Application Note

AN1954/D
Rev 2, 05/2003

MPXY8000 Series Tire Pressure Monitoring Sensor

MOTOROLA intelligence everywhere™ | digital dna™

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

by Ador Reodique

Freescale Semiconductor, Inc.

The MPXY8000 Series Tire Pressure Monitoring Sensor

The MPXY8000 Series sensor is a fully integrated sensor targeted specifically for tire pressure monitoring systems (TPMS). The sensor includes a capacitive absolute pressure sensor with a range from 0 kPa to 637.5 kPa and a temperature sensor capable of measuring temperature in the range of -40°C to 125°C. A custom transfer function is available upon request.

The sensor leaves the factory fully trimmed and signal conditioned. Its digital interface uses the SPI (serial peripheral interface) protocol and is designed to work with low-voltage Motorola microcontrollers such as the 68HC05 and 68HC08.

This sensor is optimized for low-voltage and low-power draw suitable for powering with a single 3-volt lithium battery.

This sensor is also designed to be media compatible with fluids and media that are commonly found within the tire environment.

Definitions

The following terms used within this application note are defined below.

DAR - Digital to analog converter register. The DAR is an 8-bit serial register that holds the DAC (digital to analog converter) value.

TPMS - Tire pressure monitoring system. A complete system includes a transmitter module and a receiver module. A transmitter module consists of a pressure sensor, a temperature sensor, a controller (a state machine or a microcontroller), an RF transmitter, and a battery power supply.

© Motorola, Inc., 2003

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

SPI - Serial peripheral interface (usually pronounced *spy*). SPI is a standard protocol and a synchronous serial interface that includes a clock line and a data line. The MPXY8000 Series sensor expects that the 8-bit data is transferred serially on the rising edge of the clock and that the data and clock lines idle in their low states.

LFO - Low frequency oscillator. The MPXY8000 Series sensor uses a free-running, low-frequency, low-power oscillator that runs typically at 5.4 kHz. This clock is divided down to serve as either a wake-up or a reset pulse.

Wake Up - A term used to wake up a microcontroller that has been placed in a low-power stop mode. An active low wake-up pulse occurs approximately every 3 seconds at the OUT pin of the MPXY8000 Series sensor.

Interface Description and Programming Models

This section explains the I/O interface of the MPXY8000 Series TPMS sensor, its modes of operation (Standby/Reset, Measure Pressure, Measure Temperature, and Output Read), its serial interface, and how to acquire and convert data.

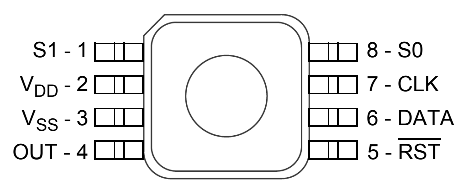


Figure 1 8-Pin Super Small Outline Package (SSOP)

Figure 2 shows the I/O of the MPXY8000 Series sensor as it is interfaced to a microcontroller unit.

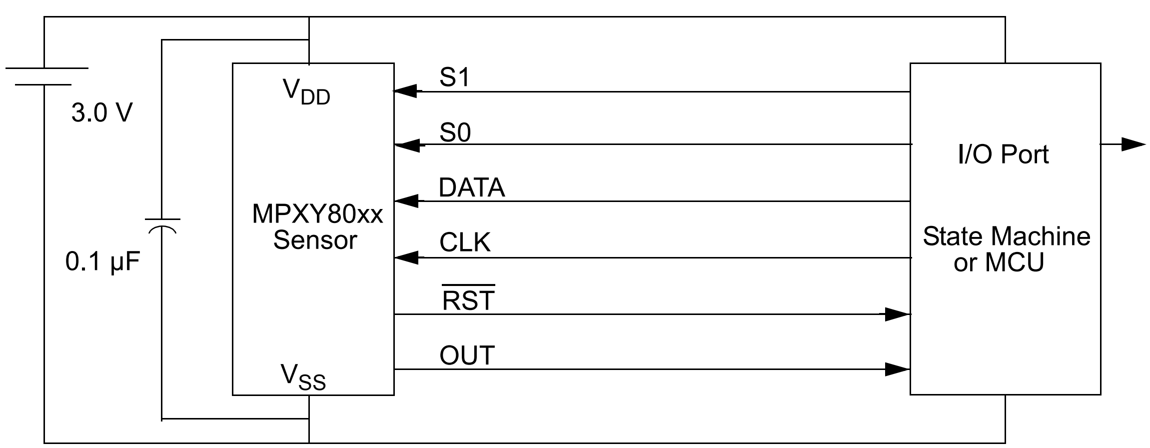


Figure 2 The MPXY8000 Series Sensor-MCU Interface

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Table 1 MPXY8000 Series Sensor Pin Description

Pin Number	Pin Name	Pin Function	Direction
1	S1	Mode Select	Input
2	V_{DD}	Positive Power	Power
3	V_{SS}	Ground	Power
4	OUT	Comparator or Wake-Up Output (active low)	Output
5	$\overline{RST}$	MCU Reset (active low)	Output
6	DATA	Data	Input
7	CLK	Data Clock	Input
8	S0	Mode Select	Input

Interface Description

The MPXY8000 Series sensor has six I/O pins and two power supply pins (V_{DD} and V_{SS}). The sensor is interfaced to a microcontroller as shown in **Figure 2** on page 2.

There are four input pins and two output pins on the MPXY8000 Series sensor. The input pins are S0, S1, DATA, and CLK. The two output pins are OUT and $\overline{RST}$.

The S1/V_{PP} and S0 input pins place the MPXY8000 Series sensor in any of four modes: Standby/Reset, Measure Pressure, Measure Temperature, and Output Read (see **Mode Control Pins S0 and S1** on page 7).

The DATA and CLK pins are the serial interface to the MPXY8000 Series sensor's DAR. Depending on the mode selected, the OUT pin is the comparator output pin in either measure mode and also becomes the wake-up pin in the Standby/Reset mode. The OUT pin is discussed in the following sections.

The $\overline{RST}$ pin is an output pin that serves as a periodic low-pulse reset (approximately every 52 minutes) to a microcontroller reset pin.

The interface is discussed in the following sections.

V_{DD} and V_{SS} Power Supply Pins

Bypass the V_{DD} and V_{SS} pins to decouple power supply noise caused by digital switching currents and parasitic inductances. This noise may be coupled into the sampling circuitry and cause inaccuracies in readings. A surface mount 0.1 μF mica or ceramic capacitor mounted close to the V_{DD} and V_{SS} pins is preferred.

