Examining some IR capabilities of the Boe-Bot

J. C. Rogers

March 23, 2006

I - INTRODUCTION

Infrared, IR, capabilities of the Parallax Boe-Bot can provide significant enhancement to projects. However, there are several characteristics of the IR system components that must be understood to achieve desired results. Understanding features of the transmitting IR diode, the integrated circuit IR receiver, Boe-Bot command timing, etc are essential.

ET 370 laboratories 1, 2, and 10 all dealt with IR system capabilities. These provide a basis for this paper which presents a series of measurements that reveal additional details of IR applications for the Boe-Bot. A principal intention of the measurements is to answer all or part of the following questions:

1. Can a Boe-Bot transmit an IR code without using a 555 timer to generate the 38.5 kHz carrier?
2. What are the transmitting details for code generation without a timer and what are the implications for a receiver that must interpret this code?
3. What are the timing and sensitivity issues that affect the Boe-Bot's ability to use IR sensing to avoid (or seek) objects and to follow (or avoid) black lines on a white background?

The sections that follow are arranged in the same order as these questions.

II - TRANSMITTING IR CODE WITHOUT A 555 TIMER

Laboratory 1 investigated characteristics of an IR photo detector. A key feature of this integrated circuit is the fact that it had an internal band pass filter tuned to a particular carrier frequency, 38.5 kHz. This feature is of immense importance for avoiding IR interference from incandescent lamps, fluorescent lamps and sunlight. All IR systems that use this receiver must modulate their transmitted IR at 38.5 kHz (or another carrier frequency required by a particular receiver).

In laboratory 10 an astable 555 timer circuit produced a 5 volt peak to peak square wave with an approximate 50% duty cycle. This was suitable for driving an IR diode in series with a current limiting resistor. It was seen in ET 250, laboratory 14 and in the text by Hambley, that square waves have spectral components consisting of the fundamental frequency and all odd harmonics. The size of the harmonics fall off as 1/n where n is the order of the harmonic. The third harmonic of a 1 kHz square wave, 3 kHz, has an amplitude 1/3 that of the fundamental. Consequently, the 38.5 kHz square wave out of the timer has spectral content at the fundamental, 38.5 kHz, at the third harmonic, 115.5 kHz and all the other harmonics. Only the fundamental passes through the receiver. In effect all energy in harmonics is "wasted". Maximum results would be possible if a single sinusoid were produced to drive the transmitting IR diode. However, the astable timer circuit is easy to produce and served well in laboratory 10.

Figure 1 contains a 5 volt peak to peak, 38.5 kHz, square wave produced with the Wavtek function generator in the lab. Its spectral content from 0 to 62,500 Hz is also shown. Only the fundamental frequency is seen, it is a 38.5 kHz sinusoid with an rms amplitude of 1.79 volts. This component is suitable for reception by an IR receiver.

