Varo Family
Force/Torque Sensors

Document #: 9610-05-1052

Table of Contents

1. Safety

The safety section describes general safety guidelines to be followed with this product, explanations of the notifications found in this manual, and safety precautions that apply to the product. Product specific notifications are embedded within the sections of this manual (where they apply).

1.1 Explanation of Notifications

These notifications are used in all of ATI manuals and are not specific to this product. The user should heed all notifications from the robot manufacturer and/or the manufacturers of other components used in the installation.

⚠	**DANGER:** Notification of information or instructions that if not followed will result in death or serious injury. The notification provides information about the nature of the hazardous situation, the consequences of not avoiding the hazard, and the method for avoiding the situation.

⚠	**WARNING:** Notification of information or instructions that if not followed could result in death or serious injury. The notification provides information about the nature of the hazardous situation, the consequences of not avoiding the hazard, and the method for avoiding the situation.

⚠	**CAUTION:** Notification of information or instructions that if not followed could result in moderate injury or will cause damage to equipment. The notification provides information about the nature of the hazardous situation, the consequences of not avoiding the hazard, and the method for avoiding the situation.

NOTICE: Notification of specific information or instructions about maintaining, operating, installing, or setting up the product that if not followed could result in damage to equipment. The notification can emphasize, but is not limited to: specific grease types, best operating practices, and maintenance tips.

1.2 General Safety Guidelines

The customer should verify that the transducer selected is rated for maximum loads and moments expected during operation. Refer to *Section 8—Specifications* or contact ATI Industrial Automation for assistance. Particular attention should be paid to dynamic loads caused by robot acceleration and deceleration. These forces can be many times the value of static forces in high acceleration or deceleration situations.

1.3   Safety Precautions

⚠️	**CAUTION:** Do not remove any fasteners or disassemble transducers without a removable mounting adapter plate. These include Nano, Mini, IP-rated, and some Omega transducers. This will cause irreparable damage to the transducer and void the warranty. Leave all fasteners in place and do not disassemble the transducer.
⚠️	**CAUTION:** Do not probe any openings in the transducer. This will damage the instrumentation.
⚠️	**CAUTION:** Do not exert excessive force on the transducer. The transducer is a sensitive instrument and can be damaged by applying force exceeding the single-axis overload values of the transducer and cause irreparable damage. Small Nano and Mini transducers can easily be overloaded during installation.

2. Product Overview

ATI's Varo Force/Torque (F/T) sensors measure six components of force and torque (F_x \ F_y \ F_z \ T_x \ T_y \ T_z) and communicate gage data to a user's device over an RS422 serial communication interface. The Varo family of sensors operate and function in the same manner. However, the Varo55 sensor has a smaller diameter, height, and strength compared to the Varo62. Refer to *Section 8—Specifications* for detailed differences between the Varo sensors.

The Varo sensors include the following features:

- Communication over two RS422 protocols: Modbus or console

- Small footprint ideal for sensors integrated into larger systems

- 8-Pin connector on robot mounting side

- 10 mm through hole for compact cable routing

Figure 2.1—Varo Sensor

Robot Side — 8-Pin RS422 and Power Connector — (6) Mounting Holes to Mount Sensor to Robot

Tool Side — Internal Through Hole — (6) Mounting Holes to Mount Sensor to Tooling
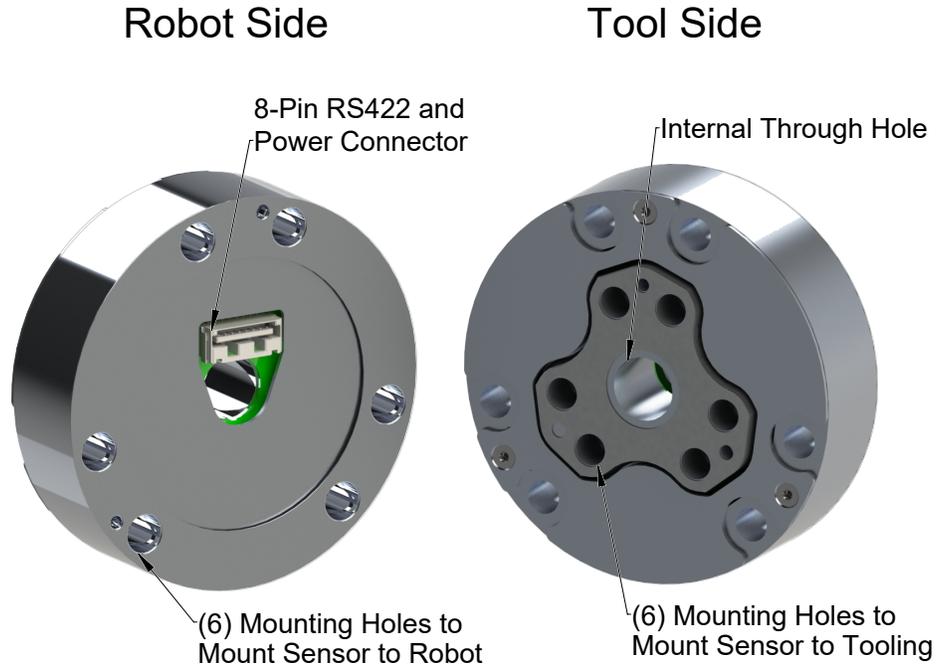
Table 2.1—Varo Sensor Family	
Part Number	**Sensor Name**
9105-SE1-VARO55-M40	VARO55
9105-SE1-VARO62-M120	VARO62

3. Installation

⚠	**WARNING:** Performing maintenance or repair on the sensor when circuits (for example: power, water, and air) are energized could result in death or serious injury. Discharge and verify all energized circuits are de-energized in accordance with the customer's safety practices and policies.

⚠	**CAUTION:** Using fasteners that exceed the customer interface depth penetrates the body of the sensor, damages the electronics, and voids the warranty. For more information, refer to the customer drawings.

⚠	**CAUTION:** Thread locker applied to fasteners must not be used more than once. Fasteners might become loose and cause equipment damage. Always apply new thread locker when reusing fasteners.

⚠	**CAUTION:** Avoid damage to the sensor from electrostatic discharge. Ensure proper grounding procedures are followed when handling the sensor or cables connected to the sensor. Failure to follow proper grounding procedures could damage the sensor.

⚠	**CAUTION:** Do not apply excessive force to the sensor and cable connector during installation, or damage will occur to the connectors. Align the keyway on the sensor and cable connector during installation to avoid applying excessive force to the connectors.

The following section outlines the procedures required to install ATI's Varo sensors to the customer's equipment.

3.1 Interface Plates

The sensor's mounting side attaches to the robot arm, and the sensor's tool side attaches to the customer tooling. For technical information on the sensor's mounting features, refer to *Section 9—Drawing*.

⚠	**CAUTION:** Incorrect installation of robot mounting and tool interface plates results in the failure of the F/T sensor to function properly.

⚠	**CAUTION:** The customer tool should only touch the tool interface plate. If the customer tool touches any other part of the sensor, it will not properly sense loads.

If the customer chooses to design and build an interface plate, consider the following points:

- The interface plate should include bolt holes for mounting fasteners as well as locating features for accurate positioning to the robot or customer's device.
- The thickness of the interface plate must provide sufficient thread engagement for the mounting fasteners.
- The mounting fasteners should not extend through the sensor's housing or interfere with the internal electronics.
- Do not use dowel pins that exceed length requirements and prevent the interface plate from mating flush with the robot and customer tooling. Fasteners that exceed length requirements create a gap between the interfacing surfaces and cause damage.
- The interface plate must not distort from the maximum force and torque values that can be applied to the sensor.
- The interface plate must provide a flat and parallel mounting surface for the sensor.

3.2   Installing the Sensor

There are two methods in which ATI's Varo sensors can be installed to the user's equipment: bolt-up installation (mounting from tool side) and bolt-down installation (mounting from robot side). Each of these methods are outlined below.

3.2.1   Bolt-Down Installation

Supplies required:

- *Loctite 242® threadlocker*
- *Loctite 7649® or equivalent threadlocker/primer*
- *Hex Key: 3 mm for Varo55, 4 mm for Varo62*
- *Torque Wrench*

Parts required:

- *(6x) Mounting Screws: (6x) M4 for Varo55; (6x) M5 for Varo62*
- *(2x) 2 mm Dowels*
- *Interface Plate*

Figure 3.1—Varo Bolt-Down Installation

(6) M5 or M4 Mounting Screws (Customer Supplied)

(2) 2 mm Dowels (Customer Supplied)

Tiered Interface Plate To Avoid Mounting Screws Interference (Customer Supplied)

1. Attach customer-supplied cable to 8-pin connector on sensor. Refer to *Section 3.4—Sensor Connector* for detailed pin-out information on the connector.

