



杭州微元科技有限公司  
MuFTAD CRC MD5详细设计  
MU-KD-080004-3F-102

编 制	沈胜文
审 核	
批 准	
实施责任人	

### Amendment history 修改历史记录

版本号	修改说明	修改批准人	修改人	日期	签收人
101	创建文档		沈胜文	2008-3-21	
102	修改		沈胜文	2008-4-11	
103	修改	沈胜文	王可心	2008-5-7	



编 制	沈胜文
审 核	
批 准	
实施责任人	

Table of Contents 目录

1. Introduction 简介 .....	3
1.1. Objective 编写目的 .....	3
1.2. Background 背景 .....	3
1.3. Terms & Abberviation 术语&缩写 .....	3
1.4. Reference Material 参考资料 .....	4
2. Rules 规则 .....	4
2.1. Name Rules 命名规则 .....	4
2.2. Illuminate 说明 .....	4
2.3. Note Rules 注释规则 .....	5
2.4. File Structure 文件结构 .....	5
3. Structure Of Routines 程序结构 .....	6
3.1. Overview 总述 .....	6
3.2. Routines List 函数列表 .....	7
4. Global Description 全局描述 .....	7
4.1. Global Type 全局类型 .....	7
4.2. Global Error 全局错误码 .....	8
5. Routines Detail 函数细节 .....	9
5.1. Mu_Crc32File .....	9
5.2. Mu_Crc32Segment .....	14
5.3. Mu_Md5File .....	18
5.4. Mu_Md5Segment .....	23



编	制	沈胜文
审	核	
批	准	
实施责任人		

## 1. Introduction 简介

### 1.1. Objective 编写目的

本文档是在《CRC-32 MD5概要设计》的基础上，就文件或分段完整性校验所需要的扩展库，进行详细设计而完成的详细设计说明；

在 MuFTAD 项目中，涉及二种不同的校验方式：CRC 和 MD5，对此，在本设计文档中，将会详细说明二者的实现；

在概要设计文档中，介绍了待校验文件的内容，因此，针对具体的校验内容，必须提供具体的校验方法；

本文档将尽可能详尽地说明相关数据结构的设计和开发，但是设计和开发不相符之处，需讨论决定，并且修改本文档；

最终的设计以代码为准；

### 1.2. Background 背景

本程序是法电 MuFTAD 项目的一部分，提供内容完整性校验。但是程序按扩展库的形式组织和开发，以使得该程序独立于本项目，方便于其他项目的使用和移植；

本软件的提出者：沈胜文

本软件的开发者：沈胜文

本软件的用户：MuFTAD 和微元其他项目

### 1.3. Terms & Abberviation 术语&缩写

Terms&Abbreviation 术语&缩写	Description 解释
--------------------------	----------------



编 制	沈胜文
审 核	
批 准	
实施责任人	

CRC	循环冗余检验（Cyclic Redundant Check），使原始数据通过某种算法，得到一个新的数据，而这个数据与原数据有着固有的内在关系。把原数据和新数据组合在一起，使得新数据具有自我校验功能；
MD5	md5的全称是 message-digest algorithm 5，MD5的典型应用是对一段信息（Message）产生信息摘要（Message-Digest），以防止被篡改。
openssl	OpenSSL 是一个开放源代码的实现了 SSL 及相关加密技术的软件包，由加拿大的 Eric Yang 等发起编写的。

## 1.4. Reference Material 参考资料

CRC-32 MD5 概要设计  
AN42-循环冗余检验（CRC）原理与实现1.1.pdf  
并行 CRC\_32校验码生成算法研究及其实现.pdf  
openssl.org

## 2. Rules 规则

### 2.1. Name Rules 命名规则

该程序的建立不仅仅是为了 MuFTAD 项目，我们希望其能为微元其他项目提供服务，因此在开发过程中，对其按照扩展库的形式进行组织和开发，所有的函数形如：Mu\_XXXX()；

### 2.2. Illuminate 说明

针对每个程序，都必须注明其开发目的，开发者，开发时间，等等。以下字段必须被包含于程序的开头部分。

```
/*  
=====Microunit Techonogy Co.,LTD.=====  
* File Name:  
*  
*/
```



