

ADC Measurement Correction and Optimization for MCP19114/5

*Author: Yiwei Xiong
Microchip Technology Inc.*

INTRODUCTION

In today's highly competitive and highly technological world, analog signal measurements and regulation accuracies are ever increasing. The MCP19114/5 Digitally Enhanced Power Analog Synchronous Low-Side Pulse-Width Modulation Controller from Microchip Technology Inc. is an analog controller capable of implementing several switch-mode power supply topologies. This analog device has the distinct advantage of a built-in PIC[®] core microcontroller. The microcontroller can be used to improve overall performance without the penalties of additional cost or device area. This application note provides several explanations complete with firmware examples on how to improve ratiometric and non-ratiometric analog signal measurements.

As a digitally enhanced power analog controller, the MCP19114/5 has a built-in 10-bit Analog-to-Digital Converter (ADC). When using the ADC in practice, several factors will affect the accuracy of the measurement. These factors include noise, offset errors, DNL/INL errors and variations in the ADC reference voltage. These error sources will have some initial errors specified at room temperature (+25°C). Also, the error due to temperature variation of these parameters should not be overlooked. This application note will provide a means of alleviating these errors.

Utilizing factory calibration is advantageous when attempting to compensate for measurement errors. However, reference voltage inaccuracies and temperature drift will directly affect the ADC measurement result and cannot be easily corrected. The unique feature of an integral 8-bit PIC microcontroller allows MCP19114/5 to implement software solutions to help resolve these issues. To address high-accuracy requirements of ADC measurements and the temperature drift issues, two methods are discussed in this application note. Both methods require software coding and hardware configurations. Examples are provided to better understand these methods.

ASSUMPTIONS

This application note assumes the user:

- is familiar with MCP19114/5 digitally enhanced analog controllers;
- has basic knowledge of Analog-to-Digital Converters (ADC);
- has working knowledge of C programming language.

NON-RATIOMETRIC MEASUREMENT CORRECTION

Non-ratiometric measurements are measurements where the signal being measured is not related to the ADC reference. The reference voltage of MCP19114/5 ADC is V_{DD} . Non-ratiometric measurement correction can be achieved by measuring a signal of known accuracy and using this measurement to correct for other signal measurements. This method eliminates the need to consider variations in the ADC reference (V_{DD}). The MCP19114/5 internal signal V_{BGR} (1.23V) is factory trimmed to within 1% and has an overtemperature tolerance of $\pm 2.5\%$. This V_{BGR} signal can be read internally at the MCP19114/5 ADC through configuration of the analog test MUX (refer to registers ABECON and ADCON0). Non-ratiometric correction is a mathematical approach to eliminate the ADC reference (V_{DD}) tolerance errors. This method is presented in [Figure 1](#).

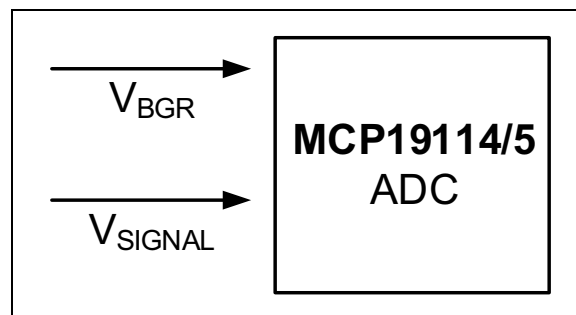


FIGURE 1: Non-ratiometric Measurement Correction.

AN1882

First, use the ADC to measure V_{BGR} . The measured value V_{ADC_BGR} is calculated in [Equation 1](#).

EQUATION 1:

$$V_{ADC_BGR}(counts) = \frac{V_{BGR}(V)}{AV_{DD}(V)} \times 1024$$

Then, use the ADC to measure V_{SIGNAL} . The measured value V_{ADC_SIGNAL} can be represented as shown in [Equation 2](#).

EQUATION 2:

$$V_{ADC_SIGNAL}(counts) = \frac{V_{SIGNAL}(V)}{AV_{DD}(V)} \times 1024$$

Dividing [Equation 1](#) by [Equation 2](#) will cancel $AV_{DD}(V)$, as shown in [Equation 3](#).

EQUATION 3:

$$\frac{V_{ADC_BGR}(counts)}{V_{ADC_SIGNAL}(counts)} = \frac{V_{BGR}(V)}{V_{SIGNAL}(V)}$$

The accuracy impact of AV_{DD} and the drift over temperature is nullified in exchange for the more precise V_{BGR} signal accuracy over temperature. V_{BGR} is factory trimmed to 1.23V \pm 1% (\pm 2.5% over temperature). V_{SIGNAL} can be calculated as follows from measured signals V_{ADC_BGR} and V_{ADC_SIGNAL} by rearranging [Equation 3](#), as shown in [Equation 4](#).

EQUATION 4:

$$V_{SIGNAL}(V) = V_{BGR}(V) \frac{V_{ADC_SIGNAL}(counts)}{V_{ADC_BGR}(counts)}$$

If V_{SIGNAL} threshold detection is desired, rearranging [Equation 4](#) and utilizing the known V_{BGR} value will provide a signal value in counts, as shown in [Equation 5](#).

EQUATION 5:

$$V_{ADC_SIGNAL}(counts) = V_{ADC_BGR}(counts) \frac{V_{SIGNAL}(V)}{V_{BGR}(V)}$$

In [Equation 4](#) and [Equation 5](#), the reference voltage AV_{DD} is replaced by V_{BGR} so the measurement accuracy will incur the tolerance of V_{BGR} .

RATIOMETRIC MEASUREMENT CORRECTION

Another means to reduce measurement error is to implement ratiometric measurements. Typically, a ratiometric ADC measurement is accomplished by utilizing the same voltage source to excite the target circuit as is used for the ADC reference. If a signal level is beyond the ADC maximum measurement range, a resistor divider is applied to lower the input signal with a K-factor. The measured signal is proportional to the reference voltage, thus, the accuracy of the measurement depends on the sensor resistors instead of the reference voltage. In the case of the MCP19114/5, the ADC reference is powered from AV_{DD} (4V) and not the V_{DD} (5V) regulator. The V_{DD} (5V) source is available externally to power-up target circuits. To assist users desiring to make ratiometric ADC measurements with the MCP19114/5, a factory-stored ADC measurement value of V_{DD} is available.

