Application Note

A Keypad Routine Using the eZ80F91 MCU

AN020003–0608

Abstract

This application note demonstrates how to interface a general-purpose 16-button (4x4) matrix keypad with the eZ80F91 MCU, a member of Zilog's eZ80Acclaim*Plus!*™ family of Flash-based products. The eZ80F91 MCU scans the keypad and displays an entered character in a HyperTerminal session.

eZ80Acclaim*Plus!*™ Overview

eZ80Acclaim*Plus!* on-chip Flash microcontrollers are an exceptional value for customers designing high-performance, 8-bit MCU-based systems. With speeds up to 50 MHz and an on-chip Ethernet MAC (eZ80F91 MCU only), designers have the performance necessary to execute complex applications quickly and efficiently. Combining Flash and SRAM, eZ80Acclaim*Plus!* devices provide the memory required to implement communication protocol stacks and achieve flexibility when performing in-system updates of application firmware.

eZ80Acclaim*Plus!* Flash MCUs can operate in full 24-bit linear mode, addressing 16 MB without a Memory Management Unit. Additionally, support for the Z80-compatible mode allows Z80/Z180 customers to execute 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 communications peripherals, designers have several options when interfacing to external devices.

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

Discussion

Interfacing a general-purpose 16-button (4x4) matrix keypad can be achieved via a couple of I/O pins and firmware. The 4x4 matrix keypad is scanned via 4 outputs that write logical Low and read via 4 inputs to provide keypress data. Four 10 kΩ resistors are provided for pulling up the rows to reflect a logical High.

4x4 Matrix Keypad Description

The 4x4 matrix keypad is a general-purpose keypad. It consists of 16 switches arranged in 4 rows and 4 columns. It can connect to the MCU 8-bit port directly. Figure 1 displays an internal construction of a typical 4x4 keypad.

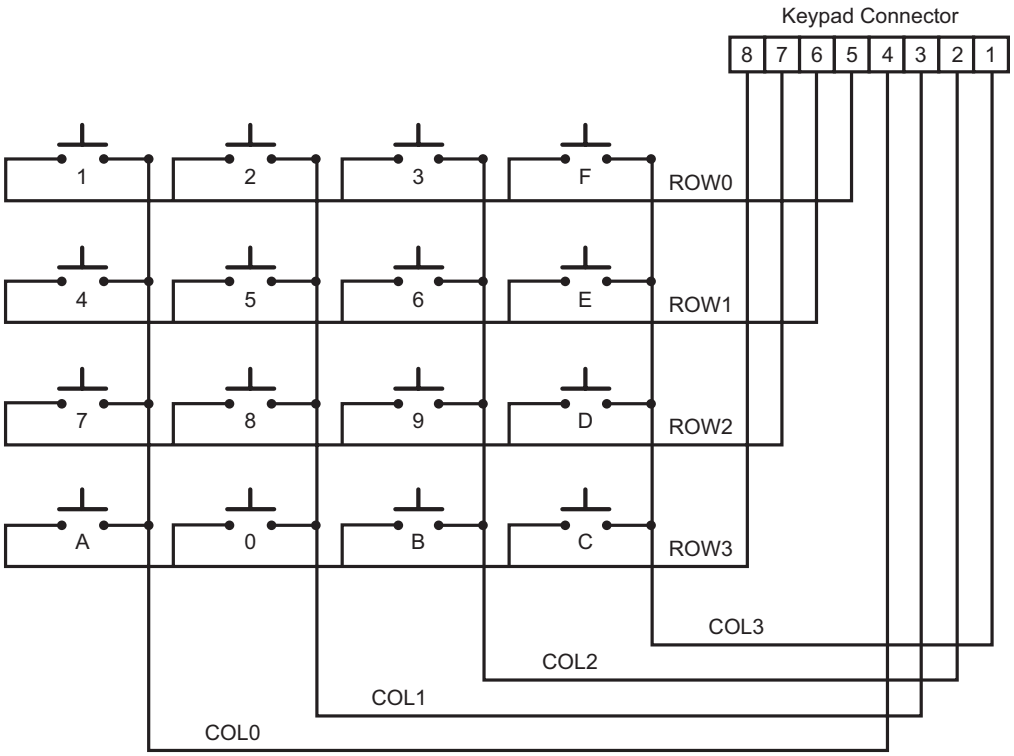

Figure 1. Typical Keypad Construction

The keypad is marked with numeric keys (0–9) and with function keys (A–F), as displayed in Figure 2. In this application, all keys are used as data keys.

