



复旦微电子

FM33 系列 MCU

Cortex-M0+内核自测软件库

应用手册

V1.0.0



本资料是为了让用户根据用途选择合适的上海复旦微电子集团股份有限公司（以下简称复旦微电子）的产品而提供的参考资料，不转让属于复旦微电子或者第三者所有的知识产权以及其他权利的许可。

在使用本资料所记载的信息最终做出有关信息和产品是否适用的判断前，请您务必将所有信息作为一个整体系统来进行评价。

采购方对于选择与使用本文描述的复旦微电子的产品和服务全权负责，复旦微电子不承担采购方选择与使用本文描述的产品和服务的责任。除非以书面形式明确地认可，复旦微电子的产品不推荐、不授权、不担保用于包括军事、航空、航天、救生及生命维持系统在内的，由于失效或故障可能导致人身伤亡、严重的财产或环境损失的产品或系统中。

未经复旦微电子的许可，不得翻印或者复制全部或部分本资料的内容。

今后日常的产品更新会在适当的时候发布，恕不另行通知。在购买本资料所记载的产品时，请预先向复旦微电子在当地的销售办事处确认最新信息，并请您通过各种方式关注复旦微电子公布的信息，包括复旦微电子的网站(<http://www.fmsh.com/>)。

如果您需要了解有关本资料所记载的信息或产品的详情，请与上海复旦微电子集团股份有限公司在当地的销售办事处联系。

商标

上海复旦微电子集团股份有限公司的公司名称、徽标以及“复旦”徽标均为上海复旦微电子集团股份有限公司及其分公司在中国的商标或注册商标。

上海复旦微电子集团股份有限公司在中国发布，版权所有。



目录

1 简介	2
2 功能说明	2
3 内核自测软件库参数	3
3.1 存储占用空间	3
3.2 执行时间	3
4 内核自测软件库集成到用户应用系统	4
4.1 内核自测软件库相关文件	4
4.1.1 内核自测软件库的测试项	4
4.1.2 内核自测软件库中断向量表和服务程序	4
4.1.3 内核自测软件库的功能接口	5
4.2 中断(异常)处理	6
4.3 汇编文件的编译	6
4.4 链接器文件关于自定义区域的存储分配	7
5 内核自测软件库与用户应用程序的交互	8
5.1 内核自测软件库的功能接口	8
5.2 内核自测软件库的栈使用情况	9
5.3 内核自测软件库功能接口函数的测试结果处理	9
5.4 支持对单个测试项故意返回错误签名	10
6 数据类型	10
7 全局变量	11
7.1 m0_scst_accumulated_signature 全局变量	11
7.2 m0_scst_last_executed_test_number 全局变量	11
7.3 m0_scst_fault_inject_test_index 全局变量	12
8 功能接口函数	12
8.1 m0_scst_execute_core_tests 函数	12
9 内核各测试项信息	13
版本信息	17
上海复旦微电子集团股份有限公司销售及服	18

1 简介

本文件包含了 FM33 系列 MCU Cortex-M0+平台的内核自测软件库的接口信息和测试项信息，并描述了内核自测软件库可能返回的测试结果。该软件库包含代码和数据部分，应用了中断服务资源。

2 功能说明

- 内核自测软件库提供相关测试项，从而实现与安全功能相关的故障诊断覆盖率。
- 内核自测软件库可以周期性的执行，如在 FTTI 内执行，则针对 SPF 的检测将予以有效的帮助。该自测软件库保留了应用程序和设备配置的相关执行环境。
- 内核自测软件库的测试涵盖了 Cortex-M0+内核的大部分指令，以及针对内核特定 IP 块的功能测试：
 - 内核控制逻辑（分支控制、异常控制）
 - 内核数据路径，包括：
 - 通用寄存器和特殊寄存器；
 - ALU、乘法器, 加载/存储和其他执行单元；
 - 指令解码器，16 位指令和 32 位指令；
- 在内核自测软件库执行过程中，部分测试项会使用中断服务，自测软件库为异常测试提供了自己的中断向量表和服务程序。在异常测试期间，自测软件库禁用了所有的外部中断(含用户系统)，由自测软件库引起的用户 ISR 调用延迟时间在表 9-1 中规定。在测试过程中发生的且自测软件库检测到非预期的异常，则该异常会基于用户向量表被重定位给用户应用程序对应的异常服务。

