
Using the XMEGA® Watchdog Timer

Features

- Watchdog Timeout Programmable Through Fuses
- Watchdog Settings Lockable Through Fuses
- Watchdog with Normal and Window mode
 - Normal mode:
Programmable timeout period between 8ms and 8 seconds
 - Window mode:
Programmable closed window period between 8ms and 8 seconds
 - Total timeout period between 16ms and 16 seconds, using Window mode
- Example Source Code
 - Initialization of Normal and Window mode operation
 - Disabling and reconfiguration of watchdog

Introduction

Author: Qubo Hu

Watchdog timers are used to ensure that a system can recover from unforeseen failures in firmware or hardware. The watchdog timer will, if used correctly, be able to detect abnormalities in the program execution and respond by resetting the microcontroller (MCU). This brings the MCU to a well-defined and known state where normal operation can be resumed.

The XMEGA® AVR® family offers a very robust internal watchdog: Ordinary integrated watchdog timers often use the CPU clock as clock source, but the XMEGA watchdog timer's clock source is independent from the CPU clock. This means that failure of the main clock would not affect the watchdog timer operation.

Further, the XMEGA watchdog timer offers the Normal mode, where the watchdog timer must be Reset before a given timeout period, and it offers a Window mode where the watchdog timer can be Reset within a limited period. In Window mode, if the watchdog timer is Reset too early (or too late), a system Reset is triggered.

More information about using watchdog timers can be found in application note [AN2551](#).

Table of Contents

Features.....	1
Introduction.....	1
1. Theory of Operation.....	3
1.1. Normal Mode.....	3
1.2. Window Mode.....	3
1.3. Timer Clock.....	4
1.4. Timeout Periods.....	4
1.5. Digital Timing of the Watchdog Timer.....	5
1.6. Watchdog Enabling and Timing Configuration.....	6
1.7. Intended Use of the Watchdog.....	7
2. Examples.....	9
2.1. Files.....	9
2.2. Doxygen Documentation.....	9
3. Revision History.....	10
The Microchip Web Site.....	11
Customer Change Notification Service.....	11
Customer Support.....	11
Microchip Devices Code Protection Feature.....	11
Legal Notice.....	12
Trademarks.....	12
Quality Management System Certified by DNV.....	13
Worldwide Sales and Service.....	14

1. Theory of Operation

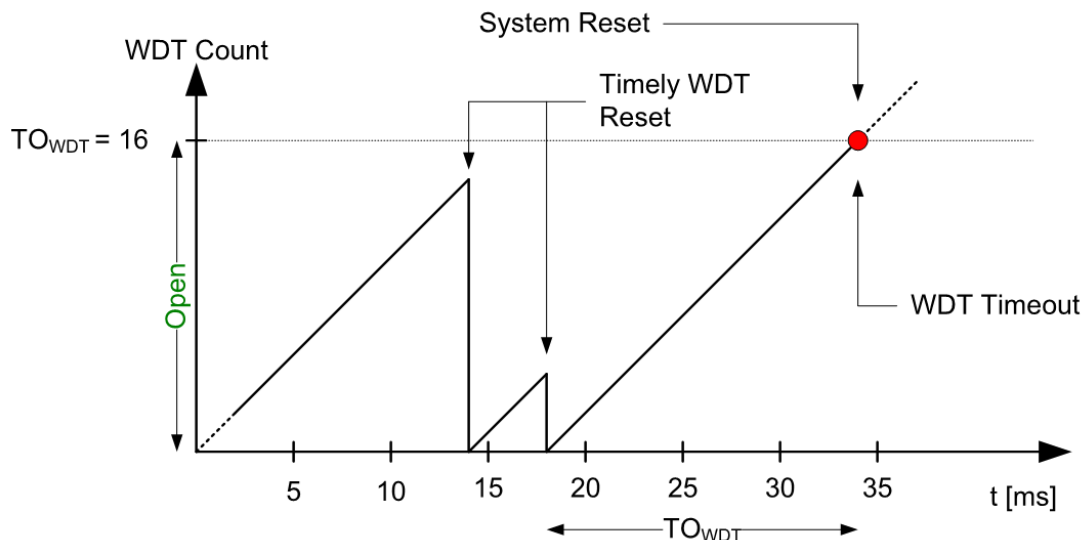
Before the XMEGA watchdog timer can be discussed, it is important to get a few terms clear:

- The Watchdog Timer (WDT) is the peripheral module that can be configured to generate a system Reset, if the timer is Reset too early or too late according to specified timeout periods. The timer value itself cannot be read or written, only Reset.
- A Watchdog Timer Reset (WDT Reset) is when the timer in the WDT is cleared (or Reset). This will make the timer start counting from zero again, and thus restart the timeout period.
- A system Reset is when the AVR microcontroller is Reset, resetting the CPU and I/O register to default values and restarting program execution from address 0x0000 (or boot section). The WDT can cause a system Reset if it times out or, in Window mode, if the WDT is Reset too early.

1.1 Normal Mode

The WDT can be used in Normal mode, which is when a single timeout period is set for the WDT. If the WDT is not Reset before the timeout occurs the WDT will cause a system Reset. Figure 1-1 illustrates this. The open range along the WDT Count axis indicates that the WDT can be Reset at any time before the WDT Timeout (TO_{WDT}) expires (refer to section [Digital Timing of the Watchdog Timer](#) for exact timing details), as opposed to when using the Window mode where the WDT cannot be Reset until after the Watchdog Window Timeout (TO_{WDTW}) expires.

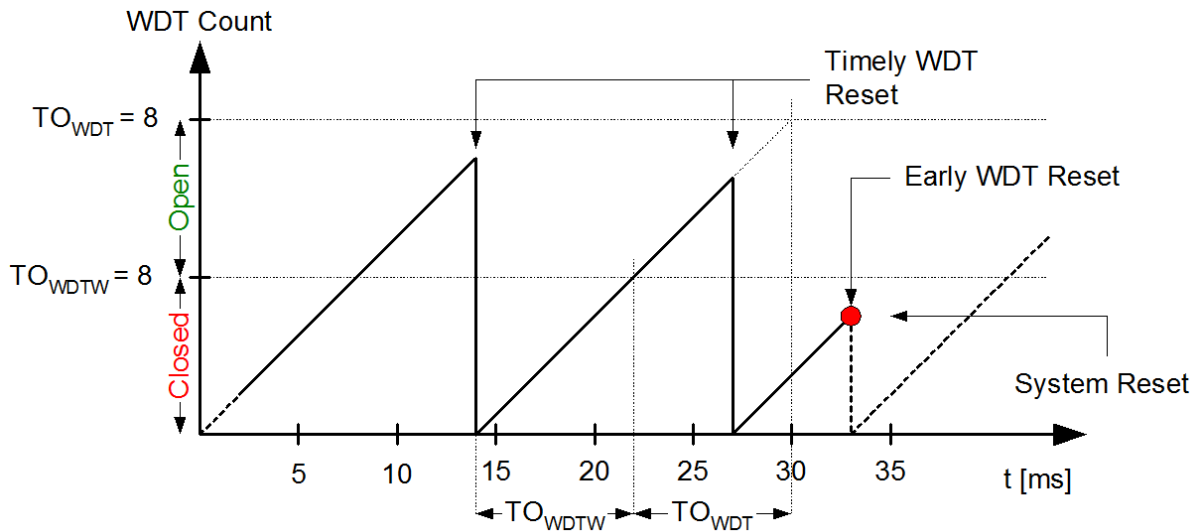
Figure 1-1. Timely and Late (Missing) Watchdog Timer Reset in Normal Mode



1.2 Window Mode

When the WDT is used in Window mode, it uses two different timeout periods: a closed window timeout period (TO_{WDTW}) and the normal WDT timeout period (TO_{WDT}). The first defines a period from 8ms to 8s where the WDT cannot be Reset. If the WDT is Reset in this period the WDT will cause a system Reset. The normal WDT timeout period, which is also 8ms to 8s, defines the duration of the open period, in which the WDT can (and should) be Reset. The open period will always follow the closed period, and thus the total duration of the WDT timeout is the sum of the windowed and the normal timeouts. The closed and open periods used in Window mode are illustrated in Figure 1-2.

