

**SOFTWARE PWM CONTROLLER ATMEGA 8**  
**TYP: PWM18\_V7 (OHNE EEPROM)**

**INHALTSVERZEICHNIS**

Vorwort .....	2
Features .....	3
PWM- Controller Pinout .....	4
TWI- Registerfunktionen 1/2 .....	5
TWI- Registerfunktionen 1/2 .....	6
Das TWI- Protokoll .....	7
TWI_ADDR (Slaveadresse).....	8
PWM_vs_BUS .....	9
CS_Soft_PWM (Frequenz Kanal 1-16) .....	10
Res_Soft_PWM (Auflösung Kanal 1- 16) .....	11
CS_HW_PWM_16 (Frequenz Kanal 17,18).....	12
Res_HW_PWM_HB + Res_HW_PWM_LB (Auflösung Kanal 17, 18) .....	13
Initialisierungswerte .....	14
TWI (IIC) Ansteuerungs- Beispiele 1/2 .....	15
Schaltungsbeispiele 1/2 .....	16
Schaltungsbeispiele 2/2 .....	17
Notizen .....	18
Change Log 1/2 .....	19

## **Vorwort**

PWM 18 ist eine Firmware zur Generierung von 18- PWM Kanälen.

Die Kanäle 1-16 werden in Software generiert, und bieten Auflösungen bis zu 8-Bit. Die weiteren zwei Kanäle 17,18 werden im  $\mu$ Controller von Hardware Timer generiert, und bieten Auflösungen bis zu 16-Bit.

Einstellungen wie PWM- Frequenz und Auflösungen können über bestimmte Register variiert werden.

Die Kommunikation läuft über das TWI(IIC)- Interface.

Die TWI- Schnittstelle wurde aus Gründen der Flexibilität und Busfähigkeit gewählt. Des Weiteren stehen 186 Byte SRAM- Speicher für allgemeine Datenspeicherung zur Verfügung.

PWM18 V7 Firmware:

Diese Firmwareversion ist quasi der kleine Bruder der Version 3. Die dritte Version erlaubt es sämtliche Parameter im EEPROM- Speicher zu sichern, und im Einschaltmoment wieder zu laden(preload).

Diese Version(7) besitzt keine EEPROM- Speichermöglichkeiten.

Bei Bedarf kann die Initialisierung durch einen Master erfolgen.

