

Ausarbeitung

Mikrocontroller ATmega16



Verfasst von:

Armagan Hamza
Universität Koblenz
Fachbereich Informatik

Inhalt

1. Systemtakt
 - Einführung
 - Systemtakt und die dazugehörigen Komponenten
 - Zielkomponenten der AVR Clock Control Unit
 - Quellkomponenten der AVR Clock Control Unit
2. Der Watchdog Timer
 - Einführung
 - Eigenschaften
 - Arbeitsweise
3. Power Management und Sleep – Modi
 - Einführung
 - Arbeitsweise
4. Anhang
 - Fuse Bits
 - Das Oszillator Kalibrierungsregister OSCCAL
 - Das MCU Control Register – MCUCR
 - Das Watchdog Timer Control Register – WDTCR
 -
5. Quellen

Einführung

Kaum noch ein technischer Bereich unseres Lebens, ob Multimedia-, Kommunikations-, Automobil- oder Industriemarkt, in den nicht Mikrocontroller die wichtigsten Steuerfunktionen übernommen haben. Dabei stellen die neuen, immer leistungsfähigeren Produkte wie SAT-Receiver, Set-Top-Box, Internetbox, DVD - Player, GPS-Navigations-systeme und Mobiltelefone stetig steigende Anforderungen an Mikrocontroller und zu entwickelnde Software.

Bei den nach Register- und Datenbusbreite in 8-, 16- und 32 Bit eingeteilten Mikrocontrollern, die auch nur kurz MCU's (Microcontroller Unit) genannt werden, sind bis jetzt noch die 8 und 16 >>Bitter<< in nahezu 80% aller Anwendungsfälle vertreten. Der Anteil der 32 >>Bitter<< in Embedded Systems steigt vor allem in hoch integrierten Chips wie MPEG- Decodern oder GPS-Aufbereitungen, bei denen nur eine Komponente die 32 Bit CPU ist. Für viele Anwendungsfälle bleibt jedoch die 8-Bit MCU noch immer aus Preis-Leistungs-Sicht die erste Wahl. Untersuchungen belegen, dass die Nachfrage nach 8-Bit Mikrocontrollern noch weit bis ins neue Jahrhundert anhalten wird.

Nicht verwunderlich also dass der inzwischen über 20 Jahre alte Mikrocontroller Markt mit neuen Familien und Architekturen belebt. So sind unter anderem in den letzten Jahren neue 8

Bitter wie PIC, AVR und 16 Bit-Familien wie H8S, XA, C16, M16C hinzugekommen. Mit Sicherheit werden weitere folgen.

Im Rahmen meines Seminars habe ich mich mit dem ATMEL 8-bit AVR Mikrocontroller ATmega16 beschäftigt. Zu meinen Themen gehört der Systemtakt, Power Management und Sleep Modi und der Watchdog Timer.

1. Systemtakt

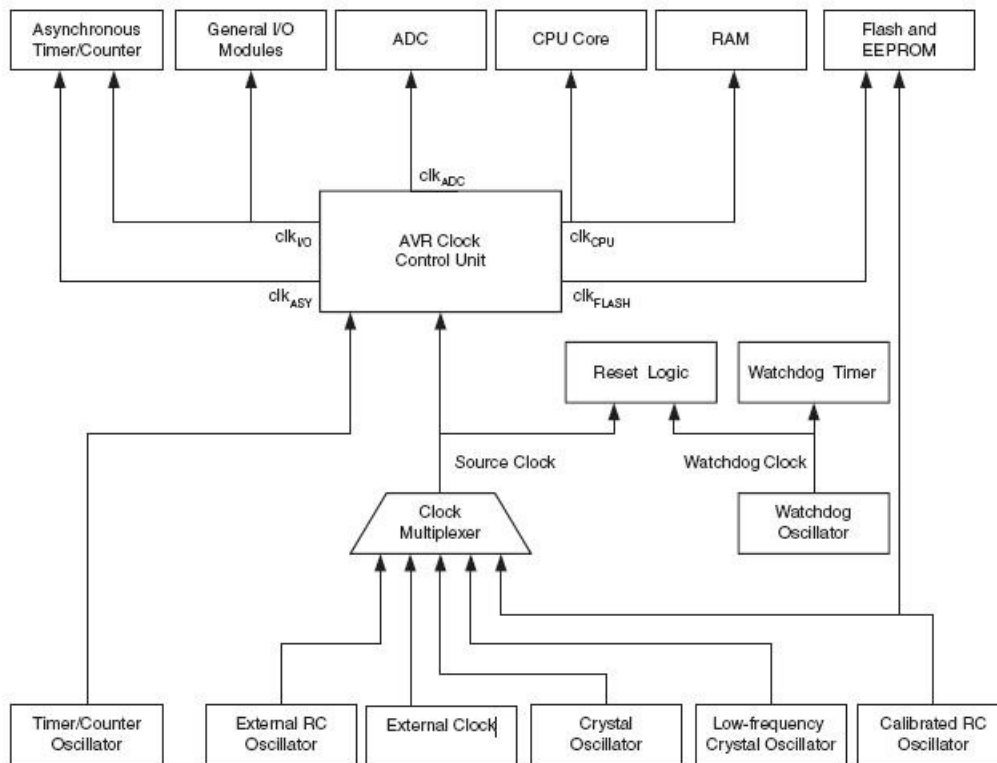
Einführung

Für die Erzeugung des Systemtaktes ist bei den Controllern der AVR – Familie ein integrierter Oszillator verantwortlich, der über die Anschlüsse XTAL1 und XTAL2 mit dem System. Bzw. externen Komponenten kommuniziert. XTAL1 und XTAL2 sind dabei Ein- bzw. Ausgänge eines invertierenden Verstärkers, der als On – Chip – Oszillator zur Erzeugung des Systemtaktes verwendet werden kann. Man kann die Schwingung entweder mit einem Quarz oder mit einem keramischen Resonator erzeugen. Alternativ dazu kann auch ein externes Taktsignal zur Erzeugung des Systemtaktes herangezogen werden. In diesem Fall wird das Signal in den Eingang XTAL1 eingespeist und der Ausgang XTAL2 des internen Oszillators bleibt offen.

