Introduction

Authors: Eivind Berntsen, Per Andreas Gulbrandsen, Microchip Technology Inc.

Embedded timekeeping applications often use a 32.768 kHz external crystal oscillator as the clock source, due to the higher accuracy of these, compared to internal oscillators. The accuracy is, however, adversely affected by several factors. This document looks at PCB design, aging and drift over temperature. In addition, crystal manufacturing tolerance will be briefly explained.

For highly accurate timekeeping, the accuracy of the oscillator can be improved. This is done by measuring the error of the oscillator and compensating for it. Frequency measurement can be done using an external reference and internal peripherals of tinyAVR 1-series and megaAVR 0-series, or it can be done using an external component. Both approaches are outlined in this document.

When the error is known, software based compensation can be implemented to reduce the error. A few examples are shown, with some useful techniques for efficient implementation of software based compensation.

Features

Introduction and Measurement:
• A brief introduction to contributing factors of crystal frequency inaccuracy
• Explanation of how to measure the frequency of the 32.768 kHz crystal clock on the tinyAVR 1-Series and megaAVR 0-Series

Examples:
• An example showing how the RTC can compensate for the static crystal error
• An example showing how the RTC, internal temperature sensor, and the ADC can be used to compensate for static crystal error and error due to temperature drift
• Four examples available in Atmel | START, two examples on frequency measurement and two examples showing compensation algorithms
# Table of Contents

Introduction......................................................................................................................1

Features..........................................................................................................................1

1. Relevant Devices.......................................................................................................3
   1.1. tinyAVR® 1-series..........................................................................................3
   1.2. megaAVR® 0-series.......................................................................................3

2. Sources of Inaccuracy and Drift............................................................................ 5
   2.1. Capacitive Load Mismatch.......................................................................... 5
   2.2. Crystal Manufacturing Tolerance.................................................................6
   2.3. 32.768 kHz Crystal Oscillator Temperature Drift......................................6
   2.4. 32.768 kHz Crystal Oscillator Aging Drift..................................................7

3. Frequency Measurement......................................................................................... 9
   3.1. Start-up Time..................................................................................................9
   3.2. External Frequency Measurement.................................................................9
   3.3. Internal Clock Measurement........................................................................11

4. RTC Compensation.................................................................................................. 13
   4.1. Compensation for Crystal and Load Capacitance Deviation........................ 13
   4.2. Compensation for Temperature Drift.............................................................13
   4.3. Crystal Compensation Look-Up Table Example..........................................13

5. Get Source Code from Atmel | START....................................................................15

6. Revision History......................................................................................................16

The Microchip Web Site..............................................................................................17

Customer Change Notification Service.................................................................17

Customer Support.....................................................................................................17

Microchip Devices Code Protection Feature.........................................................17

Legal Notice...............................................................................................................18

Trademarks.................................................................................................................18

Quality Management System Certified by DNV......................................................19

Worldwide Sales and Service....................................................................................20
1. **Relevant Devices**

This chapter lists the relevant devices for this document.

1.1 **tinyAVR® 1-series**

The figure below shows the tinyAVR® 1-series devices, laying out pin count variants and memory sizes:

- Vertical migration upwards is possible without code modification, as these devices are pin compatible and provide the same or more features. Downward migration may require code modification due to fewer available instances of some peripherals.
- Horizontal migration to the left reduces the pin count and therefore, the available features.

![Figure 1-1. tinyAVR® 1-series Overview](image)

Devices with different Flash memory size typically also have different SRAM and EEPROM.

1.2 **megaAVR® 0-series**

The figure below shows the megaAVR® 0-series devices, laying out pin count variants and memory sizes:

- Vertical migration is possible without code modification, as these devices are fully pin and feature compatible.
- Horizontal migration to the left reduces the pin count and therefore the available features.
Figure 1-2. megaAVR® 0-series Overview

Devices with different Flash memory size typically also have different SRAM and EEPROM.
2. **Sources of Inaccuracy and Drift**

32.768 kHz crystal oscillators have higher precision than RC oscillators. However, crystal oscillators are also subject to error in frequency. This includes static error, which is primarily due to PCB design and manufacturing tolerance. Frequency error is also impacted by drift over temperature, meaning that the output frequency will change as the operating temperature changes.

