BEST IR System Theory of Operation

This document has a schematic of the BEST IR (InfraRed) transmitter and receiver circuit and describes how the circuit operates.

Background Information

The basic idea is that the transmitter emits a beam of light and the receiver pulls its output line low when it detects that light. By running the transmitter continuously and monitoring the receiver output line one can tell if something blocks the beam. By having the receiver look for reflected light one can make a line follower. By turning the transmitter on and off using its control input, one can send signals to whatever is connected to the receiver. Cool!

Of course it's not quite that simple. There is a *lot* of light around, for some reason people don't like to stumble around in the dark. Some of this light is flashing on and off, such as a fluorescent light which turns on and off 120 times each second. Our eyes don't even notice this but a fast electronic detector easily notices the flicker. And then there's the intensity of sunlight, orders of magnitude brighter than any of our electric lights. The sunlight trumps everything. Don't expect this circuit to work well in direct sun; the bright light will "wash out" any signal.

Ignoring sunlight, which we can't do anything about, how can we make our system immune to the flickering light all around us even when we are indoors?

The BEST IR system uses two techniques to minimize interference from our man-made light sources – the frequency of light used (its "color") and by turning our light on and off at a specific rate ("modulation").

Everyone is familiar with colors of light; the color is an indication of the light's frequency. Blue light has a high frequency, red light has a low frequency, and the other colors we see are between the two. But … the fact we don't see any light does not mean there is no light – it might be a color that we cannot see. Ultraviolet light has a very high frequency, too high for our eyes to detect, but the energy in ultraviolet light can damage our skin – we call that a sunburn. At the low frequency end we have infrared, or IR, light. Again, our eyes cannot detect infrared light but electronic detectors are very good at detecting it.

Look at the package of the IR detector. It looks almost black. Just as we can make plastic that lets red light thru but blocks blue or green light, we can make plastic that lets infrared light thru but blocks all the visible light colors. Such a plastic is used on the IR detector package. To us it looks black because no visible light can get thru it but it is transparent to IR light which passes easily to the detector chip. If we use an IR light source on our transmitter and this IR detector, just like that we have eliminated much of the light that could cause interference with our detector. The dark plastic package blocks all but the IR frequencies of light that our transmitter produces.

Now let's talk about modulation. Here is a good definition for modulation: "The exertion of a modifying or controlling influence on something". Instead of merely producing IR light that is always on, we can modulate it by rapidly turning it on and off. In fact, our IR detector chip is designed with this in mind. It is looking for an IR source that turns on and off 38,000 times per second, which is also known as 38 kiloHertz, abbreviated as 38 kHz. The detector ignores any IR light that is not turning on and off 38 kHz, such as a fluorescent light flickering 120 times each second (120 Hz). This gives us an added measure of rejection of unwanted signals.

All we need to do is control our IR source, turning it on and off 38,000 times each second. It's time to dive into the schematic to see how we do this.