编 制	沈胜文
审 核	
批 准	
实施责任人	

```
*      CRC-32 & MD5.c
*
* Description:
*
*      Use the Routines contained in this file, to checksum the files ,
*      or the segment of the file.
*
* Revision History:
*
*      20-3-2008 ver1.0
*
* Author:
*
*      ssw  (fzqing@gmail.com)
*
*      ***PROTECTED BY COPYRIGHT***
*****/
```

## 2.3. Note Rules 注释规则

程序中的各个函数均需要明确注释其功能,并能简要描述其实现,及注意点。  
特别应该注意的是:在描述时,应该详细包括对锁,输入和输出进行详细说明。  
可参考模板

```
/******
*Description:
*      This Function checksum the crc-32,
*      Value pointer;
*Input:
*      filename: the file name , which stored programs
*Output:
*      Pointer:
*LOCK:
*      NONE
*Modify:
*      ssw (fzqing@gmail.com  10-3-2008)
*****/
```

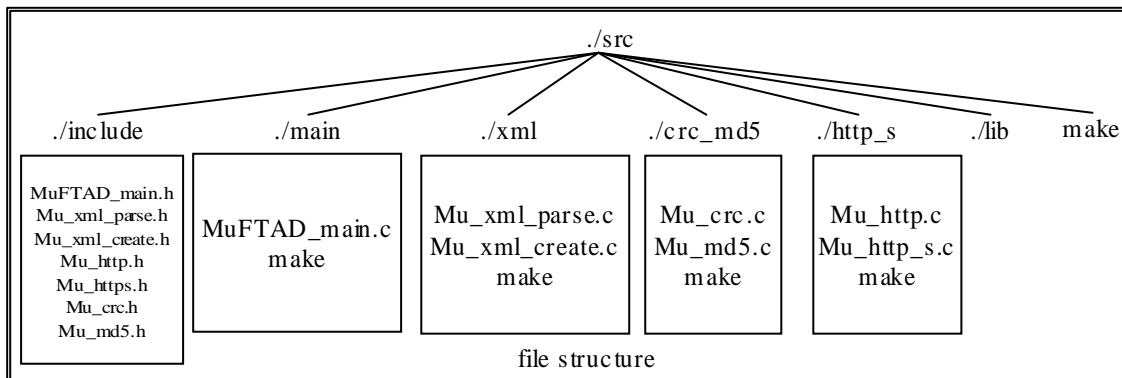
## 2.4. File Structure 文件结构

[注意]:

1、此为整个项目的文件组织方式;



编 制	沈胜文
审 核	
批 准	
实施责任人	



- ./src/include: 文件夹，包含该项目中的所有头文件；
- ./src/main: 文件夹，包含所有按法电《do3c00\_SoftProtocol\_0.1.0\_RC1》流程所开发的程序；
- ./src/xml: 文件夹，包含项目中所需要的 xml 处理库函数源代码；
- ./src/crc\_md5: 文件夹，包含项目中所需要的校验函数源代码，包括 CRC 和 MD5校验代码；
- ./src/http\_s: 文件夹，包含项目中所需要的与服务器交互的方式，包括 HTTP(s) GET、POST 方式；
- ./src/lib: 文件夹，用于存储编译所生成的 xml、http 和 https、crc/md5库。软件编译连接时使用该文件夹下的库；
- ./src/make: 文件，总的编译入口；

