

RUDI VAN DRUNEN

signals



Rudi van Drunen is a senior UNIX systems consultant with Competa IT B.V. in The Netherlands. He also has his own consulting company, Xlexit Technology, doing low-level hardware-oriented jobs.

rudi-usenix@xlexit.com

WHEN TRANSMITTING INFORMATION, we do not need to transmit power (covered in my previous article) but information and data expressed with signals. In this article I explain how data gets encoded, sources of noise, and various solutions. We will primarily examine Ethernet as an example.

Encoding Information

In the digital world we distinguish two discrete values a signal can have, 0 or 1, and these values correspond to certain voltages. The mapping between the voltage on a point and the digital value it represents is determined by the electronics family the circuit is built in and the supply voltage [1]. This chart gives the HIGH (often “1”) and LOW (often “0”) values and thresholds for most logic families. We can learn from this that the lower the supply voltage is, the lower the voltage margin between a HIGH and a LOW signal is.

The larger the (voltage) margin between the “1” and “0” signal, the less susceptible the system is to external noise, but the more power is needed in the driver electronics to drive the circuit. There is a trend to lower the voltages in systems to be more energy (and heat) efficient, but this will make the system less immune to outside factors.

For external interfaces the levels can be completely different. For example, the serial RS 232 interface is originally specified with a “0” value corresponding to 12 V and a “1” value corresponding to -12 V. Nowadays most systems are able to receive the 12 .. -12 V levels, but send only 5 and 0 V levels.

Information can also be encoded in the current, as used in RS 422 or MIDI interfaces. Here, when a current is flowing a “1” is represented, whereas no current represents a “0” value.

Analog Versus Digital

With simple analog signals the information is encoded in the value (mostly the voltage) at a given time. More complex signals (often modulated signals) can have different encodings. For example, with the FM radio the information is encoded in the frequency of the analog signal. With analog TV the color and intensity are complexly encoded in phase differences. Another encoding example can be found in process control systems, where a current loop interface is commonly used. This in-

terface uses current (4 to 20 mA) to encode the (analog) sensor value. An advantage of current loop interfaces is that they can operate over moderately long distances (>10 miles).

Here, note the difference between the current loop interfaces in the analog and digital worlds. The analog interface uses the amount of current to show the value; the digital interface uses current or no current to show the two values.

Conversion

Conversion from analog to digital is done using an analog to digital converter (ADC). These components or circuits have an analog voltage input and some way to output the signal digitally. We find these components, for example, in your sound input circuitry or the environmental sensor system.

The other way around, converting digital to analog, is done by a digital to analog converter (DAC), as found in the sound output or your (analog) video (VGA) output.

Transmission

As we transmit an analog value using a conductor that (continuously) carries a certain voltage to ground, to transmit the very same amount of digital information we might need *more than one* conductor, each representing a bit. The number of bits we need is determined by the accuracy and resolution we need to express the value. These bits can be transmitted either serially or in parallel, where you need as many signal lines as bits you want to transmit.

If you need to transport temperature information, for example, 70 def F (with a resolution of 1 deg F), from a sensor to a computer system, you can encode this in analog by sending a voltage of 70 mV through an analog channel (using a ground and a signal wire) or digitally by using 8 bits, which gives you a range of values from 0 to 255 ($= 2^8 - 1$). (You can do with less; the number of bits you need is determined by the range of the information you want to send, the resolution, and the encoding.) The value of the 8-bit word determines the temperature value.

To do this in parallel you need eight signal lines carrying 0100 0110 (the actual voltage on the lines is not relevant) or just one line where you send a bit, say, each second, by putting the appropriate voltage on the signal line (see also Figure 1).

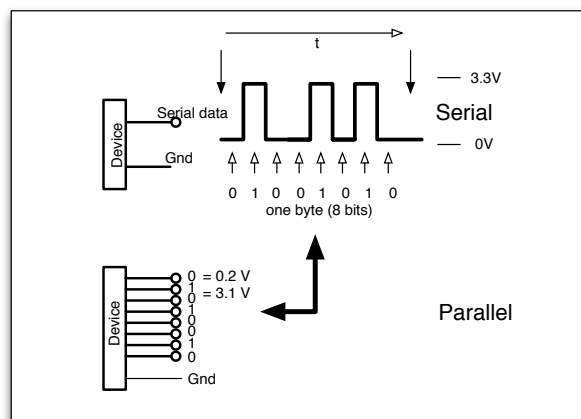


FIGURE 1: SERIAL VS PARALLEL

In general, analog signals are more susceptible to external interference, such as noise or hum, as the voltage directly relates to the value. With digital signals we have defined a noise margin. Even though the absolute value of the voltage is not relevant, it should be within boundaries defined by the logic family or standard for that interface. If you transmit serially, the time factor comes into play; if in the transmission the timing of the signal changes, the read value might change.

