Relay Fault Detection with the PIC16F15245 and the Functional Safety Class B Libraries

AN5710

www.microchip.com Product Pages: PIC16F15244, PIC16F15245

Introduction

Author: Robert Perkel, Microchip Technology Inc.

Relays are electromechanical devices that use the magnetic field of their coil(s) to open or close electrical connection(s). This can create physical isolation between two points, which has advantages over semiconductor-based solutions. However, since a relay is a moving part, it can fail in several ways, from contacts welding together to burning out the energizing coil. Safety-critical applications can use special relays, like those with force-guided contacts, to detect failures, but these components are cost-prohibitive in many cases. Applications that need a higher level of reliability, but are not in sensitive applications, may consider a simple failure detector circuit instead. In this application note, a PIC16F15245 microcontroller monitors and drives a relay while implementing Microchip's free-to-use IEC 60730 Class B software diagnostic library.

Click to view code examples on MPLAB DISCOVER

1. Failure Modes

1.1 Relay Failures

While there is a wide range of failure modes for relays, this application note looks at the following ones:

- Open Circuit/Closed Circuit
- Brown-out

Most of these faults appear as a difference between the detected and expected voltage. In other words, if the relay coil is not being driven, but voltage is on the output, the relay has presumably failed short through some means. This can be expanded upon to include brown-out conditions on the output.

1.2 Drive Transistor Failure

An external transistor is used to switch the relay position. If the transistor fails short, these failures could be detected by matching the internal output value of the I/O pin (LATx) with the actual value of the I/O pin (PORTx). For instance, if the internal output value of LATx is '1', but the transistor is shorted to ground, then the output state seen by the digital logic (PORTx) is '0'.

CAUTION If the relay coil is driven by a higher voltage than the microcontroller, a short between the drain and gate could expose the microcontroller to a voltage that exceeds its absolute maximum rating. If this failure mode is anticipated, current-limiting resistors, Zener diodes and/or other protection devices can protect the I/O pins.

1.3 Microcontroller Monitoring

The Class B Functional Safety libraries ensure the microcontroller operates correctly. These libraries are free to use and a part of MPLAB® Code Configurator (MCC). MCC is a graphical tool used to configure the hardware peripherals on the microcontroller.

WARNING This example is not certified for functional safety. It is intended to illustrate the usage of the Class B libraries, but a detailed failure mode analysis is required before utilizing this circuit in an application.

The Class B libraries are designed for IEC 60730. This application implements the following tests from the Class B libraries to improve failure mode coverage:

- CPU
  - Verifies the CPU registers are operating as expected
- Flash
  - Scans the program Flash and verifies the memory has not been corrupted
- SRAM
  - Runs a Checkboard (or March) test pattern through the memory to verify the SRAM cells are functioning correctly
- Stack
  - Periodically tests the hardware stack to make sure it is operating correctly
- Watchdog Timer

- Tests the Watchdog Timer (WDT) on start-up to make sure the hardware module is working properly

Important: The SRAM test is only recommended with level S compiler optimizations.

2. Software Design

2.1 Relay States

The microcontroller runs a simple state machine to determine if the non-latching relay is operating correctly. There are five states defined:

- RELAY_OPEN
- RELAY_OPEN_TRANSITION_CLOSED
- RELAY_CLOSED
- RELAY_CLOSE_TRANSITION_OPEN
- RELAY_ERROR

2.1.1 Steady States

The RELAY_OPEN and RELAY_CLOSED states represent the relay being in steady state open or close.

2.1.2 Switching States

RELAY_OPEN_TRANSITION_CLOSED and RELAY_CLOSE_TRANSITION_OPEN occur when the system transitions from open to closed or closed to open. During these states, the usual voltage monitoring is not active. These states transition to RELAY_OPEN or RELAY_CLOSED when the minimum switching time has passed and the voltage on the output is above/below the threshold required.

2.1.3 Error States

The RELAY_ERROR state is set if an unexpected condition or malfunction is detected. All relay states can reach the error state. A few examples of the conditions that can cause this transition are shown below:

- Unexpected voltage during RELAY_OPEN
- Output did not transition during RELAY_OPEN_TRANSITION_CLOSED or RELAY_CLOSE_TRANSITION_OPEN
- Output voltage is invalid (brown-out)
- Microcontroller self-test failed

While in RELAY_ERROR, the coil is de-energized and the microcontroller does not switch the relay.

2.2 Error Monitoring

To keep track of the various error conditions, a bitfield indicates which errors have occurred. A list of the errors defined is in Table 2-1.

