AN3521

Analog Sensor Measurement and Acquisition

Introduction

Author: Jason Layton, Microchip Technology Inc.

With automation and connectivity becoming increasingly prevalent in industrial, commercial and residential settings, the need for real-time monitoring using sensors has continued to grow. The types of sensors found in these settings are commonly used to measure and monitor a wide variety of physical properties, such as temperature, occupancy and pressure. This application note will discuss how the advanced analog peripherals found in many 8-bit PIC® microcontrollers, such as the 12-bit ADC with Computation (ADCC) and the integrated Operational Amplifier (OPA) modules, can be used for analog signal conditioning and acquisition in real-time sensor applications.

This document will briefly cover sensor basics and terminology, as well as demonstrate how several different types of dispersed analog sensors can be modularly interfaced to create a sensor net. The sensors used in this demonstration will measure the temperature, humidity, air quality and differential air pressure of the environment in which they are placed. The principles covered in this document can be expanded upon and implemented in many different sensor and real-time control applications.

Table of Contents

1. Sensors Overview

A sensor is a device that measures and monitors changes to the physical properties of a material or its environment. Sensors provide an electrical signal that represents the physical property being measured, which can be used in embedded systems for real-time monitoring or feedback control. The data acquired from electrical sensors can be used to determine if a system is operating correctly and often used for real-time monitoring in a wide variety of applications.

Several different types of electrical sensors are used in such areas as industrial manufacturing, automobiles, consumer electronics and home appliances. Embedded sensor applications can range from simple systems that use one sensor to control one component, to complex systems using data acquired from various sensors to monitor and control a number of diverse components. The feedback provided from sensors can be used to control actuators (e.g.,a motor, pump, valve or fan), as well as to make changes within the system to compensate for external factors and ensure operation stays within the limits of this application.

1.1 Types of Sensors

Many types of sensors are available to measure physical properties such as temperature, pressure, sound, light, motion and force. These physical properties measured by sensors will generally be mechanical, thermal, electrical, magnetic, chemical or optical. Sensors used in embedded applications typically fall under the following categories, based on how each respective physical property is measured by the sensor:

- Resistive Sensors can sense changes to a material’s resistive properties. Examples include:
  - Potentiometers
  - Strain Gauges (Load Cell)
  - Thermocouples
  - Photo-resistors
  - Thermistors
- Capacitive Sensors measure changes in a material’s capacitance. Examples include:
  - Pressure Sensors
  - Proximity Detection
  - Touch Sensing
  - Moisture Detection
- Inductive Sensors detect changes based on the electromagnetic induction of a material. Examples include:
  - Coil Detectors
  - Metal Detectors
  - Proximity Detection
  - Hall Effect Sensors

1.2 Digital and Analog Sensors

The types of sensors discussed in this application note can also be classified as analog or digital sensors. Analog sensors provide an analog output voltage directly related to the sensor measurement. This analog sensor output can range anywhere from 0% to 100% of the sensor operating voltage. Digital sensors typically provide an output signal that has already been converted to a digital signal.

When using analog sensors in embedded systems, an Analog-to-Digital Converter (ADC) is needed to convert the raw analog sensor output voltage to a digital value that can be used by the microcontroller. Frequently, the raw analog sensor output must be conditioned to amplify the signal or filter out unwanted noise before being converted by the ADC.

1.3 Sensor Characteristics and Terminology

The following list defines some common phrases and terminology used when discussing sensor measurement and acquisition in embedded applications:

Table 1-1. Sensor Characteristics and Terminology

Characteristic	Description
Linearity	Occurs when the output of a sensor is proportional to the sensor input (the change in the physical property being measured).
Range	The difference between maximum and minimum values that a sensor is capable of reliably measuring. Sensor manufacturers typically document the sensor's range of operation and specify the maximum and minimum recommended values that should be measured.
Precision	The degree in which a sensor measurement can be reproduced (how close each successive sensor measurement is to each other when the sensor input is unchanged).
Accuracy	The maximum difference (worst-case error) between the actual measured sensor value and the expected value of the sensor (how close a sensor can measure the correct value).
Sensitivity	The ratio describing how the sensor output signal changes in comparison to the change of the physical property being measured (the smallest increment of physical change a sensor can detect before the output changes).
Noise	A general term referring to any unwanted variation or disturbance of an electrical signal. Noise can appear in many different forms but generally follows a Gaussian distribution with some level of repeatability. Different methods of analog signal conditioning can be used to reduce and filter out noise.
Signal-to-Noise Ratio (SNR)	The ratio of the average power of a signal to the average power of its noise, normally expressed in decibels (dB). An SNR of greater than 0dB indicates the power of the signal is greater than the noise. An SNR of less than 0dB indicates the power of the noise is greater than the signal.
Drift	The change in a sensor's output over time, independent of the input. Drift is a result of a sensor physically changing due to age, degradation or environmental exposure and can lead to inaccurate sensor measurements over time. Sensor drift can be detected and corrected by performing periodic calibration.
Error	The difference between the expected output of a sensor and the actual measured value (any deviation from a sensor's expected performance curve).
Calibration	A method to improve a sensor's accuracy by performing adjustments (in software or hardware) used to correct for error in the sensor output. Calibrate a sensor when its performance is outside of an acceptable tolerance.
Response Time	The time needed for the output of a sensor to change from its previous state to reflect a new measured value. Some sensor outputs may take longer than others to settle when performing measurements to ensure accurate data collection.

1.4 Using Sensors in Control Systems

Sensors are commonly used in control systems to monitor operation and ensure components are performing within an expected tolerance. Control systems can generally be classified as open-loop systems or closed-loop feedback systems, depending on how the sensors are used in the application.

An open-loop control system does not rely on sensor feedback or the system output to control operation and will operate regardless of the system output. Open-loop systems operate without any means of accurately controlling the process in real-time since there is no sensor feedback injected into the system to control the output. Open-loop systems do not rely on real-time monitoring for error correction, and the output of an open-loop system can only be determined by the state of the inputs. Sensors are normally used in open-loops systems for real-time monitoring or data logging.

An example of an open-loop system is a building's water meter. The sensor used in this system measures the amount of water consumed and logs the data. This data is then read at a later date for billing purposes. In this example, there is no feedback to detect system failures (such as a water leak) and shut the water off. The only purpose of this type of control system is to measure and log the amount of water used.

Closed-loop control systems use real-time measurements and sensor data to provide feedback that allows the system to compensate for external factors and maintain a steady operation. The measured values from sensors can be used as feedback, giving a baseline to make changes within the system to compensate for external factors and control the output of the system. Feedback from sensors in closed-loop systems can be used to control actuators or other components to make changes that ensure operation stays within specification.

An example of a closed-loop feedback system using a sensor is the electric radiator fan used in most automobiles. The electric fan operates based on the feedback of a temperature sensor used to determine if the engine coolant is within the specified operating range. If the sensor feedback determines the engine coolant is getting too hot, the control system can decide to turn the fan on high speed until the coolant has dropped down within the specified operating temperature and, subsequently, shut the fan off until the coolant heats up again.

1.5 Sensor Nets

A sensor net is a group of dispersed sensors used to measure or monitor various physical properties throughout an environment or location. Sensor nets are useful when multiple sensors placed in different locations need to be monitored in real-time and the collected data needs to be sent to a central system for monitoring or real-time control. Sensor nets can be used in agricultural, industrial, residential and medical applications, the main advantages being the system modularity and connectivity.

Each sensor within the net can be viewed as an individual module (or node) whose job is to measure one thing and transmit that information to a central system. The behavior of each sensor in the system is independent of the others, meaning and if one sensor node fails, the others will remain unaffected. So when a sensor fails, or if a new sensor needs to be added to the net, the change can be made quickly and easily due to the system's modularity. Each module within a sensor net is configured to collect the data expected by the central system and transmit it in the correct form using the proper protocol. This allows the central system to successfully receive the information and be able to understand the data from each sensor.

A sensor net can take several different types of sensors that are distributed throughout a location and connect them to a main system for real-time control or monitoring. The connectivity component of a sensor net can either be a wired or wireless connection between each module (or node) and the main system. Each sensor used in a sensor net often has its own hardware that enables it to transmit data, either wirelessly or on a wired connection, using the appropriate communications protocol. The communication component in a sensor net can meet the speed requirements of the application and be reliable enough to avoid connectivity or data integrity issues.

2. Application Theory of Operation

This application is designed to demonstrate analog sensor measurement and acquisition tips by interfacing several different dispersed sensors to create a sensor net. Each sensor in this application is treated in a modular nature in hardware and software. Sensors can be added or removed to the system without major modifications to the firmware. The firmware is written to treat each sensor as a modular node, and each sensor node is setup to send its acquired data to the main system in a standardized format.

This application note mainly focuses on using the advanced analog peripherals found on many 8-bit PIC® microcontrollers for each sensor interface in the sensor net. The setup and implementation of the peripherals is covered below, as is the firmware used to measure and convert the raw analog output of each sensor into a format that can be used for real-time monitoring.

