ATMEL® CORPORATION

AVR® Microcontrollers: Product Line Reference

January 2006 Customer Edition

Table of Contents

1 AVR Product Family

1.1 Product Selection Guide - tinyAVR®

Device	Status	Flash (Kbytes)	EEPROM (Kbytes)	SRAM (Bytes)	Max I/O Pins	F.max (MHz)	Vcc (V)	16-bit Timers	8-bit Timer	PWM (ch)	RTC	SPI	USART	TWI	ISP	10-bit A/D (ch.)	Analog Comparator	Brown Out Detector	Watchdog	On Chip Oscillator	Hardware Multiplier	Interrupts	Ext Interrupts	Self Program Memory	Packages	Green Packages
ATtiny11	N	1	--	--	6	6	2.7-5.5	--	1	--	--	--	--	--	--	--	Yes	--	Yes	Yes	--	4	1	--	PDIP8, SOIC8	PDIP8, SOIC8
ATtiny12	N	1	0.064	--	6	8	1.8-5.5	--	1	--	--	--	--	--	Yes	--	Yes	Yes	Yes	Yes	--	5	1	--	PDIP8, SOIC8	PDIP8, SOIC8
ATtiny13	P	1	0.064	64	6	20	1.8-5.5	--	1	2	--	--	--	--	Yes	4	Yes	Yes	Yes	Yes	--	9	6	Yes	PDIP8, SOIC8, SSOIC8, MLF20	PDIP8, SOIC8, SSOIC8, MLF20
ATtiny15L	N	1	0.064	--	6	1.6	2.7-5.5	--	2	1	--	--	--	--	Yes	4	Yes	Yes	Yes	Yes	--	8	1(+5)	--	PDIP8, SOIC8	
ATtiny25	P	2	0.128	128	6	20	1.8-5.5	--	2	4	--	USI	--	USI	Yes	4	Yes	Yes	Yes	Yes	--	15	7	Yes		PDIP8, SOIC8, MLF20
ATtiny45	P	4	0.256	256	6	20	1.8-5.5	--	2	4	--	USI	--	USI	Yes	4	Yes	Yes	Yes	Yes	--	15	7	Yes		PDIP8, SOIC8, MLF20
ATtiny85	I	8	0.512	512	6	20	1.8-5.5	--	2	4	--	USI	--	USI	Yes	4	Yes	Yes	Yes	Yes	--	15	7	Yes		PDIP8, SOIC8, MLF20
ATtiny28L	P	2	--	--	11	4	1.8-5.5	--	1	--	--	--	--	--	--	--	Yes	--	Yes	Yes	--	5	2(+8)	--	PDIP28, MLF32, TQFP32	PDIP28, MLF32, TQFP32
ATtiny24	P	2	0.128	128	12	20	1.8-5.5	1	1	4	--	USI	--	USI	Yes	8	Yes	Yes	Yes	Yes	--	17	12	Yes		PDIP14, SOIC14, MLF20
ATtiny44	P	4	0.256	256	12	20	1.8-5.5	1	1	4	--	USI	--	USI	Yes	8	Yes	Yes	Yes	Yes	--	17	12	Yes		PDIP14, SOIC14, MLF20
ATtiny84	I	8	0.512	512	12	20	1.8-5.5	1	1	4	--	USI	--	USI	Yes	8	Yes	Yes	Yes	Yes	--	17	12	Yes		PDIP14, MLF20
ATtiny26	P	2	0.128	128	16	16	2.7-5.5	--	2	2	--	USI	--	USI	Yes	11	Yes	Yes	Yes	Yes	--	11	1	--	PDIP20, SOIC20, MLF32	PDIP20, SOIC20, MLF32
ATtiny261	I	2	0.128	128	16	20	1.8-5.5	1	1	5	--	USI	--	USI	Yes	11	Yes	Yes	Yes	Yes	--	19	18	Yes		PDIP20, SOIC20, MLF32
ATtiny461	I	4	0.256	256	16	20	1.8-5.5	1	1	5	--	USI	--	USI	Yes	11	Yes	Yes	Yes	Yes	--	19	18	Yes		PDIP20, SOIC20, MLF32
ATtiny861	I	8	0.512	512	16	20	1.8-5.5	1	1	5	--	USI	--	USI	Yes	11	Yes	Yes	Yes	Yes	--	19	18	Yes		PDIP20, SOIC20, MLF32
ATtiny2313	P	2	0.128	128	18	20	1.8-5.5	1	1	4	--	USI	1	USI	Yes	--	Yes	Yes	Yes	Yes	--	8	2	Yes	PDIP20, SOIC20, MLF20	PDIP20, SOIC20, MLF20

Status: F - Future Design, I - Device under introduction, P - Device in production, N - not recommended for new designs

