N.D. Mehta, Assistant Professor, VGEC Chandkheda

Digital Communication:

A **line code** is the code used for data transmission of a digital signal over a transmission line. This process of coding is chosen so as to avoid overlap and distortion of signal such as inter-symbol interference.

Properties of Line Coding

Following are the properties of line coding −

- As the coding is done to make more bits transmit on a single signal, the bandwidth used is much reduced.
- For a given bandwidth, the power is efficiently used.
- The probability of error is much reduced.
- Error detection is done and the bipolar too has a correction capability.
- Power density is much favorable.
- The timing content is adequate.
- Long strings of **1s** and **0s** are avoided to maintain transparency.

Types of Line Coding

There are 3 types of Line Coding

- Unipolar
- Polar
- Bi-polar

Unipolar Signaling:

Unipolar signaling is also called as **On-Off Keying** or simply **OOK**. The presence of pulse represents a **1** and the absence of pulse represents a **0**.

There are two variations in Unipolar signaling −

- Non Return to Zero (NRZ)
- Return to Zero (RZ)

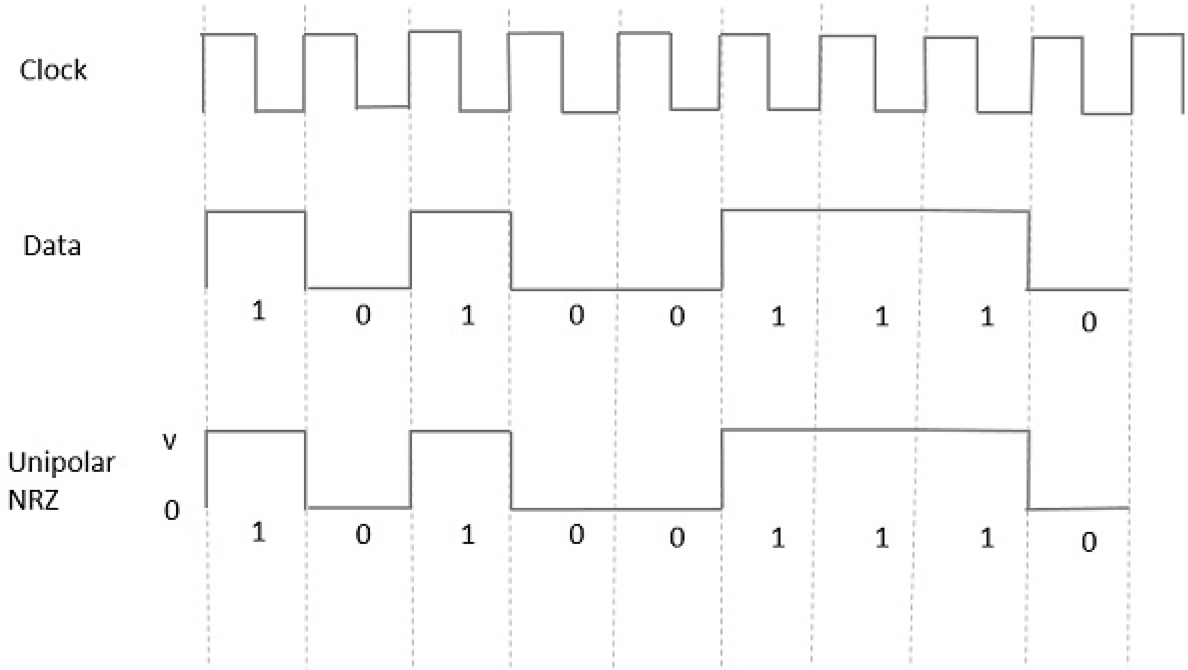
Unipolar Non-Return to Zero (NRZ)

In this type of unipolar signaling, a High in data is represented by a positive pulse called as **Mark**, which has a duration $\mathbf{T_0}$ equal to the symbol bit duration. A Low in data input has no

N.D. Mehta, Assistant Professor, VGEC Chandkheda

pulse.

The following figure clearly depicts this.



Advantages

The advantages of Unipolar NRZ are −

- It is simple.
- A lesser bandwidth is required.

