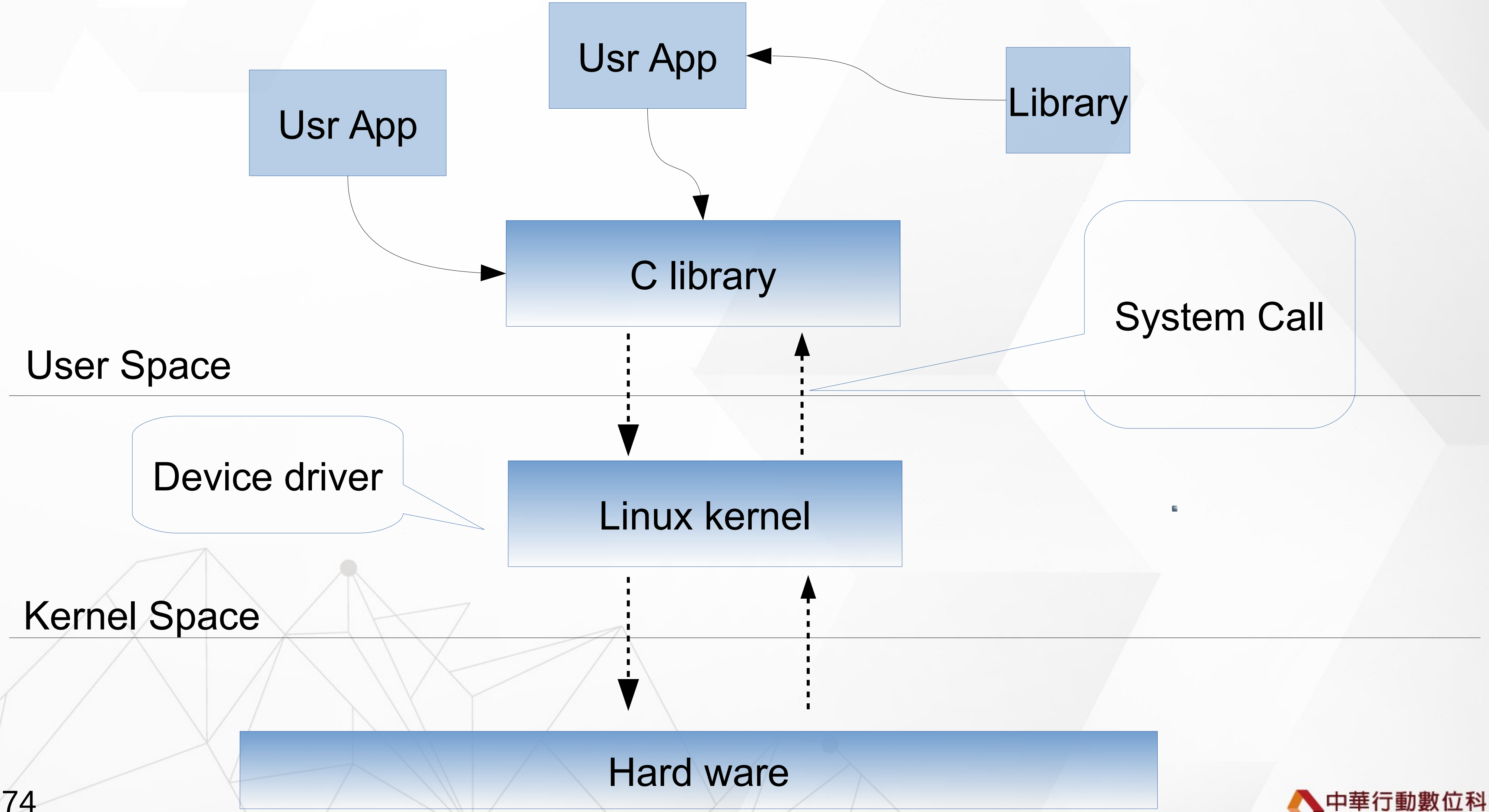
CH9 Linux User Land

Linux kernel

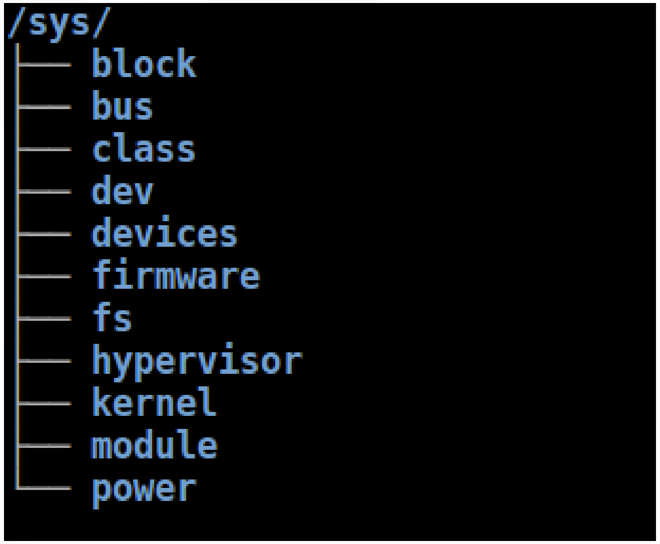


Sys Filesystem

- Allows kernel code to export information to user processes
- sysfs is an in-memory filesystem
- It provides two components
  - a kernel programming interface for exporting these items via sysfs
  - user interface to view and manipulate these items that maps back to the kernel objects which they represent

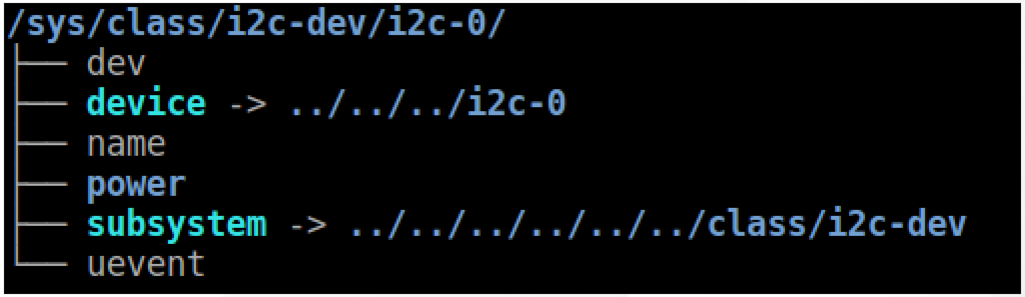
Sys File System

# tree -L 1 /sys/



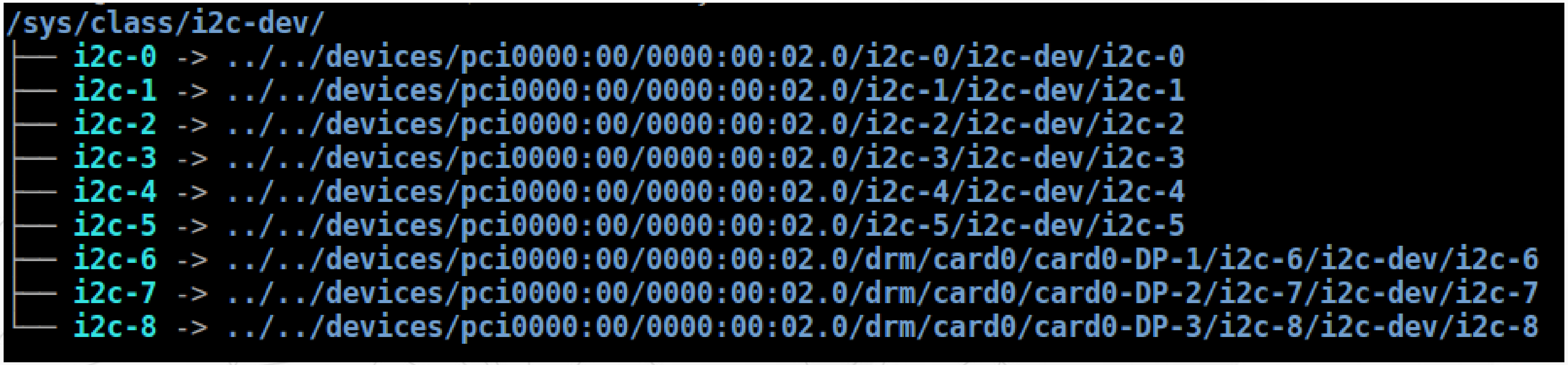
```
/sys/
├── block
├── bus
├── class
├── dev
├── devices
├── firmware
├── fs
├── hypervisor
├── kernel
├── module
└── power
```

