FED-XXXG-NPT-I2C-3.3VDC Data Sheet

Product Naming & Performance Specifications

- **FED-XXXG-NPT-I2C-3.3VDC**
  - HydroSmart Technologies Digital Pressure Transducer
  - Lower Range Limit: 0 PSI
  - Upper Range Limit: **XXX** PSI
  - Pressure Type: Gage
  - Fitting Type: ¼ - 18 NPT
  - Supply Voltage: 3.3V DC
  - Interface: I2C
  - Digital Output: 10-90%
  - Pressure:
    - Output: 24-bit signed integer
    - Accuracy: ± 0.2% of range
  - Temperature:
    - Output: 24-bit signed integer
    - Accuracy: ± 1.25 °C
  - Housing: 316L stainless steel
  - Wetted Diaphragm: 316L stainless steel
  - Overpressure: 2X
  - Burst pressure: 3X
  - Operating Temperature: -40 °C to +125 °C
  - Media – Pressure: Liquids and gases compatible with 316/316L Stainless Steel

Figure 1 - Product Image

Product Wiring

- **Wiring Interface**
  - Green: VDD
  - Blue: SCL
  - Black: SDA
  - Red: GND

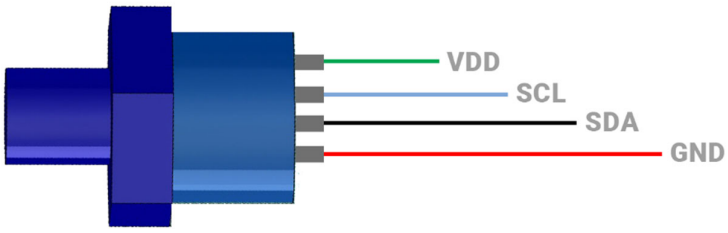

Figure 2 - Wiring Interface

I2C Communication Protocol Specifications

- **I2C Address**

A7	A6	A5	A4	A3	A2	A1	W/R
1	1	0	1	1	0	1	0/1

- **I2C Communication Pin Electrical Characteristics**

SYMBOL	PARAMETERS	CONDITION	MIN	MAX	UNIT
f_{scl}	CLOCK FREQUENCY			400	kHz
t_{LOW}	CLOCK LOW FREQUENCY HOLD TIME		1.3		μs
t_{HIGH}	CLOCK HIGH FREQUENCY HOLD TIME		0.6		μs
t_{SUDAT}	SDA SETUP TIME		0.1		μs
t_{HDDAT}	SDA SETUP TIME		0.0		μs
t_{SUSTA}	DATA SETUP TIME		0.6		μs
t_{HDSTA}	START CONDITION HOLD TIME		0.6		μs
t_{SUSTO}	STOP CONDITION SETUP TIME		0.6		μs
t_{BUS}	BUS FREE TIME BETWEEN STOP AND START CONDITION		1.3		μs

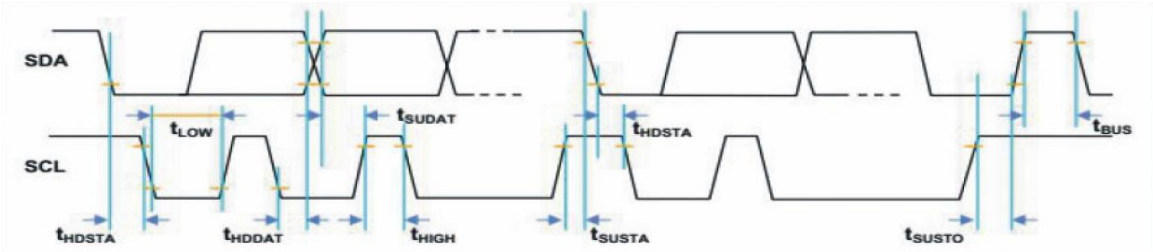

Figure 3 - I2C Timing Diagram

- **I2C Data Transfer Sequences**

(1): Pressure Measurement

Measuring data is a 24-bit signed integer, stored in 3 registers of address 0x06, 0x07, 0x08.

0x06								0x07								0x08							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Pressure Value (P) can be read as per below steps:

```
Lower_Range_Limit = 0;                          //Set Lower Range Limit in PSI
Upper_Range_Limit = 200;                        //Set Upper Range Limit in PSI
Range = Upper_Range_Limit - Lower_Range_Limit;   //Range is intermediate variable
float fadc;
iic_start();                                    //I2C start
iic_write(0xDA);                                //Device address,write operation
iic_write(0x06);                                //Write data start address
iic_start();                                    //Restart
iic_write(0xDA+1);                              //Device address, read operation
dat=iic_readbyte(1);                            //Read firstbyte(0x06)，ACK
dat <<= 8;                                      //Shift
dat += iic_readbyte(1);                         //Read second byte(0x07)，ACK
dat <<= 8;                                      //Shift
dat += iic_readbyte(0);                         //Read third byte(0x08)，NACK
iic_stop();                                     //Stop
//Calculate pressure value
if(dat & 0x800000)
    {
        fadc= dat - 16777216.0;
    }
    else
    {
        fadc = dat;
    }
ADC = 3.3* fadc /8388608.0;                     //ADC is intermediate variable
 P = (Range * (ADC-0.5)/2.0)+Lower_Range_Limit;  //P: pressure value in PSI
```

	PSI		Comments
ADC	1.5		*Example ADC value calculated from sensor I2C Value*
(ADC-0.5)/2.0	0.5		
Transducer Range	0	200	*Specified Transducer Operating Range (in PSI)*
Lower_Range_Limit	0		
Upper_Range_Limit	200		
Range	200		*Range = Upper_Range_Limit - Lower_Range_Limit*
P	**100**		*P = (Range * (ADC-0.5)/2) + Lower_Range_Limit*

Figure 4 - Example P Calculation (in PSI)

(2): Temperature measurement

Measuring data is a 24-bit signed integer, stored in 3 registers of address 0x09, 0x0A, 0x0B.

0x09								0x0A								0x0B							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Temperature Value (T) can be read as per below steps:

```
float fadc;
iic_start();                          //I2C start
iic_write(0xDA);                      //Device address,write operation
iic_write(0x09);                      //Write data start address
iic_start();                          //Restart
iic_write(0xDA+1);                    //Device address, read operation
dat=iic_readbyte(1);                  //Read firstbyte(0x09)，ACK
dat <<= 8;                            //Shift
dat += iic_readbyte(1);               //Read second byte(0x0A)，ACK
dat <<= 8;                            //Shift
dat += iic_readbyte(0);               //Read third byte(0x0B)，NACK
iic_stop();                           //Stop
//Calculate temperature value
if(dat & 0x800000)
    {
        fadc= dat - 16777216.0;
    }
    else
    {
        fadc = dat;
    }
T = 25.0+fadc /65536.0;               // T: temperature value in ℃
```