# AVR042: AVR® Hardware Design Considerations

8-bit AVR® RISC Microcontroller

Application Note

## Features

- Providing Robust Supply Voltage, Digital and Analog
- Connecting the RESET Line
- SPI Interface for In-System Programming
- Using External Crystal or Ceramic Resonator Oscillators

## Introduction

This application note covers the most common problems encountered when switching to a new microcontroller architecture like the AVR. Solutions and considerations for the most common design-challenges are covered. The scope is to provide an introduction to potential design problems rather than being an exhaustive walk-through of how to design applications using the AVR microcontrollers. This document is thus a collection of information from existing Atmel AVR documents, combined with information that is not covered elsewhere.

It is highly recommended to read the application note “AVR040: EMC Design Considerations” before initiating a new design, specially if the aim of the design is a commercial application that needs to meet the requirements of the EMC directive (or similar directives in countries outside Europe). The application note is available from the Atmel web site, http://www.atmel.com.

Note that the measures described in this document are not required in all cases. In most cases, only a minimum of external components (decoupling capacitor, etc) is required. In fact, the embedded low-cost solutions such as the BOD and internal pull-ups will do the trick in many designs.

## Supply Voltage

Two aspects should be considered when designing the power supply for the discrete/digital elements of an AVR; ESD protection and reducing noise emission. Both these topics are treated in detail in the AVR040 application note, and only a short summary is included in this document.

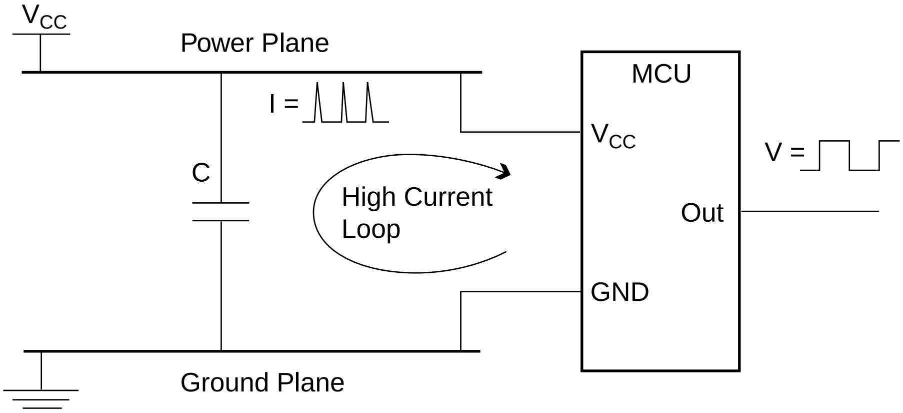
### Digital Supply Voltage

Looking at the data sheet for an AVR microcontroller, one can be fooled to believe that power supply is not critical. The device has a very wide voltage range, and draws only a few mA supply current. But as with all digital circuits, the supply current is an average value. The current is drawn in very short spikes on the clock edges, and if I/O lines are switching, the amplitude of the spikes increase. The current pulses on the power supply lines can be several hundred mA if all eight I/O lines of an I/O port changes value simultaneously. If the I/O lines are not loaded, the pulse will only last for a few nano seconds.

Rev. 2521B–AVR–01/04

This kind of current spike cannot be supplied over long power supply lines; The main source to draw the current from is (or should be) a decoupling capacitor.

**Figure 1.** Insufficient Decoupling[1]

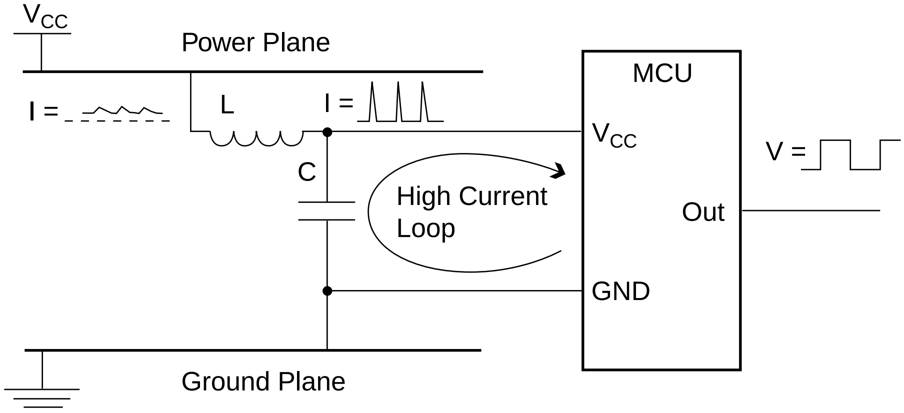


Note: 1. Decoupling Capacitor is to far away from the device.

