www.ams.com

Documentation
FW Interface - AS73210-AS89010-C3

Version 2.0

Project #: 120898-25
Status: certified

ams Sensors Germany GmbH docu15412e V2.0 - 2017-04-01

1 INTRODUCTION

1.1 Development Objective

The described firmware is used to control the AS73210-AS89010-C3.

1.2 Abbreviations and Synonyms

CL	Client
CO	Contractor
CB	Circuit board
PCB	Printed Circuit Board
TBD	To be defined
POR	Power on reset
WRGB	White-Red-Green-Blue
MCDC	**M**ulti-**C**hannel Programmable analog current integrator with **d**igital **c**onverter

2 DESCRIPTION OF FIRMWARE MODULES

2.1 General Description

The firmware uses the real-time operating system RTX-ARM of the company KEIL.

The following functions have been implemented to meet the requirements:

1. System initialization
2. Controlling communication with host system
    a.) I/O controller via the communication interface
    b.) Command interpreter
3. Register management
    c.) Register initialization after RESET
    d.) Read/Write of registers
    e.) Field monitoring of individual registers
    f.) Secured retentive storage of individual registers
4. Color measurement
    g.) Parameterization of the color measurement
    h.) Activate color measurement
    i.) Single step mode

Figure 1: Demonstrated functions, data streams and connection to the computer (master function)

