I2C Communications Guide

Communicating with Piera Sensors on an I2C Bus

6/17/21 - Version 1.0a

Summary of I2C Mode

Piera's IPS Line of Particle Sensors default to UART/Serial mode of operating however they also support operating on an I2C (Inter-Integrated Circuit) Bus when correctly wired in certain deployment modes with a variety of I2C Masters and clock speeds This document will focus on communication with Piera IPS-Line of sensors specifically on I2C Bus

For customers that wish to evaluate Piera sensors for Particle Count capabilities and Air Quality Monitoring we recommend you evaluate the device using SenseiAQ Software on a Windows PC/Mac device using the USB Cable provided in the Piera Evaluation Kit. This operates the sensor in UART/Serial mode which has less dependencies on 3rd party hardware and other I2C bus devices operating correctly for basic sensor communication.

Basics: I2C Wiring for Piera sensors

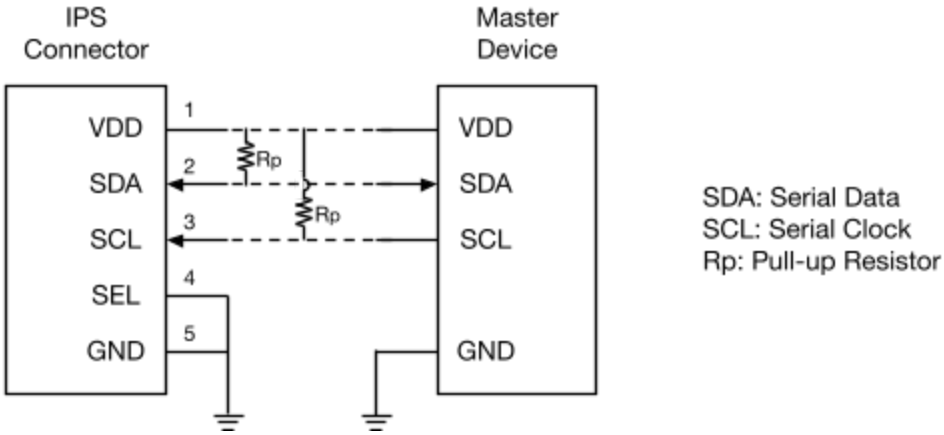
The Piera IPS-Series must be wired properly to operate in I2C Mode, the default mode is UART unless pins 4 and 5 are grounded together. A cable is included in the Piera Evaluation Kit or alternatively the standard JST Connector is built using these parts

Connector Plastic Housing:
https://www.digikey.com/en/products/detail/w%C3%BCrth-elektronik/648005113322/2508649?s=N4IgTCBcDaIGwBYAcAGFBWAjJgzDsEAugL5A
Pre-crimped wires:
https://www.digikey.com/en/products/detail/jst-sales-america-inc/ASZHSZH28K305/6009456?s=N4IgTCBcDaICwFYEFoDMAGAHO5A5AIiALoC%2BQA

Fig. 1. Interface connector I/O pins

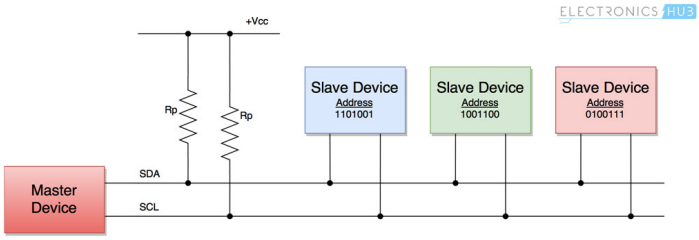
Pin	Name	Description	Comments
1	VDD	Supply voltage	5V ± 10%
2	Rx	UART: Receiving pin for communication	LVTTL 3.3V
	SDA	**I²C: Serial data input/output**	
3	Tx	UART: Transmitting pin for communication	LVTTL 3.3V
	SCL	**I²C: Serial clock input**	
4	SEL	Interface select	Floating: UART
			GND for I²C
5	**GND**	**Ground**	**GND for I²C**