EQUATION 6:

$$K = \frac{R2}{(R1 + R2)}$$

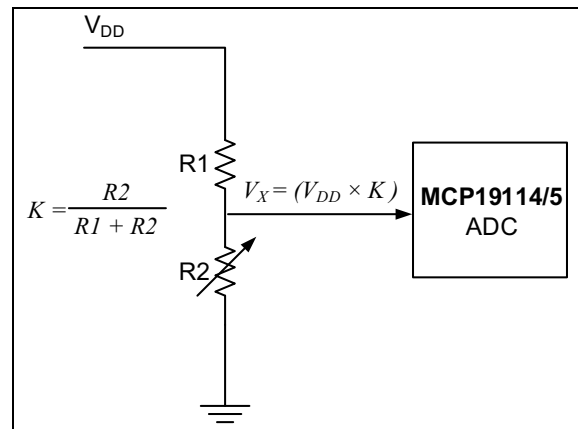


FIGURE 2: Ratiometric Measurement Correction.

Calibration Word 11 stores the internal ADC reading (in counts) of $V_{DD}/2$ or half V_{DD} (HFVDD). This is a factory-stored value at +25°C. This value can be used to calibrate ratiometric ADC measurements powered from V_{DD} . V_{DD} and AV_{DD} temperature drift track to within 1.1% over temperature.

EQUATION 7:

$$HFVDD(\text{counts}) = \frac{V_{DD25^\circ\text{C}} \times \frac{1}{2}}{AV_{DD25^\circ\text{C}}} \times 1024$$

At $x^\circ\text{C}$, the ADC measurement of signal $V_{x^\circ\text{C}}$ is shown in Equation 8.

EQUATION 8:

$$V_{x^\circ\text{C}}(\text{counts}) = \frac{V_{DDx^\circ\text{C}} \times K_{x^\circ\text{C}}}{AV_{DDx^\circ\text{C}}} \times 1024$$

V_{DD} and AV_{DD} voltage drift will track respectively together over temperature. Assume at temperature $x^\circ\text{C}$, V_{DD} and AV_{DD} drift tracks to $y\%$.

EQUATION 9:

$$\begin{aligned} V_{x^\circ\text{C}}(\text{counts}) &= \frac{V_{DD25^\circ\text{C}} \times (1 + y\%) \times K_{x^\circ\text{C}}}{AV_{DD25^\circ\text{C}} \times (1 + y\%)} \times 1024 \\ &= \frac{V_{DD25^\circ\text{C}} \times K_{x^\circ\text{C}}}{AV_{DD25^\circ\text{C}}} \times 1024 \end{aligned}$$

Divide Equation 7 by Equation 9:

EQUATION 10:

$$\frac{HFVDD(\text{counts})}{V_{x^\circ\text{C}}(\text{counts})} = \frac{\frac{1}{2}}{K_{x^\circ\text{C}}}$$

EQUATION 11:

$$K_{x^\circ\text{C}} = \frac{V_{x^\circ\text{C}}(\text{counts})}{HFVDD(\text{counts})} \times \frac{1}{2}$$

Once $K_{x^\circ\text{C}}$ is known, the $R2$ value can be calculated from Equation 6. The reference voltage AV_{DD} is eliminated from the calculation. Since the tracking tolerance is neglected during calculation, a tracking error will be expected in the result. See “ V_{DD} vs. AV_{DD} Temperature Drift Tracking” figure in Section 2.0 Typical Performance Curves of the MCP19114/5 Data Sheet (DS20005281).

VOLTAGE THRESHOLD LEVEL CHECKING USING NON-RATIOMETRIC MEASUREMENT

Systems requiring detection of several operation points or thresholds can implement a comparison table. Using Equation 5, the V_{ADC_SIGNAL} (counts) can be compared to the predetermined table values to perform system operation decisions. The example shown in Figure 3 describes the technique.

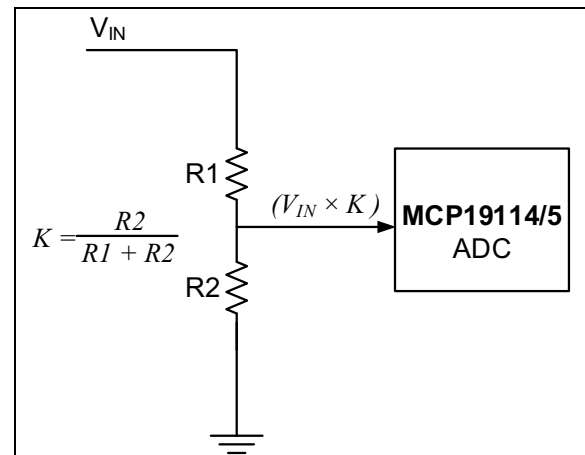


FIGURE 3: Voltage Threshold Level Checking Using Non-ratiometric Measurement.

In Figure 3, assume a system wants to detect three operating thresholds of input voltage V_{IN} :

- $V_{IN\ op1} = 8\text{V}$ (low)
- $V_{IN\ op2} = 12\text{V}$ (normal)
- $V_{IN\ op3} = 18\text{V}$ (high)

Calculate the divider K based on the hardware configuration. In Example 1, the divider K is set to 0.14977.

EXAMPLE 1:

$$K = \frac{R2}{(R1 + R2)} = 0.14977$$

Ratio Reference (R) can be defined as shown in Equation 12

EQUATION 12:

$$R = \frac{V_{SIGNAL}(V)}{V_{BGR}}$$

The relationship between V_{SIGNAL} and V_{IN} is shown in Equation 13.

EQUATION 13:

$$V_{SIGNAL}(V) = V_{IN}(V) \times K$$

AN1882

Calculate the ratio reference of the three operating points $V_{IN\ op1} = 8V$, $V_{IN\ op2} = 12V$ and $V_{IN\ op3} = 18V$, as in [Example 2](#).

EXAMPLE 2:

$$R_{op1\ ref} = \frac{(V_{IN\ op1} \times K)}{V_{BGR}} = \frac{8V \times 0.14977}{1.23V} \approx 0.9741$$
$$R_{op2\ ref} = \frac{(V_{IN\ op2} \times K)}{V_{BGR}} = \frac{12V \times 0.14977}{1.23V} \approx 1.4611$$
$$R_{op3\ ref} = \frac{(V_{IN\ op3} \times K)}{V_{BGR}} = \frac{18V \times 0.14977}{1.23V} \approx 2.1917$$

All calculations need to be accomplished in firmware via the microcontroller. However, in a digital system, numbers (in counts) can only be represented as whole numbers. This means an LSB can only represent 1 or 0. In order to get a 2^{-10} (0.00098) decimal resolution, the numerical representation of $R_{op\ ref}$ is multiplied by 2^{10} , as shown in [Example 3](#).

