User Manual

RS08™

Rotary Shutter

Limited Warranty

Nanomotion Ltd. provides a limited warranty on all its products. For warranty details contact your Nanomotion sales representative.

Patents

Nanomotion products are covered under one or more of the following registered or applied for patents.

5,453,653; 5,616,980; 5,714,833; 111597; 5,640,063; 6,247,338; 6,244,076; 6,747,391; 6,661,153; 69838991.3; 6,384,515; 7,119,477; 7,075,211; 69932359.5;1186063; 7,211,929; 69941195.5; 1577961; 4813708; 6,879,085; 6,979,936; 7,439,652; 7061158;1800356; 1800356; 1800356; 2007-533057 (pending); 2011-093431 (pending); 7,876,509; 10-2007-7009928 (pending); 200780019448.6; 7713361.9 (pending); 12/294,926 (pending); GB2008000004178 (pending); GB2009000003796 (pending); 12/398,216 (pending); GB2446428; 12/517,261 (pending); 08702695.1 (pending); 10-2009-7017629 (pending); 12/524,164 (pending); 12/581,194 (pending)

Website

www.nanomotion.com

Customer Service

Contact your local distributor or email Nanomotion Ltd. Technical Support Department at nano@nanomotion.com, with detailed problem description, additions, corrections or suggestions.

Headquarters

Israel

Mordot HaCarmel Industrial Park
HaYetsira Street, PO Box 623
Yokneam 20692
Tel: +972-73-249-8000
Fax: +972-73-249-8099
Email: nano@nanomotion.com

USA

1 Comac Loop, Suite 14B2
Ronkonkoma
NY 11779
Tel: +1-800-821-6266
Fax: +1-631-585-1947
Email: nanoUS@nanomotion.com

Revision History

The following table shows the last three revisions to this document.

ECO	Doc Rev	Date	Description
--	A	Oct 2013	Initial Release
--	B	Nov 2013	Revised as follows: • Clarified the content of parameters in the Master>Slave format • Added definition of Open and Closed for Blade position • Revised table on page 10 to show Grid part numbers
--	C	Dec 2013	Revised as follows: • Added instructions for torque to mounting screws • Corrected incorrect values or units • Revised text to clarify intent
0801	D	Nov. 2014	Revised as follows: • Corrected page numbering to follow sequentially from front matter • Changed IIC to I2C throughout the manual • In figure 6 on page 19 changed wait $\geq$ 1ms to $\geq$ 4ms • Added note defining the maximum latency transmission bit response from the RS08 as 4ms • In the description of the sleep mode on page 28 changed description of exiting from Sleep mode • Added note on page 15 indicating standard unit delivered with flying-leads (no connector)
	E	Jan. 2015	Administrative change to front matter. Added instructions for trimming the flat cable.
--	F	Apr. 2015	Corrected part number errors on pages 10 and 11.
0916	G	Aug. 2015	Changes based on customer comments: page 15 note re power cable length and information about the barcode label. page 22 added info regarding I2C page 32 added info re extended command format page 36 added fault range

Table of Contents

INTRODUCTION

This user manual provides information about the Nanomotion RS08™ Rotary Shutter family of products. The manual contains the following chapters:

Users should ensure that they have available the NanoCommander software (page 30). This program provides a platform for developing scripts and integration of the Shutter into user systems. The program is available in the RS08 Evaluation Kit or as a download from www.nanomotion.com.

CONVENTIONS USED IN THIS MANUAL

Throughout this manual commands are shown in BOLD and parameter values are shown in italics

NOTE: Notes provide additional information that is not included in the normal text flow.

CAUTION: Caution provides information about actions that will adversely affect system performance.

Best Known Methods (BKM): Provides additional detailed information about operations and methods.

BKM

Danger: Indicates operations or activities that may cause damage to equipment or injury to personnel.

PRODUCT INFORMATION

The RS08™ Rotary Shutter is the first shutter of its kind, utilizing a silent, miniature piezo actuator in an ϕ8mm x 20mm package including the drive electronics. The Shutter is designed to work from a 3.3v battery and supports standard angles of movement from 35° to 120°. Shutter blades can vary in size and material, and blade diameter. Using Nanomotion's proprietary feedback system, the Shutter is fully operational as a closed loop device.

Product Features

- Best Silence, Size, Weight, and Power (SSWaP)
  - Mass: 2g
  - Embedded drive & control electronics in the Shutter body
- Holding and braking with no power consumption
- Silent operation - undetectable
- Low Surge Current
- MIL-STD-810F Compliant for shock, vibration, and temperature - Safe & Reliable

PRODUCT STRUCTURE

The RS-08 is based on a standard actuator and modular parts for adapting the Actuator to specific user needs. The following table shows the part numbers for RS08 Rotary Shutters with standard angle of movement and blade.

Product	Part Number
RS08 Rotary Shutter 35° Stroke Angle with standard blade	RS080350AA-01
RS08 Rotary Shutter 45° Stroke Angle with standard blade	RS080450AA-01
RS08 Rotary Shutter 50° Stroke Angle with standard blade	RS080500AA-01
RS08 Rotary Shutter 55° Stroke Angle with standard blade	RS080550AA-01
RS08 Rotary Shutter 70° Stroke Angle with standard blade	RS080700AA-01
RS08 Rotary Shutter 90° Stroke Angle with standard blade	RS080900AA-01
RS08 Rotary Shutter 120° Stroke Angle with standard blade	RS081200AA-01
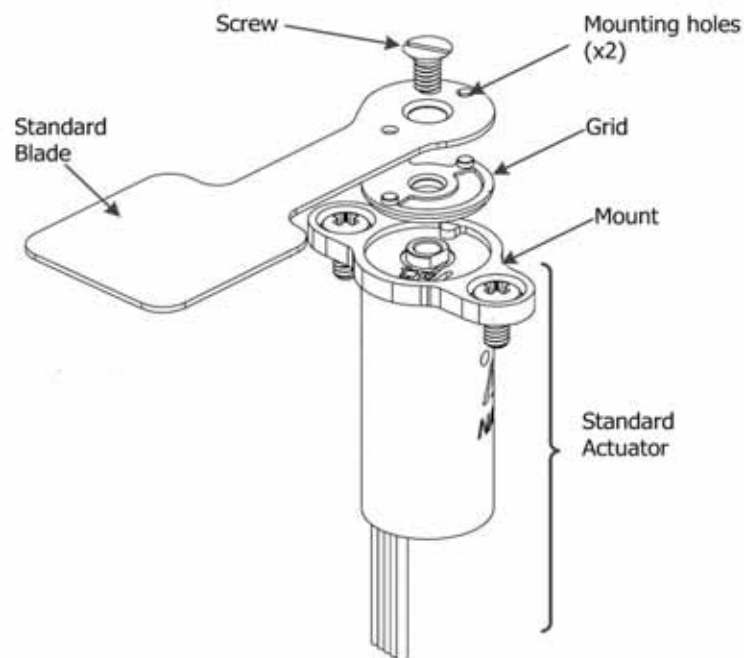
Figure 1: RS08 Rotary Shutter Exploded View

RS08 ROTARY SHUTTER MODULAR PARTS

The Shutter can be configured by assembling different modular parts (Blade, Grid) on the standard Actuator (part number: RS080000CA-XX)

- Grid - controls the angle of movement of the Blade. Refer to the following table for part numbers.

Table 1: Grid Part Numbers

Angle of movement	Part Number
35	RS080350GA-XX
45	RS080450GA-XX
50	RS080500GA-XX
55	RS080550GA-XX
70	RS080700GA-XX
90	RS080900GA-XX
120	RS081200GA-XX

- Blade - S60013C006-XX (Standard Nanomotion model)

> ⚠ Ensure that the correct torque is used when mounting the Blade. The recommended torque is 0.06Nm. Applying greater torque may cause damage to the Shutter Actuator.

- Screw - part number SM16FP03 (standard for all models)

TECHNICAL INFORMATION

The RS08™ Rotary Shutter moves a blade between two positions in one of the standard angles of movement. The action and movement of the blade is controlled by an internal, self-contained Controller/Driver.

