AN2405

应用笔记

UM32x42x LIN 配置指南

版本：V1.0

Unicmicro

UNICMICRO
广芯微电子

广芯微电子（广州）股份有限公司

http://www.unicmicro.com/

条款协议

本文档的所有部分，其著作产权归广芯微电子（广州）股份有限公司（以下简称广芯微电子）所有，未经广芯微电子授权许可，任何个人及组织不得复制、转载、仿制本文档的全部或部分组件。本文档没有任何形式的担保、立场表达或其他暗示，若有任何因本文档或其中提及的产品所有资讯所引起的直接或间接损失，广芯微电子及所属员工恕不为其担保任何责任。除此以外，本文档所提到的产品规格及资讯仅供参考，内容亦会随时更新，恕不另行通知。

1. 本文档中所记载的关于电路、软件和其他相关信息仅用于说明半导体产品的操作和应用实例。用户如在设备设计中应用本文档中的电路、软件和相关信息，请自行负责。对于用户或第三方因使用上述电路、软件或信息而遭受的任何损失，广芯微电子不承担任何责任。
2. 在准备本文档所记载的信息的过程中，广芯微电子已尽量做到合理注意，但是，广芯微电子并不保证这些信息都是准确无误的。用户因本文档中所记载的信息的错误或遗漏而遭受的任何损失，广芯微电子不承担任何责任。
3. 对于因使用本文档中的广芯微电子产品或技术信息而造成的侵权行为或因此而侵犯第三方的专利、版权或其他知识产权的行为，广芯微电子不承担任何责任。本文档所记载的内容不应视为对广芯微电子或其他人所有的专利、版权或其他知识产权作出任何明示、默示或其它方式的许可及授权。
4. 使用本文档中记载的广芯微电子产品时，应在广芯微电子指定的范围内，特别是在最大额定值、电源工作电压范围、热辐射特性、安装条件以及其他产品特性的范围内使用。对于在上述指定范围之外使用广芯微电子产品而产生的故障或损失，广芯微电子不承担任何责任。
5. 虽然广芯微电子一直致力于提高广芯微电子产品的质量和可靠性，但是，半导体产品有其自身的具体特性，如一定的故障发生率以及在某些使用条件下会发生故障等。此外，广芯微电子产品均未进行防辐射设计。所以请采取安全保护措施，以避免当广芯微电子产品在发生故障而造成火灾时导致人身事故、伤害或损害的事故。例如进行软硬件安全设计（包括但不限于冗余设计、防火控制以及故障预防等）、适当的老化处理或其他适当的措施等。

目录

Unicmicro

 Copyright © 2025 广芯微电子（广州）股份有限公司

1 摘要

本篇应用笔记主要介绍 UM32x42x LIN 的功能以及配置方法。

本篇应用笔记主要包括:

- LIN 概述
- LIN 配置流程
- 使用参考例程

注: 具体功能及寄存器的操作等相关事项请以用户手册为准。

Unicmicro

Copyright © 2025 广芯微电子（广州）股份有限公司

2 LIN 概述

2.1 什么是 LIN

LIN，全称为 Local Interconnect Network，是基于 UART/SCI（通用异步收发器/串行通信接口）的低成本串行通信协议，可用于分层次车内网络在低端（速度和可靠性要求不高、低成本的场合）的需求。特性如下:

1. 网络由一个主机节点和多个从机节点构成，无 CAN 总线那样的仲裁机制，最多可连接 16 个节点（1 主 15 从）。
2. 对硬件要求简单，仅需 UART/SCI 接口，辅以简单驱动程序便可实现 LIN 协议。
3. 不需要单独的晶振，便能完成主、从节点的同步，硬件成本大幅降低。
4. 仅使用一根信号线便可完成信息的传输，即所谓的单总线设备。
5. 传输速率最高可达 20Kbps，符合 A 类网络标准，满足车身控制需要。
6. LIN 网络中新节点的加入，对网络中其他原有节点的软硬件设计不会造成影响。

2.2 LIN 帧结构

LIN 帧（Frame）包含报文头（Header）和报文响应（Response）两部分。主机任务负责发送帧头；从机任务接收帧头并对帧头所包含信息进行解析，然后决定是发送应答，还是接收应答，还是不作任何反应。

