PICOCAP®

Data Sheet

PCapØ1-EVA-Kit

Evaluation System for PCapØ1A

February 26th, 2013
Document-No.: DB_PCapØ1-EVA-Kit_en VO.3

Published by acam-messelectronic gmbh
©acam-messelectronic gmbh 2012

Disclaimer / Notes

"Preliminary" product information describes a product which is not in full production so that full information about the product is not available yet. Therefore, acam-messelectronic gmbh ("acam") reserves the right to modify this product without notice. The information provided by this data sheet is believed to be accurate and reliable. However, no responsibility is assumed by acam for its use, nor for any infringements of patents or other rights of third parties that may result from its use. The information is subject to change without notice and is provided "as is" without warranty of any kind (expressed or implied). **PICO**CAP® is a registered trademark of acam. All other brand and product names in this document are trademarks or service marks of their respective owners.

Support / Contact

For a complete listing of Direct Sales, Distributor and Sales Representative contacts, visit the acam web site at:

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

PCapØ1

Content

1 Introduction

1.1 General

The PCapØ1-EVA-KIT evaluation system provides a complete system for generally evaluating the PCapØ1 chip. It comprises of a main board, a plug-in module, a Windows based evaluation software, assembler software and the PICOPROG programming device. The PCapØ1 evaluation board is connected to the PC's USB interface through the PICOPROG V2.0 programming device.

The evaluation kit offers user friendly configurations for evaluating the PCapØ1 single-chip solution for capacitance measurement. This kit can be used to evaluate the capacitance measurement, temperature measurement and the pulse generation capabilities of the PCapØ1 chip.

For a proper use of the evaluation system it is strongly recommended to refer to the current PCapØ1A datasheets (DB_PCapO1-O3O1, DB_PCapO1_DSP etc.). You can download these datasheets from www.acam.de/download-center/picocap .

1.2 Component List

- PCapØ1-MB   Motherboard
- PCO1-AD Plug-in module   Based on PCapØ1-AD in QFN32 package
- PICOPROG V2.0   Programmer
- High density DSUB15 cable   Connecting the Evaluation board to the programmer
- USB cable   Connects PICOPROG V2.0 to the PC
- Wall power supply unit   9 V
- CD-ROM   Includes software and data sheets

Figure 1-1: Components of the evaluation kit

2 Connecting Capacitors and Resistors

This evaluation kit can be used for evaluating capacitance measurement by connecting capacitive sensors. Further, it can be used for evaluating temperature measurement by connecting external temperature sensitive resistors or for generating quasi analog voltage (pulse width/density modulated) that is dependent on the sensor connected to the system.

Depending on the purpose of evaluation, a modification has to be made to the same plug-in module. Following is a picture of the Mother board with the plug-in module.
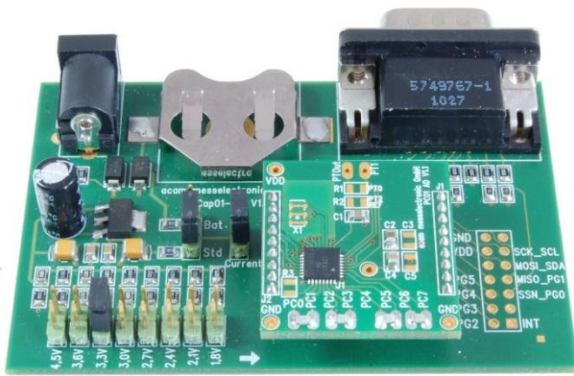

Figure 2-1: The evaluation kit's motherboard and plug-in module

The following sections describe the modifications for each application in detail.

2.1 Capacitance Measurement

For the purpose of evaluating the capacitance measurement using PCapØ1, the plug-in module is pre-assembled with ceramic capacitors to emulate capacitive sensors. These capacitors, each 47pF in value, are connected to the 8 ports PCO to PC7. They are connected as single sensors in floating mode, i.e. each capacitor is connected between 2 ports, and hence there are 4 x 47pF on-board capacitors. Please refer to Section 3.4 of the PCapØ1 Data Sheet for more information on how to connect capacitors to the chip. The capacitor connected between ports PCO and PC1 is taken as the reference capacitor.
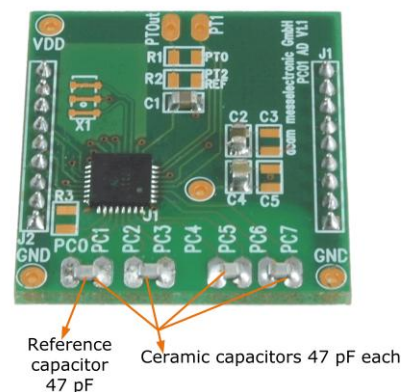

Figure 2-2: Details of the plug-in module

In the process of evaluation, when you are comfortable with interpreting the measurement results from the chip, these fixed capacitors can be replaced with the actual capacitive sensors of your application.

If you want to connect your capacitive sensors in grounded mode, then GND points are provided at the two ends of the module, where the sensor ground connections ought to be soldered.

The typical value of the capacitive sensors that can be connected to the evaluation kit lies in the range of 30pF to 3.5nF. The reference capacitor should be in the same order of magnitude as the sensor. Depending on the value of the sensor, the value of the internal resistor for performing the measurement has to be selected. For the pre-assembled 47pF capacitors, an internal discharge resistor of 180kOhm works well. See section 3.5 of the PCapØ1 data sheet on how to select the value of the internal discharge resistor.

2.2 Temperature Measurement

Temperature measurement or other resistive tasks may also be of interest for the user of this kit. The evaluation kit offers this possibility through the RDC (resistive-to-digital converter) ports. An on-chip thermistor coupled with an on-chip temperature-stable reference resistor made of polysilicon is sufficient for observing the temperature measurement capability of the PCapØ1 chip.
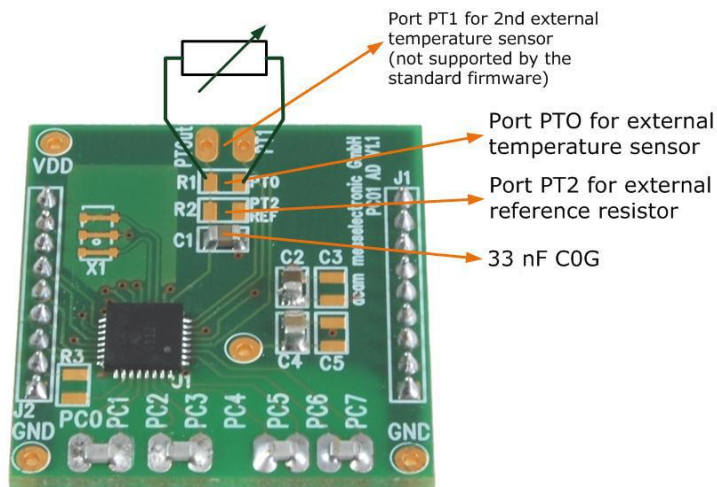

Figure 2-3: Temperature sensor connection pads

However, there is a possibility to connect the reference resistor and the thermistor externally to the chip, too. In case of external resistors, the temperature-stable reference resistor ought to be connected at port PT2REF on the plug-in module. The module allows you to connect the external thermistor, e.g. a PT1000 sensor at port PTO (or PT1, not supported yet by the standard firmware).

In any case, for the temperature measurement, an external capacitor 33 nF COG has to be connected to the chip; it is already pre-assembled on board.

2.3   Pulse Code Generation

Any of the capacitance or temperature measurement results from the PCapØ1 chip can be given out as a pulse width modulated or pulse density modulated signal. This output can be filtered to generate an analog output signal that can be used for further controlling.

These pulse width or pulse density codes can be generated at Ports PGO, PG1, PG2 or PG3. Since ports PGO and PG1 are used for the SPI Interface in the module, the hardware allows to get a valid pulse width/density modulated signal on PG2 or PG3. However, when I2C communication mode is used the pulsed signals can be optionally obtained on the ports PGO and PG1.
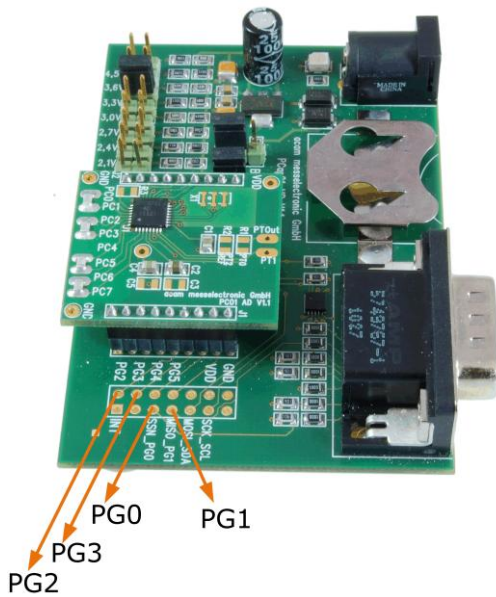

Figure 2-4: General purpose interface ports PGO to PG3

2.4   Motherboard

The motherboard connects to the PICOPROG programmer. It serves the various power options. It can be powered via wall plug supply, the voltage being set from 1.8V to 4.5V by jumpers. Further, it supports a battery power option. Power present is indicated by a green LED.

There is a jumper 'Current' on the mother board. The current consumption of the PCapØ1 chip during operation can be directly measured from these jumper terminals.

All interface signals and general purpose I/O signals can be monitored by means of a separate jumper.

3   Evaluation Software

3   Evaluation Software

3.1   Installing the Software

The PCapØ1 -EVA software runs under the following operating systems

- Windows 2000
- Windows XP
- Windows Vista (please use the software/drivers for Windows 7)
- Windows 7

Please follow the described procedure to install the software and driver:

NOTE: Ensure that the Picoprog V2.0 programmer is disconnected before starting the procedure.

The steps are slightly different for Windows XP and Windows 7. Please note as follows.

Installation on Windows 7 systems:

- Install the PCapØ1 Evaluation software from Software\ PCapO1-Evaluation-Software\ setup.exe from the CD. This .exe file installs the evaluation software and the respective drivers for the Picoprog programmer too.
- At the end of step 1, a batch file comes up in a separate window. Eventually, it will ask you to unplug the Picoprog V2.0 programmer. Please follow further the instructions on the window.
- The driver will be installed; a windows message may pop up asking you to install the driver though it is not signed. Please install it anyway.
- The batch file asks you to plug in the programmer and it is now enumerated.
- You will be asked to restart the system at this point, please do so. After restart connect the evaluation board to the Picoprog programmer.
- On restart, launch the application from the start menu, if possible as administrator. The software opens and a pop up window asks to select between a 'Standard' mode and 'Humidity'. Please select 'Standard'.
- The LED on the Picoprog programmer will turn green. The software will initialize.
- Perform Verify Interface from the Eval.Software main window. If every is correctly installed, the verify interface must result in an 'Interface OK' message. Also, under *Help → Programmer → hdlUSB → USBO::Ox194E::Ox100B::NI-VISA* ought to be seen.

- If you want to change from the default SPI to I2C interface, please select under *Tools → Interface → I2C interface*. The LED on the Picoprog programmer should now turn red.

Installation on Windows XP systems:

- Install the PCapØ1 Evaluation software from Software\ PCapØ1-Evaluation-Software\ setup.exe from the CD. This .exe file installs the evaluation software and the respective drivers for the Picoprog programmer too.
- At the end of step 1, a batch file comes up in a separate window. It will ask you to reconnect the Picoprog V2.0 programmer.
- You will then be asked to restart the system at this point, please do so.
- On restart, Found New Hardware wizard comes up. Please select 'No, not this time' and continue further with 'automatic installation'.
- The driver will be installed; a windows message may pop up asking you to install the driver though it is not signed. Please install it anyway.
- Now, launch the evaluation software application from the start menu. The software opens and a pop up window asks to select between a 'Standard' mode and 'Humidity'. Please select 'Standard'.
- The Found New Hardware wizard comes up again. Please select again 'No, Not this time' and continue further with 'automatic installation'.
- A message would pop up warning that the driver is not digitally signed. Please install the driver anyway.
- Now the driver installation is complete.
- The LED on the Picoprog programmer will turn green. The software will then initialize and ask you to connect the Evaluation board to the Picoprog programmer.
- Perform Verify Interface from the Eval.Software main window. If every is correctly installed, the verify interface must result in an 'Interface OK' message. Also, under *Help → Programmer → hdlUSB → USB0::0x194E::0x100B::NI-VISA* ought to be seen.
- If you want to change from the default SPI to I2C interface, please select under *Tools → Interface → I2C interface*. The LED on the Picoprog programmer should now turn red. When the LED does not glow at all, then it indicates that the interface is faulty.

3.2   The Graphical User Interface

The software comes up with a small window offering selections for the operating mode. In general, start with the standard mode as it offers all options. The humidity option simplifies and specifies the displays for the humidity firmware. For details please see the datasheet for the humidity evaluation kit.
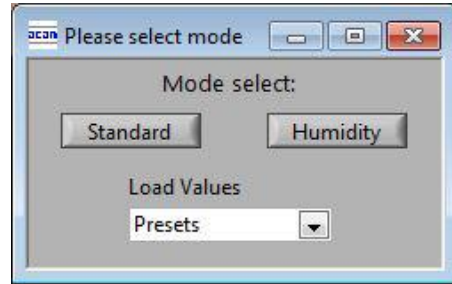

