



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Active
Core Processor	AVR
Core Size	8-Bit
Speed	16MHz
Connectivity	EBI/EMI, I ² C, SPI, UART/USART, USB OTG
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	48
Program Memory Size	64KB (64K x 8)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-VFQFN Exposed Pad
Supplier Device Package	64-QFN (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at90usb647-mu

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

- Real time counter with separate oscillator
- Four 8-bit PWM channels
- Six PWM channels with programmable resolution from 2 to 16 bits
- Output compare modulator
- 8-channels, 10-bit ADC
- Programmable serial USART
- Master/slave SPI serial interface
- Byte oriented 2-wire serial interface
- Programmable watchdog timer with separate on-chip oscillator
- On-chip analog comparator
- Interrupt and wake-up on pin change
- Special microcontroller features
 - Power-on reset and programmable brown-out detection
 - Internal calibrated oscillator
 - External and internal interrupt sources
 - Six sleep modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and packages
 - 48 programmable I/O lines
 - 64-lead TQFP and 64-lead QFN
- Operating voltages
 - 2.7 5.5V
- Operating temperature
- Industrial (-40°C to +85°C)
- Maximum frequency
 - 8MHz at 2.7V industrial range
 - 16MHz at 4.5V industrial range



Note: The large center pad underneath the MLF packages is made of metal and internally connected to GND. It should be soldered or glued to the board to ensure good mechanical stability. If the center pad is left unconnected, the package might loosen from the board.

2. Overview

The Atmel® AVR® AT90USB64/128 is a low-power CMOS 8-bit microcontroller based on the Atmel® AVR® enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the AT90USB64/128 achieves throughputs approaching 1MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block diagram

Figure 2-1. Block diagram.



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting

architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The Atmel AT90USB64/128 provides the following features: 64/128Kbytes of In-System Programmable Flash with Read-While-Write capabilities, 2K/4Kbytes EEPROM, 4K/8K bytes SRAM, 48 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), four flexible Timer/Counters with compare modes and PWM, one USART, a byte oriented 2-wire Serial Interface, a 8-channels, 10-bit ADC with optional differential input stage with programmable gain, programmable Watchdog Timer with Internal Oscillator, an SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using the Atmel high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the AT90USB64/128 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The AT90USB64/128 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

2.2.18	XTAL2	
		Output from the inverting oscillator amplifier.
2.2.19	AVCC	
		AVCC is the supply voltage pin for Port F and the A/D Converter. It should be externally connected to V_{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.
2.2.20	AREF	
		This is the analog reference pin for the A/D Converter.

3. Resources

A comprehensive set of development tools, application notes and datasheets are available for download on http://www.atmel.com/avr.

4. About code examples

This documentation contains simple code examples that briefly show how to use various parts of the device. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