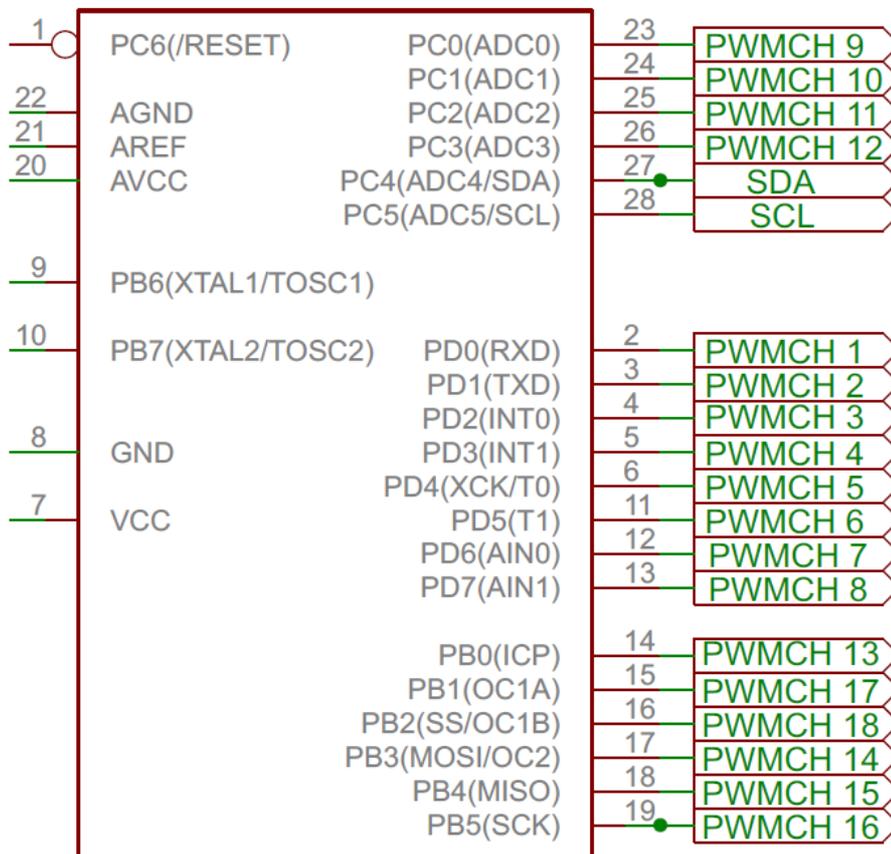
## **Features**

- ***TWI- Interface, Standard Mode, 7-Bit Adressierung***
- ***Watchdog Überwachung***
- ***Insgesamt 18- Frei wählbare PWM- Kanäle***
  - ***16 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***

# PWM- Controller Pinout

## AVR ATMEL ATMEGA 8 8-Bit $\mu$ Controller

### PWM 18



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

## TWI- Registerfunktionen 1/2

<b>TWI - Register [HEX]</b>	<b>Funktion</b>	<b>R/W*</b>	<b>Notes</b>
0	Temporary	R/W	1- Byte Speicher, ohne besondere Funktion
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	Temporary3	R/W	1- Byte Speicher, ohne besondere Funktion
16	Temporary2	R/W	1- Byte Speicher, ohne besondere Funktion
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 (Kanal 1-16)
1C	PWM_vs_BUS	R/W	Interruptzeit für Software PWM
3C- F5	SRAM	R/W	SRAM Speicher 186 Byte
FB	TWI_ADDR(IIC)	R/W	Ein Beispiel zum ändern der Slaveadresse 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 1/2

<b>TWI - Register [HEX]</b>	<b>Funktion</b>	<b>R/W*</b>	<b>Notes</b>
0	Temporary	R/W	1- Byte Speicher, ohne besondere Funktion
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	Temporary3	R/W	1- Byte Speicher, ohne besondere Funktion
16	Temporary2	R/W	1- Byte Speicher, ohne besondere Funktion
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 (Kanal 1-16)
1C	PWM_vs_BUS	R/W	Interruptzeit für Software PWM
3C- F5	SRAM	R/W	SRAM Speicher 186 Byte
FB	TWI_ADDR(IIC)	R/W	Ein Beispiel zum ändern der Slaveadresse 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.

# Das TWI- Protokoll

## Slave Receiver Modus

Es lassen sich je Schreibzugriff maximal 255 Byte empfangen.

Empfangsbuffer = 255 Byte.

Wird der Empfangsbuffer überfüllt, so werden die Daten verworfen mit der Annahme eines Bedienfehlers!

### 1) Schreiben

#### 2) Lesen

### 1) Schreiben

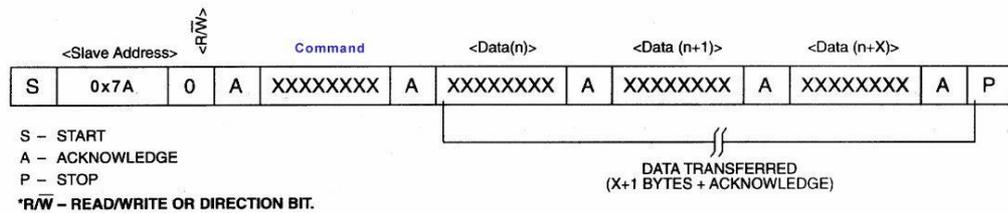
Ein einfacher Schreibzugriff besteht aus **mindestens 2 Byte**.