Figure 1 shows an example of insufficient decoupling. The capacitor is placed too far away from the microcontroller, creating a large high current loop. The power and ground planes here are parts of the high current loop. As a result of this, noise is spread more easily to other devices on the board, and electro magnetic radiation from the board is increased. The whole ground plane will act as an antenna for the noise, instead of only the high current loop. This will be the case if 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. The same is often seen for boards with surface-mount components if the integrated circuits are placed on one side of the board and the decoupling capacitors are placed on the other.

Figure 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 radiate noise. Further, Figure 2 shows another improvement of the decoupling. A series inductor is inserted to reduce the switching noise on the power plane. The series resistance of the inductor must, of course, be low enough to ensure that there will be no significant DC voltage drop. The component values are depending on the application. Values for the inductor and the capacitor could be respectively 47 nH and 100 nF.

**Figure 2.** Decoupling with Capacitor and Series Inductor



The AVR devices which have power and ground lines placed close together (like the AT90S8535) can be better decoupled than devices with industry standard pinout (like the AT90S8515), where the power and ground pins are placed in opposite corners of the DIP package. This disadvantage can be overcome by using the TQFP package, which allows decoupling capacitors to be placed very close to the die. For devices with multiple pairs of power and ground pins, it is essential that every pair of pins get its own decoupling capacitor.

## Analog Supply Voltage

The AVR devices that have built-in ADC have a separate analog supply voltage pin, AVCC. This separate voltage supply is provided to make the analog circuits less prone to the digital noise originating from the switching of the digital circuits. The analog ground, AGND, is also separated from the digital ground – for the same reasons.

To be able to obtain good accuracy with the ADC the analog supply voltage must be decoupled separately, in the same manner as the digital supply voltage. The analog ground should be separated from the digital ground, so that the analog and digital ground are only connected at one point.

## Connecting the RESET Pin on AVRs

The RESET pin on the AVR is active LOW, and setting the pin LOW externally will thus result in a reset of the AVR. The RESET line serves two purposes:

1. To release all 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 also the PEN line can be used to enter Programming mode). Furthermore it is possible to enter High-voltage/Parallel Programming mode by pulling the RESET pin “very” high, where very high means 11.5 - 12.5V (refer to the data sheet of the device for more information).

The RESET line has an internal pull-up resistor of 100 - 500 kΩ. Theoretically, this ensures that the RESET line is not floating. In practice, the internal pull-up of 100 - 500 kΩ will prove insufficient if the surrounding environment is a bit noisy; Resetting of the AVR will occur if the RESET line is not actively driven high externally.

Different approaches can be used to connect the RESET pin so that noise will not cause unintended resetting of the AVR. An External Brown-out or supervisory circuit can be used to control the RESET pin. If an External Brown-out Circuit, like the ones described in application note AVR180, is applied one would not need to consider how to connect the RESET pin further. However, if the AVR device used in the application has built-in

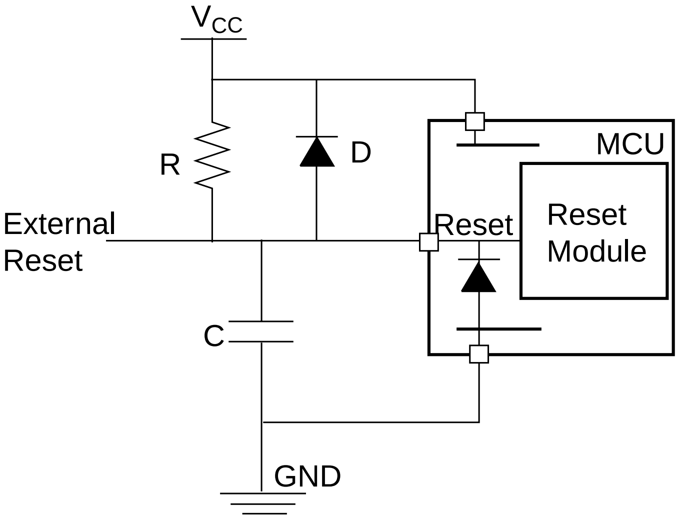
Brown-out Circuit, the External Brown-out can be eliminated and a more simple solution can be chosen to control the state of the RESET pin.