#### [注意]:

1、各对应文件夹下的源文件按需要添加，但是所作修改必须对 makefile 文件作相应的修改，以正确编译；

## 3. Structure Of Routines 程序结构

### 3.1. Overview 总述

待开发的各程序之间是相互独立的，它们为项目 MuFTAD 中的其他部分服务，按功能可以分为如下二类：

#### 1、CRC-32校验

使用 CRC-32对完整的文件或是文件分段进行完整性校验；

#### 2、MD5校验

使用 MD5散列算法，对完整的文件或是文件分段进行完整性校验；



编 制	沈胜文
审 核	
批 准	
实施责任人	

## 3.2. Routines List 函数列表

# 4. Global Description 全局描述

## 4.1. Global Type 全局类型

### 4.1.1. Macro 宏定义

```
#define MAX_CRC_LEN 1023  
#define unit_fast32_t unsigned int
```

### 4.1.2. Structure 结构体

```
static uint_fast32_t CRC32_Tab[256] = {  
0x0,  
0x77073096, 0xEE0E612C, 0x990951BA, 0x076DC419, 0x706AF48F,  
0xE963A535, 0x9E6495A3, 0x0EDB8832, 0x79DCB8A4, 0xE0D5E91E,  
0x97D2D988, 0x09B64C2B, 0x7EB17CBD, 0xE7B82D07, 0x90BF1D91,  
0x1DB71064, 0x6AB020F2, 0xF3B97148, 0x84BE41DE, 0x1ADAD47D,  
0x6DDDE4EB, 0xF4D4B551, 0x83D385C7, 0x136C9856, 0x646BA8C0,  
0xFD62F97A, 0x8A65C9EC, 0x14015C4F, 0x63066CD9, 0xFA0F3D63,  
0x8D080DF5, 0x3B6E20C8, 0x4C69105E, 0xD56041E4, 0xA2677172,  
0x3C03E4D1, 0x4B04D447, 0xD20D85FD, 0xA50AB56B, 0x35B5A8FA,  
0x42B2986C, 0xDBBBC9D6, 0xACBCF940, 0x32D86CE3, 0x45DF5C75,  
0xDCD60DCF, 0xABD13D59, 0x26D930AC, 0x51DE003A, 0xC8D75180,  
0xBFDD06116, 0x21B4F4B5, 0x56B3C423, 0xCFBA9599, 0xB8BDA50F,  
0x2802B89E, 0x5F058808, 0xC60CD9B2, 0xB10BE924, 0x2F6F7C87,  
0x58684C11, 0xC1611DAB, 0xB6662D3D, 0x76DC4190, 0x01DB7106,  
0x98D220BC, 0xEFD5102A, 0x71B18589, 0x06B6B51F, 0x9FBBE4A5,  
0xE8B8D433, 0x7807C9A2, 0x0F00F934, 0x9609A88E, 0xE10E9818,  
0x7F6A0DBB, 0x086D3D2D, 0x91646C97, 0xE6635C01, 0xB6B51F4,  
0x1C6C6162, 0x856530D8, 0xF262004E, 0x6C0695ED, 0x1B01A57B,  
0x8208F4C1, 0xF50FC457, 0x65B0D9C6, 0x12B7E950, 0x8BBEB8EA,  
0xFCB9887C, 0x62DD1DDF, 0x15DA2D49, 0x8CD37CF3, 0xFBD44C65,  
0x4DB26158, 0x3AB551CE, 0xA3BC0074, 0xD4BB30E2, 0x4ADFA541,  
0x3DD895D7, 0xA4D1C46D, 0xD3D6F4FB, 0x4369E96A, 0x346ED9FC,  
0xAD678846, 0xDA60B8D0, 0x44042D73, 0x33031DE5, 0xAA0A4C5F,
```



