

Introduction

This document explains how to use the Microchip Metrology library, and it is considered as a complement of the PIC32CXMTx Metrology Reference Guide (*DS50003461*). It describes the calibration procedure as well as other complex features of the library, such as the waveform capture.

For additional information, refer to the full description of the registers of the “PIC32CXMTx Metrology Reference Guide (*DS50003461*)”.

Table of Contents

Introduction.....	1
1. Configuration.....	3
1.1. Current Conversion Factor.....	3
1.2. Voltage Conversion Factor.....	4
1.3. Sag, Swell, and Interrupt.....	4
1.4. Pulse Test Constant.....	5
1.5. CF Pulses.....	5
1.6. Creep (Start-up/No load).....	6
1.7. AFE.....	7
1.8. Other Features.....	8
2. Calibration.....	9
2.1. Magnitude.....	9
2.2. Phase Delay.....	10
2.3. Power Offset.....	14
2.4. Application Sequence of Power Offsets and Creeps.....	15
2.5. Conclusions.....	15
3. Metrology Firmware Special Function.....	16
3.1. Waveform Capture.....	16
3.2. Harmonic Analysis.....	17
4. Monitor Metrology Firmware.....	18
5. Revision History.....	19
Microchip Information.....	20
Trademarks.....	20
Legal Notice.....	20
Microchip Devices Code Protection Feature.....	20
Product Page Links.....	21

1. Configuration

1.1. Current Conversion Factor

1.1.1. Current Transformer (CT)

$$K_{Ix} = \frac{K_{coil}}{G_{PGA} \times R_{LOAD}}$$

Where:

K_{coil} is the CT transformer ratio

G_{PGA} is the ADC current channel gain

R_{LOAD} is the total burden resistance

Example:

If $K_{coil} = 1000$, $G_{PGA} = 1$ and $R_{LOAD} = 3.24$ ohm, then the result is:

$$K_{Ix} = 1000 / (1 \times 3.24) = 308.642$$

The metrology firmware uses uQ22.10 to represent both integer and fractional numbers; therefore, the value to set is:

$$K_{Ix} = \text{ROUND}(308.642 \times 2^{10}) = 0x0004D291$$

1.1.2. Resistive Shunt

$$K_{Ix} = \frac{1}{G_{PGA} \times R_{shunt}}$$

Example:

If $R_{shunt} = 100$ uOhm = 0.0001 ohm and $G_{PGA} = 8$, then the result is:

$$K_{Ix} = 1 / (8 \times 0.0001) = 1250$$

The metrology firmware uses uQ22.10 to represent both integer and fractional numbers; therefore, the value to set is:

$$K_{Ix} = \text{ROUND}(1250 \times 2^{10}) = 0x00138800$$

1.1.3. Rogowski Coil

$$K_{Ix} = \frac{K_{Rogowski} (60Hz)}{G_{PGA}}$$

Where:

60 Hz is the Rogowski coil scale factor at 60 Hz and it represents the coil sensitivity, measured in millivolts per ampere (mV/A) at 60 Hz.

Note: The scale factor $K_{Rogowski}$ is defined at the reference frequency $f_r = 60$ Hz because the Digital Integrator (DI) filter was designed for unity gain at 60 Hz. If it is desired to use $K_{Rogowski}$ defined at another frequency, the gain of the DI filter must be accounted for in the computation as follows:

$$K_{Ix} = \frac{K_{Rogowski} (f)}{G_{PGA} \times K_{di}(f)}$$

Where:

$$K_{di(f)} = 60 \text{ Hz/f}$$

For example, a Rogowski coil is used with an output of 500 uV/A at 60 Hz (or 416.6 uV/A at 50 Hz) and a $G_{PGA} = 4$, then,

$$K_{Rogowski}(60 \text{ Hz}) = 1000000 / 500 = 2000$$

$$K_{Ix} = 2000 / 4 = 500$$

The metrology firmware uses uQ22.10 to represent both integer and fractional numbers; therefore, the value to set is:

$$K_{Ix} = \text{ROUND}(500 * 2^{10}) = 0x007D000$$

Note: The METER_TYPE control register must be configured correctly: 0x00000CCC for CT, 0x00000DDD for SHUNT, and 0x00000EEE for Rogowski coil.

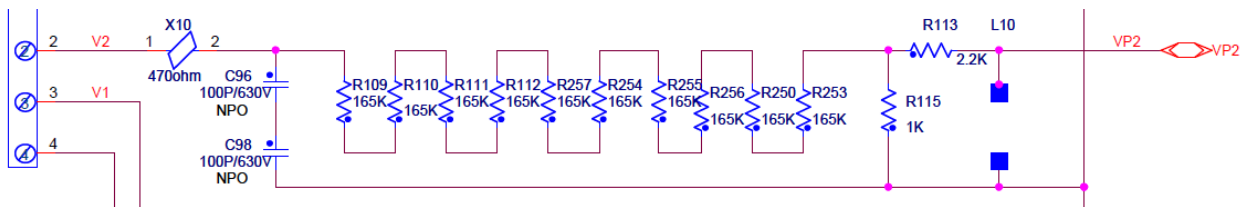
1.2. Voltage Conversion Factor

K_{Vx} is the voltage divider ratio.

Example:

Voltage sensor inputs circuit is illustrated in the figure below.

Figure 1-1. Voltage Sensors Input Circuit



$$K_{Vx} = \frac{R109 + R110 + R111 + R112 + R257 + R254 + R255 + R256 + R250 + R253 + R115}{R115} = 1651$$

The metrology firmware uses uQ22.10 to represent both integer and fractional numbers; therefore, it is recommended this to be set:

$$K_{Vx} = \text{ROUND}(1651 * 2^{10}) = 0x0019CC00$$

1.3. Sag, Swell, and Interrupt

The metrology firmware supports respective sag, swell, interrupt thresholds for all voltage channels x (where x = [A, B, C]):

$$SAG_THRESHOLD_Vx = \left(\frac{V_{SAG-RMS}}{K_{Vx}} \right)^2 \times 2^{32}$$

$$SWELL_THRESHOLD_Vx = \left(\frac{V_{SWELL-RMS}}{K_{Vx}} \right)^2 \times 2^{32}$$

$$INTERRUPT_THRESHOLD_Vx = \left(\frac{V_{INTERRUPT-RMS}}{K_{Vx}} \right)^2 \times 2^{32}$$

There are respective flags SAG_DET_Vx, SWELL_DET_Vx, PH_X_ACTIVE to indicate the sag, swell, interrupt status for every voltage channels. The sag and swell flags are updated every half cycle and the interrupt flags are updated every full cycle.

