8-bit AVR Microcontroller

AVR042: AVR Hardware Design Considerations

APPLICATION NOTE

Introduction

This application note provides basic guidelines to be followed while designing hardware using Atmel® AVR® microcontrollers. Some of the known problems faced in real-time designs have been addressed by providing possible solutions and work-arounds to resolve them.

The scope of this application note is to provide an introduction to potential design problems rather than being an exhaustive documentation on designing applications using the AVR microcontrollers. This document is a collection of information from existing Atmel AVR documents along with some additional information that were not documented earlier.

Note: Read the application note AVR040 - "EMC Design Considerations" – before starting a new design, especially if the design is expected to meet the requirements of the EMC directive or any other similar directives in countries outside Europe.

Features

- Guidelines for providing robust analog and digital power supply
- Connection of RESET line
- Interfacing Programmers/Debuggers to AVR
- Using external Crystal or Ceramic Resonator Oscillators

Table of Contents

1. Abbreviations

ADC	Analog to Digital Converter
AREF	Analog Reference Voltage
CPU	Central Processing Unit
DC	Direct Current
DIP	Dual In-line Package
EEPROM or E^2PROM	Electrically Erasable Programmable Read-Only Memory
EMC	Electromagnetic Compatibility
ESD	Electrostatic Discharge
GND	Ground
HVPP	High-Voltage/Parallel Programming
Hz	Hertz
I/O	Input and Output
IDE	Integrated Development Environment
ISP	In-System Programming
KHz	KiloHertz
LED	Light Emitting Diode
MCU	Microcontroller Unit
MHz	MegaHertz
MISO	Master In Slave Out
MOSI	Master Out Slave In
PCB	Printed Circuit Board
PDI	Program and Debug Interface
RC Filter	Resistor-Capacitor Filter
RST	Reset
SPI	Serial Peripheral Interface
TPI	Tiny Programming Interface
UPDI	Unified Program and Debug Interface
V_{CC}	Supply Voltage
XTAL	Crystal Oscillator

2. Power Supply

Power Supply is the most critical part of any hardware design, which directly affects the performance of the system. Two important aspects to be considered while designing a power supply for the discrete/ digital elements of an Atmel AVR are **ESD Protection** and **Noise Emission**. These aspects are covered in-detail the AVR040 application note and hence only a short summary is included in this document.

2.1. Digital Supply

Most AVR microcontrollers operate over a wide voltage range and draws only a few milliamps of supply current. This may give an impression that power supply is not critical. But as with any digital circuits, the supply current is an average value. The current is drawn in very short spikes on the clock edges. If I/O lines are switching, the spikes will be even higher. If all eight I/O lines of an I/O port changes value, simultaneously, the current pulses on the power supply lines can be several hundred mA. If the I/O lines are not loaded, the pulse will last for only a few nanoseconds.

Such a current spike cannot be delivered over long power supply lines; the main source is (or should be) the decoupling capacitor.

Figure 2-1. Incorrect Decoupling

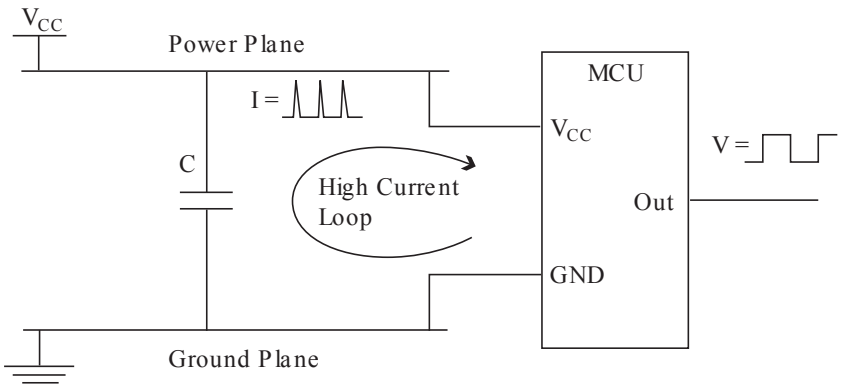
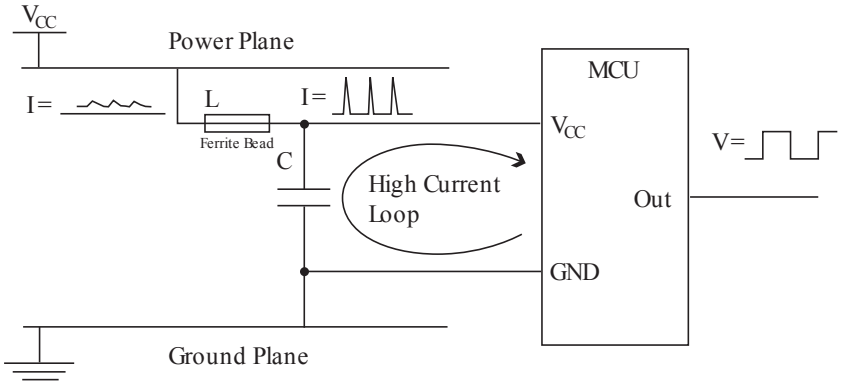


Figure 2-1 shows an example of insufficient decoupling. The capacitor is placed too far away from the microcontroller, creating a larger high current loop. The power and ground planes are parts of the high current loop. As a result of this, noise is spread more easily to other devices on the board, and radiated emission from the board is increased even further. The whole ground plane will act as an antenna for the noise, instead of only the high current loop. This will be the case when the power and ground pins are connected directly to the planes (typical for hole-mounted components) and the decoupling capacitor is connected the same way. This is often seen in boards with surface-mount components where the integrated circuits are placed on one side of the board and the decoupling capacitors are placed on the other side.

