



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	4MHz
Connectivity	SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	32
Program Memory Size	8KB (4K x 16)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 6V
Data Converters	-
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C
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/at90s8515-4ji

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 ALU (Arithmetic Logic Unit) 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-, Y-, and Z-register.

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 AT90S8515 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 such 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 Programmable 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 16-bit Stack Pointer (SP) is read/write-accessible in the I/O space.

The 512-byte 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.



In the different addressing modes these address registers have functions as fixed dis-
placement, 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 i The AT90S8515 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 AT90S8515 Program Counter (PC) is 12 bits wide, thus addressing the 4096 program memory addresses.

See page 86 for a detailed description of Flash data downloading.

See page 13 for the different program memory addressing modes.



AT90S8515

d

31

	two additional clock cycles is used per byte. This has the following effect: Data transfer instructions take two extra clock cycles, whereas interrupt, subroutine calls and returns will need four clock cycles more than specified in the instruction set manual.
	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.
	The direct addressing reaches the entire data space.
	The Indirect with Displacement mode features 63 address locations reached 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, the 512 bytes of internal data SRAM, and the 64K bytes of optional external data SRAM in the AT90S8515 are all accessible through all these addressing modes.
	See the next section for a detailed description of the different addressing modes.
Program and Data Addressing Modes	The AT90S8515 AVR RISC microcontroller supports powerful and efficient addressing modes for access to the program memory (Flash) and data memory (SRAM, Register file and I/O memory). This section describes the different addressing modes supported by the AVR architecture. In the figures, OP means the operation code part of the instruction word. To simplify, not all figures show the exact location of the addressing bits.
Register Direct, Single Register RD	Figure 9. Direct Single Register Addressing
	15 4 0 OP d

The operand is contained in register d (Rd).



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





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

Figure 14. Data Indirect Addressing



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

Figure 15. Data Indirect Addressing with Pre-decrement





Data Indirect with Displacement

Data Indirect

Data Indirect with Pre-

decrement

Figure



Address Hex	Name	Function				
\$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				

Table 1.	AT90S8515	I/O Space ((Continued)
----------	-----------	-------------	-------------

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

All AT90S8515 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 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.

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.

Status Register – SREG

The AVR status register (SREG) at I/O space location \$3F (\$5F) is defined as:



• 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 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

Sleep Modes	To enter the sleep modes, the SE bit in MCUCR must be set (one) and a SLEEP instruc- tion 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.
Idle Mode	When the SM bit is cleared (zero), the SLEEP instruction forces the MCU into the Idle Mode, stopping the CPU but allowing Timer/Counters, Watchdog and the interrupt system to continue operating. This enables the MCU to wake up from external triggered interrupts as well as internal ones like Timer Overflow interrupt and Watchdog reset. If wake-up from the Analog Comparator interrupt is not required, the Analog Comparator can be powered down by setting the ACD-bit in the Analog Comparator Control and Status Register (ACSR). This will reduce power consumption in Idle Mode. When the MCU wakes up from Idle Mode, the CPU starts program execution immediately.
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 and the Watchdog (if enabled) continue operating. Only an external reset, a Watchdog reset (if enabled), or 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.





EEPROM Read/Write Access

The EEPROM access registers are accessible in the I/O space.

The write access time is in the range of 2.5 - 4 ms, depending on the V_{CC} voltages. A self-timing function, however, lets the user software detect when the next byte can be written. If the user code contains code that writes the EEPROM, some precaution must be taken. In heavily filtered power supplies, V_{CC} is likely to rise or fall slowly on power-up/down. This causes the device for some period of time to run at a voltage lower than specified as minimum for the clock frequency used. CPU operation under these conditions is likely cause the program counter to perform unintentional jumps and eventually execute the EEPROM write code. To secure EEPROM integrity, the user is advised to use an external under-voltage reset circuit in this case.

In order to prevent unintentional EEPROM writes, a specific write procedure must be followed. Refer to the description of the EEPROM Control Register for details on this.

