Laser Align

Reference

Laser Align

Reference

CyberOptics

CyberOptics Corporation
5900 Golden Hills Drive
Minneapolis, Minnesota
55416 USA

World Wide Web:
http://www.cyberoptics.com

USA ☎ (763) 542-5000
USA FAX (763) 542-5100

INTERNET:
info@cyberoptics.com
service@cyberoptics.com

For Customer Service and Support, call 800-526-2540 (USA)

Laser Align is a measuring device intended for use in an industrial or a laboratory environment. Use of this equipment in a manner not specified in the operating instructions may impair safe operation.

Laser Align Reference
January 2001
Part Number 7900176, revision B

Contents

Chapter 1

Overview

Introducing Laser Align

Laser Align is a non-contact sensor system that measures component position and orientation. It is designed to be integrated into the placement head of a pick-and-place machine.

Laser Align measures a component as the placement head transports it to the circuit board. The pick-and-place machine uses the measurement information to correctly orient the component for accurate placement on the board.

Laser Align eliminates the time-consuming step of positioning the component over a vision system to determine component orientation. Other benefits include:

- Laser Align provides accurate, non-contact sensing of angular orientation (θ) and X and Y lateral position.
- It measures components during transportation to the circuit board, which increases efficiency.
- It does not interfere with the placement operation.

Laser Align measures the position and orientation of a broad array of electrical components, including chip capacitors, MELF resistors, SOIC packages, QFPs, BGAs, and J-lead devices.

Laser Align Components and Configuration

The Laser Align system contains three different types of components:

- Hardware
- Firmware
- Software

This section provides an overview of these components.

Hardware

Figure 1-1 shows the main hardware components of the Laser Align system in a typical configuration with a host system.

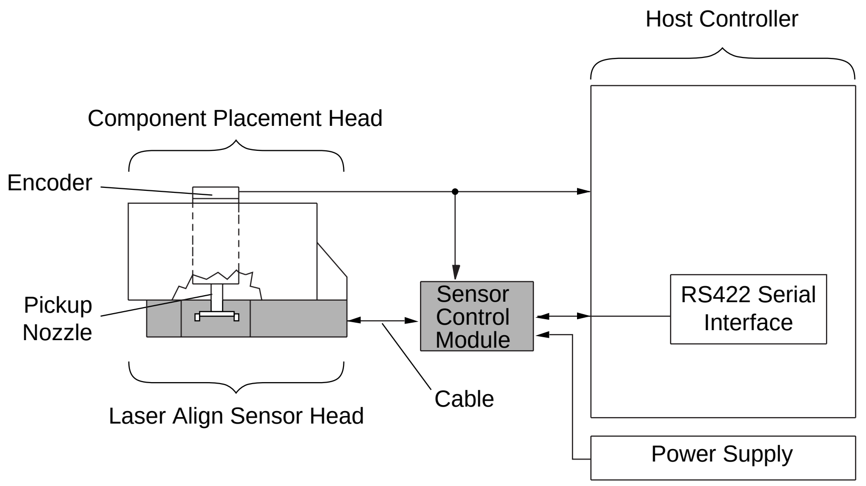
The CyberOptics Laser Align hardware components are:

- Sensor Head
- Sensor Control Module (SCM)
- Cable (coaxial)

The customer system components are:

- Host Controller
- RS422 Serial Interface
- Component Placement Head
- Pickup Nozzle
- θ Encoder
- Power Supply

Figure 1-1: Overview of Laser Align System



The *host controller* communicates with the Laser Align *Sensor Control Module (SCM)* via the *RS422 Serial Interface*. The host controller can instruct the SCM to initiate a measurement using one or more Laser Align measurement algorithms. Then the host controller begins component rotation.

After the component has been rotated and measured, the host controller can instruct Laser Align to report the results. The SCM:

- Gathers the component angle and shadow information collected by the *sensor* and *encoder*
- Calculates the alignment angle, center position, and width of the component
- Transmits the results back to the host controller

➤ See Chapter 2 for more information about Laser Align hardware.

➤ See Chapter 8 for more information about the measurement algorithms.

Firmware

The SCM is programmed with firmware that controls Laser Align operations and communication with the host controller and encoder.

See Chapters 7-10 for details about the firmware commands and codes.

Software

There are three MS-DOS software programs included with Laser Align that are used for sensor setup and diagnostics:

- **COM_422** is a program that facilitates RS422 communications between a DOS-based PC and the SCM.
- **COMTEST** is a program that allows you to interactively send commands from a DOS-based PC to Laser Align, one byte at a time. It also displays Laser Align responses.
- **LA_View** is a program that verifies the proper working condition of the Laser Align sensor, and allows you to view a visual representation of the laser image.

See Chapter 11 for more information about Laser Align software.

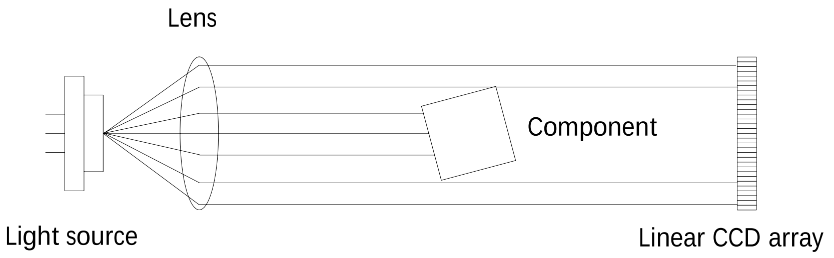
Laser Align Operation

Laser Align determines the position and orientation of a component so that it can be accurately placed by a pick-and-place machine. It projects a light stripe onto a *CCD* (charge-coupled detector) array, then analyzes the shadow that is cast as the component is rotated in the light stripe (Figure 1-2).

Principle

Figure 1-2 illustrates how Laser Align determines component position and orientation. A thin stripe of light is projected onto the component being aligned. It may be projected either onto the component body or onto its leads. A linear CCD array detects the shadow cast by the component.

Figure 1-2: Principle for determining component orientation



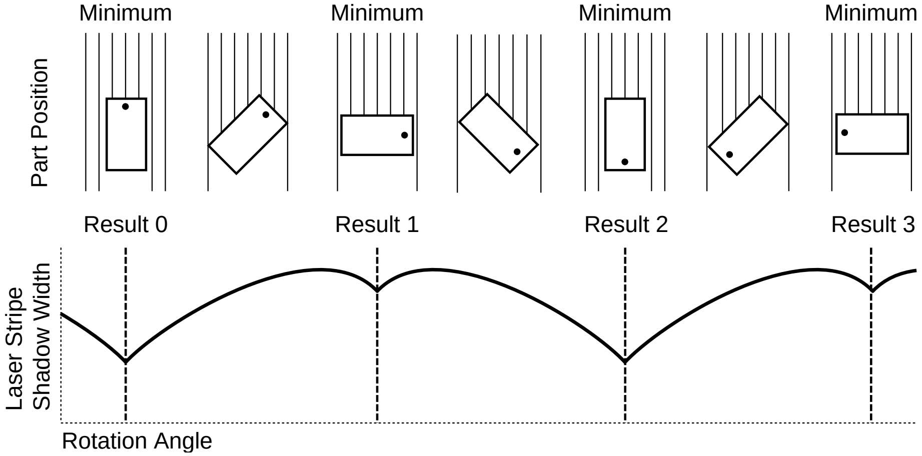
As the quill of the placement head rotates the component, the shadow cast on the CCD varies. When the component edges are aligned parallel to the light stripe, the shadow width is at a minimum.

Normally, there are four minimums in a 360° rotation. Laser Align reports results for up to four of these minimums.

Each minimum provides a θ *(theta) alignment angle* for the component.

Figure 1-3 shows the relationship between component position in the light stripe and the width and angle of the component shadow. It also identifies the points where the measurement results for each minimum are calculated.

Figure 1-3: Component Rotation and Results



For each minimum you request, Laser Align reports the encoder angle (alignment angle) at which the minimum shadow width occurs. It also reports the *center* and *width* of the shadow at the minimum.

If a component extends beyond the limits of the CCD array, you can program Laser Align to calculate the alignment angle based on the varying edge position of the component shadow, rather than the shadow width. If the component edge position is used in this manner, Laser Align will report the *edge* rather than the center of the component to indicate component position.

The results information—angle, center (or edge), and width for each minimum requested—can be used with information on nozzle position and component size to properly position the component.

Measurement Process Overview

The general steps in determining component position and orientation using Laser Align are:

1. The host controller initializes Laser Align using Laser Align commands to identify the measurement parameters for the type of component to be placed.
2. The host controller begins a pickup sequence that positions the component in the sensor light path.
3. The host controller sends the measurement command to Laser Align.
4. The host controller rotates the component while Laser Align collects the measurement information previously specified using Laser Align commands.
5. The host requests measurement data from Laser Align at the conclusion of rotation.
6. Laser Align returns the requested information (e.g., alignment angle, center (or edge), and width).
7. The host controller uses the results to accurately place the part, making any corrections required to compensate for component pickup errors.

Implementing Laser Align

There are three phases involved in successful Laser Align implementation:

- Setup
- Integration
- Operation

Setup

During setup, the Laser Align sensor, SCM, and software are unpacked and installed. If this is your first Laser Align implementation, you normally install into a test system first. If you are using Laser Align in a system where its use is already defined, you may install directly into your pick-and-place machine and controller.

Whether you install into a test system or production system, two critical procedures following installation are:

- Establishing and testing RS422 communications between the host controller and Laser Align
- Establishing the link between the machine encoder and Laser Align

Note Laser Align will not operate properly if these procedures are not completed successfully.

See Chapter 4 for more information about Laser Align setup.

Integration

During integration you identify the operating parameters that Laser Align will use to synchronize with the host system. You also identify the parameters and operational controls that will be used to perform, report, and use measurements. Then, you develop the programs required for successful implementation in your system.

There are several procedures involved in Laser Align integration:

- Setting general system parameters
- Synchronizing with the encoder
- Calibrating pickup nozzle position
- Calibrating theta (θ) position
- Determining the measurement position of the component
- Identifying measurement parameters for the component
- Identifying reporting requirements
- Determining component rotation requirements
- Defining courses of action based on measurement results

The choices that you make during integration depend upon your system requirements. However, if Laser Align is not correctly integrated, you may receive unexpected measurements, or you may receive error indications instead of measurements.

➲ See Chapter 6 for more information about Laser Align integration.

Operation

This phase involves the actual, day-to-day use of Laser Align. The key procedure in this phase is performing measurements using Laser Align.

During the measurement process, the host controller:

- Uses **LOAD_CMD (40)** to send the measurement parameters identified during integration.
- Uses **SWEEP_CMD (30)** or one of its alternatives to request that the measurement be taken.
- Uses **REPORTW_CMD (42)** to request that Laser Align report the information identified during integration.
- Uses the courses of action defined during integration to handle the measured part, based on the measurement results.

As part of Laser Align operation, you may need to monitor and adjust your Laser Align setup. These procedures include:

- Monitoring image quality
- Resetting the firmware
- Resetting the hardware (only some SCM2 models permit this)

See Chapter 6 for more information about Laser Align operation.

See Chapter 7 for more information about the Laser Align commands.

Chapter 2

Hardware

Overview

There are three CyberOptics components in the Laser Align system:

- Laser Align Sensor
- Sensor Control Module (SCM)
- Coaxial Cable

See Laser Align Components and Configuration in Chapter 1 for general information about how Laser Align is integrated into a pick-and-place system.

See Chapter 4 and Chapter 5 for detailed information about Laser Align setup and integration.

Laser Align Sensor

The sensor head contains:

- Light source
- Optics that generate the stripe of light
- Optical filters
- Linear CCD array (detector)
- EEPROM that stores calibration information

Communications and power for the sensor are provided by a single coaxial cable to the Sensor Control Module (SCM).

Where applicable, the laser diode power measured in the slot of the sensor head is below IEC 825 and CDRH Class I power levels.

See Chapter 3 for information on laser safety.

Sensor Models

There are five Laser Align sensor models:

- Low Profile Laser Align
- Focused Modular Laser Align (with side connector)
- Focused Modular Laser Align (with top connector)
- MiniLAM
- High Density Laser Align (LAHD)

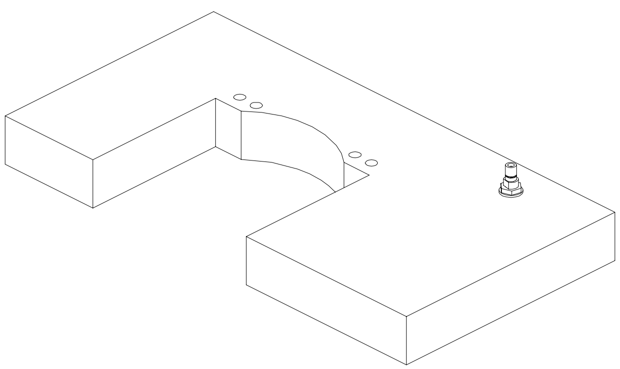
All Laser Align sensors operate in the same manner, but differ in size and configuration. This allows you to choose the best model for your specific needs.

See Chapter 14 for technical specifications for the different sensor models.

Low Profile Laser Align

This sensor is designed for use in systems with limited clearance in the vertical (Z-axis) direction. It handles parts up to 30 mm (1.18 in) in size

Figure 2-1: Low Profile Laser Align.



Focused Modular Laser Align

This sensor is designed with a small footprint in the X- and Y-axes, and can have either top or side connectors. This makes it particularly useful when the sensor must be retrofitted into a machine. Like the Low Profile model, it measures parts up to 30 mm (1.18 in) in size.

Figure 2-2: Focused Modular Side Connector Model

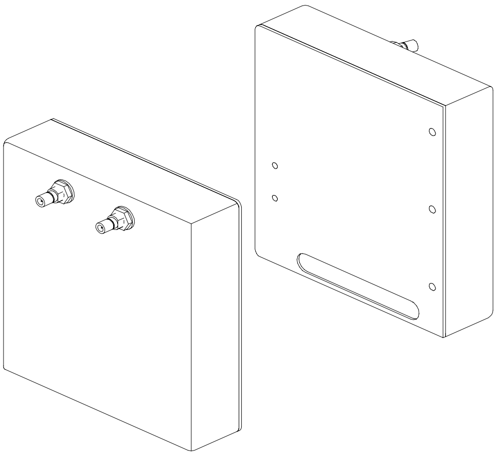
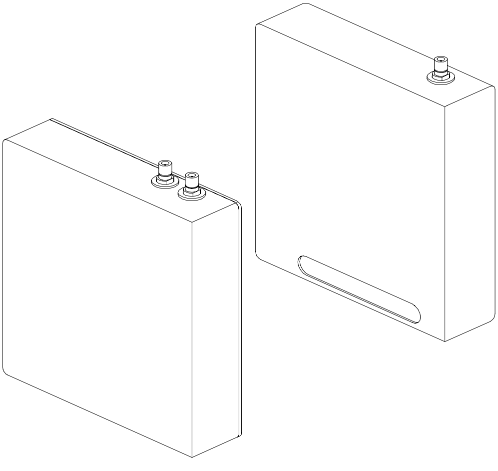


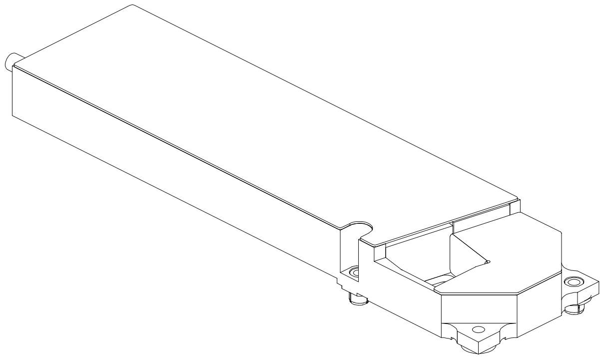
Figure 2-3: Focused Modular Top Connector Model



MiniLAM

This sensor is designed especially for systems that measure small parts; parts 10 mm (0.39 in) or smaller.

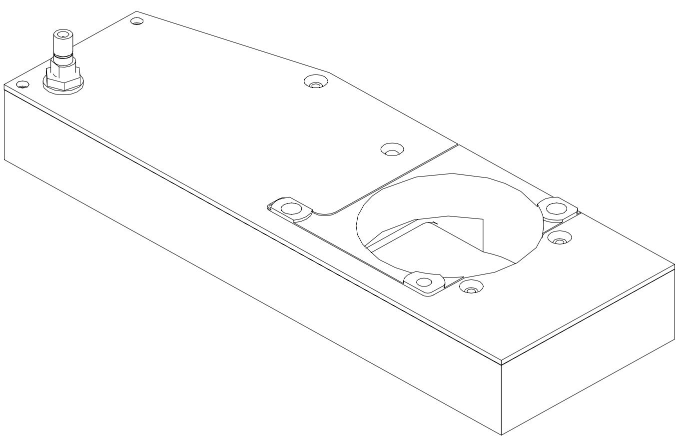
Figure 2-4: MiniLAM



High Density Laser Align (LAHD)

This sensor provides mid-range functionality, measuring parts that are 20 mm (0.79 in) or smaller.

Figure 2-5: High Density Laser Align



SCM (Sensor Control Module)

The SCM:

- Receives data and timing information from the Laser Align sensor head
- Receives input from the encoder
- Stores and processes the video image
- Handles communication with the host controller

There are two basic models of the SCM:

- **SCM1 -** stores the Laser Align application firmware in a single EPROM (erasable programmable read-only memory); upgrades are made by changing the EPROM. Distinguishing characteristics of the SCM1 include a second, smaller board (sometimes called a *piggyback board*) that is mounted on the SCM1, as well as a large 40-pin connector located near the power connector and three diagnostic LEDs. This board is no longer in production (Figure 2-6).
- **SCM2 -** stores the Laser Align applications firmware in a FLASH memory; upgrades are made through a port on the SCM2. The distinguishing characteristics of the SCM2 are a DIP switch bank with eight switch positions, four diagnostic LEDs, and a programming point for the FLASH memory.

  There are two versions of the SCM2. One has connectors that are compatible with the SCM1 (Figure 2-7). The other has all of its connections at one end (Figure 2-8).

The SCM is connected to the sensor by a coaxial cable.

Figure 2-6: SCM1

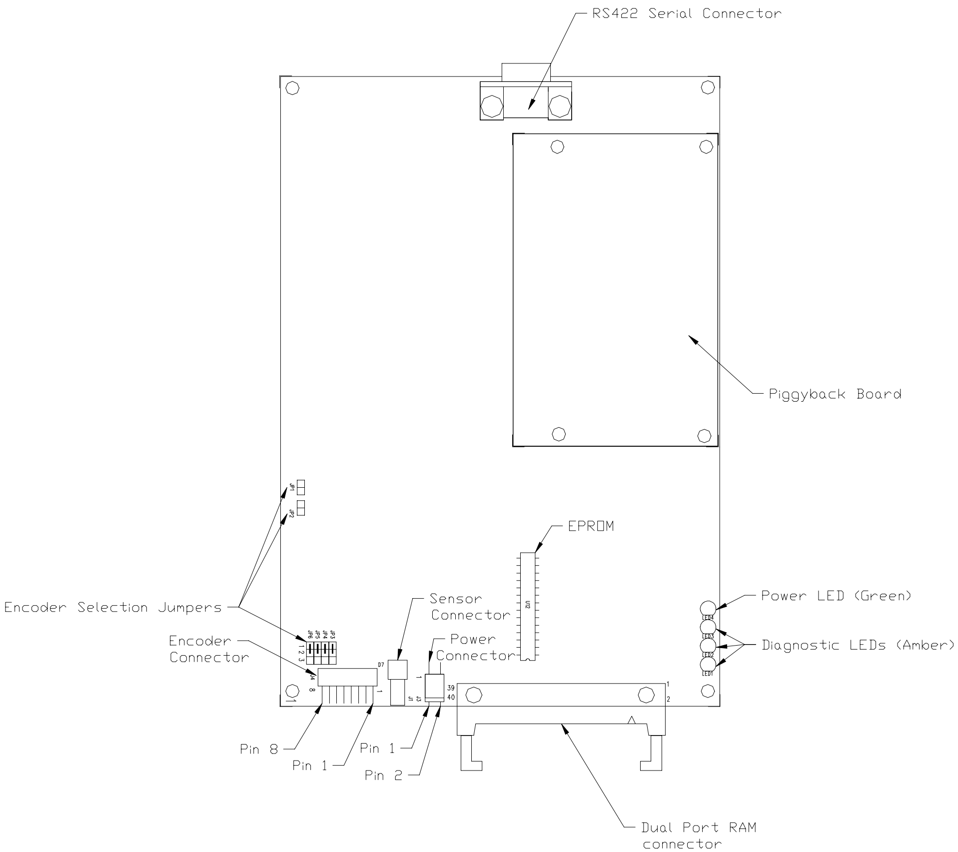


Figure 2-7: SCM2 with SCM1 Compatible Connectors

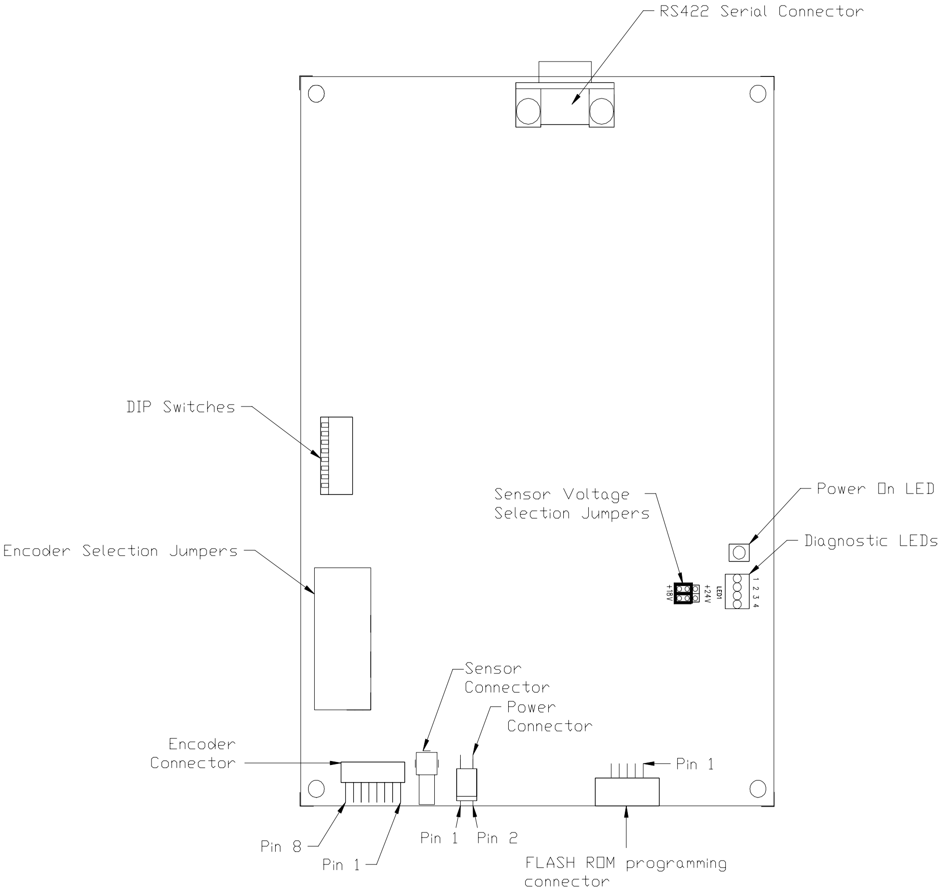
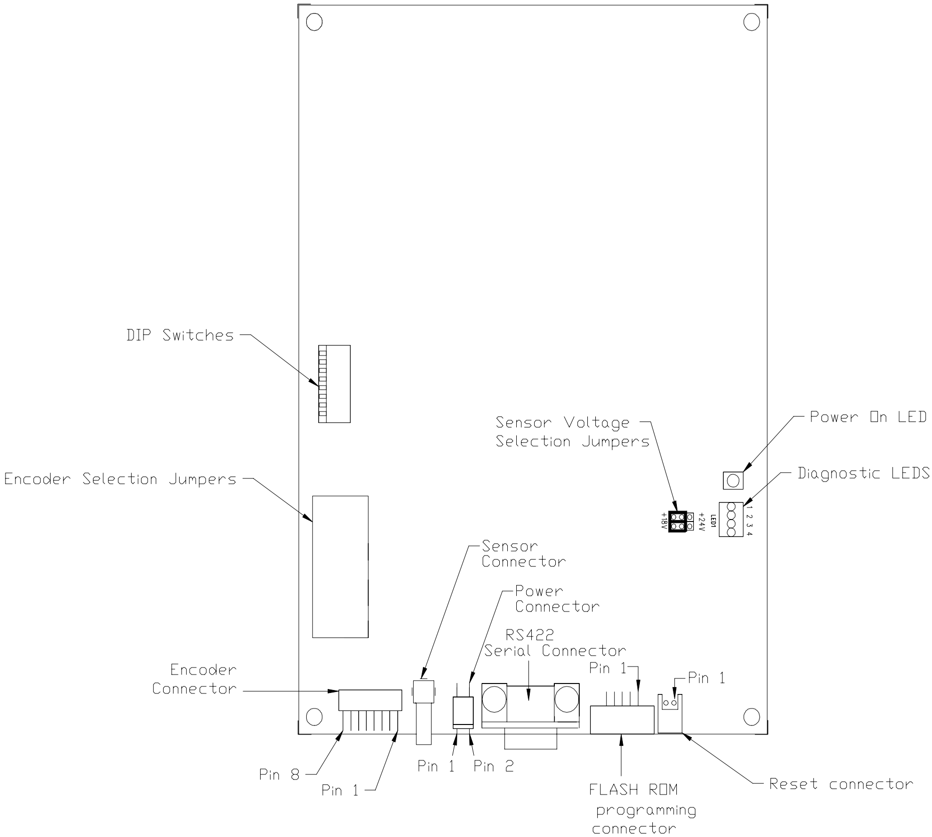


Figure 2-8: SCM2 with all connectors at one end



Sensor Control Module (SCM) Components and Functions

Table 2-1 identifies the components included on the various SCM boards and their functions, as well as which SCM models use each component.

Table 2-1: SCM Components and Functions

Component	Function	SCM1	SCM2
Diagnostic LEDs	Indicate firmware status/ error conditions. ➲ See *SCM LEDs* in Chapter 13 for details.	X (3 lights)	X (4 lights)
DIP switches	Used to set baud rate and encoder type.		X
Dual port RAM connector	Used to connect the SCM to the host controller when RS422 connections are not implemented.	X	
Encoder connector	Used to connect the SCM to the encoder.	X	X
Encoder selection jumpers	Used to set SCM/encoder communications. ➲ See Encoder Specifications in Chapter 14 for details.	X	X
EPROM	Contains SCM firmware.	X	
FLASH ROM programming connector	Used to upgrade firmware.		X

Table 2-1: SCM Components and Functions (continued)

Component	Function	SCM1	SCM2
Piggyback board	Contains processor.	X	
Power connector	Used to connect SCM to power supply.	X	X
Power LED (green)	Indicates when power is on.	X	X
Reset connector	Used to reset hardware.		X (Only on SCM2 model with all connectors at one end)
RS422 serial Connector	Used to connect SCM to host controller using an RS422 serial cable.	X	X
Sensor connector	Used to connect the SCM to the Laser Align sensor using a coaxial cable.	X	X
Sensor voltage selection jumpers	Sets voltage level for power from the SCM to the sensor.		X

See Chapter 14 for SCM1 and SCM2 specifications.

Coaxial Cable

The coaxial cable provides the communication link between the SCM and the sensor.

Chapter 3

Laser Align Safety

Safety Precautions

Laser Align meets IEC 825 Class I and CDRH Class I regulations.

	Use of controls or adjustments or performance of procedures other than those specified herein may result in hazardous radiation exposure.

Notifications

- Use of controls or adjustments, or performance of procedures other than those specified herein, may result in hazardous radiation exposure.
- Modifications to a sensor could result in hazardous radiation exposure.
- A sensor may require recertification if it is integrated into another system, resold, or otherwise modified. If you have questions, contact CyberOptics Service and Support.
- CyberOptics laser products conform with 21 CFR subchapter J at date of manufacture.
- CyberOptics disclaims liability for incidental or consequential damages arising from the use or misuse of any of its products. The customer is responsible for any operation of CyberOptics products not recommended in the product documentation.

Chapter 4

Setup

4

Overview

Part of the process of selecting a Laser Align system is identifying the characteristics of the system where the sensor is to be used. These characteristics include information on encoder type and settings, host controller, communications rate, part sizes, machine dimensions, etc.

Based on that information, your CyberOptics representative helps you select an appropriate sensor, and the Laser Align system is configured according to your specifications.

When you receive the Laser Align system you need to:

- Unpack the Laser Align components
- Install Laser Align hardware
- Install Laser Align software (optional)
- Verify system functionality

Unpacking

The following items are normally included in the initial Laser Align shipment:

- Laser Align sensor (one of five different models)
- Sensor Control Module (SCM)
- RS422 serial interface card (ISA Bus)
- RS422 serial interface cable
- Coaxial cable for SCM / sensor connection
- Laser Align software
- *Laser Align Reference* manual

Subsequent Laser Align shipments are tailored to meet your specific needs.

The Laser Align components you receive are configured with the information you specified when you ordered the system.

Installing Laser Align Hardware

You may install Laser Align into a test configuration or directly into the system where you will use Laser Align. If this is your first Laser Align implementation, CyberOptics recommends that you begin with a test configuration.

Parts

- RS422 serial interface card or port (installed in host controller)
- Laser Align sensor
- Sensor Control Module (SCM)
- RS422 serial interface cable
- Coaxial cable for SCM / sensor connection
- Encoder cable
- Power cable and connector

Your specific needs and specifications determine which of these parts are supplied by CyberOptics and which are user-supplied. Some cables may require assembly. Table 4-1 cross-references information on cable pinouts and connectors.

Table 4-1: Cable Assembly Cross-References

Cable	Pinout Information	Connector Information
RS422 serial interface cable	Table 14-5 in Chapter 14.	Table 14-14 in Chapter 14.
Coaxial cable for sensor/SCM connection	Not applicable.	
Encoder cable	Table 14-7 and Table 14-8 in Chapter 14.	
Power cable	Table 14-2 in Chapter 14.	

Mounting Requirements

The MiniLAM and High Density Laser Align sensors require a three-point coplanar mounting surface. The Low Profile Laser Align requires a two-point coplanar mounting surface. Both Focused Modular Laser Align models have specific mounting requirements that are available from CyberOptics.

See Sensor Specifications in Chapter 14 for information on mounting torques for the various sensor models.

Installation Procedure

▼▼ To install Laser Align hardware

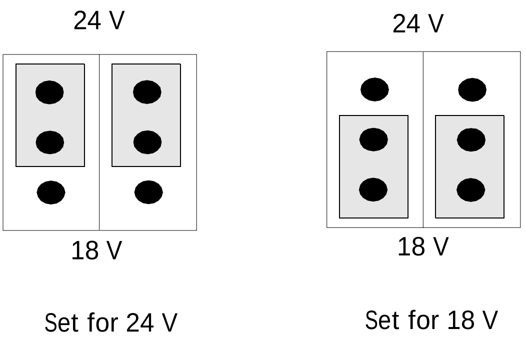
1. Power down the host controller.
2. Install an RS422 Serial Interface card or port in the host controller, if one is not already present.

   Follow the manufacturer's instructions for proper installation of the RS422 Serial Interface into your host controller.

   Note CyberOptics generally uses **COM3** with a base address of **$3E8_{(hex)}$ IRQ5**.
3. Power down the pick-and-place machine where you are installing the Laser Align.
4. If you are using an SCM2, make sure the voltage jumpers are set to the appropriate voltage for your sensor.

   SCM2 has two jumpers that must be configured to select the voltage that is required by the sensor model. Both jumpers must be set to the same voltage (Figure 4-1).

Figure 4-1: SCM2 Sensor Voltage



Use the 24-volt setting for:

- Low Profile
- Focused Modular (side or top connector)

Use the 18-volt setting for:

- MiniLAM
- High Density (LAHD)

Incorrect configuration of these jumpers may damage the sensor.

5 Make sure the encoder jumpers (all SCM models) and switches (SCM2 only) are set appropriately for the encoder.

See Encoder Specifications in Chapter 14 for more information about jumpers and switches for encoder integration.

6 Connect the encoder to the SCM using the encoder cable (user-supplied).

This cable should also be connected to the host controller.

7 Connect the coaxial cable to the Laser Align sensor and to the SCM.

The cable has a threaded (SMA) connector at the SCM, and a threaded (SMA) or snap-on (SMB) connector at the sensor end.

8 Use an RS422 serial interface cable to connect the RS422 serial port in the host controller to the RS422 connector on the SCM.

➡ See Figure 4-2, Figure 4-3, and Figure 4-4 for SCM jumper and connector locations.

