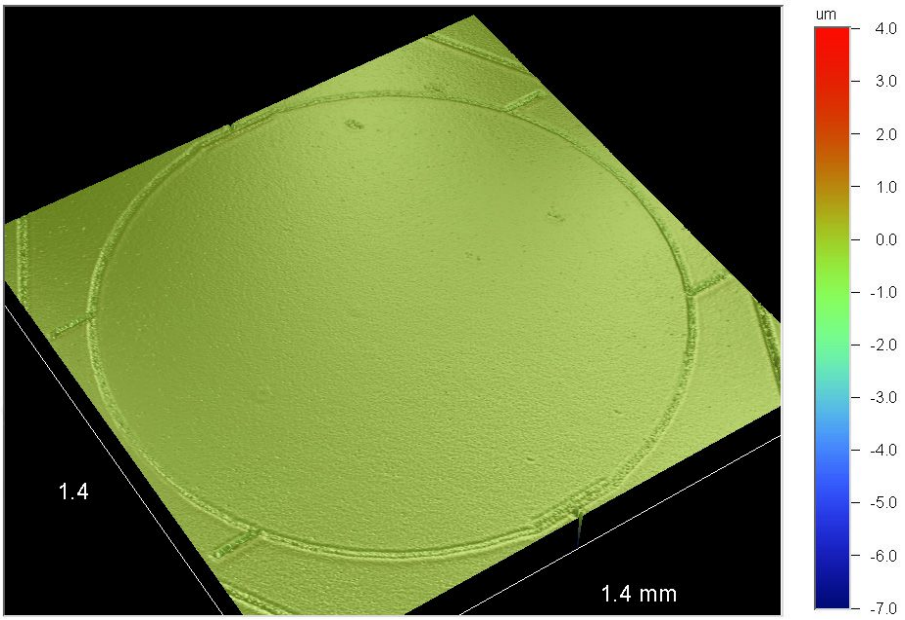
TM MEMS Tilt Mirror
Driving Board with Interface



Revision 1.16

August 2014

TABLE OF CONTENTS

1 In This Manual

This manual describes the optical, electrical and mechanical specifications of the TM MEMS Tilt mirror Driving Board with interfaces. It additionally details the communication protocol required to drive the device.

2 Description

sercalo's MEMS Driver Board is used together with TM series MEMS mirrors for precise control of the mirror pointing angle. The angle is set using electrostatic actuation, provided by a 12-bit digital-to-analog converter (DAC). To avoid optical feedback loops, the micro-mirror is designed to minimize effects such as drift, hysteresis and temperature-dependent performance.

sercalo's MEMS Driver Board is composed of an optical system and an electrical driver interface (see Figure 1). The device can store in the internal flash memory a user-defined look-up table of characteristic points. During operations, user can drive the MEMS mirror either to a directly defined position, or to one of the previously stored points. The device can be independently controlled by a UART interface (with TTL voltage levels) or an SMBus/I^2C interface.

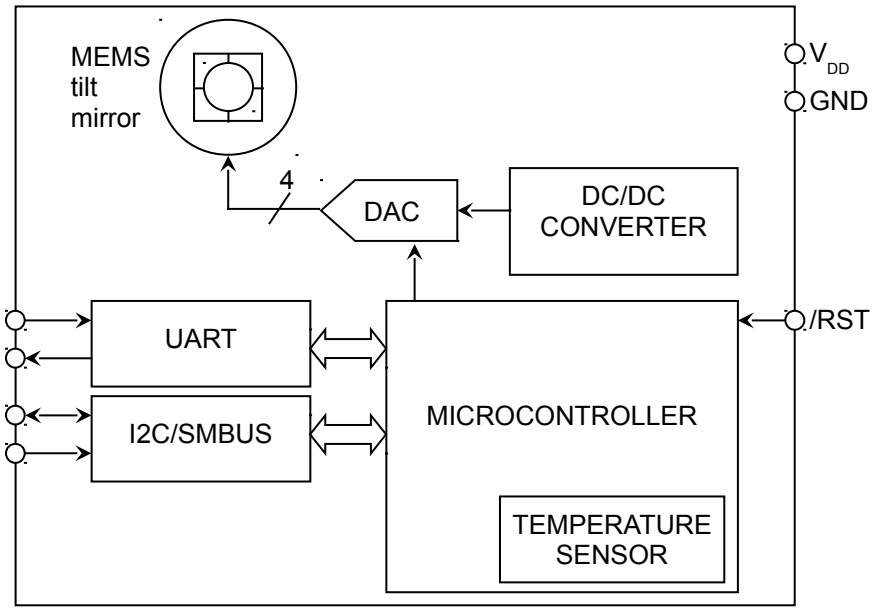


Figure 1 – Functional block diagram

3 Product Dimensions

Figure 2 depicts the devices layouts, view from pin side. All dimensions are shown in millimeters.

