SOFTWARE PWM CONTROLLER ATMEGA 8
TYP: PWM18_V3

INHALTSVERZEICHNIS

Features

- ***TWI- Interface, Auto Increment***
- ***Watchdog Überwachung***
- ***Insgesamt 18- Frei wählbare PWM- Kanäle***

16x Kanal (8-Bit) Software PWM, Variable Auflösung 2-8 Bit
@ 8MHz Internal RC and 8- Bit Resolution = 128- 250Hz PWM- Frequency
@ 16 MHz Crystal Clock and 8- Bit Resolution = 250- 500Hz PWM- Frequency
2x Kanal 16- Bit Hardware PWM(Fast PWM), Variable Auflösung 2-16

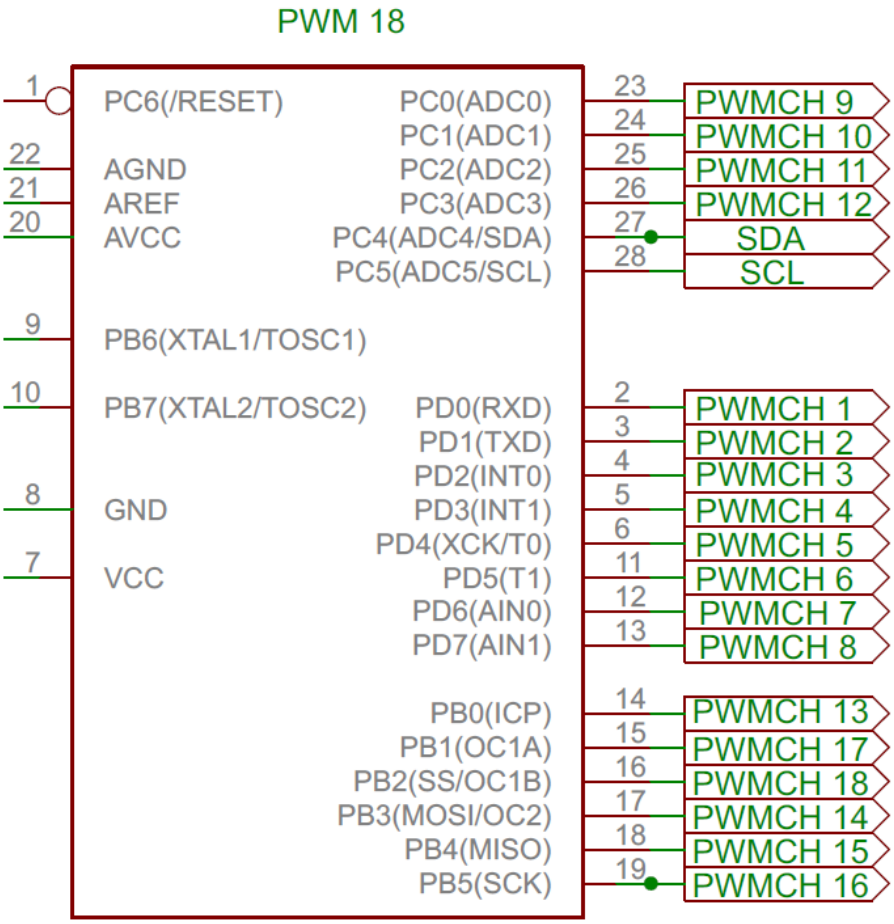
- ***186 Byte SRAM für eigene Datenspeicherung***
- ***Über TWI- Bus zugängliche und veränderbare Hardware Einstellungen***

─ TWI- Slave Adresse, *Standart SLA= 0x7A*
─ Clock Select Bytes Timer 1, 2 (PWM- Frequency)
─ Auflösung 16- Bit PWM
─ Auflösung 8-Bit Software PWM
─ Interruptzeit für Software PWM(!)
─ *Alle Hardware Einstellungen lassen sich im EEPROM*
über einen einzigen Befehl abspeichern.
─ *Alle PWM- Kanalwerte lassen sich im EEPROM*
über einen einzigen Befehl abspeichern.
─ „*Preload- Funktion*“ für gespeicherte PWM- Kanalwerte

Anwendungsgebiete:
LED- Helligkeitsregelung, Modellbau, Leistungsregelung.

PWM- Controller Pinout

AVR ATMEL ATMEGA 8
8-Bit µController



Pin	Function
1	Reset
2-6, 11-19, 23-26	PWM- Outputs
22, 8	0V- GND
21, 20, 7	VCC(3-5V)
9	Clock In
10	Clock Out
9,10	Crystal, or Ceramic Resonator
27	SDA- Serial Data, TWI or IIC- Bus
28	SCL- Serial Clock, TWI or IIC- Bus