Figure 4-2: SCM1 connectors

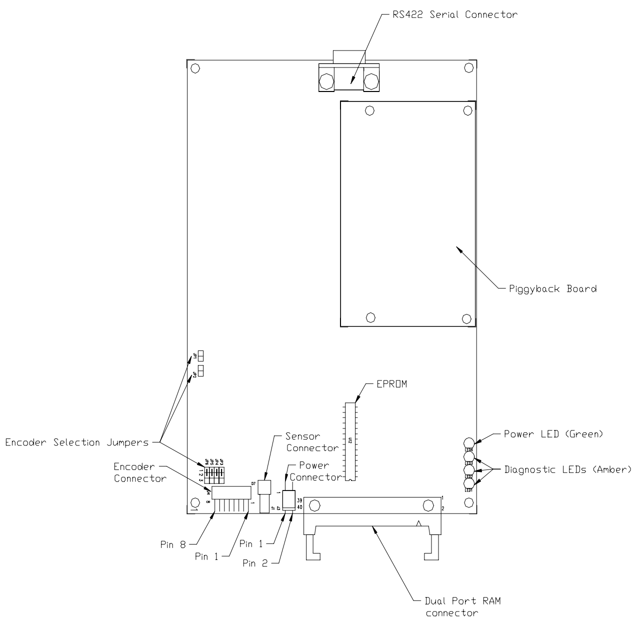


Figure 4-3: SCM2 with SCM1 compatible connectors

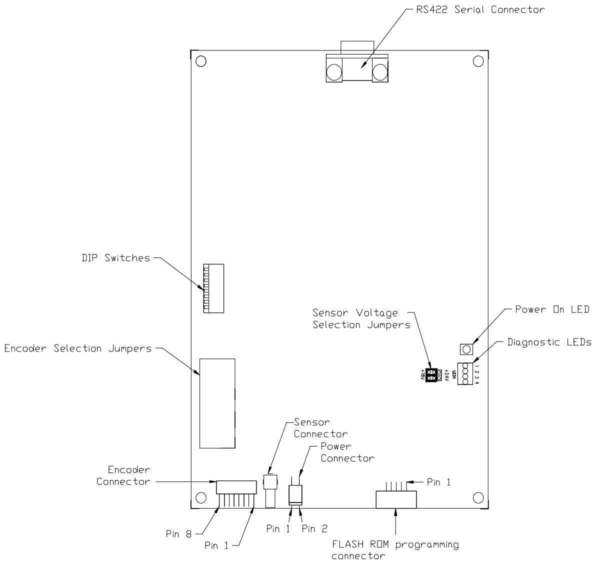
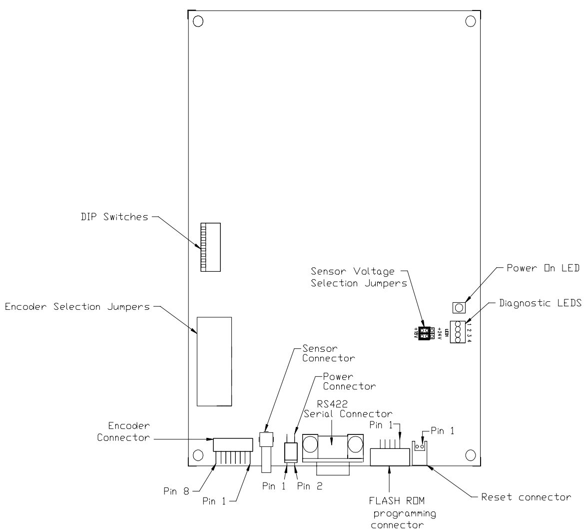


Figure 4-4: SCM2 with connectors all at one end



9 Assemble the power cable and connector (user-supplied).

10 Attach the power cable to the power connector on the SCM (Figure 4-2, Figure 4-3, or Figure 4-4).

11 Connect the power cable to the power source:

- SCM1 power source voltage is sensor-dependent.
- SCM2 power source voltage should always be at least the required voltage for the sensor and the jumpers must be set correctly.

See Sensor Specifications in Chapter 14 for more information about the power requirements.

12 Power up the SCM / sensor power supply.

The system completes a power-on-self-test (POST). The activities and responses of the POST can be obtained by looking at the diagnostic LEDs on the SCM.

See *SCM LEDs* in Chapter 13 for more information about interpreting LED status.

Installing Laser Align Software

There are three software programs provided with the Laser Align system. These programs facilitate development of the host-system application that integrates Laser Align. The programs also facilitate initial system debugging. The programs are PC-compatible, and DOS-based (MS-DOS version 3.3 or higher). If your system platform does not support the use of these programs, you may need to develop programs to support some of the functionality that the Laser Align software provides.

The Laser Align software programs are:

- **COM_422 -** handles RS422 communications between a PC, DOS-based host controller and the Laser Align system. It is a TSR (Terminate and Stay Resident) program set up to provide a low-level communications software driver.
- **COMTEST -** allows the interactive exchange of byte-by-byte commands and responses with the Laser Align system.
- **LA_View -** allows you to verify the proper working condition of the Laser Align sensor by providing a graphical image of the sensor CCD image, along with Laser Align system operating parameters.

▼▼ To install Laser Align software on the host controller PC

1. Go to the DOS prompt and switch to the **c:** drive root directory.
2. Type **cd c:** , then press J.
3. Type **md lasalign,** then press J to make a directory for your Laser Align software.
4. Type **cd c:\lasalign,** then press J to switch to your new directory.
5. Put your Laser Align software diskette in your diskette drive.
6. Type **copy a: *.* c:\lasalign,** then press J. (Use your own drive designation for **a**.)

   The files for the Laser Align software are copied into the designated directory.

See Chapter 11 for more information about Laser Align software.

Verifying Laser Align Functionality

Once you have Laser Align set up, you should check to make sure that the sensor and host system are communicating, and that the sensor is functioning properly. The software provided with Laser Align can be used to perform the necessary tests, or you can develop your own programs to check communications and sensor function.

Communications and Sensor Testing Using Laser Align Software

The LA_View software tests both communications and sensor function.

Note COM_422 must be installed before you can use LA_View.

▼▼ To verify communications and sensor function using LA_View

1. Go to the DOS prompt and switch to the **c:** drive.
2. Type **cd c:\lasalign** to make sure you are in the correct directory.
3. Type **set la_view_cfg = <pathname>**, then press J where:

   <pathname> = the full path and filename of the LA_View configuration file (e.g., **c:\lasalign\la_view.cfg**).

   This sets an environmental variable used by LA_View to find the LA_View configuration file.

4 Type **com_422 [-annn] [-in] [-o** or **-u] [-q] [-vnn]** where:

Parameter	Description
-*annn*	Specifies the hexadecimal base I/O address for the RS422 card to the value of *nnn*. (Default: 3E8 hex)
-i*n*	Specifies the hardware interrupt number (IRQ) for the RS422 card to the value *n*. (Default: 4)
-o or -u	Removes the driver at the specified vector. (The -v*nn* switch must be used with this switch if removing from non-default software interrupt vector.)
-q	Inhibits messages to the standard output device.
-v*nn*	Specifies the hexadecimal software interrupt vector to use for the driver. (Default: 60 hex)

5 Press J.

This executes the COM_422 program which establishes communication through the RS422 interface.

6 Type **la_view**, then press J.

This starts the LA_View software.

Note If your COM_422 address is not set to the default vector of 60 (hex), you must type
la_view -v <vector>
where **-v <vector>** is the COM_422 address.

◆ If the LA_View software does not display, check to make sure that all Laser Align software is installed and that all cabling is correct, then repeat **Step 1** through **Step 5**.

- If LA_View does not execute after checking software installation and cabling and repeating **Step 1** through **Step 5**, use **COMTEST** to verify communications.
- If you still cannot get LA_View to run, or you cannot establish communications, call CyberOptics Service and Support.

7 Press **i** to switch to the image check display mode.

8 Verify that no obstructions are in the path of the sensor laser stripe.

9 Check that the **Image Quality** parameter reads PASSED.

- If the **Image Quality** parameter reads NOT PASSED, clean the windows on the sensor and run the image check again.
  - See Cleaning the Sensor Windows in Chapter 12 for details.
- If the sensor still doesn’t pass, call CyberOptics Service and Support.

10 Press E to exit LA_View.

COMTEST software allows you to check communications, and to work with the various Laser Align firmware commands, one at a time.

▼▼ To verify communications using COMTEST

1. Go to the DOS prompt and switch to the **c:** drive.
2. Type **cd c:\lasalign** to verify that you are in the correct directory.
3. Type **com_422 [-annn] [-in] [-o** or **-u] [-q] [-vnn]** where:

Parameter	Description
-a*nnn*	Specifies the hexadecimal base I/O address for the RS422 card to the value of *nnn*. (Default: 3E8 hex)
-i*n*	Specifies the hardware interrupt number (IRQ) for the RS422 card to the value *n*. (Default: 4)
-o or -u	Removes the driver at the specified vector. (The -v*nn* switch must be used with this switch if removing from non-default software interrupt vector.)
-q	Inhibits messages to the standard output device.
-v*nn*	Specifies the hexadecimal software interrupt vector to use for the driver. (Default: 60 hex)

4. Press J.

   This executes the COM_422 program which establishes communication through the RS422 interface.
5. Type **comtest**, then press J.
6. Type **C** to select the **<C> Command** option.
7. Type **16**, then press J to send **HELLO_CMD (16)**.

8 Wait for the **HELLO_ACK** response, which displays as decimal **128**.

- If you receive the **HELLO_ACK (128)** response, communications were successful, and you can proceed with your installation and setup process.
- If you do not receive the **HELLO_ACK (128)** response, recheck all of your equipment and all of your setup processes to identify possible areas of error. If you still do not get the correct response, contact CyberOptics Service and Support.

See Chapter 11 for more information about using Laser Align software.

Communications and Sensor Testing Without Laser Align Software

If you do not use the Laser Align software provided by CyberOptics to check communications and sensor functionality, you need to develop your own programs.

- To verify communications, develop code to ensure that the host controller can send a **HELLO_CMD (16)** and receive a **HELLO_ACK (128)** response from Laser Align.
- To check sensor image quality, develop program code that incorporates the use of **IMG_CHK_CMD (43)** functionality.

See Monitoring Image Quality in Chapter 6 for more information about monitoring image quality with **IMG_CHK_CMD (43)**.

Note To completely duplicate Laser Align software functionality requires substantial additional program development.

Chapter 5

Integration

Overview

Integrating Laser Align into your manufacturing system involves developing and testing the program code required to:

- Synchronize (initialize) Laser Align with your system
- Specify the measurement, reporting, and placement protocols Laser Align is to use for various component types

There are a number of different procedures associated with this process. The procedures used to synchronize Laser Align and your system are:

- Defining general system parameters
- Synchronizing with the encoder
- Calibrating pickup nozzle position, both height (Z) and center (X/Y)
- Calibrating theta (θ) position

The procedures used to define the measurement, reporting, and placement protocols for various component types are:

- Determining measurement position of the component
- Identifying measurement parameters for the component
- Identifying reporting requirements
- Determining component rotation requirements
- Defining courses of action for your pick-and-place machine based on Laser Align results

Note All of these procedures must be completed before you can perform any component measurements.

The program code you develop will include the use of commands and codes specific to Laser Align:

- **Commands** are documented in Chapter 7.
- **Parameter indexes**, which link the commands to a specific measurement or synchronization parameter, are documented in Chapter 8.
- **Status codes** associated with Laser Align results are documented in Chapter 9.

Defining General System Parameters

Each time you power up or reset your Laser Align system, you need to set the following parameters:

- **Result units -** identifies the measurement units Laser Align will use for reporting results. The options are **1/10 pixels** or **microns**.

  This parameter **must** be set every time you power up or reset Laser Align.

- **Flow control -** identifies whether flow control (hardware "handshaking") is part of the system. Flow control involves using RTS/CTS signals on the data line to control the data flow between the SCM and the host controller. The options are **enabled** (which gives slower, but more reliable data transfer) and **disabled** (which gives faster data transfer). The default is **disabled**.

- **Error level -** identifies how Laser Align reports error/status codes. The options are **0, 1, 2,** or **3**, where **0** has the least specific messages/codes and **3** has the most specific messages/codes. The default is **0**.

  The less specific error levels are provided to give backward compatibility with older versions of Laser Align. Typically, it is best to use the most specific error level supported by your firmware (level **3**) to get the most complete error feedback via Laser Align status codes.

  Because the error level resets to the default of **0** whenever you power cycle or reset the system, it is recommended that you set the error level every time you power up or reset the system.

The following procedures provide the command sequences required to define the general system parameters.

See Chapter 7, Chapter 8, and Chapter 9 for more information about commands, indexes, and codes.

▼▼ To set reporting units to 1/10 pixels

1 Send **LOAD_CMD** sequence:

40 0 255 0 0 0 0 217

2 Check Laser Align response:

- If response is **152 (LOAD_ACK)**, reporting units are set to 1/10 pixels.
- If response is not **152 (LOAD_ACK)**, repeat **Step 1**.

Note If you power down and then power up, or use **RESET_CMD (17)** to reset Laser Align, the reporting units setting is removed. You must use **LOAD_CMD (40)** with the specified parameters to set your reporting units.

▼▼ To set reporting units to microns

1 Send **LOAD_CMD** sequence:

40 0 255 1 0 0 0 216

2 Check Laser Align response:

- If response is **152 (LOAD_ACK)**, reporting units are set to microns.
- If response is not **152(LOAD_ACK)**, repeat **Step 1**.

Note If you power down and then power up, or use **RESET_CMD (17)** to reset Laser Align, the reporting units setting is removed. You must use **LOAD_CMD (40)** with the specified parameters to set your reporting units.

▼▼ To enable flow control

1. Send **LOAD_CMD** sequence:

   40 1 255 1 0 0 0 215

2. Check Laser Align response:

   - If response is **152 (LOAD_ACK)**, flow control is enabled.
   - If response is not **152 (LOAD_ACK)**, repeat **Step 1**.

Note If you power down and then power up, or use **RESET_CMD (17)** to reset Laser Align, flow control is disabled. You must use **LOAD_CMD (40)** with the specified parameters to enable flow control again.

▼▼ To disable flow control

1. Send **LOAD_CMD** sequence:

   40 1 255 0 0 0 0 216

2. Check Laser Align response:

   - If response is **152 (LOAD_ACK)**, reporting units are set.
   - If response is not **152 (LOAD_ACK)**, repeat **Step 1**.

Note You only need to use this procedure if you want to disable flow control without resetting or powering down. If you power down and then power up, or use **RESET_CMD (17)** to reset Laser Align, the system automatically disables flow control.

▼▼ To set error level

1. Send **LOAD_CMD (40)** sequence:

   40 2 255 *n* 0 0 0 y

   where:

   n = desired error level
   - **0** = valid for all firmware versions
   - **1** = valid for firmware versions 1.0 and higher
   - **2** = valid for firmware versions 2.0 and higher
   - **3** = valid for firmware versions 2.7 and higher

   y = checksum (256 **-** [(Σ bytes 0 through 6) mod 256]; must be ≤ 255)

2. Check Laser Align response:

   - If response is **152 (LOAD_ACK)**, error level is set.
   - If response is not **152 (LOAD_ACK)**, repeat **Step 1**.

Note The system sets itself to error level **0** when you power down and then power back up, or if you use **RESET_CMD (17)** to reset Laser Align. You must use **LOAD_CMD (40)** with the **error level** parameter index to set an error level other than **0**.

▼▼ To verify a general system parameter setting

1 Send **READ_CMD (41)** sequence:

41 *n* 255 y

where:

n = the index identifying the setting you want reported
0 = units
1 = flow control
2 = error level

y = checksum (256 **-** [(Σ bytes 0 through 2) mod 256]; must be ≤ 255)

2 Check Laser Align response.

- If the response is not a 5-byte response string, repeat **Step 1**.
- If the 5-byte response string is provided, check the value for the current setting of the requested parameter.

See *READ_CMD (41)* in Chapter 7 to interpret the response.

Synchronizing with the Encoder

Before measuring components, the Laser Align position counter must be set to a known value relative to the position counter of the host processor. You synchronize Laser Align with your encoder based on your specific application.

See Chapter 7, Chapter 8, and Chapter 9 for more information about Laser Align commands and codes.

▼▼ To set the Laser Align position counter to 0 (zero) using the encoder index pulse

1. Send **RUN_CMD (19)** sequence:

   19 10

2. Check Laser Align response:

   - If response is **131 (RUN_ACK)**, the command was successfully received and the Laser Align position counter can be reset by an index pulse.
   - If response is not **131 (RUN_ACK)**, repeat **Step 1**.

3. Use motion control to spin the pickup nozzle past its index pulse point (may require 360°).

   The encoder index pulse resets the position counter.

4. Send **REPORTW_CMD (42)** sequence:

   42 2 0

5. Check the report data returned:

   - If status (Byte 0) is **1 (NO ERROR)**, the next four data bytes indicate the encoder position, which can be used to synchronize Laser Align with the system encoder.
   - If status (Byte 0) is not **1 (NO ERROR)**, you need to determine where the error occurred and repeat the procedure.

▼▼ To set the Laser Align position counter to 0 (zero) independent of the encoder index, or in a stepper motor system

1 Send **ZERO_CMD (22)** sequence:

22

2 Check Laser Align response:

- If response is **134 (ZERO_ACK)**, the command was successfully received and the Laser Align position counter is reset to zero.
- If response is not **134 (ZERO_ACK)**, repeat **Step 1**.

▼▼ To set the Laser Align position counter to a specified number

1 Send **LOAD_CMD (40)** sequence:

40 247 0 n_0 n_1 n_2 n_3 y

where:

n_0 = value for Byte 0 (LSB) of desired encoder position
n_1 = value for Byte 1 of desired encoder position
n_2 = value for Byte 2 of desired encoder position
n_3 = value for Byte 3 (MSB) of desired encoder position
y = checksum (256 **-** [(Σ bytes 0 through 6) mod 256]; must be ≤ 255)

Note Use window **0** when setting the encoder counter value.

2 Check Laser Align response:

- If response is **152 (LOAD_ACK)**, the command was successfully received, and the encoder value was successfully loaded.
- If response is not **152 (LOAD_ACK)**, repeat **Step 1**.

3 Send **PRESET_CMD (35)** sequence:

35 n

where **n** = the encoder number (**0** or **1;** 1 is valid only for SCM1).

4 Check Laser Align response:

- If response is **147 (PRESET_ACK)**, the command was successfully received and the position counter for the designated encoder is now set to the specified value.
- If response is not **147 (PRESET_ACK)**, repeat **Step 3**.

Calibrating Pickup Nozzle Position

Whenever you start the pick-and-place system or change the pickup nozzle, you must create a correlation between the Laser Align coordinate system and the placement machine coordinate system. This involves two procedures:

- Determining nominal height (Z) calibration of the nozzle
- Determining center position of the nozzle

The Z-height of the nozzle tip when it is in the middle of the light stripe (**Z_{cal}**) is the nominal height for calculating component location. The host placement machine must use this value, plus an *offset*, to determine the ideal location for component measurement.

The *offset* (in a vertical direction) is typically one-half of the component thickness, but can vary with component package type.

Offsets should be determined based on component type. Your host machine should have a component library with Z-offsets for each component type to be placed by the Laser Align system.

See *Determining Component Measurement Position* later in this chapter for more information.

The center position of the nozzle (**C_{cal}**) is used to determine the position of the nozzle in relation to the Laser Align system. This center position is used as the calibration each time a component is measured. The reported center (or edge) position of the component is reported in relation to the Laser Align system. Your host controller must use this reported position and compare it to **C_{cal}** to determine the pickup error in the X- and Y-directions.

Determining Nominal Nozzle Height

To calibrate the nominal height Z_{cal} of a nozzle, position the nozzle above the light stripe, lower the nozzle gradually until the light stripe is blocked, and record the height.

Next, lower the nozzle through the light stripe, raise it gradually until the light stripe is no longer blocked, then record this height.

Finally, average the two different heights to get Z_{cal} (the height of the nozzle tip when it is in the middle of the light stripe).

Note This procedure should be repeated and the Z_{cal} recorded for every different nozzle.

▼▼ To calibrate pickup nozzle Z-height (Z_{cal})

1. Move the nozzle to a position approximately 1 mm above the nominal light beam height.
2. Send **MEASURE_ONCE** command sequence:

   23 14
3. Send **REPORTW_CMD** command sequence:

   42 2 0
4. Check status (Byte 0).
   - If status is **64 (No Part)**, move the nozzle down by one step (~10 microns) and go back to **Step 2**.
   - If status is **1 (NO ERROR)**, record nozzle height as Z_1.
5. Move the nozzle down by 1 mm.
6. Send **MEASURE_ONCE** command sequence:

   23 14
7. Send **REPORTW_CMD** command sequence:

   42 2 0

8 Check status (Byte 0).

- If status is **1 (NO ERROR)**, move the nozzle up by one step (~10 microns) and go back to **Step 6**.
- If status is **64 (No Part)**, record nozzle height as Z_2.

9 Calculate the average of Z_1 and Z_2:

$$(Z_1 + Z_2) \div 2 = Z_{cal}$$

Determining Nozzle Center Position

To determine the center position of the pickup nozzle (C_{cal}), you move the nozzle into the light stripe, then measure and report its center position. If your nozzle has runout, C_{cal} should be determined for different encoder θ positions for better placement accuracy.

▼▼ To calculate pickup nozzle center position (C_{cal})

1 Move the pickup nozzle to Z_{cal} minus **0.5mm** (0.5mm closer to the board than Z_{cal}).

Note This assumes that 0 is board level, and Z_{cal} is a positive number. When you perform this step you are moving the nozzle into the light stripe.

2 Send **MEASURE_ONCE** command sequence:

23 14

3 Send **REPORTW_CMD** sequence:

42 2 0

4 The reported center position (bytes 5 through 8 of the Laser Align response) is your C_{cal} value.

See Chapter 7, Chapter 8, and Chapter 9 for more information about Laser Align commands and codes.

Calibrating Theta Position

Because of the nature of the Laser Align sensors, there can be variations in the exact theta (θ) position of the light stripe in relation to the sensor case. To compensate for this, you must determine the amount of offset between the light stripe and the X-Y coordinate axis; the *theta offset*.

The theta offset must be determined and included in your system programming as part of each new sensor installation. It must also be done any time you swap out sensors or remount sensors.

Note You do not need to perform theta calibration for the Laser Align MiniLAM, because the beam offset for this sensor model is stored in the EEPROM in the sensor head.

▼▼ To determine theta offset

1. Use the Laser Align sensor to measure and align a square block.
2. Use a system camera or dial gauge to determine the angle of the block in relation to the X-Y coordinate axis for the system.

   The resulting value is the theta offset. All placement calculations should include compensation for this offset value.

Determining Component Measurement Position

Effective component measurement requires proper placement of the component in the light stripe. The height of the pickup nozzle determines the effective position of the light stripe on the component.

By changing nozzle height, the orientation measurement can be performed on either the component body or its leads. For symmetrical components, either measurement typically works well. However, some components, such as those with asymmetrical bodies or loose mechanical tolerances between leads and the component body, require placement of the light stripe on the leads.

For example, Figure 5-1 illustrates two possible light stripe positions on an SOIC package with an asymmetrical package body design. If **Light Stripe Position 1** is used, Laser Align determines an incorrect center position. Using **Light Stripe Position 2**, however, Laser Align can accurately determine center position by reporting the minimum distance between the shadows cast by the two opposite sets of leads. Note that this method does not measure each individual lead, but instead measures the shadow distance between the two opposite rows of leads.

Figure 5-1: Possible light stripe positions

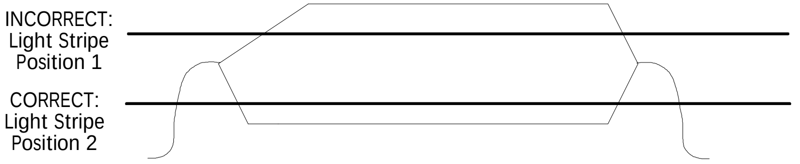


Figure 5-2 and Figure 5-3 show the appropriate range of light stripe locations for J-lead components and SOIC components.

Figure 5-2: Range of light stripe positions for J-lead components

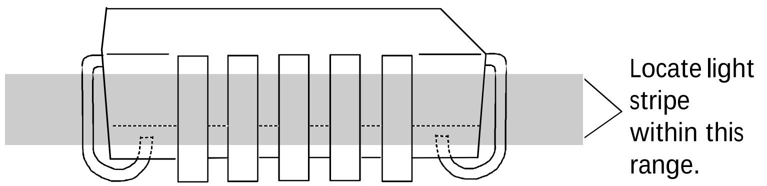
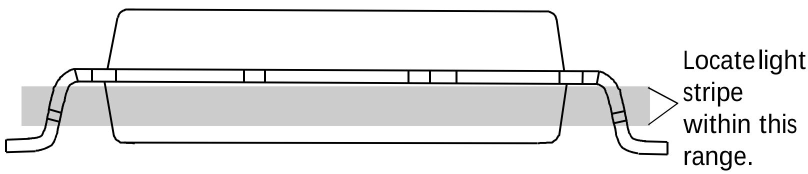


Figure 5-3: Range of light stripe positions for SOIC components



You must identify the correct position of the light stripe for each type of component you will be measuring with Laser Align, and then use this information to calculate the offset for the component.

Note The component must completely block the light stripe in the Z axis.

▼▼ To calculate the Z-offset for a component

1. Determine the proper light stripe position for the component.
2. Measure the distance from the top of the component, as it is picked up by the nozzle, to the position on the side of the component where the top of the light stripe should strike the component.

   This is the offset value for the component type.

The host controller uses the offset and the calibrated Z-height of the nozzle (Z_{cal}) to properly position the component.

Offsets for each type of component you measure should be maintained in a library on your host controller.

Identifying Measurement Parameters

After defining system parameters, synchronizing with the encoder, calibrating pickup-nozzle position, and determining component measurement position, you must identify measurement parameters, including:

- **Window identification** - tells Laser Align which measurement window to use (only SCM1 allows multiple windows).
- **Window size** - tells Laser Align the size of the window(s).
- **Encoder inputs** - identifies the encoder that is providing information on rotation, etc. (SCM1 only).
- **Scanning parameters** - tells Laser Align how many alignment angles to measure, and how to measure them.

These parameters are sent to Laser Align using a series of **LOAD_CMD (40)** commands prior to initiating measurement.

See Chapter 7 for more information about **LOAD_CMD (40).**

See Chapter 8 for more information about the measurement parameters used with **LOAD_CMD (40).**

Window Identification

The area of the detector array used for measurement is called a *window*. You can use two windows if you are using an SCM1. This feature allows simultaneous measurement of two parts. The windows in the SCM1 are defined as **window 0** and **window 1**. In the SCM2, only **window 0** is available for measurement data.

The **window identification** is part of every **LOAD_CMD (40)** that you send. The identifying parameter index occupies the second byte of the command.

Example: The following command sequence specifies the measurement algorithm for a result coming from **window 0**.

40 1 0 14 0 0 0 201

window identifier

Window Size

The **window size** is handled as a specific **LOAD_CMD (40)** in which you indicate that you are defining a left or right edge, identify which window it applies to, and identify the number of the pixel in the array where the edge is located.

Sizing the window is optional if you are only working with one measurement window. If no specific window size is set, the window size remains at its default, which is the entire illuminated detector range.

However, if you are using an SCM1 and setting it up to handle two parts, you must define the left and right edges of both windows. The windows can use overlapping portions of the sensor detector. For example, the first **window 0** could have a left edge of 8 and a right edge of 1000, and **window 1** could have a left edge of 800 and a right edge of 2500. Generally, however, this is improper, since this would imply that the parts being measured could collide during rotation, or that their shadows could overlap.

Note Minimum window size is 65 pixels if used in conjunction with IMG_CHK_CMD (43).

Example: The following command sequence sets a **left edge** at **pixel 100** for **window 0**.

40 240 0 100 0 0 0 132

pixel number
window identifier
index specifying left edge
load command identifier

Encoder Inputs

This information is required only if you:

- have an SCM1;
- measure two parts at once; and,
- use separate encoders for the two placement heads.

Given these conditions, a **LOAD_CMD (40)** that identifies which encoder to activate for subsequent measurements must be included as part of the measurement protocol. The encoders are identified as **encoder 0** and **encoder 1**. If no encoder is specified, Laser Align uses the default, which is **encoder 0**.

Example: The following command sequence tells Laser Align that subsequent measurement parameters apply to a scan using **encoder 1**, in **window 1**.

40 242 1 1 0 0 0 228

encoder number
window identifier
index specifying encoder identification
load command

Scanning Parameters

For each type of component you want to measure with Laser Align, you must identify the number of component positions or *alignment angles* to be measured / reported, and how the measurements are to be taken. You send this information to Laser Align from your host controller by using a series of **LOAD_CMD (40)** command sequences.

You can select up to four different alignment angles to be measured and reported for each component. These are identified as **Result 0, Result 1, Result 2,** and **Result 3**.

For each desired result, you need to create a series of load commands that provide Laser Align with the following parameters:

- **Measurement algorithm** - selects the alignment criterion and image processing technique to be used for the alignment scan.
- **Angle limit** - specifies the delay in encoder counts before Laser Align stops looking for a local minimum shadow width and reports an error.
- **Holdoff angle** - specifies the delay in encoder counts between the previous alignment angle and the start of the current alignment scan. (In the case of the first result, the holdoff angle specifies the delay in encoder counts between the issuance of the SWEEP_CMD (30) and the beginning of the measurement process.)

See Scanning Parameter Details in Chapter 8 for more information about measurement algorithms, angle limits, and holdoff angles.

You must provide these three parameters for each result unless you want to use the Laser Align defaults for those values. This means you will use several load commands to define a particular measurement protocol.

For example, if you want to measure two different alignment angles for a single component, the measurement specifications could require the creation of six different load commands (three for each result). If you wanted all four alignment angles measured, it could take up to 12 different load commands to load the scanning parameters.

Typically, you should establish and maintain a library of the load command sequences defining the measurement protocol for each component type you measure.

Example: The following series of commands would be required in order to set the measurement parameters for a component where you want to measure two alignment angles.

Table 5-1: Sample Measurement Parameters

Desired Settings	Angle 1 (Result 0)	Angle 2 (Result 1)
Algorithm	14	14
Holdoff	0	1500
Angle limit	750	750

For Angle 1 (*result 0*):

40 1 0 14 0 0 0 201
Loads algorithm 14.

40 0 0 0 0 0 0 216
Loads the holdoff angle of 0 counts.

40 2 0 238 2 0 0 230
Loads the angle limit of 750 counts.

First minimum expected within 750 encoder counts of the start of scan.

For Angle 2 (*result 1*):

40 17 0 14 0 0 0 185
Loads algorithm 14.

40 16 0 220 5 0 0 231
Loads the holdoff angle of 1500 counts.

40 18 0 238 2 0 0 214
Loads the angle limit of 750 counts.

Second minimum expected between 1500 and 2250 counts after the first minimum.

Laser Align should respond with a **LOAD_ACK (152)** after each **LOAD** command.

Identifying Reporting Requirements

You must ask Laser Align to provide the results for each measurement taken during a scan that you want reported. This is accomplished by using a series of **REPORTW_CMD (42)** commands. For each result to be reported, you must identify whether you want angle, center, and width information, or only the angle measurement.

Example: The following command sequence would be required to get results for a component where you are measuring two sides, and want the angle, center, and width for the first side, and the angle only for the second side.

42 2 0
Tells Laser Align to report angle, center, and width for the first alignment angle (*result 0)*.

42 17 0
Tells Laser Align to report angle only for the second alignment angle (*result 1*).

➲ See Chapter 7 for details about **REPORTW_CMD (42)**.

Note When angle only reporting is requested, center and width values should be ignored in the results.

Determining Component Rotation Requirements

Determining Component Initial Position

When the component is moved to its initial position, it should be rotated a distance slightly more than twice the maximum pickup uncertainty. This technique, called *prerotation*, ensures that the width of the shadow cast onto the detector array passes through a minimum during the beginning of rotation.

The initial position should be set to ensure that the first minimum (alignment angle) that the component is rotated through is the lesser minimum (narrower dimension). The lesser minimum provides better definition and more repeatable results than the wider dimension.

Rotating the Component

The component is rotated as Laser Align collects exposure data. This is called *sweeping* or *scanning*.

Laser Align records the angular position at which the shadow widths are minimized. To determine these minimums, Laser Align looks for increases in shadow width immediately after consistent decreases. A data filter ignores noise in the data that may otherwise indicate incorrect minimums.

In these position minimums, the component is oriented parallel to the light stripe. The Laser Align system calculates the component θ position, the lateral position, and the width for each requested minimum.

Working with Results

After defining your measurement parameters and results, you define what your system response should be, based on the information received from Laser Align. Laser Align results are reported in the following RESULT data structure. Integer values are transmitted in two's-complement little-endian format.