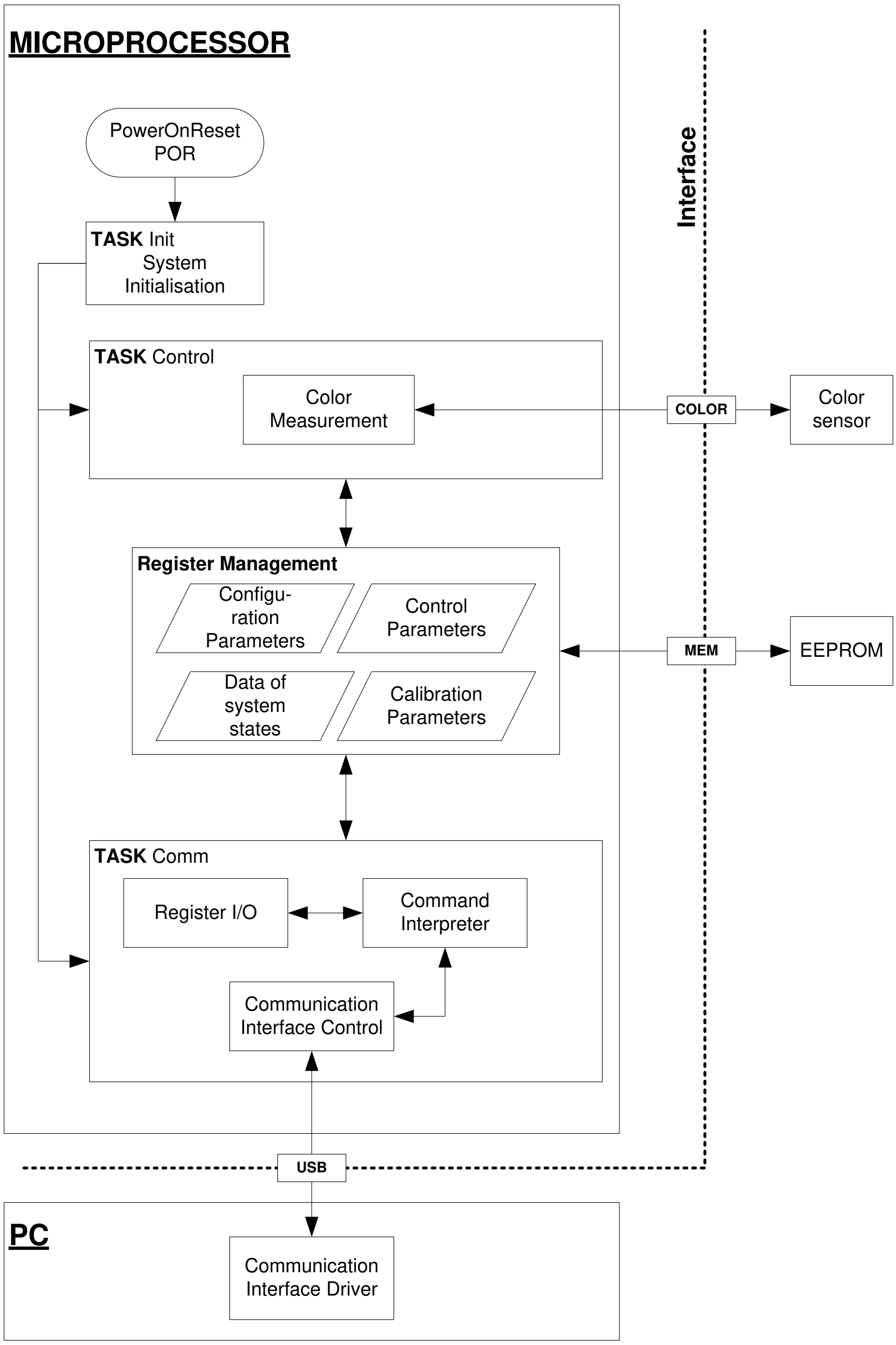


Figure 1 shows the block diagram visualizing the demonstrated functions and data streams. Furthermore, the connection to the computer (master function) is shown.

The startup behavior and the individual processes described in the following sections are largely parameterized. This means that the parameters can be read and written using commands from the master (Computer). We distinguish two types of parameters.

1. Temporary parameters: They can be modified at run time, but after POR they return to a set of default values.
2. Remnant stored parameters: These parameters are stored on an EEPROM. Therefore, the values remain even after a POR.

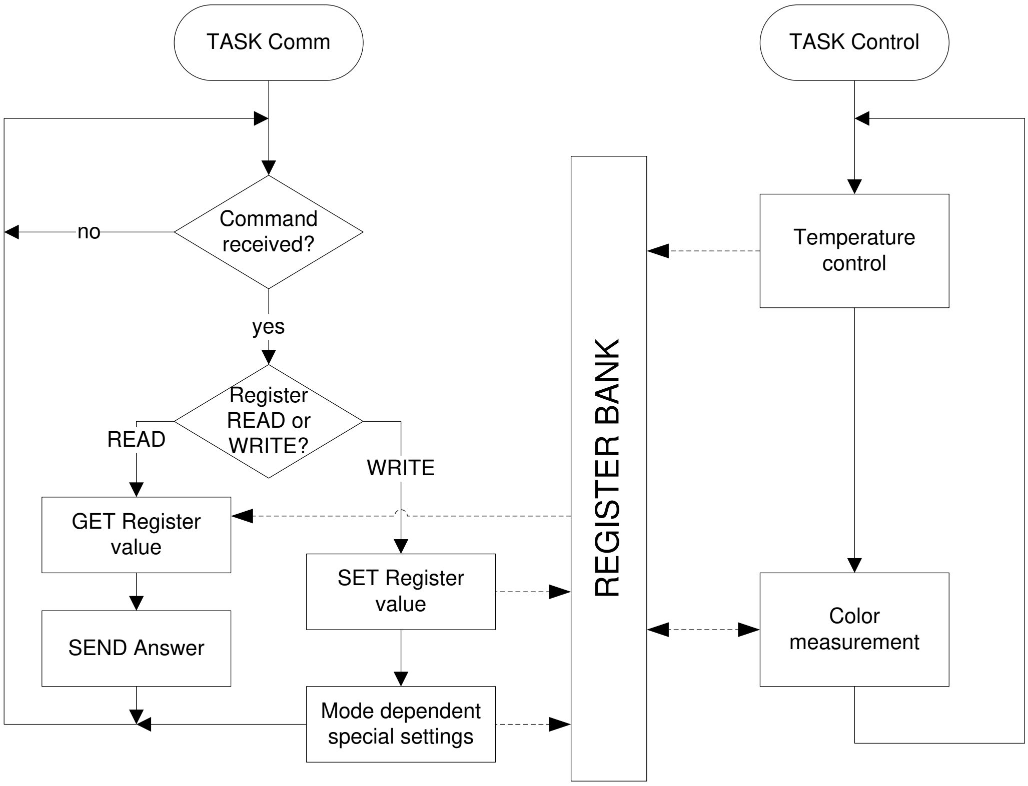
The parameters represent a set of software registers and are stored in the so-called register bank (compare 4).

2.2 Description of the Main Components

2.2.1 General sequence control

The following image shows the main components of the sequence control.

Figure 2: Block diagram of the sequence control



The sequence control is implemented via the tasks **Comm** and **Control**. Both tasks work with read and write options with the register bank.

