iSight Programming Guide

Hardware & Drivers > Graphics & Imaging

2004-10-05

Contents

Tables

The iSight Video Camera

This document introduces you to the Apple iSight video camera and describes Apple's enhancements to the product. If you're a developer who needs to understand fully how the iSight camera works, you should also be familiar with the 1394 Trade Association's IIDC 1.30 specification, as well as other standards. These standards are listed in the "Reference and Additional Resources" (page 13) section of this document.

Introduction

The Apple iSight camera is a FireWire video camera with audio input, based on the 1394 Trade Association's IIDC 1.30 specification, with several Apple enhancements.

The iSight camera uses a CCD video sensor to generate a color image as large as 640 x 480 pixels. Digital image processing is performed within the camera before uncompressed video is sent over FireWire. Several frame sizes and frame rates are supported. The sensor is coupled to a motorized focus mechanism, which can be set to an auto-focus mode. An auto-exposure mode is also supported. Focus, exposure, and other video settings can be controlled over FireWire using IIDC mechanisms, as described in this document.

The iSight camera has two microphones arranged front-to-back on the long axis of the top of the enclosure. Audio from these microphones is digitized in the camera and transmitted uncompressed over FireWire to a host computer. Apple software in Mac OS X combines the two signals using beam forming into a single, monaural audio channel.

The iSight camera sends video packets on one FireWire isochronous channel and audio packets on another FireWire isochronous channel. These channels are allocated by a host computer, which then informs the iSight which channels to use. In any single FireWire cycle (1/8000 of a second) the iSight will send one video packet, or one audio packet, or no packet at all––but never both an audio and a video packet in the same cycle. Consequently, the most efficient isochronous allocation on FireWire can be obtained by taking the larger of the audio packet size and the video packet size, rather than by making independent allocations for both streams.

The iSight camera provides a notification mechanism for the iris state. When configured by a host computer, the iSight sends a status packet each time the iris is opened or closed. Closing the iris turns off isochronous audio and video streams. The FireWire interface remains fully active, except that it will not send any isochronous data until the iris is opened and the data flow is re-enabled by a host computer.

Although the audio and video portions of iSight are assembled into a single physical enclosure, they operate as separate logical devices. Each portion can be operated independently of the other.

The following sections describe in detail the video, audio, iris, and factory units of the iSight video camera. Each unit is identified in the Configuration ROM with a specified unit specification ID, as well as a unit software version.

IIDC Video Unit

The video unit is identified in the Configuration ROM as a Unit Directory with Unit Specification ID 0x00A02D and Unit Software Version 0x000102. This unit is based on the IIDC 1.30 specification, with Apple-specific modifications. Developers can determine that the Apple-specific modifications are supported by finding the following two keys in the Root Directory: The Vendor ID key (value 0x000A27 or 0x080007) and the Model ID key (value 0x000008), when both present, uniquely identify the iSight camera.

iSight supports IIDC Video Format 0 (VGA), and (in Format 0) Mode 1, 2, and 3, as reported by the IIDC V_MODE_INQ_0 register.

Apple has added an additional vendor-specific Format (CIF) and Modes 8 through D (hex) for CIF-sized video used by Internet video chat software. The additional Modes have the parameters shown in Table 1-1.

Table 1-1   Apple-added modes with parameters specified

	Packet payload bytes, by frame rate				Image size (pixels)		YUV Format	Bits per Pixel
Mode	3.75 FPS	7.5 FPS	15 FPS	30 FPS	Wide	High		
8	12	24	48	96	128	96	4:1:1	12
9	16	32	64	128	128	96	4:2:2	16
A	33	66	132	264	176	144	4:1:1	12
B	44	88	176	352	176	144	4:2:2	16
C	132	264	528	1056	352	288	4:1:1	12
D	176	352	704	1408	352	288	4:2:2	16

Modes 8 and 9 send 5760 video packets per second, and Modes A through D send 4320 video packets per second. Image data is packed in the same way as in Format 0 Modes 1, 2, and 3. As explained in the"Introduction" (page 7), iSight will never send a video packet in the same isochronous cycle as it sends an audio packet.