These code examples assume that the part specific header file is included before compilation. For I/O registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0xBE)	Reserved	-	-	-	-	-	-	-	-	
(0xBD)	TWAMR	TWAM6	TWAM5	TWAM4	TWAM3	TWAM2	TWAM1	TWAM0	-	
(0xBC)	TWCR	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	-	TWIE	
(0xBB)	TWDR		1	T	2-wire Serial Inte	erface Data Regis	ter		1	
(0xBA)	TWAR	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	
(0xB9)	TWSR	TWS7	TWS6	TWS5	TWS4	TWS3	-	TWPS1	TWPS0	
(0xB8)	I WBR Bosonrod	_	_	2	-wire Serial Intern	ace Bit Hate Heg	Ister	_	_	
(0xB7)	ASSB	-	EXCLK	AS2	TCN2UB	OCB2AUB	OCB2BUB	TCB2AUB	TCB2BUB	
(0xB5)	Reserved	-	-	-	-	-	-	-	-	
(0xB4)	OCR2B			Tim	ner/Counter2 Out	put Compare Reg	jister B			
(0xB3)	OCR2A			Tin	ner/Counter2 Out	put Compare Reg	jister A			
(0xB2)	TCNT2		-	-	Timer/Co	unter2 (8 Bit)	_	_		
(0xB1)	TCCR2B	FOC2A	FOC2B	-	-	WGM22	CS22	CS21	CS20	
(0xB0)	TCCR2A	COM2A1	COM2A0	COM2B1	COM2B0	-	-	WGM21	WGM20	
(0xAF)	UPDATX				PD	AT7:0	TYOUTE	DYOTALLE	DVINE	
(UXAE)		FLERRE	NAKEDE	-			TXOUTE	RXSTALLE	RAINE	
(0xAD)	LIPSTAX	CEGOK	OVEBEI				EQ1:0	NBUS	YBK1.0	
(0xAB)	UPCFG1X	ordon	OVENIT	PSIZE2:0		PB	K1:0	ALLOC		
(0xAA)	UPCFG0X	PTY	PE1:0	PTOK	EN1:0		PEP	NUM3:0		
(0xA9)	UPCONX		PFREEZE	INMODE		RSTDT			PEN	
(0xA8)	UPRST			•		PRST6:0				
(0xA7)	UPNUM							PNUM2:0		
(0xA6)	UPINTX	FIFOCON	NAKEDI	RWAL	PERRI	TXSTPI	TXOUTI	RXSTALLI	RXINI	
(0xA5)	UPINRQX				INF	RQ7:0				
(0xA4)	UHFLEN				FL	EN7:0	1	ENIL IN 110-0		
(0xA3) (0xA2)					EN	LIM7:0		FINUMI10:8		
(0xA1)	UHADDR					HADD6:0				
(0xA0)	UHIEN		HWUPE	HSOFE	RXRSME	RSMEDE	RSTE	DDISCE	DCONNE	
(0x9F)	UHINT		HWUPI	HSOFI	RXRSMI	RSMEDI	RSTI	DDISCI	DCONNI	
(0x9E)	UHCON						RESUME	RESET	SOFEN	
(0x9D)	OCR3CH			Timer/Co	unter3 - Output C	compare Register	C High Byte			
(0x9C)	OCR3CL			Timer/Co	unter3 - Output C	Compare Register	C Low Byte			
(0x9B)	OCR3BH			Timer/Co	unter3 - Output C	compare Register	B High Byte			
(0x9A)	OCR3BL			Timer/Co	unter3 - Output C	Compare Register	A High Byte			
(0x98)	OCB3AI			Timer/Co	unter3 - Output C	Compare Register	A Low Byte			
(0x97)	ICR3H			Timer/	Counter3 - Input (Capture Register	High Byte			
(0x96)	ICR3L		Timer/Counter3 - Input Capture Register Fight Byte							
(0x95)	TCNT3H			Tim	er/Counter3 - Cou	unter Register Hig	jh Byte			
(0x94)	TCNT3L		-	Tim	er/Counter3 - Co	unter Register Lo	w Byte	_		
(0x93)	Reserved	-	-	-	-	-	-	-	-	
(0x92)	TCCR3C	FOC3A	FOC3B	FOC3C	-	-	-	-	-	
(0x91)	TCCR3B	ICNC3	ICES3	- COM2R1	WGM33	WGM32	CS32 COM2C0	CS31	CS30	
(0x90) (0x8E)	Beserved	CONSAT	CONISAU	CONISBI	CONISBO	CONISCI	CONISCO	WGIVI31	VVGIVI30	
(0x8E)	Reserved	-	-	-	-	-	-	-	-	
(0x8D)	OCR1CH		Timer/Counter1 - Output Compare Register C High Ryte							
(0x8C)	OCR1CL		Timer/Counter1 - Output Compare Register C Low Byte							
(0x8B)	OCR1BH		Timer/Counter1 - Output Compare Register B High Byte							
(0x8A)	OCR1BL		Timer/Counter1 - Output Compare Register B Low Byte							
(0x89)	OCR1AH	Timer/Counter1 - Output Compare Register A High Byte								
(0x88)	OCR1AL	Timer/Counter1 - Output Compare Register A Low Byte								
(0x87)	ICR1H	I imer/Counter1 - Input Capture Register High Byte								
(UX86) (0x85)		i imer/Counter1 - Input Capture Register Low Byte								
(0x84)	TCNT1L		Timer/Counter1 - Counter Register Right Byte							
(0x83)	Reserved	-	-	-	-	-	-	-	-	
(0x82)	TCCR1C	FOC1A	FOC1B	FOC1C	-	-	-	-	-	
(0x81)	TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	
(0x80)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	COM1C1	COM1C0	WGM11	WGM10	
(0x7F)	DIDR1	-	-	-	-	-	-	AIN1D	AIN0D	
(0x7E)	DIDR0	ADC7D	ADC6D	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	
(0x7D)	-	-	-	-	-	-	-	-	-	

