

1.3 Atmega-Programmierung in ASM/Modell des Mikrocontrollers

1.3.1 Zusammenhänge

Was bedeuten nun die Befehle im ersten Programm? Und was passiert im Mikrocontroller während des Programms?

Man kann sich den Mikrocontroller vorstellen als eine Anordnung von Funktionsblöcken, die durch Busleitungen miteinander verbunden sind (Abbildung 1). Einige der Blöcke sind mit physikalischen Ein-/Ausgangsleitungen verbunden. Sie werden Ein-/Ausgaberegister genannt. Die Zuordnung der Ein-/Ausgaberegister zu den Ein-/Ausgangsleitungen ist in Abbildung 2 aufgeführt.

Außerhalb des Mikrocontrollers sind die Ausgangsleitungen an diskrete physikalische Hardware angeschlossen. Abbildung 3 zeigt die Zuordnung für die LEDs und den Summer. In Tabelle 1 sowie in der Beschreibung des Board-Herstellers findet man die Zuordnungen für die Taster und einige andere Peripherie.

Funktion	Port	Ein/Aus
LED 7	PC7	S1.8
LED 6	PC6	S1.7
...
LED 0	PC0	S1.1
Summer	PD7	–
RS232 RX	PD0	–
RS232 TX	PD1	–
Taster	PA7	–

Tabelle 1: Hardware des benutzten Boards (Auszug)

1.3.2 Analyse des Programms

Das Ansprechen der Funktionsblöcke im Mikrocontroller geschieht durch Befehle im Programm, und zwar nach den Spielregeln (=Befehlssatz) des Mikrocontrollers:

- Die `include`-Anweisung zu Beginn entspricht der gleichnamigen Präprozessoranweisung in einem C-Programm. Nur die Schreibweise ist etwas anders.
- Der Befehl `ldi r16, 0xb1111111` lädt die unmittelbar angegebene Zahl `0b11111111=0xFF=255` in das Universalregister `r16`. Für diesen Befehl kann man jedes der acht Bit breiten Universalregister (*general purpose register*, GPR) `r16` bis `r31` nehmen.¹ Die Zahl darf zwischen 0 bis 255 liegen und binär, oktal, dezimal oder hexadezimal angegeben werden. Ein Universalregister entspricht also einer Variablen vom Typ `unsigned char` in C.
- Der Befehl `out DDRC, r16` gibt den Inhalt des Universalregisters `r16` in das Datenrichtungsregister `DDRC`. Genauer gesagt: Im Programm ist `DDRC` nur eine symbolische Konstante mit dem Wert 14. Und dieser Wert 14 ist die Adresse des Datenrichtungsregisters mit dem Namen `DDRC`. Die Datenrichtungsregister gehören zu den Spezialregistern (*special function registers*, SFR). Das Datenrichtungsregister mit dem Namen `DDRC` gehört zum Ein-/Ausgabeport C: Mit dieser Zeile werden alle acht Bits des Ein-/Ausgabeports C auf *Ausgabe* gestellt (1=Ausgabe, 0=Eingabe).
- Der Befehl `out PORTC, r16` gibt den Inhalt des Universalregisters `r16` in das Ein-/Ausgaberegister `PORTC` (=15). Auch die Ein-/Ausgaberegister gehören zu den Spezialregistern (SFR). Mit dieser Zeile werden zwei der acht Bits des Ein-/Ausgabeports C auf Low-Potential gebracht und die anderen sechs auf High-Potential. Die beiden angesprochenen LEDs leuchten.

¹Es gibt noch 16 weitere Universalregister mit den Namen `r0` bis `r15`, aber der Befehl `ldi` kann sie nicht verwenden.