Comm is used for the following tasks

1. Reception and evaluation of commands from the master.
2. For read requests: Send the response to the master.
3. On write command: Set of the transferred register. Depending on the mode currently set, additional registry settings are changed if required.

The Task **Control** cyclically operates the following steps:

1. Temperature measurement
2. Color measurement

Whether only one cycle is performed (STEP), or the sequence in is running in continuous operation (RUN), is determined by the **CycleState** parameters.

2.2.2 Temperature measurement

A measurement is performed after the defined parameter time of **TempPeriod**. The value is stored in the **Temperature** register. The basis for the temperature values are in stored temperature calibration of **TempCalib**.

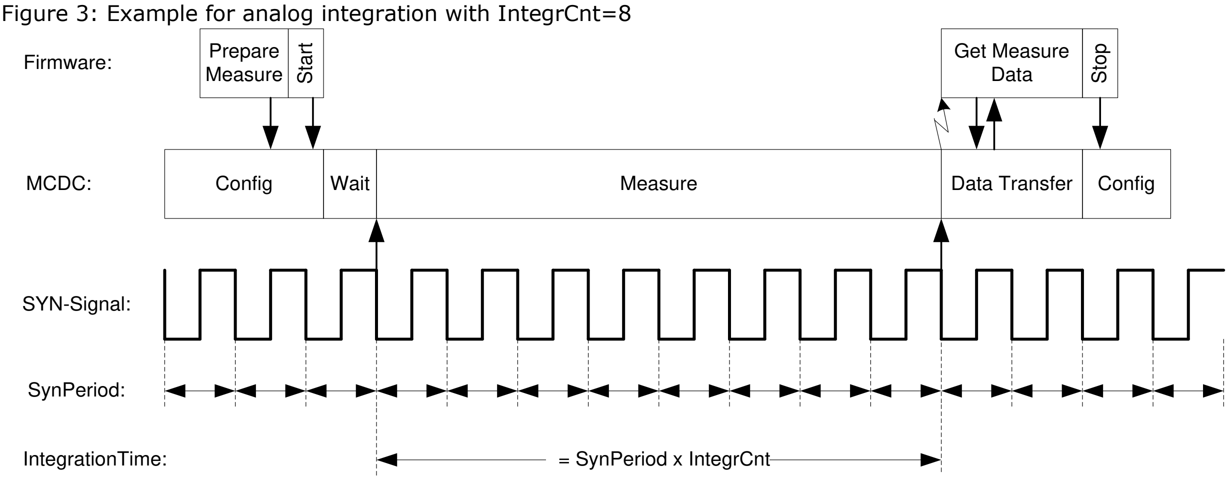
2.2.3 Color measurement

For color measurement, the color sensor and converter (AS73211 or AS73210 and AS89010) is activated via interface **COLOR**. The converter is solely used in SYND mode (compare [1]). The measured value is performed by analog integration. The integration process is controlled via the trigger signals **SYN**. The integration time is determined by the following parameters:

SynPeriod = Time period of the **SYN** signal,

IntegrCnt = Number of **SYN** periods that are used for integration.

Figure 3: Example for analog integration with IntegrCnt=8



The **SynPeriod** parameter is retentive. The **IntegrCnt** parameter is set to the retentive parameters of **IntegrCntReset** after POR.

The following parameters are used to determine the optimum operating range of the converter:

Ampl = Amplification settings,

Divider = Resolution settings.

Additional retentive stored RESET values exists for these parameters (**AmplReset** and **DividerReset**).

The color measurement values are stored in register **ActAdc** and can be read out there.

2.2.4 Register management

The register management organizes the read and write access of the parameters stored in the register bank.

Access to retentive parameters via the MEM interface. Write accesses to remanent stored parameters are only executed if a new value has been changed in comparison to the current value.

2.2.5 Communication with the Master

The communication with a Master is performed via the USB interface. Commands are used to access parameter settings, measurement and system states and the measured data. For testing purposes, additional actions can be performed.

2.3 Interfaces

2.3.1 COLOR: Addressing the Color Sensors

The control of the color sensor is performed via I²C interface. The data record shows [1] or [4]).

2.3.2 MEM: Addressing the EEPROM

Parameters to be permanently stored are managed on an EEPROM connected via I²C. Detailed information to control the EEPROM are to be shown in [2].

2.3.3 USB: Connection to the Master

The USB interface acts as a virtual COM port and enables to communicate with the Master.