# tree -L 1 /sys/class/i2c-dev/i2c-0/



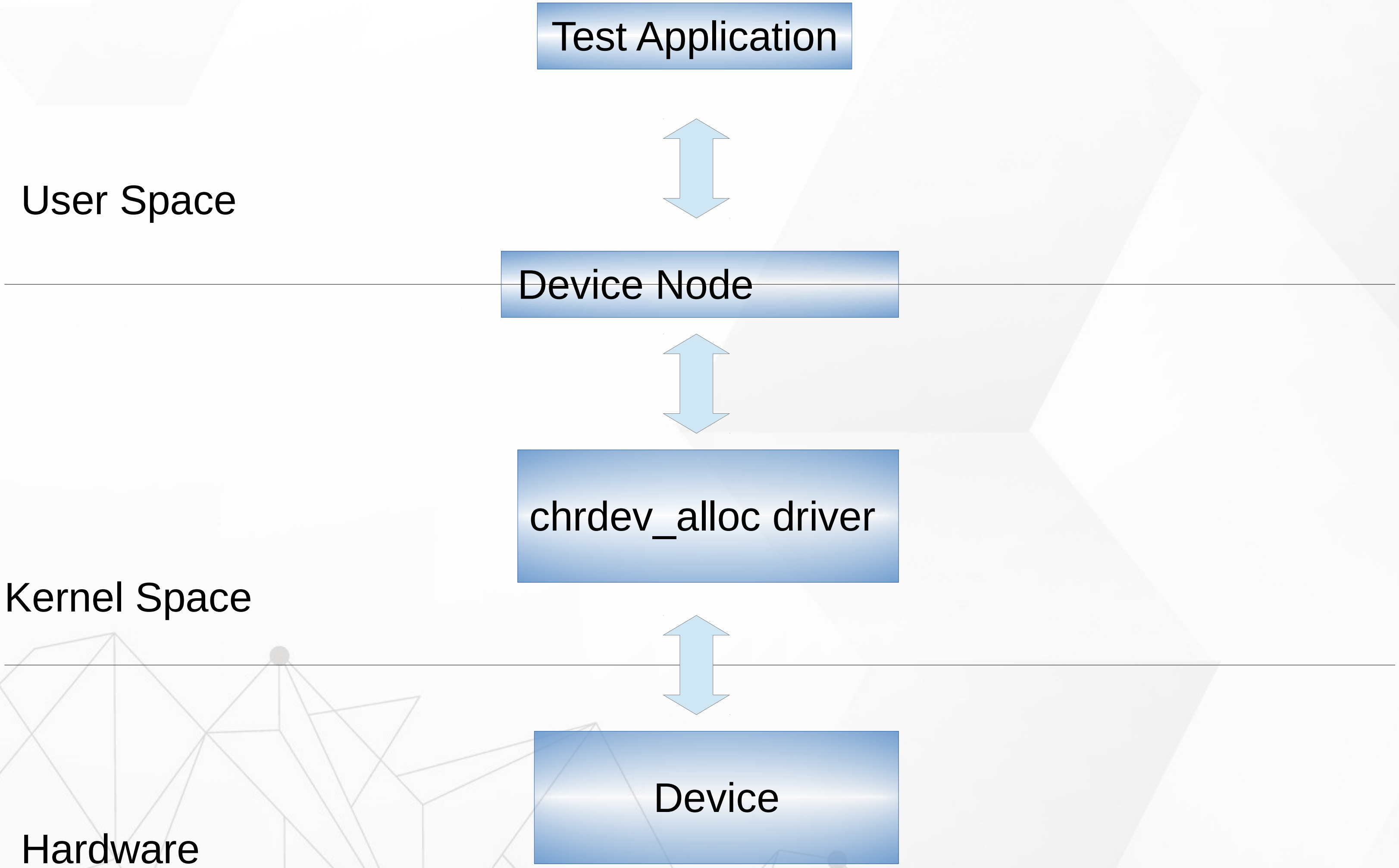
```
/sys/class/i2c-dev/i2c-0/
├── dev
├── device -> ../../../i2c-0
├── name
├── power
├── subsystem -> ../../../../../../class/i2c-dev
└── uevent
```

# tree -L 1 /sys/class/i2c-dev



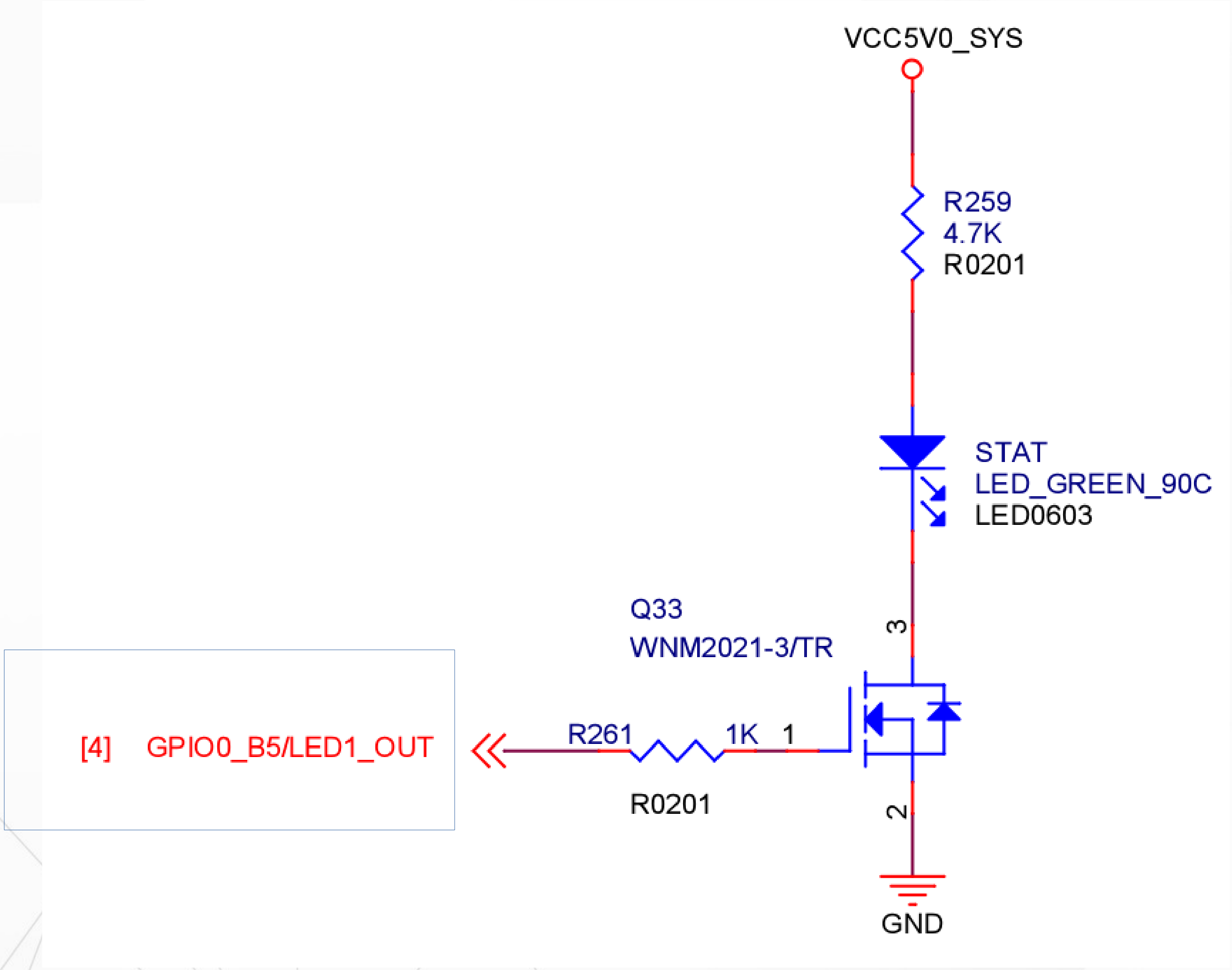
```
/sys/class/i2c-dev/
├── i2c-0 -> ../../devices/pci0000:00/0000:00:02.0/i2c-0/i2c-dev/i2c-0
├── i2c-1 -> ../../devices/pci0000:00/0000:00:02.0/i2c-1/i2c-dev/i2c-1
├── i2c-2 -> ../../devices/pci0000:00/0000:00:02.0/i2c-2/i2c-dev/i2c-2
├── i2c-3 -> ../../devices/pci0000:00/0000:00:02.0/i2c-3/i2c-dev/i2c-3
├── i2c-4 -> ../../devices/pci0000:00/0000:00:02.0/i2c-4/i2c-dev/i2c-4
├── i2c-5 -> ../../devices/pci0000:00/0000:00:02.0/i2c-5/i2c-dev/i2c-5
├── i2c-6 -> ../../devices/pci0000:00/0000:00:02.0/drm/card0/card0-DP-1/i2c-6/i2c-dev/i2c-6
├── i2c-7 -> ../../devices/pci0000:00/0000:00:02.0/drm/card0/card0-DP-2/i2c-7/i2c-dev/i2c-7
└── i2c-8 -> ../../devices/pci0000:00/0000:00:02.0/drm/card0/card0-DP-3/i2c-8/i2c-dev/i2c-8
```