Disadvantages

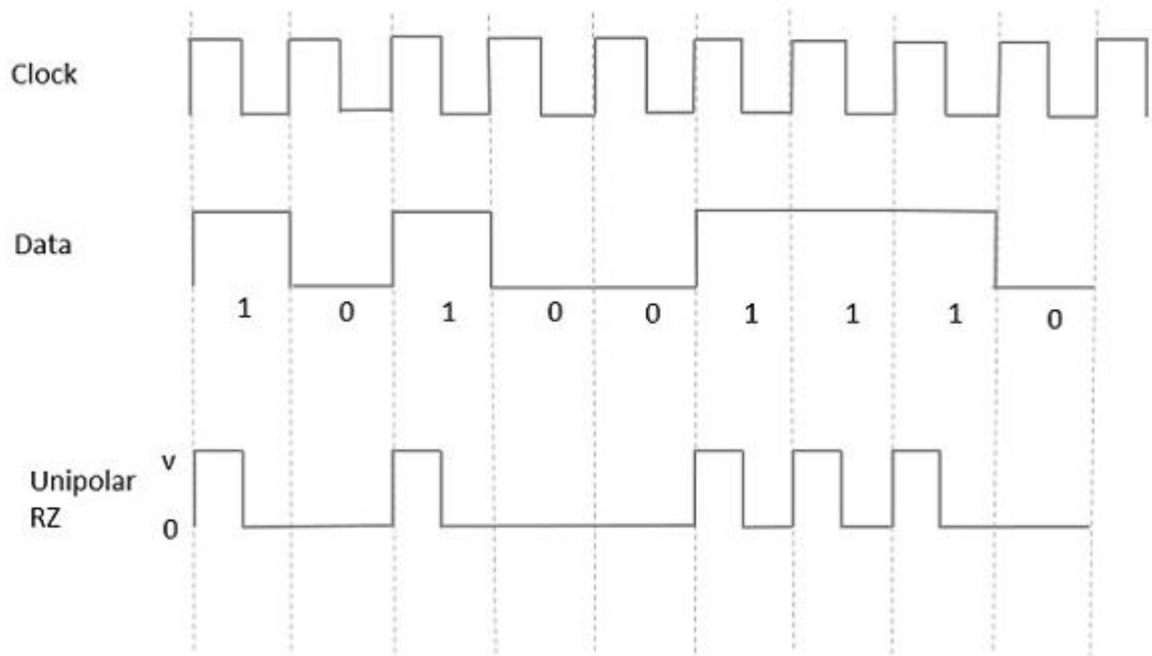
The disadvantages of Unipolar NRZ are −

- No error correction done.
- Presence of low frequency components may cause the signal droop.
- No clock is present.
- Loss of synchronization is likely to occur (especially for long strings of **1s** and **0s**).

Unipolar Return to Zero (RZ)

In this type of unipolar signaling, a High in data, though represented by a **Mark pulse**, its duration $\mathbf{T_0}$ is less than the symbol bit duration. Half of the bit duration remains high but it immediately returns to zero and shows the absence of pulse during the remaining half of the bit duration.

It is clearly understood with the help of the following figure.



Advantages

The advantages of Unipolar RZ are −

- It is simple.
- The spectral line present at the symbol rate can be used as a clock.

Disadvantages

The disadvantages of Unipolar RZ are −

- No error correction.
- Occupies twice the bandwidth as unipolar NRZ.
- The signal droop is caused at the places where signal is non-zero at 0 Hz.