2.1 Analog to Digital Converter with Computation (ADCC) Module Overview

The ADCC module is a 12-bit Analog-to-Digital Converter (ADC) that has integrated hardware components that allow for advanced core independent computation such as averaging, oversampling, low-pass filtering, threshold / reference comparison and selectable interrupts. The ADCC module can also be configured to automate double sampling, compare current conversion results to previous conversion results, as well as setup auto-conversion triggers. The auto-conversion triggers can be setup to allow periodic ADC measurements without software intervention whenever the rising edge of the selected trigger source occurs. The ADC can also be configured to wake from Sleep mode to perform a measurement when an external trigger is received. This is advantageous in sensor net applications where each node only needs to make a measurement and send a value periodically, since the device can be put to Sleep for power saving when not active and wake up when another measurement is needed.

These advanced features make this 12-bit ADC an excellent choice for sensor measurement and acquisition applications. Most of the sensors used in this application have a raw analog output, and the ADCC module is used to measure the output voltage from each sensor and convert the raw analog value to a digital value that can be used by the microcontroller.

When using an ADC for sensor acquisition, two important considerations are the resolution and speed of the ADC. Select an ADC that is fast enough to accurately acquire and convert the output from an analog sensor. Analog signals from sensors are not normally one constant voltage, so the frequency in which the ADC can sample the analog signal is an important factor in ensuring the sensor is used in its full capability. The resolution of an ADC describes the number of bits used to represent the measured analog voltage as a digital value.

For example, the ADC used in this application provides 12-bits of resolution, meaning the ADC conversion result ranges anywhere from 0 to 4095, depending on the measured analog voltage. When using an ADC, each bit of resolution represents a fraction of a voltage based on the negative reference and positive reference. If a 12-bit ADC is used with V_{SS} as the negative reference, and the internal fixed voltage reference output of 4.096V is the positive reference, each bit of the ADC conversion result represents a step size of 1 mV. Higher ADC resolutions allow for more accurate measurements of analog signals since the step size of the digital representation is smaller.

Tip: Please refer to the Analog-to-Digital Converter with Computation (ADCC) chapter of the device data sheet for more information about setup and use of this peripheral.

2.1.1 Hardware Capacitive Voltage Divider (CVD)

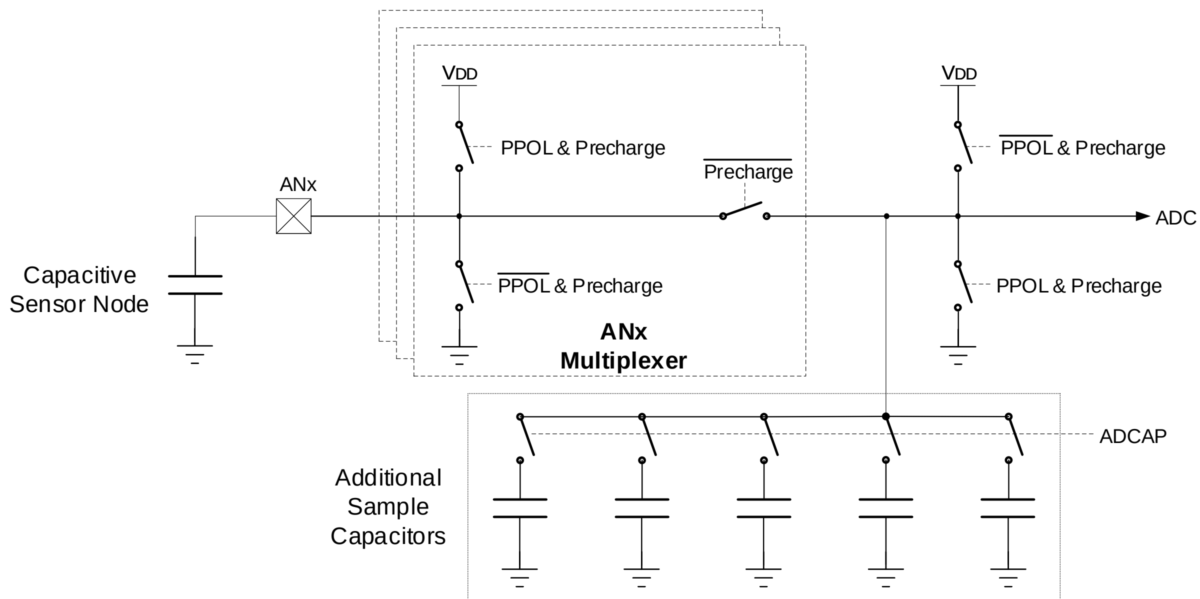
The ADCC module features a built-in hardware Capacitive Voltage Divider (CVD) that allows for capacitive sensing, which is used to measure relative humidity in this application. The integrated CVD hardware allows for capacitive measurements on any of the designated ADC channels while using the internal ADC Sample-and-Hold capacitance as a reference. The hardware CVD also features a pre-charge timer, an adjustable Sample-and-Hold capacitor array that can be added in parallel to the ADC Sample-and-Hold capacitor, and designated guard ring outputs.

The CVD includes the hardware needed to perform relative capacitive measurements core-independently. This hardware handles the pre-charging and discharging of the internal sample and hold capacitor, as well as the external capacitive sensor being measured. The CVD not only automates the charging and discharging of the respective capacitors, but it also core-independently creates a capacitive voltage divider during the ADC acquisition stages to allow for a conversion. The capacitive sensor node in the CVD block diagram (Figure 2-1) represents the capacitive humidity sensor used in this application.

Although the ADCC handles most of the CVD operation core-independently in hardware, the peripheral still must be configured correctly to ensure proper operation. The port functionality of the pin measured can be setup as it normally would for an ADC conversion, as can most of the ADC Configuration registers. When performing a CVD conversion, double sampling must be enabled by setting the DSEN bit since CVD operation depends on two conversions happening one after another.

ADPRE and ADACQ registers can be used to specify the pre-charge and acquisition time periods for the conversion. Given that during a CVD operation two conversions happen at once, the GO bit of the ADCC must be triggered accordingly, depending on the setup of the ADC Continuous Operation Enable bit (CONT). If continuous operation is enabled (CONT = `1`), the ADC automatically performs both conversions from a single trigger, which means the GO bit only needs to be set once. If continuous operation is not enabled (CONT = `0`), each conversion must then be triggered separately.

Figure 2-1. Hardware Capacitive Voltage Divider Block Diagram



2.2 Fixed Voltage Reference (FVR) Module Overview

When using analog peripherals like an ADC in sensor applications, a constant and reliable reference voltage is vital to ensure the accuracy of any measurements being made. The reference voltage determines the resolution and range of the analog-to-digital conversion. The Fixed Voltage Reference (FVR) module is a feature found on many PIC® microcontrollers and can be used to provide a stable output voltage reference that is independent of V_{DD}.

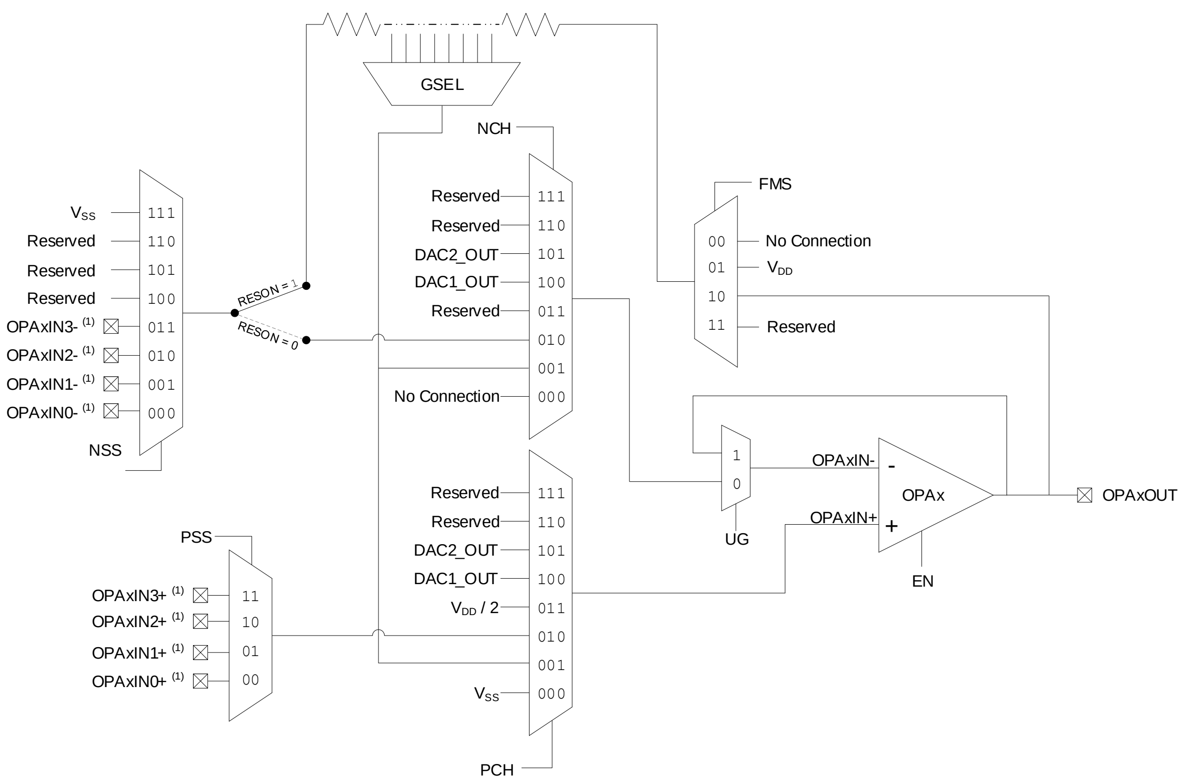
The ADREF register of the ADCC module is used to select an ADC positive reference voltage. The positive reference voltage of the ADC can be programmed to be provided from V_{DD}, an external voltage source, or the FVR. In this application, the FVR is selected as the positive reference source to the ADC and configured to generate a 4.096V output, unless otherwise noted.