3 内核自测软件库参数

3.1 存储占用空间

内核自测软件库的存储占用空间参数见表 3-1 中规定。注意，测量是针对自测软件库支持的某一种编译器进行，不同编译器因对齐要求不同、编译等级不同等，测试结果会有细微差别。

表 3-1 内核自测软件库的存储占用空间

栈(bytes)	静态 RAM(bytes)	Flash(bytes)
88	152	5576

3.2 执行时间

内核自测软件库的执行时间见表 3-2 中规定，具体详细时间取决于功能接口数调用方式。

每个单独测试项的执行时间以及用户 ISR 调用的延迟时间在表 9-1 中规定。

表 3-2 内核自测软件库的执行时间表

功能接口函数调用方式	测试执行时间 (us @ 48 MHz)
<code>result=m0_scst_execute_core_tests(0, 18)</code> 用户应用程序通过功能接口函数一次完成所有测试项测试， 但该功能函数内部实际仍是逐个调用独立测试项进行的。	306, 41
<code>result=m0_scst_execute_core_tests(0, 0)</code> <code>result=m0_scst_execute_core_tests(1, 1)</code> <code>result=m0_scst_execute_core_tests(18, 18)</code> 用户应用程序通过功能接口函数一次完成一个测试项测试， 直至所有测试项全部完成。	343, 60



4 内核自测软件库集成到用户应用系统

4.1 内核自测软件库相关文件

内核自测软件库文件由三部分组成：

- 针对内核的独立测试项，一个测试项一个文件。
- 为独立测试项服务的相关环境文件，如自测软件库的中断向量表、自测软件库的中断服务程序，以及用于用户应用程序和内核自测两种模式切换的关于内核数据的保存及恢复等。

自测软件库的中断服务程序用于捕捉和处理自测软件库的预期异常，如产生非预期的异常，自测软件库则将该中断服务重定向到用户应用程序对应的中断服务。

- 自测软件库对外的应用接口文件。

下面的章节提供了各部分文件的相关概述，用户可以将这些文件的名称和相对路径添加到项目中。

4.1.1 内核自测软件库的测试项

内核自测软件库提供了一组测试项，从而实现针对 Cortex-M0+内核的相关诊断覆盖率，表 4-1 提供了针对 Cortex-M0+内核的测试项相关文件：

表 4-1 内核测试项相关文件

文件	描述
\SCST_Library\src\asm\core_tests*.S	内核测试项，一个测试项一个独立汇编文件。
\SCST_Library \src\asm\m0_scst_lib.S	内核测试项所需变量、常量的定义，以及退出内核测试模式的代码等。

4.1.2 内核自测软件库中断向量表和服务程序

内核自测软件库包括并分配了自己的中断向量表，且提供了一套完整的 ISR 来捕获所有的预期中断或异常。下面的表 4-2 提供了对应相关源文件的概述：

表 4-2 内核自测软件库中断向量表和服务程序相关文件

文件	描述
\SCST_Library\src\asm\m0_scst_exception_wrappers.S	内核自测软件库的异常服务程序
\SCST_Library\src\asm\m0_scst_exception_lib.S	用户应用程序和内核自测两种模式切换时关于内核数据的保存及恢复
\SCST_Library\src\asm\m0_scst_vector_table.S	内核自测软件库特定的中断向量表

4.1.3 内核自测软件库的功能接口

内核自测软件库提供了执行内核测试的功能接口，用于用户集成到其系统后调用对应功能函数，表 4-3 列出了功能接口相关文件。

表 4-3 内核自测库功能接口相关文件

文件	描述
\SCST_Library\src\c\m0_scst_test_shell.c	执行内核测试的功能接口函数
\SCST_Library \src\c\m0_scst_data.c	内核测试项列表
\SCST_Library\src\h\m0_scst_data.h	内核测试软件库的数据类型定义
\SCST_Library\src\h\m0_scst_test_shell.h	m0_scst_test_shell.c 的头文件，包含函数原型和可能的返回值
\SCST_Library\src\h\m0_scst_configuration.h	关于内核寄存器及外设的配置定义
\SCST_Library\src\h\m0_scst_typedefs.h	编译器的数据类型定义
\SCST_Library\src\h\m0_scst_compiler.h	包含编译器关键字的抽象宏

