Hardware spinlock overview

Stable: 11.02.2019 - 12:21 / Revision: 15.01.2019 - 14:27

Summary

This article gives information about the Linux® hardware spinlock framework.

It explains how to activate the hardware spinlock framework and, based on examples, how to use it.

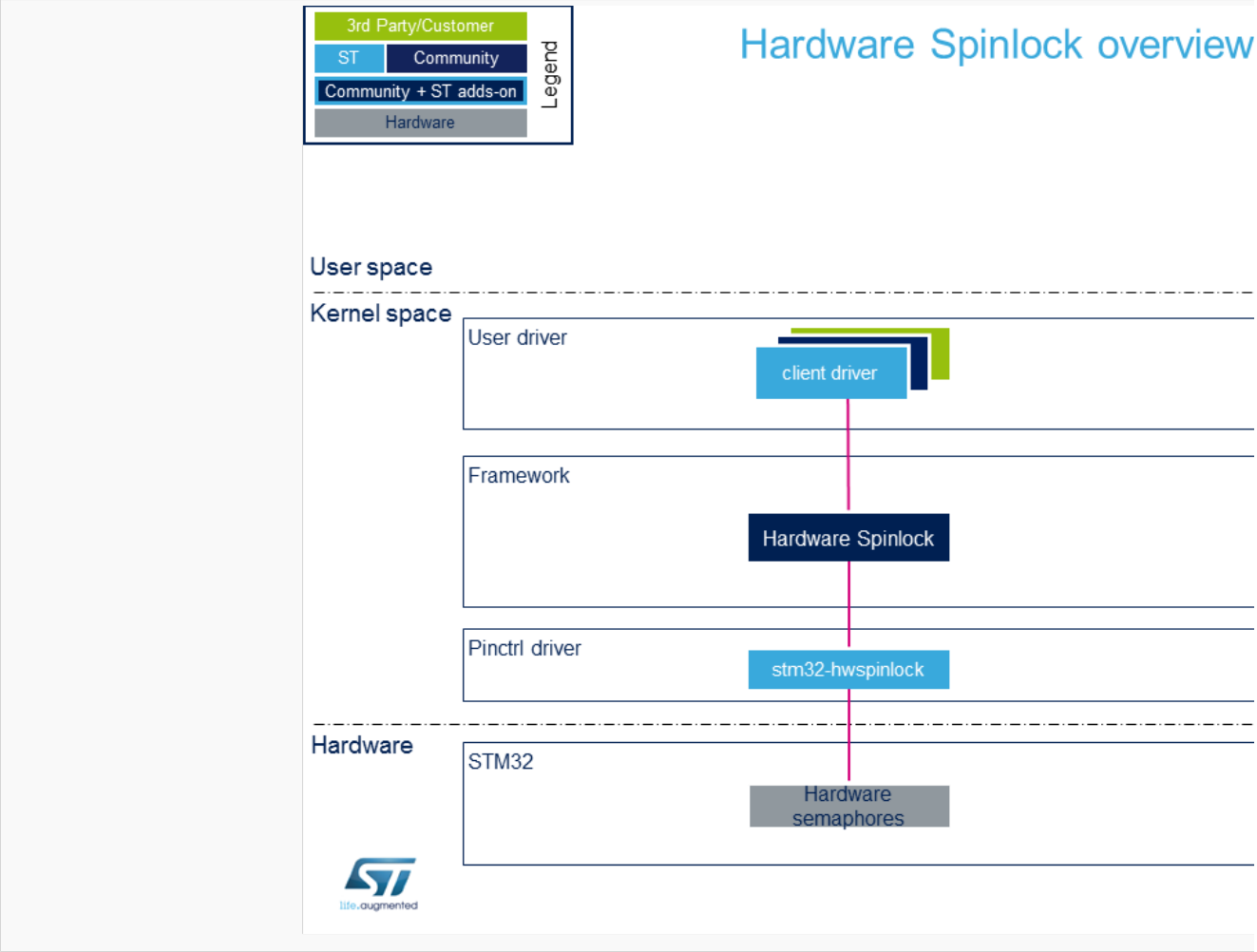
Contents

1 Framework purpose

Hardware spinlock modules provide hardware assistance for synchronization and mutual exclusion between heterogeneous processors and those not operating under a single, shared operating system.

