Designing a Homemade Digital Output for Analog Voltage Output Sensors

by: Eric Jacobsen, Systems and Applications Engineer
Sensor Products Division, Phoenix, AZ

INTRODUCTION

A digital output is more desirable than an analog output in noisy environments (e.g., automotive, washing machines) and remote sensing applications (e.g., building controls, industrial applications) because a digital signal inherently has better noise immunity compared to analog signals. Additional applications requiring a sensor with a digital output include microcontroller-based systems that have no A/D in the system or that have no A/D channels available for the sensing function. For these applications, there is no other option but a digital output to further process the signal.

Via a design example this paper shows how to easily convert an analog voltage output sensor to a digital output sensor. For the design example, each of the required circuit components is discussed in detail. While the design is applicable to analog voltage output sensors (differential or single-ended output) in general, the design example and following discussions will pertain specifically to semiconductor pressure sensors.

The digital output sensor in Figure 1 consists of the following:

- MPX2000 series pressure sensor
- A two op amp gain stage to amplify the sensor's signal
- An integrator (i.e., a low pass filter consisting of one resistor and one capacitor)
- An LM311 comparator
- An MC68HC05P9 microcontroller with which only two pins are used: the output compare timer channel (TCMP) and one general I/O pin (the input capture timer channel, TCAP, can be used in place of the general I/O pin). Since only two of the MC68HC05P9's pins are used, the remaining pins are available for other system functions.

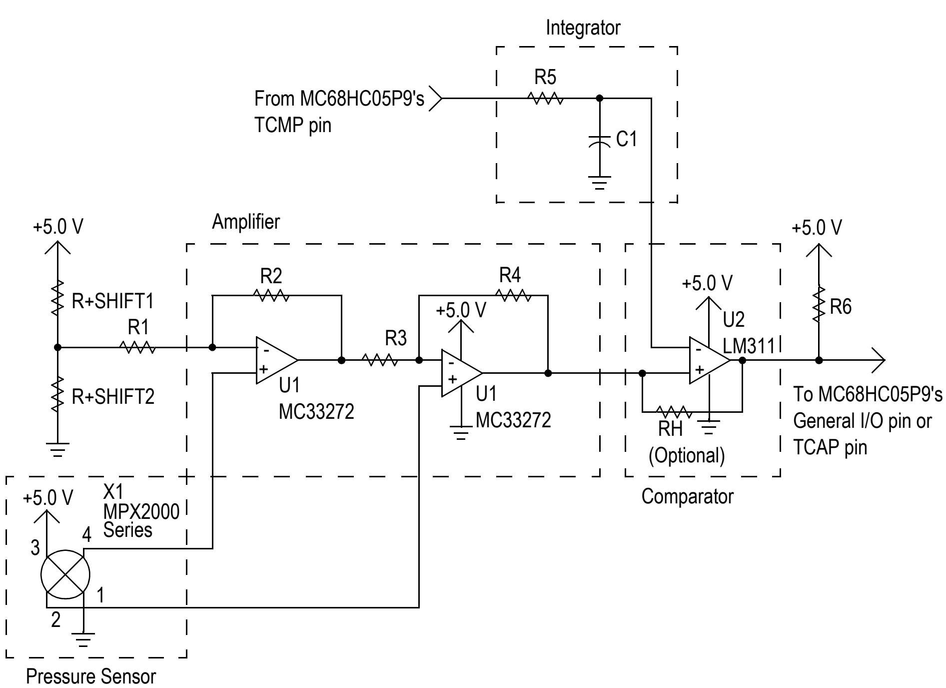

Figure 1. The Digital Output Sensor Schematic

© Freescale Semiconductor, Inc., 2006. All rights reserved.

After the discussion of the circuit components, the following system-related issues will be discussed simultaneously using the design example:

- How the system works
- Defining and designing the digital output for a desired signal resolution
- A step-by-step procedure that shows you how to digitize the signal
- A procedure to show you how to software calibrate the digital output
- Related software examples

This system, in addition to the benefits of a digital output (noise immunity, etc.), also has the following additional inherent benefits. These benefits will be addressed in more detail in the systems topics.

