



UART 协议分析

——如何分析串口/RS232/RS485 等信号

1. 协议基础.....	3
1.1. 协议简介	3
1.2. 不同的物理层实现	4
1.2.1. 基于 TTL 的 UART 通讯	5
1.2.2. 基于 RS232 的 UART 通讯.....	6
1.2.3. 基于 RS485 的 UART 通讯.....	6
2. 协议规范.....	8
2.1. UART 帧结构.....	8
2.1.1. 波特率.....	8
2.1.2. 空闲位.....	9
2.1.3. 起始位.....	9
2.1.4. 数据位.....	10
2.1.5. 校验位.....	10
2.1.6. 停止位.....	10
3. 使用逻辑分析仪分析 UART 通讯	11
3.1. 示波器 vs 逻辑分析仪	11
3.2. 建立逻辑分析仪的测试环境	12
3.2.1. DSLogic 的连接	12
3.2.2. UART 信号连接.....	13
3.2.3. 阈值设置.....	15
3.3. 简单的 UART 信号采集.....	16
3.3.1. 信号采集.....	16
3.3.2. UART 解码.....	18
3.4. 单次 UART 传输的采集.....	20
3.4.1. Buffer 模式.....	21
3.4.2. 触发设置.....	21
3.4.3. 重复采集.....	24
3.5. 长时间 UART 传输的采集.....	25

3.5.1.	Stream 模式	25
3.5.2.	协议列表显示	26
3.5.3.	解码数据搜索	27
3.5.4.	波形搜索	28
3.5.5.	光标运用	29
3.5.6.	鼠标测量	30
4.	常见问题	31
4.1.	确定未知的波特率	31
4.2.	反相信号的解码	32
4.3.	多组 UART 通讯的解码	33
4.4.	起始位置导致的解码错误	33
5.	进阶分析	35
5.1.	多层协议解析	35

1. 协议基础

1.1. 协议简介

UART 是“Universal Asynchronous Receiver/Transmitter”通用异步收发器的缩写。在 19 世纪 60 年代，为了解决计算机和电传打字机通信，Bell 发明了 UART 协议，将并行输入信号转换成串行输出信号。因为 UART 简单实用的特性，其已经成为一种使用非常广泛的通讯协议。我们日常接触到的串口，RS232，RS485 等总线，内部使用的基本都是 UART 协议。

为了更好的理解和分析协议与总线的关系，我们通常把一个完整的通讯规范划分成物理层，协议层以及应用层。物理层只定义真实的信号特性（比如电压，电流，驱动能力等），以及电信号与逻辑信号 0 和 1 的对应关系；协议层不关心底层的 0 和 1 具体怎么实现，只规定逻辑信号的协议规范以及通讯过程（例如起始，数据以及结束等）；应用层不关心数据是怎么获取的，只定义数据表示的意义，以及如何实现具体的业务逻辑。

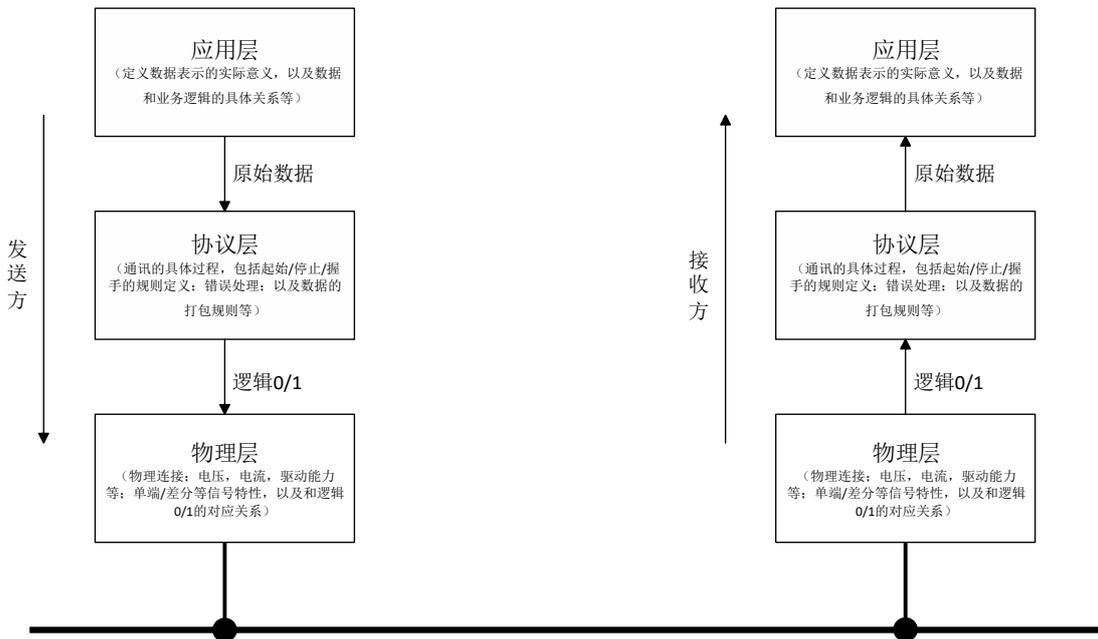


图 1-1 通讯协议的分层实现

最简单的 UART 协议应用，通常物理层只需要两根传输线，一根用于发送，一根用于接收，从而实现全双工通讯。对于单向传输，也可以只使用一根传输线。此类应用最典型的实例就是单片机的 RX/TX 端口互相连接，从而实现基于 TTL 电平的 UART 通讯。对于不同的传输距离以及可靠性的要求，替换不同的物理层实现既可以得到我们常见的 RS232、RS485 等通讯总线。

1.2. 不同的物理层实现

由于 UART 协议层的输入是逻辑 0/1 信号，而逻辑 0/1 信号在物理层可以通过不同的电平标准来区分。针对不同的通讯需求，便可以使用不同的物理层实现。例如简单的板内通讯，或者常见的设备调试场景，使用简单的 LVTTTL/TTL 电平即可在两个设备间进行 UART 协议通讯。

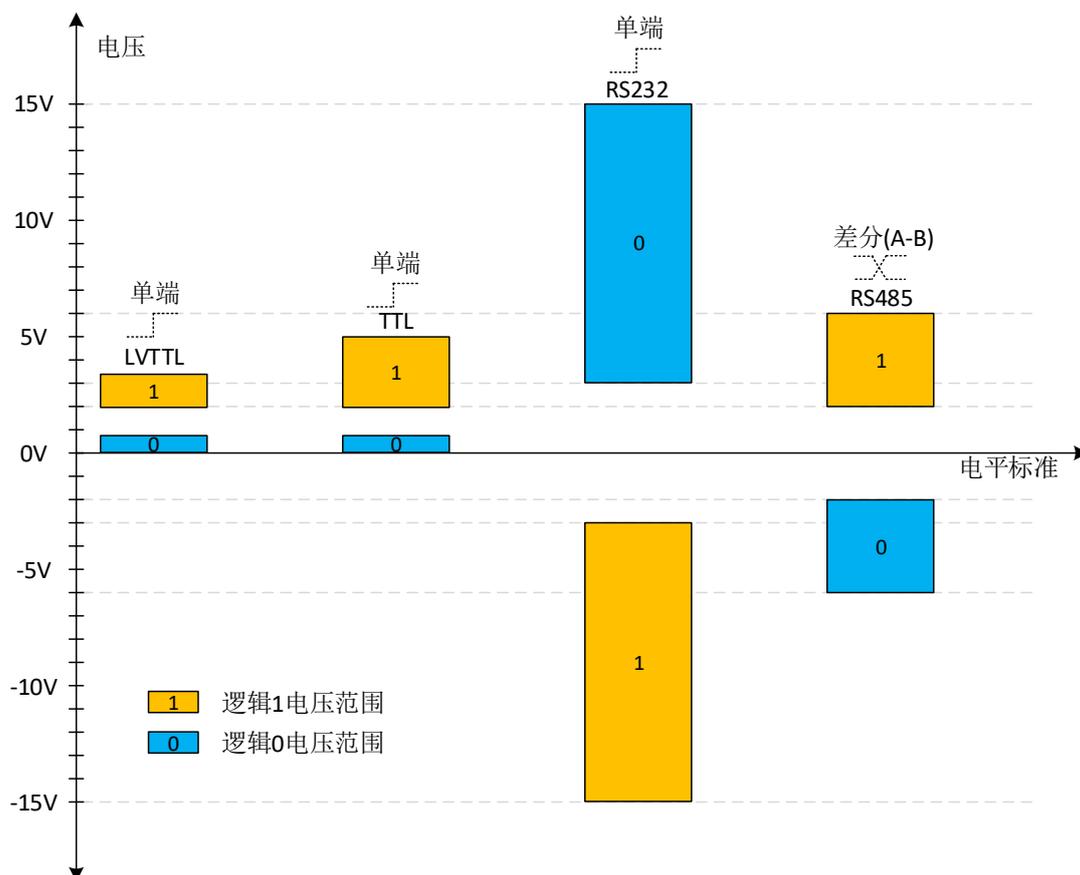


图 1-2 不同的物理层电平标准

通用的串口则使用的是 RS232 电平，可以增加传输距离，并且抵抗一定程度的信号干扰。付出的成本则是在物理层需要对应的电平转换芯片来实现，发送端需要将内部的高低电平信号转换成电压更高的+/-电压信号，接收端需要将+/-电压信号转换成内部的高低电平信号。

在工业通讯的场景下，为了进一步提高传输距离，以及增强信号的可靠性，一般会采用 RS485 的电平标准。在发送端将普通的高低电平信号转换成一对差分信号，在接收端将差分信号再转换成普通的高低电平信号。另外，RS485 允许总线上连接多达 128 收发器，而 TTL 或者 RS232 则是点对点的连接。

1.2.1. 基于 TTL 的 UART 通讯

基于 TTL 的 UART 通讯，是 UART 协议应用最简单的使用场景。即直接把数字 I/O 输出的高低电平作为实际的物理信号进行传输。在物理连接上，只需要设备共地，通过一根信号线即可完成单向的设备通讯。如果需要双向全双工，使用两根信号线即可。

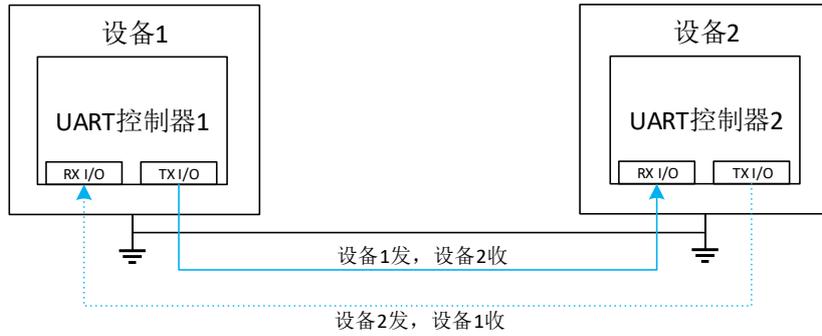


图 1-3 基于 TTL 的 UART 通讯

为了对比不同物理层实现的差别，我们可以观察发送相同数据时，不同物理层的实际信号有何不同。这里以发送字符‘D’为例，通过璞石示波器，直接观察 TTL 实现传输的信号（探头接地端连接设备共地端，探头信号端连接上图蓝色信号线），可以获得如图 1-4 所示的信号波形。从波形可以看出，当没有数据传输时，UART 信号会一直保持在高电平（具体信号幅度由 I/O 的供电电压决定），数据传输时信号发生跳变，传输完成后信号重新回到空闲的高电平状态。

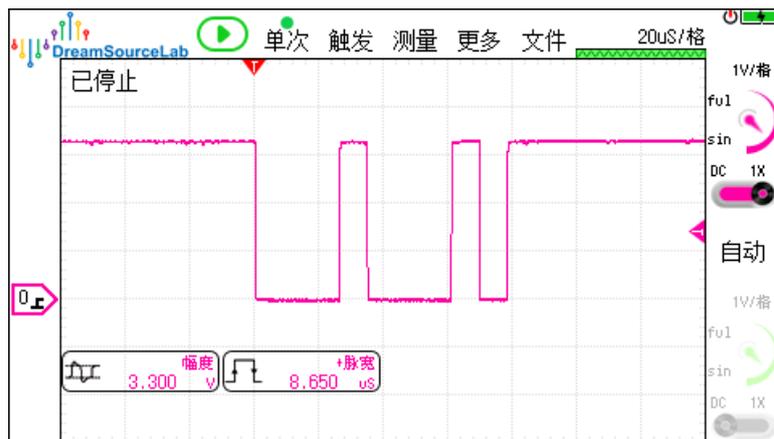


图 1-4 TTL 的 UART 信号波形

1.2.2. 基于 RS232 的 UART 通讯

为了增强驱动能力，以增加传输距离和可靠性，RS232 总线采用了双极性电压信号来进行物理传输。信号在发送/接收之前，通过电平转换芯片实现内部信号和总线信号的互相转换。连接方式和 TTL 电平完全相同，整个物理层只是多了一层电平转换。

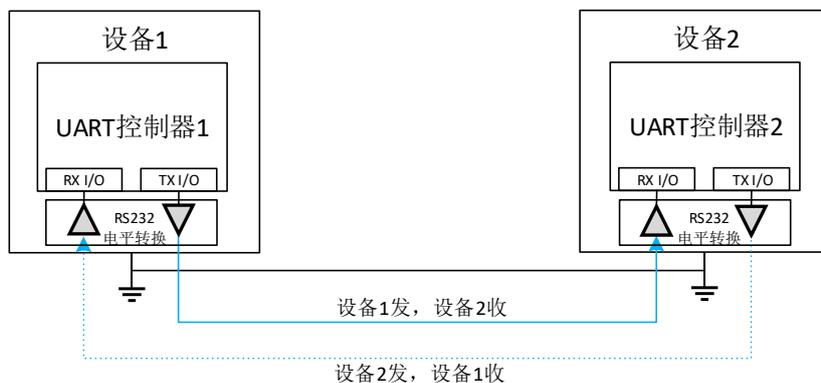


图 1-5 基于 RS232 的 UART 通讯

同样以发送字符‘D’为例，璞石示波器的探头连接到信号端，可以采集到如图 1-6 所示的实际波形。可以看出，RS232 波形在空闲时为负电压，当有数据传输时，信号开始在正负电压之间跳变，传输完成后重新回到空闲的负电压状态。