SPECIFICATIONS

Weight: 2g

Dimensions: ϕ8mm x 20mm Long

Blade, standard:

- Mass - 0.3g
- Center of Gravity - 16.2mm
- MOI - 30g·mm^2

Performance

- Drive Mode: Closed Loop
- Stroke Angle: standard angles of movement up to 120°. The angle of movement is determined by the assembled Grid.
- Blade position accuracy: ±2°
- Stroke Time, typical: <60msec for 90° (adjustable motion profile)
- Operating Temperature: -40°C to 71°C
- Drive/Control Board: Embedded
- Power Consumption:
    - During motion: peak 400mW in Closed Loop mode
    - Idle (on): 8mW
- Drive Voltage: 2.7 to 3.6VDC
- Communication: I2C

MECHANICAL SPECIFICATIONS

Figure 2: RS08 Rotary Shutter Mechanical Dimensions - Top View

⚠️ Ensure that the correct torque is used when mounting the Shutter.

The recommended torque is 0.06Nm into a metal base material.

Applying greater torque may cause damage to the mounting flange.
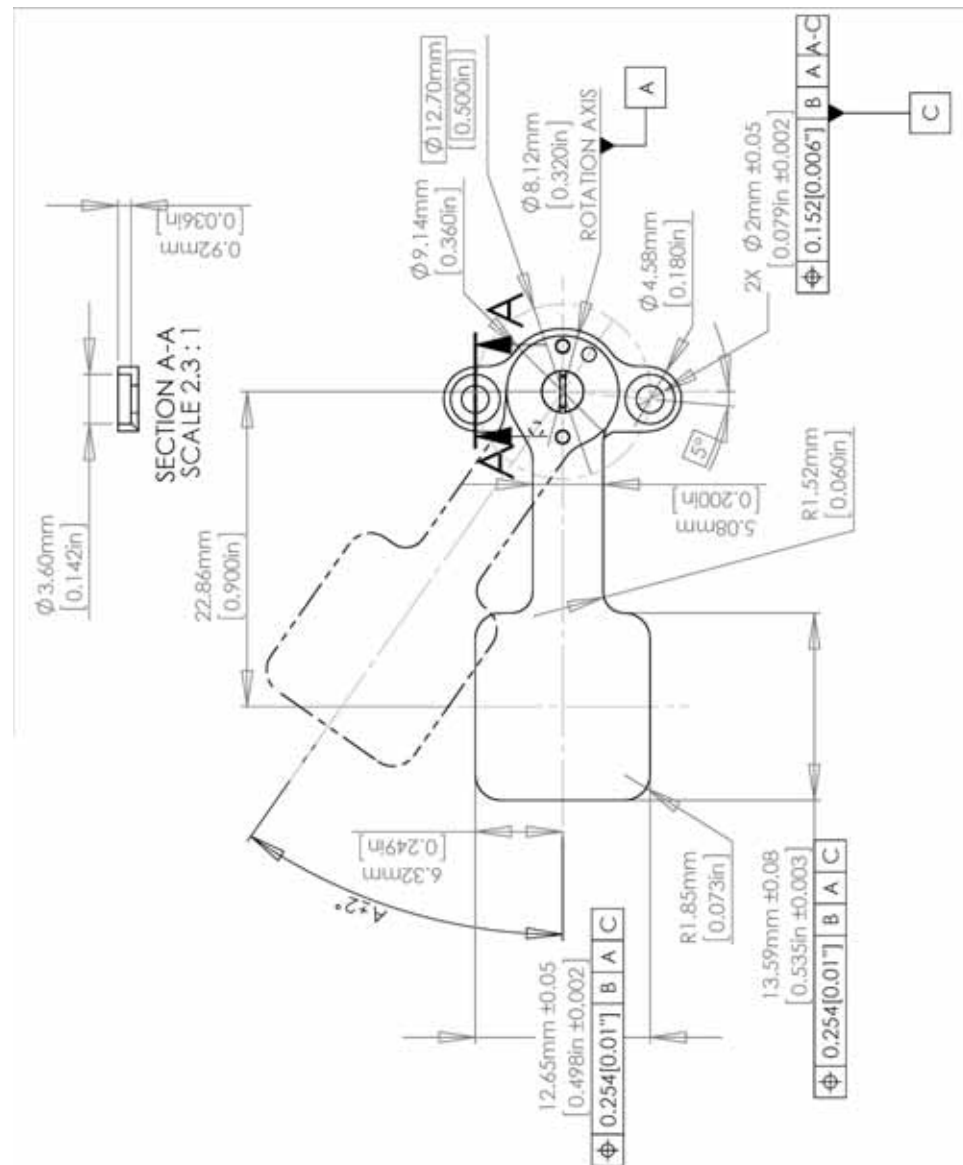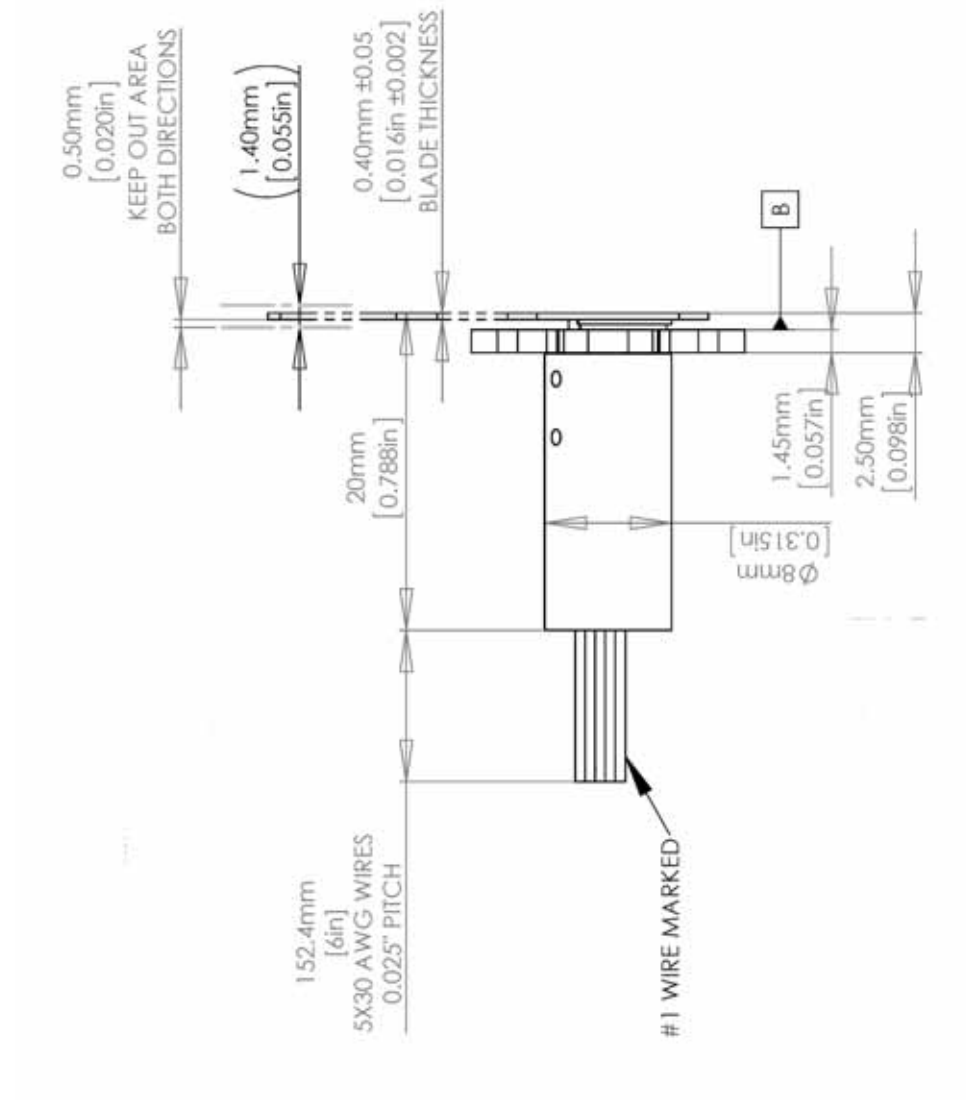
Figure 3: RS08 Rotary Shutter Mechanical Dimensions - Side View

LABELING

RS08 Rotary Shutters have a barcode label attached to the bottom plate.

