STM32MP15 Tamper configuration

Contents

Stable: 28.06.2021 - 07:23 / Revision: 28.06.2021 - 07:22

Stable: 28.06.2021 - 01.2.0 / Revision: 28.06.2021 - 01.22

A quality version of this page, approved on *28 June 2021*, was based off this revision.

Contents

1 Overview

The STM32 MPUs embed tamper detection management. Tamper management and configuration have been added to the secure OS to select and detect events.

STM32MP15 microprocessors offer 5 internal tampers and 3 external tampers.

The internal tampers are the following:

- RTC voltage domain monitoring
- LSE monitoring
- HSE monitoring
- RTC calendar overflow
- Monotonic counter overflow

External tampers can be configured to be passive or active.

On a tamper event detection, the backup registers are cleared and the Backup SRAM is read-protected and cannot be accessed until the next reset.

The Automatic erase mode can be configured for external tampers. It is enabled by default but can be turned off if the user application needs to control erase operations.

2 Software configuration

The tamper driver only exists in the Trusted Firmware-A. It is not yet available in OP-TEE

Internal and external tampers have to be configured in:

- TAMP device tree configuration
- Secure OS main security configuration file .

This second part is statically defined and must be customized depending on the application needs. The file contains two static tables, one for internal tampers, another for external ones.

2.1 Internal tampers

Below the structure that registers internal tamper:

```
struct stm32_tamp_int {
        int id;
        void (*func)(int id);
};
```

Internal tamper structure contains:

- an ID, linked to the existing SoC internal tamper
- a function that is called when a tamper event is detected. This function can be customized. By default, it just prints the tamper ID and resets the SoC.

A static list of tampers is automatically registered during the main security loop.

```
static struct stm32_tamp_int int_tamp[PLAT_MAX_TAMP_INT] = {
        {
                .id = ITAMP1,
                .func = stm32mp1_tamper_action,
        },
...
}
```

By default, only internal tampers 1, 2, 3 and 4 are enabled.

2.2 External tampers

When an external tamper is enabled, the associated PIN is compulsorily attached to the tamper (whatever the previous GPIO config). There is also some priority on the settings inside RTC/TAMP. See Reference Manual RTC section "GPIOs controlled by the RTC and TAMP".

Below the structure that registers external tampers:

```
struct stm32_tamp_ext {
        int id;
        uint8_t mode;
        uint8_t erase;
        uint8_t evt_mask;
        void (*func)(int id);
};
```

External tamper structure contains:

- an ID, linked to the SoC external tamper
- a mode that depends on the filter count parameter (described below in this article):
  - TAMP_TRIG_OFF: Low trigger for passive tamper or input rising edge
  - TAMP_TRIG_ON: High trigger for passive tamper or input falling edge
  - TAMP_ACTIVE: Active tamper selected
- an erase mode: for backup registers and Backup SRAM
  - TAMP_NOERASE: no automatic erase
  - TAMP_ERASE: automatic erase
- an event mask:
  - TAMP_NO_EVT_MASK: The tamper event must be cleared by software
  - TAMP_EVT_MASK: When the event is detected, the tamper is masked and internally cleared. No erase is performed.
- a function pointer: function that is called when the tamper is detected

Below a configuration example to enable two external tampers:

```
static struct stm32_tamp_ext ext_tamp[PLAT_MAX_TAMP_EXT] = {
{
                .id = EXT_TAMP1,                        // External tamper 1
                .mode = TAMP_TRIG_ON,                   // Tamper trigger event
                .erase = TAMP_NOERASE,                  // Backup registers are not erased
                .evt_mask = TAMP_NO_EVT_MASK,           // Mask is not set
                .func = NULL,                           // No function
},
{
                .id = EXT_TAMP2,                        // External tamper 2
                .mode = TAMP_ACTIVE,                    // Active tamper selected
                .erase = TAMP_NOERASE,                  // Backup registers and backup SRAM
are not erased
                .evt_mask = TAMP_NO_EVT_MASK,           // Mask is not set
                .func = NULL,                           // No function
},
        TAMP_UNUSED,
}
```

External tampers require to configure the filtering mode (for passive tamper) or active mode selection. The configuration is performed in the same secure OS main security configuration file inside the init_sec_peripherals function.

Filter_conf and active_conf values must be configured according to the values defined in stm32_tamp.h .

Below an example of filtering configuration for passive tamper with pull-up disabled, 2 RTC cycles for precharge, 1 count event for detection (level detection), and a sampling frequency set to 16,384 Hz:

```
        uint32_t filter_conf = TAMP_FLTCR(TAMP_FILTER_PULL_UP_DISABLE,
                                          TAMP_FILTER_DURATION_2_CYCLES,
                                          TAMP_FILTER_COUNT_1,
                                          TAMP_FILTER_SAMPLING_16384);
```

Below an example of active tamper filtering configuration.
Based on the external tamper declaration above, tamper 2 is the only one defined as active.
The configuration (for all active tampers) is the following:

- global active filtering enabled
- tamper compared to defined tamp_out selection
- output charge period set to 2 x CK_ATPRE * prescaler clock period, with clock configured to RTCCKL/2
- tamp_out selection 2 set to external tamper 1 output

```
        uint32_t active_conf = TAMP_ACT(TAMP_ACTIVE_FILTER_ON,
                                        TAMP_ACTIVE_ATO_TAMPOUTSEL,
                                        TAMP_ACTIVE_APER_1_OUTPUT,
                                        TAMP_ACTIVE_CKSEL_DIV_2,
                                        TAMP_ACTIVE_ATOSEL_OUT2_(0));
```

Operating System

Real Time Clock

Low Speed External oscillator (STM32 clock source)

High Speed External oscillator (STM32 clock source)

Open Portable Trusted Execution Environment

Tamper

General-Purpose Input/Output (A realization of open ended transmission between devices on an embedded level. These pins available on a processor can be programmed to be used to either accept input or provide output to external devices depending on user desires and applications requirements.)

uniprocessor