RT1170 LIN demo User Guide

ConstYu

功能需求

RT117X系列MCU在汽车和工业类产品中有广泛应用，有很多客户对LIN通讯有需求，RT1176有12路独立的LPUART接口，最大支持的波特率能支持到20M，而且每一路都支持Break发送和中断接收，可以用来配合定时器实现LIN的主机和从机通讯。但是目前RT117X的EVK板没有放置LIN的收发器，SDK也没有相关LIN的示例代码和LIN协议栈支持，所以本示例目的是移植KW36工程中的LIN 2.1版本的代码到RT1176 EVK板子上，在硬件上通过跳线将LIN Master主节点和 Slave从节点的LPUART TX/RX线连接到FRDM-KW36板载的LIN收发器TJA1027上，分别实现LIN 2.1版本协议栈在Master和Slave节点的通讯功能验证，同时还需支持Auto Baud Rate自动波特率调整。为客户做二次开发或者移植用户自己的LIN stack提供底层驱动，提高开发效率。

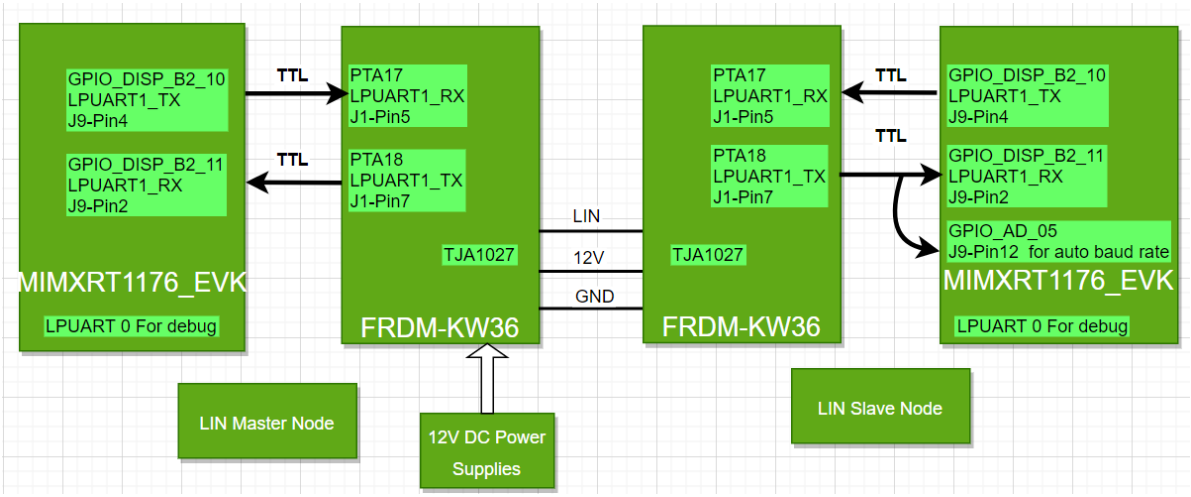
代码包软件

- RT1176 LIN Master节点代码：RT1170_LIN_Porting_Demo_Master.7z
- RT1176 LIN Slave节点代码(支持自动波特率): RT1170_LIN_Porting_Demo_Slave_with_Auto_Baud_Rate.7z
- 配置FRDM-KW36板载LIN 收发器的代码: KW36_LIN_PHY_Board_Init.7z

硬件Setup

- MIMXRT1170-EVK： 2pcs，分别用作LIN Master节点和Slave节点。
- FRDM-KW36：2pcs， 分别用作Master节点的收发器，和 Slave节点的收发器。

下图是系统连接，2块RT1170 EVK板分别和2块FRDM-KW36板通过Arduino接口连接在一起，然后将两块KW36之间的LIN收发器通过 J13 连接在一起，需要使用外部12V adapter为FRDM-KW36供电，否则板上的LIN收发器无法工作。特别强调的是，如果需要使能自动波特率检测的话，还需要将Slave节点RT1176 Arduino接口的J9-Pin2引脚连接到RT1176 Arduino接口的J9-Pin12引脚，作为Timer 脉冲捕捉的输入，即可完成系统硬件的setup。



