



LINK Overview



**Instruction
Manual**
RG350678

***LINK* Overview**

Instruction Manual

for use with SSD *LINK* Systems

Copyright © 2009 by Parker Hannifin Corporation, SSD Drives Division

All rights strictly reserved. No part of this document may be stored in a retrieval system, or transmitted, in any form of by any means to persons not employed by a Parker company without written permission from Parker.

Although every effort has been taken to ensure the accuracy of this specification, it may be necessary, without notice, to make amendments or correct omissions in this document. Parker cannot accept responsibility for damage, injury, or expenses resulting therefrom.

Printed in the USA

RG350678 Issue 2

Safety Information



PLEASE READ THIS INFORMATION BEFORE INSTALLING THE SOFTWARE.

Personnel

Qualified personnel should carry out installation, operation and maintenance of the software. A qualified person is someone who is technically competent and familiar with the installation process, operation and maintenance of this software.

Procedures in this manual may contain Warnings and Notes. A Warning gives the reader the information, which, if disregarded, could cause injury or death. A Note furnishes additional information for added emphasis or clarity.

The customer is responsible for assessing his or her ability to carry out the procedures in this manual. Make sure you understand a procedure or the precautions necessary to carry it out safely before beginning. If you are unsure of your ability to perform a function, or have questions about the procedures listed in the manual, contact Parker SSD Drives customer service at (704) 588-3246.

WARNING!

This software is used specifically to configure the control of potentially dangerous motor control. The user assumes all liability and risk for the performance, application, and reliability of all control systems configured with this software. It is the users responsibility to understand thoroughly and to check independently all configurations before commissioning any equipment controlled by this software. Parker SSD Drives accepts no liability for the application of the software.

Table of Contents

Chapter 1 Introduction

What is SSD <i>LINK</i> ?	1
---------------------------------	---

Chapter 2 Computer Tools

SSD <i>LINK</i> Configuration Tools	3
Manager	4
SAM.....	4
ConfigEd.....	4
Communicating With SSD <i>LINK</i> Networks	4

Chapter 3 SSD *LINK* Network

Network Topologies	5
Simple Network Topology	5
Tapped Network Topology	6
Redundant Ring Topology.....	6
PREFERRED DIRECTION FOR THE FIBER OPTIC NETWORK	7
CONSTRUCTING THE FIBER OPTIC NETWORK	8
Transmission Distance for Plastic Cable	8

Chapter 4 Hardware Requirements

MODULE PRODUCT LABEL.....	11
SSD <i>LINK</i> MODULE WIRING	11
Power Supply	12
CABLE VOLTAGE DROP (24V MODULES)	13
SIGNAL ACCURACY AND ANALOG I/O	14
USING AN L5201 ANALOG MODULE FOR DIGITAL INPUT	14
PLC & SUPERVISOR COMPUTER INTERFACE	15
Direct Wiring	15

Chapter 5 Planning a System

SAFETY AND SSD <i>LINK</i> SYSTEMS	17
Emergency Stop	17
The SSD <i>LINK</i> L590 Drive	17
Active Low Inputs and Safety	22
CUSTOM SYSTEMS.....	22

Contents (cont.)

OEM SYSTEMS	23
Chapter 6 Configuring SSD <i>LINK</i>	
CONFIGURING SSD <i>LINK</i> MODULES	25
Function Block Data Sheets	25
Application-Specific Function Blocks	26
General Details of Function Blocks	26
Persistent Inputs	26
Organizing Configurations	26
Network Addresses and Slots	27
Configuration Drawings	28
SSD <i>LINK</i> DATA TYPES	30
Ordinals	30
Logic	31
Values	31
Accuracy and Resolution with Values	31
COMMUNICATION BETWEEN FUNCTION BLOCKS	34
EVENT PROCESSING	35
Event Queue Description	35
Trigger Inputs	35
Forcing Function Blocks to Trigger	36
Disconnecting Trigger Inputs	37
SSD <i>LINK</i> Inputs to a Module Configuration	37
Timebase Function Blocks	38
Multiple Timebase Function Blocks	39
Event Queue Overflow	39
Events and System Synchronization	39
Initialization	40
Execution Times	40
INSTALLATION AND STARTUP	42
Chapter 7 Configuration Procedures	
CONFIGURATION STORAGE AND SECURITY	45
Project Ownership and Write Protection	46
DESIGNING SSD <i>LINK</i> SYSTEMS	46
Generate Project	46
Floppy Disks	46
Configuration Design	46
TESTING (CONNECTED TO HUB DATABASE)	47

CONTENTS (CONT.)

Test Procedure.....	47
Backing Up and Updating.....	47
TESTING (NOT CONNECTED TO HUB DATABASE)	47
Before Testing	47
Test Procedure.....	47
Backing Up and Updating.....	47
Modifying Configurations on the Hub Database	48
SHIPPING SYSTEMS AFTER TESTING	48
COMMISSIONING SSD <i>LINK</i> SYSTEMS.....	48
Obtaining the Configurations	48
Daily Commissioning	48
Backing Up and Updating.....	49
Before Leaving Site	49
SERVICING SSD <i>LINK</i> SYSTEMS.....	49
Restoring Failed Modules.....	49
Obtaining the Configurations	49
Changing the Configuration or Set Up.....	49
Backing Up and Updating.....	49
Before Leaving Site	50

Chapter 8 Troubleshooting

NETWORK EXCEPTIONS.....	51
DETECTING SSD <i>LINK</i> NETWORK PROBLEMS.....	51
Operator Station Network Monitor	52
Logic/System Monitor.....	52
Logic/Watchdog	52
Network Errors and the L5203L/Exception Logger	52
DETECTING MODULE FAILURES	53
Health LED	54
Module List	54
L Error Failures.....	55
IMPROPER OPERATION	56

Appendix A SSD *LINK* Database

DATABASE DESCRIPTION.....	57
SAM AND CONFIGED DATABASES	58
THE ID.DAT FILE.....	58

Contents (cont.)

Copying the SSD <i>LINK</i> Database	59
DATABASE AND PROJECT MANAGEMENT	59
PROJECT CONFIGURATION FILES	59
Configuration Files and SAM.....	59
Updating the Configuration Files	60
Upgrading the Project	61

Appendix B Module Memory

THE I/O LIBRARIES.....	63
PATCH FILE.....	64
HOW MUCH CAN FIT IN EACH MODULE?	64
TEST POINTS	64
MEMORY BLOCKS.....	65
MEMORY SPACE AND THE L5102 OPERATOR STATION	65
SSD <i>LINK</i> OPERATING SYSTEM AND MEMORY	65
What is a Class?	65
What is an Instance?	65
What is Stored in the Module?	65

Appendix C Configuration Guidelines

SYSTEM CONTROL	67
HANDLERS	67
Operator Station Display Handler.....	68
Alarms Handler.....	68
L5102 OPERATOR STATION	69
Memory Limitations	69
Display Legends and Highlights.....	69
Operator Station Potentiometers and Meters	71
Potentiometers.....	71
L5203 SERIAL MODULE	74
Process Control and Monitoring	75
Event/Fault Logging and Diagnostics	77
FILTERS	77
Simple Filter.....	77
Filter.....	78
Decimation Filter	78

CONTENTS (CONT.)

TEST POINTS	78
Appendix D ROM Issues	
BACKUP AND RESTORE	81
Appendix E Grounding Issues	
GROUND LOOPS	83
CROSS-COUPLED GROUNDS	84
USE OF A 0V BUSBAR TO CONNECT POWER SUPPLIES	86

Contents (cont.)

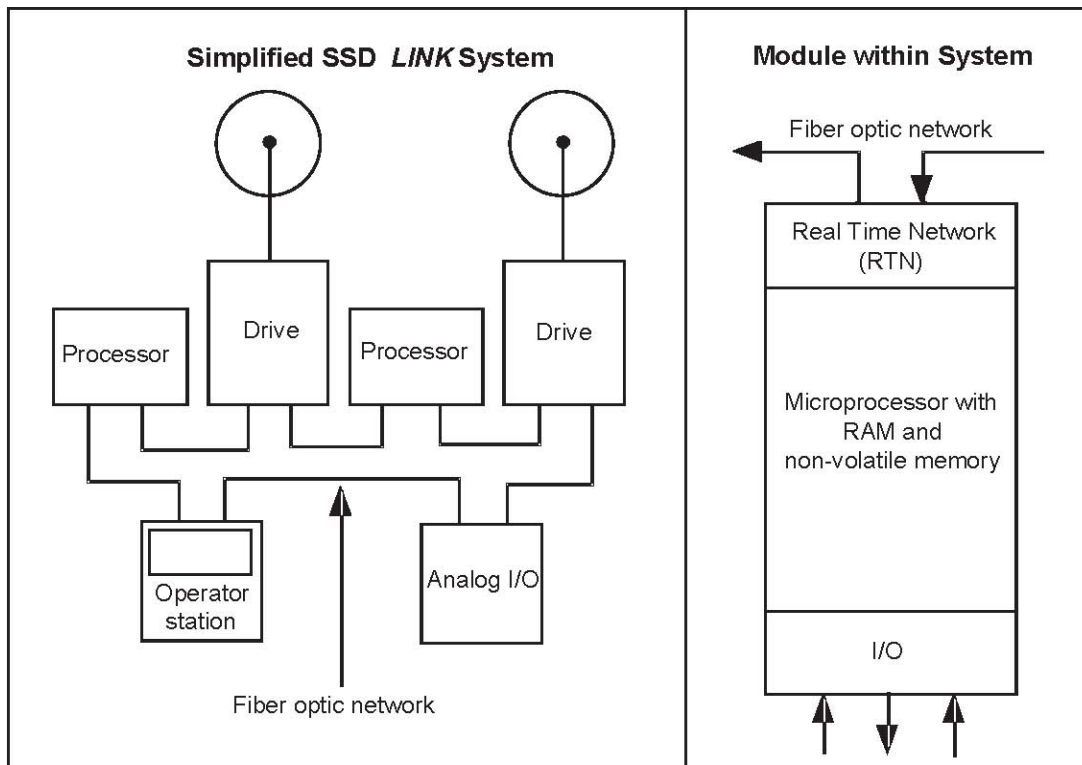
This page intentionally left blank

Chapter 1 Introduction

WHAT IS SSD *LINK* ?

SSD *LINK* is a distributed control system designed for the management and control of motor drives and associated equipment; it is also well suited for process control and other operations. SSD *LINK* is event-driven, in that an event taking place in one module on the network triggers the sending of a message to another module, which then responds to the message in an appropriate manner.

All the modules in an SSD *LINK* system are connected by a high speed fiber optic cable network, which eliminates most of the signal and control wiring present in a conventional system. This network may be configured in different ways to suit specific needs; see Chapter 7 for information on network topology.



Most modules in an SSD *LINK* system are software-configurable and can perform a wide range of control and processing functions; others are more specialized and constrained in their abilities.

Currently, elements comprising the SSD *LINK* system include:

- Drives connected directly to motors to control torque and speed;
- Operator stations presenting information to the operator and accepting commands from the operator;
- Analog I/O modules interfacing with analog signals such as dancer potentiometers and E to P converters;
- Digital I/O modules interfacing with digital signals such as push buttons, lamps, relays, and PLCs;
- Serial I/O modules interfacing with processor-based equipment such as temperature controllers, gauges, PLCs, and ticket printers;
- Interface cards used in supervisory computers;

- Gateway modules providing an interface to other factory networks;
- Repeater modules used to extend the physical size (i.e., the distance covered by the fiber optic cable) of the network;
- Software tools used to configure, commission, and monitor the SSD *LINK* system.

The software tools are designed to run on IBM PC-compatible computers, and include Manager, Setup And Monitor (SAM), and Configuration Editor (ConfigEd). Module configurations (i.e., operating instructions programmed and installed by the user) are composed of function blocks and other elements designated through the use of ConfigEd. ConfigEd's graphical interface simplifies the design and implementation of module configurations. It and the other software tools also allow the user to install, monitor, service, modify, and backup configurations throughout the network. For more information on the software tools, see Section 2 and their respective manuals.

Chapter 2 Computer Tools

SSD *LINK* is a configurable drive systems product range. In order to configure the modules, an IBM PC-compatible computer with a 386 (or greater) processor running Windows 3.1 (or later) and a mouse or other pointing device is required along with configuration software tools from SSD Drives. For requirements specific to individual tools, refer to the manuals for those tools.

SSD *LINK* CONFIGURATION TOOLS

In order for SSD *LINK* modules to be configured and set up, software tools (Manager, Setup And Monitor [SAM], and Configuration Editor [ConfigEd]) from SSD Drives are required. The software tools use program-specific hardware keys which must be plugged into the parallel printer port of the programming computer. A hardware key for ConfigEd also allows the use of SAM; a key for SAM alone will not allow the use of ConfigEd. Manager is included with both SAM and ConfigEd and will work with either key.

The software tools access the `ssd_link` database, which contains both pre-defined files (libraries, new module configurations, function blocks, etc.) and user-created projects. Project directories contain user-created configurations and support files for each system being designed. See Appendix A for details of the SSD *LINK* database, and the individual tools manuals for information on creating and maintaining the database and projects.

WARNING

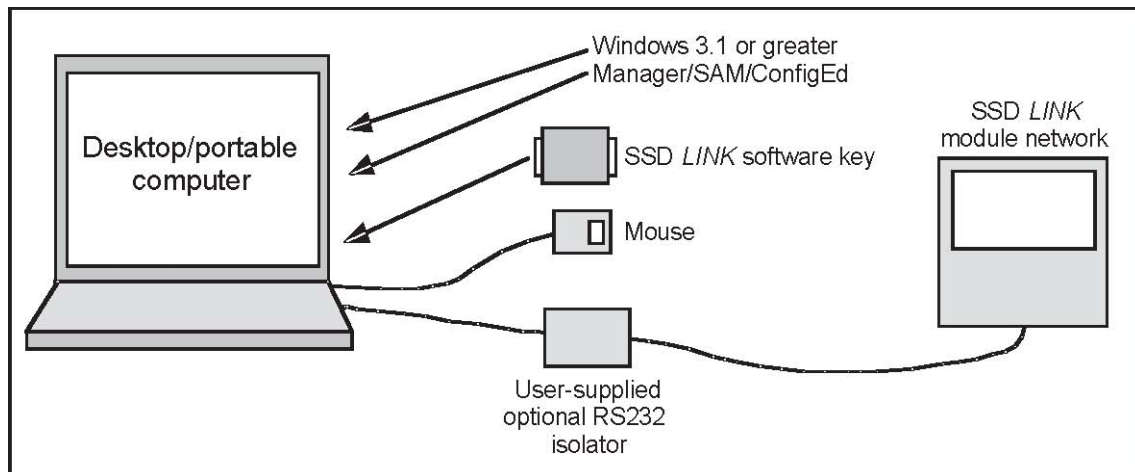
The SSD *LINK* database must *never* be copied from one computer to another. If this is done, the `id.dat` file will be duplicated and project corruption will occur. SSD Drives will not accept any responsibility for project corruption due to duplicate databases.

The tools are also supplied with a cable to allow connection of a serial port on the computer to the RTNX connection (similar in appearance to a telephone handset connection) on the SSD *LINK* module.

WARNING

Do not use any connectors, adapters, and/or cables other than those supplied or recommended by SSD Drives. Failure to use recommended materials can result in severe damage to equipment and injury to personnel, and will void the SSDDrives warranty.

The software tools are supplied on 3.5" disks and installation is provided through the Manager program, also included on the disks.



Manager

Manager allows users to import, export, archive, and delete projects and import databases. Manager is required to install SAM and ConfigEd, as well as to install new and revised databases. Manager is included with the purchase of either SAM or ConfigEd.

SAM

SAM allows users to commission, test, backup, restore, and service configured SSD *LINK* systems on site. All users and customers of SSD *LINK* systems should have access to a copy of SAM. Your purchase of SAM includes a key to enable your computer to run the software.

ConfigEd

ConfigEd allows users to design configurations using a full block diagram graphical interface and install them into SSD *LINK* modules. The key supplied with ConfigEd also allows SAM to be used, and SAM is included in your purchase of ConfigEd.

Communicating With SSD *LINK* Networks

All SSD *LINK* I/O modules are supplied with an RTNX serial port connection (similar in appearance to a telephone handset socket) for communicating with an RS232 serial port on a computer equipped with either SAM or ConfigEd. L590 drive modules and most gateways do not contain these ports, and so cannot be accessed or controlled directly from your computer. Once you connect your computer to any single SSD *LINK* module through the RS232/RTNX connection, however, you will have global access to all modules, drives, and gateways in that network.

WARNING

Care should be taken to use only connectors, adapters and cables supplied or recommended by SSD Drives. Since most modules also provide 24vdc power through the RTNX connection, failure to use recommended materials can result in severe damage to equipment.

Communication rates between your computer and the network are selected in the Comms menu in either SAM or ConfigEd; communication between SSD *LINK* modules through the fiber optic network is fixed at 2.7 megabaud.

If the computer being used is grounded through its power cord and the module through which it connects to the SSD *LINK* system is also grounded, a ground loop is formed and the ground noise produced may disturb the RTNX communications link. Both SAM and ConfigEd have built-in retry mechanisms to deal with occasional electrical noise, but this type of continuous noise will cause error messages and/or extremely slow communications between the computer and the attached module.

In systems with a large amount of electrical noise (e.g., systems containing inverters) it may be necessary to break the ground loop to achieve usable communications. Breaking the ground loop may be accomplished by:

- Installing a commercially-available RS232 isolator module between the computer and the RTNX socket;
- Connecting the computer to the RTNX port of an ungrounded SSD *LINK* module (e.g., an L5207 processor module connected to an ungrounded 24V power supply);
- Using a battery-powered notebook computer rather than a plug-in model (since most notebook computers are not grounded).

Chapter 3 SSD LINK Network

The SSD *LINK* modules are interconnected using a high speed digital network with the following specifications:

- Fully distributed peer-to-peer communications.
- Up to 3,000 SSD *LINK* modules possible on a single network;
- 2.7M Bits per second data transfer;
- 12,000 messages per second network bandwidth per segment;
- Hardware error detection;
- Hardware fiber optic failure detection and recovery;
- Optional fault-tolerant dual ring.

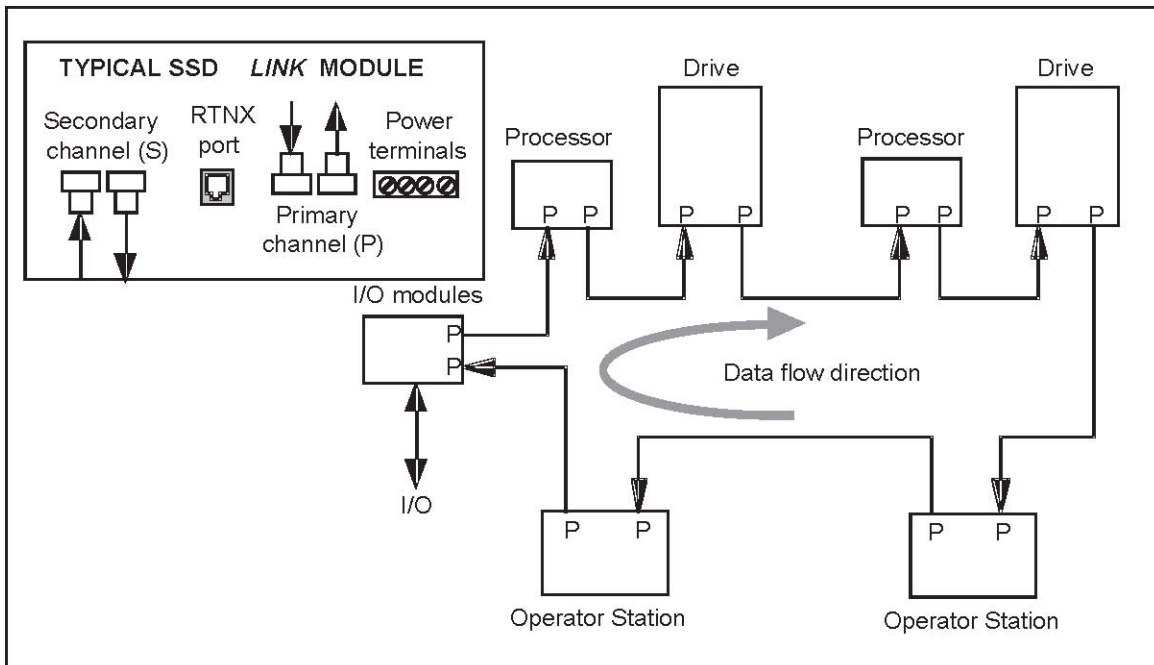
The SSD *LINK* network is based on a ring topology and uses fiber optic cable to transmit the data. Each SSD *LINK* module has one or two network interfaces.

NETWORK TOPOLOGIES

The network is arranged in one of three topologies: simple, tapped, or redundant ring.

Simple Network Topology

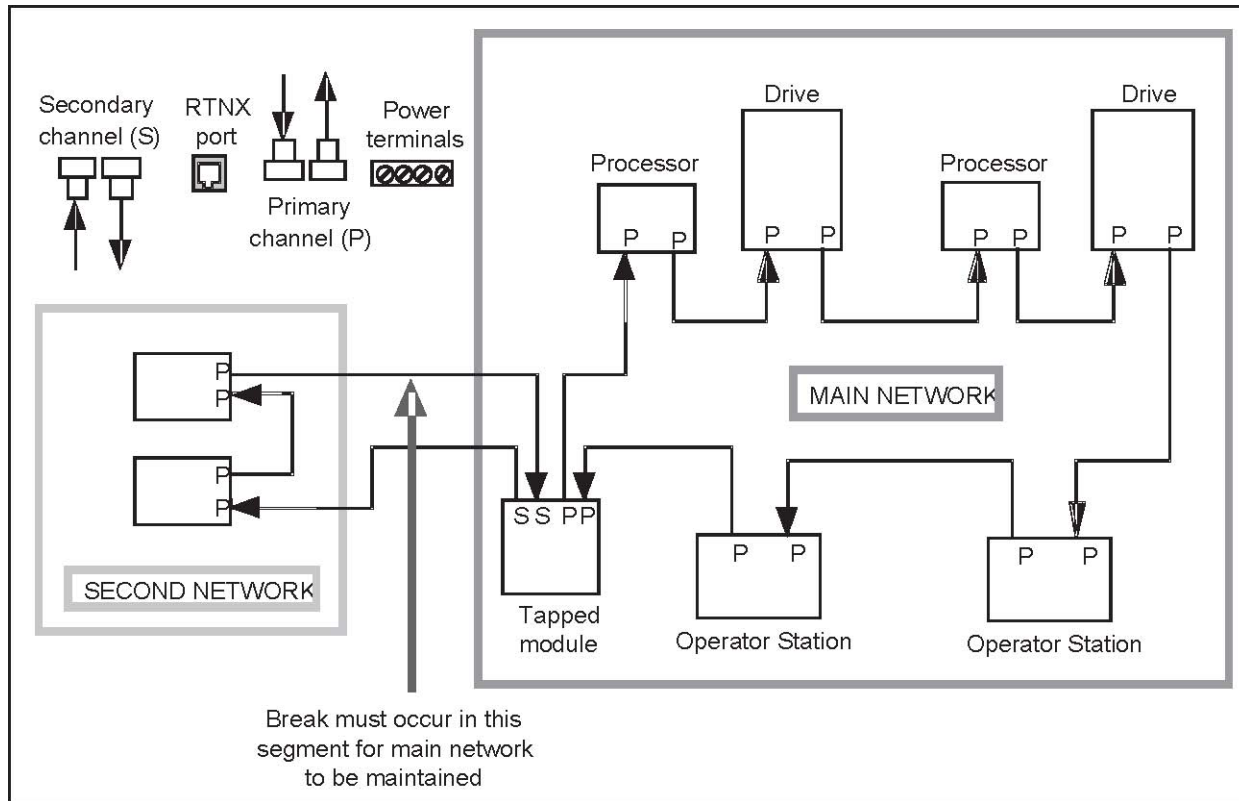
This is the most cost-effective SSD *LINK* network topology. Each SSD *LINK* module has at least one fiber optic transmitter (red) and fiber optic receiver (black). When the simple network is used, the set next to the module's power terminals is used (as shown in the diagram). This is known as the *primary* channel.



The data passes around the network in the direction shown by the arrow. The network operates by the source module placing data onto the network. Data then passes through the various SSD *LINK* modules until the final destination is reached. At this point, the data is removed from the network. Protection is built in to remove messages that pass completely around the network to the source module.

Tapped Network Topology

The tapped arrangement is useful for larger systems (e.g., those with multiple cabinets). It is also useful if part of the SSD *LINK* network has to be disconnected from the main network or powered off.



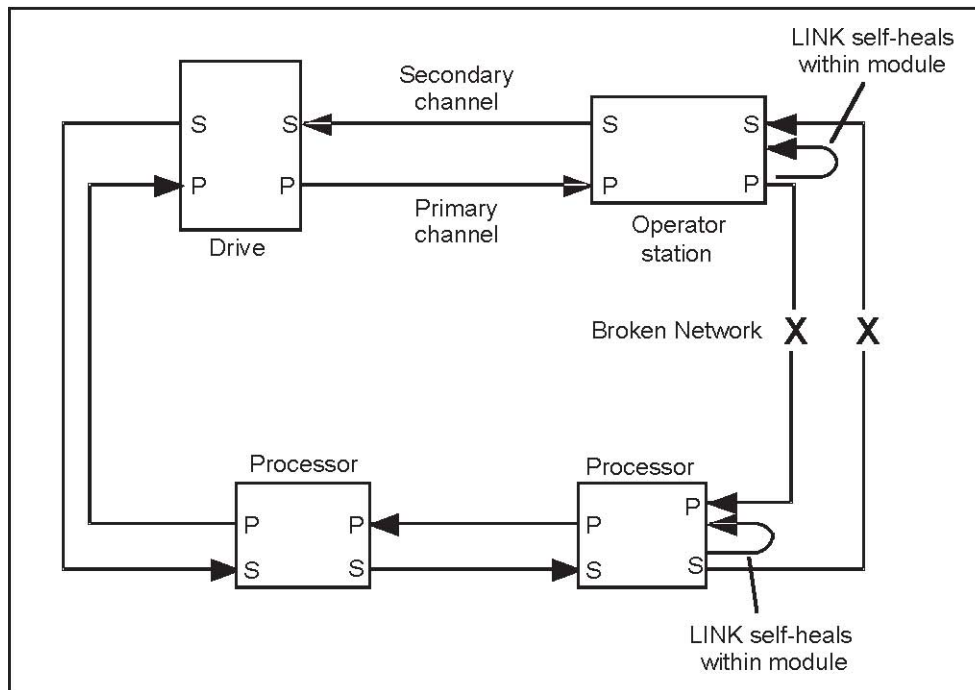
The above diagram shows the tapped network topology. Data flows from the main network to the second network via the second set of fiber optic connectors, or *secondary* channel connectors, on the tapped module. The tapped module is a standard SSD *LINK* module using both sets of fiber optic connectors. All other SSD *LINK* modules on both networks use only the primary connectors.

If the second SSD *LINK* network is disconnected from the tap module (either accidentally through a break in the fiber optic cable or on purpose by removing the fiber optic cable from the secondary channel terminals), the main SSD *LINK* network is automatically maintained if the break occurs in the segment from the second network back to the tapped module. The SSD *LINK* modules can be programmed to indicate when the second network has been disconnected, which can be useful to allow different functionality when part of the system is disconnected or powered-down.

Powering down the tapped module will break both networks, as the tapped module is part of the main ring. However, the second network can be powered down without affecting the main network.

Redundant Ring Topology

The simple networks described above have no fault tolerance. If the network is broken or disconnected in some way, most data transfer will cease. This will usually cause the application controlled by SSD *LINK* to stop.



The optional second network can be used to provide enhanced system security as shown in the following diagram.

The second network is inactive but checked when the primary network is healthy. If a break occurs on the primary network, the SSD *LINK* modules on either side of the break route the data via the secondary channel as shown. This "self-healing" action recreates the continuous ring and allows data transfer to continue. The redundant ring will also self-heal if both networks are broken at the same place, but the redundant ring topology will repair only a single network failure.

SSD *LINK* modules can be programmed to indicate partial or complete SSD *LINK* network failure (e.g., to allow a controlled machine shutdown). Examples of this are given later in this manual.

The network may also self-heal if an SSD *LINK* module fails. However, since the module is typically an integral part of the system, the application may fail anyway.

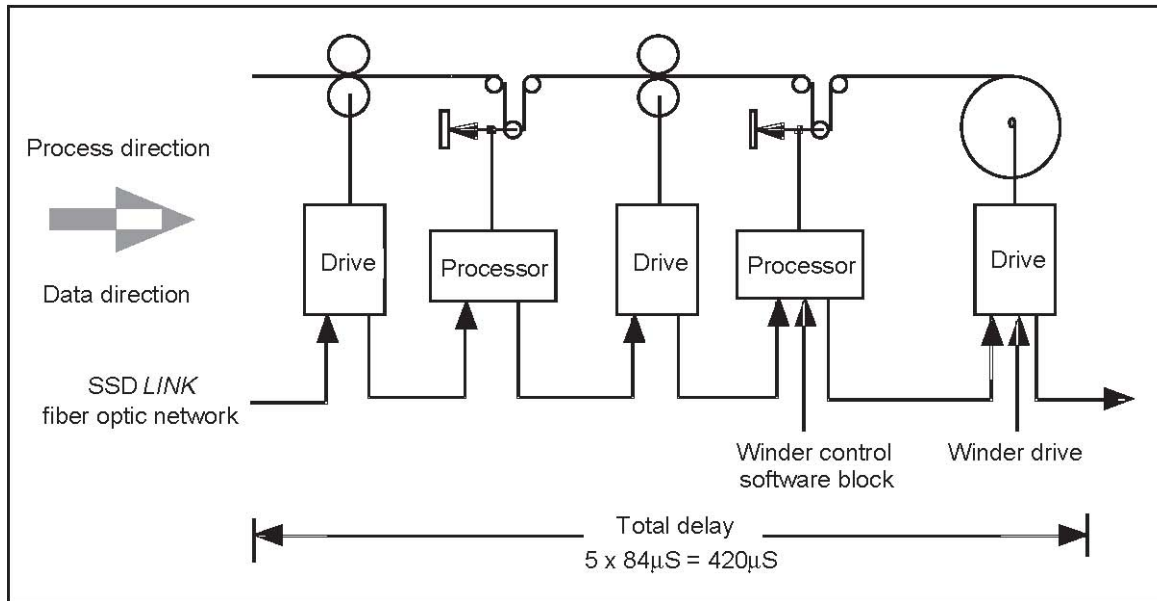
Note

Since they both use the secondary channel, it is not possible to have a redundant ring and a tapped topology on the same networked system.

PREFERRED DIRECTION FOR THE FIBER OPTIC NETWORK

Data messages may have to pass through many SSD *LINK* modules before they reach their final destination. It takes 84μS for a message not addressed to a particular module to pass through it. Therefore, the total delay through a chain of modules (including repeaters) is equal to the number of modules in the chain x 84μS.

This delay is quite short and therefore the assignment of functions to modules and their order in the ring is not critical. It is recommended, however, that the data flow follows the process direction shown below where possible. This will result in optimum system performance.



CONSTRUCTING THE FIBER OPTIC NETWORK

The most common type of fiber optic used with SSD *LINK* systems is plastic multi-mode cable which is inexpensive, easy to terminate and suitable for short to medium transmission distances. For long transmission distances, the glass cable option is preferable. The following guideline information is given for plastic cable.

Transmission Distance for Plastic Cable

Any fiber optic system will have a defined operating transmission distance. If the actual transmission distance is too short for the transmitter in operation, the fiber optic receiver may saturate due to excessive light; if it is too long, the fiber optic receiver will be under driven. In either case, data will not be received.

Most SSD *LINK* modules have a fixed transmission power and operate over a range of 0-20m/0-65', assuming the cable is correctly terminated and the minimum bend radius of 50mm/2" is not compromised. The fiber optic cable must not be abused during installation as it is sensitive to stretching, which fogs the cable and greatly increases attenuation.

For some SSD *LINK* modules, typically drive and repeater modules, the transmission power can be selected. The following table gives the transmission distance for these SSD *LINK* modules:

Fiber Optic Drive Power Level	Distance
Low	0-20m/0-65'
Medium	20-40m/65-130'
High	40-60m/130-195'

Drive levels are set by means of a switch on the module. These levels are intended only as a guide; they are by no means perfectly defined, since other factors must be taken into account (e.g., the number and sharpness of any bends in the fiber).

If the power level is not set properly, the resulting marginal operation may cause a spurious L Error of the module in the system and the errant operation may not become apparent for some time. Therefore, light meters should be used during initial system startup to measure the intensity of the light at the receiving module and the level adjusted at the sending module to ensure proper light intensity.

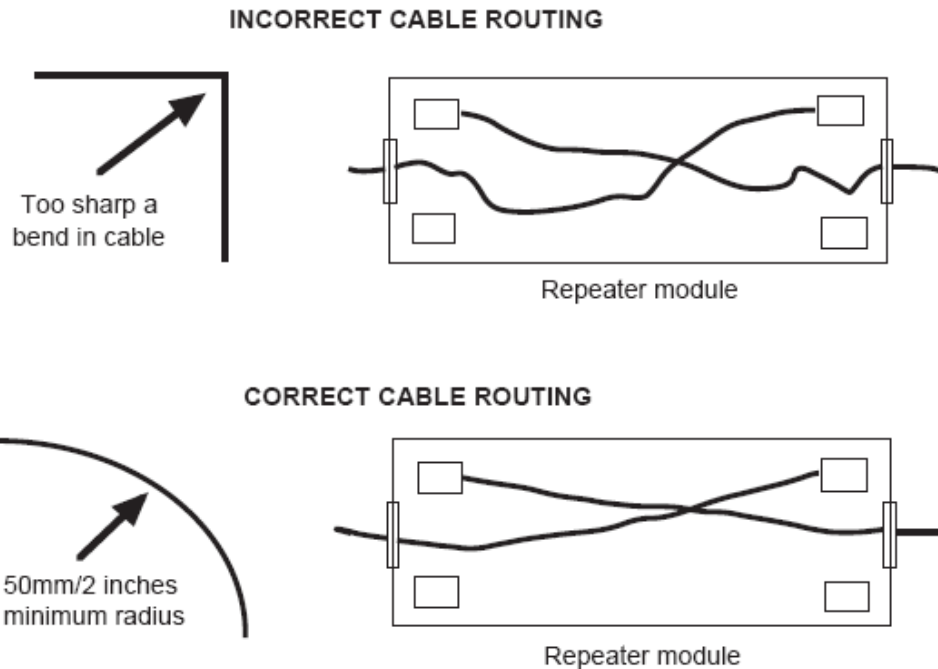
SSD Drives recommends the purchase of a fiber optic light meter by all customers involved in the design of new SSD *LINK* systems; one suitable light meter is the 3M Photodyne 18XTA. Follow the meter's operating instructions carefully and refer to

SSD Drives guidelines (part #HW351772) for appropriate methods of taking light level readings. The T&B plastic receivers have an ideal operating range of -20 to -31.5 dbm.