EXAMPLE 3:

$$R'_{op1\ ref} = R_{op1\ ref} \times 2^{10} \approx 0x03E5$$
$$R'_{op2\ ref} = R_{op2\ ref} \times 2^{10} \approx 0x05D8$$
$$R'_{op3\ ref} = R_{op3\ ref} \times 2^{10} \approx 0x08C4$$

Store the Ratio Reference table values in Program Memory. These values, shown in [Table 1](#), will not change unless there is a hardware change in the resistor divider.

TABLE 1: RATIO REFERENCE TABLE

0x03E5	0x05D8	0x08C4
--------	--------	--------

Part-to-part variations in the V_{BGR} along with temperature drift tolerances require a fresh V_{BGR} ADC measurement before calculating $V_{(ADC_SIG_REF)}$. Configure the ABECON and ADCON0 registers to read V_{BGR} using the ADC (see [Example 4](#)). Read V_{BGR} and store the value. In this example we'll assume the ADC reading of V_{BGR} is 0x013D. To generate the table of ADC readings of these three operating points, substitute [Equation 13](#) with [Equation 5](#).

EQUATION 14:

$$V_{ADC_SIG_REF}^{(counts)} = V_{ADC_BGR}^{(counts)} \times R$$

$$V_{ADC_VIN_REF1} = \frac{R'_{op1\ ref} \times V_{ADC_BGR}}{2^{10} \text{ (Note)}} = \frac{0x03E5 \times 0x013D}{2^{10}} = 0x0134$$

$$V_{ADC_VIN_REF2} = \frac{R'_{op2\ ref} \times V_{ADC_BGR}}{2^{10}} = \frac{0x05D8 \times 0x013D}{2^{10}} = 0x01CF$$

$$V_{ADC_VIN_REF3} = \frac{R'_{op3\ ref} \times V_{ADC_BGR}}{2^{10}} = \frac{0x08C4 \times 0x013D}{2^{10}} = 0x02B6$$

Note: After calculation of $V_{ADC_VIN_REF}$ divide by 2^{10} to shift the value back to a 16-bit word.

[Table 2](#) is the reference checking table.

TABLE 2: REFERENCE CHECKING TABLE

0x0134	0x01CF	0x02B6
--------	--------	--------

Variations in AV_{DD} will result with changes in temperature. The values in [Table 2](#) will also change in accordance with the ADC measurements of V_{BGR} . Since the drift of V_{BGR} with temperature is very small, AV_{DD} 's temperature errors can be accounted for with updated measurements of V_{BGR} for [Table 2](#).

Measure V_{SIGNAL} with the ADC:

- If the ADC reading is smaller than 0x0134, V_{IN} is below 8V.
- If ADC reading is the same as 0x0134, V_{IN} equals 8V.
- If the ADC reading is larger than 0x0134, V_{IN} is higher than 8V.

The same applies to 12V and 18V. Bench test results are shown in [Table 3](#).

TABLE 3: BENCH TEST RESULTS

Calculated Reference Checking Table 2		Bench Test			
		V_{IN} Applied (V)	ADC Reading of V_{IN}	V_{IN} Represented (V)	Error
VADC_8V_REF	0134	8	0133	7.953	1 LSB
VADC_12V_REF	01CF	12	01CE	11.969	1 LSB
VADC_18V_REF	02B6	18	02B7	18.006	1 LSB

EXAMPLE 4: VOLTAGE THRESHOLD LEVEL CHECKING CODE EXAMPLE

```
ADCON0 = 0x0D;           // ADC channel set to VBGR
unsigned int adc;         // VBGR ADC Reading routine
adc = 0;
__delay(10);
for (unsigned char i = 16; i > 0; i--)
{
    ADCON0bits.GO_nDONE = 1;    // Start ADC Reading
    while(ADCON0bits.GO_nDONE);
    adc += ADRES;
}
adc >>= 4;                // Average by 16, reduce noise
unsigned int VBGRREF;
VBGRREF = adc - A2DOFFSET; // Subtract the typ ADC offset and Store ADC reading of
                          // VBGR As reference

/** TABLE 1 **/
#define Ratio1ref (unsigned int) 0x0134 // When Vin = 8V   Ratio1ref = 0x0134
#define Ratio2ref (unsigned int) 0x01CF // When Vin = 12V  Ratio2ref = 0x01CF
#define Ratio3ref (unsigned int) 0x02B6 // When Vin = 18V  Ratio3ref = 0x02B6
/** Generate/Update Table 2 for ADC reading Reference **/
unsigned long RIN_ADCTemp = (unsigned long) Ratio1ref * VBGRREF;
unsigned int Vref1op_ADC = (unsigned int)( RIN_ADCTemp >> 10); // Calculate Vadc_8V_ref ADC Reference
RIN_ADCTemp = (unsigned long) Ratio2ref * VBGRREF;
unsigned int Vref2op_ADC = (unsigned int)( RIN_ADCTemp >> 10); // Calculate Vadc_12V_ref ADC Reference
RIN_ADCTemp = (unsigned long) Ratio3ref * VBGRREF;
unsigned int Vref3op_ADC = (unsigned int)( RIN_ADCTemp >> 10); // Calculate Vadc_18V_ref ADC Reference
/** ADC READING VIN **/
TRISGPAbits.TRISA0 = 1;    // Set GPA0 as input
ANSELAbits.ANSA0 = 1;     // Analog Selected
__delay_ms(30);
__nop();
ADCON0 = 0x61;            // ADC channel set to GPA0
__delay_ms(30);
adc = 0;                  // Clear ADC
for (unsigned char i = 16; i > 0; i--) {
    ADCON0bits.GO_nDONE = 1;
    while(ADCON0bits.GO_nDONE);
    adc += ADRES;
}
adc >>= 4;                // Average by 16 reduce noise error
unsigned int VIN = adc;    // IF VIN = Vadc_8V_ref, Vin equals to 8V,
```

DIRECT MEASUREMENT USING NON-RATIOMETRIC MEASUREMENT

It may be preferable to read the V_{SIGNAL} directly instead of comparing the measurement to a threshold. Equation 4 is applied to correct the V_{SIGNAL} measurement value.