Systemtakt und die dazugehörigen Komponenten

Das nachfolgende Bild zeigt den prinzipielle Aufbau des Systemtaktes im AVR mit den dazugehörigen Komponenten. Im zentralen Punkt steht die AVR Clock Control Unit. Das ist die zentrale Stelle wo verschiedene Komponenten mit unterschiedlichem Systemtakt versorgt werden. Als Quelle für diese AVR Control Unit kann man wiederum aus verschiedenen Quellen aussuchen, was ich im folgenden auch beschreiben werde. Was ich hier noch erwähnen möchte ist das es bei dieser Ansteuerung möglich sein soll, bestimmte Systemtakte abzuschalten. Das wiederum gewährleistet eine Senkung des Stromverbrauches was natürlich ein wichtiger Faktor bei den Controllern ist. Dafür sind verschiedene Sleep Modi vorgesehen auf denen später eingegangen wird.

Bild 1: AVR clock control Unit und die dazugehörigen Komponenten



Zielkomponenten der AVR Clock Control Unit

Bei dem Atmega16 werden 6 Komponenten (Asynchronous Timer/Counter, General I/O Module, ADC, CPU Core, RAM, Flash and EEPROM) mit 5 verschiedenen Taktleitungen versorgt. Bei den Takten, die von der AVR Clock Control Unit ausgehen handelt es sich dabei um den clk_{CPU} , clk_{FLASH} , clk_{IO} , clk_{ASY} und den clk_{ADC} Takt. Im folgenden werden die Takte mit den dazugehörigen Eigenschaften näher beschrieben.

CPU Takt - clk_{CPU}

Der CPU Takt ist mit Systemkomponenten verbunden, die mit AVR Befehlen arbeiten. Solche Komponente sind z.B. das General Purpose Register File, das Status Register und das Data Memory der den Stack Pointer beinhaltet. Wenn der CPU Takt angehalten wird, dann verliert der CPU Kern an Performanz bezüglich allgemeine Befehlsabarbeitung und Berechnungen.

I/O Takt - clk_{IO}

Der I/O Takt wird zu einem großen Teil der I/O Modulen wie Timer/Counter, SPI und der USART benutzt. Außerdem wird der I/O Takt von externen Modulen benutzt. Einige externe Interrupts werden von einer asynchronen Logik erkannt. Solchen Interrupts muss man

gewährleisten erkannt zu werden. Auch wenn der I/O Takt angehalten wurde. Deswegen wird die Adresserkennung nach im TWI Modul asynchron nach außen getragen. Sobald die $clk_{I/O}$ angehalten wurde ist der Empfang von TWI Adressen in allen Sleep Modi zu aktivieren.

Flash Takt – clk_{FLASH}

Der Flash Takt überwacht die Ausführungen vom Flash Interface. Der Takt vom Flash ist gewöhnlich zu gleichen Zeit mit der CPU aktiv.

Asynchroner Timer Takt – clk_{ASY}

Der asynchroner Timer Quarz erlaubt dem asynchronen Timer/Counter direkt von einer externen 32 kHz Takt Quarz getaktet zu werden. Der entsprechende Takt Bereich erlaubt diesen Timer/Counter als einen Real Time Counter zu benutzen. Sogar dann, wenn sich das Gerät im Sleep Modus befindet.

ADC Takt – clk_{ADC}

Der ADC wird von einem bestimmten Taktbereich versorgt. Dies ermöglicht das Anhalten der CPU und I/O Takt. Das Ziel hierbei ist, den Störungen zu verringern, die von digitalen Schaltkreisen erzeugt werden. Dadurch werden dann genauere ADC Umwandlungsergebnisse erzielt.

Quellkomponenten der AVR Clock - Control Unit

Beim ATmega16 hat man die Möglichkeit zwischen 6 verschiedene Taktquellen als Eingang für die AVR Clock – Control Unit zu wählen. Dabei handelt es sich um einen Timer/Counter Oszillator, Externer RC – Oszillator, Externer Takt, Quarz Oszillator, Low Frequenz Quarz Oszillator, kalibrierter interner RC Oszillator. Die Taktquellen kann man anhand der CKSEL Bits im Fuse Low Byte wählen.

Device Clocking Options Select⁽¹⁾

Device Clocking Option	CKSEL3..0
External Crystal/Ceramic Resonator	1111 – 1010
External Low-frequency Crystal	1001
External RC Oscillator	1000 – 0101
Calibrated Internal RC Oscillator	0100 – 0001
External Clock	0000

1. For all fuses „1“ means unprogrammed while „0“ means programmed

Die genaueren Eigenschaften der verschiedenen Taktversorgungsquellen werden weiter unten beschrieben.

Wenn die CPU von einem Power – Down oder Power – Save Modus aufgeweckt wird, dann wird von der ausgewählten Eingangstaktquelle eine Start up Time gemessen. Diese Start up Time ist für eine stabile Oszillator Befehlsausführung notwendig. Wenn die CPU aber von einem Reset aus gestartet wird, dann gibt es noch ein zusätzliche Verzögerung, die den Strom

erlauben soll einen stabilen Zustand zu erreichen. Die dabei erforderliche Zeit wird als Time Out bezeichnet. Die Anzahl der WDT Oszillator Zyklen, die für einen Time Out benutzt werden, sind in der folgenden Tabelle aufgeführt. Dabei ist zu beachten, dass die Frequenz des Watchdog Oszillators und somit auch die Time Out von der Versorgungsspannung Vcc abhängig ist.