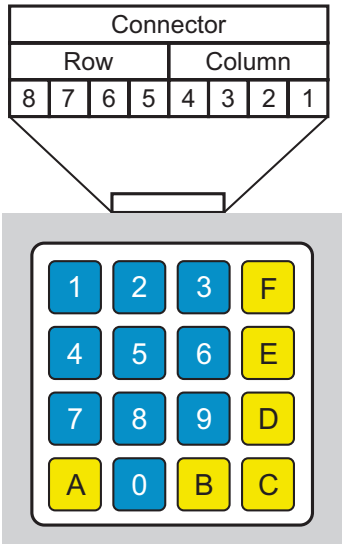

Figure 2. Keypad Detail

Developing the Application for the eZ80F91 MCU

The following sections describe the hardware and software components required to build the keypad routine.

Hardware Architecture

Figure 3 displays a block diagram of the hardware architecture featuring the eZ80F91 MCU, a keypad, and the HyperTerminal application.

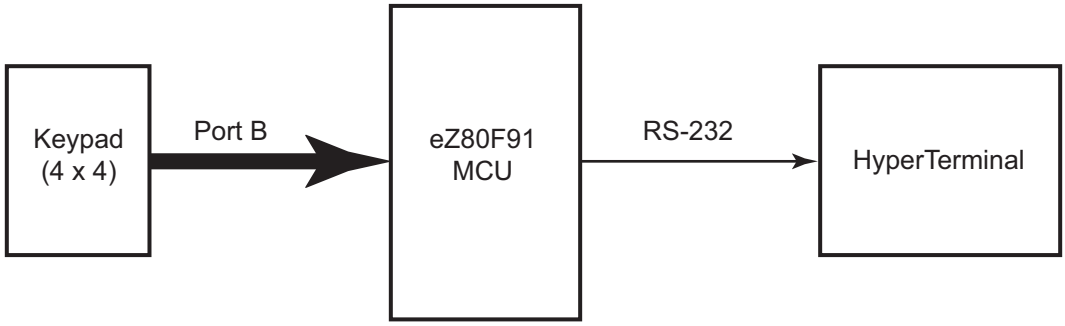

Figure 3. Block Diagram of the Keypad Interface with the eZ80F91 MCU

In this application, the keypad is connected to Port B. Port B is connected to the keypad, and scans the keys continuously. The columns of the keypad are pulled up with 10 kΩ resistance to set them normally High.

Figure 4 displays the Keypad and Port B connection details.