Speicherbelegung 1/2

Funktion(Register)	*Bereichs-Unterteilung*	
—	—	
PWM Kanal 1	**Software PWM Channels**	**PWM-Values Space**
PWM Kanal 2		
PWM Kanal 3		
PWM Kanal 4		
PWM Kanal 5		
PWM Kanal 6		
PWM Kanal 7		
PWM Kanal 8		
PWM Kanal 9		
PWM Kanal 10		
PWM Kanal 11		
PWM Kanal 12		
PWM Kanal 13		
PWM Kanal 14		
PWM Kanal 15		
PWM Kanal 16		
PWM Kanal 17 HB	**Hardware PWM Channels**	
PWM Kanal 17 LB		
PWM Kanal 18 HB		
PWM Kanal 18 LB		
Control_Reg	**Hardware Settings space**	
Temporary2		
CS_Soft_PWM		
CS_HW_PWM_16		
Res_HW_PWM_HB		
Res_HW_PWM_LB		
Res_Soft_PWM		
PWM_vs_BUS		

In dieser Tabelle kann man sehen welche Register Zu welchen Gruppen gehören.
Dies ist von Bedeutung wenn die Befehle 0x3A, 0x3B ausgeführt werden.

Der Befehl **HW_ Settings Save**(0x3A) ist für die Speicherung von allen Registern aus dem Bereich Hardware Settings space zuständig.

Der Befehl **PWM_ Values Save** (0x3B) ist für die Speicherung von allen Registern aus dem Bereich PWM- Values Space zuständig. Es werden die Werte für die Kanäle 1-18 auf einmal gespeichert.

Die Register können auch einzeln abgespeichert werden. Mehr Informationen auf Seite TWI- Registerfunktionen 2/2.

Die **Hardware Settings** inklusive Slaveadresse werden im Einschaltmoment (Power ON) immer aus den EEPROM geladen!

Die **PWM- Values** können wahlweise, entweder mit festwerten aus dem Flash, oder den gespeicherten EEPROM- Werten geladen(Preload) werden.

TWI- Registerfunktionen 1/2

TWI - Register [HEX]	*Funktion*	*R/W**	*Notes*
0	----	----	none
1	PWM Kanal 1	R/W	PWM Kanal - Wert
2	PWM Kanal 2	R/W	PWM Kanal - Wert
3	PWM Kanal 3	R/W	PWM Kanal - Wert
4	PWM Kanal 4	R/W	PWM Kanal - Wert
5	PWM Kanal 5	R/W	PWM Kanal - Wert
6	PWM Kanal 6	R/W	PWM Kanal - Wert
7	PWM Kanal 7	R/W	PWM Kanal - Wert
8	PWM Kanal 8	R/W	PWM Kanal - Wert
9	PWM Kanal 9	R/W	PWM Kanal - Wert
0A	PWM Kanal 10	R/W	PWM Kanal - Wert
0B	PWM Kanal 11	R/W	PWM Kanal - Wert
0C	PWM Kanal 12	R/W	PWM Kanal - Wert
0D	PWM Kanal 13	R/W	PWM Kanal - Wert
0E	PWM Kanal 14	R/W	PWM Kanal - Wert
0F	PWM Kanal 15	R/W	PWM Kanal - Wert
10	PWM Kanal 16	R/W	PWM Kanal - Wert
11	PWM Kanal 17 HB	R/W	PWM Kanal High_Byte - Wert
12	PWM Kanal 17 LB	R/W	PWM Kanal Low_Byte - Wert
13	PWM Kanal 18 HB	R/W	PWM Kanal High_Byte - Wert
14	PWM Kanal 18 LB	R/W	PWM Kanal Low_Byte - Wert
15	**Control_Reg**	R/W	**Steuerregister**
16	Temporary2	R/W	Temporäres Speicherregister
17	CS_Soft_PWM	R/W	Clock Select für Soft- PWM Zeitbasis
18	CS_HW_PWM_16	R/W	Frequenz für 16-Bit PWM(Kanal 17,18)
19	Res_HW_PWM_HB	R/W	Auflösung HB 16-Bit PWM (Kanal 17,18)
1A	Res_HW_PWM_LB	R/W	Auflösung LB 16-Bit PWM (Kanal 17,18)
1B	Res_Soft_PWM	R/W	Auflösung für Soft PWM
1C	PWM_vs_BUS	R/W	Soft PWM- Frequenz vs BUS Speed
1D	SREG_3	R	Statusinformationen
FB	TWI_ADDR(IIC)	R/W	Ein Beispiel zum ändern der Slaveadresse
FC	SLA+W Save	W	gibt's auf der Seite „TWI_ADDR".
FD	TWI- ID High_Byte	R	0x0B
FE	TWI- ID Low_Byte	R	0x0A

R/W*- Read /Write, können gelesen und geschrieben werden.
R- nur Leseregister sind identisch zu den R/W- Registern, nur dass in die Schreibrichtung keine Parameter ausgewertet werden können.
W- nur Schreibregister, können zwar Parameter annehmen aber nichts verwertbares beim auslesen abgeben.

