MGC3130 GestIC® 库
接口说明
用户指南

© 2014 Microchip Technology Inc.

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下， Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是 “ 牢不可破 ” 的。

代码保护功能处于持续发展中。 Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了 《数字器件千年版权法案 （Digital Millennium Copyright Act）》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。**Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。** Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和 / 或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。在 Microchip 知识产权保护下，不得暗中或以其他方式转让任何许可证。

商标

Microchip 的名称和徽标组合、 Microchip 徽标、 dsPIC、 FlashFlex、 KEELOQ、 KEELOQ 徽标、 MPLAB、 PIC、 PICmicro、 PICSTART、 PIC32 徽标、 rfPIC、 SST、 SST 徽标、 SuperFlash 和 UNI/O 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

FilterLab、 Hampshire、 HI-TECH C、 Linear Active Thermistor、 MTP、 SEEVAL 和 The Embedded Control Solutions Company 均为 Microchip Technology Inc. 在美国的注册商标。

Silicon Storage Technology 为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

Analog-for-the-Digital Age、 Application Maestro、 BodyCom、 chipKIT、 chipKIT 徽标、 CodeGuard、 dsPICDEM、 dsPICDEM.net、 dsPICworks、 dsSPEAK、 ECAN、 ECONOMONITOR、 FanSense、 HI-TIDE、 In-Circuit Serial Programming、 ICSP、 Mindi、 MiWi、 MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、 mTouch、 Omniscient Code Generation、 PICC、 PICC-18、 PICDEM、 PICDEM.net、 PICkit、 PICtail、 REAL ICE、 rfLAB、 Select Mode、 SQI、 Serial Quad I/O、 Total Endurance、 TSHARC、 UniWinDriver、 WiperLock、 ZENA 和 Z-Scale 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 是 Microchip Technology Inc. 在美国的服务标记。

GestIC 和 ULPP 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. & KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2014, Microchip Technology Inc. 版权所有。

ISBN：978-1-63276-842-1

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

═ ISO/TS 16949 ═

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。 Microchip 的 PIC® MCU 与 dsPIC® DSC、KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品严格遵守公司的质量体系流程。此外， Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

© 2014 Microchip Technology Inc.

目录

前言 **5**

第 1 章 简介

1.1 本文档的用途 10
1.2 MGC3130 软件架构 10
1.3 GestIC® 库 11
1.4 桥接器 11
1.5 GestIC API 12
1.6 应用软件 12

第 2 章 MGC3130 主机接口

2.1 MGC3130 硬件接口 13
2.2 传输状态线 （TS）用法 14
2.3 编码示例 15

第 3 章 GestIC 库消息接口

3.1 消息概述 16
3.2 消息格式 17
3.3 消息头 17
3.4 消息有效内容 18
3.5 消息编码和解码 18
3.5.1 消息头提取 18
3.5.2 有效内容提取 19
3.6 消息控制流和编码示例 20
3.6.1 消息控制流 20
3.6.2 读取 GestIC 库版本 21
3.6.2.1 示例：请求 FW 版本信息 21
3.6.3 运行时控制 22
3.6.3.1 示例：使能接近检测 22
3.6.3.2 示例：使能所有手势 22
3.6.3.3 示例：使能数据输出 23
3.6.3.4 示例：锁定数据输出 23
3.6.4 传感器数据输出 24
3.6.4.1 示例：读取传感器数据输出 24

第 4 章 GestIC 库消息参考

4.1 System_Status 25
4.2 Request_Message 27
4.3 FW_Version_Info 28
4.4 Set_Runtime_Parameter 29
4.4.1 触发 29
4.4.2 持久 30
4.4.3 模拟前端 （AFE）类型 30

4.4.3.1 信号匹配 30
4.4.3.2 电极映射 31
4.4.4 数字信号处理 （DSP）类 32
4.4.4.1 发送频率选择 32
4.4.4.2 触摸检测 33
4.4.4.3 接近检测 33
4.4.5 系统类 34
4.4.5.1 AirWheel 34
4.4.5.2 手势处理 （HMM） 34
4.4.5.3 校准工作模式 34
4.4.5.4 数据输出使能屏蔽 35
4.4.5.5 数据输出锁定屏蔽 36
4.4.5.6 数据输出请求屏蔽 37
4.4.5.7 手势进行中标志控制 37
4.5 Sensor_Data_Output 38
第 5 章 GestIC 库更新消息
5.1 库加载程序更新过程 41
5.2 FW_Update_Start 42
5.3 FW_Update_Block 43
5.4 FW_Update_Completed 45
附录 A I²C™ 命令示例 46
附录 B 术语表 53
全球销售及服务网点 54

© 2014 Microchip Technology Inc.

前言

客户须知

所有文档均会过时，本文档也不例外。 **Microchip** 的工具和文档将不断演变以满足客户的需求，因此实际使用中有些对话框和 **/** 或工具说明可能与本文档所述之内容有所不同。请访问我们的网站（**www.microchip.com**）获取最新文档。

文档均标记有 **"DS"** 编号。该编号出现在每页底部的页码之前。 **DS** 编号的命名约定为 **"DSXXXXXXXXA_CN"**，其中 **"XXXXXXXX"** 为文档编号， **"A"** 为文档版本。

欲了解开发工具的最新信息，请参考 **MPLAB® IDE** 在线帮助。从 **Help** （帮助）菜单选择 **Topics** （主题），打开现有在线帮助文件列表。

简介

本章包含使用 MGC3130 GestIC® 库接口前需要了解的一般信息。内容包括：

- 文档编排
- 本指南使用的约定
- 保修登记
- 推荐读物
- Microchip 网站
- 开发系统变更通知客户服务
- 客户支持
- 文档版本历史

文档编排

本文档介绍了 MGC3130 GestIC 库且内容编排如下：

- **第 1 章 简介**
- **第 2 章 MGC3130 主机接口**
- **第 3 章 GestIC 库消息接口**
- **第 4 章 GestIC 库消息参考**
- **第 5 章 GestIC 库更新消息**

本指南使用的约定

本指南采用以下文档约定：

文档约定

说明	表示	示例
Arial 字体：		
斜体字	参考书目	*MPLAB® IDE User's Guide*
	需强调的文字	…… 为*仅有*的编译器 ……
首字母大写	窗口	Output 窗口
	对话框	Settings 对话框
	菜单选择	选择 Enable Programmer
引用	窗口或对话框中的字段名	“Save project before build”
带右尖括号且有下划线的斜体文字	菜单路径	*File>Save*
粗体字	对话框按钮	单击 **OK**
	选项卡	单击 **Power** 选项卡
N‘Rnnnn	verilog 格式的数字，其中 N 为总位数， R 为基数， n 为其中一位。	4‘b0010， 2‘hF1
尖括号 < > 括起的文字	键盘上的按键	按下 <Enter>，<F1>
Courier New 字体：		
常规 Courier New	源代码示例	`#define START`
	文件名	`autoexec.bat`
	文件路径	`c:\mcc18\h`
	关键字	`_asm`， `_endasm`， `static`
	命令行选项	`-Opa+`，`-Opa-`
	二进制位值	`0`， `1`
	常量	`0xFF`， `‘A’`
斜体 Courier New	可变参数	*`file`*`.o`，其中 *`file`* 可以是任一有效文件名
方括号 [ ]	可选参数	`mcc18 [选项] file [选项]`
花括号和竖线：{ | }	选择互斥参数；“或”选择	`errorlevel {0|1}`
省略号 ...	代替重复文字	`var_name [, var_name...]`
	表示由用户提供的代码	`void main (void) { ... }`

© 2014 Microchip Technology Inc.

保修登记

请填写随附的保修登记卡 （Warranty Registration Card）并尽快寄出。寄出保修登记卡的客户将可收到新产品更新信息。可在 Microchip 网站上获得临时软件版本。

推荐读物

本用户指南介绍了如何使用 MGC3130 GestIC 库接口。下面列出了其他有用的文档。以下 Microchip 文档均已提供，并建议读者作为补充参考材料。

- 《MGC3130 单区 3D 手势控制器数据手册》（DS40001667C_CN） —— 有关 MGC3130 3D 跟踪和手势控制器的信息，请参见此文档。
- 《MGC3130 Aurea 图形用户界面用户指南》（DS40001681_CN） —— 介绍如何使用 MGC3130 Aurea 图形用户界面。
- 《MGC3130 GestIC® 设计指南》（DS40001716A_CN） —— 本文档介绍了 MGC3130 系统特性参数及设计过程。用户借助本文档可实现良好的电极设计并确定整个 GestIC 系统的参数。

MICROCHIP 网站

Microchip 网站 （www.microchip.com）为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的互联网浏览器即可访问，网站提供以下信息：

- **产品支持** —— 数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持** —— 常见问题解答 （FAQ）、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务** —— 产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、 Microchip 销售办事处、代理商以及工厂代表列表

可通过 http://www.microchip.com/gestic 直接获取有关 GestIC 技术和 MGC3130 的信息。

开发系统变更通知客户服务

Microchip 的客户通知服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

要注册，请先访问 Microchip 网站 www.microchip.com，点击 “ 变更通知客户 ” （Customer Change Notification），然后按照注册指示完成注册。

开发系统产品组的分类如下：

- **编译器** ——Microchip C 编译器、汇编器、链接器及其他语言工具的最新信息，包括所有 MPLAB® C 编译器、所有 MPLAB 汇编器 （包括 MPASM™ 汇编器）、所有 MPLAB 链接器 （包括 MPLINK™ 目标链接器）以及所有 MPLAB 库管理器 （包括 MPLIB™ 目标库管理器）。
- **仿真器** ——Microchip 在线仿真器的最新信息。其中包括 MPLAB® REAL ICE™ 和 MPLAB ICE 2000 在线仿真器。
- **在线调试器** ——Microchip 在线调试器的最新信息。其中包括 MPLAB ICD 3 在线调试器和 PICkit™ 3 Debug Express。
- **MPLAB IDE**—— 关于开发系统工具的 Windows® 集成开发环境 Microchip MPLAB IDE 的最新信息，主要针对 MPLAB IDE、 MPLAB IDE 项目管理器、MPLAB 编辑器、 MPLAB SIM 软件模拟器以及一般编辑和调试功能。
- **编程器** ——Microchip 编程器的最新信息，其中包括生产用编程器 （如 MPLAB REAL ICE 在线仿真器）、 MPLAB ICD 3 在线调试器以及 MPLAB PM3 器件编程器。还包括非生产用开发编程器，例如 PICSTART® Plus 以及 PICkit 2 和 3。

© 2014 Microchip Technology Inc.

客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师 （FAE）
- 技术支持

客户应联系其代理商、代表或应用工程师 （FAE）寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 http://www.microchip.com/support 获得网上技术支持。

文档版本历史

版本 A （2013 年 8 月）

- 本文档的初始版本。

版本 B （2013 年 11 月）

- 更新了第 1、 2、 3 和 4 章；新增第 5 章；更新了 GestIC 库 V1.0 及更高版本的内容。

版本 C （2014 年 5 月）

- 更新了第 3.5.2 节 （有效内容提取）、第 4.2 节 （`Request_Message`）、第 4.4 节 （`Set_Runtime_Parameter`）和第 4.5 节 （`Sensor_Data_Output`）；更新了表 3-7、 3-12、 3-13、 3-14、 5-2、 5-4 和 5-6 ；添加了附录 A （I^2C™ 命令示例）。

第 1 章 简介

1.1 本文档的用途

本文档是 MGC3130 的 GestIC 库的接口说明；概述了该库的 I²C™ 消息接口的功能，并包含了控制和操作 MGC3130 系统的完整消息。

涵盖的主要部分有：

- 消息接口和数据协议的说明
- GestIC 库的消息参考

本文档不涉及 Colibri Suite 的参数化。这只能通过 Aurea PC 软件实现。请参见《MGC3130 Aurea 图形用户界面用户指南》（DS40001681_CN）。

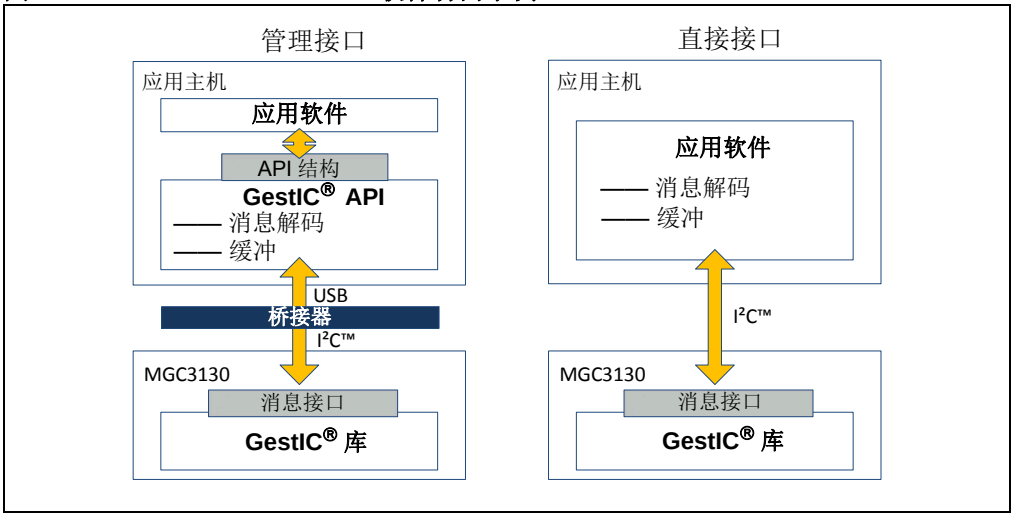
1.2 MGC3130 软件架构

MGC3130 系统可从两个软件层访问：

- 通过 GestIC 库消息接口的 I²C 直接访问 （直接接口）
- 通过作为消息抽象层的 GestIC API （管理接口）

图 1-1 给出了这两种主要选项的示例。

图 1-1： MGC3130 软件访问示例



直接接口是访问 MGC3130 的最简单方式，但它需要用户接收和解码所有 I²C 消息，并验证接收到的数据。如果应用程序采用缩减的传感器数据集 （如仅手势和仅位置），则推荐直接访问。通过 GestIC API 的管理接口提供已解码和已验证的传感器数据，这些数据可由应用程序直接采用。通常， GestIC API 运行在 PC 应用程序或操作系统驱动程序中，这些程序将数据提供给应用软件。

以下章节将简要描述这两种接口模式的构建模块。

© 2014 Microchip Technology Inc.