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0x7C)	ADMUX	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	
(0x7B)	ADCSRB	ADHSM	ACME	-	-	-	ADTS2	ADTS1	ADTS0	
(0x7A)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	
(0x79)	ADCH				ADC Data Re	egister High byte				
(0x78)	ADCL			-	ADC Data R	egister Low byte	-		-	
(0x77)	Reserved	-	-	-	-	-	-	-	-	
(0x76)	Reserved	-	-	-	-	-	-	-	-	
(0x75)	XMCRB	XMBK	-	-	-	-	XMM2	XMM1	XMM0	
(0x74)	XMCRA	SRE	SRL2	SRL1	SRL0	SRW11	SRW10	SRW01	SRW00	
(0x73)	Reserved	-	-	-	-	-	-	-	-	
(0x72)	Reserved	-	-	-	-	-			-	
(0x71)	TIMSK3	-	-	ICIE3	-	OCIE3C	OCIE3B	OCIESA	TOIE3	
(0x70)	TIMSK2	-	-		-		OCIE2B		TOIE2	4
(0x6E)	TIMSKI			ICILI		OCILIC			TOIET	
	Beserved						OCILOB	OCILOA	TOILU	
(0x6C)	Reserved	-	-	-	-		-		-	
(0x6B)	PCMSK0	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	
(0x6A)	EICRB	ISC71	ISC70	ISC61	ISC60	ISC51	ISC50	ISC41	ISC40	
(0x69)	EICRA	ISC31	ISC30	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00	
(0x68)	PCICR	-	-	-	-	-	-	-	PCIE0	
(0x67)	Reserved	-	-	-	-	-	-	-	-	
(0x66)	OSCCAL				Oscillator Cal	ibration Register	•		•	
(0x65)	PRR1	PRUSB	-	-	-	PRTIM3	-	-	PRUSART1	
(0x64)	PRR0	PRTWI	PRTIM2	PRTIM0	-	PRTIM1	PRSPI	-	PRADC	
(0x63)	Reserved	-	-	-	-	-	-	-	-	
(0x62)	Reserved	-	-	-	-	-	-	-	-	
(0x61)	CLKPR	CLKPCE	-	-	-	CLKPS3	CLKPS2	CLKPS1	CLKPS0	
(0x60)	WDTCSR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	
0x3F (0x5F)	SREG	I	Т	н	S	V	N	Z	С	
0x3E (0x5E)	SPH	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	
0x3D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	
0x3C (0x5C)	Reserved	-	-	-	-	-	-	-	-	
0x3B (0x5B)	RAMPZ	-	-	-	-	-	-	RAMPZ1	RAMPZ0	
0x3A (0x5A)	Reserved	-	-	-	-	-	-	-	-	
0x39 (0x59)	Reserved	-	-	-	-	-	-	-	-	
0x38 (0x58)	SPMCSP	- SDMIE	- D\\/\\/SB		- D\\/\\/SDE	- BI BSET	- PGW/PT	- PCEPS	- SDMEN	
0x37 (0x57)	Beconvod	SF WILL	HWW3B	SIGILD	nww.sni	DEDGET	FOWIT	FGENS	3F WILIN	
0x35 (0x55)	MCUCB	- ITD			PUD			IVSEI	IVCE	
0x34 (0x54)	MCUSB	-	-	-	JTBE	WDBE	BORE	EXTRE	POBE	
0x33 (0x53)	SMCB	-	-	-	-	SM2	SM1	SMO	SE	
0x32 (0x52)	Reserved	-	-	-	-	-	-	-	-	
	OCDR/	OCDR7	OCDR6	OCDR5	OCDR4	OCDR3	OCDR2	OCDR1	OCDR0	
0x31 (0x51)	MONDR		1		Monitor D	Data Register				
0x30 (0x50)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	
0x2F (0x4F)	Reserved	-	-	-	-	-	-	-	-	
0x2E (0x4E)	SPDR				SPI Da	ta Register				
0x2D (0x4D)	SPSR	SPIF	WCOL	-	-	-	-	-	SPI2X	
0x2C (0x4C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	
0x2B (0x4B)	GPIOR2				General Purpo	ose I/O Register 2				
0x2A (0x4A)	GPIOR1				General Purpo	ose I/O Register 1				
0x29 (0x49)	PLLCSR	-	-	-	PLLP2	PLLP1	PLLP0	PLLE	PLOCK	
0x28 (0x48)	OCR0B			Tim	ner/Counter0 Out	put Compare Reg	ister B			
0x27 (0x47)	OCR0A			Tin	ner/Counter0 Out	put Compare Reg	ister A			
0x26 (0x46)	TCNT0				Timer/Co	unter0 (8 Bit)				
0x25 (0x45)	TCCR0B	FOCOA	FOCOB	-	-	WGM02	CS02	CS01	CS00	
0x24 (0x44)	ICCR0A	COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00	
0x23 (0x43)	GICCR	ISM	-	-	-	-	-	PSRASY	PSRSYNC	
0x22 (0x42)	EEARH	-	-	-		Pagiotar Law D	LEPHOM Addres	s Register High B	yte	
0x21 (0x41)	EEAKL					os riegisier LOW B	yıe			
0x1E (0x2E)	FECP		_	EEDM1	EEPMO		FEMDE	FEDE	FEDE	+
0x1F (0x3F)	GPIORO	-			General Purpo	se I/O Begister 0				1
0x1D (0x3D)	EIMSK	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INTO	
0x1C (0x3C)	EIFR	INTF7	INTF6	INTE5	INTF4	INTF3	INTF2	INTF1	INTFO	1
								· · · · · · · · · · · · · · · · · · ·		

