



Abstract

This application note describes how to use Zilog’s eZ80F91 MCU as a webserver to send emails using the Simple Mail Transfer Protocol (SMTP). The SMTP is provided by Zilog TCP/IP Software Suite (ZTP). The mail () function of the standard ZTP sends the SMTP mail message to the specified SMTP Server/Port.

The source code file associated with this application note, AN0207-SC01.zip, is available for download at www.zilog.com.

► **Note:** *The source code associated with this document is intended for use with the ZTP v2.1.*

Zilog® Product Overview

This section provides a brief overview of Zilog products used in this application note, which includes the eZ80AcclaimPlus!TM microcontrollers and the full-feature ZTP software suite.

eZ80AcclaimPlus! MCU Family Overview

The eZ80AcclaimPlus! family of microcontrollers includes Flash and non-Flash products. The Flash-based eZ80AcclaimPlus! MCUs, device numbers eZ80F91, eZ80F92, and eZ80F93, are an exceptional value for customers designing high performance embedded applications.

With speeds up to 50 MHz and an on-chip Ethernet MAC (eZ80F91 only), you can execute complex applications supporting networking functions quickly and efficiently. Combining on-chip Flash and SRAM, eZ80AcclaimPlus! devices provide

the memory required to implement communication protocol stacks and achieve flexibility when performing in-system updates of application firmware.

Zilog also offers two eZ80® devices without Flash memory: eZ80L92 and eZ80190 microprocessors.

ZTP Overview

ZTP integrates a rich set of networking services with an efficient real-time operating system (RTOS). The operating system is a compact pre-emptive multitasking, multi threaded kernel with inter-process communications (IPC) support and soft real-time attributes.

Table 1 lists the standard network protocols implemented as part of the embedded TCP/IP protocol stack in ZTP.

Table 1. Standard Network Protocols in ZTP

HTTP	TFTP	SMTP	Telnet	IP	PPP
DHCP	DNS	TIMEP	SNMP	TCP	UDP
ICMP	IGMP	ARP	RARP	FTP	

Many TCP/IP application protocols are designed using the client-server model. The final stack size is link-time configurable and determined by the protocols included in the build.

Discussion

SMTP is a simple ASCII protocol. After establishing the TCP connection to Port 25, the operating client (request-sending machine) sends the request and waits for the operating server (request-receiving machine) to answer. The server sends a line of



text providing its identity and readiness to receive email. If the server is not ready, the client releases the connection and tries again later. After the connection is established, the client provides the information about sending and receiving the email. If such a recipient exists at the destination, the server provides the client with the go-ahead message.

The client then sends the actual message and the server acknowledges it. Checksums are generally not required because the TCP provides a reliable byte stream. If more than one email is to be sent, the connection is released only after all the emails are exchanged in both directions. ZTP functions only as a client, with a message body size of 32 K.