Number of Watchdog Oscillator Cycles

Typ Time-out (Vcc = 5.0 V)	Typ Time – out (Vcc = 3.0V)	Number of Cycles
4.1 ms	4.3 ms	4K (4,096)
65 ms	69 ms	64K (65,536)

Default Taktquelle

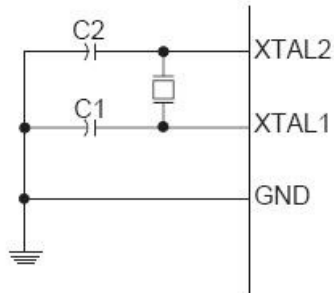
Das Gerät wird dabei mit CKSEL = „0001“ und SUT = „10“ programmiert. Für die Default Taktquelle ist der 1 MHz interne RC Oszillator vorgesehen, der die längste Start up Time hat. Anhand dieser Default Taktquelle hat jeder Anwender die Möglichkeit seine Wunschtaktquelle auszuwählen, indem man einen In – System oder Parallel Programmierer verwendet.

Quarz Oszillator

Der Eingang bzw. Ausgang XTAL1 und XTAL2 kommt von einem invertierendem Verstärker. Dieser kann als ein On – Chip Oszillator eingestellt werden. Dabei hat man die Möglichkeit zwischen einem Quarzoszillator und einem Keramik Resonator zu wählen. Das CKOPT Bit wählt zwischen 2 verschiedenen Oszillator Verstärker Modi aus. Wenn das CKOPT programmiert ist, dann wird der Oszillatorausgang mit seiner vollen Leistung arbeiten. Dieser Modus ist notwendig wenn sich das System in einer sehr störrischen Umgebung oder wenn der XTAL2 Ausgang einen 2. Takt ansteuert. Dieser Modus hat einen weiten Frequenzbereich. Wenn das CKOPT Bit nicht programmiert ist, dann hat der Oszillator eine kleinere Ausgangsschwingung. Das verringert folglich den Stromverbrauch. Dieser Modus hat einen begrenzten Frequenzbereich und kann keine weiteren Taktquellen ansteuern.

Mit Resonatoren kann man eine maximale Frequenz von 8 MHz erzielen, wenn das CKOPT Bit nicht programmiert ist. Mit dem programmierten CKOPT Bit erreicht man eine maximale Frequenz von 16 MHz. Dabei sollte der Werte der Kondensatoren (C1, C2) für beide Quarze und Resonatoren gleich sein. Der ideale Wert für den Kondensator hängt davon ab, ob man einen Quarz oder Resonator benutzt. Hinzu kommen die elektromagnetischen Störungen der Umgebung die den Wert auch beeinflussen. Einige grundlegende Richtwerte bei der Auswahl von Kondensatoren werden in der nachfolgenden Tabelle gegeben.

Bild2: Verbindungen Quarz Oszillator



Der Oszillator kann in 3 verschiedene Modi arbeiten. Jeder dieser Modi ist für eine spezifischen Frequenzbereich abgestimmt. Der jeweilige Arbeitsmodus ist anhand der Bits CKSEL3..1 wie in der Tabelle angezeigt auswählbar.

Crystal Oscillator Operating Modes

CKOPT	CKSEL3..1	Frequency Range (MHz)	Recommended Range for Capacitors C1 and C2 for Use with Crystals (pF)
1	101 ⁽¹⁾	0.4 – 0.9	-
1	110	0.9 – 3.0	12 – 22
1	111	3.0 – 8.0	12 – 22
0	101,110,111	1.0 <=	12 – 22

1. This option should not be used with crystals, only with ceramic resonators

Das CKSEL0 Fuse Bit bestimmt zusammen mit dem SUT1..0 Bits die Start – up Time.

Start – up Times for the Crystal Oscillator Clock Selection

CKSEL0	SUT1..0	Start-up Time from Power – down and Power - save	Additional Delay Form Reset (Vcc = 5.0V)	Recommended Usage
0	00	258 CK ⁽¹⁾	4.1 ms	Ceramic resonator, fast Rising power
0	01	258 CK ⁽¹⁾	65 ms	Ceramic resonator, slowly Rising power
0	10	1K CK ⁽²⁾	-	Ceramic resonator, BOD Enabled
0	11	1K CK ⁽²⁾	4.1 ms	Ceramic resonator, fast Rising power
1	00	1K CK ⁽²⁾	65 ms	Ceramic resonator, slowly Rising power
1	01	16K CK	-	Crystal Oscillator, BOD enabled
1	10	16K CK	4.1 ms	Crystal Oscillator, fast Rising power
1	11	16K CK	65 ms	Crystal Oscillator, slowly Rising power

1. This options should only be used when not operating close to the maximum frequency of the device, and only if frequency stability at start-up is not important for the application. These options are not suitable for crystals.
2. These options are intended for use with ceramic resonators and will ensure frequency stability at start-up. They can also be used with crystals when not operating close to the maximum frequency of the device, and if frequency stability at start – up is not important for the application.

Quarz Oszillator mit niedriger Frequenz

Es gibt auch eine Möglichkeit den Controller mit einer niedrigen Quarzfrequenz zu betreiben. Um einen 32.768 kHz Quarzuhr als eine Taktquelle zu benutzen müssen die CKSEL Bits auf „1001“ gesetzt sein. Der Quarz muss dann wie im Bild 2 angeschlossen werden. Wenn man das CKOPT Bit im Fuse Byte programmiert wird dann ein interner Kondensator zwischen XTAL1 und XTAL2 geschaltet. Demzufolge muss dann der externe Kondensator entfernt werden. Die Kapazität des internen Kondensators beträgt dabei 36 pF.

Wenn dieser Oszillator als Quelle ausgewählt wurde, dann wird die Start up time anhand der SUT Bits wie folgt bestimmt.

