Features

- **AVR® 8-bit RISC Microcontroller with 83 ns Instruction Cycle Time**
- **USB Hub with One Attached and Two External Ports**
- **USB Keyboard Function with Three Programmable Endpoints**
- **16 KB Program Memory, 512 Bytes Data SRAM**
- **32 x 8 General-purpose Working Registers**
- **32 Programmable I/O Port Pins**
- **Support for 18 x 8 Keyboard Matrix**
- **Keyboard Scan Inputs with Pull-up Resistor**
- **Four LED Driver Outputs**
- **One 8-bit Timer/Counter with Separate Pre-scaler**
- **External and Internal Interrupt Sources**
- **Programmable Watchdog Timer**
- **6 MHz Oscillator with On-chip PLL**
- **5V Operation with On-chip 3.3V Power Supply**
- **48-lead LQFP Package**

Description

The Atmel AT43USB326 is an 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the AT43USB326 achieves throughputs approaching 12 MIPS. The AVR core combines a rich instruction set with 32 general-purpose working registers. All 32 registers are directly connected to the 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.

Furthermore, the AT43USB326 features an on-chip 16-Kbyte program memory and 512 bytes of data memory. It is supported by a standard set of peripherals such as timer/counter modules, watchdog timer and internal and external interrupt sources. The major peripheral included in the AT43USB326 is the USB Hub with an embedded keyboard controller function.

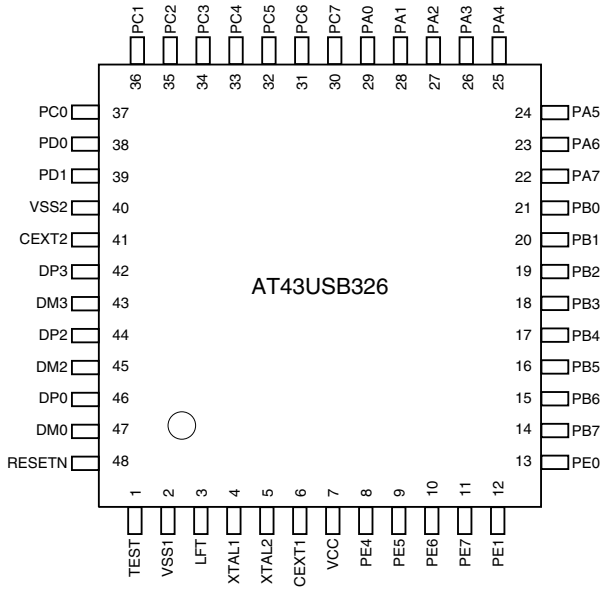
Multimedia USB Keyboard Controller with Embedded Hub

AT43USB326

3313D–USB–04/06

Pin Configuration

Figure 1. AT43USB326 48-lead LQFP



Pin Assignment

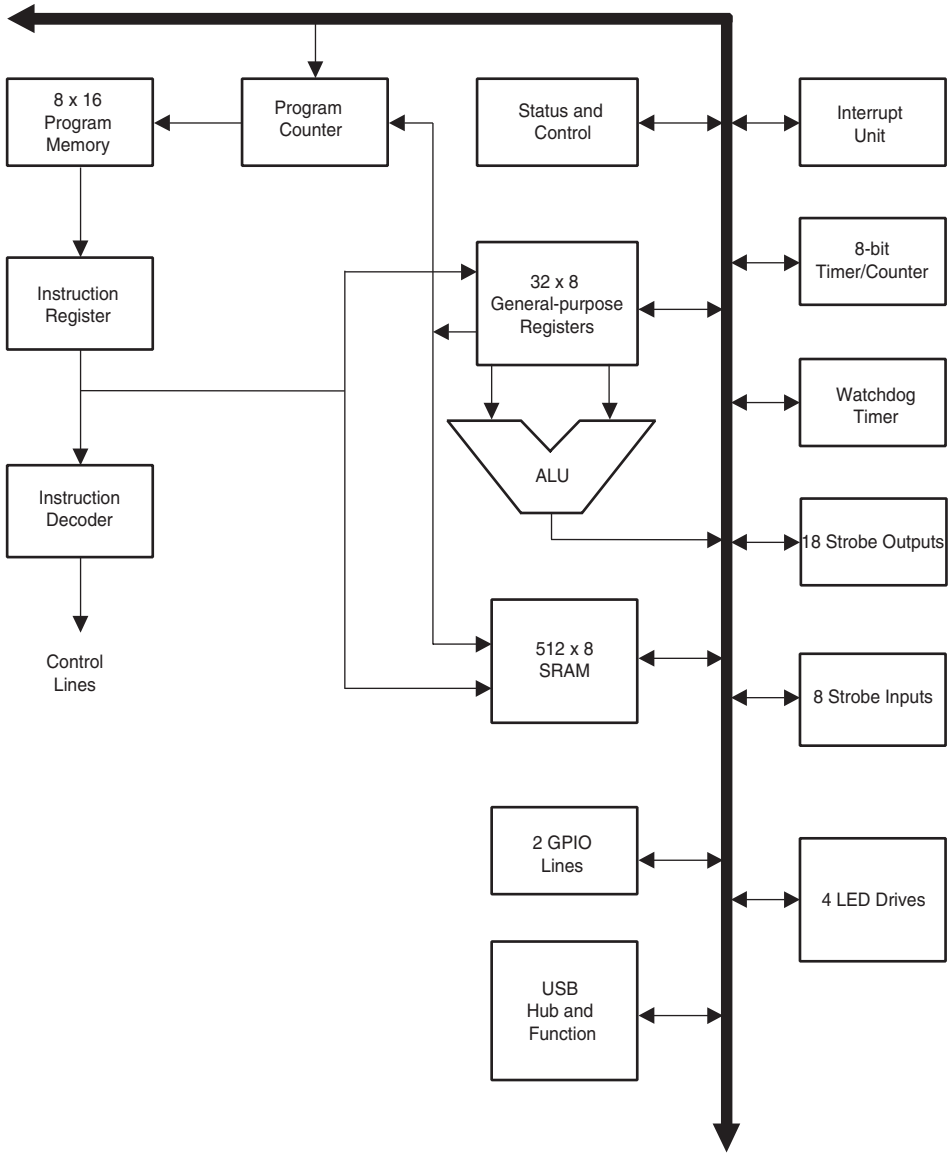
Pin#	Signal	Type
1	TEST	Input
2	VSS1	Power Supply/Ground
3	LFT	Output
4	XTAL1	Input
5	XTAL2	Output
6	CEXT1	Power Supply/Ground
7	VCC	Power Supply/Ground
8	PE4	Bi-directional
9	PE5	Bi-directional
10	PE6	Bi-directional
11	PE7	Bi-directional
12	PE1/COL17	Bi-directional
13	PE0/COL16	Bi-directional
14	PB7/COL15	Bi-directional
15	PB6/COL14	Bi-directional
16	PB5/COL13	Bi-directional
17	PB4/COL12	Bi-directional
18	PB3/COL11	Bi-directional
19	PB2/COL10	Bi-directional
20	PB1/COL9	Bi-directional
21	PB0/COL8	Bi-directional
22	PA7/COL7	Bi-directional
23	PA6/COL6	Bi-directional
24	PA5/COL5	Bi-directional

Pin#	Signal	Type
25	PA4/CL4	Bi-directional
26	PA3/CL3	Bi-directional
27	PA2/CL2	Bi-directional
28	PA1/CL1	Bi-directional
29	PA0/CL0	Bi-directional
30	PC7/ROW7	Bi-directional
31	PC6/ROW6	Bi-directional
32	PC5/ROW5	Bi-directional
33	PC4/ROW4	Bi-directional
34	PC3/ROW3	Bi-directional
35	PC2/ROW2	Bi-directional
36	PC1/ROW1	Bi-directional
37	PC0/ROW0	Bi-directional
38	PD0	Bi-directional
39	PD1	Bi-directional
40	VSS2	Power Supply/Ground
41	CEXT2	Power Supply/Ground
42	DP3	Bi-directional
43	DM3	Bi-directional
44	DP2	Bi-directional
45	DM2	Bi-directional
46	DP0	Bi-directional
47	DM0	Bi-directional
48	RESETN	Bi-directional

Signal Description

Name	Type	Function
V_{CC}	Power Supply/Ground	**5V Digital Power Supply**
$V_{SS1, 2}$	Power Supply/Ground	**Ground**
CEXT1, 2	Power Supply/Ground	**External Capacitors for Power Supplies** – High quality 2.2 µF capacitors must be connected to CEXT1 and 2 for proper operation of the chip.
XTAL1	Input	**Oscillator Input** – Input to the inverting oscillator amplifier.
XTAL2	Output	**Oscillator Output** – Output of the inverting oscillator amplifier.
LFT	Input	**PLL Filter** – For proper operation of the PLL, this pin should be connected through a 0.01 µF capacitor in parallel with a 100Ω resistor in series with a 0.1 µF capacitor to ground (VSS). Both capacitors must be high quality ceramic.
DPO	Bi-directional	**Upstream Plus USB I/O** – This pin should be connected to CEXT1 through an external 1.5 kΩ.
DMO	Bi-directional	**Upstream Minus USB I/O**
DP[2,3]	Bi-directional	**Downstream Plus USB I/O** – Each of these pins should be connected to VSS through an external 15 kΩ resistor. DP[2,3] and DM[2,3] are the differential signal pin pairs to connect downstream USB devices.
DM[2,3]	Bi-directional	**Downstream Minus USB I/O** – Each of these pins should be connected to VSS through an external 15 kΩ resistor.
PA[0:7]	Bi-directional	**Port A[0:7]** – Bi-directional 8-bit I/O port with controlled slew rate. These pins are used as eight of the keyboard matrix column output strobes. PA[0:7] = COL[0:7].
PB[0:7]	Bi-directional	**Port B[0:7]** – Bi-directional 8-bit I/O port controlled slew rate. These pins are used as the eight of the keyboard matrix column output strobes: PB[0:7] = COL[8:15].
PC[0:7]	Bi-directional	**Port C[0:7]** – Bi-directional 8-bit I/O port with internal pull-ups. These pins are used as keyboard matrix row input signals. PC[0:7] = ROW [0:7].
PD[0:1]	Bi-directional	**Port D[0:1]** – Bi-directional I/O ports.
PE[0:1]	Bi-directional	**Port E[0:1]** – Bi-directional I/O port with controlled slew rate which can be used as two additional keyboard column output strobes, COL 16, 17.
PE[4:7]	Bi-directional	**PE[4:7]** have built-in series limiting resistors and can be used to drive LEDs directly.
TEST	Input	**Test Pin** – This pin should be tied to ground.
RESETN	Input	**Reset** – Active Low.

Figure 2. The AT43USB326 Enhanced RISC Architecture with USB Keyboard Controller and Hub



Architectural Overview

The AT43USB326 is a USB microcontroller with special peripherals for use as a programmable keyboard controller.

The peripherals and features of the AT43USB326 microcontroller are similar to those of the AT90S8515, with the exception of the following modifications:

- A masked ROM for program memory
- No EEPROM
- No external data memory accesses
- No UART
- No SPI
- No 16-bit timer/counter
- Idle mode not supported
- USB Hub with attached function

The embedded USB hardware of the AT43USB326 is a compound device, consisting of a 3 port hub with a permanently attached function on one port. The hub and attached function are two independent USB devices, each having its own device addresses and control endpoints. The hub has its dedicated interrupt endpoint, while the USB function has two additional programmable endpoints with 8-byte FIFOs.

The microcontroller always runs from a 12 MHz clock that is generated by the USB hardware. While the nominal and average period of this clock is 83.3 ns, it may have single cycles that deviate by ±20.8 ns during a phase adjustment by the SIE's clock/data separator of the USB hardware.

The microcontroller shares most of the control and status registers of the megaAVR Microcontroller Family. The registers for managing the USB operations are mapped into its SRAM space. The I/O section on page 14 summarizes the available I/O registers. The “AVR Register Set” on page 34 covers the AVR registers. Please refer to the Atmel AVR manual for more information.

The fast-access register file contains 32 x 8-bit general-purpose working registers with a single clock cycle access time. This means that during one single clock cycle, one Arithmetic Logic Unit (ALU) operation is executed. Two operands are output from the register file, the operation is executed, and the result is stored back in the register file – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing - enabling efficient address calculations. One of the three address pointers is also used as the address pointer for look-up tables in program memory. These added function registers are the 16-bit X-, Y- and Z-registers.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations are also executed in the ALU. Figure 2 on page 5 shows the AT43USB326 AVR Enhanced RISC microcontroller architecture.

In addition to the register operation, the conventional memory addressing modes can be used on the register file as well. This is enabled by the fact that the register file is assigned the 32 lowest Data Space addresses ($00 - $1 F), allowing them to be accessed as though they were ordinary memory locations.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, Timer/Counters, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the register file, $20 - $5F.

The AVR uses a Harvard architecture concept – with separate memories and buses for program and data. The program memory is executed with a single-level pipelining. While one

instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is a downloadable SRAM or a mask programmed ROM.

With the relative jump and call instructions, the whole 24K address space is directly accessed. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the stack. The stack is effectively allocated in the general data SRAM, and consequently, the stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the Stack Pointer (SP) in the reset routine (before subroutines or interrupts are executed). The 10-bit SP is read/write accessible in the I/O space.

The 1-Kbyte data SRAM can be easily accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps. A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the status register. All interrupts have a separate interrupt vector in the interrupt vector table at the beginning of the program memory. The interrupts have priority in accordance with their interrupt vector position. The lower the interrupt vector address, the higher the priority.

The General-purpose Register File

Table 1. AVR CPU General-purpose Working Register

Register	Address	Comment
R0	$00	
R1	$01	
R2	$02	
..		
R13	$0D	
R14	$0E	
R15	$0F	
R16	$10	
R17	$11	
..		
R26	$1A	X-register low byte
R27	$1B	X-register high byte
R28	$1C	Y-register low byte
R29	$1D	Y-register high byte
R30	$1E	Z-register low byte
R31	$1F	Z-register high byte

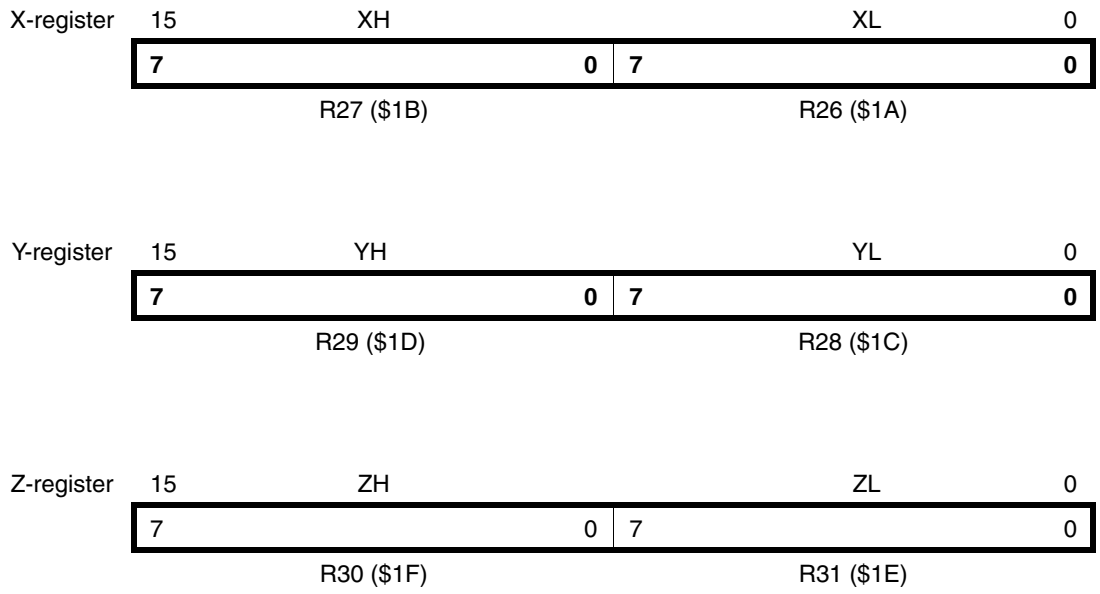
All register operating instructions in the instruction set have direct and single cycle access to all registers. The only exception is the five constant arithmetic and logic instructions SBCI, SUBI, CPI, ANDI, and ORI between a constant and a register, and the LDI instruction for load

immediate constant data. These instructions apply to the second half of the registers in the register file – R16..R31. The general SBC, SUB, CP, AND, and OR and all other operations between two registers or on a single register apply to the entire register file.

As shown in Table 1, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y-, and Z-registers can be set to index any register in the file.

X-, Y- and Z-Registers

Registers R26..R31 contain some added functions to their general-purpose usage. These registers are address pointers for indirect addressing of the Data Space. The three indirect address registers X, Y, and Z are defined as:



In the different addressing modes these address registers have functions as fixed displacement, automatic increment and decrement (see the descriptions for the different instructions).

ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all 32 general-purpose working registers. Within a single clock cycle, ALU operations between registers in the register file are executed. The ALU operations are divided into three main categories – arithmetic, logical and bit-functions.

Program Memory

The AT43USB326 contains 16K bytes on-chip masked programmable ROM. Since all instructions are 16- or 32-bit words, the program memory is organized as 8K x 16. The AT43USB326 Program Counter (PC) is 13 bits wide, thus addressing the 8,192 program memory addresses.

Constant tables can be allocated within the entire program memory address space (see the LPM - Load Program Memory instruction description).

SRAM Data Memory

Table 3 summarizes how the AT43USB326 SRAM Memory is organized. The lower 608 Data Memory locations address the Register file, the I/O Memory and the internal data SRAM. The first 96 locations address the Register File + I/O Memory, and the next 512 locations address the internal data SRAM. The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement and Indirect with Post-increment. In the register file, registers R26 to R31 feature the indirect addressing pointer registers. Direct addressing reaches the entire data space.

The Indirect with Displacement mode features 63 address locations that reach from the base address given by the Y- or Z-register.

When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers X, Y, and Z are decremented and incremented.

The 32 general-purpose working registers, 64 I/O registers and the 512 bytes of internal data SRAM in the AT43USB326 are all accessible through these addressing modes.

To manage the USB hardware, a special set of registers is assigned. These registers are mapped to SRAM space between addresses $1F00 and 1FFF. Table 3 and Table 4 give an overview of these registers.

Table 2. SRAM Organization

Register File		Data Address Space
R0		$0000
R1		$0001
R30		$001E
R31		$001F

I/O Registers

$00		$0020
$01		$0021
$3E		$005E
$3F		$005F

Internal SRAM

$0060
$0061
$025E
$045F

USB Registers

$1F00
$1FFE
$1FFF

Table 3. USB Hub and Function Registers

Address	Name	Function
$1FFD	FRM_NUM_H	Frame Number High Register
$1FFC	FRM_NUM_L	Frame Number Low Register
$1FFB	GLB_STATE	Global State Register
$1FFA	SPRSR	Suspend/Resume Register
$1FF9	SPRSIE	Suspend/Resume Interrupt Enable Register
$1FF8	SPRSMSK	Suspend/Resume Interrupt Mask Register
$1FF7	UISR	USB Interrupt Status Register
$1FF6	UIMSKR	USB Interrupt Mask Register
$1FF5	UIAR	USB Interrupt Acknowledge Register
$1FF3	UIER	USB Interrupt Enable Register
$1FF2	UOVCER	Overcurrent Detect Register
$1FEF	HADDR	Hub Address Register
$1FEE	FADDR	Function Address Register
$1FE7	HENDP0_CNTR	Hub Endpoint 0 Control Register
$1FE5	FENDP0_CNTR	Function Endpoint 0 Control Register
$1FE4	FENDP1_CNTR	Function Endpoint 1 Control Register
$1FE3	FENDP2_CNTR	Function Endpoint 2 Control Register
$1FDF	HCSR0	Hub Controller Endpoint 0 Service Routine Register
$1FDD	FCSR0	Function Controller Endpoint 0 Service Routine Register
$1FDC	FCSR1	Function Controller Endpoint 1 Service Routine Register
$1FDB	FCSR2	Function Controller Endpoint 2 Service Routine Register
$1FD7	HDR0	Hub Endpoint 0 FIFO Data Register
$1FD5	FDR0	Function Endpoint 0 FIFO Data Register
$1FD4	FDR1	Function Endpoint 1 FIFO Data Register
$1FD3	FDR2	Function Endpoint 2 FIFO Data Register
$1FCF	HBYTE_CNT0	Hub Endpoint 0 Byte Count Register
$1FCD	FBYTE_CNT0	Function Endpoint 0 Byte Count Register
$1FCC	FBYTE_CNT1	Function Endpoint 1 Byte Count Register
$1FCB	FBYTE_CNT2	Function Endpoint 2 Byte Count Register
$1FC7	HSTR	Hub Status Register
$1FC5	HPCON	Hub Port Control Register
$1FBA	HPSTAT3	Hub Port 3 Status Register
$1FB9	HPSTAT2	Hub Port 2 Status Register
$1FB8	HPSTAT1	Hub Port 1 Status Register
$1FB2	HPSCR3	Hub Port 3 Status Change Register

Table 3. USB Hub and Function Registers (Continued)

Address	Name	Function
$1FB1	HPSCR2	Hub Port 2 Status Change Register
$1FB0	HPSCR1	Hub Port 1 Status Change Register
$1FAA	PSTATE3	Hub Port 3 Bus State Register
$1FA9	PSTATE2	Hub Port 2 Bus State Register
$1FA7	HCAR0	Hub Endpoint 0 Control and Acknowledge Register
$1FA5	FCAR0	Function Endpoint 0 Control and Acknowledge Register
$1FA4	FCAR1	Function Endpoint 1 Control and Acknowledge Register
$1FA3	FCAR2	Function Endpoint 2 Control and Acknowledge Register

