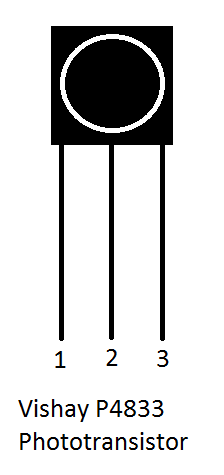
**The Infrared (IR) Receiver and Transmitter**

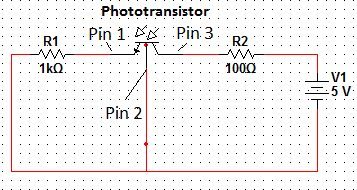
Using these parts:

1kΩ Resistor

100Ω Resistor

IR Phototransistor

Build the following circuit:



This is your **receiver** circuit. It will detect received IR signals.

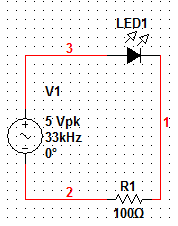
Apply power to the **receiver** circuit. Using channel 1 of your oscilloscope, check the DC voltage on pin 1 of your **IR photodetector** with respect to the battery negative terminal(**The channel in use MUST be set to DC coupling to get appropriate readings.**). Make a note of the voltage. (My value = 3.8v)

Using these parts:

IR LED

100Ω Resistor

Build the following circuit:



This is your **transmitter** circuit: when powered the **IR LED** will emit infrared light. Infrared light cannot penetrate surfaces that are too opaque, so you need to make sure your transmitter and photodetector are within sight of each other.

Turn on power to the circuit. Check the DC voltage on pin 1 of the **IR Photodetector**: you should get a much different value than the one you got before. If not, check your connections (My value = 0v)

You now have your ON and OFF values: the **IR Photodetector**, will permit current to flow to the output pin when it does not detect a 33kHz IR signal. When it does detect a 33kHz signal it will block current. Adjust the signal frequency slightly: the detector will still trip at slightly higher and lower frequencies at values equidistant from 33kHz: 33kHz is the “center frequency,” it is the critical frequency of the filter circuitry within the IR photodetector.

Play with the different signal shapes on the signal generator: you’ll find that the receiver works just fine with both sine and square waves at 33kHz but has some difficulty with other waveforms.

If you have a phone camera with a video viewfinder it may be able to display infrared light: point it at the center of the **IR LED** while the LED is powered and you may see a very dim purple glow.

**Using IR to transmit a AM signal**

Using Channel 2 of your oscilloscope, monitor the signal being sent from the signal generator to the IR LED

Press the “Mod” (Modulation) button on your signal generator: From the “Type” Menu, select AM. You are now using the IR LED to send a simple AM signal on a 33kHz carrier frequency. From the AMFreq menu you can adjust the frequency of the AM waveform from the default of 100Hz.

Now press the “shape” button and select “square.” Your AM signal is now a square wave: This is especially useful because, by comparing the signal passing through the LED and the signal on pin 1 of your receiver you can estimate how long it takes for your receiver to activate and deactivate after receiving a 33kHz signal. Make a note of both activation and deactivation times for the receiver (My values = 500us activation time 250us deactivation time.)

**Consumer IR signals**

Turn off power to the IR LED. Use your oscilloscope to monitor the voltage on pin 1 of your IR Photodetector (**The channel in use MUST be set to DC coupling to get appropriate readings.**) Now press any button on the IR remote to send a signal to your photodetector: you should get a series of pulses on your oscilloscope when a key on the remote is pressed. Getting a good signal here can take some work: I’ve gotten consistently good results with the following configuration using a Tektronix TDS2014B oscilloscope:

Volts/Division set to 2V

Seconds/Division set to 10ms

Trigger level set to 1v

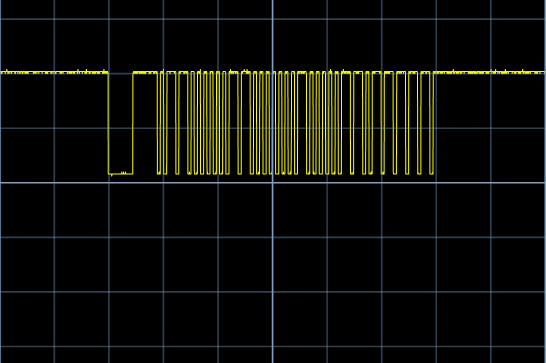
From the Trigger Menu (Press the “Trig Menu” button to bring up these options):

Set “Source” to the channel being used

“Type” set to “Pulse”

“When” set to “Less than” (<)

You should get a pulse that lingers whenever you press a button on the remote, like this:



Once you have a good signal you can pause the scope by pressing the RUN/STOP button.

Each dip in the voltage represents a period where the IR LED in the remote flashed at a frequency of 33kHz. Try a few different buttons on the remote and note how the pattern changes: Think of the dips as 1’s and the gaps as 0’s: if there was additional hardware attached to the receiver circuit, such as a DVD player, it could interpret this as a binary signal and activate the associated command.

Try pressing a few different buttons on the remote: You’ll notice that only a few bits actually change. (There are 45 bits in total, not counting the long flash and pause at the start: the first 21 bits and the last 9 bits are static and do not change regardless of which button is pressed. The middle 15 bits do change depending on the button pressed.) The other bits, as well as the long flash and pause at the start of the signal, allow the receiver to identify the specific signal. Most consumer IR remotes operate at 33kHz so these identification bits are necessary to prevent unintended interference.