Editor's Note: Becoming familiar with an advanced MCU is no easy task. Becoming a true expert is an ongoing challenge. In taking that journey, between that first detailed look at an advanced MCU and becoming an expert, it helps to have the help of a good guide. For engineers, this book offers that good guide in learning about PIC32 MCUs - and more important, how to program with it using the popular chipKIT development board described in detail in an earlier excerpt. In this particular excerpt, the author shows how to use the chipKIT MX3 development board and the MPIDE development environment to build projects that are simple in concept but highly illustrative of common development challenges.

Elsevier is offering this and other engineering books at a 25% discount. To use this discount, click on the coupon to the right and use code PBTY14 (valid through March 31, 2015) during checkout.

Adapted from "PIC32 Microcontrollers and the Digilent chipKIT, 1st Edition - Introductory to Advanced Projects" by Dogan Ibrahim (Newnes)

Reprinted with permission from Elsevier/Newnes © 2015.

Chapter 7. Simple chipKIT MX3-Based Projects

In this chapter, we shall be looking at the design of simple PIC32 32-bit microcontroller-based projects using the chipKIT MX3 development board and the MPIDE development environment, with the idea of becoming familiar with basic interfacing techniques and learning how to use the various microcontroller peripheral registers. We will look at the design of projects using LEDs, push-button switches, keyboards, LED arrays, sound devices, and so on. It is recommended that the reader move through the projects in their given order. The following are provided for each project:

- Description of the project
- Block diagram of the project
- Circuit diagram of the project
- Description of the hardware
- Algorithm description [in program description language (PDL)]
- Program listing
- Photos of the project (where applicable)
- Suggestions for further development (where applicable)

In this book, we will be using the PDL for all the projects.
The use of the MPIDE development environment will be discussed in detail in Project 7.1.

### 7.1 Project 7.1 - Flashing LED

#### 7.1.1 Project Description

This is a simple project that flashes one of the LEDs on the chipKIT MX3 development board at a rate of once a second. There are two LEDs on the development board, and in this project the LED connected to I/O port pin RF0 (LED named LD4) is used. This LED has the logical pin number 42 (see Table 5.8).

An LED can be connected to a microcontroller output port in two different modes: *current-sinking* mode and *current-sourcing* mode.

#### 7.1.2 Current-Sinking

As shown in Figure 7.1, in current-sinking mode the anode leg of the LED is connected to the Vdd supply, and the cathode leg is connected to the microcontroller output port through a current-limiting resistor.

![Figure 7.1: LED Connected in Current-Sinking Mode](image)

The voltage drop across an LED is around 2 V. The brightness of the LED depends on the current through the LED, and this current can vary between a few milliamperes to 16 mA depending on the type of LED used.

The LED is turned ON when the output of the microcontroller is at logic 0 so that current flows through the LED. We can calculate the value of the required resistor as follows:
where Vdd is the supply voltage (3.3 V); VLED is the voltage drop across the LED (2 V); ILED is the current through the LED (say, 10 mA).

Substituting the values into the equation, we get

\[ R = \frac{V_{dd} - V_{LED}}{I_{LED}} \]

\[ R = \frac{3.3 - 2}{10} = 130 \, \Omega \]

The nearest physical resistor is 120 Ω.

### 7.1.3 Current-Sourcing

As shown in Figure 7.2, in current-sourcing mode the anode leg of the LED is connected to the microcontroller output port and the cathode leg is connected to the ground through a current-limiting resistor.

![Figure 7.2: LED Connected in Current-Sourcing Mode](image)

In this mode, the LED is turned ON when the microcontroller output port is at logic 1, that is, Vdd. In practice, the output voltage is about 3.3 V and the value of the resistor can be determined as:

\[ R = \frac{V_{dd} - V_{LED}}{I_{LED}} \]

which gives the same resistor value of 120 Ω.
Project Hardware

7.1.4 Project Hardware

In this project, the LED named LD4 on the chipKIT MX3 development board is used. This LED is connected to port pin RF0 through a switching transistor as shown in Figure 4.4 (it is also possible to connect the LED through a current-limiting resistor as shown in Figures 7.1 and 7.2).

7.1.5 Project PDL

The operation of the project is described in the PDL given in Figure 7.3. At the beginning of the program, the LED port (logical port 42) is configured as output. The program then flashes the LED at a rate of 1 s by sending logic 1 and then logic 0 to the LED with 1-s delay between each output.

