QuBit Documentation

Benjamin Moore

ICT Technical Report
No: ICT TR 01.2003

QuBit Documentation <back>

This is a collection of documents related to the QuBits. These cover basic information about the QuBits as well as specific procedures(HOWTOs) for performing operations with the QuBits.

QuBit Playback HOWTO
QuBit Recording HOWTO
QuBit Extraction HOWTO
QuBit Demo Button HOWTO
QuBitTools Reference

Software

- QuBitControl v0.2
- QuBitExtractor v0.1

Conacts at QuVis

Phone: 800-554-8116
George Scheckel
- VP Digital Cinema & Content Production
# He's the local contact with QuVis is you need something and it is
# REALLY important!
Email: gscheckel@quvis.com
Work: 818-704-6873
Cell: 818-917-1439

Robert Stratton
- Director of Systems Integration
# Robert is our contact point for QuBit related help. Generally I
# contact him by sending mail to support@quvis.com and CCing him.
Email: rstratton@quvis.com
Work: 800-554-8116 x123
AIM: rstratton

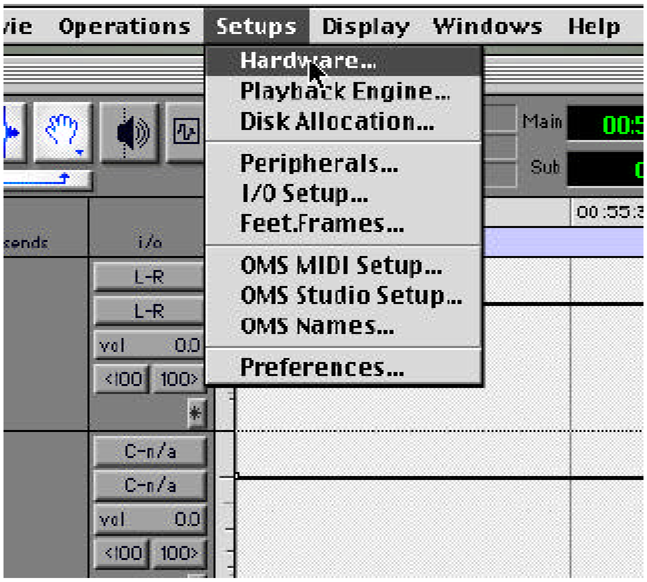
Running Demos on the QuBit Step by Step

Configuring the Room

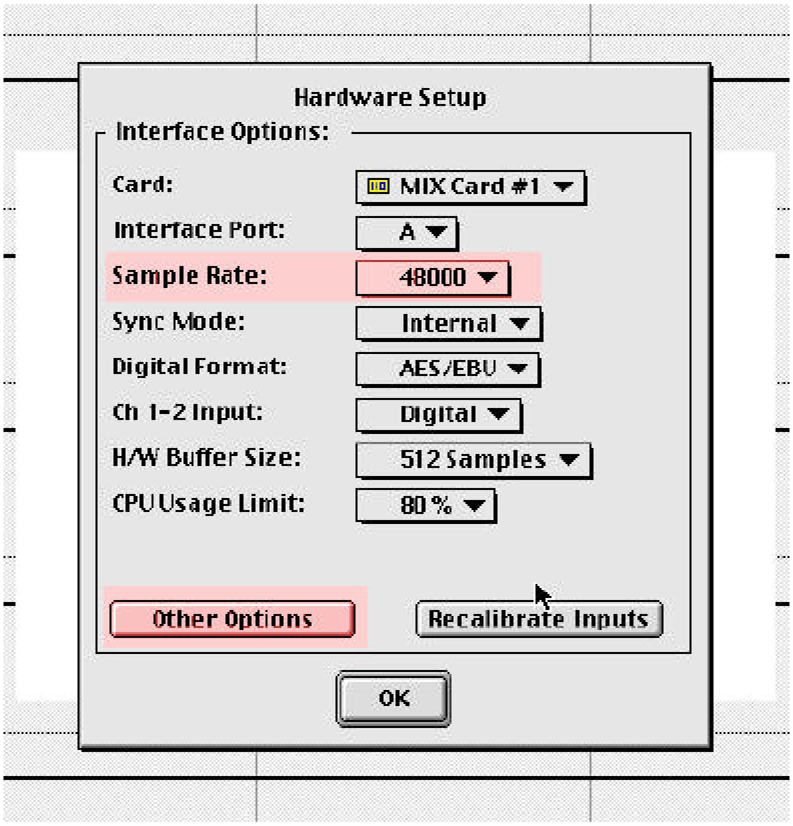
1. Turn on the projectors
2. Set inputs to QuBit1, QuBit2, and QuBit3