Connecting the RESET so that it is possible to enter both High-voltage Programming and ordinary Low Level Reset can be achieved by applying a pull-up resistor to the RESET line. This pull-up resistor makes sure that Reset does not go low unintended. The pull-up resistor can in theory be of any size, however there is no reason to use a bigger pull-up than the internal. If the AVR should be programmed from the STK500, the pull-up should not be so strong that the programmer cannot activate RESET by drawing the RESET line low. If STK500 is used for programming the recommended pull-up resistor is 4.7 kΩ - 10 kΩ.

Further, to protect the RESET line from noise, it is an advantage to connect a capacitor from the RESET pin to ground. This is not directly required since the AVR internally has a low-pass filter to eliminate spikes and noise that could cause a reset. Applying an extra capacitor is thus an additional protection. If not using High-voltage Programming it is recommended to insert an ESD protecting diode from RESET to $V_{CC}$, since this is not internally provided due to High-voltage Programming. The components should be located physically close to the RESET pin of the AVR. Figure 3 illustrates the suggested circuit on the RESET line. Component values given in the figures are suggested values rather that recommended values. It is possible to choose different component values. To choose the optimal component values one should have information about the nature of the noise that the system will be exposed to.

If no reset functionality is nedded, and no In-System Programming is used, the reset line can be connected directly to $V_{CC}$.

**Figure 3.** Recommended Reset Pin Connection[1]



Note: 1. Typical values are:
R = 4.7 kΩ
C = 10 nF
D = 1N4148

## Connecting the In-System Programmin Lines

The In-System Programming lines can be used for programming the Flash, EEPROM, Lock bits, and most Fuse bits in all AVRs[1]. This Three-wire interface and reset can be used to make the AVR In-System Programmable, from this the abbreviation ISP. This feature makes it possible to program the AVR on the last stage of production of a target application board, reprogram if software bugs are identified late in the process or even update the AVR in the field if required. It is therefore highly recommended to always design the target application board so that the ISP connectors can be accessed easily.

Note: 1. Except the ATtiny10, ATtiny11 and ATtiny28.

On most devices the ISP interface is located on the same pins as the SPI interface. However, there are exceptions from this rule, e.g. the ATmega103 and the ATmega128 do not have the ISP lines on the SPI lines. In this case the UART lines are used for the ISP interface. Common for the ISP interfaces on all devices is that the AVR acts as a SPI Slave during access through the ISP interface. Consult the data sheet of the device to find out which lines are used for ISP Programming.

Be aware that the In-System Programming interface can be disabled through In-System Programming. By clearing (logic “0”) the SPI Enable (SPIEN) Fuse the SPI interface is disabled - and therby In-System Programming is no longer possible. If a device is mounted in a target board and the SPIEN Fuse is cleared no further In-System Programming is possible. The only way to regain the possibility to ISP program the device is by unsoldering the device and change the SPIEN Fuse using Parallel Programming. This can be done e.g., using the STK500.

Some devices further have the possibility to disable the RESET pin by clearing the RSTDISBL Fuse. When the RESET pin is disabled it can be used as an IO pin by the application. To be able to enter Programming mode when the RESET pin is disabled the RESET pin has to be pulled “very high” – to 12V. If the RESET pin is disabled in the application the designer has to consider that the application should tolerate 12V on the RESET pin if ISP Programming should be possible. See the data sheet for details on the RESET Disable Fuse and High-voltage ISP Programming.

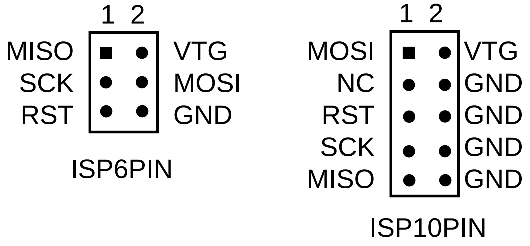
Some fuses cannot be changed through ISP Programming. If fuses cannot be changed through ISP Programming, Parallel Programming is required to alter the fuses. An overview of devices that have fuses that are not accessible through ISP Programming are found in Table 1.