1.2 Product Selection Guide - megaAVR®

Device	Status	Flash (Kbytes)	EEPROM (Kbytes)	SRAM (Bytes)	Max I/O Pins	F.max (MHz)	Vcc (V)	16-bit Timers	8-bit Timer	PWM (ch)	RTC	SPI	USART	TWI	ISP	10-bit A/D (channels)	Analog Comparator	Brown Out Detector	Watchdog	On Chip Oscillator	Hardware Multiplier	Interrupts	Ext Interrupts	Self Program Memory	Packages	Green Packages
ATmega48	P	4	0.256	512	23	20	1.8-5.5	1	2	6	Yes	1+USART	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	26	26	Yes	PDIP28, MLF32, TQFP32	PDIP28, MLF32, TQFP32
ATmega8	P	8	0.512	1024	23	16	2.7-5.5	1	2	3	Yes	1	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	18	2	Yes	PDIP28, MLF32, TQFP32	PDIP28, MLF32, TQFP32
ATmega88	P	8	0.512	1024	23	20	1.8-5.5	1	2	6	Yes	1+USART	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	26	26	Yes	PDIP28, MLF32, TQFP32	PDIP28, MLF32, TQFP32
ATmega168	P	16	0.512	1024	23	20	1.8-5.5	1	2	6	Yes	1+USART	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	26	26	Yes	PDIP28, MLF32, TQFP32	PDIP28, MLF32, TQFP32
ATmega8535	P	8	0.512	512	32	16	2.7-5.5	1	2	4	--	1	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	20	3	Yes	PDIP40, MLF44, TQFP44, PLCC44	PDIP40, MLF44, TQFP44
ATmega16	P	16	0.512	1024	32	16	2.7-5.5	1	2	4	Yes	1	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	20	3	Yes	PDIP40, MLF44, TQFP44	PDIP40, MLF44, TQFP44
ATmega32	P	32	1	2048	32	16	2.7-5.5	1	2	4	Yes	1	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	19	3	Yes	PDIP40, MLF44, TQFP44	PDIP40, MLF44, TQFP44
ATmega644	P	64	2	4096	32	20	1.8-5.5	1	2	6	Yes	1+USART	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	31	32	Yes		PDIP40, MLF44, TQFP44
ATmega8515	P	8	0.512	512	35	16	2.7-5.5	1	1	3	--	1	1	--	Yes	--	--	Yes	Yes	Yes	Yes	16	3	Yes	PDIP40, MLF44, TQFP44, PLCC44	PDIP40, MLF44, TQFP44
ATmega162	P	16	0.512	1024	35	16	1.8-5.5	2	2	6	Yes	1	2	--	Yes	--	Yes	Yes	Yes	Yes	Yes	28	3	Yes	PDIP40, MLF44, TQFP44	PDIP40, MLF44, TQFP44
ATmega128	P	128	4	4096	53	16	2.7-5.5	2	2	8	Yes	1	2	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	34	8	Yes	MLF64, TQFP64	MLF64, TQFP64
ATmega165	N	16	0.512	1024	54	16	1.8-5.5	1	2	4	Yes	1+USI	1	USI	Yes	8	Yes	Yes	Yes	Yes	Yes	23	17	Yes	MLF64, TQFP64	MLF64, TQFP64
ATmega325	P	32	1	2048	54	16	1.8-5.5	1	2	4	Yes	1+USI	1	USI	Yes	8	Yes	Yes	Yes	Yes	Yes	23	17	Yes	MLF64, TQFP64	MLF64, TQFP64
ATmega64	P	64	2	4096	54	16	2.7-5.5	2	2	8	Yes	1	2	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	34	8	Yes	MLF64, TQFP64	MLF64, TQFP64
ATmega645	I	64	2	4096	54	16	1.8-5.5	1	2	4	Yes	1+USI	1	USI	Yes	8	Yes	Yes	Yes	Yes	Yes	23	17	Yes	MLF64, TQFP64	MLF64, TQFP64
ATmega1281	I	128	4	8192	54	16	1.8-5.5	4	2	10	Yes	1+USART	2	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	48	17	Yes		MLF64, TQFP64
ATmega2561	P	256	4	8192	54	16	1.8-5.5	4	2	10	Yes	1+USART	2	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	48	17	Yes		MLF64, TQFP64
ATmega3250	I	32	1	2048	69	16	1.8-5.5	1	2	4	Yes	1+USI	1	USI	Yes	8	Yes	Yes	Yes	Yes	Yes	32	17	Yes	TQFP100	TQFP100
ATmega6450	I	64	2	4096	69	16	1.8-5.5	1	2	4	Yes	1+USI	1	USI	Yes	8	Yes	Yes	Yes	Yes	Yes	32	17	Yes	TQFP100	TQFP100
ATmega640	I	64	4	8192	86	16	1.8-5.5	4	2	16	Yes	1+USART	4	Yes	Yes	16	Yes	Yes	Yes	Yes	Yes	57	32	Yes		TQFP100, BGA100
ATmega1280	I	128	4	8192	86	16	1.8-5.5	4	2	16	Yes	1+USART	4	Yes	Yes	16	Yes	Yes	Yes	Yes	Yes	57	32	Yes		TQFP100, BGA100
ATmega2560	P	256	4	8192	86	16	1.8-5.5	4	2	16	Yes	1+USART	4	Yes	Yes	16	Yes	Yes	Yes	Yes	Yes	57	32	Yes		TQFP100, BGA100

Status: F - Future Design, I - Device under introduction, P - Device in production, N - not recommended for new designs

1.3 Product Selection Guide – picoPower™ AVR

Device	Status	Flash (Kbytes)	EEPROM (Kbytes)	SRAM (Bytes)	Max I/O Pins	F.max (MHz)	Vcc (V)	Specific Features	16-bit Timers	8-bit Timer	PWM (ch)	RTC	SPI	USART	TWI	ISP	10-bit A/D (channels)	Analog Comparator	Brown Out Detector	Watchdog	On Chip Oscillator	Hardware Multiplier	Interrupts	Ext Interrupts	Self Program Memory	Green Packages
ATmega48P	F	4	0.256	512	23	16	1.8-5.5		1	2	6	Yes	1+USART	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	26	26	Yes	MLF32, TQFP32
ATmega88P	F	8	0.512	512	23	16	1.8-5.5		1	2	6	Yes	1+USART	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	26	26	Yes	MLF32, TQFP32
ATmega168P	F	16	1	1024	23	16	1.8-5.5		1	2	6	Yes	1+USART	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	26	26	Yes	MLF32, TQFP32
ATmega164P	I	16	0.512	1024	32	20	1.8-5.5		1	2	6	Yes	1+USART	2	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	31	32	Yes	PDIP40, MLF44, TQFP44
ATmega324P	P	32	1	2048	32	20	1.8-5.5		1	2	6	Yes	1+USART	2	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	31	32	Yes	PDIP40, MLF44, TQFP44
ATmega644P	I	64	2	4096	32	20	1.8-5.5		1	2	6	Yes	1+USART	2	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	31	32	Yes	PDIP40, MLF44, TQFP44
ATmega165P	I	16	0.512	1024	54	16	1.8-5.5		1	2	4	Yes	1+USI	1	USI	Yes	8	Yes	Yes	Yes	Yes	Yes	23	17	Yes	MLF64, TQFP64
ATmega325P	I	32	1	2048	54	16	1.8-5.5		1	2	4	Yes	1+USI	1	USI	Yes	8	Yes	Yes	Yes	Yes	Yes	23	17	Yes	MLF64, TQFP64
ATmega3250P	I	32	1	2048	69	16	1.8-5.5		1	2	4	Yes	1+USI	1	USI	Yes	8	Yes	Yes	Yes	Yes	Yes	32	17	Yes	TQFP100
ATmega169P	P	16	0.512	1024	54	16	1.8-5.5	LCD 4x25	1	2	4	Yes	1+USI	1	USI	Yes	8	Yes	Yes	Yes	Yes	Yes	23	17	Yes	MLF64, TQFP64
ATmega329P	I	32	1	2048	54	16	1.8-5.5	LCD 4x25	1	2	4	Yes	1+USI	1	USI	Yes	8	Yes	Yes	Yes	Yes	Yes	25	17	Yes	MLF64, TQFP64
ATmega3290P	I	32	1	2048	69	16	1.8-5.5	LCD 4x40	1	2	4	Yes	1+USI	1	USI	Yes	8	Yes	Yes	Yes	Yes	Yes	25	32	Yes	TQFP100

Status: F - Future Design, I - Device under introduction, P - Device in production, N - not recommended for new designs