- The circuit topology and method of "digitizing" the sensor's analog output is very stable and accurate. The system uses the microcontroller's precise, internal, digital time base to digitize the analog signal.
- The signal resolution is user-programmable via software - i.e. the user can program whether the resolution is 8-bit, 10-bit, etc.
- The digital output is calibrated in software so that component tolerances can be nullified.
- The software required to digitize the signal requires very little CPU time and overhead.
- The required circuitry is minimal, simple, and cost-effective.

THE PRESSURE SENSOR

The Freescale Semiconductor, Inc. MPX2000 series sensors are temperature compensated and calibrated (i.e., offset and span are precision trimmed) pressure transducers. These sensors are available in full scale pressure ranges from 10 kPa (1.5 psi) to 700 kPa (100 psi). Although the specifications (see Table 1) in the data sheets apply to a 10 V supply voltage, the output of these devices is ratiometric with the supply voltage. For example, at the absolute maximum supply voltage rating, 16 V, the sensor will typically produce a differential output voltage of 64 mV at the rated full scale pressure of the given sensor. One exception to this is that the span of the MPX2010 (10 kPa sensor) will be only 40 mV due to the device's slightly lower sensitivity. Since the maximum supply voltage produces the largest output signal, it is evident that even the best case scenario will require some signal conditioning to obtain a usable signal (input to an A/D, for example). For this specific design, an MPX2100 and 5.0 V supply are used, yielding a typical maximum sensor output of 20 mV (typical zero pressure offset is 0.0 mV and typical span is 20 mV). The sensor's output is then signal conditioned (amplified and level shifted) to provide a four volt span with a zero pressure offset of 0.5 V.

Table 1. MPX2100 Electrical Characteristics for V_S= 10 V, T_A = 25°C

Characteristics	Symbol	Min	Typ	Max	Unit
Pressure Range	P_{OP}	0	—	100	kPa
Supply Voltage	V_S	—	10	16	V_{DC}
Full Scale Span	V_{FSS}	38.5	40	41.5	mV
Zero Pressure Offset	V_{OFF}	-1.0	—	1.0	mV
Sensitivity	$\Delta V/\Delta P$	—	0.4	—	mV/kPa
Linearity	—	-0.25	—	0.25	$\%V_{FSS}$
Temperature Effect on Span	TCV_{FSS}	-1.0	—	1.0	$\%V_{FSS}$
Temperature Effect on Offset	TCV_{OFF}	-1.0	—	1.0	mV

AMPLIFIER STAGE

The amplifier circuitry, shown in Figure 1, is composed of two op amps. This interface circuit has a much lower component count than conventional quad op amp instrumentation amplifiers. The two op-amp design offers the high input impedance, low output impedance, and high gain desired for a transducer interface, while performing a differential to single-ended conversion. The amplifier incorporates level shifting capability. The amplifier has the following transfer function:

$$V_O = \left(1 + \frac{R_4}{R_3}\right) \cdot (V_{sensor}) + V + shift$$

where R1 = R4; R2 = R3,

the gain is $1 + \frac{R_4}{R_3}$

V_{sensor} is the sensor's differential output ($S^+ - S^-$)

and

V+shift is the positive dc level shift voltage created by the resistor divider comprised of R+shift1 and R+shift2.

V+shift is used to position the zero pressure offset at the desired level.

Table 2 summarizes the 1% resistor values used to obtain a four-volt span with a zero pressure offset of 0.5 V (assuming the typical sensor offset and span values of 0.0 mV and 20 mV, respectively).

Table 2. Resistor Values for the MPX2100 Amplifier Design

R+shift1	R+shift2	R1	R2	R3	R4
4.99 kΩ	549 Ω	20.0 kΩ	100 Ω	100 Ω	20.0 kΩ

THE INTEGRATOR

As shown in Figure 1, the integrator consists of a single resistor and single capacitor. A programmable duty cycle pulse train from the microcontroller is input to the integrator. Assuming that the RC time constant of the integrator is sufficiently long compared to the pulse train's frequency, the resulting output which is input to the inverting terminal of the comparator is a dc voltage that is linearly proportional to the pulse train's duty cycle, i.e.:

$$\text{DC Output Voltage} = \text{Pulse Train's Duty Cycle (\%)} \times 5.0\ \text{V}$$

Where the Pulse Train's Duty Cycle is multiplied by the pulse train's logic-level one voltage value which is typically the same voltage as the microcontroller's 5.0 V supply.