Table 2-1. Bitfield of Error Bits

Name	Bit Number	Description
Unused	7	N/A
Unused	6	N/A
ERROR_MEMORY_WRITE_FAIL	5	NVM Write Failure
ERROR_OUTPUT_BROWNOUT	4	Output voltage is between high and low thresholds
ERROR_ILLEGAL_STATE	3	An invalid error code or an invalid state is detected
ERROR_SELF_TEST_FAIL	2	Self-test failed
ERROR_TRANSISTOR_SHORT	1	LAT and PORT mismatch on the transistor

.....continued

Name	Bit Number	Description
ERROR_RELAY_STUCK	0	The relay fails to switch within the expected time

2.3 Voltage Monitoring

The core function of this example is to monitor the voltage on the output of the relay using the integrated Analog-to-Digital Converter (ADC). Since this output is likely at a higher voltage than the microcontroller, a voltage divider is used to scale the signal (see the Implementation chapter for more information). The ADC runs almost continuously by being triggered every time the ADC Result High (ADRESH) register is read.

When a new ADC sample is ready, the function `Relay_onADCReady` is called. This function reads the results (triggering the next conversion) and checks to see if the result is above the threshold (`ADC_THRESHOLD_HIGH`), below the threshold (`ADC_THRESHOLD_LOW`), or neither. Then, it looks at the current relay state to determine if the value makes sense.

For instance, if the relay is in the RELAY_CLOSED state, but the voltage is below the threshold, this would be considered a fault, as voltage is expected, but not present.

2.4 Periodic Self-Tests

Once per millisecond, the microcontroller runs a scan to look for detectable faults, to manage scheduled tasks such as running the Class B library self-tests and switching the relay, and to clear the Watchdog Timer.

2.4.1 Passive Fault Testing

Some faults are detectable without disturbing the system. The microcontroller tests the critical state machine variables to ensure they did not experience memory corruption, verifies the internal I/O pin output matches the state of the I/O pin at the switching transistor and also to ensure time has not expired during a state transition.

2.4.2 Task Scheduling

Since this function is called at regular intervals, it can be used as a time base for scheduling tasks, similar to a cooperative Real-Time Operating System (RTOS). The following tasks are scheduled:

- Automatic Relay Switching
- Functional Safety Self-Tests
- Functional Safety Memory Self-Tests

2.4.2.1 Automatic Relay Switching

For simplicity in this example, the microcontroller automatically switches the relay every five seconds. In a complete application, it is expected that the Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I^2C), or General-Purpose Input/Output (GPIO) would be used to signal when the relay needs to be switched.

2.4.2.2 Functional Safety Self-Tests

This task runs the Class B library self-tests for the CPU registers and one of the banks of the SRAM. Each time this test runs, the SRAM bank under test is changed.

