3. Interface Specifications

The serial interface of the SFM4100 series is optimized in terms of sensor readout and power consumption. It is compatible with I²C interfaces. For detailed specifications of the I²C protocol, see *The I2C Bus Specification*, Version 2.1, January 2000 (source: NXP). (www.nxp.com/products/interface_control/i2c)

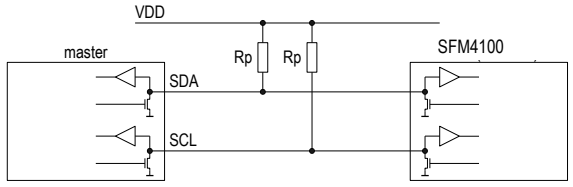
3.1 Interface connection – external components

Bi-directional bus lines are implemented by the devices (master and slave) using open-drain output stages and a pull-up resistor connected to the positive supply voltage.

The recommended pull-up resistor value depends on the system setup (capacitance of the circuit or cable and bus clock frequency). In most cases, 10 kΩ is a reasonable choice.

The capacitive loads on SDA and SCL line have to be the same. It is important to avoid asymmetric capacitive loads.

I²C Transmission Start Condition



Both bus lines, SDA and SCL, are bi-directional and therefore require an external pull-up resistor.

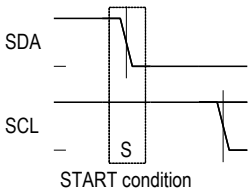
3.2 I²C Address

The I²C address consists of a 7-digit binary value. By default, the I²C address of the SFM4100 is gas specific (see chapter 2.2). The address is always followed by a write bit (0) or read bit (1).

3.3 Transfer sequences

Transmission START Condition (S): The START condition is a unique situation on the bus created by the master, indicating to the slaves the beginning of a transmission sequence (the bus is considered busy after a START).

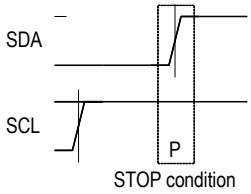
I²C Transmission Start Condition



A HIGH to LOW transition on the SDA line while SCL is HIGH

Transmission STOP Condition (P): The STOP condition is a unique situation on the bus created by the master, indicating to the slaves the end of a transmission sequence (the bus is considered free after a STOP).

I²C Transmission Stop Condition

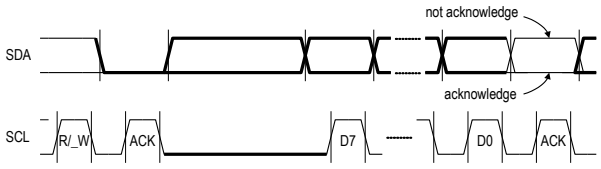


A LOW to HIGH transition on the SDA line while SCL is HIGH.

Acknowledge (ACK) / Not Acknowledge (NACK): Each byte (8 bits) transmitted over the I²C bus is followed by an acknowledge condition from the receiver. This means that after the master pulls SCL low to complete the transmission of the 8th bit, SDA will be pulled low by the receiver during the 9th bit time. If after transmission of the 8th bit the receiver does not pull the SDA line low, this is considered to be a NACK condition.

If an ACK is missing during a slave to master transmission, the slave aborts the transmission and goes into idle mode.

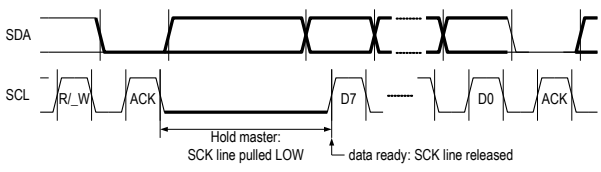
I²C Acknowledge / Not Acknowledge



Each byte is followed by an acknowledge or a not acknowledge, generated by the receiver

Handshake procedure (Hold Master): In a master-slave system, the master dictates when the slaves will receive or transmit data. However, in some situations a slave device may need time to store received data or prepare data to be transmitted. Therefore, a handshake procedure is required to allow the slave to indicate termination of internal processing.

