AN2521

CRCSCAN on tinyAVR® 0- and 1-series, and megaAVR® 0-series

Introduction

Authors: Irun Walberg, Qubo Hu, Microchip Technology Inc.

Devices in the tinyAVR® 1-series and megaAVR® 0-series feature a Cyclic Redundancy Check (CRC) Memory Scan (CRCSCAN) peripheral, which can be used to detect errors in the target section of the Flash.

This application note will demonstrate how to configure CRCSCAN using Priority mode, and how to precalculate a CRC checksum and add this at the end of Flash. The application note also features a section on bootloader considerations, and an example of post-build commands to be used when combining bootloader and application code.

Features

- Pre-calculation of CRC Checksums
- Code Example for Priority mode
- Bootloader Considerations

Table of Contents

1. Relevant Devices

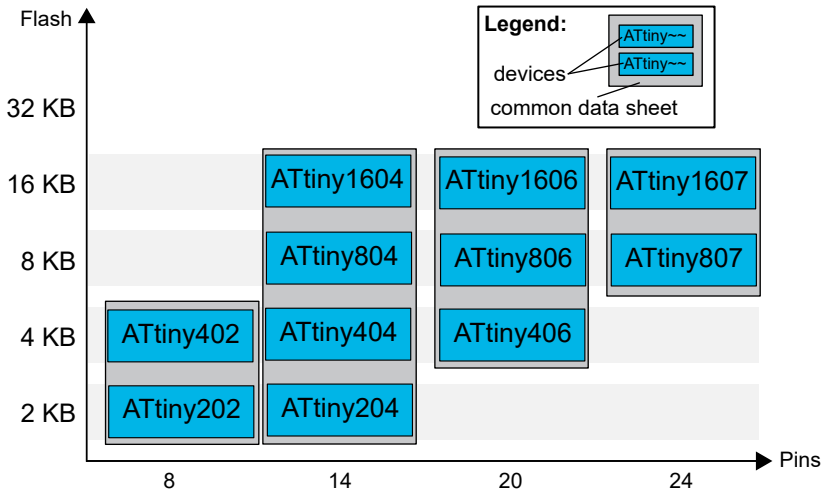
This chapter lists the relevant devices for this document.

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



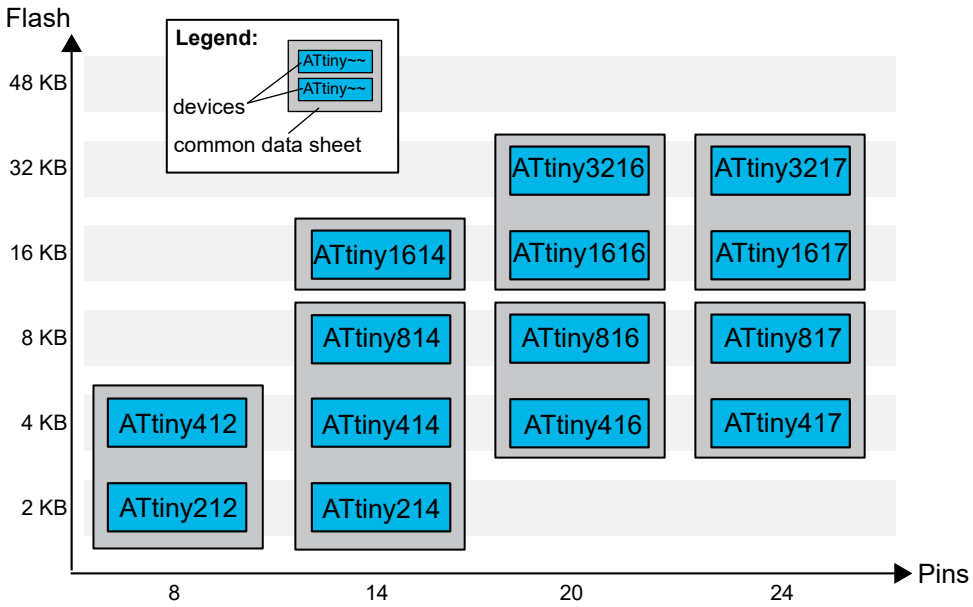
Devices with different Flash memory size typically also have different SRAM and EEPROM.