Figure 1-2. Timely and Early Watchdog Timer Reset in Window Mode



1.3 Timer Clock

The WDT is clocked from the 1 kHz output from the internal 32 kHz Ultra Low Power (ULP) oscillator. This oscillator can also be used by the RTC timer and is used by the brown-out detection circuit, if used in Sampled mode. If any of these modules are configured to use the ULP oscillator the oscillator is running. The additional current consumption by enabling additional modules using the ULP RC oscillator is very low. Refer to the datasheet for more information about power consumption.

It is important to be aware that the clock for the WDT is not very accurate. This is due to the fact that the oscillator is designed to draw very little power to be able to use the WDT even in long-life battery powered applications. The downside of low power oscillators is low accuracy. The typical accuracy of the clock for the WDT is $\pm 30\%$ (refer to data sheet for exact information on accuracy of the clock). This means that the clock frequency can vary from one device to another. When designing software using the WDT, the device-to-device variation must be addressed to ensure that the timeout periods used are valid for all devices, and not only the ones used in the lab during development.

Further, the clock source may vary over temperature and supply voltage – though this variation is significantly less than the $\pm 30\%$ device-to-device variations. Refer to the data sheet for more information on this topic.

1.4 Timeout Periods

The WDT offers a wide range of timeout periods, from 8ms to 8 seconds (refer to data sheet for details). The timeout period for the WDT period and the WDT window period is controlled by the WDT Period (PER) bits and the WDT Window Period (WPER) bits located in the WDT Control Register (CTRL) and WDT Window Control Register (WINCTRL).

The WDT is Reset when a valid write access to CTRL or WINCTRL register is performed (refer the description of the timed sequence required to write the control registers found in section [Run-time Enabling of the Watchdog Timer](#)).

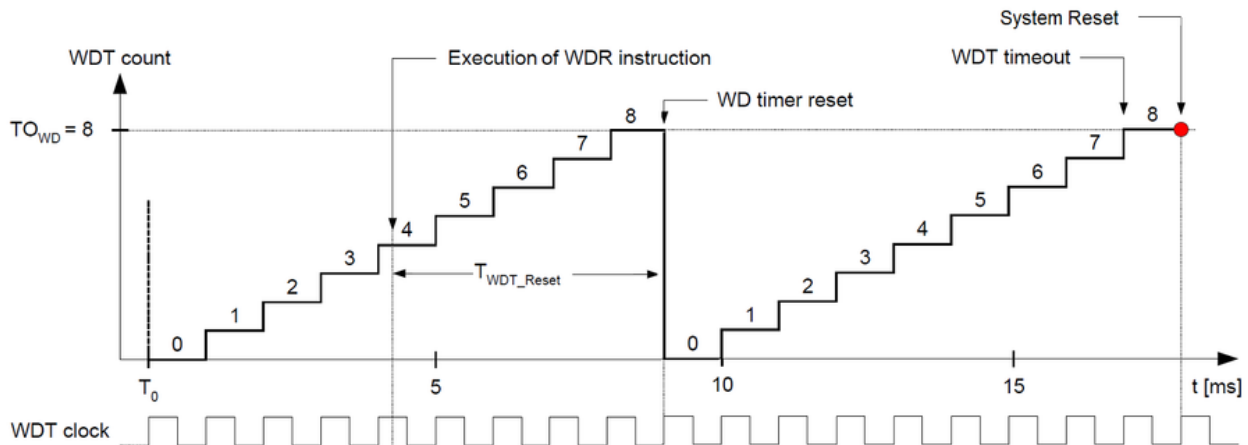
1.5 Digital Timing of the Watchdog Timer

The WDT is operating in a different clock domain than the CPU and synchronization between the two domains should be considered when using the WDT.

