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

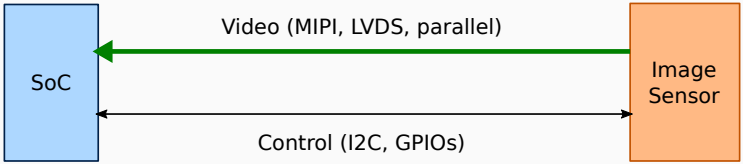
About me

- 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

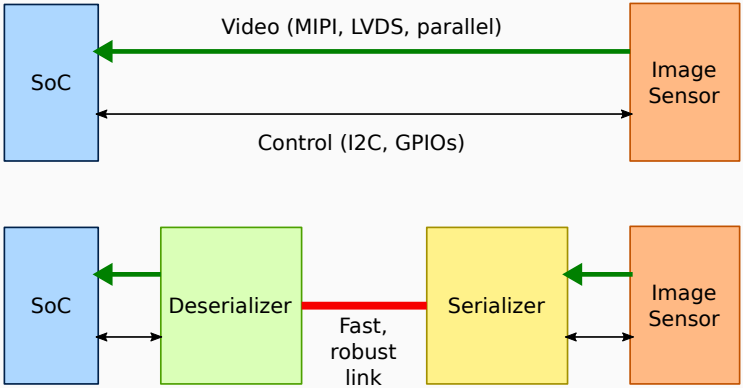
Contents

Video serdes chips

Serializer/deserializers chipset



Serializer/deserializers chipset



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

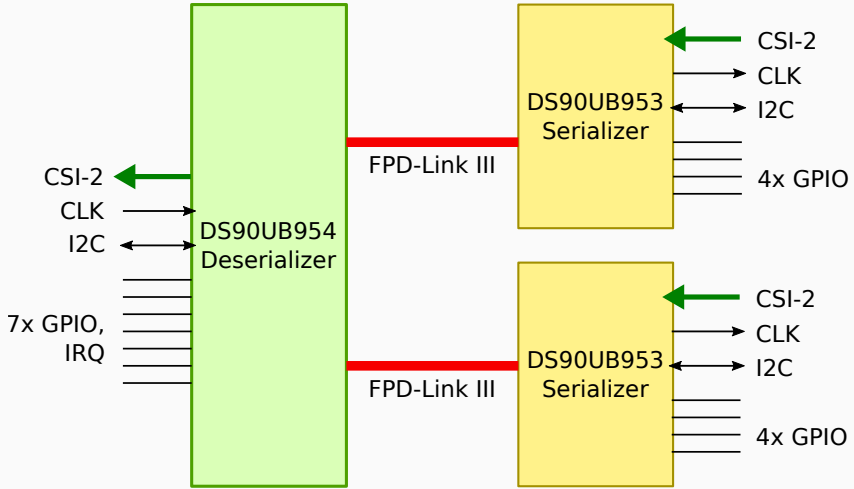
My application

- 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



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

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

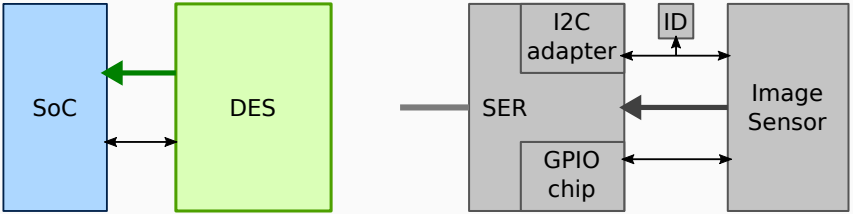
`[PATCH 0/7] mfd/pinctrl: add initial support of TI DS90Ux9xx ICs`