Figure 2-2 shows a better placement of the capacitor. The lines that are part of the high current loop are not part of the power or ground planes. This is important, as the power and ground planes otherwise will spread a lot of noise. Further, the figure shows another improvement in the decoupling. A series ferrite bead is inserted to reduce the switching noise on the power plane. The series impedance of the ferrite bead must be low enough to ensure that there is no significant drop in the DC voltage.

Figure 2-2. Decoupling with Series Inductor



In Atmel AVR devices where power and ground lines are placed close together there will be better decoupling than the devices with industry standard pin-out. In industry standard pin-out, the power and ground pins are placed in opposite corners of the DIP package. This disadvantage can be overcome by placing decoupling capacitors very close to the die. For devices with multiple pairs of power and ground pins, it is essential that there is a decoupling capacitor for every pair of pins.

The main power supply should also have a tantalum or ceramic capacitor to stabilize it.

2.2. Analog Supply

The AVR devices that have a built-in ADC that have a separate analog supply voltage pin, A_{Vcc}. This separate voltage supply ensures that the analog circuits are less prone to the digital noise originating from the switching of the digital circuits.

To improve the accuracy of the ADC, the analog supply voltage must be decoupled separately, similar to the digital supply voltage. AREF must also be decoupled. The typical value for the capacitor is 100nF. If a separate analog ground (AGND) is present, the analog ground should be separated from the digital ground so that the analog and digital grounds are only connected at a single point (at the power supply GND).

2.3. Noise Implications

When AVR devices are operate at CPU speeds around 2MHz with varying supply voltage and/or temperature conditions, it is affected by noise issues. These noise related issues are prominent after power-up, wake-up, or after any changes to the clock prescaler.

To resolve such issues, select either lower or higher CPU speeds and to use high quality low-noise digital and analog power supply.

3. Connection of RESET Pin on AVRs

The RESET pin on the AVR is active LOW, and setting the pin LOW externally will reset of the AVR. The RESET has two purposes:

1. To release all the lines by tri-stating all pins (except XTAL pins), initialize all I/O registers and set program counter to zero.
2. To enter programming mode (for some parts, the PEN line is also used to enter programming mode). It is also possible to enter High-Voltage/Parallel Programming (HVPP) mode by drawing the RESET pin very high (11.5V – 12.5V). Refer to the respective device datasheet for more specific information about RESET pin and its functionality.

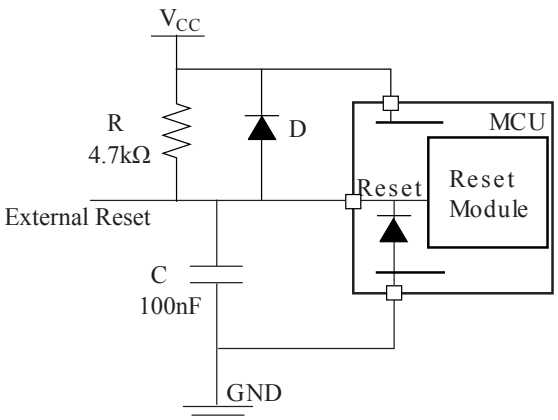
The reset line has an internal pull-up resistor. But if the environment is noisy, it can be insufficient and reset may occur sporadically. Refer to the device datasheet for the value of pull-up resistor that must be used for specific devices.

Connecting the RESET such that it is possible to enter both high-voltage programming and ordinary low level reset can be achieved by using a pull-up resistor to the RESET line. This pull-up resistor avoids any unindented low signal that will trigger a RESET. Theoretically, the pull-up resistor can be of any value, but if the Atmel AVR should be programmed using an external programmer, the pull-up should not be in such a high state that the programmer is not able to activate RESET by drawing the RESET line low. The recommended pull-up resistor value is 4.7kΩ or larger when using STK600 for programming. For DebugWIRE to function properly, the pull-up must not be lesser than 10kΩ.

To protect the RESET line from further noise, connect a capacitor from the RESET pin to ground. This is not directly required since the AVR internally have a low-pass filter to eliminate spikes and noise that could cause reset. Useing an extra capacitor is an additional protection. However, such extra capacitor cannot be used when DebugWIRE or PDI is used.

ESD protection diode is not provided internally from RESET to Vcc in order to allow HVPP. If HVPP is not used, it is recommended to add an ESD protection diode externally from RESET to Vcc. Alternatively, a Zener diode can be used to limit the RESET voltage relative to GND. A Zener diode is highly recommended in noisy environments. The components should be located physically close to the RESET pin of the AVR. Recommended circuit on RESET line is shown in the following circuit diagram.

Figure 3-1. Recommended Reset Pin Connection



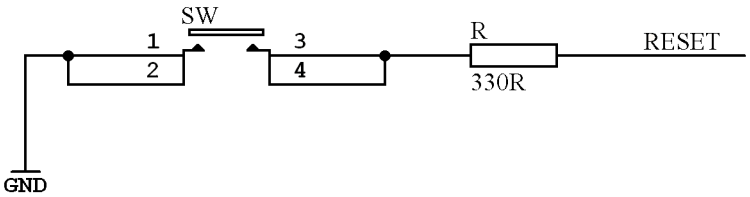
Note: The values of resistor R and capacitor C are typical values used for the RESET pin. For specific design requirements of an application, these values must be changed accordingly.

