Chapter #8: Real-Time Programming with Bumpers and Infrared

Note to reader: This chapter requires the Toddler Toes Kit (#27312) from Parallax.

WHAT IS MEANT BY REAL TIME?

Real time programming is one of those terms that can mean different things at the same time. In plain language, it is code that allows the computer system to keep up with what is happening in the world around it – WHILE it is happening. Like in everything else, there are degrees of "real time". Programming a system to respond to events happening quickly is more difficult than if they happen more slowly. Programming events to happen at a very precise time can also be challenging – even if they don't happen at high speed.

Let's look at a simple example.

You want to flash 5 LEDs. You can determine if they need to be on or off by testing switch inputs. It's an easy task.

1. Turn off all the LEDs to start
2. Test each switch input and turn on the LED if the switch requires it.
3. PAUSE for ½ second – or however long you want the LEDs on.
4. Turn off all the LEDs
5. PAUSE for ½ second – or however long you want the LEDs off.
6. GOTO to #2 and start over.

You probably ignored how long it takes to test the inputs in step #2 before running the PAUSE in step #3. What if the program were required to flash the LEDs at precisely a 1 second rate (+- 0.001%)? Ignoring the calculation times can lead to disaster in that case. Not only would you have to measure or calculate the instruction times to correct the duration of the `PAUSE` but also you'd have to make sure that all possible cases of switch inputs take the same amount of time to compute.

Now, what if the switches only make contact for 20 milliseconds? Your code would have to constantly be scanning to see if the switch was making contact and remember it. The scanning has to happen while still keeping track of the flashing LEDs. If you had a loop to check the switches, you'd have to complete that loop more than once every 20 milliseconds. The BASIC Stamp makes it so easy to turn on or off LEDs that it probably still isn't a big problem. What if you had to do some calculations before deciding to turn the LED on or off? And what if those calculations take 30 ms for each LED?

A processor's got to know its own limitations!

Actually the processor doesn't care. But it sure helps a programmer to know them. Of course, one way to help in high-speed real time programming is to use a faster processor. But every processor has some limit. Often it isn't price effective to just throw a faster processor at a job. Besides, where's the fun in that? One key to getting the most of any given processor is to use some simple techniques (and sometimes not so simple techniques) to maximize the use of the performance that is there.

In this situation with the BASIC Stamp, more speed adds some help but it would really be nice to do more than one thing at a time.

Where's Real Time in Toddler?

Toddler Program 6.2: Object Detection And Avoidance sensed the outside surroundings using Infrared headlights and decided to where to move accordingly. It responded rather slowly because the sensing was only done between discrete moves. Wouldn't it be better to sense all the time during moves and respond immediately? As mentioned before, Toddler has a more difficult time dealing with sensor inputs because it is walking instead of rolling. In a rolling robot like Boe-Bot, you can back up or turn at any time. Toddler can't. If both feet are on the ground, it may first have to lift a leg before trying to move. This adds complexity to the program compared to a roller, but it can still be done. Better yet, it adds some personality to your Toddler too.

In this experiment we will use the Toddler Toes and the infrared object detection circuit from prior experiments to demonstrate real-time programming on the Toddler.

ACTIVITY #1: BUILDING AND TESTING TODDLER TOES AND INFRARED CIRCUIT

Parts Required

(1) Piezoelectric speaker
(2) Shrink wrapped IR LEDs
(2) IR detectors
(1) Set of Toddler Toes (Parallax #27312, not included in this kit)
(misc) wires

Toddler Toes kit (#27312) is not included in the stock Toddler Kit (#27310 or #27311) because it would have raised the retail price, prohibiting more robot enthusiasts from purchasing the kit. You can make your own toes using common materials.

This is what is included in the Parallax Toddler Toes Kit (#27312):

Parallax Part Number	Quantity	Description
550-27312	1	Twinkle Toes PCB, right side
550-27313	1	Twinkle Toes PCB, left side
700-00002	6	4/40 3/8” machine screw
700-00003	8	4/40 nut
720-27312	6	Bumper wire (extras for possible mistakes)
805-00002	2	14” servo extension cable

Assembling the Toddler Twinkle Toes

Figure 8.1 shows the schematic for the experiment.

Bending the Wires Bend the wire to match the outline of the bumper in Figure 8-2. You will need to use needle nose pliers to do this job.

The two curled ends need to make a 50% to 75% closed circle to keep from sliding off the screw. Try to keep the wire flat while making these bends. The distance from one of the curls to the outside bend is not to be exceeded. Follow the template exactly and the bumpers will not hit each other.

You have six wires with which to experiment. You only need two good bumpers so don't worry if the first ones don't come out quite right. And if you really destroy all of your wires you can call Parallax for a few more.

