Mikrocontroller: Bits und Bytes

Wir erzeugen ein Lauflicht

Für ein Lauflicht müssen an alle Anschlüssen von PortB Leuchtdioden angeschlossen werden: Dazu werden in die Buchsenleiste LEDs bei PortB.0, Portb.1, ..., PortB.7 angeschlossen. (Geduld: In der nächsten Stunde wirst Du das auch machen; jetzt kümmern wir uns um das entsprechende Programm!)
Was muss das Programm leisten? Erst soll die LED bei PortB.0 leuchten; nach 100 Millisekunden soll sie ausgehen und jetzt soll die LED bei PortB.1 leuchten. Nach weiteren 100 Millisekunden soll auch diese wieder ausgehen und nun soll die LED bei PortB.2 leuchten. Das Ganze soll weitergehen, bis die LED bei PortB.7 leuchtet. Diese soll auch nur für 100 Millisekunden leuchten; und danach soll alles wieder von vorne beginnen. Und weil die LEDs jetzt sehr rasch der Reihe nach an- und ausgehen, sieht das dann so aus, als ob das Licht laufen würde. Daher der Name Lauflicht!

Aufgabe 1: Schreibe ein entsprechendes Programm. (Tipps: Benutze den Warte-Befehl waitms 100 und eine Do-Loop-Schleife!)

Dein Programm ist sicherlich ganz schön lang, denn Du hast die LEDs einzeln an- und ausschalten müssen. Es geht aber auch einfacher. Man kann nämlich mit einem einzigen Befehl alle LEDs an PortB gleichzeitig ein- bzw. ausschalten. Ein Beispiel soll zeigen, wie das geht:

```
PortB = &B11101001
```

usw.

LED an PortB.0 einschalten
LED an PortB.1 ausschalten
LED an PortB.2 ausschalten
LED an PortB.3 einschalten

Beachte: Vor der Folge von Einsen und Nullen muss hier unbedingt "&B" stehen.

Aufgabe 2: Welche LEDs werden durch den Befehl PORTB = 10010110 eingeschaltet, welche werden ausgeschaltet?

Aufgabe 3: Es sollen nur die LEDs bei PortB.3, PortB.5 und PortB.7 leuchten. Gib den zugehörigen Befehl an.

Mit diesem neuen Befehl sieht der Anfang des Lauflicht-Programms nun so aus:

```
do
 PortB = &B00000001
 waitms 100
 PortB = &B00000010
 waitms 100
 PortB = &B00000100
...
```

Aufgabe 4: Schreibe das vollständige Programm in Dein Heft.

In der Ziffernfolge unseres neuen Port-Befehls können sinnvollerweise nur die Ziffern 0 und 1 stehen: Schließlich kann die LED nur eingeschaltet (1) oder ausgeschaltet (0) werden. Eine Folge von 8 Nullen oder

Mikrocontroller: Bits und Bytes

Einsen wird ein **Byte** genannt; die Nullen bzw. Einsen selbst werden auch **Bits** genannt.

Das Zweiersystem

Das Zweiersystem (manchmal auch Dualsystem oder Binärsystem genannt) besitzt nur **zwei** Ziffern: die Null und die Eins. In der Tat soll die Kennung &B vor der Ziffernfolge dem Compiler deutlich machen, dass es sich hier um Zahlen im Zweiersystem handeln soll. In der 5. Klasse hast Du schon einmal gelernt, wie man mit diesen Zahlen umgeht. Hier zur Wiederholung ein Vergleich zwischen dem Zehnersystem und dem Zweiersystem:

Zehnersystem	**Zweiersystem**
10 Ziffern: 0, 1, 2, ..., 9	2 Ziffern: 0 und 1
Jede Stelle hat einen eigenen Wert, von rechts nach links: Einer, Zehner, Hunderter,	Jede Stelle hat einen eigenen Wert, von rechts nach links: Einer, Zweier, Vierer, Achter, ...
0	&B0
1	&B1
2	&B10
3	&B11
4	&B100

Auch beim Zweiersystem müssen führende Nullen nicht unbedingt hingeschrieben werden.

Wie lautet nun &B10001101 im Zehnersystem? Für die Umrechnung schreiben wir die Stufenzahlen 1, 2, 4, 8, ... über die zugehörigen Stellen:

	128	64	32	16	8	4	2	1	
Zweiersystem &B...	**1**	**0**	**0**	**0**	**1**	**1**	**0**	**1**	
Zehnersystem	128				+8	+4		+1	141

Aufgabe 5: Rechne um ins Zehnersystem: &B11001100, &B10101010, &B10000001

Aufgabe 6: Die größte Zahl, die man mit 8 Bits darstellen kann, ist &B11111111. Wie lautet diese Zahl im Zehnersystem?

Der Compiler kann den neuen Portbefehl auch mit Zahlen im Zehnersystem verarbeiten. Beispiel: PortB = 137 (ohne &B). Allerdings muss er beim Compilieren diese Zahl ins Zweiersystem umwandeln.

Aufgabe 7: Überlege oder probiere aus, wie 137 im Zweiersystem lautet. Welche LEDs müssten demnach durch PortB = 137 zum Leuchten gebracht werden?