TWI- Registerfunktionen 2/2

TWI - Register [HEX]	*Funktion*	*R/W**	*Notes*
1E	PWM Kanal 1 Save	W	Aktuellen Kanal Wert abspeichern
1F	PWM Kanal 2 Save	W	Aktuellen Kanal Wert abspeichern
20	PWM Kanal 3 Save	W	Aktuellen Kanal Wert abspeichern
21	PWM Kanal 4 Save	W	Aktuellen Kanal Wert abspeichern
22	PWM Kanal 5 Save	W	Aktuellen Kanal Wert abspeichern
23	PWM Kanal 6 Save	W	Aktuellen Kanal Wert abspeichern
24	PWM Kanal 7 Save	W	Aktuellen Kanal Wert abspeichern
25	PWM Kanal 8 Save	W	Aktuellen Kanal Wert abspeichern
26	PWM Kanal 9 Save	W	Aktuellen Kanal Wert abspeichern
27	PWM Kanal 10 Save	W	Aktuellen Kanal Wert abspeichern
28	PWM Kanal 11 Save	W	Aktuellen Kanal Wert abspeichern
29	PWM Kanal 12 Save	W	Aktuellen Kanal Wert abspeichern
2A	PWM Kanal 13 Save	W	Aktuellen Kanal Wert abspeichern
2B	PWM Kanal 14 Save	W	Aktuellen Kanal Wert abspeichern
2C	PWM Kanal 15 Save	W	Aktuellen Kanal Wert abspeichern
2D	PWM Kanal 16 Save	W	Aktuellen Kanal Wert abspeichern
2E	PWM Kanal 17 HB Save	W	Aktuellen Kanal Wert abspeichern
2F	PWM Kanal 17 LB Save	W	Aktuellen Kanal Wert abspeichern
30	PWM Kanal 18 HB Save	W	Aktuellen Kanal Wert abspeichern
31	PWM Kanal 18 LB Save	W	Aktuellen Kanal Wert abspeichern
32	**Control_Reg Save**	W	**Steueregister Zustand speichern**
33	Temporary2 Save	W	Temporäres Speicherregister sichern
34	CS_Soft_PWM Save	W	Soft PWM- Timebase ClockSelect save
35	CS_HW_PWM_16 Save	W	PWM CH 17,18 ClockSelect save
36	Res_HW_PWM_HB Save	W	Resolution High_Byte save
37	Res_HW_PWM_LB Save	W	Resolutin Low_Byte save
38	Res_Soft_PWM Save	W	Software PWM Resolution save
39	PWM_vs_BUS Save	W	PWM vs TWI- BUS speed save
3A	HW_ Settings Save	W	Betroffene Register 0x15- 0x1C Hardware Settings space
3B	PWM_ Values Save	W	PWM Kanäle 1-18 auf einmal abspeichern
3C- F5	SRAM	R/W	SRAM Speicher 186 Byte

R/W*- Read /Write, können gelesen und geschrieben werden.
R- nur Leseregister sind identisch zu den R/W- Registern, nur dass in die Schreibrichtung keine Parameter ausgewertet werden können.
W- nur Schreibregister, können zwar Parameter annehmen aber nichts verwertbares beim auslesen abgeben.

Die Register 0x1E- 0x3B müssen mit groß ‚C'(0x43) beschrieben werden um die jeweilige Aktion auszuführen. Zugriffsbeispiele siehe Seite „Anwendungs- Beispiele X/X" .

Slave Receiver Modus

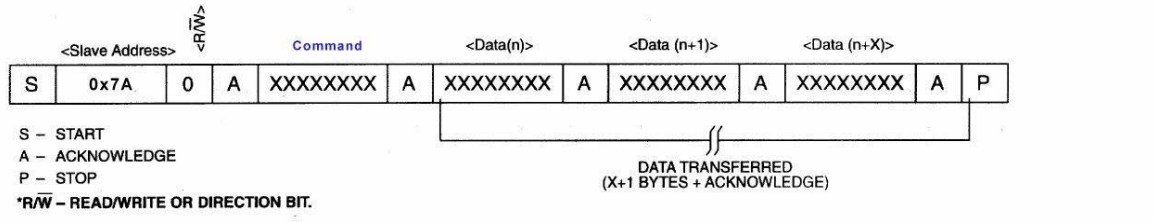
Es lassen sich je Schreibzugriff maximal 255 Byte empfangen.
Empfangsbuffer = 255 Byte.

1) Schreiben
2) Lesen

1) Schreiben

Ein einfacher Schreibzugriff besteht aus **mindestens 2 Byte**.
Byte1: Ist das Adressbyte siehe Seite „TWI- Registerfunktionen“
Byte2: Datenbyte, je nach Register, notfalls ein Dummybyte,
aber immer mindestens zwei Byte.
Byte3 +(n) wird durch Autoinkrement ins nächste Register hineinkopiert.