1.4 Product Selection Guide – AVR32

Device	Status	SRAM (Kbytes)	DSP Instructions	Vector Multiplier Coprocessor	Ethernet MAC 10/100	USB Device	LCD Controller	UART USART DBGU	10-bit A/D (channels)	PWM (channels)	Max I/O Pins	Audio DAC (16-bit)	PDC (channels)	External Bus Interface	SDRAM Interface	16-bit Timers	RTC	SPI	AC97	Camera Interface	PS/2	SSC	TWI / I2C	MCI	Watchdog Timer	POR	BOD	On-chip RC Oscillator	Crystal Oscillator	PLL	CPU Core	ECCC	MMU/MPU	JTAG	Power Supply (V)	Green Packages
AT32AP7000	I	32	Yes	Yes	2	1 x HS	2048 x 2048 24-bit	4	-	4	160	Stereo	20	Yes	Yes	3	1	2	1	CMOS	Yes	3	1	1	Yes	Yes	-	-	2	2	AP	Yes	MMU	Yes	1.65-1.95 Core3.0-3.6 IO	256 CTBGA
AT32AP7001	F	32	Yes	Yes	0	1 x HS	-	4	-	4	90	Stereo	20	Yes	Yes	3	1	2	1	CMOS	Yes	3	1	1	Yes	Yes	-	-	2	2	AP	Yes	MMU	Yes	1.65-1.95 Core3.0-3.6 IO	208 VQFP
AT32AP7002	F	32	Yes	Yes	0	1 x HS	2048 x 2048 18-bit	4	-	4	85	Stereo	20	Yes	Yes	3	1	2	1	CMOS	Yes	3	1	1	Yes	Yes	-	-	2	2	AP	Yes	MMU	Yes	1.65-1.95 Core3.0-3.6 IO	196 CTBGA
AT32AP7003	F	32	Yes	Yes	2	1 x HS	-	4	-	4		-	20	Yes	Yes	3	1	2	-	-	Yes	3	1	1	Yes	Yes	-	-	2	2	AP	Yes	MMU	Yes	1.65-1.95 Core3.0-3.6 IO	
AT32AP7004	F	32	Yes	Yes	1	1 x HS	2048 x 2048	4	-	4		-	20	Yes	Yes	3	1	2	-	-	Yes	3	1	1	Yes	Yes	-	-	2	2	AP	Yes	MMU	Yes	1.65-1.95 Core3.0-3.6 IO	

Status: F - Future Design, I - Device under introduction, P - Device in production, N - not recommended for new designs

1.5 Product Selection Guide – megaAVR LCD and ASSP AVR

Device	Status	Flash (Kbytes)	EEPROM (Kbytes)	SRAM (Bytes)	Max I/O Pins	F.max (MHz)	Vcc (V)	Specific Features	16-bit Timers	8-bit Timer	PWM (ch)	RTC	SPI	USART	TWI	ISP	10-bit A/D (channels)	Analog Comparator	Brown Out Detector	Watchdog	On Chip Oscillator	Hardware Multiplier	Interrupts	Ext Interrupts	Self Program Memory	Packages	Green Packages
ATmega406	I	40	0.512	2048	18	1	4-25	Smart battery	1	1	1	Yes	--	--	Yes	Yes	--	Yes	Yes	Yes	Yes	Yes	23	4	Yes		LQFP48
AT90PWM2	P	8	0.512	512	19	16	2.7-5.5	PWM	1	1	7	Yes	1	Yes	--	Yes	8	Yes	Yes	Yes	Yes	Yes	--	4	Yes		SO24
AT90PWM3	P	8	0.512	512	27	16	2.7-5.5	PWM	1	1	10	Yes	1	Yes	--	Yes	11	Yes	Yes	Yes	Yes	Yes	--	4	Yes		SO32, MLF32
AT90CAN32	I	32	1	2048	53	16	2.7-5.5	CAN controller	2	2	8	Yes	Yes	2	--	Yes	8	Yes	Yes	Yes	Yes	Yes	34	8	Yes	MLF64, TQFP64	MLF64, TQFP64
AT90CAN64	I	64	2	4096	53	16	2.7-5.5	CAN controller	2	2	8	Yes	Yes	2	--	Yes	8	Yes	Yes	Yes	Yes	Yes	34	8	Yes	MLF64, TQFP64	MLF64, TQFP64
AT90CAN128	P	128	4	4096	53	16	2.7-5.5	CAN controller	2	2	8	Yes	Yes	2	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	34	8	Yes	MLF64, TQFP64	MLF64, TQFP64
AT90USB646	I	64	2	4096	48	16	2.7-5.5	USB	2	2	8	Yes	Yes	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	34	8	Yes		MLF64
AT90USB647	I	64	2	4096	48	16	2.7-5.5	USB+OTG	2	2	8	Yes	Yes	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	34	8	Yes		MLF64, TQFP64
AT90USB1286	I	128	4	8192	48	16	2.7-5.5	USB	2	2	8	Yes	Yes	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	34	8	Yes		MLF64
AT90USB1287	I	128	4	8192	48	16	2.7-5.5	USB+OTG	2	2	8	Yes	Yes	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	34	8	Yes		MLF64, TQFP64
ATmega169	N	16	0.512	1024	54	16	1.8-5.5	LCD 4x25	1	2	4	Yes	1+USI	1	USI	Yes	8	Yes	Yes	Yes	Yes	Yes	23	17	Yes	MLF64, TQFP64	MLF64, TQFP64
ATmega329	P	32	1	2048	54	16	1.8-5.5	LCD 4x25	1	2	4	Yes	1+USI	1	USI	Yes	8	Yes	Yes	Yes	Yes	Yes	25	17	Yes	MLF64, TQFP64	MLF64, TQFP64
ATmega3290	P	32	1	2048	69	16	1.8-5.5	LCD 4x40	1	2	4	Yes	1+USI	1	USI	Yes	8	Yes	Yes	Yes	Yes	Yes	25	32	Yes	TQFP100	TQFP100
ATmega649	I	64	2	4096	54	16	1.8-5.5	LCD 4x25	1	2	4	Yes	1+USI	1	USI	Yes	8	Yes	Yes	Yes	Yes	Yes	25	17	Yes	MLF64, TQFP64	MLF64, TQFP64
ATmega6490	I	64	2	4096	69	16	1.8-5.5	LCD 4x40	1	2	4	Yes	1+USI	1	USI	Yes	8	Yes	Yes	Yes	Yes	Yes	25	32	Yes	TQFP100	TQFP100

Status: F - Future Design, I - Device under introduction, P - Device in production, N - not recommended for new designs

1.6 Product Selection Guide – AVR Z-Link®

Product	Status	AVR	Radio	Flash (KB)	EEPROM (KB)	RAM (KB)	ISM Band [GHz]	Sensitivity [dBm]	Output power [dBm]	Supply Vcc [V]	I/Os
ATmega64RZA	I	mega644	RF230	64	1	4	2.4	-101	3	1.8-3.6	32
ATmega128RZA	I	mega1281	RF230	128	4	8	2.4	-101	3	1.8-3.6	54
ATmega128RZB	I	mega1280	RF230	128	4	8	2.4	-101	3	1.8-3.6	86
ATmega256RZA	I	mega2561	RF230	256	4	8	2.4	-101	3	1.8-3.6	54
ATmega256RZB	I	mega2560	RF230	256	4	8	2.4	-101	3	1.8-3.6	86

Status: F - Future Design, I - Device under introduction, P - Device in production, N - not recommended for new designs

