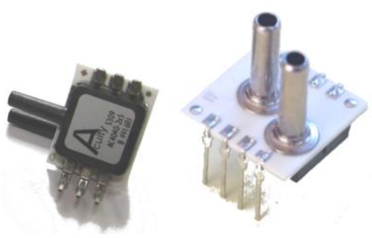
Acuity Incorporated
Fremont, California 94539
510-943-6205

Application Note – AC12-005
I2C Digital Communications for the AC4040 and AC4065 Series Amplified Pressure Sensors

This Application Note presents information on digital communications of Pressure and Temperature data between the AC4040 or AC4065 and a host, or master, computer. This document should only be used as a design guide.



Both the AC4040 and AC4065 series amplified pressure sensors achieve their accuracy thru digital calibration of the pressure sensor by an ASIC in the module. The ASIC is calibrated using an I2C communications protocol. A side benefit of this is that the same digital interface, used to calibrate the sensor, can also be used to read out the sensor's calibrated digital output. This allows simple communications with the sensor without the need to take the analog signal and process it thru an A-to-D converter at the host computer. The information below discusses the communications protocols and provides example programs for communicating with a module.

1. I2C protocol
1.1. General description

In I2C communication, a serial data line (SDA) and a serial clock line (SCL) communicate between all connected devices on an I2C bus. Both connected lines SDA and SCL are bidirectional lines; these are connected to the supply voltage with pull-up resistors (See application circuit on Figure 6). As shown in Figure 1, there is a Master Device (the controller computer) and multiple slaves (Pressure Sensors) on a bus. In this case, each slave must have a unique address. Theoretically, up to 127 separate sensors with separate addresses can be on the bus at once. This is limited with the number of 7 bit slave address. The I2C bus is free when both connection lines are HIGH, and can be put LOW by devices connected to the I2C bus. Note that for simplicity, all Acuity Sensors come with a default factory-set address of 0x78 (1111000). Thus, while possible to address multiple sensors on a single bus, as the standard module is provided from the factory, only one module can be on the bus. For customers needing to use multiple sensors on a single bus, please consult Acuity for special ordering details.

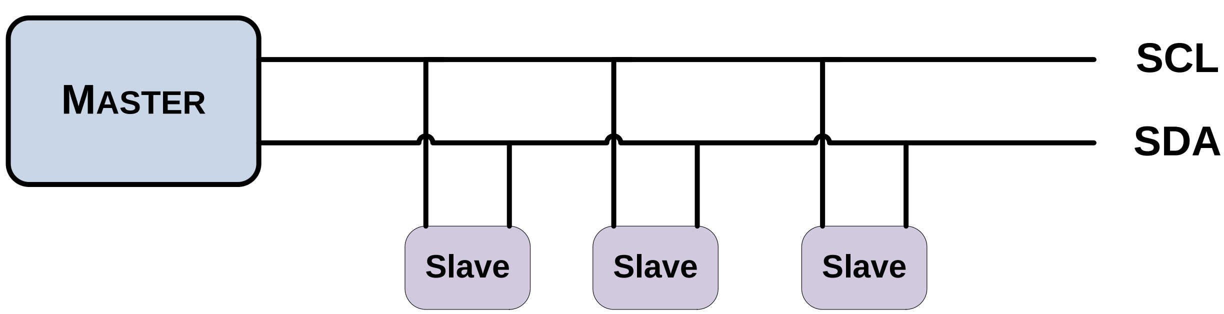


Figure 1 - I2C communication example

The I2C communications behave on the Master – Slave principle (See Figure 2), where the Master leads the communication and asks for certain information from the Slave. The Master device generates the clock (SCL) and generates START & STOP command for data transition.

