Table of contents

1. I²C protocol.................................... 2
1.1. General description.. 2
1.2. Communication phases... 3
1.3. I²C communication overview.................................... 4
2. Digital data transfer on I²C bus......... 4
2.1. Pressure & temperature data transfer on I²C bus..... 4
2.2. Calculation pressure formula.................................. 5
2.3. Calculation temperature formular............................ 6
2.4. I²C Timings parameters.. 6
3. Application scheme.......................... 7
4. Example program code 8

Huba Control AG
Industriestrasse 17
CH-5436 Würenlos

www.hubacontrol.com
info.ch@hubacontrol.com
Phone: +41 56 436 82 00

1. I^2C protocol

1.1. General description

In I^2C communication a serial data line (SDA) and a serial clock line (SCL) are required for the communication between the connected devices to I^2C bus. Both connected lines SDA and SCL are bidirectional, which are connected to the supply voltage with pull-up resistors (see application circuit on **Figure 7**).

As seen on **Figure 1** there can be more slave devices (up to 127) connected to I^2C bus, which is limited to 7 bit slave address. I^2C bus is free when both connection lines are HIGH and can be set to LOW by the devices which are connected to the I^2C bus.

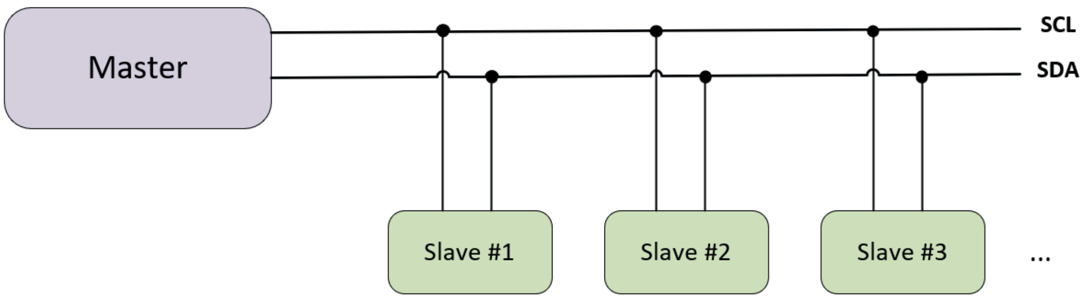

Figure 1. I^2C communication example

I^2C communication acts as a Master – Slave principle (see **Figure 2**), there is a master device which generates the clock (SCL) and also generates START & STOP command for data transition.

Figure 2. Master-Slave principle

Masters and slaves can act as transmitter or receiver depending on the information, if that needs to be sent or read. Transmitter is the device which sends data to the I^2C bus (“master transmitter” normally sends requests to the slave and the “slave transmitter” normally sends information replies to the master). The receiver is the device which receives data from I^2C bus.
I^2C standard protocol is showed in **Figure 3**.

