Features

- RF Frequency Range of 250 - 450 MHz
- 6 dBm RF Output into Matched Antenna
- RF Output Power Adjustable in 1 dB Steps
- Phase-Locked Loop (PLL) Based Frequency Synthesizer
- Data Bandwidth of up to 20 Kb/s
- Operates Down to 2V from a CR2032/CR2016 $LiMnO_2$ Battery
- 8-bit AVR® RISC Microcontroller Core
- Minimal External Components
- Space-saving 20-pin TSSOP
- 2 KB (1K x 16b) of Flash Program Memory
- 128 Bytes of EEPROM
- 128 Bytes of SRAM
- In-System Programmable Data and Program Memory
- Six I/O (Serial I/F, LED Drive Outputs, Button Input Interrupts)
- Low Battery Detect and Brown-out Protection

Applications

- Remote Keyless Entry (RKE) Transmitters
- Wireless Security Systems
- Home Appliance Control (Gas Fireplace, Ceiling Fans)
- Radio Remote Control (Hobby, Toys)
- Garage Door Openers
- Wireless PC Peripherals (Keyboard, Mouse)
- Telemetry (Tire Pressure, Utility Meter, Asset Tracking)

SmartRF™ Wireless Data Transmitter

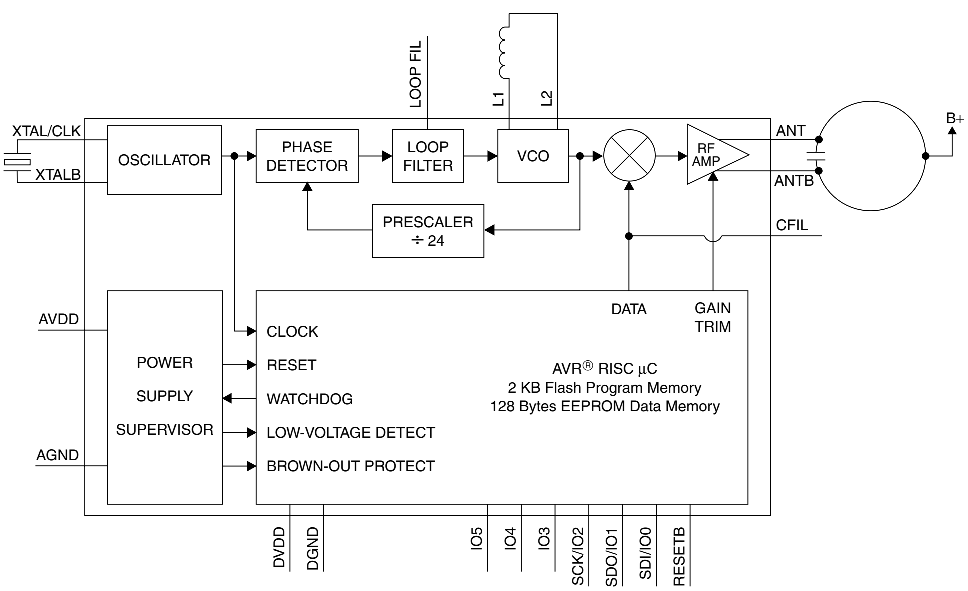
AT86RF401

Preliminary

Description

The Atmel AT86RF401 SmartRF™ MicroTransmitter is a highly integrated, low-cost RF transmitter, combined with an AVR RISC microcontroller. It requires only a crystal, a single $LiMnO_2$ coin cell (CR2032 or similar), an inductor and a tuned-loop antenna to implement a complete on-off keyed (OOK) wireless RF data transmitter.

Block Diagram



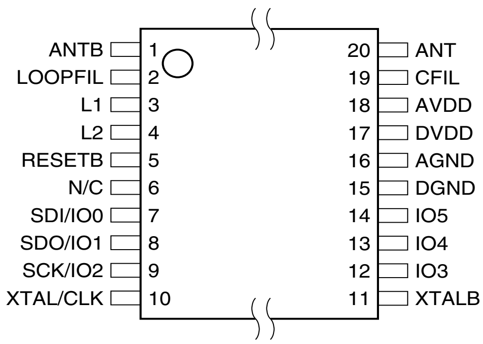
Rev. 1424B–03/01

In-system, programmable, nonvolatile Flash program memory and EEPROM data storage makes possible rapid time to market and lower inventory costs.

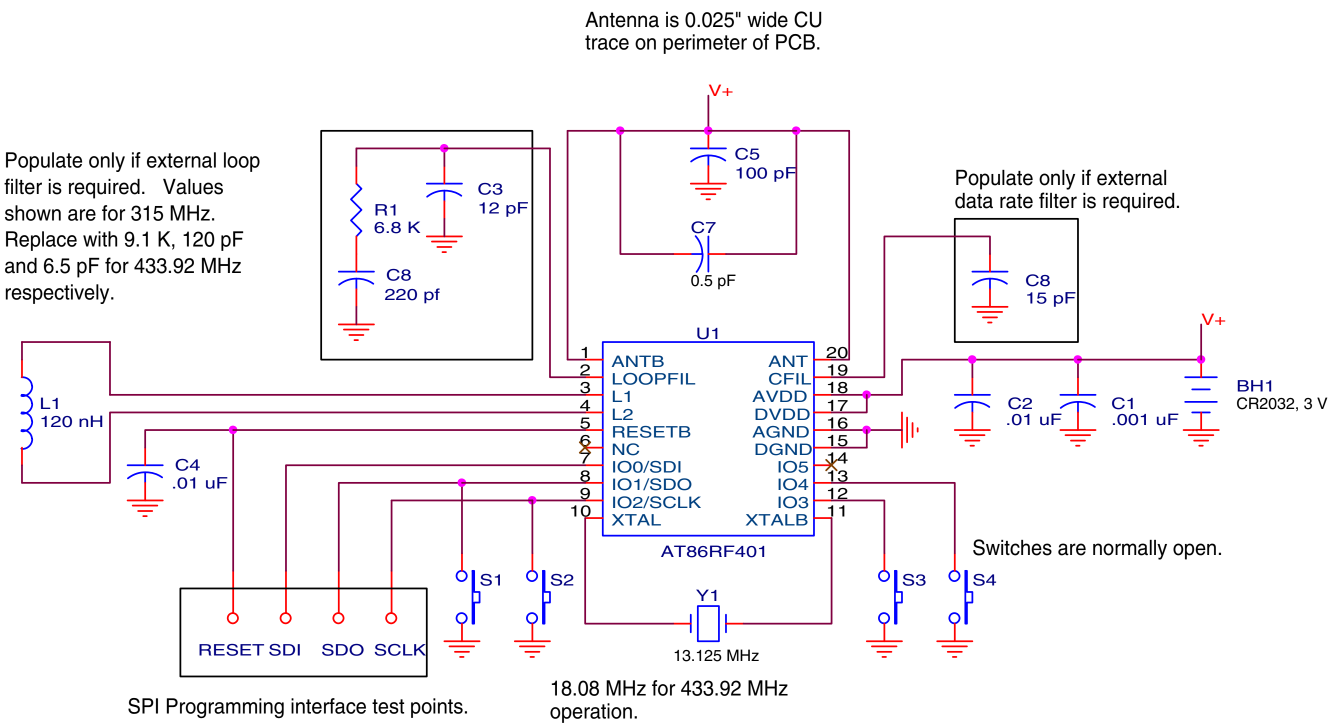
Static current consumption is kept to a minimum with an ultra-low current shutdown mode. Normal operation resumes when a button is pressed. This activates the crystal oscillator circuit which serves as the clock for the AVR microcontroller. The RF carrier is synthesized utilizing an on-board Voltage Controlled Oscillator (VCO). Its accuracy is maintained with a PLL detector which compares the crystal oscillator to a frequency-scaled version (divided by 24) of the RF carrier. The resulting error signal adjusts the VCO to produce a very stable RF carrier. An interrupt based bit-timer structure, integral to the AVR microcontroller, simplifies the implementation of user specific, data-bit encoding routines, such as PWM or Manchester, for modulating the RF carrier. The RF signal output is placed differentially on a tuned-loop antenna, which may be realized as a counterspread copper trace on a PCB.

The AT86RF401 is fabricated in Atmel's 0.6 µm Mixed Signal CMOS + EEPROM process, enabling true system-level integration (SLI).

20-Lead TSSOP



Sample Circuit



Pin Descriptions: 20-lead TSSOP

Symbol	Pin	Description
ANTB	1	Differential Antenna Output.
LOOPFIL	2	External VCO Loop-filter Connection.
L1	3	External VCO Inductor Connection.
L2	4	External VCO Inductor Connection.
RESETB	5	SPI Reset Input. Reset input. A "low" on this pin resets the device and puts the part into SPI mode. A logic-high on this pin causes the device to execute its program if the V_{DD} is above the brown-out voltage level. This pin has a 50 kΩ pull-up to DVDD.
NC	6	No Connect. Float Pin.
SDI/IO0	7	SPI Data In/Input/Output 0. General-purpose I/O and button input. In SPI mode, this pin serves as SDI (Serial data input). This pin has a 50 kΩ pull-up to DVDD.
SDO/IO1	8	SPI Data Out/Input/Output1. General-purpose I/O and button input. In SPI mode, this pin serves as SDO (Serial Data Output). This pin has a 50 kΩ pull-up to DVDD.
SCK/IO2	9	SPI Clock/Input/Output 2. General-purpose I/O and button input. In SPI mode, this pin serves as SCK (SPI Clock Input). This pin has a 50 kΩ pull-up to DVDD.
XTAL/CLK	10	Crystal /Clock Input. Input to the inverting oscillator amplifier and input to the internal clock operating circuit. This pin may be driven externally for test purposes.
XTALB	11	Crystal Output. Output from the inverting oscillator amplifier.
IO3	12	Input/Output 3. General-purpose I/O and button input. This pin has a 50 kΩ pull-up resistor to DVDD.
IO4	13	Input/Output 4. General-purpose I/O and button input. This pin has a 50 kΩ pull-up resistor to DVDD.
IO5	14	Input/Output 5. General-purpose I/O and button input. This pin has a 50 kΩ pull-up resistor to DVDD.
DGND	15	Digital Ground.
AGND	16	Analog Ground.
DVDD	17	Digital Voltage Supply.
AVDD	18	Analog Voltage Supply.
CFIL	19	External Data Rate Filter.
ANT	20	Differential Antenna Output.