User land and Driver



LED Drivers

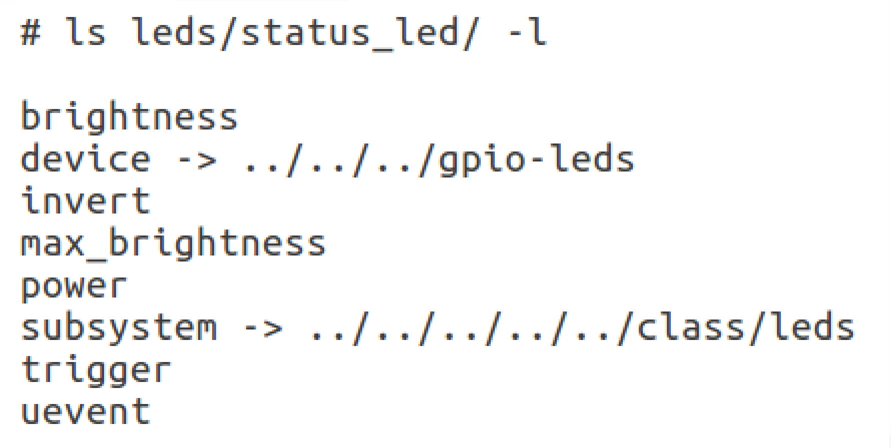
LED Schematic



LED Subsystem

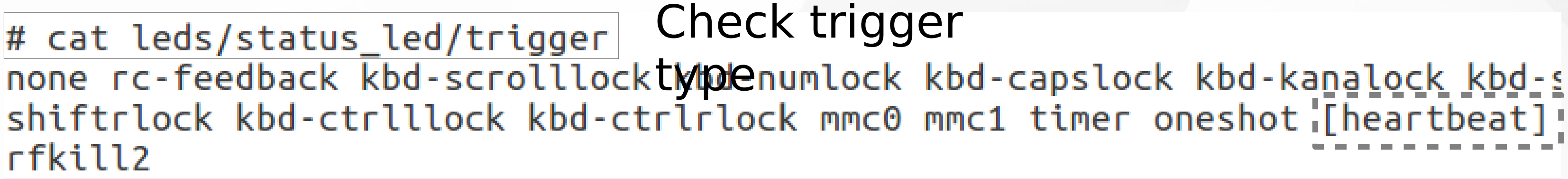
- Control LED convenient with SysFS
  - For example
    - echo 1 > /sys/class/leds/status_led/shot
- Switch different LED trigger type in SysFS
  - For example
    - echo "oneshot" > leds/status_led/trigger
    - echo "heartbeat" > /sys/class/leds/status_led/trigger

LED SysFS



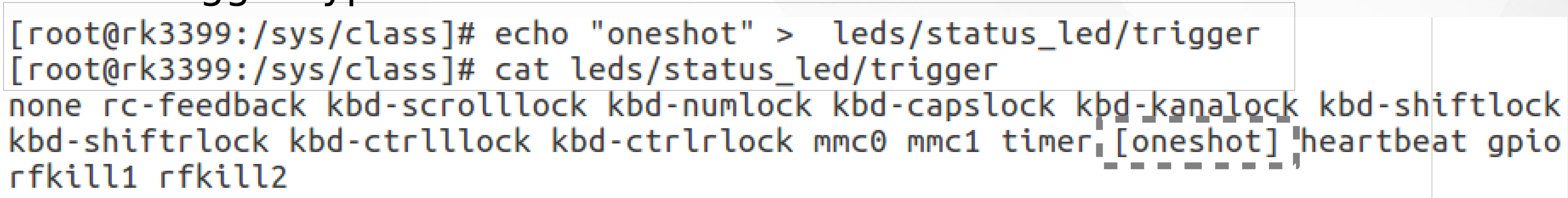
```
# ls leds/status_led/ -l

brightness
device -> ../../../gpio-leds
invert
max_brightness
power
subsystem -> ../../../../../class/leds
trigger
uevent
```

Check trigger type



```
# cat leds/status_led/trigger
none rc-feedback kbd-scrolllock kbd-numlock kbd-capslock kbd-kanalock kbd-s
shiftrlock kbd-ctrlllock kbd-ctrlrlock mmc0 mmc1 timer oneshot [heartbeat]
rfkill2
```

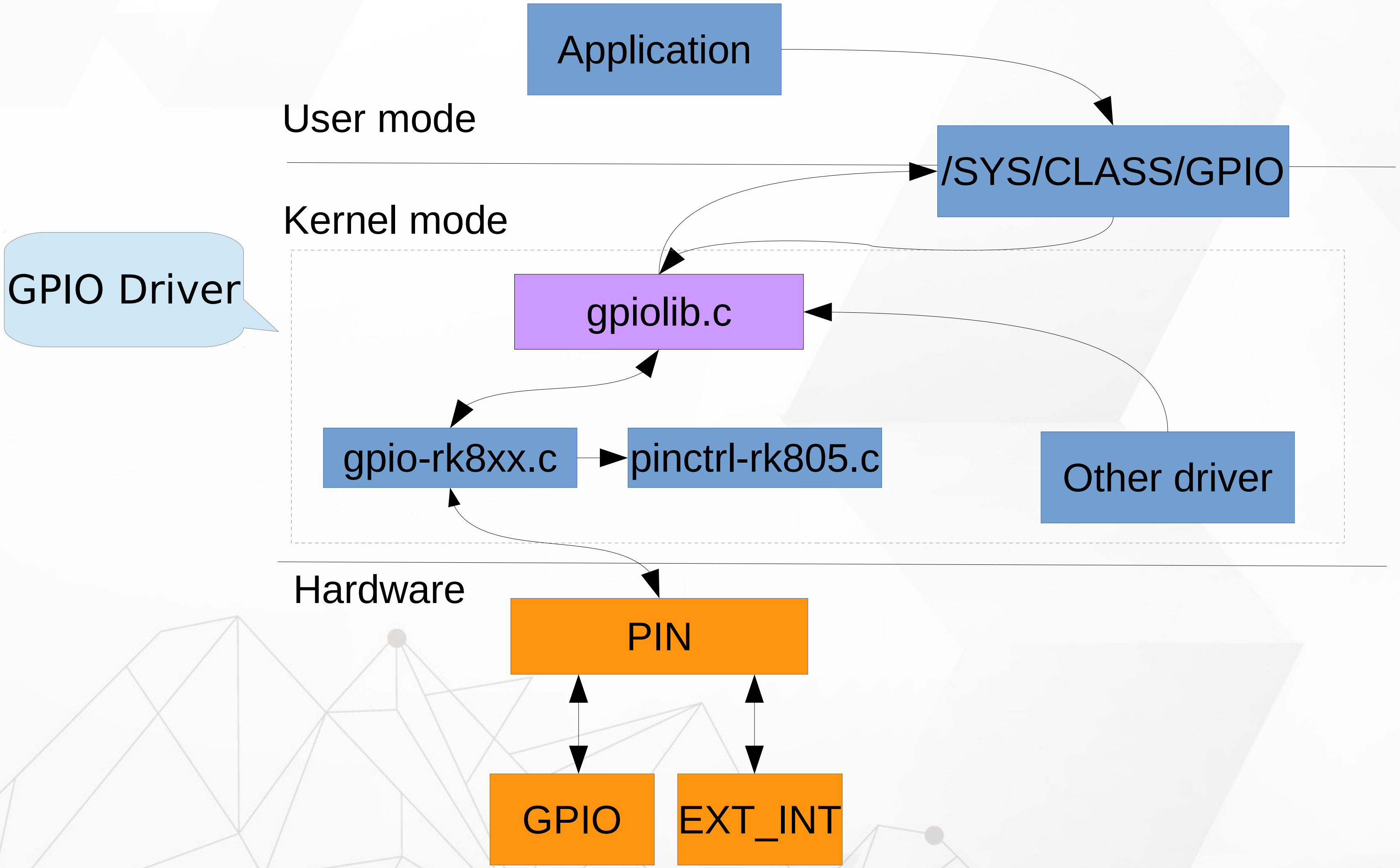
Switch trigger type



```
[root@rk3399:/sys/class]# echo "oneshot" >  leds/status_led/trigger
[root@rk3399:/sys/class]# cat leds/status_led/trigger
none rc-feedback kbd-scrolllock kbd-numlock kbd-capslock kbd-kanalock kbd-shiftlock
kbd-shiftrlock kbd-ctrlllock kbd-ctrlrlock mmc0 mmc1 timer [oneshot] heartbeat gpio
rfkill1 rfkill2
```