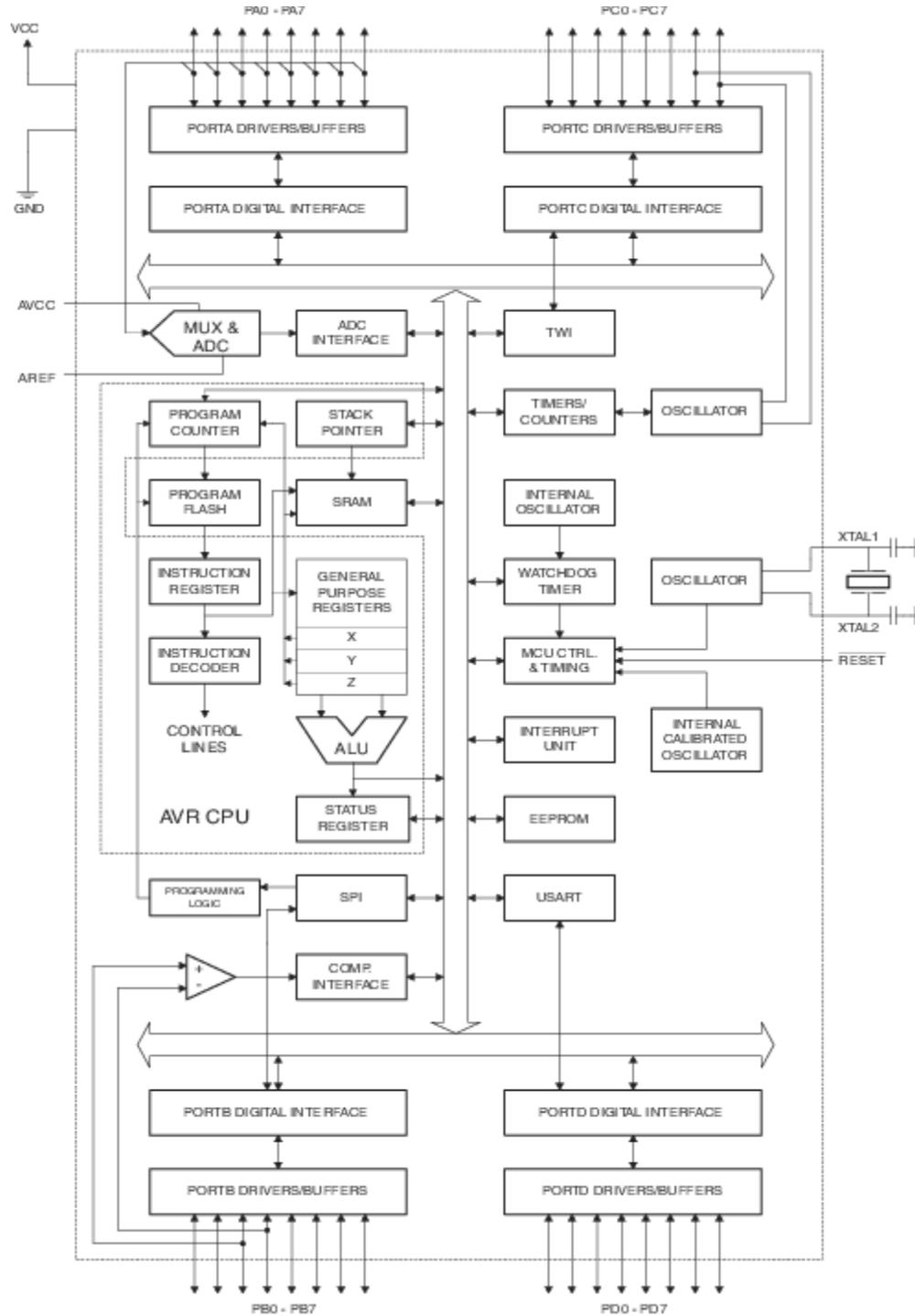


Abbildung 1: Blockschaltbild ATmega32 (Quelle: Atmel)

PB0	1	40	PA0
PB1	2	39	PA1
PB2	3	38	PA2
PB3	4	37	PA3
PB4	5	36	PA4
PB5	6	35	PA5
PB6	7	34	PA6
PB7	8	33	PA7
/RESET	9	32	Uref (ADC)
VCC	10	31	0V (ADC)
GND	11	30	+5V (ADC)
XTAL2	12	29	PC7
XTAL1	13	28	PC6
PD0	14	27	PC5
PD1	15	26	PC4
PD2	16	25	PC3
PD3	17	24	PC2
PD4	18	23	PC1
PD5	19	22	PC0
PD6	20	21	PD7

Abbildung 2: Zuordnung E-/A-Register zu Anschlüssen (Quelle: Atmel)

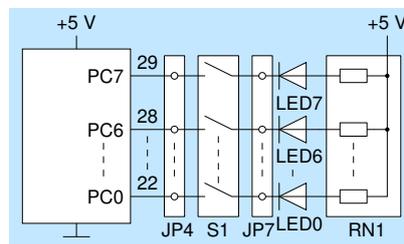


Abbildung 3: Hardware des benutzten Boards (Auszug)

- Der Befehl `rjmp ende` springt an die Stelle des Programms, die durch die Marke `ende` gekennzeichnet wurde. Die Marke steht in der Zeile davor: ein beliebiges Wort, gefolgt von einem Doppelpunkt.

Tabelle 2 zeigt eine Übersicht über diese Befehle. Die Befehle (Programmstruktur) und Register

Befehl	a	b
<code>ldi a, b</code>	Univ.-Register r16–r31	Zahl 0-255
<code>out a, b</code>	SFR-Nummer	Univ.-Register r0-r31
<code>rjmp a</code>	Sprungmarke	—

Tabelle 2: Bisher benutzte Befehle

(Datenstruktur) hängen auch hier voneinander ab bzw. sind aufeinander bezogen. Daher muss bei den Befehlen vermerkt sein, auf welche Register sie wie anwendbar sind und wirken.

Es werden dazu nicht nur die Ein-/Ausgaberegister des uC gebraucht, sondern auch einige andere Blöcke. Manche werden im Programmtext erwähnt und sind daher in der Programmierung wichtig. Andere tauchen nicht auf. Letztere nennt man *transparent*.

Eine Übersicht über die bisher benutzten Register zeigt Tabelle 3.

Name									Adresse	
r16	Bit	7	6	5	4	3	2	1	0	GPR 16
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	Init.	0	0	0	0	0	0	0	0	
r17	Bit	7	6	5	4	3	2	1	0	GPR 17
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	Init.	0	0	0	0	0	0	0	0	
DDRC	Bit	7	6	5	4	3	2	1	0	SFR 14
	R/W	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	
	Init.	0	0	0	0	0	0	0	0	
PORTC	Bit	7	6	5	4	3	2	1	0	SFR 15
	R/W	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	
	Init.	0	0	0	0	0	0	0	0	

Tabelle 3: Bisher benutzte Register