Absolute Maximum Ratings

Antenna Voltage (Pins 1, 20)	–1V to 10V
Operating Temperature	–40 to +85°C
Storage Temperature (without bias)	–55°C to +125°C
Voltage on V_{DD} with respect to ground	6.0V
Voltage on Pins 2 - 19 (TSSOP 20)	–0.1 to V_{DD} +0.3V

*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or 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.

DC Characteristics

V_{DD} = 2.5V; Typical values at T_A = 25°C unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Supply						
V_{DD}	Supply Voltage		2.0	–	3.5	V
I_{DD}	Stand-by Current (off)		–	0.1	0.5	µA
	AVR Active	f_{XTAL} = 13.125 MHz, f_{AVR} = 1 MHz	–	–	–	mA
	Frequency Synthesizer + AVR Active	f_{XTAL} = 13.125 MHz, f_{AVR} = 1 MHz	–	13	–	mA
	Transmit (FS, AVR and Power Amp active)	f_0 = 433.92 MHz, 50% DC, f_{avr} = 1 MHz	12.5	17	22	mA
Digital Inputs (SDI, SCK,RESTB,IOx)						
V_{IH}	High-level Input Voltage		0.8* V_{DD}	–	V_{DD}	V
V_{IL}	Low-level Input Voltage		0	–	0.2* V_{DD}	V
I_{IH}	High-level Input Current	V_{IH} = V_{DD} = 3.5V		–	1	µA
I_{IL}	Low-level Input Current	V_{IL} = 0V, V_{DD} = 3.5V	–100	–	0	µA
Digital Outputs (SDO,IOx)						
V_{OH}	High-level Output Voltage	I_{OH} = 500 µA	V_{DD}–0.4	–	–	V
V_{OL}	Low-level Output Voltage	I_{OL}= –2 mA	–	–	0.4	V
V_{th}	Brown-out Threshold	T_A = 25°C T_A = –40 to +85°C	1.98 1.93	2.03 2.03	2.08 2.13	V V

Analog/RF Specs

V_{DD} = 2.5V; f_0 = 433.92 MHz, P_0 = maximum output, Typical values at T_A = 25°C unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
RF Amplifier						
t_{ON}	Turn-on Time	Idle to Ready	–	500	–	µs
P_O	Output Power Range[1]	Transmitting (RF "ON")	–	–	6	dBm
P_{CSS}	Power Control Step Size		–	1	–	dB
I_O(min)	Output Current[2]	Transmitting (RF "ON")	–	–	10	mA
Crystal Oscillator						
t_{ON}	Turn-on Time		–	50	100	µs
f_{OSC}	Oscillation Frequency Range		6	–	20	MHz
Frequency Synthesizer /PLL						
F_{OUT}	Output Frequency Range		250	–	450	MHz
DIV	PLL Divider Ratio		–	24	–	–
PSRR	Power Supply Rejection		–	0.05	–	%/V
	Spurious Emission	Outside Transmit Band	–	–	–60	dBm
BW_{AM}	OOK Data Bandwidth Manchester Data Rate	Software Using Polling Method	– –	–	20 20	kHz kbps

Notes: 1. Differential outputs (pins 1 and 20) matched and transformed into 50Ω.
2. Measured differentially at the antenna between pins 1 and 20.

Functional Description

The complete circuit consists of the following functional blocks:

Transmitter

Crystal Oscillator

The crystal oscillator circuit is designed to work with crystals with fundamental frequencies between 6 and 20 MHz. 40 pF of internal capacitance is connected between each of the crystal input pins and (chip) ground. Alternatively, an external clock can be used for these functions.

This circuit provides the master clock for the entire chip. A programmable divider is used to provide the AVR system clock.

Radio Frequency Power Amplifier

The RF power amplifier generates a differential output suitable for driving an off-chip tuned-loop antenna from the PLL output. The PLL output signal is gated using on-off keyed (OOK) modulation before transmission. It is used as the RF carrier frequency for the transmitted data stream. The amplifier can be configured via software to reduce the power output by 5 dB (in 1 dB steps).

Frequency Synthesizer

The frequency synthesizer utilizes a PLL, which consists of a phase detector, a ÷24 prescaler, an on-chip loop filter, and an integrated Voltage Controlled Oscillator (VCO). The VCO output is buffered prior to the output amplifier. The output frequency is 24 times the crystal frequency.

Bandgap Reference

The device uses a 1.2V (nominal) bandgap reference generator to provide consistent performance over a wide range of input supply voltages. This reference voltage is used throughout the device.

Brown-out Protection/Low Battery Detection

The brown-out protection and low battery detection functions consist of a voltage reference, a sampling block, and an autozero comparator. The circuit's primary operating mode is brown-out protection.

Brown-out Protection

The brown-out protection circuit detects when the level of V_{DD} drops below the minimum voltage that guarantees proper operation. The brown-out voltage for this device is typically 2.0 volts.

If a brown-out occurs, the device enters a reset state. It stays in this state until either of the following occurs:

1. The level of V_{DD} increases ~0.1 - 0.2 volts above the brown-out voltage. This causes the device to enter a warm reboot state.
2. The level of V_{DD} drops to ~0 volts, then increases above the POR level. This places the device into the "cold start" mode of operation, identical to battery insertion.

Low Battery Detection

The low battery detection feature allows the programmer to select a value for V_{DD} at which a warning is issued to the user. This warning may be utilized to activate an I/O port, for example.

If low battery detection occurs, bit 7 of register BL_CONFIG is set. Bit 6 of register BL_CONFIG is used to indicate that bit 7 is valid. It is left to the programmer to poll both bits to ensure the potential warning is valid.

Bits 5 - 0 of register BL_CONFIG are used to program the low battery detect level. This warning level is programmable between ~1.5 - 2.7 volts.

Note: The warning level can be set below the brown-out voltage level.

Bit 6 of register IO_ENAB is used to control the amount of hysteresis present in the low battery detect level. If bit 6 is 0, ~50 - 100 mV of hysteresis is used. If bit 6 is 1, then ~100 - 200 mV of hysteresis is applied. The amount of hysteresis varies with the low battery detect level, becoming less as the detect level is lowered.

The formulas for calculating low battery detection thresholds and hysteresis are located in Table 1.

Table 1. Low Battery Detection Threshold Formulas (V_{REF} is approximately 0.7 volts)

V_{DD} Falling	V_{DD} Rising	
	bo_hyst = 1 (low hysteresis)	**bo_hyst = 0 (large hysteresis)**
$VDD= \frac{3.887 \times V_{REF}}{1 + \left[\frac{0.887}{63} \times \text{BL[5:0]}\right]}$	$VDD= \frac{4.05 \times V_{REF}}{1 + \left[\frac{0.887}{63} \times \text{BL[5:0]}\right]}$	$VDD= \frac{4.22 \times V_{REF}}{1 + \left[\frac{0.887}{63} \times \text{BL[5:0]}\right]}$
$\text{BL[5:0]} = 71 \times \left[3.887 \times \left[\frac{V_{REF}}{V_{DD}}\right] - 1\right]$	$\text{BL[5:0]} = 71 \times \left[4.05 \times \left[\frac{V_{REF}}{V_{DD}}\right] - 1\right]$	$\text{BL[5:0]} = 71 \times \left[4.22 \times \left[\frac{V_{REF}}{V_{DD}}\right] - 1\right]$

Bit Timer

A hardware assist has been included in the AT86RF401 to make transmission of data easier. Keying of the transmitter is timed by this logic and interrupts are generated when data is needed by the timer, or when transmission is complete. The timer also supports code that uses polling instead of interrupts. Using polling instead of interrupts may facilitate higher bit rates. Additionally, this timer may be used to time pulses arriving at the IO3 pin. This enables the AT86RF401 to be used to decode the signal detected by an external receiver chip.

Transmit Mode Bit Coding and Timing

Bit coding is done by the AVR before data is sent to the bit timer. Bit timing is controlled by the count value in the Bit Timer Count (BTCNT) register and the 2 most significant bits in the Bit Timer Control Register (BTCR). Generally the time of each bit is:

$$P_{xx} = P \times (countval + 1)$$

Where P_{xx} is the period of each time slot, *countval* is the counter value in the BTCNT and BTCR registers. P is the AVR clock period which is set in the PWR_CTL register. *countval* = {BTCR[7:6], BTCNT[7:0]}.