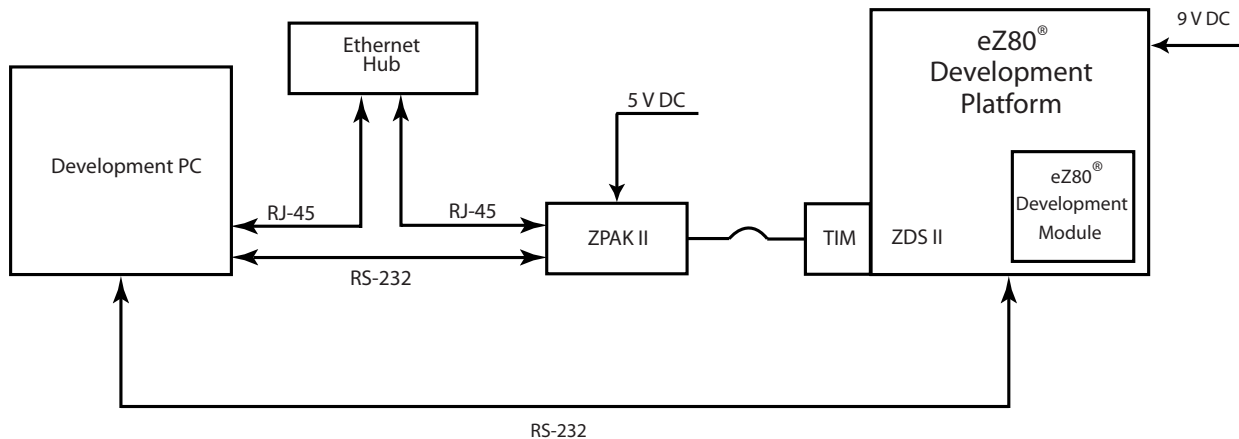


Figure 1. Hardware Setup for ZTP Applications using ZPAK II

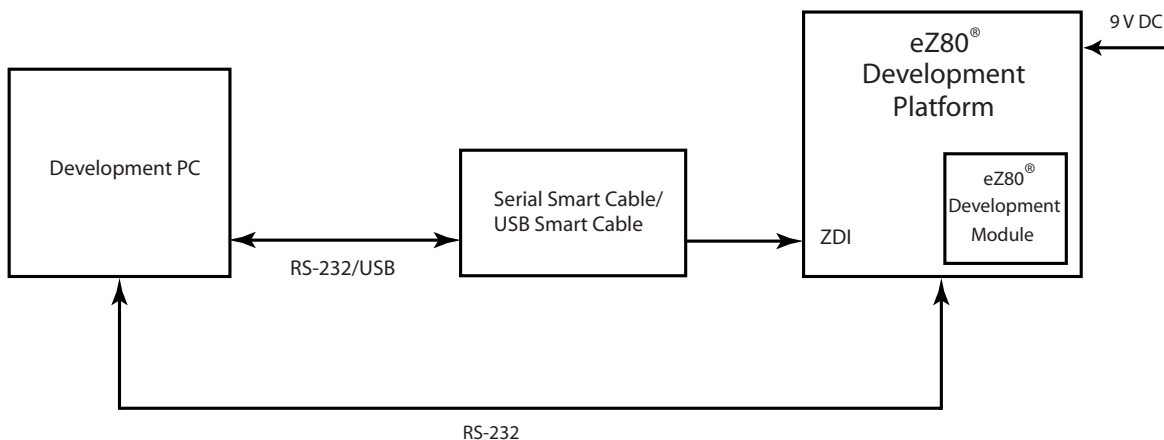


Figure 2. Hardware Setup for ZTP Applications using Serial Smart Cable/ USB Smart Cable

ZTP mail() Function

ZTP provides the `mail()` function which allows you to send email messages using the SMTP. The `mail()` function sends an SMTP mail message to a specified SMTP Server/Port and establishes a TCP connection for the mail transfer. The `mail()` function works with either the Ethernet or the PPP network interface.

► **Note:** *An SMTP server and the domain name or the IP address of the server must be specified. Domain names or IP addresses can also be used as the sender's and recipient's email address.*

The `mail()` function is provided below:

```
INT16 mail(
    INT8 *Addr,
    UINT16 port,
    INT8 *subject,
    INT8 *to,
    INT8 *from,
    INT8 *username,
    INT8 *passwd,
    INT8 *data,
    INT8 *error,
    UINT16 errorlen);
```

Table 2 describes the parameters for the `mail()` function.

Table 2. mail() Function Parameters

Parameter	Description
Addr	A pointer to a character string containing the name or IP address (in decimal/dotted notation) of the SMTP Server.
port	The SMTP Port to use (normally it is Port 25).
subject	A character string containing the Subject: text in the mail message.
to and from	Character strings containing the email addresses of the recipient and sender, respectively.

Table 2. mail() Function Parameters

Parameter	Description
data	A character string containing the body of the email along with any additional headers. The data buffer must contain a mime-content type header. An example of this type of header is given below: <pre>MIME-Version: 1.0\r\n Content-Type: TEXT/PLAIN; charset=US-ASCII\r\n\r\n</pre>
username and passwd	A valid username and password for SMTP Server.
error	A pointer to a buffer-pointer in which ZTP can place a text string describing the reason why the <code>mail()</code> function failed to send the message. The user is responsible for allocating and freeing this buffer.
errorlen	The maximum size (in bytes) of the buffer referenced by the <code>error</code> parameter.