2. Mount the interface plate to the customer equipment.

3. Pre apply (6) mounting screws with Loctite 7649 and then Loctite 242.

4. Align the sensor to interface plate using (2) 2 mm customer-supplied dowels.

5. Use hex key to mount sensor to the interface plate.

6. Use torque wrench to fasten screws to torque specified in *Table 3.1*.

Table 3.1—Varo Torque Values		
Sensor	**Bolt Type**	**Torque**
Varo55	M4	41 in-lb
Varo62	M5	89 in-lb

6. After initial installation, run accuracy check procedure. Refer to *Section 7.3—Accuracy Check Procedure*
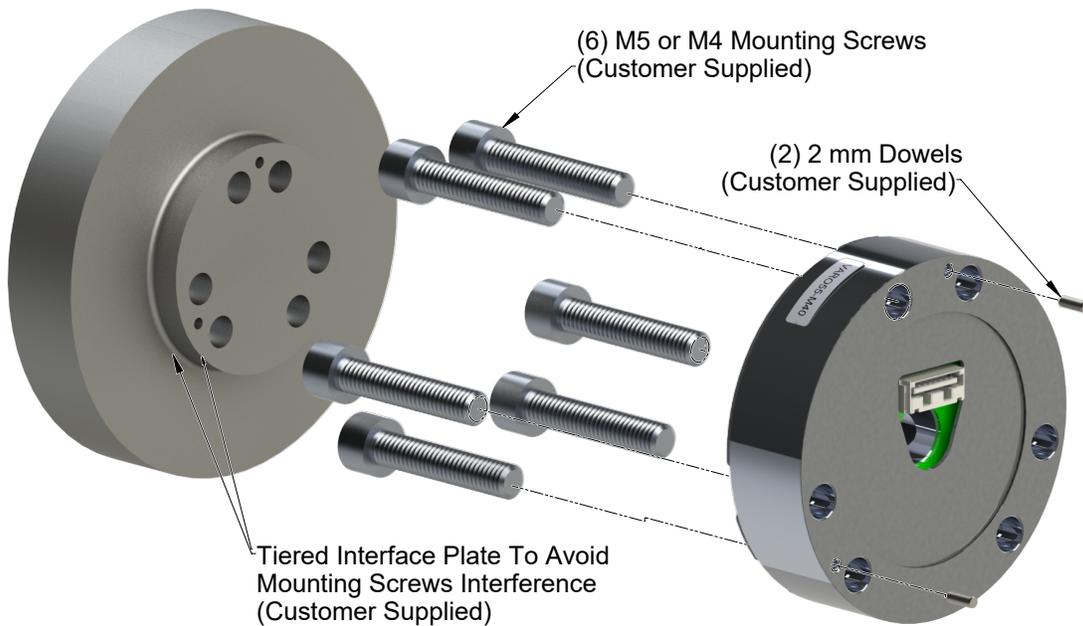
3.2.2 Bolt-Up Installation

Supplies required:

- *Loctite 242® threadlocker*
- *Loctite 7649® or equivalent threadlocker/primer*
- *Hex Key: 4 mm for Varo55, 5 mm for Varo62*
- *Torque Wrench*

Parts required:

- *(6x) Mounting Screws: (6x) M5 for Varo55; (6x) M6 for Varo62*
- *(2x) 2 mm Dowels*

Figure 3.2—Varo Bolt-Up Installation
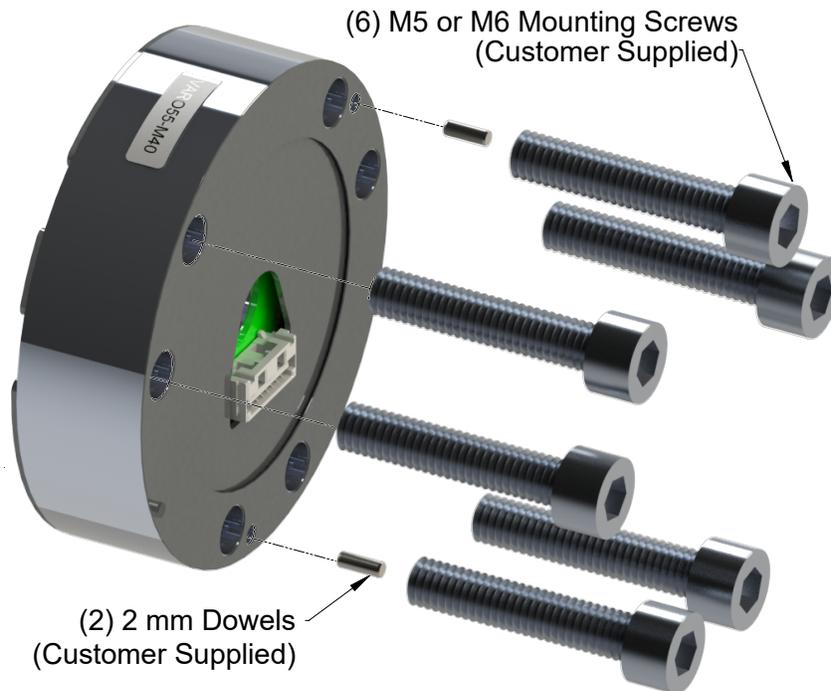
1. Attach customer-supplied cable to 8-pin connector on sensor. Refer to *Section 3.4—Sensor Connector* for detailed pin-out information on the connector.

2. Pre apply (6) mounting screws with Loctite 7649 and then Loctite 242.

3. Align the sensor to customer equipment using (2) 2 mm customer-supplied dowels.

4. Use hex key to mount sensor to customer equipment.

5. Use torque wrench to fasten screws to torque specified in *Table 3.2.*

Table 3.2—Varo Sensor Torque Values		
Sensor	**Bolt Type**	**Torque**
Varo55	M5	65 in-lb
Varo62	M6	108 in-lb

6. After initial installation, run accuracy check procedure. Refer to *Section 7.3—Accuracy Check Procedure*

3.3 Tooling Installation

Supplies required:

- *Loctite 242® threadlocker*
- *Loctite 7649® or equivalent threadlocker/primer*
- *Hex Key: 3 mm for Varo55, 4 mm for Varo62*
- *Torque Wrench*

Parts required:

- *(6x) Mounting Screws: M4 for Varo55; M5 for Varo62*
- *(2x) 2 mm Dowels*
- *Tooling Adapter Plate*

Figure 3.3—Tooling Installation
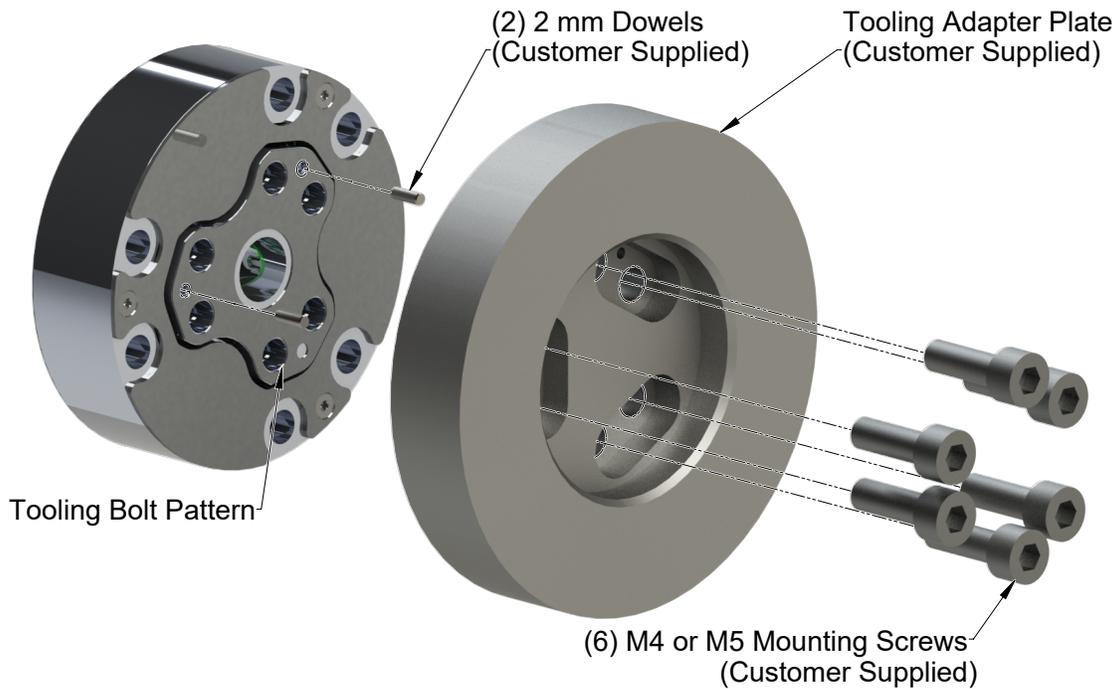
1. Pre apply (6) mounting screws with Locktite 7649 and then Loctite 242.