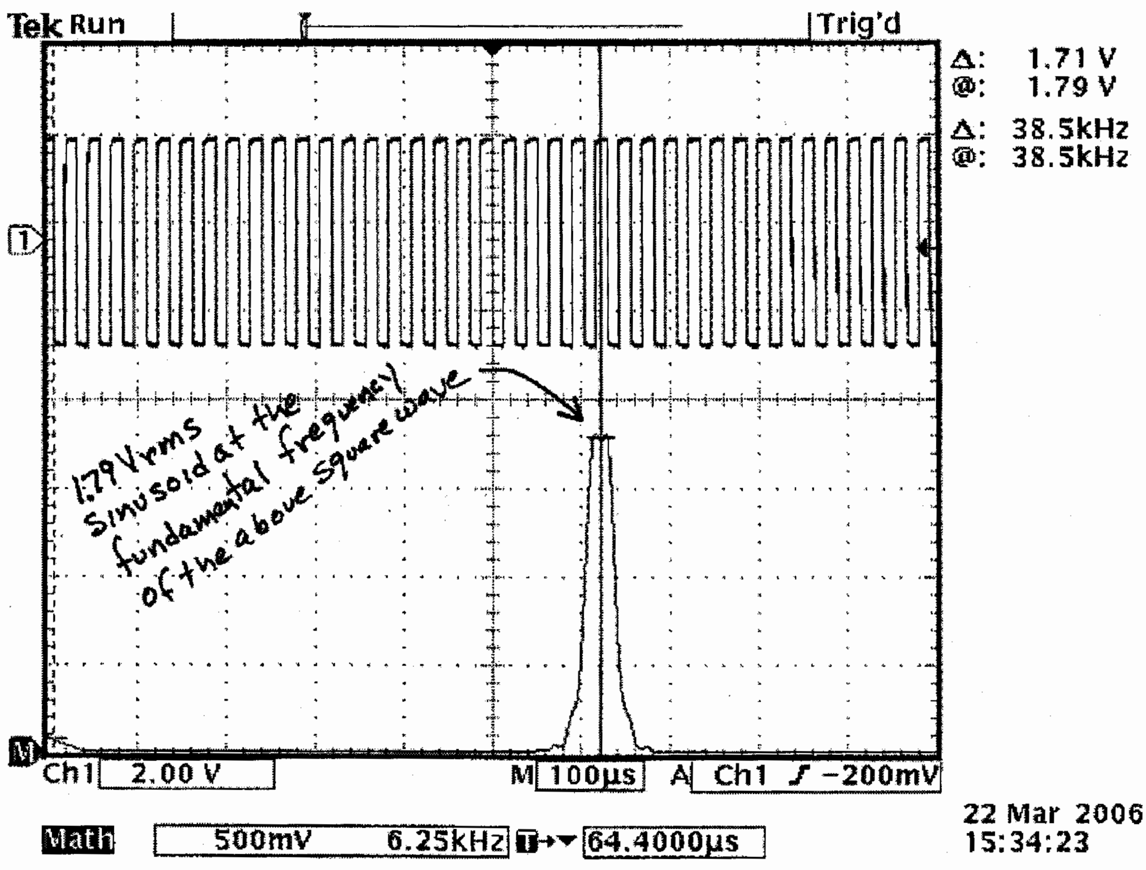

Figure 1. A 38.5 kHz square wave and its fundamental harmonic.

The FREQOUT command of the BS2 microcontroller can produce voltages with varying frequencies and these can be heard on a piezospeaker. Whereas the Parallax manual indicates the frequency range possible for the BS2 is 0 to 32,767 Hz. It is possible to exceed the upper limit and produce useful signals. However, the BS2 is limited in its timing capabilities by its master clock. An example of this is that the PULSIN command measures the duration of a 1 or a 0 applied to a particular pin with the unit of measurement, a count, being 2 microseconds in duration. As a result, the FREQOUT command produces some rather strange looking waveforms, these are not sinusoids, which is impossible for a digital device which can produce only 0 or 1 as an output (0 Volts or 5 Volts in this case). Nor are the output waveforms single frequency square waves. Instead the output is switched on and off with a repetition governed by the internal clock and calculated to produce significant spectral content at the desired frequency. Figure 2 shows the result of executing the command FREQOUT 2, 100, 38500. This is a command to produce a 38,500 Hz wave at pin 2 for a duration of 100 ms. Note that while significant spectral intensity is seen at 38.5kHz, slightly more is produced at about 27 kHz. The later would be rejected by the IR receiver. Also, note the large dc component of the spectrum. This is a result of the fact that the waveform's lowest voltage is 0 Volts and its highest is 5. Consequently, its average, or dc value is approximately 2.5 volts.