Start – up Times für the Low-frequency Crystall Oscillator Clock Selection

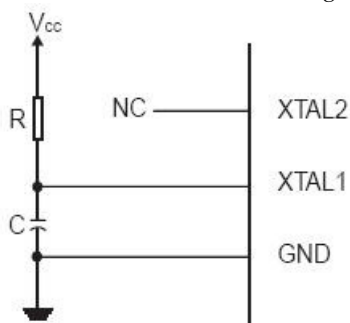
SUT1..0	Start – up Timer from Power – down and Power – save	Additional Delay From Reset (Vcc = 5.0V)	Recommended Usage
00	1K CK (1)	4.1 ms	Fast rising power or BOD enabled
01	1K CK (1)	65 ms	Slowly rising power
10	32K CK	65 ms	Stable frequency at start - up
11	Reserved	Reserverd	Reserverd

1. These options should only be used if frequency stability at start – up is not important for the application

Externer RC Oszillator

Für Anwendungen die viel Zeit beanspruchen gibt es auch die Möglichkeit eine externe RC Beschaltung zu machen. Die Frequenz wird hierbei auf $f = 1 / (3RC)$ abgeschätzt. Die Kapazität sollte mindestens 22 pF sein. Der interne Kondensator mit 36 pF wird zwischen XTAL1 und XTAL2 geschaltet wenn das CKOPT Bit im Fuse Byte programmiert wird. Demzufolge muss der externe Kondensator entfernt werden.

Bild3: externe RC Beschaltung



Der Oszillator hat hierbei die Möglichkeit mit 4 verschiedenen und spezifisch angepassten Frequenzbereichen zu arbeiten.

External RC Oscillator Operating Modes

CKSEL3..0	Frequency Range (MHz)
0101	0.1 <= 0.9
0110	0.9 – 3.0
0111	3.0 – 8.0
1000	8.0 – 12.0

Bei diesem Oszillatortyp bestimmen die SUT Bits im Fuse Byte über die Start up Time.

Start – up Times for the External RC Oscillator Clock Selection

SUT1..0	Start-up Timer from Power-down and Power - save	Additional Delay From Reset (Vcc = 5.0 V)	Recommended Usage
00	18 CK	-	BOD enabled
01	18 CK	4.1 ms	Fast rising power
10	18 CK	65 ms	Slowly rising power
11	6 CK ⁽¹⁾	4.1 ms	Fast rising power or BOD enablede

1. This option should not be used when operating close to the maximum frequency of the device

Kalibrierter interner RC Oszillator

Der kalibrierte interne RC Oszillator versorgt die AVR Clock Control Unit mit einer festen Taktfrequenz von 1.0, 2.0, 4.0 oder 8.0 MHz. Bei den ganzen Frequenzen handelt es sich um Sollwerte bei 5 V und 25 Grad. Für den gewünschten Frequenzbereich sind die CKSEL3..0 wie in der folgende Tabelle zu programmieren. Bei diesem kalibriertem internen RC Oszillator Modus wird der Controller mit keiner anderen externen Oszillator arbeiten. Bei dieser Taktung sollte der CKOPT Bit immer nicht programmiert sein. Während dem Reset lädt die Hardware das Kalibrierungsbyte in das OSCCAL Register. Demzufolge wird der RC Oszillator automatisch kalibriert. Es wird einen Kalibrierungsspielraum von plus/minus 3 % gewährleistet bei 5V, 25 Grad und 1 MHz Frequenzbereich.

Internal Calibratet RC Oszillator Operating Modes

CKSEL3..0	Nominal Frequency (MHz)
0001 ⁽¹⁾	1.0
0010	2.0
0011	4.0
0100	8.0

1. The device is shipped with this option selected

Bei diesem Oszillatortyp werden die Start – up Times auch anhand der SUT Bits im Fuse Byte bestimmt. Der XTAL1 und XTAL2 Pin sollt offen bleiben (NC).

Start up Times for the Internal Calibrated RC Oscillator Clock Selection

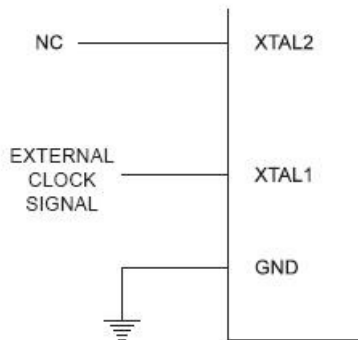
SUT1..0	Start – up Time from Power – down and Power – save	Additional Delay From Reset (Vcc = 5.0 V)	Recommended Usage
00	6 CK	-	BOD enabled
01	6 CK	4.1 ms	Fast rising power
10 ⁽¹⁾	6 CK	65 ms	Slowly rising power
11	Reserved	Reserved	Reserved

1. The device is shipped with this option selct

Externer Takt

Um den Controller mit einem externen Takt zu versorgen sollte das Gerät wie im Bild angeschlossen werden. Zudem müssen die CKSEL Bits mit „0000“ programmiert werden. Mit der Programmierung von CKOPT Bit kann noch ein interner 36pF Kondensator zwischen XTAL1 und GND geschaltet werden.

Bild4: externe Taktansteuerung



Bei dieser Art von Taktansteuerung werden die Start – up Times auch anhand der SUT Fuse Bits bestimmt.

Start-up Times for the External Clock Selection

SUT1..0	Start-up Times from Power – down and Power - save	Additional Delay From Reset (Vcc = 5.0V)	Recommended Usage
00	6 CK	-	BOD enabled
01	6 CK	4.1 ms	Fast rising power
10	6 CK	65 ms	Slowly rising power
11	Reserved	Reserved	Reserved

Wenn man sich für diesen externen Taktbeschaltung entscheidet, dann soll man schnelle Veränderungen in der Taktfrequenz vermeiden. Damit gewährleistet man eine sichere MCU Befehlsausführungen. Eine Abweichung der Frequenz von mehr als 2 % von einer zum anderen Taktzyklus kann schon ein unvorhersagbares Verhalten hervorbringen. In solch einem Fall muss es gewährleistet werden die MCU zu Reseten.

Timer/Counter Oszillator

Für die AVR Mikrocontroller die mit Timer/Counter Oszillator (TOSC1 und TOSC2) Pin bestückt sind kann der Quarz direkt zwischen den beiden Pins verbunden werden. Dann werden keinen externen Kondensatoren benötigt. Der Oszillator ist dann angepasst auf 32.768 kHz Quarzuhr.