Table 4. USB Hub and Function Registers

Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
GLB_STATE	$1FFB	–	KB INT EN	–	SUSP FLG	RESUME FLG	RMWUPE	CONFG	HADD EN
SPRSR	$1FFA	–	–	–	–	–	FRWUP	RSM	GLB SUSP
SPRSIE	$1FF9	–	–	–	–	–	FRWUP IE	RSM IE	GLB SUSP IE
SPRSMSK	$1FF8	–	–	–	–	–	FRWUP MSK	RSM MSK	GLB SUSP MSK
UISR	$1FF7	SOF INT	EOF2 INT	–	–	HEP0 INT	FEP2 INT	FEP1 INT	FEP0 INT
UIMSKR	$1FF6	SOF MSK	SOF2 MSK	–	–	HEP0 MSK	FEP2 MSK	FEP1 MSK	FEP0 MSK
UIAR	$1FF5	SOF INTACK	EOF2 INTACK	–	–	HEP0 INTACK	FEP2 INTACK	FEP1 INTACK	FEP0 INTACK
UIER	$1FF3	SOF IE	EOF2 IE	–	–	HEP0 IE	FEP2 IE	FEP1 IE	FEP0 IE
UOVCER	$1FF2	–	–	–	–	–	OVC	–	–
HADDR	$1FEF	SAEN	HADD6	HADD5	HADD4	HADD3	HADD2	HADD1	HADD0
FADDR	$1FEE	FEN	FADD6	FADD5	FADD4	FADD3	FADD2	FADD1	FADD0
HENDP0_CNTR	$1FE7	EPEN	–	–	–	DTGLE	EPDIR	EPTYPE1	EPTYPE0
FENDP0_CNTR	$1FE5	EPEN	–	–	–	DTGLE	EPDIR	EPTYPE1	EPTYPE0
FENDP1_CNTR	$1FE4	EPEN	–	–	–	DTGLE	EPDIR	EPTYPE1	EPTYPE0
FENDP2_CNTR	$1FE3	EPEN	–	–	–	DTGLE	EPDIR	EPTYPE1	EPTYPE0
HCSR0	$1FDF	–	–	–	–	STALL SENT	RX SETUP	RX OUT PACKET	TX CEMPLETE
FCSR0	$1FDD	–	–	–	–	STALL SENT	RX SETUP	RX OUT PACKET	TX COMPLETE
FCSR1	$1FDC	–	–	–	–	STALL SENT	RX SETUP	RX OUT PACKET	TX COMPLETE
FCSR2	$1FDB	–	–	–	–	STALL SENT	RX SETUP	RX OUT PACKET	TX COMPLETE
HDR0	$1FD7	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
FDR0	$1FD5	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
FDR1	$1FD4	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
FDR2	$1FD3	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
HBYTE_CNT0	$1FCF	–	–	BYTCT5	BYTCT4	BYTCT3	BYTCT2	BYTCT1	BYTCT0
FBYTE_CNT0	$1FCD	–	–	BYTCT5	BYTCT4	BYTCT3	BYTCT2	BYTCT1	BYTCT0
FBYTE_CNT1	$1FCC	–	–	BYTCT5	BYTCT4	BYTCT3	BYTCT2	BYTCT1	BYTCT0
FBYTE_CNT2	$1FCB	–	–	BYTCT5	BYTCT4	BYTCT3	BYTCT2	BYTCT1	BYTCT0
HSTR	$1FC7	–	–	–	–	OVLSC	LPSC	OVI	LPS
HPCON	$1FC5	–	HPCON2	HPCON1	HPCON0	–	HPADD2	HPADD1	HPADD0
HPSTAT3	$1FBA	–	LSP	PPSTAT	PRSTAT	POCI	PSSTAT	PESTAT	PCSTAT
HPSTAT2	$1FB9	–	LSP	PPSTAT	PRSTAT	POCI	PSSTAT	PESTAT	PCSTAT
HPSTAT1	$1FB8	–	LSP	PPSTAT	PRSTAT	POCI	PSSTAT	PESTAT	PCSTAT
HPSCR3	$1FB2	–	–	–	RSTSC	POCIC	PSSC	PESC	PCSC
HPSCR2	$1FB1	–	–	–	RSTSC	POCIC	PSSC	PESC	PCSC
HPSCR1	$1FB0	–	–	–	RSTSC	POCIC	PSSC	PESC	PCSC
PSTATE3	$1FAA	–	–	–	–	–	–	DPSTATE	DMSTATE
PSTATE2	$1FA9	–	–	–	–	–	–	DPSTATE	DMSTATE
HCAR0	$1FA7	CTL DIR	DATA END	FORCE STALL	TX PACKET READY	STALL_SENT-ACK	RX_SETUP_ACK	RX_OUT_PACKET_ACK	TX_COMPLETE-ACK
FCAR0	$1FA5	CTL DIR	DATA END	FORCE STALL	TX PACKET READY	STALL_SENT-ACK	RX_SETUP_ACK	RX_OUT_PACKET_ACK	TX_COMPLETE-ACK
FCAR1	$1FA4	CTL DIR	DATA END	FORCE STALL	TX PACKET READY	STALL_SENT-ACK	RX_SETUP_ACK	RX_OUT_PACKET_ACK	TX_COMPLETE-ACK
FCAR2	$1FA3	CTL DIR	DATA END	FORCE STALL	TX PACKET READY	STALL_SENT-ACK	RX_SETUP_ACK	RX_OUT_PACKET_ACK	TX_COMPLETE-ACK

I/O Memory

The I/O space definition of the AT43USB326 is shown in the following table:

Table 5. I/O Memory Space

I/O (SRAM) Address	Name	Function
$3F ($5F)	SREG	Status Register
$3E ($5E)	SPH	Stack Pointer High
$3D ($5D)	SPL	Stack Pointer Low
$3B ($5B)	GIMSK	General Interrupt Mask Register
$3A ($5A)	GIFR	General Interrupt Flag Register
$39 ($59)	TIMSK	Timer/Counter Interrupt Mask Register
$38 ($58)	TIFR	Timer/Counter Interrupt Mask Register
$35 ($55)	MCUCR	MCU General Control Register
$33 ($53)	TCCR0	Timer/Counter0 Control Register
$32 ($52)	TCNT0	Timer/Counter0 (8 bit)
$21 ($41)	WDTCR	Watchdog Timer Counter Register
$1B ($4B)	PORTA	Data Register, Port A
$1A ($3A)	DDRA	Data Direction Register, Port A
$19 ($39)	PINA	Input Pins, Port A
$18 ($38)	PORTB	Data Register, Port B
$17 ($37)	DDRB	Data Direction Register, Port B
$16 ($36)	PINB	Input Pins, Port B
$12 ($32)	PORTD	Data Register, Port D
$11 ($31)	DDRD	Data Direction Register, Port D
$10 ($30)	PIND	Input Pins, Port D
$03 ($23)	PORTE	Data Register, Port E
$02 ($22)	DDRE	Data Direction Register, Port E
$01 ($21)	PINE	Input Pins, Port E

All AT43USB326 I/O and peripherals, except for the USB hardware registers, are placed in the I/O space. The I/O locations are accessed by the IN and OUT instructions transferring data between the 32 general-purpose working registers and the I/O space. 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. Refer to the instruction set documentations of the AVR for more details. When using the I/O specific commands, IN and OUT, the I/O address $00 – $3F must be used. When addressing I/O registers as SRAM, $20 must be added to this address. All I/O register addresses throughout this document are shown with the SRAM address in parentheses.

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

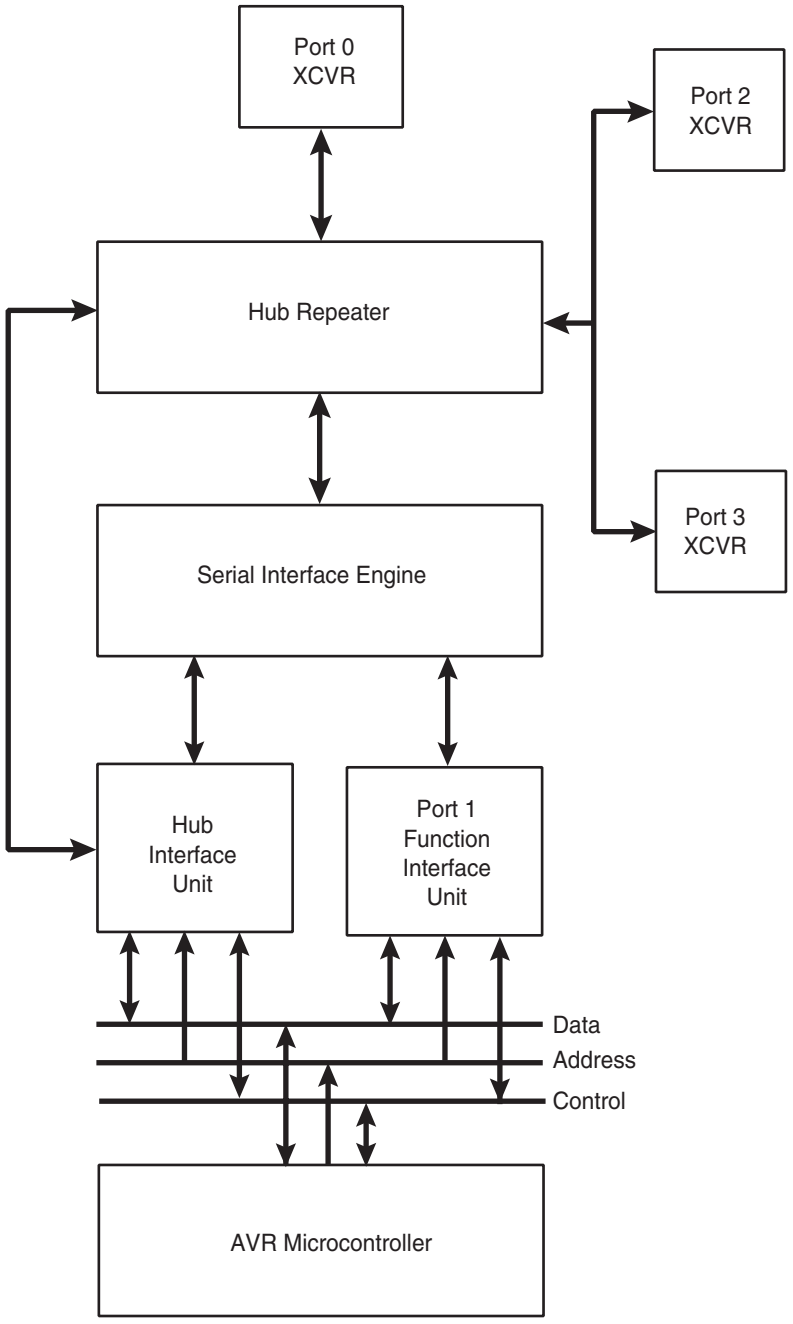
USB Hub

A block diagram of the USB hardware of the AT43USB326 is shown in Figure 3. The USB hub of the AT43USB326 has 3 downstream ports. The embedded function is permanently attached to Port 1. Ports 2 and 3 are available as external ports. The actual number of ports used is strictly defined by the firmware of the AT43USB326 and can vary from 0 to 2. Because the exact configuration is defined by firmware, ports 2 and 3 may even function as permanently attached ports as long as the Hub Descriptor identifies them as such.

USB Function

The embedded USB function has its own device address and has a default endpoint plus 2 other programmable endpoints with their own 8-byte FIFOs. Endpoints 1 and 2 can be programmed as interrupt IN or OUT or bulk IN or OUT endpoints.

Figure 3. USB Hardware



Functional Description

On-chip Power Supply

The AT43USB326 contains two on-chip power supplies that generate 3.3V with a capacity of 30 mA each from the 5V power input. The on-chip power supplies are intended to supply the AT43USB326 internal circuit and the 1.5K pull-up resistor only and should not be used for other purposes. External 2.2 µF filter capacitors are required at the power supply outputs, CEXT1 and CEXT2. The internal power supplies can be disabled as described in the next paragraph.

The user should be careful when the GPIO pins are required to supply high-load currents. If the application requires that the GPIO supply currents beyond the capability of the on-chip power supply, the AT43USB326 should be supplied by an external 3.3V power supply. In this case, the 5V V_{CC} power supply pin should be left unconnected and the 3.3V power supplied to the chip through the CEXT1 and CEXT2 pins.

I/O Pin Characteristics

The I/O pins of the AT43USB326 should not be directly connected to voltages less than V_{SS} or more than the voltage at the CEXT pins. If it is necessary to violate this rule, insert a series resistor between the I/O pin and the source of the external signal source that limits the current into the I/O pin to less than 2 mA. Under no circumstance should the external voltage exceed 5.5V. To do so will put the chip under excessive stress.

Oscillator and PLL

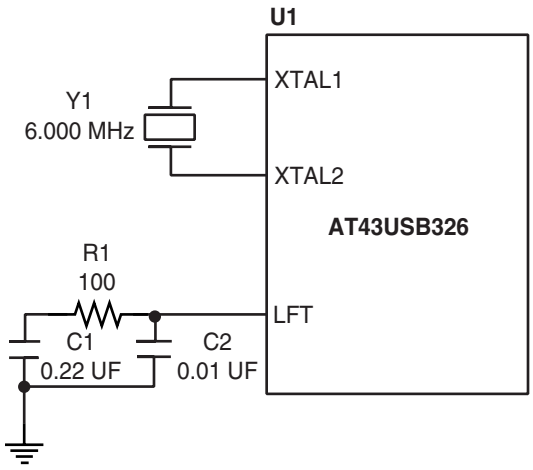
All clock signals required to operate the AT43USB326 are derived from an on-chip oscillator. To reduce EMI and power dissipation, the oscillator is designed to operate with a 6 MHz crystal. An on-chip PLL generates the high frequency for the clock/data separator of the Serial Interface Engine. In the suspended state, the oscillator circuitry is turned off.

The oscillator of the AT43USB326 is a special, low-drive type, designed to work with most crystals without any external components. The crystal must be of the parallel resonance type requiring a load capacitance of about 10 pF. If the crystal requires a higher value capacitance, external capacitors can be added to the two terminals of the crystal and ground to meet the required value. To assure quick start-up, a crystal with a high Q, or low ESR, should be used. To meet the USB hub frequency accuracy and stability requirements for hubs, the crystal should have an accuracy and stability of better than 100 PPM. The use of a ceramic resonator in place of the crystal is not recommended because a resonator would not have the necessary frequency accuracy and stability.

The clock can also be externally sourced. In this case, connect the clock source to the XTAL1 pin, while leaving XTAL2 pin floating. The switching level at the OSC1 pin can be as low as 0.47V and a CMOS device is required to drive this pin to maintain good noise margins at the low switching level.

For proper operation of the PLL, an external RC filter consisting of a series RC network of 100Ω and 0.1 µF in parallel with a 0.01 µF capacitor must be connected from the LFT pin to V_{SS}. Use only high-quality ceramic capacitors.

Figure 4. Oscillator and PLL



Reset and Interrupt Handling

The AT43USB326 provides 12 different interrupt sources with 4 separate reset vectors, each with a separate program vector in the program memory space. Nine of the interrupt sources share 2 interrupt reset vectors. These nine are the USB related interrupts. All interrupts are assigned individual enable bits which must be set (one) together with the I-bit in the status register in order to enable the interrupt.

The lowest addresses in the program memory space are automatically defined as the Reset and Interrupt vectors. The complete list of vectors is shown in Table 6. The list also determines the priority levels of the different interrupts. The lower the address, the higher is the priority level. RESET has the highest priority, and next is INT0 – the USB Suspend and Resume Interrupt, etc.

Table 6. Reset and Interrupt Vectors

Vector No.	Program Address	Source	Interrupt Definition
1	$000	RESET	External Reset, Power-on Reset and Watchdog Reset
2	$002	INT0	USB Suspend and Resume
8	$00E	TIMER0, OVF	Timer/Counter0 Overflow
13	$018	USB HW	USB Hardware

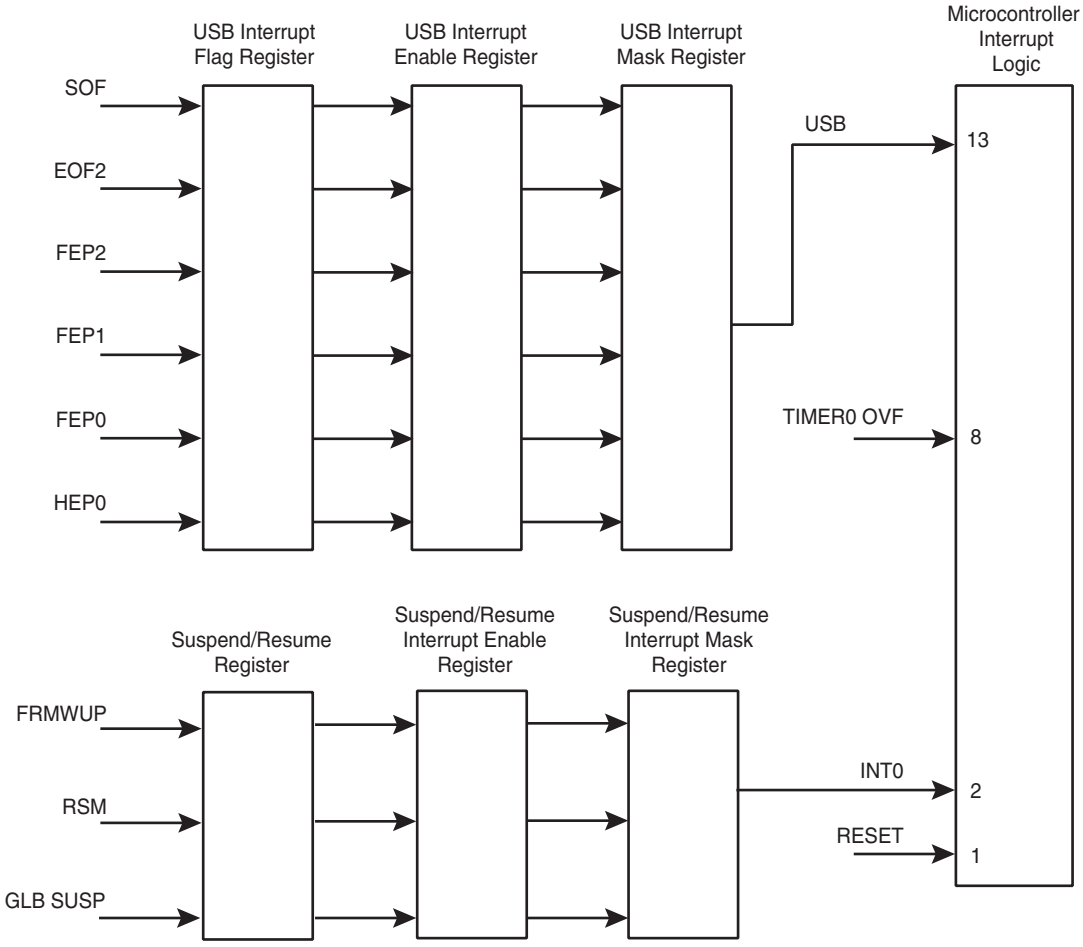
The most typical and general program setup for the Reset and Interrupt Vector Addresses are:

```
Address          Labels         Code                      Comments
$000                            jmp    RESET              ; Reset Handler
$002                            jmp    EXT_INT0           ; IRQ0 Handler
$00E                            jmp    TIM0_OVF           ; Timer0
Overflow
Handler
$018                            jmp    USB_HW             ; USB Handler
;
$00d             MAIN:          ldi r16, high (RAMEND)    ; Main Program
start
$00e                            out SPH, r16
$00f                            ldi r16, low (RAMEND)
$010                            out SPL, r16
$011                            <instr> xxx


...              ...      ...   ...
```

USB related interrupt events are routed to reset vectors 13 and 2 through a separate set of interrupt, interrupt enable and interrupt mask registers that are mapped to the data SRAM space. These interrupts must be enabled though their control register bits. In the event an interrupt is generated, the source of the interrupt is identified by reading the interrupt registers. The USB frame and transaction related interrupt events, such as Start of Frame interrupt, are grouped in one set of registers: USB Interrupt Flag Register, USB Interrupt Enable Register and USB Interrupt Mask Register. The USB Bus reset and suspend/resume are grouped in another set of registers: Suspend/Resume Register, Suspend/Resume Interrupt Enable Register and Suspend/Resume Interrupt Mask Register.

Figure 5. AT43USB326 Interrupt Structure



Reset Sources

The AT43USB326 has four sources of reset:

- **Power-on Reset** – The MCU is reset when the supply voltage is below the power-on reset threshold.
- **External Reset** – The MCU is reset when a low level is present on the RESET pin for more than 50 ns.
- **Watchdog Reset** – The MCU is reset when the watchdog timer period expires and the watchdog is enabled.
- **USB Reset** – The AT43USB326 has a feature to separate the USB and microcontroller resets. This feature is enabled by setting the BUS INT EN, bit 3 of the SPRSIE register. A USB bus reset is defined as a SE0 (single ended zero) of at least 4 slow speed USB clock cycles received by Port0. The internal reset pulse to the USB hardware and microcontroller lasts for 24 oscillator periods.
  - Resets not separated: A USB bus reset will also reset the microcontroller.
  - Separated reset: A USB bus reset will only reset the USB hardware, while an interrupt to the microcontroller will be generated if the BUS INT MSK bit, bit 3 of SPRSMSK register, is also set.

When the USB hardware is reset, the compound device is de-configured and has to be re-enumerated by the host. When the microcontroller is reset, all I/O registers are then set to their initial values, and the program starts execution from address $000. The instruction placed in address $000 must be a JMP instruction to the reset handling routine. If the program never enables an interrupt source, the interrupt vectors are not used, and regular program code can be placed at these locations. The circuit diagram in Figure 6 shows the reset logic. The user can select the start-up time according to typical oscillator start-up. The number of WDT oscillator cycles used for each time-out is shown in Table 7.

Figure 6. Reset Logic

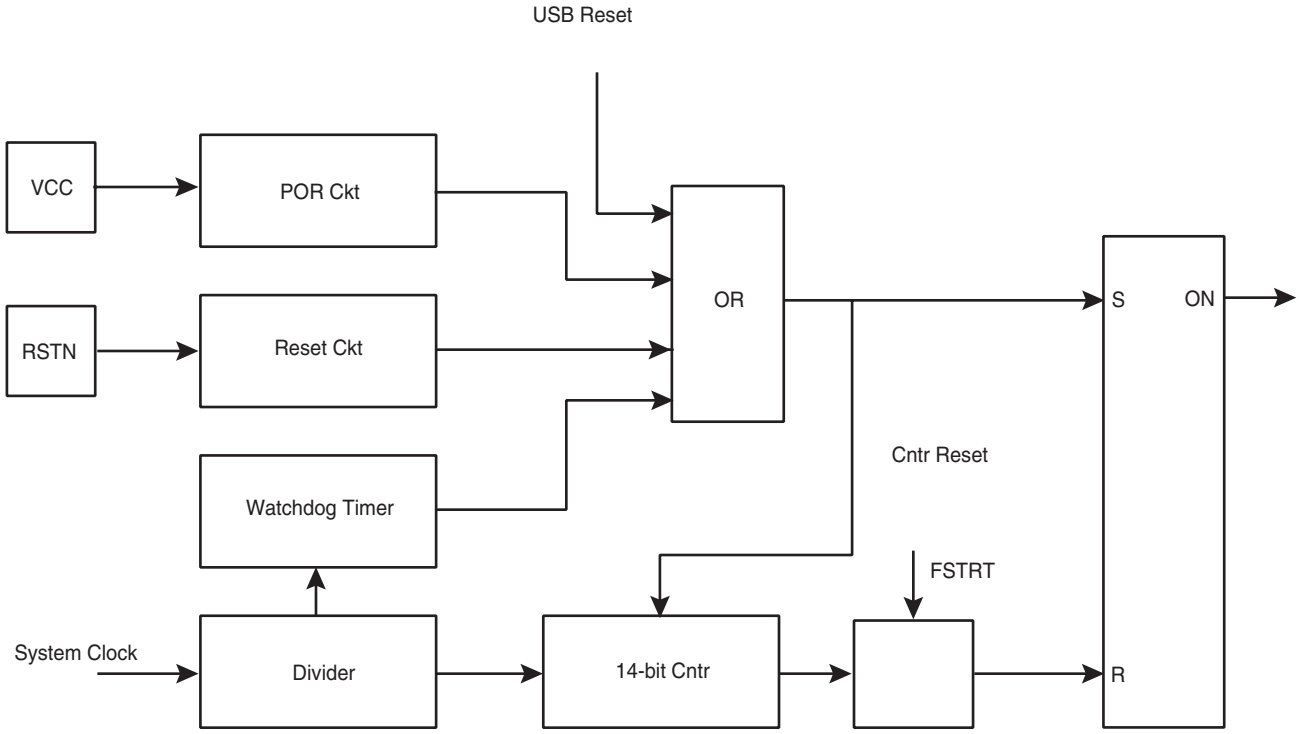


Table 7. Number of Watchdog Oscillator Cycles

FSTRT	Time-out at V_{CC} = 5V	Number of WDT cycles
Programmed	1.1 ms	1K
Unprogrammed	16.0 ms	16K

Power-on Reset

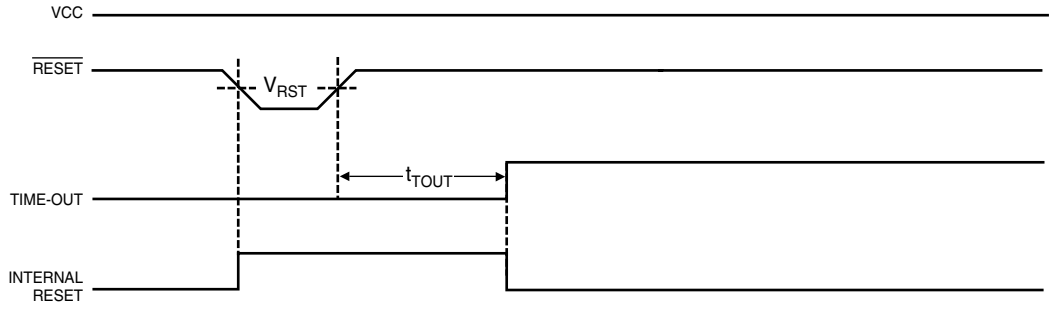
A Power-on Reset (POR) circuit ensures that the device is reset from power-on. An internal timer clocked from the Watchdog timer oscillator prevents the MCU from starting until after a certain period after V_{CC} has reached the power-on threshold voltage, regardless of the V_{CC} rise time.