Figure 3-1: Mode selection

Next, the main front panel comes up. Overall, the graphical user interface offers various windows for on-line configuration, for parameter and calibration data setting, and of course for the graphical and numerical display of the measurement data. The various windows will be explained in this chapter.
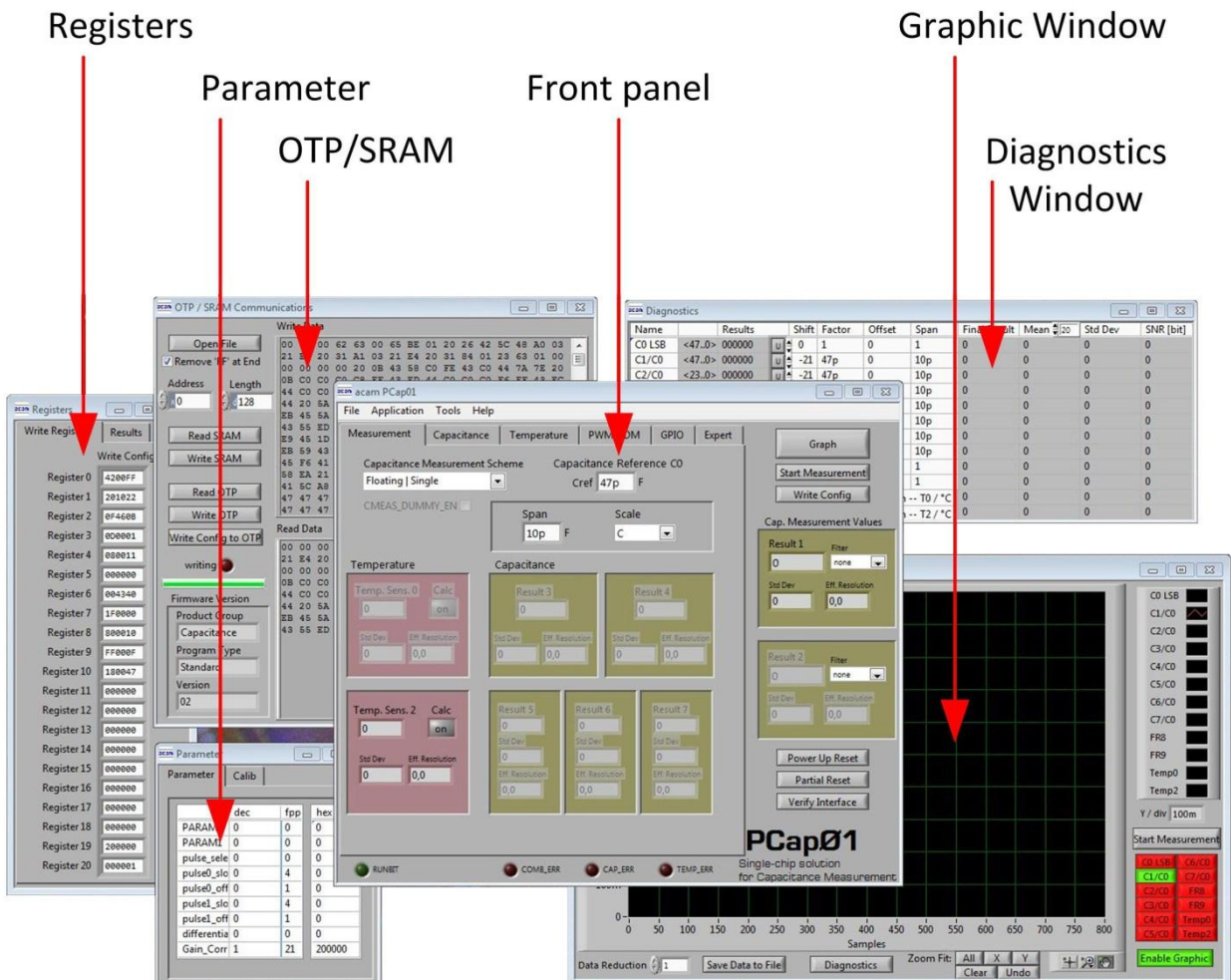
Figure 3-2: GUI Overview

3.3   Front Panel

This is the major window. On the right side, the front panel shows six general buttons:

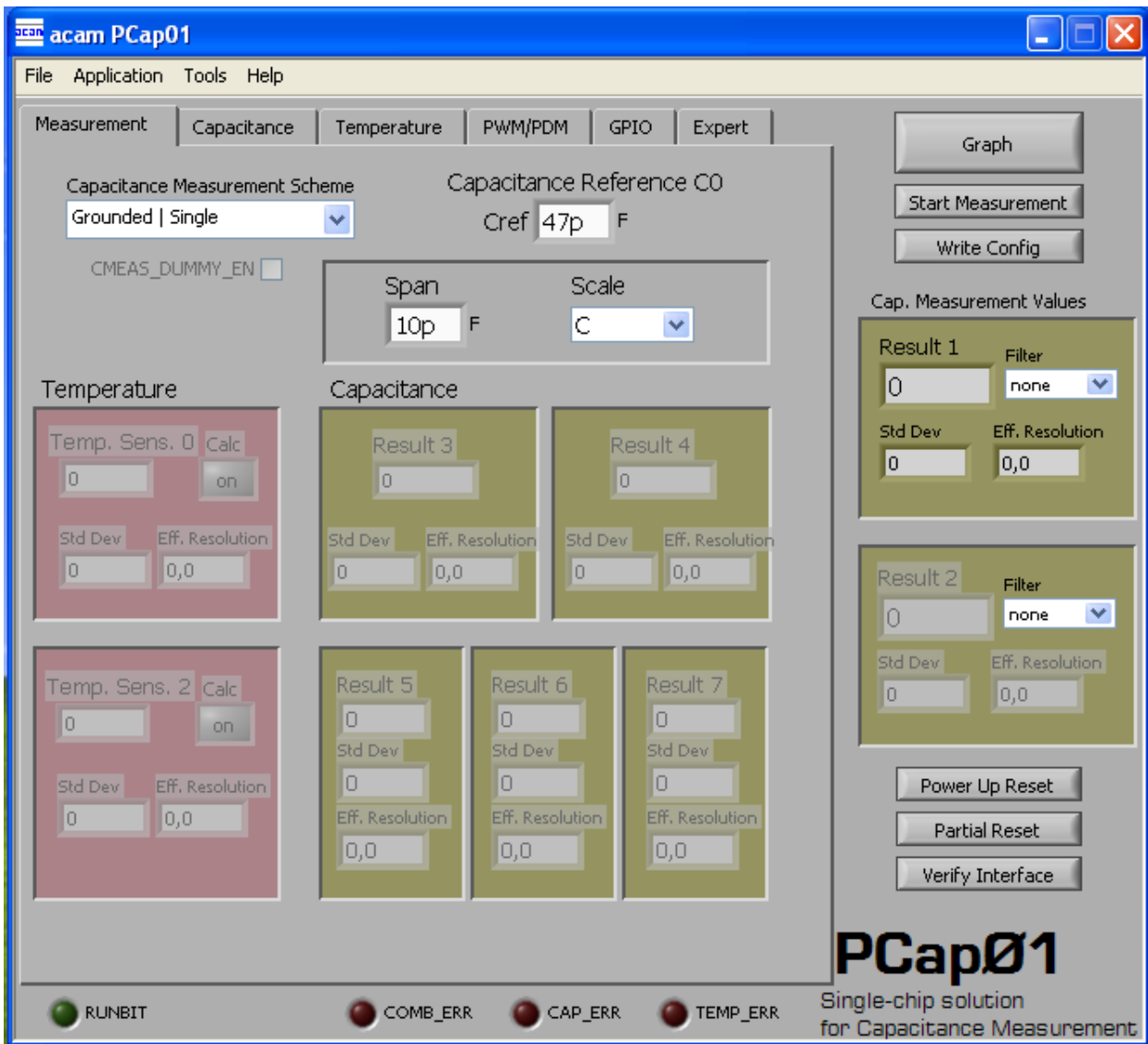|	
Graph	Open a window for graphic representation of measurement data
Start Measurement	Start or stop a running measurement
Write Config.	Transfer once more, the present settings in the evaluation software to the chip (in case of doubt)
Power up Reset	After Power up reset, 'Write Config.' may be necessary.
Partial Reset	With a partial reset, the chip is re-initialized with respect to its frontend and processor.
Verify Interface	When everything is in order, then pressing this button will confirm if an SPI / I2C interface is present. It also indicates the release version number of the software.

3.3.1 Measurement Page

Figure 3-3: Measurement page

Options on 'Measurement' page:

Capacitance Measurement Scheme	**Grounded	Single** – Single capacitive sensor connected between a port and ground. **Grounded	Differential**- Differential capacitive sensor connected between 2 ports with the middle tap of the sensor connected to ground. **Floating	Single** – Single capacitive sensor connected between 2 ports. **Floating	Differential** – Differential capacitive sensor connected between 2 ports with the middle tap of the sensor connected to another 2 ports. Please see Section 3.4 of PCapØ1 data sheet for more details.

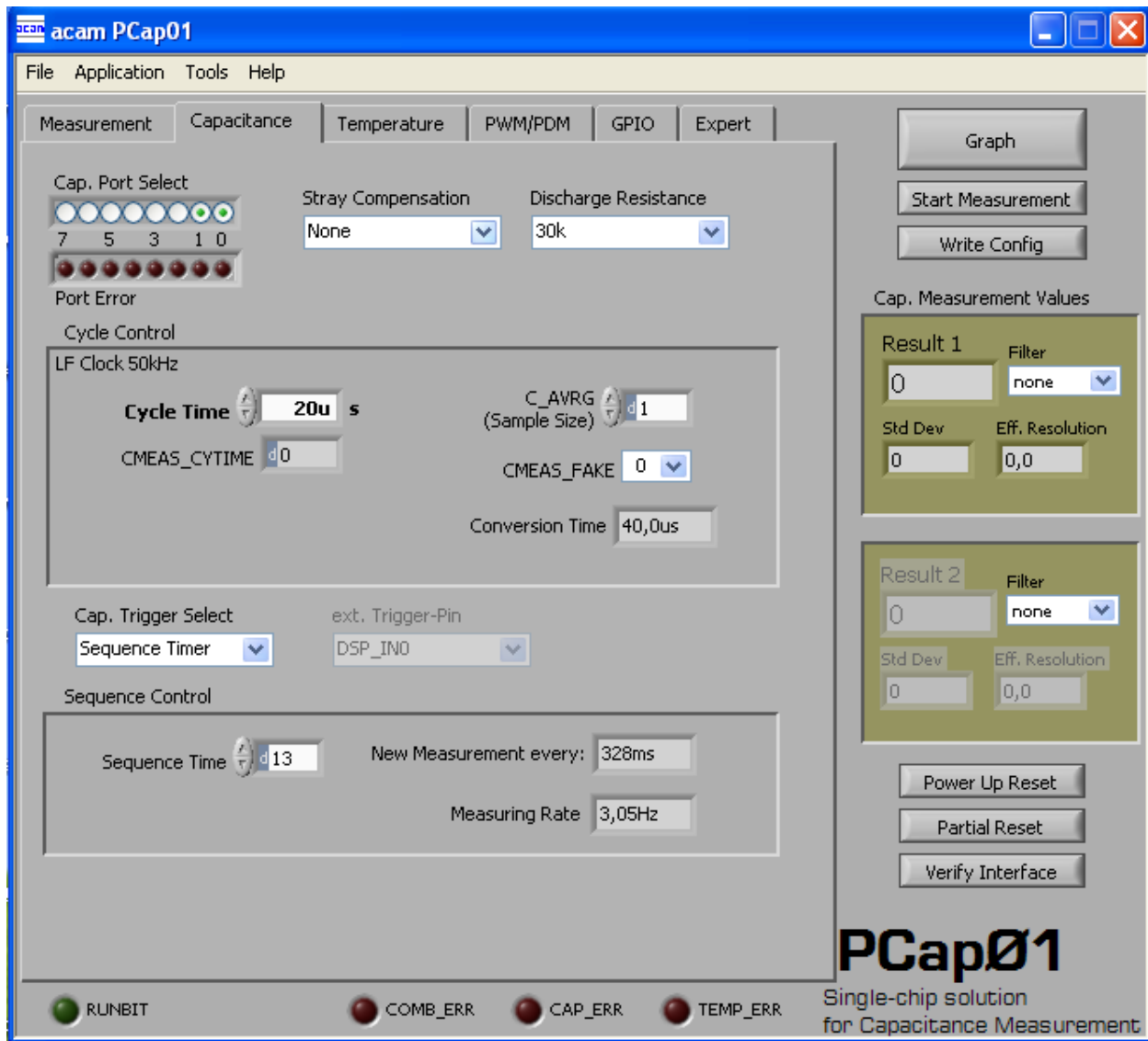| Capacitive Reference CO	Reference capacitance value. This setting has no effect on the chip itself; it is purely a visual aid that helps to interpret the measurement results better.
Span	Maximum span of the reference capacitive sensor connected. This setting has no effect on the chip; it is only used for scaling the "Eff. Resolution" indication.
Capacitance	These fields with an olive green background display the measurement result at capacitive ports 3-7 provided these have been enabled on the 'Capacitance' sheet.
Temperature	These fields with a pink background display the measurement result at each temperature measurement port that has been enabled on the 'Temperature' Sheet. 'on' Button : Pressing the on Button in this part of the sheet comes up with a sub window. This helps to visually manipulate the display of the temperature measurement result – either to display the actual measurement value from chip or to display the temperature directly in Celsius, Fahrenheit or Kelvin (or any other scale) by using polynomial approximation. See section 3.2.4 for further details.
Cmeas_dummy_en	Some differential sensors (MEMS) require mirror symmetry with respect to the charges applied on the plates. This is ensured by "dummy charging" and is activated by this option.

3.3.2   Capacitance Page

Figure 3-4: Capacitance page

Options on 'Capacitance' page:

Cap. Port Select	Select which capacitive ports have to be measured (Ports 0-7), i.e. at which ports the sensors have been connected in hardware.
Stray compensation	**Internal**: One additional measurement performed through only the chip-internal stray capacitance with respect to ground. **External**: One additional measurement per port pair, performed through a parallel connection of the capacitance at the two ports with respect to ground. **Both**: Both internal and external compensation together. **None**: No compensation

| Discharge Resistance	Selects the value of the internal resistance through which the discharge cycles during measurement are to be performed. This value has to be selected in accordance with the capacitance value of the sensor. Please see Section 3.5 of PCapØ1data sheet for more details.
Cycle Time	Can be set in multiples of 20 µs that corresponds to one Cmeas_cytime that is displayed below this box.
C_AVRG (Sample size)	Enables averaging the measurement results over multiple measurement cycles. Setting to 1 → No averaging, Setting to any number N, will result in averaging over N measurement cycles for generating one measurement result. 8192 maximum.
CMEAS_FAKE	Number of fake measurements per measurement cycle. Performing fake measurements may help in reducing noise.
Conversion Time	Displays the entire conversion time per measurement, taking into account, the number of ports opened and the the cycles for compensation and fake measurements.
Cap. Trigger Select	Selects the source that triggers the start of a capacitance measurement **Single** – Started by SPI Command 0x8C (Expert > Capacitance page) **Continuous** – Continuous measurement, self-triggering. Recommended when no temperature measurement is made in parallel. **Sequence timer** – Depending on the setting in the 'Sequence control' panel. Generally recommended setting → less prone to error conditions. **Pin triggered**- Triggered by external Pin, selectable from option ext.Trigger-Pin
ext. Trigger-Pin	Used to select the pin to be used as the source of trigger for the capacitance measurement. NOTE: In the delivered EVA module, the pins DSP_IN0 and DSP_IN1 are part of the SPI communication interface, hence only DSP_IN2 and DSP_IN3 selections are relevant.
Sequence Control :	When the timer is set to N, the capacitive measurement is triggered once every $[2^{(N+1)} * 20]$ µs
New Measurement begins every	Displays the rate at which the capacitive measurement is triggered based on the setting of the Sequence timer. It includes the conversion time and the pause time before the beginning of the next cycle
Measuring rate	Displays the frequency at which capacitive measurement data (with fakes and with averaging) is transferred from the DSP to the interface (SPI or I2C).

