory continuously, and it can do this for any of the seven parameters.

On the following page we will show you how to use your ladder logic program to read and write data to shared memory.

**Shared Memory Map**

**Data Flow Direction**

(From DL405 CPU Perspective)

Understanding Operation

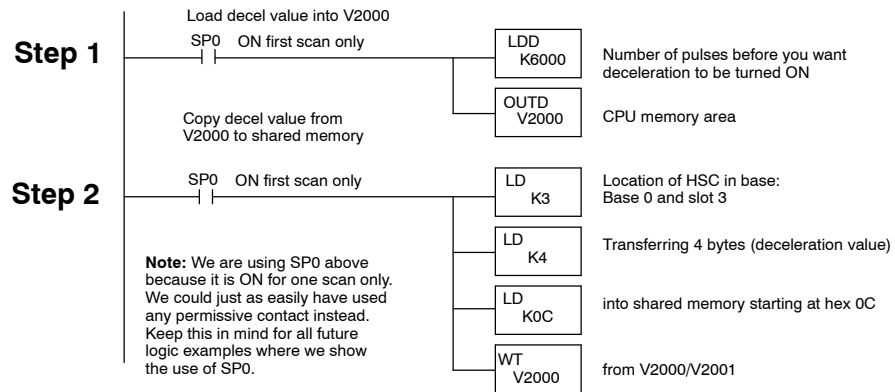
Data	Address (hex)	Address (octal)	V-Memory	Shared Memory
Current Count (4 bytes)	00 to 03	Range= -8388608 thru 8388607 Format: 8-digit BCD	00 to 03	Read ←
Offset Value (4 bytes)	04 to 07	Range= -8388608 thru 8388607 Format: 8-digit BCD	04 to 07	Read ← ↔ Write
Preset Value (4 bytes)	08 to 0B	Range= -8388608 thru 8388607 Format: 8-digit BCD	10 to 13	Read ← ↔ Write
Deceleration (4 bytes)	0C to 0F	Range= -8388608 thru 8388607 Format: 8-digit BCD	14 to 17	Read ← ↔ Write
Latched Count (4 bytes)	10 to 13	Range= -8388608 thru 8388607 Format: 8-digit BCD	20 to 23	Read ←
Timebase (2 bytes)	14 to 15	Range= 1 thru 9999 (Total time=above value x 3ms) Format: 4-digit BCD	24 to 25	Read ← ↔ Write
Sampled Count (4 bytes)	16 to 19	Range= 0 thru 8388608 Format: 8-digit BCD	26 to 31	Read ←



**NOTE:** In the vast majority of cases, you will use your RLL program to exchange data between the CPU and the HSC. However, if you have a handheld programmer, you can use AUX 47 to read and write shared memory. This can be helpful in some troubleshooting or startup situations. There are no commands available in *DirectSOFT* at this time to directly write data to shared memory.

**The Two-Step Process for Writing Data to Shared Memory**

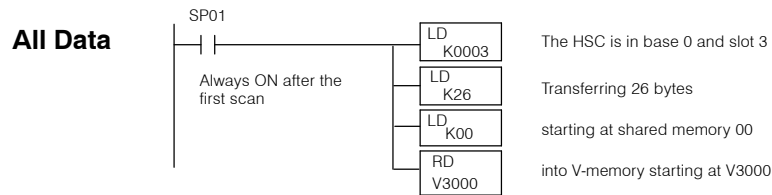
When using your RLL program to write data into shared memory, you must first load the data into V-memory. Then as a second step, you write it from V-memory into the shared memory area. The ladder logic below shows how you would use this two-step process to load a deceleration value into shared memory. Here we have loaded the deceleration value into its reserved address of shared memory. As you can tell from the table on the previous page, the starting address is hex 0C and it occupies 4 bytes.



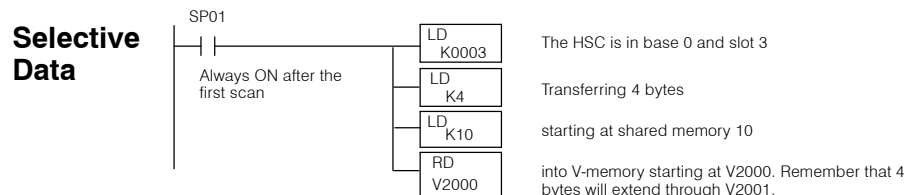
**NOTE:** If you are using the octal address of shared memory instead of hex, use LDA instead of LD. If you are loading a BCD number with more than 4 digits, use LDD.

**Reading Data From Shared Memory**

Quite often in your RLL you will want to move data the opposite direction--from shared memory to V-memory. The example below shows you how to read all 26 bytes (7 parameter values) from shared memory into V-memory.



You don't absolutely have to read all 26 bytes. You could only read those bytes you need. For example, you might wish to latch the count at some point. In that instance, the latched count value would be stored in shared memory (hex 10) and you could store it in V-memory for later use.



# Understanding How Numbers are Stored In Shared Memory

## Why is it important?

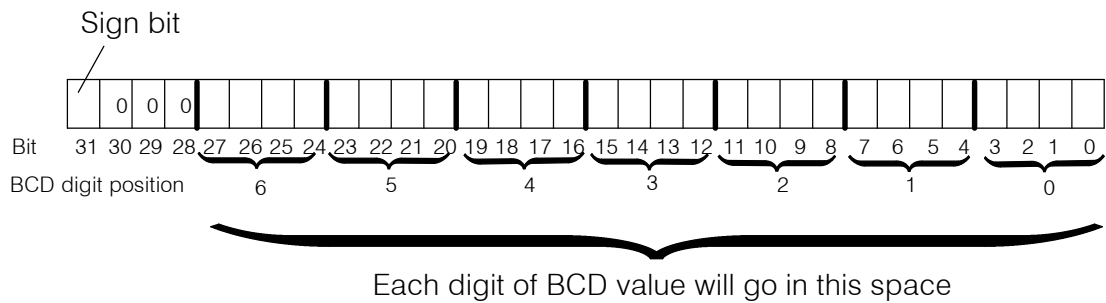
With the exception of the sampling timebase, all of the numbers stored in shared memory are 4 bytes in size. With the exception of the sampled count these numbers can either be positive or negative. If you are not familiar with a signed bit BCD format, this can be confusing.

You need to know how the numbers are stored in memory, so that you do not make mistakes when writing to shared memory or become confused when using your Watch Window in **DirectSOFT** to monitor the V-memory in which you are transferring shared memory data.

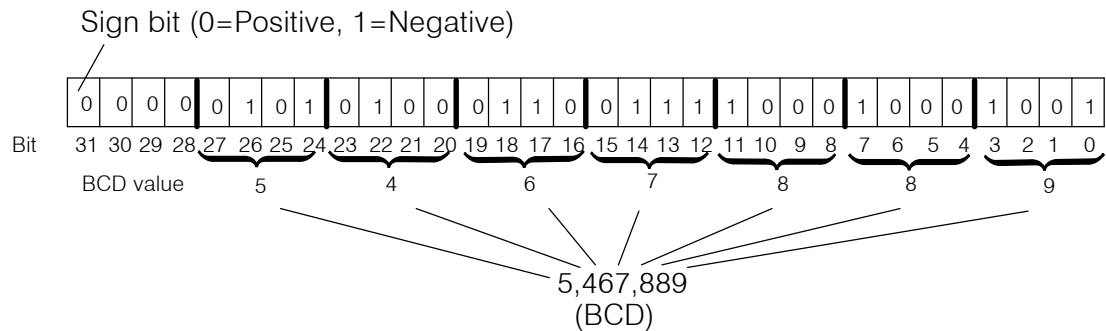
## Allocating the 4 Bytes of Memory

With the exception of the timebase and the sampled count, the numbers available for each parameter of shared memory can have a maximum positive value of 8,388,607 and a negative value of -8,388,608.

28-bits of the 4 byte space are used for the absolute value of the number and the remaining 4 bits are used to indicate the sign. The diagram below explains how all the bits are used:



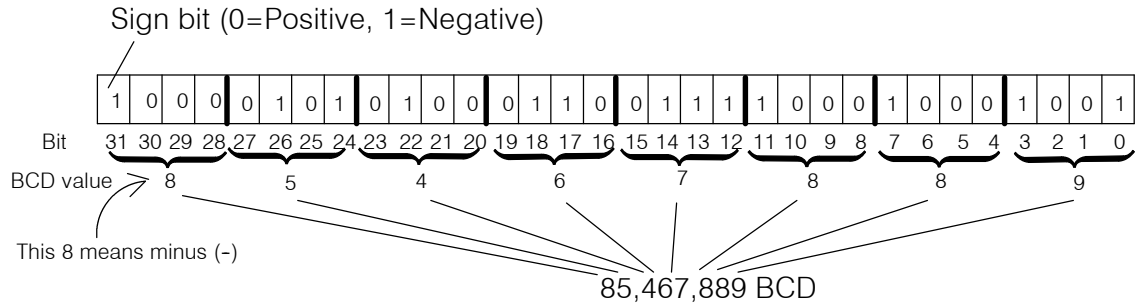
Take for example a preset value equal to 5,467,889. The diagram below shows how it is placed into the shared memory. Notice that the sign bit is zero. If you were to read this value from shared memory into V-memory; and then use the Watch Window to monitor its BCD value, you would see the BCD number 5,467,889.



But what would happen if this number was a negative number, say -5,467,889? The next page in this chapter will explain what you do.

**Dealing With the Negative Numbers**

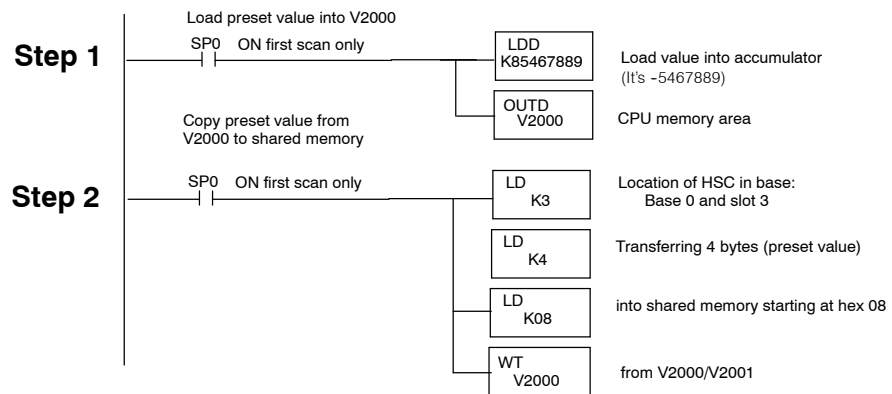
Continuing with our example from the previous page, the number -5,467,889 would have a 1 in the most significant bit (MSB) when placed in shared memory. What confuses many people is that this would show up in a Watch Window as 85,467,889 in BCD.



**Placing a Negative Number into Shared Memory**

Let's now take a look at some actual ladder logic to see how you would write a negative number into shared memory. In the example below, we are writing a preset into shared memory. Notice that we have used LDD to load values into the accumulator and OUTD to output them into CPU memory when they are more than 2 bytes in size. With the exception of the timebase, all of the shared memory data consumes 4 bytes of information each.

**Ladder Logic Example**



Notice that we loaded an 8-digit number with an 8 as the first digit. Remember you are loading a BCD number and not a decimal value. Remember that the MSB is always "8" when you are writing a negative number.

**The Next Chapter**

In the next chapter, you will learn how to put much of this knowledge to work by writing the initial relay ladder logic that you need in order to setup your application.

# Setting Up and Controlling the Counting

---

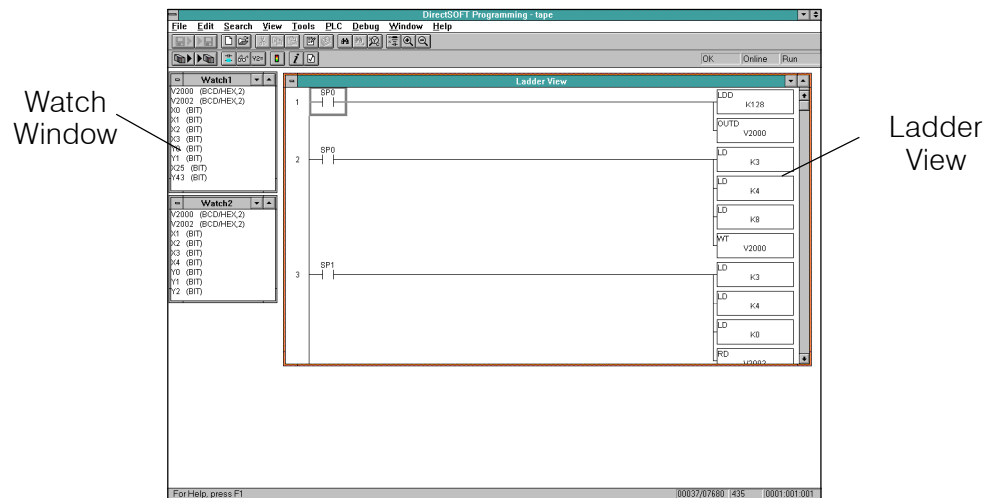
In This Chapter. . . .

- Introduction to Using **DirectSOFT**
  - Selecting the Counting Mode
  - Selecting the Counting Direction
  - Specifying an Offset
  - Specifying a Preset
  - Starting and Resetting the Current Count
  - Latching or Inhibiting the Current Count
  - Monitoring Overflow and Resetting Flags
-

## Introduction to Using *DirectSOFT*

You may recall an earlier example that showed you how to use the CPU RLL program to move the HSC parameters in and out of shared memory. The easiest way to create the RLL program is by using our Windows®-based software, **DirectSOFT**. We won't try to show you all of the **DirectSOFT** features here, but it may be helpful for you to understand a few simple concepts. You should refer to your **DirectSOFT** User Manual for a complete overview of the software features.

Once you enter the **Edit** mode, you will have several ways to enter your program elements. Below is a screen showing a portion of the program that has been entered while in the Edit mode. We are using the Ladder View.



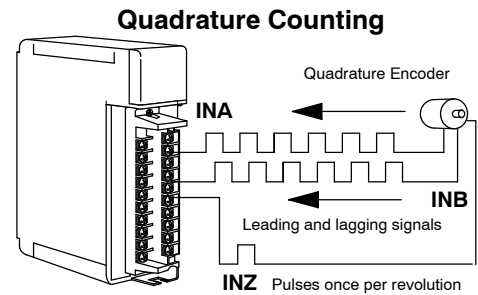
You can use a Watch Window by “clicking” on the Watch Window icon or by using the **Debug-New Watch** menu option. You can also use the hot key, **CTRL+SHIFT+F3**, to select the same option. You can open several Watch Windows if you like. Refer to your **DirectSOFT** documentation for details.

One example usage for the Watch Window feature (when working with the HSC) is to monitor the V-memory area where you might be exchanging information back and forth with the HSC's shared memory.

## Selecting the Counting Mode

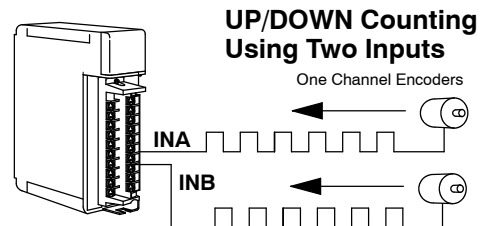
### Determining Which Mode to Use

You need to decide which mode of counting to use. If you are using a quadrature signal input device, then obviously you will need to use the quadrature mode. If you are using a single-channel encoder, you will want to use the standard UP/DOWN mode. The following page shows you the RLL for selecting the counting mode.