I²C Hold Master



After the SCL pulse for the acknowledge signal, the SFM4100 series sensor (slave) can pull down the SCL line to force the master into a wait state. By releasing the SCL line, the sensor indicates that its internal processing is completed and transmission can resume. (The bold lines indicate that the sensor controls the SDA/SCL lines.)

3.4 Data transfer format

Data is transferred in byte packets in the I²C protocol, which means in 8-bit frames. Each byte is followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first.

A data transfer sequence is initiated by the master generating the Start condition (S) and sending a header byte. The I²C header consists of the 7-bit I²C device address and the data direction bit (R/_W).

The value of the R/_W bit in the header determines the data direction for the rest of the data transfer sequence. If R/_W = 0 (WRITE) the direction remains master-to-slave, while if R/_W = 1 (READ) the direction changes to slave-to-master after the header byte.

4. Command Set and Data Transfer Sequences

A command is represented by an 8-bit command code. The data direction may not change after the command byte, since the R/_W bit of the preceding I²C header has already determined the direction to be master-to-slave. In order to execute commands in Read mode using I²C, the following principle is used. On successful (acknowledged) receipt of a command byte, the sensor stores the command nibble internally. The Read mode of this command is then invoked by initiating an I²C data transfer sequence with R/_W = 1.

If a correctly addressed sensor recognizes a valid command and access to this command is granted, it responds by pulling down the SDA line during the subsequent SCL pulse for the acknowledge signal (ACK). Otherwise it leaves the SDA line unasserted (NACK).

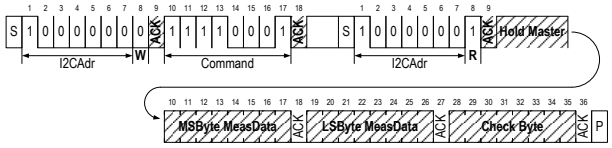
The two most important commands are described in this data sheet, and the data transfer sequences are specified. Contact Sensirion for advanced sensor options.

4.1 Measurement triggering

Each individual measurement is triggered by a separate read operation.

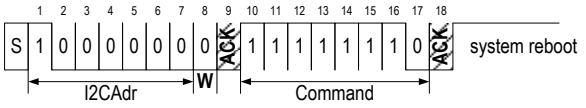
Note that two transfer sequences are needed to perform a measurement. First write command byte hF1 (trigger measurement) to the sensor, and then execute a read operation to trigger the measurement and retrieve the flow or differential pressure information.

On receipt of a header with R/_W=1, the sensor generates the Hold Master condition on the bus until the first measurement is completed (see Section 3.3 for timing). After the Hold Master condition is released, the master can read the result as two consecutive bytes. A CRC byte follows if the master continues clocking the SCL line after the second result byte. The sensor checks whether the master sends an acknowledge after each byte and aborts the transmission if it does not.

I²C Measurement
8-bit command code: ***hF1*** Command: **Trigger flow measurement**
 Hatched areas indicate that the sensor controls the SDA line.

4.2 Soft reset

This forces a sensor reset without switching the power off and on again. On receipt of this command, the sensor reinitializes the control/status register contents from the EEPROM and starts operating according to these settings.

I²C Soft Reset
8-bit command code: ***hFE*** Command: **Soft reset**


4.3 CRC-8 Redundant Data Transmission

Cyclic redundancy checking (CRC) is a popular technique used for error detection in data transmission. The transmitter appends an n-bit checksum to the actual data sequence. The checksum holds redundant information about the data sequence and allows the receiver to detect transmission errors. The computed checksum can be regarded as the remainder of a polynomial division, where the dividend is the binary polynomial defined by the data sequence and the divisor is a “generator polynomial”.

The SF04 sensor implements the CRC-8 standard based on the generator polynomial

$x^8 + x^5 + x^4 + 1$.

Note that CRC protection is only used for date transmitted from the slave to the master.

For details regarding cyclic redundancy checking, please refer to the relevant literature.