GPIO Control

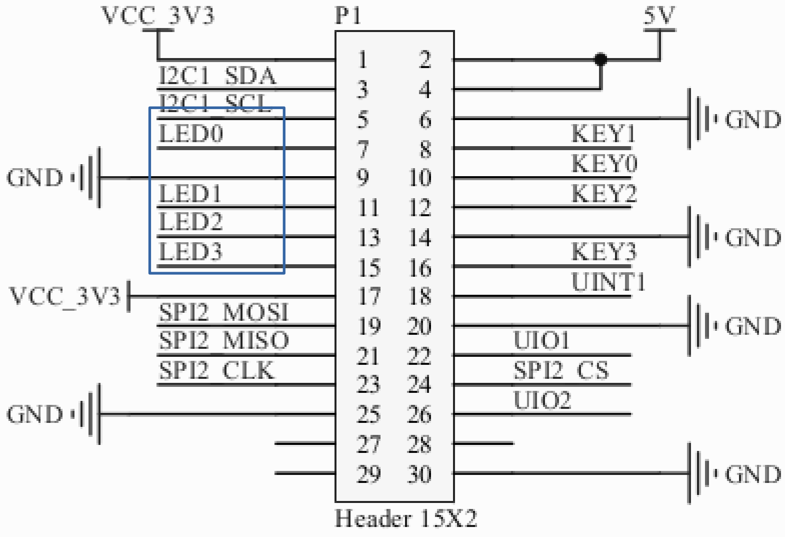
GPIO Subsystem

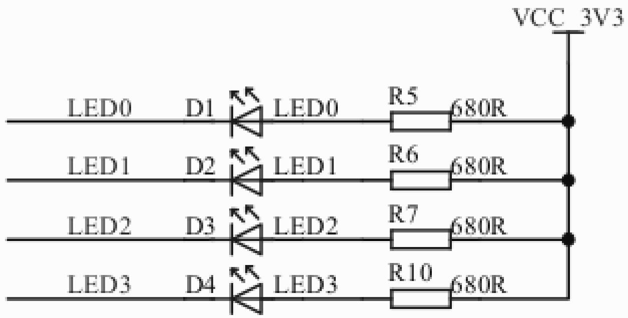


Driver LED in User Space

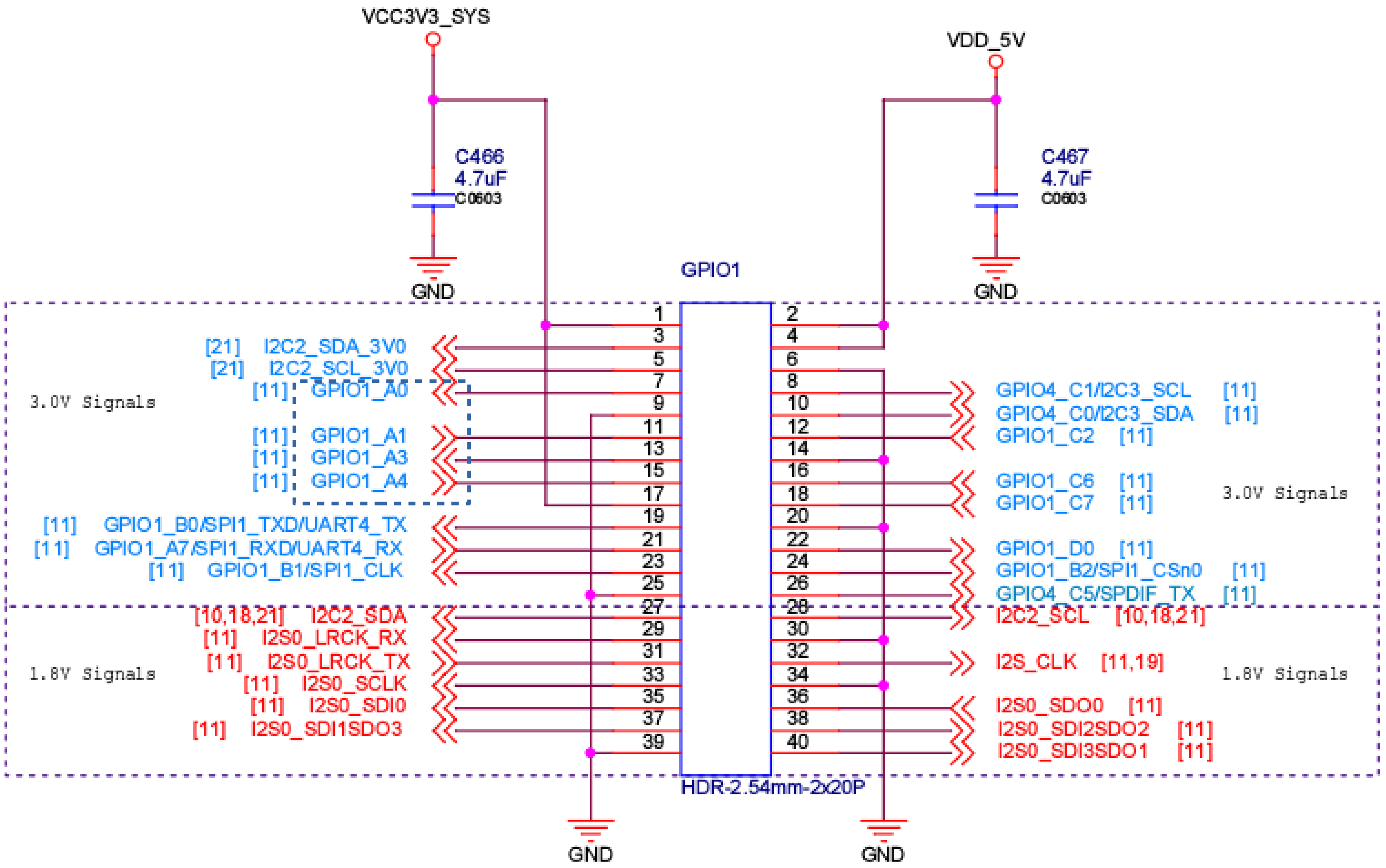
- Paths in Sysfs
- /sys/class/gpio:
  - Control interfaces used to get userspace control over GPIOs;
  - GPIOs themselves
  - GPIO controllers("gpio_chip" instances)
- /sys/class/gpio/
  - "export" : ask the kernel to export GPIO to userspace by writing
    - "echo 19 > export"
    - create a "gpio19" node in /sys/class/gpio
  - "unexport" : Reverses the effect of exporting to userspace
    - "echo 19 > unexport"
    - remove "gpio19" node from /sys/class/gpio