2. Align the tooling adapter plate to the sensor using (2) 2 mm customer-supplied dowels.

3. Use hex key to tooling adapter plate to sensor..

4. Use torque wrench to fasten screws to torque specified in *Table 3.3*.

Table 3.3—Varo Tooling Torque Values		
Sensor	**Bolt Type**	**Torque**
Varo55	M4	41 in-lb
Varo62	M5	89 in-lb

3.4 Sensor Connector

⚠	**CAUTION:** Ensure the cable shield is properly grounded. Improper shielding on the cables can cause communication errors and an inoperative sensor.

⚠	**CAUTION:** Subjecting the connector to repetitive motion will cause damage to the connector. Restrain the cable close to the connector so that the repetitive motion of the robot does not interfere with the cable connector.

⚠	**CAUTION:** Improper cable routing may cause injury to personnel, poor functionality of critical electrical lines, or damage to the equipment. The electrical line, especially where attached to the senor's connector, must be routed to avoid stress failure, sharp bends, or a disconnection from the equipment. Damage to the sensor from improper routing will void the warranty.

The sensor has an 8-pin latching surface connector for power to the sensor and data transfer. The user must supply their own RS422 cable. A suitable mating connector is Molex 5023800800. Refer to the *Figure 3.4* and *Table 3.4* for the connector's pinout.

Figure 3.4—Connector for User Interface

10mm Sensor Through Hole

Pin 1   Pin 8
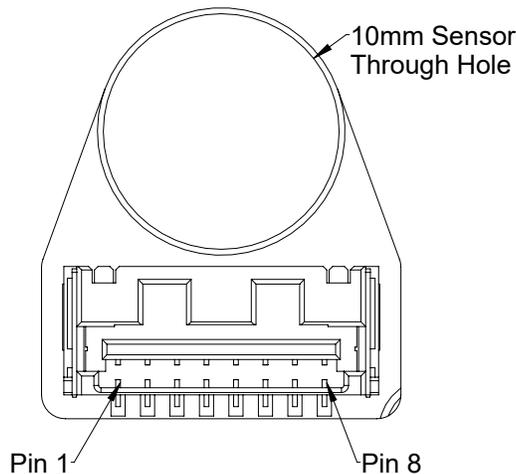
Table 3.4—Connector's Pinout		
Pin Number (#)	**Signal**	**Function**
1	VIN+	Positive supply voltage (5 V nominal)
2	VIN-	Negative supply voltage (ground)
3	RX_P	RS422 received data positive
4	RX_N	RS422 received data negative
5	TX_P	RS422 transmitted data positive
6	TX_N	RS422 transmitted data negative
7	CLK_SYNC	Clock Sync functionality
8	CHASSIS	Sensor body ground
Note: A suitable mating connector to the 8-pin latching surface connector is Molex 5023800800.		

4. Operation

> ⚠️ **CAUTION:** Varo sensors are specified to operate within an environment between 0 and 70 degrees Celsius, and up to 95 percent humidity, non-condensing. Failure to maintain environmental conditions within the appropriate range may damage the sensor and void its functionality. See *Section 8.2—Environmental Specifications.*

> **NOTICE:** Sensors may react to exceptionally strong and changing electromagnetic fields, such as those fields created by magnetic resonance imaging (MRI) machines.

The ATI Varo sensors support two modes of communication over the RS422 protocol:

- *Console Commands*
- *Modbus Interface*

Varo sensors will load in console mode by default, but can be configured by the user (refer to *Section 5.5—SET Command*). Each mode of communication allows the user to:

- Stream continuous or single lines of force and torque values in counts
- View or edit sample rate, RDT rate, and baud rate
- Monitor the health of the sensor
- Print sensor-related temperature data

4.1 Sample Rate

By default, the firmware samples internally at 1000 Hz. This rate can be changed by using the `SET adcRate` command: refer to *Section 5.5—SET Command* and *Table 5.3*.

4.2 Baud Rate

Data transmits at a serial baud rate of 3 megabaud by default. To change the baud rate, use the `SET baud` command: refer to *Section 5.5—SET Command* and *Table 5.3*.

4.3 Raw Data Output

The sensor supplies compensated gage data to the end user. To convert this gage data into force and torque data, complete the steps in the following section:

4.3.1 To Read Calibrated F/T Data

The sensor outputs gage data in counts. Therefore, the customer must convert the raw data to force/torque units.

1. Set the customer-supplied equipment to use the correct UART baud rate.
   - The UART port runs at 3 megabaud by default.
   - To configure the baud rate, use the `set baud` command, as referenced in *Table 5.3.*
2. Read the calibration information to create the active calibration matrix, which is shown in *Figure 4.1*.
3. Send the `STREAM` command: refer to *Section 5.8—STREAM Command*.
4. When each streaming sample is received, check the status register (*Table 7.1*) for all zeros, which would indicate system health. Also check the checksum for sample validity.

5. Bias, or tare, the sensor:

    a. Take a reading when the sensor is in its default or unloaded state.

    b. Save these gage values as the "bias vector". Refer to *Figure 4.1*.

    c. For each subsequent sample, subtract the bias vector from the sampled strain gage vector to remove the bias from the calculated F/T data.

6. Multiply the active calibration matrix by strain gage vector (with the bias vector subtracted) to calculate the F/T values. *Figure 4.1* outlines this equation visually.

- The resulting vector contains the scaled force and torque values in N and Nm (or other units per the calibration certificate). Reference *Section 4.3.1.1—Sample F/T Calibration Matrix Calculation* for an example calculation.

Figure 4.1—F/T Calibration Matrix

$$
\begin{bmatrix}
mat00 & mat01 & mat02 & mat03 & mat04 & mat05 \\
mat10 & mat11 & mat12 & mat13 & mat14 & mat15 \\
mat20 & mat21 & mat22 & mat23 & mat24 & mat25 \\
mat30 & mat31 & mat32 & mat33 & mat34 & mat35 \\
mat40 & mat41 & mat42 & mat43 & mat44 & mat45 \\
mat50 & mat51 & mat52 & mat53 & mat54 & mat55
\end{bmatrix}
\begin{bmatrix}
G0 \\ G1 \\ G2 \\ G3 \\ G4 \\ G5
\end{bmatrix}
=
\begin{bmatrix}
Fx \\ Fy \\ Fz \\ Tx \\ Ty \\ Tz
\end{bmatrix}
$$

Where:

    matXX is the sensor calibration matrix, *G0* to *G5* is the biased strain gage vector,

    and *Fx*, *Fy*, *Fz*, *Tx*, *Ty*, and *Tz* are the resulting scaled force and torque values in N and N m (or other units per the calibration certificate).

7. To stop streaming data, send the `CONSOLE` command: refer to *Section 5.6—CONSOLE Command*.

4.3.1.1 Sample F/T Calibration Matrix Calculation

The sample calibration matrix calculation in the following figure uses example data to calculate the calibration.

Figure 4.2—Sample Calculation

$$
\begin{bmatrix}
-1.948E-05 & 1.924E-05 & 3.705E-08 & 1.643E-07 & 1.871E-05 & -1.870E-05 \\
1.031E-05 & -1.028E-05 & -2.204E-05 & 2.206E-05 & 9.642E-06 & -9.703E-06 \\
-8.942E-06 & -1.136E-05 & -9.256E-06 & -1.086E-05 & -9.197E-06 & -1.056E-05 \\
2.018E-07 & 1.879E-07 & -6.692E-08 & 6.520E-08 & -1.505E-07 & -2.379E-07 \\
-4.426E-08 & -1.875E-07 & 2.098E-07 & 2.486E-07 & -1.605E-07 & -6.624E-08 \\
-3.433E-07 & 3.353E-07 & -3.864E-07 & 3.843E-07 & -3.270E-07 & 3.375E-07
\end{bmatrix}
\begin{bmatrix}
-2182310 \\ -125985 \\ 2016149 \\ 2042713 \\ 108226 \\ -2008978
\end{bmatrix}
=
\begin{bmatrix}
80.09 \\ -0.04 \\ 0.33 \\ -0.004 \\ 1.167 \\ 0.000
\end{bmatrix}
$$

5. Console Commands

These commands can be used to view the status, parameters, and adjust settings of the sensor.