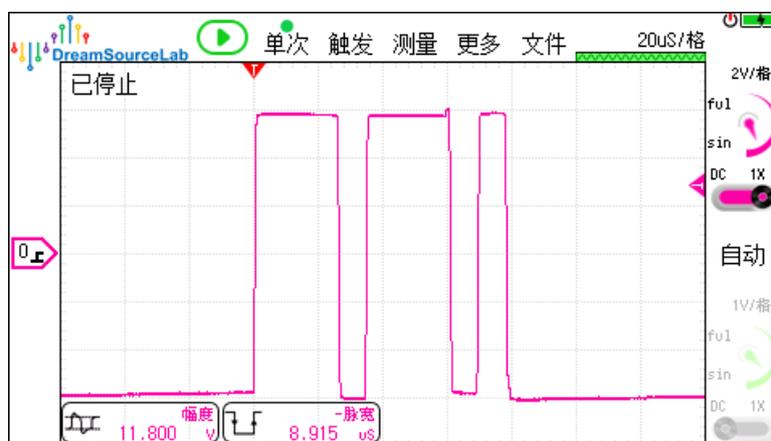


图 1-6 RS232 的 UART 波形

1.2.3. 基于 RS485 的 UART 通讯

RS485 为复杂的工业环境而设计，和其它 UART 协议的物理层相比，RS485 总线最大的特点就是使用了差分信号传输。信号在发送之前，通过 RS485 的收发器把单端信号转换成差分信号，再发送到总线上进行传输；同样在接收之前，总线上的差分信号通过收发器的转换变成单端信号再送给 UART 控制器进行接收。在 RS485 总线上，

如果希望进行全双工的双向通讯，需要两对差分信号线（即 4 根信号线）。如果只进行半双工的双向通讯，则仅需要一对差分信号线即可。

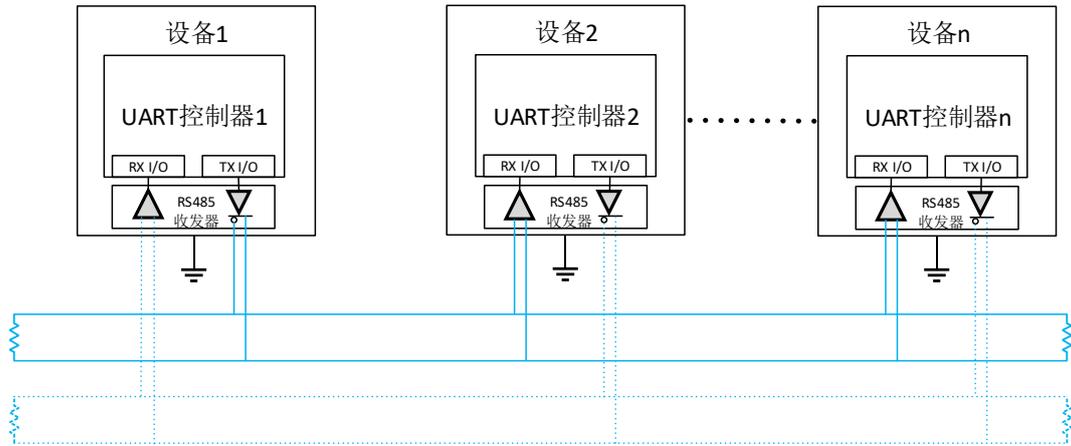


图 1-7 基于 RS485 的 UART 通讯

还是以发送字符‘D’为例，使用璞石示波器 2 个通道的探头（共参考地），分别连接到其中一对差分信号的 A/B 端，可以采集到如图 1-8 所示的实际波形。可以看出，A/B 端的波形为互补关系。A 端波形为正向逻辑（空闲时为正电压），B 端波形为反向逻辑（空闲时为负电压）。

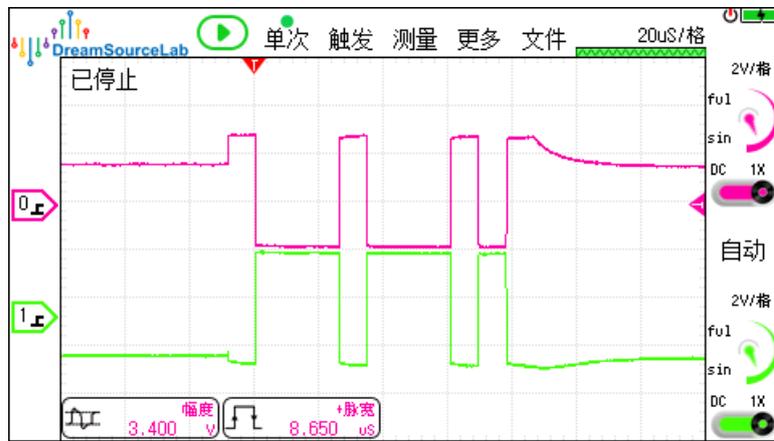


图 1-8 RS485 的 UART 波形

2. 协议规范

2.1. UART 帧结构

在上一节的介绍中，我们通过示波器观察了实际的 UART 波形，那么这个波形到底是怎么被准确的识别成字符‘D’，而不是其它内容的呢？这就涉及到 UART 协议帧结构的定义。如图 2-1 所示：



图 2-1 UART 协议帧结构

当两个设备需要通过 UART 协议进行通讯时，它们需要同时约定好以下内容：

- 每一位信号的时间长度 T （波特率 = $1/T$ ）
- 帧结构中每一项的具体位数
- 是否有校验位，以及校验位的机制（奇/偶/..）

有了这些约定，接收设备只需要等待起始位的到来，再对之后的波形进行固定间隔的采样即可获得传输的具体信息。以字符‘D’的波形为例，其解析过程如图 2-2 所示：

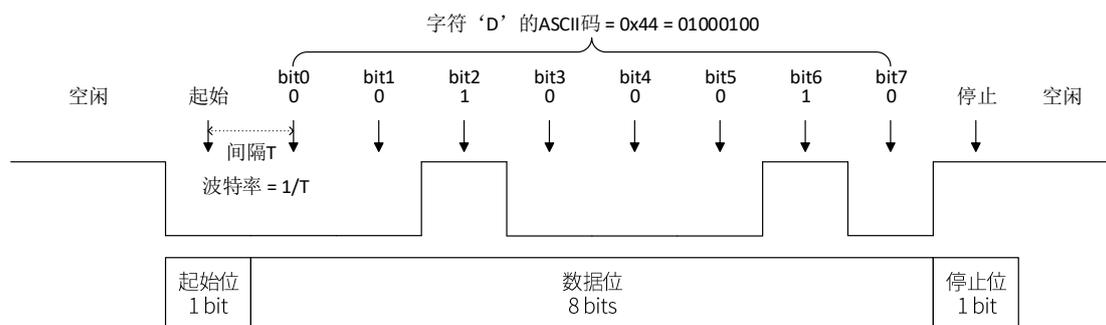


图 2-2 UART 波形分析

2.1.1. 波特率

波特率是 UART 协议，或者说所有异步串行协议，非常重要的一个概念，即单位时间内（1 秒）可表示的 bit 个数，或者也可以表述为 bit 位宽的倒数。例如一个波特率为 115200 的 UART 波形表示 1 秒可容纳 115200 个 bit 位，也就是说每一位 bit 数据占大约 8.68 μ s 的时长。

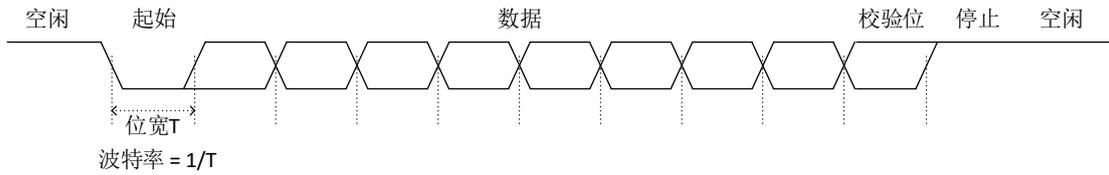


图 2-3 波特率的定义

UART 等异步串行协议，为了简化信号物理连接，降低通讯成本，一般只有一根信号线，无法同时传输数据和时钟信号。收/发设备为了正确解析波形就需要在相同的波特率设置下。而相同的波形使用不同的波特率获取的信息可能会完全不同。对于接收设备来讲，只有起始位可以作为一帧数据的同步点，其它数据都是通过波特率来确定具体的取样位置。

还是以字符'D'的波形为例，如图 2-4 所示，如果用错误的波特率接收，就可能会得到完全错误的信息。

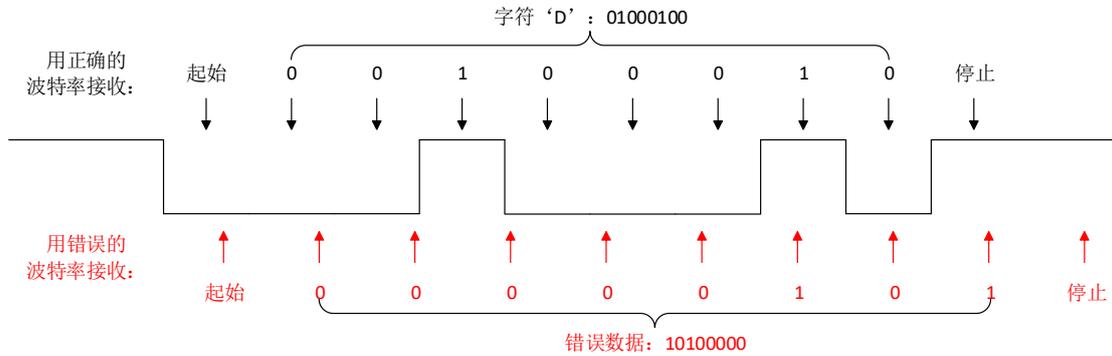


图 2-4 波特率错误

2.1.2. 空闲位

设备之间不传输数据时以持续的高电平表示空闲。空闲位持续时间越长，两个数据帧间隔也越长，单位时间传输的数据就越少。



图 2-5 空闲位

2.1.3. 起始位

UART 接收端会一直检测信号线上的电平变化，开始传输数据时，发送端将信号线从高电平拉到低电平结束空闲状态，并保持一个 bit 位的时长。接收器检测到高低电平转换时，开始接收信号。



图 2-6 起始位

2.1.4. 数据位

数据位包含传输的实际数据，如果使用了奇偶校验，那么数据位长为 5~8 bits，如果没有使用奇偶校验，则位长为 5~9 bits。在一般情况下，数据位为 8 bits，数据首先从最低有效位开始发送，高位在后。



图 2-7 数据位

2.1.5. 校验位

校验位可以用来提高传输的可靠性。如果信号在传输过程中因为干扰而导致某些位置的电平产生错误，通过计算接收的数据和校验位是否匹配即可判断数据是否有传输错误，从而给应用层提供有效信息来决定接受/丢弃对应的数据。



图 2-8 校验位

2.1.6. 停止位

停止位表示一帧数据的结束，实际信号是一段长时间的高电平。停止位的时间长短可以设置为 1、1.5 或者 2bits 的停止位。大部分情况使用 1bit 的停止位。

因为 UART 是一个异步协议，每一帧的开头可以用跳变沿来同步，但是停止位只能通过波特率来计算相对位置，如果在停止位的位置识别到一个低电平，则会产生帧错误。在通讯过程中，为了减少波特率的误差导致的问题，可以设置不同的停止位长度来适配。



图 2-9 停止位

3. 使用逻辑分析仪分析 UART 通讯

3.1. 示波器 vs 逻辑分析仪

示波器作为主要的通用测试测量仪器，在观察模拟信号的细节方面有着天然的优势。比如说实时运行，配合高波形刷新率，可以方便地观察到突发的异常信号；再比如进行噪声评估以及信号质量优化时，示波器可以很好的对比同一条总线上，不同节点上信号的异同，从而找出隐藏的信号完整性问题。

但是示波器亦天然有着一些不方便的地方，除去通道少，屏幕普遍偏小，操作没有电脑+鼠标的方式灵活可控这些形态上的问题，针对数字信号的采集和分析，示波器最大的问题就是存储深度太小。一台几十万元的示波器存储深度也可能只有几百 Mpts。同时，因为存储深度的限制，当需要做稍微长一点时间的采集时，示波器的真实采样率就会剧烈的下降，导致采集到的波形无法还原真实的信号，失去了分析的意义。

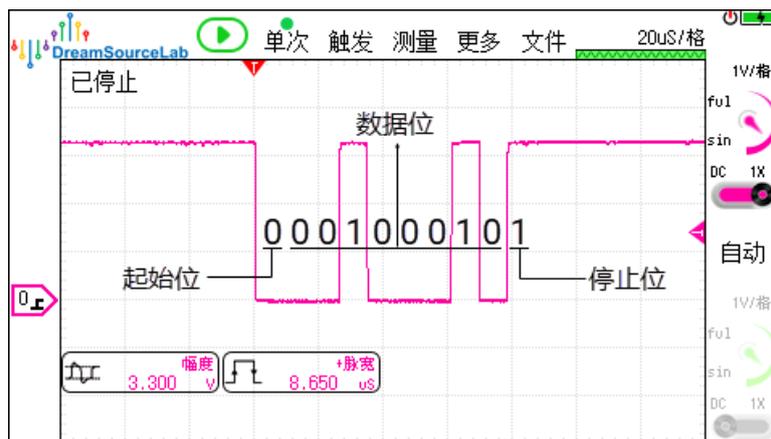


图 3-1 璞石示波器分析 UART 信号

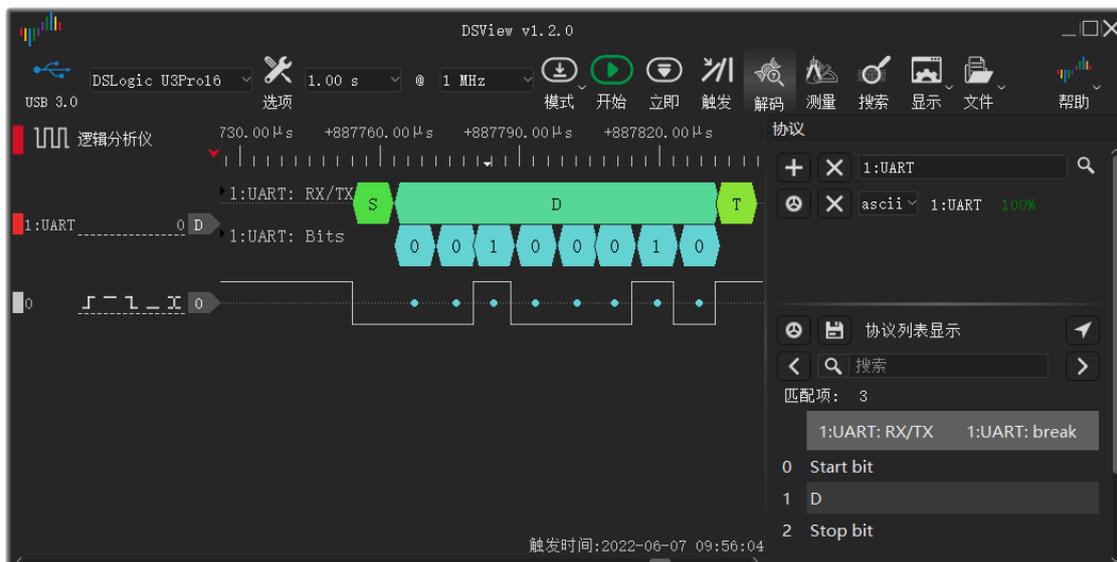


图 3-2 DSLogic 分析 UART 信号

逻辑分析仪是只针对数字信号采集和分析而专门设计的测试测量仪器，其采集出来的波形结果只有高低电平两个状态（也就是逻辑上的 0 和 1）。但是却很好的弥补了示波器在采集和分析数字信号上的不足。