Serial Interface CLK and DATA Pins

Serial communication between the MPXY8000 Series sensor and the controller is achieved through the CLK and DATA pins. The serial interface is compatible with the SPI found in many Motorola MCUs.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

The SPI loads data into the DAR of the MPXY8000 Series sensor. The SPI interface and DAR are active during Standby/Reset and Output Read modes (refer to the following for more information about the different modes of operation).

The CLK pin is designed to operate at a maximum of 1 MHz. To avoid conversion errors, the minimum clock period multiplied by 64 should not be longer than the specified sample capacitor discharge time when performing successive approximations (for example, $64/f_{CLK} < 20$ ms).

When utilizing SPI, the MCU becomes the master and the MPXY8000 Series sensor is the slave so that data transmission is only from the controller to the sensor.

Before data is clocked into the DAR, the SPI of the MPXY8000 Series sensor must be synchronized with the data to be clocked in by the MCU or controller. Refer to **Synchronizing the Data to be Clocked into the DAR** on page 6 for information about synchronization.

DATA is clocked into the serial register on the rising edge of the CLK pin starting with the MSB. At the eighth falling edge of the CLK pin, the 8-bit data in the serial register is transferred into the DAR as shown in **Figure 3**.

Because data is transferred on the rising edges of the CLK signal, it is important that the CLK signal from the MCU is idled in the low state to avoid shifting in undesired data.

NOTE: Make sure the CLK pin from the MCU/controller idles in the low state.

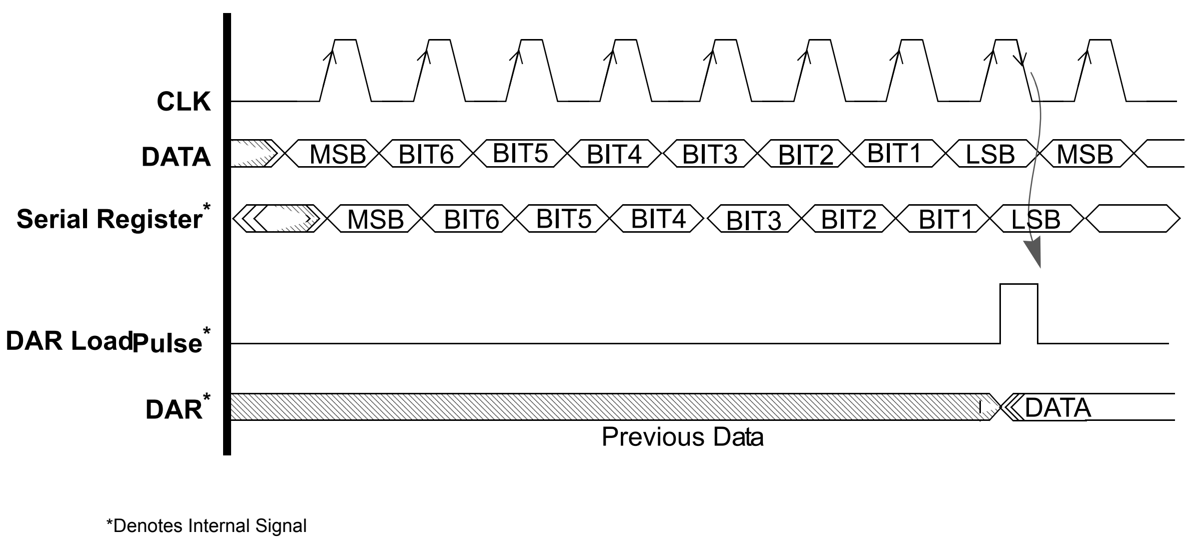


Figure 3 Serial DATA and DAC Register Timing

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

On MCUs without a SPI, use software routines to emulate the SPI protocol. This is easily accomplished using general purpose I/O pins. **Figure 4** displays a flowchart for a software SPI emulator that functions on HC05 and HC08 MCUs.

