DESIGN
CONTROL
DRIVE
CONNECT

Soloist®

Power and Flexibility in a Drive-Based Controller

The power of the Soloist drive-based motion controller comes from both its powerful features and its flexibility. We develop our controller technology from the ground up, allowing us to deliver an automation product that is centered around precision. This approach gives our products a new level of performance, especially when compared to typical motor drive solutions. This often results in a new and rewarding experience for those used to working with motor drives from general automation suppliers.

By using the single-axis Soloist controller, enabled by the AeroBasic language, unique single-axis functionality is available. Features such as Aerotech's Position Synchronized Output (PSO) distance-based process triggering, gearing, advanced data collection, file access, analog and digital I/O control, and standard program flow and mathematical concepts are all integrated with powerful servo control. This can all be executed at up to 20 kHz.

Combined with powerful software tools, the Soloist controller excels in simple testing and R&D applications and is robust enough for production environments.

Powerful Single-Axis Trajectory Generation

At the heart of the controller is the ability to translate commands into the industry's best trajectory. Our controller includes the Aerotech motion engine — a robust technology that differentiates Aerotech control from other providers. This technology translates what you programmed for the single-axis trajectory into the drive's servo control loop.

Soloist servomotor drives can control DC brush servomotors, AC brushless servomotors, voice coil motors, and stepper motors.

Advanced trajectory features include:

- Point-to-point (PT, PVT)
- Target position and target velocity
- Linear and freerun
- Torque (current) mode operation
- Digital gearing to a global variable, to an auxiliary encoder input, or to an analog input
- Cubic spline curve-fitting

PRODUCT HIGHLIGHTS

Drive-based motion controller - part of the Soloist Platform

Powerful controller operates side-by-side on a motor drive, eliminating the need for additional hardware

Multiple single-axis controllers can be controlled by a single PC via Ethernet or USB

Delivers productivity and quality through combined motion and process control

Executes real-time application code developed in the Soloist Motion Composer Suite

Directly programmable through the .NET, C, C++, MATLAB, and LabVIEW libraries

Controller data is accessible to SCADA and MES control systems via libraries

Advanced motion trajectory features

- Velocity profiling
- Backlash compensation
- Error compensation
- Advanced queuing and deferred execution features

Single-Axis Automation

Single-axis drives should do more than simply move a motor. With Soloist drives, a complete controller is included, which allows sensors and feedback devices to come together with servomotor control applications.

A single-axis Soloist motor drive has a controller with direct access to up to 50 I/O points. Using the AeroBasic language's logic and control, algorithms are easily programmed based on the state of I/O.

Ethernet and RS-232 Support

Soloist drive-hardware includes both ethernet and RS-232 communication ports. These ports allow greater expandability of your automation solution — beyond the I/O available on the drives.

Modbus

The Ethernet port is configurable for Modbus communications. Class 1, Class 2, and part of Class 3 of the OpenModbus/TCP specifications are supported. While the Soloist always functions as Modbus/TCP slave, it can also act as a Modbus/TCP master when enabled, thus acting as a master and slave at the same time. Several blocks of Modbus registers are used to exchange data between the master and slave devices.

Aerotech includes standard procedures in its help files for integrating both WAGO and AutomationDirect devices over Modbus.

EtherNet/IP
Aerotech offers a Soloist EtherNet/IP™ option that communicates with Aerotech Soloist drives and the various ODVA Common Industrial Protocol (CIP™) objects that are used. The Soloist supports a Generic EtherNet/IP I/O interface by using Class 1 (Implicit Messaging) communications.

EPICS Compatible
Aerotech's Soloist controller is compatible with EPICS. Motor records are available from Argonne National Lab. Use your existing EPICS interface to control the most advanced motion controller in the industry.

Motion Composer Suite Engineering Software

Aerotech's Soloist Motion Composer Suite is a feature-rich engineering software package. Simply connect to the controller using the Motion Composer Suite and then use several user-friendly tools to set up the drive's control loops and controller-specific parameters.