BEGIN
  Configure logical I/O pin 42 (port pin RF0) as output
  DO FOREVER
    Send 1 to the port pin (turn LED ON)
    Wait 1 second
    Send 0 to the port pin (turn LED OFF)
    Wait 1 second
  ENDDO
END

Figure 7.3: PDL of Project

7.1.6 Project Program

The program is called LED1, and the program listing is shown in Figure 7.4. At the beginning of the program, the port pin where LD4 is connected to is configured as an output inside the setup() function. The LED is then flashed ON and OFF by sending 1 and 0 continuously with 1-s delay between each output.
/*
     -----------------------------------
    LED FLASHING PROGRAM
   -----------------------------------

This program flashes the LED connected to I/O pin 42 (RF0) of the chipKIT MX3 development Board every second.

Author: Dogan Ibrahim
Date: May, 2014
File: LED1.
Board: chipKIT MX3

-----------------------------------*/
int ledPin = 42;  // LED port
void setup() {    // initialization
    pinMode(ledPin, OUTPUT);
    // set digital pin as output
}
void loop() {
    digitalWrite(ledPin, HIGH);  // turn LED ON
    delay(1000);  // wait 1 second
    digitalWrite(ledPin, LOW);   // turn LED OFF
    delay(1000);  // wait 1 second
}

Figure 7.4: Program Listing

Figure 7.5: Enter the Program Using the MPIDE Development Environment
The following steps describe how to create the program and upload it to the flash memory of the

target microcontroller on the chipKIT MX3 development board:

**Step 1:** Start the MPIDE development environment.

**Step 2:** Enter the program shown in Figure 7.4 (see Figure 7.5).

**Step 3:** Select the development board type by clicking **Tools → Board → Cerebot → Mx3cK.**

**Step 4:** Compile the program by clicking the **Verify** button in the toolbar. If the program compiles

successfully, you should get no error messages (see Figure 7.6).

**Step 5:** Connect the chipKIT MX3 development board to the USB port of your computer. You should

see the red power LED coming on and the green BootLoader LED (LD4) flashing rapidly.

**Step 6:** Select the serial COM port number assigned to the board. This number can be found from

**Control Panel → System → Device Manager** under the heading **Ports (COM & LPT) → USB Serial Port**

(see Figure 7.7).

Click **Tools → Serial Port** to select or confirm the assigned port as shown in Figure 7.8.

![Figure 7.6: Program Compiled With No Errors](image)
Figure 7.7: The Assigned COM Port Is COM28
Step 7: Upload the program to the target microcontroller by clicking File → Upload to I/O Board. You should see the message Done uploading as in Figure 7.9.

Step 8: The microcontroller should reset automatically and the program execution should start. You should now see LED LD4 flashing continuously at a rate of 1 s.

Notice that after pressing the Reset button, the green LED flashes rapidly and the microcontroller waits for about 10 s to communicate with the MPIDE programmer. If there is no response from the programmer, then execution of the user program starts.

Project 7.2 – Alternately Flashing LEDs

7.2 Project 7.2 – Alternately Flashing LEDs

7.2.1 Project Description

This is a simple project similar to Project 7.1. Here, two LEDs flash alternately with 1-s delay between each output. The two LEDs LD4 (I/O port pin RF0, logical pin number 42) and LD5 (I/O port pin RF1, logical pin number 43) on the chipKIT MX3 development board are used in this project. See Table 5.8 for I/O pin assignments.
7.2.2 Project Hardware

Figure 4.4 shows how the LEDs LD4 and LD5 are connected on the chipKIT MX3 development board through switching transistors (it is also possible to connect the LEDs through current-limiting resistors as shown in Figures 7.1 and 7.2).

7.2.3 Project PDL

The operation of the project is described in the PDL given in Figure 7.10. At the beginning of the program, LD4 (logical port 42) and LD5 (logical port 43) are configured as outputs. The program then flashes the two LEDs alternately at a rate of 1 s by sending logic 1 and then logic 0 to the LEDs alternately with 1-s delay between each output.

```
BEGIN
  Configure logical I/O pin 42 (port pin RF0) as output
  Configure logical I/O pin 43 (port pin RF1) as output
  Turn OFF LD4 and LD5
DO FOREVER
  Send 1 to port pin RF0 (turn LD4 ON)
  Send 0 to port pin RF1 (turn LD5 OFF)
  Wait 1 second
  Send 0 to port pin RF0 (turn LD4 OFF)
  Send 1 to port pin RF1 (turn LD5 ON)
  Wait 1 second
ENDDO
END
```