3 COMMUNICATION PROTOCOL

3.1 General Information

The communication from the host system is performed exclusively by reading or writing registers. The following general rules apply:

- Each command sequence contains the information **read** or **write with response** or **write with no response**.
- Each command contains the address of the read / write register.
- When writing the correct number for the register argument must be used.
- Each answer contains a completion code.
- A register-based data sequence is returned after an error-free reading of the register content completion code.

3.2 Commando Build

The commands are ASCII strings. One starts with the command header consisting of command code and register ID code. The data to be written follows on write commands. The command is terminated with CR/LF.

Read commands and write commands with response be answered by the slave via ASCII string. First, an echo of the command code appears. It follows the completion code. For the case of an error-free read command the first register ID code follows, and then the data of the read register continues. The response is terminated with CR/LF.

The following table shows the general command and response structure:

Table 1: General command build

	ARG. #	MEANING	DATA TYPE	REG. WRITE WITH ANSWER	REG. READ	REG. WRITE WITHOUT ANSWER
COMMAND	1	Command code	BYTE	1	2	3
	2	RegisterID	USHORT	rid	rid	rid
	3	Register content	$DATATYPE_{rid}$	rw_1		rw_1
	…			…		…
	…			rw_{ran}		rw_{ran}
ANSWER	1	Command code	BYTE	1	2	
	2	Completion code	BYTE	cc	cc	
	3	RegisterID	USHORT		rid	
	4	Register content	$DATATYPE_{rid}$		rw_1	
	…				…	
					rw_{ran}	

Notes:

1. The individual arguments are separated by a SPACE. The row terminator is "\n\r".
2. All arguments are purely numeric ASCII strings in hexadecimal notation **without** a leading "0x". Leading zeros can be omitted.
3. Command or reply arguments of type FLOAT are transferred in accordance with its internal 32-bit representation, packed in a ULONG.
4. Command or reply arguments of type FLOAT are transferred in accordance with its internal 64-bit representation, packed in a ULONG.
5. The completion code the successful completion of the command via cc = 0x00. cc^{1}0 represents a warning or an error message. During error messages bit 7 is set (cc^{3}0x80); the command was not executed. A warning message indicates that a limited command execution took place.
6. The content of a register is either a single value or an array of values of a specified data type. Therefore, the argument quantity and data type of the register contents depend on respective registers.
7. In case of an error when reading out registers (cc> = 0x80) only the arguments 1-2 are transmitted-

3.3 Completion Codes

Each command response contains a completion code. This byte indicates:

- OK: Successful completion of the command, without restrictions

or

- Warning: Successful completion of the command, with restrictions

or

- Error: command not executed

The following table lists all defined completion codes:

Table 2: Completion codes

STATUS	COMPL. CODE cc	MEANING
OK	0x00	Command successfully executed.
Warning	0x01	Command executed. Register value is set to maximum.
	0x02	Command executed. Register value is set to minimum.
	0x03	Command executed. Register value changed.
	0x04	Command executed. Changed other register value.
	0x05	Register value unchanged.
	0x06	Register not supported
Error	0x81	Unsupported character in an argument
	0x82	Argument to large
	0x83	Unknown command
	0x84	Unsupported data quantity
	0x85	Unknown register address
	0x86	Register address missing
	0x87	Register is write protected

4 REGISTER DESCRIPTION

4.1 Register List

Table 3: Register description

Operation modes and status

ID	DATA TYPE	LEN.	REGISTER	R/W	SAVED	VALUE RANGE	DEFAULT	EXPLANATION
0x0011	BYTE	1	CycleState	r+w		0=STOP 1=STEP 2=RUN	RUN	Cycle status (Stop, single cycle, continuous operation)
0x0013	ULONG	3	State	r		Compare 4.2	0	System state: Idx0 = Error State (ErrorState) Idx1 and Idx2 = reserved

