```
/* ========================================
 *
 * Copyright YOUR COMPANY, THE YEAR
 * All Rights Reserved
 * UNPUBLISHED, LICENSED SOFTWARE.
 *
 * CONFIDENTIAL AND PROPRIETARY INFORMATION
 * WHICH IS THE PROPERTY OF your company.
 *
 * ========================================
*/
#include <device.h>
#include "Kommunikation.h"
#include "Haeldningsregulering.h"
#include "Init.h"
#include "Haeldningssensorblok.h"
#include "VBTE-Driver.h"



void main()
{
    init();
    CyDelay(1000);
    int retval = 0;
    struct smflags sm;
    sm.autoflag = 0;
    sm.VBTE1Niveau = 101; // Niveau > 100 = ikke noget niveau modtaget
    sm.VBTE2Niveau = 101; // Niveau > 100 = ikke noget niveau modtaget
    sm.VBTE1Status = 4; // Ingen kontakt til VBTE1
    sm.VBTE2Status = 4; // Ingen kontakt til VBTE2
    sm.levelVal = STLEVEL;
    sm.vinkelVal = STLEVEL;
    CyGlobalIntEnable; /* Uncomment this line to enable global interrupts. */

    for(;;)
    {
        getFromKI(&sm);
        LED_Control_Reg_Write(7);
        autoReg(&sm);


        /* Integrationstest stub
        sm.levelVal = getLevel();
        if(sm.autoflag == 1){
        sm.VBTE2Niveau = writeToVbte(VBTE2Addr, VBTENIVEAU);
        if(sm.levelVal == sm.vinkelVal);

        if(sm.levelVal > sm.vinkelVal ){
                    writeToVbte(VBTE2Addr, TOPVENTIL);
                    sm.VBTE2Status = 1;
            }
        if(sm.levelVal < sm.vinkelVal ){
                    writeToVbte(VBTE2Addr, BUNDVENTIL);
                    sm.VBTE2Status = 1;
            }
        }*/
```

```
        LED_Control_Reg_Write(sm.VBTE2Niveau);

        /* STUB TESTS */
        /* writeToVbte test

        retval = writeToVbte(VBTE1Addr, 3);
        CyDelay(500);
        LED_Control_Reg_Write(retval);
        retval = writeToVbte(VBTE1Addr, 4);
        CyDelay(500);
        LED_Control_Reg_Write(retval);
        retval = writeToVbte(VBTE1Addr, VBTENIVEAU);
        CyDelay(500);
        LED_Control_Reg_Write(retval);
        retval = writeToVbte(VBTE1Addr, TOPVENTIL);
        CyDelay(500);
        LED_Control_Reg_Write(retval);
        retval = writeToVbte(VBTE1Addr, BUNDVENTIL);
        CyDelay(500);
        LED_Control_Reg_Write(retval);
        retval = writeToVbte(VBTE1Addr, LUKVENTIL);
        CyDelay(500);
        LED_Control_Reg_Write(retval);
        retval = writeToVbte(VBTE1Addr, 9);
        CyDelay(500);
        LED_Control_Reg_Write(retval); */

        /* VBTE i2c test
        writeToVbte(VBTE2Addr, VBTENIVEAU);*/

        /* convertToEnum test
        int val = 2000;
        int i = 0;
        int res[200];
        for(i = 1; i < 201; i++){
            retval = convertToEnum(val+(17*i));
            res[i] = retval;
            }*/

        /* convertToValue
        int res[30];
        int i = 0;
        for(i = 0; i < 30; i++){
            convertToValue(i, &sm);
            res[i] = sm.vinkelVal;
        }*/
        /* Get Level test */
        /*sm.levelVal = getLevel();
        CyDelay(1000);*/
    }
}

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