Configuring Protools

1. Start QuBit session on the Mac(for vistarama demo run Vistarama session) (double click QuBit or Vistarama icon on desktop)
2. Setup Protools Hardware
    1. Go into the Setup | Hardware dialog



    2. If not already, set the sample rate to 44100



Hardware Setup

Interface Options:

Card: MIX Card #1

Interface Port: A

Sample Rate: 48000

Sync Mode: Internal

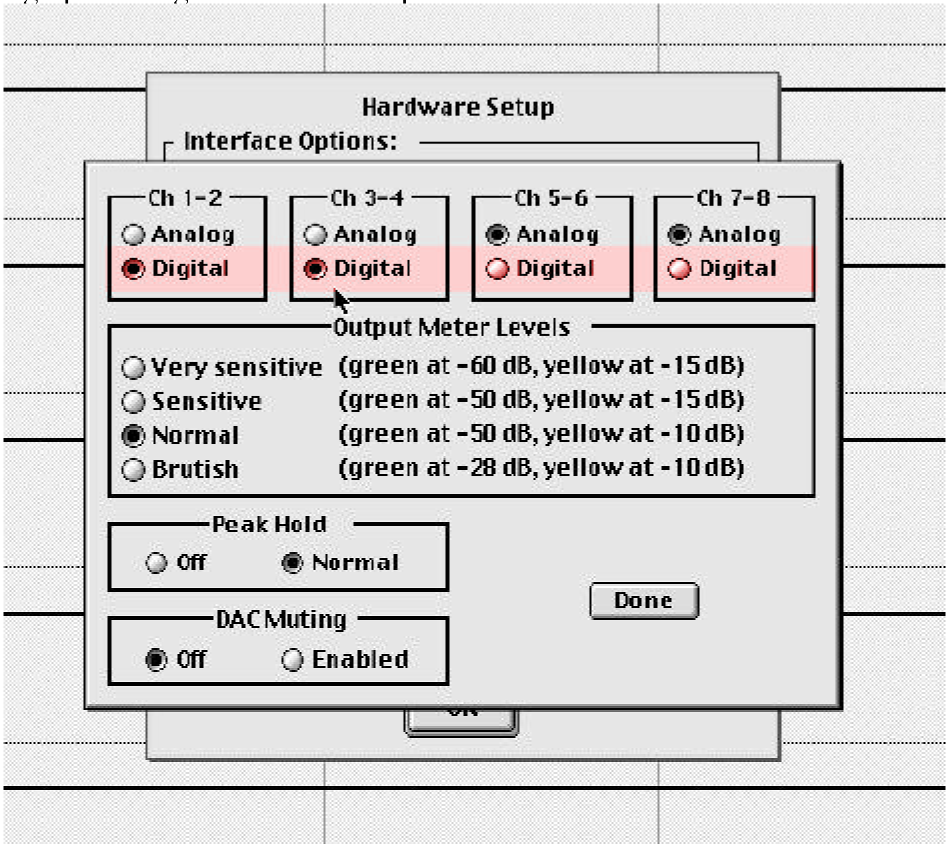
Digital Format: AES/EBU

Ch 1-2 Input: Digital

H/W Buffer Size: 512 Samples

CPU Usage Limit: 80 %

3. For each Interface Port(A and B) on each MIX Card (#1 and #2) hit the "Other Option" button This will bring up a dialog to control the input source of an 888. Set all channels to use their digital inputs



Hardware Setup

Interface Options:

Ch 1-2: ○ Analog ● Digital

Ch 3-4: ○ Analog ● Digital

Ch 5-6: ● Analog ○ Digital

Ch 7-8: ● Analog ○ Digital

Output Meter Levels

○ Very sensitive (green at -60 dB, yellow at -15 dB)
○ Sensitive (green at -50 dB, yellow at -15 dB)
● Normal (green at -50 dB, yellow at -10 dB)
○ Brutish (green at -28 dB, yellow at -10 dB)

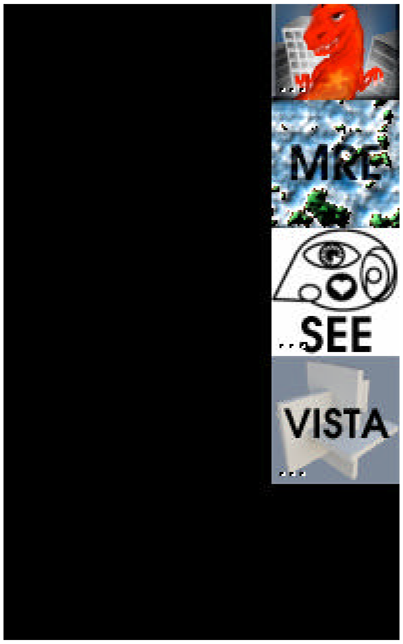
Peak Hold: ○ Off ● Normal

DAC Muting: ● Off ○ Enabled

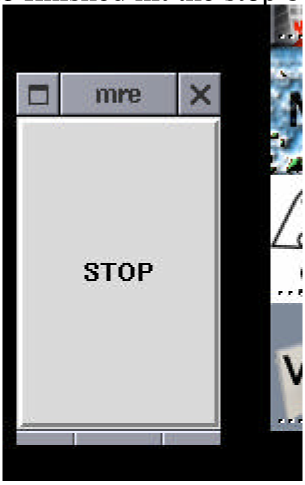
Running the demo

1. Log into kenshin(logitech keyboard and flat panel) as dimr(you should see icons for the various demos at the

right)



2. Double click on the icon for the demo you want to run
3. Until QuViz send us fixes check that the QuBits are responding correctly
   1. after the stop button comes up watch the qubits The [Play] or [Record] will turn to [Busy] and then switch back. If any of them don't hit the Big Stop Button and power cycle the qubits(the power switch is in the bottom righthand corner on the back of the qubit). Repeat from step 2.
4. After the demo finished hit the stop button. (Don't forget to do this! nobody else will be able to access the qubits if you don't)



Playing Back a Clip Without a Button

Clips without buttons can also be played back. This can be done using the runclips script. It lives in ~dimr/qubit/QubitControl/ on kenshin. To run it:

1. cd ~dimr/qubit/QubitControl/
2. ./runclips <clipname> [<clipname>]
   - EX1: ./runclips MREtest5
   - EX2: ./runclips vistarama parthenon
3. When the clips finish press the stop button.

This program is the same one which is used for running clips when they are attached to a button.

A clip list can be aquired by running clipls in the ~dimr/qubit/QubitControl/ directory.

Further help

If you need aditional help contact Ben Moore(moore@ict.usc.ecu, ext. 1948)

Recording Demos on the QuBit Step by Step

Configuring the Room

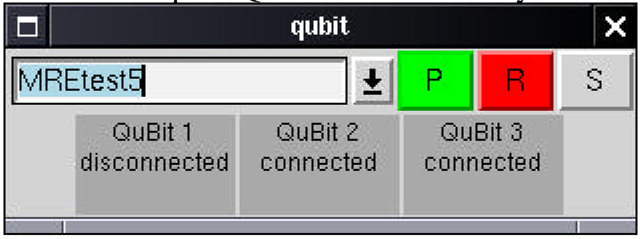
1. Configure the room for running your demo as usual.
2. Route signals from the SGI to the QuBits using the top Extron Matrix 50 switch
   - 2->6
   - 3->7
   - 4->8

Run your software

If you don't know how to run your own software, you're in trouble... :-) Hopefully your software waits until you press some button to actually start running it. Due to the way the QuBits syncronize this process takes ~60 seconds. If your demo doesn't wait for you to start it after the system is running, you will need to time when you start the QuBits in the record process to end at about the same time your demo starts going. The author(Ben Moore) recommends that you make your software have a 2 stage start up process to make your life easier. You will also need to set protools to use its analog input. Perform step 2 from the qubit playback howto's "Configuring Protools" section and select analog instead of digital.