1.3 GestIC® 库

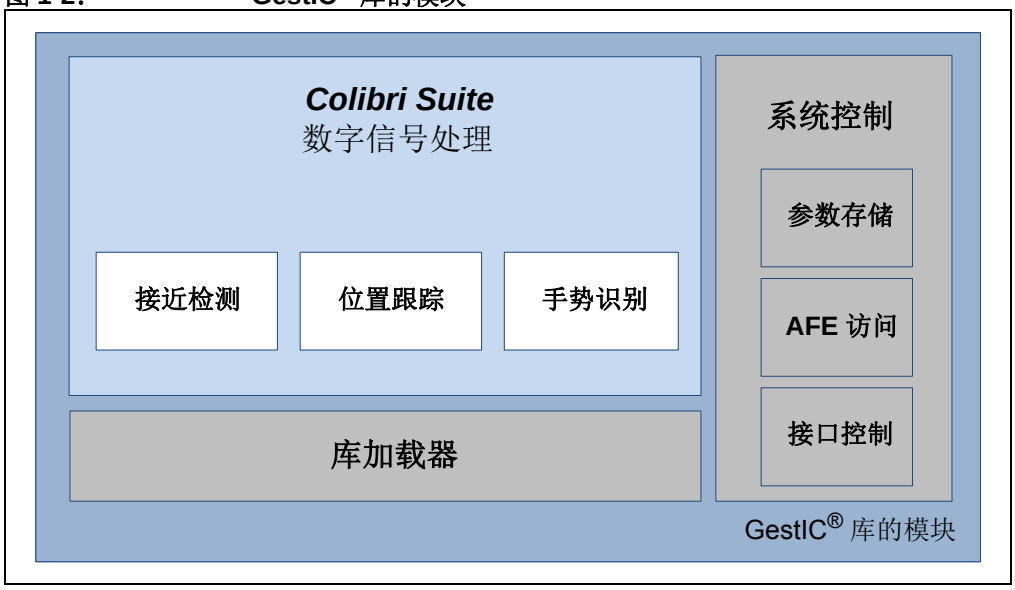
GestIC 库是存储在 MGC3130 内部闪存中的嵌入式固件。它包含：

- Colibri Suite，涵盖实现 GestIC 功能的数字信号处理算法 （GestIC 的核心功能有：接近检测、位置跟踪和手势识别）
- 系统控制模块，提供对主机接口、参数存储和 AFE 访问的完全控制
- 用于 GestIC 库更新的库加载程序

主要模块如图 1-2 所示。

GestIC 库包含一个基于消息的接口，可用于配置芯片并将传感器数据传输给主机应用程序。

图 1-2： GestIC® 库的模块



1.4 桥接器

如果应用主机不支持本地 I^2C 接口，则需要附加硬件桥接器。桥接器将 I^2C 硬件协议转换成 USB/UART。

如果引入桥接器硬件，则应用主机需要一个附加设备驱动程序，以注册接口并提供操作系统内的 MGC3130 数据。

例如：

- Windows CDC 驱动程序将 MGC3130 数据发送给一个虚拟 COM 端口。在这种情况下，驱动程序不了解 MGC3130 数据格式。
- HID 驱动程序直接将 MGC3130 数据作为操作系统中的 USB HID 类使用。此类驱动程序必须解码 MGC3130 消息，因此，建议将 GestIC API 参考代码作为其一部分。

1.5 GestIC API

Microchip 开发了 GestIC API 作为 MGC3130 消息的抽象层，以提供一个简化的用户接口，它可方便地集成到客户的应用程序中。

GestIC API 随附一个 C 参考代码，其包括消息缓冲器、解码器和事件处理程序，使接口独立于低层协议且不受其时序限制。

1.6 应用软件

传感器输出在用户的应用程序中使用，用户应用程序根据用户手的移动集成了不同的操作。

通常，应用软件提供一个图形用户界面 （Graphical User Interface， GUI），将 MGC3130 控制选项可视化，如 MGC3130 评估和开发工具包中提供的 Aurea。

© 2014 Microchip Technology Inc.

第 2 章 MGC3130 主机接口

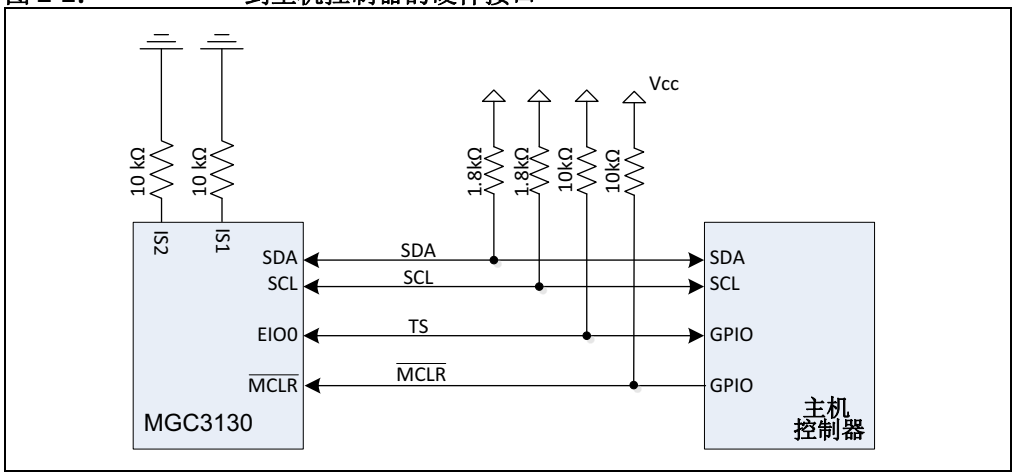
2.1 MGC3130 硬件接口

通过一个双线 I²C 兼容串口来实现与 MGC3130 的通信，该串口支持用户读取传感器数据并将控制消息发送给芯片。它通过串行接口与主控制器通信，工作速度高达 400 kHz。一个引脚 （IS2）用于地址选择，使用户能够在同一条总线上连接最多两个 MGC3130 器件，而不会发生地址冲突。

> **注：** MGC3130 I²C 地址是 0x42 和 0x43。它们是无 R/$\overline{W}$ 位的器件地址。请参见《MGC3130 单区 3D 手势控制器数据手册》（DS40001667C_CN）。

另外， MGC3130 需要一条专用的能提供数据传输状态的传输状态线 （TS）。它由 I²C 主从器件使用来控制数据流。 I²C SCL、I²C SDA 和 TS 线要求在 MGC3130 和连接的主机控制器上采用漏极开路连接。为了正常工作，需要在 I²C SCL 和 Vcc 以及 I²C SDA 和 Vcc 之间连接 1.8 kΩ 的上拉电阻，在 TS 线和 Vcc 之间连接 10 kΩ 的上拉电阻。

图 2-1： 到主机控制器的硬件接口



为了实现 MGC3130 的控制选项，建议主机控制器控制 MGC3130 $\overline{MCLR}$ 线。尤其是硬件复位对于 GestIC 库更新过程很必要。

2.2 传输状态线 （TS）用法

传输状态线用于检查 I²C 数据是否有效以及能否将其从 MGC3130 发送到主机控制器。

MGC3130 （I²C 从器件）使用该线通知主机控制器 （I²C 主器件）有可传输的数据。主机控制器使用 TS 线指示数据正在传输，并阻止 MGC3130 更新其数据缓冲区。

表 2-1 显示了在不同通信状态下如何使用 TS 线。

只有在双方均释放 TS 线后， MGC3130 才能更新 I²C 缓冲区；只有当 MGC3130 将 TS 拉为低电平时，才能启动数据传输。

该步骤可确保：

- 在有新传感器数据可用时始终通知主机
- MGC3130 缓冲区始终在数据发送至 I²C 总线前更新完毕

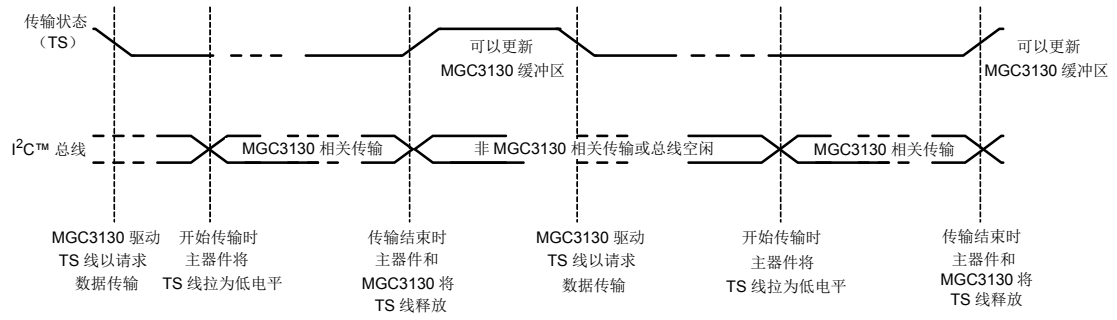
图 2-2 显示了完整的通信协议。

表 2-1： 传输状态线用法

MGC3130	主机控制器	TS 线	状态
释放 （H）	释放 （H）	高	主机结束数据读取 （传输结束）。不再向主机传输更多数据。允许 MGC3130 更新数据缓冲区。
有效 （L）	释放 （H）	低	来自 MGC3130 的数据可供发送，但主机尚未开始读取。如果在下一次数据更新 （5 ms）开始前，主机忙碌并且未开始读取， MGC3130 会在更新数据缓冲区之前使 TS 线为高电平。
有效 （L）	有效 （L）	低	主机开始读取。传输结束 （主机释放 TS 线，使之为高电平）前，不会更新 MGC3130 数据缓冲区。
释放 （H）	有效 （L）	低	MGC3130 准备更新数据缓冲区，但是主机仍在读取之前的数据。只允许 MGC3130 在主机释放 TS 线，使之为高电平时，更新数据。

© 2014 Microchip Technology Inc.

图 2-2： MGC3130 通信协议



> 注 1： 在数据传输完成以后，主机控制器 （I²C 主器件）在 I²C 数据传输完成后产生停止条件。因此，建议校验报文头指定的要读取的字节数 （长度字段）。
>
> 2： 只有在将数据从 MGC3130 传输到主机控制器时，才需要传输状态。写入 MGC3130 无需额外的 TS 信号。

2.3 编码示例

除了标准 I²C 接口，MGC3130 和主机控制器之间的通信需要适当处理传输状态。为便于集成，将要求放在下面的代码示例中。

例 2-1： 主机控制器中的代码实现

```
I²C™ Read Function - requires TS:
I²C™ Master read loop:
    Read TS
    If TS == 0:
       Assert TS
       Send I²C™ start condition
       Send I²C™ device address + read indication
       Receive I²C™ payload (the GestIC® Library message)
       Send I²C™ stop condition
       Release TS
    Wait 200 µs (to assure that MGC3130 released TS line, too)
I²C™ Write Function - does not require TS:
I²C™ Master write loop:
       Send I²C™ start condition
       Send I²C™ device address + write indication
       Send I²C™ payload (the GestIC® Library message)
       Send I²C™ stop condition
```

第 3 章 GestIC 库消息接口

3.1 消息概述

定义 GestIC 库消息，以向主机应用程序提供传感器数据并控制 MGC3130 及其嵌入特性。这些消息作为 I^2C 数据包的有效内容发送。

表 3-1： 系统控制消息

ID	名称	所在页
0x15	System_Status	25
0x06	Request_Message	27
0x83	FW_Version_Info	28
0xA2	Set_Runtime_Parameter	29

表 3-2： 传感器数据输出消息

ID	名称	所在页
0x91	Sensor_Data_Output	38

表 3-3： GestIC® 库更新消息

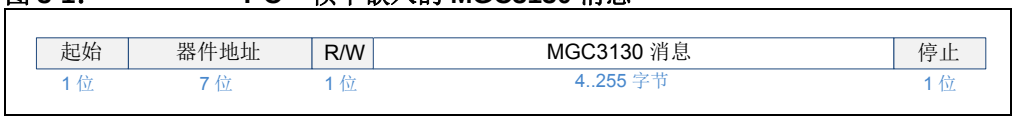
ID	名称	所在页
0x80	FW_Update_Start	42
0x81	FW_Update_Block	43
0x82	FW_Update_Completed	45

© 2014 Microchip Technology Inc.

