

8 位 PIC[®] 单片机上的通用异步收发器 (UART)

作者: Mary Tamar Tan
Microchip Technology Inc.

简介

UART 是用于在器件之间串行传输数据的通信硬件。8 位 PIC[®] 单片机提供了三种 UART 模块:

- 通用同步/异步收发器 (Universal Synchronous Asynchronous Receiver Transmitter, USART)
- 增强型通用同步/异步收发器 (Enhanced Universal Synchronous Asynchronous Receiver Transmitter, EUSART)
- 具有协议支持的通用异步收发器 (Universal Asynchronous Receiver Transmitter, UART)

USART 主要用于传统 PIC MCU。除了硬件 UART 的基本功能 (例如, 异步发送和接收) 外, USART 还可在同步主模式和从模式下工作。与 USART 相比, EUSART 具有“增强型”功能, 例如, 用于 LIN 的间隔 (Break) 字符支持和自动波特率检测。它通常用于增强型中档和高端器件。具有协议支持的 UART 是最新的 UART 模块, 针对 DMX512、数字可寻址照明接口 (Digital Addressable Lighting Interface, DALI) 和局域网互联网 (Local Interconnect Network, LIN) 等标准协议提供了硬件支持。PIC18FXXK42 系列是具有此类 UART 的器件示例之一。单个 8 位 PIC MCU 器件可能包含一个或多个 UART 模块。

本技术简介旨在概述 UART 模块的不同功能, 同时还给出了示例配置, 以展示模块在各种可能应用中实现时的稳健性和灵活性。

注: 除非另有说明, 否则本文档使用“UART 模块”作为通用术语来指代三种模块 (USART、EUSART 和具有协议支持的 UART)。

特性和协议支持

为了在器件之间保持稳定的串行通信, 并在数据传输过程中最大程序减少软件干预, 硬件全面支持中断驱动发送和接收、FIFO 缓冲和错误检测。许多器件还具有高级功能, 例如, 奇偶校验、校验和计算以及对 LIN、DMX 和 DALI 等标准协议的硬件支持。

表1给出了8位PIC单片机中三种现有类型UART模块的比较。

表1: 支持的特性和协议

硬件特性	USART	EUSART	具有协议支持的UART
全双工异步收发	✓	✓	✓
半双工同步主模式	✓	✓	—
半双工同步从模式	✓	✓	—
双字符输入缓冲区	✓	✓	✓
单字符输出缓冲区	✓	✓	✓
可编程字符长度	8位和9位	8位和9位	7位、8位和9位
第9位地址检测	—	✓	✓
第9位偶校验或奇校验	—	—	✓
输入缓冲区溢出错误检测	✓	✓	✓
接收字符帧错误检测	✓	✓	✓
硬件和软件流控制	—	—	✓
自动校验和	—	—	✓
可编程停止位	—	—	1、1.5和2
可编程数据极性	—	✓	✓
曼彻斯特编码器/解码器	—	—	✓
波特率的自动检测和校准	—	✓	✓
接收到间隔字符时唤醒	—	✓	✓
自动间隔周期生成	—	固定13位字符	自动LIN和DALI间隔周期和用户定时
协议支持			
LIN主模式和从模式	—	—	✓
DMX模式	—	—	✓
DALI控制装置和控制器件	—	—	✓

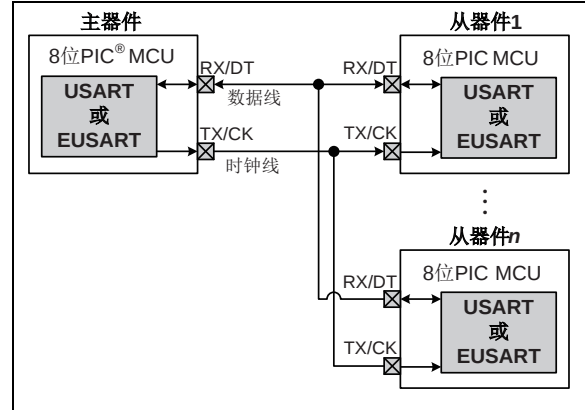
同步模式

USART和EUSART模块中的“S”代表“同步”。“同步”意味着器件之间的数据传输正在通过时钟信号同步。这种类型的通信无需使用信号传输方法，例如，异步系统中实现的启动位和停止位。

同步通信通常用于具有一个主器件和一个或多个从器件的系统中。主器件包含生成波特率所需的电路，并为系统中的所有器件提供时钟。从器件可使用主器件时钟，从而无需内部时钟发生电路。

在同步模式下，使用两个单片机引脚：RX/DT引脚和TX/CK引脚。RX/DT是双向数据线，而TX/CK是时钟线。从器件使用主器件提供的外部时钟将串行数据移入或移出相应的接收和发送移位寄存器。由于数据线是双向的，所以同步操作只能是半双工的。半双工指主从器件都能够接收和发送数据，但不能同时进行。USART和EUSART可作为主器件，也可作为从器件。典型配置如图1所示。

图1： 同步配置

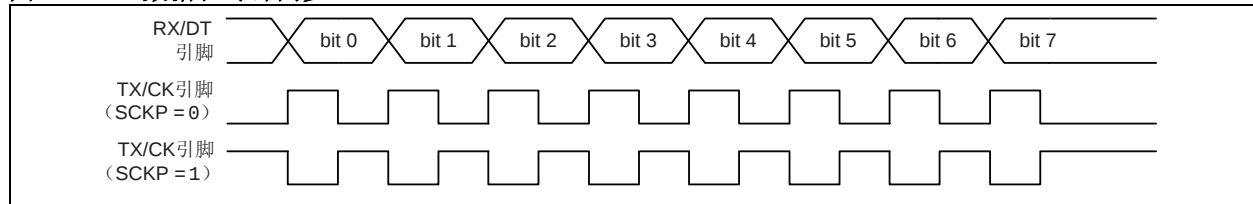


同步主模式

在同步主模式下，主器件在TX/CK线上发送时钟信号。可选时钟极性能够与Microwire兼容。请参见图2。

发送期间，将在模块配置为同步主操作时自动使能RX/DT和TX/CK引脚输出驱动器。数据位通过RX/DT引脚移出。每个数据位在主时钟的前沿改变，并在下一个时钟的前沿到来前保持有效。

