AN14319

用FlexIO模拟带IRDA功能的UART

第1.0版—2024年7月15日

应用笔记

文档信息

信息	内容
关键词	AN14319、MCXC444、MCXC242、FlexIO、UART、IRDA
摘要	本应用笔记介绍了如何使用FlexIO模拟带IRDA功能的UART。

1 介绍

本应用笔记介绍了如何使用通用外设模块FlexIO来模拟带IRDA功能的UART总线。FlexIO外设最初在MCXC242和MCXC444系列上推出，它是一个高度可配置的模块，能够模拟多种不同的通讯协议。这些通讯协议包括UART、I2C、SPI、I2S等。

独立的FlexIO外设模块并非要取代UART外设，而是作为MCU的一个附加外设模块。此外设的重要特点在于它使用户能直接在MCU中构建自己的外设。

这个UART模块用例基于独立的接收器和发送器创建了一个简单的软件驱动程序。在本演示中，使用了恩智浦FRDM系统。模拟的UART总线的最大测试波特率为115200波特。

此用例使用FlexIO UART驱动程序来创建一个IRDA协议，UART支持IRDA。除了FlexIO UART驱动程序外，还使用了另外两个定时器，将UART信号编码和解码为IRDA波形。

2 特性

FlexIO外设模块的主要特性如下：

- FlexIO意味着灵活的输入和输出外设
- 具有多种功能的高度可配置模块
- 允许模拟标准通讯接口
- 支持多种协议和外设，包括：
  –UART
  –I2C
  –SPI
  –I2S
- 在软件模拟GPIO的方法和硬件外设模块之间建立连接

3 硬件和软件要求

本文档介绍了基于恩智浦FRDM-MCXC242开发板的示例应用程序。这种基本概念也可以在定制硬件上轻松地实现。

本示例展示了使用FlexIO模块进行的UART通讯，配置如下：

- 8个数据位
- 1个停止位
- 无奇偶校验位
- 无硬件流控制

图1所示为FlexIO模拟的UART数据帧。

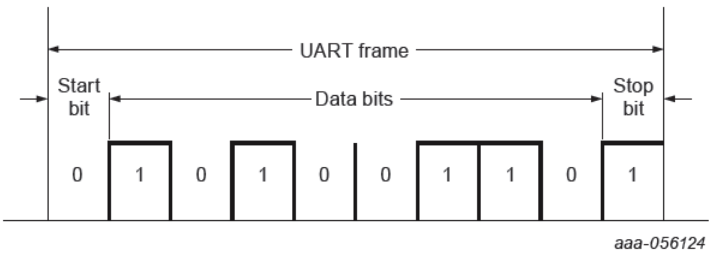

图1. UART 8位数据帧

4 UART概述

通用异步接收器/发送器是一种在并行和串行之间转换数据的计算机硬件。UART通常与EIA、RS-232、RS-422或RS-485等通讯标准一起使用。通用名称表示数据格式和传输速度是可配置的。UART外部的驱动电路可处理电信号电平和方法，如差分信号。UART通常是一个单独的（或一部分）集成电路，用于通过计算机或外设串行端口进行串行通讯。

发送和接收UART必须设置为相同的比特、字符长度、奇偶校验位和停止位，才能正常工作。接收UART可以检测到一些不匹配的设置，并为主机系统设置一个“帧错误”标志位。在特殊情况下，接收UART会产生一个不稳定的残缺字符流，并将其传输到主机系统。

与连接到调制解调器的PC配套使用的典型串行端口使用了以下配置：

- 8个数据位
- 无奇偶校验位
- 1个数据位

对于此种配置，每秒的ASCII字符数等于比特率除以10。

5 使用FlexIO模块进行UART模拟

本节介绍了如何使用FlexIO模块进行UART模拟。

- 可以通过使用两个定时器、两个移位器和两个引脚来支持UART总线：
  - 使用一个定时器、一个移位器和一个引脚支持发送器。
  - 使用一个定时器、一个移位器和一个引脚支持接收器。发送器和接收器部件都可以独立使用。
- FlexIO外设会自动处理起始位和停止位插入。
- 所模拟外设的最大波特率为115200波特。
- 软件实现允许在中断或轮询模式下使用UART。
- 中断和空闲字符需要软件干预，未在此示例应用程序中实现。
- 可配置的位顺序（位交换缓冲区MSB在前），且使用DMA控制器可以支持多次传输。
- FlexIO模块不允许自动插入奇偶校验位。