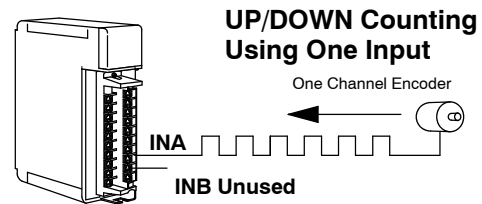


Quadrature encoders require that you connect to both the INA and INB terminals. They can sense direction and are inherently more immune to noise than single encoders. Quadrature encoders have a Z-marker that will aid in home search applications when connected to INZ by determining a zero or reference point in the angular displacement of the encoder's shaft. The single channel encoders (used for standard UP/DOWN counting) do not have Z-markers. When the HSC is not engaged in home search, you can use the Z-marker signal at INZ to reset the counter. We will show you how to do that in a moment on Page 4-10.

With standard counting you can use the two counting input signals (**INA** and **INB**) of the D4-HSC. One input is used for counting UP and the other used for counting DOWN. You can't use both inputs for the same direction of counting.

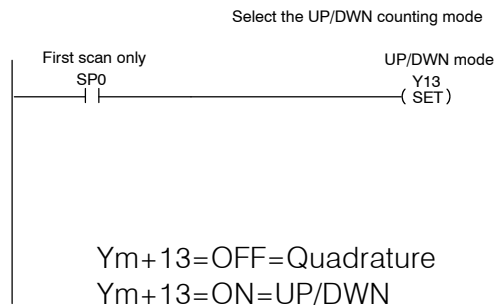


You could be using only one of the inputs if desired. In this case, the other input terminal should be left unwired. You control the direction of counting by the manner in which you set a certain bit in your control program (shown later).



### Ladder Logic for Determining the Counting Mode

You will recall from the I/O configuration table that the HSC uses Ym+13 to control the counting mode. If you have your HSC in slot 0, this means Y13 is the data point you use in your ladder logic. By default, the mode is set to count as if the signals at INA and INB are from a quadrature encoder (Ym+13 OFF). If you want standard, non-quadrature UP/DOWN counting you have to set Ym+13 to ON. Below is a sample rung of logic that selects the UP/DOWN mode. For simplicity, we have assumed the HSC is in slot 0. When SP0 is turned ON (first scan only), the HSC will be configured to count in the UP/DOWN mode.



## Selecting the Counting Direction

### How Ym+13 and Ym+14 Together Determine Counting Direction

The status of Ym+14 determines the direction of the counting, that is, UP or DOWN. If you are in the quadrature mode, the HSC will determine whether it is to count UP or DOWN by looking at the status of Ym+14 and seeing which of the signals (INA or INB) is leading.

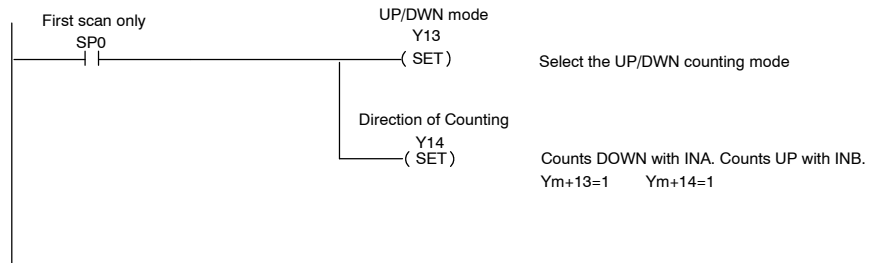
The HSC will determine direction of counting by looking at the status of Ym+14, the counting mode Ym+13, and (if a quadrature signal), whether INA or INB is leading or lagging. The table below summarizes how this information is used.

Mode Status	Direction	Criteria Used For Determining Direction
Ym+13=0	Ym+14=0	Counts UP if INA leads INB. Counts DOWN if INB leads INA (quadrature)
Ym+13=0	Ym+14=1	Counts UP if INB lead INA. Counts DOWN if INA leads INB (quadrature)
Ym+13=1	Ym+14=0	Counts UP with INA. Counts DOWN with INB (standard UP/DOWN)
Ym+13=1	Ym+14=1	Counts DOWN with INA. Counts UP with INB (standard UP/DOWN)

Using this criteria, the following sample ladder logic would cause the HSC to count in the UP/DOWN mode. The count from INA would be DOWN and the count from INB would be UP.

### Ladder Logic to Select Counting Direction

Assuming that the HSC is in Slot 0:



# Selecting the Counting Resolution

## Choose From 3 Resolution Settings

In the UP/DOWN mode, the resolution is fixed at 1x. However, in the quadrature mode, you can control which signal (INA or INB) and what edges of the signal cause a count change. This allows you to effectively double or quadruple the resolution. You have three choices:

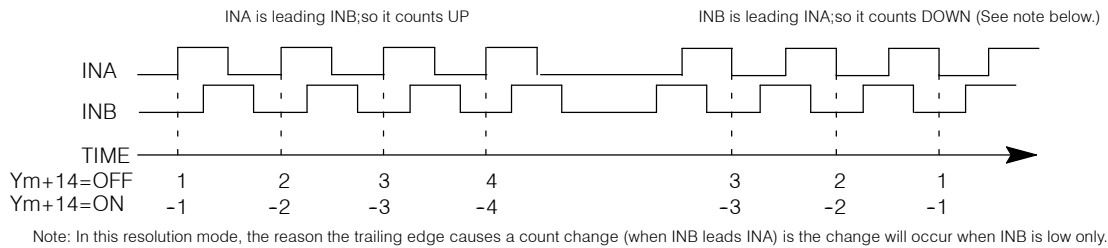
- 1x: One edge of INA causes count change
- 2x: Both edges of INA cause count change
- 4x: All edges of INA and INB cause count change

Ym+14 controls the direction of the counting, but Ym+16 and Ym+17 in combination control which signal and how many edges will cause the count to change.

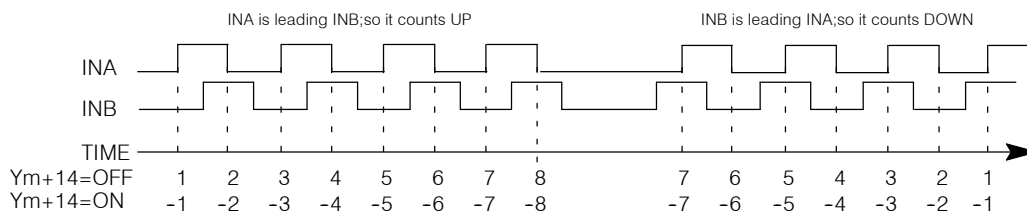
Ym+16	Ym+17	What Causes Count Change
OFF	OFF	1x: One edge of INA
ON	OFF	2x: Both edges of INA
OFF	ON	4x: All edges of INA and INB
ON	ON	4x: All edges of INA and INB

Y No.	Function
Ym+14	Change state to change count direction
Ym+15	ON will invoke home search
Ym+16	ON for x2 count operation (quadrature mode only/Ym+17 must be OFF)
Ym+17	ON for x4 count operation (quadrature mode only)

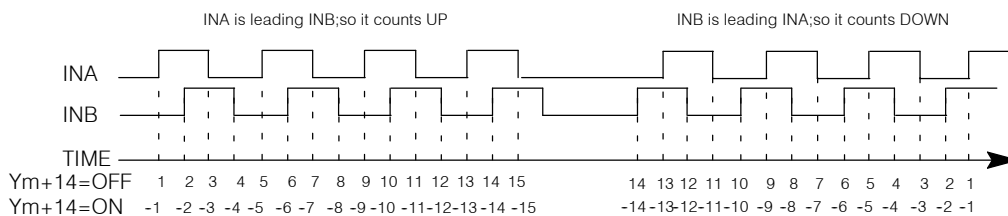
### Quadrature 1x Operation (One Edge: INA trigger)



### Quadrature 2x Operation (Two Edge: INA trigger)



### Quadrature 4x Operation (All Edges: INA and INB trigger)



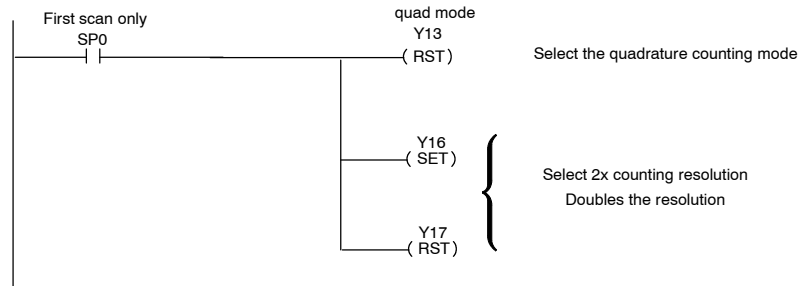


### Why Change the Counting Resolution?

The answer to “Why change the counting resolution?” is simply a matter of how much control you need for precise positioning. You may want to increase the resolution so that you receive a higher number of counts per encoder shaft revolution. This gives you more control.

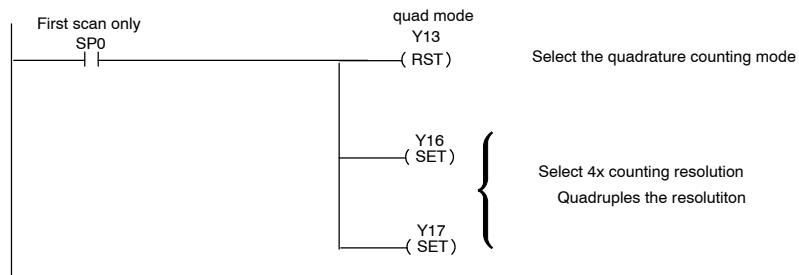
#### Example RLL: 2x Resolution

Assuming that the HSC is in Slot 0, the following logic would select 2x resolution:



#### Example RLL: 4x Resolution

Assuming that the HSC is in Slot 0, the following logic would select 4x resolution:



#### Default Setting

By default,  $Y_{m+16}=0$  and  $Y_{m+17}=0$ . This means that **you are in the 1x resolution mode for quadrature counting until you change the resolution in your ladder logic.**

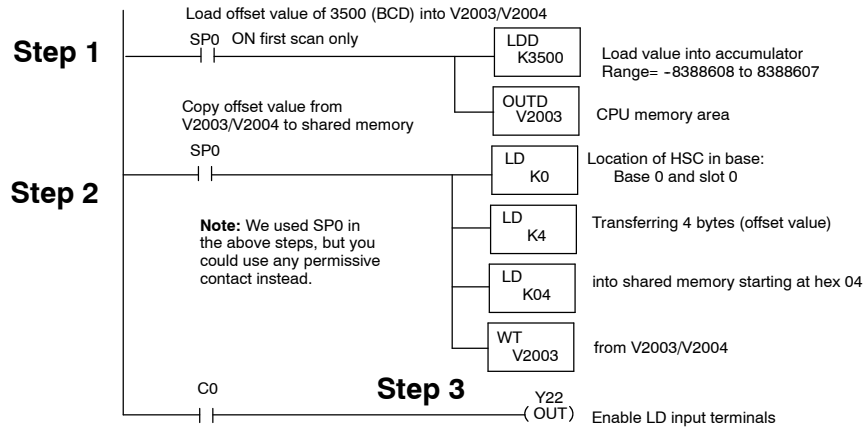
## Specifying an Offset

**What is an Offset?** This is an optional feature, but sometimes you may want to start your counting with some number other than zero. This is a perfect example of using an offset. You can also change the current count “on the fly” by using an offset. Either way, it is a three step process:

- Step1 - Load the offset value (Range=-8388608 to 8388607) into V-memory
- Step2 - Transfer the value out of V-memory by writing it to shared memory.
- Step3 - Write the offset value to the current count by either of two ways:
  - Send a signal from a field device attached to the external LD input of the module. Ym+22 must be ON to enable this feature. This is the **external method**.
  - Use your ladder logic to turn ON Ym+2. This is the **internal method**.

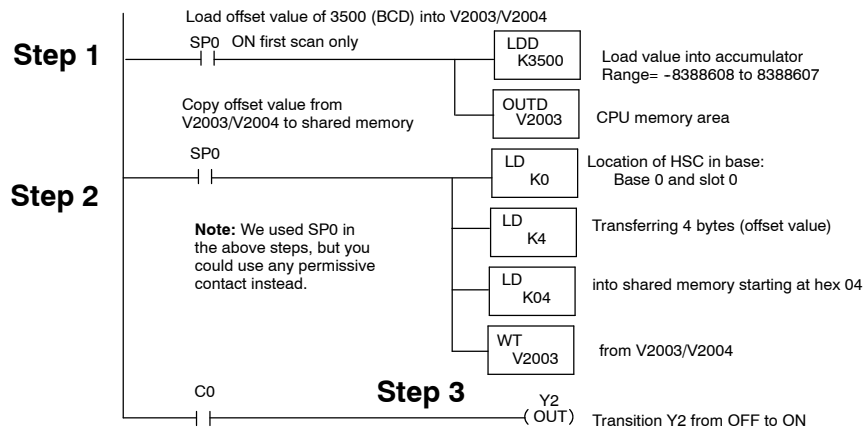
### External Method

For simplification purposes, let’s look at an example where you have your HSC in Slot 0 of your base. The rung of logic shown below will prepare the HSC to use an offset value. Then, if you have a field device hooked to the LD terminal connections, and you turn the device ON, the HSC will will copy the value stored in the shared memory address 04 to 07 (offset) to the current count. In contrast, if Ym+22 was OFF, the HSC would not respond to any signal at the LD connection. In the example below, C0 will determine if the offset gets written to current count.



### Internal Method

If you are using the internal method, everything would remain the same except the final rung of logic (Step 3). Here you would use C0 to turn ON Y2.



## Specifying a Preset

### What is a Preset?

Another way of saying “preset” is to use the word “target”. When you place a preset in shared memory, it tells the HSC, “This is my target!”. Your target can be any number of pulses in the range -8388608 thru 8388607. (Remember, negative presets must have an “8” in front of them.)

**NOTE:** If you do not use a preset (i.e. you have no target count), **always set Ym+20 = ON** to ensure continuous counting without inadvertent resets.

### How Does the Preset Affect the Outputs?

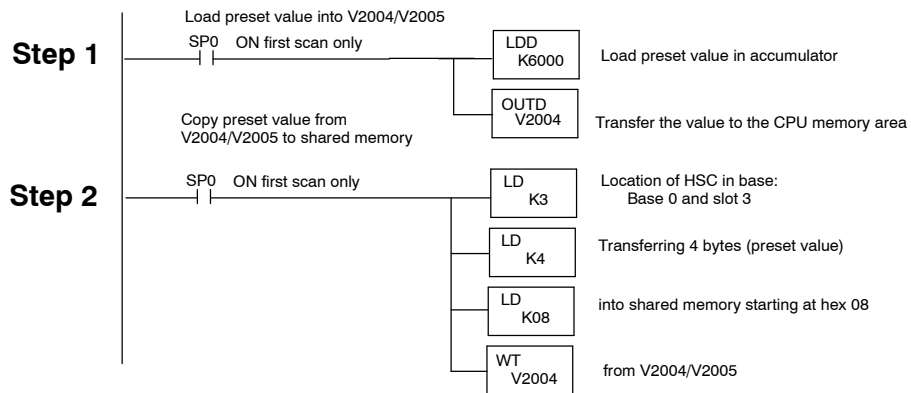
Each time your ladder logic instructs the HSC to enable your HSC outputs, the HSC will look at three parameters that are stored in shared memory in order to know which output to turn ON:

- Step1 - Current count
- Step2 - Preset
- Step3 - Deceleration.