2.1 **Capacitive Load Mismatch**

Crystal manufacturers specify an external capacitive load to apply to the crystal. Mismatching the capacitive load will result in lower accuracy of the crystal, or even failure to oscillate. A typical characteristic curve of frequency versus load capacitance, as can be found in a crystal oscillator data sheet, is shown in **Figure 2-1**.

**Figure 2-1. Frequency vs. Load Capacitance**

![Frequency vs. Load Capacitance](image)

**Figure 2-2** show schematics for a typical 32.768 kHz crystal circuit. The crystal is connected to the TOSC1 and TOSC2 pins of the MCU. C1 and C2 are the capacitors used to balance the load capacitance.

**Figure 2-2. 32.768 kHz Crystal Circuit**

![32.768 kHz Crystal Circuit](image)

Multiple sources contribute to the load capacitance in a crystal circuit:
• The PCB parasitic capacitance, which is capacitance between PCB traces.
• The MCU parasitic capacitance. Refer to the electrical characteristics of the device data sheet for further details.
• The mounted capacitors C1 and C2, which are placed between the TOSC1 pin of the MCU and GND, and the TOSC2 pin of the MCU and GND, respectively.

When calculating the value of C1 and C2, the MCU parasitic capacitance and the PCB parasitic capacitance must be taken into account. PCB parasitic capacitance is dependent on the PCB design. For example, the choice of track length and width will impact the capacitance. For more information on best practices on this, refer to AVR4100: Selecting and testing 32.768 kHz crystal oscillators for Atmel AVR microcontrollers.

All components are subject to manufacturing tolerance. This means that correctly selecting C1 and C2 will not ensure an identical accuracy for all units produced. Because of this, software-based compensation must be performed to achieve optimal precision in timekeeping.

2.2 Crystal Manufacturing Tolerance
When properly loaded, the accuracy of a 32.768 kHz crystal oscillators is typically in the ±20 ppm range. An error of one ppm equals an error of 30.5s per year. An error of 20 ppm equals an error of approximately 10 minutes per year. Typically, what contributes to the manufacturing tolerance is the precision of the crystal cut and the purity of the crystal.

2.3 32.768 kHz Crystal Oscillator Temperature Drift
Low-cost 32.768 kHz crystal oscillators typically have a parabolic frequency dependency over temperature, as shown in Figure 2-3, for which the crystal manufacturers specify:

• A temperature coefficient (B) in ppm/°C²
• A turnover temperature (T0) in °C
The temperature coefficient $B$ is negative, meaning that the crystal oscillator slows down at cold and hot temperatures. The temperature coefficient has a tolerance; typically this is ±15%. The turnover temperature, $T_0$, is typically 25°C, with a tolerance typically in the range ±5°C. The frequency drift at temperature $T$ is given by:

$$\Delta f / F_0 \text{ (ppm)} = B \cdot (T - T_0)^2$$

As an example, assume $B = -0.035 \text{ ppm/°C} \pm 15\%$, $T_0 = 25\text{°C} \pm 5\text{°C}$, and $T = 85\text{°C}$. This leads to a typical drift of:

$$\Delta f / F_0 = 126 \text{ppm} \pm 40 \text{ppm}$$

### 2.4 32.768 kHz Crystal Oscillator Aging Drift

Most crystal manufacturers specify the aging of the crystal after one year of operation. A typical range is ±3ppm for the first year. This specification means that the drift of a crystal is unpredictable in direction and magnitude. Although most frequency drift will happen in the first year, some drift will also occur after the first year. Some of the contributing factors in frequency drift is oxidation, and residue inside the encasement attaching to the crystal. This has the effect of changing the total mass of the crystal and thus the resonant frequency. For a more in-depth explanation on this, refer to [https://en.wikipedia.org/wiki/Crystal_oscillator#Stability_and_aging](https://en.wikipedia.org/wiki/Crystal_oscillator#Stability_and_aging).

Load capacitance can change over the years. If it does, it will cause frequency drift, but due to the magnitude of this change and due to the change being symmetric between the crystal pins, this effect is negligible. PCB parasitic capacitance aging, as well as aging of MCU parasitic capacitance, is virtually non-existent. The aging of NP0/C0G type capacitors is less than ±0.1 % for the whole life of the capacitor,
i.e. less than 10 fF for a 10 pF capacitor. Considering ppm error, changes in load capacitors give less than 0.1 ppm of drift.

