Digital I²C Bus Communication with HCLA Pressure Sensors

1   INTRODUCTION

Sensortechnics' digital HCLA series offers precision gage or differential pressure measurement in miniature package styles. These devices are digitally calibrated and temperature compensated with an on-board ASIC, which provides a corrected pressure value with up to 15-bit resolution. The response time depends on the adjusted internal resolution. For standard parts with 12 bit resolution it is typ. 500 µs. HCLA pressure sensors comply with the I²C protocol and can be easily connected to most commonly used microcontrollers and microprocessors.

The I²C bus is a simple, serial 8-bit oriented computer bus for efficient Inter-IC (acronym I²C) control. It provides good support for communication between different IC's across short circuit-board distances, such as interfacing microcontrollers with various low-speed peripheral devices.

Each device connected to the bus is software addressable by a unique address and a simple master-slave relationship exists at all times. The I²C bus requires only two bi-directional bus lines, a serial data line (SDA) and a serial clock line (SCL). Data can be transferred at rates of up to 100 kbit/s in standard mode or up to 400 kbit/s in fast mode.

2   I²C BUS PROTOCOL

2.1   General characteristics

Two lines, a serial data line (SDA) and a serial clock line (SCL), carry information between the devices connected to the bus (see Fig. 1). Any device with I²C compatible protocol can be attached to the bus. Both SDA and SCL are bidirectional lines, connected to a positive supply voltage via pull-up resistors (see section 4 Application Circuit). When the bus is free, both lines are HIGH. They can only be pulled LOW by the devices connected to the bus.
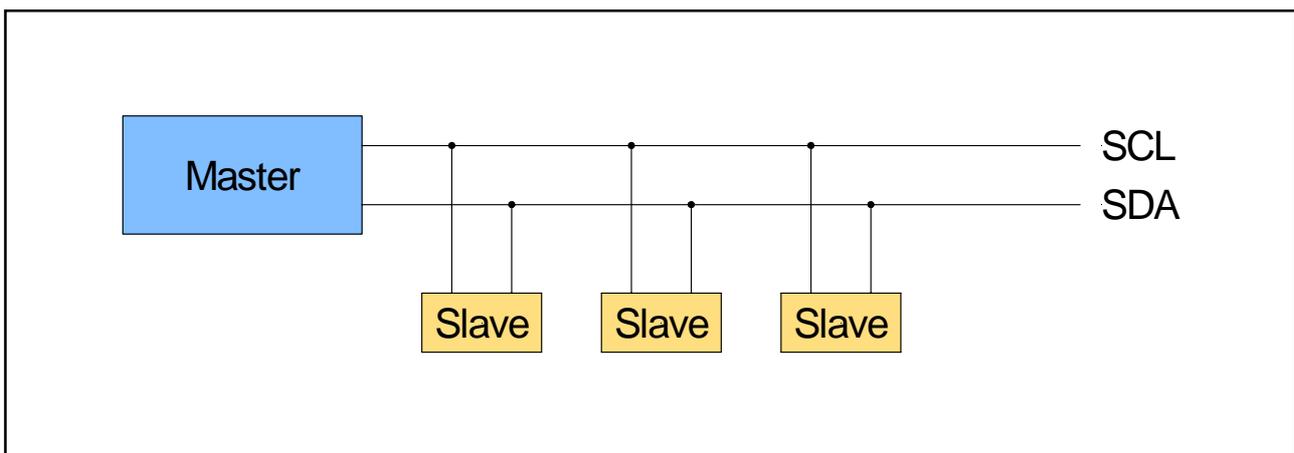

