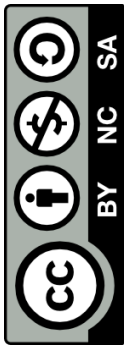
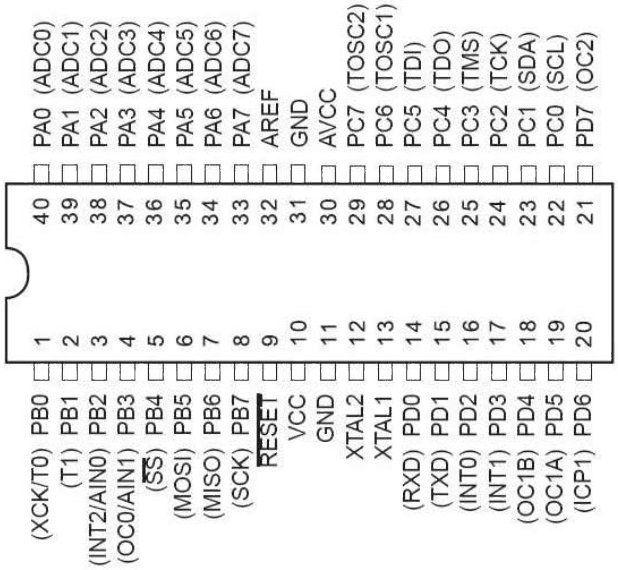


Dieses Cheatsheet soll bei der Arbeit mit dem ATmega16 von Atmel helfen und als Nachschlageblatt dienen.

Korrekturen an: steffen.vogel@rwth-aachen.de
Fork me on GitHub: github.com/stv0g

ATmega16

Pinbelegung



Facts

- 16 kByte Flash
- 1 kByte SRAM
- 512 Byte EEPROM
- 16 MHz Maximum System Clock
- 4 * 8 Bit GPIO Ports
- 4,5 - 5,5 V Operating Voltage
- 40 mA DC Current per IO Pin
- 100 mA DC Current per IO Port
- 200 mA (400 mA) DC Current on Supply Pins for PDIP (TQFP/MLF)

Register

Register	Beschreibung	Seite
SREG	ALU Status & Interrupt	9
MCUCR	MCU Control	32
MCUCSR	MCU Control and Status	41
GICR	Global Interrupt Control	48
GIFR	Global Interrupt Flags	70
TIMSK	Timer Interrupt Mask	85
TIFR	Timer Interrupt Flags	86
SPIDR	SPI Data	142
SPICR	SPI Control	140
SPISR	SPI Status	142
SFIOR	Special Function I/O	57
DDRx	IO Port Data Direction	66
PORTx	IO Port Data	66
PINx	IO Port Input Pins Address	66

Für eine Übersicht aller Register siehe Datenblatt Seite 331.

IO Ports

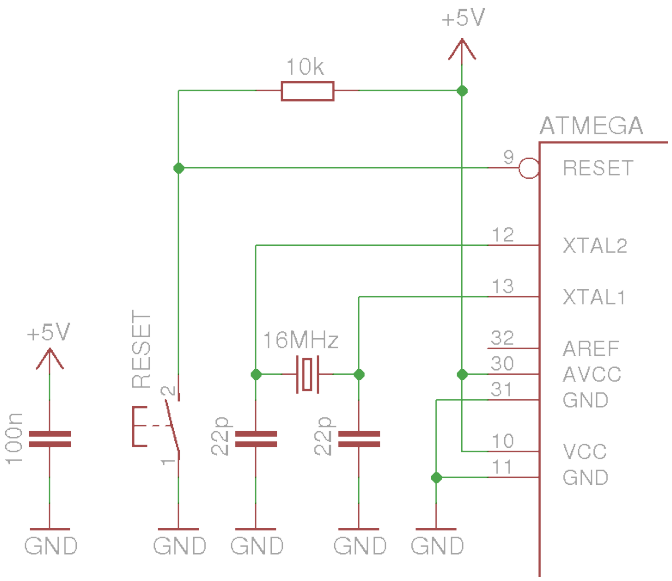
DDR	PORT	PUD	IO	Output
0	0	*	In	Tri-state
0	1	1	In	Tri-state
0	1	0	In	Tri-state (PullUp)
1	0	*	Out	Low (Sink)
1	1	*	Out	High (Source)

`PUD` steht für das `PUD`-Bit im `SFIOR` Register.

