SAW Components

Application note

Digital barometric pressure sensor

Miniature sensors

Series/type:	**T5403** **B39000T5403P810**
Date:	February 21, 2014
Version:	2.1.A1

Interface selection

For communication with the digital pressure sensor T5403, two serial interfaces are available. Depending on the state of pin SEL, either the two-wire protocol **I²C** or the four-wire-protocol **SPI** can be used for transactions.

Logic state SEL	Interface protocol
low	SPI
high	I²C

Pin configuration

The following table describes the available pins and their functionality with respect to communication of T5403. Please note that the functionality of pins 5 to 8 depends on the applied interface protocol while the functionality for pins 1 to 4 is fixed regardless of the interface.

Pin	Name	I/O	Function in I²C mode	Function in SPI mode
1	VDD	Supply	Supply voltage	
2	SEL	I	Interface protocol selection	
3	GND	Supply	Ground supply	
4	EOC	O	End of conversion	
5	RST&SS	I	Reset	Slave select
6	MISO	O	High-Z	Master In Slave Out (MISO)
7	SDA/MOSI	I/O	I²C data signal	Master Out Slave In (MOSI)
8	SCL/SCLK	I	I²C clock signal	SPI clock signal

Electrical interface characteristics

Unless otherwise specified, the following table depicts the electrical interface characteristics.

		Min.	Typ.	Max.	Unit	Comment
External cpacitance between V_{DD} and GND	C_{VDD}	90	—	—	nF	
Capacitances of I/O pins	C_{IO}	—	—	10	pF	
Voltage input low level	V_{IL}	—	0	$0.3 \cdot V_{DD}$	V	
Voltage input high level	V_{IH}	$0.7 \cdot V_{DD}$	V_{DD}	—	V	
Voltage output low level	V_{OL}	—	0	$0.2 \cdot V_{DD}$	V	I_{OL} = 1 mA
Voltage output high level	V_{OH}	$0.8 \cdot V_{DD}$	V_{DD}	—	V	I_{OH} = 1 mA

Registers

There are two kinds of registers implemented for communication with T5403. Data registers are used in order to get data from the device. Control registers are intended to provide the ability of choosing options and issuing commands for the pressure sensor.

Information read from any other bit than the ones described in the following does not provide reasonable data! Accessing other registers than the ones mentioned in this chapter by write or read sequences might result in unintentional chip behaviour and the potential requirement of a hardware reset!

Summary of available data registers:

Data register name	Register address (hex)	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
data_msb	F6	data_msb<7:0>							
data_lsb	F5	data_lsb<7:0>							
calib_data	8E-A1	Calibration data							
i2c_slave	88	X[1)]	1	1	1	0	1	1	1
iface_settings	87	X	0 spi_pol	0 spi_ph	0 ss_pol	1 pol	X	X	X

1) X means don't care

Two bytes of data need to be evaluated in order to achieve raw pressure and temperature data according to the bit order in the table above.

Calibration data consists of 20×8 bits, which are mandatory for correct calculation of absolute pressure and temperature values.

Please note that the calibration coefficients **c5 to c10** have to be regarded as **signed 16-bit** values, while the remaining coefficients **c1 to c4** are provided as **unsigned 16-bit** integer values.

One coefficient is stored in two data registers and needs to be composed according to the table below.

Overview of calibration registers:

Data register name	Register address (hex)	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
c1_LSB	8E	calibration_coefficient_c1<7:0>							
c1_MSB	8F	calibration_coefficient_c1<15:8>							
c2_LSB	90	calibration_coefficient_c2<7:0>							
c2_MSB	91	calibration_coefficient_c2<15:8>							
c3_LSB	92	calibration_coefficient_c3<7:0>							
c3_MSB	93	calibration_coefficient_c3<15:8>							
c4_LSB	94	calibration_coefficient_c4<7:0>							
c4_MSB	95	calibration_coefficient_c4<15:8>							
c5_LSB	96	calibration_coefficient_c5<7:0>							