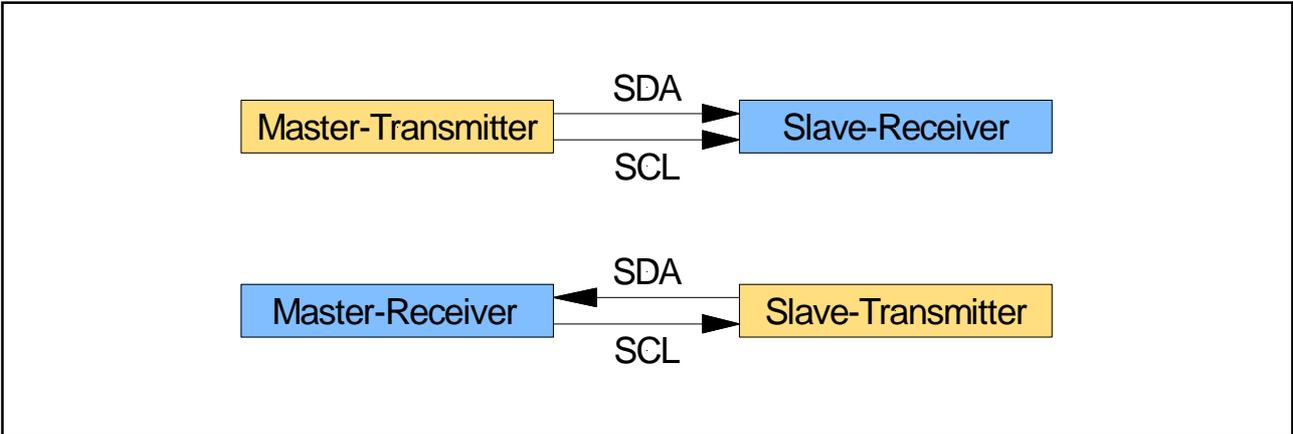**Fig. 1**   Example of an I²C bus configuration

SENSORTECHNICS   **www.sensortechnics.com**

Digital I²C Bus Communication with HCLA Pressure Sensors

Fig. 2   I²C master-slave concept

2.2 Master-slave concept

The I²C bus uses a simple master-slave concept. The master initialises a data transfer (START command), generates the clock (SCL) signal and terminates the transfer (STOP command). Any device addressed by the master is considered a slave.

Masters and slaves can operate as either transmitters or receivers, depending on their function (see Fig. 2). The transmitter is the device that sends data to the bus. A transmitter can either put data on the bus on its own accord (master-transmitter), or in response to a request from the master (slave-transmitter). The receiver is the device that receives data from the bus. A receiver can either receive data on its own request (master-receiver) or in response to a request from the master (slave-receiver).

> *__NOTE:__*
> *__The HCLA pressure sensors are designed to work as slave-transmitters and will therefore only respond to requests from a master device.__*

2.3 Data validity

Data is only valid during the HIGH period of the clock. During this HIGH period the data on the SDA line must be stable HIGH to transmit a "1" or stable LOW for a "0". The HIGH or LOW state of the data line is only allowed to change when the clock signal on the SCL line is LOW (see Fig. 3).
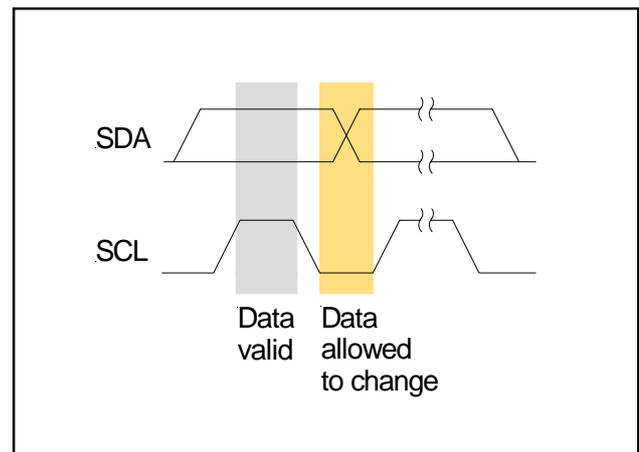

Fig. 3   I²C bus data validity

2.4 START (S) and STOP (P) conditions

Prior to any transaction on the bus, a START condition needs to be issued. The START condition acts as a signal to all connected IC's that something is about to be transmitted on the bus. As a result, all connected devices will listen to the bus and the bus is considered to be busy. After a message has been completed, a STOP condition is sent by the master. In this way the master can notify that no further data will be received or sent. This is the signal for all devices that the bus is idle again. The bus stays busy if a repeated START is generated instead of a STOP condition. START and repeated START conditions are functionally identical. START and STOP conditions are always generated by the master.

