

ROM FILE MAKER

MANUAL

V2.5

Wiznet, Inc

Copyright 2002 WIZnet, Inc. All Rights Reserved.



Technical Support: support@wiznet.co.kr

Sales & Distribution: sales@wiznet.co.kr

General Information: info@wiznet.co.kr

For more information,

visit our website at <http://www.iinchip.com> or <http://www.wiznet.co.kr>

CONTENTS

1. What is 'ROM File System'	3
2. About 'ROM File Maker'	4
3. Installation	4
4. How to use	5

FIGURES

<Fig 1. iinChip™ EVB B/D의 ROM File System 구성>	3
<Fig 2 ROM File Maker 실행 화면>	5
<Fig 3 EVB-A1's "types.h">	6
<Fig 4 'ROM File' add Files>	7
<Fig 5 'ROM File System' Generation>	7
<Fig 6 "romfs.h">	8
<Fig 7 "romfs.c">	9

TABLES

<Table 1 search_file(>	4
------------------------------	---

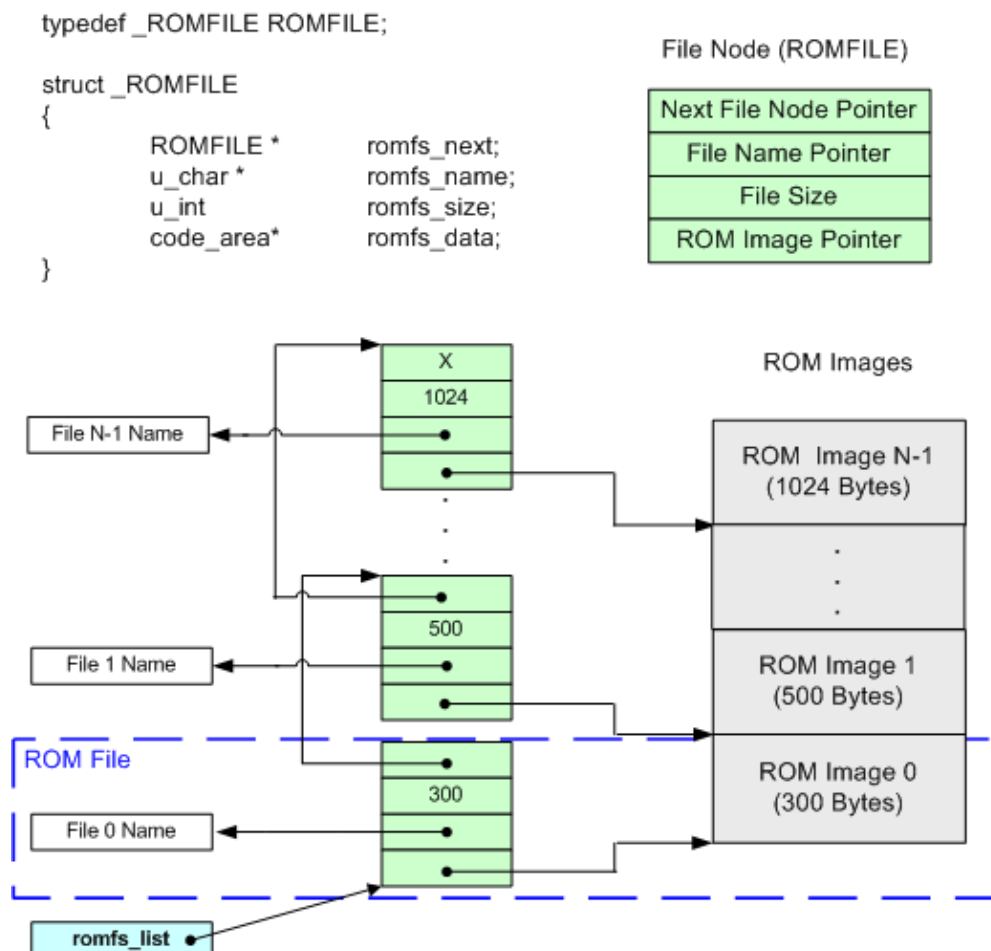
1. What is 'ROM File System'

iinChip™ EVB B/D의 'ROM Files'이란 Code Memory Area(ROM, Flash 등)에 적재되어 있는 File들을 말하며, 하나의 'ROM File'은 File Name, File Size, File Contents로 구성된다. 각 File들의 Contents들은 Binary Code로 변환되어 ROM에 적재된다.(이하 ROM에 적재된 File의 Contents를 'ROM Image'라 칭함).