- Type of barcode: standard 2D data matrix
- Label Dimensions: 5x5mm
- Barcode content: product serial number
- SN:WWYYXXXX

The barcode can be read using a common wireless or USB barcode reader.

Figure 4: Barcode Label on Rotary Shutter Base

ELECTRICAL SPECIFICATIONS

The following table provides pinout information.

Table 2: Actuator Cable Wiring

Pin	Name	Description
1	2.7-3.6VDC	V_{in} **NOTE**: Ensure that the input power wire size is appropriate for the length of the power cable.
2	GND	system ground

Table 2: Actuator Cable Wiring

Pin	Name	Description
3	Reset	Connected internally to V_{in} through pull-up resistor Active low: <0.4VDC Active high: >1.05VDC
4	SDA	I2C Data
5	SCL	I2C Clock

The standard Rotary Shutter is delivered with flying-leads (no connector). For instructions on trimming the flat cable refer to .

!	For special connector requirements contact your Nanomotion sales or engineering representative.

I2C COMMUNICATION

The Shutter supports I2C communication at 100 kbit/sec to 400 kbit/sec signal in I2C Fast mode, over a 2 foot cable, with a line capacitance of 35pF. For longer cables calculate line capacitance based on the desired cable specification.

!	The I2C 100 and 400 kbit/sec communication signals were tested with the MCU operating at a 1 KHz message rate request.

BKM	Calculations of line capacitance and user end pull-up requirements shall be in compliance with the **I²C-Bus Specification and User Manual**, UM10204 rev 03.
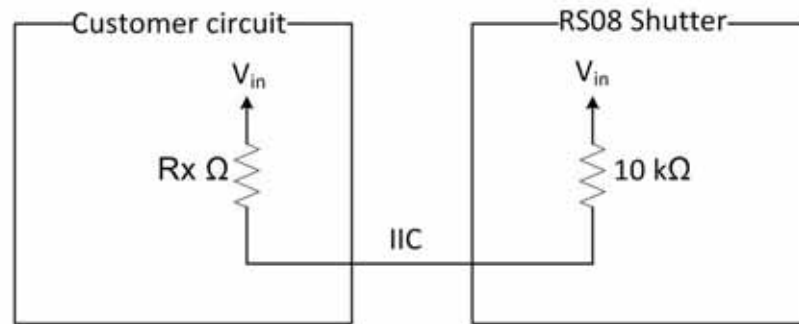
Figure 5: I2C Circuit

Power input of Shutter should be decoupled by low ESR ceramic capacitor of 10uF minimum. The decoupling capacitor should be placed near to the Shutter cable connection points.

GND and Supply wires should be connected to low impedance power plains.

I2C signal lines do not require termination because the Shutter has built in pull-up resistors.

CURRENT CONSUMPTION

The peak current consumption is <100 mA. When measuring current consumption spikes may be visible on the scope. These are typical and result from the Shutter's internal switching power supply. The internal power supply raises the V_{in} to 4.2VDC. To measure the actual power consumption average the signal over time (e.g. 10ms).

OPERATING THE RS08™ ROTARY SHUTTER

The RS08™ Rotary Shutter is delivered with a internal Controller/Driver that is pre-configured and calibrated. The Shutter's cable is open-ended allowing the user to provide system specific connections.

The Shutter is connected to a user supplied MCU and power as shown below. The MCU can be either a standard computer running the NanoCommander program, or a user supplied unit that is capable of I2C communication. The user is able to control the Shutter using scripts, and read/modify parameters over I2C wires (SDA, SCL, Reset).

For additional information refer to:

-
-
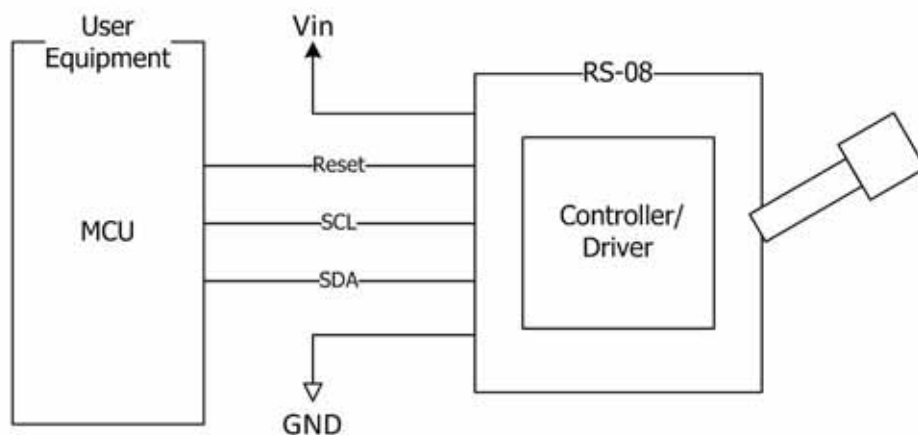
Figure 6 System Block diagram

I2C COMMUNICATION

The controller driver implements I2C Slave protocol. The communication is done by exchanging messages between the Master and the Slave using standard I2C communication at a frequency of 100 kHz or 400 kHz. For detailed information refer to **I2C-Bus Specification and User Manual**, UM10204 rev 03.

OPERATING SEQUENCES

This section describes the following system operations:

- Power-On
- Command Sequence
- Shutter Operation Command Sequence

! The RS08 Rotary Shutter is pre-programmed and calibrated before delivery.

POWER-ON

The Shutter is delivered with a pre-configured Power-On sequence that prepares the Shutter to receive commands. The user can add a calibration step as shown in . The additional step must be saved to the Flash to ensure its operation on all future Power-On sequences.
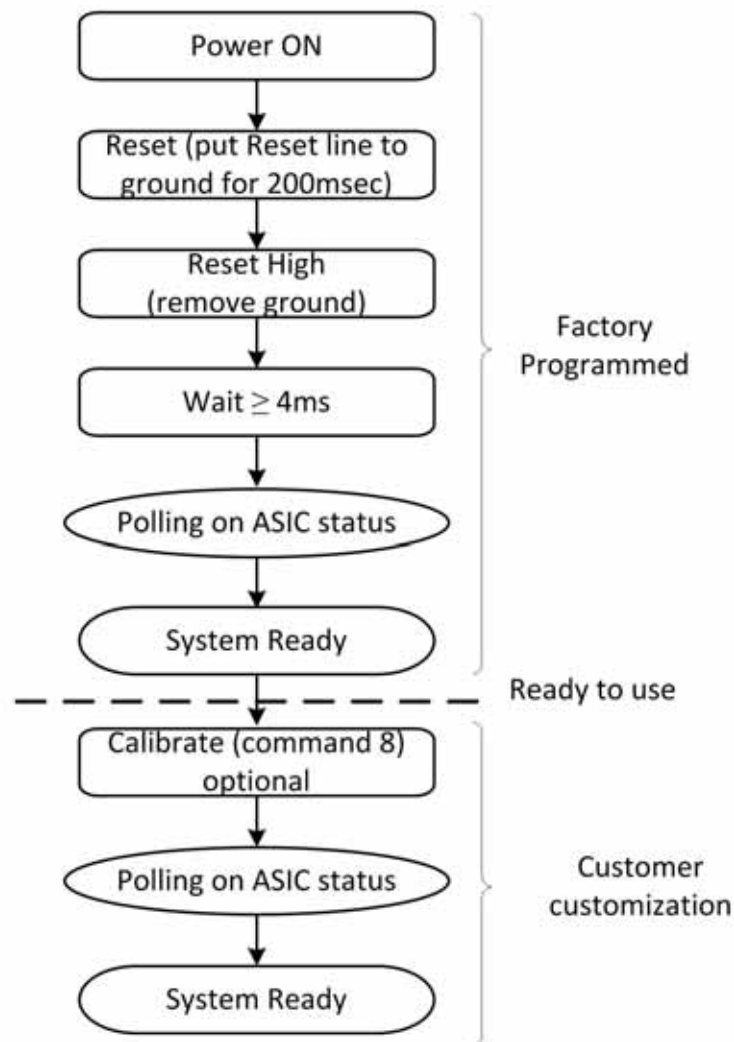
Figure 7 Power on sequence

RESET

The Reset line provides a Power-Down to the Shutter. The Shutter is reset on the negative transition of the Reset line. This causes the Shutter to clear its RAM and reload the last saved parameter set from Flash memory.

