Configurable Protocol Decoding of Manchester And NRZ-Encoded Signals

WHITE PAPER

March 15, 2013

Summary

The industry's first NRZ and Manchester configurable protocol decoders accept a broad range of physical characteristics for NRZ- or Manchester-encoded signals.

Introduction

There are many ways to approach design of a serial data-communications interface. It's often easier and more cost-effective to use one of the industry-standard protocols, such as I2C, UART, or SPI. More specialized applications may be better served by one of the dedicated protocols such as CAN, LIN, MIL-1553, ARINC 429, MIPI, or Ethernet.

In some cases, though, design teams choose to roll their own customized protocol. A popular approach is to base a customized protocol on a *line code*, or as it's more broadly known, digital baseband modulation. Line coding consists of representing the digital signal to be transported by an amplitude- and time-discrete signal that is optimally tuned for the specific properties of the physical channel (and of the receiving equipment). The waveform pattern of voltage or current used to represent the 1s and 0s of digital data on a transmission link is called *line encoding*. Among the more common types of line encoding are NRZ (non-return-to-zero) and Manchester encoding.

NRZ encoding is a binary code in which 1s are represented by one significant condition (usually a positive voltage) and 0s are represented by some other significant condition (usually a negative voltage), with no other neutral or rest condition. The pulses have more energy than a RZ code. Unlike RZ, NRZ does not have a rest state. NRZ is not inherently a self-clocking signal, thus some additional synchronization technique (for example a run-length-limited constraint, or a parallel synchronization signal) must be used for avoiding bit slip.

For a given data-signaling rate (or bit rate), the NRZ code requires only half the baseband bandwidth required by Manchester code (the passband bandwidth is the same). An example of NRZ encoding is shown in Figure 1.

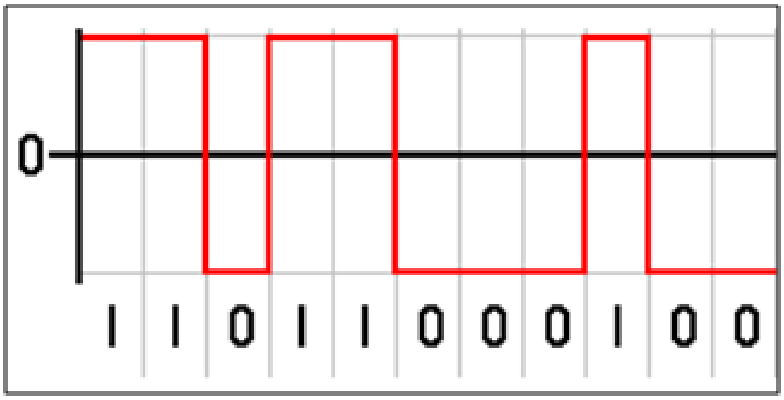

Figure 1: An example of NRZ encoding

Manchester encoding (also known as phase encoding) is a line code in which the encoding of each data bit has at least one transition and occupies the same time. Thus, it has no DC component and is self-clocking. This means that the signal may be inductively or capacitively coupled. It also means that a clock signal can be recovered from the encoded data. Figure 2 shows an example of Manchester encoding.