3.2 消息格式

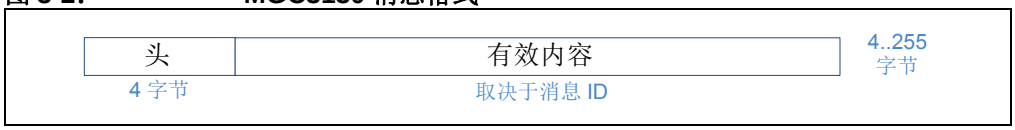
消息是 GestIC 库和应用主机之间交换数据的容器。每条消息的最小长度为 4 字节，最大长度为 255 字节，且适合通信接口 （如 I²C）的数据包。每帧传输一条消息 （见图 3-1）。

图 3-1： I²C™ 帧中嵌入的 MGC3130 消息



消息始终由一个 4 字节头和可变有效内容组成。格式如图 3-2 所示。

图 3-2： MGC3130 消息格式



3.3 消息头

GestIC 库消息头是固定的，长度为 4 字节。它包含四个数据字段，分别如图 3-3 所示和表 3-4 所述。

图 3-3： MGC3130 消息头

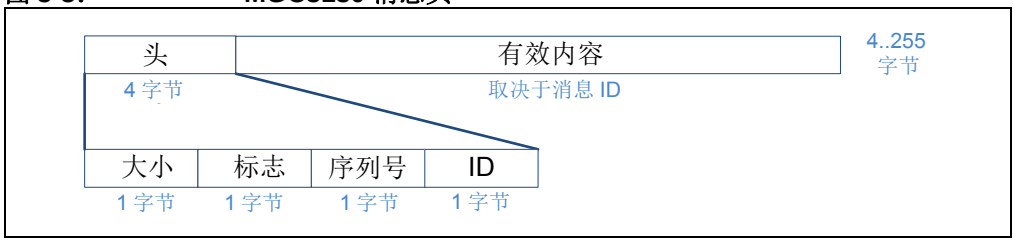


表 3-4： MGC3130 消息头的数据字段

字段	大小 （字节）	说明
消息大小	1	消息的完整大小 （字节）（包括头）。
标志	1	保留，供将来使用。
序列号	1	MGC3130 每发送一条消息序列号就会增 1。范围为 0…255。主机控制器可使用该信息来验证在 I²C™ 传输过程中是否丢失消息。 MGC3130 忽略接收到的消息的序列号。
ID	1	消息的 ID。对于每个 ID， GestIC® 库保持专用结构，包括消息方向、其有效内容元素及可能的回复操作。

3.4 消息有效内容

消息有效内容的长度可变，由一个或多个有效内容元素组成，这些元素包含要交换的信息。根据内容，这些元素可以是数值或专用数字。

图 3-4： MGC3130 消息有效内容

> **注：** 有效内容元素按照从大到小的字节顺序进行交换。这意味着首先写入最低有效字节。
>
> 示例：4 字节元素：[Byte0]:[Byte1]:[Byte2]:[Byte3]

第 4 章 “GestIC 库消息参考 ” 介绍了有效内容元素的结构和内容。

3.5 消息编码和解码

GestIC 库消息可读为一行十六进制值。要进行解码，需要提取消息头和有效内容元素，并映射到消息参考中的定义 （见**第 4 章 “GestIC 库消息参考 ”**）。

示例消息：下面给出了 ID 为 `0x83, FW_Version_Info` 的解码。

例 3-1： 消息 `0x83` 的十六进制表示

```
84 00 00 83 AA 63 80 E6 0C 64 15 20 31 2E 30 2E 30 3B 70 3A 48 69 6C 6C 73 74 61 72
56 30 31 3B 44 53 50 3A 49 44 39 30 30 30 72 31 38 34 39 3B 69 3A 42 3B 66 3A 32 32
35 30 30 3B 6E 4D 73 67 3B 73 3A 42 65 74 61 32 72 31 30 34 30 3A 31 30 34 39 3A 4D
4F 3B 63 3A 4D 4B 49 3B 74 3A 32 30 31 33 2F 31 31 2F 30 38 20 31 33 3A 30 33 3A 30
00 10 00 00 55 AA 90 65 20 20 80 0F FF 00 FF 00 E1 EA 00 00
```

3.5.1 消息头提取

例 3-2： 消息头

```
84 00 00 83 AA 63 80 E6 0C 64 15 20 31 2E 30 2E 30 3B 70 3A 48 69 6C 6C 73 74 61 72
56 30 31 3B 44 53 50 3A 49 44 39 30 30 30 72 31 38 34 39 3B 69 3A 42 3B 66 3A 32 32
35 30 30 3B 6E 4D 73 67 3B 73 3A 42 65 74 61 32 72 31 30 34 30 3A 31 30 34 39 3A 4D
4F 3B 63 3A 4D 4B 49 3B 74 3A 32 30 31 33 2F 31 31 2F 30 38 20 31 33 3A 30 33 3A 30
00 10 00 00 55 AA 90 65 20 20 80 0F FF 00 FF 00 E1 EA 00 00
```

消息头包含以下信息：

- *大小： `0x84` 包含头的消息长 132 字节*
- *标志： `0x00` 标志未设置*
- *序列号： `0x00` 消息已发出，序列号为 0*
- *ID： `0x83` 消息 ID 是 `0x83, FW_Version_Info`*

© 2014 Microchip Technology Inc.

3.5.2 有效内容提取

例 3-3： 消息有效内容

```
84 00 00 83 AA 63 80 E6 0C 64 15 20 31 2E 30 2E 30 3B 70 3A 48 69 6C 6C 73 74 61 72 56
30 31 3B 44 53 50 3A 49 44 39 30 30 30 72 31 38 34 39 3B 69 3A 42 3B 66 3A 32 32 35 30
30 3B 6E 4D 73 67 3B 73 3A 42 65 74 61 32 72 31 30 34 30 3A 31 30 34 39 3A 4D 4F 3B 63
3A 4D 4B 49 3B 74 3A 32 30 31 33 2F 31 31 2F 30 38 20 31 33 3A 30 33 3A 30 00 10 00 00
55 AA 90 65 20 20 80 0F FF 00 FF 00 E1 EA 00 00
```

根据**第 4.3 节 “FW_Version_Info”**， `FW_Version_Info` 包含七个有效内容元素：

- **`FWValid`** GestIC 库的状态 （1 个字节）
- **`HwRev`** HW 版本信息 （2 个字节）
- **`ParameterStartAddr`** 参数的起始地址 （1 个字节）
- **`LibraryLoaderVersion`** GestIC 库加载程序版本 （3 个字节）
- **`FWStartAddr`** GestIC 库的起始地址 （1 个字节）
- **`FWVersion`** 有效 GestIC 库的版本信息 （120 个字节）

现在，可将数值转换并映射为有效内容元素的说明：

`FWValid` = `AA` （`170`）：存在有效的 GestIC 库

`HwRev` = `63 80` （读为 `0x80 0x63` ）：HW 版本为 `128.99`

`ParameterStartAddr` = `0xE6` （`230x128=29440`）：参数的起始地址是 29440

`LibraryLoaderVersion` = `0C 64` （读为 `0x64 0x0C`）：库加载程序版本为 `100.12`

`LibraryLoaderPlatform` = `15` （读为 `0x15`）：库加载程序平台为 `21`

`FWStartAddr` = `0x20` （`32x128=4096`）：GestIC 库的起始地址是 4096

`FWVersion` = `31 2E 30 2E 30 3B 70 3A 48 69 6C 6C 73 74 61 72 56 30 31 3B 44 53 50 3A 49 44 39 30 30 30 72 31 38 34 39 3B 69 3A 42 3B 66 3A 32 32 35 30 30 3B 6E 4D 73 67 3B 73 3A 42 65 74 61 32 72 31 30 34 30 3A 31 30 34 39 3A 4D 4F 3B 63 3A 4D 4B 49 3B 74 3A 32 30 31 33 2F 31 31 2F 30 38 20 31 33 3A 30 33 3A 30 00 10 00 00 55 AA 90 65 20 20 80 0F FF 00 FF 00 E1 EA 00 00`

版本字符串根据 ASCII 码字符解析。它是一个以分号分隔的字符串，始终以版本号开始，后跟不同标记：

`1.0.0;p:HillstarV01;DSP:ID9000r1849;i:B;f:22500;nMsg;s:Beta2r1040:1049:MO;c:MKI;t:2013/11/08 13:03:0;...`

3.6 消息控制流和编码示例

3.6.1 消息控制流

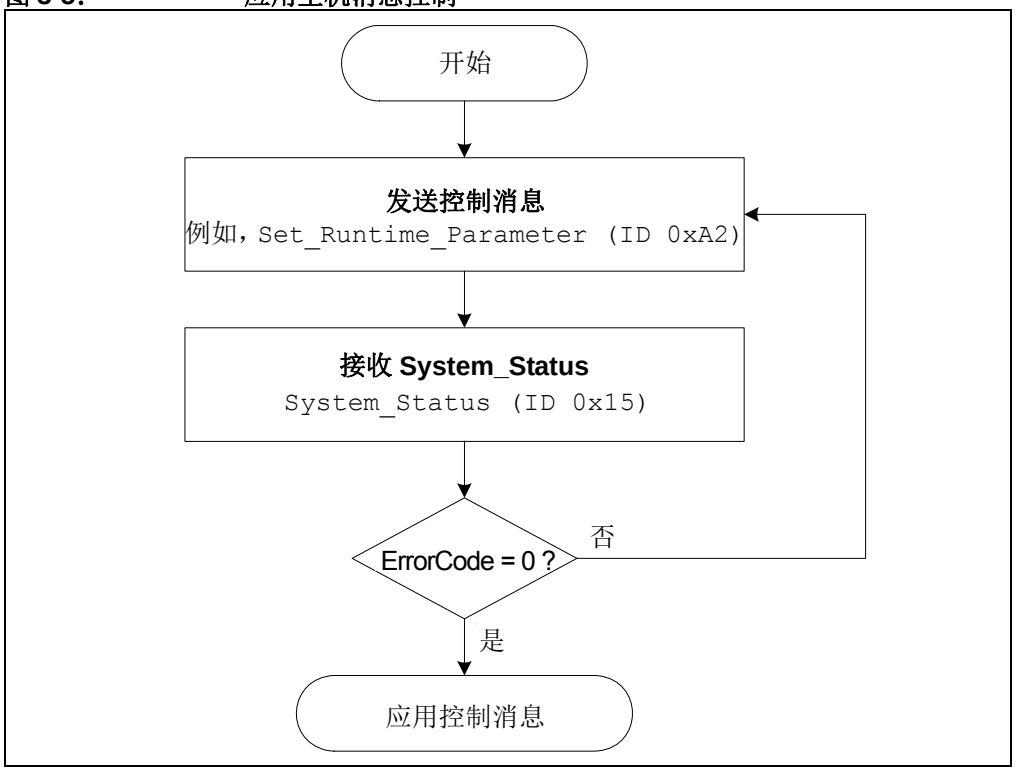
通过下列消息实现对 MGC3130 GestIC 库的控制：

- `Set_Runtime_Parameter (ID 0xA2)`
- `Request_Message (ID 0x06)`

MGC3130 通过包含原始消息 ID 和 2 字节错误代码的 `System_Status (ID 0x15)` 确认每条控制消息。如果错误代码是 `0`，则将消息正确应用于 MGC3130。

从应用主机的角度看，消息控制流如图 3-5 所示。

图 3-5： 应用主机消息控制



> **注：** 在通过 USB 的 UART 仿真发送数据包前， Hillstar 和 Sabrewing I^2C 转 USB 桥接器为每个 I^2C 数据包添加前缀 `0xFEFF`。这是为了在 PC 数据流内部进行帧分隔。对于从终端程序 （如 Hterm）发送到 MGC3130 的消息，也必须添加前缀。

© 2014 Microchip Technology Inc.

3.6.2 读取 GestIC 库版本

在上电或复位后， MGC3130 运行库加载程序，并发送消息 FW_Version_Info (0x83)。应用主机会收到该消息并作为首次通信检查。在 200 ms 超时后， GestIC 库处理模式自动启动。

应用主机可在运行时使用 Request_Message (0x06) 请求 FW_Version_Info。

3.6.2.1 示例：请求 FW 版本信息

下面示例显示了如何使用 Request_Message (0x06) 请求 FW_Version_Info (0x83) 消息。

表 3-5： 从主机到 MGC3130 的消息：REQUEST_MESSAGE (0X06)

原始消息	0C 00 00 06 83 00 00 00 00 00 00 00		
有效内容元素	MessageID	Reserved	Parameter
小尾数 16 进制数	83	00 00 00	00 00 00 00
解码的 16 进制数	0x83	不适用	不适用
说明	FW_Version_Info	不适用	不适用

MGC3130 以消息 FW_Version_Info (0x83) 后跟 System_Status (0x15)（包含错误代码）回复。

表 3-6： 从 MGC3130 到主机的消息：FW_VERSION_INFO (0X83)

原始消息	84 00 01 83 AA 00 00 FF 00 00 00 20 31 2E 30 2E 30 3B 70 3A 48 69 6C 6C 73 74 61 72 56 30 31 3B 44 53 50 3A 49 44 39 30 30 30 72 31 38 34 39 3B 69 3A 42 3B 66 3A 32 32 35 30 30 3B 6E 4D 73 67 3B 73 3A 42 65 74 61 32 72 31 30 34 30 3A 31 30 34 39 3A 4D 4F 3B 63 3A 4D 4B 49 3B 74 3A 32 30 31 33 2F 31 31 2F 30 38 20 31 33 3A 30 33 3A 30 38 3B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 E1 EA 00 00					
有效内容元素	FWValid	HWRev	Parameter-StartAddr	LibraryLoad-erVersion	FWStartAddr	FWVersion
小尾数 16 进制数	AA	00 00	FF	00 00 00	20	...
解码的 16 进制数	0xAA	不适用	不适用	不适用	0x20	...
说明	170 ValidFW	仅在 MGC3130 启动后有效	仅在 MGC3130 启动后有效	仅在 MGC3130 启动后有效	GestIC® 库的起始地址	请参见下面的内容

FWVersion 根据 ASCII 码字符解析为：

1.0.0;p:HillstarV01;DSP:ID9000r1849;i:B;f:22500;nMsg;s:Beta2r1040:1049:MO;c:MKI;t:2013/11/08 13:03:08;…

- GestIC 库版本： 1.0.0
- 平台： HillstarV01
- Colibri Suite 版本： ID9000r1849
- 创建时间： 2013/11/08 13:03:08

3.6.3 运行时控制

消息 Set_Runtime_Parameter (0xA2) 提供了一组专用的运行时控制选项。它可用于控制有效特性集和传感器数据输出，因此，它支持组合 MGC3130 的上下文相关操作。欲了解详细的消息说明，请参见**第 4.4 节 "Set_Runtime_Parameter"**。

下面示例显示了如何设置相关运行时参数。

3.6.3.1 示例：使能接近检测

该示例显示了如何使用消息 Set_Runtime_Parameter (0xA2) 使能接近检测模式。

表 3-7： 从主机到 MGC3130 的消息：SET_RUNTIME_PARAMETER (0XA2)

原始消息	10 00 00 A2 97 00 00 00 01 00 00 00 01 00 00 00			
有效内容元素	RuntimeParameterID	Reserved	Argument0	Argument1
小尾数 16 进制数	97 00	00 00	01 00 00 00	01 00 00 00
解码的 16 进制数	0x0097	不适用	0x00000001	x00000001
说明	ApproachDetection	不适用	使能接近检测模式	接近检测屏蔽位

MGC3130 以消息 System_Status (0x15)（包含错误代码）回复。

表 3-8： 从 MGC3130 到主机的消息：SYSTEM_STATUS (0X15)

原始消息	10 00 08 15 A2 34 00 00 00 00 00 00 00 00 00 00				
有效内容元素	MsgID	MaxCmdSize	ErrorCode	Reserved	Reserved
小尾数 16 进制数	A2	34	00 00	00 00 00 00	00 00 00 00
解码的 16 进制数	0xA2	0x34	0x0000	不适用	不适用
说明	确认 ID 为 0xA2	不适用	无错误	不适用	不适用

3.6.3.2 示例：使能所有手势

该示例显示了如何使用消息 Set_Runtime_Parameter (0xA2) 使能所有手势（弹指和画圈）。

表 3-9： 从主机到 MGC3130 的消息：SET_RUNTIME_PARAMETER (0XA2)

原始消息	10 00 00 A2 85 00 00 00 7F 00 00 00 7F 00 00 00			
有效内容元素	RuntimeParameterID	Reserved	Argument0	Argument1
小尾数 16 进制数	85 00	00 00	7F 00 00 00	7F 00 00 00
解码的 16 进制数	0x0085	不适用	0x0000007F	0x0000007F
说明	despGestureMask	不适用	使能手势 0...6	使能手势 0 至 6 的屏蔽位

MGC3130 以消息 System_Status (0x15) 回复。请参见表 3-8。

© 2014 Microchip Technology Inc.