图2所示为FlexIO模拟UART的内部连接。

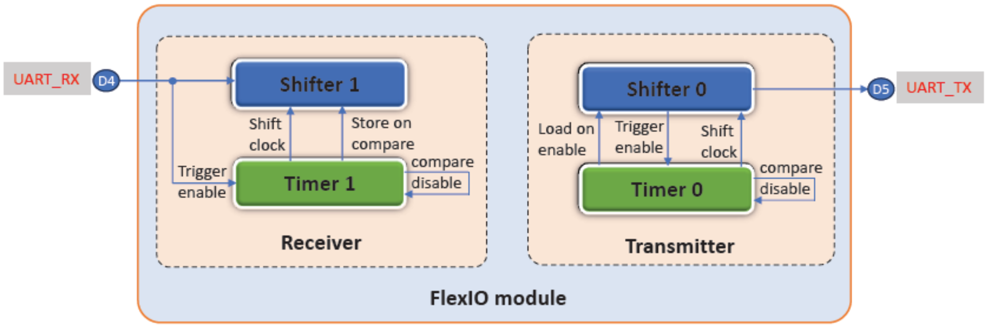

图2. FlexIO模拟UART

5.1 发送器

发送的过程包括以下步骤：

1. 将移位器设置为发送器（Transmit）模式
2. 从移位器缓冲区移位加载的数据
3. 将数据移至引脚输出
4. 起始位和停止位是在数据之前或之后自动加载的
5. 使用定时器状态标志发送下一个数据帧

图3所示为UART发送器模拟的原理。在轮询模式下检查定时器状态标志，当中断设置启用时，模块会产生一个中断。

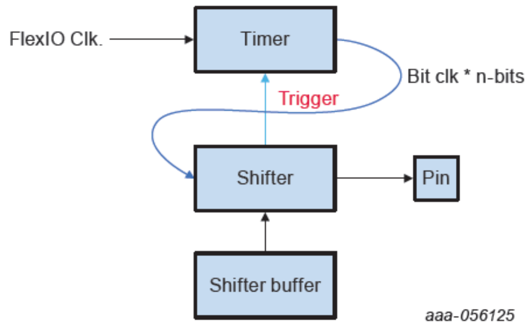

图3. FlexIO模块的UART发送器结构框图

5.2 接收器

接收的过程包括以下步骤：

1. 将移位器设置为接收器（Receiver）模式
2. 当存储事件发出信号时，数据被移入
3. 状态标志会指示何时可以读取数据（生成中断）
4. 在轮询模式下等待移位器状态标志
5. 存储到移位器缓冲区中
6. 读取位交换移位器缓冲区（不进行任何逻辑操作）

图4所示为UART接收器模拟的原理。在轮询模式下检查移位器状态标志，当中断设置启用时，模块会产生一个中断。

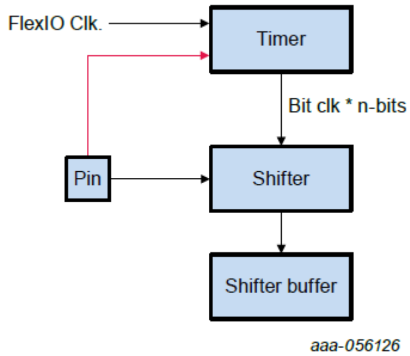

图4. FlexIO模块的UART接收器结构框图

6 用FlexIO模块实现的带IRDA编码/解码功能的UART

UART数据采用NRZ格式。为了将这些数据编码到IRDA协议中，需要一个双8位计数器PWM模式的FlexIO定时器来调制NRZ数据。为了接收IRDA信号，需要一个双8位计数器波特/位模式的FlexIO定时器将IRDA信号解码为NRZ格式。