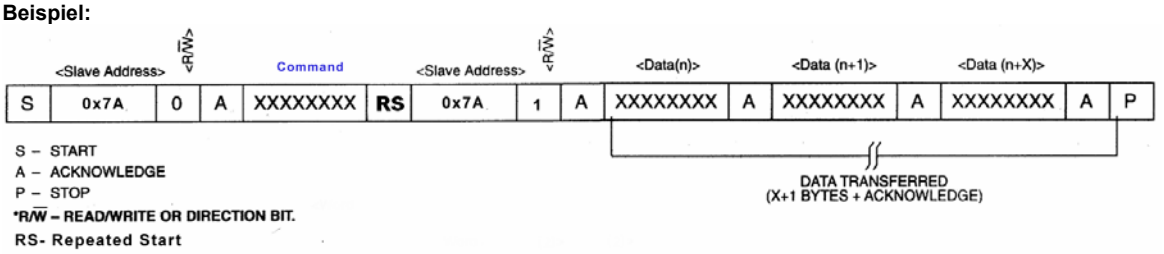
Beispiel:



2) Lesen

Ein Lesezugriff besteht aus **nur 1 Byte**.
Byte1 (Pointer Adresse): Ist das Adressbyte siehe Seite Register Adressen „TWI-Registerfunktionen“

Beispiel:



Control_Reg

Beschreibung:
Dieses Register ist für das Verhalten nach einem Reset(Einschaltphase) zuständig.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	PWUPL
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 –
- Bit 6 –
- Bit 5 –
- Bit 4 –
- Bit 3 –
- Bit 2 –
- Bit 1 –

- Bit 0 – **PWUPL: Power Up load saved PWM- Values**

Bei Logisch „1“ werden die PWM- Werte für Kanal 1-18 aus dem EEPROM geladen. Dies ist sinnvoll wenn die PWM- Kanäle nach dem einschalten einen Wert abweichend von 0x00- besitzen sollen.
Bei Logical =0 werden alle oben aufgelistete PWM- Kanäle by default mit 0 geladen.

SREG_3- Statusregister

Beschreibung:
Dieses Register beinhaltet den Status über die EEPROM- Speicherbefehle.
Dessen Status wird nur dann verändert, wenn ein EEPROM– Speicherzugriff erfolgte.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	SBWSL	HSWSL	PVWSL
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 –
- Bit 6 –
- Bit 5 –
- Bit 4 –
- Bit 3 –

- Bit 2 – **SBWSL: Single Byte write successful**

Die Speicherregister 0x1E- 0x39 und 0xFB greifen auf eine und dieselbe EEPROM Speicherroutine zu. Die Daten werden zuerst in den EEPROM geschrieben danach ausgelesen und verglichen. Sollten die Daten nicht übereinstimmen, so wird diese Prozedur bis zu dreimal nacheinander wiederholt und bei bestehendem Misserfolg wird das Flag „**SBWSL**“ gelöscht und umgekehrt.

Weitere Details lassen sich auch dem PAP „EEP_Byte_Save“ entziehen.

- Bit 1 – **HSWSL: Hardware Settings write successful**

Ist high wenn der Befehl „HW_ Settings Save“ (0x3A) erfolgreich ausgeführt wurde.

- Bit 0 – **PVWSL: PWM- Values write successful**

Ist high wenn der Befehl „PWM_ Values Save“(0x3B) erfolgreich ausgeführt wurde.

TWI_ADDR (Slaveadresse)

Beschreibung:
Das Register TWI_ADDR- beinhaltet die TWI- Busadresse. Die Busadresse befindet sich im EEPROM und wird nach anlegen der Versorgungsspannung von da aus geladen.

Bit	7	6	5	4	3	2	1	0
	—	TWI_ADDR						
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 6..0 – **TWA: TWI (Slave) Address Register**
Slaveadresse 0- 127

● Bit 7 –

Aufgabe:
Slave Adresse ändern.

Lösung:

	Slave Write	CMD	SLA neu	Confirm	
Start	**SLA+W**	**0xFB**	**0xXX**	**0x43**	**STOP**

Die neue Slaveadresse wird ins Register 0xFB und in den EEPROM geschrieben.
Der Zugriff hat nach dem Beispiel oben zu erfolgen. Sollte das Confirm Zeichen(0x43) nicht übereinstimmen, so wird die alte Adresse beibehalten und die neue nicht berücksichtigt.

NOTES:
Bus Adresse wiederherstellen:
Die default Adresse $7A kann durch erneutes Flashen des E²PROM's wiederhergestellt werden, wobei alle andere Hardware Settings flöten gehen :-p

PWM_vs_BUS

Beschreibung:
Default Wert 0xFF.
Wird dessen Wert reduziert so beschleunigt sich die Soft- PWM um einige Hz.
Wurde ein zu geringer Wert ausgewählt, so hat der AVR viel zu wenig Zeit um sich um den TWI- Driver zu kümmern!
Der kleinste Wert von „PWM_vs_BUS“ darf keinesfalls geringer als die längste Interruptzeit sein.
Interruptzeit Max. clock cycles = **86**
Interruptzeit Min. clock cycles = **47**

Bit	7	6	5	4	3	2	1	0
	PWM_vs_BUS							
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

Dieses Register ist Softwaremäßig gegen falsche Eingaben gesichert.
Werte die die Grenzen überschreiten werden Ignoriert.
Mindestwert: 90(Dez) oder 0x5A(HEX)
Maxwert: nicht definiert.

Für wem die default Frequenz nicht ausreichend ist!
Getestet wurde mit einem Wert von **128**. Die Firmware läuft weiterhin Fehlerfrei.
Die PWM- Frequenz ist dabei ums doppelte gestiegen wobei es sich immer noch um 8- Bit Soft- PWM handelt.