The START condition is defined as a HIGH to LOW transition on the SDA line while SCL is HIGH. The STOP condition is defined as a LOW to HIGH transition on the SDA line while SCL is HIGH (see Fig. 4).
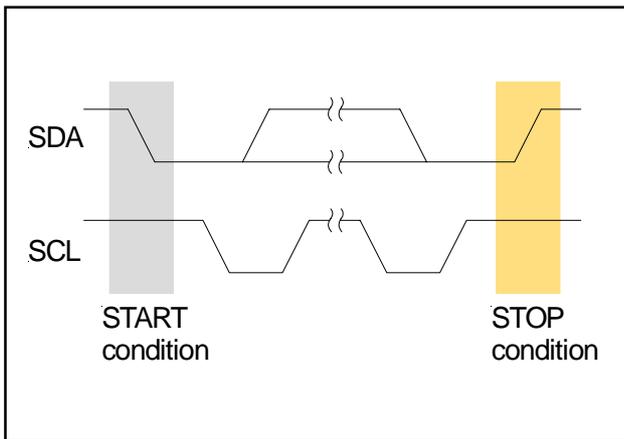
Fig. 4   I²C bus START and STOP conditions

2.5 Byte format

Every byte put on the SDA line must be 8-bits long. Data is transferred with the most significant bit (MSB) first and the other bits following down to bit 8, the least significant bit (LSB). As you can see in Fig. 5, the bit representing decimal number 128 will be transmitted first (MSB).

	MSB							LSB
Dual	0	1	1	0	1	1	0	1
Decimal	128	64	32	16	8	4	2	1

Fig. 5   I²C bus byte format

2.6 Addressing and data direction

The first byte after the START condition determines which slave will be selected by the master. Each slave has a unique address to identify it on the bus. This address is 7 bits long followed by bit 8 which is the data direction bit (R/$\overline{W}$) (see Fig. 6). A '0' for the R/$\overline{W}$ bit indicates that the master wants to transmit information to a selected slave (WRITE), a '1' indicates that the master requests information from the slave (READ).

MSB							LSB
7 bit address							R/$\overline{W}$
b7	b6	b5	b4	b3	b2	b1	b0

Fig. 6   I²C bus address byte

Digital I²C Bus Communication with HCLA Pressure Sensors

When an address is sent, each device in the system compares the first seven bits after the START condition with its own address. If they match, the device considers itself addressed by the master as a slave-receiver or slave-transmitter, depending on the R/$\overline{\text{W}}$ bit.

The 7-bit addressing allows 128 (2^7) different addresses on the same bus. Since 16 addresses are reserved (in the Phillips specification), in practice 112 addresses remain for communication on the bus. However, the total number of devices connected to the bus is limited by the maximum bus capacitance of 400 pF which corresponds to about 20 to 30 devices.

A slave address can be made-up of a fixed and a programmable part. Typically, the four most significant bits are fixed and assigned to specific categories of devices. The three least significant bits are programmable through hardware address pins allowing up to eight (2^3) devices of identical categories on the same I²C bus.

