



Fail-Safe Clock Feature of the Z8 Encore! XP[®] Microcontroller

AN020102-0708



Abstract

This Application Note describes the fail-safe clock feature of Zilog's Z8 Encore! XP[®] F042A Series microcontroller unit.

- **Note:** *The source code file associated with this application note, AN0201-SC01.zip, demonstrates the fail-safe clock feature and is available for download at www.zilog.com.*

Z8 Encore! XP F042A Series Flash Microcontrollers

Zilog's Z8 Encore![®] products are based on Zilog's new eZ8[™] CPU and introduce Flash Memory to Zilog's extensive line of 8-bit microcontrollers. Flash Memory in-circuit programming capability allows for faster development time and program changes in the field. The high-performance register-to-register based architecture of the eZ8 core maintains backward compatibility with Zilog's popular Z8 MCU. Z8 Encore! MCUs combine a 20 MHz core with Flash Memory, linear-register SRAM, and an extensive array of on-chip peripherals.

The Z8 Encore! XP F042A Series of devices support up to 4 KB of Flash Program Memory and 1 KB register RAM. An on-chip temperature sensor allows temperature measurement over a range of -40 °C to +105 °C. These devices include two enhanced 16-bit timer blocks featuring PWM and Capture and Compare capabilities. An on-chip Internal Precision Oscillator (5 MHz/32 kHz) can be used as a trimmable clock source requiring no external components.

The Z8 Encore! XP devices include 128 Bytes of Non-Volatile Data Storage (NVDS) memory where individual bytes can be written or read. The full-duplex UART, besides providing serial communications and IrDA encoding and decoding capability, also supports multidrop address processing in hardware.

The rich set of on-chip peripherals make the Z8 Encore! XP MCUs suitable for various applications including motor control, security systems, home appliances, personal electronic devices, and sensors.

Discussion

The Z8 Encore! XP Series of devices use the following five possible user-selectable clocking schemes:

- On-chip precision trimmed RC oscillator
- On-chip oscillator using off-chip crystal or resonator
- On-chip oscillator using external RC network
- External clock drive
- On-chip low precision Watchdog Timer (WDT) oscillator

You must select one of the above oscillators by writing appropriate hex values to the Oscillator Control register (OSCCTL).

The Z8 Encore! XP Series of devices generate an interrupt when the primary oscillator fails, provided the fail-safe clock feature is enabled.

To maintain system functionality in this condition, the clock failure recovery circuitry automatically forces the WDT oscillator to drive the system clock. The WDT oscillator must be enabled to allow the recovery. Although this oscillator runs at a much lower frequency than the original system clock, the CPU continues to operate as before. You can add the software routines to the clock failure vector to provide either a remedy for the oscillator failure or issue a clock failure alert.

This automatic switch-over is not available if the WDT is the primary oscillator. It is also unavailable if the WDT oscillator is disabled. The following section discusses the software implementation of the fail-safe clock option.

► **Note:** *For more details on the OSCCTL and the WDT peripheral, refer to the Z8 Encore! XP[®] F082A Series Product Specification (PS0228).*

Developing the Application with Z8 Encore! XP MCU

This section provides a detailed description of the software implementation and demonstration of the fail-safe clock feature.

Software Implementation

System clock selection, GPIO and timer initialization, and the oscillator failure interrupt routine are discussed in the following sections:

- [Selecting the System Clock](#)
- [Initializing the GPIO](#)
- [Initializing the Timer](#)
- [Oscillator Failure Interrupt Routine](#)

Selecting the System Clock

The `init_external_systemclock()` function selects the type of oscillator used with the Z8 Encore! XP F042A Series MCU.

The Z8 Encore! XP F042A Series Development Board uses an 8 MHz off-chip ceramic resonator. The Oscillator Control register (OSCCTL) must be unlocked before writing. Writing the two-step sequence E7h followed by 18h unlocks the Oscillator Control register (OSCCTL). The register automatically locks on successful completion of a register write to the OSCCTL.

The following sequence writes to the Oscillator Control register (OSCCTL):

```
OSCCTL = 0xE7; // Unlock sequence for
              // OSCCTL write
OSCCTL = 0x18;
OSCCTL = 0x72;
```

With the above setting, the Internal Precision Oscillator (IPO) is enabled, the WDT oscillator is enabled, failure detection and recovery of the primary oscillator is enabled, and the external RC oscillator is enabled as the system clock.

