

OsmocomBB - Feature #1461

include some version information / negotiation in the L1CTL protocol

02/19/2016 10:48 PM - laforge

Status:	Feedback	Start date:	
Priority:	High	Due date:	
Assignee:	fixeria	% Done:	60%
Category:	OsmocomBB Firmware		
Target version:			
Resolution:		Spec Reference:	
Description			
The host software should have a way to determine the firmware build/version, as well as the enabled features (TX support or not, burst_ind, ...).			
Related issues:			
Related to SIMtrace 2 - Feature #1460: include some version information / neg...		New	
Blocks OsmocomBB - Feature #3400: mobile: implement GAPK based audio capture ...		Stalled	07/17/2018

History

#1 - 02/21/2016 04:34 PM - laforge

- Assignee deleted (laforge)

#2 - 03/05/2017 07:18 AM - fixeria

- Assignee set to fixeria

- % Done changed from 0 to 30

#3 - 03/05/2017 08:08 AM - fixeria

The following information we can provide to layer23 applications:

```
struct l1ctl_l1_info_ind {
    /* HW info */
    uint8_t hw_dev_id;
    uint8_t hw_dev_ver;
    uint8_t hw_dev_arm_ver;
    uint8_t hw_cdsp_ver;

    /* Supported features */
    uint8_t tx_support;
    uint8_t sim_support;
    uint8_t burst_ind_support;
    uint8_t transceiver_support;
} __attribute__((packed));
```

Any ideas? Maybe something else?

#4 - 03/05/2017 09:36 AM - laforge

On Sun, Mar 05, 2017 at 08:08:08AM +0000, fixeria [REDMINE] wrote:

The following information we can provide to layer23 applications:

please provide some more background on the usage of those fields

```
struct l1ctl_l1_info_ind {
    /* HW info */
    uint8_t hw_dev_id;
    uint8_t hw_dev_ver;
    uint8_t hw_dev_arm_ver;
    uint8_t hw_cdsp_ver;
```

is it well-defined what the above fields are used for? Are we sure that uint8_t is sufficient for those? Are all of the values directly read from hardware? If so, where from?

```
/* Supported features */
uint8_t tx_support;
uint8_t sim_support;
uint8_t burst_ind_support;
uint8_t transceiver_support;
```

I would structure this in a way that it can be extended in the future. One option would be to do something similar to the 'classmark' feature on GSM, where the phone indicates a variable-length bitmask to the network, and new bits can be added at the end. Old software will simply only read the bits it understands and ignore the bits at the end. As we don't care about the size of the structure, we could do it with bytes instead of bits, but that's a matter of taste.

so generally I would counter-propose:

```
struct llctl_ll_info_ind {
    /* version of this structure, if we ever have to change it */
    uint16_t version;
    struct {
        uint8_t dev_id;
        uint8_t dev_ver;
        uint8_t dev_arm_ver;
        uint8_t cdsp_ver;
    } hw;
    struct {
        uint16_t len;
        uint8_t data[0];
    } features;
} __attribute__((packed));
```

so then the code can have a variable-length list of features, and you can add more over time, with forward/backward compatibility

Any ideas? Maybe something else?

I would also include the information about

- build target (e88/e99/...)
- git version of firmware compile
- whether code is running from flash or ram

#5 - 05/03/2017 08:13 AM - fixeria

- Status changed from New to Stalled

#6 - 05/19/2018 11:58 PM - fixeria

- Status changed from Stalled to In Progress

#7 - 06/10/2018 01:47 PM - fixeria

- Status changed from In Progress to Stalled

Switched to USSD related work...

#8 - 07/16/2018 11:19 PM - fixeria

- Related to Feature #1460: include some version information / negotiation in the USB protocol added

#9 - 07/16/2018 11:30 PM - fixeria

As [OsmocomBB](#) is not the only project, for which it would be great to have some info / caps negotiation, I think we can share some helper functions in [libosmocore](#) to avoid code duplication. Having TVL-based API for info negotiation, and bitvec-based (one bit is one feature) API for capabilities negotiation on top of msgb API looks fine for me...

#10 - 07/16/2018 11:30 PM - fixeria

- Blocks Feature #3400: mobile: implement GAPK based audio capture / playback (via ALSA) added

#11 - 07/17/2018 09:09 PM - fixeria

- Category changed from OsmocomBB Layer 1 (Coding) to OsmocomBB Firmware

- Status changed from Stalled to In Progress

#12 - 07/18/2018 02:16 AM - fixeria

- Status changed from In Progress to Feedback

- % Done changed from 30 to 60

The initial version of helper functions has been submitted:

<https://gerrit.osmocom.org/10034/>

I will be happy to get some feedback.

#13 - 09/02/2018 06:43 PM - fixeria

- Target version set to Improvement of the higher layers of OsmocomBB

#14 - 06/21/2019 11:56 AM - ptrkrysik

- Target version deleted (Improvement of the higher layers of OsmocomBB)