AN2573

ADC Basics with tinyAVR® 0- and 1-series, and megaAVR® 0-series

Introduction

Author: Qubo Hu, Microchip Inc.

Microchip tinyAVR® 0- and 1-series, and megaAVR® 0-series devices feature a 10-bit successive approximation register (SAR) Analog-to-Digital Converter (ADC) and is capable of conversion rates up to 115 ksps. It features a flexible multiplexer, which allows the ADC to measure the voltage at multiple single-ended input pins. Single-ended input channels are referred to ground. The ADC input signal is fed through a Sample-and-Hold circuit which ensures that the input voltage to the ADC is held at a constant level during sampling. It also features multiple internal ADC reference voltages between 0.55V and V_{DD}.

An ADC conversion can be started by software, or by using the Event System (EVSYS) to route an event from other peripherals. A window compare feature is available for monitoring the input signal and can be configured to trigger an interrupt on user-defined thresholds for under, over, inside, or outside a window, with minimum software intervention required.

This application note describes the basic functionality of the ADC in Microchip tinyAVR® 0- and 1-series, and megaAVR® 0-series devices in Single-Ended mode. It contains code examples for Microchip devices to get started.

Features

- ADC Free-Running mode
- ADC Single Conversion mode
- ADC Window Comparator mode
- ADC Sample Accumulator mode
- Atmel | START Example code

Table of Contents

1. Relevant Devices

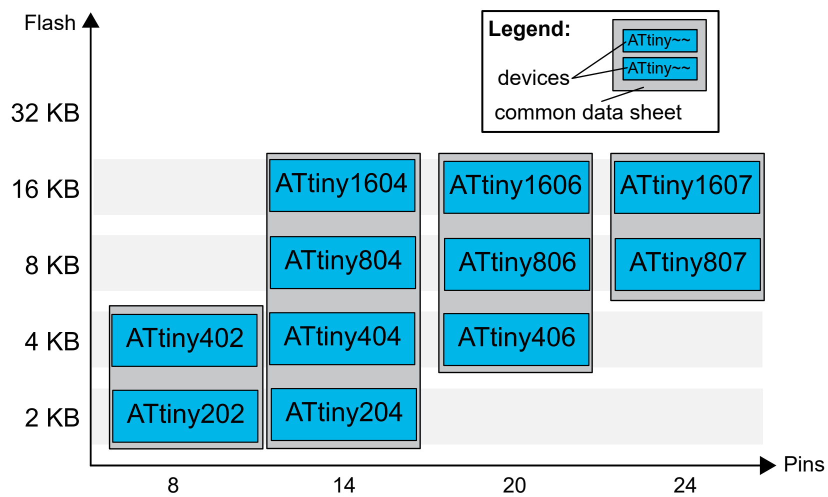
This chapter lists the relevant devices for this document.

1.1 tinyAVR 0-series

The figure below shows the tinyAVR 0-series, 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 1-1. tinyAVR® 0-series Overview



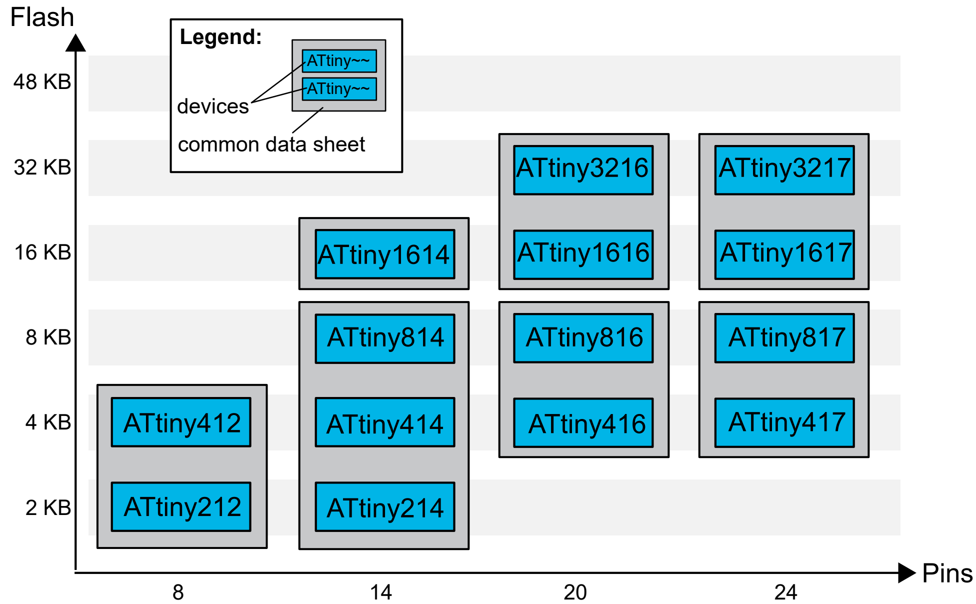
Devices with different Flash memory size typically also have different SRAM and EEPROM.