Schematic Time

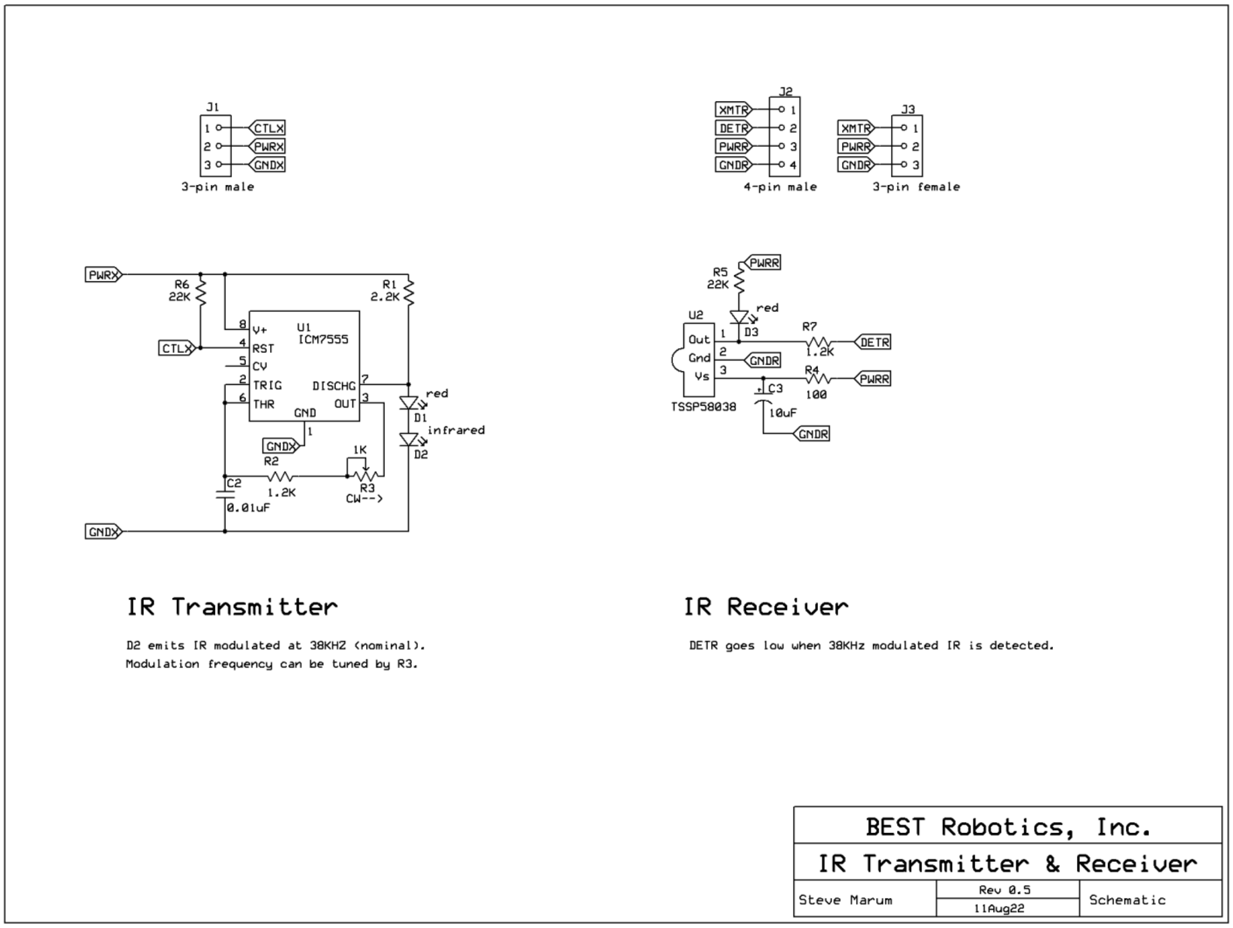

Figure 1, schematic, transmitter on the left and receiver on the right.

The IR Transmitter

The transmitter uses a 555 timer, U1, to control the current thru the infrared LED, D2. As you can see the 555 timer has several control pins, allowing it to be used in many different ways. We are using it as an astable multivibrator, that is, it does not have a stable state and continuously switches back and forth between two states.

The frequency at which the 555 timer runs is set by the time constant of an RC network – a capacitor (C_{timing} = C2) in series with a resistor (R_{timing} = R2 + R3). In this case one end of the resistor is connected to the pin named "OUT", which is a digital signal. When OUT is high the capacitor charges thru the resistor and the voltage across the capacitor increases. When OUT is low the capacitor discharges thru the resistor and the voltage across it decreases.

You can see this action in figure 2. Look at channel 2 (pink). The x-axis is time, the y-axis is voltage. This produces a graph showing how the voltage changes over time. The probe is connected across C2 (at the node near the “C2” label) and you can see that the voltage alternates between ramping up and ramping down. The 555 timer monitors this voltage and switches the output low when this voltage reaches the upper trip-point. Now the voltage begins ramping down until it reaches the lower trip-point, when the output switches high again and the cycle repeats.

