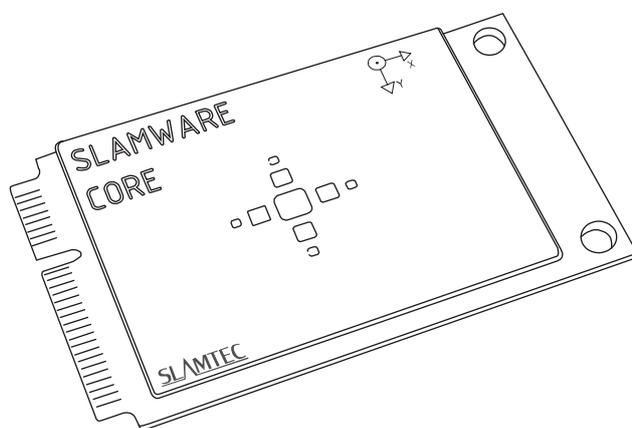


SLAMWARE

模块化自主定位导航解决方案

通讯接口协议手册

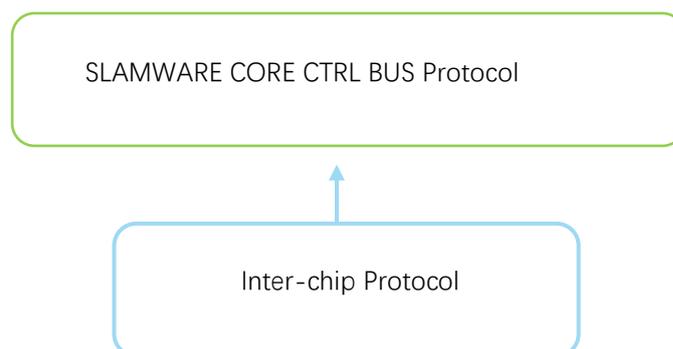


目录	1
简介	3
功能概览	4
内部模块框图和使用方法.....	4
INTER-CHIP 通讯协议规范	5
INTER-CHIP 通讯协议简介.....	5
INTER-CHIP 通讯协议规范.....	6
STANDARD PROFILE.....	9
SLAMWARE CORE 协议命令	13
SLAMWARE CORE CTRL BUS 通讯协议.....	13
SLAMWARE CORE 请求命令.....	14
外部系统连接请求(CONNECT_BASE).....	15
外部系统参数获取请求(GET_BASE_CONF).....	16
外部系统配置获取请求(GET_BINARY_CONF).....	18
外部系统状态获取请求(GET_BASE_STATUS).....	18
外部系统轮组状态获取请求(GET_BASE_MOTOR_DATA).....	19
外部系统距离传感器数据获取请求(GET_BASE_SENSOR_DATA).....	20
外部系统触碰传感器数据获取请求(GET_BASE BUMPER_DATA).....	21
外部系统自动回充数据获取(GET_AUTO_HOME_DATA).....	22
外部系统运动控制请求(SET_BASE_MOTOR).....	23
外部系统设置速度及获取 DEADRECKON 数据请求(SET_V_AND_GET_DEADRECKON).....	23
外部系统命令查询请求(POLL_BASE_CMD).....	26
外部系统命令回应请求(POLL_BASE_ANS_CMD).....	27
SLAMWARE CORE 系统事件通知命令(SEND_EVENT).....	27
外部系统健康管理命令请求(HEALTH_MGMT).....	28
外部系统辅助定位数据获取请求 (GET_AUXILIARY_ANCHOR).....	31
修订历史	33
附录	34
图表索引.....	34

SLAMTEC SLAMWARE CORE 模块与外部系统通过 CTRL BUS 低速总线连接。通过 CTRL BUS 总线协议，SLAMWARE CORE 可以获取外部系统的状态参数，例如外部系统的电池电量信息，是否在充电状态，外部系统的电机转速，传感器状态等信息。同时，SLAMWARE CORE 也通过 CTRL BUS 总线协议控制外部系统的电机运动，从而实现完整的室内定位导航功能。

CTRL BUS 总线协议是运行在串口上的，波特率为 115200bps，外部系统必须实现此总线协议才可以使 SLAMWARE CORE 正常工作。

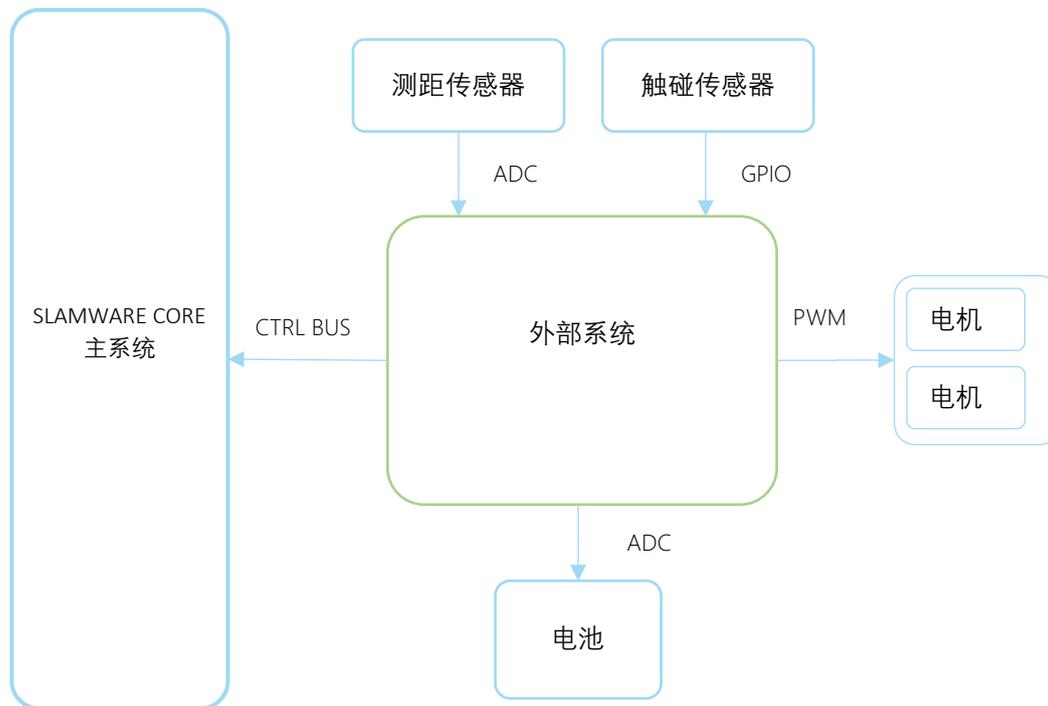
SLAMWARE CORE CTRL BUS 通讯协议使用的是 Inter-chip 通讯协议的扩展模式。Inter-chip 通讯协议是 SLAMTEC 专门为底层电子系统，诸如微控制器之间、微控制器与宿主系统、微控制器与设备之间相互通讯而设计的协议。它以数据包为单元在通讯媒介上传递数据，通讯包中包含了数据校验机制以及帧同步机制，可以实现可靠的通讯链路。目前该协议广泛用于 SLAMTEC 的机器人系统、传感器和模块设备中，并可在 USART、USB、I²C、TCP 以太网中使用。



图表 1-1 SLAMWARE CORE CTRL BUS Protocol 和 Inter-chip Protocol 关系示意图

内部模块框图和使用方法

下图描述了 SLAMWARE CORE 和外部系统之间的通信连接框图。



图表 2-1 SLAMWARE CORE 内部模块框图

Inter-chip 通讯协议简介

通讯特性

本协议所假设的通讯介质是以点对点或者总线方式组织的网络，其中同一时间最多只有 2 个通讯端点(Peer)可以相互之间通讯，但是如果通讯介质网络允许实现广播，则可以允许由一个通讯端点同时向多个其他端点单向的发送消息。