Example:

If $V_{SAG-RMS} = 220 * 0.8 = 176V$ and $K_{Vx} = 1651$, then the result is:

$$SAG_THRESHOLD_Vx = ((220 * 0.8 / 1651)^2) * 2^{32} = 0x02E8C044$$

If $V_{SWELL-RMS} = 220 * 1.2 = 264V$ and $K_{Vx} = 1651$, then the result is:

$$SWELL_THRESHOLD_Vx = ((220 * 1.2 / 1651)^2) * 2^{32} = 0x068BB09A$$

If $V_{INTERRUPT-RMS} = 220 * 0.5 = 110V$ and $K_{Vx} = 1651$, then the result is:

$$INTERRUPT_THRESHOLD_Vx = ((220 * 0.5 / 1651)^2) * 2^{32} = 0x0122EB1B$$

1.4. Pulse Test Constant

The metrology library version v03.04.00 introduces significant updates to the configuration of pulse test constants. Previously, four registers were used to set pulse test constants:

- P_K_t: Active energy pulse generator constant
- Q_K_t: Reactive energy pulse generator constant
- I_K_t: Current energy pulse generator constant
- S_K_t: Apparent energy pulse generator constant

In version v03.04.00, these registers have been replaced with the following:

- PULSE0_K_t: Pulse test constant for PULSE 0
- PULSE1_K_t: Pulse test constant for PULSE 1
- PULSE2_K_t: Pulse test constant for PULSE 2

The units for PULSE0_K_t, PULSE1_K_t, and PULSE2_K_t are determined by the energy type configuration of the corresponding pulse, which is set through the PULSE0_CTRL, PULSE1_CTRL, and PULSE2_CTRL control registers:

- [Wh/pulse] for active energy
- [VARh/pulse] for reactive energy
- [VAh/pulse] for apparent energy
- [Amp²-h/pulse] for current energy

This new register structure enables the configuration of multiple pulses with the same energy type, each with distinct pulse constants, providing greater flexibility in system setup.

Refer to the “PIC32CXMTx Metrology Reference Guide (DS50003461)” for a detailed description.

1.5. CF Pulses

The metrology firmware can output the following pulses simultaneously, and they are controlled by PULSE0_CTRL, PULSE1_CTRL, and PULSE2_CTRL:

- PULSE0_CTRL controls the pulse output from PD17
- PULSE1_CTRL controls the pulse output from PD18
- PULSE2_CTRL controls the pulse output from PD19

The pulse module supports the following features: ENABLE, DETENT, POLARITY, OVERRIDE, ACC_HOLD, TYPE, and WIDTH.

All features can be changed at run time except for the POLARITY feature. If POLARITY changed, it will take effect only after the metrology is reset.

Refer to the “PIC32CXMTx Metrology Reference Guide (DS50003461)” for a detailed description.

Example:

Enable the first pulse and set DETENT = DETENT_ABSOLUTE, negative polarity, pulse type = TYPE_P_T_F (total active energy), and the pulse width is 40 ms.

Then, it is recommended to set PULSE_CTRL0 = 0x90014880.

1.6. Creep (Start-up/No load)

The metrology firmware includes support for start-up and no load (creep) detection features. The following creep thresholds are available:

- Global creeps, affecting the pulses and the pulse accumulators ACC_Tx:
 - CREEP_THRESHOLD_I
 - CREEP_THRESHOLD_P
 - CREEP_THRESHOLD_Q
 - CREEP_THRESHOLD_S
- Per-phase creeps, affecting the pulses and the pulse accumulators ACC_Tx, and additionally the integration period and per-cycle accumulators:
 - CREEP_THRESHOLD_I_A
 - CREEP_THRESHOLD_I_B
 - CREEP_THRESHOLD_I_C
 - CREEP_THRESHOLD_P_A
 - CREEP_THRESHOLD_P_B
 - CREEP_THRESHOLD_P_C
 - CREEP_THRESHOLD_Q_A
 - CREEP_THRESHOLD_Q_B
 - CREEP_THRESHOLD_Q_C

Current creep:

Energy measurements (I²-hour, Watt-hour, Var-hour, VA-hour) for any phase x (x = A, B, or C) will not be accumulated if the RMS current value is below the global threshold, CREEP_THRESHOLD_I. The individual thresholds enable the configuration of distinct creep thresholds for each phase, allowing for more precise control and customization.

Active power creep:

The total active energy is accumulated only when the active energy in a full cycle is greater than CREEP_THRESHOLD_P.

Similarly, active energy for each individual phase (A, B, or C) is accumulated only when the phase's active energy in a full cycle exceeds its respective threshold, CREEP_THRESHOLD_P_x.

Reactive power creep:

The total reactive energy is accumulated only when the reactive energy in a full cycle is greater than CREEP_THRESHOLD_Q.

Similarly, reactive energy for each individual phase (A, B, or C) is accumulated only when the phase's reactive energy in a full cycle exceeds its respective threshold, CREEP_THRESHOLD_Q_x.

Apparent power creep:

The total apparent energy is accumulated only when the apparent energy in a full cycle is greater than CREEP_THRESHOLD_S.

The "PIC32CXMTx Metrology Reference Guide (DS50003461)" provides example calculations for each one of the thresholds.

1.7. AFE

When using the PIC32CXMTC, the connected AFE (ATSENSE301, MCP3910, MCP3912, MCP3913, or MCP3914) is selected by the AFE_SELECTION register. When using the PIC32CXMTSH (internal ATSENSE203), this register is not used.