2. Der Watchdog Timer

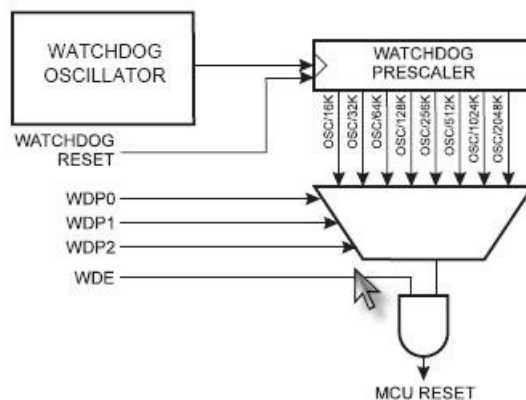
Einführung

Der Watchdog Timer ist dazu gedacht, um den Controller neu zu starten, wenn sich das Programm in einem undefiniertem Zustand befindet.

Eigenschaften

Der Watchdog Timer wird von einem separatem On - Chip Oszillator angesteuert, welcher mit 1 MHz bei 5V Vcc arbeitet. Wenn man den Watchdog Timer vom Prescaler aus ansteuert können dann verschiedenen Watchdog Reset Zeitintervalle eingestellt werden (siehe Anhang). Das WDR (Watchdog Reset) Befehl startet den Watchdog Timer. Es können 8 verschiedene periodische Taktzyklen ausgewählt werden um die Reset Periode zu bestimmen. Der ATmega16 wird von dem Reset Vector aus gestartet wenn die Reset Periode abgelaufen und wenn kein anderer Reset vorgefallen ist.

Watchdog Timer



Arbeitsweise

Nachdem der Watchdog aktiviert und der gewünschte Vorteile eingestellt wurde beginnt der Counter von 0 an hoch zuzählen. Wenn nun die je nach Vorteiler eingestellte Anzahl Zyklen erreicht wurde, löst der Watchdog einen Reset aus. Um nun also im Normalbetrieb den Reset zu verhindern muss der Watchdog regelmäßig wieder neu starten bzw. Rücksetzen (Watchdog Reset). Dies sollte innerhalb der Hauptschleife passieren.

Das WDE - Bit (Watchdog- Enable) aktiviert oder deaktiviert den Watchdog Timer. Falls es gesetzt ist wird der Watchdog Timer aktiv. Andernfalls (logisch 0) ist der Watchdog Timer inaktiv. Um ein unbeabsichtigtes Abschalten des Watchdogs zu verhindern, kann das WDE – Bit nur zurückgesetzt werden , wenn das WDTOE – Bit (Watchdog Turn Off Enable) gesetzt ist.

Um den Watchdog abzuschalten muss also folgende Prozedur durchgeführt werden :

- Mit dem gleichen Befehl müssen die Bits WDE und WDTOE gesetzt werden. Obwohl WDE zwangsläufig bereits gesetzt ist wenn der Timer aktiv ist, muss es ebenfalls gesetzt werden.
- Innerhalb der nächsten 4 Taktimpulse (das ist nötig da nachdem das WDTOE – Bit gesetzt wurde, bleibt es für 4 Perioden des Systemtaktes auf logisch 1 und danach wird es hardwaremäßig auf logisch 0 zurückgesetzt. Nur während dieser Zeit ist es möglich für das Anwenderprogramm den Watchdog abzuschalten) des Systemtaktes wird eine logische 0 in das Bit WDE eingeschrieben. Damit ist der Watchdog abgeschaltet.

Die nachfolgenden Codebeispiele zeigt einen Assembler Code und ein C Code wie man den Watchdog Timer ausschaltet. Dieses Beispiel geht davon aus, dass keine Interrupts während der Ausführung auftreten.

```
; Assembly Code Example
```

```
WDT_off:
    ; Reset WDT
    WDR

    ; Write logical one to WDTOE and WDE
    in r16, WDTCR
    ori r16, (1<<WDTOE) | (1<<WDE)
    out WDTCR,r16

    ; Turn off WDT
    ldi r16, (0<<WDE)
    out WDTCR, r16
    ret
```

```
// C Code Example
```

```
void WDT_off(void)
{
    /* Reset WDT */
    _WDR();

    /* Write logical one to WDTOE and WDE */
    WDTCR |= (1<<WDTOE) | (1<<WDE);

    /* Turn off WDT */
    WDTCR = 0x00;
}
```

Power Management und Sleep Modi

Einführung

In den vielen Bereichen wie Multimedia, Multimedia-, Kommunikations-, Automobil- oder Industriemarkt werden bekanntlich Controller eingesetzt. Es ist aber auch bekannt, dass in diesen Bereichen es auch sehr wichtig ist einen Controller zu haben, der wenig Strom verbraucht. Angesichts des Wunsches nach Reduzierung des Stromverbrauches gibt es beim Atmel Controller 6 verschiedene Sleep Modi, die es ermöglichen verschiedene Komponenten der MCU, die nicht gebraucht werden, abzuschalten. Demzufolge wird der Stromverbrauch erheblich reduziert.

Arbeitsweise

Um einen der 6 Sleep Modi zu aktivieren muss das SE Bit im MCUCR Register mit einer logischen 1 beschrieben werden und zudem soll noch ein Sleep Befehl ausgeführt werden. Die SM2, SM1 und SM0 Bits im MCUCR Register wählen zwischen verschiedenen Sleep Modi (Idle, ADC Noise Reduction, Power-down, Power – save, Standby, Extended Standby) aus, welche mit dem Sleep Befehl aktiviert werden. Wenn sich die MCU in einem Sleep Modus befindet und ein gesetzter Interrupt erkannt wird, dann wird die MCU geweckt. Die MCU wird dann zuzüglich zu der Start up Time 4 weitere Zyklen verweilen. Erst wird eine Interrupt Routine ausgeführt und dann wird das Programm nach dem Sleep Befehl weitermachen. Wenn das Gerät vom Sleep aufgeweckt worden ist, ist der Inhalt vom Register File und SRAM nicht verändert worden. Sobald ein Reset während dem Sleep Modus auftritt wird dann die MCU aufgeweckt und führt die Abarbeitung vom Reset Vector weiter aus. Das MCU Control Register – MCUCR ist im Anhang beschrieben.

