

# HARP-5 用户手册

手持式多总线仿真设备，带显示屏和键盘

(目录序号与英文版保持一致)

6 参数列表 .....	8
6.1 产品照片 .....	8
6.2 产品介绍 .....	8
6.3 操作模式 .....	9
6.4 仿真模式 .....	10
6.5 CAN 和 LIN 总线性能 .....	10
6.6 LinWorks 套件 .....	11
6.7 技术规格 .....	11
6.8 用例 .....	14
6.9 硬件需求 .....	14
6.10 软件要求 .....	15
6.11 LINWorks 工作流程 .....	15
6.12 供货清单 .....	16
7 开始工作 .....	16
7.1 简介 .....	16

7.2 安装 .....	16
7.3 创建 SDF 文件 .....	17
7.4 上传 SDF 文件 .....	18
7.5 连接 .....	18
7.6 开始总线通信 .....	19
8 处理凭证和激活码 .....	20
8.1 将凭证代码转换为激活码 .....	20
8.2 兑换激活码 .....	22
8.3 检查已安装的激活码 .....	23
9 硬件 .....	24
9.1 总览 .....	24
9.2 连接器 .....	25
9.2.1 X5-LIN,CAN 和 IO .....	25
9.2.2 X7-LIN 和 CAN .....	26
9.2.3 X8-PC 连接器 .....	30
9.2.4 X9-SD 卡槽 .....	30
9.2.5 X10-逻辑电源和充电电源 .....	31
9.3 电源供电 .....	31
9.4 KB1-键盘 .....	32
9.5 D1-显示屏 .....	32
9.6 硬件适配 .....	33
9.6.1 如何更换充电电池 .....	33

9.6.2 如何更换实时时钟(RTC)电源 .....	35
10 固件 .....	38
10.1 固件升级 .....	38
10.1.1 介绍 .....	38
10.1.2 更新需要的软件 .....	39
10.1.3 更新固件 .....	39
10.2 PC 模式 .....	40
10.2.1 PC 模式介绍 .....	40
10.2.2 启用 PC 模式 .....	41
10.3 独立运行模式和自启动 .....	41
10.3.1 允许独立运行模式 .....	41
10.3.2 配置自动运行的宏 .....	41
10.3.3 SDF 永久储存 .....	42
10.3.4 配置设备自动加载一个 SDF 文件并运行一个宏 .....	42
10.4 记录 .....	43
10.4.1 简介 .....	43
10.4.2 配置和启用记录功能 .....	43
10.4.3 混合模式 .....	45
10.4.4 记录文件目标 .....	46
10.4.5 记录数据格式 .....	48
10.5 设置 HARP-5 .....	50
10.6 手动控制 .....	54

10.6.1 简介 .....	54
10.6.2 键盘布局 .....	54
10.6.3 菜单 .....	55
10.6.4 导航 .....	67
11 SDF-会话配置文件 .....	68
11.1 通道和 SDF 配置组 .....	68
11.1.1 介绍 .....	68
11.1.2 配置组 .....	69
11.1.3 通道 .....	69
11.1.4 映射关系 .....	70
11.2 配置属性 .....	71
11.3 设备通道设置 .....	72
11.3.1 介绍 .....	72
11.3.2 配置选项列表 .....	72
11.3.3 设置目标特定选项 .....	74
11.4 虚拟信号 .....	75
11.4.1 介绍 .....	75
11.4.2 信号长度 .....	76
11.4.3 范围 .....	76
11.4.4 总线开始时重置 .....	76
11.5 系统变量 .....	77
11.5.1 介绍 .....	77

11.5.2 概览 .....	78
11.5.3 定时 .....	81
11.5.4 数字 IO .....	86
11.5.5 系统 .....	89
11.5.6 存储 .....	93
11.5.7 记录 .....	96
11.5.8 LIN 报头 .....	99
11.5.9 协议 .....	102
11.6 宏命令 .....	104
11.6.1 概览 .....	104
11.6.2 宏条件 .....	108
11.6.3 宏变量 .....	109
11.7 设备相关选项 .....	109
11.7.1 简介 .....	109
11.7.2 键盘 .....	109
11.7.3 记录 .....	110
12 LINWorks 软件 .....	111
12.1 总览 .....	111
12.2 Baby-LIN 驱动 .....	112
12.3 LDFEdit .....	113
12.4 SessionConf .....	113
12.5 SimpleMenu .....	114

12.6 LogViewer .....	114
12.7 Baby-LIN-DLL .....	115
12.8 Baby-LIN DLL .NET 封装 .....	116
12.9 Baby-LIN DLL LabView 封装 .....	116
12.10 Serial writer .....	117
12.11 BLProg.....	118
12.12 Switch to ASCII tool.....	118
13 协议和应用 .....	118
13.1 自定义协议.....	118
13.1.1 介绍 .....	118
13.1.2 需求.....	119
13.1.3 执行协议服务 .....	120
13.1.4 协议相关的系统变量.....	121
13.1.5 协议设置 .....	122
13.1.6 服务属性.....	122
13.1.7 常量映射 .....	124
13.1.8 信号映射 .....	124
13.1.9 字节顺序和位顺序 .....	126
13.2 巨型帧：具有 8 个以上数据字节的 LIN 帧.....	133
13.2.1 简介 .....	133
13.2.2 通过注入命令的巨型帧 .....	134
13.2.3 使用 SimpleMenu 测试巨型帧 .....	135

13.2.4 使用 Baby-LIN-DII 自定义应用程序 .....	137
13.3 诊断模式 .....	139
13.3.1 支持的诊断模式 .....	139
13.3.2 睡眠指令（特殊的主请求帧） .....	162
13.3.3 目标配置 .....	162
13.4 ASCII 模式 .....	164
13.4.1 简介 .....	164
13.4.2 切换 ASCII 模式 .....	164
13.4.3 关闭 ASCII 模式 .....	165
13.4.4 ASCII 命令 .....	166
13.4.5 应用实例 .....	170

## 6 参数列表

### 6.1 产品照片



### 6.2 产品介绍

HARP-5 是一款无需 PC 的手持设备，用于 LIN 和 CAN 总线的休眠总线仿真。使用全套设备，可以同时控制 4 条总线（2 路 LIN 总线，1 路高速 CAN，1 路低速 CAN）。

设备中已包含所有 4 个总线接口的接口硬件。作为基本系统的一部分，一路 LIN 总线（SDF-V2 格式）立即可用。第二路 LIN 总线接口，CAN 总线接口和第一路 LIN 接口支持 SDF-V3 格式都将由单独订购的凭证代码激活。因此，您只需支付您真正需要使用的部分费用，并且如果您的需求发生变化，您可以随时轻松升级。

HARP-5 拥有 128x64 像素的单色图形 LCD 显示屏，可调背光亮度和对比度。额外的显示内容可以在 SDF 中定义。除了显示信号的原始值或转换后的物理值外，还可以在显示器上放置宏和宏选项。

键盘菜单提供多达 10 个功能键。可以将按键配置为执行宏或触发一些操作，如信号更改或总线启动或总线停止。另外，键盘菜单允许显示最多 5 个信号值。



系统的 SD 卡可以存储多个 SDF 文件。用户可以通过菜单选择哪一个用于当前配置。因此，可以用一个设备来操作多个需要不同 SDF 文件的节点。

HARP-5 可以由 6 个附带的可充电电池，附带的电源或 LIN 总线电源供电。充满电的电池可以供电 16 小时。HARP-5 有一个充电电路，电池可以在不取出的条件下使用附带的电源进行充电。充满电需要的时间约为 8 小时。

集成的可开关 12V 电压发生器可用于为连接的总线节点提供 150mA 的电流。这个 12V 电源可以通过 SDF 系统变量开关。

数据记录器功能允许在 SD 卡或通过 USB 连接的 PC 上记录总线报文帧。可配置信号过滤和阈值触发。

单个共享数字输入/输出可用。

使用 PC 模式时，HARP-5 和普通的 Baby-LIN 设备相同，可以通过 LINWorks 软件或使用 Baby-LIN-DLL 的应用程序进行控制。

HARP-5 可以处理 8-18 VDC 范围内的总线电压。

USB 接口与所有其他接口电隔离，消除了 PC 与其通信对象的电路板上的电子元件间的干扰。

HARP-5 单元包含自己的 32 位微控制器，负责处理 LIN 和 CAN 总线协议的所有时间关键任务。

设备固件可在现场更新，因此可轻松调整总线规格或更新新的系统功能。

如果购买了可选的凭证代码，HARP-5 可以支持 SDF-V3。新一代的 SDF 支持更多总线功能，例如条件宏命令，新系统变量，新的 CRC 功能和子宏调用等。

### **6.3 操作模式**

任何需要与 LIN 或 CAN 设备通信的情况都是 HARP-5 的潜在应用领域。它是一种多功能工具，可用于研究实验室，测试部门和生产（EOL 应用）。

HARP-5 允许不同的操作模式来支持典型的使用情况，例如：

- 监视并记录总线上的所有报文帧。如果拥有 SDF 文件，则也可以同时监视信号值。
- 通过 LINWorks 软件或通过客户利用 Baby-LIN-DLL 开发的应用程序控制总线。
- 在 HARP-5 中编程并存储命令序列时，可脱离 PC 作为独立设备运行。因此，您可以在未连接任何 PC 的情况下操作总线驱动的 ECU 进行耐久性测试或 EOL 应用。

## 6.4 仿真模式

HARP-5 能够模拟 LIN-和 CAN 总线节点的不同配置。可以模拟从零到全部的任意数量的节点。

这些是一些典型的配置：

- LIN 总线：模拟 LIN 总线主节点操作从节点。
- LIN 总线：模拟任意数量的 LIN 总线从节点。
- LIN 和 CAN 总线：模拟除一个节点以外的所有节点，并实现剩余总线模拟。
- LIN 和 CAN 总线：模拟所有节点，从而实现总线上的完整通信。
- LIN 和 CAN 总线：不模拟节点只监视总线通信。
- 可以同时完成 LIN 和 CAN 总线的仿真。

## 6.5 CAN和LIN总线性能

所用的 LIN 驱动器支持 8-18 VDC 的总线电压，支持高达 125k 波特率。甚至可以用 HARP-5 控制在 LIN 规范的标准限制之外运行的节点。

支持的 LIN 版本是 V.1.2, V.1.3, ... V.2.2。

HARP-5 的一路 CAN 总线被设计为符合 ISO-11898 标准的高速接口，带有 SN65HVD251 驱动器。

HARP-5 的一路 CAN 总线使用符合 ISO-11519 标准的 TJA1055T 驱动器，为容错低速物理层。

LIN-和 CAN 总线支持的最大信号电缆长度为 30 米。

## 6.6 LinWorks套件

购买的 HARP-5 包括免费的 LINWorks 套件。该套件是一系列 PC 软件，可在整个工作流程中为您提供支持。

LDFedit 允许检查，创建和编辑 LDF 文件（LIN 描述文件）。

SessionConf 允许查看，创建和编辑 SDF 文件（会话描述文件），可以通过导入 LDF 文件（用于 LIN 总线模拟）或 DBC 文件创建。它定义了不同总线上完整模拟所需的内容，例如：每个总线上的哪些节点可用以及哪些节点应由 HARP-5 模拟。此外，它允许定义应用程序逻辑。这种编程能力适用于每个设备。

SimpleMenu 用于建立与 HARP-5 的连接并上传 SDF 文件到 HARP-5，更改设备目标配置，控制总线以及监控总线上的帧和信号。即使没有 LDF/DBC 文件，也可以监视总线并记录帧。

Baby-LIN-DLL 允许客户创建自己的应用程序并使用 HARP-5 的所有功能，如控制和监控 LIN 和 CAN 总线接口。Baby-LIN-DLL 是 C / C ++的本地 DLL。.NET 应用程序的封装也通过 LabView 文件提供。示例适用于所有受支持的编程语言。

LogViewer 可以显示和转换 HARP-5 以及 SimpleMenu 的记录文件。

LINWorks 软件在 32 位和 64 位 Windows 版本上运行。Baby-LIN DLL 也可提供 Linux 版本。

## 6.7 技术规格

### 设备

- CPU: ARM Cortex-M4,168 MHz

- 内存: 1 MB SRAM
- 可调背光的图形 LCD 显示屏 (128x64 字符)
- 带 12 个键的薄膜键盘
- 声音信号发生器 (可通过 SDF 控制)
- 实时时钟 (电池供电)
- 用于节点供电的集成 12 伏电源 (150mA)
- 三种供电方式: 6 节可充电电池, 附带的电源或 LIN 总线电源
- 使用电池时的运行时间: 最长 16 小时。使用显示器背光、内部 12V 电源、记录到 SD 卡或是其他功能时, 将会缩短此时间。
- 电池充电时间: 最长 8 小时
- 可通过电源连续工作
- 电源供电: 8-32 VDC
- 最大电流消耗: 60mA @ 24VDC

## LIN接口

- 1 路 LIN 总线接口
- 1 路额外的 LIN 总线接口, 硬件可用但未激活, 需要激活码
- 1 路 LIN 总线通过 6 针连接器 (MC 1,5/ 6-ST-3,81) 连接
- 2 路 LIN 总线通过 9 针 Sub-D 连接器连接
- LIN 总线电源电压: 8-18 VDC
- LIN 总线波特率: 最高 125 kBaud (支持 LIN 规范之外的协议)
- 支持的 LIN 版本: V1.2, V1.3, ... V2.2

- 支持的 LIN 相关协议: Cooling 和 SAE J2602
- LIN 总线的最大信号电缆长度: 30m

### CAN接口

- 1 路高速 CAN 总线接口 (CAN-HS) , 符合 ISO-11898 标准, 硬件可用但未激活, 需要激活码
- 1 路容错低速 CAN 总线接口 (CAN-LS) , 符合 ISO-11519 标准, 硬件可用但未激活, 需要激活码
- 通过 6 针连接器 (MC 1,5/ 6-ST-3,81) 连接低速 CAN 总线
- 通过 9 针 Sub-D 连接器进行高速 CAN 总线连接 (引脚分配按照 CiA DS102)
- CAN 总线的最大信号电缆长度: 30 米

### USB设备插口

- USB 2.0 接口
- 通过 USB 型 B-Mini 连接

### SD卡插口

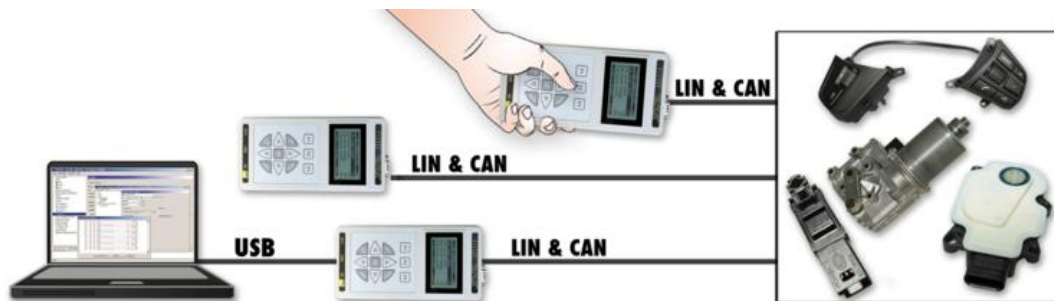
- 支持的卡类型: SD 卡, SDHC 卡
- 支持的文件系统: FAT-32, FAT-16
- SD 卡最大容量: 32 GB

### 外盒

- 防护等级: IP20
- 工作温度: 0 - +45°C
- 重量: 580 克
- 尺寸[mm]: 94 x 183 x 41 (长 x 宽 x 高)

以上不包括连接器等元素。

## 6.8 用例



## 6.9 硬件需求

操作 Baby-LIN 需要以下设备:

需求	目的
一台有大约 200 MB 可用硬盘空间的 PC	安装 LINWorks 软件时需要。请参考软件要求和用例。
可用的 USB 接口	由 SimpleMenu 控制设备的 PC 模式需要。升级固件时需要。
SDHC 读卡器	将 SDF 文件传输到 HARP-5 的 SDHC 中时需要。

## 6.10 软件要求

LINWorks 软件可以在以下操作系统中使用：

- Windows XP
- Windows Vista (32 位和 64 位)
- Windows 7 (32 位和 64 位)
- Windows 8 (32 位和 64 位)
- Windows 10 (32 位和 64 位)
- Linux

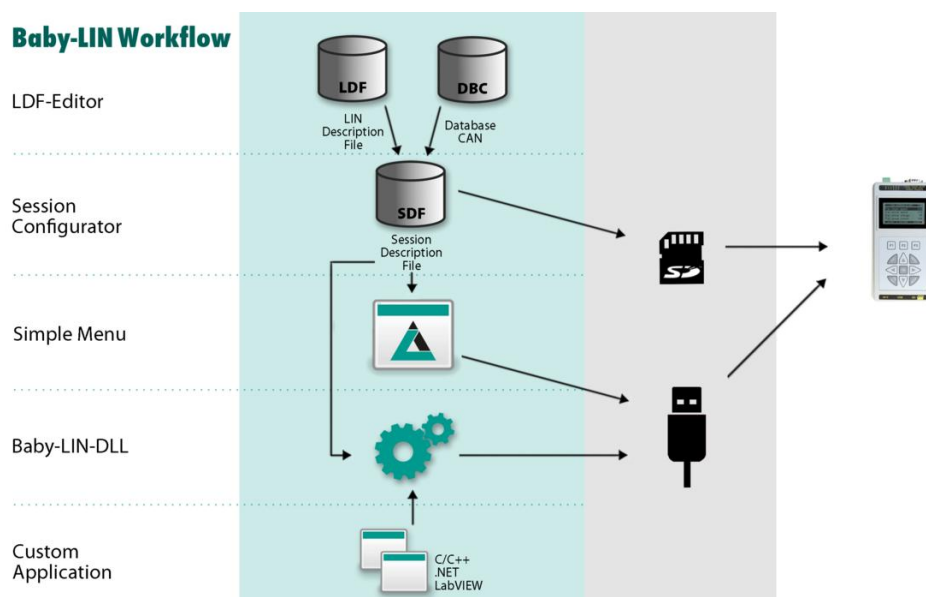
**Version incompatibility:** 对于 Linux 系统，我们只能按照要求提供 Baby-LIN-DLL。DLL

允许在 Linux 上编写自己的自定义应用程序。所有其它 LINWorks 组件仅适用于 Windows 操作系统。

LINWorks 软件套件中提供的一些其他工具需要安装 .NET Framework v4.0。

安装 LINWorks 组件可能需要管理员权限。

## 6.11 LINWorks 工作流程



## 6.12 供货清单

HARP-5 系统包括以下组件：

- HARP-5 设备
- 8 GB SDHC 卡
- 6 个可充电电池（已安装在设备中）
- 充电电源（24 V，1.25 A）
- 1 个螺钉连接的 6 针插头（MC 1,5/ 6-ST-3,81）
- LINWorks 套件下载许可证（包括 LINWorks PC 软件，USB 驱动程序，示例文件和文档）

## 7. 开始工作

### 7.1 简介

本章节将向您展示如何设置 HARP-5 与 LIN 或 CAN 总线进行通信或进行监控。操作步骤如下。

**Tip:** 本章节适用于新的 HARP-5 用户。如果您已经具备使用 Baby-LIN 产品的经验，或者您是高级 LIN 或 CAN 总线用户，那么本章节可能不适合您。

**Advice:** 本指南假定您使用的是 Microsoft Windows 操作系统。

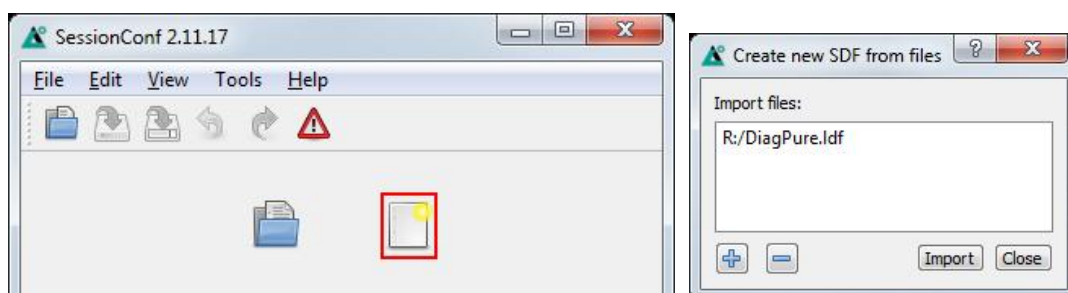
### 7.2 安装



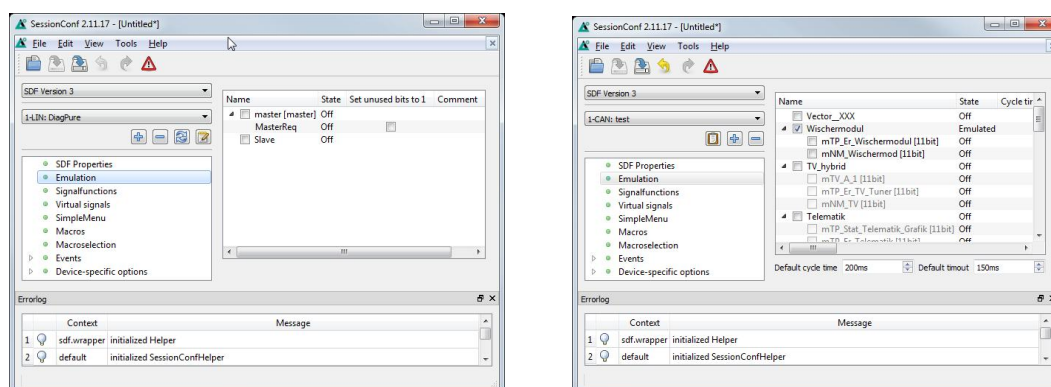
在开始使用 HARP-5 之前，您必须安装 LINWorks 软件的几个组件。请向您的供应商寻求安装文件。

### 7.3 创建SDF文件

首先，我们必须创建一个 SDF 文件。该文件包含 HARP-5 为模拟 LIN-或 CAN 总线上的节点或监视总线所需的所有信息。为此，请启动 SessionConf 并点击右边的图标来创建一个新的 SDF 文件。添加你的 LDF 或 DBC 文件，然后点击导入。要导入 DBC 文件，您还必须选择要导入的节点，帧和信号。

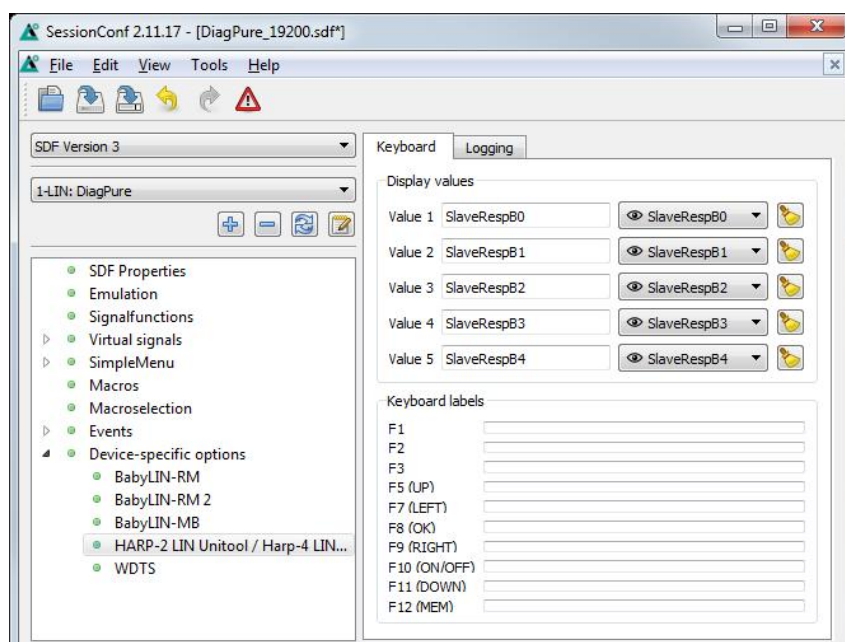


在左边的导航菜单中选择 Emulation，然后选择想要 HARP-5 模拟的节点。如果你只想监视 LIN 或 CAN 总线，不要选择任何节点。



**Attention:** 对于 CAN 总线仿真，您不仅需要选择 HARP-5 要仿真的节点，还要选择节点应该在总线上传输的帧。如果选择了一个节点，但没有选择帧，则不会在该节点总线上看到任何数据。

在左侧菜单中选择设备特定选项的 HARP 条目。在这里，您可以在 HARP-5 显示屏上添加最多 5 个您想观看的信号。



**Advice:** 还有其他方法可以监视帧和信号，但这是一个很好且可配置的起点。

## 7.4 上传SDF文件

要将 SDF 传输到 HARP-5，您需要一个 SDHC 读卡器。将您新创建的 SDF 复制到 SDHC 卡的根目录（HARP-5 将一起提供一张 SDHC 卡）。从读卡器上取下 SDHC 卡并将其插入 HARP-5 的 SDHC 卡插槽中。

## 7.5 连接

需要建立以下连接：

- 将 HARP-5 连接到 CAN 或 LIN 总线。

**CAUTION:** 将 LIN 总线电压保持在以下范围内：8-18 VDC。否则 HARP-5 可能会损坏，并可能对您造成伤害。

**CAUTION:** 在使用高于 18 VDC 的电压之前检查所有 LIN 总线节点规格。

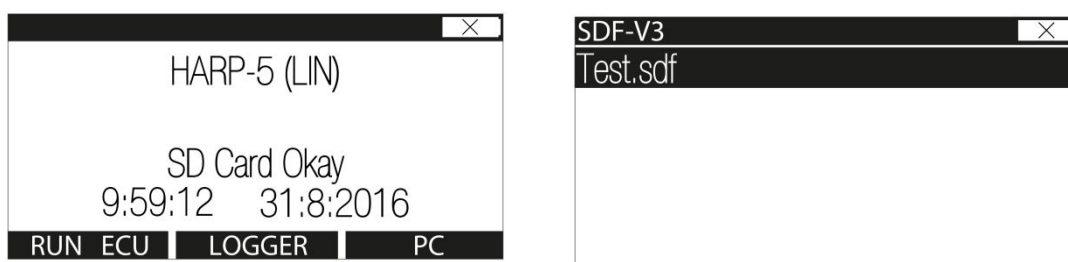
如果超过 18 VDC 的电压用作 LIN 总线电源电压，则必须确保所有连接的节点都可以应对该电压。由于 LIN 规范规定最大电压为 18 VDC，因此有些节点可能无法在超过 18 VDC 的电压下正常工作。

**Tip:** 确保所有其他节点已连接并正常运行。

- 如果可充电电池电量不足，则必须连接提供的充电电源。

## 7.6 开始总线通信

在主菜单中点击“F1”键打开“RUN ECU”菜单，找到刚刚创建的 SDF，选择它并按下“确定”键。



现在可以看到添加到屏幕的变量。要开始模拟/监测，请点击“F1”键选择“START”选项。



现在您可以看到这些信号的实时变化。


**Tip:** HARP-5 具有更多的功能和可能性，可用于多种应用。请继续阅读本手册以了解有关 HARP-5 的更多信息。

## 8. 处理凭证和激活码

### 8.1 将凭证代码转换为激活码

HARP-5 的某些功能在使用之前需要激活。因此，您必须购买所需功能的凭证代码。购买后，您可以在发货单上找到您的凭证代码。

LIPOWSKY Industrie-Elektronik GmbH, Römerstr. 57, D-64291 Darmstadt



**LIPOWSKY**  
**INDUSTRIE-ELEKTRONIK**

Embedded Solutions  
CAN- and LIN-TOOLS

Record Date	23.08.2016
Record ID	62836
Contact	Frau Kerstin Lipowsky
Your Customer ID	10801
Our Supplier ID	511389
Your VAT-ID	
Your Order dated	22.08.2016
Order Reference	

**Delivery Note Nr. 62836** **Page 1 of 3**


Pos.	Art-No.	Article Name	Amount	Batch Number
1	8000800	Option BL-HARP SDFV3-LIN Licence code for Baby-LIN-RM-II / HARP-4 to support enhanced LIN functions of LINWorks V.2.x (SDF V.3.x)	1 Stück	1 Total amount delivered batch 19722001

Convert your Voucher Code on  
[www.optionshop.de/lipowsky](http://www.optionshop.de/lipowsky)

Customs Tarriffno 90309085  
country of origin DE

Voucher codes: 57BC00CFB5A2

Lipowsky Industrie-Elektronik GmbH  
Römerstrasse 57, 64291 Darmstadt  
CEO: Dipl.-Ing. Andreas Lipowsky  
Domicile: Darmstadt, Germany  
Trade register: Darmstadt HRB 5139  
VAT-ID: DE 111647423  
WEEE-Reg-Nr. DE 43826474



Phone: +49 6151 93591-0 Fax: +49 6151 93591-28  
Email: [info@lipowsky.de](mailto:info@lipowsky.de) Web: [www.lipowsky.de](http://www.lipowsky.de)  
D-U-N-S® Nr. 341224640  
Commerzbank AG BLZ 508 400 05 Account No.: 1408756  
IBAN: DE19508400050140875600  
SWIFT/BIC: COBADEFF508

您需要使用相关设备的序列号将这些凭证代码转换为该设备专属的激活码。这可以通过 Lipowsky 商店完成: [www.optionshop.de/lipowsky](http://www.optionshop.de/lipowsky)。在这个网站上点击“Convert voucher code”。



**Welcome to the Optionshop of the company Lipowsky Industrie-Elektronik GmbH!**

**Do you want to...**

**...Convert a Voucher code?**

Here you can enter voucher codes and Device IDs of your designated device to generate the activation codes.

**...or get activation code?**

You already converted voucher codes but can not find your activation codes?

**Sprache wählen / Select language**

English ▼

在下一个界面上，您需要输入上述凭证代码和设备序列号（也称为“设备 ID”）。然后输入您的电子邮件地址，然后单击“获取激活码”。



**Convert voucher codes**

Please type in your voucher codes and Device IDs for which the activation codes should be generated. Additionally, we need your email-address to which we should send the codes after generation.