图5所示为此用例的框图。

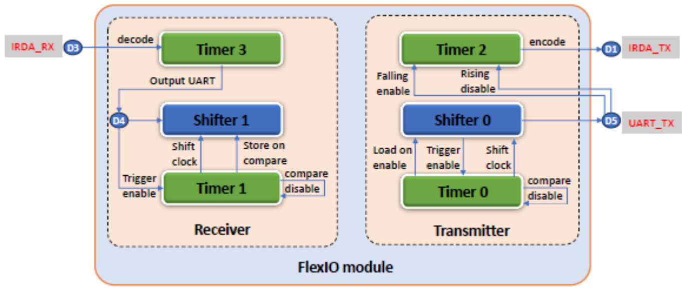

图5. FlexIO模块实现带IRDA编码/解码功能的UART

图6所示为波形图。

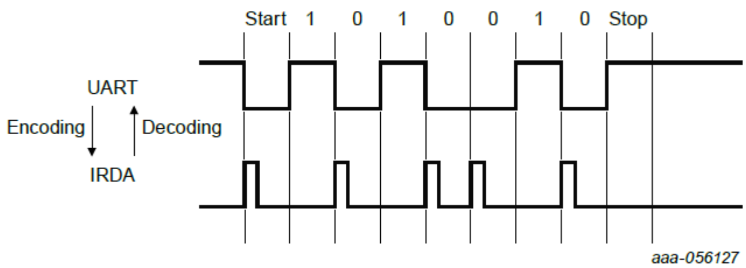

图6. FlexIO模块的 UART和IRDA波形图

6.1 IRDA编码定时器的配置

编码定时器被配置为由UART NRZ数据的下降沿触发，该下降沿代表起始信号的初始沿。

定时器配置的详细信息如下：

- 定时器控制寄存器（FLEXIO_TIMCTLn）：
  - TRGSEL：选择FlexIO UART TX输出引脚。

  – TRGPOL：选择触发极性低电平有效。
  – TRGSRC：选择内部触发源。
  – PINCFG：选择定时器引脚输出。
  – PINSEL：选择要使用的定时器引脚。选择一个FlexIO UART TX未使用的引脚。
  – PINPOL：选择定时器的输出极性。根据所使用的外部IRDA设备，它可以是高真值或低真值。
  – TIMOD：选择定时器的运行模式，双8位计数器PWM高模式。
- 定时器配置寄存器（FLEXIO_TIMCFGn）：
  – TIMOUT：选择定时器输出逻辑1，不受定时器复位的影响
  – TIMDEC：选择FlexIO时钟的定时器递减
  – TIMRST：选择定时器永不重置
  – TIMDIS：选择在触发下降沿时禁用定时器
  – TIMENA：选择在触发上升沿时启用定时器
  – TSTOP：禁用
  – TSTART：禁用
- 定时器比较寄存器（FLEXIO_TIMCMPn）：
  – CMP：要设置的定时器值。
  在双8位计数器PWM模式下：
  – 低8位配置高电平周期输出=（CMP[7:0] +1）。
  – 高8位配置低电平周期输出=（CMP[15:8] +1）。
  对于IRDA，高电平周期必须为3/16的UART周期时间。

