

# LED Transient Trigger

The leds timer trigger does not currently have an interface to activate a one shot timer. The current support allows for setting two timers, one for specifying how long a state to be on, and the second for how long the state to be off. The delay\_on value specifies the time period an LED should stay in on state, followed by a delay\_off value that specifies how long the LED should stay in off state. The on and off cycle repeats until the trigger gets deactivated. There is no provision for one time activation to implement features that require an on or off state to be held just once and then stay in the original state forever.

Without one shot timer interface, user space can still use timer trigger to set a timer to hold a state, however when user space application crashes or goes away without deactivating the timer, the hardware will be left in that state permanently.

Transient trigger addresses the need for one shot timer activation. The transient trigger can be enabled and disabled just like the other leds triggers.

When an led class device driver registers itself, it can specify all leds triggers it supports and a default trigger. During registration, activation routine for the default trigger gets called. During registration of an led class device, the LED state does not change.

When the driver unregisters, deactivation routine for the currently active trigger will be called, and LED state is changed to LED\_OFF.

Driver suspend changes the LED state to LED\_OFF and resume doesn't change the state. Please note that there is no explicit interaction between the suspend and resume actions and the currently enabled trigger. LED state changes are suspended while the driver is in suspend state. Any timers that are active at the time driver gets suspended, continue to run, without being able to actually change the LED state. Once driver is resumed, triggers start functioning again.

LED state changes are controlled using brightness which is a common led class device property. When brightness is set to 0 from user space via echo 0 > brightness, it will result in deactivating the current trigger.

Transient trigger uses standard register and unregister interfaces. During trigger registration, for each led class device that specifies this trigger as its default trigger, trigger activation routine will get called. During registration, the LED state does not change, unless there is another trigger active, in which case LED state changes to LED\_OFF.

During trigger unregistration, LED state gets changed to LED\_OFF.

Transient trigger activation routine doesn't change the LED state. It creates its properties and does its initialization. Transient trigger deactivation routine, will cancel any timer that is active before it cleans up and removes the properties it created. It will restore the LED state to non-transient state. When driver gets suspended, irrespective of the transient state, the LED state changes to LED\_OFF.

Transient trigger can be enabled and disabled from user space on led class devices, that support this trigger as shown below:

```
echo transient > trigger
echo none > trigger
```

NOTE:

Add a new property trigger state to control the state.

This trigger exports three properties, activate, state, and duration. When transient trigger is activated these properties are set to default values.

- duration allows setting timer value in msecs. The initial value is 0.
- activate allows activating and deactivating the timer specified by duration as needed. The initial and default value is 0. This will allow duration to be set after trigger activation.
- state allows user to specify a transient state to be held for the specified duration.

activate

- one shot timer activate mechanism. 1 when activated, 0 when deactivated. default value is zero when transient trigger is enabled, to allow duration to be set.

activate state indicates a timer with a value of specified duration running. deactivated state indicates that there is no active timer running.

duration

- one shot timer value. When activate is set, duration value is used to start a timer that runs once. This value doesn't get changed by the trigger unless user does a set via echo new\_value > duration

state

- transient state to be held. It has two values 0 or 1. 0 maps to LED\_OFF and 1 maps to LED\_FULL. The specified state is held for the duration of the one shot timer and then the state gets changed to the non-transient state which is the inverse of transient state. If state = LED\_FULL, when the timer runs out the state will go back to LED\_OFF. If state = LED\_OFF, when the timer runs out the state will go

back to LED\_FULL. Please note that current LED state is not checked prior to changing the state to the specified state. Driver could map these values to inverted depending on the default states it defines for the LED in its brightness\_set() interface which is called from the led brightness\_set() interfaces to control the LED state.

When timer expires activate goes back to deactivated state, duration is left at the set value to be used when activate is set at a future time. This will allow user app to set the time once and activate it to run it once for the specified value as needed. When timer expires, state is restored to the non-transient state which is the inverse of the transient state:

echo 1 > activate	starts timer = duration when duration is not 0.
echo 0 > activate	cancels currently running timer.
echo n > duration	stores timer value to be used upon next activate. Currently active timer if any, continues to run for the specified time.
echo 0 > duration	stores timer value to be used upon next activate. Currently active timer if any, continues to run for the specified time.
echo 1 > state	stores desired transient state LED_FULL to be held for the specified duration.
echo 0 > state	stores desired transient state LED_OFF to be held for the specified duration.

## What is not supported

- Timer activation is one shot and extending and/or shortening the timer is not supported.

## Examples

use-case 1:

```
echo transient > trigger
echo n > duration
echo 1 > state
```

repeat the following step as needed:

```
echo 1 > activate - start timer = duration to run once
echo 1 > activate - start timer = duration to run once
echo none > trigger
```

This trigger is intended to be used for the following example use cases:

- Use of LED by user space app as activity indicator.
- Use of LED by user space app as a kind of watchdog indicator -- as long as the app is alive, it can keep the LED illuminated, if it dies the LED will be extinguished automatically.
- Use by any user space app that needs a transient GPIO output.