Equation 15 shows the definition of variable M.

EQUATION 15:

$$M = \frac{V_{\text{ADC_SIGNAL}}(\text{counts})}{V_{\text{ADC_BGR}}(\text{counts})}$$

Substituting M in Equation 4 results in Equation 16.

EQUATION 16:

$$\begin{aligned} V_{\text{SIGNAL}}(V) &= V_{\text{BGR}}(V) \times M \\ &= 1.23V \times M \end{aligned}$$

Users have the option of reporting the value M to a higher system to avoid performing local multiplication and consuming processor resources. The signal V_{SIGNAL} can be calculated with Equation 16. In order to get a 2^{-10} (0.00098) decimal resolution, $V_{\text{ADC_SIGNAL}}$ needs to be multiplied by 2^{10} prior to dividing by $V_{\text{ADC_BGR}}$.

Equation 17 shows the definition of variable M'.

EQUATION 17:

$$M' = \frac{V_{\text{ADC_SIGNAL}}(\text{counts}) \times 2^{10}}{V_{\text{ADC_BGR}}(\text{counts})}$$

TABLE 4: BENCH TEST RESULTS

V_{IN} Applied (V)	M'_{ideal} (counts)	M'_{Measured} (counts)	V_{IN} Represented (V)	Error (%)
8	03E5	03E2	7.972	-0.348
9	0462	0460	8.983	-0.191
10	04DE	04DB	9.969	-0.308
11	055B	0559	10.980	-0.184
12	05D8	05D9	12.006	0.053
13	0654	0655	13.001	0.007
14	06D1	06D3	14.011	0.082
15	074E	0751	15.022	0.147
16	07CA	07CF	16.033	0.204
17	0847	084D	17.043	0.254
18	08C4	08CC	18.062	0.343

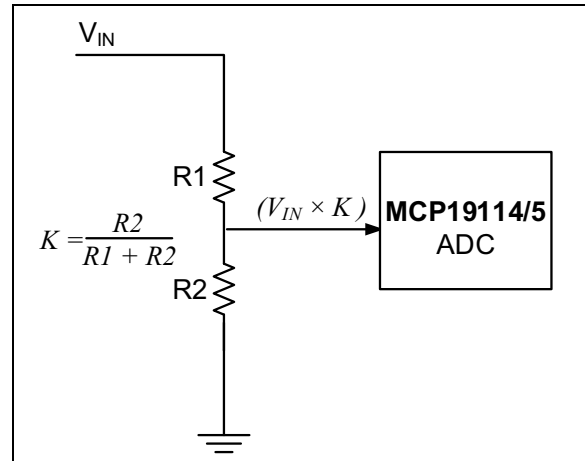


FIGURE 4: Direct Measurement Using Non-ratiometric Measurement.

Apply $V_{\text{IN}} = 12V$, $K = 0.14977$. The ideal M'_{ideal} is shown in Equation 18.

EQUATION 18:

$$\begin{aligned} M'_{\text{ideal}} &= \frac{V_{\text{SIGNAL}}(V) \times 2^{10}}{V_{\text{BGR}}(V)} \\ &= \frac{12(V) \times 0.14977}{1.23(V)} \times 1024 \approx 0x05D8 \end{aligned}$$

AN1882

EXAMPLE 5: DIRECT MEASUREMENT CODE EXAMPLE

```
/** ADC READING VBGR**/  
ADCON0 = 0x0D;           // ADC channel set to VBGR  
unsigned int adc;  
adc = 0;  
_delay(100);  
for (unsigned char i = 16; i > 0; i--)  
{  
    ADCON0bits.GO_DONE = 1;    // Start ADC Reading  
    while(ADCON0bits.GO_DONE);  
    adc += ADRES;  
}  
adc >>= 4;                // Average by 16 reduce noise error  
unsigned int VBGRREF;  
VBGRREF = adc - A2DOFFSET;   //Subtract the offset and Store ADC reading of VBGR As  
                             //reference  
  
/** ADC READING EXAMPLE Vsignal = Vin x K **/  
TRISGPAbits.TRISA0 = 1;     // Example here using GPA0 to read VIN Set GPA0 as input  
ANSELAbits.ANSA0 = 1;      // Analog Selected  
_nop();  
ADCON0 = 0x61;             // ADC channel set to GPA0  
_delay(100);  
adc = 0;                   // Clear ADC  
for (unsigned char i = 16; i > 0; i--) {  
    ADCON0bits.GO_DONE = 1;  
    while(ADCON0bits.GO_DONE);  
    adc += ADRES;  
}  
adc >>= 4;                 // Average by 16 reduce noise error  
unsigned long VIN_ADC = (unsigned long) adc * 1024 ;  
                             // ADC Reading of( Vin x K ) Counts x1024  
unsigned int M = (unsigned long) VIN_ADC / VBGRREF;  
                             // M'= (ADC Reading of( Vin x K) x 1024)/(ADC Reading  
                             // of (V_BGR ))
```

FIRMWARE OPTIMIZATION USING SUCCESSIVE APPROXIMATION AND HARDWARE APPROACH

This section and [Example 6](#) are provided to help optimize firmware efficiency. The successive approximation method saves code space by performing subtraction functions instead of division. In [Equation 15](#), read internal V_{BGR} (assume the ADC reading is 300 counts). Reading V_{SIGNAL} from an external pin (assume the ADC reading is 800 counts) results in [Equation 19](#).

EQUATION 19:

$$M = \frac{800}{300}$$

The result, in decimal system, is shown in [Equation 20](#).

EQUATION 20:

$$\frac{800}{300} \approx 2.66$$

Using the successive approximation approach, the calculation is:

$$\begin{aligned} 800 - 300 &= 500 && \text{count 1;} \\ 500 - 300 &= 200 && \text{count 1;} \end{aligned}$$

When the remainder (200) is smaller than the divider (300), the divider cannot be subtracted from the remainder and result in a positive number. Therefore, the integer part of the quotient is set to 2. To get the tenths unit of the quotient, multiply 200 by 10, then use 2000 to subtract 300 as follows:

$$\begin{aligned} 2000 - 300 &= 1700 && \text{count 1;} \\ 1700 - 300 &= 1400 && \text{count 1;} \\ 1400 - 300 &= 1100 && \text{count 1;} \\ 1100 - 300 &= 800 && \text{count 1;} \\ 800 - 300 &= 500 && \text{count 1;} \\ 500 - 300 &= 200 && \text{count 1;} \end{aligned}$$

Again, 200 is smaller than 300, so the total counts for the tenths unit is $1+1+1+1+1 = 6$. Next, determine the 1/100 unit. Multiply 200 by 10, then subtract.