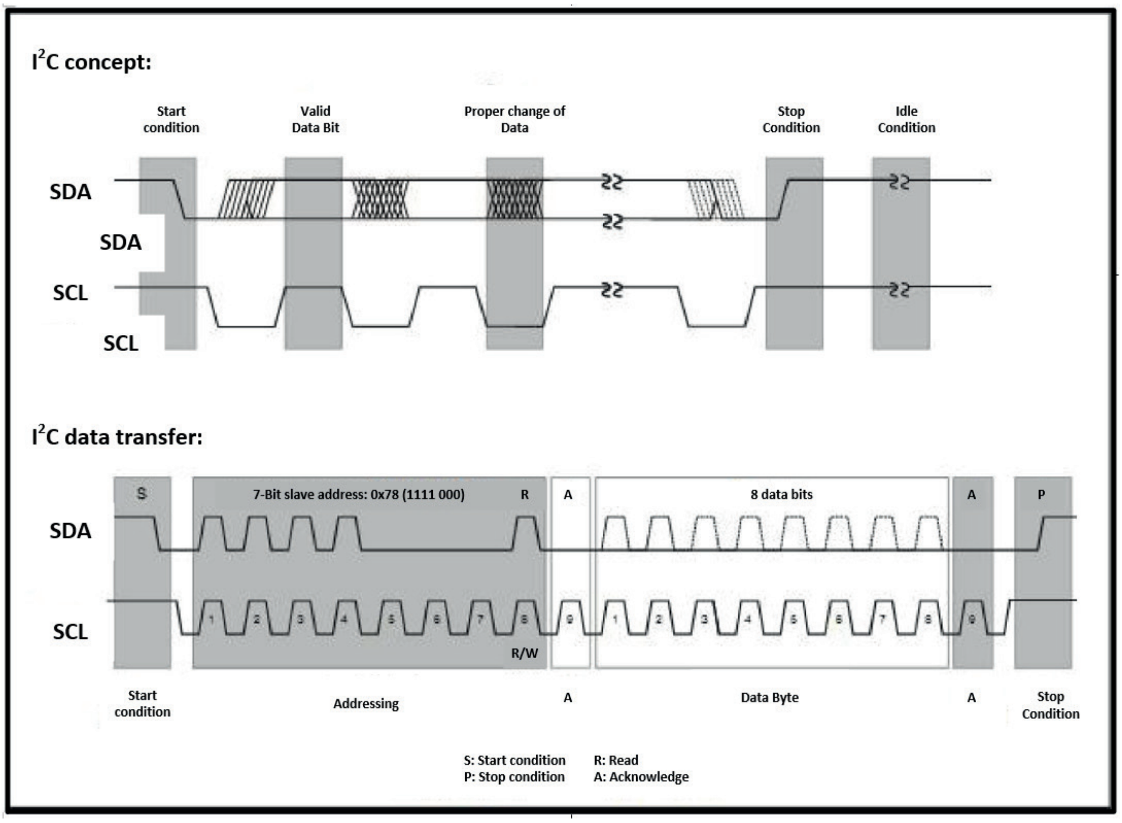

Figure 3. I^{2}C Standard communication protocol

1.2. Communication phases

- **Buss free – Idle state:** When the bus is free both lines SCL and SDA are pulled up – HIGH.
- **START condition (S):** Each data transfer starts with the start condition, which is always sent by the master. This start condition acts as a signal to all I^{2}C connected devices giving information that there will be something transmitted. The start condition is defined as a transition from HIGH to LOW on the SDA line when the SCL line is HIGH (see Figure 3).
- **STOP condition (P):** Each data transfer stops with the stop condition, which is also generated by the master when a data transfer has finished. Stop condition is defined as a transition from LOW to HIGH on the SDA line when the SCL line is HIGH (see Figure 3).
- **Valid data:** Data is always transmitted in bytes (8 bits) starting with the MSB (most significant bit). One data bit is transferred with each clock pulse. Transmitted data are valid (after generating start condition) only during HIGH period of clock and data changes can be done during LOW period of clock (see Figure 3).
- **Acknowledge (A):** Each sent byte needs to be followed with the acknowledge bit generated from the receiver that correct data has been received. Acknowledge means also that device can continue with further data transfer. For that purpose, the master must generate extra clock pulse. The transmitter releases clock HIGH during acknowledge clock pulse, if not then further bytes will not be sent.

- **Slave address:** After the start condition, the master sends the addressing byte - slave address to define with which slave device he wants to communicate. This addressing byte includes 7-bit slave ad dress (up to 128 devices) + 1 R/W bit (data direction bit). If R/W bit is set to "0" (W) then master wishes to transmit data to the selected slave. If R/W bit is set to "1" (R) then the master request data from the slave. The addressed slave answers with an acknowledge, all other slaves connec ted to the I²C-bus ignore this communication.

HUBA CONTROL Type 664 pressure sensor have a default slave address programmed to 0x78 (1111 000b). For connecting more slave devices to I²C bus each connected device should have its own slave address (up to 128 devices).

1.3. I²C communication overview

Figure 3 shows a complete data transfer. After generating the start condition, the master also sends the slave address with data direction bit (R/W), which gives read or write transfer. The addressed slave replies to this request always with the acknowledge (A) first. Now, unlimited numbers of data (bytes) can be transferred, which needs to be always confirmed with the acknowledge bit. This transfer can be stopped by the master by generating the stop condition. If master wishes to communicate also with another slave address, then can generate a second start condition without stopping the first one.

