

EMDC Calibration Process

NOTE: These instructions only apply to designs using shunts or current transformer (CT) current sensors.

1. Gain calibration process ($\theta_{Applied_{degrees}} = 0^\circ$)

$$voltageSF(new)_{ph} = voltageSF(initial)_{ph} \times \left(\frac{vrmsApplied_{ph}}{vrmsAverage_{ph}} \right)$$

$$currentSF(new)_{ph} = currentSF(initial)_{ph} \times \left(\frac{irmsApplied_{ph}}{irmsAverage_{ph}} \right)$$

Basic mode

This mode can be used for applications that don't require precise active energy measurements. It is equivalent to skipping Step 3 in the EMDC GUI.

$$actApplied_{ph} = vrmsApplied_{ph} \times irmsApplied_{ph} \times \cos(\theta_{Applied_{radians}})$$

$$activePowerSF(new)_{ph} = activePowerSF(initial)_{ph} \times \left(\frac{actApplied_{ph}}{actAverage_{ph}} \right)$$

$$reactivePowerSF(new)_{ph} = activePowerSF(new)_{ph}$$

Advanced mode

This mode is important for applications such as revenue-grade electricity meters that do require precise active energy measurements. The pulses must be enabled in the EMDC code for this mode to work properly with your test equipment. A specific active energy will be applied by the test equipment and the pulses from the MCU will be counted. The test equipment will display an error percentage between what is provided and what is measured. This error is used to adjust the active and reactive power scaling factors and is equivalent to using Step 3 in the EMDC GUI. Compared to Basic mode, Advanced mode allows the active energy error over time to be reduced.

$$activePowerSF(new)_{ph} = activePowerSF(initial)_{ph} \times \left(1 - \frac{actError_{ph}}{100} \right)$$

$$reactivePowerSF(new)_{ph} = activePowerSF(new)_{ph}$$

Continue

2. Phase calibration process ($\theta_{Applied_{degrees}} = 60^\circ$)

Determine the initial total preload unit (TPU) from the whole sample delay (WSD) and fractional sample delay (FSD) from the EMDC-programmed MCU. FSD has units of preload bits. OSR is the ADC oversampling rate and equals the maximum number of preload bits that can be used for fractional sample delays.

$$FSD(initial)_{ph} = phaseCorrection(initial)_{ph} \& (OSR - 1)$$

$$WSD(initial)_{ph} = phaseCorrection(initial)_{ph} \gg 10$$

$$TPU(initial)_{ph} = (WSD(initial)_{ph} \times OSR) + FSD(initial)_{ph}$$

Recalculate $actApplied_{ph}$ since $\theta_{Applied_{degrees}}$ was changed from 0 to 60 degrees.

$$actApplied_{ph} = vrmsApplied_{ph} \times irmsApplied_{ph} \times \cos(\theta_{Applied_{radians}})$$

Basic mode

Calculate the active power error based on the applied active power.

$$actError_{ph} = 100 \times \left(\frac{actAverage_{ph}}{actApplied_{ph}} - 1 \right)$$

Advanced mode

Enter the active power error provided by your test equipment.

$$actError_{ph} = test\ equipment\ error$$

Continue

Calculate the whole sample resolution (WSR) based on the ADC sampling frequency (4kHz) and AC signal frequency (e.g. 60 Hz). One AC cycle is 360 degrees, so WSR indicates how many degrees can be delayed per whole sample. Normally, whole sample delays are done in software by shifting one array index with respect to another array index. The WSR units are degrees.

$$WSR = 360^\circ \times \left(\frac{f_{AC\ signal}}{f_{ADC\ sampling}} \right)$$

Calculate the fractional sample resolution (FSR) based on the WSR and ADC oversampling rate (OSR). The FSR indicates how many degrees an ADC sample can be delayed per preload bit. The maximum number of preload bits supported is equal to the OSR. The FSR units are degrees.

$$FSR = \frac{WSR}{OSR}$$

Calculate the minimum error delta for each FSR unit. This value will be used later to correlate the number of FSR units required to address the total active power error and update the new TPD.

$$errorDelta_{ph} = 100 \times \left(\frac{\cos(\theta_{Applied_{degrees}}) - \cos(\theta_{Applied_{degrees}} + FSR)}{\cos(\theta_{Applied_{degrees}})} \right)$$

Calculate the total preload delta based on the calculated active power error.

$$preloadDelta_{ph} = -(int)round\left(\frac{actError_{ph}}{errorDelta_{ph}}\right)$$

Calculate the new TPU.

$$TPU(new)_{ph} = TPU(initial)_{ph} + preloadDelta_{ph}$$

Make sure the new TPU does not exceed the limits in either direction.

HAL_ADC_V_ARRAY_LENGTH and HAL_ADC_I_ARRAY_LENGTH are the sizes of the voltage and current arrays and are defined in the 'hal_adc.h' file.

$$preloadMin = -OSR \times HAL_ADC_V_ARRAY_LENGTH$$

$$preloadMax = OSR \times HAL_ADC_I_ARRAY_LENGTH$$

$$preloadMin \leq TPU(new)_{ph} \leq preloadMax$$

Extract the new WSD and FSD values from the new TPU.

For TPU values greater than or equal to zero:

$$FSD(new)_{ph} = TPU(new)_{ph} \% OSR$$

$$WSD(new)_{ph} = (int)floor\left(\frac{TPU(new)_{ph}}{OSR}\right)$$

$$TPU(initial)_{ph} = (WSD(initial)_{ph} \times OSR) + FSD(initial)_{ph}$$

For negative TPU values:

$$FSD(new)_{ph} = (TPU(new)_{ph} \% OSR) + OSR$$

$$WSD(new)_{ph} = (int)floor\left(\frac{TPU(new)_{ph}}{OSR}\right)$$

$$TPU(initial)_{ph} = (WSD(initial)_{ph} \times OSR) + FSD(initial)_{ph}$$

Set the new WSD value equal to the upper 6 bits in $phaseCorrection(new)_{ph}$.

$$phaseCorrection(new)_{ph} = WSD(new)_{ph} \ll 10 \& 0xFC00$$

Set the new FSD value in $phaseCorrection(new)_{ph}$.

$$phaseCorrection(new)_{ph} = phaseCorrection(new)_{ph} | FSD(new)_{ph}$$

Update the $phaseCorrection_{ph}$ value on the EMDC-programmed MCU.