



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

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

Details

Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	8MHz
Connectivity	SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	32
Program Memory Size	8KB (4K × 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-LCC (J-Lead)
Supplier Device Package	44-PLCC (16.6x16.6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at90s8535-8ji

Email: info@E-XFL.COM

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

Architectural Overview

The fast-access register file concept contains 32×8 -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 the constant table look-up function. These added function registers are the 16-bit X-register, Y-register, and Z-register.



Figure 4. The AT90S8535 AVR RISC Architecture

The ALU supports arithmetic and logic functions between registers or between a constant and a register. Single register operations are also executed in the ALU. Figure 4 shows the AT90S8535 AVR 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 lowermost Data Space addresses (\$00 - \$1F), 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, A/D converters 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 two-stage pipeline. 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 in-system downloadable Flash memory.

With the relative jump and call instructions, the whole 4K 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 SP in the reset routine (before subroutines or interrupts are executed). The 10-bit stack pointer (SP) is read/write-accessible in the I/O space.

The 512 bytes 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.



Figure 5. 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 the different interrupts have a separate interrupt vector in the interrupt vector table at the beginning of the program



X-register, Y-register and Z-register

The registers R26..R31 have 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 in Figure 7.





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 the 32 generalpurpose 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.

In-System Programmable Flash Program Memory Flash Program Memory inter AT90S8535 contains 8K bytes On-chip, In-System Programmable Flash memory for program storage. Since all instructions are 16- or 32-bit words, the Flash is organized as 4K x 16. The Flash memory has an endurance of at least 1000 write/erase cycles. The AT90S8535 Program Counter (PC) is 12 bits wide, thus addressing the 4096 program memory addresses.

See page 99 for a detailed description on Flash data downloading.

See page 12 for the different program memory addressing modes.

AT90S/LS8535

Data Direct





A 16-bit data address is contained in the 16 LSBs of a 2-word instruction. Rd/Rr specify the destination or source register.

0

0

а

Data Space

\$0000

025F



10

Y OR Z - REGISTER

n

65

15

15

OP



Operand address is the result of the Y- or Z-register contents added to the address contained in six bits of the instruction word.

+

Data Indirect

Figure 14. Data Indirect Addressing



Operand address is the contents of the X-, Y-, or the Z-register.









Figure 21 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed and the result is stored back to the destination register.





The internal data SRAM access is performed in two System Clock cycles as described in Figure 22.







I/O Address (SRAM Address)	Name	Function
\$17 (\$37)	DDRB	Data Direction Register, Port B
\$16 (\$36)	PINB	Input Pins, Port B
\$15 (\$35)	PORTC	Data Register, Port C
\$14 (\$34)	DDRC	Data Direction Register, Port C
\$13 (\$33)	PINC	Input Pins, Port C
\$12 (\$32)	PORTD	Data Register, Port D
\$11 (\$31)	DDRD	Data Direction Register, Port D
\$10 (\$30)	PIND	Input Pins, Port D
\$0F (\$2F)	SPDR	SPI I/O Data Register
\$0E (\$2E)	SPSR	SPI Status Register
\$0D (\$2D)	SPCR	SPI Control Register
\$0C (\$2C)	UDR	UART I/O Data Register
\$0B (\$2B)	USR	UART Status Register
\$0A (\$2A)	UCR	UART Control Register
\$09 (\$29)	UBRR	UART Baud Rate Register
\$08 (\$28)	ACSR	Analog Comparator Control and Status Register
\$07 (\$27)	ADMUX	ADC Multiplexer Select Register
\$06 (\$26)	ADCSR	ADC Control and Status Register
\$05 (\$25)	ADCH	ADC Data Register High
\$04 (\$24)	ADCL	ADC Data Register Low

Table 1. AT90S8535 I/O Space (Continued)

Note: Reserved and unused locations are not shown in the table.