Table 3 shows a few examples of Pulse Train Duty Cycles and the corresponding DC Output Voltage assuming a typical pulse train logic-level one value of 5.0 V.

Table 3. Example Pulse Train Duty Cycles and the Integrator's Corresponding dc Voltage Output

Pulse Train's Duty Cycle (%)	0	25	50	75	100
DC Output Voltage (V)	0	1.25	2.5	3.75	5

To establish a stable constant dc voltage at the integrator's output, its time constant must be sufficiently long compared to the frequency of the pulse train. However, the system resolution and thus performance are directly related to the pulse train's frequency. The design of the time constant and choice of the resistor and capacitor values is discussed in *System Design: Defining and designing for a desired signal resolution.*

COMPARATOR

The LM311 chip is designed specifically for use as a comparator and thus has short delay times, high slew rate, and an open-collector output. A pull-up resistor (R6 = 5 kΩ) at the output is all that is needed to obtain a rail-to-rail output. As Figure 1 illustrates, the pressure sensor's amplified output voltage is input to the non-inverting terminal of the op-amp and the integrator's dc output voltage is input to the inverting terminal. Therefore, when the pressure sensor's output voltage is greater than the integrator's dc output voltage, the comparator's output is high (logic-level one); conversely, when the pressure sensor's output voltage is less than the integrator's dc output voltage, the comparator's output is low (logic-level zero).

An optional resistor, RH is used as positive feedback around U2 in Figure 1 to provide a small amount of hysteresis to ensure a clean logic-level transition (prevents multiple transitions (squegging)) when the comparator's inputs are similar in value. The amount of hysteresis increases as the value of RH decreases. For this design, the value of RH is not critical but should be on the order of 100 kΩ.

THE MC68HC05P9 MICROCONTROLLER

The microcontroller for this application requires an output compare timer channel and one general I/O pin. The output compare pin is programmed to output the pulse train that is input to the integrator, and the general I/O pin is configured as an input to monitor the logic-level of the comparator's output. The remainder of this paper discusses the system and software requirements.

SYSTEM DESIGN: HOW THE SYSTEM WORKS

For any analog sensor voltage output, there's a pulse train with a duty cycle that when integrated will equal the sensor's output. Therefore, by incrementing via software the pulse train's duty cycle from 0% to 100%, there's a duty cycle that when integrated will be larger than the sensor's current voltage output. When the integrated pulse train voltage becomes larger than the sensor's output voltage, the comparator's output will change from a logic-level one to a logic-level zero. This logic-level, in turn, is monitored on the general I/O pin. The pulse train's duty cycle creating the integrated voltage that caused the comparator's logic-level transition is the digital representation of the sensor's voltage. Thus every sensor analog output voltage is mapped to a specific duty cycle. This design inherently has outstanding performance (very stable and accurate) since the digital representation of the sensor signal is created by the microcontroller's digital time base. Also the pressure measurement, made via software that first increments the pulse train's duty cycle and then determines if an edge transition occurred on the general I/O pin, is straightforward and easy.

In a calibration routine (discussed below) the sensor's output at two known pressures (e.g. zero and full-scale pressure) can be mapped to two corresponding pulse train duty cycles. Since the pressure sensor's output voltage is linear with the applied pressure, and the integrator's dc output voltage is linear with the input pulse train duty cycle, then the pulse train's duty cycle that causes the logic-level transition at the comparator's output will also be linear with the applied pressure. Thus by knowing the duty cycles for two known pressures, a linear interpolation of any duty cycle gives an accurate measurement of the current pressure. The following equation is used to interpolate the pressure measurement where the pressure units are in kPa.