6.2 IRDA解码时间的配置

解码定时器被配置为由IRDA数据的上升沿触发，该上升沿代表起始信号的初始沿。

定时器配置的详细信息如下：

- 定时器控制寄存器（FLEXIO_TIMCTLn）：
  – TRGSEL：选择FlexIO引脚输入。
  – TRGPOL：选择触发极性高电平有效，IRDA接收器输出信号极性。
  – TRGSRC：根据TRGSEL配置选择内部触发源。
  – PINCFG：选择定时器引脚输出使能。
  – PINSEL：选择要使用的定时器引脚。
  – PINPOL：选择定时器的输出极性：低电平有效。
  – TIMOD：选择定时器的运行模式：单16位计数器模式。
- 定时器配置寄存器（FLEXIO_TIMCFGn）：
  – TIMOUT：当启用和定时器复位时，选择定时器输出逻辑1。
  – TIMDEC：选择FlexIO时钟的定时器递减。
  – TIMRST：选择在定时器触发上升沿时复位定时器。
  – TIMDIS：选择在比较定时器时禁用定时器。
  – TIMENA：选择在触发上升沿时启用计时器。
  – TSTOP：选择禁用定时器停止位。
  – TSTART：选择禁用定时器起始位。
- 定时器比较寄存器（FLEXIO_TIMCMPn）：
  – CMP：要设置的定时器值。
  定时器以双8位计数器波特/位模式运行。
  在16位计数器模式下，比较值可用于生成波特率除法器=（CMP[15:0]+1）*2。

7 软件实现

本节介绍了用于UART和IRDA的FlexIO的配置。MCUXpresso配置工具能生成代码。要了解详情，请参阅《MCX C24X子系列参考手册》（文档MCXC242RM）中相应的寄存器章节。

7.1 FlexIO UART的驱动程序初始化

寄存器配置如下：

```
/* Definitions for FlexIO_UART_Init functional group */
/* FLEXIO_CTRL: DOZEN=1, DBGE=1, FASTACC=0, FLEXEN=1 */
#define FLEXIO_UART_INIT_CTRL_INIT 0xC0000001U
/* FLEXIO_SHIFTSIEN: SSIE=0 */
#define FLEXIO_UART_INIT_SHIFTSIEN_INIT 0x0U
/* FLEXIO_SHIFTEIEN: SEIE=0 */
#define FLEXIO_UART_INIT_SHIFTEIEN_INIT 0x0U
/* FLEXIO_TIMIEN: TEIE=0 */
#define FLEXIO_UART_INIT_TIMIEN_INIT 0x0U
/* FLEXIO_SHIFTSDEN: SSDE=0 */
#define FLEXIO_UART_INIT_SHIFTSDEN_INIT 0x0U
/* FLEXIO_SHIFTCTL0: TIMSEL=0, TIMPOL=0, PINCFG=3, PINSEL=5, PINPOL=0, SMOD=2 */
#define FLEXIO_UART_INIT_SHIFTCTL0_INIT 0x30502U
/* FLEXIO_SHIFTCTL1: TIMSEL=1, TIMPOL=1, PINSEL=4, PINPOL=0, SMOD=1 */
#define FLEXIO_UART_INIT_SHIFTCTL1_INIT 0x1800401U
/* FLEXIO_SHIFTCFG0: INSRC=0, SSTOP=3, SSTART=2 */
#define FLEXIO_UART_INIT_SHIFTCFG0_INIT 0x32U
/* FLEXIO_SHIFTCFG1: INSRC=0, SSTOP=3, SSTART=2 */
#define FLEXIO_UART_INIT_SHIFTCFG1_INIT 0x32U
/* FLEXIO_TIMCTL0: TRGSEL=1, TRGPOL=1, TRGSRC=1, PINCFG=0, PINSEL=5, PINPOL=0,
 TIMOD=1 */
#define FLEXIO_UART_INIT_TIMCTL0_INIT 0x1C00501U
/* FLEXIO_TIMCTL1: TRGSEL=8, TRGPOL=1, TRGSRC=1, PINCFG=0, PINSEL=0, PINPOL=0,
 TIMOD=1 */
#define FLEXIO_UART_INIT_TIMCTL1_INIT 0x8C00001U
/* FLEXIO_TIMCFG0: TIMOUT=0, TIMDEC=0, TIMRST=0, TIMDIS=2, TIMENA=2, TSTOP=2,
 TSTART=1 */
#define FLEXIO_UART_INIT_TIMCFG0_INIT 0x2222U
/* FLEXIO_TIMCFG1: TIMOUT=2, TIMDEC=0, TIMRST=6, TIMDIS=2, TIMENA=6, TSTOP=2,
 TSTART=1 */
#define FLEXIO_UART_INIT_TIMCFG1_INIT 0x2062622U
/* FLEXIO_TIMCMP0: CMP=3874 */
#define FLEXIO_UART_INIT_TIMCMP0_INIT 0xF22U
/* FLEXIO_TIMCMP1: CMP=3874 */
#define FLEXIO_UART_INIT_TIMCMP1_INIT 0xF22U
```

