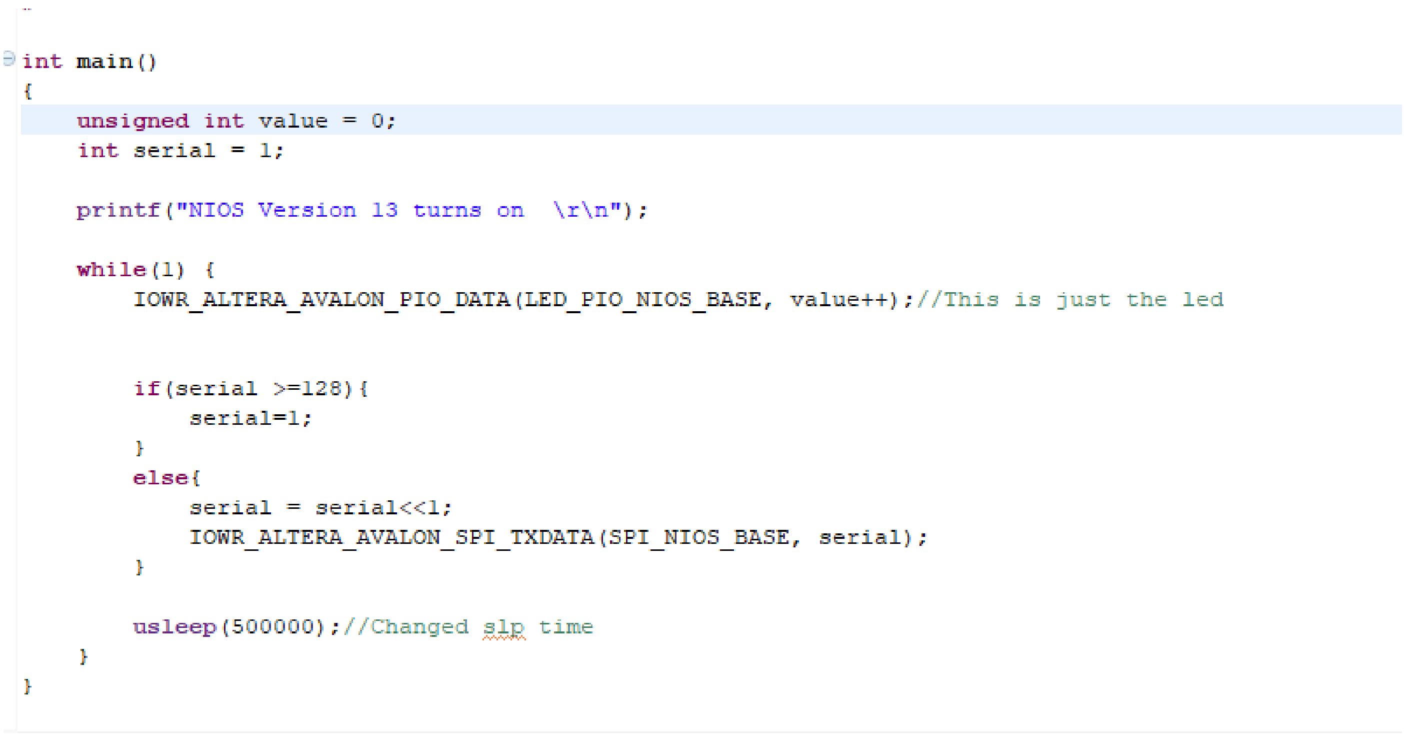
LINGI 2315 - Homework 5 - My Nios App

P1

Name	Delcoigne Ben	Noma	38771700

Description of the software on Nios with relevant screenshots of your code



```
int main()
{
    unsigned int value = 0;
    int serial = 1;

    printf("NIOS Version 13 turns on  \r\n");

    while(1) {
        IOWR_ALTERA_AVALON_PIO_DATA(LED_PIO_NIOS_BASE, value++);//This is just the led


        if(serial >=128){
            serial=1;
        }
        else{
            serial = serial<<1;
            IOWR_ALTERA_AVALON_SPI_TXDATA(SPI_NIOS_BASE, serial);
        }

        usleep(500000);//Changed slp time
    }
}
```

For the NIOS part I just created a LED mask (called serial) that I move to the right by one every time. Meanwhile the LED pio just increments itself each time.
The serial is then written in the SPI for the nios.

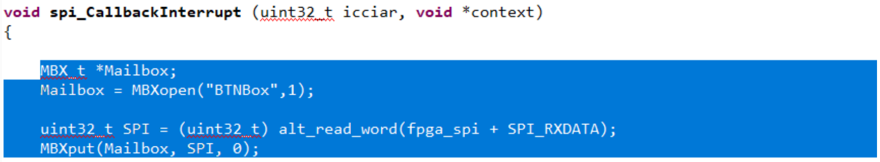
Do note that i had to `#include "altera_avalon_spi_regs.h"` in order for the code to work.