1.2 tinyAVR 1-series

The following figure 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 1-2. tinyAVR® 1-series Overview



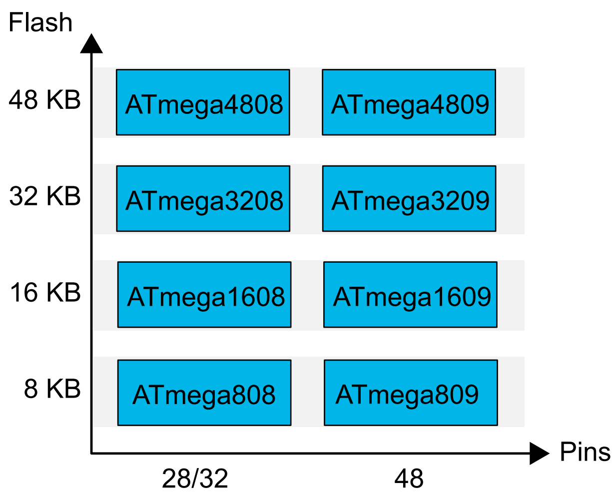
Devices with different Flash memory size typically also have different SRAM and EEPROM.

1.3 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 1-3. megaAVR® 0-series Overview

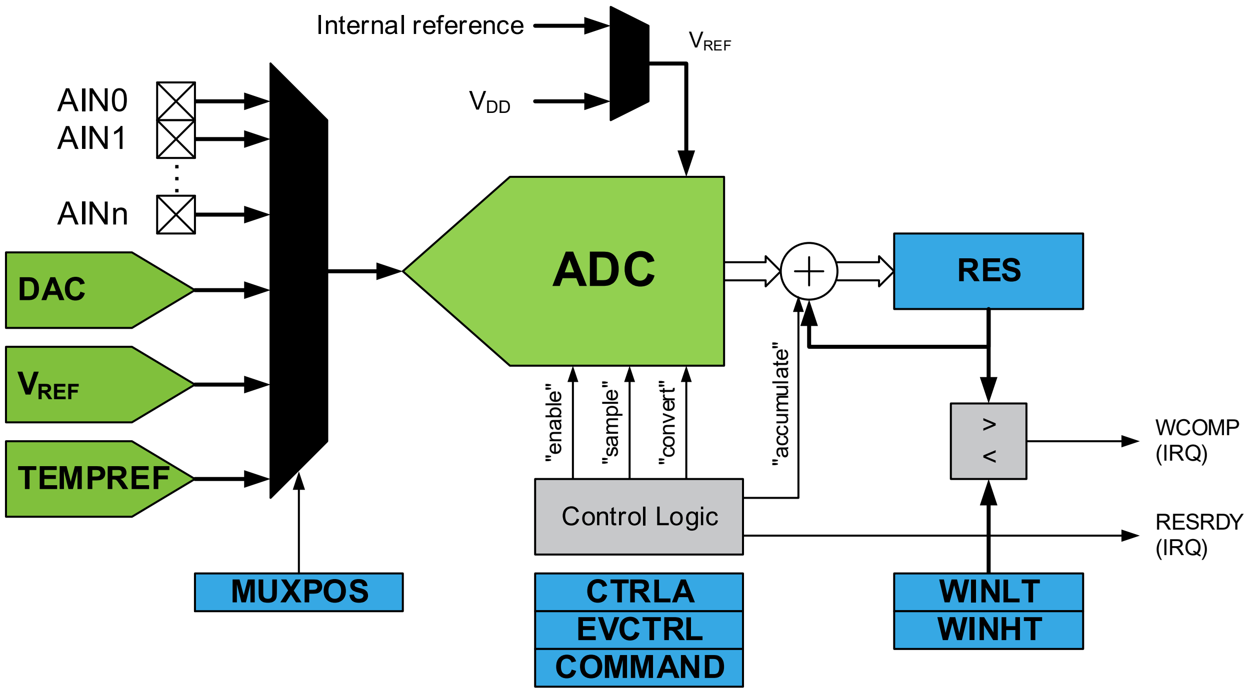


Devices with different Flash memory size typically also have different SRAM and EEPROM.

2. Module Overview

This chapter gives an overview of the ADC functionality and basic configuration options of the ADC.

Figure 2-1. Block Diagram

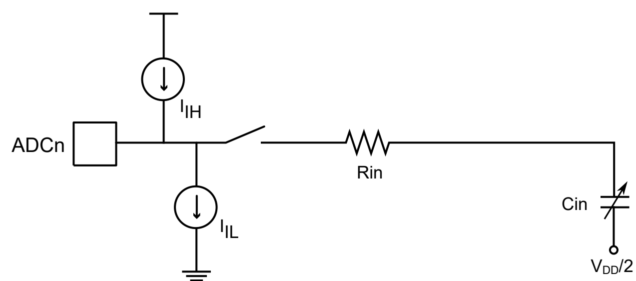


2.1 Analog Input Circuitry

The analog input circuitry is illustrated in Figure 2-2. An analog source applied to ADCn is subjected to the pin capacitance and input leakage of that pin (represented by I_H and I_L), regardless of whether that channel is selected as input for the ADC. When the channel is selected, the source must drive the S/H capacitor through the series resistance (combined resistance in the input path).

The ADC is optimized for analog signals with an output impedance of approximately 10 kΩ or less. If such source is used, the sampling time will be negligible. If a source with higher impedance is used, the sampling time will depend on how long the source needs to charge the S/H capacitor, which can vary substantially.

Figure 2-2. Analog Input Schematic



2.2 ADC Operation