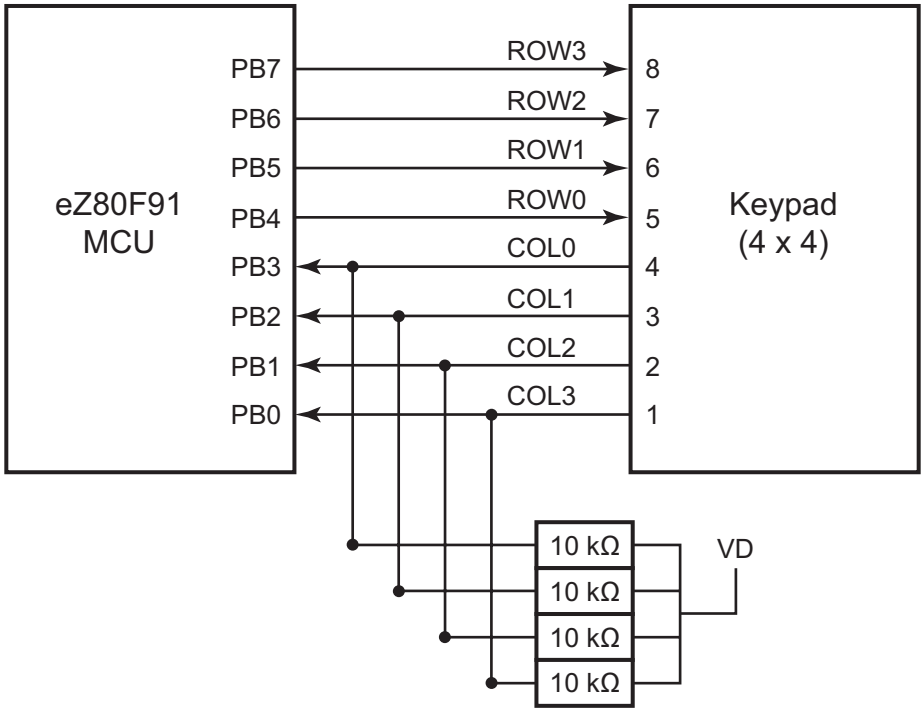

Figure 4. Connection Details between the eZ80F91 MCU and the Keypad

Software Implementation

The keypad routine scans all the keys continuously to find a key that has been pressed. If the keypad routine finds a pressed key, it displays the key character in the HyperTerminal window. Appendix A—Flowcharts on page 6 displays the flowchart for keypad routine.

The following two functions describe this activity in detail.

Key_Fun ()

`Key_Fun()` function scans all keys continuously. When `Key_Fun()` finds a pressed key, it calls a function that has been mapped in a function pointer array to the detected key and updates the HyperTerminal display showing the pressed key. This function is called repeatedly by the **main** function.

isr_timer2()

The timer interrupt service routine (ISR) scans the keypad continuously. When it detects a pressed key, it updates a global variable for that key.

Testing

The following sections describe the test setup and procedure to test the keypad routine.

Test Setup

Figure 5 displays the basic setup for testing the keypad routine on the eZ80F91 MCU. The application emphasizes how to interface the 4x4 keypad using MCU port pins. This setup illustrates the connection between the PC, a LAN/WAN, and the eZ80® Development Platform on which the eZ80F91 Module is affixed.

