APPLICATION NOTE 6105

CONFIGURING THE MAX35101 TIME-TO-DIGITAL CONVERTER AS AN ULTRASONIC WATER METER

By: Ben Smith, Software Manager

Abstract: With water resources becoming increasingly constrained, it is important to deploy accurate and flexible metering capability. Older, mechanical water meters have poor performance at low flow rates and cannot be networked. In this application note, we show how the MAX35101 time-to-digital converter can be used to build a water meter that overcomes the limitations of a mechanical meter.

Introduction

The single most critical natural resource on our planet is water. And as the population of our planet increases, managing the distribution of this precious resource becomes more and more critical. In regions of high population density water shortages are becoming an increasingly frequent fact of life.

For these reasons, it is clear that the old ways of deploying mechanical meters and reading them once a month for billing purposes is no longer sufficient. Meters must be updated to be more accurate, more sensitive to small leaks, and able to keep track of when water is actually used. In this application note, we will look at how the MAX35101 time-to-digital converter can be used to build a water meter that meets all of these criteria.

Using Sound to Measure Flow

At its core, the MAX35101 is an extremely precise interval timer. It can report the time interval between two events down to a resolution of 4ps. With this resolution, once a baseline time is established, anything that disturbs the time interval between the start event and the stop event can be measured with a high degree of precision.

The MAX35101 also contains a pair of pulse drivers and receivers. These drivers and receivers are connected to ultrasonic transducers to launch and detect pulses in a conductive medium (see **Figure 1**). The MAX35101 launches a pulse and, a few microseconds later, detects the pulse in the other transducer. The time that elapses from the launch time to the detection time is called the time of flight (TOF).
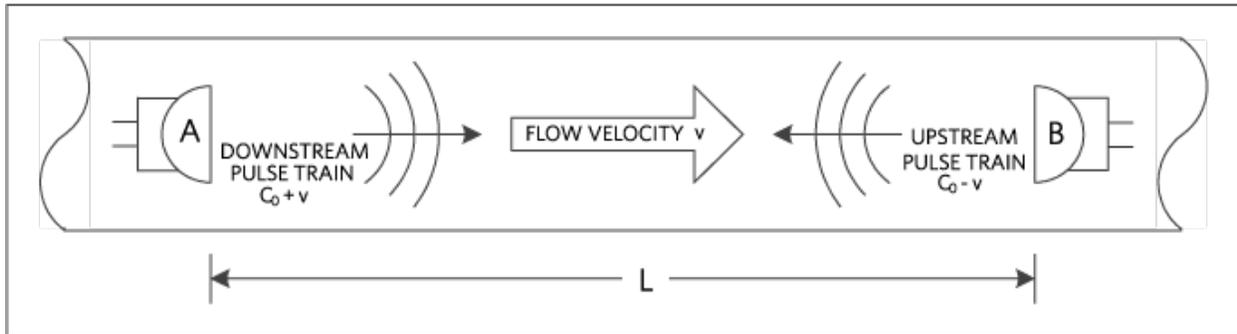
Figure 1. Transducer arrangement in a pipe to measure water flow.

If there are two transducers, whether I launch the pulse from transducer 'A' and receive the pulse in transducer 'B', or I launch the pulse from transducer 'B'and receive the pulse in transducer 'A' should make no difference—the distance is the same, and so the TOF should be the same. But if the medium—say, water in a pipe—is flowing, the pulse launched from the upstream transducer will take less time to reach the downstream transducer than if the medium is stationary. Similarly, a pulse launched from the downstream transducer will take a longer period to reach the upstream transducer in a flowing medium versus a stationary medium.

And that, in very simple terms, is how you use sound to measure flow. Launch a pulse in the downstream direction and count the time it takes to be received, then launch a pulse in the upstream direction and count the time that pulse takes to be received. If the two times are equal, then the medium is stationary. But if the upstream count is greater than the downstream count, the difference gives you a very precise indication of the velocity of the medium. Factor in the pipe diameter and apply a conversion constant, and you can report the flow rate in your units of choice—gallons per minute, liters per second, or cubic meters per hour.