Data register name	Register address (hex)	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
c5_MSB	97	calibration_coefficient_c5<15:8>							
c6_LSB	98	calibration_coefficient_c6<7:0>							
c6_MSB	99	calibration_coefficient_c6<15:8>							
c7_LSB	9A	calibration_coefficient_c7<7:0>							
c7_MSB	9B	calibration_coefficient_c7<15:8>							
c8_LSB	9C	calibration_coefficient_c8<7:0>							
c8_MSB	9D	calibration_coefficient_c8<15:8>							
c9_LSB	9E	calibration_coefficient_c9<7:0>							
c9_MSB	9F	calibration_coefficient_c9<15:8>							
c10_LSB	A0	calibration_coefficient_c10<7:0>							
c10_MSB	A1	calibration_coefficient_c10<15:8>							

The slave address for the I^2C interface is set to a default value of 0x77. It is stored in register 0x88 and can be read out.

The register iface_settings involves the definition for SPI conditions as well as the polarity (bit3: pol) of the interrupt signal (end of conversion) and the reset trigger. The next table represents the meanings of the values stored in iface_settings.

Bit	Value	Consequence
spi_pol	0	clock is high active
spi_ph	0	events on data lines are triggered on rising clock edge
ss_pol	0	slave select is low active I^2C: reset on RST pad triggered with log. low pulse
pol	1	end of conversion provides high signal pulse

Overview of control registers:

Control register name	Register address (hex)	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
command	F1	X[1]	X	**0**[2]	mode<1:0>		pt<1:0>		sco
reset	F0	reset<7:0>							

1) 'X' means don't care
2) value '0' is mandatory for applying command correctly

mode<1:0>: measurement modes

There are four different measurement modes available with T5403 for pressure measurement. The noise level can be selected by the mode option within the command for start of conversion.

mode<1:0>	Measurement mode	Typ. RMS noise[1)] [Pa]	Conversion time [ms]
00	Low	5.1	2
01	Standard	3.5	8
10	High	3.2	16
11	Ultra high[2)]	2.7	66

1) The noise data is derived from the standard deviation of ten successive samples
2) Requires the average of two temperature measurements: one before and one after the pressure measurement.

For a temperature measurement, there is no option with respect to the measurement mode, i. e. mode<1:0> is don’t care for triggering a temperature measurement. The conversion time for a temperature measurement is 2 ms.

Please keep in mind that you have to add the max. wakup time of 2.5 ms to the conversion time to get the overall delay between start of conversion and read out the raw measurement data.

pt<1:0>: measurement options

There are two options for measuring either pressure or temperature depending on register pt:

pt<1:0>	Measurement sequence
00	execute pressure measurement
01	execute temperature measurement

The ressults of a pressure or temperature measurement are stored in the data registers data_msb (0xF6) and data_lsb (0xF5).

pt<1:0>	Data format	Data registers
00	16 bits unsigned integer	data_msb, data_lsb (0xF6, 0xF5)
01	16 bits signed integer (two’s complement)	data_msb, data_lsb (0xF6, 0xF5)

sco: start of conversion

When sending a byte to register 0xF1 and, thus, applying commands, a measurement and data conversion is triggered by setting bit ‘**sco**’ to logical ‘1’.

Reset

A software reset can be triggered by sending 0x73 to the reset register (0xF0).

In I^2C mode, a reset can be issued by applying a logical low pulse to pad RST. The polarity for this pulse is defined by pol=1 (data register 0x87).

Interrupt

The potential on EOC pad rises at the end of measurement and falls as soon as the registers holding the measured data are read (regardless of pt register value).