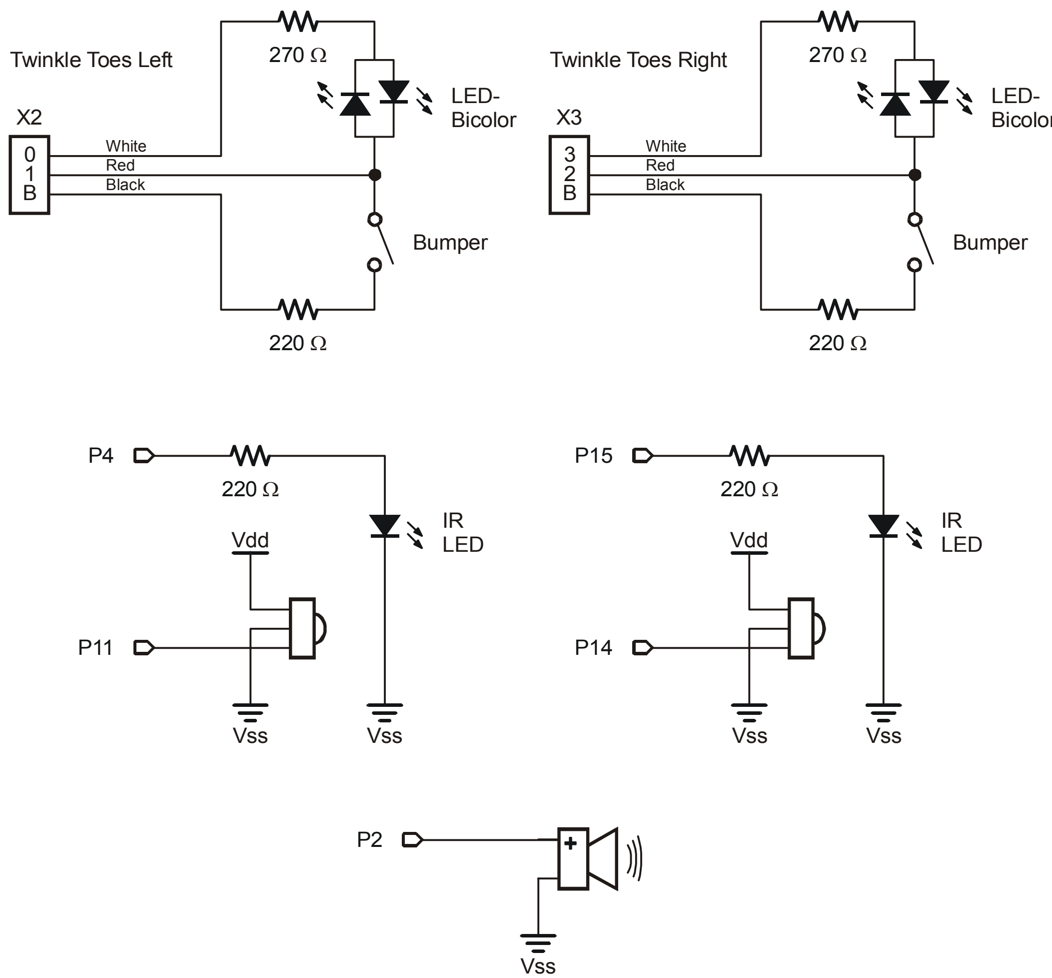

Figure 8-1: Real-Time Programming with Bumpers and Infrared

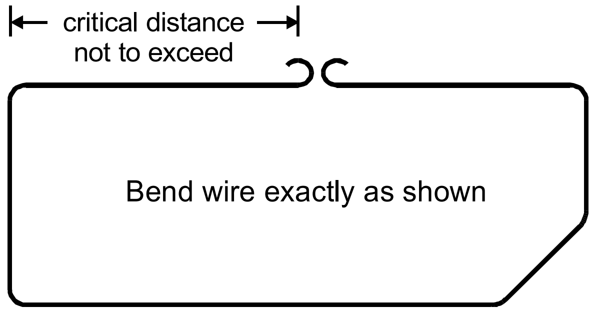

Figure 8-2: Wire bending template

Fine Tuning the Wire Tension Put the 3/8" screw through the center hole of each Twinkle Toes and secure tightly with a nut on top.

Attach the bumper hook ends over the screw you just mounted in the center of each board. The beveled corner goes on the side with the LED. Slip the wire in-between the prongs (two-post header pins) at each end of the board.

The bumper should be touching the pin closest to the front of the board. When pressing the bumper from the front the wire will touch the pin farthest from the front of the board. The pressure should feel very light to trip the bumper and will need to be adjusted until it is lightly springy.

Figure 8-3 shows how to adjust the bumper wire to be less stiff. For a final adjustment, take the bumper off, adjust and retest the tension. Once both sides of the bumper trip with a soft bump, secure with the locking nut. Do not tighten on the wire; the wire needs to be able to move freely around the screw.