All AT90S8535 I/Os and peripherals 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 section for more details. When using the I/O specific commands IN and OUT, the I/O addresses \$00 - \$3F must be used. When addressing I/O registers as SRAM, \$20 must be added to these addresses. 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.

Some of the status flags are cleared by writing a logical "1" to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a "1" back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers \$00 to \$1F only.

The I/O and peripherals control registers are explained in the following sections.

• Bit 5 – TICIE1: Timer/Counter1 Input Capture Interrupt Enable

When the TICIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Input Capture Event Interrupt is enabled. The corresponding interrupt (at vector \$005) is executed if a capture-triggering event occurs on pin 20, PD6 (ICP) (i.e., when the ICF1 bit is set in the Timer/Counter Interrupt Flag Register [TIFR]).

• Bit 4 – OCIE1A: Timer/Counter1 Output CompareA Match Interrupt Enable

When the OCIE1A bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 CompareA Match interrupt is enabled. The corresponding interrupt (at vector \$006) is executed if a CompareA match in Timer/Counter1 occurs (i.e., when the OCF1A bit is set in the Timer/Counter Interrupt Flag Register [TIFR]).

• Bit 3 – OCIE1B: Timer/Counter1 Output CompareB Match Interrupt Enable

When the OCIE1B bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 CompareB Match interrupt is enabled. The corresponding interrupt (at vector \$007) is executed if a CompareB match in Timer/Counter1 occurs (i.e., when the OCF1B bit is set in the Timer/Counter Interrupt Flag Register [TIFR]).

• Bit 2 – TOIE1: Timer/Counter1 Overflow Interrupt Enable

When the TOIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Overflow Interrupt is enabled. The corresponding interrupt (at vector \$008) is executed if an overflow in Timer/Counter1 occurs (i.e., when the TOV1 bit is set in the Timer/Counter Interrupt Flag Register [TIFR]).

• Bit 1 - Res: Reserved Bit

This bit is a reserved bit in the AT90S8535 and always reads zero.

• Bit 0 – 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 \$009) is executed if an overflow in Timer/Counter0 occurs (i.e., when the TOV0 bit is set in the Timer/Counter Interrupt Flag Register [TIFR]).

Timer/Counter Interrupt Flag Register – TIFR

Bit	7	6	5	4	3	2	1	0	
\$38 (\$58)	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	-	TOV0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7 – OCF2: Output Compare Flag 2

The OCF2 bit is set (one) when compare match occurs between the Timer/Counter2 and the data in OCR2 (Output Compare Register2). OCF2 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF2 is cleared by writing a logical "1" to the flag. When the I-bit in SREG and OCIE2 (Timer/Counter2 Compare Match Interrupt Enable) and the OCF2 are set (one), the Timer/Counter2 Compare Match Interrupt is executed.

Bit 6 – TOV2: Timer/Counter2 Overflow Flag

The TOV2 bit is set (one) when an overflow occurs in Timer/Counter2. TOV2 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV2 is cleared by writing a logical "1" to the flag. When the SREG I-bit and TOIE2 (Timer/Counter2 Overflow Interrupt Enable) and TOV2 are set (one), the Timer/Counter2 Overflow Interrupt is executed. In up/down PWM mode, this bit is set when Timer/Counter1 advances from \$0000.





Timer/Counter0 Control Register – TCCR0



• Bits 7..3 – Res: Reserved Bits

These bits are reserved bits in the AT90S8535 and always read zero.

• Bits 2, 1, 0 – CS02, CS01, CS00: Clock Select0, Bits 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, Timer/Counter0 is stopped.			
0	0	1	СК			
0	1	0	СК/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 prescaled CK modes are scaled directly from the CK oscillator clock. If the external pin modes are used, the corresponding setup must be performed in the actual Data Direction Control Register (cleared to zero gives an input pin).

Timer Counter 0 – TCNT0



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.

TEMP, interrupts must be disabled during access from the main program (and from interrupt routines if interrupts are allowed from within interrupt routines).