If the build-in start-up delay is sufficient, RESET can be connected to V_{CC} directly or via an external pull-up resistor. By holding the pin low for a period after V_{CC} has been applied, the Power-on Reset period can be extended.

External Reset

An external reset is generated by a low-level on the RESET pin. Reset pulses longer than 200 ns will generate a reset. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the Reset Threshold Voltage - V_{RST} on its positive edge, the delay timer starts the MCU after the Time-out period t_{TOUT} has expired.

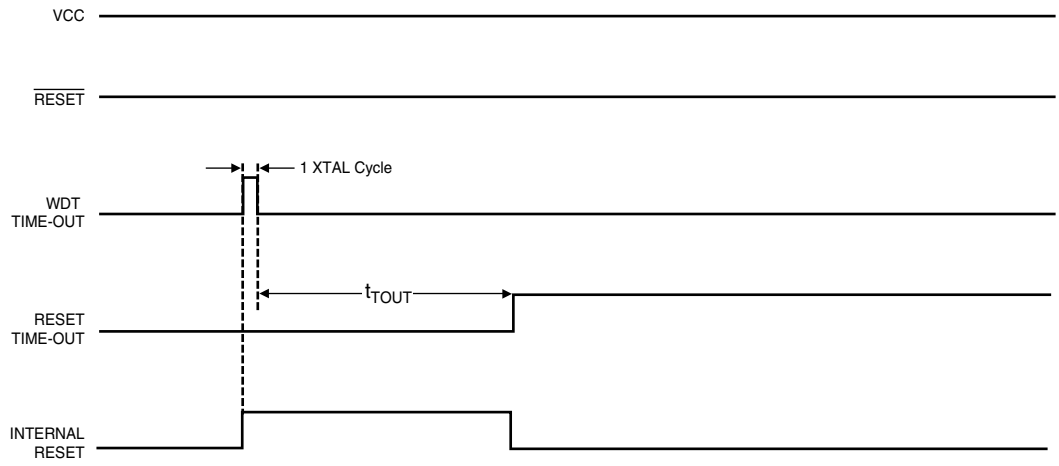
Figure 7. External Reset During Operation



Watchdog Timer Reset

When the watchdog times out, it will generate a short reset pulse of 1 XTAL cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT}.

Figure 8. Watchdog Reset During Operation



Non-USB Related Interrupt Handling

The AT43USB326 has two non-USB 8-bit Interrupt Mask control registers; GIMSK (General Interrupt Mask Register) and TIMSK (Timer/Counter Interrupt Mask Register).

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared (zero) and all interrupts are disabled. The user software can set (one) the I-bit to enable nested interrupts. The I-bit is set (one) when a Return from Interrupt instruction, RETI, is executed.

For Interrupts triggered by events that can remain static (e.g. the Output Compare register1 matching the value of Timer/Counter1) the interrupt flag is set when the event occurs. If the interrupt flag is cleared and the interrupt condition persists, the flag will not be set until the event occurs the next time.

When the Program Counter is vectored to the actual interrupt vector in order to execute the interrupt handling routine, hard-ware clears the corresponding flag that generated the interrupt. Some of the interrupt flags can also be cleared by writing a logic one to the flag bit position(s) to be cleared.

If an interrupt condition occurs when the corresponding interrupt enable bit is cleared (zero), the interrupt flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software.

If one or more interrupt conditions occur when the global interrupt enable bit is cleared (zero), the corresponding interrupt flag(s) will be set and remembered until the global interrupt enable bit is set (one), and will be executed by order of priority.

Note that external level interrupt does not have a flag, and will only be remembered for as long as the interrupt condition is active.

General Interrupt Mask Register – GIMSK

Bit	7	6	5	4	3	2	1	0	
$3B ($5B)	–	INT0	–	–	–	–	–	–	GIMSK
Read/Write	R	R/W	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – Res: Reserved Bit**
- **Bit 6 – INT0: Interrupt Request 0 (Suspend/Resume Interrupt) Enable**

When the INT0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control0 bits 1/0 (ISC01 and ISC00) in the MCU general Control Register (MCUCR) defines whether the external interrupt is activated on rising or falling edge of the INT0 pin or level sensed. Activity on the pin will cause an interrupt request even if INT0 is configured as an output. The corresponding interrupt of Interrupt Request 0 is executed from program memory address $002. See also “External Interrupts” on page 26.

- **Bits 5..0 – Res: Reserved Bits**

These bits are reserved bits in the AT43USB326 and always read as zero.

General Interrupt Flag Register – GIFR

Bit	7	6	5	4	3	2	1	0	
$3A ($5A)	–	INT F0	–	–	–	–	–	–	GIFR
Read/Write	R	R/W	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – Res: Reserved Bit**
- **Bit 6 – INTF0: Interrupt Flag0 (Suspend/Resume Interrupt Flag)**

When an event on the INT0 (that is, a USB event-related interrupt) triggers an interrupt request, INTF0 becomes set (one). If the I-bit in SREG and the INT0 bit in GIMSK are set (one), the MCU will jump to the interrupt vector at address $002. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

- **Bits 5..0 – Res: Reserved Bits**

These bits are reserved bits in the AT43USB326 and always read as zero.

Timer/Counter Interrupt Mask Register – TIMSK

Bit	7	6	5	4	3	2	1	0	
$39 ($59)	–	–	–	–	–	–	TOIE0	–	TIMSK
Read/Write	R	R	R	R	R	R	R/W	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:2 – Res: Reserved Bits**

These bits are reserved bits in the AT43USB326 and always read zero.

- **Bit 1 – TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt (at vector $007) is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the TIFR.

- **Bit 0 – Res: Reserved Bit**

This bit is a reserved bit in the AT43USB326 and always reads zero.

Timer/Counter Interrupt Flag Register – TIFR

Bit	7	6	5	4	3	2	1	0	
$38 ($58)	–	–	–	–	–	–	TOV0	–	TIFR
Read/Write	R	R	R	R	R	R	R/W	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:2 – Res: Reserved Bits**

These bits are reserved bits in the AT43USB326 and always read zero.

- **Bit 1 – TOV: Timer/Counter0 Overflow Flag**

The bit TOV0 is set (one) when an overflow occurs in Timer/Counter0. TOV0 is cleared by the hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG I- bit, and TOIE0 (Timer/Counter0 Overflow Interrupt Enable), and TOV0 are set (one), the Timer/Counter0 Overflow interrupt is executed.

- **Bit 0 – Res: Reserved Bit**

This bit is a reserved bit in the AT43USB326 and always reads zero.

External Interrupts

While in the suspend state, the depression of any key will trigger a resume interrupt. This is the only available external interrupt in the AT43USB326.

Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is 4 clock cycles minimum. 4 clock cycles after the interrupt flag has been set, the program vector address for the actual interrupt handling routine is executed. During this 4 clock cycle period, the Program Counter (2 bytes) is pushed onto the Stack, and the Stack Pointer is decremented by 2. The vector is normally a jump to the interrupt routine, and this jump takes 3 clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served.

A return from an interrupt handling routine (same as for a subroutine call routine) takes 4 clock cycles. During these 4 clock cycles, the Program Counter (2 bytes) is popped back from the Stack, the Stack Pointer is incremented by 2, and the I flag in SREG is set. When the AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

MCU Control Register – MCUCR

Bit	7	6	5	4	3	2	1	0	
$35 ($55)	–	–	SE	SM	–	–	–	–	MCUCR
Read/Write	R	R	R/W	R/W	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7, 6 – Res: Reserved Bits**
- **Bit 5 – SE: Sleep Enable**

The SE bit must be set (1) to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode, unless it is the programmer's purpose, it is recommended to set the Sleep Enable SE bit just before the execution of the SLEEP instruction.

- **Bit 4 – SM: Sleep Mode**

This bit selects between the two available sleep modes. When SM is cleared (zero), Idle Mode is selected as Sleep Mode. When SM is set (1), Power Down mode is selected as sleep mode. The AT43USB326 does not support the Idle Mode and SM should always be set to one when entering the Sleep Mode.

- **Bit 3:0 – Res: Reserved Bits**

USB Interrupt Sources

The USB interrupts are described below.

Table 8. USB Interrupt Sources

Interrupt	Description
SOF Received	Whenever USB hardware decodes a valid Start of Frame. The frame number is stored in the two Frame Number Registers.
EOF2	Activated whenever the hub's frame timer reaches its EOF2 time point.
Function EP0 Interrupt	See “Control Transfers at Control Endpoint EP0” on page 47 for details.
Function EP1 Interrupt	For an OUT endpoint it indicates that Function Endpoint 1 has received a valid OUT packet and that the data is in the FIFO. For an IN endpoint it means that the endpoint has received an IN token, sent out the data in the FIFO and received an ACK from the Host. The FIFO is now ready to be written by new data from the microcontroller.
Function EP2 Interrupt	For an OUT endpoint it indicates that Function Endpoint 2 has received a valid OUT packet and that the data is in the FIFO. For an IN endpoint it means that the endpoint has received an IN token, sent out the data in the FIFO and received an ACK from the Host. The FIFO is now ready to be written by new data from the microcontroller.
Hub EP0 Interrupt	See “Control Transfers at Control Endpoint EP0” on page 47 for details.
FRWUP	USB hardware has received a embedded function remote wakeup request.
GLB SUSP	USB hardware has received global suspend signaling and is preparing to put the hub in the suspend mode. The microcontroller's firmware should place the embedded function in the suspend state.
RSM	USB hardware received resume signaling and is propagating the resume signaling. The microcontroller's firmware should take the embedded function out of the suspended state.

All interrupts have individual enable, status, and mask bits through the interrupt enable register and interrupt mask register. The Suspend and Resume interrupts are cleared by writing a 0 to the particular interrupt bit. All other interrupts are cleared when the microcontroller sets a bit in an interrupt acknowledge register.

USB Endpoint Interrupt Sources

An assertion or activation of one or more bits in the endpoint's Control and Status Register triggers the endpoint interrupts. These triggers are different for control and non-control endpoints as described in the table below. Please refer to the Control and Status Register for more information.

Table 9. USB Endpoint Interrupt Sources

Bit	Endpoint type
RX_OUT_PACKET	CONTROL, OUT
TX_COMPLETE	CONTROL, IN
STALL_SENT	CONTROL, IN
RX_SETUP	CONTROL

USB Interrupt Status Register – UISR

Bit	7	6	5	4	3	2	1	0	
$1FF7	SOF INT	EOF2 INT	–	–	HEP0 INT	FE2 INT	FE1 INT	FE0 INT	UISR
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SOF INT: Start of Frame Interrupt**

This bit is asserted after the USB hardware receives a valid SOF packet.

- **Bit 6 – EOF2 INT: EOF2 Interrupt**

This bit is asserted 10 clocks before the expected start of a frame.

- **Bit 5, 4 – Res: Reserved Bits**

These bits are reserved and always read as zero.

- **Bit 3 – HEP0 INT: Hub Endpoint 0 Interrupt**
- **Bit 2 – FEP2 INT: Function Endpoint 2 Interrupt**
- **Bit 1 – FEP1 INT: Function Endpoint 1 Interrupt**
- **Bit 0 – FEP0 INT: Function Endpoint 0 Interrupt**

The hub and function interrupt bits will be set by the hardware whenever the following bits in the corresponding endpoint's Control and Status Register are modified by the USB hardware:

1. RX OUT Packet is set (control and OUT endpoints)
2. TX Packet Ready is cleared AND TX Complete is set (control and IN endpoints)
3. RX SETUP is set (control endpoints only)
4. TX Complete is set

USB Interrupt Mask Register – UIMSKR

Bit	7	6	5	4	3	2	1	0	
$1FF6	SOF IMSK	EOF2 IMSK	–	–	HEP0 IMSK	FEP2 IMSK	FEP1 IMSK	FEP0 IMSK	UIMSKR
Read/Write	R/W	R/W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SOF IMSK: Enable Start of Frame Interrupt Mask**

When the SOF IMSK bit is set (1), the Start of Frame Interrupt is masked.

- **Bit 6 – EOF2 IMSK: Enable EOF2 Interrupt**

When the EOF2 IMSK bit is set (1), the EOF2 Interrupt is masked.

- **Bit 5, 4 – Res: Reserved Bits**

These bits are reserved and always read as zero.

- **Bit 3 – HEP0 IMSK: Enable Endpoint 0 Interrupt**

When the HEP0 IMSK bit is set (1), the Hub Endpoint 0 Interrupt is masked.

- **Bit 2 – FEP2 IMSK: Enable Endpoint 2 Interrupt**

When the FE2 IMSK bit is set (1), the Function Endpoint 2 Interrupt is masked.

- **Bit 1 – FEP1 IMSK: Enable Endpoint 1 Interrupt**

When the FE1 IMSK bit is set (1), the Function Endpoint 1 Interrupt is masked.

- **Bit 0 – FEP0 IMSK: Enable Endpoint 0 Interrupt**

When the FE0 IMSK bit is set (1), the Function Endpoint 0 Interrupt is masked.

USB Interrupt Acknowledge Register – UIAR

Bit	7	6	5	4	3	2	1	0	
$1FF5	SOF INTACK	EOF2 INTACK	–	–	HEP0 INTACK	FEP2 IMSK	FEP1 INTACK	FEP0 INTACK	UIAR
Read/Write	W	W	R	R	W	W	W	W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SOF INTACK: Start of Frame Interrupt Acknowledge**

The microcontroller firmware writes a 1 to this bit to clear the SOF INT bit.

- **Bit 6 – EOF2 INTACK: EOF2 Interrupt Acknowledge**

The microcontroller firmware writes a 1 to this bit to clear the EOF2 INT bit.

- **Bit 5, 4 – Res: Reserved Bits**

These bits are reserved and are always read as zero.

- **Bit 3 – HEP0 INTACK: Hub Endpoint 0 Interrupt Acknowledge**

The microcontroller firmware writes a 1 to this bit to clear the HEP0 INT bit.

- **Bit 2 – FEP2 INTACK: Function Endpoint 2 Interrupt Acknowledge**

The microcontroller firmware writes a 1 to this bit to clear the FEP2 bit.

- **Bit 1 – FEP1 INTACK: Function Endpoint 1 Interrupt Acknowledge**

The microcontroller firmware writes a 1 to this bit to clear the FEP1 bit.

- **Bit 0 – FEP0 INTACK: Function Endpoint 0 Interrupt Acknowledge**

The microcontroller firmware writes a 1 to this bit to clear the FEP0 INT bit.

USB Interrupt Enable Register – UIER

Bit	7	6	5	4	3	2	1	0	
$1FF3	SOF IE	EOF2 IE	–	–	HEP0 IE	FEP2 IE	FEP1 IE	FEP0 IE	UIER
Read/Write	R/W	R/W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SOF IE: Enable Start of Frame Interrupt**

When the SOF IE bit is set (1), the Start of Frame Interrupt is enabled.

- **Bit 6 – EOF2 IE: Enable EOF2 Interrupt**

When the EOF2 IE bit is set (1), the EOF2 Interrupt is enabled.

- **Bit 5, 4 – Res: Reserved Bits**

These bits are reserved and always read as zero.

- **Bit 3 – HEP0 IE: Enable Endpoint 0 Interrupt**

When the HEP0 IE bit is set (1), the Hub Endpoint 0 Interrupt is enabled.

- **Bit 2 – FEP2 IE: Enable Endpoint 2 Interrupt**

When the FE2 IE bit is set (1), the Function Endpoint 2 Interrupt is enabled.

- **Bit 1 – FEP1 IE: Enable Endpoint 1 Interrupt**

When the FE1 IE bit is set (1), the Function Endpoint 1 Interrupt is enabled.

- **Bit 0 – FEP0 IE: Enable Endpoint 0 Interrupt**

When the FE0 IE bit is set (1), the Function Endpoint 0 Interrupt is enabled.

Suspend/Resume Register – SPRSR

Bit	7	6	5	4	3	2	1	0	
$1FFA	–	–	–	–	–	FRWUP	RSM	GLB SUSP	SPRSR
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..3 – Res: Reserved Bits**

These bits are reserved and are always read as zeros.

- **Bit 2 – FRWUP: Function Remote Wakeup**

The USB hardware sets this bit to signal that External Interrupt 1 is detected indicating remote wakeup. An interrupt is generated if the FRWUP IE bit of the SPRSIE register is set.

- **Bit 1 – RSM: Resume**

The USB hardware sets this bit when a USB resume signaling is detected at any of its port except Port 1. An interrupt is generated if the RSM IE bit of the SPRSIE register is set.

- **Bit 0 – GLB SUSP: Global Suspend**

The USB hardware sets this bit when a USB global suspend signaling is detected. An interrupt is generated if the GLBSUSP IE bit of the SPRSIE register is set.

Suspend/Resume Interrupt Enable Register – SPRSIE

Bit	7	6	5	4	3	2	1	0	
$1FF9	–	–	–	–	–	FRWUP	RSM	GLB SUSP	SPRSIE
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..3 – Res: Reserved Bits**

These bits are reserved and are always read as zeros.

- **Bit 2 – FRWUP IE: Function Remote Wakeup Interrupt Enable**

Setting the FRWUP IE bit will initiate an interrupt whenever the FRWUP bit of SPRSR is set.

- **Bit 1 – RSM IE: Resume Interrupt Enable**

Setting the RSM IE bit will initiate an interrupt whenever the RSM bit of SPRSR is set.

- **Bit 0 – GLB SUSP IE: Global Suspend Interrupt Enable**

Setting the GLB SUSP IE bit will initiate an interrupt whenever the GLB SUSP bit of SPRSR is set.

Suspend/Resume Interrupt Mask Register – SPRSMSK

Bit	7	6	5	4	3	2	1	0	
$1FF8	–	–	–	–	–	FRWUP MSK	RSM	GLB SUSP	SPRSMSK
Read/Write	R	R	R	R	R	W	W	W	
Initial Value	0	0	0	0	0	0	0	0	

The bits of the Suspend/Resume Mask Register are used to make an interrupt caused by an event in the Suspend/Resume Register visible to the microcontroller. The Suspend/Resume Interrupt Enable Register bits enable the interrupt while the Suspend/Resume Interrupt Mask Register allows the microcontroller to control when it wants visibility to an interrupt. 1 = Enable Mask, 0 = Disable Mask.

- **Bit 7..3 – Res: Reserved Bits**

These bits are reserved and are always read as zeros.

- **Bit 2 – FRWUP MSK: Function Remote Wakeup Interrupt Mask**
- **Bit 1 – RSM MSK: Resume Interrupt Mask**
- **Bit 0 – GLB SUSP MSK: Global Suspend Interrupt Enable**

AVR Register Set

Status Register and Stack Pointer

Status Register – SREG

Bit	7	6	5	4	3	2	1	0	
$3F ($5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – I: Global Interrupt Enable**

The global interrupt enable bit must be set (one) for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the global interrupt enable bit is cleared (zero), none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by the hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts.

- **Bit 6 – T: Bit Copy Storage**

The bit copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T bit as source and destination for the operated bit. A bit from a register in the register file can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the register file by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The half carry flag H indicates a half carry in some arithmetic operations. See the Instruction Set Description for detailed information.

- **Bit 4 – S: Sign Bit, S = N⊕V**

The S-bit is always an exclusive or between the negative flag N and the two's complement overflow flag V. See the Instruction Set Description for detailed information.

- **Bit 3 – V: Two's Complement Overflow Flag**

The two's complement overflow flag V supports two's complement arithmetics. See the Instruction Set Description for detailed information.

- **Bit 2 – N: Negative Flag**

The negative flag N indicates a negative result after the different arithmetic and logic operations. See the Instruction Set Description for detailed information.

- **Bit 1 – Z: Zero Flag**

The zero flag Z indicates a zero result after the different arithmetic and logic operations. See the Instruction Set Description for detailed information.

- **Bit 0 – C: Carry Flag**

The carry flag C indicates a carry in an arithmetic or logic operation. See the Instruction Set Description for detailed information.

Note that the status register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt routine. This must be handled by software.

Stack Pointer Register – SP

Bit	15	14	13	12	11	10	9	8	
$3E ($5E)	**I**	**T**	**H**	**S**	**V**	**N**	**Z**	**C**	**SPH**
$3D ($5D)	**SP7**	**SP6**	**SP5**	**SP4**	**SP3**	**SP2**	**SP1**	**SP0**	**SPL**
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The Stack Pointer points to the data SRAM stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The stack pointer must be set to point above $60. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when an address is pushed onto the Stack with subroutine calls and interrupts. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction and it is incremented by two when an address is popped from the Stack with return from subroutine RET or return from interrupt RETI.

Sleep Modes

To enter the sleep modes, the SE bit in MCUCR must be set (one) and a SLEEP instruction must be executed. If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU awakes, executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the register file, SRAM and I/O memory are unaltered. If a reset occurs during sleep mode, the MCU wakes up and executes from the Reset vector.

