Mikroprozessorsysteme Praktikum 2

Aufgabe 1

```
#include "../h/pmc.h"
#include "../h/pio.h"

int main(void)
{
  volatile StructPMC* pmcbase = PMC_BASE;       // Basisadresse des PMC
  volatile StructPIO* piobaseB = PIOB_BASE;     // Basisadresse PIOB
  piobaseB->PIO_PER = ALL_LEDS;
  piobaseB->PIO_OER = ALL_LEDS;
  pmcbase->PMC_PCER = 0x4000;
  volatile int i;
  volatile int j;
  //Aufgabe 1
 /* while(1)
  {
    piobaseB->PIO_CODR = ALL_LEDS;
    piobaseB->PIO_SODR = ALL_LEDS;
  }


  //Aufgabe 2

  while(1)
  {
        if(!(piobaseB->PIO_PDSR & KEY1))
        {
                piobaseB->PIO_CODR = ALL_LEDS;
        }
        if(!(piobaseB->PIO_PDSR & KEY2))
        {
                piobaseB->PIO_SODR = ALL_LEDS;
        }
  }*/

  //Aufgabe 3

  for(i = 0; i < 500000; i++)
  {
        piobaseB->PIO_CODR = ALL_LEDS;
        for(j = 0; j < 500000; j++)
        {
                if(j == 70000)
                {
                        piobaseB->PIO_SODR = ALL_LEDS;
                }
        }

  }
  return 0;
}
```

Mit welchen Anweisung werden die LED‘s ein- bzw. ausgeschaltet und warum?

piobaseB->PIO_CODR = ALL_LEDS -> anschalten
piobaseB->PIO_SODR = ALL_LEDS -> ausschalten

Aufgabe 2

In welchem Register muss welches Bit gesetzt sein/werden, damit der Zustand der Tasten über das Pin Data Status Register (PDSR) erfasst werden können?

Sobald man die Taste 1 drückt wird KEY1(1<<3) angeschaltet, dh. Es bekommt eine 1 an seinen Pin. Das Pin 12 im PDSR (Pin Data Status Register) muss deshalb eine 1 sein, da man man KEY1 drückt und das Pin Data Status Register den Wert des Pins an der Stelle von KEY1 zurückgibt. Angenommen das PDSR hat noch im Register weitere einser, werden diese einser mit dem UND-Operator verhindert bzw. Auf 0 gesetzt. Dh. Das nur KEY1 aktiviert bzw. benutzt werden kann. Da das Board Low-Activ ist, kann KEY1 nur benutzt werden, wenn der PIN an KEY1 auf 0 steht. Deswegen negiert man alles nochmal.

Aufgabe 3

Wie reagieren Ihre Tastendrücke an SW1 und SW2?

Die Tasten müssen gedrückt sein , wenn die if-Anweisung ausgewählt ist. Das Problem hierbei ist, das man durch einen kurzen Tastendruck die Leuchte anschaltet im Regelfall und nicht durch Gedrückthalten einer Taste bis die Leuchte angeschaltet wird.

Ab optimierungsstufe 1 funktioniert das Programm nicht mehr ordentlich.

OP = 1 ->Nachdem j = 499999, wird j auf 0 gesetzt und nicht mehr erhöht,
OP = 2 -> j wird nicht erhöht, Anweisung der IF wird ignoriert (springt automatisch in die IF)

Aufagbe 4

Wie reagieren nun ihre Tastendrücke an SW1 und SW2?

Sie reagieren sofort, bei Betätigung der Tasten. Davor musste man die Taste gedrückt halten bis im Code die IF-Anweisung aufgerufen wurde.

Hat das Bedienen der Tasten Einfluss auf die Blinkfrequenz von DS2?

Ja

```
#include "../h/pmc.h"
#include "../h/pio.h"
#include "../h/aic.h"

void taste_irq_handler (void) __attribute__ ((interrupt));


void taste_irq_handler (void)
{
        StructPIO* piobaseB = PIOB_BASE;
        StructAIC* aicbase = AIC_BASE;
        if(!(piobaseB->PIO_PDSR & KEY1))
        {
                piobaseB->PIO_CODR = ALL_LEDS;
        }
                if(!(piobaseB->PIO_PDSR & KEY2))
        {
                piobaseB->PIO_SODR = ALL_LEDS;
        }

        aicbase->AIC_EOICR = piobaseB->PIO_ISR; // InterruptStatusRegister lesen und in End of Interrupt Command Register schreiben

}

int main(void)
{
        while(1)
        {
        StructAIC* aicbase = AIC_BASE;
        StructPIO* piobaseB = PIOB_BASE;
        aicbase->AIC_IDCR = 1<<14; //Interrupt Disable Command Register(Interrupt PIOB ausschalten)
        aicbase->AIC_ICCR = 1<<14; // Interrupt Clear Command Register(Interrupt PIOB anschalten)
        aicbase->AIC_SVR[PIOB_ID] = (unsigned int) taste_irq_handler; //Adresse vom Handler in Source Vector Register in Vector speichern
        aicbase->AIC_SMR[PIOB_ID] = 0x7; //Source Mode Register(höchste Priorität)
        aicbase->AIC_IECR = 1<<14; //Interrupt PIOB einschalten
        piobaseB->PIO_IER = KEY1|KEY2; //Interrupt für KEY1 und KEY2 wird angeschaltet
        while(1);
        aicbase->AIC_IDCR = 1<<14;
        aicbase->AIC_ICCR = 1<<14;
        }
        return 0;
}
```