- TCNT1 Timer/Counter1 Write: When the CPU writes to the high byte TCNT1H, the written data is placed in the TEMP register. Next, when the CPU writes the low byte TCNT1L, this byte of data is combined with the byte data in the TEMP register, and all 16 bits are written to the TCNT1 Timer/Counter1 register simultaneously. Consequently, the high byte TCNT1H must be accessed first for a full 16-bit register write operation.
- TCNT1 Timer/Counter1 Read: When the CPU reads the low byte TCNT1L, the data of the low byte TCNT1L is sent to the CPU and the data of the high byte TCNT1H is placed in the TEMP register. When the CPU reads the data in the high byte TCNT1H, the CPU receives the data in the TEMP register. Consequently, the low byte TCNT1L must be accessed first for a full 16-bit register read operation.

The Timer/Counter1 is realized as an up or up/down (in PWM mode) counter with read and write access. If Timer/Counter1 is written to and a clock source is selected, the Timer/Counter1 continues counting in the timer clock cycle after it is preset with the written value.

Compare Register – OCR1AH	Bit	15	14	13	12	11	10	9	8	
AND OCR1AL	\$2B (\$4B)	MSB								OCR1AH
	\$2A (\$4A)								LSB	OCR1AL
		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	
Timer/Counter1 Output										
Compare Register – OCR1BH	Bit	15	14	13	12	11	10	9	8	
AND OCR1BL	\$29 (\$49)	MSB								OCR1BH
	\$28 (\$48)								LSB	OCR1BL
		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 output	compare	e registe	rs are 16	6-bit read	d/write re	egisters.			

The Timer/Counter1 Output Compare registers contain the data to be continuously compared with Timer/Counter1. Actions on compare matches are specified in the Timer/Counter1 Control and Status registers. A compare match only occurs if Timer/Counter1 counts to the OCR value. A software write that sets TCNT1 and OCR1A or OCR1B to the same value does not generate a compare match.

A compare match will set the compare interrupt flag in the CPU clock cycle following the compare event.

Since the Output Compare Registers (OCR1A and OCR1B) are 16-bit registers, a temporary register (TEMP) is used when OCR1A/B are written to ensure that both bytes are updated simultaneously. When the CPU writes the high byte, OCR1AH or OCR1BH, the data is temporarily stored in the TEMP register. When the CPU writes the low byte,



Timer/Counter1 Output

Asynchronous Operation of Timer/Counter2

When Timer/Counter2 operates asynchronously, some considerations must be taken.

