TB3290

MPLAB Harmony v3 Synchronous Drivers and Their Usage in FreeRTOS-Based Applications

Introduction

MPLAB® Harmony v3 allows users to configure the MPLAB Harmony v3 drivers in one of the following two operating modes: Synchronous or Asynchronous. The Asynchronous drivers support a non-blocking implementation model for the data transfer operations, and the Synchronous drivers support a blocking implementation model. Asynchronous drivers can be used in both Real Time Operating System (RTOS) and non-RTOS based environments while the synchronous drivers are suitable for RTOS-based environments. Synchronous drivers are designed to adhere to the blocking need of applications using RTOS.

This document describes using MPLAB Harmony v3 synchronous drivers with FreeRTOS-based applications.

1. Description

A typical design model for a microcontroller-based embedded application is to factor the application functionality into multiple tasks. These tasks are run cooperatively in a super loop (`while(1)` loop) in the `main` function.

Each application task performs a small portion of its functionality and allows the control to move to the other tasks. These tasks will use hardware features (For example, reading/writing to peripherals, such as I^2C, USART, SPI) to implement portions of the work. For the application to work correctly, the peripheral drivers used by these tasks need to be implemented in a non-blocking manner.

The user need to consider the following application, implemented using the cooperative multitasking model.

Problem Statement: The application reads the current room temperature from a temperature sensor. The temperature reading is displayed on a serial console periodically every second. Further, the application writes the temperature readings to the EEPROM. When a character is entered on the console, the last five written temperature values are read from the EEPROM and displayed on the console.

The application is divided into two tasks, the Sensor task and the EEPROM task, and each running their own state machines. The Sensor task and the EEPROM tasks are called from the `SYS_Tasks` routine which is run in a `while(1)` loop. The user triggers the read of the last five stored values in the EEPROM by a user input (keypress on the keyboard) through an asynchronous USART read complete event.

Figure 1-1. Application Design

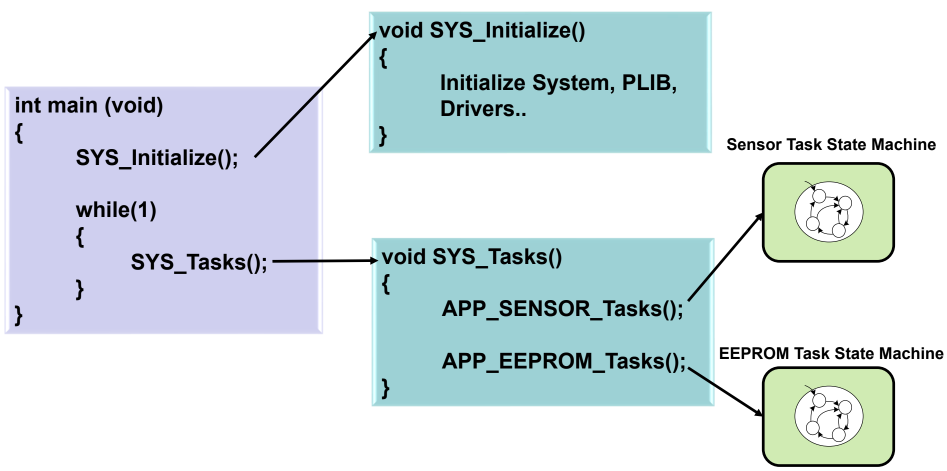
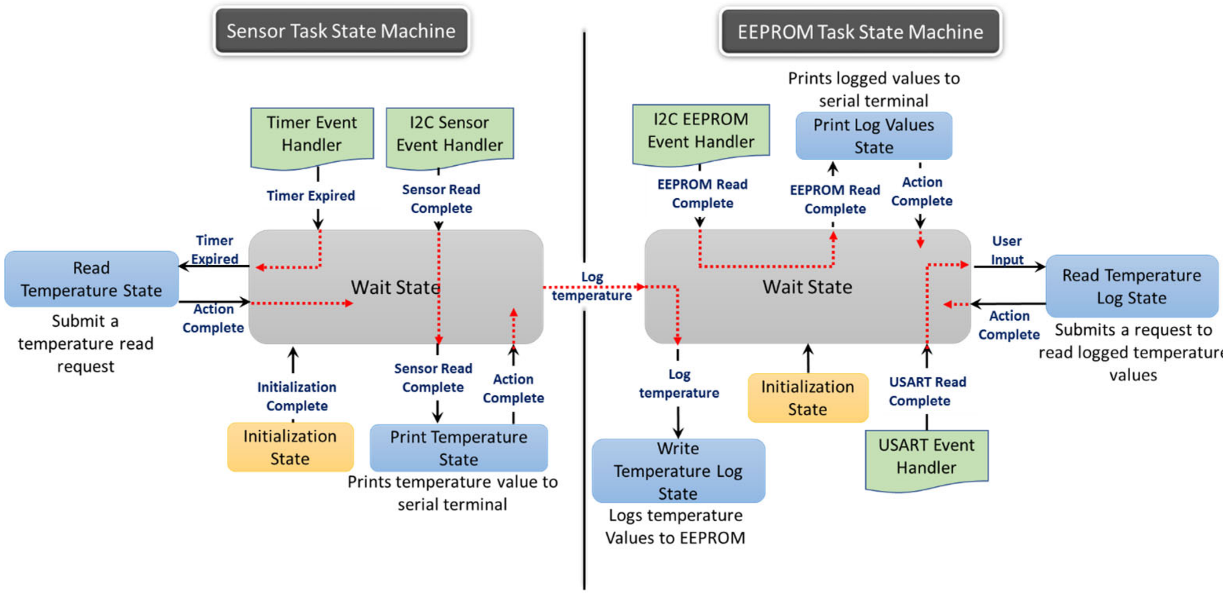