Table 4. Interface connector I/O pin description

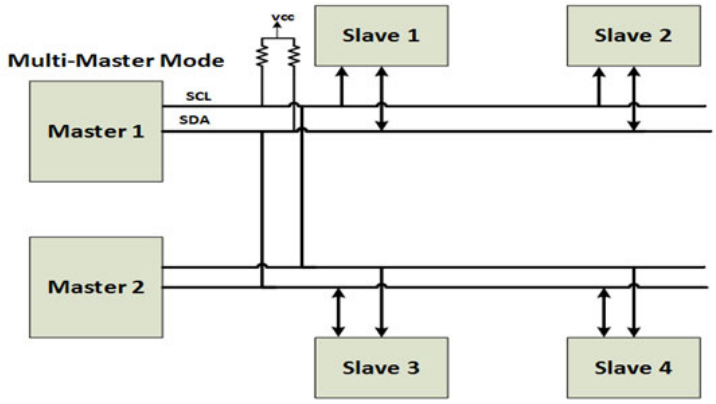


SDA is the serial data input/output pin and SCL is the serial clock pin between IPS and a master device. The Rp is pull-up resistor as the SDA and SCL are open drains, and IPS has an **internal / integrated 4.7kΩ resistors** on SDA and SCL paths. Because the I²C interfacing is generally used for communication between short distance devices less than 10cm, a particular attention must be paid to electromagnetic interference and crosstalk with well shielded connection cables/PCB patterns and total bus resistance. Piera IPS Series **I²C device ID is 0x4b** and cannot be changed

Wiring on I2C Bus with multiple devices



Synchronous transmission with Shared clock (single master)



Synchronous transmission with Shared clock (multi- master)

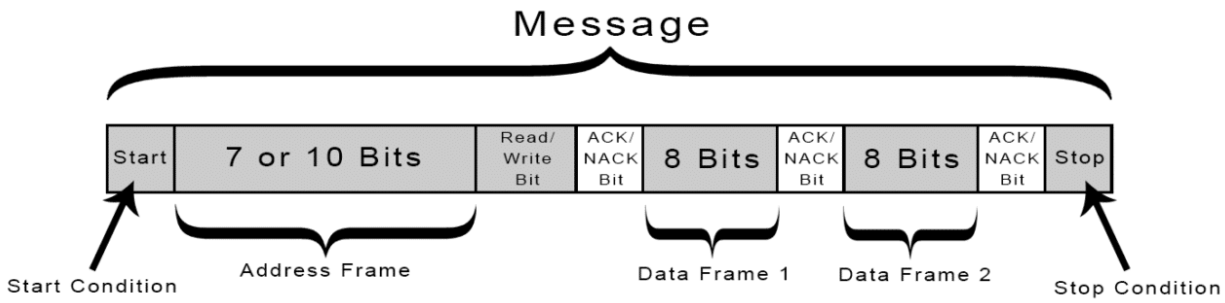
I2C Features Supported

- Slave mode only : Slow Mode Standard Mode, Fast Mode
- Minimum Read Size by Master: 28 bytes
  OEMs must provide a ***supported*** master controller for I2C bus.
- Integrated resistor **4.7kΩ inside Piera IPS Series**
- Default I2C Slave Address: **0x4b** **7 Bit Addressing Supported**
    - Control Model: Request-Response
- Voltage Supported: 5v, 3.3v only
- Max Capacitance on bus 400pf total
- Piera induced capacitance 4.7kΩ + Other Sensors = total resistance

Use-Cases and Applications

- Single Piera sensor on I2C Bus
- Multiple Piera sensors on a single I2C Bus
- 3rd party I2C Slave devices on the same I2C bus as other Piera sensors
- Multi-Master, Multi-Slave I2C bus

I2C Byte-Stream Command (with human readable conversions)



Update the tables below from datasheet for more information and details

1.1.1. Read Format for Piera IPS Series sensors

Command expecting data from the host

S	I²C address (7)	W	A	Command (8)	A

S	I²C address (7)	W	A	Data1 (8)	A	●●●	DataN (8)	N	P

Ex: 0x11, 0x12, 0x61, 0x62, 0x64, 0x65, 0x66, 0x6a, 0x77, 0x78, 0x79 Commands

Read commands are defined on the following table:

BYTES	Function	Parameters	Description
0x11	Read PC data (3 Series)	[n1],…,[n12] [n13][n14]	Read measured PC data for all 3 bins Each data has 4 unsigned bytes Number of data: [n1~n28] = 4byte x 3 CRC16: [n13]x256+[n14]
	Read PC data (5 Series)	[n1],…,[n20] [n21][n22]	Read measured PC data for all 5 bins Each data has 4 unsigned bytes Number of data: [n1~n28] = 4byte x 5 CRC16: [n21]x256+[n22]