In the previous example this might be the case if, due to the nature of the transmission channel, a “0” takes longer to transmit than a “1.”

Outside Influence

All digital signals with a serial nature or changing parallel values generate noise. As the value of the signal changes rapidly (from a logic 1 to a logic 0, or the other way around), the cable carrying this signal generates noise in the electric and magnetic spectrum.

A source of interference is called Radio Frequency (RF) noise. We experience this if we put a radio receiver next to a piece of computing equipment. You then can “tune in” to a number of different sounds that are generated from the computing equipment. In the computer we find all kinds of signals that have a square wave nature.

The Fourier series tells us that we can see a square wave as the sum of a number of sine waves (harmonics) all of different frequencies. For example, a square wave with a frequency of 1 MHz can be described by adding up sine waves of 1, 3, 5, 7, 9 . . . MHz with different amplitudes. So, the square wave voltage over time can be described as:

$$U(t) = \frac{4}{\pi} (\sin(2\pi ft) + \frac{1}{3} \sin(6\pi ft) + \frac{1}{5} \sin(10\pi ft) + \dots)$$

As we use radials, the factor 2π is used in this expression.

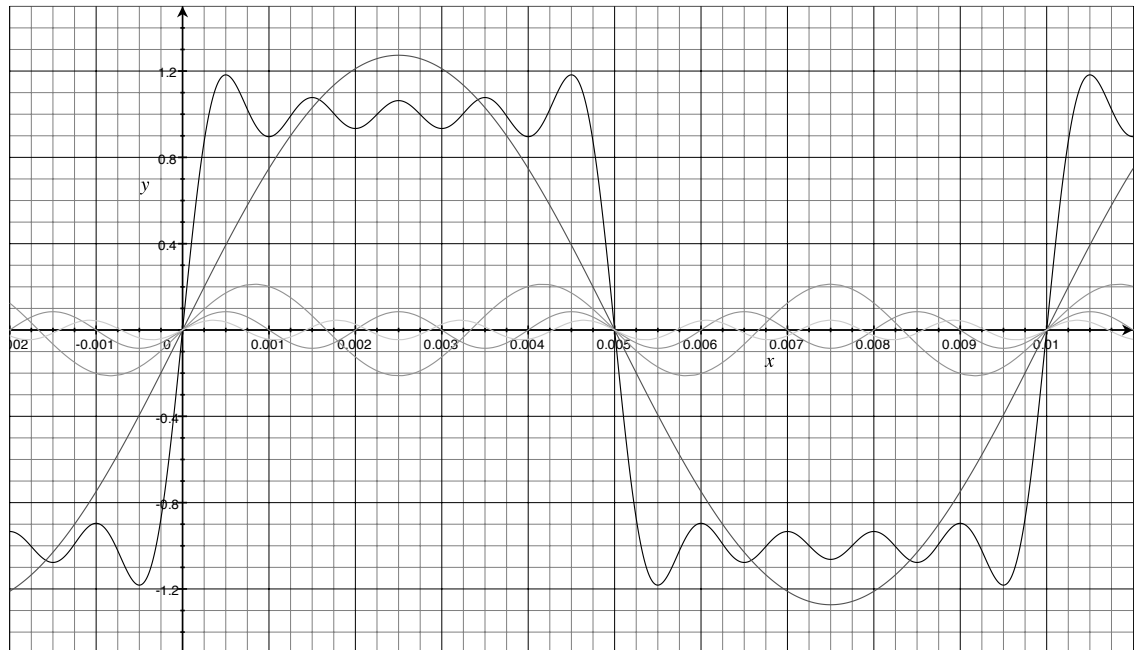


FIGURE 2: BUILDING A SQUARE WAVE FROM SINE WAVES

Figure 2 (preceding page) shows a square wave built from different sine waves. In our example, with a square wave of 1 MHz you would hear signals on your radio receiver at 1, 3, 5 . . . MHz when tuned in.

The higher-frequency harmonics are decreasing in amplitude. Compared to the ground frequency (the frequency of the square wave), the third harmonic is 1/3 of the ground wave amplitude, the 5th harmonic is 1/5 of the amplitude.