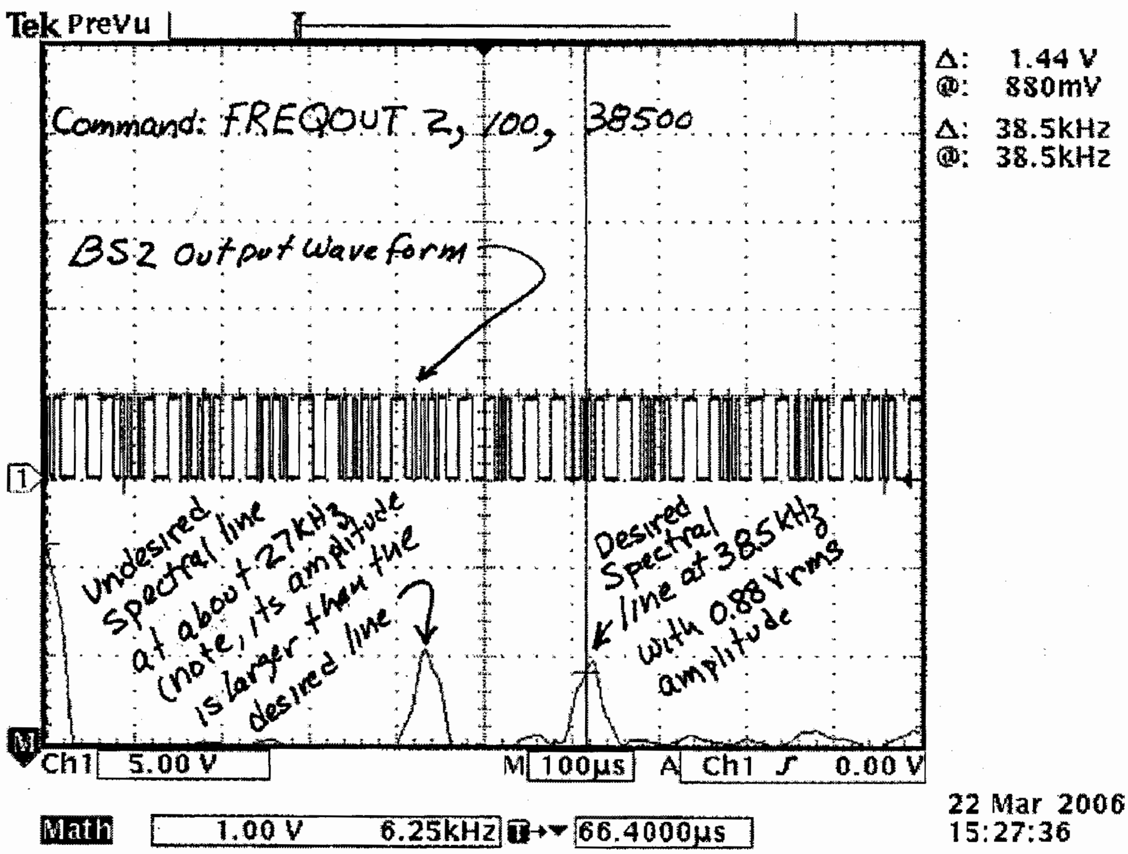

Figure 2. BS2 output waveform and its spectrum. The FREQOUT command frequency was 38,500 Hz.

The peak to peak amplitude of both the waveforms seen in Figures 1 and 2 is 5 V. However, the amplitude of the fundamental frequency for the square wave is 1.79 V rms while that produced by the BS2 is 0.88 V rms. The BS2 desired frequency spectral output is about 49% of that produced by a 38.5 kHz square wave with the same amplitude. While the microcontroller output is less, this signal is significant and can be expected to serve effectively as the 38.5 kHz carrier required by an IR receiver. Question 1 in the Introduction can be answered affirmatively. Question 2 is now addressed.

III - PRODUCING A SONY – LIKE PWM CODE WITH THE BS2 ALONE

In laboratory 10, an astable 555 timer producing a 38.5 kHz carrier was switched on by use of the PULSOUT command applied to a microcontroller pin connected to the timer reset. The resolution for that command is in counts, 2 microseconds per count. As a result, the on times specified for the Sony protocol (see laboratory 2) were easily achieved. These times are: start pulse, 2.4 ms (1200 counts); "zero" pulse, 0.6 ms (300 counts); "one" pulse, 1.2 ms (600 counts). The off time between bits for the Sony code is 0.6 ms, shorter than the shortest pause (1 ms) produced by the BS2. A work around was to produce a PULSOUT command for an unused pin having the proper duration so that the next "on" segment for the carrier started exactly 0.6 ms after the last "on" segment ended. The duration of the work around command was 100 counts. (See the program Ir_Code_generator.BS2) This code was successfully interpreted with another program, Get_Ir_remote_code.bs2 which mimicked a program written by Andy Lindsay to interpret messages from a Sony IR remote control.