1.2 tinyAVR® 1-series

The 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 1-2. tinyAVR® 1-series Overview



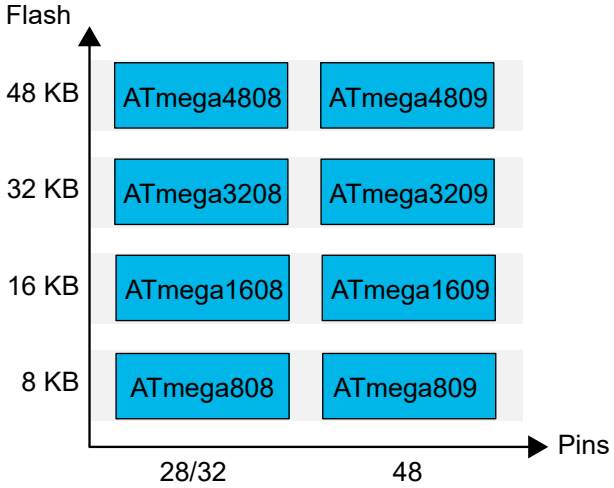
Devices with different Flash memory size typically also have different SRAM and EEPROM.

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

The CRCSCAN takes a stream of data from the Nonvolatile Memory (NVM) (either the entire Flash, only the Boot section, or both application code and Boot section) and generates a checksum. The last location in the section to check must contain the correct pre-calculated 16-bit checksum for comparison. If the checksum calculated by CRCSCAN and the pre-calculated checksum matches, a status bit in CRCSCAN will be set. If they do not match, the same status bit will be cleared. CRCSCAN can be configured to generate a non-maskable interrupt (NMI) if the checksums don’t match. The CRC-generator supports CRC-16 (CRC-CCITT).

CRCSCAN is a Core Independent Peripheral (CIP) which can be configured to run without any intervention from the CPU after initialization. It can be enabled in the firmware or via a debugger. When the NMIEN bit in the CTRLA register is set, any CRC failure will trigger the NMI, and the CRC failure can be handled in the NMI interrupt routine. Priority scan mode is supported, and the module can also be enabled during startup to ensure the Flash is OK before letting the CPU execute the code. Refer to fuse description in the data sheet for further information.