通讯介质本身可以是基于字节流模式的信道，比如 USART、TCP 以太网，也可以是基于数据包的信道，比如 USB、I2C。Inter-chip Protocol 并不要求通讯介质实现数据包的同步机制，协议本身可以利用每个数据包头部的同步信号自动实现同步。

当通讯发生时，传输在通讯介质上的消息永远是以具有固定格式的数据包形式存在的。每个数据包允许承载不同长度的数据，但在数据包头部均具有长度一致、数据结构定义一致的数据，该数据用语描述数据包自身的信息以及期望接收端采取的处理方式。

数据传输顺序

Inter-chip Protocol 要求信道传输数据时遵循 FIFO 原则，并且要求使用小字端传输数据包中协议所定义的公共数据结构，如包头部的描述数据。

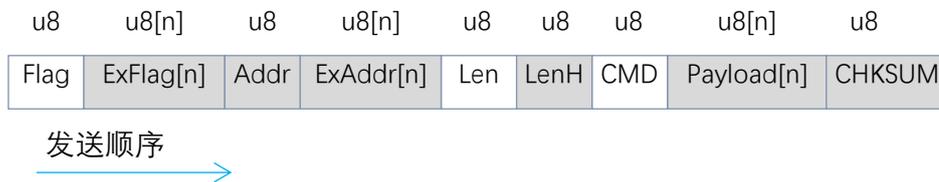
协议子模式

基于不同应用场合的需求，Inter-chip Protocol 包含了几个具体的子模式(sub-profile)。不同的子模式定义了各自的网络通讯方式、具体采用的包头数据结构等信息。在当前的实现中，Standard Profile 是协议默认使用的子模式。它定义了一种采用请求-应答机制的通讯网络。

Inter-chip 通讯协议规范

数据包结构纵览

每个 Inter-chip Protocol 规范的数据包均符合如下的格式定义。



图表 3-1 Inter-chip Protocol 数据包格式定义

其中，底色为灰色的部分为可选字段。不同的 Sub-Profile 会使用这些可选字段的某些部分。白色的字段是所有 Inter-chip Protocol 规范的数据包均会带有的。后文将就单个字段进行描述。

Flag 字段

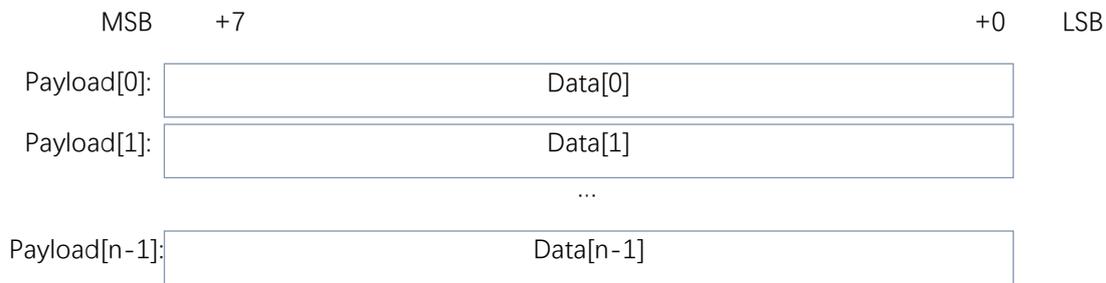


图表 3-2 Inter-chip Protocol Flag 字段格式说明

该字段为一个数据包的起始标识，占用一个字节。随着 Sub-Profile 的不同，其取值有着不同的定义。其每个标志位的具体解释如下：

字段名	描述
ExtBit	设置为 1 时表示数据包的下一个字节是 ExFlag 字段，即 ExFlag[n]将出现在本数据包中。其中将携带更多的标志位信息。
LongFrame	设置为 1 时表示本数据包将使用 16bit 的长度字段，此时 LenH 字段将包含在本数据包中。该数据包可以装载最多 65534 字节的数据。
AddrEn	设置为 1 时表示 Addr 将包含在数据包中，并且 ExAddr[n]也可能存在于本数据包。Inter-chip Protocol 协议栈将可能解析存在于 Addr 以及 ExAddr[1..n]中的地址进行相应处理。
CheckSumEn	设置为 1 时，本数据包将在最末尾包含一个校验和字节数据。即 CHKSUM 字段将包含在本数据包中。Inter-chip Protocol 协议栈将被要求对本数据包的数据进行校验并和 CHKSUM 做对比。

图表 3-3 Inter-chip Protocol Flag 字段标志位说明



图表 3-6 Inter-chip Protocol Payload 字段格式说明

该字段记录了数据包发送方希望携带的有效信息负载。其数据长度由 Len/LenH 字段表示。如果发送方没有任何需要传输的信息时，该字段将被省略。

CHKSUM 字段



图表 3-7 Inter-chip Protocol CHKSUM 字段格式说明

当 Flag 中 CheckSumEn 位被设置为 1 时，该字段将附加在数据包的末尾，用于记录对数据包中正确数据的校验和。其计算方式如下：

$$\text{CHKSUM}[0..7] = 0 \text{ xor Packet}[0] \text{ xor Packet}[1] \text{ xor ... xor Packet}[N]$$

其中，Packet 表示即将发送的当前数据包去除 CHKSUM 字段的其他所有部分。Packet[N] 表示即将发送的当前数据包的第 N 个字节。

当 CHKSUM 字段启用后，接收方协议栈将会在完成一个数据包接收后，重新计算一个新的 CHKSUM 并和数据包中附带的原始 CHKSUM 字段进行对比。如果不匹配，协议栈将丢弃当前数据包，并按照 Sub-Profile 所定义的行为，进行额外的处理。

ExFlag[N] 字段



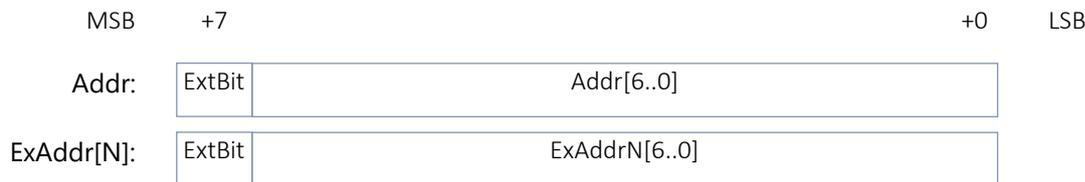
图表 3-8 Inter-chip Protocol ExFlag[N] 字段格式说明

当 Flag 中 ExtBit 位被设置为 1 时，数据包中将包含一个或多个字节的 ExFlag 字段。该字段包含了额外的标志位用于描述数据包本身的信息或者 Sub-Profile 所

额外定义的内容。同时，用户应用程序也可以将额外的信息附加在本字段中。

每一个字节 ExFlag 中的 ExtBit 位用于表示是否还存在一个额外的 ExFlag 字节紧跟在本 ExFlag 字节之后。当设置为 0 时，表示当前的 ExFlag 字节即是整个 ExFlag[N] 字段的终结。请参考具体 Sub-Profile 的定义了解本字段具体位定义。

Addr 与 ExAddr[N]字段



图表 3-9 Inter-chip Protocol Addr 和 ExAddr[N]字段格式说明

当 Flag 中 AddrEn 位被设置为 1 时，数据包中将包含一个 Addr 字段，用于记录本数据包需要发送的目标设备地址。不同的 Sub-Profile 实现下，协议栈将对该字段的数据作出不同的处理。