Doing the Math

The velocity of propagation in any medium under stated conditions of temperature and pressure is generally considered to be a constant, C_0. For a particular path length, L, the propagation time is given by:

$$t_p = \frac{L}{C_0}$$

This formula applies only if the medium is stationary. If the medium is moving in a direction parallel to and in the same direction as the pulse path, the propagation velocity is increased and the propagation time is reduced:

$$t_p = \frac{L}{C_0 + v}$$

Similarly, if the medium is moving in a direction parallel to and in the opposite direction from the pulse path, the propagation velocity is decreased and the propagation time is increased:

$$t_p = \frac{L}{C_0 - v}$$

Now, if we take a measurement in the upstream direction and a second measurement in the *downstream* direction and take the difference, we obtain this:

$$\Delta t_p = \frac{L}{C_0 - v} - \frac{L}{C_0 + v}$$

We can find a common denominator on the right side of the equation:

$$\Delta t_p = \frac{L(C_0 - v) - L(C_0 + v)}{C_0^2 - v^2}$$

But the velocity of sound in the medium is much greater than the velocity of the medium, so we can eliminate the v^2 term. We can also cancel the C_0 term in the numerator.

$$\Delta t_p = \frac{2Lv}{C_0^2}$$

Solving this equation for v we obtain:

$$v = \frac{\Delta t_p \times C_0^2}{2L}$$

If C_0 is assumed to be about 1,497m/s at room temperature, and if we assume a path length of 10cm, then the fluid velocity would be the time difference multiplied by about 11.2×10^6m/s. Conversely, a velocity of 1m/s would provide a time difference of 89.2ns; at a resolution of 4ps, this would give rise to about 22,311 counts.

Since the equation strongly depends on the velocity of sound in water, and since that is strongly dependent on temperature, it is necessary to measure the temperature and adjust C_0 accordingly. The MAX35101 has interfaces for as many as four RTD-type temperature sensors.

How the MAX35101 Works

To take a TOF measurement, host software sends a command to begin the measurement. Here's what happens next (**Figure 2**):
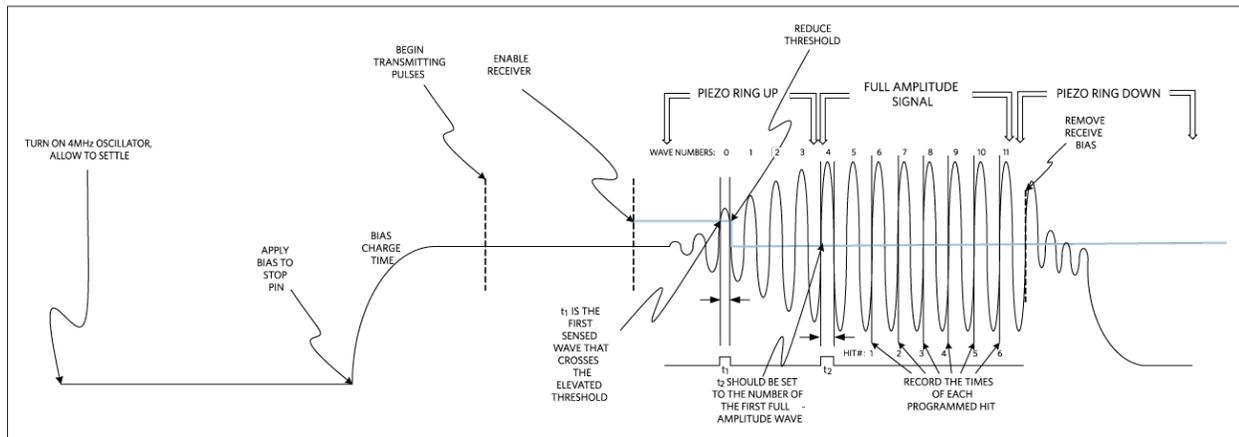
Figure 2. Timing in a MAX35101-based flow meter.