The rate at which the capacitor charges or discharges is set by the value of the capacitor and the value of the resistor. All electronic components have tolerances, given as a percentage indicating how much different the actual value may be from the value stamped on the component. This deviation is typically in the range of 1% to 10%, depending on the component. What this means to us is that we cannot assume the resistor and capacitor are the exact values we need to obtain the oscillation frequency we want, 38 KHz in our case. We need to be able to adjust either the capacitance or the resistance. This is why our resistance, R_{timing}, is comprised of a fixed resistor, R2, in series with a variable resistor, R3. Resistances in series add, so R_{timing} = R2 + R3. By changing the value of R3 we alter the frequency of the oscillation and can adjust it to be the 38 kHz to which the detector responds. If you look in the upper right corner of figure 2 you can see that in this example the circuit is tuned to 38.071 kHz.

Channel 1, the yellow trace, shows the voltage at the DISCH pin of U1. When this signal is high, current is flowing from the positive supply thru R1 and then thru the two diodes D1 and D2. D2 is the infrared LED and produces the IR signal we use. D1, a red LED, is just there so you can see that something is happening, since humans cannot see IR light. When the signal on channel 1 is low, the current passing thru R1 is shunted to ground via the DISCH pin of U1 and does not pass thru the two diodes. No light is produced.

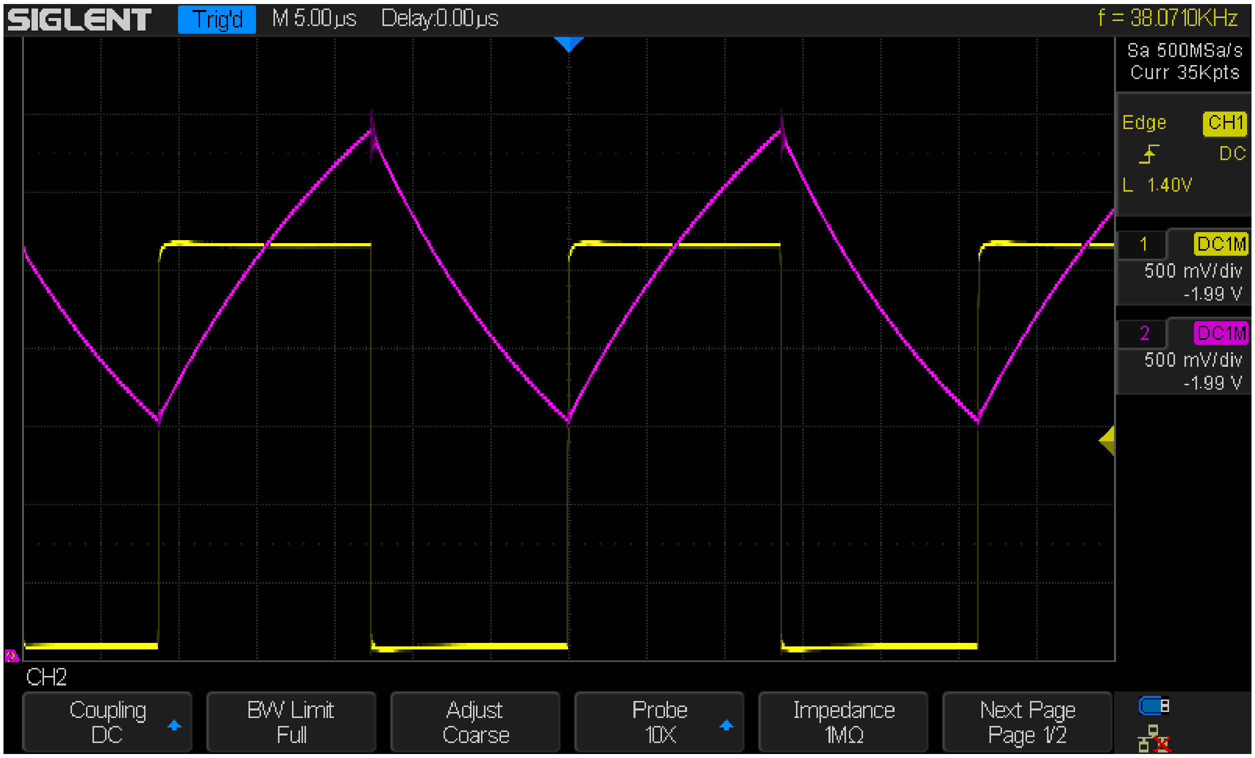

Figure 2, Oscilloscope screenshot showing 555 timer operation.