The Soloist Motion Composer Suite includes the Motion Composer IDE — a feature-rich programming environment for developing real-time application code. The Motion Composer IDE allows you to quickly build, debug, load, and run real-time application code on the controller.

Programming Libraries Extend Functionality

When your goal is to deploy a custom user experience or access data from your drive, the Soloist controller enables many paths to success. Standard programming libraries for .NET, C, and C++ are included with the controller. If you are developing MATLAB or LabVIEW applications, support for each product is available.

No matter how you want to approach your precision motion control application, the Soloist controller helps you to succeed.

System Setup with the Soloist Drive-Based Motion Controller

The center of a powerful setup experience is the drive-based controller, which offers several options. As seen in the graphic below, these 'smart-drives' allow your design ideas to become your automation solutions. Our drive options enable the controller, fieldbus support, and servomotor control to be set up simultaneously with a single parameter file.

| DESIGN | CONTROL | DRIVE | CONNECT |
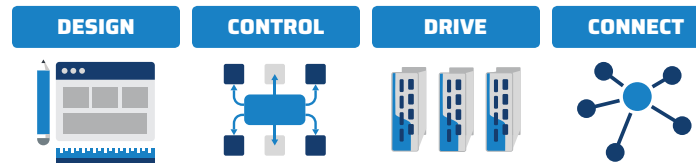
Programming Libraries

.NET, C, and C++ Libraries Plus REST Interface

Aerotech's standard controllers are flexible enough to handle almost any control application. From simple motion to coordinating and synchronizing the motion of multiple axes in your machine, Aerotech automation controllers provide the flexibility and power required for today's automation challenges, and it is all accessible through powerful programming libraries.

MATLAB® Library and LabVIEW® VIs

Users who program in MATLAB and LabVIEW can benefit from Aerotech's powerful automation tools. Integrate into your native environment with well-documented libraries.

.NET Library – Fully Functional for Developers

The Soloist .NET library is made accessible to our customers in the same structure and format that we use to develop our own applications. Therefore, developers should never worry that using our API will somehow reduce the capabilities of their own products or developments.

Some of the high-level functionality available in our APIs include:

Functionality	Description
Connect	Connect to one or several controllers
Configure	Configure settings, set/retrieve controller and drive parameters, tune motors and encoders, and load drive firmware
Execute Commands	Execute AeroBasic commands in .NET
Program Files	Compile AeroBasic programs, get build errors, load and run AeroBasic programs on the controller, and handle the controller tasks that run AeroBasic programs
Communicate	Set up and command drive and controller communication ports and protocols
Monitor Status	Get status and monitor information coming from the controller
Collect Data	Collect data on the controller in real time; exposes functionality used by the Digital Scope

AEROTECH

Accessing Data with the Soloist Drive-Based Motion Controller

Accessing data from your controller is simple. The Soloist controller allows fast and easy access to data through both the Soloist Motion Composer Suite and through the APIs.

Through the Motion Composer Suite

More than 30 axis status items are available for each controller axis, and 10 task state items are available for each controller task.
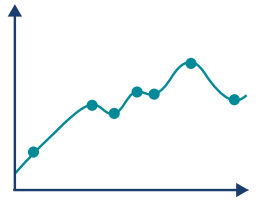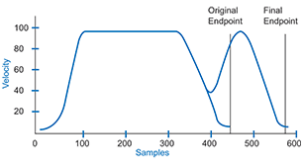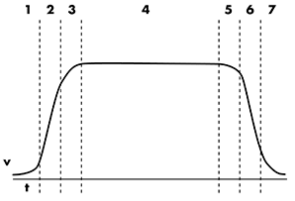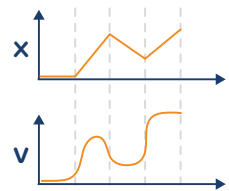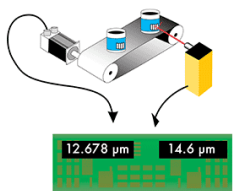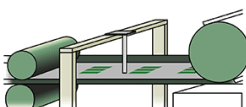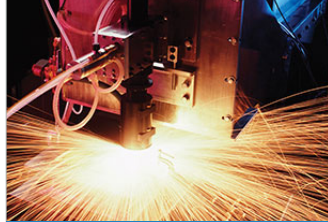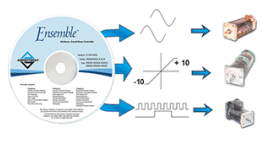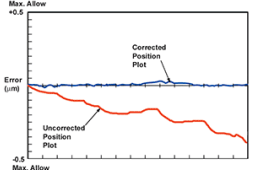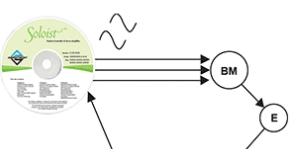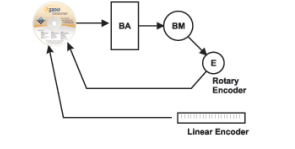
Through the API