图2： 数据和时钟同步



在同步主接收期间，自动禁止RX/DT引脚输出驱动器。生成的时钟周期数可配置为连续的数据位，也可配置为仅等于单个字符中的数据位数。

同步从模式

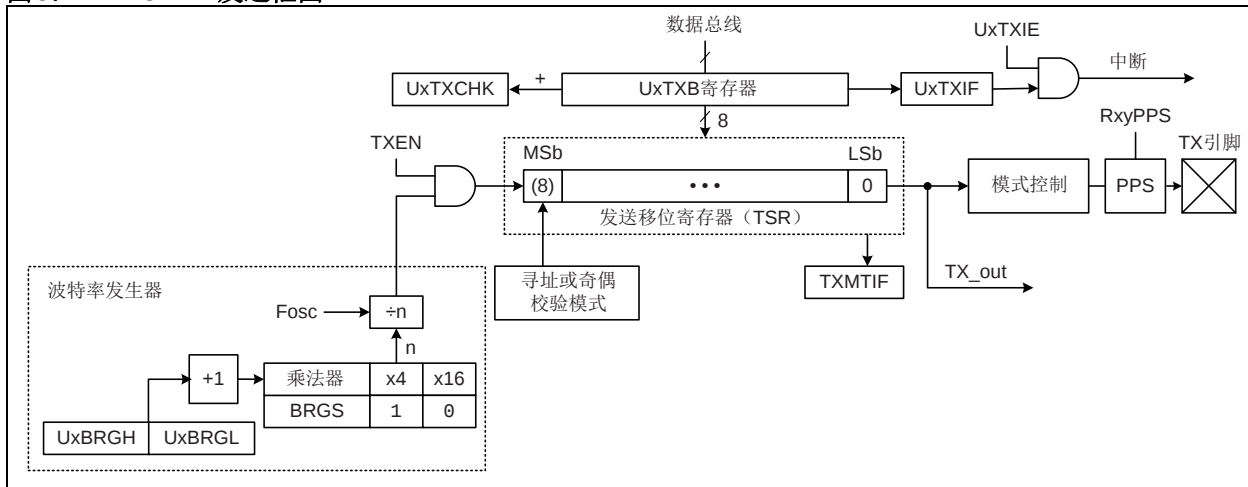
从器件在TX/CK线上接收时钟信号（见图1）。器件配置为同步从发送或接收操作时，自动禁止TX/CK引脚的输出驱动器。串行数据位在每个时钟的前沿改变，确保其在时钟的后沿有效。每个时钟周期传送一个数据位。数据位有多少，就产生多少个时钟周期。

异步发送

图3给出了PIC18FXXK42器件的UART模块框图。PIC18FXXK42是第一款采用具有协议支持的UART的8位PIC单片机系列。

注： 发送框图、寄存器和位名称可能因器件而异。有关器件特定的信息，请参见数据手册。

图3： UART发送框图



通过向发送缓冲区（UxTXB）写入一个字符来启动发送。如果这是第一个字符，或前一个字符被完全从TSR中送出，UxTXB中的数据就立即被传送到TSR。如果TSR仍包含前一个字符的全部或部分，则新字符数据保存在UxTXB中，直到前一个字符的发送完毕。然后，在前一个字符的停止位发送开始时，立即将UxTXB中待处理的字符传送到TSR。在前一个字符的全部停止位发送完毕后，立即开始启动位、数据位和停止位序列的发送。

将UxCON0寄存器的TXEN位置1可使能发送器。TSR以用户在UxBRGH:UxBRGL寄存器和UxCON0寄存器的BRGS位中定义的速率移出数据位。通过UxCON0寄存器的MODE<3:0>位选择UART模式（包括7位模式、8位模式和带地址或奇偶校验的8位模式）以及任一协议支持模式。

UART发送输出（TX_out）可通过TX引脚实现，而TX引脚则通过外设引脚选择（Peripheral Pin Select, PPS）模块的RxyPPs寄存器选择。它还可在软件中通过相应外设的寄存器配置为可配置逻辑单元（Configurable Logic Cell, CLC）等其他片上外设的输入。与其他外设的内部连接减少了外部接线，并释放了几个引脚（TX引脚和外设输入引脚）用于其他用途。

以下两个标志位可用于确定发送器的状态：UxTXIF和UxTXMTIF。只要UxTxB为空，无论UxTXIE发送中断允许位的状态如何，UxTXIF发送中断标志位均置1。只有在TSR忙于处理字符且UxTXB中还有排队等待发送的新字符时，该标志位才被清零。另一方面，TSR寄存器为空且空闲时，UxTXMTIF位置1，当有字符从UxTXB传送到TSR时，该位清零。这两个位的实现取决于数据写入模块的方式，即可以基于中断或基于轮询。

异步接收

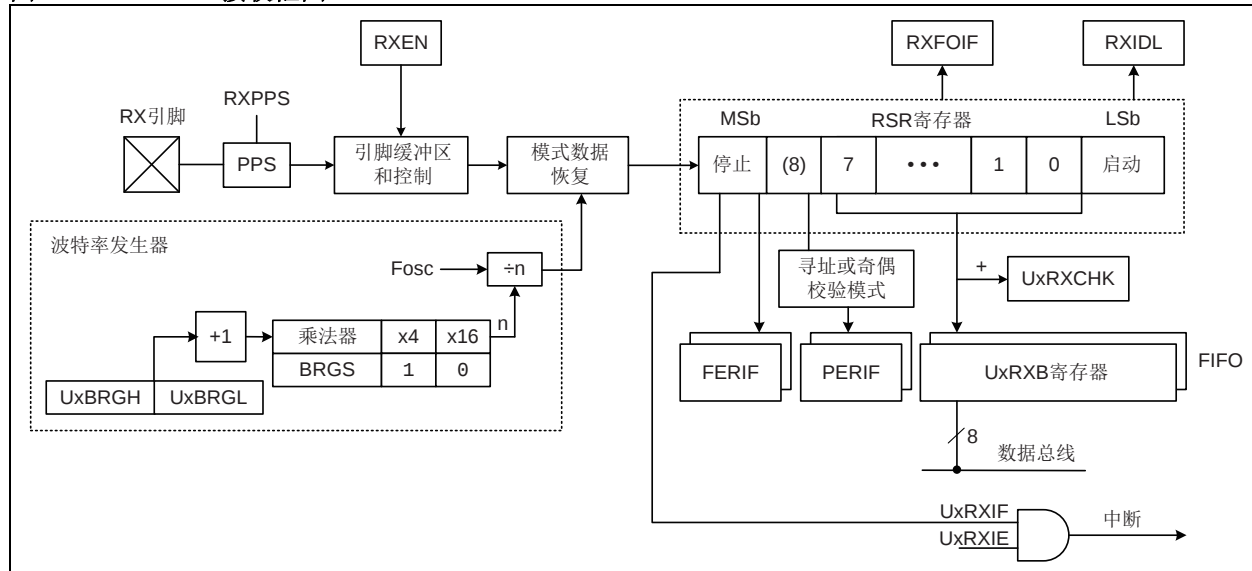
图4给出了PIC18FXXK42器件的UART接收器框图。

注： 接收框图、寄存器和位名称可能因器件而异。有关器件特定的信息，请参见数据手册。

通过PPS模块的RXPPS寄存器选择RX引脚。将UxCON0寄存器的RXEN位置1来使能UART的接收器电路。数据通过RX引脚接收并驱动数据恢复模块。数据恢复模

块是以4倍或16倍波特率运行的高速移位器，具体取决于UxCON0寄存器的BRGS位的配置。串行接收移位寄存器（Receive Shift Register, RSR）以配置的波特率移入字符位。字符的所有位移入后被立即传送到双字符的先进先出（First-In-First-Out, FIFO）存储区中。如果PIR1寄存器中的UxRXIF中断标志位未受到抑制，则将在此时置1。FIFO缓冲区允许先接收2个完整字符，再由软件将数据提供给UART接收器。FIFO寄存器和RSR不能直接用软件访问。可通过接收缓冲区（UxRXB）寄存器访问所接收的数据。

图4: UART接收框图



UxERRIR寄存器的帧错误中断标志位（FERIF）表示接收FIFO顶部未读取字符的帧状态。当FIFO顶部的字符不具有帧错误或接收FIFO中的所有字节都被读取时，该位清零。

当在任何奇偶校验模式下工作时，UxERRIR寄存器中的奇偶校验错误中断标志位PERIF均表示接收FIFO的顶部未读取字符的奇偶校验错误，而非奇偶校验位本身。当使能寻址模式时，它也可以表示为地址或数据指示符；对于DALI模式，则可表示前向帧或后向帧指示符。