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x1B (0x3B)	PCIFR	-	-	-	-	-	-	-	PCIF0	
0x1A (0x3A)	Reserved	-	-	-	-	-	-	-	-	
0x19 (0x39)	Reserved	-	-	-	-	-	-	-	-	
0x18 (0x38)	TIFR3	-	-	ICF3	-	OCF3C	OCF3B	OCF3A	TOV3	
0x17 (0x37)	TIFR2	-	-	-	-	-	OCF2B	OCF2A	TOV2	
0x16 (0x36)	TIFR1	-	-	ICF1	-	OCF1C	OCF1B	OCF1A	TOV1	
0x15 (0x35)	TIFR0	-	-	-	-	-	OCF0B	OCF0A	TOV0	
0x14 (0x34)	Reserved	-	-	-	-	-	-	-	-	
0x13 (0x33)	Reserved	-	-	-	-	-	-	-	-	
0x12 (0x32)	Reserved	-	-	-	-	-	-	-	-	
0x11 (0x31)	PORTF	PORTF7	PORTF6	PORTF5	PORTF4	PORTF3	PORTF2	PORTF1	PORTF0	
0x10 (0x30)	DDRF	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	
0x0F (0x2F)	PINF	PINF7	PINF6	PINF5	PINF4	PINF3	PINF2	PINF1	PINF0	
0x0E (0x2E)	PORTE	PORTE7	PORTE6	PORTE5	PORTE4	PORTE3	PORTE2	PORTE1	PORTE0	
0x0D (0x2D)	DDRE	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	
0x0C (0x2C)	PINE	PINE7	PINE6	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	
0x0B (0x2B)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	
0x0A (0x2A)	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	
0x09 (0x29)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	
0x08 (0x28)	PORTC	PORTC7	PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0	
0x07 (0x27)	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	
0x06 (0x26)	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	
0x05 (0x25)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	
0x04 (0x24)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	
0x03 (0x23)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	
0x02 (0x22)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	
0x01 (0x21)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	
0x00 (0x20)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	

Note: 1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

2. I/O registers within the address range \$00 - \$1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.

3. Some of the status flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

4. When using the I/O specific commands IN and OUT, the I/O addresses \$00 - \$3F must be used. When addressing I/O registers as data space using LD and ST instructions, \$20 must be added to these addresses. The Atmel AT90USB64/128 is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from \$60 - \$1FF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

