C2 port support

Overview

This driver implements the support for Linux of Silicon Labs (Silabs)
C2 Interface used for in-system programming of micro controllers.

By using this driver you can reprogram the in-system flash without EC2
or EC3 debug adapter. This solution is also useful in those systems
where the micro controller is connected via special GPIOs pins.

References

The C2 Interface main references are at (http://www.silabs.com)
Silicon Laboratories site], see:

- AN127: FLASH Programming via the C2 Interface at
http://www.silabs.com/public/documents/tpub_doc/anote/Microcontrollers/Small_Form_Factor/en/an127.pdf, and

- C2 Specification at
http://www.silabs.com/public/documents/tpub_doc/spec/Microcontrollers/en/C2spec.pdf,

however it implements a two wire serial communication protocol (bit
banging) designed to enable in-system programming, debugging, and
boundary-scan testing on low pin-count Silicon Labs devices. Currently
this code supports only flash programming but extensions are easy to
add.

Using the driver

Once the driver is loaded you can use sysfs support to get C2port's
info or read/write in-system flash.

```
# ls /sys/class/c2port/c2port0/
access            flash_block_size  flash_erase       rev_id
dev_id            flash_blocks_num  flash_size        subsystem/
```

```
flash_access      flash_data        reset             uevent
```

Initially the C2port access is disabled since you hardware may have such lines multiplexed with other devices so, to get access to the C2port, you need the command:

```
# echo 1 > /sys/class/c2port/c2port0/access
```

after that you should read the device ID and revision ID of the connected micro controller:

```
# cat /sys/class/c2port/c2port0/dev_id
8
# cat /sys/class/c2port/c2port0/rev_id
1
```

However, for security reasons, the in-system flash access in not enabled yet, to do so you need the command:

```
# echo 1 > /sys/class/c2port/c2port0/flash_access
```

After that you can read the whole flash:

```
# cat /sys/class/c2port/c2port0/flash_data > image
```

erase it:

```
# echo 1 > /sys/class/c2port/c2port0/flash_erase
```

and write it:

```
# cat image > /sys/class/c2port/c2port0/flash_data
```

after writing you have to reset the device to execute the new code:

```
# echo 1 > /sys/class/c2port/c2port0/reset
```