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 "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	AVR
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	27
Program Memory Size	16KB (8K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	32-TQFP
Supplier Device Package	32-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega168pb-anr

3. Ordering Information

3.1. ATmega48PB

Speed [MHz](3)	Power Supply [V]	Ordering Code(2)	Package(1)	Operational Range
20	1.8 - 5.5	ATmega48PB-AU ATmega48PB-AUR(4) ATmega48PB-MU ATmega48PB-MUR(4)	32A 32A 32MS1 32MS1	Industrial (-40°C to +85°C)
		ATmega48PB-AN ATmega48PB-ANR(4) ATmega48PB-MN ATmega48PB-MNR(4)	32A 32A 32MS1 32MS1	Industrial (-40°C to +105°C)

Note: 1. This device can also be supplied in wafer form. Contact your local Atmel sales office for detailed ordering information and minimum quantities.

2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.

3. See ”Speed Grades” on page 304.

4. Tape & Reel.

Package Type	
32A	32-lead, Thin (1.0mm) Plastic Quad Flat Package (TQFP)
32MS1	32-pad, 5.0x5.0x0.9mm body, Lead Pitch 0.50mm, Very-thin Fine pitch, Quad Flat No Lead Package (VFQFN)

3.2. ATmega88PB

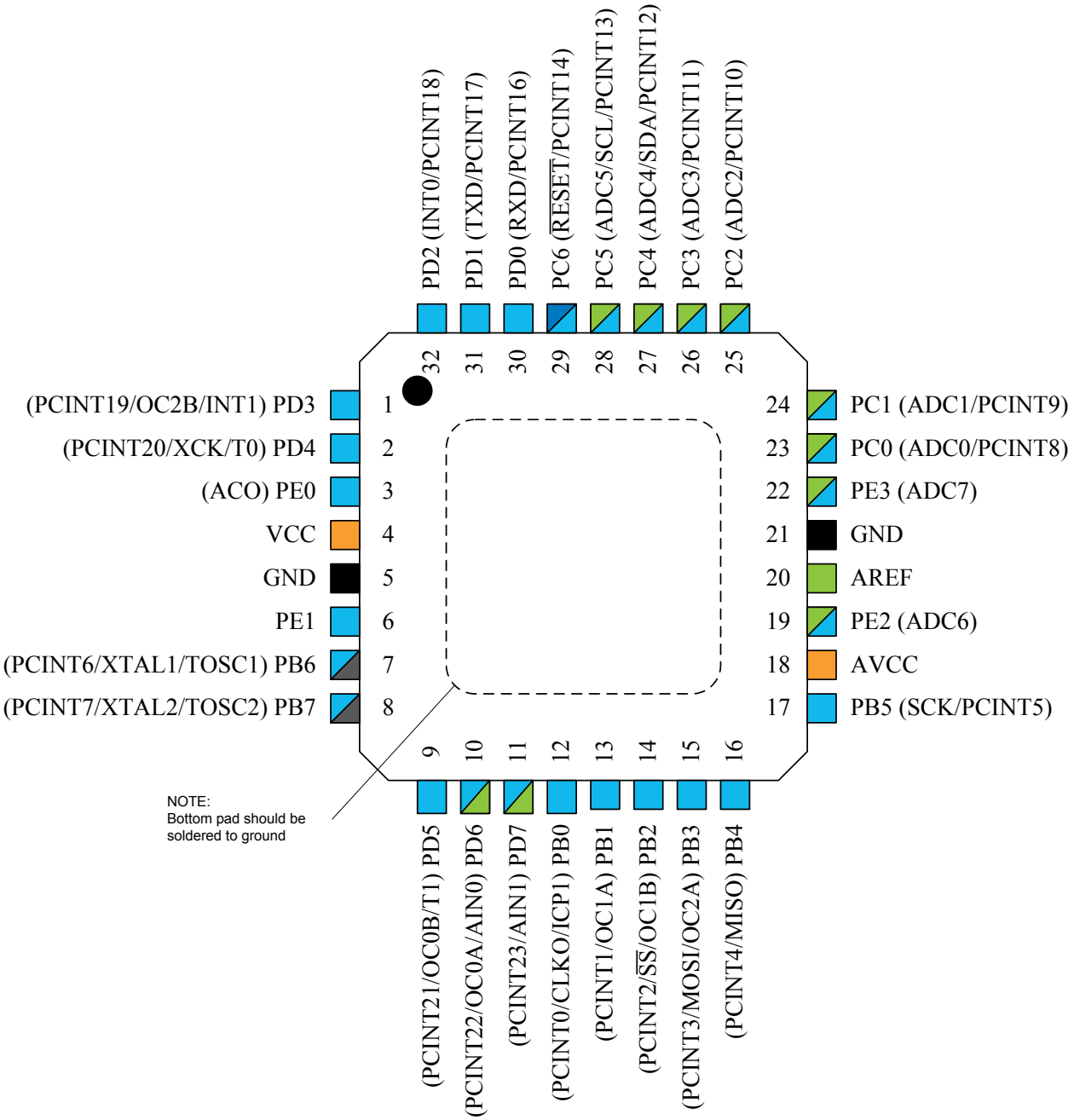
Speed [MHz](3)	Power Supply [V]	Ordering Code(2)	Package(1)	Operational Range
20	1.8 - 5.5	ATmega88PB-AU ATmega88PB-AUR(4) ATmega88PB-MU ATmega88PB-MUR(4)	32A 32A 32MS1 32MS	Industrial (-40°C to +85°C)
		ATmega88PB-AN ATmega88PB-ANR(4) ATmega88PB-MN ATmega88PB-MNR(4)	32A 32A 32MS1 32MS1	Industrial (-40°C to +105°C)

Note: 1. This device can also be supplied in wafer form. Contact your local Atmel sales office for detailed ordering information and minimum quantities.

2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive).Also Halide free and fully Green.

3. See ”Speed Grades” on page 304.

Figure 5-2. 32 VQFN Pinout ATmega48PB/88PB/168PB



5.1. Pin Descriptions

5.1.1. VCC

Digital supply voltage.

5.1.2. GND

Ground.

5.1.8. AV_{CC}

AV_{CC} is the supply voltage pin for the A/D Converter, PC[3:0], and PE[3:2]. 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. Note that PC[6:4] use digital supply voltage, V_{CC}.

5.1.9. AREF

AREF is the analog reference pin for the A/D Converter.