<b>Voucher code</b>	<b>Device ID</b>	<input type="button" value="Remove voucher code"/>
<input type="text"/>	<input type="text"/>	
<input type="button" value="Add voucher code"/>		
<b>E-Mail-Address</b>	<b>Confirm E-Mail-Address</b>	
<input type="text"/>	<input type="text"/>	
<input type="button" value="Get activation codes"/>		

**Note:** Please remind that generating the codes might take up to 20 minutes, please do not send your request more than once.

一个表格将反馈所有重要信息，包括创建的激活码。

Device ID	Voucher code	Option	Part-no	Activation code
1382331	5333DD5EDEDE1	LIN-SDF3.X Support	8000800	EB201 HINRT BK00U OHZ1F 3DSVF IH

此外，您将收到一封包含激活码的电子邮件。

**Betreff:** Activation code - summary / Freischaltcode - Auflistung  
**Von:** "lipowsky@optionshop.de" <lipowsky@optionshop.de>  
**Datum:** 14.05.2014 11:59  
**An:**

Verehrter Kunde,  
Dear customer,

wie gewünscht, senden wir Ihnen die erstellten Freischaltcodes.  
as requested, we send you the created activation codes.

Option: LIN-SDF3.X Support  
Part-no: 8000000  
Serial: 1382331  
Activation code: EB201 HINRT BK00U OH21F 3DSVF IH

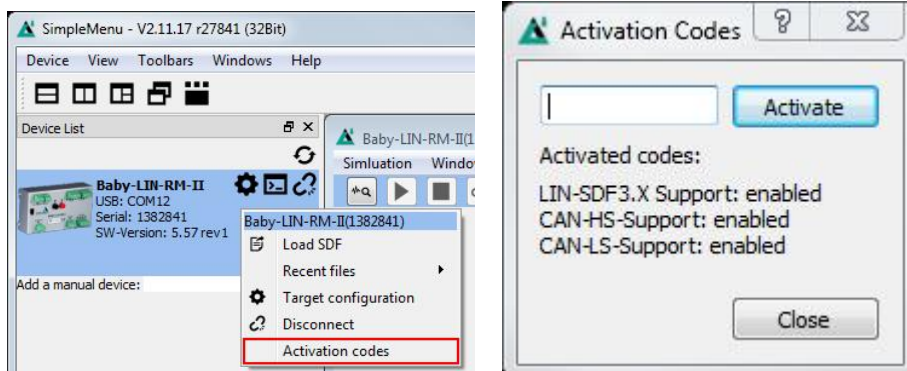
Hinweis: Bitte beachten Sie bei der Eingabe des Freischaltcodes in Ihr Gerät den Unterschied zwischen O = Großbuchstabe "O" und 0 = Zahl "0"!  
Notice: When entering the activation codes into your device, please note the difference between O = capital letter "O" and 0 = number "0"!

**Version incompatibility:** 将凭证代码转换为激活码可能需要一些时间。在您通过电子邮件收到激活码之前，最多可能需要20分钟。

**Version incompatibility:** 如果您将凭证代码转换为激活码，它将被绑定到您输入的序列号的设备。本次激活后，凭证代码不能再次用于其他设备。同时也无法从设备导出激活码以重新获得凭证代码。

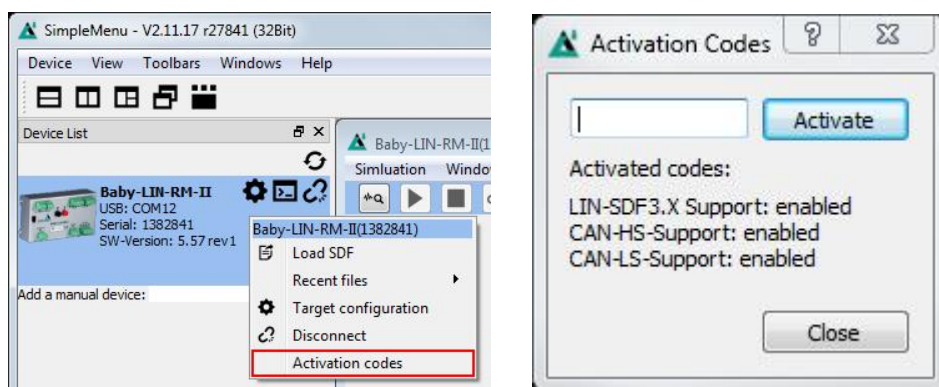
## 8.2 兑换激活码

SimpleMenu 可用于兑换激活码。请将 HARP-5 与 PC 上的空闲 USB 端口连接。然后需要为 HARP-5 启用 PC 模式（请参阅“启用 PC 模式”一章）。启动 SimpleMenu 并与 HARP-5 建立连接。右键单击左侧设备列表中的设备图像，然后选择“激活码”。在新对话框中输入您的激活码并点击“激活”。



### 8.3 检查已安装的激活码

安装的激活码可以使用 SimpleMenu 显示。请将 HARP-5 与 PC 上的空闲 USB 端口连接。需要为 HARP-5 启用 PC 模式（请参阅“启用 PC 模式”一章）。然后启动 SimpleMenu 并与 HARP-5 建立连接，右键单击左侧设备列表中的设备图像，然后选择“激活码”，SimpleMenu 会弹出一个对话框将显示已安装的激活码。



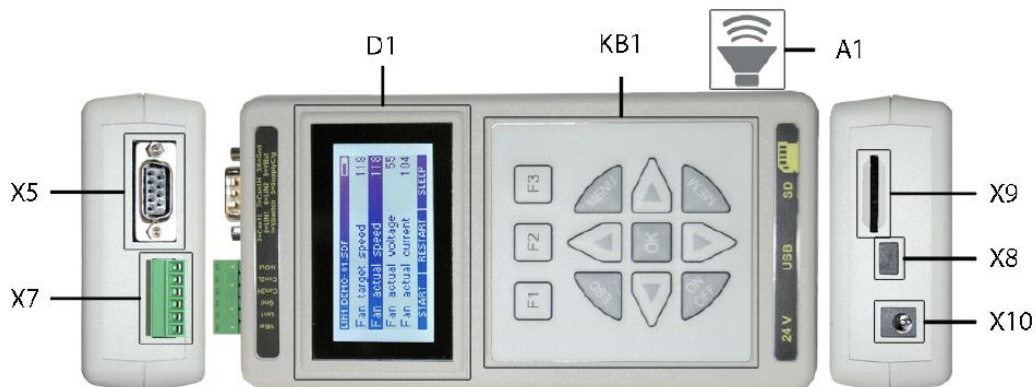


## 9 硬件

### 9.1 总览

以下图表显示了 HARP-5 的功能。将显示以下功能：

缩写名称	描述
X	访问各个接口的连接器。
KB	用于触发用户自定义操作的键盘。 用于浏览 HARP-5 菜单的键盘。
D	用于显示 LIN 和 CAN 总线信息并配置设备的显示器
A	一个可以由自定义事件触发的声音信号发生器。
SD	一个 SDHC 卡插槽。



缩写名称	类型	描述
X7	Sub-D-9 公头	LIN 和 CAN 总线连接器
X5	MC 1,5 / 6-ST-3,81 和 MCVR 1,5 / 6-ST-3,81 的插头	LIN 和 CAN 总线连接器
X10	NEB 21 R	逻辑电源和充电电源连接器



X8	USB 2.0 类型 B-Mini	PC 连接器
X9	SDHC 卡插槽	用于存储 SDF 和日志数据的 SD 卡
KB1	12 键键盘	浏览操作菜单的键盘
D1	图形液晶显示器	显示菜单和总线信息
A1	声音信号发生器	一个可以由自定义事件触发的声音信号发生器

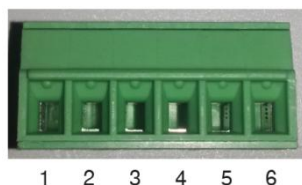
## 9.2 连接器

### 9.2.1 X5-LIN,CAN和IO

LIN 总线接口可通过 MC 1,5 / 3-ST-3,81 连接器使用。

HARP-5 的 LIN 总线接口可以在 8-26 VDC 的电压之间运行。

根据 ISO-11519, 该 CAN 总线接口可用作容错低速接口 (CAN-LS) 。



PIN 脚	信号	描述
X5-1	VCC	LIN 总线电源。该引脚同引脚 X7-9 相连。
X5-2	LIN-1	LIN 总线信号。该引脚同引脚 X7-8 相连。
X5-3	GND	LIN 总线接地。该引脚同引脚 X7-3 和 X7-6 相连。
X5-4	CAN-LS-H	低速 CAN 总线的 CAN 高接口。
X5-5	CAN-LS-L	低速 CAN 总线的 CAN 低接口。
X5-6	DigIO	数字 I/O

**CAUTION:** 将LIN总线电压保持在以下范围内：8-18 VDC。否则HARP-5可能会损坏，并可能对您造成伤害。

**CAUTION:**

在使用高于18 VDC的电压之前检查所有LIN总线节点规格。

如果超过18 VDC的电压用作LIN总线电源电压，则必须确保所有连接的节点都可以应对该电压。

由于LIN规范规定最大电压为18 VDC，因此有些节点可能无法在超过18 VDC的电压下正常工作。

**Advice:** LIN总线接口与逻辑电源电流是隔离的。

**Warning:** LIN总线电源必须由外部电源提供，并且在LIN通信期间不得中断。

**Warning:** LIN总线1和2默认是连接的，这允许使用HARP-5作为一个网关。通过设置系统变量

@@ SYSDIGOUT37，可以在运行时开关此行为。查看章节“@@ SYSDIGOUT”获取更多信息。

**Tip:** 可以通过设置和获取系统变量@@ SYSDIGOUT1来读写数字量I/O。查看章节“@@

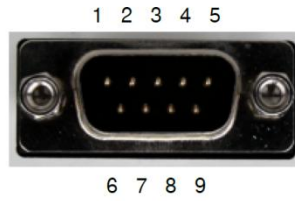
SYSDIGOUT”获取更多信息。

### 9.2.2 X7-LIN 和 CAN

LIN 和 CAN 总线接口可通过 Sub-D-9 公头连接器提供。

HARP-5 的 LIN 总线接口可以在 8-18VDC 的电压之间运行。

该 CAN 总线接口遵从 ISO-11898 协议，可用作高速接口（CAN-HS）。



PIN 脚	信号	描述
X7-1	VSwitch	一个可切换的输出，可以输出 LIN-Supply 电源。
X7-2	CAN-HS-L	高速 CAN 总线的 CAN 低接口。
X7-3	GND	LIN 和 CAN 总线接地。该引脚连接到引脚 X7-6 和 X5-3。
X7-4	LIN-2	LIN 总线第二个信号。
X7-5	SupplyConfig	一个引脚，用于配置 HARP-5 是否由 LIN-Supply 引脚供电。
X7-6	GND	LIN 和 CAN 总线接地。该引脚连接到引脚 X7-3 和 X5-3。
X7-7	CAN-HS-H	高速 CAN 总线的 CAN 高接口。
X7-8	LIN-1	LIN 总线信号。该引脚同引脚 X5-2 相连。
X7-9	LIN-Supply	LIN 总线电源。该引脚同引脚 X5-1 相连。

**Warning:** LIN总线电源必须由外部电源或集成12V发生器提供，并且在LIN通信期间不得中断。

**Warning:** LIN总线1和2默认是连接的，这允许使用HARP-5作为一个网关。通过设置系统变量 @@ SYSDIGOUT37，可以在运行时开关此行为。查看章节“@@ SYSDIGOUT”获取更多信息。

**Attention:** 不要在8-18 VDC的电压范围之外给LIN总线供电。

**Attention:** 如果超过18 VDC的电压用作LIN总线电源电压，则必须确保所有连接的节点都可以应对该电压。由于LIN规范规定最大电压为18 VDC，因此有些节点可能无法在超过18 VDC的电压下正常工作。

**Warning:** CAN接口不包含终端电阻，如果在总线安装中没有提供，则必须将其连接到外部。终端电阻的大小应为120Ω。

VSwitch, SupplyConfig 和 LIN-Supply 三个引脚处理 LIN 总线电源的产生，使用和输出。它们具有以下功能：

PIN 脚	描述								
LIN-Supply	该引脚可用于一下几种情况：								
	<table border="1"> <thead> <tr> <th>使用情况</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>输入</td> <td>LIN 总线由外部电源供电。默认情况下，HARP-5 也将通过此引脚供电。</td> </tr> <tr> <td>输出</td> <td>HARP-5 使用其内部集成的 12V 电源发生器为自己的 LIN 总线供电。其他 LIN 总线节点也可以通过此引脚供电。</td> </tr> <tr> <td>不使用</td> <td>如果没有使用任何 LIN 总线接口，则该引脚可以保持开路。</td> </tr> </tbody> </table>	使用情况	描述	输入	LIN 总线由外部电源供电。默认情况下，HARP-5 也将通过此引脚供电。	输出	HARP-5 使用其内部集成的 12V 电源发生器为自己的 LIN 总线供电。其他 LIN 总线节点也可以通过此引脚供电。	不使用	如果没有使用任何 LIN 总线接口，则该引脚可以保持开路。
	使用情况	描述							
	输入	LIN 总线由外部电源供电。默认情况下，HARP-5 也将通过此引脚供电。							
输出	HARP-5 使用其内部集成的 12V 电源发生器为自己的 LIN 总线供电。其他 LIN 总线节点也可以通过此引脚供电。								
不使用	如果没有使用任何 LIN 总线接口，则该引脚可以保持开路。								
<p><b>Warning:</b> 当使用 LIN-Supply 引脚输出内部产生的 LIN 总线电压时，独立于 SupplyConfig 引脚的状态自动禁用从 LIN 电源向 HARP-5 供电。</p>									
<p><b>Advice:</b> 可通过设置系统变量@@ SYSDIGOUT35 来操作 12V 电源发生器。 查看章节“@@ SYSDIGOUT”获取更多信息。</p>									

	可能值	描述						
	0	12V 电源发生器关闭						
	1 或更高	12V 电源发生器开启						
SupplyConfig	<p>该输入决定了 LIN-Supply 引脚上的电压是否可用于为 HARP-5 供电。</p> <p>如果该引脚开路（未连接），HARP-5 将使用外部 LIN 总线电压作为电源。这可以节省电池运行时间。</p> <p>如果引脚连接到 GND（通常在电缆内部短路），那么 HARP-5 不会由 LIN-Supply 引脚供电。</p> <p><b>Advice:</b> 通过设置系统变量@@ SYSCFG100，也可以关闭对 LIN-Supply 的供电。 ， 查看章节“@@ SYSCFG”获取更多信息。</p> <table border="1"> <thead> <tr> <th>可能值</th> <th>描述</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>设备通过 LIN 总线供电。</td> </tr> <tr> <td>1</td> <td>设备不通过 LIN 总线供电。</td> </tr> </tbody> </table>		可能值	描述	0	设备通过 LIN 总线供电。	1	设备不通过 LIN 总线供电。
可能值	描述							
0	设备通过 LIN 总线供电。							
1	设备不通过 LIN 总线供电。							
VSwitch	<p>该引脚通过内部开关连接至 LIN-Supply 引脚。</p> <p>如果激活，该引脚可通过它连接的内部或外部 LIN 总线电源为外部 LIN 总线节点供电。</p> <p>无论 LIN 总线电压是由外部电源还是集成 12V 发生器供电，此功能都可以工作。</p> <p><b>Advice:</b> 可通过设置系统变量@@ SYSDIGOUT36 来开关连接。查看章节“@@ SYSDIGOUT”获取更多信息。</p> <table border="1"> <thead> <tr> <th>可能值</th> <th>描述</th> </tr> </thead> <tbody> </tbody> </table>		可能值	描述				
可能值	描述							

	0	无可用电压
	1 或更高	可用 LIN 电源供电

### 9.2.3 X8-PC 连接器

该连接器型号是 USB type B-Mini。它用于将 HARP-5 连接到 PC。要使用此接口，必须在 PC 上安装 Baby-LIN 的 USB 驱动程序。



该连接器使用 USB type B-Mini 的默认引脚分配。

### 9.2.4 X9-SD 卡槽

此卡插槽支持 SD 卡。它用于在 HARP-5 上存储 SDF 并写入日志文件。



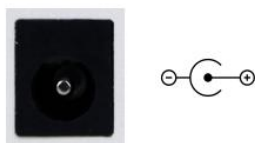
**Advice:** 支持以下 SD 卡类型

类型	类型说明	最大容量
SD	安全数字标准容量	2GB
SDHC	安全数字高容量	32GB

**Warning:** SD 卡必须格式化为 FAT16 或 FAT32。

### 9.2.5 X10-逻辑电源和充电电源

该连接器是 NEB 21 R。它用于为 HARP-5 供电和装载电池。



PIN 脚	极性	描述
外部连接	-	逻辑地
内部链接	+	正逻辑电源 (8-32 VDC) 。

**Tip:** 为了更好使用电池，需要接 24VDC 的电源电压。

**Attention:** 请勿供给 8-32 VDC 的电压范围之外的逻辑电源。

## 9.3 电源供电

HARP-5 可以由以下任意一种电源供电：

- 通过附带的电源适配器供电。支持的电源电压：8-32 VDC
- LIN-Supply 上的电源。支持的电源电压：8-32 VDC
- 通过内部电池供电。

HARP-5 的典型功耗为 60 mA @ 24 VDC。

### CAUTION:

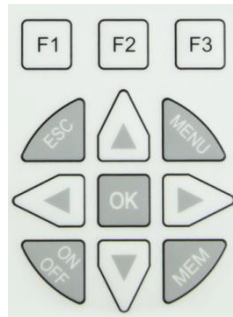
使用电池时，请注意以下几点，以防止造成伤害：

- 仅在规定的工作温度范围内操作设备。
- 插入电池时请注意正确的极性。

- 只能使用提供的电源适配器。
- 如果电池损坏，请勿触摸电池。
- 如果电池被浸湿，请勿触摸电池。
- 如果超过 4 周末使用 HARP-5，请取出电池。

## 9.4 KB1-键盘

HARP-5 的键盘具有 12 个键用于浏览菜单和配置设备。以下按键可用：



按键	功能
F1,F2,F3	这些按键可以触发软件定义的菜单选项。每个菜单最多拥有 3 个内容相关的选项。
上,下,左,右	这些按键用于浏览菜单并更改值。
ESC	这个按键用于退出菜单。
MENU	这个按键用于打开设置菜单。
OK	这个按键用于用户确认。
MEM	这个按键用于保存更改。
ON/OFF	这个按键用于开关 HARP-5。



## 9.5 D1-显示屏

HARP-5 的图形 LCD 显示屏具有 128x64 像素和可切换的背光灯。它用于辅助配置 HARP-5 并显示 LIN-和 CAN 总线信息。

背光亮度可调，并且可以在一定时间后会自动关闭。



**Advice:** 可以通过设置系统变量@@ SYSDIGOUT34 来关闭 LCD 背光。查看章节“@@ SYSDIGOUT”获取更多信息。

## 9.6 硬件适配

### 9.6.1 如何更换充电电池

本节将指导您如何更换供应 HARP-5 的充电电池。

在开始操作之前，请完整阅读本节内容。确保你了解如何操作，并拥有所需的所有工具和材料。

**NOTICE:** HARP-5 可能会因静电(ESD)损坏。在打开 HARP-5 外壳并触摸电路板之前，请遵守静电保护措施！理想情况下，您可以使用 ESD 垫和 ESD 腕带。

请您在开始操作之前准备以下工具和材料：

- 匹配的硬币或螺丝刀。

- 全新的 1 组 6 个可充电电池（2600-2700 mAh 1.2V NiMH AAMignon）。

**Advice:** 可充电电池类型必须是 2600-2700 mAh 1.2V NiMH AAMignon。允许使用其他电池容量，但会影响运行和充电时间。另外，电池必须能够以 1A 的电流充电。我们建议您从我们购买可充电电池，因为我们对电池进行过测试。

将 HARP-5 倒置，然后拧下 4 个电池盖螺丝。



取下电池盖。



现在您可以取出旧电池并插入六个新的可充电电池。

**Version incompatibility:** 插入电池时请注意正确的极性。电池舱内有极性标识。

**Advice:** 请根据当地的法律法规安全，妥善地回收或处理电池。

现在您可以通过以下步骤重新组装 HARP-5:

- 盖上电池盖。
- 拧回 4 颗电池盖螺丝。

## 9.6.2 如何更换实时时钟(RTC)电源

### 9.6.2.1 简介

为 HARP-5 的 RTC 供电的电池通常可连续使用 9 年以上。一旦其电量用完，HARP-5 将不再能记忆日期和时间。这时就是更换 RTC 电池的时候了。

本节将指导您如何更换 RTC 的电池。

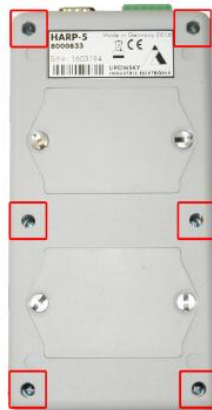
### 9.6.2.2 所需工具和材料

请您在开始操作之前准备以下工具和材料:

- PH1 螺丝刀
- 全新的 3V CR2430 纽扣电池

### 9.6.2.3 拆卸

将 HARP-5 倒置，然后拧下 6 个外壳螺丝



小心打开盒子，直到拿下带状电缆。然后将外壳放在安全的地方。



抓住电路板的两侧，小心地将其拉起，直至与盒子分离。

**Advice:** 顶部和底部连接器面板未连接到电路板。它们依靠盒子固定，此时可能会脱落。

**Version incompatibility:** 在电路板和正面外壳之间存在另一条电缆。因此，不要将电路板拉得太远。



放下电路板并拿住盒子的正面。



现在您应该可以看到右侧显示屏下面的电池座内的纽扣电池。

**Advice:** 如果您不小心从电路板上拔下柔性扁平电缆，则可以简单地将其重新插入。这可能会  
有点烦琐，您可以考虑使用钳子。但请注意不要损坏电缆。

#### 9.6.2.4 更换电池

现在您可以卸下旧的纽扣电池并更换一个新的 3V CR2430 纽扣电池。

插入电池时请注意正确的极性，弹簧夹需要接触电池的正极。

**CAUTION:** 使用电池时，请注意以下几点，以防止造成伤害

- 仅在规定的工作温度范围内操作设备。
- 插入电池时请注意正确的极性。
- 只能使用提供的电源适配器。
- 如果电池损坏，请勿触摸电池。
- 如果电池被浸湿，请勿触摸电池。
- 如果超过 4 周末使用 HARP-5，请取出电池。

**Advice:** 请根据当地的法律法规安全，妥善地回收或处理电池。

**Version incompatibility:** RTC 的日期和时间存储在 SBC 板的处理器中。它由 UPS 板的电池供电。如果该电源中断，则 RTC 会被复位。可能是以下原因之一

- SBC 板被移除。
- UPS 板被移除。
- 从 UPS 板上取下 RTC 电池。
- RTC 电池没电了。

如果发生这些事件之一，则必须重新设置 RTC 日期和时间。

#### 9.6.2.5 组装

现在您可以通过以下步骤重新组装 HARP-5:

- 抓住电路板的两侧，并将顶部和底部连接器面板推回到盒子正面。
- 将外壳背面的带状电缆插回电路板。
- 将外壳放上。
- 拧回 6 个螺丝。

## 10 固件

### 10.1 固件升级

#### 10.1.1 介绍

固件存储在 HARP-5 的闪存中，并可以通过 PC 轻松编程。更新过程使用 ISP（系统内编程）操作并可直接应用。

我们一直致力于产品改进，所以固件会定期更新。这些更新提供新功能和解决已知问题，这些问题由我们的内部测试发现，或者由早期版本的客户反馈。所有固件更新都以同样的方式完成，更新后的 HARP-5 依然可以配套旧版 LINWorks 使用，因此更新 Baby-LIN 固件并不意味着您必须更新您的 LINWorks 安装。我们强烈建议您始终将 HARP-5 更新为最新的可用固件版本。

### 10.1.2 更新需要的软件

需要下载以下内容更新 HARP-5 的固件：

下载文件	描述
BabyLinDriverSetup.exe	这是 Baby-LIN 驱动程序，更新固件必须安装。
Harp_4_5_FirmwarePackage.zip	该软件包包含 HARP-5 的固件以及更新工具 BLProg。

### 10.1.3 更新固件

要更新 HARP-5 的固件，您必须执行以下步骤：

- 如果您尚未使用过 HARP-5，请安装驱动程序。
- 将 HARP-5 与 PC 连接。

**Tip:** 如果与 PC 连接的设备除了 HARP-5 以外，还有其他 Baby-LIN 产品，您将在稍后被要求选择其中一个。这很容易导致混淆和更新错误的设备。因此，我们建议在固件更新期间只连接一个 Baby-LIN 产品。

- 在 HARP-5 上选择固件更新模式：
- 打开 HARP-5。
- 按“MENU”键进入配置菜单。
- 选择“固件更新”选项。
- 按“OK”键确认固件更新模式。
- 显示屏关闭。
- 解压缩固件安装包。
- 启动“blprog.exe”。一个提示窗口将会被打开并指导您完成整个过程。
- 如果除了与 PC 连接的 HARP-5 之外还有其他 Baby-LIN 产品，则会询问您要更新哪个 Baby-LIN 产品。

**Tip:** 如果您不知道哪个虚拟 COM 端口对应于您的 HARP-5，则可以按“CTRL + C”停止固件更新。然后，您可以断开其他 Baby-LIN 产品，直到只连接了需要更新的 HARP-5，并重启“blprog.exe”了。

- 按“y”键确认正确的固件并开始更新闪存。
- 等到 LED 不再闪烁。
- 闪烁完成后，按“Enter”键退出工具。

## 10.2 PC 模式

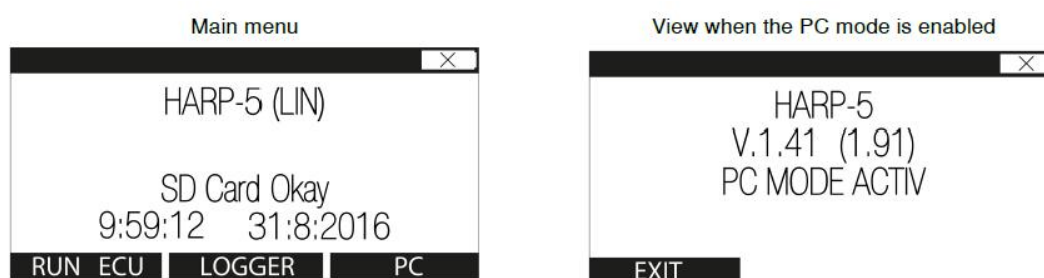
### 10.2.1 PC 模式介绍



PC 模式使 HARP-5 能够像 Baby-LIN 产品系列的其他产品一样与 PC 通信。这意味着您可以使用 SimpleMenu 及其所有功能以及使用 Baby-LIN-DLL 编写自己的应用程序，另外更新固件也是必要的。

## 10.2.2 启用 PC 模式

要启用 HARP-5 的 PC 模式，首先需要在开机状态。如果您不在主菜单，请反复按 ESC 退出至主菜单。然后按“F3”进入 PC 模式。



如果 PC 模式当前已启用，只需按“F1”键退出 PC 模式。

## 10.3 独立运行模式和自启动

### 10.3.1 允许独立运行模式

HARP-5 可以在没有 PC，PLC 或操作者的情况下独立运行。要启用此模式，需要满足几个要求：

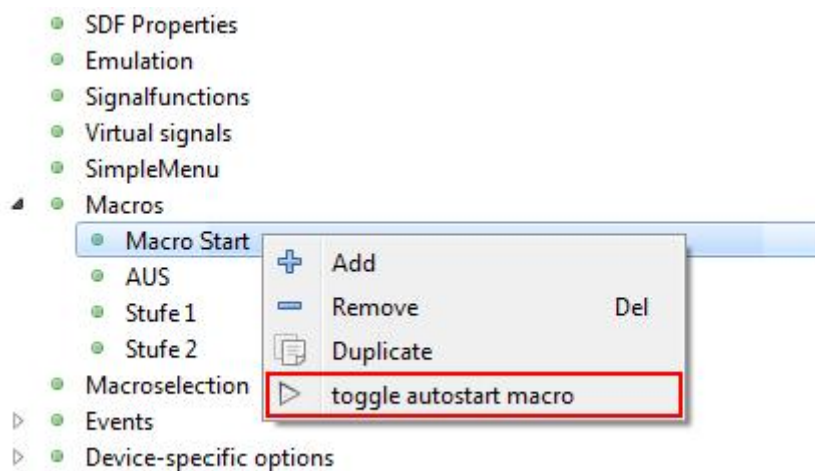
- 您需要一个带有被标记为“自动启动”的宏的 SDF 文件。
- 设备需要被配置为上电时会运行永久储存在设备的 SDF 文件中的被标记为自动运行的宏。

### 10.3.2 配置自动运行的宏

您可以将 SDF 中的宏标记为自动启动宏。这意味着这个宏在加载 SDF 时自动启动。通常这个宏将启动 LIN 或 CAN 总线通信并执行必要的初始化工作。

要将宏标记为自动启动的宏，需要按照以下步骤操作。

- 使用 SessionConf 打开您的 SDF 文件。
- 如果此 SDF 文件中尚未有要用作自动启动宏的宏，请创建一个。
- 右键单击宏并选择“toggle autostart marco”。该宏现在将具有“[autostart]”标记。