Table 5-2: RESULT Data Structure

Byte Number	Description
0	Status
1 - 4	Align angle
5 - 8	Center position for WIDTH_ALG (11) and WIDE_ALG (14) Edge position for LEFT_ALG (12) and RIGHT_ALG (13) Zero for REZERO_ALG (10)
9 - 12	Width Zero for REZERO_ALG (10)
13	Result Index
14	Window
15	Checksum: 256 - (Σ(bytes 0 to 14) mod 256); must be $\leq$ 255

The status, alignment angle, center (or edge), and width values that are returned are used with your information on nozzle position (C_{cal}), theta offset, and part size to determine proper part placement.

See Placing Parts Using Laser Align Results in Chapter 6 for a sample procedure.

As with other aspects of Laser Align, how you use your results is dependent upon your specific application.

Chapter 6

Operation

Using Laser Align

Before you can use Laser Align to perform measurements, you must have completed installation and integration, which are described in earlier chapters.

As you put Laser Align into operation, remember that:

- Laser Align resets itself to its initial default values if you power down and power up, or if you use **RESET_CMD (17)** to reset the firmware.
- If you do not specify a value for a required parameter, Laser Align uses the default value.

Performing Measurements

Before you begin performing measurements, you should have created a measurement protocol for the component type that includes the commands needed to:

- Set general system parameters
- Synchronize with the encoder
- Position the component
- Define measurement parameters

After this information is sent to Laser Align, you can issue a **SWEEP_CMD (30)** and rotate the component to perform the actual measurements.

Next, you send the series of **REPORTW_CMD (42)** sequences needed to obtain the results. Typically this is done after the sweep rotation stops; however, REPORTW_CMD (42) can also be issued during the sweep.

▼▼ To perform a measurement

1 Make sure that your general system parameters are correctly set. These parameters are:

- **Measurement units** (microns or $^{1}/_{10}$ pixels)
- **Flow control** (enabled or disabled)
- **Error level** (0, 1, 2, or 3)

See *Defining General System Parameters* in Chapter 5 for more information.

2 Make sure that your pickup nozzle, encoder, and component positions are correctly set for the desired component.

See *Synchronizing with the Encoder*, *Calibrating Pickup Nozzle Position*, and *Determining Component Measurement Position* in Chapter 5 for more information.

3 Send the series of **LOAD_CMD (40)** commands required to relay the measurement parameters defined for the component.

Your command series should include a **LOAD_CMD (40)** to specify:

- **Measurement algorithm** for each desired measurement result, unless you are using a previously specified value or the default (no measurement).
- **Angle limit** for each measurement result, unless you are using a previously specified value or the default (no angle limit).
- **Holdoff angle** for each measurement result, unless you are using a previously specified value or the default (0).
- **Window size** - if you aren't using the entire detector range or if you are using two windows.
- **Encoder inputs** - if you are using an SCM1 to measure two parts and you want to use a separate encoder for each placement head.

Laser Align should respond with a **LOAD_ACK (152)** for each **LOAD_CMD (40)**.

See *Identifying Measurement Parameters* in Chapter 5 for more information about setting up a measurement using **LOAD_CMD (40)**.

See Chapter 7 for more details about **LOAD_CMD (40)**.

See Chapter 8 for more information about measurement parameters.

4 Send **SWEEP_CMD (30)** to perform the measurement.

Laser Align should respond with a **SWEEP_ACK (142)**; the host controller must initiate component rotation.

5 Send the **REPORTW_CMD (42)** sequence required to obtain your first measurement result after rotation is complete.

Laser Align returns report data for the requested result.

Note You need a separate **REPORTW_CMD (42)** defining the information to be reported for each requested measurement result.

See Chapter 7 for more information about **REPORTW_CMD (42)**.

6 Check the reported result:

- If the result values are within tolerances, repeat **Steps 5** and **6** for the next desired result, until all results have been reported.
- If the result values are not within tolerances, initiate whatever action you have decided upon for your application (e.g., notifying the operator).

7 Place or discard the part, depending on the results.

- If you want to measure another part using the same measurement parameters, repeat **Steps 4** through **7**.
- If you want to measure a part using different measurement parameters, go back to **Step 1**.

Placing Parts Using Laser Align Results

After angle, center (or edge), and width results are received, you can use the information to accurately place a component. In addition to the results, you need to know:

- C_{cal} (nozzle position)
- Desired part orientation
- Expected part width

The calculations and adjustments you need to perform with the results depend upon your application; however, the following procedure offers some basic guidelines.

▼▼ To place a part using Laser Align results (basic guidelines)

1 Check status information.

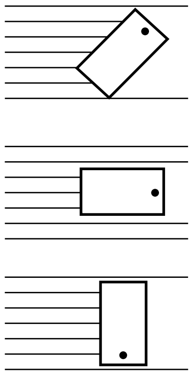
- If status for all results is **1 (NO ERROR)**, go to **Step 2**.
- If status is not **1 (NO ERROR)**, take corrective action based on the status.

See Chapter 9 for more information about status codes.

2 Correct part position in θ (theta).

Result 0 and **Result 1** angles give you the amount of offset required for proper component placement (Figure 6-1).

Figure 6-1: Positions of rotated part



For Example: To place the part (Figure 6-1) in the correct orientation, you would correct the θ position of the encoder to the pickup position plus a value (***n***, ***m***, or a value between the two), depending upon how the component is to be oriented on the board.

Note For rectangular components, the angle (θ position) related to the smallest shadow width (Result 0 in Figure 6-1) is the more accurate of the two angles. It is sometimes referred to as the angle at the lesser minimum, and gives the most repeatable results.

3 Compare the reported center at **Result 0** and **Result 1** to $\mathbf{C_{cal}}$ to obtain **x** and **y** displacement values for the component.

The difference between the reported centers and $\mathbf{C_{cal}}$ represents the amount of offset required in the **x** and **y** directions.

Note If you are working with edge measurement algorithms (12 and 13) and the edges they report, you will need to factor part width and pickup orientation into any calculations of **x** and **y** center positions.

4. Perform a general comparison of reported widths at **Result 0** and **Result 1** with known component dimensions to ensure that the component is properly oriented (i.e., that the component is not tombstoned).
5. Place the part.

Monitoring Image Quality

The quality of the sensor image produced by Laser Align can be assessed in one of two ways:

- **LA_View** software
- **IMG_CHK_CMD (43)**

LA_View is used to find out whether the sensor image passes the image quality parameters set for the sensor at the factory, and to view the actual image quality parameter values generated by the sensor image.

See Chapter 11 for more information about **LA_View**.

Using the **IMG_CHK_CMD (43)**, you can develop programs for the Laser Align system that monitor image quality as part of the measurement process in your manufacturing operation. Use **IMG_CHK_CMD (43)** to:

- Set upper and lower limits for the various image quality parameters
- Read the upper and lower limits that have been set
- Read the actual image quality parameter values that describe the sensor image
- Determine whether the sensor is violating any of the image quality limits you've established
- Detect which limits are being violated.

This section documents the procedures used in working with **IMG_CHK_CMD (43)**.

See Chapter 8 for a list of the image quality parameters and associated indexes used in the following procedures, as well as definitions of the parameters.

See Chapter 7 for more information about **LOAD_CMD (40)** and **READ_CMD (41)**.

▼▼ To set image quality parameter limits

1 Send **LOAD_CMD (40)** sequence:

40 n w X X X X y

where:

n = index for the upper or lower limit you want to set
w = window **0** or **1** (window **1** valid for SCM1 only)
X = value for the limit (four-byte value)
y = checksum (256 **-** [(Σ bytes 0 through 6) mod 256]; must be ≤ 255)

2 Check Laser Align response:

- If response is **152 (LOAD_ACK)**, the parameter limit is set.
- If response is not **152 (LOAD_ACK)**, repeat **Step 1**.

3 Repeat **Steps 1** and **2** for each image quality parameter limit you want to set.

▼▼ To read image quality parameter limits

1 Send **READ_CMD (41)** sequence:

41 n w y

where:

n = index for the upper or lower limit you want to read
w = window **0** or **1** (window 1 valid for SCM1 only)
y = checksum (256 **-** [(Σ bytes 0 through 2) mod 256]; must be ≤ 255)

2 Check Laser Align response:

- Response should be a 5-byte value, the first four bytes provide the value that is set for the image quality parameter limit you specified. (Byte 5 is the checksum).
- If response is not a 5-byte data string, repeat **Step 1**.

3 Repeat **Steps 1** and **2** for each image quality parameter limit you want to read.

▼▼ To read actual values for image quality parameters

1 Send **IMG_CHK_CMD (43)** sequence:

43 n

where

n = the desired image check mode.

➲ See *IMG_CHK_CMD (43)* in Chapter 7 for more information about image check modes.

Laser Align checks the image quality.

2 Check Laser Align response:

- ◆ If response is **155 (IMG_CHK_ACK)**, the image check has successfully completed. Go to **Step 3**.
- ◆ If response is not **155 (IMG_CHK_ACK)**, repeat **Step 1**.

3 Request image quality parameter value; send **READ_CMD (41)** sequence:

41 n w y

where:

n = index for the actual image quality value you want to read
w = window **0** or **1** (window 1 valid for SCM1 only)
y = checksum (256 - [(Σ bytes 0 through 2) mod 256]; must be ≤ 255)

Laser Align provides results of the image check.

4 Check Laser Align response:

- ◆ Response should be a 5-byte value, the first four bytes provide the actual sensor image value for the image quality parameter specified. (The fifth byte is the checksum).
- ◆ If response is anything other than a 5-byte data string, repeat **Steps 1** and **2**.

5 Repeat **Steps 3** and **4** for each actual image quality parameter value you want to read.

▼▼ To check sensor image quality against established image quality parameters

1 Make sure that upper and lower limits are set for the image quality parameters that are critical to your operation.

2 Send **IMG_CHK_CMD (43)** sequence:

43 n

where

n = the desired image check mode.

➲ See *IMG_CHK_CMD (43)* in Chapter 7 for more information about image check modes.

Laser Align checks the image quality.

3 Check Laser Align response:

- If response is **155 (IMG_CHK_ACK)**, the image check has successfully completed. Go to **Step 4**.
- If response is not **155 (IMG_CHK_ACK)**, repeat **Step 1**.

4 Request image status; send **READ_CMD (41)** sequence:

41 208 w y

where:

208 = image status index
w = window **0** or **1** (window 1 valid for SCM1 only)
y = checksum (256 **-** [(Σ bytes 0 through 2) mod 256]; must be ≤ 255)

Laser Align provides results of the image check against limits.

5 Check Laser Align response (byte 0 of the 5-byte response provides status):

- If status is **1 (NO ERROR)**, the sensor is within the specified image quality parameter limits. You have completed the procedure.
- If the status is **105 (IMAGE CHECK FAIL)**, at least one upper or lower limit has been violated. Go to **Step 6**.
- If response is not **1 (NO ERROR)** or **105 (IMAGE CHECK FAIL)** go back to **Step 2** or take action based on the status received.

➲ See Image Quality Parameter Indexes in Chapter 8 for more information about image check parameters.

➲ See Chapter 9 for more information about status codes.

6 Check upper limit status; send **READ_CMD (41)** sequence:

41 192 w y

where:

192 = check upper limits index
w = window **0** or **1** (**window 1** valid for SCM1 only)
y = checksum (256 - [(Σ bytes 0 through 2) mod 256]; must be ≤ 255)

Laser Align reports whether any upper limits have been violated.

7 Check Laser Align response.

Bytes 0 and 1 of the 5-byte **READ_CMD (41)** response contain a bitmap that identifies which, if any, of the upper limits set for the image quality parameters have been violated (0 = pass; 1 = fail).

➲ See Table 8-5 in Chapter 8 for bitmap details.

◆ If all bits of the bitmap are **0**, no limits were violated. Go to **Step 8**.

◆ If the first bit of the bitmap is **1**, no limits were set. Depending upon your situation, you can go to **Step 8**, or you can go back to **Step 1** and set limits.

◆ If any bits of the bitmap, other than the first bit, are **1,** the associated image quality parameter limit has been violated. Depending on your situation, you can:

- Check and/or reset the limit using the procedures provided earlier in this section.
- Use a **READ_CMD (41)** with the appropriate index to obtain the actual value for the image quality parameter.
- Send a message to the system operator indicating that sensor window cleaning or sensor replacement may be necessary.
- Go to **Step 8.**

8 Check lower limits status; send **READ_CMD (41)** sequence:

41 176 w y

where:

176 = check lower limits index
w = window **0** or **1** (**window 1** valid for SCM1 only)
y = checksum (256 **-** [(Σ bytes 0 through 2) mod 256]; must be ≤ 255)

Laser Align reports whether any lower limits have been violated.

9 Check Laser Align response.

Bytes 0 and 1 of the 5-byte **READ_CMD (41)** response contain a bitmap that identifies which, if any, of the lower limits set for the image quality parameters have been violated (0 = pass; 1= fail).

➲ See Table 8-5 in Chapter 8 for bitmap details.

◆ If all bits of the byte 0 bitmap are **0**, no limits were violated. You have completed the procedure.

◆ If the first bit of the byte 0 bitmap is **1**, no limits were set. Depending upon your situation, you can consider the procedure complete, or you can go back to **Step 1** and set limits.

◆ If any bits of the byte 0 bitmap, except for the first bit, are **1,** the associated image quality parameter limit has been violated. Depending on your situation, you can:

- Check and/or reset the limit using the procedures provided earlier in this section.
- Use a **READ_CMD (41)** with the appropriate index to obtain the actual value for the image quality parameter.
- Send a message to the system operator indicating that sensor window cleaning or sensor replacement may be necessary.
- Consider the procedure complete.

Resetting Laser Align Firmware

All of the Laser Align variables can be set back to their defaults by resetting the firmware. This is a good idea if you have any questions about the current settings of your Laser Align system.

Note Resetting Laser Align firmware does not affect the programming within your host system.

▼▼ To reset the Laser Align firmware

1. Send **RESET_CMD (17)** sequence:

   17

   Laser Align returns a **RESET_ACK (129)**.

2. Wait for at least one second to allow Laser Align to finish initializing.

3. Send the command sequences required to restore your system settings and to perform the next desired operation.

   See Chapter 7 for details about **RESET_CMD (17).**

Resetting Laser Align Hardware (SCM2 only)

You can reset the Laser Align hardware only if you have an SCM2 with all of its connectors at one end. This action should be reserved for those situations where the system is completely locked up and resetting the firmware does not solve the problem. Typically, this kind of situation only occurs during integration/programming.

The hardware is reset using the reset connector. Pin 1 of the connector is the active low reset input. Pin 2 is ground.

You can reset the software using a TTL or CMOS logic line, or you can short it to ground (Pin 2) with a switch. The minimum pulse width required is 140 ms. It must be pulled below 0.8V and the low-level sink requirement is 600 µA

6

- See SCM Specifications in Chapter 14 for information about connector locations.

Chapter 7

Commands Reference

Working With Laser Align Commands

All Laser Align operations are initiated by commands from the host controller. The Laser Align system cannot measure a component or report any results until a command is received from the host system.

A command (CMD) followed by any required parameters is referred to as a *command sequence* in the procedures of this manual. Decimal notation of these commands is used throughout this manual.

After each command and all parameters are received, the Laser Align system responds by returning an acknowledgment (ACK) that will be either a single byte response or a data stream, depending upon the command.

There are four different types of commands:

- **Diagnostic commands** - provide status and troubleshooting information about the Laser Align system.
- **Initialization commands -** ensure that the Laser Align system is working and communicating properly.
- **Run commands** - perform scanning and data retrieval operations.
- **Setup commands -** provide Laser Align with parameter information to be used in scanning and retrieving/reporting data.

This chapter describes the Laser Align commands, their formats, and their usage. Information about the various parameter indexes and status codes used with the commands follows in Chapter 8 and Chapter 9.

A summary chart of the commands is provided first, organized by the decimal numbers used to identify each command. A breakdown of commands by command type, and detailed information on each command follows the table. Command details are organized alphabetically by command name.

Command Summary and Cross-Reference

Table 7-1 provides a list of all Laser Align commands.

Compatibility commands are indicated by gray blocks behind the decimal for the command. These commands are included only for backward compatibility with earlier versions of Laser Align firmware. The compatibility commands are documented in Chapter 10.

Table 7-1: Command Summary and Cross-Reference

Command decimal/name		Purpose	Command Type	# of Add'l Bytes	Response	See Page
16	HELLO_CMD	Verify communication with Laser Align.	Diagnostic	None	HELLO_ACK (128)	7-11
17	RESET_CMD	Initialize the Laser Align SCM after power-up or for error recovery.	Initialization	None	RESET_ACK (129)	7-26
19	RUN_CMD*	Enable external reset of encoders.	Run	1	RUN_ACK (131)	7-27
20	REPORT_CMD Replaced by: REPORTW_CMD (42)	Request the RESULT data for the first alignment angle.	Run	None	13-byte data structure	10-10

* Compatibility command for some uses.

Table 7-1: Command Summary and Cross-Reference (continued)

Command decimal/name		**Purpose**	**Command Type**	**# of Add'l Bytes**	**Response**	**See Page**
21	IMAGE_CMD	Return the raw sensor data collected for the previous ONCE_CMD (23) or IMG_CHK_CMD (43).	Diagnostic	None	Grayscale data	7-12
22	ZERO_CMD	Set current encoder counter to zero.	Run	None	ZERO_ACK (134)	7-34
23	ONCE_CMD	Gather information (status, angle, center or edge, width) of the part in its present position.	Run	1	ONCE_ACK (135)	7-18
25	TEST_CMD Replaced by: IMG_CHK_CMD (43)	Perform a self-test on the Laser Align sensor and SCM.	Diagnostic	None	TEST_ACK (137)	10-19
26	WINDOW_CMD	Change the active window.	Setup	1	WINDOW_ACK (138)	7-32
27	PARAM_CMD Replaced by: LOAD_CMD (40)	Write next byte of PARAM data for the current window.	Setup	1	PARAM_ACK (149)	10-7

Table 7-1: Command Summary and Cross-Reference (continued)

Command decimal/name		Purpose	Command Type	# of Add'l Bytes	Response	See Page
28	R_PARAM_CMD Replaced by: READ_CMD (41)	Read next byte of PARAM data for the current window.	Setup	1	One byte of the 13-byte data structure that identifies the measurement parameters	10-14
29	REPORTS_CMD Replaced by: REPORTW_CMD (42)	Request the RESULT data for the second alignment angle.	Run	None	13-byte data structure	10-12
30	SWEEP_CMD	Scan alignment of one or two parts at up to four angular orientations.	Run	None	SWEEP_ACK (142)	7-30
31	SCAN_CMD	Report raw angle and width data.	Diagnostic	1	Variable length data structure.	7-28
32	VERSION_CMD	Verify the version of firmware currently on the Laser Align SCM.	Diagnostic	None	13-byte data structure	7-31
33	POINTER_CMD Replaced by: LOAD_CMD (40)	Prepare the sensor to receive a multiple-byte value.	Setup	None	POINTER_ACK (145)	10-9

Table 7-1: Command Summary and Cross-Reference (continued)

Command decimal/name		Purpose	Command Type	# of Add'l Bytes	Response	See Page
34	VALUE_CMD Replaced by: LOAD_CMD (40)	Send the sensor one byte of a multiple-byte value.	Setup	1	VALUE_ACK (146)	10-21
35	PRESET_CMD	Set the encoder value to the value placed into the VALUE buffer.	Setup	1	PRESET_ACK (147)	7-20
38	EEPROM_CMD	Check the operating parameters of the Laser Align sensor.	Diagnostic	1	3-byte data structure	7-9
39	ERR_LVL_CMD Replaced by: LOAD_CMD (40)	Set the system error level (0 is the default).	Setup	1	ERR_LVL_ACK (149)	10-5
40	LOAD_CMD	Load the parameters for scanning and data retrieval operations.	Setup	7	LOAD_ACK (152)	7-16
41	READ_CMD	Read parameters.	Run	3	3-byte data structure	7-21
42	REPORTW_CMD	Read the results of scanning and data retrieval operations.	Run	2	16-byte data structure	7-23
43	IMG_CHK_CMD	Check light stripe illumination.	Diagnostic	1	IMG_CHK_ACK (155)	7-13

Commands By Type

Table 7-2 groups the Laser Align commands by type. Compatibility commands, documented in Chapter 10, are indicated by an asterisk (*).

Table 7-2: Commands by Command Type and Cross Reference

Command Type	Command	See Page
Diagnostic	EEPROM_CMD (38)	7-9
	HELLO_CMD (16)	7-11
	IMAGE_CMD (21)	7-12
	IMG_CHK_CMD (43)	7-13
	SCAN_CMD (31)	7-28
	TEST_CMD (25)*	10-19
	VERSION_CMD (32)	7-31
Initialization	RESET_CMD (17)	7-26
Run	ONCE_CMD (23)	7-18
	READ_CMD (41)	7-21
	REPORT_CMD (20)*	10-10
	REPORTS_CMD (29)*	10-12
	REPORTW_CMD (42)	7-23
	RUN_CMD (19)**	7-27
	SWEEP_CMD (30)	7-30
	ZERO_CMD (22)	7-34

* Compatibility commands
** Compatibility command for some uses

7

Table 7-2: Commands by Command Type and Cross Reference (continued)

Command Type	Command	See Page
Setup	ERR_LVL_CMD (39)	10-5
	LOAD_CMD (40)	7-16
	PARAM_CMD (27)*	10-7
	POINTER_CMD (33)*	10-9
	PRESET_CMD	7-20
	R_PARAM_CMD (28)*	10-14
	VALUE_CMD (34)*	10-21
	WINDOW_CMD (26)	7-32

* Compatibility commands
** Compatibility command for some uses

Laser Align Command Details

Each command name is followed by the command's decimal value in parentheses. The commands are organized alphabetically.

EEPROM_CMD (38)

Command Type: Diagnostic

EEPROM_CMD (38) reports the operating parameters of the Laser Align sensor. The EEPROM_CMD must be followed by a parameter that identifies which operating parameter is requested.

Table 7-3: EEPROM_CMD Format

Byte	Description	Valid Value(s)
0	Command byte	38 = EEPROM_CMD
1	Parameter (operating parameter)	0 = Sensor serial number 2 = Left edge of the usable image (raw pixel number) 3 = Right edge of the usable image (raw pixel number) 4 = Array length (in pixels) 5 = Pixel size (in nanometers) 9 = Offset angle of the light beam relative to the sensor case (MiniLAM only)

Response

EEPROM_CMD reports the data on the requested operating parameter in a three-byte structure (Table 7-4).

Table 7-4: EEPROM_CMD Response

Byte Number	Description
1	Status. Laser Align returns **1 (NO_ERROR)** when the operating parameter information was read correctly from the EEPROM. Otherwise, it returns an error code specifying why the EEPROM read failed. ➲ See Chapter 9 for more information on status codes.
2	The low byte of the requested value.
3	The high byte of the requested value.

Note If the EEPROM read failed (i.e., status is not equal to **1 (NO_ERROR)**), Laser Align returns a default value.

Additional Information

The operating parameters of the sensor are obtained from the sensor during execution of a **RESET_CMD (17)** and are stored locally on the SCM. The EEPROM_CMD retrieves this information from the SCM.

HELLO_CMD (16)

Command Type: Diagnostic

HELLO_CMD (16) queries the Laser Align system to verify that the host processor is able to communicate with the system.

If issued after SWEEP_CMD (30), HELLO_CMD aborts SWEEP_CMD if processing is not complete. Otherwise, the command has no effect on the Laser Align system.

Table 7-5: HELLO_CMD Parameter Settings

Byte	Description	Valid Value(s)
0	Command byte	16 = HELLO_CMD

Response

HELLO_ACK (128)

IMAGE_CMD (21)

Command Type: Diagnostic

IMAGE_CMD (21) returns the raw sensor image data collected for the previous ONCE_CMD (23) or IMG_CHK_CMD (43).

See ONCE_CMD (23) and IMG_CHK_CMD (43) later in this section for more information.

Table 7-6: IMAGE_CMD Format

Byte	Description	Valid Value(s)
0	Command byte	21 = IMAGE_CHK_CMD

Response

The image data from the last ONCE_CMD (23) or IMG_CHK_CMD (43) is transmitted in a block that is the length (in bytes) of the detector array.

The detector array length may be obtained by using EEPROM_CMD (38). Each byte of the data can range from 0 (no light detected at that pixel) to 255 (that pixel is saturated).

IMG_CHK_CMD (43)

Command Type: Diagnostic

Note IMG_CHK_CMD is valid for firmware versions 2.5 and greater. Values returned by firmware versions prior to 2.5 in response to IMG_CHK_CMD are meaningless.

IMG_CHK_CMD (43) returns laser stripe intensity quality, and optical system alignment and cleanliness data, by applying various algorithms to the detected image.

Before you send the IMG_CHK_CMD, you can use the LOAD_CMD (40) to set up the upper and lower limits for the laser stripe image check. If you do not set limits, the default limits to be checked for each parameter are the lowest and highest possible number that can be represented. Therefore, any limit that is to be checked should be specified during Laser Align initialization. A specified region of the sensor can also be identified, so that errors beyond the usable region are not detected.

The IMG_CHK_CMD expects one byte to specify the mode of operation.

The IMG_CHK_CMD acquires a frame of video data and processes that data. When the IMG_CHK_CMD is acquiring the frame of video data, the light stripe must be unobstructed. If the view is obstructed (e.g., there is a part in the light stripe) invalid data will be reported.

See *LOAD_CMD (40)* and *READ_CMD (41)* in this section for more information about the use of these commands.

Table 7-7: IMG_CHK_CMD Format

Byte	Description	Valid Value(s)
0	Command byte	43 = IMG_CHK_CMD
1	Parameter (mode)	0 = uses the window (pixel) edges of window 0 1 = uses the window (pixel) edges of window 1 16 = uses the edges as stored in the sensor EEPROM 17 = uses a region of the laser stripe specified previously with the LOAD_CMD (40). The left edge of the region of interest is specified with LOAD_CMD window 0 or 1, index 243. The right edge is specified with LOAD_CMD window 0 or 1, index 244. The value specified is in units of one-tenth (1/10) pixels or microns, depending on the setting of the REPORT_UNITS variable. 18 = same as value 17, but also updates the window edges for all windows to the region edges 19 = same as value 17, but also updates the window edges for all windows to the edges of illumination

Response

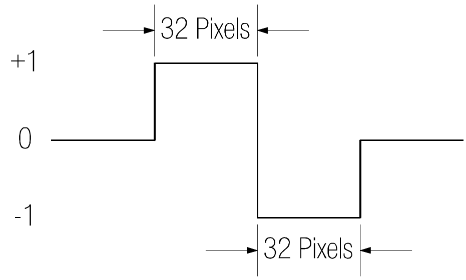
IMG_CHK_ACK (155)

After you send the IMG_CHK_CMD, you can send the READ_CMD (41) to examine the values calculated for each result.

Additional Information

The results of the IMG_CHK_CMD are calculated by processing the images described in Table 7-8.

Table 7-8: Images Processed by IMG_CHK_CMD

Image	Description
Background	The intensity level exceeded by 25% of the pixels.
Low-Pass-Filtered	The raw image convolved with a four-pixel rectangular kernel.
Median-Filtered	The raw image filtered with a 65-pixel sliding-median filter.
Mid-Frequency	The raw image convolved with the kernel shown below. 

LOAD_CMD (40)

Command Type: Setup

LOAD_CMD (40) sets the parameters that Laser Align uses during operation.

Table 7-9: LOAD_CMD Format

Byte	Description	Valid Value(s)
0	Command byte	40 = LOAD_CMD
1	Parameter index	See Chapter 8 for more information about parameter indexes.
2	Window	0 = window 0 parameters 1 = window 1 parameters 255= general system parameters
3	Parameter value (byte 0 LSB)	The meaning and valid values for bytes 3 through 6 depends on the parameter index set and window chosen using bytes 1 and 2. See Chapter 8 for values and information about parameter indexes.
4	Parameter value (byte 1)	
5	Parameter value (byte 2)	
6	Parameter value (byte 3 MSB)	
7	Checksum	256 - [Σ(bytes 0 to 6) mod 256]; must be $\leq$ 255

Response

LOAD_ACK (152)
BAD_PARAM (86)

Additional Information

As with the RESULT data, integer values are transmitted in two's-complement little-endian format. Thus, 254 (FE_{hex}) followed by 255, 255, 255 (FF_{hex}) would represent a -2 ($FFFFFFFE_{hex}$). The value 5000 (00001388_{hex}) would be transmitted as a 136 (88_{hex}), followed by a 19 (13_{hex}), followed by two zeros.

ONCE_CMD (23)

Command Type: Run

ONCE_CMD (23) initiates a single-frame measurement using a specified algorithm. It gathers information (status, angle, center or edge, and width) about the part in its current position. The command:

- Resets the RESULT data.
- Selects **window 0**.
- Reads the Laser Align sensor once using the **window 0** parameter data (except for the algorithm) and buffers RESULT data.

Note RESULT data is the information returned by Laser Align. It is accessed using the REPORTW_CMD (42).

➲ See *REPORTW_CMD (42)* for more information about the RESULT data.

ONCE_CMD requires a parameter identifying the measurement algorithm that is to be used.

Table 7-10: ONCE_CMD Parameter Settings

Byte	Description	Valid Value(s)
0	Command byte	23 = ONCE_CMD
1	Parameter (measurement algorithm)	11 = WIDTH_ALG Measure single frame using original edge finder 12 = LEFT_ALG Report component's left edge 13 = RIGHT_ALG Report component's right edge 14 = WIDE_ALG Measure single frame using optimized edge finder ➲ See *Measurement Algorithms* in Chapter 8 for more information.

Response

ONCE_ACK (135)

Additional Information

Unlike SWEEP_CMD (30), a ONCE_CMD is not aborted by a command sent immediately after the ONCE_CMD. It is not aborted because the ONCE_CMD measurement does not require any angular rotation and finishes in a short, fixed amount of time.

PRESET_CMD (35)

Command Type: Run

PRESET_CMD (35) sets the specified encoder equal to a value already placed in the value buffer by VALUE_CMD (34) or LOAD_CMD (40). Therefore, the VALUE_CMD or LOAD_CMD must precede the PRESET_CMD.

The PRESET_CMD expects one byte to specify the encoder.

- See *Synchronizing with the Encoder* in Chapter 5 for more information about the use of this command with LOAD_CMD (40).
- See Compatibility Procedures in Chapter 10 for more information on the use of this command with POINTER_CMD (33) and VALUE_CMD (34).

Table 7-11: PRESET_CMD Format

Byte	Description	Valid Value(s)
0	Command byte	35 = PRESET_CMD
1	Parameter (encoder)	0 = Encoder 0 1 = Encoder 1 (SCM1 only)

Response

PRESET_ACK (147)

READ_CMD (41)

Command Type: Run

READ_CMD (41) is used to read the parameters downloaded to the Laser Align SCM with LOAD_CMD (40) or to access data generated by Laser Align.

Table 7-12: READ_CMD Format

Byte	Description	Valid Value(s)
0	Command byte	41 = READ_CMD
1	Parameter index	See Chapter 8 for more information about parameter indexes.
2	Window	0 = window 0 parameters 1 = window 1 parameters 255= general system parameters
3	Checksum	256 - [Σ(bytes 0 to 2) mod 256]; must be ≤ 255

Response

Table 7-13: READ_CMD Response Format

Byte #	Description
0	Parameter Value (byte 0 LSB)
1	Parameter Value (byte 1)
2	Parameter Value (byte 2)
3	Parameter Value (byte 3 MSB)
4	256 - [Σ(bytes 0 to 3) mod 256]; must be ≤ 255

If an invalid checksum is received for the READ_CMD, or if the window or parameter index is invalid, the four bytes of the value and the checksum are all returned as 255.

See *Image Quality Parameter Indexes* in Chapter 8 for more information about how the READ_CMD results are reported when READ_CMD is used with IMG_CHK_CMD (43).

REPORTW_CMD (42)

Command Type: Run

REPORTW_CMD (42) requests the RESULT data generated by a previous command. RESULT data always includes status information, as well as any additional information generated by the preceding command. For example, a REPORTW_CMD issued after a SWEEP_CMD (30) would obtain RESULT data that would include the requested angle, center (or edge), and width information resulting from the measurement.

The REPORTW_CMD sequence must include parameters that identify the desired data, and the window for which RESULT data is to be reported.

Note Before issuing REPORTW_CMD, you must use LOAD_CMD (40) to set the measurement parameters and result units.

When using REPORTW_CMD (42) with the SWEEP_CMD (30), you can request any of the four possible results while the SWEEP_CMD is active or after it has completed.

The REPORTW_CMD is the only command that does not abort SWEEP_CMD. If REPORTW_CMD is sent with a "wait for" index while the system is processing a sweep, the system waits until it has calculated the requested result and then reports that result even if the SWEEP_CMD is still active.