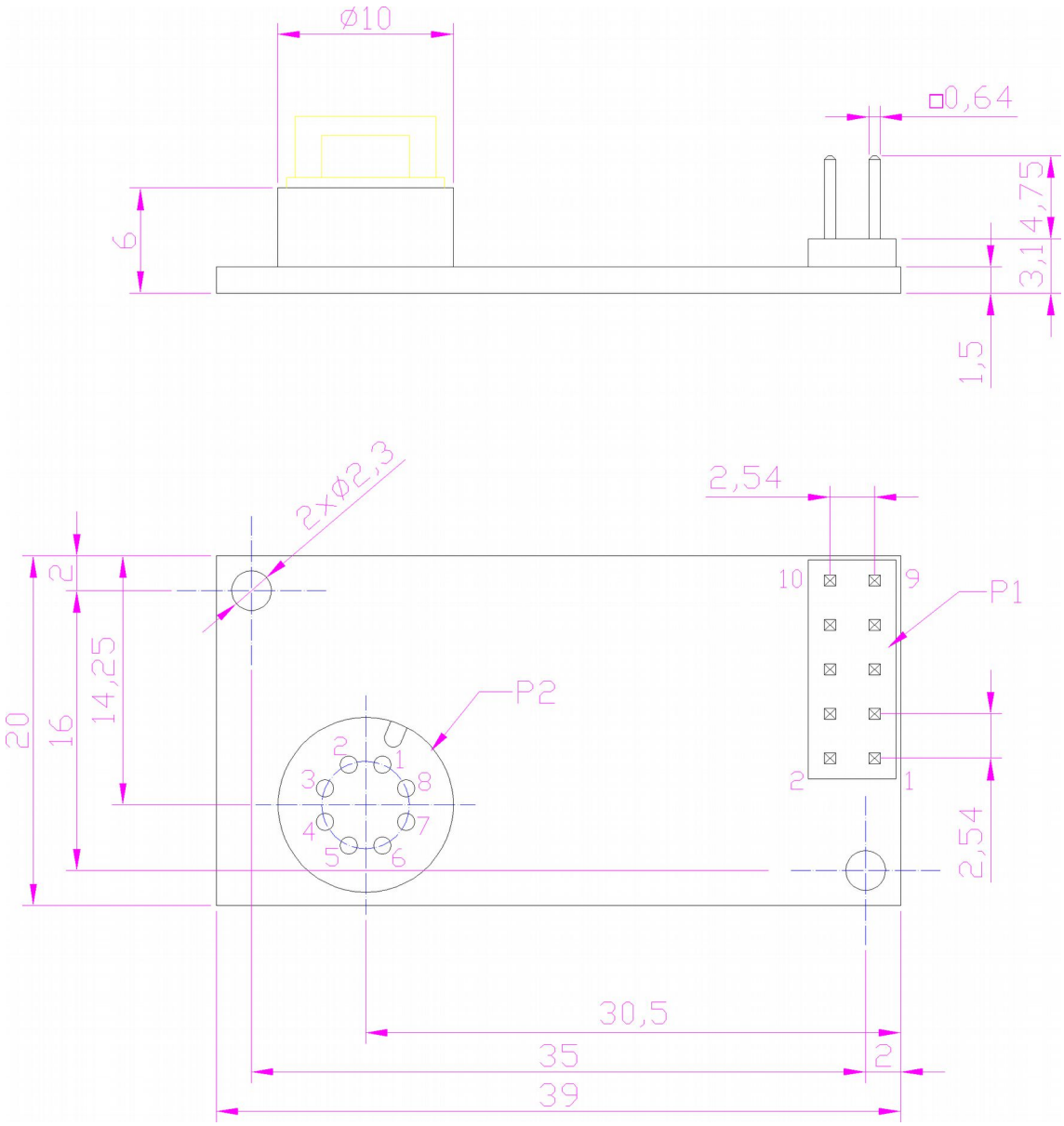


Figure 2 – Device layout

4 Connector Pin Assignments

The pin assignments for the connectors are detailed in Table 1 and Table 2. Please note that *UART TX* signal carries data from the product, while *UART RX* signal carries data to the product. Let reserved pins unconnected.

Pin number	Description
1	Ground (GND)
2	Supply voltage (V_{DD})
3	Reserved
4	UART TX data
5	Reserved
6	UART RX data
7	System reset (/RST)
8	SMBus/I^2C SDA
9	SMBus/I^2C SCL
10	Ground (GND)

Table 1 – Connector P1 pin assignments

Pin number	Description
1	Ground (GND)
2	Axis beta -
3	Axis alpha -
4	Axis beta +
5	Axis alpha +
6	Not connected
7	Not connected
8	Not connected

Table 2 – Connector P2 pin assignments

5 Absolute Maximum Ratings

Applicable absolute maximum ratings for the full operating temperature range without causing irreversible damage to the device are listed in Table 3.

Parameter	Unit	Min	Typ	Max
Maximum supply voltage V_{DD}	V	-	-	5.5
Voltage on any I/O pin with respect to GND	V	-0.3	-	5.5
Maximum supply current	mA	-	-	300

Table 3 – Absolute maximum ratings

6 Using the UART Interface

6.1 Command syntax

Commands consist of ASCII characters strings having the following structure:

CMD <param1> <…> [opt_param1] […]↵

where:

- **CMD** is the command
- **<…>** the angle brackets indicate that the enclosed parameter is *mandatory*
- **[…]** the square brackets indicate that the enclosed parameter is *optional*
- **↵** is the string terminator (end of line).