The line is held high (>1.05VDC) during normal operation. To reset the Shutter drive the line low (<0.4VDC) and hold it there for 200 msec. After the reset pulse is removed, the line returns to a high condition. A delay of 1 msec is needed before a command is sent to the Shutter to allow completion of the reset cycle.

During reset the Shutter Blade remains in the same position until a new move command is received.

CALIBRATE

The *CALIBRATE* command sets the Shutter's optimal operating frequency and position sensor output.

COMMAND SEQUENCE

The command sequence begins with the Master sending a command to the Shutter. The Shutter reads that command and checks the status of current commands. If commands are being executed, the new command is queued. When the all queued commands are completed and the system is Idle, the reads the command. If the command is to move the Blade, the Shutter checks the ambient temperature and adjusts its operating frequency for peak performance before executing the command. For all other commands the Shutter executes the command.

Refer to Temperature Processing on page 35 for information about enabling/disabling this function. A general command execution sequence is shown in Command Execution Flow on page 21.
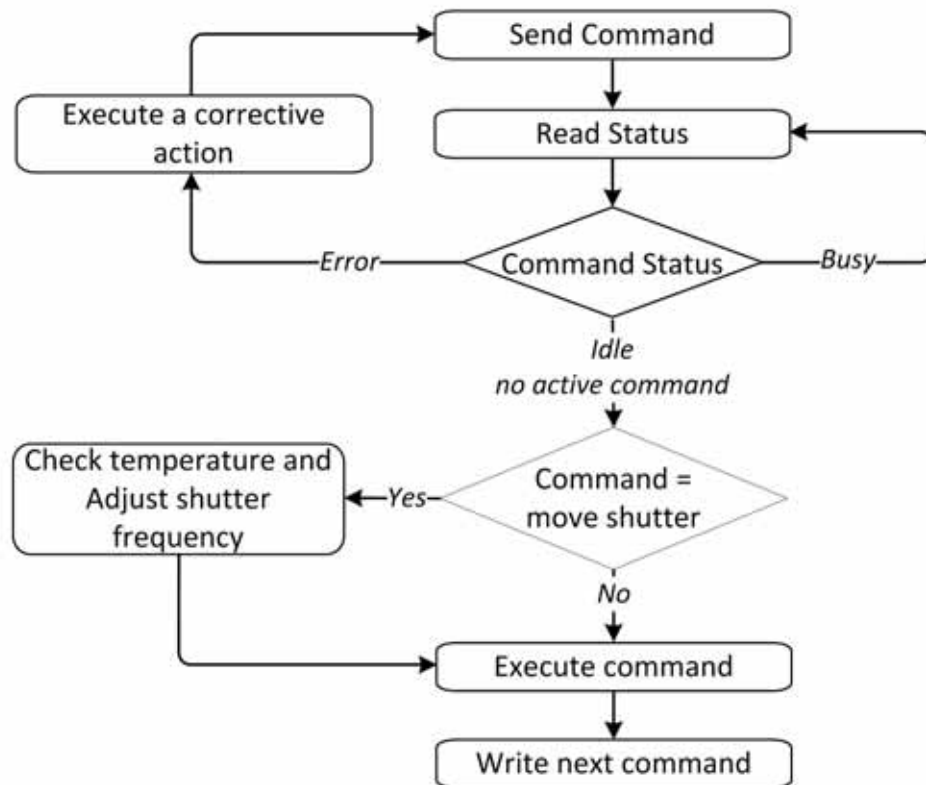
Figure 8 Command Execution Flow

SHUTTER OPERATION COMMAND SEQUENCE

Shutter motion is initiated by *OPEN/CLOSE SHUTTER* (command 23). The movement of the Shutter has four stages as shown in Figure 9.

- Initial movement from stop to High Velocity (msec)
- High Velocity (degrees/sec)
- Transition to low velocity (msec)
- Low velocity (degrees/second)

> **!** The velocity of movement is defined by the programmed FW and is not changeable by the user.

The user can create specific scripts (page 24) to define unique Shutter actions. These scripts can be run from the user MCU, or Host computer using the NanoCommander program (page 30).
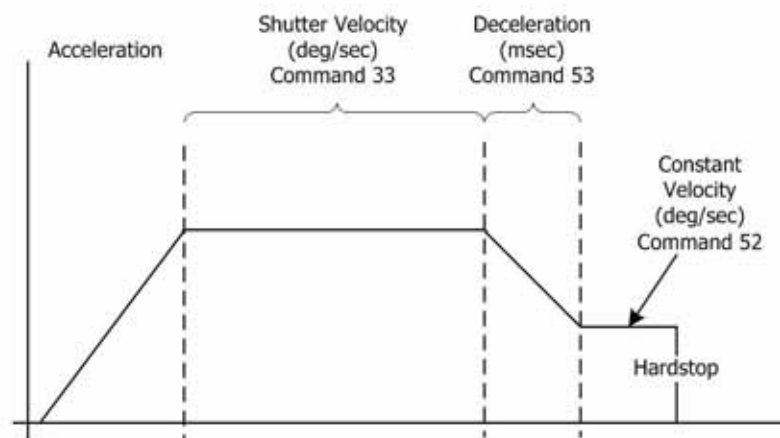
Figure 9 Shutter motion profile

KEEP POSITION

When the Shutter reaches the stop, the position is maintained by monitoring the position sensor's signal. If the blade has been moved by an external force, the sensor changes its state and the Controller-Driver changes the **Shutter in Position** status to **0**. If *KEEP POSITION* (command 46) is enabled, the Shutter applies motion towards the stop in order to return the Shutter to its correct position.

The *POWER SAVE* (command 45) command affects the timing of the *KEEP POSITION* command

- *POWER SAVE* disabled - the Shutter monitors the Position sensor continuously.
- *POWER SAVE* enabled - the Shutter switches to low-power mode after motion termination. The Shutter wakes up periodically (every 100 msec) and checks the status of the Position sensor. If action is required the Shutter moves the Shutter to its correct position and then returns to low-power mode again.

CONTROLLING SHUTTER OPERATIONS

The RS08 Rotary Shutter is factory programmed with a Nanomotion Firmware (FW) version. The FW controls the Shutter in response to commands from either of two sources.

> **!** For detailed information about the commands, responses, and variables recognized by the Shutter refer to .

NanoCommander

When control is from a Host computer using the NanoCommander program, commands can be sent individually or executed as a script from the NanoCommander's window. Refer to for NanoCommander operating instructions.

User MCU

The Shutter operation can be controlled by sending individual commands or running a script saved on the user's MCU over the I2C lines. Command formatting and syntax must comply with the Shutter's FW requirements. The following sections provide information on message formatting.

IIC COMMUNICATION.

The controller driver implements IIC slave protocol. The communication is done by exchanging messages between the master and the slave using standard IIC communication at a frequency of 100 kHz or 400 kHz.

!   400 kHz rate is not compatible with Power Save mode. By default, Power Save mode is enabled; only 100 kHz rate is functional. To enable 400 kHz rate, first disable Power Save with command 45, parameter 0.

There are two types of messages:

- Master->Slave: the master sends a message to the slave (see Master->Slave Message)
- Slave->Master: The master reads a message from the slave (see ? "Slave>Master Reply).

Both types of messages are initiated by the master.

The slave write address is A4H and the slave read address is A5H.

EXAMPLE OF IIC COMMUNICATION

See attached two scope diagrams of I2C signals. The first diagram (Figure 2) is an example of sending Get Info command, the second one is reading of 16-byte response.

On the first diagram, four bytes easily identified:

- Address (0xA4)
- Get info Command (0x13)
- Parameter low (0)
- Parameter high (0)

All bytes are sent by the master.