Configuration of the color measurement

ID	DATA TYPE	LEN.	REGISTER	R/W	SAVED	VALUE RANGE	DEFAULT	EXPLANATION
0x0015	BYTE	1	IntegrCntReset	r+w	x	1 … 255	10	RESET value of the color measurement integration counter
0x0016	BYTE	1	IntegrCnt	r+w		1 … 255	IntegrCntReset	Color measurement integration counter
0x0017	BYTE	1	AmplReset	r+w	x	0 … 4	2	RESET value for amplification of MCDC
0x0018	BYTE	1	AmplAmpl	r+w		0 … 4	AmplReset	Amplification of MCDC
0x0019	BYTE	1	DividerReset	r+w	x	1 | 2 | 4 | 8 | 16	1	RESET value of divider of MCDC
0x001a	BYTE	1	Divider	r+w		1 | 2 | 4 | 8 | 16	DividerReset	Divider of MCDC
0x011c	BYTE	1	SynPeriodReset	r+w	x	**1000 … 50000**	**1000**	SYN period [µsec]
0x001c	BYTE	1	SynPeriod	r+w		**1000 … 50000**	**SynPeriodReset**	RESET value of SYN period

Color values

ID	DATA TYPE	LEN.	REGISTER	R/W	SAVED	VALUE RANGE	DEFAULT	EXPLANATION
0x0040	FLOAT	6	ActAdc	r		Normalized values: 0.0 … 100.0 Raw values: 0.0 … 65535.0	0,0,0,0,0,0	Current ADC measurement values (Compare 4.3) Idx 0-2: normalized ADC values Special cases: Idx0 | 1 | 2 = -1 = too high Ampl Idx0 | 1 | 2 = -2 = to high integration time Idx0 = -3: there are no new readings Idx 3-5: ADC raw values Idx 0 and 3 for RED (X) Idx 1 and 4 for GREEN (Y) Idx 2 and 5 for BLUE (Z)

Board and firmware IDs

ID	DATA TYPE	LEN.	REGISTER	R/W	SAVED	VALUE RANGE	DEFAULT	EXPLANATION
0x00c0	USHORT	1	BoardDate	r+w	x		1	Manufacturing date of the board 0xYYWW with: YY = year WW = week
0x00c1	ULONG	1	SerNum	r+w	x		0	Serial number of the board
0x00c2	ULONG	1	FirmwareVersion	r				Firmware version 0x00XXYYZZ: XX = YY = Major Minor1 ZZ = Minor2
0x00c4	ULONG	1	BoardIdentHID	r			HID	Board ID (6-digit. Origin ID) = 090010
0x00c5	ULONG	1	BoardIdentTKE	r			TKE	Board ID 0x00XXYYZZ with: XX = 2 digit. = 17 YY = 1 digit. Construct. version = 2 ZZ = 2 digit. Extension = 25
0x00c60x00c6	BYTE	1	BoardIdentAEZ	r			AEZ	Board ID (2 digit. Change state)

Application-specific register

ID	DATA TYPE	LEN.	REGISTER	R/W	SAVED	VALUE RANGE	DEFAULT	EXPLANATION
0x01d0	SHORT	2	Temperature	r			0	Current Temperature Idx0 = AD value Idx1 = Temperature in °C
0x01d1	USHORT	1	TempPeriod	r+w	x		1	Time period for temperature measurement [sec]; =0 means: Temperature measurement turned off
0x01d2	SHORT	2	TempCalibData	r+w	x		1750, 500	Temperature calibration values Idx0 = AD value at 25 °C Idx1 = AD difference per °C, multiplied with 100
0x01d3	DOUBLE	9	SensCalibMx	r+w	x		0	Sensor calibration matrix
0x01d4	DOUBLE	3	SensCalibOffset	r+w	x		0	Dark current offset in photo current units for sensor calibration
0x01d5	SHORT	1	SensCalibTemp	r+w	x		0	Temperature at sensor calibration
0x01d6	BYTE	4	SensCalibConfig	r+w	x		0,0,0,0	Sensor configuration parameter at sensor calibration Idx0 = IntegrCnt Idx1 = Ampl Idx2 = Divider Idx3 = SynPeriod

Offset Compensation

ID	DATA TYPE	LEN.	REGISTER	R/W	SAVED	VALUE RANGE	DEFAULT	EXPLANATION
0x01e0	BYTE	1	OffsetCurrent	r+w		0 … 3	1	Value for MCDC OPTREG Register: ‘ZERO’ See [1] for possible values
0x01e1	BYTE	1	EnableOffsetSubtraction	r+w		0 = FALSE 1 = TRUE	0	If TRUE, the OffsetCurrent is calculated as described in [1] and subtracted from the adc values

4.2 ErrorState

Table 4: ErrorStates of state registers

BIT	SHORTCUT	MEANING OF SET BIT
0-19		reserved
20	EEPROM_READ	EEPROM read error
21	EEPROM_WRITE	EEPROM write error
22	MCDC_READ	AS89010 read error
23	MCDC_WRITE	AS89010 write error
24	RDY_TIMEOUT	RDY timeout during color measurement
25	COMM_OVERFLOW	Command receive overflow
26-31		reserved

After a read request from the State register the Registry Error State is set to the value 0.