**Table 1.** Devices with Fuses that are not Accessible Through ISP Programming

| Device | Fuses Not Programmable by ISP |
|---|---|
| AT90LS/S1200 | SPIEN and RCEN |
| AT90LS/S2313 | SPIEN and FSTRT |
| AT908515 | SPIEN and FSTRT |

Two standard connectors are provided by the Atmel ISP programmers: A 6-pin and a 10-pin connector. These are shown in Figure 4. In addition to the data lines (MOSI and MISO) and the bus clock (SCK), target voltage (VTG), ground (GND), and RESET (RST) are provided through these connectors.

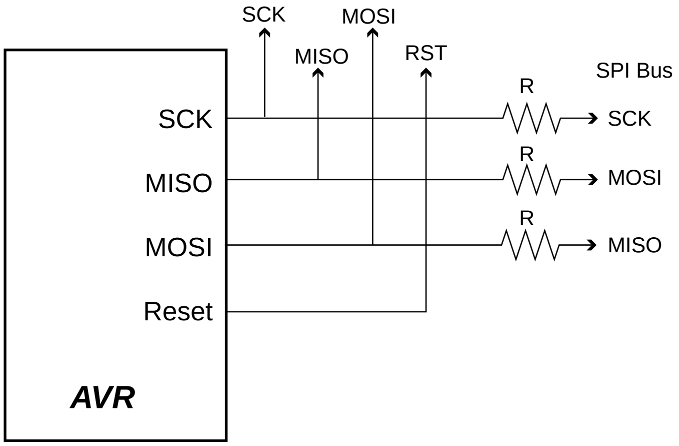
**Figure 4.** Standard ISP Connectors on STK500, AVR ISP and STK200/STK300 Programmers



Some ISP programmers are supplied by the target power supply. In this way they easily adapt to the correct voltage level of the target board. Other ISP programmers, like STK500, can power the target board via the VTG line. In that case it is important that the power supply on the target is not switched on. Read the user guide of the ISP Programmer to find out what capabilities your programmer has and what kind of physical interface it has.

Since the ISP-lines can be used as general purpose I/O ports there is a possibility for a conflict on the lines during In-System Programming. In case other devices than the AVR are driving the SPI lines the programmer must be protected from the device driving the lines. Applying series resistors on the SPI lines, as depicted in Figure 5, is a simple way of achieving this.

**Figure 5.** Connecting the ISP Interface[1]



Note: 1. Typical value is: R = 4.7 kΩ

There will never occur drive contention of the ISP lines when entering programming mode. The programmer will draw the RESET line low resulting in the AVR floating all lines. The AVR will however listen to the MOSI line, waiting for an “enter Programming mode” instruction. While in Programming mode the AVR will only drive the MISO. The MISO line is configured as input to the programmer and the programmer will therefore not drive this line actively. The programmer controls all lines, except for the MISO line.

Multiple AVRs in a single application can share the same ISP interface to make it possible to program all of the devices through a minimal interface. However, the AVR devices will all respond to the ISP instructions if special design considerations are not made. If it is desired to have only one ISP interface on the target board, the ISP Programming can be designed so that only one of the AVR devices is provided with an SPI clock at a time.

All other SPI lines can then be shared. In this way all AVRs can be located behind the same protection resistors, since they are all held in RESET while the ISP RESET line is activated. The gating of the ISP clock can be accomplished, using jumpers. Alternatively, a solution is to have multiple ISP interfaces, one for each device, all protected as shown in Figure 5.

## Using Crystal or Ceramic Resonators

Most AVR MCUs can use different clock sources. The optional external clock sources are Digital Clock, RC Oscillator, Crystal or Ceramic Resonator.

The choice of Oscillator type used in an application should depend on the accuracy required. If the application uses an asynchronous communication interface, e.g., the UART, requirement to the accuracy of the system clock could make it necessary to choose a ceramic resonator or even a crystal. In many applications the use of a high accuracy system clock is not required and in these applications an external or internal RC Oscillator could be used to reduce the total system cost.

The use of crystals and ceramic resonators are in some designs causing problems due to the fact that the use of these clock sources are not well understood. This section discusses using crystals and ceramic resonators with 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.