The AFE_CTRL_0 register configures the selected AFE, mainly the PGA gains. This register has a different meaning depending on the selected AFE. For PIC32CXMTC + MCP3910 this register is not used.

The CHANNEL_MATRIX register allows to map each physical channel (0, 1, 2, 3, 4, 5, 6, or 7) to a specific logical channel (I_A, V_A, I_B, V_B, I_C, V_D, I_N, or V_D). When using PIC32CXMTSH or PIC32CXMTC + ATSENSE301, channels 0, 1, 3, and 5 (differential channels) should be used for currents and channels 2, 4, and 6 (single-ended channels) should be used for voltages.

Several configuration examples are given below.

1.7.1. PIC32CXMTSH, PGA = 1, CH1: I_A, CH2: V_A, CH3: I_B, CH4: V_B

AFE_CTRL_0 = 0x00000000

CHANNEL_MATRIX = 0xFFFF3210F

1.7.2. PIC32CXMTSH, PGA = 2, CH1: I_B, CH2: V_A, CH3: I_A, CH4: Not used

AFE_CTRL_0 = 0x00000014

CHANNEL_MATRIX = 0xFFFF012F

1.7.3. PIC32CXMTC + ATSENSE301, PGA = 4, CH0: I_N, CH1: I_A, CH2: V_A, CH3: I_B, CH4: V_B, CH5: I_C, CH6: V_C

AFE_SELECTION = 0x00000000

AFE_CTRL_0 = 0x000000AA

CHANNEL_MATRIX = 0xF5432106

1.7.4. PIC32CXMTC + ATSENSE301, PGA = 8, CH0: I_N, CH1: I_A, CH2: V_A, CH3: I_C, CH4: V_C, CH5: I_B, CH6: V_B

AFE_SELECTION = 0x00000000

AFE_CTRL_0 = 0x000000FF

CHANNEL_MATRIX = 0xF3254106

1.7.5. PIC32CXMTC + 3xMCP3910, CH0: I_A, CH1: V_A, CH2: I_B, CH3: V_B, CH4: V_B, CH5: I_C

AFE_SELECTION = 0x00000001

CHANNEL_MATRIX = 0xFF543210

1.7.6. PIC32CXMTC + 4xMCP3910, CH0: I_A, CH1: V_A, CH2: I_B, CH3: V_B, CH4: V_B, CH5: I_C, CH6: I_C, CH7: V_C

AFE_SELECTION = 0x00000002

CHANNEL_MATRIX = 0x76543210

1.7.7. PIC32CXMTC + MCP3912, PGA = 1, CH0: I_A, CH1: I_B, CH2: V_A, CH3: V_B

AFE_SELECTION = 0x00000003

AFE_CTRL_0 = 0x0B000000 (dithering: on, maximum strength; boost: x1)

CHANNEL_MATRIX = 0xFFFF3120

1.7.8. **PIC32CXMTx + MCP3913, PGA = 2, CH0: I_A, CH1: I_B, CH2: I_C, CH3: V_A, CH4: V_B, CH5: V_C**

AFE_SELECTION = 0x00000004

AFE_CTRL_0 = 0x0E000049 (dithering: on, medium strength; boost: x2)

CHANNEL_MATRIX = 0xFF531420

1.7.9. **PIC32CXMTx + MCP3914, PGA = 4, CH0: I_A, CH1: I_B, CH2: I_C, CH3: I_N, CH4: V_A, CH5: V_B, CH6: V_C, CH7: V_D**

AFE_SELECTION = 0x00000005

AFE_CTRL_0 = 0x05002492 (dithering: on, minimum strength; boost: x0.66)