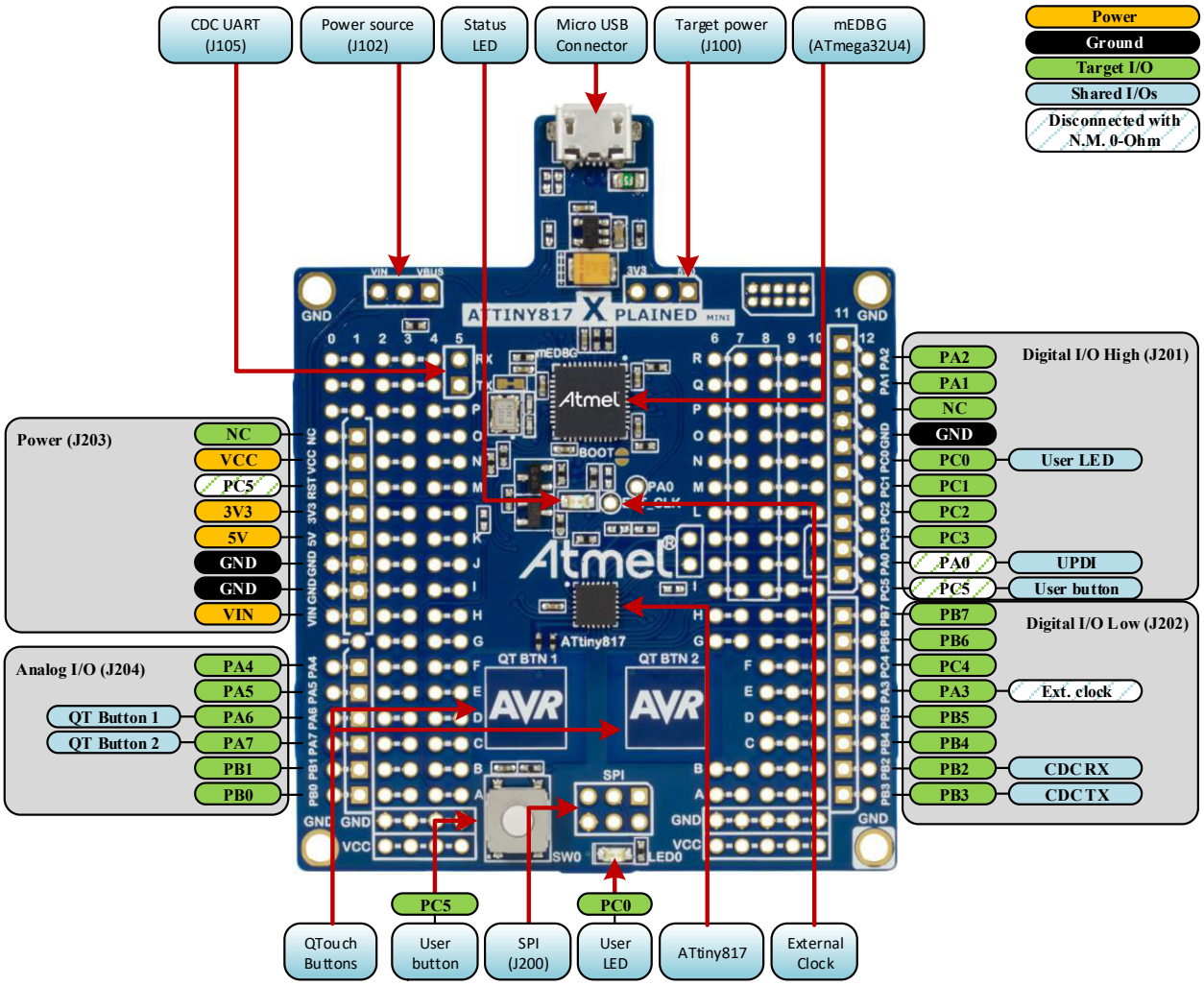
3. Hardware Setup

The example in this application note is written for the ATtiny817 Xplained Mini, which features the ATtiny817 device, but the example applies to all devices in the tinyAVR 1-Series. The example is available through Atmel | START. Refer to 7. Get Source Code from Atmel | START for further information.

In this application note, the onboard LED, LED0, is used to give a visual indication of the CRCSCAN status, and it has the following connection to ATtiny817 (also shown in the figure below):