Figure 1-2. Sensor and EEPROM Task State Machine



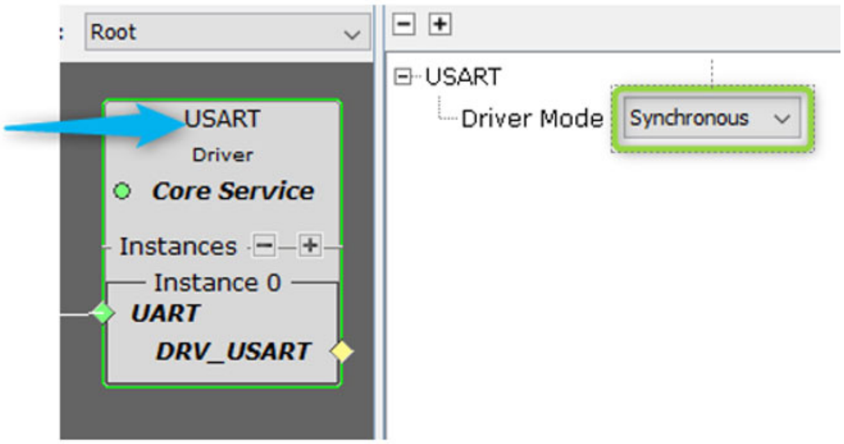
The above state machine based (non-RTOS) application design can be implemented using the asynchronous mode of the MPLAB Harmony v3 drivers.

The cooperative multitasking application design model works well in many situations, but has limitations and faces implementation challenges when the application being designed is large and feature rich constituting multiple sub tasks. This design model faces implementation challenges when an existing application is scaled or extended as the tasks rely on each other regularly giving up the CPU. Also, there are specific situations when a task cannot give up the CPU to other tasks due to its inherent blocking implementation. For example, the file system style read and write functions block and do not return until the entire transfer has completed. In such situations, the application designer can use a Real Time Operating System to perform multitasking and simultaneously allow the use of blocking driver function calls.

2. Configuration

In the MPLAB Harmony v3 Configurator (MHC) project graph, click on the driver block and configure Synchronous mode as shown in the following figure.

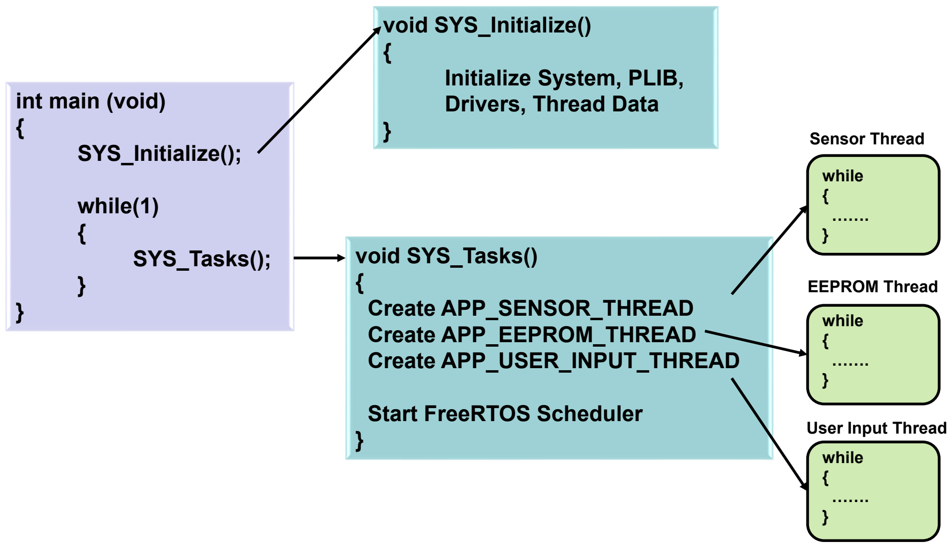
Figure 2-1. Synchronous Mode



3. Implementation

The application design with FreeRTOS using the MPLAB Harmony v3 synchronous driver is shown in the following figure.