When the EEPROM is written, the CPU is halted for two clock cycles before the next instruction is executed. When the EEPROM is read, the CPU is halted for four clock cycles before the next instruction is executed.

EEPROM Address Register – EEARH and EEARL

Bit	15	14	13	12	11	10	9	8	
\$1F (\$3F)	-	-	-	-	-	-	-	EEAR8	EEARH
\$1E (\$3E)	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
-	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R/W	
	R/W								
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The EEPROM address registers (EEARH and EEARL) specify the EEPROM address in the 512-byte EEPROM space for AT90S8515. The EEPROM data bytes are addressed linearly between 0 and 512.

EEPROM Data Register – EEDR



• Bits 7..0 – EEDR7..0: EEPROM Data

For the EEPROM write operation, the EEDR register contains the data to be written to the EEPROM in the address given by the EEAR register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

EEPROM Control Register – EECR

Bit	7	6	5	4	3	2	1	0	
\$1C (\$3C)	-	-	-	-	-	EEMWE	EEWE	EERE	EECR
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 AT90S8515 and will always read as zero.

<u>AIMEL</u>

Prevent EEPROM Corruption

During periods of low V_{CC}, the EEPROM data can be corrupted because the supply voltage is too low for the CPU and the EEPROM to operate properly. These issues are the same as for board level systems using the EEPROM and the same design solutions should be applied.

An EEPROM data corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the EEPROM requires a minimum voltage to operate correctly. Second, the CPU itself can execute instructions incorrectly if the supply voltage for executing instructions is too low.

EEPROM data corruption can easily be avoided by following these design recommendations (one is sufficient):

- Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This is best done by an external low V_{CC} Reset Protection circuit, often referred to as a Brown-out Detector (BOD). Please refer to application note AVR 180 for design considerations regarding power-on reset and low-voltage detection.
- Keep the AVR core in Power-down Sleep mode during periods of low V_{CC}. This will prevent the CPU from attempting to decode and execute instructions, effectively protecting the EEPROM registers from unintentional writes.
- 3. Store constants in Flash memory if the ability to change memory contents from software is not required. Flash memory cannot be updated by the CPU and will not be subject to corruption.

AT90S8515

SPI Data Register – SPDR



The SPI Data Register is a read/write register used for data transfer between the register file and the SPI Shift Register. Writing to the register initiates data transmission. Reading the register causes the Shift Register Receive buffer to be read.





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 1 denote the first zero-sample. Following the 1-to-0 transition, the receiver samples the RXD pin at samples 8, 9 and 10. If two or more of these three samples are found to be logical "1"s, the start bit is rejected as a noise spike and the receiver starts looking for the next 1-to-0 transition.

If, however, a valid start bit is detected, sampling of the data bits following the start bit is performed. These bits are also sampled at samples 8, 9 and 10. The logical value found in at least two of the three samples is taken as the bit value. All bits are shifted into the Transmitter Shift register as they are sampled. Sampling of an incoming character is shown in Figure 40.





When the stop bit enters the receiver, the majority of the three samples must be "1" to accept the stop bit. If two or more samples are logical "0"s, the Framing Error (FE) flag in the UART Status Register (USR) is set. Before reading the UDR register, the user should always check the FE bit to detect framing errors.

Whether or not a valid stop bit is detected at the end of a character reception cycle, the data is transferred to UDR and the RXC flag in USR is set. UDR is in fact two physically separate registers, one for transmitted data and one for received data. When UDR is read, the Receive Data register is accessed, and when UDR is written, the Transmit Data register is accessed. If 9-bit data word is selected (the CHR9 bit in the UART Control Register, UCR is set), the RXB8 bit in UCR is loaded with bit 9 in the Transmit Shift register when data is transferred to UDR.

If, after having received a character, the UDR register has not been read since the last receive, the OverRun (OR) flag in USR is set. This means that the last data byte shifted into the shift register could not be transferred to UDR and has been lost. The OR bit is buffered and is updated when the valid data byte in UDR is read. Thus, the user should always check the OR bit after reading the UDR register in order to detect any overruns if the baud rate is high or CPU load is high.