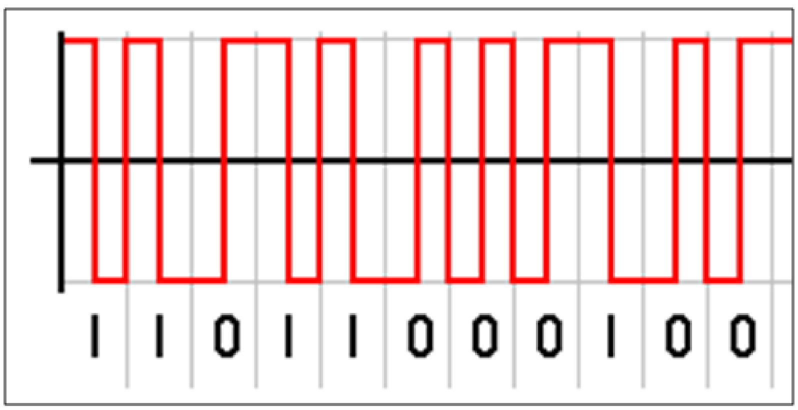

Figure 2: An example of Manchester encoding

Many of today's specialized data-communication protocols, such as the Digital Addressable Lighting Interface (DALI) for control of building lighting and the Peripheral Sensor Interface 5 (PSI5) used to connect sensors to controllers in automotive applications, are based on Manchester encoding. Likewise, the Single-Edge Nibble Transmission (SENT) protocol used for automotive sensor-to-controller links is based on NRZ encoding, as is the Controller Area Network (CAN) bus commonly used to enable communication between microcontrollers and other devices in automotive applications. In all of these cases, basic Manchester and NRZ schemes were modified to create the more complex, specialized protocols. Meanwhile, designers around the globe are designing and/or debugging their own unique protocols built on top of the Manchester and NRZ encoding schemes.

A convenient means of visualizing and debugging line encoding schemes is with an oscilloscope equipped with appropriate decoding software. Typically offered as options for oscilloscopes and available for many industry-standard protocols, such software greatly increases the ability to debug and analyze serial bus communications. Advanced software algorithms deconstruct the waveform into protocol information, and then overlay the decoded data on the waveform, so understanding messages is easy. These time-saving decode tools are very useful—unless there is no off-the-shelf decode software for a given serial bus. In such cases, designers have not benefited from the ability to decode data busses with an oscilloscope.

To serve the need for NRZ and Manchester decoding, Teledyne LeCroy now offers the industry's first NRZ and Manchester configurable protocol decoders. The decoders enable users to specify a broad range of physical characteristics for NRZ- or Manchester-encoded signals, which makes short work of analysis or custom and/or proprietary protocols based on those generic encoding styles.

A Look into How NRZ/Manchester Decode Software Functions

The main attraction of using NRZ or Manchester line encoding for building custom protocols is the degree to which the line codes can be modified. Designers can change a myriad of parameters in the physical layer to customize the implementation. This flexibility demands similar capabilities in the protocol decode software on the oscilloscope.

Thus, the user must configure a number of controls in the decoder's user interface, making it possible for the general algorithms to execute on his particular signal. This process requires a little more knowledge of serial data-encoding logic than other protocol decoders, but the explanations below should shed some light on the procedure. Once the settings have been determined for a given signal, they can be stored and recalled later, when analysis on the same signal is required.

The Underlying Methodology

The fundamental methodology behind the decoder is a three-step model. The signal needs to be decoded as:

- Burst(s) first with their constituent transitions, based on the bit rate and the timeout definition.
- Bursts are separated by "holes" or, electrically speaking, quiescent times on the transmission line.
- Then each burst is sliced into bits using the transitions, the bit rate and polarity, and the type (either NRZ or Manchester).
- Finally, the bursts of bits can be converted to words using the bits previously extracted, and based on grouping rules.

The bit conversion will not be functional before the bursts are correctly separated with their transitions, and the word extraction will not function correctly until the bits are properly decoded. The process is a three-level stepwise method.

The tuning of the Decoder must therefore be conducted in this order: **Transitions → Bits → Words**.
The following material will guide users through every step of the process, beginning at the Transitions level and ending at the Word level.

The Decoder in Action

After acquiring an NRZ or Manchester burst (or frame of relevant data) with the oscilloscope, stop the acquisition and adjust the oscilloscope so that the burst is well centered on the display with a good amount of idle segment before and after its duration.

With a burst acquired and positioned on the display as in Figure 3, it is ready for interpretation using either the Manchester or the NRZ decoder. The first step is identification of the burst controls, followed by extraction of the bits and their grouping into words. Incidentally, the words “burst,” “frame,” and “packet” are used interchangeably in this document as they are in technical literature.