$$\text{Current Pressure} = \frac{\text{Current Duty Cycle} - \text{Duty Cycle @ Zero Pressure}}{\text{Duty Cycle @ Full-Scale Pressure} - \text{Duty Cycle @ Zero Pressure}} \times \text{Full-Scale Pressure in kPa}$$

For example:

At zero pressure, if the pulse train's duty cycle required to cause a logic-level transition at the comparator's output is 25% and at full-scale pressure the pulse train's duty cycle is 75%, then the current pressure that corresponds to a duty cycle of 50% (required to obtain the logic-level one to logic-level zero transition at the comparator's output) is

$$\text{Current Pressure} = \frac{50\% - 25\%}{75\% - 25\%} \times 100\ \text{kPa} = 50\ \text{kPa}$$

Until now, the pulse train has been defined in terms of duty cycle. However, in practice duty cycle is calculated from the ratio of the high time to the total period of the pulse train. Therefore, there is a high time (typically in μs) of the pulse train that causes the logic-level transition of the comparator's output. The interpolation of the current pressure can then be calculated directly from the high time of the pulse train that is programmed by the user to be generated by the microcontroller's output compare pin. The equation is similar to the one above for Current Pressure:

$$\text{Current Pressure} = \frac{\text{Current High Time} - \text{High Time @ Zero Pressure}}{\text{High Time @ Full-Scale Pressure} - \text{High Time @ Zero Pressure}} \times \text{Full-Scale Pressure in kPa}$$

Via this equation, the digital nature of the design is revealed. The analog voltage signal has been translated into a signal in the time domain where the high time generated by the output compare pin is actually the digital time representation of the sensor's output. Since the user precisely controls the high time of the pulse train (and period) via software which is based on the accurate digital time base of the microcontroller, the digital representation of the signal is very stable and accurate. Additionally, the high accuracy of the digital representation is possible since all the user must do to digitize the signal is detect a single logic-level transition at the comparator's output.

SYSTEM DESIGN: DEFINING AND DESIGNING FOR A DESIRED SIGNAL RESOLUTION

The resolution is directly related to the period (and thus frequency) of the pulse train. In our design, the difference between the pulse train's high time at full scale pressure and the pulse train's high time and zero pressure must be 512 μs to obtain at least 8-bit resolution. This is determined by the fact that a 4.0 MHz crystal yields a 2.0 MHz clock speed in the MC68HC05P9 microcontroller. This, in turn, translates to 0.5 μs per clock tick. There are four clock cycles per timer count. This results in 2 μs per timer count. Thus, to obtain 256 timer counts (discrete high-time time intervals or 8-bit resolution), the difference between the zero pressure and full scale pressure high times must be at least 2 μs x 256 = 512 μs.

To determine the pulse train's maximum frequency (or minimum period), the sensor's analog dynamic range (span) must be known. For this design, the span is 4.0 V. Thus the 4.0 V span of the sensor must translate to 512 μs of time for 8-bit resolution. But the pulse train typically has a logic-level high value of 5.0 V, indicating that for a 100% duty cycle or a period with all high time, the integrator's output would be 5.0 V; likewise, for a duty cycle of 0% or a period with no high time, the output would be 0 V. Therefore, 512 μs accounts for only 4.0 V/5.0 V (80%) of the pulse train's total period. See Figure 2. To calculate the pulse train's total period, divide the 512 μs by 4/5 (0.8) to obtain the required minimum period for the pulse train of 640 μs. The reciprocal of this minimum period is the maximum frequency (1.56 kHz) of the pulse train to obtain at least 8-bit resolution.

To summarize:

The MC68HC05P9 runs off a 4.0 MHz crystal. The microcontroller internally divides this frequency by two to yield an internal clock speed of 2.0 MHz.

$$\frac{1}{2\text{ MHz}} => \frac{0.5\ \mu s}{\text{Clock Cycle}}$$

and

$$4\text{ Clock Cycles} = 1\text{ Timer Count}$$

Therefore,

$$\frac{4\text{ Clock Cycles}}{\text{Timer Count}} \times \frac{0.5\ \mu s}{\text{Clock Cycle}} = \frac{2\ \mu s}{\text{Timer Count}}$$

For 8-bit resolution,

$$\frac{2\ \mu s}{\text{Timer Count}} \times 256\text{ Timer Counts} = 512\ \mu s$$

which is the required minimum time into which the sensor’s 4.0 V span is translated.

