Getting Started with ICP-20100

InvenSense, a TDK Group Company
1745 Technology Drive, San Jose, CA 95110 U.S.A
+1(408) 988–7339
invensense.tdk.com

Document Number: AN-000270
Revision: 1.0
Release Date: 12/18/2020

TABLE OF CONTENTS

1 PURPOSE AND SCOPE

This document discusses some important features of TDK InvenSense pressure sensor ICP-20100 and provide application level information. Some of the topics covered in this application note include Modes of operation, FIFO, and Interrupt usage.

2 INTRODUCTION

The ICP-20100 pressure sensor family is based on MEMS capacitive technology, which provides high-accuracy, low power, barometric pressure, and temperature output. The pressure sensor operation range is 30 to 110 kPa with ±1 Pa accuracy, an accuracy enabling altitude measurement differentials as small as 8.5 cm, less than the height of a single stair step.

The pressure sensor features user-programmable interrupts, FIFO, and programmable output data rates, which makes it ideally suited for various applications such as mobile phones, wearable fitness monitoring, drones, and battery-powered IoT.

2.1 DEVICE INFORMATION

PART NUMBER	PACKAGE	LID OPENING
ICP-20100	2x2x0.8 mm LGA-10L	1-Hole

Table 1. Device information

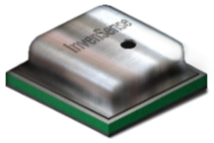

1-Hole Lid Opening
ICP-20100

Figure 1. Device information

3 OPERATION MODES FOR ICP-20100

3.1 INTRODUCTION FOR MODE 0 – 4

The ICP-20100 sensor features various operation modes and can be selected based on power consumption vs. output accuracy requirements of the application. Table 2 lists all the operation modes for the ICP-20100 sensor.

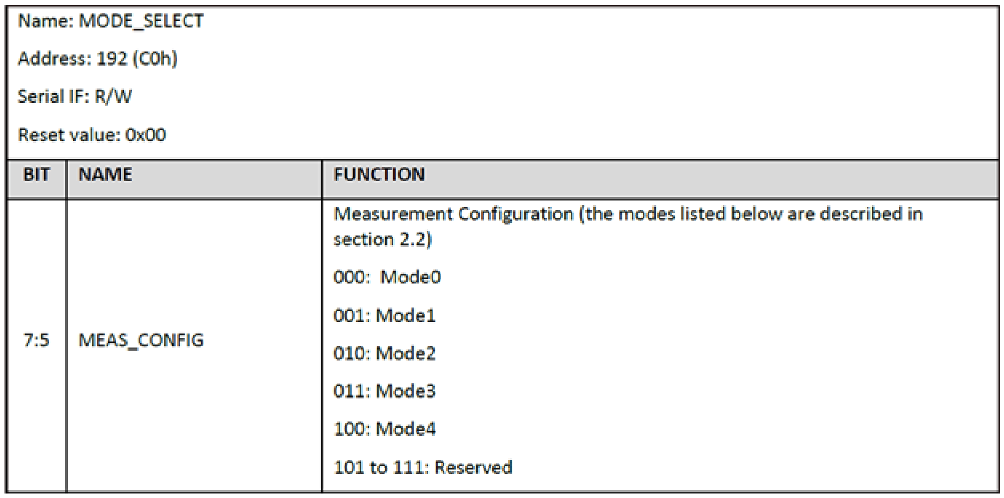
PARAMETER	BW (HZ)	ODR (HZ)	PRESSURE NOISE (PARMS)	CURRENT CONSUMPTION (UA)	IIR FILTER ENABLED	FIR FILTER ENABLED
			TYP	TYP		
MODE0*	6.25	25	0.5	211	No	Yes
MODE1*	30	120	1	222	No	Yes
MODE2*	10	40	2.5	49	No	Yes
MODE3*	0.5	2	0.5	23	No	Yes
MODE4**	12.5	25	0.3	250	No	No

Table 2. Operation modes

* MODE 0 – 3 requires initial 14 samples to be discarded because of FIR filter.
** MODE4 is user configurable as explained in the application note “AN-000238 ICP-20100 User Configurable Operation Mode and IIR Filter”

3.2 MODE 0 – 4 SELECTION THROUGH REGISTER

To enable required mode MEAS_CONFIG [bits 7:5] of MODE_SELECT register can be used, as shown below:



