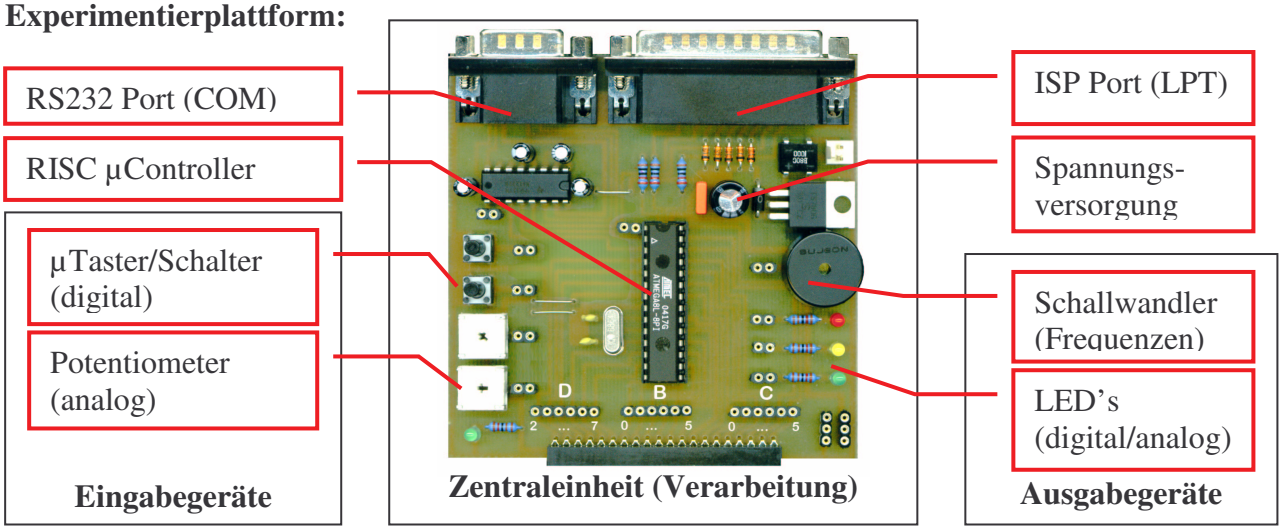
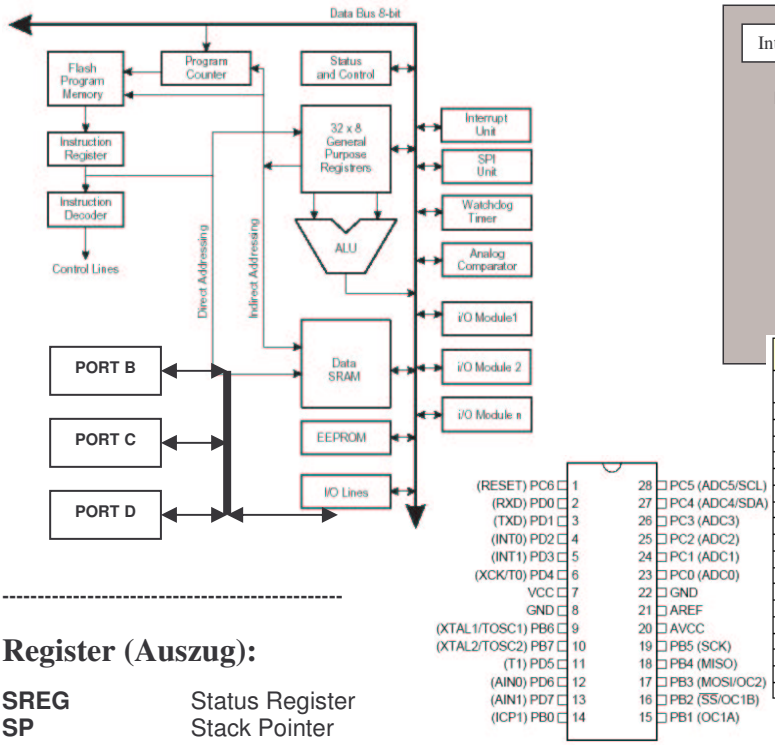
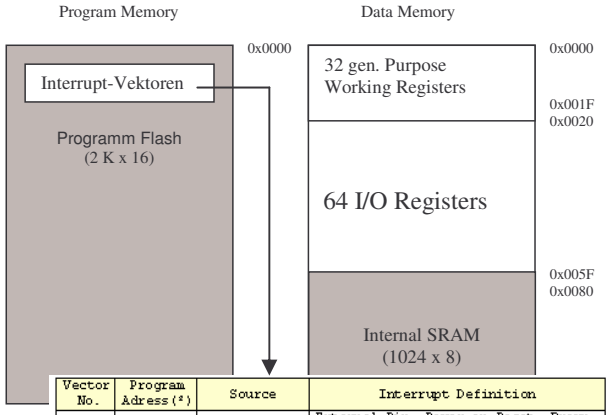
Experimentierplattform:



Prozessor:





Vector No.	Program Adress(²)	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 Match A
8	0x007	TIMER1 COMPB	Timer/Counter1 Compare Match B
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

Register (Auszug):

SREG	Status Register
SP	Stack Pointer
GIMSK	General Interrupt MaSK Register
GIFR	General Interrupt Flag Register
MCUCR	MCU general Control Register
MCUSR	MCU general Status Register
WDTCR	Watchdog Timer Control Register

32 General Purpose Working Register			
Register	Adresse		
R0	0x00		
R1	0x02		
R2	0x03		
...			
R13	0x0D		
R14	0x0E		
R15	0x0F		
R16	0x10		
R17	0x11		
...			
R26	0x1A	Low Byte	X
R27	0x1B	High Byte	
R28	0x1C	Low Byte	Y
R29	0x1D	High Byte	
R30	0x1E	Low Byte	Z
R31	0x1F	High Byte	

Flag-Register (Status-Register, SREG):

7	6	5	4	3	2	1	0
I	T	H	S	V	N	Z	C

I = Interrupt, **T** = Richtung, **H**= Halb-Überlauf , **S** = Vorzeichen,
V = Überlauf Zweierkomplement, **N** = Negativ, **Z** = Null, **C** = Überlauf

---

I/O Ports / Register (Auszug)

	Data, Out	Data Direction	Input
Port B:	**PORTB** = 0x18	**DDRB** = 0x17	**PINB** = 0x16
Port C:	**PORTC** = 0x15	**DDRC** = 0x14	**PINC** = 0x13
Port D:	**PORTD** = 0x12	**DDRD** = 0x11	**PIND** = 0x10

SPI	**SPDR**=SPI Daten, **SPSR**=SPI Status, **SPRC**=SPI Control,
UART	**UDR** = UART Daten, **UCSRA/UCSRB** = Control & Status, **UBBR** = Baudrate, **UBRRHI** = BaudHI,
COMPERATOR	**ACSR** = Analog Comperator Status & Control,
ADC	**ADMUX** = Port-Select, **ADCSR** = Status & Control, **ADCH** = DatenHI, **ADCL** = DatenLO,
TIMER allgemein	**TIMR**= Timer Interrupt Maske, **TIFR** = Timer Interrupt Flag,
TIMER0	**TCCR0** = Timer0 Control, **TCNT0** = Timer0 Daten,
TIMER1	**TCCR1A/TCCR1B** = Timer1 Control HI&LO, **TCNT1H/TCNT1L** = Timer1 Daten HI&LO

Befehlssatz (Auszug)

ARITHMETIC AND LOGIC INSTRUCTIONS			
ADD Rd, Rr (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 (SBC Rd, Rr)	Subtract (with Carry) Two Registers Rd ← Rd − Rr (− C)	Z,C,N,V,H	1
SUBI Rd, K (SBCI Rd, K)	Subtract (with Carry) Constant from Register 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 ← $FF − Rd	Z,C,N,V	1
NEG Rd	Two's Complement Rd ← $00 − Rd	Z,C,N,V,H	1
SBR Rd,	K Set Bit(s) in Register Rd ← Rd v K	Z,N,V	1
CBR Rd,	K Clear Bit(s) in Register Rd ← Rd • ($FF − 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 (SER Rd)	Clear Register Rd ← Rd □ Rd (Set Register Rd ← $FF)	Z,N,V (None)	1
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 (RETI)	Subroutine Return (Interrupt Return) PC ← STACK	None (I)	4
CPSE	Rd, Rr Compare, Skip if Equal if (Rd = Rr) PC ← PC + 2 or 3	None	1/2/3
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
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	ranch 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
DATA TRANSFER INSTRUCTIONS			
MOV Rd, Rr	Move between Registers Rd ← 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 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 (k) ← Rr	None	2
LPM	Load Program Memory R0 ← (Z)	None	3
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 / CLC	Set Carry C ← 1 / Clear Carry C ← 0	C	1
SEN / CLN	Set Negative Flag N ← 1 / Clear Negative Flag N ← 0)	N	1
SEZ / CLZ	Set Zero Flag Z ← 1 / Clear Zero Flag Z ← 0	Z	1
SEI / CLI	Global Interrupt Enable I ← 1 / Global Interrupt Disable I ← 0	I	1
SES / CLS	Set Signed Test Flag S ← 1 / Clear Signed Test Flag S ← 0	S	1
SEV / CLV	Set Two's Complement Overflow V←1 / Clear Two's Complement Overflow V←0	V	1
SET / CLT	Set T in SREG T ← 1 / Clear T in SREG T ← 0	T	1
SEH / CLH	Set Half-carry Flag in SREG H←1 / Clear Half-carry Flag in SREG H←0	H	1
NOP	No Operation	None	1
SLEEP	Sleep (see specific descr. for Sleep function)	None	1
WDR	Watchdog Reset (see specific descr. for WDR/timer)	None	1