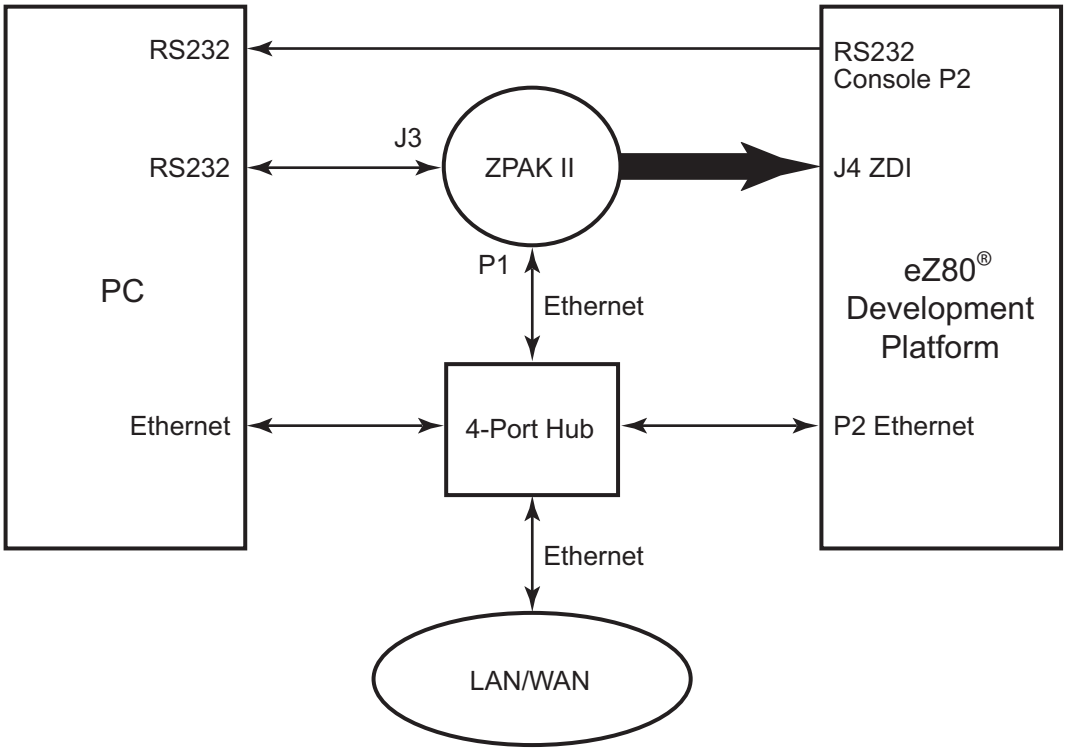

Figure 5. Test Setup for the Keypad Routine Using the eZ80F91 MCU

Equipment Used

The following equipment are used for testing:

- eZ80F91 Development Kit containing an eZ80® Development Platform and the eZ80F91 Module
- Oscilloscope

Procedure

Follow the steps below to test the keypad routine:

1. Set HyperTerminal to 57.6 Kbps and 8-N-1 protocols.
2. Connect the eZ80® Development Platform and the ZPAKII unit to the PC and to the Ethernet via a 4-port hub as displayed in Figure 5.
3. Open the HyperTerminal application to set the IP address for ZPAKII to ensure that you can download the application code.
4. To set the IP address, press the **Z** key while simultaneously pressing the Reset button to launch the menu. Follow the menu instructions.
5. Using ZDS II–eZ80Acclaim!® v4.11.0, open the `keypad_routine.zdsproj` file that is located in the following filepath:

   `..\project\keypad routine\`

6. Use HyperTerminal to initialize UART communication with console port P2.
7. Build the project and download it to the eZ80F91 Module using ZDS II.
8. Run the program.
9. Press any key on the keypad.
10. Monitor the effect of different keypresses in the HyperTerminal window; the character that corresponds to the key should be displayed.

Test Results

The keypad routine successfully runs using the eZ80F91 MCU.

Summary

This application note demonstrates how to interface a 4x4 general-purpose keypad with the eZ80F91 MCU. The keypad routine is developed in such a way that it can be used easily in other projects, and it can be modified for other keypad matrices, such as 3 x 3, 3 x 4, 5 x 5.

References

Further details about the eZ80F91 MCU and eZ80® CPU can be found in the references listed below:

- eZ80F91 MCU Product Specification (PS0192)
- eZ80F91 ASSP Product Specification (PS0270)
- eZ80® CPU User Manual (UM0077)

Appendix A—Flowcharts

Figure 6 displays the flowchart for setting up and initializing the keypad interface, UART, and timer.

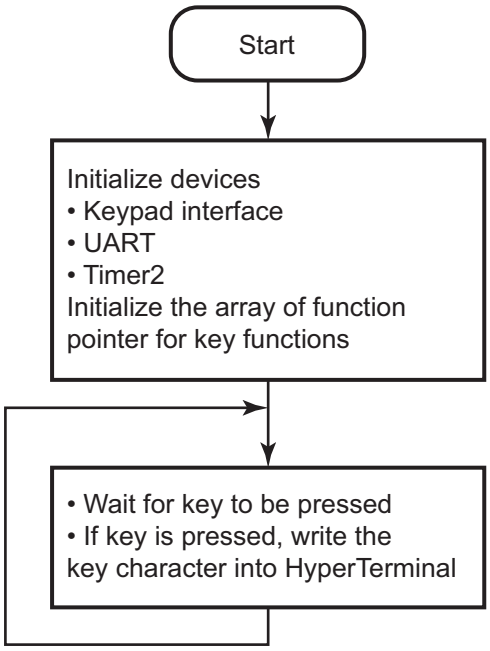

Figure 6. Initializing the Main Program

Figure 7 displays the key scan routine.