**Advice:** 请注意，只有一个宏可以标记为自动启动宏。

**Tip:** 您可能想要在宏中启动 LIN 或 CAN 总线，因为它不会通过加载 SDF 而自动启动。

### 10.3.3 SDF 永久储存

存储在 HARP-5 的 SDHC 卡上的所有 SDF 都被永久保存。无需以特殊方式配置设备，SDF 选项

“Store SDF in device persistently”对 HARP-5 没有影响。

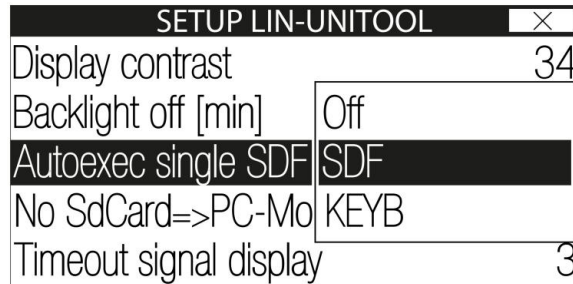
### 10.3.4 配置设备自动加载一个 SDF 文件并运行一个宏

在配置设备以自动加载 SDF 并启动宏之前，必须确保 SDHC 卡上只存储了一个 SDF 文件。这个

SDF 文件需要一个标记为自动启动的宏。

启用 HARP-5 上的自动启动功能开关。按“MENU”键并向下滚动到“Autoexec single SDF”选项。

按下“确定”键并选择“SDF”可以在 SDF 菜单中开始或者“KEYB”在键盘菜单中开始。



按“ESC”退出菜单并按“MEM”保存更改。如果现在关闭 HARP-5，然后再次打开，则会直接执行自动启动的宏，并显示 SDF 或键盘菜单。

**Warning:** 如果您在 SDHC 卡上存储了多个 SDF，则不会自动加载任何 SDF。

**Warning:** 只有宏被标记为自动启动，才会在开启设备时自动启动这个宏。

## 10.4 记录

### 10.4.1 简介

HARP-5 支持记录 LIN 和 CAN 总线上的帧。记录总线数据的方法如下：

- 如果 HARP-5 连接到 PC，则可以使用 SimpleMenu 记录总线数据。
- 关于使用 SimpleMenu 的记录功能可以查看章节“SimpleMenu”获取更多信息。
- 记录数据可以无需 PC 直接写入 SD 卡或通过 CAN-HS 接口传输。
- 记录数据可以通过 USB 连接发送到 PC 上的自定义应用程序。

### 10.4.2 配置和启用记录功能

记录功能的配置取决于 HARP-5 正在使用的模式。

配置	SDF 模式	纯监控模式
开始, 停止或暂停记录	设置以下系统变量: “@@SYSLOGCONTROL”	开始记录而不加载 SDFile。从主菜单中选择记录器, 配置记录器并按下开始。
记录数据的来源通道	源通道是间接选择的: 记录系统变量单独存在于一个部分。如果某个部分配置并启动记录器, 则此部分映射到的通道将成为源通道。查看章节“通道和部分”了解更多信息。	在 Logger 菜单中的 Source 设置。
记录文件格式	设置以下系统变量: “@@SYSLOGFORMAT”	在 Logger 菜单中的 Format 设置。
记录数据的目标	设置以下系统变量: “@@SYSLOGTARGET”	在 Logger 菜单中的 Target 设置。
记录的时间戳的分辨率	设置以下系统变量: “@@SYSLOGTIMEBASE”	记录的时间戳的分辨率总是以 $\mu\text{s}$ 为单位。
最大记录文件体积	设置以下系统变量: “@@SYSLOGFILESIZE”	最大记录文件大小始终为 4096 MB。

记录菜单允许为每个源通道配置一个记录器, 并可以配置以下选项:

选项	值	描述
Source	LIN1	第一路 LIN 通道将成为记录数据的来源。

	LIN2	第二路 LIN 通道将成为记录数据的来源。
	CAN1	第一路 CAN 通道将成为记录数据的来源。
	CAN2	第二路 CAN 通道将成为记录数据的来源。
Target	USB	请查看“USB 连接”一章，获取更多信息。
	SDcard	请查看“SD 卡”一章，获取更多信息。
	CAN1	请查看“高速 CAN 接口”一章，获取更多信息。
Format	Binary	请查看“二进制格式”一章，获取更多信息。
	Text	请查看“ASCII 格式”一章，获取更多信息。
	Bin-Old	这是一种旧的二进制格式，已经不再被使用。
	Text-Old	这是一种旧的 ASCII 格式，已经不再被使用。
Mixed	No	如果目标不是 SD 卡，则此选项强制为“否”。
	Yes	如果目标是 SD 卡，则将为已激活此功能的所有记录器创建一个公共的输出文件。该公共文件的名称将以 MIX 做后缀。

### 10.4.3 混合模式

在手持模式下使用 HARP-5 时，可以从菜单配置记录。该菜单允许配置混合模式。

通常每个记录器都会将来自其通道的信息记录到单独的文件中。如果激活混合模式，则所有数据都将记录在同一个记录文件中。这个记录文件的文件名将有一个“MIX”后缀，而不是通道后缀。

如果使用混合模式，所有记录器将使用相同的格式和目标设置。

**Warning: 即使混合模式被激活，每个记录器仍然必须单独启动**

## 10.4.4 记录文件目标

### 10.4.4.1 概述

HARP-5 可以将记录数据写入以下目标：

目标	描述
SimpleMenu	可以使用“SimpleMenu”查看记录数据并将其储存成文件。
USB 连接	记录数据通过 USB 连接发送到 PC。
SD 卡	记录数据作为文件存储在 SD 卡上。
高速 CAN 接口	记录数据通过高速 CAN-1 接口重新发送。

### 10.4.4.2 SimpleMenu

“SimpleMenu”可以显示帧监视器中 HARP-5 的总线通信。另外，该通信数据可以以“二进制格式”写入记录文件。

### 10.4.4.3 USB 连接

记录数据通过“X8-PC 连接器”接口连接 USB 传输。Baby-LIN 驱动程序将 HARP-5 连接显示为虚拟 COM 端口。您可以使用终端程序在 PC 上接收和记录这些数据。使用以下设置连接到

HARP-5：

- COM 端口：HARP-5 的虚拟 COM 端口。
- 波特率：取决于 HARP-5 115,200 Bd 或 921,600 Bd 的设置。
- 数据位：8
- 停止位：1
- 奇偶校验：无

- 流量控制：无

**Warning:** 如果 HARP-5 的 PC 模式没有在设置中被激活。请查看“PC 模式”一章获取更多信息。

#### 10.4.4.4 SD 卡

日志数据作为文件存储在 Micro SD 卡上。存储日志文件的文件夹是 HARP-5\Logg\。

文件名将包含记录文件开始的日期，时间和通道。

**Advice:** 支持以下 SD 卡类型：

类型	类型说明	最大容量
SD	安全数字标准容量	2GB
SDHC	安全数字高容量	32GB

**Warning:** SD 卡必须格式化为 FAT16 或 FAT32。

请查看“X9 - SD 卡插槽”一章获取更多信息。

#### 10.4.4.5 高速 CAN 接口

接收到的 LIN 帧可以通过 CAN-HS 接口重新发送为 CAN 帧。为此，您可以编辑传输的 CAN 帧

的以下属性：

属性	描述
ID 偏移量	该偏移量被添加到 LIN 接收帧的帧 ID 中。得到的结果将是传输的 CAN 帧的帧 ID。

29 位 ID	该值决定 CAN 帧是使用 11 位 ID 还是 29 位 ID 传输。
---------	--------------------------------------

请查看“X7 - LIN 和 CAN”一章获取更多信息。

**警告：**只有来自 LIN 总线的帧才可以通过高速 CAN 接口重新发送。无法从低速 CAN 接口重新发送帧。如果使用“混合模式”，则无法设置这个选项。

## 10.4.5 记录数据格式

### 10.4.5.1 格式一览

格式	描述	帧大小
二进制格式	这是 HARP-5 当前固件的默认格式。这种格式使用专用的二进制数据格式来存储数据。它针对速度和文件大小进行了优化。这种格式的文件可以使用“LogViewer”转换为其他格式。	24 字节
ASCII 格式	这种格式是可读的 ASCII 格式，可以由一些第三方软件处理。	88 字节
旧的二进制格式	这是一种旧的二进制格式，已经不再被使用。	108 字节
旧的 ASCII 格式	这是一种旧的 ASCII 格式，已经不再被使用。	65 字节

帧大小是指日志文件中 8 字节帧所需的大小。额外的头文件和其他信息会进一步增加日志文件的大小。

### 10.4.5.2 二进制格式



这种格式使用专有的二进制数据格式来存储数据。它针对速度和文件大小进行了优化。这种格式的文件可以使用“LogViewer”转换为其他格式。

### 10.4.5.3 ASCII 格式

这种格式是一种可读的 ASCII 格式。它由标题，记录的帧数据和注释组成。它可以被许多第三方软件处理。ASCII 格式具有以下结构：

```

date Fri May 12 13:38:13 2017
base hex timestamps absolute
internal events logged
// version HARP-4 V.1.43 Build1
2.788508 Li      11 Tx  1 00      checksum = ff   CSM = classic
3.288498 Li      12 Rx  8 43 a1 16 d0 a7 53 29 00 checksum = 10   CSM = classic
3.538493 Li      13 Rx  0      NodeResponseMissing
3.788488 Li      11 Tx  1 01      checksum = fe   CSM = classic
4.288531 Li      12 Rx  8 4d a6 19 d3 a9 54 2a 00 checksum = f6   CSM = classic
4.538525 Li      13 Rx  0      NodeResponseMissing
4.788519 Li      11 Tx  1 02      checksum = fd   CSM = classic
5.288509 Li      12 Rx  8 56 ab 1d d5 ab 55 2a 00 checksum = df   CSM = classic
5.538504 Li      13 Rx  0      NodeResponseMissing
5.788499 Li      11 Tx  1 03      checksum = fc   CSM = classic
6.288489 Li      12 Rx  8 60 b0 20 d8 ad 56 2b 00 checksum = c6   CSM = classic
6.538536 Li      13 Rx  0      NodeResponseMissing
6.788531 Li      11 Tx  1 04      checksum = fb   CSM = classic
7.288521 Li      12 Rx  8 6a b5 23 da af 57 2b 00 checksum = af   CSM = classic
7.538515 Li      13 Rx  0      NodeResponseMissing
7.788510 Li      11 Tx  1 05      checksum = fa   CSM = classic

```

以第一帧为例进行解释

数据	解释
2.788508	这是一个时间戳。从 HARP-5 上电后开始计时。
Li	总线标识符。
11	帧 ID。
Tx	帧传输方向。
1	帧长度。

00	帧的所有数据字节。
checksum = ff	帧的校验和。
CSM = classic	帧使用的校验和类型。

请使用 // 进行注释。

## 10.5 设置 HARP-5

设置项	可能的值	描述
设置语言	<ul style="list-style-type: none"> <li>● 英语</li> <li>● 德语</li> </ul>	在这里可以更改菜单使用的语言。
屏幕亮度	0 到 20	<p>在这里你可以改变显示器的亮度。</p> <p>Tip 亮度越高，电池消耗越快。</p>
屏幕对比度	30 到 60	在这里你可以改变显示器的对比度。
背光灯关闭时间(min)	<ul style="list-style-type: none"> <li>● 常亮</li> <li>● 0 到 20</li> </ul>	<p>在这里您可以设置显示屏的背光自动关闭的时间。</p> <p>Tip 背光打开时间越长，电池消耗越快。</p>
自动运行单一 SDF	<ul style="list-style-type: none"> <li>● 关闭</li> <li>● SDF</li> <li>● 键盘</li> </ul>	<p>在这里您可以设置 SDF 文件的启动方式。</p> <p>含义取决于以下条件：</p> <ul style="list-style-type: none"> <li>• SD 卡上只有一个 SDF，并且它包含一个自动启动宏：</li> </ul>

		<p>°关闭: SDF 不会自动启动</p> <p>°SDF: 在 SDF 菜单中自动启动 SDF。</p> <p>°键盘: 在键盘菜单中自动启动 SDF。</p> <ul style="list-style-type: none"> <li>• SD 卡上有多个 SDF 文件或 SDF 文件没有自动启动宏:</li> </ul> <p>°关闭: 用户决定每次启动, SDF 启动的模式。</p> <p>°SDF: SDF 在 SDF 菜单中启动。</p> <p>°键盘: SDF 在键盘菜单中启动。</p> <p>查看章节“独立模式和自动启动”以获取更多信息。</p>
无 SD 卡时自动切换到 PC 模式	<ul style="list-style-type: none"> <li>● 关</li> <li>● 开</li> </ul>	<p>如果没有插入 SD 卡, 您可以在这里配置 HARP-5 是否会自动切换到 PC 模式。查看章节“PC 模式”了解更多信息。</p>
超时信号显示	<ul style="list-style-type: none"> <li>● 关闭</li> <li>● 1 到 60</li> </ul>	<p>在这里, 您可以配置一个超时, 如果该信号在该时间内在总线上没有出现, 将会重置该信号值。超时时间以秒为单位。</p>
通过 24V 电压自动开启	<ul style="list-style-type: none"> <li>● 关</li> <li>● 开</li> </ul>	<p>如果连接了 24V 电源, 在这里将配置 HARP-5 是否自动打开。</p> <p>查看章节“电源”了解更多信息。</p>
记录器波特率	<ul style="list-style-type: none"> <li>● 115.200</li> <li>● 921.600</li> </ul>	<p>在这里, 您可以配置波特率, 如果记录器选择 USB 作为目标, 将使用该波特率。</p>

		查看章节“USB 连接”了解更多信息。
电池充电	<ul style="list-style-type: none"> <li>● 否</li> <li>● 是</li> </ul>	<p>您可以在这里配置, HARP-5 的电池能否通过 24V 电源充电。</p> <p>查看章节“USB 连接”了解更多信息。</p>
断开 LIN 总线	<ul style="list-style-type: none"> <li>● 否</li> <li>● 是</li> </ul>	<p>在这里你可以配置, 如果两个 LIN 总线接口连接。如果要单独使用每个 LIN 总线接口, 请选择是; 如果您想要连接两个 LIN 总线接口, 请选择是。</p>
设置时间和日期	<p>子菜单有以下选项:</p> <ul style="list-style-type: none"> <li>● 夏令时</li> <li>● 年</li> <li>● 月</li> <li>● 日</li> <li>● 小时</li> <li>● 分钟</li> </ul>	<p>HARP-5 具有实时时钟。在这个子菜单中, 您可以编辑该时钟的日期和时间。</p>
夏令时	<ul style="list-style-type: none"> <li>● 关</li> <li>● 开</li> </ul>	<p>HARP-5 具有实时时钟。这个选项可以为您的时钟激活夏令时设置。夏令时配置与设置日期和时间子菜单中的夏令时效果相同。</p>
选择 SDF	<ul style="list-style-type: none"> <li>● 文件名</li> <li>● 部分名称</li> </ul>	<p>在此您可以配置, “ECU 运行菜单”将显示 SDF 的文件名还是第一部分的名称。</p> <p>Version incompatitbility</p>

		<table border="1"> <thead> <tr> <th>SDF 版本</th> <th>相关部分</th> </tr> </thead> <tbody> <tr> <td>SDF-V2</td> <td>只有一个 LIN 部分。</td> </tr> <tr> <td>SDF-V3</td> <td>将显示设备部分的名称。</td> </tr> </tbody> </table> <p>因此，如果将 SDF-V2 转换为 SDF-V3，则所示的部分名称可能会改变。</p>	SDF 版本	相关部分	SDF-V2	只有一个 LIN 部分。	SDF-V3	将显示设备部分的名称。
SDF 版本	相关部分							
SDF-V2	只有一个 LIN 部分。							
SDF-V3	将显示设备部分的名称。							
固件升级	一个对话框会弹出并向您确认	<p>使用此选项，您可以设置 HARP-5 的固件模式。</p> <p>Attention</p> <p>只能进入此模式更新 HARP-5 的固件。</p> <p>请查看“固件更新”一章获取更多信息。</p>						
SD 卡格式化	一个对话框会弹出并向您确认	<p>使用此选项，您可以格式化 SD 卡。</p> <p>Attention</p> <p>SD 卡上的数据将会丢失。在格式化之前，您应该对 SD 卡进行备份。</p>						
CAN 终端	一个对话框会打开	<p>该功能可用于基本的 CAN 调试。功能非常有限。</p> <p>打开时，窗口将显示高速 CAN 接口上 ID 为 0x100（十进制：256）的接收帧的所有数据字节，显示为 ASCII 字符。高速 CAN 的比特率是 250 kbit/s。</p>						

## 10.6 手动控制

### 10.6.1 简介

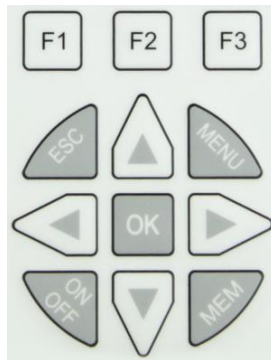
HARP-5 具有图形 LCD 显示屏和带 12 个按键的小键盘。这可以用手动控制来监视和控制 LIN 或 CAN 总线。

以下章节将会描述“键盘布局”以及 HARP-5 可以显示的不同“菜单”。

最后，“导航”一章为您提供了一个概述，如何浏览不同的菜单。

### 10.6.2 键盘布局

HARP-5 有一个 12 键的键盘。其中 3 个可以用作功能键。



按键	名称	功能
	功能键: F1,F2,F3	根据当前显示的菜单，这些功能键将具有不同的功能。
	ESC	ESC 键通常用于退出菜单或回到上一菜单。  Tip 如果您不知道现在在哪个菜单中，请连续按 ESC 键，最终您将返回主菜单。

	MENU	菜单键将根据当前菜单打开另一个菜单。
	OK	确定键用于确认信息和输入对话框。
	ON/OFF	开/关键用于 HARP-5 开关机。
	MEM	MEM 键用于保存更改的设置。
	UP DOWN	向上/向下键用于浏览列表菜单。
	LEFT RIGHT	左/右键用于更改值。

### 10.6.3 菜单

#### 10.6.3.1 引导屏幕

设备启动时，启动界面会短时间显示。在此期间按下 ESC，启动屏幕将保持可见状态，直到 ESC 被释放。

引导屏幕向您显示以下信息：

- 产品名称：HARP-5。
- 激活的总线接口。
- 主要固件版本和括号内的 CPU 固件充电。
- 硬件版本和已安装的 Display-I-O 连接板的版本。
- 激活的选项：

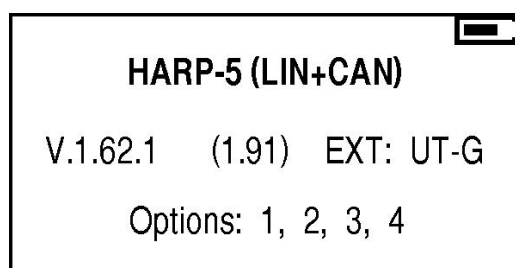
0 支持 SDF-V3

1 激活的高速 CAN 通道

2 激活的低速 CAN 通道

3 激活的第二个 LIN 通道

4 支持巨型帧

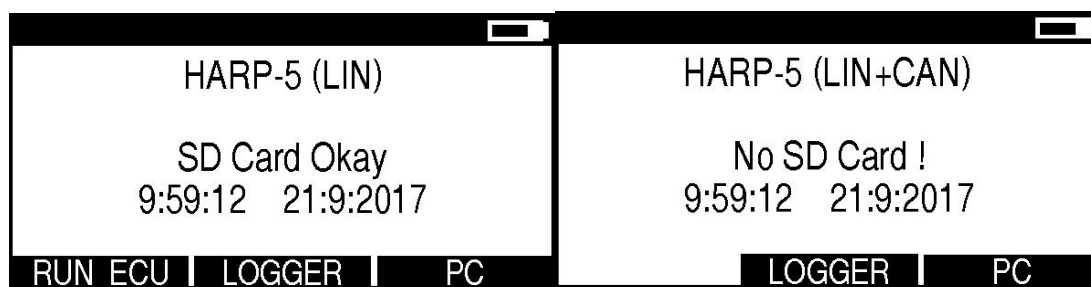


按键	功能按键	功能
ESC		ESC 键按下时，引导屏幕将不会关闭。

### 10.6.3.2 主菜单

主菜单显示以下信息：

- 产品名称：HARP-5。
- 激活的总线接口。
- SD 卡的当前状态。
- 当前时间和日期。



有 SD 卡，没激活 CAN

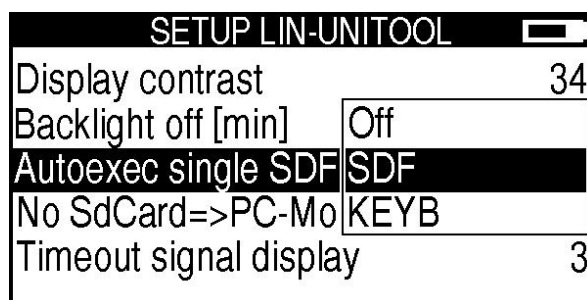
无 SD 卡，已激活 CAN



按键	功能按键	功能
F1	RUN ECU	打开“ECU 运行菜单”。 Advice 插入了可读的 SD 卡后，此项目才可用。
F2	LOGGER	打开“Logger 菜单”。
F3	PC	进入“PC 模式”并打开“PC 模式菜单”。
MENU		打开“设置菜单”。

### 10.6.3.3 设置菜单

设置菜单显示 HARP-5 固件的所有选项列表。请查看“HARP-5 的设置”一章获取更多信息。



按键	功能按键	功能
ESC		回到“主菜单”。 如果您尚未保存所做的更改，则在您离开设置菜单之前，会出现一个对话框，询问您是否要保存更改。按 MEM 键保存更改，或按 ESC 键丢弃更改。
MEM		保存所做的更改。

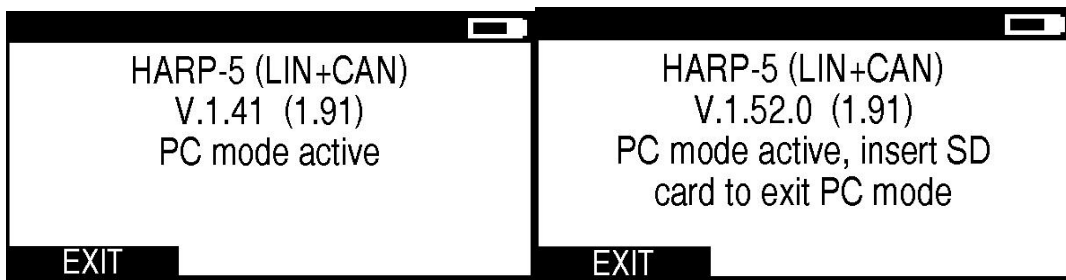
UP DOWN		移动光标。
OK LEFT RIGHT		打开用于编辑当前选项的对话框。

#### 10.6.3.4 PC 模式菜单

HARP-5 处于 PC 模式时，可以使用 PC 模式菜单。

PC 模式菜单显示以下信息：

- 产品名称：HARP-5。
- 已激活的总线接口。
- 主固件版本和充电 CPU 固件版本(括号内)。
- HARP-5 处于 PC 模式的提示。如果 HARP-5 是因为没有插入 SD 卡而自动使用 PC 模式的，那么您只有插入 SD 卡后，才能退出 PC 模式。



按键	功能按键	功能
F1	EXIT	退出 PC 模式并回到“主菜单”。

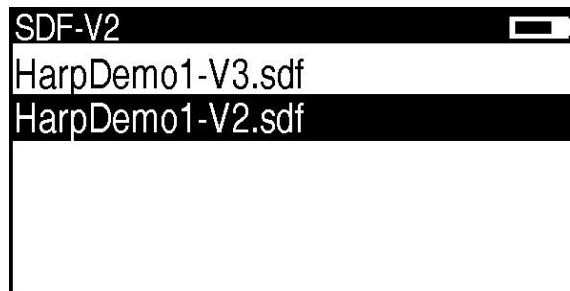
ESC		<p>Advice</p> <p>如果激活设置 No SD card =&gt; PC mode, 则只有插入可读 SD 卡后, 才能使用此项目。请查看“HARP-5 的设置”一章获取更多信息。</p>
-----	--	---

请查看“PC 模式”一章获取更多信息。

### 10.6.3.5 ECU 运行菜单

主菜单显示以下信息：

- 当前所选项目的类型。
- 插入的 SD 卡的文件夹树, 包括所有 SDF 文件。



按键	功能按键	功能
ESC		回到“主菜单”。
MENU		
UP		移动光标。
DOWN		
OK		选择光标悬浮的条目。
LEFT		如果所选条目是 SDF 文件, 则加载并显示“SDF 开始菜单”。

RIGHT		如果所选条目是文件夹，则显示该文件夹的内容。
-------	--	------------------------

**Advice:** 加载 SDF 文件后，将检查是否有宏被标记为自启动宏。如果有，则首次显示“SDF 菜单”或“键盘菜单”时将执行自启动宏。

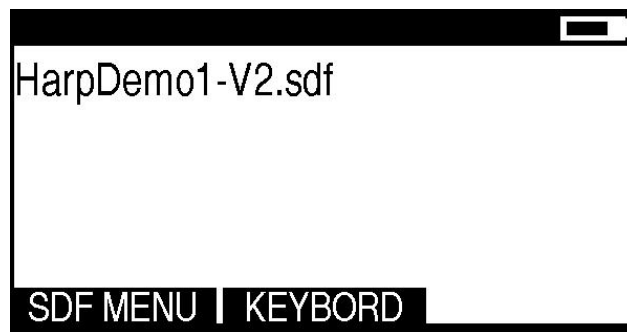
### 10.6.3.6 SDF 开始菜单

SDF 开始菜单显示以下信息：

- 所选的 SDF 文件名。
- SDF 文件第一部分的名称（需要在 HARP-5 中先进行设置）。
- 可能的启动选项：

°SDF MENU：显示 SDF 的 GUI 元素。

°KEYBOARD：显示 SDF 中 HARP-5 设备特定选项中设置的显示值和键盘标签。



按键	功能按键	功能	
F1	SDF MENU	启动 SDF 文件并转到“SDF 菜单”。	
F2	KEYBOARD	SDF 文件	功能
		SDF 文件只有一个部分	启动 SDF 并转到“键盘菜单”。

		SDF 文件有多个部分	启动 SDF 并转到“部分选择菜单”。
ESC		回到“主菜单”。	

**Warning:** 总线启动后，将会持续运行，直到您按下 ESC 键。总线和所有与 SDF 相关功能将停止。

### 10.6.3.7 SDF 菜单

SDF 菜单显示以下信息：

- 通道和 SDF 文件/第一部分的名称。
- 一个滚动列表，显示包含在 SDF 文件中的 GUI 元素中定义的元素。

**Advice:** 如果加载的 SDF 文件具有多个部分，则将显示每部分在 GUI 元素列表内的所有元素。

- 控制总线的按钮。

LINO:HARPDEMO1-V2.SDF		LINO:HARPDEMO1-V2.SDF	
ActSpeed	OFF	Fan target speed	47.2%
VehicleSpeed	0	Fan actual speed	236 rpm
MsgCounter	0	Fan actual voltage	13.75 V
ParkPosition	0	Fan actual current	49 A
START   RESTART   SLEEP		START   SLEEP	

按键	功能按键	功能
F1	START	Advice 只在总线没有运行时可用。

		总线还没有运行时，将启动总线。所有信号都将重置为默认值。
	STOP	Advice 只在总线运行时可用。 将停止总线。
F2	RESTART	Advice 只在总线没有运行时可用。 总线没有运行时，可以重启总线。重启和开始的区别在于，它不会信号将重置为其默认值。
F3	SLEEP	Advice 只在总线运行时可用。 向总线发送一条睡眠指令。
ESC		返回“SDF 开始菜单”。
MENU		打开“记录菜单”。
ON/OFF		切换信号表示方法： • 十进制原始信号值 • 编码后的信号值
UP DOWN		移动光标。
OK LEFT RIGHT		编辑当前选定的条目。 Warning 如果信号在该部分的 GUI 元素中添加为编辑信号，则您只能

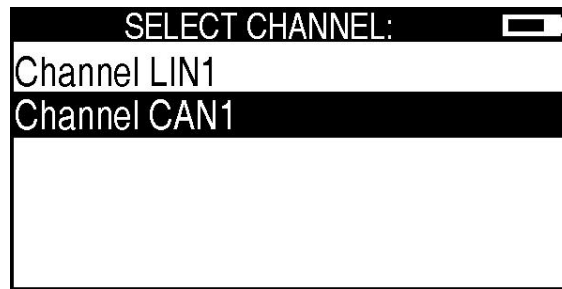
		编辑信号。
--	--	-------

**Advice:** 如果总线启动，即使按 ESC 键进入“SDF 开始菜单”，它也会继续运行。再次在“SDF 开始菜单”中按 ESC，总线才会停止。

### 10.6.3.8 部分选择菜单

部分选择菜单显示以下信息：

- 显示所有可用通道

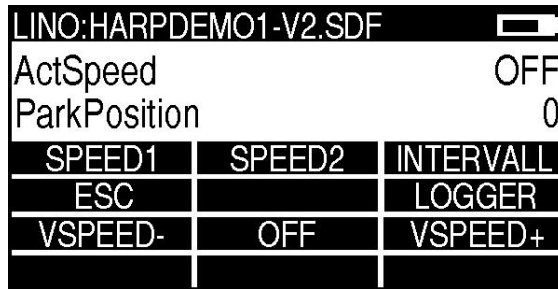


按键	功能按键	功能
ESC		返回“SDF 开始菜单”。
UP DOWN		移动光标。
OK LEFT RIGHT		显示该部分在键盘菜单中设置的变量，然后打开“键盘菜单”。

### 10.6.3.9 键盘菜单

键盘菜单显示以下信息：

- 通道和 SDF 文件或 LIN/CAN 部分名称。
- 一个固定列表用于显示在 SDF 文件中设备特定选项中定义的元素。
- 最多可设置 10 个按钮，用于触发用户自定义的操作。