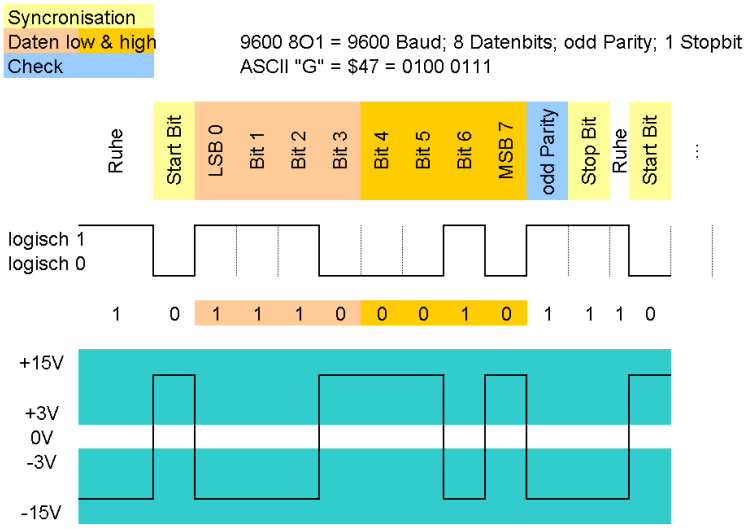
Interruptvektoren

Nr	Adr.	Quelle	Beschreibung
1	0x00	RESET	External, Power-on, Watchdog
2	0x02	INT0	External Interrupt Request 0
3	0x04	INT1	External Interrupt Request 1
4	0x06	TIMER2_COMP	Timer/Counter2 Compare Match
5	0x08	TIMER2_OVF	Timer/Counter2 Overflow
6	0x0A	TIMER1_CAPT	Timer/Counter1 Capture Event
7	0x0C	TIMER1_COMPA	Timer/Counter1 Compare Match A
8	0x0E	TIMER1_COMPB	Timer/Counter1 Compare Match B
9	0x10	TIMER1_OVF	Timer/Counter1 Overflow
10	0x12	TIMER0_OVF	Timer/Counter0 Overflow
11	0x14	SPI_STC	Serial Transfer Complete
12	0x16	USART_RXC	USART, Rx Complete
13	0x18	USART_UDRE	USART Data Register Empty
14	0x1A	USART_TXC	USART, Tx Complete
15	0x1C	ADC	ADC Conversion Complete
16	0x1E	EE_RDY	EEPROM Ready
17	0x20	ANA_COMP	Analog Comparator
18	0x22	TWI	Two-wire Serial Interface
19	0x24	INT2	External Interrupt Request 2
20	0x26	TIMER0_COMP	Timer/Counter0 Compare Match
21	0x28	SPM_RDY	Store Program Memory Ready

Grundbeschaltung



RS232 / UART



Baudrate: $UBRR = \frac{F\_CPU}{16 * Baudrate} - 1$

Hello World

```c
#include <avr/io.h>
#include <util/delay.h>

int main() {
  /* aktiviere Ausgänge */
  DDRB = 0xff;

  while (1) {
    /* lets blink */
    PORTB ^= (1<<PB0);

    /* wartet 1000 ms */
    _delay_ms(1000);
  }

  return 0; /* wird nie erreicht */
}
```

C-Tricks

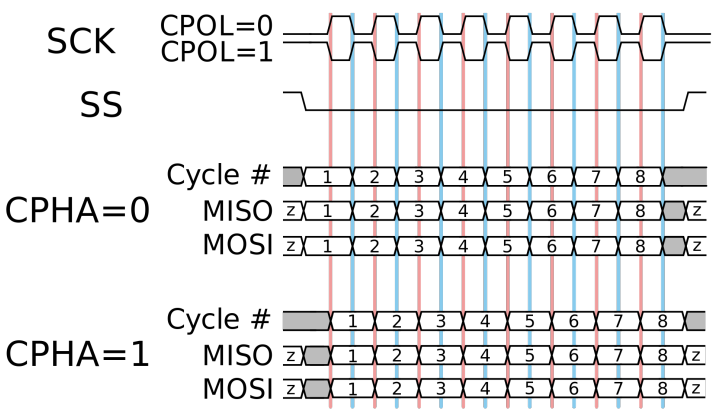
Operator	Beschreibung
`&`	binär UND
`|`	binär ODER
`^`	binär XOR (Togglen)
`~`	binär NOT (Invertieren)
`>>`	binär Shift-rechts
`<<`	binär Shift-links
`&&`	logisch UND
`||`	logisch ODER
`!`	logisch NOT (Invertieren)

Auch in Kombination mit direkter Zuweisung: `&=`, `|=`, `^=`, ...