It was concluded in the previous section that the FREQOUT command can produce 38.5 kHz carrier signals comparable to that used in laboratory 10 with a 555 timer. However, the time unit for the FREQOUT command is a millisecond, not two microseconds. Consequently, the BS2 is incapable of using FREQOUT to generate Sony Code; the code's basic unit of time measure is 0.6 milliseconds. It is possible to generate a Sony-like code with the basic time measure being 1.0 milliseconds. Thus a start pulse could be 3 ms, a "0" could be 1 ms, and a "1" could be 2 ms. The off time, time between FREQOUT commands, would ideally be one time measure or 1 ms. Inserting a 1 ms PAUSE between the 38.5 kHz carrier pulses will produce an off time greater than 1 ms since there is time required for code interpretation and execution. This additional time appears entirely in the time between carrier pulses produced by the FREQOUT command. The four bit code below generates a start pulse, two "0's" and two "1's".

```
top:
FREQOUT 2, 3, 38500        'the start pulse is 3ms
PAUSE 1                    '1ms pause, off time is > 1ms
FREQOUT 2, 1, 38500        'the on time for "0" is 1ms
PAUSE 1
FREQOUT 2, 1, 38500
PAUSE 1
FREQOUT 2, 2, 38500        'the on time for "1" is 2ms
PAUSE 1
FREQOUT 2, 2, 38500
PAUSE 10
GOTO top
```

Figure 3 shows the microcontroller's Sony-like code output for the above program. The output of an IR receiver that intercepted the code is shown in the upper wave form. A 240 microsecond delay is seen between the transmitted code and the received code.

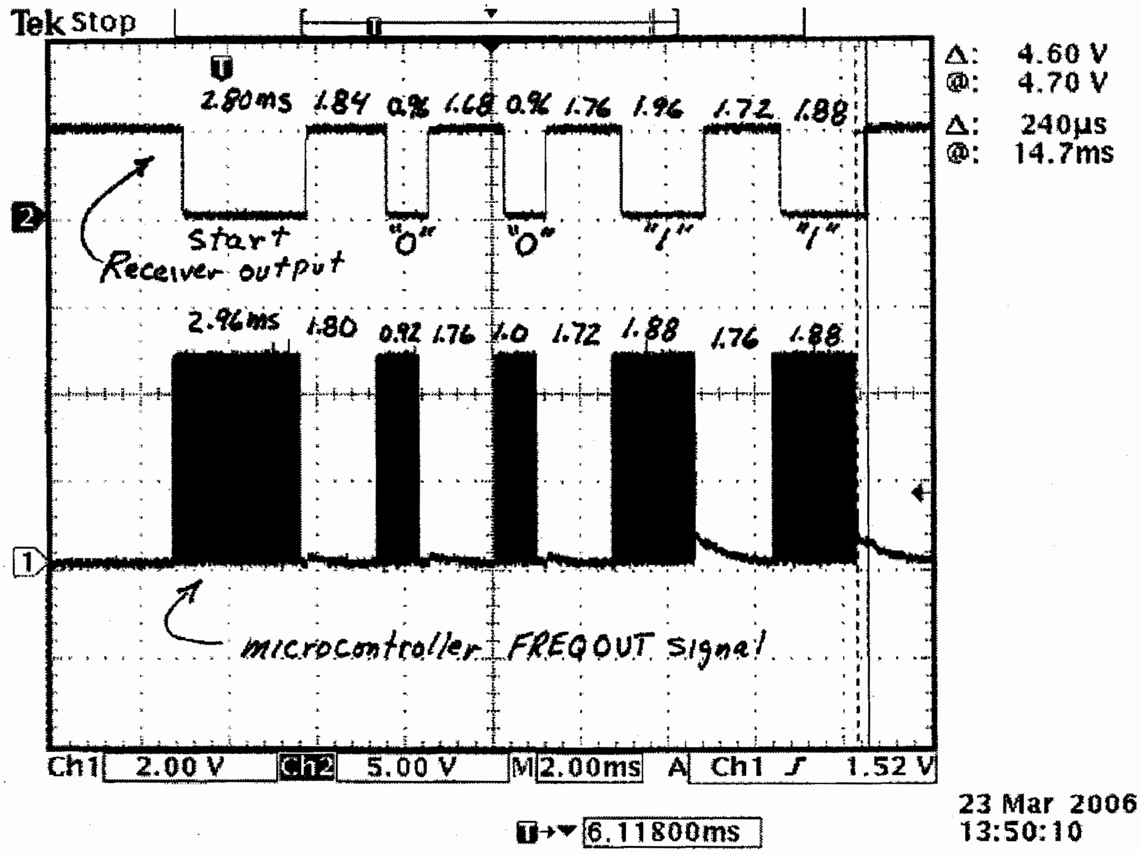

Figure 3. Four bit Sony-like PWM code produced by microcontroller and received by IR receiver.