To calculate the required period of the pulse train to yield the 0 to 5.0 V output (from 0% to 100% duty cycle based on the pulse train's logic-level high value of 5.0 V):

$$\text{Minimum Required Period} = \frac{512\ \mu s\text{ for a 4 V Sensor Span}}{4/5\text{ of Integrator's Output}} = 640\ \mu s$$

Translating this to frequency, the maximum pulse train frequency is thus:

$$\frac{1}{640\ \mu s} = 1.55\text{ KHz}$$

The above procedure can be implemented easily for other resolution requirements (i.e. a resolution of 1%, 2%, etc.).

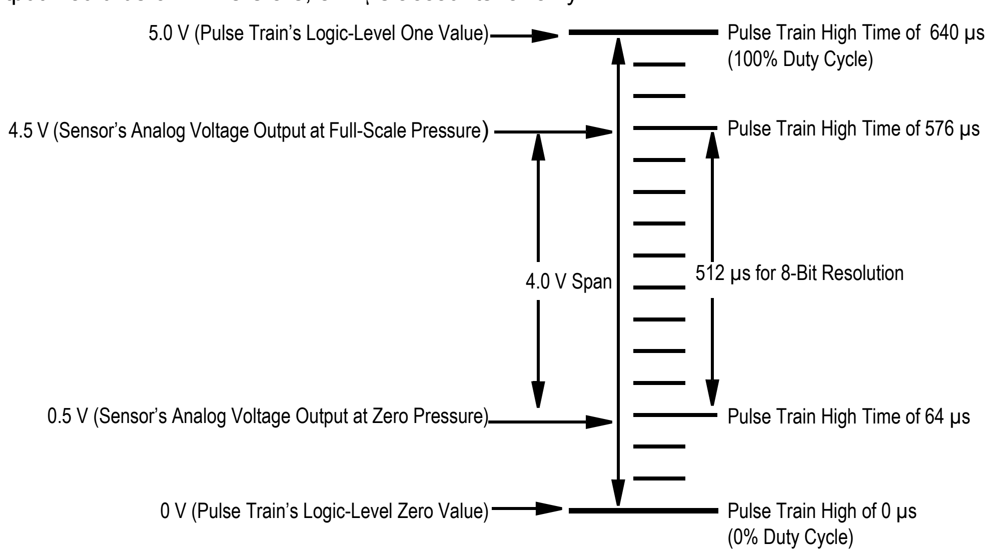

Figure 2. Designing the Pulse Train’s Period for 8-Bit Resolution

NOTE: Very small and very large high times (assuming a fixed period) are typically unattainable due to the finite amount of time it takes to generate the pulse train on the output compare pin. This amount of time will vary depending on the microcontroller's clock speed and the latency of the actual software routines implemented. Thus the sensor's analog voltage to which the integrator's dc voltage is compared must be within the possible ranges of voltages created by the integrator's input pulse train - i.e. the sensor's zero pressure offset voltage must be greater than the smallest voltage created by the integrator (corresponding to the pulse train's smallest possible high time) and the sensor's full scale output voltage must be less than the largest voltage created by the integrator (corresponding to the pulse train's largest possible high time).

After establishing the frequency of the pulse train, the RC time constant for the integrator can be determined and the resistor and capacitor value can be chosen. The RC time constant should be long compared to the period of the pulse train so that a stable dc voltage (very little ripple due to the capacitor's charging and discharging) is obtained at the output of the comparator.

Follow these steps to design the RC time constant and integrator's component values. The design example's calculations are presented simultaneously.