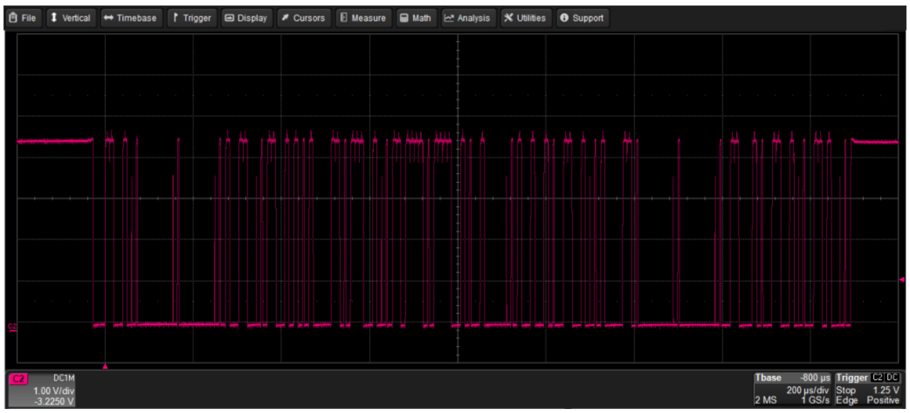

Figure 3: A burst of raw data

Another aspect of this process is the decoding of many packets into the same record, therefore allowing the observation of the encoded data values over a period of time. The decoder settings determined on a few packets will be reused when handling many packets.

Before decoding, first become familiar with the decoder’s user interface. In the Serial Decode dialog, shown in Figure 4, select the signal source (in this example, C1), and the protocol, which in this case is Manchester.

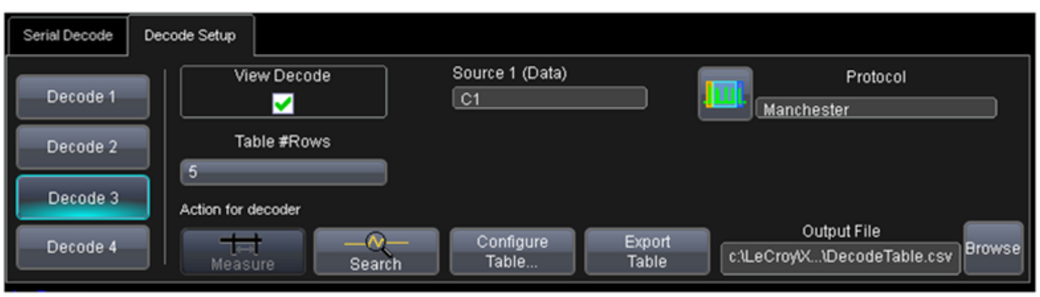

Figure 4: Signal source and protocol selection dialog

After selecting the Manchester protocol, three tabs will appear in the right side of the Serial Decode dialog: Basic, Decode, and Level (shown in Figure 5). Various values must be set in these three tabs for proper decoding of the data burst. When working on a given signal, some of the values in the tabs will not change because they are primary properties of the signal, such as bit rate or polarity. Other values will be tuned to obtain optimal results.

Figure 5: The three tabs (Basic, Decode, and Levels) driving Manchester bus decode

The Basic Tab: Bit-Level Decode

The Basic tab presents all the fundamental controls necessary to allow proper bit-level decoding. First, set the bit rate of your signal. If the value is unknown, use the cursor readouts, on a single bit or a sequence of bits, to determine the exact bit rate of the signal (another option may be to use the oscilloscope's measurement menu to directly measure the bit rate). The value should be correct within about 5%. Note that a mismatched bit rate will cause various confusing side effects on the decoding, so it is best to correctly adjust this fundamental value. Bit rates can be selected from 10 bits/s to 10 Gb/s.

The idle state complements the timeout value. To declare that a new burst is to be started, the decode algorithm looks at the time elapsed between two consecutive transitions and the state of the idle level between these transitions. This mechanism allows a precise definition of what the separation gap should be between two bursts. In most cases the idle state is specified and therefore provides an additional condition to the timeout to define the burst start. If this distinction is not desired, select "Don't care" in the idle state popup box.