Initializing the GPIO

The Z8 Encore! XP F042A Series Development Board used to implement the fail-safe clock option has three LEDs — D2, D3, and D4. In this example, the GPIO pins are configured as output pins to switch the D2, D3, and D4 LEDs ON or OFF.

The D4 LED is connected to Port C, Pin 3 (PC3), the D3 LED is connected to Port A, Pin 7 (PA7), and the D2 LED is connected to Port A, Pin 6 (PA6). For schematic details on these connections, refer to the *Z8 Encore! XP[®] F042A Series Development Kit User Manual (UM0166)*, available for download at www.zilog.com.

After these connections are made, the `init_led_gpio()` function initializes the GPIO ports. On the target board, PA6 and PA7 control the D2 and D3 LEDs, respectively, and PC3 controls the D4 LED. As a result of the initialization, each of these port pins are configured as output ports.

Port C pins provide a current sourced output capable of driving an LED without requiring an external resistor. The current output is programmed to source 13 mA. This mode is enabled through the Alternate Function subregisters. For details on the Alternate Function subregisters, refer to the *Z8 Encore! XP[®] F082A Series Product Specification (PS0228)*.

Initializing the Timer

Timer0 is initialized to generate an interrupt every 4.05 ms with an external oscillator frequency of 8 MHz. The `init_timer0(void)` function loads appropriate values to the T0H, T0L, T0RH and T0RL registers, loads the prescale value, and selects the timer-operating mode.

The interrupt service routine (ISR) for the timer `void isr_timer0(void)` sets the ProgStat variable to RUN mode.

Oscillator Failure Interrupt Routine

The `SET_VECTOR (POTRAP, Oscillator_Failure)` function called in the main routine initializes the vector location PORTAP (003Ah) with the address of the `oscillator_failure(void)` interrupt routine.

When the primary oscillator fails, the program jumps to the `oscillator_failure(void)` ISR, which disables the external system clock and sets the internal precision clock either to 5.5 MHz or 32 kHz based on the `#define` statement in the project setting. Then, the program execution continues as before. The details of this operation is demonstrated in the next section.

Demonstrating the Application

This section provides details on the equipment used and the procedure to demonstrate the fail-safe clock feature.

Equipment Used

The equipment used to demonstrate the fail-safe clock feature include:

- Z8 Encore! XP Development Board
- PC with HyperTerminal application

Procedure

Follow the steps below to demonstrate the fail-safe clock feature:

1. Download and extract the `AN0201-SC01.zip` file.
2. Launch ZDS II-Z8 Encore![®] and open the `fail-safe clock` project from the extracted zip folder.
3. Click **Project Settings**. The **Project Settings** dialog box is displayed.
4. Select the **Preprocessor** category from the **C** tab.
5. Add the `_LOW_FREQ` definition to the **Preprocessor Definitions** field. This definition is required to select 5.5 MHz as the frequency of the internal precision oscillator. [Figure 1](#) displays the **Preprocessor Definitions** field in the **Project Settings** dialog box.