- Warning: When switching between asynchronous and synchronous clocking of Timer/Counter2, the timer registers TCNT2, OCR2 and TCCR2 might get corrupted. A safe procedure for switching clock source is:
 - 1. Disable the Timer/Counter2 interrupts OCIE2 and TOIE2.
 - 2. Select clock source by setting AS2 as appropriate.
 - 3. Write new values to TCNT2, OCR2 and TCCR2.
 - 4. To switch to asynchronous operation: Wait for TCN2UB, OCR2UB and TCR2UB.
 - 5. Clear the Timer/Counter2 interrupt flags.
 - 6. Clear the TOV2 and OCF2 flags in TIFR.
 - 7. Enable interrupts, if needed.
- When writing to one of the registers TCNT2, OCR2 or TCCR2, the value is transferred to a temporary register and latched after two positive edges on TOSC1. The user should not write a new value before the contents of the temporary register have been transferred to their destination. Each of the three mentioned registers have their individual temporary register. For example, writing to TCNT2 does not disturb an OCR2 write in progress. To detect that a transfer to the destination register has taken place, an Asynchronous Status Register (ASSR) has been implemented.
- When entering a Power Save Mode after having written to TCNT2, OCR2 or TCCR2, the user must wait until the written register has been updated if Timer/Counter2 is used to wake up the device. Otherwise, the MCU will go to sleep before the changes have had any effect. This is extremely important if the Output Compare2 interrupt is used to wake up the device; Output Compare is disabled during write to OCR2 or TCNT2. If the write cycle is not finished (i.e., the user goes to sleep before the OCR2UB bit returns to zero), the device will never get a compare match and the MCU will not wake up.
- If Timer/Counter2 is used to wake up the device from Power Save Mode, precautions must be taken if the user wants to re-enter Power Save Mode: The interrupt logic needs one TOSC1 cycle to be reset. If the time between wake up and re-entering Power Save Mode is less than one TOSC1 cycle, the interrupt will not occur and the device will fail to wake up. If the user is in doubt whether the time before re-entering Power Save is sufficient, the following algorithm can be used to ensure that one TOSC1 cycle has elapsed:
 - 1. Write a value to TCCR2, TCNT2 or OCR2.
 - 2. Wait until the corresponding Update Busy flag in ASSR returns to zero.
 - 3. Enter Power Save Mode.
- When the asynchronous operation is selected, the 32 kHz oscillator for Timer/Counter2 is always running, except in Power-down mode. After a power-up reset or wake-up from power-down, the user should be aware of the fact that this oscillator might take as long as one second to stabilize. The user is advised to wait for at least one second before using Timer/Counter2 after power-up or wake-up from power-down. The content of all Timer/Counter2 registers must be considered lost after a wake-up from power-down due to the unstable clock signal upon start-up, regardless of whether the oscillator is in use or a clock signal is applied to the TOSC pin.
- Description of wake-up from Power Save Mode when the timer is clocked asynchronously: When the interrupt condition is met, the wake-up process is started on the following cycle of the timer clock, that is, the timer is always advanced by at





is selected (the CHR9 bit in the UART Control Register, UCR is set), the TXB8 bit in UCR is transferred to bit 9 in the Transmit shift register.

On the baud rate clock following the transfer operation to the shift register, the start bit is shifted out on the TXD pin. Then follows the data, LSB first. When the stop bit has been shifted out, the shift register is loaded if any new data has been written to the UDR during the transmission. During loading, UDRE is set. If there is no new data in the UDR register to send when the stop bit is shifted out, the UDRE flag will remain set until UDR is written again. When no new data has been written and the stop bit has been present on TXD for one bit length, the TX Complete flag (TXC) in USR is set.

The TXEN bit in UCR enables the UART Transmitter when set (one). When this bit is cleared (zero), the PD1 pin can be used for general I/O. When TXEN is set, the UART Transmitter will be connected to PD1, which is forced to be an output pin regardless of the setting of the DDD1 bit in DDRD.

Data Reception Figure 42 shows a block diagram of the UART Receiver.



Figure 42. UART Receiver

The receiver front-end logic samples the signal on the RXD pin at a frequency 16 times the baud rate. While the line is idle, one single sample of logical "0" will be interpreted as the falling edge of a start bit and the start bit detection sequence is initiated. Let sample

Operation

The ADC converts an analog input voltage to a 10-bit digital value through successive approximation. The minimum value represents AGND and the maximum value represents the voltage on the AREF pin minus one LSB. The analog input channel is selected by writing to the MUX bits in ADMUX. Any of the eight ADC input pins ADC7..0 can be selected as single-ended inputs to the ADC.

The ADC can operate in two modes – Single Conversion and Free Running. In Single Conversion Mode, each conversion will have to be initiated by the user. In Free Running Mode, the ADC is constantly sampling and updating the ADC Data Register. The ADFR bit in ADCSR selects between the two available modes.

The ADC is enabled by setting the ADC Enable bit, ADEN in ADCSR. Input channel selections will not go into effect until ADEN is set. The ADC does not consume power when ADEN is cleared, so it is recommended to switch off the ADC before entering power-saving sleep modes.

A conversion is started by writing a logical "1" to the ADC Start Conversion bit, ADSC. This bit stays high as long as the conversion is in progress and will be set to zero by hardware when the conversion is completed. If a different data channel is selected while a conversion is in progress, the ADC will finish the current conversion before performing the channel change.