Interrupts

There are two interrupts associated with transmit mode:

1. Transmit Buffer Empty Interrupt. This vectors to address 0x04. Flag 0 is set, and, if enabled, this interrupt is generated when the timer removes the value from the DATA bit in the BTCR. This interrupt service routine should load the next bit into the DATA bit in the BTCR.
2. TXDONE Interrupt. This vectors to address 0x02. Flag 2 is set, and, if enabled, an interrupt is generated when the counter has counted down to zero and the buffer is empty. This indicates that transmission is complete. This interrupt service routine should turn off the transmitter and turn off the bit timer using the mode bits.

Bit Timer in Receive Mode

When put into receive mode, the bit timer times pulses arriving at the IO3 pin. When enabled, the counter counts up from zero and places that value in the BTCNT register when an edge occurs. If the edge is rising, the DATA bit in the BTCR is set. If the edge is falling, the key bit in the BTCR is reset. This mode may be used to decode signals from a receiver chip easily.

Bit Timer Operation as a Generic Timer/Counter

The Bit Timer may be used as a generic timer by not allowing it to key of the transmitter. An interrupt is generated after the amount of time dictated by the count value.

Watchdog Timer

When enabling the watchdog timer, the status of the watchdog time is unknown. The user is advised to execute a WDR instruction before enabling the watchdog. Otherwise, the device might get reset before the first WDR after enabling is reached. To prevent the unintentional disabling of the watchdog, a special turn-off procedure must be followed when the watchdog is disabled. Refer to the description of the Watchdog Timer Control Register for details (See Register $22 in I/O Memory). The watchdog timer prescaler determines the number of system clocks that occur before the watchdog reset is asserted. The system clock is determined by Bits[7:5] of the PWR_CTL register.

Reset and Interrupt Handling

The AT86RF401 Reset and Interrupt vectors are defined in Table 2. The I-bit in the status register must be set to enable the interrupts.

Table 2. Reset and Interrupt Vectors

Vector Number	Program Address	Source	Interrupt Definition
1	$000	RESETB, Watchdog, Buttons	Hardware Pin or Watchdog or Button Reset
2	$002	Transmission Done (TXDONE)	Bit Timer Flag 2 Interrupt
4	$004	Transmit Buffer Empty	Bit Timer Flag 0 Interrupt