**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:

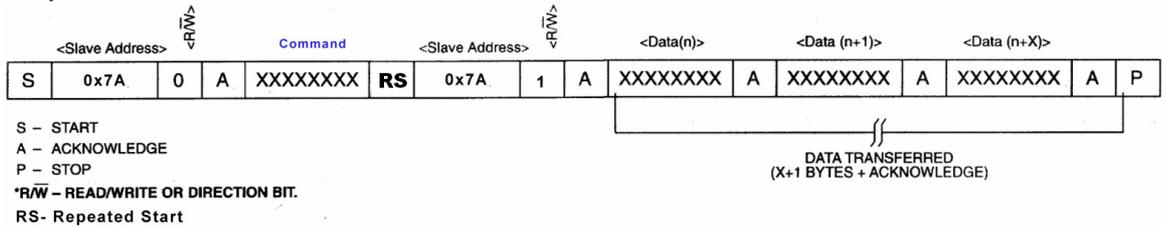


### 2) Lesen

Ein Lesezugriff besteht aus **nur 1 Byte**.

**Byte1 (Pointer Adresse):** Ist das Adressbyte siehe Seite Register Adressen „TWI- Registerfunktionen“

Beispiel:



## TWI\_ADDR (Slaveadresse)

### Beschreibung:

Das Register TWI\_ADDR- beinhaltet die TWI- Busadresse.

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	
Initial Value	0	0	0	0	0	0	0	0

- Bit 7..1 – TWA: TWI (Slave) Address Register  
Slaveadresse 0- 127
- Bit 0 – none

## 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!

Dieses Register ist Softwaremäßig gegen falsche Eingaben gesichert.

Werte die die Grenzen überschreiten werden Ignoriert.

**Mindestwert:** 90(Dez) oder 0x5A(HEX)

**Maxwert:** nicht definiert.

Der kleinste Wert von „PWM\_vs\_BUS“ darf nicht geringer als die längste Interruptzeit sein.

Interruptzeit Max. clock cycles = **68** ,

Interruptzeit Min. clock cycles = **46**

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

## 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.

Dieses Register ist Softwaremäßig gegen falsche Eingaben geschützt.

Werte die, die Grenzen überschreiten werden Ignoriert.

**Mindestwert:** nicht definiert.

**Maxwert:** 7(Dez) 0x07(HEX), CS22=1, CS21=1, CS20=1

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	$\text{clk}_{T2S}/(\text{No prescaling})$
0	1	0	$\text{clk}_{T2S}/8$ (From prescaler)
0	1	1	$\text{clk}_{T2S}/32$ (From prescaler)
1	0	0	$\text{clk}_{T2S}/64$ (From prescaler)
1	0	1	$\text{clk}_{T2S}/128$ (From prescaler)
1	1	0	$\text{clk}_{T2S}/256$ (From prescaler)
1	1	1	$\text{clk}_{T2S}/1024$ (From prescaler)

## Res\_Soft\_PWM (Auflösung Kanal 1- 16)

### Beschreibung:

Auflösung für Software PWM- Kanäle 1-16.