Name: MODE_SELECT

Address: 192 (C0h)

Serial IF: R/W

Reset value: 0x00

BIT	NAME	FUNCTION
7:5	MEAS_CONFIG	Measurement Configuration (the modes listed below are described in section 2.2) 000: Mode0 001: Mode1 010: Mode2 011: Mode3 100: Mode4 101 to 111: Reserved

Table 3. Mode selection through MODE_SELECT register

4 READING DATA THROUGH FIFO

4.1 FIFO WATERMARK FUNCTION

There are two FIFO Watermark Area in FIFO Config Register (0xC3)

FIFO Watermark High (bit 7-bit 4): When FIFO reaches the level defined in upward direction the interrupt is triggered.

FIFO Watermark Low (bit 3-bit 0): When FIFO reaches the level defined in downward direction the interrupt is triggered.

The FIFO_WMK_LOW_MASK bit 3 and FIFO_WMK_HIGH_MASK bit 2 should not be high in INTERRUPT_MASK (0xC2) registers.

For example, to have interrupt only when FIFO High Value of 10 and FIFO Low value of 2

0xC3 = 0xA2 and 0xC2 = 0x0C

Reg Name	Reg Addr (Hex)	Reg Value
INTERRUPT_MASK	0xC2	0x0C
FIFO_CONFIG	0xC3	0xA2

On interrupt

- Read INTERRUPT_STATUS register (0xC1)
- Read FIFO_FILL register and AND it with 0x1F. Result is number of packets in FIFO to be read.
- Read packets from FIFO (each packet is 3 bytes for pressure and 3 bytes for temperature).
- Read register FIFO base register (0xFA).
- Clear interrupt status register (0xC1).

4.2 FIFO POLLING FUNCTION

Poll packets from FIFO based on ODR (Mode Selected). FIFO base register is (0xFA) and 6 bytes are to be read 3 bytes for pressure and 3 bytes for temperature.

5 ICP-20100 INTERRUPT FUNCTIONS

ICP-20100 only has latched type interrupt.

This section explains how to start ICP-20100 in interrupt-based use cases and read data from ICP-20100.

5.1 FIFO OVERFLOW

This section explains how to configure ICP-20100 to start measurement, update FIFO, and raise interrupt when FIFO is full (16 pressure and temperature measurement pairs are filled)

- Update Interrupt mask register to enable FIFO Overflow interrupt

REG NAME	REG ADDR (HEX)	REG VALUE
INTERRUPT_MASK	0xC2	0xFE

- Update mode config register for ICP-20100 to start measurements in Mode0. Mode_config can be set to Mode0 to Mode4 based on requirements. Refer to the datasheet for Mode0 to Mode4 configurations.

REG NAME	REG ADDR (HEX)	REG VALUE
MODE_SELECT	0xC0	0x08 • Meas_config[7:5]: **000 for Mode 0 (001 for Mode1, 010 for mode 2, 011 for mode3)** • Meas_Trigger [4]: 0 • Meas_Mode[3]: 1 (continuous mode) • Power_Mode[2]: 0 • FIFO_Readout_mode [1:0]: 0 (Pressure first followed by temp)

On interrupt

- Read INTERRUPT_STATUS register (0xC1)
    - Int_status = regMap. INTERRUPT_STATUS
    - If FIFO_Overflow interrupt is not set, ignore interrupt.
- Read 16 packets (each packet is 3 bytes for pressure and 3 bytes for temp) i.e. 16 * 6 bytes.
- Read 96 bytes from FIFO base register (addr 0xFA).
- Clear interrupt status register (0xC1).
    - regMap. INTERRUPT_STATUS = Int_status

5.2 FIFO WATERMARK HIGH

This section explains how to configure ICP-20100 to start measurement and raise interrupt for every measurement.

- Update FIFO config register to set FIFO water mark high to 1

REG NAME	REG ADDR (HEX)	WRITE VALUE
FIFO_CONFIG	0xC3	0x10

Note: FIFO watermark can be set to any value from 1 to 16 as per requirement.

- Update Interrupt mask register to enable FIFO watermark high register.

