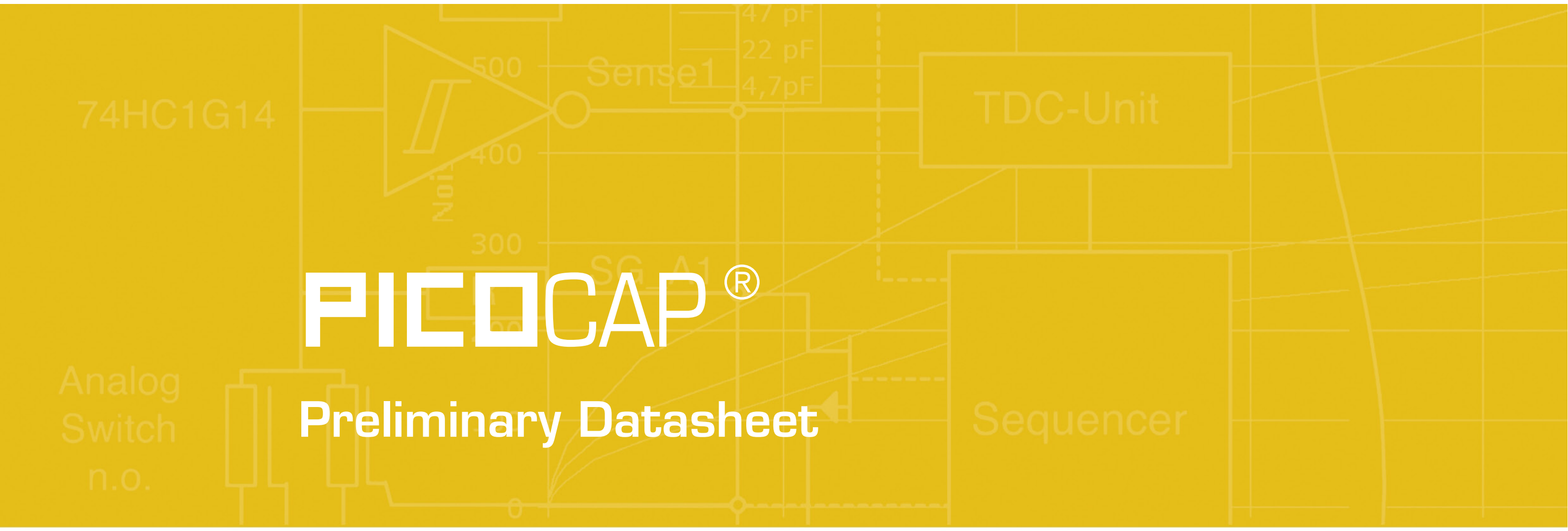
acam
mess·electronic



PICOCAP®

Preliminary Datasheet

PCapØ1AK-0101xx

Single-chip Solution for Humidity Measurement

October 25th, 2010
Document-No.: DB_PCapØ1-AK-0101xx_en

Published by acam-messelectronic gmbh

Legal note

The present data sheet is still under development, which may result in corrections, modifications or additions. acam cannot be held liable for any of its contents, neither for accuracy, nor for completeness. The compiled information is believed correct, though some errors and omissions are likely. We welcome any notification, which will be integrated in succeeding releases.

The acam recommendations are believed useful, the firmware proposals and the schematics operable, nevertheless it is of the customer's sole responsibility to modify, test and validate them before setting up any production process.

acam products are not designed for use in medical, nuclear, military, aircraft, spacecraft or life-support devices. Nor are they suitable for applications where failure may provoke injury to people or heavy material damage. acam declines any liability with respect to such non-intended use, which remains under the customer's sole responsibility and risk. Military, spatial and nuclear use subject to German export regulations.

acam do not warrant, and it is not implied that the information and/or practice presented here is free from patent, copyright or similar protection. All registered names and trademarks are mentioned for reference only and remain the property of their respective owners. The acam logo and the PicoCap logo are registered trademarks of acam-messelectronic gmbh, Germany.

Support / Contact

For a complete listing of direct sales contacts, distributors and sales representatives visit the acam website at:

http://www.acam.de/company/distributors.

For technical support you can contact the acam support team in the headquarter in Germany or the distributor in your country. The contact details of acam in Germany are:

sales@acam.de or by phone +49 7244 7419 0.

1 System Overview

PCapØ1AK-0101xx is a capacitance-to-digital converter which is especially dedicated to capacitive humidity sensors. Its front end is based on based on acam's patented PICOCAP® principle. The circuit has an integrated DSP to do the linearization of the humidity and temperature results by means of polynomials. The software can be stored in an OTP-memory to make the chip run autonomously after power-up. Via a serial I^2C or SPI interface, the chip can communicate with an external master.

1.1 Features

- Digital measuring principle in CMOS technology
- Single power supply (2.1 to 3.6 V)
- Extremely low current consumption possible: 8 µA at 5 Hz including display
- Integrated EPROM for easy calibration
- Re-calibration capability (8-fold)
- Self-boot and stand-alone capability
- Analog-voltage output capability via PDM
- One PDM port for humidity and one for temperature
- Serial interface (SPI, I^2C, selectable)

1.2 Humidity measurement

- Capacitance range 1 to 1000 pF
- Polynomial linearization
- 0.05% r.H. peak-to-peak resolution
- Update rate 1 Hz to 25 Hz

1.3 Temperature measurement

- PT500 or PT1000 sensors
- Polynomial linearization
- 30 mK peak-to-peak resolution ($\sigma = 5$ mK)
- Update rate 1 Hz to 10 Hz

2 Characteristics & Specifications

2.1 Electrical Characteristics

2.1.1 Absolute Maximum Ratings

Supply voltage V_{DD}-to-GND	- 0.3 to 4.0 V
Storage temperature Tstg	- 55 to 150 °C
ESD rating (HBM), each pin	> 2 kV
Junction temperature (Tj)	max. 125 °C
OTP Data Retention Period	10 years at 95 °C temperature

2.1.2 Recommended Or Typical Operating Conditions

Parameter	Symbol	Remarks	Min.	Typ.	Max.	Unit
Supply voltage	V_{DD}		2.1		3.6	V
Operating current	I_{op}	at 3.0 volts and at 5 Hz update rate		8		µA
Digital port voltage	$V_{io\_digital}$	Relative to ground	– 0.3	3.3	V_{DD} +0.6 ≤ 3.6	V
Digital ports switching level		HIGH -> LOW LOW -> HIGH		0.3 * V_{DD} 0.7 * V_{DD}		V
Analog port voltage	V_{io_analog}		– 0.3	3.3	V_{DD} +0.6 ≤ 3.6	V
OTP programming voltage	V_{OTP}	Between "V_{PP_OTP}" port and ground. Do not expose any other ports to programming voltage	6.5		7.0	V
OTP write speed		Write time for programming	30		500	µs
SPI bus frequency	$f_{SPI\text{-}bus}$	Clock frequency for the 4-wire SPI bus operation	0		20	MHz
I2C bus frequency	$f_{IIC\text{-}bus}$		0		100	kHz
GPIO input rise time		general-purpose I/O			1000	ns
GPIO output rise time				t.b.d.		