Pay attention to acknowledge bits sent on each 9th clock. The acknowledge bits (zero level) are sent by the controller/driver.
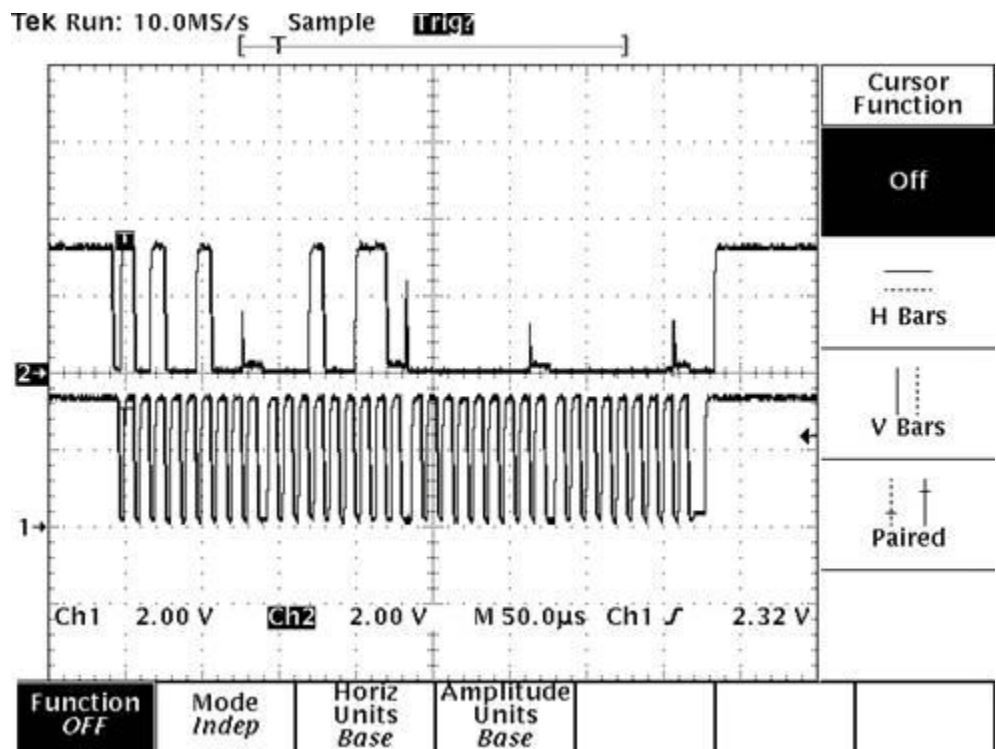
Figure 10 Figure 2:IIC Get info (#19) command example

On the second diagram, the whole sequence is in the bottom, and zoomed picture above shows three first bytes:

- Address (0xA5) - sent by the master
- Last executed command (0x13) - sent by the drive
- Status Idle (0x01) - sent by the drive

In this case the acknowledge bit after the address is sent by the drive, and all other acknowledge bits are sent by the master. The master sends negative acknowledge (high level) after the last byte to indicate the end of transaction. The master also can send negative acknowledge after any byte to break the transaction if necessary.

Pay attention to delay in clock after the first byte. The clock is always provided by the master. However, the delay is commanded by the drive, because it requires time to prepare response sending. To cause delay, the controller/ driver holds the clock low for a while. The master should react as defined in IIC specification.
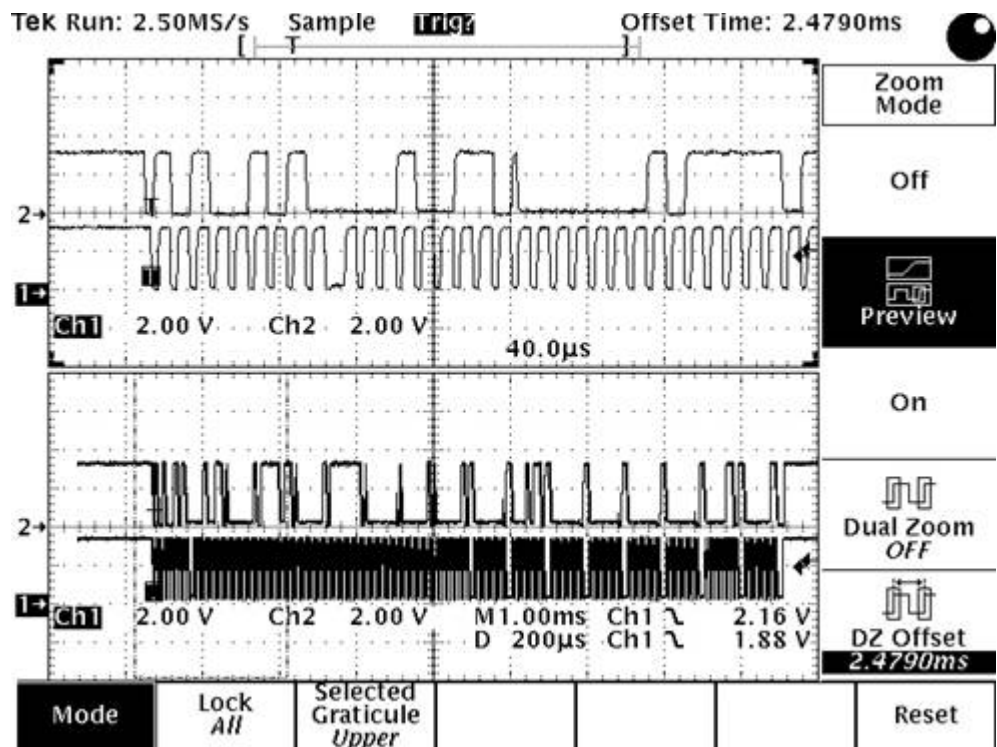
Figure 11 Figure 3:IIC Read command example

I2C MESSAGE FORMAT

The command format includes an address, the command, and parameters if required for the command. The following table defines each part of the format.

Table 3: Master>Slave Command Format

Field Name	Size in Bytes	Description
Address	1	I2C device address (0xA4)
Command	1	Code number in Hex
Parameter	2	The parameters are binary and in little-endian format. If no parameters are needed for the command the two Bytes must contain zeroes (00 00)

According to I2C standards the full command is bracketed by Start and Stop bits. After sending a single Byte of the command, the Master waits to receive an Acknowledge bit from the Shutter. The next Byte is not sent until the Acknowledge bit is received. This allows the Shutter to complete other

operations before receiving new information. A reply message is sent from the Shutter, if required by the command. The format of the reply is determined by the command.

!   The maximum latency acknowledge bit transmission from the RS08 is 4msec.

COMMAND AND SCRIPT EXAMPLES

The following examples provide a basis for constructing both simple and complex scripts. These examples are based on the NanoCommander specifications. Command and script formatting for a user's MCU not using the NanoCommander, may be different.

COMMAND FORMAT EXAMPLE

When using the NanoCommander, commands and scripts are written as a text file (.TXT). The command names may be written in full with the parameter, or the command code written with parameter.

BKM   When working with the NanoCommander, individual commands can be sent directly from the NanoCommander's window.

Using either of the examples below the **FREQUENCY** command (page 33) sets the Shutter frequency to 128 KHz (20MHz/128=156KHz).

	Command	Parameters
as Text	Frequency	128
as Code	command 12	128

If the command contains more than one word the command is written as a single word with no space between the words. For example **SET SHUTTER** is written as **setshutter**. There must be a space between the command and its parameter.

examples:

```
openloop X (X=0 to ±30000)
setshutter X (X=0, 1)(open or close the shutter)
frequency XXX (XXX is the divider in the formula f=20M/XXX)
```

Open and close the Shutter with a 500msec delay

In this example the script causes the Shutter to open and close 100 times with a delay of 500msec in between each blade movement.

The **REPEAT** and **DELAY** commands are functional within the NanoCommander program and cannot be sent to the Shutter as individual commands.

Script Commands	Explanation
`repeat 100`	Repeat the script 100 times
`setshutter 1`	move Shutter's blade to open (Set Shutter, [page 33](#))
`delay 500`	wait 500msec
`setshutter 0`	Close Shutter's blade
`delay 500`	wait 500msec
`endrepeat`	End script

Open/Close Shutter at 50% PWM, and 50% Duty Cycle

In this example the script causes the Shutter to open and close 1000 times using a 50% PWM with a delay of 200msec between each blade movement.