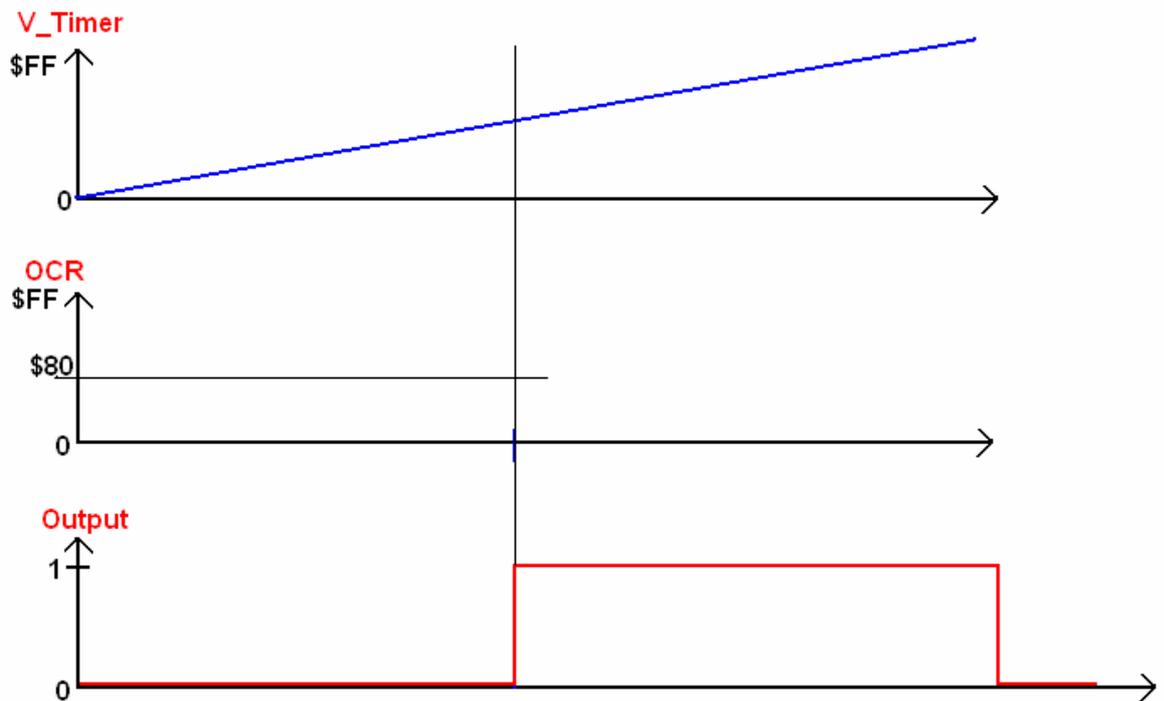
Wird die Auflösung von 8 auf 7-Bit umgestellt, so steigt die Software PWM- Frequenz auf das doppelte.

**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(PWM CH17) und OC1B(PWM CH18) ab.

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

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	$clk_{IO}/1$ (No prescaling)
0	1	0	$clk_{IO}/8$ (From prescaler)
0	1	1	$clk_{IO}/64$ (From prescaler)
1	0	0	$clk_{IO}/256$ (From prescaler)
1	0	1	$clk_{IO}/1024$ (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.

## **Res\_HW\_PWM\_HB + Res\_HW\_PWM\_LB (Auflösung Kanal 17, 18)**

### **Beschreibung:**

Dieses Doppelregister ist für die Auflösung von PWM- Output CH 17 und CH 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 ändern.

### **Lösung:**

	Slave Write	Register	Res HB	Res LB	
<b>Start</b>	<b>SLA+W</b>	<b>0x19</b>	<b>0xXX</b>	<b>0xXX</b>	<b>STOP</b>

## Initialisierungswerte

### Diese Firmware Benötigt nur die Flashdatei

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

### FUSE Settings:

- 1) RSTDISBL sollte Logisch „0“ bleiben.
- 2) WTDON, kann dauerhaft an sein. Wird softwaremäßig sowieso Aktiviert.

## TWI (IIC) Ansteuerungs- Beispiele 1/2

### Allgemeine Infos:

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

### Aufgabe1:

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

### Lösung:

	Slave Write	Pointer Adr.	Neuer Wert	
<b>Start</b>	<b>SLA+W</b>	<b>0x0A</b>	<b>0x64</b>	<b>STOP</b>

### Aufgabe2:

Mehrere Kanäle **auslesen**.