2.2 Converter Precision[2-1]

Parameter	Symbol	Remarks	Min.	Typ.	Max.	Unit
Humidity	m	Standard deviation Peak-to-peak		0.008 0.05		% r.H.
Temperature	θ	Standard deviation Peak-to-peak		5 30		mK

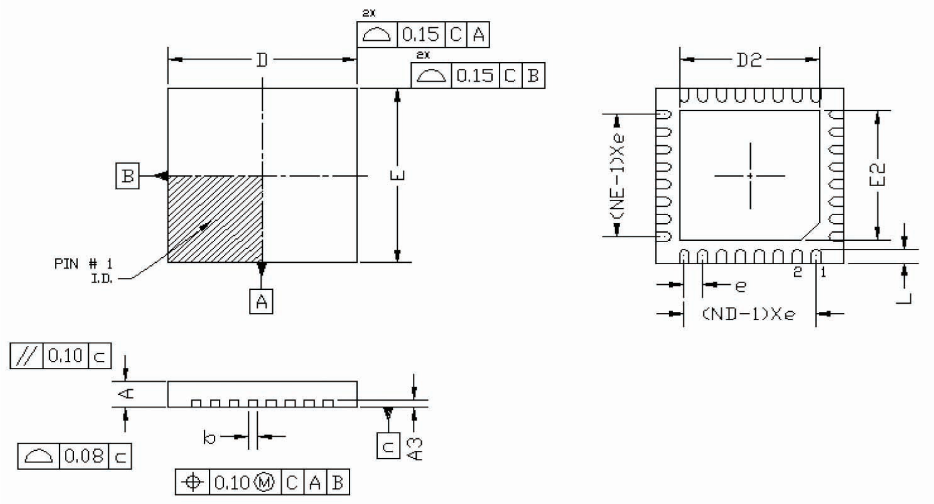
[2-1] Obtained with a PT1000 thermometer and a P14 humidity sensor fabricated by IST AG, Wattwil, Switzerland

2.3 Dice - Pad Layout

To be released. Die dimensions: 2.04 mm x 1.56 mm with pad pitch 120 µm and pad opening 85 µm x 85 µm

2.4 Package

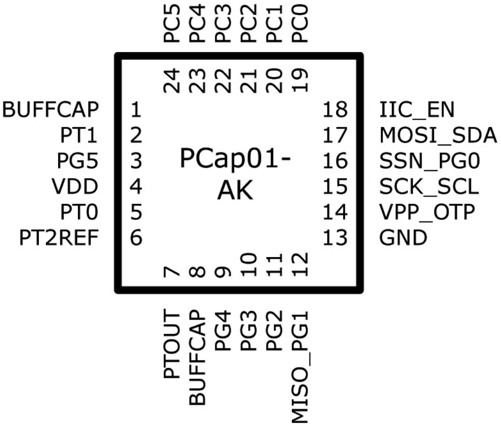
Figure 2-1

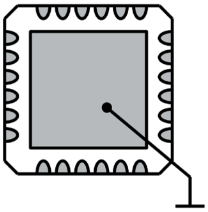


		Dimensions in mm							
Device Name	Package	D, E	D2, E2	N	e	L	b	A	A3
PCap01-AK	QFN24	4.00	2.70	6	0.5	0.35	0.25	0.75/0.9	0.20

2.5 Pin / Pad Assignment

Figure 2-2





Note:
This ground connection is very important!

Pad Name	Pin#	Description
BUFFCAP	1, 8	Bypassing via a ceramic capacitor is mandatory. Bridge #1 and #8
GND	13 and bottom pad	
IIC_EN	18	Interface selection switch. See examples paragraph 3.2 and 3.3.
MISO_PG1	12	
MOSI_SDA	17	
PC0	19	Reference capacitor port
PC1	20	
PC2	21	Capacitive sensor port
PC3	22	
PC4	23	Touch sensor port
PG2	11	Alert output (inverted): ‚low'=alert; ‚high'=idle
PT0	5	Resistive sensor port for temperature measurement (PT1000)
PT2REF	6	Reference resistor port
PTOUT	7	Common port for resistive measurement
SCK_SCL	15	
SSN_PG0	16	
VDD	4	Bypassing via a ceramic capacitor is mandatory.
VPP_OTP	14	must not be unconnected! Ground for operation, 6.5 to 7.0 volts for writing into OTP memory
others	2, 3, 9, 10, 24	leave unconnected

2.6 Interfaces

The PCap01 offers two serial interfaces for programming and for data output, SPI compatible and I²C, selectable. Use the IIC_EN pin to select the one you prefer; see examples in paragraph 3.2 and 3.3.

Further inputs and outputs are a capacitive touch sensor, connected to the capacitive port PC4, two analog outputs for measured temperature and humidity (via general-purpose ports PG0 and PG1), and an alert signal line via PG2. On other versions of firmware, PG2 to PG5 are used for driving an LCD driver (HOLTEK HT1620).

2.7 I²C Interface

For a detailed description of the interface, see ACAM`s dedicated data sheet; see also the bug report in paragraph 7.1.1.

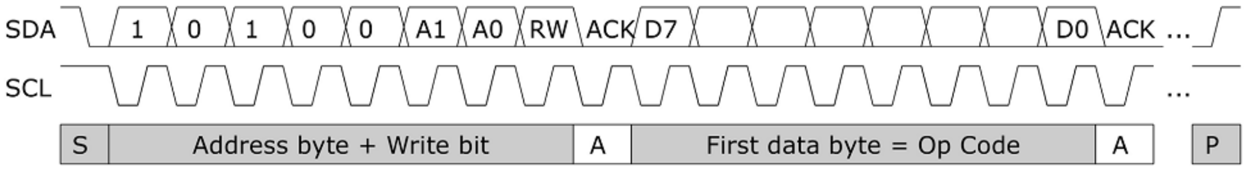
The PCap01 device is slave-only and obeys only to the most basic subset of the I²C standard; an outside master creates a Start condition ("S"), a falling edge on the SDA line while the SCK line is high. Data bits are transferred with every rising edge of SCK.

