



FACHHOCHSCHULE DARMSTADT

FACHBEREICH INFORMATIK

MIKROPROZESSORPRAKTIKUM

WS2003/04

Termin 2

C-Programmierung für eingebettete Systeme

Pointer, Peripherie, PIO, Interrupt

Lernziele:

Erster Umgang mit Peripherie in eingebetteten Systemen.

Arbeitsverzeichnis:

Sie arbeiten in dem Verzeichnis `/home/milabuser/mi2/Termin2`. Dort stehen die Dateien *Termin2AufgabeX.c* als Programmgerüstbeispiele und *makefile1* zur Verfügung.

Aufgabe 1:

In eingebetteten Systemen und für Kartentreiber muss man oft auf Register zugreifen, die auf festen Adressen liegen. Wir werden uns zuerst mit den Registern einer PIO beschäftigen.

Name	Adresse	Bedeutung
PIOB_PER	0xFFFF0000	PIOB Port Enable Register
PIOB_OER	0xFFFF0010	PIOB Output Enable Register
PIOB_SODR	0xFFFF0030	PIOB Set Output Data Register
PIOB_CODR	0xFFFF0034	PIOB Clear Output Data Register

Legen Sie zunächst Pointer Variablen (PIOB\_PER, PIOB\_OER, ..) an und initialisieren Sie diese im Code auf die Adresse des Registers. Danach können Sie über diese Pointer auf die Register zugreifen. Schreiben Sie zunächst den Wert 0x100 ins PIOB\_PER und dann ins PIOB\_OER. Danach schreiben Sie nacheinander den Wert 0x100 einige Male alternierend ins PIOB\_SODR und PIOB\_CODR. Dadurch sollte die LED DS1 auf dem Board AT91EB63 an und aus gehen. Testen Sie dieses mit Einzelschritten aus. Sollte die LED nicht blinken, so überprüfen Sie ob im Power Management Controller der Clock für PIOB eingeschaltet ist. In welchem Register muß welches Bit gesetzt sein?

Aufgabe 2:

Gut, wir können jetzt eine LED (DS1) kontrollieren. Schreiben Sie die benötigten Pointer durch entsprechende "define" in ein allgemein zugängliches Header File. Ihr Header File benötigen Sie auch in den nächsten Versuchen. Sichern Sie deshalb Ihr Header File in geeigneter Weise.

Erweitern Sie Ihr Programm so, dass die LED DS1 durch drücken der Taste SW1 eingeschaltet und durch drücken der Taste SW2 ausgeschaltet wird.

Aufgabe 3:

Lassen Sie im nächsten Programm zusätzlich die LED DS2 mit ca. 0,5Hz 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.