4.3 Notes for the Register ActAdc

6 FLOAT values are returned when reading out the content of the ActAdc register. The positions 4-6 are the so-called ADC raw values (Xraw, Yraw, Zraw). These raw values correspond to the data read from the output register bank of the signal amplifier AS89010 OUTx register data (compare Readout of the Current ADC Values, chapter 5.2). The following shows the register allocation in relation to the circuit layout of the board AS73210-AS89010-C3:

X_{raw} = OUT1

Y_{raw} = OUT3

Z_{raw} = OUT0

Next to the the OUTx values the AS89010 also provides the value OUTINT data, that is determined during the last integration process of the integration time (output in clock counts). The following shows the normalized ADC values are calculated by the firmware:

$$X_{norm} = \frac{Xraw * 100.0}{OUTINT * Divider}$$

Y_{norm} and Z_{norm} similar.

Divider is currently set divider register value.

The calculated normalized ADC values X_{norm}, Y_{norm}, Z_{norm}) are returned to the positions 1-3 while reading the register **ActAdc**.

During the evaluation of the normalized ADC values the following special cases need to be considered:

X_{norm}, Y_{norm}, Z_{norm} = -1.0: Overdrive due to too high gain settings.

X_{norm}, Y_{norm}, Z_{norm} = -2.0: Override due to too high integration time settings.

X_{norm} = -3.0: There are currently no new measurement values.

The 6-FLOAT values are transmitted via the internal 32-bit representation, packed in a ULONG. The following table shows examples of the FLOAT value transmission.

Table 5: Examples for the transmission of FLOAT values

FLOAT values	Transmitted ULONG value
58.71603	426add37
22.10949	41b0e03c
-3.0	c0400000
29688.0	46e7f000

C example code for the conversion an ULONG value **ulValue** into a FLOAT value **fValue**:

```
unsigned long ulValue;

float fValue = *((float*)&ulValue);
```

5 TYPICAL REGISTER ACCESS METHODS

5.1 General notes regarding color measurement

The following table shows the state of registers that are important for color measurement after power on reset:

Table 6: State of important registers for color measurements after POR

REGISTER	Value after POR	EXPLANATION
IntegrCnt	IntegrCntReset	The xxxReset registers contain the permanent stored settings within the EEPROM with that the four configuration registers are initialized for the color measurement after POR.
AmplAmpl	AmplReset	
Divider	DividerReset	
SynPeriod	SynPeriodReset	
CycleState	=2 (RUN)	The cyclic color measurement starts immediately after POR.

Regardless of the state of the current color measurement a readout of the ActAdc register can be performed at any time using a communication command. Check the following criteria when evaluating the ADC values:

1. The readout of the measured values is always asynchronous and independent of the currently running measurement process. Only the last valid values are read.
4. If the write command is used to change the configuration register (IntegrCnt, Ampl, Divider, SynPeriod) a possibly ongoing measurement process is stopped immediately and the current measured values are flagged as invalid by setting the value of x_{norm} to -3.0. In the case that CycleState = RUN a new measurement is started immediately with the changed configuration settings. Even when setting the register **CycleState** to 1 (= STEP) or 2 (= RUN) the value x_{norm} is set to -3.0.

Note: Valid measurement values exist after the next measurements ($x_{norm} \neq -3.0$).

5.2 Readout of the Current ADC Values

General method to read out current ADC values

```
    Cmd: 2 40
    Answ: 02 00 0040 4287fcc1 425aef38 41cccb04 45d69000 45acb800 45219000
    Explanation:
 02 = Echo of the Commando code (2=read)
 00 = Completion code (00 = error free)
 0040 = Echo of the register ID (0040 = ActAdc)
 4287fcc1 425aef38 41cccb04 = normalized ADC values
   (= 67.99366  54.73361  25.59913)
 45d69000 45acb800 45219000 = ADC raw values
   (= 6866  5527  2585)
```

Important Notice: If the first normalized ADC value (x_{norm}) is = -3.0, the reading of the ADC values must be repeated until $x_{norm} \neq -3.0$. This hint applies to basically to any read command of the ADC values. This is shown in the following examples.