	Read PC data (7 Series)	[n1],…,[n28] [n29][n30]	Read measured PC data for all 7 bins Each data has 4 unsigned bytes Number of data: [n1~n28] = 4byte x 7 CRC16: [n29]x256+[n30]
0x12	Read PM data (3 Series)	[n1],…,[n12] [n13][n14]	Read measured PM data for all 3 bins Each data has 4 bytes Float (IEEE-754) format The ordering method of the data conversion is Little Endian, i.e, if 4 byte data of ABCD is input, then DCBA is output with real number (floating point number). Number of data: [n1 – n28] = 4byte x 3 CRC16: [n13]x256+[n14]
	Read PM data (5 Series)	[n1],…,[n20] [n21][n22]	Read measured PM data for all 5 bins Each data has 4 bytes Float (IEEE-754) format The ordering method of the data conversion is Little Endian, i.e, if 4 byte data of ABCD is input, then DCBA is output with real number (floating point number). Number of data: [n1 – n28] = 4byte x 5 CRC16: [n21]x256+[n22]
	Read PM data (7 Series)	[n1],…,[n28] [n29][n30]	Read measured PM data for all 7 bins Each data has 4 bytes Float (IEEE-754) format The ordering method of the data conversion is Little Endian, i.e, if 4 byte data of ABCD is input, then DCBA is output with real number (floating point number). Number of data: [n1 – n28] = 4byte x 7 CRC16: [n29]x256+[n30]
0x61	Read Cleaning Interval	[n1],…,[n4] [n5][n6]	4 bytes are unsigned integer in second Default value = 604,800 [s] or 1 week CRC16: [n5]x256+[n6]
0x62	Read Mode Selection	[n1],…,[n3]	0: Auto Mode 1: Command Mode CRC16: [n2]x256+[n3]
0x64	Read Data Unit	[n1=0,1,2,3] [n2][n3]	For PC: 0 for #/L, 1 for #/ft^3, 2 for #/m^3 and 3 for #/L For PM: 0 for ug/m^3, 1 for ug/ft^3, 2 for ug/m^3 and 3 for ug/L CRC16: [n2]x256+[n3]
0x65	Read Start/Stop	[n1],…,[n4]	Measurement period reading, [n1]x256+[n2] in ms CRC16: [n3]x256+[n4]
0x66	Read Vth	[n1],…,[n4]	Detection range control voltage reading, [n1]x256+[n2] CRC16: [n3]x256+[n4]
0x69	Read Vref	[n1],…,[n4]	Sensitivity control voltage reading, [n1]x256+[n2] CRC16: [n3]x256+[n4]
0x6a	Read Status	n1],…,[n3]	Read status byte (b0: fan on/off, b1: cleaning on/off, b2: PSM on/off) Each bit is 0 for off and 1 for “On” b3=1 is for UART and 0 is for I^2C.

			CRC16: [n2]x256+[n3]
0x77	Read Serial Number	[n1],...,[n20]	[n1] – [n18] : Serial Number CRC16: [n19]x256+[n20]
0x78	Read version number	[n1],...,[n16]	[n1] – [n14] : Version number CRC16: [n15]x256+[n16]
0x79	Read Network Serial key	[n1],...,[n26]	[n1] – [n24] : Serial key CRC16: [n25]x256+[n26]

CRC16 error checking code for reference:

If you wish to implement error checking in the datastream the following C code is provided.

```
#define POLY 0x8408

uint16_t CRC16(uint8_t *byte, int len)
{
  int i, j;
  u16 data=0;
  u16 crc=0xffff;

  for(j=0;j<len;j++)
  {
     data = (u16)0xff & byte[j];
     for(i=0;i<8;i++, data >>= 1)
     {
       if((crc & 0x0001) ^ (data & 0x0001))
         crc = (crc >> 1) ^ POLY;
       else
         crc >>= 1;
     }
  }

  crc = ~crc;
  data = crc;
  crc = (crc << 8)|(data>>8 &0xff);

  return crc;
}
```