Modulating the IR Transmitter to Send Information

"But," you say, "the IR transmitter is already modulated at 38 KHz." True, it is, but we can add additional modulation to the IR transmitter to send information over this beam of IR light.

The 38 KHz on/off switching of the IR LED is a "carrier" frequency. It doesn't actually have any information useful to our robot, instead it "carries" the information. The 38 KHz carrier frequency is used by the receiver IC to detect when a valid IR signal is being received.

Ted has a lil' red wagon he uses to haul all his things on contest day. Ted doesn't really want the wagon but wants all the things piled *in* the wagon. The wagon is a tool to get those things to where Ted needs them. In the BEST IR system the 38 KHz carrier is like Ted's lil' red wagon and the information we want to send out is like the stuff in Ted's wagon, riding on top of the 38 KHz carrier.

We can send information over this IR light by turning the 38 KHz carrier pulsed IR light on and off at a much lower frequency, producing bursts of 38 KHz IR light. Perhaps our information is encoded into the duration of this burst of 38 KHz IR light – a burst of 2 milliseconds means one thing and a burst of 4 milliseconds means something else. Or maybe we connect the transmitter to the UART output of the Cortex and send ASCII data.

This is accomplished by using a Cortex output to drive the CTLX line. When CTLX is left unconnected or pulled high the IR transmitter runs and emits light that switched on and off at 38 KHz – our carrier. When CTLX is pulled low the IR transmitter turns off, emitting no light.

This action is shown in figure 3. Channel 1 (yellow) shows the signal on CTLX, this is our information we wish to send. Channel 2 (pink) shows the voltage at the DISCH pin of the 555 timer, this is our carrier being modulated by the information. A high voltage on DISCH means the IR LED is emitting light, a low voltage on DISCH means the IR LED is off.