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      BT_F2_ISR      ; Bit timer flag 2 interrupt service routine
$004               jmp      BT_F0_ISR      ; Bit timer flag 0 interrupt service routine
$006  MAIN:        <instr>  xxx            ; Main program start
 …          …        …       …
```

Reset Sources

The AT86RF401 has several sources of reset:

- Power-on Reset. The AT86RF401 is reset when the supply voltage is applied between the VDD and GND pins. There are 10^6 cycles of delay between Power-on Reset occurring and the part becoming active. This is to ensure that the power is stable.
- External Reset. The AT86RF401 is reset when a logic low level is present on the RESETB pin. This resets all I/O Registers and puts the part into SPI mode. The I/O Registers may be read and written by the SPI interface after two AVR System Clocks.
- Watchdog Reset. This is similar to power-on reset, but is caused by the watchdog timer and doesn't have a 10^6 cycle delay prior to becoming active.
- Brown-out Reset. This is caused by the battery voltage dropping below the Brown-out Threshold voltage trip point.
- Button Reset (software reset). The part is placed into a special reset state by software. The part is released from reset when a properly configured button is activated, and the part is not in external reset or brown-out reset. In the button reset state, most I/O registers are not reset.

During power-on reset and watchdog reset, all I/O registers are set to their initial values, and the program starts execution from address $000.

Note: The instruction placed in address $000 must be an RJMP – relative jump – instruction or a JMP (absolute jump) to the reset handling routine. If an RJMP or JMP instruction is not present at address $000, the part is placed into a "no program" reset state. This is to protect the part from fetching instructions when no program is present.

Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is 4 clock cycles minimum. After the 4 clock cycles, the program vector address for the actual interrupt handling routine is executed. During this 4 clock cycle period, the Program Counter is pushed onto the Stack. The vector is a jump to the interrupt routine, and this jump takes 2 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 takes 4 clock cycles. During these 4 clock cycles, the Program Counter is popped back from the Stack. When AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

Note: The Status Register – SREG – is not saved by the AV*R* hardware. This must be performed by user software when required.

Memory Programming

Program Memory Lock Bits

The AT86RF401 MCU provides two lock bits which can be left unprogrammed ("1") or can be programmed ("0") to obtain the additional features listed in Table 3.

Table 3. Lock Bit Protection Modes

Program Lock Bits			
Mode	**LB1**	**LB2**	**Protection Type**
1	1	1	No program lock features.
2	0	1	Further programming of the EEPROM is disabled (Both program and data memory).
3	0	0	Same as mode 2, but Verify is also disabled.

Note: The Lock Bits can only be erased with the Chip Erase operation.

In-system Flash and EEPROM

The AT86RF401 offers 2 Kbytes (1K x 16) of in-system reprogrammable Flash program memory and 128 bytes of EEPROM data memory. This memory can be programmed serially via the SPI interface.

SPI Interface

Both the Program and Data memory arrays can be programmed using the serial SPI bus while RESETB is pulled to GND. The serial interface consists of pins SCK, SDI (input) and SDO (output).

When programming, an auto-erase cycle is built into the self-timed programming operation, and there is no need to first execute the Chip Erase instruction. The Chip Erase operation sets every memory location in EEPROM array to $FF.

Either an external system clock is supplied at pin XTALB or a crystal needs to be connected across pins XTAL and XTALB. The minimum low and high periods for the serial clock (SCK) input are defined as follows:

Low: 4 XTAL Clock Cycles *High:* 16 XTAL Clock Cycles

Serial Programming Algorithm

Refer to Table 4. and Figures 1 and 2. To program and verify the AT86RF401 in the serial programming mode, the following sequence is recommended.

Power-up Sequence:

1. Apply power between VDD and GND while RESETB and SCK are set to “0”. If a crystal is not connected across pins XTAL and XTALB, apply a clock signal to the XTAL pin. If the programmer can not guarantee that SCK is held low during power-up, RESETB must be given a positive pulse after SCK has been set to “0”.
2. Wait for at least 20 ms and enable serial programming by sending the Programming Enable instruction to pin SDI. This must occur prior to any program/erase operations.
3. If a chip erase is performed, wait 4 ms, give RESETB a positive pulse and start over again from Step 2.
4. The array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. The memory location is first automatically erased before new data is written. The next byte can be written after 4 ms.
5. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output SDO.
6. At the end of the programming session, RESETB must be set high to commence normal operation.

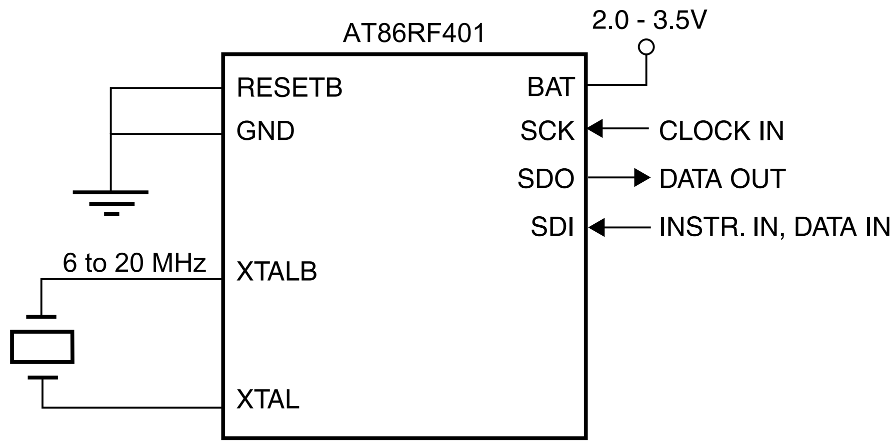
Data EEPROM Access from the AVR

Table 4. AT86RF401 Serial Programming Instruction Set

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte 4	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx	Enable Serial Programming after RESETB goes low.
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip erase EEPROM
Read Program Memory	0010 **H**000	0000 00**aa**	**bbbb bbbb**	**oooo oooo**	Read **H** (high or low) data **o** from Program memory at word address **a:b**
Write Program Memory	0100 **H**000	0000 00**aa**	**bbbb bbbb**	**iiii iiii**	Write **H** (high or low) data **i** to Program memory at word address **a:b**
Read EEPROM Memory	1010 0000	0000 0000	x**bbb bbbb**	**oooo oooo**	Read data **o** from EEPROM memory at address **b**
Write EEPROM Memory	1100 0000	0000 0000	x**bbb bbbb**	**iiii iiii**	Write data **i** to EEPROM memory at address **b**
Write Lock Bits	1010 1100	111x x***21***x	xxxx xxxx	xxxx xxxx	Write lock bits. Set bits **21** = "0" to program lock bits.
I/O Read	10110000	0000 0000	**00bbbbbb**	**oooo oooo**	Read data 0 from I/O memory address b
I/O Write	11010000	0000 0000	00bbbbbb	**iiii iiii**	Write data i to I/O memory address b

Note: **a** = address high bits
b = address low bits
H = 0 - Low byte, 1- High byte
o = data out
i = data in
x = don't care
1= lock bit 1
2= lock bit 2

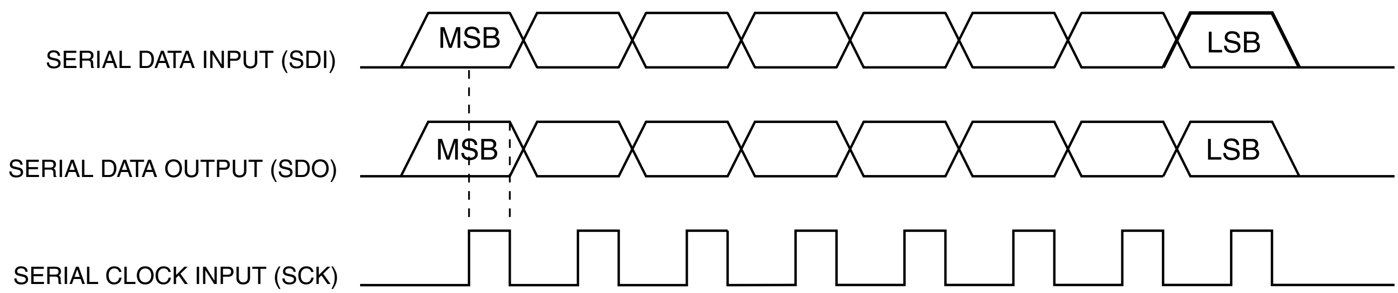
Figure 1. Serial Programming and Verify



Notes: 1. When *writing* serial data to the AT86RF401, data is clocked on the *rising* edge of CLK.

2. When *reading* data from the AT86RF401, data is clocked on the *falling* edge of CLK. See Figure 2. for an explanation.

Figure 2. Serial Programming Waveforms



Note: The AT86RF401 includes an integrated 128-byte EEPROM, which is accessed by 3 registers located in the I/O memory space. These are the DEECR, DEEDR, and DEEAR registers. For more information, refer to I/O Register Description.

AVR Core

Architectural Overview

The fast-access register file concept 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.

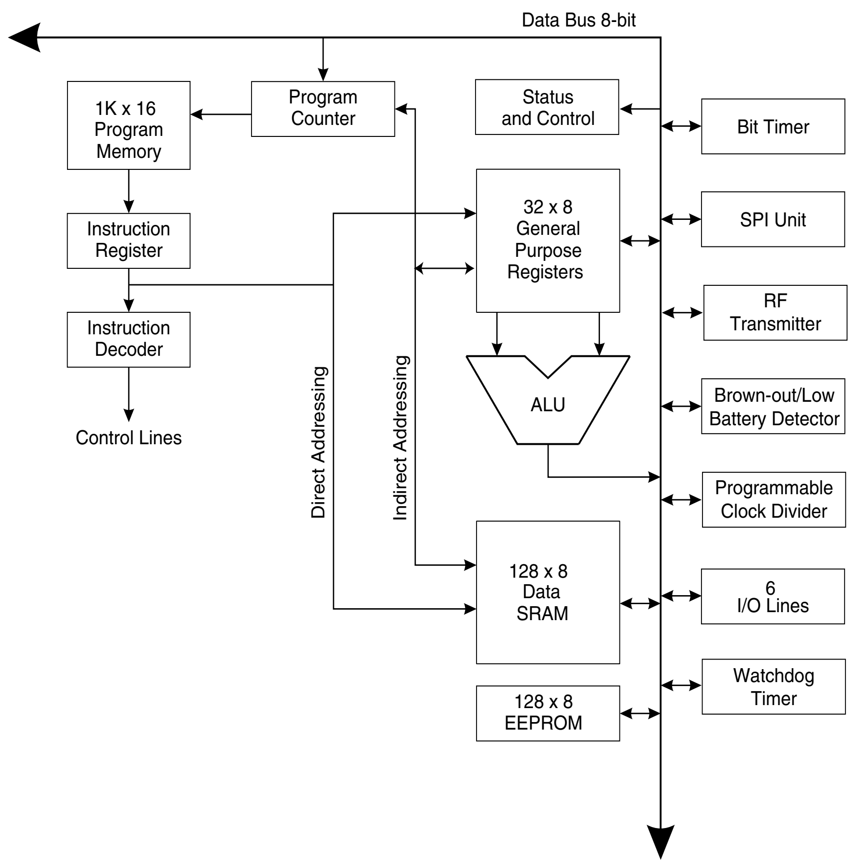