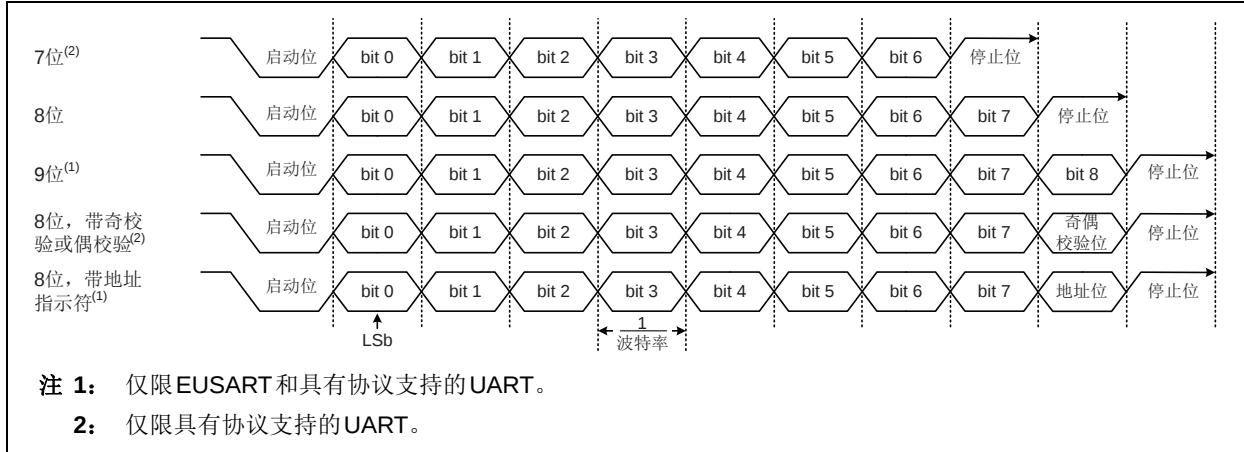
当接收到的字符超过接收FIFO可容纳的字符时，RXFOIF接收FIFO溢出中断标志位置1。引起溢出条件的字符始终会被丢弃。当RX引脚处于空闲状态时，RXIDL接收引脚空闲状态位置1，当UART接收启动、停止、数据、自动波特率或间隔字符时，RXIDL接收引脚空闲状态位清零。

异步模式

UART使用标准不归零（Non-Return-to-Zero, NRZ）格式发送和接收数据，这种格式有两种电平：VOH标记状态（1数据位）和VOL空格状态（0数据位）。在NRZ中，连续发送相同值的数据位时，将保持该位的输出电平不变，而不会在发送完每个位后到端口在标记状态空闲的这段时间内返回到中间电平。启动位始终处于空格状态，停止位始终处于标记状态。每个位的持续时间为1/(波特率)，波特率源自片上专用16位波特率发生器（见“波特率发生器”章节）。

UART可以不同的异步模式工作，如图5所示。

图5: UART异步模式



7位模式

在7位模式下，启动位和停止位之间仅传输数据的七个LSb。当正在交换的数据为7位ASCII字符时，通常会实现该模式。

8位模式

该模式由一个启动位、八个无奇偶校验的数据位和一个或多个停止位组成。PC串行通信中最常见的数据格式是具有8个数据位、无奇偶校验且具有1个停止位的8N1配置。

9位模式

EUSART模块支持使用由软件确定的第九个数据位的功能（即，奇偶校验或地址指示符）进行9位异步通信。另一方面，具有协议支持的UART允许使用第九个位由硬件进行奇偶校验或地址检测。

具有偶校验或奇校验的8位数据模式

具有协议支持的UART模块完全支持奇偶校验。奇偶校验是一种简单的错误检测方法，它使用附加位来表示1数据位的数量，并存在偶校验和奇校验两种形式。

在偶校验中，当1数据位的计数为奇数时，奇偶校验位设置为1，否则设置为0。在奇校验中，当1数据位的计数为偶数时，奇偶校验位设置为1，否则设置为0。

当选择偶校验或奇校验时，硬件将在传输过程中自动确定奇偶校验位并将其插入串行数据流中。UART接收器也可以执行奇偶校验。只要接收过程中检测到奇偶校验错误，错误标志便会置1。

具有地址指示符的8位数据模式

该模式将地址位实现为第九个数据位。有关详细信息，请参见“异步寻址模式”一节。

异步寻址模式

多个接收器共用同一传输线路的多分支应用（例如在RS-485系统中）需要通过寻址来避免通信总线上发生争用。提供两种类型的寻址模式，具体取决于所选器件的UART模块。

EUSART地址检测模式

在EUSART中，通过将ADDEN位置1来使能地址检测。使能该位时，只有第9个数据位置1的字符会被传送到接收FIFO缓冲区，所有其他字符都将被忽略。接收到地址字符后，用户软件判断此地址是否与其匹配。

具有协议支持的UART的寻址模式

具有协议支持的UART支持异步寻址模式，其中数据以9位字符的形式发送和接收，如图6所示。第9个位确定字符是地址还是数据。当第9个位置1时，8个LSb为地址。当第9个位清零时，8个LSb为数据。

图6: UART 寻址模式

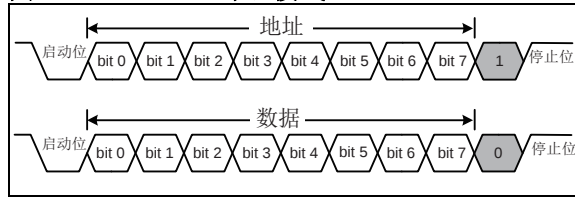
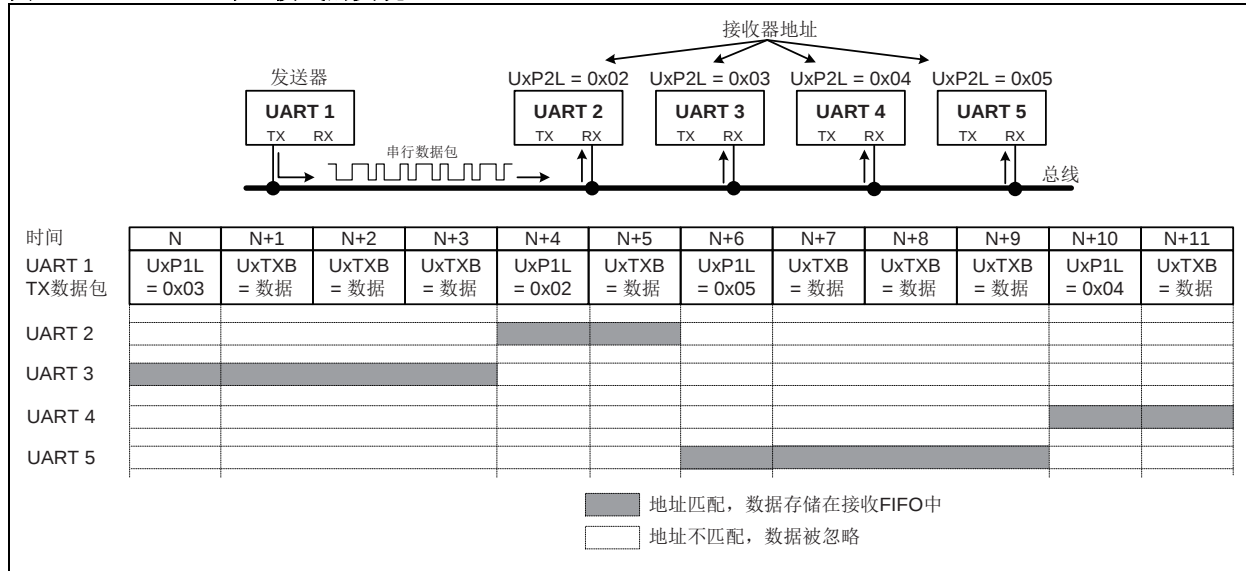