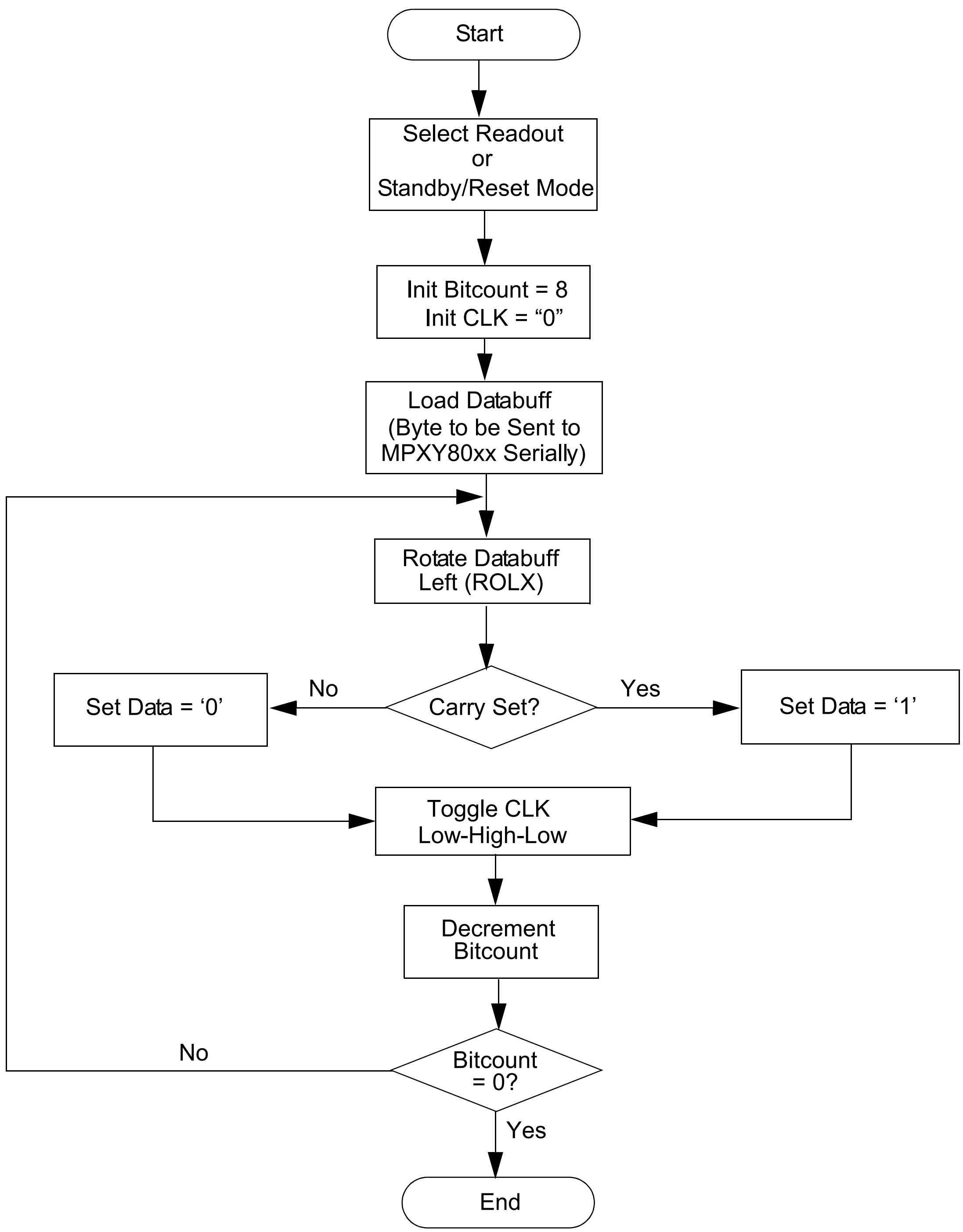


Figure 4 Software Serial Communication (SPI) with the MPXY8000 Series Sensor

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

The following is a 68HC08 software example.

```
*****************************************************************
Software serial interface (SPI) routine to send DATA MSB first
Called with:   mov #XX,DATABUFF  -- DATABUFF contains 8-bit word to be sent
               jsr  SEND_BYTE
Returns:       none
Variables:
BITCOUNT = 8-bit counter
SDATA = DATA pin connected to PORTA
DCLK = CLK pin connected to PORTA
*****************************************************************

Send_Byte:      PSHX                            ; Store the XR.
                BCLR    DCLK,PORTA              ; Clock is initially low.
                MOV     #$08,BITCOUNT           ; Initialize number of bits to be sent (1 byte).
                LDX     DATABUFF                ; DATABUFF contains the byte to be sent serially
                                                  to the MPXY8000 Series sensor.
Send_Nextbit:   ROLX                            ; Rotate x left and watch the Carry bit.
                BCC     Zero                    ; Send a '1' if Carry set else send a '0.'
One:            BSET    SDATA,PORTA             ; SDATA = '1.'
                BRA     Toggle_Clock

Zero:           BCLR    SDATA,PORTA             ; SDATA = '0.'

Toggle_Clock:   BSET    DCLK,PORTA              ; Shift SDATA state into DAR by toggling clock.
                BCLR    DCLK,PORTA              ;
                DBNZ    BITCOUNT,Send_Nextbit   ; Determine if there are more bits to be sent.
                PULX                            ; Restore the XR.
                RTS                             ; Return.
```

Figure 5 Software Example 1 - Software SPI

Synchronizing the Data to be Clocked into the DAR

It is important to ensure that the MPXY8000 Series sensor's internal SPI bit counter is synchronized with the data that is shifted in. During power up, this bit counter defaults to an unknown state.

The bit counter resets to zero when entering into either the Measure Pressure or Measure Temperature modes. This provides a convenient method for providing one hundred percent assurance that the data shifted in is always synchronized with the DAR bit counter. The program flow for ensuring data synchronization is shown in **Figure 6** on page 7.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

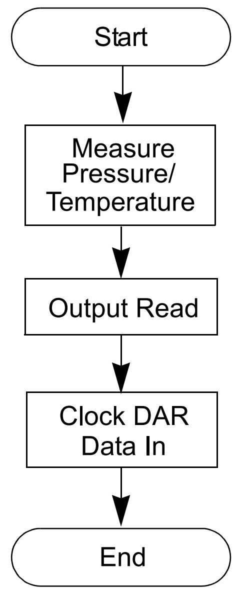


Figure 6 Software Flowchart for Resetting the Internal Bit Counter

Mode Control Pins S0 and S1

Mode control is accomplished by two pins, S0 and S1. The MPXY8000 Series sensor's modes of operation are summarized in the following table.

Table 2 The MPXY8000 Series Sensor Modes of Operation

S1	S0	Mode
0	0	Standby/Reset (Idle)
0	1	Measure Pressure
1	0	Measure Temperature
1	1	Output Read

Standby/Reset Mode

When in Standby/Reset mode (also referred to as Idle mode), the MPXY8000 Series sensor is at its lowest power consumption. In this mode, the analog sections of the sensor are off and only the low frequency oscillator (LFO), SPI, DAR, wake-up pulse, and reset pulse dividers are on.

In this mode, the controller may be able to shift threshold data into the DAR register. This function is discussed later in the document.

During Standby mode, the MPXY8000 series sends a 370 µs wake-up pulse at its OUT pin at approximately 3-second intervals. This negative pulse functions as a wake-up signal to a microcontroller that is in a stop mode. The microcontroller must place the MPXY8000 Series sensor in Standby mode *before* it places itself in stop mode.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