> **_NOTE:_**
> **_All HCLA sensors have a general preconfigured slave address (0x1111000b). By factory programming it is possible to define an additional secondary slave address for each individual device. According to I²C specification 127 different addresses are available. The sensor will then listen to both slave addresses._**

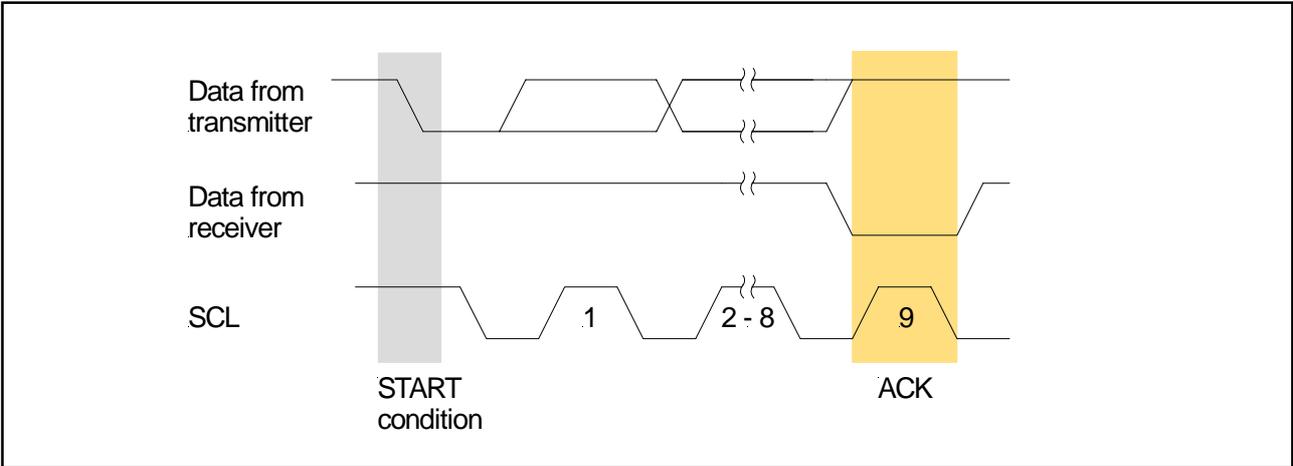## 2.7 Acknowledge (ACK or A)

Each byte sent on the bus has to be followed by an acknowledge bit issued from the receiver for correct receipt of data. The acknowledge also means that the device is ready to continue the data transfer.

The master must generate an extra clock pulse for this purpose. The transmitter releases the SDA line (HIGH) during the acknowledge clock pulse. The receiver must pull down the SDA line so that it is stable LOW from the beginning (rising edge) of the acknowledge clock pulse until its end (LOW level after falling edge) (see Fig. 7).

If the receiver does not want to receive a further byte it leaves the SDA line HIGH and a not acknowledge will be sent.

**SENSORTECHNICS**   **www.sensortechnics.com**

Fig. 7 Acknowledge (ACK) on the I²C bus

2.8 Data transfer overview

A complete data transfer is shown in Fig. 8. After the START condition, the slave address is sent. The data direction bit (R/$\overline{\text{W}}$) determines whether it is a READ or WRITE transfer. The addressed slave acknowledges (ACK) the correct receipt of the first byte. Now an unrestricted number of data bytes can be transmitted, always followed by an acknowledgement.