Der Idle Modus

Wenn die SM2..0 Bist mit „000“ beschrieben werden, dann wird beim Sleep Befehl die MCU in den Sleep Modus versetzt. Dabei wird die CPU angehalten. Die SPI, USART, Analog Komparator, ADC, Two wire Serial Interface, Timer/Counters, Watchdog und das Interrupt System sind davon nicht betroffen und dürfen ihre Abarbeitung weiter fortführen. Dieser Sleep Modus hält im Grunde genommen die clk_{CPU} und clk_{FLASH} während andere Takte weiter machen dürfen.

Der Idle Modus ermöglicht das Aufwecken der MCU sowohl von einem extern getriggerten Interrupt als auch von einem internen wie dem Timer Overflow Interrupt und USART Transmit Complete Interrupt. Wenn das Aufwecken anhand des Analog Komparators nicht gebraucht wird, dann kann der Analogkomparator abgeschaltet werden indem man das ACD Bit im Analog Komparator und Statusregister – ACSR setzt. Damit wird dann eine weitere Reduktion des Stromverbrauches im Idle Modus erzielt. Eine Umwandlung wird automatisch gestartet wenn das ADC gesetzt ist.

Der ADC Noise Reduction – Modus

Wenn die SM2..0 Bits mit einer „001“ beschrieben werden, dann wird beim Sleep Befehl die MCU in den ADC Noise Reduction Modus versetzt. Dabei wird die CPU angehalten. Die external Interrupts, ADC, Two – wire Serial Interface address watch, Timer/Counter2 und dem Watchdog sind davon nicht betroffen und dürfen ihre Abarbeitung weiter fortführen (wenn eingeschaltet). Dieser Sleep Modus hält im Grunde genommen die clk_{IO} , clk_{CPU} und clk_{FLASH} während andere Takte weitermachen dürfen.

Das verbessert die störrische Umgebung für den ADC um bessere Messwerte zu erzielen. Eine Umwandlung startet automatisch, wenn der ADC aktiviert ist und in diesem Modus beigetreten wird. Abgesehen vom ADC Conversion Complete Interrupt kann nur ein externer Reset, ein Watchdog Reset, ein Brown – out Reset, ein Two – wire Serial Interface Address Match Interrupt, ein Timer/Counter2 interrupt, ein SPM/EEPROM ready Interrupt, ein External level Interrupt am INT0 oder INT1, oder ein externer Interrupt am INT2 die MCU vom ADC Noise Reduction Modus aufwecken.

Der Power – Down Modus

Wenn die SM2..0 Bits mit einer „010“ beschrieben werden, dann wird der Sleep Befehl die MCU in den Power – down Modus versetzen. In diesem Modus wird der externe Oszillator angehalten, während die externen Interrupts, der Two – wire Serial Interface address watch und der Watchdog weitermachen (falls eingeschaltet) dürfen. Nur ein externer Reset, ein Watchdog Reset, ein Brown – out Reset, ein Two – wire Serial Interface address match Interrupt, ein externer level Interrupt am INT0 oder INT1, oder ein externer Interrupt am INT2 hat die Möglichkeit die MCU aufzuwecken. Dieser Sleep Modus hält im Grunde genommen die erzeugten Takte während nur Befehle von den asynchronen Komponenten erlaubt sind.

Es ist darauf hinzuweisen, dass wenn ein level getriggert Interrupt für das Aufwecken vom Power – down Modus benutzt wird, dann muss das gewechselte Level für einige Zeit angehalten werden um die MCU aufzuwecken. Für genauere Informationen siehe Datenblatt.

Wenn der Controller vom Power – down Modus aufgeweckt wird, dann gibt es noch eine Zeitverzögerung bis der Controller anfängt zu arbeiten. Dieses Zeitintervall ermöglicht den Takt neu zu starten und ermöglicht des weiteren für eine stabile Arbeitsweise nach dem Anhalten. Der Zeitintervall (wake - up period) ist in den gleichen CKSEL Fuses welches die Reset time – out Periode definieren.

Der Power – save Modus

Wenn die SM2..0 Bits mit einer 011 beschrieben werden, dann wird beim Sleep Befehl die MCU in den Power – save Modus versetzt. Dieser Modus ist gleich mit dem Power – down Modus außer einer Ausnahme.

Wenn Timer/Counter2 asynchron getaktet wird, z.B. das AS2 Bit im ASSR Register ist gesetzt, dann wird der Timer/Counter2 während seinem Schlafzustand weitermachen. Der Controller kann dann entweder vom Timer overflow oder von einem Output Compare Event des Timer/Counter2 aufgeweckt werden.

Wenn der asynchrone Timer nicht asynchron getaktet ist, dann wird der Power – down Modus anstelle vom Power – save Modus empfohlen.

Dieser Sleep Modus hält im Grunde genommen alle Takte auf außer clk_{ASY} . Der clk_{ASY} erlaubt nur Ausführungen der asynchronen Komponenten.

Der Standby Modus

Wenn die SM2..0 Bits „110“ sind und ein externer Quarz/Resonator clock ausgewählt ist, dann wird beim Sleep Befehl die MCU in den Standby Modus versetzt. Dieser Modus ist gleich mit dem Power – down Modus mit der Ausnahme dass der Oszillator nicht angehalten wird. Vom Standby Modus aus wird der Controller in 6 Taktzyklen aufgeweckt.