Table 5.1—Console Commands		
Command	**Command with Operand(s)**	**Description**
HELP H MAN	N/A	Prints main commands and the software version. Reference *Section 5.1—HELP Command.*
C	012345678CUSHDB<>#@!?	Continuous mode. Sensor will report continuous lines of data until another key is pressed. Reference *Section 5.3—S or C Query Commands.*
S		Single mode. The sensor will report a single line of data. Reference *Section 5.3—S or C Query Commands.*
SET	set	Prints all sensor's settings or stored parameters. Reference *Section 5.5—SET Command.*
	set [field]	Prints all fields matching the operand.
	set [field] [value]	Sets specific setting field with value entered. Reference *Section 5.5—SET Command.*
RESET	reset	Restarts the MCU's program. Reference *Section 5.2—RESET Sensor Command.*
	reset delay	Adds a delay of up to 5000 miliseconds before reset command is completed.
SAVEALL	N/A	Writes all values that remain through a power cycle to non-volatile memory (NVM). Reference *Section 5.4—SAVEALL Command.*
CONSOLE	N/A	Switches a serial link from Modbus to console mode. Console mode is on by default on initial start-up. This command can also be used to exit a streaming mode. Reference *Section 5.6—CONSOLE Command.*
MODBUS	N/A	Switches serial link from console mode to Modbus. Reference *Section 5.7—MODBUS Command.*
STREAM	N/A	Outputs compensated F/T data. Reference *Section 5.8—STREAM Command.*

5.1 HELP Command

The HELP command reports a list of the main commands and software version.

H command format:

user: H

response:

```
======================================================================
9031-05-1086 Transducer Version: 1.0.0+gb92a02c8f => BL Version: 5
=> Enter most commands without operands to display current status.
==============================================================================
HELP    [string] => Print help for commands that start with the given string
RESET   [delay] => Reset the MCU, optional delay in ms
SET     => Print all fields
|       [field-name] => Print matching field(s)
|       [field-name] [value] => Write field with value
SAVEALL => Save all parameters to NVM
C       012345678UCHDB<>S#@!; in any order => Continuous mode
|       0->8=ADC channels
|       U=Temperature compensated & linearized data
|       C=Raw ADC counts
|       H=Hex D=Decimal B=Binary '>'=condensed '<'=formatted
|       S=checkSum #=LineCounter @=ADC_SampleCounter !=StatusWord
|       ;=Comma Delimiter ^=Sync char -- Press multiple keys to exit
S       => Single Sample, same format as Continuous mode
MODBUS  => Change to MODBUS Server
CONSOLE => Change to Console
STREAM  => Start binary streaming

======================================================================
```

5.2   RESET Sensor Command

The RESET command restarts the MCU's program. This command can be used with the operand delay, which waits no more than 5000 miliseconds to complete the RESET command.

5.3   S or C Query Commands

The query command starts the high-speed data transmission of data. The S command reports a single line of the requested data. The C command reports continuous lines of the requested data and stops when the user holds another key, for example the Enter key, until the output of data ceases. The C command reports data at the rate specified in the rdtRate, except when the speed is restricted by the serial port rate or by compute time.

Users can adjust how the data is reported by issuing a query command with one or more specifiers: refer to *Table 5.2*. If a S or C command is issued without a specifier(s), the specifier(s) from the previous S or C command is used in the data print out. The power-on default specifier is 01234567.

Query S command format:

```
user:        S
response:     >
001cbe 0017a6 001f03 0018a3 001d70 001693 00208e 001583
```

Query C command format:

```
user:        C
response:     >
001c55 001791 001e87 001879 001dd0 0015fe 0020d9 00164c
001cd4 001735 001e3b 001857 001d41 0015fc 002079 001631
001ca4 001750 001e09 001864 001d71 0016ab 002020 001598
001c82 00177a 001eaa 0018c6 001d93 001646 0020ab 00154c
001bf4 00173e 001e98 0017fb 001e06 001658 002068 0015e4
             ...
          ...
          ...
```

user: <presses and holds another key, such as the Enter key, to stop the data transmission >

No return data.

5.3.1 Secondary Commands for the Query C or S Command

The type of data reported from the query C or S command can be adjusted using secondary commands or specifiers. This feature is useful for users who want to develop their own program for storing the data to an external file or view the data in figures such as charts. A list of secondary commands is found in *Table 5.2*.

Table 5.2—Secondary Commands for S or C		
Category	**Secondary Command or Specifier**	**Description**
Gage number(s)	0	Gage values are printed in counts only. As many as all gage numbers can be reported or as few as a single gage number.
	1	
	2	
	3	
	4	
	5	
Temperature	6	Tmap is temperature of the mounted side of the sensor. Temperature values are printed in degrees C * 10.
	7	Ttap is the temperature of the tool flange side of the sensor. Temperature values are printed in degrees C * 10.
	8	Tpcb is the temperature of PC Board (Tpcb). Temperature values are printed in degrees C * 10.
Data Type	C	Raw Gage Counts (for troubleshooting only)
	U	Compensated Gage Data (Default)
Numeric System	H	The data is displayed as a hexadecimal number. Hexadecimal is the default setting.
	B	The data is displayed in binary in raw bytes that are not human readable. There are no field separator characters. The number of binary bytes printed will vary as follows:

Field	**Specifier**	**Field Length (Bits)**
Sync character	^	8
Sequence character	#	16
CRC	s	16
ADC data	0 to 8	24
Status word	!	32
ADC tick	@	32

	D	The data is displayed as a decimal number.
Format	<	The data is displayed in a formatted human-readable output. This is the default setting.
	>	The data is displayed in a compressed output that has no leading zeros, trailing zeros, or unnecessary blanks. This output is intended for high-speed applications that are used in an automated setting.
	;	This switches field separator from blank to a comma.

Table 5.2—Secondary Commands for S or C		
Category	**Secondary Command or Specifier**	**Description**
Additional inputs to aid in the development of a customer software program	S	This command specifies a CRC. A standard CRC 16-bit is used. Refer to *Section 7.4—CRC 16-Bit Source Code*
	#	This command specifies a sample counter that is incremented each time that a c or s line is printed.
	@	This command specifies an ADC read counter that is incremented each time that the ADC is read.
	^	This command prints sync value. 23 in decimal, 0x17 in hex, one-byte character 0x17 in binary. The sync value will display where it is inserted, but is typically used at the beginning of each command.
Troubleshooting	!	This command specifies the 32-bit status code. Refer to *Section 7.1—Status Register*

5.3.2 Examples of Secondary Commands (Specifiers)

The following are examples of an S or C command with specifiers:

1. C 0123 is interpreted as

    user: C 0123

    response: 9e6c1 95a9 963   96cb0

    a. The C is a command for reporting continuous lines of data.

    b. A number 0 through 5 specifies to print the data for the corresponding gage number. For example, the 0 specifies to print data for gage 0, and the 3 specifies to print data for gage 3.

    c. Data is displayed in counts by default.

    d. Data is displayed in hexadecimal by default.

2. >s >012345d678

    user: >s >012345d678

    response: >15c2 1d5c 1dca 1c13 1d03 1d86 217 249 272

    a. The S is a command for reporting a single line of data.

    b. The > specifies the data will be displayed in a compressed output, intended for high-speed applications.

    c. The 012345 specifies to print the data for the corresponding gage number.

    d. The D specifies any data following this operand will display in decimal format.

    e. The 678 specifies to print the Tmap, Ttap, and Tpcb values. Temperature values display in degrees Celsius multiplied by 10. So, the value of 272 would equate to 27.2 degrees C.

3. >c b^012345678s

    user: >c b^012345678s

    response:

    Offset(d) 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
    00000000 17 00 14 FE 00 1D AD 00 1C 98 00 1C 72 00 1D 06 00 1C CC 00 00 D9 00 00 F9 00 01 12 3C B8

    a. The C is a command for printing continuous lines of data.

    b. The B specifies to print the data in binary.

    c. The ^ specifies a sync value will be placed.

    d. The 012345 specifies to print the data for the corresponding gage number.

    e. The 678 specifies to print the Tmap, Ttap, and Tpcb values.

    f. The s specifies to print the 16-bit CRC. Refer to *Section 7.4—CRC 16-Bit Source Code*

5.3.3 Clock Sync Functionality

Clock sync functionality activates when the user applies a rising edge of at least 5 V to the CLK_ SYNC input, which is pin 7 on the sensor connector (reference *Figure 3.4)*.