3.6.3.3 示例：使能数据输出

该示例显示了如何使能手势数据、触摸数据、 AirWheel 数据和位置数据的传感器数据输出。请参见**第 4.4.5.4 节 " 数据输出使能屏蔽 "**。

表 3-10： 从主机到 MGC3130 的消息：SET_RUNTIME_PARAMETER (0XA2)

原始消息	10 00 00 A2 A0 00 00 00 1E 00 00 00 FF FF FF FF			
有效内容元素	RuntimeParameterID	Reserved	Argument0	Argument1
小尾数 16 进制数	A0 00	00 00	1E 00 00 00	FF FF FF FF
解码的 16 进制数	0xA0	0x0000	0x0000001E	0xFFFFFFFF
说明	DataOutputEn-ableMask	不适用	使能 bit 1...bit 4 ；禁止所有其他位	改写当前配置

MGC3130 以消息 System_Status (0x15) 回复。请参见表 3-8。

3.6.3.4 示例：锁定数据输出

该示例显示了如何锁定手势数据、触摸数据、 AirWheel 数据和位置数据的传感器数据输出。请参见**第 4.4.5.5 节 " 数据输出锁定屏蔽 "**。

表 3-11： 从主机到 MGC3130 的消息：SET_RUNTIME_PARAMETER (0XA2)

原始消息	10 00 00 A2 A1 00 00 00 1E 00 00 00 FF FF FF FF			
有效内容元素	RuntimeParameterID	Reserved	Argument0	Argument1
小尾数 16 进制数	A1 00	00 00	1E 00 00 00	FF FF FF FF
解码的 16 进制数	0x00A1	0x0000	0x0000001E	0xFFFFFFFF
说明	DataOutputLockMask	不适用	使能 bit 1...bit 4 ；禁止所有其他位	改写当前配置

MGC3130 以消息 System_Status (0x15) 回复。请参见表 3-8。

3.6.4 传感器数据输出

GestIC 库以 5 ms 的默认更新速率处理传感器数据。这意味着 I²C 消息缓冲区以该时间间隔定期更新。当有新数据时， MGC3130 则驱动 TS 线以请求 I²C 主器件传输该数据。从 MGC3130 向主机发送的传感器数据包含在消息 Sensor_Data_Output (0x91) 中。

传感器数据输出的内容可通过消息 Set_Runtime_Parameter (0xA2) 配置。

3.6.4.1 示例：读取传感器数据输出

在下面的示例中，传感器数据输出根据**第 3.6.3.3 节 " 示例：使能数据输出 "** 和**第 3.6.3.4 节 " 示例：锁定数据输出 "** 配置。

表 3-12： **从 MGC3130 到主机的消息：从东到西滑动**

原始消息	18 08 FF 91 1E 01 57 8C 03 10 04 00 00 00 00 00 00 00 00 00 00 00 00 00				
有效内容元素	SystemInfo	GestureInfo	TouchInfo	Air-WheelInfo	xyzPosition
小尾数 16 进制数	8C	03 10 04 00	00 00 00 00	00 00	00 00 00 00 00 00
解码的 16 进制数	0x8C	0x00041003	0x00000000	0x0000	0x000000000000
说明	Bit 2：RawDataValid Bit 3：NoisePowerValid Bit 7：DSPRunning	从东到西划动	无触摸	无 AirWheel	无可用位置数据

表 3-13： **从 MGC3130 到主机的消息：触摸中央电极**

原始消息	18 08 3B 91 1E 01 38 8D 00 00 00 00 10 00 00 00 00 00 5A A6 12 53 6B 0A				
有效内容元素	SystemInfo	GestureInfo	TouchInfo	Air-WheelInfo	xyzPosition
小尾数 16 进制数	8D	00 00 00 00	10 00 00 00	00 00	5A A6 12 53 6B 0A
解码的 16 进制数	0x8D	0x00000000	0x00000010	0x0000	字节 1 和 2：0xA65A 字节 3 和 4：0x5312 字节 5 和 6：0x0A6B
说明	Bit 0：PositionValid Bit 2：RawDataValid Bit 3：NoisePowerValid Bit 7：DSPRunning	未检测到手势	触摸中央电极	无 AirWheel 数据	x：42586 y：21266 z：2667

表 3-14： **从 MGC3130 到主机的消息：位置**

原始消息	18 08 44 91 1E 01 41 8D 00 00 00 00 00 00 00 00 00 00 2F B2 E7 87 6A 35				
有效内容元素	SystemInfo	GestureInfo	TouchInfo	Air-WheelInfo	xyzPosition
小尾数 16 进制数	8D	00 00 00 00	00 00 00 00	00 00	2F B2 E7 87 6A 35
解码的 16 进制数	0x8D	0x00000000	0x00000000	0x0000	字节 1 和 2：0xB22F 字节 3 和 4：0x87E7 字节 5 和 6：0x356a
说明	Bit 0：PositionValid Bit 2：RawDataValid Bit 3：NoisePowerValid Bit 7：DSPRunning	未检测到手势	触摸中央电极	无 AirWheel 数据	x：45615 y：34791 z：13674

© 2014 Microchip Technology Inc.

第 4 章 GestIC 库消息参考

4.1 SYSTEM_STATUS

System_Status 用于确认收到主机的消息。该消息包含错误代码，并用于确认以下消息的传输：

- Request_Message (0x06)
- Set_Runtime_Parameter (0xA2)
- FW_Update_Start (0x80)
- FW_Update_Block (0x81)
- FW_Update_Completed (0x82)

方向：MGC3130 到主机

表 4-1： 消息概述

头				有效内容				
消息大小	标志	序列号	ID	MsgID	MaxCmdSize	ErrorCode	Reserved	Reserved
1 个字节	*1 个字节*	*1 个字节*	*1 个字节*	*1 个字节*	*1 个字节*	*2 个字节*	*4 个字节*	*4 个字节*
0x10	不适用	不适用	0x15	参见下面的说明				

表 4-2： 有效内容元素

元素	元素大小（字节）	说明
MsgID	1	包含 System_Status 对应的消息 ID ***结构***：1 个字节 ***范围***：（0x00..0xFF）
MaxCmdSize	1	包含 GestIC® 库可接收的最大 I²C 数据包大小（包括头） ***结构***：1 个字节 ***范围***：（0x00..0xFF）
ErrorCode	2	错误代码，针对前一条消息返回。 ***结构***：包含专用值的 16 位字（见下面的列表） 可能值： 以下错误代码由库加载程序、库加载程序更新程序和库发送 0x0000 NoError — 正常 0x0001 UnknownCommand — 消息 ID 未知 以下错误代码由库加载程序发送 0x0002 InvalidSessionId — 会话 ID 无效或不匹配（不允许 0x0）（消息：Fw_Update_Start, Fw_Update_Completed） 0x003 InvalidCrc — CRC 无效（消息：Fw_Update_Block, Fw_Update_Start,Fw_Update_Completed） 0x0004 InvalidLength — 长度无效（消息：Fw_Update_Block） 0x0005 InvalidAddress — 地址无效（消息：Fw_Update_Block） 0x0006 InvalidFunction — 功能 ID 无效（消息：Fw_Update_Start,Fw_Update_Block, Fw_Update_Completed） 0x0008 ContentMismatch — VerifyOnly 函数发现内容和闪存中不匹配（消息：Fw_Update_Block） 0x000B WrongParameterAddr — 包含在要加载的新库固件中的参数起始地址与库加载程序假设的不匹配。因此库更新中止。（消息：Fw_Update_Start） 以下错误代码由库发送 0x0014 WrongParameterValue — RuntimeParameter 命令中实参 / 形参的值超出了有效范围（消息：Request_Message 和 Set_Runtime_Parameter） 0x0015 UnknownParameterID — 消息 ID 或运行时参数 ID 未知或超出有效范围（消息：Request_Message 或 Set_Runtime_Parameter） 0x001A WakeupHappend — 检测到主机唤醒事件 以下代码由库加载程序更新程序发送 0x0080 LoaderUpdateStarted — 启动库加载程序更新 0x0081 LoaderUpdateFinished — 完成库加载程序更新

© 2014 Microchip Technology Inc.

表 4-2： 有效内容元素 （续）

元素	元素大小 （字节）	说明
Reserved	4	保留
Reserved	4	保留

4.2 REQUEST_MESSAGE

Request_Message 强制 GestIC 库回复具有请求 ID 的消息。

方向：主机到 MGC3130

表 4-3： 消息概述

头				有效内容		
消息大小	标志	序列号	ID	MessageID	Reserved	Param.
1 个字节	*1 个字节*	*1 个字节*	*1 个字节*	*1 个字节*	*3 个字节*	*4 个字节*
0x0C	不适用	不适用	0x06	参见下面的说明		

表 4-4： 有效内容元素

元素	元素大小 （字节）	说明
MessageID	1	请求来自 GestIC® 库且 ID 为 MessageID 的消息。 GestIC 库应以请求的消息回复或保持沉默。 *结构*: 单个字节，读为十六进制值 *范围*: （0x00..0xFF）
Reserved	3	保留，写为 0。
Param.	4	可选，参数可用于指定返回类别。 示例：请求消息 SetRuntimeParameter,param. 指定 RuntimeParameterId 来回读参数。 *结构*: 32 位字，包含专用值或位域。 *范围*: （0x00000000..0xFFFFFFFF）

注 **1：** Request_Message 命令只能与 MessageID 0x83 和 0xA2 一起使用。

2： TransFreqSelect 运行时参数是只写参数，不能通过消息 Request_Message 请求。

3： 欲知 Request_Message 命令的完整列表，请参见表 A-1。

4.3 FW_VERSION_INFO

在启动时， MGC3130 向主机接口发送 FW_Version_Info 消息，表明芯片处于活动状态且准备好工作。还可使用 Request_Message (0x06) 请求 FW_Version_Info。

> **注：** 有效内容元素 HWRev、 ParameterStartAddr 和 LibraryLoaderVersion 仅在 MGC3130 启动后有效。

方向：MGC3130 到主机。

表 4-5： 消息概述

头				有效内容					
消息大小	标志	序列号	ID	FWValid	HwRev	ParameterStartAddr	LibraryLoaderVersion	FWStartAddr	FWVersion
1 个字节	*1 个字节*	*1 个字节*	*1 个字节*	*1 个字节*	*2 个字节*	*1 个字节*	*3 个字节*	*1 个字节*	*120 个字节*
0x84	不适用	不适用	0x83	参见下面的说明					

表 4-6： 有效内容元素

元素	元素大小（字节）	说明
FWValid	1	GestIC® 库的状态。 *结构*: 包含专用值的单个字节 （见下面的列表） *可能值*: 0x00 Empty 未找到有效的 GestIC® 库 0x0A InvalidFW 存储了无效的 GestIC® 库，或上次更新失败 0xAA ValidFW 存在有效 GestIC® 库
HwRev	2	硬件版本信息 *结构*: 2 个字节的向量，解析为 xx.xx 格式的十进制值 *范围*: （0x00..0xFF, 0x00..0xFF）
ParameterStartAddr	1	图像支持的参数起始地址 地址 = 128 * ParameterStartAddr 值 *结构*: 1 个字节，解析为十六进制值 *范围*: （0x00..0xFF）
LibraryLoaderVersion	3	GestIC® 库加载程序版本信息 *结构*: 3 个字节的向量，解析为 xx.xx.xx 格式的十进制值 *范围*: （0x00..0xFF, 0x00..0xFF,0x00..0xFF）
FWStartAddr	1	引导加载程序支持的 GestIC® 库的起始地址，起始地址 = 128 * FWStartAddr 值 *结构*: 1 个字节，解析为十六进制值 *范围*: （0x00..0xFF）
FWVersion	120	有效 GestIC® 库的版本信息 （FWValid 不是 0x00）。版本字符串解析为 ASCII 码字符。它是一个以分号分隔的字符串，始终以版本号开始，后跟不同的标记： **支持的标记**: p 平台 （如 HillstarVxx） DSP Colibri Suite 版本 （如 ID45r -1167） s 保留 c 保留 t 创建时间 （如 2013/04/24 14:24:50） *结构*: 120 字节的向量，解析为字符串 （ASCII 码字符） *范围*: （0x00..0xFF, 0x00..0xFF, 0x00..0xFF, ...）

© 2014 Microchip Technology Inc.

