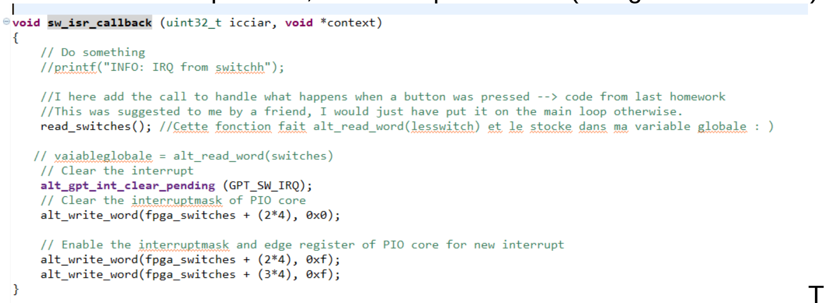
LINGI 2315 - Evaluation 1 - Bare-metal Project

P1

Name	Delcoigne Ben	Noma	38771700

Description of the software that implements the first question

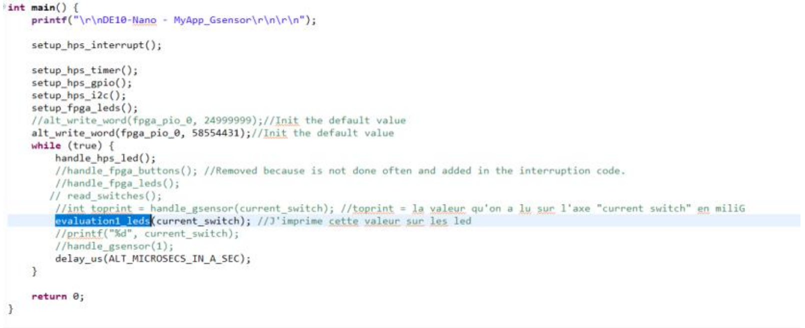
I set a global variable “current_switch” which knows what state the switches are in at all times. It defaults to 0.
When a switch is pressed, an interrupt is raised (using code from hw)



```
void sw_isr_callback (uint32_t icciar, void *context)
{
    // Do something
    //printf("INFO: IRQ from switchh");

    //I here add the call to handle what happens when a button was pressed --> code from last homework
    //This was suggested to me by a friend, I would just have put it on the main loop otherwise.
    read_switches(); //Cette fonction fait alt_read_word(lesswitch) et le stocke dans ma variable globale : )

   // vaiableglobale = alt_read_word(switches)
    // Clear the interrupt
    alt_gpt_int_clear_pending (GPT_SW_IRQ);
    // Clear the interruptmask of PIO core
    alt_write_word(fpga_switches + (2*4), 0x0);

    // Enable the interruptmask and edge register of PIO core for new interrupt
    alt_write_word(fpga_switches + (2*4), 0xf);
    alt_write_word(fpga_switches + (3*4), 0xf);
}
```

The function read_switches simply reads the value of the switches and stores it into the global variable **current_switch.**
This closes the part about updating the active switch correctly.

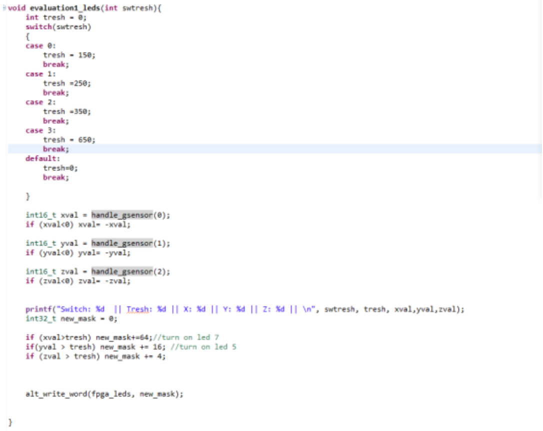
Now lets have a look at the display on the LED’s:



```
int main() {
    printf("\r\nDE10-Nano - MyApp_Gsensor\r\n\r\n");

    setup_hps_interrupt();

    setup_hps_timer();
    setup_hps_gpio();
    setup_hps_i2c();
    setup_fpga_leds();
    //alt_write_word(fpga_pio_0, 24999999);//Init the default value
    alt_write_word(fpga_pio_0, 58554431);//Init the default value
    while (true) {
        handle_hps_led();
        //handle_fpga_buttons(); //Removed because is not done often and added in the interruption code.
        //handle_fpga_leds();
       // read_switches();
        //int toprint = handle_gsensor(current_switch); //toprint = la valeur qu'on a lu sur l'axe "current switch" en miliG
        evaluation1_leds(current_switch); //J'imprime cette valeur sur les led
        //printf("%d", current_switch);
        //handle_gsensor(1);
        delay_us(ALT_MICROSECS_IN_A_SEC);
    }

    return 0;
}
```

