FDExt boards V4L2 API

Available Controls

These controls are accessible via the standard V4L2 API and can be changed by standard v4l2-ctl and qv4l2. The header file `/include/forward-v4l2-ioctl.h` contains all the necessary declarations to use the controls in your code. Most controls are accessible for both capture inputs and playback outputs. Also most of them can be changed during capture/playback (without stopping). Some controls are common for the entire board - if you change them in one device then all devices for the board will have the same value for the control.

- `V4L2_CID_FORWARD_ENABLE_VANC (bool)` – enable/disable VANC data during capture/playback. **Don't work with active capture/playback**.
- `V4L2_CID_FORWARD_WIDESCREEN (bool)` – for capture the control return flag WIDESCREEN from input VPID / for playback the control set flag WIDESCREEN.
- `V4L2_CID_FORWARD_HW_TIMESTAMP (u32)` – return current timestamp of internal hardware clock. The clock has frequency148.5MHz. All outputs in sync with this clock. **Read only**.
- `V4L2_CID_FORWARD_FREQUENCY_ADJUST (u64)` - output frequency adjustment in ppb (point-per-trillion, 10^-12). By writing values to this control, you can smoothly (without losing the signal at the output) change the carrier frequency at the output - `Fout = V4L2_CID_FORWARD_FREQUENCY_ADJUST/1e-12 * Fbase`. It is not recommended to change the value by more than 10ppm (1e-6) at a time - this can lead to signal loss at the inputs and outputs for several seconds. The adjustment limit is ±500ppm. **For the entire board**.
- `V4L2_CID_FORWARD_ASI_PERIOD (enum)` – set frame duration for ASI capture/playback. **Don't work with active capture/playback**.
  - 20ms (0) – *by default*
  - 10ms (1)
  - 5ms (2)
  - 2.5ms (3)
- `V4L2_CID_FORWARD_TIMECODE (enum)` – select timecode type for capture/playback.
  - None (0) – don't capture/playback timecode – *by default*
  - LTC (1) – capture LTC only / playback LTC
  - VITC (2) – capture VITC only / playback VITC
  - Any (3) – capture any timecode / *not valid for playback*
- `V4L2_CID_FORWARD_GENLOCK_SOURCE (enum)` – select source for hardware clock synchronization, carrier frequency lock/sync. **For the entire board**.
  - `V4L_FORWARD_GENLOCK_SRC_MASTER (0)` – hardware time use internal generator, the carrier frequency `Fbase` does not depend on the inputs. It can be changed using by `V4L2_CID_FORWARD_FREQUENCY_ADJUST`, *by default*

    - `V4L_FORWARD_GENLOCK_SRC_ANALOG (1)` - sync carrier frequency to analog input (black-burst/tri-level-sync)
    - `V4L_FORWARD_GENLOCK_SRC_IN0 (2)` - sync carrier frequency to first SDI input
    - `V4L_FORWARD_GENLOCK_SRC_IN1 (3)` - sync carrier frequency to second SDI input
- `V4L2_CID_FORWARD_GENLOCK_ENABLE (bool)` – sync output phase to selected sync source. If this control is FALSE then output is in sync with gunlock source (output rate is the same as sync source rate) but phase is undefined (any offset may be). **Only for playback/output**.
- `V4L2_CID_FORWARD_GENLOCK_STATE (enum)` – return current state of gunlock machine. **Read only**. **For the entire board**.
    - `V4L_FORWARD_GENLOCK_MASTER (0)` – use internal generator *(all right)*
    - `V4L_FORWARD_GENLOCK_NO_INPUT_SIGNAL (1)` – no signal on selected sync source *(error)*
    - `V4L_FORWARD_GENLOCK_LOCKING (2)` - selected sync source has good signal but sync machine is not locked yet *(not good)*
    - `V4L_FORWARD_GENLOCK_LOCKED (3)` - sync machine is locked to selected sync source *(all right)*
    - `V4L_FORWARD_GENLOCK_HOLDOVER (3)` - no signal on selected sync source now but sync machine use last locked sync rate *(not good)*
- `V4L2_CID_FORWARD_GENLOCK_OFFSET (u32)`- shift output phase relative to sync source phase. One step - 6.734ns = 1/148.5MHz = 1 pixel for 1080p50/p60. **Only for playback/output**.
- `V4L2_CID_FORWARD_ANALOG_RX_MODE (enum)`- analog/genlock input operating mode. **For the entire board**.
    - Genlock (0) – input is analog video signal – *by default*
    - PPS (1) – input is PPS (1Hz)
    - LTC (2) – input is LTC (audio signal)
- `V4L2_CID_FORWARD_ANALOG_RX_TIMESTAMP (struct)` – return current state of analock/gunlock input as `struct v4l2_forward_analog_rx_timestamp` with flag no signal on input, timestamp of last input event/frame, current timestamp of hardware clock and system clock (std::steady_clock). **For the entire board**.

Available Events

Driver send events by standard V4L2 API - https://linuxtv.org/downloads/v4l-dvb-apis/userspace-api/v4l/dev-event.html The header file `/include/forward-v4l2-ioctl.h` contains all the necessary declarations to use the controls in your code. Driver send follow events:

- `V4L2_EVENT_SOURCE_CHANGE` - standard format change event.
- `V4L2_EVENT_VSYNC` - standard sync event. Driver send the event once per field all time (with or without active playback/capture).

- `V4L2_EVENT_FRAME_SYNC` - standard sync event. Driver send the event once per frame before send frame buffer to the application (is playback is enabled). Contains the current frame number.
- `V4L2_EVENT_FORWARD_TIMESTAMP` - driver-specific event. It contains the current frame timestamp. Driver send the event once per field all time (with or without active playback/capture). It include `struct v4l2_event_forward_timestamp` with follow fields:
  - `buffer_sequence` – current frame number (same as buffer and `V4L2_EVENT_FRAME_SYNC)`
  - `field` – current field identifier (`V4L2_FIELD_*)`
  - `eof_timestamp` – hardware clock timestamp for the frame/field end. Format as for `V4L2_CID_FORWARD_HW_TIMESTAMP`
  - `irq_timestamp` - hardware clock timestamp for driver interrupt processing. Format as for `V4L2_CID_FORWARD_HW_TIMESTAMP`