6. Instruction set summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
	ARITHME	TIC AND LOGIC INSTRUCTIONS			
ADD	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	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	$Rd \leftarrow Rd \bullet Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \bullet K$	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	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Complement	Rd ← 0xFF - Rd	Z,C,N,V	1
NEG	Rd	I wo's Complement		Z,C,N,V,H	1
SBR	Ra,K	Clear Bit(s) in Register		Z,N,V	1
UBR	Ru,r		$Rd \leftarrow Rd \bullet (0XFF - K)$	Z,IN,V	1
	Ra	Increment		Z,N,V	1
DEC	Ru	Toot for Zoro or Minus	$Ru \leftarrow Ru - I$	Z,IN, V	1
CLP	Rd	Clear Bogister		Z,N,V	1
SEB	Bd	Set Bogister		None	1
MUI	Bd Br	Multiply Linsigned	$B1:B0 \leftarrow Bd \times Br$	ZC	2
MULS	Bd Br	Multiply Signed	$B1:B0 \leftarrow Bd \times Br$	Z,0	2
MULSU	Bd Br	Multiply Signed	$B1:B0 \leftarrow Bd \times Br$	Z,0	2
FMUI	Bd Br	Fractional Multiply Unsigned	$B1:B0 \leftarrow (Bd \times Br) << 1$	Z,0	2
FMULS	Bd Br	Fractional Multiply Signed	$B1:B0 \leftarrow (Bd \times Br) << 1$	Z,0	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	$B1:B0 \leftarrow (Bd \times Br) << 1$	Z.C	2
	BI	RANCH INSTRUCTIONS	•		
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
IJMP		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
EIJMP		Extended Indirect Jump to (Z)	PC ←(EIND:Z)	None	2
JMP	k	Direct Jump	$PC \leftarrow k$	None	3
RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	4
ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	4
EICALL		Extended Indirect Call to (Z)	$PC \leftarrow (EIND:Z)$	None	4
CALL	k	Direct Subroutine Call	$PC \leftarrow k$	None	5
RET		Subroutine Return	$PC \leftarrow STACK$	None	5
RETI		Interrupt Return	$PC \leftarrow STACK$	1	5
CPSE	Rd,Rr	Compare, Skip if Equal	if $(Rd = Rr) PC \leftarrow PC + 2 \text{ or } 3$	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 $(\operatorname{Rr}(b)=0)$ PC \leftarrow PC + 2 or 3	None	1/2/3
SBRS	Rr, b	Skip if Bit in Register is Set	if $(\text{Rr}(b)=1) \text{ PC} \leftarrow \text{PC} + 2 \text{ or } 3$	None	1/2/3
SBIC	P,D	Skip II BIT IN I/O Register Cleared	ii $(P(b)=0) PO \leftarrow PO + 2 \text{ or } 3$	None	1/2/3
BDBC	г, D	Branch if Status Elas Sat	if $(PEC(c) = 1)$ then $PC \in PC + 2$ of 3 if $(PEC(c) = 1)$ then $PC \in PC + 1$	None	1/2/3
BPBC	5, K	Branch if Status Flag Oleared	if $(SREG(s) = 1)$ then $PC \leftarrow PC + K + 1$ if $(SREG(s) = 0)$ then $PC \leftarrow PC + k + 1$	None	1/2
BREO	k s, n	Branch if Foual	if $(7 = 1)$ then PC \leftarrow PC + k + 1	None	1/2
BRNF	k	Branch if Not Foual	if $(Z = 0)$ then PC \leftarrow PC + k + 1	None	1/2
BRCS	k	Branch if Carry Set	if $(C = 1)$ then PC \leftarrow PC + k + 1	None	1/2
BRCC	 k	Branch if Carry Cleared	if (C = 0) then PC \leftarrow PC + k + 1	None	1/2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC \leftarrow PC + k + 1	None	1/2
BRLO	k	Branch if Lower	if (C = 1) then PC \leftarrow PC + k + 1	None	1/2
BRMI	k	Branch if Minus	if (N = 1) then PC \leftarrow PC + k + 1	None	1/2
BRPL	k	Branch if Plus	if (N = 0) then PC \leftarrow PC + k + 1	None	1/2
BRGE	k	Branch if Greater or Equal, Signed	if (N \oplus V= 0) then PC \leftarrow PC + k + 1	None	1/2
BRLT	k	Branch if Less Than Zero, Signed	if (N \oplus V= 1) then PC \leftarrow PC + k + 1	None	1/2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC \leftarrow PC + k + 1	None	1/2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC \leftarrow PC + k + 1	None	1/2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC \leftarrow PC + k + 1	None	1/2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC \leftarrow PC + k + 1	None	1/2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC \leftarrow PC + k + 1	None	1/2