The following conventions are adopted:

- Commands can be written either in upper-case or lower-case characters.
- Commands and parameters are separated by one or more spaces (ASCII 0x20).
- The system recognizes as end of line any of the forms: LF (Line feed, '\n', ASCII 0x0A) or CR (Carriage return, '\r', ASCII 0x0D), or CR followed by LF (CR+LF, 0x0D 0x0A). Replies always end with the sequence CR+LF.
- The device always replies to commands. The reply can be a command-dependent acknowledgment or an error message.
- Error messages always start with ERR, followed by a space and context-dependent additional data.
- The user should not attempt to send the device a new command until it has completed the current command.

6.2 Configuring the UART interface

Table 4 lists the requirements for the UART settings. The default baud rate (after power on or reset) is 9600 baud which remains in effect otherwise changed by the user. If a Windows-based PC and HyperTerminal are used, the suggested configuration is displayed in Figure 3.

Parameter	Factory setting
Transmission rate [bps]	9600
Data bits	8
Parity	none
Stop bits	1
Flow control	none

Table 4 – RS-232 settings after power-on or reset

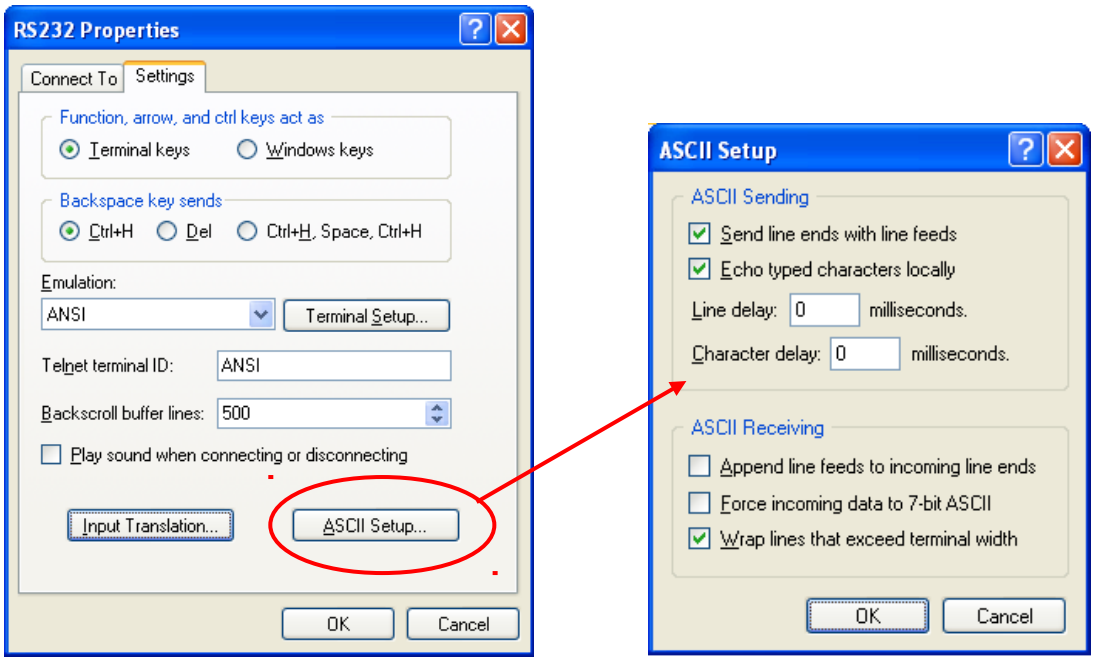


Figure 3 – HyperTerminal RS-232 properties and ASCII setup.

7 Using the SMBus/I²C Interface

The device is equipped with an SMBus serial I/O peripheral that is compliant with both the System Management Bus Specification and the I²C-Bus Specification. SMBus is a trademark of Intel; I²C is a trademark of Phillips Semiconductor.

SDA and SCL I/O pins are compatible with power supplies as low as 3.3V and as high as 5.0V. External pull-up resistors are required for full-speed operation.

The device is provided with a factory-programmed address of 0xFE. User can modify the default address using the dedicated command (see Chapter 6.2 for details). The new address is stored in the internal flash memory and is preserved during reset and power-off.

Each transmission is terminated with a Packet Error Code (PEC) byte, compliant to SMBus specification version 2.0 and 1.1. The byte is calculated over the entire message including the address and read/write bit (see Chapter 7.5 for details).

Please note that the device, after a successful write sequence, delay any further transmission by holding SCK low until the command is executed.

7.1 Send commands to device

A write sequence consists of the following elements (see Figure 4):

1. A *start* condition: it consists of a falling edge on SDA while SCL is high.
2. The *address byte*, i.e. 7 bits corresponding to the device address and the R/W bit set as “write” (the R/W bit is set to logic 1 to indicate a “read” operation and cleared to logic 0 to indicate a “write” operation).
3. A *command code*: refer to Chapter 6.2 for a list of available command codes.
4. The *parameter(s) length*, i.e. the number *n* of bytes composing the following optional parameters. This field is mandatory: if no parameters are required, set it to 0x00.

5. The *optional parameters*: refer to Chapter 6.2 for a detailed explanation.
6. *CRC-8.*
7. A *stop* condition: it consists of a rising edge on SDA while SCL is high.

The device replies to each received byte with an “ACK” as a confirmation. An ACK consists of a low level on SDA sampled when SCL is high. If the ACK signal is not received, master must abort the transfer.