1.7 Product Selection Guide – Automotive AVR

Device	Status	Flash (Kbytes)	EEPROM (Kbytes)	SRAM (Bytes)	Max I/O Pins	F.max (MHz)	Vcc (V)	Specific Features	16-bit Timers	8-bit Timer	PWM (ch)	RTC	SPI	USART	TWI	ISP	10-bit A/D (ch.)	Analog Comparator	Brown Out Detector	Watchdog	On Chip Oscillator	Hardware Multiplier	Interrupts	Ext Interrupts	Self Program Memory	Packages	Green Packages
ATtiny25 Automotive	I	2	0.128	128	6	16	2.7-5.5		--	2	4	--	USI	--	USI	Yes	4	Yes	Yes	Yes	Yes	--	15	7	Yes		SOIC8
ATtiny45 Automotive	P	4	0.256	256	6	16	2.7-5.5		--	2	4	--	USI	--	USI	Yes	4	Yes	Yes	Yes	Yes	--	15	7	Yes		SOIC8
ATtiny85 Automotive	I	8	0.512	512	6	16	2.7-5.5		--	2	4	--	USI	--	USI	Yes	4	Yes	Yes	Yes	Yes	--	15	7	Yes		SOIC8
ATmega48 Automotive	P	4	0.256	512	23	16	2.7-5.5		1	2	6	Yes	1+USART	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	26	26	Yes		MLF32, TQFP32
ATmega88 Automotive	P	8	0.512	1024	23	16	2.7-5.5		1	2	6	Yes	1+USART	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	26	26	Yes		MLF32, TQFP32
ATmega168 Automotive	P	16	0.512	1024	23	16	2.7-5.5		1	2	6	Yes	1+USART	1	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	26	26	Yes		MLF32, TQFP32
AT90CAN128 Automotive	I	128	4	4096	53	16	2.7-5.5	CAN controller	2	2	8	Yes	Yes	2	Yes	Yes	8	Yes	Yes	Yes	Yes	Yes	34	8	Yes	MLF64, TQFP64	MLF64, TQFP64

Status: F - Future Design, I - Device under introduction, P - Device in production, N - not recommended for new designs

2 Application Area in Focus: Comparing power consumption

Written by: Andreas Eieland, Applications Engineer,
System Solutions Group

It is a known fact that there are two main competitors in the ultra low power MCU market. Atmel with the picoPower megaAVRs and Texas Instruments (TI) with the MSP430F2xx series.

Lately there has been debate among development engineers, which MCU is the lowest power consumer. In most cases these discussions are based on readouts from datasheets and not real world applications. When using the datasheets to compare parts it is paramount to compare apples to apples to get the numbers right.

We will address this and other issues that are important when comparing AVRs to MSP430s in this article.

Figure 1 Atmel picoPower technology

2.1.1 AVR BOD vs. TI BOR

Many compare the TI Brown-out reset (BOR) to the AVR Brown-out detect (BOD), based on this they claim that TI have lower power consumption. This is not correct. The TI BOR circuitry is comparable in functionality, power consumption, and protection level to the AVR Power-on-Reset circuitry. The TI BOR and the AVR POR are both considered being “Zero-Power”.

To achieve the same level of protection as the AVR BOD, which is present on all picoPower megaAVRs, you have to use the TI Supply Voltage Supervisor (SVS). The SVS is not present on any parts in the ultra low power MSP430F2xx series. It is only available on larger, more expensive MSP430s.

When enabled the TI SVS is active in all operating modes. The additional current consumed by the SVS is maximum 15 µA. Though the AVR sleeping BOD has a slightly higher maximum consumption of 20 µA in active mode, power is saved compared to the SVS and other traditional BODs by automatically disabling the BOD during sleep mode. The BOD is automatically re-enabled when the controller wakes up from sleep mode and is active before the AVR executes any instructions. This approach provides superior protection with substantially less power drain as the majority of the time is normally spent in sleep mode in low power applications – not in active mode.

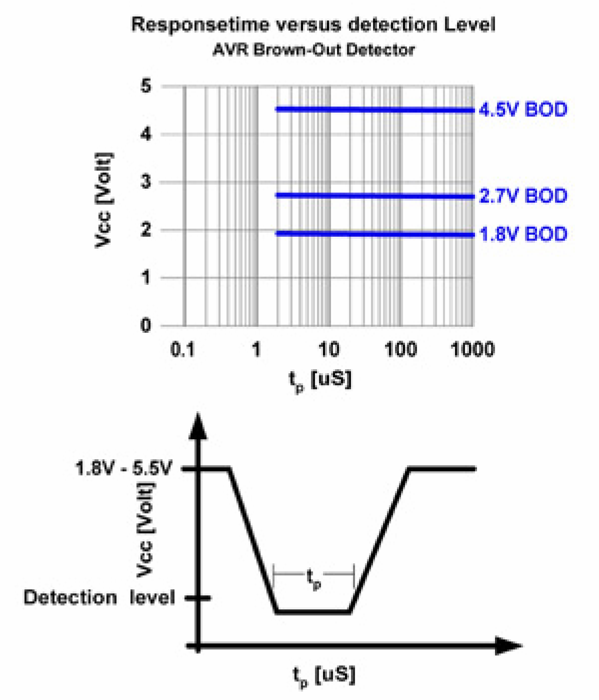

Figure 2: the Atmel picoPower BOD response time

2.1.2 Protection in sleep modes.

The accuracy of BODs is directly proportional to the current they consume. Low- or zero-power BODs tend to be both slow and inaccurate, while more accurate, faster BODs consume more power. Since BODs usually remain on in sleep mode, they represent a substantial drag on battery life. As a result, most vendors of ultra low power MCUs, sacrifice accuracy and speed to reduce current consumption.

In some comparisons it is claimed that the sleeping BOD does not provide sufficient protection in sleep modes since the Brown out detector is turned off. This claim is based on the belief that one cannot predict when a brown out condition occurs. This is not true! In any battery powered system the voltage drops when the current consumption is high. The current consumption is high when the device is in active mode. If the battery voltage drops to below the BOD threshold while the part is in sleep mode, there will be no Flash or EEPROM corruption. The POR is active in all operating modes and will prevent the MCU from performing illegal or undefined operations inside the MCU and on the I/O pins. When the part wakes up, the BOD will trigger immediately if the voltage is to low. This ensures that no code is executed when the system behavior is undefined. With the sleeping BOD the picoPower controllers get the best of both worlds, very good protection while in active mode, and no power consumption penalty in sleep.

2.2 Overall power consumption

There is no family of microcontrollers today that will have the lowest power consumption in all possible applications. Power consumption will always be dependent on the suitability of the MCU to the application. We do however claim that the AVR picoPower devices are the market leader in ultra low power technology, as it will give the lowest power consumption in the majority of applications.

Comparing total power consumption for an application implemented with different controllers is a complex task. Things one has to take into consideration are energy consumed per instruction, amount of instructions needed to perform the operation, and how long time the application can stay in sleep. Depending on the level of system integration on the MCU, it might be necessary to add extra components to the design. The additional power consumption from these parts must also be added. The MSP430 BOR is an example of this. The BOR has a very slow response time and does not give sufficient protection in many high-speed systems. An external BOD is needed to ensure reliable operation, this adds to the total current consumption.

The rule of thumb to reduce power consumption for all MCUs, when not considering external circuitry, is to stay in active mode for the shortest possible time, go to the deepest possible sleep-mode, and stay in the sleep mode for the longest possible time before waking up again. The time spent in active mode is very short for most low-power applications. Hence it is the power consumption in sleep that will make or break the power budget.