It takes up to three WDT clock cycles and three CPU clock cycles to configure the WDT. When configuration settings are written to the WDT control registers (CTRL and WINCTRL), the new configuration becomes effective from the next WDT clock edge (rising edge of WDT clock in [Figure 1-3](#)), i.e., between 4ms and 5ms after the configuration is written (three WDT clock cycles and three CPU clock cycles means it takes up to 4-5ms depending on how fast the CPU clock is running, while WDT is running at 1kHz clock frequency). This means that the initial timeout period is up to 5ms longer than the specified timeout. If the specified timeout is 8ms, the actual timeout will be between 12ms and 13ms. This will be relevant when using the Window mode and short timeout periods. This characteristic is not unique to the WDT: All asynchronous timers operate in this way due to synchronization between clock domains.

Again, the WDT is Reset when a valid write access to the control registers is performed (refer to section [Run-time Enabling of the Watchdog Timer](#) for further details).

Figure 1-3. Digital Timing of the XMEGA Watchdog Timer



Another timing characteristic is that the duration between the execution of the Watchdog Timer Reset instruction (WDR) and the actual resetting of the WDT, is subject to synchronization between clock domains. The WDT is Reset on the third WDT clock edge after the WDR is executed (see TWDT_Reset in [Figure 1-3](#)), which means that the WDT is Reset between 4ms and 5ms after the WDR instruction is executed. Considering the use of a 8ms timeout period, the first WDR instruction should be executed within 3ms after enabling the WD. Taking the $\pm 30\%$ accuracy of the ULP oscillator into account, the WDR instruction must be executed within 2.1ms or less. The interval between subsequent WDR instructions should be 4.9ms or less (8ms – 1ms uncertainty – 30% oscillator uncertainty). The effect of this synchronization is minimized when the timeout period increases. This strongly indicates that using the Window mode in combination with short timeout periods requires very strict timing.

If the WDT causes a system Reset, e.g., by timing out, the system Reset occurs on the first following WDT clock edge (see [Figure 1-3](#)). This means that the system Reset occurs 1ms after the WDT timeout period has expired. This normally does not cause a problem, but it is useful to know if trying to measure the WD timeout period by monitoring the logic level of a pin. A better way to determine the actual frequency of the WDT clock is to use the XMEGA Real Time Clock timer, which can be clocked by the ULP oscillator.

All of the above mentioned conditions apply to both the WDT Normal mode timeout and the WDT Window mode timeouts.

1.6 Watchdog Enabling and Timing Configuration

The WDT can be enabled and configured in two ways:

- Either through fuses, which makes the AVR load the specified settings before leaving system Reset.
- Or the WDT can be enabled at run-time, which means that the firmware writes the desired settings to the WDT control registers at run-time.