1. 大部分逻辑分析仪都是在 PC 上操作，借助桌面处理器的强大性能，以及成熟操作系统的交互体验，极大地提升了数字信号分析的便捷性和效率。
2. 逻辑分析仪通道数多，入门级别的 DSLogic 逻辑分析仪就有 16 个通道可以同时采集，用户也可以以非常合理的成本获取拥有几十个，甚至上百个通道的逻辑分析仪产品。
3. 逻辑分析仪的存储深度更大，比如 DSLogic 在 stream 模式下可以达到 16G 的存储深度，并且同时保持相当高的采样率，远超绝大多数示波器的采集能力。
4. 逻辑分析仪普遍提供丰富的协议解码功能，可以对数字信号进行深入的分析。例如 DSLogic 目前支持的解码协议就多达 100 多种。大部分示波器的解码能力有限，而且大多需要单独购买相应的插件。

我们接下来将重点介绍如何使用逻辑分析仪进行各种情形下 UART 信号的采集和分析。

3.2. 建立逻辑分析仪的测试环境

3.2.1. DSLogic 的连接

DSLogic 是一款基于 USB 连接的便携式逻辑分析仪。DSLogic 主机负责信号采集和缓存，然后通过 USB 把数据传输给 PC 上的上位机软件 DSView。DSView 负责波形数据处理，显示，分析，以及对应的人机交互功能。

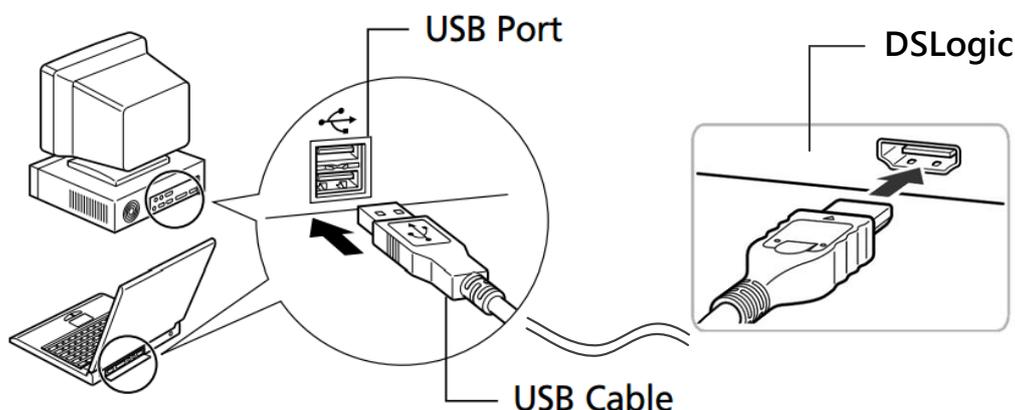


图 3-3 连接 DSLogic 到 PC

使用时，只需要将 DSLogic 主机通过 USB 数据线直接连接到 PC 主板上的 USB 端口，打开 DSView 软件，并确认 DSLogic 主机上的指示灯变为常绿状态。此时便完成了逻辑分析仪的连接和初始化工作。如图 3-4 所示，在 DSView 软件上可以观察到左上角的当前设备名称，以及对应的连接状态。

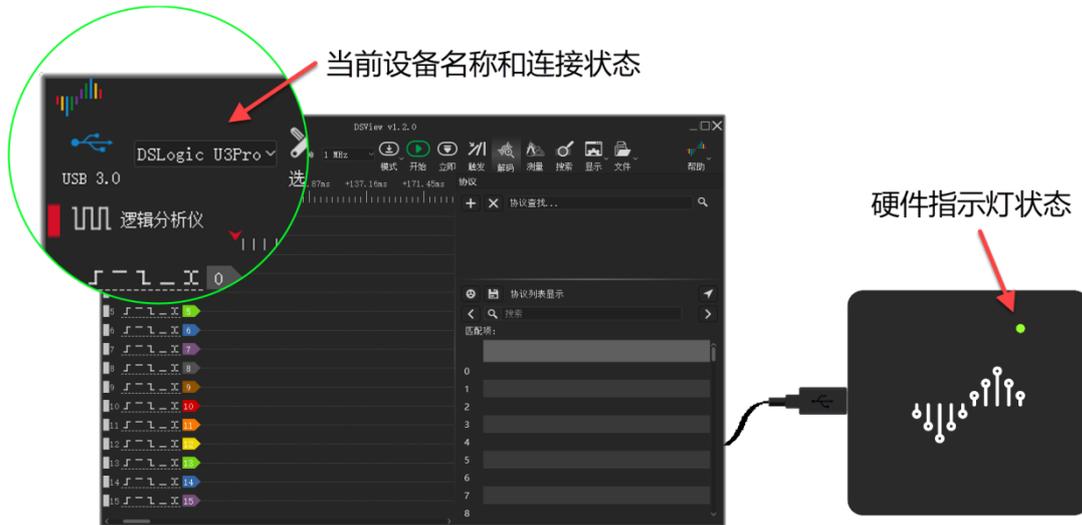


图 3-4 DSLogic 连接状态

3.2.2. UART 信号连接

对于所有的测量来讲，测试设备和被测信号必须基于一个共同的参考平面，也就是通常我们所说的接地，这样被测信号才能被正确采集和测量。在某些情况下，会遇到没有做显性的接地，但是测量结果貌似没有问题的情景，这是因为测量设备与被测信号通过其它的方式进行了共地连接，比如使用的同一个电源供电，或者参考地同时接入了市电的接地端等等。但是在进行严谨的测量时，测量设备应根据被测信号进行就近接地处理。被测信号到测量设备的信号接地端环路面积越大，信号就越容易被干扰，从而产生测量误差甚至测量错误。

针对 UART 信号来讲，接入逻辑分析仪时，最简单的情况只需要连接 2 根线，一根接参考地，一根接被测信号即可。针对我们之前提到的 UART 协议的不同物理层实现，信号连接的方式会稍有不同。

例如对于 TTL 电平的 UART 信号，除了连接参考地之外，只需要连接需要测试的信号（TX 或者 RX）分别接入逻辑分析仪对应的通道输入即可。

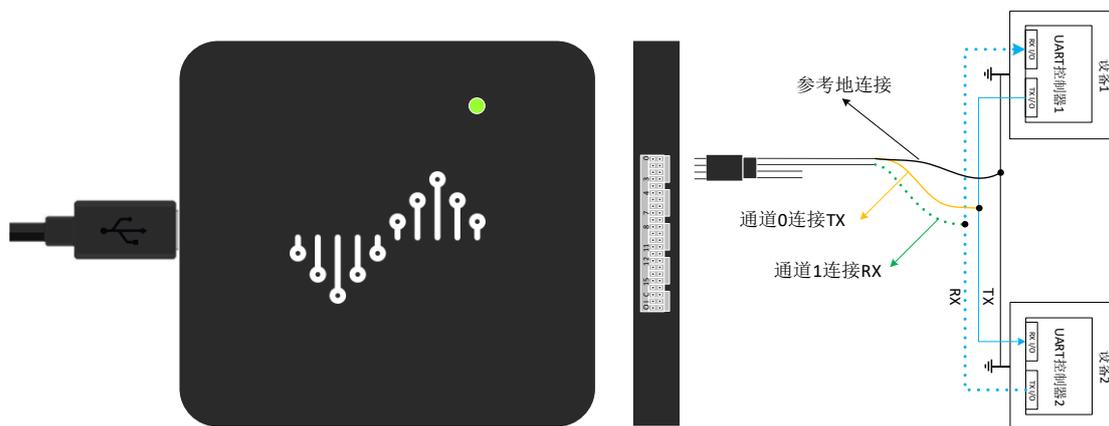


图 3-5 TTL 信号与逻辑分析仪连接

对于 RS232 总线信号，我们既可以测量电平转换之后的总线信号（DSLogic 在排线输入端的耐压范围为 $\pm 30V$ ，RS232 的 $\pm 15V$ 的信号范围可以直接接入），也可以测量电平转换之前的内部数字信号。其接线的方式和 TTL 信号一致，唯一需要注意的是 RS232 信号对电平进行了翻转（空闲时为负电压，即低电平），在进行 UART 协议解码时，我们可以在 DSView 的解码设置窗口进行对应的信号电平翻转（Invert Signal 选项设置为 yes）。

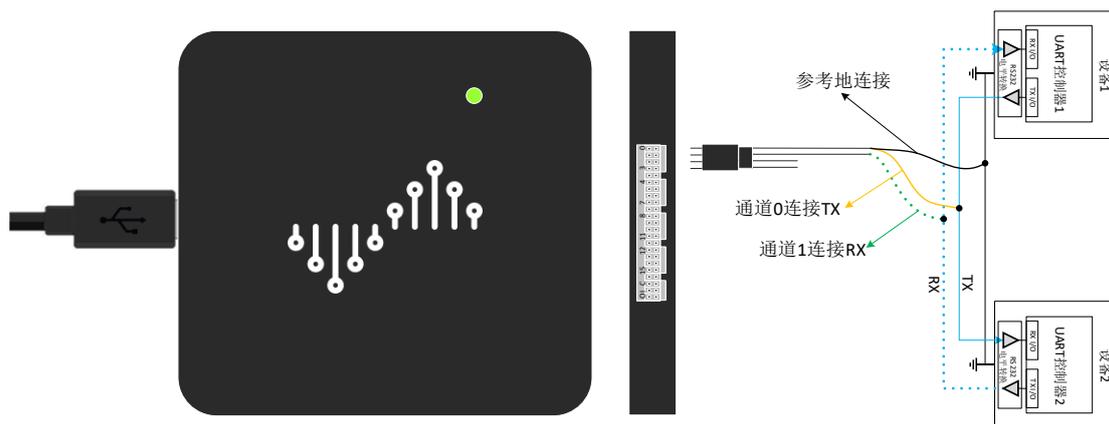


图 3-6 RS232 信号与逻辑分析仪连接

RS485 总线使用的是差分信号，当我们用逻辑分析仪进行信号采集时，需要特别注意被测信号的连接方式。首先，逻辑分析仪的参考地需要连接到对应的被测设备 RS485 收发器的参考地，这样才可以保证逻辑分析仪有正确的电平参考平面；其次，对于差分信号，逻辑分析仪只需要采集到与内部信号同极性的高低电平即可。从数字信号的角度来讲，差分的方式只是提高信号传输的可靠性，不增加额外的信息量。如图 3-7 所示，使用逻辑分析仪测试 485 总线信号时，连接对应设备收发器的参考地，以及差分总线上的 A+ 信号即可。

当然，如果只能采集到 B-信号，我们也可以在 DSView 进行解码时，设置 UART 解码器进行信号翻转的方式来得到正确的协议解析结果。

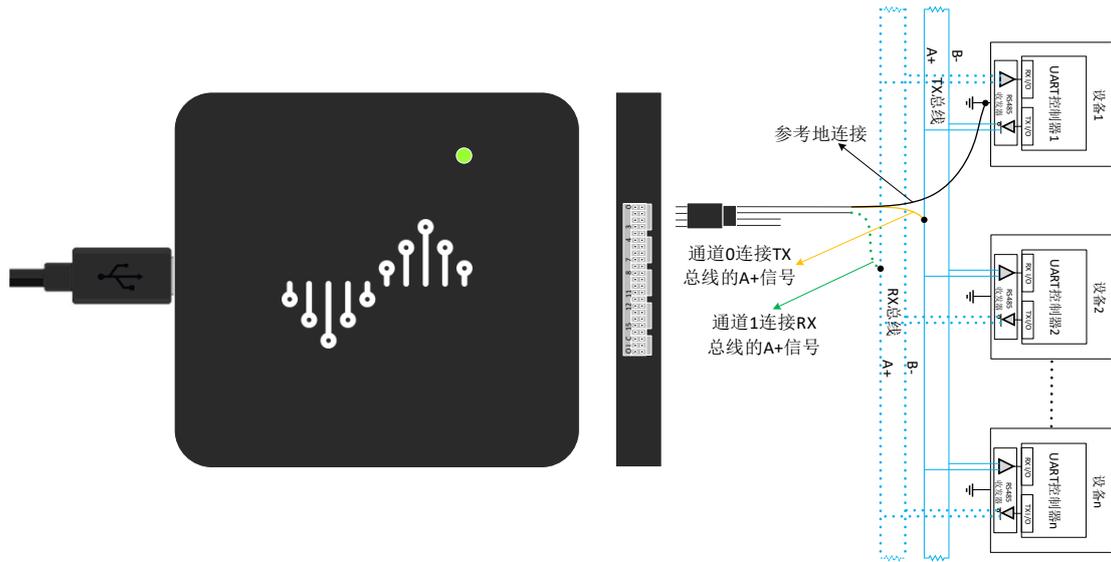


图 3-7 RS485 总线与逻辑分析仪连接

3.2.3. 阈值设置

逻辑分析仪的阈值为高低电平的判决电压，也就是说当输入信号电压高于阈值电压时，采集到的便是高电平；当输入信号电压低于阈值电压时，采集到的便是低电平。

针对 UART 协议，我们之前已经介绍了其有不同的物理层实现，不同的物理层实现有不同的电平标准。我们只需要选择一个对应的阈值可以区分不同情况下的高低电平即可。如图 3-8 所示的阈值基本可以满足这个要求。