图7给出了使用PIC18F25K42的单发送器（主器件）多接收器（从器件）系统的示例。主器件启动所有地址和数据包的传输。

使用PIC18FXXK42 UART作为主器件时，字符写入非常简单：将地址写入UxP1L寄存器，将数据写入UxTXB发送缓冲区寄存器。在传输过程中，硬件会在第9个位自动追加1或0。

图7: UART 寻址模式的实现



从器件将接收器地址（UxP2L）与在第9个位置1时接收到的每个字符的8个LSB进行比较，并通过屏蔽寄存器验证结果，以此来执行地址匹配。匹配发生时，所有后续数据都存储在接收FIFO中，并可通过接收器缓冲区寄存器进行读取。否则，所有后续数据将被忽略，从器件将等待总线中的另一个地址。

寻址模式可用于任何多发送器多接收器系统。用户必须确保不会有两个发送器同时工作，以防止总线上发生争用。这可以通过使用软件适当调度任务来实现。

DMX模式

仅具有协议支持的UART支持DMX模式。DMX（即DMX512）是用于控制照明设备和配件的异步串行数字数据传输标准。它由一个可发出命令的控制台组成，这些命令由同一数据链路上连接的其他设备（例如，剧院灯调光器、烟雾机和闪光灯）接收。

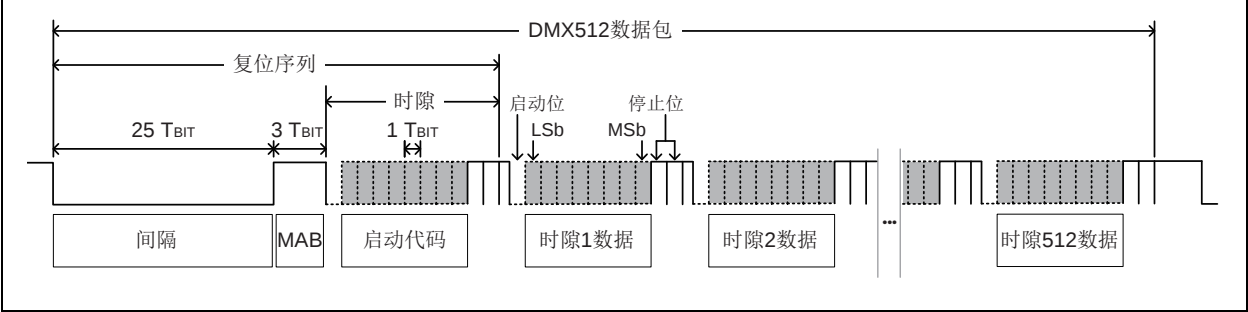
具有协议支持的UART硬件支持以下DMX特性：

- 自动生成间隔（Break）序列和间隔后接通（Make After Break）序列
- 间隔和间隔后接通序列检测
- 可配置内部数据字节计数器
- 用于接收器第一个和最后一个字节检测的专用寄存器

DMX512通常以250 Kbps的比特率工作。比特率使用器件数据手册中给出的公式计算。“波特率发生器”一节给出了PIC18FXXK42的比特率公式示例。

DMX512数据包由复位序列和随后最多512个数据字节的序列组成，如图8所示。

图8: DMX512数据包



DMX控制台发送器

UART 能够生成由间隔、“间隔后接通”（Make After Break, MAB）和启动代码组成的复位序列。使用模块时，间隔的宽度固定为25个位时间。之后是MAB，这是一段大约为3个位时间的空闲周期。启动代码指定数据包上后续数据字节的功能。

启动代码和数据字节写入发送缓冲区寄存器。软件需要跟踪写入发送缓冲区的字节数，以确保发送的字节数等于所需字节数，因为DMX状态机将在写入“n”个字节后自动插入间隔并复位其内部计数器。

每个数据时隙均以启动位开始，并以两个停止位终止。

DMX 设备接收器

DMX 设备的间隔字符阈值为23个位时间。短于该周期的间隔将被忽略，DMX 状态机保持空闲模式。接收到间隔后，DMX 计数器将复位以与输入数据流一致。UART 将接收到“间隔后接通”，并将启动代码存储在接收FIFO中。

在启动代码之后，将接收到第1个至第512个字节，但仅会将所需字节存储在接收FIFO中。UART通过几个专用的16位寄存器来识别何时启动和停止数据字节存储。

DMX 模式的实现

使用PIC16F25K42时，通过设置MODE<3:0> = 1010和STP<1:0> = 10，将UART设置为DMX模式。

图9: DMX模式的实现

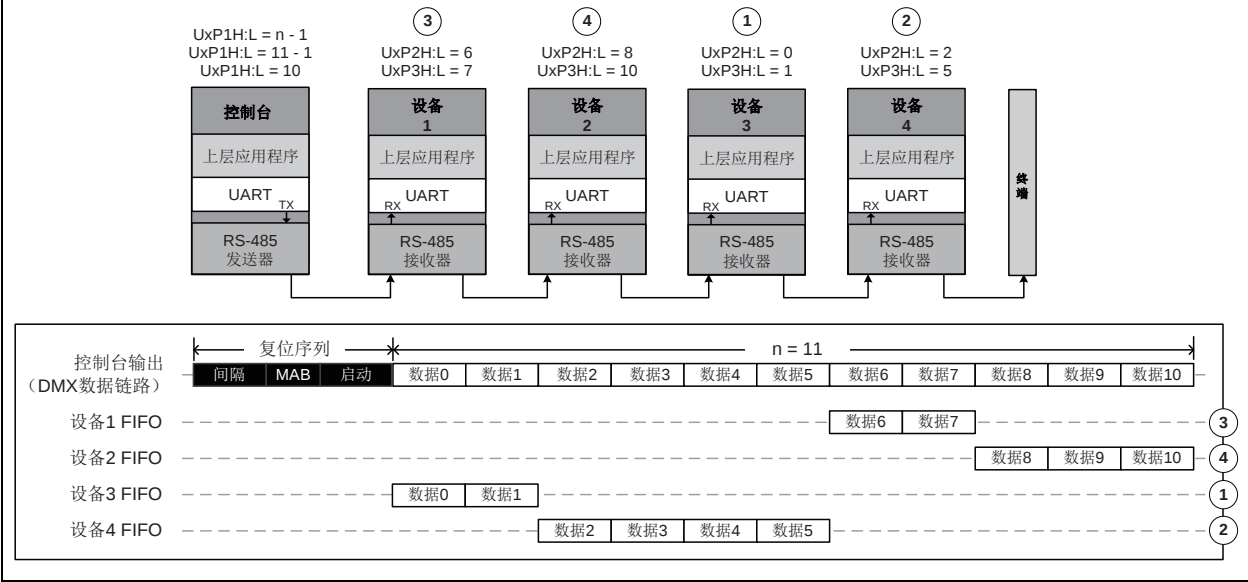


图9给出了DMX512通用示例。一个控制器控制台以菊花链配置方式连接四个单元负载或设备。通过控制台发送的数据字节数由UxP1H:L寄存器决定。然后，设备将分别通过UxP2H:L和UxP3H:L寄存器识别何时启动和停止在FIFO中存储接收到的数据。数据包将不断重新发送，以在数据发生变化时将数据更新到设备中。