7.2 FlexIO IRDA的驱动程序初始化

寄存器配置如下：

```
/* Definitions for FlexIO_UART_IRDA_Init functional group */
/* FLEXIO_CTRL: DOZEN=1, DBGE=1, FASTACC=0, FLEXEN=1 */
#define FLEXIO_UART_IRDA_INIT_CTRL_INIT 0xC0000001U
/* FLEXIO_SHIFTSIEN: SSIE=0 */
#define FLEXIO_UART_IRDA_INIT_SHIFTSIEN_INIT 0x0U
/* FLEXIO_SHIFTEIEN: SEIE=0 */
```

```
#define FLEXIO_UART_IRDA_INIT_SHIFTEIEN_INIT 0x0U
/* FLEXIO_TIMIEN: TEIE=0 */
#define FLEXIO_UART_IRDA_INIT_TIMIEN_INIT 0x0U
/* FLEXIO_SHIFTSDEN: SSDE=0 */
#define FLEXIO_UART_IRDA_INIT_SHIFTSDEN_INIT 0x0U
/* FLEXIO_SHIFTCTL0: TIMSEL=0, TIMPOL=0, PINCFG=3, PINSEL=5, PINPOL=0, SMOD=2 */
#define FLEXIO_UART_IRDA_INIT_SHIFTCTL0_INIT 0x30502U
/* FLEXIO_SHIFTCTL1: TIMSEL=1, TIMPOL=1, PINSEL=4, PINPOL=0, SMOD=1 */
#define FLEXIO_UART_IRDA_INIT_SHIFTCTL1_INIT 0x1800401U
/* FLEXIO_SHIFTCFG0: INSRC=0, SSTOP=3, SSTART=2 */
#define FLEXIO_UART_IRDA_INIT_SHIFTCFG0_INIT 0x32U
/* FLEXIO_SHIFTCFG1: INSRC=0, SSTOP=3, SSTART=2 */
#define FLEXIO_UART_IRDA_INIT_SHIFTCFG1_INIT 0x32U
/* FLEXIO_TIMCTL0: TRGSEL=1, TRGPOL=1, TRGSRC=1, PINCFG=0, PINSEL=5, PINPOL=0,
 TIMOD=1 */
#define FLEXIO_UART_IRDA_INIT_TIMCTL0_INIT 0x1C00501U
/* FLEXIO_TIMCTL1: TRGSEL=8, TRGPOL=1, TRGSRC=1, PINCFG=0, PINSEL=0, PINPOL=0,
 TIMOD=1 */
#define FLEXIO_UART_IRDA_INIT_TIMCTL1_INIT 0x8C00001U
/* FLEXIO_TIMCTL2: TRGSEL=10, TRGPOL=1, TRGSRC=1, PINCFG=3, PINSEL=1, PINPOL=0,
 TIMOD=2 */
#define FLEXIO_UART_IRDA_INIT_TIMCTL2_INIT 0xAC30102U
/* FLEXIO_TIMCTL3: TRGSEL=6, TRGPOL=0, TRGSRC=1, PINCFG=3, PINSEL=4, PINPOL=1,
 TIMOD=3 */
#define FLEXIO_UART_IRDA_INIT_TIMCTL3_INIT 0x6430483U
/* FLEXIO_TIMCFG0: TIMOUT=0, TIMDEC=0, TIMRST=0, TIMDIS=2, TIMENA=2, TSTOP=2,
 TSTART=1 */
#define FLEXIO_UART_IRDA_INIT_TIMCFG0_INIT 0x2222U
/* FLEXIO_TIMCFG1: TIMOUT=2, TIMDEC=0, TIMRST=6, TIMDIS=2, TIMENA=6, TSTOP=2,
 TSTART=1 */
#define FLEXIO_UART_IRDA_INIT_TIMCFG1_INIT 0x2062622U
/* FLEXIO_TIMCFG2: TIMOUT=0, TIMDEC=0, TIMRST=0, TIMDIS=6, TIMENA=6, TSTOP=0,
 TSTART=0 */
#define FLEXIO_UART_IRDA_INIT_TIMCFG2_INIT 0x6600U
/* FLEXIO_TIMCFG3: TIMOUT=2, TIMDEC=0, TIMRST=6, TIMDIS=2, TIMENA=6, TSTOP=0,
 TSTART=0 */
#define FLEXIO_UART_IRDA_INIT_TIMCFG3_INIT 0x2062600U
/* FLEXIO_TIMCMP0: CMP=3874 */
#define FLEXIO_UART_IRDA_INIT_TIMCMP0_INIT 0xF22U
/* FLEXIO_TIMCMP1: CMP=3874 */
#define FLEXIO_UART_IRDA_INIT_TIMCMP1_INIT 0xF22U
/* FLEXIO_TIMCMP2: CMP=14348 */
#define FLEXIO_UART_IRDA_INIT_TIMCMP2_INIT 0x380CU
/* FLEXIO_TIMCMP3: CMP=69 */
#define FLEXIO_UART_IRDA_INIT_TIMCMP3_INIT 0x45U
```