Power Down Mode

When the SM bit is set (one), the SLEEP instruction forces the MCU into the Power Down Mode. In this mode, the external oscillator is stopped, while the external interrupts continue operating. Only an external reset, an external level interrupt on INT0 or INT1, can wake up the MCU.

Note that when a level triggered interrupt is used for wake-up from power down, the low level must be held for a time longer than the reset delay time-out period t_{TOUT}. Otherwise, the MCU will fail to wake up.

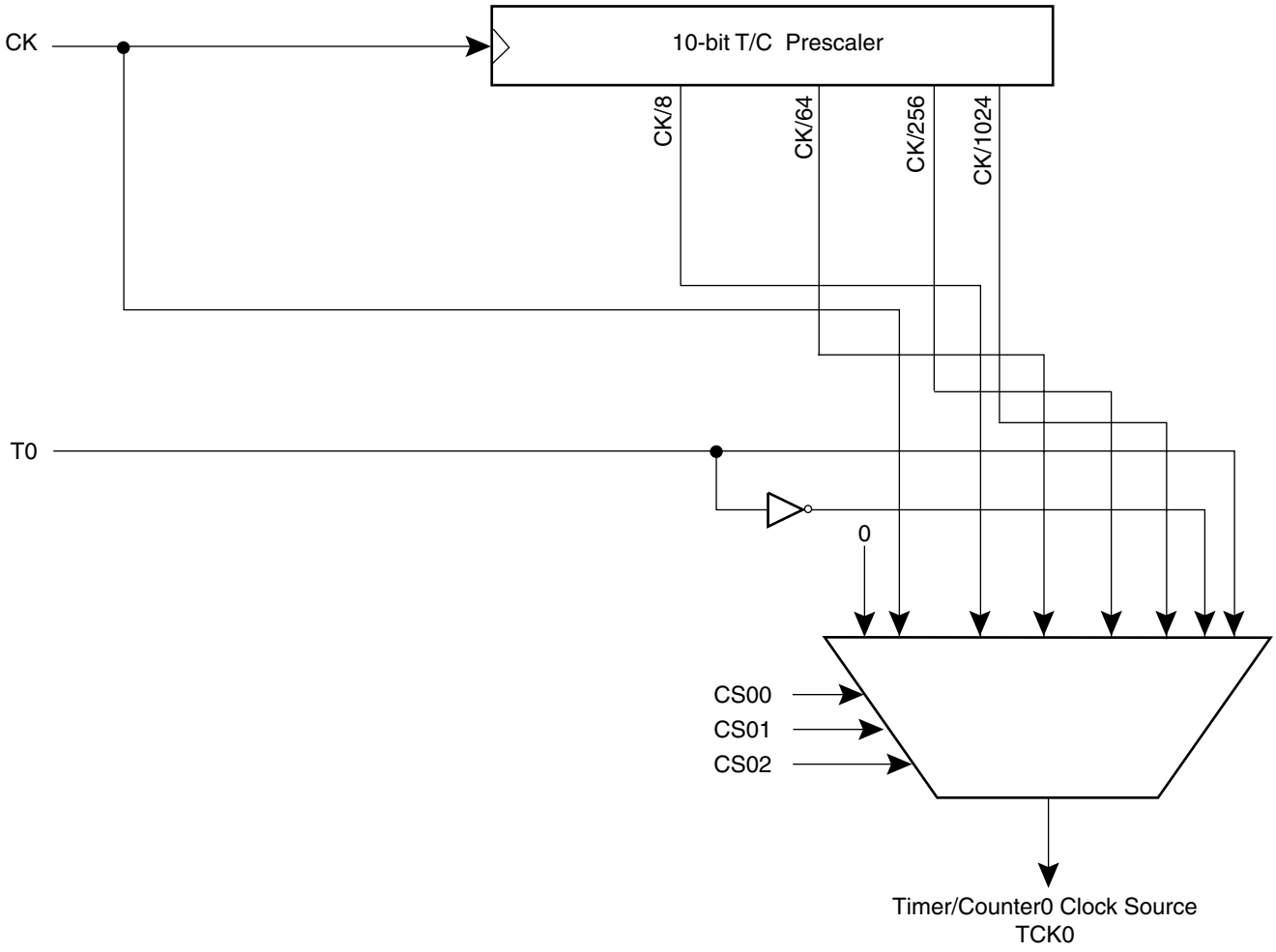
Timer/Counter

The AT43USB326 provides one general-purpose 8-bit Timer/Counter (T/C). The Timer/Counter has prescaling selection from a 10-bit prescaling timer. The Timer/Counter can either be used as a timer with an internal clock timebase or as a counter with an external pin connection which triggers the counting.

Timer/Counter Prescaler

The four different prescaled selections are: CK/8, CK/64, CK/256 and CK/1024 where CK is the oscillator clock. Added selections such as CK, external source and stop, can be selected as clock sources.

Figure 9. Timer/Counter Prescaler



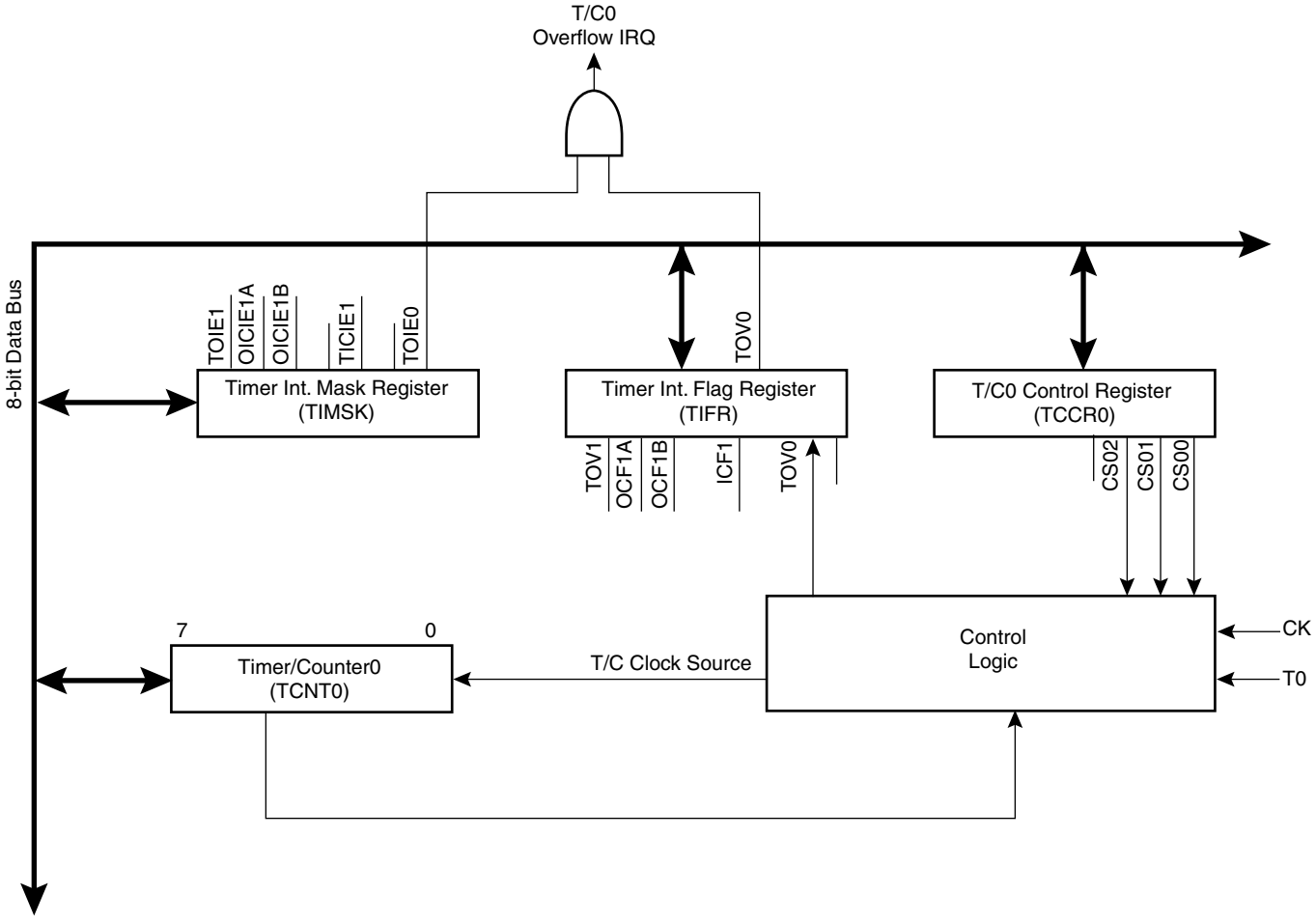
8-bit Timer/Counter0

The 8-bit Timer/Counter0 can select clock source from CK, prescaled CK or an external pin. In addition it can be stopped as described in the specification for the Timer/Counter0 Control Register (TCCR0). The overflow status flag is found in the Timer/Counter Interrupt Flag Register (TIFR). Control signals are found in the Timer/Counter0 Control Register (TCCR0). The interrupt enable/disable settings for Timer/Counter0 are found in the Timer/Counter Interrupt Mask Register - TIMSK.

When Timer/Counter0 is externally clocked, the external signal is synchronized with the oscillator frequency of the CPU. To assure proper sampling of the external clock, the minimum time between two external clock transitions must be at least one internal CPU clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

The 8-bit Timer/Counter0 features both a high resolution and a high accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities make the Timer/Counter0 useful for lower speed functions or exact timing functions with infrequent actions.

Figure 10. Timer/Counter0 Block Diagram



Timer/Counter0 Control Register – TCCR0

Bit	7	6	5	4	3	2	1	0	
$33 ($53)	–	–	–	–	–	CS02	CS01	CS00	**TCCR0**
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..3 – Res: Reserved Bits**

These bits are reserved bits in the AT43USB355 and always read as zero.

- **Bits 2, 1, 0 – CS02, CS01, CS00: Clock Select0, bit 2, 1 and 0**

The Clock Select0 bits 2, 1 and 0 define the prescaling source of Timer/Counter0.

Table 10. Clock 0 Prescale Select

CS02	CS01	CS00	Description
0	0	0	Stop, the Timer/Counter0 is stopped
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	External Pin T0, falling edge
1	1	1	External Pin T0, rising edge

The Stop condition provides a Timer Enable/Disable function. The CK down divided modes are scaled directly from the CK oscillator clock. If the external pin modes are used for Timer/Counter0, transitions on PB0/(T0) will clock the counter even if the pin is configured as an output. This feature can give the user SW control of the counting.

Timer/Counter0 – TCNT0

Bit	7	6	5	4	3	2	1	0	
$32 ($52)	**MSB**	–	–	–	–	–	–	**LSB**	**TCNT0**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

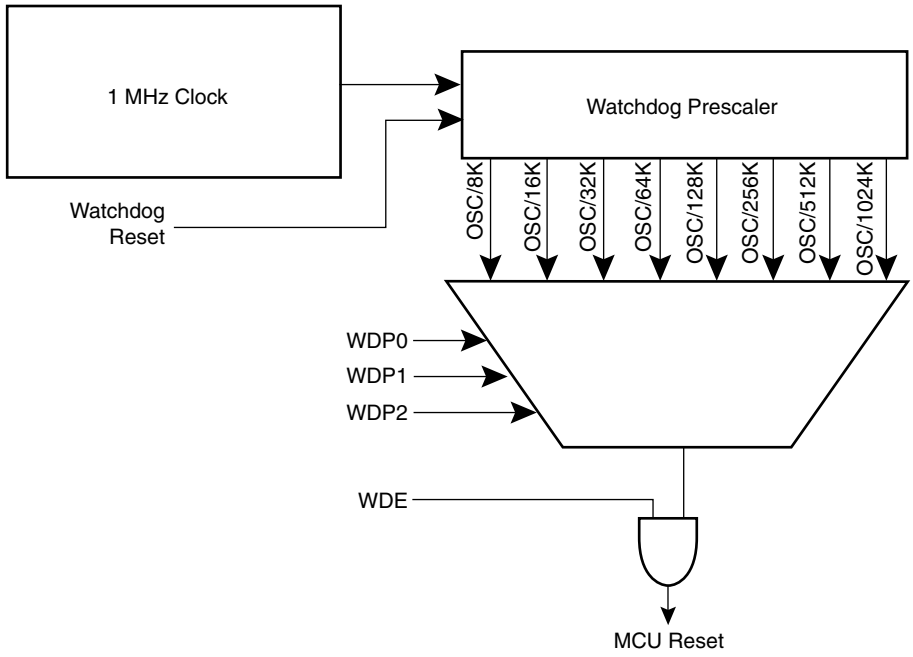
The Timer/Counter0 is realized as an up-counter with read and write access. If the Timer/Counter0 is written and a clock source is present, the Timer/Counter0 continues counting in the clock cycle following the write operation.

Watchdog Timer

The Watchdog Timer is clocked from a 1 MHz clock derived from the 6 MHz on chip oscillator. By controlling the Watchdog Timer prescaler, the Watchdog reset interval can be adjusted, see Table 11 for a detailed description. The WDR (Watchdog Reset) instruction resets the Watchdog Timer. Eight different clock cycle periods can be selected to determine the reset period. If the reset period expires without another Watchdog reset, the AT43USB326 resets and executes from the reset vector.

To prevent unintentional disabling of the watchdog, a special turn-off sequence must be followed when the watchdog is disabled. Refer to the description of the Watchdog Timer Control Register for details.

Figure 11. Watchdog Timer



Watch Dog Timer Control Register – WDTCR

Bit	7	6	5	4	3	2	1	0	
$21 ($41)	–	–	–	WDTOE	WDE	WDP2	WDP1	WDP0	WDTCR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..5 – Res: Reserved Bits**

These bits are reserved bits in the AT43USB326 and will always read as zero.

- **Bit 4 – WDTOE: Watch Dog Turn-Off Enable**

This bit must be set (one) when the WDE bit is cleared. Otherwise, the watchdog will not be disabled. Once set, the hardware will clear this bit to zero after four clock cycles. Refer to the description of the WDE bit for a watchdog disable procedure.

- **Bit 3 – WDE: Watch Dog Enable**

When the WDE is set (one) the Watchdog Timer is enabled, and if the WDE is cleared (zero) the Watchdog Timer function is disabled. WDE can only be cleared if the WDTOE bit is set (one). To disable an enabled watchdog timer, the following procedure must be followed:

1. In the same operation, write a logical one to WDTOE and WDE. A logical one must be written to WDE even though it is set to one before the disable operation starts.
2. Within the next four clock cycles, write a logical 0 to WDE. This disables the watchdog.

- **Bits 2..0 – WDP2, WDP1, WDP0: Watch Dog Timer Prescaler 2, 1 and 0**

The WDP2, WDP1 and WDP0 bits determine the Watchdog Timer prescaling when the Watchdog Timer is enabled. The different prescaling values and their corresponding Time-out Periods are shown in Table 11.

Table 11. Watchdog Timer Prescale Select

WDP2	WDP1	WDP0	Number of WDT Oscillator cycles	Time-out
0	0	0	8K cycles	8.2 ms
0	0	1	16K cycles	16.4 ms
0	1	0	32K cycles	33.8 ms
0	1	1	64K cycles	65.6 ms
1	0	0	128K cycles	0.131 s
1	0	1	256K cycles	0.262 s
1	1	0	512K cycles	0.524 s
1	1	1	1,024K cycles	1.048 s

Note: The WDR (Watchdog Reset) instruction should always be executed before the Watchdog Timer is enabled. This ensures that the reset period will be in accordance with the Watchdog Timer prescale settings. If the Watchdog Timer is enabled without reset, the watchdog timer may not start to count from zero. To avoid unintentional MCU reset, the Watchdog Timer should be disabled or reset before changing the Watchdog Timer Prescale Select.

I/O Ports

All AVR ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies for changing drive value if configured as output or enabling/disabling of pull-up resistors if configured as input.

The keyboard matrix strobe output pins, PA[0:7], PB[0:7] and PE[0:1] have controlled slope drivers. With a load of 100 pF, the output fall time ranges between 75 ns and 300 ns. The keyboard matrix strobe input pins, PC[0:7] have built-in pull-up resistors, 20 KΩ nominal value, to the internal 3.3V power supply.

PE[4:7] have 5V tolerant outputs and each has a built-in series resistor of 330Ω nominal value. These output pins are designed for driving an LED connected to the 5V supply.

The dedicated functions are summarized in Table 12.

Table 12. GPIO Function Assignments

Function	GPIO
Scan out[0:7]	PA[0:7]
Scan out[8:15]	PB[0:7]
Scan out[16,17]	PE[0,1]
Scan in[0:7]	PC[0:7]
LED drivers	PE[4:7]

Port A

Port A is an 8-bit bi-directional I/O port with open drain outputs and controlled slew rate. It is designed for use as the column driver in a keyboard controller. The Port A output buffers can sink or source 4 mA.

Three I/O memory address locations are allocated for the Port A, one each for the Data Register PORTA, $1B($3B), Data Direction Register (DDRA), $1A($3A) and the Port A Input Pins (PINA) $19($39). The Port A Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

Port A Data Register – PORTA

Bit	7	6	5	4	3	2	1	0	
$1B ($3B)	**PORTA7**	**PORTA6**	**PORTA5**	**PORTA4**	**PORTA3**	**PORTA2**	**PORTA1**	**PORTA0**	**PORTA**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port A Data Direction Register – DDRA

Bit	7	6	5	4	3	2	1	0	
$1A ($3A)	**DDA7**	**DDA6**	**DDA5**	**DDA4**	**DDA3**	**DDA2**	**DDA1**	**DDA0**	**DDRA**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port A Input Pins Address – PINA

Bit	7	6	5	4	3	2	1	0	
$19 ($39)	**PINA7**	**PINA6**	**PINA5**	**PINA4**	**PINA3**	**PINA2**	**PINA1**	**PINA0**	**PINA**
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

The Port A Input Pins address (PINA) is not a register, and this address enables access to the physical value on each Port A pin. When reading PORTA the Port A Data Latch is read, and when reading PINA, the logical values present on the pins are read.

Port B

Port B is an 8-bit bi-directional I/O port with open drain outputs and controlled slew rate. It is designed for use as the column driver in a keyboard controller. The Port B output buffers can sink or source 4 mA.

Three I/O memory address locations are allocated for the Port B, one each for the Data Register - PORTB, $18($38), Data Direction Register (DDRB), $17($37) and the Port B Input Pins (PINB), $16($36). The Port B Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

Port B Data Register – PORTB

Bit	7	6	5	4	3	2	1	0	
$18 ($38)	**PORTB7**	**PORTB6**	**PORTB5**	**PORTB4**	**PORTB3**	**PORTB2**	**PORTB1**	**PORTB0**	**PORTB**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port B Data Direction Register – DDRB

Bit	7	6	5	4	3	2	1	0	
$17 ($37)	**DDB7**	**DDB6**	**DDB5**	**DDB4**	**DDB3**	**DDB2**	**DDB1**	**DDB0**	**DDRB**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port B Input Pins Address – PINB

Bit	7	6	5	4	3	2	1	0	
$16 ($36)	**PINB7**	**PINB6**	**PINB5**	**PINB4**	**PINB3**	**PINB2**	**PINB1**	**PINB0**	**PINB**
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

The Port B Input Pins address (PINB) is not a register, and this address enables access to the physical value on each Port B pin. When reading PORTB, the Port B Data Latch is read, and when reading PINB, the logical values present on the pins are read.

Port C

Port C is an 8-bit bi-directional I/O port with an internal pull-up resistor at each pin. Port C is designed for use as the row inputs of a keyboard controller. Its output buffers can sink 4 mA.

Three I/O memory address locations are allocated for the Port C, one each for the Data Register - PORTC, $15($35), Data Direction Register - DDRC, $14($34) and the Port C Input Pins – PINC, $13($33). The Port C Input Pin's address is read only, while the Data Register and the Data Direction Register are read/write.

Port C Data Register – PORTC

Bit	7	6	5	4	3	2	1	0	
$15 ($35)	**PORTC7**	**PORTC6**	**PORTC5**	**PORTC4**	**PORTC3**	**PORTC2**	**PORTC1**	**PORTC0**	**PORTC**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port C Data Direction Register – DDRC

Bit	7	6	5	4	3	2	1	0	
$14 ($34)	**DDC7**	**DDC6**	**DDC5**	**DDC4**	**DDC3**	**DDC2**	**DDC1**	**DDC0**	**DDRC**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port C Input Pins Address – PINC

Bit	7	6	5	4	3	2	1	0	
$13 ($33)	**PINC7**	**PINC6**	**PINC5**	**PINC4**	**PINC3**	**PINC2**	**PINC1**	**PINC0**	**PINC**
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

The Port C Input Pins address (PINC) is not a register, and this address enables access to the physical value on each Port C pin. When reading PORTC, the Port C Data Latch is read, and when reading PINC, the logical values present on the pins are read.

Port D

Port D is a 2-bit bi-directional I/O port. Its output buffers can sink or source 2 mA.

Three I/O memory address locations are allocated for the Port D, one each for the Data Register - PORTD, $12($32), Data Direction Register (DDRD), $11($31) and the Port D Input Pins (PIND), $10($30). The Port D Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

Port D Data Register – PORTD

Bit	7	6	5	4	3	2	1	0	
$12 ($32)	–	–	–	–	–	–	**PORTD1**	**PORTD0**	**PORTD**
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port D Data Direction Register – DDRD

Bit	7	6	5	4	3	2	1	0	
$11 ($31)	–	–	–	–	–	–	**DDD1**	**DDD0**	**DDRD**
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port D Input Pins Address – PIND

Bit	7	6	5	4	3	2	1	0	
$10 ($30)	–	–	–	–	–	–	**PIND1**	**PIND0**	**PIND**
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

The Port D Input Pins address (PIND) is not a register, and this address enables access to the physical value on each Port D pin. When reading PORTD, the Port D Data Latch is read, and when reading PIND, the logical values present on the pins are read.

Port E

Port E[0,1] are bi-directional I/O ports with open drain outputs and controlled slew rate and are designed for use as the column drivers in a keyboard controller. The Port E[0,1] output buffers can sink 4 mA. Port E[4:7] are bi-directional I/O with open drain outputs capable of driving LEDs directly. Each pin of Port E[4:7] has a series resistor to limit the LEDs current.

Three I/O memory address locations are allocated for the Port E, one each for the Data Register – PORTE, $03($23), Data Direction Register – DDRE, $02($22) and the Port E Input Pins – PINE, $01($21). The Port E Input Pin's address is read only, while the Data Register and the Data Direction Register are read/write.

Port E Data Register – PORTE

Bit	7	6	5	4	3	2	1	0	
$03($23)	PORTE7	PORTE6	PORTE5	PORTE4	–	–	PORTE1	PORTE0	PORTE
Read/Write	R/W	R/W	R/W	R/W	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

Port E Data Direction Register – DDRE

Bit	7	6	5	4	3	2	1	0	
$02 ($22)	DDE7	DDE6	DDE5	DDE4	–	–	DDE1	DDE0	DDRE
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port E Input Pins Address – PINE

Bit	7	6	5	4	3	2	1	0	
$01 ($21)	PINE7	PINE6	PINE5	PINE4	–	–	PINE1	PINE0	PINE
Read/Write	R/W	R/W	R/W	R/W	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

The Port E Input Pins Address – PINE (Port E Input Pins) – is not a register, and this address enables access to the physical value on each Port E pin. When reading PORTE the Port E Data Latch is read, and when reading PINE, the logical values present on the pins are read.