Upon activation of the sync function, the sensor outputs the most recently collected data point which is equivalent to the output from the "s" command (*Section 5.3—S or C Query Commands)* sent over an RS422 cable interface. Data output speeds can be matched up to 2 kHz.

An electrical pulse is shown in the following figure. The rising edge of the pulse starts at 5 V. The falling edge of the pulse is when the voltage is no longer within 5-30 V. 6 V is the maximum voltage that the cable allows. The sync functionality is no longer activated when the voltage is outside the 5-6 V range.

Figure 5.1—Electrical Pulse
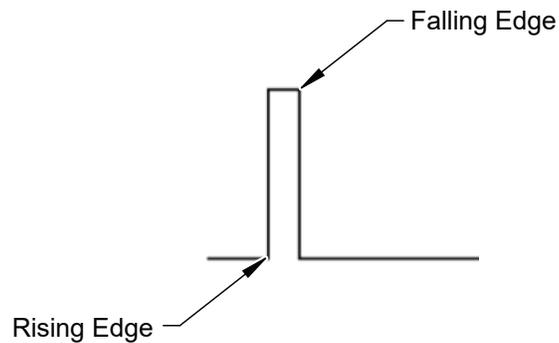
5.4 SAVEALL Command

The `SAVEALL` command writes all settings to non-volatile memory (NVM). This is typically used if any settings have been updated.

`SAVEALL` command format:

```
user:        saveall

response:    Parameters saved to NVM bank 0

             Parameters saved to NVM bank 1
```

5.5 SET Command

The `SET` command reports all settings. Note that `CAL` is synonymous for `SET`, but in this manual, the command is referenced to as `SET`. Many settings are read-only fields that are configured onto the sensor during ATI factory calibration. All setting fields are listed in *Table 5.3*. All write commands are temporary until a `SAVEALL` command is issued. When a `SAVEALL` command is given, the parameter is stored in non-volatile memory (NVM).

`SET` command format, for example:

```
user:        set

response:    Field                       Value

             -----                       -----

             serialNum                   FT22835

             partNum                     SI-150-8

             calFamily                   RS422

             ...
```

To read a stored parameter in NVM for a field from *Table 5.3*, type `SET` [field], for example:

```
user:        set adcrate

response:    Field                       Value

             -----                       -----

             adcrate                     1000
```

Table 5.3—`SET` Fields				
Field	**Read/Write**	**Example**	**Type**	**Description**
serialNum	Read	`FT33859`	STRING(8)	The FT serial number.
partNum	Read	`SI-150-8`	STRING(30)	The calibration part number.
calFamily	Read	`SE1`	STRING(4)	Communication type
calTime	Read	`9/21/2021 00:00`	STRING(20)	The date and time the sensor was calibrated.
boardSerNum	Read	`Board-1`	STRING(10)	Board Serial Number.
forceUnits	Read	`1`	8-bit unsigned integer	Designates the unit of the output data: 1: N for force 2: Nm for torque.
torqueUnits	Read	`2`		

Table 5.3—SET Fields				
Field	**Read/Write**	**Example**	**Type**	**Description**
mat00		`-8.068078e-04`		
mat01		`7.229595e-04`		
mat02		`9.738127e-05`		
mat03		`8.846054e-05`		
mat04		`7.276585e-04`		
mat05		`-8.419771e-04`		
mat10		`3.544661e-04`		
mat11		`-5.250728e-04`		
mat12		`-9.002674e-04`		
mat13		`9.197624e-04`		
mat14		`5.349651e-04`		
mat15		`-3.920001e-04`		
mat20		`-6.579991e-04`		
mat21		`-6.277593e-04`		
mat22		`-6.513078e-04`		These 36 total elements are broken up in a 6x6 matrix, which is used for force/torque calculation.
mat23		`-6.416455e-04`		
mat24		`-6.442259e-04`		
mat25	Read	`-6.612773e-04`	float	
mat30		`1.698986e-05`		
mat31		`9.682020e-06`		These calibration values are stored in the sensor in order to tie the calibration to the sensor.
mat32		`-6.898061e-06`		
mat33		`7.349188e-06`		
mat34		`-9.875973e-06`		
mat35		`-1.686716e-05`		
mat40		`-1.877622e-06`		
mat41		`-1.348794e-05`		
mat42		`1.568340e-05`		
mat43		`1.505260e-05`		
mat44		`-1.421625e-05`		
mat45		`-1.153442e-06`		
mat50		`-1.150356e-05`		
mat51		`1.167877e-05`		
mat52		`-1.271140e-05`		
mat53		`1.329944e-05`		
mat54		`-1.183128e-05`		
mat55		`1.231831e-05`		

Table 5.3—SET Fields				
Field	**Read/Write**	**Example**	**Type**	**Description**
gageMax0	Read	111369318	32-bit unsigned integer	The maximum rated value for this axis in gage counts.
gageMax1				
gageMax2				
gageMax3				
gageMax4				
gageMax5				
gageMax6				Internal thermistor. This does not have a maximum value, but is helpful to reference to confirm thermistor's functionality.
gageMax7				
gageMin0	Read	–93699427	float	The minimum rated value for this axis in gage counts.
gageMin1				
gageMin2				
gageMin3				
gageMin4				
gageMin5				
gageMin6				Internal thermistor. This does not have a minimum value, but is helpful to reference to confirm thermistor's functionality.
gageMin7				
adcRate	Read and Write	1000	16-bit unsigned integer	The ADC update rate in Hz. Any value entered is set to the closest value: 500, 1000, or 2000.
rdtRate[1]	Read and Write	40	16-bit unsigned integer	The RDT transmission rate in units of Hz. rdtRate must be an integer between 1 and 2000.
baud	Read and Write	3000000 (default)	32-bit unsigned integer	UART baud rate. Must be in range from 300 baud to 3 megabaud. Any baud rate change is temporary until a SAVEALL command is issued.
syncType	Read and Write	0 (default)	8-bit unsigned integer	On Clock Sync Falling Edge: 0: same as "s" command 1: one Modbus streaming packet. See *Section 6—Modbus Interface*
modbus	Read and Write	0 (default)		0: Start in console mode 1: Start in Modbus mode

Note:
1. The actual rate may be slower if the ADC sampling rate is less than the rdtRate or if the serial port cannot transmit all of the data at the rdtRate.

5.6 CONSOLE Command

When connecting through a serial link, the Varo sensors will boot in console mode by default. If the user toggles to Modbus mode, entering a `CONSOLE` command will switch back to console mode.

The `CONSOLE` command can also be used to exit streaming mode by typing `CONSOLE` in the command line.

> **NOTICE:** If using a console software, such as PuTTY, turn off any console features that completes commands according to data from the serial port.

5.7 MODBUS Command

Varo sensors boot automatically in console mode, by default. In order to switch to Modbus mode, the user can enter the `MODBUS` command. At this point, the only console command that will be recognized is `CONSOLE`.

Refer to *Section 6—Modbus Interface* for detailed Modbus communication information.

5.8 STREAM Command

The `STREAM` command starts outputting compensated data in binary packets. These data packets do not appear as recognizable characters in the console software. Refer to *Section 6.2—Data Packet Structure* for more detail.

To exit the streaming mode, use the `CONSOLE` command.

6. Modbus Interface

The user should have prior understanding of Modbus protocol and computing operations. The Modbus interface follows the timing and checksum rules specified in the *Modbus Over Serial Line* specification available online at *http://www.modbus.org/specs.php*.

To enter Modbus mode, enter a `MODBUS` command. The analog board runs as a Modbus slave until the 'start streaming' custom Modbus command is sent. The board continues to stream data until the 'stop streaming' custom Modbus command is sent.

Users who are writing a custom Modbus interface to the sensor must ensure the interface supports the following:

- "Read Holding Registers" command (*Section 6.3* in the *Modbus Serial Line Protocol and Implementation Guide v1.02*)
- One or both of the "Write Single Register" (*Section 6.6*) and "Write Multiple Registers" (*Section 6.12*) commands
- Ability to send custom (user-defined) function codes

6.1 Data Structure

6.1.1 Endianness

The sensor stores all numbers in big-endian (aka "Network Order") format. If your host system uses a little-endian ("Byte-swapped") format, any custom software you write will have to swap the bytes in each numeric field before performing any calculations using those values.