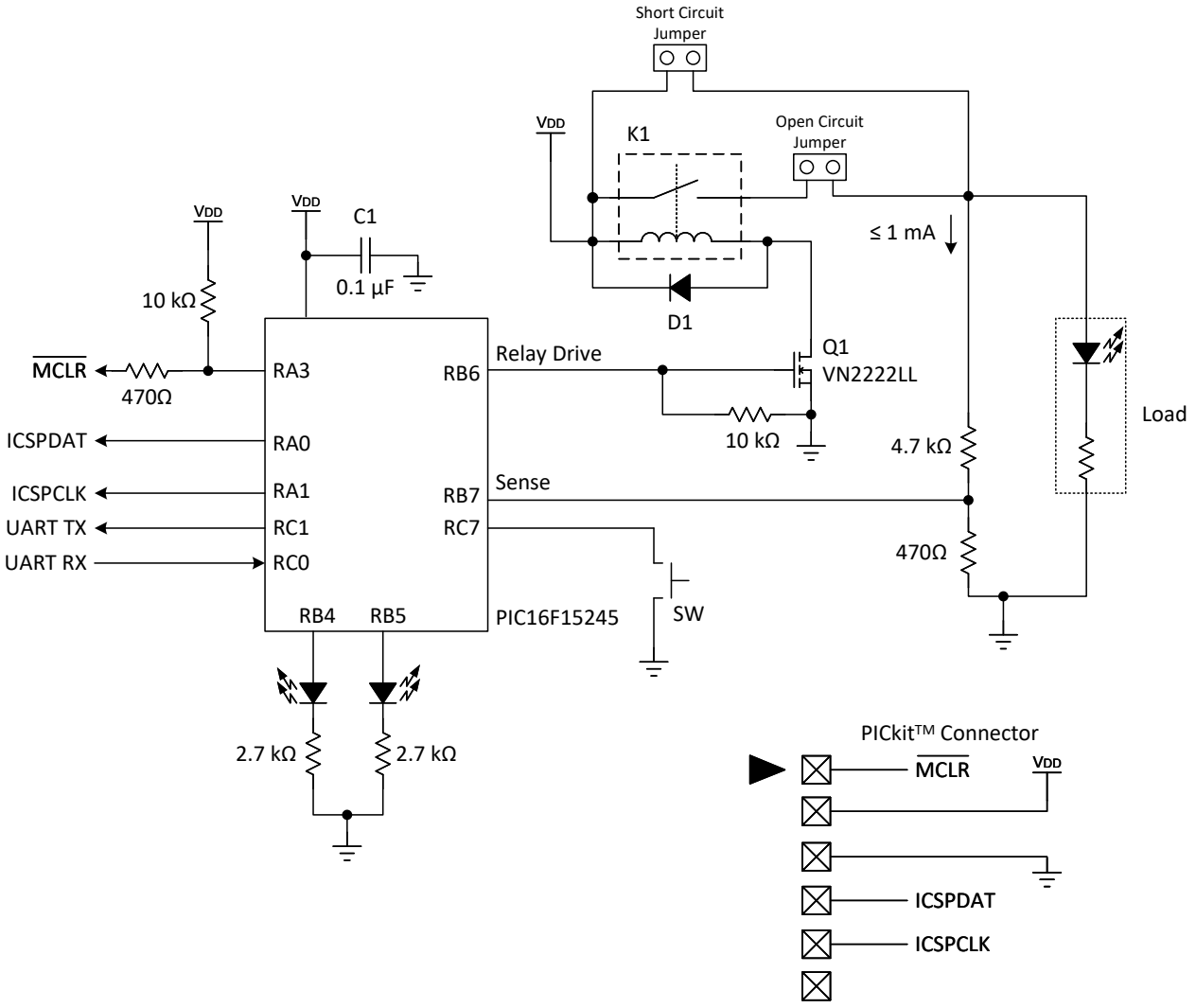
2.4.2.3 Functional Safety Memory Self-Tests

This task scans the microcontroller memory to ensure the Program Flash has not become corrupted and tests the hardware stack to ensure it functions correctly.

3. Implementation

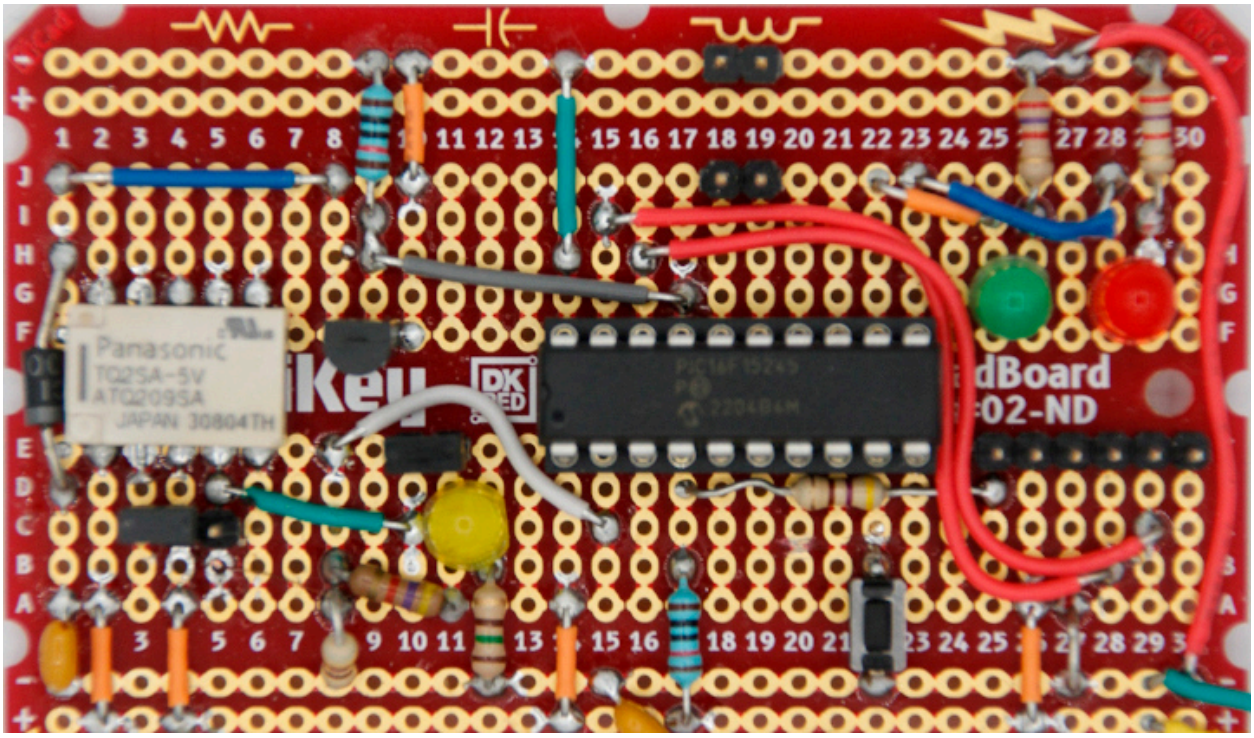
The circuitry required is dependent on the specifics of each application. The test circuit in this application used a 5V non-latching relay and a switched 5V signal divided down to make it operate similarly to a high-voltage application. Most applications will have different operating conditions. The following sub-chapters detail how to set up the circuit to work with different operating voltages. Figure 3-1 shows the reference circuit.