Routing and Terminating Correctly

In order to achieve the specified transmission distance, the fiber optic cable must be correctly terminated. Termination is achieved through the use of special connectors which cut the cable when they are assembled. It is vital the assembly instructions are followed closely to achieve the specified transmission range.

If the fiber optic cables are routed incorrectly, the transmission range will suffer. Cable runs should be as straight as possible, avoiding sharp bends. The minimum bend radius for the single plastic fiber cable is 50mm/2". When fiber cables are terminated inside a housing, the connection should be direct to the connector; unnecessary bends and turns should be avoided. For examples of bend radius and routing within a housing, see the diagram below.



Cable Damage

If the fiber optic cable is damaged in any way, transmission distance will be severely compromised. Always ensure:

- The cable external sheath is completely undamaged;
- The cable is never subjected to tension or other mechanical forces. Optical cable is not installed the same way as electrical wiring (e.g., cable ties should not be used to secure cabling) and installation personnel must be properly trained in its use.

If the cable is compressed (e.g., by using cable ties or by walking on it), it may be irreparably damaged. Transmission distance will be impaired even though the cable appears externally undamaged.

Composite Cable

A composite fiber optic cable is available from SSD Drives. This consists of a single fiber, with two electrical cables, in a single sheath. This provides a much stronger cable than the single fiber. This cable should be used whenever damage to the fiber may occur. This is especially true where the cable runs between enclosures. The two electrical cables can be used to power remote SSD *LINK* network repeater modules (part #L5206-2-00).

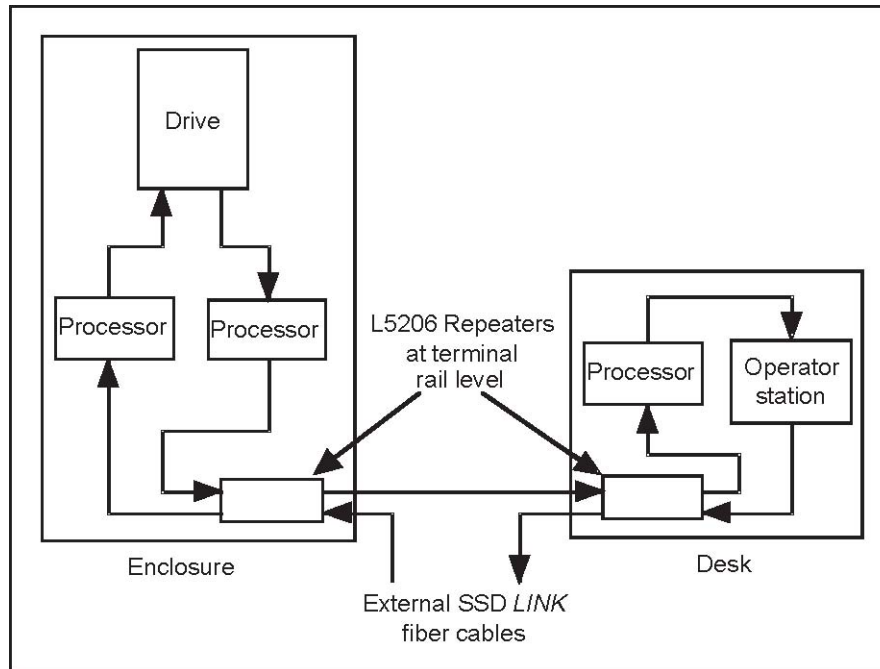
Unused Fiber Optic Receivers

If there are SSD *LINK* modules in which the secondary channel is not used, the unused fiber optic receiver must be blocked using the block foam plugs provided with the modules or a spare fiber optic plug. Failure to block this receiver could cause spurious network failure due to ambient light entering the uncovered receiver. These failures are extremely difficult to find.

Repeaters

Repeaters are provided to allow maximum transmission distance to be increased. Each repeater is mounted in a heavy duty die-cast box, designed for under-floor mounting. It is powered from +24V DC and the composite cable is ideal for supplying the power.

Each repeater has two channels and switchable drive power (see data sheet). Often, the transmission distance between enclosure and desk is not known. As a general rule, therefore, a repeater should be used at the exit and entry of the network to an enclosure or desk as shown below.



Use of a repeater in this way creates a convenient fiber optic termination point to allow the external network to connect easily to the enclosures and desks.

Plastic and Glass Cable

If the fiber optic cable run is greater than 60m/195', another option is to convert to glass cable. This is done via a special L5206 repeater module (part #L5206-2-02), which serves to convert between the plastic cable and the glass cable.

On-site termination of the present generation glass cable may be difficult, since special tools are required to cut and terminate the cable correctly and safely. A correctly terminated and installed glass fiber cable, however, will allow transmission distances in excess of 500m/ 1625'. SSD Drives can provide terminated lengths of glass cable or arrange for on-site termination if required.

Chapter 4 Hardware Requirements

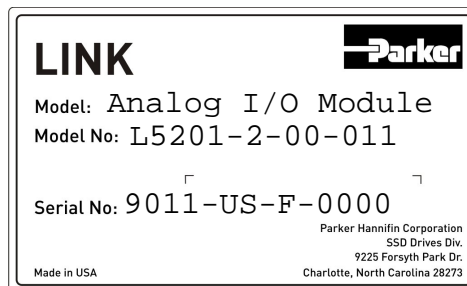
The SSD *LINK* system must be installed and wired properly to get consistently safe and accurate operation of each module. The guidelines that follow provide general information on power and signal wiring, grounding, and some specific wiring do's and don'ts for particular SSD *LINK* modules.

MODULE PRODUCT LABEL

Each SSD *LINK* module carries a product label that identifies all SSD *LINK* system products except drives; drives have their own product code format and should be ordered using that code. In either case, the full module name and model number must be completed to specify any SSD *LINK* module.

The module product label contains three pieces of information about the module: the model name (Analog I/O, Processor, etc.); the model number (L5201-2-02, L5207-2-00, etc.); and the serial number. Modules already in the field may have the serial number typed on the product label; current module product labels have a second label attached with the serial number represented by both a bar code and a row of numerals.

The following diagram shows a typical SSD *LINK* module label; in this example, the label is for an L5201 Analog I/O module.



The complete product model number should be specified when ordering an SSD *LINK* module; it follows the format L5XXX-Y-ZZ. This number may be followed by an issue number, which designates the manufacturing revision level (e.g., ISSUE 2) for that type of module. The model number consists of the following three parts:

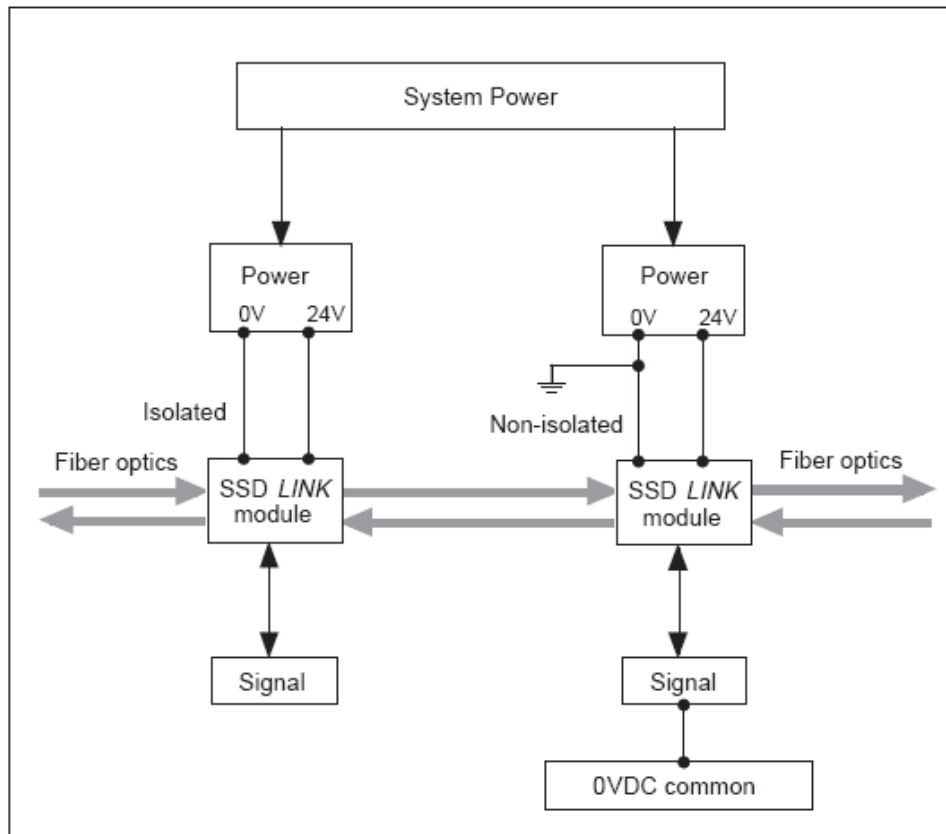
- Product Number (L5XXX) ... Identifies the type of module (e.g., L5207 processor;
- Network Option (Y) ... A single digit code specifying either just a primary (1) or both primary and secondary (2) fiber optic rings. The default build standard is a module with both primary and secondary rings;
- Hardware Number (ZZ) ... A two digit code relating to any hardware builds included in the product (e.g., 02).

SSD LINK MODULE WIRING

The diagram below illustrates the two different ways SSD *LINK* modules can be wired into a system. The module on the left is isolated with respect to the system ground and 0VDC references, and the module on the right is non-isolated. Isolated modules are used when interfacing to signals which are referenced to a local common which is floating or at a different potential than the 0VDC reference. Non-isolated modules are used when interfacing to signals referenced to the system 0VDC line, or signals which are grounded for safety reasons.

WARNING

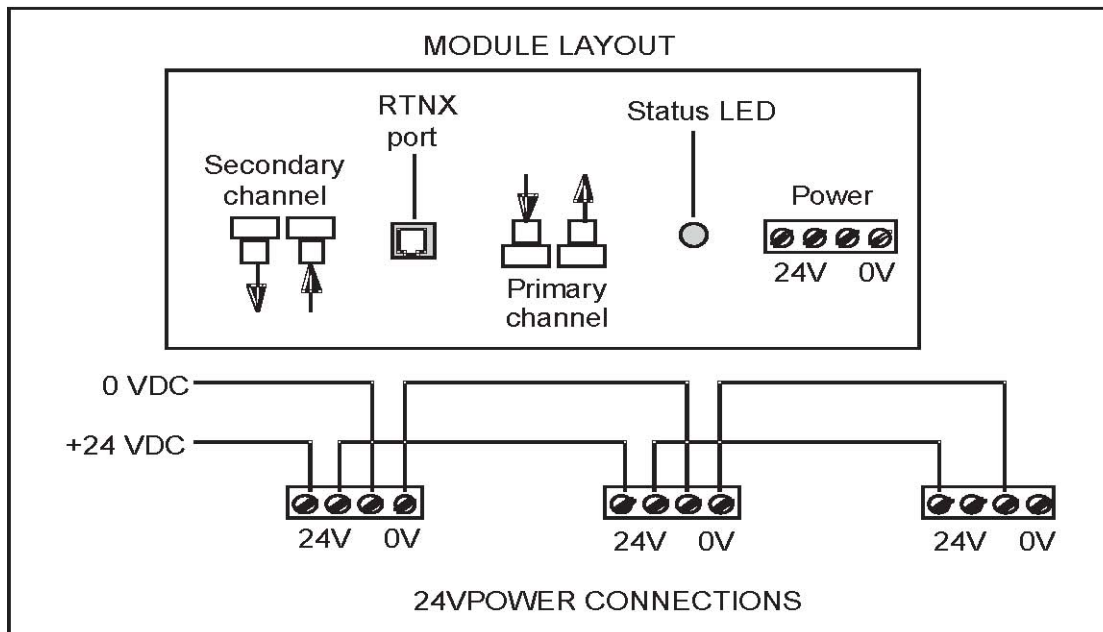
Safety concerns are of paramount importance when configuring and wiring any industrial equipment. Always obey local electrical codes and regulations and adhere to industry standards when wiring SSD *LINK* Systems.



Power Supply

The following diagram shows the network fiber optic connections (primary channel and secondary channel) and power terminals for a typical SSD LINK module. The power terminals are duplicated to allow daisy-chaining of the +24V and 0V power connections. The maximum current that can flow through a pair of terminals is 8A.

Some SSD LINK modules (e.g., drives, gateways, repeaters, and modems) may not conform exactly to the above diagram. Appropriate documentation should always be consulted before connecting SSD LINK modules.



Use the module data sheets to calculate the total 24VDC supply amperage required by your particular assortment of modules and then choose a power supply rated for that load. In general, it is not permissible to connect two or more

power supplies together in parallel in an attempt to boost available current capacity. This could damage the supplies and cause the SSD *LINK* system to shut down. Some SSD *LINK* modules, especially the L5102 operator station, draw significant surge current on power-up and power supplies must be chosen to tolerate this. Power supplies with foldback current limits can be problematic unless carefully sized.

Use only wire designed, approved, and intended for use in power circuits when wiring from the 24VDC supply to the SSD *LINK* modules. Select power cable rated for the continuous load-current required by the SSD *LINK* modules, and use an appropriately-sized fuse to protect the power supply and cables. Do not use signal wire for this purpose, as its undesirable high impedance (which also results in power losses) is a potential fire hazard in the case of overloads caused by an accidental short-circuit or other fault.

In most cases, the power common or 0VDC signal common will be grounded at one place in a system. However, it may sometimes be necessary to provide a separate, isolated power supply for a particular module. This is the case for L5202 Digital I/O modules, which use active-low input channels.

An active-low logic channel is energized when its input is pulled low. An external wiring fault which causes an input wire to be accidentally connected to ground will look like a valid input if the power supply common or signal common lines are grounded. If this input is a safety interlock, or can cause machinery to go into motion (such as a start-switch input), then a safety hazard exists. To prevent a potential safety hazard, do not connect the module's power common or 0VDC signal commons to ground when using a module with active-low digital inputs (see Section 5.1.3).

The L5102 and the L5206 have the 0V terminal internally connected to the case, which will cause the 0V to be grounded. The L5209 digital input module uses active-high inputs, and can be safely grounded.

Signal wiring

Signal wiring should be kept separate from power wiring wherever possible to prevent power transients from interfering with signal voltages and currents. Shielded-pair or twisted-pair conductors (e.g., Belden 8723, UL 2493) should be used for analog signals and serial communications signals. Signal wiring should be run in separate wiring troughs or raceways, and where power and signal wires must cross, route them so they cross at right angles to minimize interference.

When using shielded cables for sensitive signals, only one end of the cable shield will normally be terminated. The shield should be left unconnected at the signal source and terminated at the signal destination; this is done to prevent currents from flowing along the shield as a result of slightly different ground voltages which may exist at the cable ends, and to keep interference to a minimum at the receiving end of the signal.

For best performance, pay close attention to the interconnection of grounds to prevent ground loops and cross-coupling, as described in detail below.

CABLE VOLTAGE DROP (24V MODULES)

Since currents on the order of 1 to 1.5A may be required for each unit, it is recommend that you supply modules from power supplies local to the enclosure or desk, thus avoiding long cable runs. Driving modules over long power supply cables may cause problems related to voltage drop.

The following table shows the voltage drop per meter of typical stranded cable:

AWG	Cable Size		Voltage Drop Volts/A/m
	SWG	mm ²	
20	21	0.5	0.07
18	19	0.75	0.053
16	18	1.0	0.035
14	17	1.5	0.024
12	15	2.5	0.015
10	13	4.0	0.0091

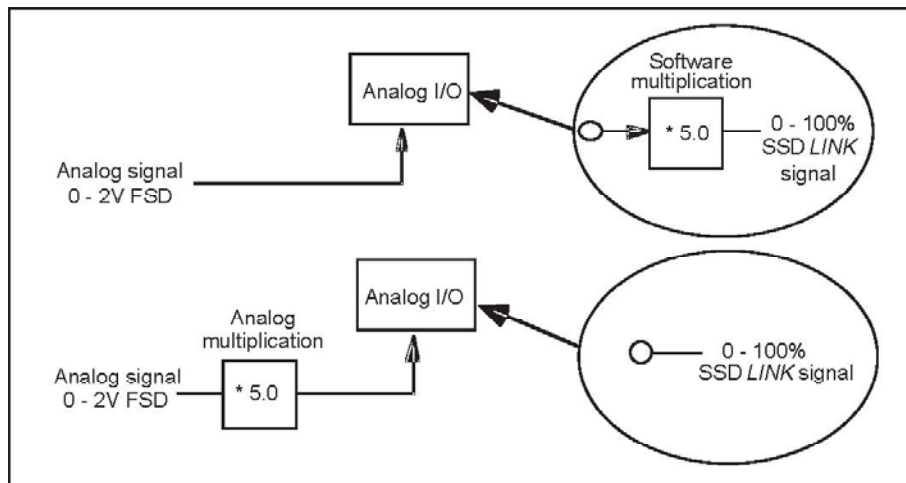
WARNING

Under no circumstances should any line voltage supply be connected to any terminal on an SSD *LINK* systems module.

The use of +24V and 0V common busbars is recommended for the power supplies, and the cables connecting the modules to the power supply should be kept short (see Appendix E). The I/O modules may be connected in a small chain, providing the voltage drop is carefully monitored. The supply range for most 24V SSD *LINK* modules is 20V to 28V. Any voltage supply below 20V will cause module shutdown; any voltage above 30V will damage the module. In all cases, refer to the relevant data sheet before connecting SSD *LINK* modules to power supplies. Additional power supplies may be desirable for modules that are remotely mounted (e.g., an operator station mounted on a panel some distance from the modules it is monitoring).

SIGNAL ACCURACY AND ANALOG I/O

With an analog interface to a digital system, it is good practice to ensure the analog input covers the full range of the input. Always use as much of the 0 to 10V range as possible to maximize input resolution. The following diagram shows where a 0 to 100% value derived from a low voltage input is required.



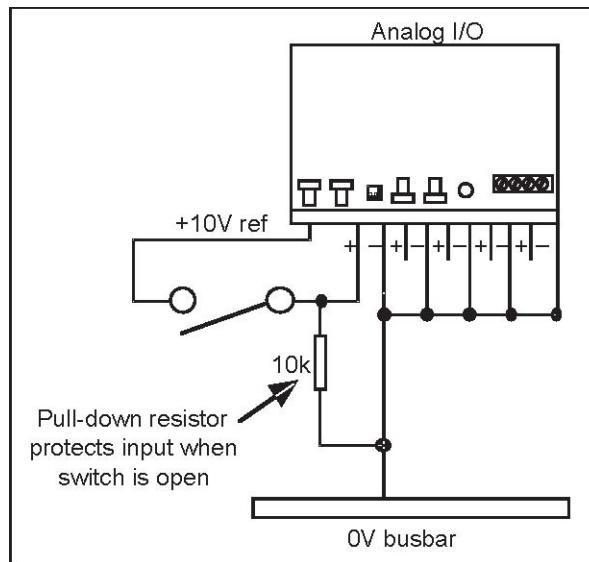
The preferred treatment is to amplify the analog signal before connecting to the analog input. If this is not done, the resolution and accuracy of the analog input is degraded by the amount of software multiplication required.

The most common case for amplification is where load cells are required to be interfaced to the SSD *LINK* analog inputs. Two load cell amplifiers are available from SSD Drives:

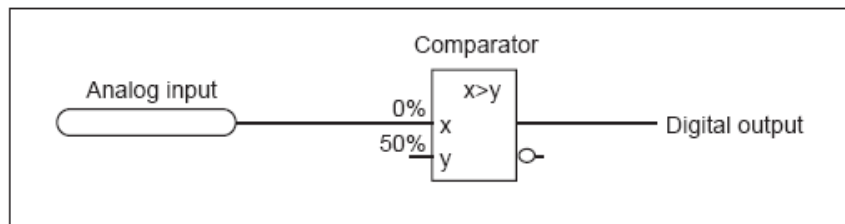
- 5530, for strain gauge (SG) type load cells;
- 5530-1, for LVDT-type load cells.

USING AN L5201 ANALOG MODULE FOR DIGITAL INPUT

It is possible to use an analog input of an L5201 module as a digital input, as shown in the illustration on the following page.



The 10k resistor will pull down the analog input to 0% when the switch is open. When the switch is closed, the analog input will go to 10V (100%). Inside the module, the analog input can be connected to a Logic Comparator function block (with the “y” input preset to 50%) to convert the input value to a logic state. See the following illustration.

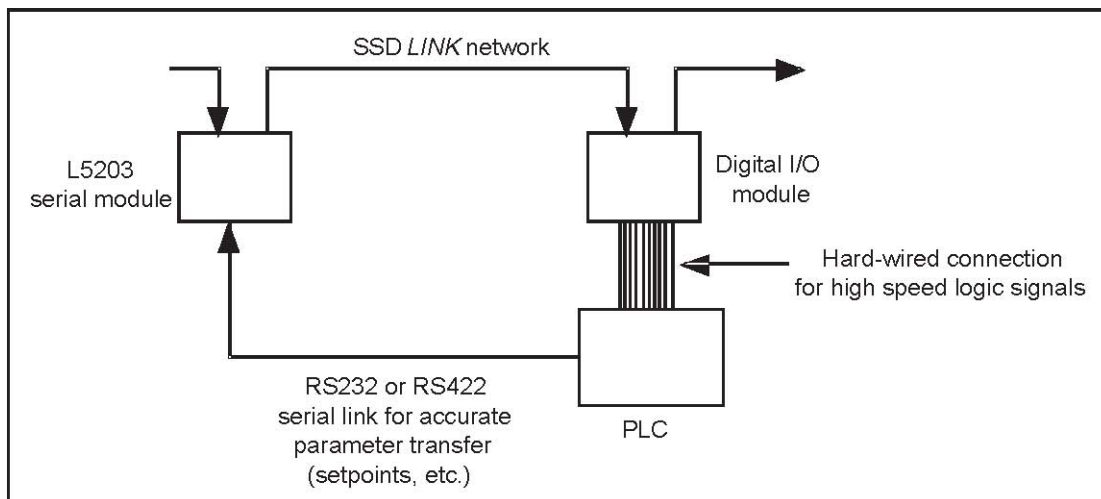


PLC & SUPERVISOR COMPUTER INTERFACE

There are several ways an SSD *LINK* system can interface to PLC and supervisor computers.

Direct Wiring

The simplest and potentially most cost-effective method uses direct wiring between SSD *LINK* digital I/O modules and the PLC as illustrated in the following diagram.



This technique is suitable only for the connection of logic signals between the PLC and SSD *LINK*. If it is required to transfer values, a hybrid system can be used. A digital I/O module is used to transfer the logic signals and an L5203 serial module can be used to transfer values.

Many PLCs and supervisor computers have programmable communications drivers. These can be easily programmed to suit the L5203 BiSynch protocol (see the L5203 operating manual for details).

Since SSD *LINK* is a programmable control system, the amount of hard-wiring between the PLC and SSD *LINK* can be reduced. For example, a single wire connection can be used to start a number of drives; the drive start logic can be implemented within the SSD *LINK* modules.

For OEM customers who are responsible for machine sequencing, the application sequencing can be split into two areas:

- Drive sequencing, implemented in the SSD *LINK* modules;
- Machine sequencing, implemented in a PLC.

Chapter 5 Planning a System

Before an SSD *LINK* system can be configured, the number of modules required to give the desired functionality must be determined. Total system functionality can be split into three types:

- I/O interface;
- Drives;
- Signal processing.

With SSD *LINK*, a dramatic increase in system functionality can be achieved. This section provides some guidelines to help reduce the amount of engineering effort needed when designing systems.

SAFETY AND SSD *LINK* SYSTEMS

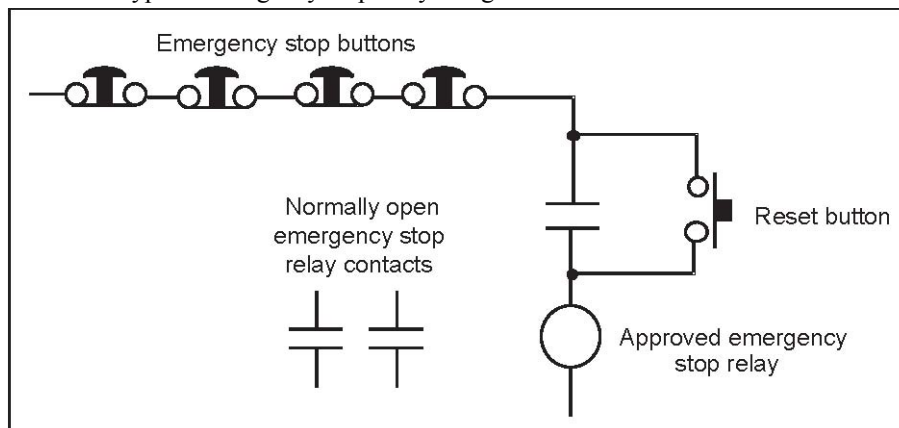
The SSD *LINK* system is typically used to control potentially hazardous moving machinery. With any control system, it is vital to ensure operator safety and the possible effects of failure must be considered to avoid any possibility of injury or damage due to unexpected movement of machine parts. It is the responsibility of the machine designer to ensure that a thorough examination of all hardware and software is carried out prior to start up of a system. This section gives some guidelines as a starting point for these safety considerations.

It must be stressed these are guidelines only. Each system should be carefully considered in its own right, and appropriate standards and regulations should be followed. The SSD *LINK* network should not be used to implement an emergency stop since, if a failure occurs, the emergency stop may operate in an unpredictable way.

Emergency Stop

An effective emergency stop system is important to avoid equipment damage and injury to personnel should the process or the control system fail. A typical emergency stop system would consist of approved mushroom-head push buttons placed at strategic intervals around the machine to be protected. Pushing any of these buttons should stop the movement of the machine in a highly defined manner, satisfying the appropriate safety standards.

The following diagram shows a typical emergency stop relay design.



The emergency stop relay should be of an approved design to ensure reliable operation. The emergency stop relay contacts are used to provide predictable machine operation when the buttons are operated.

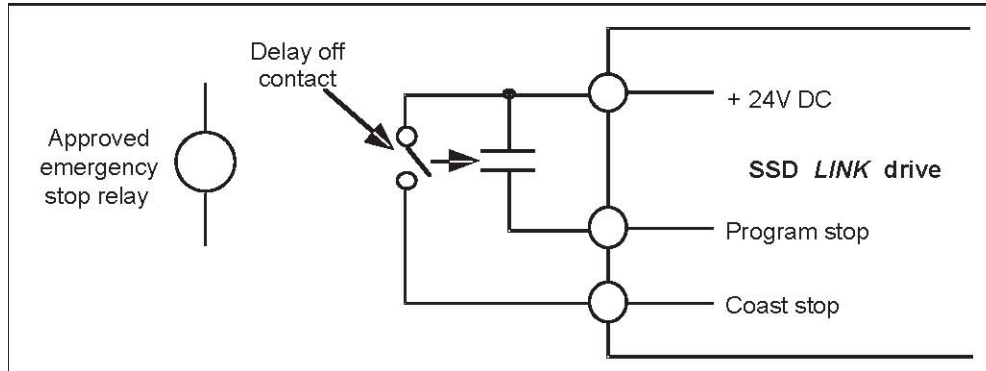
The SSD *LINK* L590 Drive

The L590 drive has hard-wired terminals for connection to the emergency stop relay as outlined above. Two terminals are used, which have the following function:

- Program Stop: Forces the drive to ramp to zero speed at a pre-defined rate and predefined torque limit;

- **Coast Stop:** Forces the main contactor of the drive to be opened. There are no electronic switching devices in series with this terminal. The coast stop terminal is connected directly to the internal relay controlling the drive start contactor. The armature current is quenched before the contactor opens.

The program stop and coast stop terminals are designed to stop the drive independently of the SSD *LINK* network. The following diagram indicates how the two hardware terminals are used with the emergency stop relay outlined above.



The emergency stop relay contacts are wired to the program stop and coast stop terminals. The program stop terminal is operated via a direct contact and the coast stop a delay off contact of the emergency stop relay.

When the emergency stop relay is opened, the drive is forced to program stop. The drive will ramp to zero speed using regenerative operation, if applicable. When zero speed is achieved, the drive will stop.

The delay off contact shown should be set to a time which exceeds the ramp to zero time under program stop. If the drive fails to stop due to a malfunction, the delay off contact forces the drive to shut down via the coast stop terminal.

Safety and Drive Start

The SSD *LINK* network can be used to start and stop drives controlling potentially dangerous machinery; the touch screen operator station can also be used for a similar function. As with any electronic industrial control system, the effect of a single failure should be carefully examined. It is quite possible that a single programming error or a module failure could cause drives to start. This could cause equipment damage or injury to personnel.

It is most important that some form of electro-mechanical protection be provided to prevent this single failure risk when personnel are working on the machine. This could be implemented using the emergency stop system outlined above. It is the responsibility of the machine designer to ensure the final machine user is aware of the need to trip the emergency stop before working on the machine.

There are cases where moving machinery is unguarded and, therefore, exposed to the operator (e.g., cleaning a machine where the machine may be jogged while mechanical guards are open). Operation of the drive hardware stop terminals during machine jog and start should be considered.

SSD *LINK* Network Fail and the SSD *LINK* Drive

If the SSD *LINK* network should fail for any reason, it is likely that the drive will lose the ability to communicate with other SSD *LINK* modules. The drive can be programmed to perform a number of actions, should network communications be lost:

- Regenerate to zero speed;
- Coast to zero;
- Do nothing.

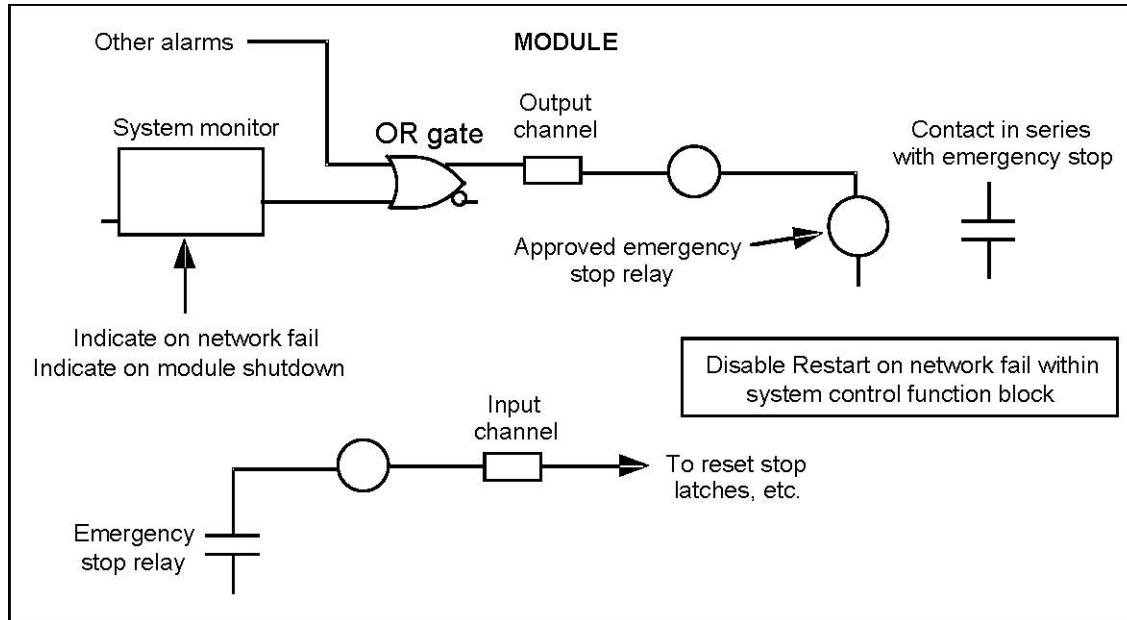
If the SSD *LINK* network fails, the action taken should be carefully considered for each specific application. These actions are overridden by the hard-wired drive stop inputs described above. Refer to the L570 or L590 drive manuals for more information on SSD *LINK* network failure actions.

Emergency Stop and the SSD *LINK* Network

The SSD *LINK* network itself can have the ability to operate the emergency stop relay if required under the following conditions:

- SSD *LINK* network fail;
- Normal programmed system alarms.

Function blocks exist which allow the normal system alarms to be collected and prioritized within an SSD *LINK* module. The following diagram gives guidelines to interface an SSD *LINK* digital I/O module to the hard-wired emergency stop system. This configuration must be placed in a *single* digital I/O module.



A system monitor function block is used to indicate an SSD *LINK* network failure. This will cause a digital output to trip the emergency stop relay as indicated. The system monitor function block is configurable to allow detection of a number of SSD *LINK* network faults. The default configuration is to detect SSD *LINK* network failures only.

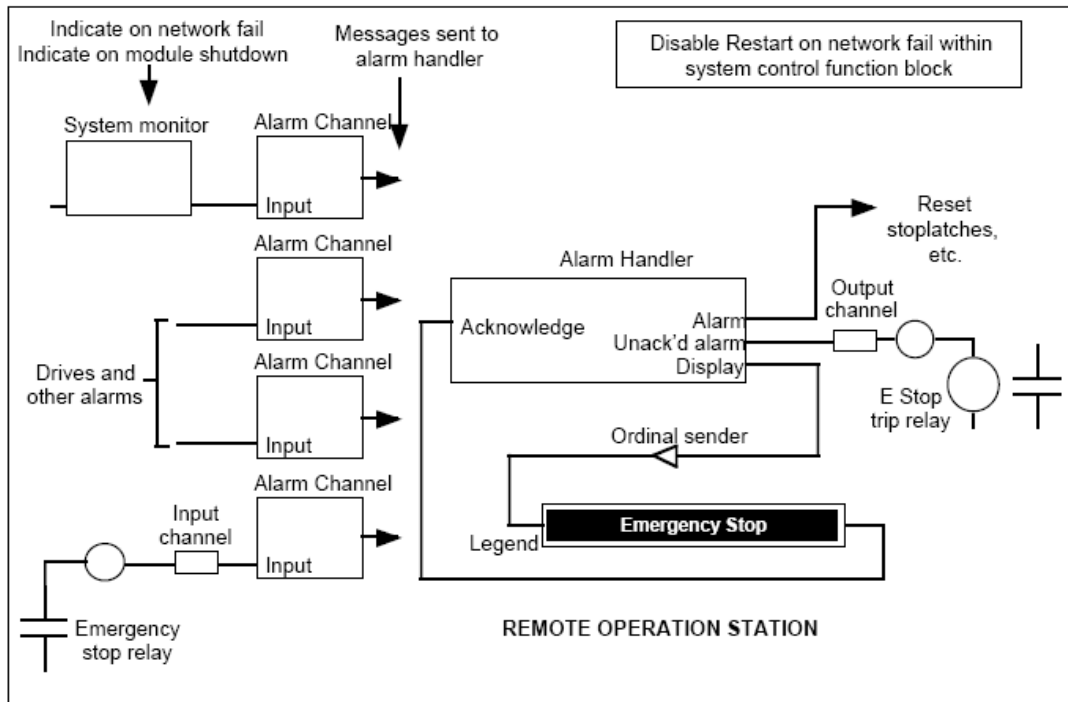
Other alarms generated within SSD *LINK* can also trip the relay via the OR gate. If the emergency stop is operated outside the SSD *LINK* network, a contact is used to indicate this to the network as shown. This could be used, for example, to reset any drive start latches within the configuration. This is important to prevent the drives from restarting when the emergency stop system is reset.