The analog input channel is selected by writing to the MUXPOS bits in the MUXPOS register (ADC.MUXPOS). Any of the ADC input pins, GND, internal Voltage Reference (V_{REF}), or temperature sensor, can be selected as single-ended input to the ADC. The ADC is enabled by writing a '1' to the ADC ENABLE bit in the Control A register (ADC.CTRLA). Voltage reference and input channel selections will not go into effect before the ADC is enabled. The ADC does not consume power when the ENABLE bit in ADC.CTRLA is '0'. The RESSEL bit in ADC.CTRLA register selects the resolution of ADC to 8 or 10 bits.

2.2.1 Starting Conversion

In Single Conversion mode, a conversion is triggered by writing a '1' to the ADC Start Conversion bit (STCONV) in the Command register (ADC.COMMAND). This bit is '1' as long as the conversion is in progress, and will be cleared when the conversion is completed. If a different input channel is selected while a conversion is in progress, the ADC will finish the current conversion before changing the channel.

In Free-Running mode, the first conversion is started by writing the STCONV bit to '1' in ADC.COMMAND. A new conversion cycle is started immediately after the previous conversion cycle has completed. The Result Ready flag (RESRDY) in the Interrupt Flag register (ADC.INTFLAG) is set.

If the Result Ready Interrupt Enable (RESRDY) bit in the Interrupt Control (ADC.INTCTRL) register and the Global Interrupt Enable bit in the CPU Status register (CPU_SREG.I) are '1', the Result Ready Interrupt vector will be executed.