在 Addr 字段中，使用了 7 个 bit 的区域用于表示 0-127 个设备地址号。如果希望使用更长的设备地址，则可以将 Addr 字段的 ExtBit 设置为 1，此时，更多的地址数据位将被使用。它们将存储在后续的 ExAddr[N]字段当中。

当 Addr 以及每个 ExAddr[N]字段字节中的 ExtBit 设置为 1 时，表示当前字节后将紧跟一个 ExAddr 字节，直到 ExtBit 为 0 时为止。

Standard Profile

Inter-chip Protocol 的 Standard Profile 定义了一种基于应答请求通讯模式的可靠点对点包交换网络。并且预留了一定的 CMD 取值用于协议栈内部实现通讯机制。

通讯模式和术语

Standard Profile 要求网络中同一时刻只有 2 个通讯端点相互进行通讯。并且，其中一方将始终首先向另一方发送数据。另一方将始终在接收到发送数据方的数据包后做出回应，发送对应的回应数据。

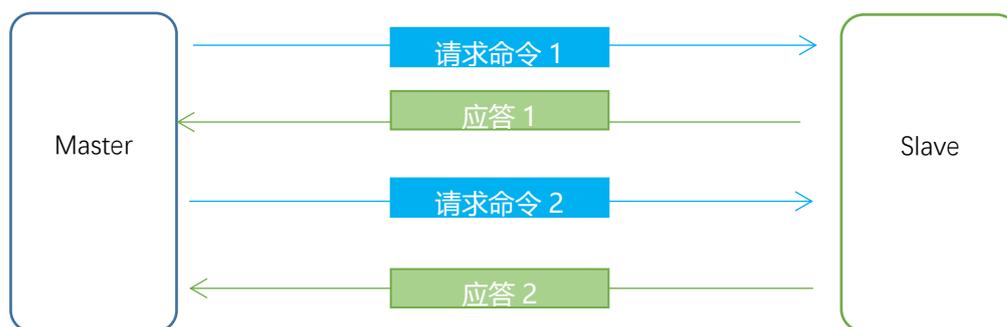
这里定义上述始终首先发送数据的通讯端点为**宿主(Master)**，而始终先接收数据再发出回应数据包的数据端点为**从机(Slave)**。

Standard Profile 使用了相同的格式用于 Master 以及 Slave 发出的数据包，即 Flag

字段的定义一致。不过根据通讯角色的不同，其发送的数据包的某些字段具有不同的含义。

对于由 Master 发出的数据包被称为**请求命令数据包（简称请求命令）**，而由 Slave 发出的数据包被称为**应答数据包（简称应答）**。

请求命令与应答必须是成对出现的，每个应答总对应与一个请求命令。而 Master 必须在发送一个请求命令后，在接收到 Slave 发送应答数据包或者通讯超时后，方可进行下一个请求命令的发送。



图表 3-10 基于 Inter-chip Protocol 的 Master 和 Slave 通讯规则

Flag 字段定义

	MSB		+7						+0	LSB
Standard:	0	0	0	1	0	0	0	0	0	
Long Frame:	0	1	0	1	0	0	0	0	0	

图表 3-11 Inter-chip Protocol Flag 字段格式说明

Standard Profile 允许使用如下 2 种 Flag 字段的数据包格式：

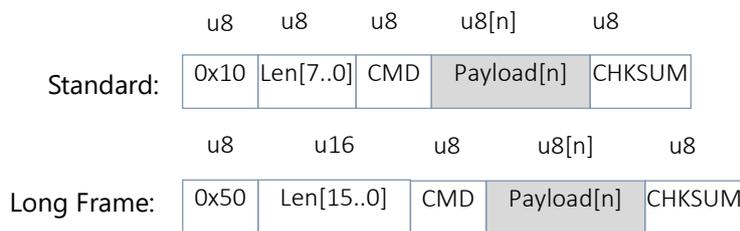
包类型	Flag 取值	描述
Standard	0x10	Payload 最大 254 字节长度，无地址信息，带有 CHKSUM
Long Frame	0x50	Payload 最大 65534 字节长度，无地址信息，带有 CHKSUM

图表 3-12 Inter-chip Protocol Flag 数据包格式说明

采用 Standard Profile 的协议栈必须支持 Standard 类型的数据包。而当协议栈支持 Long Frame 模式的数据包时，Master 和 Slave 在通讯过程中均可以随意的按

照实际需要选择 Standard 或者 Long Frame 模式的数据包进行发送。

请求命令数据包



图表 3-13 Inter-chip Protocol 请求命令数据包格式说明

由 Master 发送的请求命令数据包采用如上 2 种固定的数据包结构。其中，CMD 字段的大部分取值可由应用程序使用。用于对 Slave 具体功能的调用或者数据请求。其具体取值请参考对应的设备文档。

符合 Standard Profile 规范的协议栈将保留如下的 CMD 字段取值，用于完成协议栈本身的处理任务：

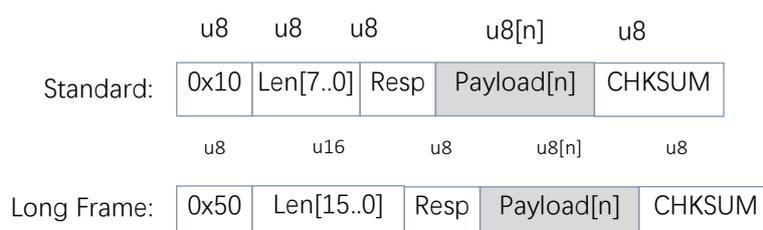
保留的 CMD 取值	描述
0x00	强制同步帧。用于实现通讯信道同步
0x01	ECHO 帧，用于测试 Slave 通讯状况
0x02 – 0x0F	保留

图表 3-14 Inter-chip Protocol 保留的 CMD 说明

应用程序需要避免使用上述保留的 CMD 字段取值。

Standard Profile 规范对 Payload[n]的使用不做出假设。但具体的 Slave 设备会要求在不同的 CMD 字段取值情况下，对 Payload[n]字段的数据格式做出假设。

应答数据包



图表 3-15 Inter-chip Protocol 应答数据包格式说明

由 Slave 发送的应答数据包满足上述 2 种格式。其中，CMD 字段被用于对 Slave 先前收到的请求命令处理情况的回应。其取值必须为如下的几种：

保留的 Resp(CMD)	描述
0x00	强制同步帧的应答包
0x01	ECHO 帧的应答包
0x02	<OK> 表示上一个请求命令被正确执行或者从协议栈角度没有遇到问题。
0x03	<Error> 表示上一个请求命令执行过程出现问题
0xFF	<Invalid> 表示上一个请求命令接收的不完整、或者数据包本身结构不正确

图表 3-16 Inter-chip Protocol 保留的 Resp 说明

Payload[n]字段在应答数据包中用于存放 Slave 对先前 Master 发来的请求命令的执行结果。当本次应答的 Resp 为<OK>时，Payload[N]包含的具体数据定义与相关的请求命令有关。请参考具体的 Slave 设备手册了解详情。

当 Resp 为<Error> (0x03)或者<Invalid>(0xFF)时，Payload[N]将存放一个 16bit 的错误码，描述出错情况。Standard Profile 规范要求协议栈保留如下的错误码取值，请含义见下表。Slave 上的应用程序也可以定义更多的错误代码。

错误码	描述
0x40	请求命令包校验和不匹配
0x20	请求命令包长度超过了当前 Slave 协议栈缓冲区承受范围
0x10	Slave 协议栈信道没有完成同步
0x8000	请求命令包中的 CMD 字段不被 Slave 支持
0x8001	请求命令在执行中遇到格式类错误，例如请求参数不符合规范
0x8002	请求命令在执行中遇到操作类错误，例如执行过程遇到问题而无法继续