Value calculation

Since the digital pressure sensor T5403 does only provide raw data for pressure and temperature, an additional calculation has to be executed in order to optain the actual values in Pa and ˚C. This calculation can be done using the calibration data stored in the memory of the device in the following way:

The actual temperature t_a can be calculated with the calibration parameters c_1,c_2 and the raw reading t_r:

$$t_a = \left(100 \cdot \left(\frac{c_1 \cdot t_r}{2^8} + c_2 \cdot 2^6\right)\right) / 2^{16}$$

The unit of t_a is centi degree Celsius. To get t_a in degree Celsius divide it by 100. To calculate the actual pressure value p_a from the raw reading p_r the coefficients c_3 to c_{10} are needed:

$$S = c_3 + \frac{c_4 \cdot t_r}{2^{17}} + \left(\frac{c_5 \cdot t_r}{2^{15}} \cdot t_r\right) / 2^{19} + \left(\left(\left(\frac{c_9 \cdot t_r}{2^{15}} \cdot t_r\right) / 2^{15}\right) \cdot t_r\right) / 2^{16}$$

$$O = c_6 \cdot 2^{14} + \frac{c_7 \cdot t_r}{2^3} + \left(\frac{c_8 \cdot t_r}{2^{15}} \cdot t_r\right) / 2^4 + \left(\left(\frac{c_9 \cdot t_r}{2^{15}} \cdot t_r\right) / 2^{16}\right) \cdot t_r$$

$$X = \frac{S \cdot p_r + O}{2^{14}}$$

$$p_a = X + ((X - 75000)^2 / 2^{16} - 9537) \cdot c_{10} / 2^{16}$$

The unit of p_a is Pascal. Please note that some multiplication results are 32 bit long. t_r and c_5 to c_{10} are 16-bit signed integers (two’s complement), all other coefficients are 16-bit unsigned integer values. A calculation example with sample data is given on the next page. Pseudocode for calculating t_a and p_a is given below:

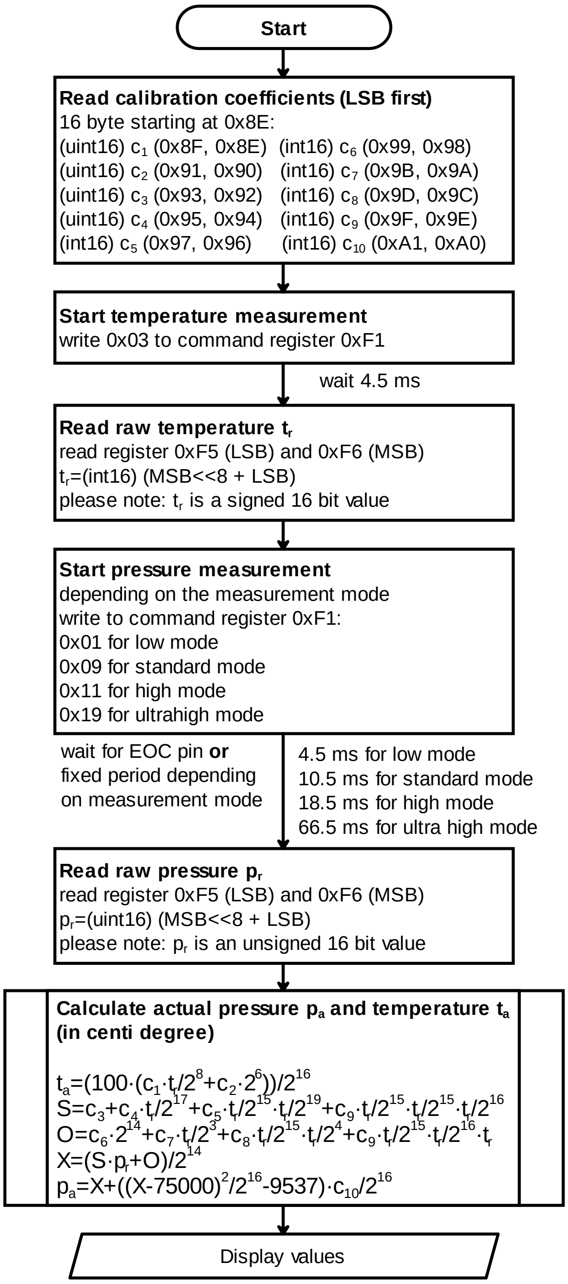
```
uint16 c1,c2,c3,c4;      //declare unsigned calibration coefficients
int16 c5,c6,c7,c8,c9,c10; //declare signed calibration coefficients
uint16 p_r;              //declare raw pressure
int16 t_r;               //declare raw temperature
int32 t_a,p_a,S,O,X;    //declare act. temperature and pressure,interm.variables
...
```