2. Digital data transfer on I²C bus

2.1. Pressure & temperature data transfer on I²C bus

Digital data transfer is presented in **Figure 4.**

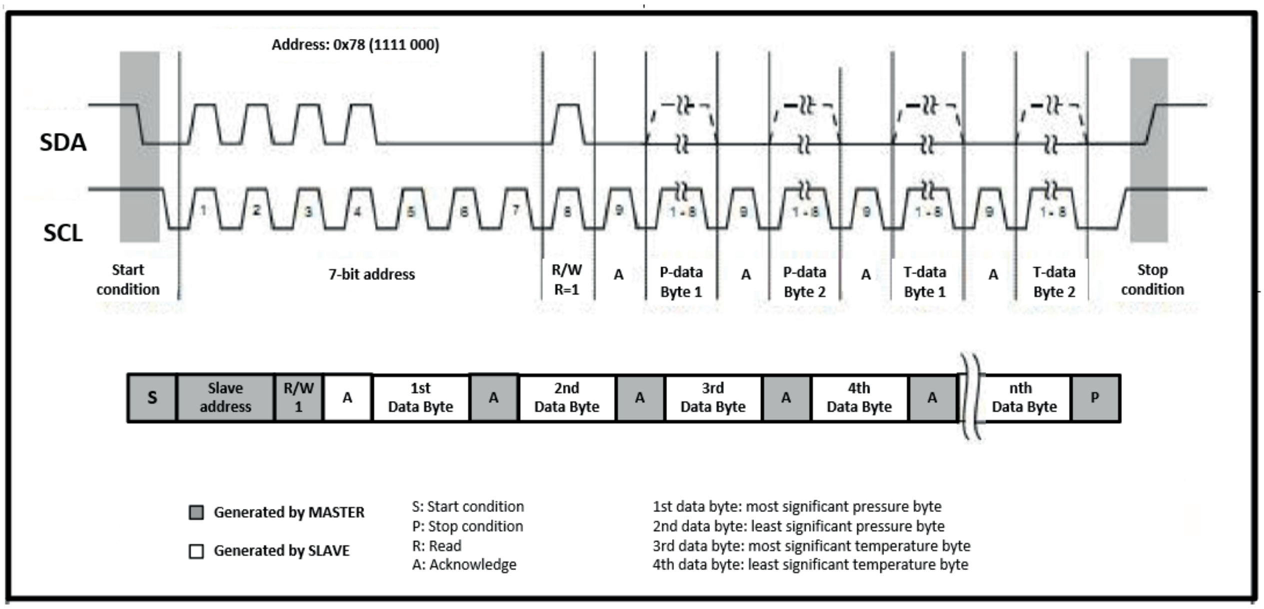

Figure 4. Digital pressure & temperature data transfer from HUBA Type 664 sensor

Pressure and temperature output signals from HUBA CONTROL Type 664 come as 15-bit values to the output register. Master, which would like to read this data, starts communication with the start condition. After that, the master sends 7-bit slave address (factory default is 0x78) and data direction bit R/W (for read data R/W="1"). The slave confirms this address with the acknowledge (A) bit first and afterwards sends the desired data with bytes (8 bits): first byte is most significant byte for pressure value, second byte is least significant for pressure value, third byte is most significant byte for temperature value, fourth byte if least significant byte for temperature value. The master must confirm each received byte with the acknowledge bit (see **Figure 4**). Master can stop the data transfer by sending the stop condition or can generate additional acknowledge bit after 4 receiving bytes of data (pressure and temperature) for continuing data receiving from slave (type 664 sensor).

2.2. Calculation pressure formula

Master receives pressure data as a 15-bit values which can be converted to actual pressure data with pressure units in mbar using the simple formula below.

Definitions:

- P = pressure (mbar)
- P_{min} = min pressure (mbar)
- P_{max} = max pressure (mbar)
- D = digital pressure (counts)
- D_{max} = max digital pressure (counts)
- D_{min} = min digital pressure (counts)
- S = sensitivity (count/mbar)

$$S = \frac{D_{max} - D_{min}}{P_{max} - P_{min}}$$

Equation 1.

$$P = \frac{D - D_{min}}{S} + P_{min}$$

Equation 2.

Example: for our pressure sensor with pressure range 0 to 500 mbar with analogue output 0.5 to 4.5 V (equivalent digital output 3277 to 29491 counts), a digital value of 7850 counts is measured. Let's calculate this value in pressure units mbar:

$$S = \frac{29491 - 3277}{500\ mbar - 0\ mbar} = 52{,}43\ counts/mbar$$

Equation 3.

$$S = \frac{7850 - 3277}{52{,}43} + 0\ mbar = 87{,}22\ mbar$$

Equation 4.

2.3. Calculation temperature formula

	Symbol	*Type*	*Unit*
Digital output (temperature), 15 bits			
Temperature output @ -20°C	T_0	8192	Counts
Temperature output @ 70°C	T_S	24576	Counts

The digital output signal for temperature is not ratiometric to power supply, the temperature data are read directly on the sensing element.

Temperature values from digital temperature values are calculated in the same manner as pressure formula calculation.