按键	功能按键	功能						
F1 UP F2 DOWN F3 LEFT OK RIGHT MEM ON/OFF	用户自定义	用户可以在 SDF 文件中自由定义这些按钮的显示文本和功能。						
MENU		<p>Advice</p> <p>MENU 按键将根据按键的时长触发不同的操作。</p> <table border="1"> <thead> <tr> <th>按键的持续时间</th> <th>功能</th> </tr> </thead> <tbody> <tr> <td>按键时间短</td> <td>打开“记录菜单”。</td> </tr> <tr> <td>按键时间长</td> <td>显示或隐藏按键网格。</td> </tr> </tbody> </table>	按键的持续时间	功能	按键时间短	打开“记录菜单”。	按键时间长	显示或隐藏按键网格。
按键的持续时间	功能							
按键时间短	打开“记录菜单”。							
按键时间长	显示或隐藏按键网格。							



ESC		返回“SDF 开始菜单”。

**Advice:** 如果总线启动，即使按 ESC 键进入“SDF 开始菜单”，它也会继续运行。再次在“SDF 开始菜单”中按 ESC，总线才会停止。

### 10.6.3.10 记录菜单

记录菜单允许您配置不同的记录器，可以更改以下设置。

- 来源

来源可以是任何已激活的通道。

- 目标

请查看“记录数据目标”一章。

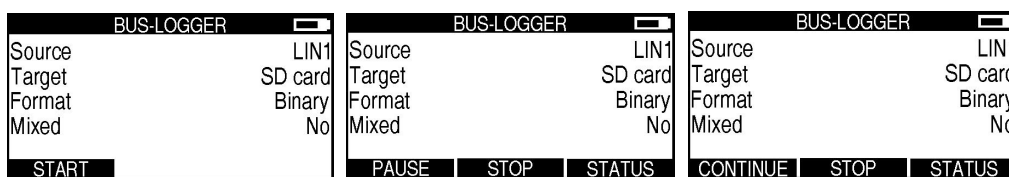
- 格式

请查看“记录数据格式”一章。

- 混合

使用此选项，多个通道的记录可以写入同一个记录文件。只要简单地激活此选项并启动每个通道单独的记录器。

请查看“配置和启用记录功能”一章，获取更多信息。



记录器未启动

记录器启动

记录器暂停

按键	功能按键	功能
F1	START	Advice 仅在该通道的记录器未激活时，该项目可用。 启动该通道的记录。
	PAUSE	Advice 仅在该通道的记录器激活且未暂停时，该项目可用。 暂停该通道的记录。
	CONTINUE	Advice 仅在该通道的记录器激活并暂停时，该项目可用。 恢复该通道的记录。
F2	STOP	Advice 仅在该通道的记录器激活时，该项目可用。 停止该通道的记录。
F3	STATUS	Advice 仅在该通道的记录器激活时，该项目可用。 显示该通道的记录状态。
ESC		返回“主菜单”。
MENU		
UP DOWN		移动光标。
OK LEFT		编辑当前选定的条目。

RIGHT		
-------	--	--

#### 10.6.4 导航

当前菜单	按键	下一菜单	
主菜单	F1	ECU 运行菜单	
	F2	记录菜单	
	F3	PC 模式菜单	
	MENU	设置菜单	
设置菜单	ESC	主菜单	
PC 模式菜单	F1	主菜单	
	ESC	Warning 如果启用选项“无 SD 卡=> PC 模式”且未插入 SD 卡，则无法使用此功能。	
ECU 运行菜单	ESC	主菜单	
	MENU		
	LEFT	(如果选择了 SDF 文件) SDF 开始菜单	
	RIGHT		
OK			
SDF 开始菜单	F1	SDF 菜单	
	F2	SDF 文件	功能
		只有一个部分	键盘菜单
	有多个部分	部分选择菜单	

	ESC	主菜单
SDF 菜单	ESC	SDF 开始菜单
	MENU	记录菜单
部分选择菜单	ESC	SDF 开始菜单
	OK	键盘菜单
	LEFT	
RIGHT		
键盘菜单	ESC	SDF 开始菜单
	MENU	记录菜单
记录菜单	ESC	回到上一个菜单。
	MENU	主菜单/SDF 菜单/键盘菜单

## 11. SDF-会话配置文件

### 11.1 通道和 SDF 配置组

#### 11.1.1 介绍

随着第二代 Baby-LIN 产品和 SDF-V3 的推出，我们新增了对多总线接口的全面支持。

Baby-LIN 产品上的每个总线接口现在由一个通道代表。每个通道都有自己的 SDF 功能，如虚拟信号，信号功能，宏，事件等。

SDF-V2 格式的 SDF 文件描述单个总线的一组配置。SDF-V3 格式的 SDF 文件可以包含多组配置。

以下章节解释了加载 SDF 文件时的通道，配置组以及从配置组到通道的映射。

### 11.1.2 配置组

通道由 SDF 文件进行配置。其中每个配置组都有自己的一套 SDF 功能，如虚拟信号，信号功能，宏，事件等。SDF 文件按照配置组分为不同的部分。

SDF-V2 格式的 SDF 文件只有一个部分，包含单个 LIN 通道和设备通道的配置。

SDF-V3 格式的 SDF 文件始终有一个设备的配置组。此外，它可以保留任意数量的 LIN 和 CAN 配置组。

配置类型	SDF 版本	描述
设备配置	仅 SDF-V3	主要是除了通道之外的配置。 在 SDF-V2 格式中, 这些配置可以在 LIN 部分中找到。
CAN 通道配置	SDF-V2 和 SDF-V3	该通道表示 LIN 总线接口。 在 SDF-V2 格式中, 这部分还包含了与通道无关的配置。
LIN 通道配置	仅 SDF-V3	该通道表示 CAN 总线接口。

### 11.1.3 通道

Baby-LIN 产品的每个总线接口由一个通道表示。每个通道都配置有 SDF 文件中的单个部分。以下通道类型可用：

下通道类型可用：

通道类型	描述
设备通道	该通道代表设备本身。它用于总线之外的功能。它通过使用第二代软件 (LINWorks v2, Baby-LIN-DLL v8 或更高版本的 BLC 接口) 提供。

LIN 通道	<p>该通道表示 LIN 总线接口。</p> <p>第一代的 Baby-LIN 产品有一个 LIN 总线接口可用。</p> <p>第二代 Baby-LIN 产品的某些设备可以使用多个 LIN 总线接口。</p>
CAN 通道	<p>该通道表示 CAN 总线接口。全功能的 CAN 总线接口仅适用于第二代 Baby-LIN 产品的某些设备。</p>

HARP-5 提供以下通道：

索引	通道	连接
0	设备	设备通道没有特定的总线接口。
1	LIN-1	"X5 - LIN, CAN and IO"
2	LIN-2	"X7 - LIN and CAN"
3	高速 CAN	"X7 - LIN and CAN"
4	低速 CAN	"X7 - LIN and CAN"

**Warning:** 通道索引为您提供通道的顺序。此通道索引由“Baby-LIN-DLL”使用。即使通道未激活，它也会显示。因此必须先检查通道信息的禁用标志。

#### 11.1.4 映射关系

加载 SDF 文件时，配置必须映射到 HARP-5 的通道。应用以下算法：

SDF 文件的各个部分从开始到结束依次加载。它们按照它们在 SDF 文件中出现的顺序进行映射。

LIN 配置会被映射到下一个未使用且激活的 LIN 通道。CAN 配置会映射到下一个未使用且激活

的 CAN 通道。如果已经没有更多的 LIN 或 CAN 通道可用，这部分将被丢弃。设备配置始终映射到设备通道。

这是 HARP-5 的一个例子：

SDFFile	
Section type	Section name
CAN	CAN-Section-1
LIN	LIN-Section-1
LIN	LIN-Section-2
CAN	CAN-Section-2
LIN	LIN-Section-3

→

HARP-5 (All channels activated)	
Channel type	Mapped section
LIN-1	LIN-Section-1
LIN-2	LIN-Section-2
CAN-HS	CAN-Section-1
CAN-LS	CAN-Section-2

**Version incompatibility:** 如果 SDF 文件包含无法映射到通道的部分，较旧版本的固件可能会返回错误。您应该检查以下一些典型原因

- SDF 文件是为其它 Baby-LIN 产品开发或使用的，该产品有更多的 LIN 或 CAN 通道。
- 通道不可用，因为您没有激活它。
- 没有激活 SDF-V3 支持。

## 11.2 配置属性

每个部分都有一些共同的属性和一组描述该部分的单独属性和信息。下表显示了这些属性：

属性	对象	描述
编号	所有	配置的索引编号。
名称	所有	每个配置都可以轻松利用 SessionConf 软件命名。
描述	所有	在这里您可以自由输入描述，例如注释，版本历史或类似的东西。
永久储存 SDF	所有	此全局属性显示在所有通道配置中，即使它是用来设置整

		个 SDF 文件的。这是由于这个设置十分重要。
LDF 协议版本	LIN	显示 LDF 文件的 LIN 协议版本，基于每个通道。
LDF 语言版本	LIN	显示 LDF 文件的 LIN 语言版本，基于每个通道。
波特率(比特/s)	LIN 和 CAN	这是总线的波特率。在这里，您可以覆盖 LDF/DBC 文件中的设置值。

## 11.3 设备通道设置

### 11.3.1 介绍

HARP-5 有一个永久存储器来保存配置设置。目标配置由特定于目标的选项组成。有两种可能性来设置它们：

- 使用 SimpleMenu 直接更改 HARP-5 中的目标配置。
- 使用 SessionConf 在 SDF 中设置特定于目标的选项。如果将 SDF 加载到 HARP-5 中，则 SDF 的目标特定选项可以覆盖设备配置。

### 11.3.2 配置选项列表

以下列表显示了可用于 HARP-5 的所有目标特定选项：

目标特定选项	描述
自启动	<p>Version incompatibility</p> <p>该选项不会对 HARP-5 产生影响。</p> <p>Advice</p> <p>请查看“独立模式和自启动”一章获取更多信息。</p>



空闲时的主请求帧	该选项决定了如果主请求帧的信号没有改变会发生什么。	
	值	描述
	Silent frame	HARP-5 不会发送任何主请求帧。
	Dummy frame	HARP-5 会发送所有数据字节设置为 0xFF 的主请求帧。
PC 超时后 USB 重置	Version incompatibility 该选项不会对 HARP-5 产生影响。	
LIN 总线供电时自启动	Version incompatibility 该选项不会对 HARP-5 产生影响。	
在 LIN 启动时跳过唤醒信号	该选项确定在 LIN 启动时是否发送唤醒信号。	
	值	描述
	关	唤醒信号在 LIN 启动时发送。
	开	唤醒信号在 LIN 启动时跳过发送。
Debug 模式	可能取值: 0 到 3 Attention 此选项仅用于调试目的。除非咨询 Lipowsky 公司并得到明确指示, 否则请勿更改此值。	
CRC-XOR 模式	该选项确定模式“CRC-8”和“CRC-16”的信号功能的 CRC-XOR 值。	
	值	描述
	同 0xffff 异或。	异或的值是 0xffff。
	同 0x0000 异或。	异或的值是 0x0000。

	<p>Version incompatibility</p> <p>仅在使用 SDF v2 时才包含此选项。</p> <p>SDF v3 将 CRC-XOR 值保留在文件中。</p>
--	---

### 11.3.3 设置目标特定选项

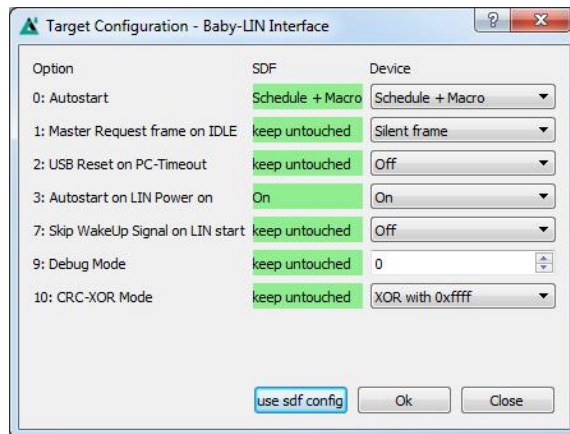
#### 11.3.3.1 直接在设备上配置

您可以使用 SimpleMenu 更改 HARP-5 的目标配置：

- 用电脑连接 HARP-5。
- 启动 SimpleMenu 并连接到您的 HARP-5。
- 右键单击设备列表中的 HARP-5，然后选择“目标配置”。

将打开一个对话框，并在设备菜单中显示 HARP-5 的当前配置。在这里您可以更改每个特定于目标的选项。

在连接到 HARP-5 后，如果您将 SDF 下载到 HARP-5，则“SDF”列将显示该 SDF 的设置。



#### 11.3.3.2 通过 SDF v3 间接设置目标特定选项

如果 HARP-5 需要特定的目标特定选项, 或者您想为多台 Baby-LIN 应用此配置, 可以利用 SDF v3 文件进行设置。每次将 SDF 下载到设备时, 都可以用 SDF 的设置覆盖设备的目标配置。另外每个选项也可以设置为不覆盖设备的。

- 启动 SessionConf 并加载 SDF v3 或创建一个新的。
- 选择设备部分。
- 展开左侧菜单中的“目标特定选项”页面。

目标特定选项分成两种, 一种是公共选项, 包含每种 Baby-LIN 设备上都可用的选项; 另外还有仅在特定种类的设备上可用的选项。每个选项还可设置为“保持不变”, 此选项是默认值, 并确保 Baby-LIN 设备中的对应值不会更改。因此, 可以选择只设置特定的选项。

## 11.4 虚拟信号

### 11.4.1 介绍

信号是 LIN 或 CAN 帧的一部分。它们在 LDF 或 DBC 文件中定义。

SDF 文件提供了定义虚拟信号的机会。虚拟信号的处理方式与普通总线信号相似。区别在于它们是在 SDF 中定义的, 而不是在 LDF 或 DBC 文件中定义的。

虚拟信号可用于多种用途。以下是一些常见用途:

- 它们可以用作计算的缓冲或记录中间结果的变量。
- 它们可以用作计数器或状态变量等辅助变量。
- 它们可以用作信号事件的触发器。
- 自定义的协议需要用到它们。查看章节“自定义协议”了解更多信息。
- 授予对系统特定资源的访问权限。查看章节“系统变量”了解更多信息。

### 11.4.2 信号长度

SDF-V2 格式中的虚拟信号的大小出于兼容性原因始终为无符号 16 位值。SDF-V3 允许 1 到 64 位的大小以及有符号的值。

这个方案的一个例外是系统变量。它们始终是无符号 32 位数值。查看章节“系统变量”了解更多信息。

	大小	符号
SDF-V2	16Bit	无符号
SDF-V3	1 到 64Bit	有符号或无符号
系统变量	32Bit	无符号

### 11.4.3 范围

虚拟信号的范围限于其定义的部分。因此，在该部分中，每个加载到该通道虚拟信号都应在该通道有一个实例。

即使虚拟信号的名称相同，也无法在多个部分上共享常规的虚拟信号的值。

为了确认这个特殊的系统变量是可用的。查看章节“@@ SYSGLOBAL”获取更多信息。

**Warning:** 由于每个通道都有自己的系统变量实例（“@@ SYSGLOBAL”除外），因此可以通过多个通道同时访问单个系统特定资源。请相应地分配您的系统变量。一种解决方案是仅从单个通道访问单个系统资源。

### 11.4.4 总线开始时重置

虚拟信号对总线启动可能会有不同的反应。下表是一个概述：

SDF 版本	功能
SDF-V2	当总线启动/重启时，所有虚拟信号都被重置为其初始值。
SDF-V3	SDF-V3 允许用户改变虚拟信号的行为。当总线启动/重启时，可以选择让虚拟信号保持其值不变。

**Version incompatibility:** 为了保证向后兼容，当将 SDF-V2 转换为 SDF-V3 时，所有虚拟信号都设置为在总线启动时重置。因此，只有设置在总线启动时将所有虚拟信号复位，才能将 SDF-V3 转换为 SDF-V2。

**Warning:** 一些系统变量会忽视上述设置，并且将始终以特定的方式运行。查看章节“系统变量”了解更多信息。

## 11.5 系统变量

### 11.5.1 介绍

系统变量是具有保留名称的虚拟信号。它们允许访问系统特定的资源，如定时器，输入和输出等。

在使用这些系统变量之前，需要将它们添加到虚拟信号中。系统变量始终以“@@”字符串开头。

系统变量的大小始终为无符号 32 位数值。

**Warning:** 某些系统变量忽略总线启动时，默认的重置行为或设置。

有三种类型的系统变量：

类型	描述
单一	该名称代表一个单一的变量, 可以直接使用该名称。示例: SYSTIMER_RTC_HOUR。
前缀	变量的名称使用已定义的前缀和自定义后缀。该组合表示一个单独的变量。示例: SYSTIMER_DOWN_MYTIME。
组	变量的名称由组名和数字组成。该组合表示一个单独的变量。示例: SYSDIGOUT1

系统变量不会在总线启动时以相同的方式作出反应。一些像正常的虚拟信号一样被重置为其初始值, 而另一些则不会被重置:

重置动作	描述
默认	这些系统变量和普通的虚拟信号具有相同的行为。 有关详细信息, 请查看“总线开始时重置”一章。
从不重置	这些变量的值永远不会在总线启动时复位。即使在 SDF-V2 中, 或者在 SDF-V3 中进行过配置。

一些系统变量是只读的。它们的设置将会被忽略。

**Warning:** 由于每个通道都有自己的系统变量实例 (“@@ SYSGLOBAL”除外), 因此可以通过多个通道同时访问单个系统特定资源。请相应地分配您的系统变量。一种解决方案是仅从单个通道访问单个系统资源。

### 11.5.2 概览

下表概述了可用的系统变量。

类别	系统变量名称	描述	类型	重置动作	只读
定时	"@@SYSTEMER_UP"	加计时器	前缀	默认	
定时	"@@SYSTEMER_FAST_UP"	快速加计时器	前缀	默认	
定时	"@@SYSTEMER_DOWN"	减计时器	前缀	默认	
定时	"@@SYSTEMER_FAST_DO WN"	快速减计时器	前缀	默认	
定时	"@@SYSTEMER_RTC_YEA R"	实时时钟的年份	单一	从不重置	是
定时	"@@SYSTEMER_RTC_MO NTH"	实时时钟的月份	单一	从不重置	是
定时	"@@SYSTEMER_RTC_DAT E"	实时时钟的日期	单一	从不重置	是
定时	"@@SYSTEMER_RTC_HOU R"	实时时钟的小时	单一	从不重置	是
定时	"@@SYSTEMER_RTC_MIN"	实时时钟的分钟	单一	从不重置	是
定时	"@@SYSTEMER_RTC_SEC"	实时时钟的秒数	单一	从不重置	是
数字 I/O	"@@SYSDIGIN"	获取数字输入的状态	组	默认	是
数字 I/O	"@@SYSDIGOUT"	设置数字输出的状态	组	默认	
系统	"@@SYSINFO"	获取一些系统的信息	组	从不重置	是
系统	"@@SYSCFG"	设置一些系统的配置	组	从不重置	
系统	"@@SYSBUSSTATE"	获取 LIN 总线状态	单一	从不重置	是
系统	"@@SYSMACROS_CONC"	决定是否可以同时运行	单一	默认	

	URRENT"	多个宏			
存储	"@@SYSGLOBAL"	在当前使用的所有部分/ 通道之间交换数据。	组	从不重置	
存储	"@@SYSPERSISTENT"	永久储存数据	组	从不重置	
记录	"@@SYSLOGTIMEBASE"	设置时间戳的分辨率	单一	从不重置	
记录	"@@SYSLOGCONTROL"	允许启动, 停止, 暂停和 恢复日志记录	单一	从不重置	
记录	"@@SYSLOGFORMAT"	设置记录文件的格式	单一	从不重置	
记录	"@@SYSLOGTARGET"	设置记录数据的目标	单一	从不重置	
记录	"@@SYSLOGFILESIZE"	设置记录文件的大小	单一	从不重置	
LIN 报头	"@@SYSBREAKTIME"	设置中断脉冲的持续时 间	单一	默认	
LIN 报头	"@@SYSBREAKPAUSE"	设置中断脉冲和下一个 同步段之间的暂停	单一	默认	
LIN 报头	"@@SYSINTERBYTETIME" "	设置在两个字符间的字 节间隔持续时间	单一	默认	
LIN 报头	"@@SYSWAKEUPTIME"	设置一个唤醒帧的持续 时间	单一	默认	
LIN 报头	"@@SYSWAKEUPSCHE DULEDELAY"	设置唤醒帧和随后的第 一帧之间的延时	单一	默认	
协议	"@@SYS_SERVICE_REQU EST_NAD"	设置 DTL 协议中服务请 求的 NAD。	单一	默认	



协议	"@@SYS_SERVICE_RESP ONSE_NAD"	获取 DTL 协议中收到的 服务响应的 NAD。	单一	默认	是
协议	"@@SYS_SERVICE_RESP ONSE_LEN"	获取收到的服务响应的 长度。	单一	默认	是
协议	"@@SYS_SERVICE_FLOW CTRL_BS"	设置 CAN 多帧报文的流 控制帧的大小。	单一	默认	

### 11.5.3 定时

#### 11.5.3.1 @@SYSTIMER\_UP

**Advice:** 此系统变量名称只是一个前缀。要创建一个系统变量，必须为它添加一个后缀。您可以使用任何字符串作为后缀。关于此类型的系统变量的名称或数量没有限制。

@@ SYSTIMER\_UP 系统变量前缀将为加计数器分配以下属性：

- 如果其值不为 0，它将开始计数。
- SDF-V2 中计数的最大值为 65535。
- SDF-V3 中计数的最大值 4294967295。
- 计时器计数周期为 1 秒。

#### 11.5.3.2 @@SYSTIMER\_FAST\_UP

**Advice:** 此系统变量名称只是一个前缀。要创建一个系统变量，必须为它添加一个后缀。您可以使用任何字符串作为后缀。关于此类型的系统变量的名称或数量没有限制。

@@SYSTIMER\_FAST\_UP 系统变量前缀将为加计数器分配以下属性：

- 如果其值不为 0，它将开始计数。
- SDF-V2 中计数的最大值为 65535。
- SDF-V3 中计数的最大值 4294967295。
- 计时器计数周期为 10 毫秒。

### 11.5.3.3 @@SYSTIMER\_DOWN

**Advice:** 此系统变量名称只是一个前缀。要创建一个系统变量，必须为它添加一个后缀。您可以使用任何字符串作为后缀。关于此类型的系统变量的名称或数量没有限制。

@@ SYSTIMER\_DOWN 系统变量前缀将为减计数器分配以下属性：

- 如果其值不为 0，它将开始计数。
- 计数器计数直到达到最小值 0。
- 计时器计数周期为 1 秒。

### 11.5.3.4 @@SYSTIMER\_FAST\_DOWN

**Advice:** 此系统变量名称只是一个前缀。要创建一个系统变量，必须为它添加一个后缀。您可以使用任何字符串作为后缀。关于此类型的系统变量的名称或数量没有限制。

@@SYSTIMER\_FAST\_DOWN 系统变量前缀将为减计数器分配以下属性：

- 如果其值不为 0，它将开始计数。
- 计数器计数直到达到最小值 0。

- 计时器计数周期为 10 毫秒。

### 11.5.3.5 @@SYSTIMER\_RTC\_YEAR

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYSTIMER\_RTC\_YEAR 系统变量包含由电池供电的实时时钟的年份值。

**Warning:** 无法通过设置此变量改变时间。要设置时钟，请按“MENU”按钮并选择“设置日期和时间”。

**Advice:** 由@@ SYSTIMER\_RTC\_HOUR, @@ SYSTIMER\_RTC\_MIN 和@@ SYSTIMER\_RTC\_SEC 组成的时钟将运行至 23:59:59 后，会重置为 00:00:00。

### 11.5.3.6 @@SYSTIMER\_RTC\_MONTH

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYSTIMER\_RTC\_MONTH 系统变量包含由电池供电的实时时钟的月份值。

**Warning:** 无法通过设置此变量改变时间。

要设置时钟，请按“MENU”按钮并选择“设置日期和时间”。

**Advice:** 由@@ SYSTIMER\_RTC\_HOUR, @@ SYSTIMER\_RTC\_MIN 和@@

SYSTIMER\_RTC\_SEC 组成的时钟将运行至 23:59:59 后, 会重置为 00:00:00。

### 11.5.3.7 @@SYSTIMER\_RTC\_DATE

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能, 虚拟信号的名称必须完全匹配, 不能使用任何后缀。

@@ SYSTIMER\_RTC\_DATE 系统变量包含由电池供电的实时时钟的日期值。

**Warning:** 无法通过设置此变量改变时间。要设置时钟, 请按“MENU”按钮并选择“设置日期和时间”。

**Advice:** 由@@ SYSTIMER\_RTC\_HOUR, @@ SYSTIMER\_RTC\_MIN 和@@

SYSTIMER\_RTC\_SEC 组成的时钟将运行至 23:59:59 后, 会重置为 00:00:00。

### 11.5.3.8 @@SYSTIMER\_RTC\_HOUR

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能, 虚拟信号的名称必须完全匹配, 不能使用任何后缀。

@@ SYSTIMER\_RTC\_DATE 系统变量包含由电池供电的实时时钟的小时值。

**Warning:** 无法通过设置此变量改变时间。要设置时钟，请按“MENU”按钮并选择“设置日期和时间”。

**Advice:** 由@@ SYSTIMER\_RTC\_HOUR, @@ SYSTIMER\_RTC\_MIN 和@@ SYSTIMER\_RTC\_SEC 组成的时钟将运行至 23:59:59 后，会重置为 00:00:00。

#### 11.5.3.9 @@SYSTIMER\_RTC\_MIN

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYSTIMER\_RTC\_DATE 系统变量包含由电池供电的实时时钟的分钟值。

**Warning:** 无法通过设置此变量改变时间。要设置时钟，请按“MENU”按钮并选择“设置日期和时间”。

**Advice:** 由@@ SYSTIMER\_RTC\_HOUR, @@ SYSTIMER\_RTC\_MIN 和@@ SYSTIMER\_RTC\_SEC 组成的时钟将运行至 23:59:59 后，会重置为 00:00:00。

#### 11.5.3.10 @@SYSTIMER\_RTC\_SEC

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYSTIMER\_RTC\_DATE 系统变量包含由电池供电的实时时钟的秒值。

**Warning:** 无法通过设置此变量改变时间。要设置时钟，请按“MENU”按钮并选择“设置日期和时间”。

**Advice:** 由@@ SYSTIMER\_RTC\_HOUR, @@ SYSTIMER\_RTC\_MIN 和@@ SYSTIMER\_RTC\_SEC 组成的时钟将运行至 23:59:59 后，会重置为 00:00:00。

## 11.5.4 数字 IO

### 11.5.4.1 @@SYSDIGIN

**Advice:** 此系统变量名称表示一组索引变量。要使用其中的一个变量，必须将它的索引作为后缀添加到名称中。

@@ SYSDIGIN 系统变量组允许访问 HARP-5 的数字输入。可以通过读取这些系统变量，查询数字输入的状态。

状态	变量值
输入低电平	0
输出高电平	1

变量组包含以下系统变量：

变量	数字输入	硬件描述
@@SYSDIGIN1	数字 IO X5-6	请查看“X5 - LIN, CAN 和 IO”一章获取更多信息。

### 11.5.4.2 @@SYSDIGOUT

**Advice:** 此系统变量名称表示一组索引变量。要使用其中的一个变量，必须将它的索引作为后缀添加到名称中。

@@ SYSDIGOUT 系统变量组允许访问 HARP-5 的数字输出。可以通过这些系统变量设置数字输出的状态。

变量组包含以下系统变量：

变量	数字输出	硬件描述								
@@SYSDIGOUT1	数字 IO X5-6	<p>设置数字输出。</p> <p>请查看“X5 - LIN, CAN 和 IO”一章获取更多信息。</p> <table border="1"> <thead> <tr> <th>变量值</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>关闭：输出没有接地。</td> </tr> <tr> <td>1</td> <td>开启：输出接地。</td> </tr> <tr> <td>2 或更高</td> <td>开启指定的时间(ms)：输出暂时接地。</td> </tr> </tbody> </table>	变量值	状态	0	关闭：输出没有接地。	1	开启：输出接地。	2 或更高	开启指定的时间(ms)：输出暂时接地。
变量值	状态									
0	关闭：输出没有接地。									
1	开启：输出接地。									
2 或更高	开启指定的时间(ms)：输出暂时接地。									
@@SYSDIGOUT33	声音信号发生器	<p>激活声音信号发生器发出一次蜂鸣声。持续时间以毫秒为单位。</p> <table border="1"> <thead> <tr> <th>变量值</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>任意</td> <td>蜂鸣持续时间(ms)</td> </tr> </tbody> </table>	变量值	状态	任意	蜂鸣持续时间(ms)				
变量值	状态									
任意	蜂鸣持续时间(ms)									
@@SYSDIGOUT34	LCD 背光灯	<p>打开或关闭 LCD 背光灯。</p> <table border="1"> <thead> <tr> <th>变量值</th> <th>状态</th> </tr> </thead> <tbody> </tbody> </table>	变量值	状态						
变量值	状态									

		0	关闭						
		1 或更高	打开						
@@SYSDIGOUT35	12V 供电	<p>打开或关闭 12V 供电。</p> <p>请查看“X7 - LIN 和 CAN”一章获取更多信息。</p> <table border="1"> <thead> <tr> <th>变量值</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>12V 供电关闭</td> </tr> <tr> <td>1 或更高</td> <td>12V 供电打开</td> </tr> </tbody> </table>		变量值	状态	0	12V 供电关闭	1 或更高	12V 供电打开
变量值	状态								
0	12V 供电关闭								
1 或更高	12V 供电打开								
@@SYSDIGOUT36	VSwitch X7-1	<p>该引脚通过内部开关连接到 LIN-Supply 引脚。</p> <p>如果激活，该引脚可通过 VSwitch 引脚上的内部或外部 LIN 总线电源为外部 LIN 总线节点供电。</p> <p>无论 LIN 总线电压是由外部电源还是由集成的 12V 电源提供，此功能都有效。</p> <p>请查看“X7 - LIN 和 CAN”一章获取更多信息。</p> <table border="1"> <thead> <tr> <th>变量值</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>无供电</td> </tr> <tr> <td>1 或更高</td> <td>LIN 总线供电</td> </tr> </tbody> </table>		变量值	状态	0	无供电	1 或更高	LIN 总线供电
变量值	状态								
0	无供电								
1 或更高	LIN 总线供电								
@@SYSDIGOUT37	LIN 总线 1 通道和 2 通道连接	<p>两个 LIN 总线接口默认连接。通过设置此变量，可以单独使用它们。</p> <p>请查看“X5 - LIN, CAN 和 IO”和“X7 - LIN 和 CAN”章节获取更多信息。</p> <table border="1"> <thead> <tr> <th>变量值</th> <th>状态</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LIN 总线 1 通道和 2 通道连接在一</td> </tr> </tbody> </table>		变量值	状态	0	LIN 总线 1 通道和 2 通道连接在一		
变量值	状态								
0	LIN 总线 1 通道和 2 通道连接在一								