'ROM File System'은 여러 개의 'ROM File'을 손쉽게 관리하기 위해 고안된 것으로, 각 'ROM File'들을 File Node Structure인 **ROMFILE**로 정의하고, 그 **ROMFILE**들을 ROM 적재 순서에 따라 참조할 수 있는 Linked List 구조인 **romfs_list**로 관리한다.

'ROM File System'은 iinChip™ EVB 개발 시 요구될 수 있는 각종 File들을-예로 Web Page File(HTML,JPGE, GIF등)들이나, etc.- 보다 효율적으로 개발하고, 관리하고자 고안된 것이다.

<Fig 1>은 iinChip™ EVB B/D의 ROM File System의 구조를 설명한 것으로, ROMFILE과 ROM Image간, ROMFILE과 ROMFILE간의 상관관계를 도식화 한 것이다.



<Fig 1. iinChip™ EVB B/D의 ROM File System 구성>

<Fig 1>의 'ROM Image'는 iinChip™ EVB B/D의 ROM에 위치하며, 각 File Node들과 Linked List인 **romfs_list**는 iinChip™ EVB B/D의 RAM에 위치한다.

2. About 'ROM File Maker'

'ROM File Maker'은 iinChip™ EVB F/W 개발 시 요구되는 'ROM File System'을 구성하기 위한 Tool로서, Windows 기반의 PC Program이다.

'ROM File Maker'은 'ROM File System'의 **ROMFILE** Structure 정의, 'ROM Image' 생성, 'ROM File' 검색 함수 등을 C Language로 작성된 Source Files – romfs.h & romfs.c – 을 자동으로 생성한다.

'ROM File System' 내에 존재하는 각 'ROM File'은 **search_file()**를 이용해 File Name 으로 검색할 수 있다.

<Table 1>은 **search_file()**에 대한 설명이다.

<Table 1 search_file()>

Prototype	u_char search_file(u_char * name, code_area ** buf, u_int * len)	
Parameter	name	검색할 ROM File Name
	buf	검색된 ROM File Location Return
	len	검색된 ROM File Size Return
Result	> 0 : Success, 0 : Fail	

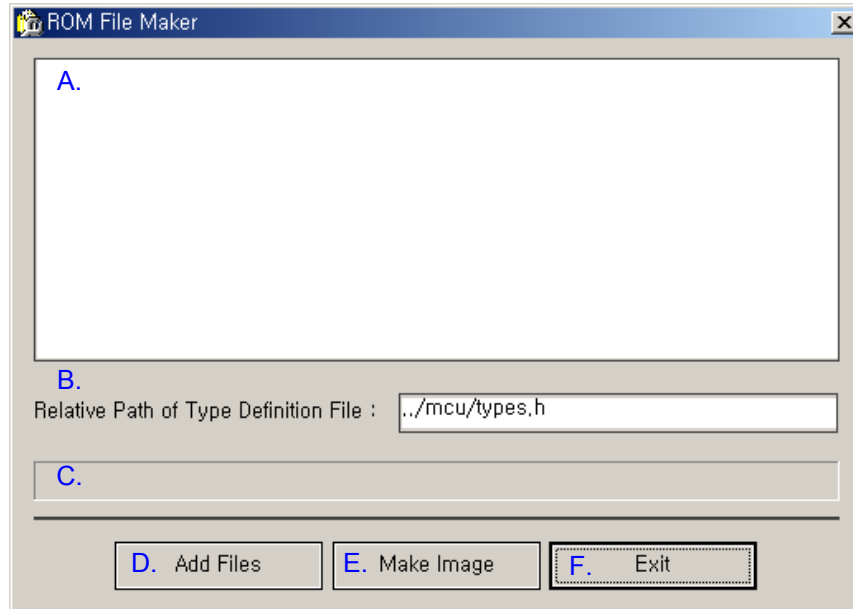
3. Installation

'ROM File Maker'는 설치파일 "RFMakerV¹X.X.exe"를 실행함으로써 설치할 수 있다.
설치파일은 iinChip™ EVB B/D 구입 시 받은 Software CD이나, WIZnet사의 Homepage 인 <http://www.wiznet.co.kr/>에서 최신 Version을 download 받아 설치할 수 있다.