软件Setup：

在以上硬件连接完成后，按照如下步骤下载对应软件：

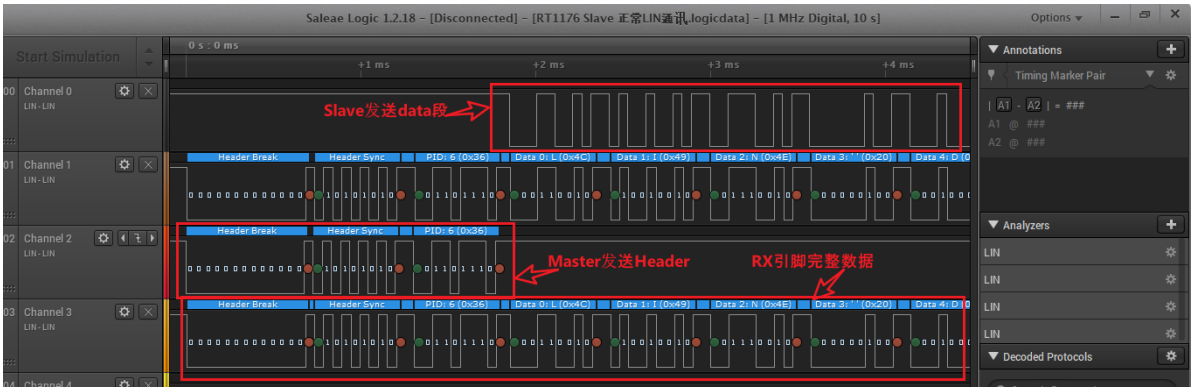
- **Step1**: 下载KW36_LIN_PHY_Board_Init.7z代码到两块FRDM-KW36板子上；

> 该代码中主要实现两个功能：第1个拉高板子的PTC5引脚，唤醒LIN收发器TJA1027。第2个将PTA18引脚配置成disable高阻状态。如果该引脚作为GPIO输出或者LPUART TX功能，会导致LIN slave回应数据出错(bit位丢失或者错误)。究其原因猜测应该是短路导致，当这个引脚作为GPIO输出或者LPUART TX功能，内部会有上拉，当RX1176 TX引脚输出Low时，由于电路上没有串联电阻(板子上使用的0Ω)，会导致引脚上出现大电流。尤其是第2个点，花费了很多时间去查这个问题，从波形去看，是有数据输出的，但只是数据不对，很具有迷惑性。当然如果客户是自己打的板子，板子上已经有LIN收发器就不需要这一步，直接跳到Step2即可。

- **Step2**: 下载RT1170_LIN_Porting_Demo_Master.7z代码到作为Master节点的IMRT1176-EVK板；
- **Step3**: 下载RT1170_LIN_Porting_Demo_Slave_with_Auto_Baud_Rate.7z代码到作为Slave节点的IMRT1176-EVK板，如果需要使能自动波特率调整，需要配置宏 linUserConfigSlave.autobaudEnable = true; 代码中默认是打开的。

实验结果

打开两个IMRT1176-EVK板串口，波特率配置115200，单击RT1176 Master节点上的按键SW7，便可以启动Master节点开始发送数据，通讯波形和串口打印信息如下两张图所示。



代码移植的几个难点

1. LIN通讯协议栈的调度流程的理解，包括Wakeup段，Break段，Sync段，PID段，Data段的状态切换和跳转，每个段的超时监测和错误处理，其核心思想有两个：一个在于LIN的RX引脚要不断去monitor TX引脚的状态，然后去切换状态机，具体调度的流程在后文会详细介绍，这里不展开。第二个是准确获取在每个段的定时器时间，尤其是超时超过一个overflow周期的情况，需要对timerGetTimeIntervalCallback0函数有理解。
2. 自动波特率调整功能的支持，该功能的原理是测量SYNC段的8个脉冲的脉宽，如果每个脉宽差异在2%范围内，再根据脉冲宽度去判断对应的波特率。在原来KW36的代码中是使用TPM的Overflow中断来作为计时，Edge中断来触发，而RT1176没有TPM，只能使用Qtimer (Qtimer功能上要更强于TPM)，但是不巧的是Qtimer不支持Overflow中断(参见芯片ERRATA 050194)，所以只能使用compare中断来实现类似的功能，而原有的计时定时计算都是基于overflow的，因此就需要对定时器部分的代码做大范围的更改。