8 演示

本节介绍了以下两个演示：

- FlexIO UART演示
- 带IRDA的FlexIO UART演示

MCUXpresso配置工具会生成代码。

8.1 FlexIO UART演示

在本演示中，FlexIO模拟的UART通过USB串口线连接到PC。开发板返回PC发送给它的所有字符。

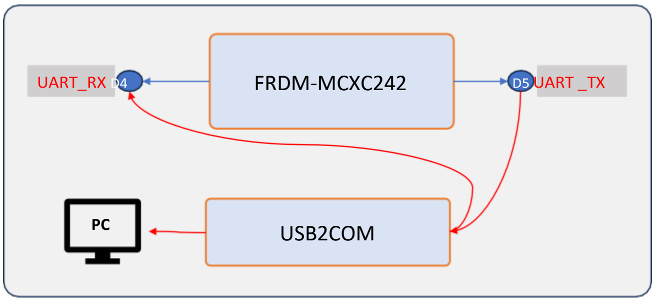

图7. FlexIO UART演示

8.1.1 软件配置

对于软件连接，使能FlexIO UART的定义如下：

```
#define FLEXIO_UART//FLEXIO_UART_IRDA//
#ifdef FLEXIO_UART

  /* Initialize components */
     FlexIO_UART_InitPins();
     FlexIO_UART_Init();

     uartDev.flexioBase       = BOARD_FLEXIO_BASE;
     uartDev.TxPinIndex       = FLEXIO_UART_TX_PIN;
     uartDev.RxPinIndex       = FLEXIO_UART_RX_PIN;
     uartDev.shifterIndex[0] = 0U;
     uartDev.shifterIndex[1] = 1U;
     uartDev.timerIndex[0]   = 0U;
     uartDev.timerIndex[1]   = 1U;

#endif
```

8.1.2 硬件连接

对于硬件连接，请执行以下步骤：

1. 将一根USB Type-C线插入J9，给FRDM-MCXC242开发板上电。
2. 使用UART引脚将一个USB连接到UART板：
   - J1-9，连接USB2COM的TX
   - J1-11，连接USB2COM的RX
   - J2-14，连接USB2COM的接地
3. 对于一个串行设备，在PC上打开串行终端，设置如下：
   - 波特率115200
   - 8个数据位
   - 无奇偶校验位
   - 1个停止位
   - 无流量控制
4. 将程序下载到目标板。
5. 要运行该演示，请按下开发板上的复位按钮或在IDE中启动调试器。