¹ X.X means Version Information of Program

4. How to use

- ① "RFMaker.exe"를 실행한다.



<Fig 2 ROM File Maker 실행 화면>

- A. File List
- B. iinChip™ EVB B/D에서 사용하는 Data Type들을 정의한 파일의 Relative Path Name
- C. 'ROM File System' 생성 과정
- D. 'ROM File'로 생성될 File 추가
- E. 'ROM File System' 생성
- F. Program 종료

- ② "code_area" Type이 정의된 파일의 Relative Path를 지정한다.

<주의> 지정한 파일에서 "code_area" Type을 반드시 정의해야 한다.

"code_area" Type은 Program이 생성하는 "romfs.h"와 "romfs.c"에서 참조하는 CODE Memory Type으로 사용되기 때문이다.

iiinChip™ EVB F/W에서는 "types.h"에 "code_area" Type을 정의하였다.

<Fig 3> 참조.

```

#ifndef _TYPE_H_
#define _TYPE_H_

#include <avr/pgmspace.h>

#ifndef NULL
#define NULL ((void *) 0)
#endif

typedef enum { false, true } bool;

#ifndef _SIZE_T
#define _SIZE_T
typedef unsigned int size_t;
#endif

typedef unsigned char BYTE; /* 8-bit value */
typedef unsigned char UCHAR; /* 8-bit value */
typedef unsigned int INT; /* 16-bit value */
typedef unsigned int UINT; /* 16-bit value */
typedef unsigned short USHORT; /* 16-bit value */
typedef unsigned short WORD; /* 16-bit value */
typedef unsigned long ULONG; /* 32-bit value */
typedef unsigned long DWORD; /* 32-bit value */

/* bsd */
typedef unsigned char u_char; /* 8-bit value */
typedef unsigned short u_short; /* 16-bit value */
typedef unsigned int u_int; /* 16-bit value */
typedef unsigned long u_long; /* 32-bit value */

typedef UCHAR SOCKET;

typedef union _un_l2cval {
    u_long lVal;
    u_char cVal[4];
}un_l2cval;

typedef union _un_i2cval {
    u_int iVal;
    u_char cVal[2];
}un_i2cval;

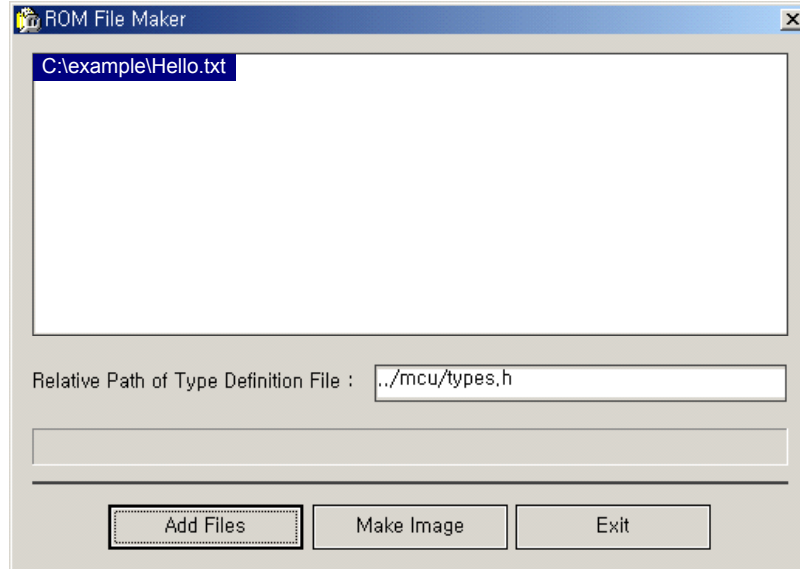
typedef prog_char code_area; ←
#endif /* _TYPE_H_ */