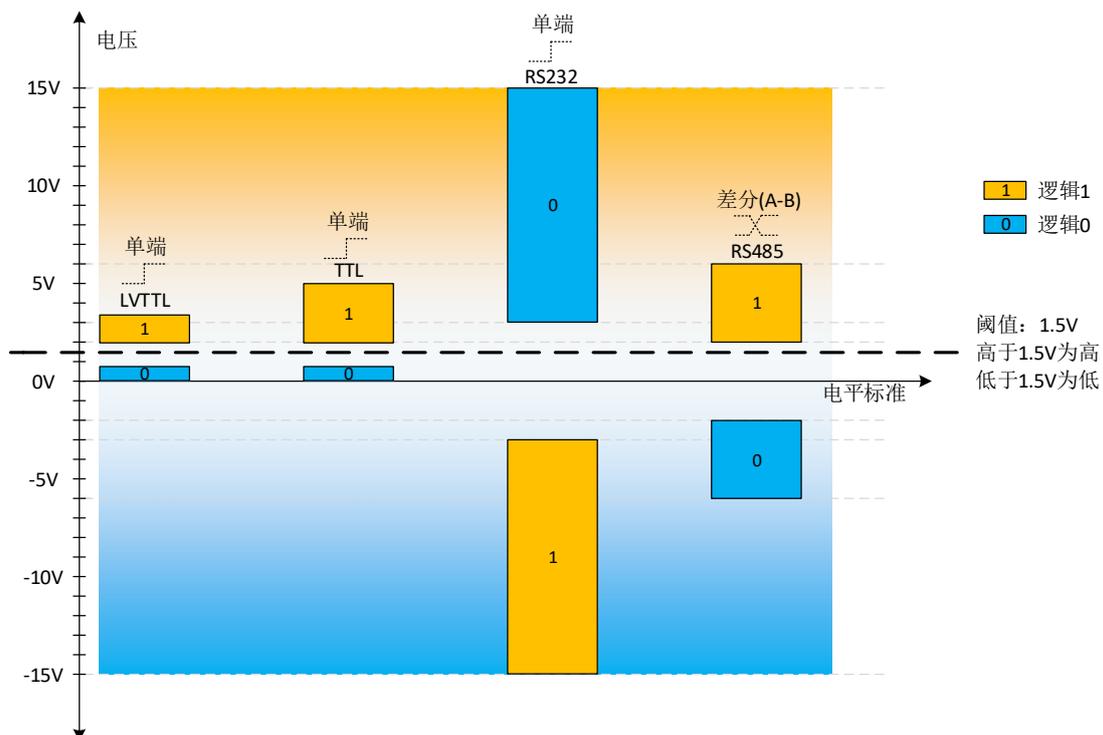


图 3-8 UART 信号的阈值设置

在某些情况下，因为配置或者调试的需要，RS485 总线的差分信号幅度可能在比较大的范围变动。因为 RS485 的规范规定只要 A-B 之间的差分电压大于 200mV 既可以认为是高电平。此时我们可以借助璞石示波器，观察实际的信号电平幅度，再来决定合适的阈值设置。只需要确保设置的阈值电压可以合理的分辨出高低电平即可。

3.3. 简单的 UART 信号采集

绝大部分 UART 信号都是低速信号，例如最常用的波特率 115200，表示信号频率只有 115.2KHz。使用 DSLogic 的默认设置即可直接采集到此类信号。此处以主机固定 1s 间隔，重复发送字符‘D’为例，介绍如何使用逻辑分析仪采集和分析此类 UART 信号。

3.3.1. 信号采集

在准备好硬件和信号连接之后，直接点击 DSView 软件的“开始”按钮，就可以以默认设置（1MHz 采样率采集 1s 时长）采集被测信号。如果希望修改采样率和采集时长，直接在对应下拉框中选择对应的参数即可。采样率的设置原则推荐为被测信号最高频率的 10 倍或以上，例如对于 115200 波特率的 UART 信号，最快的信号频率为 115.2KHz，当选择 1M 采样率时，每 bit 信号会有 8 或者 9 个采样点，对应的相位误差为 $\pm 1/8$ （12.5%）；当选择 10M 采样率时，每 bit 信号会有 86 或者 87 个采样点，对应的相位误差为 $\pm 1/86$ （1.16%）。当然一般情况下也不推荐使用过高的采样率，因为采样率越高能采集的时长就越短，另外采样率过高可能采集到慢速信号单个跳变沿的多次跳变，从而给协议解码带来困扰。图 3-9 便是使用默认设置采集得到的信号波形。

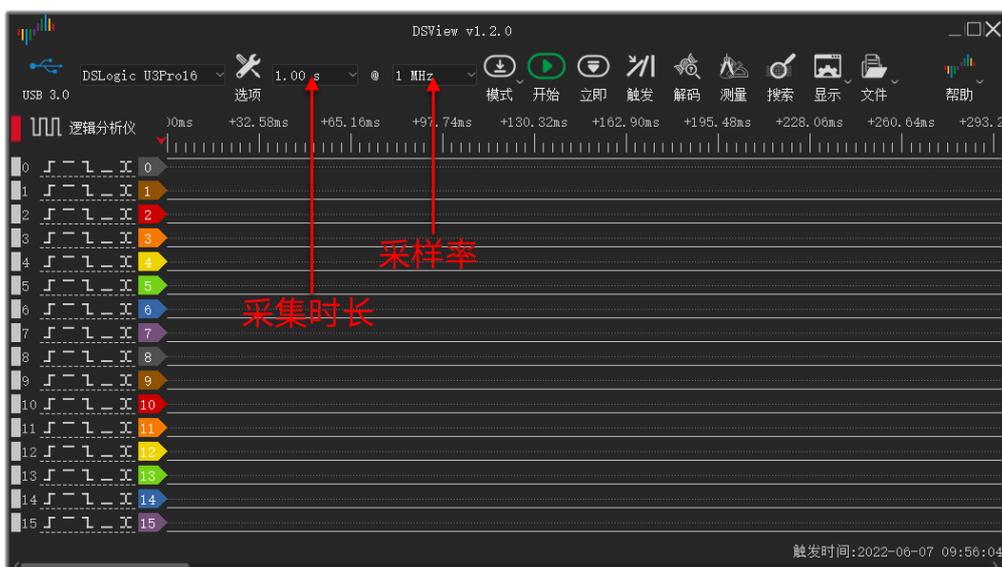


图 3-9 默认采集

从图 3-9 可以看出，通道 0 采集到持续为高电平的信号，但是并没有发现有任何跳变。仔细观察波形窗口上面的时间标尺以及窗口底部的滚动条，可以得知当前窗口显示的只是采集到的部分波形，不是全部信号。此时可以通过鼠标滚轮缩放，或者在波形区域右键双击的方式显示整个采集时长的所有波形。得到如图 3-10 所示的结果。

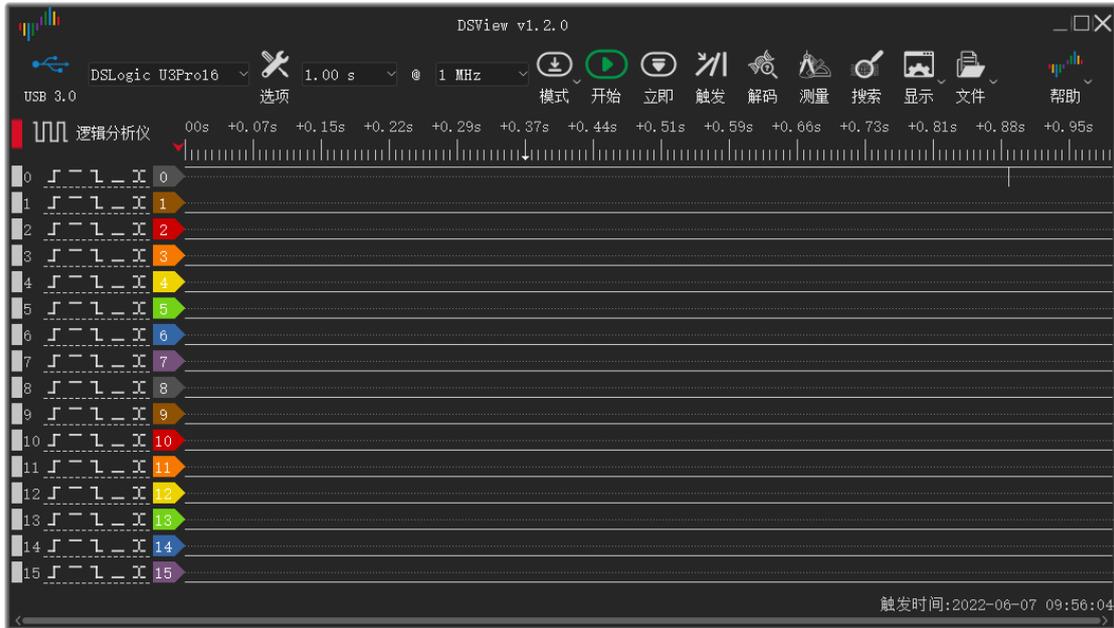


图 3-10 显示整个采集窗口

在上图的基础上，我们希望只观察通道 0 的跳变部分。可以在“选项”窗口关闭其它通道的波形显示，调节通道信号的显示高度。然后将鼠标指向目标波形，并同时通过鼠标滚轮放大所指位置的波形；或者可以通过按住鼠标右键框选希望放大的波形区域。另外，按住鼠标左键可以对波形进行左右移动。得到如图 3-11 所示的波形细节。

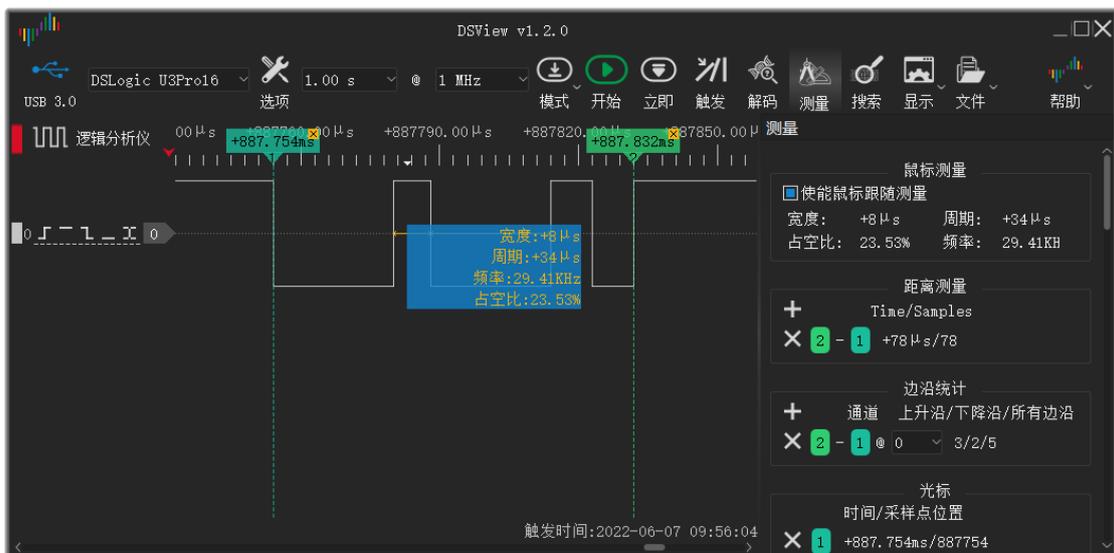


图 3-11 观察波形细节

在观察波形时，把鼠标放到脉冲波形上既可以显示当前信号的脉冲宽度，周期，频率以及占空比等信息。同时，也可以通过鼠标双击左键的方式添加光标，移动光标时会自动吸附到靠近的跳变沿。

3.3.2. UART 解码

逻辑分析仪采集到的信号波形只有高低电平（0/1）在时间上的变化，这不利于我们对信号传输的过程和内容进行有针对性的分析。虽然我们可以根据协议规范，用肉眼来翻译波形对应的协议内容，但是将大大降低分析的效率。

DSView自带的解码器可以提供丰富的协议解码内容，使得枯燥的波形变成有意义的字符，标记以及数据内容。极大的方便了工程师对信号的分析理解，从而找出错误信息或者关键数据。

在DSView软件里对已经采集到的波形进行协议解码非常简单，只需要打开“解码”面板，搜索到对应的解码器（这里我们选择1:UART协议），设置对应解码器的选项（包括显示/通道/帧结构/格式等），确定之后软件就会开始对选定通道进行UART的协议解码。并在协议通道显示解码的结果。如图 3-12所示。



图 3-12 添加 UART 解码

在选择解码器时，DSView提供便捷的协议搜索功能，只需要在协议搜索框输入关键字，DSView便会列出所有包含这些字符的解码器，用户在其中选择正确的解码器即可。如图 3-13所示和UART相关的有两个协议0:UART和1:UART，其中0:UART是一个简化的解码器，其不进行bits信息的解析，也不支持高层协议的堆叠，当我们只想获取波形对应的数据时，可以使用此简化协议来减少解码的时间和占用的内存资源。



图 3-13 协议搜索

解码器的设置窗口有很多和解码相关的设置选项，不同协议的设置窗口内容也会不同。其中有一些所有解码器都必须设置的选项，比如通道和协议信号之间的对应关系。此处我们使用的是通道0抓取的UART信号，那么对应的需要解码的通道就应该设置成通道0。

和协议相关的设置选项我们在帧结构的部分都有详细讲解，在实际操作种也可以通过串口的上位机软件（串口调试助手等）来获取这些信息。如图 3-14所示。



图 3-14 协议相关的选项

对于解码的数据内容，DSView 支持不同的显示格式。例如之前采集的 UART 信号按照 ASCII 码的方式显示字符 ‘D’，也可以修改成十六进制（Hex）的显示方式，结果便会显示 0x44。如图 3-15 所示。目前支持的显示格式有：hex（十六进制）、dec（十进制）、oct（八进制）、bin（二进制）以及 ascii 码。

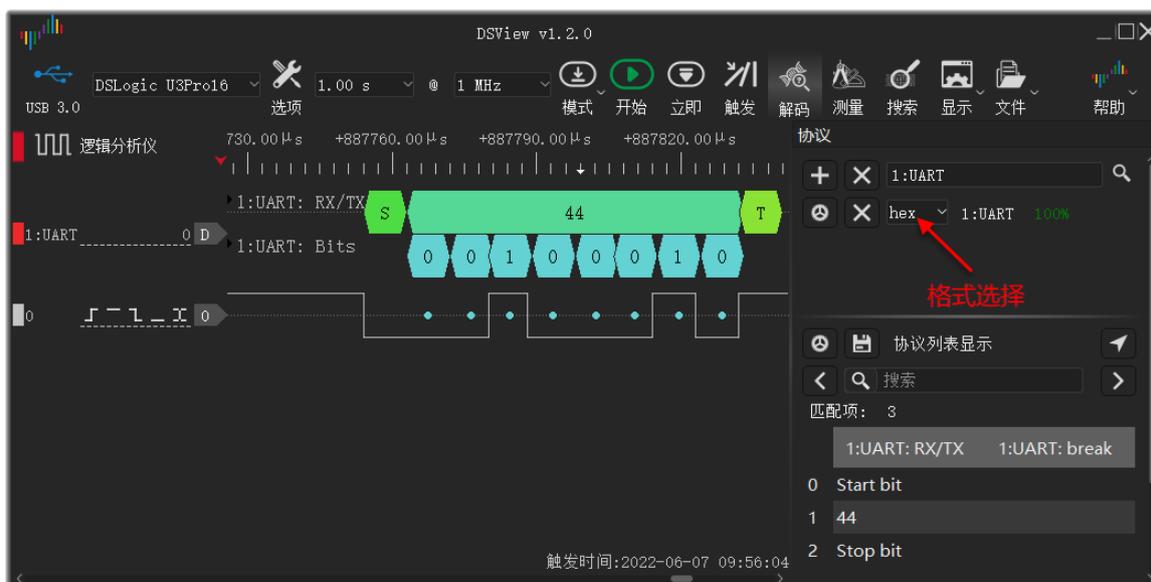


图 3-15 格式选择

3.4. 单次 UART 传输的采集

当我们要捕获的波形是一个偶发信号时，我们并不知道什么时候波形会出现，如果使用直接的采集方式可能大部分时间就只能捕获到空闲的信号波形，无法抓取到实际传输时的信号。

这个时候便可以通过触发来进行捕获。通过设置对应的触发条件，我们可以根据波形的变化来控制信号采集的开始时间，从而精准的捕获到希望看到的信号波形。

3.4.1. Buffer 模式

DSLogic 支持两种采集模式 Stream 模式和 Buffer 模式。在 Stream 模式下采集到的数据会实时的传输到 PC，然后由 DSView 处理。Buffer 模式采集到的数据会先缓存在 DSLogic 的内部硬件缓存中，等采集完成后再传输到 PC，由 DSView 处理。