4.2 中断(异常)处理

内核自测软件库包含一系列的测试项来检查内核关于中断异常的逻辑功能是否正确, 这些测试将使用内核自测软件库自己特定的中断向量表, 然后执行触发异常条件的测试代码, 并检查是否能正常进入及退出异常服务, 以及异常服务的处理程序能否返回预期值。当该测试项执行完毕后, 内核自测软件库会恢复原来应用程序/操作系统所使用的向量表,

在执行测试过程中, 部分测试项的测试程序会破坏内核的特殊寄存器、配置寄存器和控制寄存器等, 为了支持这些类型的测试执行, 内核自测软件库在破坏这些专用寄存器之前将禁止用户应用中断处理, 但为了不影响用户中断处理的及时性, 测试项在测试执行过程中设立了几个“窗口”, 用于及时恢复专用寄存器, 从而不影响用户中断处理的及时性。自测库引起的用户 ISR 调用延迟时间见表 9-1 中的规定。

内核自测软件库的异常服务程序位于自定义区域的.m0_scst_test_code 存储区域中, 自测库的特定向量表位于自定义区域的.m0_scst_vector_table 存储区域中。

4.3 汇编文件的编译

大多数汇编编译器支持汇编文件包含 C 指令, 因此, 在汇编文件编译之前, 它们必须由 C 语言编译器进行预处理。内核自测软件库的汇编文件使用了 C 语言定义的预处理符号。

当使用 IAR 编译器时, 预处理会自动完成, 不需要编译器选项。

当使用 GreenHills 编译器时, 应使用以下选项:

-预处理_assembly_files

当使用 GCC 编译器时, 应使用以下选项:

-x assembler-with-cpp

当使用 WindRiver Diab 编译器时, 应使用下列选项:

-Xpreprocess-assembly

-Xalign-power2

注 1: 如果不使用, -Xalign-power2 选项是 WindRiver Diab 编译器的默认值。

注 2: IAR 编译器, 根据版本的不同, 可能会报 410 告警, 在编译器选项中关闭该告警即可。

IAR 编译器版本建议至少大于 V8.11.2。

4.4 链接器文件关于自定义区域的存储分配

内核自测软件库的所有代码、常量、变量和数据结构都被放置在 C 代码和汇编代码指定的自定义区域，表 4-4 提供了内核自测软件库所有自定义区域的名称，用户应该通过链接器文件引用这些自定义区域名称，从而实现在设备存储器中正确分配这些数据和代码。

表 4-4 内核自测软件库的自定义区域

自定义区域名称	目标存储器	描述
.m0_scst_test_code	RAM 或 FLASH	汇编代码，内核所有测试项的执行代码。
.m0_scst_test_shell_code	RAM 或 FLASH	C 代码，集成到用户系统中的功能接口代码。
.m0_scst_vector_table	FLASH, 或由代码复制 到对应 RAM	内核自测软件库特定的中断向量表
.m0_scst_rom_data	FLASH, 或由代码复制 到对应 RAM	C 代码，内核自测软件库测试项的函数地址及签名列表。
.m0_scst_ram_data	RAM	各测试项的全局变量及数据
.m0_scst_test_shell_data	RAM	C 代码，集成到用户系统中的功能接口的全局变量及数据
.m0_scst_ram_data_target0	RAM	该区域由加载/存储测试访问。可以位于任何地址，具体取决于可用内存和测试特定目标地址的动机。
.m0_scst_ram_data_target1	RAM	该区域由加载/存储测试访问。可以位于任何地址，具体取决于可用内存和测试特定目标地址的动机。
.m0_scst_ram_test_code	RAM	该区域用于 fetch 测试执行的代码。可以位于任何地址，具体取决于可



		用内存。
--	--	------

表 4-5 内核自测软件库的自定义区域存储空间

自定义区域名称	自定义区域空间 (bytes)
.m0_scst_test_code	5088
.m0_scst_test_shell_code	144
.m0_scst_vector_table	192
.m0_scst_rom_data	152
.m0_scst_ram_data	20
.m0_scst_test_shell_data	16
.m0_scst_ram_data_target0	52
.m0_scst_ram_data_target1	52
.m0_scst_ram_test_code	12

5. 内核自测软件库与用户应用程序的交互

5.1 内核自测软件库的功能接口

为了使用内核自测软件库，客户的应用程序中必须包含以下头文件：

- m0_scst_test_shell.h

在正常运行过程中，用户应用程序应通过调用 `m0_scst_execute_core_tests()` 函数与内核自测软件库进行交互，该函数由内核自测软件库的功能接口文件提供，由两个函数形参来指定要执行的测试项索引范围，该函数按顺序逐个执行测试项，如所有测试项均成功执行，则最终返回一个 32 位的值作为所有测试项成功测试的执行结果。如果中间有测试项失败，则停止执行剩余的测试项，并返回测试失败值或执行测试失败动作，具体由各测试项决定。