2.4. I²C Timings parameters

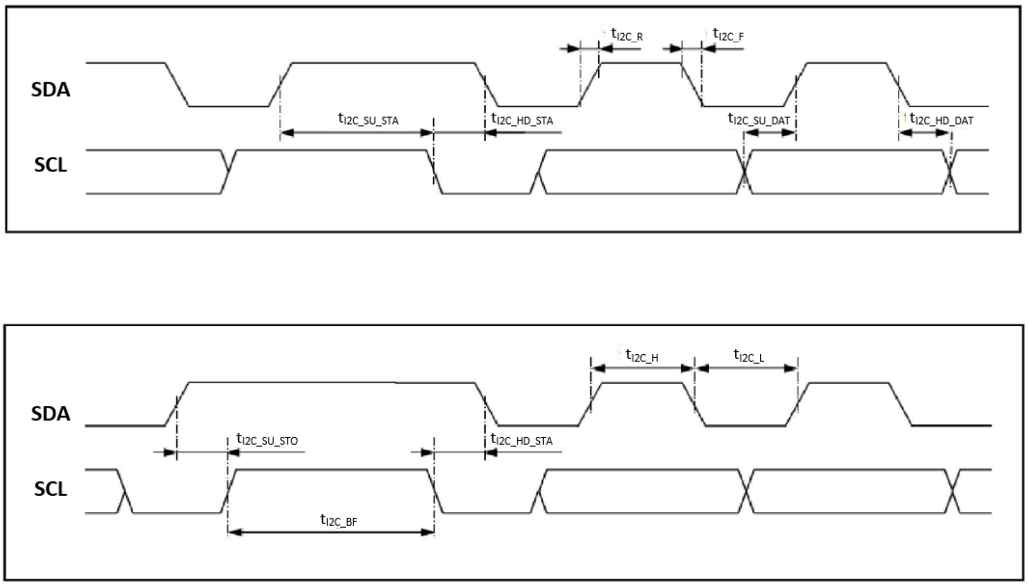

Figure 5. Timings parameters

Nr.	Parameter	Symbol	Min	Typ	Max	Unit	Conditions
1	SCL Clock frequency	f_{scl}			100	kHz	
2	Bus free time betw. start and stop condition	t_{I2C_BF}	1.3			µs	
3	Hold time start condition	$t_{I2C\_HD\_STA}$	0.6			µs	
4	Setup time repeated start condition	$t_{I2C\_SU\_STA}$	0.6			µs	
5	Low period SCL/SDA	t_{I2C_L}	1.3			µs	
6	High period SCL/SDA	t_{I2C_H}	0.6			µs	
7	Data hold time	$t_{I2C\_HD\_DAT}$	0			µs	
8	Data setup time	$t_{I2C\_SU\_DAT}$	0.1			µs	
9	Rise time SCL/SDA	t_{I2C_R}			0.3	µs	
10	Fall time stop condition	t_{I2C_F}			0.3	µs	
11	Setup time stop condition	$t_{I2C\_SU\_STO}$	0.6			µs	
12	Noise interception SDA	t_{I2C_NI}			50	ns	Spikes are suppressed

Figure 6. Timing I2C protocol

3. Application scheme

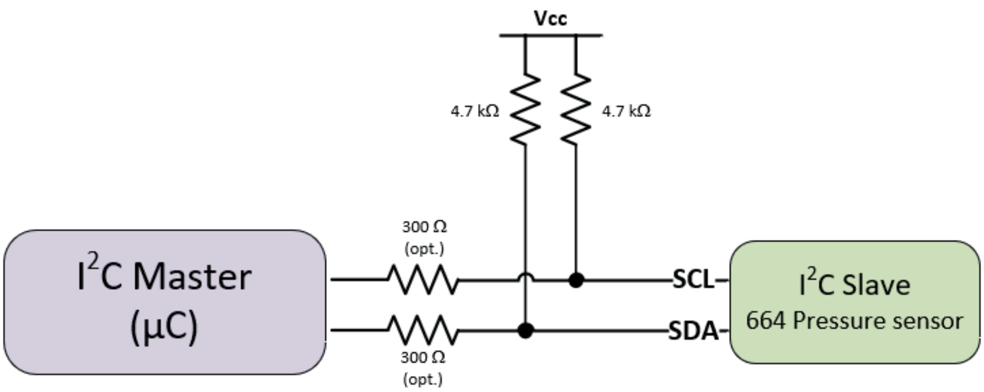

Figure 7. Application circuit

SCL and SDA lines needs to be connected to power supply via pull-up resistors as shown in **Figure 7.** HUBA CONTROL recommends using 4.7k ohms resistors as a pull-up resistors and 300 ohm resistors as serial resistors.

4. Example program code

A simple example code for pressure readings from HUBA CONTROL Type 664 pressure sensor is showed below:

```
byte msb, lsb; // 2 x 8bit values
int16 pressure; // 1 x 16bit value
// Set I2C unit to I2C master mode, clock speed 100 kHz and 7 bit addressing
configureI2C (I2C_MASTER | CLK_SPEED_100KHZ | ADDRESSING_7BIT);
// Set the target default slave address (0x78 = 120dec)
I2C_set_target(0x78);
// Send start condition (slave)
I2C_send_start_read();
// Read first data byte (msb) & answer with ACK (continue communication)
I2C_read (&msb, SEND_ACK);
// Read second data byte (lsb) and answer with NACK (end communication)
I2C_read (&lsb, SEND_NACK);
// Send stop condition
I2C_send_stop();
// Put both values together
pressure = ((int16)msb << 8) | lsb;
```