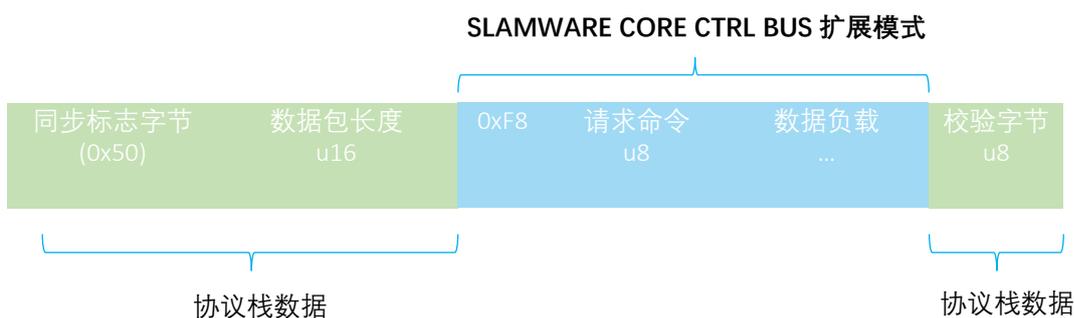
图表 3-17 Inter-chip Protocol 错误码说明

SLAMWARE CORE CTRL BUS 通讯协议

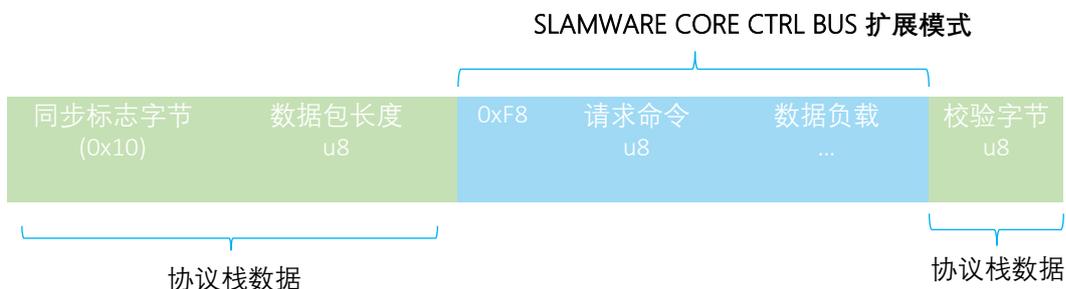
SLAMWARE CORE CTRL BUS 通讯协议使用 Inter-chip 通讯协议的扩展模式。

它的数据包类型域为标识 SLAMWARE CORE CTRL BUS 扩展模式的 0xF8。在此扩展模式下，数据负载的格式为请求/应答通讯模式，应答数据包格式与标准 Inter-chip 通讯协议一致。根据数据包长度的类型分为如下两种格式：

格式一：

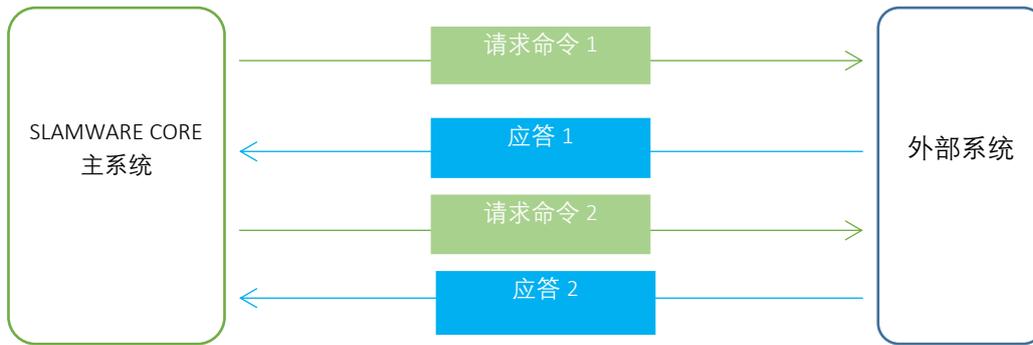


格式二：



图表 4-1 SLAMWARE CORE CTRL BUS 通讯协议数据包格式

SLAMWARE CORE 和外部系统之间的 CTRL BUS 通讯协议设计为单向通讯模式，始终由 SLAMWARE CORE 发起数据包请求。



图表 4-2 SLAMWARE CORE CTRL BUS 请求应答通讯模式说明

外部系统需要及时回应 SLAMWARE CORE 的请求命令，否则 SLAMWARE CORE 将认为外部系统连线中断，可能停止正常工作。

请求命令与应答必须是成对出现的，每个应答总对应于一个请求命令。SLAMWARE CORE 在发送一个请求命令后，总会在接收到外部系统发送应答数据包或者通讯超时后，才会进行下一个请求命令的发送。

SLAMWARE CORE 也可以接受外部系统的控制命令，但由于单向通讯的模式，外部系统只能在 SLAMWARE CORE 定期发送外部系统控制命令查询请求的时候才可以发送控制命令。但外部系统可以通过#CMD 信号通知 SLAMWARE CORE 尽快发送外部系统控制命令查询请求。

SLAMWARE CORE 请求命令

SLAMWARE CORE 室内定位导航模块会发送以下请求命令，外部系统可以根据需要响应请求命令，否则 SLAMWARE CORE CTRL BUS 协议栈会做默认响应。标*的命令为外部系统必须响应的请求命令。否则 SLAMWARE CORE 将不能正常工作。

请求命令	命令代码	描述
外部系统连接请求 (CONNECT_BASE) *	0x10	请求连接外部系统，外部系统将会返回硬件、固件版本信息表示连接成功。
外部系统参数获取请求 (GET_BASE_CONF) *	0x20	请求外部系统的配置参数，包括尺寸类型，尺寸半径，轮组类型，传感器位置角度等。
外部系统配置获取请求 (GET_BINARY_CONF)	0x21	请求外部系统配置参数。该指令用于替代上述的 0x20 指令，具有更大的可扩展性。SLAMWARE CORE 会首先尝

		试该指令，如果该指令返回了 Not Support，则再调用 0x20 获取外部系统配置。
外部系统状态获取请求 (GET_BASE_STATUS) *	0x30	请求获取外部系统的状态，包括电池电量，充电状态等。
外部系统轮组状态获取请求 (GET_BASE_MOTOR_DATA) *	0x31	请求外部系统返回轮组运行距离累计值。
外部系统距离传感器数据获取请求 (GET_BASE_SENSOR_DATA) *	0x32	请求外部系统返回距离传感器的数据。
外部系统触碰传感器数据获取请求 (GET_BASE BUMPER_DATA) *	0x33	请求外部系统返回触碰传感器的数据。
外部系统自动回充数据获取请求 (GET_AUTO_HOME_DATA) *	0x34	请求外部系统返回自动回充的数据。
外部系统辅助定位数据获取请求 (GET_AUXILIARY_ANCHOR)	0x35	请求辅助定位传感器数据
外部系统运动控制请求 (SET_BASE_MOTOR) *	0x40	请求外部系统控制轮组运动。
外部系统设置速度及获取 Deadreckon 请求 SET_V_AND_GET_DEADRECKON *	0x41	请求外部系统设置速度变量并获取上一个周期的 Deadreckon 数据。
外部系统命令查询请求 POLL_BASE_CMD *	0x50	查询外部系统是否有命令发送。
外部系统命令回应请求 POLL_BASE_ANS_CMD *	0x5F	回应外部系统命令请求
SLAMWARE CORE 系统事件通知命令 (SEND_EVENT) *	0x60	通知外部系统 SLAMWARE CORE 事件
外部系统健康管理命令请求 (HEALTH_MGMT) *	0x90	获取外部系统健康状况