3.1. External RESET Switch

If an external switch is connected to the RESET pin, it is important to add a series resistance. Whenever the switch is pressed, it will short the capacitor and the current **(I)** through the switch can have high peak values. This causes the switch to bounce and generate steep spikes in 2ms - 10ms **(t)** periods until the capacitor is discharged. The PCB tracks and the switch metal introduces a small inductance **(L)** and the high current through these tracks can generate high voltages up to $\mathbf{V_L = L * dI / dt}$.

This spike voltage $\mathbf{V_L}$ is most likely outside the specification of the RESET pin. By adding a series resistor between the switch and the capacitor, the peak currents generated will be significantly low and it will not be large enough to generate high voltages at the RESET pin. An example connection is shown in the following diagram.

Figure 3-2. Switch Connection for Reset Pin



4. Connecting Programmer / Debugger Lines

Atmel AVR microcontrollers feature one or more interfaces for Programming or Debugging. In-System Programming (ISP) is a programming interface used for programming the Flash, EEPROM, Lock-bits, and Fuse-bits in almost all AVRs. This feature makes it possible to program the AVR in the last stage of production of a target application board, reprogram if SW bugs are identified late in the process, or even update the AVR in the field, if required. Some ISP interfaces may also be used for on-chip debugging. It is therefore recommended to design the target application board such that the ISP connectors are easily accessible.

Note: Refer the respective device datasheet to know the programming / debugging interfaces support by the device.

4.1. SPI Programming Interface

On devices that use a Serial Peripheral Interface (SPI) for ISP, these lines are usually located on the same pins as regular SPI, or else on pins that can be used for other purposes. Refer the device data sheet to determine the pins are used for ISP.

Two standard SPI connectors are provided by the Atmel ISP programmers; a 6-pin and a 10-pin connector. In addition to the data lines (MOSI and MISO) and the bus clock (SCK), target voltage VTG, GND, and RESET (RST) are also provided through these connectors.

Figure 4-1. Connections for 6/10 Pin ISP Headers

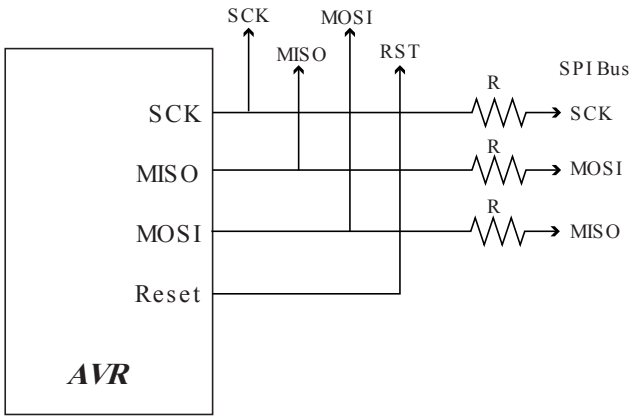
A few ISP programmers are powered by the target power supply. In this way they easily adapt to the correct voltage level of the target board. Other ISP programmers, such as Atmel STK600 can alternatively power the target board via the VTG line. In such a case, it is important that the power supply on the target is not switched on.

Note: Refer the respective programmer user guide for more information about the capabilities and physical interface.

4.1.1. Shared Use of SPI Programming Lines

If additional devices are connected to the ISP lines, the programmer must be protected from any device that may try to drive the lines, other than the AVR. This is important with the SPI bus, as it is similar to the ISP interface. Applying series resistors on the SPI lines, as depicted in Connecting the SPI Lines to the ISP Interface, is the easiest way to achieve this. Typically, the resistor value R can be of 330Ω(1).

Figure 4-2. Connecting the SPI Lines to the ISP Interface



Note:

1. These typical values are used to limit the input current to 10mA for a supply voltage (V_{CC}) of 3.3V. It may vary depending on the programmer/debugger used and requirements of specific hardware design.
2. The AVR will never drive the SPI lines in a programming situation. Since the AVR is held in RESET to enter programming mode, which puts all AVR pins to tri-states.

In a single application, multiple Atmel AVR devices can share the same ISP interface. This enables to program all the devices through a minimal interface. However, if there are no special design considerations, then all the AVR devices will respond to the ISP instructions. The SPI clock lines should be seperately provided (can be gated using jumpers or DIP switches) so that only one of the AVR devices recieves SPI clock at a time. Other SPI lines (MOSI & MISO) can be shared. This method ensures that AVRs are seperated from the programmer by the same protection resistors, since they are all held in RESET while the ISP reset line is activated. The ISP clock can be gated using jumpers or DIP switches.

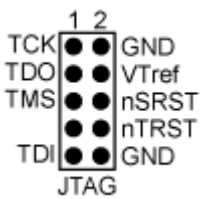
An alternate solution is to use multiple ISP interfaces, one for each device, all protected seperately with series resistors.

4.2. JTAG Interface

Few devices have a JTAG interface that can be used for both programming and debugging. The JTAG lines are shared with analog input and mustbe connected such that the JTAG programmer can control the lines. JTAG programming tools can drive a resistive load, however, it is better to avoid capacitive load.

The following figure shows the standard JTAG connector supplied with Atmel ISP programmers. For the SPI programming connector, the target's voltage supply allows to power the device or ensuring correct signal levels when programming.