Finally the successive approximation of M is 2.666...

This successive approximation approach can also be used with the binary system.

$$\begin{aligned} 800 - 300 &= 500 && \text{count 1;} \\ 500 - 300 &= 200 && \text{count 1;} \end{aligned}$$

$200 - 300$ is a negative number, so the total number of counts for integer part is 2, which is $0b10$;

R	Whole Number		Fraction						
	2	1	1/2	1/4	1/8	1/16	1/32	1/64	...
Weight	2	1							
0b	1	0							

Because it is a binary system, instead of multiplying by 10, 200 is multiplied by 2. The counts of the 1/2 unit of 300 is:

$$400 - 300 = 100 \quad \text{count 1;}$$

R	Whole Number		Fraction						
	2	1	1/2	1/4	1/8	1/16	1/32	1/64	...
Weight	2	1							
0b	1	0	1						

Multiply 100 by 2 again and the counts of 1/4 unit of 300 is:

$$200 - 300 \quad \text{negative,} \quad \text{count 0;}$$

R	Whole Number		Fraction						
	2	1	1/2	1/4	1/8	1/16	1/32	1/64	...
Weight	2	1	1/2	1/4	1/8	1/16	1/32	1/64	...
0b	1	0	1	0					

Redo the process as described above.

R	Whole Number		Fraction						
	2	1	1/2	1/4	1/8	1/16	1/32	1/64	...
Weight	2	1	1/2	1/4	1/8	1/16	1/32	1/64	...
0b	1	0	1	0	1				

The resolution depends on the cycles of subtraction. More cycles will generate a more accurate approximation.

R	Whole Number		Fraction						
	2	1	1/2	1/4	1/8	1/16	1/32	1/64	...
Weight	2	1	1/2	1/4	1/8	1/16	1/32	1/64	...
0b	1	0	1	0	1	0	1	0	...

$$M_{(\text{Approximation})} = 1*2 + 0*1 + 1*1/2 + 0*1/4 + 1*1/8 + 0*1/16 + 1*1/32 + 0*1/64 \dots$$

After six cycles, the approximation is:

$$M_{(\text{Approximation})} = 2.66$$

In a binary system:

$$M_{(\text{Approximation})} = 0b10101010$$

AN1882

At this point, since V_{BGR} is known as 1.23V, using it in Equation 16 will result Equation 21:

EQUATION 21:

$$V_{SIGNAL(Approximation)}(V) = V_{BGR}(V) \times M_{Approximation}$$

In this example:

$$V_{SIGNAL(Approximation)}(V) = 1.23 \times 2.66 \approx 3.27V$$

Using the successive approximation approach saves processor resources when compared to division or multiplication functions. One way to eliminate multiplication is to use a special hardware configuration designed for system mathematics convenience. Figure 5 shows an example of how V_{OUT} is measured.

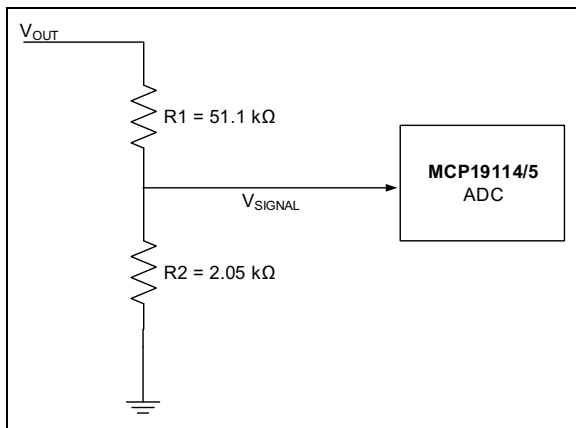


FIGURE 5: V_{OUT} Measurement Use Successive Approximation Approach.

EQUATION 22:

$$V_{SIGNAL} = V_{OUT} \times \frac{R2}{R2 + R1}$$

$R1$ and $R2$ are chosen to set a special value of $R2/R2+R1$, as shown in Equation 23.

EQUATION 23:

$$\frac{R2}{R2 + R1} = \frac{1.23}{2^N}$$

In this example, $R1 = 2.05\Omega$ and $R2 = 51.1\Omega$, thus resulting in Equation 24.

EQUATION 24:

$$V_{SIGNAL} = V_{OUT} \times \frac{2.05}{2.05 + 51.1} \approx V_{OUT} \times \frac{1.23}{32}$$

$$V_{OUT} \approx V_{SIGNAL} \times \frac{32}{1.23}$$

Substituting Equation 16 into Equation 24 results in Equation 25.

EQUATION 25:

$$V_{OUT} \approx V_{BGR}(V) \times M \times \frac{32}{1.23}$$

Since we know $V_{BGR}(V) = 1.23 V$, we can find V_{OUT} using Equation 26.

EQUATION 26:

$$V_{OUT} \approx M \times 32(V)$$

$$\approx M \times 2^5$$

In a digital system, especially when doing a multiplication of 2^N , the result can shift N bits instead of multiplying by 2^N . In this case, the number 32 is chosen based on the applications requirement. At this point, division and multiplication is eliminated from coding calculation.

TEMPERATURE MEASUREMENT USING RATIOMETRIC MEASUREMENT

Temperature measurement is a common application (see [Figure 6](#)). The resistance of a thermistor will change with temperature. Using the manufacturer's specification for resistance versus temperature of a given thermistor, K can be calculated. Once K is established for target temperatures, a temperature reference table can be generated.

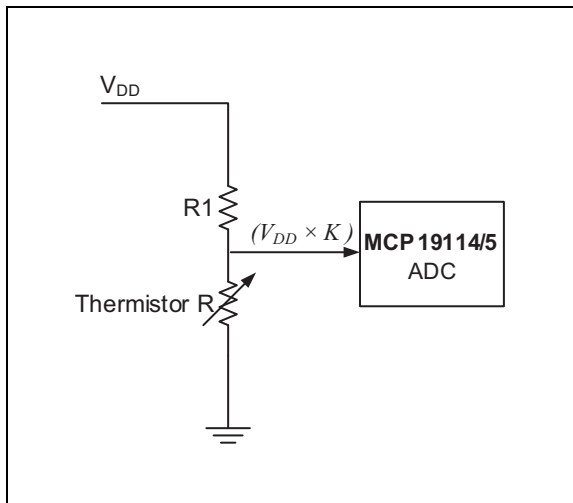


FIGURE 6: Temperature Measurement Using Ratiometric Measurement.