start conversion and read p_r and t_r

```
...
//calculate temperature
t_a=((int32)c1*t_r/0x100+(int32)c2*0x40)*100/0x10000;
```

```
//calculate pressure

S=c3+(int32)c4*t_r/0x20000+(int32)c5*t_r/0x8000*t_r/0x80000
      +(int32)c9*t_r/0x8000*t_r/0x8000*t_r/0x10000;

O=c6*0x4000+(int32)c7*t_r/8+(int32)c8*t_r/0x8000*t_r/16
      +(int32)c9*t_r/0x8000*t_r/0x10000*t_r;

X=(S*p_r+O)/0x4000;

p_a=X+((X-75000)*(X-75000)/0x10000-9537)*c10/0x10000;
```

Example calculation



Read out 20 byte starting at register 0x8E. Example data is given in the last two columns of the following table:

Type	Coeff.	Register	Hex. val.	Dec. val.
uint16	c_1	0x8F, 0x8E	0xAE89	44681
uint16	c_2	0x91, 0x90	0x7079	28793
uint16	c_3	0x93, 0x92	0xB3A7	45991
uint16	c_4	0x95, 0x94	0x86B4	34484
int16	c_5	0x97, 0x96	0x1807	6151
int16	c_6	0x99, 0x98	0x0924	2340
int16	c_7	0x9B, 0x9A	0xFB62	-1182
int16	c_8	0x9D, 0x9C	0x1906	6406
int16	c_9	0x9F, 0x9E	0x012C	300
int16	c_{10}	0xA1, 0xA0	0xFEFE	-258

Start a temperature measurement and read out two bytes at register 0xF5 (LSB) and 0xF6 (MSB). For this example MSB=0xE6 and LSB=0x43.

$$t_r = (\text{int16})(0\text{xE}6 \cdot 2^8) + 0\text{x}43 = 0\text{xE}643 = -6589$$

Start a pressure measurement and read out two bytes at register 0xF5 (LSB) and 0xF6 (MSB). For this example MSB=0x88 and LSB=0xD7.

$$p_r = (\text{uint16})(0\text{x}88 \cdot 2^8) + 0\text{xD}7 = 0\text{x}88\text{D}7 = 35031$$

Calculate the actual temperature t_a with c_1, c_2 and t_r:

$$t_a = \left(100 \cdot \left(\frac{44681 \cdot -6589}{2^8} + 28793 \cdot 2^6\right)\right) / 2^{16}$$

$$t_a = 1057$$

The unit of t_a is centi degree Celsius, so the actual temperature is 10.57 °C (pressure see next page).

The pressure p_a can be calculated with c_3 to c_{10}, t_r and p_r:

$$S = 45991 + \frac{34484 \cdot -6589}{2^{17}} + \left(\frac{6151 \cdot -6589}{2^{15}} \cdot -6589\right)/2^{19} + \left(\left(\left(\frac{300 \cdot -6589}{2^{15}} \cdot -6589\right)/2^{15}\right) \cdot -6589\right)/2^{16}$$

$$S = 44272$$

$$O = 2340 \cdot 2^{14} + \frac{-1182 \cdot -6589}{2^{3}} + \left(\frac{6406 \cdot -6589}{2^{15}} \cdot -6589\right)/2^{4} + \left(\left(\frac{300 \cdot -6589}{2^{15}} \cdot -6589\right)/2^{16}\right) \cdot -6589$$

$$O = 39802587; \text{ (39802964, using integer arithmetics)}$$

$$X = \frac{44272 \cdot 35031 + 39802587}{2^{14}} = 97088$$

The actual pressure p_a in Pa is then:

$$p_a = 97088 + ((97088 - 75000)^2/2^{16} - 9537) \cdot -258/2^{16} = 97096$$

I²C mode

In order to run T5403 in I²C mode, V_{DD} needs to be applied to pins SEL and RST.