Polar Signaling

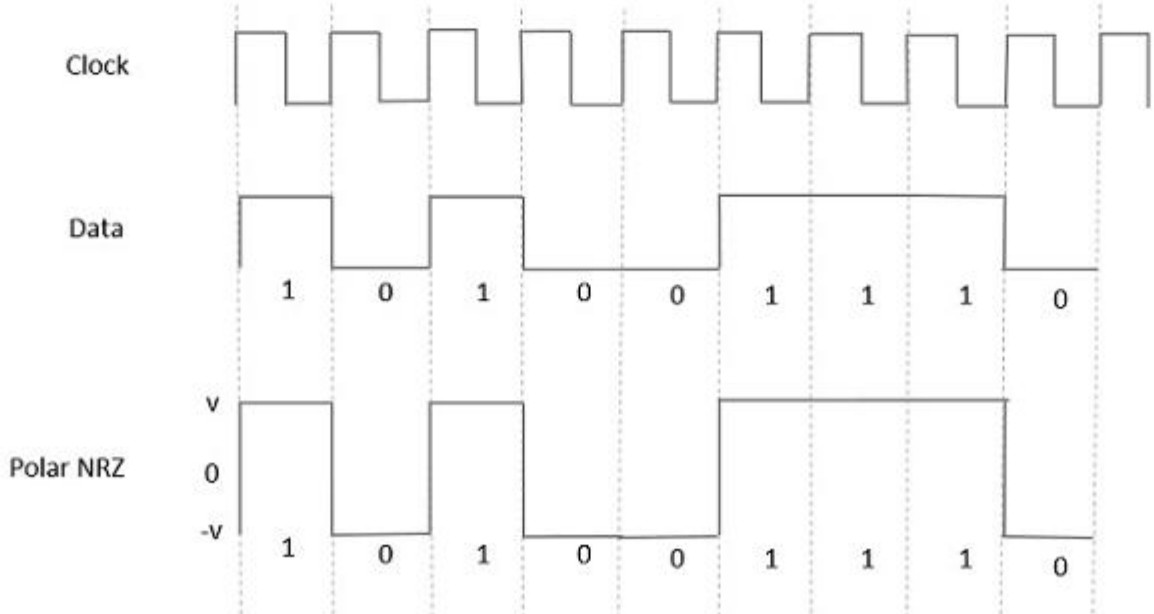
There are two methods of Polar Signaling. They are −

- Polar NRZ
- Polar RZ

Polar NRZ

In this type of Polar signaling, a High in data is represented by a positive pulse, while a Low in data is represented by a negative pulse. The following figure depicts this well.

N.D. Mehta, Assistant Professor, VGEC Chandkheda



Advantages

The advantages of Polar NRZ are −

- It is simple.
- No low-frequency components are present.

Disadvantages

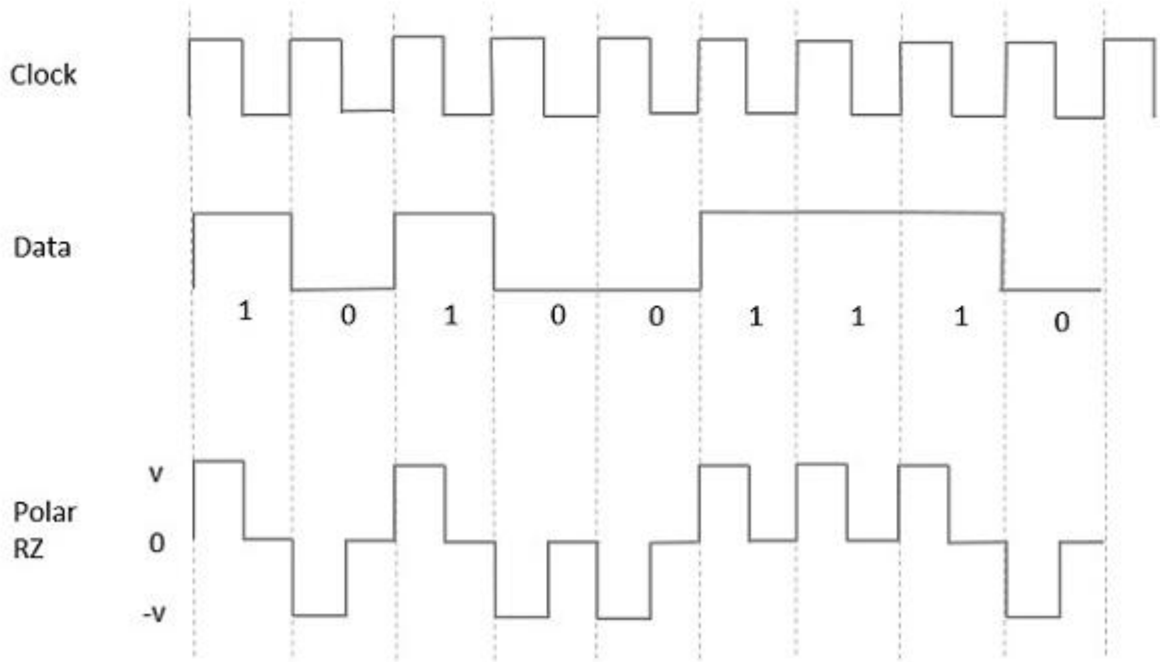
The disadvantages of Polar NRZ are −

- No error correction.
- No clock is present.
- The signal droop is caused at the places where the signal is non-zero at **0 Hz**.

