

Soll ein Microcontrollermodul auch mobil genutzt werden, ist es sinnvoll, ihn in Pausen, wenn der Controller eigentlich nichts zu tun hat, in einen Schlafmodus mit reduzierter Stromaufnahme zu versetzen. BASCOM bietet dazu einige Optionen wie "Idle", "Powersave" und "Powerdown" an. Die BASCOM-Help gibt sich dazu aber recht bedeckt. Also arbeiten wir uns wieder einmal an dem Datenblatt ab. Hier wieder für die ATmega48/88/168/328-Familie in Kap. 9. Bei älteren Controllern wie den ATmega8/16/32 sieht das etwas anders aus.

Mit diesen Maßnahmen lässt sich vorab schon einmal Strom sparen:

- 1 Stromversorgung kleiner als 5V-Standard.  
Bei sehr geringen Spannungen ist i.a. auch die Taktfrequenz zu reduzieren, z.B. max. 4MHz bei 1,8V für den ATmega328P.
- 2 Wahl der Taktquelle, Taktfrequenz reduzieren.  
Wenn keine quarzgenauen Frequenzen notwendig sind, z.B. für Timings, und eine etwas gemächliche Gangart ausreicht, steht der interne RC-Taktgenerator, je nach Controller 1 oder 8MHz zur Auswahl. Dieser spart auch wieder Strom gegenüber einem externen Quarz, bei einem ATmega328P mit 16MHz Quarz gemessen ca. 5mA. Außerdem wacht der Controller mit RC-Takt schneller wieder auf.
- 3 Alle nicht benutzten I/O-Ports auf Input mit internem Pullup konfigurieren.
- 4 ADC deaktivieren, wenn er nicht gebraucht wird. Ist standardmäßig aktiviert.

Register ADCSRA – ADC Control and Status Register A (Kap. 24.9.2)

Bit	7	6	5	4	3	2	1	0
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0

**Bit 7 - ADEN: ADC Enable**

= 1: Aktiviert den ADC

= 0: Deaktiviert den ADC

- 5 Analog Comparator deaktivieren

Register ACSR – Analog Comparator Control and Status Register (Kap. 23.3.2)

Bit	7	6	5	4	3	2	1	0
	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	N/A	0	0	0	0	0

**Bit 7 - ACD: Analog Comparator Disable**

= 1: Deaktiviert den Analog Comparator

= 0: Aktiviert den Analog Comparator

- 6 Powermanagement konfigurieren

Register SMCR – Sleep Mode Control Register (Kap. 10.11.1)

Bit	7	6	5	4	3	2	1	0
	-	-	-	-	SM2	SM1	SM0	SE
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0

Bits 3 ... 1: Sleep Mode Select Bits SM2 ... SM0

SM2	SM1	SM0	Sleep Mode
0	0	0	Idle
0	0	1	ADC Noise Reduction
0	1	0	Power-down
0	1	1	Power-save
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Standby
1	1	1	Extended Standby

Im wirkungsvollsten Power-down-Modus werden alle Oszillatoren (außer Watchdog, falls per Fuses aktiviert) angehalten. Externe Interrupts (INT0, INT1 und Pin change) können den Controller wieder aufwecken, hier INT0, Set-Taster an PD.2/INT0, und PCINT11, Alarmausgang INT des PCF8563 an PC.3/PCINT11.

**Bit 0 – SE: Sleep Enable**

= 1: Versetzt den Controller in den Schlafmodus, wenn der Sleep-Befehl ausgeführt wird.

## 7 Brown-out Detector deaktivieren

Das zeitweilige Beschreiben des EEPROMs mit den Alarmzeiten wurde mit dem Brown-out-Fuse abgesichert. Ein Abschalten im Schlafmodus spart zusätzlich Strom. Das geht mit den ATmega 48/88/168/328, bei anderen Controllern ggf. nicht.