5.1.10. ADC[7:6] (TQFP and VFQFN Package Only)

In the TQFP and VFQFN package, ADC[7:6] serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

8. Resources

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

18. EXINT - External Interrupts

The External Interrupts are triggered by the INT0 and INT1 pin or any of the PCINT pins. Observe that, if enabled, the interrupts will trigger even if the INT0 and INT1 or PCINT[23:0] pins are configured as outputs. This feature provides a way of generating a software interrupt. The pin change interrupt PCI2 will trigger if any enabled PCINT pin toggles. The pin change interrupt PCI1 will trigger if any enabled PCINT[14:8] pin toggles. The pin change interrupt PCI0 will trigger if any enabled PCINT[7:0] pin toggles. The PCMSK2 , PCMSK1, and PCMSK0 Registers control which pins contribute to the pin change interrupts. Pin change interrupts on PCINT are detected asynchronously. This implies that these interrupts can be used for waking the part also from sleep modes other than Idle mode.

The INT0 and INT1 interrupts can be triggered by a falling or rising edge or a low level. This is set up as indicated in the specification for the External Interrupt Control Register A (EICRA). When the INT0 or INT1 interrupts are enabled and are configured as level triggered, the interrupts will trigger as long as the pin is held low. Note that recognition of falling or rising edge interrupts on INT0 or INT1 requires the presence of an I/O clock. Low level interrupt on INT0 and INT1 is detected asynchronously. This implies that this interrupt can be used for waking the part also from sleep modes other than Idle mode. The I/O clock is halted in all sleep modes except Idle mode.

Note: Note that if a level triggered interrupt is used for wake-up from Power-down, the required level must be held long enough for the MCU to complete the wake-up to trigger the level interrupt. If the level disappears before the end of the Start-up Time, the MCU will still wake up, but no interrupt will be generated. The start-up time is defined by the SUT and CKSEL Fuses.

Related Links

System Control and Reset on page 70

18.1. Pin Change Interrupt Timing

An example of timing of a pin change interrupt is shown in the following figure.

Table 19-2. Generic Description of Overriding Signals for Alternate Functions

Signal Name	Full Name	Description
PUOE	Pull-up Override Enable	If this signal is set, the pull-up enable is controlled by the PUOV signal. If this signal is cleared, the pull-up is enabled when {DDxn, PORTxn, PUD} = 0b010.
PUOV	Pull-up Override Value	If PUOE is set, the pull-up is enabled/disabled when PUOV is set/cleared, regardless of the setting of the DDxn, PORTxn, and PUD Register bits.
DDOE	Data Direction Override Enable	If this signal is set, the Output Driver Enable is controlled by the DDOV signal. If this signal is cleared, the Output driver is enabled by the DDxn Register bit.
DDOV	Data Direction Override Value	If DDOE is set, the Output Driver is enabled/disabled when DDOV is set/cleared, regardless of the setting of the DDxn Register bit.
PVOE	Port Value Override Enable	If this signal is set and the Output Driver is enabled, the port value is controlled by the PVOV signal. If PVOE is cleared, and the Output Driver is enabled, the port Value is controlled by the PORTxn Register bit.
PVOV	Port Value Override Value	If PVOE is set, the port value is set to PVOV, regardless of the setting of the PORTxn Register bit.
DIEOE	Digital Input Enable Override Enable	If this bit is set, the Digital Input Enable is controlled by the DIEOV signal. If this signal is cleared, the Digital Input Enable is determined by MCU state (Normal mode, sleep mode).
DIEOV	Digital Input Enable Override Value	If DIEOE is set, the Digital Input is enabled/disabled when DIEOV is set/cleared, regardless of the MCU state (Normal mode, sleep mode).
DI	Digital Input	This is the Digital Input to alternate functions. In the figure, the signal is connected to the output of the Schmitt Trigger but before the synchronizer. Unless the Digital Input is used as a clock source, the module with the alternate function will use its own synchronizer.
AIO	Analog Input/Output	This is the Analog Input/output to/from alternate functions. The signal is connected directly to the pad, and can be used bi-directionally.

The following subsections shortly describe the alternate functions for each port, and relate the overriding signals to the alternate function. Refer to the alternate function description for further details.

19.3.1. Alternate Functions of Port B

The Port B pins with alternate functions are shown in the table below:

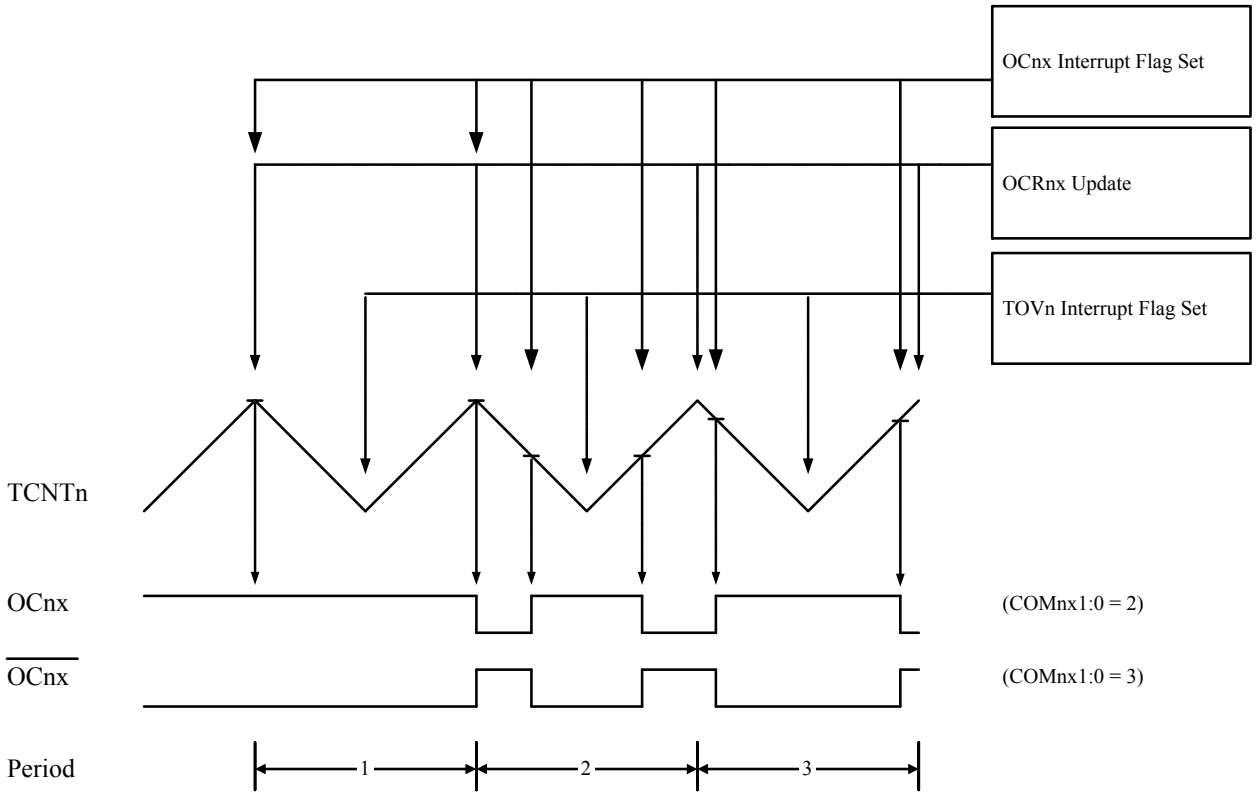
have a maximum frequency of $f_{OC0} = f_{clk\_I/O}/2$ when OCR0A=0x00. This feature is similar to the OC0A toggle in CTC mode, except double buffering of the Output Compare unit is enabled in the Fast PWM mode.