The HSC then makes a decision on what to do with the outputs, CW, CCW, OUT1 and OUT2 based on the relationship that it sees. (On Pages 5-4 and 5-5, we will show you how the relationship between preset and current count determines the status of each output.)

### Loading the Preset Into Shared Memory

First, you load a preset into shared memory using the same 2-step procedure shown earlier:



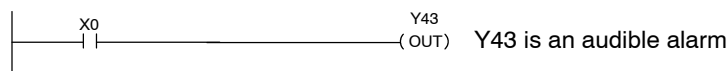
**NOTE:** Preset may be loaded at any time, but it is not accepted by the HSC until the HSC run bit transitions from off to on.

### Checking the Status of a Preset Relative to Current Count

There are three X inputs in the I/O assignment table that report the status of preset versus the current count. You can use the status of each of these in your RLL to trigger events. Here is the portion of the table showing you the three X assignments.

X No.	Function
Xn+0	ON if current count is greater than preset
Xn+1	ON if current count is equal to preset
Xn+2	ON if current count is less than preset

For example, the one line of logic below could turn on an alarm when the current count **exceeds** preset. Assume that the HSC is in Slot 0:



## Starting and Resetting the Current Count

### Starting the Counter

Assuming you have installed the HSC module in the base properly and connected an encoder to the proper inputs, you are ready to start the counting process. All that is required is to put the PLC in RUN mode, and have the encoder (or encoders) sending valid signals. With this done, the HSC will start counting any pulses received at INA or INB. It will automatically be storing the accumulated count as the current count in the shared memory.

### Automatically Resetting the Counter

Ym+20 determines when the current count is reset to zero. You have two options:

- (A.) If Ym+20=OFF, the counter will reset to 0 when current count = preset.
- (B.) If Ym+20=ON, the counter will reset to 0 when it reaches the maximum number (8388607) or the minimum number (-8388608).

### Internal Reset

You can also use Ym+12 to reset your counter. Simply turn it ON in your ladder logic. As long as you have Ym+12 ON, the current count will remain zero.

### External Reset Using RST

In order to reset the counter externally, you can turn ON the device connected to the RST terminals of the HSC. As long as this signal stays HIGH, the current count will remain zero.

### Summary of Reset Count Relays

The chart below summarizes the Y output assignments discussed above.

Y No.	Function
Ym+12	When set to ON, HSC resets current count to zero.
Ym+20	If OFF, counter will reset to 0 when current count = preset. If ON, counter will reset when count is at max. or min.

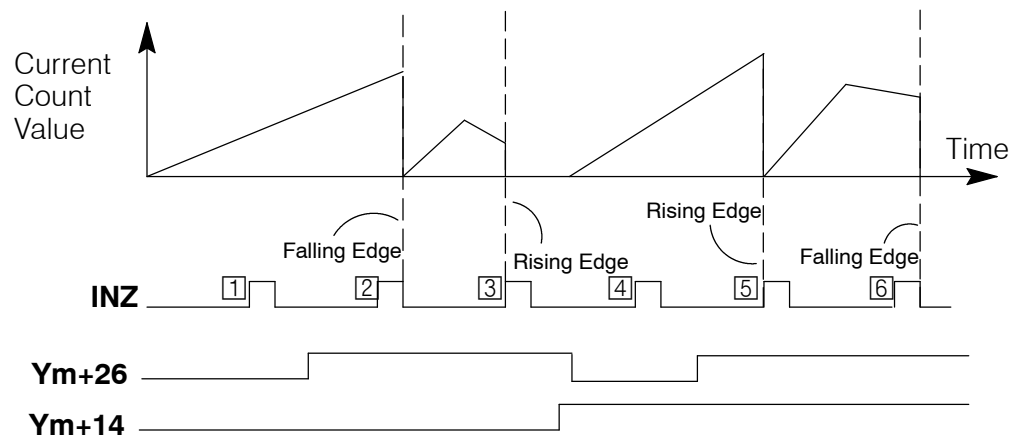
### External Reset Using INZ

If you have not invoked Home Search with Ym+15, you can use INZ to reset the counter. You enable the INZ reset feature by turning Ym+26 ON. Since direction of the encoder shaft rotation affects when the Z-marker will send the reset pulse, the status of Ym+14 (change direction output) affects which edge of the pulse actually triggers the reset. By using INZ to reset the counter, you are able to trigger reset at the same shaft position every time.

The table below shows the relationships of the various outputs, the count direction, the INZ signal and which part of the pulse actually resets the counter:

Home Search Ym+15	INZ Reset Ym+26	Count Direction- Ym+14	Characteristics of the Reset Using INZ
OFF	ON	OFF	Resets on rising edge when counting DOWN Resets on falling edge when counting UP
OFF	ON	ON	Resets on rising edge when counting UP Resets on falling edge when counting DOWN

### An Example of INZ Resetting Current Count Under Various Conditions



- 1 On the first pulse of INZ, there is no reset because Ym+26 is OFF.
- 2 On the second pulse of INZ, there is a reset because Ym+26 is ON. Ym+14 is OFF and we're counting UP; so the counter resets on the falling edge.
- 3 On the third pulse of INZ, there is a reset because Ym+26 is ON. Ym+14 is OFF and we are counting DOWN; so the counter resets on the leading edge.
- 4 On the fourth pulse of INZ, there is no reset because Ym+26 is OFF.
- 5 On the fifth pulse of INZ, Ym+26 is ON and Ym+14 is ON. Because we were counting UP, there is a reset on the rising edge.
- 6 On the sixth pulse of INZ, Ym+26 is ON and Ym+14 is ON. Because we were counting DOWN, there is a reset on the falling edge.

## Latching or Inhibiting the Current Count

### What Does Latching Do?

There may be an application where you want to store the current count after a certain amount of time passes or when a certain event has taken place. You can capture this information and store it in shared memory without stopping the counting. This is called “latching”. It gives you a “snap shot” of the pulse count for later use in your program.

### How Do You Trigger the Latching Process?

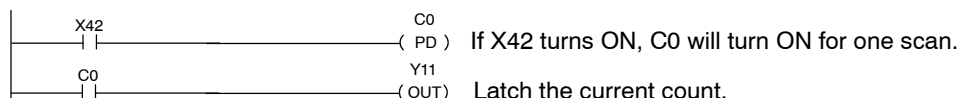
You have two options for triggering the latching process:

- You can do it **externally** via a field device attached to the terminals marked “LATCH”, or
- you can do it **internally** by using Ym+11.

In both cases, the latching will take place each time there is a transition from OFF to ON. If you leave either the field device or Ym+11 in the ON state, it will only latch one time at the OFF to ON transition. You will have to do a separate transition from OFF to ON every time you make a LATCH request in order for values to actually be stored.

### Sample RLL for Latching

Here is a short segment of ladder logic showing you how to latch the count by using the internal output Ym+11. We have assumed your HSC is in Slot 0 of the base. We have used a one-shot command here so that C0 and Y11 would be ON for only one scan when the CPU sees that X42 is ON.



### What Is Meant By Inhibiting the Count?

There may be some reason, during the course of the program, that you want the counter to temporarily suspend its counting without resetting or in any way disturbing the the current count. This is what the “inhibiting” feature does. When this feature is ON, inputs from INA and INB are ignored.

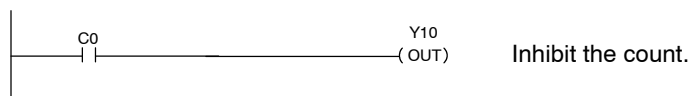
### How Do You Inhibit the Count?

You have two options here also:

- You can do it **externally** via a field device attached to the terminals marked “C.INH”.
- You can do it **internally** by using Ym+10.

### Sample RLL for Inhibiting the Count

Here is a short segment of ladder logic showing you how to inhibit the count by using the internal output Ym+10. We have assumed your HSC is in Slot 0 of the base.



### Summary of Latch and Inhibiting Output Relays

The chart below summarizes the Y output assignments discussed above.

Y No.	Function
Ym+10	If turned ON, the HSC will temporarily inhibit (suspend) the count.
Ym+11	If turned ON, the HSC will latch the current count into shared memory. Rising edge triggered.

## Monitoring Overflow and Resetting Flags

### What is a Counting Overflow?

As mentioned earlier, the HSC counter can count UP to +8388607 maximum or count DOWN to -8388608. If you pulse the counter beyond these two maximum counts (and Ym+20 ,reset, is OFF), the following will happen:

- Counting UP past +8388607 will cause the count to wrap around and start counting from -8388608 UP. (i.e. -8388608, -8388607, -8388606, etc.)
- Counting DOWN past -8388608 will cause the count to wrap around and start counting from +8388607 DOWN (i.e. 8388607, 8388606, 8388605, etc.).

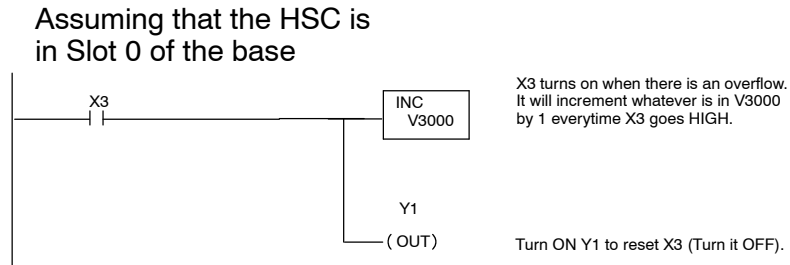
If this happens, the overflow LED will come ON to let you know this has occurred. It would remain ON until power is removed, or you manually reset it using by Ym+1.

### Status Flag for Overflow

In addition to turning on the OVF LED when there is a counting overflow, the HSC will also mirror the status in Xn+3. This flag will stay ON until Ym+1 is turned ON or power is removed. You can use this flag to sound an alarm, trigger other events, etc.

### Tracking Overflows

You may want to track the total number of overflows that occur. The program below shows you some example logic that could accomplish this task.



When Ym+1 goes HIGH, the overflow flag will not be set again until an overflow occurs again.

### Summary of Input and Output Relays for Overflow and Flag Reset

The chart below summarizes the X and Y output assignments discussed above.

X or Y No.	Function
Xn+3	If ON, it means that you are in overflow. If OFF, it means you are not in overflow.
Ym+1	This output relay will reset (turn OFF) the overflow flag Xn+3 and the OVF LED.

# Controlling the Outputs

---

In This Chapter. . . .

- Introduction
  - Manual Output Control
  - Automatic Output Control
  - Using the Direction Outputs
  - Timing Diagrams for HSC RUN
  - Using the Speed Outputs with the Direction Outputs
-



## Introduction

### The 4 Control Outputs

The HSC has four output signals:

- **CW**—a signal usually connected to the motor controller's input that has been programmed for clockwise motion.
- **CCW**—a signal usually connected to the motor controller's input that has been programmed for counter-clockwise motion.
- **OUT1**—a signal usually connected to the motor controller's input that has been programmed for deceleration.
- **OUT2**—a signal usually connected to the motor controller's input that has been programmed to brake.

### All the Output Signals Look the Same

If you were to look at each of the above four signals with a VOM or an oscilloscope, they would all look the same. They are simply constant DC output signals whose voltage is directly dependent on the rating of the external power supply. What is important is the point in time in your RLL program that you turn them ON or OFF and how your motor controller (to which they are connected) is either hard-wired or programmed.

### The 4 Y Output Relays for Turning ON Each Control Output

The HSC has four output relays for turning ON each of the HSC control outputs:

- **Ym+6**—turns ON CW.
- **Ym+4**—turns ON CCW.
- **Ym+7**—turns ON deceleration signal.
- **Ym+5**—turns ON brake signal.

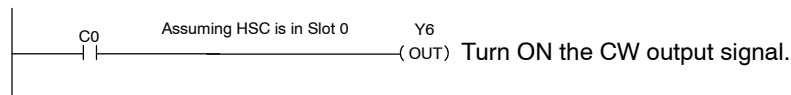
### Using the Y Output Relays to Control the Outputs is Optional

You do not have to use the output relays shown above, but they are there if you need them. In the pages that follow, we will show you how to either invoke a program that will turn ON and OFF each of the control outputs in an ordered sequence **automatically**, or how to do it **manually** by using the 4 internal relays just described. Both ways require ladder logic, but they are very different in terms of flexibility. We will be explaining each method in great detail.

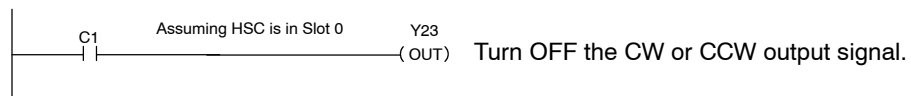
## Manual Output Control

### Using Ym+6 or Ym+23 to turn Direction Outputs ON or OFF.

Ym+4, Ym+5, Ym+6 and Ym+7 manually control the outputs CCW, OUT2, CW and OUT1 respectively. You can turn them ON at any time in your ladder logic, with the exception being that you cannot have CW and CCW on at the same time and you cannot be in Home Search or HSC RUN modes. For example, here is a rung of logic that would turn ON the CW output:



Using the above method, you could turn OFF CW and CCW by using Ym+6 and Ym+4 respectively or you can turn off (reset) CW or CCW by turning ON Ym+23:



## Automatic Output Control

### Also Called HSC RUN

Automatic output control is also called, HSC RUN. It is simply a mode that allows the HSC to automatically control when each of the 4 outputs turn ON by looking at relative values of current count and preset. Upon entering HSC RUN, there is a built-in algorithm that determines whether CW or CCW turns ON. The algorithm also determines the sequence for turning the HSC outputs ON or OFF.

**NOTE:** When you invoke HSC RUN, the HSC looks at current count and preset **only when it enters** HSC RUN. At that time, it uses the relationship between these two variables to decide whether it should turn ON either CW or CCW. If you change the relationship during the course of your program, it will not take effect until you exit HSC RUN and then invoke HSC RUN again. Also, preset and decel values in shared memory (at the time HSC RUN is invoked) are used until HSC RUN is exited. *If new values are written during HSC RUN=ON, they will be ignored until you exit HSC RUN and re-enter again.*

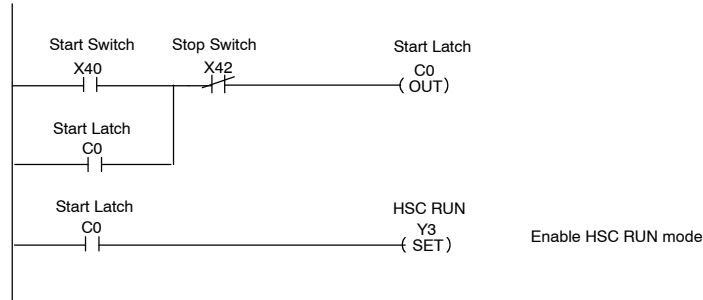
### How Do You Invoke HSC RUN?

You have two ways in which you can invoke HSC RUN --**either externally or internally.**

- Externally, you turn ON the field device connected to the RUN terminals.
- Internally, you turn ON Ym+3.

### Example of Using Ym+3 to Activate HSC RUN

Below is a short segment of ladder logic showing how to use Ym+3 to turn ON HSC RUN. We are again assuming that the HSC is located in Slot 0 of the base. Notice that we have included in our sample logic a start latch using an internal control relay of the DL405 CPU, combined with a start and a stop switch connected to the terminals of an I/O module:



## Using the Direction Outputs

### What is a Direction Output?

There are two direction outputs for the HSC: **CW** (clockwise) and **CCW** (counter-clockwise). The actual turning of a motor shaft in these two directions is controlled by the internal logic of the motor controller that you are using, not the HSC.

