Application Note

An Automatic Temperature Control System Using RZK

AN019902-0908

Abstract

This application note demonstrates how an application running on Zilog’s Real-Time Kernel (RZK) can be used to control various devices to maintain a temperature. This temperature control application reads a value from a temperature sensor and determines when to switch a fan or bulb off or on according to minimum and maximum temperature limit settings. These settings can be changed using upper and lower limit set switches. The limit settings and the current temperature are displayed on an LCD panel.

Note: *The source code file associated with this application note,* `AN0199-SC01`*.zip is available for download at* www.zilog.com.

RZK Overview

The RZK is a real-time, preemptive, multitasking kernel designed for time-critical embedded applications. It can be used with Zilog’s eZ80Acclaim!® family of microcontroller unit (MCU) and microprocessors.

A real-time multitasking kernel also called a real-time operating system (RTOS) is a software that ensures that time-critical events are processed efficiently. The use of an RTOS generally simplifies the design process of a system by allowing the application to be divided into multiple independent elements called tasks.

Features of RZK

The RZK functional architecture is configurable, scalable, and modular in design, and provides a rich set of features and easy-to-use APIs. RZK features are tuned to the stringent memory and performance requirements of the 8-bit domain.

RZK is compact with minimal footprint and can be accommodated in the Flash area of the target processor. RZK allows rapid context switching between Threads, in addition to offering quick interrupt responses. It features preemptive, priority-based, multitasking scheduler. It also provides timing support for delays, time-outs, and other periodic events. You can take advantage of RZK's time-slicing option with adjustable time slices. In addition to various configuration options, RZK provides priority inheritance facility.

RZK Objects

RZK objects used for real-time application development are Threads, message queues, event groups, semaphores, Timers, partitions and regions (memory objects), and interrupts. RZK objects use the kernel services for resource management and provide a set of APIs as an interface to the application. Based on its requirements, an application can use the objects optionally.

This application note focuses on the use of RZK’s Threads.

Copyright ©2008 by Zilog®, Inc. All rights reserved.
www.zilog.com

Thread Control APIs

Thread APIs are used to create Threads and perform the following operations on a created Thread:

- Suspend a Thread finitely or infinitely
- Resume a Thread that suspends infinitely
- Delete a Thread
- Change the priority of a Thread
- Yield control to other Threads
- Get Thread parameters

For more information about RZK, refer to the *Zilog Real-Time Kernel Reference Manual (RM0006),* available for download at www.zilog.com. It is also available along with RZK software.

eZ80Acclaim!® Overview

eZ80Acclaim! ® on-chip Flash Microcontrollers are an exceptional value for designing high-performance, 8-bit MCU-based systems. Its possible to obtain performance necessary to execute complex applications quickly and efficiently with speed up to 50 MHz and an on-chip Ethernet MAC (eZ80F91 MCU only). Combining Flash and SRAM, eZ80Acclaim! devices provide the memory required to implement communication protocol stacks and achieve flexibility when performing in-system updates of application firmware.

eZ80Acclaim! Flash MCUs can operate in full 24-bit linear mode, addressing 16 MB without a Memory Management Unit (MMU). Additionally, support for the Z80®compatible mode allows you to execute Z80/Z180 devices legacy code within multiple 64 KB memory blocks with minimum modification. With an external bus supporting eZ80®, Z80, Intel, and Motorola bus modes and a rich set of serial communication peripherals, there are several options when interfacing to external devices.

Some of the applications suitable for eZ80Acclaim! devices include vending machines, point-of-sale (POS) terminals, security systems, home and building automation, communications, industrial control and facility monitoring, and remote control.

Discussion

This section provides the hardware and software implementation details for developing the automatic temperature control system using RZK.

Hardware Architecture

Figure 1 is a block diagram of the hardware architecture featuring the eZ80F91 Development Board. It has a temperature sensor for reading temperature, fan for cooling the sensor, bulb for heating the sensor, switches for setting the upper and lower temperature limits, and the Character LCD module for displaying the current temperature, upper, and lower limit.