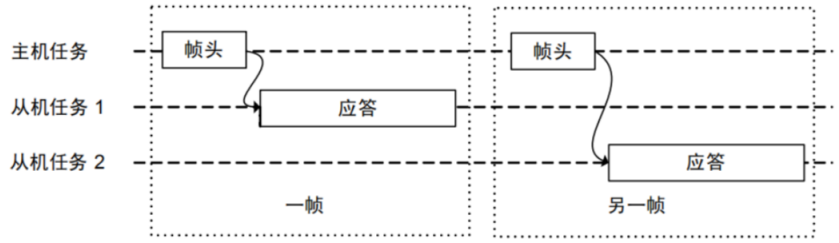


图 2-1：帧在总线上的传输

Copyright © 2025 广芯微电子（广州）股份有限公司

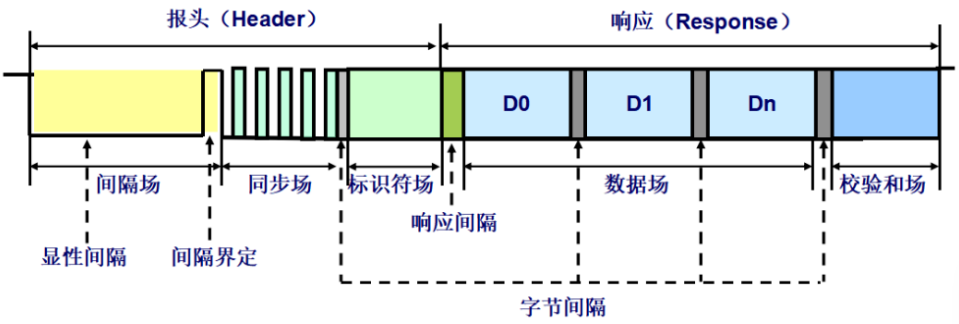


图 2-2：LIN 帧格式

2.2.1 报文头

报文头是由一个主机节点的主机任务发出的。报文头由同步间隔场（最小 13 个显性位）、同步场（1 个字节，数据不变，0x55）、和 PID 场（1 个字节）三部分组成。

2.2.2 报文响应

报文响应是由一个主机节点或从机节点的从机任务发出的。报文响应由 2/4/8 个字节的数据场、校验和场（1 个字节）所组成。

2.2.3 帧内空间分隔

报文头和响应之间有一个帧内空间分隔，最小空间为 0。

2.2.4 LIN 的字节场

LIN 的字节场格式就是通常的“SCI”或“UART”串行数据格式（N81 编码），即每个字节场的长度是 10 个位定时（BIT TIME）：1bit 起始位 + 8bits 数据位 + 1bit 停止位。起始位（START BIT）是一个“显性”位，它标志着字节场的开始。接着是 8 个数据位，首先发送最低位。停止位（STOP BIT）是一个“隐性”位，它标志着字节场的结束。LIN 报文帧中的同步场、标识符场、数据场、校验和场的格式都符合上述字节场的格式。

Copyright © 2025 广芯微电子（广州）股份有限公司

2.2.4.1 同步间隔场

间隔场是唯一一个不符合字节场格式的场。从节点需要检测到至少连续 11 个显性位才认为是间隔信号。

2.2.4.2 同步场

一个字节，即 0x55。

2.2.4.3 LIN 的被保护标识符场（PID 场）

PID（Protected Identifier）是 LIN 帧中定义的 8 位受保护标识符场，由 6 位标识符（ID）和 2 位奇偶校验位组成（如图 2-3）。其中 ID（Identifier）特指高 6 位标识符，用于定义报文内容与数据长度，取值范围为 0x00−0x3F ，其低 2 位为数据长度控制位。

需特别注意：ID ≠ PID，二者不可混淆，即：**ID** = 4 位报文 ID + 2 位数据长度控制位 = **6 位**；**PID** = ID + 2 位奇偶校验位 = **8 位**。