### How the HSC Knows When to Turn ON CW or CCW

As mentioned earlier, if you have not invoked HSC RUN or Home Search, you can turn CW and CCW ON or OFF in your ladder logic at any time using either Ym+6 (CW) or Ym+4 (CCW).

You can, on the other hand, let the HSC automatically decide when to turn them ON or OFF by invoking HSC RUN. The HSC will decide if it should turn ON CW or CCW **based on the relationship between preset and current count at the time HSC RUN is entered**. The table below shows the relative states of CW and CCW based on current count and preset at time of entry:

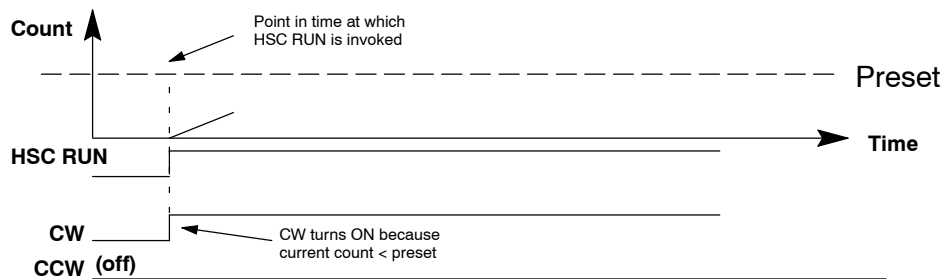
Relationship Upon Entering HSC RUN	CW	CCW
Current count is less than preset (cc<preset)	ON	OFF
Current count is equal to preset (cc=preset)	OFF	OFF
Current count is greater than preset (cc>preset)	OFF	ON

## Timing Diagrams for HSC RUN

The following diagrams show you the timing relationships between HSC RUN and the direction outputs for the three possible scenarios of preset versus current count.

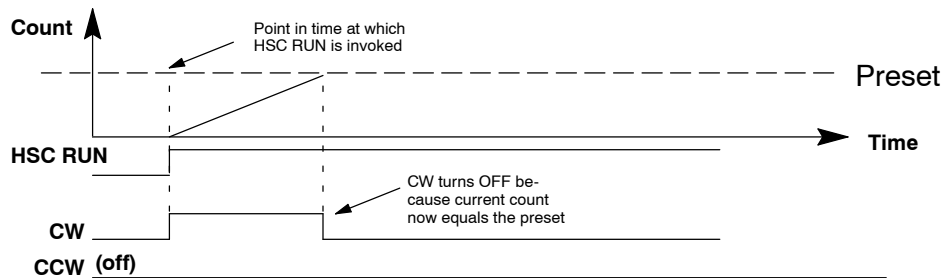
As you can see from the following diagram, the clockwise output (CW) turns on because the current count is less than the preset.

### Current Count < Preset



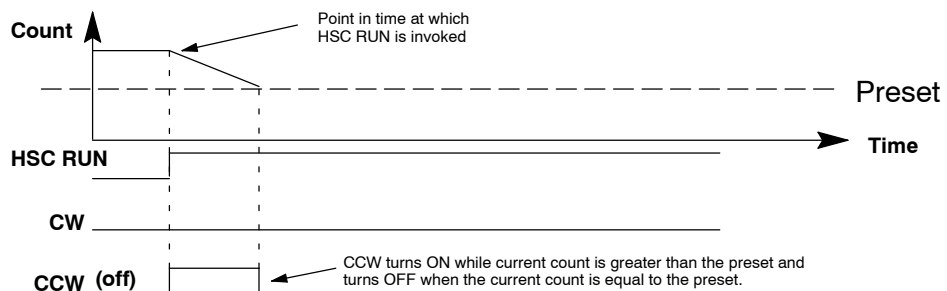
As the current count continues to approach the preset, the HSC will monitor the relationship between the current count and the preset. When the current count is equal to the preset, the HSC automatically turns off the clockwise output (CW).

### Current Count = Preset



There may be occasions when you are counting down towards a preset. For example, you may have loaded a negative preset, or, you may have used an offset to change the current count so that it is above the preset. The following diagram shows how the HSC can also automatically control the counter clockwise (CCW) output.

### Current Count > Preset



## Using the Speed Outputs with the Direction Outputs

**What are the Speed Outputs?** There are two speed outputs: OUT1 (deceleration) and OUT2 (brake). While ON, these two signals are constant DC signals. Their ability to decelerate or brake a motor is entirely dependent on the logic of the motor controller to which they are attached. The motor controller must do the speed changing. These signals merely initiate these functions in your controller.

**2 Ways to Initiate OUT1 and OUT2** You can initiate OUT1 (decel) or OUT2 (brake) in either the HSC RUN mode, or invoke them in your ladder logic using the appropriate Y outputs. To use the manual method, you turn OUT1 ON by turning ON Ym+7. In a similar fashion, when you turn ON Ym+5, OUT2 will turn ON.

**Using HSC RUN to Initiate OUT1** When using HSC RUN, OUT1 (decel) is a relative value that establishes two different points (relative to preset) at which OUT1 will turn ON. For example if your preset is 3000, and you have stored a deceleration value of 1000, the deceleration will start (OUT1=ON) when the count is at 2000 counts (counting UP to preset) **or** 4000 counts (counting DOWN to preset). The HSC derives these numbers by looking at the number you have stored in shared memory hex 0C, then adding and subtracting this value from the preset.

In Figure A at the top of the next page, we have illustrated how OUT1 is triggered when the current count equals 4000. It does this because 4000 is the point at which the current count is within 1000 pulses (decel value) from preset as we count DOWN toward preset (current count > preset). In Figure B, we show how OUT1 is triggered when the current count is at 2000, because in this second example we are approaching preset counting UP (current count < preset).

As you can see, when the current count **reaches** the threshold area (defined by your stored decel value), from either counting direction, the decel signal (OUT1) will turn ON and stay ON until it is reset. You can reset OUT1 either with Ym+0 (as we have done in our examples) or exiting and re-entering HSC RUN.

**How OUT2 is Initiated in HSC RUN** When using HSC RUN, OUT2 (brake) will automatically turn ON **when the current count equals the preset value.**

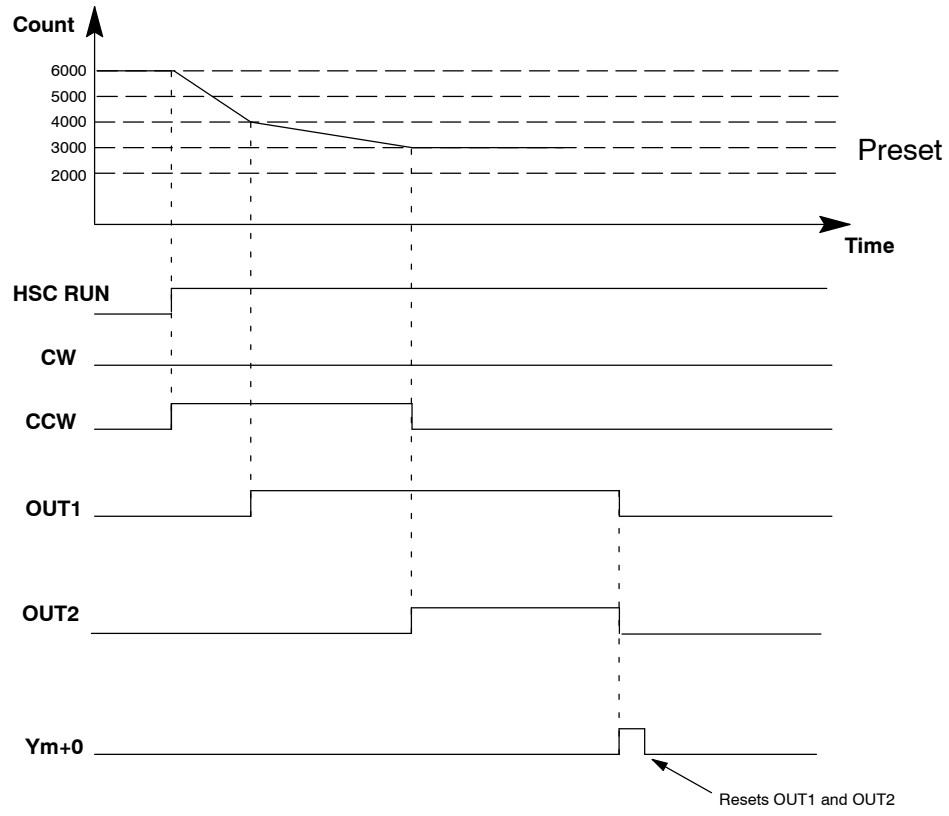
In both Figure A and Figure B on next page, you see that OUT2 turns ON (and the direction output turns OFF) at the very moment that the current count equals 3000 (the stored preset value).

**Monitoring Speed Output Status** Both OUT1 and OUT2 have flags that can be monitored to check the status. You can use these flags in ladder logic to trigger events.

Flag	Function
Xn+7	Will turn ON when OUT1 is ON and turn OFF when OUT1 is OFF.
Xn+5	Will turn ON when OUT2 is ON and turn OFF when OUT2 is OFF.

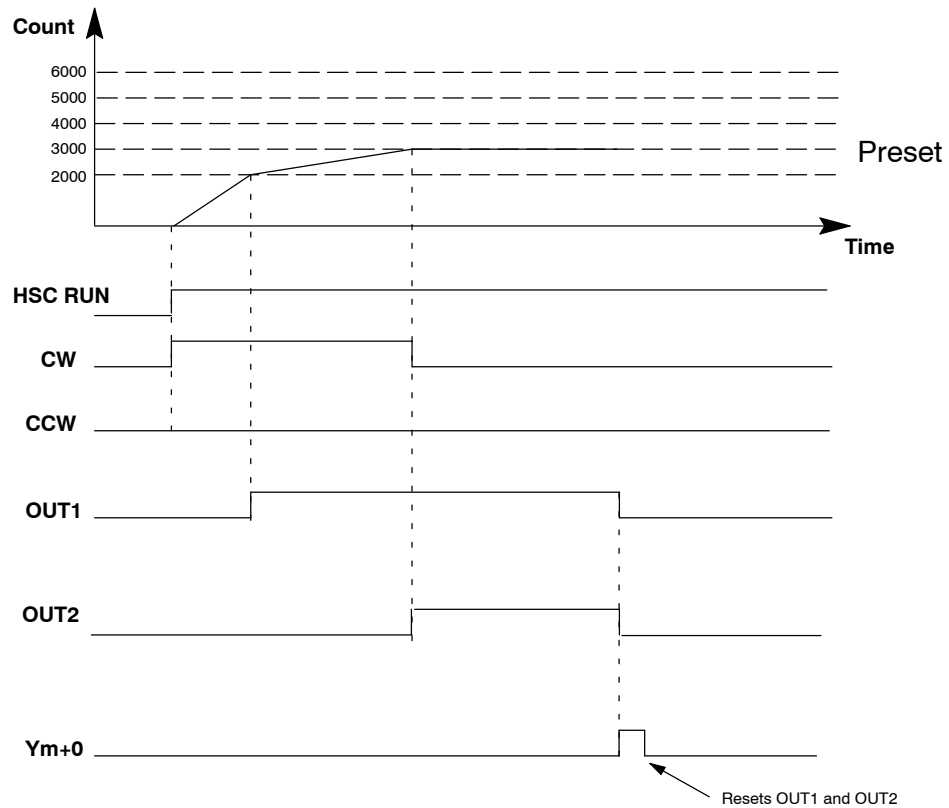
**Figure A:**

**Example of Outputs when Decel=1000, Preset=3000, and Current Count > Preset**



**Figure B:**

**Example of Outputs when Decel=1000, Preset=3000, and Current Count < Preset**



# Special Features

---

In This Chapter. . . .

- Planning a Home Search Application
- Using Sampling

## Planning a Home Search Application

### Requirements for Home Search

We discussed briefly in Chapter 1 some of the basics concerning the automatic home search feature of the HSC. Here we will elaborate further.

The built-in algorithm of the HSC assumes that you have specified certain types of field devices for your home search system. It also assumes they have been physically configured in a certain manner. Let's take a closer look:

- LS2 must be a negative logic switch (i.e. turn OFF when work is near).
- The status of LS1 must always be ON during Home Search. An actual physical device is not required as long as the input point (LS1) is always HIGH during Home Search. If you attach a switch to LS1, make sure it is the negative logic type also.
- LD must be a positive logic switch (i.e. turn ON when work is near).
- In relationship to the motor and its shaft rotation, the LS1 switch must be located so that when the motor turns clockwise, the moving work apparatus travels in the direction of the LS1 switch. Likewise, when the motor moves counter-clockwise, the work apparatus moves toward the LS2 switch.
- LS1 and LS2 define the two ends of the work area. The Home Search algorithm uses the status of these two input points while locating home position.
- Home position is defined to be **near** LD on the LS2 side.

### Example of Home Search Application

In the diagram on the opposite page, assume that the drill head has just performed its last task on the work piece and has been raised. Your ladder logic has invoked home search (OFF to ON transition of Ym+15). The status of all your outputs are stored in memory, because when Home Search is complete the outputs will be returned to their original state. We will assume that the pitch of the lead screw is such that CCW movement is to the left toward LS2:

Step1 - CCW turns ON and moves the drill head toward LS2 at full speed.

Step2 - When LS2 detects the drill head and turns OFF, CW is turned ON (CCW OFF), and movement is back toward LD at full speed.

Step3 - When LD turns ON, OUT1 (deceleration) turns ON.

Step4 - The drill head moves slowly past LD until LD turns OFF.

Step5 - When LD has turned OFF, CCW is once again turned ON, and with OUT1 still ON, it moves slowly toward LD again. LD turns ON. Drill head continues moving until LD turns OFF.

Step6 - If you were in the UP/DOWN mode of counting, the drill head would stop as soon as LD turns OFF. Home is defined as that position. This would be the end of Home Search for the UP/DOWN mode. Read the note below for use of Ym+27 (enabling OUT2, the brake) to assure that momentum does not cause overshoot.

Step7 - If you were in the Quadrature mode of counting, the drill head would not stop when LD turns OFF. It would, continue until INZ turns ON. It would stop at this point and the Home Search would be finished. Read the note below for use of Ym+27 (enabling OUT2, the brake) to assure that momentum does not cause overshoot.