Mnemonics	Operands	Description Operation			#Clocks
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC \leftarrow PC + k + 1	None	1/2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC \leftarrow PC + k + 1	None	1/2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC \leftarrow PC + k + 1	None	1/2
	BIT AN	ID BIT-TEST INSTRUCTIONS	1	1	-
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	$I/O(P,b) \leftarrow 0$	None	2
LSL	Rd	Logical Shift Left	$Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0$	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Logical Shift Right $Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0$		1
ROL	Rd	Rotate Left Through Carry	btate Left Through Carry $Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7)$		1
RUR	Rd	Avithmetic Chift Dight	Arithmetia Chift Dickt		1
SWAP	Rd	Swan Nibbles	$Rd(3, 0) \leftarrow Rd(7, 4) Rd(7, 4) \leftarrow Rd(3, 0)$	Z,C,N,V	1
BSET	s	Flag Set	$SBEG(s) \leftarrow 1$	SBEG(s)	1
BCLR	s	Flag Clear	$SBEG(s) \leftarrow 0$	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	$T \leftarrow Rr(b)$	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	С	1
CLC		Clear Carry	C ← 0	С	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	← 1	1	1
CLI		Global Interrupt Disable		1	1
SES		Clear Signed Test Flag	S ← I	S	1
SEV		Set Twee Complement Overflow	S ← 0	5 V	1
CLV		Clear Twos Complement Overflow		V	1
SET		Set T in SBEG	T ← 1	т	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	н	1
CLH		Clear Half Carry Flag in SREG	H ← 0	Н	1
	DATA	TRANSFER INSTRUCTIONS		-	
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	$Rd+1:Rd \leftarrow Rr+1:Rr$	None	1
LDI	Rd, K	Load Immediate	$Rd \leftarrow K$	None	1
LD	Rd, X	Load Indirect	$Rd \leftarrow (X)$	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	$Rd \leftarrow (X), X \leftarrow X + 1$	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	$X \leftarrow X - 1, Rd \leftarrow (X)$	None	2
LD	Ra, Y	Load Indirect	$Hd \leftarrow (Y)$	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	$Ra \leftarrow (Y), Y \leftarrow Y + I$	None	2
	Rd Y+a	Load Indirect with Displacement	$Bd \leftarrow (Y + q)$	None	2
	Bd Z	Load Indirect	$Bd \leftarrow (Z)$	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	$Rd \leftarrow (Z), Z \leftarrow Z+1$	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	$Z \leftarrow Z - 1, Rd \leftarrow (Z)$	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	$Rd \leftarrow (Z + q)$	None	2
LDS	Rd, k	Load Direct from SRAM	$Rd \leftarrow (k)$	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	$(X) \leftarrow Rr, X \leftarrow X + 1$	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	$X \leftarrow X - 1, (X) \leftarrow Rr$	None	2
ST	Y, Rr	Store Indirect	$(Y) \leftarrow Rr$	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	$(Y) \leftarrow Rr, Y \leftarrow Y + 1$	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	$Y \leftarrow Y - 1, (Y) \leftarrow Rr$	None	2
510	r+q,Kr	Store Indirect with Displacement	$(Y + q) \leftarrow Hr$	None	2
<u>२</u> २	∠, Hr 7± Pr	Store Indirect and Poet-Inc	$(2) \leftarrow \text{Rr}$	None	2
ST	-7 Rr	Store Indirect and Pre-Dec	$7 \leftarrow 7 - 1 (7) \leftarrow Rr$	None	2
STD	7+a Br	Store Indirect with Displacement	$(7 + \alpha) \leftarrow Br$	None	2
STS	k. Rr	Store Direct to SRAM	$(k) \leftarrow \operatorname{Rr}$	None	2
LPM	,	Load Program Memory	$R0 \leftarrow (Z)$	None	3
LPM	Rd, Z	Load Program Memory	$Rd \leftarrow (Z)$	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	$Rd \leftarrow (Z), Z \leftarrow Z+1$	None	3
ELPM		Extended Load Program Memory	$R0 \leftarrow (RAMPZ:Z)$	None	3
ELPM	Rd, Z	Extended Load Program Memory	$Rd \leftarrow (Z)$	None	3
FI PM	Bd 7+	Extended Load Program Memory	$Bd \leftarrow (BAMP7'7) BAMP7'7 \leftarrow BAMP7'7 + 1$	None	3

Mnemonics	Operands	Description	Description Operation		#Clocks
SPM		Store Program Memory $(Z) \leftarrow R1:R0$		None	-
IN	Rd, P	In Port Rd ← P		None	1
OUT	P, Rr	Out Port $P \leftarrow Rr$		None	1
PUSH	Rr	Push Register on Stack	$STACK \leftarrow Rr$	None	2
POP	Rd	Pop Register from Stack	$Rd \gets STACK$	None	2
	MCU	CONTROL INSTRUCTIONS			
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
BREAK		Break	For On-chip Debug Only	None	N/A

8. Packaging information

8.1 TQFP64





not set the SUSPI bit anymore. The internal USB engine remains in suspend mode but the USB differential receiver is still enabled and generates a typical 300µA extra-power consumption. Detection of the suspend state after the transient perturbation should be performed by software (instead of reading the SUSPI bit).

Problem fix/workaround

USB waiver allows bus powered devices to consume up to 2.5mA in suspend state.

6. VBUS session valid threshold voltage

The VSession valid threshold voltage is internally connected to VBus_Valid (4.4V approx.). That causes the device to attach to the bus only when Vbus is greater than VBusValid instead of V_Session Valid. Thus if VBUS is lower than 4.4V, the device is detached.

Problem fix/workaround

According to the USB power drop budget, this may require connecting the device toa root hub or a self-powered hub.

5. UBS signal rate

The average USB signal rate may sometime be measured out of the USB specifications $(12MHz \pm 30kHz)$ with short frames. When measured on a long period, the average signal rate value complies with the specifications. This bit rate deviation does not generates communication or functional errors.