Although PCB parasitic capacitance in general does not age much, adding silicon or urethane on the PCB to add vibration or moisture resistance can slightly alter the capacitance of the PCB tracks as these materials has a different dielectric constant compared to air.
3. **Frequency Measurement**

It is necessary to know the magnitude of the frequency error to be able to compensate for it. Therefore, the frequency of the crystal must be measured. The measurement must be performed when the crystal and device are mounted on the PCB.

Multiple measurements over a range of temperatures will typically give better results, compared to relying on the manufacturer's data.

The frequency of the 32.768 kHz crystal oscillator can be measured using an external measurement tool, or by using an internal resource. When using an external tool, this tool will take the frequency of the crystal as input. When using an internal resource for measurement, for example a timer/counter, a high-precision reference must be supplied. The precision of the measurement will never be higher than the precision of the reference, as the reference is what the 32.768 kHz crystal frequency is compared against.

In both cases, it is important that the crystal is given time to start up and settle before the frequency is measured.

3.1 **Start-up Time**

All crystals have a start-up time, which is the time it takes for the crystal output to become stable. The start-up time of a given crystal is specified in the crystal data sheet. There are multiple choices on the tinyAVR 1-series and megaAVR 0-series for setting the start-up time, with the maximum value being 65535. This means that the device counts 65536 cycles with a frequency of 32.768 kHz, which translates to a start-up time of two seconds. When the start-up time has expired, the XOSC32K Status bit in the Main Clock Status register (MCLKSTATUS.XOSC32KS) will be set.

**Note:** The 32.768 kHz crystal will not automatically be started when the Enable bit in the 32.768 kHz Crystal Oscillator Control A register (XOSC32KCTRLA.ENABLE) is written to ‘1’. It will only be started when the 32.768 kHz clock is requested by a peripheral, for example the main clock or RTC, or if the Run Standby bit in XOSC32KCTRLA (XOSC32KCTRLA.RUNSTDBY) is set.

**Note:** MCLKSTATUS.XOSC32KS will not be set unless the 32.768 kHz clock is requested by a peripheral. This is also true when XOSC32KCTRLA.RUNSTDBY is set.

3.2 **External Frequency Measurement**

When using an external measurement tool to do frequency measurement, the 32.768 kHz crystal oscillator output must be available on a pin. This can be done either using an Event from the RTC or PIT, and configuring a pin as a user, or by using the 32.768 kHz crystal as the main clock.

The RTC, when enabled, will generate the following output events:

- **Overflow (OVF):** Generated when the counter has reached its top value and wrapped to zero. The generated strobe is synchronous with CLK_RTC and lasts one CLK_RTC cycle.
- **Compare (CMP):** Indicates a match between the counter value and the Compare register. The generated strobe is synchronous with CLK_RTC and lasts one CLK_RTC cycle.

When enabled, the PIT generates the following 50% duty cycle clock signals on its event outputs:

- **Event 0:** Clock period = 8192 RTC clock cycles
- **Event 1:** Clock period = 4096 RTC clock cycles
- Event 2: Clock period = 2048 RTC clock cycles
- Event 3: Clock period = 1024 RTC clock cycles
- Event 4: Clock period = 512 RTC clock cycles
- Event 5: Clock period = 256 RTC clock cycles
- Event 6: Clock period = 128 RTC clock cycles
- Event 7: Clock period = 64 RTC clock cycles

The event users are configured by the Event System (EV SYS).

When using the 32.768 kHz crystal as the main clock source, there are several ways of routing the clock out of the pin.

- Routing the clock directly to the CLKOUT pin.
- Using TCA, TCB, TCD or SPI to output the main clock frequency.

It is not recommended to attach a probe directly to the crystal circuit, to measure the frequency. This will change the characteristics of the circuit and alter the crystal frequency.

The following two examples shows how to output the 32.768 kHz crystal to a pin. One example shows how to use the system clock, and one shows how to use the RTC and Event System.

### 3.2.1 Using the Crystal Clock as System Clock

This example shows how to configure the 32.768 kHz crystal to be the system clock and how the system clock can be set as an output to the CLKOUT pin. Refer to 5. Get Source Code from Atmel | START for a link to the project.

