

## Hotrod Your Billy Bass

by Steve Ravet

Perhaps you have a "Big Mouth Billy Bass," given to you circa 2001. If so, its limited repertoire has probably grown stale. As a Gadget Freak reader it probably crossed your mind to teach your "fish" to sing new songs, such as Monty Python's "Lumberjack Song." Well, you can give Big Mouth Billy Bass an ARM-powered brain transplant so it will play .wav files from an attached micro-SD card, and you can choreograph its movements to match.

Billy Bass contains three small motors that when energized move the body, mouth, and tail. When a motor is not energized, a spring pulls the connected part back to its rest position.

Open Billy Bass and you'll see a printed-circuit board (PCB) that contains the driver circuitry, and soldered to the driver PCB a smaller control PCB. Familiarize yourself with the innards. Carefully desolder the 10 connections between the small control PCB and the driver PCB. Set aside the control PCB in case you want to restore Billy's original behavior. (You can replicate the original behavior using this Gadget Freak modification, too.)

**Table 1** lists the signals at the ten (10) contact points on the control PCB and the equivalent functions on the mbed ARM-processor board and the associated mbed-module pin you'll use in your modification. This table is for reference only: Don't make any connections just yet. (Before I removed the control PCB I "scoped" the signals to determine their function.)

Table 1. Control PCB Pinouts and Signals.

Control module pin	Function	MBED resource	MBED pin
1	GND	None	1 (GND)
2	Light sensor, dark=Vbatt	AnalogIn	15
3	Push button	DigitalIn	24
4	Tail motor	PWM	23
5	Mouth motor	PWM	22
6	Body motor	PWM	21
7	NC	None	None
8	???	None	None
9	???	None	None
10	Audio	None	None

I used eight (8) wires to connect the mbed board to the driver PCB. Contact point #1 is nearest the top of the photo and contact #10 is nearest the bottom of the photo. Contacts at positions 8 and 9 are not used.

Make the connections between the Billy Bass driver PCB and the mbed module as shown in **Table 2**. Note that the connection to pin 2 on the mbed module comes from the positive (+) end of the four-C-cell battery holder. This connection powers the mbed module.

Table 2. Connections Between Billy Bass and mbed Module.

Driver PCB signal	Function	To MBED pin
Control/Driver pin 1	Ground	1 (GND)
Vbatt, taken from the positive end of the 4 C cells	Power	2 (VIN)
Control/Driver pin 2	Light sensor	15*
Control/Driver pin 3	Push button	24
Control/Driver pin 4	Tail motor	23
Control/Driver pin 5	Mouth motor	22
Control/Driver pin 6	Body motor	21
Control/Driver pin 10	Audio	18*

\* See Notes at the end of this project description.

You can make a direct connection between the driver PCB and the mbed module or use a connector. I used an RJ-45 (Ethernet) punch-down connector glued to Billy through a hole cut in the back. This provided a clean installation and let me easily

separate Billy from the mbed module. The RJ-45 connector's 8 contacts can accommodate all of the signals listed in Table 2.

Outside Billy I cut apart a network cable to get from the RJ-45 socket to the mbed module on a breadboard.

Now, connect the mbed module to a micro-SD-card socket as noted in **Table 3**. The mbed module connects to the SD card through a basic 3-wire SPI port.

Table 3. Connections from a micro SD socket to an mbed module.

SD card socket pin	Function	To MBED pin
1	CS, chip select	13
2	DI, serial data in	5
3	VCC	40 (VOUT)
4	SCK, serial clock	7
5	GND	1 (GND)
6	DO, serial data out	6

### About the mbed Module

For complete information regarding mbed modules, tools, and documentation, visit: [www.mbed.co.uk](http://www.mbed.co.uk) where you will find an online Handbook that thoroughly explains how to download a bin file to your embed module. Once you have received an mbed module and have properly initialized it, you can start Billy's brain translate. Drag/drop the project (audio.bin) file onto the mbed device, which appears like an external USB drive to your PC. You will need to connect the mbed module to your host PC through a cable with a standard USB A-type connector on one end and a USB Mini-B connector on the other. I breadboarded the mbed module and wired a micro-SD-card socket to it. You could hardwire these connections on a prototype PCB.

If you wish, you can import my Billy Bass source-code file into the mbed online integrated development environment (IDE) and build your own .bin file. You must register on the mbed Web site to have access to the online development tools. The mbed Compiler lets you write programs in C/C++, and then compile and download them to run on the mbed MCU.

I used mbed's C++ class library to implement a basic WAV-file player. While playing back the WAV file my program reads a companion movement file that specifies which body parts to move

and when (in milliseconds) to move them. The format of the movement file is self explanatory.

If you want to choreograph Billy to a new WAV file, load the file into a WAV-file editor. I used the open-source program Audacity, available at: [audacity.sourceforge.net](http://audacity.sourceforge.net). Play the WAV file again and again and pause as needed to decide which parts of Billy to move and when to move them. Use the WAV-editor's cursor to note the elapsed time from the beginning of the file in milliseconds, and put this time information into the movement file. The mbed module uses pulse-width modulator (PWM) outputs to drive the body-part motors, and the movement file takes a percentage value from 0 to 100 for each motor's action. I found friction in the moving parts meant that body positions other than 0% and 100% were not very repeatable. Your Billy may vary, however.

After you have installed the .bin file in the mbed module, copy the WAV and movement files onto the micro-SD card. Insert the card into its socket, push the blue reset button, and off Billy goes!

This Gadget Freak project package includes two WAV files, bushfool.wav and clinton16.wav, I downloaded from the Internet, and companion movement files (bushfool.txt and clinton.txt) I created. The fish.txt file tells the mbed module where to find these .wav and .txt files. Enjoy them, but be sure to create some of your own.

#### **Parts list:**

MBED ARM Cortex-M3 based microcontroller development board:  
[www.mbed.co.uk](http://www.mbed.co.uk).

Big Mouth Billy Bass: Commonly available on ebay for \$10-\$15.

Micro-SD card socket: Easiest is the breakout version from [www.sparkfun.com](http://www.sparkfun.com), part number BOB-00544. It plugs directly into a solderless breadboard. Or a surface-mount socket, Allied part number 367-0344.

Solderless breadboard, Allied part number 237-0019.

RJ45 punchdown jack, Allied part number 805-0486.

#### **Files**

sound files:

09-30-09 Billy Bass  
10-02-09 at 1500 5

Final

[http://www.hendrielande.com/billy\\_bas/clint16.wav](http://www.hendrielande.com/billy_bas/clint16.wav)  
[http://www.hendrielande.com/billy\\_bas/bushfool.wav](http://www.hendrielande.com/billy_bas/bushfool.wav)

movement files:

[http://www.hendrielande.com/billy\\_bas/clinton.txt](http://www.hendrielande.com/billy_bas/clinton.txt)  
[http://www.hendrielande.com/billy\\_bas/bushfool.txt](http://www.hendrielande.com/billy_bas/bushfool.txt)

batch file: fish.txt

[http://www.hendrielande.com/billy\\_bas/fish.txt](http://www.hendrielande.com/billy_bas/fish.txt)

Binary file for mbed MCU: audio.bin - not included

**Notes for Table 2:** In my project I did not use the speaker-driver portion of the driver PCB, or the analog signal from the light sensor. I could not drive the audio input successfully, possibly because of an impedance mismatch, or perhaps my experimentation blew up the audio section. It used a computer speaker for the audio output and it sounded better anyway. In this project I did not implement the light sensor.