These same signals are available for collection and analysis through industry-standard tools using the programming APIs.

Machine Performance with the Soloist Drive-Based Motion Controller

Once set up and ready to go, the system performs. The Soloist controller contains a long list of standard controller capabilities as well as advanced controller capabilities.

These features are all easy to access with the powerful Motion Composer Suite software and controller APIs.

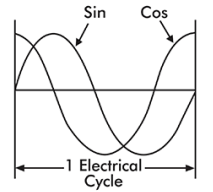|			
Arbitrary Path Generation (PVT)	Electronic Gearing	On the Fly End-Point Modification	Seven Segment Acceleration
Velocity Blending	Fast Position Capture	High-Speed Registration	Analog Power Control
Motor Control	Axis Calibration	Sinusoidal Commutation	Dual-Loop Control
Quadrature Encoder	Analog Feedback	Resolver/Inductosyn	Laser Interferometer
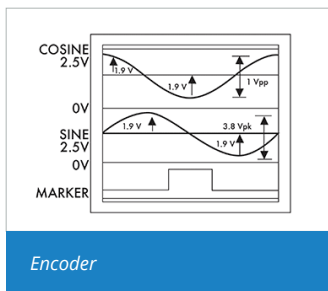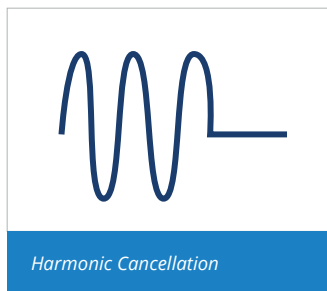
	Encoder
	Harmonic Cancellation
	Dynamic Gain Scheduling
	Command Shaping
	Position Synchronized Output (PSO)
	EasyTune
	Iterative Learning Control
	Enhanced Throughput Module
	Motion Designer
	Alignment Algorithms
	Enhanced Tracking Control
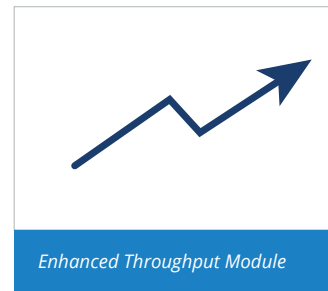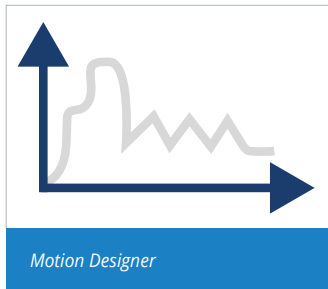
Soloist Drive-Based Motion Controller **Specifications**