20.7.4. Phase Correct PWM Mode

The Phase Correct PWM mode (WGMn[2:0]=0x1 or WGMn[2:0]=0x5) provides a high resolution, phase correct PWM waveform generation. The Phase Correct PWM mode is based on dual-slope operation: The counter counts repeatedly from BOTTOM to TOP, and then from TOP to BOTTOM. When WGMn[2:0]=0x1 TOP is defined as 0xFF. When WGMn[2:0]=0x5, TOP is defined as OCRnA. In non-inverting Compare Output mode, the Output Compare (OCnx) bit is cleared on compare match between TCNTn and OCRnx while up-counting, and OCnx is set on the compare match while down-counting. In inverting Output Compare mode, the operation is inverted. The dual-slope operation has a lower maximum operation frequency than single slope operation. Due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

In Phase Correct PWM mode the counter is incremented until the counter value matches TOP. When the counter reaches TOP, it changes the count direction. The TCNTn value will be equal to TOP for one timer clock cycle. The timing diagram for the Phase Correct PWM mode is shown below. The TCNTn value is shown as a histogram for illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNTn slopes represent compare matches between OCRnx and TCNTn.

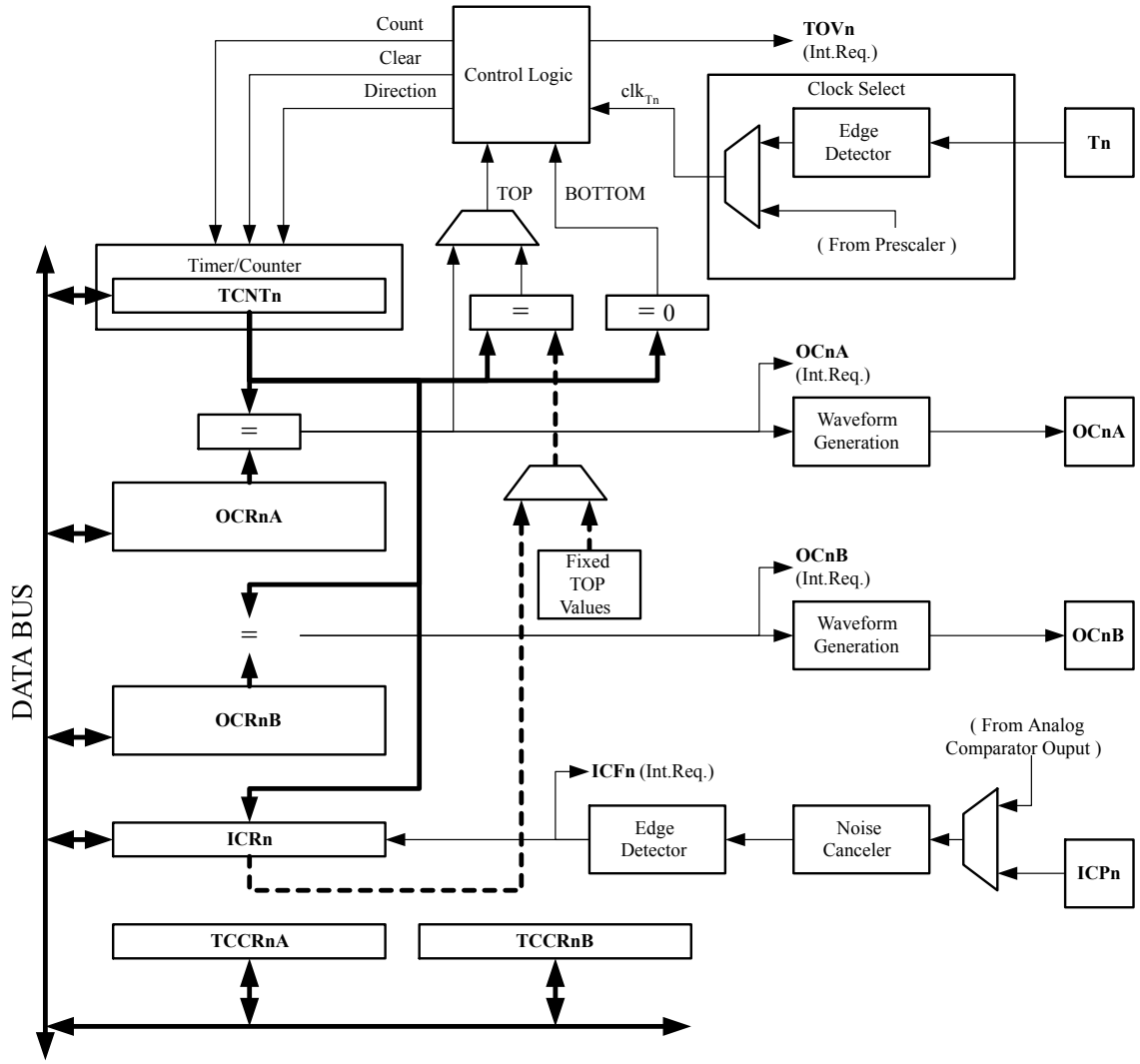
Figure 20-7. Phase Correct PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV0) is set each time the counter reaches BOTTOM. The Interrupt Flag can be used to generate an interrupt each time the counter reaches the BOTTOM value.