Polar RZ

In this type of Polar signaling, a High in data, though represented by a **Mark pulse**, its duration $\mathbf{T_0}$ is less than the symbol bit duration. Half of the bit duration remains high but it immediately returns to zero and shows the absence of pulse during the remaining half of the bit duration.

However, for a Low input, a negative pulse represents the data, and the zero level remains same for the other half of the bit duration. The following figure depicts this clearly.



Advantages

The advantages of Polar RZ are −

- It is simple.
- No low-frequency components are present.

Disadvantages

The disadvantages of Polar RZ are −

- No error correction.
- No clock is present.
- Occupies twice the bandwidth of Polar NRZ.
- The signal droop is caused at places where the signal is non-zero at **0 Hz**.

Bipolar Signaling

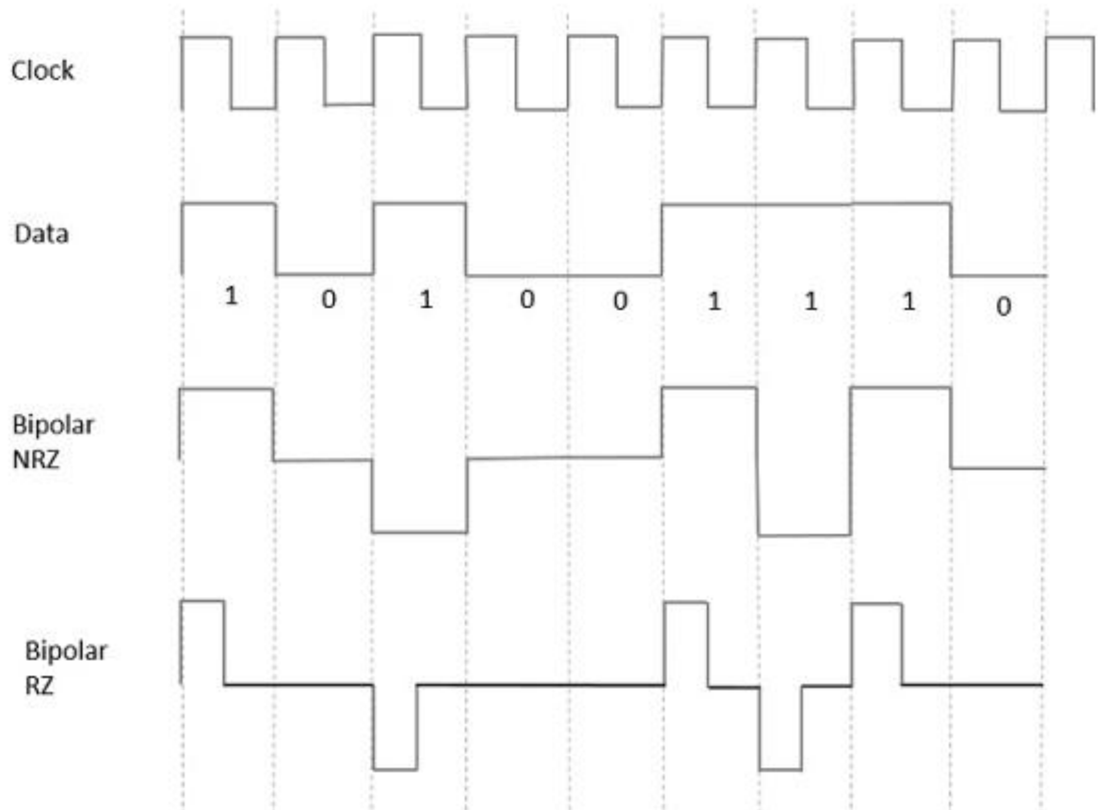
This is an encoding technique which has three voltage levels namely **+, -** and **0**. Such a signal is called as **duo-binary signal**.

An example of this type is **Alternate Mark Inversion (AMI)**. For a **1**, the voltage level gets a transition from + to – or from – to +, having alternate **1s**to be of equal polarity. A **0** will have a zero voltage level.

Even in this method, we have two types.

- Bipolar NRZ
- Bipolar RZ

From the models so far discussed, we have learnt the difference between NRZ and RZ. It just goes in the same way here too. The following figure clearly depicts this.



The above figure has both the Bipolar NRZ and RZ waveforms. The pulse duration and symbol bit duration are equal in NRZ type, while the pulse duration is half of the symbol bit duration in RZ type.