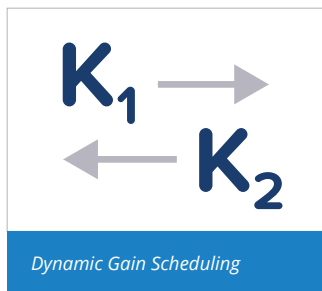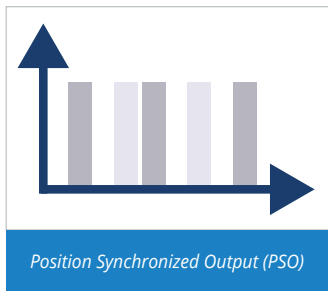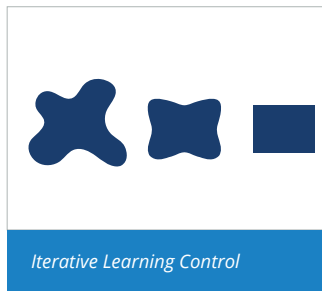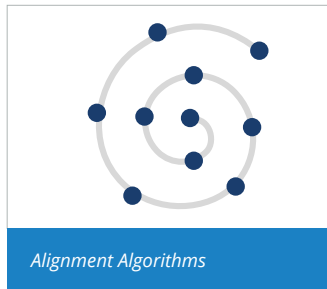| Specification	Description
Axes	Single axis
Programming Tasks	5 tasks available
Position Modes	Absolute, incremental, dynamic trajectory correction
Motion Commands	The Soloist controller includes the following motion commands: • ABORT: Stop any generated trajectory motion on the axis and decelerate the axis to zero velocity • AUTOFOCUS: Enable an outer tracking loop (usually used to control vertical distance from a substrate) • BLOCKMOTION: Prevents motion. The BLOCKMOTION command is a safety feature and is usually executed in a monitor program or a control task program. • DISABLE: Disables an axis • ENABLE: Enable an axis • FAULTACK: Clear axis faults • FREERUN: Move the axis without specifying a target distance • HALT: Cause the profiled motion FIFO queue to hold when new moves are loaded • HOME: Send the axis to its reference position (home) • LINEAR: Move an axis an incremental amount or to an absolute position at a specified speed • MOVEABS: Set the point-to-point movement of an axis to an absolute distance • MOVEINC: Incrementally set the point-to-point movement of an axis from its position • PT: Executes a cubic spline between two adjacent points. Unlike the PVT command, the PT command uses an average velocity to execute motion. • PVT: Executes a cubic spline between two adjacent points • START: Cause the moves that are in the profiled motion FIFO queue to start executing (motion starts) • VELOCITY: Set a velocity profiling mode when blended moves are required • WAIT INPOS: When you use the WAIT INPOS command, the program waits for the preceding motion to finish and reach its specified position before it executes the next program line • WAIT NOWAIT: When you use the WAIT NOWAIT command, the program does not wait for the preceding motion to finish and reach its specified position before it executes the next program line • WAIT MOVEDONE: If you set WAIT MODE to INPOS, the controller determines that the preceding motion has reached its position by using the values specified in the InPositionTime and InPositionDistance parameters. After the position error of the axis is less than the value specified by InPositionDistance, and this is true for an amount of time specified by InPositionTime, the controller considers the axis in position.

Homing Types	Multiple procedures are available in order to establish the home position of an axis • Home past limit switch to home marker • Home to limit switch and reverse to home marker • Home to home marker • Home to limit switch • Home at current position and set to zero • Home at current position and set to absolute position
FIFO queue	In the Soloist, profiled motion can be loaded into a first-in, first-out (FIFO) queue. The controller uses profiled motion FIFO queue when you issue LINEAR motion or PVT motion. Moves loaded in the profiled motion FIFO queue can be halted by using the HALT command. They can be restarted by using the START command. Note: the HALT command is used to set the profiled motion FIFO queue to hold commands instead of executing them. HALT and START pairs can be used to synchronize moves with I/O and other non-motion commands.