图9中DMX控制台和DMX设备2的初始化代码分别如例1和例2所示。

例1: DMX控制台初始化代码

```
// Initialize DMX Console to transmit 11 data bytes
void DMX_Console_Initialize (void){
    U1CON0bits.MODE = 0b1010;      // DMX mode
    U1CON0bits.TXEN = 1;            // Enable transmit
    U1CON0bits.RXEN = 0;            // Disable receive
    U1CON2bits.TXPOL = 0;           // Data polarity is not inverted
    U1P1H = 0x00;                   // n-1 = 11-1
    U1P1L = 0x0A;
    U1BRGH = 0x00;                  // Initialize baud rate to 250k; Fosc = 64 MHz
    U1BRGL = 0x0F;
    U1CON2bits.STP = 0b10;          // 2 Stop bits

    Pin_Initialize();               // Set the corresponding TX pin via PPS

    U1CON1bits.ON = 1;              // Enable serial port
}
```

例2: DMX设备2的初始化代码

```
// Initialize DMX Equipment 2 to receive data 8 through 10 from the console packet
void DMX_Equipment2_Initialize (void){
    U1CON0bits.MODE = 0b1010;      // DMX mode
    U1CON0bits.TXEN = 0;            // Disable transmit
    U1CON0bits.RXEN = 1;            // Enable receive
    U1CON2bits.RXPOL = 0;           // Data polarity is not inverted
    U1P2H = 0x00;                   // Receive data 8
    U1P2L = 0x08;                   // :
    U1P3H = 0x00;                   // :
    U1P3L = 0x0A;                   // through 10
    U1BRGH = 0x00;                  // Initialize Baud Rate to 250k; Fosc = 64 MHz
    U1BRGL = 0x0F;
    U1CON2bits.STP = 0b10;          // 2 Stop bits

    Pin_Initialize();               // Set the corresponding RX pin via PPS

    U1CON1bits.ON = 1;              // Enable serial port
}
```

例3给出了DMX控制台的示例发送程序。将发送数据值存储在无符号8位类型的数组中可轻松对要发送的字节数进行计数，因为数组索引对应于字节编号。

例3: DMX控制台发送

```
void DMX_Console_Transmit (uint8_t txLength, uint8_t *txDataArray){
    UART_Write (START_CODE);

    for(uint8_t i = 0; i < txLength; i++){    // txLength = n; Max value = 512
        UART_Write(txDataArray[i]);
    }
}
```

例4中给出了一个示例程序，它使用一个变量指定要接收的字节数，并使用一个数组存储接收到的数据以供在用户应用程序代码中使用。

例4: DMX设备接收

```
void DMX_Equipment_Receive (uint16_t rxLength){ // rxLength = number of bytes to
                                                    // receive

    uint8_t startCode;
    uint8_t rxDataArray[arraySize];              // arraySize value should be at least
                                                    // equal to rxLength

    while (!U1ERRIRbits.RXBKIF);                 // Wait for break reception
    U1ERRIRbits.RXBKIF = 0;

    startCode = UART_Read();                      // Save Start code
    for (uint8_t i = 0; i < rxLength; i++){
        rxDataArray[i] = UART_Read();            // Save Data
    }

    /**Some code for Start code and received data processing**/
}
```

LIN 模式

局域互联网（LIN）是一种主要用于汽车和工业应用的协议。

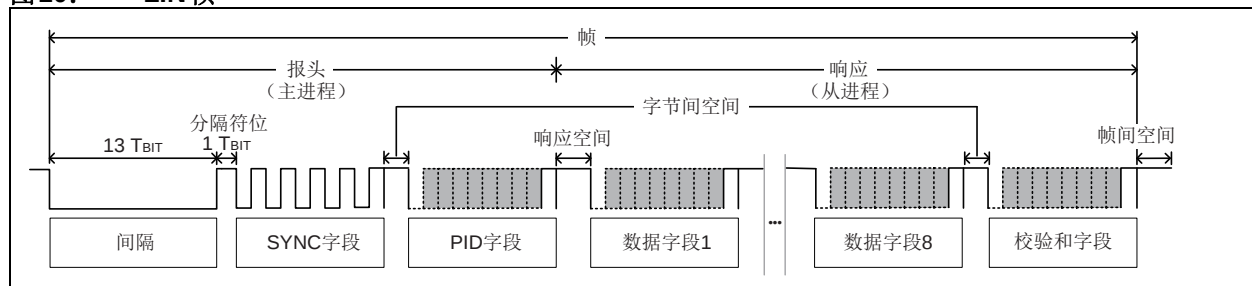
注： 仅具有协议支持的 UART 支持不同的 LIN 模式，但 EUSART 也有一些针对 LIN 协议的支持特性（见“[EUSART LIN 支持](#)”一节特性部分）。

具有协议支持的 UART 模块为 LIN 提供以下硬件支持：

- 自动计算校验和
- 在传统校验和与增强型校验和之间选择
- 自动计算 PID 奇偶校验位
- 在串行流中自动插入间隔、定界符位、同步和 PID
- 在 RX 引脚上进行间隔检测
- 为从节点进行自动波特率检测

LIN 帧由主进程和从进程组成（见图 10）。具有协议支持的 UART 为 LIN 提供两种模式：LIN 主/从模式和 LIN 从模式。主/从模式支持主进程和从进程，因此在主节点中实现。而从模式仅支持从进程，在从节点中实现。

图 10： LIN 帧



例 5 给出了使用 UART1 的 LIN 主/从模式的初始化代码示例。LIN 从模式遵循相同的初始化模式，但 MODE<3:0> 位的配置除外，这些位必须设置为 0b1011。

例5: LIN主/从模式初始化代码

```
void LIN_MasterSlaveMode_Initialize (void){
    U1CON0bits.BRGs = 0;           // BRG operates in normal speed; Baud = Fosc/[16(n+1)]
    U1BRGH = 0x01;                 // Initialize Baud Rate to 9600; Fosc = 64 MHz
    U1BRGL = 0x9F;
    U1CON0bits.MODE = 0b1100;     // LIN Master/Slave mode
    U1CON0bits.TXEN = 1;           // Enable transmit
    U1CON0bits.RXEN = 1;           // Enable receive
    U1CON2bits.TXPOL = 0;          // Data polarity is not inverted
    U1CON2bits.STP = 0b00;         // 1 Stop bit

    Pin_Initialize();              // Set the corresponding TX and RX pins via PPS

    U1CON2bits.COEN = 1;           // Enhanced checksum; Clear to '0' for legacy checksum
    U1CON1bits.ON = 1;             // Enable serial port
}
```

主进程

UART产生间隔字段作为13个位周期宽的空格，产生间隔定界符作为1个位时间长的标记。它为同步、PID和数据字段实现了8个数据位异步格式。停止位的数量可在1、1.5和2个位周期之间选择。同步字节为一个“U”（55h）字符，而PID字节由6个帧标识符位和2个由硬件自动计算的奇偶校验位组成。将6位ID写入专用寄存器后，将自动发送间隔、定界符位和同步信号。

例6给出了主进程的代码示例。

例6: 主进程（发送报头）

```
void LIN_MasterTransmitHeader (uint8_t slave_ID){
    U1P1L = slave_ID;
}
```

从进程

PID确定哪些从进程将响应主器件。从进程在其作为发布方时负责发送帧响应，在其作为订阅方时负责接收帧响应。它由数据字段和校验和字段组成。数据字段是在从进程中发送或接收的实际信息，而校验和字段用于错误检测。例7给出了如何在从进程中获取PID的示例代码，而例8和例9则给出了如何使用数组发送和接收数据字段。

例7: 在从进程中获取PID

```
uint8_t LIN_Get_PID (void) {
    uint8_t PID;

    while (!U1ERRIRbits.RXBKIF); // Wait for break reception
    U1ERRIRbits.RXBKIF = 0;

    PID = UART_Read();

    return PID;
}
```