5.3 Changing the Amplification Settings

Read out of the current ADC values:

```
    Cmd: 2 40
    Answ: 02 00 0040 bf800000 bf800000 bf800000 467c8000 467c8000 467c8000
    Explanation:
```

X_{norm} = Y_{norm} = Z_{norm} = bf800000 = -1.0, Overdrive, too high amplification.

Readout of the amplification:

```
    Cmd: 2 18
    Answ: 02 00 0018 00
    Explanation:
 Ampl=0, that means largest possible amplification setting
```

Write command of a by 1 decreased amplification:

```
    Cmd: 1 18 1
    Answ: 01 00
    Explanation:
 Error free setting of the changed amplification settings.
```

Next read out of the ADC values:

```
    Cmd: 2 40
    Answ: 02 00 0040 bf800000 bf800000 4299d533 467c2400 467c2400 4641f000
    Explanation:
```

X_{norm} = Y_{norm} = bf800000 = -1,0, Amplification still too high

Write command of a further by 1 decreased amplification:

```
    Cmd: 1 18 2
    Answ: 01 00
    Explanation:
 Error free setting of the changed amplification settings.
```

Next read out of the ADC values:

```
    Cmd: 2 40
```

```
    Answ: 02 00 0040 42320fa8 421b872e 41a1d79f 45e0b800 45c44800 454c4000
    Explanation:
All ADC values positive --> error-free measurement, valid measurement data.

Operation with gain amplification setting 2 is used immediately after the next POR.
Writing of the amplification 2 into the AmplRegister register.
Cmd: 1 17 2
    Answ: 01 00
    Explanation:
 Setting of the AmplReset register successful
```

5.4 Change of the Integration Time

```
Read out of the ADC values:
    Cmd: 2 40
    Answ: 02 00 0040 41882198 4181657f 40f5d192 452bd000 45235000 449b2000
    Explanation:
 All ADC values positive error-free measurement, valid measurement data,
 Raw data: 452bd000 45235000 449b2000 = 2749  2613  1241
 Possible value range not used by far.
 Increase the integration time so that higher raw data ADC values are at about 43000. For
 example 43000/2749 the increase is about factor 15.

Read out of the IntegrCnt:
    Cmd: 2 16
    Answ: 02 00 0016 10
    Explanation:
 IntegrCnt = 0x10, means highst possible amplification setting

Setting of the gain factor 15 (decimal) in the IntegrCnt:
    Cmd: 1 16 f0
    Answ: 01 00
    Explanation:
 Setting the changed IntegrCnt successful

Next read out of the ADC values:
    Cmd: 2 40
    Answ: 02 00 0040 4190040a 4188da1a 410204ae 472a6800 4721ee00 4699d800
    Explanation:
 Raw data: 472a6800 4721ee00 4699d800 = 43624  41454  19692
```

5.5 Start a Command Controlled Measurement

```
Set CycleState to STOP:
    Cmd: 1 11 0
    Answ: 01 00
    Explanation:
 Setting the CycleState to STOP successful.
  The cyclic measurement is now stopped.

Setting CycleState to STEP:
    Cmd: 1 11 1
    Answ: 01 00
```

```
    Explanation:
 Setting CycleState to STEP successful.
 Now exactly one measurement will be performed.
 Afterwards the internally controlled CycleState returns to STOP.
Read out of the ADC values:
    Cmd: 2 40
    Answ: 02 00 0040 422d1e6a 4217c83d 419ce196 45da9000 45bfa000 45461000
    Explanation:
 Measurement data without errors.

Next read out of the ADC values:
    Cmd: 2 40
    Answ: 02 00 0040 422d1e6a 4217c83d 419ce196 45da9000 45bfa000 45461000
    Explanation:
 The measurement values remain unchanged, CycleState=STOP.

Return CycleState setting to STEP:
    Cmd: 1 11 1
    Answ: 01 00
    Explanation:
 Setting of CycleState to STEP successful.
 Now an additional single measurement will be performed.
 Afterwards the internally controlled CycleState returns to STOP.

Read out of the ADC values:
    Cmd: 2 40
    Answ: 02 00 0040 423a525c 4222a76a 41ab1245 45eb3000 45cd5000 4557f000
    Explanation:
 Measurement values of a new single measurement will be read.
```

--- END ---

List of Literature

[1] Data sheet AS89010, ams Sensors Germany GmbH

[2] RM0008, Reference manual, stm32F100xx.pdf, Doc ID 13902, Rev 9, ST Microelectronics

[3] Datasheet CAT24C32, Rev. 12, Semiconductor Components Industries, July

[4] UM10204 I²C- bus specification and user manual, Rev 03 19 June 2007, NXP

List of Figures

List of Tables

For more information please contact:

ams Sensors Germany GmbH:
Göschwitzer Straße 32
07745 JENA | GERMANY
Phone: +49 3641 2809-0
Fax: +49 3641 2809-12
sales-europe@ams.com
www.ams.