- A 4MHz oscillator is turned on and stabilizes. This oscillator can use an external crystal or an external resonator for reference. While a crystal oscillator provides the best accuracy and stability, resonators are generally less expensive and may be more suitable for cost-sensitive applications.
- Once the oscillator is stable, a bias voltage is applied to the receiver pin. Some time is provided to allow the receiver pin to settle.
- After the bias on the receiver pin has stabilized, the transmitting transducer is activated to begin sending a series of pulses. The number and timing of pulses is programmable. When the first pulse is transmitted, the timer begins running.
- After some programmable time, the receiver is turned on to begin listening for the incoming pulses. The receive comparator threshold is held at an elevated level to eliminate triggering on noise.
- The first pulse to exceed the measurement threshold is designated the T_1 wave. The 'on' period of this pulse is measured and stored. Once the T_1 wave has been measured, the comparator threshold is reduced to roughly the midpoint of the receive range.
- Now, a programmable number of pulses are skipped. The next pulse to be measured—ideally, the first full-amplitude pulse—is called the T_2 wave. That pulse is measured and the 'on' period is stored.
- Next, one or more of the subsequent pulses are detected. These are called "hits." Each pulse designated as a "hit" is correlated to the beginning of the T_1 pulse; consequently, in just one TOF measurement several actual TOF samples can be taken.
- After the final "hit" has been sensed, the MAX35101 turns off the receive bias and begins performing calculations and storing them in registers for the host microcontroller to retrieve. Once all of the calculations have been performed, an interrupt is generated toward the host microcontroller and the active circuitry in the MAX35101 is turned off to await the next command.

This sequence will be repeated in the opposite direction to compute the differential TOF. Once both TOF measurements have been complete, the host microcontroller can compute the flow velocity, and then the flow rate.

Connections

Of course, doing anything with the MAX35101 requires communicating with it. The MAX35101 has a standard SPI slave port that expects to be connected to a SPI master belonging to the host microcontroller. The MAX35101 SPI peripheral operates in SPI mode 1, which means that the SCLK circuit idles in a low

state and that data is expected to change on the rising edge of the clock and be sampled on the falling edge.

The SPI slave connects directly to a register interface that contains three kinds of registers:

Command registers (0x00–0x0E): When the MAX35101 sees a byte in this range on the SPI interface, it interprets the event as a command. This is how the host microcontroller requests a measurement, resets, or initializes the part or halts a measurement in progress. No further information is required for any of the commands. As soon as the command is received an internal state machine begins executing the command without further intervention by the host microcontroller. The MAX35101 returns nothing on the MISO circuit.

Configuration and status registers (0x30–0x43 for write), (0xB0–0xFF for read): These registers provide access to the internal configuration registers for the host microcontroller, and reflect internal status when read. When the host microcontroller sends a byte in the write range (0x30–0x43), the MAX35101 accepts a 16-bit word to store in the designated configuration register. When the host microcontroller sends a byte in the read range (0xB0–0xFF), the MAX35101 drives the 16 bits from the register onto the MISO circuit.

Flash registers (0x90, 0x10, and 0x13): These three registers provide access to the 8KB block of flash used to store calibration information and other persistent information.

Setting up the MAX35101

1. Set up the TOF1 register. This sets up the basic operating parameters for TOF measurements.

TOF1 (Write: 0x38 Read: 0xB8)

Bit	15	14	13	12	11	10	9	8
Name					DPL			
Access					R/W			
Reset					0000 0000			

Bit	7	6	5	4	3	2	1	0
Name		DPL			STOP_POL	RESERVED	CT	
Access		R/W			R/W	R	R/W	
Reset		0010			0	0	00	

Bit	Description
TOF1[1:0]– CT[1:0]	**Bias Charge Time.** This field sets up the time from activating bias voltage on the receive pin to the launch of the first pulse.

CT[1:0]	32kHz Clocks	Time (µs)
0b00	2	61
0b01	4	122
0b10	8	244
0b11	16	488

In general, choose the shortest charge time consistent with good results. Increasing bias charge time increases the active time of the MAX35101, increasing battery usage.

Bit	Description
TOF1.2	**Reserved.** Reading this bit returns zero; writing has no effect.
TOF1.3– STOP_POL	**Stop Polarity.** This bit defines which edge on the receive input will be used as the leading edge. If zero, the rising edge will be considered the leading edge and the period measurement will count the period the signal remains above the measurement threshold. If one, the falling edge will be considered the leading edge and the period measurement will count the period the signal remains below the measurement threshold.