```

<Fig 3 EVB-A1's "types.h">

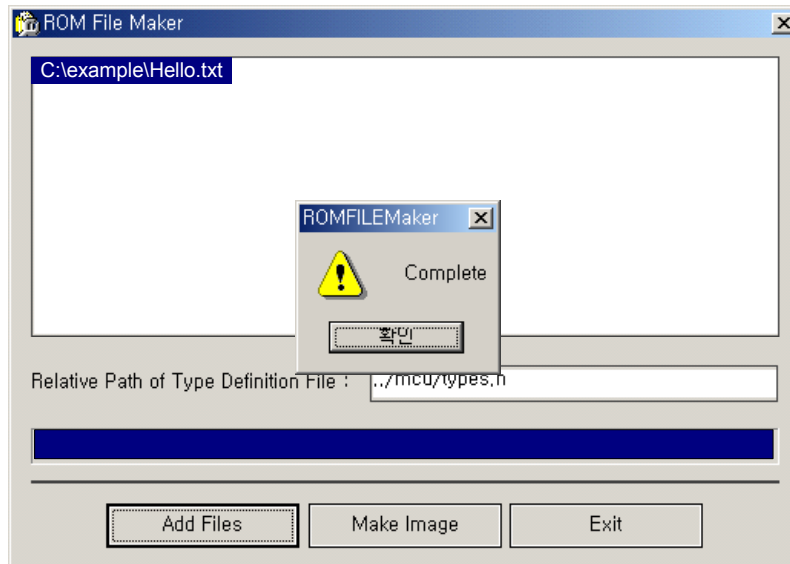
③ [Add File]을 Click한다.

"File Open Dialog"에서 'ROM File System'에 추가할 File들을 선택한다.



<Fig 4 'ROM File' add Files>

④ [Make Image]를 Click하여, 'ROM File System'을 생성한다.



<Fig 5 'ROM File System' Generation>

'ROM File System'은 "romfs.h" & "romfs.c"의 C Language Source File로 생성된다. 생성된 "romfs.h" & "romfs.c" 파일을 iinChip™ EVB F/W의 Project나 "Makefile"에 추가하고 Compile함으로써 쉽게 'ROM File System'을 구현할 수 있다.

```
#ifndef _ROMFS_H_
#define _ROMFS_H_

#include "../mcu/types.h"

typedef struct _ROMFILE ROMFILE;

struct _ROMFILE
{
    ROMFILE *romfs_next;    /* Link to next ROMFILE structure. */
    u_char *romfs_name;     /* Filename */
    u_int romfs_size;       /* File size. */
    code_area *romfs_data;  /* File contents. */
};

extern ROMFILE* romfs_list;

u_char search_file(u_char * name, code_area ** buf, u_int * len);  // Search a file from ROM FILE

#endif /* ROMFS H */
```

<Fig 6 "romfs.h">


```

/*
 * This file is automatically created by ROM File Maker
 */
#include <string.h>
#include "romfs.h"

/*
 * File 0 : D:\WIZCHIP_TEST_APP\ROMFileMaker\example\Hello.txt
 */
code_area file0data[] = {
0x48, 0x65, 0x6c, 0x6f, 0x2c, 0x20, 0x57, 0x49, 0x5a, 0x6e, 0x65, 0x74, 0x2e,
};
static ROMFILE file0entry = { 0, "Hello.txt", 14, (code_area *)file0data };

/*
Description : Search a file from ROM FILE
Argument    : name - file name
              buf - file contents to be return
              len - file length to be return
Return Value :
Note        :
*/
u_char search_file(u_char * name, code_area ** buf, u_int * len)
{
    int i;
    ROMFILE *romfs;

    i = 0;

    for (romfs = romfs_list; romfs; romfs = romfs->romfs_next)
    {
        if (!strcmp(name, romfs->romfs_name))
        {
            *len = romfs->romfs_size;
            *buf = romfs->romfs_data;
            return ++i;
        }
    }

    return 0;
}

```

ROM IMAGE →

Linked List of ROMFILE →

Search API →

<Fig 7 "romfs.c">