In order for the configuration to work properly, it is important that the digital module shown above is prevented from shutting down during SSD *LINK* network failure conditions. This can be done by disabling the Restart On Network Fail option in the system control function block in the module configuration file.

It is important that the emergency stop relay is set to a safe state if the above digital module shuts down due to other reasons. It is recommended that the relay on the output of the digital module should be energized in order to reset the emergency stop line. If the digital module shuts down or power is lost, the relay will de-energize and the emergency stop will be activated.

Emergency Stop and the Alarms Handler

The alarms handler function block is used to collect many SSD *LINK* system alarms and to provide indication on an operator station display. The alarms handler can also provide logic outputs to reset SSD *LINK* system latches and the emergency stop system. The following diagram shows a digital I/O module with the alarms handler used for this application.

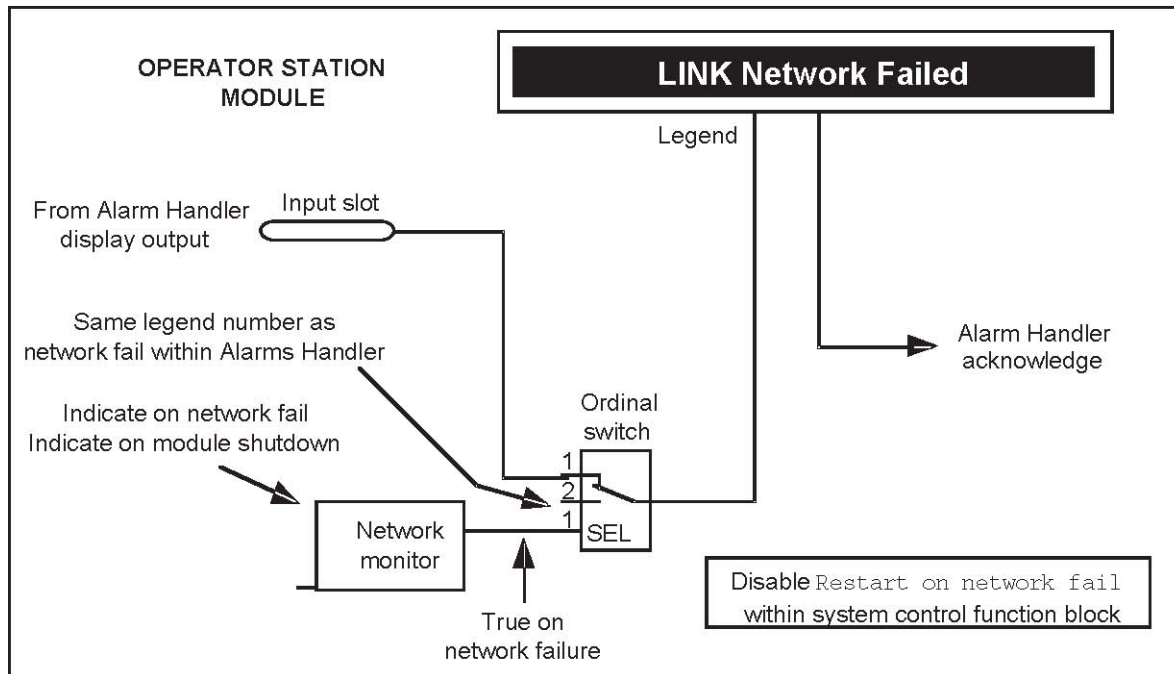


The alarms handler and the emergency stop relay inputs and outputs must be in the same digital I/O module. In order for the configuration to work properly, it is important that the above digital module is prevented from shutting down during SSD *LINK* network failure conditions. This can be done by disabling Restart On Network Fail option in the system control function block in the module configuration file. The alarm output from the handler is used to reset any latches within the SSD *LINK* system. This is important to prevent the drives from restarting when the emergency stop system is reset.

The unacknowledged alarm output is used to trip the emergency stop relay, and the display output is connected to a remote operator station legend to indicate the alarm.

In order to reset the alarms handler, the operator station annunciator must be pressed until all alarms have been acknowledged. This clears the unacknowledged alarms output and allows the hardware emergency stop to be reset. When the emergency stop is reset, the alarms handler will be cleared, allowing the machine to start (assuming all other alarms have been cleared). Refer to the SSD *LINK* function block data sheets for more information on the alarms handler.

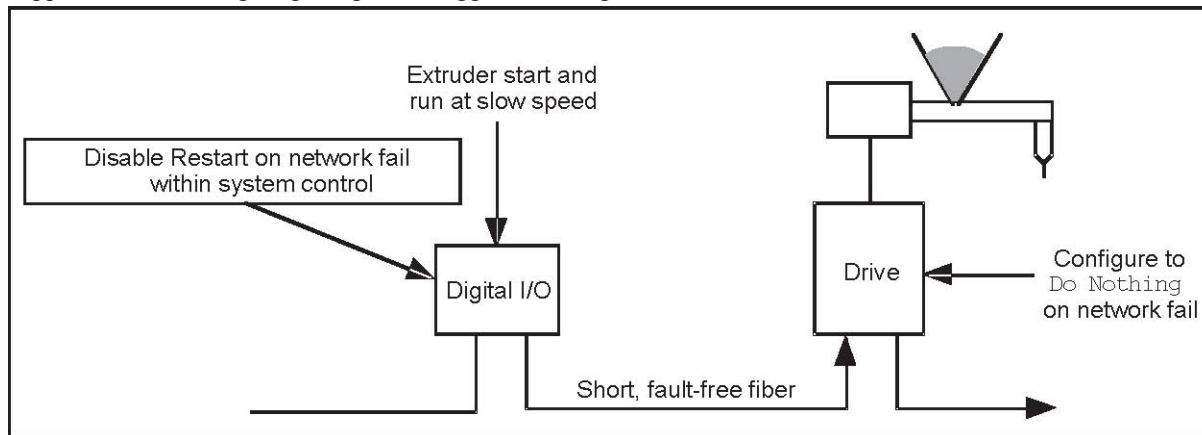
If the SSD *LINK* network is broken, it is possible that the break is between the digital module shown above and the display operator station. If this happens, the reason for the alarm will not be shown since the operator station will shut down. The following suggested L5102 operator station configuration indicates how this can be overcome.



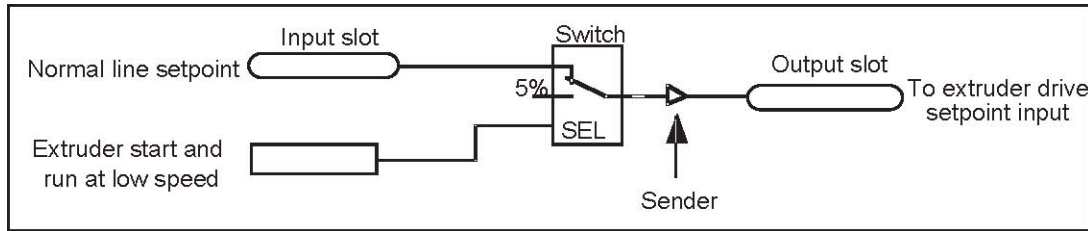
The operator station legend is switched to indicate the SSD *LINK* network fail condition. The switched legend number must be the same as the SSD *LINK* network fail legend number from the alarms handler. Since the SSD *LINK* system is event-driven, it is possible to connect the input slot from the alarms handler and the SSD *LINK* fail switch together. The last operation will take priority.

Forcing Drives to Run at a Slow Speed on SSD *LINK* Fail

Certain applications require some drives within a system to run at a slow speed under SSD *LINK* network fail conditions. Typical applications are extruder systems, where mechanical damage could occur if the drive is stopped should a network fail happen. The following diagram gives a suggested configuration.



The SSD *LINK* digital module shown should be mounted close to the extruder drive. The length of fiber optic cable between the digital module and drive should be designated \pm fault free \sqrt and suitable precautions taken to prevent damage. The digital module should be configured to provide a slow speed setpoint to the drive in the event of failure. In this example, the slow speed setpoint is switched from a hardware switch connected to a single input of the digital module. A sample digital module configuration is shown below.

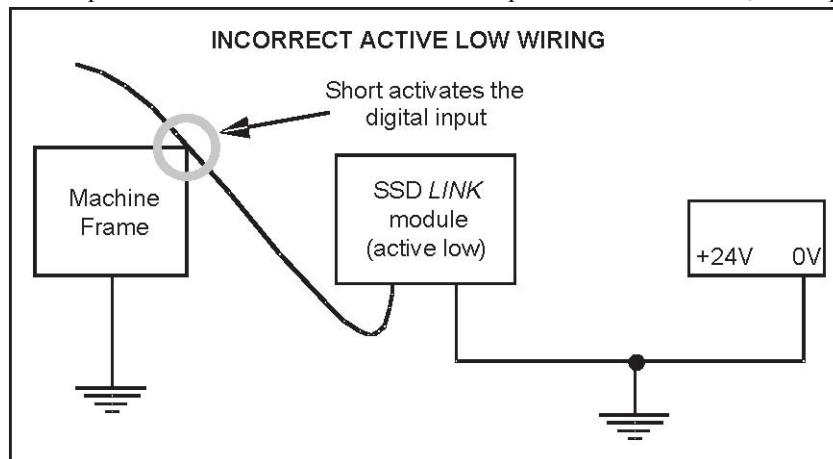


The drive setpoint must be disconnected from the normal line setpoint using the switch shown. Note the use of senders to ensure the start and setpoint are updated correctly. A ramp (not shown) may be required. Some logic will also be required to start and stop the drive in response to the digital input.

It is important the above digital module be prevented from shutting down during SSD *LINK* network failure conditions. This can be done by disabling the Restart On Network Fail option in the system control function block in the module configuration file. The drive should be configured to "Do nothing" when an SSD *LINK* network fail occurs.

Active Low Inputs and Safety

The L5202 digital I/O module operates as an active low device. If an input is connected to 0V, that input is activated.



It is common practice to ground the 0V of electrical equipment. With active low devices, such as the L5202, the input will be activated if a short occurs between an input cable and ground. This could be a safety hazard if, for example, that particular input was used to start a drive.

It is highly recommended that active high digital modules (such as the L5209) be used for digital inputs on safety critical applications. If active low modules are used, the supply should be isolated within the same system enclosure. The 0V terminal must not be connected to ground.

Safety and Senders

For safety critical areas of the configuration, senders must be used to ensure SSD *LINK* network data reaches the destination module. This is most important for logic data, which in general is sent once upon change. If, for example, a push button is pressed, a single message may be sent over the SSD *LINK* network. If this message is missed by the destination module, the push button operation will be ignored. Messages may also be missed due to partial failure of the SSD *LINK* network, such as where damage to a fiber optic cable occurs. See the function block data sheets for more information on senders.

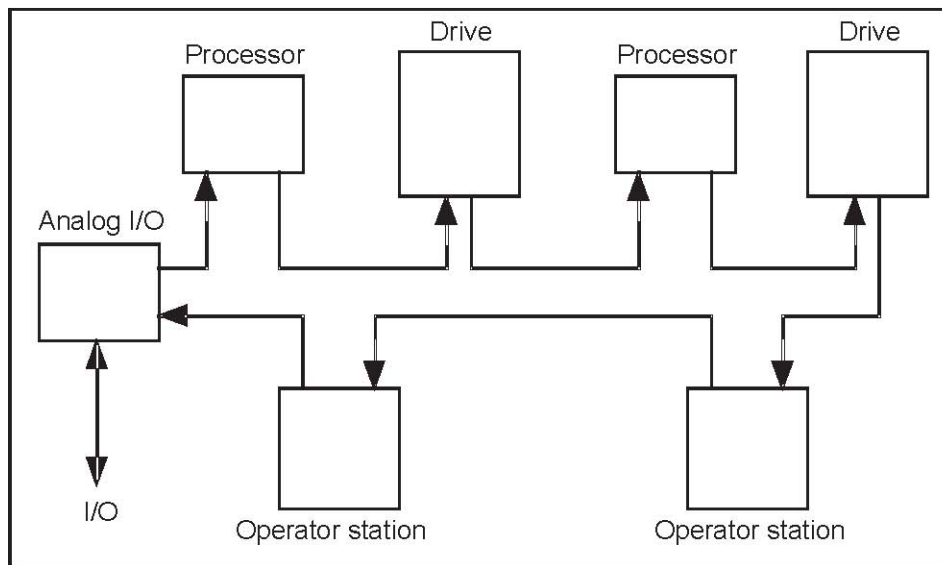
CUSTOM SYSTEMS

The following table highlights the way in which the number of SSD *LINK* modules should be initially planned for custom SSD *LINK* systems.

STEP ACTION

1. Specify number of drives;
2. Allow one L5207 processor module per L590 drive;
3. Add I/O modules (L5201 Analog, L5202 Digital, or L5209 Digital) to suit I/O requirements;
4. Add L5102 Operator Stations to suit operator I/O requirements;
5. Add one repeater module per remote enclosure, desk, or machine location;
6. Add one repeater for each 50m/165' of fiber optic run;
7. Add specialized modules (e.g., L5203 Serial I/O, L5210 PLC Gateway, or L5211 VME Interface) as required;
8. Add the engineering time.

The following diagram shows a typical custom configured system with the minimum number of modules specified using the above steps.



Including one processor per drive allows standard processor module configurations to be used for each drive section of the system. These standard configurations will cover all line section control and winding/unwinding functionality. This may seem to be a more costly approach, but it has proved cost effective by reducing engineering time.

OEM SYSTEMS

One reason to allocate fewer modules is when there is considerable repeat order business for identical systems. It might be worthwhile considering an *SSD LINK* system that is more value engineered with regard to hardware. With such a system, time is required to determine the minimum number of *SSD LINK* modules needed for the actual configuration.

This page intentionally left blank.

Chapter 6 Configuring SSD LINK

Each SSD *LINK* system consists of modules connected by a high speed fiber optic network. The SSD *LINK* system functionality is, in general, determined by function blocks contained within the module configurations. An SSD *LINK* function block is a highly defined software object with the following properties:

- Data inputs;
- Data outputs;
- State (data);
- Behavior (functions), as designed by the system engineer.

Each of these properties is defined in the applicable function block data sheet. Function blocks may be combined in a configuration up to the limit of the target module's memory capacity.

In general, SSD *LINK* function blocks execute only when messages are sent to their inputs. When a message is sent, this activity is known as an *event*. Since actions take place only when events occur, the SSD *LINK* system is said to be *event-driven*.

When an SSD *LINK* function block is executed, output results are generated and passed to other connected function blocks. The connection may be local or over the SSD *LINK* network. When one function block sends data to another, a *message* is sent which may invoke execution of the receiving function block; whether it does or not depends on the characteristics of the input to which it is sent.

It is common for many function blocks to send messages only when there is a physical action on the configurable inputs (e.g., pressing a push button or changing a setpoint value from an operator station potentiometer). The advantage is that processing capacity is not wasted when data is not changing, and network traffic is reduced since messages are not sent if data variables are constant.

Many function blocks with multiple inputs will only generate an output when a specific input receives a message. This input is referred to as the *trigger* input, since sending a message to it triggers the generation of an output message (see Section 6.4.2).

There are certain specialized function blocks which send messages continuously at a given rate. These are used when it is important to update the input of a function block continuously (e.g., for feedback variables of closed loop control systems).

Two SSD Drives computer software tools, ConfigEd and SAM, are used to respectively configure and commission SSD *LINK* systems. To configure SSD *LINK* modules, function blocks are placed into configurations in a CAD-like environment on the computer running ConfigEd. The blocks are then connected, both to other function blocks in the same module and to other module configurations via the SSD *LINK* network. Refer to the appropriate manuals for more information on these tools.

Function blocks (both new and revised versions) are generated by SSD Drives, Inc. Each function block has a number within the RG35XXXX series, and is under SSD Drives engineering revision control. Refer to the Function Block Data Sheets for information on the interconnect ability and use of the function blocks.

The SSD *LINK* function blocks are found within the SSD *LINK* database. This database must be present to configure SSD *LINK* systems. Not all function blocks may be used in all module types; this is especially true with SSD *LINK* drive modules. Refer to specific SSD *LINK* product manuals for more details.

CONFIGURING SSD LINK MODULES

When engineering with SSD *LINK*, a module-level rough sketch of what function will be where should be done before starting to load configurations from ConfigEd. These provide a ready visual indication as to the function of the SSD *LINK* configuration. The goal is to assign functionality (groups of function blocks) to specific modules as accurately as possible before starting individual, detailed configurations. The following section gives general guidelines on how the SSD *LINK* modules should be configured into a working system.

Function Block Data Sheets

Data sheets for the currently available function blocks within the database are included with your purchase of SAM or ConfigEd, and should be readily available to all design engineers. The library is split into various categories to allow

function blocks to be grouped either by similar functionality or application. Those categories may change as the database is updated.

Application-Specific Function Blocks

A number of application-specific function blocks are available, including:

- Master Ramp;
- Drive Section Control;
- Speed Programmed Winder (SPW);
- Current (torque) Programmed Winder (CPW).

Application-specific blocks have the following advantages:

- They implement very complex functions at high speeds and with minimal memory requirement (thus saving module memory);
- They are already tested and documented;
- They offer greater performance than discrete function blocks could;
- They allow standard solutions that are easy to understand and implement.

It is highly recommended these standard application-specific blocks are used to implement applications. If your particular application requires additional function blocks not currently available in the library, contact SSD Drives. The object-oriented software allows the rapid addition of new function blocks to meet most customer needs.

General Details of Function Blocks

In general, a function block has three types of inputs/outputs:

- Inputs, which accept data from other function blocks;
- Outputs, which send data to other function blocks;
- Parameters, which are preset data not modifiable during execution of the module.

Most inputs can also be configured to have a preset state. That data will be used until it is changed by a message being sent to that input. Thus, it is possible to use these inputs as parameters. Parameters can be set with ConfigEd; SAM may be used to set all inputs and get the current state of most inputs and outputs.

Persistent Inputs

A persistent item is any state that is retained in the event of a loss of power. Like any other value, input, or parameter, they can be set by SAM, ConfigEd, or another function block output.

Persistent items set by SAM will not be lost when the module is powered down. This can be quite useful, since the last value (or state) of the persistent input will be restored when power returns. Most common inputs to function blocks are persistent, and these are usually specified in the data sheets.

Organizing Configurations

The following sections provide some recommendations that should be followed when organizing configurations for an SSD *LINK* system.

Choosing Function Blocks

When a new hardware module is specified in ConfigEd, a configuration appears on the computer screen, containing application-specific and common software blocks.

To set up the configuration, function blocks are placed into the on-screen representation, and the function blocks are then connected, both within the module and over the SSD *LINK* network, through the graphical interface of ConfigEd.

Two names are used to reference each function block:

- Type name - Name as defined in the function block data sheets (e.g., Annunciator)
- Name - User-defined name, referring to a specific block in that configuration (e.g., Status).

Some function blocks are module-specific and may not be used in every type of module. Each function block type has a data sheet describing what it does and how to use it. Please read and understand the data sheets before attempting to configure function blocks into a system.

Duplication of Components

SSD *LINK* is a very flexible, configurable drive control system. As with any flexible system, steps can be taken to reduce the amount of engineering time used to configure the system by standardization.

Many line drive systems consist of a number of identical control sections (e.g., in-line load cell controlled nip rolls). To save time, design one of these sections completely, then duplicate it for the additional control sections. This approach can be taken further by designing standard module configurations and modifying them to suit the current project.

It is possible, in some cases, for jobs to be duplicated in their entirety and used as templates for later jobs. Theoretically, this approach could eliminate virtually all engineering time associated with configuring additional drive control systems.

Standard Function Blocks

The use of standard application-specific function blocks can reduce engineering time considerably. For example, implementing a closed-loop, load cell controlled center wind control is simply a matter of using a single function block.

There are a number of standard function blocks available, including:

- CPW - Open-loop center winder control;

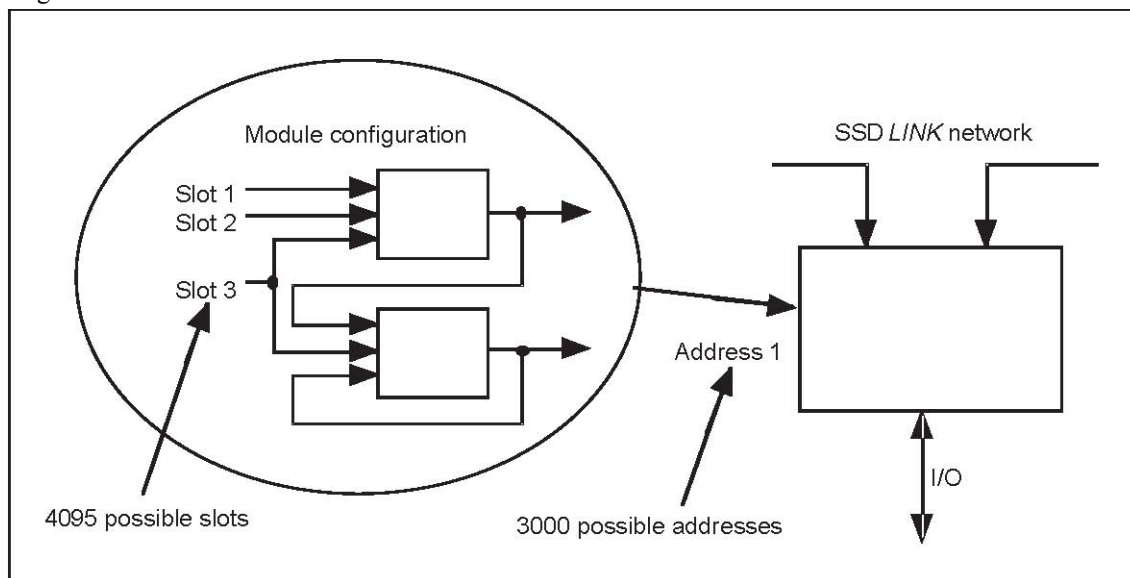
- SPW - Closed-loop center winder control;
- Section control - Draw/ratio control with closed-loop trim;
- Master Ramp - Implements an S Ramp function with start/stop and status logic.

It is recommended these application-specific function blocks be allocated to separate L5207 processor modules to give the best performance and to reduce engineering time.

Network Addresses and Slots

The SSD *LINK* system uses numeric addresses to allow data to be passed between modules on the network. At configuration time in ConfigEd, each hardware module is assigned an address from 1-3000. In addition, each module can have up to 4095 SSD *LINK* input slots (numbered from 1-4095), specified using ConfigEd.

A slot can be thought of as a software terminal configured within a hardware module. This allows data to be sent to other modules on the network by specifying the address and slot number of the destination module that is to receive the data. See the following diagram.



Note that using small ordinals (those from 0 to 255) for slot and module addresses will make configurations more compact and thus allow more functionality to fit in each module.

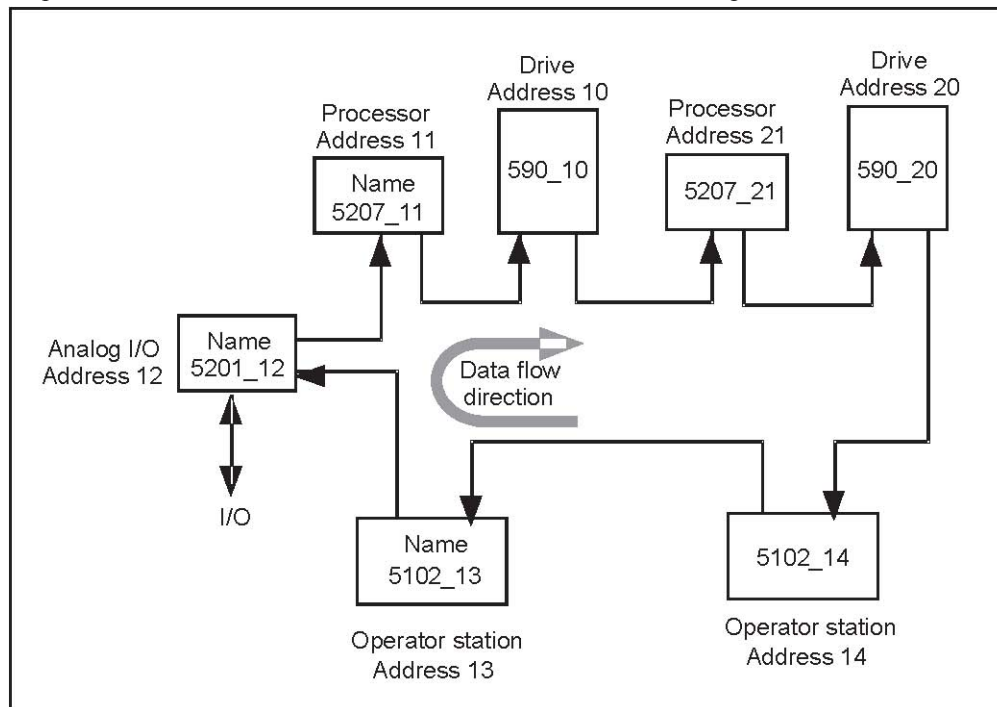
Standard Addressing

It is best to adopt a standard method of assigning addresses to the modules on your network. This simplifies the task of grouping modules by function within your system and easily keeping track of them.

For example, grouping of a drive and its associated modules (processor, operator station, I/O, etc.) can be accomplished by giving the drive an address of 10 and then assigning addresses of between 11 and 19 to the other modules associated with that drive. The next drive can be given an address of 20 and its associated components addresses of between 21 and 29. This pattern can be repeated throughout your system. Remember that using smaller address numbers (those <256) saves configuration space.

Standard Module Names

Each SSD *LINK* configuration must be given a valid DOS name of eight or fewer characters. One example of a naming protocol is using the module type number followed by an underscore followed by the module address. For example, 5207_11 would be an L5207 processor module with a module address of 11. Additional examples are shown in the following diagram.



Other module naming schemes can be developed to suit the individual needs of the user (e.g., using the application section name).

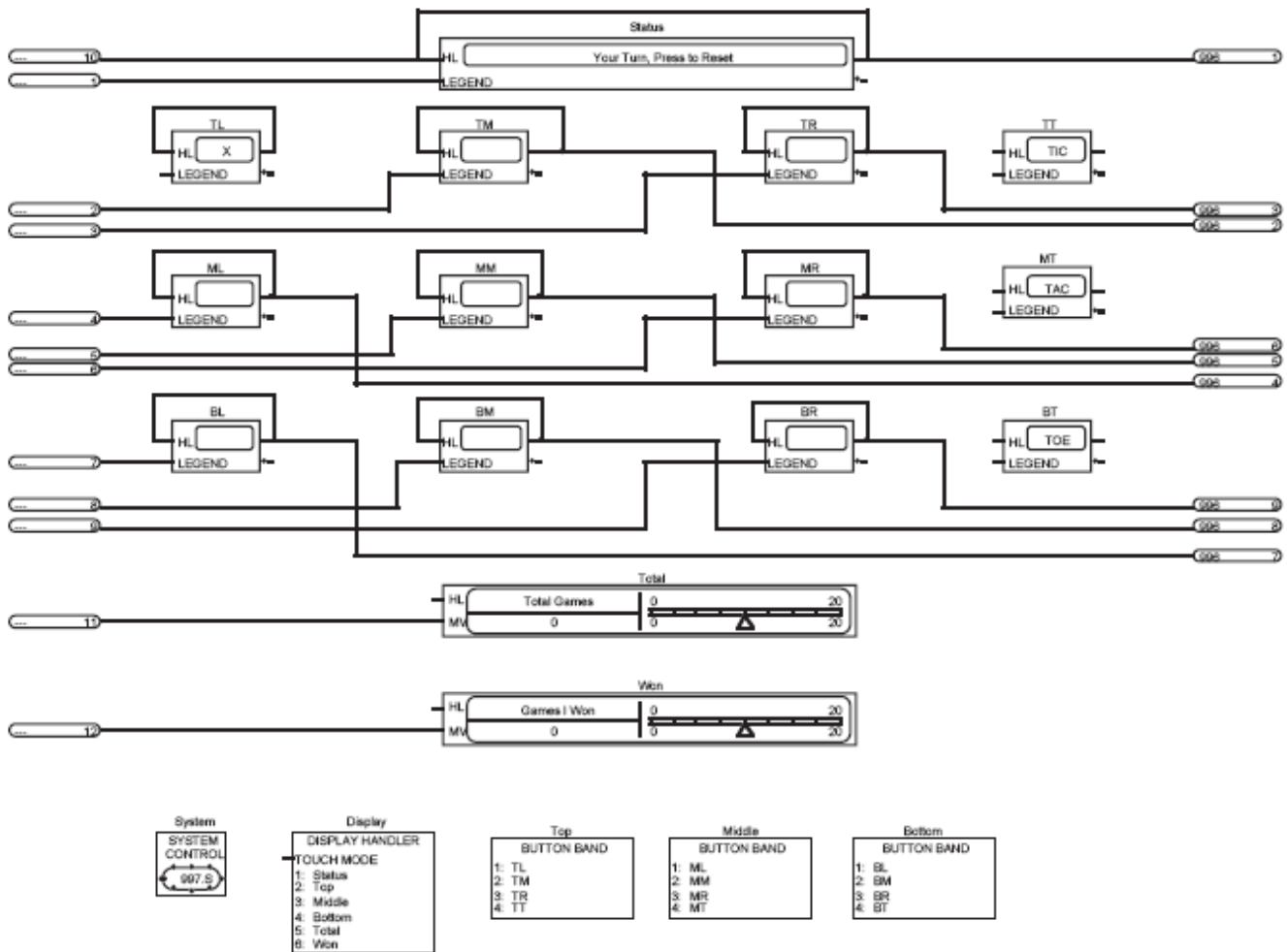
Configuration Drawings

Configuration drawings are produced through the graphical user interface of ConfigEd. All the information to allow the system to be configured must reside in the drawings. As with most engineering activities, the completeness and accuracy of the configuration drawings will aid smooth completion of the total project.

ConfigEd incorporates a simple CAD package, where the function block diagram is actually drawn on the computer screen. When completed, configuration files are installed into the SSD *LINK* modules. The screen drawings can also be printed as a part of the final system documentation.

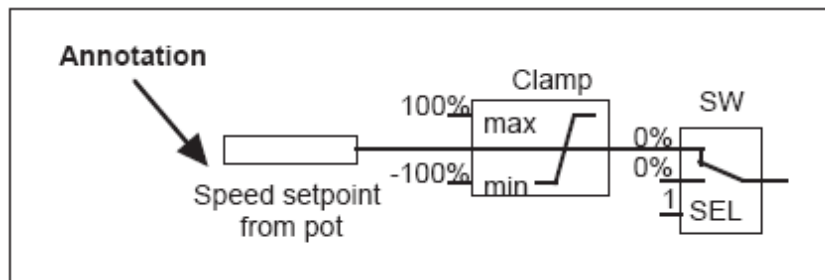
ConfigEd therefore removes the manual documentation step in the design process, dramatically reducing the chance of configuration errors due to mismatch between the block diagram documentation and the module configuration. Refer to the user manual for more detail about designing configurations with ConfigEd.

The following diagram shows a module configuration block diagram. This example is for a Tic Tac Toe display for an operator station, which is a training exercise included in the ConfigEd package.

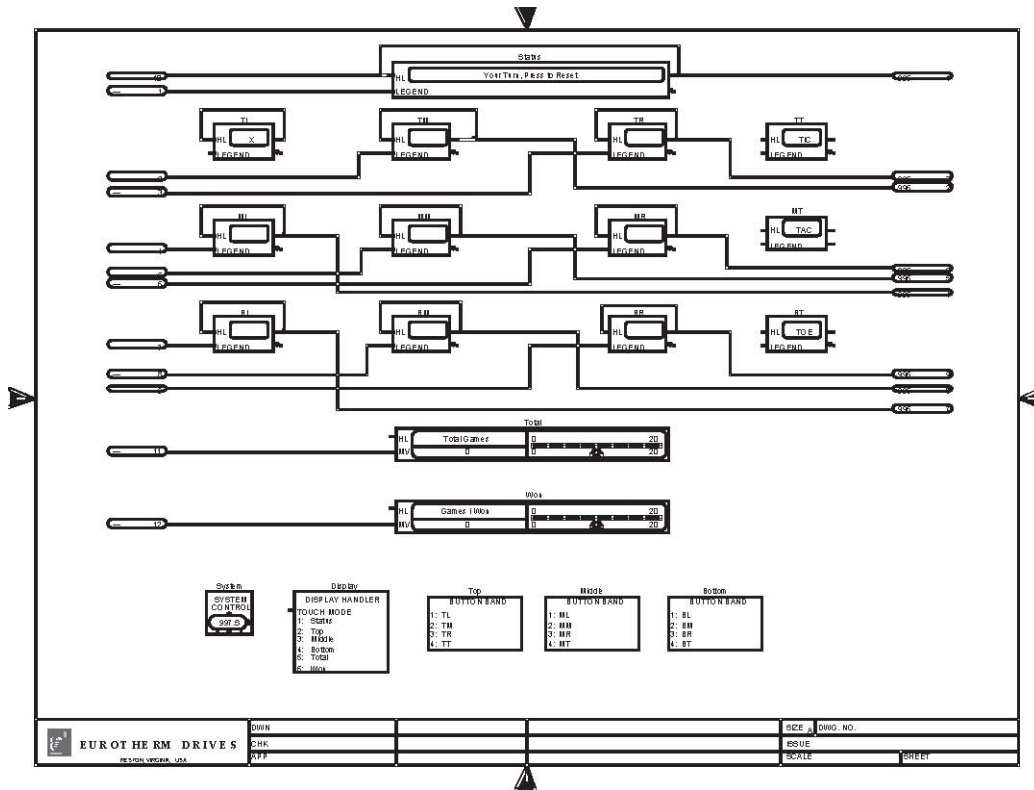


Various signal inputs and outputs on the block diagram are shown. It is recommended that the general signal flow should be from left to right, with the SSD *LINK* network inputs and hardware inputs on the left of the drawing and the SSD *LINK* network outputs and hardware outputs on the right.

SSD *LINK* can be used to create complex sequencing and control systems. It is good practice to annotate the function block diagrams with text and even drawings; ConfigEd provides the tools needed to annotate these diagrams. Annotation is especially important if the project is to be commissioned and serviced by a third party engineer. Complex sequencing should always be annotated to indicate the significance of a particular logic state. See the example below.

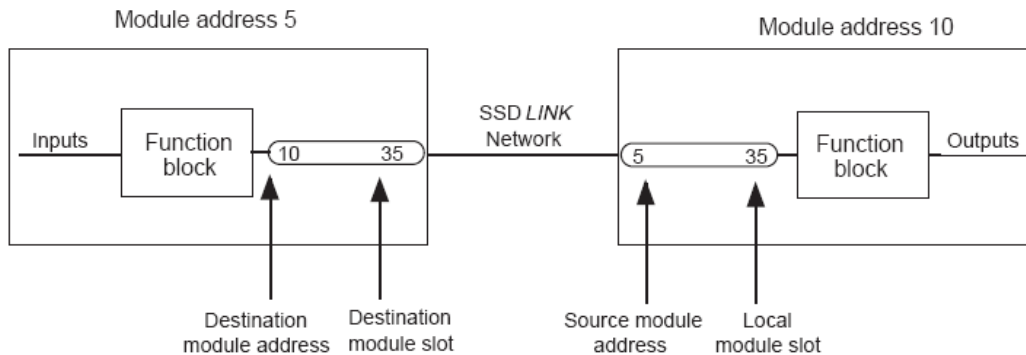


It is highly recommended that you include a form with your configurations, as shown in the following diagram.