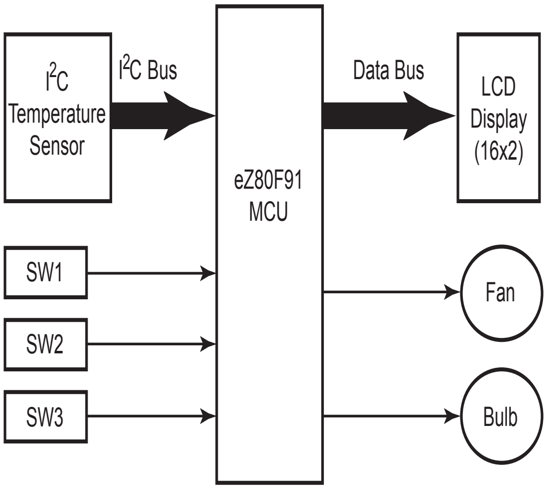

Figure 1. Hardware Block Diagram of an Automatic Temperature Control System Using RZK

Figure 2 displays the connections between the eZ80F91 MCU and a thermostat board. The data bus is connected to the Character LCD Module. Port pins PB0, PB1, and PB2 are connected to switches SW1, SW2, and SW3. Pins PB3 and PB7 are connected to the lamp and fan respectively. The MAX6625 temperature sensor is connected to the I^2C bus.

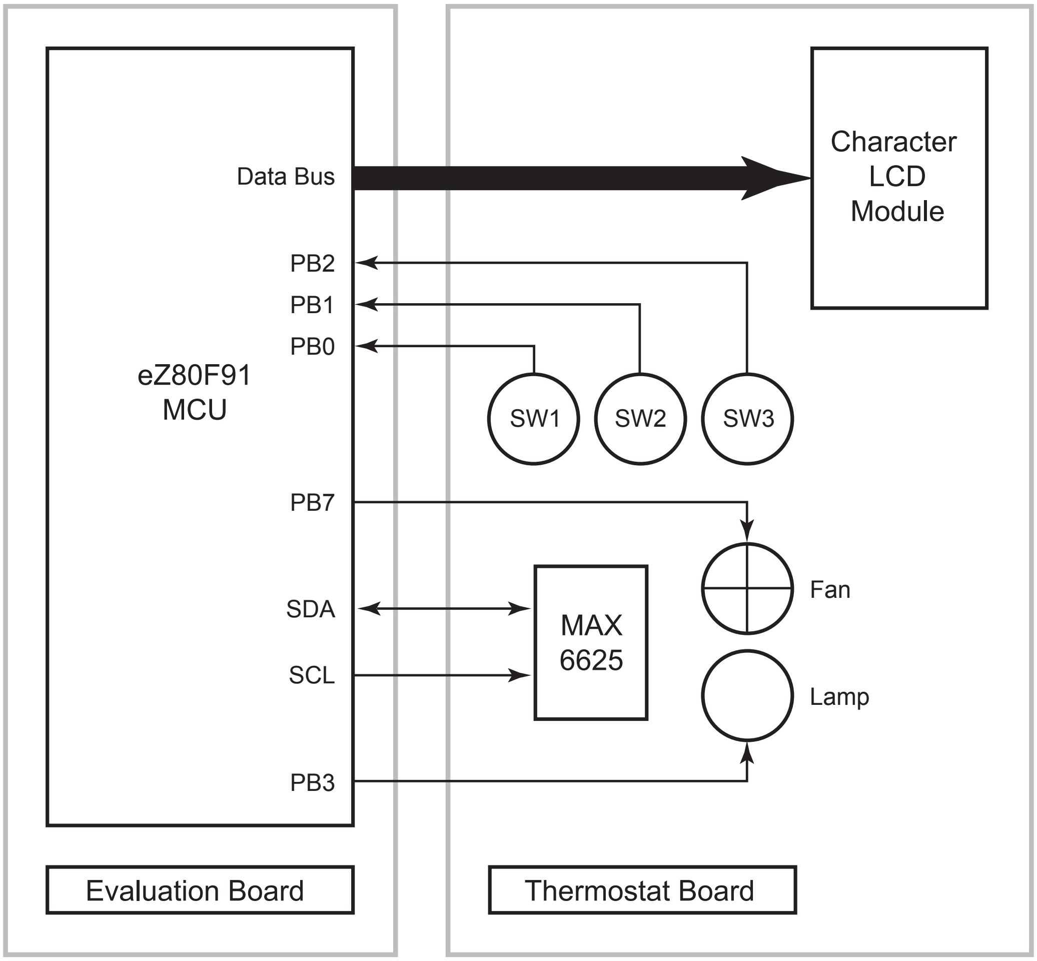

Figure 2. Connection Between the eZ80F91 Development Platform and the Thermostat Board

