**Pressure Sensors**

There are several types of pressure sensors in the market such as mechanical pressure sensors, semiconductor-based and MEMS-based devices. The latter two types are characterized with high sensitivity and long-term repeatability. Their output signals can be in various formats such as analog voltage output and serial (SPI or IIC) output.

A semiconductor type temperature sensor with analog voltage output will be discussed in this subsection. The particular temperature sensor is Freescale’s MPXA6115A. This is an integrated silicon pressure sensor for measuring absolute pressure. This sensor is on-chip signal conditioned, temperature compensated and calibrated.

This section shows how to connect the pressure sensor to the CSM-12C32 module and provide the C-codes for initializing the on-chip analog-to-digital converter for capturing the analog output voltage from the temperature sensor. Pictures, ordering information and web link of the datasheet are shown below.

![MPXA6115AC7U CASE 482C-03](source: MPXA6115A datasheet)

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Part Number</th>
<th>Weblink for the part</th>
<th>Description</th>
<th>Unit Price</th>
</tr>
</thead>
</table>
Hardware interfacing to the Freescale 9S12C32 MCU

There are eight pins in the package shown above but five of them have no connection. Only three pins require wiring. The power supply is connected to two of the three pins, $V_s$ and GND. Consult the datasheet for the acceptable range of $V_s$. The last of the three pins, $V_{out}$, is the analog output voltage pin and is to be connected to the analog-to-digital converter. Additional resistor and capacitors are required. See the wiring diagram below for details.

$$V_{out} = V_s \times (0.009 \times \text{pressure} - 0.095)$$

A graph for $V_{out}$ vs. pressure is shown below. Notice that no amplification of analog voltage is necessary for $V_s=5$ V.

Source: MPXA6115A datasheet
Software development

A code snippet in C below can be used for initializing the ADC of the MCU to interface with the MPXA6115A. It is assumed that the analog voltage is fed to the channel 0 of the ADC module. The digitized analog signal value will be stored in ATDDR0. Other assumptions are included in the comment statements. The code snippet can be easily modified for other conversion settings. Note that 10 bit resolution is recommended. Such resolution is 5 mV per bit.

```c
void initADC(void) {
    ATDCTL2 = 0x80;   // turn on ADC  ATDCTL2_ADPU=1
    ATDCTL3 = 0x08;   // 1 conversion/sequence
    ATDCTL4 = 0x01;   // 10-bit conv., 2Mhz ADC clock (assume bus clock=8MHz),
                      // fastest conv.
    ATDCTL5 = 0x80;   // right justify, unsigned, non-scanning,
                      // non-multichannel
    return;
}
```

To compute the pressure in kPa from the ADC reading, the equation

\[ \text{Vout} = \text{Vs} \times (0.009 \times \text{pressure} - 0.095) \]

can be used. Note that \( \text{Vout} = \text{ATDDR0} \times 5 \text{ mV} \). Manipulate the equation into pressure as the subject, then

\[ \text{Pressure} = \frac{\text{ATDDR0} + 95}{9} \text{ kPa} \]

A C function for reading the analog voltage and converting it into pressure in kPa is shown below. The delay for 1 ms in the function is to give sufficient time for the ADC to finish the conversion. Every execution of this function will result in a new pressure reading stored in the variable `pressure`.

```c
void readPressure(float pressure) {
    ATDCTL5 = 0x80;  // initiate ADC conversion
    waitms(1);       // delay 1 ms, this function was covered in class
    pressure = (ATDDR0+95)/9;  // compute pressure in kPa
    // note that for right justified data, there needs not the
    // instruction “pressure >> 6” as in left justified case
}
```