			起。
		1 或更高	LIN 总线 1 通道和 2 通道单独使用。

## 11.5.5 系统

### 11.5.5.1 @@SYSINFO

**Advice:** 此系统变量名称表示一组索引变量。要使用其中的一个变量，必须将它的索引作为后缀添加到名称中。

@@ SYSINFO 系统变量组授予对某些系统信息的访问权限。变量组包含以下变量：

系统变量	描述	变量值		
		设备	值（十六进制）	值（十进制）
@@SYSINFO 1	该变量包含一个表示硬件类型的数字。	Baby-LIN-II	503	1283
		Baby-LIN-RC-II	501	1281
		Baby-LIN-RM-II	500	1280
		HARP-4/HARP-5	502	1282
		Baby-LIN-MB-II	700	1792
		Version incompatibility 老版本的 Baby-LIN-MB-II 固件将返回 0x503 (1283).		
@@SYSINFO 2	该变量检测第一个 LIN 总线的 LIN 电源	LIN 总线电源	变量值	
		关闭	0	

	线的状态。	打开	1	
	Advice 该系统变量已失效。请使用“@@ SYSBUSSTATE”。			
@@SYSINFO 3	该变量包含低速	错误状态	变量值	
	CAN 收发器的 Error	无 CAN 错误	0	
	引脚的状态。	CAN 错误	1	
@@SYSINFO 4	该变量包含 HARP-5 的序列号。	例如：1065776		
@@SYSINFO 5	该变量包含 HARP-5 的固件版本。	例如：571 表示 v5.71 固件版本。		
@@SYSINFO 6	该变量包含 HARP-5 的固件编号。	例如：10		

### 11.5.5.2 @@SYSCFG

**Advice:** 此系统变量名称表示一组索引变量。要使用其中的一个变量，必须将它的索引作为后缀添加到名称中。

@@ SYSCFG 系统变量组允许配置影响 HARP-5 行为的某些设置。变量组包含以下变量：

系统变量	总线	描述	变量值	
@@SYSCFG1	LIN-1	该变量允许将 LIN 线强制置为低电平。	值	描述
			0	将 LIN 线强制置为低电平。

@@SYSCFG3 0	LIN-2		Warning	
			由于 MC33662 LIN 收发器中的硬件显性电平控制超时，电平强制置为低电平后，将在约 5 ms 的最大周期时间后返回高电平。	
			1	将 LIN 线强制置为高电平。
			2	LIN 电平由固件控制。
@@SYSCFG2	LIN-1	该变量允许强制配置上拉。	值	描述
@@SYSCFG3	LIN-2		0	强制配置从节点上拉。
1			1	强制配置主节点上拉。
			2	上拉由固件控制。
@@SYSCFG1 00	LIN	禁用 LIN 供电。	该系统变量与“X7 - LIN 和 CAN”中的 Supply-Config 引脚功能相同。如果激活，则不会通过 LIN 总线为 HARP-5 供电。	
			值	描述
			0	设备从 LIN 总线供电。
			1	设备不从 LIN 总线供电。
@@SYSCFG1 0	CAN-1	该变量允许将 CAN 总线强制置为显性电平。	值	描述
			0	将 CAN 总线强制置为隐性电平。
@@SYSCFG2 0	CAN-2		1	将 CAN 总线强制置为显性电平。
			2	电平行为由固件控制。

			值	描述
@@SYSCFG1 1	CAN-1	该变量允许配置跳变沿的斜率。	0	高速模式：对输出的上升/下降沿斜率没有限制。
			1	斜率控制模式：降低跳变沿的压摆率以改善电磁兼容性（EMC）。
			2	睡眠模式：进入省电模式，只能监控总线。

### 11.5.5.3 @@SYSBUSSTATE

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYSBUSSTATE 系统变量提供 LIN 总线的状态。

**Warning:** 此系统变量的返回值仅在 LIN 通道时有意义。

定义了以下值：

值	描述
0	LIN 总线上没有电压。
1	LIN 总线上有电压，但是没运行调度表。
2	LIN 总线上有电压，并有一个调度表正在运行。

#### 11.5.5.4 @@SYSMACROS\_CONCURRENT

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYSMACROS\_CONCURRENT 系统变量允许您配置最大同时执行宏的数量。仅支持 SDF-V2 的旧设备一次只能执行一个宏。新设备可以一次执行多个宏。出于兼容性原因，HARP-5 可以主动限制为一次只执行一个宏。

定义了以下值：

值	SDF 文件行为	描述
0	SDF-V3	一次最多可以执行 32 个宏。
1	SDF-V2	一次只能执行一个宏。

**Warning:** 设备加载 SDF 文件时，就会评估变量的初始值。之后请不要再更改变量，以防止意外的结果。

### 11.5.6 存储

#### 11.5.6.1 @@SYSGLOBAL

**Advice:** 此系统变量名称表示一组索引变量。要使用其中的一个变量，必须将它的索引作为后缀添加到名称中。

@@ SYSGLOBAL 系统变量组允许在所有当前使用的通道之间交换数据。

普通虚拟信号的使用范围仅限于它所定义的部分。因此，一个通道对它所在部分中的每个虚拟信号都有一个实例，该实例仅被加载到这个通道。@@ SYSGLOBAL 系统变量的行为是不同的，所有通道将共用一个实例。因此，它们可用于在不同通道之间共享数据。

要访问@@ SYSGLOBAL 系统变量的共享内存，每个部分都必须先定义这个变量。

**Warning:** 尽管该系统变量需要在每个部分中事先定义，因为它希望在彼此之间交换数据。每个部分都可以定义一个初始值。但是这没有任何意义，因为它们使用相同的值。

如果使用不同的初始值，则上次初始化通道的部分设置的值将占主导地位。有关详细信息，请查看“通道和部分”一章。

如果一个部分更改了@@ SYSGLOBAL 系统变量的值，其他部分也会对应更改该变量的值，也可以对该变量的信号变化做出反应。使用此机制可以将值从一个部分传输到另一个部分。

通过设置 CAN 部分中的信号值并在 LIN 部分读取并处理此信号值可以实现网关的功能。

系统变量	描述
@@SYSGLOBAL1	全局系统变量，各部分通用。
.....	
@@SYSGLOBAL32	

#### 11.5.6.2 @@SYSPERSISTENT

**Advice:** 此系统变量名称表示一组索引变量。要使用其中的一个变量，必须将它的索引作为后缀添加到名称中。

@@ SYSPERSISTENT 系统变量组允许在设备上持久存储数据。即使设备断电后重启，数据仍然可用。

典型用例是：

- 时间计数器。
- 错误计数器。
- 将值从一个 SDF 文件传递到另一个。

使用@@ SYSPERSISTENT 系统变量后，将在加载 SDF 文件时加载其值。

**Warning:** 当值被其它部分或 SDF 文件外部更改时，在加载 SDF 文件之后，该值可能会发生改变。请记住以下规则

- 仅在加载 SDF 文件时加载@@ SYSPERSISTENT 系统变量的值。
- 最后一次写入值有效。

您应该只在一个部分内设置同一个@@ SYSPERSISTENT 系统变量。

初始值将由第一个包含@@ SYSPERSISTENT 系统变量定义的部分分配。初始值分配之后，将忽略所有其他部分的初始值设置。有关详细信息，请查看“通道和部分”一章。

系统变量	描述
@@SYSPERSISTENT1	永久存储的系统变量,在 HARP-5 断电后值也不会重置。
.....	
@@SYSPERSISTENT6	
4	

@@ SYSPERSISTENT 系统变量的值也可以通过特定的命令获取和设置:

命令	描述
persistent_rd <index>	获取@@ SYSPERSISTENT 系统变量的值。
persistent_wr <index> <value>	设置@@ SYSPERSISTENT 系统变量的值。
persistent_wr <index>	重置@@ SYSPERSISTENT 系统变量的值。

**Warning:** persistent\_wr <address>命令是删除以前设置的@@ SYSPERSISTENT 系统变量的唯一方法。

**Advice:** 该索引从 0 开始。请使用索引 0 来获取和设置@@ SYSPERSISTENT1。

## 11.5.7 记录

### 11.5.7.1 @@SYSLOGTIMEBASE

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYSLOGTIMEBASE 系统变量允许更改记录的帧的时间戳分辨率。

@@ SYSLOGTIMEBASE 的默认时间戳分辨率是 1 $\mu$ s。由于时间戳的数据大小是 32 位，因此在大约 4,295 秒（大约 1.2 小时）之后时间戳将会溢出。

要增加溢出周期，可以将时间戳的分辨率更改为 1ms。然后会在 4,294,967 秒（大约 49 天）之后发生溢出。



值	时间戳分辨率	溢出用秒	溢出用时	溢出用天
0	1 微秒	约 4295 秒	约 1.2 小时	约 0.05 天
1	1 毫秒	约 4294967 秒	约 1193 小时	约 49.7 天

### 11.5.7.2 @@SYSLOGCONTROL

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYSLOGCONTROL 系统变量可以启动，停止，暂停/恢复数据记录。因此，可通过宏或事件控制报文记录。

值	描述
0	停止记录，关闭记录文件。
1	通过新建记录文件或恢复暂停的记录文件来开始记录。
2	暂停记录，通过设置 1 来恢复记录，或者通过设置 0 来停止记录。

### 11.5.7.3 @@SYSLOGFORMAT

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYSLOGFORMAT 系统变量允许更改记录文件的格式。默认值是二进制格式。

值	描述
0	二进制格式。请查看“二进制格式”一章获取更多信息。

1	ASCII 文本格式。请查看“ASCII 格式”一章获取更多信息。
2 或更高	重置为默认值：二进制格式。

#### 11.5.7.4 @@SYSLOGTARGET

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYSLOGTARGET 系统变量设置接收记录数据的目的地。

值	描述
0	SD 卡: 记录数据在 SD 卡上储存为文件。储存的文件路径是 HARP-5 \ Loggs。文件名会包含开始记录的日期，时间和通道。
1	<p>USB: 记录数据通过“X8 - PC 连接器”接口传输。您可以使用终端程序在 PC 上记录这些数据。使用以下设置连接到 HARP-5:</p> <ul style="list-style-type: none"> <li>• COM 端口: HARP-5 的虚拟 COM 端口。</li> <li>• 波特率: 取决于 HARP-5 的设置, 115200 或 912600。</li> <li>• 数据位: 8</li> <li>• 停止位: 1</li> <li>• 奇偶校验: 无</li> <li>• 流量控制: 无</li> </ul> <p>Warning</p> <p>HARP-5 的 PC 模式可能没有被激活。请查看“PC 模式”一章获取更多信息。</p>

2 或更高	重置为默认值：SD 卡。
-------	--------------

### 11.5.7.5 @@SYSLOGFILESIZE

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYSLOGFILESIZE 系统变量用于设置记录文件的最大容量，单位为 MB。如果记录志文件超过该数值，它将自动关闭，并启动新的日志文件。

该系统变量可以设置在 1 MB 到 4096 MB 之间。所有其它值将视为 4096 MB。

## 11.5.8 LIN 报头

### 11.5.8.1 简介

有时我们需要将帧内的 LIN 时间参数更改为与 LIN 规范中定义的典型值不同的值。

一个示例是使用 shunt 方式的 SNPD（从节点位置检测）的节点，要使用较长的中断时间脉冲长度。

### 11.5.8.2 @@SYSBREAKTIME

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYSBREAKTIME 系统变量允许以位时间的倍数定义中断脉冲时间。

设置系统变量@@SYSBREAKTIME 会将中断时间更改为该位时间值。默认值为 16，表示中断时间为 16 个位时间。将值设置为 0 会将中断时间重置为默认值 16 个位时间。

属性	描述
单位	1 个位时间。
默认值	16 => 16 个位时间。
设置为 0 时	重置为默认值。

### 11.5.8.3 @@SYSBREAKPAUSE

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYSBREAKPAUSE 系统变量允许定义中断脉冲和后续同步字符间的空闲时间。

设置此系统变量会按照数值更改中断暂停时间，单位为 0.1 个位时间。默认值为 15，表示 1.5 个时间的中断暂停。将值设置为 0 会将中断暂停重置为默认值 1.5 个位时间。

属性	描述
单位	0.1 个位时间。
默认值	15 => 1.5 个位时间。
设置为 0 时	重置为默认值。
最大值	100

### 11.5.8.4 @@SYSINTERBYTETIME

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYSINTERBYTETIME 系统变量允许定义 HARP-5 发送的字符之间的字节间时间。

设置此系统变量会按照数值更改字节间时间，单位为 $\mu\text{s}$ 。默认值为 0，表示字节间时间为  $0\mu\text{s}$ 。

属性	描述
单位	$\mu\text{s}$
默认值	0 => $0\mu\text{s}$ 。

#### 11.5.8.5 @@SYSWAKEUPTIME

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYSWAKEUPTIME 系统变量允许定义唤醒事件的长度。每次 LIN 总线启动时，HARP-5 都会产生唤醒事件。

设置此系统变量会按照数值更改唤醒事件的长度，单位为 $\mu\text{s}$ 。默认值为 500，表示唤醒事件长度为  $500\mu\text{s}$ 。

属性	描述
单位	$\mu\text{s}$
默认值	500 => $500\mu\text{s}$ 。

#### 11.5.8.6 @@SYSWAKEUPSCHEDULEDELAY

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYSWAKEUPSCHEDULEDELAY 系统变量允许定义唤醒事件和第一帧之间的延迟。每次 LIN-Bus 启动时，HARP-5 都会产生唤醒事件。

设置此系统变量会按照数值更改延迟，单位为 $\mu\text{s}$ 。默认值为 100000，表示延迟 100 ms。

属性	描述
单位	$\mu\text{s}$
默认值	100000 => 100 ms。

## 11.5.9 协议

### 11.5.9.1 @@SYS\_SERVICE\_REQUEST\_NAD

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYS\_SERVICE\_REQUEST\_NAD 系统变量可以设置在 DTL 协议的服务请求中使用的 NAD。

请查看“自定义协议”一章获取更多信息。

**Warning:** 该系统变量仅对 DTL 协议的服务有效。

### 11.5.9.2 @@SYS\_SERVICE\_RESPONSE\_NAD

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYS\_SERVICE\_RESPONSE\_NAD 系统变量可以获取在 DTL 协议的服务响应中接收的 NAD。

请查看“自定义协议”一章获取更多信息。

**Warning:** 该系统变量仅对 DTL 协议的服务有效。

### 11.5.9.3 @@SYS\_SERVICE\_RESPONSE\_LEN

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

@@ SYS\_SERVICE\_RESPONSE\_LEN 系统变量允许获取服务响应中收到的有效负载长度。如果协议类型是“Raw”，则有效载荷长度等于数据长度。如果有效负载长度是可变的，则该值特别有趣。请查看“自定义协议”一章获取更多信息。

### 11.5.9.4 @@SYS\_SERVICE\_FLOWCTRL\_BS

**Advice:** 此系统变量名称表示单个变量。要使用此系统变量的功能，虚拟信号的名称必须完全匹配，不能使用任何后缀。

在 HARP-5 从另一个 CAN 节点接收多帧消息时，系统变量@@ SYS\_SERVICE\_FLOWCTRL\_BS 可以设置发送的流控制帧的字节数。这定义了需要在需要额外处理时间之前可以发送的连续帧的数量。请查看“自定义协议”一章获取更多信息。

值	描述
0..255	需要在额外的处理时间之前发送的连续帧的数量。
>255	高于 255 的值都将视为 255。

**Warning:** 该系统变量仅对 DTL 协议的服务有效。

## 11.6 宏命令

### 11.6.1 概览

宏由一个或多个宏命令组成。每个宏命令都是一条指令，允许修改信号、总线状态、流向控制等等。每个宏命令都有自己的参数。

**Version incompatibility:** 所有宏命令都在 SDF-V3 可用，但部分命令在 SDF-V2 不可用。

宏命令可以有执行条件，只有在条件满足时才执行。

所有宏命令如下表：

类别	命令	描述	需求
信号	Set signal	为信号分配一个值。	<ul style="list-style-type: none"> <li>目标信号必须被仿真</li> </ul>



	Increment signal	将给定值添加到信号（例如，-1 将使信号递减）。	<ul style="list-style-type: none"> <li>目标信号必须被仿真</li> </ul>
	Set signal from	将另一个信号的值传递给该信号。	<ul style="list-style-type: none"> <li>目标信号必须被仿真</li> <li>SDF-V3</li> </ul>
	Set bit	设置/清除信号中某一位。	<ul style="list-style-type: none"> <li>目标信号必须被仿真</li> <li>SDF-V3</li> </ul>
	Set using mathematical operation	对一个信号值和常数/另一个信号进行数学运算来设置信号。	<ul style="list-style-type: none"> <li>目标信号必须被仿真</li> <li>SDF-V3</li> </ul>
总线	Start	启动 LIN 总线。如果 HARP-5 模拟 LIN 主节点，则可以选择一个执行的调度表。	
	Restart	停止 LIN 总线后再次启动它，但信号值不会重置为默认值。	
	Stop	停止 LIN 总线。	
	Wakeup	开始一个唤醒事件。	<ul style="list-style-type: none"> <li>LIN</li> </ul>
	Sleep	开始一个睡眠事件。	<ul style="list-style-type: none"> <li>LIN</li> </ul>
	Set speed	设置 HARP-5 设备的波特率。	
	Freeze signals	停止总线上所有后续的信号变化。这可用于手动创建连贯的信号变化。	<ul style="list-style-type: none"> <li>SDF-V3</li> </ul>
	Unfreeze signals	应用所有之前的信号更改，让它们再次	<ul style="list-style-type: none"> <li>SDF-V3</li> </ul>

		出现在总线上。	
	Inject frame	在总线上插入一个自定义帧。	<ul style="list-style-type: none"> <li>• HARP-5 仿真</li> </ul> 主节点 <ul style="list-style-type: none"> <li>• SDF-V3</li> </ul>
	Set frame mode	禁用或隐藏 LIN 帧。 禁用或设置 CAN 帧的时隙。	<ul style="list-style-type: none"> <li>• SDF-V3</li> </ul>
	Execute service	执行自定义协议中的服务。	<ul style="list-style-type: none"> <li>• SDF 包含自定义协议的服务</li> <li>• SDF-V3</li> </ul>
LIN	Select schedule	更改由 LIN 主节点执行的调度表。也可以同时设置调度表模式。	<ul style="list-style-type: none"> <li>• LIN</li> <li>• HARP-5 仿真</li> </ul> 主节点
	Set schedule mode	永久更改调度表的执行模式： <ul style="list-style-type: none"> <li>• 循环</li> <li>• 单次运行</li> <li>• 完成退出</li> </ul>	<ul style="list-style-type: none"> <li>• LIN</li> <li>• HARP-5 仿真</li> </ul> 主节点 <ul style="list-style-type: none"> <li>• SDF-V3</li> </ul>
	Force checksum	强制设置校验和类型： <ul style="list-style-type: none"> <li>• 自动</li> <li>• V1</li> <li>• V2</li> </ul>	<ul style="list-style-type: none"> <li>• LIN</li> </ul>
	Send Master Request	在 LIN 总线上发送主请求帧（帧 ID 0x3C）。	<ul style="list-style-type: none"> <li>• LIN</li> <li>• HARP-5 仿真</li> </ul>

			主节点
	Send DTL Request	不可用的命令。由于兼容性原因可见。	<ul style="list-style-type: none"> <li>• LIN</li> <li>• HARP-5 仿真</li> </ul> 主节点
流向控制	Delay	指定延迟时间，暂停执行宏命令。	
	Jump	跳转到宏中的另一个命令。可用于实现循环或大的条件命令块。经常与条件语句一起使用。	
	Event	禁用或启用事件。	<ul style="list-style-type: none"> <li>• SDF 包含一个事件</li> </ul>
	Goto macro	切换到另一个宏。将不执行当前宏的其余命令。	<ul style="list-style-type: none"> <li>• SDF 包含一个宏</li> <li>• SDF-V3</li> </ul>
	Gosub macro	调用另一个宏。当前宏的剩余命令将在被调用的宏完成后执行。	<ul style="list-style-type: none"> <li>• SDF 包含一个宏</li> <li>• SDF-V3</li> </ul>
	Exit	这可以用于终止宏，如果它是由 Gosub 调用的宏。然后它会立即返回。	<ul style="list-style-type: none"> <li>• SDF-V3</li> </ul>
宏	Start	启动另一个宏。新的宏运行时将会独立于调用它的宏。	<ul style="list-style-type: none"> <li>• SDF 包含一个宏</li> <li>• SDF-V3</li> </ul>
	Stop	停止同时运行的“另一个宏”。	<ul style="list-style-type: none"> <li>• SDF 包含一个</li> </ul>

			宏 • SDF-V3
	Macroselection	利用一个宏选择调用宏。	• SDF 包含一个 宏选择 • SDF-V3

### 11.6.2 宏条件

#### Version incompatibility:

此功能仅在 SDF V3 可用。

根据您的设备，您可能需要额外的激活码。

需要更新最新固件。

在 SDF-V3 中，可以为每个宏命令设置条件。只有满足条件，才会执行宏命令。

条件本身会一直检查值的状态。合理的条件如下：

- 将信号值与常数进行比较：

=常量, <常量, >常量, !=常量, <=常量, >=常量

- 将信号值与另一个信号的值进行比较：

=信号, <信号, >信号, !=信号, <=信号, >=信号

- 检查信号值是否在特定范围内。

- 检查信号值是否超出特定范围。

- 利用位掩码检查是否设置了某些位。

### 11.6.3 宏变量

以下系统变量仅在宏的范围内可用，并且只能在此宏中访问，您无法在宏之外访问其值。

系统变量	描述
__ResultLastMacroCommand	包含最后一个宏命令的返回值。
__Return	用于更改当前宏的返回值。
__Failure	如果设置为 0 以外的其他值，则宏运行将失败，并返回这个值。
__LocalVariable1 ... __LocalVariable10	用于临时存储宏内的数据，而无需创建其他虚拟信号。

## 11.7 设备相关选项

### 11.7.1 简介

HARP-5 的某些功能仅适用于该设备。因此，这些功能在设备相关选项中按类型分组。

如果 SDF 文件包含特定设备的设备选项，当它被加载到其它的 Baby-LIN 产品上时，会自动忽略这些选项。

### 11.7.2 键盘

如果将 SDF 文件加载到 HARP-5 并打开“键盘菜单”，键盘选项可以配置界面。

界面及其选项包括以下两个方面：

- 显示值

在这里，您最多可以选择 5 个信号，这些信号将显示在“键盘菜单”中。您还可以编辑这些显示信号的名称。

**Warning:** 默认情况下, 只有前 2 个变量可见。另外 3 个隐藏在键盘视图下。可以通过长按 MENU 键隐藏键盘视图，就可以看到它们了。

- 键盘标签

在这里，您可以设置 10 个可配置按键的名称，它将显示在“键盘菜单”中。如果名称文本过长，则显示器只能显示一部分。

这些键的功能在事件中定义。

### 11.7.3 记录

这些记录选项可用于减少 HARP-5 正在记录的数据量。它们代表 HARP-5 记录器的过滤器。更多信息请查看“记录”一章。

首先，需要设置过滤器的类型：

过滤器类型	描述
Off	过滤器没有激活。所有帧都将会被记录。
Block	包含任何过滤器中列出的信号的帧将不会被记录。
Pass	包含任何过滤器中列出的信号的帧将会被记录。

然后，您可以将信号添加到过滤器列表。这些信号用于识别应该阻止或记录的帧。

## 12 .LINWorks 软件

### 12.1 总览

LINWorks 是用于操作 HARP-5 的软件集合。LINWorks 包含以下组件：

软件	路径(子文件夹)	描述
"Baby-LIN driver"	根目录	想要通过 USB 将 HARP-5 连接到 PC 上，必须先安装 Baby-LIN 驱动程序。HARP-5 会作为虚拟 COM 端口连接。
"LDFEdit"	LINWorks	LDFEdit 允许检查, 创建和编辑 LDF 文件(LIN 描述文件)。
"SessionConf"	LINWorks	SessionConf 允许查看, 创建和编辑 SDF 文件(会话描述文件), 可以通过导入 LDF 文件(用于 LIN 总线模拟)或 DBC 文件创建。它定义了不同总线上完整模拟所需的内容, 例如: 每个总线上的哪些节点可用以及哪些节点应由 HARP-5 模拟。此外, 它允许定义应用程序逻辑。这种编程能力适用于每个设备。
"SimpleMenu"	LINWorks	SimpleMenu 用于建立与 HARP-5 的连接并上传 SDF 文件到 HARP-5, 更改设备目标配置, 控制总线以及监控总线上的帧和信号。即使没有 LDF/DBC 文件, 也可以监视总线并记录帧。
"LogViewer"	LINWorks	LogViewer 可以显示和转换 HARP-5 以及 SimpleMenu 的记录文件。
"Baby-LIN-DLL"	Development	Baby-LIN-DLL 允许客户创建自己的应用程序并使用 HARP-5 的所有功能, 如控制和监控 LIN 和 CAN 总线接口。Baby-LIN-DLL 是 C / C ++的本地 DLL。 .NET 应用程

		序的封装也通过 LabView 文件提供。示例适用于所有受支持的编程语言。
"Baby-LIN-DLL .NET Wrapper"	Development	Baby-LIN-DLL .NET Wrapper 允许在 .NET 应用程序中使用 Baby-LIN-DLL 的所有功能。Baby-LIN-DLL .NET Wrapper 提供了与 Baby-LIN-DLL 相同的接口，只需对数据类型进行很少的调整。
"Serial writer"	Tools	Serial writer 用于更改存储在 HARP-5 的持久存储器中的序列号。序列号会影响虚拟 COM 端口号的分配。
"BLProg"	Tools	BLProg 用于更新 HARP-5 的固件。如果您从我们的门户网站( <a href="http://portal.lipowsky.de">portal.lipowsky.de</a> )下载固件更新包，则将始终包含当前版本的 BLProg。
"Switch to ASCII tool"	Tools	Switch to ASCII tool 允许将 HARP-5 切换到 ASCII 模式。在此模式下，可以发送 ASCII 命令以将 HARP-5 用作电平转换器，或通过 LIN 硬件层直接发送字符以跳过 LIN 帧的封装协议。

## 12.2 Baby-LIN 驱动

在安装 Baby-LIN 驱动程序后，才能通过 USB 将 HARP-5 连接到 PC 端。HARP-5 将作为虚拟 COM 端口连接。

Baby-LIN 驱动程序将 HARP-5 作为虚拟 COM 端口连接。与常规 COM 端口一样，HARP-5 将以类似“COM1”的名称显示在设备管理器中。使用“SimpleMenu”或“Baby-LIN-DLL”建立连接时也将使用此名称。



所有必需的安裝文件都將由安裝文件安裝。因此，它將檢查您的 Windows 版本並為您的操作系統安裝正確的版本。

**Advice: 驅動程序版本 v6.7 支持 Windows XP 系統。更高版本的驅動程序支持更新的操作系統。**

### 12.3 LDFedit

LDFedit 是 LDF 文件編輯器。它可用於檢查，創建和編輯 LDF 文件，其中包含 LIN 總線的完整描述。這包括節點，幀，信號和調度表等等以及它們相互的關係。

LDFedit 還具有後台檢查功能，可防止您保存無效或不兼容的 LDF 文件。

### 12.4 SessionConf

SessionConf 是 SDF 文件編輯器。它可用於檢查，創建和編輯 SDF 文件，其中包含模擬總線節點的相关配置。

要創建 SDF 文件, SessionConf 允許導入 LDF 和 DBC 文件。SDF 文件的配置和功能在“SDF - 會話配置文件”一章中介紹。

SessionConf 還具有後台檢查和兼容性幫助程序，可防止您保存無效或不兼容的 SDF 文件。

如果要編寫自定義應用程序，SessionConf 可以導出某些編程語言的頭文件。這些頭文件包含名為項目名稱和 SDF 文件索引的常量。

可以導出以下項索引：

- 節點
- 時間表
- 框架
- 部分
- 信號
- 宏

## 12.5 SimpleMenu

SimpleMenu 让您可以轻松访问 HARP-5 并控制它。对于 Baby-LIN-DLL 来说，它基本上是一个舒适且对用户友好的 GUI。

SimpleMenu 使您可以访问以下功能：

- 管理所有连接的 Baby-LIN 产品。
- 管理 Baby-LIN 产品的所有通道。
- 加载 SDF 文件并控制总线。
- 在监视模式下使用不带 SDF 文件的 Baby-LIN 产品，可以观察总线通信。
- 使用自定义的图形用户界面查看和设置信号或是启动宏。
- 在帧监视器中观察总线通信。
- 设置目标配置。
- 查看和添加激活码。
- 直接向 Baby-LIN 产品发送命令。