TOF1[7:4]– **Pulse Launch Divider.** This field determines the frequency of the pulse train emitted by the
DPL[3:0   transmitter.

DPL[3:0]	Frequency
0b0000	Reserved
0b0001	1MHz
0b0010	667kHz
0b0011	500kHz
0b0100	400kHz
0b0101	333kHz
0b0110	286kHz
0b0111	250kHz
0b1000	222kHz
0b1001	200kHz
0b1010	182kHz
0b1011	167kHz
0b1100	154kHz
0b1101	143kHz
0b1110	133kHz
0b1111	125kHz

TOF1[15:8]– **Pulse Launcher Size.** This field sets the number of pulses to be emitted by the transmitter.
PL[7:0]   Setting the field to zero disables the pulse launcher entirely. Any other value allows the
pulse launcher to operate, and when a TOF command is received, the pulse launcher emits
the number of pulses indicated in this field, up to 127 pulses. Values greater than 127 are
capped at 127 pulses.

2. Set up the TOF2 register. This sets up the details of how TOF will be measured.

TOF2 (Write: 0x39 Read: 0xB9)

Bit	15	14	13	12	11	10	9	8
Name	STOP				T2WV[5:1]			
Access	R/W				R/W			
Reset	000				000			

Bit	7	6	5	4	3	2	1	0
Name	T2WV.0	TOF_CYC			RESERVED	TIMEOUT		
Access	R/W	R/W			R	R/W		
Reset	0	000			0	00		

Bit	Description
TOF2[2:0]–TIMOUT[2:0]	**Timeout.** This field sets the timeout for the time-to-digital measurement block. The timeout counter is reset at the beginning of the first transmitted pulse. If any expected pulse does not arrive before the timeout counter expires, the TO bit in the interrupt status register is set and the measurement is aborted.

TIMOUT[2:0]	Time (µs)
0b000	128
0b001	256
0b010	512
0b011	1,024
0b100	2,048
0b101	4,096
0b110	8,192
0b111	16,384

TOF2.3	**Reserved.** Reading this bit returns zero; writing has no effect.

TOF2[6:4]– TOF_CYC[2:0]	**TOF Duty Cycle.** This field determines the delay between successive TOF measurements when executing a TOF_DIFF command. The time delay is measured from the start time of the upstream TOF measurement to the start of the downstream TOF measurement. If the actual TOF of the upstream measurement exceeds the delay time, the downstream TOF measurement begins immediately after the upstream measurement is complete – that is, the system behaves as though TOF_CYC were zero. Note that for longer delays (TOF_CYC=0b110 and TOF_CYC=0b111) the 4MHz oscillator is gated off between measurements.

TOF_CYC[2:0]	32kHz Cycles	Time	4MHz On Between Measurements?
0b000	0	0µs	Yes
0b001	4	122µs	Yes
0b010	8	244µs	Yes
0b011	16	488µs	Yes
0b100	24	732µs	Yes
0b101	32	976µs	Yes
0b110	546	16.65ms	No
0b111	655	19.97ms	No

TOF2[12:7]– T2WV[5:0]	**T_2 Wave Selection.** This field determines which wave number will be measured for T_2. You should empirically determine which wave should be selected for T_2 and program the wave number here. The wave you select should be the first full-height wave after the early-detect wave. Any value less than two in this field will select wave 2 as the T2 wave; any value greater than two will select that number wave.

TOF2[15:13]– STOP[2:0]	**Stop Hits.** This field selects the number of waves that will be measured as HITx waves. The particular waves that will be selected as HITx waves are chosen in the TOF3, TOF4, and TOF5 registers. After the measurement is taken, the time result for all HITx waves will be averaged and reported.

3. Set up the TOF3, TOF4, and TOF5 registers, as needed. These registers contain fields that select the particular waves to use as HITx waves; that is, they select which waves will be used in time measurements.

