# CH3 Basic Software and Tool

# Software and Tool

- Open Source License
- Develop Tool
  - → Geany, gedit, vim
  - → Git
  - → diff, patch
- Build Code Tool
  - → ARM toolchain
  - → make
  - → automake, autoconfig

# Software and Tool

- Network
  - → WiFi, Ethernet, Net tool
  - → Bluetooth
  - → SSH, SSHFS
  - → NFS
- Media Software
  - → gstreamer
  - → ALSA Tool - aplay, arecord

# Software and Tool

- Bus
  - → I2C – I2cset, i2cget, i2cdump
  - → USB – lsusb

# Open Source License

## GNU General Public License

→ 只要在一個軟件中使用 (" 使用 " 指類庫引用，修改後的代碼或者衍生代碼 )
GPL 協議的產 品，則該軟件產品必須也採用 GPL 協議，既必須也是開源和免費 .
這就是所謂的 " 傳染性 "

## BSD License

→ 基本上使用者可以 " 為所欲為 ", 可以自由的使用，修改源代碼，
也可以將修改後的代碼作為開源或者專有軟件再發佈 .

## LGPL

→ LGPL 是 GPL 的一個為主要為類庫使用設計的開源協議 . LGPL 允許商 業軟件
通過類 庫引用 (link) 方式使用 LGPL 類庫而不需要開源商業軟件的代碼 . 這使得採用
LGPL 協議的開源代碼可以被商業軟件作為類庫引用並發布和銷售 .

# Develop Tool

# Ubuntu Package Management

- apt-get : command-line tool for handling packages
- apt-get --help
  - → apt-get update
  - → apt-get  install ${PACKAGE_NAME}
  - → apt-get  remove ${PACKAGE_NAME}
  - → apt-get  autoremove
  - → apt-get  clean

# Geany

- You can find a good edit for programing

→ Geany

https://www.geany.org/
$ sudo apt-get install geany

→ Vim
$ sudo apt-get install vim

→ gedit

# Tracking code command

- Linux command
  - Filter :
    → grep -r -n “function name”
  - Fine special file include “String”
    → find -name “*.c” | xarge grep -n “String”

