```
/***********************************/
/**** PIN changed Sensing **********/
/***********************************/

/* onliy a specific Period and signalfwith < fck/2 */

/***** Pin Definitions *************/
/***** Depends on the AVR **********/

/***** Light Switches **************/

#define Switch1     PC0
#define Switch2     PC1
#define Switch3     PC2
#define Switch4     PC3

#define Switch_PORT     PORTC
#define Switch_DDR      DDRC
#define Switch_PIN      PINC

/***** Functions *******************/

void Switchsensing_Init(void);
void OC0A_Timer_Init(void);
void SearchforChanges(void);
```

```
/**********************************/
/**** PIN changed Sensing **********/
/**********************************/

#include <avr/io.h>
#include <stdint.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include "Pin_changed_Sensing_Dec.h"
#include "Converter.h"
#include "SwitchLights_Dec.h"
#include "NRF24L01_Dec.h"
#include "NRF_Send_Receive_dec.h"

#define Senscount 40

/*** for ATmega164P ***************/

volatile unsigned char newSwitch = 0x00;
volatile unsigned char oldSwitch = 0x00;
volatile unsigned char Switch_act = 0x00;

extern volatile unsigned char RelayState;
extern volatile uint8_t SensingTime;
extern unsigned char RelayPWM[4][1];
uint8_t lastSens = Senscount;

/*** Timer ************************/
void OC0A_Timer_Init(void)
{
    TCCR0A |= (1<<WGM01);
    TCCR0B |= (1<<CS02) | (1<<CS00);
    TIMSK0 |= (1<<OCIE0A);              // Interrupt
    OCR0A = SensingTime;
}

/*** Switchsensing *****************/

void Switchsensing_Init(void)
{
    Switch_DDR &= ~(1<<Switch1) | ~(1<<Switch2) | ~(1<<Switch3) | ~(1<<Switch4);
    Switch_PORT |= (1<<Switch1) | (1<<Switch2) | (1<<Switch3) | (1<<Switch4);
    newSwitch = 0x00;
    Switch_act = 0x00;
}

ISR(TIMER0_COMPA_vect)
{
    uint8_t tmp_sreg;

    TIFR0 &= ~(1<<OCF0A);       // Clears the Flag

    tmp_sreg = SREG;            // save Interruptstateregister

    cli();

    if(!(Switch_PIN & (1<<Switch1)))
    {
        Switch_act |= 0x01;
    }
    if(!(Switch_PIN & (1<<Switch2)))
    {
        Switch_act |= 0x02;
    }
```

```
    if(!(Switch_PIN & (1<<Switch3)))
    {
        Switch_act |= 0x04;
    }
    if(!(Switch_PIN & (1<<Switch4)))
    {
        Switch_act |= 0x08;
    }

    lastSens--;

    if(lastSens == 0)
    {
        newSwitch = Switch_act;
        Switch_act = 0x00;

        lastSens = Senscount;
    }

    SREG = tmp_sreg;
}

void SearchforChanges()
{
    if((oldSwitch & 0x01) != (newSwitch & 0x01))
    {
        SwitchInternalGroup(0x01);
    }
    if((oldSwitch & 0x02) != (newSwitch & 0x02))
    {
        SwitchInternalGroup(0x02);
    }
    if((oldSwitch & 0x04) != (newSwitch & 0x04))
    {
        SwitchInternalGroup(0x03);
    }
    if((oldSwitch & 0x08) != (newSwitch & 0x08))
    {
        SwitchInternalGroup(0x04);
    }

    oldSwitch = newSwitch;
}
```