3.3.3   Temperature Page
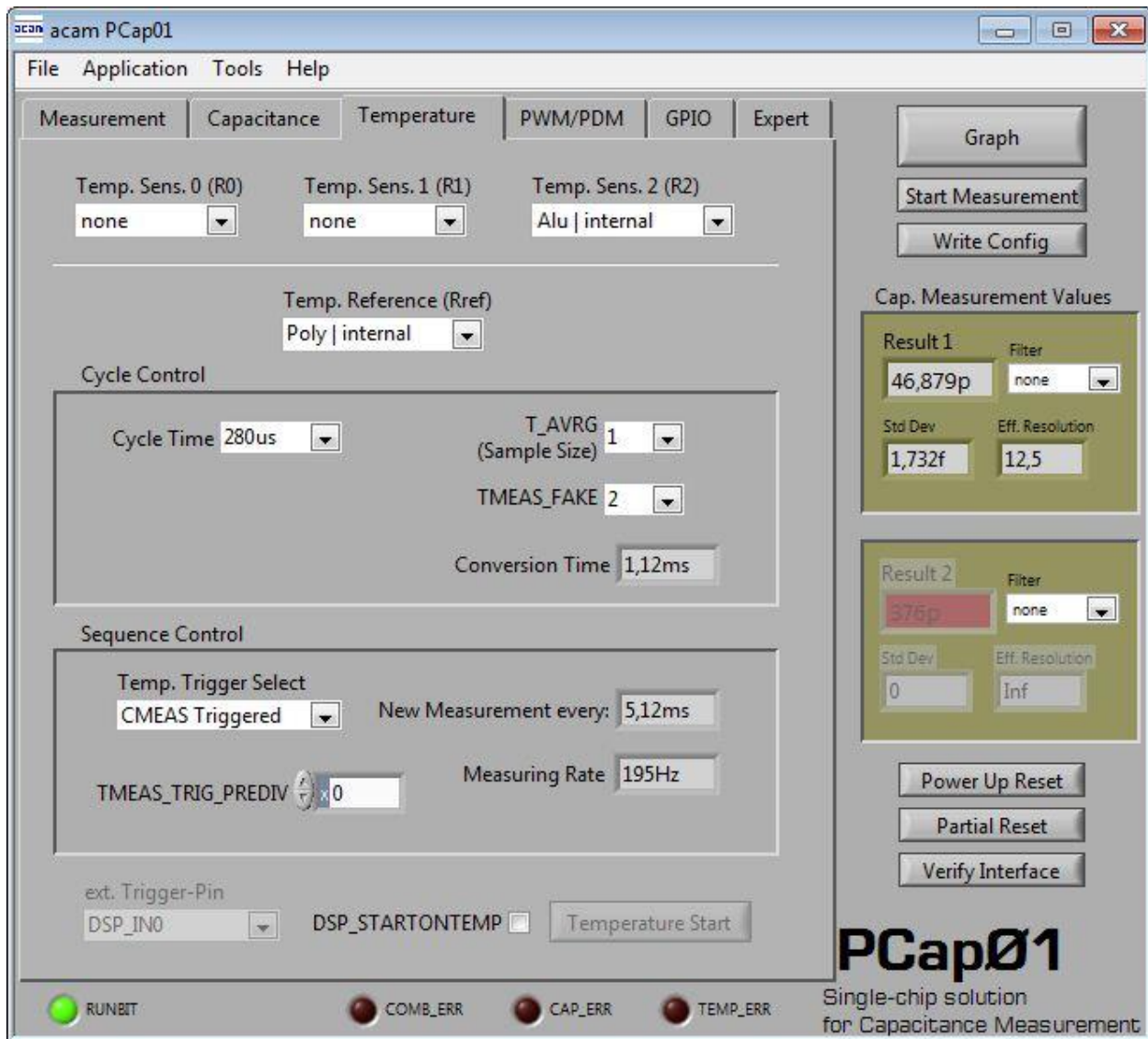
Figure 3-5: Temperature page

Options on 'Temperature' page:

Temp.Sens.0 (R0)	To select a thermistor connected to port PT0 for temperature measurement. This could be e.g. an external PT1000.
Temp.Sens.1 (R1)	To select a thermistor connected to port PT1 for temperature measurement*
Temp.Sens.2 (R2)	To select either the internal aluminum thermistor or an external reference resistor at port PT2for temperature measurement.
Temp. Reference (Rref)	To select either the internal Poly or external resistor at port PT2 as the reference resistance to be used in temperature measurement.

Cycle Time	Can be set to 140 µs or 280 µs. 280 µs is recommended.
T_AVRG (Sample size)	Enables averaging the measurement results over multiple measurement cycles. Setting to 1 → No averaging, Setting to any number N, will result in averaging over N measurement cycles for generating one measurement result.
TMEAS_FAKE	Number of fake measurements per temperature measurement cycle
Conversion Time	Displays the entire conversion time per measurement, taking into account, cycles for averaging and fake measurements.
Temp. Trigger Select	Selects the source that triggers the start of a temperature measurement ▪ **Off / Opcode triggered**: Default setting when no temperature measurement has to be performed automatically. In this case, a temperature measurement can still be started by SPI Command 0x8E. The SPI Command can be sent by pressing the button 'Temperature Start' ▪ **CMEAS triggered**: A temperature measurement is triggered every time when a capacitance measurement is complete. → **Recommended setting for working with temperature measurements.** ▪ **Timer triggered**: Depending on the setting in the 'TMEAS_TRIG_PREDIV' counter in the Sequence Control panel. This counter is steps up in 20 µs steps. Not recommended, not supported by standard firmware. ▪ **Pin triggered**: Triggered by external Pin, selectable from option ext.Trigger-Pin
ext. Trigger-Pin	Used to select the pin to be used as the source of trigger for the temperature measurement. NOTE: In the delivered EVA module, the pins DSP_IN0 and DSP_IN1 are part of the SPI communication interface, hence only DSP_IN2 and DSP_IN3 selections are relevant.
DSP_STARTONTEMP	This setting is used to start the DSP on the completion of temperature measurement. With the standard firmware, the DSP is started after every capacitance measurement sequence, the temperature values are processed at this time. If the capacitance measurement is switched off, then setting this option will start the DSP after every temperature measurement completion.
Temperature Start	See Temp. Trigger select → Off/Opcode triggered above

Typical configurations are:

1. No temperature measurement:

     Set Temp.Trigger Select to 'Off / Opcode triggered'.

2. Internal temperature measurement:

Setting like in figure 3-4. Temp.Sens.2 = Alu | internal, Temp.Reference = Poly | internal, Temp.Trigger Select to 'CMEAS triggered'.

3. External sensor, internal reference:

Temp.Sens.0 = PT0 | external, Temp.Reference = Poly | internal, Temp.Trigger Select to 'CMEAS triggered'. The external sensor has to be connected at pads R1.

4. External sensor, external reference:

Temp.Sens.0 = PT0 | external, Temp.Reference = PT2REF | internal, Temp.Trigger Select to 'CMEAS triggered'. The external sensor has to be connected at pads R1, the external reference resistor has to be connected at pads R2.

3.3.4 Coefficients for Temperature Linearization

When you connect a temperature sensor at Port PT0 for temperature measurement, then you can set the coefficients of the 3rd degree polynomial that is used to linearize the temperature measurement within the evaluation software.

When you configure the Temp.Sens.0 to PT0 on the 'Temperature' sheet, then on the 'Measurement' sheet, the 'on' button in the box with the pink background is enabled. There are two boxes as shown in figure 3-6, the top box is to feed coefficients for the temperature sensor if connected at port PT0 and the bottom box is to feed coefficients for the internal temperature sensor at port PT2, if selected.
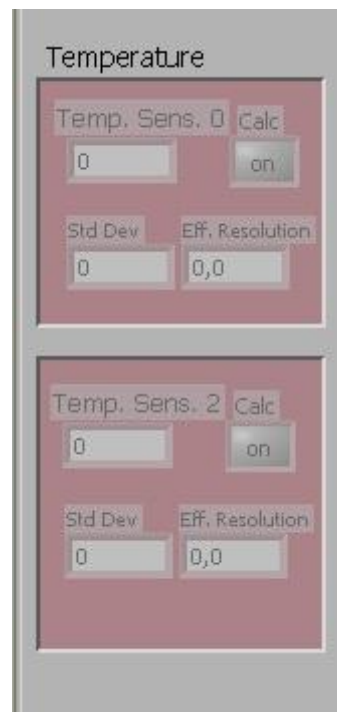

Figure 3-6: Temperature linearization

Click on the 'on' button to feed in the coefficients. For e.g. when an external temperature sensor is connected to Port PT0, then click on the off button on the top. The window figure 3-7 pops up.

Here you can feed in the coefficients of the 3rd degree polynomial. The coefficients displayed by default are for a PT1000 sensor. Change them according to the sensor you use. You can additionally also choose a filter to be applied to temperature measurement result in software. Finally select the 'on' option at the top and click OK. If you do not want the result to be linearized at all, then select the 'off' option.

In case you use the internal aluminum sensor for temperature measurement, then click on the 'on' button in the bottom window. The window in figure 3-8 pops up.
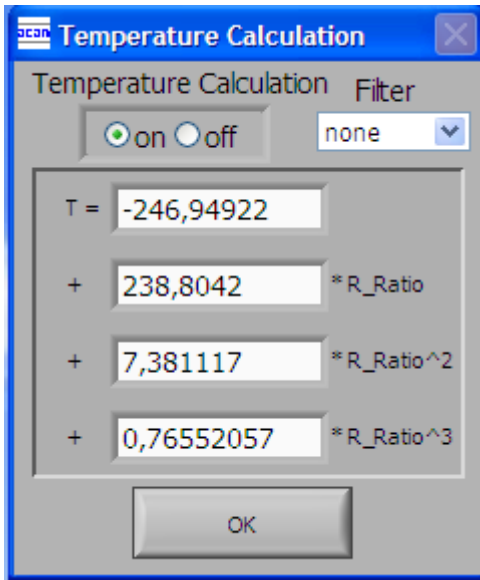

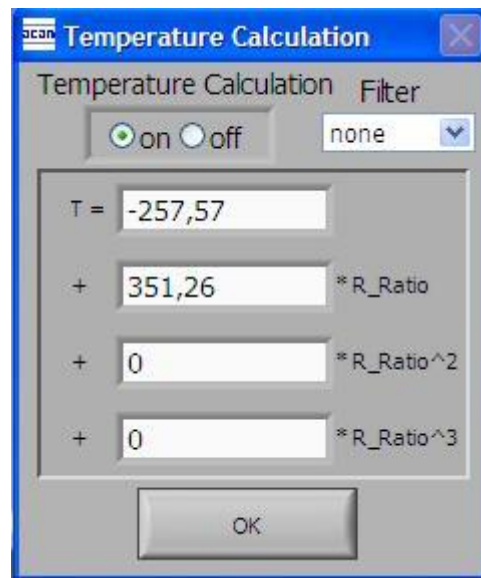Figure 3-7: Temperature linearization coefficients, e.g. PT1000

Figure 3-8: Temperature linearization coefficients, e.g. internal aluminum sensor

The coefficients to linearize the temperature measurement when using the internal aluminum sensor are set by default. You just have to enable it by selecting the 'on' option at the top. If you do not want the result to be linearized, then select the 'off' option.
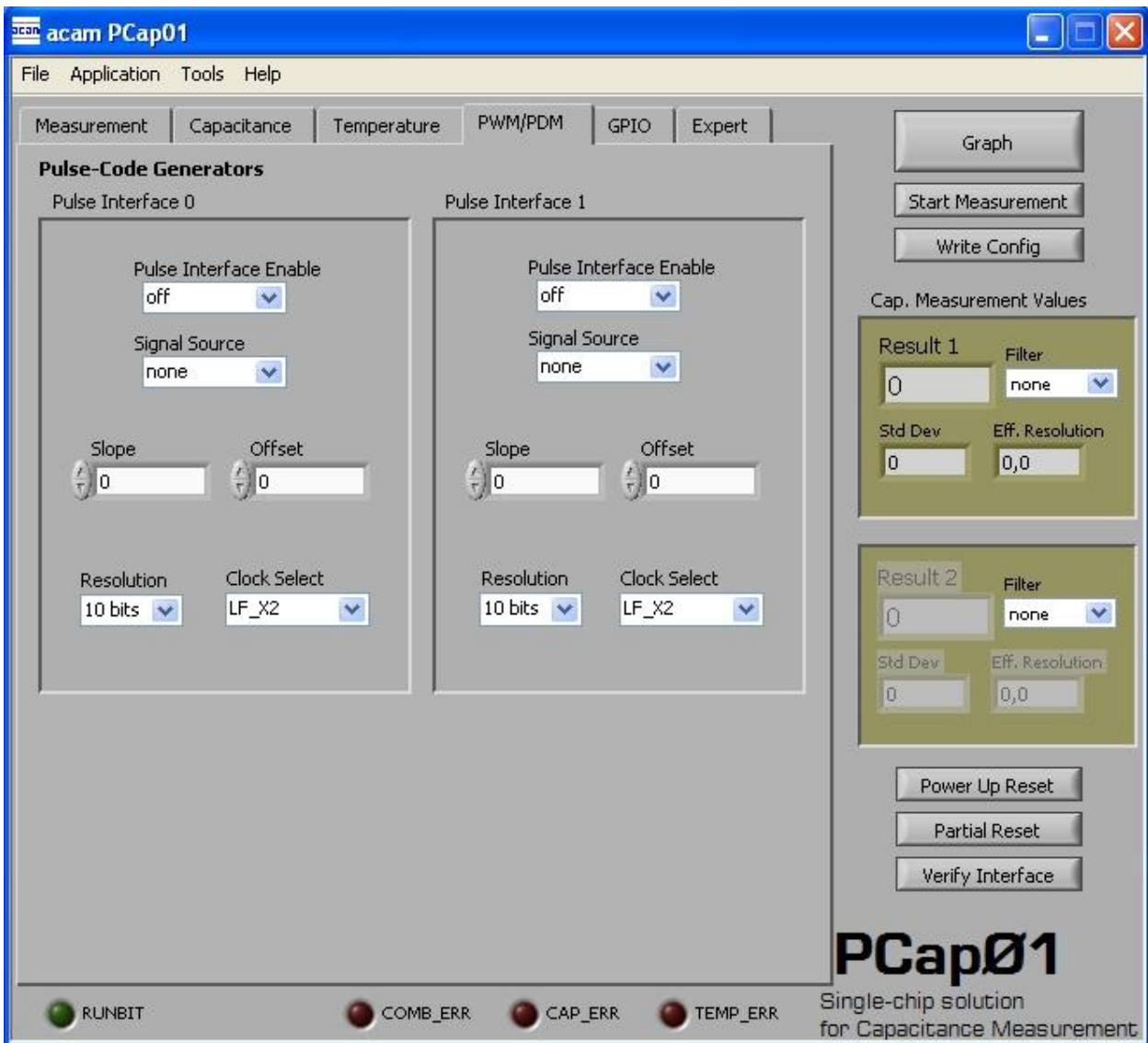
3.3.5 PWM / PDM Page

Figure 3-9: PWM/PDM page

Options on 'PWM / PDM' Page:

