



MIKROPROZESSORPRAKTIKUM

WS2016

Termin 2

C-Programmierung für eingebettete Systeme Pointer, Peripherie, PIO,
Interrupt

Name, Vorname	Matrikelnummer	Anmerkungen
Datum	Raster (z.B. Mi3x)	Testat/Datum

Legende: V: Vorbereitung, D: Durchführung, P: Protokoll/Dokumentation, T: Testat

Lernziele:

Der Umgang mit digitalen Ein- und Ausgängen und Unterbrechungen (Interrupt).

Arbeitsverzeichnis:

Kopieren Sie sich aus dem Ordner /mnt/Originale das Verzeichnis mpsWS2016. Dort finden Sie zu jedem Termin vorgegebene Dateien.

Aufgabe 1:

Wir werden uns nach Termin 1 nun nochmals mit den Registern der PIO des hier eingesetzten Mikrocontroller (AT91M63200) beschäftigen. Durch die strukturierte Anordnung der vielen verschiedenen Register der Peripherie des Mikrocontroller ist es möglich durch angelegte Strukturen sich die Programmierung und den Umgang mit der Peripherie zu vereinfachen. Schauen Sie sich Termin2Aufgabe1.c an. Testen und beschreiben Sie die Funktion des Programms.

Aufgabe 2:

Gut, wir können nun die Leuchtdioden (DS1..DS8) kontrollieren. Ändern und erweitern Sie das Programm so, dass die LED DS1 durch drücken der Taste SW1 eingeschaltet und durch drücken der Taste SW2 ausgeschaltet wird. Sollten die Tasten nicht funktionieren, so überprüfen Sie ob im Power Management Controller (PMC) der Clock für PIOB eingeschaltet ist. In welchem Register muss welches Bit gesetzt sein?

Aufgabe 3:

Lassen Sie im nächsten Programm zusätzlich die LED DS2 mit ca. 0,5Hz (Zeitschleife programmieren) blinken. Wie reagieren Ihre Tastendrucke an SW1 und SW2?

Aufgabe 4:

Schreiben Sie für die Tasten SW1 und SW2 eine passende Interruptserviceroutine. Erklären Sie die Wechsel der ARM-Betriebsmodi. Wie reagieren nun Ihre Tastendrucke an SW1 und SW2? Welchen Einfluss hat das Bedienen der Tasten nun auf die Blinkfrequenz von DS2?

Aufgabe 5:

Erstellen Sie zu diesem Termin ein Protokoll mit den Lösungen zu den Aufgaben und Ihren Erkenntnissen. Das Protokoll sollen Sie zum nächsten Termin vorlegen können. Denken Sie daran, dass einige Programmteile nochmals benötigt werden.

Programmgerüstbeispiele:

```
// Loesung zu Termin 2
// Aufgabe 1
// Namen: _____; _____
// Matr.: _____; _____
// vom : _____
//

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

int main(void)
{
    StructPMC* pmcbase = PMC_BASE;          // Basisadresse des PMC
    StructPIO* piobaseB = PIOB_BASE; // Basisadresse PIO B
    piobaseB->PIO_PER = ALL_LEDS;
    piobaseB->PIO_OER = ALL_LEDS;

    while(1)
    {
        piobaseB->PIO_CODR = ALL_LEDS;
        piobaseB->PIO_SODR = ALL_LEDS;
    }

    return 0;
}
```

```
// Loesung zu Termin2
// Aufgabe 4
// Namen: _____; _____
// Matr.: _____; _____
// vom : _____
//

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

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

void taste_irq_handler (void)
{

}

int main(void)
{

    return 0;

}
```