AN1543

Using MRF24W with PIC32 Internal Program Flash Memory For EZ_CONFIG_STORE

Author: *Amy Ong*
Microchip Technology Inc.

OVERVIEW

This application note describes the EZ_CONFIG_STORE feature used in the Wi-Fi G Demo Board and TCPIP-WiFi EZConfig demo applications. It also describes how to use PIC32 internal program Flash memory to support EZ_CONFIG_STORE.

SCOPE

The EZ_CONFIG_STORE features mentioned in this application note is based on the following, and it can be extended to other PIC32 processors or MRF24WB0MA.

- Hardware platform: PIC32MX695F512H (4k bytes/page) and MRF24WG0MA
- MPLAB® XC32 compiler
- MLA v5 based releases

ALTERNATIVE LOW-COST SOLUTIONS

With the Explorer 16 Development Board, which has an external SPI-EEPROM, EZ_CONFIG_STORE is performed on the external EEPROM. If sufficient internal program Flash memory is available in the PIC32 processor, EZ_CONFIG_STORE provides a low-cost alternative solution to an external EEPROM.

Limitations and Considerations

The following are the limitations and considerations for using the PIC32 internal program Flash memory:

- The program Flash memory has a limited number of read/write cycles. After read/write cycle limit is reached, the operation may become unreliable. Therefore, if the product needs to erase and write the internal program memory frequently, an external EEPROM is recommended.
- To use the PIC32 internal program Flash memory for EZ_CONFIG_STORE, sufficient memory must be available. Refer to the selected PIC32 memory map section in "*PIC32MX5xx/6xx/7xx Family Data Sheet*" (DS61156G).
- The data to be written into the internal program Flash memory must be word-aligned, or else it might cause "hanging" symptom.
- The largest block of data that can be programmed or written in a single operation is 1 row, which is 128 instructions or 512 bytes.

WHAT IS EZ_CONFIG_STORE

EZ_CONFIG_STORE is a feature used in the Wi-Fi G Demo Board and TCPIP-WiFi EZConfig demo applications. For example, when MRF24W is started as a SoftAP, it may choose to be redirected to another AP/router. EZ_CONFIG_STORE stores these wireless network configurations (`AppConfig`) into the nonvolatile memory (NVM). When MRF24W is powered off and power-up again, the wireless network configurations (`AppConfig`) can be retrieved from the NVM, thereby eliminating the need to repeat the process of redirecting to the selected AP/router. EZ_CONFIG_STORE retains the wireless network configurations (`AppConfig`), and thereby saving the time to reconnect to the chosen AP/router.

OPERATION OF EZ_CONFIG_STORE

The reference source files are available in the following location: `TCPIP\ WiFi EZConfig` and `TCPIP\WifiGDemo`.The wireless network configurations are defined in the data struct `AppConfig`.

When MRF24W is redirected to the selected AP/router, the following sequence of actions will happen:

1. `(CustomHTTPApp.c)` in `HTTPPostWifiConfig()`, the data struct `CFGCXT` will be copied to the data struct `AppConfig`, and then `AppConfig.DataValid` will be set to `1`.
2. `(MainDemo.c) AppConfig.DataValid` will set `CFGCXT.isWifiDoneConfigure` to `1`, and which leads to `WFEasyConfigProcess()` being triggered.
3. `(WFEasyConfig.c) WFEasyConfigProcess()` will initiate `SaveAppConfig()`, which will program `AppConfig` into the NVM, then MRF24W enters Hibernate mode and exits Hibernate mode to connect to the newly selected network.
4. `(MainDemo.c)`, when MRF24W is powered off and power-on again, `InitAppConfig()/ InitAppConfig2()` will be initialized with `AppConfig`, retrieved from the internal program Flash memory.
5. `(MainDemo.c)`, when the reset button is pressed to return to default factory conditions, `RestoreWifiConfig()` will be invoked to erase the internal program Flash memory `(NVMErasePage())` and then perform a `Reset()`.