Figure 3-1. Schematic of the Test Circuit



A small protoboard was used to build a prototype of this circuit. Depending on the specific implementation, it may be possible to prototype using the Curiosity Development Board (DM164137) with fly-wired elements next to it. With the LPC, the amount of soldering and assembly is reduced due to the on-board programmer/debugger, USB-UART bridge, and LEDs. This implementation is available on MPLAB Discover.

Figure 3-2. Protoboard Implementation



Click to view code examples on MPLAB DISCOVER

3.1 Voltage Divider

The acquisition time is a fixed 2 µs on the PIC16F15244 microcontroller family. For best results, the capacitor must charge to 0.5 Least Significant Bits (LSb) within the 2 µs time. 1 LSb is defined as:

$$V_{LSB} = \frac{V_{REF}}{2^n}$$

Where V_{REF} is the voltage reference and n is the number of bits in the ADC result.

Note: n = 10 on the PIC16F15244 family.

Additionally, the impedance into the ADC is recommended to be 10 kΩ or less. For a voltage divider, the equivalent external impedance seen by the ADC is the top and bottom resistors in parallel:

$$R_{EQ} = \frac{1}{\frac{1}{R_{TOP}} + \frac{1}{R_{BOTTOM}}}$$

Note: This does not consider the microcontroller's internal resistances. See the ADC chapter of the device data sheet for more information.

If it is not possible to reduce the impedance of the resistor ladder, adding capacitance (≤ 33 nF) across the bottom resistor may help by providing a low-impedance source of charge for the ADC at the cost of response time. This is discussed in detail in Maximizing the Signal: Tips and Tricks to Properly Acquiring Analog Signals (DS00004225A).

Tip: Another solution for high voltages is to switch to another measurement topology (see the Isolated Sensing chapter).

3.2 Setting the Software Parameters

Two parameter sets must be set for each application – the voltage thresholds and relay timing.

3.2.1 Voltage Thresholds

The voltage thresholds define the minimum valid output levels. Voltages above and below the max and min are considered valid, with anything between the two considered a brown-out. These constants are defined as ADC_THRESHOLD_HIGH and ADC_THRESHOLD_LOW and represent the desired value as *seen by the ADC*. To convert a number into this format, the following formula can be used:

$$VALUE = \frac{V_{NUM}}{V_{REF}} \times 2^n$$

Where V_{NUM} is the desired value to convert, V_{REF} is the reference voltage of the ADC, and n is the resolution of the ADC (n = 10 on the PIC16F15245).

3.2.2 Relay Timing

This example monitors relay timing in two phases. The first phase is the mechanical open/close stage, where the contacts are moving and may not be at stable levels. During this phase, the voltage on the output is allowed to be at invalid or unstable levels. These parameters are specified in the relay data sheet as the operate time/release time and are entered in the program as the constants RELAY_CLOSE_TIME_MAX and RELAY_OPEN_TIME_MAX.

The next phase is the settling time. Capacitance, start-up currents, and other activities may cause the output to drop or be invalid during this phase, but once it has risen to a valid level, it is assumed that it has stabilized at this point. Once this occurs, the program switches into the RELAY_CLOSED or RELAY_OPEN state. The length of time in this phase is determined by circuit requirements and entered as RELAY_MARGIN.