The RESRDY Interrupt flag in ADC.INTFLAG will be set, even if the specific interrupt is disabled, allowing software to check for finished conversion by polling the flag. A conversion can thus be triggered without causing an interrupt.

Alternatively, a conversion can be triggered by an event. This is enabled by writing a '1' to the Start Event Input bit (STARTEI) in the Event Control register (ADC.EVCTRL). Any incoming event routed to the ADC through the Event System (EVSYS) will trigger an ADC conversion. This provides a method to start conversions at predictable intervals or at specific conditions. The event trigger input is edge sensitive. When an event occurs, STCONV in ADC.COMMAND is set. STCONV will be cleared when the conversion is complete.

2.2.2 ADC Conversion Result

After the conversion is complete (RESRDY is '1'), the conversion result RES is available in the ADC Result Register (ADCn.RES). The result for a 10-bit conversion is given as:

$$\mathrm{RES} = \frac{1023 \times V_{\mathrm{IN}}}{V_{\mathrm{REF}}}$$

where V_{IN} is the voltage on the selected input pin and V_{REF} the selected voltage reference (see description for REFSEL in ADCn.CTRLC and ADCn.MUXPOS).

2.2.2.1 ADC Result With 10- or 8-bit Resolution

With full 10-bit resolution, the 10-bit ADC results are stored to the ADC Result register (ADC.RES). With 8-bit resolution, the conversion results are truncated to eight bits (MSBs) before they are stored to the ADC Result register (ADC.RES). The two Least Significant bits are discarded.

2.3 ADC Clock and Conversion Timing

The ADC uses the CLK_PER peripheral clock and has an internal prescaler to generate the ADC clock source CLK_ADC with a frequency between 50 kHz and 1.5 MHz for maximum resolution. If a lower resolution than 10 bits is selected, the input clock frequency to the ADC can be higher than 1.5 MHz to get a higher sample rate. The prescaling is selected by writing to the Prescaler bits (PRESC) in the Control C register (ADC.CTRLC).

When a conversion is initiated, the conversion starts at the following rising edge of the CLK_ADC clock cycle. The prescaler is kept in reset as long as there is no ongoing conversion. This assures a fixed delay from the trigger to the actual start of a conversion in CLK_PER cycles as $\mathrm{StartDelay} = \frac{\mathrm{PRESC}_{\mathrm{factor}}}{2} + 2$.

A normal conversion takes 13 CLK_ADC cycles. The actual Sample-and-Hold takes place two CLK_ADC cycles after the start of a conversion.

Both sampling time and sampling length can be adjusted by using the Sample Delay bit field in the Control D (ADC.CTRLD) and the Sample Length bit field in the Sample Control register (ADC.SAMPCTRL). Both of these control the ADC sampling time in a number of CLK_ADC cycles. Total sampling time is given by:

$$\mathrm{SampleTime} = \frac{(2 + \mathrm{SAMPDLY} + \mathrm{SAMPLEN})}{f_{\mathrm{CLK\_ADC}}}.$$

In Free-Running mode, the sampling rate R_S is calculated by:

$$\mathrm{SampleRate} = \frac{f_{\mathrm{CLK\_ADC}}}{(13 + \mathrm{SAMPDLY} + \mathrm{SAMPLEN})}$$

2.4 Changing Channel

The MUXPOS bits, the ADC.MUXPOS register, and the REFSEL bits in the ADC.CTRLC register are buffered through a temporary register to which the CPU has random access. This ensures that the channel and reference selections only take place at a safe point during the conversion.

Once the conversion starts, the channel and reference selections are locked to ensure sufficient sampling time for the ADC.

In Single Conversion mode, the channel may be selected before starting the conversion. The channel selection may be changed one ADC clock cycle after writing '1' to the STCONV bit.

In Free-Running mode, the channel may be selected before starting the first conversion. The channel selection may be changed one ADC clock cycle after writing '1' to the STCONV bit. Since the next conversion has already started automatically, the next result will reflect the previous channel selection.

2.5 Reference Selection

This ADC features multiple internal ADC reference voltage between 0.55V and V_{DD} and external reference input V_{REFA}. Different internal reference voltages are 0.55V, 1.1V, 1.5V, 2.5V, and 4.3V.

