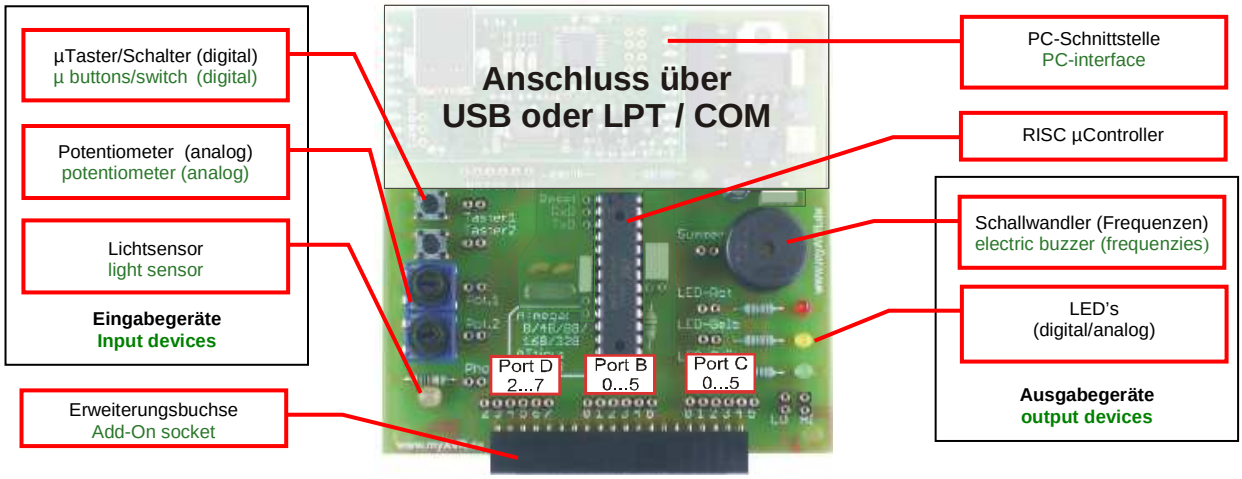
Experimentierplattform:

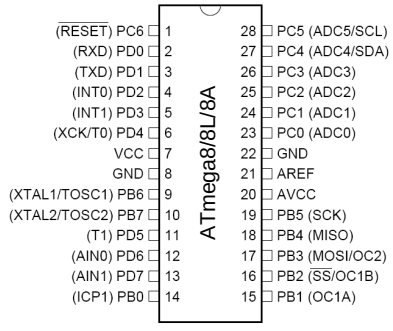
myAVR Board MK2 USB Version 2.2 / myAVR Board light Version 1.1 / (myAVR Board MK1 LPT Version 1.6)