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

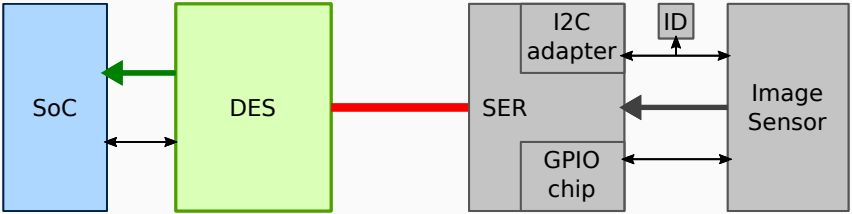
`[RFC,v2 0/6] TI camera serdes and I2C address translation`

- By Luca Ceresoli
- For TI DS90Ux9xx chips
- See also: this talk :)

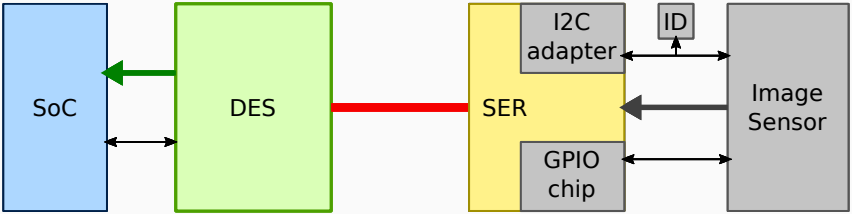
Ideal implementation



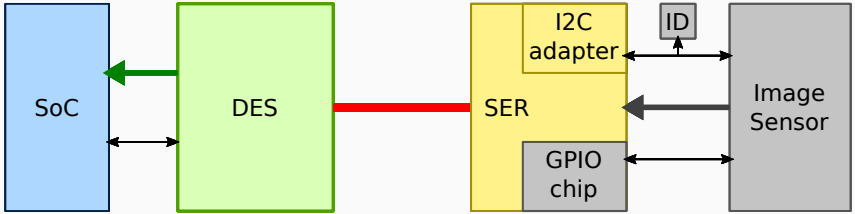
Ideal implementation



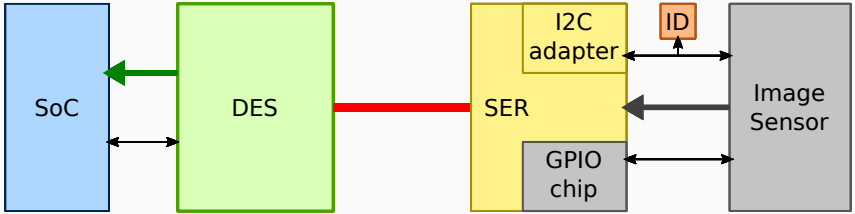
Ideal implementation



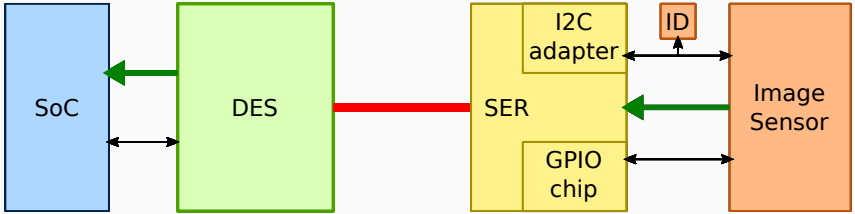
Ideal implementation



Ideal implementation



Ideal implementation



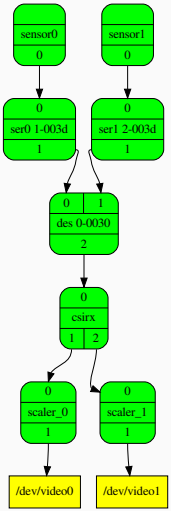
Ideal implementation

- 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

The ideal pipeline →



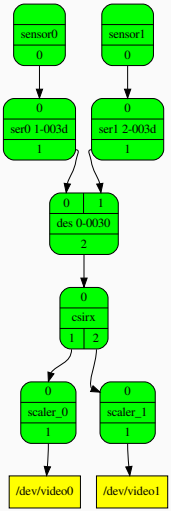
1. Stream multiplexing: no support in mainline yet
2. Reliability: pipe should work with some nodes disabled
   - A sensor goes faulty
3. Dynamic pipe: remove some nodes, add different ones
   - A sensor is removed, a different model added