Unless otherwise specified, the I²C interface is compatible to NXP UM10204 I²C-bus specification and user manual Rev. 03 (19 June, 2007). Standard, fast, fast plus and high-speed modes are supported. For further details concerning I²C protocol basics, please refer to the specification document mentioned above.

		Min.	Typ.	Max.	Unit	Comment
SCL clock frequency	f_{SCL}	0	—	3.4	MHz	Standard, fast, fast-plus and Hs-mode
Voltage output low level	V_{OL}	—	0	$0.2 \cdot V_{DD}$	V	I_{OL} = 3 mA
Pull-up resistor on SDA and SCL	R_{PU}	1	4.7	—	kΩ	
Capacitive load for each bus line		—	—	100	pF	

For starting any measurement and conversion using the I²C interface, the device needs to be addressed by its predefined slave address 0x77. Afterwards, the corresponding control register needs to be applied and the desired measurement resolution and sequence has to be submitted as shown in Figure 1.

Start	7-bit slave address of device							R/W	ACKS	8-bit register address (0xF1)								ACKS
S	1	1	1	0	1	1	1	0		1	1	1	1	0	0	0	1	
								WR=0		control byte								

➔

mode settings and start of conversion								ACKS	STOP
X	X	0	m1	m0	pt1	pt0	1		P
data byte (resolution & measurement to be decided)									

Figure 1: Start measurement and data conversion sequence using I^2C

Reading data from the device can be achieved by

- addressing the device with its slave address and the write bit set to active ($R/\overline{W}$)=0
- sending the address of the first register to be read
- addressing the device again with its slave address and the read bit set to active

After the acknowledge bit has been sent by the slave, the content of the register is transferred. Please note that the register address is incremented automatically by T5403 after sending 'acknowledge by master', so that no additional register address has to be submitted for reading out the following registers. The data transfer can be stopped by sending 'not acknowledge by master'. Figure 2 depicts this sequence schematically.

Start	7-bit slave address of device							R/W	ACKS	8-bit register address (0xF5)								ACKS
S	1	1	1	0	1	1	1	0		1	1	1	1	0	1	0	1	
								WR=0		control byte								

Start	7-bit slave address of device							R/W	ACKS	data from register 0xF5								ACKM
Sr	1	1	1	0	1	1	1	1		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	
								RD=1		data byte								

➔

data from register 0xF6								NACKM	STOP
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	X	P
data byte									