4.4 SET_RUNTIME_PARAMETER

该消息用于设置 GestIC 库内的运行时参数。它支持用于 AFE 参数化、特性配置和传感器数据输出的参数。为在闪存中持久保存参数而定义一个特殊值。持久保存的参数可分为三类：

- **模拟前端 （Analog Front End， AFE）类**
- **数字信号处理 （Digital Signal Processing， DSP）类**
- **系统类**

方向：主机到 MGC3130。

表 4-7： 消息概述

头				有效内容			
消息大小	标志	序列号	ID	RuntimeParameterID	Reserved	Argument0	Argument1
1 个字节	*1 个字节*	*1 个字节*	*1 个字节*	*2 个字节*	*2 个字节*	*4 个字节*	*4 个字节*
0x10	不适用	不适用	0xA2	参见下面的说明			

表 4-8： 有效内容元素

元素	元素大小 （字节）	说明
RuntimeParameterID	2	运行时参数 ID。请参见**第 4.4.1 节 " 触发 "** 至**第 4.4.5.5 节 " 数据输出锁定屏蔽 "**。 ***结构***: 16 位字，解析为十六进制值 ***范围***: （0x0000..0xFFFF）
Reserved	2	写为 0
Argument0	4	参数值，取决于运行时参数 ID。如果未使用， Argument0 应为 0。 ***结构***: 32 位字：Argument0 ***范围***: 取决于运行时参数
Argument1	4	参数值，取决于运行时参数 ID。如果未使用， Argument1 应为 0。 ***结构***: 32 位字：Argument1 ***范围***: 取决于运行时参数

4.4.1 触发

该参数强制执行 Argument0 中定义的触发。

RuntimeParameterID 0x1000 Trigger 参数强制触发。

Argument0 0x00000000：强制再校准

0x00000002：进入深度休眠 1：深度休眠 1 的唤醒源：I2C0 起始位检测或 $\overline{\text{MCLR}}$ 复位。总线上发送的任何 I^2C 消息均能使系统从深度休眠恢复工作，且第一条 I^2C 消息将丢失。

0x00000003：进入深度休眠 2：深度休眠 2 的唤醒源：外部中断引脚 （IRQ0）的下降沿或 $\overline{\text{MCLR}}$ 复位。发出进入深度休眠 2 的命令后，应将 IRQ0 （EIO2）引脚拉至高电平，否则 MGC3130 将在接收到该命令后直接被唤醒。

范围:（0x00000000, 0x00000002, 0x00000003）

Argument1 未使用

4.4.2 持久

使用该 ID 使 Argument0 中定义的参数集持久 （存储在闪存中）。

RuntimeParameterID	0xFF00 MakePersistent 将参数存储在闪存中。
Argument0	0x00000000：存储 AFE 类的 RTP 0x00000001：存储 DSP 类的 RTPy 0x00000002：存储系统类的 RTP *范围*: （0x00000000, 0x00000001, 00000002）
Argument1	未用

4.4.3 模拟前端 （AFE）类型

4.4.3.1 信号匹配

信号匹配参数用于调整采样点处的 Rx 信号电平。

RuntimeParameterID	0x50	afeRxAtt_S	南电极的信号匹配参数
	0x51	afeRxAtt_W	西电极的信号匹配参数
	0x52	afeRxAtt_N	北电极的信号匹配参数
	0x53	afeRxAtt_E	东电极的信号匹配参数
	0x54	afeRxAtt_C	中央电极的信号匹配因子
Argument0	包含值 *范围*: （0x00000000..0x000000FF）		
Argument1	未使用		

© 2014 Microchip Technology Inc.

4.4.3.2 电极映射

分配给电极的物理通道编号。这些参数表示电极与 MGC3130 Rx 通道的物理连接。为了可正确工作，必须在电路设计中查看映射。

RuntimeParameterID	0x65	Channelmapping_S	分配给南电极的物理通道
	0x66	Channelmapping_W	分配给西电极的物理通道
	0x67	Channelmapping_N	分配给北电极的物理通道
	0x68	Channelmapping_E	分配给东电极的物理通道
	0x69	Channelmapping_C	分配给中央电极的物理通道
Argument0	包含物理接收通道的编号（Rx0、Rx1、Rx2、Rx3 和 Rx4） *范围*：（0x00000000, 0x00000001, 0x00000002, 0x00000003, 0x00000004）		
Argument1	未使用。		

4.4.4 数字信号处理 （DSP）类

4.4.4.1 发送频率选择

设置使用的发送器频率数量以及它们进行跳频测试的顺序。

RuntimeParameterID	0x82	TransFreqSelect	参数用于设置使用的频率 ID

Argument0	所用 Tx 频率的数量。该参数可以为 1、 2、 3、 4 或 5。
Argument1	该参数决定发送器频率的测试顺序。索引编号0至4表示相应的发送器频率： - 频率 ID 0 对应 115 kHz - 频率 ID 1 对应 103 kHz - 频率 ID 2 对应 88 kHz - 频率 ID 3 对应 67 kHz - 频率 ID 4 对应 44 kHz 这些索引必须以半字节形式提供。
示例：	Argument0 = 0x04 与 Argument1 = 0x3104 表示索引 4、 0、 1 和 3 对应的频率以该特定顺序使用和测试。 索引 – 默认频率映射 （Argument 0 = 0x5， Argument 1 = 0x43210） 频率 ID 0 – 发送器频率：115 kHz 频率 ID 1 – 发送器频率：103 kHz 频率 ID 2 – 发送器频率：88 kHz 频率 ID 3 – 发送器频率：67 kHz 频率 ID 4 – 发送器频率：44 kHz

> **注：** TransFreqSelect 运行时参数是一个只读参数，不能用消息 REQUEST_MESSAGE (0x06) 请求。

© 2014 Microchip Technology Inc.

4.4.4.2 触摸检测

该参数用于使能 / 禁止触摸检测。

RuntimeParameterID	0x97	dspTouchConfig	参数用于使能 / 禁止触摸检测
Argument0	Argument0 设置为 0x08 以使能触摸检测， Argument0 设置为 0x00 以禁止触摸检测 **注：** 如果 Argument1 未正确设置，系统将显示故障。		
Argument1	0x08		

4.4.4.3 接近检测

该参数用于使能 / 禁止接近检测模式。

RuntimeParameterID	0x97	dspAutoWakeupMode	参数用于使能 / 禁止接近检测模式
Argument0	Argument0 设置为 0x01 以使能接近检测， Argument0 设置为 0x00 以禁止接近检测 **注：** 如果 Argument1 未正确设置，系统将显示故障。		
Argument1	0x01		

> **注：** 对于 V1.0 之前的版本，接近检测 RuntimeParameterID 为 0x81，Argument0 和 Argument1 的定义和现在一样。V1.1 及之后版本不再支持该 RTC。为兼容起见， Aurea PC 软件仍然使用该 RTC。

4.4.5 系统类

4.4.5.1 AIRWHEEL

该参数用于使能 / 禁止 AirWheel。

RuntimeParameterID	0x90 dspAirWheelConig	参数用于使能/禁止AirWheel
Argument0	Argument0 设置为 0x20 以使能 AirWheel， Argument0 设置为 0x00 以禁止 AirWheel **注：** 如果 Argument1 未正确设置，系统将显示故障。	
Argument1	0x20	

4.4.5.2 手势处理 （HMM）

该参数用于使能内置手势。禁止一种手势将提高其他手势的识别率。

若 Argument0 中的某位设置为 1，将使能相应的手势。若 Argument0 中的某位设置为 0，将禁止相应的手势。

RuntimeParameterID	0x85 dspGestureMask	参数用于使能 / 禁止手势
Argument0	Bit 0：垃圾模型 Bit 1：从西到东划动 Bit 2：从东到西划动 Bit 3：从南到北划动 Bit 4：从北到南划动 Bit 5：顺时针画圈 Bit 6：逆时针画圈	
Argument1	充当屏蔽，将适当位置 1 以更改标志。所有其他标志保持不变。	

4.4.5.3 校准工作模式

该参数用于使能 / 禁止选定的自动校准特性。

如果 Argument0 中的某位设置为 0，将使能相应的自动校准功能。

如果 Argument0 中的某位设置为 1，将禁止相应的自动校准功能。

RuntimeParameterID	0x80 dspCalOpModeFlags	参数用于使能 / 禁止自动校准
Argument0	Bit 0：使能 / 禁止启动校准 Bit 1：使能 / 禁止手势触发校准 Bit 2：使能 / 禁止负校准 Bit 3：使能 / 禁止空闲校准 Bit 4：使能 / 禁止无效值校准，若值完全超出范围 Bit 5：使能 / 禁止由 AFA 触发校准	
Argument1	充当屏蔽，将适当位置 1 以更改标志。所有其他标志保持不变。	

© 2014 Microchip Technology Inc.

4.4.5.4 数据输出使能屏蔽

该参数确定消息 Sensor_Data_Output (0x91) 的数据输出。如果 Argument0 中的某位置 1，则对应的有效内容元素将是消息 Sensor_Data_Output (0x91) 的一部分。如果 Argument0 中的某位设置为 0，则对应的有效内容元素将不是消息 Sensor_Data_Output (0x91) 的一部分。

使用 DataOutputEnableMask 从 I^2C 利用率和主机代码效率方面优化传感器数据输出。

请注意，使能所有有效内容元素可能因 I^2C 总线的带宽限制而导致故障。

RuntimeParameterID 0xA0 DataOutputEnableMask 参数确定数据输出。

Argument0	Bit 0：DSP 状态 Bit 1：手势数据 Bit 2：触摸数据 Bit 3：AirWheel 数据 Bit 4：位置数据 Bit 5：噪声功率 Bit 6...10：这些位保留，且必须设置为 0。 Bit 11：未校准信号 （CIC）数据。 Bit 12：信号偏差 （Signal Deviation， SD）数据。 Bit 13...15：这些位保留，且必须设置为 0。
Argument1	充当屏蔽，将适当位置 1 以更改标志。所有其他标志保持不变。

4.4.5.5 数据输出锁定屏蔽

该参数确定消息 Sensor_Data_Output (0x91) 的数据输出。如果 Argument0 中的某位置 1，则对应的有效内容元素将是消息 Sensor_Data_Output (0x91) 的一部分，无论是否有新数据 （有效内容元素被“锁定”）。

如果 Argument0 中的某位设置为 0，则在数据更新时 （有效内容元素是变量），有效内容元素将只是消息 Sensor_Data_Output (0x91) 的一部分。

RuntimeParameterID 0xA1 DataOutputLockMask 参数确定数据输出。

Argument0	Bit 0：DSP 状态 Bit 1：手势数据 Bit 2：触摸数据 Bit 3：AirWheel 数据 Bit 4：位置数据 Bit 5：噪声功率 Bit 6...10：这些位保留，且必须设置为 0。 Bit 11：未校准信号 （CIC）数据。 Bit 12：信号偏差 （SD）数据。 Bit 13...15：这些位保留，且必须设置为 0。
Argument1	充当屏蔽，将适当位置 1 以更改标志。所有其他标志保持不变。

© 2014 Microchip Technology Inc.

4.4.5.6 数据输出请求屏蔽

该参数仅决定下一条消息 Sensor_Data_Output (0x91) 的数据输出。若 Argument0 中的位设置为 1，则相应的有效内容元素将是下一条消息 Sensor_Data_Output (0x91) 的一部分。

若 Argument0 中的位设置为 0，则数据更新时，有效内容元素将不是下一条消息 Sensor_Data_Output (0x91) 的一部分。

这将强制 MGC3130 发送新的消息 Sensor_Data_Output (0x91)，即使没有有效的事件和数据。根据 Argument0 的选择，该消息将包含数据。然后，仅在有效事件和数据发生时，根据数据输出使能和锁定屏蔽发送 Sensor_Data_Output (0x91)。

RuntimeParameterID	0xA2	DataOutputRequestMask 决定下一个数据输出的参数。
Argument0	Bit 0：DSP 状态 Bit 1：手势数据 Bit 2：触摸数据 Bit 3：AirWheel 数据 Bit 4：位置数据 Bit 5：噪声功率 Bits 6...10：这些位保留，必须设置为 0。 Bit 11：未校准信号 （CIC）数据。 Bit 12：信号偏差 （Signal Deviation， SD）数据。 Bit 13...15：这些位保留，必须设置为 0。	
Argument1	充当屏蔽，将适当位置 1 以更改标志。所有其他标志保持不变。	

4.4.5.7 手势进行中标志控制

该参数决定手势进行中输出是否是消息 Sensor_Data_Output (0x91) 中 GestureInfo 数据输出的一部分。若 Argument0 设置为 0x1，手势进行中将会在消息 Sensor_Data_Output (0x91) 的 GestureInfo 字段 （bit 31）中输出。

若 Argument0 设置为 0x0，手势进行中将不会在消息 Sensor_Data_Output (0x91) 中的 GestureInfo 字段 （bit 31）中输出。

欲知更多详细信息，请参见**第 4.5 节 “Sensor_Data_Output”**。

RuntimeParameterID	0xA3	DataOutputGestureInProgress 该参数使能或禁止手势进行中标志在 GestureInfo 字段中输出。
Argument0	0x00000000：禁止手势进行中输出 0x00000001：使能手势进行中输出	
Argument1	0x00000001	

> **注：** 欲知 Set_Runtime_Parameter 命令示例的完整列表，请参见表 A-2。

4.5 SENSOR_DATA_OUTPUT

该消息包含 MGC3130 的传感器数据输出。消息内容可通过位屏蔽进行配置（见**第 4.4 节 "Set_Runtime_Parameter"** 中的 DataOutputEnableMask 和 DataOutputLockMask）。

元素 DataOutputConfigMask、 TimeStamp 和 SystemInfo 始终是消息的一部分。包含更多有效内容元素取决于具体配置，且实际配置可通过有效内容元素 DataOutputConfigMask 读取。

方向：MGC3130 到主机

表 4-9： 消息概述

头				有效内容			
大小	标志	序列号	ID	DataOutputConfigMask	TimeStamp	SystemInfo	取决于 DataOutputConfigMask 的变量
1 个字节	*1 个字节*	*1 个字节*	*1 个字节*	*2 个字节*	*1 个字节*	*1 字个节*	取决于 DataOutputConfigMask 的变量
变量	不适用	不适用	0x91	参见下面的说明			

表 4-10： 有效内容元素

元素	元素大小（字节）	说明
DataOutputConfig-Mask	2	位屏蔽，指示哪些数据是消息的一部分。 使用以下位： Bit 0： DSPStatus 字段。 Bit 1： GestureInfo 字段。 Bit 2： TouchInfo 字段。 Bit 3： AirWheelInfo 字段。 Bit 4： xyzPosition 字段。 Bit 5： NoisePower 字段。 Bit 6： 该位保留。 Bit 7： 该位保留。 Bit 8...10： ElectrodeConfiguration 000:ChCnt = 4，四种电极配置，没有中央电极 001:ChCnt = 5，五种电极配置，有中央电极 Bit 11： 带 chCnt 条目的 CICData 字段。 Bit 12： 带 chCnt 条目的 SDData 字段。 Bit 13...15： 这些位保留。 *结构*: 16 位字，读为位屏蔽。 *范围*: （0x0000..0xFFFF）
TimeStamp	1	200 Hz （采样时间间隔）的 8 位计数器 200 Hz 计数器值在达到 256 后计满返回。这指示事件何时发生，并支持测量两次事件之间间隔的时间，前提是该时间大约在 1.25 秒以下。 *结构*: 8 位字，读为十进制值。 *范围*: （0x00..0xFF）
SystemInfo	1	位屏蔽，指示相应的传感器数据是否有效。在应用程序中，仅当各位置 1 时，才应进一步处理传感器数据输出。 使用以下位： Bit 0： PositionValid，如果置 1，则表示 xyzPosition 字段中的位置有效。 Bit 1： AirWheelValid，如果置 1，则表示 AirWheel 有效，且 AirWheelInfo 字段中的数据有效。 Bit 2： RawDataValid，如果置 1，则表示 CICData 和 SDData 字段的数据有效。否则，必须忽略这些字段。 Bit 3： NoisePowerValid，如果置 1，则表示 NoisePower 字段有效。 Bit 4： EnvironmentalNoise，如果置 1，则表示检测到环境噪声。 Bit 5： Clipping，如果置 1，则表示 ADC 产生削波。 Bit 6： 该位保留。 Bit 7： DSPRunning，如果置 1，则表示系统正在运行。如果未置 1，系统即将休眠。 *结构*: 8 位字，读为位屏蔽。 *范围*: （0x00..0xFF） *注：* 禁止位置数据从传感器数据输出并使能 AirWheel：位置有效将和 SystemInfo 一起发送，并在 AirWheel 检测开始时发送新消息。

© 2014 Microchip Technology Inc.