图表 4-3 SLAMWARE CORE 请求命令列表

外部系统连接请求(CONNECT_BASE)

SLAMWARE CORE 室内定位导航模块在准备好后会一直发送外部系统连接请求确认外部系统已正常工作。直到外部系统响应此请求后 SLAMWARE CORE 才会开始正常工作。

- 请求数据包：

命令代码	负载数据
0x10	协议版本
	u8

- 应答数据包：

应答代码	负载数据			
<OK>	型号	固件版本	硬件版本	序列号
	u8[12]	u16	u16	u32[3]

数据域	类型	描述
协议版本	u8	协议的版本信息，版本信息不匹配时外部系统应应答错误
型号	u8[12]	外部系统的型号，最多 12 个字符
固件版本	u16	外部系统的固件版本
硬件版本	u16	外部系统的硬件版本
序列号	u32[3]	外部系统的序列号，最多 12 个字节

图表 4-4 外部系统连接请求与应答数据域描述

外部系统参数获取请求(GET_BASE_CONF)

SLAMWARE CORE 室内定位导航模块在与外部系统连接后会发送参数获取请求以获得外部系统的基本结构信息，包括尺寸类型，尺寸半径，轮组类型以及传感器安装位置信息等。目前版本 SLAMWARE CORE 只支持圆形和方形尺寸以及两轮差动轮组。

- 请求数据包：

命令代码
0x20

- 应答数据包：

应答代码	负载数据		
<OK>	尺寸类别	尺寸半径	轮组类别
	u8	u32	u8
	距离传感器数量	距离传感器安装位置	
	u8	pos[8]	
	触碰传感器数量	触碰传感器安装位置	
	u8	pos[8]	

数据域	类型	描述
尺寸类型	u8	外部系统的尺寸类型，目前只支持圆形和方形
尺寸半径	u32	外部系统的尺寸半径，以 mm 为单位,Q8 定点小数
轮组类型	u8	外部系统的轮组类型，目前只支持两轮差动轮组
距离传感器数量	u8	外部系统的距离传感器数量，最多支持 8 组
距离传感器安装位置	pos[8]	外部系统的距离传感器安装位置
触碰传感器数量	u8	外部系统的触碰传感器数量，最多支持 8 组
触碰传感器安装位置	pos[8]	外部系统的触碰传感器安装数据

图表 4-5 外部系统参数获取请求的应答数据域描述

注：当使用 0x20 底盘配置指令时，最多可支持 8 组距离传感器；当使用 0x21 Binary Config 指令配置时，最多可支持 16 组距离传感器

目前支持的尺寸类型有：

类型编号	描述
0x00	圆形
0x01	方形

图表 4-6 外部系统所支持的外部系统尺寸类型

目前支持的轮组类型有：

类型编号	描述
0x00	两轮差动

图表 4-7 外部系统所支持的轮组类型

传感器安装位置的 pos 数据表示为：

数据域	类型	描述
X 轴向距离	s32	传感器中心距离外部系统中心的 X 轴向距离，Q8 定点小数
Y 轴向距离	s32	传感器中心距离外部系统中心的 Y 轴向距离，Q8 定点小数
Z 轴向距离	s32	传感器中心距离外部系统中心的 Z 轴向距离，Q8 定点小数
角度	u32	传感器中心跟外部系统中心点的逆时针夹角，Q8 定点小数

图表 4-8 外部系统传感器安装位置 pos 数据

外部系统配置获取请求(GET_BINARY_CONF)

这是上述 0x20 指令的升级指令，SLAMWARE CORE 模块会优先使用该指令获取外部系统的配置信息，如果外部系统返回 Not Support 的话才会选择去调用 0x20 指令。

这个指令相对于 0x20 指令来说提供了更多的功能，如雷达安装位置的配置等。

- 请求数据包：

命令代码

0x21

- 应答数据包：

应答代码

<OK>

负载数据

外部系统配置数据

u8*n

外部系统的配置数据由 SDK 中所附的 Slamware Configuration Tool 自动生成。只需将生成的.c 文件加入到工程项目中，将其中的 slamware_config 变量作为外部系统配置数据字段数据即可。

注：当使用 0x21 Binary Config 指令配置时，请将 0x20 底盘配置指令中的所有传感器数据设置为 0。

外部系统状态获取请求(GET_BASE_STATUS)

SLAMWARE CORE 室内定位导航模块在准备好后会一直发送外部系统连接请求确认外部系统已正常工作。直到外部系统响应此请求后 SLAMWARE CORE 才会开

始正常工作。SLAMWARE CORE 正常工作后将不断轮询获取外部系统的状态，外部系统应当在规定的时间内给予答复。否则 SLAMWARE CORE 将认为外部系统断开连接。

- 请求数据包：

命令代码

0x30

- 应答数据包：

应答代码

<OK>

负载数据

电池电量百分比

u8

电池充电状态

u8

数据域	类型	描述
电池电量百分比	u8	外部系统的电池电量百分比，0-100
电池充电状态	u8	外部系统的电池状态，该变量为一个位图。 从低位到高位分别是：是否处于充电状态、是否连接了外部电源、是否连接了充电座。 当机器人通过电缆连接了充电器，且电池处于充电状态时，该状态应为 3，即二进制 00000011；当机器人正在通过充电座进行充电，该状态应为 5，即二进制 00000101

图表 4-9 外部系统状态获取请求的应答数据域描述

外部系统轮组状态获取请求(GET_BASE_MOTOR_DATA)

SLAMWARE CORE 室内定位导航模块正常工作后将不断轮询获取外部系统的轮组状态，外部系统应当在规定的时间内给予答复。否则 SLAMWARE CORE 将认为外部系统断开连接。

- 请求数据包：

命令代码

0x31

- 应答数据包：

应答代码	负载数据	
<OK>	左轮电机累计里程 s32	右轮电机累计里程 s32

数据域	类型	描述
左轮电机累计里程	s32	外部系统的左轮电机累计运动里程，单位为 mm。
右轮电机累计里程	s32	外部系统的右轮电机累计运动里程，单位为 mm。

图表 4-10 外部系统轮组状态获取请求的应答数据域描述

备注：该命令支持的轮组类型为两轮。

外部系统距离传感器数据获取请求 (GET_BASE_SENSOR_DATA)

SLAMWARE CORE 室内定位导航模块正常工作后将不断轮询获取外部系统的距离传感器数据，外部系统应当在规定的时间内给予答复。

如果在系统参数获取请求时外部系统告知没有配置距离传感器，SLAMWARE CORE 将不会发送此请求。

- 请求数据包：

命令代码
0x32

- 应答数据包：

应答代码	负载数据
<OK>	距离传感器数据 u32[16]

数据域	类型	描述
距离传感器数据	u32[16]	<p>外部系统的距离传感器数据，结果为距离传感器到障碍物的距离信息，单位为 mm，Q16 定点小数，最多支持 16 组距离传感器。</p> <p>距离传感器包含红外测距传感器和超声波传感器。</p> <p>注：当使用 0x20 底盘配置指令时，最多可支持 8 组距离传感器，此时最多可获取 8 组距离传感器数据；当使用 0x21 Binary Config 指令配置时，最多可支持 16 组距离传感器，此时最多可获取 16 组距离传感器数据。</p>

