The Ghost in the Machine

Prof. Hank Dietz

University of Kentucky
Electrical & Computer Engineering
Things You Already Know

- There is a computer inside
- There's an analog sensor to digitize
- There's a lot of processing to be done
- There are lots of I/O devices
What Is The processor?

- Lots of “branding” going on... Sony Bionz, Canon Digic, etc.
- Most cameras use ARM processors, cross compilers for development
- CHDK PowerShots use ARM946ES
  - ARM9E-S Thumb 32-bit SoC
  - ARM5vTE DSP enhancements
  - Other special function units...?
ARM946ES

- SRAM on chip
- More RAM off chip
- Stuff gets copied from flash mem to run
- ARM core(s) augmented by I/O, coprocessors, special fn units...
The Processor? Only One?

- Of course not!
- Multiple cores, coprocessors (e.g., JPEG)
- Interchangeable lenses usually talk via SPI
- SD cards contain processors
  - Usually, just fakes disk drive interface
  - Transcend WiFi SD runs hackable Linux
- Even batteries often contain processors...
An Example: Canon EF Lenses

- 8-bit + 1 stop bit SPI (mode 3)
- Protocol at http://birger.us/downloads.htm
- Pin layout:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vbat</td>
<td>Focusing Motor Power</td>
</tr>
<tr>
<td>A_GND</td>
<td>Motor Ground</td>
</tr>
<tr>
<td>VDD</td>
<td>Lens Digital Power</td>
</tr>
<tr>
<td>Dout</td>
<td>SPI Data Camera Output</td>
</tr>
<tr>
<td>Din</td>
<td>SPI Data Camera Input</td>
</tr>
<tr>
<td>Dclk</td>
<td>SPI Clock and signalisation</td>
</tr>
<tr>
<td>D_GND</td>
<td>Lens Digital Ground</td>
</tr>
<tr>
<td></td>
<td>Connector view from camera side (face)</td>
</tr>
</tbody>
</table>
Other Examples

- Minolta AF / Sony A was first:

- Sony E mount is open, right?
  http://camera-wiki.org/wiki/Sony_E_mount_reverse_engineer
  https://github.com/LexOptical/E-Mount

- Canon’s latest mount, RF, is not only closed, but Canon has sent cease-and-desist letters to all 3rd-party lens makers..
The Sensor Is A Processor

- In many cases, the sensor now contains the ADCs and at least a DMA engine; stacked sensors usually have a layer of DRAM
- The main processor can set parameters
- Sensor data is provided in a “raw” buffer
  - The raw data can be saved (e.g., as DNG)
  - JPEGs are compressed from the raw
  - Live view is sampled/scaled from raw or takes an alternative raw path
Operating System

- ARM provides full protected environment
- Various versions of Linux (e.g., in Sony)
- Proprietary real-time Oses (e.g., in Canon)
  - Canon A4000 is DryOS v2.3
  - VxWorks
- Real-time OSes usually provide a DOS environment – C: AUTOEXEC.BAT
Camera Internal I/O Devices

- Lens extend/retract
- Power zoom
- Image sensor
- Power focus
- Focus assist
- Aperture control
- ND/NightShot filters
- Shutter
- Accelerometers
- Stabilization
- PD & light sensors
- Battery level
- Temp sensors
- Clock, GPS
- Flash (strobe)

...
External I/O Devices

- User interface:
  - Buttons, dials, touch, eye sensing ...
  - LEDs, LCDs, EVF, etc.
- Audio mics & speakers
- Audio/Video outputs (e.g., HDMI)
- Mass storage (internal + SD card)
- NIR/wired remote
- USB, 802.11, NFC, ...
Camera Control

• Is stunningly complicated...
• Lots of the process is from sensor SDK or purchased code (e.g., both Sony DRO and Nikon ADL came from Apical iridix)
• Packages like CHDK & ML do not replace basic camera control, but augment it; you can still call any known functions in the original camera control code
• Webcams often don't allow programming
Internal Camera Control

- Not standardized since cameras diverged from http://photopc.sourceforge.net/protocol.html
- Fake camera UI events:
  + works for everything UI can do
  - version dependent, awkward, nothing new
- API call for each type of operation:
  + extendable, potentially more portable
  - complexity, danger, not reflected in UI
External Camera Control

- Not standardized since cameras diverged from [http://photopc.sourceforge.net/protocol.html](http://photopc.sourceforge.net/protocol.html) but there are standards like PTP...
- Often, image upload only
- FireWire standardizes camera control
- UVC (Universal Video Class) allows a fair degree of video-centric webcam control
- CHDK now pretty much allows everything
Conclusion

• There is a lot of compute power inside, and it can be programmably extendable
• Most cameras lock users out
  • Nothing is standardized/open
  • Security features prevent hacking
• It is very difficult to make major changes, even for the manufacturer
• We use CHDK/ML sites for details