表 4-10： 有效内容元素 （续）

元素	元素大小（字节）	说明
DSPstatus	2	该元素包含两个字节。第一个字节包含校准事件相关信息。第二个字节指示目前使用的 Tx 频率。 Bit 0： 该位保留。 Bit 1： CalibrationInfo：（主机）强制校准 Bit 2： CalibrationInfo：启动校准 Bit 3： CalibrationInfo：触发手势 Bit 4： CalibrationInfo：负值 Bit 5： CalibrationInfo：空闲校准 Bit 6： CalibrationInfo：无效值校准 Bit 7： 校准：由 AFA 触发校准 Bit 8...15： Tx 频率 （kHz）手势，为十进制值 （44..115） *结构*：2 个字节，第一个字节读为位屏蔽，第二个字节读为十进制值。 *范围*：（0x00..0xFF； 44..115）
GestureInfo	4	该字段包含 32 位手势信息字。 *识别的手势*： 识别的手势是 HMM 分类的结果。边缘检测可用于将完成手势的位置进一步分类 （边缘弹指）。此外，手势属性提供了弹指方向的相关信息。手势信息作为位域给出，可解码如下： Bit 7...0： 识别的手势，为十进制数 0： 无手势 1： 垃圾模型 2： 从西到东划动 3： 从东到西划动 4： 从南到北划动 5： 从北到南划动 6： 顺时针画圈 （仅在 AirWheel 禁止时有效） 7： 逆时针画圈 （仅在 AirWheel 禁止时有效） Bit 8...11： 这些位不得解析。 Bit 12…15： 手势类，读为十进制数 0： 垃圾模型 1： 弹指手势 2： 画圈手势 Bit 16： 边缘弹指 —— 如果弹指手势分类为边缘弹指，则为 1。 Bit 17...30： 这些位保留。 Bit 31： 手势识别正在进行。当手势识别器激活时，该位置 1 ；手势被识别后，该位复位，识别器关闭。 0： 手势识别不在进行 1： 手势识别正在进行 *结构*：32 位字，读为位屏蔽 *范围*：（0x00000000..0xFFFFFFFF）
TouchInfo	4	包含触摸信息 下列位用于指示各电极上发生的触摸事件： Bit 0： 触摸南电极 Bit 1： 触摸西电极 Bit 2： 触摸北电极 Bit 3： 触摸东电极 Bit 4： 触摸中央电极 Bit 5： 单击南电极 Bit 6： 单击西电极 Bit 7： 单击北电极 Bit 8： 单击东电极 Bit 9： 单击中央电极 Bit 10： 双击南电极 Bit 11： 双击西电极 Bit 12： 双击北电极 Bit 13： 双击东电极 Bit 14： 双击中央电极 Bit 15: 该位保留。 Bits 16...23: 触摸计数器：8 位计数器。该计数器确定手开始移动到检测到触摸之间的周期。该周期等于 [触摸计数器值] x 5 (ms)。计数器在检测触摸事件所需的最小接近速度被超出开始计数，直到检测到触摸。每次触摸检测后，计数器复位。 Bits 24...31： 这些位保留 *结构*：32 位字，读为位屏蔽 *范围*：（0x00000000..0xFFFFFFFF）
AirWheelInfo	2	第一个字节包含一个计数器，指示 AirWheel 旋转的进展情况。递增值表示顺时针旋转。递减值表示逆时针旋转。32 次递增约为一个完整旋转。AirWheelInfo 仅在元素 SystemInfo 的 AirWheelValid 位置 1 时才有效。 第二个字节保留。 *结构*：两个 8 位字向量，读为十进制值 *范围*：（0x0000..0x00FF）

表 4-10： 有效内容元素 （续）

元素	元素大小（字节）	说明
xyzPosition	6	该元素包含 x、 y 和 z 位置数据。两个字节用于每个位置 x、 y 和 z。 字节 1 和 2： x 位置 字节 3 和 4： y 位置 字节 5 和 6： z 位置 位置信息仅在元素 SystemInfo 的 PositionValid 位置 1 时才有效。 数据给出了用户的手在笛卡尔坐标系中的位置。位置数据 [0, 0, 0] 表示坐标系原点，数据 [65535, 65535, 65535] 是传感空间的最大尺寸。原点定义为系统表面传感空间的左下角 （南 — 西）。 *结构*: 3 个 16 位字向量，读为每个位置 x、 y 和 z 的十进制值 *范围*: 每个位置 x、 y 和 z 为 （0x0000..0xFFFF）
NoisePower	4	GestIC® 系统的噪声功率。 NoisePower 仅在元素 SystemInfo 的 NoisePowerValid 位置 1 时才有效。 *结构*: 32 位字，读为浮点值 *范围*: （0..3.402823e+38）
CICData	4xChCnt	未校准传感器数据 （CIC 数据） 元素大小取决于有效内容元素 DataOutputConfigMask bit 8...10 指示的 ChCnt。 CICData 仅在元素 SystemInfo 的 RawDataValid 位置 1 时才有效。 *结构*: 4 个向量 （每个向量均由 5 个 32 位字组成），解析为浮点值，格式为 xxxx.xxxx.xxxx.xxxx.xxxx（南 . 西 . 北 . 东 . 中央） *范围*: 每个通道为 （-3.402823e+38..3.402823e+38）
SDData	4xChCnt	信号偏差 （SD） 元素大小取决于有效内容元素 DataOutputConfigMask bit 8...10 指示的 ChCnt。 SDData 仅在元素 SystemInfo 的 RawDataValid 位置 1 时才有效。 *结构*: 4 个向量 （每个向量均由 5 个 32 位字组成），解析为浮点值，格式为 xxxx.xxxx.xxxx.xxxx.xxxx（南 . 西 . 北 . 东 . 中央） *范围*: 每个通道为 （-3.402823e+38...3.402823e+38）
Reserved	—	保留：将来可添加其他有效内容元素或供内部使用。

注： 欲知 Sensor_Data_Output 命令示例的完整列表，请参见表 A-3。

© 2014 Microchip Technology Inc.

第 5 章 GestIC 库更新消息

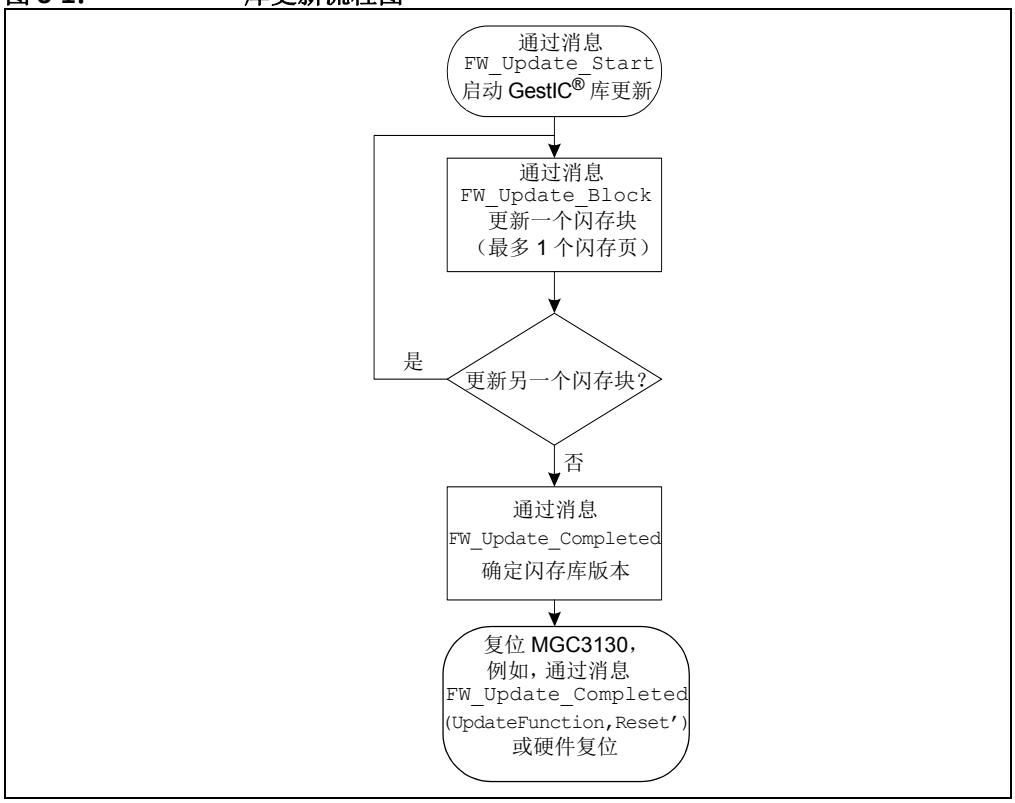
5.1 库加载程序更新过程

通用库更新过程如图 5-1 所示。请注意，只可在 MGC3130 上更新 Microchip Technology 提供的库。此外， Microchip 仅依照请求提供详细描述库更新过程的应用笔记。

对于库更新过程，需要三条不同消息：

- FW_Update_Start （消息 ID—0x80）
- FW_Update_Block （消息 ID—0x81）
- FW_Update_Completed （消息 ID—0x82）

图 5-1： 库更新流程图



5.2 FW_UPDATE_START

该消息用于启动 MGC3130 器件的更新会话。

方向：主机到 MGC3130

表 5-1： 消息概述

头				有效内容				
消息大小	标志	序列号	ID	Crc	SessionID	IV	UpdateFunction	Reserved
1 个字节	*1 个字节*	*1 个字节*	*1 个字节*	*4 个字节*	*4 个字节*	*14 个字节*	*1 个字节*	*1 个字节*
0x1C	不适用	不适用	0x80	参见下面的说明				

表 5-2： 有效内容元素

字段	大小（字节）	说明
Crc	4	根据消息余下信息 （20 字节）计数出的值 CRC32 （以太网，多项式：0x04C11DB7） *结构*: 32 位字 *范围*: （0x00000000..0xffffffff）
SessionId	4	SessionID 是主机生成的随机数。它必须在 FW_Update_Completed 消息中再发送，否则，会话将无效。 0x00000000 是无效 SessionID，用于强制器件进入等待循环。在这种情况下，消息中的其余信息被丢弃。 *结构*: 32 位字 *范围*: （0x00000000..0xffffffff）
IV	14	14 字节值，用于加密数据。 *结构*: 14 字节向量 *范围*: （0x00..0xFF, 0x00..0xFF, 0x00..0xFF, ...）
UpdateFunction	1	UpdateFunction 设置整个更新会话的模式： - 如果会话模式设置为 ProgramFlash，则下列 FW_Update_Block 消息的有效内容写入闪存。 - 如果会话模式设置为 VerifyOnly，则仅验证代码 （比较闪存内容与 FW_Update_Block 消息的解密有效内容），但不写入闪存。如果解密的有效内容和闪存不匹配，则返回带有 Error 8 （ContentMismatch）的 System_Status 消息。 **注：** 下列 FW_Update_Block 消息还包含一个 UpdateFunction 字段。该字段定义了单个更新块的模式。 然而： - 如果会话模式通过 FW_Update_Start 设置为 ProgramFlash，则单个 FW_Update_Blocks 的 UpdateFunction 可设置为 ProgramFlash 或 VerifyOnly。 - 如果会话模式通过 FW_Update_Start 设置为 VerifyOnly，则单个 FW_Update_Blocks 的 UpdateFunction 只能设置为 VerifyOnly。 *结构*: 单个字节，包含专用值 （见下面的列表） *可能值*: 0 ProgramFlash 1 VerifyOnly
Reserved	1	保留

© 2014 Microchip Technology Inc.