PCap01 holds a 7-bit slave address, partly fixed, partly variable. The default values for the variable parts are A1=A0=0.

MSB							LSB
1	0	1	0	0	A1	A0	R/W
fixed					variable		

After the address including Read/Write switch follows an op code and – in most cases – a memory address and one or three "payload" data bytes (details in paragraph 6.3). Each transferred byte is followed by an acknowledge bit ("A" or "ACK" with 'low'=acknowledged).

Figure 2-3: I2C principle sequence (master=gray and slave=white)



2.7.1 I²C Write

During write transactions, master sends data and PCap01 sends acknowledge bits. To initialize, master puts a Start condition and sends an address byte with R/W switch ='low'="write". Here is an example: "write byte AA to address 47 in volatile SRAM program memory":

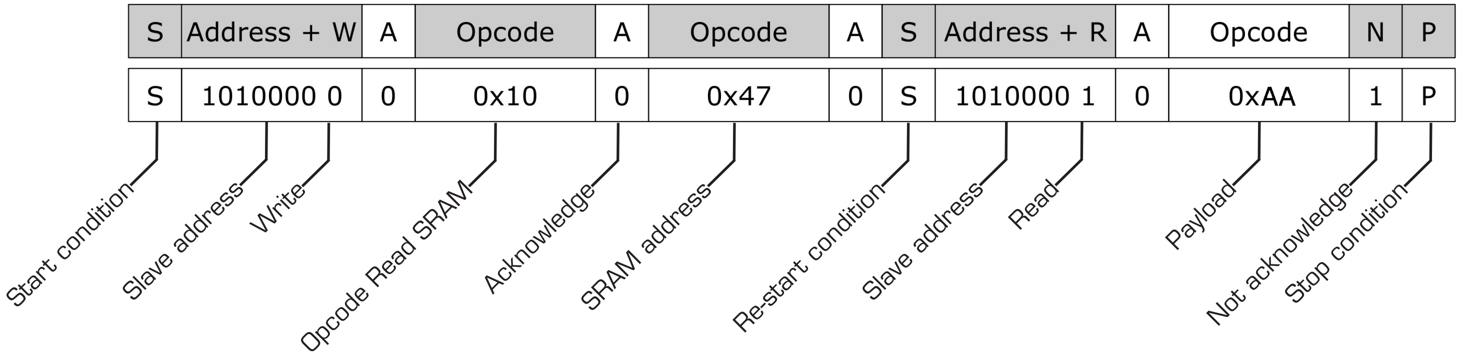
Figure 2-4: I²C write sequence

S	Address + W	A	Opcode	A	Address	A	Data	A	P
S	1010000 0	0	0x90	0	0x47	0	0xAA	0	P

2.7.2 I²C Read

Read transaction means, that the direction of traffic is to be changed. This time, master initializes the communication and writes the op code including the memory address bits, as before, but now toggles from "write" to "read" and listens for the data ("payload") to be sent by PCap01. The write-to-read permutation is operated by a repeated Start condition, slave address sent again, but with R/W switch = 'high' = "read". Here is an example: "read SRAM from address 47". Now the previously written payload "AA" should come out.

Figure 2-5: I2C read sequence



2.8 Timing For I²C Interface

The I²C interface has been specified for a 100 kHz maximum frequency. For data output this is more than fast enough. For programming purposes, one may try faster, but this would be done at customer's own risk.

2.9 The SPI compatible Interface

PCap01 can communicate via SPI interface, too. Again, PCap01 can communicate as slave only and supports one mode (out of four possible ones).

The following choice is necessary for successful operation:

CPOL	Clock polarity	0
CPHA	Clock phase	1
DORD	Bit sequence order	0, MSB first

The clock starts with LOW level. New data are requested by the rising edge of SCK, data are transfered with the falling edge of SCK.

Figure 2-6: SPI Write

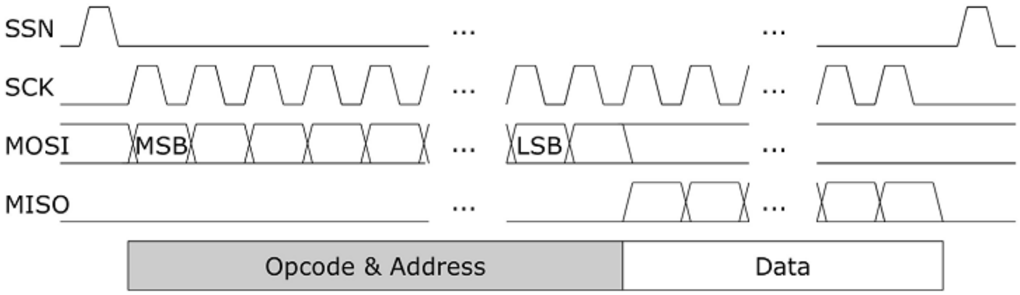
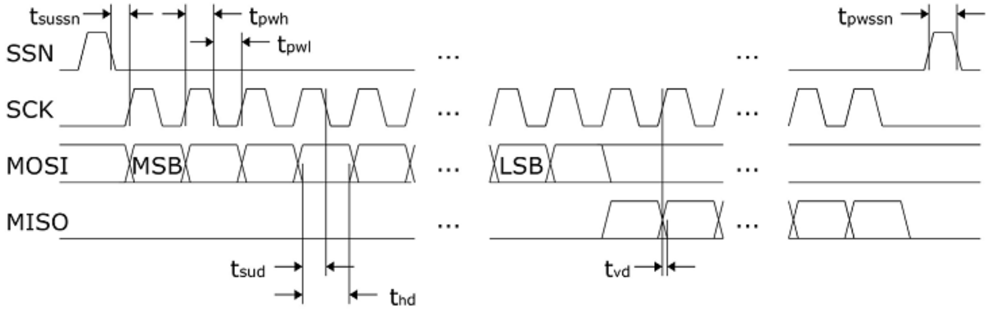
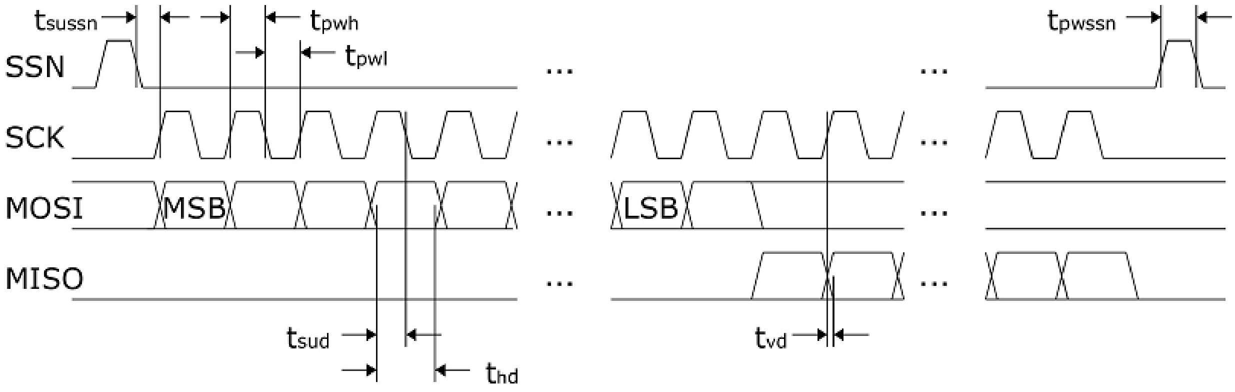