2.3 Operational Amplifier (OPA) Module Overview

The OPA module provides PIC18 devices with a built-in general purpose operational amplifier used for analog signal conditioning. In sensor applications, raw analog outputs often require some level of analog signal conditioning such as amplification, filtering or buffering. The OPA module includes an integrated internal resistor ladder that allows the peripheral to be used as a programmable gain amplifier without external components. The OPA module can be configured in software in a non-inverting, inverting or unity gain configuration without needing to physically connect any of the pins outside of the device. Alternatively, the peripheral also provides the flexibility to setup the operational amplifier circuit externally if desired. This allows for external feedback resistors or components to be connected to any of the OPA pins.

Tip: Please refer to the Operational Amplifier (OPA) module chapter of the device data sheet for more information about setup and use of the peripheral. The MPLAB Code Configurator (MCC) provides an intuitive graphical user interface used to generate initialization code and functional APIs quickly and easily for the OPA module. Please visit www.microchip.com/mcc for more information.

Figure 2-2. OPA Module Functional Block Diagram



Note: Refer to the **"Pin Allocation Table"** section of the device data sheet for details about the OPAxIN- and OPAxIN+ availability per port.

2.3.1 Unity Gain Buffer for Analog Signal Conditioning

The OPA module is easily configured as a unity gain buffer. The module is configured by setting the UG bit of the OPAxCON0 register and selecting the appropriate non-inverting input source and channel, using the PSS and PCH bits of OPAxCON3 and OPAxCON2 respectively. A unity gain buffer is an operational amplifier circuit configuration that has a gain of “1”. When used in this configuration, the output signal of the OPA module is the same as the input signal provided. The unity gain configuration allows the OPA module to be used as a buffer, which protects the source of the analog signal from being affected by any load that may be on the circuit.

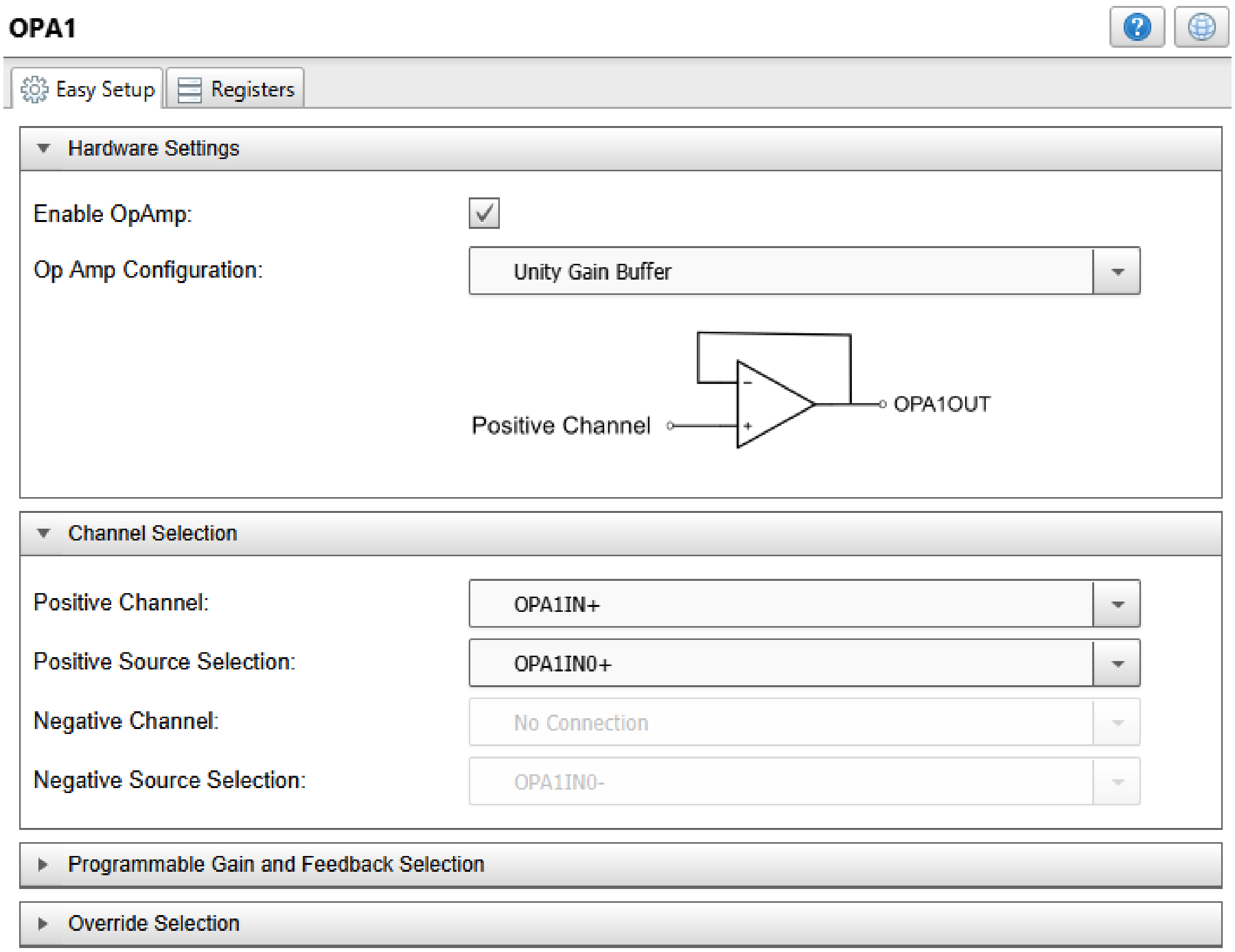
Buffering the analog output of a sensor also reduces the input impedance to the ADC. This is beneficial when a sensor has a high output impedance and is directly connected to an ADC for acquisition, then the resulting conversion result may be inaccurate due to the effects of the high impedance signal. Using the ADC to measure a high impedance analog signal can be problematic as the high impedance can lead to increased current at the time of sampling, which will ultimately affect the integrity of the ADC conversion result.

Operational amplifiers by nature have high impedance inputs and low impedance outputs, making them suitable as buffers in sensor applications. The OPA module can be used as a unity gain buffer on a high impedance sensor output in order to provide a low impedance signal to the ADC for conversion. Using a unity gain buffer helps ensure the data being received from a sensor is accurate, unaffected by any other load present in the system, and maintains signal integrity.

In this application, the OPA module is used as a unity gain buffer for the MQ-135 air quality sensor interface. When the Unity Gain Enable (UG) bit of OPAxCON0 is set, it connects the output of the OPA to the inverting input, forming a unity gain circuit configuration that needs no external jumpers between pins. When configured in a unity gain configuration, the designated inverting input pins of the microcontroller (OPAxIN-) are free to be used as general purpose I/O.

In this instance, the OPA module is used as a unity gain buffer to isolate the sensing circuit and ensure accurate readings from the ADC. The MPLAB Code Configurator (MCC) plug-in can be used to configure the OPA module quickly and easily, using the provided graphical user interface. Figure 2-3 shows an example of the MCC setup and configuration of the OPA module as a unity gain buffer.

Figure 2-3. MCC OPA Module Configuration (Unity Gain Buffer)