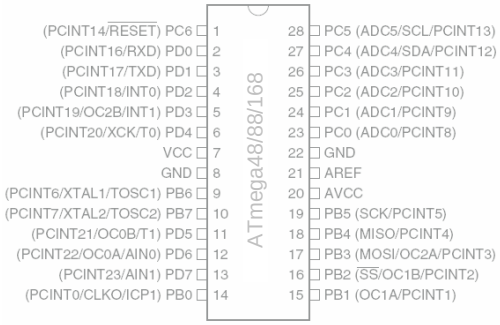


I/O Register

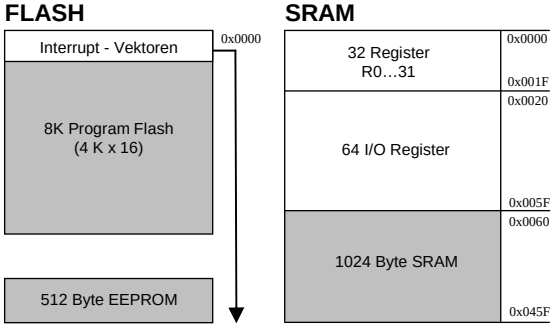
I/O	MEM	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0x3F	0x5F	SREG	I	T	H	S	V	N	Z	C
0x3E	0x5E	SPH	–	–	–	–	–	SP10	SP9	SP8
0x3D	0x5D	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
0x3C	0x5C	Reserved								
0x3B	0x5B	GICR	INT1	INT0	–	–	–	–	IVSEL	IVCE
0x3A	0x5A	GIFR	INTF1	INTF0	–	–	–	–	–	–
0x39	0x59	TIMSK	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	–	TOIE0
0x38	0x58	TIFR	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	–	TOV0
0x37	0x57	SPMCR	SPMIE	RWWSB	–	RWWSRE	BLBSET	PGWRT	PGERS	SPMEN
0x36	0x56	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE
0x35	0x55	MCUCR	SE	SM2	SM1	SM0	ISC11	ISC10	ISC01	ISC00
0x34	0x54	MCUCSR	–	–	–	–	WDRF	BORF	EXTRF	PORF
0x33	0x53	TCCR0	–	–	–	–	–	CS02	CS01	CS00
0x32	0x52	TCNT0	Timer/Counter0 (8 Bits)							
0x31	0x51	OSCCAL	Oscillator Calibration Register							
0x30	0x50	SFIOR	–	–	–	–	ACME	PUD	PSR2	PSR10
0x2F	0x4F	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	FOC1A	FOC1B	WGM11	WGM10
0x2E	0x4E	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10
0x2D	0x4D	TCNT1H	Timer/Counter1 – Counter Register High byte							
0x2C	0x4C	TCNT1L	Timer/Counter1 – Counter Register Low byte							
0x2B	0x4B	OCR1AH	Timer/Counter1 – Output Compare Register A High byte							
0x2A	0x4A	OCR1AL	Timer/Counter1 – Output Compare Register A Low byte							
0x29	0x49	OCR1BH	Timer/Counter1 – Output Compare Register B High byte							
0x28	0x48	OCR1BL	Timer/Counter1 – Output Compare Register B Low byte							
0x27	0x47	ICR1H	Timer/Counter1 – Input Capture Register High byte							
0x26	0x46	ICR1L	Timer/Counter1 – Input Capture Register Low byte							
0x25	0x45	TCCR2	FOC2	WGM20	COM21	COM20	WGM21	CS22	CS21	CS20
0x24	0x44	TCNT2	Timer/Counter2 (8 Bits)							
0x23	0x43	OCR2	Timer/Counter2 Output Compare Register							
0x22	0x42	ASSR	-	–	–	–	AS2	TCN2UB	OCR2UB	TCR2UB
0x21	0x41	WDTCR	–	–	–	WDCE	WDE	WDP2	WDP1	WDP0
0x20[1]	0x40[1]	UBRRH	URSEL	–	–	–	UBRR[11:8]			
		UCSRC	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL
0x1F	0x3F	EEARH	–	–	–	–	–	–	–	EEAR8
0x1E	0x3E	EEARL	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0
0x1D	0x3D	EEDR	EEPROM Data Register							
0x1C	0x3C	EECR	–	–	–	–	EERIE	EEMWE	EEWE	EERE
0x1B	0x3B	Reserved								
0x1A	0x3A	Reserved								
0x19	0x39	Reserved								
0x18	0x38	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
0x17	0x37	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
0x16	0x36	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
0x15	0x35	PORTC	–	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0
0x14	0x34	DDRC	–	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
0x13	0x33	PINC	–	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
0x12	0x32	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
0x11	0x31	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
0x10	0x30	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
0x0F	0x2F	SPDR	SPI Data Register							
0x0E	0x2E	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X
0x0D	0x2D	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
0x0C	0x2C	UDR	USART I/O Data Register							
0x0B	0x2B	UCSRA	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM
0x0A	0x2A	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
0x09	0x29	UBRRL	USART Baud Rate Register Low byte							
0x08	0x28	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0
0x07	0x27	ADMUX	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0
0x06	0x26	ADCSRA	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0
0x05	0x25	ADCH	ADC Data Register High byte							
0x04	0x24	ADCL	ADC Data Register Low byte							
0x03	0x23	TWDR	Two-wire Serial Interface Data Register							
0x02	0x22	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE
0x01	0x21	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0
0x00	0x20	TWBR	Two-wire Serial Interface Bit Rate Register 168							

Pinbelegung / pin assignments





Speicheraufbau / memory structure



Interruptvektoren / interrupt vectors

Vector No.	Program Adress	Source	Interrupt Definition
1	0x000	RESET	External Pin,Power-on Reset,Brown-out Reset, and Watchdog Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	TIMER2 COMP	Timer/Counter2 Compare Match
5	0x004	TIMER2 OVF	Timer/Counter2 Overflow
6	0x005	TIMER1 CAPT	Timer/Counter1 Capture Event
7	0x006	TIMER1 COMPA	Timer/Counter1 Compare MatchA
8	0x007	TIMER1 COMPB	Timer/Counter1 Compare MatchB
9	0x008	TIMER1 OVF	Timer/Counter1 Overflow
10	0x009	TIMER0 OVF	Timer/Counter0 Overflow
11	0x00A	SPI; STC	Serial Transfer Complete
12	0x00B	USART , RXC	USART, Rx Complete
13	0x00C	USART , UDRE	USART, Data Register Empty
14	0x00D	USART , TXC	USART, TX Complete
15	0x00E	ADC	ADC Conversion Complete
16	0x00F	EE_RDY	EEPROM Ready
17	0x010	ANA_COMP	Analog Comperator
18	0x011	TWI	Two-wire serial Interface
19	0x012	SPM_RDY	Store Program Memory Ready

Arbeitsregister / working registers

Register			Adresse	Bemerkung
R0			0x00	siehe LPM
R1			0x01	**KEIN** ADIW, SUBI, SUBIW, ANDI, ORI, CPI, LDI,
...			...	
R15			0x0F	
R16			0x10	keine Einschränkungen
...			...	
R26	XL	X	0x1A	siehe LD/ST
R27	XH		0x1B	
R28	YL	Y	0x1C	siehe LD/ST
R29	YH		0x1D	
R30	ZL	Z	0x1E	siehe LD/ST
R31	ZH		0x1F	