Figure 4-3. Pinout of the Standard JTAG Connector

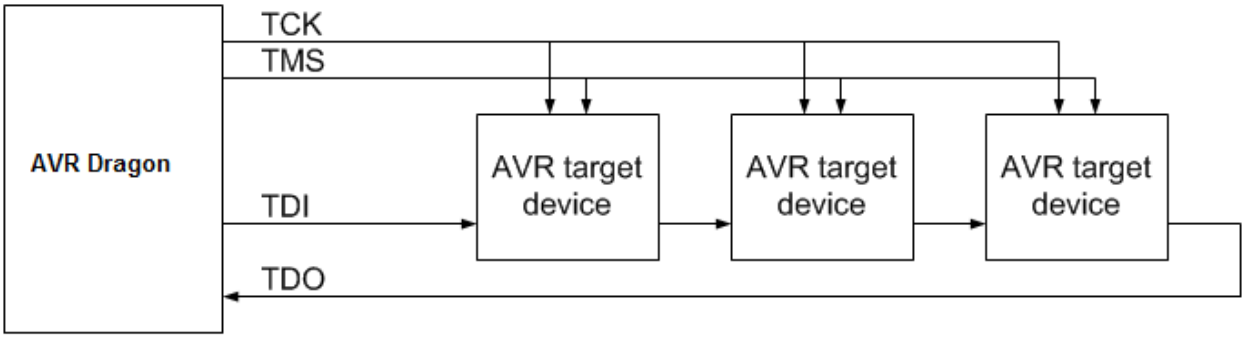


Note: Refer the specific user guide of programmers / debuggers for more information about the JTAG interfacing with AVR.

4.2.1. Shared Use of JTAG Lines

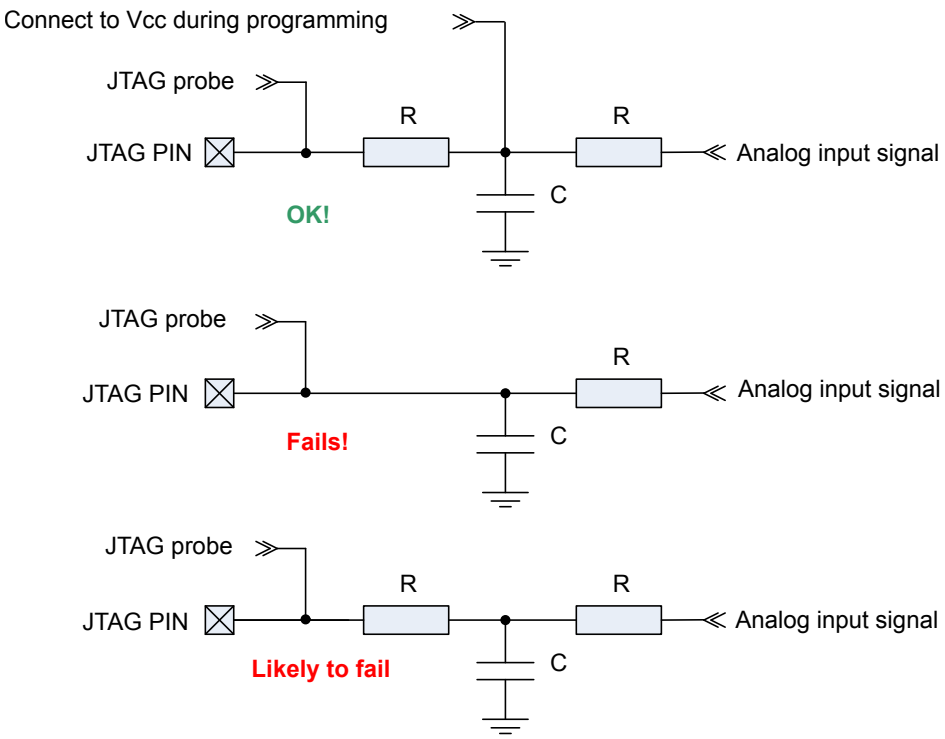
By creating a JTAG daisy-chain, a single JTAG connector can serve as an ISP interface for several devices. Typical connection for a daisy-chain using JTAG for AVR Dragon is shown in the following schematic. The daisy-chain configuration can be used for any programmer/debugger that uses JTAG interface. The GND and VT_{REF} of JTAG, which is not shown in the figure, must be connected to the target board.

Figure 4-4. JTAG Daisy-Chain



The protection resistors shown in Figure 4-2 are required if the JTAG lines are used in the application. For example, ADC input pins often have analog filters on the lines. In such cases, the filter capacitor must be removed while programming, to ensure that the load is resistive. The following figure illustrates the steps.

Figure 4-5. JTAG Interface Connections – Correct and Incorrect Ways

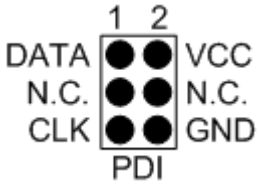


4.3. PDI Interface

The Program and Debug Interface (PDI) is an Atmel proprietary two-line interface that was introduced with the Atmel AVR XMEGA™ microcontroller family. As the name implies, this interface can be used for both In-System Programming and On-chip debugging of devices.

The following figure shows the standard PDI connector supplied with Atmel programmers. Only two pins on the device are required for using this interface: RESET, also called PDI_CLK, and the dedicated PDI_DATA pin. The target's voltage supply allows to power the device or ensuring correct signal levels during programming.

Figure 4-6. Standard PDI Header



Note: Refer the respective programmer user guide for more information about the capabilities and physical interface of PDI.