The SWEEP_CMD is suspended during calculations, so early results should not be requested during two-window operation (SCM1 only).

➲ See *SWEEP_CMD (30)* later in this chapter for more information about its use.

Table 7-14: REPORTW_CMD Format

Byte	Description	Valid Value(s)
0	Command byte	42 = REPORTW_CMD
1	Result index	➲ See Table 1.
2	Window	0 = window 0 result data 1 = window 1 result data

Table 1: REPORTW_CMD Result Index Settings

Result #	Index		
	Wait for Angle Only	Wait for Angle, Center and Width	Do Not Wait — Report Results Immediately*
0	1	2	3
1	17	18	19
2	33	34	35
3	49	50	51

* Only valid for firmware versions 2.9 and higher.

Note Waiting for minima other than the final minimum can cause processing to overrun subsequent minima. Specifying any wait requires angle limits or timeouts to be implemented. For simplest operation, use the Do Not Wait indices for firmware versions 2.9 and later.

Response

The result data is reported in a 16-byte structure.

Table 7-15: RESULT Data

Byte Number	Description
0	Status
1 - 4	Align angle
5 - 8	Center position for measurement algorithms 11 and 14. Edge position for measurement algorithms 12 and 13. Zero for parameter 10 sent with RUN_CMD (19). ➲ See Chapter 8 for more information about measurement algorithms.
9 - 12	Width Zero for parameter 10 sent with RUN_CMD (19).
13	Result Index
14	Window
15	Checksum: 256 - [Σ(bytes 0 to 14) mod 256]; must be ≤ 255

With REPORTW_CMD, values are reported in the following way:

- The status code is an unsigned 8-bit value.

  ➲ Refer to Chapter 9 for a list of the status codes and their meanings.

- The calculated values of the angle, center (or edge), and width measurements are in two's complement 32-bit form with the LSB in the lowest address.
- The angle (or θ) value is reported in absolute encoder counts. It has 24 bits of significance and is sign-extended to 32 bits.
- The center and width values are reported in tenths of pixels or microns, depending on the units set using the LOAD_CMD (40).

RESET_CMD (17)

Command Type: Initialization

RESET_CMD (17) resets the Laser Align SCM to its initial state:

- Resets the RESULT data.
- Selects **window 0**.
- Reads calibration data from the sensor.
- Reports NO_ERROR (1) or LINK_FAIL (5) in **window 0, result 0** status. A LINK_FAIL (5) status in this context means that the calibration data was not read from the sensor. The specific reason for the failure can be obtained using EEPROM_CMD (38).
- Sets the error level to 0.
- Resets parameters set with LOAD_CMD (40) to defaults.
- Disables handshaking.
- Sets reporting units to “no units,” which must be set to either microns or 1/10 pixels using **LOAD_CMD (40)** before performing any other operation.

Note RESET_CMD must be executed after powering up Laser Align.

Table 7-16: RESET_CMD Format

Byte	Description	Valid Value(s)
0	Command byte	17 = RESET_CMD

Response

RESET_ACK (129)

Laser Align could be busy initializing for up to one second following the RESET_ACK. It is important to wait at least one second after receiving the RESET_ACK before sending any additional commands.

RUN_CMD (19)

Command Type: Run (with parameter **10**)

RUN_CMD (19) with parameter 10 initiates an external reset of the position counter.

The RUN_CMD:

- Resets the RESULT data.
- Selects **window 0**.

Table 7-17: RUN_CMD Format

Byte	Description	Valid Value(s)
0	Command byte	19 = RUN_CMD
1	Parameter	10 = REZERO_ALGExternal reset of position counter See Chapter 10 for information on the use of parameters 11, 12, 13, and 14 with this command.

Response

RUN_ACK (131)

Additional Information

For compatibility with firmware prior to version 1.0, RUN_CMD can be used to measure a rotating part by using the appropriate mode parameter. Versions 1.0 and newer of the firmware have replaced the need for this function with the SWEEP_CMD (30).

See Chapter 10 for more information about backward compatibility of this command.

SCAN_CMD (31)

Command Type: Diagnostic

SCAN_CMD (31) reads the raw rotational angle and light stripe shadow width obtained at each frame during a SWEEP_CMD (30).

SCAN_CMD requires a parameter that identifies the window and result for which you want information.

See *SWEEP_CMD (30)* later in this chapter for more information about the SWEEP_CMD.

Table 7-18: SCAN_CMD Format

Byte	Description	Valid Value(s)
0	Command byte	31 = SCAN_CMD
1	Parameter	0 = Window 0, Result 0 16 = Window 0, Result 1 32 = Window 0, Result 2 48 = Window 0, Result 3 1 = Window 1, Result 0 17 = Window 1, Result 1 33 = Window 1, Result 2 49 = Window 1, Result 3

Response

SCAN_CMD reports raw shadow widths in units of pixels, while all other commands report width in one-tenth pixels or microns. The rotational angles (in encoder counts) are truncated to 16 bits in the transmitted data.

Table 7-19: SCAN_CMD Response

Byte #	Description
0	Number of frames (LSB)
1	Number of frames (MSB)
2	Frame 0 angle (LSB)
3	Frame 0 angle (MSB)
4	Frame 0 width (LSB)
5	Frame 0 width (MSB)
6	Frame 1 angle (LSB)
7	Frame 1 angle (MSB)
8	Frame 1 width (LSB)
9	Frame 1 width (MSB)
and so on until (Number of frames) of data is transmitted	

Additional Information

SWEEP_CMD (30) fills a buffer with frame data consisting of the raw rotational angles and shadow widths. The buffer can hold data from 512 frames. SWEEP_CMD clears this buffer at the start of each scan.

With SCAN_CMD, you can read the buffer and obtain the data from the last 512 frames (or fewer, if the SWEEP_CMD did not fill the buffer). You can use the data to plot laser stripe shadow width versus rotational angle.

SCAN_CMD is a useful debugging tool for investigating the causes if Laser Align is unable to detect a minimum.

Note You must issue REPORTW_CMD (42) if you want to read the alignment angle calculated by Laser Align.

SWEEP_CMD (30)

Command Type: Run

As a component is rotated, SWEEP_CMD (30) measures the angle, center (or edge), and width for up to four orientations. SWEEP_CMD uses parameters previously defined with LOAD_CMD (40) to determine the results of the measurement. SWEEP_CMD:

- Resets the RESULT data.
- Selects **window 0** as the active window.
- Scans the part alignment as configured in the parameter table for each window, as sent using the LOAD_CMD.
- Updates RESULT data.

Alignment data is written to the RESULT structures. It can be retrieved by using REPORTW_CMD (42).

With the exception of REPORTW_CMD (42), any command sent while SWEEP_CMD is processing aborts SWEEP_CMD.

Table 7-20: SWEEP_CMD Format

Byte	Description	Valid Value(s)
0	Command byte	30 = SWEEP_CMD

Response

SWEEP_ACK (142)

To obtain the RESULT data calculated by the SWEEP_CMD, you must issue a REPORTW_CMD (42).

See *LOAD_CMD (40)* and *REPORTW_CMD (42)* in this chapter for more information about these commands and the RESULT structure.

VERSION_CMD (32)

Command Type: Diagnostic

VERSION_CMD (32) returns the version of firmware currently on the Laser Align SCM.

Table 7-21: VERSION_CMD Format

Byte	Description	Valid Value(s)
0	Command byte	32 = VERSION_CMD

Response

The command returns a 13-byte, null-terminated string of ASCII characters that indicates the firmware version.

WINDOW_CMD (26)

Command Type: Run

WINDOW_CMD (26) changes the active window. It must be accompanied by a parameter that identifies **window 0** or **window 1** as the window to which subsequent commands will apply. If an invalid window is specified, **window 0** will be selected.

Table 7-22: WINDOW_CMD Parameter Settings

Byte	Description	Valid Value(s)
0	Command byte	26 = WINDOW_CMD
1	Parameter	0 = window 0 1 = window 1 (SCM1 only)

Response

WINDOW_ACK (138)

Additional Information

WINDOW_CMD is only used with ZERO_CMD (22) and the compatibility commands in Chapter 10.

The SCM2 has only one window for measurement (**window 0**).

Several commands automatically select window 0 as the active window. If you want to change to window 1, make sure you issue the WINDOW_CMD with a window 1 parameter only after you have issued all commands that automatically select window 0.

The commands that automatically select **window 0** are:

- ONCE_CMD (23)
- RESET_CMD (17)
- RUN_CMD (19)
- SWEEP_CMD (30)

See *Window Identification* in Chapter 5 for more information about windows.

ZERO_CMD (22)

Command Type: Run

ZERO_CMD (22) resets the position counter for the current window. This command can be used to set a zero position independent of an encoder's zero reference signal, or in a stepper motor system.

Table 7-23: ZERO_CMD Format

Byte	Description	Valid Value(s)
0	Command byte	22 = ZERO_CMD

Response

ZERO_ACK (134)

- See *Synchronizing with the Encoder* in Chapter 5 for more information about the use of this command.

Chapter 8

Parameter Indexes Reference

Overview

There are a number of indexes used with LOAD_CMD (40) and READ_CMD (41) that tell Laser Align what data parameter you are setting (LOAD_CMD) or reading (READ_CMD). The values that identify these indexes are often tied to the associated window parameter in the command sequence.

The following tables identify the available indexes. They are grouped according to the parameters they define:

- General system parameters
- Scanning (or measurement) parameters
- Image quality parameters

After the table, further information is provided about the image quality results, and about the scanning parameter indexes, their meanings, and their values.

See Chapter 7 for details about the command structures that use the indexes.

General System Parameter Indexes

The following indexes are used with LOAD_CMD (40) or READ_CMD (41) to define/read the general system parameters that control and monitor Laser Align operation. The various parameter indexes use either **window 255** or **window 0.**

Table 8-1: Indexes and Related Settings for General System Parameters (Window 255)

Parameter	Index	Settings
Units (required)	0	0 = center and width values are reported in 1/10 pixels 1 = center and width values are reported in microns *default* = none; must be set by user.
Flow Control	1	0 = disables flow control (RTS/CTS ignored) 1 = enables flow control *default* = 0
Error Level	2	0, 1, 2, or 3 = error level 0, 1, 2, or 3 *default* = 0

See *Defining General System Parameters* in Chapter 5 for more information about using these indexes with Laser Align.

8

Table 8-2: Indexes and Related Settings for General System Parameters (Window 0)

Parameter	Index	Settings
Value Buffer	247	Any 32-bit number. ➲ See *Synchronizing with the Encoder* in Chapter 5 for more information.
Link Fail	248	Read-only. Number of Link Fail occurrences, including those outside of monitored measurements.
Overcurrent	249	Read-only. Number of Overcurrent occurrences outside of monitored measurements for SCM2 systems. SCM1 returns a value of -1.
Power Low	250	Read-only. Number of Power Low occurrences outside of monitored measurements for SCM2 systems. SCM1 returns a value of -1.
Power Fail	251	Read-only. Number of Power Fail occurrences outside of monitored measurements for SCM2 systems. SCM1 returns a value of -1.
Part Number	252	Read-only. SCM2 part name. SCM1 returns a value of -1.
Serial Number	253	Read-only. SCM2 serial number. SCM1 returns a value of -1.
Boot ROM Checksum	254	Read-only. SCM1 Boot ROM Checksum. SCM2 returns a value of 0.
Application ROM Checksum	255	Read-only. Application ROM Checksum for SCM1 systems. Application CRC for SCM2 systems.

Scanning Parameter Indexes

The following indexes are used to define the parameters used in performing the scanning operations initiated by SWEEP_CMD (40) and RUN_CMD (19).

Table 8-3: Indexes and Related Settings for Scanning Parameters (Windows 0 and 1)

Parameter	Index				Settings
	Result 0	Result 1	Result 2	Result 3	
Holdoff Angle	0	16	32	48	Non-negative number, regardless of rotation direction. *default* = 0
Algorithm	1	17	33	49	7, 11, -11, 12, -12, 13 -13, 14, -14 *default* = 7 See *Measurement Algorithms* later in this chapter for more information.
Angle Limit	2	18	34	50	Non-negative number, regardless of rotation direction. *default* = infinity (∞)

8

Table 8-3: Indexes and Related Settings for Scanning Parameters (Windows 0 and 1) (continued)

Parameter	Index				Settings
	Result 0	Result 1	Result 2	Result 3	
Left Edge of Window	240				*pixel number:* 0 through (detector array length - 1) *default* = read from sensor calibration data
Right Edge of Window	241				
Encoder Number	242				0 = encoder 0 (valid for SCM1 and SCM2) 1 = encoder 1 (valid for SCM1 only) *default* = 0

Image Quality Parameter Indexes

The following indexes are used to set and read image quality information associated with the IMAGE_CHK_CMD (43). The upper and lower limits indexes are used with both the LOAD_CMD (40), and the READ_CMD (41); while the actual value indexes are valid only with the READ_CMD (41).

Table 8-4: Indexes for Image Quality Parameters

Parameter Name	Actual Value Index	Lower Limit Index	Upper Limit Index
Image Status	208	N/A	N/A
Check Upper Limits	192	N/A	N/A
Check Lower Limits	176	N/A	N/A
Dark Level	209	177	193
Min Pixel	210	178	194
Min Low Pass	211	179	195
Background	212	180	196
Max Low Pass	213	181	197
Max Pixel	214	182	198
RMS Mid Frequency	215	183	199
Max Mid Frequency	216	184	200
RMS High Frequency	217	185	201
Absolute High Frequency	218	186	202
Even-Odd	219	187	203
Non-Flat	220	188	204

8

Image Quality Results

When a READ_CMD (41) is sent with an image quality parameter index of **208 (Check Status)**, Laser Align will return a status based on the last IMG_CHK_CMD (43) results:

- A status of **1 (NO ERROR)** means that the sensor passed whatever limits were established.
- A status of **105 (IMAGE CHECK FAIL)** means that at least one limit has been violated.

When a READ_CMD (41) is sent with an image quality parameter index of **192 (Check Upper Limits)** or **176 (Check Lower Limits)**, Laser Align returns the IMG_CHK_CMD (43) result as a bitmap value that is detailed in Table 8-5. In the bitmap, **0=pass** and **1=fail** for each of the specified parameters.

Table 8-5: Bitmap for READ_CMD (41) Response Value for Check Upper and Check Lower Limits

Bit Number	Meaning
0	Bit zero will be 1 if no limits were set up by the user to be checked against.
1	Dark Level
2	Min Pixel
3	Min Low Pass
4	Background
5	Max Low Pixel
6	Max Pixel
7	RMS Mid Frequency

Table 8-5: Bitmap for READ_CMD (41) Response Value for Check Upper and Check Lower Limits (continued)

Bit Number	Meaning
8	Max Mid Frequency
9	RMS High Frequency
10	Absolute High Frequency
11	Even - Odd
12	Non-Flat

In Table 8-6, the parameter name, data format, and description obtained using READ_CMD (41) with any image quality parameter index from Table 8-3 other than **192 (Check Upper Limits)** and **176 (Check Lower Limits)** are given. The 8.8 format indicates that the eight most significant bits (Byte 1) represent the integer portion of the result and the eight least significant bits (Byte 0) represent the fractional portion of the value.

- See *READ_CMD (41)* in Chapter 7 for more information about its use and syntax.

- See *IMG_CHK_CMD (43)* in Chapter 7 for more information about the image quality results and how they are calculated

Table 8-6: Image Quality Results

Parameter	Format	Description
Image Status	Laser Align Status Code	Check for image limits violations.
Check Upper Limits	Bitmap to READ_CMD result format (See Table 8-5)	Identify any upper limits that the image violates.
Check Lower Limits	Bitmap to READ_CMD result format (See Table 8-5)	Identify any lower limits that the image violates.

Table 8-6: Image Quality Results (continued)

Parameter	Format	Description
Dark Level	Pixel intensity value (unsigned 8 bit)	Not implemented.
Min Pixel	Pixel intensity value (unsigned 8 bit)	Minimum raw (unfiltered) pixel intensity value.
Min Low Pass	Pixel intensity value (unsigned 8 bit)	Minimum pixel intensity value in low-pass filtered image.
Background	Pixel intensity value (unsigned 8 bit)	The gray level exceeded by 25% of the pixels. This is the gray level representation of the illuminated regions of the image, with no obstruction.
Max Low Pass	Pixel intensity value (unsigned 8 bit)	Maximum pixel intensity value in low-pass filtered image.
Max Pixel	Pixel intensity value (unsigned 8 bit)	Maximum raw (unfiltered) pixel intensity value.
RMS Mid Frequency	Calculated value in 8.8 format	RMS value of mid-frequency noise in the image. This summary measure correlates well with actual edge-finder errors.
Max Mid Frequency	Calculated value in 8.8 format	Maximum of the absolute values of the mid-frequency noise in the image.
RMS High Frequency	Relative to calculated background level (8.8)	RMS value of the raw image minus the median filtered image. This parameter and the **Absolute High Frequency** value indicate the amount of dirt on the optical surfaces.

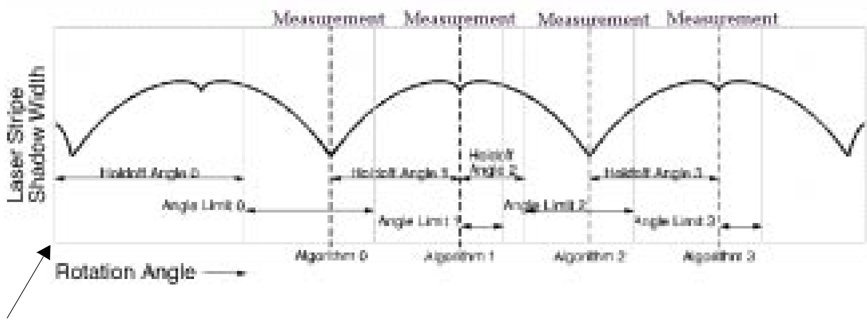
Table 8-6: Image Quality Results (continued)

Parameter	Format	Description
Absolute High Frequency	Relative to calculated background level (8.8)	Average of the absolute values of the raw image minus the median filtered image. This parameter and the **RMS High Frequency** value indicate the amount of dirt on the optical surfaces.
Even-Odd	Relative to calculated background level (8.8)	The CCD array in some sensors has two channels: one for the even pixels and one for the odd pixels. The even-odd value is a measure of internal channel balance. The two channels pass through separate amplifiers before being converted. The displayed even-odd value is the difference between even (e) and odd (o) pixels divided by their sum: $(\Sigma e - \Sigma o)/(\Sigma e + \Sigma o)$.
Non-Flat	Relative to calculated background level (8.8)	RMS value of the median-filtered image values minus the background level. The non-flat value is a measure of overall non-flatness of the image. Errors in the sensor's optical alignment are detected by this summary measure.

Scanning Parameter Details

The use of the various scanning parameter indexes set with LOAD_CMD (40) and read using READ_CMD (42) are illustrated in Figure 8-1. Additional details about the various scan parameters follow.

Figure 8-1: How parameters are used during SWEEP_CMD execution



SWEEP_CMD issued

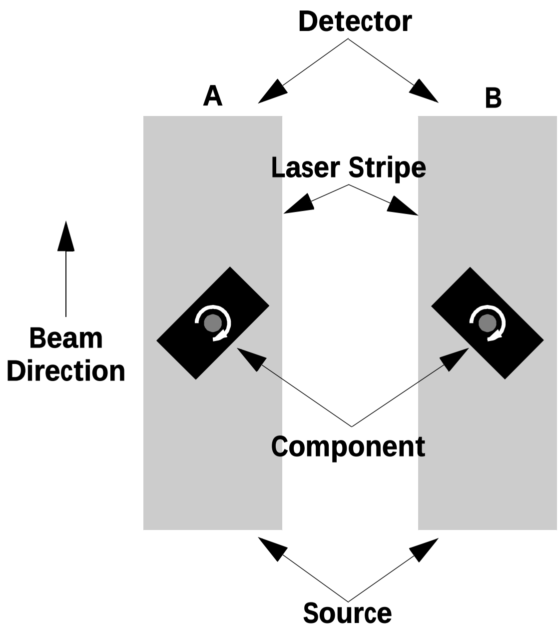
Holdoff Angle

The holdoff angle specifies the delay in encoder counts between the encoder value at the previous alignment angle [or the issuance of the SWEEP_CMD (30)] and:

- The position at which the measurement process begins for the next alignment angle, if you are using a positive measurement algorithm (Results 0 and 2 in Figure 8-1).
- The position at which the measurement is taken for the next alignment angle, if you are using a negative measurement algorithm (Results 1 and 3 in Figure 8-1).

The first holdoff angle (Holdoff Angle 0) is commonly used to wait until the component has been rotated so that the first minimum found is the lesser minimum. The first holdoff angle depends on the angle at which the component is picked up. As illustrated in Figure 8-2, if the component is picked up such that the first minimum found is the larger minimum (A), you should use a holdoff angle of 90° plus twice the pickup error. However, if the first minimum found is the lesser minimum (B), you can use a small holdoff angle.

Figure 8-2: Holdoff Angle



Valid parameter values consist of any non-negative number (regardless of rotational direction). The default holdoff angle is 0.

Angle Limit

The angle limit parameter specifies the number of encoder counts from the current start of measurement, until the Laser Align stops looking for a local minimum shadow width and reports an error of **98 (SCAN LIMIT)**.

Left and Right Edges

The edges of a window define the portion of sensor data to be processed during SWEEP_CMD. Any sensor data between the left and right edges is considered part of the window's data. A window can cover the entire range of the detector or just a portion of it. The default window covers the entire illuminated detector range.

Note If you use **IMG_CHK_CMD (43)**, the minimum window size is 65 pixels. If the window is set to a smaller size, a status of **78 (SMALL WINDOW)** is returned.

For SCM1 only: Because two windows are allowed, you can define different left and right edges for each window. The windows can use overlapping portions of the sensor. For example, window 0 could have a left edge of 8 and a right edge of 1000, and window 1 could have a left edge of 800 and a right edge of 2500. Generally, however, this is improper, since this would imply that the parts being measured could collide during rotation or that their shadows could overlap.

Encoder (SCM1 only)

The encoder parameter specifies which encoder input is used for alignment angle, holdoff angle, and angle limit data. Placement heads accommodating two parts may use separate angular encoders for the two parts by selecting **encoder 0** for one window and **encoder 1** for the other.

Measurement Algorithms

The measurement algorithm selects the alignment criterion and image processing technique to be used for a portion of the alignment scan. Not all measurement algorithms are valid for each of the scanning commands. Table 8-7 identifies the numbers or *modes* used to select the various algorithms, as well as a description of each algorithm, how it works, and the commands for which it is valid.

A definition that may be useful while using the table is:

- **Optimized edge finder -** measurement method that tolerates noise associated with part edges so that cleaner measurement data can be collected.

Table 8-7: Measurement Algorithms

Algorithm Mode	Description	Use
7	No algorithm	Used to specify that you do not want a minimum to be calculated. This mode specifies that there is no algorithm for the alignment scan. As a result, Laser Align disables scanning for that minimum. The holdoff angle is added to subsequent scans. Mode 7 is the default algorithm. **Valid for:** SWEEP_CMD (30)
11	Measure using single frame	Laser Align looks for a minimum, then calculates component position using only the single image frame at that minimum. Measurement mode 11 is faster than mode 14; mode 14 requires about 20 ms to compute a measurement while mode 11 requires only 10 ms. **Valid for:** ONCE_CMD (23), SWEEP_CMD (30)
-11	Measure using single frame (one frame only)	Works like positive measurement mode 11, but adds the holdoff angle to the encoder position of the frame with the smallest measured width from the previous minimum, and calculates component position using the image frame at the resulting encoder position. **Valid for:** SWEEP_CMD (30)

Table 8-7: Measurement Algorithms (continued)

Algorithm Mode	Description	Use
12	Measure left	Finds an alignment angle by using the left edge of a component which possibly extends out of the light stripe to the right. To fully utilize this mode, you must ensure that the component center is offset from the center of the light stripe sufficiently for the component to block a right edge. **Valid for:** ONCE_CMD (23), RUN_CMD (19), SWEEP_CMD (30)
-12	Measure left (one frame only)	Works like measurement mode 12, but adds the holdoff angle to the encoder position of the frame with the smallest measured width from the previous minimum, and calculates component position using the image frame at the resulting encoder position. **Valid for:** SWEEP_CMD (30)
13	Measure right	Finds an alignment angle by using the right edge of a component which possibly extends out of the light stripe to the left. To fully utilize this mode, you must ensure that the component center is offset from the center of the light stripe sufficiently for the component to block a left edge. **Valid for:** ONCE_CMD (23), RUN_CMD (19), SWEEP_CMD (30)

Table 8-7: Measurement Algorithms (continued)

Algorithm Mode	Description	Use
-13	Measure right (one frame only)	Works like measurement mode 13, but adds the holdoff angle to the encoder position of the frame with the smallest measured width from the previous minimum, and calculates component position using the image frame at the resulting encoder position. **Valid for:** SWEEP_CMD (30)
14	Measure averaging 3 frames	Like mode 11, it finds a minimum at which to collect component measurement data; however, unlike mode 11, it averages three frames of data to determine the measurement. This makes measurements more repeatable. Mode 14 uses an optimized edge finder that tolerates excess noise, which is a common problem in large components (such as PLCC-84 components) and leaded components. Mode 14 results in better accuracy on all components. It is the only recommended algorithm for leaded components, which typically require better accuracy. Measurement mode 14 is slower than mode 11; mode 14 requires about 20 ms to compute a measurement while mode 11 requires about 10 ms. **Valid for:** ONCE_CMD (23), RUN_CMD (19), SWEEP_CMD (30)

Table 8-7: Measurement Algorithms (continued)

Algorithm Mode	Description	Use
-14		Works like measurement mode 14, but adds the holdoff angle to the encoder position of the frame with the smallest measured width from the last minimum, and processes only a single frame of image data for component position measurement (as occurs when ONCE_CMD is used with positive 14). **Valid for:** SWEEP_CMD (30)

Choosing the Appropriate Measurement Algorithm

Choosing the proper component measurement algorithm ensures that Laser Align correctly determines orientation. Table 8-8 presents information on selecting the correct algorithm for your operation.

Table 8-8: Guidelines for Selecting a Measurement Algorithm

Algorithm	Guidelines for Use
11 (Measure Using Single Frame)	Do not use this algorithm for components with leads.
12	Use this algorithm if the component cannot fit within the light stripe and extends to the right of the light stripe.*
13	Use this mode if the component cannot fit within the light stripe and extends to the left of the light stripe.*
14 (Measure Averaging 3 Frames)	Use this mode for the most repeatable results.

* Use mode 12 or 13 if a part might not fit entirely within the light stripe and if aligning to only one edge of the component is acceptable. If the part does fit entirely within the light stripe, the part will be aligned using both edges even if you select mode 12 or 13. To fully utilize either of these modes, you must ensure that the component center is offset from the center of the light stripe.

Table 8-9 details the relationship of the algorithm number to the gray-scale edge finder and frames used for processing.

Table 8-9: Algorithm Relationships

		Search for Minimum Width			Process One Frame Only at Specified Encoder Position		
	Algorithm	**11**	**12, 13**	**14**	**-11**	**-12, -13**	**-14**
Edge Finder	Original	X			X		
	Optimized		X	X		X	X
Position Calculation	Single Frame	X	X		X	X	X
	Average 3 Frames			X			
Width Calculation	Single Frame	X	X	X	X	X	X

Chapter 9

Status Code Reference

Status Codes and Error Levels

Some Laser Align commands return a status code as the first byte of their response. The status code indicates whether the operation just performed was successful or whether an error condition exists.

Note You must send a REPORTW_CMD (42) in order to receive a status for those commands that do not return status codes as part of their response structure.

Mapping Status Codes to Error Levels

There are multiple error levels available in Laser Align. The error level you set determines how Laser Align reports errors using the status codes. The higher the number of the error level, the more specific the status codes.

If you set an error level that is lower than what is available in the firmware version you are using (e.g., setting error level 0 when you have version 2.7 of the firmware, which supports error level 3), errors will be reported using only those status codes that are valid for the lower error level. Since the lower error levels do not have all of the status codes generated by the most recent firmware, the more specific status codes generated by the newest firmware are mapped backwards to codes that exist at the lower level.

This allows newer firmware to be backward compatible with older versions, but it can hide a clear indication of what is causing a problem. As a result, it is important to always set the error level to the highest available level supported by your application.

➲ See *Defining General System Parameters* in Chapter 5 for more information on how to set error levels and on which firmware versions support which error levels.

Summary of Status Codes

Table 9-1 summarizes Laser Align status codes. It is organized numerically, and identifies each status code, what it means, what actions to take in response to the status, and the error levels for which the status is valid. It also shows the status codes that newer status codes map backward to at lower error levels.