Pulse Interface Enable	Select the pulse interface – Pulse Width Modulated Output (PWM) or Pulse Density Modulated (PDM) Output. Of the two, the PDM is the recommended interface. With PWM option, 100 kHz clock and 10-bit resolution the resulting PWM output frequency = (100 kHz / 1024) ~ 100 Hz.
Signal source	Select the measurement result which has to be given out as pulsed output – any of the capacitance or temperature measurement results.
Slope	Used to set the slope (m) of the linearization function used to scale

	the range of the PWM / PDM output generation. See Section 4.2 in PCapØ1 data sheet for more details.
Offset	Used to set the Offset (b) of the linearization function. This value determines the range of the PWM / PDM output in the y direction. See Section 4.2 in PCapØ1 data sheet for more details.
Resolution	Resolution of the output in bits. This resolution also determines the pulsed output range.
Clock_select	Selects the clock frequency to be used for the PWM/PDM generation. Recommended setting LF_X2 → 100 kHz

These settings refer to the use of the standard firmware. The information set is saved in the parameter registers.
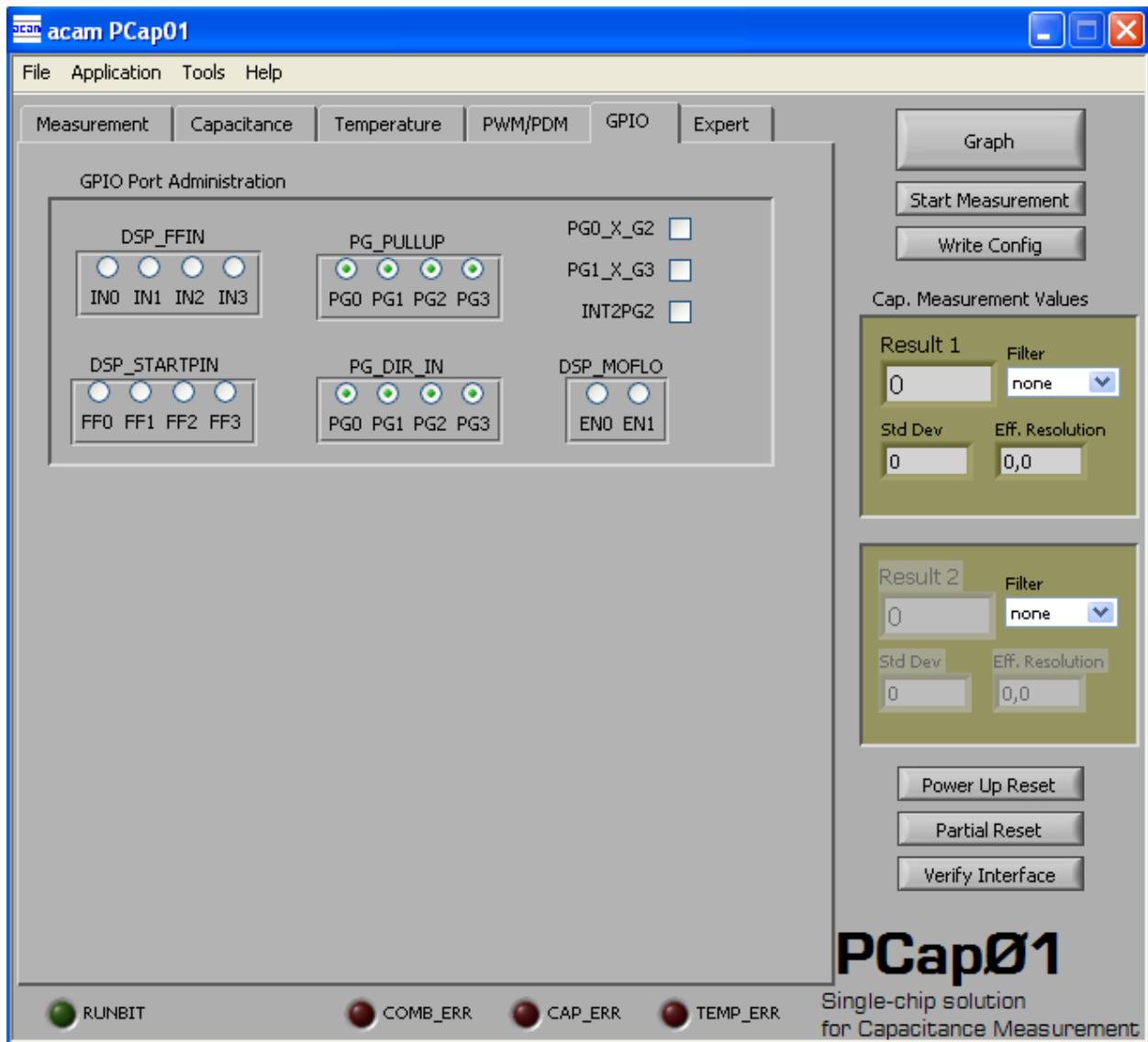
3.3.6   GPIO Page

Figure 3-10 GPIO page

Options on 'GPIO' Page:

DSP_FF_IN	Pin mask for latching flip-flop activation (PGO to PG3)
DSP_STARTPIN	Not supported by standard firmware The DSP can be started externally by a signal on a pin; these buttons select the pin that has to be sensed for detecting the start signal.
PG_PULLUP	To enable the internal pull up on the ports PGO-PG3
PG_DIR_IN	To configure the ports PGO-PG3 as input (otherwise output)
PGO_X_G2	Possible only when the selected interface for communication is IIC.

	Interchange PortGO with PortG2. This is useful when the Pulsed output is needed on Port PGO instead of PG2.
PG1_X_G3	Possible only when the selected interface for communication is IIC. Interchange PortG1 with PortG3. This is useful when the Pulsed output is needed on Port PG1 instead of PG3.
INT2PG2	Map the Interrupt output from chip, INTN to Port PG2. This setting is useful for 24 pin QFN package, because the dedicated INTN pin is absent in this version.
DSP_MOFLO	Activates anti-bouncing filter in PGO and PG1 lines

3.3.7 Expert Page

Please modify the settings on the Expert page only in consultation with acam Support team.

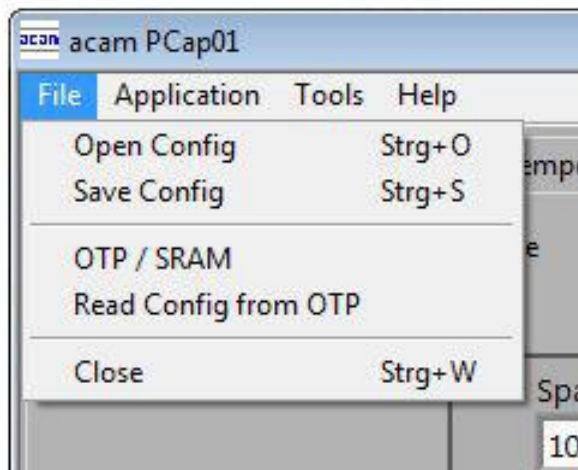## 3.4 Front Panel Menus

3.4.1 File Menu

Figure 3-11: File Menu

Open Config	Open configuration file *.cfg that contains the content for the configuration and parameter registers. PCapO1_Initial.cfg is the one to start with when using the kit as delivered.
Save Config	Here you can save your own

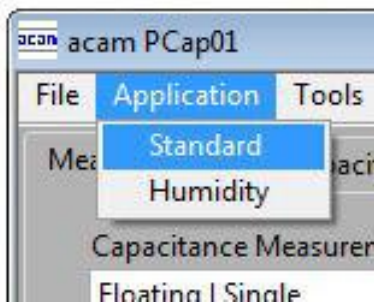|	configuration.
OTP/SRAM	Opens the window to download the firmware. See section 3.5.1.
Read Config from OTP	Reads back the configuration information in the OTP.
Close	Close the evaluation software

3.4.2 Application Menu

Figure 3-12: Application Menu

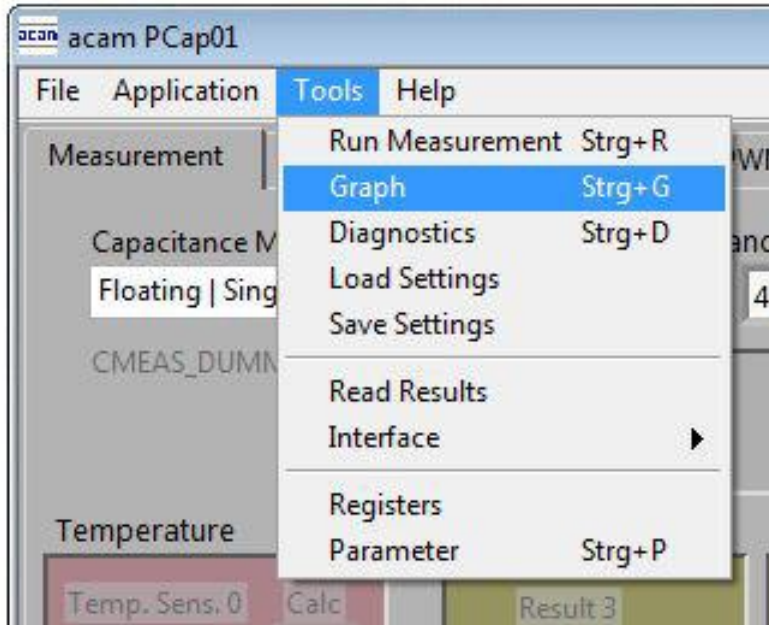| Standard	Sets the evaluation software in the standard display mode, as described in this datasheet.
Humidity	Sets the evaluation software in a special display mode dedicated to our humidity firmware. See the datasheet of the humidity evaluation kit for further information.

3.4.3   Tools Menu

Figure 3-13: Tools Menu

Run Measurement	Start the measurement
Graph	Opens the window for graphical display of the various measurement results (section 3.5.2)
Diagnostics	Opens the window for numerical display of the various measurement results (section 3.5.3)
Load Settings	Load the settings file *.set that contains the display information (labels, number of fractional digits e.t.c.)
Save Settings	Saves the current display settings.
Read Results	Single read of the result registers
Interface	Select between SPI and I2C interface
Register	Opens the Register window (section 3.5.4)
Parameter	Opens the Parameter/Calibration

	window (section 3.5.5)

3.4.4   Help Menu

Figure 3-14: Help Menu

Help Contents	Opens the help window
Programmer	Opens the window with information about the NI VISA interface and the possibility for interface tests
About	Version

3.5 Special Windows

3.5.1 OTP/SRAM Communication Window

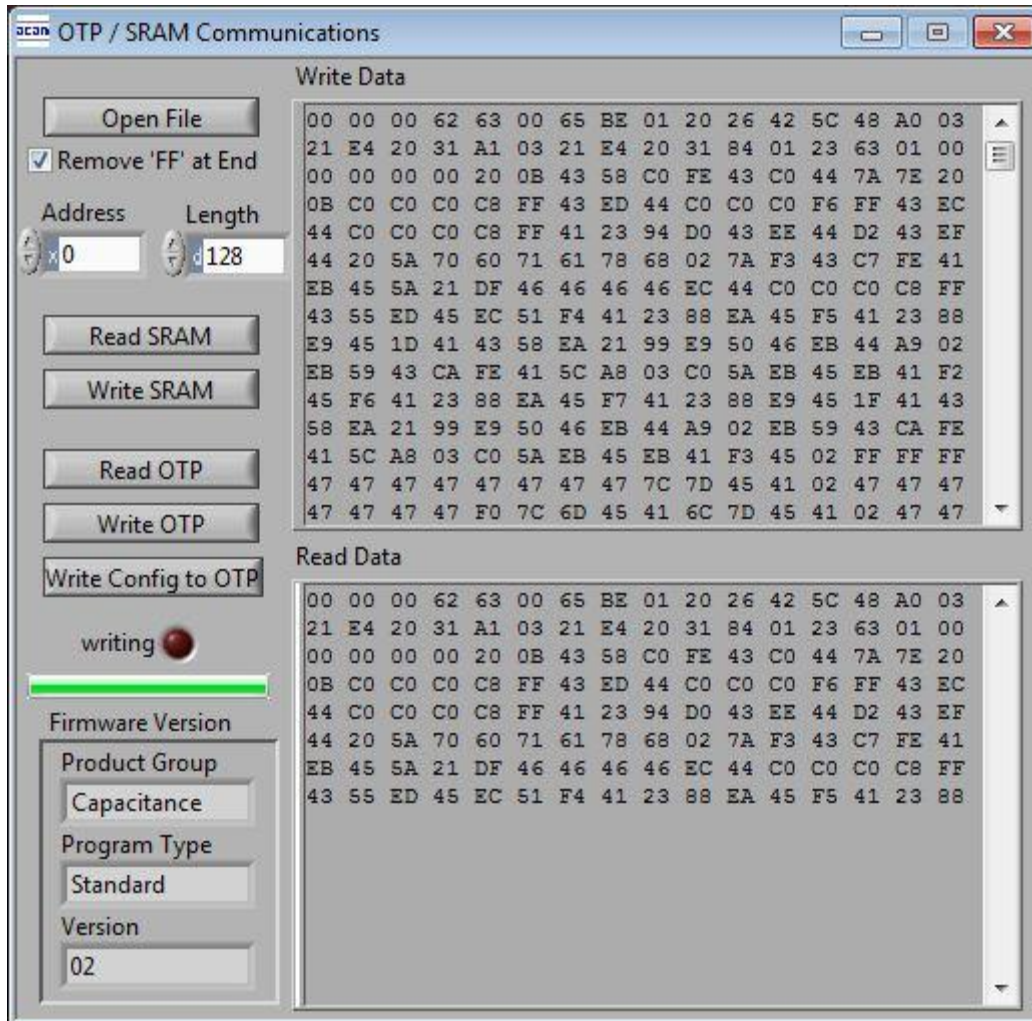In the 'File' Menu you can select 'OTP/SRAM' and the following window pops up:

Figure 3-15: OTP/SRAM Communication

Open File	Select and open a firmware file (.hex). The content is shown in the 'Write Data' window.
Read SRAM	Pressing this button, the content of the SRAM is read and shown in the 'Read Data' window. In 'Address' and 'Length' you can specify how many bytes you want read, starting at which address.
Write SRAM	Writes the firmware into the chips SRAM. The status of the write process is indicated by the green bar. The successful end is indicated by a pop-up window. For verification we recommend to read back the SRAM afterwards and compare it with the source.