NanoPi-M4 Ext Board LED





NanoPi-M4 GPIO1 HEAD



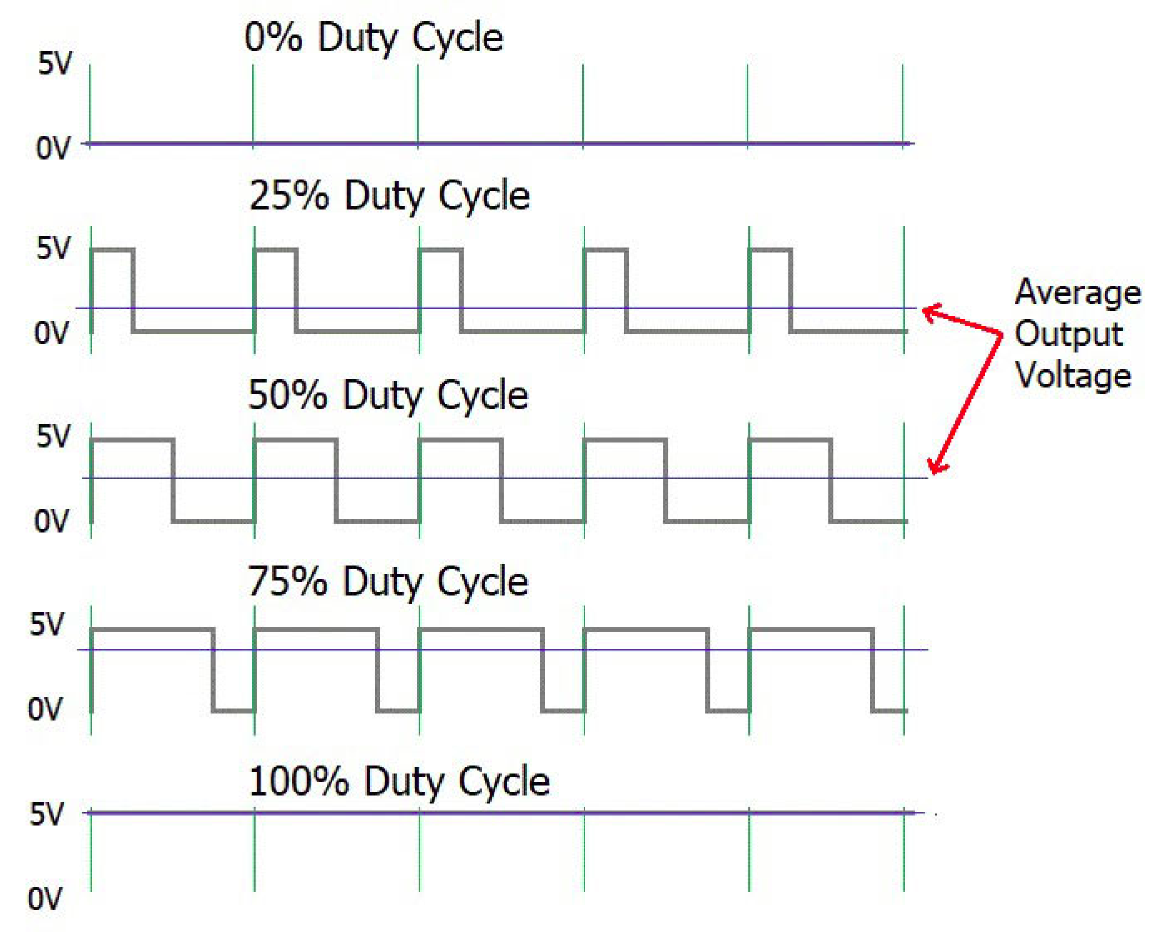
Exercise

- Use “/sys/class/gpio” to setting LED
- $KERNEL/Documentation/gpio/sysfs.txt

PWM Sub System

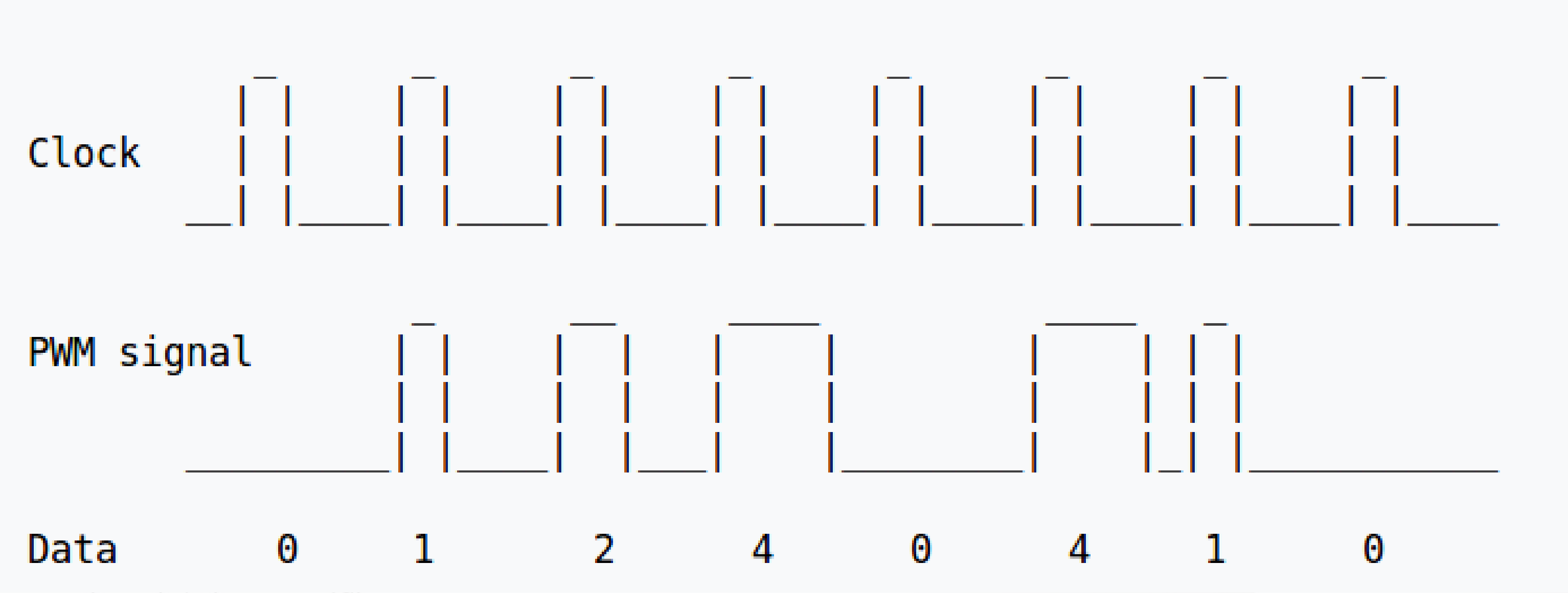
PWM

- PWM : Pulse Width Modulation



https://circuitdigest.com/tutorial/what-is-pwm-pulse-width-modulation

PWM



```
          _      _      _      _      _      _      _      _
         | |    | |    | |    | |    | |    | |    | |    | |
Clock    | |    | |    | |    | |    | |    | |    | |    | |
       __| |____| |____| |____| |____| |____| |____| |____| |____

                 _      __     ____          ____   _
PWM signal      | |    |  |   |    |        |    | | |
                | |    |  |   |    |        |    | | |
       _________| |____|  |___|    |________|    |_| |___________

Data       0     1       2      4      0      4      1      0
```

https://en.wikipedia.org/wiki/Pulse-width_modulation

PWM Parameter in Linux

- Period
  - The total period of the PWM signal
  - Value is in nanoseconds
  - sum of the active and inactive time of the PWM
- duty_cycle
  - The active time of the PWM signal
  - Value is in nanoseconds
  - must be less than the period.

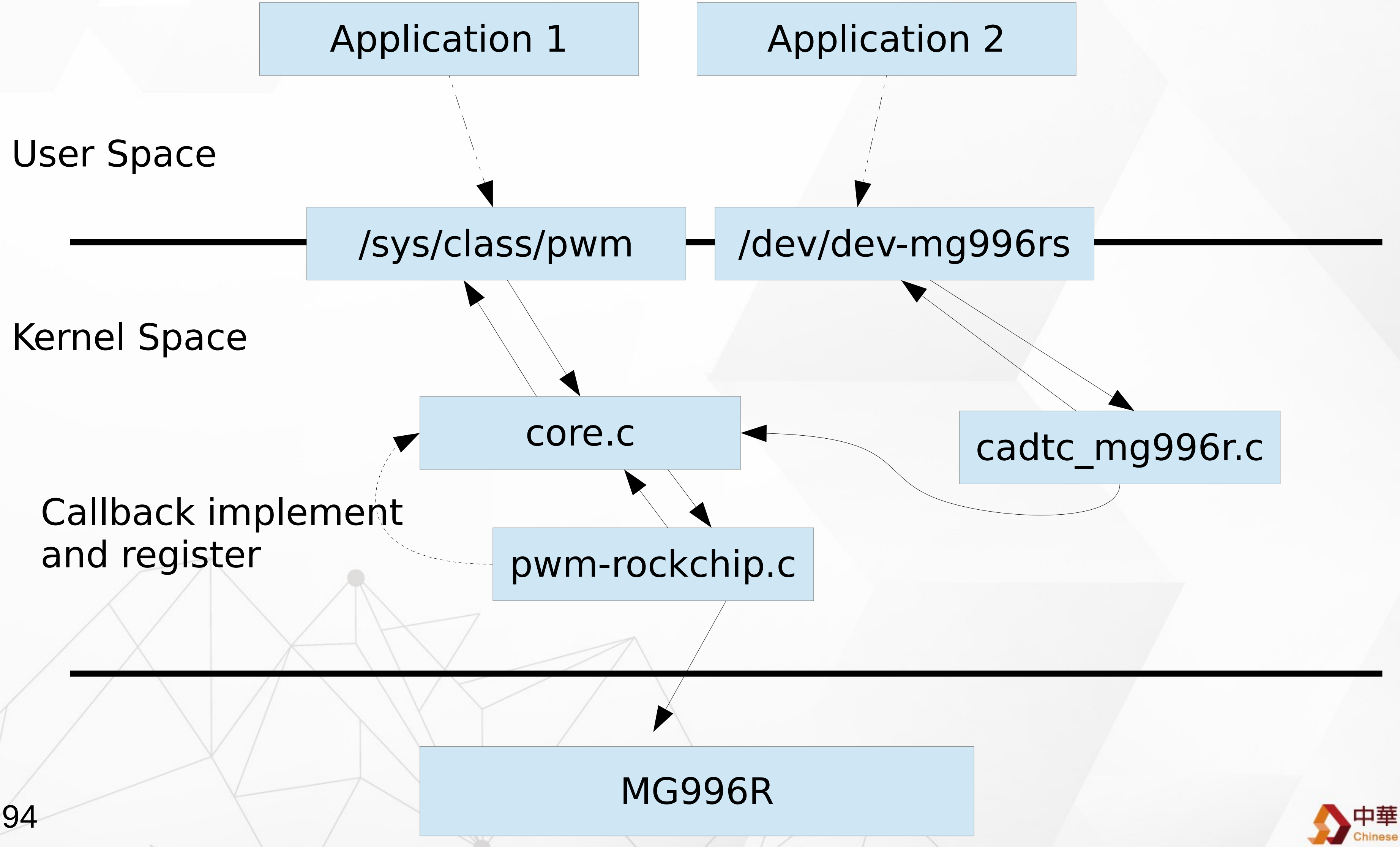
PWM Parameter in Linux

- Polarity
  - The polarity of the PWM signal
- Enable

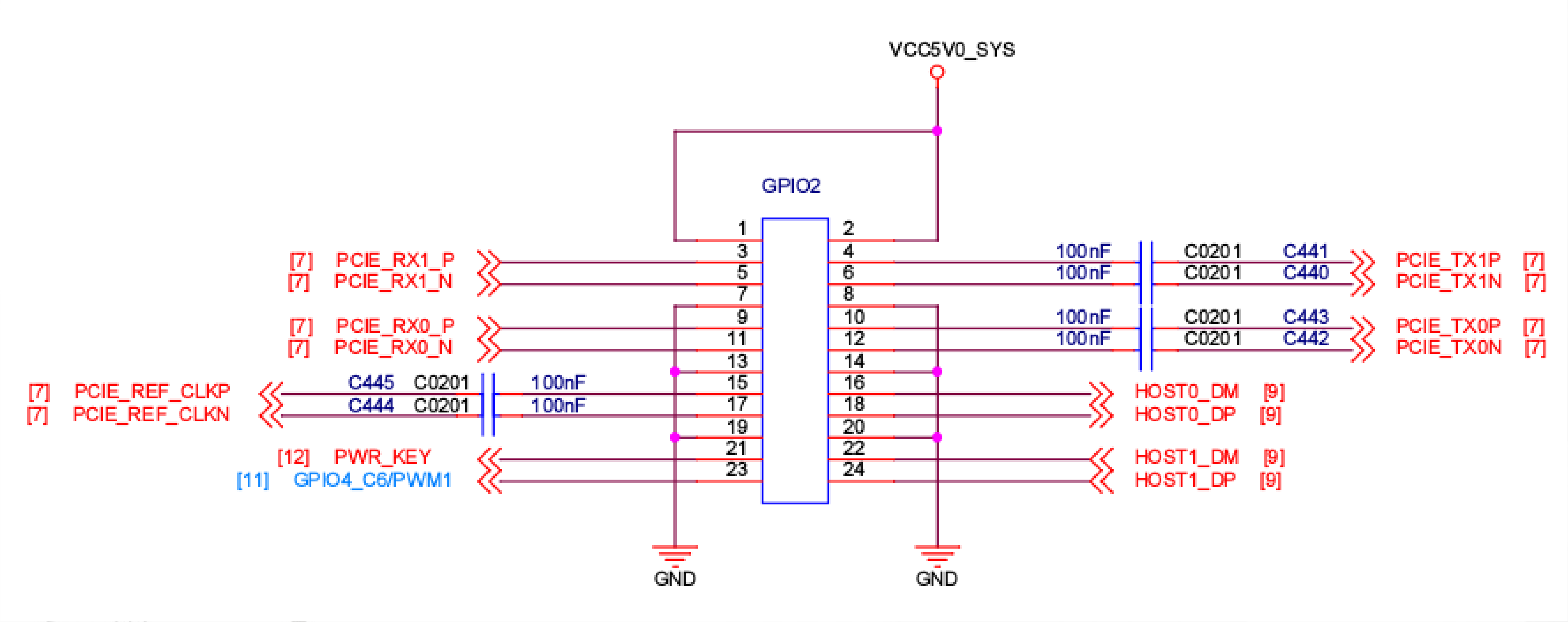
PWM Driver

- $(KERNEL_SRC)/Documentation/pwm.txt
- Platform Driver
  - drivers/pwm/
  - drivers/pwm/core.c
  - drivers/pwm/pwm-rockchip.c