编 制	沈胜文
审 核	
批 准	
实施责任人	

0xDD0D7CC9, 0x5005713C, 0x270241AA, 0xBE0B1010, 0xC90C2086,  
0x5768B525, 0x206F85B3, 0xB966D409, 0xCE61E49F, 0x5EDEF90E,  
0x29D9C998, 0xB0D09822, 0xC7D7A8B4, 0x59B33D17, 0x2EB40D81,  
0xB7BD5C3B, 0xC0BA6CAD, 0xEDB88320, 0x9ABFB3B6, 0x03B6E20C,  
0x74B1D29A, 0xEAD54739, 0x9DD277AF, 0x04DB2615, 0x73DC1683,  
0xE3630B12, 0x94643B84, 0x0D6D6A3E, 0x7A6A5AA8, 0xE40ECF0B,  
0x9309FF9D, 0x0A00AE27, 0x7D079EB1, 0xF00F9344, 0x8708A3D2,  
0x1E01F268, 0x6906C2FE, 0xF762575D, 0x806567CB, 0x196C3671,  
0x6E6B06E7, 0xFED41B76, 0x89D32BE0, 0x10DA7A5A, 0x67DD4ACC,  
0xF9B9DF6F, 0x8EBEEFF9, 0x17B7BE43, 0x60B08ED5, 0xD6D6A3E8,  
0xA1D1937E, 0x38D8C2C4, 0x4FDDFF252, 0xD1BB67F1, 0xA6BC5767,  
0x3FB506DD, 0x48B2364B, 0xD80D2BDA, 0xAF0A1B4C, 0x36034AF6,  
0x41047A60, 0xDF60EFC3, 0xA867DF55, 0x316E8EEF, 0x4669BE79,  
0xCB61B38C, 0xBC66831A, 0x256FD2A0, 0x5268E236, 0xCC0C7795,  
0xBB0B4703, 0x220216B9, 0x5505262F, 0xC5BA3BBE, 0xB2BD0B28,  
0x2BB45A92, 0x5CB36A04, 0xC2D7FFA7, 0xB5D0CF31, 0x2CD99E8B,  
0x5BDEAE1D, 0x9B64C2B0, 0xEC63F226, 0x756AA39C, 0x026D930A,  
0x9C0906A9, 0xEB0E363F, 0x72076785, 0x05005713, 0x95BF4A82,  
0xE2B87A14, 0x7BB12BAE, 0x0CB61B38, 0x92D28E9B, 0xE5D5BE0D,  
0x7CDCEFB7, 0x0BDBDF21, 0x86D3D2D4, 0xF1D4E242, 0x68DDB3F8,  
0x1FDA836E, 0x81BE16CD, 0xF6B9265B, 0x6FB077E1, 0x18B74777,  
0x88085AE6, 0xFF0F6A70, 0x66063BCA, 0x11010B5C, 0x8F659EFF,  
0xF862AE69, 0x616BFFD3, 0x166CCF45, 0xA00AE278, 0xD70DD2EE,  
0x4E048354, 0x3903B3C2, 0xA7672661, 0xD06016F7, 0x4969474D,  
0x3E6E77DB, 0xAED16A4A, 0xD9D65ADC, 0x40DF0B66, 0x37D83BF0,  
0xA9BCAE53, 0xDEBB9EC5, 0x47B2CF7F, 0x30B5FFE9, 0xBDBDF21C,  
0xCABAC28A, 0x53B39330, 0x24B4A3A6, 0xBAD03605, 0xCDD70693,  
0x54DE5729, 0x23D967BF, 0xB3667A2E, 0xC4614AB8, 0x5D681B02,  
0x2A6F2B94, 0xB40BBE37, 0xC30C8EA1, 0x5A05DF1B, 0x2D02EF8D  
};

[注意]:

- 1、该校验表由CRC32算法多项式生成, 见文件 *Mu\_crc.c*, 方法: *crcgen()*。
- 2、使用上表, *crcgen()* 方法默认被注释掉; 或开启注释, 清空校验表, 在运行中生成。

## 4.2. Global Error 全局错误码

[注意]:

- 1、该错误码应该被放在一个单独的, 只用来定义错误的头文件中;
- 2、该头文件定义为: *mu\_error.h*

```
#define MUOK 0
```





编	制	沈胜文
审	核	
批	准	
实施责任人		

```
#define MUFLLEN -18      //File over length
#define MUFRAD -19      //Read file error
#define MUCNOM -20      //CRC32 or MD5 code not match
```

## 5. Routines Detail 函数细节

### 5.1. Mu\_Crc32File

#### 5.1.1. Name 函数名称

```
Void crcgen();

int Mu_Crc32File(int fd, unsigned int crc32)
```

#### 5.1.2. Description 函数描述

本函数用于校验文件的完整性，使用 CRC32方式完成。函数对 fp 文件句柄所关联的文件内的所有内容，与 crc32一并进行 CRC32运算，返回校验状态；

本函数对进行校验的文件不作任何修改，并且在校验完毕后，也不关闭该文件句柄，该操作由调用者完成；

为了提高校验效率，使用查表法完成，CRC32表见4.1.章节所述；

#### 5.1.3. Function 功能

用 CRC-32的校验方式对整个文件内容进行校验，以函数返回值的形式返回校验结果；

#### 5.1.4. Capability 性能

产用查表方式，以提高 CRC-32的校验速度；



编 制	沈胜文
审 核	
批 准	
实施责任人	

### 5.1.5. Input 输入

fp: 待校验文件的文件句柄;

crc32: 待校验文件的 CRC 值, 同原文件一起校验文件的完整性;

### 5.1.6. Output 输出

校验状态值;

[注意]:

1、可以参考4.2. 章节;