Buffer 模式除了可以达到更高的采样率这个优点外，还支持更多的触发选项和设置，比如多级触发，串行触发以及触发位置的设置等。而 Stream 模式仅支持简单触发。



图 3-16 模式选择

3.4.2. 触发设置

对于简单触发，我们可以直接设置通道旁边的触发选项。其依次对应上升沿，高电平，下降沿，低电平以及任意边沿触发。在触发面板保持默认的简单触发模式即可。例如对于不知道什么时候发生的 UART 传输，我们可以选择下降沿触发。当 UART 信号一直处于空闲状态时，触发就不会被满足。DSLogic 会一直等待触发条件的到来，如图 3-17 所示。



图 3-17 等待触发

当 UART 开始传输第一个起始位时，便会激活触发条件，DSLogic 将采集到触发时的信号波形。如图 3-18 所示。

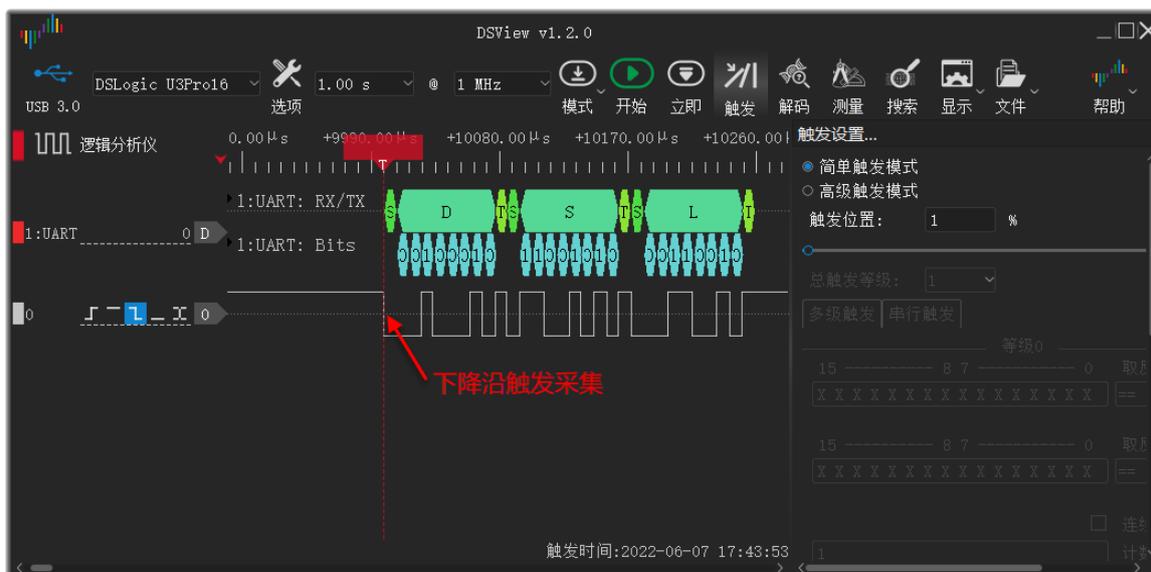


图 3-18 触发采集

我们考虑复杂一点的触发场景，例如设备 A 不断的随机给设备 B 发送数据，设备 B 则根据设备 A 发送的数据内容回复不同的数据。如果我们想要知道设备 B 发送数据 0 之前，设备 A 给设备 B 发送了哪些数据，该如何设置触发条件抓到对应时刻的波形呢？

这里有两个问题需要解决，首先我们需要设置触发的位置，从而看到触发之前的数据，而不是触发之后的数据；其次我们要找出设备 B 发送数据 0 的波形特征，并把其转换成对应的触发条件。

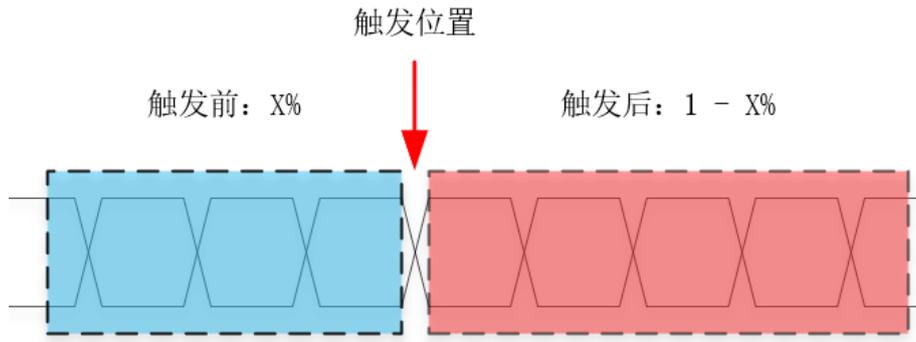


图 3-19 触发位置

根据 UART 的协议规则，发送数据 0 的波形会是一个比较宽的持续低电平脉冲信号。如果 UART 的波特率为 115200，我们设置的采样率为 1MHz，数据 0 的波形如图 3-20 所示，将会是一个 78.125uS 左右的低电平脉冲。

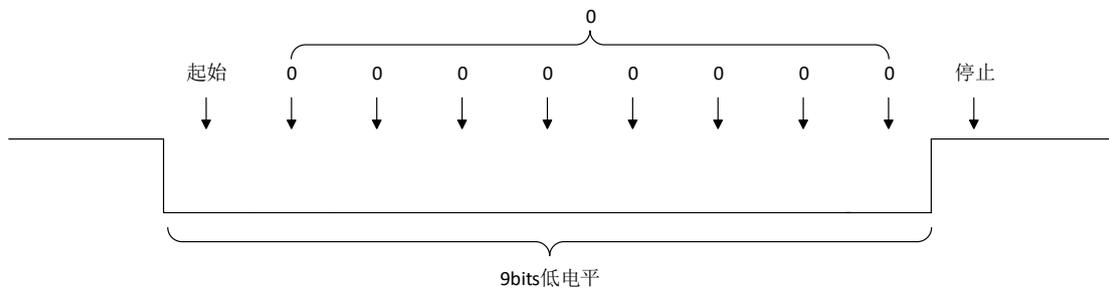


图 3-20 UART 发送数据 0

要设置高级触发或者触发位置都需要 buffer 模式的支持，需要首先在“选项”设置“运行模式”为“Buffer 模式“，同时不要选中“RLE 硬件压缩“的选项。设置高级触发条件为通道 1 出现持续的 75 个采样点的低电平（对应 1MHz 采样率，也就是 75us 的低电平），触发位置设置到 80%，便于观察触发之前的数据。具体设置如图 3-21 所示。

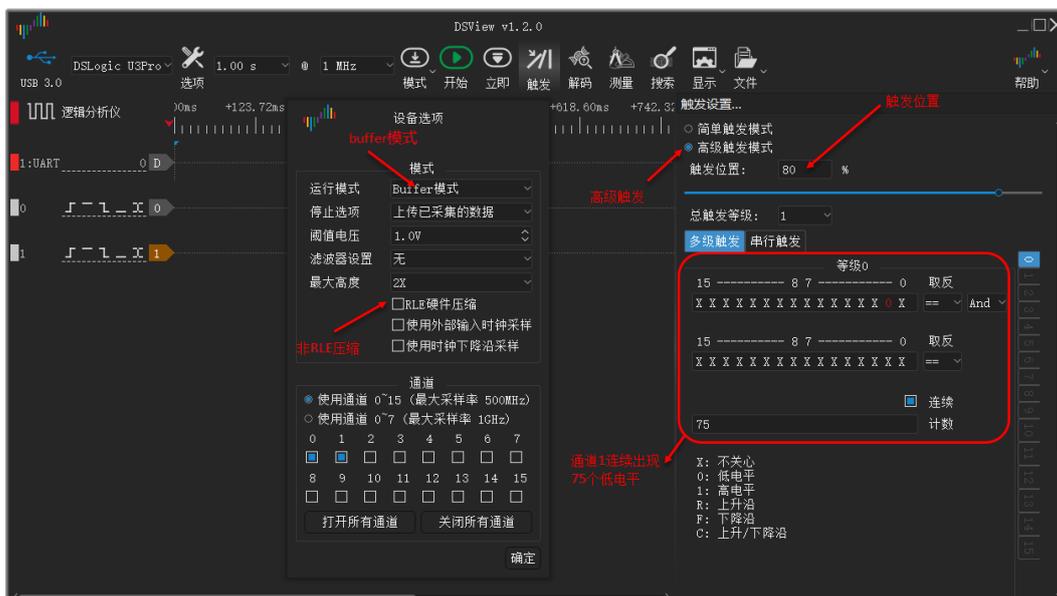


图 3-21 高级触发设置

根据这个触发条件，当设备 B 发送数据 0 时就会触发，采集到的波形如图 3-22 所示。

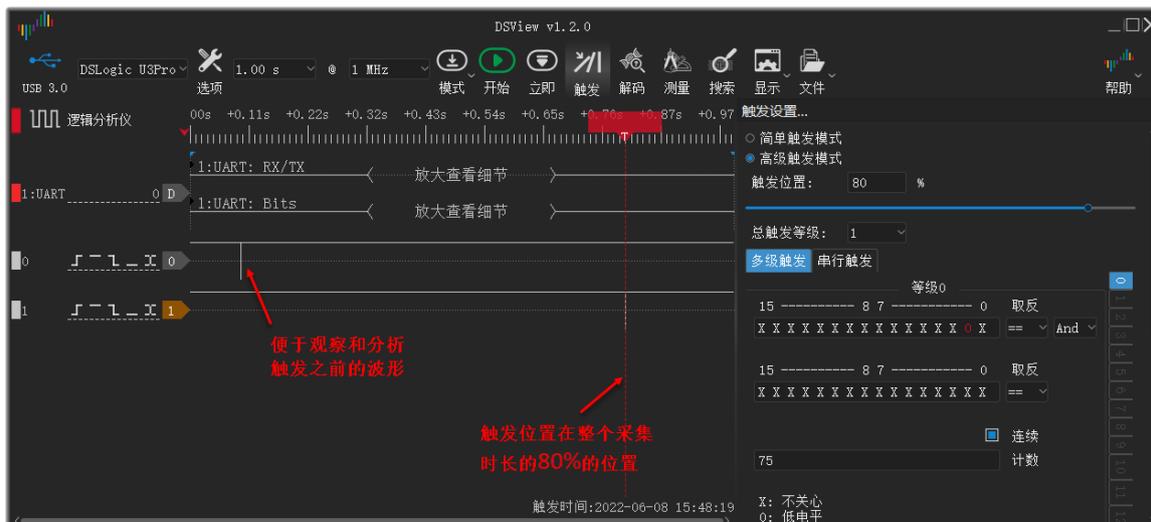


图 3-22 触发位置

3.4.3. 重复采集

在某些情况下，我们需要动态的观察一些信号的状态，比如每次 Burst 发送的数据内容，但是却不知道数据什么时候发出，不可能频繁的去点击“开始”按钮。DSView 提供了“重复”采集的模式，可以在满足触发时自动采集并更新波形。当我们调试一些周期的突发信号时，便可以通过重复模式非常方便的观察动态的采集和协议解析的结果。

重复模式可以按图 3-23 所示进行设置。重复间隔是指两次采样之间至少停留的时间，方便观察上一次采集的波形和解码结果。

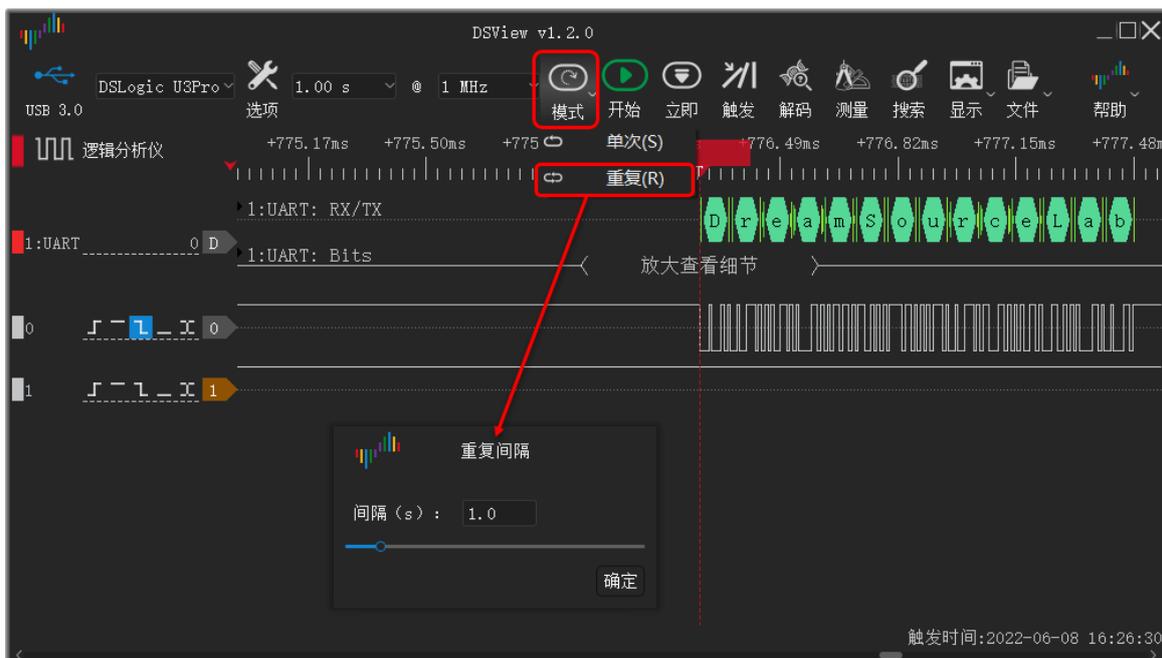


图 3-23 重复采集模式

3.5. 长时间 UART 传输的采集

当需要采集超长时间的波形进行分析时，我们可以使用 Stream 模式。DSLogic 在 Stream 模式下可以实现每个通道 16G 的采样深度。在 1MHz 的采样率下，采集时长可以达到 $16G/1M \approx 16000s$ ，也就是 4 个多小时。