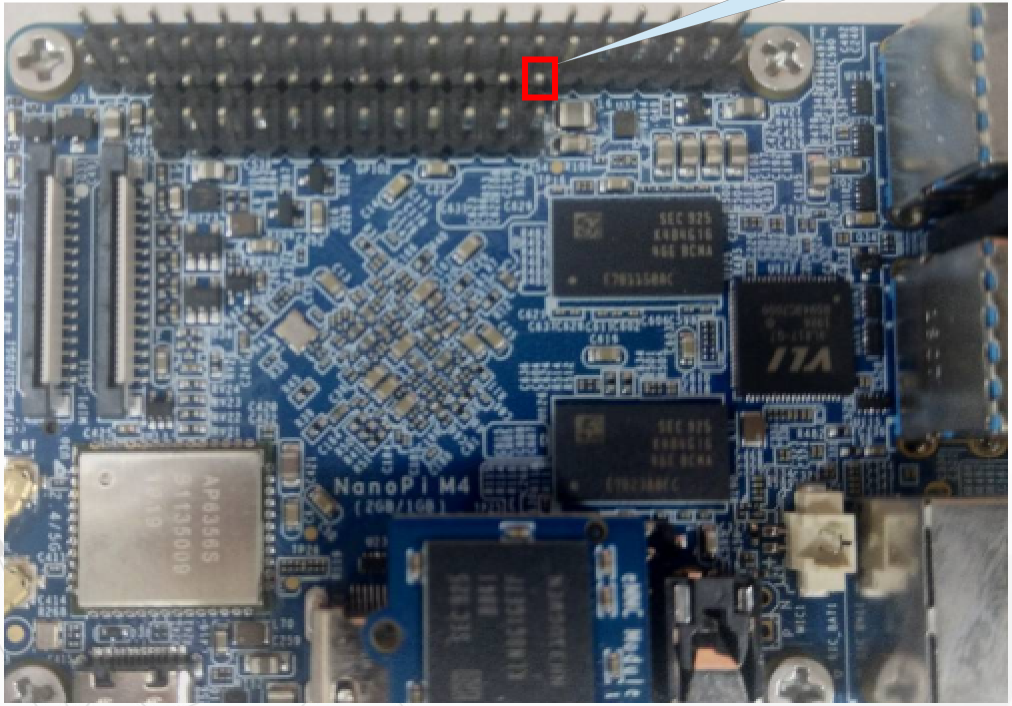
PWM Subsystem



NanoPi-M4 and PWM



NanoPi-M4 and PWM



GPIO4_C6PWM1

PWM SYSFS

```
/sys/class/pwm/pwmchip0

device    export    npwm    power    subsystem  uevent    unexport
```

```
echo 0 > export
capture    enable    polarity    uevent    duty_cycle  period    power
```

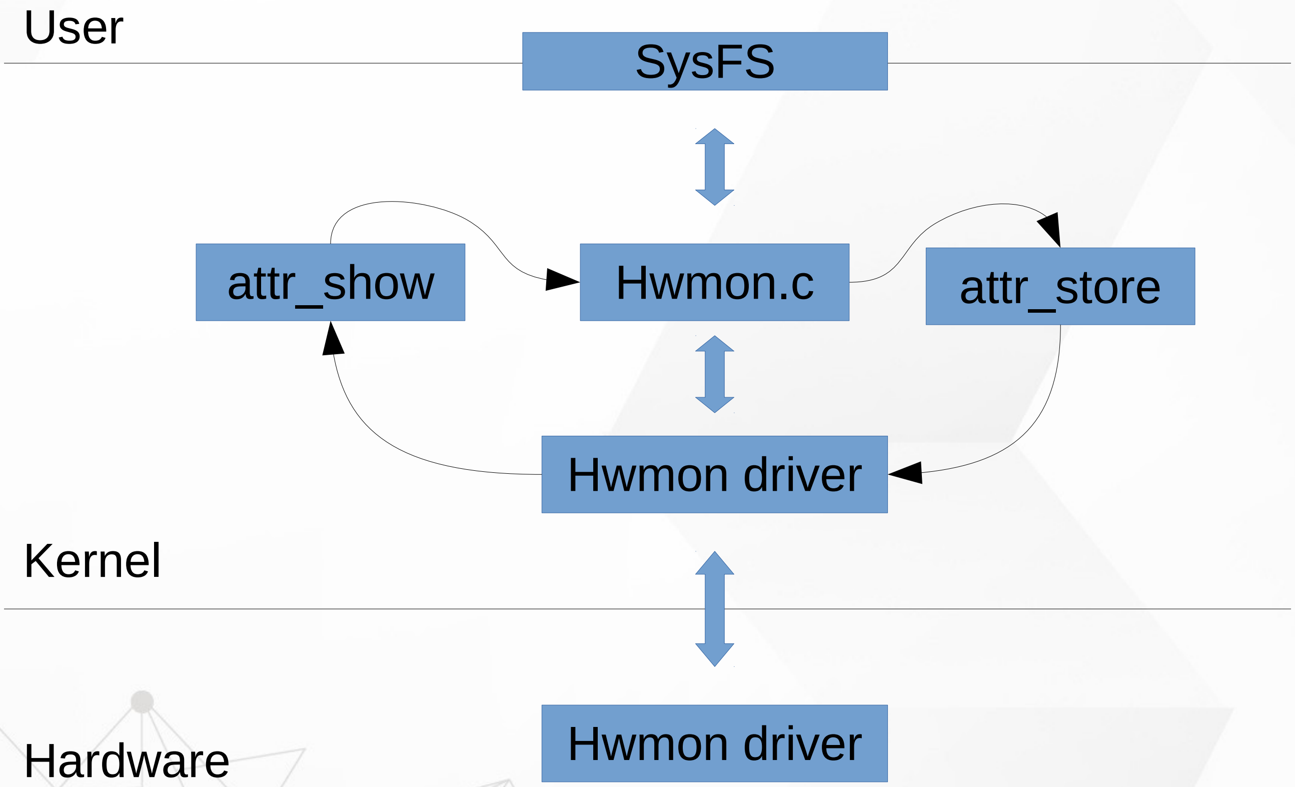
```
echo “20000000” > period        //20ms, 50 Hz
echo "2000000" > duty_cycle     //2ms
echo 1 > enable                 //Enable
```

PWM DoReMi

	Frequency (Hz)
C4	261.63
C4#	277.18
D4	293.66
D4#	311.13
E4	329.63
F4	349.23
F4#	369.99
G4	392.00
G4#	415.30
A4	440.00
A4#	466.16
B4	493.88
C5	523.25

Hwmon Subsystem

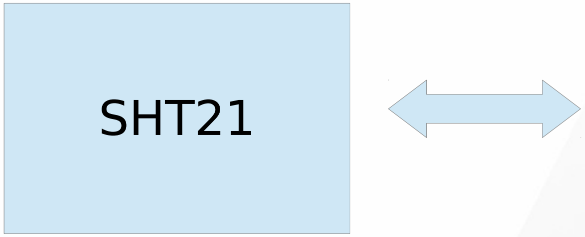
Hwmon Subsystem

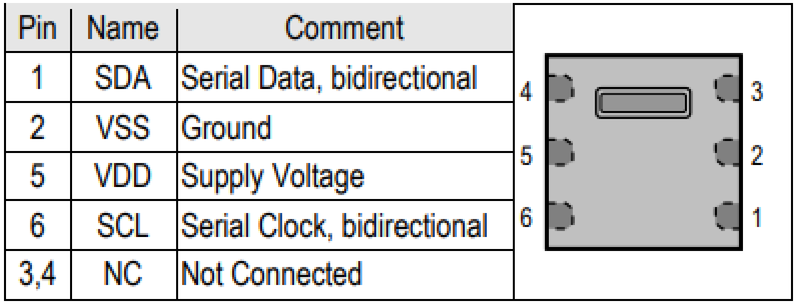


SHT21

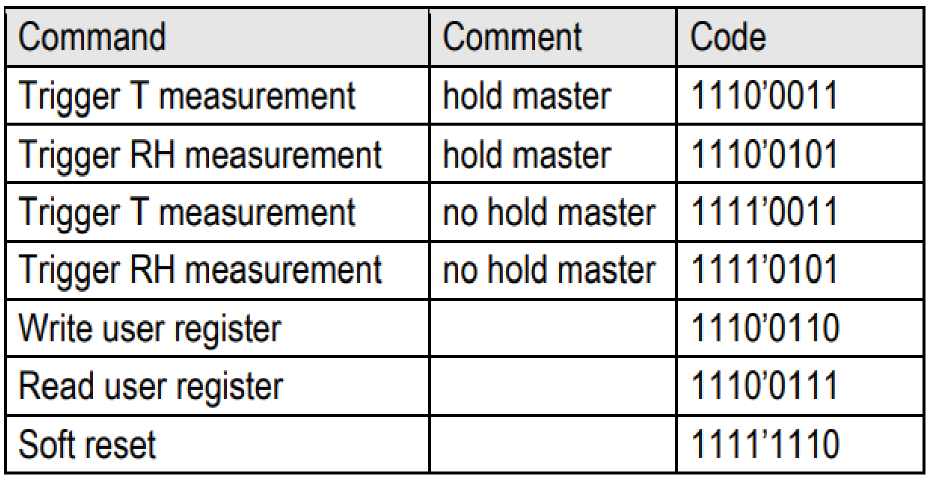
- Simple interface
- Bus interface
  - I2C, GPIO, SPI
- Sensors
  - Temperature
  - Voltage
  - Humidity
  - Fan speed
  - PWM control