Programming the USB Module

The USB hardware consists of two devices, hub and function, each with their own device address and endpoints. Its operation is controlled through a set of memory mapped registers. The exact configuration of the USB device is defined by the software and it can be programmed to operate as a compound device, or as a hub only or as a function only. The hub has the required control and interrupt endpoints. The number of external downstream ports is programmable from 0 to 2. The DP and DM pins of the unused port(s) must be connected to ground. The USB function has one control endpoint and 2 programmable endpoints. All the endpoints have their own 8-byte FIFOs. If the hub is disabled, one extra endpoint becomes available to the function.

The USB Function

The USB function hardware is designed to operate in the single packet mode and to manage the USB protocol layer. It consists of a Serial Interface Engine (SIE), endpoint FIFOs and a Function Interface Unit (FIU). The SIE performs the following tasks: USB signaling detection/generation, data serialization/de-serialization, data encoding/decoding, bit stuffing and unstuffing, clock/data separation, and CRC generation/checking. It also decodes and manages all packet data types and packet fields.

The endpoint FIFO buffers the data to be sent out or data received. The FIU manages the flow of data between the SIE, FIFO and the internal microcontroller bus. It controls the FIFO and monitors the status of the transactions and interfaces to the CPU. It initiates interrupts and acts upon commands sent by the firmware.

The USB function hardware of the AT43USB326 makes the physical interface and the protocol layer transparent to the user. To start the process, the firmware must first enable the endpoints and which place them in receive mode by default. The device address by default is address 0. The USB function hardware then waits for a setup token from the host. When a valid setup token is received, it automatically stores the data packet in endpoint 0 FIFO and responds with an ACK. It then notifies the microcontroller through an interrupt. The microcontroller reads the FIFO and parses the request.

Transactions for the non-control endpoints are even simpler. Once the endpoint is enabled, it waits for an IN or an OUT token depending whether it is programmed as an IN or OUT endpoint. For example, if it is an IN endpoint, the microcontroller simply loads the data into the endpoint's FIFO and sets a bit in the control and status register. The USB hardware will assemble the data in a USB packet and waits for an IN token. When it receives one, it automatically responds by transmitting the data packet and completes the transaction by waiting for the host's ACK. When one is received, the USB hardware will signal the microcontroller that the transaction has been completed successfully. Retries and data toggles are performed automatically by the USB hardware. When the IN endpoint is not ready to send data, in the case where the microcontroller has not filled the FIFO, it will automatically respond with a NAK.

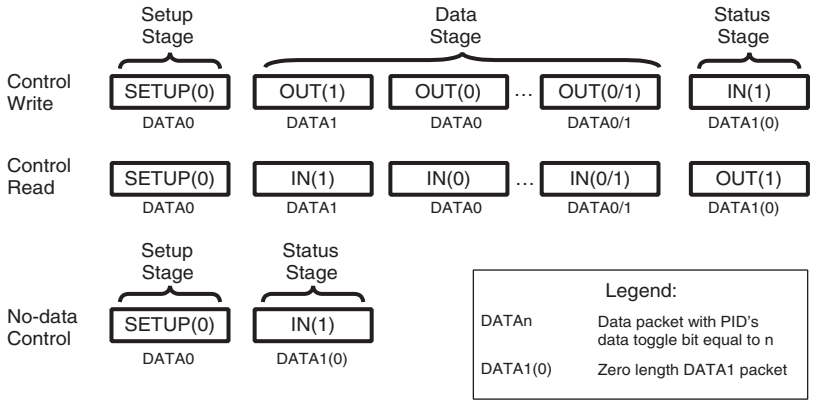
Similarly, an OUT endpoint will wait for an OUT token. When one is received, it will store the data in the FIFO, completes the transaction and interrupt the microcontroller, which then reads the FIFO and enables the endpoint for the next packet. If the FIFO is not cleared, the USB hardware will responds with a NAK.

A detailed description of how USB transactions are handled is described in the following sections. First for a control endpoint and then for non-control endpoints.

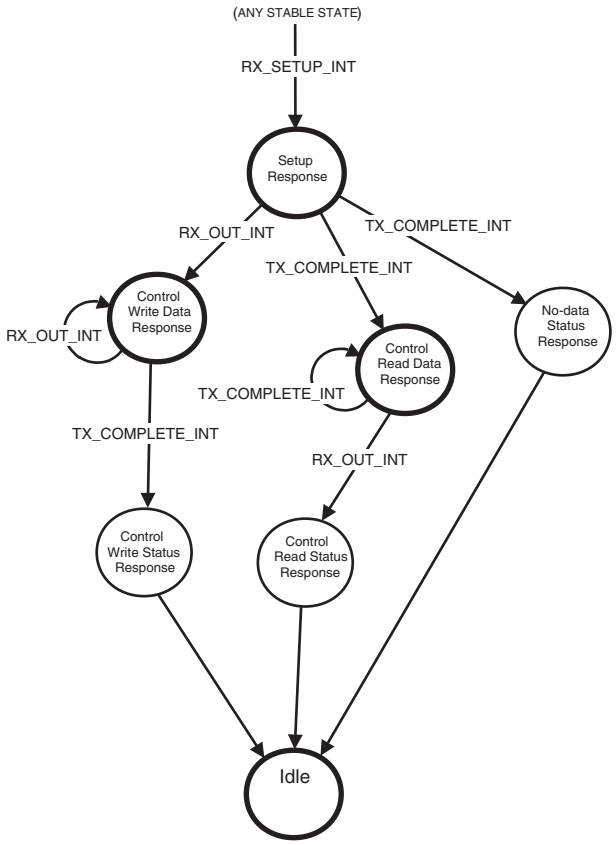
Control Transfers at Control Endpoint EP0

The description given below is for the function control endpoint, but applies to the hub control endpoint as well if the proper registers are used.

The following illustration describes the three possible types of control transfers – Control Write, Control Read and No-data control:



The following state diagram shows how the various state transitions are triggered. Additional decision making may take place within the response states to determine the next expected state. Unmarked arcs represent transitions that trigger immediately following completion of the response state processing. Stable states, those requiring an interrupt to exit having no unmarked arcs as exit paths, are shown in bold.



The following information describes how the AT43USB326's USB hardware and firmware operates during a control transfer between the host and the hub's or function's control endpoint.

Legend: DATA1/DATA0 = Data packet with DATA1 or DATA2 PID

DATA1(0) = Zero length DATA1 packet

Idle State

This is the default state from power-up.

Setup Response State

The Function Interface Unit (FIU) receives a SETUP token with 8 bytes of data from the Host. The FIU stores the data in the FIFO, sends an ACK back to the host and asserts an RX_SETUP interrupt.

Hardware	Firmware
1. SETUP token, Data from Host	
2. ACK to Host	
3. Store data in FIFO	
4. Set RX SETUP → INT	
	5. Read UISR
	6. Read CSR0
	7. Read Byte Count
	8. Read FIFO
	9. Parse command data
	10. Write to H/FCAR0: a. If Control Read: set DIR, clear RX SETUP, fill FIFO, set TX Packet Ready in CAR0 b. If Control Write: clear DIR in CAR0 c. If no Data Stage: set Data End, clear DIR, set Force STALL in CAR0
	11. Set UIAR[EP0 INTACK] to clear the interrupt source

No-data Status Response State

The Function Interface Unit receives an IN token from the Host. The FIU responds with a zero length DATA1 packet until receiving an ACK from the host, then asserts a TX_COMPLETE interrupt.

Hardware	Firmware
1. IN token from Host	
2. Send DATA1(0)	
3. ACK from Host	
4. Set TX COMPLETE → INT	
	5. Read UISR
	6. Read CSR0
	7. If SET ADDRESS, program the new Address, set ADD_EN bit
	8. Clear TX_COMPLETE, clear Data End, set Force STALL in CAR0
	9. Set UIAR[EP0 INTACK]

Control Read Data Response State

The Function Interface Unit receives an IN token from the Host. The FIU responds with NAKs until TX_PACKET_READY is set. The FIU then sends the data in the FIFO upstream, retrying until it successfully receives an ACK from the host. Finally, the FIU clears the TX_PACKET_READY bit and asserts a TX_COMPLETE interrupt.

Hardware	Firmware
1. IN token from Host	
2. a. If TX Packet Ready = 1, send DATA0/DATA1 b. If TX Packet Ready = 0, send NAK	
3. ACK from Host	
4. Clear TX Packet Ready Set TX Complete → INT	
	5. Read UISR
	6. Read CSR0
	7. Clear TX COMPLETE in CAR0: a. If more data: fill FIFO, set TX Packet Ready, set DIR in CAR0 b. If no more data: set Force STALL, set DATA END in CAR0
	8. Set UIAR[EP0 INTACK] to clear interrupt source
Repeat steps 1 through 8	

Control Read Status Response State

The Function Interface Unit receives an OUT token from the Host with a zero length DATA1 packet. The FIU responds with a NAK until TX_COMPLETE is cleared. The FIU will then ACK the retried OUT token from the Host and assert an RX_OUT interrupt.

Hardware	Firmware
1. OUT token from Host	
2. DATA1(0) from Host	
3. TX Complete = 0 ? a. If yes, ACK to Host Set RX OUT → INT b. If no, NAK to Host	
	4. Read UISR
	5. Read CSR0
	6. Clear RX OUT, set Data End, set Force Stall in H/FCAR0. Note: A SETUP token will clear Data End, therefore, it is not cleared by FW in case Host retries.
	7. Set UIAR[EP0 INTACK] to clear interrupt source

Control Write Data Response State

The Function Interface Unit receives an OUT token from the Host with a DATA packet. The FIU places the incoming data into the FIFO, issues an ACK to the host, and asserts an RX_OUT interrupt.

Hardware	Firmware
1. OUT token from Host	
2. Put DATA0/DATA1 into FIFO	
3. ACK to Host	
4. Set RX OUT → INT	
	5. Read UISR
	6. Read CSR0
	7. Read FIFO
	8. Clear RX OUT If last data packet, set Force STALL, set DATA END.
	9. Set UIAR[EP0 INTACK] to clear the interrupt source

Repeat steps 1 through 9 until last DATA PACKET:

Control Write Status Response State

The Function Interface Unit receives an IN token from the Host. The FIU responds with a zero length DATA1 packet, retrying until it receives an ACK back from the Host. The FIU then asserts a TX_COMPLETE interrupt.

Hardware	Firmware
1. IN token from Host	
2. Send Data1(0)	
3. ACK from Host	
4. Set TX Complete → INT	
	5. Read UISR
	6. Read CSR0
	7. Clear TX COMPLETE, clear Data End, set Force STALL in CAR0
	8. Set UIAR[EP0 INTACK] to clear the interrupt source

Interrupt/Bulk IN Transfers at Function Endpoint

The firmware must first condition the endpoint through the Endpoint Control Register, FENDP1/2_CNTR:

Set endpoint direction: set EPDIR

Set interrupt or bulk: EPTYPE = 11 or 10

Enable endpoint: set EPEN

The Function Interface Unit receives an IN token from the Host. The FIU responds with NAKs until TX_PACKET_READY is set. The FIU then sends the data in the FIFO upstream, retrying until it successfully receives an ACK from the host. Finally, the FIU clears the TX_PACKET_READY bit and asserts a TX_COMPLETE interrupt.

1. Read UISR
2. Read FCSR1/2
3. Clear TX_COMPLETE

   If more data: fill FIFO, set TX Packet Ready

   Wait for TX_COMPLETE interrupt

   If no more data: set DATA END in FCAR1/2
4. Set UIAR[FEP1/2 INTACK] to clear the interrupt source

Interrupt/Bulk OUT Transfers at Function Endpoint EP1 and 2

The firmware must first condition the endpoint through the Endpoint Control Register, FENDP1/2_CNTR:

Set endpoint direction: clear EPDIR

Set interrupt or bulk: EPTYPE = 11 or 10

Enable endpoint: set EPEN

The Function Interface Unit receives an OUT token from the Host with a DATA packet. The FIU places the incoming data into the FIFO, issues an ACK to the host, and asserts an RX_OUT interrupt.

1. Read UISR
2. Read FCSR1/2
3. Read FIFO
4. Clear RX_OUT

   If more data:

   Wait for RX_OUT interrupt

   If no more data: set DATA END
5. Set UIAR[FEP1/2 INTACK] to clear the interrupt source

USB Registers

The following sections describe the registers of the AT43USB326's USB hub and function units.

Reading a bit for which the microcontroller does not have read access will yield a zero value result. Writing to a bit for which the microcontroller does not have write access has no effect.

Hub Address Register – HADDR

The USB hub contains an address register that contains the hub address assigned by the host. This Hub Address Register must be programmed by the microcontroller once it has received a SET_ADDRESS request from the host. The USB hardware uses the new address only after the status phase of the transaction is completed when the microcontroller has enabled the new address by setting bit 0 of the Global State Register. After power-up or reset, this register will contain the value of 0x00.

Hub Address Register – HADDR

Bit	7	6	5	4	3	2	1	0	
$1FEF	**SAEN**	**HADD6**	**HADD5**	**HADD4**	**HADD3**	**HADD2**	**HADD1**	**HADD0**	**HADDR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – SAEN: Single Address Enable**

The Single Address Enable bit allows the microcontroller to configure the AT43USB326 into a single address or a composite device. Once this capability is enabled, the hub endpoint 0 (HEP0) is converted from a control endpoint to a programmable function endpoint FEP3; all the endpoints would then operate on the single address.

- **Bit 6..0 – HADD6...0: Hub Address[6:0]**

Function Address Register – FADDR

The USB function contains an address register that contains the function address assigned by the host. This Function Address Register must be programmed by the microcontroller once it has received a SET_ADDRESS request from the host and completed the status phase of the transaction. After power up or reset, this register will contain the value of 0x00.

Function Address Register – FADDR

Bit	7	6	5	4	3	2	1	0	
$1FEE	**FEN**	**FADD6**	**FADD5**	**FADD4**	**FADD3**	**FADD2**	**FADD1**	**FADD0**	**FADDR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – FEN: Function Enable**

The Function Enable bit (FEN) allows the firmware to enable or disable the function endpoints. The firmware will set this bit after receipt of a reset through the hub, SetPortFeature[PORT_RESET]. Once this bit is set, the USB hardware passes to and from the host.

When the Single Address bit is set, the condition of FEN is ignored.

- **Bit 6..0 – FADD6...0: Function Address[6:0]**

Endpoint Registers

Hub Endpoint 0 Control Register – HENDP0_CR

Function Endpoint 0 Control Register – FENDP0_CR

Bit	7	6	5	4	3	2	1	0	
$1FE7	**EPEN**	**–**	**–**	**–**	**DTGLE**	**EPDIR**	**EPTYPE1**	**EPTYPE0**	**HENDP0_CR**
$24 ($44)	**EPEN**	**–**	**–**	**–**	**DTGLE**	**EPDIR**	**EPTYPE1**	**EPTYPE0**	**FENDP0_CR**
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – EPEN: Endpoint Enable**

0 = Disable endpoint

1 = Enable endpoint

- **Bit 6..4 – Reserved**

These bits are reserved in the AT43USB326 and will read as zero.

- **Bit 3 – DTGLE: Data Toggle**

Identifies DATA0 or DATA1 packets. This bit will automatically toggle and requires clearing by the firmware only in certain special circumstances.

- **Bit 2 – EPDIR: Endpoint Direction**

0 = Out

1 = In

- **Bit 1, 0 – EPTYPE: Endpoint Type**

These bits must be programmed as 0, 0.

Function Endpoint 1, 2 Control Register – FENDP1,2_CR

Bit	7	6	5	4	3	2	1	0	
$1FE4	**EPEN**	**–**	**–**	**–**	**DTGLE**	**EPDIR**	**EPTYPE1**	**EPTYPE0**	**FENDP1_CR**
$1FE3	**EPEN**	**–**	**–**	**–**	**DTGLE**	**EPDIR**	**EPTYPE1**	**EPTYPE0**	**FENDP2_CR**
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – EPEN: Endpoint Enable**

0 = Disable endpoint

1 = Enable endpoint

- **Bit 6..4 – Reserved**

These bits are reserved in the AT43USB326 and will read as zero.

- **Bit 3 – DTGLE: Data Toggle**

Identifies DATA0 or DATA1 packets. This bit will automatically toggle and requires clearing by the firmware only in certain special circumstances.

- **Bit 2 – EPDIR: Endpoint Direction**

0 = Out

1 = In

- **Bit 1, 0 – EPTYPE: Endpoint Type**

These bits program the type of endpoint.

Bit1	Bit0	Type
0	1	Isochronous
1	0	Bulk
1	1	Interrupt

Hub Endpoint 0 Data Register – HDR0

Function Endpoint 0..2 Data Register – FDR0..2

Bit	7	6	5	4	3	2	1	0	
$1FD7	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0	HDR0
$1FD5	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0	FDR0
$1FD4	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0	FDR1
$1FD3	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0	FDR2
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

This register is used to read data from or to write data to the Hub Endpoint 0 FIFO.

- **Bit 7..0 – FDAT7..0: FIFO Data**

Hub endpoint 1 has a single byte data register instead of a FIFO. This data register contains the hub and port status change bitmap. This data register is automatically updated by the USB hardware and is not accessible by the firmware. The bits in this register when read by the host will be:

Bit	7	6	5	4	3	2	1	0	
$	–	–	–	–	P3 SC	P2 SC	P1 SC	H SC	HDR1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7...4 – Reserved**

These bits are reserved in the AT43USB326 and will read as zero.

- **Bit 3 – P3 SC: Port 3 Status Change**
- **Bit 2 – P2 SC: Port 2 Status Change**
- **Bit 1 – P1 SC: Port 1 Status Change**
- **Bit 0 – H SC: Hub Status Change**

Hub Endpoint 0 Byte Count Register – HBYTE_CNT0

Function Endpoint 0..2 Byte Count Register – FBYTE_CNT0..2

The contents of these registers stores the number of bytes to be sent or that was received by USB Hub and Function endpoints. This count includes the 16-bit CRC. To get the actual byte count of the data, subtract the count in the register by 2. The maximum byte count supported by the AT43USB326 is 8 bytes. Hub endpoint 1 has no byte count register.

Bit	7	6	5	4	3	2	1	0	
Hub EP0 $1FCF	–	–	BYTCT5	BYTCT4	BYTCT3	BYTCT2	BYTCT1	BYTCT0	HBYTE_CNT0
Function EP0 $1FCD	–	–	BYTCT5	BYTCT4	BYTCT3	BYTCT2	BYTCT1	BYTCT0	FBYTE_CNT0
Function EP1 $1FCC	–	–	BYTCT5	BYTCT4	BYTCT3	BYTCT2	BYTCT1	BYTCT0	FBYTE_CNT1
Function EP2 $1FCB	–	–	BYTCT5	BYTCT4	BYTCT3	BYTCT2	BYTCT1	BYTCT0	FBYTE_CNT2
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..6 – Reserved**

These bits are reserved in the AT43USB326 and will read as zero.

- **Bit 5..0 – BYTCT5..0: Byte Count – Length of Endpoint Data Packet**

Hub Endpoint 0 Service Routine Register – HCSR0

Function Endpoint 0 Service Routine Register – FCSR0

Bit	7	6	5	4	3	2	1	0	
Function EP0 $1FDF	–	–	–	–	STALL SENT	RX SETUP	RX OUT PACKET	TX COMPLETE	HCSR0
Function EP0 $1FDD	–	–	–	–	STALL SENT	RX SETUP	RX OUT PACKET	TX COMPLETE	FCSR0
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..4 – Reserved**

These bits are reserved in the AT43USB326 and will read as zero.

- **Bit 3 – STALL SENT**

The USB hardware sets this bit after a STALL has been sent to the host. The firmware uses this bit when responding to a Get Status[Endpoint] request. It is a read only bit and that is cleared indirectly by writing a one to the STALL_SENT_ACK bit of the Control and Acknowledge Register.

- **Bit 2 – RX SETUP: Setup Packet Received**

This bit is used by control endpoints only to signal to the microcontroller that the USB hardware has received a valid SETUP packet and that the data portion of the packet is stored in the FIFO. The hardware will clear all other bits in this register while setting RX SETUP. If interrupt is enabled, the microcontroller will be interrupted when RX SETUP is set. After the completion of reading the data from the FIFO, firmware should clear this bit by writing a one to the RX_SETUP_ACK bit of the Control and Acknowledge Register.

- **Bit 1 – RX OUT PACKET**

The USB hardware sets this bit after it has stored the data of an OUT transaction in the FIFO. While this bit is set, the hardware will NAK all OUT tokens. The USB hardware will not overwrite the data in the FIFO except for an early set-up. RX OUT Packet is used for the following operations:

1. Control write transactions by a control endpoint.
2. OUT transaction with DATA1 PID to complete the status phase of a control endpoint.

Setting this bit causes an interrupt to the microcontroller if the interrupt is enabled. FW clears this bit after the FIFO contents have been read by writing a one to the RX_OUT_PACKET_ACK bit of the Control and Acknowledge Register.

- **Bit 0 – TX COMPL: Transmit Completed**

This bit is used by a control endpoint hardware to signal to the microcontroller that it has successfully completed certain transactions. TX Complete is set at the completion of a:

1. Control read data stage.
2. Status stage without data stage.
3. Status stage after a control write transaction.

This bit is read only and is cleared indirectly by writing a one to the TX_COMPLETE_ACK bit of the Control and Acknowledge Register.

Hub Endpoint 0 Control and Acknowledge Register – HCAR0

Function Endpoint 0 Control and Acknowledge Register – FCAR0

Bit	7	6	5	4	3	2	1	0	
Hub EP0 $1FA7	DIR	DATA END	FORCE STALL	TX PACKET READY	STALL_ SENT_ ACK	RX_ SETUP_ ACK	RX_OUT_ PACKET_ ACK	TX_ COMPLETE_ ACK	HCAR0
Function EP0 $1FDD	DIR	DATA END	FORCE STALL	TX PACKET READY	STALL_ SENT_ ACK	RX_ SETUP_ ACK	RX_OUT_ PACKET_ ACK	TX_ COMPLETE_ ACK	FCAR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – DIR: Control transfer direction**

It is set by the microcontroller firmware to indicate the direction of a control transfer to the USB hardware. The FW writes to this bit location after it receives an RX SETUP interrupt. The hardware uses this bit to determine the status phase of a control transfer.

0 = control write or no data stage

1 = control read

- **Bit 6 – DATA END**

When set to 1 by firmware, this bit indicate that the microcontroller has either placed the last data packet in FIFO, or that the microcontroller has processed the last data packet it expects from the Host. This bit is used by control endpoints only together with bit 4 (TX Packet Ready) to signal the USB hardware to go to the STATUS phase after the packet currently residing in the FIFO is transmitted. After the hardware completes the STATUS phase it will interrupt the microcontroller without clearing this bit.