- LED0 is connected between PC0 and GND

Figure 3-1. LED Connection on ATtiny817 Xplained Mini



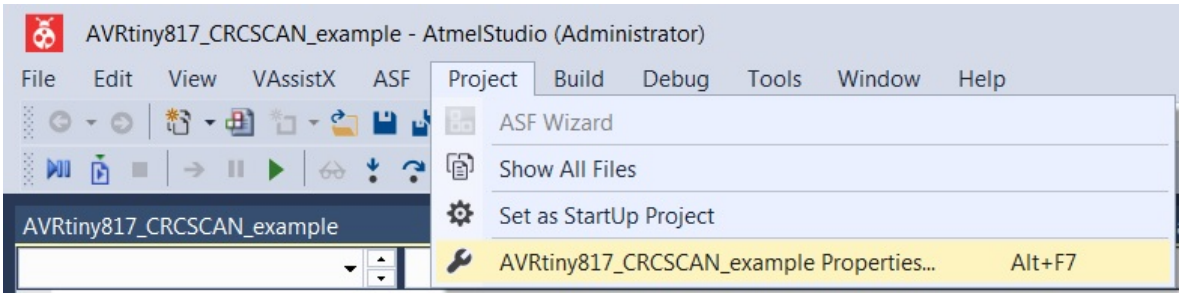
4. CRC Checksum Setup

The CRC checksum needs to be contained in the last location of the section which the CRCSCAN is configured to check. This means that a CRC checksum needs to be calculated for an application using CRCSCAN, and added at the end, before the application is programmed to the device.

In this application note, the *srec_input* is the tool used to generate the checksum and add it to the end of the hex file. This is done through Atmel Studio, where *srec_input* is added as a post-build command for the project, and the end result will be a hex file which will contain the application code and its CRC checksum:

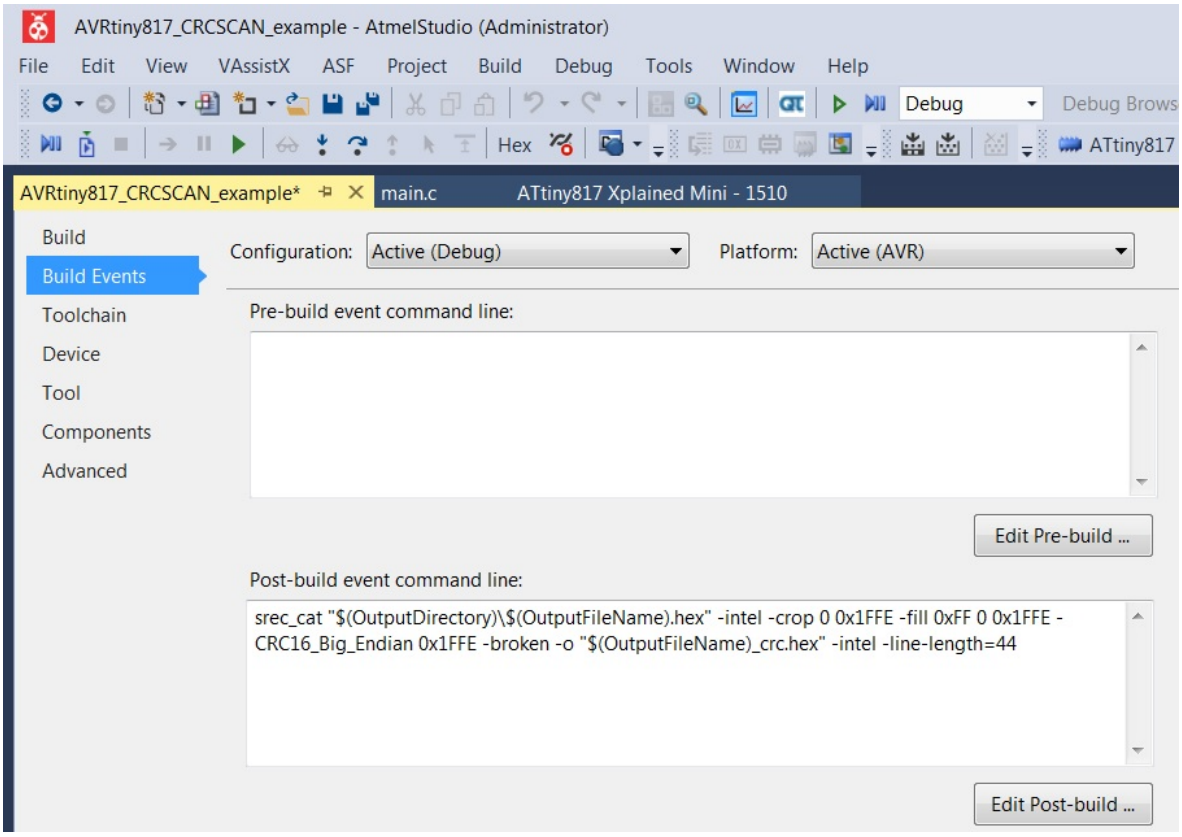
1. Open the project properties by clicking *Project → <project_name> Properties*, as shown in the figure below.