Apple has added registers where the CIF formats may be detected, using the same bit conventions as defined by IIDC, as shown in Table 1-2.

Table 1-2   Apple-added registers

Offset	Name	Description
0xF00	CIF_V_MODE_INQ_0	CIF modes supported
0xF10	CIF_V_RATE_INQ_0_8	Valid frame rates for CIF mode 8
0xF14	CIF_V_RATE_INQ_0_9	Valid frame rates for CIF mode 9
0xF18	CIF_V_RATE_INQ_0_A	Valid frame rates for CIF mode A
0xF1C	CIF_V_RATE_INQ_0_B	Valid frame rates for CIF mode B

Offset	Name	Description
0xF20	CIF_V_RATE_INQ_0_C	Valid frame rates for CIF mode C
0xF24	CIF_V_RATE_INQ_0_D	Valid frame rates for CIF mode D
0xF90	CIF_Cur_V_Mode	Set or detect CIF video modes

The CIF Modes supported, as indicated by CIF_V_MODE_INQ_0, have the following bit assignments within CIF_V_MODE_INQ_0 (Table 1-3).

Table 1-3   Bit assignments

InqBitMode_8	0x00000800	128 x 96 4:1:1	12 bits
InqBitMode_9	0x00000400	128 x 96 4:2:2	16 bits
InqBitMode_A	0x00000200	176 x 144 4:1:1	12 bits
InqBitMode_B	0x00000100	176 x 144 4:2:2	16 bits
InqBitMode_C	0x00000080	352 x 288 4:1:1	12 bits
InqBitMode_D	0x00000040	352 x 288 4:2:2	16 bits

If a CIF Mode has been selected, it will be indicated in CIF_Cur_V_Mode as shown in Table 1-4:

Table 1-4   CIF Mode selections indicated in CIF_Cur_V_Mode

Mode_8_128__96_411	0x00000010	128 x 96 4:1:1 12 bits
Mode_9_128__96_422	0x00000020	128 x 96 4:2:2 16 bits
Mode_A_176_144_411	0x00000030	176 x 144 4:1:1 12 bits
Mode_B_176_144_422	0x00000040	176 x 144 4:2:2 16 bits
Mode_C_352_288_411	0x00000050	352 x 288 4:1:1 12 bits
Mode_D_352_288_422	0x00000060	352 x 288 4:2:2 16 bits

To select a CIF video mode, store the desired CIF mode in CIF_Cur_V_Mode and do not change Cur_V_Mode. To select a VGA video mode, store the desired VGA mode in Cur_V_Mode, and do not change CIF_Cur_V_Mode. When a CIF mode has been selected, reading Cur_V_Mode will return 0xE0000000.

Apple has added additional registers for vendor-unique features, as shown in Table 1-5.

Table 1-5   Added Apple registers

Offset	Name	Description
0xD00-0xDFC	Defaults	Read-only copy of IIDC defaults in register offsets 0x800-0x8FC

Offset	Name	Description
0xE00	Default Edge Enhancement	Read-only copy of 0xFBC default
0xFBC	Edge Enhancement Inquiry	Works like IIDC brightness inquiry (0x500)
0xFC0	Edge Enhancement	Works like IIDC brightness register (0x800)

At power up, the iSight sets its exposure settings to known good values. If a driver on the host changes the settings, they may no longer have suitable values. For best results, the driver on the host should not change the settings unless requested by the user. At any time, the driver can revert to known good values by reading the default registers and restoring the settings to the recommended values.

Audio Unit

The audio unit is identified in the Configuration ROM as a Unit Directory with Unit Specification ID 0x000A27 and Unit Software Version 0x000010. Like the video unit, the audio unit is controlled by a register file. The base address of the register file is specified in the Unit Directory of the audio unit. The register file is organized as shown in Table 1-6.