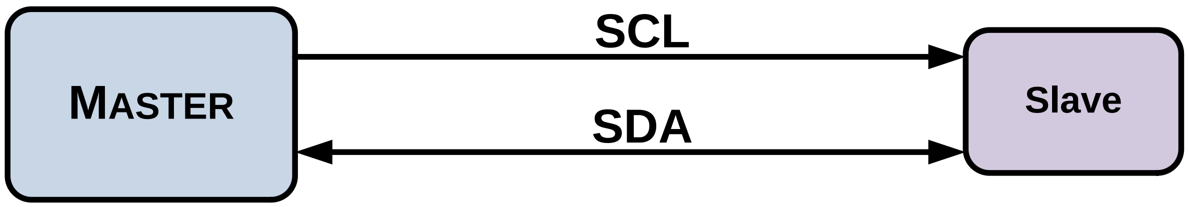
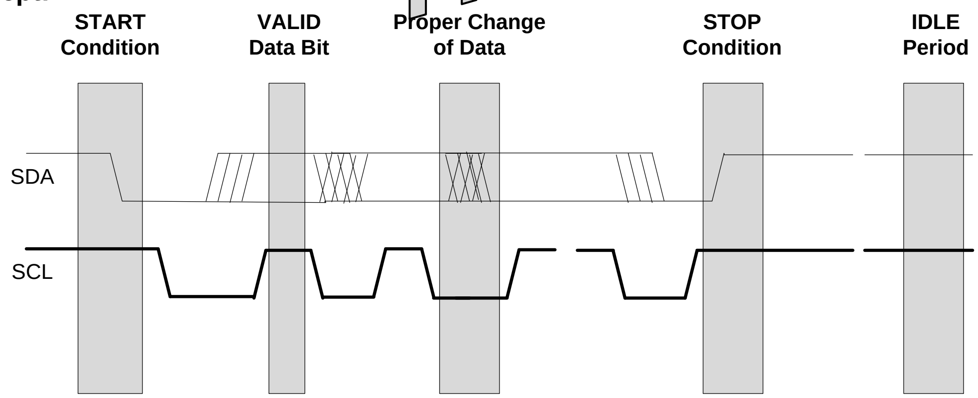


Figure 2 - Master – Slave principle

Masters and Slaves can act as transmitters or as receivers depending of the information which needs to be sent or read. A transmitter is the device which sends data to the I2C bus. The Master transmitter normally sends requests to the slave, while the Slave transmitter normally replies to the Master by sending requested information. The Receiver is device which receives data from the I2C bus. I2C standard protocol is presented in Figure 3.

I2C Concept:



I2C data Transfer:

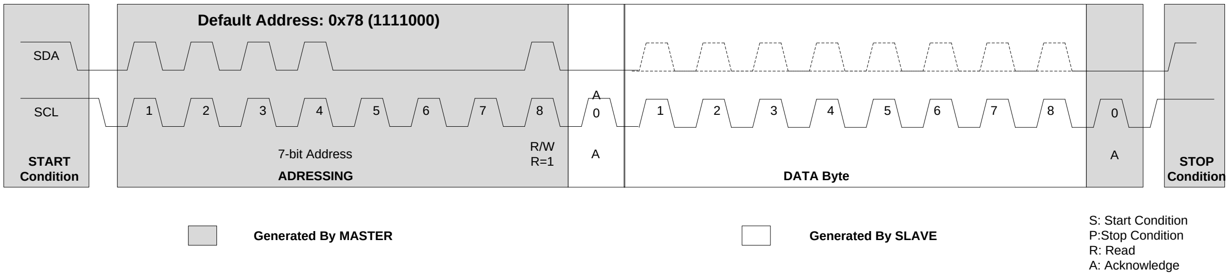


Figure 3 - I2C standard communication protocol

1.2. Communication phases

Bus 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. This is always sent by the Master. This start condition acts as a signal to all I2C connected devices giving information that there will be something transmitted. The Start condition is defined as a transition of from HIGH to LOW on SDA line when the SCL line is HIGH (See Figure 3).

STOP condition (P): Data transfer stops when the stop condition occurs; this is also generated by a Master when a data transfer has finished. The Stop condition is defined as transition from LOW to HIGH on SDA line when 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 on the SCL line. The transmitted data is valid (after generating start condition) only during HIGH period of the clock (on the SCL line) and data changes can occur 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, indicating that correct data has been received. Acknowledge means also that the device can continue with further data transfer. For that to occur, the Master must generate an extra clock pulse. The transmitter then releases SDA HIGH during the acknowledge clock pulse. If no Acknowledgement, then no further byte will be sent.

Slave address: After the start condition, the Master sends a addressing byte, the slave address in order to define with which slave device he wants to communicate with. This addressing byte includes 7 bit slave address (up to 128 devices) + 1 R/W bit (data direction bit). If the R/W bit is set to “0” (W), then the Master wishes to transmit data to selected Slave. If R/W bit is set to “1” (R), then the Master is requesting data from the Slave. The addressed Slave answers with an acknowledge; all other Slaves connected with the I2C-bus ignore this communication. Acuity pressure sensors have default programmed slave address to 0x78 (1111 000b). For connecting more slave devices to I2C bus, each connected device must have its own slave address (up to 127 devices). Consult Acuity for custom addresses for multiple sensors on a single bus.