This document is related to devices produced in 0.5 µm process. The devices covered by this document are those listed in Table 2. For devices not listed in the table, refer to the data sheet of the device.

## Selecting the Clock Source in the AVR

The clock source used by the AVR is selected through the fuses. Most ISP and parallel programmers can program the fuses that are related to selecting a clock source. The fuses are not erased when the AVR memory is erased and the fuses therefore only need to be programmed if the fuse settings should be altered. Programming the fuses each time the device is erased and reprogrammed is 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 available can vary between AVR devices – an overview of the clock options of the different devices are seen from Table 2. For information about devices not present in the table refer to the data sheet for the relevant device to determine the clock options.

**Table 2.** Overview of Clock Options Available on AVR Devices

| Device | Int. RC (MHz) | Ext. RC | Low-freq. Xtal (kHz) | Xtal (kHz) | Ceramic Resonator (kHz) | Digital Clock |
|---|---|---|---|---|---|---|
| ATtiny11 | 1.0 | Yes | 32.768 | >450 | >450 | Yes |
| ATtiny12 | 1.2 | Yes | 32.768 | >450 | >450 | Yes |
| ATtiny15 | 1.6 | – | – | – | – | – |
| ATtiny28 | 1.2 | Yes | 32.768 | >450 | >450 | Yes |
| AT90S1200 | 1.0 | – | – | >450 | >450 | Yes |

Table 2. Overview of Clock Options Available on AVR Devices

| Device | Int. RC (MHz) | Ext. RC | Low-freq. Xtal (kHz) | Xtal (kHz) | Ceramic Resonator (kHz) | Digital Clock |
|---|---|---|---|---|---|---|
| AT90S2313 | – | – | – | >450 | >450 | Yes |
| AT09S2323 | – | – | – | >450 | >450 | Yes |
| AT90S2343 | 1.0 | – | – | – | – | Yes |
| AT90S4433 | – | – | – | >450 | >450 | Yes |
| AT90S8515 | – | – | – | >450 | >450 | Yes |
| AT90S8535 | – | – | – | >450 | >450 | Yes |
| ATmega161 | – | – | – | >450 | >450 | Yes |
| ATmega163 | 1.0 | Yes | 32.768 | >450 | >450 | Yes |
| ATmega323 | 1.0 | Yes | 32.768 | >450 | >450 | Yes |
| ATmega103 | – | – | – | >450 | >450 | Yes |

Be aware that if selecting a different clock source than what is actually applied, the AVR may not run. Different oscillator circuits are activated internally in the AVR dependent on the selected clock option. Since the fuses are not cleared by a memory erase, it can cause problems if incorrect settings are chosen.

## General Information about Crystals and Ceramic Resonators

The typical type of crystal used for the AVR is the AT-cut parallel resonant crystal. The behavior of a ceramic resonator is very similar to an AT-cut parallel resonant crystal, but is more like 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 oscillation frequency of the ceramic resonator can more easily be tuned to a desired frequency, but is also more sensitive to temperature and load changes, causing undesired frequency variations. The biggest advantage of the ceramic resonator is that it has a faster start-up than crystals.

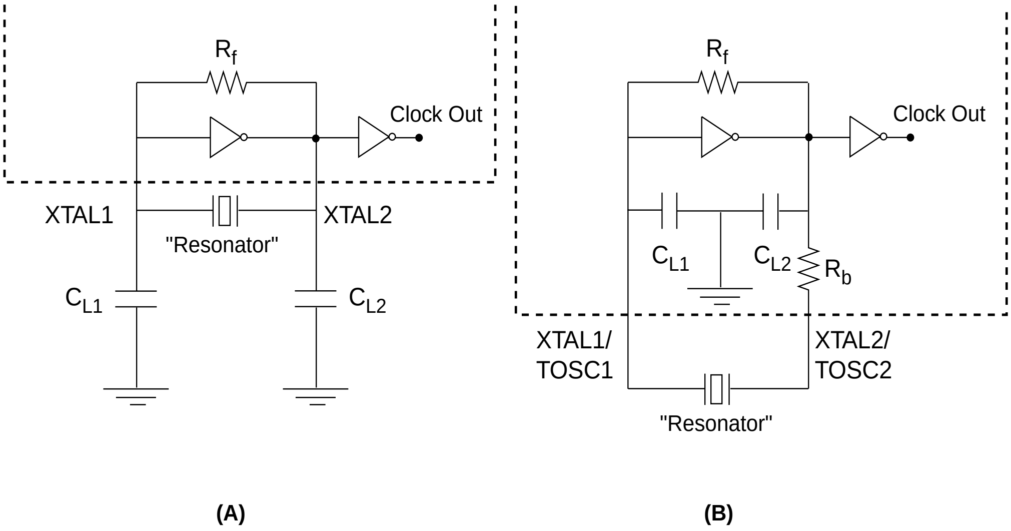
In general there will be no distinction between crystals and ceramic resonators in this section and the term “resonator” will thus refer to both devices.