8.1.3 演示结果

当演示成功运行后，日志将出现在连接到USB2COM的UART终端端口上，如图8所示。

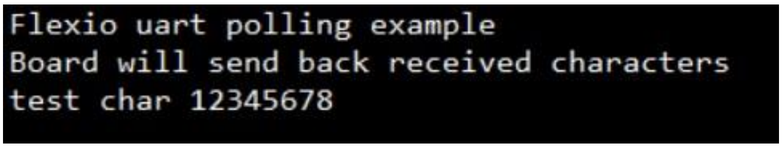

图8. 终端端口

该开发板会返回从PC UART工具发送的字符。

图9所示为逻辑设备中的时序捕获图。

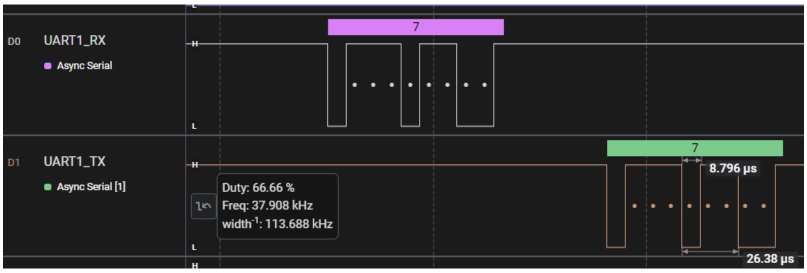

图9. 时序捕获图

8.2 带IRDA的FlexIO UART演示

在此演示中，一个FRDM-MCXC242模拟IRDA编码器，另一个FRDM-MCXC242模拟IRDA解码器。将一块开发板的IRDA_TX引脚连接到另一块开发板的IRDA_RX引脚。为了进行调试，IRDA_RX接收到的数据通过UART_TX引脚发送到USB2COM。

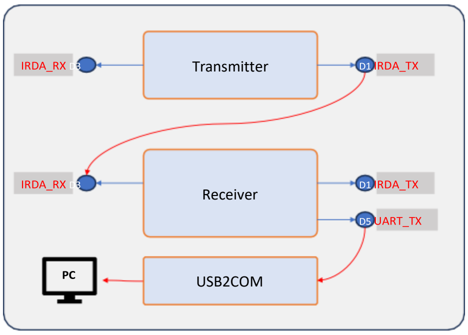

图10. 带IRDA的FlexIO UART演示

8.2.1 软件配置

对于软件连接，请执行以下步骤：

1. 启用FlexIO UART IRDA的定义如下：

```
#define FLEXIO_UART_IRDA//FLEXIO_UART//
#ifdef FLEXIO_UART_IRDA

    /* Initialize components */
    FlexIO_UART_IRDA_InitPins();
    FlexIO_UART_IRDA_Init();
```

```
    uartDev.flexioBase       = BOARD_FLEXIO_BASE;
    uartDev.TxPinIndex       = FLEXIO_UART_TX_PIN;
    uartDev.RxPinIndex       = FLEXIO_UART_RX_PIN;
    uartDev.shifterIndex[0] = 0U;
    uartDev.shifterIndex[1] = 1U;
    uartDev.timerIndex[0]   = 0U;
    uartDev.timerIndex[1]   = 1U;

#endif
```

2. 发送器的代码如下：

```
#define TRANSMITTER//RECEIVER//
#ifdef  TRANSMITTER
    ch = 0x30;
    while (1)
    {
     ch++;
        FLEXIO_UART_WriteBlocking(&uartDev, &ch, 1);
        if(ch == 0x5A)
        {
            ch = 0x30;
        }
        SDK_DelayAtLeastUs(10000, 48000000);
    }
#endif
```

3. 接收器的代码如下：

```
#define RECEIVER//TRANSMITTER//
#ifdef  RECEIVER
    FLEXIO_UART_WriteBlocking(&uartDev, txbuff, sizeof(txbuff) - 1);
    while (1)
    {
       FLEXIO_UART_ReadBlocking(&uartDev, &ch, 1);
        FLEXIO_UART_WriteBlocking(&uartDev, &ch, 1);
    }
#endif
```