Figure 2-7: SPI Read



2.10 SPI Interface Timing

Figure 2-8: SPI Timings

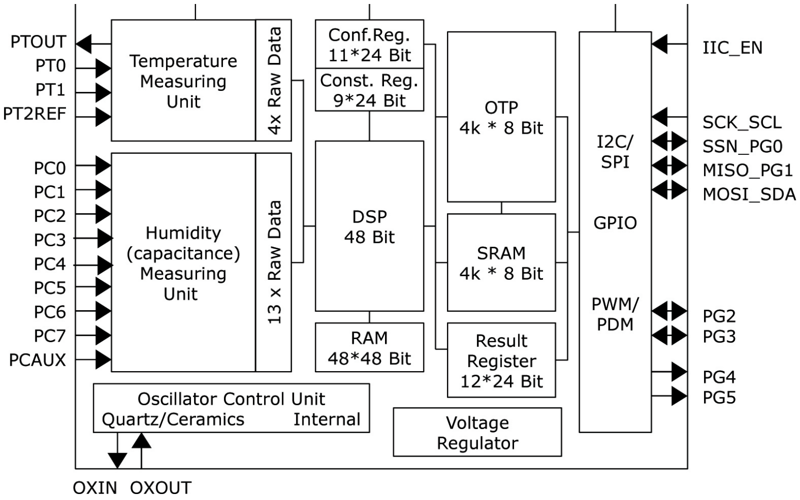


Name	Symbol	V_{DD}=2.2 V max.	V_{DD}=3.0 V max.	V_{DD}=3.6 V max.	Units
Serial clock frequency	fSPI-bus	10	17	20	MHz
Serial clock pulse width HI state	tpwh	50	30	25	ns
Serial clock pulse width LO state	tpwl	50	30	25	ns
SSN enable-to-valid latch	tsussn	10	8	7	ns
SSN pulse width between write cycles	tpwssn	50	30	25	ns
Data setup time prior to clock edge	tsud	7	6	5	ns
Data hold time after clock edge	thd	5	4	3	ns
Data valid after clock edge	tvd	40	26	16	ns

3 Hardware Aspects

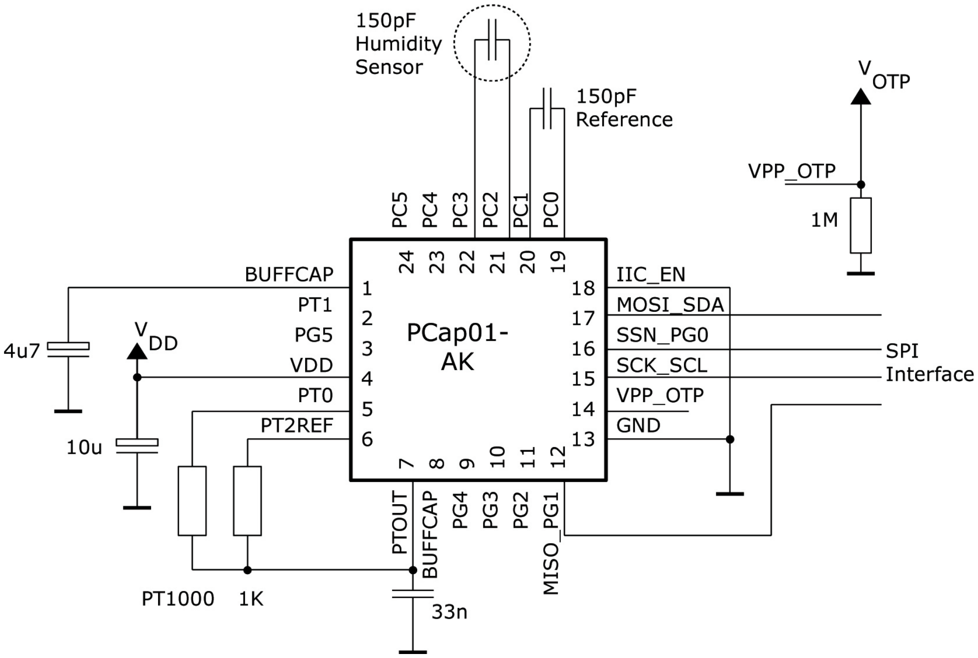
3.1 Chip Device Block Diagram

Figure 3-1: Block Diagram of PCap01



3.2 Typical schematics with SPI as an interface

Figure 3-2 Example Schematics with SPI Interface



Sensor capacitances up to 1000 pF are possible. Sensor and reference same order of magnitude.

In this variant, the SPI interface is used for programming the device and for polling measurement results at regular intervals. Polling should take place with the humidity measurement period, see paragraph 4.2