Datentyp	Beschreibung	Bereich
`void *`	void-Pointer auf beliebige Daten	$0 .. 2^{16}$
`int8_t` `char`	8 Bit vorzeichenbehaftet	-127 .. 128
`uint8_t` `unsigned char`	8 Bit vorzeichenlos	0 .. 255
`int16_t`	16 Bit vorzeichenbehaftet	
`uint16_t`	16 Bit vorzeichenlos	

Manipulation	Beispiel
Setzen	`PORTA |= (1 << PB0)`
Löschen	`PORTA &= ~(1 << PB0)`
Togglen	`PORTA ^= (1 << PB0)`
Abfragen	`if (PORTA & (1 << PB0)) ...`

SPI Modi



Integer Literale

Der avr-gcc unterstützt als Erweiterung des C-Standards folgende Notationen für Konstanten (Literale):

Zahlensystem	Beispiel
Dezimal	`123`
Hexadezimal	`0x7B`
Octal	`0173`
Binär	`0b01111011`

Dez - Hex - Bin - Okt

Dezimal	Hex	Binär	Oktal
1	1	1	1
2	2	10	2
3	3	11	3
4	4	100	4
5	5	101	5
6	6	110	6
7	7	111	7
8	8	1000	10
9	9	1001	11
10	A	1010	12
11	B	1011	13
12	C	1100	14
13	D	1101	15
14	E	1110	16
15	F	1111	17

avr-libc

Funktionen

`#include <avr/cpufunc.h>`

`void _NOP()` no operation

`#include <avr/interrupt.h>`

`void cli()` clear interrupts

`void sei()` set interrupts

`#include <avr/sleep.h>`

`void sleep_cpu()` put the device into sleep mode

`void sleep_disable()` clear the SE (sleep enable) bit

`void sleep_enable()` set the SE (sleep enable) bit

`#include <avr/wdt.h>`

`void wdt_reset()` reset watchdog timer

`void wdt_enable(value)` enable watchdog timer

`void wdt_disable()` disable watchdog timer

`#include <util/delay.h>`

`void _delay_ms(double ms) ms <= 262.14 ms / F_CPU` in MHz

`void _delay_us(double us) ms <= 768 us / F_CPU` in MHz

Makros

Makro	Beschreibung
RAMEND	letzte Adresse des RAMs
E2END	letzte Adresse des EEPROMs
FLASHEND	letze (Byte-)Adresse des Flashs
SPM_PAGESIZE	Größe einer Flash-Page in Bytes
E2PAGESIZE	Größe einer EEPROM-Page in Bytes

GNU AVR Toolchain

avr-gcc Kompiler Frontend

avr-gdb Debugger

avr-objcopy Objektdateien konvertieren

avr-objdump Objektdateien dumpen, disassemblieren etc.

Argument	Beschreibung
-mmcu=MCU	Ziel Mikrocontroller
-E	nur Präprozessor aufrufen
-c	nur kompileren, nicht linken
-o <file>	Ausgabe in <file>
-Wa,<options>	Assembler Optionen
-Wp,<options>	Preprocessor Optionen
-Wl,<options>	Linker Optionen
-g	Generiere Debugging Informationen
-O1,2,3,s	Optimierung (1-3: Geschwindigkeit, s: Größe)
-f	Kompiler Optimierungen / Tuning
-Wall,extra	Kompiler Warnungen aktivieren
-B <directory>	Füge <directory> zum Suchpfad hinzu
-std=c99	Nutze C99 Standard
-lm	Linke mit Mathematik Bibliothek

avrdude

Argument	Beschreibung
-p m16	Mikrocontroller Typ
-c usbasp	Programmer Hardware
-P usb	Programmer Port
-V	Verifikation unterdrücken
-e	Mikrocontroller löschen
-U <mem>:<op>:<file>:<fmt>	
<mem> = {flash,eeprom}	Speicher
<op> = {r,w,v}	Read, Write, Verify
<file>	Dateiname
<fmt> = {a,i,r,m,h,o,b}	Dateiformat

Beispiele:

```
avrdude -p m8 -c usbasp -U flash:w:USBasp-2013-03-05-16Mhz.hex
avrdude -p m8 -c usbasp -U lfuse:r:-:h -U hfuse:r:-:h
avrdude -p m8 -c usbasp -U lfuse:w:0xab:m -U hfuse:w:0xbc:m
```

Widerstandstabelle

Farbe	Ringe				
	1	2	3	4	5
Silber				$\cdot 10^{-2}$	$\pm 10\%$
Gold				$\cdot 10^{-1}$	$\pm 5\%$
Schwarz	0	0	0	$\cdot 10^{0}$	
Braun	1	1	1	$\cdot 10^{1}$	$\pm 2\%$
Rot	2	2	2	$\cdot 10^{2}$	
Orange	3	3	3	$\cdot 10^{3}$	
Gelb	4	4	4	$\cdot 10^{4}$	
Grün	5	5	5	$\cdot 10^{5}$	
Blau	6	6	6	$\cdot 10^{6}$	
Violett	7	7	7	$\cdot 10^{7}$	
Grau	8	8	8	$\cdot 10^{8}$	
Weiß	9	9	9		

Kondensatorbeschriftung

Grundsätzlich gilt, dass Kleinbuchstaben für die Einheit stehen. Wobei auf F für Farad verzichtet wird.