As the 32.768 kHz clock will be clocking the CPU, it is recommended to:

1. Enable the crystal with Run Standby as one of the first actions performed in main().
2. If there are any, run other production tests and calibrations of the circuit.
3. Switch the main clock to the crystal.
4. Configure the system clock as output on the CLKOUT pin.

This procedure will allow time for the crystal to settle while other production tests are being performed.

In the below example code the crystal is started by the first protected write. Note that the Run Standby bit is written at the same time. This is to ensure that the crystal starts even when no peripheral or the CPU is requesting the clock. The main clock is switched before checking that the crystal is stable as the XOSC32KS bit will not be set unless something is requesting the clock.

```c
//Deactivate change protection for register Start XOSC32K here
//start-up time is set to the maximum 64K cycles (~2 seconds)
_PROTECTED_WRITE(CLKCTRL_XOSC32KCTRLA , CLKCTRL_ENABLE_bm |
   CLKCTRL_CSUT_64K_gc | CLKCTRL_RUNSTDBY_bm);

//Deactivate change protection for register, Switch main clock
//to the 32K oscillator and output the clock to the pin
_PROTECTED_WRITE(CLKCTRL_MCLKCTRLA , CLKCTRL_MCLKCTRLA =
   (CLKCTRL_CKSEL1_bm) | CLKCTRL_CLKOUT_bm);

//Wait for XOSC32K to be reported as stable
while(!(CLKCTRL_MCLKSTATUS & CLKCTRL_XOSC32KS_bm));

//Deactivate change protection for register, Make certain the prescaler is disabled.
_PROTECTED_WRITE(CLKCTRL_MCLKCTRLB, 0);
```

In the case of outputting the 32.768 kHz clock, it is important to consider the capacitive load on the output pin. A high load can make it difficult to get a good signal out on the test pin. Also, the clock signal must...
not adversely impact other components connected to the pin at the time of testing. Production testing should be planned early in the design process to make sure that a pin is available to handle this signal.

3.2.2 Using RTC and Event System

In this example, the Periodic Interrupt Timer (PIT) functionality of the RTC is used together with the Event System. In this case, the system clock will not have to be switched to a slow oscillator. Refer to 5. Get Source Code from Atmel | START for a link to the project.

Periodic events generated in the PIT can be output using the Event System. As seen above, there are many different periods available. In this example the lowest period PIT event output is used, as can be seen by the line `EVSYS_ASYNCCH3 = EVSYS_ASYNCCH3_PIT_DIV64_gc;`. With the division of the RTC clock frequency set to 64, this will generate a pulse on the selected output pin with a period of 1.95 ms.

Execute the protected write for enabling the 32.768 kHz crystal as early as possible. To better utilize time, it is recommended to perform other production tests while the crystal stabilizes. The below code shows how to setup the PIT as an event generator, and how to output the signal on a pin:

```c
// Deactivate change protection for register Start XOSC32K
// here start-up time is set to the maximum 64K cycles (~2 seconds)
PROTECTED_WRITE(CLKCTRL_XOSC32KCTRLA , CLKCTRL_ENABLE_bm
| CLKCTRL_CSUT_64K_gc | CLKCTRL_RUNSTDBY_bm);

// Use event system channel 3 as path for the PIT event
EVSYS_ASYNCCH3 = EVSYS_ASYNCCH3_PIT_DIV64_gc;
// Connect channel 3 to async user 8 (evout0)
EVSYS_ASYNCUSER8 = EVSYS_ASYNCUSER8_ASYNCCH3_gc;
// muxout evout0 to PA2
PORTMUX_CTRLA = PORTMUX_EVOUT0_bm;

// Enable PIT in RTC
RTC_PITCTRLA = RTC_PITEN_bm;
// Set crystal as clock source for RTC
RTC_CLKSEL = RTC_CLKSEL_TOSC32K_gc;
// Enable RTC
RTC_CTRLA = RTC_RTCEN_bm;
```

3.3 Internal Clock Measurement

Internal frequency measurements can be done at any time, not only in production. A reference clock is needed to do internal measurements. A high or low-frequency clock can be used, or a tick, such as the one-second tick from a GPS module.

Of the two methods, using a high frequency reference will give the best result in the shortest amount of time.