CS_Soft_PWM (Frequenz Kanal 1-16)

Beschreibung:
Default Wert 0x01
Software PWM- Clock Select Bits.
Werte die größer als 1 sind führen zur Verringerung der Software PWM Frequenz.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	CS22	CS21	CS20
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 –
- Bit 6 –
- Bit 5 –
- Bit 4 –
- Bit 3 –
- Bit 2:0 – CS22:0: Clock Select

Table 46. Clock Select Bit Description

CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk_{T2S}/(No prescaling)
0	1	0	clk_{T2S}/8 (From prescaler)
0	1	1	clk_{T2S}/32 (From prescaler)
1	0	0	clk_{T2S}/64 (From prescaler)
1	0	1	clk_{T2S}/128 (From prescaler)
1	1	0	clk_{T2S}/256 (From prescaler)
1	1	1	clk_{T2S}/1024 (From prescaler)

Dieses Register ist Softwaremäßig gegen falsche Eingaben gesichert.
Werte die, die Grenzen überschreiten werden Ignoriert.

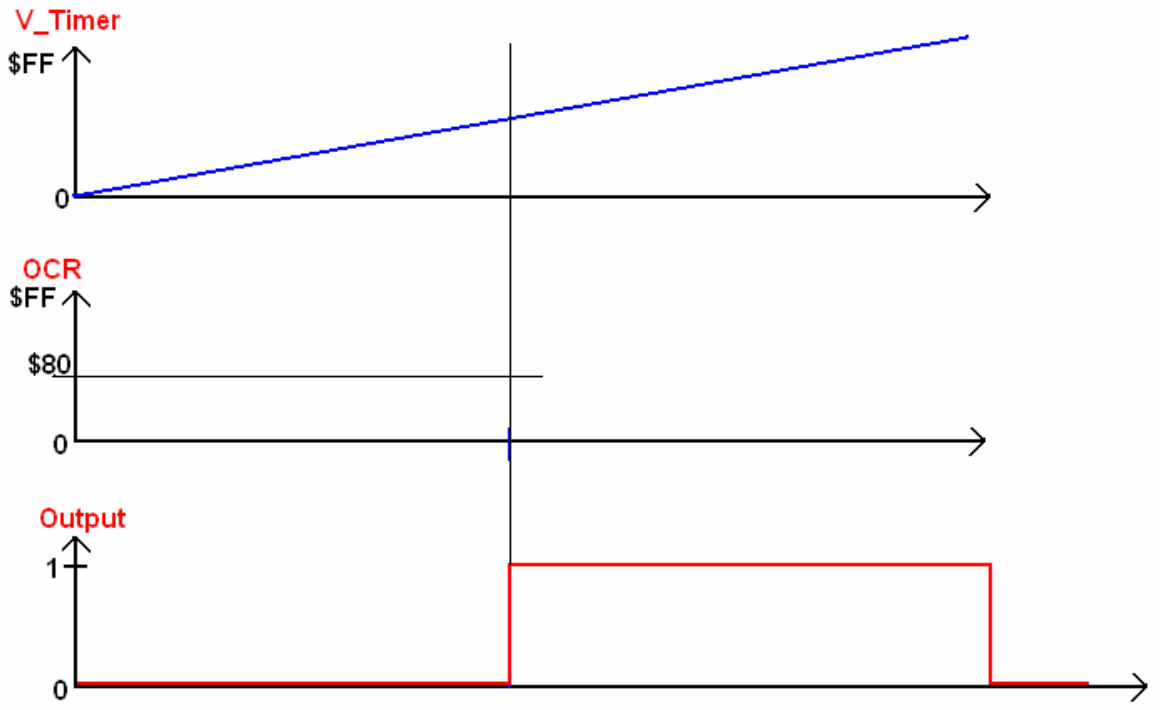
Mindestwert: nicht definiert.
Maxwert: 7(Dez) 0x07(HEX), CS22=1, CS21=1, CS20=1

Res_Soft_PWM (Auflösung Kanal 1- 16)

Beschreibung:
Auflösung für Software PWM- Kanäle 1-16.
Wird die Auflösung von 8 auf 7-Bit umgestellt, so steigt die Software PWM- Frequenz auf das doppelte.
Achtung! mit sinkender Auflösung in 1- Bit Schritten verdoppelt sich die PWM- Frequenz ums doppelte.

Bit	7	6	5	4	3	2	1	0
	Res_Soft_PWM							
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

Das Register Res_Soft_PWM bestimmt den Top- Wert des PWM_Counters(V_Timer). Dieser ist in seiner Funktion Identisch mit einem Hardware Timer in CTC- Modus.



CS_HW_PWM_16 (Frequenz Kanal 17,18)

Beschreibung:
Default Wert 0x01
Von dessen Wert hängt die Geschwindigkeit des PWM- Outputs OC1A(PWMCH17) und OC1B(PWMCH18) ab.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	CS12	CS11	CS10
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

Table 40. Clock Select Bit Description

CS12	CS11	CS10	Description
0	0	0	No clock source. (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/1 (No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

Dieses Register ist Softwaremäßig gegen falsche Eingaben gesichert.
Werte die, die Grenzen überschreiten werden Ignoriert.

