Logging Data and Touble-Shooting via Modbus

Document KFR.IN05.V2

Updated Aug. 4, 2017

KODAI FLOW RESEARCH

Disclaimer:
Use of this product is at the users own risk. No liability is implied except for the value of the measurement system. Every effort has been made to insure the equipment is designed to meet required standards in safety, emissions, and susceptibility, however it is left to the user to insure no interference with other equipment exists, and that the installation complies with codes and regulations.

Making Connection

Before logging data, connect to the transmitter with the methods described in KFR.IN05.V1. Make sure to use the correct Meter ID if on a multi-node Modbus network.

KFR Instrument Interface

Figure 1 is the landing screen of the PC application. Besides configuring the transmitter to output data on its analog and digital ports to the users PLC, it can be used to log data and trouble shoot the instrument.

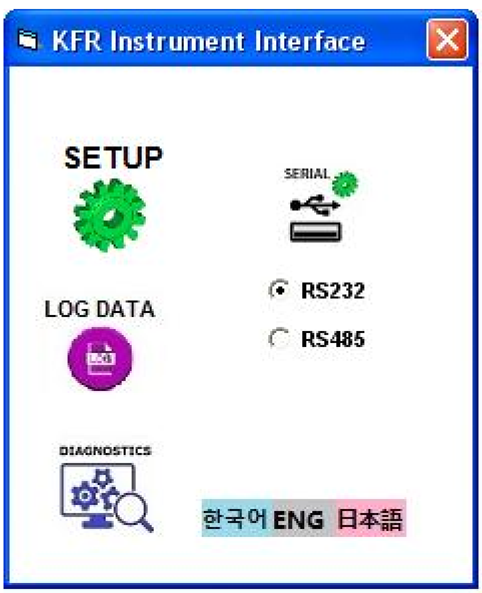

Figure 1 – Landing screen of Instrument Interface.

Diagnostics

To verify correct operation or trouble shoot the system, press the diagnostic icon on the Home Screen. The diagnostics screen will appear (Figure 2). Select the flow measurement channel you wish to examine. Set the update rate and select “Start”. If you wish to save the data to a file, then enter a file name or select an existing file and begin logging.

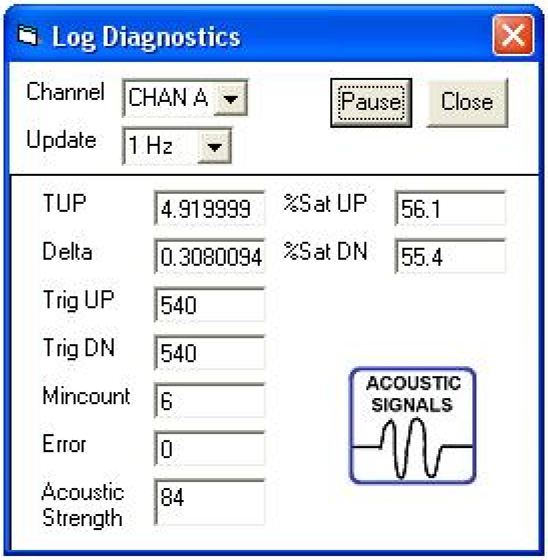

Figure 2 – Diagnostics Screen.

Below is a listing of the diagnostics and how to use them to find a problem.

1. TUP - This is the transit time from the downstream sensor to the upstream sensor (microsec).
2. Delta - This is the difference in arrival time between the up and down stream signals (nanosec).
3. Trig UP - The point in the upstream digital array that that is the approximate arrival time. Usually between 700 and 1200.
4. Trig DN - The point in the downstream digital array that that is the approximate arrival time. Usually between 700 and 1200.
5. Error - The code for the error condition of the meter

   0 – No Error

   1 – Gain Error, Amplifier out side gain limits (Dry Pipe)

   2 – Sound Speed Error, Measured sound speed outside programmed limits

   3 – Velocity Error, Measured velocity outside programmed limits

   4 – Saturation error – Received signal is outside range of ADC

   5 – Signal Quality

   6 – Signal Distortion
6. Acoustic Strength -

   The strength of the ultrasonic pulse that has traveled through the fluid. !00 is the maximum. Ideally it should be between 90 and 60. Signals weaker than 50 probably have a problem that needs addressed.
7. Sat% UP - The saturation of the 10-bit ADC converter in the upstream direction (60% ideal).
8. Sat% DN - The saturation of the 10-bit ADC converter in the downstream direction (60% ideal).

Press the “Acoustic Signals” icon on the screen to stop logging data and view the raw ultrasonic pulses that are being transmitted through the fluid. Select the channel to veiw and press “Down Load” to capture the signal. Either press save to file or capture the screen and send it to service for analysis.