In Phase Correct PWM mode, the compare unit allows generation of PWM waveforms on the OCnx pin. Writing the COMnx[1:0] bits to 0x2 will produce a non-inverted PWM. An inverted PWM output can be generated by writing COMnx[1:0]=0x3: Setting the Compare Match Output A Mode bit to '1'

Figure 21-1. 16-bit Timer/Counter Block Diagram



See the related links for actual pin placement.

Related Links

I/O-Ports on page 105

Alternate Port Functions on page 109

Alternate Functions of Port B on page 111

Alternate Functions of Port D on page 118

Pin Configurations on page 14

PRR on page 69

21.4. Accessing 16-bit Registers

The TCNT1, OCR1A/B, and ICR1 are 16-bit registers that can be accessed by the AVR CPU via the 8-bit data bus. The 16-bit register must be accessed byte-wise, using two read or write operations. Each 16-bit timer has a single 8-bit TEMP register for temporary storing of the high byte of the 16-bit access. The same temporary register is shared between all 16-bit registers within each 16-bit timer.

Accessing the low byte triggers the 16-bit read or write operation: When the low byte of a 16-bit register is written by the CPU, the high byte that is currently stored in TEMP and the low byte being written are both

A change of the COM2x1:0 bits state will have effect at the first compare match after the bits are written. For non-PWM modes, the action can be forced to have immediate effect by using the FOC2x strobe bits.

23.7. Modes of Operation

The mode of operation, i.e., the behavior of the Timer/Counter and the Output Compare pins, is defined by the combination of the Waveform Generation mode (WGM2[2:0]) and Compare Output mode (COM2x[1:0]) bits. The Compare Output mode bits do not affect the counting sequence, while the Waveform Generation mode bits do. The COM2x[1:0] bits control whether the PWM output generated should be inverted or not (inverted or non-inverted PWM). For non-PWM modes the COM2x[1:0] bits control whether the output should be set, cleared, or toggled at a compare match (See Compare Match Output Unit).

For detailed timing information refer to Timer/Counter Timing Diagrams.

23.7.1. Normal Mode

The simplest mode of operation is the Normal mode (WGM2[2:0] = 0). In this mode the counting direction is always up (incrementing), and no counter clear is performed. The counter simply overruns when it passes its maximum 8-bit value (TOP = 0xFF) and then restarts from the bottom (0x00). In normal operation the Timer/Counter Overflow Flag (TOV2) will be set in the same timer clock cycle as the TCNT2 becomes zero. The TOV2 Flag in this case behaves like a ninth bit, except that it is only set, not cleared. However, combined with the timer overflow interrupt that automatically clears the TOV2 Flag, the timer resolution can be increased by software. There are no special cases to consider in the Normal mode, a new counter value can be written anytime.

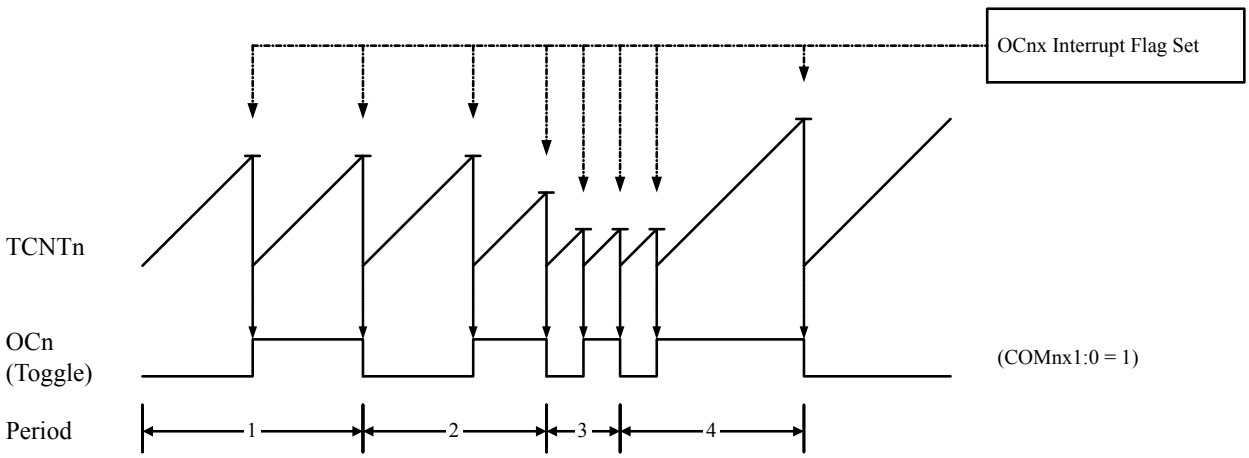
The Output Compare unit can be used to generate interrupts at some given time. Using the Output Compare to generate waveforms in Normal mode is not recommended, since this will occupy too much of the CPU time.

23.7.2. Clear Timer on Compare Match (CTC) Mode

In Clear Timer on Compare or CTC mode (WGM2[2:0] = 2), the OCR2A Register is used to manipulate the counter resolution. In CTC mode the counter is cleared to zero when the counter value (TCNT2) matches the OCR2A. The OCR2A defines the top value for the counter, hence also its resolution. This mode allows greater control of the compare match output frequency. It also simplifies the operation of counting external events.

The timing diagram for the CTC mode is as follows. The counter value (TCNT2) increases until a compare match occurs between TCNT2 and OCR2A, and then counter (TCNT2) is cleared.

Figure 23-5. CTC Mode, Timing Diagram



When using the feature without start interrupt, the start detection logic activates the internal 8MHz oscillator and the USART clock while the frame is being received, only. Other clocks remain stopped until the Receive Complete Interrupt wakes up the MCU.