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 information about read or write transfer. Addressed slave replies to this always with acknowledge (A) bit first. Now the unlimited numbers of data (bytes) can be transferred (every transferred byte needs to be confirmed with acknowledge bit). This transfer can be stopped by the master with generating the stop condition. If master wishes to communicate also with other slave address, it can generate also 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

The digital data transfer is presented in Figure 4.

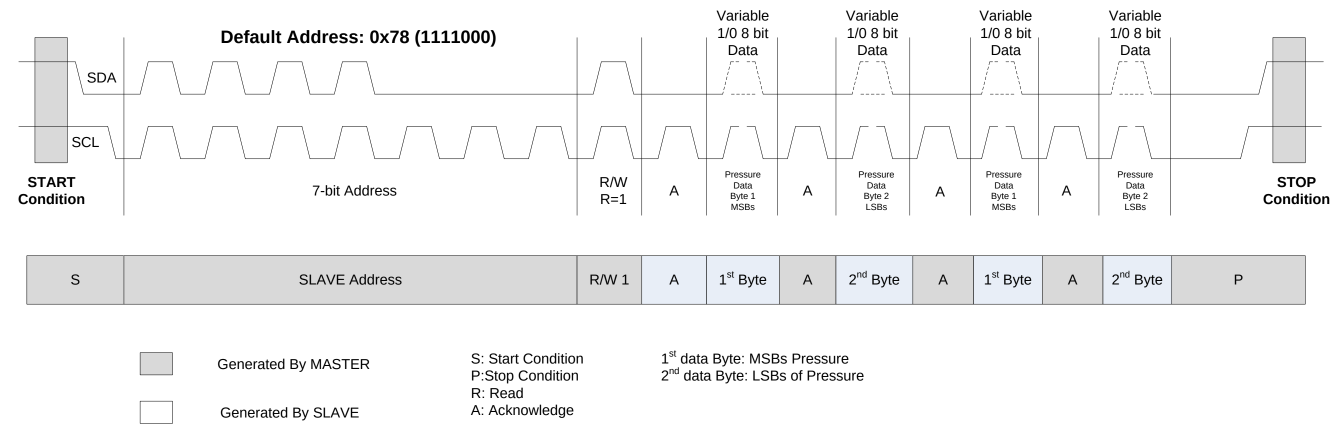


Figure 4a - Digital pressure data transfer from Acuity AC4040 and AC4065 sensors.

The standard Acuity configuration is to transmit Pressure only. Byte 1 and Byte 2 will continue to be transmitted alternately as long as the Master continues to send an acknowledge bit after each byte is sent from the Sensor.