Table 3. Differences between Ceramic Resonators and Crystal Resonators

|  | Ceramic Resonator | Crystal Resonators |
|---|---|---|
| Aging | ± 3000 ppm | ± 10 ppm |
| Frequency Tolerance | ± 2000 - 5000 ppm | ± 20 ppm |
| Frequency Temperature Characteristics | ± 20 - 50 ppm/°C | ± 0.5 ppm/°C |
| Frequency Pullability | ± 100 - 350 ppm/pF | ± 15 ppm/pF |
| Oscillator Rise Time | 0.01 - 0.5 ms | 1 - 10 ms |
| Quality Factor (Qm) | 100 - 5000 | $10^3 - 5 \cdot 10^5$ |

Basic Oscillator circuits used for parallel resonators are illustrated in Figure 6. The part of the circuit above the dashed line, represent the oscillator circuit internally in the AVR.

**Figure 6.** Basic Inverter Circuits Equivalent to the Oscillator Circuits in the AVR[1]



Note: 1. (A) Represents the Oscillator Circuit for Crystals and Ceramic Resonators Faster than 400 kHz, while (B) is for Low Frequency Crystals (32.768 kHz).

Simplifying the description of the AVR's built-in Oscillator circuits, they can be understood as the inverter based Oscillator circuits illustrated in Figure 6. The circuit used with resonator frequencies above 400 kHz is depicted in Figure 6 (A). When using this circuit, a capacitive load must be applied externally. The oscillator circuit seen in Figure 6 (B) is the circuit used for low frequency crystals. It is optimized for 32.768 kHz crystals. This circuit provides the capacitive load required by the crystal internally and further adds the resistor $R_b$ to bias the crystal and limit the drive current into the crystal. The resistor $R_f$ is, approximately 1 MΩ, and provides a feedback to bias the inverter to operate it in its linear region.

When using resonators with the AVR, it is necessary to apply (external) capacitors according to the requirements of the chosen resonator. A parallel resonator will not be able to oscillate in a stable manner if the capacitive load applied is insufficient. If the capacitive load is too high the oscillation may have problems starting due to drive level dependency of the load. The solution is 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 Equation1.

$$C_L = \frac{C_{L1} \cdot C_{L2}}{C_{L1} + C_{L2}} + C_S$$

Equation1

Where $C_{L1}$ and $C_{L2}$ refers to the external capacitors seen in Figure 6 and $C_S$ is the combined capacitive load of the XTAL pins of the AVR and stray capacitances of the PCB.

$C_S$ can be estimated to be 5 - 10 pF. If $C_{L1} = C_{L2}$ then the external capacitors can be determined by Equation 2.

$$C_{L1} = C_{L2} = 2(C_L - C_S) \tag{Equation2}$$

If one should find an appropriate capacitive load for the crystal where the recommended capacitive load is 16 pF, the value of the external capacitance could be estimated like shown in Equation 3.

$$C_{L1} = C_{L2} = 2(16pF - 5pF) = 22pF \tag{Equation3}$$

A reasonable choice would therefore be to use 22 pF capacitors connected to GND as shown in Figure 6(A). Do not choose a lower value, rather a higher value since this gives a more reliable and stable oscillation. However, note that the power consumption of the Oscillator increases with increasing capacitive load.

## Recommended Capacitor Values

The recommendations here will work well in most applications, but there is no way to provide general values for the external capacitors that can be guarantee to work with all resonators.

When using the clock option "External Crystal Oscillator", crystals with a nominal frequency from 400 kHz and up can be used. For these standard high frequency crystals the recommended capacitor value is in the range 22 - 33 pF.