The `mail()` function automatically adds the *Date*, *Subject*, *From*, and *To* lines in the body of the message. The `mail()` function returns an OK when successful and a SYSERR when it fails to deliver the email.

Developing the ZTP-SMTP Mail Server Application

To develop the Mail Server application using the eZ80AcclaimPlus!TM MCU, certain functions must be added to the ZTP to send electronic mail via SMTP. The software implementation of the Mail Server application focuses on the additional functions developed exclusively for this application.

Software Implementation

The software implementation for the Mail Server application consists of developing functions for three stages of the mail transfer process:

- [Initializing the Port](#)
- [Monitoring the Trigger Event to Send Email](#)
- [Sending Electronic Mail](#)

Initializing the Port

The `init()` function (located in the `Test_cont.c` file) initializes the Port B bit 0 (PB0) as an input to obtain the switch status. The switch SW1 is used to generate the trigger event to the microcontroller to send an email.

Monitoring the Trigger Event to Send Email

Pressing the switch (SW1) on the eZ80[®] Development Platform is the trigger event for sending the email to the required destination. However, any other trigger event can be used instead.

The `Test_switch()` function (located in the `Test_cont.c` file) polls the SW1 switch continuously and reads the switch status in the main program.

If the switch is pressed (logic 0), the `Test_switch()` function calls another function, `sendmail()` to send the mail to the specified address.

Sending Electronic Mail

The `sendmail()` function (located in the `send_mail.c` file) is responsible for sending the email to the specified email ID.

The standard ZTP `mail()` function is called within the `sendmail()` function.

► **Note:** *You must provide all the information required for sending the email in the `mail()` function.*

A sample `mail()` function with all the information is illustrated below:

```
strcpy(mail_parm.mail_Addr, "SmtprServer.mycompany.com" );
strcpy(mail_parm.mail_Subject, "TestMail");
strcpy(mail_parm.mail_To, "JohnDoe@mycompany.com");
```

```
strcpy(mail_parm.mail_From, "eZ80EvalBoard@zilog.com");
strcpy(mail_parm.mail_Body, "MIME-Version: 1.0\r\nContent-Type: TEXT/PLAIN;\r\n\r\n");
```

Adding and Integrating Mail Server Files to ZTP

The Mail Server application described in this application note requires the eZ80[®] Development Platform that contains the eZ80F91 MCU with the ZTP stack. To execute the Mail Server application, the files specific to the application must be added and integrated to the ZTP stack before it is downloaded onto the eZ80[®] Development Platform. This section provides the details of adding the Mail Server application files to the ZTP stack. The Mail Server application files that must be added to the ZTP project files are in the `AN0207-SC01.zip` file available for download at www.zilog.com. The Mail Server application files are of the following types:

- C (*.c) files
- Header (*.h) files

The ZTP stack is available on the www.zilog.com and can be downloaded to a PC with a user registration key. ZTP can be installed in any user-specified location, its default location is `C:\Program Files\ZiLOG`.

► **Note:** *See [Requirements](#) on page 5, for the ZDS II and ZTP version used in this application.*

Follow the steps below to add and integrate the Mail Server application files to the ZTP stack:

1. Download ZTP. Browse to the location where ZTP is downloaded, and open the `..\ZTP_2.1.0_Lib_ZDS\ZTP\SamplePrograms` folder.
2. Download the `AN0207-SC01.zip` file and extract its contents to a folder on your PC (this



folder is referred to as \AN0207-SC01, in the rest of the application note).

