

Mikroprozessorpraktikum WS 2011/12
Aufgabenkomplex: 1

Teilnehmer:

Marco Träger, Matr. XXXXXXXXX

Alexander Steen, Matr. 4357549

Gruppe: XXX, Arbeitsplatz: XXX

A 1.1 Output

A 1.1.1

A 1.1.2

A 1.1.3

A 1.1.4

A 1.1.5

A 1.1.6

A 1.2 Input

A 1.2.1

P1DIR entscheidet, ob der jeweilige Pin als Eingang oder Ausgang fungiert, dabei beschreibt 0 einen Eingang, 1 einen Ausgang

P1IN besteht aus einem Byte, deren Bits den aktuellen Logikpegel an dem jeweiligen Pin des Ports 1 darstellen

P1OUT zeigt an dem jeweiligen Bit an, welcher Logikpegel an dem zugehörigen Port anliegen soll, falls P1DIR auf Ausgang und P1SEL auf I/O-Funktion geschaltet ist

P1SEL gibt an, ob die einzelnen Pins des Port 1 direkt als I/O benutzt werden oder für ein angeschlossenes Modul genutzt werden

P1IE de-/aktiviert die Interrupt-Flags (P1IFG) für die Pins des ersten Ports.

P1IES entscheidet, ob man Interrupt durch eine low-high-Flanke (0) oder eine high-low-Flanke (1) auf dem jeweiligen Pin ausgelöst werden soll.

P1IFG bezeichnet die Interrupt-Flags der Pins von Port 1. Ist ein Bit von P1IFG auf 1 gesetzt, so wurde von dem zugehörigen Pin ein Interrupt ausgelöst.

A 1.2.2

Der AND-Operator (&) führt bitweise die Verundung der Bits der Arguments aus. Geht man für das Codebeispiel

```
if (P1IN & Taster) {...}
```

davon aus, dass man Pin i ein Taster angeschlossen ist, so kann man durch Wahl der Variable **Taster** als Bitmaske, die nur an der Stelle i eine 1 enthält ($\text{Taster} = 2^i$), erreichen, dass die Abfrage genau dann erfolgreich ist, falls der Taster gedrückt wurde.

A 1.2.3

```
1 #define Taster_rechts (0x01)
2 #define Taster_links (0x02)
3 P1DIR = 0x00;
4 P4DIR = 0xFF;
5 P4OUT = 0;
6 a = 7;
7 P4OUT = a;
8 P1OUT = a;
9 a = P1IN & 0x30; //beide Tasten
10 a = P1IN & 0x00; //Taste rechts
```

```
11 | a = P1IN & 0x01; //Taste rechts
12 | a = P1IN & 0x02; //Taste rechts
13 | a = P1IN & 0x03; //Taste links
14 | a = P1IN & 0x03; //beide Tasten
15 | P4OUT = P1IN & Taster_rechts; //Taster an P1.0 nicht
16 | P4OUT = P1IN & Taster_links; //Taster an P1.0
```

Alle Pins von Ports 1 werden auf Eingang geschaltet. Hier werden nun alle Pins von Ports 4 auf Ausgang geschaltet. An alle Pins von Port 4 werden die Logikpegel 0 angelegt.

A 1.2.4

A 1.3 Ampel

A 1.3.1

A 1.4 Taster

A 1.3.1