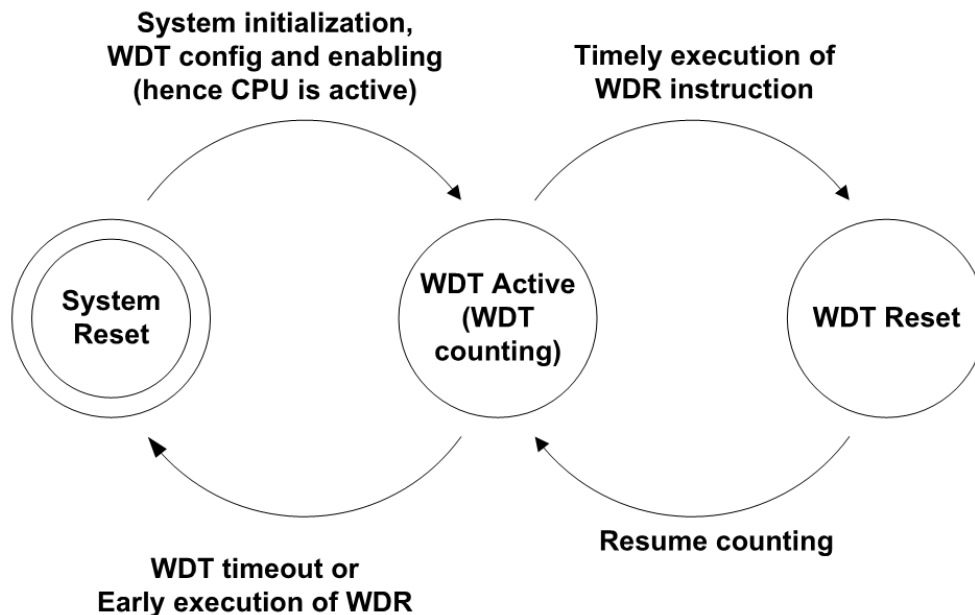
1.6.1 Fused Enabling of the Watchdog

The WDT can be initialized automatically during system Reset by writing the fuses corresponding to the Normal mode and to Window mode. These fuses will determine both the watchdog timer timeout period and the duration of the WDT closed window, and can be used to enable the WDT at start-up. Refer to the data sheet for more information about the fuses controlling the WDT.

Note: The Window mode cannot be enabled through fuses as it is an advantage to be able to enable it run-time to ensure proper timing when using Window mode.

To provide maximum protection, it is possible to program the Watchdog Lock (WDLOCK) fuse, which ensures that the WDT cannot be disabled during run-time – neither accidentally, nor on purpose. This can be desirable if considering that a system failure may for some reason impair the enabling of the watchdog at run-time (in the firmware).

Figure 1-4. State Chart for Enabling WD through Fuses



1.6.2 Run-time Enabling of the Watchdog Timer

In some applications it can be required to disable the WDT to reduce the power consumption to an absolute minimum in Sleep mode. This can be the case in long life battery powered applications. If it is desirable to use the WDT in periods where the CPU is active, but not in Sleep modes, it is convenient to enable and disable the WDT, though this may reduce the protection offered by the WD.

Note: If other modules are using the ULP oscillator in Sleep mode (BOD and RTC) the additional current consumption spent by leaving the WDT on is very low.

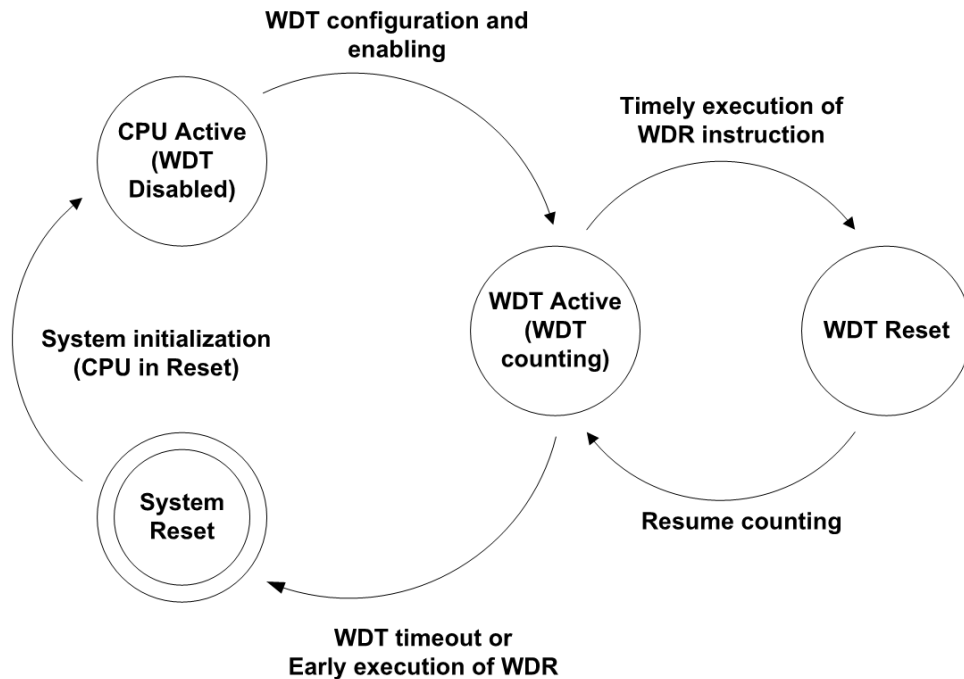
To enable the WDT in run mode or to change the timeout period, a timed sequence must be used: The timed sequence involves the Configuration Change Protection register (CCP), which controls access to certain registers. After writing the signature 0xD8 to the CCP, the WDT can be reconfigured within the next four instruction cycles. During these four instruction cycles the global interrupts are automatically disabled to ensure that the timed sequence is not invalidated by an interrupt.

