

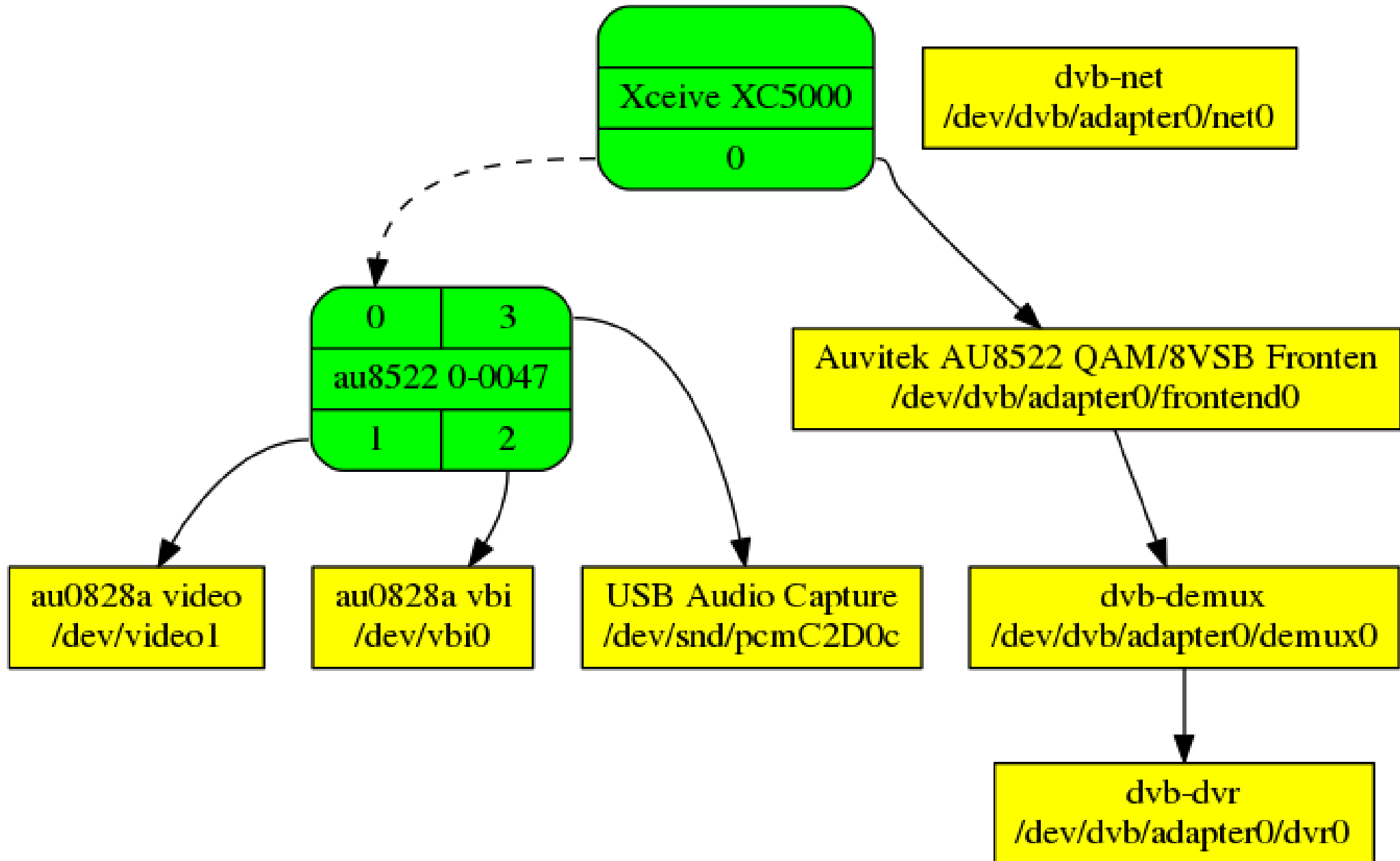
ALSA and au0828 resource sharing using Media
Controller API