CHANNEL_MATRIX = 0x75316420

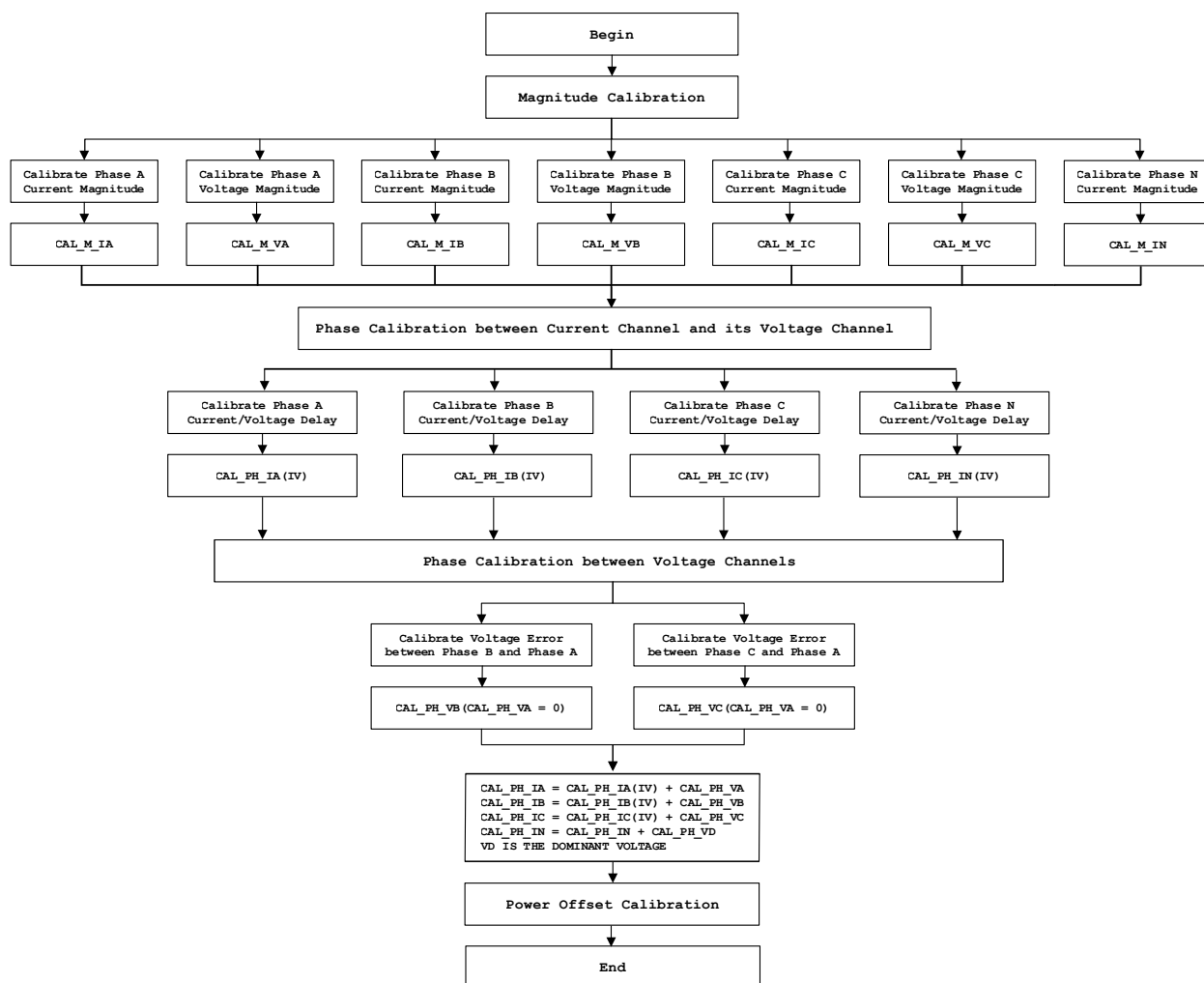
1.8. **Other Features**

Refer to the “PIC32CXMTx Metrology Reference Guide (DS50003461)” for a detailed description of the registers.

2. Calibration

It is recommended to calibrate and verify each meter before delivery. The metrology library can be calibrated after it is configured correctly. The following figure illustrates the metrology firmware calibration flowchart:

Figure 2-1. Firmware Calibration Flowchart with Neutral Current



Since all DSP filters are mathematically correct, once the magnitude and phase have been properly calibrated, both active power and reactive power will be automatically correct.

2.1. Magnitude

CAL_M_Ix (x = A, B, C, N) and CAL_M_Vx (x = A, B, C, D) are used to calibrate the current and voltage magnitudes of phase x. Calibration may be iterative, using prior or old magnitude calibration factors to perform the next calibration cycle, or they may be set to 0x20000000 (unity) by the application prior to calibration.

The following formulas are used to calibrate the magnitude of phase x:

$$CAL_M_Vx_new = CAL_M_Vx_old \times \frac{Vx_RMS_IN}{\frac{K_Vx}{2^{10}} \times \sqrt{\frac{ACC_Vx}{N \times 2^{40}}}}$$

$$CAL_M_Ix_new = CAL_M_Ix_old \times \frac{Ix_RMS_IN}{\frac{K_Ix}{2^{10}} \times \sqrt{\frac{ACC_I_x}{N \times 2^{40}}}}$$

Where,

- $x = [A, B, C, N]$ for CAL_M_Ix and $x = [A, B, C, D]$ for CAL_M_Vx , and:
- Ix_RMS_IN is the input test current, with units [A]
- Vx_RMS_IN is the input test voltage, with units of [V]
- K_Ix is the current conversion factor in uQ22.10
- K_Vx is the voltage conversion factor in uQ22.10
- N is the number of samples in the integration period
- ACC_I_x is the accumulated all harmonics current quantity
- ACC_V_x is the accumulated all harmonics voltage quantity

Example:

An integration period of 1 second is used to gather the following accumulator values calibrate phase-A, with conditions: 120V, 50A, with the old calibrations set to unity (1.0000 = 0x20000000 [sQ2.29]).

- $K_IA = 0x9A522$ [uQ22.10] = 617.283
- $K_VA = 0x19CC00$ [uQ22.10] = 1651
- $N = 4000$ (number of samples in integration period)
- $ACC_I_A = 0x1AF61FD09BBD = 29644398042045$
- $ACC_V_A = 0x144B58A21387 = 22313842119559$

$$\begin{aligned} CAL_M_Ix_new &= CAL_M_Ix_old \times \frac{Ix_RMS_IN}{\frac{K_Ix}{2^{10}} \times \sqrt{\frac{ACC_I_x}{N \times 2^{40}}}} = \\ &= 1.0000 \times \frac{50 \text{ A}}{617.283 \times \sqrt{\frac{29644398042045}{4000 \times 2^{40}}}} = 0.986607 = 0x1F924941 \text{ [sQ2.29]} \end{aligned}$$

$$\begin{aligned} CAL_M_Vx_new &= CAL_M_Vx_old \times \frac{Vx_RMS_IN}{\frac{K_Vx}{2^{10}} \times \sqrt{\frac{ACC_V_x}{N \times 2^{40}}}} = \\ &= 1.0000 \times \frac{120 \text{ V}}{1651 \times \sqrt{\frac{22313842119559}{4000 \times 2^{40}}}} = 1.020415 = 0x20A73DB4 \text{ [sQ2.29]} \end{aligned}$$

Note: The CAL_M_VD register is used for magnitude calibration of the 8th channel of the MCP3914 AFE. This channel is not displayed in the calibration graph, as it is typically reserved for application-specific measurements that are not necessarily related to metrology.

2.2. Phase Delay

CAL_PH_Ix ($x = A, B, C, N$) and CAL_PH_Vx ($x = A, B, C$) are used to align the phase delays between each pair of current and voltage channels, and to optionally equalize the phase delays between voltage channels. All phase shift coefficients used by the metrology DSP must be normalized to equivalent linear-phase shift amounts at 60 Hz.

Phase calibration steps are given below:

1. Phase calibration between current channel and its voltage channel.
2. Phase calibration between voltage channels (optional).

If interphase vector manipulation or optimal phase-phase voltage measurement are not required, then step 2 is optional. If there is no need to make phase calibration between voltage channels, set all CAL_PH_Vx registers to the value: 0x00000000.

However, if interphase vector manipulation is desired or it is desired to achieve the highest accuracy in phase-phase voltage measurements, then it is required to adjust the voltage phase correction coefficients.

Usually, acceptable calibration performance for energy-types of values may be obtained with a single calibration pass using the data from one integration period (usually 1 second or longer) using voltage and current values read directly from a standard or measurement device, with all phase correction coefficients computed simultaneously with the magnitude correction factors from that same set of integration data.