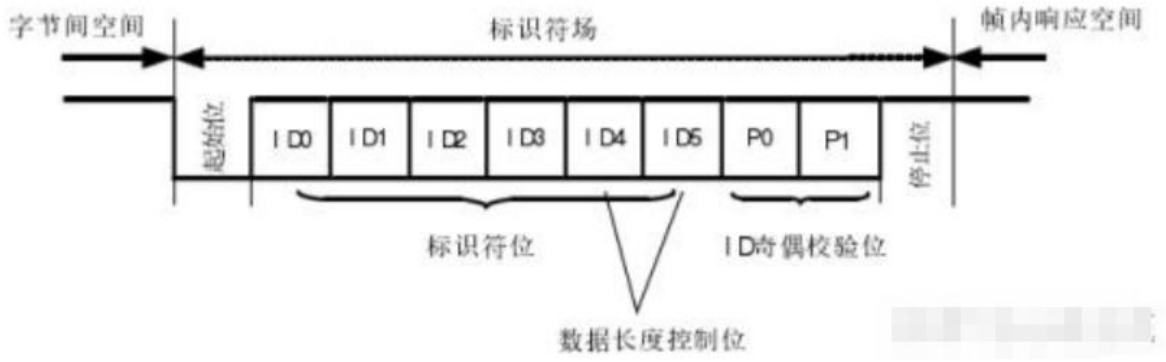


图 2-3：PID 场

图中的 ID4 和 ID5 为数据长度控制位（ID4 是低位），其编码与理论数据长度的对应关系如下：

表 2-1：编码与理论数据长度的对应关系

数据长度控制位（ID5:ID4）	理论数据长度
00/10	2 字节
01	4 字节
11	8 字节

Copyright © 2025 广芯微电子（广州）股份有限公司

需注意的是，在实际应用场景中，所传输的 LIN 报文数据长度统一为 8 字节，因此未体现出数据长度控制位（ID4、ID5）与实际数据长度的关联。

P0 和 P1 为奇偶校验位，奇偶校验就是在发送的每一个字节后都加上一位，使得每个字节中 1 的个数为奇数个或偶数个。接收方通过计算数据中 1 的个数是否满足奇偶性来确定数据是否有错。

2.2.4.4 数据场

数据场主要需注意每个字节先传输的是最低位。即如果某一信号长度超过 1 个字节，采用低位在前的方式发送**（小端）**。

2.2.4.5 校验和场（Checksum）

校验和场是数据场所有字节的和的反码。所有数据字节的和的补码，与校验和字节相加所得的和必须是 0xFF。

校验和场通常会有两种不同的类型，英文简称为 CST（Checksum Type）。一种是 Classic Checksum（LIN 1.3），一种是 Enhanced Checksum（LIN 2.0 及以上）。上文讲到的校验和场算法实际上是 Classic 的，即只对 Data（数据场）进行校验和的计算。Enhanced Checksum 在计算时需要把 PID 也加入到计算队列中。

Copyright © 2025 广芯微电子（广州）股份有限公司

3 LIN 配置流程

3.1 发送 LIN 数据帧

1. 开启 USART 时钟，释放复位，USART LIN 功能管脚复用。
2. 配置 USART_CR[3:2]，复位发送器和接收器。
3. 配置 USART_MR[3:0]，USART 工作在 LIN 主机模式。
4. 配置 USART_MR[19]，选择 16 倍过采样。
5. 配置 USART_CR[6]，使能 TX 发送器。
6. 配置 USART_CR[4]，使能 RX 接收器。
7. 配置 USART_BRGR, 设置波特率。
8. 配置 USART_LINMR[15:8]，定义传输数据的长度。
9. 配置 USART_LINIR[7:0]，定义传输的标识符字符。
10. 配置 USART_LINMR[1:0]，选择 PUBLISH 发送应答。
11. 发送数控写入 USART_THR 寄存器。
12. 等待状态寄存器 USART_CSR[15]置 1 后, 数据传输完成。

3.2 接收 LIN 数据帧

1. 开启 USART 时钟，释放复位，USART LIN 功能管脚复用。
2. 配置 USART_CR[3:2]，复位发送器和接收器。
3. 配置 USART_MR[3:0]，USART 工作在 LIN 从机模式。
4. 配置 USART_MR[19]，选择 16 倍过采样。
5. 配置 USART_CR[6]，使能 TX 发送器。
6. 配置 USART_CR[4]，使能 RX 接收器。
7. 配置 USART_BRGR, 设置波特率。

