News for users of BugBrain™ & ServoCenter™

Robotics Newsletter

"A learned blockhead is a greater blockhead than an ignorant one."

— Benjamin Franklin

Yost Engineering, Inc. **www.YostEngineering.com** **November 2006**

Ideas for Adding Features to Your BugBrain (submitted by other users)

- Make it hide from light or be attracted to light
- Have it keep watch at your door and sound an alert if someone comes in
- Use bump sensors to know to back up if it steps off an edge
- Use sonar to detect objects and avoid them or follow them
- Use sound activation to make BugBrain wake up when people talk to it

How to Play .WAV Files with Your BugBrain (Advanced)

The following tutorial will explain how to load a .wav file onto your BugBrain. It's not possible to store a very large file, but it is possible to play almost any short audio clip with your BugBrain. There is no quick and simple way to do this using the BasicX programming environment, but it can be done. This is not for beginners, so pay attention!

Step 1: OK, the first thing you'll need is a .wav file. Wav is a file format used to store audio information on PCs. Wav files usually contain uncompressed PCM (pulse-code modulated) data, which is what we need to get BugBrain's speaker to play our audio clip. Where do we get a .wav file?

There are lots of different programs that can convert various audio formats into .wav, but we are going to use Audacity. Audacity is a free digital audio editor, and can be downloaded at http://audacity.sourceforge.net. Even if you already have the audio clip that you want to put on your BugBrain in .wav format, you will still want to download Audacity, because we will be using it in several other steps (plus it's a pretty cool program).

After you have downloaded and installed Audacity, start it up. From Audacity's File menu, select Open, and browse to the audio file

(continued on page 3)

BugBrain: 101 — I/O (pronounced eye-oh)

Memory and the CPU are very important to every computer system, but they wouldn't be of much use without input and output devices, known collectively as I/O devices. Some examples of output devices are monitors, printers, and speakers. Output devices are responsible for presenting the data to us in a way that we can use, like a movie on a monitor or a song being played by speakers. Mice, keyboards, and joysticks are examples of input devices, which are how we talk to the computer.

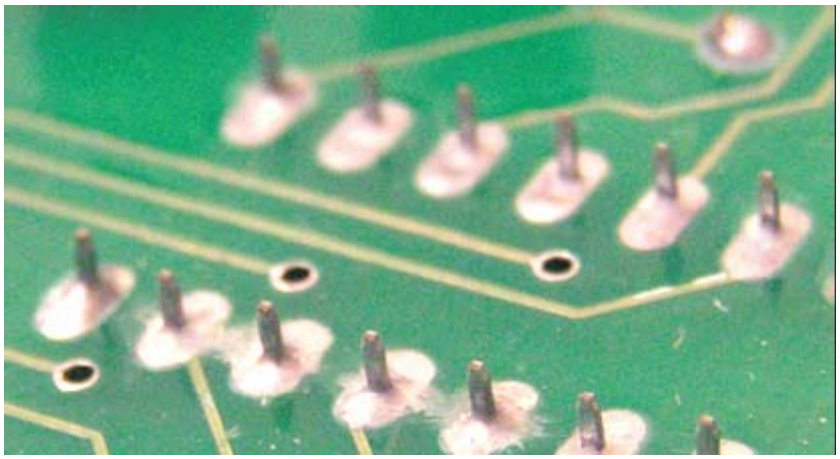
The BugBrain: 101 CoursePak is available for classroom, home school, or self-study. The coursepak is designed to teach robotics and programming basics, making use of the BugBrain robotic kit. Each month we will feature some quick tips or definitions, taken from the CoursePak.

Soldering Do's and Don'ts

There are two problems that cause 90% of the problems that are reported to us. The first is errors in the programming code (typos, misspellings, etc). The second is soldering problems during assembly. So let's take a brief look at basic soldering practices and how to tell if you are soldering correctly (for the purposes of electronics work).

What Does a Good Solder Look Like?

The joint has a smooth appearance, and the lead and pad look continuous. The solder does not overlap any other leads or pads. There is not a large blob of solder, just a smooth, rounded drop.



What Should I Fix?

If the joint looks bulky or connects multiple pads, re-heat the joint and remove the solder with a solder vacuum. Then re-solder it. If the joint looks fuzzy or dull, or if there is a bumpy or non-continuous surface between the pad and the component lead, you should also repair the joint by reheating, removing the solder with a solder vacuum, and re-soldering it.

How Can I Make Soldering Easier?

1. Adjust the board to get a good working angle.
2. Spread the legs of each component when possible, to hold it in place while you solder.
3. For small components that stick up from the board, use small objects to support the board and keep the component from tilting while you are soldering it.
4. Avoid overheating the components or traces of the circuit board. Overheating parts can damage them. In fact, overheating the traces can cause them to fall off the board!

How to Play .WAV Files with Your BugBrain (cont. from page 1)

of your choice. Audacity supports several different audio formats, but some formats, such as .m4a and .ram, are not supported. In our example, we converted a .mp3 to a .wav. If you want to convert from some other audio format, check Audacity's documentation to see if it's supported.