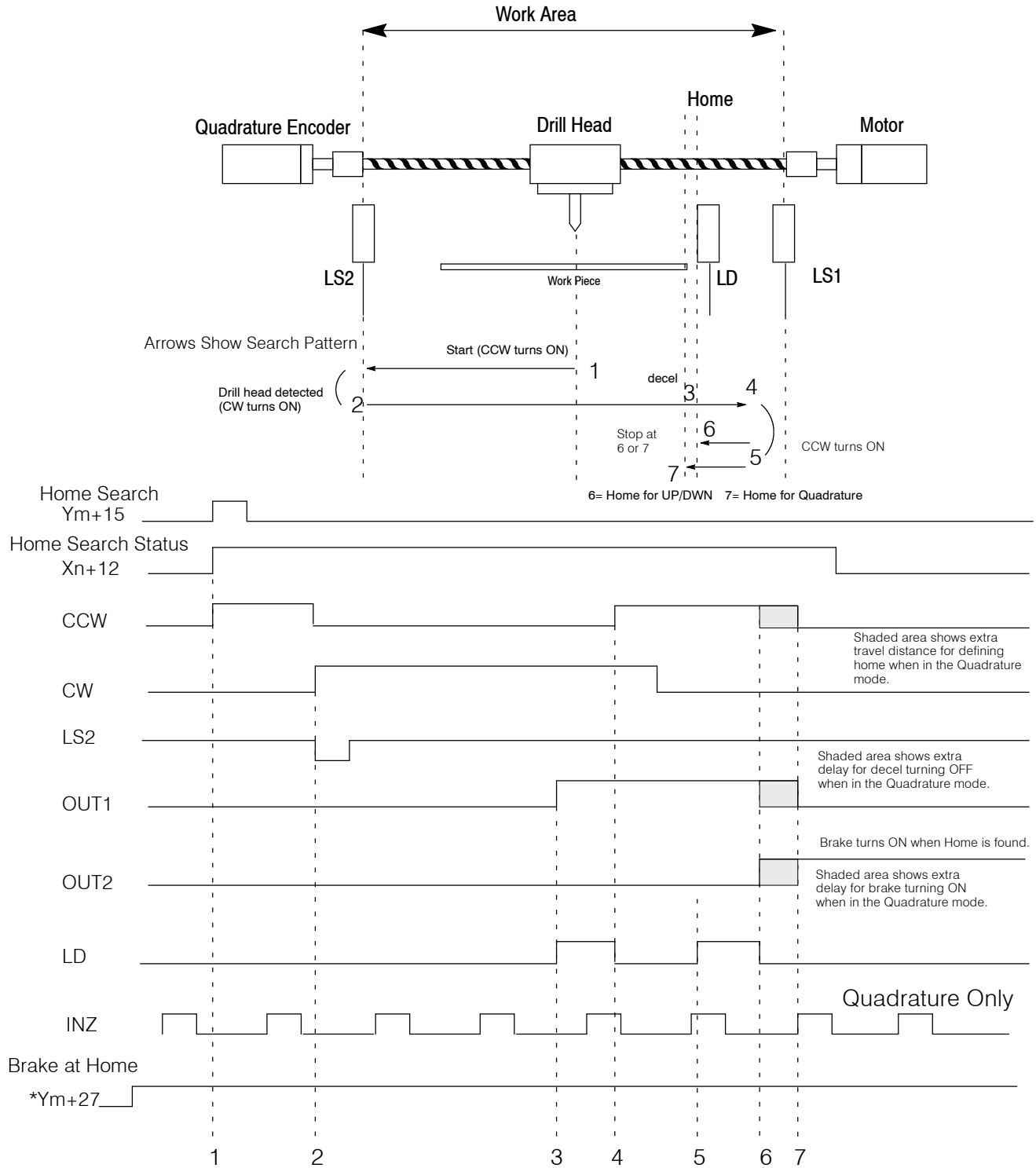
---

**NOTE: If you want the brake (OUT2) to turn ON when Home is found, then you must turn ON Ym+27 before invoking Home Search. This feature is only supported in HSC modules with date codes 9502 (Feb. '95) or later. You will find this production date on the bar coded label located on the side of the module.**

---



Example Home Search Setup



**\*Note:** Ym+27 only affects the status of OUT2. The rest of the Home Search algorithm operates regardless of the status of Ym+27.

## Using Sampling

### Why Use Sampling?

The D4-HSC allows you to determine the difference in the current count over a specified time period in the range of 3 to 29997ms. This is *sampling*. By knowing how many pulses have been counted in a specified period of time, sampling can be useful for determining flow rate, the speed of a rotating shaft, etc. Sampling uses two shared memory locations:

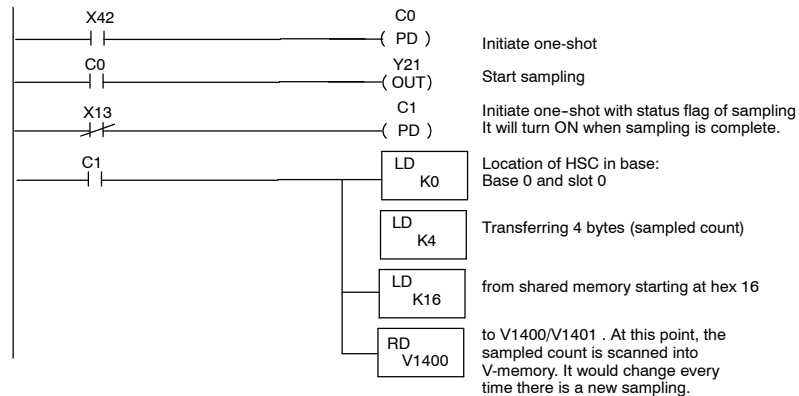
Step1 - **Timebase** (length of measuring period)-Address hex 14 through 15

Step2 - **Sampled Count** (number of pulses counted)-Address hex 16 through 19

The timebase is entered into shared memory (hex 14-15) by your ladder logic. You don't enter the actual time, such as 3-29997ms. Instead, the value actually entered is a number in the range 1 through 9999 (BCD). The HSC multiplies this number by 3ms in order to convert it to the actual timebase. That is, if your program writes a 4 there, then the sampling timebase is  $4 \times 3\text{ms} = 12\text{ms}$ . When executed, the counter will count for 12 ms and then the HSC will store the count for that time period into hex 16 through 19.

### Enabling and Monitoring the Sampling

The act of sampling with the HSC is invoked by using  $Ym+21$ .  $Xn+13$  will turn ON when sampling is initiated and remain ON until it is complete. The short example logic shown below illustrates how to enable the sampling feature and then use the status flag to initiate a reading of the sampled count into V-memory.



You should probably take note that  $Ym+21$  does not have to stay ON. If  $Ym+21$  is turned OFF, the sampling process would continue until completion anyway.

**NOTE: When using sampling, be aware that any other function of the HSC that alters or inhibits the current count (i.e reset, offset, count inhibit) will affect the sampling.**

**How to Calculate the Timebase**

As mentioned, if you want to compute the value of the number to store in shared memory for your timebase, then you will have to divide your actual timebase value by 3ms. For example, suppose you wanted a time base of 300ms. You would store 300ms/3ms in shared memory hex 14 thru 15:

$$300\text{ms} \quad 3\text{ms} = 100$$

**What Happens If You Want to Enter a Value with Decimal Points?**

Not every timebase computation is going to come out with integer values. For example, if you want a time base of 1000ms, here is the math:

$$1000\text{ms} \quad 3\text{ms} = 333.33$$

A normal question at this juncture is “How do I enter a value with ladder logic that has a decimal point?”. The answer is: You can’t. You will have to enter the nearest integer, which is 333. This means your actual time base is 999ms.

In most cases, this will not cause you any problems. You could, for example, in most cases accomplish whatever you want with the sampling even though it is missing 1ms of sampling time. However, with very high speed counting and a need to use this feature to determine frequency, you could experience difficulty. You should be aware of this when planning your application.

**Summary of Input and Output Relays for Sampling**

The chart below summarizes the X and Y output assignments for sampling.

X or Y No.	Function
Xn+13	Reports the status of sampling process. ON during the process. OFF when sampling is finished or not taking place.
Ym+21	Transition from OFF to ON enables the sampling process. Turning this relay OFF will not stop the sampling process. It continues to completion.

# Program Applications

---

In This Chapter. . . .

- A Quick Checkout of the Module
  - Application No. 1: Drilling Operation
  - Application No. 2: Cut-to-Length Operation
-

## A Quick Checkout of the Module

### What It Does

The configuration shown below and the RLL on the adjacent page are not intended for actual application of the D4-HSC, but rather is shown as a means to quickly checkout the main functions of the module — namely its counting and output capability. In this configuration, we are using an internal special purpose relay (SP7) of the DL-405 to generate a low frequency pulse signal for the inputs of the D4-HSC. **DirectSOFT** programming and its Watch Window capability can be used to monitor the counting and preset versus current count relationship. The color coding of **DirectSOFT**'s ladder logic can show you the status of your various outputs. You can also visually watch the LED's of the module to witness all of these functions.

### How It Works

It is assumed the HSC is in slot 0 and you have a sourcing DC output module in Slot 1. There is also an input module with a switch attached or an input simulator in Slot 2. Here is how the program works:

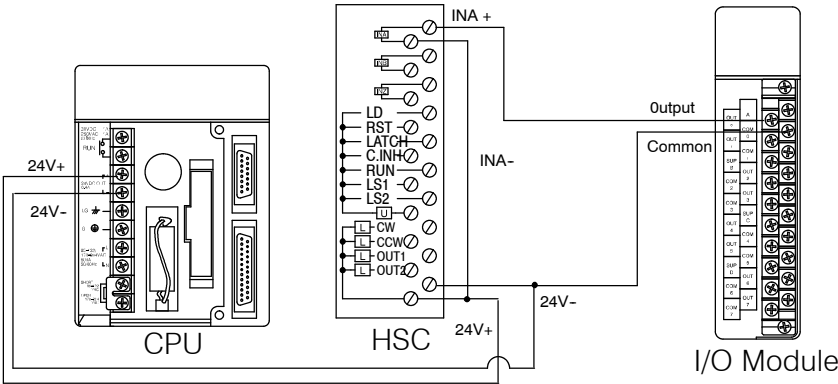
1. On power up, SP0 comes ON, loading your preset and deceleration values into the shared memory of the HSC.
2. When you turn on X20 (start switch) via the input module or simulator, Y12 sets the counter to zero.
3. When the counter is set to zero, X2 turns ON and this causes Y3 (HSC RUN mode) to be activated.
4. At this point, CW turns on because current count is less than preset. The HSC is constantly comparing the value of your current count with your preset.
5. When the count value equals your deceleration value, OUT1 will turn ON.
6. When current count and preset are equal, CW turns OFF and OUT2 (brake) turns ON.
7. At this point, X1 turns ON and Y40 stops pulsing. Y3 is also reset causing an exit of HSC RUN. You will notice that OUT1 and OUT2 stay ON even though we exit the HSC RUN mode. If you wish these to turn OFF before exiting, you will have to turn ON Y0 first, and then exit. The outputs will also reset when HSC RUN is invoked again.

### Things You Need for the Example

This is a list of items that you will need in order to set up the Quick Checkout for the module:

- DL405 series CPU/Power Supply unit (DL430 or DL440)
- Mounting base for the above (with at least 3 empty slots)
- Any current sourcing DC output module
- D4-HSC module (mounted in Slot 0 of the base)
- Either an input simulator or an input module with switch attached

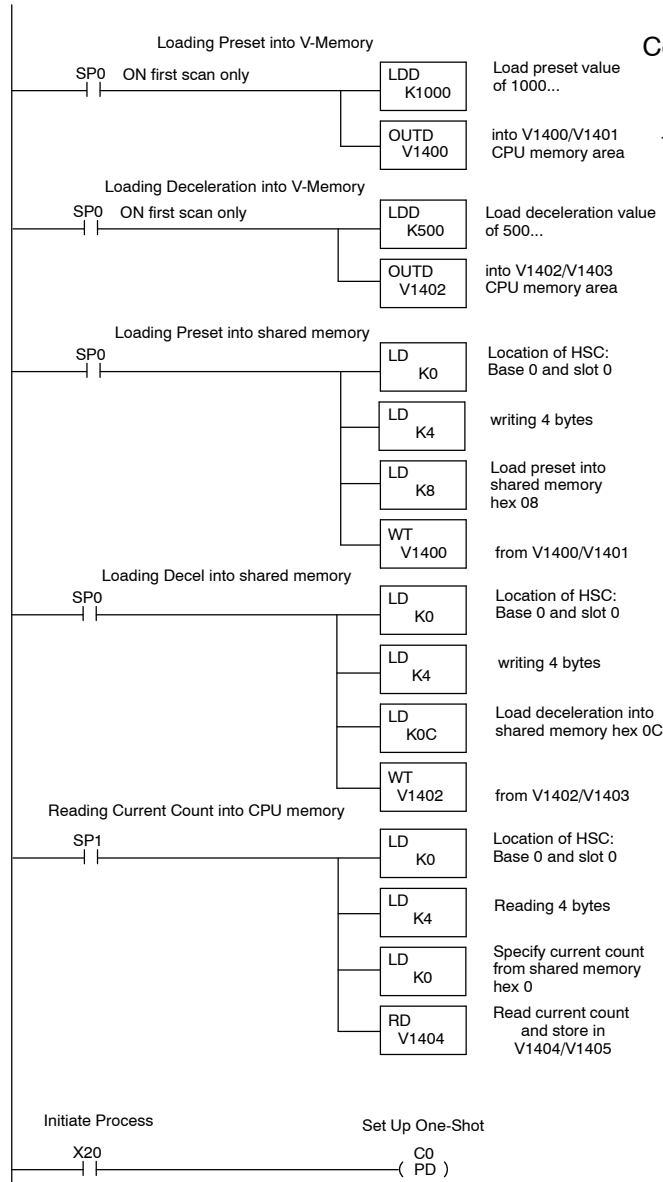
**Wiring Diagram for Example**



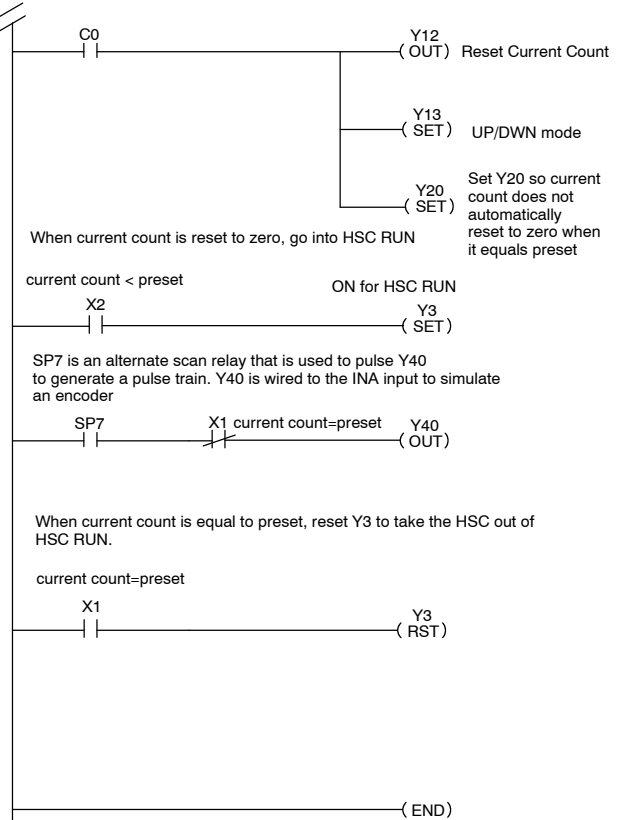
### RLL Program for Quick Checkout of the Module (Assumes the HSC is in Slot 0 of the Base 0.)