4.3.1. External Reset Circuitry

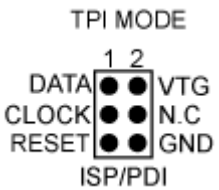
Since the RESET line is used for clocking the PDI, it is important to bypass or avoid any circuitry that can distort the clock signal during programming or debugging such as capacitors and external reset sources. During normal operation, the RESET pin has an internal filter to prevent unintentional resets such as those caused by short spikes on the reset line. Despite the fact that the clock signal is deformed, capacitive loads up to 1nF have been tested to work with the Atmel STK600, and AVR Dragon™ during programming. Pull-up resistors should be at least 10kΩ, or removed from the RESET line, if an Atmel programmer is used.

4.4. TPI Interface

The Tiny Programming Interface is featured on the Atmel tinyAVR® devices with the lowest pin count.

The following figure shows the standard TPI connector supplied along with the Atmel programmer device. Only three pins on the device are required for use of this interface: RESET, TPICLK, and TPIDATA. The latter two pins are multiplexed with regular I/O pins.

Figure 4-7. Standard TPI Header



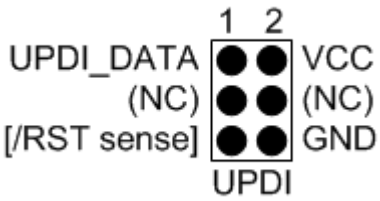
The RESET pin can be reconfigured as an I/O pin by programming the RSTDISBL fuse of the device. This disables the reset functionality and requires +12V to be applied to RESET for programming to work. Only a few programming tools are capable of generating this voltage.
Note: Refer the respective programmer user guide for more information about the capabilities and physical interface of TPI.

4.5. UPDI Interface

The Unified Program and Debug Interface (UPDI) is an Atmel proprietary interface for external programming and on-chip debugging of a device.

Programming and debugging are performed using the UPDI Physical interface (UPDI PHY), which is UART based half-duplex 1-wire interface for data reception and transmission. It uses the Reset line to detect the debugger probe.

Figure 4-8. Standard UPDI Header



Single wire interface can be enabled by setting a fuse or by 12V programming, which disables the reset functionality. Not all programming tools are capable of generating this voltage.

Note: Refer the respective programmer user guide for more information about the capabilities and physical interface of UPDI.

5. Using Crystal and Ceramic Resonators

Most Atmel AVR MCUs can use different clock sources. The optional external clock sources are external clock, RC oscillator, crystal or ceramic resonator. The use of crystals and ceramic resonators are in some designs causing problems due to the fact that the use of these clock sources is not well understood. This section therefore treats the topic of using crystals and ceramic resonators in relation to Atmel AVR MCUs. The description focus on features and parameters relevant for designing applications where crystals or ceramic resonators are used rather than trying to be a complete description of the theory related to the topic. For more information and theory regarding crystals, refer to application note "AVR4100: Selecting and testing 32kHz crystal oscillators for AVR microcontrollers".

5.1. Selecting the Clock Source in the AVR

The clock source used by the AVR devices are selected by setting the appropriate fuses. However, for the Atmel AVR XMEGA family, the clock source is configured using software. Most ISP and parallel programmers can program the fuses for selecting a clock source. The fuses are not erased when the AVR memory is erased and the fuses must only be programmed if the fuse settings should be altered. Programming the fuses each time the device is erased and reprogrammed is thus not necessary. The clock options that are relevant for this document are:

- External low-frequency crystal
- External crystal oscillator
- External ceramic resonator

Several sub-settings relating to the start-up time of the AVR can be selected, but the three clock options mentioned are the fundamental settings that should be focused on. The clock options can vary across different AVR devices, as not all devices support external oscillators. Refer the datasheet for the specific device to determine the available clock options.

The AVR might not run if a different clock source is selected other than the clock source actually configured. Since the oscillator circuits are activated internally in the AVR, based on the configured clock option. The fuses are not cleared by a memory erase. Hence, it can cause problems if incorrect settings are selected.

5.2. About Crystals and Ceramic Resonators

The typical crystal used for the AVR is the AT-cut parallel resonant crystal. The ceramic resonator is very similar to the AT-cut parallel resonant crystal, but it is a low cost, low quality version of the crystal. The ceramic resonator has a lower Q-value, which is both an advantage and disadvantage. Due to the lower Q-value the oscillator frequency of the ceramic resonator can more easily be tuned to a desired frequency. But, it is also more sensitive to temperature and load changes, causing undesired frequency variations. The advantage of the ceramic resonators is that it has a faster start-up than crystals.

In this section, the term *resonator* refers both Quartz Crystals and Ceramic Resonator.

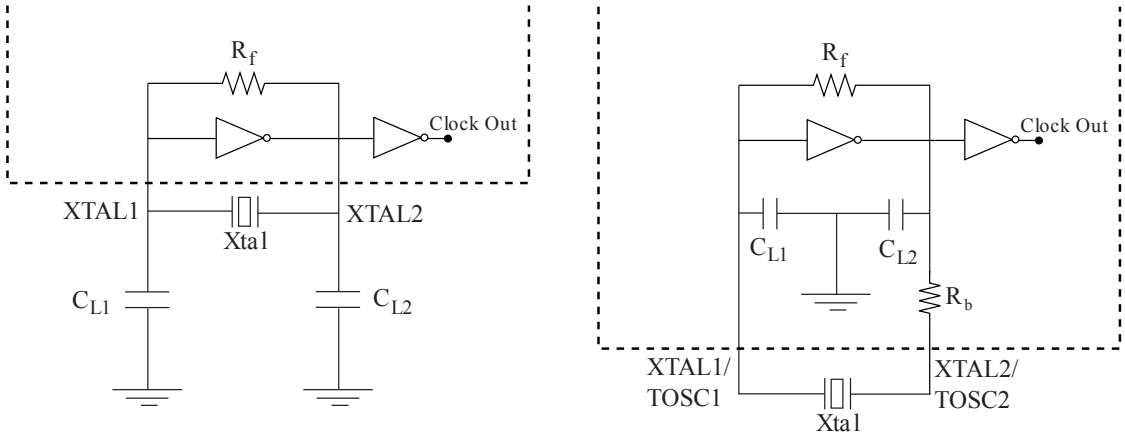
	Ceramic Resonator	Quartz Crystal
Aging	±3000ppm	±10ppm
Frequency tolerance	±2000 - 5000ppm	±20ppm
Frequency temperature characteristics	±20 - 50ppm/˚C	±0.5ppm/˚C
Frequency pullability	±100 - 350ppm/pF	±15ppm/pF