The clock option "External Low-frequency Crystal" is intended for 32.768 kHz crystals. When selecting this clock source the internal oscillator circuit provides the required capacitive load. The value of the internal capacitors is typically 20 pF, but can vary. If using a 32.768 kHz crystal that does not require more load than this, external capacitors can be left out. Otherwise external capacitive load must be added.

Using the clock option that selects "External Ceramic Resonator" it is strongly recommended to consult the data sheet to determine what capacitors to apply. Always use the capacitive load recommended there since the resonant frequency of the ceramic resonators is very sensitive to capacitive load.

## Pullability of the Nominal Frequency

The frequency of the resonator is depending on the capacitive load that is applied. Applying the capacitive load specified in the data sheet of the resonator will provide a frequency very close to the nominal frequency (intended oscillating frequency). If other capacitive loads are applied the oscillating frequency will change. The frequency will increase if the capacitive load is decreased and decrease if the load is increased. The frequency pullability – how far from the nominal frequency the resonant frequency can be forced by applying load – is dependent on the type of resonator used or actually the Q-factor of the resonator. Typically crystals have a very high Q-factor, meaning that the pullability of the resonant frequency is relatively low. Some crystals have especially high Q-factors, e.g., 32.768 kHz crystals, since they are typically used for timing purposes, which require that the frequency is very exact. The high Q-factor gives high frequency stability. Ceramic resonators do not have high Q-factors and are therefore more sensitive to changes in capacitive load.

Regardless of the resonator type the resonant frequency can be pulled by changing the capacitive load. If the resonator is overloaded, the oscillation can have problems starting or have a prolonged start-up time, but once the oscillation has started it will seldom stop again due to overload.

To be able to determine the pullability of the resonator the motional capacitance of the resonator should be known. This value is not listed in all crystal data sheet and it is recommended to not intentionally try to pull the resonant frequency of the resonator unless understanding the theory behind this.

## Unbalanced External Capacitors

In noisy environments the Oscillator can be affected crucially by the noise. If the noise is strong enough the oscillator can lock up and stop oscillating. To make the Oscillator less sensitive 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 use unbalanced capacitive loads with caution.

Ensure that neither of the half-periods of the clock signal are shorter than half a clock period of the fastest clock specified for the device. In other words, if an 8 MHz device is used neither of the half-periods should be shorter than (1/8 MHz)/2 = 62.5 ns. If running the device at its maximum speed it is recommended to have 50% duty cycle, otherwise the “real” frequency of the clock is higher that recommended in the specification. Running at speeds lower than the maximum recommended speed allow for some deviation from a 50% duty cycle.

## RTC Crystals

Some AVR devices have the possibility use asynchronous clocking of the built-in Timer/Counter. The Counter can be used for real time functions through this feature. A 32.768 kHz crystal should then be connected to the TOSCx pins of the AVR.

The internal Oscillator circuit used with the Real Time Counter provides a capacitive load of approximately 20 pF, which should be appropriate for common 32.768 kHz crystals. Note that the internal capacitors can vary between 10 pF and 30 pF from device to device. The oscillation frequency of the asynchronous RTC Oscillator will therefore vary slightly from device to device due to the variation in capacitive load. External capacitors can be applied if the load is insufficient for the applied crystal.