3.3.1 Calibration with High Frequency Input

The idea is to use TCD to count a fast reference frequency within a set period given by the RTC running from a crystal oscillator. The number of TCD overflows, and the final capture value can be compared to an expected value. Refer to 5. Get Source Code from Atmel | START for a link to the projects.

The method consists of the following steps:

1. Apply an external reference frequency to the EXTCLK pin of the device.
2. Configure TCD to use this as its clock source.
3. Configure the RTC to use the XOSC32K clock.
4. Configure the Event System to transmit the overflow signal from the RTC to get a capture value from the TCD.
TCD will start counting from zero after the first capture, therefore the first captured value should be discarded. Count the number of TCD overflows within the RTC period. When the RTC period is done, store the second capture value and use this and the number of TCD overflows as the measured value. The difference between the ideal expected value and the measured value indicates the frequency error of the crystal.

In the example, the RTC period that is measured is set to correspond to one second, provided that the crystal is exact. The external reference used in this example has a frequency of 20 MHz. If the crystal is running at exactly 32.768 kHz, and the period is set to one second, TCD should reach 20,000,000 counts. As TCD is only 12 bits wide it cannot reach this count value. It will overflow 4882 times, and the final capture value should be 3328.

Now, let's say that the final capture value does not reach this ideal value, but instead it reaches 2296, while the number of overflows is still 4882. TCD has counted 3328-2296=1032 cycles less than it should. The crystal is running faster than it should, causing the RTC period to be shorter than expected. In seconds the time difference is 1032/20,000,000=51.6µs/s.
4. **RTC Compensation**

As seen in 2.3 *32.768 kHz Crystal Oscillator Temperature Drift*, a typical 32.786 kHz crystal will drift symmetrically around 25°C. By applying a compensation algorithm to the RTC, it is possible to adjust for the drift.

4.1 **Compensation for Crystal and Load Capacitance Deviation**

In the above example, the crystal was found to be 51.6 µs/s too fast. If we count two extra cycles in the RTC per second, the RTC will become $2\times(1/32768)=60$ µs/s slower. The error is now reduced to being 60-51.6=8.6 µs/s too slow. A software variable can be used to keep track of the accumulated error in each second. When the accumulated error overflows one period of the 32.768 kHz crystal clock, the RTC period is reduced by one cycle for one second.

In the crystal compensation example found in Atmel | START, the precision of the compensation variable is 0.1 µs/s. Increasing the number of decimals counted as the error is possible. In the temperature compensation example, the precision of the compensation is increased to 0.01 µs/s. Refer to 5. Get Source Code from Atmel | START for links to the projects.

4.2 **Compensation for Temperature Drift**

To be able to control the capacitive load of the crystal, and to reduce the noise induced into the crystal oscillator circuit, the 32.786 kHz crystal should be placed close to the device. This also means that the device and the crystal will be exposed to the same temperature.

The devices in the tinyAVR 1-series and megaAVR 0-series have an internal temperature sensor. From the characteristics section in the corresponding device data sheet, it can be seen that the temperature sensor has relatively poor accuracy. This is true, even when using the supplied calibration values located in the signature row of the device, as these values are found when the silicon is still on the wafer. Cutting and packaging can have an impact on the silicon and therefore change temperature readout. The calibration value can further deviate when the device is soldered onto the PCB.

It is possible to achieve better results with the internal temperature sensor if new gain and offset values are measured in production. This requires calibration steps, preferably done after the device has been soldered onto the PCB. If good performance over a wide temperature range is needed, two or three point calibration should be performed. Two of the test points should be slightly above and below the temperature range where accuracy is needed, and one close to the middle. If the crystal frequency is measured at these three points, it is possible to get a more accurate parabolic temperature curve than what is given in the characteristics in the crystal data sheet.

*Note:* While having a high accuracy temperature sensor is optimal, using a low accuracy temperature sensor to do temperature compensation is still useful. This is especially true if the operating temperature is far from 25°C, as the compensation values chosen based on an inaccurate temperature reading will be more accurate than values given for 25°C.