This allows important information (such as the revision number) to be present on the printed drawings, making it easier to trace a project as part of a quality system. Refer to the ConfigEd manual for more information on forms.

SSD *LINK* function block diagrams contain outputs and inputs where signals are routed to and from other SSD *LINK* modules via the network. The following diagram shows the SSD *LINK* input and output connectors.



Source and destination SSD *LINK* network connectors are shown. At module address 5, data is being passed to module address 10, local slot 35. At module address 10, local slot 35 must be specified correctly. The source module address is optional, and is used to act as a cross reference for the source of the data. ConfigEd can fill in the source module addresses automatically at print time and it is recommended this be done.

SSD LINK DATA TYPES

Decimal numbers are used for all SSD *LINK* operations, and the three main types of numbers within the system are ordinals, logic, and values.

Ordinals

Ordinals are whole, unsigned integers used for counters and enumeration. The ordinal number range is at least 0 ~ 65535, but may be larger for some module types; refer to the module data sheets for details.

Logic

These are used for all logic operations. The logic range is:

- 0 = Logic 0 (False)
- 1 = Logic 1 (True)

Note: Logic 0 = Ordinal 0 and Logic 1 = Ordinal 1

This equality between logic and ordinals is very important when configuring SSD *LINK* systems. It is possible to connect logic gates which output logics to allow selection of operator station highlights which accept ordinals.

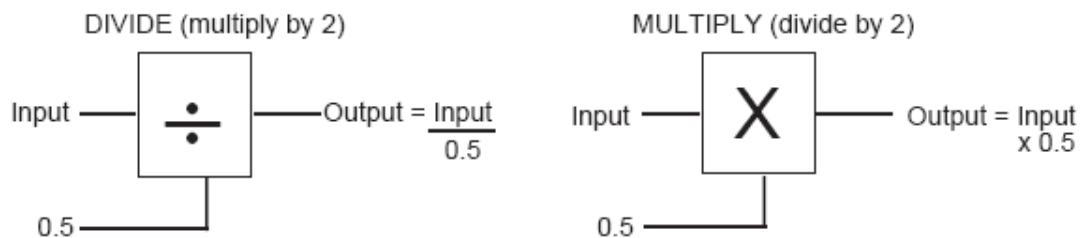
Values

These are used for all signal processing. The SSD *LINK* number range for values is

± 1.00000 . Here:

- $+1.00000 = +100.000\%$
- $-1.00000 = -100.000\%$

The number resolution at present is 0.00003, which is equivalent to 0.003%. Since all values are normalized to unity, to multiply a value by 2 you must divide it by 0.5. Similarly, if a value has to be divided by 2, this has to be done by multiplying by 0.5.



In general (and henceforth in this manual), values are represented by percentages. For example:

- 0.5 is represented by 50%
- -0.83 is represented by -83%.

Values are Two's Complement

SSD *LINK* uses 16 bit numbers internally to represent numbers. Regarding values, the 16 bit number is used to cover the full range of +1 to ≈ 1 . The two's complement system is used to represent signed values:

- $+1 = 32767$ (7FFF Hex);
- $0 = 0$;
- $-1 = 32769$ (8001 Hex).

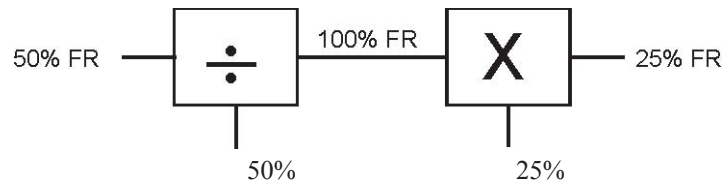
The number 32768 (8000H) is currently an invalid value within SSD *LINK*. If this number is encountered as a value, the SSD *LINK* module could generate an L Error. This could happen if ordinal numbers are directly used as values. If ordinal numbers must be used as values, there are ordinal-to-value conversion function blocks available for this purpose.

Accuracy and Resolution with Values

SSD *LINK* function blocks can be used to perform many signal processing functions. In order to maintain good accuracy, it is important to keep values as close to full scale as possible as a calculation proceeds.

The following diagram shows the correct way to implement a multiply and divide function using discrete function blocks (FR refers to full range).

Best Accuracy



If possible, the divide should always be performed before the multiply in order to maximize accuracy, but only if the result of the divide cannot result in a value greater than 100% (which would saturate to 100% and cause inaccurate results). See Section 6.2.4.3 for more information.

Over-Range Allowance

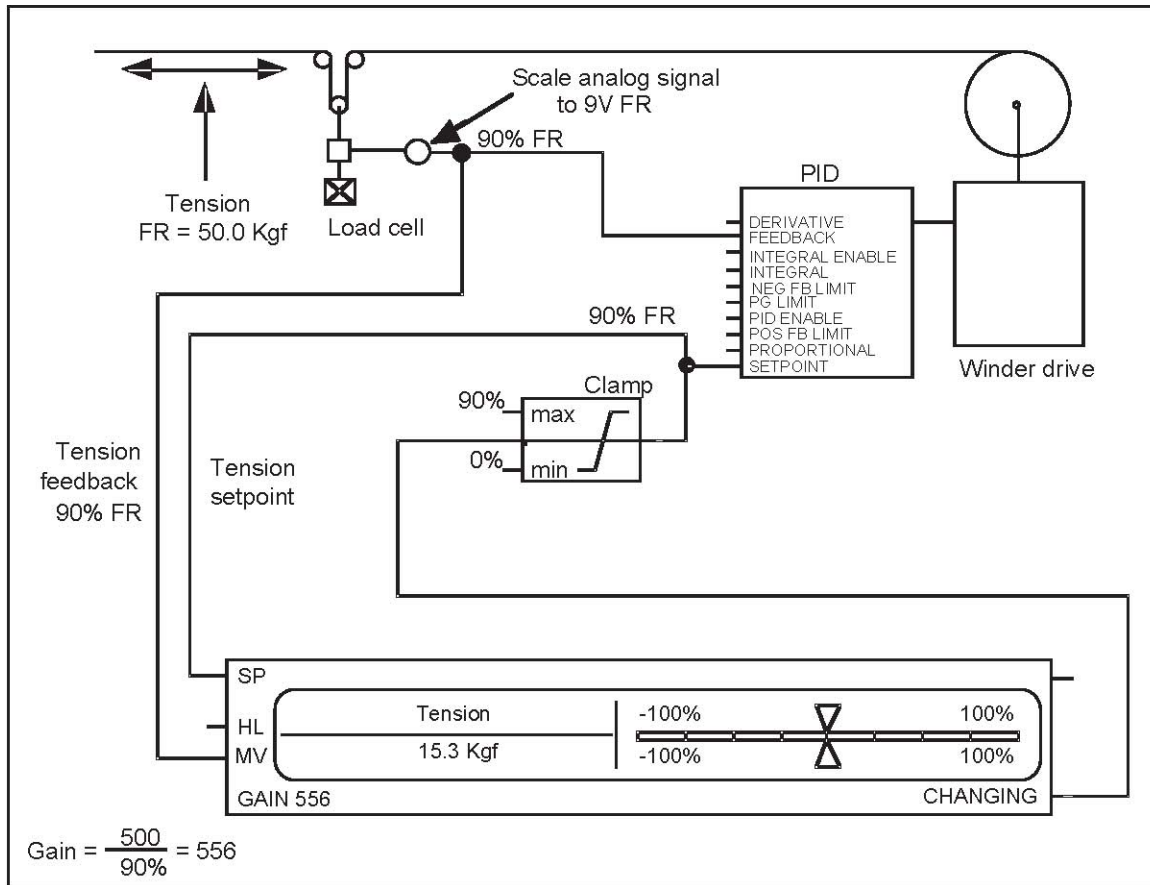
There is no over-range built into the SSD *LINK* value number. However, some form of over-range is needed to allow feedback control of a variable when it is operating at full range. Therefore, over-range must be built into the number range when the system is configured. For example, in the L590 drive:

+100% current = 0.50000 (or 50% SSD *LINK* variable)

This means that under normal steady state conditions, the maximum SSD *LINK* variable observed is 0.5 or 50%. There is, however, an over-range of 50% to currents up to 200%.

Closed Loop Control Over-Range

The following diagram shows a load cell tension control loop using a PID operating at 90% full scale set points. The feedback is scaled to 9.0V full scale before entry into the analog input of an L5201 module.



The PID operates with a setpoint from a Pot+Meter function block clamped at 90% full range. Scaling the analog input at 10V full scale and reducing it to 90% in software would not give any over-range, since the analog input *SSD LINK* value would saturate at 1.00000.

The feedbacks for the Pot+Meter function should be scaled to take into account:

- The 90% full scale operating point;
- The actual machine number to be displayed on the potentiometer (e.g., 50.0Kg/110 lbs.).

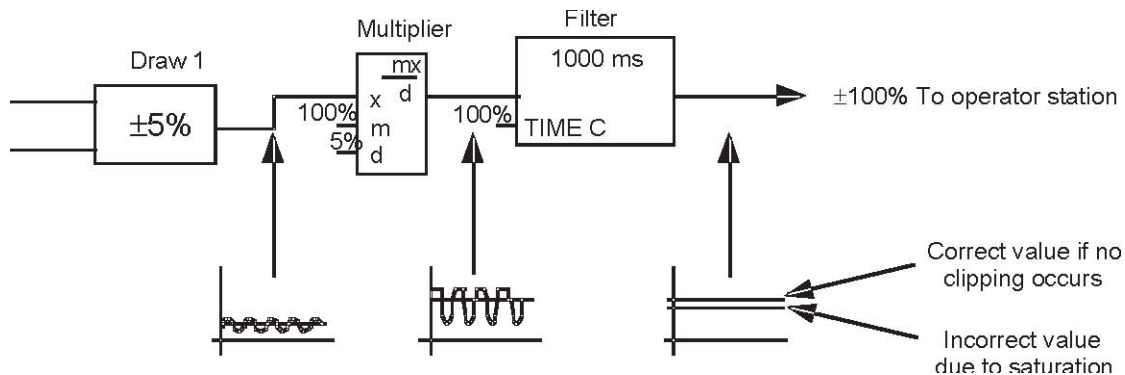
The winder tension loop shown in the above diagram is an example only; application-specific function blocks should always be used to implement winders.

Saturation and Filtering

With any industrial control equipment, it is important that internal numbers produced during computation do not saturate. The maximum possible *SSD LINK* value range is ± 1 or $\pm 100\%$; any computations producing results outside this number range will be clamped at $\pm 100\%$.

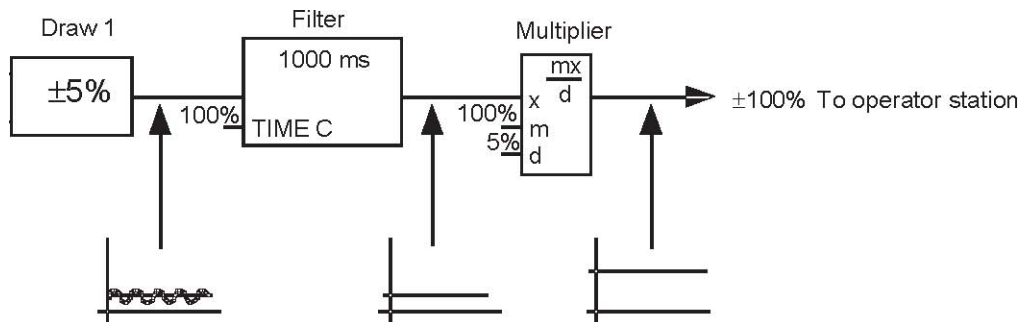
Even if the average of a value is less than the saturation limit, it is possible for ripple components to cause saturation.

The diagram below shows the effect of saturation on an *SSD LINK* value. The function blocks are used to provide a $\pm 100\%$ draw feedback to any operator station. The actual draw range is $\pm 5\%$, and therefore this must be multiplied by 20 to achieve the correct range to drive the operator station display.



The actual steady state draw on the output of Draw 1 is within the specified range, but the draw may exceed this range occasionally. This gives rise to considerable ripple on the actual draw feedback. Filtering is required in this application in order to give an accurate steady state draw indication.

The position of the filter is important to prevent saturation of the intermediate value. If the filter is positioned at the end of the function block chain (as above), the intermediate variable will saturate due to the ripple, causing the filtered value to indicate an incorrect value lower than actual. As shown in the diagram below, the solution is to filter the raw draw before the multiplication.



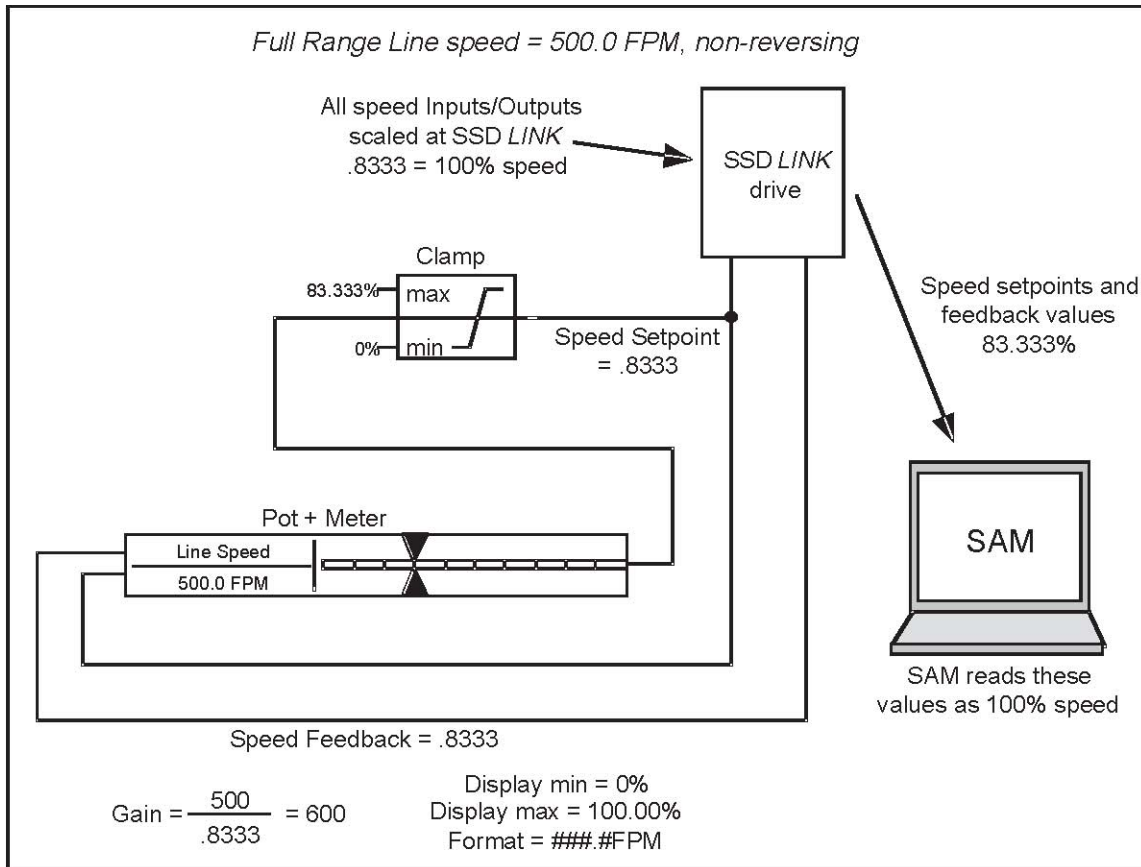
Since the raw draw value is filtered before the multiplication, the actual value to be sent to the operator station display will not saturate.

Over-Range and the *SSD LINK* DC Drive

The *SSD LINK* DC drive has over-range built into the speed demand inputs and outputs.

These I/O are scaled:

- +100% speed = SSD *LINK* variable 0.8333;
- -100% speed = SSD *LINK* variable -0.8333;
- Conversely, ± 1.0000 in SSD *LINK* = $\pm 120\%$ in the drive. Internally, the drive speed demand is clamped to a maximum of $\pm 105\%$. This would relate to an SSD *LINK* variable of .875. The following diagram shows a drive with a speed setpoint supplied from a potentiometer function block.



The setpoint from the potentiometer function block is clamped at 83.333% full range. This yields a motor speed of 100%. Once again, the feedback for the potentiometer function should be scaled to take into account the 83.333% full range operating point.

The actual machine number is displayed on the potentiometer (e.g., 500FPM). The Pot+Meter scaling functionality will be described later. The output of some application-specific function blocks (such as the Master Ramp) is already scaled at 83.333% full range. These are intended to be used as speed demand variables for drives. SAM automatically displays these variables normalized to 100%.

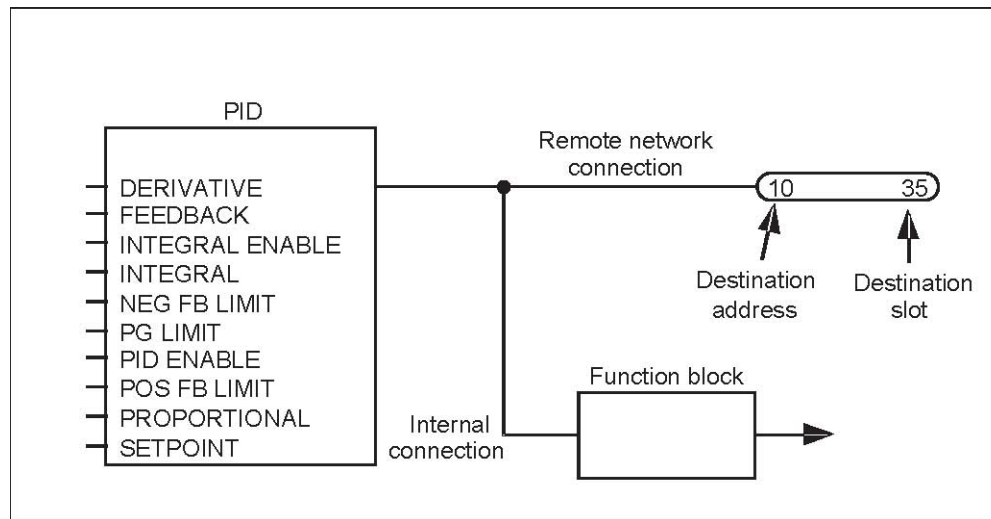
SAM and SSD LINK Numbers

When using SAM to access parameters in an SSD *LINK* system, those parameters will be presented scaled to their full range value. For most values, 1.0 = 100%, but for some parameters this may not be the case (e.g., Drive Current 1.0 = 200%, Drive Speed 1.0 = 120%, etc.).

COMMUNICATION BETWEEN FUNCTION BLOCKS

Function blocks are connected using ConfigEd to form the total module configuration. Blocks in different modules are also connected together over the SSD *LINK* network. To specify connections, the outputs of function blocks are always connected to specified inputs. In ConfigEd, input connections are always drawn from source to destination.

The following diagram shows typical function block connections, both within the local module and over the network.

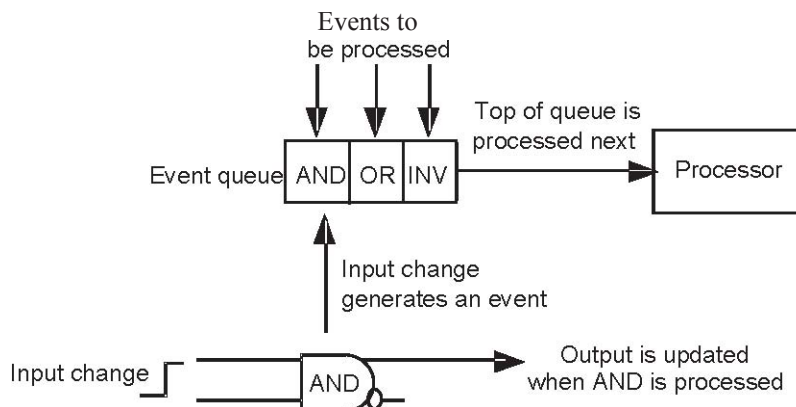


EVENT PROCESSING

SSD *LINK* is an event-driven system, not a scan-based system as found in PLCs.

Event Queue Description

The microprocessor within each module performs one task at a time. Therefore, an event queue is used to allow events to be stored pending processing. The following diagram shows three events lined up on the queue.



The module devotes all unused RAM to the event queue, a first-in first-out (FIFO) stack. Overtaxing a module causes the event queue to exhaust all of this free memory and cause an L Error (Out of Heap).

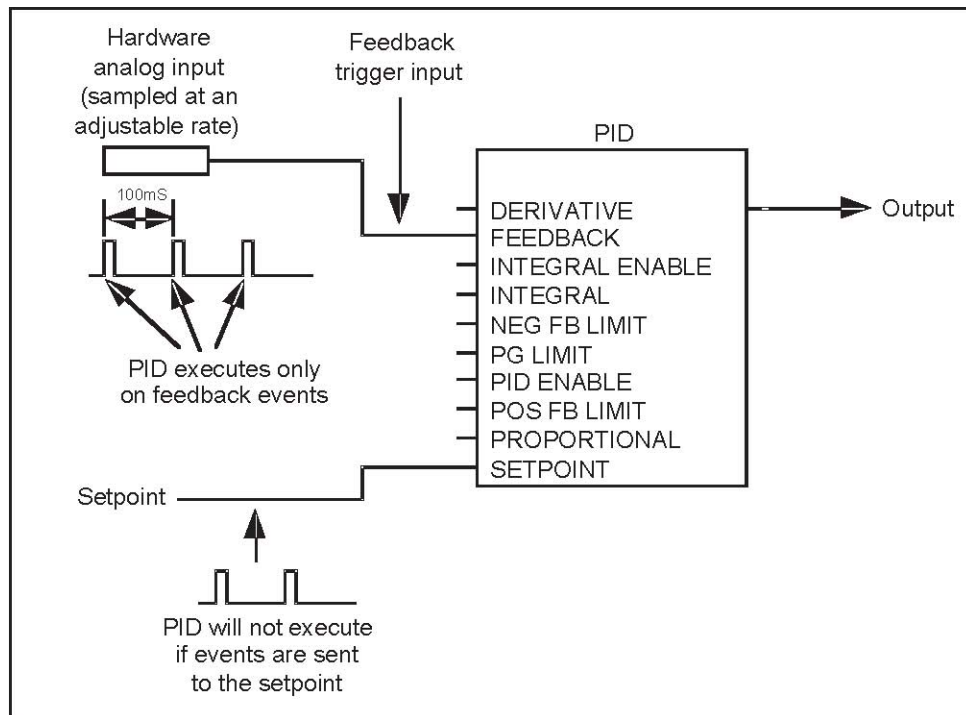
When an event occurs, the following actions take place:

- The event is posted to the event queue, generally to the back of the queue;
- When the event reaches the front of the queue, it is removed and the relevant software block is processed to produce its outputs;
- If the outputs are connected to other function block inputs, as is normally the case, this may create other events.

Trigger Inputs

below, an analog input Each function block has one or more trigger inputs, special inputs which allow the function block to be executed when messages are received. If no messages are received at the trigger input, the function block will not execute. It is important to connect trigger inputs correctly.

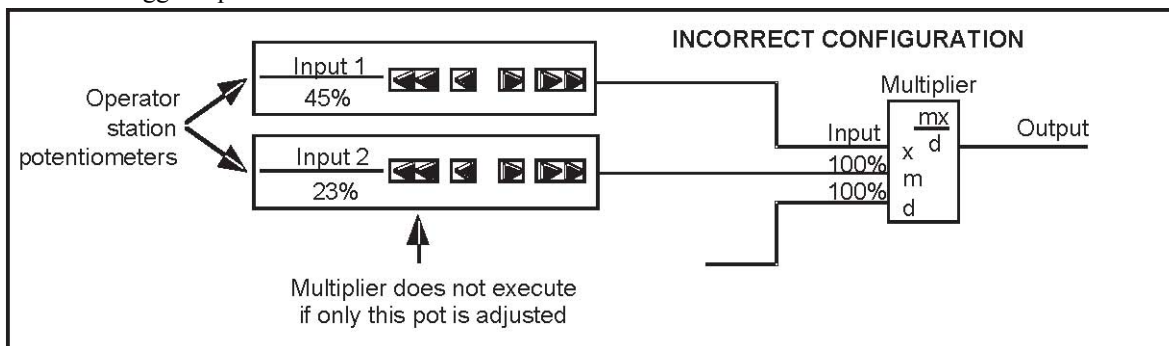
A PID function block with a trigger input is shown below. It is common for the feedback value to be the trigger input. Any messages on the feedback cause an event to be generated and execution of the PID. In the example feeds the trigger input with messages at a constant pre-set rate of 100mS.



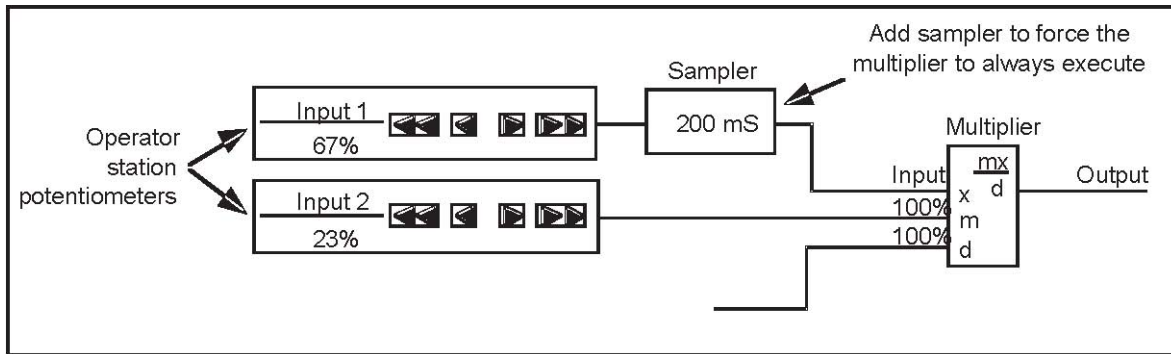
In general, trigger inputs of function blocks are specified in the data sheets. If no trigger inputs are specified, it is assumed that all of the inputs are trigger inputs. In ConfigEd, application-specific blocks, such as the speed profiled winder (SPW) and current profiled winder (CPW), have their trigger inputs labeled T. See function block data sheets for details.

Forcing Function Blocks to Trigger

Sampler function blocks can be used to force other function blocks to trigger in a certain way. The following diagram shows two operator station potentiometers connected to a multiplier. If input 2 is adjusted, the multiplier output will not be updated since the multiplier *input* (X) is the trigger input. The multiplier output will, however, be updated if input 1 is adjusted, since it is connected to the trigger input.

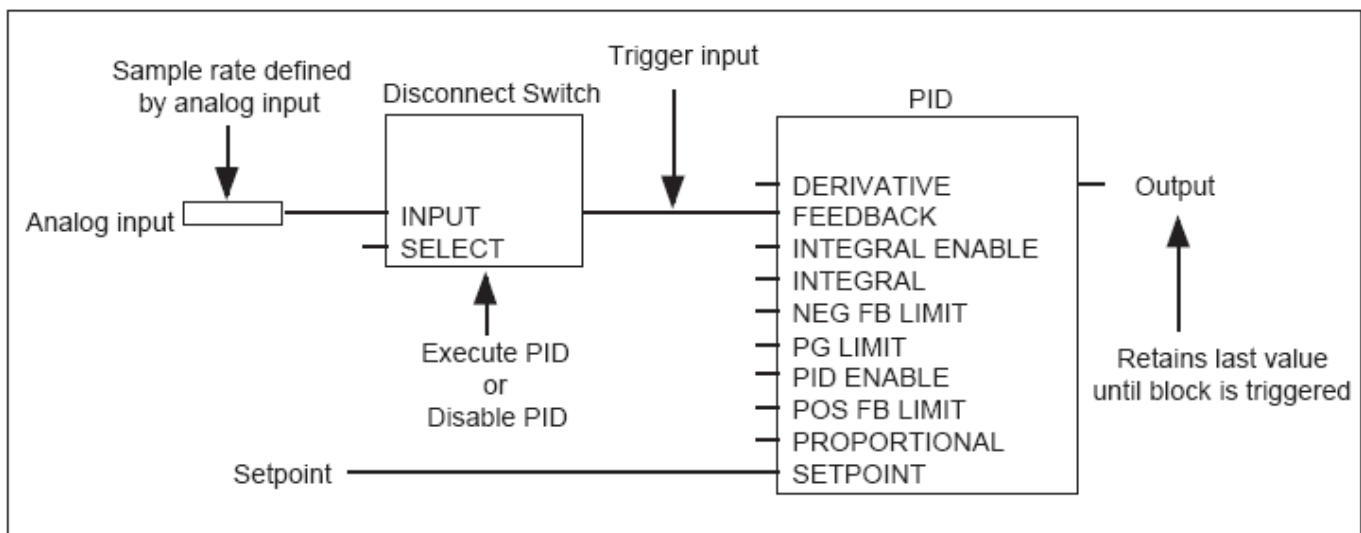


The following diagram shows a sampler connected to the multiplier input. This will trigger the multiplier and update the output on a regular basis.



Disconnecting Trigger Inputs

Under certain circumstances, it may be desirable to prevent function blocks from executing. A special switch function block allows events to be prevented from triggering function blocks, as shown in the following diagram.



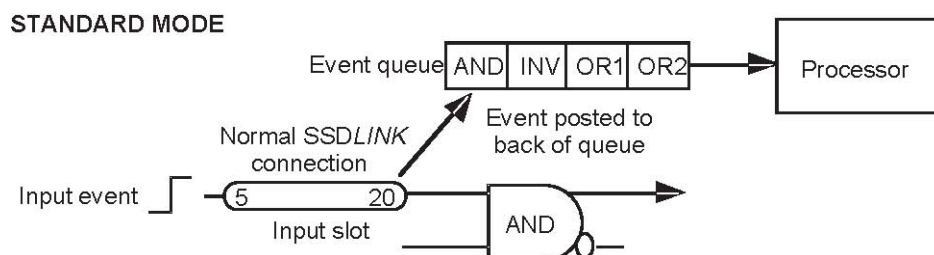
When this switch is closed, input events are passed to the output. When open, no events can pass through to the PID trigger input. This allows the trigger event chain to be broken. The switch is opened and closed using a logic input. A typical application is one in which it is required to prevent execution of a complex function block chain. This could save large amounts of processing time.

SSD LINK Inputs to a Module Configuration

When a module is required to receive data over the SSD LINK network from other modules, input slots are required to receive this data. The input slots are specified with a slot number (unique within each module's configuration) during configuration. There are two types of input slots which can be specified from ConfigEd: normal and priority.

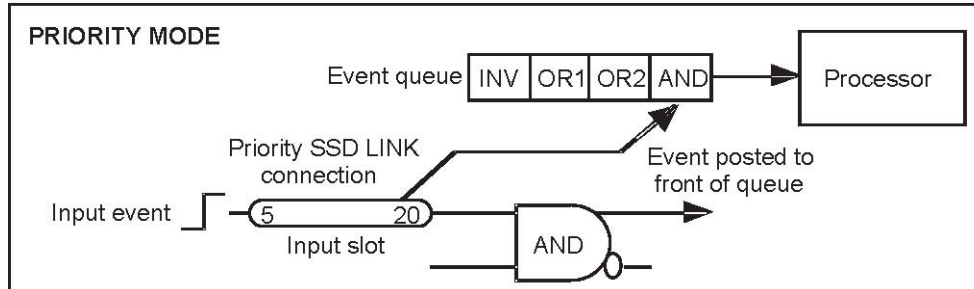
Normal Input Slots

Normal input slots are the standard input slots. They should be used for almost all data inputs from the SSD LINK network to the configuration. When a message is received to a normal input slot, an event is generated which is placed on the event queue in the manner described in earlier sections.



Priority Input Slots

Priority input slots are used for special, time-critical applications. They are typically used when functions are distributed across modules but must respond as though they were located in the same module. When a message is received to a priority input slot, an event is generated which is placed at the front of the event queue. This forces the message to be executed very quickly. Other events already in the event queue are delayed by the priority message.



Restricting the number of priority inputs is wise since the larger the number of priority inputs, the less “priority” each will be given. Using many priority inputs could also slow significantly the processing of the normal inputs.

Timebase Function Blocks

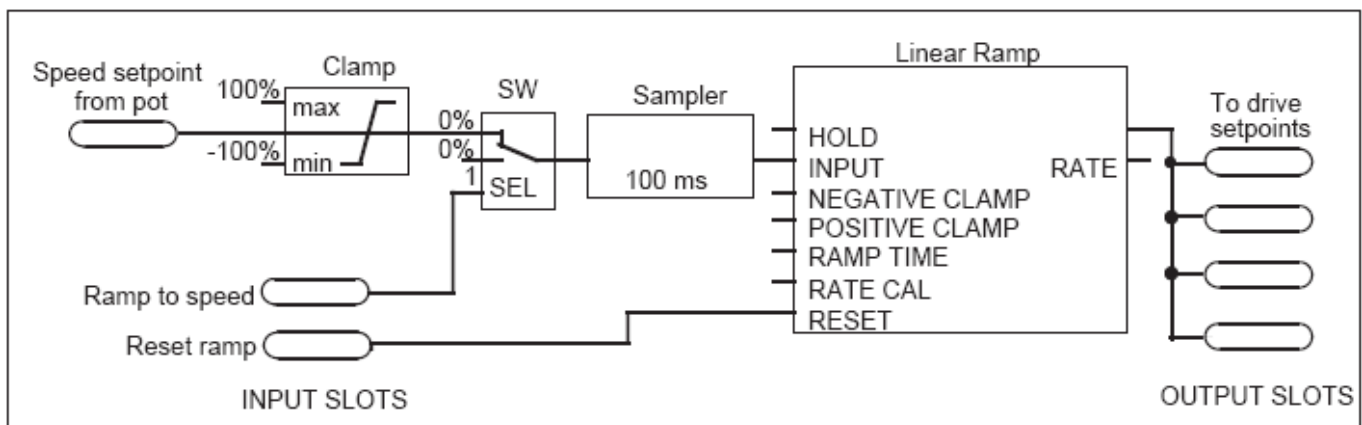
In *SSD LINK*, certain specialized function blocks execute and send data at a constant rate. They are generally analog processing functions, where a group of connected blocks must execute continuously. This would be the case for feedback variables for closed loop control.

With these blocks, it is often possible to specify in ConfigEd the timebase or the rate at which data is sent from the output. Blocks which implement a timebase are usually specified in the data sheets. Examples of these blocks include:

- Analog inputs of L5201 module.
- Drive parameters
- Samplers
- Master Ramp
- Senders
- S Ramp
- Push button ramp.

Refer to the *SSD LINK* function block data sheets for details of these function blocks. In general, there must always be a timebase function block to allow analog function blocks to trigger continuously. This is often an analog input.

Certain function blocks (e.g., the simple filter) must be clocked at a fixed rate. These blocks are indicated in the function block data sheets.