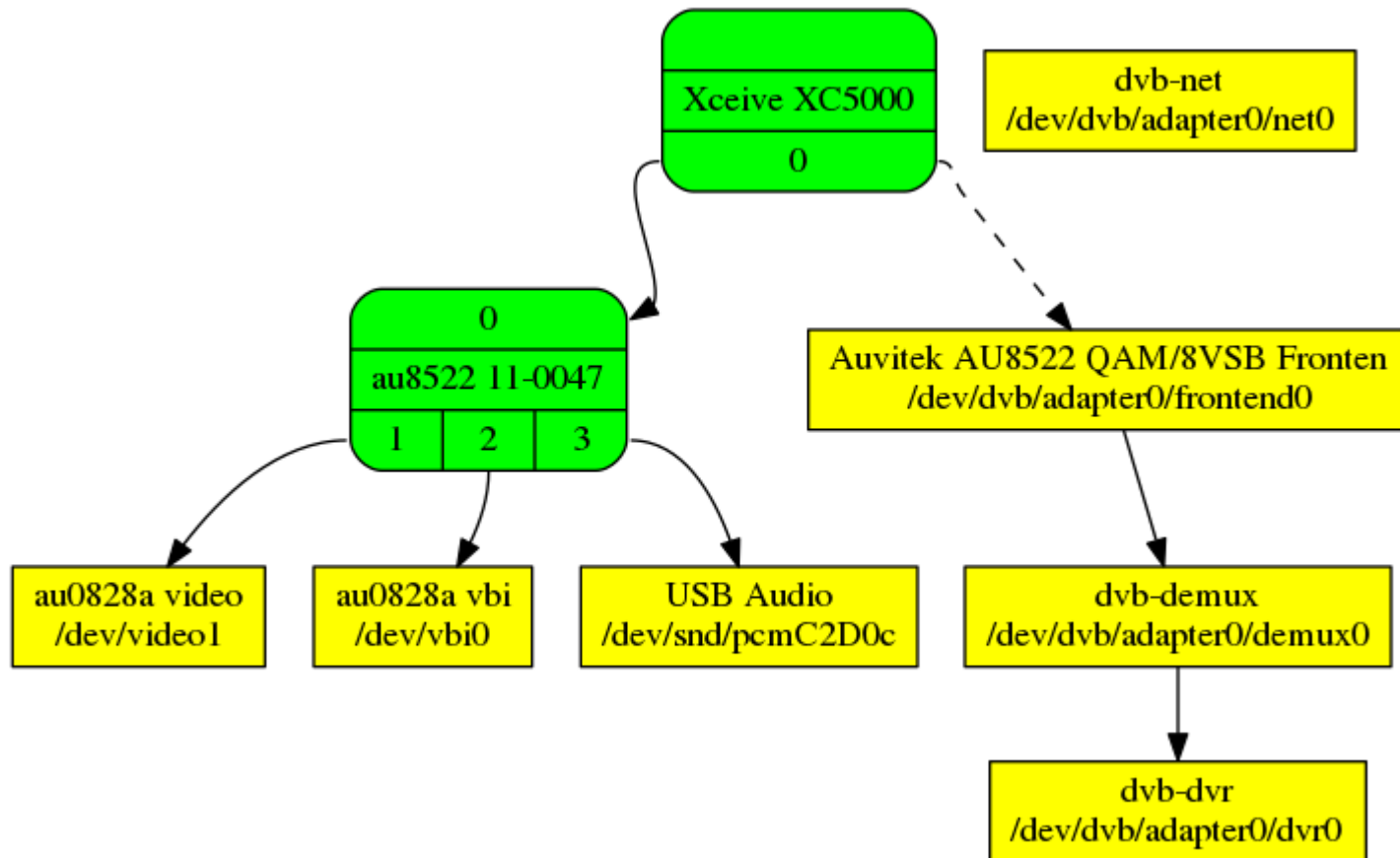
Shuah Khan July 30th 2015

Media Summit, Finland

shuahkh@osg.samsung.com

Comment: au8522 pad 3 should be a Source Pad for ALSA entity





This is after the au8522 changes to fix the problem mentioned in the previous slide.

What are the changes?

- Managed Media Controller API
- Au8522
- V4L2 Core
- DVB Core
- Bridge driver – au0828 (core and video)
- ALSA – snd-usb-audio

Media Controller API Changes

- Managed Media Controller API
 - `media_device_get_devres()`
 - `media_device_find_devres()`
- Register Notify Interface
 - `media_device_register_entity_notify()`
 - `media_device_unregister_entity_notify()`
- New `enable_source` handler in struct `media_device`
- IRQ Safe start/stop pipeline

Media Controller API Changes

- Add new media controller API to allocate media device as a device resource. When a media device is created on the main struct device which is the parent device for the interface device, it will be available to all drivers associated with that interface. For example, if a usb media device driver creates the media device on the main struct device which is common for all the drivers that control the media device, including the non-media ALSA driver, media controller API can be used to share access to the resources on the media device. This new interface provides the above described feature. A second interface that finds and returns the media device is added to allow drivers to find the media device created by any of the drivers associated with the device.

Media Controller API Changes

- Add new interfaces to register and unregister `entity_notify` hook to media device to allow drivers to take appropriate actions when as new entities get added to the shared media device. When a new entity is registered, all registered `entity_notify` hooks are invoked to allow drivers or modules that registered hook to take appropriate action. For example, ALSA driver registers an `entity_notify` hook to parse the list of registered entities to determine if decoder has been linked to ALSA entity. `au0828` bridge driver registers an `entity_notify` hook to create media graph for the device.

Media Controller API Changes

- Add a new field to enable_source handler to find source entity for the sink entity and check if it is available, and activate the link using media_entity_setup_link() interface. Bridge driver is expected to implement and set the handler when media_device is registered or when bridge driver finds the media_device during probe. This is to enable the use-case to find tuner entity connected to the decoder entity and check if it is available, and activate the using media_entity_setup_link() if it is available. This handler can be invoked from media core (v4l-core, dvb-core) as well as other drivers such as ALSA that control the media device.