The ADC generates a 10-bit result, which is presented in the ADC data register, ADCH and ADCL. When reading data, ADCL must be read first, then ADCH, to ensure that the content of the data register belongs to the same conversion. Once ADCL is read, ADC access to data register is blocked. This means that if ADCL has been read and a conversion completes before ADCH is read, neither register is updated and the result from the conversion is lost. Then ADCH is read, ADC access to the ADCH and ADCL register is re-enabled.

The ADC has its own interrupt that can be triggered when a conversion completes. When ADC access to the data registers is prohibited between reading of ADCH and ADCL, the interrupt will trigger even if the result is lost.

Prescaling

Figure 46. ADC Prescaler



The successive approximation circuitry requires an input clock frequency between 50 kHz and 200 kHz to achieve maximum resolution. If a resolution of lower than 10 bits is required, the input clock frequency to the ADC can be higher than 200 kHz to achieve a





higher sampling rate. See "ADC Characteristics" on page 75 for more details. The ADC module contains a prescaler, which divides the system clock to an acceptable ADC clock frequency.

The ADPS2..0 bits in ADCSR are used to generate a proper ADC clock input frequency from any CPU frequency above 100 kHz. The prescaler starts counting from the moment the ADC is switched on by setting the ADEN bit in ADCSR. The prescaler keeps running for as long as the ADEN bit is set and is continuously reset when ADEN is low.

When initiating a conversion by setting the ADSC bit in ADCSR, the conversion starts at the following rising edge of the ADC clock cycle.

A normal conversion takes 13 ADC clock cycles. In certain situations, the ADC needs more clock cycles for initialization and to minimize offset errors. Extended conversions take 25 ADC clock cycles and occur as the first conversion after the ADC is switched on (ADEN in ADCSR is set).

The actual sample-and-hold takes place 1.5 ADC clock cycles after the start of a normal conversion and 13.5 ADC clock cycles after the start of an extended conversion. When a conversion is complete, the result is written to the ADC data registers and ADIF is set. In Single Conversion Mode, ADSC is cleared simultaneously. The software may then set ADSC again and a new conversion will be initiated on the first rising ADC clock edge. In Free Running Mode, a new conversion will be started immediately after the conversion completes, while ADSC remains high. Using Free Running Mode and an ADC clock frequency of 200 kHz gives the lowest conversion time with a maximum resolution, 65 μ s, equivalent to 15 kSPS. For a summary of conversion times, see Table 26.







• ADC9..0: ADC Conversion result

These bits represent the result from the conversion. \$000 represents analog ground and \$3FF represents the selected reference voltage minus one LSB.

Scanning Multiple Channels Since change of analog channel always is delayed until a conversion is finished, the Free Running Mode can be used to scan multiple channels without interrupting the converter. Typically, the ADC Conversion Complete interrupt will be used to perform the channel shift. However, the user should take the following fact into consideration: The interrupt triggers once the result is ready to be read. In Free Running Mode, the next conversion will start immediately when the interrupt triggers. If ADMUX is changed after the interrupt triggers, the next conversion has already started and the old setting is used.

ADC Noise Canceling Techniques

Digital circuitry inside and outside the AT90S8535 generates EMI that might affect the accuracy of analog measurements. If conversion accuracy is critical, the noise level can be reduced by applying the following techniques:

- 1. The analog part of the AT90S8535 and all analog components in the application should have a separate analog ground plane on the PCB. This ground plane is connected to the digital ground plane via a single point on the PCB.
- 2. Keep analog signal paths as short as possible. Make sure analog tracks run over the analog ground plane and keep them well away from high-speed switching digital tracks.
- 3. The AV_{CC} pin on the AT90S8535 should be connected to the digital V_{CC} supply voltage via an LC network as shown in Figure 50.
- 4. Use the ADC noise canceler function to reduce induced noise from the CPU.
- 5. If some Port A pins are used as digital outputs, it is essential that these do not switch while a conversion is in progress.