Figure 2: Principle of reading data from T5403 with I^2C

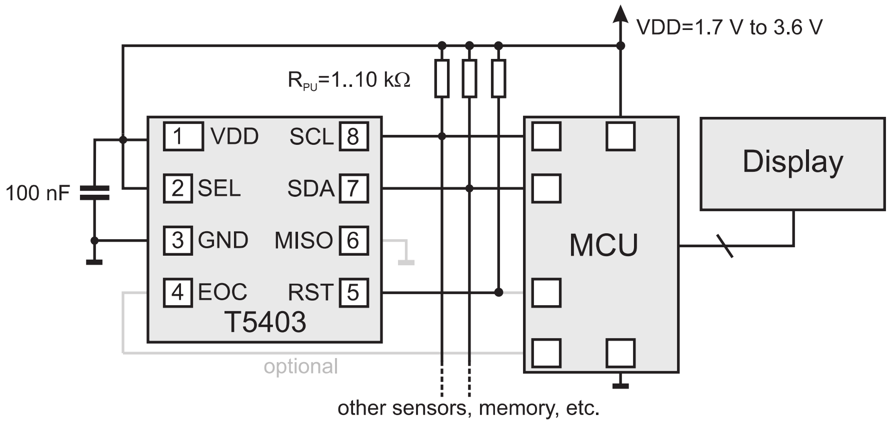


Figure 3: I2C application: example circuit

For proper functionality in I2C mode, the RST pin has to be on logical high level. This can be achieved by connecting RST either to VDD or to a specially assigned microcontroller port (marked as optional in Figure 3).

SPI mode

The synchronous serial peripheral interface (SPI) of the T5403 is defined in the following way:

- Clock polarity: high active
- Clock phase: rising clock edge triggers events on data lines
- Slave select polarity: low active

Figure 4 shows the nomenclature for timings related to the implemented SPI mode.

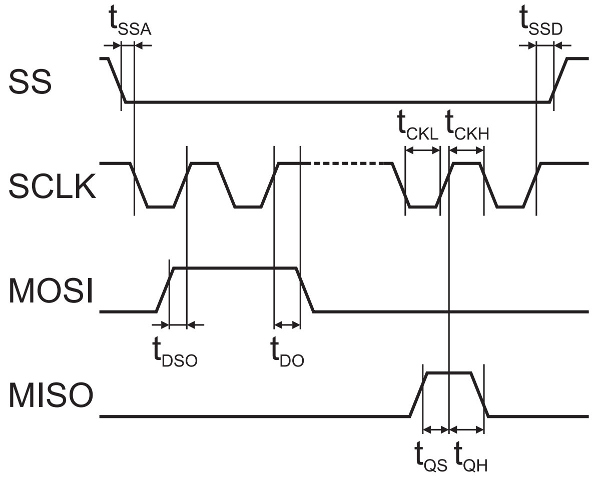


Figure 4: Definition of timings for SPI

The maximum and minimum timings for SPI mode can be taken from the table below.

		Min.	Typ.	Max.	Unit
SCLK clock frequency	f_{SCL}	—	—	20	MHz
SS activation time	t_{SSA}	21	—	—	ns
SCLK clock low time	t_{CKL}	21	—	—	ns
SCLK clock high time	t_{CKH}	21	—	—	ns
SS deactivation time	t_{SSD}	21	—	—	ns
MOSI setup time	t_{DSO}	6	—	—	ns
MISO setup time	t_{QS}	10	—	—	ns
MOSI hold time	t_{DO}	6	—	—	ns
MISO hold time	t_{QH}	0	—	—	ns
Maximum capacitive load	C_{MAX}	—	25	—	pF

Figure 5 provides the SPI sequence for starting a measurement and data conversion depending on mode settings, while Figure 6 shows the SPI sequence required for reading out temperature data. It can be observed that the register address is incremented automatically.

Start	R/W	*7-bit register address (0xF1), A7=1*							*mode settings and start of conversion*								Stop
SS active	0	1	1	1	0	0	0	1	X	X	0	m1	m0	pt1	pt0	1	SS inact.
	control byte								data byte (resolution & measurement to be decided)								

Figure 5: SPI: Start of measurement and data conversion

Start	R/W	*7-bit register address (0xF5), A7=1*							data from register 0xF5								
SS active	1	1	1	1	0	1	0	1	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	➔
	control byte								data byte								

	data from register 0xF6								*Stop*
➔	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	SS inact.
	data byte								

Figure 6: Readout of measurement data with SPI; automatic incrementation of register address