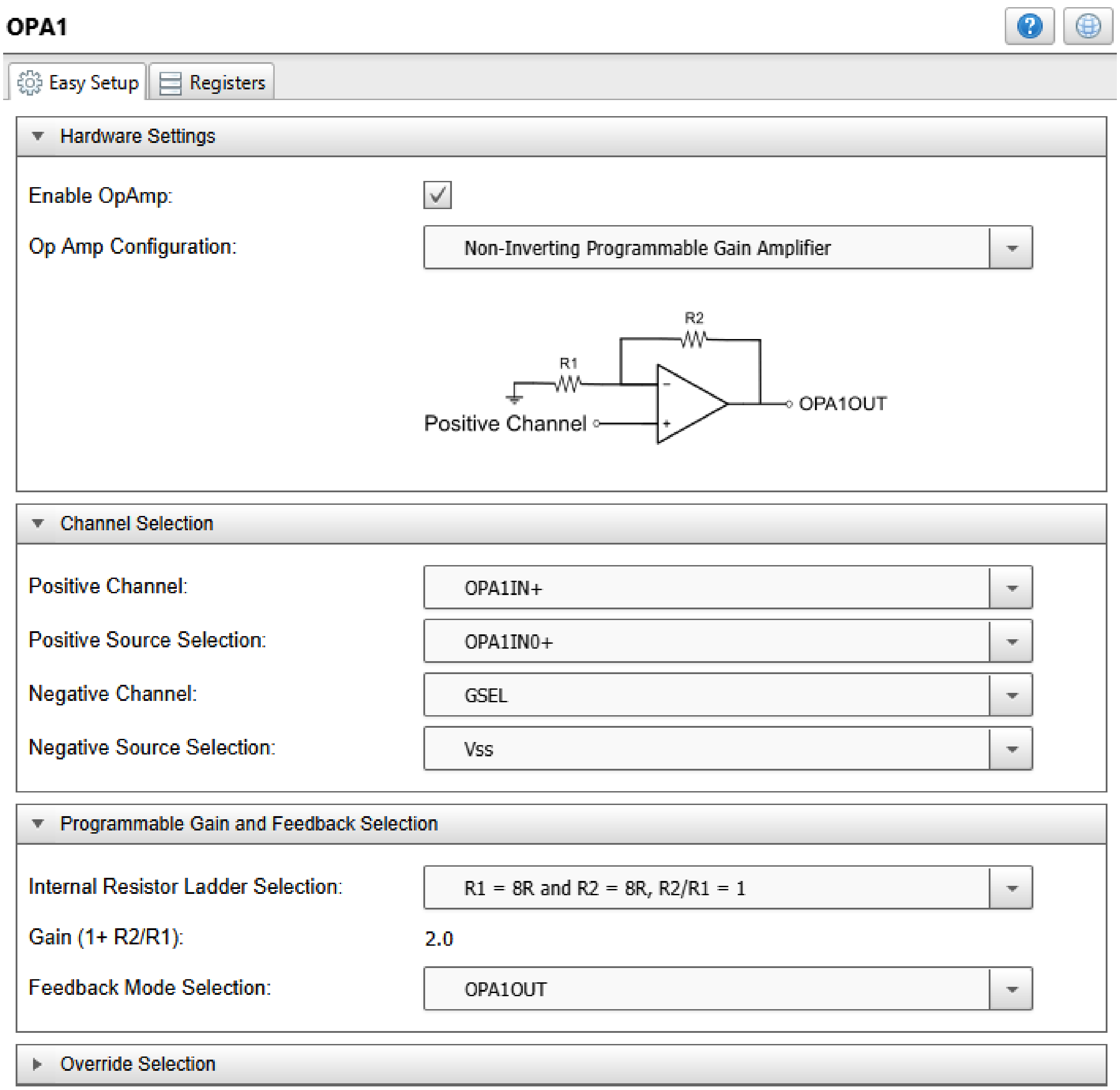


2.3.2 Programmable Gain Amplifier for Analog Signal Conditioning

The OPA module can also used as a programmable gain amplifier. It can be used to amplify or attenuate an analog signal using the internal resistor ladder, or amplify or attenuate an external feedback network connected to the specified OPA module pins. The module can be setup as a non-inverting programmable gain amplifier or an inverting programmable gain amplifier, depending on how the OPA source and channel bits are programmed. The internal resistor ladder is used to control the gain of the OPA circuit by setting the RESON bit of OPAxCON1 to enable the resistor ladder. It also uses the GSEL bits of OPAxCON1 to select from the available resistor ratios to control the gain. The internal resistor ladder must then be connected to the inverting input of the OPA module by setting the NCH bits of OPAxCON2 accordingly. The actual gain of the circuit depends on whether the module is setup in an inverting or non-inverting configuration and how the inputs to the OPA module were programmed. The resistor ratio selected, using the GSEL bits, is used to calculate the expected gain from the circuit.

Using the OPA module as a programmable gain amplifier is beneficial in sensor applications where the output voltage is a very small signal and, therefore, needs to be amplified to a level suitable for the analog-to-digital conversion. Many analog sensors provide a raw analog output that has not had any signal conditioning performed. The raw signal often needs to be amplified or filtered before being measured and used by the microcontroller. By amplifying the analog signal to match the operating voltage range of the ADC being used, the signal is converted with higher resolution and precision. There may also be cases where an analog signal coming from a sensor is outside of the ADC voltage range, causing any measurements outside of the operating range to get clipped. The OPA module can be used to attenuate, or shrink, the analog signal so that it stays within the readable range of the ADC used for conversion. Figure 2-4 shows an example of the MCC setup and configuration of the OPA module as a non-inverting programmable gain amplifier.

Figure 2-4. MCC OPA Module Configuration - Non-Inverting Programmable Gain Amplifier

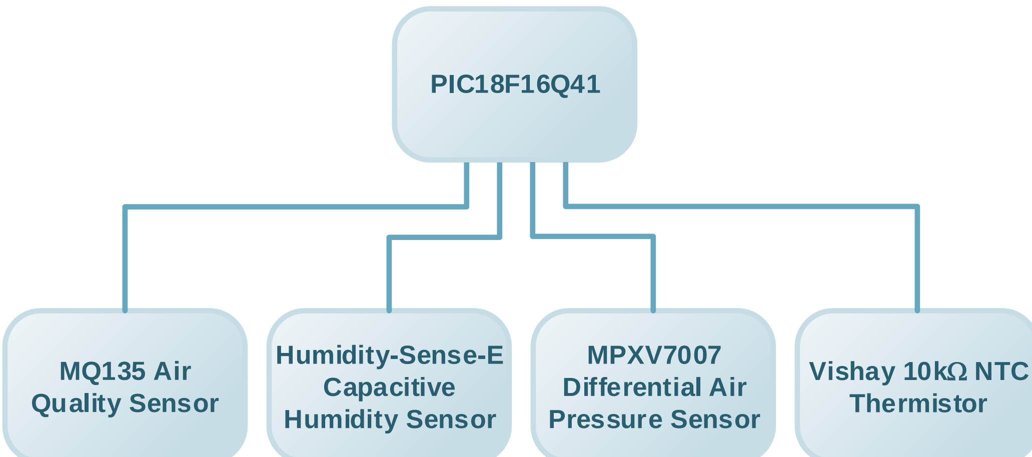


3. Analog Sensor Net Hardware Components

This application uses the following hardware components:

- PIC18F16Q41 Microcontroller
- Curiosity Low Pin Count Development Board (DM164137 Rev4)
- MikroElektronika Air Quality Click Board (MQ-135 Gas Sensor)
- NXP Integrated Silicon Pressure Sensor (MPXV7007)
- Vishay Humidity-Sense-E Capacitive Humidity Sensor
- Vishay NTC (10 kΩ) Thermistor

Figure 3-1. Application Block Diagram



3.1 PIC18F16Q41 Microcontroller Overview

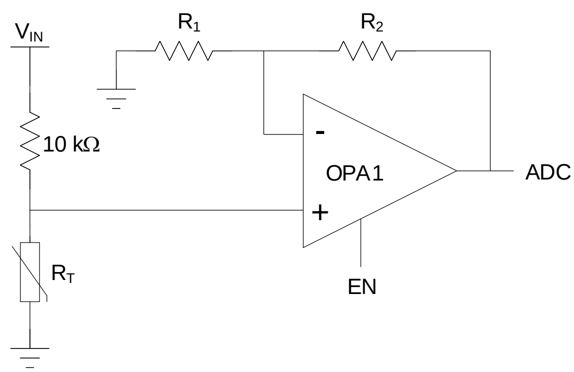
The PIC18F16Q41 microcontroller is a low pin count, high performance device suitable for sensor and real-time control applications. This device family features several advanced analog peripherals such as an integrated Operational Amplifier (OPA) module, 12-bit Analog-to-Digital Converter with Computation (ADCC) and two 8-bit Digital-to-Analog Converters (DACs). This device family also features a dual output 16-bit PWM, a programmable 32-bit CRC with memory scan, Direct Memory Access (DMA) capabilities and standalone SPI and I^2C peripherals. Please refer to PIC18F16Q41 for more information about the PIC18-Q41 device family.

3.2 Temperature Sensor Overview

The temperature sensor used in this application is a Vishay 10 kΩ NTC Thermistor. A thermistor is a component whose resistance changes in relation to temperature. Thermistors are generally classified as NTC (negative temperature coefficient) or PTC (positive temperature coefficient) components. The resistance of NTC thermistors decreases as temperature rises while the resistance of PTC thermistors increases as temperature rises. Thermistors are low-cost components capable of quickly and accurately measuring ambient temperatures. Since the relation between resistance and temperature is non-linear, please note that any compensation routine used to calculate degrees from the thermistor resistance can account for this. Thermistors are often specified by the resistance in ohms at room temperature (25°C). Thermistors with a lower resistance offer more noise immunity at the cost of increased current consumption. Thermistors with a higher resistance draw less current but are more susceptible to noise.

Figure 3-2 shows the circuit used to interface the Vishay 10 kΩ NTC thermistor with the PIC18F16Q41 microcontroller in this application. In the circuit interface, a voltage divider is formed using the 10 kΩ thermistor and a 10 kΩ resistor. The voltage divider supply is represented by V_{IN} and is 3.3V. The output of the voltage divider is connected to the non-inverting input of the PIC18F16Q41 OPA module. The OPA module is configured as a non-inverting programmable gain amplifier for this sensor interface. In this setup, the resistors labeled R_1 and R_2 in the circuit represent the operational amplifier feedback network, which is provided using the OPA module internal resistor ladder for an overall gain of 2. Thermistors are generally capable of measuring a wide range of temperatures, but the thermistor is used in this application to measure the ambient temperature inside of a building. The thermistor sensing circuit output voltage is amplified in this application to take full advantage of the ADC operating voltage and gain more accuracy and precision when measuring the voltages within the desired range.

