Success Story of the Open-Source Camera Stack: The Nokia N9 Case

Embedded Linux Conference Europe - 2011

Laurent Pinchart
laurent.pinchart@ideasonboard.com
HOW TO WRITE GOOD CODE:

START PROJECT.

DO THINGS RIGHT OR DO THEM FAST?

FAST

CODE FAST

RIGHT

CODE WELL

DOES IT WORK YET?

NO

ALMOST, BUT IT'S BECOME A MASS OF KLUDGES AND SPAGHETTI CODE.

NO, AND THE REQUIREMENTS HAVE CHANGED.

THROW IT ALL OUT AND START OVER.

ARE YOU DONE YET?

NO

GOOD CODE

?
camera

8-megapixel area auto better called
continuous conversations details device dual-led
edit facebook features film focus happens
hard image interest keen key lets loads n9 name
nc nokia phone photo press ready reduction screen seconds sharing
something speed subject taken taking talking
test thanks turns video work zeiss
Media devices increasingly handle multiple related functions. Many USB cameras include microphones, video capture hardware can also output video, or SoC camera interfaces also perform memory-to-memory operations similar to video codecs.

Independent functions, even when implemented in the same hardware, can be modelled as separate devices. A USB camera with a microphone will be presented to userspace applications as V4L2 and ALSA capture devices. The devices’ relationships (when using a webcam, end-users shouldn’t have to manually select the associated USB microphone), while not made available directly to applications by the drivers, can usually be retrieved from sysfs.

With more and more advanced SoC devices being introduced, the current approach will not scale. Device topologies are getting increasingly complex and can’t always be represented by a tree structure. Hardware blocks are shared between different functions, creating dependencies between seemingly unrelated devices.

Kernel abstraction APIs such as V4L2 and ALSA provide means for applications to access hardware parameters. As newer hardware expose an increasingly high number of those parameters, drivers need to guess what applications really require based on limited information, thereby implementing policies that belong to userspace.

The [censored] API aims at solving those problems.
Community QA Process