TOF3 (Write: 0x3C Read: 0xBA)

Bit	15	14	13	12	11	10	9	8
Name	RESERVED				HIT1			
Access	R				R/W			
Reset	00				00 0000			
Bit	7	6	5	4	3	2	1	0
Name	RESERVED				HIT2			
Access	R				R/W			
Reset	00				00 0000			

Bit	Description
TOF3[5:0]–HIT2[5:0]	**HIT2 Wave select.** This field selects the wave number for the HIT2 event. The value in the HIT2 field must be at least one greater than the value in the HIT1 field. Wave 4 is the earliest wave that can be used for HIT2; values less than 0x04 will cause wave 4 to be measured.
TOF3[7:6]	**Reserved.** Reading this field returns zero, and writing has no effect.
TOF3[13:8]–HIT1[5:0]	**HIT1 Wave select.** This field selects the wave number for the HIT1 event. The value in the HIT1 field must be at least one greater than the value in the T2WV field in the TOF2 register. Wave 3 is the earliest wave that can be used for HIT1; values less than 0x03 will cause wave 3 to be measured.
TOF3[15:14]	**Reserved.** Reading this field returns zero, and writing has no effect.

TOF4 (Write: 0x3C Read: 0xBB

Bit	15	14	13	12	11	10	9	8
Name	RESERVED				HIT3			
Access	R				R/W			
Reset	00				00 0000			
Bit	7	6	5	4	3	2	1	0
Name	RESERVED				HIT4			
Access	R				R/W			
Reset	00				00 0000			

Bit	Description
TOF4[5:0]–HIT4[5:0]	**HIT4 Wave select.** This field selects the wave number for the HIT4 event. The value in the HIT4 field must be at least one greater than the value in the HIT3 field. Wave 6 is the earliest wave that can be used for HIT4; values less than 0x06 will cause wave 6 to be measured.
TOF4[7:6]	**Reserved.** Reading this field returns zero, and writing has no effect.
TOF4[13:8]–HIT4[5:0]	**HIT3 Wave select.** This field selects the wave number for the HIT3 event. The value in the HIT3 field must be at least one greater than the value in the HIT2 field in the TOF3 register. Wave 5 is the earliest wave that can be used for HIT3; values less than 0x05 will cause wave 5 to be measured.
TOF4[15:14]	**Reserved.** Reading this field returns zero, and writing has no effect.

TOF5 (Write: 0x3C Read: 0xBC)

Bit	15	14	13	12	11	10	9	8
Name	RESERVED				HIT5			
Access	R				R/W			
Reset	00				00 0000			
Bit	7	6	5	4	3	2	1	0
Name	RESERVED				HIT6			
Access	R				R/W			
Reset	00				00 0000			

Bit	Description
TOF5[5:0]–HIT6[5:0]	**HIT6 Wave select.** This field selects the wave number for the HIT6 event. The value in the HIT6 field must be at least one greater than the value in the HIT5 field. Wave 8 is the earliest wave that can be used for HIT6; values less than 0x08 will cause wave 8 to be measured.
TOF5[7:6]	**Reserved.** Reading this field returns zero, and writing has no effect.
TOF5[13:8]–HIT5[5:0]	**HIT5 Wave select.** This field selects the wave number for the HIT5 event. The value in the HIT5 field must be at least one greater than the value in the HIT4 field in the TOF4 register. Wave 7 is the earliest wave that can be used for HIT5; values less than 0x07 will cause wave 7 to be measured.
TOF5[15:14]	**Reserved.** Reading this field returns zero, and writing has no effect.

Related Parts		
MAX35101	Time-to-Digital Converter with Analog Front-End	Free Samples
MAX35101EVKIT	Evaluation Kit for the MAX35101	
MAX35102	Time-to-Digital Converter Without RTC	Free Samples
MAX35103	Reduced Power Time-to-Digital Converter with AFE, RTC, and Flash	Free Samples
MAX35103EVKIT	Evaluation Kit for the MAX35103	

More Information

For Technical Support: https://www.maximintegrated.com/en/support
For Samples: https://www.maximintegrated.com/en/samples
Other Questions and Comments: https://www.maximintegrated.com/en/contact

Application Note 6105: https://www.maximintegrated.com/en/an6105
APPLICATION NOTE 6105, AN6105, AN 6105, APP6105, Appnote6105, Appnote 6105
© 2014 Maxim Integrated Products, Inc.
The content on this webpage is protected by copyright laws of the United States and of foreign countries.
For requests to copy this content, contact us.
Additional Legal Notices: https://www.maximintegrated.com/en/legal