Mindestwert: nicht definiert.
Maxwert: 7(Dez) 0x07(HEX), CS12=1, CS11=1, CS10=1

Res_HW_PWM_HB + Res_HW_PWM_LB (Auflösung Kanal 17,8)

Beschreibung:
Ist für die Auflösung von PWM- Output CH 17 und 18 zuständig.

Bit	7	6	5	4	3	2	1	0
	Res_HW_PWM_HB [15:8]							
	Res_HW_PWM_LB [7:0]							
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

Aufgabe:
Auflösung für Kanäle 17,18 temporär ändern.
Die Auflösung wird bis zum Reset gespeichert
Lösung:

	Slave Write	Register	Res HB	Res LB	
Start	**SLA+W**	**0x19**	**0xXX**	**0xXX**	**STOP**

Somit wurde die neue Auflösung übernommen und temporär gespeichert.
Will man die gesetzte Auflösung auch nach einem Reset nicht missen, so sollte anschließend ins Register **0x36, 0x37** oder 0x 3A groß ‚C' geschrieben werden.
Mehr Details auf Seite „Die Register".

Aufgabe:
Auflösung für Kanäle 17,18 im EEPROM speichern.
Die Auflösung wird nach einem Reset aus den EEPROM geladen
Lösung:

	Slave Write	CMD	Confirm	Confirm	
Start	**SLA+W**	**0x36**	**0x43**	**0x43**	**STOP**

Werkseinstellungen

EEPROM und Flash Programmieren sonst funktioniert
Die Software nicht richtig !

Nr	Register	Wert
1	Control Register	0x00
2	TWI(IIC)- Busadresse	0x7A
3	CS_Soft_PWM (schnellste Software PWM Frequenz)	0x01
4	CS_HW_PWM_16 (schnellste Hardware Frequenz)	0x01
5	Res_HW_PWM (Höchste Auflösung 16 Bit)	0xFFFF
6	Res_Soft_PWM (Höchste Auflösung 8 Bit)	0xFF
7	PWM_vs_BUS (Schnellste Busgeschwindigkeit)	0xFF
8	PWM- Kanalwerte 1-18	0x00

FUSE Settings:

1) RSTDISBL sollte Logisch „0“ bleiben.
2) WTDON, kann dauerhaft an sein.
3) Clock Select External Crystal >8MHz.

TWI (IIC) Ansteuerungs- Beispiele 1/2

Allgemeine Infos:

ASCII: C = HEX: 0x43 = DEC: 067

Aufgabe:

PWM Kanal 10 sollte einen neuen Wert erhalten. (Register **schreiben**)

Lösung:

1) Start
2) SLA+W
3) **Byte1: 0x0A** (Registeradresse)
4) **Byte2: 0x64** (neuer Wert).
5) Stoppkondition

Jetzt wurde der neue PWM- Wert für Kanal 10 temporär übernommen.

Aufgabe:

PWM Kanal 6 sollte im EEPROM **gespeichert** werden.

Lösung:

1) Start
2) SLA+W
3) **Byte1: 0x23** (Registeradresse)
4) **Byte2: ‚C'** (Großbuchstabe).
5) Stoppkondition

Jetzt wurde der neue PWM- Wert für Kanal 6 im EEPROM abgespeichert.

Aufgabe:

Mehrere Kanäle mit neuen Werten **überschreiben.**

PWM- Kanäle 8-11 sollte mit neuen Werten beschrieben werden.

Lösung:

1) Start
2) SLA+W
3) **Byte 1: 0x08** (Register Adresse)
4) **Byte 2: 0x32** (neuer Wert für Kanal 8).
5) **Byte 3: 0x16** (neuer Wert für Kanal 9).
6) **Byte 4: 0x80** (neuer Wert für Kanal 10).
7) **Byte 5: 0x30** (neuer Wert für Kanal 11).
8) **Stoppkondition**

u.s.w

Jetzt wurden die neuen PWM- Wert für die Kanäle 8-11 temporär übernommen.

Aufgabe:

Mehrere Kanäle **auslesen.**

PWM- Kanäle 8-11 sollte ausgelesen werden.

Lösung:

1) Start
2) SLA+W
3) **Byte 1: 0x08** (Register Adresse)
4) **Repeated Start**
5) SLA+R
6) **RX_Byte 1: 0xXX** (Wert von Kanal 8).
7) **RX_Byte 2: 0xXX** (Wert von Kanal 9).
8) **RX_Byte 3: 0xXX** (Wert von Kanal 10).
9) **RX_Byte 4: 0xXX** (Wert con Kanal 11).
10) **Stoppkondition**

u.s.w

Aufgabe:
Alle PWM- Kanäle 1-18 auf einmal ab**speichern**. (Registeradresse = PWM_ Values Save)
Lösung:

1) Start
2) SLA+W
3) **Byte1: 0x3B** (Registeradresse)
4) **Byte2: ‚C'** (Großbuchstabe).
5) Stoppkondition

Danach sind die neuen PWM- Werte für Kanal 1-18 im EEPROM gespeichert.