To achieve the highest accuracy in voltage and/or current measurements, it is a common practice to first calibrate the voltage or current channels individually, typically using a pulse reference standard. For example, if all current channels are pulse magnitude calibrated for A²hr, then all the remaining calibration coefficients (voltage magnitude, current phase and voltage phase correction factors) may be determined with a single additional calibration pass using one set of integration data.

2.2.1. Phase Calibration Between Current and Voltage Channels

Phase calibration may be accomplished using whatever current-voltage phase angle the user desires, θ_{VI} . Calibration may be iterative, using prior or old phase calibration factors to perform the next calibration cycle, or they may be set to 0x00000000 (zero) by the application prior to calibration.

2.2.1.1. Calculating Phase Calibration Factor Using Accumulator Values

The following formula is used to calibrate the phase delay of phase x making use of values from the integration period accumulators:

$$CAL_PH_Ix_{IV} = \frac{\left(\arctan\left(\frac{ACC_Q_x_F}{ACC_P_x_F}\right) - \theta_{VI} \right)}{180^\circ} \times \frac{60Hz}{f_line} \times 2^{31}$$

$CAL_PH_Vx = 0x00000000$

$CAL_PH_Ix = CAL_PH_Ix_{IV}$ (if voltage channel phase calibration is not required)

Where:

- ACC_Q_x_F – Reactive power fundamental-only integration accumulator value.
- ACC_P_x_F – Active power fundamental-only integration accumulator value.
- θ_{VI} – Test condition phase angle between voltage and current (+ inductive).
- f_line – The power line frequency used for calibration.

Note: The above calibration equation can be used for manual calibration at any angle, while the auto calibration routines provided in the demonstration application program may be constrained to run at PF = 0.5Lag (60° lagging) condition only.

Note: For calibrating the neutral current phase delay, the fictitious accumulators ACC_Q_x_F and ACC_P_x_F have been added. These are computed using the neutral current and the dominant voltage.

Example:

An integration period of 1 second is used to gather the following accumulator values used for calibration, with conditions: 120V, 30A, PF = 0.5 Lag (50 Hz), K_V = 11557, K_I = 308.642

ACC_P_x_F = 0x33C7784C5B = 222389881947

ACC_Q_x_F = 0x5A1E99C15E = 387060449630

$$CAL_{PH_Ix_{IV}} = \frac{\left(\arctan\left(\frac{Q_x}{P_x}\right) - \theta_{VI}\right)}{180^\circ} \times \frac{60\text{Hz}}{f_{line}} \times 2^{31}$$

$$= \frac{\left(\arctan\left(\frac{387060449630}{222389881947}\right) - 60^\circ\right)}{180^\circ} \times \frac{60\text{Hz}}{50\text{Hz}} \times 2^{31} = 0x001A36E3$$

Note: In the above discussion and example, phase calibration factors are computed for the current channels, regardless of whether voltage-voltage phase correction is computed. When phase calibration between voltage channels is not desired, all other calibration factors (magnitude and current channel phase correction factors) may be computed from accumulator registers from a single integration period. For best results when phase calibration between voltage channels is desired, first calibrate and apply all magnitude gain factors before measuring and calibrating the voltage phase correction values. For additional information, refer to the [Phase Calibration Between Voltage Channels](#).

2.2.1.2. Calculating Phase Calibration Factor Using Pulse Calibration Error

Another method to calibrate the phase delay is using the CF pulse error.

Before calibrating the phase delay, set PF to 0.5 Lag (inductive), CAL_PH_Ix (x = A, B, C, N) and CAL_PH_Vx (x = A, B, C) to 0x00000000.

The following formulas are used to calculate the phase calibration constant of phase x:

$$CAL_{PH_Ix} = -\frac{\text{percent error}}{100 \times \sqrt{3} \times \pi} \times \frac{60\text{Hz}}{f_{line}} \times 2^{31}$$

$$CAL_{PH_Vx} = 0$$

Example:

After magnitude calibration, pulse measurement of a single channel at PF = 0.5 Lag (50 Hz) shows initial channel accuracy of 99.638%, equivalent to a -0.362% error.

$$CAL_{PH_Ix} = -\frac{\text{percent error}}{100 \times \sqrt{3} \times \pi} \times \frac{60\text{Hz}}{f_{line}} \times 2^{31} =$$

$$= -\frac{-0.362\%}{100 \times \sqrt{3} \times \pi} \times \frac{60\text{Hz}}{50\text{Hz}} \times 2^{31} = (6.6527 \times 10^{-4}) \times \frac{6}{5} \times 2^{31} = 0x001A28D4$$

The intermediate quantity (6.6527e-4) provided above, corresponds to a phase shift of +0.12° at 50 Hz, but needs to be normalized to an equivalent linear-phase shift at 60 Hz of +0.144°. All phase shift coefficients used by the metrology DSP must be normalized to equivalent linear-phase shift values at 60 Hz.

Note: When using the pulse method, each phase must be calibrated independently.

2.2.1.3. Calculating Phase Calibration Factor in the Presence of Large Parasitic Reactive Load

Some applications may require calibration in the presence of a noticeable parasitic reactive load. In those applications, optimal calibration will be obtained if that parasitic load can be temporarily removed. However, if the parasitic load cannot be removed during calibration, then the arctan() approach of [Calculating Phase Calibration Factor Using Accumulator Values](#) is not recommended, but rather, it is suggested to set test conditions with a considerable reactive test load, such as using a $\theta_{VI} = 60^\circ$, PF = 0.5, and use the arccos() approach:

$$CAL_{PH_Ix_{IV}} = \frac{\left(\arccos\left(\frac{P_{measured}}{V_{RMS} \times I_{RMS}}\right) - \theta_{VI}\right)}{180^\circ} \times \frac{60\text{Hz}}{f_{line}} \times 2^{31}$$

2.2.2. Phase Calibration Between Voltage Channels

When phase-to-phase vector measurement or manipulation is necessary, it is recommended to correct the phase errors among all voltage channels. One voltage phase, usually Phase-A, is chosen as a reference that all other voltages will be corrected to. The *Law of Cosines* is used to determine voltage phase correction factors using the fundamental only integration accumulator values.

