```c
#include "defs.h"
#include "adc.h"
#include "unicon.h"


ADC_IntRegs_TypeDef ADC_InternalRegisters;


static uint16_t    ADC_StartConversion(uint32_t channel, uint32_t resolution);

/*  */
void ADC_Init(void){

    ADC_InternalRegisters.VRefInt.Resolution = LL_ADC_RESOLUTION_12B;
    ADC_InternalRegisters.McuTemp.Resolution = LL_ADC_RESOLUTION_12B;
    ADC_InternalRegisters.VBat.Resolution = LL_ADC_RESOLUTION_10B;

    LL_ADC_StartCalibration(ADC1);
    while (LL_ADC_IsCalibrationOnGoing(ADC1));

    LL_ADC_ClearFlag_ADRDY(ADC1);

    ADC_Read_VREFINT();
    ADC_Read_MCUTEMP();
    ADC_Read_VBAT();
}


/*  */
static uint16_t ADC_StartConversion(uint32_t channel, uint32_t resolution) {

    uint16_t result;

    ADC1->CHSELR = channel;
    LL_ADC_SetResolution(ADC1, resolution);

    do {
        LL_ADC_Enable(ADC1);
    } while( !LL_ADC_IsActiveFlag_ADRDY(ADC1) );

    LL_ADC_REG_StartConversion(ADC1);
    while( !LL_ADC_IsActiveFlag_EOC(ADC1) );

    switch(resolution) {
    case LL_ADC_RESOLUTION_6B:
        result = LL_ADC_REG_ReadConversionData6(ADC1);
        break;
    case LL_ADC_RESOLUTION_8B:
        result = LL_ADC_REG_ReadConversionData8(ADC1);
        break;
    case LL_ADC_RESOLUTION_10B:
        result = LL_ADC_REG_ReadConversionData10(ADC1);
        break;
    case LL_ADC_RESOLUTION_12B:
        result = LL_ADC_REG_ReadConversionData12(ADC1);
        break;
    }

    do {
        LL_ADC_Disable(ADC1);
    } while( LL_ADC_IsEnabled(ADC1) );

    //Delay_ms(1);

    return result;
}
```

```
 67
 68
 69  /* Skaitom VREF reiksme */
 70  void ADC_Read_VREFINT(void){
 71
 72      LL_ADC_SetCommonPathInternalCh( __LL_ADC_COMMON_INSTANCE(ADC1),
LL_ADC_PATH_INTERNAL_VREFINT );
 73
 74      ADC_InternalRegisters.VRefInt.AdcVal = ADC_StartConversion(
LL_ADC_CHANNEL_VREFINT, ADC_InternalRegisters.VRefInt.Resolution);
 75      ADC_InternalRegisters.VRefInt.ConvertedValue = __LL_ADC_CALC_VREFANALOG_VOLTAGE(
ADC_InternalRegisters.VRefInt.AdcVal, LL_ADC_RESOLUTION_12B );
 76  }
 77
 78
 79  /* Skaitom MCUTEMP reiksme */
 80  void ADC_Read_MCUTEMP(void){
 81
 82      LL_ADC_SetCommonPathInternalCh( __LL_ADC_COMMON_INSTANCE(ADC1),
LL_ADC_PATH_INTERNAL_TEMPSENSOR );
 83
 84      ADC_InternalRegisters.McuTemp.AdcVal = ADC_StartConversion(
LL_ADC_CHANNEL_TEMPSENSOR, ADC_InternalRegisters.McuTemp.Resolution);
 85      ADC_InternalRegisters.McuTemp.ConvertedValue = __LL_ADC_CALC_TEMPERATURE(
ADC_InternalRegisters.VRefInt.ConvertedValue, ADC_InternalRegisters.McuTemp.AdcVal,
LL_ADC_RESOLUTION_12B );
 86  }
 87
 88
 89  /*  */
 90  void ADC_Read_VBAT(void){
 91
 92      LL_ADC_SetCommonPathInternalCh( __LL_ADC_COMMON_INSTANCE(ADC1),
LL_ADC_PATH_INTERNAL_VBAT );
 93
 94      ADC_InternalRegisters.VBat.AdcVal = ADC_StartConversion(LL_ADC_CHANNEL_VBAT,
ADC_InternalRegisters.VBat.Resolution);
 95      ADC_InternalRegisters.VBat.ConvertedValue = __LL_ADC_CALC_DATA_TO_VOLTAGE(
ADC_InternalRegisters.VRefInt.ConvertedValue, ADC_InternalRegisters.VBat.AdcVal,
LL_ADC_RESOLUTION_10B ) * 2;
 96  }
 97
 98
 99  /*  */
100  uint16_t ADC_ReadAnalog(uint32_t channel){
101
102      LL_ADC_SetCommonPathInternalCh( __LL_ADC_COMMON_INSTANCE(ADC1),
LL_ADC_PATH_INTERNAL_NONE );
103
104      //return ADC_StartConversion( __LL_ADC_DECIMAL_NB_TO_CHANNEL(channel),
LL_ADC_RESOLUTION_10B );
105      return ADC_StartConversion( channel, LL_ADC_RESOLUTION_10B );
106  }
107
108
109  /*  */
110  uint16_t ADC_ConvertTo_mVolts(uint16_t adcval, uint32_t resolution){
111      return __LL_ADC_CALC_DATA_TO_VOLTAGE(ADC_InternalRegisters.VRefInt.
ConvertedValue, adcval, resolution );
112  }
```