The reference voltage for the ADC controls the conversion range of the ADC. Input voltages that exceed the selected reference voltage will be converted to the maximum result value of the ADC. For an ideal 10-bit ADC, the maximum result value is 0x3FF. V_{REF} can be selected by writing the Reference Selection bits (REFSEL) in the Control C register (ADC.CTRLC) as either V_{DD}, external reference V_{REFA}, or an internal reference. V_{DD} is connected to the ADC through a passive switch. The internal reference is generated from an internal bandgap reference through an internal amplifier, controlled by the Voltage Reference (V_{REF}) peripheral.

3. Source Code Overview

One application with four use cases has been developed and tested on the ATtiny817 Xplained Pro board. Here are the general system configurations which are common for all four use cases:

- CPU clock: 3.33 MHz
- Peripherals used:
  – ADC, V_{REF}, and USART
- Details of the peripheral configurations:
  – ADC
    - Resolution at 10 bits
    - Input channel is AIN 6: pin PA6
  – V_{REF} set to 2.5V
  – USART
    - TXEN: Transmission Enable is set
    - Baud Rate: 9600
  – GPIO output pin PB4: LED0

The projects available in Atmel START generate peripheral driver functions and files, as well as a '`main()`' function that initializes all drivers.

- Driver header and source files are in the *src* and *include* folder.
- In the `atmel_start.c` file, '`atmel_start_init()`' initializes MCU, drivers, and middleware in the project.

In the following section, the macro configurations used in `main.c` file and the source code for the four cases application will be explained.

3.1 Macro Configurations

Below are the macro configurations in the `main.c` file.

- Application with four use case options, with defined values 1 to 4 respectively, as shown below:

```
#define FREE_RUNNING                  1
#define SINGLE_CONVERSION             2
#define WINDOW_COMPARATOR_MODE        3
#define SAMPLE_ACCUMULATOR            4
```

- Which use case to run is selected by changing the value of the macro `EXAMPLE_CODE`:

```
#define EXAMPLE_CODE  SAMPLE_ACCUMULATOR
```

- The following macros are also used in case it can easily be changed afterward. Maximum input voltage:

```
#define MAX_VOL 2.5  //VREF=2.5V
```

Maximum input reference voltage is configured to 2.5V to calculate the voltage for measured ADC reading.

Note: ADC reference (V_{REF}) has been configured to 2.5V.

- ADC result resolution:

```
#define RES_10BIT    0x3FF
```

  ADC resolution has been configured to be 10 bits, with the maximum result value of 0x3FF.
- ADC input channel:

```
#define ADC_CHANNEL  5
```

  ADC input signal has been connected to channel 5: AIN5 (pin PA5).

3.2 ADC Free-Running

In this use case, ADC runs in Free-Running mode. It runs by configuring the `EXAMPLE_CODE` definition to `FREE_RUNNING` in `main.c` as:

```
#define EXAMPLE_CODE  FREE_RUNNING
```

The first conversion is started when the initial configuration of the Free-Running mode is executed. When the conversion cycle is completed, the ADC result is read from the ADC0.res register. The formula of how ADC0.res is calculated is described in 2.2.2 ADC Conversion Result. The result is then converted to voltage format in the next line of the code, which is a reverse process of calculating ADC0.res. The converted voltage is then printed through the USART to the terminal. One can choose to use the terminal already embedded in Data Visualizer in Atmel Studio or any other external terminal. A new conversion cycle is then started immediately after the previous conversion cycle is completed.

Note: The ‘`ADC_0_is_conversion_done()`’ function is defined in `adc_basics.c`. One can open the `src/adc_basics.c` file to check it. It will be redefined in the `adc_windows.c` file when the ADC Window Compare mode is used and all related drivers will be included in the corresponding driver file.