3.5.1. Stream 模式

Stream 模式的设置如图 3-24 所示，不同的 DSLogic 硬件版本支持不同的 stream 模式通道选项。对于 U3Pro16 来讲，基于 USB3.0 的端口，可以实现 3 通道@最高 1GHz 采样率，6 通道@最高 500MHz 采样率，12 通道@最高 250MHz 采样率以及 16 通道@最高 125MHz 采样率。

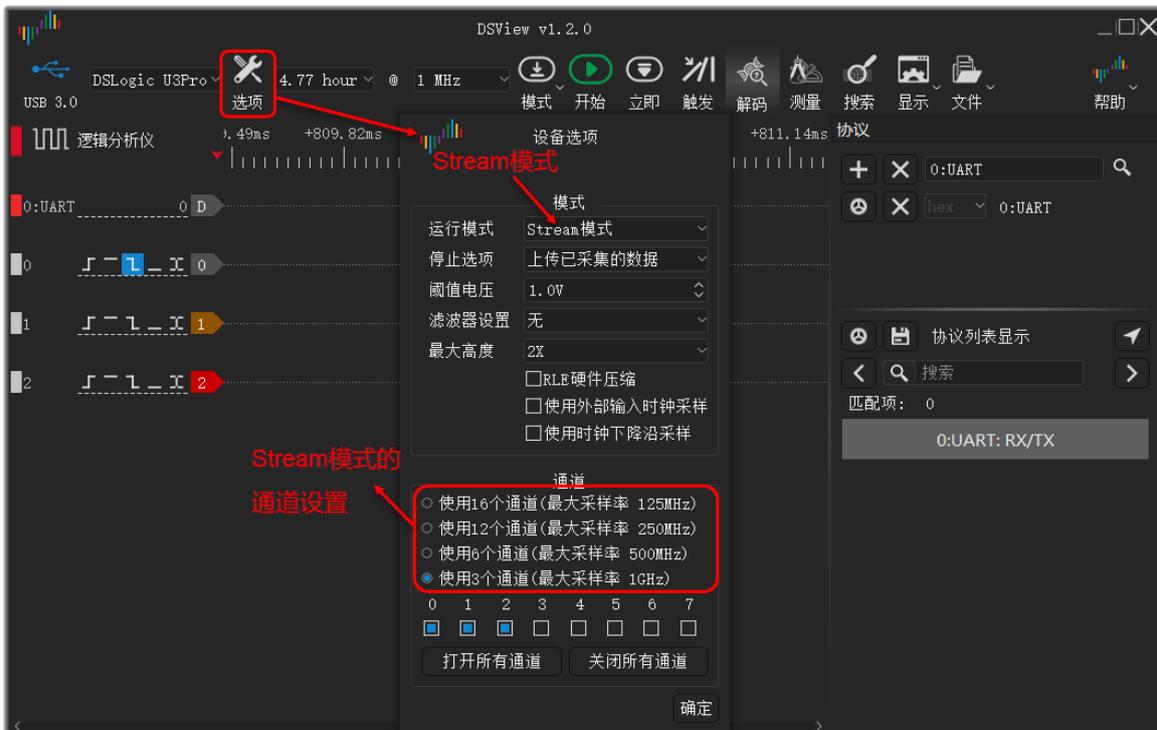


图 3-24 Stream 模式设置

当采集的数据量比较大时，可以使用 0:UART 协议进行解码，和 1:UART 相比，其省略了 bits 信息的输出，不支持高层协议堆叠，可以缩短解码时间，减少内存占用。其它功能和 1:UART 相同。解码效果如图 3-25 所示。

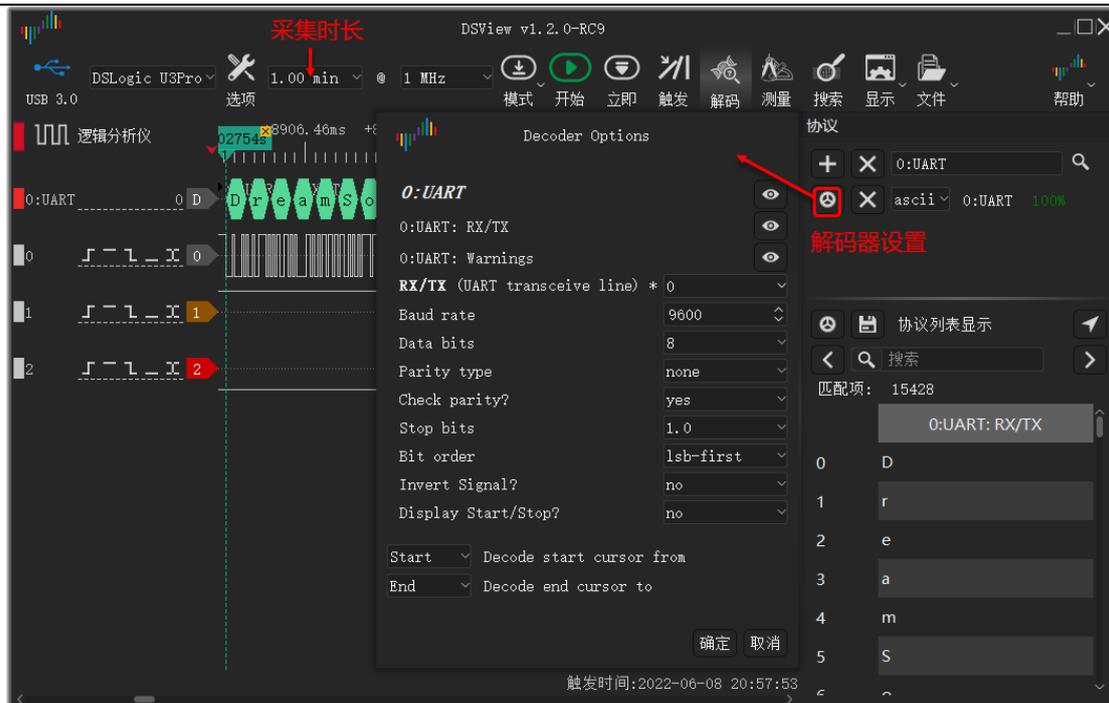


图 3-25 0:UART 解码大量数据

3.5.2. 协议列表显示

如下图所示，图形化的解码结果是和采集到的波形在时间上完全保持同步的，这有助于我们理解波形和找到问题，但同时因为时间的显示比例问题，不利于对解码结果的查看。列表显示可以很好的弥补这一缺陷，使得查看和分析大量的解码内容变得简单。

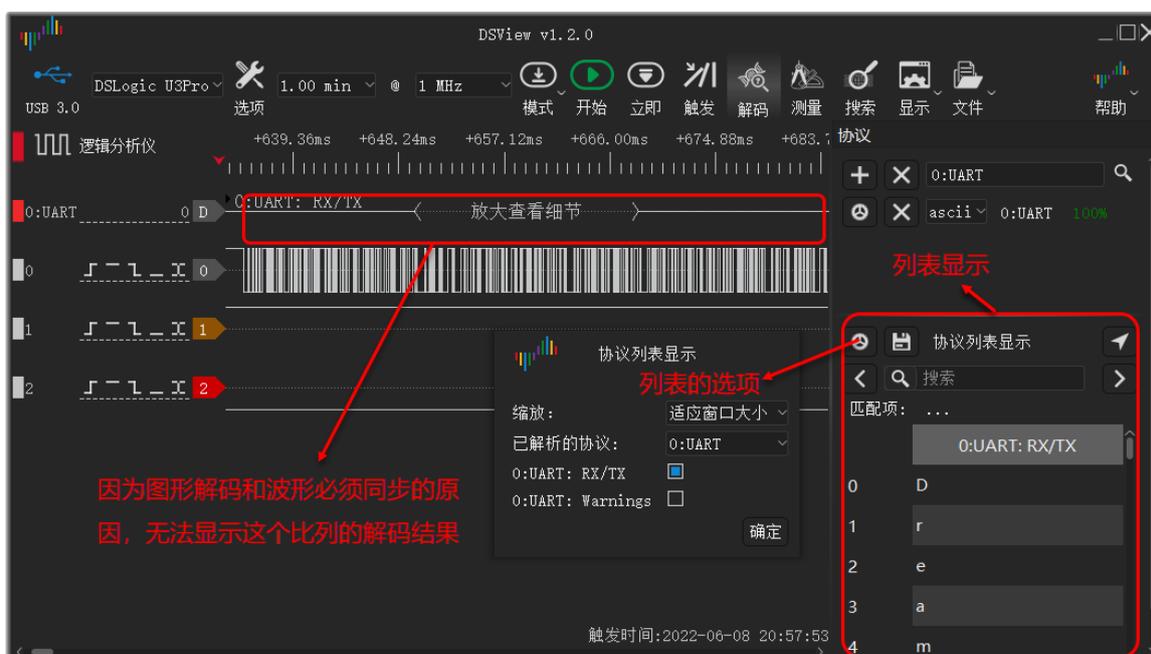


图 3-26 列表显示

DSView 提供在图形结果和列表结果之间进行跳转的方式。在列表结果中单击任意一行即可自动跳转到对应的波形结果，同时你还可以在列表的选项里面选择跳转时是自动放大波形进行显示，还是固定波形当前的缩放比例。

另外，当我们查看某一个时刻的波形时，如果希望跳转到这段波形对应的解码结果在列表显示的位置，只需要点击列表显示窗口右上角的定位按钮即可。

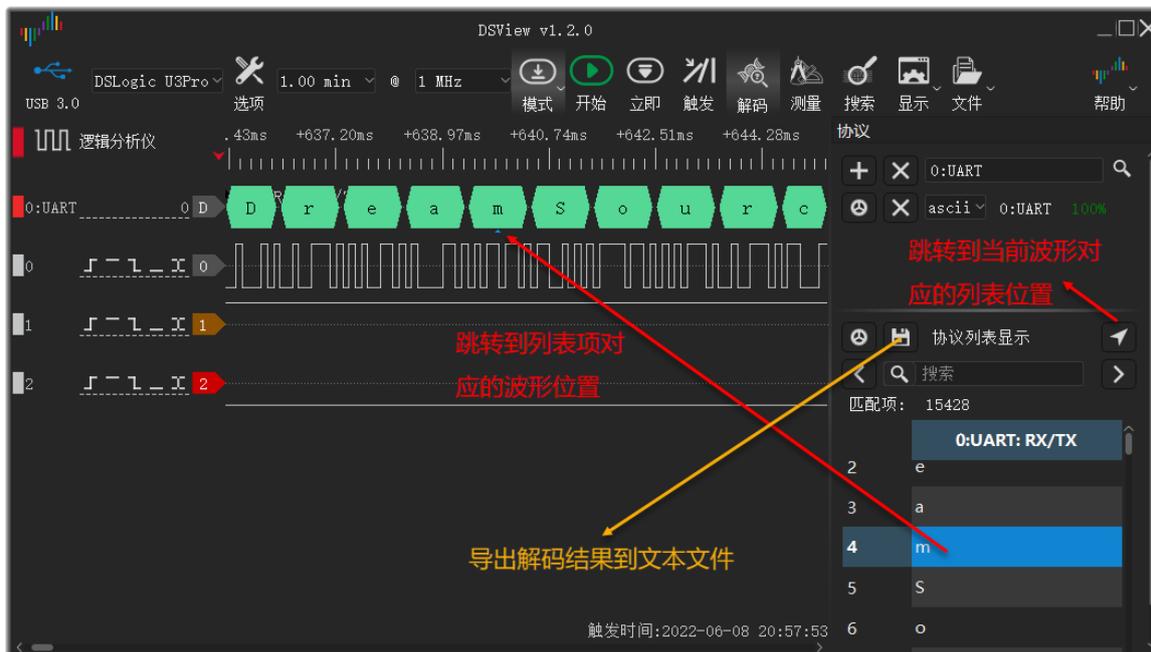


图 3-27 列表和图形解码的跳转

另外，通过列表显示还可以把解码的内容导出到文件。如上图所示，点击列表显示的“保存”按钮，可以选择把当前解码的内容导出到 csv 或者 txt 文件，方便进行更多的第三方数据处理，比如输入到 matlab 进行分析，或者作为其它软件的数据输入。

3.5.3. 解码数据搜索

列表显示的另一个优势就是可以把解码结果当成字符串进行结果搜索，比如搜索特定操作，或者特定字节数据，可以快速定位需要查看的位置。

在搜索框中输入需要搜索的关键字，然后按向前/向后按钮即可找到包含关键字的解码内容，同时波形界面也会跳转到相应位置，以供分析。如果正在分析某一段的解析结果，可以单击选中列表的某一行，向前/向后按钮则会以这一行为起始位置，查找前一个和后一个匹配项，而不用从头开始查找。

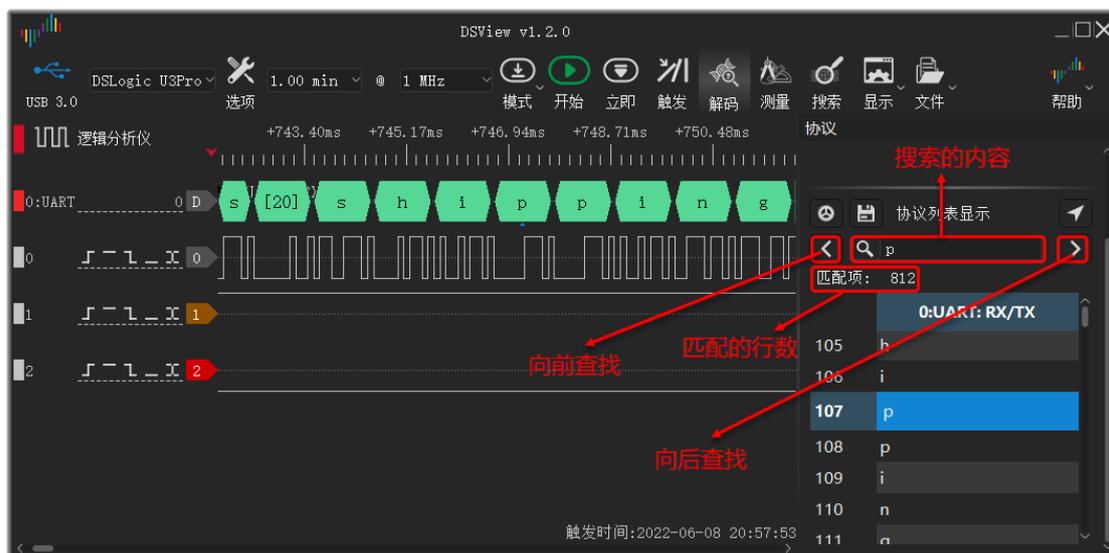


图 3-28 协议内容搜索

对于 UART 协议, DSVIEW 还支持多行数据搜索。可以用于查找数据中出现的某些连续的字符串结果。如图 3-29 所示, 多行数据搜索通过“-”字符来连接多个关键字。