	Ceramic Resonator	Quartz Crystal
Oscillator rise time	0.01ms - 0.5ms	1ms - 10ms
Quality factor (Qm)	100 - 5000	103 – 5 x 105

Note: The information provided in the table are provided to showcase the differences. For more details about the oscillator, refer the specific part datasheet.

The parallel resonator is used in circuits which contains reactive components such as capacitors. Such circuits depend on the combination of the reactive components and the resonator to accomplish the phase shift required to start and maintain the oscillation at a specific frequency. Basic oscillator circuits used for parallel resonators are illustrated in the following diagram. The part of the circuit above the dashed line, represent the oscillator circuit present internally in the AVR. Simply, Atmel AVR built-in oscillator circuits can be understood as an inverter based oscillator circuit shown in the following figure.

Figure 5-1. Basic Inverter Circuits Equivalent to the Oscillator Circuits in AVRs



(1) (2)

1. Oscillator circuit for crystals and ceramic resonators faster than 400kHz.
2. Circuit for low frequency crystals (32.768kHz). This is not present on all Atmel AVR.

A circuit with resonator frequency beyond 400kHz is depicted in (1). In this circuit, capacitive load must be applied externally. The oscillator circuit seen in (2) is used for low frequency crystals on a few AVRs that are optimized for 32.768kHz crystals. This circuit provides the capacitive load required by the crystal internally. Further, it adds the resistor R_b to bias the crystal and limit the drive current into the crystal. When using CMOS inverters, the resistor R_f (~1MΩ) provides a feedback to bias the inverter and operate it in its linear region.

Note: Refer datasheet of specific device to see if it has internal circuitry for low frequency crystals.

When using resonators with the Atmel AVR, it is necessary to apply (external) capacitors according to the requirements of the resonator used. A parallel resonator will not be able to provide stable oscillation if insufficient capacitive load is applied. When the capacitive load is too high, the oscillation may not start as expected due to drive level dependency of the load. The trick is therefore to find an appropriate value for the capacitive load. The value to look for in the data sheet of the crystal is C_L, the recommended capacitive load of the resonator (viewed from the terminals of the resonator). The capacitive load (C_L) of the oscillator circuit, including stray capacitances and the capacitances of the XTAL pins of the AVR can be determined empirically or it can be estimated by the following equation.

Equation - 1

$$C_L = \frac{C'_{L1} \cdot C'_{L2}}{C'_{L1} + C'_{L2}} \qquad C'_{L1} = C_{L1} + C_{L1S} \qquad C'_{L2} = C_{L2} + C_{L2S}$$

Where C_{L1} and C_{L2} refer to the external capacitors seen in Figure 5-1 and C_{L1S} and C_{L2S} are stray capacitances at the XTAL pins of the AVR. Assuming symmetric layout, so that $C_{L1} = C_{L2} = C$ and $C_{L1S} = C_{L2S} = C_S$ (CS can be estimated to be 5pF - 10pF), then the external capacitors can be determined by following equation.

Equation - 2

$$C = 2 \cdot C_L - C_S$$

5.3. Recommended Capacitor Values

The recommendations are applicable for most of the application designs. However, a generic value cannot be provided for the external capacitors as they may not work as expected with all resonators.

When using the external crystal oscillator, crystals with a nominal frequency range starting from 400kHz can be used. For the standard high frequency crystals, the recommended capacitor value range is in the range of 22pF - 33pF.

The external low frequency crystal is intended for 32.768kHz crystals. When selecting this clock source, the internal oscillator circuit might provide the required capacitive load. By programming the CKOPT Fuse(1), the user can enable internal capacitors on XTAL1 and XTAL2. The value of the internal capacitor is typical 20pF, but can vary. External capacitors are not required when using a 32.768kHz crystal that does not require more load. Then the value of the external capacitor can be determined using the *Equation - 2*. The CKOPT Fuse should not be programmed when using external capacitors.

In other cases, an external capacitive load specified by the manufacturer of the crystal must be used.

For using the external ceramic resonator, refer the device datasheet for determining the capacitors values. Always use the recommended capacitive load as the resonant frequency of the ceramic resonators is very sensitive to capacitive load.

Note:

1. Few AVR devices may not come with internal capacitors. Some AVR devices may not have the CKOPT fuse, instead they have dedicated pins (TOSC1-TOSC2), to connect the 32.768kHz crystal.
2. Refer the device datasheet for specific details related to oscillator connections.

5.4. Unbalanced External Capacitors

In noisy environments the oscillator can be affected crucially due to the noise. If the noise is strong enough the oscillator can “lock up” and stop oscillating. To reduce the sensitivity of the oscillator to noise, the size of the capacitor at the high impedance input of the oscillator circuit, XTAL1 can be increased slightly. Increasing only one of the capacitors does not affect the total capacitive load much, but unbalanced capacitors can affect the resonant frequency to a higher degree than the change of the total capacitive load. However, unbalanced capacitive loads will affect the duty cycle of the oscillation and therefore one should in general not use unbalanced capacitive loads. This is especially critical if the Atmel AVR device is utilized close to its maximum speed limit.