Figure 3-1. Application Threads

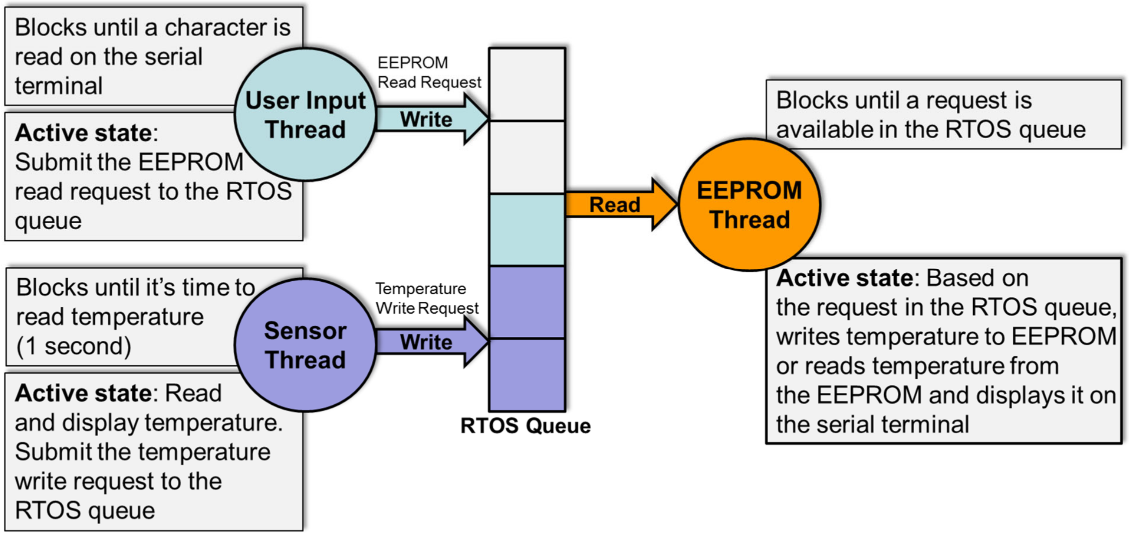


- Sensor Thread: To read and display the temperature periodically.
- EEPROM Thread: To write the temperature values to the EEPROM and display it on the Serial COM-port terminal when requested by the user.
- User Input Thread: To read the character entered on the Serial COM-port terminal to retrieve the last five stored values in the EEPROM.

The Sensor thread, EEPROM thread, and User Input thread run the corresponding application functions in an infinite loop. These threads are created from `SYS_Tasks` routine. Following the creation of these threads, the FreeRTOS scheduler is invoked. The scheduler performs the preemptive scheduling of these threads based on the waiting, ready and running state of each of them. Once the scheduler is invoked, the threads will run based on the default scheduling method (preemptive).

The inter-thread communication between the three threads using the FreeRTOS queue is shown in the following figure.

Figure 3-2. Application Inter-Thread Communication



By default, the sensor thread is blocked and wakes up every time the temperature sampling period gets expired (1 second). Once active, the sensor thread reads the latest room temperature value from the temperature sensor and prints the value on the serial terminal. It also notifies the EEPROM thread through the RTOS queue of the availability of the latest temperature value which needs to be stored in the EEPROM. Once notified, the sensor threads block again for temperature sampling period duration. The following code examples illustrate this with the usage of the FreeRTOS APIs and the MPLAB Harmony v3 driver APIs in Synchronous mode.

```
/* Allow other threads to run until it is time to read temperature */
/* Runs the following code every 1 second. */
vTaskDelay(APP_SENSOR_SAMPLING_RATE_IN_MSEC / portTICK_PERIOD_MS);

/* Submit a blocking I2C write-read request (for temperature). */
/* Calls I2C driver API in synchronous mode  */
if (true == DRV_I2C_WriteReadTransfer(app_sensorData.i2cHandle,
        APP_SENSOR_I2C_SLAVE_ADDR, (void*) &registerAddr, 1,
        (void *) app_sensorData.i2cRxBuffer, 2))
{
    /* ………… Application code ………… */
    /* Use FreeRTOS Queue to notify the temperature write event and temperature
     * value to EEPROM thread. */
    xQueueSend(eventQueue, (void*) &app_sensorData.eventInfo, portMAX_DELAY);

    /* ………… Application code ………… *
    /* Print the temperature value by submitting a blocking USART write request. */
    DRV_USART_WriteBuffer(app_sensorData.usartHandle,
            app_sensorData.usartTxBuffer, strlen);
}
```

