BMS_UI_Measure.c - BMS_UI_Measure.c

```c
/*********************************************************************
 *
 *                  BMS UI Measure File
 *
 *********************************************************************
 * FileName:        BMS_UI_Measure.c
 * Processor:       PIC18F25K80
 * Compiler:        Microchip C18 v3.41
 * Company:         KIT - CN - IPE
 *
 * Author           Date         Comment
 *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 * Reiling V.       03.08.2012   Release
 *********************************************************************/

/*********************************************************************
 *
 *                          Include Files
 *
 ********************************************************************/
#include "BMS_UI_Main.h"


/*+++++++ Measure BMS_UI +++++++++++++++++++++++++++++++++++++++++++*/
uint16_t Measure( void )
{
   uint16_t i = 0;
   uint32_t VoltageOverS = 0;
   int32_t  CurrentOverS = 0;
   uint16_t LEMref       = 0;
   uint32_t LEMrefSUM    = 0;


   for(i=0;i<OverSample;i++)
   {
     VoltageOverS += ADCread( HV );
     CurrentOverS += ADCread( HI );
   }

   VoltageOverS = VoltageOverS >> FirstShift;
   VoltageOverS = VoltageOverS *  UMulpli;
   gVoltage     = VoltageOverS >> ULastShift;

   for(i=0; i<32; i++)
   {
     ADCON0bits.GO=1;                // starts conversion
     while(ADCON0bits.DONE==1u);     // wait, it's converting
     LEMref = ADRESH & 0xF;          // read high nibble
     LEMref <<= 8;                   // shift high nibble into right Position
     LEMref += ADRESL;               // add low Byte
     LEMrefSUM += LEMref;
   }

   CurrentOverS = CurrentOverS >> FirstShift;
   CurrentOverS = CurrentOverS *  IMulpli;
   CurrentOverS = CurrentOverS >> ILastShift;
```

```c
/*********************************************************************
 *
 *                  BMS UI Measure File
 *
 *********************************************************************
 * FileName:        BMS_UI_Measure.c
 * Processor:       PIC18F25K80
 * Compiler:        Microchip C18 v3.41
 * Company:         KIT - CN - IPE
 *
 * Author           Date         Comment
 *~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 * Reiling V.       03.08.2012   Release
 *********************************************************************/

/*********************************************************************
 *
 *                          Include Files
 *
 ********************************************************************/
#include "BMS_UI_Main.h"


/*+++++++ Measure BMS_UI +++++++++++++++++++++++++++++++++++++++++++*/
uint16_t Measure( void )
{
   uint16_t i = 0;
   uint32_t VoltageOverS = 0;
   int32_t  CurrentOverS = 0;

   for(i=0;i<OverSample;i++)
   {
     VoltageOverS += ADCread( HV );
     CurrentOverS += ADCread( HI );
   }

   VoltageOverS = VoltageOverS >> FirstShift;
   VoltageOverS = VoltageOverS *  UMulpli;
   gVoltage     = VoltageOverS >> ULastShift;
   //gVoltage+=32;

   CurrentOverS = CurrentOverS >> FirstShift;
   CurrentOverS = CurrentOverS *  IMulpli;
   CurrentOverS = CurrentOverS >> ILastShift;
```

```
   CurrentOverS = CurrentOverS -  LEMrefSUM - gOffset;
   gCurrent     = CurrentOverS;


   return 0;
}
```

```
   gCurrent     = CurrentOverS -  IOffset;
   //gCurrent+=16;


   return 0;
}



/*+++++++ CAL_LEM ++++++++++++++++++++++++++++++++++++++++++++++++++*/
uint16_t CAL_LEM( void )
{
  uint8_t  i=0;
  uint16_t j=0;
  uint16_t LEM_Status = LEM_NOK;
  uint16_t DAC_VAR = DAC_MID_RANGE;
  int16_t  LEM_ADC = 0;
  int32_t  CurrentOS = 0;


  for(i=0; i<16; i++)                  // 16 Try to cal LEM Offset
  {
    DAC_VAR -= LEM_ADC;                // compute new DAC
    DACwrite(DAC_VAR);                 // Set DAC to Mid Range = LEM Zero
    Delay10KTCYx( 128 );               // wait 75ms for DAC

    CurrentOS = 0;                     // Sigma OS reset
    for(j=0; j<1024; j++)              // 1024x OS
    {
      CurrentOS += ADCread( HI );      // measure of Current_ADC
    }
    LEM_ADC = (CurrentOS >> 15)-1247;  // /1024 & reduce to 11Bit & mid 40000

    if(!LEM_ADC)                       // if Offset = 0
    {
      LEM_Status = LEM_OK;             // show good
      break;                           // then break Try
    }
  }
  return LEM_Status;
}
```