Table 9-1: Laser Align Status Codes

Status code (decimal)	Name	Description/Action to Take	Valid for error level(s):	Maps Backward to Status Code X at Error Level 0	1	2
1	OK	The command executed without errors.	0, 1, 2, 3			
2	Laser Fail	The laser diode failed. Contact CyberOptics for assistance.	0, 1, 2, 3			
3	No FIFO Data (SCM1 only)	The sensor is not connected or a hardware failure occurred. Power down, check the cable connections and reboot. If repeated attempts fail to clear this error, call CyberOptics for assistance.	0, 1, 2, 3			
4	No RAM Data	The sensor is not connected or a hardware failure occurred. Power down, check the cable connections, and reboot. If repeated attempts fail to clear this error, call CyberOptics for assistance.	0, 1, 2, 3			

Table 9-1: Laser Align Status Codes (continued)

Status code (decimal)	Name	Description/Action to Take	Valid for error level(s):	Maps Backward to Status Code X at Error Level		
				0	1	2
5	Link Fail (sensor data error)	A data error has occurred on the communication link from the Laser Align sensor to the SCM. A link error may be caused by electrical noise or a broken connection to the sensor. Refit the sensor cable connections and execute the command again. If the error is repeated, contact CyberOptics for assistance.	0, 1, 2, 3			
6	Dirty Sensor Windows	The sensor windows are dirty. Clean the sensor windows and initialize the sensor by sending a RESET_CMD (17).	0, 1, 2, 3			
7	No Algorithm	The NO_ALG action was requested. No results are valid. Set a valid algorithm and sweep again.	0, 1, 2, 3			
8	No Start	Either the sensor is not connected or there is a hardware error. Power down, check the cable connections, and power cycle the system.	0, 1, 2, 3			
9	FIFO Error (SCM1 only)	There was a hardware failure on the SCM. Power cycle the system.	0, 1, 2, 3			
10	Sensor Error	An unspecific hardware error related to the sensor/coaxial link occurred. Reboot the Laser Align system.	1, 2, 3	5		

Table 9-1: Laser Align Status Codes (continued)

Status code (decimal)	Name	Description/Action to Take	Valid for error level(s):	Maps Backward to Status Code X at Error Level		
				0	1	2
11	SCM Error	An unspecific hardware error related to the CPU board of the SCM occurred. Power cycle the Laser Align system.	1, 2, 3	8		
12	Bad EEPROM Version	The firmware does not recognize the sensor EEPROM version. A firmware upgrade is needed. Contact CyberOptics for assistance.	2, 3	5	10	
13	Bad EEPROM Data	There is an error reading calibration data from the sensor EEPROM. Try cycling power to the system. If the error persists, contact CyberOptics for assistance.	2, 3	5	10	
14	Bad EEPROM Sync	There is an error reading calibration data from the sensor EEPROM. Try cycling power to the system. If the error is repeated, contact CyberOptics for assistance.	2, 3	5	10	

Table 9-1: Laser Align Status Codes (continued)

Status code (decimal)	Name	Description/Action to Take	Valid for error level(s):	Maps Backward to Status Code X at Error Level 0	1	2
15	Power Low (SCM2 only)	The incoming voltage level is below what is required for proper sensor operation. ◆ For 18 volt sensors, this error occurs at some voltage less than 17 volts. ◆ For 24 volt sensors, this error occurs at some voltage less than 22 volts. Adjust the power as required.	3	5	5	5
16	Power Fail (SCM2 only)	The incoming voltage level is below what is required to power the logic on the SCM2. This error occurs when the voltage is less than that required for the 5-volt regulator to operate reliably. Adjust the power as required.	3	5	5	5
17	Over Current (SCM2 only)	The solid state circuit breaker has tripped, and power to the sensor is turned off. Send a RESET_CMD to restore power and read the sensor calibration data. ONCE_CMD and SWEEP_CMD also restore power, but immediately enter a data collection function that will fail the first time after the circuit breaker has tripped.	3	5	5	5

Table 9-1: Laser Align Status Codes (continued)

Status code (decimal)	Name	Description/Action to Take	Valid for error level(s):	Maps Backward to Status Code X at Error Level 0	1	2
64	No Part	Laser Align did not detect a component. Check for component presence and check component position.	0, 1, 2, 3			
65	Too Slow (Processor Unable to Keep Up with Frame Rate)	The Laser Align processor was unable to keep up with the frame rate. The processor could not read the incoming data as fast as necessary to evaluate it. This error is probably caused by too many transitions in the component image, possibly caused by an incorrectly positioned component. Check component location first. If measuring a BGA or PGA, try increasing rotation speed.	0, 1, 2, 3			
66	Start Min (Minimum Width Occurred at Start of Scan)	Laser Align did not detect a minimum component width. This error typically occurs for one of two reasons: the component was not properly rotated prior to measurements or the component has rounded edges (and therefore has no minimum width). Adjust prerotation angle.	0, 1, 2, 3			

Table 9-1: Laser Align Status Codes (continued)

Status code (decimal)	Name	Description/Action to Take	Valid for error level(s):	Maps Backward to Status Code X at Error Level 0	1	2
67	Aborted	Laser Align received a command that aborted the measurement scan before the sensor was able to detect a minimum component width. Check motion control and prerotation	0, 1, 2, 3			
68	Big Part	The component extended beyond one or both edges of the window. Try using a left or right edge measurement algorithm (12, 13).	0, 1, 2, 3			
70	Algorithm Error	The firmware detected a state that should never happen for the highest current error level. Verify that another error is not being hidden by error code mapping for lower error levels; make sure Laser Align is running at the highest error level the firmware version will support. If problems persist, call CyberOptics for assistance.	0, 1, 2, 3			

Table 9-1: Laser Align Status Codes (continued)

Status code (decimal)	Name	Description/Action to Take	Valid for error level(s):	Maps Backward to Status Code X at Error Level 0	1	2
71	Too Many Edges (SCM1 only)	Too many edges occurred in the component image; as a result, the Laser Align processor was unable to determine the component orientation. An edge is defined as either a light-to-dark or a dark-to-light transition of the CCD video signal. The Laser Align system currently allows a maximum of 128 edges, but may not have time to process that many, resulting in a TOO_SLOW (65) status code. This error is generally caused by an incorrectly positioned component. For example, if a PLCC is positioned too high (so that the laser stripe hits the leads but not the body of the component), two edges of each lead are imaged onto the sensor's detector, resulting in this error. If the component is positioned properly, the component body blocks the extra edges. Low laser intensity (near the threshold level) may also cause this error.	0, 1, 2, 3			

Table 9-1: Laser Align Status Codes (continued)

Status code (decimal)	Name	Description/Action to Take	Valid for error level(s):	Maps Backward to Status Code X at Error Level		
				0	1	2
72	More Pages	There was an SCM firmware or hardware failure resulting from an attempt to execute a window command. The algorithm sequence specified requires more windows than are available. The SCM2 is not capable of using multiple windows. ➲ See *Window Identification* in Chapter 5 for more information about windows. Verify the algorithm and window parameters. If the problem persists, contact CyberOptics for assistance.	0, 1, 2, 3			
73	More Counters	The encoder number is invalid. ◆ For SCM1, the encoder number must be 0 or 1. ◆ For SCM2, the encoder number can only be 0. Check to make sure that you have defined your encoder correctly using the LOAD_CMD (40). ➲ See *Encoder Inputs* in Chapter 5 for more information.	0, 1, 2, 3			

Table 9-1: Laser Align Status Codes (continued)

Status code (decimal)	Name	Description/Action to Take	Valid for error level(s):	Maps Backward to Status Code X at Error Level 0	1	2
74	More Windows	The window number is invalid. ◆ For SCM1, the window number must be 0 or 1. ◆ For SCM2, the window number can only be 0. Check to make sure that you have defined your windows correctly using LOAD_CMD (40). ➲ See *Window Identification* in Chapter 5 for more information.	0, 1, 2, 3			
75	Previous Error	During SWEEP_CMD, an error occurred on the first, second or third algorithm, preventing proper execution of the second, third or fourth algorithm, respectively. The error is only reported for the second, third or fourth result. ***For example:*** An accurate result is reported for Result 0, but an error occurs with the algorithm for Result 1. Result 1 would show an error status, and a 75 would be reported for Result 2 and Result 3. Check the previous result.	0, 1, 2, 3			

Table 9-1: Laser Align Status Codes (continued)

Status code (decimal)	Name	Description/Action to Take	Valid for error level(s):	Maps Backward to Status Code X at Error Level 0	1	2
76	Width Error (Edge Error for Algorithm 11)	The gray scale edge on the detector did not have enough contrast to find the actual component edge for algorithm 11. Possible causes are either a poor component shadow image (where the component did not completely block the laser beam) or an incorrectly specified algorithm, such as using mode 11 (Measure Using Single Frame) when mode 14 (Measure Averaging 3 Frames) should be used. ➲ Refer to *Measurement Algorithms* in Chapter 8 for more information on these algorithms.	1, 2, 3	68		
77	Wide Error (Edge Error for Algorithms 12, 13 and 14)	The gray scale edge on the detector did not have enough contrast to find the actual component edge for algorithms 12, 13 and 14. Possible cause is a poor component shadow image (e.g., the component did not completely block the laser beam). ➲ Refer to *Measurement Algorithms* in Chapter 8 for more information on these algorithms.	1, 2, 3	68		

Table 9-1: Laser Align Status Codes (continued)

Status code (decimal)	Name	Description/Action to Take	Valid for error level(s):	Maps Backward to Status Code X at Error Level 0	1	2
78	Small Window	The window parameters specify a window that is below the window size threshold. The window size must be larger than 65 pixels. Check to make sure that you have defined your windows correctly using LOAD_CMD (40). ➜ See *Window Identification* in Chapter 5 for more information.	1, 2, 3	68		
79	Window Blocked	The entire window is blocked. Check the sensor for foreign objects. This error can also be caused by a low laser intensity.	1, 2, 3	68		
80	Left Edge of Window Blocked	The left edge of the window is blocked. Check the position of the part, especially at the left edge. This error can also be caused by a low laser intensity.	1, 2, 3	68		
81	Right Edge of Window Blocked	The right edge of the window is blocked. Check the position of the part, especially at the right edge. This error can also be caused by a low laser intensity.	1, 2, 3	68		

Table 9-1: Laser Align Status Codes (continued)

Status code (decimal)	Name	Description/Action to Take	Valid for error level(s):	Maps Backward to Status Code X at Error Level 0	1	2
82	Both Edges of Window Blocked	Both edges of the active window are blocked. Check the position of the part, especially at the edges. This error can also be caused by a low laser intensity.	1, 2, 3	68		
83	Bad Checksum	The firmware found an error while reading the sensor EEPROM. Send the RESET_CMD (17) again. If the same error code is received again, call CyberOptics.	1, 2, 3	70		
84	Duplicate Start	The firmware found an error while reading the sensor EEPROM. Send the RESET_CMD (17) again. If the same error code is received again, call CyberOptics.	1, 2, 3	70		
85	No EEPROM Start of Data	The firmware found an error while reading the sensor EEPROM. Send the RESET_CMD (17) again. If the same error code is received again, call CyberOptics.	1, 2, 3	70		
86	Bad Parameter	An invalid parameter was specified for a command. Verify that the parameter was entered correctly.	1, 2, 3	70		
87	Bad Data	Invalid data was detected. Verify that the programming sequence contains valid information and that no command failed during the programming sequence.	1, 2, 3	70		

Table 9-1: Laser Align Status Codes (continued)

Status code (decimal)	Name	Description/Action to Take	Valid for error level(s):	Maps Backward to Status Code X at Error Level 0	1	2
90	Too Few Frames	This is a firmware failure. Check component orientation. If the problem persists, contact CyberOptics for assistance.	1, 2, 3	70		
91	Minimum Out of Range 1	This is a firmware failure. Check component orientation. If the problem persists, contact CyberOptics for assistance.	1, 2, 3	70		
92	Minimum Out of Range 2	This is a firmware failure. Check component orientation. If the problem persists, contact CyberOptics for assistance.	1, 2, 3	70		
93	Minimum Too High	The width versus angle curve is too flat. This error may occur on small parts when rotating at very low speeds. Try checking component height and/or increasing speed.	1, 2, 3	70		
94	No Cutoff 1	This is a firmware failure. Contact CyberOptics for assistance.	1, 2, 3	70		
95	No Cutoff 2	This is a firmware failure. Contact CyberOptics for assistance.	1, 2, 3	70		

Table 9-1: Laser Align Status Codes (continued)

Status code (decimal)	Name	Description/Action to Take	Valid for error level(s):	Maps Backward to Status Code X at Error Level		
				0	1	2
96	Constant Angle	The SCM did not register any changes in the encoder angle for the last scan. Check the encoder connection to the SCM. Make sure that the encoder rotates when the part is spinning. Verify that the part is spinning.	1, 2, 3	70		
97	End Min	The minimum shadow width occurred at the end of the scan. Increase rotation angle.	2, 3	67	67	
98	Angle Limit Exceeded	The angle exceeded specified angle limit before a minimum was detected. Check prerotation, angle limit, and/or component height.	2, 3	67	67	
99	Overrun	Early processing (from a REPORTW_CMD) was not completed before search for next minimum was to start (as specified by holdoff angle). Reduce rotation speed; increase holdoff angle; ask for angle only or get results after rotation.	2, 3	65	65	

Table 9-1: Laser Align Status Codes (continued)

Status code (decimal)	Name	Description/Action to Take	Valid for error level(s):	Maps Backward to Status Code X at Error Level 0	1	2
100	Units Not Set	Required initialization was not done prior to requesting information that needed initialization. Check to make sure that you are correctly specifying parameters. ◆ REPORTW_CMD requires units to be set (1/10 pixels or microns). ◆ READ_CMD requesting IMAGE_CHK limit violations requires IMAGE_CHK limits to be set up.	2, 3	70	86	
101	Page Mismatch	Firmware error. Call CyberOptics for assistance.	2, 3	70	70	
102	Invalid Algorithm	The measurement requested from Laser Align indicated that an invalid algorithm was to be used. ➲ Refer to *Measurement Algorithms* in Chapter 8 for more information.	2, 3	7	7	
103	Low Background	The laser intensity was too low during image check. Therefore, image check processing was not performed (values are not calculated). The sensor failed. Call CyberOptics for assistance.	2, 3	70	87	

Table 9-1: Laser Align Status Codes (continued)

Status code (decimal)	Name	Description/Action to Take	Valid for error level(s):	Maps Backward to Status Code X at Error Level 0	1	2
104	Bad Image Check Region	Image check region boundaries were not set properly. Try resetting boundaries using LOAD_CMD (40) with image quality indexes.	2, 3	70	87	
105	Image Check Fail	Image check failed a parameter comparison or image check values failed a threshold comparison. Check sensor for foreign objects and re-run.	2, 3	70	87	

Chapter 10

Compatibility Commands and Procedures

Purpose of Compatibility Commands and Procedures

The commands in this chapter are provided for Laser Align users who have written programs for previous versions of the Laser Align firmware, and want to continue to use those programs with the current Laser Align firmware version.

Although the commands in this chapter will still work as documented, the command set documented in Chapter 7 provides greater functionality and flexibility, and should be used whenever you create new programs or update old ones.

Summary of Compatibility Commands

Table 10-1 summarizes the Laser Align commands available to maintain backward compatibility with previous versions of the Laser Align firmware, and the commands that replace them. Command type (Cmd Type) is indicated by a single letter:

- D = Diagnostic
- I = Initialization
- R = Run
- S = Set-up

Details about the compatibility commands follow the table, and are organized alphabetically.

➲ See Chapter 7 for more information about command types and the replacement commands.

Table 10-1: Compatibility Command Summary

Command decimal/name		Purpose	Cmd Type	# of Add'l Bytes	Replaced By	As of Version	Response ACK	See Page
19	RUN_CMD*	Scan the alignment of a single part.	R	1	SWEEP_CMD (30)	1.0	RUN_ACK (131)	10-17
20	REPORT_CMD	Request the RESULT data for the first alignment angle.	R	None	REPORTW_CMD (42)	2.0	13-byte data string	10-10
25	TEST_CMD	Perform a self-test on the Laser Align SCM.	D	None	IMG_CHK_CMD (43)	2.5	TEST_ACK (137)	10-19
27	PARAM_CMD	Write next byte of PARAM data for the current window.	S	1	LOAD_CMD (40)	2.0	PARAM_ACK (149)	10-7
28	R_PARAM_CMD	Read next byte of PARAM data for the current window.	S	1	READ_CMD (41)	2.0	One byte of the 13-byte data string identifying measurement parameter structure.	10-14

* Only partially replaced. Still in use for a different function.

Table 10-1: Compatibility Command Summary (continued)

Command decimal/name		Purpose	Cmd Type	# of Add'l Bytes	Replaced By	As of Version	Response ACK	See Page
29	REPORTS_CMD	Request the RESULT data for the second alignment angle.	R	None	REPORTW_CMD (42)	2.0	13-byte data string	10-12
33	POINTER_CMD	Prepare the sensor to receive a multiple byte value.	S	None	LOAD_CMD (40)	2.8	POINTER_ACK (145)	10-9
34	VALUE_CMD	Send the sensor one byte of a multiple byte value.	S	1	LOAD_CMD (40)	2.8	VALUE_ACK (146)	10-21
39	ERR_LVL_CMD	Set the system error level (0 is the default).	S	1	LOAD_CMD (40)	2.8	ERR_LVL_ACK (149)	10-5

Command Details

The compatibility commands are organized alphabetically. The command name is followed by its decimal value in parentheses.

ERR_LVL_CMD (39)

Command Type: Set-up

ERR_LVL_CMD (39) is used to set the level of error codes Laser Align will use in reporting status information. It must be followed by a parameter specifying the desired error level.

The default error level is 0. After power-up or after a RESET_CMD is sent, error level 0 is automatically selected. If you want an error level other than the default, you must set it.

➲ See *Compatibility Procedures* later in this chapter for more information about setting the error level using ERR_LVL_CMD (39).

If you check status following an ERR_LVL_CMD by using the REPORTW_CMD (42), the ERR_LVL_CMD sets the status in **window 0 result 0** to NO_ERROR (1) if the error level is known by Laser Align firmware, or to BAD_PARAM (86) if the error level is not known by the firmware. Any previous error is erased.

Although you can still use ERR_LVL_CMD (39) to set the error level, firmware versions 2.8 and newer use **LOAD_CMD (40)** to set the error level. It doesn't erase previous errors, and can be verified with READ_CMD (41).

➲ See *LOAD_CMD (40)* in Chapter 7 and *Defining General System Parameters* in Chapter 5 for more information.

Table 10-2: ERR_LVL_CMD Format

Byte	Description	Valid Value(s)
0	Command byte	39 = ERR_LVL_CMD
1	Parameter (error level)	0 = Error level for all firmware versions 1 = Error level for firmware versions 1.0 and higher 2 = Error level for firmware versions 2.0 and higher 3 = Error level for firmware versions 2.7 and higher

Response

ERR_LVL_ACK (149)

Additional Information

The error level you set determines how Laser Align reports errors.

All new development should use the highest error level available, which provides the most informative error messages. The lower levels should only be used for backward compatibility.

When the lower error levels are used with higher level firmware, status codes are not available for all possible errors, so some errors will be mapped to the status codes that are available for that error level.

➲ See Chapter 9 for more information on status codes.

PARAM_CMD (27)

Command Type: Set-up

PARAM_CMD (27) writes parameter data that is used for the measurements executed during SWEEP_CMD (30). There are 14 bytes in the measurement parameter data structure (Table 10-4).

The parameters must be programmed sequentially. This means that you send the PARAM_CMD followed by the appropriate data for the first byte of the structure, then send the PARAM_CMD followed by data for the second byte of the structure, and so on, until you have sent data for all 14 bytes of the parameter structure. You can skip over a byte that does not need to be modified by sending R_PARAM_CMD, which simply reads the existing/default data for that parameter.

See *R_PARAM_CMD (28)* in this chapter for more information about its use.

Although PARAM_CMD (27) is still available, firmware versions 2.0 and newer use **LOAD_CMD (40)** instead of PARAM_CMD (27).

See *LOAD_CMD (40)* in Chapter 7 for more information.

Table 10-3: PARAM_CMD Format

Byte	Description	Value(s)
0	Command byte	27 = PARAM_CMD
1	Measurement parameter value	Value to be entered in current byte of measurement parameter structure (Table 10-4). The values must be entered in sequence, one byte at a time, beginning with byte 0.

Table 10-4: Measurement Parameter Structure

Byte Number	Description
0	LSB of left edge
1	MSB of left edge
2	LSB of right edge
3	MSB of right edge
4	LSB of encoder
5	MSB of encoder
6 - 9	Bytes 0 through 3 of holdoff angle (byte 0 = LSB)
10	LSB of algorithm 0
11	MSB of algorithm 0
12	LSB of algorithm 1
13	MSB of algorithm 1

Response

PARAM_ACK (149)

Additional Information

The *PARAM pointer* keeps track of the parameter to which information is currently being written. After you send PARAM_CMD and the parameter data byte, the PARAM pointer moves to the next byte of parameter information when Laser Align responds with a PARAM_ACK (139).

A valid PARAM_CMD must be preceded by a WINDOW_CMD (26), another PARAM_CMD, or an R_PARAM_CMD (28). The WINDOW_CMD sets the PARAM pointer to the first byte, so when you send PARAM_CMD you begin writing parameter data for the first parameter. If any other command is sent, the PARAM pointer is reset to the first byte.

➲ See *WINDOW_CMD (26)* in Chapter 7 for more information.

POINTER_CMD (33)

Command Type: Set-up

POINTER_CMD (33) prepares the SCM to receive a multiple byte value. This command must be followed by VALUE_CMD (34).

See *Compatibility Procedures* later in this chapter for more information about the use of this command.

Although you can still use POINTER_CMD (33) and VALUE_CMD (34) to load the VALUE buffer, firmware versions 2.8 and newer allow the use of LOAD_CMD (40) for this purpose. This works better because it allows you to load all four bytes of the buffer at once.

See *Synchronizing with the Encoder* in Chapter 5 for more information.

Table 10-5: POINTER_CMD Format

Byte	Description	Valid Value(s)
0	Command byte	33 = POINTER_CMD

Response

POINTER_ACK (145)

REPORT_CMD (20)

Command Type: Run

REPORT_CMD (20) requests the RESULT data for the first alignment angle. This RESULT data includes the angle, center and width values of the orientation measurement and the measurement status code.

Although REPORT_CMD (20) is still available, firmware versions 2.0 and higher use **REPORTW_CMD (42)** instead.

Table 10-6: REPORT_CMD Format

Byte	Description	Value(s)
0	Command byte	20 = REPORT_CMD

Response

REPORT_CMD reports the measurement data 13-byte structure (Table 10-7). The results reported will depend upon the measurement parameters set up with LOAD_CMD (40).

If you issued a SWEEP_CMD (30), you should not use REPORT_CMD (20) to request data for angle, center, and width until the SWEEP_CMD has completed, as this command will abort the sweep.

➲ See *LOAD_CMD (40)* in Chapter 7 for more information.

Table 10-7: REPORT_CMD RESULT Data

Byte Number	Description
0	Status ➲ See Chapter 9 for more information about status codes.
1 - 4	Alignment angle
5 - 8	Center position for measurement algorithms 11 and 14 Edge position for measurement algorithms 12 and 13 Zero for parameter index 10 sent with RUN_CMD (19)
9 - 12	Width Zero for parameter index 10 sent with RUN_CMD (19)

➲ See Measurement Algorithms in Chapter 8 for more information about the algorithm.

Additional Information

With REPORT_CMD, values are reported as follows:

- The status code is an unsigned 8-bit value.

  ➲ Refer to Chapter 9 for a list of the status codes and their meanings.

- The calculated values of the angle, center (or edge), and width measurements are in two's complement 32-bit form with the least-significant byte in the lowest address.
- The angle (or θ) value is reported in absolute encoder counts. It has 24 bits of significance, and is sign-extended to 32 bits.
- The center and width values are reported in one-tenth pixels or microns, depending on the units set using LOAD_CMD (40).

REPORTS_CMD (29)

Command Type: Run

REPORTS_CMD (29) requests the RESULT data for the second alignment angle. This RESULT data includes the angle, center and width values of the orientation measurement and the measurement status code.

Although REPORTS_CMD (42) is still available, firmware versions 2.0 and newer use **REPORTW_CMD (42)** for greater efficiency in obtaining result information.

Table 10-8: REPORTS_CMD Format

Byte	Description	Value(s)
0	Command byte	29 = REPORTS_CMD

Response

REPORTS_CMD reports the measurement data 13-byte structure (Table 10-9). The results reported will depend upon the measurement parameters set up with LOAD_CMD (40).

- See *LOAD_CMD (40)* in Chapter 7 and Scanning Parameter Indexes in Chapter 8 for more information.

Table 10-9: REPORTS_CMD Result

Byte Number	Description
0	Status ➲ See Chapter 9 for more information about status codes.
1 - 4	Align angle
5 - 8	Center position for measurement algorithms 11 and 14 Edge position for measurement algorithms 12 and 13 Zero for parameter index 10 sent with RUN_CMD (19)
9 - 12	Width Zero for parameter index 10 sent with RUN_CMD (19)

➲ See Measurement Algorithms in Chapter 8 for more information about the algorithm.

Additional Information

Values are reported in response to the REPORTS_CMD (29) as follows:

- The status code is an unsigned 8-bit value.

  ➲ Refer to Chapter 9 for a list of the status codes and their meanings.

- The calculated values of the angle, center (or edge), and width measurements are in two's complement 32-bit form with the least-significant byte in the lowest address.
- The angle (or θ) value is reported in absolute encoder counts. It has 24 bits of significance, and is sign-extended to 32 bits.
- The center and width values are reported in one-tenth pixels or microns, depending on the units set using LOAD_CMD (40).

If you issued a SWEEP_CMD (30), you should not use REPORTS_CMD (29) to request data for angle, center, and width until the SWEEP_CMD has completed, as it will abort the sweep.

R_PARAM_CMD (28)

Command Type: Set-up

R_PARAM_CMD (28) reads the parameter data set using PARAM_CMD (27). You must read the measurement parameters sequentially by sending an R_PARAM_CMD for each of the 14 bytes of measurement information.

R_PARAM_CMD can also be used to keep a measurement parameter set to its current value, when used in conjunction with the PARAM_CMD (27).

See *PARAM_CMD (27)* in this chapter for more information.

Although R_PARAM_CMD (28) is still available, firmware versions 2.0 and newer use **READ_CMD (41)** for greater efficiency in obtaining parameter data.

See *READ_CMD (41)* in Chapter 7 for more information.

Table 10-10: R_PARAM_CMD Format

Byte	Description	Value(s)
0	Command byte	28 = R_PARAM_CMD

Response

The system returns the information on each byte of the measurement parameter structure sequentially, in response to each R_PARAM_CMD. The first byte is returned first, the second byte is returned next, and so on (Table 10-11).

Table 10-11: Measurement Parameter Structure

Byte Number	Description
0	LSB of left edge
1	MSB of left edge
2	LSB of right edge
3	MSB of right edge
4	LSB of encoder
5	MSB of encoder
6 - 9	Bytes 0 through 3 of holdoff angle (byte 0 = LSB)
10	LSB of algorithm 0
11	MSB of algorithm 0
12	LSB of algorithm 1
13	MSB of algorithm 1

Additional Information

The *PARAM pointer* keeps track of the parameter for which information is currently being written. After you send R_PARAM_CMD (28) and receive the parameter data byte, the PARAM pointer moves to the next byte of parameter information.

A valid R_PARAM_CMD must be preceded by the WINDOW_CMD (26) command, a PARAM_CMD (27), or another R_PARAM_CMD.

If you begin with the WINDOW_CMD, the PARM pointer is automatically reset to the first byte, so when you send R_PARAM_CMD you begin reading parameter data for the first parameter.

Note Any command other than PARAM_CMD (27) or R_PARAM_CMD (28) resets the PARAM pointer to the first byte.

RUN_CMD (19) - with parameters 11, 12, 13, 14

Command Type: Run

RUN_CMD (19) initiates a measurement of one side of a rotating component using a specified firmware algorithm to determine the minimum width of the component. These algorithms are identified by a parameter that must be sent with RUN_CMD (19).

Firmware versions 1.0 and newer use **SWEEP_CMD (30)** to take measurements. RUN_CMD (19) is still used with parameter 10 to perform external reset of the SCM encoder counter.

- See *SWEEP_CMD (30)* in Chapter 7 for more information about its use in performing measurements.
- See *RUN_CMD (19)* in Chapter 7 for more information about use of this command to perform external reset of the SCM encoder counter.

Table 10-12: RUN_CMD Format

Byte	Description	Value(s)
0	Command byte	19 = RUN_CMD
1	Parameter	10 = REZERO_ALG External Reset of Position Counter **11 = WIDTH_ALG Measure Using Single Frame** **12 = LEFT_ALG Find Left Edge** **13 = RIGHT_ALG Find Right Edge** **14 = WIDE_ALG Measure Using Average of 3 Frames**

Response

RUN_ACK (131)

Additional Information

The RUN command:

- Resets the RESULT data.
- Selects **window 0**.
- Temporarily sets window 0 to range from the sensor's default LEFT_EDGE to the default RIGHT_EDGE.
- Temporarily disables window 1.
- Selects the encoder for window 0.

If you issue REPORTW_CMD (42) while Laser Align is still measuring the rotating component, RUN_CMD will be aborted.

TEST_CMD (25)

Command Type: Diagnostic

TEST_CMD (25) initiates a self-test on the Laser Align sensor and SCM.

Note There should not be a component in the field of view of the sensor during the TEST_CMD.

Although TEST_CMD (25) is still available, firmware versions 2.5 and newer use **IMG_CHK_CMD (43)** to get more detailed information about the quality of the image provided by the Laser Align system.

Table 10-13: TEST_CMD Format

Byte	Description	Valid Value(s)
0	Command byte	25 = TEST_CMD

Response

TEST_ACK (137)

A REPORTW_CMD (42) issued after the TEST_CMD provides the test result in the status byte with the bit-mapping shown in Table 10-14. A **0** signifies that the Laser Align system passed the associated test; a **1** indicates test failure.

Table 10-14: RESULT Status Byte for TEST_CMD

Bit Number	Test Result
0	Position counter
1	Threshold register
2	RAM
3	Sensor interface
4	Video quality
5	Window cleanliness
6	FIFO
7	FIFO reset

VALUE_CMD (34)

Command Type: Run

VALUE_CMD (34) places a byte of data into the VALUE buffer (4 bytes). When this command is used, the specified value is placed in the buffer at the current byte location, and the pointer is advanced by one. The least significant byte is transmitted first.

VALUE_CMD can be used in conjunction with POINTER_CMD (33) and PRESET_CMD (35) to preset the encoder to a specified number. It must be preceded by POINTER_CMD (33).

Although VALUE_CMD (34) is still available, firmware versions 2.8 and newer use **LOAD_CMD (40)** to place data into the VALUE buffer. LOAD_CMD (40) makes it possible to load all four bytes with just one command.

See *Compatibility Procedures* later in this chapter for more information about using VALUE_CMD to reset the encoder.

Table 10-15: VALUE_CMD Format

Byte	Description	Valid Value(s)
0	Command byte	34 = VALUE_CMD
1	VALUE parameter	Value to be entered into the current byte location of the VALUE buffer -- byte 0, 1, 2, or 3. (The values must be entered in sequence, one byte at a time, beginning with byte 0).

Response

VALUE_ACK (146)

Additional Information

The VALUE buffer is a four-byte buffer established for purposes of providing encoder counter values. It is loaded using either POINTER_CMD (33) and VALUE_CMD (34), or using LOAD_CMD (40). The buffered value is transmitted when the PRESET_CMD (35) is sent.

Compatibility Procedures

This section provides procedures that use the compatibility commands. This information is provided to support backward compatibility.

Included procedures:

- Setting the error level using the ERR_LVL_CMD (39) for firmware versions prior to 2.8.
- Setting the encoder counter to a specified number using the POINTER_CMD (33) and the VALUE_CMD (34) for firmware versions prior to 2.8.

Setting Error Level

▼▼ To set error level using ERR_LVL_CMD (39)

1 Send **ERR_LVL_CMD** sequence:

- For level **0** send**: 39 0**
- For level **1** send: **39 1**
- For level **2** send: **39 2**
- For level **3** send: **39 3**

2 Check Laser Align response:

- If response is **149 (ERR_LVL_ACK)**, error level is set.
- If response is anything other than **149 (ERR_LVL_ACK)**, repeat **Step 1**, making sure to enter parameters correctly.

Note The system sets itself to error level **0** any time you power down and then power back up, or if you use **RESET_CMD (17)** to reset Laser Align. You must use **ERR_LVL_CMD (39)** to set an error level other than **0**.

See *Defining General System Parameters* in Chapter 5 for more information about setting error level and about which firmware versions use which error level.

10

Setting Encoder Counter

▼▼ To set the Laser Align encoder counter to a specified value using POINTER_CMD (33) and VALUE_CMD (34)

1 Send **POINTER_CMD (33)** sequence:

33

2 Check Laser Align response:

- If response is **145 (POINTER_ACK)**, the command was successfully received.
- If response is anything other than **145 (POINTER_ACK)**, repeat **Step 1**, making sure to enter the command correctly.

3 Send **VALUE_CMD (34)** sequence:

34 ***n***

where ***n*** = the value for Byte 0 of the desired encoder position (this is a 4-byte value and Byte 0 is the LSB).

4 Check Laser Align response:

- If response is **146 (VALUE_ACK)**, the command was successfully received.
- If response is anything other than **146 (VALUE_ACK)**, repeat **Step 3**, making sure to enter the command sequence correctly.

5 Repeat **Steps 3** and **4** three more times to send the values for each of the remaining bytes that define encoder position (Bytes 1, 2, and 3).

6 Send **PRESET_CMD (35)** sequence:

35 ***n***

where ***n*** = the encoder number (0 or 1).

7 Check Laser Align response:

- If response is **147 (PRESET_ACK)**, the command was successfully received.
- If response is anything other than **147 (PRESET_ACK)**, repeat **Step 6**, making sure to enter the command sequence correctly.

8 To set the angle for a second encoder (SCM1 only), repeat **Steps 1** through **7**.

See *Synchronizing with the Encoder* in Chapter 5 for more information.

Chapter 11

Using Laser Align Software

Introducing Laser Align Software

There are three programs included with your Laser Align system:

- **COM_422:** An RS422 communications program that facilitates communications between a DOS-based PC and the SCM.
- **COMTEST**: A program that allows you to send commands from a DOS-based PC to Laser Align one byte at a time, and to view Laser Align responses.
- **LA_View:** A program that allows you to verify that the Laser Align sensor is functioning properly. It also allows you to view, save, and retrieve a visual representation of the laser image.

This chapter presents information on the features and uses of these programs.

➲ See Installing Laser Align Software in Chapter 4 for information on installation of these programs.

COM_422 Software

The host system communicates with the Laser Align using an RS422 serial communication link running at a baud rate determined by user controlled and/or fixed settings specific to the SCM firmware, with 8 data bits, 2 stop bits, and no parity. CyberOptics provides a communications program called **COM_422.EXE** to help with these communications during initial system testing. This is a DOS-based program used to communicate over an RS422 serial link via a CIO-COM-422 card (or equivalent).

COM_422.EXE is a Terminate and Stay Resident (TSR) program that is run on the host system before communication with the Laser Align system is attempted. COM_422 installs or removes a serial communication buffer for receiving characters over the Laser Align's RS422 link.

Note COM_422 is required for COMTEST and LA_View operation.

➲ See Verifying Laser Align Functionality in Chapter 4 for more information on using COM_422 to verify communications.

Program Syntax and Function

COM_422 program syntax is:

COM_422 [-annn] [-in] [-o or **-u] [-q] [-vnn]**

Parameter	Description
-a*nnn*	Specifies the hexadecimal base I/O address for the RS422 card to the value of *nnn*. (Default: 3E8 hex)
-i*n*	Specifies the hardware interrupt number (IRQ) for the RS422 card to the value *n*. (Default: 4)
-o or -u	Removes the driver at the specified vector. (The -v*nn* switch must be used with this switch if removing from non-default software interrupt vector.)

11

Parameter	**Description**
-q	Inhibits messages to the standard output device.
-v*nn*	Specifies the hexadecimal software interrupt vector to use for the driver. (Default: 60 hex)

The following example loads the COM_422 driver at software interrupt vector 61, and specifies the RS422 card as using I/O address 3F8 hexadecimal and interrupt number (IRQ) 5.

C> COM_422 -a3F8 -i5 -v61

Once loaded, the COM_422 driver initializes the host RS422 port and provides a set of 3 functions that may be called at the 80x86 class machine level (Table 11-1).

COM_422 Programmer Reference

Verifying COM_422 Load

Before using the driver in your application, you should verify that the COM_422 driver has been loaded. If the driver has not been installed and an INT <software interrupt vector> instruction is executed, you will get unpredictable (possibly catastrophic) results.