For the resolution desired, determine the number of volts (typically mV) that corresponds to the least significant bit (one timer count). For this design example, 8-bit resolution (256 timer counts) over the desired pressure sensor span corresponds to

$$\#\text{ of } \frac{\text{mV}}{\text{timer count}} = \frac{\text{Desired Pressure Sensor Span (V)}}{\text{Number of Timer Counts}}$$

$$= \frac{4.0\text{ V}}{256\text{ timer counts}} = \frac{15.6\text{ mV}}{\text{timer counts}}$$

Therefore, the stability of the integrator's output voltage should be less than 15.6 mV (least significant bit). Choosing an RC time constant that allows a ripple of approximately one-fourth of the least significant bit is sufficient (approximately 3.9 mV).

The most ripple occurs at a 50% duty cycle pulse train. For this design the entire period is 640 μs. 50% duty cycle indicates a high time (and low time) of 320 μs. Furthermore, the capacitor should discharge no more than approximately 3.9 mV (defined as ΔV) over the 320 μs. The following equation is used to calculate the value for RC:

$$V(t) = V_{INITIAL} - \Delta V =$$

$$\text{Pulse Train Logic– level one value} \times \text{Duty Cycle} \times e^{-\frac{t}{RC}}$$

where

$V_{INITIAL}$ = Pulse Train Logic–level one value × Duty Cycle

and

ΔV is the voltage discharge of the capacitor.

Solving for RC:

$$RC = \frac{t}{\ln\left(\dfrac{V(t)}{\text{Pulse Train Logic–level one value} \times \text{Duty Cycle}}\right)}$$

$$= \frac{320\ \mu\text{s}}{\ln\left(\dfrac{2.5\text{ V} - 3.9\text{ mV}}{5.0\text{ V} \times 50\%}\right)} = 0.205\text{ s}$$

Finally, choose the values of the resistor and capacitor. A typical resistor value is on the order of a tens of kΩ. The resistor's value can be higher (hundreds of kΩ) but care must be taken to avoid increased thermal noise.

For this design, the resistor value is chosen to be 49.9 kΩ (1% resistor). The capacitor's value is readily calculated to be

$$C = \frac{0.205\text{ s}}{49.9\text{ k}\Omega} = 4.1\ \mu\text{F}$$

Choose the values of the resistor and capacitor so that the actual time constant is equal to or greater than the calculated time constant.

NOTE: Be aware that temperature variations can create errors in the system (thus reducing system performance); therefore, be sure to use low temperature coefficient resistors, capacitors, etc.

SYSTEM DESIGN: STEP-BY-STEP PROCEDURE FOR PRESSURE MEASUREMENT AND CALIBRATION

To measure pressure (Note: there are other measurement algorithms that can be performed that in some cases may be more acceptable [see below, Additional notes]):

1. Start with a pulse train with the minimum high time feasible with the system's microcontroller. Pulse train should run at a frequency equal to or less than the frequency calculated above.
2. Make sure the general I/O pin's input is high (sensor's output voltage is greater than the integrator's output voltage).
3. Increment the high time of the pulse train by one timer count.
4. Check the general I/O pin to see if its input is low (sensor's output voltage has become less than the integrator's output voltage).
5. If the general I/O pin is reading a logic-level zero, store in memory the high time of the pulse train as the current pressure high time reading that created the logic-level transition in the comparator's output.
6. If the general I/O pin is reading a logic-level one, go back to step 3 and repeat.
7. Using the equation "Current Pressure = ..." shown above, calculate the current pressure (assuming the system has already been calibrated).
8. Repeat steps 1 through 7 for additional pressure measurements.

To calibrate the system:

At zero and full scale pressures, perform the above 8 step pressure measurement routine. Store the appropriate pulse train high times corresponding to zero and full scale pressure. These high times will be used to calculate the current pressure as mentioned in Step 7 above.