Table 1-6 The audio unit register file

Offset	Name	Description
0x000	AudioEnable	Write 0x80000000 to enable audio transmission (or 0 to disable)
0x204	Default Audio Gain	Read-only
0x210	Gain: Raw Start	Read-only Scale factor for Gain (at 0x500)
0x214	Gain: Raw End	Read-only Scale factor for Gain (at 0x500)
0x218	Gain: Decibels Start	Read-only Signed 32-bit value
0x21C	Gain: Decibels End	Read-only Signed 32-bit value
0x280	Sample Rate Inquiry	Read-only (see 0x400)
0x300	Isochronous TX Config	Isoch Channel + (Speed << 16)
0x400	Sample Rate	Write 0x80000000 to select 48K samples/sec
0x500	Gain	Read/Write values within range given by Raw Start/Raw End registers
0x504	Mute	Write any non-zero value to mute audio without stopping isochronous stream

Audio is sent in large chunks during gaps in the video, according to the format shown in Table 1-7. As explained in the "Introduction" (page 7), an audio packet will never be sent in the same isochronous cycle as a video packet. When 640 x 480 YUV 4:1:1 30 fps video is selected, the audio packets will be about twice as large (and half as numerous) as in any other video mode.

Table 1-7   Format of audio chunks

Byte Offset	Name	Description of Quadlet value
0x000	1394 Packet Header	See 1394 Chapter 6; Length and Channel #
0x004	1394 Header CRC*	CRC for header quadlet only
0x008	Audio Sample Count	Count of audio samples in this packet
0x00C	Audio Signature	ASCII "sght" (0x73676874)
0x010	Audio Sample Total	Total count of all previous samples
0x014	Reserved	Reserved
0x018-0xNNN	Audio Data	Two-channel audio data (see Table 1-8). Up to 1900 bytes (depending on video format)
0xNNN+4	1394 Data CRC*	CRC for payload quadlets
* Most Link hardware (such as OHCI) removes these CRCs as the packet is received		

Each audio sample is a four-byte value containing two signed 16-bit samples. Viewed on 1394, the sample is structured as shown in Table 1-8.

Table 1-8   Audio sample structures

Byte 0 (bits 0-7)	Byte 1 (bits 8-15)	Byte 2 (bits 16-23)	Byte 3 (bits 24-31)
16-bit audio sample (Front/"Left")		16-bit audio sample (Rear/"Right")	

Iris Unit

The Iris unit is identified in the Configuration ROM as a Unit Directory with Unit Specification ID 0x000A27 and Unit Software Version 0x000012. When properly configured, this unit provides asynchronous notification to a host when the iris (shutter) is opened or closed by the user.

The Iris unit provides an Iris Status Address register that specifies the 64-bit FireWire address where iris status will be sent by the iSight. This register is programmed by a host computer using an address value allocated by the host within its own FireWire address space, combined with the host's FireWire bus and node IDs. The bus ID is typically 1023 (0x3ff), and the host memory address value is typically 0x0001.0000.0000 or higher, so the full 64-bit value would (if the host node ID is 2) look something like 0xffc2.0001.0000.0000.

Iris status is sent back to the host immediately after this register is set (to confirm the setting took effect, and to confirm the present iris state), and is then sent again any time the iris state changes.

The FireWire address of the Iris Status Address register within the iSight can be determined using the key 0x40080000, which is found in the Iris Unit Directory. This key indicates an offset address of 0x80000 quadlets (0x200000 bytes) past the base of the CSR range (at 0xFFFF.F000.0000), so the 48-bit address of the Iris Status Address register is 0xFFFF.F020.0000.

The Iris Status Address register is a write-only register and must be written using a single FireWire block write of size 8 bytes.

The Iris Status Address register is invalidated by a bus reset, and the iSight does not track changes in the host node ID that a bus reset might cause. After any bus reset, the host must reprogram this register if further status is desired.