By default, the EEPROM thread is in the waiting state for an event to occur. It wakes up when there is an event in the RTOS queue to either write the latest temperature value, or the read last 5 temperature values. Based on the event, the EEPROM thread performs writing or reading, and goes back to the waiting state. The following code examples illustrate this with the usage of FreeRTOS APIs and the MPLAB Harmony v3 driver APIs in synchronous mode.

```
/* Wait for the temperature write request OR EEPROM read request. */
xQueueReceive(eventQueue, &app_eepromData.eventInfo, portMAX_DELAY);

if (app_eepromData.eventInfo.eventType == EVENT_TYPE_TEMP_WRITE_REQ)
{
    /* ………… Application code ………… *
    /* Write temperature to EEPROM */
    if (true == DRV_I2C_WriteTransfer(app_eepromData.i2cHandle,
            APP_EEPROM_I2C_SLAVE_ADDR, (void *) app_eepromData.i2cTxBuffer, 2))
    {
        /* Check if EEPROM has completed the write operation */
        while (false == DRV_I2C_WriteTransfer(app_eepromData.i2cHandle,
                APP_EEPROM_I2C_SLAVE_ADDR, (void *) &dummyData, 1));
```

```
    }

    /* ………… Application code ………… */
}
if (app_eepromData.eventInfo.eventType == EVENT_TYPE_TEMP_READ_REQ)
{
    /* Read the logged temperature values from the EEPROM */
    app_eepromData.i2cTxBuffer[0] = APP_EEPROM_LOG_MEMORY_ADDR;
    if (true == DRV_I2C_WriteReadTransfer(app_eepromData.i2cHandle, \
            APP_EEPROM_I2C_SLAVE_ADDR, app_eepromData.i2cTxBuffer, 1,\
            app_eepromData.i2cRxBuffer, 5))
    {
        /* Print the log values on the terminal */
        APP_EEPROM_PrintTemperature(app_eepromData.i2cRxBuffer,
                app_eepromData.wrIndex);
    }
}
```

By default, the user input thread is blocked to receive a character on the USART receive line. Once a character is received, the user input thread becomes active and submits an EEPROM read request to the EEPROM thread through the RTOS queue to read the last five temperature values stored in the EEPROM. After serving the user input request, the thread goes back to the blocking state to receive another character on the USART receive line. The following code examples illustrate this with the usage of FreeRTOS APIs and the MPLAB Harmony v3 driver APIs in Synchronous mode.

```
    /* Submit a blocking USART read request (user input). Here------> Step #5 */
    if (DRV_USART_ReadBuffer(app_user_inputData.usartHandle, &usartData, 1 ) == true)
    {
        app_user_inputData.eventInfo.eventType = EVENT_TYPE_TEMP_READ_REQ;
        app_user_inputData.eventInfo.eventData = usartData;

        /* Use FreeRTOS queue to notify the EEPROM task to print the logged
    temperature values */
        xQueueSend( eventQueue, &app_user_inputData.eventInfo, portMAX_DELAY );
    }
```

4. References

- Getting Started with MPLAB Harmony v3 drivers on SAM E70/S70/V70/V71 MCUs using FreeRTOS: microchipdeveloper.com/harmony3:same70-getting-started-tm-drivers-freertos
- The Differences Between MPLAB Harmony v3 Synchronous and Asynchronous Drivers and When to Use Them: ww1.microchip.com/downloads/en/DeviceDoc/The-Difference-Between-MPLAB-Harmonyv3-Synchronous-and-Asynchronous-Drivers-and-When-to-Use-DS90003269A.pdf
- MPLAB Harmony v3 Quick Docs provides standalone help pages for users to get started developing applications on Microchip's 32-bit SAM and PIC® MCUs. The offline documentation is available in the "quick_docs" repository downloaded on a PC. Path: <your download location>/quick_docs/docs/index.html. The online pages are available at microchip-mplab-harmony.github.io/quick_docs/.
- For additional information on MPLAB Harmony v3, go to the Microchip web site: www.microchip.com/mplab/mplab-harmony and microchipdeveloper.com/harmony3:start

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **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 (FAQs), technical support requests, online discussion groups, Microchip design partner 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

Product Change Notification Service

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

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: 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 specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "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 is provided for the sole purpose of designing with and using Microchip products. Information 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.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". 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 ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION. 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, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL 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, BlueSky, 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, KleerNet, KleerNet logo, memBrain, Mindi, MiWi, 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.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks 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.

ISBN: 978-1-5224-7189-9

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

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: 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: 45-4485-5910 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: 49-7131-72400 **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