Der Extended Standby Modus

Wenn die SM2..0 Bits „111“ sind und eine externer Quarz/Resonator Takt Option ausgewählt ist, dann wird beim Sleep Befehl die MCU in den Extended Standby Modus versetzt. Dieser Modus ist gleich mit dem Power – save Modus mit der Ausnahme dass der Oszillator weiterläuft. Vom Extended Standby Modus aus wird der Controller in 6 Taktzyklen aufgeweckt.

Active Clock Domains and Wake Up Sources in the Different Sleep Modes

Sleep Mode	Active Clock domains					Oscillators		Wake-up Sources					
	clk_{CPU}	clk_{FLASH}	clk_{IO}	clk_{ADC}	clk_{ASY}	Main Clock Source Enabled	Timer Osc. Enabled	INT2 INT1 INT0	TWI Address Match	Timer 2	SPM / EEPROM Ready	ADC	Other I/O
Idle			X	X	X	X	X ⁽²⁾	X	X	X	X	X	X
ADC Noise Reduction				X	X	X	X ⁽²⁾	X ⁽³⁾	X	X	X	X	
Power Down								X ⁽³⁾	X				
Power Save					X ⁽²⁾		X ⁽²⁾	X ⁽³⁾	X	X ⁽²⁾			
Standby ⁽¹⁾						X		X ⁽³⁾	X				
Extended Standby ⁽¹⁾					X ⁽²⁾	X	X ⁽²⁾	X ⁽³⁾	X	X ⁽²⁾			

- Notes:
1. External Crystal or resonator selected as clock source.
 2. If AS2 bit in ASSR is set.
 3. Only INT2 or level interrupt INT1 and INT0.

4. Anhang

Fuse Bits

Der ATmega 16 hat 2 fuse Bytes. Dabei handelt es sich um Fuse High Byte und Fuse Low Byte. Die nächsten 2 Tabelle beschreiben die Funktionalitäten der Bytes und wie sie zugeordnet sind. Wenn die Fuse Bits als 0 gelesen werden dann bedeutet das, das die Bits programmiert sind.

Fuse High Byte

Fuse High Byte	Bit No.	Description	Default Value
OCDEN ⁽⁴⁾	7	Enable OCD	1 (unprogrammed, OCD disabled)
JTAGEN ⁽⁵⁾	6	Enable JTAG	0 (programmed, JTAG enable)
SPIEN ⁽¹⁾	5	Enable SPI Serial Programm And Data Downloading	0 (programmed, SPI prog. Enabled)
CKOPT ⁽²⁾	4	Oscillator options	1 (unprogrammed)
EESAVE	3	EEPROM memory is preserved through the Chip Erase	1 (unprogrammed , EEPROM not preserved)
BOOTSZ1	2	Select Boot Size	0 (programmed) ⁽³⁾
BOOTSZ0	1	Select Boot Size	0 (programmed) ⁽³⁾
BOOTRST	0	Select reset vector	1 (unprogrammed)

1. The SPIEN Fuse is not accesible in SPI Serial Programming mode.
2. The CKOPT Fuse functionally depends on the setting of the CKSEL bits.
3. The default value of BOOTSZ1..0 results in maximum Boot Size.
4. Never ship a product with the OCDEN Fuse programmed regardless of the setting of Lock bits and the JTAGEN Fuse. A programmed OCDEN Fuse enables some parts of the clock system to be running in all sleep modes. This may increase the power consumption
5. If the JTAG interface is left unconnected, the JTAGEN fuse should if possible be discabled. This to avoid static current at the TDO pin in the JTAG interface.

Fuse Low Byte

Fuse Low Byte	Bit No.	Description	Default Value
BODLEVEL	7	Brown-out Detector trigger level	1 (unprogrammed)
BODEN	6	Brown-out Detector enable	1 (unprogrammed, BOD disabled)
SUT1	5	Select start-up time	1 (unprogrammed) ⁽¹⁾
SUT0	4	Select start-up time	0 (programmed) ⁽¹⁾
CKSEL3	3	Select Clock source	0 (programmed) ⁽²⁾
CKSEL2	2	Select Clock source	0 (programmed) ⁽²⁾
CKSEL1	1	Select Clock source	0 (programmed) ⁽²⁾
CKSEL0	0	Select Clock source	1 (unprogrammed) ⁽²⁾

1. The default value of SUT1..0 results in maximum start-up time.
2. The default setting of CKSEL3..0 results in internal RC Oscillator @ 1MHz.

Das Oszillator Kalibrierungsregister - OSCCAL

Bit	7	6	5	4	3	2	1	0	
	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	OSCCAL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	Device Specific Calibration Value								

Bit 7..0 – CAL7..0: Oszillator Kalibrierungswerte

Wenn man das Kalibrierungsbyte in dieses Register schreibt, dann wird der interne Oszillator Abweichungen vom Oszillator Frequenz anpassen. Das wird dann während dem Reset vom Chip aus gemacht. Wenn das OSCCAL Register den Wert 0 hat, dann wird die kleinste vorhandene Frequenz ausgewählt. Wenn man ein Wert ungleich 0 in das Register schreibt, dann wird die interne Frequenz des Oszillator erhöht. Man bekommt die höchste Frequenz, wenn man \$FF in das Register schreibt, Der Kalibrierungsozillator wird benutzt um den Zugriff auf EEPROM und Flash zu synchronisieren. Sobald der EEPROM oder Flash beschrieben wurde, sollte es vermieden werden mehr als 10% über der Nennfrequenz zu kalibrieren. Ansonsten wird der der Zugriff fehlschlagen. Der Oszillator ist gemacht für 1.0, 2.0, 4.0 oder 8.0 MHz.

Internal RC Oscillator Frequency Range

OSCCAL Value	Min Frequency in Percentage of Nominal Frequency (%)	Max Frequency in Percentage of Nominal Frequency (%)
\$00	50	100
\$7F	75	150
\$FF	100	200

Das MCU Control Register - MCUCR:

Dieses Register enthält die Steuerbits für bestimmte MCU (Mikrocontroller Unit) Funktionen. Hier werden unter anderem verschiedene Sleep – Modi angesteuert. Das MCUCR liegt auf der Adresse 35_{Hex}.