Figure 7.10: PDL of Project

7.2.4 Project Program

The program is called LED2, and the program listing is shown in Figure 7.11. At the beginning of the program, the port pins where LD4 and LD5 are connected to are configured as outputs inside the `setup()` function. Also, both LEDs are turned OFF to start with. The LEDs are then flashed ON and OFF alternately by sending 1 and 0 continuously with 1-s delay between each output.
**LED FLASHING PROGRAM**

This program flashes alternately two LED connected to I/O pin 42 (RF0) and I/O pin 43 (RF1) of the chipKIT MX3 development board every second.

```c
/*
 ***  LED FLASHING PROGRAM  
 */

int LD4 = 42; // LED LD4 port
int LD5 = 43; // LED LD5 port

void setup()
{
    pinMode(LD4, OUTPUT); // set LD4 pin as output
    pinMode(LD5, OUTPUT); // set LD5 pin as output
digitalWrite(LD4, LOW); // turn OFF LD4
digitalWrite(LD5, LOW); // turn OFF LD5
}

void loop()
{
digitalWrite(LD4, HIGH); // turn LD4 ON
digitalWrite(LD5, LOW); // turn LD5 OFF
delay(1000); // wait 1 second
digitalWrite(LD4, LOW); // turn LD4 OFF
digitalWrite(LD5, HIGH); // turn LD5 ON
delay(1000); // wait 1 second
}
```

**Figure 7.11: Program Listing**

### 7.3 Project 7.3 - Lighthouse Flashing LED

#### 7.3.1 Project Description

This is a simple project using LED LD4. In this project, the LED is flashed in a group of two quick flashes every second. The flashing rate is assumed to be 200 ms. This type of flashing is identified as Gp Fl(2) in maritime lighthouse lights. Thus, the flashing is repeated as follows:

LED ON
Wait 200 ms
LED OFF
Wait 100 ms
LED ON
Wait 200 ms
LED OFF
Wait 100 ms
Wait 400 ms
7.3.2 Project Hardware

Figure 4.4 shows how LD4 is connected on the chipKIT MX3 development board through a switching transistor (it is also possible to connect the LEDs through current-limiting resistors as shown in Figures 7.1 and 7.2).

7.3.3 Project PDL

The operation of the project is described in the PDL given in Figure 7.12. At the beginning of the program, LD4 (logical port 42) is configured as an output. The program then flashes the LED as a lighthouse signal Gp Fl(2) as described above.

```
BEGIN
  Configure logical I/O pin 42 (port pin RF0) as output
  DO FOREVER
    Send 1 to port pin RF0 (turn LD4 ON)
    Wait 200ms
    Send 0 to port pin RF0 (turn LD4 OFF)
    Wait 100ms
    Send 1 to port pin RF0 (turn LD4 ON)
    Wait 200ms
    Send 0 to port pin RF0 (turn LD4 OFF)
    Wait 100ms
    Wait 400ms
  ENDDO
END
```

Figure 7.12: PDL of Project

7.3.4 Project Program

The program is called LED3, and the program listing is shown in Figure 7.13. At the beginning of the program, the port pin where LD4 is connected to is configured as an output inside the `setup()` function. The LED is then flashed ON and OFF to simulate the lighthouse signalling Gp Fl(2) as described above.
This program flashes the LED connected to I/O pin 42 (RF0) as a lighthouse signal GpFl(2).

```c
/*
 * LED LIGHTHOUSE FLASHING PROGRAM
 */

int LD4 = 42; // LED LD4 port

void setup()
{
    pinMode(LD4, OUTPUT); // set LD4 pin as output
}

void loop()
{
    digitalWrite(LD4, HIGH); // turn LD4 ON
    delay(200); // wait 200ms
    digitalWrite(LD4, LOW); // turn LD4 OFF
    delay(100); // wait 100ms
    digitalWrite(LD4, HIGH); // turn LD4 ON
    delay(200); // wait 200ms
    digitalWrite(LD4, LOW); // turn LD4 OFF
    delay(100); // wait 100ms
    delay(400); // wait 400ms
}
```

**Figure 7.13: Program Listing**

### 7.4 Project 7.4 - LED With Push-Button Switch

#### 7.4.1 Project Description

This is a simple project using LED LD4 with a push-button switch. In this project, the LED is controlled from a push-button switch and is turned ON and OFF when the switch is pushed and released, respectively.

#### 7.4.2 Block Diagram

The block diagram of the project is shown in Figure 7.14.
Project Hardware

7.4.3 Project Hardware

In this project, the four-button Pmod module PmodBTN is used (see Figure 7.15). This module consists of four push-button switches labelled BTN0–BTN3 with on-board debounce filters as shown in Figure 7.16.
The PmodBTN module has a six-pin header and is connected to Pmod connector JA on the chipKIT MX3 development board.