5.3 FW_UPDATE_BLOCK

该消息更新一个闪存块。一个块的大小最高可达 128 字节。

方向：主机到 MGC3130

表 5-3： 消息概述

头				有效内容				
消息大小	标志	序列号	ID	Crc	Address	Length	UpdateFunction	Payload
1 个字节	*1 个字节*	*1 个字节*	*1 个字节*	*4 个字节*	*2 个字节*	*1 个字节*	*1 个字节*	*128 个字节*
0x8C	不适用	不适用	0x81	参见下面的说明				

表 5-4： 有效内容元素

字段	大小 （字节）	说明
Crc	4	根据消息的其余信息 （132 字节）计算 CRC32 （以太网，多项式： 0x04C11DB7）的值 *结构*: 32 位字 *范围*: （0x00000000..0xffffffff）
Address	2	将编程 / 验证的块的闪存地址。如果块小于 128 字节，它必须在每页末尾对齐。因此，如果下一个更新块是一个完整的 128 字节块，则它可再次闪存页对齐。 注： 低 4 KB 保留用于库加载程序，不能更新。如果使用低于 4 KB 的值，则返回带有 Error 5 （InvalidAddress）的 System_Status 消息。 *结构*: 16 位字 *范围*: （0x1000..0x7fff）
Length	1	将更新的块的内容长度： *结构*: 单个字节 *范围*: （0x00..0x80）
UpdateFunction	1	UpdateFunction 设置该单个更新块的模式。 - 如果模式设置为 ProgramFlash，则解密的有效内容写入闪存。 - 如果会话模式设置为 VerifyOnly，则仅验证代码 （比较闪存内容与解密的有效内容），但不写入闪存。如果解密的有效内容和闪存不匹配，则返回带有 Error 8 （ContentMismatch）的 System_Status 消息。 注： 如果整个会话的模式在 FW_Update_Start 消息中设置为 VerifyOnly，则只可在 FW_Update_Block 中设置 VerifyOnly ； 否则，返回带有 Error 6 （InvalidFunction）的 System_Status 消息。 *结构*: 单个字节，包含专用值 （见下面的列表） *可能值*: 0 ProgramFlash 1 VerifyOnly
Payload	128	Payload 包含将更新的块的加密内容： 注： 其长度始终为 128。如果内容长度小于 128，则将用 0 填满。然后，对整个 128 字节有效内容计算 Crc。 *结构*: 120 字节向量，解析为字符串 （ASCII 码字符） *范围*: （0x00..0xFF, 0x00..0xFF, 0x00..0xFF, . ..）

© 2014 Microchip Technology Inc.

5.4 FW_UPDATE_COMPLETED

该消息用于结束 MGC3130 的更新会话。

方向：主机到 MGC3130

表 5-5： 消息概述

头				有效内容				
消息大小	标志	序列号	ID	Crc	SessionID	UpdateFunction	FWVersion	Reserved
1 个字节	*1 个字节*	*1 个字节*	*1 个字节*	*4 个字节*	*4 个字节*	*1 个字节*	*120 个字节*	*3 个字节*
0x88	不适用	不适用	0x82	参见下面的说明				

表 5-6： 有效内容元素

字段	大小（字节）	说明
Crc	4	根据消息的其余信息（128 字节）计算 CRC32（以太网，多项式：0x04C11DB7）的值 *结构*: 32 位字 *范围*:（0x00000000..0xffffffff）
SessionID	4	SessionID 是用于 FW_Update_Start 的同一随机数。0x00000000 是无效 SessionID，强制器件重新启动。在这种情况下，消息中的其余信息被丢弃。 *结构*: 32 位字 *范围*:（0x00000000..0xffffffff）
UpdateFunction	1	UpdateFunction 定义了更新会话的结束方式。 - 如果会话以 ProgramFlash 会话开始，则它必须以 ProgramFlash 会话结束。否则，库版本不会存储，且库无效。如果 VerifyOnly 会话使用了 ProgramFlash，则返回带有 Error 6（InvalidFunction）的 System_Status 消息。 - 如果使用了 Restart，则器件将重新启动。FWVersion 和 SessionID 包括在 Crc 计算中，但内容被忽略。 *结构*: 单个字节，包含专用值（见下面的列表） *可能值*: 0 ProgramFlash 编程闪存 1 VerifyOnly 3 Restart
FWVersion	120	它包含库版本。只有 ID 为非 0 的库才有效。 *结构*: 120 字节向量，解析为字符串（ASCII 码字符） *范围*:（0x00..0xFF, 0x00..0xFF, 0x00..0xFF, . ..）
Reserved	3	保留

附录 A I²C™ 命令示例

表 A-1： REQUEST_MESSAGE 命令示例

请求的函数			请求消息												备注
			报文头				有效内容								
			消息长度	标志	序列	ID	消息 ID	保留			参数				
—	固件版本 （0x83）		0x0C	0x0	0x0	0x06	0x83	0x00	0x00	0x00	0x00	0x00	0x00	0x00	固定命令。
获取实时参数	触发信号 （0x1000）		0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0x00	0x10	0x00	0x00	固定命令。
	信号匹配（0x0050、0x0051、0x0052、0x0053 和 0x0054）	AFERXATT_S	0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0x50	0x00	0x00	0x00	固定命令。
		AFERXATT_W	0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0x51	0x00	0x00	0x00	
		AFERXATT_N	0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0x52	0x00	0x00	0x00	
		AFERXATT_E	0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0x53	0x00	0x00	0x00	
		AFERXATT_C	0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0x54	0x00	0x00	0x00	
	电极映射（0x0065、0x0066、0x0067、0x0068 和 0x0069）	Channelmapping_S	0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0x65	0x00	0x00	0x00	固定命令。
		Channelmapping_W	0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0x66	0x00	0x00	0x00	
		Channelmapping_N	0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0x67	0x00	0x00	0x00	
		Channelmapping_E	0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0x68	0x00	0x00	0x00	
		Channelmapping_C	0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0x69	0x00	0x00	0x00	
	触摸检测 （0x0097）和接近检测（0x0097）		0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0x97	0x00	0x00	0x00	固定命令。
	接近检测（0x0081）		0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0x81	0x00	0x00	0x00	固定命令。
	AirWheel（0x0090）		0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0x90	0x00	0x00	0x00	固定命令。
	手势处理 HMM （0x0085）		0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0x85	0x00	0x00	0x00	固定命令。
	校准操作模式 （0x0080）		0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0x80	0x00	0x00	0x00	固定命令。
	数据输出使能屏蔽 （0x00A0）		0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0xA0	0x00	0x00	0x00	固定命令。
	数据输出锁定屏蔽 （0x00A1）		0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0xA1	0x00	0x00	0x00	固定命令。
	数据输出请求屏蔽 （0x00A2）		0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0xA2	0x00	0x00	0x00	固定命令。
	手势进行中标志控制 （0x00A3）		0x0C	0x0	0x0	0x06	0xA2	0x00	0x00	0x00	0xA3	0x00	0x00	0x00	固定命令。

© 2014 Microchip Technology Inc.

表 A-2： SET_RUNTIME_PARAMETER 命令示例

请求的函数			Set_Runtime_Parameter																备注
			报文头				有效内容												
			消息长度	标志	序列	ID	实时参数 ID		保留		Argument0				Argument1				
公共类	触发信号（0x1000）	强制校准	0x10	0x0	0x0	0xA2	0x00	0x10	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	固定命令。
		进入深度休眠 1	0x10	0x0	0x0	0xA2	0x00	0x10	0x00	0x00	0x02	0x00	0x00	0x00	0x00	0x00	0x00	0x00	固定命令。
		进入深度休眠 2	0x10	0x0	0x0	0xA2	0x00	0x10	0x00	0x00	0x03	0x00	0x00	0x00	0x00	0x00	0x00	0x00	固定命令。
	持久（0xFF00）	保存 AFE 的 RTP	0x10	0x0	0x0	0xA2	0x00	0xFF	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	固定命令。
		保存 RTPs for	0x10	0x0	0x0	0xA2	0x00	0xFF	0x00	0x00	0x01	0x00	0x00	0x00	0x00	0x00	0x00	0x00	固定命令。
		Store RTPs for System	0x10	0x0	0x0	0xA2	0x00	0xFF	0x00	0x00	0x02	0x00	0x00	0x00	0x00	0x00	0x00	0x00	固定命令。
模拟前端类	信号匹配（0x0050、0x0051、0x0052、 0x0053、0x0054）	AFERXATT_S	0x10	0x0	0x0	0xA2	0x50	0x00	0x00	0x00	0x98	0x00	0x00	0x00	0x00	0x00	0x00	0x00	Argument0（8 位）定义每个电极的信号匹配值。给出的值，仅是一些示例。
		AFERXATT_W	0x10	0x0	0x0	0xA2	0x51	0x00	0x00	0x00	0x96	0x00	0x00	0x00	0x00	0x00	0x00	0x00	
		AFERXATT_N	0x10	0x0	0x0	0xA2	0x52	0x00	0x00	0x00	0x98	0x00	0x00	0x00	0x00	0x00	0x00	0x00	
		AFERXATT_E	0x10	0x0	0x0	0xA2	0x53	0x00	0x00	0x00	0x91	0x00	0x00	0x00	0x00	0x00	0x00	0x00	
		AFERXATT_C	0x10	0x0	0x0	0xA2	0x54	0x00	0x00	0x00	0xD9	0x00	0x00	0x00	0x00	0x00	0x00	0x00	
	电极映射（0x0065、0x0066、0x0067、0x0068、 0x0069）	Channelmapping_S	0x10	0x0	0x0	0xA2	0x65	0x00	0x00	0x00	0x03	0x00	0x00	0x00	0x00	0x00	0x00	0x00	Argument0（8 位）定义每个电极的相应 Rx 通道。对于 Rx0，该值为 0，对于 Rx1，该值为 1；对于 Rx2，该值为 2；对于 Rx3，该值为 3；对于 Rx4，该值为 4。给出的值，仅是一些示例。
		Channelmapping_W	0x10	0x0	0x0	0xA2	0x66	0x00	0x00	0x00	0x01	0x00	0x00	0x00	0x00	0x00	0x00	0x00	
		Channelmapping_N	0x10	0x0	0x0	0xA2	0x67	0x00	0x00	0x00	0x02	0x00	0x00	0x00	0x00	0x00	0x00	0x00	
		Channelmapping_E	0x10	0x0	0x0	0xA2	0x68	0x00	0x00	0x00	0x04	0x00	0x00	0x00	0x00	0x00	0x00	0x00	
		Channelmapping_C	0x10	0x0	0x0	0xA2	0x69	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	

© 2014 Microchip Technology Inc.

表 A-2： SET_RUNTIME_PARAMETER 命令示例 （续）

请求的函数			Set_Runtime_Parameter																备注
			报文头				有效内容												
			消息长度	标志	序列	ID	实时参数 ID		保留		Argument0				Argument1				
数字信号处理	发送频率选择（0x0082）	5 种频率	0x10	0x0	0x0	0xA2	0x82	0x00	0x00	0x00	0x05	0x00	0x00	0x00	0x10	0x32	0x04	0x00	给出的是以该顺序（0x43210）使用的 5 种频率的示例：115kHz、 103 kHz、 88 kHz、 67kH、 44kHz
		2 种频率	0x10	0x0	0x0	0xA2	0x82	0x00	0x00	0x00	0x02	0x00	0x00	0x00	0x42	0x00	0x00	0x00	给出的是以该顺序 （0x42）使用的 2 种频率的示例：103 kHz、 44kHz
	触摸检测（0x0097）	使能	0x10	0x0	0x0	0xA2	0x97	0x00	0x00	0x00	0x08	0x00	0x00	0x00	0x08	0x00	0x00	0x00	固定命令。
		禁止	0x10	0x0	0x0	0xA2	0x97	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x08	0x00	0x00	0x00	固定命令。
	接近检测（0x0097）	使能	0x10	0x0	0x0	0xA2	0x97	0x00	0x00	0x00	0x01	0x00	0x00	0x00	0x01	0x00	0x00	0x00	固定命令。
		禁止	0x10	0x0	0x0	0xA2	0x97	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x01	0x00	0x00	0x00	固定命令。
	接近检测（0x0081）	使能	0x10	0x0	0x0	0xA2	0x81	0x00	0x00	0x00	0x01	0x00	0x00	0x00	0x01	0x00	0x00	0x00	固定命令。从 V1.0 版本开始不再支持该命令。请使用 0x97 ID 取而代之。
		禁止	0x10	0x0	0x0	0xA2	0x81	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x01	0x00	0x00	0x00	固定命令。从 V1.0 版本开始不再支持该命令。请使用 0x97 ID 取而代之。

© 2014 Microchip Technology Inc.