6.1.2 Data Types

The different data types used are defined as:

Table 6.1—Data Types	
Term	**Definition**
float	IEEE 754-2008 binary32 (aka 'single') floating point number
int16	16-bit signed integer
int24	24-bit signed integer
int32	32-bit signed integer
uint8	8-bit unsigned integer
uint16	16-bit unsigned integer

6.2 Data Packet Structure

Each serial Modbus packet or Proctocol Data Unit (PDU) has the following structure per *Section 2.3* of the *Modbus Serial Line Protocol and Implementation Guide v1.02*):

Table 6.2—Data Packet Structure				
Column 1	**Column 2**	**Column 3**	**Column 4**	**Column 5**
Address Field (Always 10 for the F/T sensor)	**Function Code** (refer to *Table 6.4*)	**Data** (may be up to 252 bytes and contain a structure with multiple fields)	**CRC High Byte**	**CRC Low Byte**

The sensor uses a fixed Modbus slave address of 10, and only communicates using the binary "RTU" serial Modbus packet format. The CRC calculation for the binary RTU mode, including sample source code, is described in *Section 6.2.2* of the *Modbus Serial Line Protocol and Implementation Guide v1.02*.

6.3 Custom (User-Defined) Service Codes

Each custom service code uses the same serial Modbus command and response format:

Table 6.3—Service Code Command and Response Format				
Column 1	**Column 2**	**Column 3**	**Column 4**	**Column 5**
Slave Address (10)	Function Code Command[1] Response[2]	Data	CRC High Byte	CRC Low Byte
Notes:				
1. The function code to execute.				
2. The function code executed (same as command).				

The sensor has the following custom, user-defined codes:

Table 6.4—Custom (User-Defined) Service Codes			
Code	**Description**	**Data**	**Response**
70	Start strain gage streaming	1 data byte with value 0xAA	A Modbus packet with one data byte with a value "1" for successful reception, or a standard Modbus error packet if unsuccessful. After successful reception of this command, the sensor starts data streaming mode where a packet in Streaming Sample Format (*Section 6.5—Sample Streaming Data Format*) is sent out after every ADC read.
71	Stop strain gage streaming	1 data byte with value 0xAA	A Modbus packet with one data byte with a value "1" for successful reception, or a standard Modbus error packet if unsuccessful. After successful reception of a command to stop strain gage streaming, the sensor stops streaming data samples and resumes to a normal Modbus slave operation.
72	Single strain gage sample	1 data byte with value 0xAA	A Modbus packet with one data byte with a value "1" for successful reception, or a standard Modbus error packet if unsuccessful. After successful reception of this command, the sensor sends one packet in Streaming Sample Format (*Section 6.5—Sample Streaming Data Format*), and then resumes normal Modbus slave operation.

Table 6.4—Custom (User-Defined) Service Codes			
Code	Description	Data	Response
105	Lock calibration	0 bytes	A Modbus packet with one data byte with a value "1" for successful reception, or a standard Modbus error packet if unsuccessful. After successful reception of a command to lock the calibration data, the sensor saves the calibration data to permanent memory and prevents further modifications to the calibration data. Note: The Session ID is an exception; this field is always writeable.
106	Unlock calibration	0 bytes	A Modbus packet with one data byte with a value "1" for successful reception, or a standard Modbus error packet if unsuccessful. After successful reception of a command to unlock the holding registers, the sensor allows a user to write to the holding registers. For the newly written data to be made permanent, the "Lock calibration" command must follow. Note: The Session ID is an exception; this field is always writeable.

6.4 Holding Register Addresses and Sample Format

Addresses are base 0:

Table 6.5—Holding Register Addresses and Sample Format				
Register Name	Register Address	Number of Registers	User Read/ Write	Notes
Session ID	0x000C	1	R/W	Optional. Users may write a unique value to this register. Then, in the event the sensor loses power, this register resets to zero, which would be an indication to the user that the session was interrupted by power loss.
Status Register	0x001D	1	R	Data type is unit16_t (refer to *Table 7.1*)
Firmware major version	0x002E	1	R	
Firmware minor version	0x002F	1	R	N/A
Firmware revision	0x0030	1	R	
rawADCReadings	0x0031	16	R	Data type is int32_t Values that are read during the same Modbus command are guaranteed to be from the same ADC read. Values read using separate Modbus commands are most likely from different ADC reads. Values from different ADC reads may cause issues, especially when separately reading the two 16-bit halves of a 32-bit value. Note: These registers are not intended to be used for normal sensor operation, they are intended for debugging.
Reserved	0x0041	16	N/A	
serialNum	0x1000	4	R/W	An ATI serial number for the sensor. A string of up to 8 characters.
partNum	0x1004	16	R/W	An ATI unique identification number for a calibration. If multiple custom sensor models or different calibrations are used, this register can differentiate sensor models. A string of up to 32 characters.
calFamily	0x1014	2	R/W	The ATI Calibration family. A string of up to 4 characters. Note: These registers are not intended to be used for normal sensor operation, they are intended for debugging.

Table 6.5—Holding Register Addresses and Sample Format				
Register Name	**Register Address**	**Number of Registers**	**User Read/ Write**	**Notes**
calTime	0x1016	10	R/W	The date the calibration was released. A string of up to 20 characters. The format is 'YYYY-MM HH:MM' The time of day is in a 24 hour format. Note: These registers are not intended to be used for normal sensor operation, they are intended for debugging.
boardSerNum	0x1020	5	R/W	The ATI unique identification number for the board in the sensor. A string of up to 10 characters. Note: These registers are not intended to be used for normal sensor operation, they are intended for debugging.
forceUnits torqueUnits	0x1025	1	R/W	Data type is uint8_t Force units should always be N. Torque units should always be Nm.
mat00 to mat05	0x1026	12	R/W[1]	Data type is float. The primary sensor calibration matrix. To calculate a vector containing the force and torque values, multiply this matrix with the primary gage vector (refer to *Figure 4.1*).
mat10 to mat15	0x1032	12	R/W[1]	
mat20 to mat25	0x103E	12	R/W[1]	
mat30 to mat35	0x104A	12	R/W[1]	
mat40 to mat45	0x1056	12	R/W[1]	
mat50 to mat55	0x1062	12	R/W[1]	
Reserved	0x106E - 0x1071	1		Reserved
gageMax0 to gageMax7	0x1072	16	R/W	Data type is float.
gageMin0 to gageMin7	0x1082	16	R/W	Data type is float.
Reserved	0x1092- 0x10DA	12		Reserved
tTap0 to tTap2	0x10FE	6	R/W	Data type is float
rdtRate	0x1104	1	R/W	Data type is uint16_t
adcRate	0x1105	1	R/W	Data type is uint16_t
baud	0x1106	2	R/W	Data type is uint32_t
Sync Type / modbus	0x1108	1	R/W	Data type is uint8_t
Reserved	0x1109- 0x1110	1		Reserved
Notes:				
1. These are intended to be read-only and should not be edited by the customer. Writing to read-only fields will damage the calibration of the sensor.				

6.5   Sample Streaming Data Format

Field	Data Type	Sample Data	Notes
**Table 6.6—Streaming Data Format**			
Length	unit8	0x17	Represents the number of bytes in the streaming packet, including the bytes used for the length field. An unsigned 8-bit value.
Sequence	unit8	0x01	An 8-bit incrementing sample counter that automatically goes back to 0x00 after reaching 0xFF.
G0	int24	0xFCD7FF	Represents the gage vector. When calculating F/T values, this vector is multiplied with the calibration matrix using the standard matrix multiplication algorithm.
G1	int24	0xFC8B95	
G2	int24	0xFB3052	
G3	int24	0xF85B58	
G4	int24	0xFEA3FA	
G5	int24	0xF932CF	
Status	uint8	0x04	Represents the Status register The Status register is a bitmap (refer to *Table 7.1*) that contains information about possible errors of various subsystems of the sensor. This Status word in the streaming sample corresponds to the first 8 bits in the Status register.
Checksum	unit16	0xC07C	A standard 16-bit Modbus checksum computed on the status, sequence, and gage vector fields. See *Section 6.2.2* of the *Modbus Serial Line Protocol and Implementation Guide v1.02* for information on the CRC calculation including sample code.

7. Troubleshooting

Basic symptoms of inaccurate data and errors are listed in the following section. For each symptom, causes and appropriate solutions are suggested.