3.3 Design Variations

Depending on the specific application circuit, there are cases where a different approach to monitoring the relay is needed. These sub-chapters contain some general guidance for modifications or tweaks to the circuit but were not tested or implemented.

3.3.1 Isolated Sensing

In some cases, the voltage on the output is too high to divide down reasonably. For these situations, isolated sensing might be a better option. The simplest approach is to use a (linear) optocoupler, which is comprised of an isolated LED and phototransistor. The light from the LED is absorbed by the phototransistor, changing the current through the transistor, which can be used to determine the current flow on the measurement side.

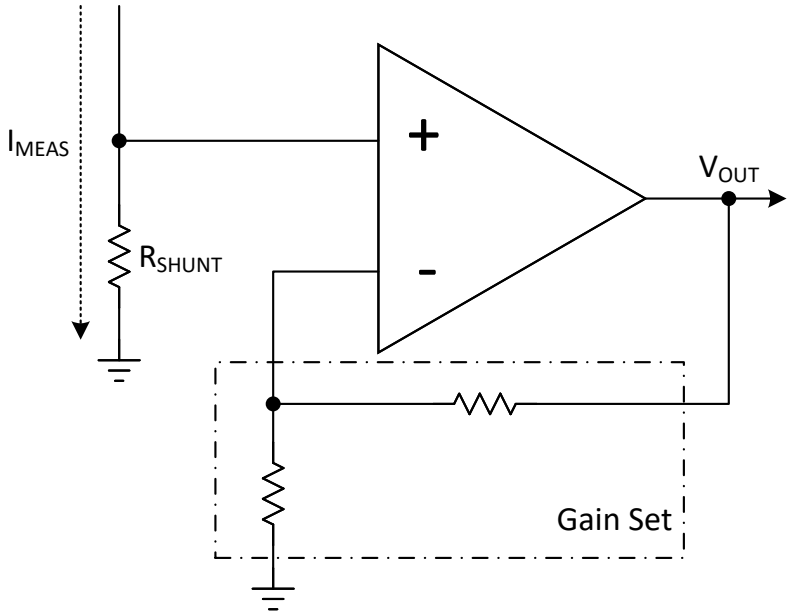
3.3.2 Coil Malfunction Detection

One failure mode not directly covered in this setup is a burned-out or shorted coil inside the relay.

Note: In many cases, this failure may cause the relay to switch incorrectly, which would be detectable.

The simplest way to detect this failure mode is to monitor the current flow through the coil. If the coil is open, then no (or minimal) current is flowing. If it is shorted, then too much current is flowing. A simple way of implementing a current amplifier is with an operational amplifier and a current shunt resistor, as shown in Figure 3-3.

Figure 3-3. Current Amplifier



3.3.3 Temperature Sensing/Monitoring

Another parameter to monitor is temperature. If the load is heating up, it may be necessary to disable the power. This can be accomplished using an NTC thermistor as part of a voltage divider. The output of the voltage divider is a function of the NTC thermistor's temperature. By obtaining the resistance versus temperature data from the manufacturer, a look-up table can be created for the microcontroller.

3.3.4 Supporting Latching Relays

The relay used in this example is a non-latching relay. However, some relays are latching, where the relay maintains the same state until reset. The coils cannot be driven continuously with this type of relay unless some changes are made to the program behavior. The simplest way to implement this is to disable the coils after the initial switching time is complete but before the max time has elapsed.

4. Conclusion

Detecting relay failures can be an important element in many applications. By off-loading the task of monitoring to a small microcontroller, such as the PIC16F15245, the larger system can handle more complex tasks, or in the case of cost-sensitive systems, a single small microcontroller may be sufficient. Then, to further increase the confidence level of the microcontroller, the freely available Class B functional safety libraries are used to improve the detection rate for failures inside the microcontroller.

5. Revision History

Document Revision	Date	Comments
A	11/2024	Initial document release

Microchip Information

Trademarks

The “Microchip” name and logo, the “M” logo, and other names, logos, and brands are registered and unregistered trademarks of Microchip Technology Incorporated or its affiliates and/or subsidiaries in the United States and/or other countries (“Microchip Trademarks”). Information regarding Microchip Trademarks can be found at https://www.microchip.com/en-us/about/legal-information/microchip-trademarks.

ISBN: 979-8-3371-0071-5

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications 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. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

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.

Microchip Devices Code Protection Feature

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

- 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, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip products are strictly prohibited and may violate the Digital Millennium Copyright Act.
- 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. Microchip is committed to continuously improving the code protection features of our products.