# Chapter 4 Laying the Groundwork

/dev: application operates on devices via nodes in the /dev directory

/sys: application gets device information using nodes in the /sys directory

/proc/interrupts: list of active IRQs on the system

## Interrupt Handling

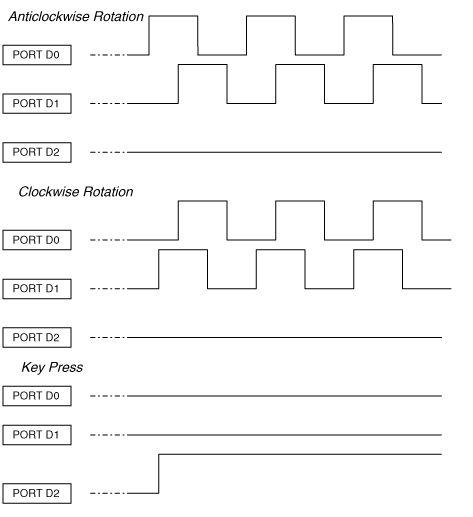
* Each interrupt device is assigned an interrupt number, IRQ number
* Interrupt context code cannot sleep
* Cannot use mutex; use spinlock instead
* Cannot exchange data with user space directly
* ISR is split into 2 sections:
  + Top half: ack the interruot; setup and defer work to bottom half
  + Bottom half: deferred interrupt handler; does the main work
* Need not design interrupt handlers to be reentrant; corresponding IRQ is disabled while in ISR; will not run simultaneously on multiple processor
* Nesting; ISR can be interrupted by IRQ with higher priority

in\_interrupt(): a function can check if it’s in interrupt context

## Roller Wheel

* 3 movements: clockwise rotation, anticlockwise rotation, key-press
* Any of 3 movements generate IRQ 7

The movement is determined from the 3 GPIO connected to the roller device:



### Request IRQ

#define ROLLER\_IRQ 7

// interrupt handler

static irqreturn\_t roller\_interrupt(int irq, void\* dev\_id);

// request IRQ

if (request\_irq(ROLLER\_IRQ, roller\_interrupt, IRQF\_DISABLED |

IRQF\_TRIGGER\_RISING, "roll", NULL)

{

printk(KERN\_ERR "Roll: cannot register IRQ %d\n", ROLLER\_IRQ);

return -EIO;

}

The IRQ is not queried or probed but hard-coded to ROLLOER\_IRQ.

IRQF\_DISABLE: fast interrupt; disable interrupt while invoking the handler

IRQF\_TRIGGER\_RISING: interrupt is generated on the rising edge of interrupt line

IRQF\_SHARED: shared by multiple devices

“roll”: entered to /proc/interrupts tree so it can be identified by applications

IRQ usually is not requested from driver initialization; instead, request when device is opened by an application, and free when application closes device.

free\_irq(int irq, void\* dev\_id);

## Softirqs

Bottom-half; reentrant; different instances of a softirq can run simultaneously on different processors

Ls