1. Aufgredruckte Einheit: p => Pikofarad, n => Nanofarad. Buchstabe ist durch ein Komma zu ersetzen

2. Kodierter dreistelliger Aufdruck: Kapazität entspricht den ersten beiden Stellen in Pikofarad. Letzte Stelle ist Zehnerexponent

3. Einheit und Dezimalpunkt fehlen: Kapazität in Pikofarad (Kerko)

4. Einheit fehlt, Dezimalpunkt vorhanden: Kapazität in Mikrofarad (Folienkondensator)

Pinouts

Wo nicht anders angegeben stehen Ïn"& Öut"für die Richtung aus Sicht des Computers / Adapters.

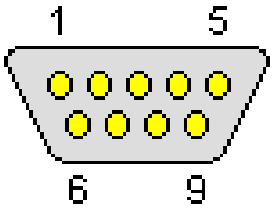


Abbildung 0.1: RS-232

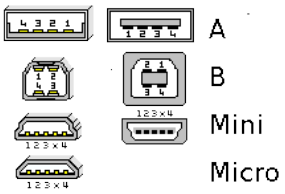


Abbildung 0.2: USB

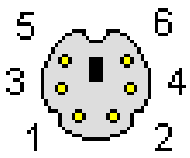


Abbildung 0.3: PS-2 Keyboard

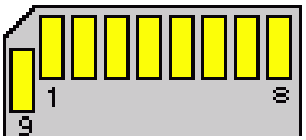


Abbildung 0.4: SD/MMC Karte

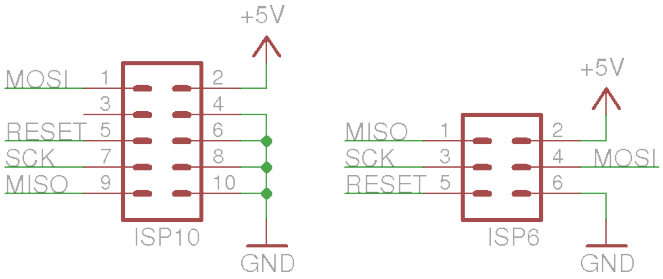
RS232

#	Pin	Typ	Farbe	Beschreibung
1	DCD	In	Braun	Data Carrier Detect
2	RXD	In	Rot	Receive Data
3	TXD	Out	Orange	Transmit Data
4	DTR	Out	Gelb	Data Terminal Ready
5	GND	Supply	Grün	System Ground
6	DSR	In	Blau	Data Set Ready
7	RTS	Out	Violett	Request to Send
8	CTS	In	Grau	Clear to Send
9	RI	In	Schwarz	Ring Indicator

USB

#	Name	Typ	Farbe	Beschreibung
1	VCC	Supply	Rot	+5 VDC
2	D-	I/O	Weiß	Data -
3	D+	I/O	Grün	Data +
4	Supply	GND	Schwarz	Ground

AVR ISP



#10	#6	Name	Typ	Beschreibung
1	4	MOSI	Out	Master-Out Slave-In
2	2	VCC	Supply	5 VDC
3	-	NC	-	Not Connected
4	-	GND	Supply	Ground
5	5	RESET	Out	Target Reset
6	6	GND	Supply	Ground
7	3	SCK	Out	Clock
8	-	GND	Supply	Ground
9	1	MISO	In	Master-In Slave-Out
10	-	GND	Supply	Ground

AVR JTAG

#	Name	Typ	Beschreibung
1	TCK	Out	Test Clock
2 GND	Supply	Ground	
3	TDO		Test Data Output
4	VTref	In	Voltage Sense
5	TMS	Out	Test Mode Select
6	nSRST	Out	Target Reset
7	(Vcc)	Supply	5 VDC (nicht immer belegt)
8	(nTRST)		(nicht immer belegt)
9	TDI	Out	Test Data Input
10	GND	Supply	Ground