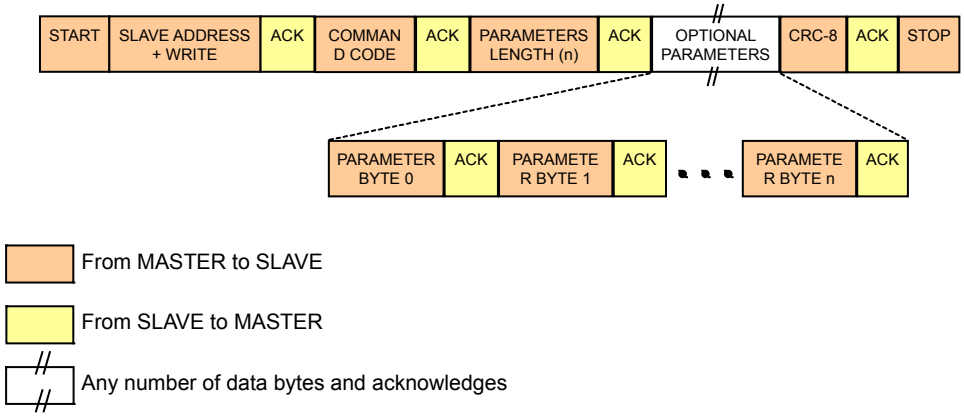


Figure 4 – Typical successful write (master to slave)

7.2 Receiving data from device

After a successful transmission, as detailed in paragraph 7.1, device always has a reply that can be optionally read by master. A read sequence for a successfully executed command consists of the following elements (see Figure 5):

1. A *start* condition: it consists of a falling edge on SDA while SCL is high.
2. The *address byte*, i.e. 7 bits corresponding to the device address and the R/W bit set as “read” (the R/W bit is set to logic 1 to indicate a “read” operation and cleared to logic 0 to indicate a “write” operation).
3. A *command code*: it is the last executed command, which this reply refers.
4. The *parameter(s) length*, i.e. the number *n* of bytes composing the following optional parameters. If no parameters will follow, this field is set to 0x00.
5. The *optional parameters*: refer to Chapter 6.2 for a detailed explanation.
6. *CRC-8*.
7. A *stop* condition: it consists of a rising edge on SDA while SCL is high.

The master must replies to each received byte with an “ACK” as a confirmation. An ACK consists of a low level on SDA sampled when SCL is high. Only the last byte must be confirmed with a “NACK”. A NACK consists of a high level on SDA sampled when SCL is high.

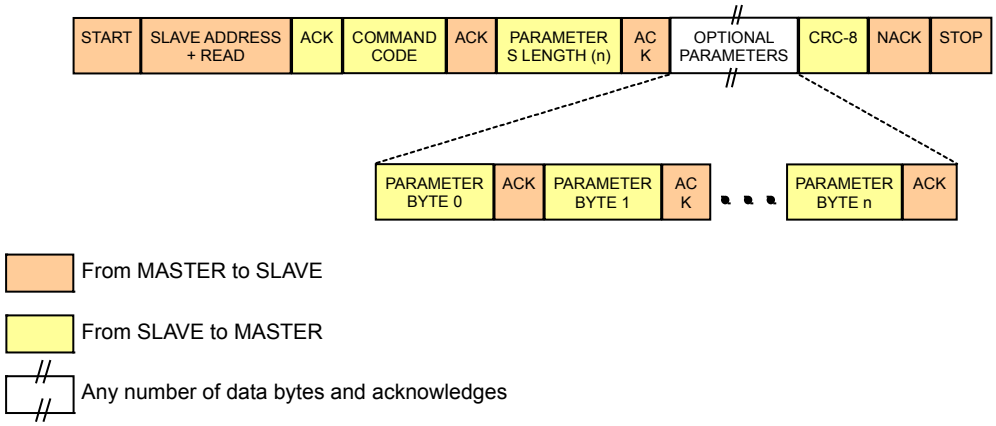


Figure 5 – Typical successful read (slave to master)

7.3 Receiving an error from device

If the previous command returns an error, it can be optionally read and detect by the master. A read sequence for a command returning an error consists in the following elements:

1. A *start* condition: it consists of a falling edge on SDA while SCL is high.
2. The *address byte*, i.e. 7 bits corresponding to the device address and the R/W bit set as “read” (the R/W bit is set to logic 1 to indicate a “read” operation and cleared to logic 0 to indicate a “write” operation).
3. A *command code* plus the value 0x80.
4. The *error number*: see Chapter 9 for a list of error codes.
5. *CRC-8*.
6. A *stop* condition: it consists of a rising edge on SDA while SCL is high.

The master must replies to each received byte with an “ACK” as a confirmation. An ACK consists of a low level on SDA sampled when SCL is high. Only the last byte must be confirmed with a “NACK”. A NACK consists of a high level on SDA sampled when SCL is high.

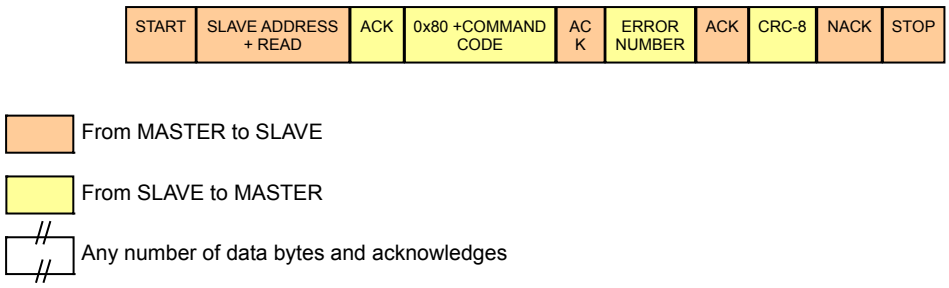


Figure 6 – Device returns an error (slave to master)

7.4 Hold master communication sequence

If the device is addressed during the execution of a command, it pulls down the SCL line in order to force the master into a wait state. By releasing the SCL line, the device indicates that

internal processing is terminated and that transmission may be continued. The maximum duration for command execution can vary depending on the command.

