NXP Semiconductors
Application Note

Document Number: AN5409
Rev 0, 04/2017

Use Metering Library in Single-Phase Multi-Channel Power Meter

Contents

1 Introduction

The Kinetis-M metering library is used in Kinetis-M series chips and mainly used in metering applications for accurate computation of the active energy, reactive energy, active power, reactive power, apparent power, RMS voltage, and RMS current.

Normally this metering library is used for three-phase four-wire, two-phase two-wire or single-phase one-wire applications. This application will guide to use the library in non-standard single-phase multi-channel metering applications. Furthermore, the implementation described in this document can also be extended to other single-phase non-standard meters.

The Kinetis-M microcontrollers, e.g. KM34Z128 and KM34Z256, can offer a high-performance analog front-end (24-bit AFE) combined with an embedded Programmable Gain Amplifier(PGA). Besides high-performance analog peripherals like auxiliary 16-bit SAR ADC, these devices integrate memory, input-output ports, digital blocks, and a variety of communication options.

Moreover, on KM34Z256, the ARM Cortex-M0+ core, with MMAU (Memory Mapped Arithmetic Unit) support for 32-bit and 64-bit math, enables fast execution of metering algorithms.

There are two metering libraries provided with Kinetis-M chips, Filter-Based library and FFT library. This application will use Filter-Based library.

The Filter-Based Metering Algorithm can be easily integrated into an electronic power meter application. The algorithm requires only instantaneous voltage and current samples to be provided at constant sampling intervals. These instantaneous voltage and current samples are usually measured by an AFE with the help of a resistor divider, in the case of a phase voltage measurement, and a shunt resistor, current transformer or a Rogowski coil in the case of a phase current measurement. All current measurement sensors introduce a phase shift into current measurement. Therefore, it is necessary to align the phases of the instantaneous voltage and current samples prior to their use by the Filter-Based Metering Algorithm using either a digital filter or with the aid of delayed sampling.

2 Hardware

2.1 Hardware introduction

This section will introduce the hardware related to single-phase and multiple current channel power meter. The KM series MCU has 4 SD ADC MAX which is suitable for current measurement in high dynamic range, at the same time it also has one SAR ADC which can be used for voltage measurement.

2.2 Hardware design

For the single-phase multi-channels power meter application, it doesn’t have isolation issue and shunt is used for current sensing.

It always use the L line as the system ground and the load is connected with each shunt, so that current flow through the shunt and providing sampling data for the SD ADC module.

Because of the CPU loading bandwidth, for KM34Z128, it support up to 3 channel current measurement while for KM34Z256 it could support up to 4 channel current measurement. See below connection for schematic design, make sure each current channel flow through the shunt and to the load.

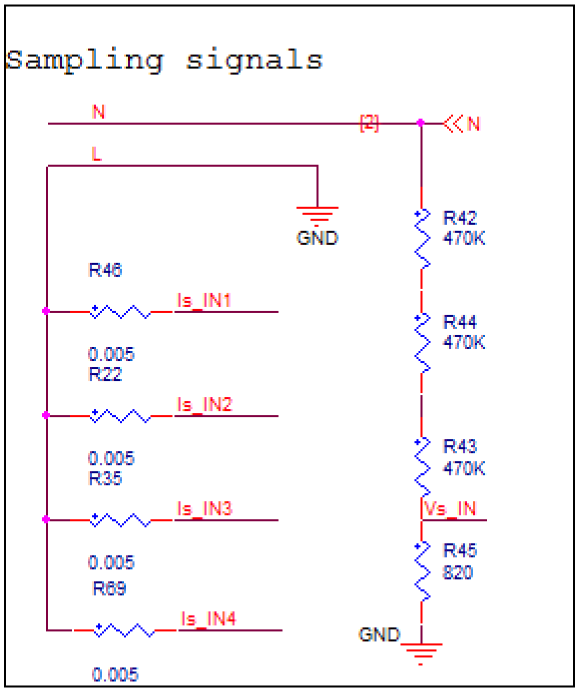

Figure 1. connection for schematic design

2.3 Hardware notes

For PCB design, the current measurement should be wired as differential signals, and note that should be grounded at the samplinig point, otherwise it will couple noise on the sampling signals.