$CAL_PH_VA = 0x00000000$ (reference phase)

The following formulas are used to calibrate the phase errors between the remaining voltage phases:

To phase-calibrate VB (relative to VA):

$$\begin{aligned}
 CAL_PH_VB &= \frac{\left(\arccos\left(\frac{V_{AB}^2 - V_A^2 - V_B^2}{-2 \times V_A \times V_B}\right) - \theta_{VV}^{\circ} \right)}{180^{\circ}} \times \frac{60\text{Hz}}{f_{line}} \times 2^{31} \\
 &= \frac{\left(\arccos\left(\frac{V_{AB_F} - V_{A_F} - V_{B_F}}{-2\sqrt{V_{A_F} \times V_{B_F}}}\right) - \theta_{VV}^{\circ} \right)}{180^{\circ}} \times \frac{60\text{Hz}}{f_{line}} \times 2^{31}
 \end{aligned}$$

Where:

V_A = RMS voltage of Phase-A

V_B = RMS voltage of Phase-B

V_{AB} = Inter-phase RMS voltage between Phase-A and Phase-B

V_{A_F} = Integration period accumulator value for Phase-A voltage (fundamental)

V_{B_F} = Integration period accumulator value for Phase-B voltage (fundamental)

V_{AB_F} = Integration period accumulator value for Phase-A to Phase-B voltage (fundamental)

θ_{VV} = Phase angle between the two voltage phases [in degrees]

Likewise, to phase-calibrate VC (relative to VA):

$$\begin{aligned}
 CAL_PH_VC &= \frac{\left(\theta_{VV}^{\circ} - \arccos\left(\frac{V_{CA}^2 - V_C^2 - V_A^2}{-2 \times V_C \times V_A}\right) \right)}{180^{\circ}} \times \frac{60\text{Hz}}{f_{line}} \times 2^{31} \\
 &= \frac{\left(\theta_{VV}^{\circ} - \arccos\left(\frac{V_{CA_F} - V_{C_F} - V_{A_F}}{-2\sqrt{V_{C_F} \times V_{A_F}}}\right) \right)}{180^{\circ}} \times \frac{60\text{Hz}}{f_{line}} \times 2^{31}
 \end{aligned}$$

Then, the total phase corrections for the current channels become:

$CAL_PH_IA = CAL_PH_IA_{IV} + CAL_PH_VA$

$CAL_PH_IB = CAL_PH_IB_{IV} + CAL_PH_VB$

$CAL_PH_IC = CAL_PH_IC_{IV} + CAL_PH_VC$

Example:

An integration period of 1 second is used to gather the following accumulator values used for calibration, with conditions: 120V, 30A, PF = 0.5 Lag (measured at 50 Hz), $K_V = 11557$, $K_I = 308.642$

- $ACC_V_A_F = 0x144BD23C9BDE = 22315882290142$
- $ACC_V_B_F = 0x1454C5C79B64 = 22354328001380$
- $ACC_V_C_F = 0x144FAD879049 = 22332446314569$
- $ACC_V_AB_F = 0x3CE0EC510A7B = 66936735074939$

- Acc_V_CA_F = 0x 3CFB87060C83 = 67050999778435

$$\begin{aligned}
 CAL_PH_VB &= \frac{\left(\arccos\left(\frac{V_{AB}^2 - V_A^2 - V_B^2}{-2 \times V_A \times V_B}\right) - \theta_{VV}^{\circ} \right)}{180^{\circ}} \times \frac{60\text{Hz}}{f_{line}} \times 2^{31} \\
 &= \frac{\left(\arccos\left(\frac{V_{AB_F} - V_{A_F} - V_{B_F}}{-2\sqrt{V_{A_F} \times V_{B_F}}}\right) - \theta_{VV}^{\circ} \right)}{180^{\circ}} \times \frac{60\text{Hz}}{f_{line}} \times 2^{31} \\
 &= \frac{\left(\arccos\left(\frac{66936735074939 - 22315882290142 - 22354328001380}{-2\sqrt{22315882290142 \times 22354328001380}}\right) - 120^{\circ} \right)}{180^{\circ}} \times \frac{60\text{Hz}}{50\text{Hz}} \times 2^{31} \\
 &= \frac{(\arccos(-0.4984649) - 120^{\circ})}{180^{\circ}} \times \frac{60\text{Hz}}{50\text{Hz}} \times 2^{31} = \frac{(119.89849^{\circ} - 120^{\circ}) \times \frac{6}{5} \times 2^{31}}{180^{\circ}} = 0xFFE9D32D
 \end{aligned}$$

$$\begin{aligned}
 CAL_PH_VC &= \frac{\left(\theta_{VV}^{\circ} - \arccos\left(\frac{V_{CA}^2 - V_C^2 - V_A^2}{-2 \times V_C \times V_A}\right) \right)}{180^{\circ}} \times \frac{60\text{Hz}}{f_{line}} \times 2^{31} \\
 &= \frac{\left(\theta_{VV}^{\circ} - \arccos\left(\frac{V_{CA_F} - V_{C_F} - V_{A_F}}{-2\sqrt{V_{C_F} \times V_{A_F}}}\right) \right)}{180^{\circ}} \times \frac{60\text{Hz}}{f_{line}} \times 2^{31} \\
 &= \frac{\left(120^{\circ} - \arccos\left(\frac{67050999778435 - 22332446314569 - 22315882290142}{-2\sqrt{22332446314569 \times 22315882290142}}\right) \right)}{180^{\circ}} \times \frac{60\text{Hz}}{50\text{Hz}} \times 2^{31} \\
 &= \frac{(120^{\circ} - \arccos(-0.5017584))}{180^{\circ}} \times \frac{60\text{Hz}}{50\text{Hz}} \times 2^{31} = \frac{(120^{\circ} - 120.11640136^{\circ}) \times \frac{6}{5} \times 2^{31}}{180^{\circ}} = 0xFFE6925D
 \end{aligned}$$

Note: For the best results when phase calibration between voltage channels is desired, first calibrate and apply all magnitude gain factors before measuring and calibrating the voltage phase correction values.