D4-HSC  
Program Applications

DirectSOFT Display



Continued from adjacent column



Continued at the top of next column

# Application No. 1: Drilling Operation

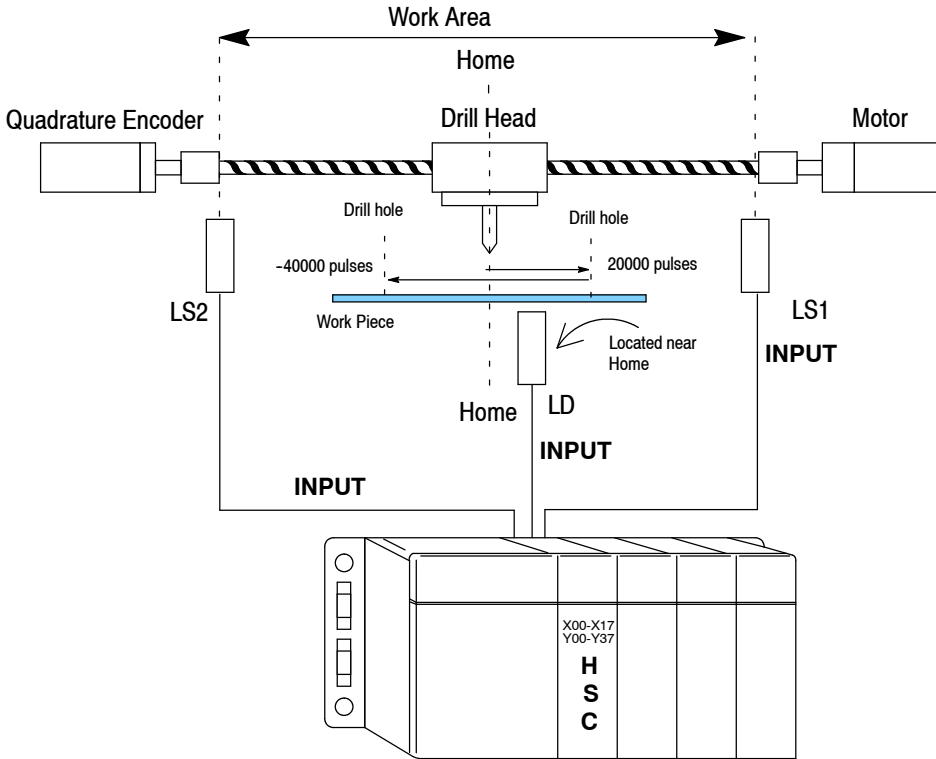
This application is designed to show you how multiple presets, negative presets and home search can be used in a program that drills holes in a work piece. The drill head is attached to a lead screw driven by a motor that is used with a quadrature encoder.

To simplify matters, we have placed the HSC in slot 0 of base 0. This means that X inputs and Y outputs of the HSC are automatically assigned X00 through X17 and Y00 through Y37. Refer to the I/O Configuration Table in Appendix B to find out what specific functions are assigned to each of these data points.

We are going to be drilling holes at two target areas (presets) 20000 pulses, and -40000 pulses. We will be using the Home Search feature to control the process. Since the position of the first hole drilled is keyed off of "home", we will always execute Home Search when the second hole has been drilled. This helps eliminate inaccuracies in positioning caused by mechanical inaccuracies or false pulses being received.

The sequence of events is as follows:

1. The drill is idle and in the raised position.
2. Home Search is enabled to make sure drill is positioned at Home.
3. X22 comes ON to indicate that the work piece is in position.
4. Turn ON HSC RUN and move in positive direction.
5. At 19000 pulses slow down the drill head, then stop at 20000. Lower the drill head and drill a hole.
6. Raise the drill head and move in a negative direction.
7. When the drill head is at -39000, slow down, then stop at -40000.
8. Lower the drill head and drill the second hole.
9. Raise the drill head, reset the outputs, and invoke Home Search again.





Ladder Logic for Drilling Operation

**WARNING: This application is included as an example and should not be used for real-world applications. It does not contain any safety-related instructions, which could possibly be needed.**

D4-HSC Program Applications

On the first CPU scan, write the value for the first preset into V1400 and V1401. Also write the value for deceleration into V1402 and V1403 and write the value for the second preset into V1404 and V1405. Reset Y20 to reset current count to zero upon reaching preset. Reset Y13 to select the quadrature mode, and set Y17 to select the 4x counting resolution.

On the first CPU scan or after the second drilling operation is complete, clear the current count in shared memory by turning ON Y12.

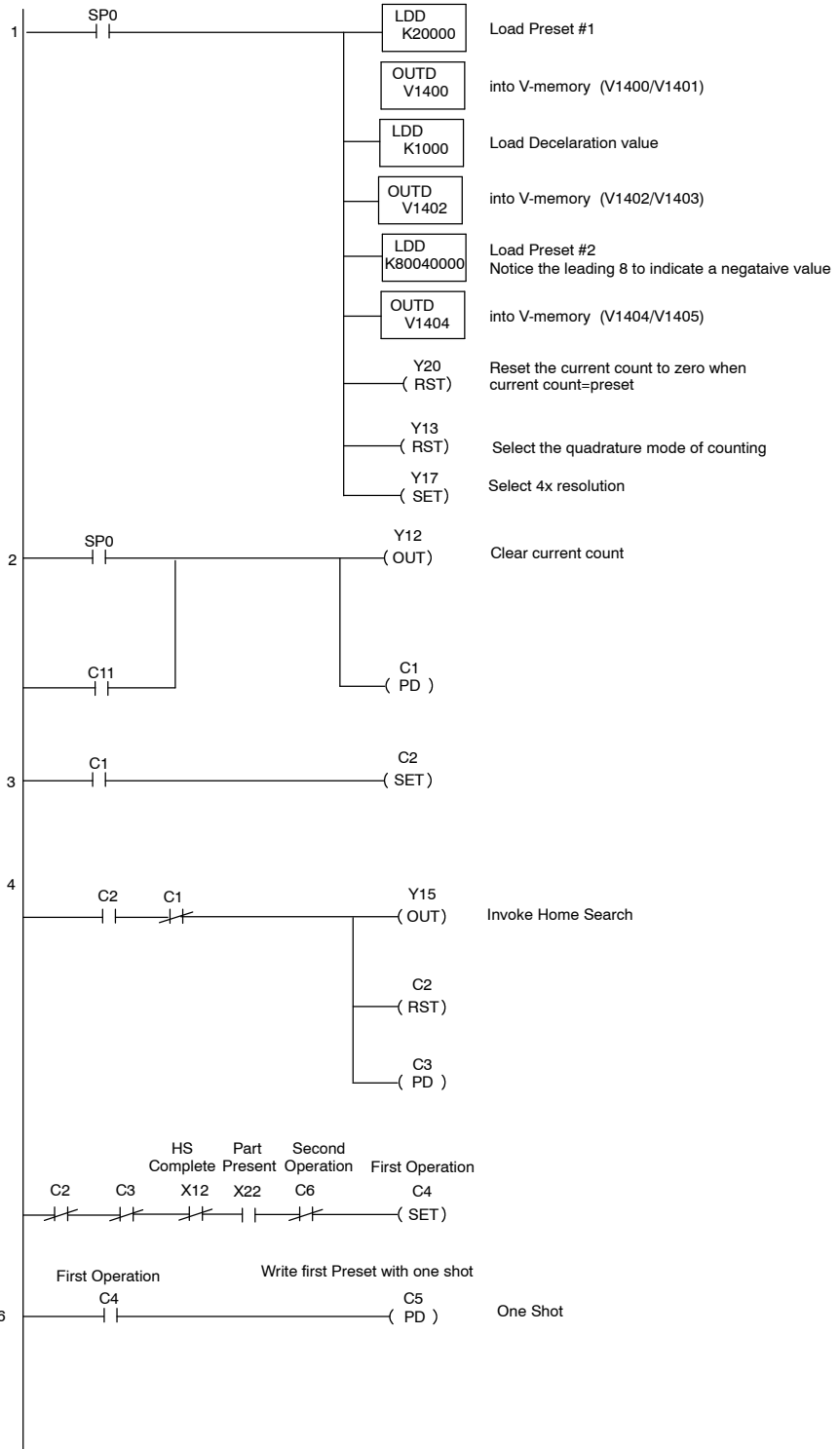
One shot pulses when second drilling operation is complete so that process can be enabled again.

C1 and C2 are used to wait one scan after clearing the current count before invoking Home Search.

Invoke Home Search

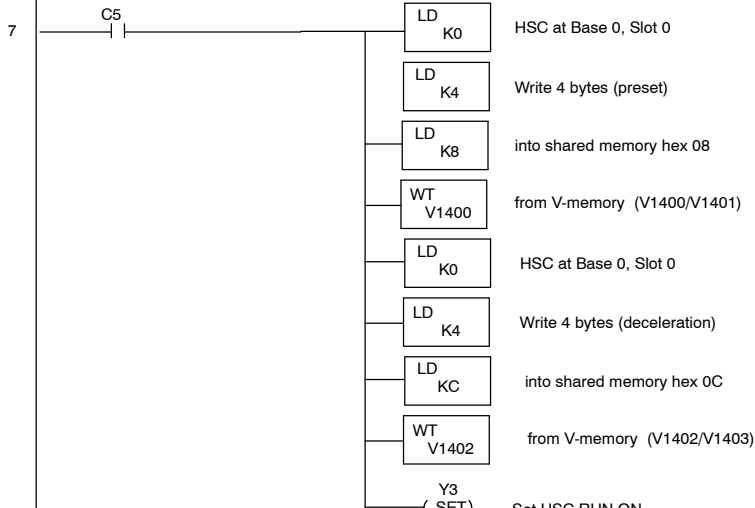
After Home Search is requested, we monitor X12 to determine when it is complete. When X12 turns OFF, we set internal relay C4.

When the first drilling operation is requested, pulse one shot C5.



Continuation of Application No. 1

Write the preset and deceleration value for the destination of the first hole to shared memory and invoke HSC RUN.

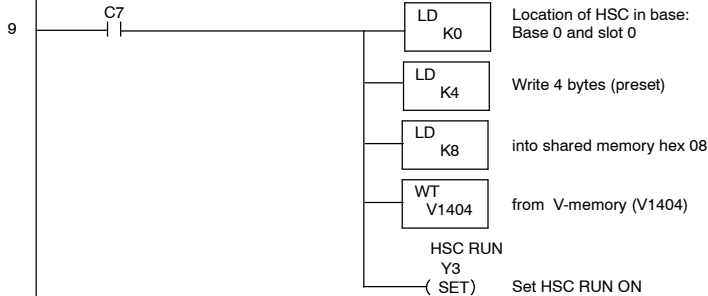


When the second drilling operation is requested, pulse one shot C7.

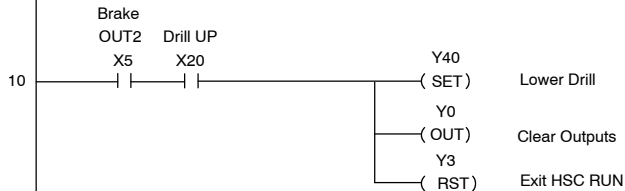


Write the preset value for the destination of the second hole to shared memory and invoke HSC RUN.

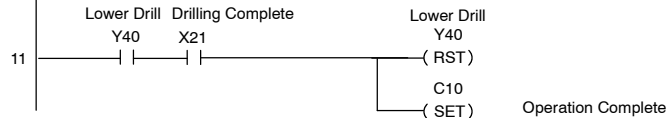
NOTE: We do not have to write a new deceleration value because we want to start deceleration at the same number of pulses from our preset value as in the first operation.



When the destination is reached, lower the drill by setting Y40. Clear the HSC's outputs by turning ON Y0, and then exit HSC RUN by resetting Y3.

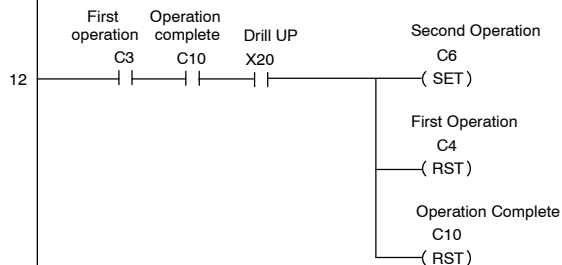


When the drill reaches the lower limit, X21 comes ON and resets Y40. This enables the drill to raise and sets Operation Complete bit C10.

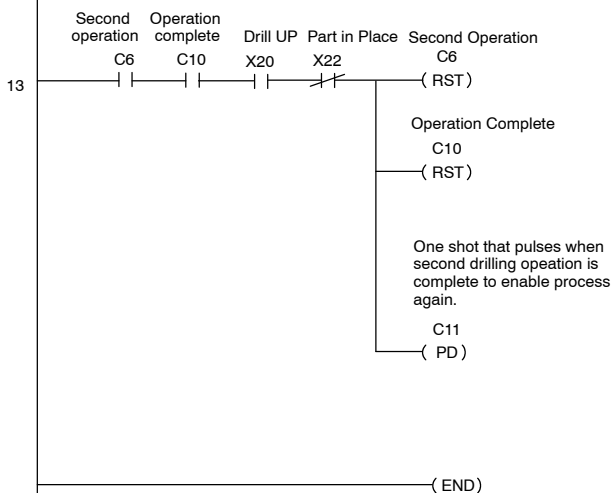


Continuation of Application No. 1

When the first drilling is complete and the drill is in the raised position, enable the logic that loads the values into shared memory for the next operation by setting C6 and resetting C4, and reset the operation complete bit, C10.



With the second drilling operation is complete, the drill is in the raised position, and the part is removed, reset C6 and C10 and enable the logic that invokes Home Search by pulsing C11.



# Application No. 2: Cut-to-Length Operation

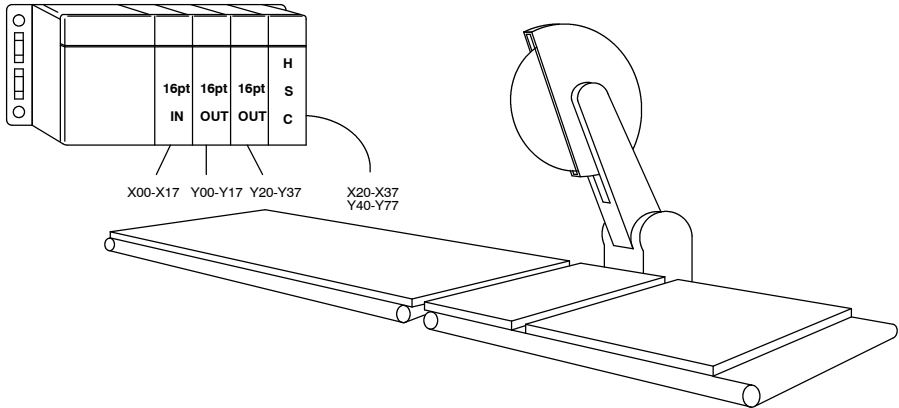
In this application, we have the HSC located in slot 3 of base 0. The I/O modules to the left of the HSC, consume data points X00 through X17 and Y00 through Y37. This means the data points assigned automatically to the HSC are X20 through X37 and Y40 through Y77. Refer to the I/O Configuration Table in the Appendix B of this manual to find out what functions of the HSC are assigned to these points.