3.3 ADC Single Conversion

In this use case, the ADC runs in Single Conversion mode. It runs by configuring the `EXAMPLE_CODE` definition to `SINGLE_CONVERSION` in `main.c` as:

```
#define EXAMPLE_CODE  SINGLE_CONVERSION
```

The conversion starts after the ‘`read_adc_single_conversion()`’ function is called. The ‘`ADC_0_get_conversion()`’ function is then called, in which ADC_COMMAND.STCONV is written to '`1`'. When the conversion cycle is completed, the ADC data is converted to voltage format which is printed through the USART to the terminal. One can choose to use the terminal like the one already embedded in Data Visualizer in Atmel Studio.

Since the ‘`read_adc_single_conversion()`’ function is called continuously in the `while(1)` loop in the ‘`main()`’ function, a single conversion is manually configured to start after the previous one is completed. This is why one can observe that the calculated voltage is continuously printed to the terminal, same behavior as in the Free-Running mode. The difference is that, in the Free-Running mode, ADC_COMMAND.STCONV is only written to '`1`' once when the ‘`ADC_0_start_conversion()`’ function is called before the `while(1)` loop starts.

Note: The ‘`ADC_0_get_conversion(ADC_CHANNEL)`’ function, which reads the ADC data, is defined in `adc_basics.c`. It will be redefined in the `adc_windows.c` file when the ADC Window Compare mode is used and all related drivers will be updated in the corresponding driver file `adc_windows.c` instead of `adc_basics.c`.

3.4 ADC Window Compare

In this use case, ADC runs in Window Compare mode. The ADC can raise a flag (ADC_INTFLAG.WCOMP) and request an interrupt (WCOMP) when the result of a conversion is above and/or below certain thresholds. There are four threshold mode options:

- The result is under a threshold
- The result is over a threshold
- The result is inside a window
- The result is outside a window

In this use case, the result below window option is chosen. This use case runs by configuring the `EXAMPLE_CODE` definition to `WINDOW_COMPARATOR_MODE` in `main.c` as:

```
#define EXAMPLE_CODE  WINDOW_COMPARATOR_MODE
```

The ADC is set in Free-Running mode. LED0 is turned ON when the function '`ADC_0_get_window_result()`' returns true, meaning the ADC result is under the window threshold value and the Interrupt Flag bit of the Windows Compare mode is set. Otherwise, LED0 is kept OFF. The calculated voltage is then printed through the USART to the terminal when a conversion is completed.

3.5 ADC Sample Accumulator

In this use case, ADC runs in Sample Accumulating mode. It is configured to accumulate 64 samples automatically in one conversion in order to average out noise or to get averaged ADC result. Consequently, the Conversion Complete flag is only raised once, after taking the last sample of the accumulation. The ADC result is averaged over the configured number of samples. It is then converted to voltage format which is printed through the USART to the terminal.

This use case runs by configuring the `EXAMPLE_CODE` definition to `SAMPLE_ACCUMULATOR` in `main.c` as:

```
#define EXAMPLE_CODE  SAMPLE_ACCUMULATOR
```

4. 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 below or the *Browse examples* button on the Atmel | START front page.

Atmel | START web page: http://start.atmel.com/

Example Code

- ADC Basics with tinyAVR® 1-series:
  – http://start.atmel.com/#example/Atmel:adc_basics_with_tinyavr_1_series:1.0.0::Application:ADC_Basics_with_tinyAVR_1-series:
- ADC Basics with megaAVR® 0-series:
  – http://start.atmel.com/#example/Atmel:adc_basics_with_megaavr:1.0.0::Application:ADC_Basics_with_megaAVR_0-Series:

Click *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 *Help* from the Atmel | START front page or *Help And Support* within the project configurator, both located in the upper right corner of the page.

5. Revision History

Doc. Rev.	Date	Comments
C	10/2018	Updated figures 1-1, 1-2, 1-3 in chapter "Relevant Devices". Fixed grammar and punctuation.
B	02/2018	Added support for tinyAVR 0-series and megaAVR 0-series.
A	11/2017	Initial document release

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.

To register, access the Microchip web site at http://www.microchip.com/. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

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

Trademarks

ISBN: 978-1-5224-3676-8

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, KEELOQ® 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.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
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 **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4450-2828 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-67-3636 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820