This diagram shows a typical line setpoint generator for one or more drives and associated configurations. A 100mS sampler ensures the linear ramp is triggered on a regular timebase. Any function blocks in this or another module supplied from the ramp output will now be triggered if the connection is to their trigger input(s).

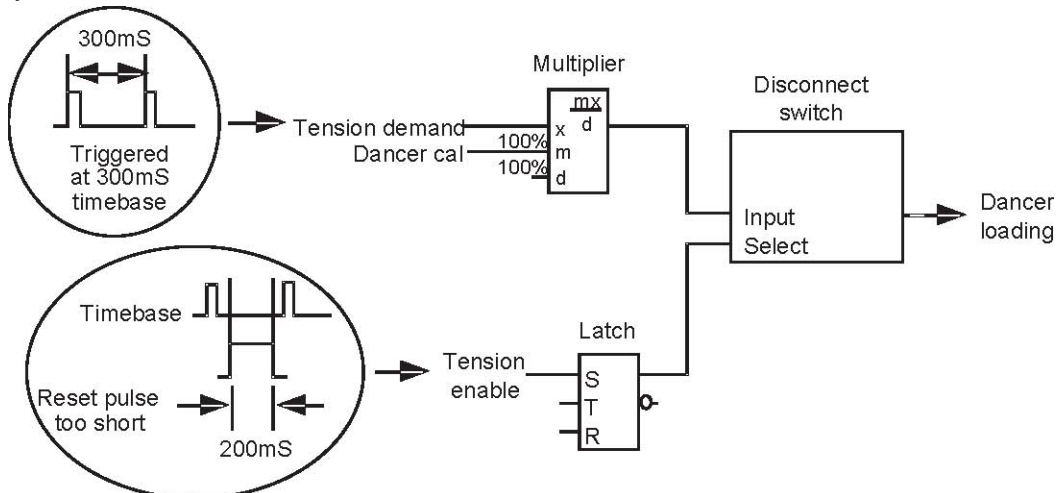
Multiple Timebase Function Blocks

Complex software blocks may have a number of internal timebase parameters triggering at different rates (as specified on the data sheets).

Multiple timebases can help yield high overall performance for critical areas of application-specific function blocks. In the complex winding blocks, for example, the critical performance area is the closed loop control and inertia compensation functions. All diameter calculation and tension demand processing is performed at a slower timebase rate. This allows more processor time to be allocated to servicing the feedback loop at a fast sample rate.

The timebase parameters may be set to quite long values. Within SPW, for example, the diameter calculation is performed at a default of 300mS intervals. If non-trigger variables are sent to blocks which operate on a timebase, the data must be applied to the inputs for *at least* the triggering timebase length. If this is not done, the data may not be sampled. Sending them more often is wasteful and does not improve performance.

The following diagram shows a simplified part of the complex winder function blocks. The tension enable input is shown as a short pulse applied for a time shorter than the function block timebase. In this example, the tension enable input will not be sampled reliably.



Event Queue Overflow

A module's CPU has a finite processing capacity, meaning there is a maximum number of events that can be processed per second. If this rate is exceeded, the event queue may exhaust its free memory pool and the module will generate an L Error.

This problem is more likely to occur with the L5101 or L5102 operator stations due to the high display processing overhead in these modules. Therefore, operator station modules should be used primarily for I/O procedures, with processing being moved to other modules.

If the module generates an L Error and the event queue has no memory available, inspection of Get Info in SAM or ConfigEd will show Out of Heap. See the SAM or ConfigEd manual for details on Get Info and its messages.

If the event queue runs out of memory, the solution is to reduce the event frequency. This can be done by increasing the timebase parameter of any timebase function blocks. The sampler function block can also be used to reduce the event rate (as can the decimation filter).

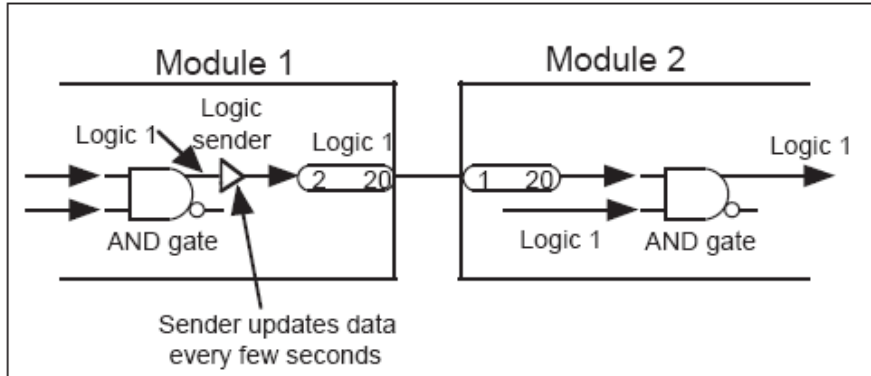
If a module is close to maximum processing capacity and a computer is connected to its RTNX port, the module may generate an L Error. Connection of the computer is the equivalent of adding events to the queue, which may be enough to cause a shortage of processing capacity. If this occurs, move the computer connection to another module.

Events and System Synchronization

Depending on the configuration, some SSD LINK messages are sent only when a change of data occurs; this is particularly true for logic messages. It is therefore possible that data in a multi-module SSD LINK system can become mismatched. There are a number of ways in which data mismatch can occur, including:

- Power failure of part of the system (asynchronous start-up);
- Modifying part of the system with SAM;
- Network communications error.

To prevent any mismatches, a sender function block is used for logic outputs transmitted over the network.



A sender has one input and one output. When a message is received at the input, it is immediately passed to the output. However, when no input messages are received, the output sends the data at a slow background rate of every two seconds; this is known as a “lazy” update.

In general, senders should be used for all data sent out over the network which is not timebase-triggered. Since logics are normally sent only on change, senders are particularly important when sending logics on the network. Senders are *not* required for:

- Connections within a single module;
- Potentiometers;
- When data is sent over the network via continuously-triggered function blocks.

Initialization

It is important that the SSD *LINK* system software blocks achieve the correct state on power up. Otherwise, data mismatch can occur on power up of the whole system.

Some function blocks have an initialization state which can be set on power up. This initialization state determines the function block state when power is applied and is set up at configuration time via ConfigEd. Initialization is commonly used for state dependent logic blocks such as latches.

Persistent inputs always store the last data set before the system is powered down.

Initializer Function Block

The Initializer function block allows power on initialization of any variable. These can be logic, values, or ordinals. The initialized variable is sent only once, on power up. The required initialization state is set from ConfigEd.

Only one initializer function block for any given initial value, logic, or ordinal is required within a single module. Initialization blocks are required only if no initialization state is available for a particular function block.

Dormant Function Blocks

On power up of an SSD *LINK* module, any function blocks which do not have persistent parameters are activated only when triggered by an incoming message. If such a function block never receives an incoming trigger message, it will remain in a dormant state in the SSD *LINK* module. Accessing an input (through SAM) which has never received any input will return its initialization state whether persistent or not.

Execution Times

Timing delays within the SSD *LINK* system can be broken down into two areas:

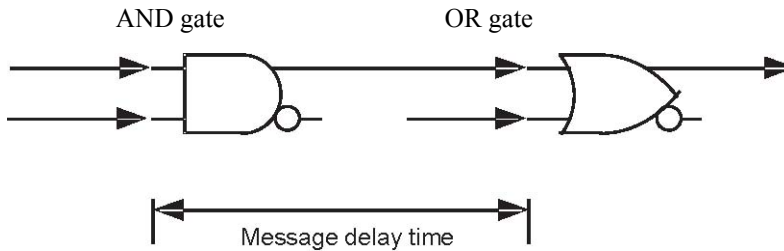
- Network message passing delays;
- Software block message passing delays.

Network Delays

These are short delays for messages passing over the SSD *LINK* network. For a message passing through a module and not being absorbed, the total delay for it to enter and exit the module is about 84µS (see Section 3.2).

Software Block Delays

These are delays within the modules caused by the processing activity within the module. The following diagram shows two connected function blocks. The delay to execute the first block and pass a message to the second block is between one and two mS.



This does not include any delays due to event queue

number of messages (connections) within the configuration. If there are a large number of cascaded software blocks, the total delays can be considerable. The use of large, application-specific function blocks is highly recommended, since this will cut down on the number of internal connections and therefore event queue activity.

The closed loop control (winder) blocks are optimized for minimal delays in the feedback path. It is very difficult to quantify the actual module delay time, due to the event-driven nature. The critical path(s) should be kept as fast as practical. Typically, the feedback signal is the critical one.

SSD *LINK* Drives

The SSD *LINK* drives operate in a way different from that of general SSD *LINK* modules. Here, the whole drive block diagram is executed at a regular scan time. The block diagram is fixed and cannot be altered with ConfigEd.

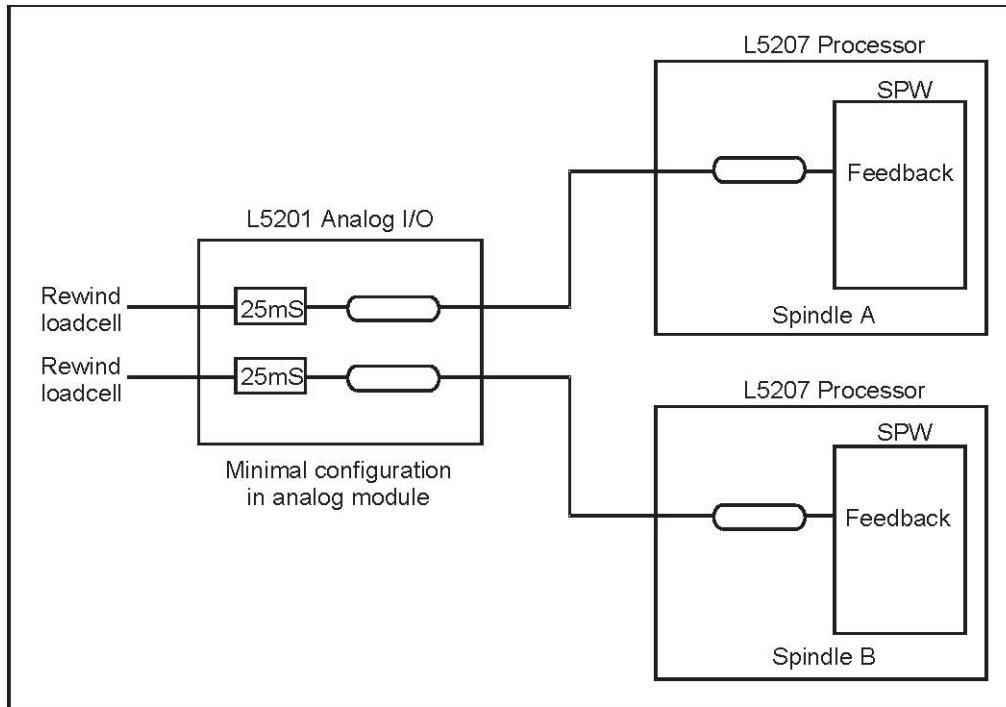
ConfigEd is, however, used to configure parameters and assign connections from the fixed block diagram to other SSD *LINK* modules. If output connectors are assigned, messages are sent at a regular rate. This rate is set when the connectors are assigned.

Application-Specific Function Blocks

In order to achieve the best performance from application-specific function blocks (e.g., the speed programmed winder - SPW), it is highly recommended that a processor module be allocated for each drive.

For optimum response, the feedback variables should be provided from analog modules which do not contain any significant configuration. If this is done, execution times for the closed loop winder function blocks will be on the order of 20mS. The analog input sample time determines the winder function block execution time and should be set to 25mS. To ensure this performance from application-specific function blocks, those blocks should be placed in processor modules with no other continuously executing function blocks.

In the case shown below, the tension feedback is via load cells, which are amplified to produce a 9V full scale analog signal to the analog module inputs. In this example, 9V full range is used to allow the necessary closed loop feedback over-range.



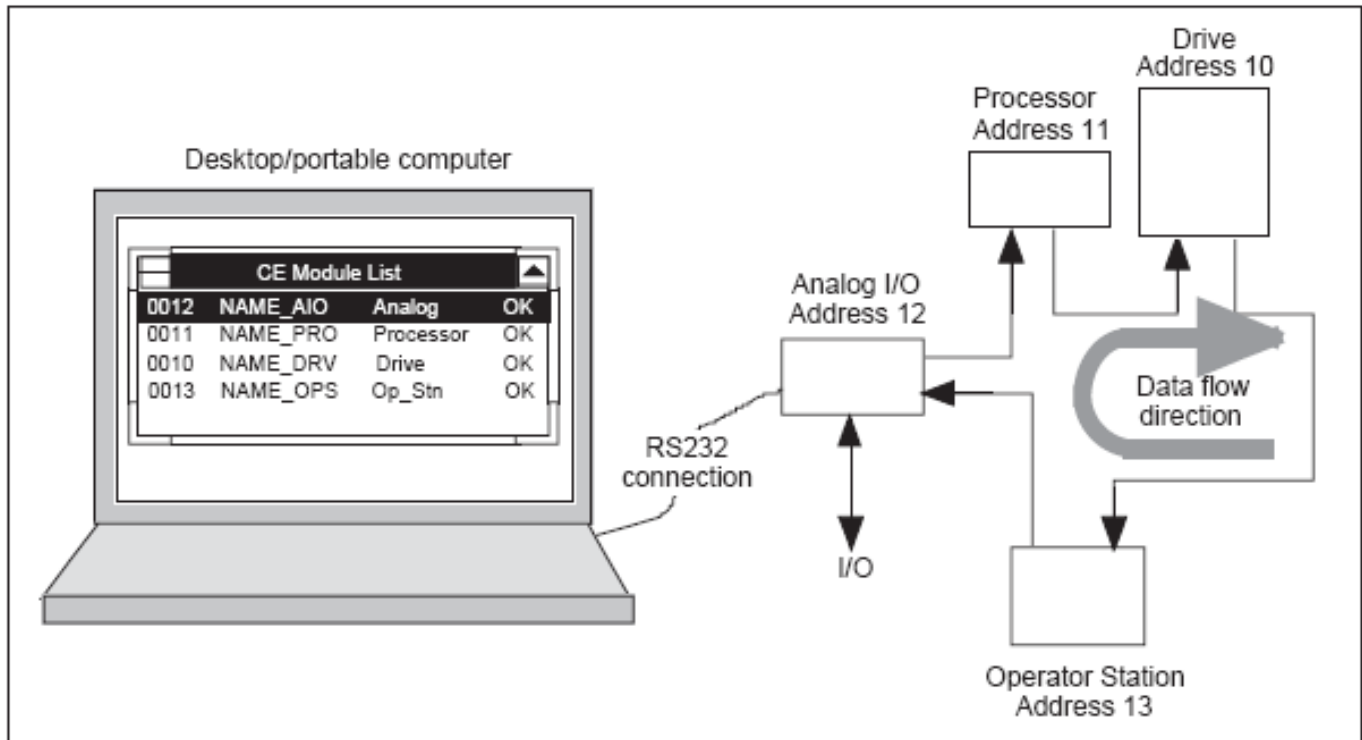
INSTALLATION AND STARTUP

After the SSD *LINK* system is designed, the configuration files have to be loaded into the modules on the network through the use of ConfigEd. Module configurations can be installed to all the modules on the network by connecting the computer to any one of them.

Each SSD *LINK* product has a serial number and a default configuration installed during manufacture. If needed, the serial number can be accessed through the Get Info command in ConfigEd or SAM.

In general, identical modules will contain identical shipping configurations and addresses. Despite the duplicate addresses, the network will function on first power up unless there are physical problems. Modules which do not enter their run state due to duplicate addresses will flash their health LED; for more information on module health LEDs, see Section 8.3.1. Operational configurations and addresses must then be installed using ConfigEd; this procedure is covered in detail in the ConfigEd user manual.

It is important to know the order and the addresses of the modules on the SSD *LINK* network to avoid installing configurations into the wrong modules. In the diagram below, the modules on the module list are displayed, from top to bottom, in the order of fiber optic data flow. The top entry in the list is the module to which ConfigEd is connected. The bottom entry is the last module on the network.



It is best not to connect the computer to an operator station since that connection will cause the operator station to enter its halt state at power up. Using Install at Address in ConfigEd automates the selection of the target modules during installation. See the ConfigEd manual for more information on installing configurations.

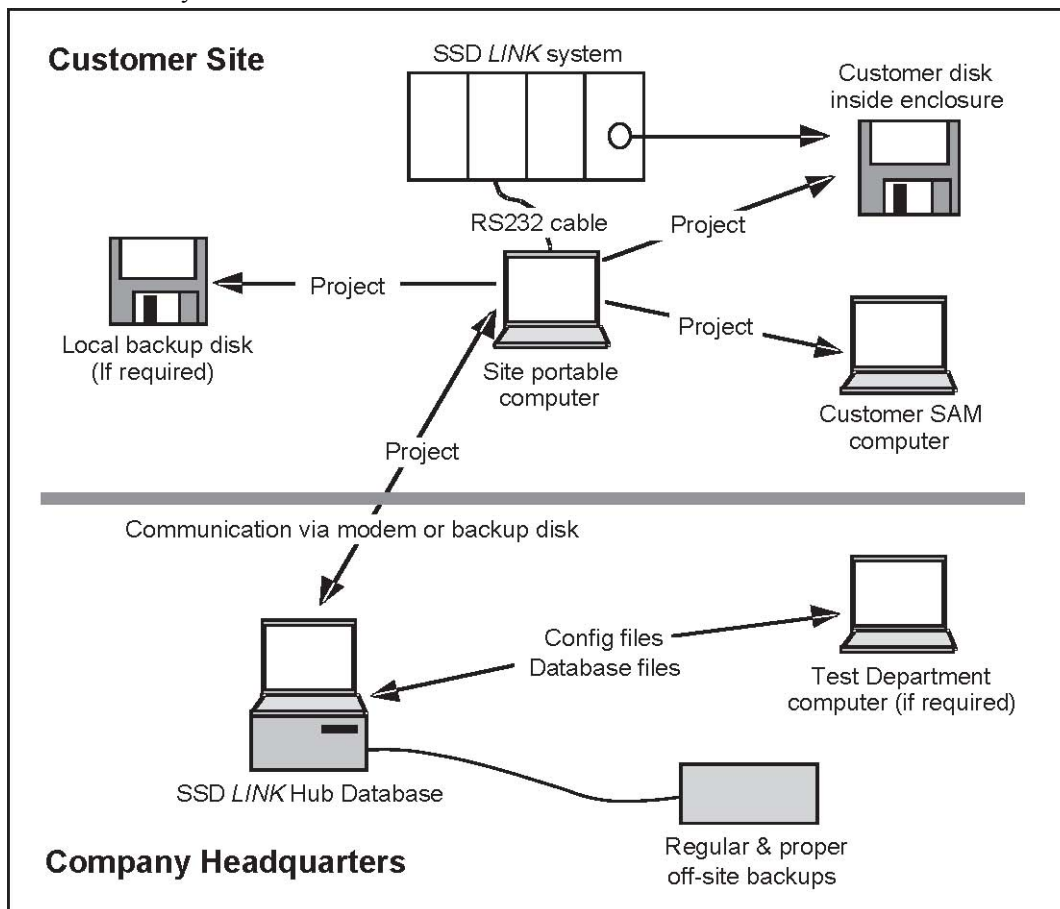
This page intentionally left blank.

Chapter 7 Configuration Procedures

This section gives recommended procedures on how SSD *LINK* configurations should be handled during the life of SSD *LINK* systems. Following these procedures will help to ensure systems are properly commissioned and backed up, and that no information is lost or corrupted.

CONFIGURATION STORAGE AND SECURITY

The following diagram shows the various computers and floppy disks that may be used for modifying and storing configurations in SSD *LINK* systems.



Once the configurations for a system are completed, they are registered with the parent Hub database (see Appendix A for details) which retains the files for security. The configuration files should also be backed up on a floppy disk on site, and left inside the appropriate system enclosure for easy access in case of equipment failure.

The procedures may appear complex, but the primary objective is to ensure the configurations in the SSD *LINK* system modules, the company Hub database, and the customer's SAM computer match exactly. This is critical since the configuration information is not stored solely inside the actual modules; some is stored on the configuration computer.

A secondary objective is to ensure sufficient backups are kept of the actual configurations so the current configurations will not be lost should any computers or other equipment fail.

It is assumed the Hub SSD *LINK* database contains the latest version of the configurations when work on the project is complete. (Manager is used to import and export projects and databases, and may also be used to delete copies of the

configurations from other computers (such as a field person's laptop machine) to conserve hard disk space on those computers. See the Manager manual for details on these procedures.)

Project Ownership and Write Protection

An important part of the Hub/user relationship is the concept of ownership of projects. When a database creates a project, that database owns the project and has write access to it. That access is represented by a write token. If a project is imported or exported in read/write mode, the write token travels with it to the new destination and confers the right to modify it.

Write token protection is intended to remove the possibility of multiple copies of the project being modified in parallel, leading to conflicting configuration revisions, overwritten files, and lost work. Project ownership is intended to ensure that one copy of the project exists somewhere in a permanent repository from which it cannot be deleted.

Transferal of project ownership typically occurs when either:

- The creating user database calls in to its parent Hub through a modem, registers the project with the Hub, and transfers ownership of (and the token for) that project to the Hub;
- Two Hubs sharing an archive move the project via the archive process.

Movement of the write token must not be done casually; a decision must be made as to where the token *legitimately* belongs. The project should only be modified by the holder of the write token. See the Manager manual for more information on project ownership and the token.

DESIGNING SSD LINK SYSTEMS

This procedure describes how the SSD *LINK* systems are designed and how the configuration files are handled in the Hub database. SSD *LINK* system hardware should be designed according to industry standard procedures, and hardware drawings should be produced to allow the system to be built and documented.

Generate Project

A new project should be generated in accordance with instructions in the ConfigEd manual and, when completed, logged in to the Hub computer database. This will ensure all projects can be referenced at the Hub database for future support.

All project data should be filled in correctly to ensure the correct information appears on the configuration drawing prints.

Floppy Disks

Two floppy disks should be generated, one labeled the customer disk, the second the backup disk. The disks should be size 3.5", 1.44MB, and formatted for IBM PC.

Each disk should be marked with the following information:

- Work order number;
- Customer name;
- Customer site location;
- ConfigEd project number and name (found on the scratch pad window title bar when the project is selected).

Configuration Design

The total system configuration should be split up into convenient blocks for allocation to the various SSD *LINK* modules. For multi-drive systems, the configuration should be split to allow standard blocks which can be easily duplicated. Application-specific function blocks should be used where possible.

When any configuration is installed into an SSD *LINK* module, there must be at least two free pages of module memory space. If in doubt, test questionable configurations to ensure there are at least two free pages by installing to the target module.

Use ConfigEd or SAM to change set up parameters within the function blocks.

TESTING (CONNECTED TO HUB DATABASE)

This procedure describes how the configurations should be handled during the testing of SSD *LINK* systems. It is assumed the test computer contains the SSD *LINK* database (or is networked to the one that does) and is capable of running the SSD *LINK* software tools.

Test Procedure

- Power should be applied to the system using standard procedures;
- The module configurations should be installed to the appropriate modules on the SSD *LINK* network;
- At least two free pages of module memory should remain when configurations are installed;
- All modules should be checked to ensure satisfactory health status after all configurations have been installed.

Sufficient tests should be applied to the system to ensure the functionality matches that of the system specification; where necessary, the configuration files should be modified. SAM may be used to aid testing of the system.

Backing Up and Updating

- If SAM has been used to make permanent modifications to SSD *LINK* module configurations, use the Update command within ConfigEd to update all the affected configurations. Check the scratch pad to ensure no errors have been generated during the project update procedure;
- If the configuration files have been modified using ConfigEd during the course of the day, export the project in read-only form to the backup floppy disk using Manager and store the backup disk in the allocated storage area.

TESTING (NOT CONNECTED TO HUB DATABASE)

This procedure describes how the configuration files should be handled during the testing of SSD *LINK* systems. It is assumed the test computer:

- Does not contain the Hub SSD *LINK* database and is not connected to the computer that does;
- Contains the up to date SSD *LINK* database (which is not a direct copy of the Hub SSD *LINK* database) and SSD *LINK* tools. If the Hub SSD *LINK* database is copied, there is a risk of future project configuration file corruption.

Before Testing

- Transfer the current configuration files from the Hub SSD *LINK* database to the test department computer; this is best done by exporting the project in read/write form from the Hub database.
- Obtain a copy of the current configuration block diagram drawings.

Test Procedure

- Power should be applied to the system using standard procedures;
- The module configuration files should be installed to the appropriate modules on the SSD *LINK* network;
- At least two free pages of module memory should remain when configuration files are installed;
- All modules should be checked to ensure satisfactory health status after all configurations have been installed.

Sufficient tests should be applied to the system to ensure the functionality matches that of the system specification; where necessary, the configuration files should be modified. SAM may be used to aid testing of the system.

Backing Up and Updating

- If SAM has been used to make permanent modifications to SSD *LINK* module configurations, use the Update command within ConfigEd to update all SSD *LINK* modules. Check the scratch pad to ensure no errors have been generated during the project update procedure;
- If the configuration files have been modified via ConfigEd during the course of the day, export the project in read-only form to the backup floppy disk using Manager and store the backup disk in the allocated storage area;
f Transfer projects to the Hub SSD *LINK* database.

Modifying Configurations on the Hub Database

- Transfer the configuration files from the test department computer to the Hub SSD *LINK* database.
- Modify the configuration files;
- Transfer the files from the Hub SSD *LINK* database back to the test department computer before continuing with the test.

SHIPPING SYSTEMS AFTER TESTING

After testing but before shipment of the system, the following actions should be performed:

- If SAM has been used to make permanent modifications to the project, use the Update command within ConfigEd to update the affected configurations. Check the scratch pad to ensure no errors have been generated during the project update; *f* Print out the configuration drawings and place them, together with the up to date hardware drawings, in the enclosure;
- Export the project in read-only form to the customer disk using Manager and place the customer disk in a secure container mounted inside the enclosure before shipment of the system;
- Export the project in read-only form to the backup disk using Manager and store the backup disk in a secure storage area.

If the testing computer was not connected to the Hub SSD *LINK* database, add the following steps:

- Register the project with the Hub SSD *LINK* database and ensure all current configuration files are present on the Hub SSD *LINK* database;
- If desired, the project may be deleted from the testing computer using the Delete function within Manager.

COMMISSIONING SSD *LINK* SYSTEMS

This procedure describes how the SSD *LINK* project should be handled during the commissioning of SSD *LINK* systems. It is assumed:

- The backup disk is available or, if not, a floppy disk is available formatted to 1.44MB for IBM PC and marked with the following information:
 - Work order number;
 - Customer name;
 - Customer site location;
 - ConfigEd project number and name.
- The current SSD *LINK* configuration block diagram drawings are available and the ConfigEd project number is shown both on them and on the ConfigEd scratch pad title block;
- The customer disk is available.

Obtaining the Configurations

To ensure the current configurations are used when modifying systems, transfer the project configurations from the SSD *LINK* Hub database to the site computer. The configurations may be transferred from the Hub database using the backup disk or through use of a modem.

Daily Commissioning

If the configuration or any parameter is to be changed in any way during the course of the day, use SAM to change set up parameters within the modules; ConfigEd must be used for all configuration changes.

- SSD *LINK* drives should be commissioned using instructions in their operating manuals and standard procedures;

- With ConfigEd, backup files are produced automatically when a configuration is installed; there is no need to use the Backup command. If SAM is used to change the parameters of a module, then the Backup command must be used from SAM or Update must be used from ConfigEd.

Backing Up and Updating

If the configuration or set up is changed in any way during the course of the day, perform the following steps:

- If SAM has been used to modify any set up parameters, use the Update command within ConfigEd to update the affected configurations. Check the scratch pad to ensure no errors have been generated during the project update procedure;
- If ConfigEd has been used to modify a configuration, backup files are produced automatically when the configuration is installed; there is no need to use the Backup command; *f* Using Manager, export the project in read-only form to the customer disk and replace the customer disk in the enclosure.

Before Leaving Site

- After performing the appropriate steps in Section 7.6.4, perform the following steps: *f* Using Manager, export the final commissioned and backed up project in read-only form to the backup disk or local backup disk; Using Manager, ensure the customer SAM database is up to date;
- Using Manager, import the project from the customer disk to the customer's SAM database and replace the customer disk in the enclosure;
- Return project configurations to the project design engineer via backup disk, local backup disk, or a modem;
- Using Manager, export the configurations to the SSD *LINK* Hub database and ensure all module configurations are present. At this point, the write token may or may not be transferred to the Hub database (see the Manager manual for more information);
- If the backup disk has not been used to transfer projects to the site portable computer, use Manager to export the configurations in read-only form to the backup disk and store the backup disk in a secure storage area;
- Ensure the configurations are up to date on the SSD *LINK* Hub database before using Manager to delete the project from the site computer.

SERVICING SSD *LINK* SYSTEMS

This section describes how the configurations should be handled during servicing of SSD *LINK* systems. It is assumed the Hub SSD *LINK* database contains the up to date configurations and the backup disk and the customer disk are available.

Restoring Failed Modules

If the service visit requires only the replacement of failed SSD *LINK* modules, replace any failed module with the same or equivalent type and then restore the configurations, using the customer's SAM database if necessary.

Obtaining the Configurations

To ensure the current configurations are used when modifying systems, transfer the project configurations from the SSD *LINK* Hub database to the site computer. The configurations may be transferred from the Hub database using the backup disk or through use of a modem.

Changing the Configuration or Set Up

This section applies if the configuration or set up is changed in any way during the course of the site visit. Before any changes are made to the configuration via ConfigEd, mark up the necessary modifications on the configuration block diagrams. SAM can be used to change set up parameters within the modules; ConfigEd must be used for all configuration changes.

Before modifying the configuration, use the Update command within ConfigEd to update the affected configurations.

Backing Up and Updating

If the configuration or set up is changed in any way, perform the following steps:

- If SAM has been used to modify any set up parameters, use the Update command within ConfigEd to update the affected configurations. Check the scratch pad to ensure no errors have been generated during the project update procedure;
- If ConfigEd has been used to modify a configuration, backup files are produced automatically when the configuration is installed; there is no need to use the Backup command;
- Export the project in read-only form to the customer disk using Manager and replace the customer disk in the enclosure.

Before Leaving Site

After performing the appropriate steps in the previous Section, perform the following steps:

- Using Manager, export the final modified and backed up project in read-only form to the backup disk or local backup disk;
- Using Manager, ensure the customer SAM database is up to date;
- Using Manager, import the project from the customer disk to the customer's SAM database and replace the customer disk in the enclosure;
- Return the printed configuration drawings and the project configurations to the project design engineer;
- Using Manager, export the configurations to the SSD *LINK* Hub database and ensure all module configurations are present. At this point, the write token may or may not be transferred to the Hub database (see the Manager manual for more information);
- If the backup disk has not been used to transfer projects to the site portable computer, export the modified configurations in read-only form to the backup disk and store the backup disk in the allocated storage area.
- Ensure the configurations are up to date on the SSD *LINK* Hub database before using Manager to delete the project from the site computer.

Chapter 8 Troubleshooting

SSD *LINK* systems feature many powerful diagnostic tools to assist in troubleshooting SSD *LINK* devices and the network itself. This section explains the best ways to find and solve operational problems with the SSD *LINK* system.

NETWORK EXCEPTIONS

The SSD *LINK* network passes either data cells or S (start/stop) cells through the fiber optic ring continuously. (S cells are signals sent through the fiber optic channel to signify the beginning and end of data messages and as filler between data messages.) At no time will a healthy ring be without a logical transition for more than a few μ S.

A number of tests and constraints are applied in hardware to the incoming data stream to ensure integrity of both the raw signal and the packets received. Whenever this hardware detects an error (lack of properly timed cells, malformed packets, bad CRCs, etc.), it rejects the data and alerts the module's CPU.

Low-level functions forward the relevant information to the System Control function block in the detecting module (the one whose fiber optic receiver detected the errant signal) and transmit global *network exception* messages to all other modules. The System Control function blocks in each module receive these messages and collectively maintain a local and global sense of the ring's health. SAM may be used to access the System Control function block in SSD *LINK* modules.

System Control also performs some event filtering by considering certain locally detected transient errors to be *warnings*. Only when such errors fail to clear in a reasonable time period does system control change the warning into a *failure*.

With its knowledge of network state, the System Control function block handles the restart operation configured for each module in response to system-wide errors and events. Upon restart, modules rerun their network checks as indicated by their health LED.

DETECTING SSD *LINK* NETWORK PROBLEMS

Failure in an SSD *LINK* system may be the result of either damage to or incorrect installation of the network. These faults can frequently be attributed to physical damage to the fiber optic cable or improper transmitter launch-power for the length of the fiber optic cable run (see Section 3.3.1).

The SSD *LINK* system is quite robust, and often a marginal network can be tolerated for some time without causing noticeable performance degradation. It is precisely because of this resistance to errors that special tools are required to ensure optimal performance during commissioning or service. Such validation is wise to maximize the system's tolerance to future damage or component aging.

Cabling problems are most likely to occur with fiber optic cable connecting remote parts of an installation. It is important to know the length of cable between each transmitter and receiver; that length must be within the transmission range for the SSD *LINK* module in question. Installation trunking systems may introduce extra fiber optic cable length not accounted for in the original network design. The possibility of this occurrence should be taken into account when designing an SSD *LINK* system.

It is highly recommended that L5206-2-00 repeater modules be used where the SSD *LINK* network must connect to remote desks or enclosures. The transmission range can be adjusted to a maximum of 60m/195'. Multiple repeater modules or the L5206-2-02 plastic-to-glass adapter module should be used for longer runs.

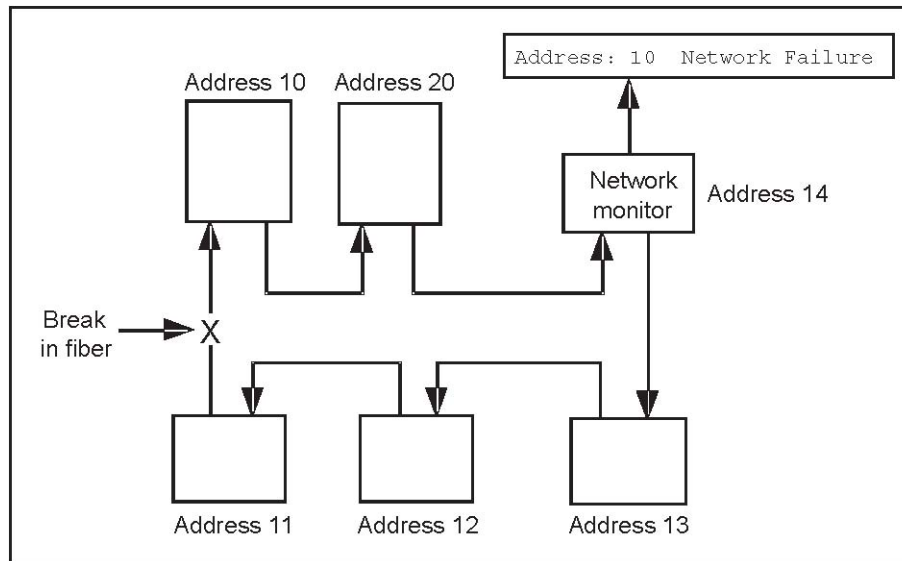
Special function blocks are available to allow configuration-level handling of some network exceptions and problems. The function blocks are:

- Operator station Network Monitor;
- Logic System Monitor;
- Watchdog;
- L5203L/Exception Logger.