My main function calls the function evaluation1_leds with the current switch value as a parameter.

ere is the code of that function:



```
void evaluation1_leds(int swtresh){
    int tresh = 0;
    switch(swtresh)
    {
    case 0:
        tresh = 150;
        break;
    case 1:
        tresh =250;
        break;
    case 2:
        tresh =350;
        break;
    case 3:
        tresh = 650;
        break;
    default:
        tresh=0;
        break;

    }

    int16_t xval = handle_gsensor(0);
    if (xval<0) xval= -xval;

    int16_t yval = handle_gsensor(1);
    if (yval<0) yval= -yval;

    int16_t zval = handle_gsensor(2);
    if (zval<0) zval= -zval;


    printf("Switch: %d  || Tresh: %d || X: %d || Y: %d || Z: %d || \n", swtresh, tresh, xval,yval,zval);
    int32_t new_mask = 0;

    if (xval>tresh) new_mask+=64;//turn on led 7
    if(yval > tresh) new_mask += 16; //turn on led 5
    if (zval > tresh) new_mask += 4;



    alt_write_word(fpga_leds, new_mask);


}
```

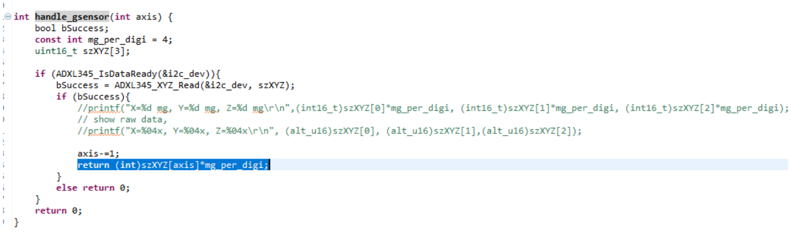
As you see, I set the treshold according to the switch value. I then read x, y and z from the accelerometer using handle_gsensor. This function was created by the assistants for the homework, I just extract the x, y and z axis. (code below).
Once done, I set the LED mask using binary values associated to the mask.

NOTE: Nicolas Chavaux evaluated me and as I showed him, for an unknown reason, the X axis seems to have an unconsistent reading. The code does the same for the three axis so there is no software reason it wouldn’t work. I flashed the fpga several times but the error persists.

LINGI 2315 - Evaluation 1 - Bare-metal Project **P2**

Description of the hardware & software that implement the second question (C, Verilog, Qsys)

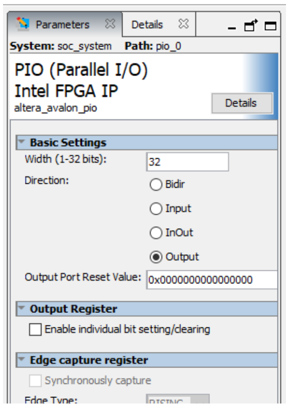
APPENDIX FROM PART 1:
Code of handle_gsensor:



```
int handle_gsensor(int axis) {
    bool bSuccess;
    const int mg_per_digi = 4;
    uint16_t szXYZ[3];

    if (ADXL345_IsDataReady(&i2c_dev)){
        bSuccess = ADXL345_XYZ_Read(&i2c_dev, szXYZ);
        if (bSuccess){
            //printf("X=%d mg, Y=%d mg, Z=%d mg\r\n",(int16_t)szXYZ[0]*mg_per_digi, (int16_t)szXYZ[1]*mg_per_digi, (int16_t)szXYZ[2]*mg_per_digi);
            // show raw data,
            //printf("X=%04x, Y=%04x, Z=%04x\r\n", (alt_u16)szXYZ[0], (alt_u16)szXYZ[1],(alt_u16)szXYZ[2]);

            axis-=1;
            return (int)szXYZ[axis]*mg_per_digi;
        }
        else return 0;
    }
    return 0;
}
```