When the RXEN bit in the UCR register is cleared (zero), the receiver is disabled. This means that the PD0 pin can be used as a general I/O pin. When RXEN is set, the UART Receiver will be connected to PD0, which is forced to be an input pin regardless of the setting of the DDD0 bit in DDRD. When PD0 is forced to input by the UART, the PORTD0 bit can still be used to control the pull-up resistor on the pin.

When the CHR9 bit in the UCR register is set, transmitted and received characters are 9 bits long, plus start and stop bits. The ninth data bit to be transmitted is the TXB8 bit in UCR register. This bit must be set to the wanted value before a transmission is initiated by writing to the UDR register. The ninth data bit received is the RXB8 bit in the UCR register.



Baud Rate	1	MHz	%Error	1.8432	MHz	%Error	2	MHz	%Error	2.4576	MHz	%Error
2400	UBRR=	25	0.2	UBRR=	47	0.0	UBRR=	51	0.2	UBRR=	63	0.0
4800	UBRR=	12	0.2	UBRR=	23	0.0	UBRR=	25	0.2	UBRR=	31	0.0
9600	UBRR=	6	7.5	UBRR=	11	0.0	UBRR=	12	0.2	UBRR=	15	0.0
14400	UBRR=	3	7.8	UBRR=	7	0.0	UBRR=	8	3.7	UBRR=	10	3.1
19200	UBRR=	2	7.8	UBRR=	5	0.0	UBRR=	6	7.5	UBRR=	7	0.0
28800	UBRR=	1	7.8	UBRR=	3	0.0	UBRR=	3	7.8	UBRR=	4	6.3
38400	UBRR=	1	22.9	UBRR=	2	0.0	UBRR=	2	7.8	UBRR=	3	0.0
57600	UBRR=	0	7.8	UBRR=	1	0.0	UBRR=	1	7.8	UBRR=	2	12.5
76800	UBRR=	0	22.9	UBRR=	1	33.3	UBRR=	1	22.9	UBRR=	1	0.0
115200	UBRR=	0	84.3	UBRR=	0	0.0	UBRR=	0	7.8	UBRR=	0	25.0
Baud Rate	3.2768	MHz	%Error	3.6864	MHz	%Error	4	MHz	%Error	4.608	MHz	%Error
2400	UBRR=	84	0.4	UBRR=	95	0.0	UBRR=	103	0.2	UBRR=	119	0.0
4800	UBRR=	42	0.8	UBRR=	47	0.0	UBRR=	51	0.2	UBRR=	59	0.0
9600	UBRR=	20	1.6	UBRR=	23	0.0	UBRR=	25	0.2	UBRR=	29	0.0
14400	UBRR=	13	1.6	UBRR=	15	0.0	UBRR=	16	2.1	UBRR=	19	0.0
19200	UBRR=	10	3.1	UBRR=	11	0.0	UBRR=	12	0.2	UBRR=	14	0.0
28800	UBRR=	6	1.6	UBRR=	7	0.0	UBRR=	8	3.7	UBRR=	9	0.0
38400	UBRR=	4	6.3	UBRR=	5	0.0	UBRR=	6	7.5	UBRR=	7	6.7
57600	UBRR=	3	12.5	UBRR=	3	0.0	UBRR=	3	7.8	UBRR=	4	0.0
76800	UBRR=	2	12.5	UBRR=	2	0.0	UBRR=	2	7.8	UBRR=	3	6.7
115200	UBRR=	1	12.5	UBRR=	1	0.0	UBRR=	1	7.8	UBRR=	2	20.0
												•
Baud Rate	7.3728	MHz	%Error	8	MHz	%Error	9.216	MHz	%Error	11.059	MHz	%Error
2400	UBRR=	191	0.0	UBRR=	207	0.2	UBRR=	239	0.0	UBRR=	287	-
4800	UBRR=	95	0.0	UBRR=	103	0.2	UBRR=	119	0.0	UBRR=	143	0.0
9600	UBRR=	47	0.0	UBRR=	51	0.2	UBRR=	59	0.0	UBRR=	71	0.0
14400	UBRR=	31	0.0	UBRR=	34	0.8	UBRR=	39	0.0	UBRR=	47	0.0
19200	UBRR=	23	0.0	UBRR=	25	0.2	UBRR=	29	0.0	UBRR=	35	0.0
28800	UBRR=	15	0.0	UBRR=	16	2.1	UBRR=	19	0.0	UBRR=	23	0.0
38400	UBRR=	11	0.0	UBRR=	12	0.2	UBRR=	14	0.0	UBRR=	17	0.0
57600	UBRR=	7	0.0	UBRR=	8	3.7	UBRR=	9	0.0	UBRR=	11	0.0
76800	UBRR=	5	0.0	UBRR=	6	7.5	UBRR=	7	6.7	UBRR=	8	0.0
115200	UBRR=	3	0.0	UBRR=	3	7.8	UBRR=	4	0.0	UBRR=	5	0.0