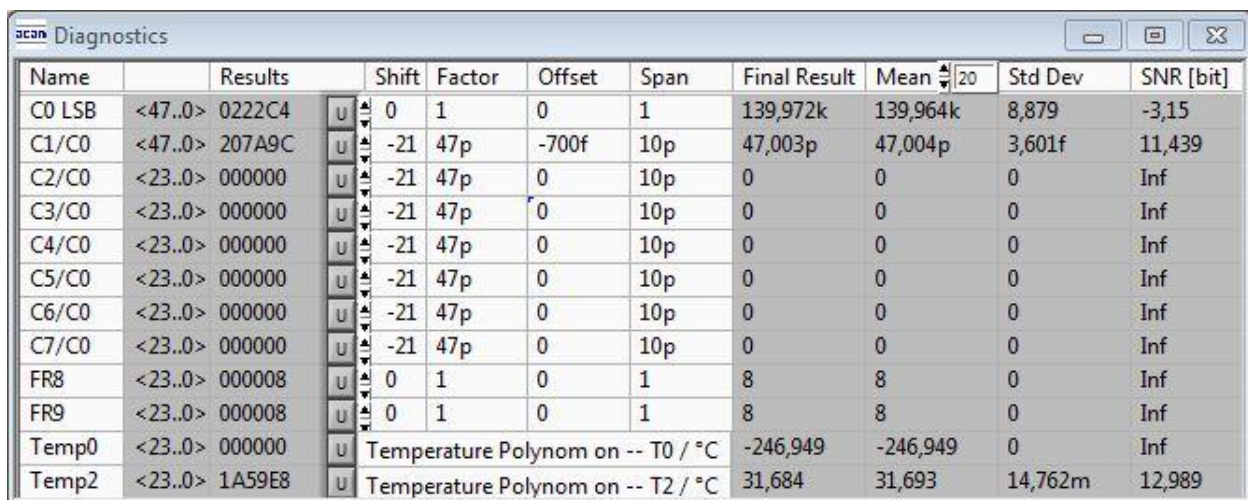| Read OTP	Pressing this button the content of the OTP is read and shown in the 'Read Data' window. In 'Address' and 'Length' you can specify how many bytes you want read, starting at which address.
Write OTP	Writes the firmware into the chips OTP. Attention: This option should be used only if you really want to test the OTP functionality. The status of the write process is indicated by the green bar. The successful end is indicated by a pop-up window. For verification we recommend to read back the OTP afterwards and compare it with the source.
Write Config to OTP	Writes to current content of the configuration registers into the reserved OTP space.
Firmware Version	In the firmware, a specific address is reserved to save 3 byte information about the application and the version of the software. The coding is specified in the pcap01a.h library (see assembler)

3.5.2   Diagnostics Window

Figure 3-16: Diagnostics Window

The 'Diagnostics' window is used for real-time numerical display of the measurement results. In principal it shows the content of the read registers. The content itself depends on the firmware. Figure 3-16 shows the content as it is given with the standard firmware. The first eight rows show the capacitance ratios, the last two rows show the temperature result (resistance ratio or linearized temperature).

The 'Diagnostics' window has 12 columns of information, defining labels, data format, resolution specification (white background) and results (grey background). The information in the white fields increase convenience of reading and is stored in the setting files (*.set).

All number may get a character to indicate the well known prefixes for denoting the factor in thousands ('p','f','a','k'...).

Name	Label for the register content, depends on the firmware.
Results	Raw hex data display of the result register content. The column before shows the width. The button column after shows whether the result is signed or unsigned.
Shift	This column shows the size of the fractional part of the fixed point number and the necessary shift. Depends on the fimware.
Factor	The factor is a scaling factor that allows to scale the result according to the reference capacitor. Factor = '1' gives back the initial capacitance ratio in column 'Final result' (see figure 3-16). '47p' scales the result to a 47 pF reference capacitor as it is assembled default on the evaluation board (see figure 3-17).
Offset	Offset to be added or subtracted in the evaluation software.
Span	Number that defines the maximum span of the sensor. Is relevant only for the calculation of the resultion in column SNR[bit]. Example figure 3-16b: The sensor has a base capacitance of 47 pF and varies by 10 pF. The 10 pF span are resolved with 11.4 bit.
Final Result	Display of the final result, scaled by 'Factor' and the 'Offset' added.
Mean	Display of the mean value. The sample size can be selected.
Std.Dev	Standard deviation of the 'Final Result'.
SNR	Signal-to-Noise ratio in bit, calculated as 'Span'/ 'Std.Dev.'

Name		Results			Shift	Factor	Offset	Span	Final Result	Mean ⧎20	Std Dev	SNR [bit]
C0 LSB	<47..0>	0222C4	U	⧎	0	1	0	1	139,972k	139,964k	8,879	-3,15
C1/C0	<47..0>	207A9C	U	⧎	-21	47p	-700f	10p	47,003p	47,004p	3,601f	11,439
C2/C0	<23..0>	000000	U	⧎	-21	47p	0	10p	0	0	0	Inf
C3/C0	<23..0>	000000	U	⧎	-21	47p	0	10p	0	0	0	Inf
C4/C0	<23..0>	000000	U	⧎	-21	47p	0	10p	0	0	0	Inf
C5/C0	<23..0>	000000	U	⧎	-21	47p	0	10p	0	0	0	Inf
C6/C0	<23..0>	000000	U	⧎	-21	47p	0	10p	0	0	0	Inf
C7/C0	<23..0>	000000	U	⧎	-21	47p	0	10p	0	0	0	Inf
FR8	<23..0>	000008	U	⧎	0	1	0	1	8	8	0	Inf
FR9	<23..0>	000008	U	⧎	0	1	0	1	8	8	0	Inf
Temp0	<23..0>	000000	U		Temperature Polynom on -- T0 / °C				-246,949	-246,949	0	Inf
Temp2	<23..0>	1A59E8	U		Temperature Polynom on -- T2 / °C				31,684	31,693	14,762m	12,989

Figure 3-16b: Diagnostics Window
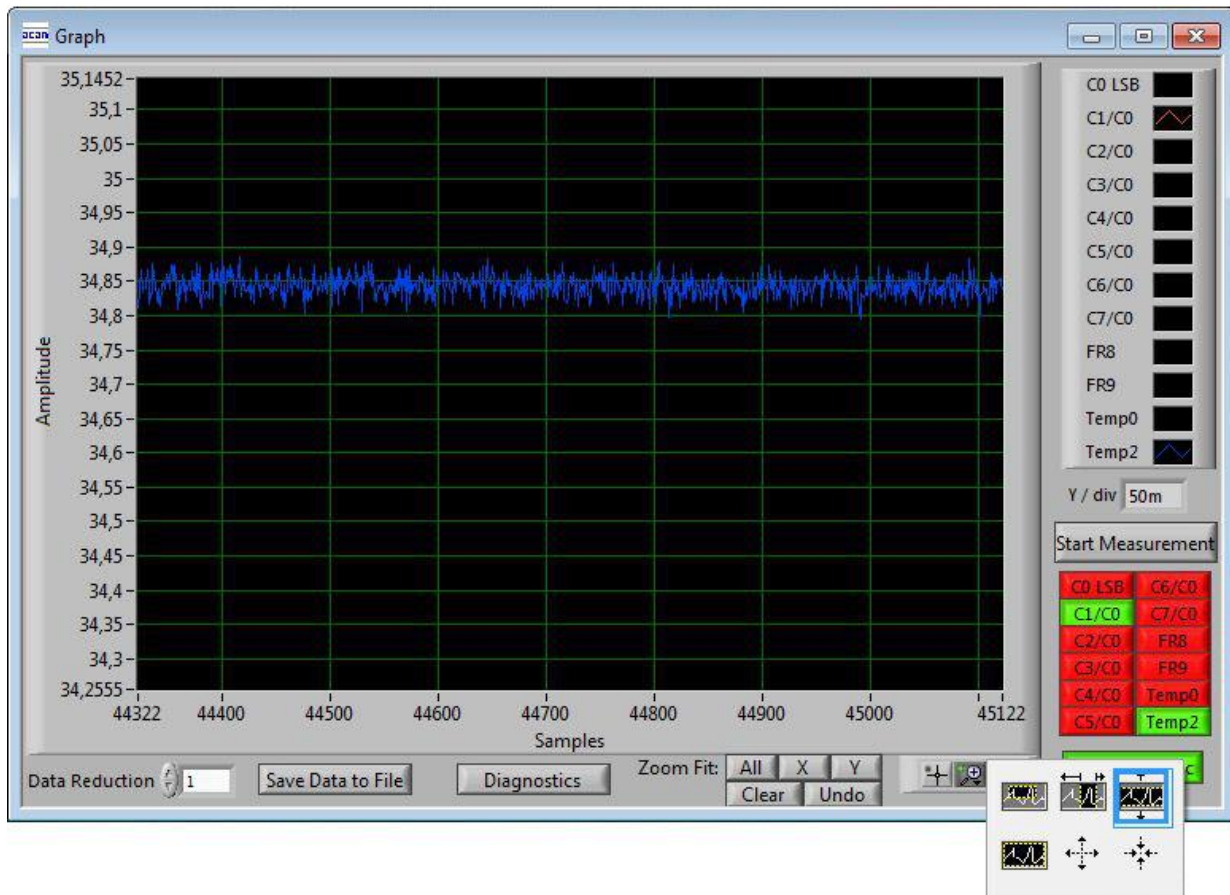
3.5.3   Graph Window

Figure 3-17: Graph Window

The data to be displayed are selected in the field at the bottom right. The labels in the buttons are the same as in the diagnostics window. To display data press the corresponding button so that it gets green. Below the graph are various options to zoom in/out, center or scale in other ways.

The data displayed can be stored into a text file. For long-term investigations it is possible to reduce the data displayed and stored. The field 'Data Reduction' allows to define the level of data reduction.

3.5.4   Registers Window

These windows display the configuration data in hexadecimal format as they are currently used. Also the result registers' content is shown in hexadecimal format, but updated only when the button is pressed. Finally, the various status bits are shown.
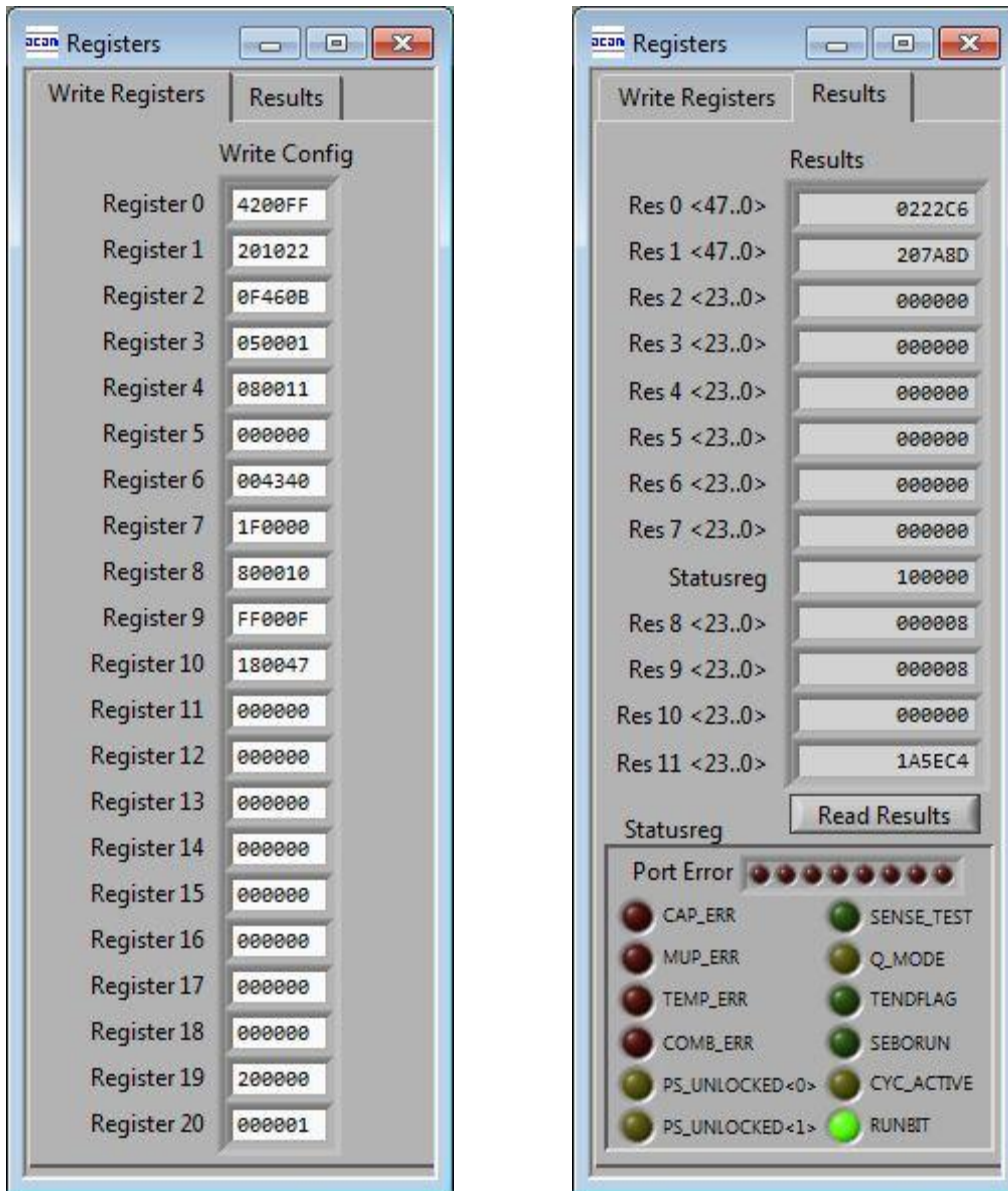
Figure 3-18: Write and Read Registers

3.5.5 Parameter Window

This window shows the content of the parameter registers and, if supported, the calibration data. The content strongly depends on the firmware. The standard firmware does not use calibration data but e.g. the humidity firmware and pressure firmware do.