Acceleration Profiles	**Acceleration Types**
	• S-curve: Trapezoidal acceleration applied resulting in an "s-curve" velocity profile
	    • SCURVE = 0: a constant acceleration applied resulting in linear velocity profile
	    • SCURVE between 0 and 99: a trapezoidal acceleration applied resulting in an "s-curve" velocity profile
	    • SCURVE = 100: a triangular acceleration applied resulting in an "s-curve" velocity profile
	Acceleration Modes
	• Time based: axis acceleration takes place over a specified time
	• Rate based: axis acceleration takes place at a specified rate
	• Distance based: axis acceleration takes place over a specified distance

Velocity Profiling	Blend multiple coordinated motion commands into one continuous motion path. In this velocity profiling mode, the controller does not decelerate to zero between consecutive coordinated moves. During the move sequence you can change the velocity if necessary. The axes will increase or decrease in speed in a coordinated way such that the programming path is maintained. Without Velocity Profiling With Velocity Profiling
Advanced Features	• High-speed registration • Electronic gearing (with optional auxiliary encoder input) • EasyTune and autotuning • Backlash compensation • Autofocus
Programming	• AeroBasic • C++/CLI, C • .NET • LabVIEW® • MATLAB®

Soloist Software **License Specifications**

Option	Configuration	Description

Soloist	Order Entry Point	Full installation of the Soloist Motion Composer Suite and unlocks certain features on the Soloist drive-based controllers
		
		Pricing is summation of selected products. Maintenance (software update) included in price for one year from date of purchase.
		
		The Soloist Motion Composer Suite is intended for deployment on desktop or industrial PCs
		
		Includes:
		  • Soloist Configuration Manager
		  • Soloist Motion Composer IDE
		  • Soloist Digital Scope
		  • Soloist Console
		  • Soloist Firmware Loader
		  • Soloist Help
		  • Programming Help
		
		The Soloist is not multilingual and cannot be used in applications that require a native-language user interface

License	Machine	Contains both the Soloist Motion Composer Suite and options that enable features on an Soloist Drive-Based Controller
		
		Provides the ability to:
		  • Write, compile, execute, debug programs in AeroBasic
		  • Full access to .NET, C, and C++ libraries
		  • Access full diagnostics, fault, and status information
		  • Access and set I/O, registers, and variables
		  • Collect, analyze, and save data
		  • View files from machine for analysis and record keeping
		  • Connect PC to machine through Ethernet TCP/IP or USB
		  • Upgrades can be installed (firmware or controller) using loader

	Machine Upgrade	• Use to change configuration options on an existing machine license • Requires the current license ID from customer • Price is based on the new options added • Maintenance extension is a separate line item
	Machine Addition	Increases the number of licenses associated with an existing key
	License Extension	• Extends the maintenance period on an existing license • Can be purchased in yearly increments
	Media Only	License ID distributed on the specified media

Soloist Drive-Based Motion Controller **Ordering Options and Details**

Option	Details
Dynamic Controls Toolbox	Contains a collection of controller-level algorithms that are designed to improve machine positioning, increase throughput, and reduce cycle times. Parameters supplied in the toolbox are used to configure these algorithms. **Dynamic and Threshold Gain Scheduling** Adjusts the servo-loop gains based on specific inputs. These algorithms can improve position performance and reduce cycle times. **Harmonic Cancellation** Improves performance by reducing the tracking error to sinusoidal reference commands, motor force ripple or other position-dependent disturbances, and cross-axis disturbances such as that created in a linear axis by an unbalanced rotational axis. **Command Shaping** Filters undesired frequencies from the commanded position, velocity, and acceleration.
Enhanced Throughput Module	The Enhanced Throughput Module (ETM) can improve machine throughput by measuring base or frame vibration directly and using this information in the servo control algorithm.
Enhanced Tracking Control	Enhanced Tracking Control (ETC) improves move-and-settle times in point-to-point positioning and decreases tracking errors that occur during contoured motion. The ETC is a control algorithm that works with a conventional Proportional-Integral-Derivative (PID) servo loop and increases the ability of the servo mechanism to reject low-frequency disturbances that can cause position errors. Examples of low-frequency disturbances include friction, isolation table movement, and cable management. Use ETC only with stages that have a high encoder resolution.