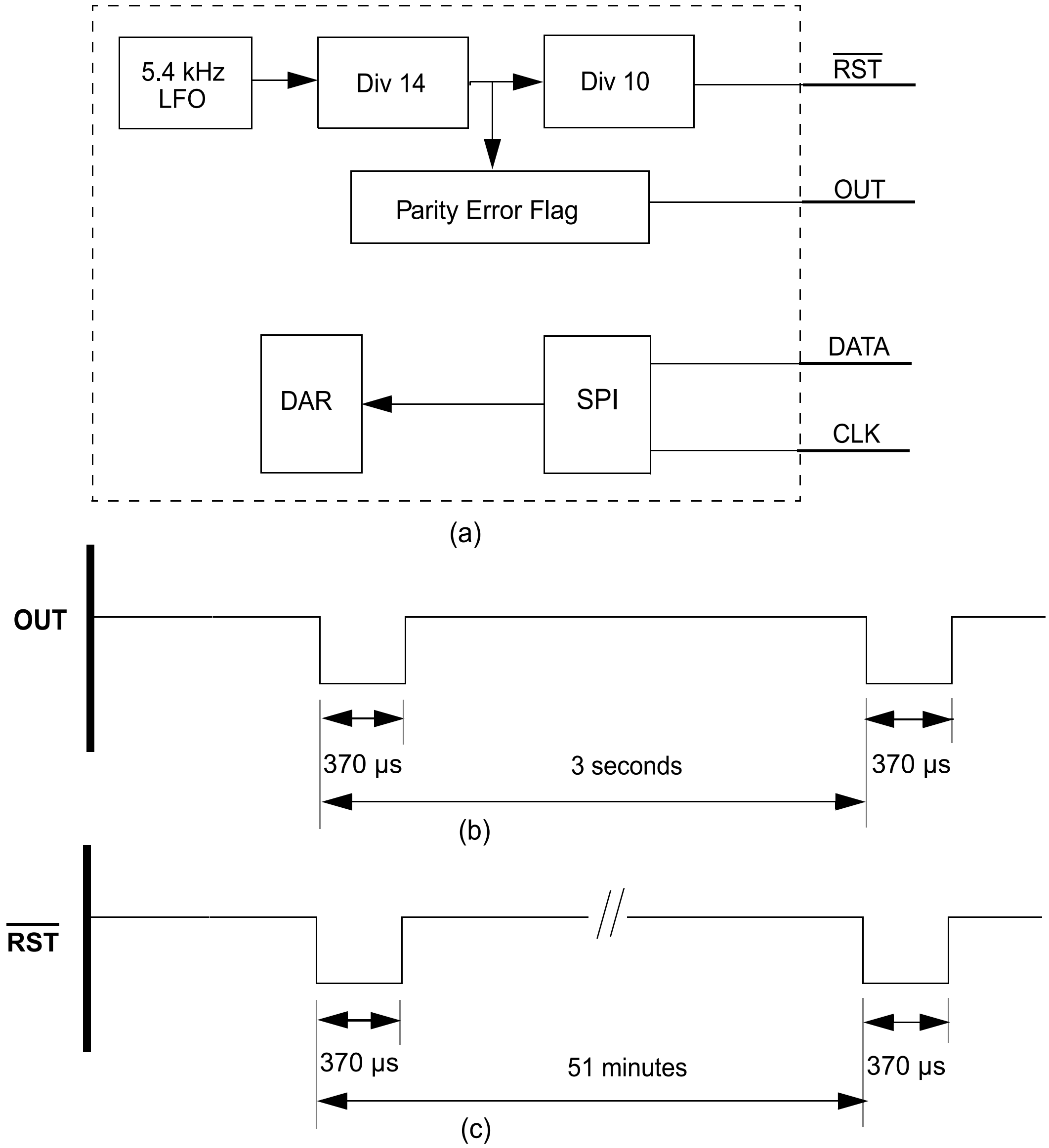


Figure 7 LFO Divider Behavior

(a) Configuration during Standby Mode (b) Resulting Wake-Up Waveform (c) Reset Waveform

Measure Pressure Mode In this mode, the sensor multiplexer connects the output of the pressure sensor to the sampling capacitor. The duration of this mode should be at least 500 µs to allow the pressure sensor switched capacitor circuit to turn on and settle. In a program sequence, a delay of at least 500 µs should be inserted after placing the MPXY8000 Series sensor in this mode to avoid sampling the pressure sensor output while the circuitry is still unstable. Failure to include a sufficient delay can cause an erroneous reading.

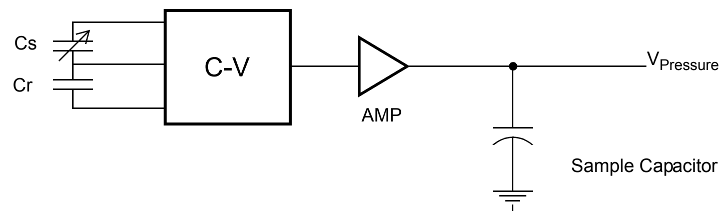
Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

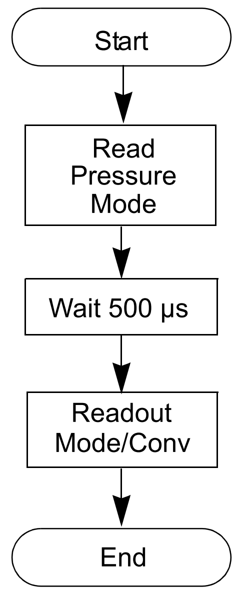
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

When entering this mode the bit counter is reset, which allows it to be synchronized with the external clock at the CLK pin.

The recommended method to read the pressure sensor output is shown in **Figure 8**.



(a)



(b)

Figure 8 Measure Pressure Mode

(a) Internal Configuration (b) Pressure Acquisition Software Flow

Measure Temperature Mode

In this mode, the sensor multiplexer connects the output of the temperature sensor to the sampling capacitor. Allow the circuit to settle for at least 200 µs after initiating this mode.

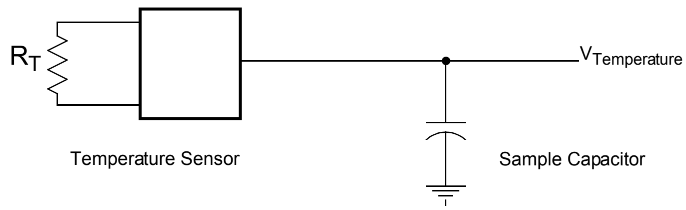
When entering this mode the bit counter is reset, which allows it to be synchronized with the external clock at the CLK pin.

Refer to **Figure 9** for the recommended way to read the temperature sensor output.

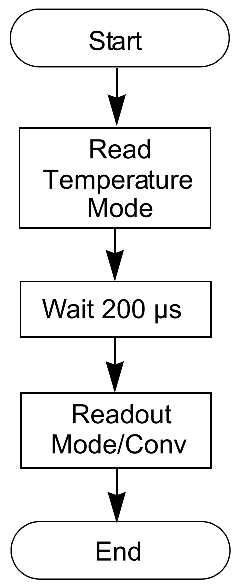
For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005



(a)



(b)

Figure 9 Measure Temperature Mode

(a) Internal Configuration (b) Temperature Acquisition Software Flow

Output Read Mode

In this mode, the output multiplexer connects the comparator output to the OUT pin as shown in **Figure 10** on page 11. After sampling the pressure or temperature sensor, initiate this mode and immediately sample the OUT pin or perform your successive approximation routine before the sensor voltage in the sampling capacitor discharges.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