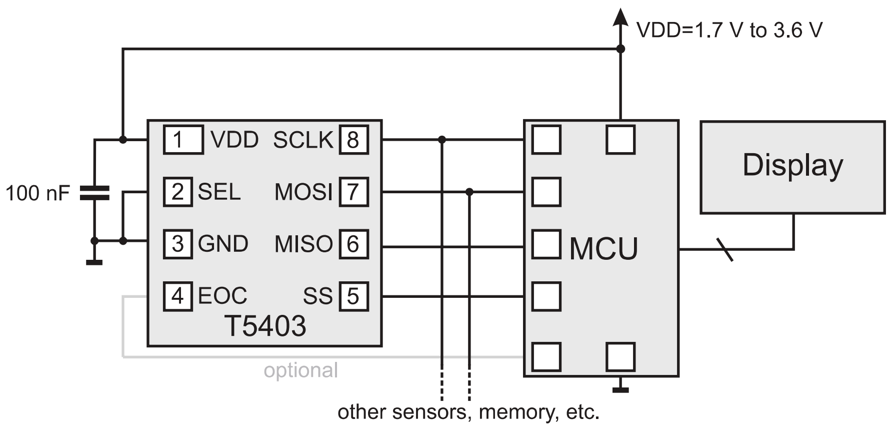


Figure 7: SPI application: example circuit

Time constraints

The following timings have to be taken into account after power-up and wake up of T5403.

		Min.	Typ.	Max.	Unit	Note or condition
Start-up time	t_S	—	—	10	ms	Delay between power on and first serial communication
Wake-up time	t_{WU}	—	2	2.5	ms	Wake up time from idle mode to start of first measurement

Time flow

Figure 8 schematically depicts the timings for interface transaction, wake up of T5403, measurement, interrupt and readout.