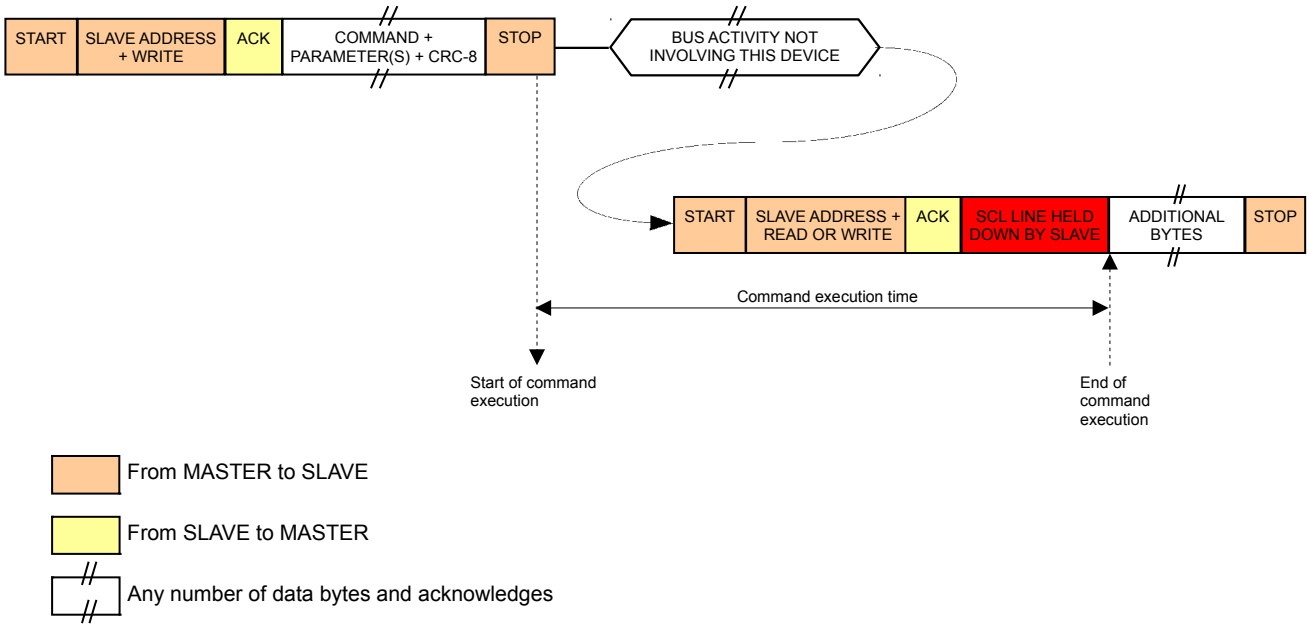


Figure 7 – Communication sequence with SCL line held down until command is executed