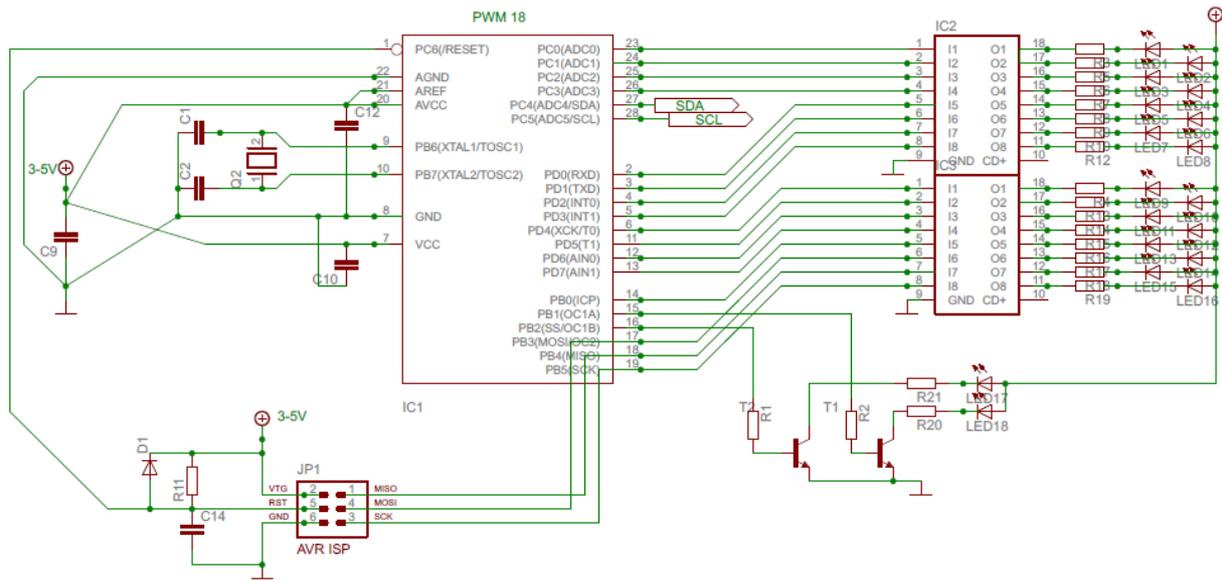
PWM- Kanäle 8-11 sollte ausgelesen werden.

### 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 von Kanal 11).
- 10) **Stoppkondition**  
**u.s.w**

## 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

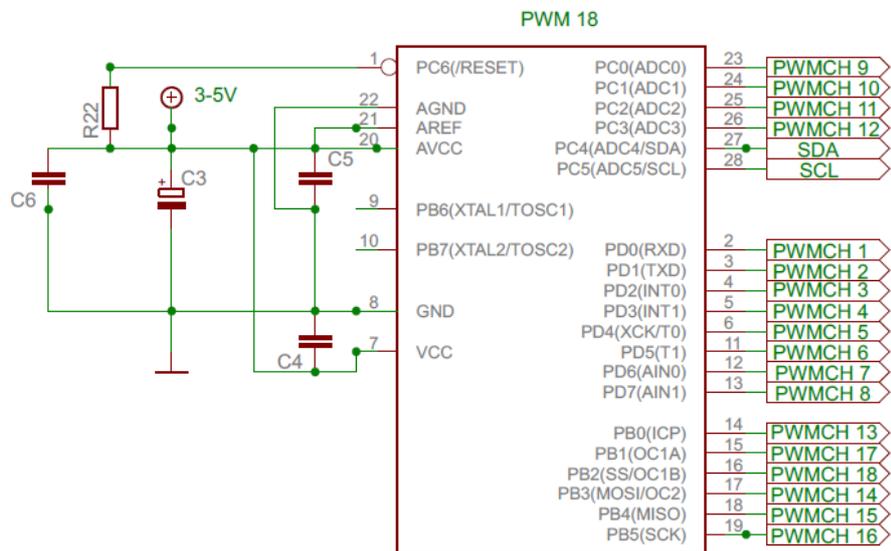
Dieser Schaltplan stellt die Mindestbeschaltung dar.

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.



## ***Notizen***

### **BEZEICHNUNG**

#### **PWM18 V1.0**

PWM– Steht für PWM- Firmware

18- Für Anzahl der PWM- Kanäle

V1.0- Firmware Version

## **Change Log 1/2**

### **PWM18 v7 DEV: Atmega8 (14.12.2009)**

- 1) Neue PWM- Controller Version ohne EEPROM- Speichernutzung.
- 2) Sämtliche Funktionen sind gleich geblieben, bis auf die Speicherbefehle und alles was dazugehörte.
- 3) Dadurch dass sämtliche Speicherfunktionen entfallen sind, ergaben sich ein Paar freie Register. Diese konnten wiederum für die Software PWM(Timer ISR) genutzt werden, weshalb die Inerrupzeit noch mal um gute 18 Takte leichter wurde.