The parameter registers are used to scale the pulse output interfaces and to tell the firmware whether differential sensors are used.
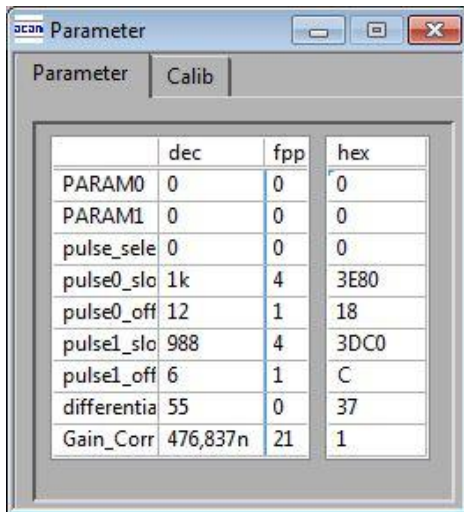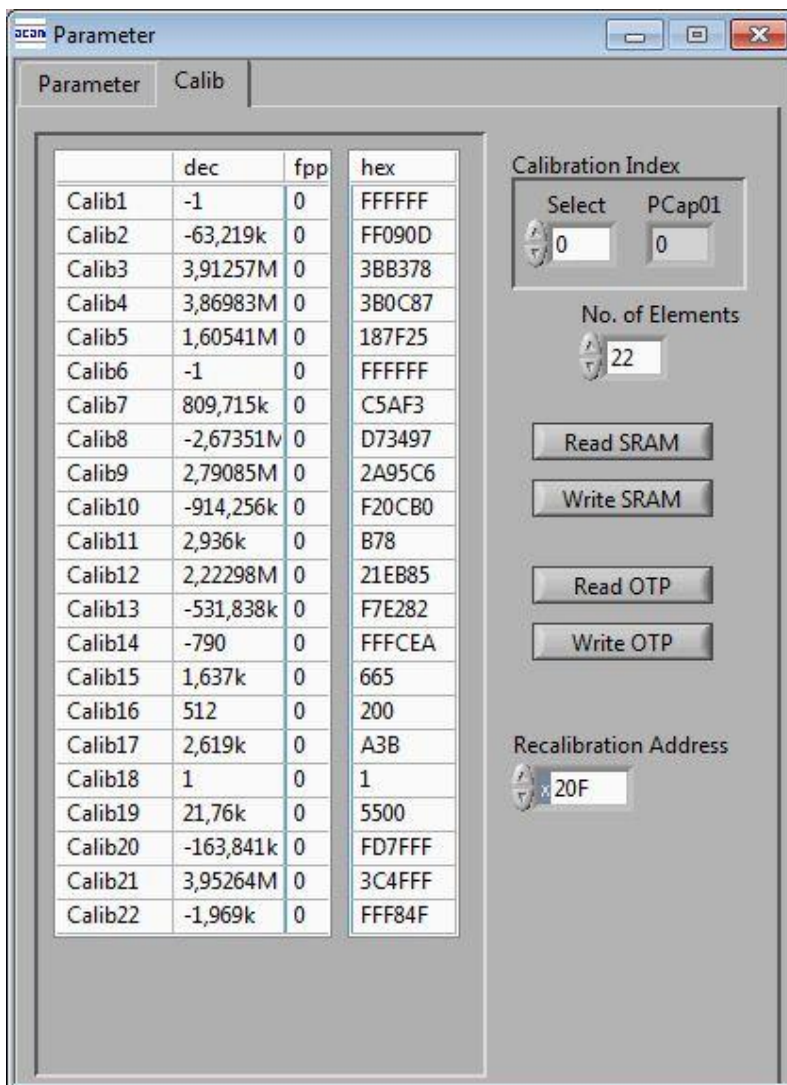
Figure 3-19: Parameter Window

Figure 3-20: Calibration Window

3.6   First Measurement: Step-by-Step

Step 1: The first to do after starting the evaluation software is to download a firmware into the PCapØ1. In the initial phase start with our standard firmware that calculates the capacitance ratios and resistance ratios. It automatically recognizes the operation mode and takes care of the set number of capacitors and the kind of connection. But it does no further processing.

Open the 'File' menu and select OTP/SRAM. The following 'OTP/SRAM communications window' opens. If you press 'Open File' initially the content in the 'data'-folder of the PCapØ1 program folder is opened. There you find the firmware PCapO1_standard.hex. Select this one.
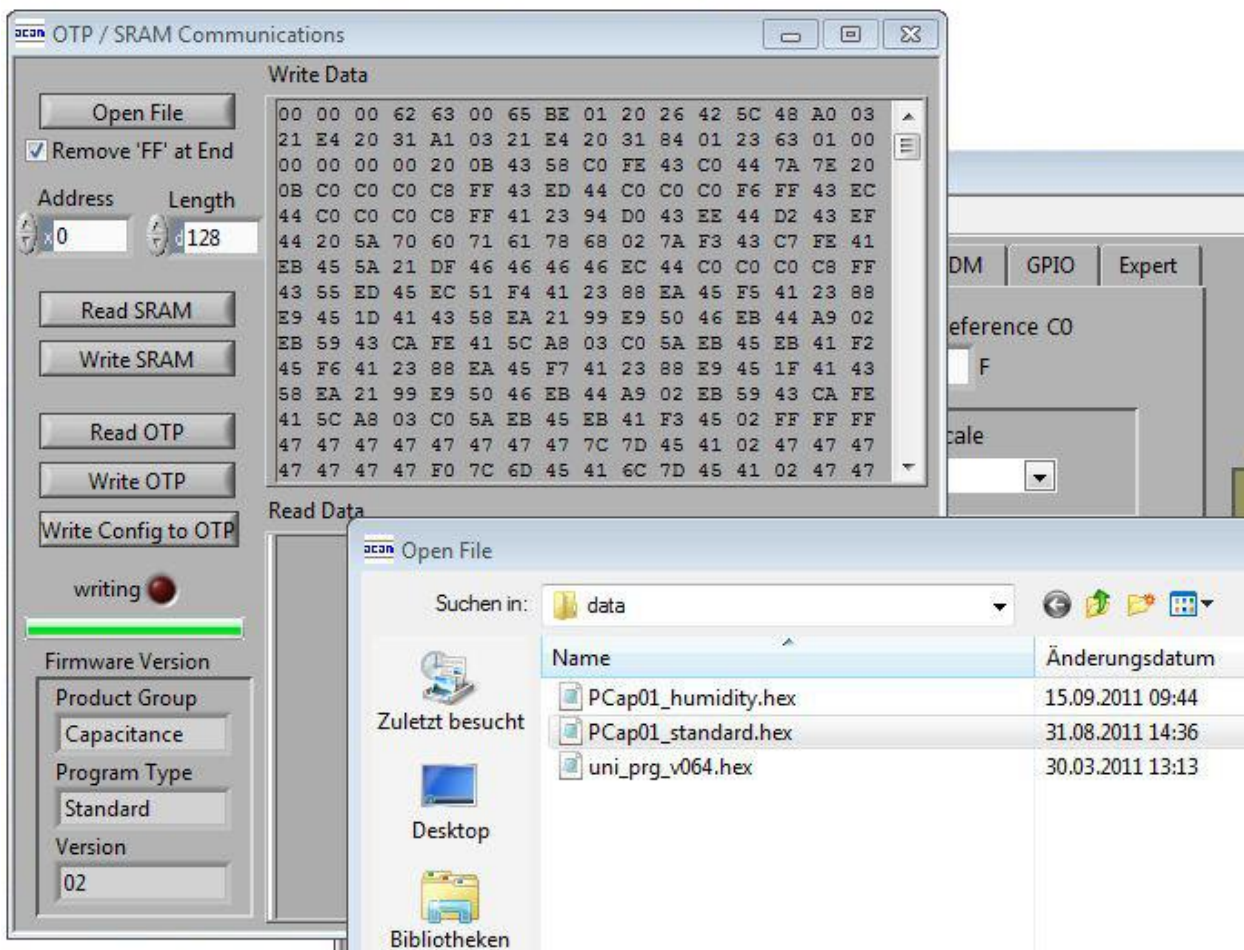

Figure 3-21 File menu

Press 'Write SRAM' and wait till the green bar indicates that the write process is finished. For verification press 'Read SRAM' to check if the content is correctly written into the

SRAM. In the 'Read Data' you should see the same content as in the first 128 byte of 'Write Data'.

Step 2: The next step you need to load a configuration file. This file contains the configuration registers content and the parameter registers content. Select in the 'File' menu item 'Open Config' and select the file PCapO1_initial.cfg. This configuration fits to the evaluation board in its initial assembly with floating 47pF capacitors.
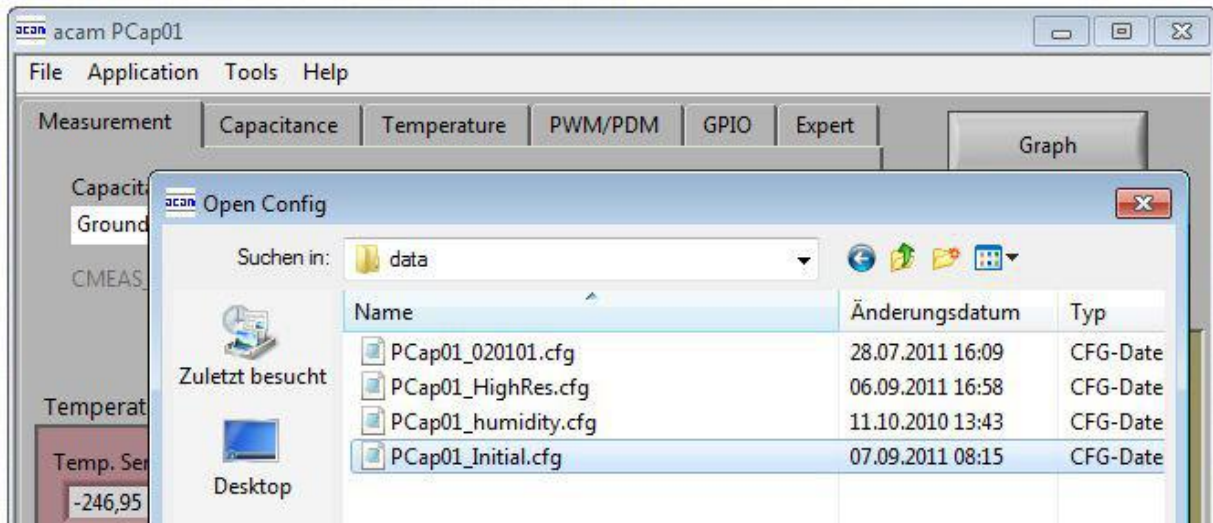

Figure 3-22: Load Configuration Data

Step 3: Finally, load the standard settings file to have a reasonable display in the Diagnostis window and Graph window.
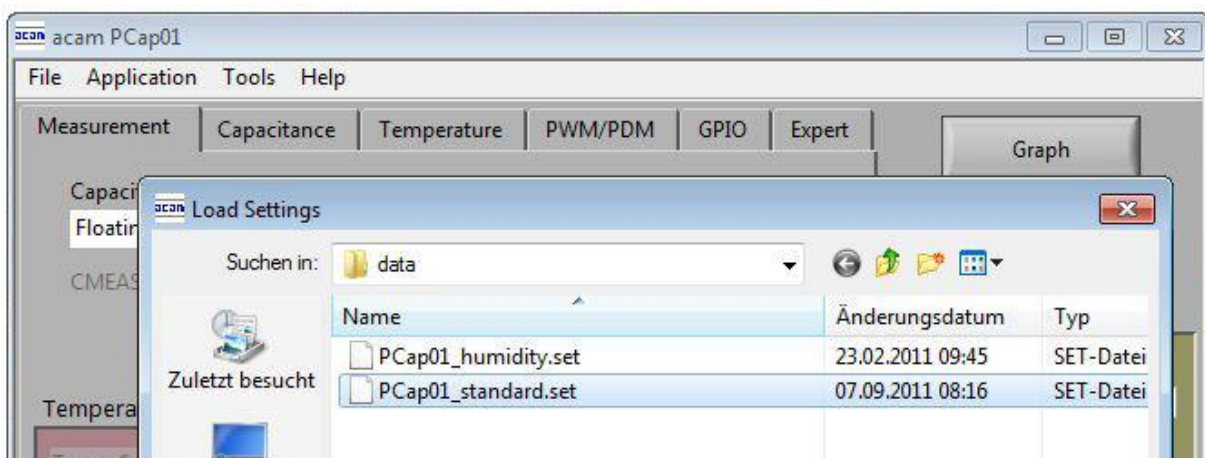

Figure 3-23: Load Settings Data

Step 4: Open the Diagnostics window and Graph window and press 'Start Measurement'.

4   Assembler

4   Assembler

Features

Platform:	Windows 2000, XP, Vista (32-Bit), Windows-7 (32 and 64)
Type of Assembler:	Single Path Assembler
Source-File:	*.asm, according to the PCapØ1 assembler syntax
Target-File:	Output. hex, downloadable Hex-File
Instruction:	<command> [<parameter1>, <parameter2>] For example: nop or add a, b or jCarC case1
Comment(s):	Single line comment: ; Multi line comment: <COMMENT> ... <ENDCOMMENT>
Includes:	<command> [<parameter1>, <parameter2>] For example: nop or add a, b or jCarC case1
Constants:	Constants can be defined with: CONST myConst8 15 The constants have to be declared before they are used for the first time.
Number formatting:	Numbers can be written in decimal or hexadecimal notation e.g.: load a, 16355 or load a, 0x3FE3
Jump labels:	To jump within the code, you can use jump labels together with the various conditional jump op-codes like jcd p1, jEQ p1 etc. E.g.: jCarC label1 label1: move a, b
Addressing:	Addressing is automatically done by the assembler. Please note that the maximum user program size is 4kB (including all header files, library files and calibration banks).
Subroutines:	Subroutines can be executed with the special opcode ‚jsb' and ‚jrt'.

4.1   Installing the Assembler

Insert the CD-ROM. Run Software\PCapO1_Assembler 2.0.3\setup.exe to install the program. Restart the computer if required. After the installation you will find in the START menu an item „PCapØ1 Assembler". Run this to start the assembler program.

4.2   Running the Assembler

In the START menu there is an item „PCapØ1 Assembler". Run this to start the assembler program.