# diff and patch

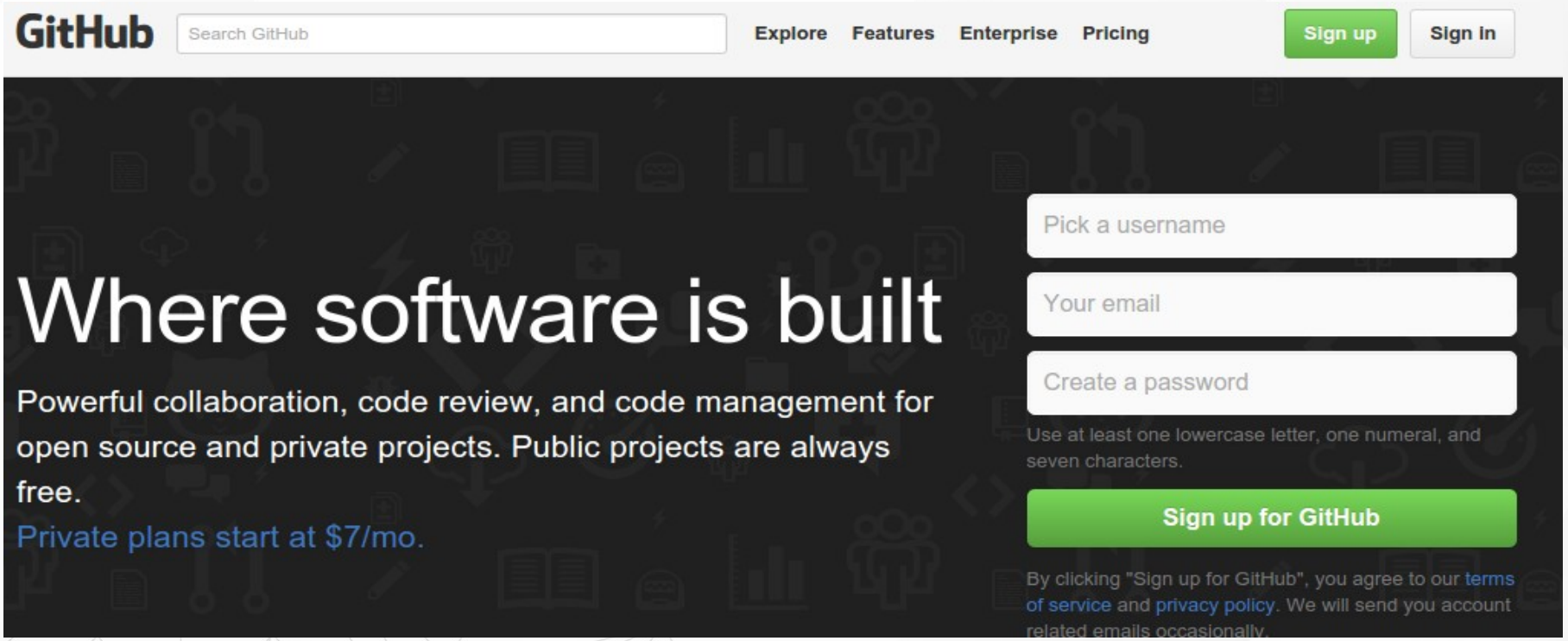
- diff - compare files line by line
  - Create a patch file
    - diff -Nuar file_a file_b >  c.patch
      - -N, treat absent files as empty
      - -a, --text
      - -u, output NUM (default 3) lines of unified context
      - -r,  recursively compare any subdirectories found
- patch - apply a diff file to an original
  - apply a patch file
    - patch ./hello_1.c < ./tmp.patch
  - Reverse a patch file
    - patch -R ./hello_1.c < tmp.patch

# Git

- https://git-scm.com/book/zh-tw/v1/
- 版本控制
- 程式回溯
- 管理多人共同開發

# GitHub

- https://github.com/



# Exercise

- 0. Create an empty Git repository (In local)
  $ git init
- 1. Clone code to local
  $ git clone https://github.com/xlloss/tiny4412-uboot.git
- 2. modify something
  $ gedit README
- 3. check source status
  $ git status
- 4. use "git add <file>..." to update what will be committed
  $ git add ./README
- 5. check status again
  $ git status

# Exercise

- 6. commit code to local repository
  $ git commit -a “test”
  Or $ git commit
- 7. check log
  $ git log
- 8. check how many branch in local repository
  $ git branch
- 9. create new branch in local repository
  $ git branch “new_branch_name”
  $ git branch cadtc_uboot

# Exercise

- 10. check out to new branch
  $ git checkout “branch_name”
  $ git checkout slash_uboot
- 11. check branch again
  $ git branch
- 12 . push log branch to remote
  $ git push origin slash-uboot
- 13. check remote branch status
  $ git branch origin/ and push tab x2

# Exercise

- reset your code, but modify code still live
  $ git reset commit hash coed
- Hard reset your code, all modify code will discard
  $ git reset - -hard hash coed
- Check log
  $ git log
  $ git show
- Download objects and refs from another repository
  $ git fetch [--all]

# BASIC Git Command

- init Create an empty Git repository
- add Add file contents to the index
- branch List, create, or delete branches
- checkout Checkout a branch or paths to the working tree
- clone Clone a repository into a new directory
- commit Record changes to the repository

# BASIC Git Command

- diff Show changes between commits, commit and working tree, etc
- rm Remove files from the working tree and from the index
- pull Fetch from and merge with another repository or a local branch
- push Update remote refs along with associated objects
- reset Reset current HEAD to the specified state
- cherry-pick apply changes introduced by some existing commits

# Build Code Tool

# Makefile

# Makefile

- Simplify compile command
- Automation compile, linker program source
- It can update source in accordance with the dependence

