

Lab8 report

B09901055 楊健綺

Experiment result:

```
28.322578] vc4-drm soc:gpu: bound 31c00000.v3d (ops vc4_v3d_ops [vc4])
28.322578] [drm] Initialized vc4 0.0.0 20140616 for soc:gpu on minor 0
28.330538] vc4-drm soc:gpu: [drm] Cannot find any crtc or sizes
28.345623] cfg80211: Loading compiled-in X.509 certificates for regulatory database
28.438496] cfg80211: Loaded X.509 cert 'sforshee: 00b28ddf47aef9cea7'
28.521388] cfg80211: loaded regulatory.db is malformed or signature is missing/invalid
28.832783] brcmfmac: FI signature read @0x18000000-0x1541a9a6
28.839822] brcmfmac: brcmf_fw_alloc_request: using brcm/brcmfmac43430-sdio for chip BCM43430/1
28.839833] usbcore: registered new interface driver brcmfmac
29.222157] brcmfmac: brcmf_c_preinit_dcmts: Firmware: BCM43430/1 wl0: Jul 19 2021 03:24:18 Version: 7.45.98 (TOB) (5Gdf937 CY) FWID 01-8e14b897
30.340874] uart-pl011 3f201000.serial: no DMA platform data
30.634394] Adding 102396k swap on /var/swap. Priority:-2 extents:1 across:102396k SSFS
30.733930] 8021q: 802.1Q VLAN Support v1.8
31.337212] brcmfmac: brcmf_cfg80211_set_power_mgmt: power save enabled
31.717648] smsc95xx 1-1.1:1.0 eth0: hardware isn't capable of remote wakeup
31.720307] smsc95xx 1-1.1:1.0 eth0: Link is Down
32.857891] IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
34.853826] ICMPv6: process `dhcpcd' is using deprecated sysctl (sysctl) net.ipv6.neigh.wlan0.retrans_time - use net.ipv6.neigh.wlan0.retrans_time_ms instead
35.069823] Bluetooth: Core ver 2.22
35.069794] NET: Registered PF_BLUETOOTH protocol family
35.069794] Bluetooth: HCI device and connection manager initialized
35.069823] Bluetooth: HCI socket layer initialized
35.069830] Bluetooth: L2CAP socket layer initialized
35.069860] Bluetooth: SCO socket layer initialized
35.085156] Bluetooth: HCI UART driver ver 2.3
35.085182] Bluetooth: HCI UART protocol H4 registered
35.085208] Bluetooth: HCI UART protocol Three-wire (H5) registered
35.085228] Bluetooth: HCI UART protocol Broadcom registered
35.085228] Bluetooth: BNEP (Ethernet Emulation) ver 1.3
35.085244] Bluetooth: BNEP filters: protocol multicast
35.085272] Bluetooth: BNEP socket layer initialized
35.085312] Bluetooth: MGMT ver 1.22
35.085323] NET: Registered PF_ALG protocol family
42.457933] systemd-journal[141]: File /var/log/journal/c9cf628b6a664116afe539d87bc30234/user-1000.journal corrupted or uncleanly shut down, renaming and replacing.
154.2066049] dht11 dht1104 Out: 0 signal input detected
230.590395] dht11 dht1104 Out: 0 signal input detected
```

1. What is Linux IIO subsystem? Please explain IIO in terms of its purpose and user interface provision.

The Linux Industrial I/O (IIO) subsystem is a kernel subsystem that provides support for devices that handle analog-to-digital conversion (ADC) and digital-to-analog conversion (DAC), as well as other types of data acquisition devices. The IIO subsystem is designed to be a generic framework for supporting a wide range of sensors, converters, and similar devices used in embedded systems.

The primary purpose of the IIO subsystem is to provide a unified interface for data acquisition devices. This includes sensors that measure physical quantities (such as temperature, pressure, or acceleration) and converters that transform analog signals to digital values or vice versa. Also, it abstracts the underlying hardware details, providing a consistent API for application developers regardless of the specific hardware being used. This abstraction allows developers to write applications that can work with various IIO-compatible devices without needing to know the intricacies of each device. Finally, IIO is integrated into the Linux kernel, ensures that the subsystem is well-supported and benefits from the Linux kernel's development and maintenance processes.

User Interface Provision:

- a. **sysfs Interface:** Device-specific information, configuration options, and data can be accessed and manipulated by interacting with the sysfs entries associated with IIO devices.
 - b. **Character Device Interface:** IIO also supports a character device interface, allowing users to interact with devices through standard file operations (read, write, etc.). This interface is more flexible than sysfs and provides a way to stream data from sensors.
 - c. **Industrial I/O Framework API:** For application developers, the IIO subsystem provides a programming interface through the Industrial I/O (IIO) framework. This framework includes a set of APIs that applications can use to communicate with IIO devices, retrieve sensor data, and configure device settings.
2. How is the efficiency difference when compared between interrupt-driven I/O and programming I/O (polling I/O)?

Interrupt-Driven I/O use CPU time more efficiently, since the CPU can perform other tasks while waiting for I/O operations to complete. Also, it can have lower latency, for the system can respond quickly to I/O events, as the CPU is notified as soon as data is ready. However, Interrupt-Driven I/O can suffer from more overhead while handling interrupts, especially when there are many interrupts.

On the other hand, Polling I/O is more predictable and is free from unexpected interruptions. Also, it is simpler to implement. But the CPU utilization can be less efficient, as the CPU may spend time repeatedly checking the status of the device even when there is no data to process. And because CPU might not be aware of the device immediately, there may be longer latency.