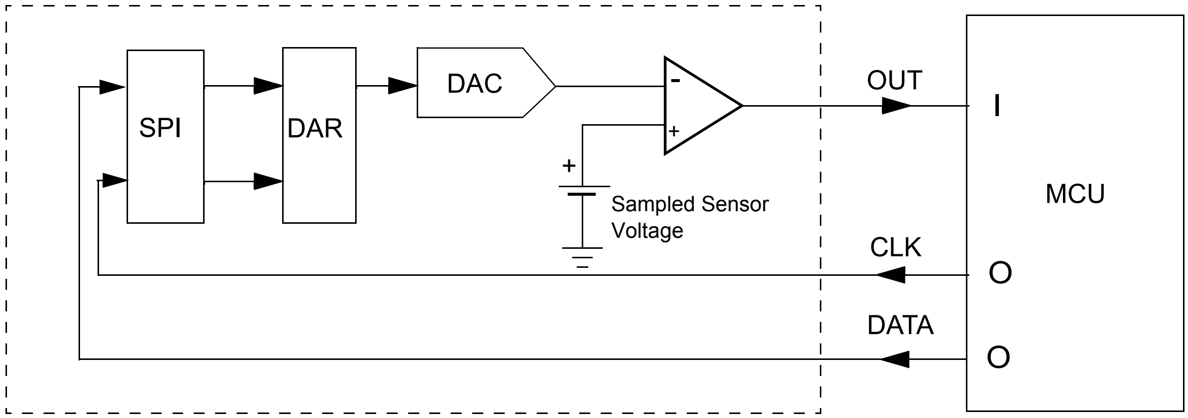


Figure 10 Configuration in Output Read Mode

In this mode, the sampled sensor voltage may be converted to an 8-bit value or a threshold check may be initiated. Each of these functions is discussed in the following sections.

OUT Pin

The OUT pin serves three different functions depending on the mode set by S1 and S0 pins. The following describes each of these modes.

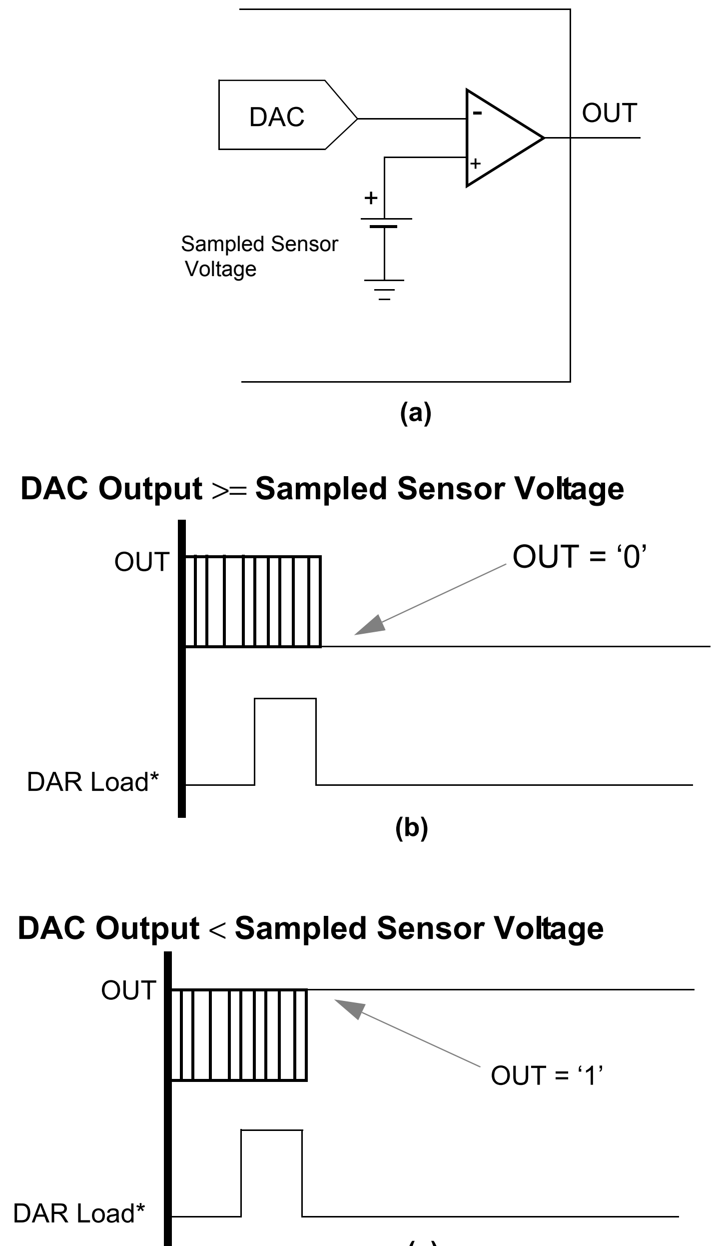
The OUT Pin During Output Read Mode

In Output Read mode, the OUT pin is connected to the analog comparator for sensor acquisition. The state of the OUT pin in this mode is high when the DAC output is lower than the sampled value in the sampling capacitor. If the DAC output is higher than the sampled value in the sampling capacitor, the OUT pin is low. Refer to **Figure 11** on page 12.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005



*DAR Load is the DAC write pulse after 8 bits have been clocked in.

Figure 11 Output Comparator Behavior during Output Read Mode

(a) OUT pin configuration during Output Read mode
(b) Shifted DAC value is greater than or equal to the sampled sensor voltage
(c) Shifted DAC value is less than the sampled sensor voltage

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Out Pin During Standby/Reset Mode

During Standby/Reset mode, the output multiplexer connects the OUT pin to an active low wake-up pulse signal. This pulse can be used to wake up an MCU in stop mode. This low pulse is generated from the internal LFO and the divider chain.

With a nominal frequency of the internal low frequency oscillator of 5.4 kHz, the wake-up pulse occurs approximately every 3 seconds and is twice the period of the low frequency oscillator, or 370 µs.

The OUT pin can be connected to an IRQ pin or a port I/O pin that supports wake up from idle or stop modes caused by a falling edge on its input.

Before initiating Standby mode on the MCU, configure the IRQ or port I/O pin as an interrupt pin to wake up the MCU on a falling edge. For example, a 68HC90RF2 can be programmed to wake up from either a keyboard interrupt or an IRQ interrupt. After this occurs, the MCU should place the MPXY8000 Series sensor in Standby mode. A wake-up pulse from the MPXY8000 Series sensor wakes up the MCU from its stop mode. The wake-up sequence is shown in **Figure 12** on page 13.