The data transfer is always terminated by a STOP condition generated by the master. However, if a master still wishes to communicate on the bus, it can generate a repeated START condition and address another slave without first generating a STOP condition. Various combinations of READ/WRITE formats are then possible within such a transfer.
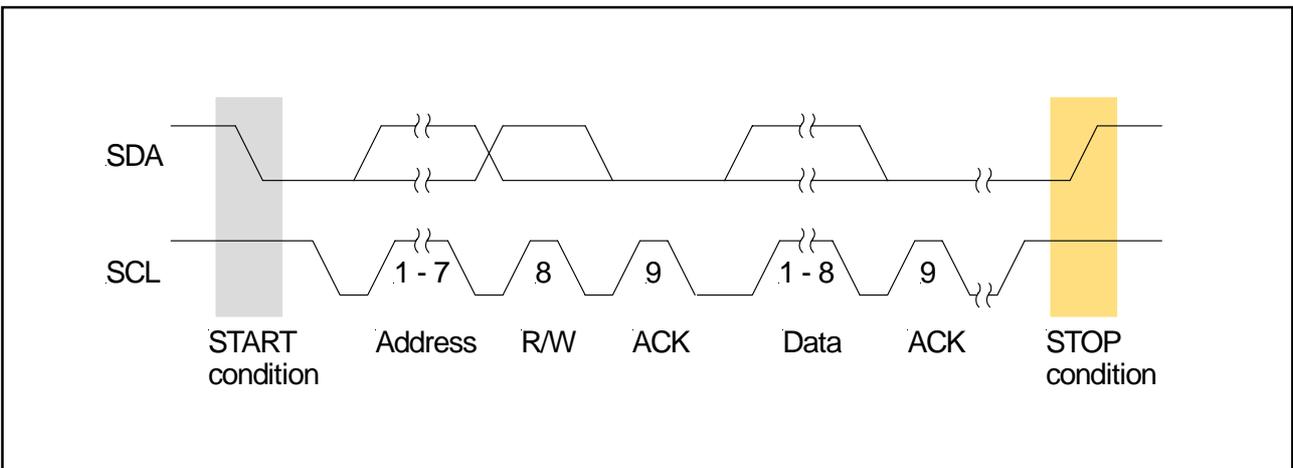

Fig. 8 Complete I²C bus data transfer

3 DATA TRANSFER WITH HCLA PRESSURE SENSORS

The HCLA pressure sensors are designed to work as slave-transmitters and will therefore only respond to requests from a master device.

3.1 Pressure reading

To read out a pressure value from a HCLA sensor the master generates a START condition and sends the HCLA slave address (either the hard coded addess 0x1111000b or the secondary factory programmed address) followed by a READ command (R/$\overline{W}$=1) (see Fig. 9). At the moment of the first acknowledge (A) from the HCLA-slave, the master-transmitter becomes a master-receiver. The HCLA sensor starts to send two data bytes containing the current pressure value as a 15 bit information placed in the output registers of its ASIC (see Fig. 10). The master must acknowledge the reception of the first data byte. To terminate the pressure reading from the sensor the master sends a STOP condition. The sensor is also able to send pressure values "online". That means if the master does not send a STOP condition but an acknowledge after the third exchanged byte the slave will go on sending the last available pressure value when it is clocked.
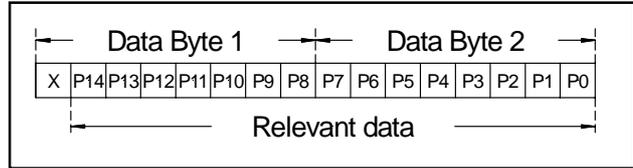

Fig. 10 Two data bytes containing the pressure value as a 15 bit information

> *NOTE:*
> *With a SCL clock frequency of 400 kHz the exchange of the 2 data bytes containing the current pressure value takes about 40 µs. However, the internal HCLA conversion cycle to obtain a new pressure value is 250 µs in standard configuration. Therefore, if the sensor is not deactivated it will send up to 7 times the same digital pressure value. For further information please call Sensortechnics.*