Script Commands	Explanation
`repeat 1000`	Repeat the script 1000 times
`openloop 15000`	Move the Shutter's blade in a positive direction at 50% PWM (**OPEN LOOP**, [page 32](#))
`delay 200`	wait 200msec
`openloop 0`	No movement of Shutter (0% PWM)
`delay 200`	wait 200msec
`openloop -15000`	Moves the Shutter's blade in a negative direction at 50% PWM
`delay 200`	wait 200msec
`openloop 0`	No movement of Shutter (0% PWM)
`delay 200`	wait 200msec
`endrepeat`	End script

NANOCOMMANDER

The NanoCommander is a proprietary program that provides the ability to communicate with the Shutter. The user can initiate scripts, apply manual commands, read parameters from the Driver's RAM, and save RAM to Flash or Flash to RAM.

Commands can be executed by:

- Direct command-initiating buttons
- Running command scripts (provided with the application)
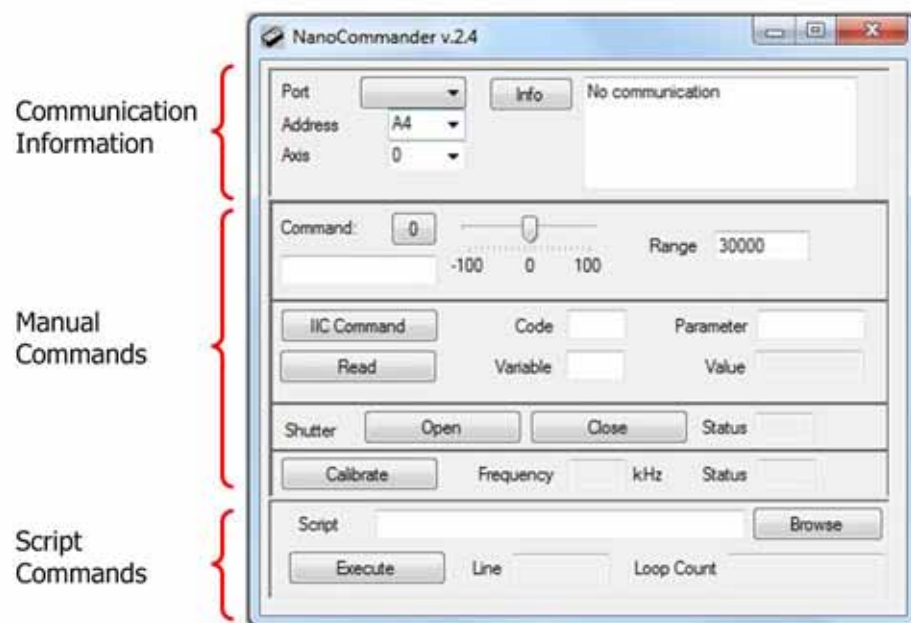- Running manual commands (I2C commands)

Figure 12 NanoCommander Panels

Connection Info Area

Determine and select the following information parameters:

- Port: Select the USB-port (COM) in which the Shutter evaluation kit is connected to your PC.
- Address: choose A4, to indicate the controller's I2C address
- Axis: choose Axis 0

Manual Commands

The Manual Commands area allows running direct commands by either pressing on a command button (i.e. **CALIBRATE**), moving a slider, or by entering an I2C command code and parameters.

Running Scripts

The system is provided with predefined scripts, allowing the user to quickly and easily operate the Shutter.

To execute a script:

In the Script Command area, perform the following:

- Press on the Browse button and select the desired script.
- Press on the Execute button. The button toggles to Cancel, when a script is running.

Verify that the desired action is performed. You may stop the script execution at any time, by pressing on the Cancel button.

FW COMMANDS, PARAMETERS AND REPLIES

This section provides information about the set of commands used to control the RS08™ Rotary Shutter operations.

Throughout this section the term Master refers to an external MCU or Host computer connected to the Shutter. The Slave is the RS08 Rotary Shutter.

COMMANDS

Table 4 and Table 10 specify valid command codes.

| ! | Some commands change Shutter operating parameters for the duration of the command. These changes are lost on restart. To make the parameter change permanent and save to flash memory use the command **SAVE PARAMETERS** (13). This applies to parameters affected by commands: 12, 25, 33, 46, 48, and 50. |

Table 4: Command

Command	Code	Parameter (DEC)	Description
Open Loop	7	PWM value	Command initiates open-loop motion with the specified PWM value. The parameter provides a division of PWM from maximum to minimum PWM. The is ±30000. For example +15000 provides 50% PWM in one direction. The motion time is limited by Timeout value. Refer to Set Timeout, page 34.
Calibrate	8	00 00	Calibrate the motion system The command execution time is within 0.8 second.
Sleep	9	00 00	Switches the Shutter to Sleep mode. To exit Sleep mode perform full power up sequence as shown in Power-On, page 19.

Table 4: Command

Command	Code	Parameter (DEC)	Description
Frequency	12	Divider	Sets the Shutter frequency based on the formula the 20MHz/Divider. Divider ranger is 120 to 132. Default Frequency is the last saved frequency. **Note**: The change is to RAM only. To save the change to Flash memory use **SAVE PARAMETERS** (13).
Save Parameters	13	00 00	Save the current RAM settings to Flash memory. **SAVE PARAMETER** must be used to save the following parameters to flash memory: 12, 25, 33, 46, 48, and 50.
Retrieve Parameters	14	00 00	Restores Flash memory settings to RAM.
Get Info	19	00 00	Requests information from the Shutter. The information is supplied in the 10-Byte reply extension: Bytes 0-3: Firmware version Bytes 4-7: Serial number Bytes 8-9: Application ID
set Shutter	23	0 - close 1 - open	Open or close the Shutter. Default position is closed. Refer to Figure 2, page 13 for explanation of Open/Closed position.

Table 4: Command

Command	Code	Parameter (DEC)	Description
Set Timeout	25	Timeout in msec	Set maximum time of *OPEN/ CLOSE SHUTTER* operation and *OPEN LOOP* motion in milliseconds. • Range: 1 to 5000msec • Default value: 500msec. If motion is not completed within the specified time, motion is stopped and the Timeout bit is set in the *MOTOR STATUS BYTE*. **Note**: Use *SAVE PARAMETERS* (13) to save change to flash memory.
Set Shutter Velocity	33	Velocity in degrees per second	Specifies Shutter velocity when executing *OPEN/CLOSE SHUTTER* command (23). Recommended range is from 800 to 2000 degrees/sec. **Note**: Use Save Parameters (13) to save change to Flash memory.
Power Save	45	0 - disable 1 - enable (default)	When enabled the Shutter is in Power Save mode. If power saving is enabled, the Shutter switches to low-power mode when the Shutter motion finishes. For details of operation in each state refer to Keep Position, page 22. **Note**: When Power Saving is enabled I2C communication at 100 kHz is supported.
Keep Position	46	0 - disable automatic return, 1 - enable automatic return (default)	Enables or disables automatic return to hard stop using the Position sensor. Refer to Keep Position, page 22 for details. **Note**: Use *SAVE PARAMETERS* (13) to save change to Flash memory.

Table 4: Command

Command	Code	Parameter (DEC)	Description
Temperature Processing	47	0 - disable 1- enable	Enables or disables automatic temperature compensation. Default is Enable
PWM Limit	48	0-30000	Defines max PWM Level during motion. 100% PWM = 30000 Default value is 18000. **Note**: Use *SAVE PARAMETERS* (13) to save change to flash memory.
Home	50	0-Open (default) 1- Close	Sets home direction after calibration. **Note**: Use *SAVE PARAMETERS* (command 13) to save change to flash memory.
Set Low Velocity	52	Velocity in degrees per second	Specifies target velocity during *OPEN/CLOSE* (command 23) when approaching the hard stop. Recommended range is between one-third and two-thirds of the normal velocity. The controller uses this value to create deceleration from *SET SHUTTER VELOCITY* (command 33).
Set Velocity Ramp	53	Velocity ramp time in milliseconds	Defines the time of the deceleration ramp from normal Shutter velocity to low Shutter velocity.
Get Variables by ID	248	Request current status of listed variables from the Shutter. For information about the command and the Shutter's response refer to Get Variables by ID Command, page 36	

GET VARIABLES BY ID COMMAND

The RS08 Shutter monitors the operation of the Shutter and saves specific information as a Variable with an ID number. The **_GET VARIABLES BY ID_** command is an special command that requests Shutter operating information based on the ID number. The command does not cause a change in the operation of the Shutter. The requested information is gathered and added to the next Shutter reply.

