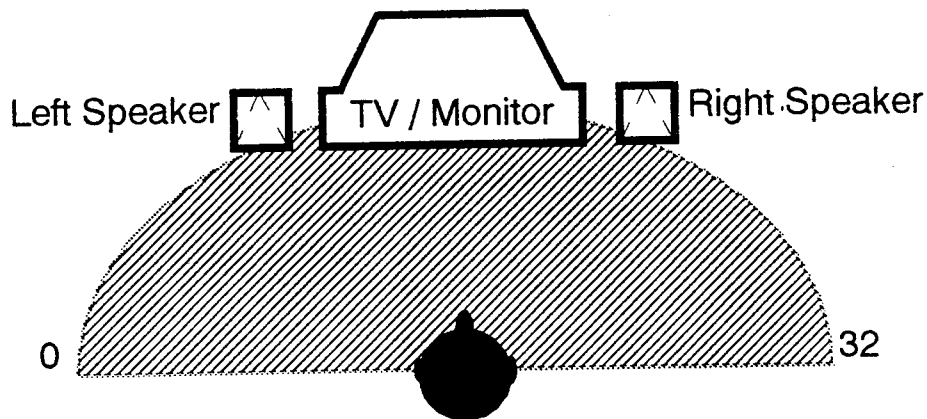


QSound™ For The Atari Jaguar

QSound is a patented, innovative process for generating a sound field that is not bound to the playback speakers. It requires only traditional stereo playback equipment for reproduction, and provides enhanced audio imaging capabilities with startling contrasts.

Using the QSound process, sound sources can be placed in "virtual space": an arc approximately ± 90 degrees in front of the listener, well outside the speakers. The QSound pan positions which map this space are numbered 0 (far left) to 32 (far right).



For game developers, QSound provides a rich environment for audio interfacing. For example, enemy fire can be heard in QSpace before the enemy appears on the screen; missiles launched off an F-16 jet fighter can be heard to drop off the wing tip before they race off into the distance; when you drive or fly past an explosion, it can appear to move beyond the player; background music can be given extra ambiance and depth.

Using QSound For Jaguar

There are two ways of using QSound for Atari Jaguar games:

For sounds which can be preprocessed and require no dynamic control of position, the QSystem II QCreator program can be used¹. The QSystem II is a sophisticated hardware & software post production mixing system which results in stereo output. QCreator is a software-only tool which runs under Microsoft Windows and allows developers to QSound process mono sound samples in WAV, RIFF, and raw sample formats. The result is a stereo sample which will include the QSound effect when played.

Sounds processed with QCreator can be played at runtime with no further processing required. However, because the samples are 16-bit stereo they will take up more room than using 16-bit mono.

QCreator program is available to Jaguar Developers from either QSound or Atari Jaguar Developer Support upon request. For more information about QCreator or to inquire about the Qsystem II, please contact QSound directly at the address given at the end of this section.

There:

1. For or pro run AI eff

So Ho

- 1 The requ add

samples processed at runtime. Note also that using lossy sound compression techniques on QSound processed files will probably result in the QSound effect being altered or lost completely.

Because they require no additional processing at runtime, pre-processed samples can be used in conjunction with the Jaguar Synth & Music driver.

2. For sounds which are to be panned dynamically at runtime, The QSound Q1 module has been implemented on the Jaguar DSP. The Q1 module takes 16-bit monophonic sound samples and creates 16-bit stereo output with the sounds positioned in 3D space using the QSound effect.

Because the QSound module must be running in the Jaguar DSP to process the samples at runtime, your ability to otherwise use the DSP at the same time is limited. For example, the Jaguar Synth & Music Driver cannot be used at the same time.

One advantage to using the Q1 module instead of pre-processed sounds is that the files will take up half as much room because you have mono samples instead of stereo. And although the sample program doesn't do it, lossy compression techniques can be used to further reduce the storage requirements. Or you could even use plain 8-bit mono samples as your starting point and expand them to 16-bit before passing them to the Q1 module.

It's entirely possible to use both options in the same program. For your title screen and option screens you might have some preprocessed QSound effects built into samples that are played as part of a music program done by the Jaguar Synth & Music Driver. Then during your game play, you could have the QSound Q1 module loaded so that you could dynamically position your sound effects in 3D space.