7.5 Example of CRC-8 generation

The SMBus/I²C uses a PEC (packet error code) compliant to SMBus specification version 1.1 and 2.0. The polynomial used is x^8+x^2+x+1 i.e. the CRC-8-ATM HEC algorithm, initialized to zero. An example of a C function that performs the calculation is presented below.

```
unsigned char crc8(unsigned char MsgLen, const unsigned char *Msg){
/* Lookup table for CRC codes
// CRC-8, poly = x^8 + x^2 + x^1 + x^0, init = 0 */

static unsigned char crc8_table[256] = {
   0x00, 0x07, 0x0E, 0x09, 0x1C, 0x1B, 0x12, 0x15, 0x38, 0x3F, 0x36, 0x31,
   0x24, 0x23, 0x2A, 0x2D, 0x70, 0x77, 0x7E, 0x79, 0x6C, 0x6B, 0x62, 0x65,
   0x48, 0x4F, 0x46, 0x41, 0x54, 0x53, 0x5A, 0x5D, 0xE0, 0xE7, 0xEE, 0xE9,
   0xFC, 0xFB, 0xF2, 0xF5, 0xD8, 0xDF, 0xD6, 0xD1, 0xC4, 0xC3, 0xCA, 0xCD,
   0x90, 0x97, 0x9E, 0x99, 0x8C, 0x8B, 0x82, 0x85, 0xA8, 0xAF, 0xA6, 0xA1,
   0xB4, 0xB3, 0xBA, 0xBD, 0xC7, 0xC0, 0xC9, 0xCE, 0xDB, 0xDC, 0xD5, 0xD2,
   0xFF, 0xF8, 0xF1, 0xF6, 0xE3, 0xE4, 0xED, 0xEA, 0xB7, 0xB0, 0xB9, 0xBE,
   0xAB, 0xAC, 0xA5, 0xA2, 0x8F, 0x88, 0x81, 0x86, 0x93, 0x94, 0x9D, 0x9A,
   0x27, 0x20, 0x29, 0x2E, 0x3B, 0x3C, 0x35, 0x32, 0x1F, 0x18, 0x11, 0x16,
   0x03, 0x04, 0x0D, 0x0A, 0x57, 0x50, 0x59, 0x5E, 0x4B, 0x4C, 0x45, 0x42,
   0x6F, 0x68, 0x61, 0x66, 0x73, 0x74, 0x7D, 0x7A, 0x89, 0x8E, 0x87, 0x80,
   0x95, 0x92, 0x9B, 0x9C, 0xB1, 0xB6, 0xBF, 0xB8, 0xAD, 0xAA, 0xA3, 0xA4,
   0xF9, 0xFE, 0xF7, 0xF0, 0xE5, 0xE2, 0xEB, 0xEC, 0xC1, 0xC6, 0xCF, 0xC8,
   0xDD, 0xDA, 0xD3, 0xD4, 0x69, 0x6E, 0x67, 0x60, 0x75, 0x72, 0x7B, 0x7C,
   0x51, 0x56, 0x5F, 0x58, 0x4D, 0x4A, 0x43, 0x44, 0x19, 0x1E, 0x17, 0x10,
   0x05, 0x02, 0x0B, 0x0C, 0x21, 0x26, 0x2F, 0x28, 0x3D, 0x3A, 0x33, 0x34,
   0x4E, 0x49, 0x40, 0x47, 0x52, 0x55, 0x5C, 0x5B, 0x76, 0x71, 0x78, 0x7F,
   0x6A, 0x6D, 0x64, 0x63, 0x3E, 0x39, 0x30, 0x37, 0x22, 0x25, 0x2C, 0x2B,
   0x06, 0x01, 0x08, 0x0F, 0x1A, 0x1D, 0x14, 0x13, 0xAE, 0xA9, 0xA0, 0xA7,
   0xB2, 0xB5, 0xBC, 0xBB, 0x96, 0x91, 0x98, 0x9F, 0x8A, 0x8D, 0x84, 0x83,
   0xDE, 0xD9, 0xD0, 0xD7, 0xC2, 0xC5, 0xCC, 0xCB, 0xE6, 0xE1, 0xE8, 0xEF,
```

```
    0xFA, 0xFD, 0xF4, 0xF3 };

register unsigned char crcVal = 0x00;

while (MsgLen-- > 0) {
    crcVal = crc_table[crcVal ^ *Msg++];
    }

return crcVal;
}
```

8 Command Set

The available commands are listed in Table 5 and detailed in the following subsections. Use binary command formatting for SMBus/I²C bus, ASCII command formatting for UART.

Command		Description
Binary	**ASCII**	
0x01	ID	Returns the equipment identification
0x02	RST	Resets the device
0x03	POW	Returns or changes the power mode of the device
0x04	ERM	Returns or changes the error returning mode
0x08	TMP	Returns the temperature of the microcontroller
0x10	UART	Returns or changes the baud rate of the UART
0x20	IIC	Returns or changes the address for SMBus/I²C
0x50	SET	Sets the network configuration
0x51	POS	Returns the network configuration
0x52	CHSET	Sets the specified user-defined channel
0x53	CHGET	Returns the position of the specified user-defined channel
0x54	CHMOD	Modifies the specified user-defined channel
0x64	MOD	Changes the mirror control mode.

Table 5 – List of available commands

8.1 ID – Returns the equipment identification

Returns the system identification, which consists of the model, the serial number and the firmware version. The three fields are separated by pipe characters '|' (ASCII 0x7C).

Parameters:			
	PRODUCT	(string)	product model
	S/N	(string)	serial number
	FW_REV	(string)	firmware version

Use with UART:

Syntax: **ID↵**

Reply on success: **ID <PRODUCT>|<S/N>|<FW_REV>**

Example:
```
ID↵
ID TF|2010-20-002|1.2
```

Use with SMBus/I²C bus:

```
(T) ADDR+W,0x01,0x00,CRC8
(R) ADDR+R,0x01,STR_LEN,PRODUCT,'|',S/N,'|',FW_REV,CRC8
```

8.2 RST – Resets the board

Resets the system.

Parameters: none

Use with UART:

Syntax: **RST↵**

Reply on success: **RST**

Example: **RST↵**
RST

Use with SMBus/I²C bus:

(T) ADDR+W,0x02,0x00,CRC8

(R) ADDR+R,0x02,0x00,CRC8

8.3 POW – Returns or changes the power mode of the device

If called without parameters, returns 0 if the device is in low power mode or 1 otherwise. If **POW_MODE** is entered, switch to low power mode if its value is 0, or switch to normal power mode if the value is 1. The MEMS control only works when the power mode is set to 1. This parameter returns to the default value after any reset or power-on.

Parameters: **POW_MODE** (char, 0 or 1) power status

Default: **0**

Use with UART:

Syntax: **POW [POW_MODE]↵**

Reply on success: **POW <POW_MODE>**

Example: query the current power mode:
POW↵
POW 0
change the current power mode:
POW 0↵
POW 0

Use with SMBus/I²C bus:

query the current power mode:

(T) ADDR+W,0x03,0x00,CRC8

(R) ADDR+R,0x03,0x01,POW_MODE,CRC8

change the current power mode:

(T) ADDR+W,0x03,0x01,POW_MODE,CRC8

(R) ADDR+R,0x03,0x01,POW_MODE,CRC8

8.4 ERM – Returns or changes the error returning mode

If called without parameters, returns 0 if the errors are identified with integer numbers or 1 if the errors are returned in plain text. If **ERR_MODE** is entered, switch to verbose mode if its value is 1, or switch to number mode if the value is 0. This parameter returns to the default value after any reset or power-on.