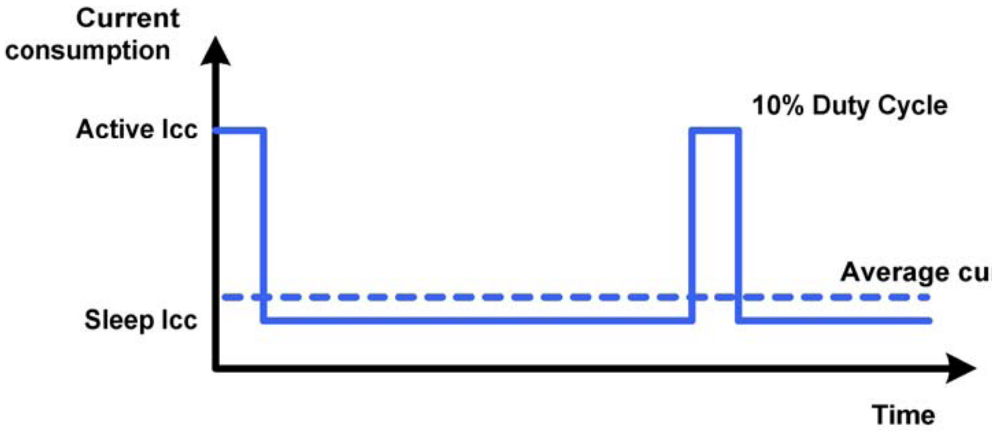

Figure 3 Average power consumption example

The computational performances of the AVR and MSP430 cores are in the same range. For some applications the AVR is better, for other the MSP430 is better. As with power consumption there is no single part, or family of parts, that give the best performance for all applications. The MSP430 is a 16 bit MCU and will be better than the AVR in many applications that use 16 bit arithmetic's, but the AVR uses fewer cycles in most other applications.

AVR MCUs can run up to 25% faster than the MSP430F2xx series. In addition to this the picoPower megaAVRs has lower power consumption than TI in sleep modes and does not require an external BOD. Shorter time in active mode combined with lower power consumption in sleep gives the AVR picoPower devices lower overall power consumption than the MSP430F2xx in most ultra low power applications.

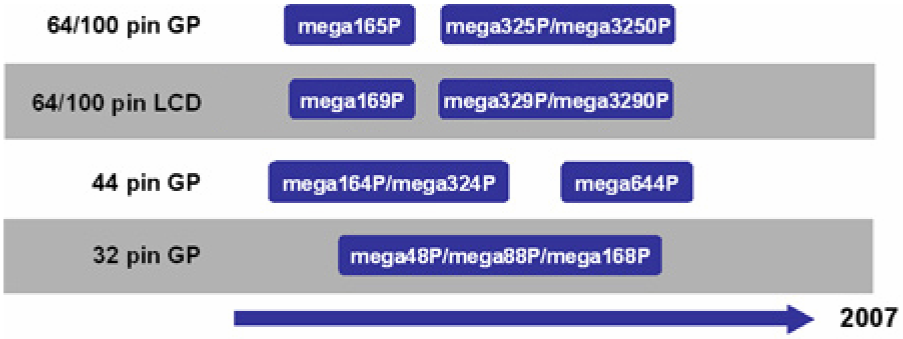

Figure 4 Atmel AVR picoPower devices

For more information about AVR picoPower see: http://www.atmel.com/products/AVR/picopower/

3 AVR Development Tools

Atmel provides a complete range of development tools for the AVR products.

3.1 Tools Reference

Part Number	**Description**
Software	
AVR Studio 4.12	Front end software for AVR development tools
Starter Kits	
STK500	AVR Starter Kit with AVR Studio Interface
STK501	Expansion of STK500 to support 64-pin megaAVR devices
STK502	Expansion of STK500 for 64-pin LCD AVR devices
STK503	Expansion of STK500 for 100-pin megaAVR devices
STK504	Expansion of STK500 for 100-pin LCD AVR devices
STK505	Expansion of STK500 for 14-pin SOIC and 20-pin PDIP AVR devices
STK520	Expansion of STK500 for 90PWM devices
STK525	Starter Kit for AT90USB devices
STK1000	Starter Kit for AVR32AP7xxx devices
Evaluation Kits	
90EIT1	AVR Embedded Internet Tool Kit
AVRBFLY	ATmega169 Demo Board with LCD and Speaker
AVRMC100	BLDC Motor Control with AT90PWM3
AVRMC200	AC Induction Motor Kit
AVRMC201	Induction Motor for ATAVRMC200
AVRFBKIT	DALI Dimmable Fluorescent Ballast Kit
AVRRTOS	AVR Real Time Operating System development kit

Part Number	Description
90USBKEY	AVR USB Key Demonstration Kit
AVRRZ200	Z-Link Demonstration Kit
AKSTK512-3	Remote Access Control – Unidirectional Kit 315 MHz
AKSTK512-4	Remote Access Control – Unidirectional Kit 434 MHz
Development Kits	
DVK90CAN1	DVK90CAN1 Development Kit for AT90CAN devices
AVRSB100	Smart Battery Development Kit for Atmega406
AVRISP2	ISP programmer for all AVR ISP devices
AVRRZ502	Z-Link RF Accessory Kit
AVRDRAGON	Starterkit supporting On-Chip Debugging and programming for AVR. (AVR Dragon will support OCD for all AVRs with 32 kB or less Flash memory and programming for all AVRs. Se section "AVR Studio Tools and Device Support" for current device support. More device support will be available soon)
Emulators	
ICE50	AVR In-Circuit Emulator for all megaAVR and new tinyAVR devices.
JTAGICE2	JTAGICE mkII On-Chip Debugger supporting all AVR and AVR32 with debugWIRE or JTAG interface
JTAGPROBE	JTAGICE mkII Probe including Flex Cables
ADAPTEST	ICE50 Test Adapter
ADAPMEGA8	ICE50 Mega8 PDIP personality adapter
ADAPMEGA32	ICE50 Mega8535/16/32 PDIP personality adapter
ADAP128_TOP	ICE50 Mega64/128 TQFP personality adapter (top module); requires one AT64PSKT_BOT as the bottom module
ADAP169_TOP	ICE50 Mega169 TQFP personality adapter (top module); requires one AT64PSKT_BOT as the bottom module
ADAPMEGA162	ICE50 Mega8515/162 PDIP personality adapter
ADAPTINY26	ICE50 Tiny26 PDIP personality adapter
ADAPTINY13	ICE50 Tiny13 PDIP personality adapter
ADAPT2313	ICE50 Tiny2313 PDIP personality adapter
ATADAPCAN01	STK500/501 90CAN128 CAN adapter
ICE50MEM	ICE50 memory extension card for mega2560/2561
ICE50PROBE	ICE40/50 Probe including Flex Cables
ICE50POD	ICE40 and ICE50 POD replacement kit

3.2 AVR Studio® Tools and Device Support

AVR Studio 4.12 with the latest Service Pack supports all new Atmel debug platforms and devices. Some of the old devices are not supported. See below for a table of currently supported tools and devices in AVR Studio. This support is in progress, and the table below is not guaranteed to be complete when this is read. This information can also be found in the AVR Studio online help and on www.atmel.com/avr

The latest AVR Studio SW can be found on: www.atmel.com/dyn/products/tools_card.asp?tool_id=2725