Also it is noted that as the load current is flowing through the shunt, due to the PCB copper width, the load capacity should be considered carefully.

The manganese copper is recommended for using in such case, otherwise during circuit short case, the shunt resistor will be damanged instantly.

For SD ADC, the sampling data should be routed as differential lines and the common ground is at the place to put the shunt. please keep in mind that don't put noisy signals across the those sampling data, otherwise there could be decoupling issue and impact the metering accuracy.

3 Software

3.1 Software introduction

The software supplied with this Application Note is a template which demonstrate the usage of Filter-Based metering library in a single-phase 4-channel power meter. This section will provide a guide in implementation and these changes can be extended to other single-phase non-standard meters.

3.2 Initialization

The commonly-used meter topology is based on the six or seven channels of sigma-delta (SD) ADC converters. The Kinetis-M microcontroller uses different topology as it utilizes the 24-bit AFE (four channels of 24-bit SD ADC) convertors and the 16-bit successive approximation (SAR) ADC converter with input analog multiplexer .

The 3 phase meter implemented on the Kinetis-M device is based on the signals dynamic ranges analysis. The metering current signal is typically from 50 mA to 120 A, therefore the current must be digitalized by very precise and linear ADC with a wide dynamic range, typically 24-bit. The SD type is the ideal solution to solve current dynamic range requirements. The metering voltage signal ranges from 80 V to 280 V. The voltage dynamic range is approximately 60 times smaller than the current dynamic range. The voltage requirements can be solved by the use of a hi-resolution SAR converter.

The initialization section in software initialize AFE and SAR-ADC to sample current and voltage.

AFE initialization for current sampling:

```
      SIM_SelAfePllClk(SIM_MCG_PLL_CLK);
        AFE_ChInit(CH0,
                AFE_CH_SWTRG_CCM_PGAOFF_CONFIG(DEC_OSR2048),
                0,
                PRI_LVL1,
                (AFE_CH_CALLBACK)metering_func_afe_ch0_callback);
    AFE_ChInit(CH1,
                AFE_CH_SWTRG_CCM_PGAOFF_CONFIG(DEC_OSR2048),
                0,
                PRI_LVL1,
                (AFE_CH_CALLBACK)metering_func_afe_ch1_callback);
    AFE_ChInit(CH2,
                AFE_CH_SWTRG_CCM_PGAOFF_CONFIG(DEC_OSR2048),
                0,
                PRI_LVL1,
                (AFE_CH_CALLBACK)metering_func_afe_ch2_callback);
    AFE_ChInit(CH3,
                AFE_CH_SWTRG_CCM_PGAOFF_CONFIG(DEC_OSR2048),
          0,
                PRI_LVL1,
                (AFE_CH_CALLBACK)metering_func_afe_ch3_callback);
    /* PLL = 32768 Hz * 375 = 12288000 */
    AFE_Init(AFE_MODULE_RJFORMAT_CONFIG(AFE_PLL_CLK, AFE_DIV4, (12.288e6)));
```

When AFE is configured as continus mode, samples per second can be calculated by:

Samples_Per_Sec = PLL_CLK / AFE_DIV / AFE_OSR

In the example, the samples per second is 12288000 / 4 / 2048 = 1500.

ADC initialization for voltage sampling:

```
        /* Do ADC calibration will have some benefits */
         ADC_ExecCalib(ADC_MODULE_DIV4_16B_HWTRG_IREF_CONFIG, &adc_calib);
        ADC_SaveCalib(&adc_calib);
         ADC_Init(ADC_MODULE_DIV4_16B_HWTRG_IREF_CONFIG,
         HWAVG_OFF,
         ADC_CH_SE_POLL_CONFIG(ADC_SE1),
         ADC_CH_DISABLE_CONFIG,
         ADC_CH_DISABLE_CONFIG,
         ADC_CH_DISABLE_CONFIG,
         PRI_LVL1,
         (ADC_CALLBACK)NULL);
```