### 5.1.7. Arithmetic 算法

CRC-32 查表法在实现原理:

本字节后的 CRC 码等于;

核心代码如下:

```
crc = ((crc>>8) & 0x00FFFFFF) ^ CRC32_Tab[(crc^buf[i])&0xFF];
```

```
crc= crc^0xFFFFFFFF;
```

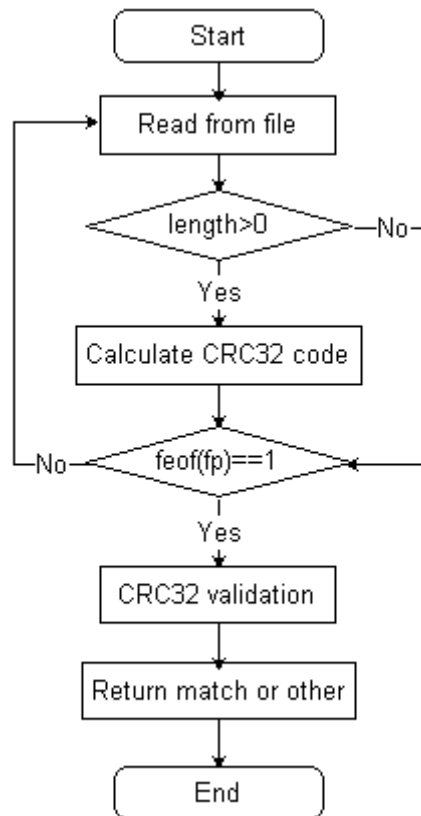
[注意]:

1、本章节所说的和是指机器语言中的异或运算;



编 制	沈胜文
审 核	
批 准	
实施责任人	

### 5.1.8. Process Flow 处理流程



..

### 5.1.9. Pseudocode 伪代码

```
/*  
* CRC32 check  
*/  
  
int Mu_Crc32File(int fd, unsigned int crc32){  
  
    unsigned char buf[MAX_CRC_LEN + 1];  
  
    unsigned int crc = 0xFFFFFFFF;  
  
    unsigned char *cp = NULL;  
  
    cp=buf;
```



编 制	沈胜文
审 核	
批 准	
实施责任人	

```
int c;

int i;

unsigned char temp;

size_t bytes_read;

size_t length = 0;

memset(buf, 0, MAX_CRC_LEN + 1);

//Call crcgen() method

//crcgen();

if(length + bytes_read < length){

    return MUFLLEN;

}

while(bytes_read=read(fd,buf,MAX_CRC_LEN)){

    length+=bytes_read;

    for(i=0;i<bytes_read;i++){

        crc = ((crc>>8) & 0x00FFFFFF) ^ CRC32_Tab[(crc^buf[i])&0xFF ];

    }

    if(bytes_read==0){

        break;

    }

}

//Return

if(crc32==(crc^0xFFFFFFFF)){

    return MUOK;
```



编	制	沈胜文
审	核	
批	准	
实施责任人		

```
}else{  
  
    return MUCNOM;  
}  
  
//Default return  
  
return MUFRAD;  
}
```

### 5.1.10. Interface 接口

略

### 5.1.11. Malloc 存储分配

函数使用局部变量，申请栈中空间，函数将文件中内容读取到该空间，进行CRC 校验，在结束后，自动将内存区域归还系统；

### 5.1.12. Restrict 限制

略

### 5.1.13. Test 测试

准备一个待检测的文件，用工具得到其 CRC32值；  
写一个单独的调用函数，按函数规定赋给参数，测试函数是否校验正确；  
Demo 文件夹下测试程序：demo

### 5.1.14. Unsolve 未解决情况

略



编 制	沈胜文
审 核	
批 准	
实施责任人	

## 5.2. Mu\_Crc32Segment

### 5.2.1. Name 函数名称

`int Mu_Crc32Segment(int fd, off_t seek, size_t char_length, unsigned int crc32)`

### 5.2.2. Description 函数描述

本函数用于校验文件分段的完整性，使用 CRC32 方式完成。函数对 fp 文件句柄所关联的文件内，从 seek 标识位置开始，length 长度的内容及 crc32 的值一并进行 CRC32 运算，返回校验状态；