2.3. Power Offset

The metrology firmware supports two methods of power offset calibration to solve the power offset issue caused by the voltage cross-talk or chip itself:

- Single value per-cycle offset corrections for P, Q, and S
- Per-phase per-sample offset correction values for P and Q

2.3.1. Single-Value Per-Cycle Power Offset Approach

The three single-value power offset compensation values only affect pulsed outputs (not integration register values) and are named as follows:

- POWER_OFFSET_P is total active energy (Wh) in a full cycle.
- POWER_OFFSET_Q is total reactive energy (Varh) in a full cycle.
- POWER_OFFSET_S is total apparent energy (VAh) in a full cycle.

POWER_OFFSET_P, POWER_OFFSET_Q, and POWER_OFFSET_S are stored in the sQ1.30 format.

The power offset depends on the DETENT type in PULSE_CTRLx:

- Energy = Input Energy – Power Offset, when DETENT is NET, DELIVERED, or GENERATED
- Energy = |Input Energy| – Power Offset, when DETENT is ABSOLUTE

2.3.2. Per-Phase Per-Sample Power Offset Approach

The per-phase power offset compensation values affect both pulsed outputs and integration register values and are named as follows:

- POWER_OFFSET_P_A, POWER_OFFSET_P_B, POWER_OFFSET_P_C
- POWER_OFFSET_Q_A, POWER_OFFSET_Q_B, POWER_OFFSET_Q_C

POWER_OFFSET_P_x is the per-phase active energy offset value in units of [W-samples scaled] with the same scaling as the 64-bit active energy accumulator P_x.

POWER_OFFSET_Q_x is the per-phase reactive quadergy offset value in units of [VAR-samples scaled] with the same scaling as the 64-bit reactive quadergy accumulator Q_x.

Computation of the per-phase POWER_OFFSET_P_x values can be as simple as applying all active voltage phases and zeroing current phases. After integrating over an acceptable period of time (at least 1 second), each 64-bit P_x accumulator is then divided by the number of samples in the integration period and applied to the POWER_OFFSET_P_x register. If voltage magnitude effects are desired, the application processor may scale POWER_OFFSET_P_x as desired and updated.

A similar approach is used for computation of the POWER_OFFSET_Q_x values.

2.3.3. Static vs. Dynamic Power Offsets

If offsets are used for fixed-voltage meter applications, the offsets may be static (computed once and not changed). However, for wide-voltage range meter applications, it is recommended that all power offset values be dynamically changed proportional to the square of the phase voltage (V^2).

2.4. Application Sequence of Power Offsets and Creeps

This section describes the application sequence for CREEP_I, CREEP_I_x, CREEP_P, CREEP_P_x, CREEP_Q, CREEP_Q_x, CREEP_S, POWER_OFFSET_P, POWER_OFFSET_Q, POWER_OFFSET_S, POWER_OFFSET_Px, and POWER_OFFSET_Qx.

The first corrections to be applied are the individual power offsets (POWER_OFFSET_Px and POWER_OFFSET_Qx), which are applied at a 4 kHz rate for each sample.

The remaining corrections are applied each line cycle:

- The corrections affecting the accumulators (CREEP_OFFSET_I_x, CREEP_OFFSET_P_x and CREEP_OFFSET_Q_x) are applied first
- POWER_OFFSET_S (a correction over the RMS current) is applied
- CREEP_I, which affects all pulse sources, is applied
- POWER_OFFSET_P and POWER_OFFSET_Q are applied
- CREEP_P and CREEP_Q are applied

2.5. Conclusions

- After active power is calibrated, reactive power will use the same calibration parameters.
- Three phase voltage and current magnitude can be calibrated at the same time.
- Three phase delays can be calibrated simultaneously by power method.
- Three phase delays can be calibrated independently by error method.
- After the magnitude is calibrated, it will take effect immediately.
- After the phase delay is calibrated, it will take effect at the next integration period.

3. Metrology Firmware Special Function

The metrology firmware provides some special functions which make it more powerful.

3.1. Waveform Capture

Before enabling waveform capture, the registers must be set correctly:

CAPTURE_SOURCE:

0 = Capture 16 kHz data for ADC [sQ1.30]

1 = Capture 4 kHz FBW data [sQ2.29] (Full BandWidth = fundamental + harmonics)

2 = Capture 4 kHz NBW data [sQ2.29] (Narrow BandWidth = fundamental only)

3 = Capture 8 kHz FBW data [sQ2.29] (Full BandWidth = fundamental + harmonics)

CAPTURE_TYPE:

0 = One-shot Capture mode

1 = Continuous Capture modes

One-shot Capture mode is recommended for general purposes; otherwise, the captured data in capture buffer may be not continuous.

CH_SEL_Ix/Vx (x = A, B, C, N for Ix, x = A, B, C, D for Vx): Capture Channel Select

Up to 8 channels of data may be captured at the same time. These registers are used to appoint which channels will be captured.

CAPTURE_ADDR:

This register is used to specify the starting address of the waveform capture buffer and is interpreted as a pointer to the 32-bit unsigned integer. The waveform capture buffer must be allocated in core 0 memory spaces.

CAPTURE_BUFF_SIZE:

This register specifies the buffer size in units of DWORD for capture function. It must be divisible by the number of channels selected. The buffer size must not be larger than the allocated size, otherwise, the application code may be corrupted by a buffer overrun.

After all these registers are set correctly, the user can enable the waveform capture function by setting CAPTURE_EN to 1. Check CAPTURE_STATUS to get capture status.