Parameters: **ERR_MODE** (char, 0 or 1) error returning mode

Default: **1**

Use with UART:

Syntax: **ERM [ERR_MODE]↵**

Reply on success: **ERM <ERR_MODE>**

Example: query the current error returning mode:
ERM↵
ERM 0
change the current error returning mode:
ERM 0↵
ERM 0

Use with SMBus/I²C bus:

query the current error returning mode:

(T) ADDR+W,0x04,0x00,CRC8

(R) ADDR+R,0x04,0x01,ERR_MODE,CRC8

change the current error returning mode:

(T) ADDR+W,0x04,0x01,ERR_MODE,CRC8

(R) ADDR+R,0x04,0x01,ERR_MODE,CRC8

8.5 TMP – Returns the temperature of the microcontroller

Returns the temperature of the microcontroller unit, expressed in Celsius degrees.

Parameters: **TEMP** (float) temperature

Use with UART:

Syntax: **TMP↵**

Reply on success: **TMP <TEMP>**

Example: **TMP↵**
TMP 38.7

Use with SMBus/I²C bus:

(T) ADDR+W,0x08,0x00,CRC8

(R) ADDR+R,0x08,0x04,TEMP,CRC8

8.6 UART – Changes the speed of the UART interface

If called without parameters, returns the speed of the UART port. If **UART_BAUD** is entered, changes the baud rate of the UART accordingly to Table 6. This parameter returns to the default value after any reset or power-on.

Value	Baud rate
0	9600
1	19200
2	38400
3	57600
4	115200

Table 6 – List of available baud rates

Parameters: **UART_BAUD** (char, 0 to 4) port speed

Default: **0**

Use with UART:

Syntax: **UART [UART_BAUD]↵**

Reply on success: **UART <UART_BAUD>**

Example: query the current baud rate:
UART↵
UART 3
change the baud rate:
UART 2↵
UART 2

Use with SMBus/I²C bus:

query the current baud rate:

(T) ADDR+W,0x10,0x00,CRC8

(R) ADDR+R,0x10,0x01,UART_BAUD,CRC8

change the current baud rate:

(T) ADDR+W,0x10,0x01,UART_BAUD,CRC8

(R) ADDR+R,0x10,0x01,UART_BAUD,CRC8

8.7 IIC – Returns or changes the address for SMBus/I²C

If called without parameters, this command returns the current address of the device, as a decimal number. If **IIC_ADDR** is entered, changes the address of the device. The new value is stored in the internal flash memory and is preserved during reset and power-off.

Parameters: **IIC_ADDR** (char, 0 to 255) SMBus/I²C address

Default: **254**

Use with UART:

Syntax: **IIC [IIC_ADDR]↵**

Reply on success: **IIC <IIC_ADDR>**

Example: query the current address:
IIC↵
IIC 254
modify the address:
IIC 2↵
IIC 2

Use with SMBus/I²C bus:

query the current address:

(T) ADDR+W,0x20,0x00,CRC8

(R) ADDR+R,0x20,0x01,IIC_ADDR,CRC8

modify the address:

(T) ADDR+W,0x20,0x01,IIC_ADDR,CRC8

(R) ADDR+R,0x20,0x01,IIC_ADDR,CRC8

8.8 SET – Moves the MEMS mirror to the specified position

Moves the MEMS mirror to the (α,β) coordinates. The signs of alpha and beta define the tilt direction of the micro-mirror. The absolute value is proportional to the full scale of the digital to analog converter. This command is not available when the board is in power saving mode (see command POW).

Parameters:	**ALPHA**	(integer, -4095 to 4095)	α-axis position
	BETA	(integer, -4095 to 4095)	β-axis position

Use with UART:

Syntax: **SET <ALPHA> <BETA>↵**

Reply on success: **SET <ALPHA> <BETA>**

Example: **SET -2000 500↵**
SET -2000 500

Use with SMBus/I²C bus:

(T) ADDR+W,0x50,0x04,ALPHA,BETA,CRC8

(R) ADDR+R,0x50,0x04,ALPHA,BETA,CRC8

8.9 POS – Returns the current position of the MEMS mirror

Returns the actual (α,β) coordinates of the mirror. The absolute value is proportional to the full scale of the digital to analog converter. This command is not available when the board is in power saving mode (see command POW).

Parameters:	**ALPHA**	(integer, -4095 to 4095)	α-axis position
	BETA	(integer, -4095 to 4095)	β-axis position

Use with UART:

Syntax: **POS↵**

Reply on success: **POS <ALPHA> <BETA>**

Example:
```
POS↵
POS -2000 500
```

Use with SMBus/I²C bus:

(T) ADDR+W,0x51,0x00,CRC8

(R) ADDR+R,0x51,0x04,ALPHA,BETA,CRC8

8.10 CHSET – Sets the specified user-defined channel

Moves the internal MEMS mirror to the coordinates stored in the memory location **P**. If **P** is empty, an error message is returned. This command is not available when the board is in power saving mode (see command POW).