Step 2: Now that Audacity has opened your audio file, some wave forms will be displayed that represent the sound. In the bottom left corner of Audacity, there is a label which says "Project rate:", followed by a box containing a number. This is the frequency (in hertz) of the current project. Click in this box, and set the frequency to 8000 (or 4000). We have to lower the frequency because the BasicX processor can't correctly play audio at high frequencies. Lowering the frequency will also make our output .wav file smaller, which will allow us to place a longer sound on the BugBrain.

Step 3: Once you have lowered your Project frequency, from the Project menu, select Quick Mix. This will apply the frequency change to your audio data.

Step 4: At the top of your track, on the left, should be a box labeled Mix. Click on this box, and select Split Stereo Track. You should now see 2 separate tracks, one labeled Left and one labeled Right. Delete the Right track by clicking on the X in

(continued on insert page)

7 Top Troubleshooting Tips for BugBrains

If you have problems getting your BugBrain programmed for the first time, use these 7 tips to solve most of them!

1. Make sure you turn the power on before trying to download a program.

2. Make sure the servos are not plugged in backwards! The white wires should face toward the inside of the board.

3. Make sure the microprocessor is fully seated in the socket (it should be snug, not perched on the top).

4. Make sure the diode isn't soldered in backwards! The stripe should face the front of the board.

5. Make sure the capacitors (C1 & C2) near the serial port are the correct type — you will see very small numbers on them that say "104." They should NOT say "103."

6. Make sure the serial cable is connected to the right port for your PC. You might need to change the port setting in the BasicX environment to COM2 or another setting.

7. Finally, double-check all the solder joints on your board. Make sure no solders are touching each other or adjacent wires, and that there aren't cold solder joints (poor connection to a pin, not enough solder, uneven solder). Remember, electrical messages flow through these joints. Don't have leaking or interrupted messages!

How to Play .WAV Files with Your BugBrain (continued from page 3)

the upper left corner of the track. Now click on the box labeled Left, and select Mono. We want to get rid of the extra stereo data, since BugBrain only has one speaker, and we need our file to be as small as possible.

Step 5: From the Edit menu, select Preferences.... This will bring up a tabbed option box. Select the File Formats tab, and make sure that Uncompressed Export Format is set to WAV (Microsoft 8 bit PCM). Click OK to apply this setting.

Step 6: Now we can generate our WAV file. Select Export As WAV from the File menu. Enter a name for your .wav file, and Audacity will generate it. You can see what your wav file will sound like by clicking on the play button at the top of Audacity. Audacity may have generated a bunch of silence at the end of your track when you performed the Quick Mix. You may wish to delete this silence before you generate your .WAV, since it will make your file larger.

Step 7: We will be using the BasicX procedure PlaySound to play our .wav file. The syntax of the PlaySound procedure looks like this:

```
PlaySound(Pin,StartAddress,Length,SampleRate,RepeatCount)
```

Pin represents the pin number which is connected to the speaker (pin 15 on a standard BugBrain). StartAddress is where the beginning of your sound file is in the memory of the BugBrain, and Length is the size in bytes of your sound file. SampleRate is the frequency of your sound file, and RepeatCount is the number of times to play the sound file. You can go ahead and add a call to PlaySound in your code, filling in all of the values except for StartAddress – just enter a dummy value for your StartAddress for now, and compile your code.

Step 8: Open the .mpp file associated with your BasicX project in a text editor. On lines 16 and 17 of this file, you should see something like:

```
Code memory available:  32768 bytes
Code memory allocated:  46 bytes
```

Code memory available tells you how big your sound file can be (in this case, 32,722). If your .wav file is larger than this, you may wish to make it smaller by editing it with Audacity; however, you don't have to make your audio file smaller. You can simply put as much of it as will fit into memory, using a small utility available for download on the Yost Engineering website.

First, however, change the StartAddress for PlaySound to one more than the value of Code memory allocated; in our example, we changed it to 47. After you recompile your code, you will be ready to proceed to Step 9.

(continued on back)

How to Play .WAV Files with Your BugBrain (continued from front)

Step 9: Download the WavAppend.exe utility, available at:

http://tech.yostengineering.com/roboticsNewsletter

This program should be run from a command prompt, and takes three filenames as arguments. The first file should be the .bxb file which you just compiled. The second file should be the .wav file which you created. The third file will be the name of the output file - this will be the file that you will rename and copy over the first .bxb file. So your command line should look something like this:

WavAppend WavExample.bxb Song.wav Output.bxb

The WavAppend program will give you a warning if the size of your sound file + the size of your original .bxb file is larger than 32768 bytes, but the output file will still be generated; it will simply include as much of the sound file as is possible.

Step 10: Delete the .bxb file from your project directory, take your Output.bxb file and copy it to your project directory (if necessary), and rename Output.bxb to yourprojectname.bxb. You are now ready to download this file to your BugBrain.

Step 11: Open your project in the BasicX Programming Environment, and click the Lightning Bolt button on the Downloader. This will put the modified .bxb file on your BugBrain. Your BugBrain will now play your .wav when PlaySound gets called in your code! If you need to recompile your code for some reason, go back to Step 9.

Note: The length of the sound clips you'll be able to fit on your BugBrain is pretty short. At 8000 Hz, you can fit about 4 seconds of audio, and at 4000 Hz, you can fit about 8 seconds of audio.