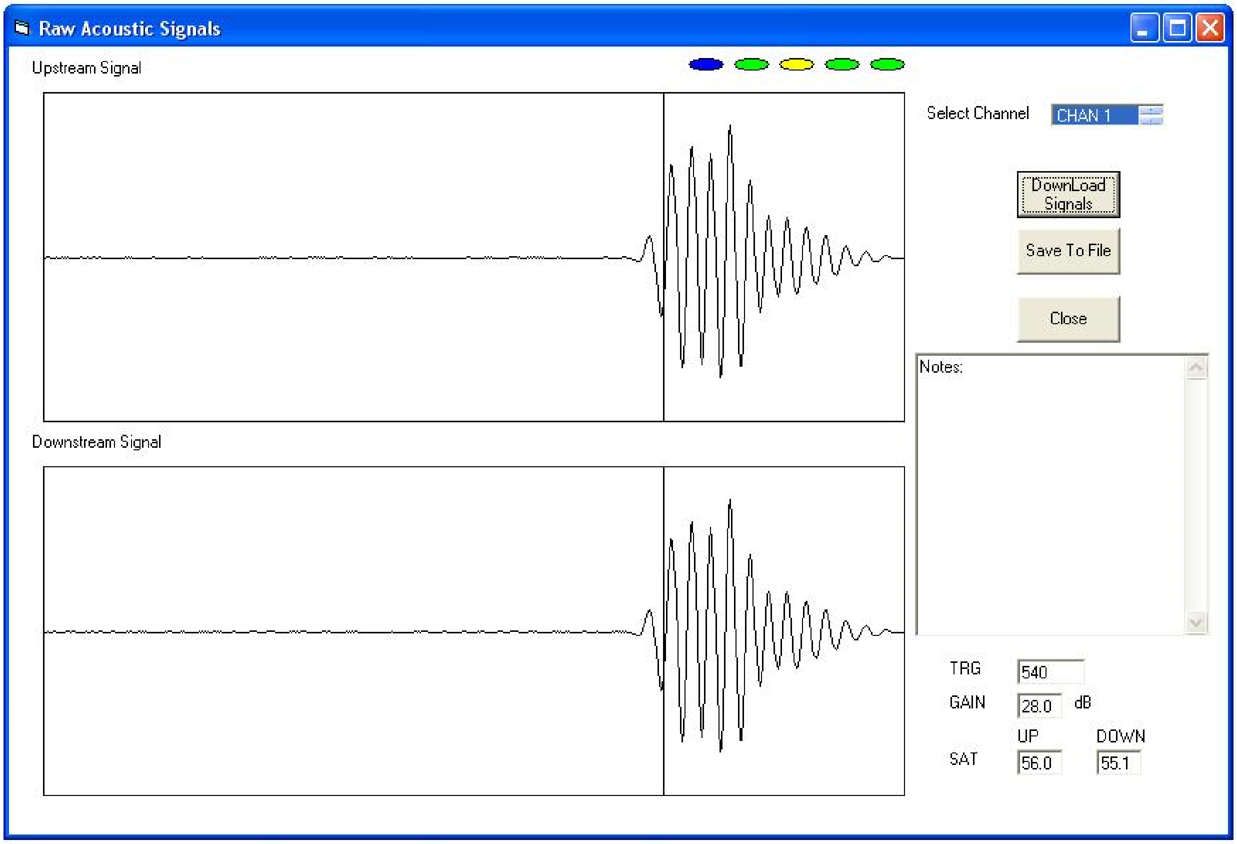

Figure 3 – Raw Acoustic Signals

Data logging:

Press the log icon to get the data logging screen of the application (Figure 4). Select whether to log data from Channel A, Channel B or both. Check the “Plot Data” to have the data displayed as a line graph on the screen. When plotting flow data from both channels, Channel A flow rate is plotted on the left axis (Blue) and Channel B flow is plotted on the right axis(Red).

When plotting data from just one channel, the user has the choice on to plot the Reynolds Number, Acoustic Velocity (Sound Speed), and size of particles or bubbles in the fluid.

Press start to begin logging data. The application will ask if you wish to store the data to a file. If you select an existing file the data will be appended to the bottom of the existing file. In addition to the flow rate, the transmitter logs several other parameters that are valuable in monitoring the process.

Acoustic Velocity – More commonly called sound speed, it is the velocity with which an acoustic wave travels through the fluid.

Acoustic Absorption - This is the level that sound energy is absorbed as it travels through the fluid. Its units are in decibels

Particle Loading -	This is an estimate using forward scattering of the percentage of the volume that is occupied by solid particle or suspended bubbles.
Particle Size -	Also using forward scattering, this is an estimate of the 1σ large particle size. This used with particle loading is used to give an indication of void fraction.

Figure 3 – Data Logging Screen in Instrument Interface

Direct Modbus Commands:

For use with system SCADA, PLC, or HMI (human machine interface), the same data can be retrieved using a Modbus-16 serial protocol. The system polls the meter using a 10 byte command. The first byte is ID of the instrument and the second byte is the command 30. Byte 5 is used to control batch total functions. The other bytes of are not used and can be set at any value. Bytes 8 and 9 are used to store the 16 bit cyclic redundancy check using the standard CRC-16 format. Example C software for calculating the Modbus CRC is follows in an appendix.