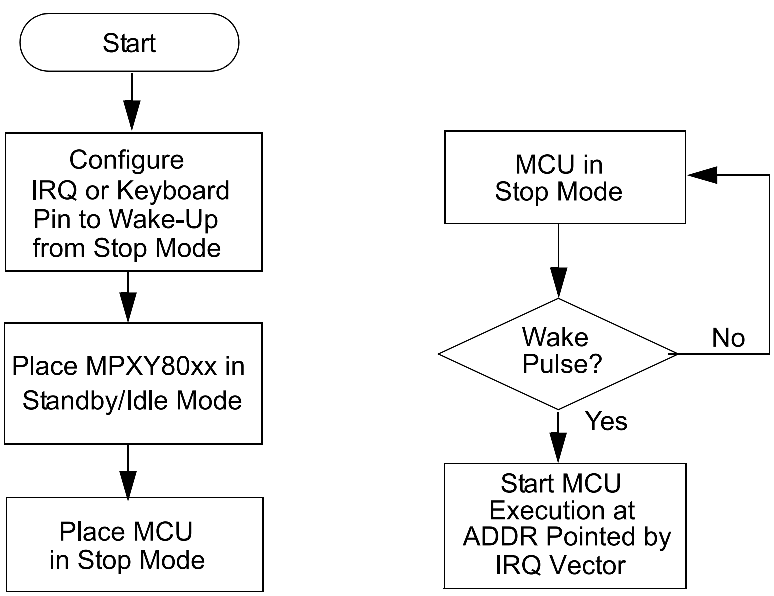


Figure 12 Configuring the MPXY8000 Series Sensor to Wake Up an MCU from its Stop Mode

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

RSTb (Reset Pulse) Output Pin

All good embedded systems such as a TPMS include a watchdog timer. Normally, a watchdog checks to see if the system has crashed and applies a reset if it has. This system is unique for its harsh environment and its cost limitations, so the MPXY8000 Series sensor provides a fail-safe mechanism in the form of a reset signal (approximately every 52 minutes) regardless of the operating mode of the sensor. The reset pulse can occur in any of the four operating modes.

If this signal is utilized in your system, then data such as running averages of pressure should be calculated and stored at the receiver end because the periodic reset would destroy any data stored in RAM.

Acquiring Pressure or Temperature Sensor Data

There are two ways to acquire sensor data from the MPXY8000 Series sensor: successive approximation and threshold check. Successive approximation provides accurate conversion of the sampled temperature or pressure reading into an 8-bit value.

In a threshold check, the DAR is preloaded with a threshold value during Standby/Reset mode to detect whether the pressure or temperature has crossed a particular level. These two methods are described in detail in the following sections.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Successive Approximation

When performing analog to digital conversion, the MCU is used as a SAR (successive approximation routine) controller. The 8-bit DAC data is serially loaded by the MCU for each of the eight guesses (as illustrated in **Figure 2** on page 2). Therefore, a complete successive approximation conversion is completed in eight guesses with eight CLK cycles per guess, or 64 CLK cycles.

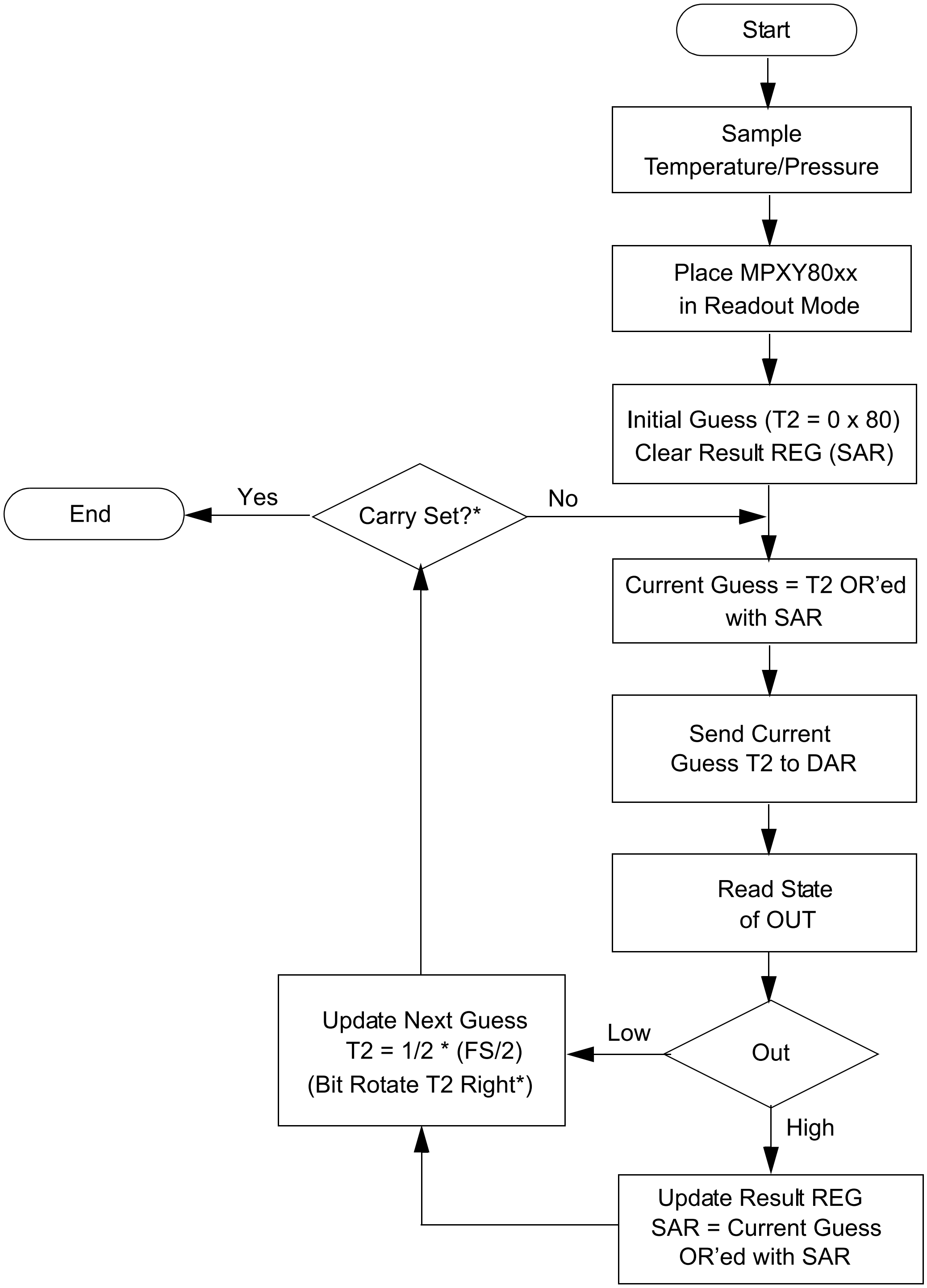
To perform a conversion, a half-scale (0xFF/2~0x80) guess byte is serially loaded into the DAR and the OUT pin is monitored by the MCU thereafter. When the guess is loaded and the state of the OUT pin is low, the guess in the DAR is too high or the guess matches the value in the sample capacitor. If the OUT pin remains high after a guess is loaded into the DAR, then the guess is too low.