The **_GET VARIABLES BY ID_** command defines the content of the reply extension Bytes. After a **_GET VARIABLES BY ID_** command, subsequent read returns values of the requested variables. Any number of reads can follow a single **_GET VARIABLES BY ID_** command. Each read returns instantaneous values of the requested variables, allowing continuous monitoring of the variables.

The content of the extension Bytes can be redefined with the next **_GET VARIABLES BY ID_** command, or the **_GET INFO_** command.

Any other command following the **_GET VARIABLES BY ID_** command can change standard Bytes of the reply, but cannot redefine the extension. This allows the Master to continue monitoring the previously requested variables.

The Slave can break the read cycle after any Byte by not acknowledging the last received Byte. Therefore, no cancellation of the extension data is required. The Master simply ignores the extension when it is not required.

EXTENDED COMMAND FORMAT

Unlike basic command that always includes 3 bytes, extended command can be of any length up to 255 bytes (tools and libraries can apply additional limitations; it's generally recommended to use commands not longer than 16 bytes).

Extended commands have the following format:

Table 5: Extended Command Format

Byte	Description
0	IIC device address (0xA4)
1	Command code - 248 (0xF8)
2	Command length including bytes 1, 2, 3, and all subsequent byte N<256
3	Extended command code (0x42 - Get Variables by ID)
4 to N	Data, format varies from command to command

Notes:

- All extended commands have the same command code 248 (0xF8). Extended command code (byte 3) identifies specific command.
- After execution of extended command, byte 1 in the controller reply reads value 249 (0xF9). After any standard command, it reads command code.

GET VARIABLES BY ID COMMAND FORMAT

The *GET VARIABLES BY ID* command is referred to as an extended command. It is the only extended command used by the Shutter. The extended command has a different format than a standard command.

Table 6: Get Variable by ID Command Format

Byte offset	Value	Comments
0	Address	I2C device address (0xA4)
1	248 (0xF8)	Command code for *GET VARIABLES BY ID*
2	4 - 8	decimal value indicating the command length, allowing request for up to 5 variables
3	66 (0x42)	fixed value defining the Extended command code
4	ID	ID of the first variable (the valid variable IDs are listed under Valid IDs, page 37)
5	ID	ID of the second variable (optional)
6	ID	ID of the third variable (optional)
7	ID	ID of the forth variable (optional)
8	ID	ID of the fifth variable (optional)

GET VARIABLES BY ID AVAILABLE VARIABLES

The following Variable ID's are valid:

Table 7: Valid IDs

ID	Size in Bytes	Comments
2	2	Measured temperature in °C.
4	1	Instantaneous Blade position voltage. ADC readings are provided in ADC units. To calculate voltage, use formula: Voltage = ADC_Value/256*2.5

Table 7: Valid IDs

ID	Size in Bytes	Comments
6	2	Instantaneous PWM value of the Shutter mechanism. The variable is updated during motion.
10	2	Frequency divider. Refer to **FREQUENCY** (command 12) in Table 4
12	2	Motion time. The value reads the motion time in units of 0.1 milliseconds. To obtain motion time in milliseconds, divide the value by 10. The measurement starts with each motion start, and measures time until three conditions are met: motion finishes; the PWM switches off; and the position sensor turns active.
13	2	Motion Path. The difference between the values of ADC1 at Motion End and the Motion Start.
31	2	PWM limit
32	2	Timeout

GET VARIABLES BY ID REPLY FORMAT

The Shutter replies to Master-initiated read cycle with a sequence of Bytes as listed in the following table.

> **!** An erroneous ID specified in the command does not cause a failure or error report. However, the returned value for the incorrect ID may be garbage.

Table 8: Slave Reply Format

Field Name	Byte offset	Description
Address	0	Slave IC device address
Command	1	Code of the last command received by the Slave.
Command Status	2	Status of the controller.
Motor Status	3	Position of the Shutter
Reserved	4,5,6	
Extension	7-16	Additional data specified in the last **GET VARIABLES BY ID** command

After ***GET VARIABLES BY ID*** command, 10 extension Bytes read the data as shown in the following table:

Table 9: Byte Offset of ID Variables

Byte offset	Description
7,8	Instantaneous value of the first requested variable
9,10	Instantaneous value of the second requested variable
11,12	Instantaneous value of the third requested variable
13,14	Instantaneous value of the forth requested variable
15,16	Instantaneous value of the fifth requested variable

If less than 5 variables were specified in the last ***GET VARIABLES BY ID*** command, the value of the extra Bytes is undefined and will not read by the Master.

The controller always transmits two Bytes per each requested address. Interpretation of the Bytes depends on the variable type:

- If variable length is two Bytes, the reply contains the Bytes in the big-endian format (most significant Byte first).
- If variable length is one Byte, the first reply Byte reads the variable's value, and the second Byte is undefined.
- If variable length is more than two Bytes, the reply contains two initial Bytes of the variable. For integer variables, these Bytes read the most significant Bytes of the value.

SLAVE>MASTER REPLY

The Master initiates a read cycle by sending the read address to the I2C bus. The Slave responds by sending reply Bytes. Permanent part of reply includes 6 Bytes. The Master can specify up to 10 extension Bytes using commands like Get Info. When an extension specifying command is issued, the Slave continues sending the extension Bytes in each read cycle.

The Master can break the read cycle after any Byte by not acknowledging the last received Byte. Therefore, no cancellation of the extension is required - the Master simply ignores the extension when it is not required.

The read command has the following structure:

Table 10: I2C Read Command Format

Sent by	Field Name	Size in Bytes	Description
Master	Address	1	I2C device read address (default 0xA5)
Slave	Command	1	Code of the last executed command.
	Command Status	1	Status of the last command execution.
	Motor Status	1	Bit-coded motor status.
	Reserved	3	
	Extension	0-10	Version data, if requested by command 19; variable values, if requested by extended command.

The following table lists the possible codes in Command Status Byte:

Table 11: Controller Driver Command Status Report

Command Status	Code	Description
Idle	1	The last command executed successfully. Driver is ready to receive the next command.
Error	2 or any number ≥ 4	The last command failed. Driver is ready to receive the next command.
Busy	3	Execution of command is in progress. Other commands should not be sent while the driver is busy.

The following table lists the bits in Motor Status Byte:

Table 12: Motor Status Byte

Bit number	Bit name	Description
0	Shutter in position	0 - Not in position 1 - in position Open/Close
1	Motion state	0 - Idle 1 - In motion.
2	Low velocity	0 - Velocity Ok 1 - Velocity was below threshold for more than 30 msec.

Table 12: Motor Status Byte

3	Timeout	0 - No timeout
		1 - The last Shutter operation took more than timeout value (defined by command 25).
4	Calibration state	0 - The system was not calibrated or calibration failed.
		After power-up, the drive starts in this state.
		1 - Calibration executed successfully.
5	Shutter position	0 - Open
		1 - close. Valid only if Bit 0 is one.
6	fault range	0 - OK
		1 - travel less than expected minimum distance

APPENDIX

TRIMMING THE FLAT CABLE

The standard RS08 Rotary Shutter is delivered with a 5 wire flat cable, flying leads (no connector). The flat cable has a silicon insulation that must be carefully trimmed to attach to a connector or other circuit.

The following provides basic instructions for trimming the cable ends. For additional details, including a short video, refer to http://www.cicoil.com/flex_cables/flat_cable_overview/cable-preparation.asp.