EQUATION 27:

$$K = \frac{\text{Thermistor}R}{(R1 + \text{Thermistor}R)}$$

In [Equation 28](#), use the data from [Equation 27](#).

EQUATION 28:

$$K_{70^{\circ}\text{C}} = \frac{\text{Thermistor}R_{70^{\circ}\text{C}}}{(R1 + \text{Thermistor}R_{70^{\circ}\text{C}})} \approx 0.6622$$

$$K_{100^{\circ}\text{C}} = \frac{\text{Thermistor}R_{100^{\circ}\text{C}}}{(R1 + \text{Thermistor}R_{100^{\circ}\text{C}})} \approx 0.3378$$

Adjust the numerical representation into hexadecimal format, as shown in [Equation 29](#).

EQUATION 29:

$$K'_{70^{\circ}\text{C}} = K_{70^{\circ}\text{C}} \times 1024 \approx 0x02A6 \text{ (in hex)}$$

$$K'_{100^{\circ}\text{C}} = K_{100^{\circ}\text{C}} \times 1024 \approx 0x0159 \text{ (in hex)}$$

This example represents two points:

- Store the values in [Table 5](#). The K' values of +70°C and +100°C are fixed values. They will not change unless there is a hardware change.

TABLE 5: K' NUMERICAL REPRESENTATION

Temperature	K'
+70°C	0x02A6
+100°C	0x0159

- Read from the Calibration Word 11 register, generate the ADC reference [Table 6](#) (as shown by [Equation 30](#)).

EQUATION 30:

$$\begin{aligned} \text{Temp}_{70^{\circ}\text{C ref}} &= K'_1 \times \text{HFVDD} \times 2 \\ &= \frac{0x02A6 \times 0x0290 \times 2 \text{ (Note)}}{1024} \\ &\approx 0x0364 \end{aligned}$$

$$\text{Temp}_{100^{\circ}\text{C ref}} = K'2 \times \text{HFVDD} \times 2 \approx 0x01BA$$

Note: To get a full range of V_{DD} , HFVDD must be multiplied by 2. After calculation of $\text{Temp}_{70^{\circ}\text{C ref}}$, shift back 10 bits to achieve a 16-bit representation.

TABLE 6: ADC REFERENCE TABLE

Temperature	Temp _{REF}
+70°C	0x0364
+100°C	0x01BA

At this point:

- If the ADC reading is larger than 0x0364, the temperature is lower than +70°C;
- If the ADC reading is 0x0364, the temperature is equal to +70°C;
- If the ADC reading is less than 0x0364 but larger than 0x01BA, the temperature is between +70°C and +100°C
- If the ADC reading is 0x01BA, the temperature is equal to +100°C;
- If the ADC reading is less than 0x01BA, the temperature is higher than +100°C;

Increasing the table contents will increase the resolution of the temperature measurement.

AN1882

TABLE 7: BENCH TEST EXAMPLE

Reference Table		Bench Test			
K	K'	Temp.	Temp. ADC reading	K represented	Error (%)
0.6621	01BA	+70°C	01BB	0.663	0.153%
0.3378	0364	+100°C	0366	0.338	-0.044%

EXAMPLE 6: TEMPERATURE MEASUREMENT CODE EXAMPLE

```
/******  
 * Example Vdd with 2 resistors divider 5.546k and 10.87K  
 *  
 * K1 calculation  $10.87 / (5.546 + 10.87) = 0.6622$   
 * Value stored in K1' = trunc (0.6622 * 2^10) = 02A6(in HEX)  
 *  
 * K2 calculation  $5.546 / (5.546 + 10.87) = 0.3378$   
 * Value stored in K2' = trunc (0.3378 * 2^10) = 0159(in HEX)  
 *  
 * Value stored @calibration VDDCAL = 0290 (in HEX)  
 * Calculate T70 REF ADC reference counts = (02A6 * 0290)*2 >> 10 = 0364 (in HEX)  
 * Calculate T100 REF ADC reference counts = (0156 * 0290)*2 >> 10 = 01BA (in HEX)  
 *****/  
#define VDDCAL (unsigned int) 0x0290 // Assuming Read from calibration word VDDCAL  
/*Pre Calculated K Tabel */  
#define K1 (unsigned int) 0x02A6 //K1 = 0.6622  
#define K2 (unsigned int) 0x0159 //K2 = 0.3378  
/*Generate Reference Table */  
unsigned long T70ADCTmp = (unsigned long)Ratio1 * VDDCAL * 2; // Full range of Vdd  
unsigned int T70_REF = (unsigned int)( T70ADCTmp >> 10); // T70_REF ADC reference  
unsigned long T100ADCTmp = (unsigned long)Ratio2 * VDDCAL * 2; // Full range of Vdd  
unsigned int T100_REF = (unsigned int)( T100ADCTmp >> 10); // T100_REF ADC reference  
/*Read from ADC, compare ADC reading with T70_REF and T100_REF reference*/  
/*Users own program*/
```

COMBINATION OF MEASUREMENTS AND OPTIMIZATION EXAMPLE

Figure 7 shows the combination of V_{IN} , V_{OUT} , Temperature and BIN measurements for MCP19114/5.