--------------------------------------PART 2 ------------------

1/ Set the PIO to 32 bits in the platform designer



System: soc_system Path: pio_0

PIO (Parallel I/O)
Intel FPGA IP
altera_avalon_pio

Basic Settings

Width (1-32 bits): 32

Direction: ○ Bidir ○ Input ○ InOut ● Output

Output Port Reset Value: 0x0000000000000000

Output Register

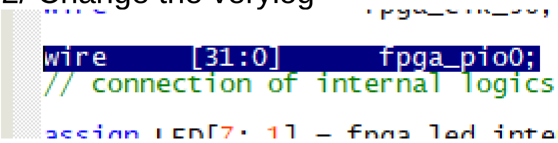
□ Enable individual bit setting/clearing

Edge capture register

□ Synchronously capture

Edge Type: RISING

2/ Change the verylog

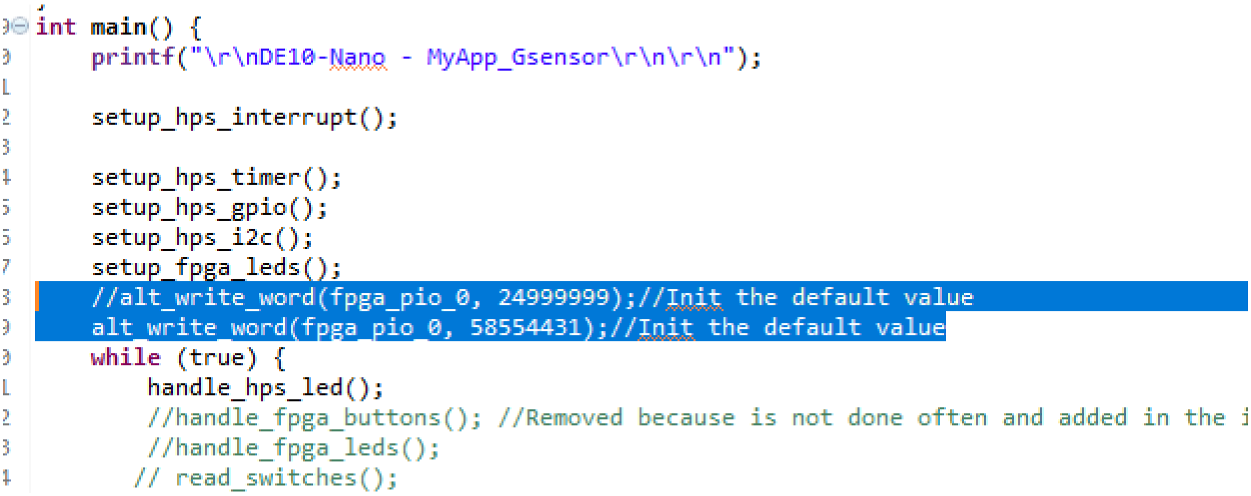


```
wire            fpga_clk_50;

wire        [31:0]      fpga_pio0;
// connection of internal logics

assign LED[7: 1] = fpga_led_internal;
```

Only the number of bits was changed in the wire.

3/ Compile the design. Once done, generate the .h file and transfer it into the gsensor Eclipse project

In the C code, I set the default value to 249999 before, now, to take advantage of the 32 bits, I set it to 58554431



```
int main() {
    printf("\r\nDE10-Nano - MyApp_Gsensor\r\n\r\n");

    setup_hps_interrupt();

    setup_hps_timer();
    setup_hps_gpio();
    setup_hps_i2c();
    setup_fpga_leds();
    //alt_write_word(fpga_pio_0, 24999999);//Init the default value
    alt_write_word(fpga_pio_0, 58554431);//Init the default value
    while (true) {
        handle_hps_led();
        //handle_fpga_buttons(); //Removed because is not done often and added in the i
        //handle_fpga_leds();
       // read_switches();
```