iSeeYou: Disabling the MacBook Webcam Indicator LED

Matthew Brocker
Stephen Checkoway
Johns Hopkins University
Embedded Systems
Embedded Systems
Processors all the way down
Toasters all the way down
MacBook iSight
MacBook iSight

EZ-USB processor
MacBook iSight

EZ-USB processor

LED
Okay. So what?

- Nation state actors
- Stalkers on the Internet
- Anyone who gives you a computer
- Biometric authentication using webcam
Can we turn the LED off while recording?

- Can malicious firmware control the LED?
- Can malicious software (on the host) replace the firmware
Hardware

USB

Image Sensor

LED

EEPROM

MCU

USB
### VGA Products

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Resolution</th>
<th>Pixel Size</th>
<th>Optical Format</th>
<th>Frame Rate</th>
<th>Maximum Data Rate</th>
<th>Power Consumption</th>
<th>Supply Voltage</th>
<th>Output</th>
<th>Shutter</th>
<th>Package</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEW! MT9V012</td>
<td>640H x 480V</td>
<td>3.6µm x 3.6µm</td>
<td>1/6 inch</td>
<td>30 fps (VGA), 60 fps (CIF)</td>
<td>13.5 megapixels per second (27 MHz master clock)</td>
<td>Active: 54mW (30 fps VGA resolution)</td>
<td>2.50V-3.10V (Analog), 1.70V-3.10V (Digital and I/O)</td>
<td>Electronic rolling shutter (ERS)</td>
<td>28-pin iBGA, die or wafer</td>
<td>Mobile, PC Cam</td>
<td></td>
</tr>
<tr>
<td>NEW! MT9V112 (SOC)</td>
<td>640H x 480V</td>
<td>3.6µm x 3.6µm</td>
<td>1/6 inch</td>
<td>30 fps @ 27 MHz</td>
<td>12-13.5 megapixels per second (master clock, 24 MHz-27 MHz)</td>
<td>&lt;78mW (30 fps)</td>
<td>Digital I/O: 1.7V-3.6V, Digital Core: 1.7V-1.9V, 2.5V-3.1V</td>
<td>10-bit, on-chip</td>
<td>Electronic rolling shutter (ERS)</td>
<td>36-pin ICSP, wafer or die</td>
<td>Mobile, PC Cam</td>
</tr>
<tr>
<td>MT9V011</td>
<td>640H x 480V</td>
<td>5.6µm x 5.6µm</td>
<td>1/4 inch (4:3)</td>
<td>30 fps at 27 MHz</td>
<td>13.5 MPS/27 MHz</td>
<td>70mW @ 2.8V, 20pF load, 27 MHz, 30 fps</td>
<td>2.8V</td>
<td>10-bit parallel</td>
<td>Electronic rolling shutter (ERS)</td>
<td>28-pin PLCC</td>
<td>Mobile, PC Cam</td>
</tr>
<tr>
<td>MT9V111 (SOC)</td>
<td>640H x 480V</td>
<td>5.6µm x 5.6µm</td>
<td>1/4 inch (4:3)</td>
<td>15 fps @ 12 MHz (default) programmable up to 30 fps at 27 MHz</td>
<td>12 - 13.5 MPS/27 MHz</td>
<td>&lt;80mW @ 2.8V, 15pF @ 12 MHz</td>
<td>2.8V±0.25V</td>
<td>10-bit, on-chip</td>
<td>Electronic rolling shutter (ERS)</td>
<td>44-Ball ICSP, wafer or die</td>
<td>Mobile, PC Cam, Automotive</td>
</tr>
<tr>
<td>MT9V403</td>
<td>659H x 494V</td>
<td>9.9µm x 9.9µm</td>
<td>1/2 inch</td>
<td>0-200 frames/sec. with source illumination at 550nm</td>
<td>66 MB/s (master clock fps 66 MHz)</td>
<td>130mW@ 200</td>
<td>+3.3V</td>
<td>10-bit digital through a single port</td>
<td>TrueSNAP™ freeze-frame electronic shutter</td>
<td>48-pin CLCC</td>
<td>High Speed</td>
</tr>
</tbody>
</table>

*Product Flyer | Data Sheets | Request Info*
Image Sensor

MT9V112: SOC VGA DIGITAL IMAGE SENSOR
Pixel Data Format

Sensor Core Registers
Table 12: Sensor Registers – Address Page 0

<table>
<thead>
<tr>
<th>Register# Decimal (HEX)</th>
<th>Register Name</th>
<th>Data Format</th>
<th>Default Value Decimal (HEX)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (0x00)</td>
<td>Chip Version</td>
<td>0001 0010 0010 1001 (LSB)</td>
<td>4649 (0x1229)</td>
</tr>
</tbody>
</table>

Write request to camera  To read register 0000  Read from camera 0x12  0x29
Typical Configuration

STANDBY from Controller or Digital GND

Master Clock

Two-Wire Serial Interface

RESET#
iSight architecture

- USB connection to host
- Configure sensor via I²C
- 16 byte I²C EEPROM
- STANDBY = output disable
- PD3 also controls LED:

<table>
<thead>
<tr>
<th>PD3</th>
<th>Output</th>
<th>LED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vcc</td>
<td>Disabled</td>
<td>Off</td>
</tr>
<tr>
<td>Gnd</td>
<td>Enabled</td>
<td>On</td>
</tr>
</tbody>
</table>
# RESET Register

<table>
<thead>
<tr>
<th>Bit Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit 7 Inhibit STANDBY</td>
<td>Setting this bit stops STANDBY from affecting entry to or exit from the low-power state.</td>
</tr>
<tr>
<td>Bit 6 Drive Signals</td>
<td>By default, asserting STANDBY causes the ball interface to enter High-Z. Setting this bit stops STANDBY from contributing to output enable control.</td>
</tr>
</tbody>
</table>
With malicious firmware

- Set RESET register via I²C
- Control LED via PD3 (independent of Standby)
With malicious firmware

- Set RESET register via I²C
- Control LED via PD3 (independent of Standby)
With malicious firmware

- Set RESET register via I²C
- Control LED via PD3 (independent of Standby)
Firmware Load

- Internal Hi-Speed USB bus
- AppleUsbVideoSupport kext
- Non-root process
Firmware Load

- Internal Hi-Speed USB bus
- AppleUsbVideoSupport kext
- Non-root process
Firmware Load

- Internal Hi-Speed USB bus
- AppleUsbVideoSupport kext
- Non-root process
Programming the EZ-USB

- Malware on host programs EZ-USB via USB
- EZ-USB sets RESET reg.
- LED under host control
Programming the EZ-USB

- Malware on host programs
- EZ-USB via USB
- EZ-USB sets RESET reg.
- LED under host control
Programming the EZ-USB

- Malware on host programs EZ-USB via USB
- EZ-USB sets RESET reg.
- LED under host control
Demo
What else can we do?

- Pretend to be a USB HID keyboard
- Break out of a Virtual Machine
Conclusions

- LED has privacy/security implications
- Software-circumventable defenses bad
- Processors in computer are vulnerable
- Security is a systems problem
Conclusions

- LED has privacy/security implications
- Software-circumventable defenses bad
- Processors in computer are vulnerable
- Security is a systems problem

Fin