Regardless of which options you choose, the starting point must be a monophonic sound sample. This can be created or edited using whatever digital sound sampler & editor you choose. This can be something like the utilities that come with many PC sound cards, or something more sophisticated. The main requirement is that you must be able to create files in either the RAW format that you would link in with your Jaguar program or files loadable by the QCreator program.

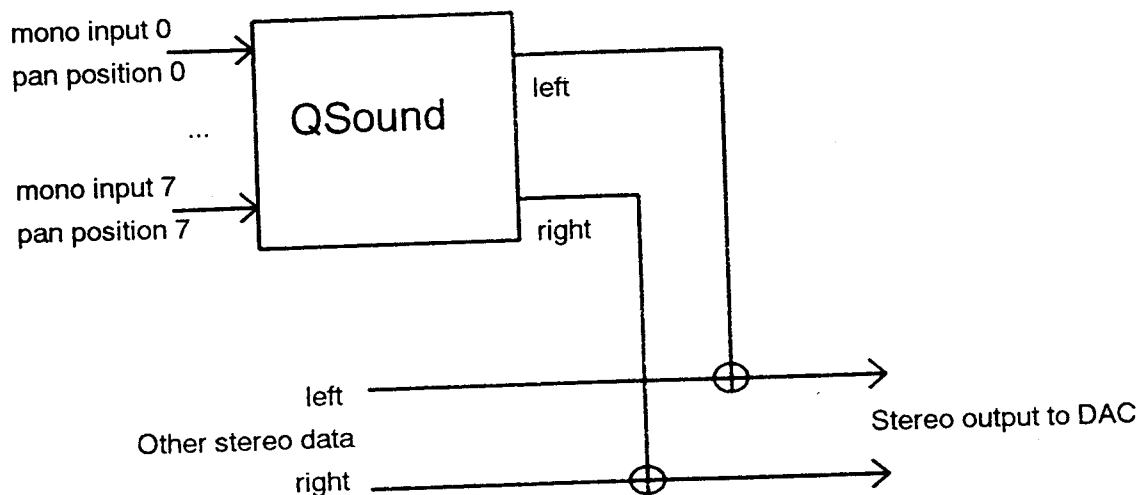
Q1 Module For Jaguar DSP

The implementation of the dynamic Q1 module on the Atari Jaguar system can be viewed as a black box with a single entry point: the *QSound* function running in the DSP. The *QSound* module can process up to eight independently panned mono voices. Regardless of the number of inputs, the output is always a stereo stream, which may be mixed with other stereo data before it is played back through the I2S interface.

Note: There is **no** internal volume scaling of the input samples within the *QSound* module. It is the responsibility of the caller to do the required volume scaling of voices to ensure that overflow does not occur.

The QSound process is dependent on the sampling rate. The current implementation is for the default sampling rate of the DSP, which is a shade under 22050 Hz (SCLK set to #19). If you are running at

any other sample rate, then please contact QSound Labs and we will provide an appropriately adjusted module for your desired sample rate.



Descriptions of the routine follows. For further information or technical help, please contact Buzz Burrowes at QSound.

The QSOUND.OT Module

The file **QSOUND.OT** is a linkable object module containing the QSound routines. This file must be linked with your program, and at runtime, the routines must be loaded into Jaguar's DSP. It has a single entry point which is documented below. See the documentation on the sample program for more information.

The QSound module is designed to be completely position-independent. You can load it anywhere in DSP memory where room is available. Usually, it follows with other DSP code supplied by you which feeds samples to the QSound module. See the demo program for an example.

QSound Function

Summary:	The QSound function is called every sample period in which at least one QSound voice is active. Typically this means once per sample (typically 22050 times per second).
Input:	<p>r16 = return address r17 = number of QSound voices to process (1 to 8) r18 = Pointer to an array of structures which define the input sample and pan position for each voice. The structures look like this:</p> <pre> struct QSound_Voice /* Values use only low 16 bits of LONG */ { long sample; /* Sample to be processed */ long pan_position; /* values from 0 (left) to 32 (right) */ } </pre>
Output:	<p>r20 = left channel of stereo output (32 bits) ready to be fed to Jaguar's I2S interface r22 = right channel of stereo output (32 bits) ready to be fed to Jaguar's I2S interface</p>

Register Usage:	uses r12 through r27
Notes:	Requires/uses about $(140 + (27 * \text{num_voices}))$ instructions.