> 应用中考虑到timerGetTimeIntervalCallback0函数在自动波特率调整时和超时监测处理时的一致性，最好使用同一个Timer的同一个channel，这就需要这个Timer既支持普通的定时中断模式，又支持input capture功能。对于TPM来说，是无法实现的因为两次在寄存器配置上时互斥的, 参见下图。幸运的是Qtimer支持这个feature，只是需要根据SDK代码做些配置

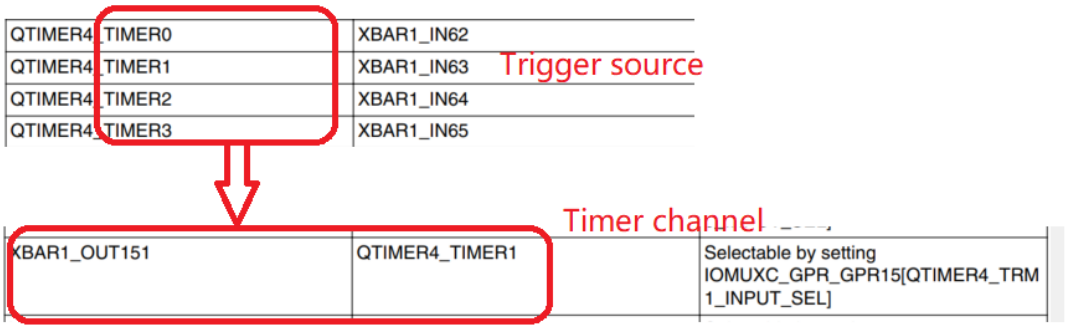
Table 33-2. Mode, Edge, and Level Selection

CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
X	00	00	None	Channel disabled
X	01	00	Software compare	Pin not used for TPM
0	00	01	Input capture	Capture on Rising Edge Only
		10		Capture on Falling Edge Only
		11		Capture on Rising or Falling Edge
	01	01	Output compare	Toggle Output on match

3. 前面提到，需要QTimer支持input capture功能, 触发信号是LPUART_RX引脚的信号，需要硬件loop到Qtimer支持的硬件引脚上，对于KW36来说，只需要把这两个物理引脚连接在一起即可，但对RT1176来说, 只有这一步还不行，还需要对XBAR进行配置，将Qtimer的TIMER 1的触发引脚(合计有4个物理引脚)Link到QTIMER对应的Channel上，因为RT1176有4个QTimer，每个Qtimer有4个通道，标称的Qtimer trigger pin有4个，那具体哪个pin触发哪个QTimer的哪个通道，是需要配置的。如果客户没有使用过XBAR配置起来有难度，还好MCUXpresso config tool支持配置，可以简便的完成配置。示例代码和触发关系如下，如果实际硬件使用的物理引脚有区别，需要对应修改。

> RT的XBAR功能非常强大，或许可以不使用外部的物理连线，直接将Qtimer的出发引脚的信号直接在内部Loop到LPUART_RX引脚，这样就更加灵活，此处只提供一个思路，不再进一步延伸。

```
IOMUXC_GPR->GPR15 = ((IOMUXC_GPR->GPR15 &
    (~(IOMUXC_GPR_GPR15_QTIMER4_TRM1_INPUT_SEL_MASK |
IOMUXC_GPR_GPR15_QTIMER4_TRM2_INPUT_SEL_MASK)))/*Mask bits to zero which are
setting*/
 | IOMUXC_GPR_GPR15_QTIMER4_TRM1_INPUT_SEL(0x00U) /*QTIMER4 TMR1 input
select: 0x00U*/
 | IOMUXC_GPR_GPR15_QTIMER4_TRM2_INPUT_SEL(0x00U) /*QTIMER4 TMR2 input
select: 0x00U*/
    );
```