A generic hardware spinlock framework allows platform-independent drivers to use the hardware spinlock device in order to access data structures that are shared between processors, that otherwise have no alternative mechanism to accomplish synchronization and mutual exclusion operations.

2 System overview



2.1 Component description

- **Hardware spinlock:**. The role of this framework is to:
  - provide an API to other drivers
  - call specific vendor callbacks to perform lock and unlock operations
- **stm32-hwspinlock** microprocessor **specific** hardware spinlock driver. The role of this driver is to:
  - register vendor-specific functions (callback) to the hardware spinlock framework
  - access HSEM peripheral registers to perform lock and unlock operations
- **Client driver**
  - Client driver could be any driver that needs to use hardware spinlock to protect a critical section of code

2.2 API description

2.2.1 Kernel space interface

Kernel drivers can be clients of the hardware spinlock framework and can request, lock, unlock and free a hardware spinlock.

Client functions are described in kernel documentation file user API section: Documentation/hwspinlock.txt[1] .

2.2.2 Driver interface

Hardware spinlock driver interfaces (registration, operations) are described in the kernel documentation file API for implementors section: Documentation/hwspinlock.txt[1].

3 Configuration

3.1 Kernel configuration

Hardware spinlock is activated by default in ST deliveries. Nevertheless, if a specific configuration is needed, this section indicates how hardware spinlock can be activated/deactivated in the kernel.

Activate hardware spinlock in the kernel configuration using the Linux Menuconfig tool: Menuconfig or how to configure kernel

```
Device Drivers  --->
  <*> Hardware Spinlock drivers--->
    <*> STM32 Hardware Spinlock device
```

3.2 Device tree configuration

Hardware spinlock bindings[2] documentation deals with all required or optional hardware spinlock generic DT properties.

Detailed DT configuration for STM32 internal peripherals:

```
hsem: hwspinlock@4c000000 {
       compatible = "st,stm32-hwspinlock";
       #hwlock-cells = <1>;
};

foo-client {
      hwlocks = <&hsem 0>; /*Client use lock 0*/
};
```

4 How to use the framework

Typical usage of hardware spinlock by drivers is taken from the kernel documentation file API for a typical usage section: Documentation/hwspinlock.txt[1].:

```
#include <linux/hwspinlock.h>
#include <linux/err.h>

int hwspinlock_example1(void)
{
        struct hwspinlock *hwlock;
        int ret;

        /* dynamically assign a hwspinlock without device tree usage*/
        hwlock = hwspin_lock_request();
        if (!hwlock)
                ...

        id = hwspin_lock_get_id(hwlock);
        /* probably need to communicate id to a remote processor now */

        /* take the lock, spin for 1 sec if it's already taken */
        ret = hwspin_lock_timeout(hwlock, 1000);
        if (ret)
                ...

        /*
        * we took the lock, do our thing now, but do NOT sleep
        */

        /* release the lock */
        hwspin_unlock(hwlock);

        /* free the lock */
        ret = hwspin_lock_free(hwlock);
        if (ret)
                ...

        return ret;
}
```

5 Source code location

Source files are located inside kernel Linux.

- **Hardware spinlock core part**: generic core[3]
- **STM32 hardware spinlock vendor part**: driver code [4]

6 References

1. ↑ [1.0](1.1)[1.1](1.2)[1.2] Documentation/hwspinlock.txt , Hardware spinlock 'inkern' API
2. ↑ Documentation/devicetree/bindings/hwlock/hwlock.txt , Linux Foundation, hardware spinlock generic DT bindings

3. ↑ Hardware spinlock framework source - hwspinlock_core.c Sources of generic hardware spinlock framework
4. ↑ STM32 hardware spinlock driver Provides all vendor specifics functions

Application programming interface

Device Tree