表 A-2： SET_RUNTIME_PARAMETER 命令示例 （续）

请求的函数			Set_Runtime_Parameter																备注
			报文头				有效内容												
			消息长度	标志	序列	ID	实时参数 ID		保留		Argument0				Argument1				
系统类	AirWheel（0x0090）	使能	0x10	0x0	0x0	0xA2	0x90	0x00	0x00	0x00	0x20	0x00	0x00	0x00	0x20	0x00	0x00	0x00	固定命令。
		禁止	0x10	0x0	0x0	0xA2	0x90	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x20	0x00	0x00	0x00	固定命令。
	手势处理 HMM（0x0085）	使能所有手势	0x10	0x0	0x0	0xA2	0x85	0x00	0x00	0x00	0x7F	0x00	0x00	0x00	0x7F	0x00	0x00	0x00	Argument 0 （8 位）定义需要配置哪个手势。 Argument 1 定义需要配置的手势的掩码。给出的值，仅为一些示例。
		仅使能滑动手势	0x10	0x0	0x0	0xA2	0x85	0x00	0x00	0x00	0x1F	0x00	0x00	0x00	0x7F	0x00	0x00	0x00	
		使能其他画圈手势	0x10	0x0	0x0	0xA2	0x85	0x00	0x00	0x00	0x60	0x00	0x00	0x00	0x60	0x00	0x00	0x00	
	校准操作模式（0x0080）	使能	0x10	0x0	0x0	0xA2	0x80	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x3F	0x00	0x00	0x00	固定命令。
		禁止	0x10	0x0	0x0	0xA2	0x80	0x00	0x00	0x00	0x3F	0x00	0x00	0x00	0x3F	0x00	0x00	0x00	固定命令。
	数据输出使能屏蔽（0x00A0）	使能所有数据	0x10	0x0	0x0	0xA2	0xA0	0x00	0x00	0x00	0x3F	0x18	0x00	0x00	0x3F	0x18	0x00	0x00	Argument 0 定义哪个数据要被使能或禁止。 Argument 1 定义需要配置的数据的屏蔽。给出的值，仅为一些示例。
		使能 DSP、手势和噪声功率	0x10	0x0	0x0	0xA2	0xA0	0x00	0x00	0x00	0x23	0x00	0x00	0x00	0x3F	0x18	0x00	0x00	
		仅使能数据：噪声 （其他不变）	0x10	0x0	0x0	0xA2	0xA0	0x00	0x00	0x00	0x10	0x00	0x00	0x00	0x10	0x00	0x00	0x00	
		仅禁止数据：CIC （其他不变）	0x10	0x0	0x0	0xA2	0xA0	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x08	0x00	0x00	
	数据输出锁定屏蔽（0x00A1）	锁定所有数据	0x10	0x0	0x0	0xA2	0xA1	0x00	0x00	0x00	0x3F	0x18	0x00	0x00	0x3F	0x18	0x00	0x00	Argument 0 定义需要锁定或解锁哪些数据。 Argument 1 定义要配置的数据的掩码。给出的值，仅为一些示例。
		锁定 DSP、手势和噪声功率	0x10	0x0	0x0	0xA2	0xA1	0x00	0x00	0x00	0x23	0x00	0x00	0x00	0x3F	0x18	0x00	0x00	
		仅锁定数据：噪声 （其他不变）	0x10	0x0	0x0	0xA2	0xA1	0x00	0x00	0x00	0x10	0x00	0x00	0x00	0x10	0x00	0x00	0x00	
		仅解锁数据：CIC （其他不变）	0x10	0x0	0x0	0xA2	0xA1	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x08	0x00	0x00	
	数据输出请求屏蔽（0x00A2）	请求所有数据	0x10	0x0	0x0	0xA2	0xA2	0x00	0x00	0x00	0x3F	0x18	0x00	0x00	0x3F	0x18	0x00	0x00	Argument 0 定义需要请求哪些数据。仅对下一条消息有效。 Argument 1 定义需要配置的数据的掩码。给出的值，仅为一些示例。
		请求 DSP、手势和噪声功率	0x10	0x0	0x0	0xA2	0xA2	0x00	0x00	0x00	0x23	0x00	0x00	0x00	0x3F	0x18	0x00	0x00	
		仅请求数据：噪声	0x10	0x0	0x0	0xA2	0xA2	0x00	0x00	0x00	0x10	0x00	0x00	0x00	0x10	0x00	0x00	0x00	
	手势进行中标志控制（0x00A3）	使能	0x10	0x0	0x0	0xA2	0xA3	0x00	0x00	0x00	0x01	0x00	0x00	0x00	0x01	0x00	0x00	0x00	固定命令。
		禁止	0x10	0x0	0x0	0xA2	0xA3	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x01	0x00	0x00	0x00	固定命令。

© 2014 Microchip Technology Inc.

表 A-3： SENSOR_DATA_OUTPUT 命令示例

请求的函数	用户操作	Sensor_Data_Output												备注
		报文头				有效内容								
		消息长度	标志	序列	ID	数据输出配置掩码		时间戳	系统信息	参数				
数据输出仅包含 DSPStatus 字段（通过 Set_Runtime_Parameter 命令：10 00 00 A2 A0 00 00 00 01 00 00 00 FF FF FF FF 配置）	无操作	0x0A	0x08	0x26	0x91	0x01	0x01	0x5D	0x80	0x10	0x73	—	—	负校准。
		0x0A	0x08	0x27	0x91	0x01	0x01	0x5E	0x80	0x00	0x73	—	—	校准结束。
		0x0A	0x08	0x28	0x91	0x01	0x01	0x5D	0x80	0x20	0x73	—	—	空闲校准。
		0x0A	0x08	0x29	0x91	0x01	0x01	0x5E	0x80	0x00	0x73	—	—	校准结束。
数据输出仅包含手势数据字段（通过 Set_Runtime_Parameter 命令：10 00 00 A2 A0 00 00 00 02 00 00 00 FF FF FF FF 配置）	从东向西滑动	0x0C	0x08	0x31	0x91	0x02	0x01	0x82	0x80	0x03	0x10	0x00	0x00	0x03：从东向西滑动 0x10：滑动手势
		0x0C	0x08	0x32	0x91	0x02	0x01	0x83	0x80	0x00	0x00	0x00	0x00	
	从北向南滑动	0x0C	0x08	0x33	0x91	0x02	0x01	0x13	0x80	0x05	0x10	0x04	0x00	0x05：从北向南滑动 0x10：滑动手势
		0x0C	0x08	0x34	0x91	0x02	0x01	0x14	0x80	0x00	0x00	0x00	0x00	
	从南向北滑动	0x0C	0x08	0x35	0x91	0x02	0x01	0x53	0x80	0x04	0x10	0x04	0x00	0x03：从南向北滑动 0x10：滑动手势
		0x0C	0x08	0x36	0x91	0x02	0x01	0x54	0x80	0x00	0x00	0x00	0x00	
	从西向东滑动	0x0C	0x08	0x37	0x91	0x02	0x01	0x5D	0x80	0x02	0x10	0x00	0x00	0x03：从西向东滑动 0x10：滑动手势
		0x0C	0x08	0x38	0x91	0x02	0x01	0x5E	0x80	0x00	0x00	0x00	0x00	

© 2014 Microchip Technology Inc.

表 A-3： SENSOR_DATA_OUTPUT 命令示例 （续）

请求的函数	用户操作	Sensor_Data_Output												备注
		报文头				有效内容								
		消息长度	标志	序列	ID	数据输出配置掩码		时间戳	系统信息	参数				
数据输出仅包含手势数据字段 （通过 Set_Runtime_Parameter 命令：10 00 00 A2 A0 00 00 00 02 00 00 00 FF FF FF FF 配置） 通过 Set_Runtime_Parameter 命令：10 00 00 A2 A3 00 00 00 01 00 00 00 FF FF FF FF 激活手势进行中	从西向东滑动	0x0C	0x08	0x3A	0x91	0x02	0x01	0x19	0x81	0x00	0x00	0x00	0x80	启动手势识别器
		0x0C	0x08	0x3B	0x91	0x02	0x01	0x45	0x81	0x03	0x10	0x00	0x00	识别到的手势 （从东向西滑动）
		0x0C	0x08	0x3C	0x91	0x02	0x01	0x46	0x81	0x00	0x00	0x00	0x00	
	仅移动手	0x0C	0x08	0x3D	0x91	0x02	0x01	0x47	0x81	0x00	0x00	0x00	0x80	启动手势识别器
		0x0C	0x08	0x3E	0x91	0x02	0x01	0x6E	0x81	0x01	0x00	0x00	0x00	识别到的无用数据
		0x0C	0x08	0x3F	0x91	0x02	0x01	0x6F	0x81	0x00	0x00	0x00	0x00	
	从东向西滑动	0x0C	0x08	0x40	0x91	0x02	0x01	0x83	0x81	0x00	0x00	0x00	0x80	启动手势识别器
		0x0C	0x08	0x41	0x91	0x02	0x01	0xAC	0x80	0x03	0x10	0x04	0x00	识别到的手势 （从东向西滑动）
		0x0C	0x08	0x42	0x91	0x02	0x01	0xAD	0x80	0x00	0x00	0x00	0x00	
	从北向南滑动	0x0C	0x08	0x43	0x91	0x02	0x01	0x67	0x81	0x00	0x00	0x00	0x80	启动手势识别器
		0x0C	0x08	0x44	0x91	0x02	0x01	0x8A	0x80	0x05	0x10	0x04	0x00	识别到的手势 （从北向南滑动）
		0x0C	0x08	0x45	0x91	0x02	0x01	0x8B	0x80	0x00	0x00	0x00	0x00	
	从南向北滑动	0x0C	0x08	0x46	0x91	0x02	0x01	0x67	0x81	0x00	0x00	0x00	0x80	启动手势识别器
		0x0C	0x08	0x47	0x91	0x02	0x01	0x8E	0x80	0x04	0x10	0x04	0x00	识别到的手势 （从南向北滑动）
		0x0C	0x08	0x48	0x91	0x02	0x01	0x8F	0x80	0x00	0x00	0x00	0x00	
	从西向东滑动	0x0C	0x08	0x49	0x91	0x02	0x01	0x6E	0x81	0x00	0x00	0x00	0x80	启动手势识别器
		0x0C	0x08	0x4A	0x91	0x02	0x01	0x9A	0x80	0x02	0x10	0x02	0x00	识别到的手势 （从西向东滑动）
		0x0C	0x08	0x4B	0x91	0x02	0x01	0x9B	0x80	0x00	0x00	0x00	0x00	
	顺时针画圈	0x0C	0x08	0x4C	0x91	0x02	0x01	0x81	0x80	0x00	0x00	0x00	0x80	启动手势识别器
		0x0C	0x08	0x4D	0x91	0x02	0x01	0xD6	0x80	0x00	0x00	0x00	0x00	画圈手势无法识别，因为使能了 AirWheel
	逆时针画圈	0x0C	0x08	0x4E	0x91	0x02	0x01	0x05	0x80	0x00	0x00	0x00	0x80	启动手势识别器
		0x0C	0x08	0x4F	0x91	0x02	0x01	0x56	0x80	0x00	0x00	0x00	0x00	画圈手势无法识别，因为使能了 AirWheel

© 2014 Microchip Technology Inc.

表 A-3： SENSOR_DATA_OUTPUT 命令示例 （续）

请求的函数	用户操作	Sensor_Data_Output												备注
		报文头				有效内容								
		消息长度	标志	序列	ID	数据输出配置掩码		时间戳	系统信息	参数				
数据输出仅包含触摸数据字段 （通过 Set_Runtime_Parameter 命令：10 00 00 A2 A0 00 00 00 04 00 00 00 FF FF FF FF 配置）	触摸中心电极	0x0C	0x08	0x45	0x91	0x04	0x01	0x51	0x81	0x10	0x00	0x09	0x00	检测到中心触摸，触摸计数器 = 0x09
		0x0C	0x08	0x46	0x91	0x04	0x01	0x52	0x81	0x10	0x00	0x00	0x00	触摸计数器复位
		0x0C	0x08	0x47	0x91	0x04	0x01	0x5D	0x81	0x00	0x02	0x00	0x00	检测到中心电极被单击
		0x0C	0x08	0x48	0x91	0x04	0x01	0x5E	0x81	0x00	0x00	0x00	0x00	

© 2014 Microchip Technology Inc.

附录 B 术语表

表 B-1： 术语表

术语	定义
AFE	模拟前端
Aurea	带图形用户界面的 MGC3130 PC 控制软件
Colibri Suite	GestIC® 库内的嵌入式 DSP 套件
电场 （E-field）	电场
垫片砖 （Spacer Brick）	传感器层和砖结构模拟手之间的垫片 （聚苯乙烯块，大小为 40x40xh mm，其中 h = 1/2/3/5/8/12 cm）
感应空间 （Sensing Space）	感应区域上方的空间
感应区域 （Sensing Area）	四个框架电极围成的区域
GestIC® 技术 （GestIC® Technology）	Microchip 的专利技术，运用电近场传感原理，提供 3D 自由空间手势识别
GestIC® 库 （GestIC® Library）	用于实现 MGC3130 的特性，并预编程到 MGC3130 中，以二进制文件形式交付
Hillstar	MGC3130 开发工具包
HMM	隐马尔可夫模型
接近检测 （Approach Detection）	GestIC® 技术特性：MGC3130 的带接近检测功能的节能模式
框架电极 （Frame Electrodes）	4 个电极的矩形组合，用于电场传感
MGC3130	单区 3D 手势检测控制器
Sabrewing	MGC3130 评估板
深度休眠 （Deep Sleep）	MGC3130 节能模式
手势集 （Gesture Set）	一组预定的手移动模式
手势识别 （Gesture Recognition）	Microchip 的随机 HMM 分类，可自动检测并分类手的移动模式
SPU	信号处理单元
位置跟踪 （Position Tracking）	GestIC® 技术特性
信号偏差 （Signal Deviation）	针对手接近时和手未接近时的传感器信号变化量的术语
应用主机 （Application Host）	PC 或嵌入式控制器，用于控制 MGC3130
自唤醒 （Self Wake-up）	MGC3130 节能模式
砖结构模拟手 （Hand Brick）	覆盖铜箔的测试块 （40x40x70 mm）

© 2014 Microchip Technology Inc.

全球销售及服务网点

美洲

公司总部 Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277
技术支持：
http://www.microchip.com/support
网址：www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

奥斯汀 Austin, TX
Tel: 1-512-257-3370

波士顿 Boston
Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 Chicago
Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

克里夫兰 Cleveland
Independence, OH
Tel: 1-216-447-0464
Fax: 1-216-447-0643

达拉斯 Dallas
Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 Detroit
Novi, MI
Tel: 1-248-848-4000

休斯敦 Houston, TX
Tel: 1-281-894-5983

印第安纳波利斯 Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

纽约 New York, NY
Tel: 1-631-435-6000

圣何塞 San Jose, CA
Tel: 1-408-735-9110

加拿大多伦多 Toronto
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

亚太总部 Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

中国 - 北京
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

中国 - 成都
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 - 重庆
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 - 杭州
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

中国 - 香港特别行政区
Tel: 852-2943-5100
Fax: 852-2401-3431

中国 - 南京
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 - 青岛
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 上海
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 - 沈阳
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

中国 - 武汉
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 - 厦门
Tel: 86-592-238-8138
Fax: 86-592-238-8130

中国 - 珠海
Tel: 86-756-321-0040
Fax: 86-756-321-0049

亚太地区

台湾地区 - 高雄
Tel: 886-7-213-7830

台湾地区 - 台北
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

台湾地区 - 新竹
Tel: 886-3-5778-366
Fax: 886-3-5770-955

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

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

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

印度 India - Pune
Tel: 91-20-3019-1500

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

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

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

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

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

马来西亚 Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

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

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

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

欧洲

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

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

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

德国 Germany - Dusseldorf
Tel: 49-2129-3766400

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

德国 Germany - Pforzheim
Tel: 49-7231-424750

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

意大利 Italy - Venice
Tel: 39-049-7625286

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

波兰 Poland - Warsaw
Tel: 48-22-3325737

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

瑞典 Sweden - Stockholm
Tel: 46-8-5090-4654

英国 UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

03/25/14

© 2014 Microchip Technology Inc.