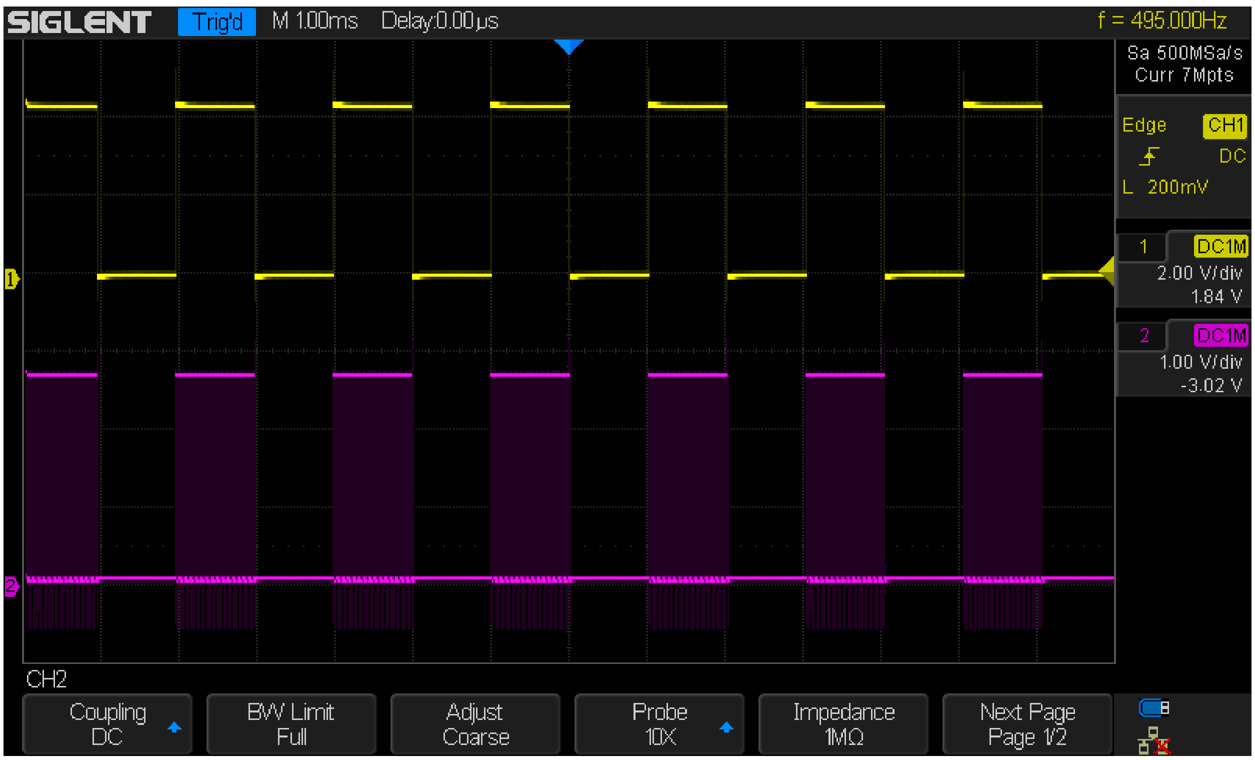

Figure 3, Screenshot of modulated 38 KHz.

Here you can see the controlling signal in yellow and IR LED signal in pink. When the yellow trace is low the IR LED is off and the pink trace is also low. When the yellow trace is high the IR LED is on and the pink trace switches high at a 38 KHz rate. At this scale the pink trace looks confusing – it appears to be both high and low! How can this be? Actually it is switching rapidly back and forth between high and low as described earlier; the oscilloscope just cannot resolve the rapid switching on this time scale. In the expanded trace below, a single positive pulse in yellow, you can see the rapid switching occurring on the pink trace.