REG NAME	REG ADDR (HEX)	WRITE VALUE
INTERRUPT_MASK	0xC2	0xFB

- Update mode config register for ICP-20100 to start measurements in Mode0. Mode_config can be set to Mode0 to Mode4 based on requirements. Refer to the datasheet for Mode0 to Mode4 configurations.

REG NAME	REG ADDR (HEX)	WRITE VALUE
MODE_SELECT	0xC0	0x08 • Meas_config [7:5]: **000 for Mode 0 (001 for Mode1, 010 for mode 2, 011 for mode3)** • Meas_Trigger [4]: 0 • Meas_Mode[3]: 1 (continuous mode) • Power_Mode[2]: 0 • FIFO_Readout_mode [1:0]: 0 (Pressure first followed by temp)

On Interrupt

- Read INTERRUPT_STATUS register (0xC1)
  - Int_status = regMap. INTERRUPT_STATUS
  - If FIFO watermark high interrupt is not set, ignore interrupt
- Read 1 packets (each packet is 3 bytes for pressure and 3 bytes for temp) i.e. 1 * 6 bytes.

Note: As FIFO watermark high is set to 1, we are reading one pair of pressure and temp. If FIFO watermark high value is set to some other value, number of bytes read from FIFO base address should change.

- Read 6 bytes from FIFO base register (addr 0xFA) and process data.
- Clear interrupt status register (0xC1).
  - regMap. INTERRUPT_STATUS = Int_status

5.3 ABSOLUTE PRESSURE THRESHOLD OVERRUN/UNDERRUN

This section explains how to configure ICP-20100 to raise interrupt if pressure underrun/overrun expected absolute pressure value.

- Set Pressure Abs value. Refer to the datasheet to compute pressure abs register value for expected pressure in kPa. For example, for the pressure value of 103 kPa pressure abs register value is 0x1A67.

REG NAME	REG ADDR (HEX)	WRITE VALUE
PRESS_ABS_LSB	0xC7	0x67
PRESS_ABS_MSB	0xC8	0x1A

- Update Interrupt mask register to enable Pressure Abs interrupt.

REG NAME	REG ADDR (HEX)	WRITE VALUE
INTERRUPT_MASK	0xC2	0xDF

- Update mode config register for ICP-20100 to start measurements in Mode0. Mode_config can be set to Mode0 to Mode4 based on requirements. Refer to the datasheet for Mode0 to Mode4 configurations.

REG NAME	REG ADDR (HEX)	WRITE VALUE
MODE_SELECT	0xC0	0x08 • Meas_config [7:5]: **000 for Mode 0 (001 for Mode1, 010 for mode 2, 011 for mode3)** • Meas_Trigger [4]: 0 • Meas_Mode[3]: 1 (continuous mode) • Power_Mode[2]: 0 • FIFO_Readout_mode [1:0]: 0 (Pressure first followed by temp)

On interrupt

- Read INTERRUPT_STATUS register (0xC1)
    - Int_status = regMap. INTERRUPT_STATUS
    - If Pressure abs interrupt is not set, ignore interrupt
- Read FIFO_FILL register (0xC4) for FIFO fill (number of packets in FIFO)
    - FIFO_fill = regMap. FIFO_FILL. FIFO_LEVEL
- Read FIFO_fill packets (each packet is 3 bytes for pressure and 3 bytes for temp) i.e. FIFO_fill * 6 bytes.
    - Bytes_to_Read = FIFO_fill * 6 bytes
- Read Bytes_to_Read from FIFO base register (addr 0xFA).
- Clear interrupt status register (0xC1).
    - regMap. INTERRUPT_STATUS = Int_status

5.4 DELTA PRESSURE VALUE OVERRUN

This section explains how to configure ICP-20100 to raise interrupt if delta pressure is overrun.

- Set Pressure delta value. Refer to the datasheet to compute pressure delta register value for expected delta pressure in kPa. For example, for pressure delta value of 0.5 kPa pressure delta register value is 0x66.

REG NAME	REG ADDR (HEX)	WRITE VALUE
PRESS_DELTA_LSB	0xC9	0x66
PRESS_DELTA_MSB	0xCA	0x00

- Update Interrupt mask register to enable Pressure Abs interrupt.

REG NAME	REG ADDR (HEX)	WRITE VALUE
INTERRUPT_MASK	0xC2	0xBF

- Update mode config register for ICP-20100 to start measurements in Mode0. Mode_config can be set to Mode0 to Mode4 based on requirements. Refer to the datasheet for Mode0 to Mode4 configurations.