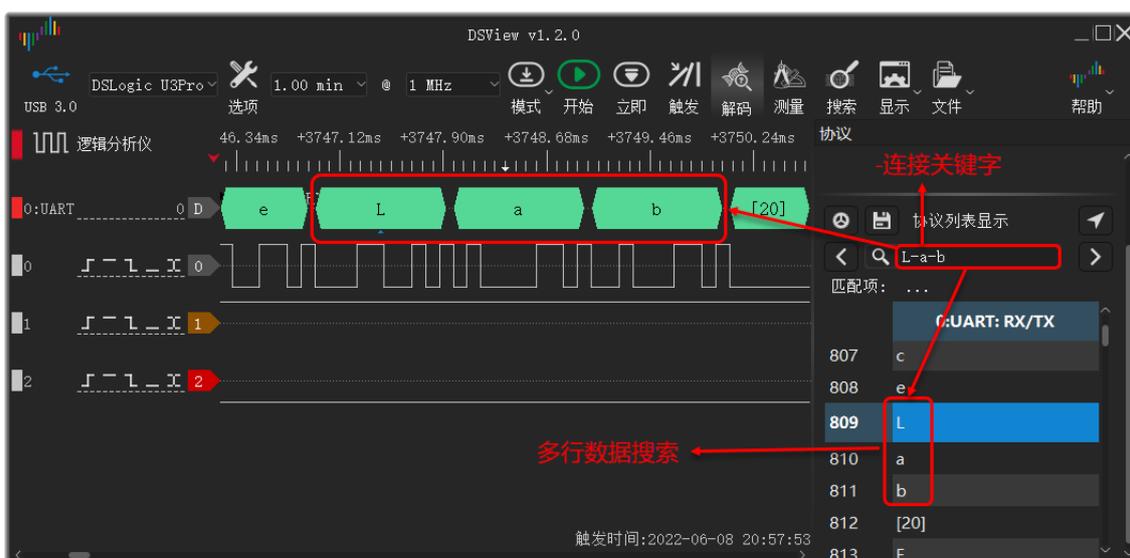


图 3-29 多行数据搜索

3.5.4. 波形搜索

DSVIEW 不仅支持解码结果的搜索, 还支持原始波形的搜索功能。在很多场景可以快速的定位波形。例如波形突发传输时, 因为空闲周期比较长, 无法同时看到多次传输的细节波形, 可以使用波形搜索, 在查看完前一刻的波形, 保证缩放比例不变的情况下, 立即跳转到下一次的波形跳变处。

选中“搜索”工具栏, 软件窗口的下方会出现搜索框, 单击搜索框, 弹出搜索设置窗口, 确定通道跳变沿, 即可通过左右键搜索符合条件的波形。同时, 也可以手动移动搜索光标, 以确定搜索的起始位置。



图 3-30 波形搜索

选中搜索之后会出现带“S”标记的光标，初始状态为灰色，如果点击左/右搜索，并查找到了符合条件的波形，光标会变成蓝色，并自动移动到搜索的位置。如果手动移动了“S”光标，光标会重新变为灰色。

3.5.5. 光标运用

DSView支持丰富的光标操作，不仅可以通过光标进行时序的测量，还可以通过光标来标记和追踪特殊波形。当波形空闲或者间隔时间较长时，可在每一个感兴趣的位置都插入光标，想要跳转到某一个感兴趣的波形时，只需要在标尺区域点击右键，再点击想要跳转的光标序号，既可以跳转到对应的光标位置。

如果想移动光标，只需要将鼠标移动到光标附近，当光标出现加粗显示时，单击鼠标左键即可选中，此时光标便会跟随鼠标移动，确定位置后再次单击左键即可释放光标。当希望光标吸附到特定通道波形的某一个跳边沿时，只需要将移动光标时的鼠标靠近跳变沿附近，光标会自动吸附到跳边沿位置。

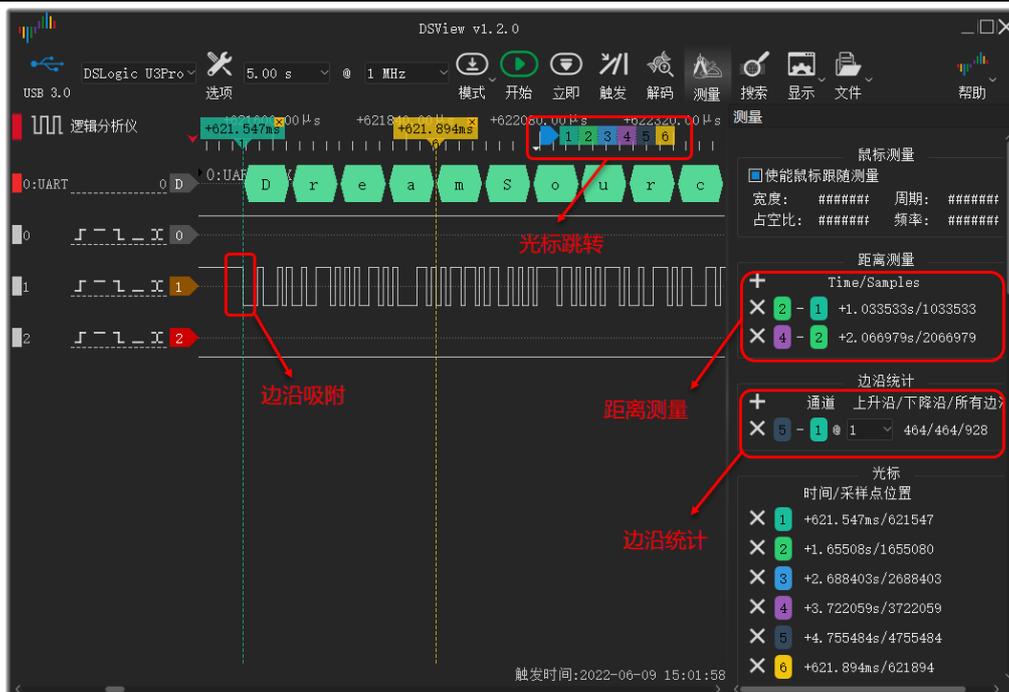


图 4-13 光标运用

3.5.6. 鼠标测量

除了基于光标的测量，DSView 还提供了更加便捷的鼠标测量功能。例如当前脉宽，周期，占空比以及频率的测量（只需要将鼠标放到变化的脉冲上）；边沿个数的测量（在波形上点击开始和结束位置便会自动统计此区间的跳变沿）；以及边沿距离的测量（鼠标点击起始边沿，再移动到结束边沿即可）。

如图 3-31 所示，移动鼠标到起始跳变沿的位置，单击鼠标左键，选中起始跳变沿，移动鼠标到结束跳变沿的位置，便会显示两个跳变沿之间的距离（时长/采样间隔点），再次单击鼠标左键完成测量。

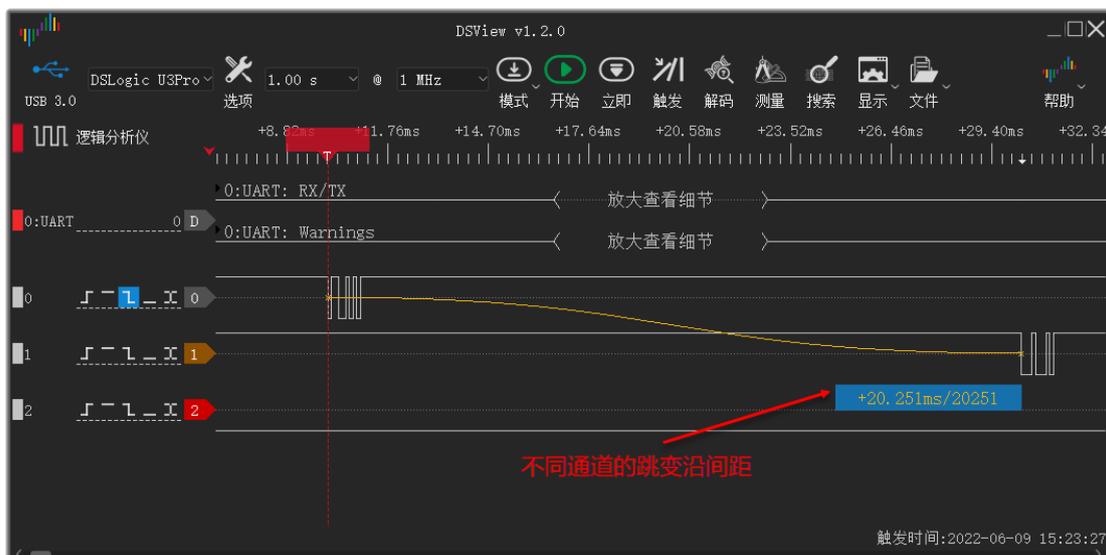


图 3-31 跳变沿间距

4. 常见问题

4.1. 确定未知的波特率

当我们无法事先获取UART波形的波特率时，也可以通过采集到的波形来猜测真实的波特率。DSView提供“Guess bitrate”的解码器来猜测任意波形串的波特率参数。Guess bitrate 的原理就是依次找到波形中最窄的脉宽，作为一个bit的波形宽度，然后计算其倒数作为波特率。所以其可能出现多个结果，我们只需要观察对应的脉冲是不是合理的脉宽，然后选择最大的波特率结果即可。如图 4-1所示。

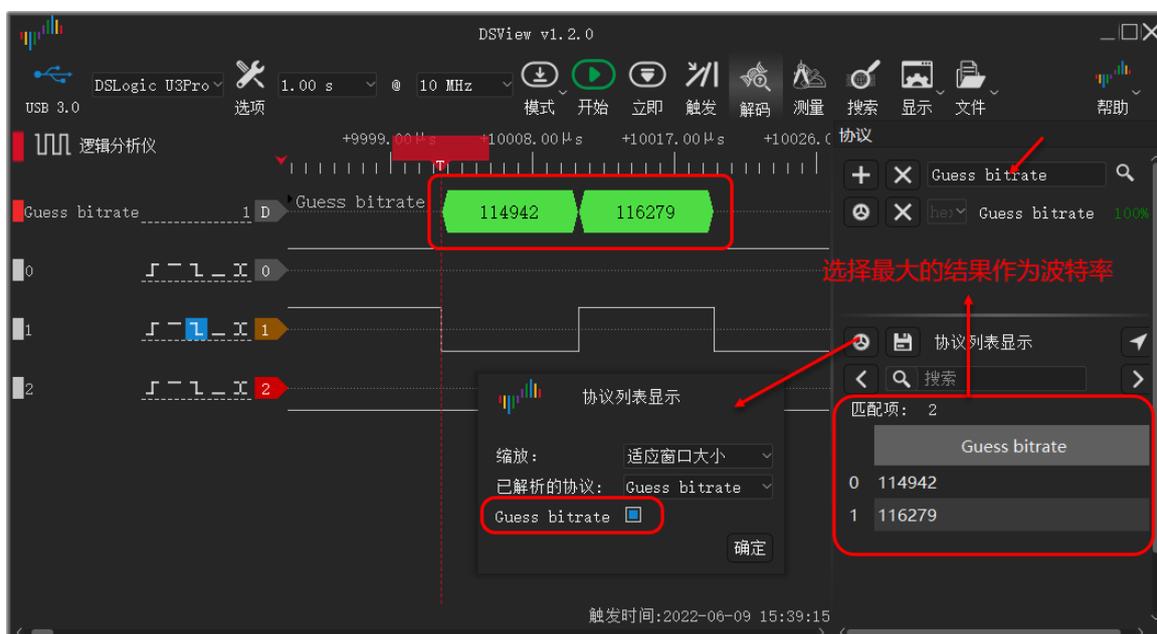


图 4-1 猜测波特率

获取到波特率之后，便可以设置到UART解码器的选项中。因为 UART协议每次起始位都会重新同步，所以波特率的误差只会在一帧数据中累计，只要一帧以内的误差没有超过bit位的边界，便可以得到正确的解码结果。

当然，如果波形样本较少，比如波形中没有出现过单个bit宽度的脉冲，那么猜测的波特率很可能不正确，导致解码结果错误。当有不符合协议的波形出现时，UART解码器也会给出可能的错误提示，比如frame error等，如图 4-2所示。有经验的工程师可以通过观察波形和解码结果，去调整波特率，以便获得正确的解码结果。

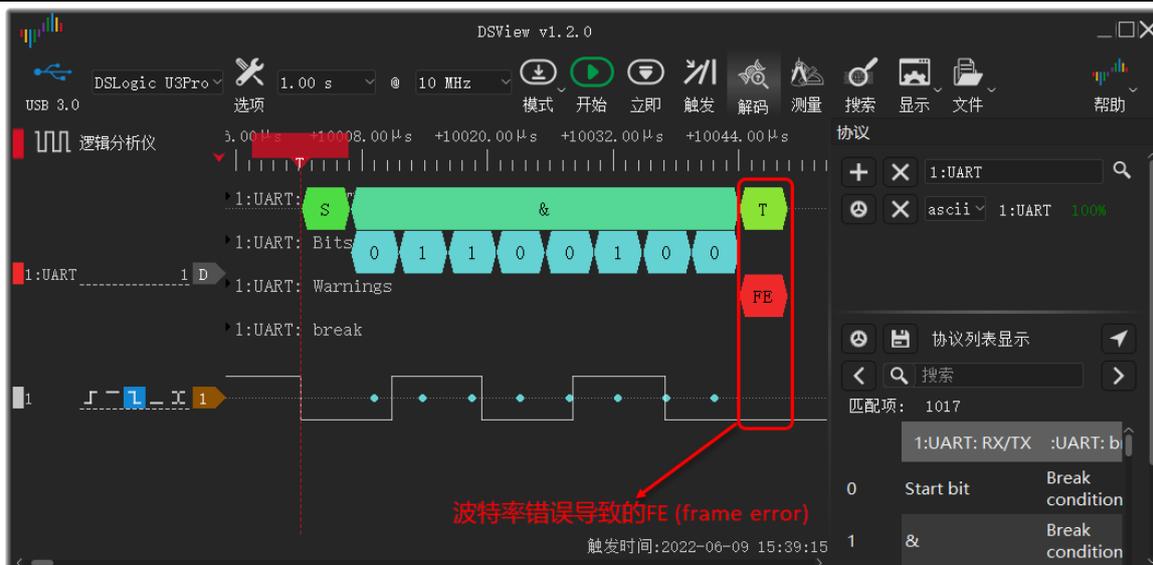


图 4-2 解码错误

4.2. 反相信号的解码

在之前的章节我们提到，UART 协议可以通过不同的物理信号进行传输，有些物理层的信号可能与协议信号是反向的关系（高低电平互换），比如 RS232，或者 RS485 差分信号当中的 B-信号。当我们采集到的信号是反向时，DSView 的 UART 解码器也可以通过设置来获取正确的解码结果。