   Figure 4-1. Project Properties in Atmel Studio

2. Add a post-build command to the project under *Build Events → Post-build event command line*, as shown in the figure below.

Figure 4-2. Post-Build Command in Atmel Studio



3. Rebuild the project by clicking *Build → Rebuild <project_name>.*

5. Check Entire Flash in Priority Mode

In priority mode, CRCSCAN will run a single check of Flash, and the CPU will be stalled until CRCSCAN has completed. In this example, CRCSCAN is configured to run in priority mode on the entire Flash. When CRCSCAN has completed its check, the application code will check the CRC calculation status, which will be reflected by the frequency of LED0 on ATtiny817 Xplained Mini.

Initialization

In this application, all the module configurations exist in the *main.c* file.

1. Configure CRCSCAN
   - In `main()`, set the MODE bit fields (MODE<1:0>) in CRCSCAN CTRLB register equal to 0x0 in order to run a single CRC check where CRCSCAN has priority to Flash. This will stall the CPU until the scan has finished.
   - Set the SRC bit fields (SRC<1:0>) in CRCSCAN CTRLB register equal to 0x0 to perform a CRC check on the entire Flash (the boot, application code, and application data sections).
2. Write the ENABLE bit (ENABLE) to '`1`' in the CRCSCAN CTRLA register to enable the CRCSCAN peripheral with the current configurations to start a scan.

Program Flow

- The STATUS register is monitored in an endless `while(1)` loop.
  1.1. If the OK bit in the STATUS register is set, LED0 toggles at the configured OK_DELAY interval (the default value is 500 ms) indicating that the CRC check has successfully completed.
  1.2. If the CRCSCAN fails, LED0 will toggle at the configured ERROR_DELAY (the default value is 50 ms). This may indicate that the NMI handler is corrupted.

Post-Build Command

Calculating the CRC checksum and storing it in Flash can be done by using a post-build command in Atmel Studio. Follow the instructions in 4. CRC Checksum Setup with the following post-build command:

```
srec_cat "$(OutputDirectory)\$(OutputFileName).hex" -intel -crop 0 0x1FFE -fill 0xFF 0 0x1FFE
-CRC16_Big_Endian 0x1FFE -broken -o "$(OutputFileName)_crc.hex" -intel -line-length=44
```

Info: The post-build command will take the generated *<project_name>.hex* file from the build output, fill the remaining space of the 8 KB Flash with 0xFF, calculate the CRC, and put the CRC checksum at the last two bytes of the 8 KB block, address 0x1FFE. The resulting file will be named *<project_name>_crc.hex*. A complete command line description and options can be found at http://srecord.sourceforge.net/man/man1/srec_input.html.

Tip: The post-build command might fail due to improper break-line interpretation during copy-paste. Fix the improper break-line or type in the command manually.

Testing the Application

Program the ATtiny817 by clicking *Tools→Device programming→(Tool=mEDBG, Device=ATtiny817, Interface=UPDI, Apply)→ Memories →Program*.

If the device is programmed with the *<program_name>.hex* file, the CRC check will fail since there is no CRC checksum added at the end of the Flash, and LED0 will toggle at FAIL_DELAY interval.

If the device is programmed with a *<program_name>_crc.hex* file, the pre-calculated checksum will be stored at the end of the 8 KB Flash address. The CRCSCAN should run through with no failure and LED0 will toggle at the OK_DELAY interval.

6. Bootloader Considerations

A bootloader is a small application located in the Boot section, and its main purpose is to update the application section when needed. Typically the bootloader communicates with a host via a serial communication interface and the host provides the bootloader with the new application code.