Software Implementation

The software implementation for the automatic temperature control system provides the functionality to maintain a temperature within a specified limit. The main functions provided by this application are listed below:

- Automatic fan ON/OFF
- Automatic bulb ON/OFF
- Set lower and upper limits by pressing a switch
- Read a temperature from a temperature sensor
- Display the current temperature and the lower and upper limits on LCD

Complete functionality is managed by the following four functions, in order of priority from high (#1) to low (#4):

```
1. RZKTempReadTask (4)
2. TempControlTask (8)
3. TempDisplayTask (12)
4. RZKKeyControlTask (16)
```

These functions are executed according to their priority.

The automatic ON/OFF of the bulb and fan, and the setting of the lower and upper limits is controlled by the **TempControlTask**. If the temperature read by **RZKTempReadTask** is greater than the set upper limit, then this task switches off the bulb and switches on the fan. If the read temperature is lower then the set limit, **RZKTempReadTask** switches the bulb on and switches off the fan.

RZKTempReadTask()

This function reads the current temperature from I^2C temperature sensor.

TempControlTask()

This function perform the following functions:

1. Set the upper and lower limit.
2. Upload the current temperature, upper, and lower limit to display array.
3. Compare the upper and lower set limit with current temperature and Switch ON/OFF the fan/bulb accordingly.

TempDisplayTask()

This function reads and updates the temperature on LCD display. It displays the current temperature, lower, and upper limit of temperature. The main operations performed by this function are read display buffer and update the display with current temperature with lower and upper limit

RZKKeyControlTask()

This function scans the switches for setting the lower and upper limit of temperature. The main operations performed by this function are:

1. Scan the switches.
2. If switch SW1 is pressed, decrease the lower limit (LL).
3. If switch SW2 is pressed, decrease the upper limit (UL).
4. If switch SW1 and SW3 are pressed, increase the lower limit (LL).
5. If switch SW2 and SW3 are pressed, increase the upper limit (UL).

RZKApplicationEntry()

In addition to the four tasks described above, this fifth RZK function is the main entry point for any application into RZK. The application program entry function performs the following operations:

1. Initializes all peripherals.
2. Creates a function for reading the temperature from the I^2C temperature sensor, controlling the temperature within a specified limit, and displaying the current temperature and limits on the LCD panel.
3. Resumes all functions.

Adding an Application Project to a Standard RZK Kernel

This section describes adding the automatic temperature control system project to the standard RZK kernel.

RZK software is available for download at www.zilog.com and can be downloaded to a PC upon user registration. RZK can be installed in any location as specified from its default location `C:\Program Files\Zilog`

Follow the steps below to add the demo files to the RZK directory:

1. Download RZK. Browse to the location where RZK is downloaded. Open the `..\eZ80F91\Sample Programs` folder.
2. Download the `AN0199-SC01.zip` file and extract its contents to a folder on your PC. Note that there is a folder labelled `\ATCS` within the extracted folder.
3. Copy this `\ATCS` folder to the `..\eZ80F91\Sample Programs` folder. Note that there are two folders within the `\ATCS` folder. These folders are:
   - `\IntFlash`, which contains the `ATCS.pro` file to be downloaded into the MCU's internal Flash memory.
   - `\RAM`, which contains the `ATCS.pro` file to be downloaded into the MCU's RAM memory.
4. Launch ZDS II and open the project file `ATCS.pro`, which is located in the following filepath:

```
..\eZ80F91\Sample Programs
\ ATCS \IntFlash or RAM
```

Note: *Do not modify any of the project settings in any of the project files.*

Testing

This section describes how the automatic temperature control system application can be tested.

Test Setup

Figure 3 displays the connection between a PC, a LAN/WAN, and the eZ80® Development Platform.