等等

## 12.6 LogViewer

LogViewer 允许查看记录数据和导出信号数据。它可以以“二进制格式”打开记录文件，其中包含基于帧的数据。

要查看信号值，LogViewer 需要导入通信所使用的 SDF 文件。这样就可以通过选择信号和时间网格来过滤并显示相关数据。这些数据也可以导出为 CSV 文件。

下表显示了典型的操作顺序：

步骤	操作方法	输出
加载记录数据	<ul style="list-style-type: none"> <li>从菜单中选择打开记录文件。</li> <li>选择一个“二进制格式”的记录文件。</li> </ul>	<ul style="list-style-type: none"> <li>“记录信息”选项卡中的“记录文件”显示记录文件的摘要。</li> <li>“原始记录文件帧数据”选项卡会显示原始帧的记录数据。</li> </ul>
加载 SDF 文件	<ul style="list-style-type: none"> <li>从菜单中选择打开 SDF 文件。</li> <li>从浏览窗口中选择需要 SDF 文件。</li> </ul>	<ul style="list-style-type: none"> <li>“记录信息”选项卡中的“SDF 文件”显示 SDF 文件的摘要。</li> <li>“报告”菜单现已可用。</li> </ul>
创建一个信号报告	<ul style="list-style-type: none"> <li>从菜单中选择新信号报告。</li> <li>选择一个选项： <ul style="list-style-type: none"> <li>在每次帧出现时列出所选信号。</li> <li>间隔列出所选信号的值。</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>打开“新建报告”选项卡。</li> </ul>
配置信号报告	<ul style="list-style-type: none"> <li>选择“新建报告”选项卡。</li> <li>在左侧选择需要的信号。</li> <li>仅适用于使用时间间隔的报告：编辑时间间隔。</li> <li>单击左下角的更新按钮。</li> </ul>	<ul style="list-style-type: none"> <li>信号值和改动显示在右侧。</li> </ul>
导出信号报告	<ul style="list-style-type: none"> <li>从菜单中选择导出当前报告 (CSV)。</li> <li>输入想要的 CSV 文件储存路径。</li> </ul>	<ul style="list-style-type: none"> <li>信号报告将保存为 CSV 文件。</li> </ul>

## 12.7 Baby-LIN-DLL

Baby-LIN-DLL 是用 C/C++编写的本机动态链接库，允许自定义应用程序控制 HARP-5。

Baby-LIN-DLL 可用于任何本机客户应用程序，主要是 C/C++应用程序。还提供用于.NET 应用程序和 LabView 的封装。请查看“Baby-LIN-DLL.NET Wrapper”和“Baby-LIN-DLL LabView Wrapper”一章获取更多信息。

Baby-LIN-DLL 可以实现 SimpleMenu 的所有功能以及更多：

- 管理所有连接的 Baby-LIN 产品。
- 管理 Baby-LIN 产品的所有通道。
- 加载 SDF 文件并控制总线。
- 在监视模式下使用不带 SDF 文件的 Baby-LIN 产品，可以观察总线通信。
- 查看和设置信号。
- 启动宏并评估结果。
- 接收完整的总线通信。
- 直接向 Baby-LIN 产品发送命令。
- 在没有 LDF 文件或 SDF 文件的情况下注入帧。
- 使用更高级的协议，如 DTL 等。

等等

Baby-LIN-DLL 可用作 x86 和 x64 Windows DLL。需要的话，我们也可以提供 Linux 版本。

## 12.8 Baby-LIN DLL .NET 封装

Baby-LIN DLL .NET 封装是一个.NET 库，可以轻松访问 Baby-LIN DLL 的功能。

它保留了 Baby-LIN DLL 的 API，只需要简单的语法相关调整。

## 12.9 Baby-LIN DLL LabView 封装

Baby-LIN-DLL LabView 封装允许您从 LabView 中访问 Baby-LIN DLL。

## 12.10 Serial writer

Serial writer 用于更改存储在 HARP-5 的持久存储器中的序列号。序列号会影响虚拟 COM 端口号的分配。

**Advice:** 序列号不会阻止在一台 PC 上使用多个 HARP-5。它只会影响如何分配虚拟 COM 端口。

以下示例给出如果 HARP-5 使用相同和不同的序列号，如何分配虚拟 COM 端口。请注意，下面给出的虚拟 COM 端口号仅仅是一个示例，在每台 PC 上可能有所不同。

	HARP-5 A	HARP-5 B	HARP-5 C	HARP-5 D
序列号	0001	0001	0002	0003
只有 A 连接	COM1			
只有 B 连接		COM1		
只有 C 连接			COM10	
只有 D 连接				COM15
按照以下顺序连接: A,B,C,D	COM1	COM2	COM10	COM15
按照以下顺序连接: D,C,B,A	COM2	COM1	COM10	COM15

使用该工具非常简单。启动时，它会显示当前的序列号，并允许您编写新的序列号。您所要做的就是确保只有一台 Baby-LIN 产品连接到您的 PC。

**Warning: Serial writer** 只允许您在只有一台 Baby-LIN 产品连接到您的 PC 上时更改它的序列号。如果找不到设备或有两个以上，则 **Serial writer** 会显示错误消息。

**Advice:** 在 HARP-5 与 USB 端口断开并重新连接后，将分配新的虚拟 COM 端口。

如果您对所有 HARP-5 执行该过程，则每个 HARP-5 将始终接收其单独的虚拟 COM 端口。

## 12.11 BLProg

BLProg 是一个小型命令行工具，可让您更新 HARP-5 的固件。官网下载固件包时会始终包含它。

## 12.12 Switch to ASCII tool

Switch to ASCII tool 是一个小命令行工具，可让您将 HARP-5 切换到 ASCII 模式。请查看“ASCII 模式”一章获取更多信息。

# 13. 协议和应用

## 13.1 自定义协议

### 13.1.1 介绍

**Version incompatibility:**

此功能仅在 SDF V3 可用。

根据您的设备，您可能需要额外的激活码。

需要更新最新固件。

在需要诊断通信场合，我们支持使用自定义的协议，它们可在设备独立运行时使用。这样就可以在无需 PC 的情况下评估样品的反馈。因此，可以从 SDF 内的宏调用诊断序列来查询诸如 LIN 总线节点的硬件编号和软件版本之类的值，并将它们与存储的值进行比较。如果查询的值与存储的值不匹配，则可以例如：拒绝该样本为 NOK。这同时防止了错误安装的风险。

通过多种配置选项，可以在 SDF 中直接自定义协议，并通过宏调用执行，如 LIN 诊断传输层 (DTL)，统一诊断服务 (UDS) 或 Cooling 协议。专用的诊断协议也是可实现的，这时模式可以设置为 raw 模式以及 DTL。后者只需要有效载荷的定义。对数据的传输信息的分段和补充由 Baby-LIN 处理。

### 13.1.2 需求

自定义协议功能支持 LIN 和 CAN 总线。但是现在它只能在 LIN 总线上使用，需要 HARP-5 仿真主节点。

使用自定义协议和服务需要在 LIN 总线上执行调度表。在执行自定义协议的服务时，调度表会被暂停。如果您的应用程序仅需要执行协议的服务，则计划表中的帧可能会出现问題。这里推荐使用一个技巧，它由两部分组成：

- 用于生成 SDF 文件的 LDF 文件需要有一个仅包含主请求帧的调度表。
- 设备特定选项“Master Request on IDLE”必须设置为“Silent frame”。

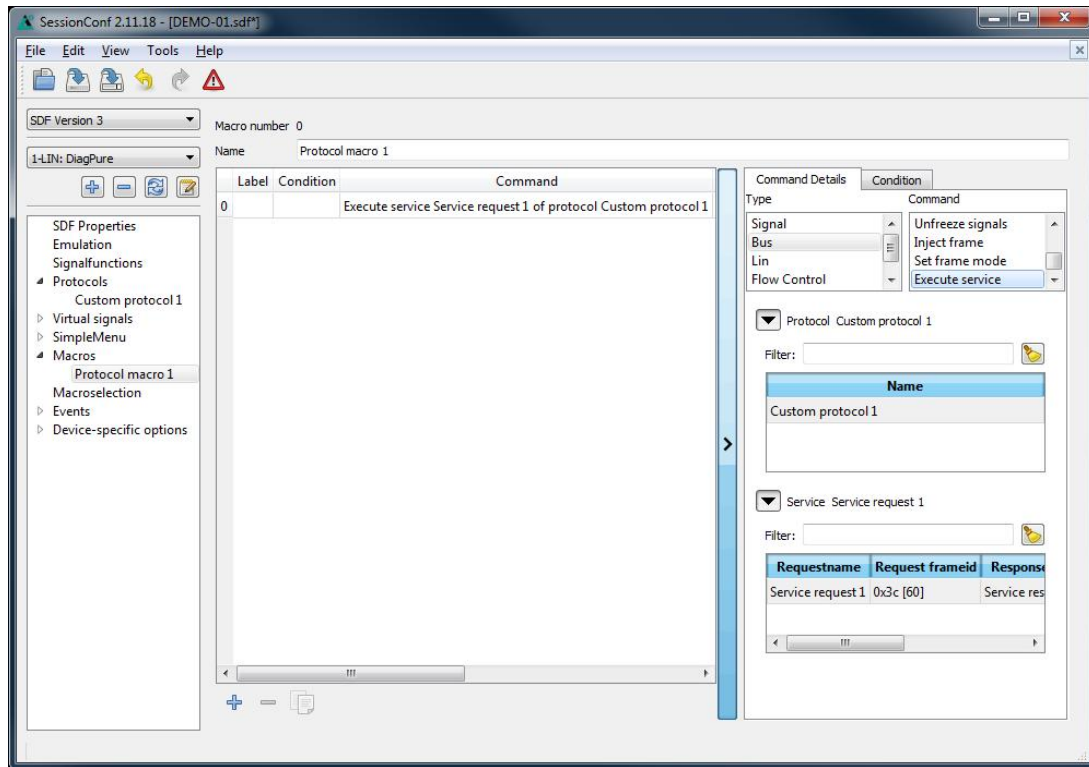
这种组合导致调度表不发送任何帧。仅当主请求帧的信号改变时才触发主请求，并且由于“Silent frame”选项，它将不会在 LIN 总线上被看到。这意味着调度表正在运行，但总线上不会出现任何帧。

### 13.1.3 执行协议服务

要执行自定义协议的服务，您必须去触发宏命令。

宏命令是“总线”命令，名为“执行服务”。作为参数，您必须从该协议中选择要执行的协议和服务。

要触发包含此命令的宏，您可以使用自动启动功能或任何类型的事件。



**Advice:** “执行服务”宏命令会导致阻塞。这意味着在总线发送所有必需的帧之前不会返回。

本地虚拟信号“\_ResultLastMacroCommand”将包含执行后执行服务的宏命令的结果。其中定

义了以下错误信息:

错误代码	描述
0	该服务已成功执行。
1	HARP-5 的存储器已满。这是一个内部错误。



2	<p>此错误可能是由以下原因之一引起的：</p> <ul style="list-style-type: none"> <li>• 无 LIN 电源电压。</li> <li>• 未启动总线。</li> <li>• LIN 总线上没有运行调度表。</li> </ul>
3	一个从节点没有及时响应。
5	如果使用 DTL 协议，则 CF（连续帧）的索引是错误的。
6	如果使用 DTL 协议，则 NAD 不是一个合理的值。
7	如果使用 DTL 协议，则帧类型不是一个合理的类型。SF（单帧），FF（第一帧）和 CF（连续帧）不符合预期。
8	客户特定扩展的错误。
9	
10	
11	同时执行的服务太多了。
12	帧数据的长度不是预期的。
255...511	如果使用 DTL 协议，则从节点设备可能返回了负响应帧。然后返回该负响应帧的错误值，偏移量为 0x100。

### 13.1.4 协议相关的系统变量

以下系统变量与协议功能相关：

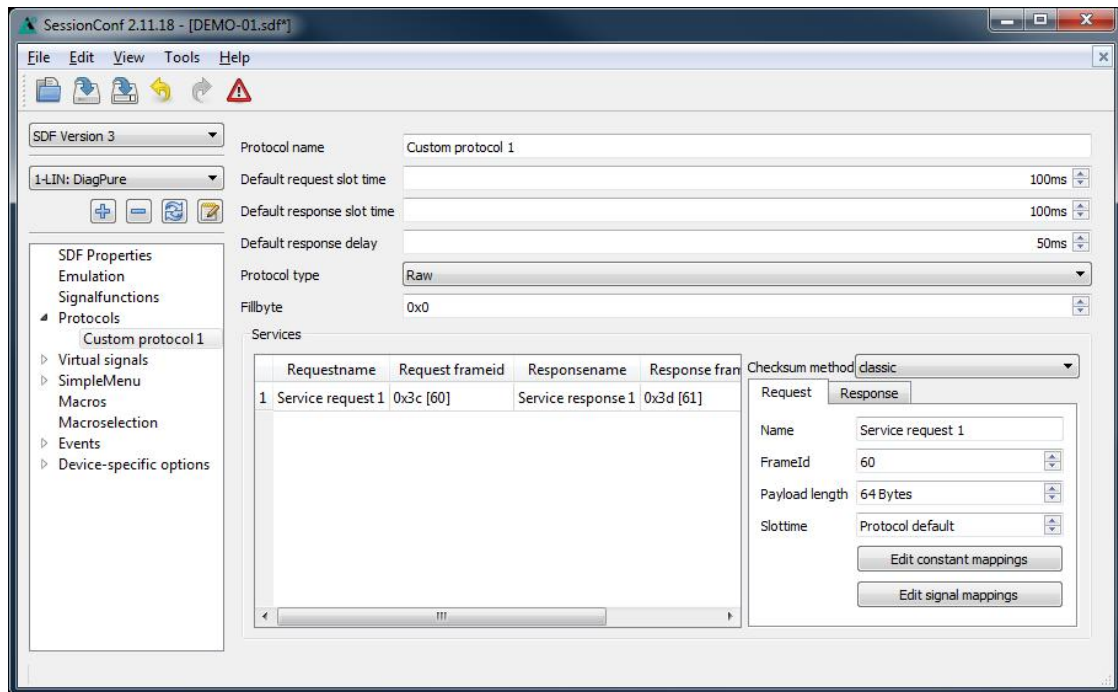
系统变量	描述
@@SYS_SERVICE_REQUEST_NAD	包含构造请求帧时将使用的 NAD。
@@SYS_SERVICE_RESPONSE_NA	包含收到回复后的 NAD。

D	
@@SYS_SERVICE_RESPONSE_LEN	包含收到响应后响应有效负载的长度。如果协议类型是“Raw”，则有效负载长度等于数据长度。如果有效负载长度是可变的，则该值需要经常关注。

另外，当在单个宏内调用多个服务时，宏变量才是有意义的。这些系统变量是宏的本地变量，只能在此宏中访问，您无法在宏之外访问其值。请查看“宏变量”一章获取更多信息。

### 13.1.5 协议设置

SessionConf 中的所有协议设置均可通过所有 LIN 和 CAN 部分中的“Protocols”部分实现。



### 13.1.6 服务属性

服务具有以下请求和响应属性，每个属性包括一个常量和信号映射列表：

服务属性		值类型	描述
校验和方法(LIN)		经典	使用经典校验和。
		增强	使用 LIN 2.0 的增强校验和。
帧 ID 长度(CAN)		11 位	帧 ID 的长度为 11 位。
		29 位	帧 ID 的长度为 29 位。
请求	名称	字符串	这是协议中的请求的名称。它用于引用 SDF 内的协议。
	帧 ID	LIN: 6 位 CAN: 11 或 29 位	请求帧的 ID。
	有效载荷长度	整型(1-4095)	有效载荷数据的长度。
	时隙时间	时间(ms)	请求的时隙时间是为发送请求帧而保留的时间。您可以在此处使用协议的默认请求时隙。
	常量映射	映射	此映射将常量映射到有效载荷中。有关更多信息，请查看“常量映射”。
	信号映射	映射	该映射将信号映射到有效载荷中。有关更多信息，请查看“信号映射”。
响应	名称	字符串	这是协议中响应的名称。它用于引用 SDF 内的协议。
	帧 ID	LIN: 6 位 CAN: 11 或 29 位	响应帧的 ID。
	有效载荷长度	整型(1-4095)	有效载荷数据的长度。
	间隙时间	时间(ms)	响应的时隙时间是为发送响应帧而保留的时

			间。在这里，您可以使用协议的默认响应时间。
	延时	时间(ms)	这是主设备等待从设备响应的最大延迟。如果从站未在时间内应答，则宏返回值将发出超时信号。
	常量映射	映射	此映射将常量映射到有效载荷中。有关更多信息，请查看“常量映射”。
	信号映射	映射	该映射将信号映射到有效载荷中。有关更多信息，请查看“信号映射”。

### 13.1.7 常量映射

常量映射将常量映射到帧中。它由以下属性定义：

属性	描述
开始位置(位)	在帧中的开始位置。
长度(位)	常量数据的长度。
映射数据	写入有效载荷的值。

**Warning:** 起始位置和结束位置（起始位置+长度）都必须在帧的有效载荷内。

### 13.1.8 信号映射

信号映射由多路复用器列表组成，每个多路复用器包含映射列表。

多路复用器会决定应用哪一组信号映射。需要存在一个可用的静态多路复用器，默认应用该多路复用器。对于其它的多路复用器，如果信号值等于它的值，则应用这个多路复用器。

多路复用器属性	描述
信号	该信号的值将用于比较。
值	该值将与实际信号值进行比较。

**Advice:** 多路复用器会从顶部到底部依次对照。在应用以前的多路复用器后，多路复用器的条件可能会改变。

映射描述帧的哪些数据被映射到哪个信号。在请求中，信号被映射到帧数据；在响应中，帧数据被映射到信号。仅当父多路复用器的信号和值相等或父多路复用器是静态多路复用器时才应用映射。

映射属性	描述
信号	该信号是映射的帧数据的源/目标。 Version incompatibility 只有虚拟信号可用于映射。
偏移量	这是从有效载荷的开头到数据最低有效位的偏移量。
字节顺序	映射数据的字节顺序。
位顺序	映射数据的位顺序。

**Advice:** 映射的位数取决于虚拟信号的长度。

**Advice:** 映射会从顶部到底部依次对照。如果多次使用同一信号，则最后一次映射将定义信号的值。

### 13.1.9 字节顺序和位顺序

#### 13.1.9.1 简介

CAN 支持英特尔，摩托罗拉，顺序和锯齿的信号映射。以下部分描述了如何从帧数据中读取信号（写入信号）。

在考虑存储器中的字节范围时，可以区分字节顺序和位顺序。“英特尔”和“摩托罗拉”表示字节顺序，“顺序”和“锯齿”表示位顺序。由于位和字节排序彼此正交，因此它们可以两两组合，导致数据存储在内存中的方式有以下四种类型：

- 英特尔锯齿
- 英特尔顺序
- 摩托罗拉顺序
- 摩托罗拉锯齿

下表显示了每种组合在内存中的数据位置：

	Sawtooth								Sequential											
Intel	Byte	Bit								Bit										
			7	6	5	4	3	2	1	0		0	1	2	3	4	5	6	7	
		0	7	6	5	4	3	2	1	0	0	1	2	3	4	5	6	7		
		1	15	14	13	12	11	10	9	8	1	8	9	10	11	10	13	14	15	
		2	23	22	21	20	19	18	17	16	2	16	17	18	19	20	21	22	23	
		3	31	30	29	28	27	26	25	24	3	24	25	26	27	28	29	30	31	
		4	39	38	37	36	35	34	33	32	4	32	33	34	35	36	37	38	39	
		5	47	46	45	44	43	42	41	40	5	40	41	42	43	44	45	46	47	
6	55	54	53	52	51	50	49	48	6	48	49	50	51	52	53	54	55			
7	63	62	61	60	59	58	57	56	7	56	57	58	59	60	61	62	63			
Motorola	Byte	Bit								Bit										
			7	6	5	4	3	2	1	0		0	1	2	3	4	5	6	7	
		7	63	62	61	60	59	58	57	56	7	56	57	58	59	60	61	62	63	
		6	55	54	53	52	51	50	49	48	6	48	49	50	51	52	53	54	55	
		5	47	46	45	44	43	42	41	40	5	40	41	42	43	44	45	46	47	
		4	39	38	37	36	35	34	33	32	4	32	33	34	35	36	37	38	39	
		3	31	30	29	28	27	26	25	24	3	24	25	26	27	28	29	30	31	
		2	23	22	21	20	19	18	17	16	2	16	17	18	19	20	21	22	23	
1	15	14	13	12	11	10	9	8	1	8	9	10	11	12	13	14	15			
0	7	6	5	4	3	2	1	0	0	0	1	2	3	4	5	6	7			

当数据以这些组合之一存储并通过 LIN 或 CAN 总线传输时，接收方必须去解释这些数据。接收器要从数据中提取信号，必须知道发送者最初如何映射数据。反之亦然，发送方可能需要在特定映射中映射信号以将其发送到接收方。通常，应该可以直接说明帧的信号映射属于哪四种类型。不过，LDF 规范不允许指定类型，并以 Intel 锯齿为默认值。DBC 规范区分了涉及锯齿位排序的 Intel 和 Motorola 映射。

### 13.1.9.2 定义

对于帧，我们有以下信息：

- fsize: 帧大小（以字节为单位）
- fdata [fsize]: 帧数据的字节数组

我们假设 fdata 存在于英特尔锯齿符号中，其中 fdata [8] 如下所示（表格单元格中的数字表示单元格的总位数）：

		Bit							
		7	6	5	4	3	2	1	0
fdata	[0]	7	6	5	4	3	2	1	0
	[1]	15	14	13	12	11	10	9	8
	[2]	23	22	21	20	19	18	17	16
	[3]	31	30	29	28	27	26	25	24
	[4]	39	38	37	36	35	34	33	32
	[5]	47	46	45	44	43	42	41	40
	[6]	55	54	53	52	51	50	49	48
	[7]	63	62	61	60	59	58	57	56

我们看到，帧数据以英特尔锯齿符号表示，因此与英特尔锯齿信号映射相匹配，这是 LIN 和 CAN 英特尔映射的标准配置。对于映射到帧的信号，我们有以下数据字段：

- ssize: 以位为单位的信号大小。
- sstartbit: fdata 中位的信号最低有效位的位置。

**Advice: fdata 仍然是英特尔锯齿格式。**

- sbyteorder: 指定 Motorola 或 Intel 映射。
- sbitorder: 指定顺序或锯齿映射。
- svalue: 信号的值（待确定或映射）。

### 13.1.9.3 英特尔锯齿映射

在英特尔锯齿映射中，信号以与可用 fdata 相同的字节和位排序映射，这使得映射非常容易。假

设信号映射使用以下参数：

- ssize = 12
- sstartbit = 10
- sbyteorder = Intel
- sbitorder = sawtooth

在帧中拥有以下参数：

- fsize = 4

信号映射如下：

		Bit							
		7	6	5	4	3	2	1	0
fdata	[0]	7	6	5	4	3	2	1	0
	[1]	15 ---	14 ---	13 ---	12 ---	11 ---	10 -LSB]	9	8
	[2]	23	22	21 [MSB-	20 ---	19 ---	18 ---	17 ---	16 ---
	[3]	31	30	29	28	27	26	25	24

如果此数据集如上所述通过总线传输并以此为定义，则它们“看”起来完全相同。读取信号非常简单，因为可以使用移位运算符和逻辑运算符提取信号。



### 13.1.9.4 英特尔顺序映射

尽管它不适用于 LDF 或 DBC 规范，但我们仍然将讲解英特尔顺序映射。

我们再次假设一个 12 位信号，最低有效位映射到位 10，这次是 4 字节的 Intel 顺序帧数据：

		Bit							
		0	1	2	3	4	5	6	7
fdata_seq	[0]	0	1	2	3	4	5	6	7
	[1]	8	9	10 [LSB-	11 ---	12 ---	13 ---	14 ---	15 ---
	[2]	16 ---	17 ---	18 ---	19 ---	20 ---	21 -MSB]	22	23
	[3]	24	25	26	27	28	29	30	31

当此数据集如上所示发送但解释为定义时，生成的 fdata 集如下所示：

		Bit							
		7	6	5	4	3	2	1	0
fdata	[0]	7	6	5	4	3	2	1	0
	[1]	15	14	13 [LSB-	12 ---	11 ---	10 ---	9 ---	8 ---
	[2]	23 ---	22 ---	21 ---	20 ---	19 ---	18 -MSB]	17	16
	[3]	31	30	29	28	27	26	25	24

- ssize = 12
- sstartbit = 13 (LSB)
- sbyteorder = Intel
- sbitorder = sequential
- fsize = 4

为了提取信号，我们希望以信号数据易于提取的方式对数据进行排序。为此，可以尝试逐位反转

存储器中的每个字节。结果如下所示：

		Bit							
		7	6	5	4	3	2	1	0
fdata	[0]	7	6	5	4	3	2	1	0
	[1]	15 ---	14 ---	13 ---	12 ---	11 ---	10 -LSB]	9	8
	[2]	23	22	21 [MSB-	20 ---	19 ---	18 ---	17 ---	16 ---
	[3]	31	30	29	28	27	26	25	24

由于 LSB 位置已被翻转，因此必须通过翻转来计算 startbit\_flip:

bit = startbit % 8

byte = startbit / 8

newBit = 7-startbit

sstartbit\_flip = byte\*8+newBit

现在可以使用 sstartbit\_flip 作为信号的起始位，然后按照英特尔锯齿格式提取信号。

### 13.1.9.5 摩托罗拉锯齿映射（低位在前）

#### 13.1.9.5.1 映射

在摩托罗拉锯齿映射中，信号以与可用 fdata 不同的字节顺序和相同的位顺序映射。

假设摩托罗拉锯齿数据中有以下信号：

		Bit							
		7	6	5	4	3	2	1	0
fdata_be	[3]	31	30	29	28	27	26	25	24
	[2]	23	22	21 [MSB-	20 ---	19 ---	18 ---	17 ---	16 ---
	[1]	15 ---	14 ---	13 ---	12 ---	11 ---	10 -LSB]	9	8
	[0]	7	6	5	4	3	2	1	0

如果按照显示的顺序发送此数据（从 fdata\_be [3]到 fdata\_be [0]），我们通过定义提取的数据将

按字节顺序反转，并且帧内的位也会是镜像的。

- ssize = 12
- sstartbit = 18 (LSB)
- sbyteorder = Motorola
- sbitorder = sawtooth
- fsize = 4

这种带有摩托罗拉信号的 fdata 结构使得从数据集中提取实际信号变得更加困难。

### 13.1.9.5.2 提取 Motorola 信号

提取信号数据的一种方法是：

- 1.将 fdata 数据集反转为 fdata\_inv 作为新的低位编址数据集(将位视为 fdata\_inv 是实际的 fdata)。
- 2.重新计算 sstartbit 到 sstartbit\_inv 以满足 fdata\_inv 中的 startbit。
- 3.使用 fdata\_inv 中的 startbit\_inv 提取信号，方法同 Intel 信号。

以下章节逐步介绍此过程。

### 13.1.9.5.3 字节反转

反转 fdata

		Bit							
		7	6	5	4	3	2	1	0
fdata_inv	[3]	31	30	29	28	27	26	25	24
	[2]	23	22	21	20	19	18 -LSB]	17	16
	[1]	15	14	13 [MSB-	12	11	10	9	8
	[0]	7	6	5	4	3	2	1	0

并且看作低位编址

		Bit							
		7	6	5	4	3	2	1	0
fdata_inv	[0]	7	6	5	4	3	2	1	0
	[1]	15	14	13	12	11	10 -LSB]	9	8
	[2]	23	22	21 [MSB-	20	19	18	17	16
	[3]	31	30	29	28	27	26	25	24

### 13.1.9.5.4 重新计算 sstartbit

正如我们在上一节中看到的那样，sstartbit = 18 更改为 sstartbit\_inv = 10。fdata 的表格显示，字节行被反转（因为所有字节都被反转），但是位列保持不变。所以我们可以这样计算 sstartbit\_inv：

1. 计算在 fdata 中映射 LSB 的字节内的字节和位（即“行数”和“列数”）：

```
byte = startbit / 8; //范围内的字节[0;FSIZE]
```

```
bit = startbit % 8; //位范围[0;7]
```

在示例中，结果是 byte = 2 和 bit = 2。

2. 从帧的后面计算 sstartbyte\_inv（反转“行数”）：

```
sstartbyte_inv = (fsize-1) -byte)
```

在示例中，结果是 sstartbyte\_inv = 1

3. 通过 startbyte\_inv 和 bit（完成反转后位置）的位置来计算 sstartbit\_inv：

```
sstartbit_inv = sstartbyte_inv * 8 +bit
```

在示例中，结果是 sstartbit\_inv = 10。

计算步骤可以简化，但分步骤说明解释地更全面。现在就可以像英特尔信号映射一样提取信号了。

### 13.1.9.6 摩托罗拉锯齿映射（高位在前）

该映射的行为与低位在前时完全相同，只是 sstartbit 值表示信号的 MSB。按照低位在前情况处理后，应该额外在计算后将 sstartbit\_inv 值减小一个信号长度。

**Advice:** 如果 DBC 文件是使用摩托罗拉映射，那么它就是指这种映射方式。

### 13.1.9.7 摩托罗拉顺序映射

尽管它不适用于 LDF 或 DBC 规范，但我们仍然将讲解摩托罗拉顺序映射。

我们再次假设一个 12 位信号，LSB 映射到位置 10，如下是摩托罗拉 4 字节的顺序帧数据：

		Bit							
		0	1	2	3	4	5	6	7
fdata_seq	[3]	24	25	26	27	28	29	30	31
	[2]	16 ---	17 ---	18 ---	19 ---	20 ---	21 -MSB]	22	23
	[1]	8	9	10 [LSB-	11 ---	12 ---	13 ---	14 ---	15 ---
	[0]	0	1	2	3	4	5	6	7

像这样发送并按照 Intel 锯齿的定义解释，数据如下所示：

		Bit							
		7	6	5	4	3	2	1	0
fdata	[0]	7	6	5	4	3	2	1	0
	[1]	15 ---	14 ---	13 ---	12 ---	11 ---	10 -MSB]	9	8
	[2]	23	22	21 [LSB-	20 ---	19 ---	18 ---	17 ---	16 ---
	[3]	31	30	29	28	27	26	25	24