Parameters:	**P**	(integer, 0 to 127)	memory location

Use with UART:

Syntax: **CHSET <P>↵**

Reply on success: **CHSET <P>**

Example:
```
CHSET 1↵
CHSET 1
```

Use with SMBus/I²C bus:

(T) ADDR+W,0x52,0x02,P,CRC8

(R) ADDR+R,0x52,0x02,P,CRC8

8.11 CHGET – Returns the content of the specified user-defined channel

Returns the coordinates stored in memory location **P**. The signs of **ALPHA** and **BETA** define the tilt direction of the micro-mirror. The absolute value is proportional to the full scale of the digital to analog converter.

Parameters:	**P**	(integer, 0 to 127)	memory location
	ALPHA	(integer, -4095 to 4095)	α-axis position
	BETA	(integer, -4095 to 4095)	β-axis position

Use with UART:

Syntax: **CHGET <P>↵**

Reply on success: **CHGET <P> <ALPHA> <BETA>↵**

Example: **CHGET 1↵**
CHGET 1 45 -1050

Use with SMBus/I²C bus:

(T) ADDR+W,0x53,0x02,P,CRC8

(R) ADDR+R,0x53,0x06,P,ALPHA,BETA,CRC8

8.12 CHMOD – Modifies the specified user-defined channel

Saves in the memory location **P** the mirror position defined by **ALPHA** and **BETA**. The signs of **ALPHA** and **BETA** define the tilt direction of the micro-mirror. The absolute value is proportional to the full scale of the digital to analog converter.

The optional **DELAY** parameter is only used in combination with the "pattern mode" (see command MOD) and represents the amount of milliseconds to be waited until setting the next point of the pattern. A **DELAY** value of 255 marks the final point of the pattern.

Warning: the overwritten points will be permanently lost!

Parameters:			
	P	(integer, 0 to 127)	memory location
	ALPHA	(integer, -4095 to 4095)	α-axis position
	BETA	(integer, -4095 to 4095)	β-axis position
	DELAY	(integer, 0 to 255)	delay

Use with UART:

Syntax: **CHMOD <P> <ALPHA> <BETA> [DELAY]↵**

Reply on success: **CHMOD <P> <ALPHA> <BETA> [DELAY]↵**

Example: **CHMOD 1 45 -1050↵**
CHMOD 1 45 -1050

Use with SMBus/I²C bus:

(T) ADDR+W,0x54,0x06,P,ALPHA,BETA,CRC8

(R) ADDR+R,0x54,0x06,P,ALPHA,BETA,CRC8

8.13 MOD – Changes the mirror control mode

If called without parameters, returns the mirror control mode. If **CMODE** is entered, changes the mirror control mode in accordance with the list below. This parameter returns to the default value after any reset or power-on.

0: normal mode, drive the mirror using the command "SET" or "CHSET"

1: scan the mirror along the alpha axis

2: scan the mirror along the beta axis

3: scan the mirror along both axes

4: pattern mode

When in *pattern mode*, the micro-mirror will cyclically set the coordinate points previously stored in the flash memory with the command CHMOD. A pattern is composed by an arbitrary number of points, starting from memory location 0 and ending with a dummy point having the **DELAY** parameter equal to 255.

This parameter returns to the default value after any reset or power-on.

Parameters: **CMODE** (integer, 0 to 4) control mode

Default: **0**

Use with UART:

Syntax: **MOD <CMODE>↵**

Reply on success: **MOD <CMODE>**

Example: **MOD 4↵**
MOD 4

Use with SMBus/I²C bus:

(T) ADDR+W,0x64,0x01,P,CRC8

(R) ADDR+R,0x64,0x01,P,CRC8

9 Error Codes

The following table gives information about error codes used with sercalo's devices.

Error number	Description
1	syntax error
2	CRC error
3	invalid parameter(s)
4	command unknown
5	timeout
6	buffer overrun
7	invalid IP/subnet mask combination
8	device is in idle mode
9	memory location is empty
10	status unknown
11	communication error

Table 7 – Overview of error codes

10 Optical Specifications

	Unit	Min	Typ	Max
Surface finish	-		Gold	
Reflectivity (800-2000 nm)	%		95	
Resonance frequency	Hz	100		
Durability	cycles	No wear out		

11 Electrical Specifications

	Unit	Min	Typ	Max
Supply voltage (Vdd)	V	4.75	5	5.25
Power consumption (operating mode)	W			1
Power consumption (idle mode)	W		0.2	
UART speed	baud	9600		115200
UART Logic Level 0	V		0	0.6
UART Logic Level 1	V	2.8	5.0	
SMBus/I^2C Logic Level 0	V		0	0.6
SMBus/I^2C Logic Level 1	V	2.8	5.0	
SMBus/I^2C bus speed	kbps	100		400
Reset inactive voltage[1]	V	2.4	5	
Reset active voltage	V		0	0.9
Reset pulse duration	µs	15		

12 Package Specifications

	Unit	Min	Typ	Max
Operation temperature	°C	0		70
Storage temperature	°C	-40		70
Dimensions	mm	39 x 20 x 10		
Weight	g	50		

[1] Through onboard pull-up resistor

13 Contact Information

SERCALO MICROTECHNOLOGY LTD

Landstrasse 151, 9494 Schaan

Principality of Liechtenstein

Tel.: +423 237 57 97

Fax: +423 237 57 48

http://www.sercalo.com

Email: info@sercalo.com