Advantages

Following are the advantages −

- It is simple.
- No low-frequency components are present.
- Occupies low bandwidth than unipolar and polar NRZ schemes.
- This technique is suitable for transmission over AC coupled lines, as signal drooping doesn’t occur here.

- A single error detection capability is present in this.

Disadvantages

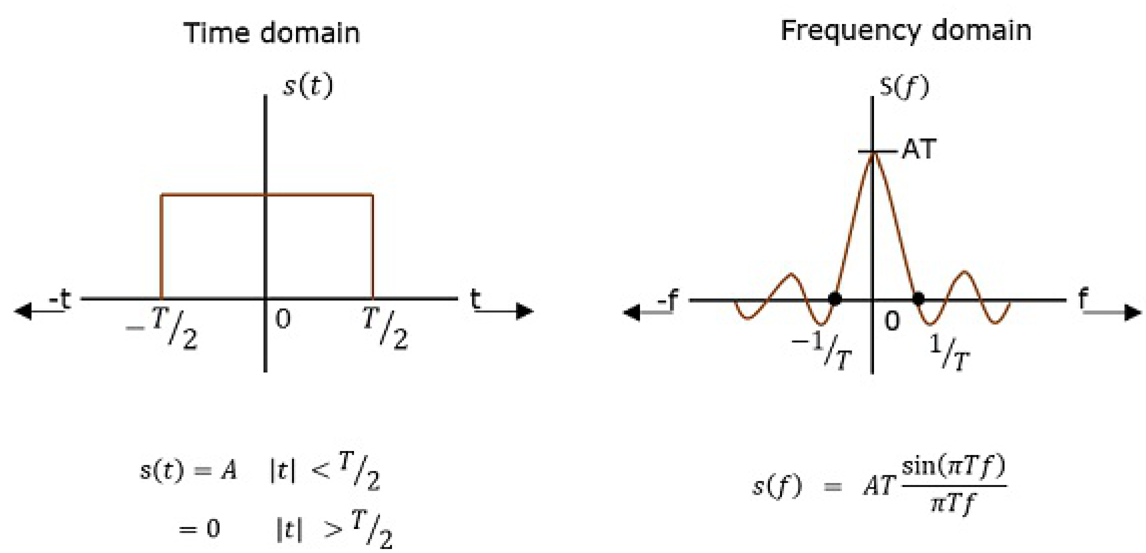
Following are the disadvantages −

- No clock is present.
- Long strings of data causes loss of synchronization.

Power Spectral Density

The function which describes how the power of a signal got distributed at various frequencies, in the frequency domain is called as **Power Spectral Density (PSD)**.

PSD is the Fourier Transform of Auto-Correlation (Similarity between observations). It is in the form of a rectangular pulse.



PSD Derivation

According to the Einstein-Wiener-Khintchine theorem, if the auto correlation function or power spectral density of a random process is known, the other can be found exactly.

Hence, to derive the power spectral density, we shall use the time auto-correlation (Rx(τ))(Rx(τ)) of a power signal x(t)x(t) as shown below.

Rx(τ)=limTp→∞1Tp∫Tp2−Tp2x(t)x(t+τ)dtRx(τ)=limTp→∞1Tp∫−Tp2Tp2x(t)x(t+τ)dt

Since x(t)x(t) consists of impulses, Rx(τ)Rx(τ) can be written as

Rx(τ)=1T∑n=−∞∞Rnδ(τ−nT)Rx(τ)=1T∑n=−∞∞Rnδ(τ−nT)

Where Rn=limN→∞1N∑kakak+nRn=limN→∞1N∑kakak+n

Getting to know that Rn=R−nRn=R−n for real signals, we have

Sx(w)=1T(R0+2∑n=1∞RncosnwT)Sx(w)=1T(R0+2∑n=1∞RncosnwT)

Since the pulse filter has the spectrum of (w)↔f(t)(w)↔f(t), we have

sy(w)=|F(w)|2Sx(w)sy(w)=|F(w)|2Sx(w)

=|F(w)|2T(∑n=−∞∞Rne−jnwTb)=|F(w)|2T(∑n=−∞∞Rne−jnwTb)

=|F(w)|2T(R0+2∑n=1∞RncosnwT)=|F(w)|2T(R0+2∑n=1∞RncosnwT)

Hence, we get the equation for Power Spectral Density. Using this, we can find the PSD of various line codes.

