Supporting Video (de)serializers in Linux: Challenges and Works in Progress

Luca Ceresoli — AIM Sportline
luca@lucaceresoli.net
https://lucaceresoli.net
Embedded Linux Conference Europe 2019
About me

- Embedded Linux engineer at AIM Sportline
  www.aim-sportline.com
  - Develop products on custom hardware
  - Kernel, drivers, bootloader, FPGA
  - Integration, build system
- Open source enthusiast
  - Contributor to the Linux kernel, U-Boot, Buildroot and others
Contents

1. Video serdes chips
2. Linux support
3. Troubles and tribulations
4. Way out
5. Remote I2C
6. Conclusions
Video serdes chips
Serializer/deserializers chipset

SoC
Video (MIPI, LVDS, parallel)
Control (I2C, GPIOs)
Image Sensor

Fast, robust link
Serializer/deserializers chipset

SoC

Deserializer

Serializer

Image Sensor

Video (MIPI, LVDS, parallel)

Control (I2C, GPIOs)

Fast, robust link
Typical application

- Typical use: automotive
  - Autonomous driving (ADAS) cameras
  - Rear camera
  - Infotainment display
- SoC-centric
  - Cameras/displays model well known in advance
  - Cameras/displays always connected
  - High electrical noise
My application

- Action camera
- Base module (SoC, processing, storage)
- Two hot-plug camera modules
  - Interchangeable
  - While recording from the other module
  - Possibly with a different model
Available serdes chips

- Main competitors
  - Texas Instruments  FPD-Link
  - Maxim  GMSL
- Camera or display
- Video bus: MIPI CSI-2, Sub-LVDS, parallel
- Robust link in high electrical noise environment
- 1 to 4 inputs per serializer
- Most have remote I2C, GPIO
- Some have remote UART, audio
Texas Instruments DS90UB954 and DS90UB953

- **DS90UB954** Deserializer
- **DS90UB953** Serializer

**FPD-Link III**
- CSI-2
- CLK
- I2C
- 4x GPIO

**7x GPIO, IRQ**

**DS90UB953**
- Serializer
- **I2C**
- 4x GPIO

**DS90UB953**
- Serializer
- **CSI-2**
- **CLK**
- **I2C**
- 4x GPIO
Linux support
Existing patches

[PATCH v4 0/4] MAX9286 GMSL Support

- By Kieran Bingham, Laurent Pinchart, Jacopo Mondi, Niklas Söderlund
- For Maxim GMSL chips
- See also the ALS 2018 talk slides

[PATCH 0/7] mfd/pinctrl: add initial support of TI DS90Ux9xx ICs

- By Vladimir Zapolskiy
- For TI DS90Ux9xx chips
- See also the ALS 2018 talk slides

[RFC,v2 0/6] TI camera serdes and I2C address translation

- By Luca Ceresoli
- For TI DS90Ux9xx chips
- See also: this talk ☝️
Existing patches

[PATCH v4 0/4] MAX9286 GMSL Support
- By Kieran Bingham, Laurent Pinchart, Jacopo Mondi, Niklas Söderlund
- For Maxim GMSL chips
- See also the ALS 2018 talk slides

[PATCH 0/7] mfd/pinctrl: add initial support of TI DS90Ux9xx ICs
- By Vladimir Zapolskiy
- For TI DS90Ux9xx chips
- See also the ALS 2018 talk slides
Existing patches

[PATCH v4 0/4] MAX9286 GMSL Support
- By Kieran Bingham, Laurent Pinchart, Jacopo Mondi, Niklas Söderlund
- For Maxim GMSL chips
- See also the ALS 2018 talk slides

[PATCH 0/7] mfd/pinctrl: add initial support of TI DS90Ux9xx ICs
- By Vladimir Zapolskiy
- For TI DS90Ux9xx chips
- See also the ALS 2018 talk slides

[RFC v2 0/6] TI camera serdes and I2C address translation
- By Luca Ceresoli
- For TI DS90Ux9xx chips
- See also: this talk :)
Ideal implementation

SoC → DES → Image Sensor

SER
  I2C adapter
  GPIO chip

Image Sensor

ID
Ideal implementation

SoC ➔ DES ➔ SER

- I2C adapter
- GPIO chip

Image Sensor ➔ ID

9
Ideal implementation
Ideal implementation
Ideal implementation
Ideal implementation

- Similar to BBB capes, RPi hats, but hot-plug
- I2C EEPROM on each camera (fixed slave address)
- When a camera is connected
  1. Add serializer, I2C adapter, GPIO chip
  2. Add I2C EEPROM, read model ID
  3. Insert model-specific device tree overlay
  4. DTO adds sensor and other remote devices
- On disconnection, remove overlay
Troubles and tribulations
V4L2 troubles