Some time variation is evident between the program code, the microcontroller output waveform and the receiver output. However, these should not provide a major hurdle in interpreting the transmitted bits. Of particular note is the time between bits. Although a 1 ms pause was used in the program, the code interpretation and execution times combine with the pause to produce received waveform off times ranging from 1.68 ms to 1.85 ms. This time is sufficiently long to allow use of the PULSIN command to scan for a "1" or a "0". (The actual Sony code, with an off time of 0.6 ms, was too quick for the microcontroller to use the PULSIN command to identify bits)

A program, Get_Sony-like_Ir_remote_code.bs2, was adapted from a program provided in laboratory 2. It successfully interprets the four-bit code generated using FREQOUT commands in the program lines shown above. Received data pulses greater than 750 counts, 1.5 ms, are interpreted to be "1". The code was successfully received at distances up to 50 cm although there was some erroneous reception. At closer distance, 20 cm, the reception and interpretation were quite reliable. The current limiting resistor in series with the diode was 1000 Ohms. Smaller resistor values will provide longer range. Figure 4 was produced according to the procedure in chapter 7, page 249 of Robotics with the Boe-Bot. In the case of detecting the rather large tablet, the IR signal had to travel to the tablet, be reflected and return to the receiver located adjacent to the transmitting diode. Reducing the series resistor from 1000 Ohms to 220 Ohms approximately doubles the detection distance seen in the figure. A similar improvement in effective communication distance could be anticipated for the IR diode transmitting to the IR receiver.