3.2 Optional temperature reading

As an option the sensor can be factory configured to deliver an additional 15 bit temperature reading. This will then be transmitted as a 3rd and 4th byte after the pressure value. If there is an acknowledge sent after these two additional bytes the HCLA sensor will continue sending alternating pressure and temperature values till it is deactivated by a STOP condition.
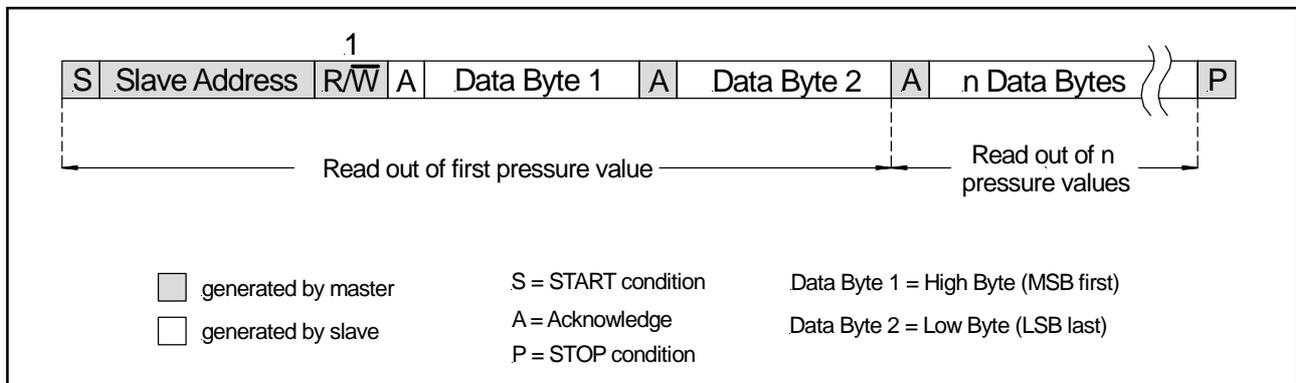
Fig. 9 Readout of pressure information from an HCLA sensor

SENSORTECHNICS   **www.sensortechnics.com**

3.3 Calculation of the actual pressure value from the digital pressure word

The following formulas show how to calculate the actual pressure value from the digital sensor output:

Definitions:

S	=	Sensitivity [counts/mbar]
Out_{max}	=	Output @ max. pressure [counts]
Out_{min}	=	Output @ min. pressure [counts]
P_{max}	=	Max. value of pressure range [mbar]
P_{min}	=	Min. value of pressure range [mbar]
P	=	Pressure reading [mbar]
P_{counts}	=	Digital pressure reading [counts]

$$S = \frac{Out_{max} - Out_{min}}{P_{max} - P_{min}} \qquad (1)$$

$$P = \frac{P_{counts} - Out_{min}}{S} + P_{min} \qquad (2)$$

The following example shows the calculation for a HCLA0050…U device (pressure range 0…50 mbar unidirectional). Please refer to the HCLA data sheet for the specified calibration values.

Out_{min} = 0666 counts hex = 1638 counts dec
Out_{max} = 6CCC counts hex = 27852 counts dec

With equation (1) the sensitivity of the sensor gives

$$S = \frac{27852\,counts - 1638\,counts}{50\,mbar - 0\,mbar} = 524{,}28 \text{ counts/mbar}$$

For an actual digital pressure reading of

P_{counts} = 0508 counts hex = 20608 counts dec

the actual pressure in mbar can be calculated from equation (2) to be

$$P = \frac{20608\,counts - 1638\,counts}{524{,}28\,counts\,/\,mbar} + 0\,mbar = \underline{\underline{36{,}18\,mbar}}$$

This pressure reading is calculated with the typical calibration values, not taking into account that the individual sensor calibrations might differ within the tolerances specified in the HCLA data sheet.

3.4 Resolution of data

Each temperature and pressure value will be transmitted as a 15 bit word. However, the actual resolution can be less than this depending upon how the internal A/D-converter is configured. Also internal calculations and signal windowing will reduce the effective resolution. The standard resolution for pressure measurement is set at 12 bits, which results in an effective resolution of 11 to 12 bits. For temperature measurement the limiting factor is the sensitivity of the sensing element. For further information please contact Sensortechnics.