3.3 Typical schematics with analog outputs

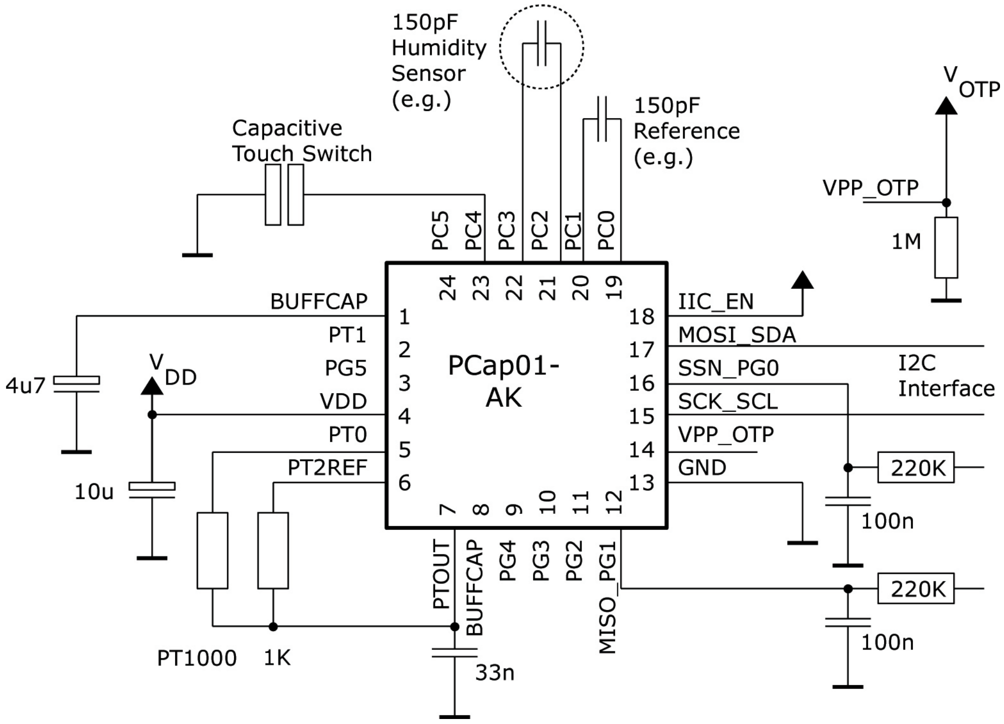


Figure 3-3 Example schematics using the analog outputs

Analog voltage outputs for humidity and temperature values, output ripple < ½ LSB. An I²C interface is provided, especially for programming the device.

4 Device Control

4.1 Op Codes

Basic control op codes:

0x88	Power-up reset - careful, this resets everything except OTP memory.
0x8A	Partial or "initial" reset; resets the front-end and the processor.
0x8E	Start a measurement.
0x84	Terminates a write-to-OTP sequence. Recommended after every byte.

Read-Write op codes:

0xCxxxxxxx	Write configuration parameter to registry (Reg0 through Reg20)
0x4xxxxxxx	Read 24-bit results from result registers (Res0 to Res6)
0x48xxxxxx	Read 24-bit status word
0x9xxxxxx	Write one program code byte to SRAM memory
0xAxxxxxx	Write one program code byte to the OTP memory.

Concerning the format, see paragraph 6.2

4.2 Speed Control (Measurement Rate)

First of all, fix the rate (speed) of your temperature measurement between 1.25 and 10 Hz as follows and note your TDIV value.

Temperature	TDIV
1.25 Hz	7FFF
2.5 Hz	3FFF
3.75 Hz	2FFF
5 Hz	1FFF
10 Hz	0FFF

Then choose the Humidity measurement rate as a multiple of it (factor *n*), but no larger than 25 Hz.

Temperature	Humidity	*n*	Reg17 (dec.)
1.25 Hz	1.25 - 25 Hz	1-20	2, 1, 3 - 20
2.5 Hz	2.5 - 25 Hz	1-10	2, 1, 3 -10
3.75 Hz	3.75 - 22.5 Hz	1-6	2, 1, 3 - 6
5 Hz	5, 10, 15, 20, 25 Hz	1-5	2, 1, 3, 4, 5
10 Hz	10 or 20 Hz	1-2	2, 1

Trivially, transform *n* into a Reg17 value according to the following rule: For $n \geq 3$, Reg17=*n* . For *n*=2, Reg17=1. For *n*=1, Reg17=2. Some fine tuning inside the chip is to be made through a parameter Reg18 as follows:

Hum(Hz)	Reg18
1.25	0212
2.5	0212
3.75	012C
5	00C8
6.25	00A0
7.5	0078
8.75	0050
10	0046
11.25	003C
12.5	0032

Hum(Hz)	Reg18
13.75	0028
15	001E
16.25	001E
17.5	0019
18.75	0014
20	000F
21.25	000F
22.5	000D
23.75	000B
25	000A

Now, write the three parameters into the chip:

TDIV: Reg5=0xC0000 + TDIV; Example: Reg5=0xC07FFF

Command Write-to-Register = OpCode + Address + Contents = 0XC0000000 + 0x05000000 + 0x00C07FFF

=0xC5C07FFF

Reg17: 17=0x11. Example Reg17=6;

OpCode + Address + Contents = 0xC0000000 + 0x11000000 + 0x00000006

=0xD1000006

Reg18: 18=0x12. Example Reg18=002E;

OpCode + Address + Contents = 0xC0000000 + 0x12000000 + 0x0000002E

= 0xD200002E.

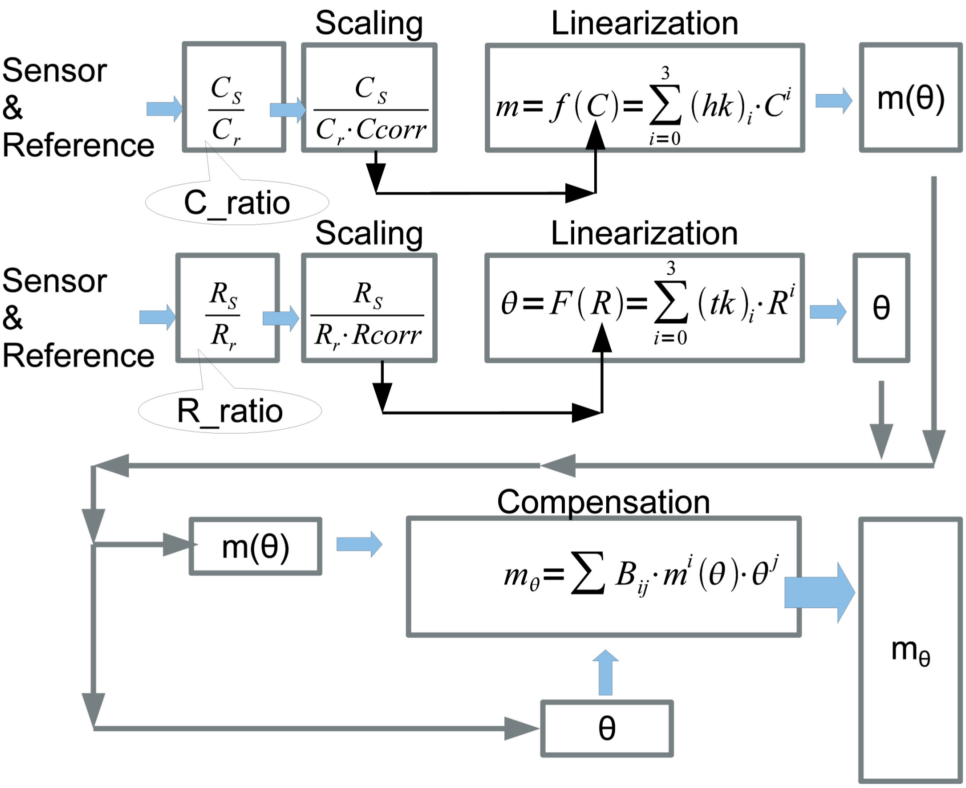
Finish with partial reset command 0x8A and start command: 0x8E.