The maximum baud rate in synchronous mode depends on the sleep mode the device is woken up from, as follows:

- Idle sleep mode: system clock frequency divided by four
- Standby or Power-down: 500kbps

The maximum baud rate in asynchronous mode depends on the sleep mode the device is woken up from, as follows:

- Idle sleep mode: the same as in active mode

Table 25-4. Maximum Total Baudrate Error in Normal Speed Mode

Baudrate	Frame Size					
	5	6	7	8	9	10
0 - 28.8kbps	+6.67 -5.88	+5.79 -5.08	+5.11 -4.48	+4.58 -4.00	+4.14 -3.61	+3.78 -3.30
38.4kbps	+6.63 -5.88	+5.75 -5.08	+5.08 -4.48	+4.55 -4.00	+4.12 -3.61	+3.76 -3.30
57.6kbps	+6.10 -5.88	+5.30 -5.08	+4.69 -4.48	+4.20 -4.00	+3.80 -3.61	+3.47 -3.30
76.8kbps	+5.59 -5.88	+4.85 -5.08	+4.29 -4.48	+3.85 -4.00	+3.48 -3.61	+3.18 -3.30
115.2kbps	+4.57 -5.88	+3.97 -5.08	+3.51 -4.48	+3.15 -4.00	+2.86 -3.61	+2.61 -3.30

Table 25-5. Maximum Total Baudrate Error in Double Speed Mode

Baudrate	Frame Size					
	5	6	7	8	9	10
0 - 57.6kbps	+5.66 -4.00	+4.92 -3.45	+4.35 -3.03	+3.90 -2.70	+3.53 -2.44	+3.23 -2.22
76.8kbps	+5.59 -4.00	+4.85 -3.45	+4.29 -3.03	+3.85 -2.70	+3.48 -2.44	+3.18 -2.22
115.2kbps	+4.57 -4.00	+3.97 -3.45	+3.51 -3.03	+3.15 -2.70	+2.86 -2.44	+2.61 -2.22

25.10. Multi-Processor Communication Mode

Setting the Multi-Processor Communication mode (MPCMn) bit in UCSRnA enables a filtering function of incoming frames received by the USART Receiver. Frames that do not contain address information will be ignored and not put into the receive buffer. This effectively reduces the number of incoming frames that has to be handled by the CPU, in a system with multiple MCUs that communicate via the same serial bus. The Transmitter is unaffected by the MPCMn setting, but has to be used differently when it is a part of a system utilizing the Multi-processor Communication mode.

Baud Rate [bps]	f_{osc} = 3.6864MHz				f_{osc} = 4.0000MHz				f_{osc} = 7.3728MHz			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error
250k	0	-7.8%	1	-7.8%	0	0.0%	1	0.0%	1	-7.8%	3	-7.8%
0.5M	–	–	0	-7.8%	–	–	0	0.0%	0	-7.8%	1	-7.8%
1M	–	–	–	–	–	–	–	–	–	–	0	-7.8%
Max.(1)	230.4kbps		460.8kbps		250kbps		0.5Mbps		460.8kbps		921.6kbps	

(1) UBRRn = 0, Error = 0.0%

Table 25-8. Examples of UBRRn Settings for Commonly Used Oscillator Frequencies

Baud Rate [bps]	f_{osc} = 8.0000MHz				f_{osc} = 11.0592MHz				f_{osc} = 14.7456MHz			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error
2400	207	0.2%	416	-0.1%	287	0.0%	575	0.0%	383	0.0%	767	0.0%
4800	103	0.2%	207	0.2%	143	0.0%	287	0.0%	191	0.0%	383	0.0%
9600	51	0.2%	103	0.2%	71	0.0%	143	0.0%	95	0.0%	191	0.0%
14.4k	34	-0.8%	68	0.6%	47	0.0%	95	0.0%	63	0.0%	127	0.0%
19.2k	25	0.2%	51	0.2%	35	0.0%	71	0.0%	47	0.0%	95	0.0%
28.8k	16	2.1%	34	-0.8%	23	0.0%	47	0.0%	31	0.0%	63	0.0%
38.4k	12	0.2%	25	0.2%	17	0.0%	35	0.0%	23	0.0%	47	0.0%
57.6k	8	-3.5%	16	2.1%	11	0.0%	23	0.0%	15	0.0%	31	0.0%
76.8k	6	-7.0%	12	0.2%	8	0.0%	17	0.0%	11	0.0%	23	0.0%
115.2k	3	8.5%	8	-3.5%	5	0.0%	11	0.0%	7	0.0%	15	0.0%
230.4k	1	8.5%	3	8.5%	2	0.0%	5	0.0%	3	0.0%	7	0.0%
250k	1	0.0%	3	0.0%	2	-7.8%	5	-7.8%	3	-7.8%	6	5.3%
0.5M	0	0.0%	1	0.0%	–	–	2	-7.8%	1	-7.8%	3	-7.8%
1M	–	–	0	0.0%	–	–	–	–	0	-7.8%	1	-7.8%
Max.(1)	0.5Mbps		1Mbps		691.2kbps		1.3824Mbps		921.6kbps		1.8432Mbps	

(1) UBRRn = 0, Error = 0.0%

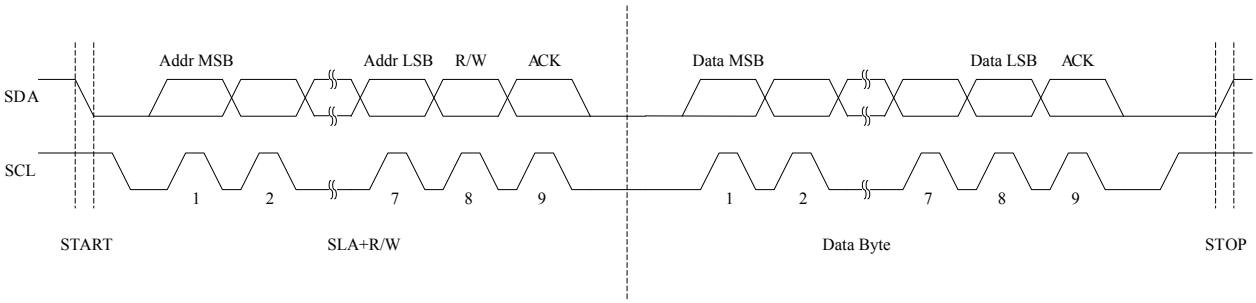
Table 25-9. Examples of UBRRn Settings for Commonly Used Oscillator Frequencies