Note: The timeout is given in instruction cycles and not clock cycles. A DMA transfer will therefore not affect the timing: DMA transfers are not counted as instruction cycles, only instructions executed by the CPU are counted.

Within four instruction cycles after writing the 0xD8 signature to the CCP, the WDT control registers (CTRL and WINCTRL) can be modified. The procedure for doing so is specified in the data sheet, and examples on how to do it can be found in the code example for this application note.

Note: It is not possible to change the WDT or the WDT Window mode configuration while the WDLOCK fuse is set, but it is possible to enable/disable the WDT Window mode.

Figure 1-5. State Chart for Enabling WDT Run-Time



1.7 Intended Use of the Watchdog

A WDT is meant to save the day if the system has an unforeseen failure that is not handled in firmware or hardware, or if an external disturbance causes the system to fail. A well-used WDT would be able to generate a system Reset, which the end-user will barely notice. Seeing this in contrast to a product that needs power cycling every now and then to return to an operational state, the difference should be apparent: Whether the end-user is satisfied with the product or not.

In general it is recommended to issue a WDT Reset from somewhere in the main loop of the firmware. Do not Reset the WDT in interrupt service routines, unless the interrupt routine checks a series of flags that

confirms correct execution of various parts of the firmware. If these simple rules are followed, the WDT is hard to misuse.

The WDT Window mode is a bit more challenging to use than the Normal mode, as it involves more strict control of the WDT Reset timing. In window mode the WDT should be Reset from somewhere within the main loop, never in interrupt service routines, as this would impair the closed window protection. Because the closed window defines the minimum expected duration of the main loop (or subsections of the main loop), it can be used to catch cases where parts of the main loop code is not executed, or cases where early exit from function calls occurs. For example, when an algorithm or other software failure causes an operation to finish too quickly, like the expected duration of writing a value to the EEPROM is 4ms. However, if it completes in a few microseconds (inspection of a flag failed?), a subsequent WDT Reset would arrive too early. Another example is corruption of the return stack or stack pointer itself, causing abnormal program execution.

Another case where the Window mode offers good protection is if the code execution is stuck in a loop where the WDR instruction is executed repeatedly. If the WDT Reset occurs more frequently than anticipated, the WDT will assume that a failure has occurred and Reset the system to bring it back to an operational state.

2. Examples

This application note includes a source code package with a basic watchdog timer driver implemented in C.

Note: This watchdog timer driver is not intended for use with high-performance code.

It is designed as a library to get started with the XMEGA watchdog timer. For timing and code space critical application development, you should access the watchdog timer registers directly. Refer to the driver source code and device data sheet for more details.

2.1 Files

The source code package consists of three files:

- wdt_driver.c – Watchdog Timer driver source file
- wdt_driver.h – Watchdog Timer driver header file
- wdt_example.c – Example code using the driver

For a complete overview of the available driver interface functions and their use, refer to the source code documentation.

2.2 Doxygen Documentation

All source code is prepared for automatic documentation generation using Doxygen. Doxygen is a tool for generating documentation from source code by analyzing the source code and using special keywords. For more details about Doxygen visit <http://www.doxygen.org>. Precompiled Doxygen documentation is also supplied with the source code accompanying this application note, available from the readme.html file in the source code folder.

3. Revision History

Doc Rev.	Date	Comments
AN2654A	02/2018	Bug fixed in Section 1.5 Digital Timing of the Watchdog Timer . Converted appnote to Microchip SDL format. Microchip AN2654 replaces Atmel 8034C.
8034C	02/2015	Bug fixed in Section 1.3 . Template updated.
8034B	04/2009	No info.
8034A	03/2007	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

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, BeaconThings, BitCloud, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Helder, JukeBlox, KeeLoq, KeeLoq logo, Kleer, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, RightTouch, 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, chipKIT, chipKIT logo, CodeGuard, CryptoAuthentication, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, QMatrix, RightTouch logo, 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.

© 2018, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-2717-9

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	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: 47-7289-7561 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