- ssize = 12
- sstartbit = 21 (LSB)
- sbyteorder = Motorola
- sbitorder = sequential
- fsize = 4

## 13.2 巨型帧：具有 8 个以上数据字节的 LIN 帧

### 13.2.1 简介

在 LIN 规范定义中，一帧最多拥有 8 个数据字节。某些应用程序希望使用具有更多数据字节的帧并从中受益。所以，HARP-5 可以处理具有 8 个以上数据字节的 LIN 帧。这些帧被称为巨型帧。

**Advice:** 巨型帧支持需要使用激活码激活。请访问可选的凭证代码“**Option-BL-HARP-Jumbo Frames**”（商品编号：**8000831**）。

以下产品支持巨型帧：Baby-LIN-II, Baby-LIN-RC-II, Baby-LIN-RM-II, Baby-LIN-MB-II, HARP-4, HARP-5

### 13.2.2 通过注入命令的巨型帧

巨型帧是具有 8 个以上数据字节的 LIN 帧。HARP-5 支持的最大数据字节数为 1024 字节。

HARP-5 作为主节点时可以通过注入命令发送巨型帧。它允许用户自由组合自定义帧并将其发送到 LIN 总线上。甚至可以在调度表中没有定义的情况下发送注入帧。注入帧将在下一个可能的时刻发送，这通常是在当前调度帧的时隙结束时。

**Attention: 如果调度表中断并插入了帧，则某些 LIN 节点可能会出现问題。HARP-5 允许只单独运行一次调度表。因此，可以在插入帧之前完成一次调度表运行。**

可以使用 Baby-LIN-DII 在自定义应用程序中注入巨型帧。也可以使用 SimpleMenu 在测试时注入巨型帧。

**Version incompatibility: 目前，不能在 SDF 文件中使用巨型帧，因此它们不能独立于应用程序使用。**

使用注入命令可以组合自定义帧。要发送巨型帧，需要以下信息：

- 校验和类型（经典，扩展，未定义）
- 帧 ID
- 数据字节本身

- 注入帧的时隙时间

要接收巨型帧，需要以下信息：

- 帧 ID
- 数据字节数
- 注入帧的时隙时间

最多可以连续发出 8 个注入命令用于连续发送或接收。

### 13.2.3 使用 SimpleMenu 测试巨型帧

测试发送或接收巨型帧的最简单方法是使用 SimpleMenu 的监视模式。监视模式允许 HARP-5 设备在 LIN 总线上无需 SDF 模拟主节点。在此模式下，必须通过注入帧手动创建调度表。

请按照以下步骤发送或接收巨型帧：

- 准备设备和 PC：

将 HARP-5 设备连接到 PC。

启动 SimpleMenu 并连接到 HARP-5 设备。

- 可选：激活或检查巨型帧选项：

右键单击设备列表中的 HARP-5 设备。

选择“Activation codes”。

输入激活码以启用对巨型帧的支持或检查列表中的“Jumbo-Frame Support: enabled”。

- 作为主节点启用监控模式：

选择 LIN 通道窗口/选项卡。

从“窗口”菜单中选择“Channelshell ...”打开 channel shell。

在命令提示符下输入 `mon_init 20000 1` (20000 是波特率, 1 表示信号 HARP-5 设备是主节点)。

- 打开帧监视器。在这里, 您可以看到 LIN 总线上的所有帧。

从“窗口”菜单中选择“帧监视器...”打开帧监视器。

- 注入巨型帧:

输入 `inject 1 0x15 [0x0 0x1 0x2 0x3 0x4 0x5 0x6 0x7 0x8 0x9 0xA 0xB 0xC 0xD 0xE 0xF] 0 0` 以注入

具有以下属性的巨型帧:

校验和: 1 (经典校验和)

帧 ID: 0x15 (十进制 21)

16 个数据字节: [0x0 0x1 0x2 0x3 0x4 0x5 0x6 0x7 0x8 0x9 0xA 0xB 0xC 0xD 0xE 0xF]

以[ms]为单位的时隙时间: 0 (0: 自动时隙时间。非 0: 在这段时间结束后, 下一帧将立即发送。)

不再等待发出的任何注入帧: 0

- 注入巨型帧请求:

输入 `inject 4 0x15 16 [ ] 0 0` 以注入具有以下属性的巨型帧请求:

从从节点请求一个帧: 4

帧 ID: 0x15 (十进制 21)

预期数据字节: 16

以[ms]为单位的时隙时间: 0 (0: 自动时隙时间。非 0: 在这段时间结束后, 下一帧将立即发送。)

不再等待发出的任何注入帧: 0



- 检查帧监视器上显示的 LIN 总线上的巨型帧:

Timestamp	FrameId	FrameData	Checksum	State
+377364	0x05 [0x85]	0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0a 0x0b 0x0c 0x0d 0x0e 0x0f	0x87	Other (injected)

### 13.2.4 使用 Baby-LIN-DII 自定义应用程序

自定义应用程序是使用巨型帧的最佳方式，因为您能够以编程方式加载和处理大量数据。

HARP-5 设备与自定义应用程序之间的通信由 Baby-LIN-DII 处理。

**Advice:** Baby-LIN-DII 可以在用于 C / C++编写的自定义应用程序。同样可以用于.NET 框架和 LabView。

有两种可用于注入帧的模式:

模式	描述
使用 SDF 文件的模式	在此模式下需要加载 SDF 文件。SDF 包含来自 LDF 的调度表信息。调度表可以单步执行或循环执行。巨型帧可以在调度表运行时注入。
作为主节点的监控模式	在此模式下，没有 SDF 文件。必须通过手动注入帧创建调度表。

在这两种情况下，应用程序通常会从这些函数开始:

Baby-LIN-DII 函数	描述
BLC_getVersionString()	此函数返回 Baby-LIN-DLL 的版本。在寻求技术支持时请告知该版本号。
BLC_getBabyLinPorts()	此函数返回有关连接到 PC 的 Baby-LIN 产品的信息。

BLC_open()	此函数启动与 HARP-5 设备的连接。
BLC_getChannelCount() BLC_getChannelHandle() BLC_getChannelInfo()	这些函数为您提供有关可通过 HARP-5 设备使用的通道信息。
BLC_registerJumboFrameCallback()	此函数注册一个回调以接收巨型帧。

如果要使用 SDF，则需要调用以下函数：

Baby-LIN-DII 函数	描述
BLC_loadSDF()	此函数将 SDF 加载到 DLL 中，然后加载到 HARP-5 设备。
BLC_sendCommand("disframe 0xC0000000 1;")	此命令表示接收部分 ID 的帧（0xC0000000 表示所有 ID）。
BLC_sendCommand("start;")	此命令将以第一个调度表启动 LIN 总线。

如果不想使用 SDF 但需要使用监控模式，则需要调用以下函数：

Baby-LIN-DII 函数	描述
BLC_sendCommand("mon_init 20000 1;")	此命令将设备作为主节点启动监控模式。
BLC_sendCommand("mon_on 0xC0000000 1;")	此命令表示接收部分 ID 的帧（0xC0000000 表示所有 ID）。

现在就可以注入巨型帧或巨型帧请求了：

Baby-LIN-DII 函数	描述
BLC_sendCommand("inject 1 0x15 [0x0 0x1 0x2 0x3 0x4 0x5 0x6 0x7 0x8 0x9 0xA 0xB 0xC 0xD 0xE 0xF ] 0 0;")	<p>此命令会注入具有以下属性的巨型帧：</p> <ul style="list-style-type: none"> <li>• 经典校验和： 1</li> <li>• 帧 ID： 0x15 (21)</li> <li>• 16 个数据字节： [0x0 0x1 0x2 0x3 0x4 0x5 0x6 0x7 0x8 0x9 0xA 0xB 0xC 0xD 0xE 0xF]</li> <li>• 以[ms]为单位的时隙时间： 0 (0： 自动时隙时间。非 0： 在</li> </ul>

	<p>这段时间结束后，下一帧将立即发送。)</p> <ul style="list-style-type: none"> <li>不再等待发出的任何注入帧：0</li> </ul>
<pre>BLC_sendCommand("inject 4 0x15 16 [ ] 0 0;");</pre>	<p>此命令会注入具有以下属性的巨型帧请求：</p> <ul style="list-style-type: none"> <li>从从节点请求一个帧：4</li> <li>帧 ID：0x15（十进制 21）</li> <li>预期数据字节：16</li> <li>以[ms]为单位的时隙时间：0（0：自动时隙时间。非 0：在 这段时间结束后，下一帧将立即发送。)</li> <li>不再等待发出的任何注入帧：0</li> </ul>

### 13.3 诊断模式

#### 13.3.1 支持的诊断模式

LIN 规范定义了一些特定帧和协议用于诊断。

一般来说有两个帧可用于诊断：

帧 ID	描述	缩写
0x3C	主请求帧	MRQ
0x3D	从响应帧	SRP

这些帧具有一些特殊属性需要注意：

- 数据场一定带有 8 个数据字节。
- 只使用经典的校验和方法。
- 第一个数据字节设置为零的 MRQ 帧表示总线的休眠命令。

与所有其他帧一样，MQR 和 SRP 帧必须包含在调度表中以在总线上发送。

因此，本手册中描述的所有功能都需要先在 HARP-5 中安装有效的 SDF 文件配置。此 SDF 必须至少定义 2 帧：主请求帧（0x3C）和从响应帧（0x3D），并且必须有一个调度表才能使用这些帧。

由于这些帧通常将与某种特定诊断协议一起使用，因此这些帧的数据字节不能通过标准 LINWorks GUI 界面（SimpleMenu）使用。

相反，我们提供了一些 API 函数，允许通过 API 调用 Baby-LIN-DLL 来发送和接收这些帧。

不同的诊断模式，可以提供不同级别的协议功能：

诊断模式	协议功能
关闭诊断	没有数据传输。
RAW 模式	发送主请求帧并接收到可设置数量的从响应帧。
DTL 模式	根据 LIN 协议 V.2.x 实现诊断传输层。

在所有模式中，都允许利用 MRQ 帧发送休眠命令。

上电时，HARP-5 的诊断模式默认为“关闭诊断”。

请调用适当的 API 函数来选择诊断模式。

API 函数 BLC\_sendRawMasterRequest 或 BLC\_sendRawSlaveResponse 将启动诊断 RAW 模式，而

API 函数 BLC\_sendDTLRequest 或 BLC\_sendDTLResponse 将启动诊断 DTL 模式。

**Version incompatibility: 请注意，不允许同时使用不同的诊断模式!**

**使用 HARP-5 的固件版本 V3.60，引入了一个新命令，该命令允许在不使用上述 API 函数的情况下定义所请求的诊断模式。**

**新命令的用途在于，某些情况下，必须在调用上述 API 函数之前定义和建立诊断模式。**

### 13.3.1.1 命令 dtlmode

命令 dtlmode 允许直接定义 HARP-5 的诊断模式。

dtlmode 0	停止使用诊断函数。
dtlmode 1	初始化诊断函数以支持 raw 诊断模式。
dtlmode 2	初始化诊断函数以支持 cooked 诊断模式。

### 13.3.1.2 诊断关闭

基本上，此模式不支持使用主请求以及从响应帧进行任何数据传输。

但是，用户可以指定在 HARP-5 仿真主节点并且主请求帧出现在调度表中时，如何去处理主请求帧。

配置选项在 SimpleMenu 的 HARP-5 菜单中的“Target Configuration”条目。配置会在 HARP-5 中永久储存。

#### 13.3.1.2.1 诊断关闭时的主请求帧处理

配置为虚拟数据	发送所有数据字节为 0xFF 的主请求帧。
配置为静音数据	不发送帧，但会处理时隙。

它的出厂默认值为静音帧。

#### 13.3.1.2.2 诊断关闭时的从响应帧处理

如果在调度表中有从响应帧，则将会发送它的报头。注意正常 HARP-5 会将帧没有响应报告为错误，但这是个例外，总线没有节点响应从响应帧时，也不会产生错误。

### 13.3.1.3 RAW 模式诊断

此模式允许发送主请求帧，在 API 定义它的所有数据字节。用于接收从属请求的 API 调用也将返回从属响应数据帧中包含的所有数据字节（raw 数据=>raw 模式）。

#### 13.3.1.3.1 BLC\_sendRawMasterRequest

在 raw 模式下发送主请求的 API 函数是：

```
int BLC_sendRawMasterRequest (BL_HANDLE handle, char* data, int count);
```

指定发送的（非 DTL）主请求帧中的 8 字节数据。

首先内部的 raw 从节点响应缓存重置，并指示 HARP-5 报告下一个从响应帧的数量，然后将其按顺序写入到缓存中，响应的报文可以通过 BL\_getRawSlaveResponse () 查询。

handle 参数	用于区别通道的句柄；由 BLC_getChannelhandle()函数返回。
data 参数	主请求帧数据（8 个字节）。
count 参数	预期的从响应帧数量。
返回值	函数运行状态，返回 0 表示无错误。

调用函数指定的数据字节将随着下一个主请求帧的调度一起发送。

在传输主请求帧之后，连续计数的从响应将被捕获并排列在字符串数组中。

**Advice:** HARP-5设备中处理主请求和从响应时，将与选择的调度表不同步。

**Version incompatibility:** 所以你必须确保这一点

- 1.目标上正在运行包含主请求帧和从响应帧的调度表。
- 2.如果发出多个主请求帧，则必须检查 BLC\_sendRawMasterRequest (...) 函数的返回值是否为 BL\_OK (= 0) 。

所有不为 0 的返回值代码表示未能成功处理先前发出的主请求命令, 因此会拒绝新的主请求命令。

这种情况一般不被视为错误, 但需要通过重试 BLC\_sendRawMasterRequest 来处理。

在这种情况下, 只要返回值为不为 0, 应用程序就应该再次调用 BLC\_sendRawMasterRequest()。

另一方面, 执行此重试操作的循环应该具有某种超时条件, 使应用程序可以处理主请求永远不会发送的情况: 例如, 因为选择了错误的调度表。

在这种情况下, 可以通过发出 dtlmode 0;命令后跟一个 dtlmode 1;命令来重置目标中的状态机。

### 13.3.1.3.2 BLC\_getRawSlaveResponse

可以调用 API 函数检索从响应的值:

```
int BLC_getRawSlaveResponse (BL_HANDLE handle, unsigned char* data, int length);
```

指定发送的 (非 DTL) 主请求帧中的 8 字节数据。

从响应帧返回的数据字节被连续写入内部的 raw 从节点响应缓存, 该缓存在每次调用 BLC\_sendRawMasterRequest()时被重置。

handle 参数	用于区别通道的句柄; 由 BLC_getChannelhandle()函数返回。
data 参数	指向 char 数组的指针 (字节数和 length 参数保持一致)。
Length 参数	返回的字节数。
返回值	函数运行状态, 返回 0 表示无错误。

### 13.3.1.3.3 BLC\_sendRawSlaveResponse

如果 HARP-5 用于模拟从节点, 则可以进行设置以响应特定的主请求帧。

这是通过 API 函数完成的:

```
int BLC_sendRawSlaveResponse (BL_HANDLE handle, unsigned char * reqdata , unsigned char * reqmask, unsigned char *data, int length);
```

handle 参数	用于区别通道的句柄；由 BLC_getChannelhandle()函数返回。
reqdata 参数	指向具有主请求数值的 8 字节数组（例如 MRQVal [8]）。
reqmask 参数	指向具有主请求掩码的 8 字节数组（例如 MRQMask [8]）。
data 参数	指向一个<len>字节长度的数组，其中包含将作为从响应发送的数据。  len 应该是 8 的倍数。（8,16,24 .....）。  如果 len 大于 8，则响应将在连续的从响应帧中发送。
len 参数	从响应的长度（应为 8 的倍数）。
返回值	函数运行状态，返回 0 表示无错误。

两个指针 reqdata 和 reqmask 指向一个 8 字节数组。这些数组提供的数据将用于过滤主请求帧。

例：

```
char MRQVal [8] = {0x55,0,0,0,0,0,0,0};
```

```
char MRQMask [8] = {0xff,0,0,0,0,0,0,0};
```

通过此滤波器设置，HARP-5 将在接收第一个数据字节= 0x55 的主请求帧后触发从响应。

```
char MRQVal [8] = {1,2,3,4,5,6,7,8};
```

```
char MRQMask [8] = {0xff,0,0,0,0,0,0,0xff};
```

通过此滤波器设置，HARP-5 将在接收第一个数据字节= 1，最后一个数据字节= 8 的主请求帧后

触发从响应。

在主请求成功匹配时，则 HARP-5 将按照在数组中定义好的数据发送从响应。响应的长度由参数

len 给出。

只要没有未处理的 API 函数，HARP-5 关于主请求帧的行为由目标配置命令（静音帧或虚拟帧）

定义。



#### 13.3.1.3.4 范例

在这里，将模拟一个主节点和一个从节点之间的两组主请求与对应的从响应，包括发送数据和监视结果。

**Advice:** 这里不检查API调用的返回值以使示例更易读，在实际应用程序中，您应检查返回值以确保操作成功。

```
unsigned char masterreq[8] = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08};
// --> Databytes of first Master Request

unsigned char masterreq1[8] = {0x09, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08};
// --> Databytes of second Master Request

unsigned char response[8];
// --> buffer to store slave response

unsigned char MRQMask[8]= {0xff,0,0,0,0,0,0,0xff};
// --> Mask for masterrequest[]

unsigned char responsedata [8] = {0x08,0x07,0x06,0x05,0x04,0x03,0x02,0x01};
// --> Response of Slave for masterrequest[]

unsigned char responsedata1 [8] = {0x0A,0x0B,0x0C,0x0D,0x0E,0x0F,0x10,0x11};
// --> Response of Slave for masterrequest1[]

BL_HANDLE conhandle [2];
// --> Connectionhandle of 2 HARP-5 devices

BL_HANDLE chanhan[2];
```

```

// --> Channelhandle of the LIN-Channel of each device

int comports[2] = {8, 9}; // COM-port of devices

int masterrequest; // Return Value for BLC_sendRawMasterrequest()

conhandle[0] = BLC_open(comports[0]); // open both HARP-5 devices and get its handle
conhandle[1] = BLC_open(comports[1]);

chanhan[0] = BLC_getChannelHandle(conhandle[0], 1); // Get the handle of the dedicated.
chanhan[1] = BLC_getChannelHandle(conhandle[1], 1); // channel of each device

BLC_loadSDF (chanhandle[0], " DiagnosticSample.sdf", 1); // load SDF in the dedicated
BLC_loadSDF (chanhandle[1], " DiagnosticSample.sdf", 1); // channel of each device

BLC_sendCommand(chanhan[0], "setnodesimu 0 0"); // set Masternode
BLC_sendCommand(chanhan[0], "setnodesimu 1 1"); // for first device
BLC_sendCommand(chanhan[1], "setnodesimu 0 1"); // set Slavenode for
BLC_sendCommand(chanhan[1], "setnodesimu 1 0"); // second device.

// start schedule for both devices

BLC_sendCommand(chanhan[0], "start;");
BLC_sendCommand(chanhan[1], "start;");

// set devices to rawmode

BLC_sendCommand(chanhan[0], "dtlmode 1;");
BLC_sendCommand(chanhan[1], "dtlmode 1;");

// send RAW Slave Response

BLC_sendRawSlaveResponse(chanhan[0], masterreq, MRQMask, respondedata, 8);

```

```

// send RAW Master Request

masterrequest = BLC_sendRawMasterRequest(chanhan[1],masterreq,1);

// GET raw Slave Response

do {

std::this_thread::sleep_for(std::chrono::milliseconds(20));

BLC_getRawSlaveResponse(chanhan[1],response,8);

printf("get RAW slave response: %d \n",sc);

}while(sc!=0);

printf("RESPONSE: ");

for(int i = 0; i < 8; i++) {

printf("%02x ",response[i]);

}

printf("\n ");

```

第二个主请求使用相同过程:

1 BLC\_sendRawSlaveResponse

专用的主请求的响应。

2 BLC\_sendRawMasterRequest

发送主请求。（检查变量“masterrequest”的返回值是否为零）。

3 while (BLC\_getRawSlaveResponse (chanhan [1], response, 8) != 0)

在此，重要的是重复执行此功能，直到从节点接收主请求帧并且成功发送响应。

4Print response

在每个从响应之后，用 BLC\_GetRawSlaveResponse()读取缓冲区。

同样，保持顺序很重要。否则，可能会出现计时问题。

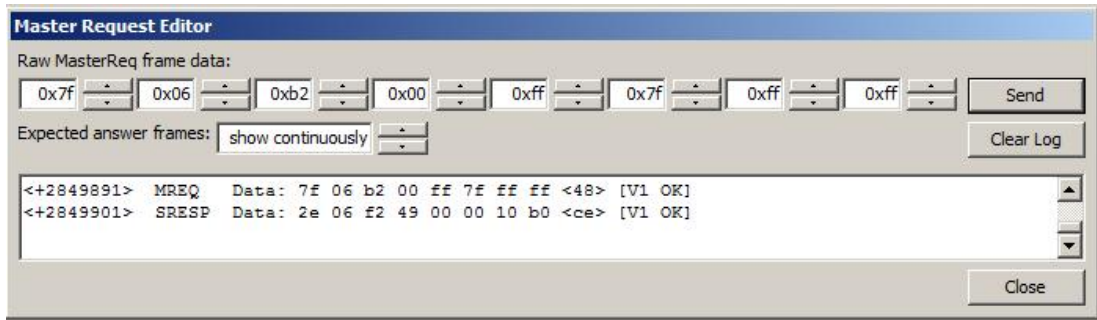
输出页面如下所示：

```
get RAW slave response: -100009 // The Return value will be executed until
get RAW slave response: -100009 // it has a valid slave response
get RAW slave response: -100009
get RAW slave response: -100009
get RAW slave response: -100009
get RAW slave response: -100009
get RAW slave response: -100009
get RAW slave response: -100009
get RAW slave response: -100009
get RAW slave response: 0 // responsedata[:
RESPONSE: 08 07 06 05 04 03 02 01 // Slaveresponse data of first master request
get RAW slave response: -100009
get RAW slave response: -100009
get RAW slave response: -100009
get RAW slave response: -100009
get RAW slave response: -100009
get RAW slave response: -100009
get RAW slave response: -100009
get RAW slave response: -100009
get RAW slave response: 0 // responsedata1[:
RESPONSE: 0a 0b 0c 0d 0e 0f 10 11 // Slave response of second master request.
```

### 13.3.1.3.5 通过 SimpleMenu 发送主请求

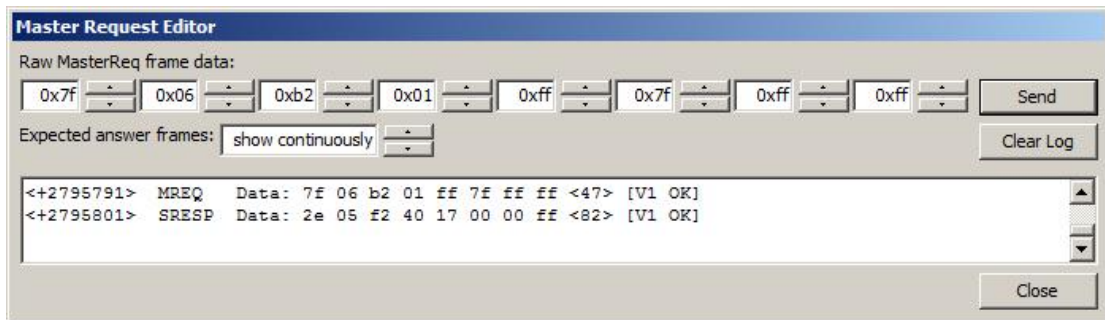
还可以通过在 Linworks 的 SimpleMenu 软件中，利用 Simulation / Send Masterrequest 选项

发送 raw 主请求。



如果 NAD, SupplierId 或 FunclId 未知, 请使用通用 ID。

NAD	0x7F
SupplierId	0x7FFF
FunclId	0x7FFF



### 13.3.1.4 诊断 cooked 模式 (DTL - 诊断传输层)

HARP-5 根据 LIN 规范中规定的诊断传输层进行帧的传输和接收。(固件版本 V.3.20 或更高版本)。

该规范定义了如何将要传输的数据打包到主请求或从响应帧中, 并定义了 3 种帧类型:

SF	单帧	用于传输最多 6 字节的有效负载 (SID + 数据字节 D1..D5)。
FF	首帧	用于多帧传输的首帧, 可传输 7 ... 4095 字节的有效载荷, 首帧携带 5 字节的有效载荷 (SID + 数据字节 D1 ... D4)。
CF	后续帧	在发送首帧之后, 多帧传输将发送后续帧, 每个后续帧承载 6 个数据字节。

主请求和从响应是传输帧。最多 4095 字节的数据对象可以在两个方向上传输。

NAD 是从节点的地址。

PCI 字节表示帧类型 (SF, FF, CF) 并编码一些附加信息。

有三种不同的 PCI 类型，它们通过 PCI 的 4 个较高位来区分。

在第一帧 (FF) 中，长度由 PCI 中的 4 位和随后的 LEN 字节中的 8 位组成，其中最多 12 位 = 4095 最大可用长度。

有关更多详细信息，请查看 LIN 规范 (诊断和配置 PDU 结构)。

SID 是服务标识符，并指定应由寻址的从节点执行的请求，该请求包含在单帧 (SF) 的 LEN 中和 FF 中的额外字节中。由于属于第一帧，后续帧不需要标识符。

Type	Coding of PCI type				Additional information			
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SF	0	0	0	0	Length (0...6) SID is included			
FF	0	0	0	1	Length/256			
CF	0	0	1	0	Frame counter (0...15)			

Diag Frametypes

SF - Single Frame	NAD	PCI = SF	D0	D1	D2	D3	D4	D5
FF - First Frame	NAD	PCI = FF	LEN	D0	D1	D2	D3	D4
CF - ConsecutiveFrame	NAD	PCI = CF	D0	D1	D2	D2	D4	D5

PCI composition

	B7...B4	B3...B0	
PCI-SF	0	Length	Length 0...6, which is maximum payload length in SingleFrame Message
PCI-FF	1	Length/256	Length&0xff In a FirstFrame (FF) PCI is always followed by additional LEN Byte. Maximum Payload length = 4095 (12 Bit)
PCI-CF	2	Framecounter	Framecounter in first CF = 1, in second CF = 2 etc. If more than 15 frames, frame counter wraps around and continues with 0, 1, 2,...