- **Bit 5 – FORCE STALL**

This bit is set by the microcontroller to indicate a stalled endpoint. The hardware will send a STALL handshake as a response to the next IN or OUT token, or whenever there is a control transfer without a Data Stage.

The microcontroller sets this bit if it wants to force a STALL. A STALL is sent if any of the following condition is encountered:

1. An unsupported request is received.
2. The host continues to ask for data after the data is exhausted.
3. The control transfer has no data stage.

- **Bit 4 – TX PACKET READY: Transmit Packet Ready**

When set by the firmware, this bit indicates that the microcontroller has loaded the FIFO with a packet of data. This bit is cleared by the hardware after the USB Host acknowledges the packet. For ISO endpoints, this bit is cleared unconditionally after the data is sent.

This bit is used for the following operations:

1. Control read transactions by a control endpoint.
2. IN transactions with DATA1 PID to complete the status phase for a control endpoint, when this bit is zero but Data End set high (bit 4).
3. By a BULK IN or ISO IN or INT IN endpoint.

The microcontroller should write into the FIFO only if this bit is cleared. After it has completed writing the data, it should set this bit. This data can be of zero length.

Hardware clears this bit after it receives an ACK. If the interrupt is enabled and if the TX Complete bit is set, clearing the TX Packet Ready bit by the hardware causes an interrupt to the microcontroller.

- **Bit 3 – STALL_SENT_ACK: Acknowledge Stall Sent Interrupt**

Firmware sets this bit to clear STALL SENT, CSR bit 3. The 1 written in the CSRACK3 bit is not actually stored and thus does not have to be cleared.

- **Bit 2 – RX_SETUP_ACK: Acknowledge RX SETUP Interrupt**

Firmware sets this bit to clear RX SETUP, CSR bit2. The 1 written in the CSRACK2 bit is not actually stored and thus does not have to be cleared.

- **Bit 1 – RX_OUT_PACKET_ACK: Acknowledge RX OUT PACKET Interrupt**

Firmware sets this bit to clear RX OUT PACKET, CSR bit1. The 1 written in the CSRACK1 bit is not actually stored and thus does not have to be cleared.

- **Bit 0 – TX_COMPLETE_ACK: Acknowledge TX COMPLETE Interrupt**

Firmware sets this bit to clear TX COMPLETE, CSR bit0. The 1 written in the CSRACK0 bit is not actually stored and thus does not have to be cleared.

Function Endpoint 1,2 Service Routine Register – FCSR1,2

Bit	7	6	5	4	3	2	1	0	
Function EP1 $1FDC	–	–	–	–	STALL SENT	–	RX OUT PACKET	TX COMPLETE	FCSR1
Function EP2 $1FDB	–	–	–	–	STALL SENT	–	RX OUT PACKET	TX COMPLETE	FCSR2
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..4 – Reserved**

These bits are reserved in the AT43USB326 and will read as zero.

- **Bit 3 – STALL SENT**

The USB hardware sets this bit after a STALL has been sent to the host. The firmware uses this bit when responding to a Get Status[Endpoint] request. It is a read only bit and that is cleared indirectly by writing a one to the STALL_SENT_ACK bit of the Control and Acknowledge Register.

- **Bit 2 – Reserved**

This bit is reserved in the AT43USB326 and will read as zero.

- **Bit 1 – RX OUT PACKET**

The USB hardware sets this bit after it has stored the data of an OUT transaction in the FIFO. While this bit is set, the hardware will NAK all OUT tokens. The USB hardware will not overwrite the data in the FIFO except for an early set-up. RX OUT Packet is used by a BULK OUT or ISO OUT or INT OUT endpoint.

Setting this bit causes an interrupt to the microcontroller if the interrupt is enabled. FW clears this bit after the FIFO contents have been read by writing a one to the RX_SETUP_ACK bit of the Control and Acknowledge Register.

- **Bit 0 – TX COMPLETE: Transmit Completed**

This bit is used by the endpoint hardware to signal to the microcontroller that the IN transaction was completed successfully. This bit is read only and is cleared indirectly by writing a one to the TX_COMPLETE_ACK bit of the Control and Acknowledge Register.

Function Endpoint 1,2 Control and Acknowledge Register – FCAR1,2

Bit	7	6	5	4	3	2	1	0	
Function EP1 $1FA4	–	DATA END	FORCE STALL	TX PACKET RDY	STALL_SENT-ACK	–	RX_OUT_PACKET_ACK	TX_COMPLETE_ACK	FCAR1
Function EP2 $1FA3	–	DATA END	FORCE STALL	TX PACKET RDY	STALL_SENT-ACK	–	RX_OUT_PACKET_ACK	TX_COMPLETE-ACK	FCAR2
Read/Write	R	R/W	R/W	R/W	R/W	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – Reserved**

This bit is reserved in the AT43USB326 and will read as zero.

- **Bit 6 – DATA END**

When set to 1 by firmware, this bit indicate that the microcontroller has either placed the last data packet in FIFO, or that the microcontroller has processed the last data packet it expects from the Host.

- **Bit 5 – FORCE STALL**

This bit is set by the microcontroller to indicate a stalled endpoint. The hardware will send a STALL handshake as a response to the next IN or OUT token. The microcontroller sets this bit if it wants to force a STALL. A STALL is send if the host continues to ask for data after the data is exhausted.

- **Bit 4 – TX PACKET RDY: Transmit Packet Ready**

When set by the firmware, this bit indicates that the microcontroller has loaded the FIFO with a packet of data. This bit is cleared by the hardware after the USB Host acknowledges the packet. For ISO endpoints, this bit is cleared unconditionally after the data is sent.

The microcontroller should write into the FIFO only if this bit is cleared. After it has completed writing the data, it should set this bit. This data can be of zero length.

The hardware clears this bit after it receives an ACK. If the interrupt is enabled and if the TX Complete bit is set, clearing the TX Packet Ready bit by the hardware causes an interrupt to the microcontroller.

- **Bit 3 – STALL_SENT_ACK: Acknowledge Stall Sent Interrupt**

Firmware sets this bit to clear STALL SENT, CSR bit 3. The 1 written in the CSRACK3 bit is not actually stored and thus does not have to be cleared.

- **Bit 2 – Reserved**

This bit is reserved in the AT43USB326 and will read as zero.

- **Bit 1 – RX_OUT_PACKET_ACK: Acknowledge RX OUT PACKET Interrupt**

Firmware sets this bit to clear RX OUT PACKET, CSR bit1. The 1 written in the CSRACK1 bit is not actually stored and thus does not have to be cleared.

- **Bit 0 – TX_COMPLETE_ACK: Acknowledge TX COMPLETE Interrupt**

Firmware sets this bit to clear TX COMPLETE, CSR bit0. The 1 written in the CSRACK0 bit is not actually stored and thus does not have to be cleared.

USB Hub

The hub in a USB system provides for the electrical interface between USB devices and the host. The major functions that the hub must supports are:

- Connectivity
- Power management
- Device connect and disconnect
- Bus fault detection and recovery
- Full speed and low speed device support

A hub consists of two major components: a hub repeater and a hub controller. The hub repeater is responsible for:

- Providing upstream connectivity between the selected device and the Host
- Managing connectivity setup and tear-down
- Handling bus fault detection and recovery
- Detecting connect/disconnect on each port

The Hub Controller is responsible for:

- Hub enumeration
- Providing configuration information to the host
- Providing status of each port to the host
- Controlling each port per host command

The first two tasks of the hub are similar to that of a USB function and are described in detail in the following section. The descriptions will cover the features of the AT43USB326's hub and how to program it to make a USB-compliant hub.

Control transactions for the hub control endpoint proceed exactly the same way as those described for the embedded function. The operation of the hub's endpoint 1 is fully implemented in the hardware and does not need any firmware support. Any status changes within the hub will automatically update hub endpoint 1, which will be sent to the host at the next IN token that is addressed to it. If no change has occurred, the interrupt endpoint will respond with a NAK.

Hub General Registers

Global State Register – GLB_STATE

Bit	7	6	5	4	3	2	1	0	
$1FFB	–	–	–	SUSP FLG	RESUME FLG	RMWUPE	CONFG	HADD EN	GLB_STATE
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7...5 – Reserved Bits**

These bits are reserved in the AT43USB326 and will read as zeros.

- **Bit 4 – SUSP FLG: Suspend Flag**

This bit is set to 1 while the USB hardware is in the suspended state. This bit is a firmware read only bit. It is set and cleared by the USB hardware.

- **Bit 3 – RESUME FLGL Resume Flag**

When the USB hardware receives a resume signal from the upstream device it sets this bit. This bit will stay set until the USB hardware completes the downstream resume signaling. This bit is a firmware read only bit. It is set and cleared by the USB hardware.

- **Bit 2 – RMWUPE: Remote Wakeup Enable**

This bit is set if the host enables the hub's remote wakeup feature.

- **Bit 1 – CONFG: Configured**

This bit is set by firmware after a valid SET_CONFIGURATION request is received. It is cleared by a reset or by a SET_CONFIGURATION with a value of 0.

- **Bit 0 – HADD EN: Hub Address Enabled**

This bit is set by firmware after the status phase of a SET_ADDRESS request transaction so the hub will use the new address starting at the next transaction.

Hub Status Register

In the AT43USB326 overcurrent detection and port power switch control output processing is done in firmware. The hardware is designed so that various types of hubs are possible just through firmware modifications.

1. Hub local power status, bits 0 and 2, are optional features and apply to hubs that report on a global basis. If this feature is not used, both these bits should be programmed to 0. To use this feature, the firmware needs to know the status of the local power supply, which requires an input pin and extra internal or external circuitry.
2. Hub overcurrent status, bits 1 and 3, apply to self powered hubs with bus powered SIE only, or hubs that are programmable as self/bus powered. The firmware should clear these two bits to 0.

The firmware uses bits 1 and 3 to generate bit 0 of the Hub and Port Status Change Bitmap which is transmitted through the Hub Endpoint1 Data Register. Bit 0 of this register is a 1 whenever bit 1 or 3 of HSTATR is a 1.

Hub Status Register – HSTR

Bit	7	6	5	4	3	2	1	0	
$1FC7	–	–	–	–	OVLSC	LPSC	OVI	LPS	HSTR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..4 – Reserved**

These bits are reserved in the AT43USB326 and will read as zero.

- **Bit 3 – OVLSC: Overcurrent Status Change**

0 = No change has occurred on Overcurrent Indicator

1 = Overcurrent Indicator has changed

- **Bit 2 – LPSC: Hub Local Power Status Change**

0 = No change has occurred on Local Power Status

1 = Local Power Status has changed

- **Bit 1 – OVI: Overcurrent Indicator**

0 = All power operations normal

1 = An overcurrent exist on a hub wide basis

- **Bit 0 – LPS: Hub Local Power Status**

0 = Local power supply is good

1 = Local power supply is lost (inactive)

Hub Port Control Register – HPCON

Bit	7	6	5	4	3	2	1	0	
$1FC5	–	HPCON2	HPCON1	HPCON0	–	HPADD2	HPADD1	HPADD0	HPCON
Read/Write	R	R/W	R/W	R/W	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – Reserved**

This bits is reserved in the AT43USB326 and will read as zero.

- **Bit 6..4 – HPCON2..0: Hub Port Control Command**

These bits are written by firmware to control the port states upon receipt of a Host request.

Bit6	Bit5	Bit4	Action
0	0	0	Disable port
0	0	1	Enable port
0	1	0	Reset and enable port
0	1	1	Suspend port
1	0	0	Resume port

Disable Port = ClearPortFeature(PORT_ENABLE)

Action: USB hardware places addressed port in disabled state. Port 1 is placed in disabled state by firmware.

Enable Port = SetPort Feature(PORT_ENABLE)

Action: USB hardware places addressed port in enabled state. Firmware is responsible for placing Port 1 in enabled state.

Reset and Enable Port = SetPort Feature(PORT_RESET)

Action: USB hardware drives reset signaling through addressed port. USB hardware and firmware resets their embedded function registers to the default state.

Suspend Port = SetPortFeature(PORT_SUSPEND)

Action: USB hardware places port in idle state and stops propagating traffic through the addressed port. Firmware places Port 1 in suspend state by disabling its endpoints and placing the peripheral function in its low power state.

Resume Port = ClearPortFeature(PORT_SUSPEND)

Action: USB hardware sends resume signaling to addressed port and then enables port. Firmware takes the embedded function out of the suspend state and enables Port 1's endpoints.

- **Bit 3 – Reserved**

This bits is reserved in the AT43USB326 and will read as zero.

- **Bit 2..0 – HPCON2..0: Hub Port Address**

These bits define which port is being addressed for the command defined by bits [2:0].

Bit2	Bit1	Bit0	Port addresses
0	1	1	Port3
0	1	0	Port2

Selective Suspend and Resume

The host can selectively suspend and resume a port through the Set Port Feature (PORT_SUSPEND) and Clear Port Feature (PORT_SUSPEND).

A port enters the suspend state after the microcontroller interprets the suspend request and sets the appropriate bits of the Hub Port Control Register, HPCON. From this point on he hub repeater hardware is responsible for proper actions in placing Ports 2:3 in the suspend mode. For Port 1, the embedded function port, the hardware will stop responding to any normal bus traffic, but the microcontroller firmware must place all external circuitry associated with the function in the low-power state.

A port exits from the suspend state when the hub receives a Clear Port Feature (PORT_SUSPEND) or Set Port Feature (PORT_RESET). If the Clear Port Feature (PORT_SUSPEND) is directed towards Ports 2:3, the USB hardware drives a “K” downstream for at least 20 ms followed by a low speed EOP. It then places the port in the enabled state. A Clear Port Feature (PORT_SUSPEND) to Port 1 (the embedded function) causes the firmware to wait 20 ms, take the embedded function out of the suspended state and then enable the port.

The ports can also exit from the suspended state through a remote wakeup if this feature is enabled. For Ports 2:3, this means detection of a connect/disconnect or an upstream directed J to K signaling. Remote wakeup for the embedded function is initiated through an external interrupt at INT0.

Hub Port Status Register

The bits in this register are used by the microcontroller firmware when reporting a port's status through the Port Status Field, *wPortStatus*. Bits 3 (POCI) and 5 (PPSTAT) are used by the USB hardware and are the only two bits that the firmware should set or clear. All other bits should not be modified by the firmware.

Hub Port Status Register – HPSTAT2, 3

Bit	7	6	5	4	3	2	1	0	
Port1 $1FB8	–	LSP	PPSTAT	PRSTAT	POCI	PSSTAT	PESTAT	PCSTAT	HPSTAT1
Port2 $1FB9	–	LSP	PPSTAT	PRSTAT	POCI	PSSTAT	PESTAT	PCSTAT	HPSTAT2
Port3 $1FBA	–	LSP	PPSTAT	PRSTAT	POCI	PSSTAT	PESTAT	PCSTAT	HPSTAT3
Read/Write	R	R	R/W	R	R/W	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – Reserved**

This bit is reserved in the AT43USB326 and will read as zero.

- **Bit 6 – LSP: Low-speed Device Attached**

0 = Full-speed device attached to this port

1 = Slow-speed device attached to this port

Set to 0 for Port 1 (full-speed only). Set and cleared by the hardware upon detection of device at EOF2.

- **Bit 5 – PPSTAT: Port Power Status**

0 = Port is powered OFF

1 = Port is powered ON

Set to 1 for Port 1. Set and cleared based on present status of port power.

- **Bit 4 – PRSTAT: Port Reset Status**

0 = Reset signaling not asserted

1 = Reset signaling asserted

Set and cleared by the hardware as a result of initiating a port reset by Port Control Register.

- **Bit 3 – POCI: Port Overcurrent Indicator**

0 = Power normal

1 = Overcurrent exist on port

Set to 0 for Port 1. Set and cleared by firmware upon detection of an overcurrent or removal of an overcurrent.

- **Bit 2 – PSSTAT: Port Suspend Status**

0 = Port not suspended

1 = Port suspended

Set and cleared by the hardware as controlled through Port Control Register.

- **Bit 1 – PESTAT: Port Enable Status**

0 = Port is disabled

1 = Port is enabled

Set and cleared by the hardware as controlled through Port Control register.

- **Bit 0 – PCSTAT: Port Connect Status**

0 = No device on this port

1 = Device present on this port

Set to 1 for Port 1. Set and cleared by the hardware after sampling of connect status at EOF2.

Overcurrent Detect Register – UOVCER

Bit	7	6	5	4	3	2	1	0	
$1FF2	–	–	–	–	–	**PORT2**	–	–	**UOVCER**
Read/Write	R	R	R	R	R	R/W	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..3 – Reserved**

These bits are reserved in the AT43USB326 and will read as zero.

- **Bit 2 – PORT 2**

Setting this bit enables the hub to detect an overcurrent on Port 2 while the hub is in the suspend state.

- **Bit 1, 0 – Reserved**

These bits are reserved in the AT43USB326 and will read as zero.

Hub Port State Register – HPSTAT2, 3

Bit	7	6	5	4	3	2	1	0	
Port2 $1FA9	–	–	–	–	–	–	**DPSTATE**	**DMSTATE**	**PSTATE2**
Port3 $1FAA	–	–	–	–	–	–	**DPSTATE**	**DMSTATE**	**PSTATE3**
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

These registers contain the state of the ports' DP and DM pins, which will be sent to the host upon receipt of a GetBusState request.

- **Bit 7..2 – Reserved**

These bits are reserved in the AT43USB326 and will read as zero.

- **Bit 1 – DPSTATE: DPlus State**

Value of DP at last EOF. Set and cleared by hardware at EOF2.

Set to 1 for Port 1.

- **Bit 0 – DMSTATE: DMinus State**

Value of DM at last EOF. Set and cleared by hardware at EOF2.

Set to 0 for Port 1.

Hub Port Status Change Register – PSCR1..3

Bit	7	6	5	4	3	2	1	0	
Port1 $1FB0	–	–	–	RSTSC	POCIC	PSSC	PESC	PCSC	PSCR1
Port2 $1FB1	–	–	–	RSTSC	POCIC	PSSC	PESC	PCSC	PSCR2
Port3 $1FB2	–	–	–	RSTSC	POCIC	PSSC	PESC	PCSC	PSCR3
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

The microcontroller firmware uses the bits in this register to monitor when a port status change has occurred, which then gets reported to the host through the Port Change Field *wPortChange*.

Except for bit 3, the Port Overcurrent Indicator Change, the bits in this register are set by the USB hardware. Otherwise, the firmware should only clear these bits.

- **Bit 7..5 – Reserved**

These bits are reserved in the AT43USB326 and will read as zero.

- **Bit 4 – RSTSC: Port Reset Status Change**

0 = No change

1 = Reset complete

This bit is set by the USB hardware after it completes RESET signaling which is initiated when the Reset and Enable Port command is detected at the Port Control Register, HPCON. The firmware sends this command when it decodes a SetPortFeature(PORT_RESET) request from the host.

At EOF2 after the hardware completes the port reset, the hardware sets the Port Enable Status bit and clears the Port Reset Status bit of the Hub Port Status Register, HPSTAT. Cleared by firmware, ClearPortFeature(PORT_RESET).

- **Bit 3 – POCIC: Port Overcurrent Indicator Change**

0 = No change has occurred on Overcurrent Indicator

1 = Overcurrent Indicator has changed

This bit is relevant to hubs with individual overcurrent reporting only. The firmware sets this bit as a result of detecting overcurrent at the ports OVC# pin. The firmware clears bit through ClearPortFeature(PORT_OVER_CURRENT). For Port 1, this bit is always cleared.

- **Bit 2 – PSSC: Port Suspend Status Change**

0 = No change

1 = Resume completed

Port 2, 3 set by hardware upon completion of firmware initiated resume process. Port 1 set by firmware 20 ms after the next EOF2 after completion of resume process. RESUME signaling is initiated through global resume, selective resume and remote wakeup. Cleared by firmware via host request ClearPortFeature(PORT_SUSPEND).

- **Bit 1 – PESC: Port Enable/Disable Status Change**

0 = No change has occurred on Port Enable/Disable Status

1 = Port Enable/Disable status has changed

Set by hardware due to babble, physical disconnect or overcurrent except for Port 1 in which case it is set by hardware at EOF2 due to hardware events. Cleared by firmware via Host request ClearPortFeature(PORT_ENABLE).

- **Bit 0 – PCSC: Port Connect Status Change**

0 = No change has occurred on Current Connect Status

1 = Current Connect Status has changed

This bit is set by hardware at EOF2 after it detects a connect or disconnect at a port, except for Port 1. Hardware sets this bit for Port 5 after a hub reset. Cleared by firmware via Host request ClearPortFeature(PORT_CONNECTION).

Hub and Port Power Management

The embedded hub in a keyboard will most likely be a bus-powered hub even though the hardware of the AT43USB326 is designed to accommodate both types of hubs. Management of the downstream port power is also defined by the firmware: per port or global overcurrent sensing, individual or gang power switching. While the interface to the external power supply monitoring and switching is achieved through the microcontroller's GPIO pins, the USB hardware of the AT43USB326 contains the circuitry to handle all the possible combinations port power management tasks.

Overcurrent Sensing

If programmed for global overcurrent protection, the AT43USB326 is capable of detecting an overcurrent in its downstream port even while it is in the suspend state. This mode is enabled through the Overcurrent Detect Register, UOVCER, which controls the PD0 pin used for overcurrent input sensing.

1. **Global Overcurrent Protection** – In this mode, the Port Overcurrent Indicator and Port Overcurrent Indicator Change should be set to 0's. For the AT43USB326 an external solid state switch, such as the Micrel MIC2545-2, is required to switch power to the external USB ports. The FLG# output of the switch should be connected to PD0. When an overcurrent occurs, FLG# is asserted and the firmware should set the Hub Overcurrent Indicator and Hub Overcurrent Indicator Change and switch off power to the hub.
2. **Individual Port Over-current Protection** – The Hub Overcurrent Indicator and Hub Overcurrent Indicator Change bits should be set to 0's. One MIC2026-2 is required for the two USB ports. Each of the FLG# outputs of the MIC2026-2 should be connected to an unused microcontroller port. An overcurrent is indicated by assertion of FLG#. The firmware sets the corresponding port's Overcurrent Indicator and the Overcurrent Indicator Change bits and switches off power to the port. At the next IN token from the Host, the AT43USB326 reports the status change.

Port Power Switching

1. **Gang Power Switching** – One of the microcontroller I/O port pins must be programmed as an output to control the external switch, PWRN. Switch ON is requested by the USB Host through the SetPortFeature(PORT_POWER) request. Switch OFF is executed upon receipt of a ClearPortFeature(PORT_POWER) or upon detecting an overcurrent condition. The firmware clears the Power Control Bit. Only if all of the Power Control Bits of ports 2 and 3 are cleared should the firmware de-assert the PWRN pin.
2. **Individual Power Switching** – One microcontroller I/O port pin must be assigned for each USB port to control the external switch, PWRxN, where x = 2, 3. Each of the Power Control Bits controls one PWRxN.
3. **Multiple Ganged Overcurrent Protection** – Overcurrent sensing is grouped physically into one or more gangs, but reported individually.