本函数对进行校验的文件不作任何修改，并且在校验完毕后，也不关闭该文件句柄，该操作由调用者完成；

为了提高校验效率，使用查表法完成，CRC32 表见 4.1 章节所述；

### 5.2.3. Function 功能

用 CRC-32 的校验方式对文件内的部分内容进行校验，以函数返回值的形式返回校验结果；

### 5.2.4. Capability 性能

采用查表方式，以提高 CRC-32 的校验速度；

### 5.2.5. Input 输入

fp: 待校验文件的文件句柄；

crc32: 待校验文件的 CRC 值，同原文件一起校验文件的完整性；

seek: 标识待检验分段在文件中的起始位置；

length: 标识进行检验时，需要输入的总的字符数；



编 制	沈胜文
审 核	
批 准	
实施责任人	

## 5.2.6. Output 输出

校验状态值;

[注意]:

1、可以参考4.2.章节;

## 5.2.7. Arithmetic 算法

同5.1.7.中说明

## 5.2.8. Process Flow 处理流程

同5.1.8.中说明

[注意]:

1、在文件读取上稍的不同;

## 5.2.9. Pseudocode 伪代码

```
//CRC32 code segment check
```

```
int Mu_Crc32Segment(int fd, off_t seek, size_t char_length, unsigned int crc32){
```

```
    unsigned int crc = 0xFFFFFFFF;
```

```
    unsigned char buf[MAX_CRC_LEN + 1];
```

```
    unsigned char *cp = NULL;
```

```
    size_t bytes_read;
```

```
    size_t length = 0;
```

```
    int i;
```

```
    memset(buf, 0, MAX_CRC_LEN + 1);
```

```
    //crcgen();
```



杭州微元科技有限公司  
MuFTAD CRC MD5详细设计  
MU-KD-080004-3F-102

编 制	沈胜文
审 核	
批 准	
实施责任人	

```
//Location of read start

lseek(fd,seek,SEEK_SET);

while(bytes_read = read(fd,buf,MAX_CRC_LEN)){

    cp = buf;

    //Keep the char_length for check

    while(length + bytes_read > char_length){

        bytes_read--;

    }

    length += bytes_read;

    for(i=0;i<bytes_read;i++){

        crc = ((crc>>8) & 0x00FFFFFF) ^ CRC32_Tab[(crc^buf[i])&0xFF ];

    }

    //Stop while loop when 'char_length' matched

    if(bytes_read==char_length){

        break;

    }

    if(bytes_read==0){

        break;

    }

}

//Return

if(crc32==(crc^0xFFFFFFFF)){

    return MUOK;
```





编	制	沈胜文
审	核	
批	准	
实施责任人		

```
}else{  
    return MUCNOM;  
}  
  
//Default return  
  
return MUFRAD;  
}
```

#### 5.2.10. Interface 接口

略

#### 5.2.11. Malloc 存储分配

同5.1.11.中说明

#### 5.2.12. Restrict 限制

略

#### 5.2.13. Test 测试

同5.1.13.中说明

#### 5.2.14. Unsolve 未解决情况

略



编	制	沈胜文
审	核	
批	准	
实施责任人		

## 5.3. Mu\_Md5File

### 5.3.1. Name 函数名称

int Mu\_Md5File(int fd, char \*md5)

### 5.3.2. Description 函数描述

本函数使用 MD5方式校验文件的完整性。函数对 fd 文件句柄所关联的文件内的所有内容进行 MD5运算，将运算结果与参数值 md5比较，返回校验状态；

本函数对进行校验的文件不作任何修改，并且在校验完毕后，也不关闭该文件句柄，该操作由调用者完成；

MD5校验使用 openssl 库完成；

### 5.1.3. Function 功能

用 MD5的校验方式对整个文件内容进行校验，以函数返回值的形式返回校验结果；

### 5.1.4. Capability 性能

略；

### 5.1.5. Input 输入

fp: 待校验文件的文件指针；

md5: 待校验文件的 MD5值，同原文件一起校验文件的完整性；

### 5.1.6. Output 输出

校验状态值；

[注意]:



编 制	沈胜文
审 核	
批 准	
实施责任人	

1、可以参考4.2. 章节;