Symptom: Noise — jumps in force torque readings greater than 0.05% of full-scale counts.	**Cause:** Noise can be caused by mechanical vibrations and electrical disturbances that are possibly from a poor ground. Electrical interference can also come from a high noise output device such as a motor.
	Solution: Make sure that the DC supply voltage for the Axia sensor has little to no noise superimposed. Ground the sensor by connecting the cable's shield to ground. In most setups, 0 V is also connected to ground. Connect the robot or other fixture to the same ground.
	Verify that sensor cables do not cross over other cables or are within close proximity to other equipment that could generate electrical noise.
	Avoid sources of mechanical noise. If not possible, apply a filter to the data.
	Cause: Noise can also indicate component failure within the system.
	Solution: Check the status code of the sensor; refer to *Table 7.1*.
	Perform an accuracy check; refer to *Section 7.3—Accuracy Check Procedure* or refer to *Section 4.5: How do I evaluate the accuracy of health of the sensor?* in the Frequently Asked Questions (FAQ) ATI document located at: *https://www.ati-ia.com/library/documents/FT_FAQ.pdf*.
	To return the sensor to ATI for inspection, contact ATI for a Returned Materials Authorization (RMA).
Symptom: Drift — when the force torque data continues to increase or decrease after a load is removed	**Cause:** Some drift from a change in temperature is normal. Drift is observed more easily in the Z axis, compared to the X and Y axes.
	Solution: For approximately thirty minutes, allow the sensor to warm up until it is at a steady state with the air and other objects touching the sensor.
	Use the bias command to shift the readings back to zero. Bias regularly.
	Use an insulator between the sensor and any tooling or fixtures which are at a different temperature. Avoid creating a temperature gradient across the sensor. Shield the sensor from excessive air flow.
	For more information about how to avoid drift from temperature change, refer to the following ATI document: *https://www.ati-ia.com/Library/Documents/DriftExplanation.pdf*.

Symptom:	**Hysteresis —** **when the sensor** **is loaded from a** **zeroed or biased** **state and then the** **load is removed,** **sensor output does** **not immediately** **return to zero.**	**Cause:**	Mechanical coupling or internal failure can cause Hysteresis which is outside of the sensor's specified and acceptable measurement uncertainty (error) range.
		Solution:	Verify the sensor is properly installed, fasteners are tightened, and the customer tooling is securely installed per *Section 3—Installation.*
			Use the bias command to shift the readings back to zero.

Symptom:	**The initial F/T** **values are non-** **zero and no** **load is applied.**	Normal. Bias the sensor to bring all the F/T values back to zero.

Symptom:	**The sensor** **does not report** **accurate F/T data.**	**Cause:**	The sensor may be in an error state.
		Solution:	Check the sensor status code. Refer to *Table 7.1.* If there are no error bits ON, continue troubleshooting.
		Cause:	The sensor is not properly installed or not mounted to a flat, stiff surface.
		Solution:	Verify the sensor is correctly installed per *Section 3—Installation.*
		Cause:	The mounting fasteners are not properly secured.
		Solution:	Verify the fasteners are secured per the installation procedures in *Section 3—Installation.*
			If fasteners are customer supplied, do not use fasteners that are too long. For maximum fastener penetration depth into the sensor, refer to the sensor drawing. When selecting fasteners: use a high quality, high strength screw or bolt and ensure the fastener's material type, fastener head, and fastener grade are proper for the application.
		Cause:	Mechanical coupling — an external object such as customer tooling or utilities contacts a sensor's surface between the mounting side and tool side.
		Solution:	Remove any debris between the tool side and interface plate. Use proper cable management for cables and hoses; do not connect them tightly between the mounting and tool side of the sensor.
			Anything that contacts surfaces such as the through hole in the sensor or cover plates on either side of the sensor induces loading or movement that could result in inaccurate F/T data.

Symptom: The F/T values do not match expected values, for example: the F/T values are fluctuating but are higher than a known applied load.	**Cause:** The sensor may be in a mode that reports gage data instead of F/T data.
	Solution: Gage data is not a 1:1 correlation to F/T axis data. View F/T data instead of gage data.
	Cause: The sensor outputs data in counts. The user must convert the counts to calibration units.
	Solution: Counts must be divided by the Counts per Force (CpF) or Counts per Torque (CpT) in order to convert them to calibration units such as N and Nm.
	In addition to CpF and CpT, depending on the communication protocol, the values may be further scaled by a 16-bit scale factor. 16-bit counts must be divided by (CpF or CpT ÷ 16-bit scale factor) in order to convert to calibration units.
	Cause: If once the F/T readings are converted to calibration units and exceed the sensor's calibration range per *Section 8—Specifications*, the reported values are inaccurate and the sensor may be overloaded.
	Solution: Check the status code. For information on how to read and interpret the sensor's status code, refer to *Table 7.1.*
	Unmount the sensor. Improper mounting methods can induce high loads in the sensor.
	Switch to a larger calibration size, if the application requires loads outside the range of the smaller calibration size.
	After using the larger calibration size and without applying a load, if errors such as "Sensing Range Exceeded", "Gage Out of Range", or "Gage Broken" persist, the sensor is likely permanently damaged due to overload

7.1 Status Register

The following bitmap contains information about the possible errors within various subsystems of the sensor:

Table 7.1—Status Register Bitmap		
Bit No.	**Error**	**Description**
0	Gage out of range	At least one primary or secondary gage is out of range.
1	Internal voltage out of range	One or more of internal supply voltages are out of range.
2	External supply out of range	The external supply voltage is out of range that is specified in *Section 8.1—Electrical Specifications*.
3	Temperature out of range	The sensor temperature is out of the range specified in *Section 8.1—Electrical Specifications*.
4	Internal hardware fault	The internal sensor hardware may not be functioning properly. Complete an Accuracy Check; refer to *Section 7.3—Accuracy Check Procedure*. If the sensor fails, return to ATI for service.
5 to 7	Reserved	

7.1.1 Interpreting ! Specifier Output

The output from ! reports an output in hexadecimals that must be converted to a 32-bit binary number. Refer to the following tables for an example of bit patterns:

If there is more than one error present, the bit pattern can be different, for example:

user: S !

response: 00000005

Using a free online calculator, the user can convert the hexadecimal number to a binary number:

Hex	0	0	0	0	0	0	0	5
Binary	0000	0000	0000	0000	0000	0000	0000	0101

The binary number has 32-bits total. The least significant bit is on the right end of the following table. "1" means the bit is on. "0" means the bit is off.

Binary Number	0	0	0	0	0	000 0000 0000 0000 0000 00	0	0	0	1	0	1
Bit Position	31	30	29	28	27	26 to 6	5	4	3	2	1	0

So in this example, bit number 0 and 2 are on. According to *Table 7.1*, the sensor has a "gage out of range" and "supply voltage" error.

7.2   Reducing Noise

7.2.1   Mechanical Vibration

In many cases, perceived noise is actually a real fluctuation of force and/or torque, caused by vibrations in the tooling or the robot arm. The Varo sensor offers digital low-pass filters that can dampen frequencies above a certain threshold. If digital low-pass filters are insufficient, a digital filter may be added to the application software.

7.2.2   Electrical Interference

To reduce the effects of electrical noise on the sensor, do the following:

- If interference by motors or other noise-generating equipment is observed, check the sensor's ground connections.

- If sufficient grounding is not possible or does not reduce noise, consider using the sensor's digital low-pass filters.

- Verify the power supply is Class 1 which has an earth ground connection.

7.3   Accuracy Check Procedure

> **NOTICE:** The sensor prints compensated gage data. Therefore, in order to accurately run an accuracy check procedure, the user must first convert the raw sensor data into force and torque units. Refer to *Section 4.3.1—To Read Calibrated F/T Data.*

Because each customer application is different, the following guidance permits the user to understand the level of accuracy measurements that the sensor provides with respect to their specific system.

1. Properly install the sensor onto the robot arm.

2. Power-on the sensor.

3. Monitor error bit outputs from the sensor to ensure there are no functional errors. Refer to *Table 7.1.*

4. On the tool side of the sensor, install customer interface plates and tooling.

> **NOTICE:** During the warm-up time and then following accuracy checks, minimize external sources of heat that could cause a temperature change of the sensor and result in inaccurate data.

5. Allow 30 minutes of warm-up time after the sensor is powered on.

6. Move the robot so that the sensor is in six orientations:

   a. In each position, align the sensor axis with the gravity vector.

   b. Record the sensor's output, $F_{x, \text{orientation n}} \setminus F_{y, \text{orientation n}} \setminus F_{z, \text{orientation n}}$ at each orientation without biasing:

   - Orientation 1: +Z up
   - Orientation 2: +X up
   - Orientation 3: +Y up
   - Orientation 4: -X up
   - Orientation 5: -Y up
   - Orientation 6: -Z up

7. Calculate $F_{x, \text{average}} \setminus F_{y, \text{average}} \setminus F_{z, \text{average}}$:

   a. Use the following equations, to complete the calculations:

$$F_{x,\text{average}} = \frac{F_{x,\text{orientation 1}} + F_{x,\text{orientation 2}} + \cdots + F_{x,\text{orientation 6}}}{6}$$

$$F_{y,\text{average}} = \frac{F_{y,\text{orientation 1}} + F_{y,\text{orientation 2}} + \cdots + F_{y,\text{orientation 6}}}{6}$$

$$F_{z,\text{average}} = \frac{F_{z,\text{orientation 1}} + F_{z,\text{orientation 2}} + \cdots + F_{z,\text{orientation 6}}}{6}$$

For each of the six orientations, calculate the following:

$$F_x = F_{x,\text{orientation n}} - F_{x,\text{average}}$$

$$F_y = F_{y,\text{orientation n}} - F_{y,\text{average}}$$

$$F_z = F_{z,\text{orientation n}} - F_{z,\text{average}}$$

$$\text{Tooling Mass} = \sqrt{F_x{}^2 + F_y{}^2 + F_x{}^2}$$

8. The calculated tooling mass in each of the six orientations should not deviate from each other by more than two times the worst-case specified force accuracy rating of the sensor.