图表 4-11 外部系统距离传感器数据获取请求的应答数据域描述

外部系统触碰传感器数据获取请求 (GET_BASE BUMPER_DATA)

SLAMWARE CORE 室内定位导航模块正常工作后将不断轮询获取外部系统的触碰传感器数据，外部系统应当在规定的时间内给予答复。

如果在系统参数获取请求时外部系统告知没有配置触碰传感器，SLAMWARE CORE 将不会发送此请求。

- 请求数据包：

命令代码

0x33

- 应答数据包

应答代码

<OK>

负载数据

触碰传感器数据

u8

数据域	类型	描述
触碰传感器数据	u8	<p>表示外部系统的触碰传感器数据。自最低位开始，每个 bit 代表一个碰撞传感器的状态，顺序与配置中的顺序相同。</p> <p>指定位为 1 时，表示该碰撞传感器未触发，指定位为 0 时，表示该碰撞传感器触发。最多支持 8 组触碰传感器</p>

图表 4-12 外部系统碰撞传感器数据获取请求的应答数据域描述

外部系统自动回充数据获取(GET_AUTO_HOME_DATA)

SLAMWARE CORE 室内定位导航模块在试图进行自动回充时，会通过该指令获取自动回充相关的信息。当前主要用于获得对接灯塔 Beacon 信息。

- 请求数据包：

命令代码	负载数据
0x34	数据类型 u8

数据域	类型	描述
数据类型	u8	SLAMWARE CORE 想要获取的自动回充的数据类型，当前支持的类型只有一种： 0 - 获取自动回充灯塔信息 如果这个值不为 0，请使用 not support 错误信息回复该请求 (0x8000)

图表 4-13 外部系统自动回充数据获取请求的请求数据域描述

- 应答数据包（当数据类型为 0 时）

应答代码	负载数据
<OK>	信标数量 u8 接收头数量 u8 接收头数据 u8[n]

数据域	类型	描述
信标数量	u8	外部系统充电座上的信标的数量。当前只支持 3 个信标时，0 号信标指的是左信标，1 号信标指的是主信标，2 号信标指的是右信标。
接收头数量	u8	外部系统机器人上接收头的数量。当前只支持 3 个接收头，0 号接收头指的是左接收头，1 号接收头指的是主接收头，2 号接收头指的是右接收头
接收头数据	u8[n]	每个接收头对应一个字节，每个字节代表这个接收头能接收到的信标。 例如 1 号接收头能接收到 0 号和 1 号信标的信息，那么 $Data[1] = (1 \ll 0) (1 \ll 1)$

图表 4-14 外部系统自动回充数据获取请求的应答数据域描述

外部系统运动控制请求(SET_BASE_MOTOR)

SLAMWARE CORE 室内定位导航模块通过发送运动控制请求控制外部系统运动。外部系统需要根据请求数据包的内容准确控制系统运动才能使 SLAMWARE CORE 正常工作。

- 请求数据包：

命令代码	负载数据
0x40	运动速度 s32[4]

- 应答数据包

应答代码
<OK>

数据域	类型	描述
运动速度	s32[4]	外部系统的轮子运动速度，单位为 mm/s

图表 4-15 外部系统运动控制请求的请求数据域描述

外部系统设置速度及获取 Deadreckon 数据请求 (SET_V_AND_GET_DEADRECKON)

该指令用于设置外部系统运动速度。

- 请求数据包：

命令代码	负载数据		
0x41	X 轴向速度量 s32	Y 轴向速度量 s32	角速度量 s32

- 应答数据包

应答代码	负载数据		
<OK>	X 轴向位移 s32	Y 轴向位移 s32	角度位移 s32

数据域	类型	描述
X 轴向速度量	s32	X 轴向速度，Q16 定点小数
Y 轴向速度量	s32	Y 轴向速度，Q16 定点小数
角速度量	s32	逆时针角速度，Q16 定点小数
X 轴向位移	s32	外部系统整体相对于上一次应答此请求时的 X 轴向位移，单位为 mm，Q16 定点小数。
Y 轴向位移	s32	外部系统整体相对于上一次应答此请求时的 Y 轴向位移，单位为 mm，Q16 定点小数。
角度位移	s32	外部系统整体相对于上一次应答此请求时的逆时针角度位移，单位为度，Q16 定点小数。

图表 4-16 外部系统设置速度及获取 Deadreckon 数据请求的数据域描述

备注: 此处的坐标轴是以机器人自身为中心的右手坐标系, 机器人正前方为 x 轴。

以两轮差动机器人为例, 当机器人收到该指令时, 应当:

1. 根据左右轮的里程累积计算三个位移量 (dx, dy, dyaw) ;
2. 将指令中提供的速度量 (vx, vy, omega) 转换成左右轮的线速度 (vl, vy) ;
3. 返回第一步中的计算结果

通过里程累积计算三个位移量的方法:

1) 变量定义:

变量名称	定义	单位
dl	左轮位移, 机器人向前为正	m
dr	右轮位移, 机器人向前为正	m
dx	机器人正方向位移, 机器人前方为正	m
dy	机器人侧向位移, 机器人左侧为正	m
dyaw	机器人角位移, 逆时针为正	rad

2) 计算公式:

$$dyaw = \frac{dr - dl}{2R}$$

$$dx = \frac{dl + dr}{2} \cos(dyaw)$$

$$dy = \frac{dl + dr}{2} \sin(dyaw)$$

注：R 是机器人轮距半径，单位为 m，下文同。

3) 代码示例

```
float d_yaw = (d_dist_r_mm_f - d_dist_l_mm_f)/2.0f/robot_radius_mm;
float displacement = (d_dist_l_mm_f + d_dist_r_mm_f)/2.0f;
float dx = cos(d_yaw)*displacement;
float dy = sin(d_yaw)*displacement;
ans_pkt->base_dx_mm_q16 = (_32)(dx*(1<<16))
ans_pkt->base_dy_mm_q16 = (_32)(dy*(1<<16))
ans_pkt->base_dtheta_degree_q16 = (_32)(d_yaw/M_PIF*180*(1<<16))
```

根据速度量计算左右轮线速度的方法：

1) 变量定义

变量名称	定义	单位
vx	机器人正向线速度，前进方向为正	m/s
vy	机器人侧向线速度，向左为正，对于二轮差动机器人，该值为 0	m/s
omega	机器人角速度，逆时针为正	rad/s
vl	机器人左轮线速度，机器人向前为正	m/s
vr	机器人右轮线速度，机器人向前为正	m/s

2) 计算公式(以二轮差动模型为例)

$$vl = vx - \omega * R$$

$$vr = vx + \omega * R$$

3) 代码示例

```
base_set_velocity_request_t *req =
(base_set_velocity_request_t*)request->payload;
float speed_l_mm = (float)req->velocity_x_q16 * 1000.0 / (1 << 16);
float speed_r_mm = speed_l_mm;
float line_speed_mm = (float)req->angular_velocity_q16 / (1 << 16)
* robot_radius_mm;
speed_l_mm -= line_speed_mm;
speed_r_mm += line_speed_mm;
```

外部系统命令查询请求(POLL_BASE_CMD)

SLAMWARE CORE 室内定位导航模块可接受外部系统的命令，但由于单向通讯的模式，外部系统只能在 SLAMWARE CORE 发送外部系统控制命令查询请求的时候才可以发送控制命令。SLAMWARE CORE 会定期向外部系统查询控制命令。对于外部系统希望 SLAMWARE CORE 尽快响应的控制命令可以通过#CMD 信号通知 SLAMWARE CORE 尽快发送外部系统控制命令查询请求。在#BUSY 信号置高电平的时候表示 SLAMWARE CORE 正处于忙碌状态，无法响应外部系统的控制命令。但外部系统仍需及时响应 SLAMWARE CORE 发送的请求命令。请注意，SLAMWARE CORE 不会应答外部系统的命令，当 SLAMWARE CORE 接受外部系统的命令后，外部系统可以检测到#BUSY 信号的脉冲。外部系统必须在应答外部系统命令查询请求后拉低#CMD 信号，否则系统将会工作在未知状态。