Six of the 32 registers can be used as three 16-bits indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of the three address pointers is also used as the address pointer for look-up tables in Flash program memory. These added function registers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

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 3 shows the AT86RF401 AVR 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 - $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.

Figure 3. The AT86RF401 AVR Architecture



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, reprogrammable Flash memory.

With the jump and call instructions, the whole 1K word 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.

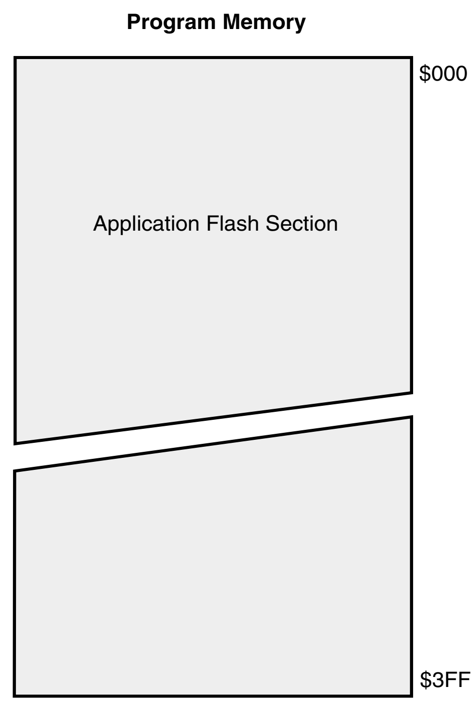
During interrupts and subroutine calls, the return address program counter (PC) is stored on the stack. The stack is effectively allocated in the general data SRAM, and consequently the stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the reset routine (before subroutines or interrupts are executed). The 7-bit stack pointer SP is read/write accessible in the I/O space.

The 128 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.

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.

Figure 4. Memory Maps



General-purpose Register File

Figure 5 shows the structure of the 32 general-purpose working registers in the CPU.

Figure 5. AVR CPU General-purpose Working Registers

	7 0	Addr.	
	R0	$00	
	R1	$01	
	R2	$02	
	…		
	R13	$0D	
General	R14	$0E	
Purpose	R15	$0F	
Working	R16	$10	
Registers	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

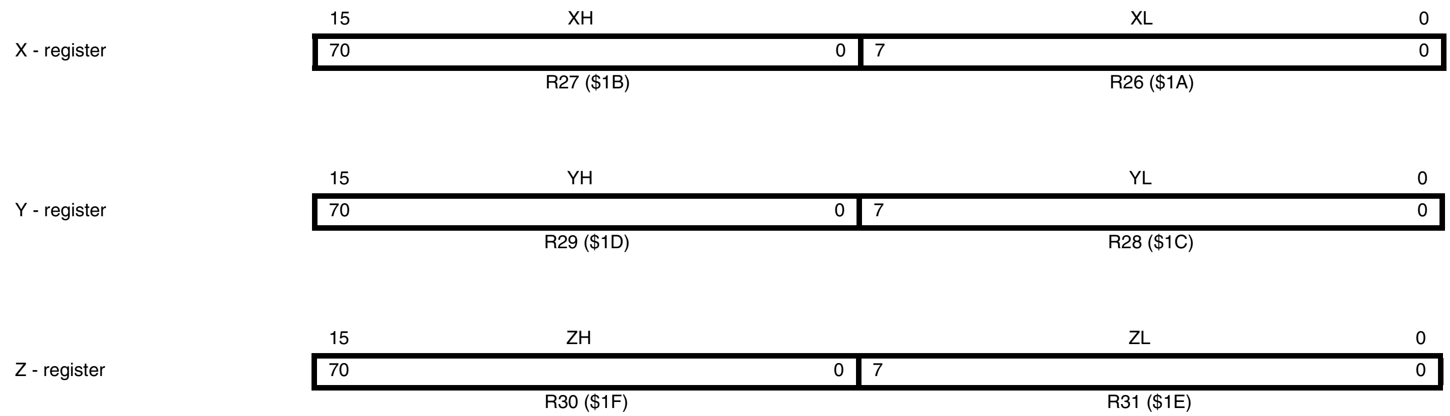
All the register operating instructions in the instruction set have direct and single cycle access to all registers. The only exception is the five constant arithmetic and logic instructions SBCI, SUBI, CPI, ANDI and ORI between a constant and a register, and the LDI instruction for load immediate constant 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 Figure 6, 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.

The 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 as:

Figure 6. The X, Y, and Z Registers



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 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. The multiplier is not present in this version of the core. Therefore, the MUL instruction is not supported.

In-system Self-programmable Flash Program Memory

The AT86RF401 contains 2 Kbytes of on-chip Flash memory for program storage. Since all instructions are 16- or 32-bit words, the Flash is organized as 1K x 16.

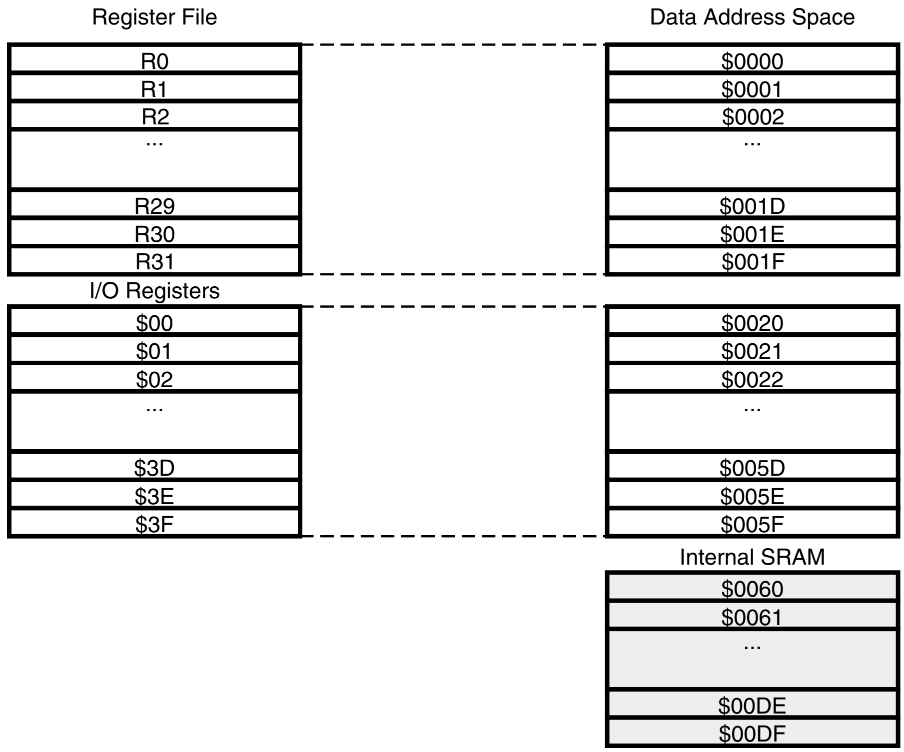
The Flash memory has an endurance of at least 1000 write/erase cycles. The Program Counter (PC) is 10 bits wide, thus addressing the 1024 program memory locations. See the Memory Programming Section for a detailed description on Flash data serial downloading.

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

SRAM Data Memory

Figure 7 shows how the AT86RF401 SRAM Memory is organized.

Figure 7. SRAM Organization



The lower 224 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 128 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.

The direct addressing reaches the entire data space.

The Indirect with Displacement mode features a 63 address locations 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.

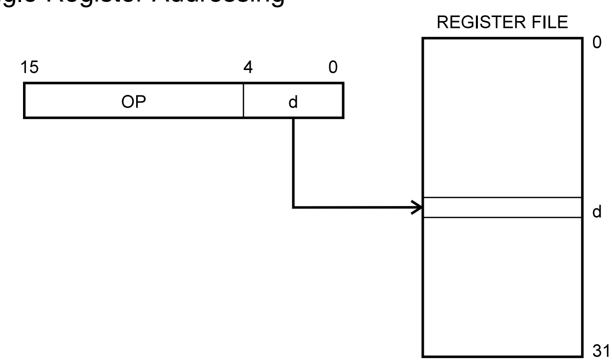
The 32 general-purpose working registers, 64 I/O registers and the 128 bytes of internal data SRAM in the AT86RF401 are all accessible through all these addressing modes.

Program and Data Addressing Modes

The AT86RF401 AVR Enhanced 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

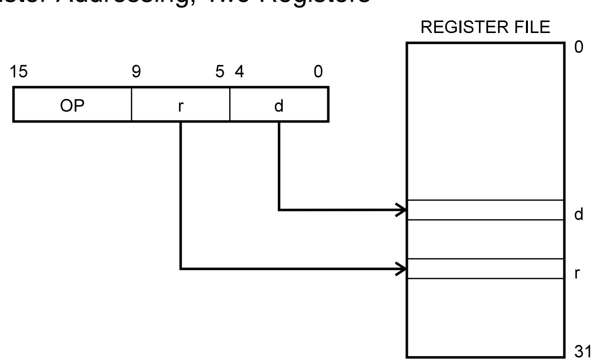
Figure 8. Direct Single Register Addressing



The operand is contained in register d (Rd).

Register Direct, Two Registers Rd And Rr

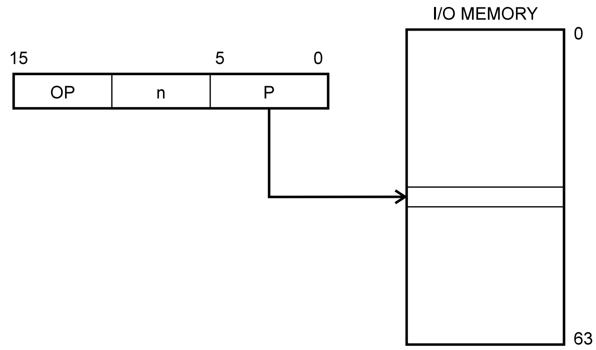
Figure 9. Direct Register Addressing, Two Registers



Operands are contained in register r (Rr) and d (Rd). The result is stored in register d (Rd).

I/O Direct

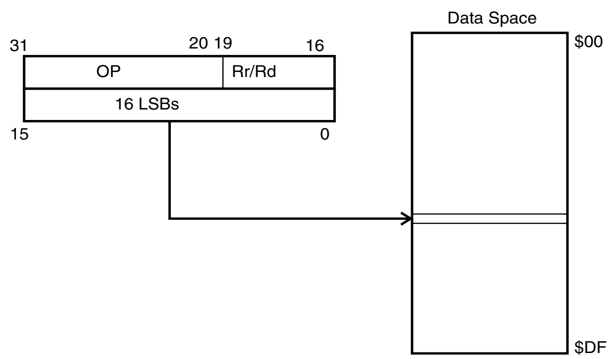
Figure 10. I/O Direct Addressing



Operand address is contained in 6 bits of the instruction word. "n" is the destination or source register address.

Data Direct

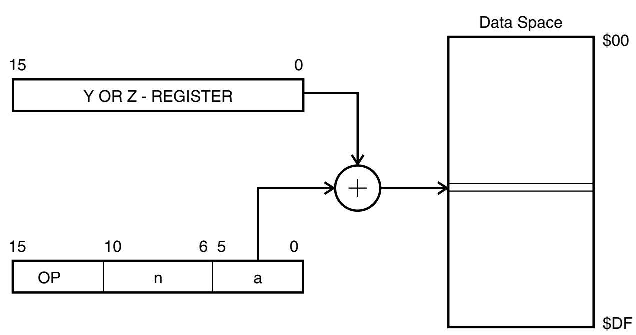
Figure 11. Direct Data Addressing



A 16-bit Data Address is contained in the 16 LSBs of a two word instruction. Rd/Rr specify the destination or source register.

Data Indirect with Displacement

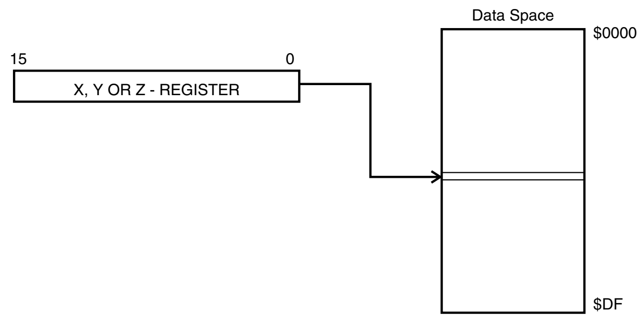
Figure 12. Data Indirect with Displacement



Operand address is the result of the Y or Z-register contents added to the address contained in 6 bits of the instruction word.

Data Indirect

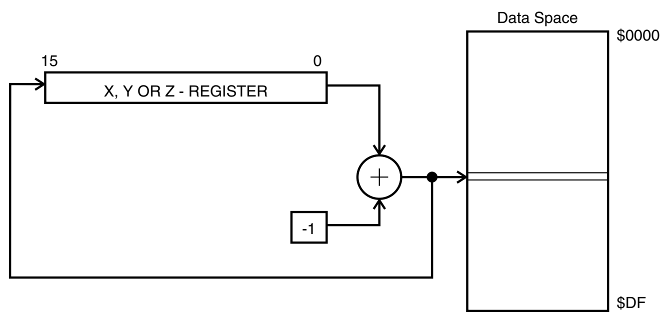
Figure 13. Data Indirect Addressing



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

Data Indirect with Pre-decrement

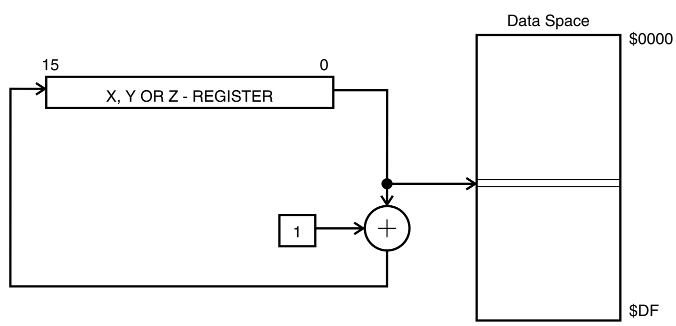
Figure 14. Data Indirect Addressing with Pre-decrement



The X, Y or the Z-register is decremented before the operation. Operand address is the decremented contents of the X, Y or the Z-register.

Data Indirect with Post-increment

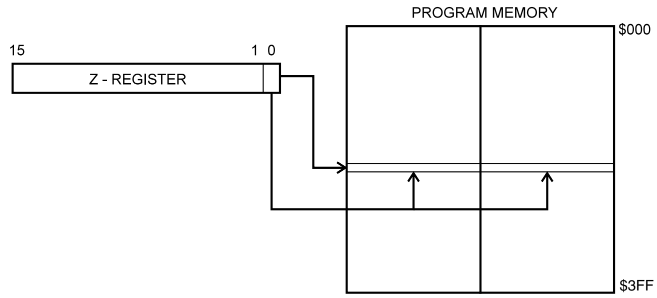
Figure 15. Data Indirect Addressing with Post-increment



The X, Y or the Z-register is incremented after the operation. Operand address is the content of the X, Y or the Z-register prior to incrementing.

Constant Addressing Using The LPM Instruction

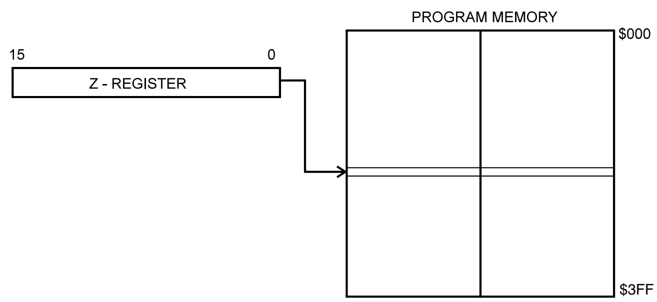
Figure 16. Code Memory Constant Addressing



Constant byte address is specified by the Z-register contents. The 10 MSBs select word address (0 - 1K). For LPM, the LSB selects low byte if cleared (LSB = 0) or high byte if set (LSB = 1).

Indirect Program Addressing, IJMP and ICALL

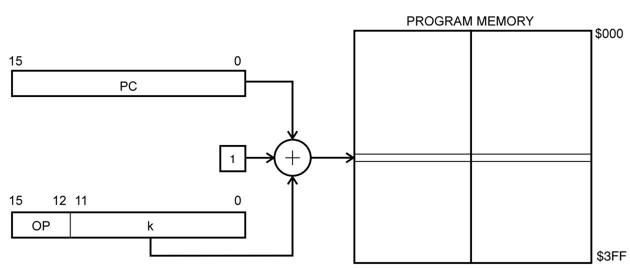
Figure 17. Indirect Program Memory Addressing



Program execution continues at address contained by the Z-register (i.e., the PC is loaded with the contents of the Z-register).

Relative Program Addressing, RJMP and RCALL

Figure 18. Relative Program Memory Addressing



Program execution continues at address PC + k + 1. The relative address k is from -2048 to 2047.

EEPROM Data Memory

The AT86RF401 contains 128 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described in the Memory Programming Section.

Memory Access Times and Instruction Execution Timing

This section describes the general access timing concepts for instruction execution and internal memory access.

The AVR CPU is driven by the System Clock Ø, generated from the main oscillator for the chip. A programmable clock divider generates this clock from the crystal oscillator input.

Figure 19 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access register file concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks and functions per power-unit.

Figure 19. The Parallel Instruction Fetches and Instruction Executions

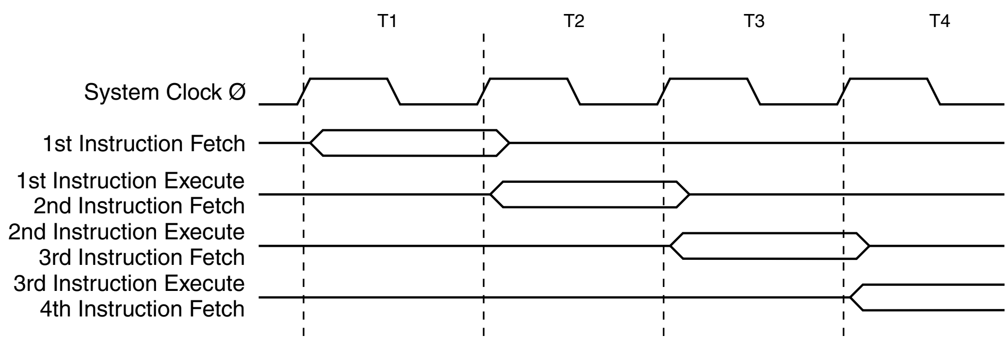
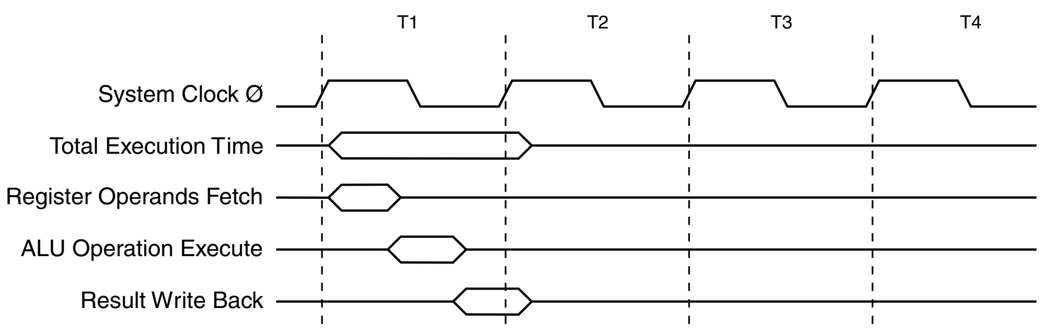


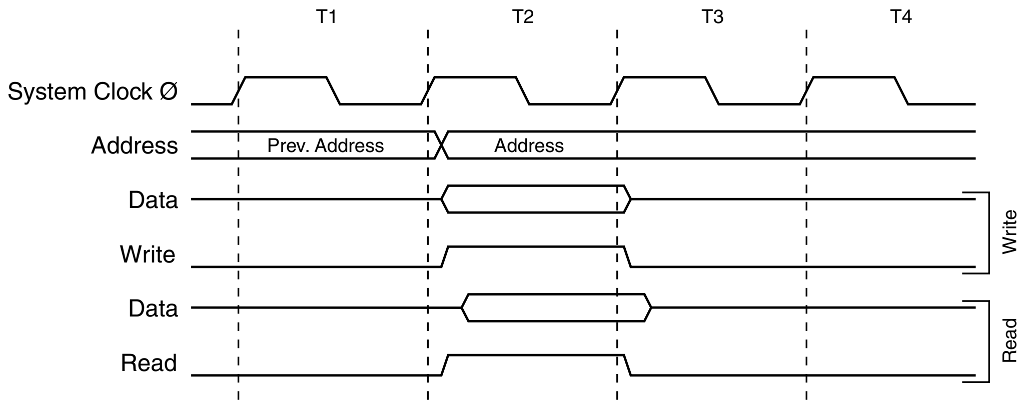
Figure 20 shows the internal timing concept for the register file. In a single clock cycle, an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 20. Single Cycle ALU Operation