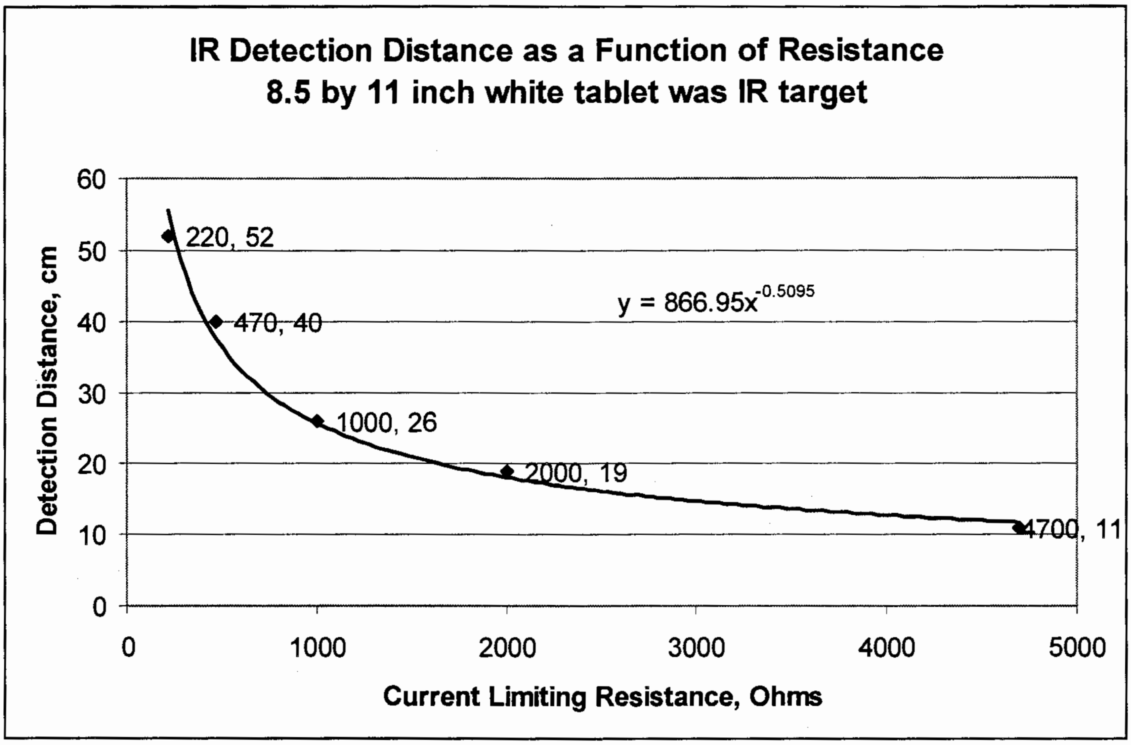

Figure 4. IR detection distance verses size of the current limiting resistor in series with the IR transmitting diode.

In summary, a successful Sony-like code was produced with one BS2 using the FREQOUT command and received by another BS2 using the PULSIN command. Any successful communication between roaming Boe-Bots depends on proper aim of the transmitting diode, which has a beam width of +/- 10^0. The receiver lens is less selective with a beam width of +/- 30^0. Thus, alignment is a major issue for successful IR communications.

IV – IR SENSING OF LINES AND BOUNDARIES WITH A Boe-Bot

Chapter 7 of Robotics with the Boe-Bot introduces methods to pulse the IR diode with a 1 millisecond duration 38,500 Hz signal and detect reflected pulses with the IR receiver. A rule of thumb for light travel is "one nano second is a light-foot". A nano second is 10^{-9} seconds. Clearly, the round trip time from the transmitter to a reflector and back, only a few nano seconds, is incredibly faster than the time for the microcontroller to interpret and execute a single instruction, approximately 0.25 milliseconds or 250,000 nano seconds! The only thing that makes it possible for the microcontroller to transmit and then receive the pulse (which arrives back to the receiver in a few nano seconds) is the fact that the circuit elements and the band pass filter in the IR receiver introduce delay. This delay is sufficient for the microcontroller to detect the fact that the receiver did indeed receive a 38,500 pulse. Figure 5 shows the receiver output delay is on the order of 0.25 ms, just enough for the microcontroller to determine whether the receiver output is a "1" or a "0". See page 240 in the Boe-Bot book for additional discussion.