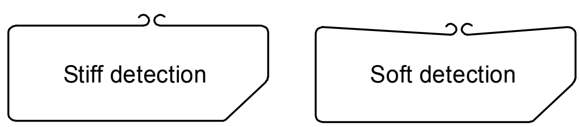

Figure 8-3: Bumper Tensioning

Figure 8-4 and Figure 8-5 show the final installation of the Twinkle Toes.

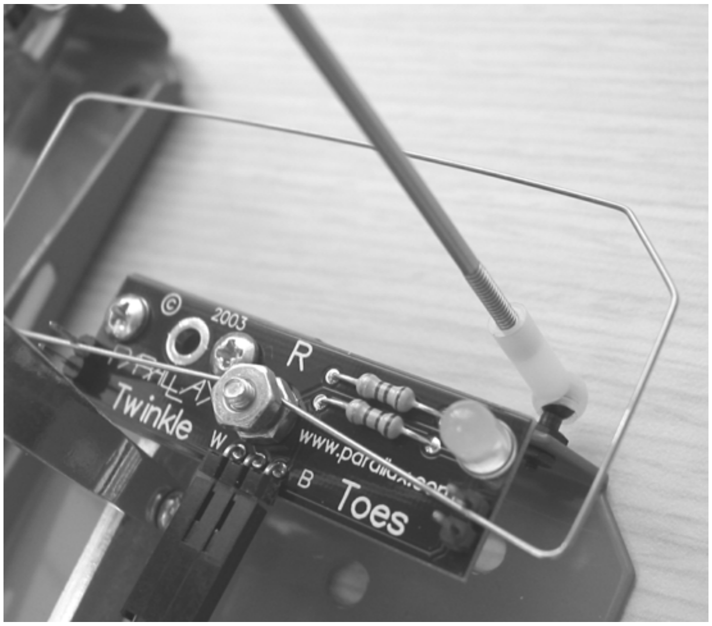

Figure 8-4: Installed Twinkle Toes

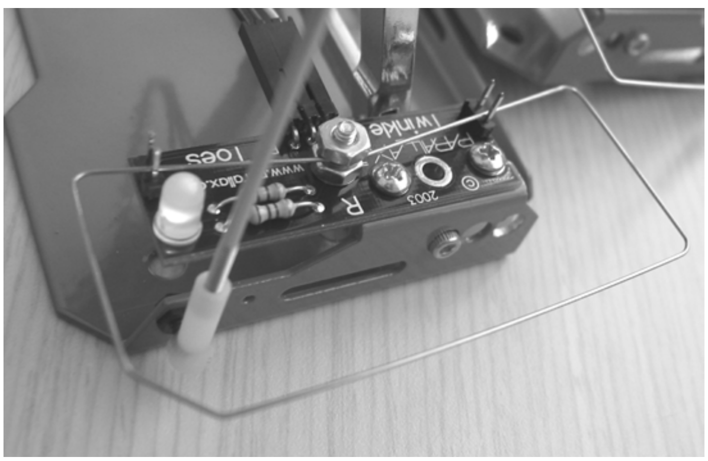

Figure 8-5: Twinkle Toes include the pushbutton resistors and bi-color LED.

Mounting Twinkle Toes Disconnect the tilt rods from the Toddler's left foot (the same side as the power switch). Mount the Twinkle Toes with two 1/4" screws (the third hole is only used if the ankle is mounted to the inside which would block one of the holes).

Make sure the bumper does not extend past the inside edge of the foot. If it does bend the wire or make a new bumper. One way to find out if the bumper is going to touch the opposite foot is to run one of the early basic walking programs in this text.

Reattach the control rod to the foot then mount the right sensor the same way and make sure it also does not extend past the inside edge of the foot. Repeat for the right foot.

Connect the cables Connect the cables from the sensors to X2 and X3 of the Toddler Board making sure the black to black connection (B to B) is maintained.

Testing the Toddler Toes

Let's test the Toddler Toes circuit before we proceed to do anything else.

```
' -----[ Title ]-----------------------------------------------------------

' Toddler Program 8.1 Twinkle Toes Test.bs2
' This program tests twinkle toes bumpers
' {$STAMP BS2}
' {$PBASIC 2.5}

' -----[ Initialization ]--------------------------------------------------

DIRS = %0000000000001001
OUTS = %0000000000001001

' -----[ Main Code ]-------------------------------------------------------

Check_Bumpers:
  DEBUG Home, "Right = ", BIN1 IN1, " Left = ", BIN1 IN2
  GOTO Check_Bumpers
```

If the Toes are properly installed and wired then you will be able to activate them and view the status of each bumper in the DEBUG window as shown in Figure 8-6.