- 请求数据包：

命令代码

0x50

- 应答数据包

应答代码

<OK>

负载数据

外部系统命令代码

u8

数据域	类型	描述
外部系统命令代码	u8	外部系统的命令代码，只能是 SLAMWARE CORE 支持的命令代码。详情请见下表。

图表 4-17 外部系统命令查询请求的应答数据域描述

外部系统命令代码	命令名称	描述
0x51	GET_INFO	获取 SLAMWARE CORE 系统信息，包含软硬件版本，网络配置。
0x52	RESET_WIFI	重置 SLAMWARE CORE 无线网络
0x53	FW_UPGRADING	更新 SLAMWARE CORE 固件
0x80	START_SWEEP	开始清扫(仅针对扫地机版本)

0x81	STOP_SWEEP	结束清扫(仅针对扫地机版本)
0x82	SPOT_SWEEP	定点清扫(仅针对扫地机版本)
0x90	GET_HEALTH	获取健康状况
0xA0	MOVE_FORWARD	向前
0xA1	MOVE_BACKWARD	向后
0xA2	TURN_LEFT	向左
0xA3	TURN_RIGHT	向右
0xAF	CANCEL_ACTION	取消上一步操作
0xB0	GET_AUXILIARY_ANCHOR	获取辅助定位传感器

图表 4-18 外部系统命令代码列表

外部系统命令回应请求(POLL_BASE_ANS_CMD)

SLAMWARE CORE 在处理完外部系统的命令请求后会给予回应，回应请求总是与外部系统命令查询请求成对出现。

SLAMWARE CORE 支持的外部系统命令代码暂未定义。

- 请求数据包：

命令代码

0x5F

- 应答数据包

应答代码

<OK>

负载数据

应答数据

u8

数据域	类型	描述
应答数据	u8	外部系统命令的应答数据，与 SLAMWARE CORE 命令代码相关

图表 4-19 外部系统命令回应请求的应答数据域描述

SLAMWARE CORE 系统事件通知命令(SEND_EVENT)

SLAMWARE CORE 室内定位导航模块会不定期的向外部系统发送系统事件通知。外部系统可以根据系统事件通知作出相应反应。

注意，SLAMWARE CORE 支持的系统事件代码可能会增减。

- 请求数据包：

命令代码	负载数据
0x60	系统事件代码
	u8

- 应答数据包

应答代码
<OK>

数据域	类型	描述
系统事件代码	u8	SLAMWARE CORE 的系统事件代码，详见下表

图表 4-20 SLAMWARE CORE 的系统事件通知命令的请求数据域描述

目前支持的系统事件代码有：

系统事件代码	命令名称	描述
0x61	LIDAR_CONN_FAIL	LIDAR 连接失败
0x62	LIDAR_RAMPUP_FAIL	LIDAR 启动失败
0x63	SYSTEM_UP_OK	系统启动成功
0x64	FIRMWARE_UPDATE	系统固件升级
0x65	CORE_DISCONNECT	系统断开连接
0x66	FIRMWARE_UPDATE_OK	固件升级成功
0x80	START_SWEEP	通知外部系统清扫开始(仅针对扫地机版本)
0x81	END_SWEEP	通知外部系统清扫结束(仅针对扫地机版本)

图表 4-21 SLAMWARE CORE 支持的系统事件代码

外部系统健康管理命令请求(HEALTH_MGMT)

SLAMWARE CORE 室内定位导航模块会不定期的向外部系统发送健康管理相关的命令请求。外部系统可以根据具体的命令请求作出相应的反应。具体支持的事件代码如下：

健康管理事件代码	命令名称	描述
0x01	HEALTH_GET_HEALTH	获取外部系统健康状况
0x02	HEALTH_GET_ERROR	请求取外部系统错误的详细信息
0x03	HEALTH_CLEAR_ERROR	请求清除外部系统错误信息

图表 4-22 SLAMWARE CORE 健康管理命令支持的事件代码

HEALTH_GET_HEALTH

- 请求数据包：

命令代码

0x90

0x01

- 应答数据包

应答代码

<OK>

Health_flag
u8

Error_count
u8

数据域	类型	描述
Health_flag	u8	表示健康信息按照严重程度的分类，具体定义请参考下表
Error_count	u8	表示错误信息的数目

图表 4-23 HEALTH_GET_HEALTH 命令数据域描述

Error Flag	Bit	描述
保留	[7:3]	保留
FATAL	[2]	表示是否有 FATAL。 0 = Has no FATAL 1 = Has FATAL
ERROR	[1]	表示是否有 ERROR。 0 = Has no ERROR 1 = Has ERROR
WARN	[0]	表示是否有 WARN。 0 = Has no WARN 1 = Has WARN

图表 4-24 HEALTH_GET_HEALTH 命令中 Health_flag 定义

HEALTH_GET_ERROR

○ 请求数据包：

命令代码

0x90

0x02

error id

u8

○ 应答数据包

应答代码

<OK>

error_code

Error_message

u32

u8[32]

数据域	类型	描述
error_id	u8	表示第几个 error，是一个小于 error count 的整数
error_code	u32	具体请参考下表。
error_message	u8[32]	表示具体错误信息的字符串

图表 4-25 HEALTH_GET_ERROR 命令数据域描述

Error Code	Bit	描述
错误等级	[31:24]	表示错误的严重程度。 0x01 = warn ; 0x02 = error ; 0x03 = fatal
错误组件	[23:16]	表示发生错误的具体位置或组件。 0 = USER ; 1 = SYSTEM ; 2 = POWER ; 3 = MOTION ; 4 = SENSOR
具体信息	[15:8]	表示具体的错误信息。
具体组件	[7:0]	表示具体某个组件有错误。

图表 4-26 HEALTH_GET_ERROR 命令中返回数据 error_code 定义

Sample:

```
// sensor errors
#define BASE_SENSOR_FATAL_CONTROLLER_DOWN
(SLAMWARECORE_HEALTH_ERROR_FATAL | BASE_COMPONENT_SENSOR | 0x0000u)
```

```

#define BASE_SENSOR_WARN BUMPER_DOWN
(SLAMWARECORE_HEALTH_ERROR_WARN | BASE_COMPONENT_SENSOR | 0x0100u)
#define BASE_SENSOR_ERROR BUMPER_DOWN
(SLAMWARECORE_HEALTH_ERROR_ERROR | BASE_COMPONENT_SENSOR | 0x0100u)
#define BASE_SENSOR_FATAL BUMPER_DOWN
(SLAMWARECORE_HEALTH_ERROR_FATAL | BASE_COMPONENT_SENSOR | 0x0100u)
#define BASE_SENSOR_WARN CLIFF_DOWN
(SLAMWARECORE_HEALTH_ERROR_WARN | BASE_COMPONENT_SENSOR | 0x0200u)
#define BASE_SENSOR_ERROR CLIFF_DOWN
(SLAMWARECORE_HEALTH_ERROR_ERROR | BASE_COMPONENT_SENSOR | 0x0200u)
#define BASE_SENSOR_FATAL CLIFF_DOWN
(SLAMWARECORE_HEALTH_ERROR_FATAL | BASE_COMPONENT_SENSOR | 0x0200u)
#define BASE_SENSOR_WARN SONAR_DOWN
(SLAMWARECORE_HEALTH_ERROR_WARN | BASE_COMPONENT_SENSOR | 0x0300u)