Devicetree troubles

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

```
deser@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 = <&deser_input>;
```



Way out

Work around main blockers

- Main blockers for hotplug applications
  - Non-modifiable pipeline
  - Runtime Device Tree overlays

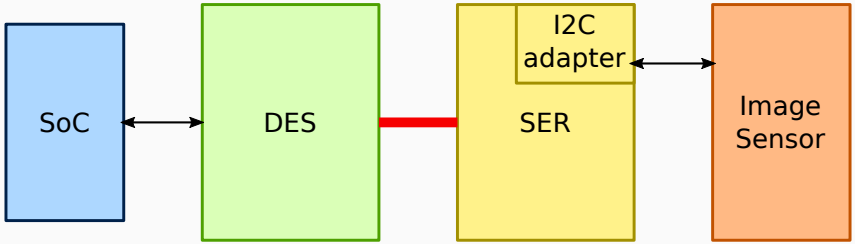
Work around main blockers

- 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

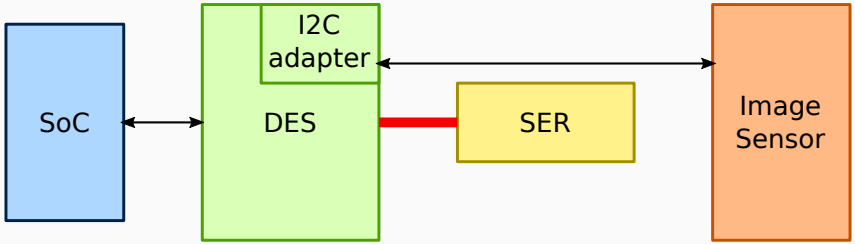
Work around main blockers

- 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

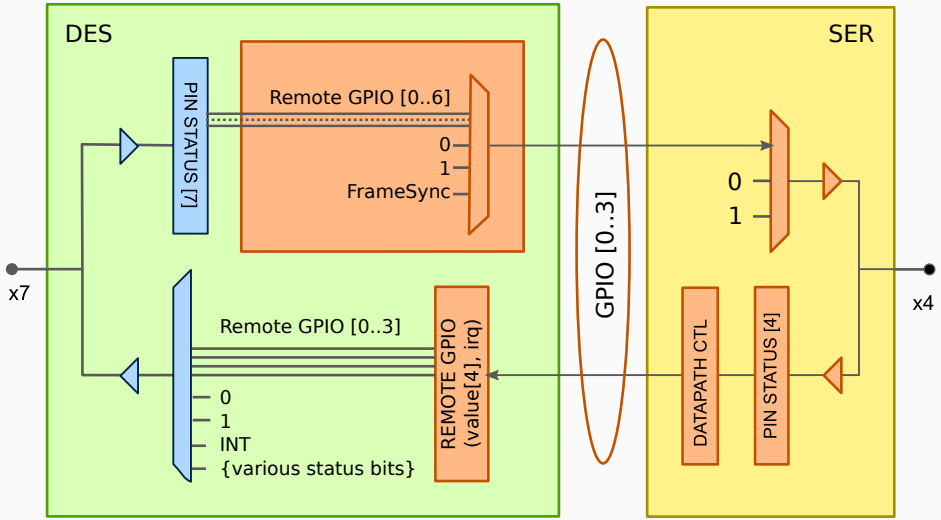
I2C: ideal solution



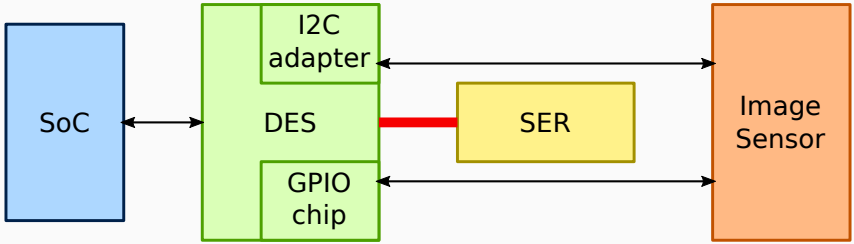
I2C: proposed solution



Remote GPIO


DES
SER
PIN STATUS [7]
Remote GPIO [0..6]
0
1
FrameSync
GPIO [0..3]
0
1
x7
x4
Remote GPIO [0..3]
REMOTE GPIO (value[4], irq)
DATAPATH CTL
PIN STATUS [4]
0
1
INT
{various status bits}

Remote GPIO

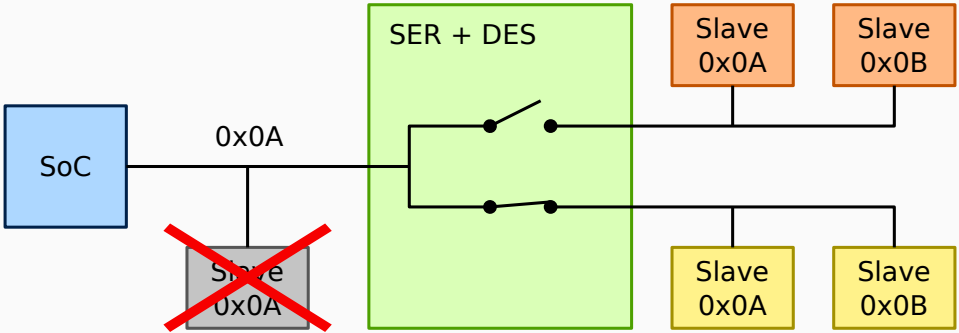


Remote I2C

Remote I2C

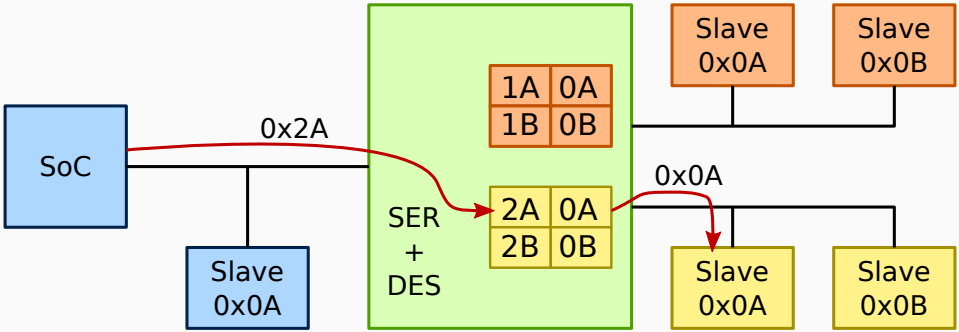
- Different between TI and Maxim chips
- Discussed in `linux-i2c`
- BoF during Linux Plumbers Conference 2019
  - `https://lucaceresoli.net/plumbers-i2c-bof`
  - `https://etherpad.openstack.org/p/2019-09-11-I2C-BoF`
- Talk by I2C core maintainer Wolfram Sang (“Linux I2C in the 21st Century”, yesterday)

Maxim GMSL: I2C switch



- SER+DES are equivalent to an I2C switch

TI FPD-Link III: Address Translation (ATR)



- I2C transactions are replicated based on an alias table

Address Translation (ATR)

```
# 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
#
```

Address Translation (ATR)

```
# 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-atr-0                         I2C adapter
i2c-5    i2c           i2c-0-atr-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-atr-0                          I2C adapter
i2c-5    i2c           i2c-0-atr-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/4-000a/eeprom
0000000 ffff ffff ffff ffff ffff ffff ffff ffff
*
0000100
#
```

Conclusions

Conclusions

- Video serdes are complex
- Issues for proper implementation
  - 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!

Questions?

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