4.3 **Crystal Compensation Look-Up Table Example**

To deal with changing temperatures, it is possible to continuously measure the temperature, and apply compensation for this in the RTC ISR. This will help keep the clock more accurate. To avoid expensive floating or fixed point math, the crystal temperature compensation values can be stored in a look-up table.
Crystal manufacturers typically give a plot or an equation to describe magnitude of the drift as a function of temperature. If higher precision is desired compared to a manufacturers specification, a three-point measurement can be done at different temperatures for each PCB. This way it is possible to find the correct curve for each crystal and circuit. If the measurement of the internal temperature sensor is done at the same time, the data points should align well between the crystal drift and the measured temperature.

In the temperature compensation example found in Atmel | START, the parabolic curve comes from the following mathematical function:

$$\frac{\Delta f}{f_0} = -0.04 \cdot (T - 25)^2$$

The compensation value used in the RTC ISR is taken directly from the table generated from the above function. The RTC overflows every second, and the error value from the table is added to a variable that keeps track of the total accumulated error in each second. When this accumulated error variable is larger than one period of the 32.768 kHz crystal, it subtracts one or more cycles from the RTC period. When the accumulated error is less than one period, the RTC period value is set to the regular 1-second period value.

As the equation is a parabola, and the values of the table will be symmetric around the top point, it is possible to store only the values from 25 °C up to 105 °C and use the subset of values from of 25 °C to up to 90 °C for temperatures between 25 °C to -40 °C. This will reduce the flash size required.

In the example most calculations are done in the RTC and ADC ISR, and the code being executed is not very time consuming.

In the RTC ISR, the accumulated error is calculated. It consists of the accumulated error not yet compensated for, plus the static crystal error per second, plus the temperature compensation error per second. The temperature compensation is found using a look-up in the ADC ISR. The ADC conversion is started from the RTC ISR.

**Note:** The RTC period register is written in the RTC ISR. The value that is written will have to be synchronized to the register in the 32.768 kHz domain. The CPU will not have to wait for this to be done, as long as the previously written value to the same register has been synchronized. In this code, this is not a problem. If the register was written outside of the ISR, or if the PER value was very low (1-3 cycles), so that the ISR is executed in quick succession, it could become a problem. Please note that the RTC has a separate synchronization mechanism for each register. Writing to two different registers in the RTC in quick succession will not be a problem.

In the ADC ISR, the ADC result is read out, gain and offset compensation is performed, and the result is converted to kelvins. The result is then used to find the compensation value to use from the temperature compensation look-up table.

The code in the example uses the internal temperature sensor of the device. It is also possible to use a more accurate external temperature sensor. If this is done, the clock accuracy should increase.
5. **Get Source Code from Atmel | START**

The example code is available through Atmel | START, which is a web-based tool that enables configuration of application code through a Graphical User Interface (GUI). The code can be downloaded for both Atmel Studio and IAR Embedded Workbench® via the direct example code-link(s) below or the *BROWSE EXAMPLES* button on the Atmel | START front page.

**Atmel | START web page:** [ATMEL START Example Code](http://start.atmel.com/#example/)

**Example Code**

- **Frequency Output:**

- **Internal Frequency Calibration:**
  - [http://start.atmel.com/#example/Atmel:rtc_calibration_and_compensation: 1.0.0::Application:RTC_Internal_Frequency_Calibration:](http://start.atmel.com/#example/Atmel:rtc_calibration_and_compensation: 1.0.0::Application:RTC_Internal_Frequency_Calibration:)

- **Crystal Error Compensation:**
  - [http://start.atmel.com/#example/Atmel:rtc_calibration_and_compensation: 1.0.0::Application:RTC_Crystal_Error_Compensation:](http://start.atmel.com/#example/Atmel:rtc_calibration_and_compensation: 1.0.0::Application:RTC_Crystal_Error_Compensation:)

- **Crystal Compensation Look-Up Table:**
  - [http://start.atmel.com/#example/Atmel:rtc_calibration_and_compensation: 1.0.0::Application:RTC_Crystal_Compensation_Look-Up_Table:](http://start.atmel.com/#example/Atmel:rtc_calibration_and_compensation: 1.0.0::Application:RTC_Crystal_Compensation_Look-Up_Table:)

Press *User guide* in Atmel | START for details and information about example projects. The *User guide* button can be found in the example browser, and by clicking the project name in the dashboard view within the Atmel | START project configurator.