The pin connections of the PmodBTN module are as follows:

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Pin Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BTN0</td>
</tr>
<tr>
<td>2</td>
<td>BTN1</td>
</tr>
<tr>
<td>3</td>
<td>BTN2</td>
</tr>
<tr>
<td>4</td>
<td>BTN3</td>
</tr>
<tr>
<td>5</td>
<td>GND</td>
</tr>
<tr>
<td>6</td>
<td>VCC</td>
</tr>
</tbody>
</table>

A switch pin is at logic 0 and goes to logic 1 when the switch is pressed. When connected to Pmod connector JA, the interface between the switch and the microcontroller I/O pins when the PmodBTN is connected to the top row of the connector is as follows:

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Microcontroller I/O Port Pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RE0 (JA-01, logical I/O pin 0)</td>
</tr>
<tr>
<td>2</td>
<td>RE1 (JA-02, logical I/O pin 1)</td>
</tr>
<tr>
<td>3</td>
<td>RE2 (JA-03, logical I/O pin 2)</td>
</tr>
<tr>
<td>4</td>
<td>RE3 (JA-04, logical I/O pin 3)</td>
</tr>
</tbody>
</table>
In this project, BTN0 is used. This switch has the logical I/O pin number 0 as shown in the above table.

Figure 7.17 shows the PmodBTN module connected to the development board.

![Figure 7.17: Connecting the PmodBTN Module to the Development Board](image)

7.4.4 Project PDL

The operation of the project is described in the PDL given in Figure 7.18. At the beginning of the program, LD4 (logical port 42) is configured as an output and connector JA-01 pin (logical port 0) is configured as an input. The LED is turned ON when the push-button switch is pressed, that is, when the **ButtonState** is at logic 1.
Figure 7.18: PDL of Project

7.4.5 Project Program

The program is called PUSH1, and the program listing is shown in Figure 7.19. At the beginning of the program, the port pin where LED LD4 is connected to is configured as an output and the port pin where button BTN0 is connected to is configured as an input inside the setup() function. The program then reads the state of the button, and the LED is turned ON whenever the button is pressed; otherwise, the LED is turned OFF.

BEGIN
    Configure logical I/O pin 42 (port pin RF0, LD4) as output
    Configure logical I/O pin 0 (port pin RE0, connector JA-01) as input
DO FOREVER
    Read Button state
    IF push-button switch is pressed THEN
        Turn ON LED LD4
    ELSE
        Turn OFF LED LD4
    ENDF
ENDDO

END
7.5 Project 7.5 - Wait Before Flashing the LED

7.5.1 Project Description

This is a simple project similar to Project 7.4. Here, LED LD4 and push-button BTN0 are used as in Project 7.4. The LED is normally OFF and starts flashing at a rate of 1 s when the push-button switch is pressed.

7.5.2 Block Diagram

The block diagram of the project is as in Figure 7.14.

7.5.3 Project Hardware

The project hardware is as in Project 7.4 where a PmodBTN module is used. This module consists of four push-button switches labelled BTN0–BTN3 with on-board debounce filters, and BTN0 is used in this project. The PmodBTN module is connected to Pmod connector JA, and thus BTN0 has the
logical I/O pin number 0 as in the previous project.

**Project PDL**

### 7.5.4 Project PDL

The operation of the project is described in the PDL given in Figure 7.20. At the beginning of the program, LD4 (logical port 42) is configured as an output and connector JA-01 pin (logical port 0) is configured as an input. The LED is normally OFF and starts flashing at a rate of 1 s as soon as the switch is pressed.

```
BEGIN
    Configure logical I/O pin 42 (port pin RF0, LD4) as output
    Configure logical I/O pin 0 (port pin RE0, connector JA-01) as input
    Turn OFF the LED
    DO UNTIL the button is pressed
        Wait
    ENDDO
    DO FOREVER
        Turn ON the LED
        Wait 1 second
        Turn OFF the LED
        Wait 1 second
    ENDDO
END
```

*Figure 7.20: PDL of Project*

### 7.5.5 Project Program

The program is called PUSH2, and the program listing is shown in Figure 7.21. At the beginning of the program, the port pin where LED LD4 is connected to is configured as an output and the port pin where button BTN0 is connected to is configured as an input inside the `setup()` function. The program then reads the state of the button in a loop and waits until the button is pressed. As soon as the button is pressed, the LED starts flashing at a rate of 1 s.
7.6 Project 7.6 - LED With Two Push-Button Switches

7.6.1 Project Description

This is a simple project similar to Project 7.4. Here, LED LD4 and two push-button switches BTN0 and BTN1 are used. The LED is normally OFF and starts flashing at a rate of 500 ms when the push-button switch BTN0 is pressed. The LED stops flashing when push-button BTN1 is pressed.
7.6.2 Block Diagram

The block diagram of the project is shown in Figure 7.22.