**Example: SF-Request with SF-Response**

Request	0x3C	0x0A	0x03	0x22	0x06	0x2E	0xFF	0xFF	0xFF
Response	0x3D	0x0A	0x06	0x62	0x06	0x2E	0x80	0x00	0x00

**Example: SF-Request with MultiFrame-Response (FF + 2\*CF)**

Request	0x3C	Nad	0x03	0x22	0x06	0x5E	0xFF	0xFF	0xFF
Response	0x3D	0x0A	0x10	0x0E	0x62	0x06	0x5E	"3"	"C"
Response	0x3D	0x0A	0x21	"8"	"9"	"5"	"9"	"5"	"3"
Response	0x3D	0x0A	0x22	"7"	" "	" "	0xFF	0xFF	0xFF

RSID 是响应服务标识符，它指定响应的内容。这允许使用 API 给出的一个命令发送长度最多为 4096 个数据字节的报文。将报文分段为 SF，FF 和 CF 帧是由 HARP-5 DLL 固件以及数据缓冲完成的。

**Version incompatibility: HARP-5 V.3.40之前的固件版本仅限使用最大1024字节的报文。**

**从V.3.40版本开始，固件支持处理最大4095字节的报文，这是LIN规范定义的最大值。**

**Version incompatibility: 要允许DTL通信工作，必须一个加载SDF，其中包括一个包含主请求帧和从属响应帧的调度表，并且选择运行这个调度表。**

Baby-LIN-DLL 提供 API 函数,用于在仿真从节点时传递应由主请求帧或从响应帧发送的报文内容。

Baby-LIN API 需要 SID 和 RSID 的值作为数据缓冲区中的第一个字节发送，用户可在调用 BL\_sendDTLRequest 和 BL\_sendDTLResponse 时给出。

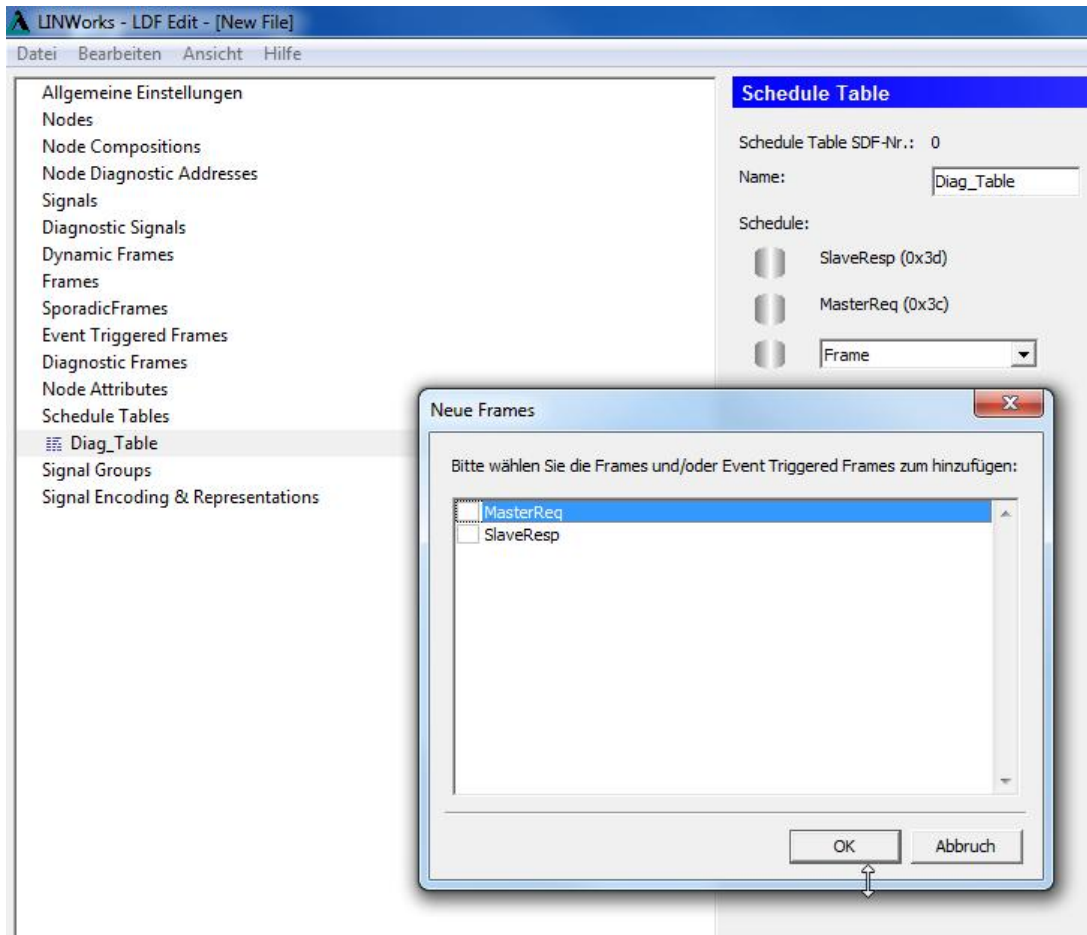
要使用 DTL 功能，您实际上需要实现一个调用 API 函数的程序，因为标准 LINWorks 应用程序不支持 DTL 机制。

#### **13.3.1.4.1 通过 SimpleMenu 的 cooked 请求**

以下示例显示如何在 LINWorks SimpleMenu 中发送 Cooked 请求：

这里，使用 HARP-5 和能够响应 DTL 请求的设备（作为从节点）。SDF 文件必须包含一个含有诊断帧的调度表。在 LINWorks LDFEdit 中，选择“Diagnostic frames”，然后单击“activate”。

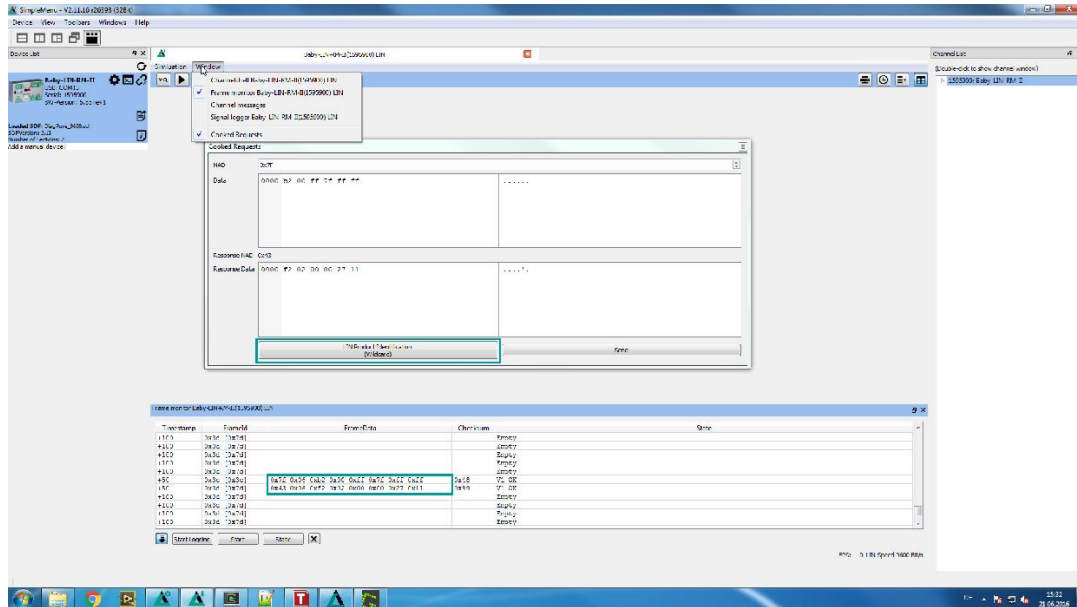
现在，可以将请求和响应的帧添加到调度表中。



使用 LDF 文件在 SessionConf 中创建 SDF 文件后，可以将其用于 SimpleMenu 中发送 Cooked 请求。

运行调度程序后，可以通过 Windows ->Cooked Request -> LIN Product Identification (wildcard)发送 cooked 请求或在专用请求中输入。如果您不知道从节点设备的 NAD，则需要使用通配符选项以获得连接到总线的所有从节点的响应。正如您将在帧监视器上看到的那样，一个空响应帧 (ID: 0x3D) 被发送，直到一个 cooked 请求被执行并发送请求帧 (ID: 0x3C)。然后，响应帧 (0x3D) 将由从站填充数据。





### 13.3.1.4.2 API

这些是使用诊断 cooked 模式的 API 调用函数：

### 13.3.1.4.3 BLC\_sendDTLRequest

要发送最多 4095 个数据字节的主请求报文并使用 DTL，请调用 API 函数：

```
int BLC_sendDTLRequest (BL_HANDLE handle, unsigned char nad, int length, char * data);
```

将给定的 DTL 主请求发送到 NAD 标识的节点。

handle 参数	用于区别通道的句柄；由 BLC_getChannelhandle()函数返回。
nad 参数	请求发送的节点的 NAD。
length 参数	下面的 data 数组的长度。
data 参数	DTL 帧数据（以 SID 开始，后面跟随最多 4095 字节的数据）。
返回值	函数运行状态，返回 0 表示无错误。

#### 13.3.1.4.4 BLC\_sendDTLResponse

int BLC\_sendDTLResponse (BL\_HANDLE handle, unsigned char nad, int length, char\* data);

将给定的 DTL 从响应发送到 NAD 标识的节点。

handle 参数	用于区别通道的句柄；由 BLC_getChannelhandle()函数返回。
nad 参数	请求发送的节点的 NAD。
length 参数	下面的 data 数组的长度。
data 参数	DTL 帧数据（以 RSID 开始，后面跟随最多 4095 字节的数据）。
返回值	函数运行状态，返回 0 表示无错误。

#### 13.3.1.4.5 BLC\_getDTLRequestStatus

int BLC\_getDTLRequestStatus (BL\_HANDLE handle);

返回上次请求操作发送的状态。

handle 参数	用于区别通道的句柄；由 BLC_getChannelhandle()函数返回。
返回值	大于等于 0 时，表示最后一次请求操作的状态，有关值的含义，请参阅 BL_DTL_STATUS。  小于 0 时，请参见错误的标准返回值或文本说明（返回值<-1000）  BLC_getLastError()。

#### 13.3.1.4.6 BLC\_getDTLResponseStatus

int BLC\_getDTLResponseStatus (BL\_HANDLE handle);

返回上次响应操作发送的状态。

handle 参数	用于区别通道的句柄；由 BLC_getChannelhandle()函数返回。
返回值	<p>大于等于 0 时，表示最后一次请求操作的状态，有关值的含义，请参阅 BL_DTL_STATUS。</p> <p>小于 0 时，请参见错误的标准返回值或文本说明（返回值&lt;-1000）BLC_getLastError()。</p>

### 13.3.1.4.7 BL\_DTL\_STATUS

0	LD_COMPLETED
1	LD_FAILED
2	LD_IN_PROGRESS

### 13.3.1.4.8 BLC\_getNextDTLResponse

如果模拟主节点，可以调用 API 函数检索从节点 DTL 响应：

```
int BLC_getNextDTLResponse (BL_HANDLE handle, BL_dtl_t *frame);
```

从接收器队列中抓取下一个完整的 DTL 响应。

handle 参数	用于区别通道的句柄；由 BLC_getChannelhandle()函数返回。
frame 参数	指向 BLC_DTL 结构的指针，用于存储响应信息。
返回值	函数运行状态，返回 0 表示无错误。

响应数据被写入结构 BLC\_DTL，指向 BLC\_DTL \* frame。

当接收器队列中没有完整的 DTL 从响应时，此 API 函数将返回值：-100009。

### 13.3.1.4.9 BLC\_getNextDTLRequest

如果仿真的是从节点，则可以调用 API 函数检索收到的 DTL 主请求：

```
int BLC_getNextDTLRequest (BL_HANDLE handle, BL_dtl_t *frame);
```

从接收器队列中获取下一个完整的 DTL 请求。

handle 参数	用于区别通道的句柄；由 BLC_getChannelhandle()函数返回。
frame 参数	指向 BLC_DTL 结构的指针，用于存储响应信息。
返回值	函数运行状态，返回 0 表示无错误。

请求数据被写入结构 BLC\_DTL，指向 BLC\_DTL \* frame。

当接收器队列中没有完整的 DTL 主请求时，此 API 函数将返回值：-100009。

### 13.3.1.4.10 BLC\_DTL

BLC\_DTL 结构的内容：

```
typedef struct _BLC_DTL{  
    // ! Status of protocol frame  
    BL_DTL_STATUS status;  
    // ! NAD of protocol frame  
    unsigned char nad;  
    // ! Length of the data-array.  
    int length;  
    // ! frame data, beginning with the (R)SID.  
    unsigned char data[4*1024];  
} BLC_DTL;
```

### 13.3.1.4.11 示例：仿真主节点

在以下代码示例中，由 HARP-5 仿真的主节点用于将 cooked 请求发送到真实从节点设备。为了获得从节点响应，轮询相应的功能，直到从节点用数据填充响应帧。

```
BL_HANDLE chanhan, conhan;

BLC_DTL Request;

BL_DTL_STATUS status;

BLC_FRAME frame;

unsigned char data [6] = {0xb2,0x00,0xff,0x7f,0xff,0xff};

int i,j,sc =0;

// Initialization

conhan = BLC_open("/dev/ttyUSB0");

BLC_loadSDF(conhan,"DiagPure.sdf",1);

chanhan = BLC_getChannelHandle(conhan, 1);

sc = BLC_loadSDF(conhan,"DiagPure_9600.sdf",1);

BLC_sendCommand(chanhan,"disframe 255 1;");

BLC_sendCommand(chanhan,"start;");

BLC_setDTLMode(chanhan,2);

// Send the Cooked Request

BLC_sendDTLRequest(chanhan,0x7F,sizeof(data),&data);

// Wait until the Status for the request is done.

do{

sc= BLC_getDTLRequestStatus(chanhan);

printf("RequestStatus? %d\n",sc);

} while((status!=0) || (sc!=0));

sleep(1); // A waiting period is required to be able to receive the respond

// Read out the response

BLC_getNextDTLResponse(chanhan,&Request);
```

```
printf("Size DTL_Respond: %d \n",sizeof(Request));  
printf("Infos about Respond:\n");  
printf("Status: %d \n",Request.status);  
printf("NAD: %d \n",Request.nad);  
printf("Length: %d \n",Request.length);  
for (int i =0; i<Request.length;i++) {  
printf("Data[%d]: c= %02x \n",i,Request.data[i]);  
}  
  
BLC_sendCommand(chanhan,"stop");
```

输出:

RequestStatus? 2

RequestStatus? 2

RequestStatus? 2

RequestStatus? 0

Size DTL\_Respond: 4108

Infos about Respond:

Status: 0

NAD: 67

Length: 6

Data[0]: c= f2

Data[1]: c= 02

Data[2]: c= 00

Data[3]: c= 00

Data[4]: c= 27

Data[5]: c= 11

### 13.3.1.4.12 Callbacks

另一种接收响应的方式（代替 BLC\_getNextDTLResponse）是使用以下回调函数：

#### 13.3.1.4.13 BLC\_registerDTLRequestCallback

```
int BLC_registerDTLRequestCallback (BL_HANDLE handle, BL_dtl_request_callback_func *cb);
```

注册回调函数，每次接收 DTL 请求时都会调用该函数，但前提是至少有一个仿真的从节点。

发出 NULL 指针会取消注册回调函数。当另一个进程调用该函数时，请注意线程安全性（即使用互斥体等）。

handle 参数	用于区别通道的句柄；由 BLC_getChannelhandle()函数返回。
cb 参数	指向与 BLC_dtl_request_callback_func 兼容的函数指针。
返回值	函数运行状态，返回 0 表示无错误。

#### 13.3.1.4.14 BLC\_registerDTLResponseCallback

```
int BLC_registerDTLResponseCallback (BL_HANDLE handle, BL_dtl_response_callback_func cb);
```

注册回调函数，每次接收 DTL 响应时都会调用该函数，仅在 HARP-5 仿真主节点时可用。

发出 NULL 指针会取消注册回调函数。当另一个进程调用该函数时，请注意线程安全性（即使用互斥体等）。

handle 参数	用于区别通道的句柄；由 BLC_getChannelhandle()函数返回。
cb 参数	指向与 BLC_dtl_request_callback_func 兼容的函数指针。
返回值	函数运行状态，返回 0 表示无错误。

回调函数都声明为 void 函数，其中调用的一个参数类型为 BL\_dtl\_t。因此，如果需要使用回调接口，则必须实现并注册适当的回调函数。

```
typedef void (BLC_dtl_request_callback_func)(BL_HANDLE, BLC_DTL frame);  
typedef void (BLC_dtl_response_callback_func)(BL_HANDLE, BLC_DTL frame);
```

每次完整的 DTL 响应或请求可用时，将调用相应的回调函数，通过 BLC\_DTL 类型的调用参数导入 DTL 报文数据。

**Version incompatibility:** 由于都需要使用缓冲区的存储，因此不可能同时在同一HARP-5上模拟DTL主节点和DTL从节点。

**Advice:** 如果在 DTL 会话中使用 HARP-5 来模拟从节点，则必须通过发出命令 dtlmode 2 来激活 cooked 模式。

#### 13.3.1.4.15 Callback example

除了使用带有相应回调函数的 BLC\_registerDTLResponseCallback 之外，此处使用与 BLC\_getNextDTLResponse 相同的代码。

```
#include <stdio.h>  
  
#include <BabyLIN.h>  
  
#include <BabyLINCAN.h>  
  
static void dtl_response_callback(BL_HANDLE bh, BLC_DTL frame) {  
    int i;  
    fprintf(stderr, "nad = %d\n", frame.nad);  
    for (i=0; i<frame.length; i++) {
```



```

fprintf(stderr, "%s Data[%d]: c= %02x \n", __func__, i,frame.data[i]);
}

fprintf(stderr, "\n");
}

int main(void) {
BL_HANDLE chanhan, conhan;

BLC_DTL Request;

BL_DTL_STATUS status;

BLC_FRAME frame;

unsigned char data [6]= {0xb2, 0x00, 0xff, 0x7f, 0xff, 0xff};

int i, j, sc =0;

conhan = BLC_open("/dev/ttyUSB0");

sc = BLC_loadSDF(conhan,"DiagPure_9600.sdf",1);

chanhan = BLC_getChannelHandle(conhan, 1);

sc = BLC_setDTLMode(chanhan, 2);

BLC_sendCommand(chanhan, "start;");

BLC_sendCommand(chanhan, "disframe 255 1;");

BLC_sendDTLRequest(chanhan,0x7F,sizeof(data),&data);

BLC_registerDTLResponseCallback(chanhan, dtl_response_callback);

do{

sc = BLC_getDTLRequestStatus(chanhan);

printf("RequestStatus? %d\n",sc);

} while((status!=0) || (sc!=0));

sleep(1);

```

```
BLC_sendCommand(chanhan, "stop");  
BLC_registerDTLResponseCallback(chanhan, NULL);  
}
```

输出:

RequestStatus? 2

RequestStatus? 2

RequestStatus? 0

nad = 67

dtl\_response\_callback Data[0]: c= f2

dtl\_response\_callback Data[1]: c= 02

dtl\_response\_callback Data[2]: c= 00

dtl\_response\_callback Data[3]: c= 00

dtl\_response\_callback Data[4]: c= 27

dtl\_response\_callback Data[5]: c= 11

### 13.3.2 睡眠指令（特殊的主请求帧）

sleep 命令允许 LIN 总线进行休眠。

sleep 命令发送特定的主请求帧，它的数据字节都设置为零。发送此帧后，HARP-5 将停止总线仿真。

用于执行此操作的 API 函数是：

```
BLC_sendCommand (handle, "sleep;");
```

如果 HARP-5 仿真的不包括主节点，则函数将返回一个负值。

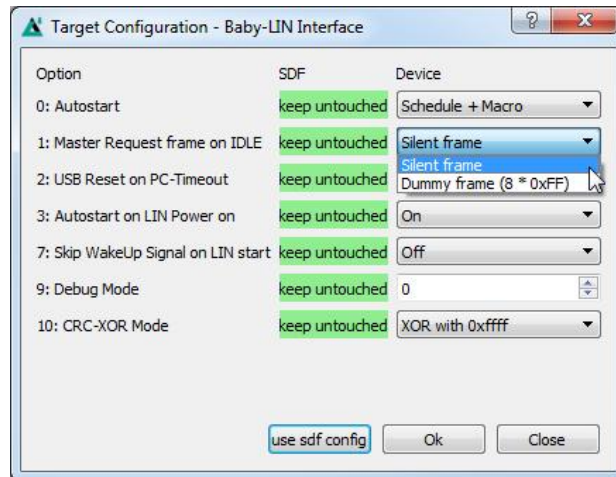
### 13.3.3 目标配置

只要 API 函数不在提供主请求帧的数据，HARP-5 就允许配置发送的主请求帧的种类。

目标配置在 LINWorks 组件 SimpleMenu 中可用。



如果您的 HARP-5 连接到 PC，会弹出一个菜单，显示可用的参数和实际的设置。



IDLE 上的主请求帧有两个选择。

静音帧	不会发送任何帧，但在执行时隙，这将导致静音帧。
虚拟帧	只要应用程序没有提供主请求帧数据，就会在调度表中的每个主请求帧出现时发送一个所有数据字节设置为 0xFF 的帧。

目标配置存储在 HARP-5 电路板的 EEPROM 存储器中。在这过程中，只有将 HARP-5 连接到

PC，改动才会有效。

## 13.4 ASCII模式

### 13.4.1 简介

ASCII 模式是 HARP-5 支持的一个特殊模式。它允许使用 HARP-5 的第一个 LIN 总线作为电平转换器或通过 LIN 硬件层发送字符而无需遵守 LIN 帧封装约定。

在 ASCII 模式下，HARP-5 将不再使用专有的二进制格式与 PC 通信，而是使用 ASCII 命令。因此，不需要使用“Baby-LIN-DLL”。

在 ASCII 模式下，您只能使用第一个 LIN 总线通道，而无法操作其它所有通道。

可以使用命令行工具或通过 USB 连接的虚拟 COM 端口发送一些特殊字节，将 HARP-5 切换到 ASCII 模式。

ASCII 模式保持激活状态，直到 HARP-5 重新启动或发送一个特殊命令为止。

### 13.4.2 切换ASCII模式

有两种方法可以将 HARP-5 切换到 ASCII 模式：

- 使用 LINWorks 提供的 Switch to ASCII 工具。

该工具可以用来测试 ASCII 模式或手动切换到 ASCII 模式。

有关更多信息，请查看“Switch to ASCII tool”一章。

- 将一个特殊的字节序列发送到 USB 连接的虚拟 COM 端口。

如果要在 API 程序中使用 ASCII 模式，需要使用这个方法。

操作步骤如下。

请按照以下步骤在应用程序中将 HARP-5 切换到 ASCII 模式：

- 使用以下设置打开 HARP-5 的 USB 连接的虚拟 COM 端口：

COM 端口：HARP-5 的虚拟 COM 端口

波特率：921,600 Bd

**Advice:** 在老的Windows操作系统上，可能无法设置此波特率。因此，也可以设置成300Bd的

**波特率。“Baby-LIN 驱动”将自动将其更改为921600 Bd。**

数据位：8

停止位：1

奇偶校验：无

流量控制：无

- 写入以下字节：

十六进制：0x80,0x95,0x61,0x73,0x63,0x69,0x69,0x8f

0x80,0x95,0x61,0x73,0x63,0x69,0x69,0x8f

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
Hex:	0x80	0x95	0x61	0x73	0x63	0x69	0x69	0x8f

- 如果切换到 ASCII 模式成功，HARP-5 将使用以下字节序列应答：

	Byte 1	Byte 2	Byte 3
Hex:	0x21	0x0D	0x0A
ASCII:	!	<CR>	<LF>

### 13.4.3 关闭 ASCII 模式

HARP-5 保持 ASCII 模式，直到发生以下事件之一：

- HARP-5 已重启。

- 发送 RESET 命令。

### 13.4.4 ASCII 命令

#### 13.4.4.1 命令一览

ASCII 命令由命令字符串、可选和可变参数、以及表示结束的<CR>组成。

所有参数都是整型，可以输入十进制和十六进制数字：

数字格式	描述	示例
十进制	输入十进制数字不需要前缀或后缀。	255
十六进制	输入十六进制数字需要后接 H。	FFH

以下是可用命令：

命令	描述
ECHO	打开 debug 输出。
VERSION	获取固件版本。
TARGETSPEED	设置 LIN 总线的波特率。
LINFRAME	在 LIN 总线上发送一帧。
TRANSPARENT	直接向 LIN 总线发送和接收字符。
RESET	关闭 ASCII 模式。

**Attention:** 以下命令存在，但保留供内部使用。请勿将其中任何一个命令发送到HARP-5。这些命令可以将HARP-5切换到一个状态，您无法自行从中恢复它。

- ERASE
- FIRMWARE

- **PRODTEST**

### 13.4.4.2 ECHO

此命令用于启用或禁用 ECHO 模式。如果启用了 ECHO 模式，则 HARP-5 会返回额外的调试文本和换行符。

来自 PC 的字符不会重复，这是由于 LIN 总线的半双工性质。

格式:	ECHO [flag]<CR>	
变量[flag]:	0	ECHO 模式关闭
	1	ECHO 模式开启
返回值:	!<CR>	成功
	?<CR>	失败
示例:	<< ECHO 1<CR> >> !<CR>	

### 13.4.4.3 VERSION

VERSION 命令用于查询 HARP-5 的固件版本。

格式:	VERSION<CR>	
返回值:	HARP-5 V.5.70 Build:1<CR>	HARP-5 的固件版本
示例:	<< VERSION<CR> >> HARP-5 V.5.70 Build:1<CR>	

### 13.4.4.4 TARGETSPEED

TARGETSPEED 命令用于设置 LIN 总线接口的波特率。

**Warning:** 现在已经不再支持SETSPEED命令，我们已经将其替换为TARGET SPEED命令。

格式:	TARGETSPEED [baudrate]<CR>	
变量[flag]:	1 Baud - 125 kBaud	LIN 总线接口的波特率
返回值:	!<CR>	成功
	?<CR>	失败
示例:	<< TARGETSPEED 19200<CR> >> !<CR>	

### 13.4.4.5 LINFRAME

LINFRAME 命令通过 LIN 总线接口发送 LIN 帧。

校验和类型可以自主选择，并且 HARP-5 将自动创建校验和。

输入帧 ID 时不需要输入 PID，按照 LDF 文件中的描述输入即可。HARP-5 会在向 LIN 总线上发送帧时，自动添加两个保护位，把它变成 PID。

帧的长度将根据输入的数据字节数得到。

格式:	LINFRAME [cstype] [id] [data1] [data2] ... [dataN]<CR>	
变量[cstype]:	0	按照 LIN v1.x 协议的传统校验和
	1	按照 LIN v2.x 协议的增强校验和
变量[id]:	0-63	帧 ID(非 PID)
变量[data1]到[dataN]:	0-255	数据字节的值
返回值:	!<CR>	成功



	?<CR>	失败
示例:	<< LINFRAME 1 10 1 2 3 4 5 6 7 8<CR> >> !<CR>	

### 13.4.4.6 TRANSPARENT

TRANSPARENT 命令将 HARP-5 切换为透明的数据交换模式。发送到 HARP-5 的所有 ASCII 字符将直接传递到 LIN 总线接口，反之亦然。

格式:	TRANSPARENT [echo] [timeout]<CR>	
变量[echo]:	0	ECHO 模式已禁用。HARP-5 发送的字符不会被回读。
	1	ECHO 模式已启用。将回读 HARP-5 发送的字符。
变量[timeout]:	1 - FFFFFFFEH	以毫秒为单位的超时时间。如果在此期间内没有收到任何字符，HARP-5 将自动退出该模式，并准备接收新的 ASCII 命令。
	0 或 FFFFFFFFH	没有设置超时。HARP-5 将保持该模式，直到电源重启。
返回值:	!<CR>	成功
	?<CR>	失败
示例:	<< TRANSPARENT 1 10000<CR> >> !<CR>	

### 13.4.4.7 RESET

RESET 命令会复位 HARP-5，并终止 ASCII 模式。

格式:	RESET<CR>	
返回值:	!<CR>	成功
	?<CR>	失败
示例:	<< RESET<CR> >> !<CR>	

### 13.4.5 应用实例

通过 LIN 总线接口发送和接收字符的自定义应用程序的步骤如下:

步骤	格式	数据	描述
1	二进制	80H 95H 61H 73H 63H 69H 69H 8FH	进入 ASCII 模式
2	ASCII	TARGETSPEED 115200	设置 LIN 总线接口的波特率为 115200Baud
3	ASCII	TRANSPARENT 0 FFFFFFFFH	进入透明模式，不启用 ECHO 模式，不设置超时
4...	ASCII 字符		从 LIN 总线发送/接收 ASCII 字 符

在 HARP-5 通电后，初始化程序部分只需执行一次。

# 虹科云课堂

HongKe Online Academy

2020年2月21日,虹科云课堂首次与大家见面,带来的第一节《CAN总线基础之物理层篇》课程,就得到了各位工程师朋友们的热情支持与参与,当晚观看人数4900+。我们非常感恩,愿不负支持与鼓励,致力将虹科云课堂打造成干货知识共享平台。

目前虹科云课堂的全部课程已经超过200节,如下表格是我们汽车相关的部分课程列表,大家通过微信扫描二维码关注公众号,点击免费课程直接进入观看,全部免费。

## 汽车以太网课程

智能网联下车载以太网的解决方案  
SOME/IP协议介绍  
基于CanEasy浅谈XCP  
TSN/AVB 基于信用点的整形

## TSN技术课程

基于TSN的汽车实时数据传输网络解决方案  
TSN时间敏感型网络技术综述  
以太网流量模型和仿真  
基于TSN的智能驾驶汽车E/E架构设计案例分享  
IEEE 802.1AS 时间同步机制  
TSN技术如何提高下一代汽车以太网的服务质量?

## CAN、CAN FD、CAN XL总线课程

CAN总线基础之物理层篇  
CAN数据链路层详解篇  
CAN FD协议基础  
CAN总线一致性测试基本方法  
CAN测试软件(PCAN-Explorer6)基本使用方法  
CAN测试软件(PCAN-Explorer6)高级功能使用  
浅谈CAN总线的最新发展: CAN FD与CAN XL  
CAN线的各种故障模式波形分析

## LIN总线相关课程

汽车LIN总线基本协议概述  
汽车LIN总线诊断及节点配置规范  
LIN总线一致性测试基本方法  
LIN自动化测试软件(LINWorks)基本使用方法  
LIN自动化测试软件(LINWorks)高级功能使用  
基于CAN/LIN总线的汽车零部件测试方案

## CAN高级应用课程

UDS诊断基础  
UDS诊断及ISO27145  
基于UDS的ECU刷写  
基于PCAN的二次开发方法  
CCP标定技术  
J1939及国六排放  
OBD诊断及应用(GB3847)  
BMS电池组仿真测试方案  
总线开发的流程及注意事项  
车用总线深入解析

## 汽车测修诊断相关课程

汽车维修诊断大师系列-如何选择示波器  
汽车维修诊断大师系列-巧用示波器  
汽车维修诊断-振动异响(NVH)诊断方案



关注获取最新课程



汽车电子bilibili主页

## 汽车电子领域专业的 资源整合及技术服务落地供应商

### 关于虹科

虹科电子科技有限公司（前身是宏科）成立于1995年，总部位于中国南方经济和文化中心-广州；还在上海、北京、成都、西安、苏州、台湾、香港，韩国、日本和美国硅谷设有分公司。

我们是一家高新技术公司，是广东省特批的两高四新、三个一批、专精特新和瞪羚企业，并与全球顶尖公司有多领域的深度技术合作，业务包括工业自动化和数字化、汽车研发测试、自动驾驶等领域；医药和风电行业等的环境监测；半导体、轨道交通、航空航天等测试测量方案。

虹科工程师团队致力于为行业客户提供创新产品和解决方案，全力帮助客户成功。

### 汽车电子事业部

虹科是一家在汽车电子，特别是汽车总线行业经验超过10年的高科技公司虹科与全球顶尖公司技术合作，提供基于CAN/CAN FD/LIN/TSN/车载以太网，从数据采集、监控和分析，网络性能仿真和优化，网络配置和一致性测试，架构开发和服务的一系列方案。虹科自主研发的EOL测试系统、CCP/XCP标定和UDS诊断服务开发包等已经在业内完成超过1000次安装和测试。虹科的总线培训课程已经累计超过20万的播放量。

虹科工程师均受过国内外专业培训，并获得专业资格认证，这些宝贵经验得以让虹科团队完成了国内多个具体项目的落地，并得到了包括蔚来、文远、比亚迪、博世、大陆等多个用户的一致好评。





华东区销售  
梁倩妮

电话/微信: 183 0218 1471  
邮箱: liang.qianni@hkaco.com



华南区销售  
董欢

电话/微信: 189 2224 3009  
邮箱: dong.huan@hkaco.com



华北区销售  
张瑞婕

电话/微信: 181 3875 8797  
邮箱: zhang.ruijie@hkaco.com



**HongKe**  
虹科

虹科电子科技有限公司

www.haomotive.com  
auto@hkaco.com

广州市黄埔区开泰大道30号佳都PCI科技园6号楼

T (+86)400-999-3848  
M (+86)135 1276 7172

各分部: 广州 | 成都 | 上海 | 苏州 | 西安 |  
北京 | 台湾 | 香港 | 日本 | 韩国 | 美国硅谷



获取更多资料