例8: 从进程中的数据字段发送

```
void LIN_SlaveDataTransmit (uint8_t txLength, uint8_t *txDataArray){
    U1P2H = 0x00;
    U1P2L = txLength; // Size of the data array; Max value = 8

    for(uint8_t i = 0; i < txLength; i++){
        UART_Write(txDataArray[i]);
    }
}
```

例9: 从进程中的数据字段接收

```
void LIN_SlaveDataReceive (uint8_t rxLength){ // rxLength = Number of data bytes
                                              // to receive

    uint8_t rxDataArray[8];

    U1P3H = 0x00;
    U1P3L = rxLength;

    for (uint8_t i = 0; i < rxLength; i++) {
        rxDataArray[i] = UART_Read(); // Save data
    }

    /** Some code for checksum verification and processing of received data */
}
```

有两种方法可用于计算校验和：传统方法和增强型方法。传统校验和仅包括数据字节，而增强型校验和包括PID和数据字节。

当从进程发送数据时，UART会在字节发送后自动计算这些字节的校验和，并将取反后的校验和字节附加到从响应中。当从器件接收数据时，将在接收到每个字节时对相应字节的校验和进行累加，校验和的算法与发送过

程相同。最后一个字节（即通过发送过程计算的校验和取反后的结果）将与通过UART本地计算的校验和相加。当结果全部为1时，校验通过，否则校验失败，此时校验和错误标志位置1。[例10](#)给出了如何验证校验和是通过还是失败。

例10: 校验和验证

```
bool LIN_VerifyRxChecksum (void){
    while(!PIR3bits.U1RXIF);

    if(U1ERRIRbits.CERIF == 0)
        return true; // Checksum passed
    }
    return false;
}
```

校验和方法、数据字节数以及是否在从进程中发送或接收数据均由软件根据PID定义。字节间周期、超时和帧周期由软件使用UART以外的方式定时。片上硬件定时器（如Timer0/1/2）可用于此目的。

EUSART LIN支持

增强型中档和高端PIC MCU的EUSART模块支持固定的13位间隔传输和接收到间隔字符时唤醒检测。其他任务可由用户使用软件确定。Microchip的MPLAB®代码配置器（MPLAB® Code Configurator, MCC）LIN模块允许用户创建调度表，并通过易于使用的GUI选择校验和类型和硬件定时器。它支持大多数具有EUSART的器件，并能够生成快速原型设计所需的软件程序。有关更多信息，请参见AN2059《LIN基础知识和8位PIC®单片机上实现的MCC LIN协议栈库》。

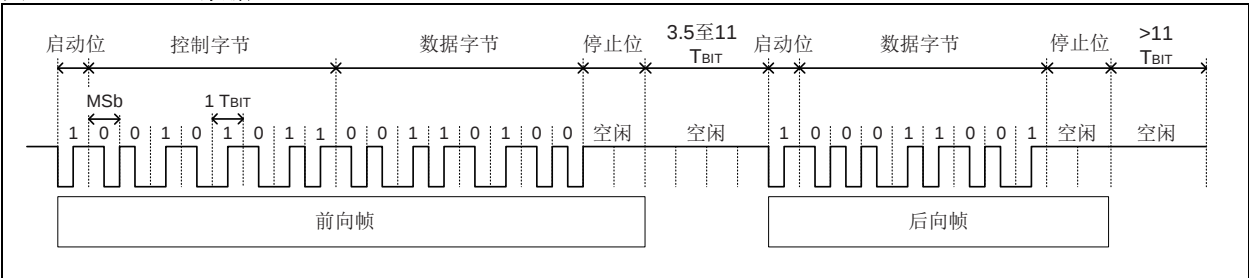
DALI模式

数字可寻址照明接口（DALI）是大型建筑物（如办公室和工厂）中的数字照明控制所使用的协议。仅具有协议支持的UART模块才提供DALI支持。它针对DALI提供以下特性：

- 可在控制器件与控制装置模式之间选择
- 曼彻斯特编码
- 最高有效位优先
- 背靠背字节之间无停止周期
- 2个停止位的传输和检测
- 前后等待周期可采用半位步长配置
- 前向/后向帧检测阈值定界符可采用半位步长配置
- 在等待时间结束时自动传输后向帧

提供两种模式：DALI控制器件模式和DALI控制装置模式。控制器件是向控制装置（照明装置）发出命令和数据的主控制器。图11给出了总线上的DALI数据包的示例。所有字节均以曼彻斯特编码格式采用MSb优先的方式进行发送和接收。

图11: DALI数据包



DALI控制器件模式

选择控制器件模式来发送前向帧。前向帧是指从控制器件发送到控制装置的数据包。它由2个字节（对于DALI 1.0）或3个字节（对于DALI 2.0）组成。一旦发送中断标志变为真，并且输出移位器尚未变为空，即可通过写入发送缓冲区实现无中断的后向字节发送。在最后一个停止位之后，TX输出将在由专用寄存器确定的额外等待时间内保持空闲状态。例10给出了DALI控制器件的初始化代码示例。

例 11: DALI 控制器件初始化代码

```

void DALI_ControlDevice_Initialize (void){
    U1CON0bits.MODE = 0b1000;    // DALI Control Device Mode
    U1CON0bits.TXEN = 1;          // Enable transmit
    U1CON0bits.RXEN = 1;          // Enable receive
    U1P1H = 0x00;                 // 16 half-bit periods wait time after complete...
    U1P1L = 0x16;                 // ...transmission of forward frame
    U1P2H = 0x00;                 // Frames following Idle periods more than 15 half-bit...
    U1P2L = 0x15;                 // ...periods are detected as Forward frames
    U1BRGH = 0x0D;
    U1BRGL = 0x04;                // Baud = 1200; Fosc = 64 MHz
    U1CON2bits.TXPOL = 0;         // Data polarity is not inverted; Set to '1' if inverted
                                   // polarity is required
    U1CON2bits.STP = 0b10;        // 2 Stop bits

    Pin_Initialize();             // Set the corresponding TX pin via PPS

    U1CON1bits.ON = 1;            // Enable serial port
}

```

在写入下一个前向帧之前，软件等待输出移位器变为空。如果在等待时间内接收到后向帧，则在后向帧加上设置的等待时间之前，可能已经写入发送缓冲区寄存器的任何字节将不会进行发送。

DALI控制装置模式

选择控制装置模式来接收前向帧和发送后向帧。后向帧是指从控制装置发回控制器件的响应数据包。DALI要求先接收前向帧，再发送后向帧。只有在超过设定阈值的空闲周期后的帧才会被检测为前向帧。例11给出了DALI控制装置的初始化代码示例。

例12: DALI控制装置初始化代码

```
void DALI_ControlGear_Initialize (void){
    U1CON0bits.MODE = 0b1001;        // DALI Control Gear Mode
    U1CON0bits.TXEN = 1;              // Enable transmit
    U1CON0bits.RXEN = 1;              // Enable receive
    U1P1H = 0x00;                     // 8 half-bit periods wait time after complete...
    U1P1L = 0x08;                     // ...reception of a forward frame
    U1P2H = 0x00;                     // Frames following Idle periods more than 15 half-
    U1P2L = 0x15;                     // bit periods are detected as Forward frames
    U1BRGH = 0x0D;
    U1BRGL = 0x04;                    // Baud = 1200; Fosc = 64 MHz
    U1CON2bits.TXPOL = 0;              // Data polarity is not inverted; Set to '1' if
                                      // inverted polarity is required
    U1CON2bits.RXPOL = 0;              // Same as TXPOL
    U1CON2bits.STP = 0b10;            // 2 Stop bits

    Pin_Initialize();                 // Set the corresponding TX pin via PPS

    U1CON1bits.ON = 1;                // Enable serial port
}
```