Running QuBitControl

QuBitControl allows you to record and playback clips from the qubit. To run it(at the ICT) login as dimr to kenshin. Go into the qubit/QubitControl directory and run ./qubit. This will bring up the following GUI.



Type the name of the clip you want to create into the text field or select a clip from the list that you want to overwrite (beware... the software default to overwriting clips with the same name. so make sure the name you enter is not on the list.) and press the record button(red 'R'). When you press record each of the monitor at the bottom will change red and then indicate how many seconds until it gets sync(take appox ~15 secs/qubit). When all 3 have sync the timecode signal that drives them is sent 5 seconds after. Once the QuBits are recieving the timecode signal they start recording when the timecode reaches 23:59:10. You can monitor this on the from panel of the QuBit. You should now start your demo.

When your demo is finished running press the stop button(grey 'S'). This will save your clip to the QuBit so it can be played back. If you want to check the clip you recorded you can select your clip from the list and then hit play. You will need to reset the audio and output to take input from the QuBit. See steps 1 and 2 in the Playback Howto: Configuring the Room and Step 2 from Configuring Protools for setup details.

QuBit Extraction and Processing

A quick note about disk space. Processing clips from the qubit can require massive amounts of disk space. The author recommends that you have at least 20GB available for each clip you wish to process. This will handle all the temporary copies, processed frames, audio, and output files. If you do one step at a time and delete everything as you go you might be able to get away with 7GB for a 6 minute clip, but life will be easier if you follow the recommendation above. (spike, currently the only machine that can run these tools, has a drive mounted as /data2 for processing QuBit data.)

1. Record your demo
    - See Recording Demos on the QuBit HOWTO
2. Extract frames using extractvideo.tcl
    - extractvideo.tcl is part of the Extrator tools built for working with the QuBit. They can be found on spike in ~moore/Work/Tcl/qubit/Extractor. A proper release hasn't been made yet since many of the tools are still embryonic WRT stability/functionality.
    - extractvideo.tcl has a known bug
        1. Locking up the QuBit
            - For whatever reason, I haven't figured it out yet, when extracting large numbers of frames from the QuBit using cURL the QuBit's FTP server will lock up. The only way to restart it is to login to the QuBit and reset it. This process takes about 3 minutes. QuVis is working on a fix, but at this time it is still an issue(1/15/03).
    - extractvideo.tcl Usage:
        - expect extractvideo.tcl [-r <storage dir>] [-s <start frame>] <qubit> <clipname>
            - For most applications you will only need to extract video from qubit2.
            - <qubit> refers to the qubit you want to get frames from. You can specify a dns name or IP address. The qubits are named qubit1(left), qubit2(center), and qubit3(right)
            - <clipname> is the name of the clip you want to extract. A list of clips can be aquired using **clipls**. See the QuBit Playback HOWTO
            - An optional stoarge directory can be specified. This is for when the local disk doesn't have enough space to store your clip(avg clip is ~5GB)
            - An optional start frame can be specified as well. This is mainly for dealing with a locked up qubit. Look at the last frame download and then specify that frame as the start frame.
            - frames are extracted into a directory called <qubit>_<clipname>
    - extractvideo.tcl may be better if used from the render farm(which is used later to process the frames). This could potentially save the time required to transfer the frames to the render farm.
3. Extract audio using extractaudio.tcl
    - extractaudio.tcl Usage:
        - expect extractaudio.tcl [-r <storage dir>] <qubit> <clipname>
            - see extractvideo.tcl Usage for details of arguments
            - audio is extracted in wav format into a directory called <qubit>_<clipname>_audio
            - to get all the audio for your demo you will need to extract audio from at least qubit1 and qubit2
                - qubit3 contains the participant recorded when using speech recognition in MRE
    - extractaudio.tcl bugs
        1. try not to interrupt the program. If the program is interrupted the pullaud function on the qubit which is used to extract the wav files doesn't exit and leaves the QuBit in a broken state. It usually requires that the QuBit be rebooted and that any temp wav files on the QuBit be deleted.
            - To delete a temporary wav file
                1. telnet to the qubit that had a problem
                2. cd \Temp
                3. del *.wav
4. Upload frames to the render farm(this step can be skipped if extraction occured on the render farm)