The ADC channels are used in order to have easier converter synchronization. All channels can be started at a precisely defined time. The KM3x devices solve this problem by using the XBAR periphery. The XBAR is an internal connection matrix on the periphery. Internal signals such as conversion complete from the SD converter can be used for starting the SAR conversion. The complete signal sampling process is based on the combination of three or four SD and one SAR with an input multiplexer that is fully supported by the device hardware. Only the conversion result must be read by the MCU core or by the DMA. XBAR initialization:

```
XBAR_Path(XBAR_AFE0COC, XBAR_ADCTRGCHA);
```

3.3 Sampling procedure

1. **Sampling functions**

   For multi-channel sampling, callback functions are assigned to each AFE channels. In AFE initilization, we can see functions of name metering_func_afe_chx_callback are assigned as callbacks.

```
/* Measurements callback @ 3000 Hz, duration us, load  % */
static void metering_func_afe_ch0_callback(AFE_CH_CALLBACK_TYPE type, int32 result)
{
    if (type == COC_CALLBACK)
    {
        i24_samplePh1 = AFE_ChRead(CH0);
    }
}

/* measurements callback @ 3000 Hz, duration us, load  % */
static void metering_func_afe_ch1_callback (AFE_CH_CALLBACK_TYPE type, int32_t result)
{
    if (type == COC_CALLBACK)
    {
        i24_samplePh2 = AFE_ChRead(CH1);
    }
}

/* measurements callback @ 3000 Hz, duration us, load  %                      */
static void metering_func_afe_ch2_callback(AFE_CH_CALLBACK_TYPE type, int32_t result)
{
    if (type == COC_CALLBACK)
    {
        i24_samplePh3 = AFE_ChRead(CH2);
    }
}

/* measurements callback @ 3000 Hz, duration us, load  %                      */
static void metering_func_afe_ch3_callback(AFE_CH_CALLBACK_TYPE type, int32_t result)
{
    int16_t temp16;

    if (type == COC_CALLBACK)
    {
        temp16 = ADC_Read(CHA) - 0x8000;
        u24_samplePh1 = (frac32)temp16 << 8;
        i24_samplePh4 = AFE_ChRead(CH3);

        if (CALI_FLAG_INIT == metering_cfg_data.calib_flag)
        {
     calib_data_update_offsets((tMETERING_CONFIG_FLASH_DATA *)&metering_cfg_data,
u24_samplePh1, i24_samplePh1, i24_samplePh2, i24_samplePh3, i24_samplePh4);
        }
        SWISR_HandlerCall(0);
    }
}
```

For each channels, the currents are sampled and we will get voltage value in last channel's IRQ handler and calculate the offset for calibration.

2. **Input data to metering library and calculate**

   After current and voltage samples are sampled, we will input the data to library.

   Function metering_func_auxcalc_swisr0_callback will input the data to metering library and do calculation. Note all channels' data are needed and user need to select appropriate metering API functions according to number of meter channels.

3. **Get metering result**

   Then SWISR_HandlerFreqCall(1, 1500, 4) is called to get the metering result.

   In the template, we'll get 1500 samples per second and so this function call will let us get metering result 4 times per second.

   Function metering_func_read_result_swisr1_callback will get metering result from metering library. Note all channels' data are needed and user need to select appropriate metering API functions according to number of meter channels.

3.4 Calibration

Calibration procedure need to applied to each channels.

The power meter is calibrated with the help of test equipment. The calibration task runs whenever a non-calibrated power meter is connected to the mains voltage. The running calibration task measures the phase voltage and phase current signals generated by the test equipment. It scans for a 220 V phase voltage and 5.0 A phase current waveforms with a 45 degree phase shift. The calibration task terminates by storing calibration gains, offsets and phase shift into the flash memory and by resetting the microcontroller device.

The calibration code are in calib_data.c and calib_data.h.

Note all channels need to be calibrated and all the calibration parameter need to be stored in flash for use.

3.5 Reference

1. Filter-Based Algorithm for Metering Applications (document AN4265)
2. KM34Z256 Bare-metal Software Drivers, (document KMSWDRVAPIRM_SW)

4 Revision history

Table 1. Revision history

Rev.No.	Date	Substantial Changes
0	04/2017	Initial release

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support