To verify that the driver has been loaded, get the interrupt vector to the COM_422 driver, and check 5 bytes from the entry point of the software interrupt for the string **'COM_INT',0**. To get the interrupt vector read memory location **0000: (interrupt vector number * 4)**.

Example

Suppose you've loaded the COM_422 driver and are using interrupt vector 60_{hex}. First, calculate the offset to find the address at the entry point of the driver:

$$60_{hex} * 4 = 0180_{hex}$$

The entry point of the driver is located at address **0000:0180**. If location **0000:0180** contained the sequence **28 F3 C3 13** then the location of the COM_422 driver is address **13C3:F328** and the string **'COM_INT',0** should be at location **13C3:F32D**. If it is not, something went wrong loading COM_422, and the loading procedure should be repeated.

Using COM_422 Functions

Table 11-1: COM_422 Functions

Register AX	Function
0	Serial Status Returns Status in DX register. (Bit 1 = Receive Data Ready).
1	Receive Data Returns Data in DL register. (DX register set to FFFF hex if no data is available).
2	Transmit Data Transmit data in DL register to SCM.

To execute a COM_422 function, you load the AX register to the proper function code, load the DL register with the proper data if transmitting data, and execute an INT <software interrupt vector> instruction.

Example

If executed, the following 80x86 assembly language example (entered using a text editor) would transmit HELLO_CMD (16) to the SCM and wait for a HELLO_ACK (128) response or timeout.

```
        mov     AX,2h               ; prepare for transmit function
        mov     DL,10h              ; value to transmit
        int     60h                 ; COM_422 -vnn software interrupt vector.

        mov     BX, 20h             ; retries
@NoData:
        dec     BX                  ; decrement number of retries
        jz      @Error              ; Should have got HELLO_ACK back from the SCM
        call    Delay               ; Delay some amount of time to allow SCM to
                                    ; respond
        mov     AX, 1h              ; prepare for receive function
        int     60h                 ; COM_422 -vnn software interrupt vector.
        cmp     DX, FFFFh           ; if no data is available...
        je      @NoData             ; (DX = FFFF) then jump to @Error
        cmp     DL,80h              ; else if HELLO_ACK
        je      @GotAcknowledge     ; Got HELLO_ACK command success!
@Error:
        .
        .
```

Example

The following C code shows how to use the Int86() function and the REGS structure (include DOS.H) to access the INT command and the PC registers.

```
#include <string.h>
#include <dos.h>
static union REGS reg;

unsigned com_status (unsigned com_int)
{
        /* a return value of 1 means data available*/
        char identity[8];
        void (__interrupt__far *int_vector)();
        int_vector = _dos_getvect(com_int);
        (void)_fstrncpy(identity, (char _far *)int_vector+5, sizeof(identity));
        if (int_vector == 0L || strcmp(identity, "COM_INT"))
                return(0);
        else
        {
                reg.x.ax = 0;
                (void) int86((int)com_int, &reg, &reg);
                return(reg.x.dx);
        }
}

int com_receive()
{
        /* a return value of -1 means no data returned */
        reg.x.ax = 1;
        (void) int86((int)com_int, &reg, &reg);
        return((int)reg.x.dx);
}

void com_transmit (char ch)
{
        reg.x.ax = 2;
        reg.x.dx = (unsigned) ch;
        (void) int86(0x60, &reg, &reg);
        return;
}
```

11

Coding Exercise

This exercise provides you with an opportunity to test COM_422 usage in a situation with predictable outcome.

1. From the DOS command shell, type **c:** and press J to switch to the c: drive root directory.
2. Change to your **lasalign** directory. Type:

   cd c:\lasalign

   then press J.
3. Copy the COM_422.EXE driver to that directory, if it isn't already installed. Type:

   copy a:\com_422.exe

   then press J.
4. Launch COM_422. Type:

   com_422 [-annn] [-in] [-o or **-u] [-q] [-vnn]**

   where:

Parameter	Description
-a*nnn*	Specifies the hexadecimal base I/O address for the RS422 card to the value of *nnn*. (Default: 3E8 hex)
-i*n*	Specifies the hardware interrupt number (IRQ) for the RS422 card to the value *n*. (Default: 4)
-o or -u	Removes the driver at the specified vector. (The -v*nn* switch must be used with this switch if removing from non-default software interrupt vector.)
-q	Inhibits messages to the standard output device.
-v*nn*	Specifies the hexadecimal software interrupt vector to use for the driver. (Default: 60 hex)

5. Press J.

   This starts COM_422.

6 To verify that COM_422 is properly loaded, start the DOS debugger. Type:

DEBUG

then press J.

7 Dump memory location containing interrupt vector 60_{hex} (assuming you chose the default interrupt).

```
-d0000:0180

0000:0180  4F 0A FA 1E 00 00 00 00-00 00 00 00 00 00 00 00   O...............
0000:0190  37 00 4C E7 00 00 00 00-00 00 00 B0 02 B9 02   7.L.............
0000:01A0  BE BE 00 F0 62 07 23 1E-BE BE 00 F0 BE BE 00 F0   ....b.#.........
0000:01B0  BE BE 00 F0 30 55 00 C0-BE BE 00 F0 BE BE 00 F0   ....0U..........
0000:01C0  52 00 B0 07 59 BD 00 F0-BE BE 00 F0 18 02 1C 13   R...Y...........
0000:01D0  B5 35 B7 DA 4A BD 00 F0-17 01 B0 07 2F 01 B0 07   .5..J......./...
0000:01E0  00 00 00 00 00 00 00 00-26 00 4C E7 00 00 00 00   ........&.L.....
0000:01F0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

8 Dump memory location of the interrupt entry point and verify that the string “COM_INT” is found.

“COM_INT” String

```
-d1EFA:0A4F

1EFA:0A40                                                55              U
1EFA:0A50  8B EC EB 11 43 4F 4D 5F-49 4E 54 00 00 5A 21 00   ....COM_INT..Z!.
1EFA:0A60  00 00 00 60 00 FB 1E 50-2E A1 5D 0A 8E D8 58 89   ...`...P..]...X.
1EFA:0A70  36 4E 04 89 3E 50 04 89-16 4C 04 89 0E 4A 04 89   6N..>P...L...J..
1EFA:0A80  1E 48 04 A3 46 04 8F 06-54 04 FF 36 54 04 E8 2E   .H..F...T..6T...
1EFA:0A90  F9 8B 36 4E 04 8B 3E 50-04 8B 16 4C 04 8B 0E 4A   ..6N..>P...L...J
1EFA:0AA0  04 8B 1E 48 04 A1 46 04-1E 07 1F 5D CF 5D C3 55   ...H..F....].].U
1EFA:0AB0  8B EC B4 35 8A 46 04 CD-21 8C C0 0B C0 74 14 8B   ...5.F..!....t..
1EFA:0AC0  FB 83 C7 05 8B 76 06 B8-02 00 FC B9 08 00 F3      .....v.........
```

9 Enter a program to test each of COM_422 drivers commands. Make sure that the interrupt values set match the interrupt vector set when COM_422 was called.

```
-a100

286A:0100 mov ax,0

286A:0103 int 60

286A:0105 mov ax,2

286A:0108 mov dl,10

286A:010A int 60

286A:010C mov ax,1

286A:010F int 60

286A:0111 int 20

286A:0113
```

10 Step through the program.

The first interrupt (INT 60) tests the status command (status returned in the DL register). The second interrupt sends the HELLO_CMD (in the DL register). The third interrupt receives the HELLO_ACK (in the DL register). The final interrupt (INT 20) terminates the program.

```
-p

AX=0000  BX=0000  CX=0000  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=286A  ES=286A  SS=286A  CS=286A  IP=0103   NV UP EI PL NZ NA PO NC
286A:0103 CD60          INT       60
-p
```

Status in DL register

```
AX=0000  BX=0000  CX=0000  DX=0060  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=286A  ES=215A  SS=286A  CS=286A  IP=0105   NV UP EI PL NZ NA PO NC
286A:0105 B80200        MOV       AX,0002
-p

AX=0002  BX=0000  CX=0000  DX=0060  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=286A  ES=215A  SS=286A  CS=286A  IP=0108   NV UP EI PL NZ NA PO NC
286A:0108 B210          MOV       DL,10
-p

AX=0002  BX=0000  CX=0000  DX=0010  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=286A  ES=215A  SS=286A  CS=286A  IP=010A   NV UP EI PL NZ NA PO NC
286A:010A CD60          INT       60
-p

AX=0002  BX=0000  CX=0000  DX=0010  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=286A  ES=215A  SS=286A  CS=286A  IP=010C   NV UP EI PL NZ NA PO NC
286A:010C B80100        MOV       AX,0001
-p

AX=0001  BX=0000  CX=0000  DX=0010  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=286A  ES=215A  SS=286A  CS=286A  IP=010F   NV UP EI PL NZ NA PO NC
286A:010F CD60          INT       60
-p
```

Final Interrupt

```
AX=0001  BX=0000  CX=0000  DX=0080  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=286A  ES=215A  SS=286A  CS=286A  IP=0111   NV UP EI PL NZ NA PO NC
286A:0111 CD20          INT       20
-p