PS2

#	Name	Typ	Beschreibung
1	DATA	In	Key Data
2	NC	-	Not Connected
3	GND	Supply	Ground
4	VCC	Supply	+5 VDC
5	CLK	In	Clock
6	NC	-	Not Connected

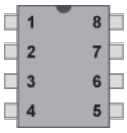
HD44780

#	Name	Typ	Beschreibung
1	Vss	Supply	GND
2	Vcc	Supply	+5 VDC
3	VE	In	Contrast adjustment
4	RS	In	Register Select
5	RW	In	Read/Write
6	EN	In	Clock (Enable)
7	D0	In	Data Bit 0
8	D1	In	Data Bit 1
9	D2	In	Data Bit 2
10	D3	In	Data Bit 3
11	D4	In	Data Bit 4
12	D5	In	Data Bit 5
13	D6	In	Data Bit 6
14	D7	In	Data Bit 7
15	BLA	Supply	Backlight Anode (+)
16	BLK	Supply	Backlight Cathode (-)

SD Karte

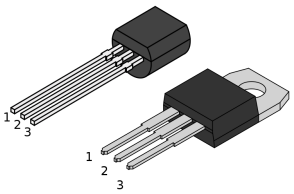
#	SPI Mode		
	Name	Typ	Beschreibung
1	CS	In	Chip selection in low status
2	DI	In	Data input
3	VSS	Supply	GND
4	VDD	Supply	Power supply
5	SCLK	In	Clock
6	VSS2	Supply	GND
7	DO	Out	Data output
8	NC	-	Not Connected
9	NC	-	Not Connected

DIP, DIL, SOIC ...



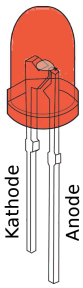
Die Pins von IC werden gegen den Uhrzeigersinn, ausgehend von einer Kerbe oder Kreis am schmalen Ende des Bausteins, gezählt.

TO-92, TO-220



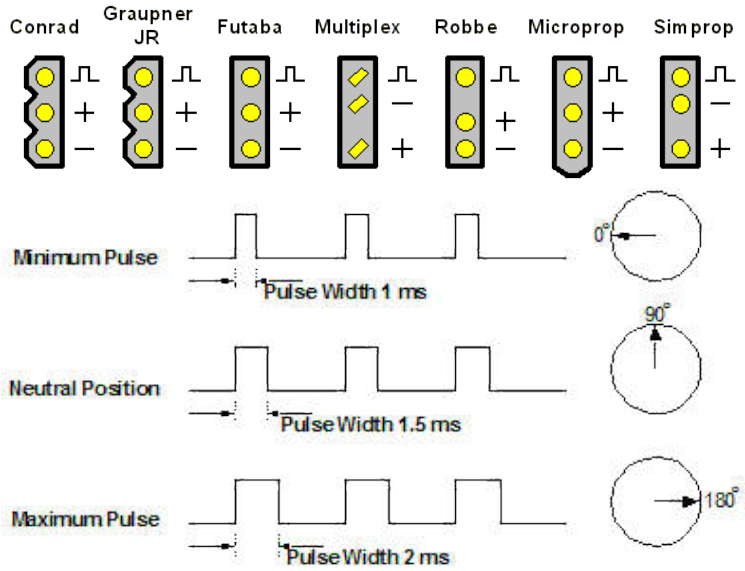
#	1	2	3
78L*xx*	In	GND	Out
BC547	B	C	E
IRF N-FET	G	D	S

LED



Farbe	Halbleiter	Standard (20mA)	Low-Current (2mA)
rot	GaAsP	1,6 V	1,9 V
	GaP	2,1 V	
orange	GaAsP	1,8 V	
grün	GaP	2,1 V	1,9 V
gelb	GaP	2,2 V	2,4 V
blau	GaN	2,9 V	

Servos



Abkürzungen

GND Masse

V_{cc}, V_{dd} positive Versorungsspannung

V_{ee}, V_{ss} negative Versorungsspannung (meist GND)

AV_{cc} Analoge Versorungsspannung

V_{ref} Analoge Referenzspannung

AGND Analoge Masse

SCLK, CLK SPI: Clock

SS, CE, CS SPI: Chip Enable/Select

MISO SPI: Master-In Slave-Out

MOSI SPI: Master-Out Slave-In

SDA I^2C: Serial Data

SCL I^2C: Serial Clock

O, Q, $\bar{O}$, $\bar{Q}$ Output (invertiert)

EN Enable

OE Output Enable

WE Write Enable

NC Not Connected

TDO JTAG: Test Data Output

TDI JTAG: Test Data Input

TCK JTAG: Test Clock