The following table provides the capture buffer distribution samples:

Channel IA is selected	All channels are selected	Channel IA and IB are selected
CH_IA_0	CH_IA_0	CH_IA_0
CH_IA_1	CH_VA_0	CH_IB_0
CH_IA_2	CH_IB_0	CH_IA_1
CH_IA_3	CH_VB_0	CH_IB_1
CH_IA_4	CH_IC_0	CH_IA_2
CH_IA_5	CH_VC_0	CH_IB_2
CH_IA_6	CH_IN_0	CH_IA_3
CH_IA_7	CH_VD_0	CH_IB_3
CH_IA_8	CH_IA_1	CH_IA_4
—	CH_VA_1	CH_IB_4
—	CH_IB_1	—
—	CH_VB_1	—

Waveform Capture (continued)

Channel IA is selected	All channels are selected	Channel IA and IB are selected
—	CH_IC_1	—
—	CH_VC_1	—
—	CH_IN_1	—
—	CH_VD_1	—
—	CH_IA_2	—
—	CH_VA_2	—
—	CH_IB_2	—

Note: All captured data is after Vref compensation was applied and are scaled values (normalized by the appropriate K_{Ix} and K_{Vx}). In the case of 16 kHz captured data, the calibration constants were not applied.

3.2. Harmonic Analysis

Harmonic analysis control-related registers are allocated in the HARMONIC_CTRL and FEATURE_CTRL.

The HARMONIC_CTRL.HARMONIC_EN (bit 31) enables or disables harmonic computation. The 0 to 30 bits are used to select the specific harmonics to be computed. Accurate results from the analysis are obtained after a complete integration period has elapsed. Therefore, the application must disregard the results generated during the integration period in which the analysis was enabled or the harmonic selection was modified.

The FEATURE_CTRL register allows the selection of channels enabled for harmonic computation. By default, all channels are enabled.



WARNING Although the control register can enable up to 31 harmonic orders, the frequency response may be limited by the metrology filters or the input network anti-aliasing filters. With respect to the metrology filters, the accuracy is better than 0.007% up to 1386 Hz, ensuring an extremely accurate response up to the 21st harmonic order (considering a fundamental frequency of 60 Hz, with a 10% deviation). However, the response remains relatively flat up to 1650 Hz, with an error margin of up to 0.05%.

Note: When enabled, the harmonic analysis result is updated every integration period. The INTEGRATION_IRQ will be generated at the end of the integration period, and this interrupt can be used to update the harmonic analysis result.

4. Monitor Metrology Firmware

It is essential to monitor the operation of the metrology firmware to ensure it is functioning correctly with the application code.

When the metrology is running, these control registers may be changed dynamically, but may induce measurement perturbations if large changes are made: CREEP_x_EN, PULSEx_CTRL, PULSEx_K_t, POWER_OFFSET_CTRL, POWER_OFFSET_P, POWER_OFFSET_Q, POWER_OFFSET_P_x, POWER_OFFSET_Q_x, K_Ix, K_Vx, CAL_M_Ix, CAL_M_Vx, CAL_PH_Ix, CAL_PH_Vx, and CAPTURE control registers.

The following registers will be cleared by metrology firmware automatically when the task is finished: V_MAX_RESET, I_MAX_RESET, and CAPTURE_EN (one-shot mode).

In the default configuration for the provided metrology examples, the following three interrupts will be generated:

- HALF_CYCLE_IRQ – Generated every half line cycle
- CYCLE_IRQ – Generated every full line cycle
- INTEGRATION_IRQ – Generated every integration period

When these interrupts are generated, related status registers and accumulated quantities are already updated and may be copied using the application processor.

It is recommended that the application monitors the metrology library running state. If the metrology firmware flags are abnormal, the application code must reset the metrology firmware and restore control register settings.

5. Revision History

Revision C - 11/2025

The following updates were performed in this revision:

Document	<ul style="list-style-type: none"> Minor changes of format Compatible with Metrology Library version v03.04.00
Configuration	<ul style="list-style-type: none"> Add Interrupt threshold in Sag, Swell, and Interrupt section Updated Pulse Test Constant section Updated Creep (Start-up/No load) section Changed EMAFE section by AFE section and added subsections Removed reference in Other Features section
Calibration	<ul style="list-style-type: none"> Added <i>D</i> in CAL_M_Vx and note in Magnitude section Added <i>CREEP_I_x</i>, <i>CREEP_P_x</i>, <i>CREEP_Q_x</i> and first bullet in Application Sequence of Power Offsets and Creeps section
Metrology Firmware Special Function	<ul style="list-style-type: none"> Added <i>D</i> in CH_SEL_Vx in Waveform Capture section Added <i>CH_VD_0</i> and <i>CH_VD_1</i> in the table of Waveform Capture section
Monitor Metrology Firmware	<ul style="list-style-type: none"> Changed <i>P_K_t</i>, <i>Q_K_t</i>, and <i>I_K_t</i> by <i>PULSEx_K_t</i> in Monitor Metrology Firmware

Revision B - 04/2025

The following updates were performed in this revision:

Document	<ul style="list-style-type: none"> Minor changes of format Compatible with Metrology Library version 3.03.00 Changed <i>Microchip Metrology data sheet</i> by <i>PIC32CXMTx Metrology Reference Guide (DS50003461)</i>
Configuration	<ul style="list-style-type: none"> Added <i>S_K_t</i> in Pulse Test Constant section Added <i>CREEP_THRESHOLD_S</i> in Creep (Start-up/No load) section
Calibration	<ul style="list-style-type: none"> Updated Figure 2-1 Added note in Calculating Phase Calibration Factor Using Accumulator Values section Updated Power Offset section Added <i>POWER_OFFSET_S</i> in Single-Value Per-Cycle Power Offset Approach section Added Application Sequence of Power Offsets and Creeps section
Metrology Firmware Special Function	<ul style="list-style-type: none"> Updated Harmonic Analysis section

Revision A - 12/2022

Document	<ul style="list-style-type: none"> Initial release Metrology library version v3.0.0b
----------	--

Microchip Information

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at <https://www.microchip.com/en-us/about/legal-information/microchip-trademarks>.

ISBN: 979-8-3371-2285-4

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications 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. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. 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 ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP’S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

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 unless otherwise stated.

Microchip Devices Code Protection Feature

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

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Product Page Links

[PIC32CX1025MTC](#), [PIC32CX1025MTSH](#), [PIC32CX2051MTC](#), [PIC32CX2051MTSH](#), [PIC32CX5112MTC](#),
[PIC32CX5112MTSH](#)