INCORPORATING PIC32 INTERNAL PROGRAM FLASH MEMORY

The Flash memory write operation must be preceded by a Flash memory erase page operation because the Flash memory erase operation only writes the `1`s, while the Flash memory write operation can only write `0`s.

Required Files for Using PIC32 Internal Program Flash Memory

The following NVM files are available in the MPLAB XC32 compiler directory:

- `C:\Program Files (x86)\Microchip\xc32\v1.20\pic32-libs\include\Plib.h`

  This header file points to `peripheral\nvm.h`.

- `C:\Program Files (x86)\Microchip\xc32\v1.20\pic32-libs\peripheral\nvm`
  - `Nvm_erase_page_lib.c`
  - `Nvm_operation_lib.c`
  - `Nvm_program_lib.c`
  - `Nvm_write_row_lib.c`
  - `Nvm_write_word_lib.c`

  These files contain the program Flash memory operations, such as writing and erasing the internal program Flash memory.

Writing Source Code for Programming PIC32 Internal Program Flash Memory

To write source codes for programming the PIC32 internal program Flash memory, perform these tasks:

1. Include `plib.h` in the source file:

```
#include      <plib.h>/*      PIC32
peripheral library */
```

2. Before NVM operation, check the data is word-aligned (4 bytes).

   The following source code will help to check the size of the variable (`AppConfig`) and `pagebuff` are word-aligned, or else it will end up in the `while(1)` loop.

```
unsigned int pagebuff[1024];
.....
if (sizeof(AppConfig)  &  0x03  ||
(unsigned int)pagebuff & 3)
      while(1);
.....
```

   or use `sizeof()` to determine the size of the variable. The size of the variable must be dividable by 4.

```
x = Sizeof(AppConfig)
```

   If `x = 10`, this violates the word-aligned requirement, and if `x = 8`, this fulfills the word-aligned requirement.

3. Start the internal program Flash memory write operation. As an example for PIC32MX-695F512H, refer to PIC32MX695F512H memory map in "*PIC32MX5xx/6xx/7xx Family Data Sheet*" (DS61156G.

```
#define                NVM_PROGRAM_PAGE
0xbd070000
```

a) Erase program Flash page.

```
NVMErasePage((void*)NVM_PROGRAM_PAG
E);
```

b) Depending on the size of data to be written, invoke the selected NVM function prototype.

- To write 156 bytes into the internal program Flash memory, choose `NVMWriteRow()` which can program up to 512 bytes.

```
NVMWriteRow((void*)NVM_PRO-
GRAM_PAGE, (void*)&AppConfig);
```

- To write 8 bytes (1 word) into the internal program Flash memory, choose `NVMWrite-Word()`.

```
NVMWriteWord((void*)NVM_PROGRAM_PAG
E, 0x12345678);
```

- To write 1500 bytes into the internal program Flash memory, select `NVMProgram ()`.

```
NVMProgram((void*)NVM_PROGRAM_PAGE,
(const              void*)databuff,
sizeof(databuff),          (void*)
pagebuff);
```

  Where, `databuff` contains the data to be written.

4. To verify, perform a memory compare operation (`memcmp`) on the selected internal program Flash memory and original data. If mismatch occurs, it will end up in the `while(1)` loop.

```
if(memcmp((const void *)&AppConfig,
(void           *)NVM_PROGRAM_PAGE,
sizeof(AppConfig)))
{
           while(1);
}
```

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

Printed on recycled paper.

ISBN: 978-1-62077-287-4

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Farmington Hills, MI
Tel: 248-538-2250
Fax: 248-538-2260

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

Santa Clara
Santa Clara, CA
Tel: 408-961-6444
Fax: 408-961-6445

Toronto
Mississauga, Ontario, Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Hangzhou
Tel: 86-571-2819-3187
Fax: 86-571-2819-3189

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828
Fax: 886-7-330-9305

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

11/29/12