# Compile a Hello_World

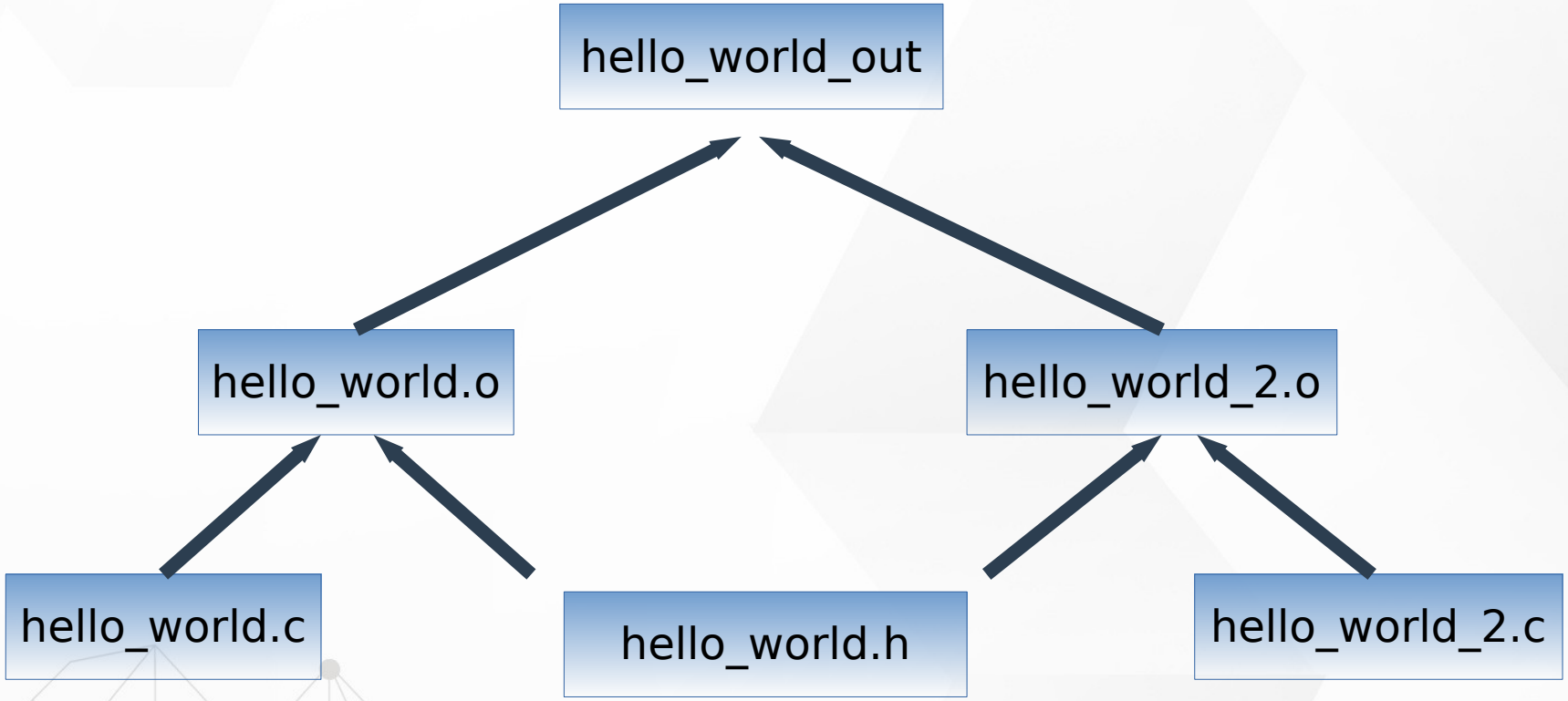
| A | B | C |
| --- | --- | --- |
| aarch64-linux-gnu-gcc | -o helloworld | ./hello.c |

A : ARM C Compile

B : ARM C Compile Parameter (Output name)