5 Firmware

5.1 Temperature and Humidity Linearization

Capacitance and resistance must be converted into humidity and temperature. Polynomial approximation is a suited method. Sensor and reference will generally be off their nominal value and need correction before linearization. A simple “scaling” factor (“Ccorr” and “Rcorr” below) may be sufficient, because relative values are important only. Once the temperature is known and humidity calculated from capacitance, the output value of the humidity linearization undergoes a temperature compensation. Relative humidity is the result.

Figure 5-1: Polynomial linearization and compensation



$m(\Theta)$ is the humidity before temperature compensation and Θ is the temperature.

(hk) and (tk) are constants (“coefficients”)

A compensation formula takes into account the temperature and yields an m_Θ value. m_Θ is the temperature compensated relative humidity. B_{ij} are coefficients.

For convenience, the Compensation formula has been rewritten as follows:

$$m_\theta = (B_1 \cdot \theta + B_3 + 1) \cdot m(\theta) + B_2 \cdot \theta + B_4$$

Finally, linearized Θ and compensated m_Θ data are available analog and digital.

5.2 Analog Output

For the analog voltage outputs, the following formulae and coefficients apply:

$$\frac{V_{PG0}}{V_{DD}} = \mathrm{N\_th} / 1023$$

$$\mathrm{N\_th} = \mathrm{V\_th\_slope} \cdot \theta + \mathrm{V\_th\_offset}$$

$$\frac{V_{PG1}}{V_{DD}} = \mathrm{N\_m\_th} / 1023$$

$$\mathrm{N\_m\_th} = \mathrm{V\_m\_slope} \cdot m_\theta + \mathrm{V\_m\_offset}$$

These relative voltages are generated through pulse-density modulation at ports PGO and PG1 (see figure 3-3), with low-pass filtering to a ripple smaller than ½ LSB.

5.3 Digital Output

Via the serial interface, not only the device can be programmed, but values can be read out and calibration can be performed. Paragraph 6.2 gives the opcodes, and the following table the details (addresses and formats), calibration is described in paragraph 5.4 and the following:

5.3.1 Readout Register Structure

Mnemonic	Res_Address	Value	Unit	Format
Res0	0	Temperature Θ	°C	24 bit signed (two's complement), fixed-point, 8 bit fractional
Res1	1	Relative humidity m_Θ	% r.H.	
Res2	2	Humidity m before temperature compensation	%	
Res3	3	C_ratio (C_s/C_r)		24 bit unsigned, fixed point, 22 bit fractional
Res4	4	R_ratio (R_S/R_r)		
Res5	5	N_th see paragraph 5.2		10 bit integer
Res6	6	N_m_th see paragraph 5.2		10 bit integer
Status	8	Status information		24-bit status word

Example: Res0 = 0x004000 <=> Θ = 64 °C

Res0 = 0xFFC000 <=> Θ = –64 °C (two´s complement)

5.3.2 Status Word Information

Bit no.	Meaning
20	The Runbit. This bit is expected to be set.
16	A combined error signal of all known error status
13	CDC overflow. An overflow condition in the capacitive measurement part.
9	Touch port error. A hardware problem with the capacitive touch switch.
8,7	Humidity sensor port error, maybe a hardware problem with the sensor element.
6,5	Capacitive reference port error
3	Temperature measurement error, maybe a hardware problem with the PT1000 element.

Please ignore all status bits not mentioned here.

5.4 Temperature Scaling

Calibration, described in the following paragraphs, is a possibly recursive process and therefore will be made in soft memory (SRAM), not directly in OTP. Hints for programming the OTP will be given in a separate paper.

Concerning temperature, thanks to the platinum thermometer (PT1000) and its standardized resistivity-temperature dependency, a simple scale correction (scaling) is sufficient. The following how-to is structured as follows:

1) load firmware and register settings
2) read out the parameter Rcorr and see that its default value is Rcorr=1.0000
3) acquire measurement data and calculate a better Rcorr value
4) write back the new, better Rcorr value
5) test and validate

#	Task	Action / Detail / Comment
1.1	get the firmware	from CDROM
1.2	load firmware (`00 00 70 62 63 BE 01 20 6E 42 5C 48 A0 01 20 3E ...`)	Write 0x9000 ***00*** 0x9001 ***00*** 0x9002 ***70*** 0x9003 ***62*** etc. etc.
1.3	get Registry bytes	from CDROM
1.4	load Registry into the chip (example of Registry bytes: 42 00 FF, 20 10 22 etc.	Write 0xC0***4200FF*** 0xC1***201022*** etc. until 0xD4***000001*** (the runbit)
2	Read Rcorr	Write 0x1217uu 0x1218ww 0x1219yy 0x121Azz and see that Rcorr (4byte) = uuwwyyzz = D0C0C0C0 meaning 1.0000 (see paragraph 6.3, 6.6 for the calibration bank address 0x217 - 0x21A)
3.1	Start measurement	Write 0x8A (reset), wait a little, then write 0x8E (start)
3.2	Make an ice-point calibration: cool the sensor system down to 0 °C and read R_ratio	Read a series of measure data on Res4 = R_ratio. The opcode is 0x44yyyyyy. The values yyyyy are unsigned, fixed point, 22 bit fractional.
3.3	Rcorr := <R_ratio>	Calculate a mean value on several R_ratio values
4.1	Rcorr(3-byte) → Rcorr(4-byte)	make the 3-to-4 byte format conversion according to paragraph 6.3 (especially examples at the bottom)
4.2	Write Rcorr(4-byte) back	Write 0x9217 ***D0*** 0x9218 ***FA*** 0x9219 ***C9*** 0x921A ***C8*** (example values)
5	Read temperature values. They should be identical to 0 °C within the tolerance and the noise band	Read a series of measure data on Res0 (op code = 0x40yyyyyy) with yyyyy being signed, fixed point, 8 bit fractional, in two's complement, thus the values will be close to 000000 and/or FFFFFF.

5.5 Humidity Scaling and Calibration[5-1]

For humidity, the coefficients are less well known than for the platinum sensor. Offset and slope coefficients must be established either individually or batchwise.