We are cutting boards to length in this application. Our preset target in this case is 9,000 pulses, and the deceleration value is 500 pulses. We are using X2, a photoeye (not shown), to detect when a board has reached the work area. If a board is not present, the RLL program will advance the conveyor by turning ON Ym+6. When a board is detected, control of the conveyor will be transferred to the HSC which will automatically control the conveyor. This will immediately transfer control of the conveyor over to the HSC. Here's how it works:

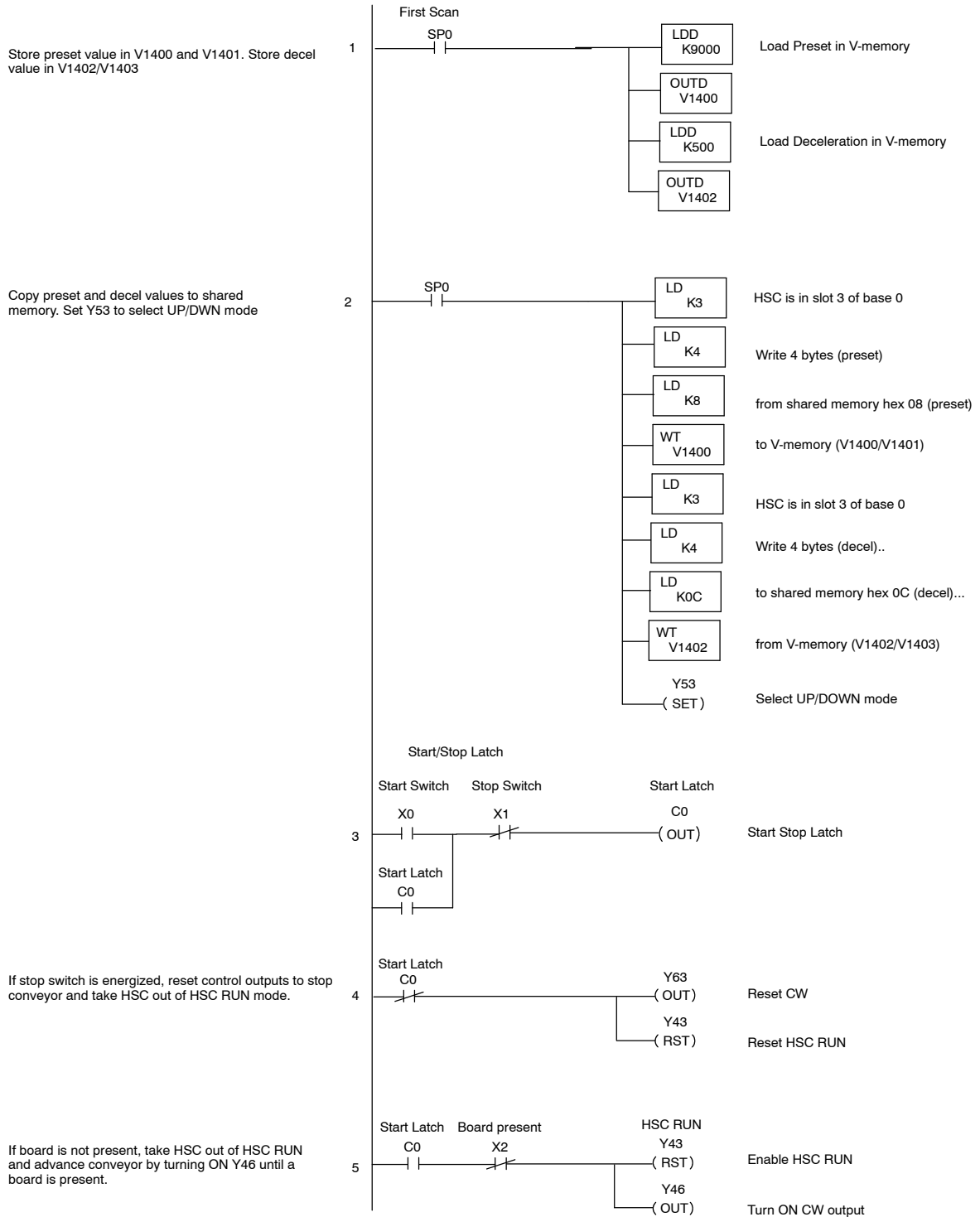
1. When the photo beam is unbroken (X2 ON), CW output is turned ON to advance board to the work area.
2. As soon as the photoeye detects a board, we clear the current count by pulsing Y52 (reset with Ym+12) and invoke HSC RUN by setting Y43 (Ym+3).
3. When current count reaches 8500 pulses, the deceleration (OUT1) is enabled; and when the current count reaches 9000 (equals preset) the brake (OUT2) is enabled. This stops the conveyor.
4. When OUT2 (brake) turns ON, X25 will turn ON (echoed value of OUT2). We use this input to clamp the board and enable the saw.
5. When the cut is complete, X3 will turn ON allowing the saw to retract and unclamp the board. X3 also resets the external outputs of the HSC and exits HSC RUN.
6. When the saw returns to its home position and a part is still present, the process will start over. If the photoeye turns OFF before we reach the target position, we will exit HSC RUN and once again advance the conveyor using Y46.

The boards in this application may be a variety of lengths. 9,000 pulses represent the length of board we want. If there are, say 3000 pulses per foot of board, our desired board length is 3 feet after the cutting. If a board trips the photoeye that is less than 3 feet in length, it is rejected and HSC RUN is not enabled. CW turns ON and a new board is advanced.

Notice also in the program that OUT1 and OUT2 have been wired to inputs of a drive that have been programmed to slow down the conveyor when a signal is received from OUT1 and brake when received from OUT2.



**WARNING: This application is included as an example and should not be used for real-world applications. It does not contain any safety-related instructions, which could possibly be needed.**



### Continuation of Application No. 2

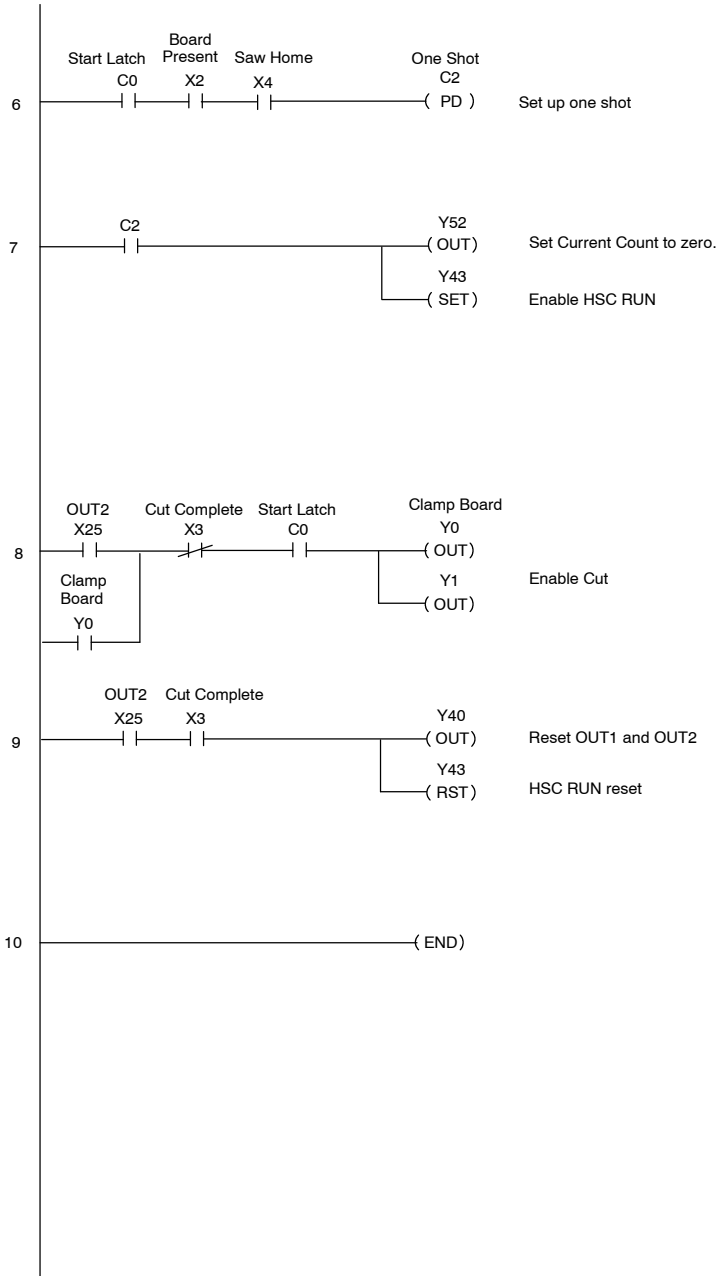
If a board is present, pulse one shot C2.

C2 will pulse Y52, clearing the current count and also setting Y43. This places the HSC in HSC RUN, allowing the HSC to automatically control the conveyor.

External output OUT1 is connected to the brake input of the drive. When the current count reaches the deceleration value, it will turn ON OUT1 which will slow the conveyor to half speed. When preset is reached OUT2 turns ON and stops the conveyor. X25 turns ON when OUT2 comes on turning ON Y0 and Y1. Y0 and Y1 clamp and cut the board. X3 turns ON when the cut is complete which releases the clamp and allows the saw to retract.

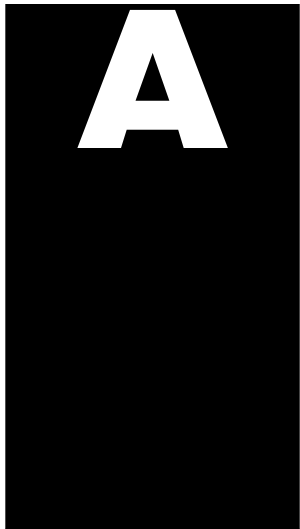
When the cut is complete, turn ON Y40 to reset the external outputs and reset Y43 to take the HSC out of HSC RUN mode.

**NOTE:** If a board is still present after the cut, Rungs 6 and 7 above will enable HSC RUN again to move the conveyor and make the next cut.



# Introduction to Motor Drives and Encoders

---



In This Chapter. . . .

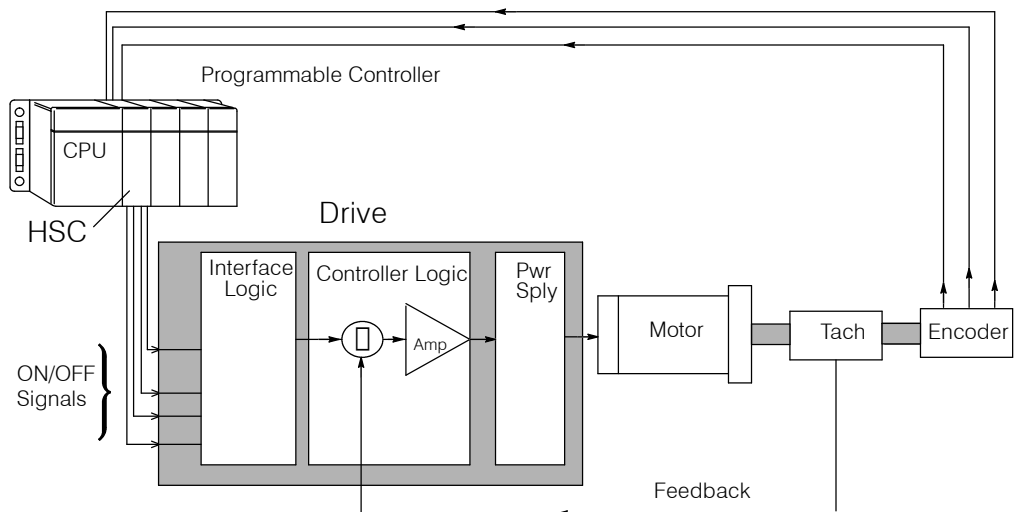
- What is a Drive and How Does it Connect to the HSC
  - What is an Encoder and How Does it Connect to the HSC
-

## What is a Drive and How Does It Connect the HSC?

The word drive can have many meanings. Here we need to limit our definition to any intelligent electronic equipment that provides adjustable speed control for a motor. The motor can be AC or DC. From the standpoint of the HSC, it doesn't matter.

Drives provide many functions for the motor. Most often the drives have some form of rectification that takes place in addition to providing intelligent control for the motor. When working with the HSC you do not have to be concerned about the drive's internal power conversion because the power requirements of the drive and the PLC are handled separately. The HSC only connects to the drive's logic function.

The following simplified diagram shows the basic sections of a common type of electronic motor drive.



If you're new to motion applications, you may think that when an output signal is received by the drive from the HSC, the signal has some inherent property that affects the speed or direction of the motor shaft. In reality, all four outputs of the HSC (CW, CCW, OUT1, and OUT2) look the same electrically. The signals are simply discrete ON/OFF signals. What makes them different is where they are connected to the drive. It is the intelligence of the drive (and not the HSC) that determines how the signals affect the motor operation. The PLC merely takes advantage of the I/O functionality that has been pre-programmed inside the drive itself.

Some drives (or intelligent motor controllers) will have connecting terminals marked CW, CCW, or maybe even deceleration or brake. In these cases the terminals have been internally set up by the drive manufacturer to properly use the signals received. In most cases however there are no such markings. You will program the I/O points on the drive to perform whatever function you wish from those available on your drive. It could be to change motor direction, change speed, or any number of functions. You then, have the responsibility of matching up the drive I/O with the proper output signals from the HSC.



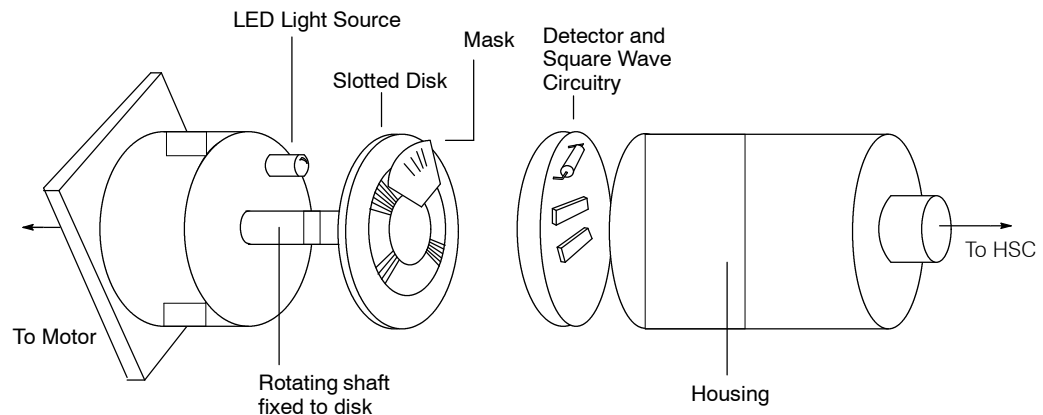
## What is an Encoder and How Does It Connect to the HSC?

The word encoder can be a confusing term to those who have never used them. There are many types of encoders and the word itself can have as broad or narrow of a definition as a particular application might dictate. We will limit our discussion here to **incremental encoders**. Even narrowing the scope down to this family of encoders can present quite a large array of products offered by various manufacturers. We will not attempt to describe every type, and will only cover the subject in the most common and general sense.

From the standpoint of the HSC, it is only looking for a square pulse train (or trains in the case of a quadrature encoder). It doesn't really care how they were produced or the physical appearance of what has sent the signal(s). As long as the incoming signals have the proper width requirements and are not being received faster than 100 kHz, the HSC can count them.

### How the Incremental Encoder Works

The most common encoder is the rotary incremental optical encoder. Inside its housing, are usually five basic components: light source (usually an LED), a slotted disk, a mask, a photo detector, and square wave circuitry. A quadrature encoder has a second light source and sensor (Phase B) located over the disk track. A single signal (non-quadrature) encoder is shown in the diagram below.



The quantity of slots on the disk is equivalent to the number of pulses per turn. Most quadrature encoders also provide a single isolated slot on the disk called the Z marker. It is aligned with its own separate LED. The pulse from this channel provides a reference once per revolution. The automatic home search function, built into the HSC, makes use of the Z-marker when you connect a quadrature encoder to the HSC with the Z-marker attached to **INZ**.

### What is the Z-Marker?

The Z-marker helps to more closely define a home base to which the working apparatus can be accurately returned. It's kind of like "top dead center" on a crankshaft for purposes of timing your spark with the correct piston position.

The D4-HSC does not use the Z-signal for error detection or correction of any lost pulses or extra pulses that are associated with a particular revolution of the encoder. The HSC merely uses the Z-signal to make sure the slotted disk of the encoder is always in the same position at the start of each work cycle.

# Summary of Tables and Charts Presented in the Text

---

In This Chapter. . . .