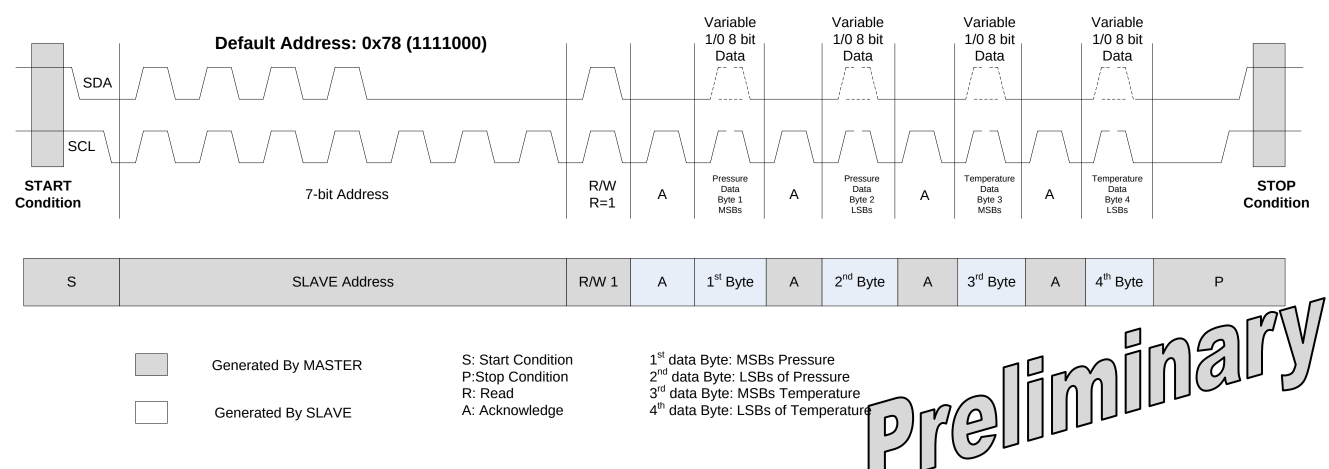


Figure 4b - Digital pressure & temperature data transfer from Acuity AC4040 and AC4065 sensors if configured for both.
The standard configuration from the factory is to transmit Pressure only.

In the standard, factory programmed configuration, only Pressure output signals from Acuity pressure sensors come as 15 bit values to the output register. The Master, which would like to read this data, starts communication with the start condition. After that Master sends 7 bit slave address (factory default is 0x78 or 1111000 binary) and data direction bit R/W (for read data R/W="1"), the Slave confirms this address with acknowledge (A) bit first and afterwards sends desired data with bytes (8 bits): first byte is most significant byte for pressure value, second byte is least significant by for pressure value.

The Master must confirm each received byte with acknowledge bit (See Figure 4). The Master can stop the data transfer by sending the stop condition or it can generate additional acknowledge bit after each receiving bytes of data for continues data receiving from the Slave (the Acuity Pressure sensor). If the Master does not Acknowledge the last byte's receipt, the Slave will not send additional bytes until the Acknowledgement or until the sequence is restarted.

2.1. Calculation pressure formula

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

Definitions:

- **P**= pressure (mbar)
- **Pmin**= min pressure (mbar)
- **Pmax**= max pressure (mbar)
- **D** = digital pressure (counts)
- **Dmax** = max digital pressure (counts)
- **Dmin** = min digital pressure (counts)
- **S**= sensitivity (count/mbar)

$$S = \frac{Dmax - Dmin}{Pmax - Pmin} \qquad P = \frac{D - Dmin}{S} + Pmin$$

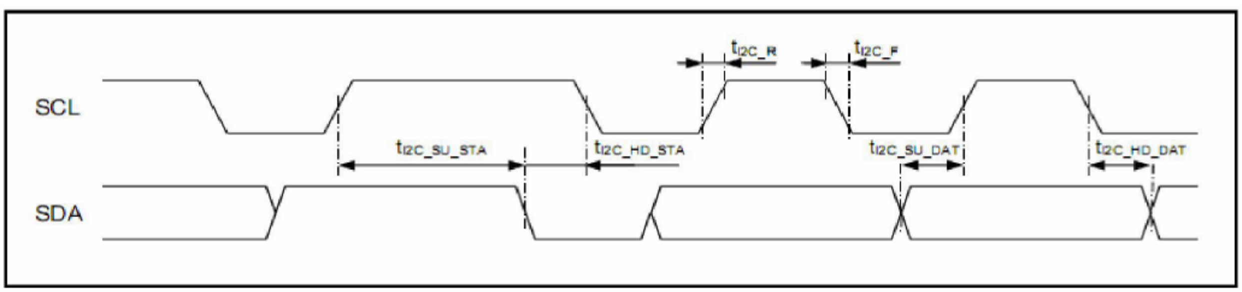
For Example: For our AC4065-010 pressure sensor with pressure range 0 to 10 mbar with analog output 0.5 to 4.5 V (equivalent digital output 3277 to 29491 counts), we measure a digital value of 7850 counts. To calculate this value in pressure with pressure units in mbar, we take the two equations above and substitute the data as required.