Baud Rate [bps]	f_{osc} = 16.0000MHz				f_{osc} = 18.4320MHz				f_{osc} = 20.0000MHz			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error
2400	416	-0.1%	832	0.0%	479	0.0%	959	0.0%	520	0.0%	1041	0.0%
4800	207	0.2%	416	-0.1%	239	0.0%	479	0.0%	259	0.2%	520	0.0%
9600	103	0.2%	207	0.2%	119	0.0%	239	0.0%	129	0.2%	259	0.2%
14.4k	68	0.6%	138	-0.1%	79	0.0%	159	0.0%	86	-0.2%	173	-0.2%
19.2k	51	0.2%	103	0.2%	59	0.0%	119	0.0%	64	0.2%	129	0.2%
28.8k	34	-0.8%	68	0.6%	39	0.0%	79	0.0%	42	0.9%	86	-0.2%
38.4k	25	0.2%	51	0.2%	29	0.0%	59	0.0%	32	-1.4%	64	0.2%
57.6k	16	2.1%	34	-0.8%	19	0.0%	39	0.0%	21	-1.4%	42	0.9%
76.8k	12	0.2%	25	0.2%	14	0.0%	29	0.0%	15	1.7%	32	-1.4%
115.2k	8	-3.5%	16	2.1%	9	0.0%	19	0.0%	10	-1.4%	21	-1.4%
230.4k	3	8.5%	8	-3.5%	4	0.0%	9	0.0%	4	8.5%	10	-1.4%
250k	3	0.0%	7	0.0%	4	-7.8%	8	2.4%	4	0.0%	9	0.0%
0.5M	1	0.0%	3	0.0%	–	–	4	-7.8%	–	–	4	0.0%
1M	0	0.0%	1	0.0%	–	–	–	–	–	–	–	–
Max.(1)	1Mbps		2Mbps		1.152Mbps		2.304Mbps		1.25Mbps		2.5Mbps	

(1) UBRRn = 0, Error = 0.0%

25.12. Register Description

Figure 27-6. Typical Data Transmission



27.4. Multi-master Bus Systems, Arbitration and Synchronization

The TWI protocol allows bus systems with several masters. Special concerns have been taken in order to ensure that transmissions will proceed as normal, even if two or more masters initiate a transmission at the same time. Two problems arise in multi-master systems:

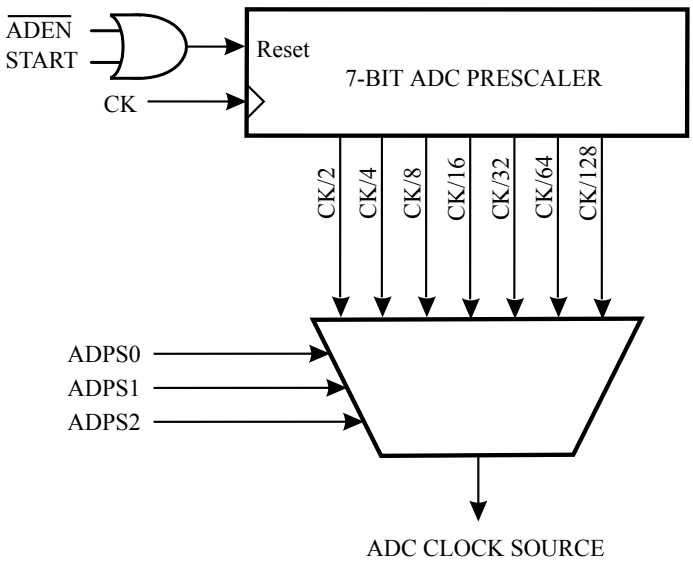
- An algorithm must be implemented allowing only one of the masters to complete the transmission. All other masters should cease transmission when they discover that they have lost the selection process. This selection process is called arbitration. When a contending master discovers that it has lost the arbitration process, it should immediately switch to Slave mode to check whether it is being addressed by the winning master. The fact that multiple masters have started transmission at the same time should not be detectable to the slaves, i.e. the data being transferred on the bus must not be corrupted.
- Different masters may use different SCL frequencies. A scheme must be devised to synchronize the serial clocks from all masters, in order to let the transmission proceed in a lockstep fashion. This will facilitate the arbitration process.

The wired-ANDing of the bus lines is used to solve both these problems. The serial clocks from all masters will be wired-ANDed, yielding a combined clock with a high period equal to the one from the Master with the shortest high period. The low period of the combined clock is equal to the low period of the Master with the longest low period. Note that all masters listen to the SCL line, effectively starting to count their SCL high and low time-out periods when the combined SCL line goes high or low, respectively.

If Auto Triggering is enabled, single conversions can be started by writing ADCSRA.ADSC to '1'. ADSC can also be used to determine if a conversion is in progress. The ADSC bit will be read as '1' during a conversion, independently of how the conversion was started.

29.4. Prescaling and Conversion Timing

Figure 29-3. ADC Prescaler



By default, the successive approximation circuitry requires an input clock frequency between 50kHz and 200kHz to get maximum resolution. If a lower resolution than 10 bits is needed, the input clock frequency to the ADC can be higher than 200kHz to get a higher sample rate.

The ADC module contains a prescaler, which generates an acceptable ADC clock frequency from any CPU frequency above 100kHz. The prescaling is selected by the ADC Prescaler Select bits in the ADC Control and Status Register A (ADCSRA.ADPS). The prescaler starts counting from the moment the ADC is switched on by writing the ADC Enable bit ADCSRA.ADEN to '1'. The prescaler keeps running for as long as ADEN=1, and is continuously reset when ADEN=0.

When initiating a single ended conversion by writing a '1' to the ADC Start Conversion bit (ADCSRA.ADSC), the conversion starts at the following rising edge of the ADC clock cycle.

A normal conversion takes 13 ADC clock cycles. The first conversion after the ADC is switched on (i.e., ADCSRA.ADEN is written to '1') takes 25 ADC clock cycles in order to initialize the analog circuitry.

When the bandgap reference voltage is used as input to the ADC, it will take a certain time for the voltage to stabilize. If not stabilized, the first value read after the first conversion may be wrong.

The actual sample-and-hold takes place 1.5 ADC clock cycles after the start of a normal conversion and 13.5 ADC clock cycles after the start of an first conversion. When a conversion is complete, the result is written to the ADC Data Registers (ADCL and ADCH), and the ADC Interrupt Flag (ADCSRA.ADIF) is set. In Single Conversion mode, ADCSRA.ADSC is cleared simultaneously. The software may then set ADCSRA.ADSC again, and a new conversion will be initiated on the first rising ADC clock edge.

When Auto Triggering is used, the prescaler is reset when the trigger event occurs. This assures a fixed delay from the trigger event to the start of conversion. In this mode, the sample-and-hold takes place two ADC clock cycles after the rising edge on the trigger source signal. Three additional CPU clock cycles are used for synchronization logic.

Note: The EEPRPOM memory is preserved during Chip Erase if the EESAVE Fuse is programmed.