Program terminated normally
-q
```

COMTEST Software

The COMTEST program allows you to send Laser Align commands from a PC-compatible, DOS-based computer to your Laser Align system one byte at a time; and to receive Laser Align responses one byte at a time.

COMTEST Command Keys

Table 11-2: Command Keys available in COMTEST

Key	Action
E	Exits the COMTEST software. Also used to exit the Function mode.
J	Sends a command to Laser Align when pressed after a decimal entry in the Command mode. Also used without a decimal entry to exit the Command mode.
C	Accesses the Command mode, which provides a SEND prompt where a Laser Align command can be typed and relayed to the Laser Align system.
F	Accesses the Function mode, which contains a secondary set of functions that were used as alternatives to some early Laser Align commands. **Although you can still select the secondary functions, they are now obsolete, so you do not need to select this option.**
H	Displays a help screen with descriptions of the various COMTEST functions.

Using COMTEST

▼▼ To launch COMTEST

1. Make sure that COMTEST and COM_422 are installed on your PC.

   See Installing Laser Align Software in Chapter 4 for information on installation.

2. Go to the DOS prompt and switch to the **c:** drive root directory.
3. Type **cd c:\lasalign** to verify that you are in the correct directory.
4. Type **com_422 [-annn] [-in] [-o** or **-u] [-q] [-vnn]** where:

Parameter	Description
-a*nnn*	Specifies the hexadecimal base I/O address for the RS422 card to the value of *nnn*. (Default: 3E8 hex)
-i*n*	Specifies the hardware interrupt number (IRQ) for the RS422 card to the value *n*. (Default: 4)
-o or -u	Removes the driver at the specified vector. (The -v*nn* switch must be used with this switch if removing from non-default software interrupt vector.)
-q	Inhibits messages to the standard output device.
-v*nn*	Specifies the hexadecimal software interrupt vector to use for the driver. (Default: 60 hex)

5. Press J.

   This executes the COM_422 program which allows communication through the RS422 interface.

6. Type **comtest** and press J.

   The system will display the COMTEST prompts. You are now ready to begin using COMTEST.

▼▼ To use COMTEST

1. Launch COMTEST using the preceding procedure.
2. Type **C** to switch to Command mode.
3. Type the decimal value for the command you want to send to Laser Align (e.g., 16 for HELLO_CMD).
4. Press J.
5. Wait for the Laser Align response.

   Important: If you are entering a command that requires a string of parameters, you must enter the entire command string before Laser Align will return a response.

   For example, if you send a LOAD_CMD (40), which has an 8-byte format, you will need type and enter each of the parameters before Laser Align will give you a response.

   If you get locked up when entering a command string (Laser Align does not return a response when you expect one), you can enter a series of HELLO_CMD (16) values until you receive a HELLO_ACK (128) response.

   See Chapter 7 for information on the commands and their formats.

6. When you are finished working with COMTEST, press E to exit.

LA_View Software

LA_View software provides a visual representation of the laser image that you can view, save, and retrieve. It also allows you to verify that your Laser Align sensors are functioning properly. The software is run on a PC-compatible, DOS-based computer.

LA_View has two display modes:

- **Normal** - provides information about the status reported by the sensor, encoder counts, and center and width information. This is the default display mode.
- **Image Check** - provides information on image quality, including dark level, minimum and maximum pixel intensity values, background, and frequency.

You can toggle between these modes while the software is running. You can also change the default mode, if desired.

LA_View Command Keys

Table 11-3: Command Keys available in LA_View

Key	Action
E	Exits the LA_View software.
r	Pans right on the laser stripe image display. You can pan on the display only when the image is magnified.
l	Pans left on the laser stripe image display. You can pan on the display only when the image is magnified.
t	Zooms out from the laser stripe display. Each time you press the t key, the display magnification decreases by one half until the display is at normal magnification.
b	Zooms in on the laser stripe display. Each time you press the b key, the display magnification increases by a factor of two.

Table 11-3: Command Keys available in LA_View (continued)

Key	Action
A	Allows you to change measurement algorithms. ➲ See Measurement Algorithms in Chapter 8 for more information.
F	Toggles the display of the median filter line through the laser stripe image on and off. When the median filter line is displayed, the laser stripe display takes slightly longer to update. By default, the filter line is off.
I	Toggles between the normal display mode with angle, center (or edge), and width values and the image check display mode with the image quality evaluation.
P	Toggles the pixel mode on and off. Normally, the image from only a sample of the detector pixels is displayed. Pressing P switches to a mode where the image from every pixel is displayed.
R	Resets the LA_View software and the SCM. This is used when you experience image problems in LA_View.
S	Saves the laser stripe image. If you press S, you are asked to enter a filename. The default filename is the serial number of the sensor with an extension of **.IMG**; however, you can specify a different filename. The filename must conform to DOS file naming conventions.

Command Line Toggles

Table 11-4 lists parameters (*n*) that can be added to the command line when starting LA_View:

la_view *n*

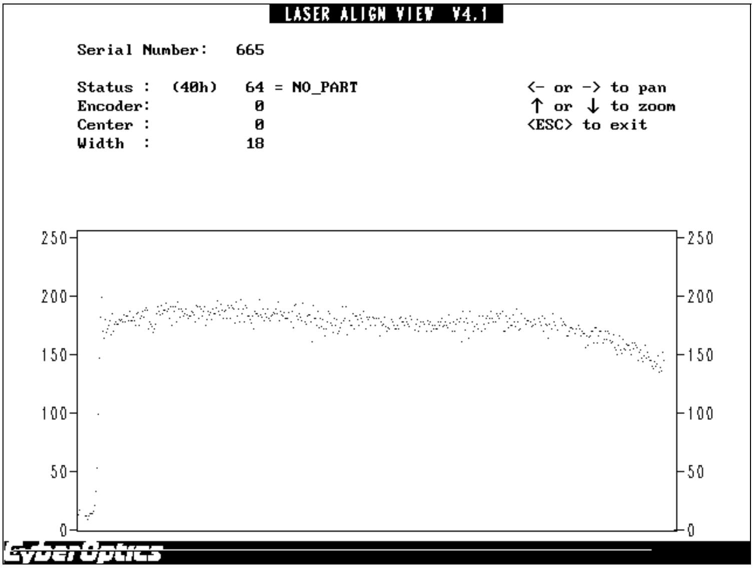
Table 11-4: Command Line Toggles available in LA_View

Parameter	Action
-a [filename]	Automatically saves the image you view to the specified file. If no filename is specified, it saves the image in a file named *sensor serial* #.img.
-d	Turns a date in the display screen on. Default is off.
-i	Activates image check mode. Default is off.
<filename>	Brings up LA_View with a stored laser stripe image displayed.
-v	Vector of COM_422.
-?	Gives you a list of command line toggles.

Normal Display Mode

The normal mode screen is shown in Figure 11-1.

Figure 11-1: Normal Mode Screen



The image at the bottom of the screen shows the laser light intensity value reported by a sample of the sensor’s detector. The intensity values range from 0 to 255.

In a regular (non-magnified, non-zoomed) image:

- Low Profile and Focused Modular Laser Align sensors display every fifth pixel.
- MiniLAM sensors display every other pixel.
- High Density Laser Align sensors display every fourth pixel.

You can press P to plot the intensity of each pixel.

You can magnify the image and pan across the image to see greater detail by using the arrow keys.

Normal Display Parameters

Table 11-5 outlines the parameters displayed in the normal mode screen.

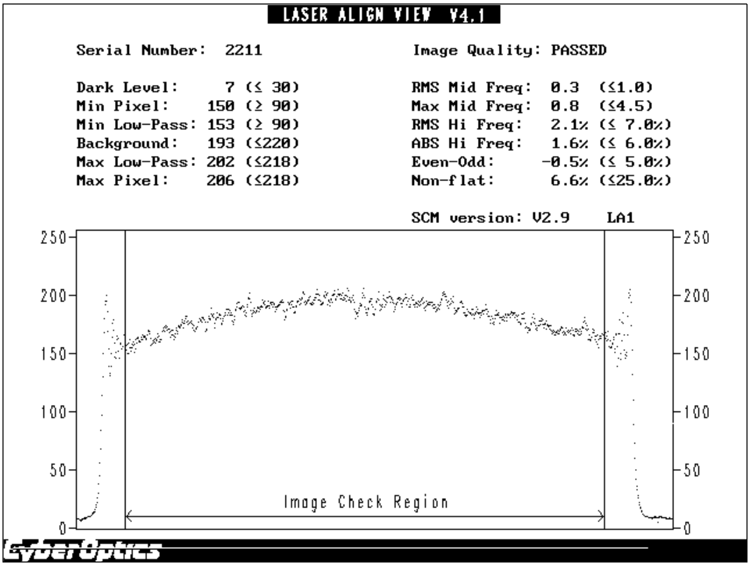
Table 11-5: Parameters in the Normal Display Screen

Parameter	Description
Serial Number	Serial number of the Laser Align sensor being used.
Status	Current status code reported by the SCM. ➡ See Chapter 9 for more information on status codes and their meanings.
Encoder	Current encoder value reported by the SCM in encoder counts.
Center	Current center position value reported by the SCM. This value is reported in units of one-tenth (1/10) pixels.
Width	Current shadow width value reported by the SCM. This value is reported in units of one-tenth (1/10) pixels.

Image Check Display Mode

The image check mode screen is shown in Figure 11-2.

Figure 11-2: Image Check Mode Screen



The image check screen displays a laser stripe image, just like the normal mode screen. However, the image check screen also displays vertical lines that indicate the region over which the image check is performed. Intensity data falling outside the *image check region* is ignored.

Note The image check region is specified in the LA_View configuration file (**la_view.cfg**).

This screen also has a pass/fail indicator for laser stripe image quality at the top of the screen.

Image Check Display Parameters

Table 11-6 outlines the parameters displayed in the image check mode screen.

Table 11-6: Parameters Displayed in the Image Check Display

Parameter	Description
Serial Number	Serial number of the Laser Align sensor used.
Dark Level	The gray level of the parts of the image receiving no illumination.
Min Pixel	Minimum raw pixel intensity value.
Min Low-Pass	Minimum pixel intensity value in a low-pass filtered image.
Background	The gray level exceeded by 25% of the pixels (25% of pixels have higher intensity and 75% have intensity ≤ background value). This is the gray level representation of the image check region.
Max Low-Pass	Maximum pixel intensity value in low-pass filtered image.
Max Pixel	Maximum raw pixel intensity value.
Image Quality	Pass/fail indication of the quality of the laser stripe image.
RMS Mid Freq	RMS value of median filtered image. This summary measure correlates well with actual edge-finder errors.
Max Mid Freq	Maximum median filtered estimated edge error.
RMS Hi Freq	RMS high frequency noise.
ABS Hi Freq	Absolute high frequency noise.

Table 11-6: Parameters Displayed in the Image Check Display (continued)

Parameter	Description
Even-Odd (not applicable for Low Profile and Focused Modular Sensors)	On some sensors, the CCD array has two channels: one for the even pixels and one for the odd pixels. The even-odd value is a measure of internal channel balance. The two channels pass through separate amplifiers before being converted. The displayed even-odd value is the difference between even and odd pixels divided by their sum: $(\Sigma e - \Sigma o)/(\Sigma e + \Sigma o)$. This value is expressed as a percentage ranging from -100% to 100%.
Non-flat	RMS value of [(median-filtered) - background)]. The non-flat value is a measure of overall non-flatness of the image. Errors in the sensor's optical alignment will be detected by this summary measure.

You can use these parameters to monitor the quality of the laser stripe image. Some parameters correlate well with the actual measured accuracy of the component edge finding. Other parameters are intended mainly for manufacturing inspection and quality control. All image analysis is done within the image check region indicated by the vertical lines.

Using LA_View

There are a number of different operations you can perform using LA_View. Your use of the program will depend upon your specific needs.

▼▼ To launch LA_View

1. Make sure that COM_422 and LA_View are installed on your PC.

   See Installing Laser Align Software in Chapter 4 for information on installation.

2. Go to the DOS prompt and switch to the **c:** drive root directory.

3 Type **cd c:\lasalign**, then press J to verify that you are in the correct directory.

4 Type **set la_view_cfg = <pathname>**, then press J where:

<pathname> = the full path and filename of the LA_View configuration file (e.g., **c:\lasalign\la_view.cfg**)

This sets an environmental variable used by LA_View to find the LA_View configuration file.

5 Type **com_422 [-annn] [-in] [-o** or **-u] [-q] [-vnn]** where:

Parameter	Description
-a*nnn*	Specifies the hexadecimal base I/O address for the RS422 card to the value of *nnn*. (Default: 3E8 hex)
-i*n*	Specifies the hardware interrupt number (IRQ) for the RS422 card to the value *n*. (Default: 4)
-o or -u	Removes the driver at the specified vector. (The -v*nn* switch must be used with this switch if removing from non-default software interrupt vector.)
-q	Inhibits messages to the standard output device.
-v*nn*	Specifies the hexadecimal software interrupt vector to use for the driver. (Default: 60 hex)

6 Press J.

This executes the COM_422 program which allows communication through the RS422 interface.

7 Type **la_view**, then press J.

This will bring up the LA_View program in normal display mode.

Note If you do not use the default address for COM_422 (60_{hex}), you must type **la_view -v <vector>** where **-v <vector>** is the COM_422 address. You can also use other command line toggles to alter the way in which LA_View is initially displayed.

➲ See *Command Line Toggles* earlier in this chapter for more information.

▼▼ To save an image file

1 Start LA_View.

2 Perform whatever steps are necessary to display the desired image and mode.

3 Press S.

The program prompts you to enter a filename.

4 Name the file:

- Type in a filename, observing DOS conventions; OR
- Accept the default filename (*sensor serial number*.IMG)

5 Press J.

The image will be saved in the file and directory specified in **Step 4** as an image check mode display. It can be retrieved for later use.

Note You can also save an image file using command line toggles. Specifically, you would use the **-a <filename>** toggle when you launch LA_View. The image will be automatically saved to the filename you specify when you exit LA_View.

➲ See *Command Line Toggles* earlier in this section for more information.

▼▼ To load an image file

To load an image into LA_VIEW software, you must use a command line toggle when you start the program.

1. Go to the DOS prompt and switch to the **c:** drive root directory.
2. Type **cd c:\lasalign**, then press J to verify that you are in the correct directory.
3. Type **set la_view_cfg = <pathname>**, then press J where:

   <pathname> = the full path and filename of the LA_View configuration file (e.g., **c:\lasalign\la_view.cfg**)

   This sets an environmental variable used by LA_View to find the LA_View configuration file.
4. Type **com_422 [-annn] [-in] [-o** or **-u] [-q] [-vnn]** where:

Parameter	Description
-a*nnn*	Specifies the hexadecimal base I/O address for the RS422 card to the value of *nnn*. (Default: 3E8 hex)
-i*n*	Specifies the hardware interrupt number (IRQ) for the RS422 card to the value *n*. (Default: 4)
-o or -u	Removes the driver at the specified vector. (The -v*nn* switch must be used with this switch if removing from non-default software interrupt vector.)
-q	Inhibits messages to the standard output device.
-v*nn*	Specifies the hexadecimal software interrupt vector to use for the driver. (Default: 60 hex)

5. Press J.

   This executes the COM_422 program, which allows communication through the RS422 interface.

6 Type **la_view <filename>** and press J.

Example: If the filename were **2234.img**, you would type:

la_view 2234.img

LA_View will display the saved laser stripe image in the image check format.

Note If you do not use the default address for COM_422 (60_{hex}), you must type **la_view -v <vector>** where **-v <vector>** is the COM_422 address.

Chapter 12

Maintenance

Cleaning the Sensor Windows

Laser Align's optical filter windows are exposed and should be treated and cleaned with the same caution used when caring for a high quality camera lens. Fingerprints and excessive amounts of dust on these windows will seriously degrade the performance of the sensor.

If the windows become dirty, CyberOptics recommends gently wiping off the dust with a camel-hair brush. Then clean the windows with isopropyl alcohol and a cotton swab or lens cleaning tissue.

Upgrading/Reinstalling Firmware

The firmware for your Laser Align system resides in the SCM. If you have a system with an SCM1, the firmware is stored in an EPROM. If you have an SCM2, the firmware is stored in a Flash ROM. As a result, the upgrade procedures are different for the two types of SCM.

Always use electrostatic device (ESD) protection when working with the sensor and SCM.

Upgrading/Reinstalling SCM1 Firmware

To upgrade a Laser Align system with an SCM1, you must obtain a new EPROM with the upgraded firmware from CyberOptics. After replacing the EPROM on your SCM, you need to return the old EPROM to CyberOptics.

▼▼ To upgrade/reinstall SCM1 firmware

1. Power down the SCM/sensor power supply.
2. Carefully remove the old EPROM from the SCM1 (Figure 12-1).

   Use a chip puller, taking care not to bend the pins.
3. Install the new EPROM on the SCM1.

   Use the silkscreen on the board as a frame of reference, and make sure that the notch on the EPROM matches the notch in the silkscreen.
4. Power up the SCM/sensor power supply.

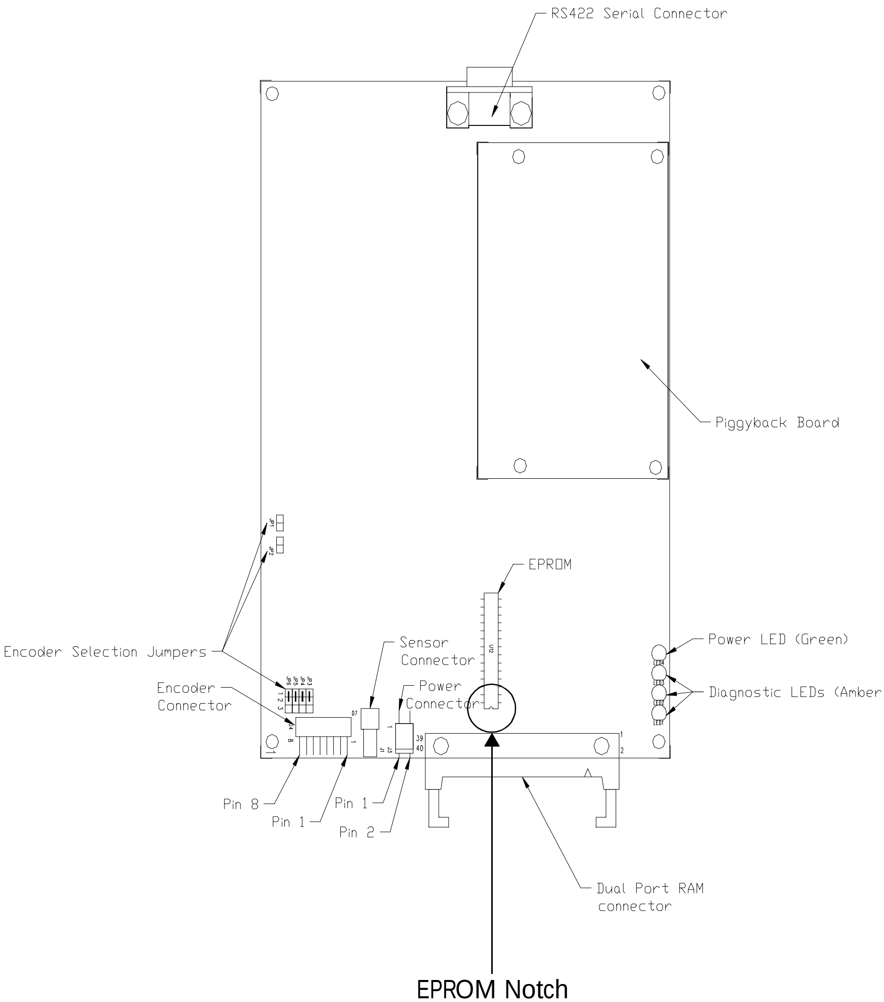
   The LEDs should go through an initialization sequence, then should go out. If one or more of the LEDs stays lit, see *SCM LEDs* in Chapter 13 for more information.

5 Verify that the expected firmware version is installed by sending **VERSION_CMD (32)** and checking the response.

- See *VERSION_CMD (32)* in Chapter 7 for more information.

This completes the SCM1 firmware upgrade.

Figure 12-1: SCM1



Upgrading/Reinstalling SCM2 Firmware

To upgrade firmware on a Laser Align system with an SCM2, you must obtain a diskette with the new firmware from CyberOptics. You will also need a Motorola Programming Module. One of these modules is supplied with the initial shipment of an SCM2 system. Additional modules are available from CyberOptics.

▼▼ To upgrade/reinstall SCM2 firmware

1. If there is no **scm2prog** directory on your hard drive for SCM2 programming updates, perform **Steps 2** through **3**. If there is an **scm2prog** directory on your hard drive, go to **Step 4**.
2. Go to the DOS prompt and switch to the **c:** drive root directory.
3. Type **md \lasalign\scm2prog** and press J to make a directory for the new firmware.

   Note The **lasalign** directory must already exist.

4. At the DOS prompt, type **cd \lasalign\scm2prog** and press J to switch to the directory for SCM2 programming updates.
5. Insert a diskette containing the new firmware and the Laser Align SCM2 Programming Utility Software into the diskette drive.
6. Type **copy a:*.* c:\lasalign\scm2prog** and press J. (Use your own drive designation for **a**.)

   The files for the SCM2 firmware upgrade will be copied into the designated directory.

7. Power down the SCM/sensor power supply.
8. Plug the Motorola Programming Module into the parallel port on your PC.

   Note A parallel port extension cable between the Programming Module and the PC is recommended for easier access to the Programming Module.

9 Plug the other end of the module into the FLASH ROM programming connector on the SCM2 (Figure 12-2 and Figure 12-3).

10 Activate the SCM/sensor power supply.

11 Type **scm2prog** at the DOS prompt, and press J.

The firmware version on the SCM will be reported back. (This step checks the communication between the PC and the SCM.)

Note If your parallel port address is not set to 378h, type **scm2prog -a <parallel port address>** and press J.

12 Type **scm2prog -w <firmware file name>** (or **scm2prog -a <parallel port address> -w <firmware file name>**) and press J.

Note The firmware file has a **.hex** extension and is on the disk supplied by CyberOptics. You must include the extension as part of the filename.

The firmware will be loaded onto the SCM and the firmware version will be reported back.

13 Type **scm2prog** (or **scm2prog -a <parallel port address>**) and press J to verify the firmware download.

The firmware version will be reported back. It should reflect the upgrade.

14 Power down the SCM/sensor power supply.

15 Disconnect the Motorola Programming Module from the SCM and from your PC.

- See the **SCM2PROG.DOC** file supplied on your upgrade diskette for more information about commands.

Figure 12-2: SCM2 with SCM1 Compatible Connectors

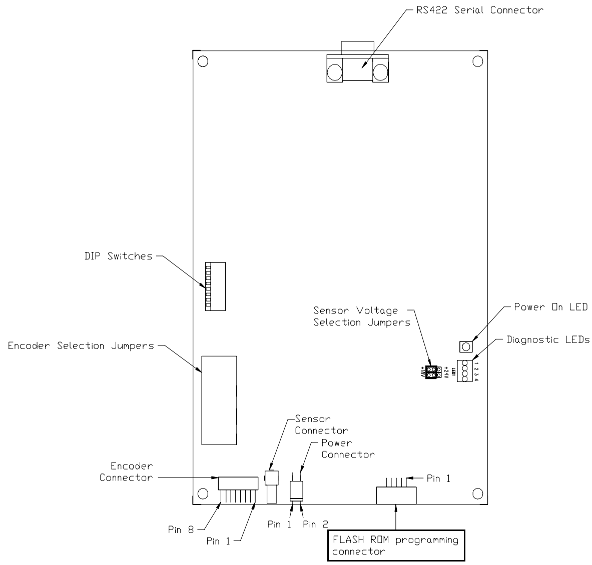
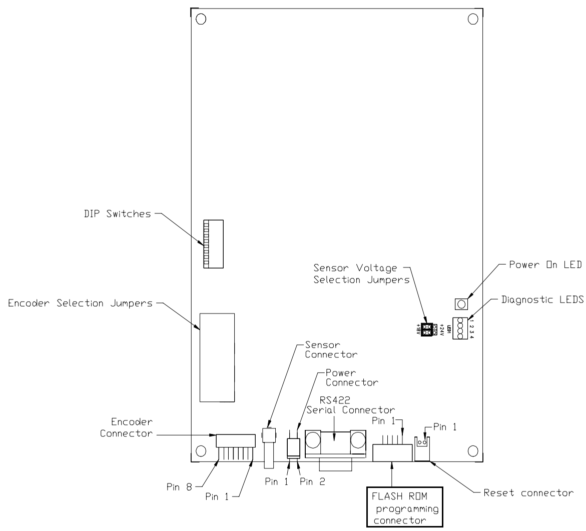


Figure 12-3: SCM2 with All Connectors at One End



Chapter 13

Troubleshooting

Hardware Troubleshooting

SCM LEDs

Only the green power LED on your SCM should remain lit within 5 seconds of powering up. If other LEDs are on or flashing after power up, there is a problem. Table 13-1 and Table 13-2 will help you isolate the cause of the problem.

Table 13-1: SCM1 LED Status

Symptom	Cause	Action to Take
All three amber LEDs stay on without flashing after power up.	Any failure on the circuit board causing the processor not to start. This includes bad boot EPROMs. **Note:** The boot EPROMs are the pair of EPROMs on the small SCM piggyback board.	Call CyberOptics for assistance.
The amber LED next to the green LED flashes after initial LED activity.	This is a checksum error on the boot EPROMs. **Note:** The boot EPROMs are the pair of EPROMs on the small SCM piggyback board.	Call CyberOptics for assistance.
The middle amber LED flashes after initial LED activity.	The application EPROM was not detected. **Note:** The application EPROM is the EPROM on the large SCM circuit board.	Install or replace the EPROM, available from CyberOptics.

Table 13-1: SCM1 LED Status (continued)

Symptom	Cause	Action to Take
The amber LED furthest from the green LED flashes after initial LED activity.	The application EPROM was detected but had a checksum error. **Note:** The application EPROM is the EPROM on the large SCM1 circuit board.	Install a new EPROM, available from CyberOptics.
The amber LED next to the green LED stays lit without flashing.	If it remained lit after power up or RESET_CMD, the calibration information was not successfully read from the sensor.	Try replacing the sensor. Otherwise, call CyberOptics for assistance.
	If it comes on during operation, a hardware failure occurred and the specific cause should be available in the status field of a REPORT_CMD, REPORTS_CMD, or REPORTW_CMD. **Note:** It is likely that RS422 communication will be possible.	Send one of the REPORT commands to obtain the status. Take appropriate action based on the status.
The middle amber LED and/or the amber LED furthest from the green LED stay lit without flashing and don't go off after a HELLO_CMD (16).	Laser Align software stopped executing during processing. **Note:** It is likely that RS422 communication is not possible.	Perform a hardware reset by cycling power. If the problem persists, call CyberOptics for assistance.

Table 13-2: SCM2 LED Status

Symptom	Cause	Action to Take
All four red LEDs remain off without flashing after power up (i.e., the lights never came on).	Any failure on the circuit board causing processor not to start. The most common cause is faulty or non-existent firmware.	Reload the firmware using SCM2PROG; available from CyberOptics. ➡ See Upgrading/ Reinstalling Firmware in Chapter 12 for more information.
Red LED 1 flashes after initial LED activity.	There is a checksum error on the FLASH ROM.	Reprogram the FLASH ROM using SCM2PROG; available from CyberOptics. ➡ See Upgrading/ Reinstalling Firmware in Chapter 12 for more information.
Red LED 2 flashes after initial LED activity.	There is an error initializing RAM or the PAL. This is a hardware error that is probably unrecoverable.	Call CyberOptics for assistance.
Red LED 3 flashes after initial LED activity.	The DIP switches are set to an incorrect configuration.	Correct DIP switch settings and re-power the board. ➡ See *Encoder Specifications* in Chapter 14 for DIP switch settings.

Table 13-2: SCM2 LED Status (continued)

Symptom	Cause	Action to Take
Red LED 1 stays lit without flashing.	If it remained lit after power up or RESET_CMD, the calibration information was not successfully read from the sensor.	Try replacing the sensor. Otherwise, or if the problem persists, call CyberOptics for assistance.
	If it lights during operation, a hardware failure occurred and the specific cause should be available in the status field of a REPORT_CMD, REPORTS_CMD, or REPORTW_CMD. **Note:** It is likely that RS422 communication is possible.	Send one of the REPORT commands to obtain the status. Take appropriate action based on the status.
Red LED 2 and/or red LED 3 stay lit without flashing and don't go off after a HELLO_CMD (16).	Laser Align software stopped executing during processing. **Note:** It is likely that RS422 communication is not possible.	Perform a hardware reset. If the problem persists, call CyberOptics for assistance.
Red LED 4 stays on without flashing.	The electronic circuit breaker to the sensor has tripped because of an over-current condition. **Note:** RS422 communication is possible.	Remove the cause of the excess current and re-power the board, or send a RESET_CMD.

13

Communications Troubleshooting

Table 13-3: Communications Troubleshooting

Problem	Description/Solution
Data loss on the RS422 link.	Except when using the RESET_CMD (17), it is not necessary to wait longer than 200 ms for a response to the serial command. On slow processors or systems with hardware interrupts that are serviced too slowly, it is possible to lose a portion of the response to a report results command due to data overrun. If these problems occur on the placement machine's controller, CyberOptics recommends using UARTs with 16-byte FIFO buffers, which are suited to the 16-byte response format of Laser Align.

CyberOptics Service and Support

Before You Call

If you have problems operating Laser Align, first check this manual for procedures and more information.

If you still need help with Laser Align, or you discover problems with the documentation, please telephone, e-mail or fax CyberOptics Service and Support. Please include the serial numbers of your Laser Align sensor and SCM, as well as your Laser Align firmware version, in all faxes and e-mails. Have this information ready when you telephone CyberOptics.

Telephone Support

Use the Customer Service 800 number, which is toll-free in the USA. Call (800) 526-2540 to speak to a representative or leave a voice-mail message.

- The 800 number is answered from 8 AM to 5 PM Central Time (USA), Monday through Friday.
- Outside of these hours, you can call the 800 number and leave a voice-mail message. CyberOptics guarantees that all voice-mail messages will receive a response during the next CyberOptics business day.

If you have problems with the 800 number, the general telephone number at the CyberOptics Minneapolis Office (USA) is (763) 542-5000.

Fax Communication

You can fax service issues or problem descriptions to CyberOptics Service and Support staff. CyberOptics guarantees that all fax transmissions sent to the attention of Service will receive a fax response during the next CyberOptics business day.

Send your fax (attention: Service) to the CyberOptics Minneapolis Office (USA) at (763) 542-5100.

Internet Communication

You can use the Internet to send service issues or problem descriptions to CyberOptics Service and Support staff. CyberOptics guarantees that all messages received by Service will be responded to before or during the next CyberOptics business day.

- Send e-mail to:

  service@cyberoptics.com

- Visit the CyberOptics home page on the World Wide Web at:

  http://www.cyberoptics.com

Limited Warranty and Disclaimer

The CyberOptics 1-year limited warranty covers the cost of all maintenance, parts, and labor for hardware and software for 1 year from the purchase date.

You are responsible for the cost of shipping defective parts to CyberOptics. If you request, CyberOptics provides shipping containers for returned parts. You are responsible for any damage resulting from improper packaging.

This warranty does not apply to any item that you install, modify, neglect or use in any way that adversely affects Laser Align as ordered and delivered. You are responsible for the cost of repairs (at CyberOptics standard service rates), including labor and parts, made as a result of these causes.

CYBEROPTICS DISCLAIMS LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING FROM THE USE OR MISUSE OF ANY OF ITS PRODUCTS. THE CUSTOMER IS RESPONSIBLE FOR ANY OPERATION OF THIS PRODUCT NOT RECOMMENDED IN THIS MANUAL.