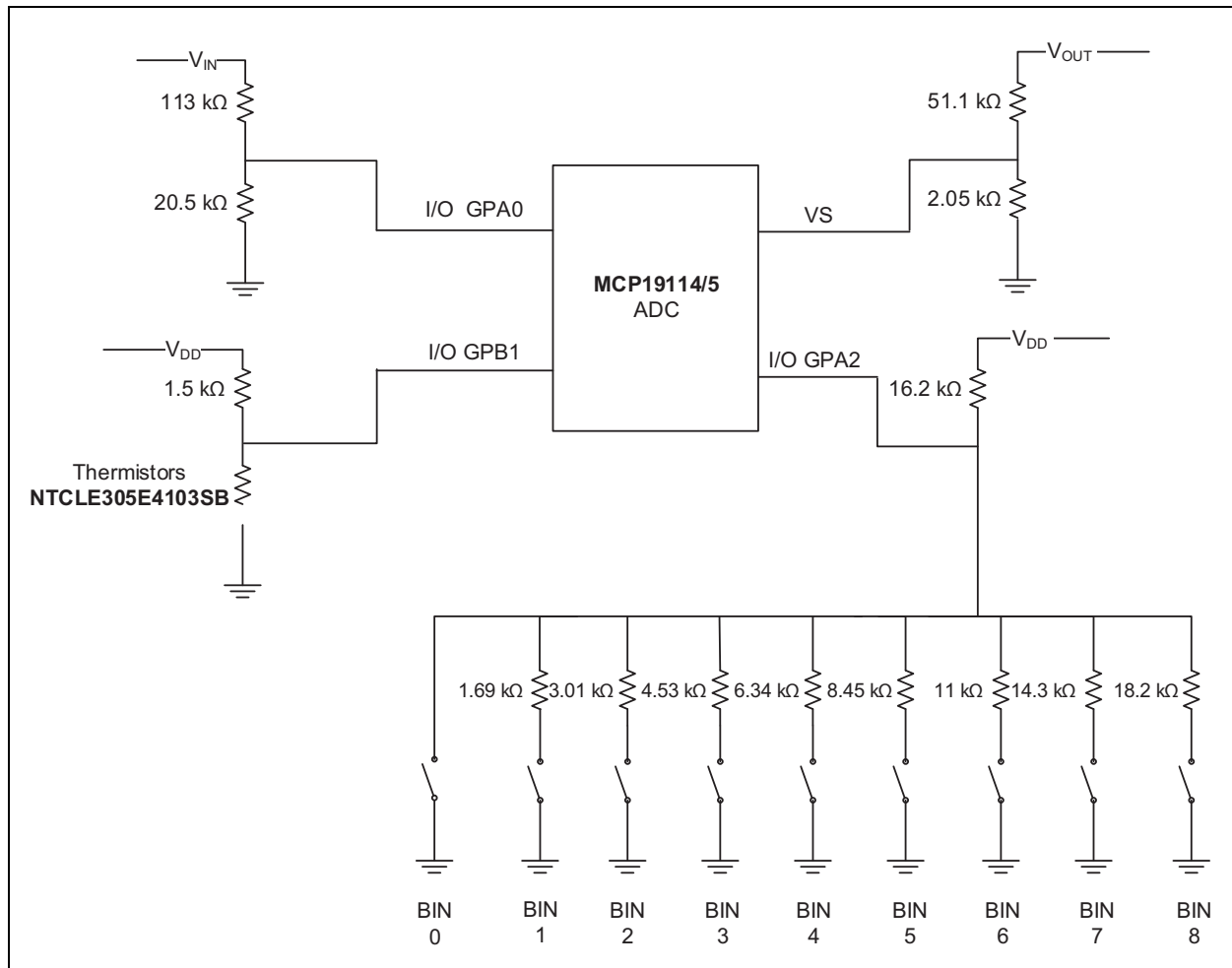


FIGURE 7: Combination of Different Measurements.

In an effort to optimize code space, the example shown in Figure 7 implements only one measurement subroutine to measure all signals.

Thermistor NTCLE305E4103SB has been chosen for this application example. Temperature readings cover the range from +50°C to +125°C with a resolution of +1°C.

V_{IN} measurements cover the range from 0 to 23.4V with a resolution of 0.125V. The resolution can be easily configured via firmware. When increasing the resolution of V_{IN} measurement, users can increase the memory size of 'ref' in the RE_CALC routine using an unsigned int instead of unsigned char. See the details in Example 7.

V_{OUT} measurements cover the range from 0 to 93.6V with a resolution of 0.5V. Resolution can be easily configured via firmware. When increasing the resolution of the V_{OUT} measurement, users can increase the memory size of 'ref' in the RE_CALC routine using an unsigned int instead of unsigned char. See the details in Example 7.

When one of the BIN selecting resistors is connected, the BIN measurement routine will return a number from 0 to 9.

EXAMPLE 7: COMBINATION MEASUREMENTS CODE EXAMPLE

```
int ADCREAD();
int RE_CALCU(unsigned int M, unsigned int R, unsigned char count);
#define HFVDD (unsigned int) 0x0286 // Users can read from calibration word 11
// Here HFVDD defined as 0x0286 is only for
// code explanation
#define A2DOFFSET (unsigned char) 0x03 // Define Typ ADC offset
unsigned char result;

while(1)
{
    /*TEMPERATURE READ ROUTINE*/
    TRISGPBbits.TRISB1 = 1; // Set GPB1 as input
    ANSELBbits.ANSB1 = 1; // Analog Selected
    ADCON0 = 0x71; // ADC channel GPB1 selected
    ADCREAD();
    result = RE_CALCU(ADRES, HFVDD, 7); // Temperature Measurement routine
    if(result < 64)
    {
        TEMPERATURE = 173 - result;
    }
    else
    {
        result = result >> 1;
        TEMPERATURE = 141 - result;
    } // Update Temperature Measurement Result

    /*BING READ ROUTINE*/
    TRISGPAbits.TRISA2 = 1; // Set GPA2 as input
    ANSELAbits.ANSA2 = 1; // Analog Selected
    ADCON0 = 0x69; // ADC channel GPA2 selected
    ADCREAD();
    BIN = RE_CALCU(ADRES, HFVDD, 3); //
    if(BIN >= 9) BIN = 9; // Update BIN Measurement Result

    /* VOUT READ ROUTINE*/
    unsigned int VBGRREF;
    ADCON0 = 0x0D; // ADC channel set to VBGR
    ADCREAD(); // Start ADC reading routine
    VBGRREF = ADRES - A2DOFFSET; // Subtract the offset and Store ADC reading of VBGR
    // As reference
    ADCON0 = 0x11; // ADC channel set to VS(VOUT)
    ADCREAD(); // Start ADC reading routine
    ADRES = ADRES - A2DOFFSET; // Subtract the offset
    VOUT = RE_CALCU(ADRES, VBGRREF, 6); // 6 -> 0.5V resolution,
    // 7 -> 0.25V resolution(RE_CALCU subroutine need
    // define 'ref' as an unsigned int),
    // 8 -> 0.125V resolution >>

    /*VIN READ ROUTINE*/
    TRISAbits.TRISA0 = 1;
    ADCON0 = 0x61; // ADC channel set to GPA1
    ADCREAD();
    VIN = RE_CALCU(ADRES, VBGRREF, 6); // 6 -> 0.125V resolution,
    // 7 -> 0.0625V resolution(RE_CALCU subroutine need
    // define 'ref' as an unsigned int),
```