The internal data SRAM access is performed in two System Clock cycles as described in Figure 21.

Figure 21. On-chip Data SRAM Access Cycles



All AT86RF401 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 chapter 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 one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O register, writing a one 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.

I/O Memory

The I/O space definition of the AT86RF401 is shown in the following table:

Table 5. AT86RF401 I/O Space Definitions

Address Hex	Name	Function
$3F	SREG	Status Register
$3E	SPH	Stack Pointer High Register (program to 0 x 00)
$3D	SPL	Stack Pointer Low Register
$3B	GIMSK	Global Interrupt Mask
$3A	GIFR	Global Interrupt Flag Register
$35	BL_CONFIG	Battery Low Configuration Register
$34	B_DET	Button Detect Register
$33	PWR_CTL	Power Control Register
$32	IO_DATIN	I/O DATA IN Register
$31	IO_DATOUT	I/O DATA OUT Register
$30	IO_ENAB	I/O Enable Register
$22	WDTCR	Watchdog Timer Control Register
$21	BTCR	Bit Timer Control Register
$20	BTCNT	Bit Timer Count Register
$1E	DEEAR	Data EEPROM Address Register
$1D	DEEDR	Data EEPROM Data Register
$1C	DEECR	Data EEPROM Control Register
$17	TXCR7	Transmitter Configuration Register 7
$14	TXCR4	Transmitter Configuration Register 4
$12	CTL0	Transmitter Control Register Zero

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

I/O and Control Registers

The AT86RF401 I/Os and peripherals are placed in the I/O space. The various 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 chapter for more details. The different I/O and peripherals control registers are explained in the following sections.

Transmitter Control Register Descriptions

Transmit Control Register Zero – CTL0

Bit	7	6	5	4	3	2	1	0
$12	–	FSK	TXE	TXK	–	LOC	–	–
Read/Write	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
Initial Value	0	0	0	0	0	–	0	0

- **Bit[7]**

Reserved.

- **Bit[6]:**

This bit is reserved; must be programmed to 0.

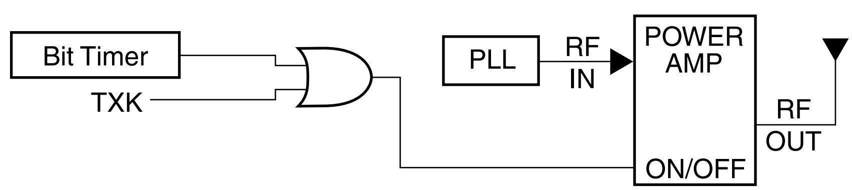
- **Bit[5]: Transmitter Enable**

This bit turns on the transmitter. The user should wait for lockdetect to be asserted before enabling the bit timer, or keying the transmitter.

- **Bit[4]: Transmitter Key**

This bit is OR'ed with the output from the bit timer. If the bit timer is used to key the transmitter, the TXK bit should be programmed to 0. If the bit timer is not used, this bit may be used to manually key the transmitter.

Figure 22. Modulation Control Logic



- **Bit[3]**

Reserved.

- **Bit[2]: LOCKDETECT**

This bit is set when the frequency synthesizer in the transmitter is locked. Usually this bit should be set before transmitting. For details on locktime, see the RF specifications.

- **Bit[1:0]**

Reserved.

Transmitter Configuration Register 4 – TXCR4

Bit	7	6	5	4	3	2	1	0
$14	RD1	RD0	–	–	–	PC2	PC1	PC0
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

- **Bits[7:6]: Refdiv[1:0]**

Refdiv determines the reference frequency of the frequency synthesizer. For most applications, these bits are set to 0.

Refdiv[1:0]	F_{ref}
00	F_{xtal}
01	$F_{xtal}/2$
10	$F_{xtal}/4$
11	$F_{xtal}/8$

- **Bits[5:3]**

Reserved.

- **Bits[2:0]: Power Control**

Attenuate the output power in 1 dB steps.

TP[2:0]	Output Power
000	No Attenuation
001	1 dB Attenuation
010	2 dB Attenuation
011	3 dB Attenuation
100	4 dB Attenuation
101	5 dB Attenuation
110	5 dB Attenuation
111	5 dB Attenuation

Transmitter Configuration Register 7 – TXCR7

Bit	7	6	5	4	3	2	1	0
$17	–	–	LDD5	LDD4	LDD3	LDD2	LDD1	LDD0
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	0

- **Bit[7:6]:**

Reserved.

- **Bits[5:0]: Lock Detect Delay Value**

Number of cycles of f_{ref} from last phase-detector cycle slip until LOCKDETECT is asserted. Maximum phase error detection and phase slip detection are used. Both conditions must be met before LOCK DETECT is asserted.

EEPROM Control Register Descriptions

Data EEPROM Control Register – DEECR

Bit	7	6	5	4	3	2	1	0
$1C	–	–	–	–	BSY	EEU	EEL	EER
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Initial Value 0	0	0	0	0	0	0	0	0

- **Bits[7:4]**

Reserved. These bits should be zero when written, otherwise results will be unpredictable.

- **Bit[3]: EEPROM Busy bit**

Initially set to 0. This bit will be set high during writes to the EEPROM.

- **Bit[2]: EEPROM Unlock bit**

Set this bit to "1" before writing the EEPROM. Reset this bit to "0" after the write is complete. This bit should be left in the zero state when the EEPROM is not being used. That way during power transients the EEPROM data will be protected.

- **Bit[1]: EEPROM Load bit**

To write the EEPROM use the following procedure:

Note: Because of noise and power considerations, the EEPROM should not be written while the transmitter is enabled.

1. Set the Unlock bit.
2. Write the address of the first byte to the DEEAR.
3. Set the load bit. This locks the page address in the DEEAR. Keep the unlock bit set.
4. Write the desired data to the DEEDR register. This byte is loaded into the EEPROM and will be written when the load bit is later de-asserted.
5. If it is desired to write another byte in the same page, write the new address to the DEEAR register, and a new byte to the DEEDR register. Continue until all bytes that are to be written are loaded into the EEPROM. Bytes may only loaded to an address once. There are 8 bytes per page.
6. De-assert the load bit. This starts the write operation. Some time after load is de-asserted, the busy bit will go high. Another read or write operation may not be started until the busy bit has returned to zero. Writes take approximately 4 ms to complete. Again, the unlock bit must still be set when de-asserting the load bit.
7. After the all writes are complete, write zero to the unlock bit.

- **Bit[0]: EEPROM Read bit**

To read the EEPROM use the following procedure.

1. Write the address to the DEEAR.
2. Set the read bit.
3. Read the data register. The read bit will reset itself.
4. If another read needs to be done, repeat number 1 to 3 again.

Data EEPROM Data Register – DEEDR

Bit	7	6	5	4	3	2	1	0
$1D	ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
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	0

- **Bits[7:0]**

This register contains the byte to be written to EEPROM. If a read operation has been done, this register contains that last byte read from the data EEPROM.

Data EEPROM Address Register – DEEAR

Bit	7	6	5	4	3	2	1	0
$1E	–	PA6	PA5	PA4	PA3	BA2	BA1	BA0
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	0

- **Bit[7]**

Reserved.

- **Bits[6:3]: Data EEPROM page address**

These bits select the page in the eeprom that is to be accessed. These bits are write locked and cannot be altered when the load bit is set.

- **Bits[2:0]: Data EEPROM byte address**

These bits select the byte in the page that is to be accessed. During a page write operation, these bits are used in combination with the DEEDR register to write bytes into a page.

Bit Timer Register Descriptions

Bit Timer Count Register – BTCNT

Bit	7	6	5	4	3	2	1	0
$20	C7	C6	C5	C4	C3	C2	C1	C0
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 [7:0]**

Lowest 8 bits of *countval*.

Bit Timer Control Register – BTCR

Bit	7	6	5	4	3	2	1	0
$21	C9	C8	M1	M0	IE	F2	DATA	F0
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[7:6]**

Count_val[9:8].

- **Bits[5:4]**