The timeout separating bursts can be defined in bits or seconds. Both methods are equivalent in terms of their results, but depending on the context, the protocol specifications, or the user preference, one or the other representation might be chosen. Note that regardless of the timeout units selected, the allowable timeout range is from 1 to 100 bits.

The polarity governs the conversion of the physical signal transition into a logical bit state. When Falling=0 is chosen, a falling edge through the threshold level is decoded as a logical 0, whereas a rising edge through the threshold level is decoded as a logical 1. Figure 6 shows an example:

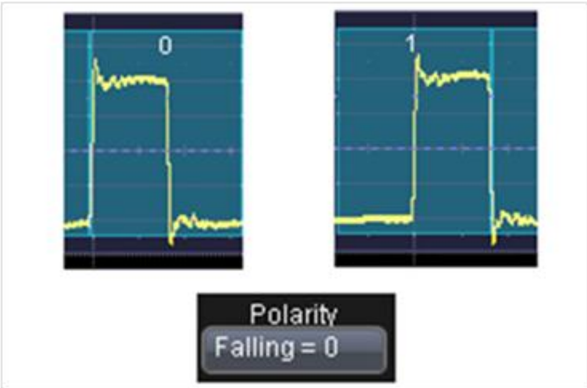

Figure 6: Manchester physical-to-logical mapping, case: Falling = 0

When Falling=1, the opposite logic is applied, leading to the following image of the low-level decode annotation (Figure 7):

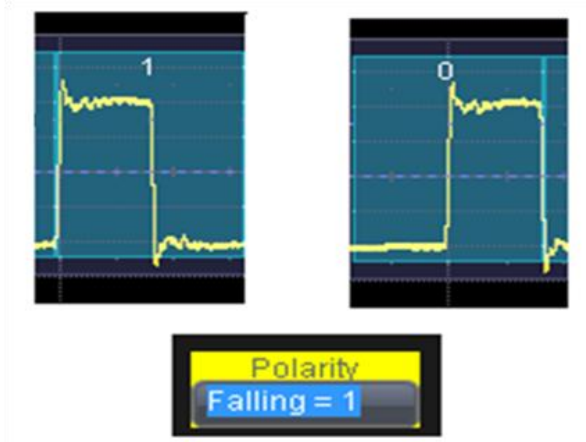

Figure 7: Manchester physical-to-logical mapping, case: Falling = 1

Verifying the Bit-Level Decode

Setting all of the Basic tab's values accomplishes bit-level decoding on the source trace. Note that by default the data mode is set to bits (on the Decode tab, covered below), so that the setting of the second tab does not matter for the initial bit-level decode. Figure 8 shows a correct decoding in which the bit transitions are all aligned with the signal transitions and the logical interpretation of the bits is consistent with the physical level. In this case, physical High equals logical 1.

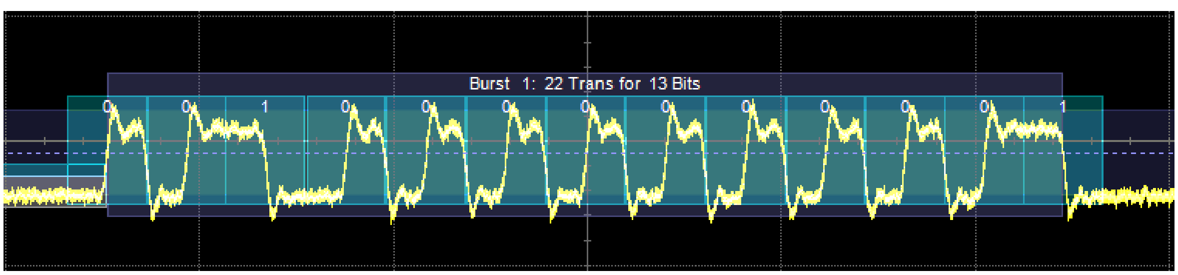

Figure 8: Correct decoding of a Manchester signal at 125 kb/s

Note that decoding at an exact multiple of the bit rate may appear correct but would not allow further interpretation of the words. Be suspicious when the bits are not aligned with the transitions, or if there are gaps between bits.