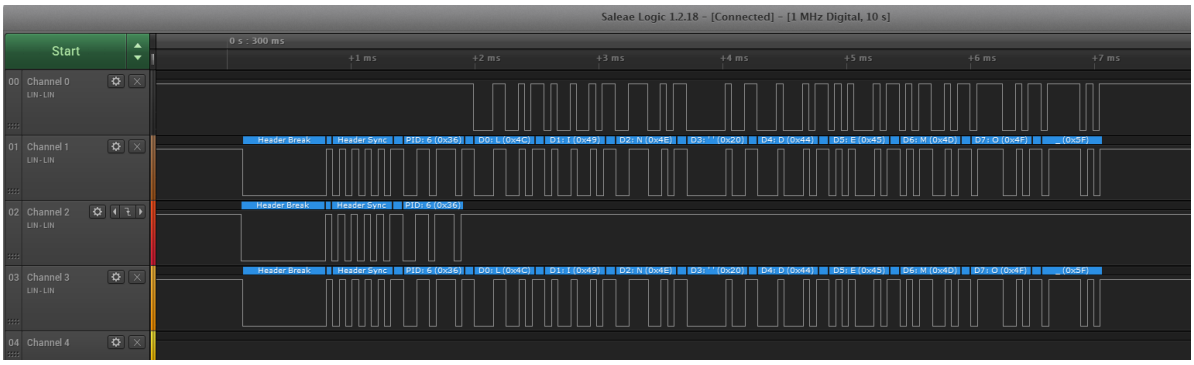
4. 在状态机切换和超时以及错误处理过程中，经常会看到两种模式Sleep模式和Idle模式，区别是什么呢?

   LIN_LPUART_GoToSleepMode: 函数会关闭Break中断，RX接收中断，帧错误中断，保留RX边沿中断;

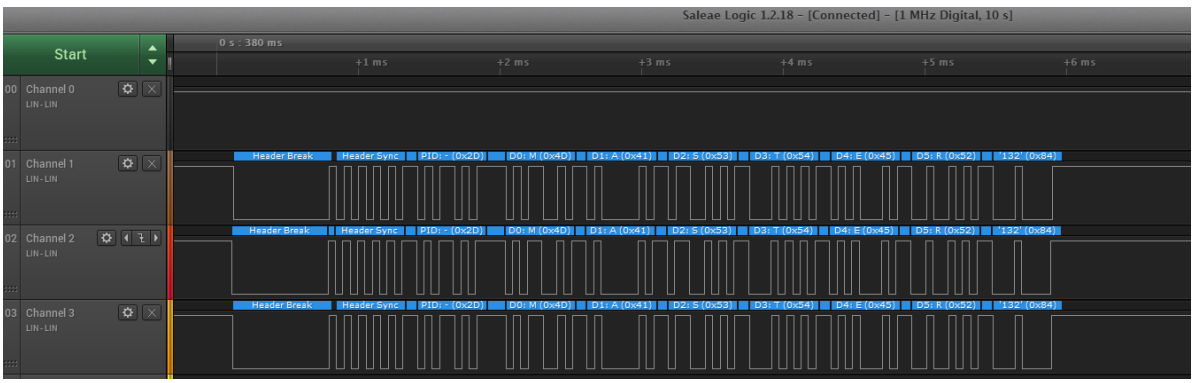
   LIN_LPUART_GotoIdleState　函数会打开Break中断，RX接收中断，帧错误中断，关闭RX边沿中断;