Figure 3-2. Vishay 10 kΩ NTC Thermistor Interface Circuit



3.2.1 Temperature Sensor Measurement and Acquisition

The Steinhart–Hart model (Equation 3-1) is widely accepted as one of the best ways of mathematically modeling the relationship between the resistance of a thermistor and its temperature. This model is used to calculate ambient temperature using the Vishay 10 kΩ NTC thermistor in this application. The equation returns the temperature in degrees Kelvin (K). This application requires a temperature output in degrees Celsius (C), so the final step in the temperature compensation routine is to convert the result from Kelvin to Celsius after using the Steinhart-Hart equation.

Equation 3-1. Steinhart–Hart Model (Thermistor Resistance to Temperature)

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{\beta} ln\left(\frac{R_T}{R_0}\right)$$

- T = Thermistor Temperature (Kelvin)
- T_0 = Room Temperature (at 25°C) in Kelvin
- β = Steinhart–Hart Model Temperature Coefficient
- R_T = Thermistor Resistance
- R_0 = Thermistor Resistance at 25°C

The OPA module is used in this sensor interface as a non-inverting programmable gain amplifier. The internal resistor ladder of the OPA module is used as the operational amplifier feedback network to obtain a gain of 2 on the signal prior to ADC conversion. For more information about the setup of the OPA module, refer to Programmable Gain Amplifier for Analog Signal Conditioning.

The amplified signal from the OPA output is connected internally to the ADCC module for conversion. The ADCC is programmed to operate in Basic mode. The positive reference voltage selected was V_{DD} (5V), instead of the FVR, to widen the operating range of the ADC and allow for a wider range of voltages to be measured. Example 3-1 shows the firmware used to measure the resistance of the NTC thermistor and to calculate the temperature using the acquired value.

Example 3-1. Temperature Sensor Measurement and Compensation Routine

```
void Temp_SendData(void) {

    #define ADCC_RESOLUTION         4096
    #define ADCC_POS_REFERENCE      4.70        // Measured VDD;
    #define TEMP_POS_REF            3.3         // Thermistor Positive
Reference;
    #define TEMP_BETA               3289.66     // Thermistor Beta Coefficiant;
    #define K_CONST                 273.15

    float TempK, TempC, Temp_Voltage = 0;
    float RT = 0; // Voltage Divider R2 (Thermistor)
    float R0 = 10000.0; // Voltage Divider R1;
    float T0 = 298.15;
    adc_result_t Temp_Raw = 0;

    ADCON0bits.GO = 1; // Trigger 1st ADC Conversion;
    while (!ADCON0bits.GO); // Wait for Conversion to Complete;
    Temp_Raw = ADCC_GetConversionResult(); // Get ADC conversion result;

    // Calculate ADC Voltage
    Temp_Voltage = ((float) Temp_Raw / ADCC_RESOLUTION) * ADCC_POS_REFERENCE;

    // Determine Thermistor Resistance (Voltage Divider with Known R0)
    RT = ((Temp_Voltage / 2) * R0);
    RT = (RT) / (TEMP_POS_REF - (Temp_Voltage / 2));

    // Determine Temperature from Resistance using Steinhart-Hart model
    TempK = log(RT / R0);
    TempK = TempK * (1 / TEMP_BETA);
    TempK = TempK + (1 / T0);
    TempK = 1 / TempK;

    TempC = TempK - K_CONST;
    printf("Temperature(C), %1.1f \r\n", TempC);
}
```

3.3 Humidity Sensor Overview

The Vishay Humidity-Sens-E is a capacitive sensor used to measure the humidity of an environment accurately from 10% to 90% relative humidity. The sensor is made from a non-conductive foil covered on both sides with a thin layer of gold. The non-conductive foil in the center acts as a dielectric between the two thin layers of gold. This serves as the conductive plates and forms a capacitor. The dielectric constant of the non-conductive foil in the sensor changes in relation to the relative humidity of the environment in which it is placed. This is due to the presence of moisture, which in turn directly changes the capacitance of the sensor. The capacitance can be measured and used to determine the relative humidity, based on the provided characteristic curve, which then compares relative humidity to the sensor capacitance.

3.3.1 Humidity Sensor Measurement and Acquisition

The Capacitive Voltage Divider (CVD) feature of the ADCC peripheral is used to measure the relative capacitance of the humidity sensor in this application. The configuration of the ADCC for CVD conversion is described in Hardware Capacitive Voltage Divider (CVD). The output of the humidity sensor ranges from 110 pF to 160 pF. In order to use the CVD to calculate the actual sensor capacitance, there must be a point of reference to provide a scale for calculating it based on the CVD conversion results.

To provide this baseline, five reference capacitors of known capacitance are measured using the CVD. The conversion results are then used to determine the relationship between the results of the CVD conversion and the known actual capacitance of the components being measured. This baseline is then used to generate the formula described in Equation 3-2, which converts the CVD conversion result to relative capacitance (pF). In this equation, CVD_RES represents the measured CVD result (converted into volts) and the ERROR component is included to compensate for any error from the conversion or parasitic capacitance that may affect the measurement.

The humidity sensor used in this application is a two-terminal capacitive device and is connected to an analog channel of the microcontroller for sensing, as shown in Figure 2-1. The capacitive sensor node in this figure represents the humidity sensor. One terminal of the humidity sensor is connected to the specified ADC positive channel while the other is connected to ground.

Equation 3-2. CVD Conversion Result (V) to Relative Capacitance (pF)

$$C = \left(301.16 * CVD\_RES^2 - 902.16 * CVD\_RES + 764.38\right) \pm ERROR$$

After converting the CVD conversion result to a capacitance value, the next step in figuring out the relative humidity is to apply a compensation routine using the information provided in the humidity sensor data sheet. The data sheet provides a characteristic curve, which describes the relationship between the sensor capacitance and the relative humidity expressed as a percentage. The data from the provided characteristic curve forms a polynomial expression that converts the calculated sensor capacitance to relative humidity, as shown in Equation 3-3.

In this equation, the term C is derived from the capacitance calculated using Equation 3-2. The software implementation of the steps required to measure the sensor and convert the raw analog data acquired to relative humidity are shown in Example 3-2. The ADCC is setup to perform a CVD conversion as previously described and the FVR is selected as the positive ADC reference source with a voltage of 4.096V.

Equation 3-3. Relative Capacitance (pF) to Relative Humidity (%)

$$RH = -0.0405C^2 + 12.915C - 929.12$$

Example 3-2. Humidity Sensor Measurement and Compensation Routine

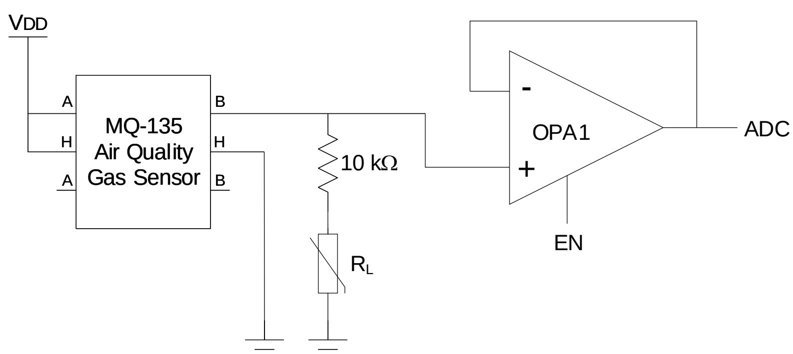
```
void Humid_SendData(void) {
    #define CAP_POLYTERM2           301.16
    #define CAP_POLYTERM1           902.16
    #define CAP_POLYTERM0           764.38
    #define CAP_ERROR               -12
    #define HUMID_POLYTERM2         -0.0405
    #define HUMID_POLYTERM1         12.915
    #define HUMID_POLYTERM0         -929.12
    #define HUMID_ERROR             0
    #define ADCC_RESOLUTION         4096
    #define ADCC_POS_REFERENCE      4.096

    float Humid, Humid_Capacitance, Humid_Voltage = 0;
    adc_result_t Humid_Raw = 0;

    ADCON0bits.GO = 1; // Trigger 1st ADC Conversion;
    while (!ADCON0bits.GO); // Wait for Conversion to Complete;
    Humid_Raw = ADCC_GetConversionResult(); // Get ADC conversion result;

    Humid_Voltage = ((float) Humid_Raw / ADCC_RESOLUTION) * ADCC_POS_REFERENCE;
    Humid_Capacitance = ((CAP_POLYTERM2 * (Humid_Voltage * Humid_Voltage))-
(CAP_POLYTERM1 * (Humid_Voltage))+(CAP_POLYTERM0))+(CAP_ERROR);

    Humid = (HUMID_POLYTERM2 * (Humid_Capacitance * Humid_Capacitance));
    Humid = Humid + (HUMID_POLYTERM1 * Humid_Capacitance);
    Humid = Humid + HUMID_POLYTERM0;

    printf("Relative Humidity, %.2f %%\r\n", Humid);
}
```

3.4 Air Quality Sensor Overview

The air quality sensor used in this application is the MQ-135 gas sensor, which provides a raw analog output voltage in relation to the concentration of different gasses. The MQ-135 gas sensor is a general-purpose air quality sensor used for detecting (with proper setup and calibration) the presence of different gasses and pollutants such as smoke, ammonia, carbon dioxide or alcohol. The air quality gas sensor has an integrated heating element that helps maintain the correct operating temperature to ensure the most accurate and consistent measurements. Drive the heating element to a constant voltage of 5V during operation. It is also recommended to preheat the sensor for 24 hours to ensure maximum sensitivity and accuracy.