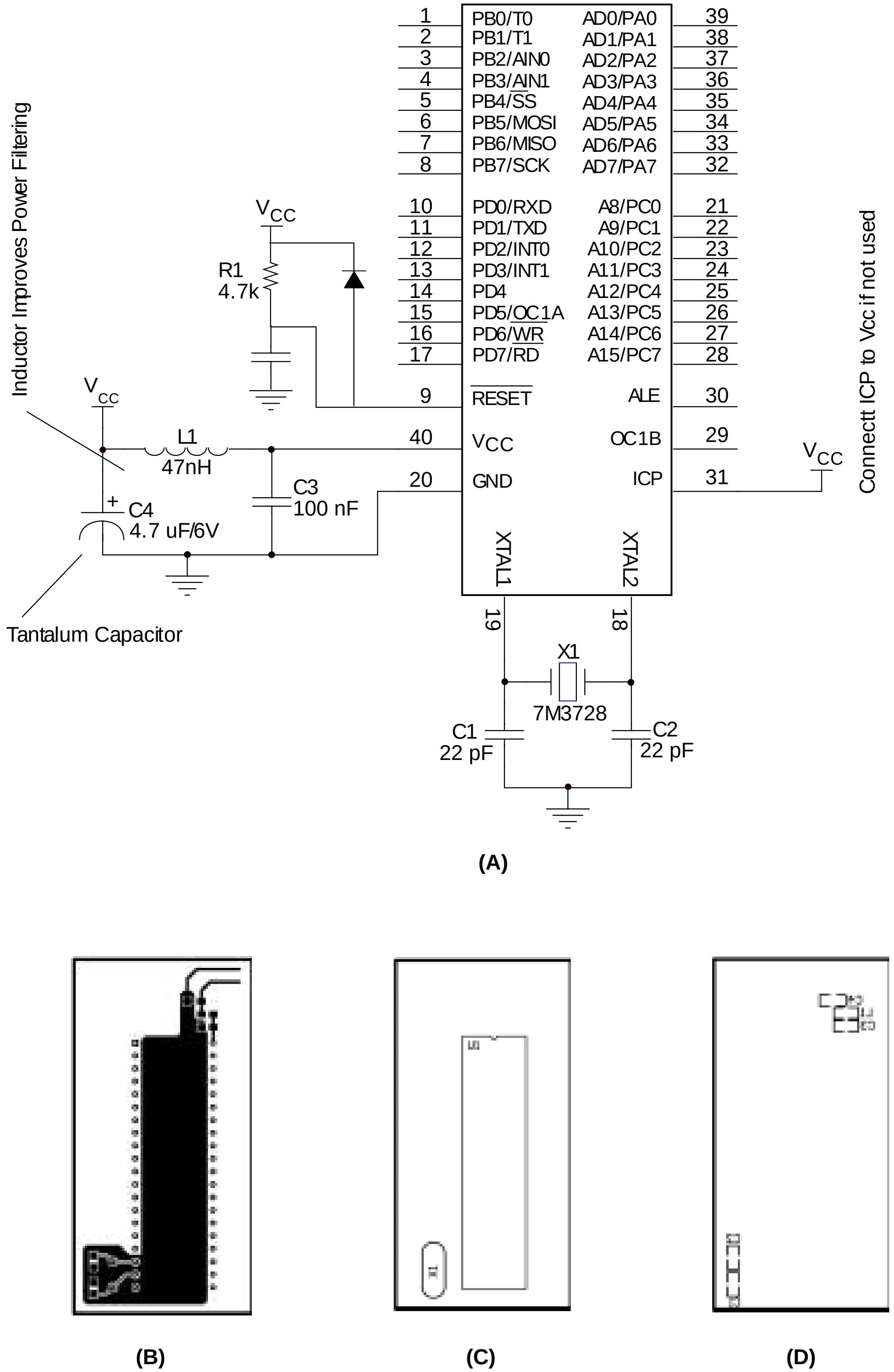
The RTC Oscillator is designed to work with a crystal and is therefore designed to work on a sine wave signal type. [If a digital clock is fed to the RTC Oscillator, incorrect behavior like double clocking of the timer may occur.] It is therefore recommended to use a 32 kHz crystal to drive the RTC.

## PCB Layout

Finally, the importance of the physical location of the resonator in relation to the AVR should be stressed. Always place the resonator as close to the AVR as possible and shield the resonator by surrounding it with a ground plane.

From Figure 7 (A-D) a schematic and PCB layout using a crystal Oscillator for AT90S8515 is seen. Note the ground plane surrounding the crystal and the very short distance between the crystal and the AT90S8515.

**Figure 7.** (A) Basic Schematic of Required/Recommended Connections for AT90S8515. (B) Copper PCB Layout. (C) and (D) Top and Bottom Silk Screen



(A)

(B)

(C)

(D)

## Atmel Corporation

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

## Regional Headquarters

### *Europe*

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

### *Asia*

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

### *Japan*

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

## Atmel Operations

### *Memory*

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

### *Microcontrollers*

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

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

### *ASIC/ASSP/Smart Cards*

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

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

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

### *RF/Automotive*

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

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

### *Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom*

Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

### *Literature Requests*

www.atmel.com/literature

**Disclaimer:** Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Printed on recycled paper.