Unicmicro

Copyright © 2025 广芯微电子（广州）股份有限公司

8. 配置 USART_LINMR[15:8]，定义传输数据的长度。

9. 配置 USART_LINMR[1:0]，选择 SUBSCRIBE 接收应答。

10. 等待状态寄存器 USART_CSR[14]置 1 后, 确认接收了一个 LIN 标识符。

11. 等待状态寄存器 USART_CSR[0] 置 1 后，确认接收器已准备就绪。

12. 读取接收缓存寄存器 USART_RHR 数据，多次读取直到取出所有数据。

13. 等待状态寄存器 USART_CSR[15]置 1 后，数据接收完成。

Unicmicro

 Copyright © 2025 广芯微电子（广州）股份有限公司

4 使用参考例程

本例程 LIN 在 UM32x42x 平台上实现，具体可参考 UM32x42x_SDK Examples 中的 LIN 应用例程。

4.1 LIN 的发送配置

以下简单介绍如何使用 hal 库对 LIN 的发送进行配置。

1. 配置 USART LIN 基本内容。含有波特率、过采样及 LIN 工作在主模式。

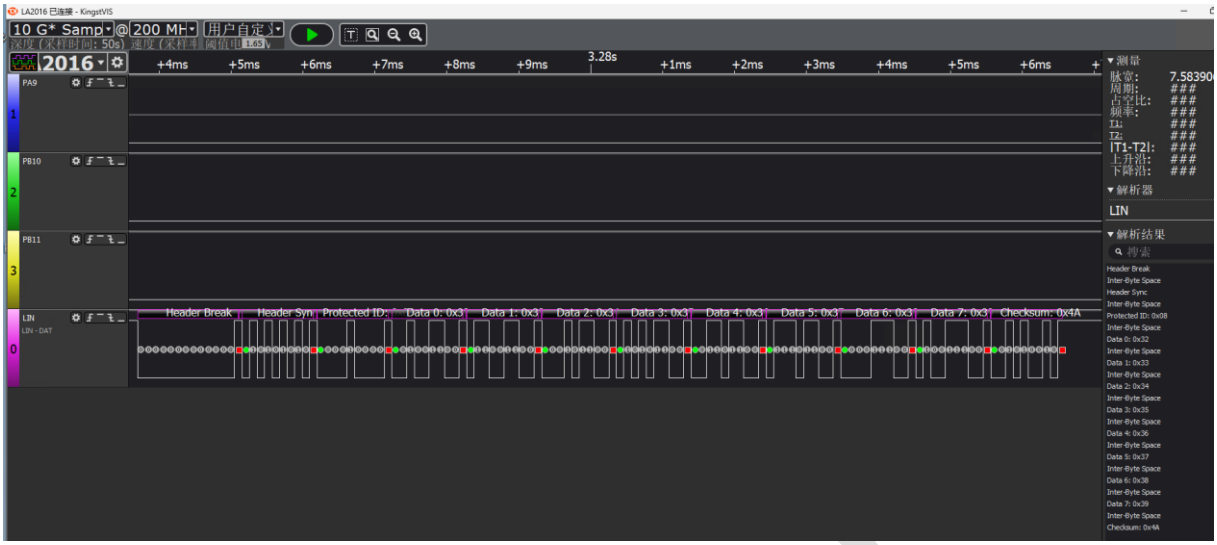
```
void USART_LINComPolling(void)
{
    uint32_t i = 0;
    /*##-1- Configure the USART peripheral ######################################*/
    /* Put the USART peripheral in the LIN mode (LIN Mode) */
    /* USART7 configured as follow:
      - BaudRate = 9600 baud
      - OverSampling =  OverSampling 16 */
    UsartHandle.Instance             = USART;

    UsartHandle.LinInit.BaudRate     = 9600;
    UsartHandle.LinInit.OverSampling = USART_OVERSAMPLING_16;


#ifdef MASTER_BOARD
    UsartHandle.LinInit.Mode         = USART_LIN_MODE_MASTER;
#else
    UsartHandle.LinInit.Mode         = USART_LIN_MODE_SLAVE;
#endif /* MASTER_BOARD */
```


2. 调用 HAL_USART_LIN_Publish 函数数据并启动发送。

```
/*##-2- Start LIN Master Publish Communication process #########################*/
if(HAL_USART_LIN_Publish(&UsartHandle, &id, aTxBuffer, BUFFERSIZE, 50000) != HAL_OK)
{
    /* Initialization Error */
    Error_Handler();
}
```


Copyright © 2025 广芯微电子（广州）股份有限公司

3. 使用支持 LIN 的逻辑分析仪，设置波特率为 9600，即可解析出数据。



4.2 LIN 的接收配置

LIN 接收的基本配置与发送配置基本一致，需要注意的是接收端配置为从模式。只有当接收到与主机发送一致的标识符位时，LIN 的接收才有可能将接收到的消息传递给 RHR。

1. LIN 接收的基本配置。

```
    UsartHandle.Instance             = USART;

    UsartHandle.LinInit.BaudRate     = 9600;
    UsartHandle.LinInit.OverSampling = USART_OVERSAMPLING_16;

#ifdef MASTER_BOARD
    UsartHandle.LinInit.Mode         = USART_LIN_MODE_MASTER;
#else
    UsartHandle.LinInit.Mode         = USART_LIN_MODE_SLAVE;
#endif /* MASTER_BOARD */
```

波特率

过采样

LIN从模式

2. 调用 HAL_USART_LIN_Subscribe 函数接收数据操作。

```
/*##-2- Start LIN Slave Subscribe Communication process ##########################*/
if(HAL_USART_LIN_Subscribe(&UsartHandle, &id, aRxBuffer, BUFFERSIZE, 50000) != HAL_OK)
{
    /* Initialization Error */
    Error_Handler();
}

for(i=0;i<BUFFERSIZE;i++)
{
    printf("Rxdata%d = 0x%x\r\n",i,aRxBuffer[i]);
}
```

超时时间

Copyright © 2025 广芯微电子（广州）股份有限公司

5 注意事项

自研 LIN 驱动时，需要格外注意：标识符（ID）为 0x3C 和 0x3D 的 LIN 帧为诊断帧（仅 LIN 2.0 及以上版本支持），必须使用经典校验（Classic Checksum），禁止将 PID（Protected ID）纳入校验计算序列。

Unicmicro

Copyright © 2025 广芯微电子（广州）股份有限公司

6 版本维护

版本	日期	描述
V1.0	2025.11.12	初始版

Unicmicro

Copyright © 2025 广芯微电子（广州）股份有限公司