**Atmel Studio**

Download the code as an .atzip file for Atmel Studio from the example browser in Atmel | START, by clicking *DOWNLOAD SELECTED EXAMPLE*. To download the file from within Atmel | START, click *EXPORT PROJECT* followed by *DOWNLOAD PACK*.

Double-click the downloaded .atzip file and the project will be imported to Atmel Studio 7.0.

**IAR Embedded Workbench**

For information on how to import the project in IAR Embedded Workbench, open the Atmel | START user guide, select *Using Atmel Start Output in External Tools, and IAR Embedded Workbench*. A link to the Atmel | START user guide can be found by clicking *About* from the Atmel | START front page or *Help And Support* within the project configurator, both located in the upper right corner of the page.
### Revision History

<table>
<thead>
<tr>
<th>Doc. Rev.</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>10/2018</td>
<td>The chapter on Relevant Devices has been updated to include 8/16 KB megaAVR 0-series devices.</td>
</tr>
<tr>
<td>A</td>
<td>07/2018</td>
<td>Initial document release.</td>
</tr>
</tbody>
</table>
The Microchip Web Site

Microchip provides online support via our web site at http://www.microchip.com/. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user’s guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Customer Change Notification Service

Microchip’s customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.


Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
• Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their
code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the
code protection features of our products. Attempts to break Microchip’s code protection feature may be a
violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software
or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for
your convenience and may be superseded by updates. It is your responsibility to ensure that your
application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR
WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY
OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS
CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE.
Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life
support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend,
indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting
from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual
property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BitCloud,
chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Helda, JukeBloxx, KeeLoq,
Klear, LANCheck, LINK MD, maXSylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB,
OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, SAM-BA, SpyNIC, SST,
SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology
Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight
Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip
Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom,
CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM,
dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming,
ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KlearNet, KlearNet logo, memBrain, Mindi, MiWi,
motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient
Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE,
Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total
Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are
trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of
Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.
Quality Management System Certified by DNV