The MQ-135 gas sensor is constructed using a material whose resistivity changes in the presence of the target gases. The output of this gas sensor is provided as a voltage across a load resistor, which is needed to calculate the PPM (parts per million) concentration of the target gases. To calculate the PPM concentration of the target gases, the sensor resistance in clean air must be calculated and stored and the load resistance must be known. Once done, the sensor resistance can be calculated using the ADC result. Figure 3-3 shows the circuit used to interface the MQ-135 air quality gas sensor. The raw analog output of the gas sensor is connected to the non-inverting input of the OPA module on the PIC microcontroller. R_L represents a potentiometer that controls the gas sensor circuit load resistance used to adjust the sensitivity of the sensor.

Figure 3-3. MQ-135 Air Quality Gas Sensor Circuit



3.4.1 Air Quality Sensor Measurement and Acquisition

In this application, the MQ-135 gas sensor is used to measure the concentration of carbon monoxide gas in parts per million (PPM). The gas sensor initialization routine not only sets up the peripherals needed to perform this measurement, but also measures the output of the air quality sensor in clean air. This is a required baseline for calculating the concentration of carbon monoxide gas. When the air quality sensor is initialized, it must be placed in a known clean air environment to ensure accurate measurements. The MQ-135 sensor is sensitive to a wide variety of gases. The MQ-135 sensor data sheet provides characteristic curves for each respective gas so that the concentration of the target gas can be calculated. The characteristic curves provided in this data sheet show the typical sensitivity characteristics of several different gases and illustrates the relationship between gas concentration to the sensor resistance.

The OPA module is used in this sensor interface as a unity gain buffer while the ADCC is used to convert the analog output voltage to a digital value. The implementation and setup of the OPA module can be found in Unity Gain Buffer for Analog Signal conditioning. The ADCC is setup to operate in Basic mode and the FVR is chosen as the positive reference voltage (4.096V). The raw analog output from the sensor is connected to the non-inverting input of the OPA module and the output of the OPA module is routed internally to the ADCC for conversion. After the analog output voltage from the sensor is acquired, a compensation routine is used to calculate the concentration of carbon monoxide gas detected. Example 3-3 shows the implementation in software used to acquire the air quality sensor output voltage. It also shows the compensation routine used to calculate the concentration of carbon monoxide. The formula used to convert the raw analog voltage from the sensor to concentration in parts per million is derived using the characteristic curves and data provided in the sensor data sheet.

Example 3-3. Air Quality Sensor Measurement and Compensation Routine

```
void MQ135_SendData(void) {
    #define PPM_CLEAN_AIR            5
    #define MQ135_TERM1              662.9382
    #define MQ135_TERM0              -4.0241
    #define ADCC_RESOLUTION          4096
    #define ADCC_POS_REFERENCE       4.096

    float R1 = 9500.0; // R1 = Load Resistor (Ohm);
    float R0; // Resistance Value in Clean Air;
    float ratio; // RS / R0 Ratio;
    float CO_PPM, MQ135_Voltage = 0;
    float RS; // Sensor Resistance;
    adc_result_t MQ135_Raw = 0;

    ADCON0bits.GO = 1; // Trigger ADC Conversion;
    while (!ADCON0bits.GO); // Wait for Conversion to Complete;
    MQ135_Raw = ADCC_GetConversionResult();

    MQ135_Voltage = ((float) MQ135_Raw / ADCC_RESOLUTION) * ADCC_POS_REFERENCE;
    RS = R1 * (ADCC_POS_REFERENCE - MQ135_Voltage) / MQ135_Voltage;

    CO_PPM = (RS / R0);
    CO_PPM = pow(CO_PPM, MQ135_TERM0);
    CO_PPM = MQ135_TERM1 * CO_PPM;

    printf("CO, %.1f PPM \r\n", CO_PPM);
}
```

3.5 Differential Air Pressure Sensor Overview

The MPXV7007DP is a differential air pressure sensor that utilizes a piezo-resistive transducer as the sensing element. This sensor provides an analog output voltage linearly proportional to the differential pressure measured between two physical input ports. Differential pressure sensors work by measuring the absolute air pressure in different areas (or rooms) by using separate physical tubes connected to the pressure sensing element and then comparing the results. The resulting differential pressure is the difference in pressure between the points of measurement.

A differential pressure sensor is used to maintain a positive pressure or negative pressure room, depending on the application. Positive pressure rooms have a higher pressure than the surrounding area, allowing for air to leave the room while not allowing for unfiltered air from the surrounding area to circulate back into the room. This principle is commonly used in hospitals, clean rooms and industrial manufacturing to ensure the positive pressure room does not become contaminated with outside air.

Negative pressure rooms work in the opposite way. Air from the surrounding area can flow in freely, but air from within the negative pressure room cannot flow out and contaminate the surrounding areas. Many residential and commercial HVAC systems also use differential pressure sensors to optimize heating and cooling of an environment by ensuring that air is flowing properly throughout a building.

3.5.1 Differential Air Pressure Sensor Measurement and Acquisition

The MPXV7007DP sensor measures differential air pressure ranging from -7 kPa to 7 kPa (-1 psi to 1 psi) and provides a corresponding analog output voltage measured using an Analog-to-Digital Converter. The sensor output voltage ranges from 0V to approximately 4.5V, depending on the measured value. The output voltage relationship to differential pressure in kPa is described using the transfer function in Equation 3-4. In this equation, V_{OUT} represents the analog output voltage of the sensor and V_S represents the sensor operating voltage. The ERROR component of the equation is used to calibrate the sensor output by performing a differential air pressure measurement at room temperature with no fittings attached to the input ports. This measurement provides a baseline that can be used to signify 0 kPa differential air pressure. If the compensated sensor value is non-zero during this calibration routine, the obtained value is used as the ERROR component to correct the compensated output value.

Equation 3-4. Differential Air Pressure Transfer Function

$$P = \frac{\frac{V_{OUT}}{V_S} - 0.5}{0.057} \pm ERROR$$

Example 3-4 shows the implementation in software used to acquire the analog output voltage of the sensor. The equation also shows the compensation routine used to attain the differential air pressure value in kPa. The ADCC is setup to operate in Basic mode and the FVR was selected as the positive reference to the ADC (4.096V). The differential air pressure value calculated is returned as a floating-point value and is either negative or positive, depending on the differential air pressure measured by the sensor. The first step in the compensation routine converts the acquired ADC value to a voltage represented as a floating-point value. The second step applies the transfer function previously described in order to obtain the final differential air pressure value in kPa.

Example 3-4. Differential Air Pressure Sensor Measurement and Compensation Routine

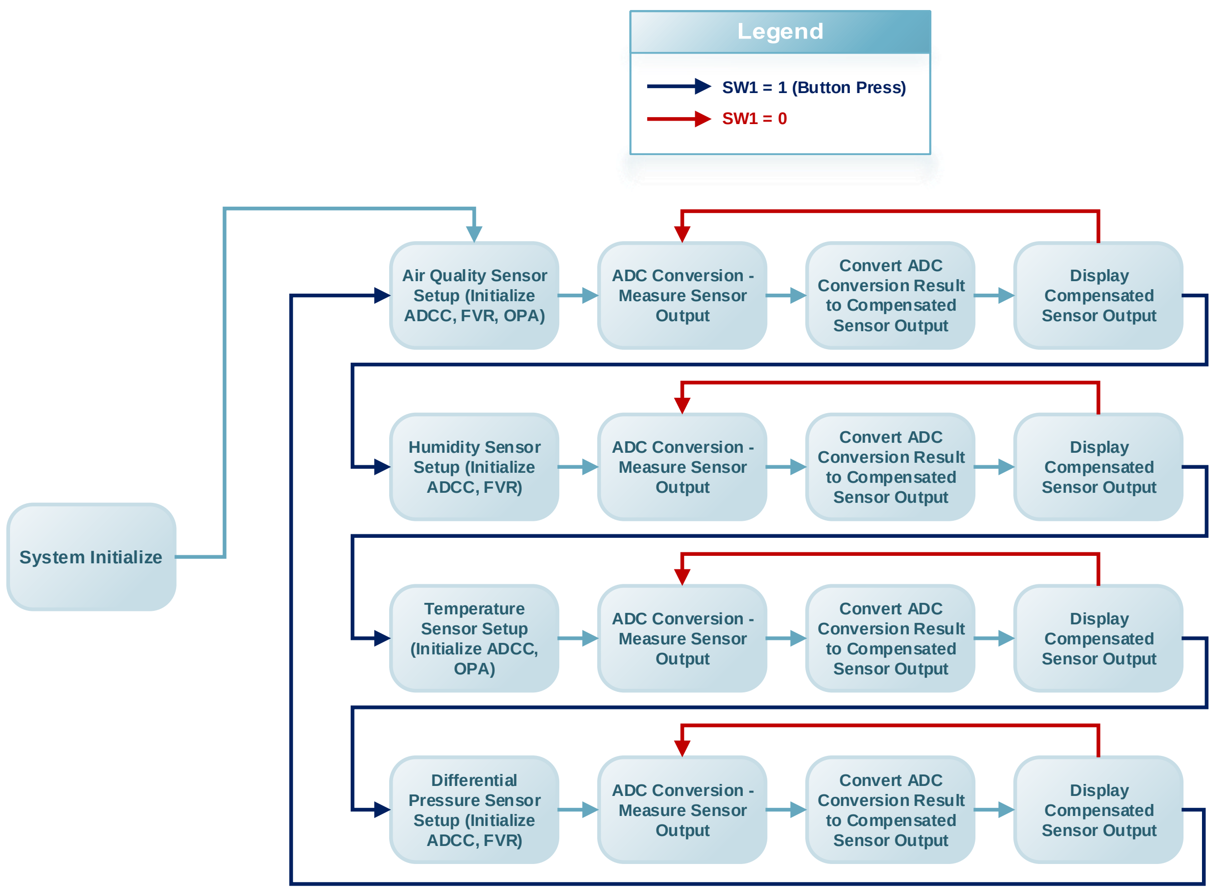
```
void Press_SendData(void) {

    #define PRESSURE_CONSTANT       0.057
    #define PRESSURE_VSUPPLY        4.75
    #define PRESSURE_ERROR          0.30
    #define ADCC_RESOLUTION         4096
    #define ADCC_POS_REFERENCE      4.096

    float Pressure, Press_Voltage = 0;
    adc_result_t Press_Raw = 0;

    ADCON0bits.GO = 1; // Trigger ADC Conversion;
    while (!ADCON0bits.GO); // Wait for Conversion to Complete;
    Press_Raw = ADCC_GetConversionResult();

    // Differnetial Air Pressure Compensation Routine;
    Press_Voltage = ((float) Press_Raw / ADCC_RESOLUTION) * ADCC_POS_REFERENCE;
    Pressure = (((Press_Voltage / PRESSURE_VSUPPLY)-(0.5)) /
(PRESSURE_CONSTANT)) + PRESSURE_ERROR;

    printf("Differential Pressure, %.2f kPa \r\n", Pressure);
}
```