Table 17. UBRR Settings at Various Crystal Frequencies

UART BAUD Rate Register – UBRR



The UBRR register is an 8-bit read/write register that specifies the UART Baud Rate according to the equation on the previous page.

Analog Comparator

The Analog Comparator compares the input values on the positive input PB2 (AIN0) and negative input PB3 (AIN1). When the voltage on the positive input PB2 (AIN0) is higher than the voltage on the negative input PB3 (AIN1), the Analog Comparator Output (ACO) is set (one). The comparator's output can be set to trigger the Timer/Counter1 Input Capture function. In addition, the comparator can trigger a separate interrupt, exclusive to the Analog Comparator. The user can select interrupt triggering on comparator output rise, fall or toggle. A block diagram of the comparator and its surrounding logic is shown in Figure 41.





Analog Comparator Control and Status Register – ACSR

Bit	7	6	5	4	3	2	1	0	
\$08 (\$28)	ACD	-	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
Read/Write	R/W	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	N/A	0	0	0	0	0	

• Bit 7 – ACD: Analog Comparator Disable

When this bit is set (one), the power to the Analog Comparator is switched off. This bit can be set at any time to turn off the Analog Comparator. This will reduce power consumption in active and idle mode. When changing the ACD bit, the Analog Comparator interrupt must be disabled by clearing the ACIE bit in ACSR. Otherwise an interrupt can occur when the bit is changed.

• Bit 6 - Res: Reserved Bit

This bit is a reserved bit in the AT90S8515 and will always read as zero.

Bit 5 – ACO: Analog Comparator Output

ACO is directly connected to the comparator output.

• Bit 4 – ACI: Analog Comparator Interrupt Flag

This bit is set (one) when a comparator output event triggers the interrupt mode defined by ACI1 and ACI0. The Analog Comparator Interrupt routine is executed if the ACIE bit is set (one) and the I-bit in SREG is set (one). ACI is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ACI is cleared by writing a logical "1" to the flag. Observe however, that if another bit in this register is modified





Figure 47. Port B Schematic Diagram (Pins PB2 and PB3)



Figure 48. Port B Schematic Diagram (Pin PB4)









Port C

Port C is an 8-bit bi-directional I/O port. 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 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 C output buffers can sink 20 mA and thus drive LED displays directly. When pins PC0 to PC7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated.

The Port C pins have alternate functions related to the optional external data SRAM. Port C can be configured to be the high-order address byte during accesses to external data memory. When Port C is set to the alternate function by the SRE (external SRAM enable) bit in the MCUCR (MCU Control Register), the alternate settings override the Data Direction Register.

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								
Initial Value	0	0	0	0	0	0	0	0	



Port C Schematics

Note that all port pins are synchronized. The synchronization latch is, however, not shown in the figure.





Port D

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors.

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.

The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated.