ISO/TS 16949
Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company’s quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001:2000 certified.
<table>
<thead>
<tr>
<th>Region</th>
<th>City</th>
<th>Contact Information</th>
</tr>
</thead>
</table>
| Americas     | Corporate Office   | 2355 West Chandler Blvd. Chandler, AZ 85224-6199  
Tel: 480-792-7200  
Fax: 480-792-7277  
Technical Support:  
http://www.microchip.com/support  
Web Address: www.microchip.com |
| Americas     | Atlanta            | Dunx, GA  
Tel: 678-957-9614  
Fax: 678-957-1455  
| Americas     | Austin             | Tel: 512-257-3370  
Fax: 678-957-1455  
| Americas     | Boston             | Westborough, MA  
Tel: 774-760-0087  
Fax: 774-760-0088  
| Americas     | Chicago            | Itasca, IL  
Tel: 630-285-0071  
Fax: 630-285-0075  
| Americas     | Dallas             | Addison, TX  
Tel: 972-818-7423  
Fax: 972-818-2924  
| Americas     | Detroit            | Novi, MI  
Tel: 248-848-4000  
| Americas     | Houston            | TX  
Tel: 281-894-5983  
| Americas     | Indianapolis       | Noblesville, IN  
Tel: 317-773-8323  
Fax: 317-773-5453  
| Americas     | Los Angeles        | Mission Viejo, CA  
Tel: 949-462-9523  
Fax: 949-462-9608  
Tel: 951-273-7800  
| Americas     | Raleigh            | NC  
Tel: 919-844-7510  
| Americas     | New York           | NY  
Tel: 631-435-6000  
| Americas     | San Jose           | CA  
Tel: 408-735-9110  
Tel: 408-436-4270  
| Americas     | Canada - Toronto   | Tel: 905-695-1980  
Fax: 905-695-2078  
| Asia/Pacific | Australia - Sydney | Tel: 61-2-9868-6733  
| Asia/Pacific | China - Beijing    | Tel: 86-10-8569-7000  
| Asia/Pacific | China - Chengdu    | Tel: 86-28-8665-5511  
| Asia/Pacific | China - Chongqing  | Tel: 86-23-8980-9588  
| Asia/Pacific | China - Dongguan   | Tel: 86-769-8702-9880  
| Asia/Pacific | China - Guangzhou  | Tel: 86-20-8755-8029  
| Asia/Pacific | China - Hangzhou   | Tel: 86-571-8792-8115  
| Asia/Pacific | China - Hong Kong SAR | Tel: 852-2943-5100  
| Asia/Pacific | China - Nanjing    | Tel: 86-25-8473-2460  
| Asia/Pacific | China - Qingdao    | Tel: 86-632-8502-7355  
| Asia/Pacific | China - Shanghai   | Tel: 86-21-3326-8000  
| Asia/Pacific | China - Shenyang   | Tel: 86-24-2334-2829  
| Asia/Pacific | China - Shenzhen   | Tel: 86-755-8864-2200  
| Asia/Pacific | China - Suzhou     | Tel: 86-186-6233-1526  
| Asia/Pacific | China - Wuhan      | Tel: 86-27-5980-5300  
| Asia/Pacific | China - Xian       | Tel: 86-29-8833-7252  
| Asia/Pacific | China - Xiamen     | Tel: 86-992-2388138  
| Asia/Pacific | China - Zhuhai     | Tel: 86-756-3210040  
| Asia/Pacific | India - Bangalore  | Tel: 91-80-3090-4444  
| Asia/Pacific | India - New Delhi  | Tel: 91-11-4160-8631  
| Asia/Pacific | India - Pune       | Tel: 91-20-4121-0141  
| Asia/Pacific | Japan - Osaka      | Tel: 81-6-6152-7160  
| Asia/Pacific | Japan - Tokyo      | Tel: 81-3-6880-3770  
| Asia/Pacific | Korea - Daegu      | Tel: 82-53-744-4301  
| Asia/Pacific | Korea - Seoul      | Tel: 82-2-554-7200  
| Asia/Pacific | Malaysia - Kuala Lumpur | Tel: 60-3-7651-7906  
| Asia/Pacific | Malaysia - Penang  | Tel: 60-4-227-8870  
| Asia/Pacific | Philippines - Manila | Tel: 63-2-634-9065  
| Asia/Pacific | Singapore          | Tel: 65-6334-8870  
| Asia/Pacific | Taiwan - Hsin Chu  | Tel: 886-3-577-8366  
| Asia/Pacific | Taiwan - Kaohsiung | Tel: 886-7-213-7830  
| Asia/Pacific | Taiwan - Taipei    | Tel: 886-2-2508-8600  
| Asia/Pacific | Thailand - Bangkok | Tel: 66-2-694-1351  
| Asia/Pacific | Vietnam - Ho Chi Minh | Tel: 84-28-5448-2100  
| Europe       | Austria - Wels     | Tel: 43-7242-2244-39  
| Europe       | Denmark - Copenhagen | Tel: 45-4450-2828  
| Europe       | Finland - Espoo    | Tel: 358-9-4520-820  
| Europe       | France - Paris     | Tel: 33-1-69-53-63-20  
| Europe       | Germany - Garching | Tel: 49-8931-9700  
| Europe       | Germany - Haan     | Tel: 49-2129-3766400  
| Europe       | Germany - Heilbronn | Tel: 49-7131-67-3636  
| Europe       | Germany - Karlsruhe| Tel: 49-721-825370  
| Europe       | Germany - Munich   | Tel: 49-89-627-144-0  
| Europe       | Germany - Rosenheim| Tel: 49-8031-354-560  
| Europe       | Israel - Ra'anana  | Tel: 972-9-744-7705  
| Europe       | Italy - Milan      | Tel: 39-0331-742611  
| Europe       | Italy - Padova     | Tel: 39-049-7625286  
| Europe       | Netherlands - Drunen| Tel: 31-416-690399  
| Europe       | Norway - Trondheim | Tel: 47-72884388  
| Europe       | Poland - Warsaw    | Tel: 48-22-3325737  
| Europe       | Romania - Bucharest| Tel: 40-21-407-87-50  
| Europe       | Spain - Madrid     | Tel: 34-91-708-08-90  
| Europe       | Sweden - Gothenburg| Tel: 46-31-704-60-40  
| Europe       | Sweden - Stockholm | Tel: 46-8-6090-4654  
| Europe       | UK - Wokingham     | Tel: 44-118-921-5800  
| Europe       | UK - Wokingham     | Tel: 44-118-921-5820  

© 2018 Microchip Technology Inc.