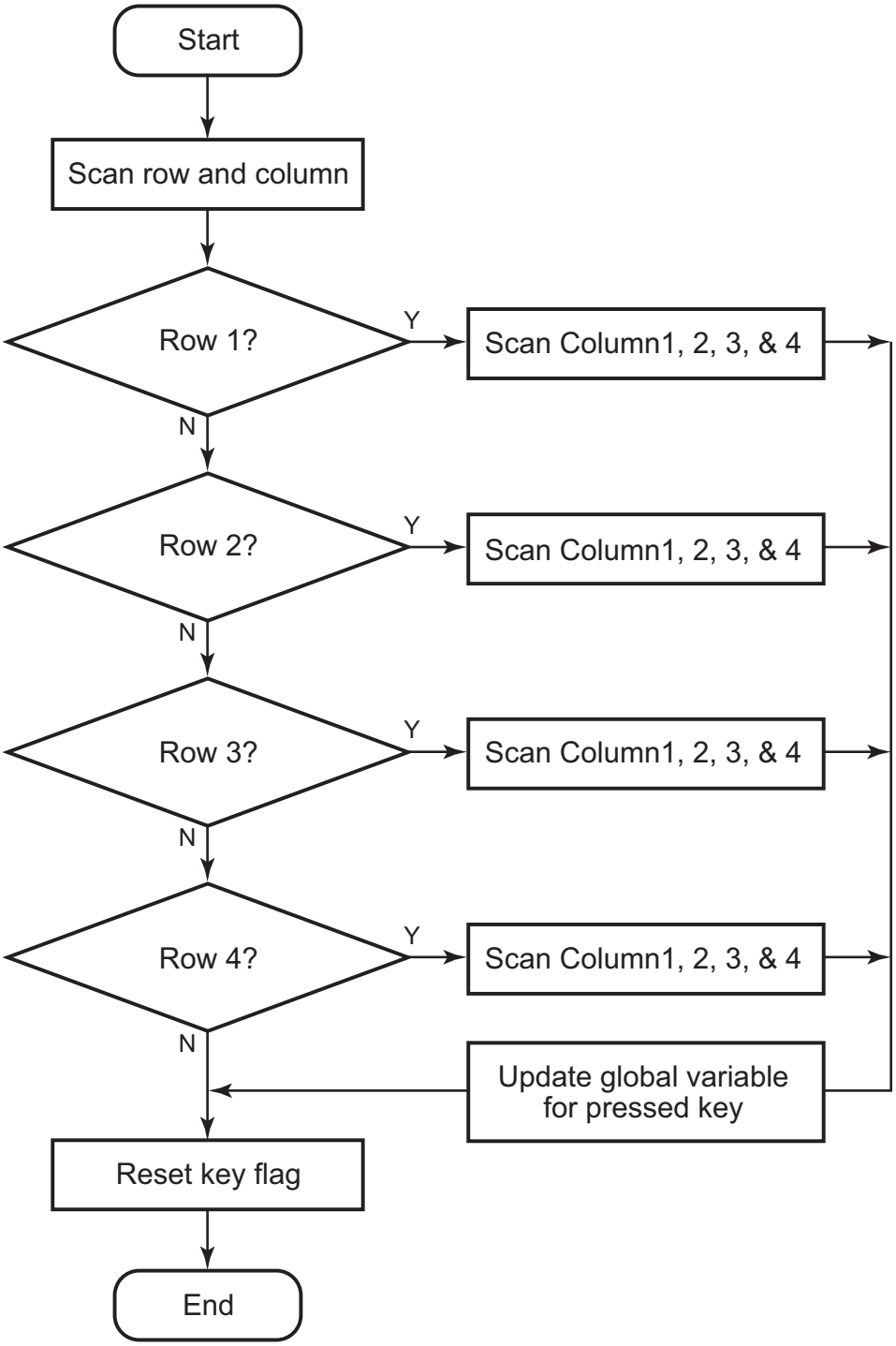

Figure 7. Key Scan Routine

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