Device	Simulator/ Assembler	JTAGICE mkII	Starter kit	AVR Dragon	AVR ISP mkII
ATtiny11	•		STK500		
ATtiny12	•		STK500		•
ATtiny13	•	•	STK500	•	•
ATtiny15	•		STK500		•
ATtiny24	•	•	STK500 + STK505		•
ATtiny25	•	•	STK500	•	•
ATtiny26	•		STK500 (+ STK505)		•

Device	Simulator/ Assembler	JTAGICE mkII	Starter kit	AVR Dragon	AVR ISP mkII
ATtiny261	•	•	STK500 (+ STK505)		•
ATtiny28	•		STK500		
ATtiny44	•	•	STK500 + STK505		•
ATtiny45	•	•	STK500	•	•
ATtiny461	•	•	STK500 (+ STK505)		•
ATtiny84	•	•	STK500 + STK505		•
ATtiny85	•	•	STK500	•	•
ATtiny861	•	•	STK500 (+ STK505)		•
ATtiny2313	•	•	STK500	•	•
ATmega48	•	•	STK500	•	•
ATmega8	•		STK500	• (Programming only)	•
ATmega88	•	•	STK500	•	•
ATmega8515	•		STK500		•
ATmega8535	•		STK500		•
ATmega16	•	•	STK500	•	•
ATmega162	•	•	STK500		•
ATmega164P	•	•	STK500		•
ATmega165	•	•	STK500 + STK502		•
ATmega165P	•	•	STK500 + STK502		•
ATmega168	•	•	STK500	•	•
ATmega169	•	•	STK500 + STK502		•
ATmega169P	•	•	STK500 + STK502		•
ATmega32	•	•	STK500	•	•
ATmega324P	•	•	STK500		•
ATmega325	•	•	STK500 + STK502		•
ATmega325P	•	•	STK500 + STK502	•	•
ATmega3250	•	•	STK500 + STK504		•
ATmega3250P	•	•	STK500 + STK504	•	•
ATmega329	•	•	STK500 + STK502		•
ATmega329P	•	•	STK500 + STK502	•	•
ATmega3290	•	•	STK500 + STK504		•
ATmega3290P	•	•	STK500 + STK504	•	•
ATmega64	•	•	STK500 +STK501		•
ATmega640	•	•	STK500 + STK503		•
ATmega644	•	•	STK500		•
ATmega644P	•	•	STK500		•
ATmega645	•	•	STK500 + STK502		•
ATmega6450	•	•	STK500 + STK504		•
ATmega649	•	•	STK500 + STK502		•
ATmega6490	•	•	STK500 + STK504		•
ATmega128	•	•	STK500 + STK501	• (Programming only)	•
ATmega1280	•	•	STK500 + STK503		•
ATmega1281	•	•	STK500 + STK501		•
ATmega2560	•	•	STK500 + STK503		•
ATmega2561	•	•	STK500 + STK501		•
ATmega406	•	•			
AT90CAN32	•	•	STK500 + STK501 + ATADAPCAN1		•
AT90CAN64	•	•	STK500 + STK501 + ATADAPCAN1		•

Device	Simulator/ Assembler	JTAGICE mkII	Starter kit	AVR Dragon	AVR ISP mkII
AT90CAN128	•	•	STK500 + STK501 + ATADAPCAN1		•
AT90PWM2	•	•	STK500 + STK520		•
AT90PWM3	•	•	STK500 + STK520		•
AT90USB646	•	•	STK500 + STK525		•
AT90USB647	•	•	STK500 + STK525		•
AT90USB1286	•	•	STK500 + STK525		•
AT90USB1287	•	•	STK500 + STK525		•
AT32AP7000		•	STK1000		
AT32AP7001		•	STK1000		
AT32AP7002		•	STK1000		

4 Documentation

All documents listed can be downloaded from Atmel Corporation's web site: http://www.atmel.com under the product section. For other documentation, please send your request to avr@atmel.com.

4.1 Datasheets

The datasheets of all AVR devices can be downloaded.
AVR: http://www.atmel.com/dyn/products/datasheets.asp?family_id=607.
AVR32: http://www.atmel.com/dyn/products/datasheets.asp?family_id=682

Family	Devices	Language	Preliminary	Summary	Complete	Last Update
Auto AVR	ATtiny25/45/85 Automotive	English	X		X	09/2006
Auto AVR	ATmega48/88/168 Automotive	English			X	09/2006
Auto AVR	ATmega88 Automotive - 150°C Specification - Appendix A	English	X			09/2006
Auto AVR	AT90CAN128 Automotive	English	X		X	09/2006
CAN AVR	AT90CAN32/64/128	English	X	X	X	*11/2006*
CAN AVR	AT90CAN128	English		X	X	05/2006
LCD AVR	ATmega169(V	English		X	X	07/2006
LCD AVR	ATmega169(V)	Chinese	X		X	9/04
LCD AVR	ATmega329/3290/649/6490	English	X	X	X	*11/2006*
USB AVR	AT90USB1287/1286/646/647	English	X		X	02/06
Lighting AVR	AT90PWM2, AT90PWM3	English	X		X	*12/2006*
AVR Z-Link	AT86RF230 ZigBee™/IEEE 802.15.4-Transceiver	English	X		X	06/2006
megaAVR	ATmega48/88/168	English	X	X	X	*12/2006*
megaAVR	ATmega48/88/168	Chinese	X		X	02/05
megaAVR	ATmega8(L)	English		X	X	10/2006
megaAVR	ATmega8(L)	Chinese			X	7/04
megaAVR	ATmega8515(L)	English		X	X	10/2006
megaAVR	ATmega8515(L)	Chinese			X	9/04
megaAVR	ATmega8535(L)	English	X	X	X	10/2006
megaAVR	ATmega8535(L)	Chinese	X		X	9/04
megaAVR	ATmega16(L)	English		X	X	10/2006
megaAVR	ATmega16(L)	Chinese			X	10/04
megaAVR	ATmega162(V)	English		X	X	04/06
megaAVR	Atmega164P/324P/644P	English	X	X	X	10/2006
megaAVR	ATmega165(V)	English	X	X	X	08/2006
megaAVR	ATmega32(L)	English		X	X	10/2006
megaAVR	ATmega32(L)	Chinese	X		X	09/04
megaAVR	ATmega325/3250/645/6450	English	X	X	X	*11/2006*