Problem fix/workaround

None.

4. VBUS residual level

In USB device and host mode, once a 5V level has been detected to the VBUS pad, a residual level (about 3V) can be measured on the VBUS pin.

Problem fix/workaround

None.

3. Spike on TWI pins when TWI is enabled

100ns negative spike occurs on SDA and SCL pins when TWI is enabled.

Problem fix/workaround

No known workaround, enable Atmel AT90USB64/128 TWI first versus the others nodes of the TWI network.

2. High current consumption in sleep mode

If a pending interrupt cannot wake the part up from the selected mode, the current consumption will increase during sleep when executing the SLEEP instruction directly after a SEI instruction.

Problem fix/workaround

Before entering sleep, interrupts not used to wake up the part from the sleep mode should be disabled.

9.1.3 Atmel AT90USB1287/6 second release

- Incorrect CPU behavior for VBUSTI and IDTI interrupts routines
- USB Eye Diagram violation in low-speed mode
- Transient perturbation in USB suspend mode generates over consumption
- VBUS Session valid threshold voltage
- Spike on TWI pins when TWI is enabled
- · High current consumption in sleep mode
- Async timer interrupt wake up from sleep generate multiple interrupts

7. Incorrect CPU behavior for VBUSTI and IDTI interrupts routines

The CPU core may incorrectly execute the interrupt vector related to the VBUSTI and IDTI interrupt flags.

Problem fix/workaround

Do not enable these interrupts, firmware must process these USB events by polling VBUSTI and IDTI flags.

6. USB Eye Diagram violation in low-speed mode

The low to high transition of D- violates the USB eye diagram specification when transmitting with low-speed signaling.

Problem fix/workaround

None.

5. Transient perturbation in USB suspend mode generates overconsumption

In device mode and when the USB is suspended, transient perturbation received on the USB lines generates a wake up state. However the idle state following the perturbation does not set the SUSPI bit anymore. The internal USB engine remains in suspend mode but the USB differential receiver is still enabled and generates a typical 300µA extra-power consumption. Detection of the suspend state after the transient perturbation should be performed by software (instead of reading the SUSPI bit).

Problem fix/workaround

USB waiver allows bus powered devices to consume up to 2.5mA in suspend state.

4. VBUS session valid threshold voltage

The VSession valid threshold voltage is internally connected to VBus_Valid (4.4V approx.). That causes the device to attach to the bus only when Vbus is greater than VBusValid instead of V_Session Valid. Thus if VBUS is lower than 4.4V, the device is detached.

Problem fix/workaround

According to the USB power drop budget, this may require connecting the device toa root hub or a self-powered hub.

3. Spike on TWI pins when TWI is enabled

100ns negative spike occurs on SDA and SCL pins when TWI is enabled.

Problem fix/workaround

No known workaround, enable Atmel AT90USB64/128 TWI first versus the others nodes of the TWI network.

2. High current consumption in sleep mode

If a pending interrupt cannot wake the part up from the selected mode, the current consumption will increase during sleep when executing the SLEEP instruction directly after a SEI instruction.

Problem fix/workaround

Before entering sleep, interrupts not used to wake up the part from the sleep mode should be disabled.

1. Asynchronous timer interrupt wake up from sleep generates multiple interrupts

If the CPU core is in sleep and wakes-up from an asynchronous timer interrupt and then go back in sleep again it may wake up multiple times.

Problem fix/workaround

A software workaround is to wait with performing the sleep instruction until TCNT2>OCR2+1.

9.1.4 Atmel AT90USB1287/6 Third Release

- Incorrect CPU behavior for VBUSTI and IDTI interrupts routines
- Transient perturbation in USB suspend mode generates over consumption
- Spike on TWI pins when TWI is enabled
- High current consumption in sleep mode
- · Async timer interrupt wake up from sleep generate multiple interrupts

5. Incorrect CPU behavior for VBUSTI and IDTI interrupts routines

The CPU core may incorrectly execute the interrupt vector related to the VBUSTI and IDTI interrupt flags.

Problem fix/workaround

Do not enable these interrupts, firmware must process these USB events by polling VBUSTI and IDTI flags.

4. Transient perturbation in USB suspend mode generates overconsumption

In device mode and when the USB is suspended, transient perturbation received on the USB lines generates a wake up state. However the idle state following the perturbation does not set the SUSPI bit. The internal USB engine remains in suspend mode but the USB differential receiver is still enabled and generates a typical 300µA extra-power consumption. Detection of the suspend state after the transient perturbation should be performed by software (instead of reading the SUSPI bit).