测试项的执行顺序和调用时间由用户应用程序决定。

`m0_scst_execute_core_tests()` 函数应在与请求执行测试项相匹配的处理/权限模式下调用(每个测试项所需的处理/权限模式在表 9-2 中指定)。内核自测软件库在调用结束后以相同的处理/权限模式将控制权(CONTROL 特殊寄存器)返回给用户应用程序。

内核自测软件库在将控制权返回给用户应用程序之前,会恢复所有被测试项破坏的通用寄存器、特殊寄存器以及控制和配置寄存器的初始内容。

5.2 内核自测软件库的栈使用情况

内核自测软件库使用了用户应用程序的栈。首先, `m0_scst_execute_core_tests()` 函数采用 C 语言编写,因此 C 编译器生成汇编代码时会自动会使用到栈,内核自测软件库整体栈的使用大小请参见表 3-1 规定。每个独立测试项也使用到栈,各独立测试项所使用的栈大小请参见表 9-2 规定。

5.3 内核自测软件库功能接口函数的测试结果处理

即使在一次调用 `m0_scst_execute_core_tests()` 接口函数时有多个测试项被执行,接口函数内仍会对每个独立测试项的测试结果进行检查,每个独立测试项执行后均返回一个 32 位的签名,如果返回的签名与预期值不匹配,则接口函数将停止剩余测试项的执行,并返回该测试项错误执行结果,此次执行失败的测试项编号存储在 `m0_scst_last_executed_test_number` 全局变量中。

接口函数管理一个 32 位的组合签名,在每一个独立测试项执行成功后立即更新。这个组合签名的计算方式为对每个独立测试项执行成功后返回的签名值进行 XOR 运算,如果所有的独立测试项均成功执行,则 `m0_scst_execute_core_tests()` 接口函数将这个组合签名返回给用户应用程序。

用户应用程序首先应确定每个独立测试项执行成功并获取其返回的签名值(各独立测试项的参考签名请参见表 9-2 规定),然后根据需要执行的独立测试项对其签名值进行 XOR 运算以获得预期的组合签名,这样在调用 `m0_scst_execute_core_tests()` 接口函数后对返回的组合签名与预期组合签名进行比较,如相等则代表所有的独立测试项均已执行成功。

内核自测库以 `m0_scst_accumulated_signature` 全局变量的形式提供了一个冗余的测试执行结果给用户应用程序，如所有测试项均成功执行，该全局变量的值与 `m0_scst_execute_core_tests()` 接口函数返回的组合签名值一致，用户应用程序可做冗余判断。

5.4 支持对单个测试项刻意返回错误签名

为了测试用户应用程序在内核自测软件库功能接口函数返回不正确执行结果时的反应，内核自测软件库支持根据编号索引对指定的测试项返回错误的签名，从而功能接口函数也返回错误的组合签名。

为此内核自测软件库提供了 `m0_scst_fault_inject_test_index` 全局变量，用户应用程序可以输入单个测试项的编号索引(如 2 等)，当 `m0_scst_execute_core_tests()` 函数执行完该编号索引的测试时，会注入一个错误签名，接口函数则停止剩余测试项测试，并返回一个错误组合签名，且与 `m0_scst_accumulated_signature` 全局变量的组合签名不一致。

刻意返回错误签名的影响仅一次，即执行后接口函数会将 `m0_scst_fault_inject_test_index` 全局变量的编号索引值清除。

6 数据类型

下表描述了内核自测软件库代码中使用的整数数据类型，这些数据类型在 `\SCST\src\h\m0_scst_typedefs.h` 文件中定义。

表 6-1. 整数数据类型

数据类型	描述
<code>m0_scst_int8_t</code>	signed 8-bit integer
<code>m0_scst_uint8_t</code>	unsigned 8-bit integer
<code>m0_scst_vint8_t</code>	volatile signed 8-bit integer
<code>m0_scst_vuint8_t</code>	volatile unsigned 8-bit integer

m0_scst_int16_t	signed 16-bit integer
m0_scst_uint16_t	unsigned 16-bit integer
m0_scst_vint16_t	volatile signed 16-bit integer
m0_scst_vuint16_t	volatile unsigned 16-bit integer
m0_scst_int32_t	signed 32-bit integer
m0_scst_uint32_t	unsigned 32-bit integer
m0_scst_vint32_t	volatile signed 32-bit integer
m0_scst_vuint32_t	volatile unsigned 32-bit integer