5.5.1 Individual calibration for a single specimen

If you have just one sensor to calibrate, we recommend you to set the scaling factor Ccorr to unity, Ccorr=1 and perform a two-point calibration via hk0 and hk1. Read C_ratio values from Res3, possibly calculate a mean value on a statistical sample. Do this at constant temperature. Do not forget the 3-byte to 4-byte conversion (paragraph 6.3).

Coefficient	**Recommended setting**		**Recommended address**	**Reference/Remark**
	decimal	**4-byte**		
Ccorr	1.00000	D0 C0 C0 C0	2D9 + ...	figure 5-1 and paragraphs 6.3 and 6.6
hk0, hk1	calibration variable		2E1 and 2E9	from a series of measurements, by classical regression analysis, calculate optimum coefficients for offset and slope
hk2, hk3	0.00000	C0 C0 C0 C0	2F1... and 2F9...	no higher order terms; keep it simple

5.5.2 Calibration procedure for batches

Batches, if large enough, may justify replacing the individual two-point calibration by something more elaborate, subject to the assumption that humidity sensor specimen within the batch behave sufficiently alike. Some statistics made on two or three dozens of specimen will provide the basic data. Calibration measurements will be done at two ore more points in the humidity/capacitance space at constant temperature. Least-squares method may help you to find optimum hk0, hk1, hk2 and possibly hk3 coefficients. Once you have fixed the coefficients for your batch, you will simply need to perform Ccorr scaling on individual sensors, much like with platinum above.

5.5.3 Compensation

Finding suitable temperature-compensation coefficients B_i is beyond the scope of this paper. Perhaps the sensor manufacturer is willing to disclose this kind of data for your use.

5.6 Alert

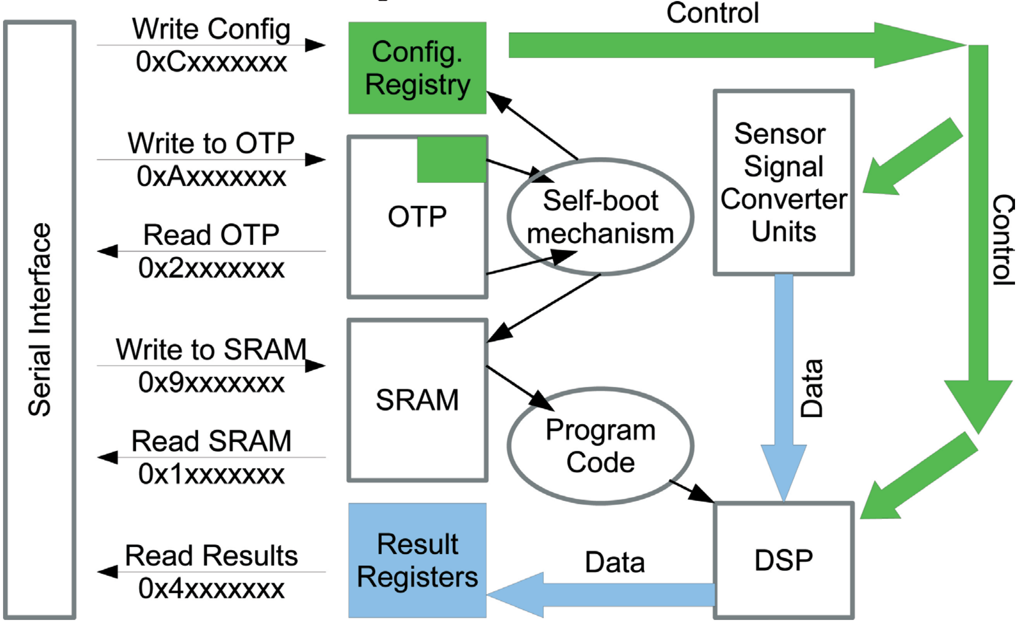
An alert level can be activated. Its default value is 85 % r.H. so the system will signal an alert as soon as m_θ exceeds 85 % r.H. etc. - The level is set in memory at the address indicated in table 6.6, last line. - In case of an alert, PG2 port goes from "high" to "low".

[5-1] When using a P14 humidity sensor, fabricated by IST AG in Switzerland (see paragraph 7.2), as in our original demo hardware, it may be sufficient to perform the Rcorr and Ccorr scalings alone and use bank#0 coefficients (paragraph 6.6) without any further action.

6 Memory Management

6.1 Volatile and Non-Volatile Memory, Registers, Control and Data

Figure 6-1: Control and Data flow between registers.



All registers are volatile with the exception of the OTP memory. Upon power-up, the device is capable of self-booting from OTP. Program code must be loaded into SRAM or stored in OTP. Configuration registry values must be loaded separately or self-booted from a dedicated OTP address range.

6.2 Read and Write Op Codes for SRAM, Registry, OTP Memory and Result Bank

24-Bit Op Code Commands

Command	Byte 2					Byte 1	Byte 0
Write to SRAM	1	0	0	1	Address<11…0>		Data<7…0>
Read SRAM	0	0	0	1	Address<11…0>		Data<7…0>
Write to OTP	1	0	1	Address<12…0>			Data<7…0>
Read OTP	0	0	1	Address<12…0>			Data<7…0>

32-Bit Op Code Commands

Command	Byte 3			Byte 2	Byte 1	Byte 0
Write Config	1	1	Cfg_Address<5…0>	Registry_Parameter<23…0>		
Read Results	0	1	Res_Address<5…0>	Measurement_Results<23…0> or Measurement_Results<47…24>		

The Cfg_Addresses range from 0x00 to 0x14. The Res_Addresses should be taken from paragraph 5.3.1

6.3 Conversion between 3-Byte and 4-Byte Format for Coefficients

Inside Program Memory (SRAM and OTP), address banks have been reserved for calibration values/ coefficients. The banks are made up of 8-byte code sequences which may, in the leading 4 bytes, contain a 3-byte numeric value. The remaining 4 bytes are fixed and must not be modified. A 3-to-4 byte format conversion must be performed in the following manner:

1st byte			2nd byte			3rd byte			4th byte			5th to 8th byte
1	1	bits 23:18	1	1	bits 17:12	1	1	bits 11:6	1	1	bits 5:0	fixed, do not touch

Example:

Rcorr = 1.05678 ; 22 bit fractional = 0x43 A2 48 = 0100 0011 1010 0010 0100 1000
= 0100 00 ' 11 1010 ' 0010 01 ' 00 1000

1st byte			2nd byte			3rd byte			4th byte			5th to 8th byte
1	1	0100 00	1	1	11 1010	1	1	0010 01	1	1	00 1000	fixed, do not touch
0x D0			0x FA			0x C9			0x C8			

