Linux I2C in the 21st century

Wolfram Sang, Renesas / Consultant

29.10.2019, ELCE 2019

Workflow

I2C: its simplicity is a problem

What could go wrong?

- specs didn't change much since 1982
- definately not rocket science[1]
- drivers are (relatively) simple

So, do an IP core, dump the driver, move on…

---

[1]yet, gory details are everywhere

I2C: its simplicity is a problem

What could go wrong?

- specs didn't change much since 1982
- definately not rocket science[1]
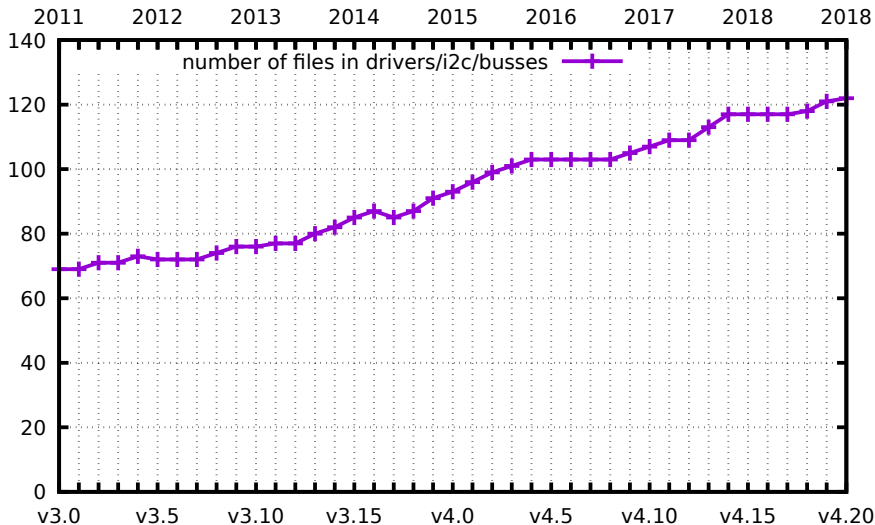- drivers are (relatively) simple

So, do an IP core, dump the driver, move on…

I2C is a good example

- a *Fly-by* subsystem
- mostly used by embedded
- largely maintained in spare time

---

[1]yet, gory details are everywhere

Growth of I2C bus master drivers

Group maintainership

Which group???

Not that I tried...

```
From: Wolfram Sang <wsa@the-dreams.de>
Subject: I2C delay due to maintainer illness

sadly, I have been tied to bed for a few days now, not in a condition to
really work on I2C...

I think it is annoying, sure, but no catastrophe. However, it shows that
I am the single-point-of-failure for I2C patches, what I don't like.
Like I said before, I am open to group maintainership. If you think you
are a reliable candidate, please get in touch with me.
```

```
From: Wolfram Sang <wsa@the-dreams.de>
Subject: I2C delay due to maintainer illness

sadly, I have been tied to bed for a few days now, not in a condition to
really work on I2C...

I think it is annoying, sure, but no catastrophe. However, it shows that
I am the single-point-of-failure for I2C patches, what I don't like.
Like I said before, I am open to group maintainership. If you think you
are a reliable candidate, please get in touch with me.
```

"Get well soon!"

Divide & conquer to the extreme

```
I2C SUBSYSTEM
M:      Wolfram Sang <wsa@the-dreams.de>
L:      linux-i2c@vger.kernel.org
...
S:      Maintained

I2C SUBSYSTEM HOST DRIVERS
L:      linux-i2c@vger.kernel.org
...
S:      Odd Fixes
```

Divide & conquer to the extreme

```
I2C SUBSYSTEM
M:      Wolfram Sang <wsa@the-dreams.de>
L:      linux-i2c@vger.kernel.org
...
S:      Maintained

I2C SUBSYSTEM HOST DRIVERS
L:      linux-i2c@vger.kernel.org
...
S:      Odd Fixes
```

66 drivers have dedicated maintainers now

My companions

```
I2C ACPI SUPPORT
M:      Mika Westerberg <mika.westerberg@linux.intel.com>
S:      Maintained

I2C MUXES
M:      Peter Rosin <peda@axentia.se>
S:      Maintained

I2C/SMBUS CONTROLLER DRIVERS FOR PC
M:      Jean Delvare <jdelvare@suse.com>
S:      Maintained
```
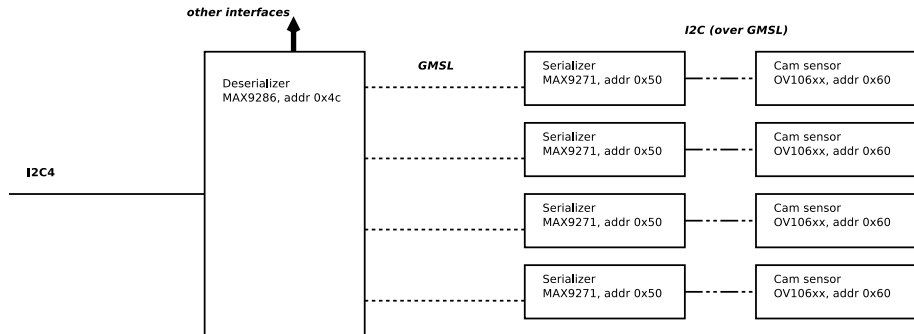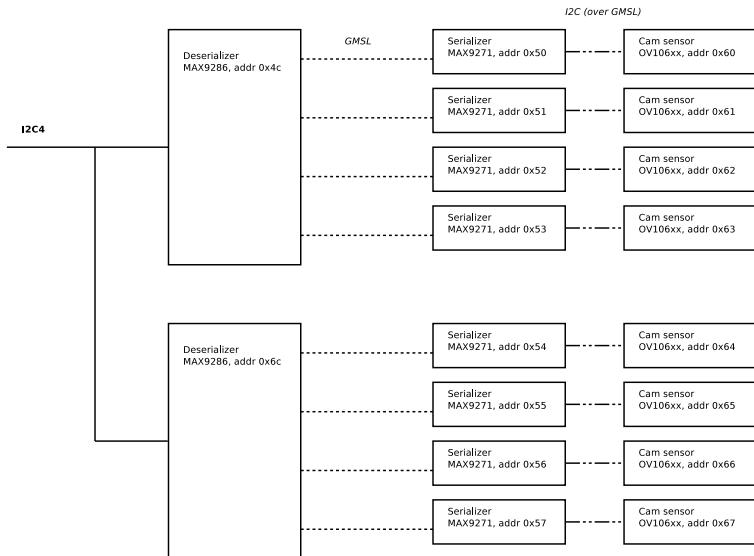
Modern complex setups