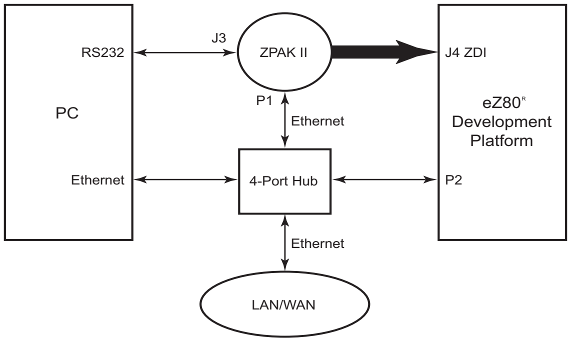

Figure 3. Test Setup for an Automatic Temperature Control System Using RZK

Equipment Used

eZ80F91 Development Kit

HyperTerminal Settings

Set the HyperTerminal application to reflect the following values:

- 57.6 kbps
- 8–N–1 protocols
- A flow control of none

Test Procedure

Follow the steps below to connect the hardware and build your project:

1. Connect the eZ80® Development Platform via the ZPAK II unit to the PC and to the Ethernet as displayed in Figure 3 on page 5.
2. To prepare for downloading the program, open HyperTerminal to set the IP address for ZPAK II.
3. Hold down the **Z** key and press the **Reset** button on the eZ80 Development Platform to launch the menu for setting the IP address. Follow the menu instructions to set the IP address for ZPAK II.
4. Launch ZDS II for eZ80 Acclaim!® v4.7.0. Open the RTC project located in the either of the following two filepaths:

   ```
   \\RZK 1.0.0\eZ80F91\Sample Pro-
   grams\ATCS\RAM\ATCS.pro
   ```

   ```
   \\RZK 1.0.0\eZ80F91\Sample Pro-
   grams\ATCS\Flash\ATCS.pro
   ```

5. Build the project and download it to the eZ80F91 MCU using ZDS II.
6. Run the project.
7. Observe the temperature on the LCD display.

Changing the Lower and Upper Limits

To change the lower limit, follow these instructions:

1. Press switch SW1 to decrease the lower limit on the eZ80 Development Platform.
2. Press switches SW1 and SW3 simultaneously to increase the lower limit.

To change the upper limit, follow these instructions:

1. Press switch SW2 to decrease the upper limit on the eZ80 Development Platform.
2. Press switches SW2 and SW3 simultaneously to increase the upper limit.

Test Results

The Automatic Temperature Control System application successfully runs on the eZ80F91 MCU, by setting the upper and lower limits functions satisfactorily.

Summary

This application note highlights the capability of RZK to control devices such as a fan or a bulb and read a temperature through the I^2C bus on the eZ80F91 MCU. There are four functions running in an infinite loop to control the entire operation.

This embedded control system can be used to control heavy motors, heaters, and other devices used in control systems.

References

The documents associated with eZ80F91 MCU, eZ80Acclaim!, ZTP/RZK, and ZDS II–IDE available on www.zilog.com are provided below:

- eZ80F91 MCU Product Specification (PS0192)
- Zilog Developer Studio II—eZ80Acclaim!® User Manual (UM0144)
- RZK Developer’s Kit Quick Start Guide (QS0027)
- Zilog Real-Time Kernel User Manual (UM0075)
- Zilog Real-Time Kernel Reference Manual (RM0006)
- eZ80® CPU User Manual (UM0077)

Appendix A—Flowchart

This appendix lists flowcharts for the application entry and key scan tasks performed in the automatic temperature control system implementation described in this application note.

Figure 4 displays the flow for the application entry Thread/function.

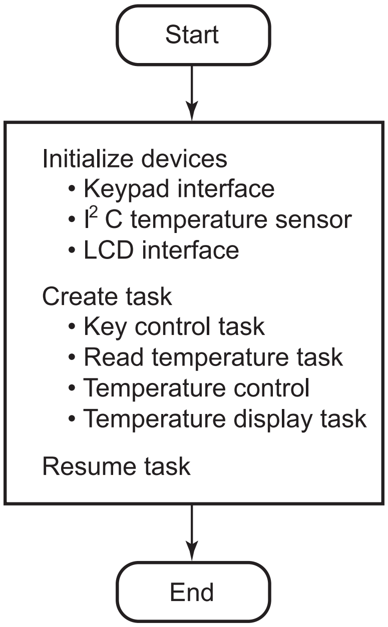

Figure 4. Application Entry Thread Flow

Figure 5 displays the flow of the key scan tasks.