The Decode Tab: Moving Up to Word Level

Bit-level decoding can be taken further so that some transitions are skipped and subsequent bits are grouped into words. Words may be interpreted with least-significant bit (LSB) first or most-significant bit (MSB) first.

Figure 9 shows the Decode tab when data mode is “Bits”. In this mode, most of the controls are greyed out and are inactive. Selecting Words as the data mode activates those controls.

Figure 9: The Decode tab in Bits data mode (left) and in Words data mode (right). Changing the mode to Words activates the greyed-out parameters seen in Bits mode

The Data Mode

The Data Mode control drives the desired level of decoding. Initially, ensure that “Bits” is selected. “Bits” is also the default value when starting the decoder. After reviewing the low-level decoding in bits in the previous section, there are a few more items which concern bit-level decoding before considering word-level decoding.

First Transition Used (FTU)

Many Manchester or NRZ encoding schemes use a preamble, a synchronization sequence, or a voluntary Manchester violation. The first transition used (FTU) can be used to start the decoding after the violation, where the real data payload starts. It avoids the intricacies of dedicated protocols in the initial segment of each packet. The default setting for the FTU is zero as seen in Figure 9, but it can vary in increments of 1 to a maximum of 400.

Bit Stretch Tolerance

The Manchester bit slicer hops from mid-bit to mid-bit. However, due to hardware or signal propagation issues, the mid-bits might not be perfectly equidistant. In this case, the tolerance can be manually increased to attempt to decode jittery signals. Conversely, it can be decreased until the decoding starts showing anomalies, to assess the stability of the mid-bit distribution.

Decoding into Words

When a correct bit-level decoding has been achieved, it is time to consider higher-level concepts. Conversion of burst bits into words is achieved by selecting Words as the data mode. Selecting this mode makes all of the fields accessible. Unlike decoding into bits, the conglomeration of bits into words no longer depends upon the physical layer. In other words, the bits can be grouped to form words, regardless of their origin (NRZ, Manchester, or other). As a consequence the mechanisms described below apply to both the NRZ and the Manchester decoders.

The tools offered by the decoders allow grouping of bits into Sync bits, PrePad, Data Bits, and PostPad categories. Figure 10 shows an example with corresponding annotations. The PrePad bits are used to group information preceding the Data Bits. There can be from zero to 32 PrePad bits, which might be used to group address bits, preambles, subaddresses, and so on.

Figure 10: This zoomed segment shows one of four words in an NRZ signal. As specified in the Grouping of Bits into Words section of the corresponding Decode tab shown below the waveform itself, the word comprises 2 sync bits (bracketed in white), 13 PrePad bits, 32 data bits, and 6 PostPad bits

The Viewing control offers a choice of hexadecimal, ASCII, or decimal for viewing of the PrePad, Data Bits, and PostPad, both in the table and the annotations on the trace. This parameter has no impact when Data Mode is Bits and is therefore grayed out.

When decoding in Words, the Bit Order control determines whether words are converted with the MSB first or the LSB first.

With the Sync Bits control, users choose at which bit the packetizing should start. The algorithm begins with the Sync Bits and groups bits into the following three fields: PrePad, Data Bits, and PostPad. It then restarts with the PrePad of the next sequence. There can be from zero to 100 Sync Bits.

Data bits are the number of bits grouped together to form a single word. The number of bits per words can range from 1 to 32 in steps of 1.

The PostPad bits serve to group information following the Data bits. There can be from zero to 32 PostPad bits. PostPad bits may be used to represent a CRC, a checksum, a value or any other protocol construct.

Comparison of a Signal Decoded as Bits or Words

The annotated screen image in Figure 11 provides a time-aligned comparison of the same signal decoded as bits at top and as words at bottom. Here again the action of every control can be verified and explained.