Figure 50. ADC Power Connections



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).
Port A	Port A is an 8-bit bi-directional I/O port.
	Three I/O memory address locations are allocated for 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.
	All port pins have individually selectable pull-up resistors. The Port A output buffers can sink 20 mA and thus drive LED displays directly. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.
	Port A has an alternate function as analog inputs for the ADC. If some Port A pins are configured as outputs, it is essential that these do not switch when a conversion is in progress. This might corrupt the result of the conversion.
	During Power-down mode, the Schmitt trigger of the digital input is disconnected. This allows analog signals that are close to $V_{\rm CC}/2$ to be present during power-down without causing excessive power consumption.

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	ſ							
	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	ſ							
	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	r
	Initial Value	N/A								

The Port A Input Pins address (PINA) is not a register; 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 D Schematics

Note that all port pins are synchronized. The synchronization latches are, however, not shown in the figures.





Figure 62. Port D Schematic Diagram (Pin PD1)







- 1. Set BS to "1". This selects high data.
- 2. Give WR a negative pulse. This starts programming of the data byte. RDY/BSY goes low.
- 3. Wait until RDY/BSY goes high to program the next byte.

(See Figure 69 for signal waveforms.)

The loaded command and address are retained in the device during programming. For efficient programming, the following should be considered:

- The command needs only be loaded once when writing or reading multiple memory locations.
- Address high byte needs only be loaded before programming a new 256-word page in the Flash.
- Skip writing the data value \$FF, that is, the contents of the entire Flash and EEPROM after a Chip Erase.

These considerations also apply to EEPROM programming and Flash, EEPROM and signature byte reading.



Figure 68. Programming the Flash Waveforms

AT90S/LS8535

Serial Programming Characteristics

Figure 73. Serial Programming Timing



Table 43. Serial Programming Characteristics, $T_A = -40^{\circ}C$ to $85^{\circ}C$, $V_{CC} = 2.7 - 6.0V$ (unless otherwise noted)

Symbol	Parameter	Min	Тур	Max	Units
1/t _{CLCL}	Oscillator Frequency ($V_{CC} = 2.7 - 6.0V$)	0		4.0	MHz
t _{CLCL}	Oscillator Period ($V_{CC} = 2.7 - 4.0V$)	250.0			ns
1/t _{CLCL}	Oscillator Frequency ($V_{CC} = 4.0 - 6.0V$)	0		8.0	MHz
t _{CLCL}	Oscillator Period ($V_{CC} = 4.0 - 6.0V$)	125.0			ns
t _{SHSL}	SCK Pulse Width High	2.0 t _{CLCL}			ns
t _{SLSH}	SCK Pulse Width Low	2.0 t _{CLCL}			ns
t _{ovsH}	MOSI Setup to SCK High	t _{CLCL}			ns
t _{SHOX}	MOSI Hold after SCK High	2.0 t _{CLCL}			ns
t _{SLIV}	SCK Low to MISO Valid	10.0	16.0	32.0	ns

Table 44. Minimum Wait Delay after the Chip Erase instruction

Symbol	3.2V	3.6V	4.0V	5.0V
t _{WD_ERASE}	18 ms	14 ms	12 ms	8 ms

Table 45.	Minimum	Wait Delay	after Writing	g a Flash o	or EEPROM Location
-----------	---------	------------	---------------	-------------	--------------------

	,	•		
Symbol	3.2V	3.6V	4.0V	5.0V
t _{wd_prog}	9 ms	7 ms	6 ms	4 ms



Packaging Information

44A

44-lead, Thin (1.0mm) Plastic Quad Flat Package (TQFP), 10x10mm body, 2.0mm footprint, 0.8mm pitch. Dimension in Millimeters and (Inches)* JEDEC STANDARD MS-026 ACB



*Controlling dimension: millimetter

REV. A 04/11/2001



44J

44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC) Dimensions in Milimeters and (Inches)* JEDEC STANDARD MS-018 AC



*Controlling dimensions: Inches