- Copy all the *.c and *.h files located in the \AN0207-SC01 folder to the ..\ZiLOG\ZTP_2.1.0_Lib_ZDS\ZTP\SamplePrograms folder.
- Launch ZDS II for eZ80Acclaim!® and open the ZTPDemo_F91.zdsproj project available in the below path:
..\ZiLOG\ZTP_2.1.0_Lib_ZDS\ZTP\SamplePrograms
- Click **Project** and select **Add Files** to add all the *.c files located in the \AN0207-SC01 folder to the ZTPDemo_F91.zdsproj project. The *.c files to be added are:
send_mail.c
Test_cont.c
- Open the main.c file of the ZTPDemo_F91.zdsproj project and add the following include file:
#include "send_mail.h"
- Look for the BootInfo structure definition in the ZTPConfig.c file.

```
struct commonServers csTbl={
    "172.16.6.28", // Default Timer Server
    "",          // Default rfs server
    "",          // Default File Server -
    "",          // Not currently Used
    "172.16.6.194"// Default Name Server
};

struct If ifTbl[MAX_NO_IF]= {

// interface 0 -> Ethernet Configurattion
{
    &usrDevBlk[0], // Control block for this
    device
    ETH,           // interface type
    ETH_MTU,      // MTU
    ETH_100,      // Speed can be ETH_10,
    AUTOSENCE
    "172.16.6.198", // Default IP address
    "172.16.6.1",  // Default Gateway
    0xffff0000UL  // Default Subnet Mask
}
}
```

The Bootrecord variable contains network parameters and settings (in the four-octet dot-

ted decimal format) specific to the local area network at Zilog, as default. Modify the above structure definition with appropriate IP addresses within your local area network (for details on modifying the structure definition, refer to documents listed in [References](#) on page 7).

- The Dynamic Host Configuration Protocol (DHCP) is disabled for this application, therefore, ensure that
byte b_use_dhcp = FALSE.
- At the end of the main.c file, add the following lines of code before the return (OK) statement:
Init();
while(1)
{
 Test_switch();
 RZKSuspendThread(RZKGetCurrentThread(
) , 1);}
- Save the files and close the ZTPDemo_F91 project.

Using the eZ80F91 Mail Server Application

This section provides the hardware and software requirements, settings and the instructions to set up and run the Mail Server application.

Requirements

The hardware and software requirements for running the Mail Server application are provided below:

Hardware

- eZ80F91 Development Kit (eZ80F910200ZCO)
- eZ80F91 Development Kit (eZ80F910300ZCOG)
- PC with an Internet Browser and HyperTerminal



Software Requirements

- Zilog Developer Studio II for eZ80Acclaim!® v4.11.0 (ZDS II-IDE v4.11.0)
- Zilog TCP/IP stack (ZTP v2.1.0)

Settings

HyperTerminal Settings

- Set HyperTerminal to 57.6 Kbps Baud, 8-N-2, with no flow control

Jumper Settings

The jumper settings for eZ80® Development Platform and eZ80F91 Module are provided below:

For the eZ80 Development Platform

- J11, J7, J2 are ON
- J3, J20, J21, J22 are OFF
- For J14, connect 2 and 3
- For J19, MEM_CEN1 is ON, and CS_EX_IN, MEM_CEN2, and MEM_CEN3 are OFF

For the eZ80F91 Module mounted on the eZ80 Development Platform

- JP3 is ON

Procedure

Follow the steps below to build and run the Mail Server application:

1. Ensure that the required Mail Server application files are added and integrated to ZTP before proceeding. For details, see [Adding and Integrating Mail Server Files to ZTP](#) on page 4.
2. Make the connections as shown in [Figure 1](#) on page 2. Follow the jumper settings provided in [Jumper Settings](#).
3. Connect the 5 V power supply to ZPAKII and the 7.5 V power supply to the Ethernet HUB.
4. Launch the HyperTerminal and follow the settings provided in the [HyperTerminal Settings](#).
5. From within the HyperTerminal, press *z* repeatedly, and then press the reset button on ZPAKII to view the menu to set the ZPAKII IP address.
6. Enter *H* to display help menu, and follow the menu instructions to obtain the IP address for ZPAKII in order to download the Demo file. This ZPAKII IP address must be entered in the ZDS II.
7. Launch ZDS II for eZ80Acclaim!® and open the ZTPDemo_F91.zdsproj project available in the below path:


```
..\ZiLOG\ZTP_2.1.0_Lib_ZDS\ZTP\SamplePrograms
```
8. Open the ZTPConfig.c file. Ensure that the BootInfo(struct commonServers, struct If ifTbl) structure contains information that is relevant to your network configuration.
9. Open the Test_Cont.c file. Ensure that the SMTP server address, the destination mailbox email ID, and the message are entered as required in the mail() function.
10. Build the project and download the resulting file to the eZ80F91 Module on the eZ80® Development Platform, using ZDS II. For details on the downloading procedure, refer to the *Zilog TCP/IP Software Suite Programmers Guide (RM0041)*.
11. Run the Mail Server application (see [Running the Mail Server Application](#) below).