Some Port D pins have alternate functions as shown in Table 23.

Port Pin	Alternate Function
PD0	RXD (UART Input Line)
PD1	TXD (UART Output Line)
PD2	INT0 (External interrupt 0 Input)
PD3	INT1 (External interrupt 1 Input)
PD5	OC1A (Timer/Counter1 Output CompareA Match Output)
PD6	WR (Write Strobe to External Memory)
PD7	RD (Read Strobe to External Memory)

Table 23. Port D Pin Alternate Functions

When the pins are used for the alternate function, the DDRD and PORTD registers have to be set according to the alternate function description.

the self-timed write operation in the serial programming mode. During programming, the supply voltage must be in accordance with Table 26.

Table 26. Supply Voltage during Programming

Part	Serial Programming	Parallel Programming	
AT90S8515	2.7 - 6.0V	4.5 - 5.5V	

Parallel Programming This section describes how to parallel program and verify Flash program memory, EEPROM data memory, Lock bits and Fuse bits in the AT90S8515.

Signal Names In this section, some pins of the AT908515 are referenced by signal names describing their function during parallel programming. See Figure 60 and Table 27. Pins not described in Table 27 are referenced by pin names.

The XA1/XA0 pins determine the action executed when the XTAL1 pin is given a positive pulse. The bit coding are shown in Table 28.

When pulsing \overline{WR} or \overline{OE} , the command loaded determines the action executed. The command is a byte where the different bits are assigned functions as shown in Table 29.

Figure 60. Parallel Programming



	Bit 5 = SPIEN Fuse bit
	Bit 0 = FSTRT Fuse bit
	Bit 7 - 6, 4 - 1 = "1". These bits are reserved and should be left unprogrammed ("1").
	3. Give \overline{WR} a t _{WLWH_PFB} -wide negative pulse to execute the programming, t _{WLWH_PFB} is found in Table 30. Programming the Fuse bits does not generate any activity on the RDY/BSY pin.
Programming the Lock Bits	The algorithm for programming the Lock bits is as follows (refer to "Programming the Flash" on page 81 for details on command and data loading):
	1. A: Load Command "0010 0000".
	2. D: Load Data Low Byte. Bit n = "0" programs the Lock bit.
	Bit 2 = Lock Bit2
	Bit 1 = Lock Bit1
	Bit 7 - 3, $0 = $ "1". These bits are reserved and should be left unprogrammed ("1").
	3. E: Write Data Low Byte.
	The Lock bits can only be cleared by executing Chip Erase.
Reading the Fuse and Lock Bits	The algorithm for reading the Fuse and Lock bits is as follows (refer to "Programming the Flash" on page 81 for details on Command loading):
	1. A: Load Command "0000 0100".
	 Set OE to "0", and BS to "1". The status of the Fuse and Lock bits can now be read at DATA ("0" means programmed).
	Bit 7 = Lock Bit1
	Bit 6 = Lock Bit2
	Bit 5 = SPIEN Fuse bit
	Bit 0 = FSTRT Fuse bit
	3. Set OE to "1".
	Observe that BS needs to be set to "1".
Reading the Signature Bytes	The algorithm for reading the signature bytes is as follows (refer to "Programming the Flash" on page 81 for details on command and address loading):
	1. A: Load Command "0000 1000".
	2. C: Load Address Low Byte (\$00 - \$02).
	Set $\overline{\text{OE}}$ to "0", and BS to "0". The selected signature byte can now be read at DATA.
	3. Set OE to "1".

AMEL

Serial Programming Characteristics





Table 33. Serial Programming Characteristics, $T_A = -40^{\circ}C$ to $85^{\circ}C$, $V_{CC} = 2.7V - 6.0V$ (unless otherwise noted)

Symbol	Parameter	Min	Тур	Max	Units
1/t _{CLCL}	Oscillator Frequency ($V_{CC} = 2.7 - 4.0V$)	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 34. 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 35. Minimum Wait Delay after Writing a Flash 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