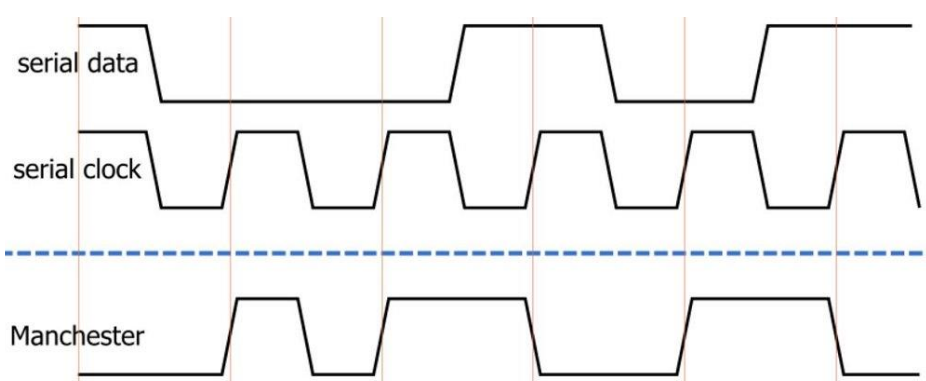
DC Avoidance

With complex systems, especially those involving high voltages, it is not always easy to ensure that the common-mode voltage of a transmitted signal is compatible with the receiver's acceptable common-mode range. (This is an issue even when using a differential standard such as RS-485.) Another concern is fault currents—DC coupling offers no protection against dangerous long-term currents resulting from a short circuit.

Thus, AC coupling is a straightforward way to mitigate inconveniences and risks associated with common-mode voltages and failure modes.

The Manchester Encoding :

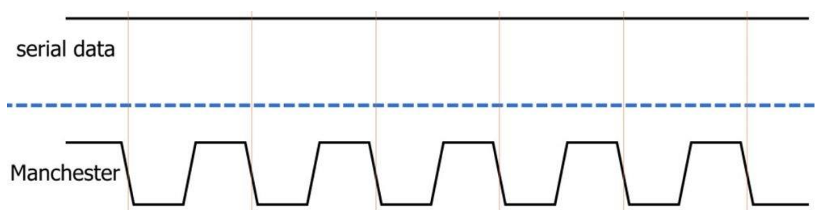
The fundamental idea behind Manchester encoding is the following: we can use voltage *transitions*, instead of voltage *levels*, to represent ones and zeros. Consider the following diagram:



In the upper part of the diagram we have a standard digital interface consisting of a data signal and a clock signal.

In the lower part of the diagram is a Manchester-encoded signal for the same data. Notice how the transitions occur in the middle of the standard-data-signal logic states (in other words, the Manchester transition is aligned with the clock edge that would be used to sample the data). Notice also that a logic-high bit always corresponds to a high-to-low transition, and a logic-low bit always corresponds to a low-to-high transition. (You could also use a low-to-high transition for logic high and a high-to-low transition for logic low; the important thing is that the receiver circuitry knows which format to expect.)

It is immediately clear that the AC-coupling problem is eliminated: every bit requires a transition, and therefore the data signal will never remain at logic low or logic high for an extended period of time. This is evident in the following diagram, which shows a standard digital signal for binary 111111 and a Manchester-encoded signal for the same binary sequence.



The synchronization issue is a little less straightforward because we still need to somehow extract the clock from the signal; nevertheless, we can intuitively see that the regularity of the transitions provides information about when the data signal should be sampled.

The previous diagram also demonstrates a nontrivial disadvantage of Manchester encoding: the data rate is cut in half relative to the bandwidth of the data signal. A Manchester-encoded signal needs a transition for every bit, which means *two* Manchester logic states are used to convey *one* standard logic state. Thus, twice as much bandwidth is needed to transfer data at the same rate.

This may not seem like a problem—why not just use a higher-frequency signal? Well, if signal bandwidth is the limiting factor in how quickly data can be moved from transmitter to receiver, and if you already are at the maximum data rate, you can’t increase the signal frequency by a factor of two; instead, you must reduce the data rate by a factor of two.

Conclusion

You now know what Manchester encoding is and why it is beneficial, despite the potentially reduced data rate and the additional circuitry or firmware needed to generate and interpret the Manchester-encoded data. A future article will provide specific implementation details for those who want to explore Manchester's more practical side.