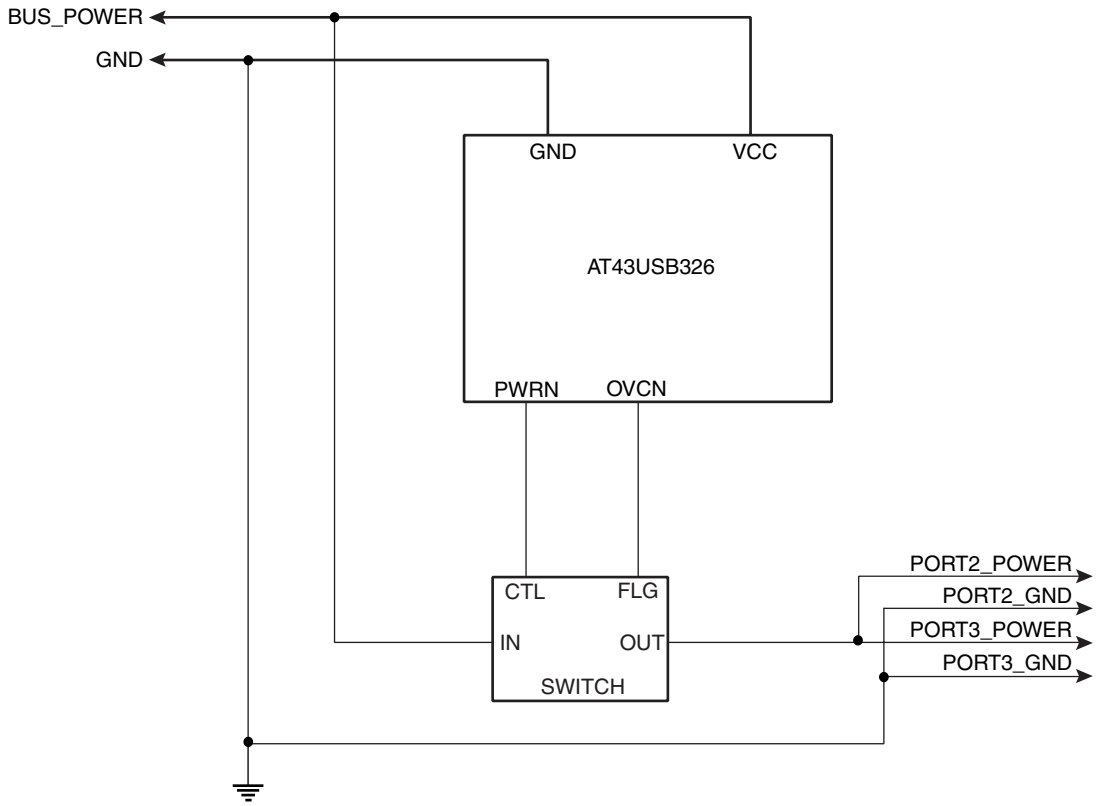
Figure 12 shows a simplified diagram of a power management circuit of an AT43USB326 based hub design with global overcurrent protection and ganged power switching for the two external downstream ports.

The over-current protection and port power switching for the AT43USB326 is best implemented in the so-called multiple gang. The Hub Characteristic's bits in the Hub Descriptor are set as follows:

- Logical Power Switching is Individual, D1D0 = 01
- Overcurrent Protection is Individual, D4D3 = 01

When an over-current condition at one of the external ports is detected, the over-current bits of both external ports are set. The host will switch off power to the external port, but the keyboard will continue to operate. The external port power is switched on as soon as one port's power is enabled, but switched off only when the power to both ports is disabled.

Figure 12. Port Power Management



Suspend and Resume

The AT43USB326 enters suspend only when requested by the USB host through bus inactivity for at least 3 ms. The USB hardware would detect this request, sets the GLB_SUSP bit of SPRSR, Suspend/Resume Register, and interrupts the microcontroller if the interrupt is enabled. The microcontroller should shut down any peripheral activity and enter the Power Down mode by setting the SE and SM bits of MCUCR and then executes the SLEEP instruction. The USB hardware shuts off the oscillator and PLL.

Global Resume

Global resume is signaled by a J to K state change on Port0. The USB hardware enables the oscillator/PLL, propagates the RESUME signaling, and sets the RSM bit of the SPRSR, which generates an interrupt. The microcontroller starts executing where it left off and services the interrupt. As part of the ISR, the firmware clears the GLB_SUSP bit.

Remote Wakeup

While the AT43USB326 is in global suspend, resume signaling is also possible through remote wakeup if the remote wakeup feature is enabled. Remote wakeup is defined as a port connect, port disconnect or resume signaling received at a downstream port or, in case of the embedded function, through an external interrupt.

A remote wakeup initiated at a downstream port is similar in many respects to a global resume. The USB hardware enables the oscillator/PLL, propagates the RESUME signaling, and sets the RSM bit of the SPRSR which generates an interrupt. The microcontroller starts executing where it left off and services the interrupt. As part of the ISR, the firmware clears the GLB_SUSP bit.

A remote wakeup from the embedded function is initiated through INT0 or the external interrupt, INT1, which enables the oscillator/PLL and the USB hardware. The USB hardware drives RESUME signaling and sets the FRMWUP and RSM bits of SPRSR which generates an interrupt to the microcontroller. The microcontroller starts executing where it left off and services the interrupt. As part of the ISR, the firmware clears the GLB SUSP bit.

At completion of RESUME signaling, the USB hardware sets the Port Suspend Status Change bits of the Hub Port Status Change Registers.

Selective Suspend and Resume

See section on Hub Port Control Register, HPCON.

Suspend and Resume Process

Global Suspend

The Host stops sending packets, the hardware detects this as global suspend signaling and stops all downstream signaling. Finally, the hardware asserts the GLB_SUSP interrupt.

Hardware	Firmware
	1.Host stops sending packets
2. Global suspend signaling detected	
3. Stop downstream signaling	
4. Set GBL SUS bit → interrupt	
	5. Shut down any peripheral activity
	6. Set Sleep Enable and Sleep Mode bits of MCUCR
	7. Set GPIO to low power state if required
	8. Set UOVCER bit 2
	9. Execute SLEEP instruction
10. SLEEP bit detected	
11. Shut off oscillator	

Global Resume

The Host resumes signaling, the hardware detects this as global resume and propagates this signaling to all downstream ports. Finally, the hardware enables the oscillator and asserts the RSM interrupt.

Hardware	Firmware
	1.Host resumes signaling
2. Resume signaling detected	
3. Propagate signaling downstream	
4. Enable oscillator	
5. Set RSM bit → interrupt	
	6. Reset RSM and GBL SUSP bits
	7. Restore GPIO states if required
	8. Clear UOVCER bit 2
	9. Enable peripheral activity

Remote Wake-up, Downstream Ports

The hardware detects a connect/disconnect/port resume and propagates resume signaling upstream. Finally, the hardware enables the oscillator and asserts the RSM interrupt.

Hardware	Firmware
1. Connect/disconnect/port resume detected	
2. Propagate resume signaling	
3. Enable Oscillator	
4. Set RSM bit → interrupt	
	5. Reset RSM and GBL SUSP bits
	6. Restore GPIO states if required
	7. Clear UOVCER bit 2
	8. Enable peripheral activity

Remote Wake-up, Embedded Function

The hardware detects an INT0/INT1 and propagates resume signaling upstream. Finally, the hardware enables the oscillator and asserts the RSM and FRWUP interrupts.

Hardware	Firmware
	1.External event activates INT0/INT1
2. Propagate resume signaling	
3. Enable Oscillator	
4. Set RSM and FRMWUP bits → interrupt	
	5. Clear GLB SUSP, RSM, FRMWUP bits
	6. Restore GPIO states if required
	7. Clear UOVCER bit 2
	8. Enable peripheral activity

Selective Suspend, Downstream Ports

Hardware	Firmware
	1. Set or Clear Port Feature PORT_SUSPEND decoded 2. Write HPCON[2:0] and HPADD[2:0] bits
3. Suspend or resume port per command	

Selective Suspend, Embedded Function

Hardware	Firmware
	1. Set Port Feature PORT_SUSPEND decoded 2. Disable Port 1's endpoints 3. Set GPIO to low power state if required

Selective Resume, Embedded Function

Hardware	Firmware
	1. Clear Port Feature PORT_SUSPEND decoded 2. Clear Port 1 suspend status bit 3. Restore GPIO states if required 4. Wait 23 ms, then set enable status bit and suspend change bit 5. Enable Port 1 endpoints
6. Send updated port status at next IN to endpoint1	

Electrical Specification

Absolute Maximum Ratings

Stresses beyond those listed below may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Table 13. Absolute Maximum Ratings

Symbol	Parameter	Condition	Min	Max	Unit
V_{CC5}	5V Power Supply			5.5	V
VI	DC input voltage		-0.3V	VCEXT+0.3 4.6 max	V
VO	DC output voltage		-0.3	VCEXT+0.3 4.6 max	V
TO	Operating temperature		-40	+125	°C
TS	Storage temperature		-65	+150	°C

Note: VCEXT is the voltage at CEXT1, CEXT2.

DC Characteristics

The values shown in this table are valid for TA = 0°C to 85°C, VCC = 4.4 to 5.25V, unless otherwise noted.

Table 14. Power Supply

Symbol	Parameter	Condition	Min	Max	Unit
V_{CC}	5V Power Supply		4.4	5.25	V
I_{CC}	5V Supply Current			40	mA
I_{CCS}	Suspended Device Current			600	uA

Table 15. USB Signals: DPx, DMx

Symbol	Parameter	Condition	Min	Max	Unit
V_{IH}	Input Level High (driven)		2.0		V
V_{IHZ}	Input Level High (floating)		2.7		V
V_{IL}	Input Level Low			0.8	V
V_{DI}	Differential Input Sensitivity	DPx and DMx	0.2		V
V_{CM}	Differential Common Mode Range		0.8	2.5	V
V_{OL1}	Static Output Low	RL of 1.5 kΩ to 3.6V		0.3	V
V_{OH1}	Static Output High	RL of 15 kΩ to GND	2.8	3.6	V
V_{CRS}	Output Signal Crossover		1.3	2.0	V
V_{IN}	Input Capacitance			20	pF

Table 16. PA, PB, PC, PD, PF

Symbol	Parameter	Condition	Min	Max	Unit
V_{OL2}	Output Low Level, PA, PB, PE[0:3]	IOL = 4 mA		0.5	V
RPU	PC Pull-up resistor current	V = 0	90	280	µA
V_{IL3}	Input Low Level, PC			0.3 VCEXT	V
V_{IH3}	Input High Level, PC		0.7 VCEXT		V
V_{IL4}	Input Low Level, PD[0,1]			0.3 VCEXT	V
V_{IH4}	Input High Level, PD[0,1]		0.7 VCEXT		V
V_{OL4}	Output Low Level, PD[0,1]	IOL = 4 mA		0.3 VCEXT	V
V_{OH4}	Output High Level, PD[0,1]	IOH = 4 mA	0.7 VCEXT		V
C	Input/Output capacitance	1 MHz		10	pF

Note: VCEXT is the voltage at CEXT1, CEXT2.

Table 17. Oscillator Signals: XTAL1, XTAL2

Symbol	Parameter	Condition	Min	Max	Unit
V_{LH}	OSC1 switching level		0.47	1.20	V
V_{HL}	OSC1 switching level		0.67	1.44	V
CX1	Input capacitance, XTAL1			10	pF
CX2	Output capacitance, XTAL2			10	pF
C12	OSC1/2 capacitance			5	pF
t_{SU}	Start-up time	6 MHz, fundamental		2	ms
DL	Drive level			50	µW

Note: XTAL2 must not be used to drive other circuitry.

Table 18. USB Driver Characteristics, Full-speed Operation

Symbol	Parameter	Condition	Min	Max	Unit
TR	Rise time	C_L = 50 pF	4	20	ns
TF	Fall time	C_L = 50 pF	4	20	ns
TRFM	TR/TF matching		90	110	%
ZDRV	Driver output resistance(1)	Steady state drive	28	44	Ω

Note: 1. With external 27Ω series resistor.

Figure 13. Full-speed Load

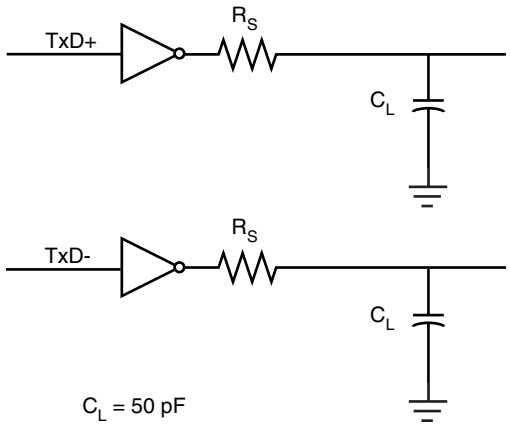


Table 19. USB Driver Characteristics, Low-speed Operation

Symbol	Parameter	Condition	Min	Max	Unit
TR	Rise time	CL = 200 - 600 pF	75	300	ns
TF	Fall time	CL = 200 - 600 pF	75	300	ns
TRFM	TR/TF matching		80	125	%

Figure 14. Low-speed Downstream Port Load

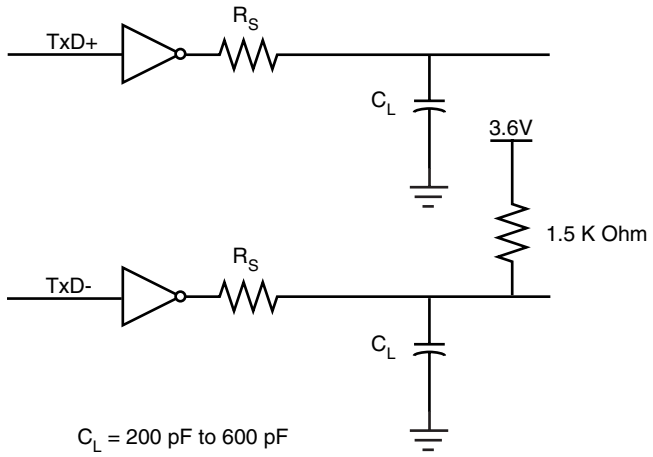


Table 20. USB Source Timings, Full-speed Operation

Symbol	Parameter	Condition	Min	Max	Unit
TDRATE	Full Speed Data Rate(1)	Average Bit Rate	11.97	12.03	Mb/s
TFRAME	Frame Interval(1)		0.9995	1.0005	ms
TRFI	Consecutive Frame Interval Jitter(1)	No clock adjustment		42	ns
TRFIADJ	Consecutive Frame Interval Jitter(1)	With clock adjustment		126	ns
TDJ1 TDJ2	Source Diff Driver Jitter To Next Transition For Paired Transitions		-2 -1	2 1	ns
TFDEOP	Source Jitter for Differential Transition to SEO Transitions		-2	5	ns
TDEOP	Differential to EOP Transition Skew		-2	5	ns
TJR1 TJR2	Receiver Data Jitter Tolerance To Next Transition For Paired Transitions		-18.5 -9	18.5 9	ns
TFEOPT	Source SEO interval of EOP		160	175	ns
TFEOPR	Receiver SEO interval of EOP		82		ns
TFST	Width of SEO interval during differential transition			14	ns

Note: 1. With 6.000 MHz, 100 ppm crystal.

Figure 15. Differential Data Jitter

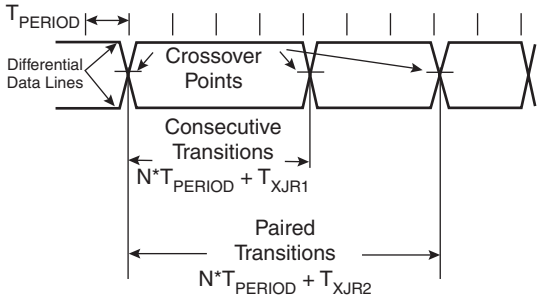


Figure 16. Differential-to-EOP Transition Skew and EOP Width

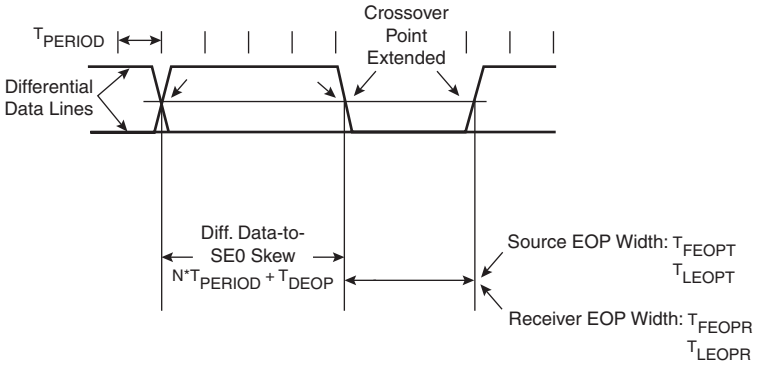


Figure 17. Receiver Jitter Tolerance

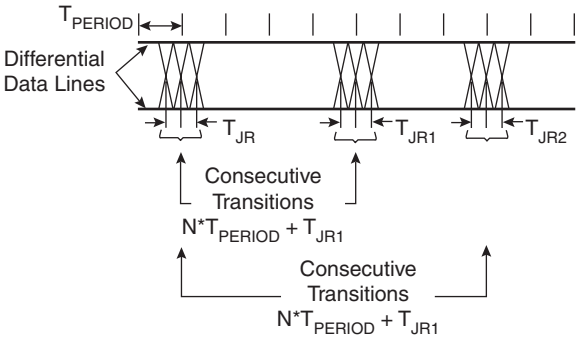


Table 21. Hub Timings, Full-speed Operation

Symbol	Parameter	Condition	Min	Max	Unit
THDD2	Hub Differential Data Delay without cable			44	ns
THDJ1 THDJ2	Hub Diff Driver Jitter to Next Transition for Paired Transitions		-3 -1	3 1	ns
TFSOP	Data Bit Width Distortion after SOP		-5	5	ns
TFEOPD	Hub EOP Delay Relative to THDD		0	15	ns
TFHESK	Hub EOP Output Width Skew		-15	15	ns

Table 22. Hub Timings, Low-speed Operation

Symbol	Parameter	Condition	Min	Max	Unit
TLHDD	Hub Differential Data Delay			300	ns
TLHDJ1 TLHDJ2 TLUHJ1 TLUHJ2	Downstr Hub Diff Driver Jitter to Next Transition, downst for Paired Transitions, downst to Next Transition, upstr for Paired Transitions, upstr		-45 -15 -45 -45	45 15 45 45	ns
TSOP	Data Bit Width Distortion after SOP		-60	60	ns
TLEOPD	Hub EOP Delay Relative to THDD		0	200	ns
TLHESK	Hub EOP Output Width Skew		-300	300	ns

Table 23. Hub Event Timings

Symbol	Parameter	Condition	Min	Max	Unit
TDCNN	Time to detect a downstream port connect event		2.5	2000	µs
TDDIS	Time to detect a disconnect event on downstream port Awake Hub Suspended Hub		2.5 2.5	2000 12000	µs
TURSM	Time from detecting downstream resume to rebroadcast			100	µs
TDRST	Duration of driving reset to a downstream device	Only for a SetPortFeature (PORT_RESET) request	10	20	µs
TDSPDEV	Time to evaluate device speed after reset		2.5	1000	µs
TURLK	Time to detect a long K from upstream		2.5	5.5	µs
TURLSEO	Time to detect a long SEO from upstream		2.5	5.5	µs
TURPSEO	Duration of repeating SEO upstream			23	FS bits
TUDEOP	Duration of sending SEO upstream after EOF1			2	FS bits

Figure 18. Hub Differential Delay, Differential Jitter and SOP Distortion

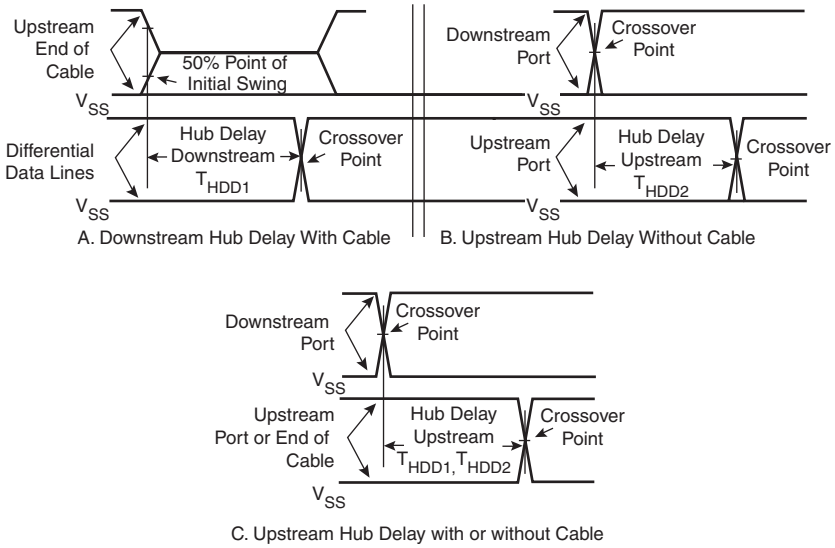
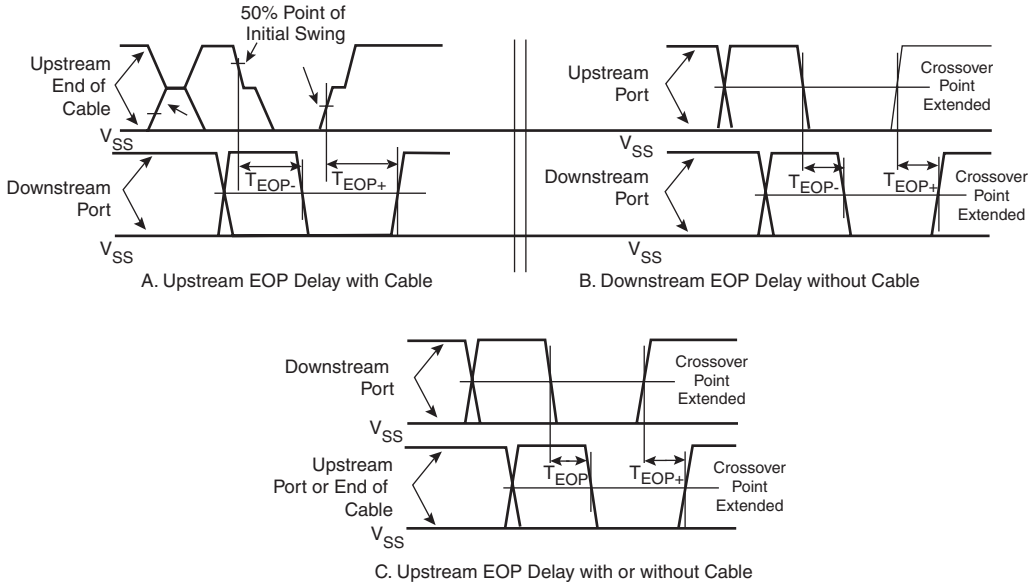


