

LipSyncLab

Steve Tiffany

Copyright © 1998 Steve Tiffany

COLLABORATORS

	<i>TITLE :</i> LipSyncLab		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Steve Tiffany	May 28, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	LipSyncLab	1
1.1	Lip Sync Lab Guide	1
1.2	LipSyncLab2	1
1.3	LipSyncLab2	2
1.4	LipSyncLab2	3
1.5	LipSyncLab2	4
1.6	Lip Sync Lab	4
1.7	Lip Sync Lab	5
1.8	Lip Sync Lab	5
1.9	Lip Sync Lab	5
1.10	LipSyncLab1	5
1.11	Lip Sync Lab	6

Chapter 1

LipSyncLab

1.1 Lip Sync Lab Guide

```
LipSyncLab2 -- For synchronizing mouth shapes to sampled speech.  
               © 1992,1994,1998 Steve Tiffany
```

TABLE OF CONTENTS

Introduction

How To Use LipSyncLab2

What To Do With The Data

Why don't you use .abks?

Quality Concerns

Known Bug

Code Acknowledgment

About LipSyncLab1...

My Other Stuff on Aminet

I'd really appreciate hearing from anyone who uses this.

Contact the Author

1.2 LipSyncLab2

INTRODUCTION

LipSyncLab2 is for AMOS programmers who want to make characters

speak using sampled speech, not the SAY command. There are four AMOS programs in the archive:

```
LipSyncLab2 -- The tool that generates the data.
Floyd -- Example of using the data with bobs.
Buster -- Example of using the data with HAM screens.
LSL1 -- Bizarre earlier attempt at lip sync using sampled phonemes.
```

LipSyncLab2 is part of a whole approach to lip sync that was designed for my own specific projects, and may or may not be useful for yours. The main limitation is that it only works with samples shorter than about 3 seconds. (One could possibly modify LSL2 to scroll horizontally through longer samples, displaying perhaps one second of the sample onscreen at a time, but it wouldn't be a real easy change to make.)

The sample length limit isn't a problem if you construct sentences out of single-word or short-phrase samples. Being able to use each sample in a number of different contexts saves disk space. In the included example programs, Buster and Floyd sound stilted because they use almost exclusively single-word samples, but sentences made of phrase-length samples combined at natural pause points can sound seamless.

1.3 LipSyncLab2

HOW TO USE LIP SYNC LAB 2 -- a brief tutorial

Run AMOS, load LipSyncLab2.AMOS. If it's not in the directory work:LipSyncLab/, change the value of CURRDIR\$ to reflect its true location. Run the program and center the screen if it asks you to. If you somehow centered it wrong and then saved it, delete the file "S:AmosXyOffsetNTSC.config" and run the program again.

Click on Load Next Sample, and pick the sample "think." Now there's a line of green boxes under the sample waveform, and the first box is highlighted in orange. Click on mouth number 2, since that covers the "th" sound. Then play the sample again and watch for where the "ih" sound seems to begin. Click the green box beneath that spot, then click on mouth 3, which covers the "ih" sound. Then watch for where the "n" begins and click the green box under that spot, then click on mouth 2, which covers "n." Note that "n" and "k" both use Mouth 2, so don't bother adding another Mouth 2 for the "k" -- there would be no point.

A couple things to know about the interface:

- * You don't have to move the mouse all the way up to the "Play" and "Play Slow" buttons, you can click on the sample itself -- above the line for slow, below the line for regular speed.
 - * You can clear a green box by clicking on it, holding down the
-

left mouse button, and moving the pointer toward the bottom of the screen. You can also set the number by clicking, holding, and moving up the screen til the green box displays the number you want.

- * Sometimes the end result looks more natural if you put the mouth shape a couple boxes to the left of where your ear tells you it should go. Experiment.

When the first sample's lip sync looks good, load another. When you quit, there will be a file in Ram: called SampleMouthData.asc, which you merge into your target program.

Note: I like to work with raw samples, all sampled at the same rate. If you use raw samples, be sure to change the value of PROJFREQ to equal the sample rate for your project. It's set in the source code at the top of the Globals. The samples that the example programs use are all raw, and should be played at 13982 samples/second.

1.4 LipSyncLab2

WHAT TO DO WITH THE DATA IT GENERATES

The file ram:SampleMouthData.asc consists of a series of lines in this format:

```
_FILENAME: Data SampleLength,NumberOfMouths,MouthNum,Wait,MouthNum,Wait, etc.
```

The way I handle it, these wind up in your program's AD[] procedure, where a WriteScene procedure uses them to write a pair of "script" files: one that holds a series of sample names, and one that holds the numbers. When the files are completely written they get closed, then the PerformScene procedure opens them up to be read. First it reads the sample names and loads a sentence-worth of samples into banks. Then, for each sample in the sentence, it reads the SampleLength and the NumberOfMouths, then plays the sample and starts a For-Next loop to animate the mouth. The loop goes something like this:

```
For J=1 to NumberOfMouths
  Input MouthNum : Input MouthWait
  Paste Bob MouthNum : Wait MouthWait
Next J
```

This mouth-display loop ends exactly when the sample is done playing. Floyd.AMOS demonstrates the Paste Bob approach, and Buster.AMOS shows the method of opening 7 screens and using Screen To Front MouthNum. This only makes sense in HAM, where you can't use bobs. To reduce archive size, I've cropped the HAM graphics to include only Buster the Talking Bull Terrier, but full screen 4096-color overscan can look really good, especially if you show it on a television, which is just muddy enough to minimize the pixels and HAM fringing.