Extended Warranty Options

CyberOptics offers you the option of extending the conditions of your 1-year Limited Warranty each year.

Warranty Services

CyberOptics offers on-site and factory service for products manufactured by CyberOptics. Upgrades or replacements can be quoted by CyberOptics.

CyberOptics reserves the right to determine the need for on-site parts repair or replacement. To do this, CyberOptics first identifies the problem via telephone or fax.

Emergency services are available on an as-required basis.

Parts that you replace are normally shipped second-day air, and CyberOptics provides a Returned Material Authorization (RMA) number and a return shipping label for the return of the defective part. If the need is urgent, overnight air shipment can be used where available.

When on-site service is required, CyberOptics personnel will visit your site, examine the problem and make the necessary repairs. When on-site repairs are not possible, CyberOptics will provide a prompt solution to the problem.

You must call, fax, or e-mail CyberOptics for a written estimate on all non-warranty service events. For on-site service, you are required to have a purchase order number before the service representative arrives at the site.

On parts covered under another manufacturer’s warranty, CyberOptics may use the original manufacturer’s service organization to handle field repairs.

A contract service organization may be employed for on-site repairs. In these instances, CyberOptics schedules the repair, with final confirmation taking place directly between you and the service contractor.

Non-Warranty Services

CyberOptics offers you these services and packages:

- On-site maintenance
- Software upgrades
- Firmware upgrades
- Hardware upgrades
- Factory repair service
- Preventive maintenance
- Laser Align certification
- Laser Align calibration
- Spare parts kits
- Maintenance training
- Software training
- Additional documentation

For prices and scheduling information, or if you have any questions about how CyberOptics can help you to maintain and use Laser Align, call, e-mail, or fax CyberOptics Service and Support staff.

Software Licensing and Upgrades

You are responsible for implementing CyberOptics software upgrades.

You are responsible for registering and maintaining the license numbers of software packages that are included in the CyberOptics product. CyberOptics is not responsible for consequences if you manipulate or upgrade the software products that are included in the CyberOptics product.

Chapter 14

Specifications

About Specifications

These specifications are provided as reference only and are subject to change at any time. For detailed specifications, contact your CyberOptics representative.

Environmental Conditions

Table 14-1: Laser Align Environmental Conditions*

Operating temperature range	10–35° C (50–95° F)
Storage temperature range	0–50° C (32–122° F)
Humidity range	0–85% non-condensing

* Stresses beyond minimums and maximums listed under Environmental Conditions may cause permanent device failure. Functionality at or beyond these limits is not implied. Exposure to minimum or maximum Environmental Conditions for extended periods may affect device reliability.

Power Input

Depending on the Laser Align sensor model, power for the Laser Align system is 18 volts or 24 volts DC (nominal) at 1.0 amps. Slightly more current may be drawn at lower voltages. No other power connections are needed.

The ground connection for power input is directly tied to the shield of the sensor cable on all sensors, and to the case on some sensors.

- See *Sensor Specifications* for details about which sensors have the ground connection tied to the case.

The input power is supplied on pin 1 of the SCM power connector. Pin 2 of that connector is ground.

Table 14-2: Power Connector Pin Connections

Pin Number	Definition
1	Power
2	Ground

- See Figure 14-2, Figure 14-3, and Figure 14-4 later in this chapter for the location of the power connector and to identify the connector pins on the various SCM models.

RS422 Serial Communication Interface

The Laser Align SCM communicates with the host controller over an RS422 serial communication link.

CyberOptics provides a Microsoft DOS low-level communications software driver (COM_422.EXE), which assists in communication over the RS422 port with the SCM system when used with a CIO-COM-422 card (or equivalent). This software driver is a Terminate and Stay Resident (TSR) program. If your software application is a DOS application and you wish to use this driver, you must run this program on the host system before communications with the Laser Align system is attempted.

➲ See Chapter 11 for more information about the software.

Communications Settings

For the host controller to communicate with the SCM, it must:

- Have hardware communication settings the same as the SCM (Table 14-3).
- Contain hardware that communicates following the RS422 standard (Table 14-4).
- Have a properly wired communications cable (Table 14-5).

Table 14-3: Hardware Communications Settings

Communications Speed	**SCM1 baud rate:** Programmed in the EPROM, various rates available. **SCM2 baud rate:** Switch selectable (Table 14-6).
Data Length	8 data bits
Synchronization	2 stop bits
Error Detection	No parity

Pin Connection

Table 14-4: SCM RS422 Serial Pin Connections

Pin Number	Definition
1	GND
2	RTS+
3	RTS-
4	TXD+
5	TXD-
6	CTS-
7	CTS+
8	RXD+
9	RXD-

Unless hardware flow control is enabled by software command, RTS and CTS are not used by Laser Align and can be left unconnected.

See *Defining General System Parameters* in Chapter 5 for more information about flow control.

Cable Specifications

Table 14-5: Cable used to connect Laser Align SCM to an RS422 interface of the type used by CyberOptics

DB9 Male Connector		DB9 Male Connector		
Pin Number	**Function**	**Function**	**Pin Number**	
1	GND	GND	1	Shield
2	RTS+	CTS+	7	Twisted Pair
3	RTS-	CTS-	6	
4	TXD+	RXD+	8	Twisted Pair
5	TXD-	RXD-	9	
6	CTS-	RTS-	3	Twisted Pair
7	CTS+	RTS+	2	
8	RXD+	TXD+	4	Twisted Pair
9	RXD--	TXD-	5	

The cable is symmetrical. Either end can be connected to the SCM or the RS422 interface on a CIO-COM-422 card (or compatible).

SCM2 Baud Rate Settings

Table 14-6: SCM2 Baud Rate

	Position	Baud Rate	Position 1	Position 2	Position 3
DIP Switch S2	1–3	57600	ON	ON	ON
		55556	OFF	ON	ON
		38400	ON	OFF	ON
		19200	OFF	OFF	ON
		9600	ON	ON	OFF
		reserved	OFF	ON	OFF
		reserved	ON	OFF	OFF
		reserved	OFF	OFF	OFF
	4–7	reserved: must be ON			
	8	Encoder type:ON= quadrature OFF = step/direction			

Encoder Specifications

To correlate orientation information with the placement head position, the Laser Align SCM must be connected to the θ axis encoder signals on the placement machine.

Connections from the encoder to the SCM consist of a signal ground line, a pair of phase-quadrature or step / direction encoder lines, an index line, and a 5-volt signal used for reference. The encoder signals are expected to be TTL-level. For phase quadrature mode, the direction of rotation is positive when the phase of *A* leads the phase of *B*. For step / direction mode, a logic 1 on the direction pin indicates count up, and a logic 0 indicates count down. The Laser Align counter is incremented on the high-to-low transition of the signal on the step pin.

The Laser Align counter decodes the quadrature or step / direction encoder data. The counter has 24 bits of significance. The 24-bit count values are sign-extended to 32 bits.

For example:

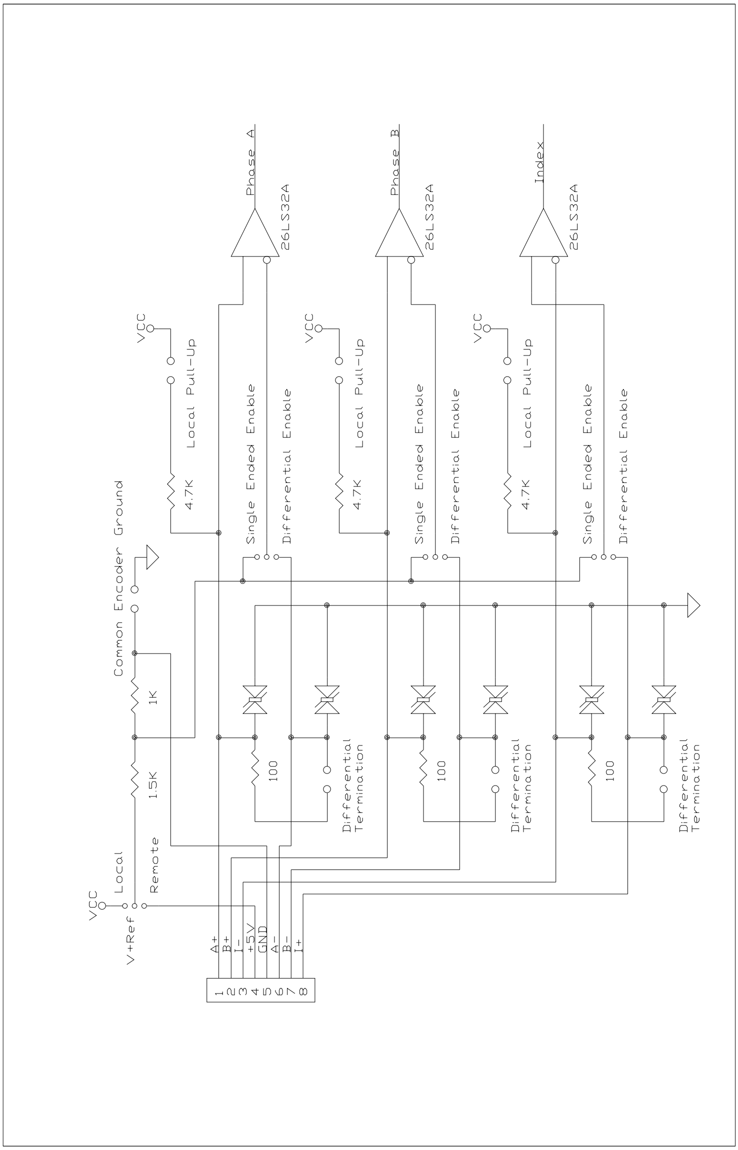
Decimal Value	SCM Encoder Value (hexadecimal)
20	00000014
-20	$\underbrace{\text{FF}}_{\text{8 bits}}\underbrace{\text{FFFFEC}}_{\text{24 bits}}$

The maximum position increment rate for the counter in the Laser Align SCM is 4.8 MHz. Each phase of the quadrature encoded signal must stay constant for a least 400 ns, and the minimum time between position counts is 200 ns. Noise, slow rise and fall times, or non-symmetric signals result in degradation of the maximum counting rate. If the placement head is spun faster than the maximum counting rate, some position counts are lost, resulting in incorrect position information. Figure 14-1 shows the encoder input circuit for an SCM2.

The encoder and SCM share a common ground when the common encoder ground jumper is set.

If the common encoder ground is not set, the encoder ground voltage should not differ from the power-in ground voltage by more than 5 volts. Excessive voltages on encoder inputs can damage the Laser Align SCM.

Figure 14-1: SCM2 encoder input circuit



Connecting the Encoder to the SCM

An encoder input may be configured to accept either a single-ended signal or a differential signal. The inputs may be either quadrature encoder or stepper motor *step* and *direction*.

SCM1 can be configured for two single-ended inputs.

The encoder input connector is a 8-pin male header; see Table 14-7 and Table 14-8 for the definitions of the encoder connector pins for differential and single-ended inputs. Figure 14-2, Figure 14-3, and Figure 14-4 show the location of the encoder connector and its pins.

There are different jumper and DIP switch connections required for differential or single-ended inputs. The type of SCM you have affects the method of setup.

- SCM1 needs to have encoder jumpers set.
- SCM2 needs to have encoder jumpers and DIP switches set.

Note The Laser Align Sensor system you receive from CyberOptics should be pre-set to the encoder settings you specify. If you change the type of encoder signals sent to the SCM, you will need to change the encoder jumpers and DIP switches accordingly.

Figure 14-2: SCM1 connectors

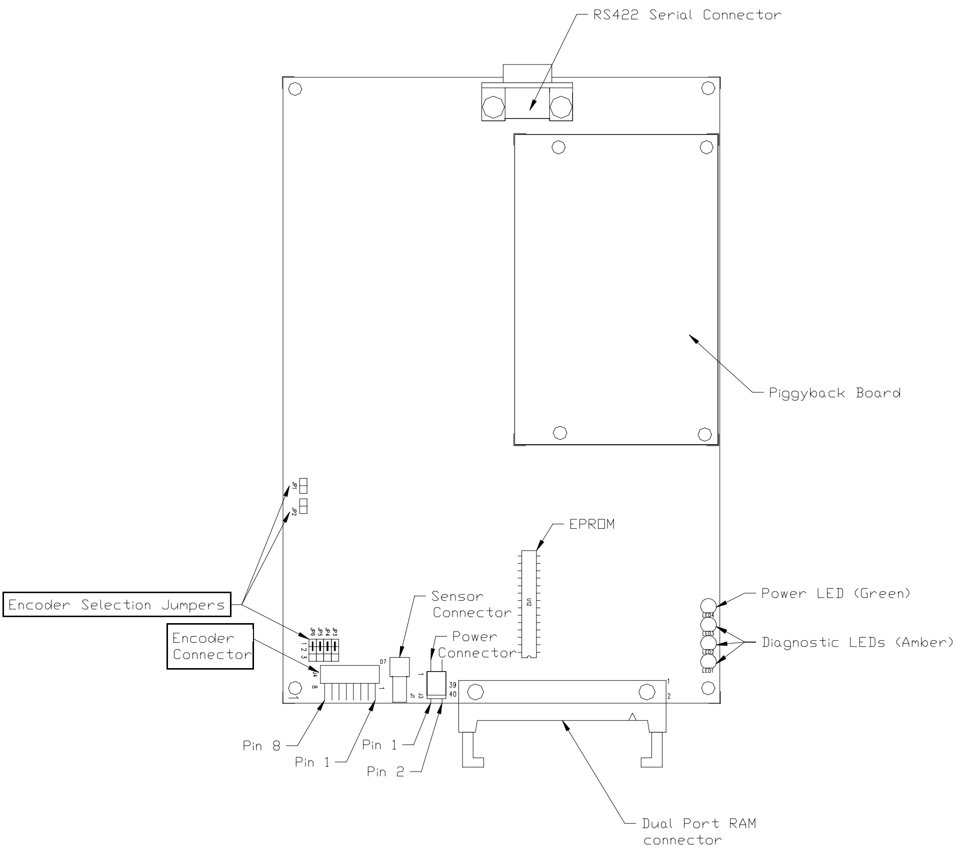


Figure 14-3: SCM2 with SCM1 compatible connectors

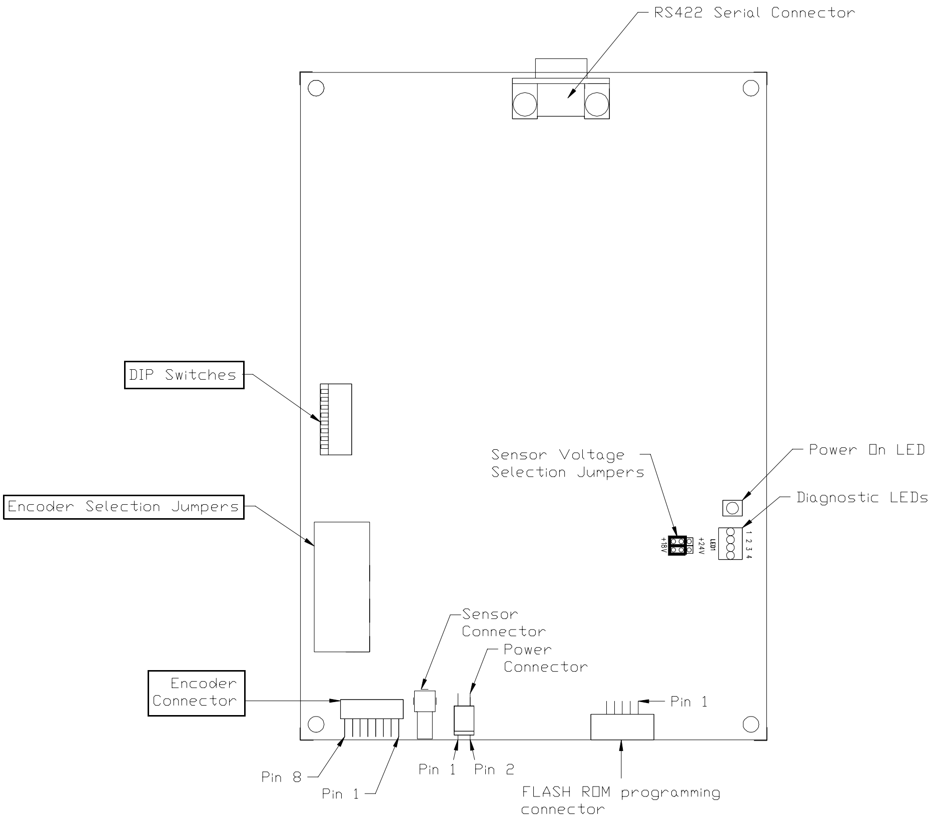
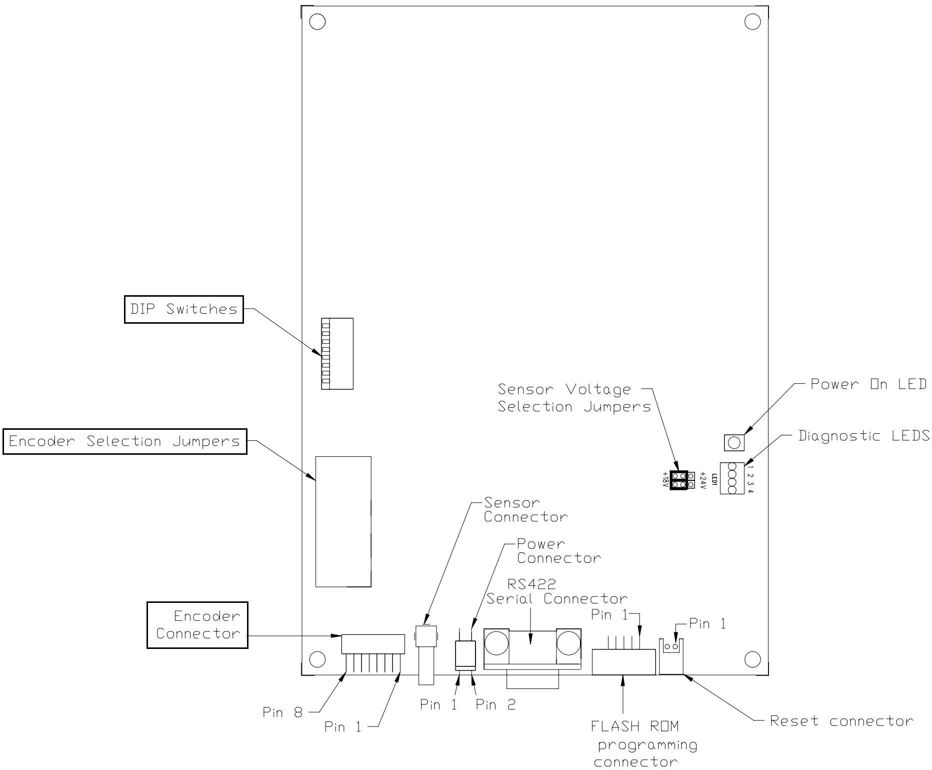


Figure 14-4: SCM2 with all connectors at one end



Encoder Pin Definitions

Table 14-7: Pin Definitions for Differential Encoder Input

Pin Number	Definition	
	Quadrature	Step and Direction
1	A+	Step+
2	B+	Direction+
3	I-	I- (not connected on SCM1)
4	not connected	not connected
5	Ground	Ground
6	A-	Step-
7	B-	Direction-
8	I+	I+ (not connected on SCM1)

Table 14-8: Pin Definitions for Single-Ended Encoder Input

<table>
<tr><th rowspan="2">Pin Number</th><th colspan="3">Definition</th></tr>
<tr><th>Encoder</th><th>Quadrature</th><th>Step and Direction</th></tr>
<tr><td>1</td><td rowspan="3">Encoder 0</td><td>A+</td><td>Step+</td></tr>
<tr><td>2</td><td>B+</td><td>Direction+</td></tr>
<tr><td>3</td><td>I-</td><td>I- (N/C on SCM1)</td></tr>
<tr><td>4</td><td colspan="3">+5V (used to generate input logic threshold voltage)</td></tr>
<tr><td>5</td><td colspan="2">Ground</td><td>Ground</td></tr>
<tr><td>6</td><td rowspan="3">Encoder 1 (SCM1 only)</td><td>A+</td><td>Step+</td></tr>
<tr><td>7</td><td>B+</td><td>Direction+</td></tr>
<tr><td>8</td><td>I-</td><td>I- (N/C on SCM1)</td></tr>
</table>

SCM1 Jumper Settings

Table 14-9: SCM1 Jumper Settings for Differential Encoder Input

Jumper Number	Position
JP1	Open - Step and Direction Closed - Quadrature
JP2	Not used
JP3 – JP5	Jumpers across pins 2 and 3
JP6	**For separate encoder and SCM grounds:** Jumpers across pins 1 and 2 (no connection between encoder cable ground and SCM ground) **For common encoder ground and SCM ground:** Jumpers across pins 2 and 3

Table 14-10: SCM1 Jumper Settings for Single-Ended Encoder Input

Jumper Number	Position
JP1, JP2	Open - Step and Direction Closed - Quadrature
JP3 – JP5	Jumpers across pins 1 and 2
JP6	**For separate encoder and SCM grounds:** Jumpers across pins 1 and 2 (no connection between encoder cable ground and SCM ground) **For common encoder ground and SCM ground:** Jumpers across pins 2 and 3

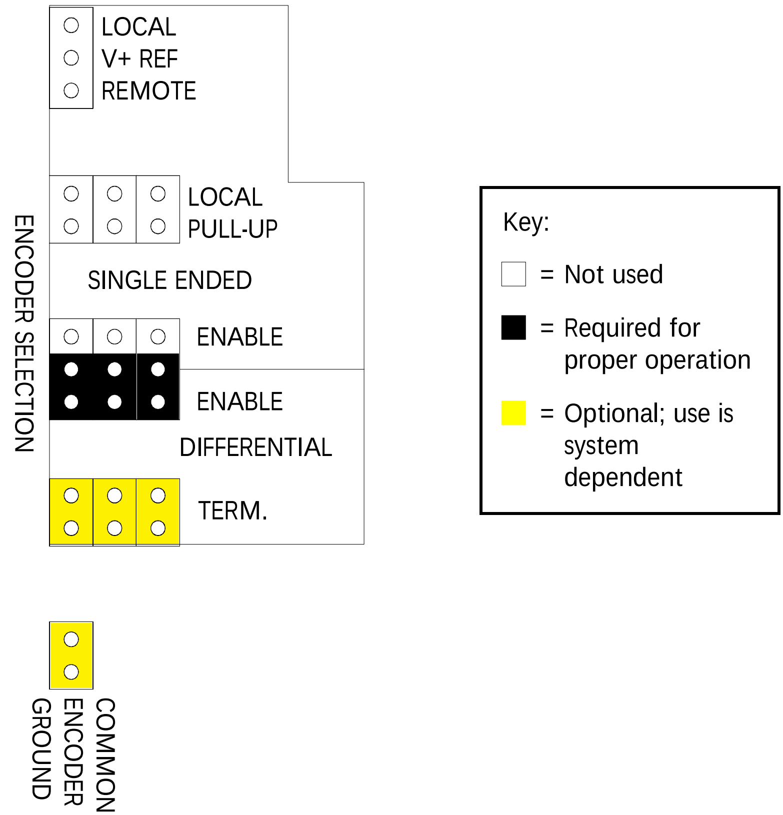
SCM2 Jumper Settings

On SCM2, the encoder interface is set with a block of jumpers.

- Figure 14-5 shows the jumpers used to set the SCM2 encoder interface to *differential*.
- Figure 14-6 shows the jumpers used to set the SCM2 encoder interface to *single-ended input*.

SCM2 - Differential Encoder Interface Settings

Figure 14-5: SCM2 Jumper Settings for Differential Encoder Interface

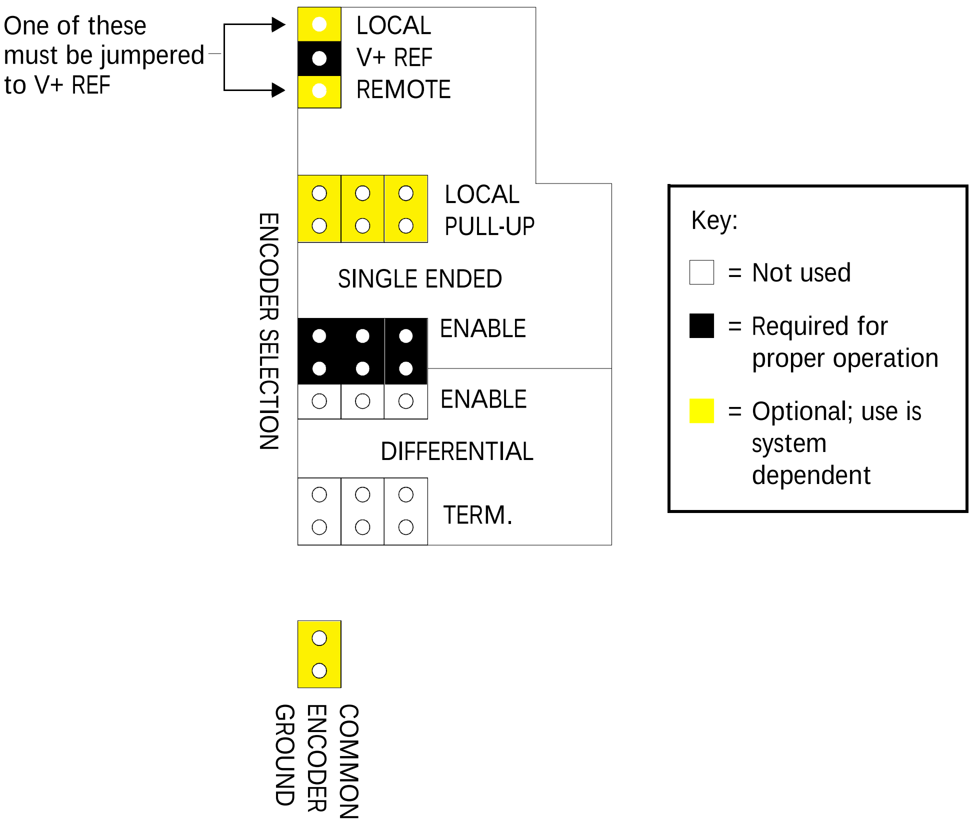


The signal termination jumpers (TERM.) should be installed if you want termination. If you do not want termination of those signals, remove the jumpers. Termination is not typically handled by the SCM.

The common encoder ground jumper should be installed if you want the encoder ground to be common with the SCM ground. This is typically done when operating in differential mode.

SCM2 - Single-Ended Encoder Interface

Figure 14-6: SCM2 Jumper Settings for Single Ended Encoder Interface



The reference voltage (V+ REF) jumper should be set to LOCAL if you want the reference voltage to be generated from the SCM VCC. If you are supplying the reference voltage (from pin 4 on the encoder connector), set this jumper to REMOTE. Typically this jumper is set to REMOTE.

If you want the pull-up voltage to be supplied by the SCM, set the LOCAL PULL-UP jumpers. If you are supplying the pull-up voltages, remove the jumpers from the LOCAL PULL-UP blocks. Typically the LOCAL PULL-UP jumpers are not connected.

The jumper setting the common encoder ground should be set if you want the encoder ground to be common with the SCM ground. Typically the common encoder ground is not set when operating in single-ended mode.

SCM2 DIP Switch Settings

On SCM2, the encoder type is set with DIP switch S2. Table 14-11 details the DIP switch settings for encoder type.

Table 14-11: SCM2 DIP Switch S2 settings

	Position	Description			
		Baud Rate	**Position 1**	**Position 2**	**Position 3**
DIP Switch S2	1-3	57600	ON	ON	ON
		55556	OFF	ON	ON
		38400	ON	OFF	ON
		19200	OFF	OFF	ON
		9600	ON	ON	OFF
		reserved	OFF	ON	OFF
		reserved	ON	OFF	OFF
		reserved	OFF	OFF	OFF
	4-7	reserved: must be ON			
	8	Encoder type:ON= quadrature OFF = step/direction			

14

SCM Specifications

Table 14-12: SCM Specifications

		SCM1	SCM2
Power	**Voltage**	as required by sensor	as required by sensor
	Absolute Maximum Voltage	30 V	30 V
	Maximum Current (includes sensor)	1.0 amp Typical power-up inrush 2.0 amps	1.0 amp See Table 14-13 for SCM2 sensor power-up inrush information.
422 Serial	**Baud Rate**	set in EPROM	switch-selectable
	Data Bits	8	8
	Parity	none	none
	Stop Bits	2	2
	Handshake	RTS/CTS can be enabled via software command	RTS/CTS can be enabled via software command
Encoder	**Maximum Count Rate**	4.8 MHz	4.8 MHz
	Minimum time between counts	200nS	200nS

SCM Sensor Power-Up Current

Table 14-13 shows the nominal peak inrush current and decay time to 1.0 amp for the SCM2 when powering up the sensor. It shows the nominal peak sensor inrush current for all Laser Align sensor models in an SCM2 system. Sensor power-up inrush for the SCM1 is shown in Table 14-12.

Table 14-13: SCM2 Nominal Peak Sensor Inrush Current and Decay Time to 1.0 Amp

SCM2 Model	Low Profile Laser Align	Focused Modular Laser Align (top and side connector models)	MiniLAM	High Density Laser Align
with SCM1 compatible connectors:				
5120126	7.7 amps 1.2 ms	7.7 amps 1.3 ms	6.4 amps 0.23 ms	6.4 amps 0.23 ms
5120185	5.5 amps 1.2 ms	5.2 amps 1.2 ms	3.3 amps 0.14 ms	3.4 amps 0.14 ms
5120089	n/a	n/a	1.47 amps 0.14 ms	n/a
with all connectors at one end:				
5120141	7.7 amps 1.2 ms	7.7 amps 1.3 ms	6.4 amps 0.23 ms	6.4 amps 0.23 ms
5120186	5.5 amps 1.2 ms	5.2 amps 1.2 ms	3.3 amps 0.14 ms	3.4 amps 0.14 ms

SCM Dimensions

All dimensions in Figure 14-7, Figure 14-8, and Figure 14-9 are in millimeters.

Figure 14-7: SCM1

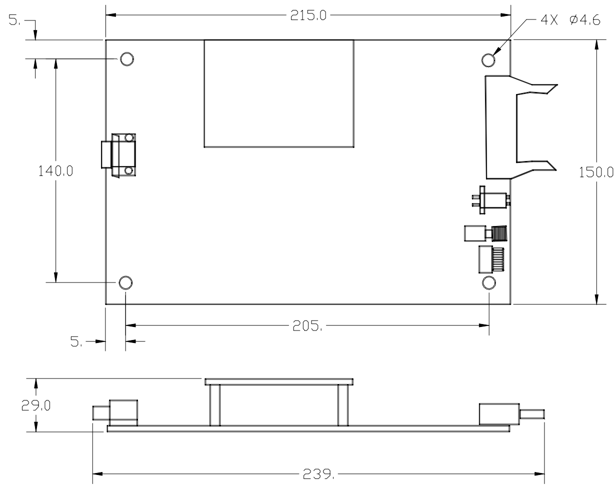


Figure 14-8: SCM2 with SCM1 Compatible Connectors

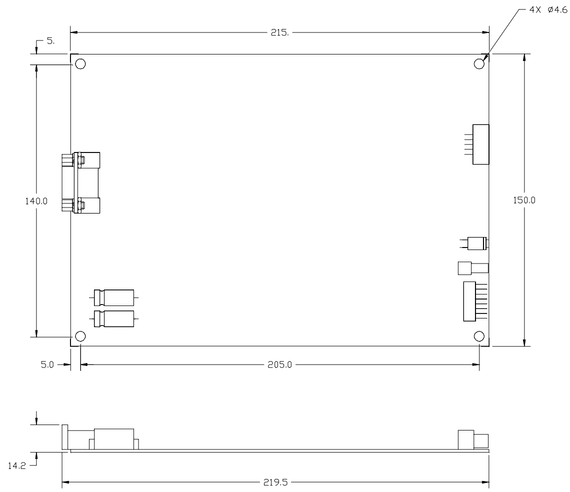
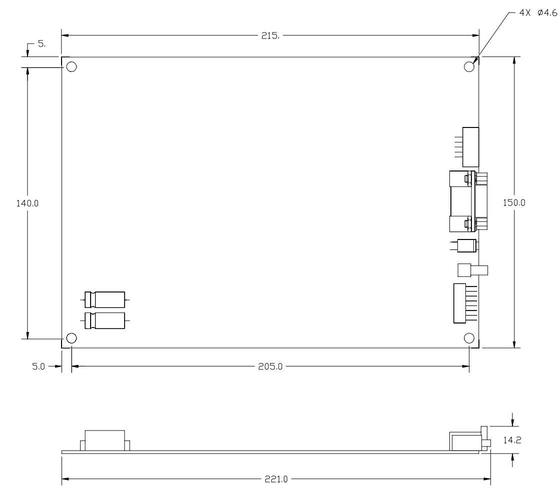


Figure 14-9: SCM2 with all connectors at one end



SCM Connectors

Table 14-14 provides information on manufacturers and part numbers for the mates CyberOptics recommends for cabling to the SCM1 and SCM2 board connectors. Part numbers are subject to change. Contact CyberOptics Service and Support for the current part number.

Table 14-14: SCM Connectors

Board	SCM Connector	Part	Manufacturer	Manufacturer's Part Number
SCM1 and SCM2	Encoder	Connector Housing	JST	H8P-SHF-AA
		Connector Contact	JST	SHF-001T-0.8SS
	Sensor	Connector	Amphenol	901-9511-3*
		Connector (alternate)	E.F. Johnson	142-0403-011
	Power	Connector Housing	JST	VHR-2N
		Connector Contact	JST	SVH-21T-1.1
SCM2 only	Reset	Connector Housing	JST	XHP-2
		Connector Contact	JST	SXH-001T-P0.6

* Available from CyberOptics

Sensor Specifications

Table 14-15: Sensor Specifications

	Low Profile Laser Align	Focused Modular Laser Align (top and side connector models)	MiniLAM	High Density Laser Align
Laser Safety Classification	IEC 825 Class I CDRH Class I	n/a	IEC 825 Class I CDRH Class I	IEC 825 Class I CDRH Class I
Detector	2500 element linear CCD array	2500 element linear CCD array	1024 element linear CCD array	2048 element linear CCD array
Effective Detector Pixel Spacing	14.0 µm	14.0 µm	12.716 µm	13.0 µm
Detector Length	35 mm	35 mm	10.35 mm	26.6 mm
Pixel Data Rate	5 MHz	5 MHz	5 MHz	5 MHz
Detector Frame Rate	563.2 µsec	563.2 µsec	409.6 µsec	512.0 µsec
Operating Voltage (supplied by SCM)	24 volts	24 volts	18 volts	18 volts
SCM power ground tied to sensor case (Yes or No)	Yes	Yes	No	Yes

Table 14-15: Sensor Specifications (continued)

	Low Profile Laser Align	Focused Modular Laser Align (top and side connector models)	MiniLAM	High Density Laser Align
Maximum Component Size*	30 × 30 mm	30 × 30 mm	10 × 10 mm	20 × 20 mm
Minimum Component Size**	0402 (1005 metric)	0402 (1005 metric)	0402 (1005 metric)	0402 (1005 metric)
Mass	310 grams (11.1 oz)	300 grams (10.7 oz)	100 grams (3.6 oz)	125 grams (4.5 oz)
Mounting Torque	2.26 Nm M4 x 0.7 screw	0.7 Nm M4 x 0.7 screw	0.3 Nm M2 x 0.4 screw	0.45 Nm M2.5 x 0.45 screw
Connector	SMA (threaded)	SMB (non-threaded)	SMB (non-threaded)	SMB (non-threaded)

* Larger sizes possible using pickup offsets and longer rotation angles.

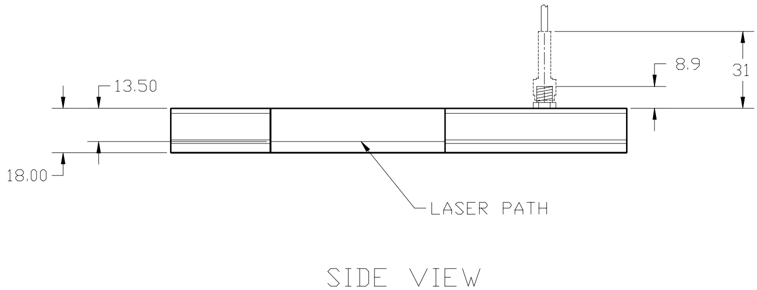
** Smaller parts possible, depending on application.

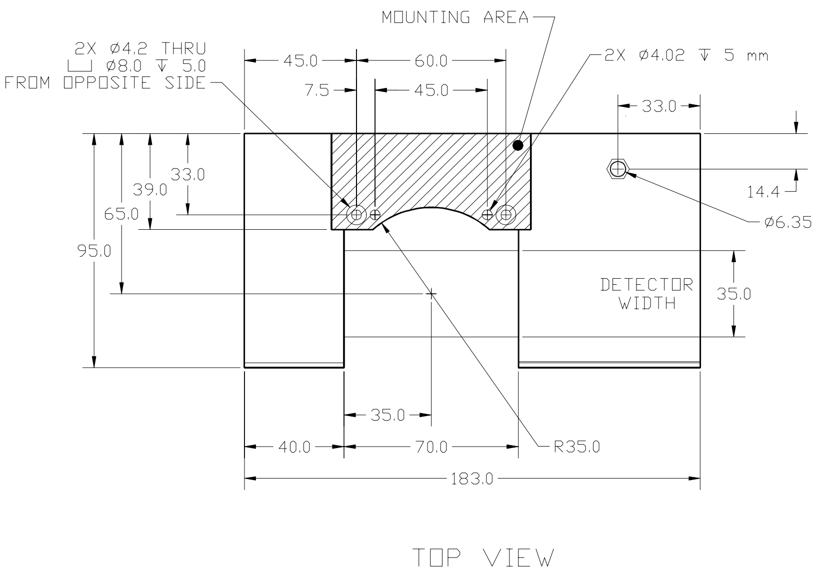
Sensor Dimensions

Low Profile Laser Align

All dimensions in Figure 14-10 are in millimeters. Drawings are not to scale.

Figure 14-10: Low Profile Laser Align Dimensions





Focused Modular Laser Align (Side Connector)

All dimensions in Figure 14-11 and Figure 14-12 are in millimeters. Drawings are not to scale.

Figure 14-11: Focused Modular Laser Align (Side Connector) Top View

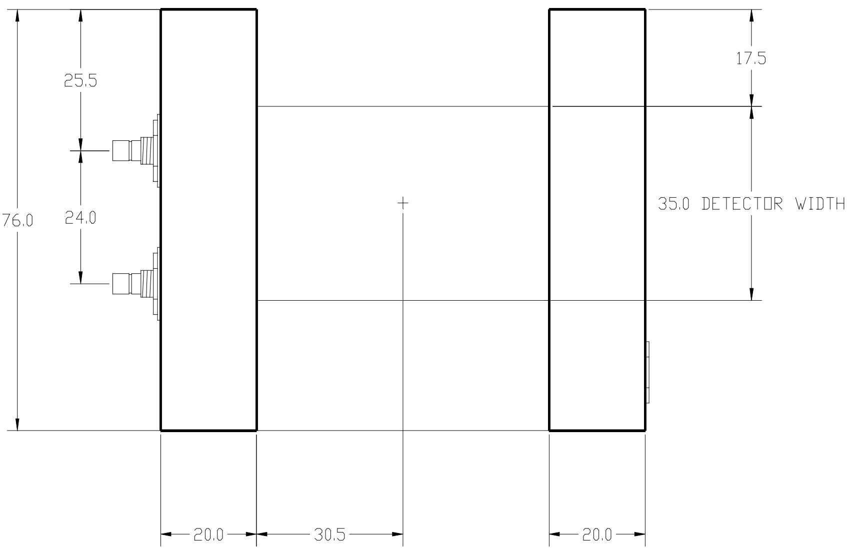
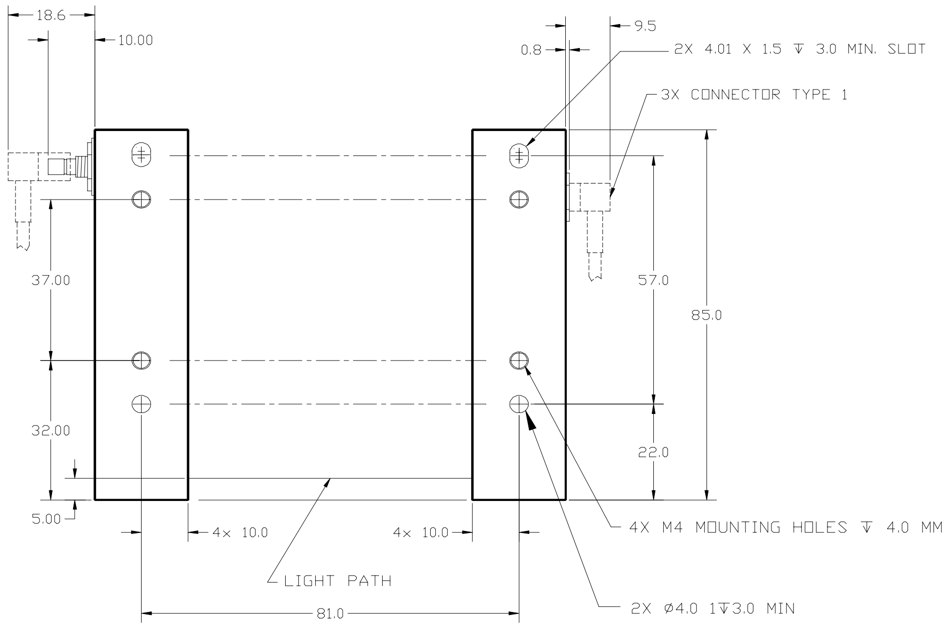


Figure 14-12: Focused Modular Laser Align (Side Connector) Side View



14

Focused Modular Laser Align (Top Connector)

All dimensions in Figure 14-13 and Figure 14-14 are in millimeters. Drawings are not to scale.

Figure 14-13: Focused Modular Laser Align (Top Connector) Top View

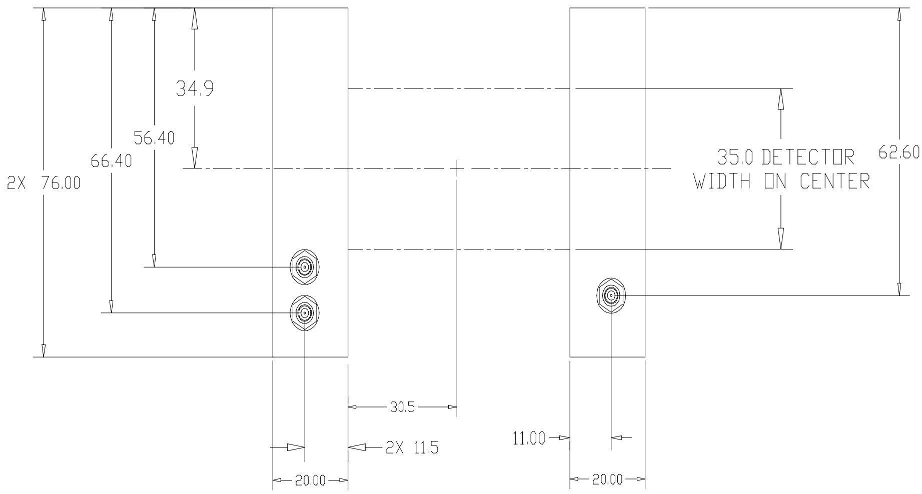
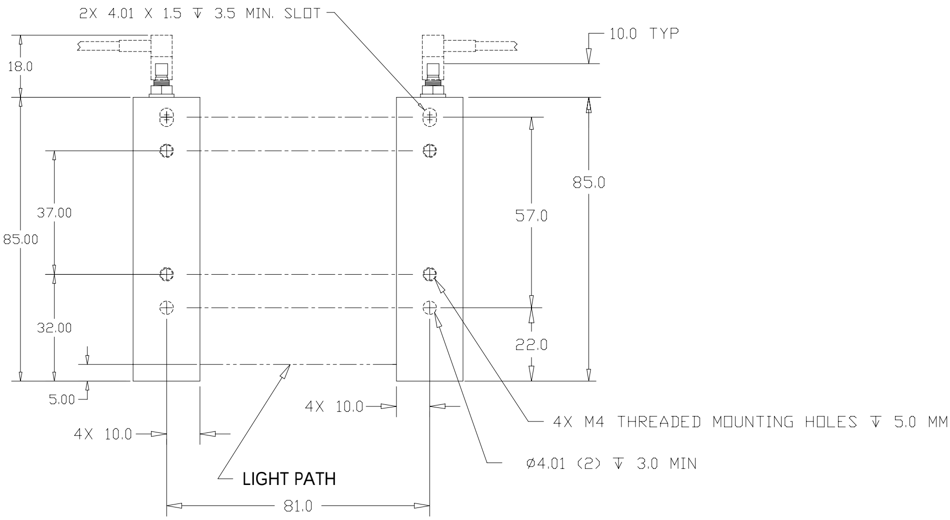


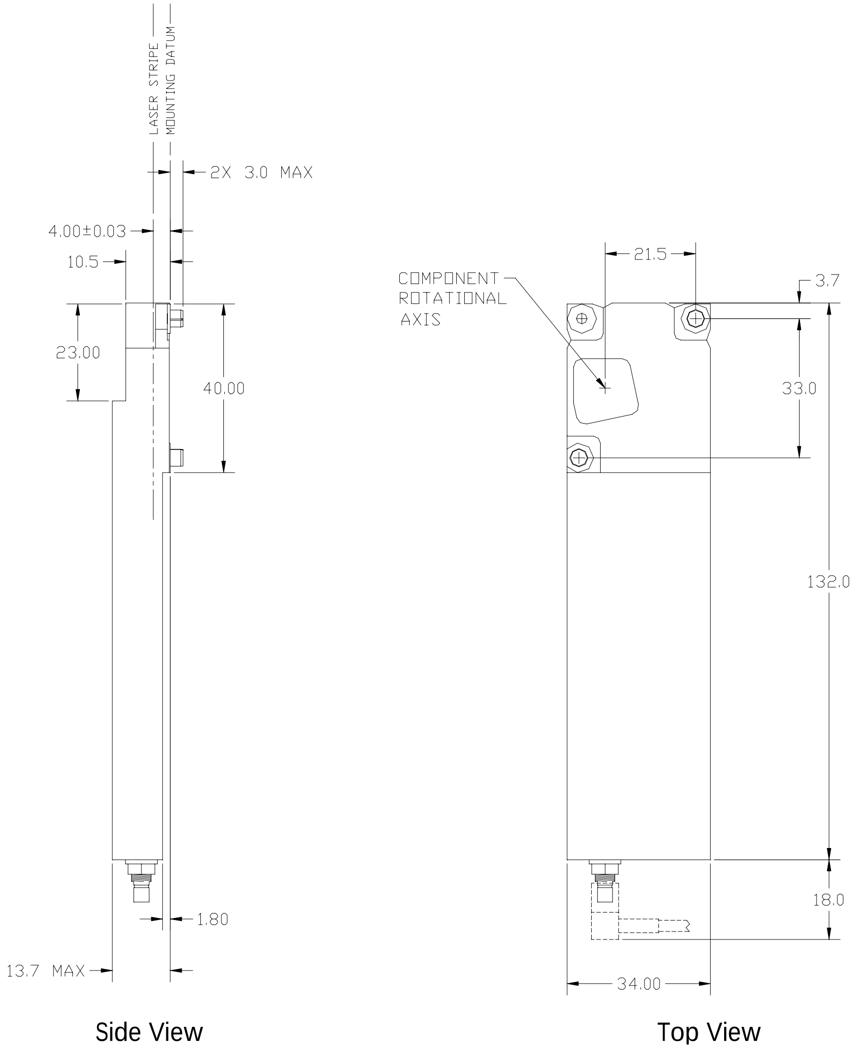
Figure 14-14: Focused Modular Laser Align (Top Connector) Side View



MiniLAM

All dimensions in Figure 14-15 are in millimeters. Drawings are not to scale.

Figure 14-15: MiniLAM Sensor Dimensions



High Density Laser Align

All dimensions in Figure 14-16 and Figure 14-17 are in millimeters. Drawings are not to scale.

Figure 14-16: High Density Laser Align Top View

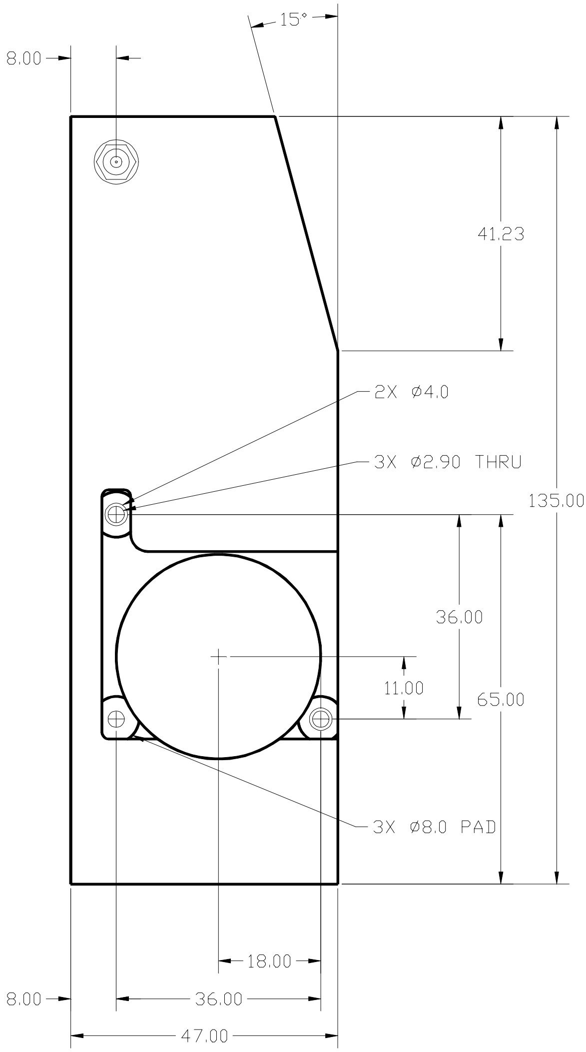
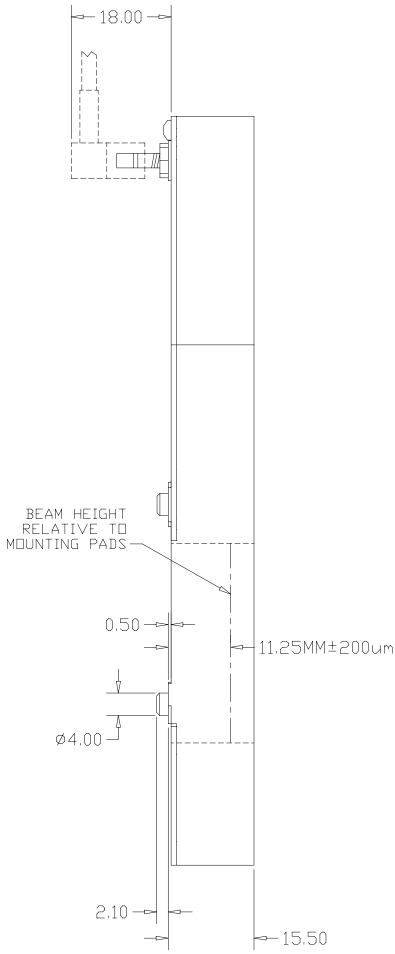


Figure 14-17: High Density Laser Align Side View



Sensor Mounting Specifications

Sensor mounting requirements vary, depending on sensor model and your application. Contact CyberOptics for detailed mounting specifications.

See *CyberOptics Service and Support* in Chapter 13 for more information.

Glossary

algorithm

see *measurement algorithm*

alignment angle

encoder position at which the component is aligned parallel to the Laser Align light stripe; one of the results reported by Laser Align is in response to a REPORTW_CMD (42).

angle limit

delay, in encoder counts, from the start of measurement before Laser Align stops looking for a minimum shadow width and reports an error. This is one of the measurement parameters that can be set using LOAD_CMD (40).

angular orientation

see *theta (θ) position*

C_{cal}

center position of the pickup nozzle in the Laser Align light stripe.

CCD array

charge-coupled detector array. This electronic device serves as the detector in the Laser Align sensor, identifying light intensity across its surface.

center

midpoint of the component shadow; one of the results reported by Laser Align in response to a REPORTW_CMD (42) issued after a sweep using measurement algorithm 11 or 14.

detector

see *CCD array*

edge

1) point where the component shadow on the Laser Align detector transitions from dark to illuminated. One of the results reported by Laser Align in response to a REPORTW_CMD (42) issued after a sweep using measurement algorithm 12 or 13.

2) boundary of a window

encoder

electronic device used in a motion control system to identify position.

encoder type

identification of an encoder based on its signal characteristics. See also *step and direction* and *quadrature.*

firmware

code that is stored in a hardware device. In Laser Align, the firmware resides on the sensor control module (SCM) and is used to control the hardware that communicates with the sensor and the host system. Component location and the alignment angle are calculated in the firmware.

flow control

method for regulating the transfer of data that involves using RTS/CTS (ready-to-send/clear-to-send) signals on a data link.

holdoff angle

delay, in encoder counts, between one Laser Align alignment angle and the start of the next alignment scan. This is one of the measurement parameters that can be set using LOAD_CMD (40).

host controller

operating system and hardware that controls the operation of the various components in a pick-and-place system. Laser Align is one of the controlled components.

image quality

measure of the usability of the light stripe information as it is detected by the CCD array; includes factors such as illumination intensity.

index pulse

signal generated by a hardware feature on the encoder that can be used to reset the Laser Align encoder counter.

lateral offset

calculated difference between the reported center of a component and C_{cal}. This information is used to make accurate part placements.

lateral position

component placement in the x- and y- axes.

LED

light emitting diode. In Laser Align, a series of these on the sensor control module (SCM) provides hardware status information.

maximum pickup uncertainty

the largest anticipated offset from a component's true center in X, Y, and θ at which a pickup nozzle will pick up the component.

measurement algorithm

identifies the alignment criterion and image processing technique to be used in generating the results at an alignment angle during a Laser Align sweep. This is one of the measurement parameters that can be set using LOAD_CMD (40).

measurement protocol

series of Laser Align commands used to designate the parameters to be used in performing measurements on a specific component type.

minimum

position in a rotation where the detected component shadow size increases after a consistent decrease. Laser Align collects results at the minimums in a component's rotation. There can be up to four minimums reported for a rotation.

offset

see *lateral offset* and *Z-offset*

parameter

value associated with a Laser Align firmware command that provides additional detail about command execution.

parameter index

value that identifies what aspect of Laser Align operation is affected by subsequent command parameters.

pickup nozzle

the part of a pick-and-place machine that lifts, holds, and positions the component being placed.

pickup error

the amount of offset from a component's true center in X, Y, and θ at which a pickup nozzle picked up the component. Laser Align results are used with information about nozzle position and component size to calculate this value.

prerotation

rotation of a component prior to performing a sweep to ensure a predictable starting position for sweep rotation.

quadrature

a type of electronic encoder signal characterized by two pulses that are out of phase.

RESULTS

data structure into which Laser Align results are collected, and in which results are reported in response to a REPORTW_CMD (42). RESULTS will always include values for status and angle. Depending on the commands issued prior to the REPORTW_CMD (42), and on the parameters sent with the REPORTW_CMD (42), RESULTS data may also include values for component center or edge, and for component width.

RS422 serial

communications protocol used between Laser Align and the host system.

scan

see *sweep*

SCM

see *sensor control module*

sensor control module

hardware component of the Laser Align system that processes sensor data and handles communication with the host system and the sensor.

shadow

dark region on the detector caused by an interruption of the light stripe.

status code

value returned by Laser Align that indicates condition of the system after execution of a command. Some Laser Align commands report status information as part of their automatic response. Others require the use of a REPORTW_CMD (42) to report status.

step and direction

a type of electronic encoder signal characterized by two signals where one signal contains pulses to be counted. The second signal provides information on whether to increment or decrement the counter when a pulse is observed on the first signal.

sweep

operation in which a component is rotated in the Laser Align light stripe, the sensor collects measurement information, and the information is processed by the SCM. Results are only reported if they are requested.

theta (θ) position

component orientation in the rotational axis.

tombstone

situation where a component is picked up in a way that causes it to be tilted out of a horizontal position.

units

unit of measure used for reporting Laser Align results; can be either microns or 1/10 pixels. Units must be set using Laser Align firmware commands every time the system is powered up or reset.

vacuum quill

see *pickup nozzle*

VALUE buffer

four-byte data store created using a LOAD_CMD (40) or a combination of POINTER_CMD (33) and VALUE_CMD (34). The value stored in this buffer is used to set the Laser Align encoder counter to a specific position by relaying the value from the buffer using a PRESET_CMD (35).

window

the area of the detector array used for measurement.

width

size of the component shadow from edge to edge, as measured at an alignment angle/shadow minimum position during a sweep. One of the results reported by Laser Align in response to a REPORTW_CMD (42).

window size

dimension, in pixels, of the measurement window. This is specified using LOAD_CMD (40) with appropriate parameters. The default value for window size is the entire illuminated area of the detector.

x-axis

lateral component position designation; may be width or length, depending upon the system configuration.

y-axis

lateral component position designation; may be width or length, depending upon the system configuration.

z-axis

vertical component position designation; height.

Z_{cal}

vertical position of the pickup nozzle when it is in the middle of the Laser Align light stripe.

Z-offset

difference between vertical position of the pickup nozzle required to attain desired component position in the Laser Align light stripe and Z_{cal}.

Index

D

E

F

G

H

I

N

O

P

Q

R

S

T