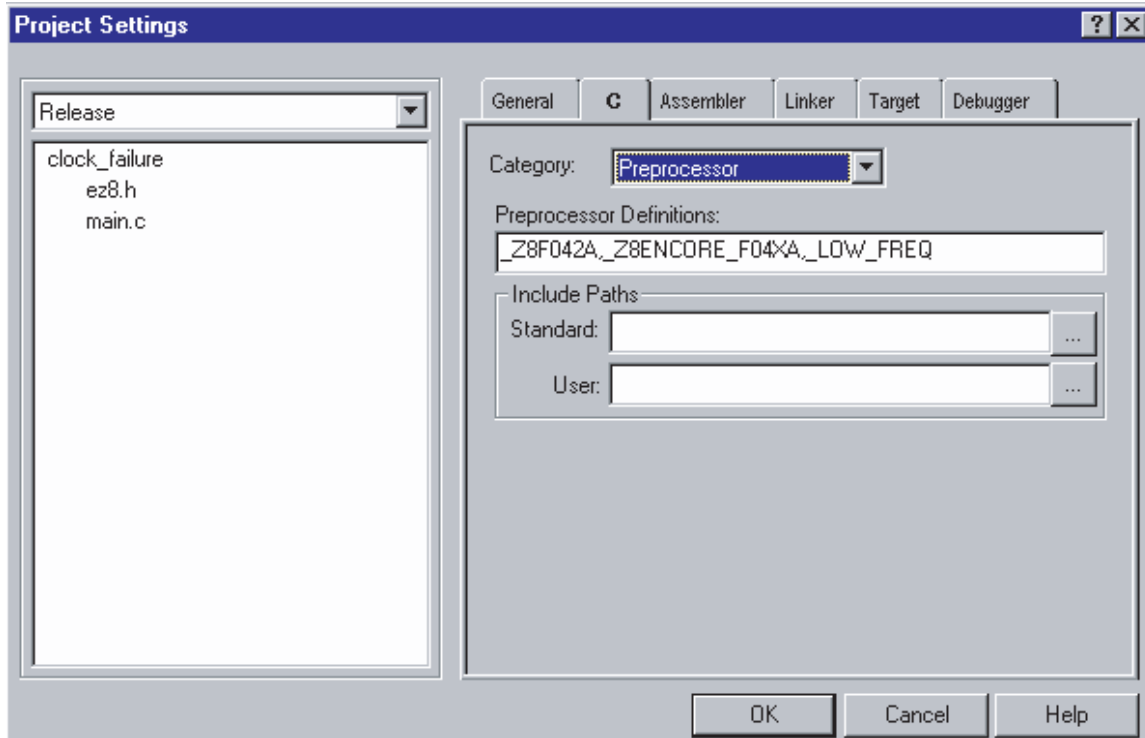


Figure 1. Preprocessor Definitions in Project Settings Dialog Box

6. Compile and build the code.
7. Using ZDS II, download the .1od output file to the target Z8 Encore! XP[®] Development Board.
8. Execute the code.
9. Observe that the D2, D3, and D4 LEDs flash sequentially.
10. Short Pin 7 of the Z8 Encore! XP MCU (U5) to GND to create a clock failure condition. As a result, an oscillator failure interrupt is immediately generated and the IPO begins to function.
11. Observe that the rate at which the three LEDs flash is reduced when compared to the observation made in [Step 9](#).
12. Remove the short between Pin 7 of U5 and ground.
13. To test the fail-safe clock feature at the IPO frequency of 32 kHz, open the project and remove the `_LOW_FREQ` definition in the **Preprocessor Definition** field. As a result of this modification, 32 kHz is selected as the frequency of the IPO.
14. Recompile and build the project. Download the .1od output file to the target board and execute the program.
15. Short Pin 7 of U5 to ground, and observe the frequency at which the LED flashes. The rate at which these LEDs flash must be greatly reduced compared to the observations made in [Step 9](#) and [Step 11](#).

Summary

This Application Note discusses the fail-safe clock feature of the Z8 Encore! XP[®] F042A Series microcontroller. With this fail-safe clock feature enabled, both program execution and system operation continue even if the external oscillator fails. You can generate an audiovisual alarm that indicates the condition of the primary clock failure. As a result, You can decide whether to allow the system to operate at a lower frequency, or to switch the system off for maintenance or troubleshooting purposes.

References

The documents associated with eZ8 CPU, Z8 Encore! XP F042A Series, and ZDS II available on www.zilog.com are provided below:

- eZ8[™] CPU User Manual (UM0128)
- Z8 Encore! XP[®] F082A Series Product Specification (PS0228)
- Z8 Encore! XP[®] F042A Series Development Kit User Manual (UM0166)
- Zilog Developer Studio II-Z8 Encore![®] User Manual (UM0130)



Warning: DO NOT USE IN LIFE SUPPORT

LIFE SUPPORT POLICY

ZILOG'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF ZILOG CORPORATION.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

©2008 by Zilog, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. ZILOG, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. ZILOG ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering.

Z8, Z8 Encore!, and Z8 Encore! XP are registered trademarks of Zilog, Inc. eZ8 is a trademark of Zilog, Inc. All other product or service names are the property of their respective owners