SOFTWARE EXAMPLES TO GENERATE PULSE TRAIN ON OUTPUT COMPARE TIMER CHANNEL

The following software examples are written in assembly language for the MC68HC05P9 (the code is applicable to any HC05 series microcontroller with TCMP pin).

```
* GENERATES THE PULSE TRAIN ON TCMP
GEN
LDA  PERIODL                  * LOW BYTE OF THE PERIOD
SUB  HIGHTIMEL                * LOW BYTE OF THE HIGHTIME
STA  LOWTIMEL                 * LOW BYTE OF THE LOWTIME
LDA  PERIODH                  * HIGH BYTE OF THE PERIOD
SBC  HIGHTIMEH                * HIGH BYTE OF THE HIGHTIME
STA  LOWTIMEH                 * HIGH BYTE OF THE LOWTIME
RTS

* INCREASE THE HIGH TIME (DUTY CYCLE) OF THE PULSE TRAIN
INCPW
LDA  HIGHTIMEL
ADD  #$01            * INCREMENT PULSE WIDTH BY 2 µs
STA  HIGHTIMEL
LDA  HIGHTIMEH
ADC  #$0
STA  HIGHTIMEH
RTS

* DECREASE THE HIGH TIME (DUTY CYCLE) OF THE PULSE TRAIN
DECPW
LDA  HIGHTIMEL
SUB  #$01            * DECREMENT PULSE WIDTH BY 2 µs
STA  HIGHTIMEL
LDA  HIGHTIMEH
SBC  #$0
STA  HIGHTIMEH
JSR  GEN
RTS

* INCREASE THE PERIOD (DECREASE FREQUENCY) OF THE PULSE TRAIN
INCPER
LDA  PERIODL
ADD  #$05            * INCREMENT PERIOD BY 10 µs
STA  PERIODL
LDA  PERIODH
ADC  #$0             * ADJUST HIGH BYTE OF PERIOD IF CARRY
STA  PERIODH
JSR  GEN
RTS

* DECREASE THE PERIOD (INCREASE FREQUENCY) OF THE PULSE TRAIN
DECPER
LDA  PERIODL
SUB  #$05            * DECREMENT PERIOD BY 10 µs
STA  PERIODL
LDA  PERIODH
SBC  #$0             * ADJUST HIGH BYTE OF PERIOD IF BORROW
STA  PERIODH
JSR  GEN
RTS

TIMER                         * INTERRUPT SERVICE ROUTINE FOR TCMP
LDA  TSR             * CLEAR OCF FLAG IN TSR
LDA  TCMPL

BRSET 0,TCR,ADDHIGH* HIGH OR LOW PULSE TIME NEEDED?

ADDLOW
BSET 0,TCR           * ADD LOW TIME TO THE PULSE TRAIN
LDA  LOWTIMEL
ADD  TCMPL
TAX
LDA  TCMPH
ADC  LOWTIMEH
STA  TCMPH
```

```
STX  TCMPL
RTI

ADDHIGH
BCLR 0,TCR            * ADD HIGH TIME TO THE PULSE TRAIN
LDA  HIGHTIMEL
ADD  TCMPL
TAX
LDA  TCMPH ADC        HIGHTIMEH
STA  TCMPH
STX  TCMPL
RTI
```

ADDITIONAL NOTES

This type of A/D conversion method (one type of A/D conversion) inherently takes a finite period of time to digitize the signal (incrementing the pulse train's high time while polling the general I/O pin); however, for most sensor applications the physical phenomenon being measured does not change quickly (<1 ms) enough to warrant an ultra-fast A/D conversion process.

An additional advantage of this design is that the measurement process may be performed only as necessary, keeping the CPU processing time and overhead minimal.

If an input capture timer channel (TCAP) is available, it may be configured to detect the logic-level one to logic-level zero transition of the comparator's output. When the edge transition occurs, an interrupt service routine is executed that stores the pulse train's high times, calculates the current pressure, etc. This is typically more convenient and eliminates the need to poll a general I/O pin every time the pulse train's high time is incremented (interrupt subroutine is executed only when the edge transition occurs).

SUMMARY

Shown above is a minimal component design that can convert an analog sensor's output into a digital output. Each major subsystem (sensor, amplifier, integrator, comparator, and microcontroller) is explained in detail simultaneously with a design example. Next the system operation is discussed including how it works and how to design a desired system resolution. Finally a flow chart for measuring and calibrating the sensor's output is presented.

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2006. All rights reserved.

AN1586
Rev.2
11/2006