4. Analog Sensor Net Description and User Interface

This application consists of four different analog sensors, each with its own sensor interface and peripheral configurations. Upon power-up, the application goes through general system initialization functions such as port configuration, oscillator setup and initialization of any other peripherals needed for the sensor net. Once system initialization is complete, the application enters the sensor net state machine. The sensor net state machine is responsible for switching between sensor nodes and ensuring all peripherals are configured properly for each respective sensor interface. Figure 4-1 shows the flow of the sensor net state machine and illustrates in detail how the application firmware works. Immediately following the system initialization, the UART module transmits a start-up message that says "PIC18F16Q41 Sensor Net Application Note" and "Press button S1 to cycle through sensor nodes".

The application was created using the Curiosity Low Pin Count (LPC) Development Board. The push button labeled S1 (connected to pin RC4) was used to cycle through sensor nodes in this example. Once the splash messages have been displayed, the state machine will transition to the initialization of the MQ-135 air quality sensor. After initialization of the peripherals needed in the air quality sensor interface, the sensor will periodically be measured and converted while the output will be displayed using the UART. If the S1 button is pressed any time during operation, the firmware will move to the next sensor in the state machine until it loops back around to the MQ-135 gas sensor. The UART TX pin is located on pin RB7 of the PIC18F16Q41 used in this application.

Figure 4-1. Analog Sensor Net State Machine

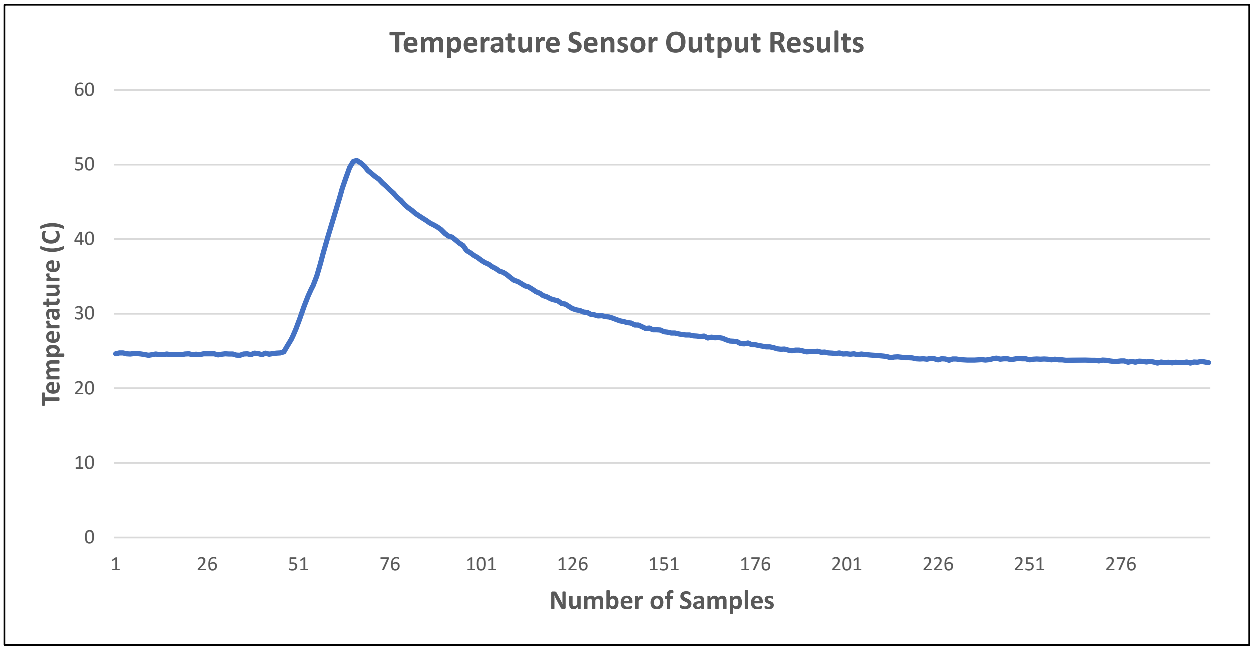


4.1 Temperature Sensor Output Results

Figure 4-2 illustrates the behavior of the temperature sensor used in this application. The temperature sensor compensation routine is written to output the measured temperature from the thermistor in degrees Celsius. The ambient temperature of the environment in which the application is placed is approximately 25°C, as shown in the graph below.

Heat is applied around the thermistor sensing circuit to demonstrate how the sensor behaves and shows the output as temperature rises. Once the sensor is heated up to around 50°C, the heat source is removed and the sensor is allowed to stabilize back to an ambient temperature of around 25°C, as shown in Figure 4-2. The thermistor used in this application provides accurate temperature measurements and is capable of measuring temperature changes in a fast response time.

Figure 4-2. Vishay NTC (10 kΩ) Thermistor Output

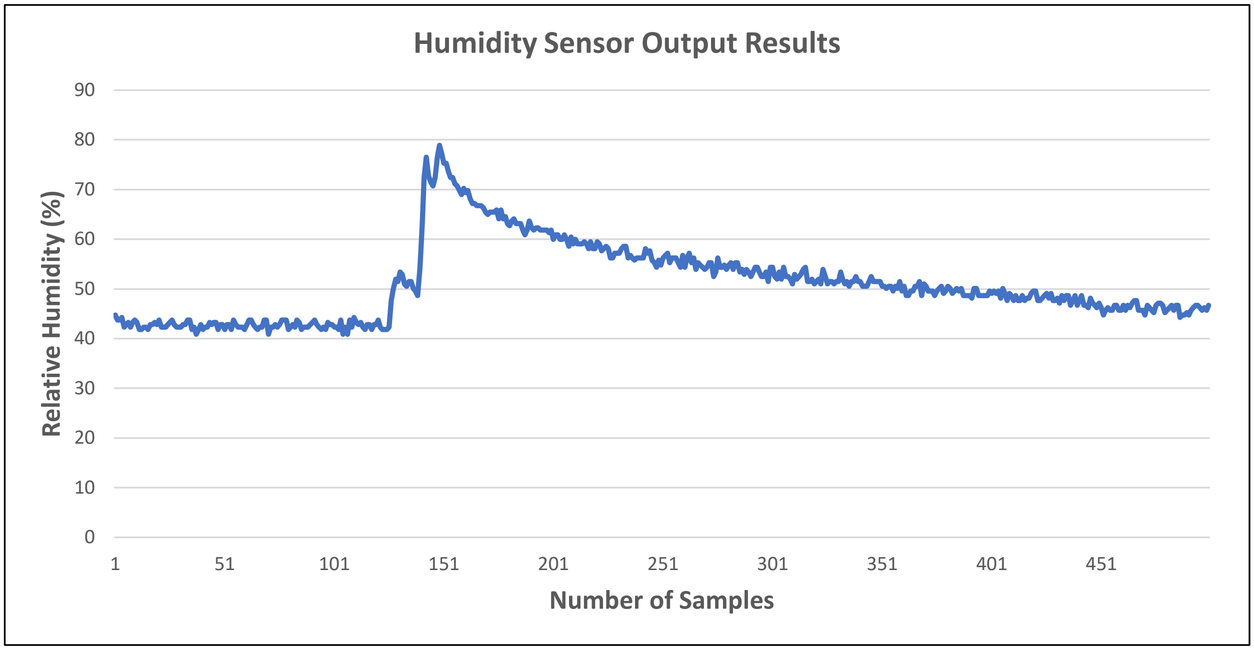


4.2 Humidity Sensor Output Results

Figure 4-3 shows the output of the capacitive humidity sensor used in this application. The compensation routine used for the capacitive humidity sensor is written to calculate and print the measured relative humidity as a percentage. The ambient relative humidity of the environment where the sensor is placed is around 40%, as shown in the graph below.