SENSORTECHNICS

www.sensortechnics.com

3.5 Timing Parameters

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Bus free time between STOP and START condition	t_{BUF}		1.3			
Hold time (repeated) START condition	$t_{HD;STA}$	to first clock pulse	0.8			
LOW period of SCL	t_{LOW}		1.3			
HIGH period of SCL	t_{HIGH}		0.6			
Setup time repeated START condition	$t_{SU;STA}$		1			µs
Data hold time	$t_{HD;DAT}$		0			
Data setup time	$t_{SU;DAT}$		0.2			
Rise time of both SDA and SCL	t_{R}		---		0.3	
Fall time of both SDA and SCL	t_{F}		---		0.3	
Setup time for STOP condition	$t_{SU;STO}$		0.6			

Table 1 Timing parameters for I²C bus communication with HCLA pressure sensors
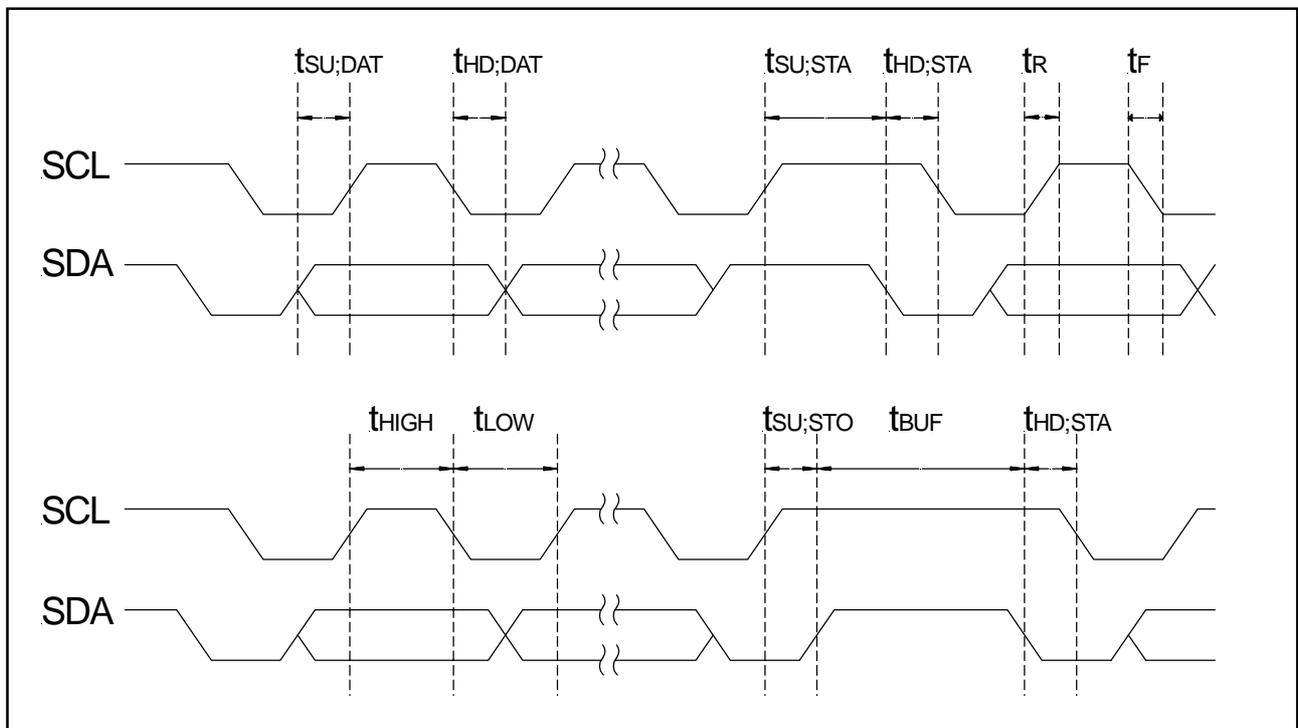
Fig. 11 Timing characteristics for I²C bus communication with HCLA pressure sensors

SENSORTECHNICS   **www.sensortechnics.com**

Parameter	Min.	Typ.	Max.	Unit
Input high level	90		100	
Input low level	0		10	% of Vs
Output low level			10	
Pull-up resistor	500			Ω
SCL clock frequency			400	kHz

Table 2 I²C bus communication parameters for HCLA pressure sensors

3.6 Communication Parameters

The maximum allowed communication speed on the bus depends on the configured internal clock frequency of the HCLA pressure sensors. It is 400 kHz in the standard configuration which corresponds to the fast mode transfer rate of the I²C bus specification.