The FCC (or in Europe, the ETSI), which regulates the radio spectrum, tests all equipment in order to be sure that the (RF) radiation that originates from this equipment is within the boundaries of the legislation that prevents interference.

An example of this interference is Gigabit Ethernet, which is notorious for polluting the RF spectrum. We can lessen the RF noise generated by Gigabit interfaces and cabling (1000 BASE-T) by using shielded patch cables. Another way to reduce RF pollution by Gigabit Ethernet is to use fiber connections (1000 BASE-X) where possible.

The RF noise signal will, for example, prevent your GPS (or long wave) antenna for your time server from working and distort your cordless phone or your paging system when these are close to a non-shielded network cabinet.

Not only does equipment itself generate RF radiation, but the cabling used generates or can pick up a considerable amount of RF noise, and also works as an antenna amplifying this noise. To minimize this effect, new (signal) cables often have a ferrite coil integrated into them. You can see that as a big lump in the cable. The ferrite coil works as a choke to prevent part of the RF noise from radiating out of the cable by virtually creating a barrier for high-frequency signals.

Preventing the noise from getting out of the system or cable is one thing, but there is still so much ambient noise getting into (or out of) our systems and cabling that we need measures to prevent it from disturbing electronics and communication channels. There are a number of ways to overcome the influence of external factors in transmission channels. These techniques are applied in various interface standards in our digital world as well. We will discuss these here.

Noise

When you need to transport a digital signal over a long distance in a harsh environment, you want to prevent the signal from becoming too noisy. Therefore you want to minimize the influence of external factors, such as fluctuating magnetic or electric fields, which cause voltage noise to be induced in the conductors of the cable. If, for example, you run a signal cable parallel to a power cable, you will see when there is a current flowing through the power cable for there will be a hum caused by an induction on the signal cable, which may distort the signal. In the analog world, the signal might be coming from a sensor (maybe your environmental monitoring sensors); in the digital world, it can be a systems interconnect of some sort.

Mobile phones are another example of noise generated by external (electric/magnetic) fields. If the transmitter of a mobile phone is communicating with a base station, for example, the electric field will generate heavy noise in all conductors in the vicinity. You will notice this in small signal analog systems, but it might disturb digital systems as well when the (voltage) margin between a 0 and a 1 is small.

To prevent the build-up of the hum voltage, you can start by not running signal cables parallel to power cables and by putting a metal shield around

the signal-carrying conductors, actually putting the signal carrying conductor or conductors in a “Faraday cage” [2]. When you add the conductive shielding, you need to be aware of ground loops (described in the first article of this series). For example, you might use FTP (foil-shielded twisted pair) cables instead of UTP cabling.

There is also a technique where you use two conductors to transmit the signal and drive them in an inverse way with the signal to transmit. Both conductors will pick up about the same stray noise in transit. At the end, both signals electronically contain the same output as the input signal, but the stray noise will be cancelled out (see Figure 3).

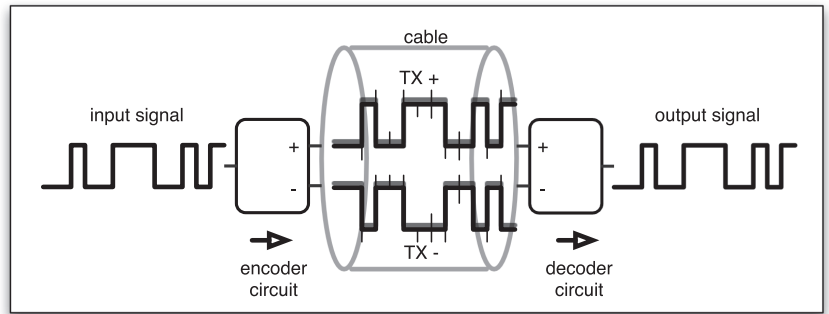


FIGURE 3: SYMMETRIC SIGNALS

This is the way Ethernet transmits the signals. To drive and receive the signals, small transformers are used. These transformers make sure that the inverse driving and subtraction of signals take place, and, by not making an electrical circuit between both systems, they also prevent ground loops from occurring (see Figure 4).