More examples:

Example	3-byte-format	4-byte-format
0	0x 00 00 00	0x C0 C0 C0 C0
1.00000 on 22 bits fractional	0x 40 00 00	0x D0 C0 C0 C0
1.05678 on 22 bits fractional	0x 43 A2 48	0x D0 FA C9 C8

6.4 Program Memory Map (SRAM & OTP)

Memory Address	SRAM Contents	OTP Contents	Remark or Example
000 - 20F	Program Code	Program Code	00 00 70 62 63 BE 01 20 6E 42 5C 48 A0 01 20 3E A1 01 20 3E 84 05 27 FD ... DC 45 5B 02 FF FF FF FF FF FF FF FF FF FF FF 00
210 - 213	Calibr. Bank Index	Calibr. Bank Index	FFFFFFFF or others, see next paragraph
214 - 216	Program Code	Program Code	
217 - 2B0	Calibr. Bank #0	Calibr. Bank #0	Original calibration coefficients
2B1 - 34A	Calibr. Bank #1	Calibr. Bank #1	First customer-generated coefficients
34B - 3E4	Calibr. Bank #2	Calibr. Bank #2	Empty, or calibration coefficients after 1st re-calibration
3E5 - 47E	Calibr. Bank #3	Calibr. Bank #3	Empty, or...after 2nd re-calibration
47F - 518	Calibr. Bank #4	Calibr. Bank #4	etc.
519 - 5B2	Calibr. Bank #5	Calibr. Bank #5	etc.
5B3 - 64C	Calibr. Bank #6	Calibr. Bank #6	
64D - 6E6	Calibr. Bank #7	Calibr. Bank #7	
6E7 - 780	Calibr. Bank #8	Calibr. Bank #8	
781 - FBF	unused	unused	
FC0 - FFE	unused[6-1]	Config. Registry	upon self-boot, to be copied into the dedicated configuration registers
FFF	unused	unused	

6.5 Calibration Bank Assignment

Calibration coefficients cannot be physically erased in OTP memory. Instead, when a calibration coefficient bank is no longer valid, new coefficients are written into the next bank. So as to open the right bank, a pointer must be made available to the DSP. This pointer is called “Calibration bank index” and is stored as indicated in the table above.

Index = FF FF FF FF points to bank#0, index = FF FF FF FE points to bank#1
Index = FF FF FF FC points to bank#2, index = FF FF FF F8 points to bank#3

and the other indexes are FF FF FF F0, FF FF FF E0, FF FF FF C0, FF FF FE C0 and FF FF FC C0. This is due to the fact that in OTP, a ‘one’ value may be overwritten by a ‘zero’ but never the other way round.

[6-1] see block diagram and op codes; there is a dedicated RAM block for Registry

6.6 Calibration Address Matrix

	Calibration Bank								
	#0	#1	#2	#3	#4	#5	#6	#7	#8
Rcorr [22]	217	2B1	34B	3E5	47F	519	5B3	64D	6E7
tk0 [8]	21F	2B9	353	3ED	487	521	5BB	655	6EF
tk1 [14]	227	2C1	35B	3F5	48F	529	5C3	65D	6F7
tk2 [19]	22F	2C9	363	3FD	497	531	5CB	665	6FF
tk3 [21]	237	2D1	36B	405	49F	539	5D3	66D	707
Ccorr [22]	23F	2D9	373	40D	4A7	541	5DB	675	70F
hk0 [8]	247	2E1	37B	415	4AF	549	5E3	67D	717
hk1 [8]	24F	2E9	383	41D	4B7	551	5EB	685	71F
hk2 [8]	257	2F1	38B	425	4BF	559	5F3	68D	727
hk3 [8]	25F	2F9	393	42D	4C7	561	5FB	695	72F
B1 [21]	267	301	39B	435	4CF	569	603	69D	737
B2 [24]	26F	309	3A3	43D	4D7	571	60B	6A5	73F
B3 [24]	277	311	3AB	445	4DF	579	613	6AD	747
B4 [8]	27F	319	3B3	44D	4E7	581	61B	6B5	74F
V_th_slope [8]	287	321	3BB	455	4EF	589	623	6BD	757
V_th_offset [1]	28F	329	3C3	45D	4F7	591	62B	6C5	75F
V_m_slope [8]	297	331	3CB	465	4FF	599	633	6CD	767
V_m_offset [1]	29F	339	3D3	46D	507	5A1	63B	6D5	76F
Alertlevel [8]	2A7	341	3DB	475	50F	5A9	643	6DD	777

Explanation:

- in bank #0 the Rcorr coefficient is stored at address 0x217 through 0x21A. Do not modify the contents of address 0x21B to 0x21E, they contain code. Next coefficient, tk0, at address 21F through 222, then four bytes of code, and so forth. Respect all intermediary, code containing cells. Modifying the code would cause desaster.
- In brackets, the position of the fixed point in 3-byte representation.

Example:

Rcorr = 1.00000 = 0x400000 after 22-fold left shift of fixed point. Three-byte 0x400000 transforms into D0 C0 C0 C0 in 4-byte format.

When a negative value makes sense, the number is written in two‘s complement (especially the tk, hk and B coefficients). For two‘s complement, see example in 5.3.1.

7 Miscellaneous

7.1 Bug Reports

7.1.1 ACK/NACK bug on I²C Bus

Situation	Expected Behavior	Actual behavior
Communication between Master and some Slave other than PCap01	PCap01 should remain completely silent.	PCap01 acknowledges any address byte. PCap01 acknowledges any data byte. Any NACK will be overwritten by PCap´s ACK.

ACK/NACK conflicts may hang up the bus. In consequence,

- you should take the SPI interface if possible;
- no I²C multi-master activity permitted; single-master only;
- you should restructure your network so that PCap is single slave;
- if you cannot, you should use a software interface inside your master, tailored to the ACK/NACK bug situation.

When PCap is single slave, or out of several the slave being addressed, it behaves correctly.

7.1.2 Reference Documents

- ACAM preliminary data sheet "PCap01" (general data sheet without any focus on a particular application)
- ACAM General Information "I²C bus data sheet"
- IST AG, Wattwil, Switzerland. "P14 Capacitive Humidity Sensor" (www.ist-ag.com)
- DIN EN 60751 "Industrial Platinum Resistance Thermometers"

7.1.3 Document History

22.10.2010 Initial Release

acam-messelectronic gmbh

Am Hasenbiel 27

76297 Stutensee-Blankenloch

Germany

ph. +49 7244 7419 - 0

fax +49 7244 7419 - 29

e-mail: sales@acam.de

www.acam.de