SHT21





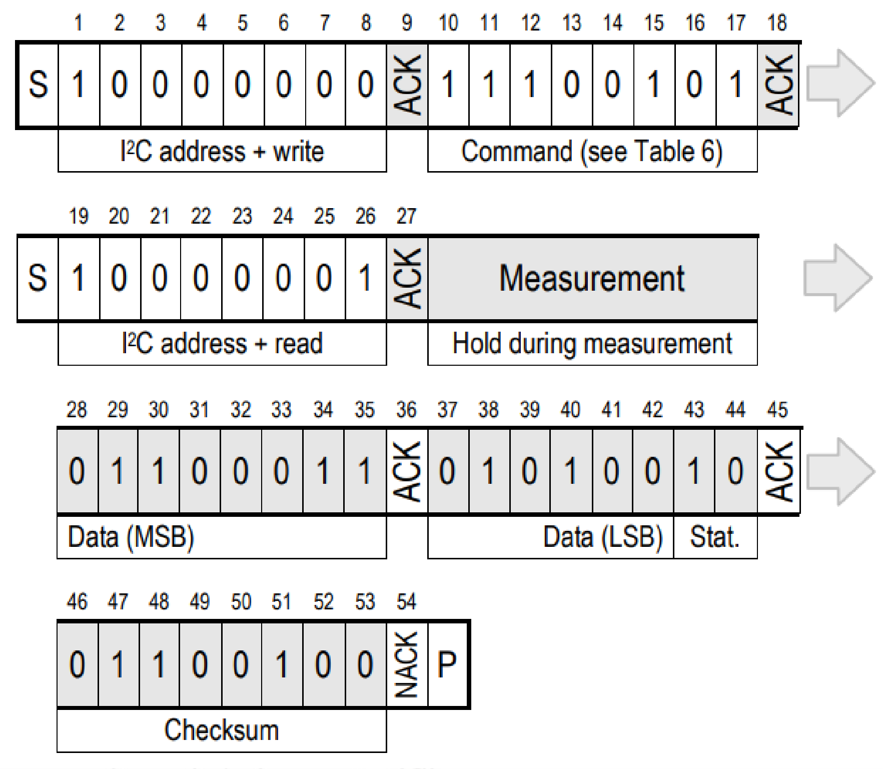
Pin	Name	Comment
1	SDA	Serial Data, bidirectional
2	VSS	Ground
5	VDD	Supply Voltage
6	SCL	Serial Clock, bidirectional
3,4	NC	Not Connected



Command	Comment	Code
Trigger T measurement	hold master	1110’0011
Trigger RH measurement	hold master	1110’0101
Trigger T measurement	no hold master	1111’0011
Trigger RH measurement	no hold master	1111’0101
Write user register		1110’0110
Read user register		1110’0111
Soft reset		1111’1110

SHT21

Hold master communication sequence



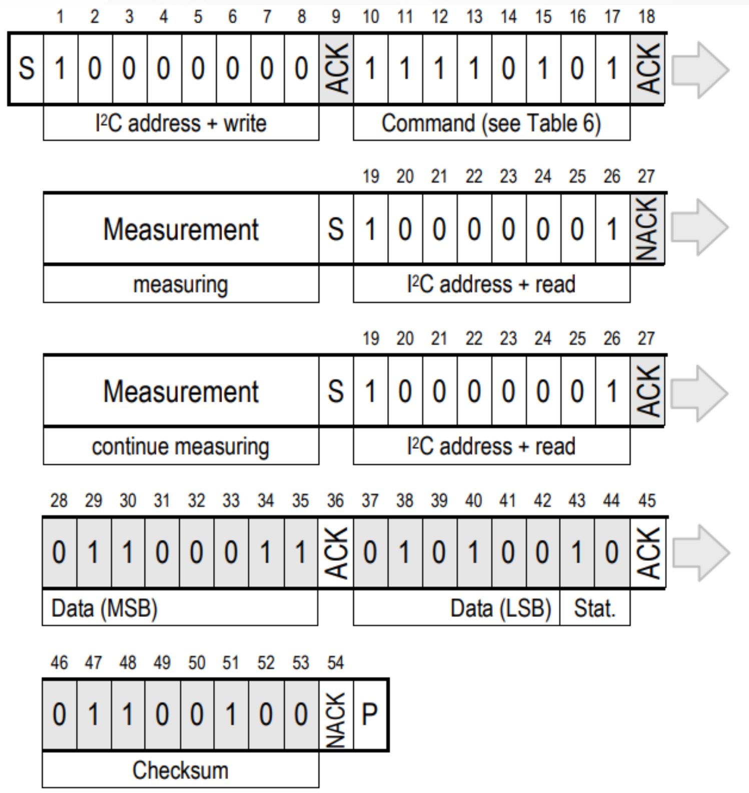
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
S	1	0	0	0	0	0	0	0	ACK	1	1	1	0	0	1	0	1	ACK
	I^{2}C address + write									Command (see Table 6)								

	19	20	21	22	23	24	25	26	27	
S	1	0	0	0	0	0	0	1	ACK	Measurement
	I^{2}C address + read									Hold during measurement

28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
0	1	1	0	0	0	1	1	ACK	0	1	0	1	0	0	1	0	ACK
Data (MSB)									Data (LSB)						Stat.		

46	47	48	49	50	51	52	53	54	
0	1	1	0	0	1	0	0	NACK	P
Checksum									

No Hold master communication sequence



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
S	1	0	0	0	0	0	0	0	ACK	1	1	1	1	0	1	0	1	ACK
	I^{2}C address + write									Command (see Table 6)								

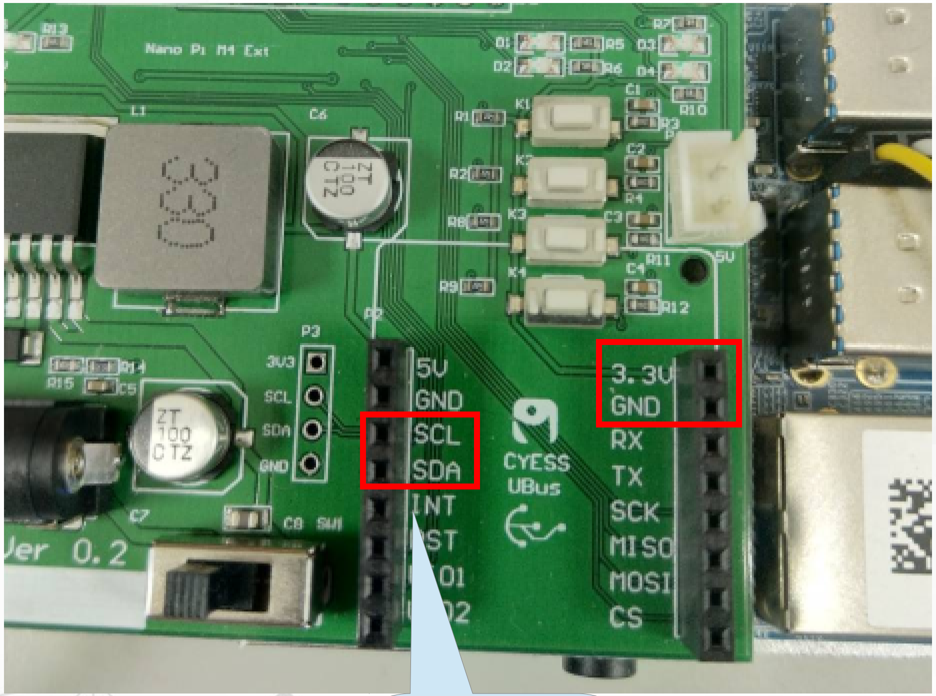
		19	20	21	22	23	24	25	26	27
Measurement	S	1	0	0	0	0	0	0	1	NACK
measuring		I^{2}C address + read								

		19	20	21	22	23	24	25	26	27
Measurement	S	1	0	0	0	0	0	0	1	ACK
continue measuring		I^{2}C address + read								

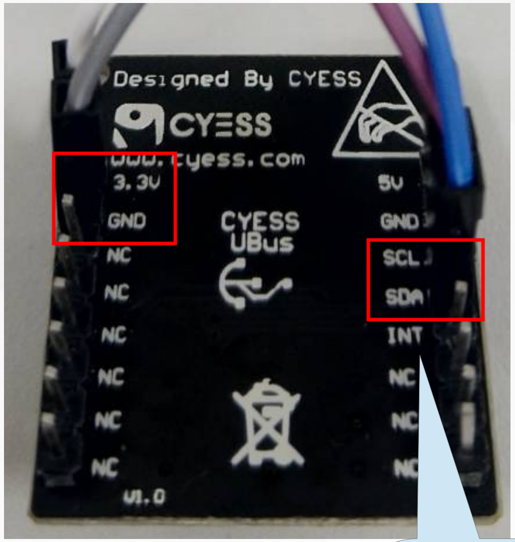
28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
0	1	1	0	0	0	1	1	ACK	0	1	0	1	0	0	1	0	ACK
Data (MSB)									Data (LSB)						Stat.		