### 5.1.7. Arithmetic 算法

选择调用系统 Openssl MD5库完成，选择通用 MD5库完成，本模块采用第一种。

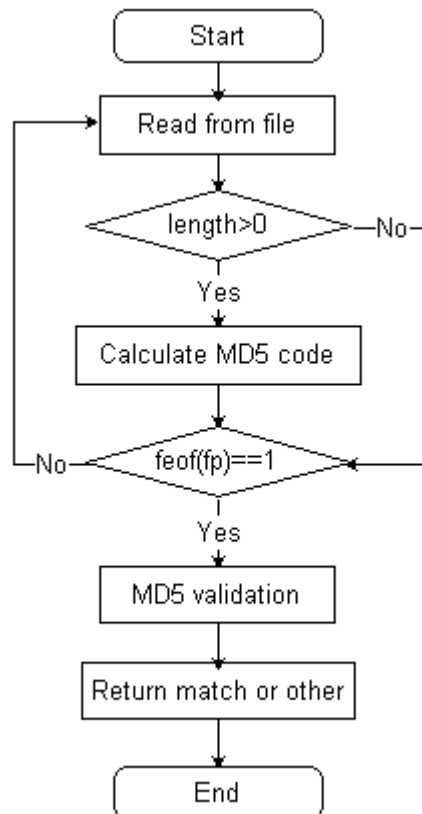
核心代码:

```
for (i = 0; i < 16; i++){  
  
    sprintf(&(output1[2*i]),"%02x",(unsigned char)digest[i]);  
  
    sprintf(&(output1[2*i+1]),"%02x",(unsigned char)(digest[i]<<4));  
  
}  
  
for(i=0;i<32;i++){  
  
    output[i]=output1[i];  
  
}
```



编 制	沈胜文
审 核	
批 准	
实施责任人	

### 5.1.8. Process Flow 处理流程



### 5.1.9. Pseudocode 伪代码

```
/*  
 * MD5 check  
 */  
  
//Default include  
  
#include <stdio.h>  
  
#include <string.h>  
  
#include <sys/types.h>  
  
//Md5 library
```



杭州微元科技有限公司  
MuFTAD CRC MD5详细设计  
MU-KD-080004-3F-102

编 制	沈胜文
审 核	
批 准	
实施责任人	

```
#include <openssl/md5.h>

//SSL library

#include <openssl/ssl.h>

//Global define head file

#include "Mu_md5.h"

//Error code define head file

#include "Mu_error.h"

//Files's MD5 code check

int Mu_Md5File(int fd, char *md5){

    char output[33] = {" "};

    char output1[32];

    MD5_CTX context;

    size_t length = 0;

    size_t bytes_read;

    int retval = MUCNOM;

    unsigned char buf[1023 + 1];

    unsigned char digest[MD5_DIGEST_LENGTH];

    MD5_Init(&context);

    memset(buf, 0, 1023 + 1);

    while (bytes_read = read (fd,buf,1024)){

        MD5_Update (&context, buf, bytes_read);

        if(bytes_read==0){

            break;
```



编 制	沈胜文
审 核	
批 准	
实施责任人	

```
    }  
}  
  
    MD5_Final (digest, &context);  
  
    int i;  
  
    for (i = 0; i < 16; i++){  
  
        sprintf(&(output1[2*i]),"%02x",(unsigned char)digest[i]);  
  
        sprintf(&(output1[2*i+1]),"%02x",(unsigned char)(digest[i]<<4));  
  
    }  
  
    for(i=0;i<32;i++){  
  
        output[i]=output1[i];  
  
    }  
  
    //Uppercase  
  
    ToUpper(output);  
  
    ToUpper(md5);  
  
    retval = strncmp(output, md5, 32);  
  
    //Return  
  
    if(retval==0){  
  
        return MUOK;  
  
    }else{  
  
        return retval;  
  
    }  
  
    return MUFRAD;  
}
```



编 制	沈胜文
审 核	
批 准	
实施责任人	

### 5.3.10. Interface 接口

略

### 5.3.11. Malloc 存储分配

同5.1.11.中说明

### 5.3.12. Restrict 限制

略

### 5.3.13. Test 测试

同5.1.13.中说明

### 5.3.14. Unsolve 未解决情况

略

## 5.4. Mu\_Md5Segment

创建函数接口

```
int Mu_Md5Segment(int fd, off_t seek, size_t length, char *md5)
```

具体实现参考5.2和5.3.章节