8.2.2 硬件连接

对于硬件连接，请执行以下步骤：

1. 将一根USB Type-C线插入J9，给FRDM-MCXC242开发板上电。
2. 将发送器的IRDA_TX引脚连接到接收器的IRDA_RX引脚。
3. 用一根USB线将接收器板的UART_TX引脚连接到UART板：
    - J1-9，连接USB2COM的TX
    - J2-14，连接USB2COM的接地
4. 对于一个串行设备，在PC上打开串行终端，设置如下：
    - 波特率115200
    - 8个数据位
    - 无奇偶校验位
    - 1个停止位
    - 无流量控制
5. 将程序下载到目标开发板。
6. 要运行该演示，请按下开发板上的复位按钮或在IDE中启动调试器。

8.2.3 演示结果

当演示成功运行后，日志将出现在连接到USB2COM的UART终端端口上，如图11所示。

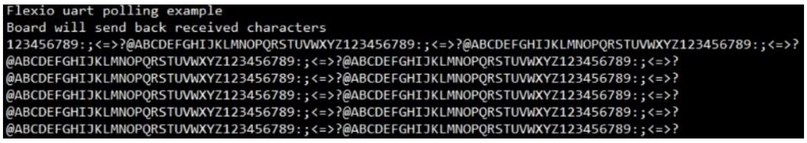

图11. 终端端口

发送器发送的字符会出现在UART终端端口上。

图12所示为信号的时序。

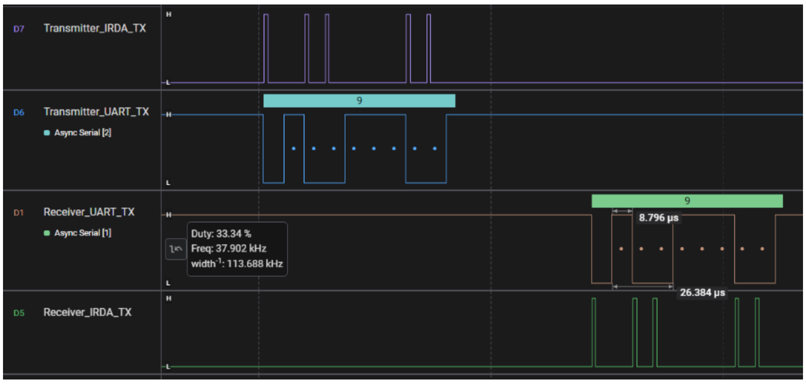

图12. 带IRDA的UART的信号时序

9 关于本文中源代码的说明

本文中所示的示例代码具有以下版权和BSD-3-Clause许可：

2024年恩智浦版权所有；在满足以下条件的情况下，可以源代码和二进制文件的形式重新分发和使用本源代码（无论是否经过修改）：

1. 重新分发源代码必须保留上述版权声明、这些条件和以下免责声明。
2. 以二进制文件形式重新分发时，必须在文档和/或随分发提供的其他材料中复制上述版权声明、这些条件和以下免责声明。
3. 未经事先书面许可，不得使用版权所有者的姓名或参与者的姓名为本软件的衍生产品进行背书或推广。

本软件由版权所有者和参与者“按原样”提供，不承担任何明示或暗示的担保责任，包括但不限于对适销性和特定用途适用性的暗示保证。在任何情况下，无论因何种原因或根据何种法律条例，版权所有者或参与者均不对因使用本软件而导致的任何直接、间接、偶然、特殊、惩戒性或后果性损害（包括但不限于采购替代商品或服务；使用损失、数据损失或利润损失或业务中断）承担责任，无论是因合同、严格责任还是侵权行为（包括疏忽或其他原因）造成的，即使事先被告知有此类损害的可能性也不例外。

10 修订历史

表1汇总了本文的修订情况。

表1. 修订历史

文档ID	发布日期	说明
AN14319 v.1.0	2024年7月15日	首次公开发布

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com.cn/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Freescale — is a trademark of NXP B.V.

MCX — is a trademark of NXP B.V.

Tower — is a trademark of NXP B.V.

目录

Date of release: 15 July 2024
Document identifier: AN14319