A simple bootloader must be forced to do an update, i.e. by a button being pushed, whereas a more advanced bootloader can initiate an update when needed. Typically, on start-up, the bootloader will check if the application code is still valid, and if it is not, it will initiate an update. If no update is needed, the execution is transferred from the bootloader to the application code.

When using CRCSCAN in combination with a bootloader and application code, it is highly recommended that the bootloader handles the use of CRCSCAN. It is reasonable to assume that the application code is more frequently updated, hence the bootloader will be assumed to be the most trustworthy part.

When a bootloader is used to upload new firmware, only the application section of the Flash will be updated, and as CRCSCAN always includes the Boot section, one must make sure that the new checksum added at the end of the application section takes the old Boot section into consideration.

One of the easiest solutions is to calculate a separate checksum for the Boot section, and adding this at the end of the Boot section which will result in a checksum of 0x0000 when CRCSCAN has calculated the Boot section. This method results in the initial seed for the application code is the same, independent of changes in the bootloader code. In other words, when updating the application code, the new pre-calculated CRC checksum can be based on the application code only. An example for matching post-build commands are given below, with the necessary configurations for CRCSCAN, fuses, and project.

CRCSCAN Configuration

In this example, any CRCSCAN mode can be used, but CRCSCAN should be configured to scan the Boot and Application sections.

Fuse Configuration

In this example, the Boot section is 512B and the Application section is 3.5 KB. The fuses will have to be set accordingly.

- BOOTEND = 0x02
- APPEND = 0x10

Atmel Studio Project Setting for Application Code

In this example, the Application section will start at 0x200, which is achieved by a memory relocation in Atmel Studio.

- *Project→<project_name> Properties→Toolchain→AVR®/GNU Linker→Memory Settings→FLASH segment→.text=0x100*

Info: *.text* is the location of the code in a project, and *Memory Settings* is 16-bit addressed.

Post-build Command for Bootloader Code

```
srec_cat "$(OutputDirectory)\$(OutputFileName).hex" -intel -crop 0 0x1FE -fill 0xFF 0 0x1FE -
CRC16_Big_Endian 0x1FE -broken -o "$(OutputFileName)_crc.hex" -intel -line-length=44
```

Info: The post-build command will take the generated *<project_name>.hex* file from the build output, fill the remaining space of the 512B Boot section with 0xFF, calculate the CRC, and put the CRC checksum at the last two bytes of the 512 KB block, address 0x1FE. The resulting file will be named *<project_name>_crc.hex*.

Post-build Command for Application Code

```
srec_cat "$(OutputDirectory)\$(OutputFileName).hex" -intel -crop 0x200 0xFFE -fill 0xFF 0x200
0xFFE -CRC16_Big_Endian 0xFFE -xmodem -o "$(OutputFileName)_crc.hex" -intel -line-length=44
```

Info: The Application section is located at address 0x200-0x1000, in this example, and *xmodem* provides the CRC checksum calculation with an initial seed of 0x0000. The post-build command will take the generated *<project_name>.hex* file from the build output, fill the remaining space of the 3.5 KB Application section with 0xFF, calculate the CRC, and put the CRC checksum at the last two bytes of the 3.5 KB block, address 0xFFE. The resulting file will be named *<project_name>_crc.hex*.

7. 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 7.0 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: http://microchip.com/start

Example Code

- CRCSCAN Priority Mode for Entire Flash:
  - http://start.atmel.com/#application/Atmel:crcscan_priority_mode_for_entire_flash:1.0.0::Application:CRCSCAN_Priority_Mode_for_Entire_Flash:

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.

8. Revision History

Doc. Rev.	Date	Comments
C	10/2018	Chapter "Relevant Devices" has been updated to include 8/16 KB megaAVR 0-series devices.
B	02/2018	Chapter "Relevant Devices" has been updated to include also tinyAVR 0-series and megaAVR 0-series.
A	08/2017	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: 978-1-5224-3659-1

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