Media Controller API Changes

- ALSA driver calls Media Controller start/stop pipeline interfaces from IRQ handler. Start/stop pipeline lock `graph_mutex` which is unsafe from a IRQ handler. Convert `graph_mutex` into a spinlock and call it `graph_lock`. IRQ safe start/stop pipeline interfaces will be added based on this change.
- Several drivers that use `graph_mutex` are changed in this patch series.

Au8522 Driver Changes

- Add new pad for ALSA Audio Out to `au8522_media_pads`. Move the `au8522_media_pads` enum to `au8522.h` from `au8522_priv.h`. This will allow `au0828-core` to use these defines instead of hard-coding the pad values when it creates media graph linking decode pads to other entities.

V4L2-core Changes

- Add a new interface to be used by v4l-core to invoke `enable_source` handler in the `media_device` to find tuner entity connected to the decoder and check if it is available. `enable_source` handler will activate the link if tuner is available.

V4L2-core Changes ...

- Change `s_input`, `s_fmt`, `s_tuner`, `s_frequency`, `querystd`, `s_hw_freq_seek`, and `vb2_streamon` interfaces that alter the tuner configuration to check for tuner availability by calling `v4l_enable_media_tuner()`. If tuner isn't free, return `-EBUSY`.

Dvb-core Changes

- Checking for tuner availability from frontend thread start disrupts video stream. Change to check for tuner and start pipeline from frontend open instead and stop pipeline from frontend release.
- DVB-core should use `enable_source` – Fix in the next patch series.

Bridge driver (au0828) changes

- Change au0828 to use Managed Media Controller API to coordinate creating/deleting media device on parent usb device it shares with the snd-usb-audio driver. With this change, au0828 uses `media_device_get_devres()` to allocate a new media device devres or return an existing one, if it finds one.
- Registers `register_notify` handler to create media graph for the device. Creates necessary links between video, vbi, and ALSA entities.
- Implements `enable_source` handler for other drivers to use to check for tuner connected to the decoder and activate the link if tuner is free.

Au0828 video driver changes

- au0828-video will initialize decoder field struct `video_device` for v4l-core to use it when it invokes `enable_source` handler.
- au0828 is changed to use `enable_source` handler to check for tuner availability from `vidioc_g_tuner()`, `au0828_v4l2_close()`, and `queue_setup()` prior to changing tuner settings. If tuner isn't free, return busy condition.

ALSA (snd-usb-audio) driver changes

- Change ALSA driver to use Managed Media Managed Controller API to share tuner with DVB and V4L2 drivers that control AU0828 media device. Media device is created based on a newly added field value in the struct `snd_usb_audio_quirk`. Using this approach, the media controller API usage can be added for a specific device. In this patch, Media Controller API is enabled for AU0828 hw. `snd_usb_create_quirk()` will check this new field, if set will create a media device using `media_device_get_devres()` interface.

ALSA (snd-usb-audio) driver changes

- Change ALSA driver to use Managed Media Managed Controller API to share tuner with DVB and V4L2 drivers that control AU0828 media device. Media device is created based on a newly added field value in the struct `snd_usb_audio_quirk`. Using this approach, the media controller API usage can be added for a specific device. In this patch, Media Controller API is enabled for AU0828 hw. `snd_usb_create_quirk()` will check this new field, if set will create a media device using `media_device_get_devres()` interface.

ALSA (snd-usb-audio) driver changes

- `media_device_get_devres()` will allocate a new media device devres or return an existing one, if it finds one.
- During probe, media usb driver could have created the Managed media device. It will then register the media device if it isn't already registered. Media device unregister is done from `usb_audio_disconnect()`.

ALSA (snd-usb-audio) driver changes

- Media device is created based on a newly added field value in the struct `snd_usb_audio_quirk`. Using this approach, the media controller API usage can be added for a specific device. In this patch, Media Controller API is enabled for AU0828 hw. `snd_usb_create_quirk()` will check this new field, if set will create a media device using `media_device_get_devres()` interface. device if it isn't already registered. Media device unregister is done from `usb_audio_disconnect()`.

ALSA (snd-usb-audio) driver changes

- New structure `media_ctl` is added to group the new fields to support media entity and links
This new structure is added to struct `snd_usb_substream`.
- A new `entity_notify` hook and a new ALSA capture media entity are registered from `snd_usb_pcm_open()` after setting up hardware information for the PCM device.

ALSA (snd-usb-audio) driver changes

- When a new entity is registered, Media Controller API interface `media_device_register_entity()` invokes all registered `entity_notify` hooks for the media device. ALSA `entity_notify` hook parses all the entity list to find a link from decoder to ALSA entity. This indicates that the bridge driver created a link from decoder to ALSA capture entity.

ALSA (snd-usb-audio) driver changes

- ALSA will attempt to enable the tuner to link the tuner to the decoder calling `enable_source` handler if one is provided by the bridge driver prior to starting Media pipeline from `snd_usb_substream_capture_trigger()`. If `enable_source` returns with tuner busy condition, then `SNDRV_PCM_TRIGGER_START` will fail with `-EBUSY`. Media pipeline is stopped in response to `SNDRV_PCM_TRIGGER_STOP`.

References

- Please find the patch series at:
 - https://git.kernel.org/cgit/linux/kernel/git/shuah/linux.git/log/?h=media_controller