Allen-Bradley EtherNet/IP Interface to Aerotech Soloist

The Aerotech Ethernet/IP interface enables AB PLCs (MicroLogix, CompactLogix, and ControlLogix) to integrate directly with the Soloist motion control solutions. Motion can be directly programmed in the RSLogix 5000 environment or separate programs can be written on the controller and triggered from the AB PLC. Aerotech has two Ethernet/IP interfaces: ASCII and Register. Choose the PLC, motion controller, and Ethernet/IP interface that best fits your application needs.

The Allen-Bradley code snippets provided here are written in the graphical "relay ladder logic" syntax. Allen-Bradley also supports function block and structured text programming languages.

ASCII Command Interface

The ASCII command interface can be used to send ASCII text strings to the Soloist and perform a set of actions such as commanding motion or retrieving diagnostic information. Our vendor-specific EtherNet/IP ASCII command interface object extends this functionality across EtherNet/IP.

The power of the ASCII command interface lies in its simplicity and ease of use. A text string is formed using an AeroBASIC™ command, followed by an End-Of-String (EOS) character. ASCII response data indicates whether or not the command was successfully executed. An ACK character is sent to indicate success, an NAK character is sent if there is a command error, and a FAULT character is sent if there is a task error. For commands that expect return data, the response character is followed by the return data, which is terminated by the EOS character. The EOS, ACK, NAK, and FAULT characters are configurable via Soloist drive parameters.

To send the ASCII command from the PLC (programmed using Rockwell RSLogix) to the Soloist, an MSG block is used (Figure 1). The EtherNet/IP message can be easily configured from within the RSLogix software as shown in Figure 2.
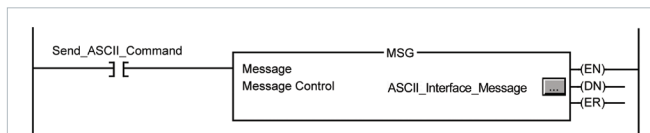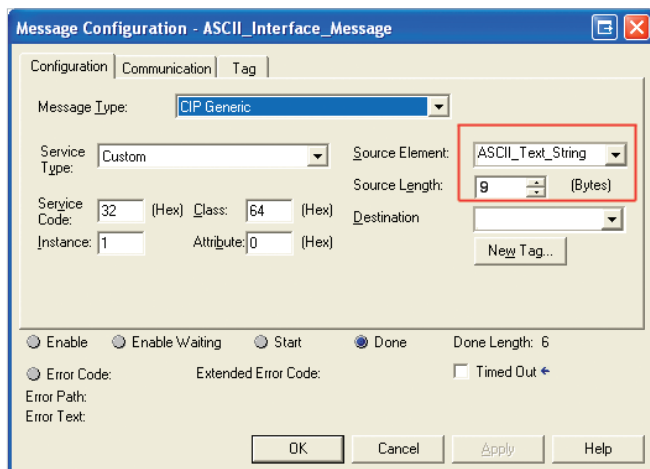

Figure 1. Allen-Bradley code snippet (Ladder Logic) for ASCII interface.

ASCII_Text_String (specified in the red box in Figure 2) is an RSLogix "tag". The value of this tag is the ASCII command that will be sent to the Soloist. The example in Figure 3 shows the command "HOME X Y" as it would be configured in RSLogix.

Figure 3. "HOME X Y" as it would be configured in RSLogix.

When the MSG block is activated, the ASCII command is sent to the Soloist via the EtherNet/IP protocol. The Destination field of the message configuration can be used to specify a local tag for the ASCII command's return data.

Soloist Code Snippet (AeroBASIC) for ASCII Interface

For this interface (see Figure 4), the user does not need to write any AeroBASIC code on the Soloist. The Soloist automatically receives and processes the commands. This functionality is built into our EtherNet/IP module.
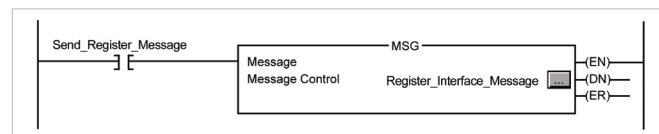

Figure 4. Allen-Bradley code snippet (Ladder Logic) for Register interface.