1. Cut the cable to the desired length.
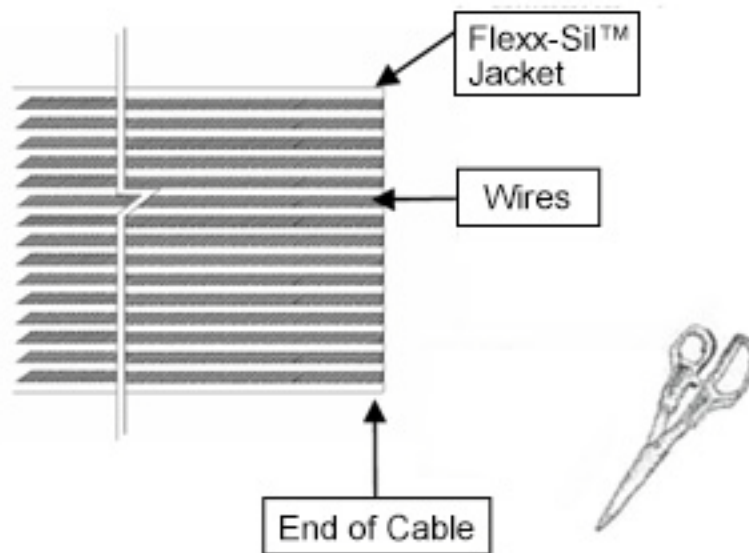
Figure 13:

2. Mark a line on the cable using a pen or marker.
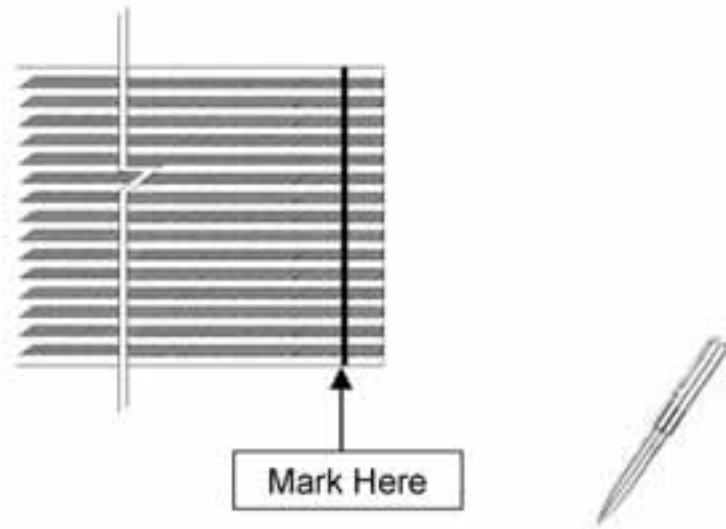
Figure 14:

3. Cut the jacket along the line of both sides oft he cable.
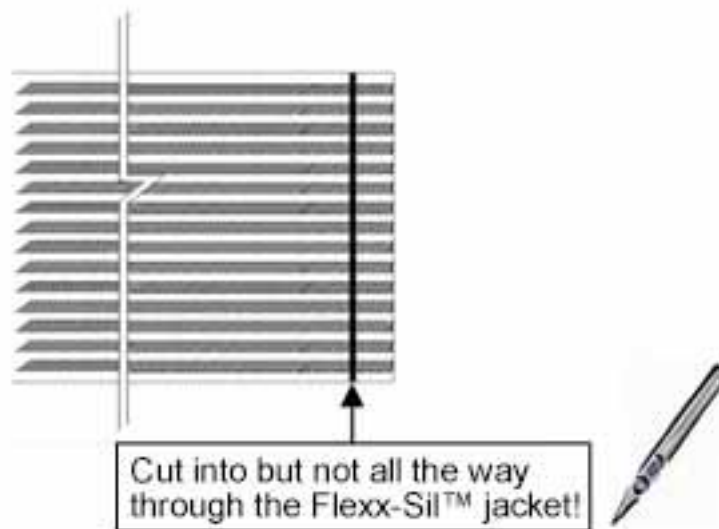
Figure 15:

4. Slice the jacket between the wires along the dotted lines as shown.

This step is optional for cables with non-shielded wires.

Slice from the end & stop at the desired length.
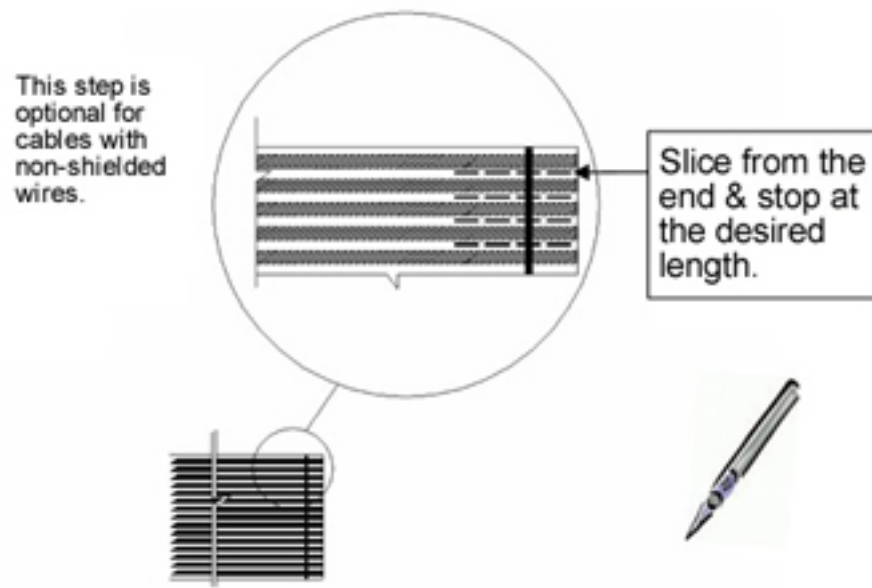
Figure 16:

5. Strip the end of each wire using tweezers.

Grab the Flexx-Sil™ jacketing here and remove by pulling it out.
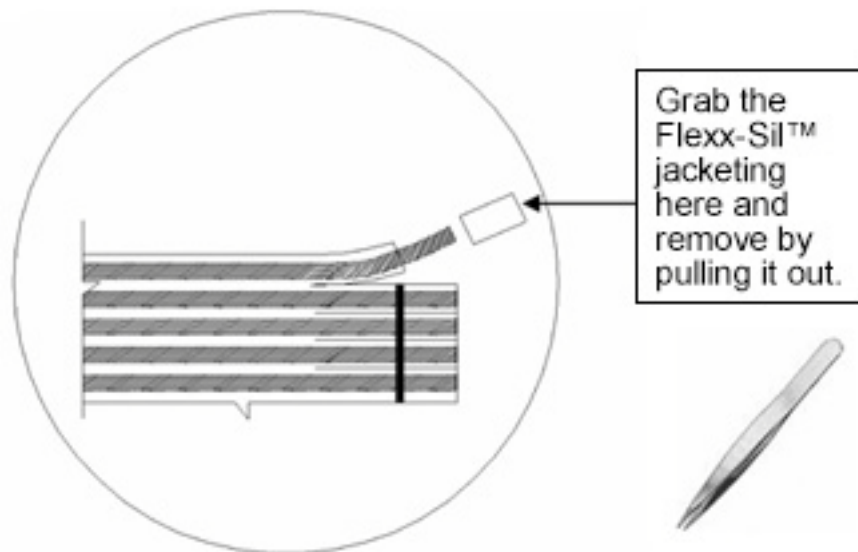
Figure 17:

6. Arrange the ends of the wires to fit into the connector shell.

Note that the ends of the wires can be arranged into various shapes so they can fit into round, oval or square back shell entries.
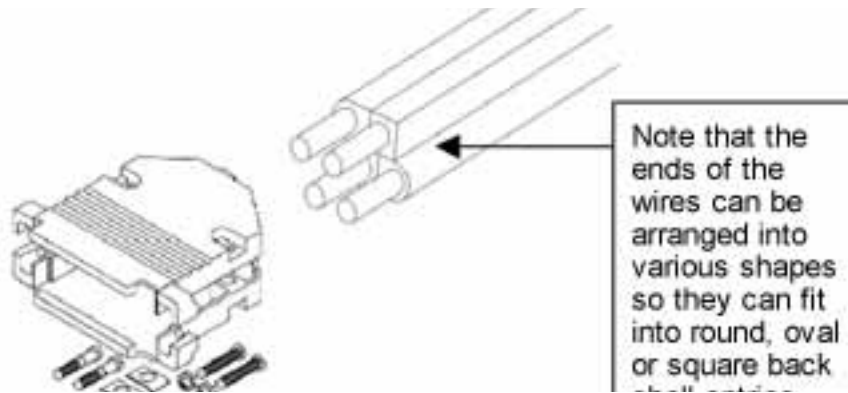
7. Terminate the wire as required by the connector type.

INDEX

INDEX