7.4 CRC 16-Bit Source Code

If the end user chooses to develop their own software program with Cyclic Redundancy Check (CRC), the following CRC 16-bit source code, which uses a table indexed by 4-bit nibbles, is available for reference.

```
//**************************************************************************
//
// CRC-16-IBM 16-bit CRC algorithm for Modbus.
//
// Polynomial = x^16 + x^15 + x^2 + 1 or 0x18005
//
// This polynomial is used by Bisync, Modbus, USB, ANSI X3.28, SIA DC-07,
// the RTA Discovery process, and many others.
//
//**************************************************************************
#define POLY        0xA001 // 0x8005 bit order reversed
#define CRC_INIT_MODBUS 0xffff // Start value used by MODBUS

uint16_t Crc16(uint16_t crc, uint8_t ch)
{
   static const uint16_t table[16] =
   {// 0xA001 polynomial indexed by nibbles => smallest
      0x0000, 0xCC01, 0xD801, 0x1400,
      0xF001, 0x3C00, 0x2800, 0xE401,
      0xA001, 0x6C00, 0x7800, 0xB401,
      0x5000, 0x9C01, 0x8801, 0x4400
   };

   crc ^= ch;
   crc  = (crc >> 4) ^ table[crc & 0xF];
   crc  = (crc >> 4) ^ table[crc & 0xF];
   return crc;
}

uint16_t Crc16Buf(const void * buff, uint32_t len)
{
   const uint8_t * buf = (const uint8_t *) buff;
   uint16_t        crc;
   uint32_t        i;

   crc = CRC_INIT_MODBUS;

   for (i = 0; i < len; i++)
   {
      crc = Crc16(crc, buf[i]);
   }

   return crc;
}
```

8. Specifications

8.1 Electrical Specifications

Table 8.1—Electrical Specifications	
Parameter	**Value**
Power Supply	4.5 to 5.5 VDC
Power Consumption	<0.5 W
Note: A suitable mating connector to the 8-pin latching surface connector is Molex 5023800800.	

8.2 Environmental Specifications

Table 8.2—Environmental Conditions	
Parameter	**Value**
Operating Temperature	0 to +70 °C
Storage Temperature	-20 to 85 °C
Drift Temperature	5 to 45 °C
Relative Humidity	Up to 95%, non-condensing

8.3 Performance and Strength Specifications

Table 8.3—Performance and Strength Conditions					
Parameter	**Value**				
Vibration Resistance	Levels up to 20 G$_{rms}$ for 30 minutes				
Calibration	Varo55	SI-2400-40			
	Varo62	SI-3000-120			
Resolution[4]		**± Fxy**	**± Fz**	**± Txy**	**± Tz**
	Varo55	0.1 N	0.14 N	0.002 N m	0.0016 N m
	Varo62	0.29 N	0.26 N	0.0048 N m	0.0055 N m
Single-Axis Overload[5]	Varo55	6,800 N	8,800 N	120 N m	120 N m
	Varo62	15,000 N	34,000 N	475 N m	620 N m
Fatigue Life[3]	Varo55	2,400 N	4,000 N	40 N m	54 N m
	Varo62	3,000 N	6,000 N	120 N m	120 N m
Measurement Uncertainty	2% FS				

Notes:

1. The F/T sensor shall be calibrated to the following sensing ranges per ATI's standard calibration procedure. Customer shall receive individualized calibration reports for each F/T sensor.
2. Offset drift due to ambient temperature changes shall not exceed 0.15% FS/°C (average) for any axis over the temperature range of +5 °C to +45 °C.
3. Fatigue life value is rated at any single axis loading up to the published sensing range for that axis. The Fatigue life is 1×10^7 fully-reserved cycles.
4. The calculated 3 sigma noise at 1 kHz data rate.
5. The resultant force or torque applied to one axis. Applying a load to more than one axis at a time may reduce the product's life number of cycles.
6. Any resultant force vector combined with any resultant moment, for example: Fxyz+Txyz, where:

$$Fxyz = \sqrt{Fx^2 + Fy^2 + Fz^2}$$
$$Txyz = \sqrt{Tx^2 + Ty^2 + Tz^2}$$

8.4   Physical Specification

Table 8.4—Physical Specifications		
Parameter	**Value**	
	Varo55	**Varo62**
Weight (maximum)	0.115 kg	0.220 kg
Height (maximum)	13.10 mm (includes two friction shims)	18.33 mm (includes two friction shims)
Diameter (maximum)	55 mm	62 mm
Through Hole	10 mm	10 mm
Mounting Pattern — Robot / Tool	Refer to *Section 9—Drawings*	

9. Drawings

Drawings are available on the *ATI website* or by contacting an ATI representative.

10. Terms and Conditions of Sale

The following Terms and Conditions are a supplement to and include a portion of ATI's Standard Terms and Conditions, which are on file at ATI and available upon request.

ATI warrants to Purchaser that force torque sensor products purchased hereunder will be free from defects in material and workmanship under normal use for a period of one year from the date of shipment. This warranty does not cover components subject to wear and tear under normal usage or those requiring periodic replacement. ATI will have no liability under this warranty unless: (a) ATI is given written notice of the claimed defect and a description thereof with thirty (30) days after Purchaser discovers the defect and in any event, not later than the last day of the warranty period and (b) the defective item is received by ATI not later than (10) days after the last day of the warranty period. ATI's entire liability and Purchaser's sole remedy under this warranty is limited to repair or replacement, at ATI's election, of the defective part or item or, at ATI's election, refund of the price paid for the item. The foregoing warranty does not apply to any defect or failure resulting from improper installation, operation, maintenance, or repair by anyone other than ATI.

ATI will in no event be liable for incidental, consequential, or special damages of any kind, even if TI has been advised of the possibility of such damages. ATI's aggregate liability will in no event exceed the amount paid by the purchaser for the item which is the subject of claim or dispute. ATI will have no liability of any kind for failure of any equipment or other items not supplied by ATI.

No action against ATI, regardless of form, arising out of or in any way connected with products or services supplied hereunder, may be brought more than one year after the cause of action accrued.

No representation or agreement varying or extending the warranty and limitation of remedy provisions contained herein is authorized by ATI, and may not be relied upon as having been authorized by ATI, unless in writing and signed by an executive officer of ATI.

Unless otherwise agreed in writing by ATI, all designs, drawings, data, inventions, software, and other technology made or developed by ATI in the course of providing products and services hereunder, and all rights therein under any patent, copyright, or other law protecting intellectual property, shall be and remain ATI's property. The sale of products or services hereunder does not convey any expressed or implied license under any patent, copyright, or other intellectual property right owned or controlled by ATI, whether relating to the products sold or any other matter, except for the license expressly granted below.

In the course of supplying products and services hereunder, ATI may provide or disclose to Purchaser confidential and proprietary information of ATI relating to the design, operation, or other aspects of ATI's products. As between ATI and Purchaser, ownership of such information, including without limitation any computer software provided to Purchaser by ATI, shall remain in ATI and such information is licensed to Purchaser only for Purchaser's use in operating the products supplied by ATI hereunder in Purchaser's internal business operations.

Without ATI's prior written permission, Purchaser will not use such information for any other purpose of provide or otherwise make such information available to any third party. Purchaser agrees to take all reasonable precautions to prevent any unauthorized use or disclosure of such information.

Purchaser will not be liable hereunder with respect to disclosure or use of information which: (a) is in the public domain when received from ATI, (b) is thereafter published or otherwise enters the public domain through no fault of Purchaser, (c) is in Purchaser's possession prior to receipt from ATI, (d) is lawfully obtained by Purchaser from a third party entitled to disclose it, or (f) is required to be disclosed by judicial order or other governmental authority, provided that, with respect to such to maintain the confidentiality of such information.