Example:

; copy 16 bit inputs to #samples

```

load    QSound_ptr,r5      ; Get stored address where we put QSound module
movei   #After,r16         ; return address for QSound
movei   #1,r17             ; number of voices
jump    T,(r5)             ; call QSound module
nop
move    #toQSound,r18      ; r18 -> input samples/pan pairs

```

After:

```

shrq    #16,r20            ; outputs in 16 bits for I2S Interface
shrq    #16,r22
...
; store results for processing at next I2S interrupt

```

toQSound: ; up to 8 consecutive 2*32 bit locations

```

.ds.l   1      ; voice 0 sample
.ds.l   1      ; pan position for voice 0

```

```

.ds.l   1      ; voice 1 sample
.ds.l   1      ; pan position for voice 1

```

```

.ds.l   1      ; voice 2 sample
.ds.l   1      ; pan position for voice 2

```

```

.ds.l   1      ; voice 3 sample
.ds.l   1      ; pan position for voice 3

```

```

.ds.l   1      ; voice 4 sample
.ds.l   1      ; pan position for voice 4

```

```

.ds.l   1      ; voice 5 sample
.ds.l   1      ; pan position for voice 5

```

```

.ds.l   1      ; voice 6 sample
.ds.l   1      ; pan position for voice 6

```

```

.ds.l   1      ; voice 7 sample
.ds.l   1      ; pan position for voice 7

```

How To Contact QSound Labs

QSound Labs Inc.
 2748 - 37 Ave N.E.
 Calgary, AB, Canada
 T1Y 5L3

Tel: (403) 291-2492
 Fax: (403) 250-1521

QSound For Jaguar

Buzz Burrowes
QSound Labs, Inc.
2521 Ripley Avenue
Redondo Beach, CA 90278

Tel: (310) 374-8017
Fax: (310) 374-0998

Notice

QSound technology is protected by patent and copyright laws. Its use on the Atari Jaguar system is restricted to, and subject to, the license agreement signed with Atari.

All third parties interested in using QSound in Jaguar applications should check with Atari regarding this licensing agreement.

QDEMO - QSound Demo Program

The QDEMO program demonstrates how to use the QSound module to play back different samples and position them in 3D-space in real-time. You use the joystick to control the location of the sounds in 3D-space.

Below is a list of all the files which make up the QSound demo program. In order to reduce the size of the archive containing the demo, the executable program itself is not provided; the project must be built using the tools in your Jaguar developer's kit.

HELICOPTER.SND	Sound file used by the program (the helicopter). This is a raw 16-bit mono sound sample file (sample rate about 20khz). Included at link stage by using -li option of ALN.
DEMO.S	This is the code module for the demo program where things happen. This copies the bitmap for the background picture into RAM using the blitter, then goes into a loop which reads the joystick and cooks the values for the QSpanner routine.
JOYPAD.S	This file contains the joystick reading routines used by the program
MAKEFILE	Used with MAKE utility to build executable program file from source code and data files.
EXPLOSION.SND	Sound file used by the program (the explosion). This is a raw 16-bit mono sound sample file (sample rate about 20khz). Included at link stage by using -li option of ALN.
GUNSHOT.SND	Sound file used by the program (the gunshot). This is a raw 16-bit mono sound sample file (sample rate about 20khz). Included at link stage by using -li option of ALN.
QDEMO.LNK	ALN linker include file specifying names of files to be linked into demo program.
QDEMO.MAC	This file takes control after the startup code has initialized the system. It creates an object list for the background picture, installs an object list for the background picture, and then calls the qdemo routine in DEMO.S.
QPANNER.DAS	MADMAC Source code file containing DSP interrupt routines and demo program's interface to QSound function.
QSOUND.INC	MADMAC include file containing declarations of labels in QSOUND.OT module
QSOUND.OT	This is a BSD-format object module containing QSound routines. Linked with demo program or with your own program to provide the QSound capabilities.
STARTUP.RGB	This file is actually in the \JAGUAR\SOURCE directory. This is the screen displayed by the startup code that is used by several of the sample programs in the Jaguar Developer's Kit.

25 April, 1995

Confidential ~~AT&T~~ Information

Filename	Description
STARTUP.S	Standard Jaguar Startup Code. This module contains all the code necessary to properly initialize the Jaguar hardware and display a simple startup picture. Then it passes control to the <code>_start</code> label in the QDEMO.S module. <i>(See the Sample Programs section for further information on the Standard Jaguar Startup Code.)</i>
VALOGO16.PIC	Binary image of picture to be displayed by demo program. This is a raw image file containing no header. The image itself is 320 pixels wide by 200 pixels tall, 16-bit Jaguar RGB format. Included at link stage by using <code>-ii</code> option of ALN.
VIDSTUFF.INC	MADMAC include file containing miscellaneous equates used by the demo program's object list setup

Below is a more in-depth description of some of the main files from this demo program.