Register Interface

Our vendor-specific EtherNet/IP register interface object allows the Soloist's built-in integer and double registers to be accessed via EtherNet/IP. This provides a flexible, general-purpose interface to the Soloist that can be adapted to many different applications. Data consistency is guaranteed internally, so there is no need for concern when accessing these registers simultaneously via EtherNet/IP and AeroBASIC programs on the controller.

The Rockwell software is configured in very much the same way as the ASCII command interface. In this case, the message source (shown in red in Figure 5) is a data array. The message configuration in Figure 5 is for the "Write Single Register" service. That service requires two pieces of data: the number of the register to write, and the value to be written. Therefore, we configure the message source tag as shown in Figure 6.
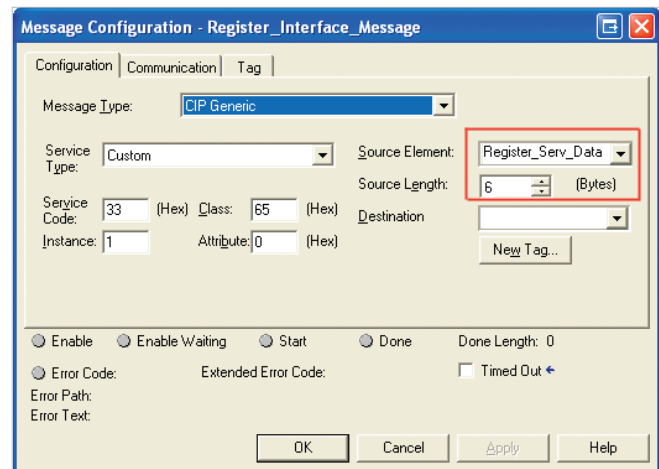
The value 100 is the register to write to, and the value to be written is 7. When the MSG block is activated, the register query is sent to the Soloist via the EtherNet/IP protocol.

Figure 6. Allen-Bradley code snippet (Ladder Logic) for Register interface.

As you can see, this is a straightforward data interface from the PLC to the Soloist. In addition to writing a single register, the interface also supports writing multiple registers in one message and reading single or multiple registers. This example was interfacing with an integer (32-bit) register, but the interface supports the same functionality with double-precision (64-bit) floating point values.

Soloist Code Snippet (AeroBASIC) for Register Interface

To make use of the data that is being transferred to and from the PLC, the user can write an AeroBASIC program to respond to incoming register transfers, as well as write outbound register data. This combination of EtherNet/IP communication with the power and flexibility of AeroBASIC allows for the implementation of many unique applications. For example, the following AeroBASIC code snippet shows how the register interface can be used to control the Soloist timebase value (set by the AeroBASIC TimeScale command).

```
HEADER
        ' The PLC writes timebase values to IntegerRegister 101
        Define timebaseRegisterIndex 101
END HEADER

DECLARATIONS
        ' Declare a global variable named "timebase"
        Global timebase as Integer
END DECLARATIONS

PROGRAM
        ' initialize the timebase register to 100% speed
        RegS IntegerRegisters, timebaseRegisterIndex, 100

        ' ... do work in AeroBASIC ...

        ' Change timebase to value sent from PLC
        Call GetTimebase()
        TimeScale timebase

        ' ... continue working ...
END PROGRAM

FUNCTION GetTimebase() as void
        ' Read the value from IntegerRegister 101 into the
        variable "timebase" timebase = RegS(IntegerRegisters,
        timebaseRegisterIndex)
END FUNCTION
```

Soloist Drive-Based Machine Controller **Ordering Information**

Soloist

Soloist	Soloist Motion Composer Suite

License Options

License	
-Machine	Soloist software installation on a single PC
-Machine upgrade	Soloist software configuration change of license
-Machine addition	Soloist software increase license count for existing key
-License extension	Soloist software extend maintenance period of license
-Media only	Soloist software distribute current license on media

Media	
-Download	Installation media provided for download only
-USB	Installation media provided on a USB drive
-CD	Installation media provided on compact disk

Version	
-Default	Current version of software/controller
-Legacy	Legacy version of software/controller

Maintenance	
-Maintenance-X-00	Software/controller maintenance for x year(s) after purchase where X is one through seven. One year is default.