7 全局变量

本节介绍了 Cortex M0+内核自测软件库提供给用户应用程序的所有全局变量。

7.1 m0_scst_accumulated_signature 全局变量

变量定义：

```
m0_scst_uint32_t m0_scst_accumulated_signature;
```

变量说明：

这个全局变量是用户应用程序获取内核自测软件库测试结果(组合签名)的一种冗余路径,用户应用程序应与 m0_scst_execute_core_tests() 功能接口函数返回的组合签名进行比较,在硬件无故障的情况下,这个全局变量的值与 m0_scst_execute_core_tests() 功能接口函数返回的组合签名完全一致。

m0_scst_accumulated_signature 变量位于. m0_scst_ram_data_target0 自定义区域。

7.2 m0_scst_last_executed_test_number 全局变量

变量定义：

```
m0_scst_uint32_t m0_scst_last_executed_test_number;
```

变量说明：



这个全局变量表示当前在执行测试的测试项编号，当某个测试项失败时，`m0_scst_execute_core_tests()` 功能接口函数停止执行剩余测试项测试，因此该全局变量可用来确定测试失败的测试项编号。如所有测试项均执行成功，则该全局变量的值等于 `m0_scst_execute_core_tests()` 功能接口函数结束参数(形参)值。

`m0_scst_last_executed_test_number` 变量位于 `.m0_scst_test_shell_data` 自定义区域。

7.3 `m0_scst_fault_inject_test_index` 全局变量

变量定义：

`m0_scst_uint32_t m0_scst_fault_inject_test_index`。

变量说明：

这个全局变量由用户应用程序设置，输入测试项编号以指定该测试项执行结束后返回一个错误的签名值，具体用途请参见 5.4 章节。

`m0_scst_last_executed_test_number` 变量位于 `.m0_scst_test_shell_data` 自定义区域。

8 功能接口函数

本节介绍了 Cortex M0+内核测试软件库提供给用户应用程序的功能接口函数。

8.1 `m0_scst_execute_core_tests` 函数

函数调用：

```
m0_scst_uint32_t m0_scst_execute_core_tests (m0_scst_uint32_t start,  
m0_scst_uint32_t end) ;
```

形参说明：

形参名称	方向	描述
start	输入	要执行的第一个测试项编号索引
end	输入	要执行的最后一个测试项编号索引

函数说明：

这个函数执行所有测试项编号大于或等于 start 和小于或等于 end 的内核测试项。该函数还执行了各种检查，可能会产生下面表格列出的错误返回值。相应的常量在文件 m0_scst_test_shell.h 中定义。

m0_scst_execute_core_tests() 函数位于 .m0_scst_test_shell_code 自定义区域。

函数返回值：

返回值	描述
M0_SCST_WRONG_RANGE	输入形参错误，如 end<start，或 end>最大的测试项编号
错误的组合签名	当某个独立测试项测试失败后，停止执行剩余独立测试项测试，并返回一个错误的测试签名
正确的组合签名	所有独立测试项均成功执行后，返回一个正确的组合签名

应用示例：

```
m0_scst_uint32_t result;

/* Request execution of the tests with indices 2, 3, and 4 */
result = m0_scst_execute_core_tests (2, 4);
/* Here comes the code, which analyses:
   - Content of local “result” variable;
   - Content of global “m0_scst_last_executed_test_number” variable (should be 4 in
the given example in fault-free case). */
```

9 内核各测试项信息

对于内核的每个测试项，表 9-1 定义了每个测试项的编号索引及对应相关信息。测试项的编号索引将用作 m0_scst_execute_core_tests() 函数的 start 和 end 参数。每个测试项成功执行后将返回一个预期的签名。部分测试项在执行过程中会关闭用户应用程序