前向帧可在接收FIFO中一次接收一个字节，并通过读取接收缓冲区寄存器来获取。通过控制装置从前向帧接收的数据由应用程序软件处理。它还决定是否应发送后向帧。如果在等待时间到期后写入后向帧数据，则在经过下一个前向帧后的等待时间前，数据将一直保存。

通用曼彻斯特

通用曼彻斯特是DALI模式的一个子集。它用于维护DALI模式的各个方面。惟一区别是等待时间设置为0。这可以实现全双工和半双工操作，因为写入不会暂停来等待接收操作完成。

波特率发生器

波特率发生器（Baud Rate Generator, BRG）是一个8位或16位定时器，专用于支持同步和异步工作模式。

UART模块可在正常和高波特率范围之间进行选择。高波特率范围用于在无法达到所需波特率时扩展波特率范围。使用任一范围均可实现所需波特率时，建议使用正常波特率范围。

波特率的计算因UART模块的类型和配置的速度而异。波特率公式通常在器件数据手册中给出。表2给出了PIC18FXXK42的波特率公式。

表2： PIC18FXXK42波特率公式

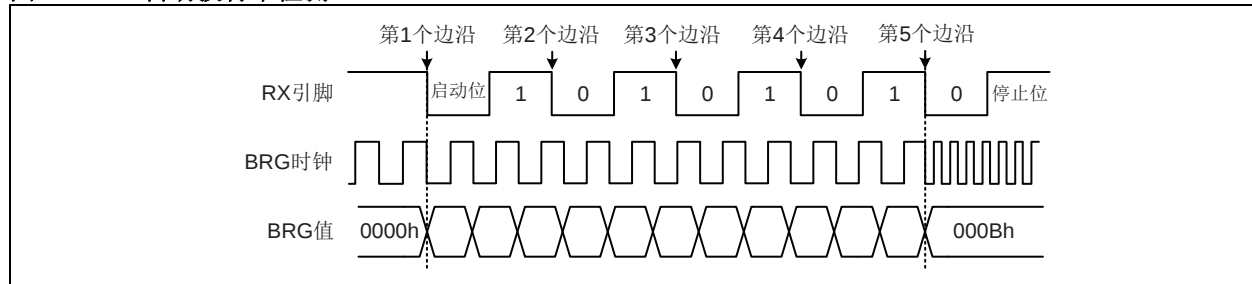
BRG/UART 模式	波特率
高波特率	$F_{osc}/[4(n+1)]$
正常波特率	$F_{osc}/[16(n+1)]$

计算得到的波特率存储在一对8位寄存器中。

自动波特率检测

当自动波特率检测（Auto-Baud Detect, ABD）有效时，提供给BRG的时钟是反向的。不是由BRG为传入的RX信号提供时钟，而是由RX信号为BRG定时。波特率发生器用来测量接收55h字符（ASCII“U”，也是LIN总线的同步字符）的周期。该字符的惟一特性是它具有5个下降沿（包括启动位边沿在内）和5个上升沿（包括停止位边沿在内）。请参见图12。

图12： 自动波特率检测



8位异步和LIN模式支持ABD。

在8位异步模式下，通过将ABDEN位置1来使能自动波特率。波特率发生器使用BRG计数器时钟在接收信号的第一个下降沿开始计数。在第5个下降沿，会将相应BRG周期内累计所得的值保存到BRG寄存器对中，ABDEN位自动清零，ABD中断标志位置1。

当选择LIN模式时，硬件在接收到同步字符时自动执行ABD，并且既不必要也不可能将ABDEN位置1。

停止位选择

具有协议支持的UART可编程停止位的数量，包括：

- 发送1个停止位，对第1个进行接收验证
- 发送1.5个停止位，对第1个进行接收验证
- 发送2个停止位，对全部2个进行接收验证
- 发送2个停止位，仅对第1个进行接收验证

在除DALI以外的所有模式中，发送器在连续发送的各个字之间处于空闲状态，持续时间为停止位周期数。在DALI中，停止位在所发送数据流中的最后一位之后生成。

自动校验和计算

具有协议支持的UART具有发送和接收校验和加法器。使能时，加法器会累加发送或接收的每个字节。累加的和包括加法的进位，并存储在校验和寄存器中。

软件负责在事务之前清零校验和寄存器，并在事务结束时执行校验。在LIN模式下，始终使能自动校验和计算，并且硬件自动清零和校验累加和。

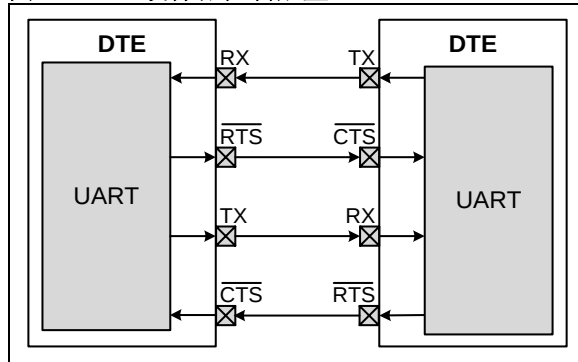
流控制

具有协议支持的UART提供硬件和软件流控制，可通过握手流控制位选择。当器件准备好发送或接收数据时，流控制可通过发出信号来降低数据丢失的风险并防止缓冲区溢出。

硬件流控制

除了TX和RX引脚外，硬件流控制还会用到单片机的几个引脚。它们的RS-232信号名称为RTS（请求发送）和CTS（允许发送）。图13给出了两个数据终端设备（Data Terminal Equipment, DTE）之间的典型配置。每个DTE使用RTS输出是否准备好接受新数据，并读取CTS来了解是否允许将数据发送到其他器件。两条线均为低电平，可通过PPS模块进行配置。

图13： 硬件流控制配置



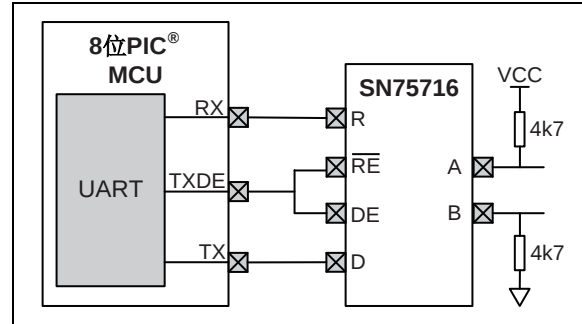
当输入FIFO为空时，接收数据的UART会将RTS输出置为低电平。当接收到一个字符时，RTS输出变为高电平，直到读取接收缓冲区以释放两个FIFO单元。

仅当检测到CTS输入为低电平时，发送数据的UART才能发送。如果CTS输入在字节开始发送后变为高电平，发送将正常完成。即使CTS输入为高电平，接收UART也可通过接受第二个FIFO单元中的字符来实现此目的。

RS-485收发器控制

单片机的另一个引脚可用于控制RS-485收发器。该引脚的信号名称为TXDE，用于发送驱动器使能。该输出在TX输出主动发送一个字符时为高电平，在所有其他时间为低电平。硬件流控制可用于控制RS-485收发器的方向。图14给出了示例配置。通过PPS模块选择TXDE引脚。

图14： RS-485配置



XON/XOFF 软件流控制

XON/XOFF使用特殊字符来暂停和恢复发送。它需要全双工操作，因为发送器必须能够接收信号以在进行发送时暂停发送。UART将13h用于XOFF，将11h用于XON。