Humidity is purposefully added into the environment to demonstrate how the sensor behaves and to show the sensor behavior with a change in humidity. The spike in Figure 4-3, where the relative humidity jumps to around 80%, shows when humidity is added into the environment. The sensor is then given time to stabilize until the output returns to its ambient value. The Vishay Humidity-Sense-E sensor has a fast response time and is highly sensitive to changes in the environment.

Figure 4-3. Vishay Humidity-Sense-E Capacitive Humidity Sensor Output

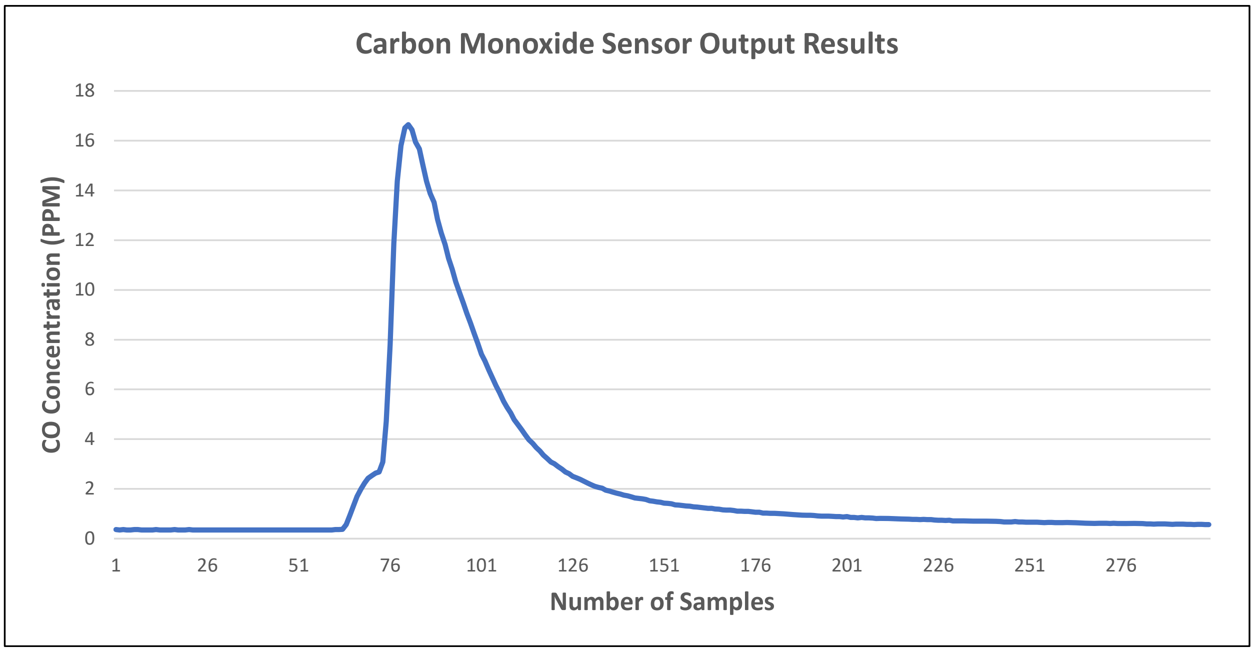


4.3 Air Quality Sensor Output Results

Figure 4-4 shows the output results of the air quality sensor used in this application. The MQ-135 air quality sensor is capable of detecting the presence of several different harmful gases. However, in this application, it is used to detect and approximate the concentration of carbon monoxide gas. The compensation routine used to interface the air quality sensor is written to calculate and print the concentration of carbon monoxide gas in parts per million (PPM). The ambient concentration of carbon monoxide gas is less than 1 PPM, as shown in Figure 4-4. Carbon monoxide is introduced into the environment where the sensor is placed in order to illustrate the sensor behavior when harmful gas is detected. As shown, the air quality sensor detected a spike in carbon monoxide gas as the concentration rose to around 18 PPM. The sensor is then allowed to stabilize and return back to the ambient gas concentration.

Important: In this application, the MQ-135 air quality sensor is used academically to detect the presence of carbon monoxide gas and provide an approximate PPM measurement. If using this sensor in other applications, additional calibration and compensation may be required to accurately measure the PPM concentration of carbon monoxide gas. Prolonged exposure to heightened levels of carbon monoxide can be dangerous, so the proper steps must be taken to ensure the sensor is accurately measuring the concentration of carbon monoxide gas.

Figure 4-4. MQ-135 Air Quality Sensor Results

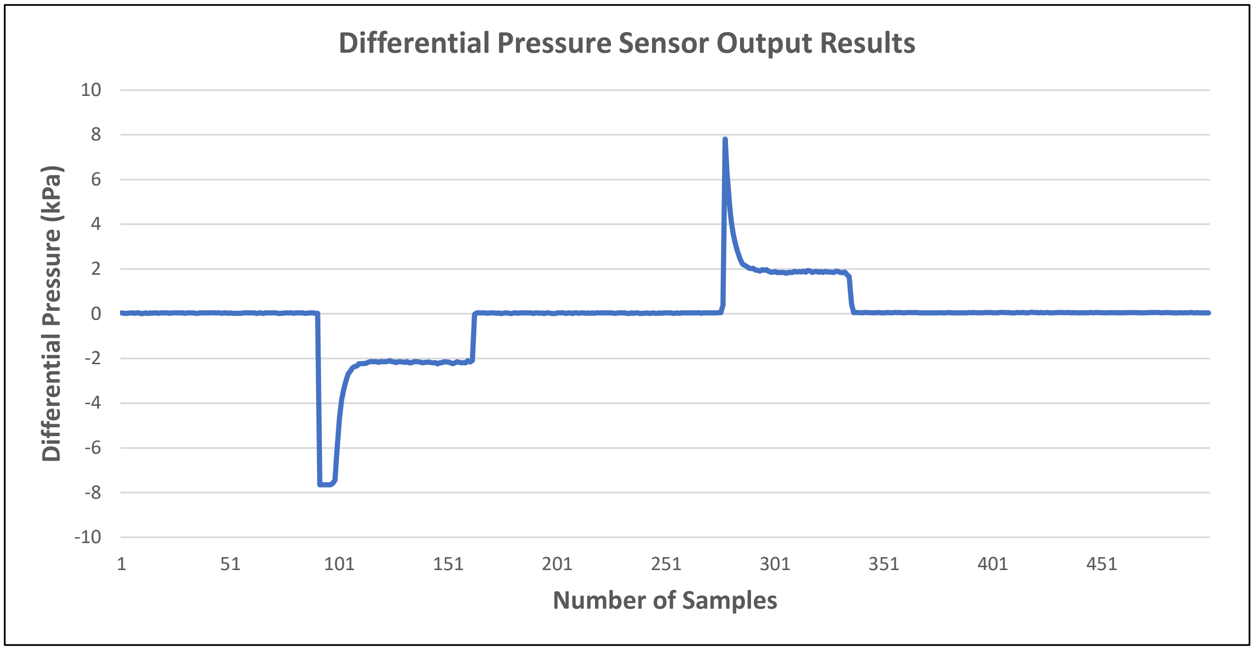


4.4 Differential Air Pressure Sensor Output Results

Figure 4-5 shows the behavior of the differential air pressure sensor in this application. As previously described, this sensor has two physical inputs and measures the difference in air pressure between the two input ports. The compensation routine used is written to calculate and print the differential air pressure between the two inputs in kilo-pascals (kPa).

In order to demonstrate the sensor response as the pressure changed for both inputs, air pressure is applied to one input at a time to show the output spike in both directions. The first spike illustrates pressure being applied to input one and shows a negative differential pressure. The second spike illustrates pressure being applied to input two and shows a positive differential pressure. The sensor output is around 0 kPa ambient when no additional pressure is applied to either input port and they are left in the same area.

Figure 4-5. MPXV7007DP Integrated Silicon Pressure Sensor Results



5. Conclusion

This application demonstrates analog sensor and measurement tips by creating a sensor net comprised of four different sensors. The sensors in this sensor net are integrated into the application in a modular nature, in both hardware and software, with each sensor treated as a node. The setup and configuration of the 12-bit Analog-to-Digital Converter with Computation (ADCC) and the Operational Amplifier (OPA) modules are covered in detail. The integrated hardware features of the ADCC that make this peripheral ideal for analog sensor acquisition are discussed, as well as the implementation of the OPA module, as both a unity gain buffer and programmable gain amplifier.

For more details about any of the peripherals discussed in this application note, refer to the device data sheet.

View Code Example on GitHub
Click to browse repository

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with

Trademarks

ISBN: 978-1-5224-6217-0

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820