序中断，如此时用户中断触发，会造成用户 ISR 的调用延迟。每个测试项的执行时间及使用栈空间请参见表 9-1。

表 9-1 测试项信息

测试项名称	编号 索引	预期签名	用户 ISR 调用延 时 (us @48Mhz)	测试执 行时间 (us@48 Mhz)	栈使用 空 间 (bytes)
m0_scst_exception_test_svc	0	0x05D86B22	7, 44	11, 91	88
m0_scst_exception_test_pends v	1	0x71DDFB5B	6, 13	10, 64	56
m0_scst_exception_test_systi ck	2	0x1A6AE17F	6, 62	10, 54	56
m0_scst_exception_hardfault_ test1	3	0x428DE81A	4, 96	22, 40	56
m0_scst_exception_hardfault_ test2	4	0xE27BBB8D	5, 85	17, 37	68
m0_scst_exception_nmi_test	5	0xC2262509	9, 79	13, 85	88
m0_scst_exception_priority_t est	6	0x1971A0EA	7, 79	18, 20	56
m0_scst_loadstore_test1	7	0x6B8D4B1D	0	8, 18	32
m0_scst_loadstore_test2	8	0xF3668954	2, 58	10, 43	44
m0_scst_loadstore_test3	9	0xD56B91A4	0	10, 33	68
m0_scst_alu_test1	10	0x8F6CFDFC	2, 48	11, 10	32
m0_scst_alu_test2	11	0x48808A28	0	7, 41	20
m0_scst_alu_test3	12	0x8E4542E0	0	88, 97	20



m0_scst_alu_test4	13	0x5058B6F2	0	7, 8	20
m0_scst_alu_test5	14	0xBFFAE07E	0	28, 91	20
m0_scst_branch_test1	15	0x444E09B5	0	13, 68	24
m0_scst_branch_test2	16	0xDB1C1FEE	2, 48	11, 00	36
m0_scst_regfile_test1	17	0x4674E8F8	0	22, 91	52
m0_scst_regfile_test2	18	0x7EE47D44	2, 90	18, 95	40

注 1: 如果用户 ISR 的调用延时时间为 0, 这表明在执行该内核测试项时并不会禁用用户中断, 如果发生中断, 会直接根据用户的向量表进入用户服务程序。

表 9-2 测试项执行时的内核模式要求

测试项编号索引	特权模式/非特权模式	处理模式/线程模式	其他限制项
1	特权	线程	无
2	特权	线程	无
3	特权	线程	无
4	特权	线程	无
5	特权	线程	无
6	特权	线程	无
7	特权/非特权	处理/线程	无
8	特权	处理/线程	无
9	特权/非特权	处理/线程	无
10	特权	处理/线程	无
11	特权/非特权	处理/线程	无
12	特权/非特权	处理/线程	无
13	特权/非特权	处理/线程	无
14	特权/非特权	处理/线程	无
15	特权/非特权	处理/线程	无



16	特权	处理/线程	无
17	特权/非特权	处理/线程	无
18	特权	处理/线程	无

注：建议均在内核的特权模式下执行测试，所有的测试项都支持这种模式，在其他模式下调用可能会有限制，如表 9-2 所说明的。



版本信息

版本号	发布日期	更改说明
1.0.0	2023.11	首次发布



上海复旦微电子集团股份有限公司销售及服务中心

上海复旦微电子集团股份有限公司

地址：上海市国泰路 127 号 4 号楼

邮编：200433

电话：(86-021) 6565 5050

传真：(86-021) 6565 9115

上海复旦微电子（香港）股份有限公司

地址：香港九龙尖沙咀东嘉连威老道 98 号东海商业中心 5 楼 506 室

电话：(852) 2116 3288 2116 3338

传真：(852) 2116 0882

北京办事处

地址：北京市东城区东直门北小街青龙胡同 1 号歌华大厦 B 座 423 室

邮编：100007

电话：(86-10) 8418 6608

传真：(86-10) 8418 6211

深圳办事处

地址：深圳市华强北路 4002 号圣廷苑酒店世纪楼 1301 室

邮编：518028

电话：(86-0755) 8335 0911 8335 1011 8335 2011 8335 0611

传真：(86-0755) 8335 9011

台湾办事处

地址：台北市 114 内湖区内湖路一段 252 号 12 楼 1225 室

电话：(886-2) 7721 1889

传真：(886-2) 7722 3888



新加坡办事处

地址: 237, Alexandra Road, #07-01, The Alexcier, Singapore 159929

电话: (65) 6472 3688

传真: (65) 6472 3669

北美办事处

地址: 2490 W. Ray Road Suite#2 Chandler, AZ 85224 USA

电话: (480) 857-6500 ext 18

公司网址: <http://www.fmsh.com/>