Drive-Based Controller Configuration

Dynamic Controls Toolbox	
-Dynamic Controls Toolbox	Aerotech Advanced Controls

Enhanced Throughput Module	
-Enhanced Throughput Module	Setup and monitoring screens for ETM modules

Enhanced Tracking Control	
-Enhanced Tracking Control	Reduced dynamic following error and settling times

EtherNet/IP	
-EtherNet/IP	EtherNet/IP class 1 I/O, ASCII command, and register interface objects

Motion Designer	
-Motion Designer	Trajectory creation and evaluation software

LabVIEW	
-LabVIEW	Includes LabVIEW 2010 (forward compatible) VI samples

MATLAB Libraries	
-MATLAB	MATLAB library for motion, parameters, and data collection

Soloist Controller Connection Cable (Items Ordered Separately)

PC to Ensemble Drive/Controller Cable	
ENET-XOVER-xx	Ethernet crossover cable (available in length xx where xx = 9, 15, 30, 45, 60, 75, or 150 dm)
USB-AMBM-xx	USB A-Male to B-Male cable (available in length xx where xx = 5, 10, 30, 50, or 200 dm)
-MATLAB	MATLAB library for motion, parameters and data collection

AeroNet Communication Network (Items Ordered Separately)

Ensemble Drive-to-Drive Cable	
ENET-CAT6-xx	Shielded Ethernet CAT6 cable (available in length xx where xx = 3, 6, 10, 20, 30, 45, 76, or 90 dm)

Remote Server

-Remote	Configure Soloist as server

Motion Composer Suite Add-Ons

CNC Operator Interface

-CNC Operator Interface	HMI CNC software for Windows. Includes the CNC option.

Motion Designer

-Motion Designer	Trajectory creation and evaluation software

Motion Simulator

-Motion Simulator	Trajectory simulation, creation, and evaluation software

LabVIEW

-LabVIEW	Includes LabVIEW 2010 (forward compatible) VI samples

MATLAB Libraries

-MATLAB	MATLAB library for motion, parameters and data collection

FireWire Communication Network (Items Ordered Separately)

NFire-PCIe

NFire-PCIe	FireWire communication network PCIe card (recommended)

NFire-PCI

NFire-PCI	FireWire communication network PCI card

NConnect-6P6P-xx*

NConnect-6P6P-45	FireWire cable (4.50 meter length), 6P to 6P
NConnect-6P6P-30	FireWire cable (3.00 meter length), 6P to 6P
NConnect-6P6P-18	FireWire cable (1.80 meter length), 6P to 6P
NConnect-6P6P-9	FireWire cable (0.90 meter length), 6P to 6P
NConnect-6P6P-5	FireWire cable (0.50 meter length), 6P to 6P
NConnect-6P6P-2.3	FireWire cable (0.23 meter length), 6P to 6P

*Note: Aerotech strongly recommends the use of FireWire cables supplied by Aerotech (NCONNECT cables). These cables are tested to ensure proper performance. A single open PCI or PCIe card slot is required to accommodate the required FireWire PCI or PCIe card for Aerotech's motion bus.

HyperWire Communication Network (Items Ordered Separately)

HyperWire-PCIe

NFire-PCIe	Firewire communication network PCIe card (recommended)

HyperWire-AO10-xx

HyperWire-AO10-200	HyperWire cable (20.0 meter length), SFP to SFP
HyperWire-AO10-50	HyperWire cable (5.0 meter length), SFP to SFP
HyperWire-AO10-30	HyperWire cable (3.0 meter length), SFP to SFP
HyperWire-AO10-10	HyperWire cable (1.0 meter length), SFP to SFP
HyperWire-AO10-5	HyperWire cable (0.5 meter length), SFP to SFP