For additional information, refer to the appropriate function block data sheet.

Operator Station Network Monitor

The Network Monitor function block allows the SSD *LINK* network to be monitored continuously using a single display band in any operator station. The diagram below shows the network monitor in use.



To ensure the most recent and accurate state is shown, the display for each address is unlatched. When each module becomes healthy, its indication disappears. The display cycles continuously through all pending messages.

The network monitor is intended to provide the operator with information about the cause of a system shutdown due to network or module failure. It may not detect borderline network segments, since those warnings can occur and then clear within one operator-station screen refresh. For diagnosing these and other potential problems during commissioning or servicing, the L5203L/Exception Logger (see Section 8.2.4) is a useful tool.

Logic/System Monitor

The System Monitor performs the same function as the network monitor, but instead of a graphical display, it generates a logic output. An example of its use would be to shut down a drive line after detection of a network error.

The function block may be configured to indicate when one or more of the following conditions occur:

- A module is halted;
- A module has a missing or bad configuration;
- A module crashes; *f* A module has an H (high level) error;
- A module fails self-test; *f* A module detects a network warning;
- A network detects a network failure.

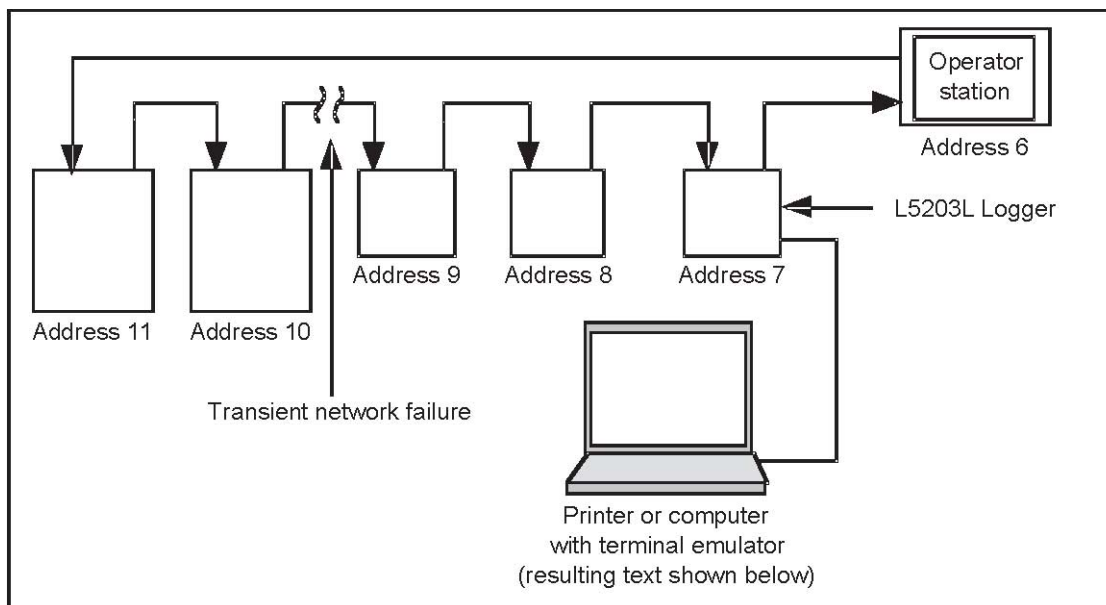
Logic/Watchdog

The watchdog function block implements a timer function and provides both non-inverted and inverted outputs. It contains a timer which is restarted by the receipt of a message at the input.

Any kind of signal may be connected to the input; the fact of its receipt, rather than the information contained in a received message is significant. The timeout time is a configuration parameter entered in milliseconds.

Network Errors and the L5203L/Exception Logger

The best way to identify transient network faults is to add an L5203L/Exception Logger module to the system. The network exception handler will report such problems to either a computer or a printer. In the example below, the fiber optic cable just before module 9 has a transient failure and the resulting messages from the L5203L/Exception Logger are shown.



Transient case:

```
1d 7h 12m 37s 571ms addr 9 Network Warning
```

```
1d 7h 12m 37s 586ms addr 9 OK
```

The SSD *LINK* system will tolerate network warnings for up to one second. If the warning persists for longer than one second, a network failure is declared and modules' System

Control function blocks will trigger a restart procedure and an LED status alarm.

Failure case:

```
1d    7h    12m    37s    571ms    addr  9    Network Warning
1d    7h    12m    38s    571ms    addr  9    Network Failure
1d    7h    12m    38s    594ms    addr  9    Shutdown
1d    7h    12m    38s    606ms    addr  8    Shutdown
```

If the L5203L/Exception Logger is not available, it is possible to monitor the individual module system control function blocks using SAM or an operator station network monitor function block. However, the L5203L/Exception Logger is the best tool to use in detecting network warnings.

L5203 Printer Connections

The L5203L configuration is preset to use the RS232 port B transmit terminals at 9600 baud, 1 stop, no parity. These settings may be configured by the user, however.

Connect the RS232 receive line of the printer (pin 3) to the port B transmit terminal (terminal 9) of the L5203L, and signal ground (pin 7) of the printer to signal ground (terminal 11) of the L5203L.

Note

Some serial devices reverse their transmit and receive lines. If nothing is received after hooking up the printer, try pin 2 of the printer instead of pin 3.

DETECTING MODULE FAILURES

Module failures or errors may be diagnosed using the module's health LED or by using SAM or ConfigEd.

Health LED

One way to check for module error is to examine the module's health LED, which produces a steady or blinking code as detailed on the following page.



Continuous off indicates the module has no power or has experienced a serious hardware failure.



Continuous on indicates the module is running.



1/2 second rapid blinking followed by 1/2 second off indicates the module is ready to start but is unable to send a message to itself around the ring to verify the ring is complete. Check for unconnected modules, loose connections, and unpowered modules.



1/2 second rapid blinking followed by 1/2 second on indicates the module is ready to start but is waiting for other modules on the network to become ready to start. Check for modules that are stopped or in an error state; the system will not run until all modules are ready.



Continuous rapid blinking indicates the module has discovered a duplicate address on the network and has removed itself from the network and stopped. If a system has two or more modules with the same address, it cannot run. Correct the problem by reconfiguring the offending modules.



One second on followed by a single 1/4 second blink off indicates the module has stopped as the result of a tools request. This typically occurs during reconfiguration of the module.



One second on followed by two 1/4 second blinks off indicates the module is unable to run because it has no configuration. Install a configuration using the tools software.



One second on followed by three 1/4 second blinks off indicates the module has stopped due to a low-level internal error (L error). Run module Get Info from either SAM or ConfigEd (see Section 8.3.3). If the problem persists, contact your service representative.



One second on followed by four 1/4 second blinks off indicates the module has stopped due to a high-level internal error (H error). If the problem persists, contact your service representative.



One second on followed by five 1/4 second blinks off indicates the module has stopped due to a hardware error. Replace the module.

Module List

To determine the cause of a module error, attach a computer running either SAM or ConfigEd to the system, call up a module list and locate all modules in an error state.

Causes of failure in an SSD *LINK* module include:

- Maximum processing throughput exceeded;
- Configuration errors;

- Hardware failure.

The following faults can be diagnosed from the module status:

Module List/Status	Possible Cause/Action
Very slow module list update	Fiber damage or incorrect length
Truncated list	Incorrect launch power (fiber optic light level) at a drive or repeater.
Errors while updating the list	Wrong software tools baud rate; reset to lower Baud rate. RTNX router module busy; plug computer's RS232 cable into another module.
Health status ≈ checking network	Broken or damaged module or fiber cable; divide and conquer by repeatedly splitting ring(s) in half until the problem module or cable length is singled out.
Health status ≈ halted	Can happen when configuration installations fail; cycle power supply or restart using tools. Replace module if problem persists.
Health status ≈ duplicate address	One or more other modules have the same address as this module; reconfigure, install, restore as needed.
Health status ≈ No Config.	No valid configuration; install or restore configuration.
Health status ≈ peer halted	Waiting for another module to start; check all modules in any of the other halted states for Their health status.
Health status ≈ H error	High level error; reinstall configuration, check data from Get Info for major function block(s).
Health status ≈ L error	Low level error; review configuration, perform Get Info using tools. Cycle power or restart as a temporary work-around.

L Error Failures

Mistakes in configurations somewhere in the system are the most common causes for L error messages, though the state is an all-encompassing runtime error. The first step is to perform a Get Info on the problem module in either SAM or ConfigEd. The processor's stack will be displayed for the module in question.

Module Overload

If the Get Info stack trace mentions any error related to Out of Heap, then the module's processing ability (*not* related to the size of the configuration) is probably being exceeded.

The solution to this problem is to examine the configuration and consider what system events precipitated the overload.

Examine related trigger inputs and signal paths and any arrangement or setting that causes an excess of messages to overwhelm the module. The reason could be as simple as trying to display six values at 30ms each or as complex as a distributed signal loop causing messages to cascade out of control. Consider resetting input rates to a lower rate wherever possible to lower the demand on the processing capability of the configuration. A burst of network warnings can also cause L errors in heavily loaded modules; use an L5203L/ Exception Logger to catch these (see Section 8.2.4).

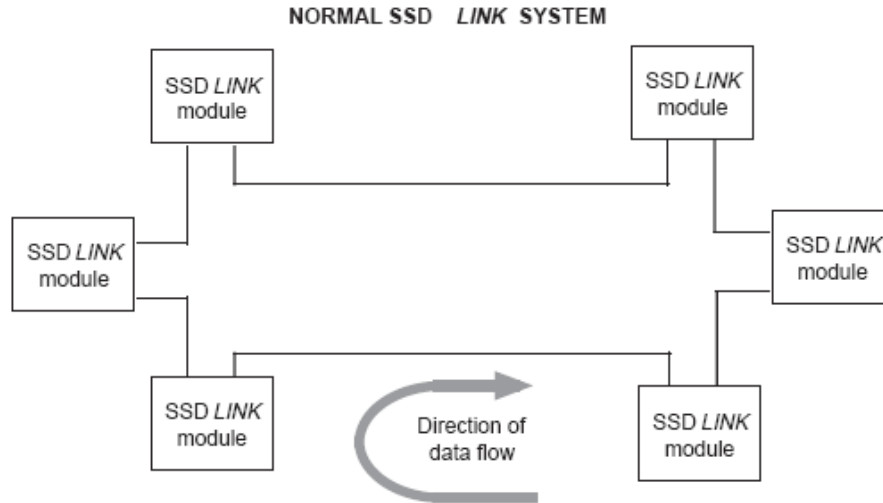
Display processing tasks occupy a significant percentage of L5102 operator station microprocessor resources. Extra attention to potential overloads should be given to these modules. High data rates should be reduced using, for example, the decimation filter.

Illegal Values

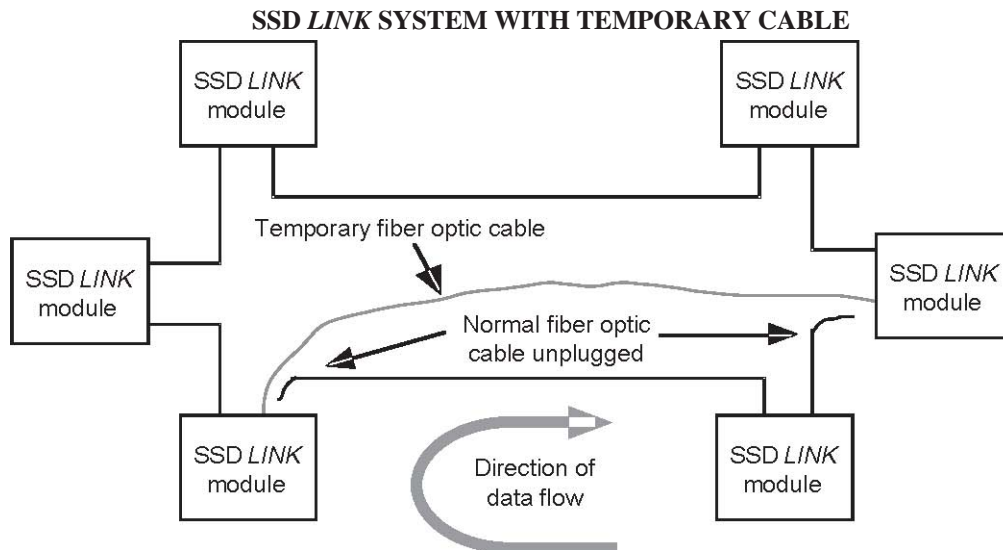
Ordinal outputs should not be connected to function blocks having value inputs. Such connections can result in illegal values which can generate L errors and cause modules to crash. An ordinal-to-value conversion function block is available and must be used when ordinal-to-value connections are required.

Shorting Links

A length of fiber optic cable can be used to temporarily "short out" modules on the network. This technique can be used to identify the module causing others on the network to enter the peer halted state.



Using a length of fiber optic cable, create a link between non-consecutive modules on the network. Bypass questionable modules one by one and attempt to restart the system. This procedure should be followed until the peer halted module will restart. Using this technique, the source modules causing the problem can be found.



IMPROPER OPERATION

Many SSD LINK function blocks have trigger inputs. If a function block is not triggered properly, its outputs will not be updated. This can bring on a situation where apparently valid configurations do not work. Other problems could include incomplete configuration or lack of familiarity with a function block.

These problems should be detected and corrected as part of the system test procedure performed prior to shipment to the end customer. Once the SSD LINK system has been properly installed and commissioned, SAM provides a powerful diagnostic tool for system troubleshooting.

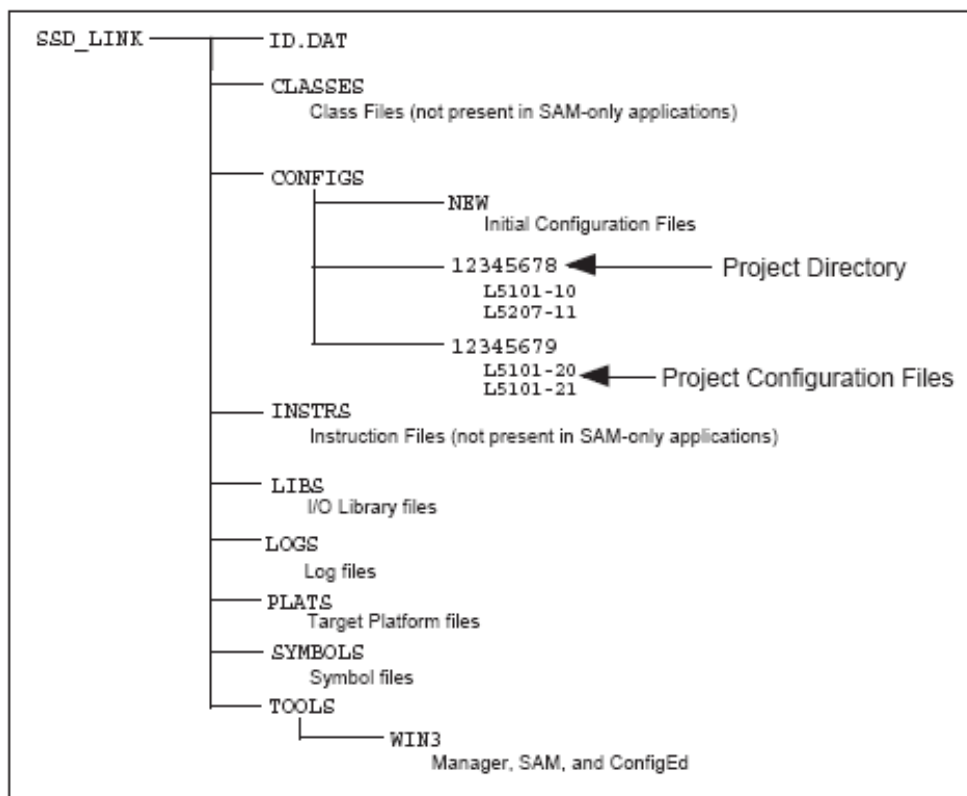
Appendix A SSD LINK Database

DATABASE DESCRIPTION

The SSD *LINK* database structure is a collection of directories containing files. The root of the directory structure is a directory named `ssd_link`. The database contains the following file items:

- Function blocks;
- New module configurations;
- I/O libraries;
- Project configuration files;
- SSD *LINK* configuration tools (Manager, ConfigEd, and SAM).

The following diagram shows a typical SSD *LINK* database directory structure.



This database must be available (either on the local disk or accessed over a network) to any computer used for configuring or maintaining SSD *LINK* systems. It is installed automatically by Manager during the tools installation procedure.

WARNING

Under no circumstances should any SSD *LINK* database be modified in DOS without consulting SSD Drives.

The SSD *LINK* database directory structure is rigidly controlled and must not be altered at any time by the user. The configuration tools expect to find files in specific places within the database structure. The tools and utilities are designed to allow the database to be handled without resorting to manual intervention via the computer operating system commands.

The current SSD *LINK* database requires approximately 10MB of space on the computer hard disk. There must be sufficient additional space to allow the database to grow due to:

- The addition of new and revised function blocks;
- Updating of the configuration tools;
- New and updated user configurations and projects.

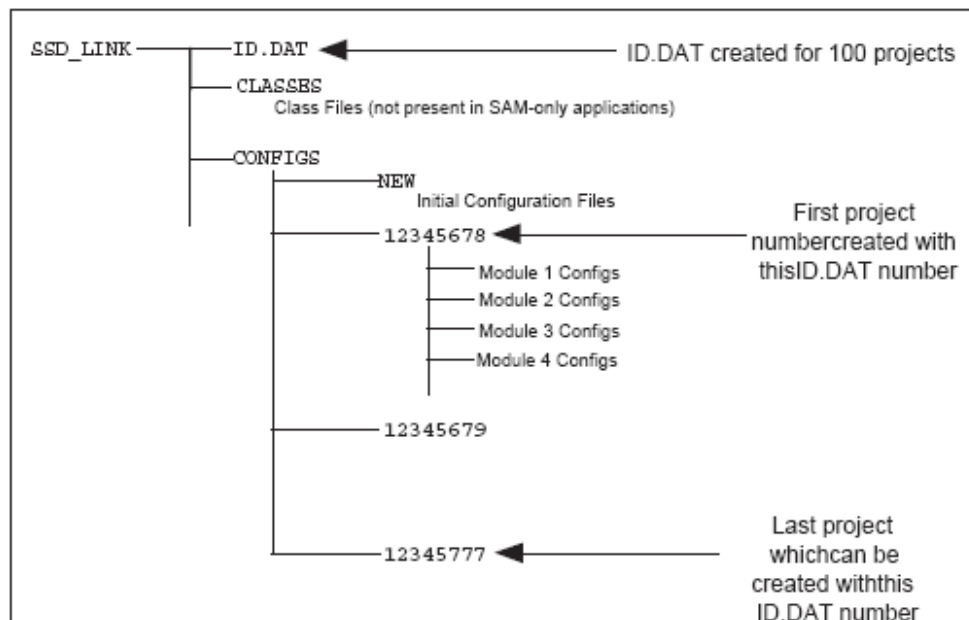
SAM AND CONFIGED DATABASES

SSD *LINK* databases are required for both SAM and ConfigEd. The SAM database is somewhat smaller than the ConfigEd database; since it is not possible to modify configurations with SAM, some of the database information is not required.

THE ID.DAT FILE

Every SSD *LINK* database has a unique identifier file, known as the id.dat file, associated with it and located in the *ssd_link* database root directory.

The id.dat file is used to allow the tools to identify the SSD *LINK* database. When a project is generated within ConfigEd, the id.dat file allocates a new, unique number to that project. This number is used as the name of the directory which will contain the various SSD *LINK* module configuration files.



The diagram above shows the project directories within the SSD *LINK* database. The directories are shown as numbers (12345678 onwards) which are created by the id.dat file when a new project is started.

The id.dat files are totally unique worldwide, and are created at SSD Drives when ConfigEd and/or SAM are ordered. The id.dat contains the following information:

- Database ID or name, which appears in the SSD *LINK* tool scratch pad;
- Remaining project directory number range.

When the id.dat files are created, the project range (the numeric IDs available to that database) which can be created is specified and stored in the id.dat file. If a database for SAM is ordered, no project configuration work can be attempted, so the supplied project range is empty.

In brief, the id.dat file assigns and tracks project numbers unique to its SSD *LINK* database. Since no two projects ever have the same number, projects may be moved between databases without risk of accidentally mixing the configurations up. If the directories were named manually by the user, the potential for conflicts and possibly dangerous results would be great. If the entire project range is used up, contact SSD Drives for an updated id.dat file.

Copying the SSD *LINK* Database

WARNING

The SSD *LINK* database must *never* be duplicated from one computer to another. If this is done, the id.dat file will be duplicated and project corruption will occur. SSD Drives will not accept any responsibility for project corruption due to duplicate databases.

The same problem can occur if a user restores the SSD *LINK* database from a backup (e.g., after some sort of disk failure). If the whole database is restored, including all project directories, all will be OK. But if, for example, three projects were created since the last backup was made and these directories (but not the id.dat file) are manually preserved, future attempts to create projects will fail. ConfigEd will try to create the first of those three project directories which already exists. To prevent this, backup and restore the file system regularly and as a whole unit (never only parts thereof).

Note

An on-site service call may be necessary for SSD Drives to repair severe database damage resulting from customer failure to adhere to proper handling guidelines. In such cases of severe damage, it may not be possible to recover the database.

DATABASE AND PROJECT MANAGEMENT

Manager software is used to transfer information between a new or revised database (either from a floppy disk or via modem) and the configuration computer, as well as for deleting and archiving projects. These features ensure the user's database can be kept current, and allow the user to manage projects and databases in an efficient and safe manner.

See the Manager manual for details on its database management functions.

PROJECT CONFIGURATION FILES

When configurations are installed with ConfigEd, a number of files are generated in the user's SSD *LINK* database, including three files relating to each specific module configuration:

- NAME.*** Main configuration file
- NAME.SYM Symbol file; generated during installation
- NAME.CLS Class file; generated during installation
- NAME.BKU Backup file generated during installation

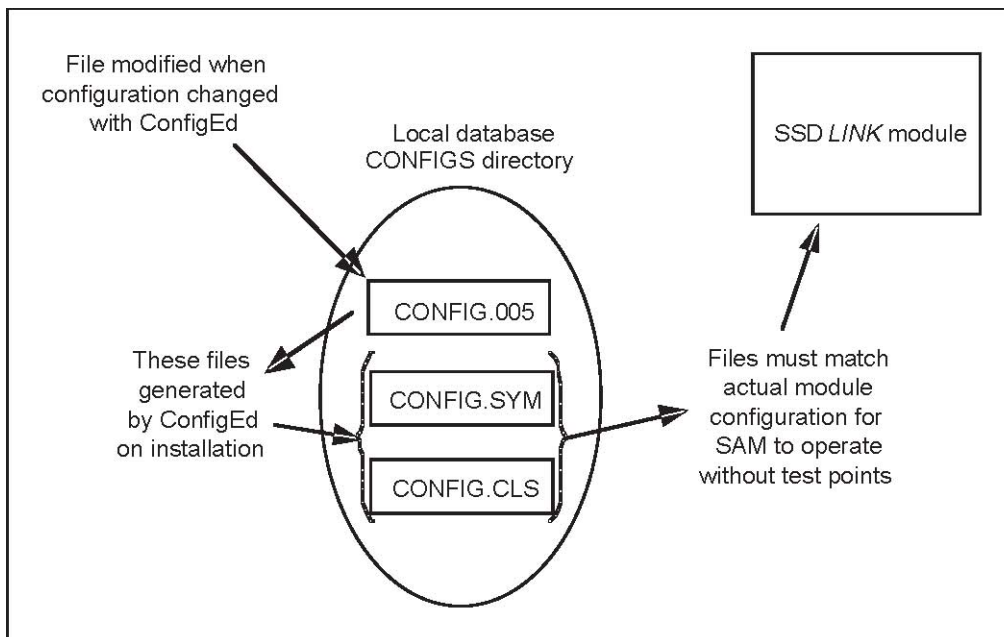
NAME refers to the name of the module configuration specified by the user in ConfigEd, and *** refers to a three digit revision number. This number starts at 001 for the first revision, and increments (when there is a change to the configuration) as it is saved. ConfigEd regulates the number of past revisions saved; see the user manual for details.

The .SYM, .CLS, and .BKU files are generated or updated automatically when the configuration is installed to the hardware module using ConfigEd. If a modified configuration is not installed to the module, then the files are not updated.

Configuration Files and SAM

There is not enough information solely within a module to allow SAM to access its parameters; the .CLS and .SYM files are required.

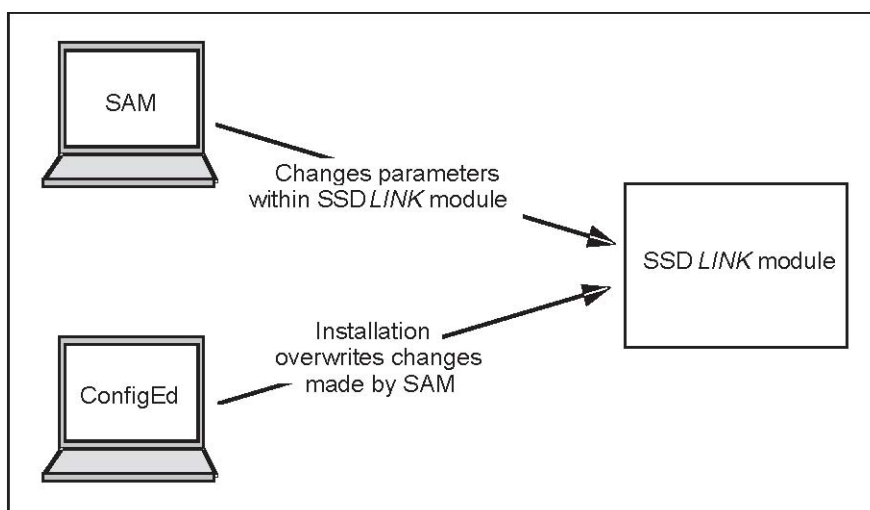
It is important these two files match the actual configuration within the hardware module. Otherwise, SAM will not work properly. The following diagram shows the three files in a local computer database, together with an actual hardware configuration.



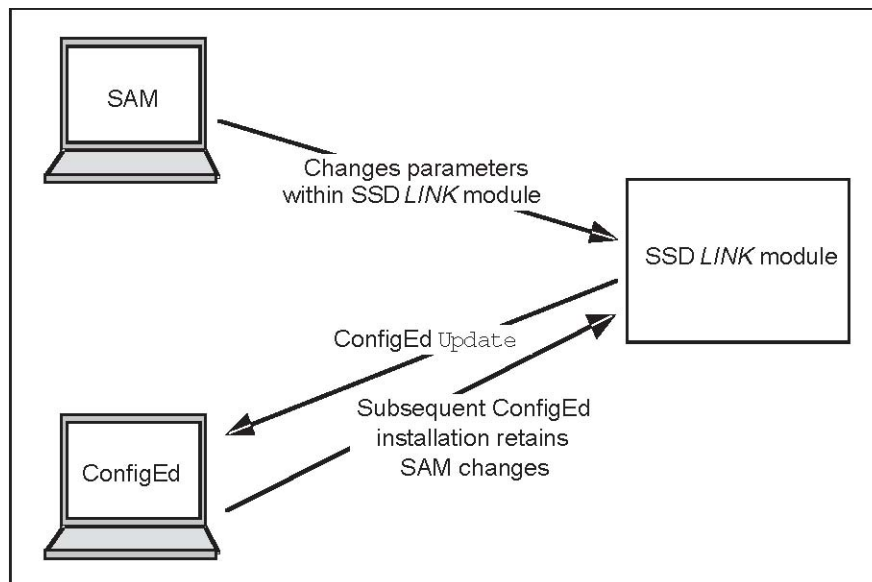
It is possible for the configuration files to become mismatched if correct handling procedures are not strictly obeyed. The following section suggests maintenance procedures for the configuration files so mismatches do not occur.

Updating the Configuration Files

SAM is used to modify parameters within SSD *LINK* modules. These modifications are *not* automatically made in the original configuration. If that configuration file is again installed into the module using ConfigEd, the changes made in SAM will be lost (see the following illustration).



This situation is resolved by using the ConfigEd Update command once work with SAM has been completed. This extracts the modified parameters from the SSD *LINK* module and updates the configuration file in the ConfigEd database so the configuration will contain the new settings.



Refer to the ConfigEd manual for more information on the Update command.

Upgrading the Project

The SSD *LINK* database, which contains function blocks as well as other items, is periodically updated to add new function blocks and modify existing ones to introduce new or enhance existing functionality and to fix bugs.

When a new project is created, information about the then-current database is stored in it. This ensures the project configurations won't require more module memory to install in the future as the core database grows. Because of this "snap shot" effect, new functionality is generally not available to existing projects.

If new functionality is needed in an existing project, ConfigEd's Upgrade command will reset the project's snap shot of the database to the current revisions of the core database.

The penalty for using Upgrade is that the configurations may require more memory in the module to be installed. For this reason, Project Upgrade should be used with great care on commissioned systems to avoid running out of configuration memory and necessitating potentially widespread redesign. The reason for leaving two free pages in a module configuration, as discussed earlier, is to allow upgrading while reducing the chance of running out of configurable memory.

Exporting a read-only copy of a project using Manager before upgrading provides a fall back if the configurations fail to fit after upgrading. See the Manager manual for more information on exporting projects.

This page left blank intentionally.

Appendix B Module Memory

Most SSD *LINK* system modules consist of three hardware PC boards:

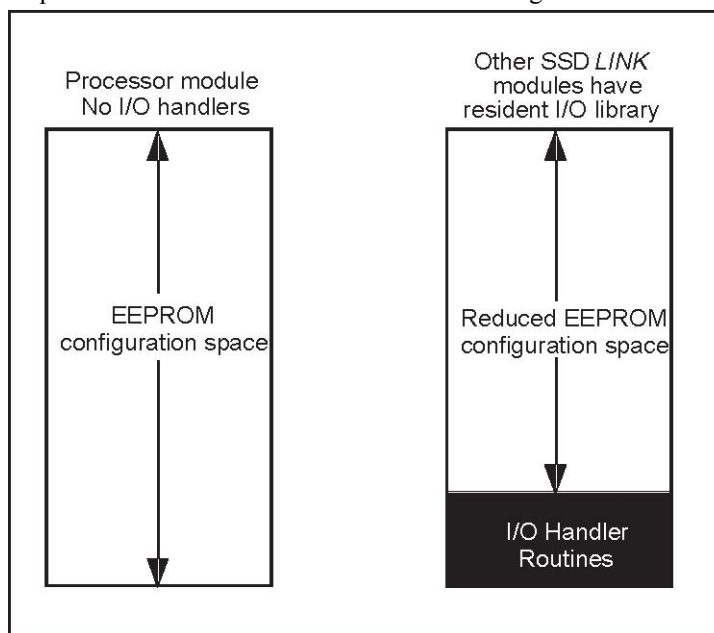
- Processor board;
- Power supply board;
- Application specific board.

The processor and power supply boards are standard throughout the SSD *LINK* system product range. The processor module contains a fixed quantity of non-volatile memory and ROM containing the operating system software. There are a number of software items that make up the final functional program for the SSD *LINK* system modules.

THE I/O LIBRARIES

I/O libraries are application-specific routines that handle the I/O interface for the modules. They also handle certain core functions (like PIDs) which not all modules support. Modules are shipped with default configurations and libraries, which allow them to power up after connection to a network. Working configurations must be installed from ConfigEd before real functionality is possible.

A completely unconfigured module has no knowledge of the type of I/O hardware present. When configurations are installed from ConfigEd, the libraries are loaded into the target module. There can be many I/O libraries for a specific SSD *LINK* hardware module; the library is specified when a new SSD *LINK* module configuration is created.

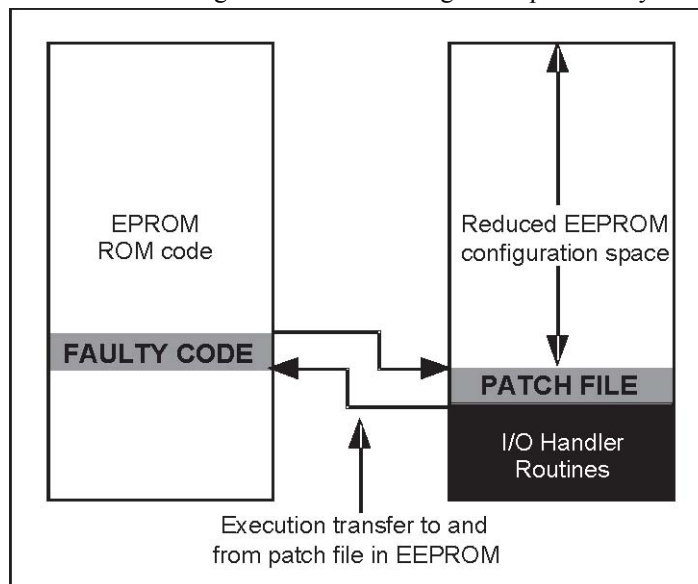


The size of the I/O library is one of the factors which determines the final memory capacity available for general function blocks. The following indicates the size of the I/O libraries for the common SSD *LINK* systems modules:

- Processor module - smallest library;
- Analog/Digital I/O - average library;
- Operator station - largest library.

PATCH FILE

This file is used to patch bugs and supply new features that have been added to the main control software in EPROM; this avoids dismantling the SSD *LINK* module to change the ROM or having to swap-out many modules.



The patch file is contained in non-volatile memory, and is loaded automatically by ConfigEd during installation. New versions of the patch file may be generated as new versions of the SSD *LINK* database become available.

As time passes after a main ROM release, the patch file grows in size due to the updates and revisions added. When a new ROM is issued, the patch file fixes are transferred to the main ROM. The patch file then shrinks back to a minimal size.

HOW MUCH CAN FIT IN EACH MODULE?

It is not easy to assess how many function blocks and connections will fit into a given module. When a finished configuration is about to be installed to the target module, it is compiled into a compact form. It is only then that the actual configuration can be assessed for size and the installation aborted if too large.

Note

It is most important that some free space remains after the configuration file has been installed so the patch file will have room to grow.

The memory space is divided into “pages” in which each page represents 256 bytes. ConfigEd indicates the number of free pages after the configuration has been installed to the module. It is highly recommended that at least two free pages of memory space be available after the configuration has been installed. Otherwise, there is a real risk that as the patch file grows, the configuration will no longer fit into the module. This is crucial for on-site equipment requiring modification. In that case, if the configuration will not fit, it will have to be re-engineered.

During installation, ConfigEd indicates in the scratch pad window the number of free pages available. If fewer than two free pages remain, the configuration should be split up into more modules.

TEST POINTS

Test points are used to allow easy access to certain parts of the configuration via SAM without the .SYM and .CLS files. Text associated with test points is stored in the configuration memory. It is, therefore, important to minimize this text to allow maximum space for the configuration. If the configuration is very short of space, it would be permissible to allocate test points outside the two page limit. If, in the future, the configuration has grown too large to fit, the test points can be removed.