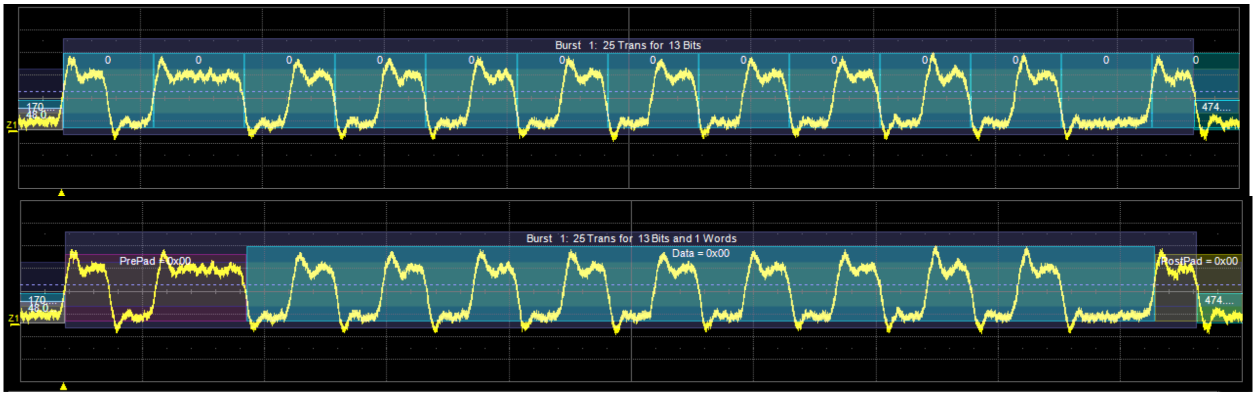

Figure 11: Comparison of Bit and Word Decode (top and bottom, respectively) on the same Manchester signal

Again, switching between the two views of the same signal as shown is simply a matter of switching the Data Mode from Bits to Words on the Decode tab. The numbers seen at the right and left of the decoded signal represent the timeout definition and interframe gap.

The Data Table

The Manchester and NRZ decode software provides a configurable tabular view of relevant data as shown in Figure 12. The table and waveform view are interactive, so tapping a given line in the table will highlight the corresponding portion of the waveform. If a zoomed view of that portion is not already open, the action of tapping the line will open a zoomed view.

Figure 12: An interactive, configurable data table provides details about each data burst in the waveform

Referring back to Figure 4 and the Decode Setup tab, the data table can be configured with a maximum of 20 rows. The contents of the table are configured using the Configure Table function, shown in Figure 13. The table also can be exported as a .csv file for future reference, printing, or archiving.

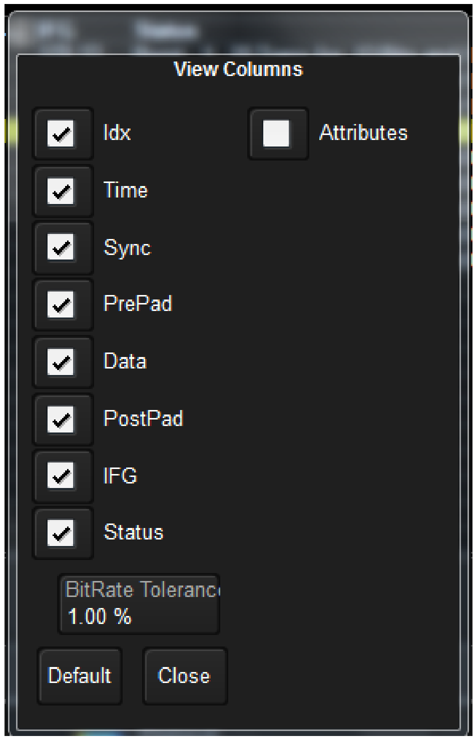

Figure 13: The Configure Table function on the Decode Setup tab brings up the View Columns dialog box, where users can decide what data relevant to the waveform under test will be display

Conclusion

Teledyne LeCroy's configurable protocol decode capabilities for signals using Manchester and NRZ encoding schemes enables unprecedented debug capabilities for systems using a variety of protocols, both standard and custom. A good understanding of the software's functionality leads to quick and accurate decoding of signals based on these schemes.