LINGI 2315 - Homework 5 - My Nios App

Name	Delcoigne Ben	38771700

Description of the software on HPS with relevant screenshots of your code

The HPS part was the main part of this homework, I first modified the spi interrupt task in order to retrieve the value that was sent to it by NIOS:

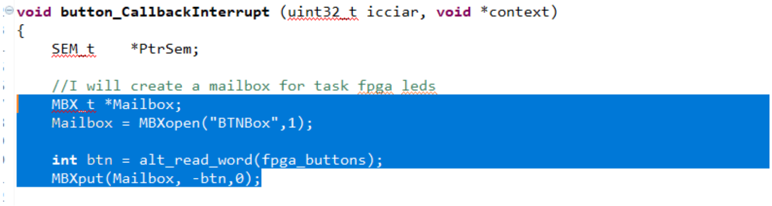


```
void spi_CallbackInterrupt (uint32_t icciar, void *context)
{

    MBX_t *Mailbox;
    Mailbox = MBXopen("BTNBox",1);

    uint32_t SPI = (uint32_t) alt_read_word(fpga_spi + SPI_RXDATA);
    MBXput(Mailbox, SPI, 0);
```

This is written in a mailbox (the value is just written into it)
Another thing I did was use the same mailbox to put the button values into:

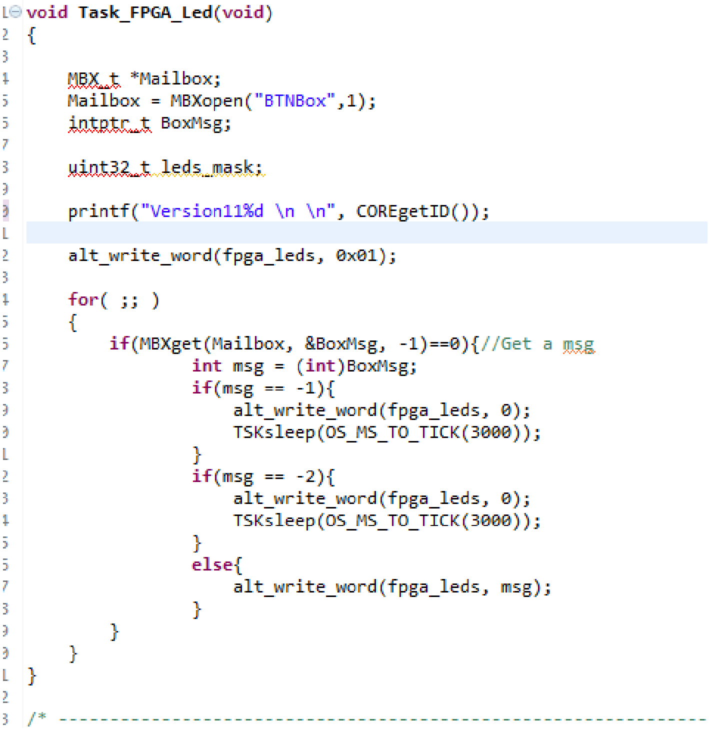


```
void button_CallbackInterrupt (uint32_t icciar, void *context)
{
    SEM_t     *PtrSem;

    //I will create a mailbox for task fpga leds
    MBX_t *Mailbox;
    Mailbox = MBXopen("BTNBox",1);

    int btn = alt_read_word(fpga_buttons);
    MBXput(Mailbox, -btn,0);
```

Note: I put the negative value of the button



```
void Task_FPGA_Led(void)
{

    MBX_t *Mailbox;
    Mailbox = MBXopen("BTNBox",1);
    intptr_t BoxMsg;

    uint32_t leds_mask;

    printf("Version11%d \n \n", COREgetID());

    alt_write_word(fpga_leds, 0x01);

    for( ;; )
    {
        if(MBXget(Mailbox, &BoxMsg, -1)==0){//Get a msg
                int msg = (int)BoxMsg;
                if(msg == -1){
                    alt_write_word(fpga_leds, 0);
                    TSKsleep(OS_MS_TO_TICK(3000));
                }
                if(msg == -2){
                    alt_write_word(fpga_leds, 0);
                    TSKsleep(OS_MS_TO_TICK(3000));
                }
                else{
                    alt_write_word(fpga_leds, msg);
                }
        }
    }
}

/* --------------------------------------------------------------
```

The last step was to modify the fgpa led task in order to read the mailbox. It either prints only zeros or ones depending on the button press (if it’s -1 or -2); otherwise it just writes the content of the mailbox.