MEMORY BLOCKS

When ConfigEd installs a configuration, the following blocks are reported in the scratch pad:

- Block 1 - EEPROM allocation table
- Block 2 - Persistent data items
- Block 3 - Patch file
- Block 8 - Fonts (operator stations only)
- Block 16 - Library (application-specific I/O primitives)
- Block 64 - Instructions
- Block 128 - Configurations (instances and classes)

The number after each block reports the number of bytes free in the last page allocated to each block. The most important block for the user is block 128, which contains the actual configuration.

MEMORY SPACE AND THE L5102 OPERATOR STATION

The L5102 operator station module has the least available memory due to the large I/O library and the necessary display function blocks. It should be used for display and input function blocks only, with an absolute minimum of other functions. All signal processing and logic functions should be done in other modules.

SSD LINK OPERATING SYSTEM AND MEMORY

SSD *LINK* uses function blocks loaded into the modules and connected together to form the configuration. The operating system uses an object-oriented approach, allowing very efficient use of the configuration memory within SSD *LINK* modules.

What is a Class?

When multiple function blocks of the same type are installed into a module, the block's executable code is included only once. Such a unit of code is known as a *class*. Classes exist in a hierarchy allowing subclasses to "inherit" the code and functionality present in the parent, again with that code appearing only once in the module. This saves configuration memory and makes the function block code easier to maintain.

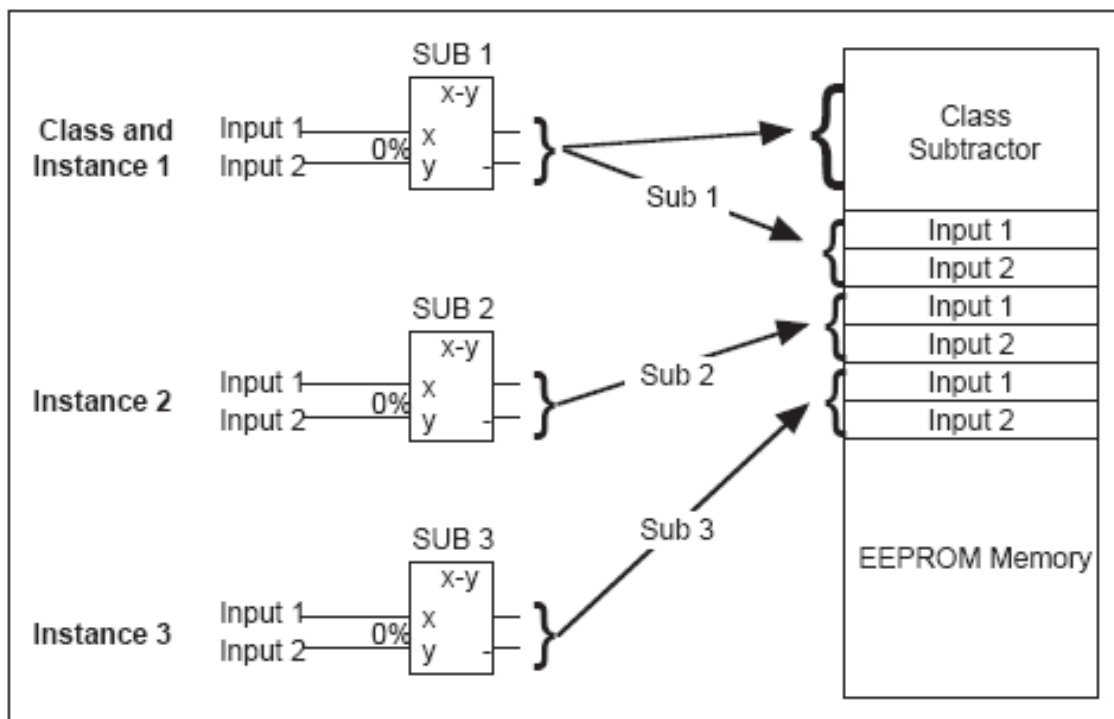
What is an Instance?

An *instance* is a particular form of a class. Instances hold the data associated with a unique and identifiable function block. When a function block is executed, the instance variables are used as inputs to the above class software. The class software is executed to form the function block outputs which may be stored in the instance or sent out the output connections or both. In essence, an instance is a function block created and named by the user within ConfigEd.

What is Stored in the Module?

In brief, there is only one *class* stored in EEPROM for each different type of function block and there is one *instance* stored for each function block.

The following diagram shows a typical class with three instances of that class.

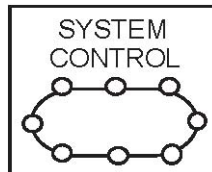


Appendix C Configuration Guidelines

This section gives some general guidelines in the use of *SSD LINK*. The sample configurations shown often relate to specific modules. Further configuration information may be found in the *SSD LINK* tools manuals.

SYSTEM CONTROL

Most *SSD LINK* modules contain a System Control block responsible for handling restart options and *SSD LINK* network monitoring and reporting.



The block is configurable to allow the following restart options:

- Restart on network failure;
- Restart on module failure;
- Restart on reconfiguration.

The restart options can be configured as Enabled or Disabled. If restart is disabled, the module will not try to restart. This is useful if it is important that the module should not shut down on *SSD LINK* network failure. The default set up for these parameters should be Disabled, but some initial configurations set them to Enabled. The L5203L logger is a notable example; in case of a failure, it would make no sense to shut down a module and have it attempt a restart at the same time it is trying to report the cause of that failure.

The System Control block can also be used with SAM to examine the state of the *SSD LINK* network. The System Control block acts as a logging device, recording the following network errors detected locally by the given module:

- Network message CRC errors;
- Network message overrun errors;
- Network message fragments.

All these logged messages indicate corrupt *SSD LINK* network messages that have been received and rejected by a particular module.

The network message fragments are the most useful for detecting fiber optic problems. Each module's System Control should be examined to find which module is receiving excessive errors. The fiber optic cable supplying that module should be examined for improper length or damage, and light levels at the receiver should be checked with a light meter. The L5203L logger described in Chapter 8 performs some of this data collection continuously.

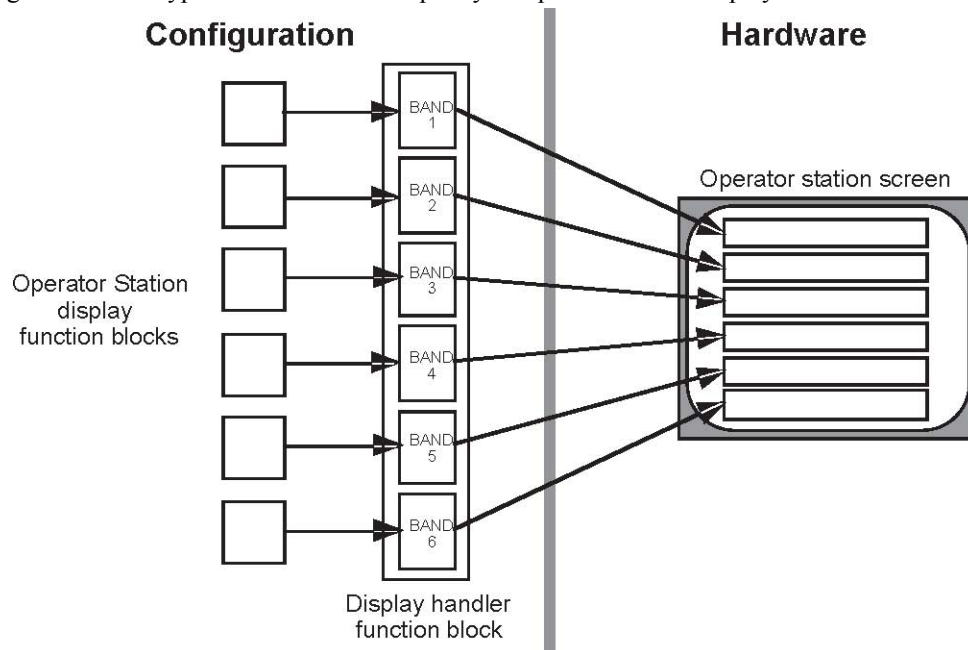
HANDLERS

A handler is a function block that groups together other specific function blocks. Handlers are used to simplify the configuration and set up of multi-channel functions. The handler may specify the position of a particular function block on the display of an operator station. Examples of handlers include:

- Digital I/O handler;
- Alarm handler;
- Operator station display handler;
- Operator station button band.

Operator Station Display Handler

The following diagram shows a typical handler used to specify an operator station display.



In order to use a handler, the following procedure should be adopted:

- 1.0 Create the individual channel function blocks, which must have unique names. Even if the channel location is not being used, a unique function block of the proper type must still be created (e.g., blank annunciators);
- 2.0 Create the handler function block if required (some exist by default);
- 3.0 Allocate in the handler the channels defined in step one. It is most important that valid handler channels are available in the list of function blocks before attempting allocation within the handler.

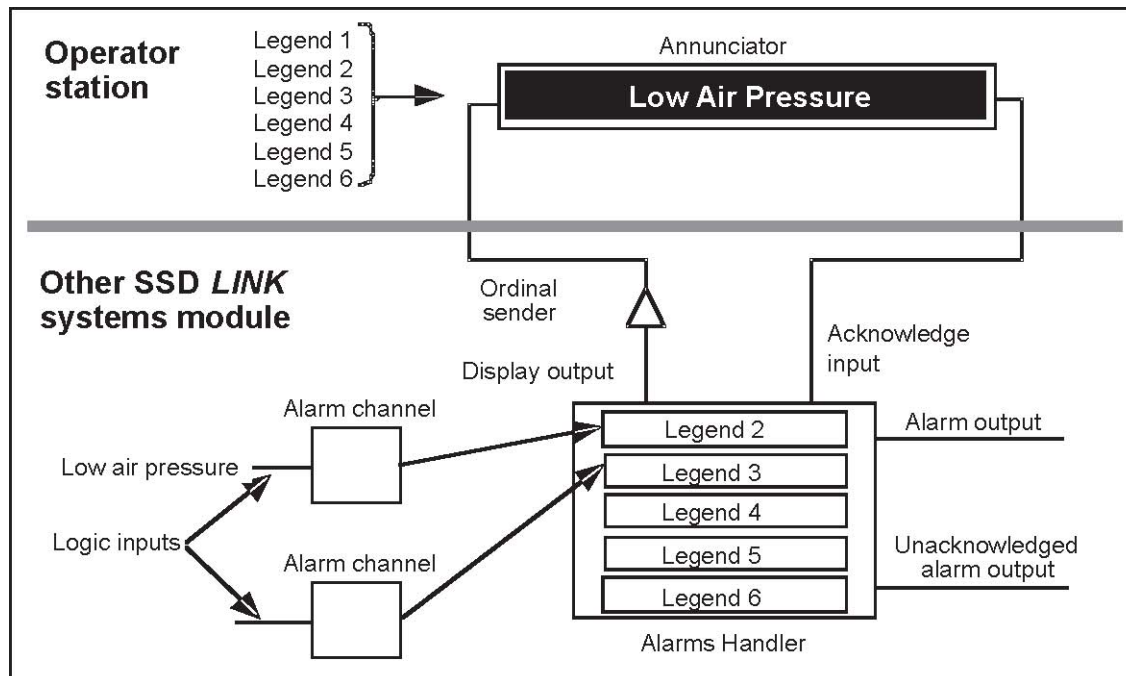
In ConfigEd, both the handler and its channel function blocks appear in the module function block window (see operation station examples in Chapter 6).

The digital module I/O handler is used to specify the placement and type of I/O allocated to the hardware terminals. It is recommended that any spare digital I/O be defined as inputs. As with any handler, there must be the same number of digital function blocks as required by the handler.

Alarms Handler

This function block allows multiple alarms to be handled. It produces logic outputs and an ordinal output to drive an operator station display.

The following diagram shows an alarm handler in use together with an operator station annunciator. Each alarm channel is associated with an operator station annunciator legend. There is a default legend (Legend 1) that is displayed until an alarm is present.



The annunciator display is shown with an alarm present. The alarm channels are designed to accept logic inputs and trigger the alarms handler.

The alarm channels are specified within the alarms handler. There are three outputs from the alarm handler. The display output is designed to connect directly to the operator station legend input. When this output is active, the appropriate legend appears and the highlight flashes. See later sections for detail about the operator station legends and highlights.

The sender is necessary to prevent mismatch between logic states in different logic modules. The two logic outputs allow alarms to perform various functions such as stopping the machine or sounding a horn.

L5102 OPERATOR STATION

The L5102 is a touch screen operator station designed for local control of a system. The module can be used to display and set any type of *SSD LINK* variable:

- Logic, via push buttons or highlights;
- Values, via pots or meters;
- Ordinals, via pots or meters.

These various display elements are function blocks loaded into the display handler within the operator station configuration. The display handler specifies which function block appears in each of the six bands of the display.

Memory Limitations

The L5102 can handle only a small number of *SSD LINK* function blocks due to its large display library. It is highly recommended that the L5102 module be used only for operator station function blocks, especially when the operator station is used for alarm annunciation where a number of different legends are stored. Legend text takes up some configuration memory.

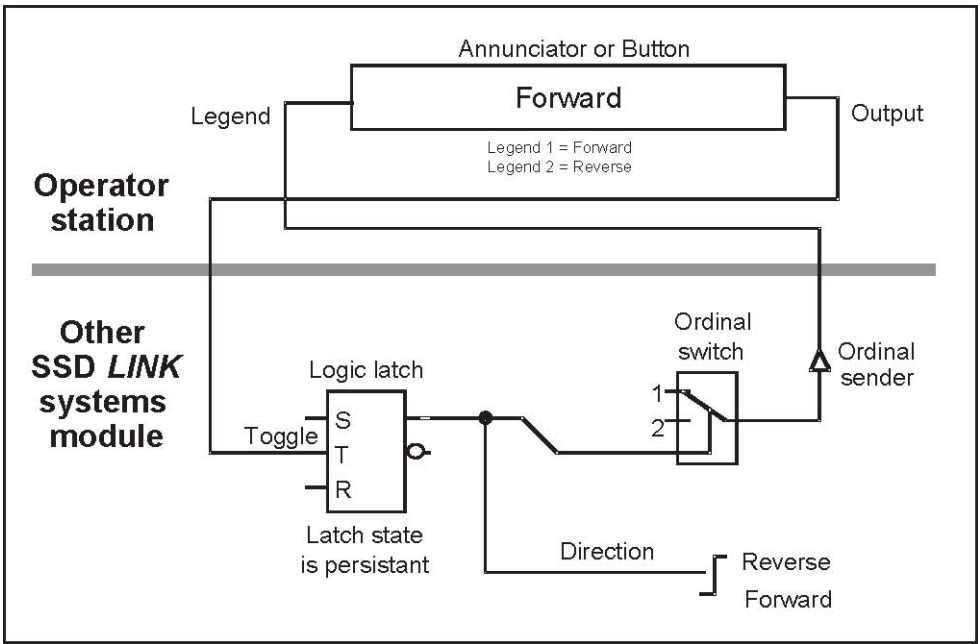
Display Legends and Highlights

The operator station has the ability to select different legends in response to a logic variable. The push button and annunciator function blocks allow different legends to be selected in response to an ordinal input derived from a number of sources.

Selecting Legends

The following sample configuration shows the use of an ordinal switch to allow two legends to be selected. A latch function block is used to implement forward/reverse selection for machine direction. The latch is used to toggle the logic variable in

response to a single push button input. The button text will change to indicate the current state. The latch function block may be persistent; therefore, its logic state is stored when the power is off.



Legends are selected using ordinal data from the switch connected to the annunciator legend input. A number of different legends can be programmed to suit the application. The theoretical limit to the number of legends that can be selected is 255. The actual total number of legends, however, will be determined by operator station memory constraints.

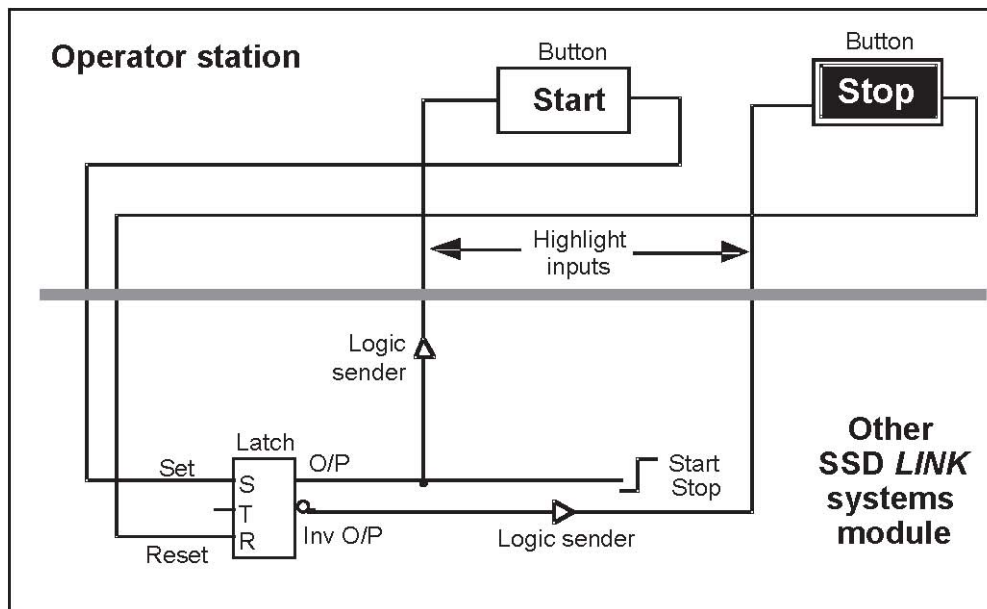
Selecting Highlights

Operator station buttons and annunciators may be displayed as highlighted to allow attention to be drawn to a particular function. Three operating states are possible for the highlights, as shown in the following table: The highlights are selected in a similar way by sending ordinal data to the highlight input. However, the equality between logic and ordinal data can be used to select the highlight on or off as shown.

Data	Highlight	Logic
0	Off	0
1	On	1
2	Flashing	n/a

It is not possible to display the flashing legend via a logic input; if a flashing highlight is required, an ordinal switch function block is necessary.

The following diagram shows a logic gate connected to a pair of operator station highlights. The inverted output of the gate is also used to drive a highlight.



It is possible to highlight potentiometer and meter function block names in a similar way. This can be useful to indicate that a particular meter is active, for example.

Operator Station Potentiometers and Meters

The potentiometer and meter function blocks allow any *SSD LINK* variable to be adjusted or displayed. As with any operator station function block, these function blocks are loaded into the display handler to determine their position on the display. These function blocks allow the variable to be formatted correctly and displayed in engineering units.

Using ConfigEd, the text on these function blocks can be changed to suit the application. Text can be highlighted as well, in a way similar to buttons and annunciators.

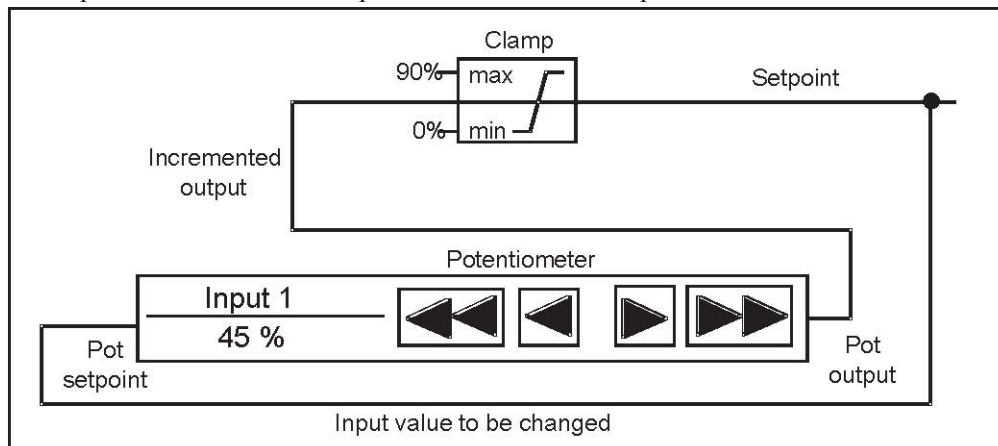
Potentiometers

Potentiometer function blocks are used to raise and lower an *SSD LINK* variable. There is a single output to change the connected variable and a single feedback input to display the variable. The feedback variable must be connected for the display to change.

Basic Use of a Potentiometer

Whenever a potentiometer function block is used, it is recommended that its output be clamped using the clamp function block. The clamp allows the variable to be restricted to allow for over-range as described in earlier sections.

The following shows a potentiometer with a clamp function block on its output.



It is important to note a potentiometer output does not store the variable to be changed. The value to be changed is the input of the function block connected to the potentiometer output. In this case, the potentiometer output simply increments or decrements the input to the clamp.

Messages from the potentiometer are sent only while the raise and lower arrows are pressed (just like buttons and annunciators). When the potentiometer buttons are pressed, the potentiometer takes the value of the setpoint, adds or subtracts a small amount, and sends this number from the output. Addition or subtraction depends on the raise or lower buttons, respectively, and the magnitude of the change depends on whether a “fast” or “slow” button is pressed.

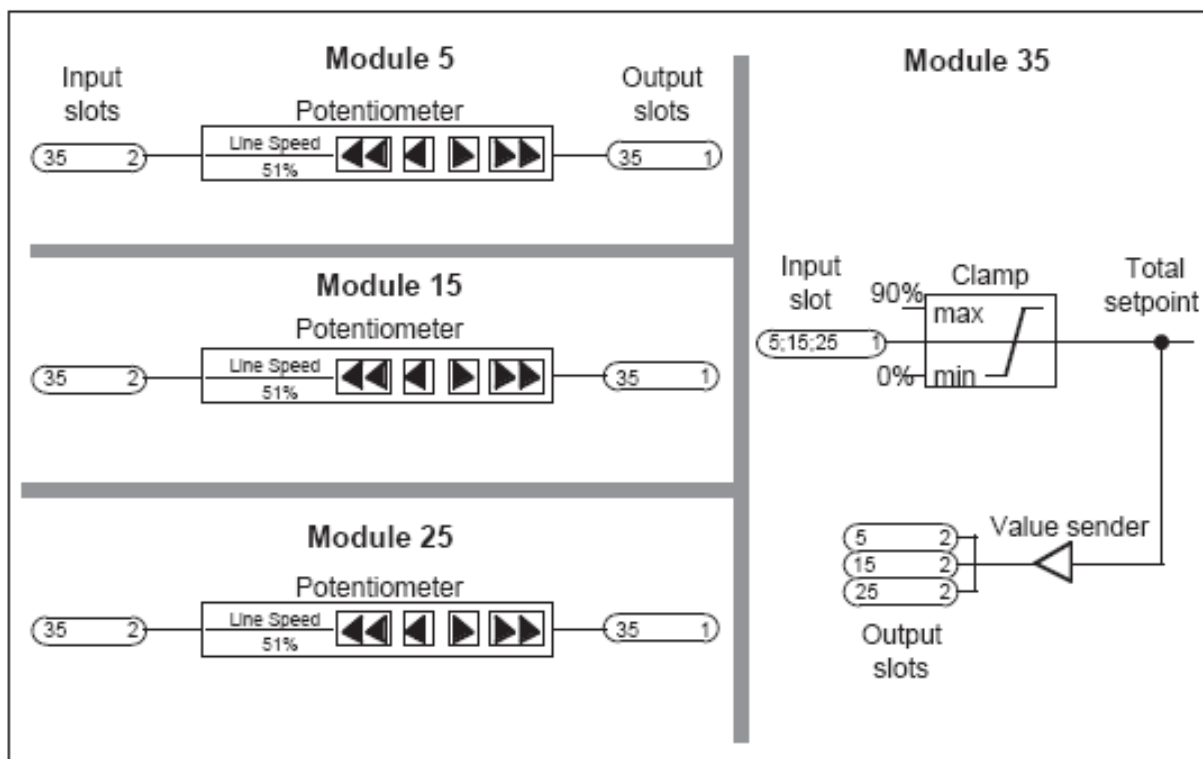
Since it is recommended that potentiometers be used with clamps, the clamp output should be used as the feedback to the potentiometer setpoint (as shown above). Since the input of the clamp is persistent, the current value of the potentiometer setpoint is stored even if the power is removed. If it is always required to power up to a known value (e.g., 0%), use an initializer function block to pre-set the clamp input.

In addition to the potentiometer function block, a dual potentiometer and meter function block is available which integrates the potentiometer and meter functionality into a single band.

Wire OR and Potentiometers

Potentiometer outputs can easily be wire OR’ed for multiple displays. Only one clamp should be used.

The diagram on the following page shows three potentiometers connected to a single clamp.



In this case, the last potentiometer to be adjusted will set the variable.

The various input and output slots are not shown in the diagram. Note the use of a sender to ensure the potentiometer setpoint is updated correctly. In fact, the value sender will send its data to input slots in three different modules (as shown). The three potentiometer outputs are connected to the same input slot within module 35. Module addresses shown are arbitrary.

Potentiometer Scaling (Gain)

Potentiometer function blocks must be scaled to allow the feedback variable to indicate the correct machine units. The maximum SSD *LINK* variable is ± 1.00 .

When the potentiometer is configured, it is scaled using the Gain parameter. This parameter is the on-display representation of 1.00. The total displayed number is referenced, neglecting any decimal points. The decimal point is set with the Format statement within the potentiometer. (e.g., if the full speed of a machine is 1000 feet per minute (FPM) this has to correspond to SSD *LINK* 1.00):

Gain = 1000

Format = ≈#### FPM (four digits, no decimal place)

The minus sign in the format statement indicates that ≈ is displayed in the potentiometer feedback when a negative number is referenced. If this is not present, the minus sign will not be displayed.

If the full speed = 100.0 FPM

Gain = 1000

Format = ≈####.# FPM (four digits, one decimal place)

If the full range SSD *LINK* variable is not 1.00, this must be taken into account in the potentiometer gain term to allow for any over-range.

$$\text{Gain} = \frac{\text{Full range required display number}}{\text{Full range SSD LINK number}}$$

This does not include the decimal point.

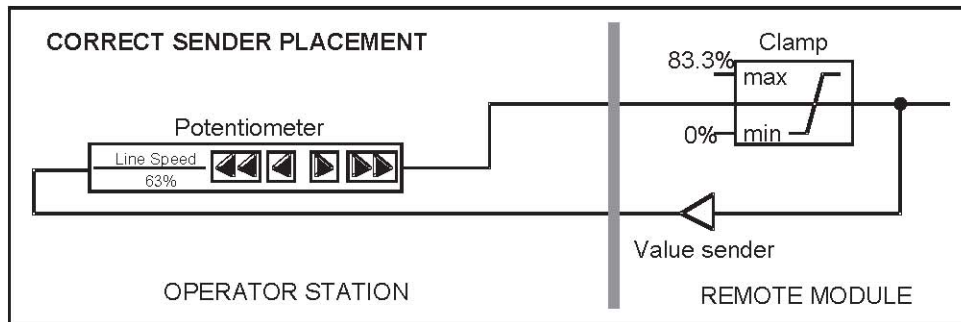
When scaling a drive, if the full speed setpoint is 0.833, and this has to correspond to an indicated top speed of 500.0 FPM:

$$\text{Gain} = \frac{500}{.8333} = 600$$

For this case, the clamp on the potentiometer output must be set at 83.333%. The Format would be set at ###.# FPM (four digits, one decimal point).

Senders and Potentiometers

It is important to use sender function blocks for certain variables going to remote modules. Since the potentiometer does not store the variable to be adjusted, the sender must be placed correctly at the clamp output.

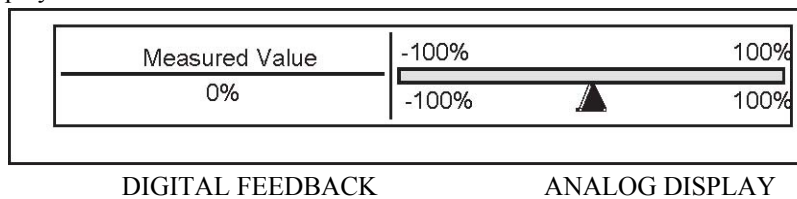


The clamp is located in a remote module to maximize operator station memory.

Meters

Meter function blocks display the connected SSD *LINK* variable in two ways on the operator station screen:

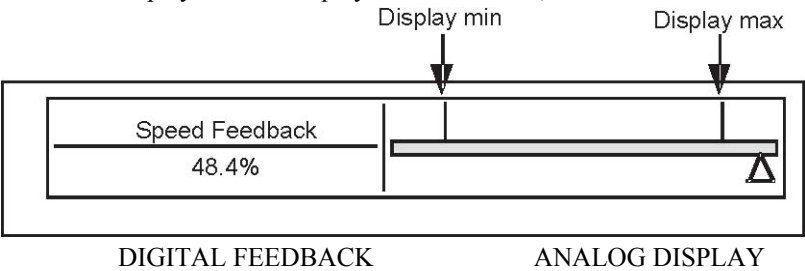
- Digital feedback variable;
- Analog trend display.



The meter feedback is connected to the measured value input. The digital feedback variable is scaled in exactly the same way as the potentiometer feedback variable, using the Gain parameter. The analog trend display has two set up parameters. These

specify the displayed full range of the metered input variable and the overload lines. The full range value in percent of the input variable should be entered into the Display Max and Display Min parameters.

The overload lines are set such that the values of Display Max and Display Min are used as full range deflection for these lines. When the analog arrows exceed the Display Max or Display Min line values, the arrows turn white.



For the example above, when the full range feedback for a reversing drive setpoint is 83.333%, the Display Max and Display Min are set to $\pm 83.333\%$. If the overload speed was 50% of drive full speed, the Max and Min values would be:

$$\text{Min and Max} = \frac{83.333 \times 50}{100} = \pm 41.666\%$$

If the drive was not reversing, the Display Min would be zero.

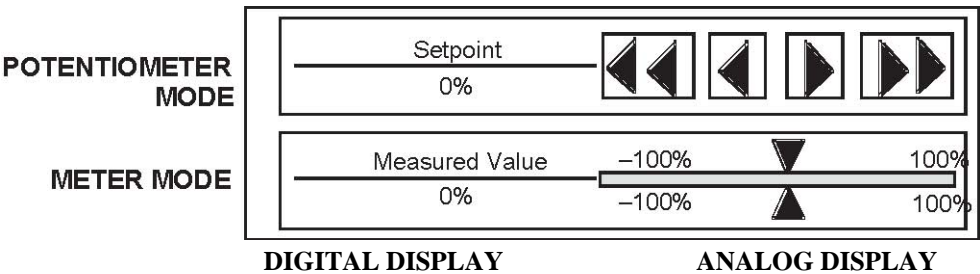
Pot+Meter Function Block

The Pot+Meter function block combines the functionality of separate potentiometer and meter blocks into one display band. There are two display modes, potentiometer mode and meter mode, selected by touching the Pot+Meter display.

The Pot+Meter has two feedback inputs:

- Setpoint for the potentiometer function;
- Measured value for the meter function.

In potentiometer mode, the digital display indicates the potentiometer setpoint. In meter mode, the digital display indicates the measured value and the analog display indicates the setpoint and the feedback variable.



Scaling of the Pot+Meter is performed in exactly the same way as the individual potentiometer and meter function blocks. Both the potentiometer feedback and the meter feedback have to be connected to allow the function block to display correctly.

Potentiometer and Meter Highlights

The text within the potentiometer and meter function blocks can be highlighted in a way similar to buttons and annunciators (e.g., to indicate a particular potentiometer is active). The highlight can be set to off, on, or blinking by sending ordinals to the highlight input.

L5203 SERIAL MODULE

The L5203 is a multipurpose module which provides RS232 and RS422 serial communications between the SSD *LINK* network and third party equipment. As the L5203's serial ports are not electrically isolated from its power supply, care must be taken to prevent ground loops when connecting to other non-isolated ports.

One option is to power the L5203 from an isolated power supply. Another solution would be use a serial line isolator. One such device, the D240 series from SSD Process Automation, provides a high degree of electrical isolation and offers a communications watchdog as well.

Two general applications to be discussed in the following sections are:

- Process-control and monitoring;
- Event/fault logging and diagnostics;

Process Control and Monitoring

For access to process variables, the L5203 supports the BiSynch protocol in the following forms:

- Binary Master;
- Binary Slave;
- ASCII Slave.

When the L5203 is configured in bisynch slave mode, it (like all bisynch slaves) is controlled by a single master device (usually a supervisory computer or PLC). When the L5203 is configured in bisynch master mode, it controls the bisynch network and commands all slaves. For any given bisynch network, there can be only one master device.

The primary difference between the ASCII and binary modes is that ASCII, being a simpler protocol, requires less complex drivers while the more complex binary protocol provides much higher throughput.

For detailed technical information on the bisynch protocol, refer to the SSD Drives document HP022047. For guidelines in developing bisynch supervisory drivers to access the L5203 in slave mode, refer to document package HR352424. Finally, information on configuring the L5203 in EI bisynch applications can be found in document HR351406 and in the SSD *LINK* function block data sheets.

Communications Speed

The EI bisynch protocol transfers information at a relatively slow rate, frequently 9600 bps, yielding approximately one parameter every 20mS. While the L5203 can communicate up to 57600 baud, this speed must be downgraded to the highest common rate supported by all devices on the bisynch network, typically 19200 baud. At any rate below this, EI bisynch should not be used for even medium-speed control parameters.

Further, SSD Drives recommends that closed-loop feedback and cascaded line drive setpoints not be transferred via bisynch; if they are, it will be nearly impossible to maintain accurate drive control while ramping. Similarly, high-speed logic signals such as drive start and jog are often inappropriate for transfer via bisynch.

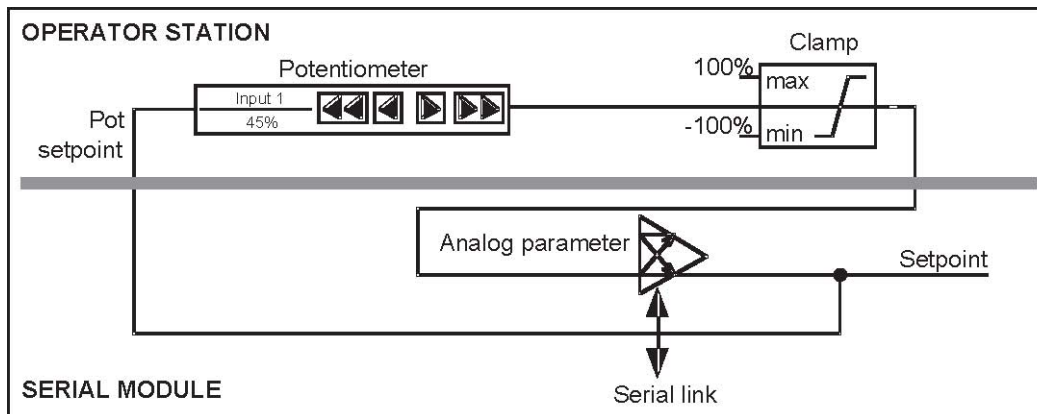
In general, EI bisynch is suitable for:

- Transferring setup parameters (setpoints, recipes, etc.);
- Transferring low to medium speed Logic signals (forward/reverse mode, etc.);
- Monitoring and diagnostics at low speeds.