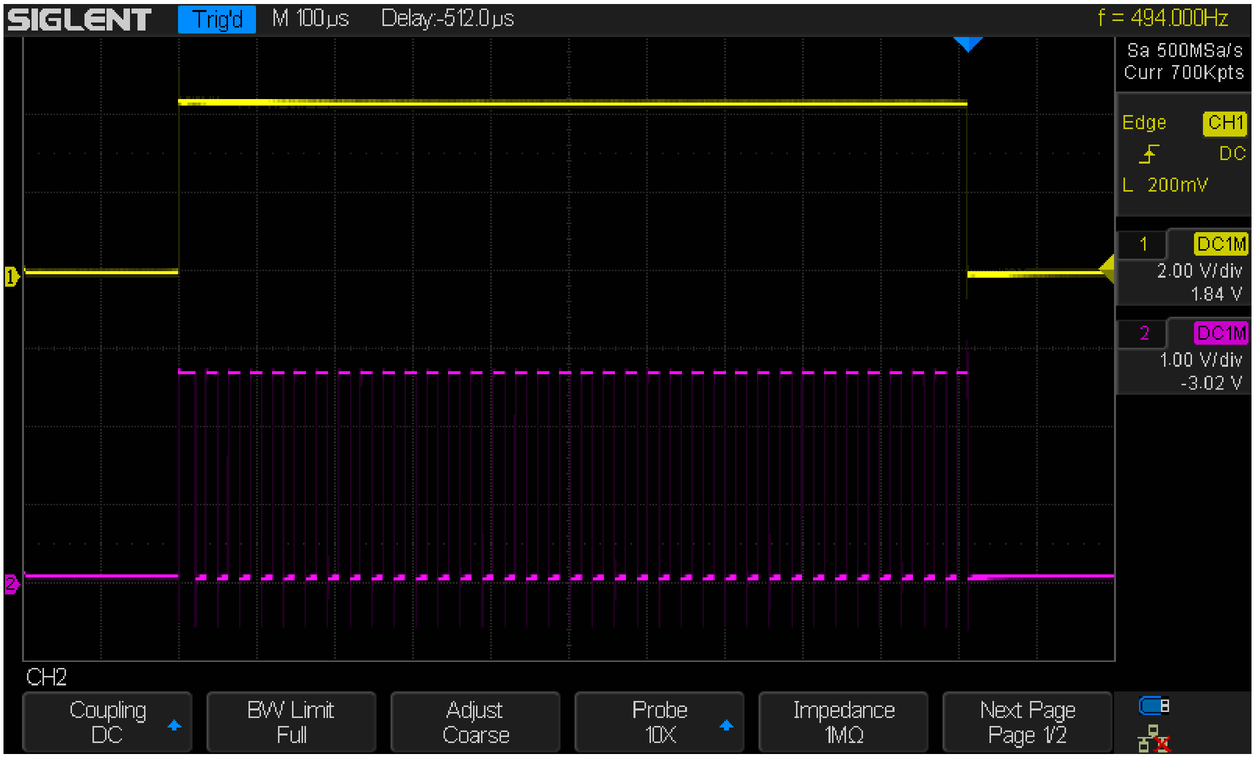

Figure 4, Expanded oscilloscope screenshot of one burst

There is an alternate way to control the IR transmitter. Forget about the CRTL line – leave it unconnected. Now the IR transmitter runs whenever power is applied. The IR transmitter only draws about 3 milli-amps and its power or ground lines can be driven directly from a Cortex digital output.

If you connect the IR transmitter PWRX to +5V and GNDX to a Cortex output the IR turns on when the output is low.

If you connect the IR transmitter PWRX to a Cortex output and GNDX to 0 volts the IR turns on when the output is high.

The IR Receiver

Phew! We made it thru the transmitter explanation. The receiver is much simpler, but only because U2, the TSSP58038 IR receiver chip, does all the heavy lifting for us.

The TSSP58038 has a lot of circuitry inside it. It starts with an IR sensitive device, followed by an amplifier for the tiny signal, then a bandpass filter tuned to 38 KHz to reject other signals, ending with a detector (is 38 KHz present, yes or no?) and an output driver. We just take that output signal and pass it back to our Cortex. R5 and D3 provide a visual indication when a valid IR signal is received.

“So what’s with all that other circuitry?” you ask. C3 is a filter on the power supply to prevent electrical motor noise from interfering with the receiver IC. R4 and R7 are “klutz protection”, preventing the receiver IC from being destroyed should someone (not you, of course!) plug the cable in backwards.