46	47	48	49	50	51	52	53	54	
0	1	1	0	0	1	0	0	NACK	P
Checksum									

NanoPi-M4 and SHT21



I2C



I2C

SHT21

Hwmon Sysfs

```
# ls /sys/class/hwmon/hwmon0
device           name   subsystem    uevent
humidity1_input  power  temp1_input
```

temperature

```
# cat /sys/class/hwmon/hwmon0/temp1_input
32279
```

humidity

```
# cat /sys/class/hwmon/hwmon0/humidity1_input
34512
```

IIO Subsystem

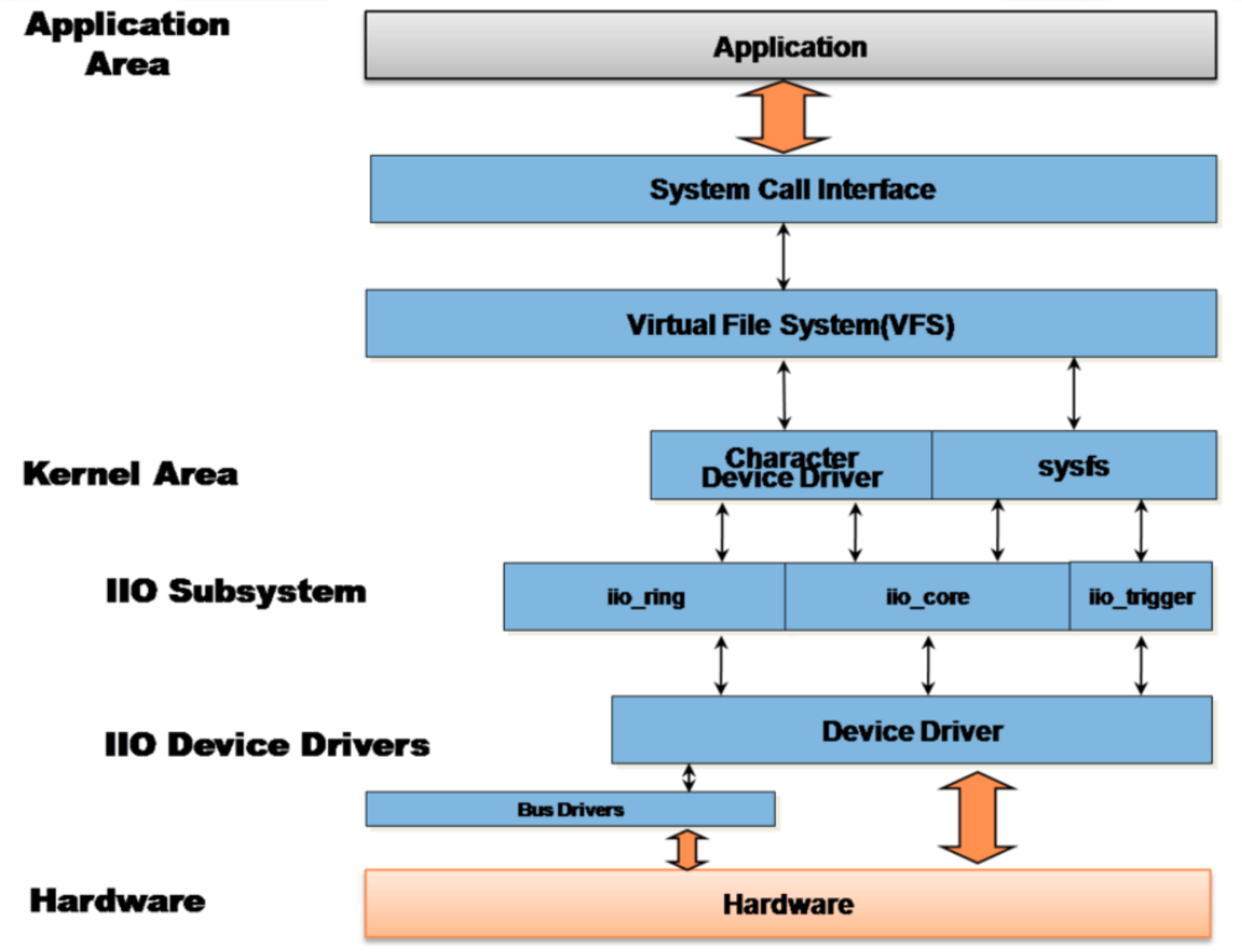
IIO Introduction

- IIO - The Industrial I/O
- support for devices that in some sense
  - analog to digital (ADC)
  - digital to analog converters (DAC)
- Devices that fall into this category are
  - ADCs
  - Accelerometers
  - Gyros
  - DAC
  - Pressure Sensors

IIO Introduction

- Fill the gap between the somewhat similar hwmon and input subsystems
- Hwmon is very much directed at low sample rate sensors used in applications
    - fan speed control
    - temperature measurement.
- Input is, as it's name suggests focused on human interaction input devices

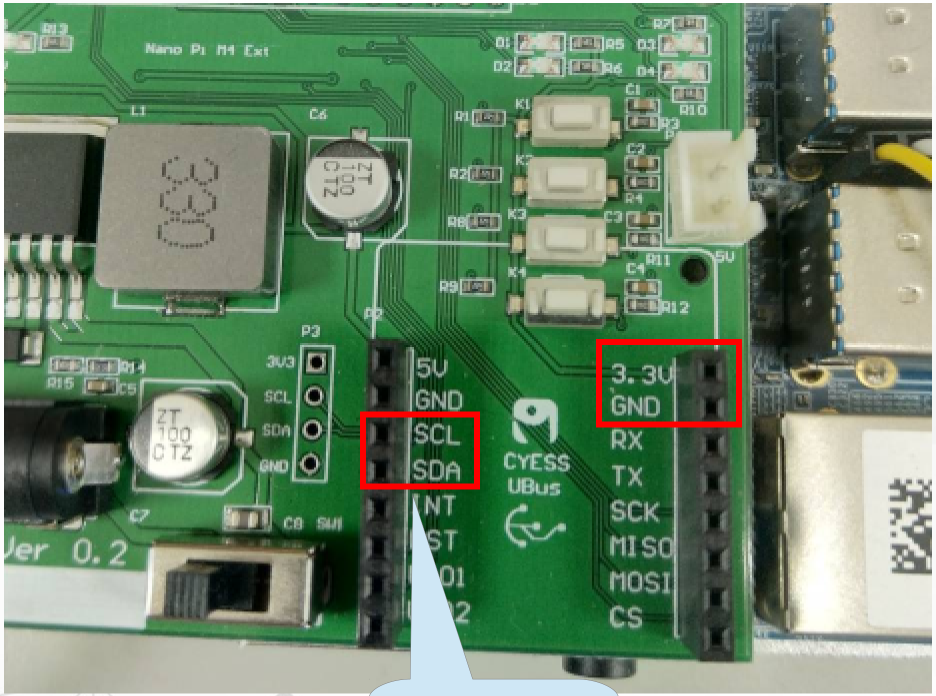
IIO Introduction



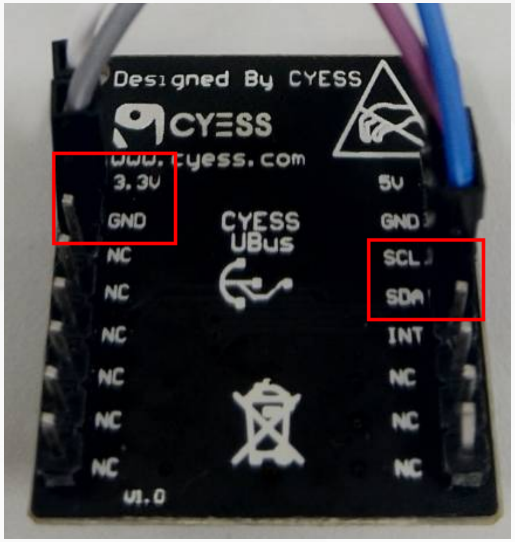
IIO Interface

- There are 2 ways for a user space application to interact with an IIO driver
- **/sys/bus/iio/iio:deviceX/**
  - data channels
- **/dev/iio:deviceX**
  - buffered data transfer
  - events information

NanoPi-M4 and ISL29023



I2C



IIO and ISL29023

Get value from SysFS attribute

```
$ cat /sys/bus/iio/devices/iio:device1/in_illuminance0_input
221
```