Both the baud rate and the size of the parameter set affect throughput, and some allowances can be made under the binary protocol if high baud rates can be used to transfer only a few parameters. Thorough testing of functionality and performance is always a must.

Distributed Parameters

The L5203 module is often used to allow supervisory computers or PLCs to set system parameters which are also accessed using L5102 operator stations. The following diagram shows how to properly connect serial parameters in systems where operator station potentiometers are also used.



When dealing with distributed parameters, the guiding principle is to ensure a parameter's data is stored in only one place. In the above example, the actual data is stored only in the serial parameter function block. The real setpoint sent to the rest of the system comes from there, *not* from the potentiometer or the clamp.

Note that in this example, the clamp only affects the setpoint as adjusted by the operator at the operator station. It does not limit the setpoint coming from the supervisor. If that is required, a second clamp would be placed on the immediate output of the parameter. Both clamps are optional.

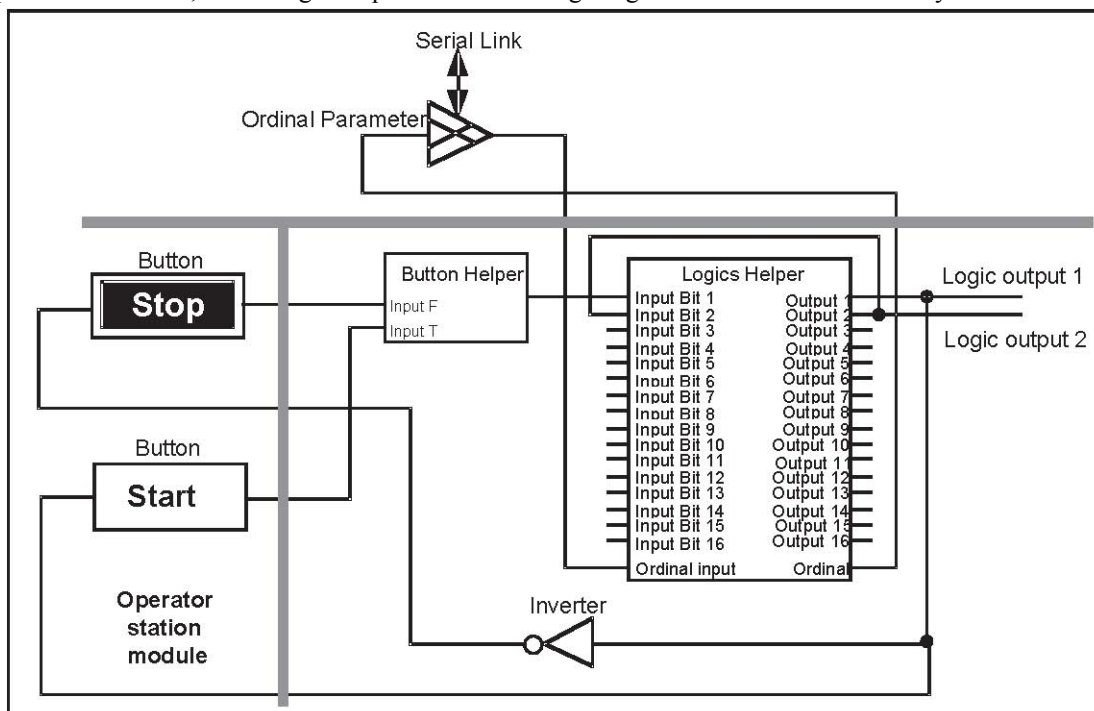
Ordinal Parameters

Ordinals in SSD *LINK* are unsigned whole numbers and ordinal parameter function blocks are used solely to transfer them between bisynch and SSD *LINK* networks (e.g., no scaling or translation is performed).

Logic Signals

Using the bisynch protocol, logics are transmitted as 16-wide bitfields using ordinal parameters. The Logics Helper function block is used to split and combine the individual bits of the ordinal. Refer to the Logics Helper data sheet for more information.

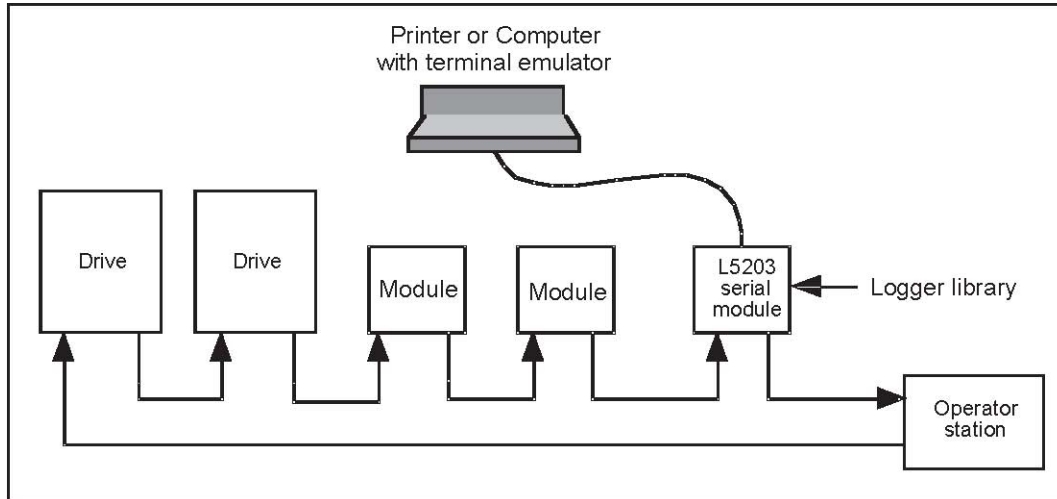
As was the case with analog data, care must be taken when the setting of logic signals is to be distributed. This example shows logic output 1 being distributed between the supervisor and two separate momentary push buttons on an operator-station. Again, the key is that the state of the bits is kept only in the parameter. Its output goes directly to the ordinal input of the Logics Helper function block, whose logic outputs are the real logic signals used elsewhere in the system.



This example also demonstrates another important helper function block: the button helper. Its function is to send its logic output continuously based on the once-per-press message received from the button. When neither button is pressed, no output is generated or sent.

Event/Fault Logging and Diagnostics

The L5203L can be configured as an output-only data logger. The data protocol is 9600N81 ASCII CR/LF text intended for capture by a serial printer or terminal emulator.



The L5203L Logger configuration will generate:

- Description of all detected network exceptions;

- Configurable message in response to an event;
- Configurable message in response to a logic edge or message;
- Configurable message in response to a value message;
- Configurable message in response to an ordinal message.

Description of detected network exceptions is the logger's default behavior and no setup beyond that for the serial port is needed. The L5203L logs module state changes throughout the system, whether during startup, shutdown, reconfiguration, failure, or when even transient fiber optic problems occur. Transient problem detection is very effective in isolating bad fiber optic cable runs, as the message logs the exact address of the module detecting the poor signal at its receiver.

The remaining functions are used by creating and configuring the appropriate function blocks and connecting standard input connectors to them. Note that unlike the exception logger which receives globally-addressed information sent by default by all modules, the event and data loggers must receive their inputs via standard connections. This necessitates some configuration at the source as well.

All logging function blocks may be configured to include a time stamp in their text. Current SSD *LINK* modules have a sense of elapsed time since their last start-up, which is very useful in recording the temporal relationship in the logged events.

For more information on the L5203L Logger, see the function block data sheets.

FILTERS

There are three types of filter function blocks available within the function block library: the simple filter, filter, and decimation filter. Each has a particular application.

Simple Filter

The simple filter takes the smallest memory space and executes when a message is received on its input.

This filter has a time constant which is dependent on the input sample time; it is required that input messages are sent to the input at a regular rate. The filter does not keep track of elapsed time. The filter input should be supplied from a constant sampling source, such as an analog input or a sampler.

The filter equation is $K = e^{-(dt/t)}$. Here, K is the filter K value, dt is the sample time and t is the time constant. The following table shows typical values of K for a sample time of 100mS: Use: Where a constant event rate is present, and it is not required to set the time constant directly.

Time Constant (in seconds)	K Value
0.5	0.820
1.0	0.900
1.5	0.936
2.0	0.950
2.5	0.960
3.0	0.967

Filter

This is the most complex filter, and takes the most memory capacity. Except for keeping track of time between input events, it is exactly the same as the simple filter. The filter time constant is a set up parameter.

Use: Where memory requirements are not important, and it is required to set the time constant directly. Even though the filter does take account of time between trigger events, triggering still should take place at reasonably regular intervals.

Decimation Filter

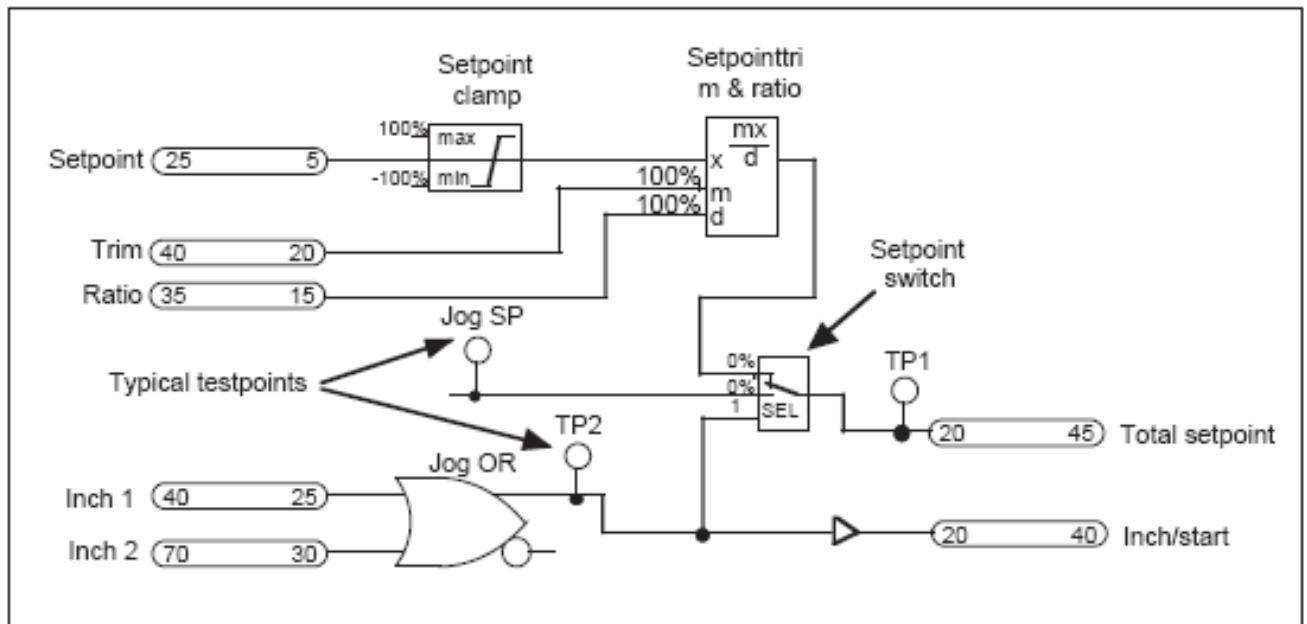
This is an averaging filter. The output is the arithmetic mean of a specified number of input events. The output is updated when the specified number of input events has taken place. This results in a reduced output event rate.

Use: Where the input event rate must be reduced to prevent module overload and an error of $\pm 0.1\%$ in output is acceptable with large decimation values. The decimation filter is particularly useful for operator stations.

TEST POINTS

Test points are used to simplify the use of SAM when configurations are to be monitored and set up. Test points are added when the modules are configured.

The following diagram shows a typical block diagram with test points (TP 1, TP 2, Jog SP) added to commonly used points within the configuration.



The actual test point text is stored within the module configuration memory, allowing SAM to access the test points without having the configuration files on the same computer. In order to achieve full SAM access to all function block information, the up-to-date configuration files must be present.

To minimize module memory requirements, the test point text should be kept short. Test points could be allocated within the two page boundary of free configuration space; if the configuration files become too large in the future, the test points could be deleted.

When SAM is used, the test points can be used to set and monitor a variable. When test points are defined, they may be read-only or read/write. This allows the use of selective test points while preventing unauthorized users from making changes.

This page intentionally left blank.

Appendix D ROM Issues

The firmware within an SSD *LINK* module is given a Module Type ID, commonly referred to as the ROM number. Standard system modules are either ROM 12 or 13. ROM 13 is the later release and most SSD *LINK* modules use this version.

The main differences between ROM 12 and ROM 13 are:

- ROM 13 modules process connections faster than ROM 12 modules;
- ROM 13 modules have approximately 10% more free configuration memory;
- ROM 13 uses an improved “startup override” system. ROM 12 modules power up in a halted state if an active connection was sensed at the RTNX port;
- Timebase function blocks may be used in all ROM 13 modules while certain ROM 12 modules (like the L5201 Analog) cannot run them. ConfigEd alerts users when attempting to install a configuration containing timebase function blocks into a ROM 12 module.

L5101 and L5102 operator stations are manufactured using ROM 12.

BACKUP AND RESTORE

The SSD *LINK* tools support a backup (.bku) file which SAM can restore into a spare module to duplicate the original functionality and parameter settings. This means:

- Customers do not have to stock spares with the same ROM issue as the modules in use; as long as the spare module’s ROM number is equal to or greater than the original module’s, the SAM restore function will work.
- Successive ROM issues offer new and better functionality. For example, ROM 13 provides improved performance in less space than ROM 12.

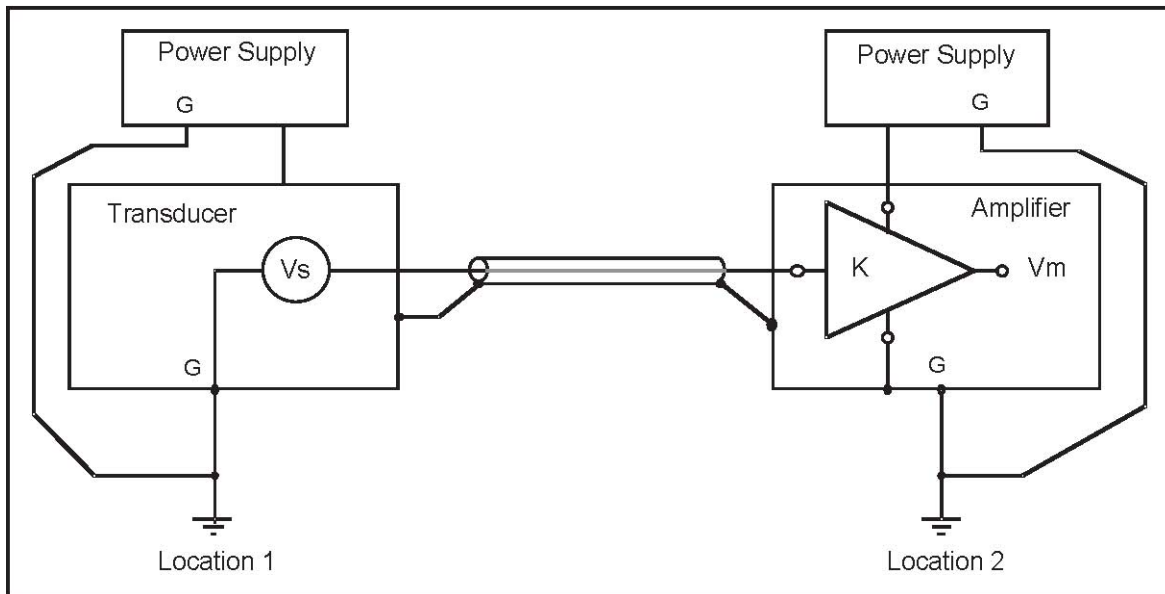
SAM will prevent restoration of a ROM 13 backup file into a ROM 12 module.

This page intentionally left blank.

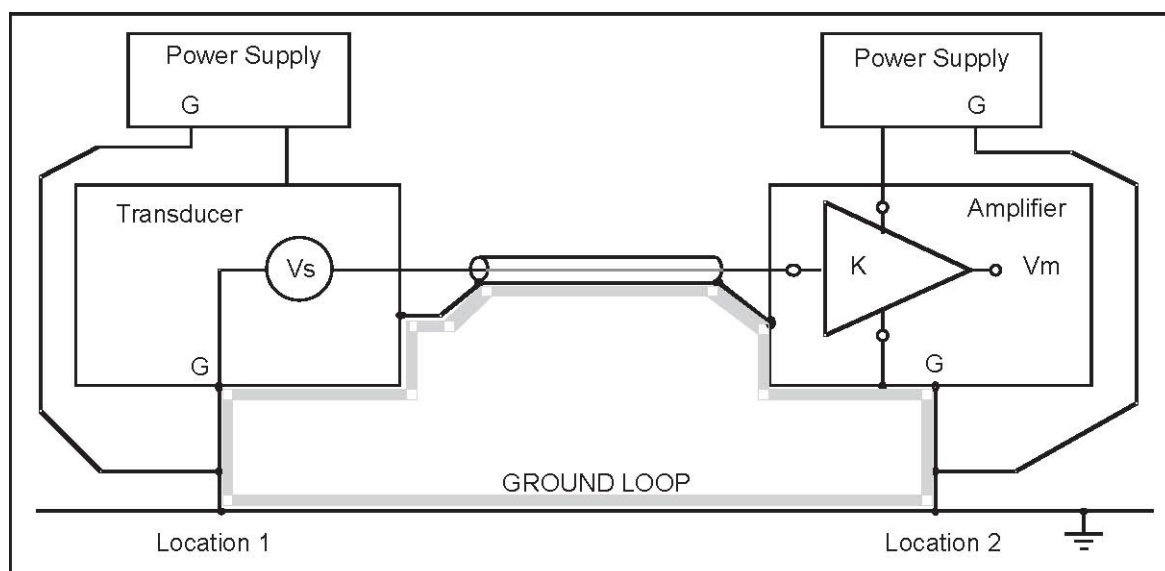
Appendix E Grounding Issues

GROUND LOOPS

A ground loop is caused by ground wiring which allows ground currents to flow by more than one path between two or more grounded points. The result is circulating interference currents caused by magnetic pick-up by the wiring loop. These currents will induce noise voltages onto adjacent signal wires and result in signal degradation.

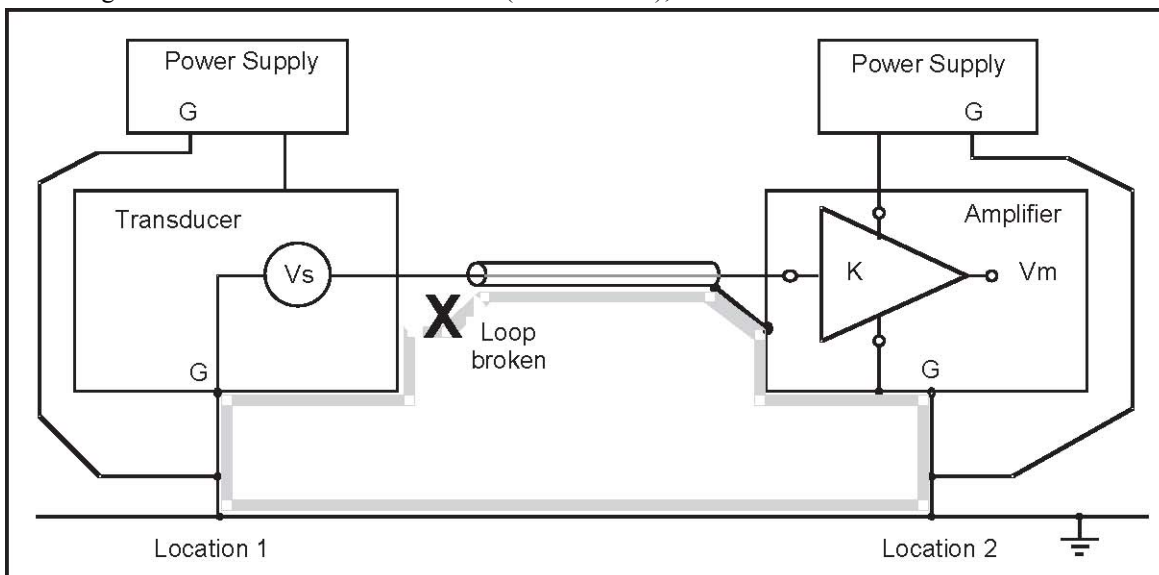


The diagram above shows a signal V_s generated at Location 1, transmitted through shielded cable, and measured as V_m at Location 2. This circuit should realize the function $V_m = K \times V_s$. This situation is typical of a load-cell or dancer position signal being generated at a machine and transmitted to a main equipment enclosure for processing. Two ground paths exist in this situation, one from the ground at Location 1, to the transducer ground, through the shield, to the amplifier ground, back to ground at Location 2, and a second from ground at Location 1 through the ground connection to Location 2. This second path is normally not shown in system electrical diagrams, and can be easily overlooked. The ground symbols used above imply this connection by showing that Locations 1 and 2 are tied to a common reference potential, as shown in the following diagram.

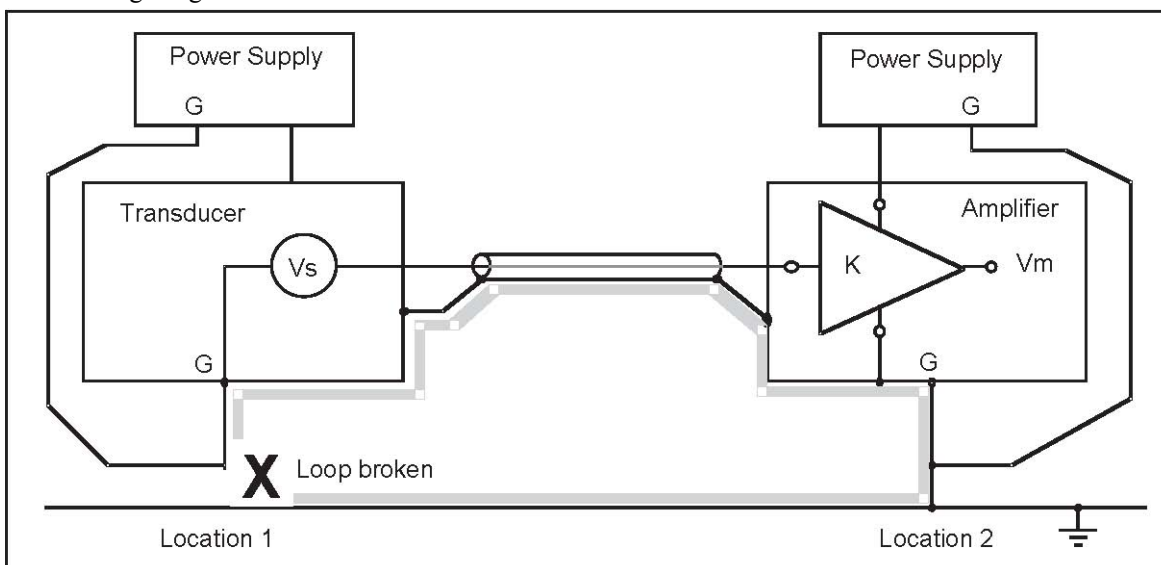


The physical connection will normally consist of a ground wire linking the two locations. The ground-loop allows extraneous magnetic fields to generate currents that will circulate through the loop, thus flowing through the signal shield. This interference will induce noise voltages onto the shielded signal, which degrades the signal accuracy. To eliminate the interference, the loop must be broken. This can be accomplished by either:

- Removing one of the connections at the shield (shown below);



- Or removing the ground connection at the transducer:

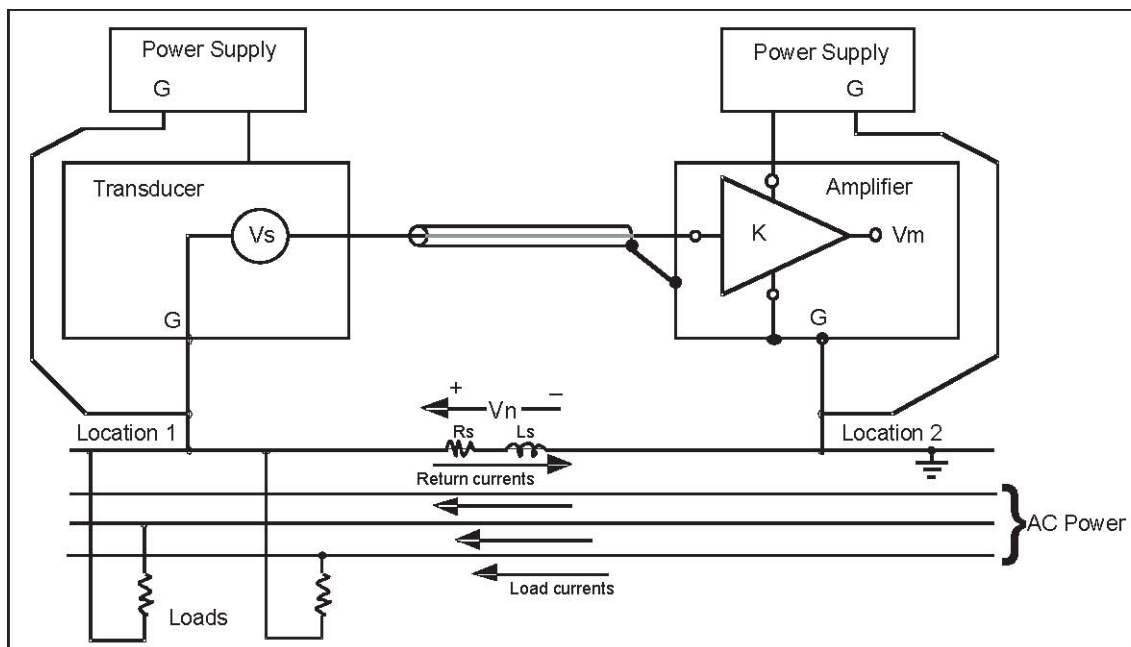


In the latter case, we will force any ground currents between the two locations to flow through the shield. If these currents are significant, this option may not help to reduce interference to the shielded signal.

CROSS-COUPLED GROUNDS

Cross-coupling occurs when a length of ground-conductor is common to two or more circuit loops. Any conductor has an associated series resistance and inductance, and a current flowing in the conductor will generate a voltage. The problem arises from ground currents generated by one circuit causing voltages to appear in series with other signals, thus degrading them.

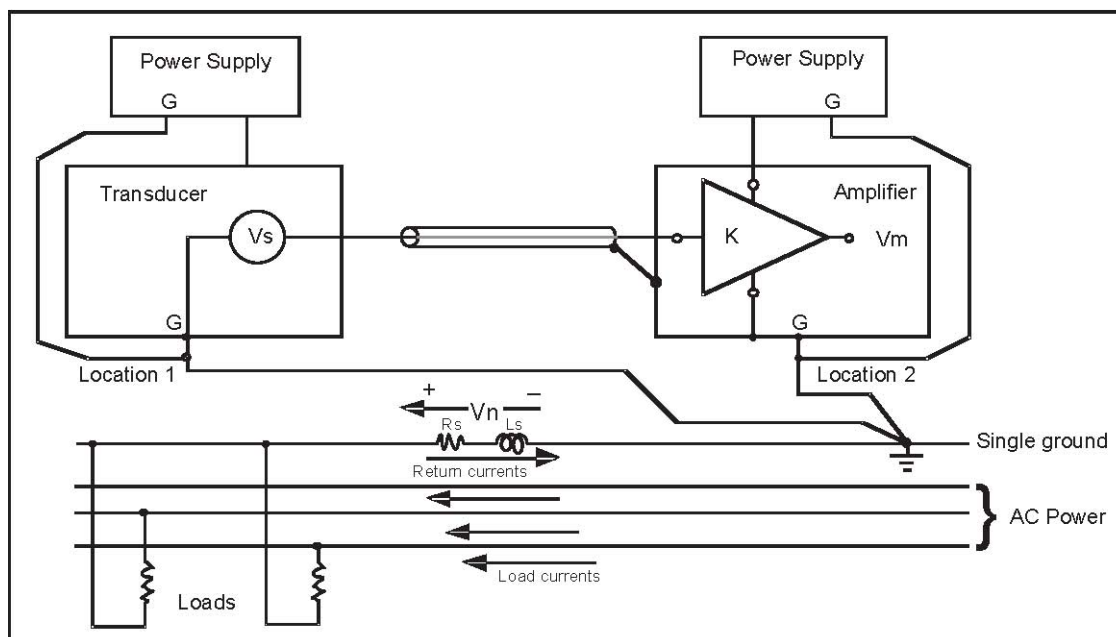
The following diagram shows an example where an AC ground return is shared by the transducer and amplifier circuits' ground line.



Such an arrangement is highly undesirable, but could arise from careless or unplanned system wiring. The noise voltage V_n , generated by the AC return currents flowing through the impedance represented by R_s and L_s , appears between the ground connections at locations 1 and 2. This will offset the signal V_s relative to the amplifier input, and we will therefore find $V_m = K \times (V_s + V_n)$. This shows that the amplifier output is corrupted, since V_n is not part of the signal we want amplified.

One solution to this problem is to maintain a single-point ground. The aim of this scheme is to provide separate ground-returns for each circuit, thereby eliminating shared conductors and ground-loops. It is very effective in reducing noise due to low-frequency ($< 1\text{MHz}$) noise sources.

The diagram on the next page shows the additional ground-wires necessary to provide separate grounds to each major section of the system.



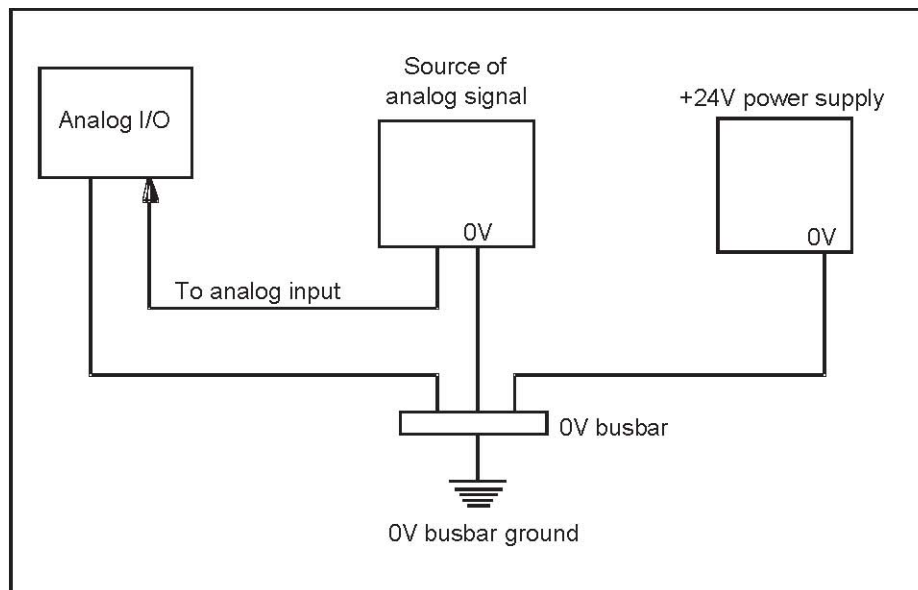
The voltage V_n is no longer part of any circuit involving the transducer and amplifier elements, so the amplifier output returns to $V_m = K \times V_s$. Note that this diagram does not show strict single-point grounding, since the system still contains some

lengths of shared ground-conductors. The transducer and its power supply share a ground between Location 1 and the grounding point, and similarly for the amplifier and its supply.

The two loads shown on the AC power bus also share a ground return. For most systems, however, the main concern is to provide separate grounds for subsystems to avoid crosscoupling between AC and DC sections, power and signal sections, and analog and digital sections.

USE OF A 0V BUSBAR TO CONNECT POWER SUPPLIES

With the SSD *LINK* system, the signal 0V and the +24V PSU 0V must be the same. The +24V 0V is normally implemented with a busbar as shown in the diagram on the following page. The busbar is normally grounded per standard wiring practice.



Sales Offices

Australia

Parker Hannifin
Pty Ltd
9 Carrington Road
Private Bag 4, Castle Hill
NSW 1765
Tel: +61 2 9634 7777
Fax: +61 2 9899 6184

Belgium

Parker Hannifin SA NV
Parc Industriel Sud Zone 11
23, Rue du Bosquet
Nivelles B -1400 Belgium
Tel: +32 67 280 900
Fax: +32 67 280 999

Brazil

Parker Hannifin Ind. e Com. Ltda.
Av. Lucas Nogueira Garcez, 2181
Esperança - Caixa Postal 148
Tel: +55 0800 7275374
Fax: +55 12 3954 5262

Canada

Parker Motion and Control
160 Chisholm Drive
Milton
Ontario L9T 3G9
Tel: +1(905)693 3000
Fax: +1(905)876 1958

China

Parker Hannifin Motion & Control
(Shanghai)
Co. Ltd.
SSD Drives
280 Yunqiao Road
Export Processing Zone
Pudong District
Shanghai 201206
P.R.China
Tel: +86 (21) 5031 2525
Fax: +86 (21) 5854 7599

France

Parker SSD Parvex
8 Avenue du Lac
B.P. 249
F-21007 Dijon Cedex
Tel: +33 (0)3 80 42 41 40
Fax: +33 (0)3 80 42 41 23

Germany

Parker Hannifin GmbH
Von-Humboldt-Strasse 10
64646 Heppenheim
Germany
Tel: +49(0)6252 798200
Fax: +49(0)6252 798205

India

SSD Drives India Pvt Ltd
151 Developed Plots Estate
Perungudi,
Chennai, 600 096, India
Tel: +91 44 43910799
Fax: +91 44 43910700

Italy

Parker Hannifin SPA
Via Gounod 1
20092 Cinisello Balsamo
Milano
Italy
Tel: +39 02 361081
Fax: +39 02 36108400

Singapore

Parker Hannifin Singapore Pte Ltd
11, Fourth Chin Bee Rd
Singapore 619702
Tel: +65 6887 6300
Fax: +65 6265 5125

Spain

Parker Hannifin (Espana) S.A.
Parque Industrial Las Monjas
Calle de las Estaciones 8
28850 Torrejonde Ardoz
Madrid
Spain
Tel: +34 91 6757300
Fax: +34 91 6757711

Sweden

Parker Hannifin AB
Montörgatan 7
SE-302 60 Halmstad
Sweden
Tel: +46(35)177300
Fax: +46(35)108407

UK

Parker Hannifin Ltd.
Tachbrook Park Drive
Tachbrook Park
Warwick
CV34 6TU
Tel: +44(0)1926 317970
Fax: +44(0)1926 317980

USA

Parker Hannifin Corp.
SSD Drives Division
9225 Forsyth Park Drive
Charlotte
North Carolina 28273-3884
Tel: +1(704)588 3246
Fax: +1(704) 588-3249

Your local authorized Parker distributor

© 2009 Parker Hannifin Corporation. All rights reserved.

RG350678 (Iss3 10/2009)



Parker Hannifin Corporation
SSD Drives Division
9225 Forsyth Park Drive, Charlotte,
North Carolina 28273-3884
Tel: (704) 588 3246 Fax: (704) 588 3249
sales.us.ssd@parker.com
www.parker.com www.ssddrives.com