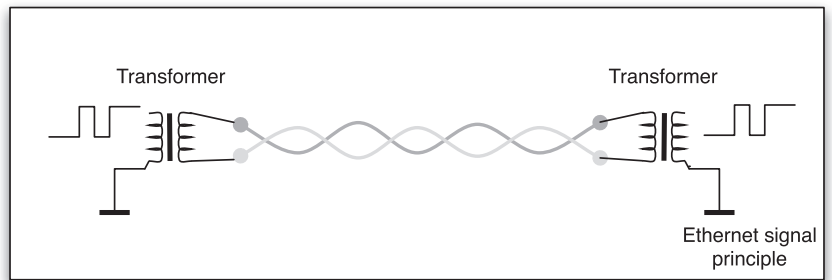


FIGURE 4: ETHERNET CABLING

The twisting of the conductors in an Ethernet cable will amplify the fact that both conductors pick up the same noise.

Differential SCSI, like Ethernet, transmits symmetrically. SCSI also protects against external influences by separating the data conductors with a ground conductor, which acts as a shield against noise or crosstalk (signals in the same cable influencing each other).

Timing Issues

Now that serial data transmission (e.g., USB or SATA/SAS) is replacing older parallel interfaces such as the Centronics printer interface (IEEE 1248 [3]), ATA, or SCSI, another factor comes into play. In order to preserve the exact timing relationship between the encoded bits in the cable it is important both to encode the bits in a way that protects the signal from outside factors and to use good cabling.

You can encode the bit (1 or 0) into the value of the voltage or into the change of voltage. Serial connections always need some kind of synchronization between the sender and receiver in order to sample the voltage values at the same relative moment in time in both sender and receiver. Having your data encoded in a steady voltage necessitates more overhead in time, and thus channel capacity to add synchronization bits, than does encoding your data in the changes of voltage. With the latter you can intelligently extract synchronization from the data signal.

I will describe how cabling can affect timing of a signal. Figure 5 shows the equivalent circuit of a piece of signal cable. The cable can be thought of as an infinite number of series resistors and parallel capacitors. The better quality the cable has, the smaller the capacitance and inductance effects are.

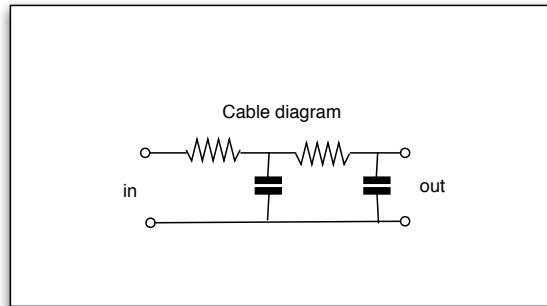


FIGURE 5: EQUIVALENT CIRCUITRY OF A CABLE

In Figure 6 we show the input signal of a cable and the output signal, in which, due to the nature of the cable, the input and output signals are quite different. There are delays (phase shift, skew) that are caused by the capacitors to be charged, and fields that have to be built up in the inductors. You also see overshoot and undershoot due to these effects. The bad thing here is that the effect is not symmetrical between the "0" and the "1" value, thus creating a different timing in the output as perceived by the electronics on the receiving end.

It is important to understand that there is often not only a delay/skew in the pulse on the output of the cable, but also an asymmetric effect changing the pulse length. The two combined can totally mess up the timing relationship in a system.

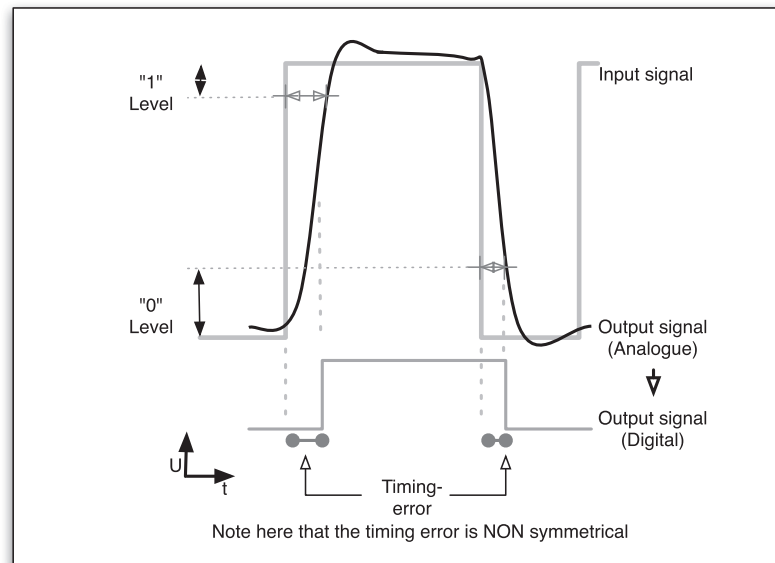


FIGURE 6: INTRODUCING TIMING ERRORS DUE TO BAD CABLING

If your data (information encoded in your signal) relies on exact timing, a short, good-quality cable is needed. A typical example can be found in the USB bus. If you make the USB connection longer than 5 m (18 ft.), data transfer will often not be reliable anymore, or the electronics will restrict the maximum transfer rate. In order to minimize these effects, the right length of high-quality, uncoiled, twisted-pair cable should be used. If you need longer transmission channels, select another technique or use regeneration boxes that receive and retransmit or decode and re-encode your information on the cable.