EXAMPLE 7: CODE EXAMPLE (CONTINUED)

```

    }
}

int ADCREAD()                                // ADC Reading Routine
{
    _delay(10);
    ADCON0bits.GO_nDONE = 1;
    while(ADCON0bits.GO_nDONE);
}

/* Measurement Sub-routine for all measurements */
int RE_CALC(U( unsigned int M, unsigned int R, unsigned char count)
{
    unsigned char ref;
    ref = 0;
    while( M >= R )
    {
        if( M >= R)
        {
            M-=R;
            ref++;
        }
    }
    while(count)
    {
        ref = ref << 1;
        M = M << 1;
        if( M >= R)
        {
            M-=R;
            ref++;
        }
        count--;
    }
    return(ref);
}

```

CONCLUSION

The MCP19114/5 Digitally Enhanced PWM Controller, with its on-chip microprocessor core, possesses flexibility not common to most Analog Power Controllers. The techniques described can be utilized to assist users to improve the accuracy and performance of their designs, whether they are attempting ratiometric or non-ratiometric analog to digital conversions. With the examples provided, the users can easily create firmware to improve measurement performance without the expense of additional hardware.

AN1882

NOTES:

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.

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 your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, dsPIC, FlashFlex, flexPWR, JukeBlox, KEELOQ, KEELOQ logo, Klear, LANCheck, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC³² logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

The Embedded Control Solutions Company and mTouch are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, BodyCom, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, ECAN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, KlearNet, KlearNet logo, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, RightTouch logo, REAL ICE, SQI, Serial Quad I/O, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademarks of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2015, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-63277-085-1

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949 ==

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.



MICROCHIP

Worldwide Sales and Service

AMERICAS

Corporate Office

2355 West Chandler Blvd.
Chandler, AZ 85224-6199

Tel: 480-792-7200

Fax: 480-792-7277

Technical Support:

[http://www.microchip.com/
support](http://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

Cleveland

Independence, OH

Tel: 216-447-0464

Fax: 216-447-0643

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

Los Angeles

Mission Viejo, CA

Tel: 949-462-9523

Fax: 949-462-9608

New York, NY

Tel: 631-435-6000

San Jose, CA

Tel: 408-735-9110

Canada - Toronto

Tel: 905-673-0699

Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office

Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong

Tel: 852-2943-5100

Fax: 852-2401-3431

Australia - Sydney

Tel: 61-2-9868-6733

Fax: 61-2-9868-6755

China - Beijing

Tel: 86-10-8569-7000

Fax: 86-10-8528-2104

China - Chengdu

Tel: 86-28-8665-5511

Fax: 86-28-8665-7889

China - Chongqing

Tel: 86-23-8980-9588

Fax: 86-23-8980-9500

China - Dongguan

Tel: 86-769-8702-9880

China - Hangzhou

Tel: 86-571-8792-8115

Fax: 86-571-8792-8116

China - Hong Kong SAR

Tel: 852-2943-5100

Fax: 852-2401-3431

China - Nanjing

Tel: 86-25-8473-2460

Fax: 86-25-8473-2470

China - Qingdao

Tel: 86-532-8502-7355

Fax: 86-532-8502-7205

China - Shanghai

Tel: 86-21-5407-5533

Fax: 86-21-5407-5066

China - Shenyang

Tel: 86-24-2334-2829

Fax: 86-24-2334-2393

China - Shenzhen

Tel: 86-755-8864-2200

Fax: 86-755-8203-1760

China - Wuhan

Tel: 86-27-5980-5300

Fax: 86-27-5980-5118

China - Xian

Tel: 86-29-8833-7252

Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen

Tel: 86-592-2388138

Fax: 86-592-2388130

China - Zhuhai

Tel: 86-756-3210040

Fax: 86-756-3210049

India - Bangalore

Tel: 91-80-3090-4444

Fax: 91-80-3090-4123

India - New Delhi

Tel: 91-11-4160-8631

Fax: 91-11-4160-8632

India - Pune

Tel: 91-20-3019-1500

Japan - Osaka

Tel: 81-6-6152-7160

Fax: 81-6-6152-9310

Japan - Tokyo

Tel: 81-3-6880-3770

Fax: 81-3-6880-3771

Korea - Daegu

Tel: 82-53-744-4301

Fax: 82-53-744-4302

Korea - Seoul

Tel: 82-2-554-7200

Fax: 82-2-558-5932 or

82-2-558-5934

Malaysia - Kuala Lumpur

Tel: 60-3-6201-9857

Fax: 60-3-6201-9859

Malaysia - Penang

Tel: 60-4-227-8870

Fax: 60-4-227-4068

Philippines - Manila

Tel: 63-2-634-9065

Fax: 63-2-634-9069

Singapore

Tel: 65-6334-8870

Fax: 65-6334-8850

Taiwan - Hsin Chu

Tel: 886-3-5778-366

Fax: 886-3-5770-955

Taiwan - Kaohsiung

Tel: 886-7-213-7828

Taiwan - Taipei

Tel: 886-2-2508-8600

Fax: 886-2-2508-0102

Thailand - Bangkok

Tel: 66-2-694-1351

Fax: 66-2-694-1350

EUROPE

Austria - Wels

Tel: 43-7242-2244-39

Fax: 43-7242-2244-393

Denmark - Copenhagen

Tel: 45-4450-2828

Fax: 45-4485-2829

France - Paris

Tel: 33-1-69-53-63-20

Fax: 33-1-69-30-90-79

Germany - Dusseldorf

Tel: 49-2129-3766400

Germany - Munich

Tel: 49-89-627-144-0

Fax: 49-89-627-144-44

Germany - Pforzheim

Tel: 49-7231-424750

Italy - Milan

Tel: 39-0331-742611

Fax: 39-0331-466781

Italy - Venice

Tel: 39-049-7625286

Netherlands - Drunen

Tel: 31-416-690399

Fax: 31-416-690340

Poland - Warsaw

Tel: 48-22-3325737

Spain - Madrid

Tel: 34-91-708-08-90

Fax: 34-91-708-08-91

Sweden - Stockholm

Tel: 46-8-5090-4654

UK - Wokingham

Tel: 44-118-921-5800

Fax: 44-118-921-5820

01/27/15