Family	Devices	Language	Preliminary	Summary	Complete	Last Update
megaAVR	ATmega64(L)	English		X	X	10/2006
megaAVR	ATmega64(L)	Chinese	X		X	09/04
megaAVR	ATmega640/1280/1281/2560/2561	English	X	X	X	09/2006
megaAVR	ATmega644	English	X	X	X	09/2006
megaAVR	ATmega128(L)	English		X	X	10/2006
megaAVR	ATmega128(L)	Chinese			X	05/04
picoPower megaAVR	ATmega164P/324P/644P	English	X	X	X	09/2006
picoPower megaAVR	ATmega165P(V)	English	X	X	X	*11/2006*
picoPower megaAVR	*ATmega325P/3250P*	*English*	*X*	*X*	*X*	*12/2006*
picoPower LCD megaAVR	ATmega169P(V)	English	X	X	X	*11/2006*
picoPower LCD megaAVR	*ATmega329P/3290P*	*English*	*X*	*X*	*X*	*12/2006*
Smart Battery AVR	ATmega406	English	X	X	X	07/2006
tinyAVR	ATtiny11/12	English		X	X	07/2006
tinyAVR	ATtiny13	English	X	X	X	10/04
tinyAVR	ATtiny13	Chinese	X		X	04/04
tinyAVR	ATtiny15L	English		X	X	06/05
tinyAVR	ATtiny2313	English	X	X	X	04/2006
tinyAVR	ATtiny2313	Chinese	X		X	07/04
tinyAVR	ATtiny24/44/84	English	X	X	X	09/2006
tinyAVR	ATtiny25/45/85	English	X	X	X	*12/2006*
tinyAVR	ATtiny26(L)	English		X	X	10/2006
tinyAVR	ATtiny26(L)	Chinese	X		X	12/03
tinyAVR	ATtiny261/461/861	English	X	X	X	11/2006
tinyAVR	ATtiny28(L)(V)	English		X	X	07/2006
USB AVR	AT90USB1286, AT90USB1287, AT90USB646, AT90USB647	English	X		X	07/2006
USB AVR	USB DFU Bootloader Datasheet	English				04/06
AVR32	AT32AP7000	English	X	X	X	10/2006
AVR32	AVR32 Architecture Manual	English	X		X	02/06
AVR32	AVR32 Technical Reference Manual	English	X		X	06/2006
AVR32	AVR32 Java Technical Reference Manual	English	X		X	10/2006

4.2 Application Notes

The application notes for all AVR devices can be downloaded.
AVR: http://www.atmel.com/dyn/products/app_notes.asp?family_id=607
AVR32: http://www.atmel.com/dyn/products/app_notes.asp?family_id=682

Note Number	Description	Last Update
AVR000	Register and Bit-Name Definitions for the AVR Microcontroller	4/98
AVR001	Conditional Assembly and Portability Macros	3/05
AVR030	Getting Started with IAR Embedded Workbench for Atmel AVR	10/04
AVR031	Getting Started with ImageCraft C for AVR	5/02
AVR032	Linker Command Files for the IAR ICCA90 Compiler	5/02
AVR033	Getting Started with the CodeVision AVR C Compiler	5/02
AVR034	Mixing C and Assembly Code with AVR Embedded Workbench for AVR	4/03
AVR035	Efficient C Coding for AVR	1/04
AVR040	EMC Design Considerations	06/2006
AVR042	AVR Hardware Design Considerations	06/2006
AVR053	Calibration of the Internal RC Oscillator	05/2006

Note Number	Description	Last Update
AVR054	Run-time calibration of the internal RC oscillator	02/2006
AVR055	Using a 32kHz XTAL for run-time calibration of the internal RC	02/2006
AVR060	JTAGICE	01/04
AVR061	STK500 Protocol	4/03
AVR063	LCD Driver for the STK®504	04/2006
AVR064	STK502 – A Temperature Monitoring System with LCD Output	02/2006
AVR065	LCD Driver for the STK502	02/2006
AVR067	JTAGICE mkII Communication Protocol	04/2006
AVR068	STK500 Communication Protocol	06/2006
AVR069	AVRISP mkII Communication Protocol	02/2006
AVR070	Modifying AT90ICEPRO to Support Emulation of AT90	5/02
AVR072	Accessing 16-bit I/O Registers	5/02
AVR073	Accessing 10- and 16-bit registers in ATtiny261/461/861	*12/2006*
AVR074	Upgrading AT90ICEPRO to ICE10	5/02
AVR080	ATmega103 Replaced by ATmega128	01/04
AVR081	Replacing AT90S4433 by ATmega8	7/03
AVR082	Replacing ATmega161 by ATmega162	01/04
AVR083	Replacing ATmega163 by ATmega16	09/05
AVR084	Replacing ATmega323 by ATmega32	7/03
AVR085	Replacing AT90S8515 by ATmega8515	1/04
AVR086	Replacing AT90S8535 by ATmega8535	7/03
AVR087	Migrating between ATmega8515 and ATmega162	7/03
AVR088	Migrating between ATmega8535 and ATmega16	1/04
AVR089	Migrating between ATmega16 and ATmega32	6/03
AVR090	Migrating between ATmega64 and ATmega128	6/03
AVR091	Replacing AT90S2313 by ATtiny2313	10/03
AVR092	Replacing ATtiny11/12 by ATtiny13	10/03
AVR093	Replacing AT90S1200 by ATtiny2313	10/03
AVR094	Replacing ATmega8 by ATmega88	4/05
AVR095	Migrating between ATmega48, ATmega88 and ATmega168	2/04
AVR096	Migrating from ATmega128 to AT90CAN128	3/04
AVR097	Migration between Atmega128 and ATmega2561	07/2006
AVR098	Migration between ATmega169, ATmega329 and ATmega649	04/2006
AVR099	Replacing AT90S4433 by ATmega48	07/04
AVR100	Accessing the EEPROM	09/05
AVR101	High Endurance EEPROM Storage	9/02
AVR102	Block Routines	5/02
AVR103	Using the EEPROM Programming Modes	3/05
AVR104	Buffered Interrupt Controlled EEPROM Writes	7/03
AVR105	Power Efficient High Endurance Parameter Storage in Flash Memory	9/03
AVR106	C functions for reading and writing to Flash memory	08/2006
AVR107	Interfacing AVR serial memories	3/05
AVR108	Setup and Use of the LPM Instructions	5/02
AVR109	Self-programming	6/04
AVR120	Characterization and Calibration of the ADC on an AVR	02/2006
AVR121	Enhancing ADC resolution by oversampling	09/05
AVR128	Setup and Use the Analog Comparator	5/02
AVR130	Setup and use the AVR Timers	2/02
AVR131	Using the AVR's High-speed PWM	9/03
AVR132	Using the Enhanced Watchdog Timer	11/03
AVR133	Long Delay Generation Using the AVR Microcontroller	2/04
AVR134	Real-Time Clock using the Asynchronous Timer	08/2006
AVR135	Using Timer Capture to Measure PWM Duty Cycle	10/2005
AVR136	Low-jitter Multi-channel Software PWM	05/06
AVR137	Writing Software Compatible for AT90PWM2/3 and AT90PWM2B/3B	12/2006
AVR140	ATmega48/88/168 family run-time calibration of the Internal RC oscillator	09/2006