Figure 19. Hub EOP Delay and EOP Skew



Typical Application

The Atmel sample version of the AT43USB326 contains firmware that supports customization of the Vendor ID, Product ID, String Descriptor and the keyboard matrix. This information is stored in an external AT24C02A serial EEPROM. Data in the EEPROM is stored in the following format:

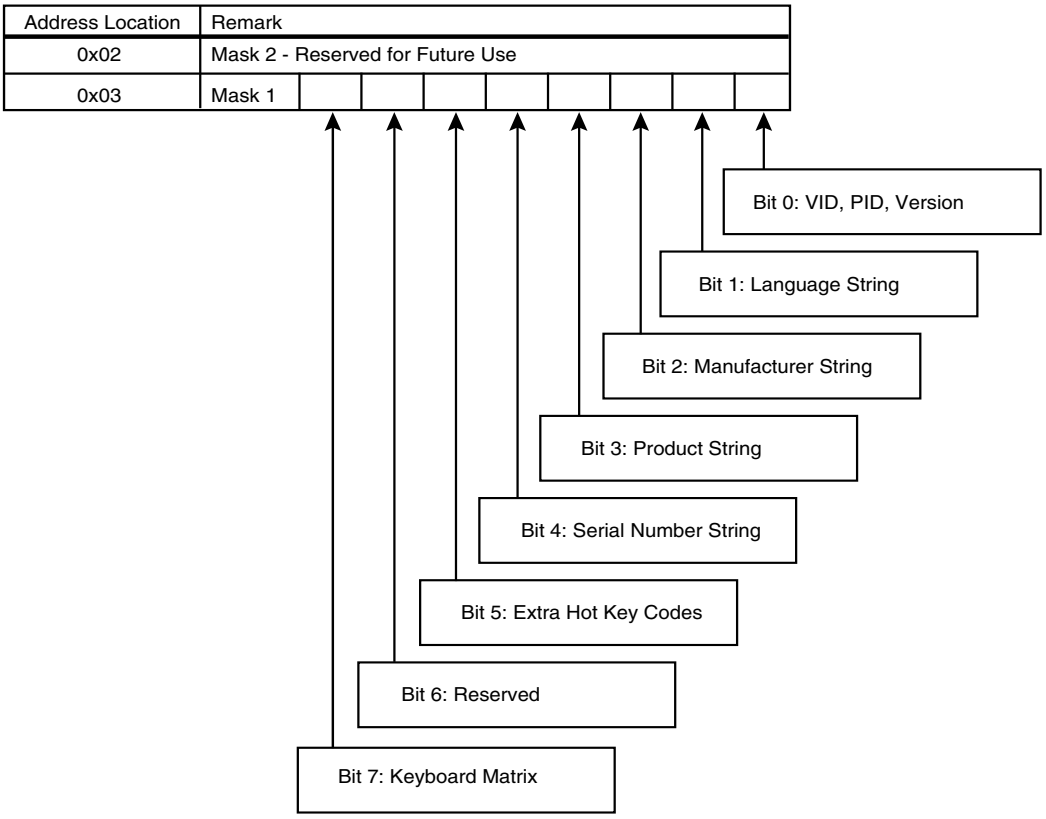
Address Range	Content
0x00 - 0x01	EEPROM's signature
0x02 - 0x03	Mask Bits
0x04 onwards	VID, PID and strings Initial Check Byte
0x70 - 0xFF	Keyboard matrix

Detailed Description

Address Location	Value/Content
0x00	0x17
0x01	0x5A

Mask Bits

The mask bits identify what features are stored in the EEPROM. A “1” in the mask bit location means that the feature is stored and should be read. A “0” means that the feature is not stored and should not be read.



Address locations 0x04 through 0x6E store the VID, PID, Release Number and String Descriptors if they are enabled by the mask bits in location 0x03. The data are stored sequentially as described below:

Table 24. VID and PID

VID Byte 1	VID Byte 2	PID Byte 1	PID Byte 2	Release # Byte 1	Release # Byte 2	Check Byte

Table 25. Language ID and Strings

Length *(N)*	Byte 1	Byte 2	...	Byte *N* - 1	Check Byte

Initial Check Byte

This check byte is located immediately after the last VID, PID or stings. It is computed as the truncated sum of Mask1 and all the string length specified.

Keyboard Matrix

The keyboard matrix is 144 bytes. If specified, its check byte is located in address 0x6F while the matrix itself is located from address 0x70 - 0xFF.

0x70 = Col 0, Row 0

0x71 = Col 0, Row 1

.....

0x78 = Col 1, Row 0

0x79 = Col 1, Row 1

.....

0xF8 = Col 17, Row 0

0xF9 = Col 17, Row 1

.....

0xFF = Col 17, Row 7

Example of SEEPROM Data

Table 26 through Table 34 illustrate the contents of a SEEPROM for a typical AT43USB326-based keyboard.

Table 26. Preamble

Address	Data	Description
00	17	Fixed ID
01	5A	Fixed ID
02	00	Reserved
03	BF	Mask bits

The mask bits in location 03 are used to indicate whether a particular function is supported or not.

0 = Function not supported

1 = Function supported

Bit 7 = Keyboard matrix

Bit 6 = Reserved. Must be set to 0

Bit 5 = Extra Hot Key USB codes

Bit 4 = Serial Number String

Bit 3 = Product String

Bit 2 = Manufacturer String

Bit 1 = Language ID String

Bit 0 = VendorID, ProductID, Version

Table 27. Vendor ID, Product ID, Version Number

Address	Data	Description
04	5E	Vendor ID
05	04	Vendor ID
06	1D	Product ID
07	00	Product ID
08	11	Version number
09	01	Version number
0A	91	Checksum

Table 28. Language ID

Address	Data	Description
0B	04	Number of bytes
0C	03	The next 3 bytes specifies the Language ID
0D	09	–
0E	10	–
0F	20	Checksum

Table 29. Manufacturer String

Address	Data	Description
10	0C	Number of bytes, 2 bytes per character
11	03	–
12	41	A
13	00	–
14	74	t
15	00	–
16	6D	m
17	00	–
18	65	e
19	00	–
1A	6C	l
1B	00	–
1C	02	Checksum

Table 30. Product String

Address	Data	Description
1D	12	Number of bytes
1E	03	–
1F	4D	M
20	00	–
21	4D	M
22	00	–
23	4B	K
24	00	–
25	42	B
26	00	–
27	44	D
28	00	–
29	48	H
2A	00	–
2B	55	U
2C	00	–

Table 30. Product String (Continued)

Address	Data	Description
2D	42	B
2E	00	–
2F	5F	Checksum

Table 31. Serial Number String

Address	Data	Description
30	08	Number of bytes
31	03	–
32	31	1
33	00	–
34	32	2
35	00	–
36	33	3
37	00	–
38	A1	Checksum

Table 32. Extra Hot Keys

Address	Data	Description
39	07	Number of bytes
3A	D0	KB matrix code
3B	23	First byte sent to host
3C	02	Second byte sent to host
3D	D1	KB matrix code
3E	8A	First byte sent to host
3F	01	Second byte sent to host
40	58	Checksum

Table 33. Miscellaneous Data

Address	Data	Description
41	F0	Checksum of Mask Byte + bytes in Language ID + bytes in Manuf String + bytes in Product String + bytes in Serial Number + bytes in extra hot keys (BF + 04 + 0C + 12 + 08 + 07 = F0)
42 through 6E	–	Don't cares. May be written with 0's
6F	2D	Checksum of keyboard matrix codes

Table 34. Keyboard Matrix

Address	Data	KB Row	KB Col	Description
70	28	0	0	KB Enter
71	31	1	0	KB \
72	50	2	0	Left Arrow
73	3F	3	0	F6
74	30	4	0	]
75	3E	5	0	F5
76	2A	6	0	Backspace
77	2E	7	0	=
78	36	0	1	Comma
79	07	1	1	D
7A	06	2	1	C
7B	25	3	1	KB 8
7C	0E	4	1	K
7D	20	5	1	KB 3
7E	08	6	1	E
7F	0C	7	1	I
80	63	0	2	KP Period
81	5E	1	2	KP 6
82	5B	2	2	KP 3
83	3D	3	2	F4
84	61	4	2	KP 9
85	3C	5	2	F3
86	4C	6	2	Delete
87	49	7	2	Insert
88	37	0	3	Period

Table 34. Keyboard Matrix (Continued)

Address	Data	KB Row	KB Col	Description
89	2F	1	3	[
8A	0F	2	3	L
8B	41	3	3	F8
8C	12	4	3	O
8D	40	5	3	F7
8E	2D	6	3	-
8F	26	7	3	9
90	05	0	4	B
91	0A	1	4	G
92	19	2	4	V
93	22	3	4	5
94	09	4	4	F
95	21	5	4	4
96	17	6	4	T
97	15	7	4	R
98	13	0	5	P
99	34	1	5	Quote
9A	38	2	5	/
9B	43	3	5	F10
9C	33	4	5	;
9D	42	5	5	F9
9E	51	6	5	Down Arrow
9F	27	7	5	0
A0	10	0	6	M
A1	0D	1	6	J
A2	11	2	6	N
A3	24	3	6	7
A4	0B	4	6	H
A5	23	5	6	6
A6	18	6	6	U
A7	1C	7	6	Y
A8	68	0	7	
A9	69	1	7	
AA	6A	2	7	
AB	2C	3	7	Space
AC	78	4	7	

Table 34. Keyboard Matrix (Continued)

Address	Data	KB Row	KB Col	Description
AD	4F	5	7	Right Arrow
AE	E7	6	7	Right GUI
AF	52	7	7	Up Arrow
B0	62	0	8	KP 0
B1	5D	1	8	KP 5
B2	5A	2	8	KP 2
B3	45	3	8	F12
B4	60	4	8	KP 8
B5	44	5	8	F11
B6	4D	6	8	End
B7	4A	7	8	Home
B8	59	0	9	KP 1
B9	5C	1	9	KP 4
BA	57	2	9	KP +
BB	65	3	9	Application
BC	5F	4	9	KP 7
BD	46	5	9	Print Screen
BE	55	6	9	KP *
BF	54	7	9	KP /
C0	1B	0	10	X
C1	1A	1	10	W
C2	16	2	10	S
C3	3B	3	10	F2
C4	1F	4	10	2
C5	3A	5	10	F1
C6	4E	6	10	Page Down
C7	4B	7	10	Page Up
C8	E4	0	11	Right Ctrl
C9	6B	1	11	
CA	D5	2	11	HK Calculator
CB	D4	3	11	HK Media
CC	E0	4	11	Left CTRL
CD	C0	5	11	HK Suspend
CE	39	6	11	Caps Lock
CF	D6	7	11	HK My Comp
D0	1D	0	12	Z

Table 34. Keyboard Matrix (Continued)

Address	Data	KB Row	KB Col	Description
D1	04	1	12	A
D2	6D	2	12	
D3	35	3	12	~
D4	14	4	12	Q
D5	29	5	12	Escape
D6	2B	6	12	Tab
D7	1E	7	12	1
D8	6E	0	13	
D9	C9	1	13	HK Vol Decrement
DA	70	2	13	
DB	71	3	13	
DC	B0	4	13	HK Mute
DD	72	5	13	
DE	73	6	13	
DF	74	7	13	
E0	E5	0	14	Right Shift
E1	C8	1	14	HK Vol Increment
E2	C1	2	14	HK Play/Pause
E3	C2	3	14	HK Scan Next
E4	E1	4	14	Left Shift
E5	48	5	14	Pause
E6	C4	6	14	HK Stop
E7	C3	7	14	HK Scan Previous
E8	D7	0	15	HK WWW
E9	E2	1	15	Left Alt
EA	E6	2	15	Right Alt
EB	53	3	15	KP Num Lock
EC	DF	4	15	HK Bookmark
ED	56	5	15	KP -
EE	D9	6	15	HK Search
EF	D8	7	15	HK Email
F0	DD	0	16	HK AC Stop
F1	74	1	16	
F2	DE	2	16	HK Refresh
F3	58	3	16	KP Enter
F4	DC	4	16	HK Forward

Table 34. Keyboard Matrix (Continued)

Address	Data	KB Row	KB Col	Description
F5	47	5	16	Scroll Lock
F6	DB	6	16	HK Back
F7	E3	7	16	Left GUI
F8	00	0	17	
F9	00	1	17	
FA	00	2	17	
FB	00	3	17	
FC	00	4	17	
FD	00	5	17	
FE	00	6	17	
FF	00	7	17	

Example Schematics

The following two pages show a schematic diagram of an AT43USB326 keyboard with an embedded hub and its BOM.

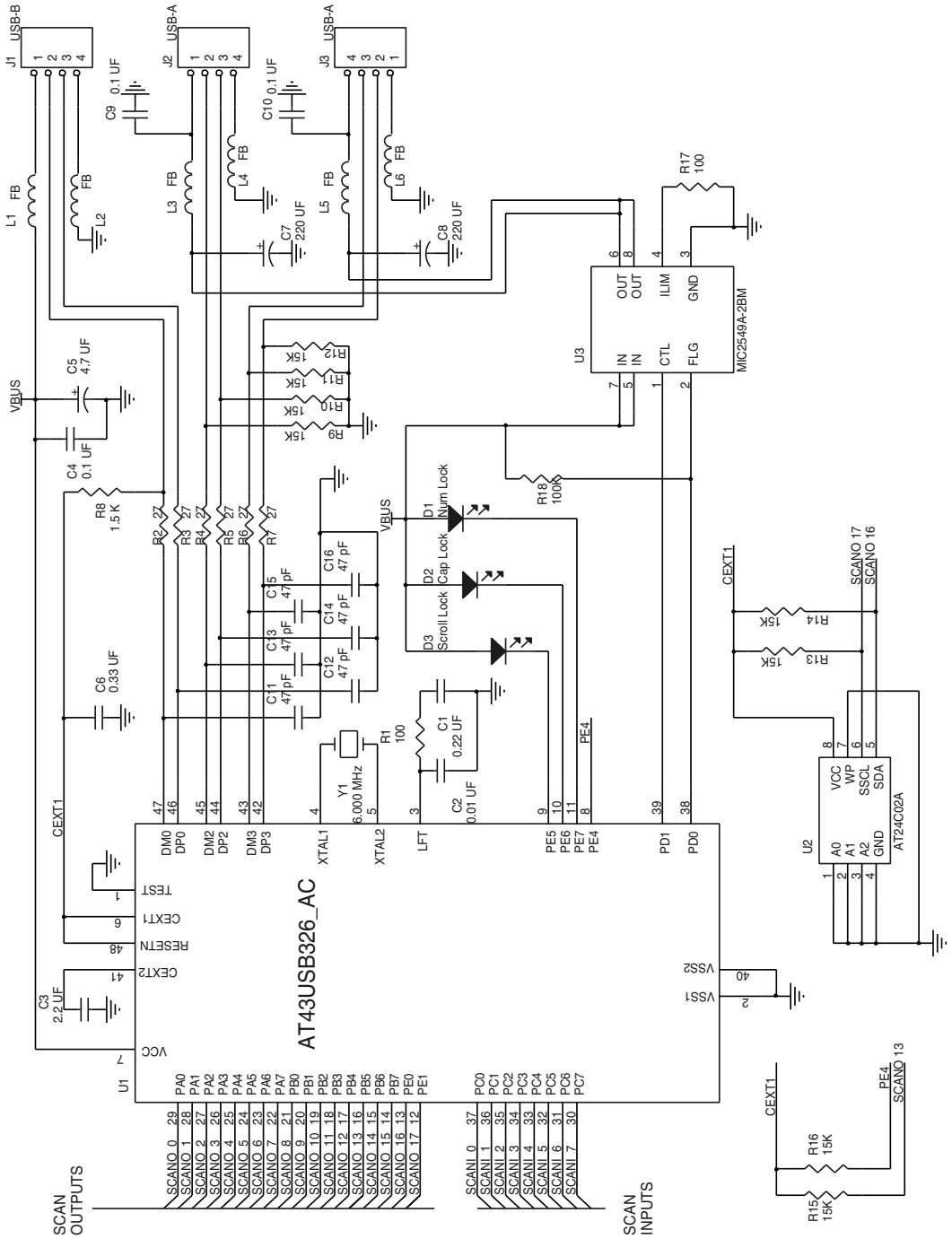


Table 35. AT43USB326 Keyboard/Hub Bill of Materials – May 2, 2002

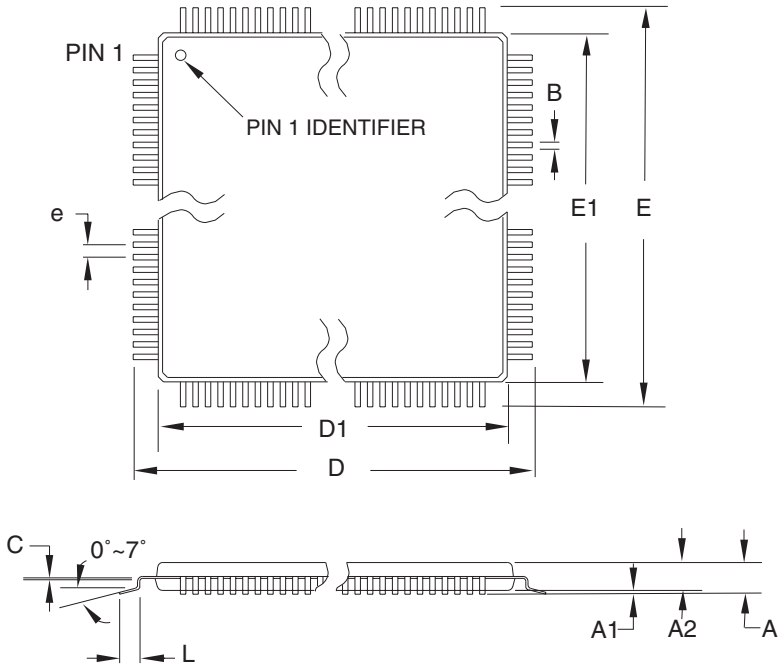
Item	Qty	Reference	Part	Supplier
1	1	C1	0.22 UF, cer, Kemet® C1812C224K5RAC	Newark® 99F6389
2	1	C2	0.01 UF, cer, Kemet C0805C103K5RAC	Newark 93F2330
3	3	C4,C9,C10	0.1 UF, cer, Kemet C1206C104M5UAC	Newark 89F5966
4	1	C5	4.7UF, el, Panasonic® ECE-V1ES4R7SR	Digikey® PCE3065CT-ND
5	1	C3	2.2 UF, cer, Panasonic ECJ-2YF1C225Z	Digikey PCC1851TR-ND
6	1	C6	0.33 UF, cer, Kemet C1210C334M5UAC	Newark
7	2	C8, C7	100 UFD, el, Panasonic ECE-V1AA101SP	Digikey PCE3176CT-ND
8	6	C11, C12, C13, C14, C15, C16	47 pF, cer, Kemet C1206C470J5GACTR	Newark 93F2371
9	1	J1	USB-B, series B connector	AMP 787780-1
10	2	J2, J3	USB-A, series A connector	AMP 787616-1
11	3	D1, D2, D3	LED, green	Jameco® 34606, XC2
12	6	L1, L2, L3, L4, L5, L6	Ferrite bead, Stewart HI 1806 N 750 R	Digikey 240-1-11-1-ND
13	1	R1, R17	100, Panasonic ERJ-GEYJ100	Digikey P-100-ACT-ND
14	6	R2, R3, R4, R5, R6, R7	27, Panasonic ERJ-GEYJ27	Digikey P-27-ACT-N
15	1	R8	1.5K, Panasonic ERJ-GEYJ1.5K	Digikey P-1.5K-ACT-ND
16	8	R9, R10, R11, R12, R13, R14, R15, R16	15K, Panasonic ERJ-GEYJ15K	Digikey P-15K-ACT-ND
17	1	R18	100K, Panasonic ERJ-GEYJ100K	Digikey P-100K-ACT-ND
18	1	U1	AT43USB326-AC	Atmel
19	1	U3	MIC2549A-2BM	Micrel
20	1	Y1	6.000 MHz crystal, CTS ATS060SM-T	Digikey CTX505CT-ND

Ordering Information

Ordering Code	Package	Operation Range
AT43USB326-AC	48 LQFP	Commercial (0°C to 70°C)
AT43USB326-AU	48 LQFP	Green, Industrial (-40°C to +85°C)

Packaging Information

48AA – LQFP



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.60	
A1	0.05	–	0.15	
A2	1.35	1.40	1.45	
D	8.75	9.00	9.25	
D1	6.90	7.00	7.10	Note 2
E	8.75	9.00	9.25	
E1	6.90	7.00	7.10	Note 2
B	0.17	–	0.27	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.50 TYP			

Notes:
1. This package conforms to JEDEC reference MS-026, Variation BBC.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.08 mm maximum.

10/5/2001

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**48AA,** 48-lead, 7 x 7 mm Body Size, 1.4 mm Body Thickness, 0.5 mm Lead Pitch, Low Profile Plastic Quad Flat Package (LQFP)	48AA	C

Errata Sheet

Errata (All Date Codes): Missed Watchdog Timer Reset

Problem

There is a synchronization problem between the watchdog clock and the AVR clock. Even though the clock inputs to both the watchdog timer and the AVR core are generated through the same crystal, the two clock sources are not going through the same PLL. The AVR is clocked at 12 MHz and the watchdog timer is clocked at 1MHz. The WDR (Watchdog Reset) instruction is a one-clock-cycle instruction. As such, when a watchdog timer reset occurs due to a WDR, the watchdog timer may miss the reset. This happens frequently if the AVR is clocked much faster than the watchdog timer.

Fix/Workaround

A workaround is to invoke the WDR repetitively to ensure that the watchdog timer will be able to receive the reset signal. If the AVR runs at 12 MHz, the WDR command must be invoked fourteen times back to back.

The following is the sample code for resetting and arming the watchdog timer, assuming the AVR is running at 12 MHz:

```
asm ( "ldi r16,15\n WDR\n WDR\n WDR\n WDR\n WDR\n WDR\n WDR\n WDR\n
WDR\n WDR\n WDR\n WDR\n out 0x21,r16 " );
```

To disarm and disable the watchdog, do the following:

```
asm ( "ldi r16,0x18\nldi r17,0x10\n\n out 0x21,r16\n out 0x21,r17 " );
```

Please note that if the AVR runs at 24 MHz, the WDR should be invoked twenty-six times.

Revision History

Doc. Rev.	Comments
3313B	• **Missing Data Correction:** Inserted the section “Watchdog Timer” on page 39. • **Additions:** Added an “Errata Sheet” on page 98, a “Revision History” on page 99, and a “Table of Contents” on page i.
3313C	• **Data Correction:** Corrected references to part number AT43USB325 to AT43USB326 in the section “Watchdog Timer” on page 39 and on “Example Schematics” on page 94.
3313D	• **Additions:** Added AT43USB326-AU part number to Ordering Information.

Table of Contents

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High-Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature


Printed on recycled paper.

3313D–USB–04/06