Load Command “Chip Erase”:

1. Set XA1, XA0 to “10”. This enables command loading.
2. Set BS1 to “0”.
3. Set DATA to “1000 0000”. This is the command for Chip Erase.
4. Give XTAL1 a positive pulse. This loads the command.
5. Give $\overline{WR}$ a negative pulse. This starts the Chip Erase. RDY/$\overline{BSY}$ goes low.
6. Wait until RDY/$\overline{BSY}$ goes high before loading a new command.

33.7.4. Programming the Flash

The Flash is organized in pages as number of Words in a Page and number of Pages in the Flash. When programming the Flash, the program data is latched into a page buffer. This allows one page of program data to be programmed simultaneously. The following procedure describes how to program the entire Flash memory:

Step A. Load Command “Write Flash”

1. Set XA1, XA0 to “10”. This enables command loading.
2. Set BS1 to “0”.
3. Set DATA to “0001 0000”. This is the command for Write Flash.
4. Give XTAL1 a positive pulse. This loads the command.

Step B. Load Address Low Byte

1. Set XA1, XA0 to “00”. This enables address loading.
2. Set BS1 to “0”. This selects low address.
3. Set DATA = Address low byte (0x00 - 0xFF).
4. Give XTAL1 a positive pulse. This loads the address low byte.

Step C. Load Data Low Byte

1. Set XA1, XA0 to “01”. This enables data loading.
2. Set DATA = Data low byte (0x00 - 0xFF).
3. Give XTAL1 a positive pulse. This loads the data byte.

Step D. Load Data High Byte

1. Set BS1 to “1”. This selects high data byte.
2. Set XA1, XA0 to “01”. This enables data loading.
3. Set DATA = Data high byte (0x00 - 0xFF).
4. Give XTAL1 a positive pulse. This loads the data byte.

Step E. Latch Data

1. Set BS1 to “1”. This selects high data byte.
2. Give PAGEL a positive pulse. This latches the data bytes. (Please refer to the figure, Programming the Flash Waveforms, in this section for signal waveforms)

Symbol	Parameter	V_{CC}= 1.8 - 5.5V		V_{CC}= 2.7 - 5.5V		V_{CC}= 4.5 - 5.5V		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
t_{CLCH}	Rise Time	-	2.0	-	1.6	-	0.5	µs
t_{CHCL}	Fall Time	-	2.0	-	1.6	-	0.5	µs
Δt_{CLCL}	Change in period from one clock cycle to the next	-	2	-	2	-	2	%

34.5. System and Reset Characteristics

Table 34-7. Reset, Brown-out and Internal Voltage Characteristics(1)

Symbol	Parameter	Condition	Min.	Typ	Max	Units
V_{POT}	Power-on Reset Threshold Voltage (rising)		1.1	1.4	1.6	V
	Power-on Reset Threshold Voltage (falling)(2)		0.6	1.3	1.6	V
SR_{ON}	Power-on Slope Rate		0.01	-	10	V/ms
V_{RST}	RESET Pin Threshold Voltage		0.2 V_{CC}	-	0.9 V_{CC}	V
t_{RST}	Minimum pulse width on RESET Pin		-	-	2.5	µs
V_{HYST}	Brown-out Detector Hysteresis		-	50	-	mV
t_{BOD}	Min. Pulse Width on Brown-out Reset		-	2	-	µs
V_{BG}	Bandgap reference voltage	V_{CC}=2.7 T_A=25°C	1.0	1.1	1.2	V
t_{BG}	Bandgap reference start-up time	V_{CC}=2.7 T_A=25°C	-	40	70	µs
I_{BG}	Bandgap reference current consumption	V_{CC}=2.7 T_A=25°C	-	10	-	µA

Note:

1. Values are guidelines only.
2. The Power-on Reset will not work unless the supply voltage has been below VPOT (falling)

Table 34-8. BODLEVEL Fuse Coding

BODLEVEL [2:0] Fuses	Min. V_{BOT}	Typ V_{BOT}	Max V_{BOT}	Units
111	BOD Disabled			
110	1.7	1.8	2.0	V
101	2.5	2.7	2.9	
100	4.1	4.3	4.5	

35.2.12. Current Consumption of Peripheral Units

Figure 35-89. ATmega168PB: ADC Current vs. V_{CC} (AREF = AV_{CC})