Note Number	Description	Last Update
AVR151	Setup and use of the SPI	09/05
AVR155	Accessing I2C LCD Display Using the AVR 2-Wire Serial Interface	09/05
AVR180	External Brown-Out Protection	5/02
AVR182	Zero Cross Detector	1/04
AVR191	Anti-Pinch Algorithm for AVR Adaptation Procedure	11/2006
AVR200	Multiply and Divide Routines	05/2006
AVR201	Using the AVR Hardware Multiplier	6/02
AVR202	16-Bit Arithmetic	5/02
AVR204	BCD Arithmetic	1/03
AVR220	Bubble Sort	5/02
AVR221	Discrete PID controller	05/2006
AVR222	8-Point Moving Average Filter	5/02
AVR223	Digital Filters with AVR	9/02
AVR230	DES Bootloader	4/05
AVR231	AES Bootloader	08/2006
AVR236	CRC Check of Program Memory	5/02
AVR240	4x4 Keypad-Wake Up on Keypress	06/2006
AVR241	Direct driving of LCD display using general I/O	5/04
AVR242	8-bit Microcontroller Multiplexing LED Drive & a 4x4 Keypad	5/02
AVR243	Matrix Keyboard Decoder	1/03
AVR244	UART as ANSI Terminal Interface	11/03
AVR245	Code Lock with 4x4 Keypad and I2C™ LCD	10/2005
AVR270	USB Mouse Demonstration	02/2006
AVR271	USB Keyboard Demonstration	02/2006
AVR272	USB CDC Demonstration UART to USB Bridge	04/2006
AVR273	USB Mass Storage Implementation	04/2006
AVR301	C Code for Interfacing AVR® to AT17CXX FPGA Configuration Memory	1/04
AVR303	SPI-UART Gateway	3/05
AVR304	Half Duplex Interrupt Driven Software UART	8/97
AVR305	Half Duplex Compact Software UART	09/05
AVR306	Using the AVR UART in C	7/02
AVR307	Half Duplex UART Using the USI Module	10/03
AVR308	Software LIN Slave	5/02
AVR309	Software Universal Serial Bus (USB)	02/2006
AVR310	Using the USI Module as a I2C Master	9/04
AVR311	Using the TWI Module as a I2C Slave	10/04
AVR312	Using the USI Module as a I2C Slave	09/05
AVR313	Interfacing the PCAT Keyboard	09/05
AVR314	DTMF Generator	5/02
AVR315	Using the TWI Module as a I2C Master	10/04
AVR316	SMBus Slave Using the TWI Module	10/2005
AVR317	Using the USART on the ATmega48/88/168 as a SPI master	11/04
AVR318	Dallas 1-Wire® Master	10/04
AVR319	Using the USI module for SPI communication	11/04
AVR320	Software SPI Master	09/05
AVR322	LIN v1.3 Protocol Implementation on Atmel AVR Microcontrollers	12/05
AVR323	Interfacing GSM modems	02/2006
AVR325	High-Speed Interface to Host EPP Parallel Port	2/02
AVR328	USB Generic HID Implementation	1/06
AVR329	USB Firmware Architecture	02/2006
AVR335	Digital Sound Recorder with AVR and Serial Data Flash	4/05
AVR336	ADPCM Decoder	1/05
AVR350	XmodemCRC Receive Utility for AVR	09/05
AVR360	Step Motor Controller	4/03
AVR400	Low Cost A/D Converter	5/02
AVR401	8-Bit Precision A/D Converter	2/03

Note Number	Description	Last Update
AVR410	RC5 IR Remote Control Receiver	5/02
AVR411	Secure Rolling Code Algorithm for Wireless Link	04/06
AVR414	User Guide - ATAVRRZ502 - Accessory Kit	*12/2006*
AVR415	RC5 IR Remote Control Transmitter	5/03
AVR433	Power Factor Corrector (PFC) with AT90PWM2 Re-triggable High Speed PSC	03/2006
AVR434	PSC Cookbook	10/2006
AVR435	BLDC/BLAC Motor Control Using a Sinus Modulated PWM Algorithm	09/2006
AVR440	Sensorless Control of Two-Phase Brushless DC Motor	09/05
AVR441	Intelligent BLDC Fan Controller with Temperature Sensor and Serial Interface	9/05
AVR442	PC Fan Control using ATtiny13	9/05
AVR443	Sensor-based control of three phase Brushless DC motor	02/2006
AVR444	Sensorless control of 3-phase brushless DC motors	10/2005
AVR446	Linear speed control of stepper motor	06/2006
AVR447	Sinusoidal driving of three-phase permanent magnet motor using ATmega48/88/168	06/2006
AVR448	Control of High Voltage 3-Phase BLDC Motro	05/2006
AVR449	Sinusoidal driving of 3-phase permanent magnet motor using ATtiny261/461/861	10/2006
AVR450	Battery Charger for SLA, NiCd, NiMH and Li-ion Batteries	09/2006
AVR452	Sensor-based Control of Three Phase Brushless DC Motors Using CAN128 64 32 or mega128 64 usi	03/2006
AVR453	Smart Battery Reference Design	02/2006
AVR454	Users Guide – ATAVRSB100 – Smart Battery Development Board	06/2006
AVR460	Embedded Web Server	5/02
AVR461	Quick Start Guide for the Embedded Internet Toolkit	5/02
AVR462	Reducing the Power Consumption of ATEIT1	3/02
AVR465	Energy meter	7/04
AVR480	Anti-Pinch System for Electrical Window	*12/2006*
AVR492	Brushless DC Motor Control using AT90PWM3	7/05
AVR493	Sensorless Commutation of Brushless DC Motor (BLDC) using AT90PWM3 and ATAVRMC100	07/2006
AVR494	AC Induction Motor Control Using the constant V/f Principle and a Natural PWM Algorithm	12/05
AVR495	AC Induction Motor Control Using the Constant V/f Principle and a Space-vector PWM Algorithm	02/2006
AVR500	Migration between Atmega64 and Atmega645	9/04
AVR501	Replacing ATtiny15 with ATtiny25	3/05
AVR502	Migration between Atmega165 and ATmega325	11/04
AVR503	Replacing AT90S/LS2323 or AT90S/LS2343 with ATtiny25	09/05
AVR504	Migrating from ATtiny26 to ATtiny261/461/861	10/2006
AVR505	Migration between Atmega16/32 and ATmega164/324/644	06/2006
AVR506	Migration from Atmega169 to ATmega169P	06/2006
AVR507	*Migration from ATmega329 to ATmega329P*	*11/2006*
AVR508	Migration from ATmega644 to ATmega644P	07/2006
AVR509	*Migration between ATmega169P and ATmega329P*	*11/2006*
AVR510	Migration between ATmega329/649 and ATmega3290/6490	07/2006
AVR511	*Migration from ATmega3290 to ATmega3290P*	*11/2006*
AVR910	In-System Programming	11/00
AVR911	AVR Open-source Programmer	7/04
AVR914	CAN & UART based Bootloader for AT90CAN32, AT90CAN64, & AT90CAN128	05/2006
AVR32000	Introduction to AVR32 header files	05/2006
AVR32100	Using the AVR32 USART	4/06
AVR32101	Configuring the AVR32 Interrupt Controller	4/06
AVR32102	Using the AVR32 SDRAM Controller	05/2006
AVR32105	Master and Slave SPI Driver	05/2006
AVR32107	Using TWI as a master on the AVR32	4/06

Note Number	Description	Last Update
AVR32108	Peripheral Direct Memory Access Driver	05/2006
AVR32110	Using the AVR32 Timer/Counter	05/2006
AVR32111	Using the AVR32 PIO Controller	05/2006
AVR32113	Configuration and Use of the Memory Management Unit	09/2006