3.7 Signal Noise due to Communication

As the pulses transmitted on the two bus lines have very sharp edges, this can cause some electromagnetic interference. Especially for very low pressure values and small PCB designs these spikes can influence the analog millivolt measurement of the sensor bridge and downgrade signal quality.

> **NOTE:**
> **Sensortechnics does not recommend digital readout for highly amplified ultra low pressure devices. Please contact Sensortechnics for more information.**

If both, digital and analog output signals of the HCLA sensors are used in parallel it is recommended to separate the signal lines as far as possible from each other. Further, decoupling capacitors of 220 nF between supply and ground and 15 nF between the analog output and ground are beneficial. It is important to place the capacitors as close to the pins as possible.

4 APPLICATION CIRCUIT

It is recommended to use a pull-up resistor of about 4k7 Ω and an additional 330 Ω serial resistor in each communication line as shown in Fig. 12.
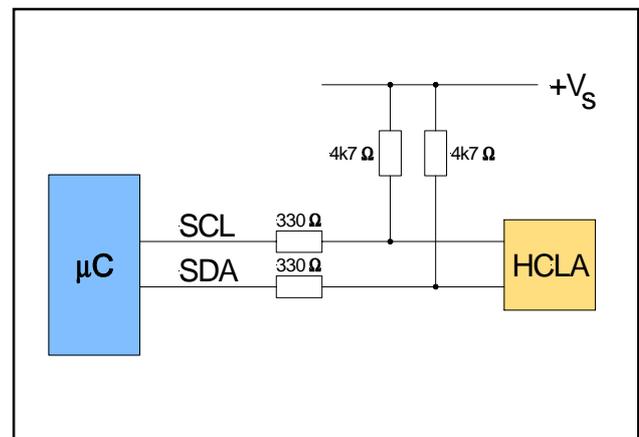
Fig. 12 I²C bus application circuit for HCLA sensors

SENSORTECHNICS

www.sensortechnics.com

5   SAMPLE PROGRAM CODE

```
//enable I2C interface
I2CON=0xC7;

// set I2C clock frequency
CRSEL=0;
I2SCLH=60;
I2SCLL=60;

//slave address and read
I2DAT=0xF1|0x1;

// send Start Bit
STA=1;

i2cdata=0;
run_I2C_loop=1;
i2c_count = 0;
while ((run_I2C_loop)&& i2c_count<10)
{
        while (!SI);

        i2c_count++;        // counter for limiting connection attempts
        switch(I2STAT)
        {
                case 0x10:
                case 0x08:          // START BIT has been sent
                                    // now send slave address with read flag
        I2DAT=0xF1|0x1; // SLAVE read
                                    STO=0;
                                    break;

                case 0x38:          // Arbitration lost -> Resend Start
        STA=1; STO=0;

                                    break;

                case 0x40:          // receive and send ACK
                                    STA=0;STO=0;AA=1;
                                    break;

                case 0x48:          // Sensor send NACK after address, send STOP followed by START bit
        STA=1; STO=1;

                                    break;

                case 0x50:          // Data has been received, ACK send
        i2cdata=I2DAT;              // Save data
                                    // we need only 2 byte, so send a NACK after next received byte
                                    STA=0; STO=0; AA=0;
                                    break;

                case 0x58:          // Data has been received, NACK send
        run_I2C_loop=0;                         // set exit flag
                                    i2cdata=(i2cdata<<8)+I2DAT;      // shift old data and add actual
                                    STA=0; STO=1;                    // send stop bit
                                    break;
                }
                SI=0;
        }
}

// now you have a resulting value in i2cdata, that represents the pressure
```