  - Use ftp to upload the files to the moore account on farm001.
    - the password is: <password> [ask Rob Groome]
    - ncftp is recommened as an ftp client since it supports sending whole directories
      - ncftp -u moore farm001
      - put -R <directory of frames from QuBit>

5. Login to the render farm
   - ssh -X moore@farm001
     - passwd: <password>
6. Generate the Submit and Render Scripts
   - run the slz script with the directory where your frames are as an argument
     - if your directory was called MREintroclip2_qubit2 you would:
       - ./slz MREintroclip2_qubit2
   - This will generate 2 scripts:
     - <dirname>_render
     - <dirname>_submit
   - The render script is run on each frame. It uses wine and ReadQubit.exe to convert each frame from the proprietary QVS format into JPG
   - The submit script starts your render job
   - **NOTE:** It may be worth while to make the slz script contain the templates, but for now just be sure to run it from moore's home directory.
7. Run Submit Script
   - ./<submit script name>
   - You can monitor the progress of the render job using irush. To use irush be sure to allow X forwarding when you ssh to farm001(ssh -X moore@farm001). irush manual
8. Download Processed Files from render farm
   - go to the directory where you want to do your final video processing
     - This is usually /data2 on spike
   - using ncftp login as moore to farm001 and download the processed frames
     - ncftp -u moore farm001
     - get -R <directory of frames from QuBit>_jpgs
9. Process the frames into an MPEG-2
   - on spike in /data2 you will find a script called frames2mpg. This will convert a directory of frames into an MPEG-2 movie.
   - to run: ./frames2mpg <directory of frames from QuBit>_jpgs
   - The directory is the finished frames from the render farm. The script should be run from the directory above the frames. It will produce an output file called <directory of frames from QuBit>.m1v. This will be used later to produce the final output movie.
10. Process the audio into an MPEG-1 layer 2 audio file(mp2)
    - on spike in /data2 run the genaudio script. It takes 2 arguments the clipname and the template type.
    - The template type can be either "SEE" or "MRE". MRE includes the Speech Recognizer track from the third QuBit. SEE does not.
    - To generate the audio for a clip called MREfinal8 using the MRE template
      - ./genaudio MREfinal8 MRE
11. Combine(multiplex) the audio and video.
    - ./mplex -o <finished mpg name> <video file> <audio file>
    - Example for MREfinal8
      - ./mplex -o MREfinal8.mpg MREfinal8.m1v MREfinal8.mp2

Setup a Demo Button in WindowMaker

1. Open a terminal
2. start an xterm and specify a class. This class will allow the icon you assign to be locked to the dock button
    - xterm -class <classname>
    - some example class names:
        - MRE
        - SEE
        - PARTHENON
        - RANDOMDEMO
3. A new button should appear at the bottom of the screen on the left, drag this to the dock at the right
4. right click on the button and select Settings
5. now edit the "Application path and arguments" to point at the runclips program with the appropriate clipname specified
    - Ex: /home/dimr/qubit/QubitControl/runclips MREfinal8 for the MRE
    - Ex: /home/dimr/qubit/QubitControl/runclips vistarama2 parthenon for Vistarama and the Parthenon Preview
6. Erase the other command fields
7. Now specify an icon.
    - The icon is a 64x64 image(tiff and pnd are good formats to use) you can see examples on kenshin in /home/dimr/GNUstep/Library/Icons

And that's all there is to say about that.