- Specifications for D4-HSC
  - X and Y Assignment Table
  - Shared Memory Table
  - Table for Determining Count Direction
  - Counting Resolution Table (Quadrature Only)
-

## Specifications

Specification (General)	Rating or Requirement
405 CPU Firmware Requirements	Any PLC <i>Direct</i> CPU or other vendor's 405 CPU (Version 1.6 or later)
Slot for Installation	Can be installed in any CPU or expansion base. Cannot be installed in a remote base.
Maximum No. HSC's per CPU	8
No. of I/O points required	Consumes 16 X-inputs and 32 Y-outputs
Intelligence Source	Has its own microprocessor (asynchronous to the DL405 CPU)
Internal Power Consumption	300 mA maximum at 5VDC
Field Wiring Connector	Removable terminal type
Count Signal Level	4.75VDC to 30VDC less than 10mA
Maximum Count Speed	100 kHz (50% duty cycle)
Minimum Input Pulse Width	5 $\mu$ s (either state)
Count Input Signal Types	Standard (UP/DOWN) or quadrature (phase differential)
Count Range	-8,388,608 to +8,388,607
Count Direction	UP or DOWN (software selectable or hardwired)
CPU Scan Time Increase per HSC in base	4.2 to 5.5 ms

Specification (INA, INB, INZ)	Rating or Requirement
Input Voltage Range	4.75VDC to 30VDC
Maximum Output Current	10 mA
ON Voltage	4.75VDC
ON Current	5mA
OFF Voltage	2.0VDC
OFF Current	1.6mA
OFF to ON Delay	1.2 $\mu$ s at 5VDC 0.8 $\mu$ s at 12VDC 0.5 $\mu$ s at 24VDC
ON to OFF Delay	1.0 $\mu$ s at 5VDC 1.2 $\mu$ s at 12VDC 2.5 $\mu$ s at 24VDC

Specification (CW,CCW,OUT1,OUT2)	Rating or Requirement
Output Power Source	External 10.2VDC-26.4VDC, 1A
Output Type	Open Collector
Maximum Output Current	100 mA per point
Output ON Voltage Drop	1.5VDC
Output OFF Leakage Current	100mA
Output OFF to ON Delay	22.5µs at 12VDC 21µs at 24VDC
Output ON to OFF Delay	210µs at 12VDC 270µs at 24VDC
Built-In Protection	Shut off when output driver IC=175°C (Recovers at 150°C) Shut off when short ( 500mA) is detected (Recovers when short is removed)

Specification (LD, LATCH, RTS, CINH, RUN, LS1, LS2)	Rating or Requirement
Input Voltage Range	10.2VDC-26.4VDC
Maximum Input Current	10mA
ON Voltage	10.2VDC
ON Current	5mA (LD and LATCH); ≥4.8mA (RTS,CINH,RUN,LS1 and LS2)
OFF Voltage	4.6VDC (LD and LATCH); ≤5.6VDC (RTS,CINH,RUN,LS1 and LS2)
OFF Current	1.6mA (LD and LATCH); ≤2mA (RTS,CINH,RUN,LS1 and LS2)
OFF to ON Delay	75µs at 12VDC (LD and LATCH) 82.5µs at 12VDC (RTS,CINH,RUN,LS1 and LS2) 30µs at 24VDC (LD and LATCH) 37.5µs at 24VDC (RTS,CINH,RUN,LS1 and LS2)
ON to OFF Delay	240µs at 12VDC (LD and LATCH) 105µs at 12VDC (RTS,CINH,RUN LS1 and LS2) 260µs at 24VDC (LD and LATCH) 105µs at 24VDC (RTS,CINH,RUN LS1 and LS2)

# X and Y Assignment Table

Tables and Charts

X No.	Function	Y No.	Function
Xn+0	ON if current count is greater than preset	Ym+0	ON to reset OUT1 and OUT2 when in HSC run
Xn+1	ON if current count is equal to preset	Ym+1	ON to reset overflow flag (Xn+3)
Xn+2	ON if current count is less than preset	Ym+2	Rising edge copies offset value into current count
Xn+3	Latched ON if overflow occurs ( reset with Ym+1 )	Ym+3	ON for HSC run
Xn+4	Status of CCW output	Ym+4	Used to control CCW when not in HSC RUN or Home Search
Xn+5	Status of OUT2 (brake) output	Ym+5	Used to control OUT2 when not in HSC RUN or Home Search
Xn+6	Status of CW output	Ym+6	Used to control CW when not in HSC RUN or Home Search
Xn+7	Status of OUT1 (deceleration) output	Ym+7	Used to control OUT1 when not in HSC RUN or Home Search
Xn+10	Status of Limit Switch 2	Ym+10	ON to temporarily suspend counting (inhibit counting)
Xn+11	Status of Limit Switch 1	Ym+11	Rising edge of this signal will latch the current count into memory
Xn+12	ON if doing a search for home position	Ym+12	If ON, it resets current count to zero
Xn+13	ON if a sampling is being conducted	Ym+13	OFF=Quadrature mode ON=UP/DOWN mode
Xn+14	NOT USED	Ym+14	Change state to change count direction
Xn+15	ON for loose or missing terminal block	Ym+15	Rising edge of this signal will invoke Home Search
Xn+16	ON if external power supply for outputs is missing or OFF	Ym+16	ON for x2 count operation (quadrature mode only/Ym+17 must be OFF)
Xn+17	ON if HSC fails its self test	Ym+17	ON for x4 count operation (quadrature mode only)
<p>*This manual was written for the latest version of the D4-HSC. If you have an HSC that was purchased from another vendor, it may not support these features.</p> <p>**Available on HSC modules with production codes 9502 (Feb.'95) or later.</p>		Ym+20	If OFF it will reset current count to zero when current count=preset. If ON will not reset unless count is at max or minimum.
		Ym+21	Rising edge of this signal will start the sampling feature
		Ym+22	Must be ON to enable external LD function
		Ym+23	ON to reset CW and CCW
		Ym +24	Not Used
		*Ym+25	ON to reset Home Search error.
		*Ym+26	ON to enable reset with INZ.
		**Ym+27	If turned ON before invoking Home Search, OUT2 (brake) will turn ON when Home is found.

# Shared Memory Table

## Shared Memory Map

## Data Flow Direction

(From DL405 CPU Perspective)

Data	Address (hex)		Address (octal)	V-Memory	Shared Memory
Current Count (4 bytes)	00 to 03	Range= -8388608 thru 8388607 Format: 8-digit BCD	00 to 03	Read ←	
Offset Value (4 bytes)	04 to 07	Range= -8388608 thru 8388607 Format: 8-digit BCD	04 to 07	Read ←	Write →
Preset Value (4 bytes)	08 to 0B	Range= -8388608 thru 8388607 Format: 8-digit BCD	10 to 13	Read ←	Write →
Deceleration (4 bytes)	0C to 0F	Range= -8388608 thru 8388607 Format: 8-digit BCD	14 to 17	Read ←	Write →
Latched Count (4 bytes)	10 to 13	Range= -8388608 thru 8388607 Format: 8-digit BCD	20 to 23	Read ←	
Timebase (2 bytes)	14 to 15	Range= 1 thru 9999 (Total time=above value x 3ms) Format: 4-digit BCD	24 to 25	Read ←	Write →
Sampled Count (4 bytes)	16 to 19	Range= 0 thru 8388608 Format: 8-digit BCD	26 to 31	Read ←	

## Table for Determining Count Direction

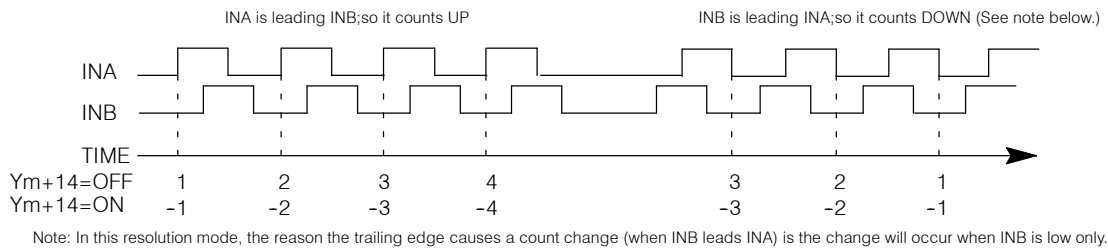
Mode Status	Direction	Criteria Used For Determining Direction
Ym+13=0	Ym+14=0	Counts UP if INA leads INB. Counts DOWN if INB leads INA (quadrature)
Ym+13=0	Ym+14=1	Counts UP if INB lead INA. Counts DOWN if INA leads INB (quadrature)
Ym+13=1	Ym+14=0	Counts UP with INA. Counts DOWN with INB (standard UP/DOWN)
Ym+13=1	Ym+14=1	Counts DOWN with INA. Counts UP with INB (standard UP/DOWN)

# Counting Resolution Table (Quadrature Only)

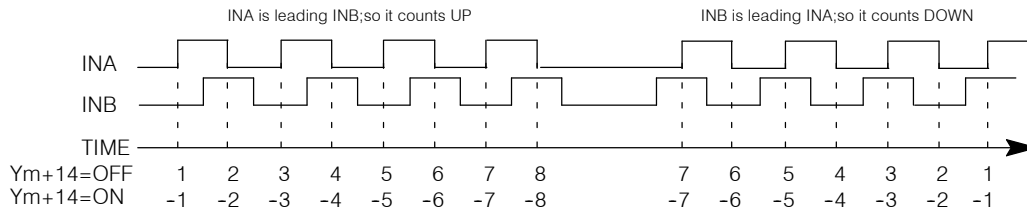
Ym+16	Ym+17	What Causes Count Change
OFF	OFF	1x: One edge of INA
ON	OFF	2x: Both edges of INA
OFF	ON	4x: All edges of INA and INB
ON	ON	4x: All edges of INA and INB

Y No.	Function
Ym+14	Change state to change count direction
Ym+15	ON will invoke home search
Ym+16	ON for x2 count operation (quadrature mode only/Ym+17 must be OFF)
Ym+17	ON for x4 count operation (quadrature mode only)

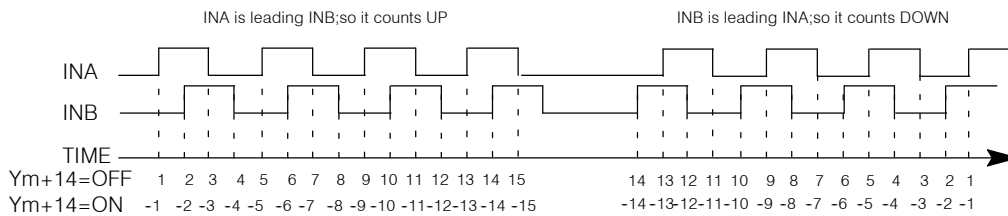
## Quadrature 1x Operation (One Edge: INA trigger)



## Quadrature 2x Operation (Two Edge: INA trigger)



## Quadrature 4x Operation (All Edges: INA and INB trigger)





# Index

---

## A

### Application, 7-2

- cut-to-length, 7-8
- sample ladder logic program, 7-9
- drilling operation, 7-4
- sample ladder logic program, 7-5

### Assignment, 3-3

- of data types, 3-3
- automatic, 3-3
- manual, 3-3
- table for inputs and outputs, 3-4, B-4

### Assistance

- technical, 1-3

## B

**Brake** *See* **Outputs**, speed, OUT2

## C

**CCW (counter-clockwise)**, 5-4

### Checkout, 7-2

- of HSC operation, 7-2
- ladder logic for, 7-3
- materials required, 7-2

**CINH**, 1-10, 4-11

### Counting, 4-3

- current count, 4-9
- starting and reset, 4-9
- direction of, 4-4
- sample RLL, 4-4
- Y outputs for, 4-4
- inhibiting, 4-11
- ladder logic for selecting mode, 4-3
- latching, 4-11

offset, 4-7

- external method, 4-7
- internal method, 4-7

overflow, 4-12

preset, 4-8

- checking status, 4-8
- loading into shared memory, 4-8

resolution, 4-5

1x, 4-5, 4-6

2x, 4-5, 4-6

4x, 4-5, 4-6

default setting, 4-6

settings available, 4-5

Y outputs for, 4-5

selection of mode, 4-3

types of

quadrature, 1-5

standard UP/DOWN, 1-4

**CW (clockwise)**, 5-4

## D

**Deceleration**, *See* **Outputs**, speed, OUT1

**Direction**. *See* **Counting**

### DirectSOFT, 4-2

edit mode, 4-2

watch window, 4-2

### Drives, A-2

basic components, A-2

connecting to HSC, A-2

---

**E****Encoders, A-3**

- connecting to HSC, A-3
- how they work, A-3
- types of, A-3
- Z-marker, A-3

**Equal Output, See Outputs, speed, OUT2**

**H****Home Search, 1-6, 6-2**

- example, 6-2
- requirements for, 6-2
- setup of components for, 6-3
- timing diagram, 6-3

**I****I/O Configuration, 3-3**

- automatic, 3-3
- manual, 3-3

**INA, 1-4,1-5,1-8,4-3,4-4,4-5**

**INB, 1-4,1-5,1-8,4-3,4-4,4-5**

**Indicators See LED Assignments**

**Inhibiting, 4-11**

- overview, 4-11
- sample RLL, 4-11

**Installation, 2-2**

- avoiding electrical shock, 2-2

**Introduction, 1-2**

- 5 steps for using HSC, 1-12
- control outputs, 1-5
- home search, 1-6
- manual organization, 1-3
- overview of inputs and outputs, 1-10
  - internal versus external activation, 1-10
- physical characteristics of HSC, 1-8
- purpose of manual, 1-2
- sampling, 1-6
- shared memory, 1-7
- specifications of HSC, 1-8
- supplemental manuals, 1-2
- technical assistance, 1-2
- types of counting available, 1-4
- what is an HSC?, 1-4
- who needs an HSC?, 1-4
- X input assignments, 1-11
- Y output assignments, 1-11

**L****Latching, 4-11**

- how to trigger, 4-11
- overview, 4-11
- sample RLL, 4-11

**LD, 1-8,4-7,6-2,6-3**

**LED Assignments, 1-8**

**Limit Switch See Home Search**

**LS1 See Home Search**

**LS2 See Home Search**

**M**

**Memory See Shared Memory**

**O**

**Offset. See Counting**

**Outputs, 5-2**

- directional, 5-4
  - CCW(counter-clockwise), 5-4
  - CW(clockwise), 5-4
  - how HSC knows which to turn ON, 5-4
  - HSC RUN timing diagrams, 5-5
- overview, 5-2
- speed, 5-6
  - examples and diagrams, 5-7
  - how to initiate, 5-6
  - OUT1 (deceleration), 5-6
  - OUT2 (brake), 5-6
- two ways to control, 5-3
  - automatic, 5-3
  - manual, 5-3
- using internal relays, 5-2
- Y outputs for, 5-2

**Overflow, 4-12**

- flag status, 4-12
  - output relays, 4-12
  - overview, 4-12
  - tracking, 4-12
-

**P****Power Supply, 2-4**

external, 2-4

internal, 2-4

**Preset. See Counting****R****RAM See Shared Memory****Resolution. See Counting****RST, 1-8, 4-7****RUN, 1-8, 5-3****S****Sampling, 1-6, 6-4**

enabling, 6-4

monitoring, 6-4

summary of internal relays, 6-5

timebase for, 6-4, 6-5

using decimal points, 6-5

why use it?, 6-4

**Shared Memory, 1-7, 3-8**

contents and data flow, 3-6

how numbers are stored, 3-8

map of, 3-6

negative numbers, 3-9

sample RLL, 3-9

reading and writing to, 3-6

RLL for writing to, 3-7

**T****Technical Assistance, 1-2****Testing, 7-2**

operation of HSC, 7-2

test program, 7-2

**Timebase. See Sampling****W****Wiring, 2-3**

control input diagram, 2-6

control output diagram, 2-7

count input diagram, 2-5

guidelines for, 2-3

size of wire, 2-3

terminal block, 2-3