The conversion result register (also known as SAR) defined as a variable by the MCU is updated accordingly; if the guess is too low, the MSB bit is set high in the variable SAR. If the guess is too high, the MSB is cleared. The process is repeated for the next bit. The next guess value to be sent to the DAR is the half-scale value of the last guess logically OR’ed with the current SAR value. The process of monitoring the OUT pin and setting or clearing the next bit in the SAR is performed until all eight guesses (8 bits) are determined.

Figure 13 on page 15 is a flowchart for performing successive approximation on the MPXY8000 Series sensor.

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005



* The carry bit is used as a sentinel bit. When ROR is performed on T2 in HC05/08 MCUs, the carry bit is set when all eight bit positions have shifted to the right.

Figure 13 Successive Approximation Flowchart for the MPXY8000 Series Sensor

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

```
*****************************************
* MPXY8000 Series Serial SAR routine (SSAR)
*
* Called with:    jsr SESAR
* Returns:  8 bit converted value in SAR variable
* Calls          Send_Byte
* Variables:
* T2 = 8-bit temporary variable
* SAR = Result variable
*
* OUTB pin is connected to PORTA3
*****************************************
SerSAR:         PSHA                    ; Save accumulator
Init_DAC:       CLR      SAR            ; SAR will contain the result of the conversion.
                MOV      #$80,T2        ; Initial DAC weight $80 ~ 1/2 FS.
                                        ;
Conv_Wait:      LDA      T2             ;
                ORA      SAR            ; Place guess in SAR, the result register.
                STA      DATABUFF       ; DATABUFF is 8-bit data to be sent serially to
                BSR      Send_Byte      ; the MPXY8000 Series DAR. This is the guess value.

Read_DOUT_LEVEL                         ; Now determine if guess is too high or too low
                                        ;    by reading the MPXY8000 Series DOUT level.
                LDA      PORTA            This is done by reading PORTA and isolating
                AND      #%00001000     ; DOUT_BAR (PDA3). Compare this to 0: if equal,
                BEQ      TOO_HIGH       ;    then guess is too high (DOUT_BAR toggled low).

                LDA      SAR            ; If DOUT did not toggle low, then guess is too low
                ORA      T2             ; so we add this value to SAR by OR'ing it with T2
                STA      SAR            ; This is the same as setting the current bit weight
                                        ;    to '1'.
Too_High:       ROR      T2             ; If DOUT did not toggle low, then guess is low.
                                        ; In this case, try the next bit weight until
                BCC      CONV_WAIT      ;    the carry bit is set.
Conv_Ready:     PULA                    ; SAR contains the converted value.
                RTS                     ; Restore accumulator and Return.
```

Figure 14 Software Example 2: Serial Successive Approximation Routine

Threshold Check

When in Output Read mode, the MCU can initiate a threshold check on the MPXY8000 Series sensor to estimate whether the pressure or temperature is above or below a particular level without having to perform a full successive approximation routine.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

Separate temperature and pressure thresholds can be set or several thresholds may be set to check for a range of pressure or temperatures.

Figure 15 shows settings for high, mid, and low threshold to monitor a pressure or temperature range.

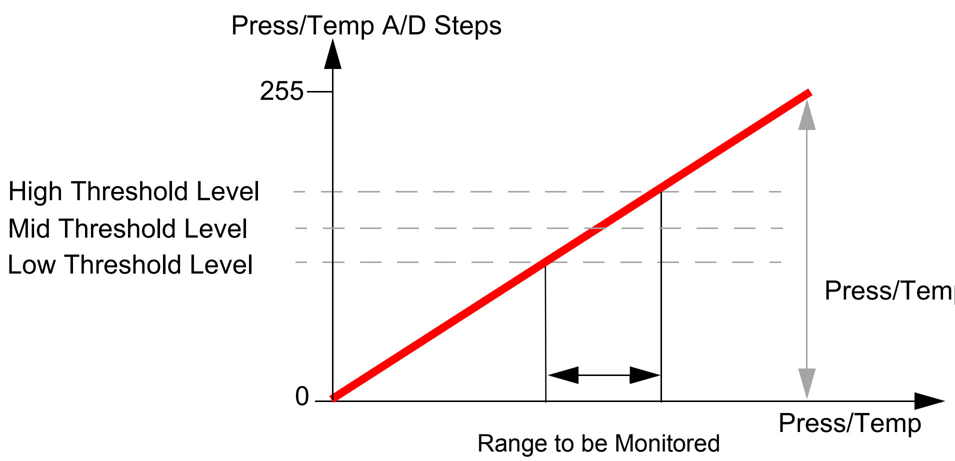


Figure 15 Illustration of a Threshold Check

When performing a threshold check, the DAR can be preloaded with a threshold value during Standby/Reset mode. The sensor output (temperature or pressure) is then appropriately sampled. When the state of the OUT pin is read in the Output Read mode, the output of the sensor can then be compared with the threshold value. A flowchart illustrating the sequence for Threshold Check is shown in **Figure 16** on page 18.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