Running the Mail Server Application

Follow the steps below to run the Mail Server application:

1. Press SW1 switch on the eZ80 Development Platform for one second.



2. Check specified mailbox after a couple of minutes (depends on the network traffic) to see if an email with the predefined text is received.

The eZ80F91 MCU-ZTP Mail Server successfully sends the email to the specified address.

Summary

This application note highlights an eZ80AcclaimPlus!TM MCU's capability to perform as efficient embedded mail servers. The advantage of using eZ80AcclaimPlus! MCU's with the ZTP is that information from the embedded device can be sent to anywhere in the world using email and the event monitoring process to send the email is controlled by the eZ80AcclaimPlus! MCU's interaction with the hardware devices.

References

The documents associated with eZ80F91 MCU available on www.zilog.com are provided below:

- eZ80[®] CPU User Manual (UM0077)
- eZ80F91 Development Kit User Manual (UM0142)
- eZ80F91 MCU Product Specification (PS0192)
- eZ80F91 ASSP Product Specification (PS0270)
- eZ80F91 Module Product Specification (PS0193)
- Zilog Developer Studio II-eZ80Acclaim![®] User Manual (UM0144)
- ZPAK II Debug Interface Tool Product User Guide (PUG0015)
- Zilog TCP/IP Stack API Reference Manual (RM0040)
- Zilog TCP/IP Software Suite Programmer's Guide (RM0041)
- Zilog TCP/IP Software Suite Quick Start Guide (QS0049)

Appendix A—Flowcharts

Figure 3 displays the flowchart for the `main()` routine in the Mail Server application.

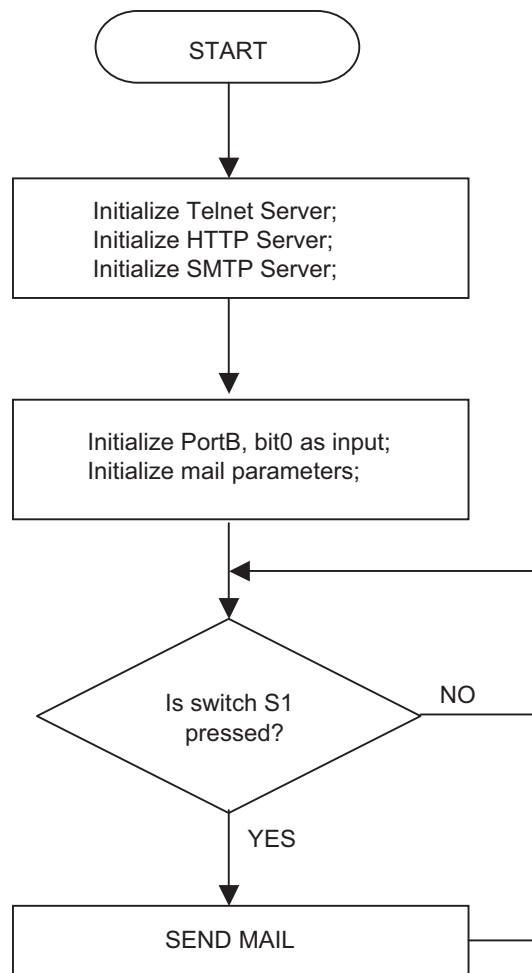


Figure 3. Flowchart for the Main Routine



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.

eZ80Acclaim*Plus!* is a trademark of Zilog, Inc. eZ80Acclaim! and eZ80 are registered trademarks of Zilog, Inc. All other product or service names are the property of their respective owners.