REG NAME	REG ADDR (HEX)	WRITE VALUE
MODE_SELECT	0xC0	0x08 • Meas_config [7:5]: **000 for Mode 0 (001 for Mode1, 010 for mode 2, 011 for mode3)** • Meas_Trigger [4]: 0 • Meas_Mode[3]: 1 (continuous mode) • Power_Mode[2]: 0 • FIFO_Readout_mode [1:0]: 0 (Pressure first followed by temp)

On interrupt

- Read INTERRUPT_STATUS register (0xC1)
    - Int_status = regMap. INTERRUPT_STATUS
    - If pressure delta interrupt is not set, ignore interrupt
- Read FIFO_FILL register (0xC4) for FIFO fill (number of packets in FIFO)
    - FIFO_fill = regMap. FIFO_FILL. FIFO_LEVEL
- Read FIFO_fill packets (each packet is 3 bytes for pressure and 3 bytes for temp) i.e. FIFO_fill * 6 bytes.
    - Bytes_to_Read = FIFO_fill * 6 bytes
- Read Bytes_to_Read from FIFO base register (addr 0xFA).
- Clear interrupt status register (0xC1).
    - regMap. INTERRUPT_STATUS = Int_status

6 ICP-20100 TRIGGER MODE

Configuring ICP-20100 in trigger mode will allow ICP-20100 in enter standby mode and setting trigger for forced measure ICP-20100 will provide one measurement. This section will explain how to configure, trigger, and read data from ICP-20100.

- Update mode config register for ICP-20100 to stay in standby mode until trigger request is received.

REG NAME	REG ADDR (HEX)	WRITE VALUE
MODE_SELECT	0xC0	0x40 • Meas_config [7:5]: **100 for Mode 4** • Meas_Trigger [4]: 0 (stay in Standby mode) • Meas_Mode[3]: 0 (Standby mode) • Power_Mode[2]: 0 • FIFO_Readout_mode [1:0]: 0 (Pressure first followed by temp)

- Every time a measurement is needed, update mode config register to send measurement trigger.

REG NAME	REG ADDR (HEX)	WRITE VALUE
MODE_SELECT	0xC0	0x40 • Meas_config [7:5]: **100 for Mode 4** • Meas_Trigger [4]: 1 (Trigger for meas) • Meas_Mode[3]: 0 (Standby mode) • Power_Mode[2]: 0 • FIFO_Readout_mode [1:0]: 0 (Pressure first followed by temp)

Poll for 1 packet in FIFO

Read FIFO_FILL register (0xC4) for FIFO fill (number of packets in FIFO).

7 ICP-20100 WHO_AM_I READ PROCEDURE

To read WHO_AM_I we need to read two registers.

Read Register 0xD3 and make sure it is 0.

Then read register 0x0C and return the value. Value should be 0x63(99).

8 APPENDIX

8.1 PROCESSING DATA BUFFER READ FROM FIFO

Raw buffer read from buffer should be interpreted based on FIFO_READOUT_MODE field in MODE_SELECT register (Addr 0XC0). Refer to the datasheet for detailed information.

- FIFO Read out Mode set to Pressure first

Pressure and temperature are reported with pressure first. Raw buffer read from FIFO should be interpreted as below.

```
uint8_t i,offset=0;
for ( i = 0 ; i < packet_cnt ; i++ )  // packet_cnt is number of packets read from FIFO
{
pressure[i] = (int32_t)((( data[offset+2] & 0x0f) << 16) | (data[offset+1] << 8) | data[offset]) ;
offset += 3;
temperature[i] = (int32_t)(((data[offset+2] & 0x0f) << 16) | (data[offset+1] << 8) | data[offset]) ;
offset += 3;
}
```

9 REVISION HISTORY

REVISION DATE	REVISION	DESCRIPTION
12/18/2020	1.0	Initial Release

10 COMPLIANCE DECLARATION DISCLAIMER

InvenSense believes the environmental and other compliance information given in this document to be correct but cannot guarantee accuracy or completeness. Conformity documents substantiating the specifications and component characteristics are on file. InvenSense subcontracts manufacturing, and the information contained herein is based on data received from vendors and suppliers, which has not been validated by InvenSense.