1. Stream multiplexing: no support in mainline yet
2. Reliability: pipe should work with some nodes disabled
   - A sensor goes faulty
3. Dynamic pipe: remove some nodes, add different ones
   - A sensor is removed, a different model added
Devicetree troubles

1. Runtime DT insertion/removal not in mainline yet
2. Video pipelines: bidirectional endpoints links

```
desr@30 { // base DT
  ports {
    port@0 {
      deser_input: endpoint {
        remote-endpoint = <&ser_output>;
    }
  }
}

ser@3d { // DT overlay
  ports {
    port@1 {
      ser_output: endpoint {
        remote-endpoint = <&desr_input>;
    }
  }
}
```

```
Way out
Main blockers for hotplug applications

- Non-modifiable pipeline
- Runtime Device Tree overlays
Work around main blockers

- Main blockers for hotplug applications
  - Non-modifiable pipeline
  - Runtime Device Tree overlays
- Find a workaround that
  - Is needed only for hotplug applications
  - Does not “infect” serdes drivers
Work around main blockers

- Main blockers for hotplug applications
  - Non-modifiable pipeline
  - Runtime Device Tree overlays
- Find a workaround that
  - Is needed only for hotplug applications
  - Does not “infect” serdes drivers
- Workaround: sensors always instantiated
  - V4L2 is happy
  - Device Tree is static
  - Sensor driver becomes a hack
I2C: ideal solution

SoC → DES → SER → I2C adapter → Image Sensor
I2C: proposed solution

SoC → I2C adapter → DES → SER → Image Sensor
Remote I2C
Remote I2C

- Different between TI and Maxim chips
- Discussed in `linux-i2c`
- BoF during Linux Plumbers Conference 2019
  - https://lucaceresoli.net/plumbers-i2c-bof
  - https://etherpad.openstack.org/p/2019-09-11-I2C-BoF
- Talk by I2C core maintainer Wolfram Sang ("Linux I2C in the 21st Century", yesterday)
- SER + DES are equivalent to an I2C switch
- I2C transactions are replicated based on an alias table
Address Translation (ATR)

```
# i2cdetect -l

i2c-0   i2c   amba:camera-i2c@0
i2c-4   i2c   i2c-0-atr-0
i2c-5   i2c   i2c-0-atr-1

#
```
Address Translation (ATR)

```sh
# i2cdetect -l
i2c-0  i2c    amba:camera-i2c@0 I2C adapter
i2c-4  i2c    i2c-0-atr-0     I2C adapter
i2c-5  i2c    i2c-0-atr-1     I2C adapter
# echo eeprom 0x0a > /sys/bus/i2c/devices/i2c-4/new_device
#```

Address Translation (ATR)

# i2cdetect -l
i2c-0   i2c   amba:camera-i2c@0          I2C adapter
i2c-4   i2c   i2c-0-attr-0             I2C adapter
i2c-5   i2c   i2c-0-attr-1             I2C adapter

# echo eeprom 0x0a > /sys/bus/i2c/devices/i2c-4/new_device
# dmesg | tail -n2
ds90ub954 0-0030: rx0: client 0x0a mapped at alias 0x4b (eeprom)
i2c i2c-4: new_device: Instantiated device eeprom at 0x0a
#
Address Translation (ATR)

# i2cdetect -l
i2c-0  i2c  amba:camera-i2c@0     I2C adapter
i2c-4  i2c  i2c-0-attr-0         I2C adapter
i2c-5  i2c  i2c-0-attr-1         I2C adapter
# echo eeprom 0x0a > /sys/bus/i2c/devices/i2c-4/new_device
# dmesg | tail -n2
ds90ub954 0-0030: rx0: client 0x0a mapped at alias 0x4b (eeprom)
i2c i2c-4: new_device: Instantiated device eeprom at 0x0a
# hexdump /sys/bus/i2c/devices/4-0000/eeprom
00000000 ffff ffff ffff ffff ffff ffff ffff
*
0000100
#

21
Conclusions
Conclusions

- Video serdes are complex
- Video pipeline is an issue
  - V4L2 limitations
  - Additional limitations for hotplug (V4L2, Device Tree overlays)
  - There’s a workaround, implies some compromise
- There is a plan for proper implementation of remote I2C (on TI chips)
Thank you for your attention!

Luca Ceresoli
luca@lucaceresoli.net
https://lucaceresoli.net

© Copyright 2019, Luca Ceresoli
Slides released under
Creative Commons Attribution - Share Alike 3.0 License
https://creativecommons.org/licenses/by-sa/3.0/