5.5. RTC Crystals

Many AVR devices have the possibility to use asynchronous clocking of the built-in timer/counter. Using this feature, the counter can be used for real time functions. A 32.768kHz crystal should be connected to the TOSCx pins of the AVR.

In some AVRs the internal oscillator circuit used with the real time counter provides a capacitive load of approximately 20pF, which should be appropriate for common 32.768kHz crystals. Refer to the data sheet for the relevant device for information about the capacitors. If the internal load is insufficient for the applied crystal, external capacitors can be used .

5.6. PCB Layout

Finally, the physical location of the resonator with respect to the AVR is important. Ensure that the resonator is placed as close as possible to the AVR and shield the resonator by surrounding it with a ground plane

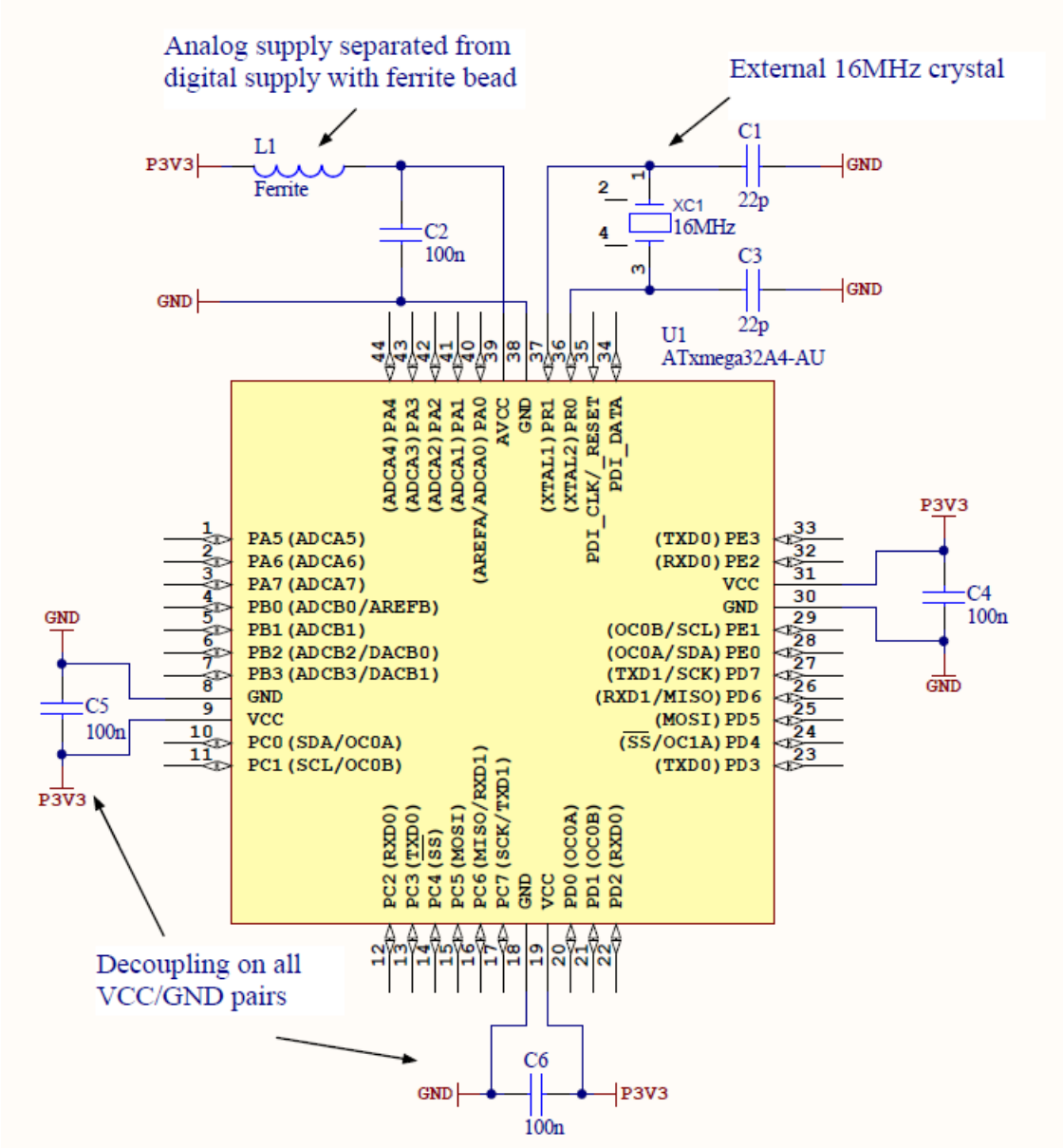
6. Example Layout of ATxmega32A4 and ATmega324PB device

The basic schematic recommended for bringing up a design using ATxmega32A and ATmega324PB devices are shown in the following figures. The key points to be considered are:

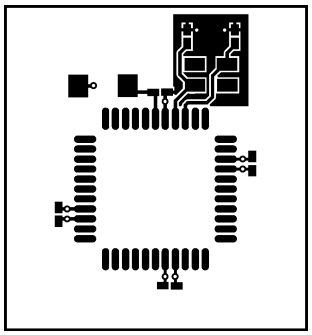
1. The connections for crystal oscillator and de-coupling capacitors.
2. The number of layers of PCB. It is recommended to have a multilayer design with supply and ground plane on separate layers.
3. Decoupling of all digital supply pairs from VCC and isolating A_{VCC} from V_{CC}.
4. Short distance between the crystal/capacitors and the MCU.
5. Ground plane surrounding the crystal and the vias connected to the planes are close to the MCU pins in the layout.

