8-Bit MCU Family
C8051F85x/86x Errata

This document contains information on the errata of revision C of C8051F85x/86x.

For errata on older revisions, please refer to the errata history for the device. The device data sheet explains how to identify chip revision, either from package marking or electronically.

Errata effective date: September 16, 2016.

1. Errata Summary

Table 1.1. Errata Status Summary

Errata #	Designator	Title/Problem	Workaround Exists	Affected Revision	Fixed Revision
1	WDT_E101	Restrictions on Watchdog Timer Refresh Interval	Yes	C	—

2. Detailed Errata Descriptions

2.1 WDT_E101 – Restrictions on Watchdog Timer Refresh Interval

Description of Errata

If the Watchdog Timer (WDT) is enabled, firmware will periodically write an 0xA5 value to the WDTCN register to refresh the timer and prevent the watchdog reset from occurring. However, if firmware writes to WDTCN more than once during the same LFOSC0 clock period, the refresh signal may be canceled, resulting in an unintended watchdog reset when the timer expires.

Affected Conditions / Impacts

If firmware refreshes the watchdog more than once in the same LFOSC0 clock period, an unexpected watchdog reset can occur.

Workaround

Systems using the Watchdog Timer (WDT) should ensure that the WDT is refreshed no more than once per LFOSC0 clock period.

Firmware can do this by using timers to count LFOSC0 clock periods. There are three methods to accomplish this:

1. If Timer 3 is not already in use, set it up to capture on the LFOSC0 clock. In this mode, the value of the Timer 3 reload registers does not matter. Instead, the WDT refresh function should check for the 16-bit timer flag (TF3H) to be set in the reset watchdog function, which indicates that a capture event occurred. If the device has another timer that can capture on the LFOSC0 clock, then that timer may be used instead of Timer 3.

```
void refresh_wdt()
{
   // Only refresh if TF3H is set
   if (TMR3CN0 & (0x80))
   {
      WDTCN = 0xA5;
      TMR3CN0 &= ~(0x80);
   }
}
```

2. If any timer is already in use and is clocked from the LFOSC0 or LFOSC0/8, firmware can check the current count against the previous count to ensure at least one clock period has passed:

```
void refresh_wdt()
{
   static uint16_t last_TMR3 = 0;
   if (last_TMR3 != TMR3)
   {
      // At least 1 LFOSC0 falling edge has occurred since the last WDT refresh
      WDTCN = 0xA5;
      last_TMR3 = TMR3;
   }
}
```

3. If the application already has an accurate and reliable time base, use that timer to establish a minimum WDT refresh interval that is longer than one LFOSC0 clock period in duration, similar to method (2) above as appropriate.

See the Knowledge Base article on this errata for more information, including examples of these firmware workarounds: http://community.silabs.com/t5/8-bit-MCU-Knowledge-Base/WDT-E101-Title/ta-p/177029.

Note: The LFOSC0 does not halt while debugging. This can cause the timer overflow flag to be set more quickly than expected when debugging the watchdog refresh function.

Resolution

There is currently no resolution for this issue.

3. Revision History

3.1 Revision 0.2

September 16, 2016

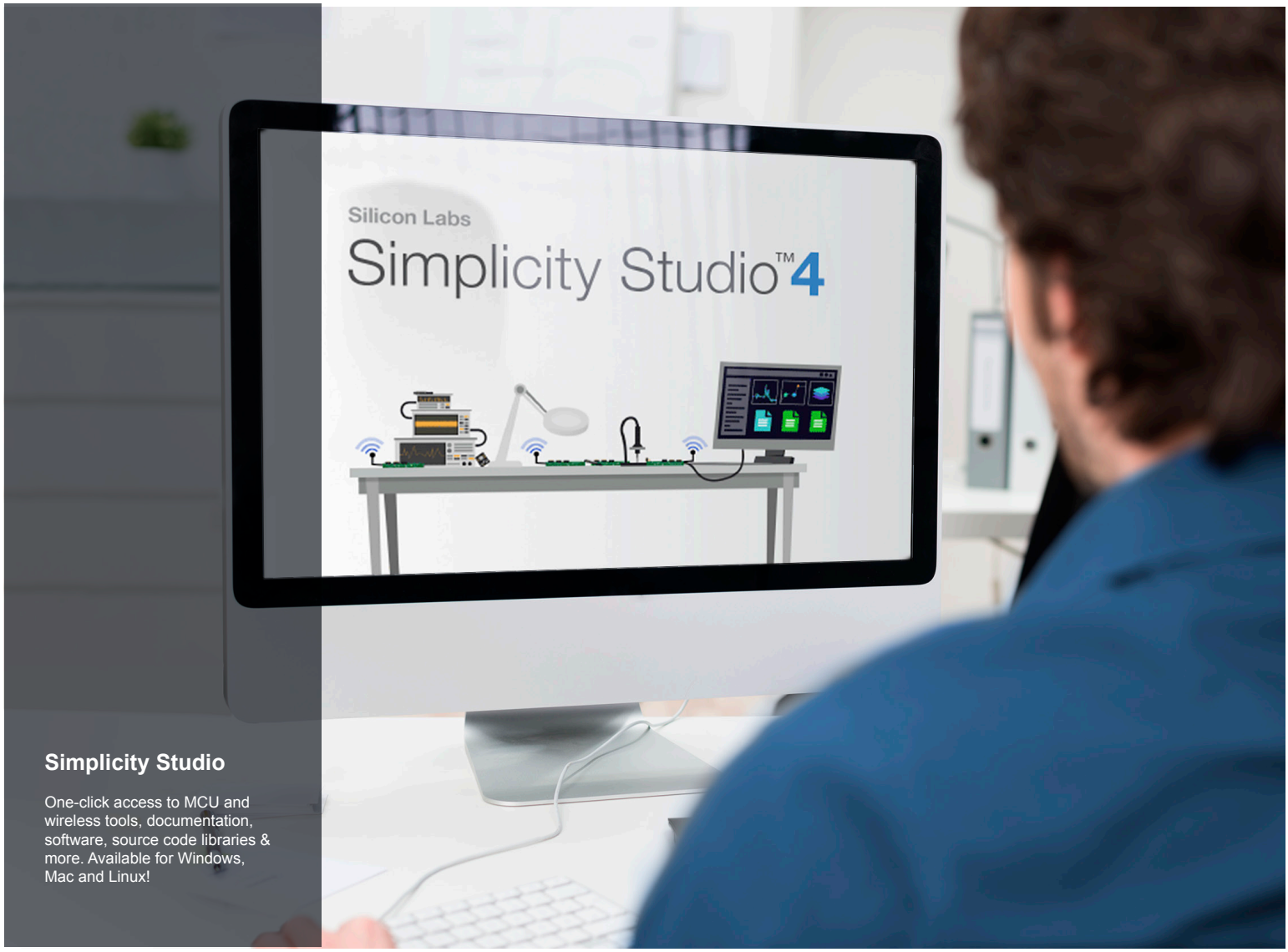
Moved UID_E101 and PKG_E103 from the errata to the errata history.

Added WDT_E101.

3.2 Revision 0.1

February 10, 2014

Initial release.



Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS®

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com