

Hardwarenahe Programmierung / Angewandte Informatik

Übungsaufgaben – 27. November 2017

Aufgabe 1: Ausgabe von Hexadezimalzahlen

Schreiben Sie eine Funktion `void print_hex (uint32_t x)`, die eine gegebene vorzeichenlose 32-Bit-Ganzzahl `x` als Hexadezimalzahl ausgibt. (Der Datentyp `uint32_t` ist mit `#include <stdint.h>` verfügbar.)

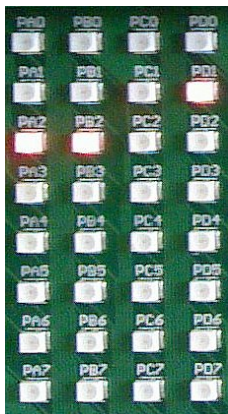
Verwenden Sie dafür *nicht* `printf()` mit der Formatspezifikation `%x` als fertige Lösung, sondern programmieren Sie die nötige Ausgabe selbst. (Für Tests ist `%x` hingegen erlaubt und sicherlich nützlich.)

Die Verwendung von `printf()` mit anderen Formatspezifikationen wie z. B. `%d` oder `%c` oder `%s` ist hingegen zulässig.

Aufgabe 2: Mikro-Controller

An die vier Ports eines ATmega16-Mikro-Controllers sind Leuchtdioden angeschlossen:

- von links nach rechts an die Ports A, B, C und D,
- von oben nach unten an die Bits Nr. 0 bis 7.



Wir betrachten das folgende C-Programm (Datei: [aufgabe-2.c](#)) für diesen Mikro-Controller:

```
#include <avr/io.h>
#include <avr/interrupt.h>

int counter = 0;

ISR (TIMER0_COMP_vect)
{
    PORTA = 1 << ((counter++ >> 6) & 7);
}

int main (void)
{
    cli ();
    TCCR0 = (1 << CS01) | (1 << CS00);
    TIMSK = 1 << OCIE0;
    sei ();
    DDRA = 0xff;
    while (1);
    return 0;
}
```

Das Programm bewirkt ein periodisches Lauflicht in der linken Spalte von oben nach unten. Eine Animation davon finden Sie in der Datei [aufgabe-2.gif](#).

- Wieso bewirkt das Programm überhaupt etwas, wenn doch das Hauptprogramm nach dem Initialisieren lediglich eine Endlosschleife ausführt, in der *nichts* passiert?
- Erklären Sie, wie die Anweisung
`PORTA = 1 << ((counter++ >> 6) & 7);`
das LED-Blinkmuster hervorruft.
Hinweis: Zerlegen Sie die eine lange Anweisung in mehrere kürzere.
Wenn nötig, verwenden Sie zusätzliche Variable für Zwischenergebnisse.
- Was bedeutet „`ISR (TIMER0_COMP_vect)`“?
- Wieso leuchten die Leuchtdioden PB2 und PD1?