I2C4

Deserializer
MAX9286, addr 0x4c

GMSL

I2C (over GMSL)

Serializer MAX9271, addr 0x50	Cam sensor OV106xx, addr 0x60
Serializer MAX9271, addr 0x51	Cam sensor OV106xx, addr 0x61
Serializer MAX9271, addr 0x52	Cam sensor OV106xx, addr 0x62
Serializer MAX9271, addr 0x53	Cam sensor OV106xx, addr 0x63

Deserializer
MAX9286, addr 0x6c

Serializer MAX9271, addr 0x54	Cam sensor OV106xx, addr 0x64
Serializer MAX9271, addr 0x55	Cam sensor OV106xx, addr 0x65
Serializer MAX9271, addr 0x56	Cam sensor OV106xx, addr 0x66
Serializer MAX9271, addr 0x57	Cam sensor OV106xx, addr 0x67

API changes

- NULL
+ ERR_PTR

```
- i2c_new_dummy
+ i2c_new_dummy_device (done)

- i2c_new_secondary_device
+ i2c_new_ancillary_device(done)

- i2c_new_device
+ i2c_new_client_device (WIP)

- i2c_new_probed_device
+ ??? (TBD)
```

API changes: more devm_*

+ devm_i2c_new_client_device (TBD)

+ devm_i2c_new_dummy_device (done)

+ devm_i2c_new_ancillary_device (TBD)

```
- i2c_new_dummy_device
-   (client->adapter, addr)

+ i2c_new_ancillary_device
+   (client, name, default_addr)
```

```
+ int (*master_xfer_atomic)(...)

+ int (*smbus_xfer_atomic)(...)
```
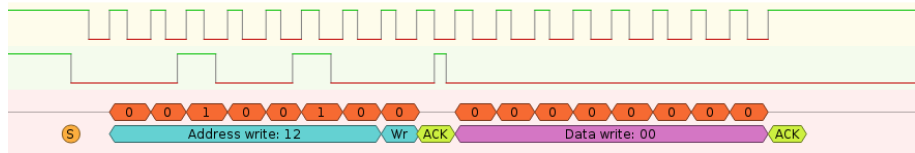
New fault injectors

incomplete_write_byte

Figure 1: Scoped with sigrok
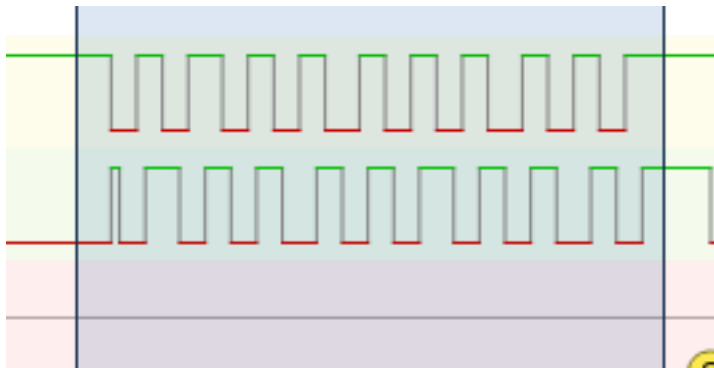
Figure 2: Scoped with sigrok

Figure 3: Scoped with sigrok

inject_panic

from the docs[2]:

"This fault injector will create a Kernel panic once the master under test started a transfer. This usually means that the state machine of the bus master driver will be ungracefully interrupted and the bus may end up in an unusual state. Use this to check if your shutdown/reboot/boot code can handle this scenario."

Remember the atomic transfers?

---

[2]Documentation/i2c/gpio-fault-injection.rst

lose_arbitration

from the docs[3]:

"Here, we want to simulate the condition where the master under test loses the bus arbitration against another master in a multi-master setup."

As simple as:

```
$ echo 200 > lose_arbitration &
$ i2cget -y <bus_to_test> 0x3f
```

---

[3]Documentation/i2c/gpio-fault-injection.rst

Questions? Comments?

Questions?

- Right here, right now...
- At the conference
- wsa@the-dreams.de

References

- Renesas Electronics Logo