Aufgabe:
Alle Hardware Settings Register auf einmal **speichern**.(Registeradresse = HW_ Settings Save)
Lösung:

1) Start
2) SLA+W
3) **Byte 1: 0x3A** (Registeradresse)
4) **Byte 2: ‚C'** (Großbuchstabe).
5) Stoppkondition

Danach sind die neuen Hardware Settings im EEPROM gespeichert.

Aufgabe:
PWM Kanal 6, 8 und 9 sollten im EEPROM **gespeichert** werden.
Lösung:

1) Start
2) SLA+W
3) **Byte1: 0x23** (Speicher Registeradresse Kanal 6)
4) **Byte2: ‚C'** (Großbuchstabe). Speicherbestätigung für Kanal 6
5) **Byte3: ‚X'** (Alles ausser ‚C').

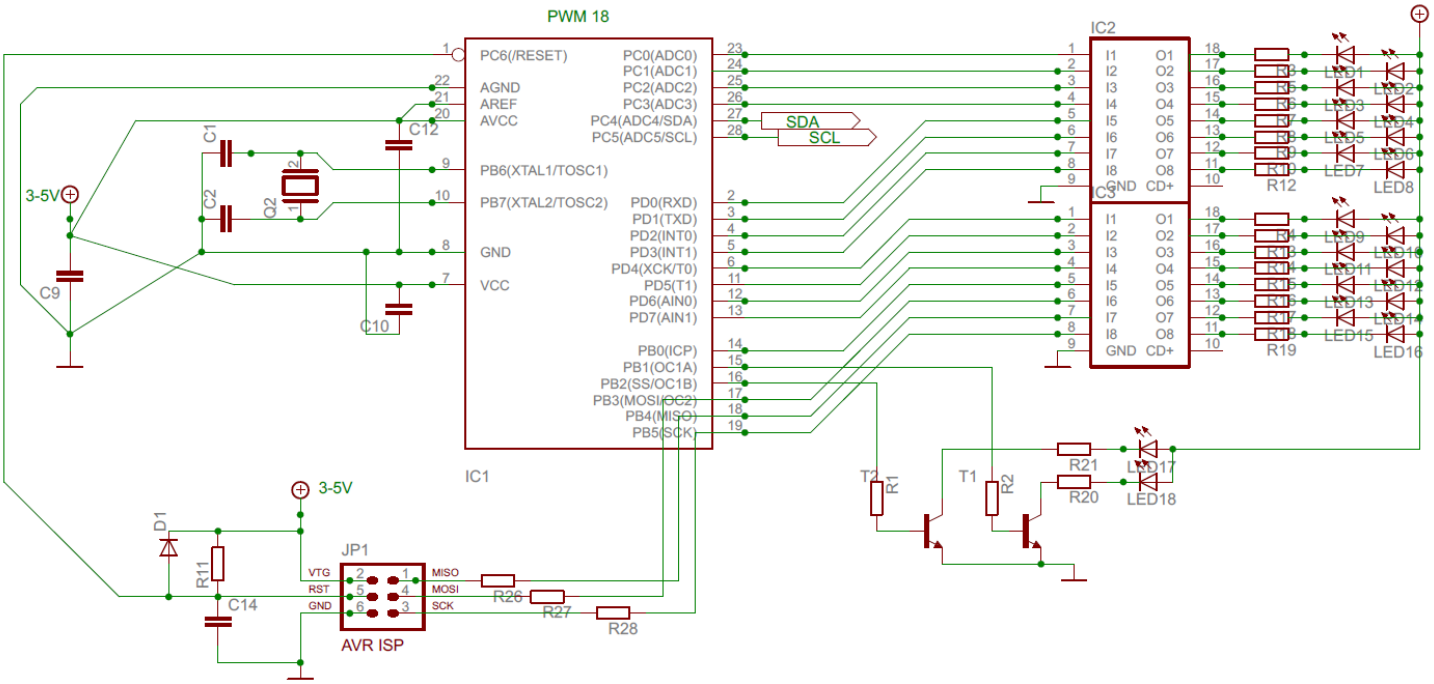
Somit wird die Speicherfunktion für Kanal 7 nicht ausgeführt.

6) **Byte4: ‚C'** (Großbuchstabe) Speicherbestätigung für Kanal 8
7) **Byte5: ‚C'** (Großbuchstabe) Speicherbestätigung für Kanal 9
8) Stoppkondition

Somit wurden die neuen PWM- Werte für Kanäle 6,8 und 9 im EEPROM gespeichert.