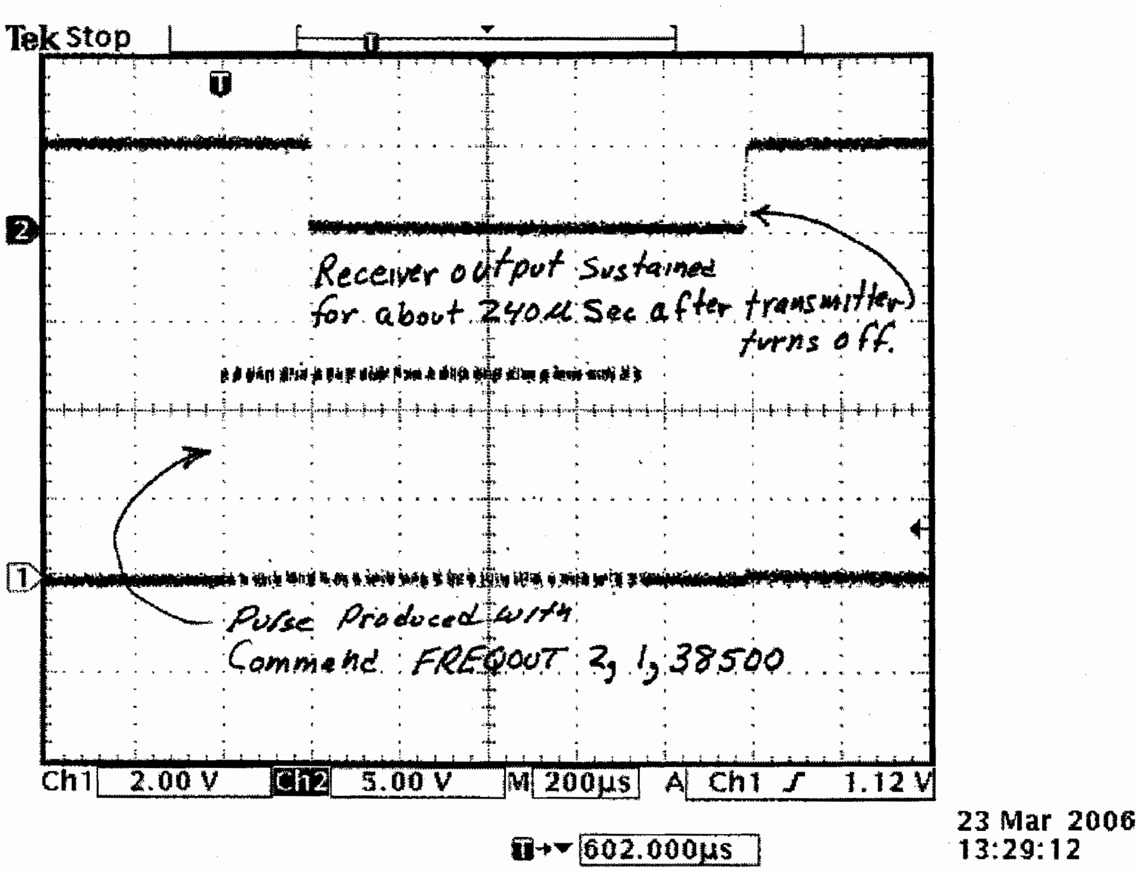

Figure 5. The lower waveform is a typical 1 ms, 38,500 FREQOUT pulse used for Boe-Bot navigation. The upper waveform is the IR receiver output. Since the Be-Bot can only do one thing at a time, it is essential that the receiver output remain low after the transmitter stops its pulse. This is the case for approximately 0.25 ms.

It may not be desirable transmit the strongest possible IR signal (using a 220 Ohm current limiting resistor) for certain tasks. Large resistors make the Boe-Bot "near sighted", it can only detect lines and objects that are close. Thus, if it is moving, it must be close to lines and objects before it can detect them. Its field of view is restricted and this could be useful in a cluttered environment. My own experiments with limiting transmitting diode current by increasing the resistor have met with limited success for the purpose of finding edges of black lines. A simple approach, I see or do not see the line, can be effective but it is limited. The following paragraph describes a more refined approach.

The method just discussed, reducing the Boe-Bots sensitivity requires replacing the current limiting resistors. Another method exists to adjust detection distance, without changing the resistors. It is described in Chapter 8, Robot Control With Distance Detection. Instead of altering the transmitter diode current to limit signal strength and consequently the detection distance, the carrier frequency is altered to reduce the effective signal that passes through the band pass filter in the receiver. Recall that in laboratory 1 a series of frequency changes were made to the signal driving the transmitting IR diode. This allowed plotting an approximate filter characteristic for the band pass filter. Considerable laboratory effort was required to make the measurements but it was concluded that tuning the applied signal away from the filter's center frequency was an effective means to reduce the system's communication ability. Transmitted signals with carrier frequency away from the filter center frequency must be stronger after they have been reflected and arrive at the receiver if they are to be detected. Only nearby objects will return strong signals needed to activate the receiver, only nearby objects can be seen.

Activity #3, in Chapter 8, Following a Stripe, has several features that are important to the Boe-Bot acting inside or outside of a ring. In the ET370 case, the ring will be three widths of ¾" electrical tape applied to white poster board.

V – SUMMARY AND CONCLUSIONS

It was seen that a method of generating the necessary carrier for IR sensing and communicating exists. The FREQOUT command can produce a 38,500 Hz signal with strength approximately 50% that of the signal strength provided by the 555 timer. Thus it is not mandatory to use the timer for communications. However, The FREQOUT command lacks the time resolution, 0.6 ms, required to implement the Sony protocol for IR communications. An alternative Sony-like protocol was proposed and two programs were developed to achieve BS2 to BS2 communications. Finally, line following with the Boe-Bot was considered. Examination of line following methods used in Chapter 8 of Robotics with the Boe-Bot suggests that the carrier frequency sweep approach is the most robust for line following.

BIBLIOGRAPHY

Lindsay, Andy, Boe-Bot book, chapter 7, Parallax Inc, Rockland CA, 2004

Lindsay, Andy, IR Remote for the Boe-Bot, Chapter 1, Parallax Inc, Rockland CA, 200