实际通讯波形

Master作为Subscribe角色时，发送Header，由Slave发送Respone

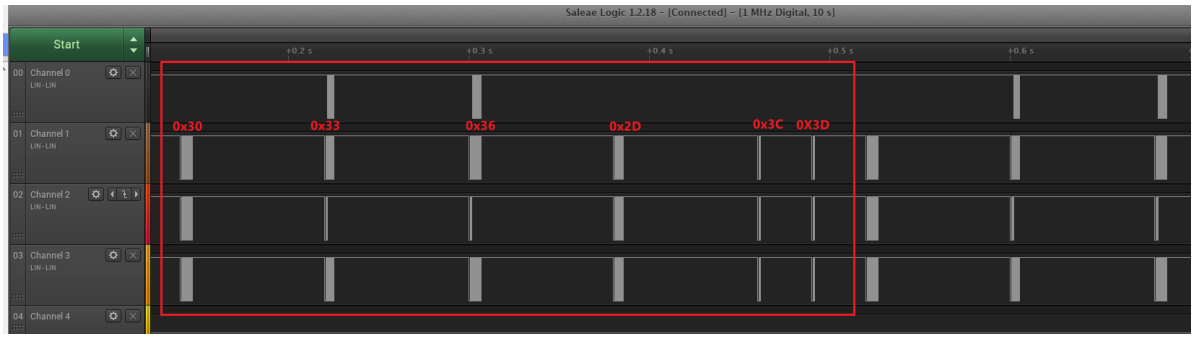


Master作为PUBLIC角色时，同时发送Header，以及Respone

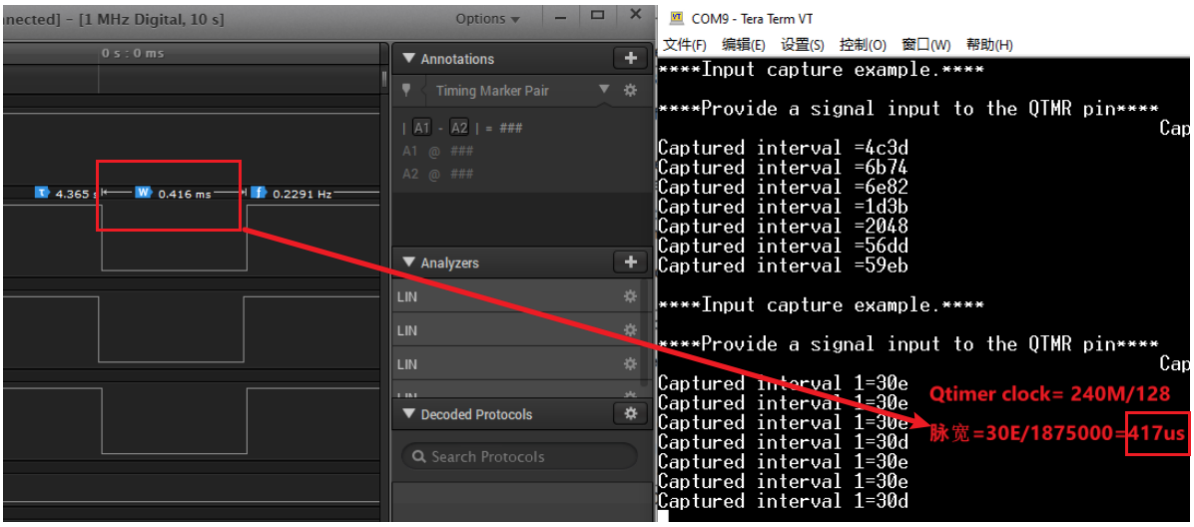


按照调度表依次发送LI0_lin_configuration_RAM数组定义的PID数据

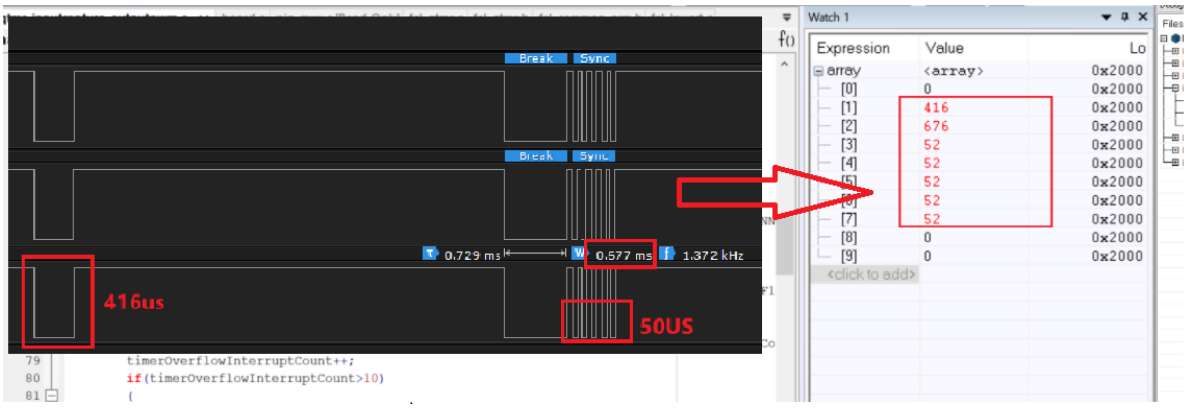
> static uint8_t LI0_lin_configuration_RAM[LI0_LIN_SIZE_OF_CFG]= {0x00, 0x30, 0x33, 0x36, 0x2D, 0x3C, 0x3D ,0xFF};



Qtimer准确读取wake up信号的脉冲宽度



Slave使能Auto baud rate后读取到的每个脉冲宽度数据



免责声明:

ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF * ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.