4.2.1   Assembler Online Help

The following sections give just a short description of the assembler program. For a detailed description of the assembler software please use the online help of the program. The online help can be opened from the Menu or by pressing the F1 button.
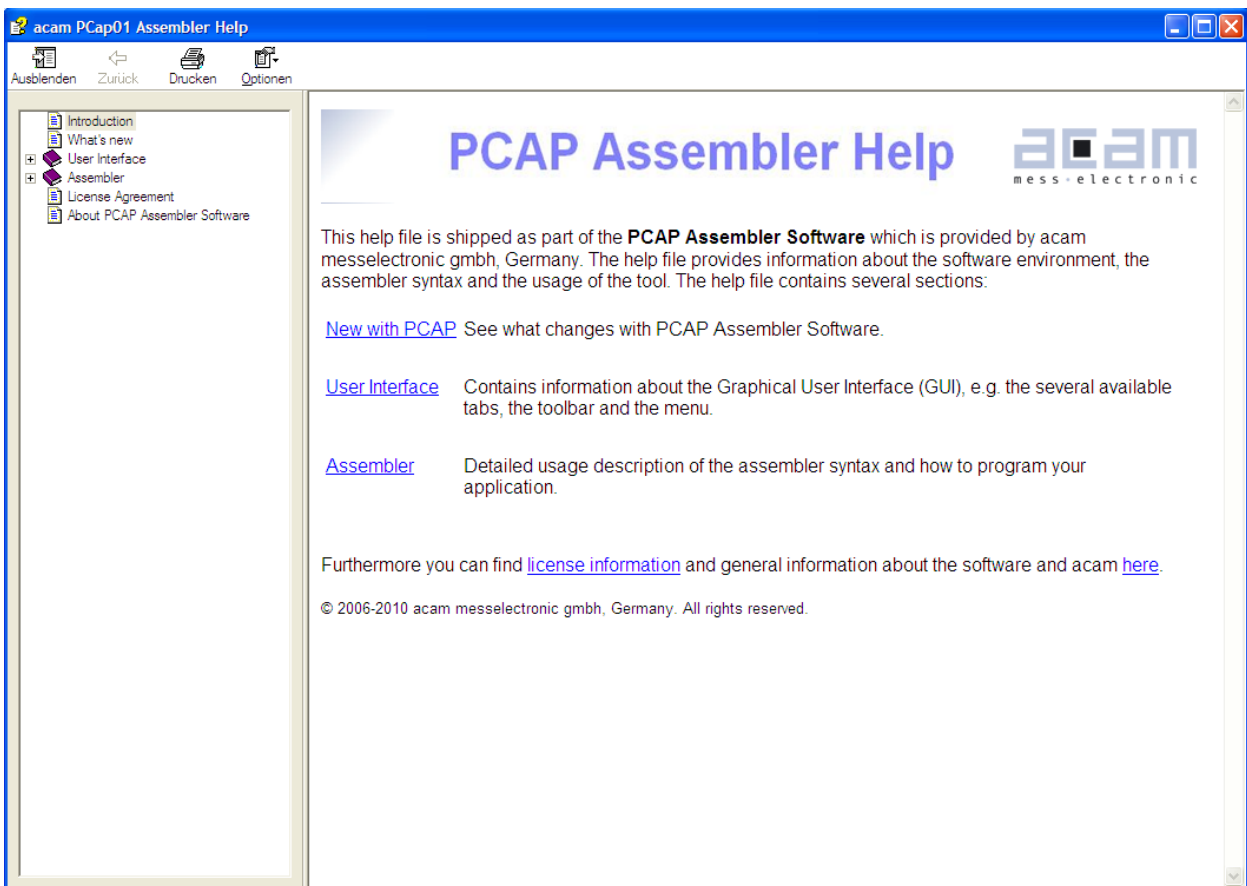
The following help window opens up:

Figure 4-1: Integrated online help of PCap01 Assembler
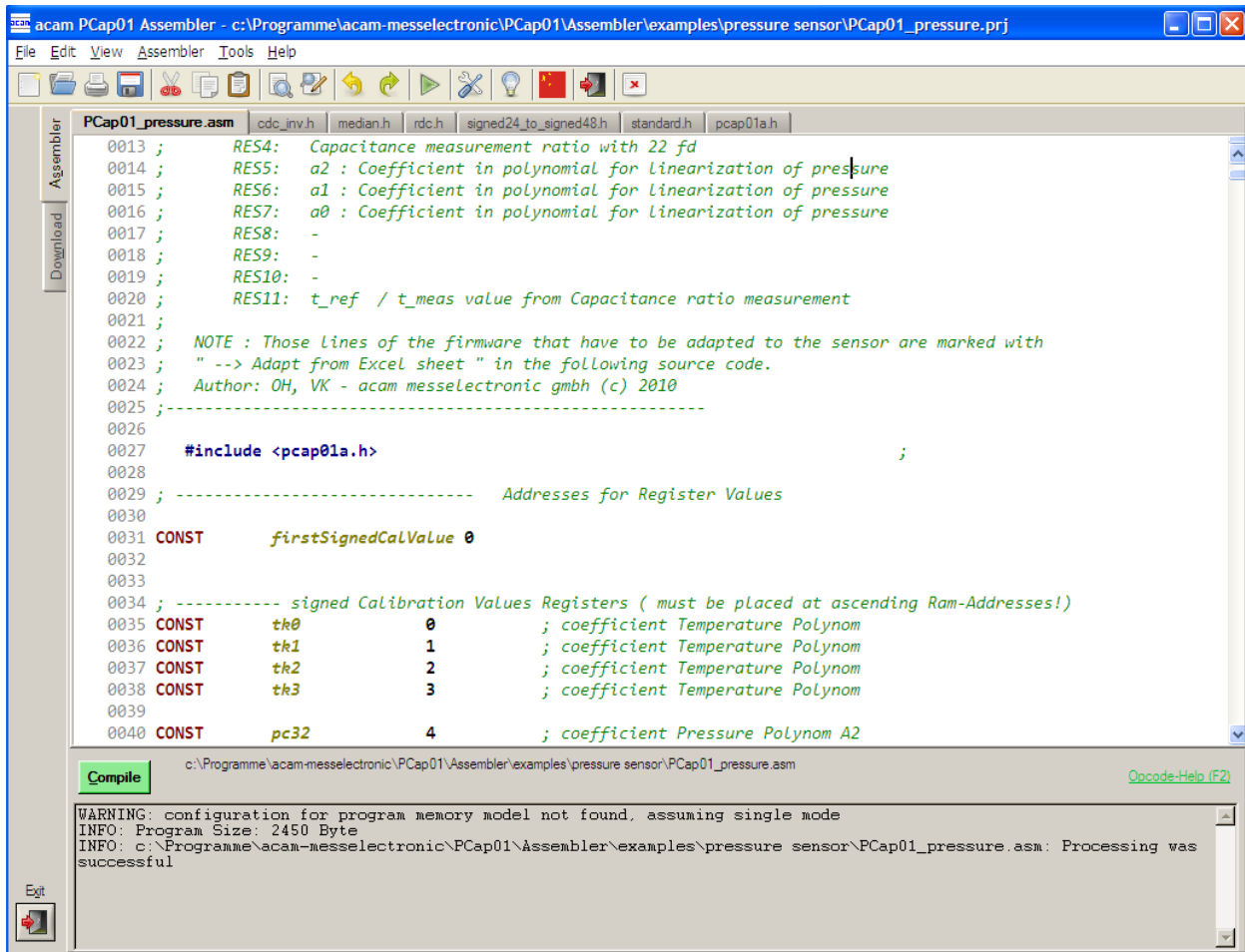
4.2.2 Assembler Page

Figure 4-2: Assembler source code window

The 'Assembler' tab is the main window to open all the source code and include files. It consists of a status message window at the bottom and buttons to operate the assembler. By clicking the green link labeled ‚Opcode-Help (F2)' an integrated online help pops up which provides detailed information about the available op codes.

The controls in detail:

Source code window:	This window is used as the source code and include files editor for the assembler listings (*.asm & *.h files). This is a pure editor window; any number of tabs can be opened in this window. This page allows the user to display and edit any file. The files can be manually opened, saved and closed. The location of the file displayed in the window currently is displayed next to the 'Compile' button.
	Additional features are line numbering, syntax highlighting and integrated comment handling. Standard editor tasks like cut, copy &

	paste or find & replace, etc. can be performed. You can modify the settings for the editor via the tool settings dialog. To know more about 'Include' files and how to use them, see section 4.2.2.1 below.
Assembler button:	The assembler button to compile the source code along with the respective header files is: Compile. On pressing the Compile button the source file is compiled to a downloadable hex code. If the compilation was successful a message is displayed accordingly. If the compiling fails, an error message occurs. If more than one source files are currently open, then the *.asm that will be compiled on pressing the Compile Button, has the filename in bold letters on the tab. It is the first opened .asm file in the window that is always compiled.
Save Project button:	When a source code (*.asm) file and all the respective include (*.h) files are open in the Assembler tab, then all the files can be combined and saved in a project file (*.prj), by pressing the Save Project button. The .prj file always takes the name of the *.asm file. For e.g. A source code file sample.asm uses 'include' files include1.h and include2.h. Then the corresponding prj file would be saved by the name sample.prj. Once saved, the project file can be opened from the menu File → Open project → *.prj file. This automatically closes all currently open files and opens all the files (*.asm and *.h) belonging to the project in one go. The *.prj file can also be opened and edited in this Window. Note1: The .asm and .h files can be in different folders or locations. The *.prj file stores the location of the file while saving the project. Note2: When the PCapØ1 Assembler software is closed, a .prj project file is automatically generated with all the files that are currently open in the window. The name of this project file will be that of the open source code file with a .prj extension. This .prj file will be used to open all these files again when the PCapØ1 Assembler software is started again.
Open Project button:	An already saved project (with a *.prj file), can be opened using this button
Status message window:	In this window there are the output messages of the assembler displayed. In case the assembly process was successful, a corresponding message appears with the path of the file that was compiled. If an error occurs while assembling, an error message appears together with the line number and the file name in which the error occurred. An error in any of the source code (*.asm) file or "include" (*.h) files is identified and intimated with the path of the file.
Opcode-Help:	Between the source code window and the status message window

	there can be found a green link labeled ‚Op code-Help (F2)'. By clicking this link another window pops up. The window contains the op code online help. Every supported op code is explained there in detail. The additional windows are based on the integrated Microsoft Windows ® Help system and can be operated separately. (Windows is a trademark of Microsoft Corporation)

Running the assembler in order to compile your source file is the first step when using the PCapØ1 Assembler Software.

An integrated 'examples' folder provides some readymade firmware code for downloading into the PCapØ1 chip. The standard firmware (.asm and .hex) is present in the 'standard' folder under 'examples'.

The user can also add his own examples to the 'Examples' folder with the respective prj file. The prj file can be written manually (See format of existing prj files in the Examples folder) or can be allowed to be generated by the assembler with the Save Project button. The Search-Examples dialog box can be accessed via the Help menu or by pressing F4 or by clicking on the 'Bulb' icon on the top. Only *.prj project files can be opened via the Search Examples option. To open individual files, use the File → Open option.

4.2.2.1　'Include' files

The basic idea of supporting 'include' files is to provide more modularity and flexibility within your code. That means, instead of packing all needed source code to a single assembler listing (*.asm files) you can roll out some pieces of code to the include files (also called header files with extension *.h). The advantage of this method is, that the readability of your code will be improved and that once written parts of code can be reused (e.g. configuration information).

To include parts of code by using a include file there are two steps needed.

Step 1: Create an include file (e.g. config.h).

Step 2: Declare the inclusion in your assembler listing as follows:　#include „config.h"

Include files that are in the same folder as the source file (*.asm) can be included with #include "filename.h". Include files that are in a path one level higher than the source file can be included with "../filename.h". If there is a set of common include files, they can be segregated in a folder called Lib in PCapO1 \Assembler\lib. The 'include' files in the lib

folder can be included in the program by using #include <filename.h>. They are automatically included from the lib folder.

On installing the assembler software, the PCapO1 \Assembler\lib is pre-installed. It contains all the library and header files that are used in the examples. You can add your library or header files to this folder.

The alternative method of using a single assembler files without using 'include' files at all is also supported.

4.2.3   Download Page

The 'Download' tab displays the compiled hex code that can be downloaded into the PCapØ1 chip. The hex-file is displayed in rows with 16 bytes per row. The location of the displayed hex file is shown at the top of the window. Now this hex file can be written into the OTP or the SRAM of the PCapØ1 chip for e.g. using the Front panel software.
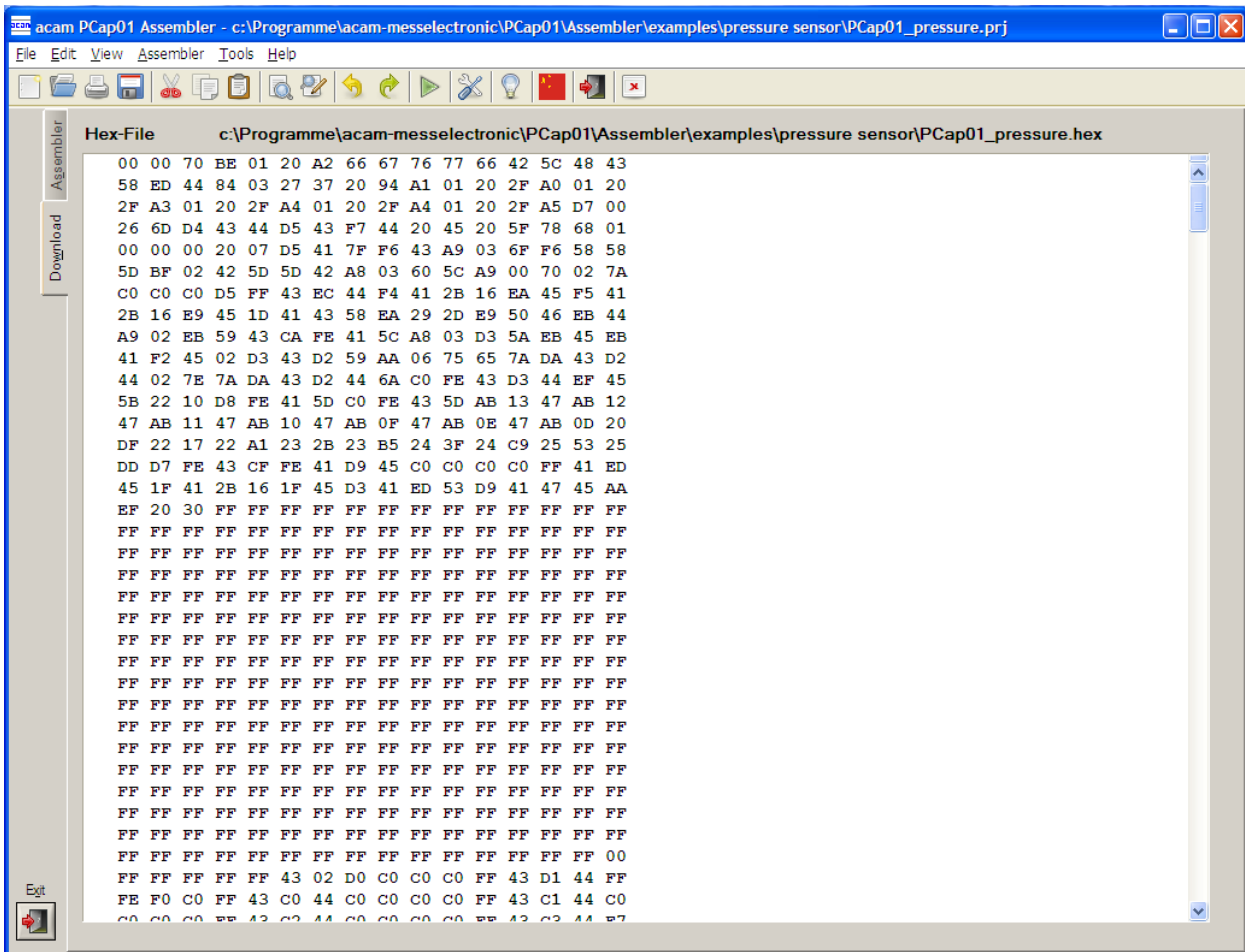