![Block Diagram of the Project](image)

Figure 7.22: Block Diagram of the Project

7.6.3 Project Hardware

The project hardware is similar to Project 7.4 where a PmodBTN module is used. This module consists of four push-button switches labelled BTN0–BTN3 with on-board debounce filters. BTN0 is used to start flashing the LED, and BTN1 is used to stop flashing the LED. The PmodBTN module is connected to Pmod connector JA. BTN0 (connector pin JA-01) and BTN1 (connector pin JA-02) have the logical I/O pin numbers 0 and 1, respectively (see Project 7.4).

7.6.4 Project PDL

The operation of the project is described in the PDL shown in Figure 7.23. At the beginning of the program, LD4 (logical port 42) is configured as an output and connector JA-01 pin (BTN0, logical port 0) and JA-02 pin (BTN1, logical port 1) are configured as inputs. The LED is turned ON when the push-button switch BTN0 is pressed. The LED is turned OFF when push-button switch BTN1 is pressed.
BEGIN
Configure logical I/O pin 42 (port pin RF0, LD4) as output
Configure logical I/O pin 0 (BTN0, connector JA-01) as input
Configure logical I/O pin 1 (BTN1, connector JA-02) as input
DO FOREVER
  IF BTN0 is pressed THEN
    Flash = 1
  END
  IF BTN1 is pressed THEN
    Flash = 0
  END
  IF Flash = 1 THEN
    Turn ON LED LD4
    Wait 0.5 second
    Turn OFF LED LD4
    Wait 0.5 second
  ENDIF
ENDDO
END

Figure 7.23: PDL of the Project

7.6.5 Project Program

The program is called PUSH2, and the program listing is shown in Figure 7.24. At the beginning of
the program, the port pin where LED LD4 is connected to is configured as an output and the port
pins where buttons BTN0 and BTN1 are connected to are configured as inputs inside the setup() function. The program then reads the state of both buttons. If BTN0 is pressed, then variable Flash
is set to 1. If, on the other hand, BTN1 is pressed, then variable Flash is cleared to zero. The LED
starts flashing at a rate of 500 ms if variable Flash is set to 1. The flashing stops when variable Flash is cleared.
This program uses two push-button switches BTN0 and BTN1 with PmodBTN module. LED LD4 starts flashing when BTN0 is pressed. The flashing stops when BTN1 is pressed. The logical port numbers of ports used are:

LD4 – port 42
BTN0 – port 0
BTN1 – port 1

Author: Dogan Ibrahim
Date: May, 2014
File: PUSH2
Board: chipKIT MX3

```c
int LD4 = 42;  // LD4 logical port number
int BTN0 = 0;  // BTN0 logical port number
int BTN1 = 1;  // BTN1 logical port number
int BTN0State, BTN1State;
int Flash = 0;

void setup()
{
    pinMode(LD4, OUTPUT);  // set LD4 as output
    pinMode BTN0, INPUT;  // set BTN0 as input
    pinMode BTN1, INPUT;  // set BTN1 as input
    digitalWrite(LD4, LOW);  // turn LD4 OFF to start with
}

void loop()
{
    BTN0State = digitalRead BTN0;  // read state of BTN0
    BTN1State = digitalRead BTN1;  // read state of BTN1
    if (BTN0State == 1) Flash = 1;  // if BTN0 is pressed
    if (BTN1State == 1) Flash = 0;  // if BTN1 is pressed

    if (Flash == 1)
    {
        digitalWrite(LD4, HIGH);
        delay(500);
        digitalWrite(LD4, LOW);
        delay(500);
    }
}
```

Figure 7.24: Program Listing of the Project
About the author

Prof Dogan Ibrahim graduated from the University of Salford with First Class Honours in Electronic Engineering. He then completed an MSc course in Automatic Control Engineering at the University of Manchester, and PhD in Digital Signal Processing at the City University in London. Prof Ibrahim worked at several companies before returning to the academic life. He is currently a lecturer at the Department of Computer Information Systems at the Near East University. Prof Ibrahim is a Fellow of the IET, and a Chartered Electrical Engineer. His interests are in the fields of microcontroller based automatic control, digital signal processing, and computer-aided design.

Reprinted with permission from Elsevier/Newnes © 2015.