Bit Timer Mode.

Mode[1:0]	Bit Timer Function
00	Bit Timer Disabled
01	Transmit Mode, Transmitter not Keyed
10	Receive Mode
11	Transmit Mode, Transmitter Keyed

- **Bit[3]: Interrupts enabled**

If this bit is set, the flag2 and flag0 will generate their respective interrupts when they are set.

- **Bit[2]: Flag2**

In transmit mode, this flag indicates the transmit done condition that occurs when the buffer is empty and the counter has count down to zero. In receive mode, this flag indicates that an edge has occurred, and the AVR should process the count value in the BTCR and BTCNT registers.

- **Bit[1]: Data bit**

In transmit modes, this is a one-bit buffer that the AVR writes data to and the bit timer extracts data from. In receive mode, the value in this register indicates whether the edge at the IO3 pin was rising or falling. A one indicates a rising edge occurred, while a zero indicates that a falling edge was detected.

- **Bit[0]: Flag0**

In transmit mode, this flag indicates the buffer is empty and the AVR should load new data into it. In receive mode, this indicates an counter overflow condition has occurred. The AVR should increment its software counter if this condition has occurred.

Watchdog Timer Control Register – WDTCR

Bit	7	6	5	4	3	2	1	0	
$22	–	–	–	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]**

Reserved. These bits will always read as zero.

- **Bit[4]: WDTOE, Watchdog Turn-off Enable**

This bit must be set (one) when the WDE bit is cleared. Otherwise, the watchdog will not be disabled. Once set, 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, Watchdog 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: 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. Within the next four clock cycles, write a logical 0 to WDE. This disables the watchdog.

- **Bits[2:0]: WDP2, WDP1, WDP0, Watchdog 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 6.

Table 6. Watchdog Timer Prescale Select

WDP2	WDP1	WDP0	Number of System Clock Cycles
0	0	0	2048 cycles
0	0	1	4096 cycles
0	1	0	8192 cycles
0	1	1	16,384 cycles
1	0	0	32,768 cycles
1	0	1	65,536 cycles
1	1	0	131,072 cycles
1	1	1	262,144 cycles

Note:

$$T_{wdt} = XTALB_{period} \times ACS_{div} \times WDT_{div}$$

Example:

If the crystal period is 50 ns and the system clock divider is set to 32 (Bits[7:5] in the PWR_CTL register are set to 010) and the WDT prescaler is set to 32k:

Watchdog Timeout =

$$50ns \times 32 \times 32768 = 52ms$$

I/O Enable Register – IO_ENAB

Bit	7	6	5	4	3	2	1	0
$30	–	BOHYST	IOE5	IOE4	IOE3	IOE2	IOE1	IOE0
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]**

Reserved.

- **Bit[6] BOHYST**

If 0, extra hysteresis is added to the battery low and brown-out logic. See BL_CONFIG register description.

- **Bits[5:0]**

If set to “1”, the corresponding bit (pin) IO[5:0] is configured as an output. Data may then be written to that output by writing to the IO_DATA register. If set to zero, the corresponding bit (pin) may be either a button input (refer to the Button Detect Register, $34) used to wake the part up or a normal digital input.

IO_ENAB[n]	IO_DAT_OUT[n]	IO[n]
0	0	Normal Input
0	I	Button Input
I	0	Output Driven Low
I	I	Output Driven High

I/O Data Out Register – IO_DATOUT

Bit	7	6	5	4	3	2	1	0
$31	–	–	IOO5	IOO4	IOO3	IOO2	IOO1	IOO0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bits[7:6]: Reserved**

These bits read 0.

- **Bits[5.0]**

If enabled in the IO_ENAB register and not in test mode, the data in Bits[5:0] goes to the corresponding general-purpose output IO [5:0].

I/O Data In Register – IO_DATIN

Bit	7	6	5	4	3	2	1	0
$32	–	–	IOI5	IOI4	IOI3	IOI2	IOI1	IOI0
Read/Write	R/W	R/W	R	R	R	R	R	R
Initial Value	0	0						

- **Bits[7:6]: Reserved**

This bit reads 0.

- **Bits[5:0]**

These bits directly read the data from the I/O pins IO[5:0]. Writes to these bits have no effect.

Power Control Register – PWR_CTL

Bit	7	6	5	4	3	2	1	0
$33	ACS2	ACS1	ACS0	TM	BD	BL	SLEEP	BBM
Read/Write	R/W	R/W	R/W	R/W	R	R	W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bits[7:5]: AVR System Clock Select**

These bits select the divide value of the XTAL input which is used to produce the AVR System Clock.

ACS[2:0]	AVR System Clock
101, 110, 111	XTLB/4
100	XTLB/8
011	XTLB/16
010	XTLB/32
001	XTLB/64
000	XTLB/128

This clock select value may be programmed on the fly by either the AVR processor in normal operation or by an I/O write SPI command during SPI mode. Note that during SPI mode, the I/O and serial programming logic runs at XTLB/4 frequency.

- **Bit[4]: Reserved**

TEST MODE. When this bit is set to "1", the part enters test mode. The I/O pins, if enabled, assume the following functionality:

	IO5	IO4	IO3	IO2	IO1	IO0
Normal Mode (RESETB = 1)	txkey (Output)	lockdetect (Output)	txenable (Output)	RFU	RFU	RFU
SPI Mode (RESETB = 0)	txkey (Output)	lockdetect (Output)	txenable (Output)	SPI_CLK*	SDO*	SDI*

Notes: 1. IO_ENAB register is NOT used for SPI pins.
2. In normal mode, I05 and I04 (if enabled by setting Bits [5:4] in the IO-ENAB register) can be used to output the data being sent to the transmitter (txkey) and to observe the lock time of the frequency synthesizer.
3. In SPI mode, the I/O Registers may be directly accessed via the SPI interface. Txkey, lock-detect may be output using this mode.

- **Bit[3]: Battery Dead**

Indicates battery is dead. Only readable by SPI interface.

- **Bit[2]: Battery Low**

Indicates battery voltage is low.

- **Bit [1]: Sleep Bit**

When set, this bit stops the crystal oscillator. This stops the AVR processor with the program counter frozen at the current instruction. Sleep will also stop the watchdog timer. The watch-dog timer is only restarted if the part wakes up. If an I/O Pin is configured as a button, a button press will start the oscillator, and check the battery level. If the battery level is greater than the Battery Dead level, the AVR system clock is started and normal program execution continues. If the battery level is below the Battery Dead level, the crystal oscillator is turned off putting the part back to sleep until a button is pressed again (care should be taken not to put the part to sleep unless a button is configured and enabled).

- **Bit[0]: Button Boot Mode**

If the BBM bit is set, the part will enter the button reset state. This is a low-power state in which there is minimal power being dissipated. The part will re-boot if a properly configured button is pressed. This bit is reset during power-on reset, and when exiting the button reset state.

Button Detect Register – B_DET

Bit	7	6	5	4	3	2	1	0
$34	–	–	BD5	BD4	BD3	BD2	BD1	BD0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bits[7:6]**

Reserved. These bits read 0.

- **Bits[5.0]**

When an I/O PIN is configured as a button using the IO_ENAB and IO_DATOUT registers and a logic low is detected on that pin, the button detect logic is activated. If the part is in sleep mode, the part responds as described in the Power Control Register description. If a good battery is present, the appropriate bit is set in this register. A bit in this register is cleared by writing a 0 to it.

Battery Low Configuration Register – BL_CONFIG

Bit	7	6	5	4	3	2	1	0
$35	BL	BLV	BL5	BL4	BL3	BL2	BL1	BL0
Read/Write	R/W	R	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bit[7]: Battery Low**