Das MCU Control Register enthält Steuerung Bits für power management.

Bit	7	6	5	4	3	2	1	0	
	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7,5,4 – SM2..0: Sleep Mode Select Bits 2,1, and 0

Diese Bits wählen zwischen 6 vorhanden Sleep Modi aus. Siehe Tabelle

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	Reserverd
1	1	0	Standby ⁽¹⁾
1	1	1	Extended Standby ⁽¹⁾

(1) : Standby Modus und Extended Standby Modus sind nur im Zusammenhang mit externem Quarz oder Resonator auswählbar.

Bit 6 – SE: Sleep Enable

Der Sleep Befehl wird wirksam gemacht, wenn dieses Bit auf 1 gesetzt ist.

Bit 3,2 – ISC11, ISC10: Interrupt Sense Control 1 Bit 1 and Bit 0

Der externe Interrupt 1 wird von einem externen Pin INT1 aktiviert, sobald das SREG I-Bit und die entsprechende Interrupt mask im GICR gesetzt sind. Die Stufen und Flanken am externen INT1 Pin, welche den Interrupt aktivieren sind unten in der Tabelle aufgeführt. Der Wert am INT1 Pin wird abgetastet bevor Flanken erkannt werden. Wenn das Flanken oder toggel Interrupt konfiguriert ist, wird der letzte Impuls, der länger als eine Takt Periode ist, einen Interrupt erzeugen. Bei kürzeren Impulsen wird ein Interrupt nicht garantiert. Wenn das low level Interrupts konfiguriert ist, muss das low level gehalten werden, bis die aktuelle Befehlsausführung beendet ist um einen Interrupt zu generieren.

ISC11	ISC10	Description
0	0	The Low level of INT1 generates an interrupt request
0	1	Any logical change on INT1 generates an interrupt request
1	0	The falling edge of INT1 generates an interrupt request
1	1	The rising edge of INT1 generates an interrupt request

Bit 1,0 – ISC01, ISC00 : Interrupt Sense Control 0 Bit 1 and Bit 0

Der externe Interrupt 0 wird von einem externen Pin INTO aktiviert, wenn das SREG I - flag und das zugehörige Interrupt mask gesetzt sind. Die Stufen und Flanken am externen INTO Pin welche den Interrupt aktivieren sind unten in der Tabelle aufgeführt. Der Wert am INTO Pin wird abgetastet bevor Flanken erkannt werden. Wenn das Flanken oder toggeln Interrupt konfiguriert ist werden letzte Impulse, die länger als eine Takt Periode sind, einen Interrupt erzeugen. Für kürzere Impulse werden keine Interrupts garantiert. Wenn das low level Interrupt konfiguriert ist, muss das low level gehalten werden bis die aktuelle Befehlsausführung beendet ist um einen Interrupt zu generieren.

ISC01	ISC00	Description
0	0	The low level of INTO generates an interrupt request
0	1	Any logical change on INTO generates an interrupt request
1	0	The falling edge INTO generates an interrupt request.
1	1	The rising edge of INTO generates an interrupt request.

Das Watchdog Timer Control Register – WDTCR:

Dieses Register enthält die Steuerbits für den Watchdog Timer. Hier wird der Watchdog Timer de - / aktiviert, Watchdog Timer Prescaler gesetzt Das WDTCR liegt auf der Adresse 21_{Hex} .

Bit	7	6	5	4	3	2	1	0	
	-	-	-	WDTOE	WDE	WDP2	WDP1	WDP0	WDTCR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7..5 – Res: Reserved Bits

Diese Bits hier sind bei ATmega16 reserviert und werden immer als null gelesen.

Bit 4 – WDTOE: Watchdog Turn - off Enable

Wenn das Bit WDE mit einer logischen 0 beschrieben werden will, muss dieses Bit gesetzt sein. Ansonsten wird der Watchdog nicht deaktiviert. Nachdem es einmal mit einer logischen 1 geschrieben wurde, wird es die Hardware dann wieder nach 4 Taktzyklen dieses Bit löschen.

Bit 3 –WDE: Watchdog Enable

Der Watchdog Timer wird aktiviert, sobald man das WDE Bit auf eine logische 1 setzt. Und wenn das WDE mit einer logischen 0 beschrieben wird, dann ist die Watchdog Timer Funktion deaktiviert. WDE kann nur dann gelöscht werden, wenn das WDTOE Bit eine logische 1 hat. Die genaue Prozedur, wie der Watchdog Timer ausgeschaltet wird, ist weiter oben beschrieben.

Bit 2..0 – WDP2, WDP1, WDP0: Watchdog Timer Prescaler 2,1, and 0

Wenn der Watchdog Timer aktiviert ist bestimmen die WDP2, WDP1 und WDP0 Bits den Watchdog Timer Prescaler. Die verschiedenen Prescaler Werte und die dazugehörigen Timeout Perioden sind in der folgenden Tabelle aufgeführt.

WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time – out At Vcc = 3.0 V	Typical Timer – out At Vcc = 5.0 V
0	0	0	16 K (16,384)	17,1 ms	16.3 ms
0	0	1	32 K (32,768)	34,3 ms	32.5 ms
0	1	0	64 K (65,536)	68,5 ms	65 ms
0	1	1	128 K (131,072)	0.14 s	0.13 ms
1	0	0	256 K (262,144)	0.27 s	0.26 s
1	0	1	512 K (524,288)	0.55 s	0.52 s
1	1	0	1,024 K (1,048,576)	1.1 s	1.0 s
1	1	1	2,048 K (2,097,152)	2.2 s	2.1 s

5. Quellen

Datenblatt ATmega16
C für Mikrocontroller von Burkhard Mann

www.avrbeginners.net
www.mikrocontroller.net
www.avr-asm-tutorial.net