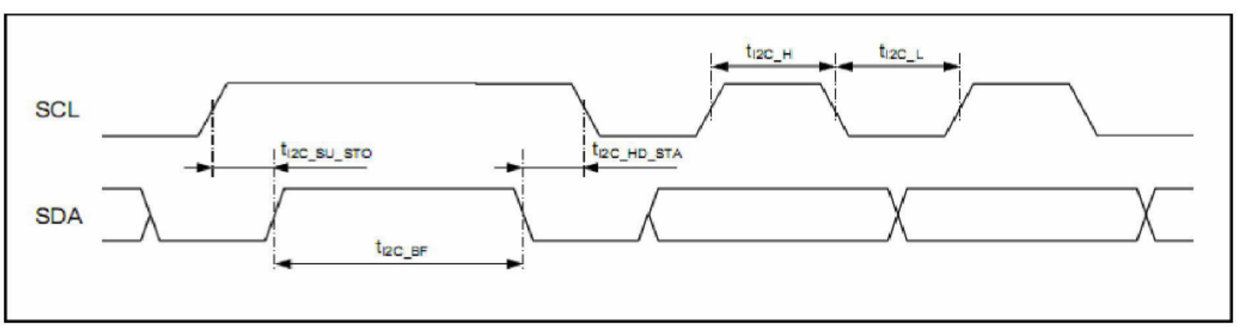
$$S = \frac{29491 - 3277}{10\ mBar - 0\ mBar} = 2621.4 \text{ counts/mBar}$$

$$P = \frac{7850 - 3277}{2621.4} + 0 \text{ mBar} = 1.744488 \text{ mBar}$$

Temperature values from digital temperature values are calculated in the same manner if the sensor is specifically ordered from the factory for both Pressure and Temperature.

2.2. I2C Timings parameters





Parameter	Symbol	Min	Max	Units	Comments
SCL Clock Frequency	f_{SCL}		400	kHz	f_{CLK}>2MHz
Bus free time between start and stop condition	t_{I2C_BF}	1.3		µS	
Hold Time Start Condition	$t_{I2C\_HD\_STA}$	0.6		µS	
Setup time repeated start condition	$t_{I2C\_SU\_STA}$	0.6		µS	
Low Period SCL/SDA	t_{I2C_L}	1.3		µS	
High Period SCL/SDA	t_{I2C_H}	0.6		µS	
Data Hold Time	$t_{I2C\_HD\_DAT}$	0		µS	
Data Setup Time	$t_{I2C\_SU\_DAT}$	0.1		µS	
Rise Time SCL/SDA	t_{I2C_R}		0.3	µS	
Fall Time SCL/SDA	t_{I2C_F}		0.3	µS	
Setup time stop condition	$t_{I2C\_SU\_STO}$	0.6		µS	
Noise Interception SDA	t_{I2C_NI}		50	nS	Spikes are suppressed

Figure 5 - Timing I2C protocol

3. Application Configuration

In the general configuration, the SCL and SDA lines needs to be connected to power supply via pull-up resistors as shown in Figure 6. Acuity recommends using 1.5k to 4.7k ohms resistors as a pull-up resistors and 200 to 300 ohm resistors as serial resistors. Note that only a single resistor network is needed. Multiple pull-up resistor pairs are NOT needed and NOT desirable. NOTE: Refer to your I2C Interface to your Master controller. Some modules have the pull-up resistors built in.

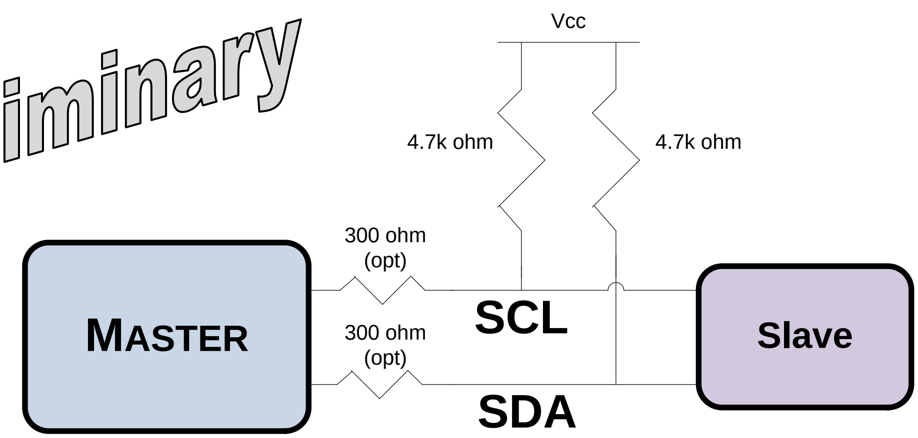


Figure 6 – Recommended wiring configuration from the Master to the Slave

4. Example Program Code

A simple example process flow code for pressure readings from Acuity AC4040 and AC4065 pressure sensor is presented 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;
```


For further information

Contact: **Acuity Incorporated**

Design@acuitymicro.com

+1-150-943-6205