发送器默认为XON，即发送器使能。当接收到XOFF字符时，发送器在完成主动发送的字符后停止发送。发送器将保持禁止状态，直到接收到XON字符。

片上软件流控制可实现可靠的数据交换，同时能够降低在硬件控制中使用额外引脚的成本。但缺点是，由于还会在串行流中插入XON和XOFF字符，因此会引起延时。

与其他外设配合时的操作

本节提供了UART与其他片上外设配合时的一些实现示例。

使用HLT时的RX/TX活动超时

偶数个定时器模块的硬件限制定时器（Hardware Limit Timer，HLT）可用于监视UART TX和RX线的活动。TX或RX线可配置为HLT的复位信号源。HLT应配置为使UART活动持续复位HLT，以防止整个HLT周期结束。当选定TX或RX线上无活动的时间超过HLT周期时，会发生HLT中断，并发出超时事件信号。

使用DMA时的UART数据传输

PIC18FXXK42等器件具有片上直接存储器访问（Direct Memory Access，DMA）控制器。DMA允许UART直接将数据从接收缓冲区寄存器传输到RAM或从RAM传输到发送缓冲区寄存器，无需CPU干预。UART RX和TX也可配置为触发对DMA的中断请求，以启动新的DMA报文。由于UART数据传输期间可使用DMA接管存储器总线，以便CPU能够执行其他重要任务，从而可更高效地利用CPU并提高数据吞吐量。

使用UART和CLC时的数字调制

一些UART模块支持将UART发送输出内部连接到可配置逻辑单元（CLC）模块。将CLC和UART配合使用的一个重要实现示例为数据信号调制器（Data Signal Modulator，DSM）。DSM支持如开关键控（On-Off Keying，OOK）、频移键控（Frequency Shift Keying，FSK）和相移键控（Phase Shift Keying，PSK）等数字调制技术。CLC使用UART作为输入数据源，使用其他信号源（例如，系统时钟、LPRC和SOSC等）作为载波信号。有关详细讨论，请参见AN2133 “*Extending PIC[®] MCU Capabilities Using CLC*”。

结论

本技术简介涵盖了8位PIC单片机上现有三种UART模块的不同功能和特性，重点介绍了具有协议支持的UART模块，旨在帮助用户快速了解USART和EUSART不支持的不同高级特性。设计人员可使用该模块在更短的时间内以更少的软件开销开发基于UART的应用程序。

参考资料

- 《PIC18FXXK42数据手册》
- ANSI E1.11-2008（R2013）标准
- AN1659, “DMX512A”（DS00001659）
- AN2059, 《LIN基础知识和8位PIC[®]单片机上实现的MCC LIN协议栈库》
- AN1465, 《数字可寻址照明接口（DALI）通信》（DS01465A_CN）
- AN2133, “*Extending PIC[®] MCU Capabilities Using CLC*”

注:

请注意以下有关 Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下，Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案 (Digital Millennium Copyright Act)》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分，因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为您提供便利，它们可能由更新之信息所替代。确保应用符合技术规范，是您自身应负的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对其使用情况、质量、性能、适销性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任，并加以赔偿。除非另外声明，在 Microchip 知识产权保护下，不得暗中以其他方式转让任何许可证。

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。Microchip 的 PIC® MCU 与 dsPIC® DSC、KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品严格遵守公司的质量体系流程。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949 ==

商标

Microchip 的名称和徽标组合、Microchip 徽标、AnyRate、AVR、AVR 徽标、AVR Freaks、BeaconThings、BitCloud、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、Heldo、JukeBlox、KEELOQ、KEELOQ 徽标、Kleer、LANCheck、LINK MD、maXStylus、maXTouch、MediaLB、megaAVR、MOST、MOST 徽标、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32 徽标、Prochip Designer、QTouch、RightTouch、SAM-BA、SpyNIC、SST、SST 徽标、SuperFlash、tinyAVR、UNI/O 及 XMEGA 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

ClockWorks、The Embedded Control Solutions Company、EtherSynch、Hyper Speed Control、HyperLight Load、IntelliMOS、mTouch、Precision Edge 和 Quiet-Wire 均为 Microchip Technology Inc. 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BodyCom、chipKIT、chipKIT 徽标、CodeGuard、CryptoAuthentication、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet 徽标、Mindi、MiWi、motorBench、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PureSilicon、QMatrix、RightTouch 徽标、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 为 Microchip Technology Inc. 在美国的服务标记。

Silicon Storage Technology 为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

GestIC 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. & KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2017, Microchip Technology Inc. 版权所有。

ISBN: 978-1-5224-1797-2

全球销售及服务网点

美洲

公司总部 **Corporate Office**
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277

技术支持:
<http://www.microchip.com/support>

网址: www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

奥斯汀 Austin, TX
Tel: 1-512-257-3370

波士顿 Boston
Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 Chicago
Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

达拉斯 Dallas
Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 Detroit
Novi, MI
Tel: 1-248-848-4000

休斯敦 Houston, TX
Tel: 1-281-894-5983

印第安纳波利斯 Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453
Tel: 1-317-536-2380

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608
Tel: 1-951-273-7800

罗利 Raleigh, NC
Tel: 1-919-844-7510

纽约 New York, NY
Tel: 1-631-435-6000

圣何塞 San Jose, CA
Tel: 1-408-735-9110
Tel: 1-408-436-4270

加拿大多伦多 Toronto
Tel: 1-905-695-1980
Fax: 1-905-695-2078

亚太地区

亚太总部 **Asia Pacific Office**
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

中国 - 北京
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

中国 - 成都
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 - 重庆
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 - 东莞
Tel: 86-769-8702-9880
中国 - 广州
Tel: 86-20-8755-8029

中国 - 杭州
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

中国 - 南京
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 - 青岛
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 - 上海
Tel: 86-21-3326-8000
Fax: 86-21-3326-8021

中国 - 沈阳
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 - 深圳
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

中国 - 武汉
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 - 西安
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 - 厦门
Tel: 86-592-238-8138
Fax: 86-592-238-8130

中国 - 香港特别行政区
Tel: 852-2943-5100
Fax: 852-2401-3431

亚太地区

中国 - 珠海
Tel: 86-756-321-0040
Fax: 86-756-321-0049

台湾地区 - 高雄
Tel: 886-7-213-7830

台湾地区 - 台北
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

台湾地区 - 新竹
Tel: 886-3-5778-366
Fax: 886-3-5770-955

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

印度 India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

印度 India - Pune
Tel: 91-20-3019-1500
日本 Japan - Osaka
Tel: 81-6-6152-7160

Fax: 81-6-6152-9310

日本 Japan - Tokyo
Tel: 81-3-6880-3770
Fax: 81-3-6880-3771

韩国 Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

韩国 Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

马来西亚 Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

菲律宾 Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

新加坡 Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

泰国 Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

欧洲

奥地利 Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

芬兰 Finland - Espoo
Tel: 358-9-4520-820

法国 France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

法国 France - Saint Cloud
Tel: 33-1-30-60-70-00

德国 Germany - Garching
Tel: 49-8931-9700

德国 Germany - Haan
Tel: 49-2129-3766400

德国 Germany - Heilbronn
Tel: 49-7131-67-3636

德国 Germany - Karlsruhe
Tel: 49-721-625370

德国 Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

德国 Germany - Rosenheim
Tel: 49-8031-354-560

以色列 Israel - Ra'anana
Tel: 972-9-744-7705

意大利 Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

意大利 Italy - Padova
Tel: 39-049-7625286

荷兰 Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

挪威 Norway - Trondheim
Tel: 47-7289-7561

波兰 Poland - Warsaw
Tel: 48-22-3325737

罗马尼亚 Romania - Bucharest
Tel: 40-21-407-87-50

西班牙 Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

瑞典 Sweden - Gothenberg
Tel: 46-31-704-60-40

瑞典 Sweden - Stockholm
Tel: 46-8-5090-4654

英国 UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820