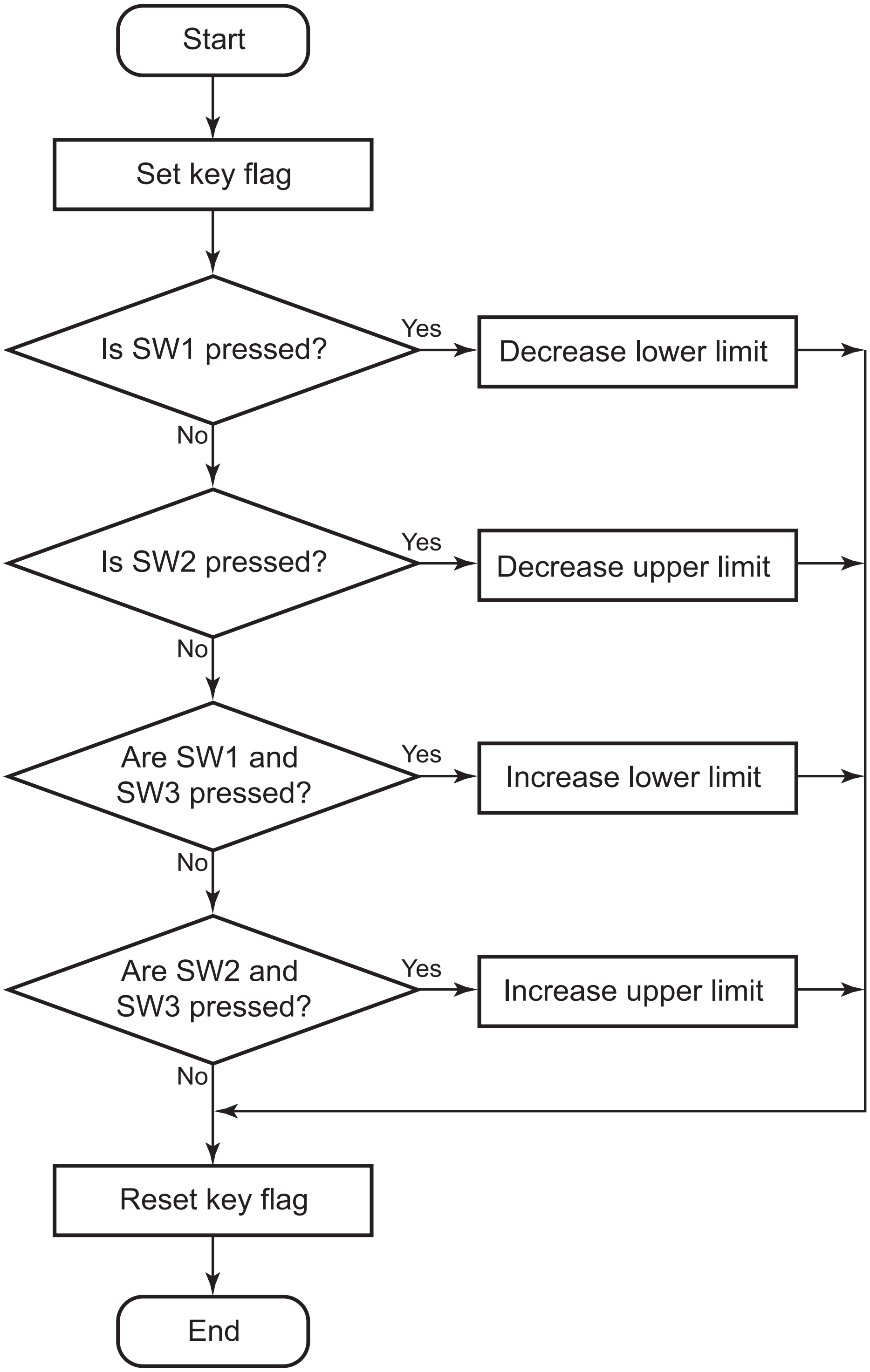

Figure 5. Key Scan Task Flow

⚠ **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.

eZ80, Z80, and eZ80Acclaim! are registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.