When bit 6 in this register is set (Battery Low Valid), the BL (Battery Low) bit indicates that the battery voltage is lower than the voltage level that is determined by bits [5:0] of this register.

- **Bit[6]: Battery Low Valid**

When the Battery Low Configuration Register is written, this bit is set to 0. When the battery voltage has been sampled and compared to the voltage determined by the BLx bits, this bit is set to 1 indicating that the data in bit 7 (Battery Low) is valid. This can take up to 3100 XTAL cycles to complete.

- **Bit[5.0]: Battery Low Detection Level**

This value is sent to the battery monitor. The threshold is calculated using the formulas shown in Table 1. Hysteresis can be added to the brown-out monitor when the BOHYST bit is set to 0 in the IO_ENAB register.

The General Interrupt Flag Register – GIFR

Bit	7	6	5	4	3	2	1	0	
$3A ($5A)	INTF1	INTF0	–	–	–	–	–	–	GIFR
Read/Write	R/W	R/W	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – INTF1: External Interrupt Flag1**

When an event on the INT1 pin triggers an interrupt request, INTF1 becomes set (one). If the I-bit in SREG and the INT1 bit in GIMSK are set (one), the MCU will jump to the interrupt vector at address $004. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT1 is configured as a level interrupt.

- **Bit 6 – INTF0: External Interrupt Flag0**

When an event on the INT0 pin 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. This flag is always cleared when INT0 is configured as a level interrupt.

- **Bits 5..0 – Res: Reserved Bits**

These bits are reserved bits and always read as zero.

The General Interrupt Mask Register – GIMSK

Bit	7	6	5	4	3	2	1	0	
$3B ($5B)	INT1	INT0	–	–	–	–	–	–	GIMSK
Read/Write	R/W	R/W	R	R	R	R	R	R	
Initial Value	0	0	x	0	0	0	0	0	

- **Bit 7 – INT1: External Interrupt Request 1 Enable**

When the INT1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is activated. The Interrupt Sense Control1 bits 1/0 (ISC11 and ISC10) in the MCU general Control Register (MCUCR) define whether the external interrupt is activated on rising and/or falling edge of the INT1 pin or level sensed. Activity on the pin will cause an interrupt request even if INT1 is configured as an output. The corresponding interrupt of External Interrupt Request 1 is executed from program memory address $004. See also "External Interrupts".

- **Bit 6 – INT0: External Interrupt Request 0 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 activated. The Interrupt Sense Control0 bits 1/0 (ISC01 and ISC00) in the MCU general Control Register (MCUCR) define 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 External Interrupt Request 0 is executed from program memory address $002. See also "External Interrupts."

- **Bits 5 – Res: Reserved Bits**

This bit is reserved and the read value is undefined.

- **Bits 4..0 – Res: Reserved Bits**

These bits are reserved bits and always read as zero.

The Stack Pointer – SP

The AT86RF401 Stack Pointer is implemented as two 8-bit registers in the I/O space locations $3E ($5E) and $3D ($5D). As the AT86RF401 data memory has 224 locations, only 8 bits are used and the SPH register should be programmed to 0x00.

Bit	15	14	13	12	11	10	9	8	
$3E	–	–	–	–	–	SP10	SP9	SP8	SPH
$3D	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	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 the return address is pushed onto the Stack with subroutine call and interrupt. 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 data is popped from the Stack with return from subroutine RET or return from interrupt RETI.

The Status Register – SREG

The AVR status register – SREG – at I/O space location $3F is defined as:

Bit	7	6	5	4	3	2	1	0
$3F	I	T	H	S	V	N	Z	C
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 the interrupt mask registers - GIMSK/TIMSK. If the global interrupt enable register is cleared (zero), none of the interrupts are enabled, independent of the GIMSK/TIMSK values. 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 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$\oplus$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.

Instruction Set Memory

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
ADIW	Rdl,K	Add Immediate to Word	Rdh:Rdl ← Rdh:Rdl + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
SBIW	Rdl,K	Subtract Immediate from Word	Rdh:Rdl ← Rdh:Rdl - K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← $FF – Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← $00 – Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd • ($FF - K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd – 1	Z,N,V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← $FF	None	1
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
IJMP		Indirect Jump to (Z)	PC ← Z	None	2
JMP	k	Direct Jump	PC ← k	None	3
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
ICALL		Indirect Call to (Z)	PC ← Z	None	3
CALL	k	Direct Subroutine Call	PC ← k	None	4
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1 / 2 / 3
CP	Rd,Rr	Compare	Rd – Rr	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	Rd – Rr – C	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	Rd – K	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) PC ← PC + 2 or 3	None	1 / 2 / 3
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) PC ← PC + 2 or 3	None	1 / 2 / 3

Mnemonics	Operands	Description	Operation	Flags	#Clocks
BRANCH INSTRUCTIONS (Continued)					
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC←PC+k + 1	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC←PC+k + 1	None	1 / 2
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1 / 2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1 / 2
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z+1	None	3
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1

Mnemonics	Operands	Description	Operation	Flags	#Clocks
BIT AND BIT-TEST INSTRUCTIONS (Continued)					
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Two's Complement Overflow	V ← 1	V	1
CLV		Clear Two's Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
NOP		No Operation		None	1
SLEEP		Sleep	Not Implemented	None	3
WDR		Watchdog Reset	(See specific description for WDR/timer)	None	1

Ordering Information

RF Output	Ordering Code	Package	Application	Temperature Operating Range
315 MHz	AT86RF401U	20T	North American	Industrial (–40°C to 85°C)
434 MHz	AT86RF401E	20T	European	Industrial (–40°C to 85°C)
250 to 450 MHz	AT86RF401X	20T	All Applications	Industrial (–40°C to 85°C)

Package Type	
U	Suffix denotes part with an internal loop filter optomized for 315 MHz operation.
E	Suffix denotes part with an internal loop filter optomized for 434 MHz operation.
X	Suffix denotes part requires an external loop filter for operation between 250 MHz and 450 MHz.

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel SarL
Route des Arsenaux 41
Casa Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
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

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Atmel Smart Card ICs
Scottish Enterprise Technology Park
East Kilbride, Scotland G75 0QR
TEL (44) 1355-357-000
FAX (44) 1355-242-743

Atmel Grenoble
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex
France
TEL (33) 4-7658-3000
FAX (33) 4-7658-3480

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

1424B–03/01/xM