ECE 327: *Electronic Devices and Circuits Laboratory I*

Infrared Remote Control Question and Answer

> *I'm writing up my section of the latest lab report (the IR link portion) and I'm having some trouble finding information for a conclusion section. I thought I would talk about why the distance between the transmitter and receiver has such an impact on the integrity of the pulse widths being transmitted, and how this problem is avoided in most commercial products (VCR remotes work from any distance within a certain range, right?), but I can't seem to find anything about this online. Can you explain why the distance is important? Is it just due to noise and interference?*

That's a great question. First, let me tackle what causes problems in our lab. Then I'll tackle how real IR products deal with it (short answer: they use *digital* schemes that are much more robust).

Infrared is cheap to use in low data rate products — because it is easily blocked by walls, you don't need a license from the FCC to use it. Unfortunately, because visible light dumps noise right on top of your signal, it's prone to noise and nearly impossible to use outside or even inside with high data rates. So that's why people use RF.

Our IR receiver uses a low pass filter (LPF) and Schmitt trigger to reproduce the transmitted signal. You can think of the output of the LPF as a measure of the received infrared power. Because of the noise, the LPF is always going to sit at some nonzero value. By adjusting the distance between the rails, we adjust that value. If the noise power is too high, the Schmitt trigger will trigger easily and will take a longer time to turn off (so pulse width is longer). If the noise power is too low, the Schmitt trigger will take longer to trigger ON but will easily trigger off (short received pulse). So we adjust the distance so that the noise doesn't bias us one way or the other.

Now, I know our system SEEMS pretty sensitive to noise. However, imagine trying to send analog data without the pulse-width modulation. That is, imagine trying to "send amplitude data" rather than "sending time data." We'd be in even worse shape.

Television remote controls (and other similar devices) are more clever about how they send their data. For one, the data they communicate is inherently digital (i.e., out of the 36 buttons, which of them did you push — that's a digital quantity). So they can use digital modulation schemes (and more powerful transmitters and better receivers).

For example, the popular "RC5 coding scheme" encodes 1's and 0's over a transmitted pulse train. The resulting string of 1's and 0's can be parsed in software to figure out what information (e.g., which button was pushed and which device was meant to receive it) was sent. Here are some links related to RC5.

- http://users.pandora.be/davshomepage/rc5.htm
- http://www.atmel.com/dyn/resources/prod_documents/doc2534.pdf

We could use a similar scheme. For example, with the random delay noise that we have, it might be hard to be certain that a received pulse encoded 5 V, but it would be pretty easy to tell the difference between 2 V and 8 V given that we knew those were the only two possibilities. We could encode "1" and "0" using 2 V and 8 V, respectively, and then use pulse code modulation (like a CD) to read off our information. For example, each group of seven received signals would be clustered to form a single digital code, which we'd run through a DAC and send out to our speaker.

You might be interested in the very first remote controls. Before people came up with these (complicated) modulation schemes, they took a simpler approach. They put four receivers across the top of the (large) television — two volume and two channel buttons — and you aimed a single infrared "flashlight" at the button you wanted. Provided your "flashlight" had enough directionality (i.e., it was a "tight" beam) your TV could detect the presence of one of the four signals and take the appropriate action. Kind of reminds me of duck hunt.

Finally, other remote controls combine multiple "colors" of invisible light. With only one "color," the TV could just tell if it was on or off. However, if the TV can "sample" all three, it could decode 8 different signals. So you could have a remote with 8 different buttons, and all your TV would have to do is be able to tell if a light was on or off (i.e., parallel digital communication rather than serial).