8 Bit Byte	Name	Value	Detail
0	Meter ID	17	Default 17, 1-15 on RS485 multi-drop networks
1	Command	30	Get Data Command
2	Reserved	0	MSB of Encrypted Password Used for Programming
3	Reserved	0	LSB of Encrypted Password Used for Programming
4	Reserved	0	
5	Total Mode	0,1,2	0 Stop Total, 1 Start Total, 3 Clear Total
6	Reserved	0	
7	Reserved	0	
8	MSB CRC	Calc	
9	LSB CRC	Calc	

Figure 4 – 10 Byte Query Command

The transmitter returns a 54 byte message with the last two bytes containing the CRC-16 check of the first 52 bits. The system data is packed in binary as shown in Figure 5.

8 Bit Byte	Type	Value
0	Meter ID	Echo
1	Command	Echo
2	Reserved	0
3	Reserved	0
4	MSB Float	Channel 1 Average Liquid Velocity in Meters per Second
5	BYTE 3	32 Bit Floating Point Format
6	BYTE 2	
7	LSB Float	
8	MSB Float	Channel 1 Liquid Velocity of Sound, Meters per Second
9	BYTE 3	32 Bit Floating Point Format
10	BYTE 2	
11	LSB Float	
12	MSB Float	Channel 1 Volume Flow Rate, Cubic Meter per Second
13	BYTE 3	32 Bit Floating Point Format
14	BYTE 2	
15	LSB Float	
16	MSB Float	Channel 1 Reynolds Number, Dimensionless
17	BYTE 3	32 Bit Floating Point Format
18	BYTE 2	
19	LSB Float	
20	MSB Float	Totalizer Clock Time in Seconds
21	BYTE 3	32 Bit Floating Point Format
22	BYTE 2	
23	LSB Float	
24	MSB Float	Channel 1 Volume Total (Batch) in Cubic Meters
25	BYTE 3	32 Bit Floating Point Format
26	BYTE 2	
27	LSB Float	
28	MSB Float	Channel 2 Average Liquid Velocity in Meters per Second
29	BYTE 3	32 Bit Floating Point Format
30	BYTE 2	
31	LSB Float	
32	MSB Float	Channel 2 Liquid Velocity of Sound, Meters per Second
33	BYTE 3	32 Bit Floating Point Format
34	BYTE 2	
35	LSB Float	
36	MSB Float	Channel 2 Volume Flow Rate, Cubic Meter per Second
37	BYTE 3	32 Bit Floating Point Format
38	BYTE 2	
39	LSB Float	
40	MSB Float	Channel 2 Reynolds Number, Dimensionless
41	BYTE 3	32 Bit Floating Point Format
42	BYTE 2	
43	LSB Float	
44	MSB Float	Channel 2 Volume Total (Batch) in Cubic Meters
45	BYTE 3	32 Bit Floating Point Format
46	BYTE 2	
47	LSB Float	
48	Total Mode	Present Totalizing Mode 0-Stopped, 1-Totalizing
49	Alarm1	0 – Error Free, 1 Error Condition
50	Alarm 2	0 – Error Free, 1 Error Condition
51	Reserved	0
52	CRC MSB	CRC-16 Value for the 52 Byte String For Error Check
53	CRC LSB	

Figure 5 – 54 Byte Data Message From Meter

Appendix – CRC-16 Example Code in ANSI-C

Note, the following code is written in TI notation for a machine with fixed 16 bit addressing.

```
#define P_16  0xA001
static Uint16 crc_tab16[256];

//******************************************************************************************
// Function CalcCRC( binary array with leading byte blanked to zero, length of array)
//******************************************************************************************
Uint16 CalcCRC(volatile Uint16 *bIn,Uint16 bLen) {
        Uint16 i;
        Uint16 crc_16;

        crc_16=0xffff;

        for (i=0;i<bLen;i++)
        {
                crc_16 = update_crc_16(crc_16,bIn[i]);
        }
        return crc_16;
}
//*******************************************************

//  Update the CRC
//*******************************************************
Uint16 update_crc_16( Uint16 crc, Uint16 c ) {

        Uint16 tmp;
        Uint16 short_c;

        short_c = 0x00ff &  c;
        tmp =  crc      ^ short_c;
        crc = (crc >> 8) ^ crc_tab16[ tmp & 0xff ];

        return crc;

}
//**********************************************************
```

```
//  Load the crc table with values for calculation on start_up
//***********************************************************
void init_crc16_tab( void ) {

        Uint32 i, j;
        Uint16 crc, c, crctemp;

        for (i=0; i<256; i++) {
                crc = 0;
                c   = (Uint16) i;
                for (j=0; j<8; j++)
                {
                        if ( (crc ^ c) & 0x0001 )
                                crctemp = ( crc >> 1 ) ^ P_16;
                        else
                                crctemp =   crc >> 1;
                        crc=crctemp;
                        c = c >> 1;
                }

                crc_tab16[i] = crctemp;

        }
}
//**********************************************************************
```

45570 Industrial Place #10
Fremont CA, 94538
Phone: (978) 557-0057
Fax: (978) 557-0058
Email: info@kodaiflow.com