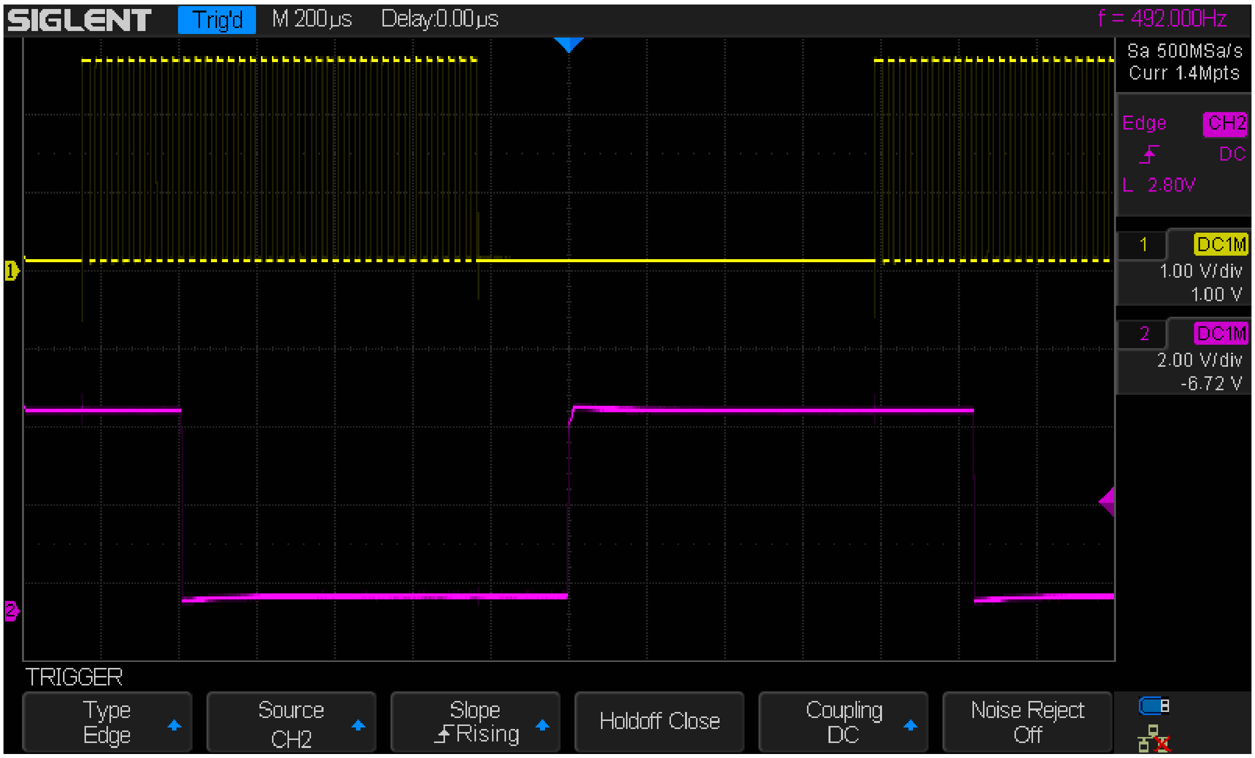

Figure 5, Oscilloscope screenshot of IR LED drive and receiver output

Finally, figure 5 shows the drive to the IR LED in yellow and the output signal from the receiver in pink. You can see there is an inversion, when the IR LED is off the output is high and when the IR LED is flashing at 38 KHz the output is low. You may need to adjust for this inversion in your software. You can also see that there is a delay between the IR LED starting to flash and the output going low as well as a delay between the IR LED turning off and the output going back high. This delay is a result of the bandpass filter, which needs a few cycles to determine that the IR signal is present at the correct frequency, or that yes, the IR light has gone away. This lag limits the rate at which we can drive the system to send data, our data pulses need to be much longer than this delay so they are not “swallowed up” by the filter.

In figure 5 the pulses are about 1 millisecond wide with about 1 millisecond separation. The delay introduced by the TSSP58038 IR receiver is about 25% of the pulse widths so 1 millisecond is about the shortest pulse width we can send reliably. Sending serial data at 300 baud (3.3 millisecond pulse widths) should be fine, maybe even 600 baud. Jumping to 1200 baud would be risky.

In Conclusion

That is a brief summary of how the BEST IR transmitter and receiver work. All the components needed to build one transmitter and one receiver are included in the kit, you just need to solder them together. See the Assembly Instructions for complete details on putting the components on the PC boards.

These two simple circuits are versatile building blocks which can be used to do any of these things:

- Beam Blocker: Run the transmitter continuously and use the receiver to detect when something blocks the light
- Line Follower: Run the transmitter continuously and use the receiver to detect the light reflected from a line under the robot
- Simple Signaling: Communicate with another robot or the field by sending and receiving pulses of various widths, or different numbers of pulses
- UART (Universal Asynchronous Receiver/Transmitter): Connect the receiver to the UART port of the Cortex and the transmitter to the receiver board. Now your robot can send ASCII characters to another robot or the field, and receive replies from another robot or the field. This would be really cool if we could just figure out a game that could use this ability. ☺
- ???: What does *your* imagination come up with?

Revision Notes

11Aug22: Removed C1 so transmitter could more easily be connected to send serial data.