```

HEALTH_CLEAR_ERROR

- 请求数据包：

命令代码

0x90

0x03

error code

u32

- 应答数据包

应答代码

<OK>

外部系统辅助定位数据获取请求

(GET_AUXILIARY_ANCHOR)

SLAMWARE CORE 室内定位导航模块会不定期的向外部系统发送获取辅助定位数据相关的命令请求。

- 请求数据包：

命令代码

0x35

○ 应答数据包

应答代码	负载数据	
<OK>	Flag	Anchors
	u8	u8[*]

数据域	类型	描述
Flag	u8	具体请参考图表 4-28。
Anchors	u8[*]	表示多个 AnchorInfo 的数据域。具体格式请参考图表 4-29。

图表 4-27 GET_AUXILIARY_ANCHOR 命令数据域描述

Flag	Bit	描述
辅助传感器类型	[7:5]	表示辅助传感器的类型。 0x00 = UWB ;
辅助传感器标准差域	[4]	表示辅助数据是否支持标准差测量。 0 = 不支持 ; 1 = 支持 ;
辅助传感器锚数量	[15:8]	表示辅助传感器锚数量。

图表 4-28 GET_AUXILIARY_ANCHOR 命令数据域描述



图表 4-29 GET_AUXILIARY_ANCHOR 命令中 AnchorInfo 的数据格式

AnchorInfo 数据域	类型	描述
id	u16	表示 Anchor 的 id。
distance	u16	表示当前 Anchor 的距离单位为毫米。
Max error	u8	表示距离的标准差。单位为毫米。（如果传感器不支持标准差测量，则没有此数据域。）

图表 4-30 GET_AUXILIARY_ANCHOR 命令中 AnchorInfo 数据域描述

日期	版本	描述
2015-03-18	0.1	初始版本，定义 CTRL BUS 通讯协议
2015-03-20	0.2	协议更新
2015-03-30	0.3	协议更新，定点小数定义
2015-09-09	0.4	协议更新，更新连接命令，轮组状态获取命令 增加外部系统命令回应请求和 SLAMWARE CORE 系统事件通知命令
2015-11-04	0.5	修正 SLAMWARE CORE 系统事件通知命令协议格式。
2015-12-30	0.6	更新版权信息和文字润色
2016-03-15	1.7	增加了一些新的指令：外部系统配置获取请求 (0x21)，外部系统自动回充数据获取请求 (0x34)
2016-05-19	1.8	增加了 Inter-chip 通讯协议规范并更新文档模板
2016-06-17	1.8	增加了外部系统命令代码和关于扫地机版本的系统事件通知命令
2016-07-15	1.8	增加了关于碰撞传感器返回数据格式的描述
2016-09-07	1.8	增加 SET_V_AND_GET_DEARECKON 接口定义
2016-09-12	1.8	增加 HEALTH_MGMT 接口定义
2016-11-02	1.8	增加 GET_AUXILIARY_ANCHOR 接口定义

图表索引

图表 1-1 SLAMWARE CORE CTRL BUS PROTOCOL 和 INTER-CHIP PROTOCOL 关系示意图.....	3
图表 2-1 SLAMWARE CORE 内部模块框图.....	4
图表 3-1 INTER-CHIP PROTOCOL 数据包格式定义.....	6
图表 3-2 INTER-CHIP PROTOCOL FLAG 字段格式说明.....	6
图表 3-3 INTER-CHIP PROTOCOL FLAG 字段标志位说明.....	6
图表 3-4 INTER-CHIP PROTOCOL LEN 和 LENH 字段格式说明.....	7
图表 3-5 INTER-CHIP PROTOCOL CMD 字段格式说明.....	7
图表 3-6 INTER-CHIP PROTOCOL PAYLOAD 字段格式说明.....	8
图表 3-7 INTER-CHIP PROTOCOL CHKSUM 字段格式说明.....	8
图表 3-8 INTER-CHIP PROTOCOL EXFLAG[N]字段格式说明.....	8
图表 3-9 INTER-CHIP PROTOCOL ADDR 和 EXADDR[N]字段格式说明.....	9
图表 3-10 基于 INTER-CHIP PROTOCOL 的 MASTER 和 SLAVE 通讯规则.....	10
图表 3-11 INTER-CHIP PROTOCOL FLAG 字段格式说明.....	10
图表 3-12 INTER-CHIP PROTOCOL FLAG 数据包格式说明.....	10
图表 3-13 INTER-CHIP PROTOCOL 请求命令数据包格式说明.....	11
图表 3-14 INTER-CHIP PROTOCOL 保留的 CMD 说明.....	11
图表 3-15 INTER-CHIP PROTOCOL 应答数据包格式说明.....	11
图表 3-16 INTER-CHIP PROTOCOL 保留的 RESP 说明.....	12
图表 3-17 INTER-CHIP PROTOCOL 错误码说明.....	12
图表 4-1 SLAMWARE CORE CTRL BUS 通讯协议数据包格式.....	13
图表 4-2 SLAMWARE CORE CTRL BUS 请求应答通讯模式说明.....	14
图表 4-3 SLAMWARE CORE 请求命令列表.....	15
图表 4-4 外部系统连接请求与应答数据域描述.....	16
图表 4-5 外部系统参数获取请求的应答数据域描述.....	17
图表 4-6 外部系统所支持的外部系统尺寸类型.....	17
图表 4-7 外部系统所支持的轮组类型.....	17
图表 4-8 外部系统传感器安装位置 POS 数据.....	18
图表 4-9 外部系统状态获取请求的应答数据域描述.....	19
图表 4-10 外部系统轮组状态获取请求的应答数据域描述.....	20
图表 4-11 外部系统距离传感器数据获取请求的应答数据域描述.....	21
图表 4-12 外部系统碰撞传感器数据获取请求的应答数据域描述.....	21
图表 4-13 外部系统自动回充数据获取请求的请求数据域描述.....	22
图表 4-14 外部系统自动回充数据获取请求的应答数据域描述.....	22
图表 4-15 外部系统运动控制请求的请求数据域描述.....	23
图表 4-16 外部系统设置速度及获取 DEADRECKON 数据请求的数据域描述.....	24
图表 4-17 外部系统命令查询请求的应答数据域描述.....	26

图表 4-18 外部系统命令代码列表.....	27
图表 4-19 外部系统命令回应请求的应答数据域描述.....	27
图表 4-20 SLAMWARE CORE 的系统事件通知命令的请求数据域描述.....	28
图表 4-21 SLAMWARE CORE 支持的系统事件代码.....	28
图表 4-22 SLAMWARE CORE 健康管理命令支持的事件代码.....	29
图表 4-23 HEALTH_GET_HEALTH 命令数据域描述.....	29
图表 4-24HEALTH_GET_HEALTH 命令中 HEALTH_FLAG 定义.....	29
图表 4-25 HEALTH_GET_ERROR 命令数据域描述.....	30
图表 4-26 HEALTH_GET_ERROR 命令中返回数据 ERROR_CODE 定义.....	30
图表 4-27 GET_AUXILIARY_ANCHOR 命令数据域描述.....	32
图表 4-28 GET_AUXILIARY_ANCHOR 命令数据域描述.....	32
图表 4-29 GET_AUXILIARY_ANCHOR 命令中 ANCHORINFO 的数据格式.....	32
图表 4-30 GET_AUXILIARY_ANCHOR 命令中 ANCHORINFO 数据域描述.....	32