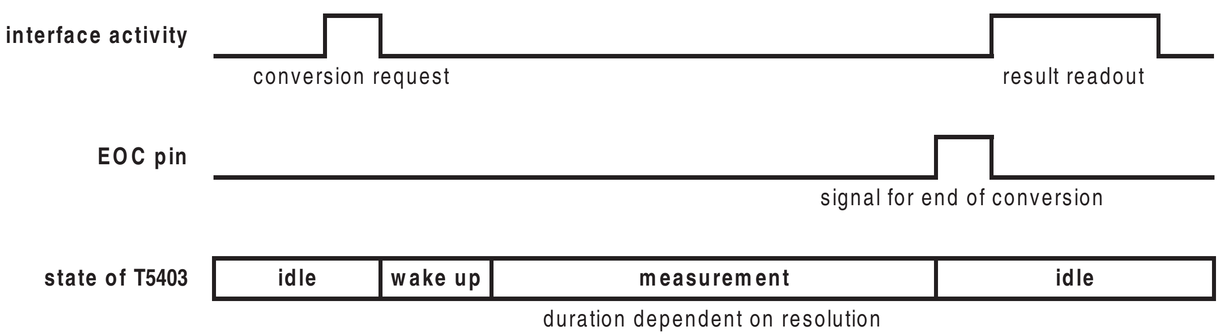


Figure 8: Time flow for start of conversion until readout of data

Application board

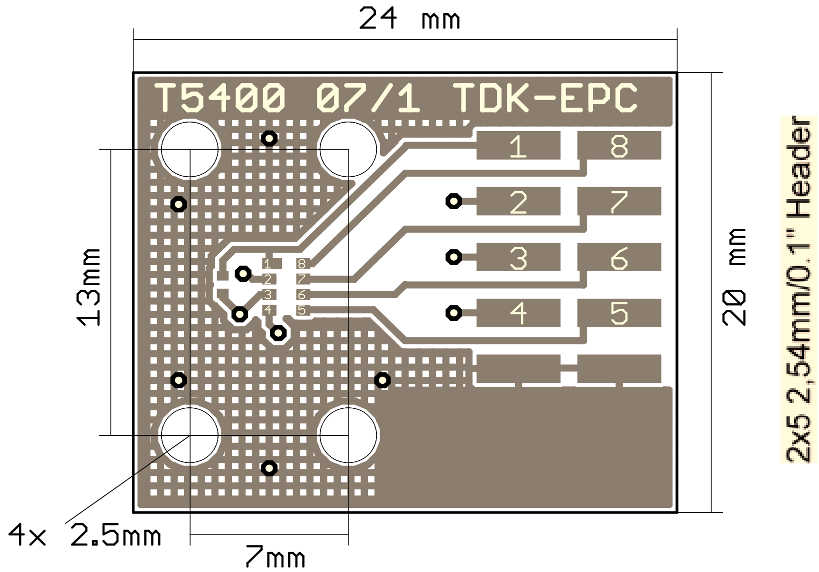


Figure 9: Pin assignment and mechanical dimensions of the application board, suitable for pressure sensor family T54xx

For further information please contact your local EPCOS sales office or visit our webpage at www.epcos.com .

Published by EPCOS AG
Systems, Acoustics, Waves Business Group
P.O. Box 80 17 09, 81617 Munich, GERMANY

For questions on technology, prices and delivery please contact the Sales Offices of EPCOS AG or the international Representatives.

Due to technical requirements components may contain dangerous substances. For information on the type in question please also contact one of our Sales Offices.

Important notes

The following applies to all products named in this publication:

1. Some parts of this publication contain **statements about the suitability of our products for certain areas of application**. These statements are based on our knowledge of typical requirements that are often placed on our products in the areas of application concerned. We nevertheless expressly point out **that such statements cannot be regarded as binding statements about the suitability of our products for a particular customer application.** As a rule, EPCOS is either unfamiliar with individual customer applications or less familiar with them than the customers themselves. For these reasons, it is always ultimately incumbent on the customer to check and decide whether an EPCOS product with the properties described in the product specification is suitable for use in a particular customer application.
2. We also point out that **in individual cases, a malfunction of electronic components or failure before the end of their usual service life cannot be completely ruled out in the current state of the art, even if they are operated as specified.** In customer applications requiring a very high level of operational safety and especially in customer applications in which the malfunction or failure of an electronic component could endanger human life or health (e.g. in accident prevention or life-saving systems), it must therefore be ensured by means of suitable design of the customer application or other action taken by the customer (e.g. installation of protective circuitry or redundancy) that no injury or damage is sustained by third parties in the event of malfunction or failure of an electronic component.
3. **The warnings, cautions and product-specific notes must be observed.**
4. In order to satisfy certain technical requirements, **some of the products described in this publication may contain substances subject to restrictions in certain jurisdictions (e.g. because they are classed as hazardous)**. Useful information on this will be found in our Material Data Sheets on the Internet (www.epcos.com/material). Should you have any more detailed questions, please contact our sales offices.
5. We constantly strive to improve our products. Consequently, **the products described in this publication may change from time to time**. The same is true of the corresponding product specifications. Please check therefore to what extent product descriptions and specifications contained in this publication are still applicable before or when you place an order.
   We also **reserve the right to discontinue production and delivery of products**. Consequently, we cannot guarantee that all products named in this publication will always be available. The aforementioned does not apply in the case of individual agreements deviating from the foregoing for customer-specific products.
6. Unless otherwise agreed in individual contracts, **all orders are subject to the current version of the “General Terms of Delivery for Products and Services in the Electrical Industry” published by the German Electrical and Electronics Industry Association (ZVEI)**.
7. The trade names EPCOS, BAOKE, Alu-X, CeraDiode, CeraLink, CeraPlas, CSMP, CSSP, CTVS, DeltaCap, DigiSiMic, DSSP, FilterCap, FormFit, MiniBlue, MiniCell, MKD, MKK, MLSC, MotorCap, PCC, PhaseCap, PhaseCube, PhaseMod, PhiCap, SIFERRIT, SIFI, SIKOREL, SilverCap, SIMDAD, SiMic, SIMID, SineFormer, SIOV, SIP5D, SIP5K, ThermoFuse, WindCap are **trademarks registered or pending** in Europe and in other countries. Further information will be found on the Internet at www.epcos.com/trademarks.