Overview of GPIO pins

Overview of GPIO pins

Stable: 31.01.2020 - 12:58 / Revision: 31.01.2020 - 12:55

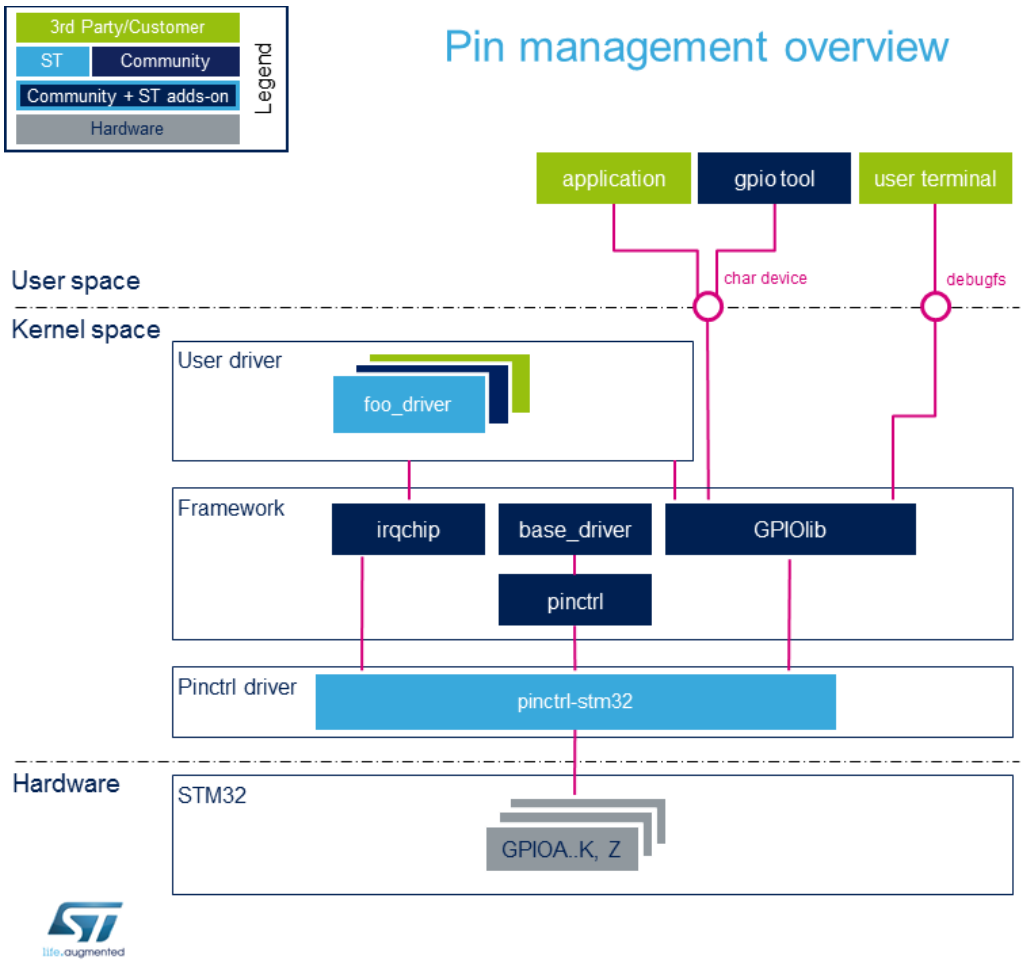
Each STM32 ball/pin is multiplexed in order to support multiple functions. For example, an STM32 pin can operate in three different modes: GPIO, alternate functions or analog. All pin settings are performed via the GPIO internal peripheral, which can be configured through Linux® kernel. This is the purpose of this article.

Contents

1 Purpose

This article introduces the Linux® kernel frameworks dedicated to pin configuration and control. Its purpose is to provide to newcomers some first insights regarding pin management frameworks. Detailed information are provided in the articles referenced in this page.

2 System overview



ST microprocessor pin can be configured in several modes:

- General-purpose input/output (GPIO): controlled by software
- Alternate functions: controlled directly by a hardware block.
- Analog: controlled directly by a hardware block.

Refer to GPIO internal peripheral for more hardware details.

Two frameworks can be used to configure and control a given pin: **pinctrl** and **GPIOLib**. They are selected according to pin usage:

- **pinctrl** is used mainly when a pin is controlled by an internal peripheral. Pinctrl handles the pin configuration and allows assigning a specific feature to a pin. The Pinctrl overview article provides an overview of the pinctrl framework.
- **GPIOLib** is used when a pin needs to be controlled dynamically at runtime (GPIO). GPIOLib is used to control a pin by software. The GPIOLib overview article provides an overview of the GPIOLib framework.

In addition, when a pin is used as external interrupt source, **Irqchip** framework[1] offers an API allowing the configuration of this interrupt.

3 Trainings

Bootlin proposes a detailed presentation of those frameworks: character device interface, *Linux Kernel and Driver Development* training document, see *Introduction to pin muxing* chapter.

4 References

- Interrupt Management training document, *Linux Kernel and Driver Development*

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.)

Application programming interface