Portfunktionen / port functions

Name	Port	Funktion
Externe Interrupts	D2-3	INT0-1
Analog-Digital-Converter	C0-5	ADC0-5
Analog-Komparator	D6-7	AIN0-1
myAVR LCD	D2	R/S
	D3	Enable
	D4-7	DB4-7
	B0 (optional)	R/W
	B1 (optional)	Backlight
TWI / I²C	C4, C5	SDA, SCL
SPI-Bus AVR-ISP	B3	MOSI
	B4	MISO
	B5	SCK
	B2 (optional)	SS
UART (RS232)	D0, D1	RxD, TxD
	D4 (für Sync.Mode)	XCK
Reset	C6	RST
Quarz/Resonator/Takt	B6, B7	XTAL1, XTAL2
Timer/Counter-Output (PWM, Waveform)	B1, B2, B3	OC1A, OC1B, OC2
Timer/Counter Input-Clock	D4, D5	T0, T1
Timer/Counter InputCapture	B0	ICP1

Befehlssatz / instruction set

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
ADIW	Rdl,K	Add Immediate to Word	Rdh:Rdl ← Rdh:Rdl + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
SBIW	Rdl,K	Subtract Immediate from Word	Rdh:Rdl ← Rdh:Rdl - K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← 0xFF − Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← 0x00 − Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	K Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd • (0xFF - K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd − 1	Z,N,V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← 0xFF	None	1
MUL	Rd, Rr	Multiply Unsigned	R1:R0 ← Rd x Rr	Z,C	2
MULS	Rd, Rr	Multiply Signed	R1:R0 ← Rd x Rr	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	R1:R0 ← Rd x Rr	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	R1:R0 ← (Rd x Rr) << 1	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	R1:R0 ← (Rd x Rr) << 1	Z,C	2
ULSU	Rd, Rr	Fractional Multiply Signed wiTh Unsigned	R1:R0 ← (Rd x Rr) << 1	Z,C	2
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
IJMP		Indirect Jump to (Z)	PC ← Z	None	2
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
ICALL		Indirect Call to (Z)	PC ← Z	None	3
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	I	4
CP	Rd,Rr	Compare	Rd − Rr	Z,N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	Rd − Rr − C	Z,N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	Rd − K	Z,N,V,C,H	1
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1/2/3
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) PC ← PC + 2 or 3	None	1/2/3
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) PC ← PC + 2 or 3	None	1/2/3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) PC ← PC + 2 or 3	None	1/2/3
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) PC ← PC + 2 or 3	None	1/2/3
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC←PC+k + 1	None	1/2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC←PC+k + 1	None	1/2
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1/2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1/2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1/2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1/2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1/2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1/2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1/2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1/2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1/2
BRLT	k	Branch if Less Than Zero, Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1/2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1/2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1/2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1/2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1/2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1/2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1/2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1/2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1/2
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM (Data Mem.)	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM (Data Mem.)	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z+1	None	3
SPM		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
MCU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		WatchdogReset	(see specific descr. for WDR/timer)	None	1

```
;+--------------------------------------------------------------------------
;| Titel       : Beispiel IN/OUT für das myAVR-Board
;+--------------------------------------------------------------------------
;| Funktion    : Solange Taster 1 gedrückt ist, wird eine LED eingeschaltet
;| Schaltung   : PORTD2=Taster1, PORTB0=LED-Rot
;+--------------------------------------------------------------------------
.include    "AVR.H"
;--- Reset and Interrupt vector  ----  ;VNr.  Beschreibung -----
            rjmp    main                ;1     POWER ON RESET
            reti                        ;2     Int0-Interrupt
            reti                        ;3     Int1-Interrupt
            reti                        ;4     TC2 Compare Match
            reti                        ;5     TC2 Overflow
            reti                        ;6     TC1 Capture
            reti                        ;7     TC1 Compare Match A
            reti                        ;8     TC1 Compare Match B
            reti                        ;9     TC1 Overflow
            reti                        ;10    TC0 Overflow
            reti                        ;11    SPI, STC Serial Transfer Complete
            reti                        ;12    UART Rx Complete
            reti                        ;13    UART Data Register Empty
            reti                        ;14    UART Tx Complete
            reti                        ;15    ADC Conversion Complete
            reti                        ;16    EEPROM Ready
            reti                        ;17    Analog Comparator
            reti                        ;18    TWI (IC) Serial Interface
            reti                        ;19    Store Program Memory Ready
;--- Start, Power ON, Reset ---------------------------------------------
main:       ldi     r16,hi8(RAMEND)     ; Stack Initialisierung
            out     SPH,r16
            ldi     r16,lo8(RAMEND)
            out     SPL,r16
            cbi     DDRD,2              ; PORTD2 auf Eingang
            sbi     PORTD,2             ; PORTD2 Pullup
            sbi     DDRB,0              ; PORTB0 auf Ausgang
;------------------------------------------------------------------------
mainloop:   ldi     r16,0               ; Wert bei Taste nicht gedrückt
            sbis    PIND,2              ; Taste auswerten
            ldi     r16,1               ; Wert bei Taste gedrückt
            out     PORTB,r16           ; LED an/aus
            rjmp    mainloop
;------------------------------------------------------------------------
```