1.5 LipSyncLab2

WHY GET SAMPLES OFF THE DISK? WHY NOT STORE THEM IN .ABKS?

In my talking head program Steve Headroom, I had all the samples and graphics stored in .abks, and while it eliminated disk access once the program was loaded, it also meant that people with under 2 megs of chip ram couldn't use the program. Even with 2 megs chip ram crammed to the gills, the program was limited to a vocabulary of 190 sampled words. If you load samples off the disk, vocabulary size is unlimited. Of course, drawers full of IFFs and samples do make your program vulnerable to hacking. Even if it were distributed compiled, anyone with a sampler could change the words your characters say, just like anyone with a paint program could give them mustaches and funny hats.

Though the LSL system is currently geared towards loading samples off the hard disk between sentences, there's nothing to prevent you from changing it so everything gets held in banks.

1.6 Lip Sync Lab

QUALITY CONCERNS

The decision to use 7 mouth shapes is a compromise based on the fact that I wanted LipSyncLab2 to work for both bobs and HAM screens, and AMOS only lets you open 8 screens. (I didn't use 8 mouths because to alternate between two characters in HAM, you need to use an unbuffered Screen 0 as a loader screen, with Screens 1 through 7 double-buffered, one character on each buffer. To change characters you do a series of Screen Swaps. Without a loader screen you get horrible flashes when you load a different view of the hidden character.)

If quality is important and you're using bobs exclusively, you may want to modify LSL2 to add a couple mouth shapes. There'd be room to add 3 without changing the interface much. Consider one for TH, one for SH and CH, and one for the vowel sounds AH and EYE, as in "hot" and "height," which could stand to open a little wider than Mouth 3.

In the stop-motion feature "The Nightmare Before Christmas" they used 11 different mouth shapes for each of the main character's expressions. Of course, he had to sing! I think using 7 mouths can look convincing if the mouths are shaped right.

Finally, no matter how many mouth shapes you use, it wouldn't be a bad idea to include a bit of jawline in your mouth bobs. Though plenty of low-budget kiddie cartoons have characters like

Floyd who speak with only their lips moving, it really does look better when the jaw moves too.

1.7 Lip Sync Lab

KNOWN BUG

If another program has called on the Amiga's sound-generating chips, it sometimes causes AMOS programs to turn a sample into a corrupted rapid looping sound. If this happens you might as well reboot, then run AMOS right away without letting any other programs make sound first.

1.8 Lip Sync Lab

ACKNOWLEDGMENT

Major chunks of the code for LipSyncLab2 were lifted from the program Sample Bank Maker by François Lionet, which is included with AMOS. It would have taken me forever to figure out how to display a sample waveform without it.

1.9 Lip Sync Lab

Also on Aminet from the creator of Lip Sync Lab:

For AMOS programmers:

dev/amos/ArtCoder.lha	(super-handy for interface design)
dev/amos/AnimFlipper.lha	(includes screen-centering code)

Assorted bits of general-interest whimsy:

misc/misc/SteveHeadroom.lha	(talking head w/pre-LSL sync)
text/misc/IconPoet.lha	(fun writing toy)
misc/misc/SeasonInRAM.lha	(poetry generator)
demo/slide/PictureGarden.lha	(collage generator)

1.10 LipSyncLab1

EARLIER LIP SYNC ATTEMPT

LipSyncLab1 was a spectacular failure which attempted to build words out of sampled phonemes, and I'm including it in this archive as a curiosity. There was just one problem with LSL1 -- nobody could understand a word it said. What's kind of neat is the way it translates from English. Its initial translation is only a starting point, however. To even approach intelligibility, you need to go through and lengthen many of the Wait times, and add emphasis by increasing the volume on certain syllables. Change values on the active gray bar by clicking on them and moving higher or lower on the screen with the left mouse button held down.

One could perhaps make it more understandable using a different phoneme set, but I doubt this approach will ever be even half as good as the Amiga's synthesized speech, circa 1985. Note that I tried half a dozen different phoneme sets before giving up on it. (Memo to Gateway: please bring back translator.library and narrator.device -- too many newer Amigans don't even know their computers can talk!)

1.11 Lip Sync Lab

CONTACT THE AUTHOR

Please drop me a line if you find this at all useful. If you use it for something that winds up on Aminet, please alert me when you upload it.

Steve Tiffany -or- stiffany@isd.net
3257 14th Ave S #4
Minneapolis, MN 55407
USA

(addresses current April 1998, and probably long after)