```
/*******************************************************************************
* File Name: ADC_DelSig_1_INT.c
* Version 2.30
*
* Description:
*  This file contains the code that operates during the ADC_DelSig interrupt
*  service routine.
*
* Note:
*
********************************************************************************
* Copyright 2008-2012, Cypress Semiconductor Corporation.  All rights reserved.
* You may use this file only in accordance with the license, terms, conditions,
* disclaimers, and limitations in the end user license agreement accompanying
* the software package with which this file was provided.
*******************************************************************************/

#include "ADC_DelSig_1.h"


/*******************************************************************************
* Custom Declarations and Variables
* - add user inlcude files, prototypes and variables between the following
*   #START and #END tags
*******************************************************************************/
/* `#START ADC_SYS_VAR`  */
#include <device.h>
    float V_SUM = 0;
    float V_ADC = 0;
    float V_AVG = 0;
    //Antal samples der skal midles:
    int n = 2;
    int i;
/* `#END`  */


/*******************************************************************************
* Function Name: ADC_DelSig_1_ISR1
********************************************************************************
*
* Summary:
*  Handle Interrupt Service Routine.
*
* Parameters:
*  void
*
* Return:
*  void
*
* Reentrant:
*  No
*
*******************************************************************************/
CY_ISR( ADC_DelSig_1_ISR1)
{
    /*******************************************************************************
    *  Custom Code
```

```
    *  - add user ISR code between the following #START and #END tags
    ******************************************************************************/
    /* `#START MAIN_ADC_ISR1`  */

    //Delta sigma'en startes og der køres n målinger der midles over.
    for(i = 0 ; i < n ; i++){
        V_ADC=ADC_DelSig_1_CountsTo_Volts(ADC_DelSig_1_GetResult32());
        V_SUM = V_SUM+V_ADC;
    }
    V_AVG = V_SUM/2 - 2.5;
    if((V_AVG > 0.3) & BurstFlag){
        DistanceTimerVal = Timer_1_ReadCounter();
        CalcDistFlag = 1;
        BurstFlag = 0;
    }
    V_SUM = 0;
    /* `#END`  */

    /* Stop the conversion if conversion mode is single sample and resolution
       is above 16 bits.
    */
    #if(ADC_DelSig_1_CFG1_RESOLUTION > 16 && \
        ADC_DelSig_1_CFG1_CONV_MODE == ADC_DelSig_1_MODE_SINGLE_SAMPLE)
        ADC_DelSig_1_StopConvert();
    #endif /* Single sample conversion mode with resolution above 16 bits */

}


/*******************************************************************************
* Function Name: ADC_DelSig_1_ISR2
********************************************************************************
*
* Summary:
*  Handle Interrupt Service Routine.
*
* Parameters:
*  void
*
* Return:
*  void
*
* Reentrant:
*  No
*
*******************************************************************************/
CY_ISR( ADC_DelSig_1_ISR2)
{
    /******************************************************************************
    *  Custom Code
    *  - add user ISR code between the following #START and #END tags
    ******************************************************************************/
    /* `#START MAIN_ADC_ISR2`  */

    /* `#END`  */

    /* Stop the conversion conversion mode is single sample and resolution
```

```
       is above 16 bits.
    */
    #if(ADC_DelSig_1_CFG2_RESOLUTION > 16 && \
        ADC_DelSig_1_CFG2_CONVMODE == ADC_DelSig_1_MODE_SINGLE_SAMPLE)
        ADC_DelSig_1_StopConvert();
    #endif /* Single sample conversion mode with resolution above 16 bits */

}


/*******************************************************************************
* Function Name: ADC_DelSig_1_ISR3
********************************************************************************
*
* Summary:
*  Handle Interrupt Service Routine.
*
* Parameters:
*  void
*
* Return:
*  void
*
* Reentrant:
*  No
*
*******************************************************************************/
CY_ISR( ADC_DelSig_1_ISR3)
{
    /************************************************************************
    *  Custom Code
    *  - add user ISR code between the following #START and #END tags
    *************************************************************************/
    /* `#START MAIN_ADC_ISR3`  */

    /* `#END`  */

    /* Stop the conversion if conversion mode is set to single sample and
       resolution is above 16 bits.
    */
    #if(ADC_DelSig_1_CFG3_RESOLUTION > 16 && \
        ADC_DelSig_1_CFG3_CONVMODE == ADC_DelSig_1_MODE_SINGLE_SAMPLE)
        ADC_DelSig_1_StopConvert();
    #endif /* Single sample conversion mode with resolution above 16 bits */
}


/*******************************************************************************
* Function Name: ADC_DelSig_1_ISR4
********************************************************************************
*
* Summary:
*  Handle Interrupt Service Routine.
*
* Parameters:
*  void
*
```

```
* Return:
*  void
*
* Reentrant:
*  No
*
*******************************************************************************/
CY_ISR( ADC_DelSig_1_ISR4)
{
    /*******************************************************************************
    *  Custom Code
    *  - add user ISR code between the following #START and #END tags
    *******************************************************************************/
    /* `#START MAIN_ADC_ISR4`  */

    /* `#END`  */

    /* Stop the conversion if conversion mode is set to single sample and
       resolution is above 16 bits.
    */
    #if(ADC_DelSig_1_CFG4_RESOLUTION > 16 && \
        ADC_DelSig_1_CFG4_CONVMODE == ADC_DelSig_1_MODE_SINGLE_SAMPLE)
        ADC_DelSig_1_StopConvert();
    #endif /* Single sample conversion mode with resolution above 16 bits */
}


/* [] END OF FILE */
```