用户只需要在 UART 解码器的设置中将“**Invert Signal?**”选项设置为“yes”即可。反向信号的解码如下图所示。

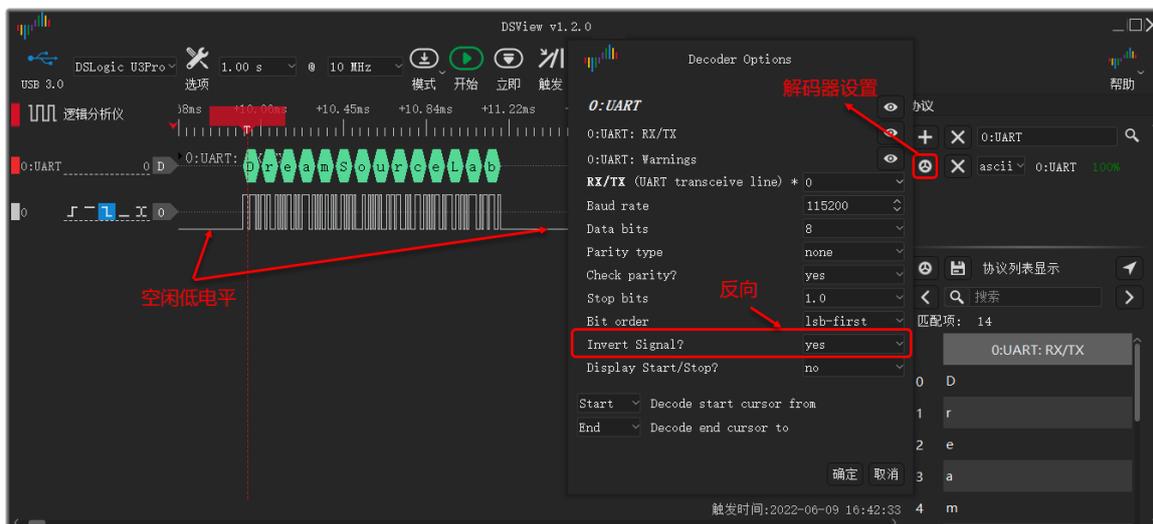


图 4-3 反向信号的 UART 解码

4.3. 多组 UART 通讯的解码

DSView 的 UART 解码器特意设计成单通道的方式，这样可以方便多路通讯进行解码时通道的任意设置。例如对于一个全双工的 UART 通讯，既有 TX 信号又有 RX 信号，那么我们只需要分别添加两个 UART 解码器来对应 TX 和 RX 的解码即可。

如下图所示，便是同时对 3 路 UART 信号进行解码的情况。

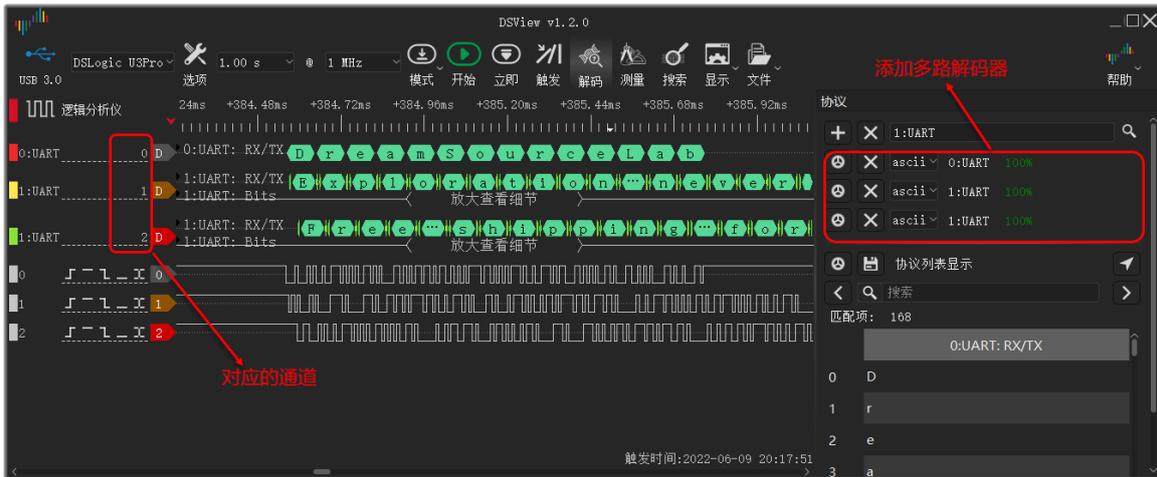


图 4-4 多路解码器

4.4. 起始位置导致的解码错误

对于任何通讯过程来讲，设备必须从初始状态出发，才能正确的接收到通讯内容，也就是正确的对协议帧进行识别和分析。对于逻辑分析仪来讲也是一样，如果信号的采集正好从一帧数据的中间开始，那么错误的起始位置很可能导致错误的解码结果。如下图所示，设备不停的发送字符串“DreamSourceLab”，但是因为首次捕获的位置正好位于数据帧的中间，导致第一段波形的解码变成了乱码。



图 4-5 波形截断错误

为了避免错误的解码数据，我们可以人为的设置解码的起始位置以及结束位置，从而避免被截断的数据帧导致的解码错误。解码位置可以是任意光标的指定位置，操作时只需要在插入对应的光标后，解码器的设置里面设置对应的起始和结束位置即可。如下图所示，设置起始位置避免了第一段截断波形的错误解码。

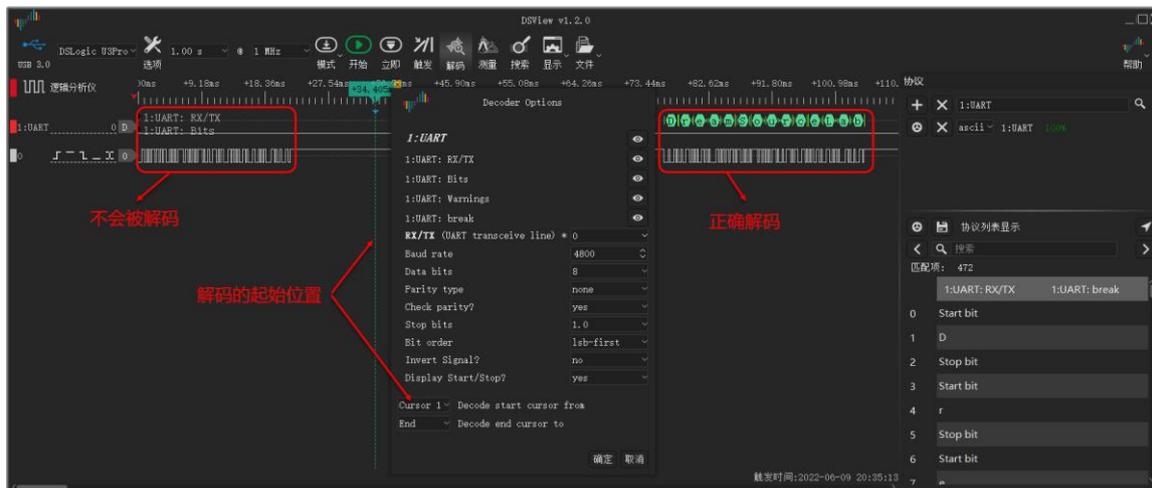


图 4-6 区域解码

5. 进阶分析

5.1. 多层协议解析

UART 是一种使用非常广泛的基础协议,有很多在实际应用中使用的传输协议都是在 UART 的基础上扩展而来。

例如:DMX512, LIN, MIDI 以及 Modbus 等,其底层的通讯基础都是 UART 协议。

根据这个特点,DSView 同时支持多层协议的解析堆叠。当我们添加一个复杂协议的解码器时,DSView 会自动从顶层构建整个协议栈,从而使得解码的结果更加丰富和完善,非常有助于用户对协议的深入分析和理解。

我们以 Modbus 为例,来了解一下 DSView 分析多层协议的过程。基于 UART 的字节传输功能,Modbus 详细的规定了每个字节的具体意义,这样通过把多个 UART 协议帧组合在一起便形成了 Modbus 的帧结构。如下图所示。

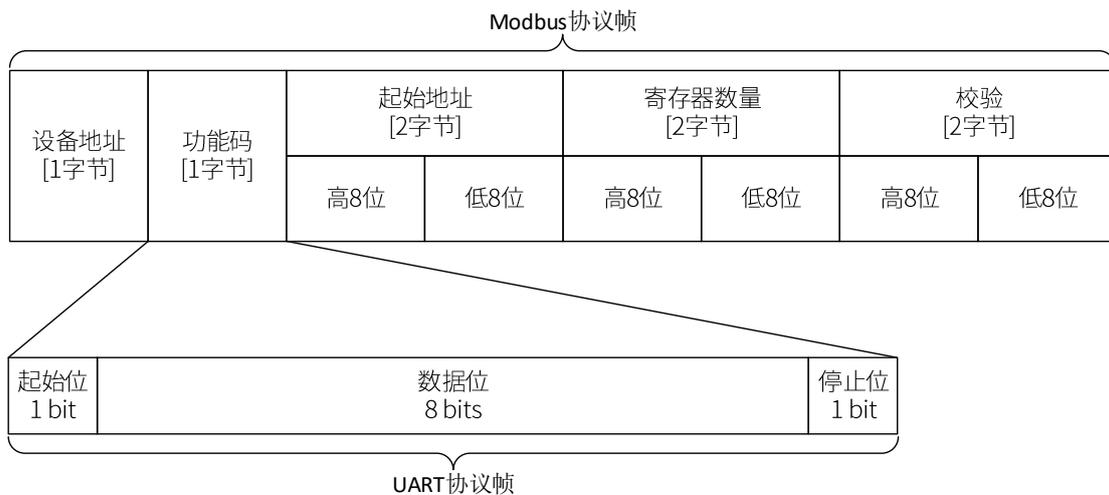


图 5-1 Modbus RTU 主站设备请求帧格式

在使用 DSView 进行此类协议解码时,只需要在协议搜索框输入对应的协议并添加即可。DSView 会自动构建整个堆叠协议的设置页面,如下图所示,用户进行相应的设置即可。

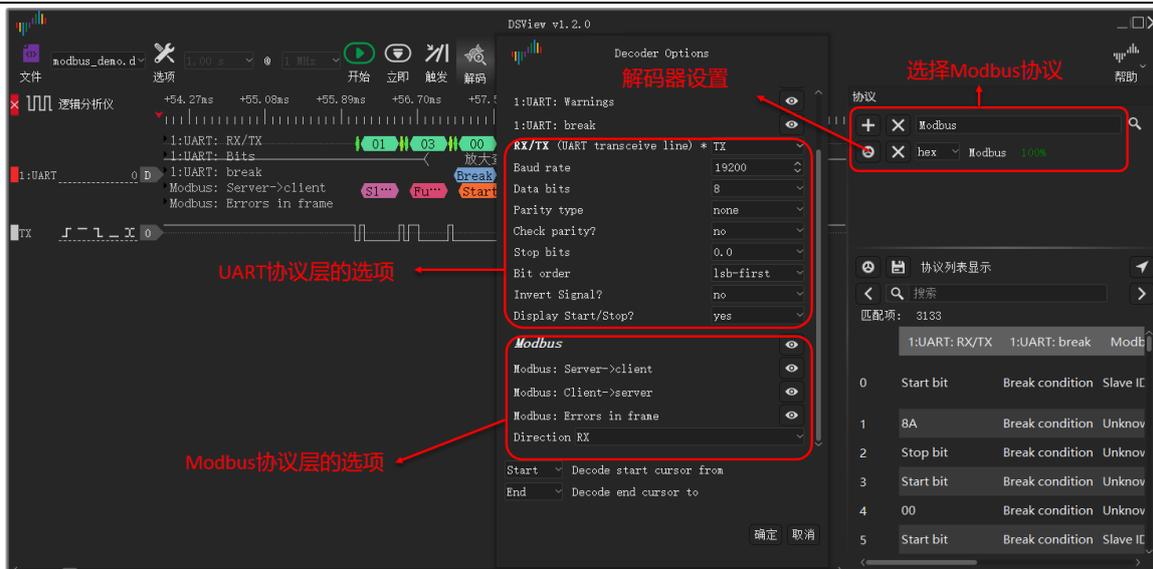


图 5-2 Modbus 解码器的设置

如下图所示，从 Modbus 的解码结果可以清晰的看出协议的堆叠结构。每一帧的 Modbus 由 8 个 UART 字节帧组成。Modbus 帧的内容也会有清晰的注释标明其具体的含义。当出现传输错误时，这样的解码结果可以非常清晰的定位错误的位置，以及错误的原因（是底层 UART 的问题，还是 Modbus 协议层的问题）。如果不希望看到底层的细节，在解码器的设置页面把对应的解码内容隐藏即可。

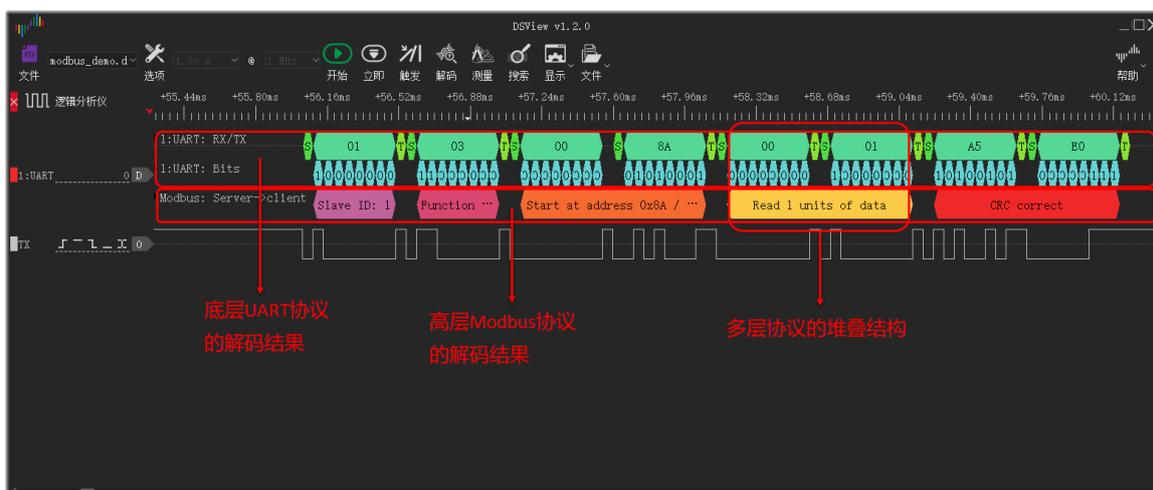


图 5-3 Modbus 解码