Note that this is also an issue present in parallel interfaces if we quickly transmit multiple words in sequence after each other.

In Figure 6 we also see that a bad cable can change the edges of the signal. This is particularly important with video interfaces. If the edges of the signal become less straight, you will see ghost images (shadow) on your screen. In comparison to analog video, this effect is much less present in digital video, since we digitize the signal first.

On this subject, we might explain the term *slew rate*. Slew rate is defined as the maximum rate of change of a signal, which in turn is defined by the environment. A cable can change the slew rate completely: when the cable is not able to let higher frequencies pass, the edges of the resulting signal on the output are not as steep.

$$SR = \frac{d(U_{out})}{d(t)}$$

The slew rate (SR) is defined in terms of units of voltage (volts, or V) per unit of time (milliseconds, or ms). On modern video cards we see slew rates of >200 V/ms.

Reflection

Another effect you might see in cabling is reflection. Each type of cable has a characteristic impedance (resistance to an AC signal). The material used and the way the cable is manufactured determine the characteristic impedance. If the impedance of the cable does not match the circuitry on both ends, you might end up with reflection. Reflection means that part of the signal is not being completely “absorbed” by the receiving end but is partly reflected by it back into the cable, thus distorting the signals.

For example, the characteristic impedance of a piece of (UTP) Ethernet cable is 100 Ω , so the receiving and transmitting circuitry need to be adapted to that in order not to create distortions in the signal. Abrupt changes in the nature of the cable—unevenly distributed twists or bends, for example (more important in other types of cable, mostly RF)—also create reflections and thus distortions of the signal.

In the early days this was extremely important in systems using coax-based, thin (10base-2) or thick (10base-5) Ethernet [4]. Coax-based Ethernet uses a cable with a characteristic impedance of 50 Ω connecting all devices in a long line with T-shaped connectors. On both ends of the line one needed a 50 Ω terminator plug to prevent signal reflections against an open end. If one of the terminators was not present the complete network failed, as the signals on the cable got too distorted to be used by the interfaces of the connected equipment.

In 10base-5 Ethernet (the thick yellow cable you had to use clamps, or “vampire taps,” with), you were only allowed to put an adapter to connect a machine to the cable every 2.5 meters (8.5 ft). This was required in order to

avoid signal distortion in the cable resulting from reflections caused by the tap.

SCSI is another interface requiring termination for proper operation. To prevent signal distortions due to reflections, one needs to terminate the SCSI bus. This can be done by a physical terminator on the cable end or by turning termination to “on” on the last device on the bus. The first device (the host) on the bus often has termination built in. SCSI terminators are just resistors connected to each data or handshake line and are commonly joined to signal ground.

By using reflection we can measure the length of a cable. This is done using a TDR (Time Domain Reflectometer). This device measures the time it takes a pulse to be reflected on the end of the cable and to arrive back at the sending end. The TDR can calculate the characteristics of the cable and its length. Often this function is integrated in a network test set.

General

All these factors get worse if the signal frequency (data rate) gets higher. At current data rates and frequencies used in systems that drop the supply voltage (thus lowering the noise margin between the 1 and 0 signals), external and internal (crosstalk) influence on signals becomes more important. Designers of printed circuit boards need to be aware of this, but the system designer connecting pieces of equipment also needs to be aware of the types and limitations of the interconnects.

Conclusion

With today’s faster systems, cable quality is crucial in keeping the timing relations of the signals within spec. Cable with low resistance and low capacitance will introduce less distortion in your signal. It is also important to align cables properly and not to use cables that are too long. Finally, make sure your transmission channel (the cable) is of the correct type and is terminated on both sides, to prevent signal distortion due to reflection.

REFERENCE

- [1] Low Voltage Logic Threshold Levels: http://www.interfacebus.com/voltage_LV_threshold.html.
- [2] http://en.wikipedia.org/wiki/Faraday_cage.
- [3] http://en.wikipedia.org/wiki/IEEE_1284.
- [4] http://en.wikipedia.org/wiki/Thin_Ethernet.