Schaltungsbeispiele 1/2

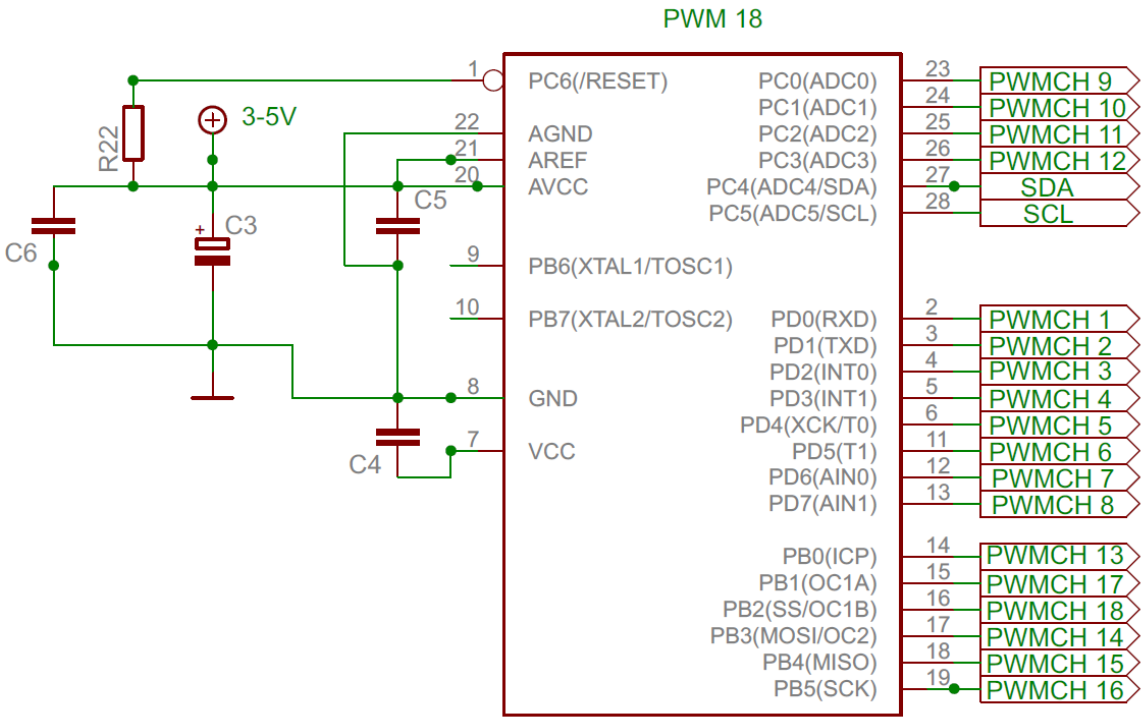
16- Kanal LED – Dimmer mit ULN 2803(8x Darlington Driver) und ISP- Programmer Interface.



Bauteil	Value	Notes
C1,2	22pF	
Q2	0-16MHz	Crystal Clock
C10, 12	100nF	Bypass Condensatoren
C9	100µF	Puffer Elko
D1	1N4148	Überspannungsschutz
R11	4,7kΩ	POR- Power On Reset
C14	10nF	POR- Power On Reset
R26, R28	4,7kΩ	
R1,2	4,7- 10kΩ	Vorwiderstand
T1,T2	BC817, 547	N-P-N Typ
R20, R21	150Ω	LED- Vorwiderstand(bei Rot 20mA)
R3- R19	150Ω	LED- Vorwiderstand(bei Rot 20mA)
LED 1-18	Rot(1,8-2V)	Leuchtdiode
IC1	ATMEGA 8	ATMEL ATMEGA 8
IC 2, 3	ULN2803 oder TD62083	8x Darlington Driver, TTL, CMOS-Inputs

Schaltungsbeispiele 2/2

Dieser Schaltplan stellt die Mindestbeschaltung dar.
Beschreibung:
R22(4,7k) und C6(10nF) bilden die POR- Power On Reset Schaltung.
Die Kondensatoren C4-C5(100nF) erfüllen eine Bypass-, und C3(10µF) Stützfunktionen.
Die Bypasskondensatoren sollten möglichst nahe am µController platziert werden.



Fehler: VCC(Pin7) sollte mit Plus verbunden werden.

Notizen

BEZEICHNUNG
PWM18 V1.0
PWM– Steht für PWM- Firmware
18- Für Anzahl der PWM- Kanäle
V1.0- Firmware Version

Diese Manual wurde von Firmware Version 1 bis 3 immer nur angepasst. Fehler sind somit vorprogrammiert.
Manual wird bei bedarf ausgebessert!

Change Log 2/2

PWM18 v3.3 DEV: Atmega8 (11.12.2009)

1) Das Statusregister SREG_3 gab keine Statusinformationen aus. Fehler wurde behoben.

PWM18 v3.2 DEV: Atmega8 (10.12.2009)

1) Adresspointer Reset hinzugefügt. Im Slavetransmitter Modus werden die Daten nach Adresse 0xFE wieder ab Register 0x00 ausgegeben.
2) Fehler behoben. Für Register 1-16 wurden nach der Initialisierung invertierte Werte ausgegeben.

PWM18 v3.1 DEV: Atmega8 (10.12.2009)

1) Der TWI- Driver wurde vereinfacht und Geschwindigkeitsoptimiert.
2) Das Hauptprogramm wurde an den neuen TWI- Driver angepasst.