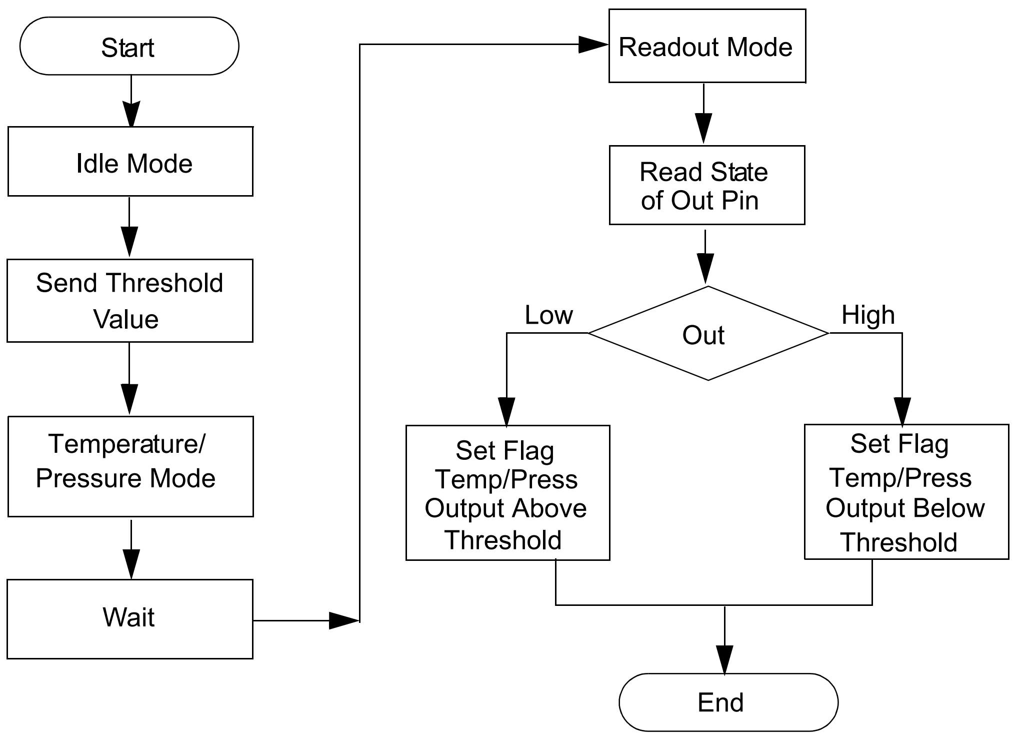


Figure 16 Threshold Check Flowchart

Sample MPXY8000 Series Sensor TPMS Application Circuit

Figure 17 on page 19 displays an example circuit using part number MC68HC908RF2. As shown, this circuit transmits a 315 MHz signal using frequency shift keying (FSK).

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

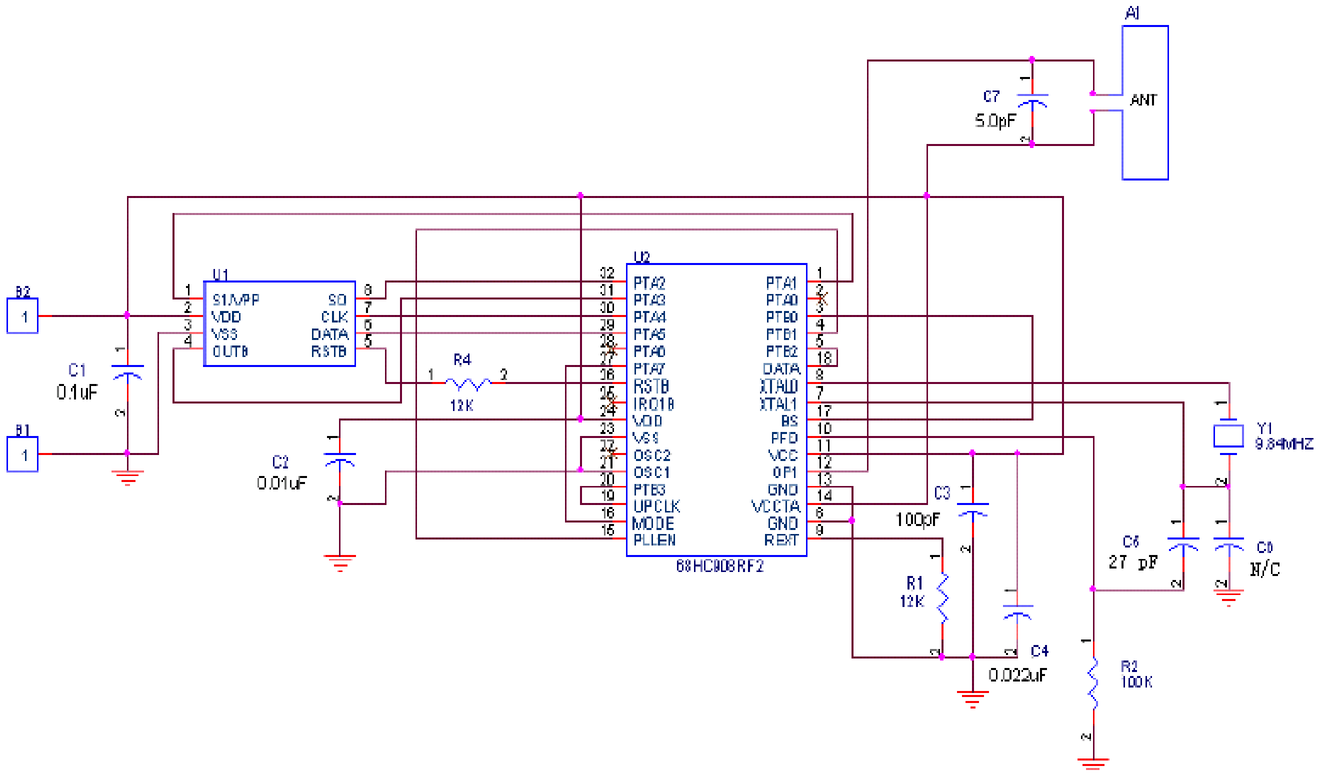


Figure 17 TPMS Application Schematic

Conclusion

A complete explanation of the I/O interface of the MPXY8000 Series sensor is explained. The four modes of operation have been discussed. The SPI interface has been discussed and a software example of SPI is given.

Successive approximation has been explained and a software example given for acquiring temperature and pressure using successive approximation. Finally, a very energy-efficient threshold check algorithm has been explained and software given to perform the threshold check.

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

For More Information On This Product,
Go to: www.freescale.com

HOW TO REACH US:

ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

USA/EUROPE/LOCATIONS NOT LISTED:
Motorola Literature Distribution
P.O. Box 5405
Denver, Colorado 80217
1-800-521-6274 or 480-768-2130

JAPAN:
Motorola Japan Ltd.
SPS, Technical Information Center
3-20-1, Minami-Azabu, Minato-ku
Tokyo 106-8573, Japan
81-3-3440-3569

ASIA/PACIFIC:
Motorola Semiconductors H.K. Ltd.
Silicon Harbour Centre
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
852-26668334

HOME PAGE:
http://motorola.com/semiconductors

Freescale Semiconductor, Inc.
ARCHIVED BY FREESCALE SEMICONDUCTOR, INC. 2005

MOTOROLA

Information in this document is provided solely to enable system and software implementers to use Motorola 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.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola 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 Motorola 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. Motorola does not convey any license under its patent rights nor the rights of others. Motorola 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 Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola 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 Motorola was negligent regarding the design or manufacture of the part.

MOTOROLA and the Stylized M Logo are registered in the US Patent and Trademark Office. All other product or service names are the property of their respective owners. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

© Motorola Inc. 2003

AN1954/D
Rev. 2
05/2003