Problem fix/workaround

USB waiver allows bus powered devices to consume up to 2.5mA in suspend state.

3. Spike on TWI pins when TWI is enabled

100ns negative spike occurs on SDA and SCL pins when TWI is enabled.

Problem fix/workaround

No known workaround, enable AT90USB64/128 TWI first, before the others nodes of the TWI network.

2. High current consumption in sleep mode

If a pending interrupt cannot wake the part up from the selected mode, the current consumption will increase during sleep when executing the SLEEP instruction directly after a SEI instruction.

Problem fix/workaround

Before entering sleep, interrupts not used to wake up the part from sleep mode should be disabled.

Asynchronous timer interrupt wake up from sleep generates multiple interrupts
 If the CPU core is in sleep mode and wakes-up from an asynchronous timer interrupt and
 then goes back into sleep mode, it may wake up multiple times.

9.2 Atmel AT90USB646/7 errata

9.2.1 AT90USB646/7 errata history TBD

Silicon Release	90USB646-16MU	90USB647-16AU	90USB647-16MU
First Release			
Second Release			

Note '*' means a blank or any alphanumeric string.

9.2.2 AT90USB646/7 first release.

- Incorrect interrupt routine execution for VBUSTI, IDTI interrupts flags
- USB Eye Diagram violation in low-speed mode
- Transient perturbation in USB suspend mode generates over consumption
- Spike on TWI pins when TWI is enabled
- High current consumption in sleep mode
- · Async timer interrupt wake up from sleep generate multiple interrupts

6. Incorrect CPU behavior for VBUSTI and IDTI interrupts routines

The CPU core may incorrectly execute the interrupt vector related to the VBUSTI and IDTI interrupt flags.

Problem fix/workaround

Do not enable these interrupts, firmware must process these USB events by polling VBUSTI and IDTI flags.

5. USB Eye Diagram violation in low-speed mode

The low to high transition of D- violates the USB eye diagram specification when transmitting with low-speed signaling.

Problem fix/workaround

None.

4. Transient perturbation in USB suspend mode generates overconsumption

In device mode and when the USB is suspended, transient perturbation received on the USB lines generates a wake up state. However the idle state following the perturbation does not set the SUSPI bit anymore. The internal USB engine remains in suspend mode but the USB differential receiver is still enabled and generates a typical 300µA extra-power consumption. Detection of the suspend state after the transient perturbation should be performed by software (instead of reading the SUSPI bit).

Problem fix/workaround

USB waiver allows bus powered devices to consume up to 2.5mA in suspend state.

3. Spike on TWI pins when TWI is enabled

100ns negative spike occurs on SDA and SCL pins when TWI is enabled.

10. Datasheet revision history for Atmel AT90USB64/128

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

10.1 Changes from 7593A to 7593B

- 1. Changed default configuration for fuse bytes and security byte.
- 2. Suppression of timer 4,5 registers which does not exist.
- 3. Updated typical application schematics in USB section

10.2 Changes from 7593B to 7593C

1. Update to package drawings, MQFP64 and TQFP64.

10.3 Changes from 7593C to 7593D

 For further product compatibility, changed USB PLL possible prescaler configurations. Only 8MHz and 16MHz crystal frequencies allows USB operation (see Table 7-11 on page 50).

10.4 Changes from 7593D to 7593E

- 1. Updated PLL Prescaler table: configuration words are different between AT90USB64x and AT90USB128x to enable the PLL with a 16MHz source.
- 2. Cleaned up some bits from USB registers, and updated information about OTG timers, remote wake-up, reset and connection timings.
- 3. Updated clock distribution tree diagram (USB prescaler source and configuration register).
- 4. Cleaned up register summary.
- 5. Suppressed PCINT23:8 that do not exist from External Interrupts.
- 6. Updated Electrical Characteristics.
- 7. Added Typical Characteristics.
- 8. Update Errata section.

10.5 Changes from 7593E to 7593F

- 1. Removed 'Preliminary' from document status.
- 2. Clarification in Stand by mode regarding USB.

10.6 Changes from 7593F to 7593G

1. Updated Errata section.

10.7 Changes from 7593G to 7593H

- 1. Added Signature information for 64K devices.
- 2. Fixed figure for typical bus powered application
- 3. Added min/max values for BOD levels
- 4. Added ATmega32U6 product
- 5. Update Errata section
- 6. Modified descriptions for HWUPE and WAKEUPE interrupts enable (these interrupts should be enabled only to wake up the CPU core from power down mode).