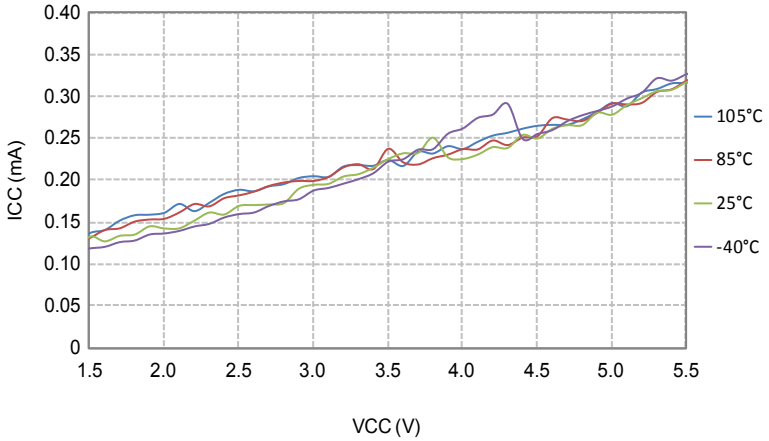


Figure 35-90. ATmega168PB: Analog Comparator Current vs. V_{CC}

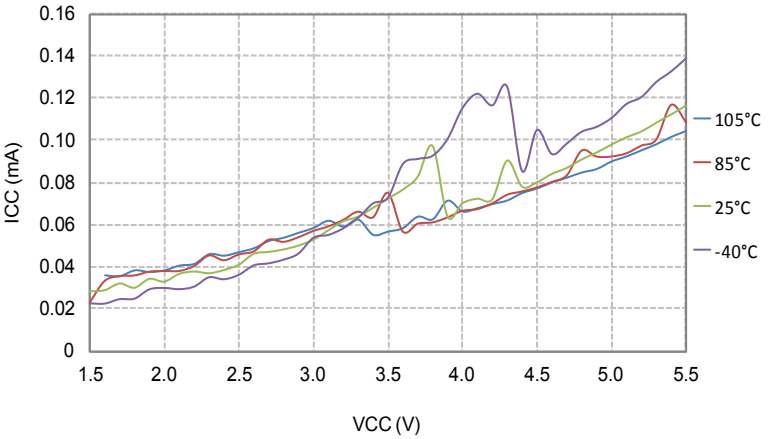


Figure 35-91. ATmega168PB: AREF External Reference Current vs. V_{CC}

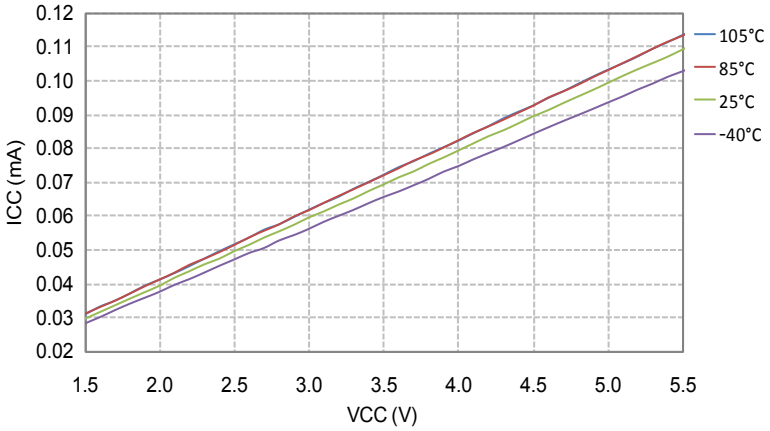


Figure 35-92. ATmega168PB: Brownout Detector Current vs. V_{CC}

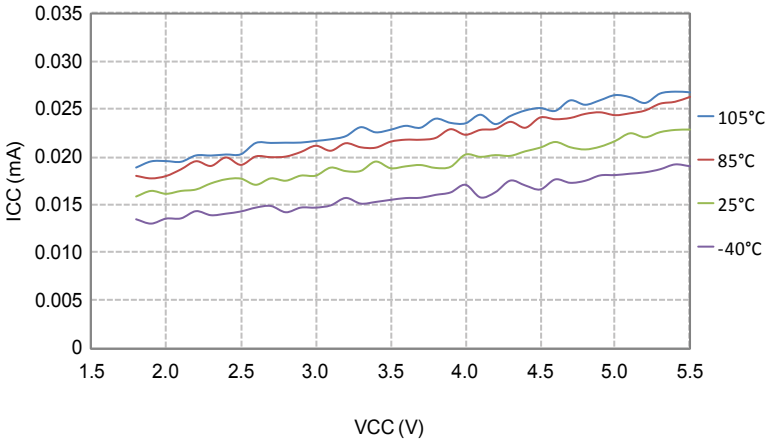
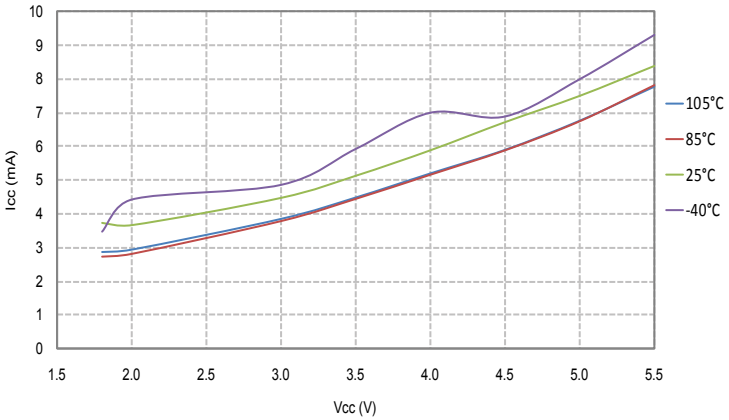


Figure 35-93. ATmega168PB: Programming Current vs. V_{CC}



36. Register Summary

Offset	Name	Bit Pos.								
0x23	PINB	7:0	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
0x24	DDRB	7:0	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
0x25	PORTB	7:0	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
0x26	PINC	7:0		PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
0x27	DDRC	7:0		DDRC6	DDRC5	DDRC4	DDRC3	DDRC2	DDRC1	DDRC0
0x28	PORTC	7:0		PORTC6	PORTC5	PORTC4	PORTC3	PORTC2	PORTC1	PORTC0
0x29	PIND	7:0	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
0x2A	DDRD	7:0	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
0x2B	PORTD	7:0	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
0x2C	PINE	7:0					PINE3	PINE2	PINE1	PINE0
0x2D	DDRE	7:0					DDRE3	DDRE2	DDRE1	DDRE0
0x2E	PORTE	7:0					PORTE3	PORTE2	PORTE1	PORTE0
0x2F ... 0x34	Reserved									
0x35	TIFR0	7:0						OCF0B	OCF0A	TOV0
0x36	TIFR1	7:0			ICF1			OCF1B	OCF1A	TOV1
0x37	TIFR2	7:0						OCF2B	OCF2A	TOV2
0x38 ... 0x3A	Reserved									
0x3B	PCIFR	7:0						PCIF2	PCIF1	PCIF0
0x3C	EIFR	7:0							INTF1	INTF0
0x3D	EIMSK	7:0							INT1	INT0
0x3E	GPIOR0	7:0	GPIOR0[7:0]							
0x3F	EECR	7:0			EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE
0x40	EEDR	7:0	EEDR[7:0]							
0x41	EEARL	7:0	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0
0x42	EEARH	7:0							EEAR9	EEAR8
0x43	GTCCR	7:0	TSM						PSRASY	PSRSYNC
0x44	TCCR0A	7:0	COM0A1	COM0A0	COM0B1	COM0B0			WGM01	WGM00
0x45	TCCR0B	7:0	FOC0A	FOC0B			WGM02	CS02	CS01	CS00
0x46	TCNT0	7:0	TCNT0[7:0]							
0x47	OCR0A	7:0	OCR0A[7:0]							
0x48	OCR0B	7:0	OCR0B[7:0]							
0x49	Reserved									
0x4A	GPIOR1	7:0	GPIOR1[7:0]							
0x4B	GPIOR2	7:0	GPIOR2[7:0]							
0x4C	SPCR	7:0	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
0x4D	SPSR	7:0	SPIF	WCOL						SPI2X
0x4E	SPDR	7:0	SPID[7:0]							
0x4F	ACSR0	7:0								ACOE
0x50	ACSR	7:0	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0
0x51	DWDR	7:0	DWDR[7:0]							