STARTUP.S

This file is where the program execution begins. This is the standard Jaguar Startup Code responsible for initializing the system. It sets up interrupts, sets the video registers correctly for either NTSC or PAL, and does other related things that must be done properly at startup time for your program to function. It also displays a startup screen. Once it is finished, it passes control to the `_start` label somewhere in your program (QDEMO.S in this example).

Note that STARTUP.S has been modified slightly from the version in `\JAGUAR\STARTUP` to allow the use of a different startup picture. This type of change is only one allowed in this file. Making changes to other portions of the file may result in errors which can prevent your program from functioning properly.

QDEMO.S

This file is where the program execution begins after the startup code has initialized the system. It basically delays for a few seconds so that we can look at the startup screen, then it creates an object list for our background picture, installs an interrupt handler to refresh the object list, and then sets the video mode to 320-pixel RGB mode. Finally, it clears the memory that will be used for our bitmap, and then jumps into the `qdemo` function, located in DEMO.S.

Note that the object list creation routine `make_list` is almost identical to the routine `InitListner` in the STARTUP.S module. The only parts that changed were the labels for the address where the list information is stored.

VIDSTUFF.INC

This file contains a number of program-specific equates that describe the video and object list requirements on the program. (Such as the memory location to be used by the object list, what are using in our object list.) This is used by QDEMO.S.

JOYPAD.S

This file contains the *readpad* routine that we use to read the joypad controller. The joypad data is only read by this routine, not interpreted. The *readpad* routine outputs one variable which describes the current joypad reading and another that indicates what's changed on the joypad since the last time we read it (buttons being pressed or released, etc.).

This file is essentially the same as the one used by the 3DDEMO sample program.

DEMO.S

This is the main program-specific part of the source code. The *qdemo* routine starts off by blitting our picture from ROM into RAM so that it can be displayed (displaying bitmaps directly from ROM is a big waste of bus bandwidth).

Next it starts the main helicopter sound, and then jumps into a loop where it reads the joypad values (by calling the *readpad* function), and calls the *interpad* function.

The *interpad* function is responsible for interpreting the joypad values and taking the appropriate action: it sets the pan positions of the sounds, and starts a gunshot and explosion sound if the 'B' button is pressed.

QANNER.DAS

This file contains source code for the Jaguar DSP. The *QSWrapper* function enables the Jaguar I2S interrupt, which is acting as the sample rate timer for our sound samples. Then it calls the *QWave* function.

The *QWave* function reads data from the sound samples being played, figures out the current pan positions, and then feeds this information to the *QSound* routine in the QSOUND.OT module, which then processes it. When an I2S interrupt occurs (about 22050 times per second), the processed samples are output to the I2S interface so we can hear the wonderful 3-D sound effects that QSound is capable of producing.

Also contained in this file is the source for the DSP interrupt routines. In many other DSP applications, the I2S interrupt would grab the current set of samples and feed them to the I2S interface (i.e. play the sound). But because *QSound* has to pre-process each set of samples, we do things a little differently. The I2S interrupt simply sets a semaphore that the main *QWave* function uses as a flag to indicate that we are ready to hand one set of samples off to the I2S interface (i.e. play the sound). As soon as this is done, it sends another set of samples off to the *QSound* function to be processed.

QSOUND.INC

This file contains declarations for the QSOUND.OT module (so you can figure out the length of the code before you copy it into the DSP). See DEMO.S for an example

VALOGO16.RGB

This is a raw binary file containing the picture which we display on screen during the demo program. It is an RGB picture with dimensions of 320 pixels wide, 200 pixels high, and 16 bits per pixel.

It is included and assigned a starting label and an ending label by using the **-ii** function of the ALN linker.

MIX3.SND, COPTER.SND, & PHASER.SND

These files contain the three raw mono 16-bit samples that will be played and passed through the QSound module. Note that the order these are specified in the link is important, as the PHASER and MIX3 sounds are sometimes played together as a single sound. If they aren't consecutive, this won't work correctly.

You may wish to substitute your own 16-bit mono sample files in order to see the results with QSound on the Jaguar.

These files are included and each assigned labels by using the **-ii** function of the ALN linker.