1.1.1. Write Format for Piera IPS Sensors

Command without Data

S	I²C address (7)	W	A	Command (8)	A	P

I²C address (7) = 0x4b

Ex: 0x2d, 0x2e Commands

Command with 1 parameter (1byte)

S	I²C address (7)	W	A	Command (8)	A	Parameter (8)	A	P

Ex: 0x10, 0x22, 0x23, 0x24, 0x2b, 0x2c Commands

Command with 2 parameters (2byte)

S	I²C address (7)	W	A	Command (8)	A	Parameter (8)	A	Parameter (8)	A	P

Ex: 0x26, 0x27, 0x28, 0x29 Commands

Command with 4 parameters (4byte)

S	I²C address (7)	W	A	Command (8)	A	Parameter (8)	A	Parameter (8)	A

Ex: 0x21 Commands

Parameter (8)	A	Parameter (8)	A	P

Write commands are defined on the following table:

CMD	Function	Parameters	Description
0x10	Start/Stop	3,2,1 or 0	Stop measurement (n=0) or start measurement with timer=n, n=1 for 200ms, 2 for 500ms and 3 for 1,000ms
0x21	Set Cleaning Interval	[n1],…,[n4]	[n1]<<24+[n2] <<16+[n3] <<8+[n2]&0xff
0x22	Operation Mode Selection	1 or 0	0: Auto Mode 1: Command Mode Sensor start to outputs data immediately upon powering up when Auto Mode is on, or sensor will wait for Set Interval command at Command Mode before starts
0x23	Power Saving Mode	1 or 0	0x01: Enter Power Saving Mode (<273uA), 0x0: Wake up pulse
0x24	Set Data Unit	[n=0,1,2,3]	For PC: 0 for #/L, 1 for #/ft^3, 2 for #/m^3 and 3 for #/L For PM: 0 for ug/m^3, 1 for ug/ft^3, 2 for ug/m^3 and 3 for ug/L
0x26	Set Vth	[n1], [n2]	Set detection range control voltage [n1] – MSB [n2] – LSB
0x29	Set Vref	[n1], [n2]	Set sensitivity control voltage [n1] – MSB [n2] – LSB
0x2b	Fan Operation	1 or 0	0x01: Start fan, 0x0: Stop fan
0x2c	Start Cleaning	1 or 0	0x01: Start cleaning, 0x0: Stop cleaning
0x2d	Reset	None	Resets the sensor module

			Allow approx. 5 seconds for normal communication
0x2e	Factory Reset	None	Restore all factory default settings

Use Case #1 Single Piera, Single master (ESP32)

The following example uses a common ESP32 and ESP-IDE Libraries to support reading data from a single Piera sensor on an I2C Enabled GPIO SCL, SDA.

The example code is available for download at https://github.com/PieraSystems/7100-I2C-example

Parts required

- Piera Systems IPS-7100
- ESP32 devkit
- 5 pin 1.50mm connector
- Crimped wires (these snap into connector)
- USB to micro USB cord (power/communication for ESP32)
- Snap crimped wires into 5 pin connector to make a plug for the 7100.
- Attach 7100 to ESP32 devkit accordingly:
    - VDD on 7100 to 5V on ESP32
    - SDA on 7100 to Pin 21 on ESP32
    - SCL on 7100 to Pin 22 on ESP32
    - SEL on 7100 to GND on ESP32
    - GND on 7100 to GND on ESP32
- Copy GitHub files and load code into VSCode (*must have Platform.io extension*).
- Flash ESP32 with code.
- Open up serial terminal (baud rate 115200) to view data.

More information / References / Troubleshooting

Piera GitHub for ESP32 Code Examples https://github.com/PieraSystems/7100-I2C-example

Wiki Page for I2C https://en.wikipedia.org/wiki/I%C2%B2C

Espressif ESP32 I2C Support Details:

https://docs.espressif.com/projects/esp-idf/en/v3.3.1/api-reference/peripherals/i2c.html

https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/peripherals/i2c.html

STM32 I2C Support Details:

https://www.digikey.com/en/maker/projects/getting-started-with-stm32-i2c-example/ba8c2bfef2024654b5dd10012425fa23

Support information

Contact us http://pierasystems.com/support

Please be willing to provide the following information about your I2C Bus

- Master Controller hardware, bus speed, total impedance if other devices on bus, minimum read bytes

Ordering information

Please visit www.pierasystems.com or email to info@pierasystems.com.