LEGAL NOTES AND WARNINGS

Failure to comply with these instructions could result in death or serious injury.

Misuse of documentation – The information contained in this document is the property of ams Sensors Germany. Photocopying or otherwise reproducing any part of the catalog, whether electronically or mechanically is prohibited, except where the express permission of ams Sensors Germany GmbH has been obtained. In general, all company and brand names, as well as the names of individual products, are protected by brand, patent or product law.

State of document - The information provided in this document is for reference only. Do not use this document as product installation guide since products can be under development to improve performance or any other purpose. Before you start any development or place an order please contact your supplier or ams Sensors Germany for the latest version of this document. ams Sensors Germany explicitly reserves the right to make technical changes to information described in the document.

Information and Disclaimer – The information provided in this document is based on the knowledge of the ams Sensors Germany GmbH as of the date of publication. The ams Sensors Germany GmbH cannot give warranty regarding the accuracy of information provided by third parties. ams Sensors Germany may not have conducted testing or chemical analysis on all incoming material or chemicals. ams Sensors Germany GmbH performs and continues to perform reasonable measures to provide the most accurate data at the given time. Additional efforts to integrate information provided by third parties are performed and continue to be performed. Certain supplier information may be proprietary or limited and not available at release.

Personal Injury: All products are conform to the specifications in accordance with the terms and conditions of ams Sensors Germanys standard warranty. Production processing does not necessarily include testing of all parameters.

RoHS compliant

RoHS Directive 2011/65/EU /REACH INFORMATION - RoHS compliance and PB free: The products of ams Sensors Germany fully comply with the current RoHS-directives. Our semiconductor products do not contain any of the six substance chemical categories, for example including the restriction on lead usage (lead weight may not exceed 0.1% in homogeneous materials). RoHS compliant products are suitable for the usage in lead-free specified processes, when designed to be soldered at high temperatures.

REACH information: ams Sensors Germany products do not contain any of the latest REACH Substances of Very High Concern (SVHC) regarding the Europe Union (EU) Regulation 1907/2006. The latest 155 substances restricted per the REACH Regulation were last updated on June 16, 2014. Please refer to the following for the most current candidate list of substances: http://echa.europa.eu/candidate-list-table.

ams Sensors Germany solutions are not designed or intended for use in critical applications, in which the failures or malfunctions of the product may result in personal injury or death. Use of ams Sensors Germany products in life support systems is expressly unauthorized and any use by customer is completely at their own risk. In the case of a restricted use of the product described here, an application of the product outside of this limitation is at your own risk.

Warranty disclaimer – The warranty expressed herein shall be in lieu of any other warranties, expressed or implied, including, without limitation, any implied warranties or conditions of merchantability and fitness for a particular purpose., which are expressly disclaimed, and is in lieu of any and all other obligations or liability on supplier’s part- For the avoidance of doubt, supplier shall not be liable for any special, incidental, indirect or consequential loss or damage, including loss pf revenue or profit, of any kind of nature, arising at any time, from any cause whatsoever resulting from the use or operation of the products or any breach of this limited warranty.

Legal liability - ams Sensors Germany assumes no responsibility for the use of any foreign products or circuits described in this document or customer product design, conveys no license, either expressed or implied, under any patent or other right, and makes no representation that the foreign circuits are free of patent infringement. ams Sensors Germany further makes no claim as to the suitability of its products for any particular purpose, nor does ams Sensors Germany assume any liability arising out of the use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages.

ESD Warning: Sensor handling precautions should be observed to avoid static discharge.

WEEE Disposal: - The product should be disposed in to according the Directive 2002/96 / EC of the European Council on Waste Electrical and Electronic Equipment [WEEE] and the German electoral law [ElektroG] of 16 March 2005. Please contact our technical support if you need more details.