Figure 6-1. Example Layout for ATxmega32A4 device

(A) Basic Schematic of Required/Recommended Connections



(B) Copper PCB Layout of Required/Recommended Connections



(C) Top Silk Prints of Required/Recommended Connections

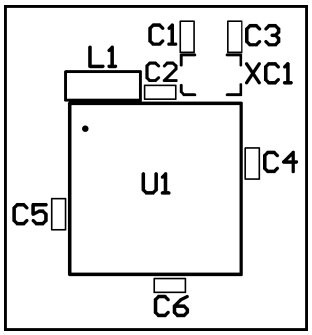
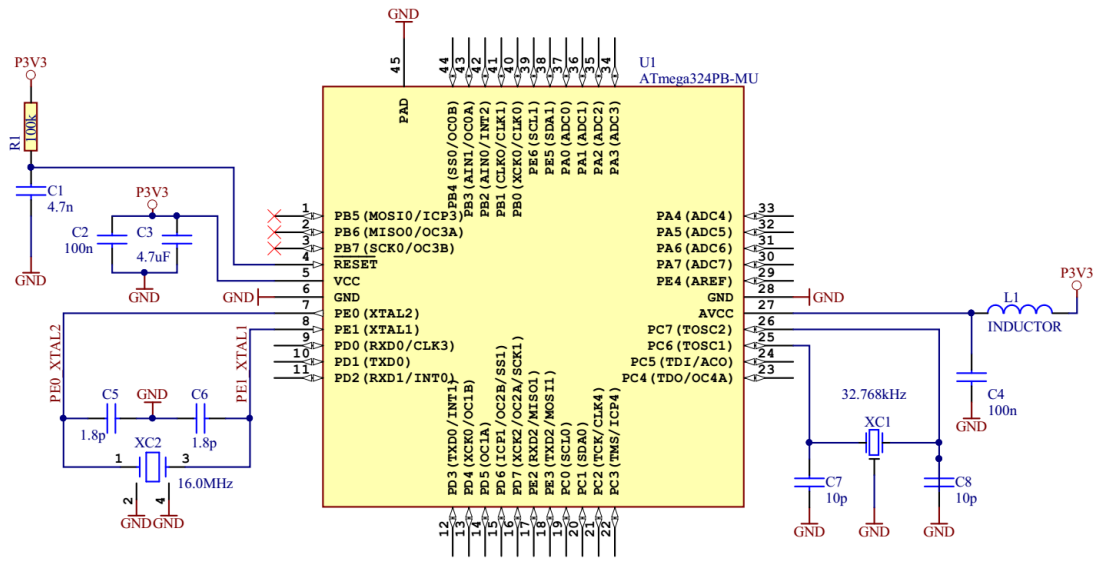
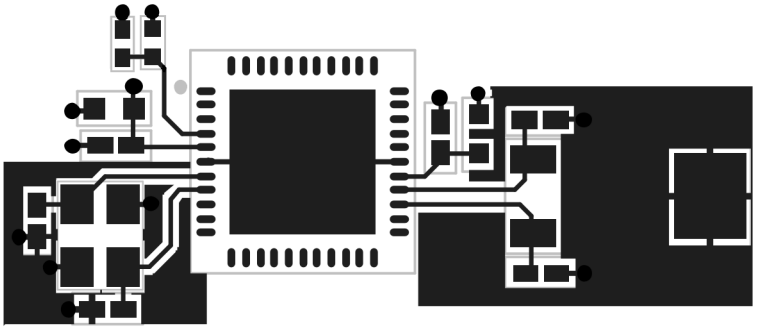


Figure 6-2. Example Layout for ATmega324PB device

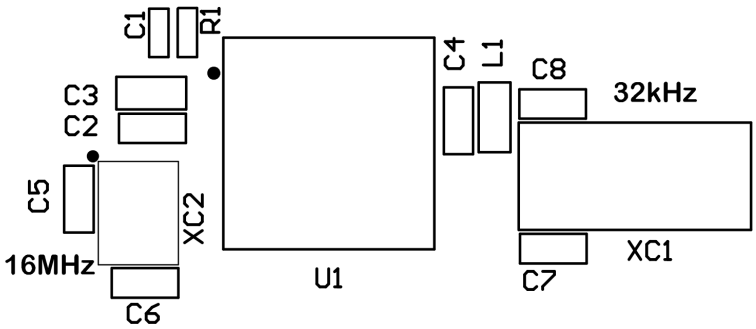
(A) Basic Schematic of Required/Recommended Connections



(B) Copper PCB Layout of Required/Recommended Connections



(C) Top Silk Prints of Required/Recommended Connections



7. Revision History

Doc Rev.	Date	Comments
2541Q	06/2016	1. General improvement of descriptions. 2. Added example layout for ATmega324PB device.
2541P	10/2015	Updated following sections: 1. About Crystals and Ceramic Resonators 2. Recommended Capacitor Values
2541O	09/2015	Corrected the figure Example Layout.
2521N	06/2015	Added Noise Implications.
2521M	09/2014	Fixed some typos in External RESET Switch.
2521L	07/2013	1. Updated Figure 4-5. 2. General improvements in regards of descriptions.

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**