Figure 4-3: Hex-file display

4.2.4 Assembler Settings

Different settings for the assembler and the editor can be done by selecting the item "Settings" in the "Tools" menu. A screen appears that shows the following sheets:
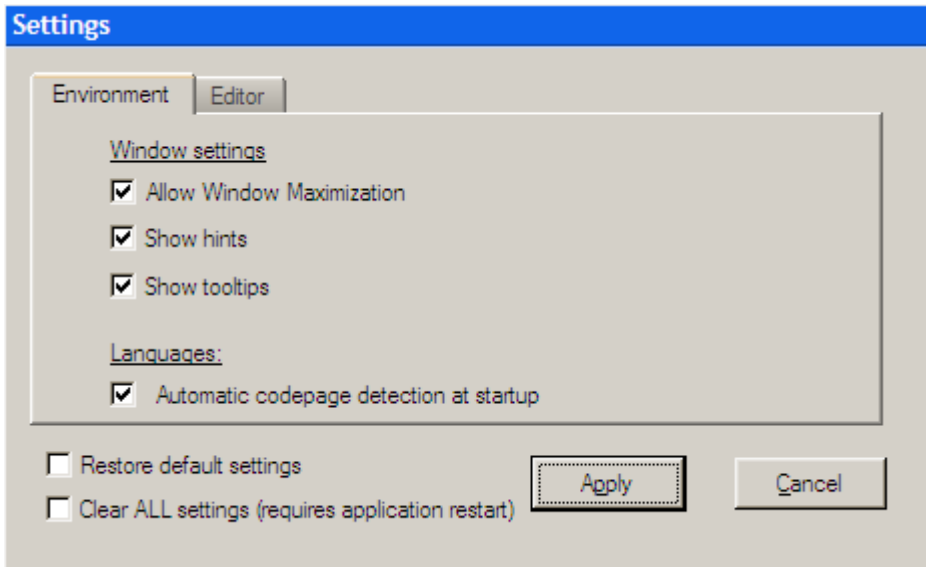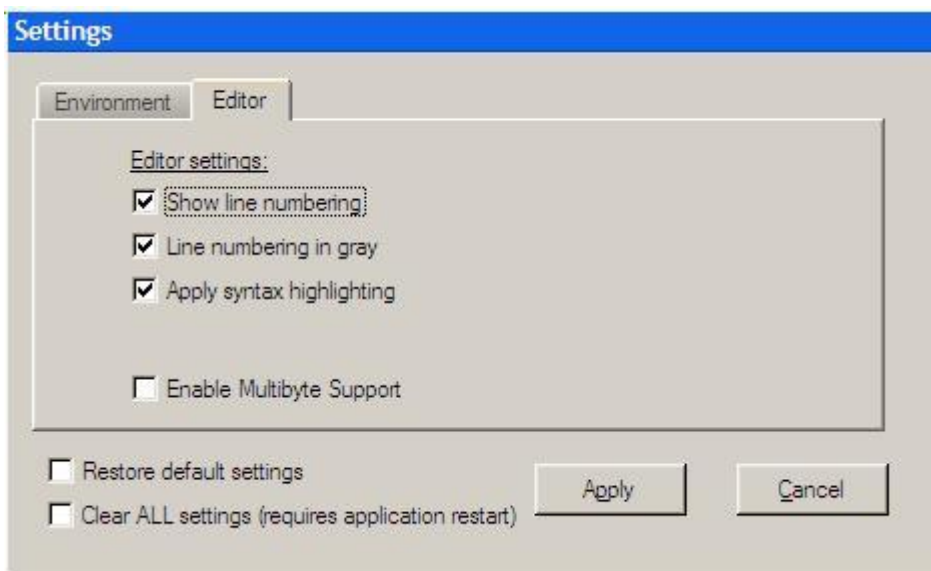

Figure 4-4: Environment settings

Figure 4-5: Editor settings

5 Miscellaneous

5.1 Literature Guide

Data Sheets

Title	Document-No	Date
PCapØ1Ax-03O1 Single-chip Solution for Capacitance Measurement with Standard Firmware version 3.0	DB_PCapO1Ax_03O1_en.pdf	May 2011
PCapØ1AK-01O1 Single-chip Solution for Humidity Measurement	DB_PCapO1AK-01O1_en.pdf	February 2011
PCapØ1-DSP Description of the digital signal processor	DB_PCapO1_DSP_e.pdf	Aug 2011

White Papers

Title	Document-No	Date
Linearization of Pressure Sensors with PCapØ1	WP005_e.pdf	Sep 2011

Application Notes

Title	Document-No	Date
Cable compensation	AN026 V1.0	In preparation

All available documents can be downloaded from the acam website at:

http://www.acam.de/download-center/picocap

5.2 Document History

26.02.2013   Correction in section 2.2, C1 = 33 nF COG, Version 0.3

12.12.2012   Minor updates and corrections, Version 0.2

27.10.2011   Assembler description added, Evaluation software description
                    expanded.

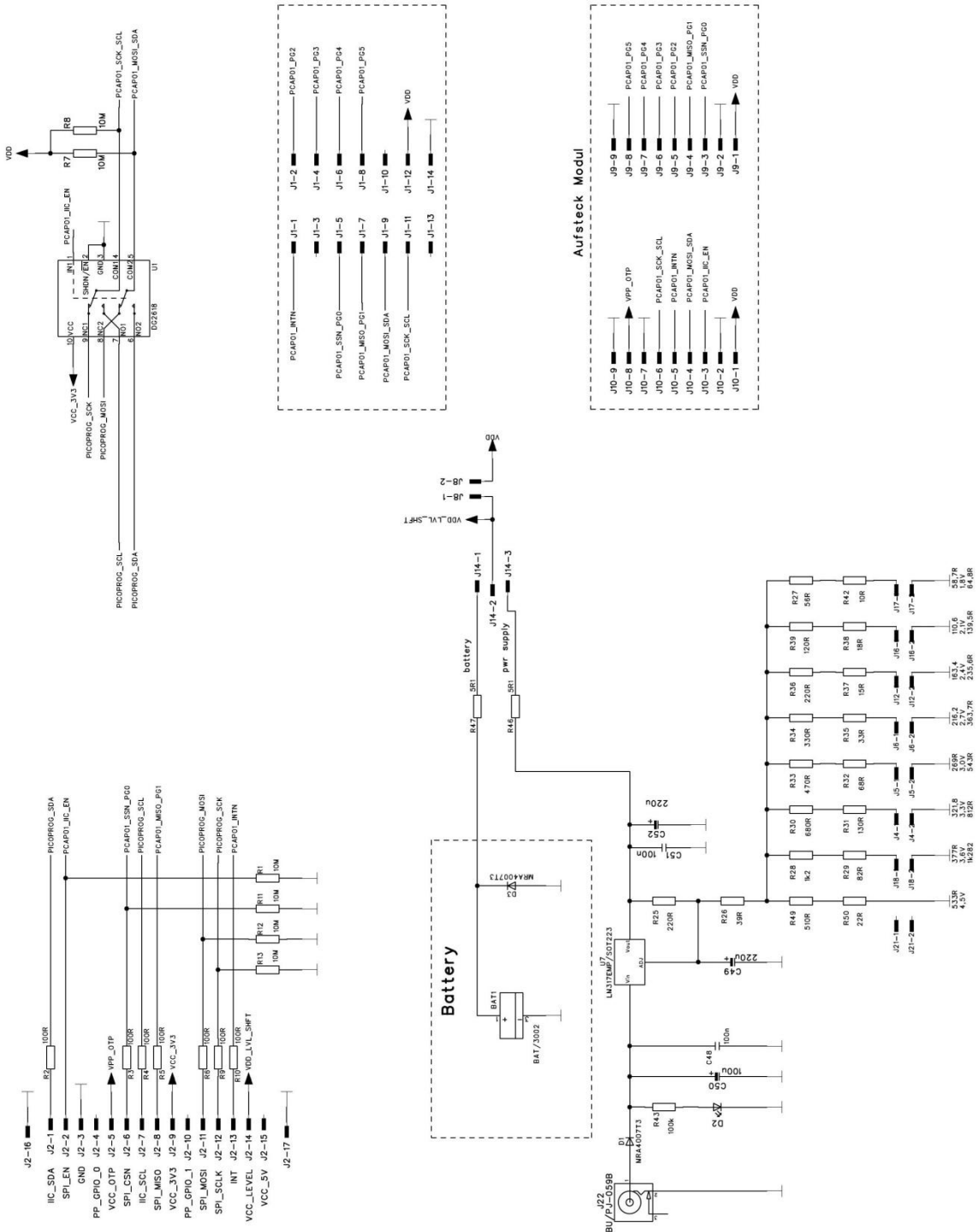29.04.2011   First release

6   Appendix

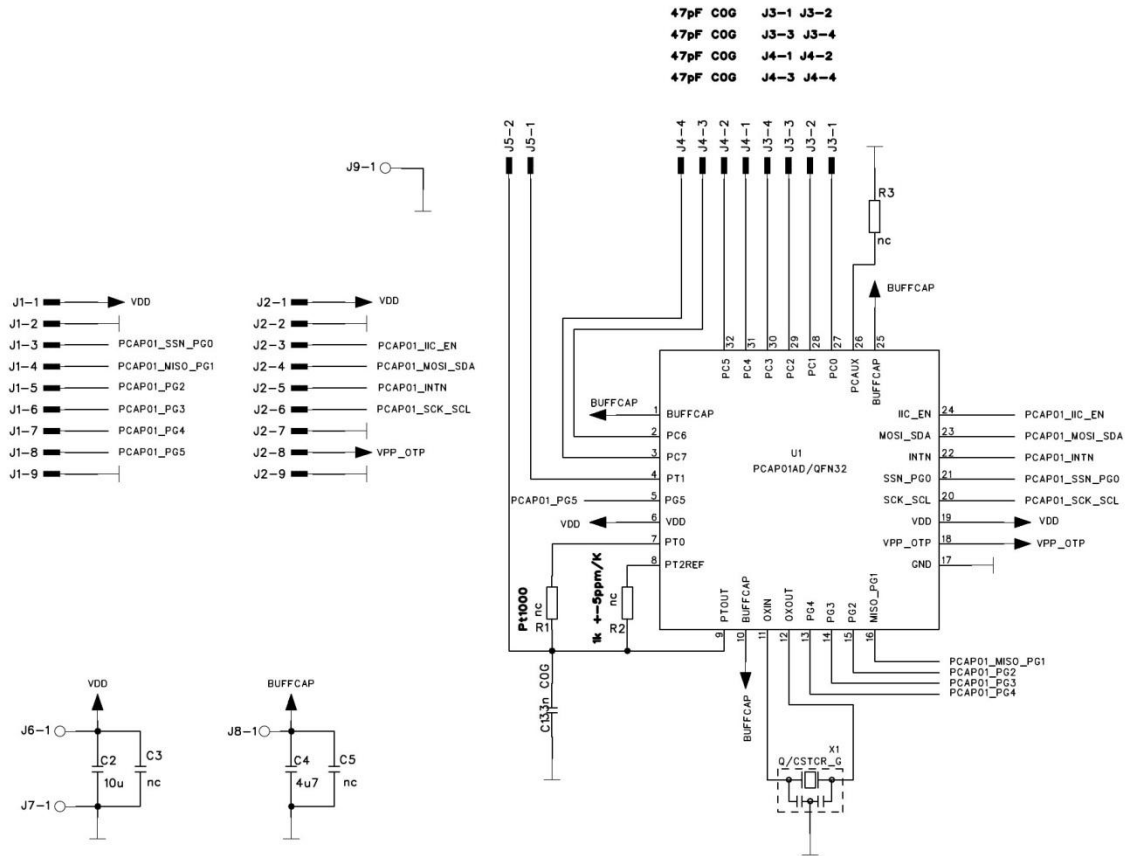## 6.1   Schematics

Figure 6-1: Motherboard schematics

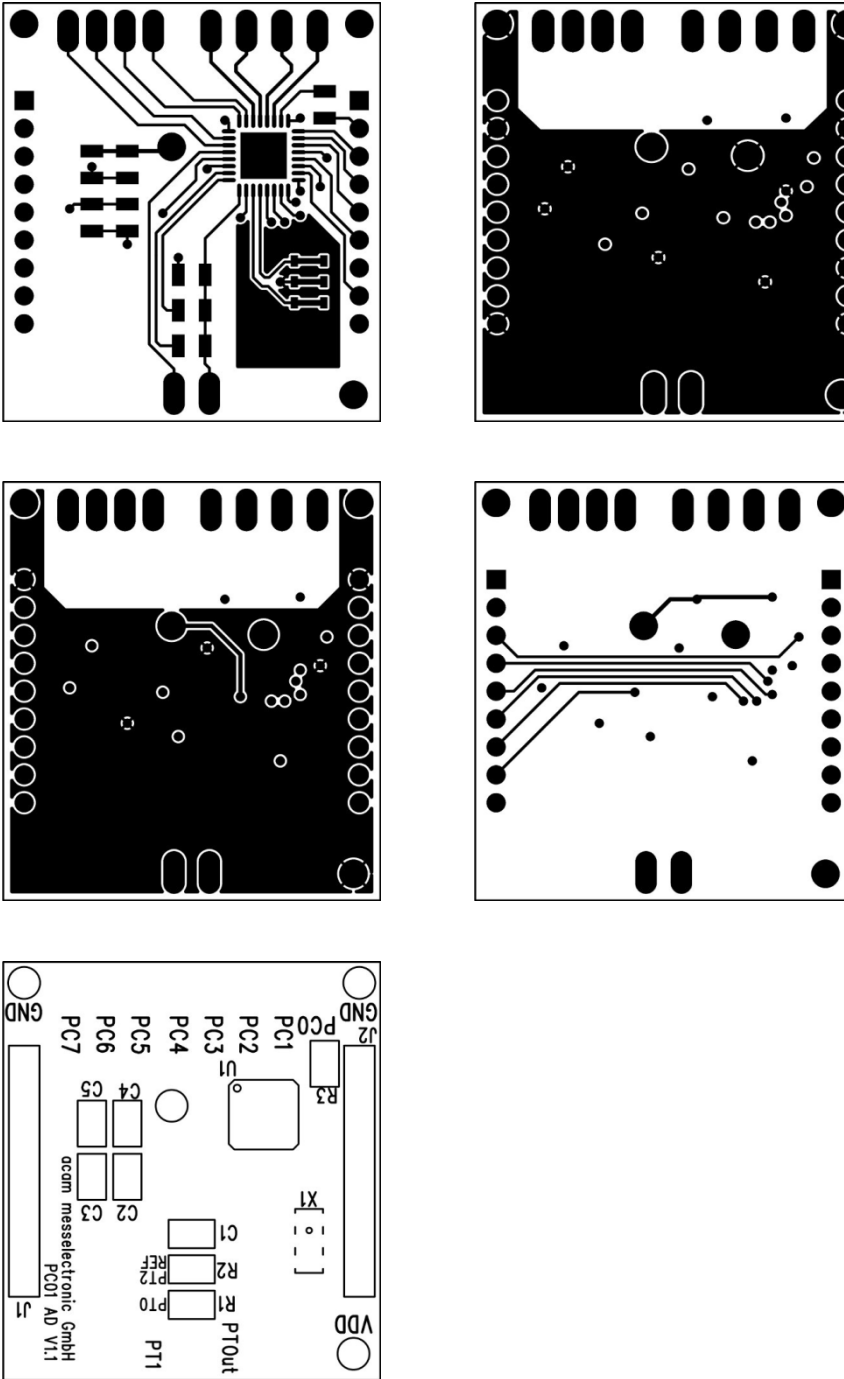Figure 6-2: PCaO1 AD module schematics

6.2 Layout

Figure 6-3: PCaO1 AD module layout