MCUCR – MCU Control Register (Kap. 12.5.1)

Bit	7	6	5	4	3	2	1	0
	-	BODS	BODSE	PUD	-	-	IVSEL	IVCE
R/W	R	R/W	R/W	R/W	R	R	R/W	R/W
Initial	0	0	0	0	0	0	0	0

**Bit 6 – BODS: BOD Sleep**

=1: Schalte Brown-out Detection (BOD) im Schlafmodus ab.

**Bit 5 – BODSE: BOD Sleep Enable**

Zum Abschalten der Brown-out Detection müssen BODS und BODSE beide zuerst auf 1 gesetzt werden. Anschließend müssen BODS = 1 und BODSE = 0 innerhalb von vier Takten, also unmittelbar danach gesetzt werden.

## 8 Power Reduction Register PPR

Auf weitere Möglichkeiten mit einzelnen Einstellungen im Power Reduction Register wird hier verzichtet.

Das ganze zusammengerührt:

```

Sub GoodNight(byVal bytOpt As Byte)
'Power down ATmega 48P/88P/168P/328P
'Executed are:
'- Enable INT0 key Set interrupt for wake up
'- Disable ADC
'- Disable Analog Comparator
'- Several sleep modes according to bytOpt
'- Disable Brown Out Detection
'Current reduced by ~20mA @ Vcc=5V and external crystal 16MHz with "Power down"
'Input:  bytOpt   =0: Mode Idle
'        =1: Mode ADC Noise Reduction
'        =2: Mode Power down
'        =3: Mode Power save
'        =4: Mode Standby
'        =5: Mode Extended Standby

Select Case bytOpt
  Case 1                                'ADC Noise Reduction
    bytMask = &B0000_0010
  Case 2                                'Power down
    bytMask = &B0000_0100
  Case 3                                'Power save
    bytMask = &B0000_0110
  Case 4                                'Standby
    bytMask = &B0000_1100
  Case 5                                'Extended Standby
    bytMask = &B0000_1110
  Case Else                              'Idle
    bytMask = &B0000_0000
End Select

EIMSK.INT0 = 1                          'Enable key Set interrupt for wake up

'Disable not used components
ADCSRA.ADEN = 0                          'Disable ADC (ADCSRA Bit 7)
ACSR.ACD = 1                             'Disable Analog Comparator (ACSR Bit7)

'Enable Sleep
SMCR = SMCR OR bytMask                    'Set SM0 to SM2 (Bits 1 to 3)
SMCR = SMCR OR &B0000_0001               'Set SM (Bit0): Sleep enable

'Disable Brown out detection BOD
MCUCR = MCUCR OR &B0110_0000             'Set BODSE (Bit5) & BODS (Bit6)
MCUCR = MCUCR AND &B1101_1111           'Reset BODSE (Bit5)
!Sleep                                    'ASM Sleep

End Sub

```

Mit dem Statement "EIMSK.INT0 = 1" wird noch der externe Interrupt INT0 (Port PD2, Set-Taster) im External Interrupt Mask Register EIMSK aktiviert, um den Controller auch mit dem Set-Taster wieder aufwecken zu können.

Das Assembler-Statement "!Sleep" versetzt den Controller schließlich in den Tiefschlaf. Das Programm stoppt sofort mit Ausführung des Sleep-Befehls, bleibt also an dieser Stelle stehen.

Aufgeweckt wird der Controller fallweise durch zwei externe Interrupts:

- 1 Alarmsignal am Ausgang INT des PCF8563 über Pin change Interrupt PCINT11 oder
- 2 Druck auf den Set-Taster (Interrupt INT0).

Unmittelbar nach dem Aufruf der Routine "GoodNight", also nach dem "!Sleep"-Statement, muss die Fortführung des Programms nach dem Aufwachen sichergestellt werden. Hier mit dem Statement "`EIMSK.INT0 = 0`", das mit Deaktivierung des INT0-Interrupts den Set-Taster an PD2/INT0 wieder für das normale Polling über den Timer0-Interrupt nutzbar macht.