An Iris status message is sent as a 1394 quadlet write packet with a payload value indicating the current iris state. The packet is sent to the address that was set in the Iris Status Address register. A payload value of 1 indicates the iris is open, and 0 indicates the iris is closed.

The following excerpt from Apple's FireBug tool (Table 1-9) shows a host (node FFC2) and an iSight (node FFC1), and the packets that are exchanged when the host programs the Iris Status Address register and then immediately receives a status packet indicating that the iris is open.

Table 1-9 An exchange of packets between host and iSight nodes

`116:4249:0583`	`Bwrite fr ffc2 to ffc1.ffff.f020.0000, sz 8 [actl 8], tLab 5 [ack 2]`
	`0000 ffc20003 00000000 ........`
`116:4249:2530`	`WrResp from ffc1 to ffc2, tLabel 5, rCode 0 [ack 1]`
`116:4250:1048`	`Qwrite from ffc1 to ffc2.0003.0000.0000, value 00000001, tLabel 0 [ack 1]`

The Iris Status Address Register provides an additional service. When a host has stored an address in this register, iSight will accept IIDC and audio register writes only from the host node ID stored in the register. Writes from other nodes will be rejected with `response_conflict_error`. This restriction prevents other nodes from changing the audio and video settings without the host's knowledge. However, other hosts can still receive the audio and video stream, and may still inspect (read) all registers. If, after a bus reset, no host has stored an address in the Iris Status Address Register, then writes to IIDC and audio registers are accepted from any FireWire node.

For good behavior when multiple hosts are on one FireWire bus, a host that has previously gained control of the iSight by writing to the Iris Status Address Register should immediately rewrite this register after any bus reset. Any other host should wait at least one second after any bus reset before attempting to write this register. Any driver attempting to write this register for the first time should also wait one second in addition to any delay required due to a bus reset.

> **Note:** Mac OS X developers can detect the iris state, and any changes in this state, using the "iris state" string property in the AppleIris object in the IO Registry. This object and property are automatically created and updated when the Apple_iSight.kext driver is loaded. Any update will generate a kIOMessagePropertiesChanged message. This property has one of four values, as shown in Table 1-10

Table 1-10 Values and meaning of Iris state properties

Value	Meaning
"pending"	Driver is waiting for iris status from camera

Value	Meaning
"open"	Iris is open
"closed"	Iris is closed
"login failed"	Driver cannot gain control of camera, probably because another node on the FireWire bus has already taken control.

Factory Unit

The Factory unit is identified in the Configuration ROM as a Unit Directory with Unit Specification ID 0x000A27 and Unit Software Version 0x000011. This unit is for Apple use only, and is inactive in typical customer usage scenarios. Developers should not attempt to access this unit.

Mac OS X Support

In Mac OS X, support for receiving video from iSight is provided by a QuickTime video digitizer. An application that uses QuickTime for video input can work with iSight.

Support for receiving audio from iSight is provided by a Core Audio HAL Plug-in. Any Core Audio or Sound Manager-based application can use audio from iSight.

The Apple_iSight kernel extension provides notification of the iris state, as described in "Introduction" (page 7).

The application iChat AV uses all of the above services to provide live audio-video chat on the Internet.

Reference and Additional Resources

If you're a developer who needs to work with the iSight video camera, you should read and understand the following documents. Note that these documents are not available from Apple and the necessary information contained therein is not repeated in this document.

- IEEE 1394-1995: Standard for a High Performance Serial Bus
- IEEE 1394a-2000: Standard for a High Performance Serial Bus––Amendment 1
- IEEE 1212-2001: IEEE Standard for a Control and Status Registers (CSR) Architecture
- IIDC 1.30 Digital Camera specification

The IEEE standards listed above can be obtained from http://www.ieee.org

The IIDC specification listed above is available from http://www.1394ta.org

Tools such as FireBug , mentioned in this document, are available in Apple's FireWire SDK for Mac OS X, which can be downloaded from http://developer.apple.com/sdk

Developers seeking further information about iSight may contact firewire@apple.com