C : C source code

# Another Sample



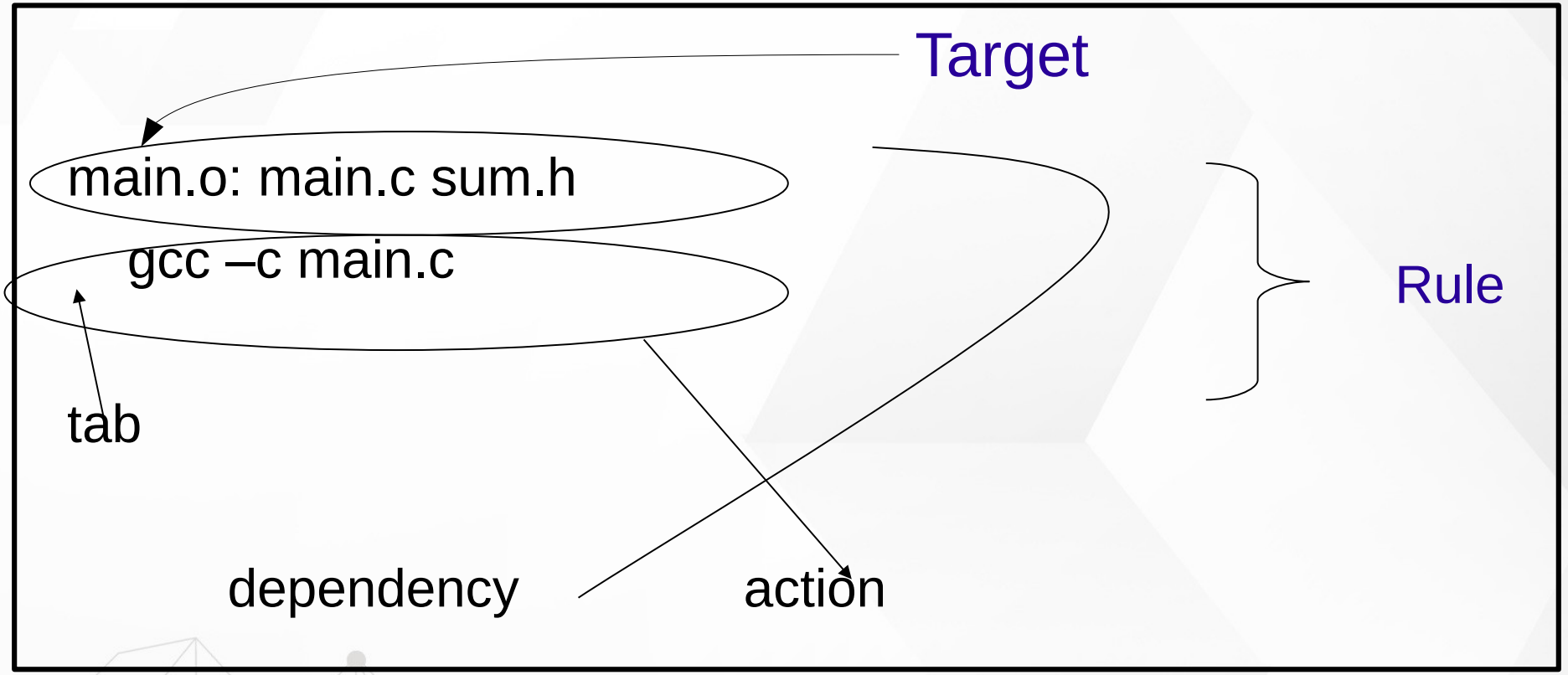
# Compile Another Sample

- Step 1 : gcc -c hello_world_2.c
- Step 2 : gcc -c hello_world_2.c
- Step 3 : gcc -o hello_world hello_world.o hello_world_2.o

# Another Sample - Makefile

```
CC=$(CROSS_COMPILE)gcc


all: hello_world

hello_world: hello_world.o hello_world_2.o
	$(CC) -o hello_world hello_world.o hello_world_2.o

hello_world.o: hello_world.c
	$(CC)  -c hello_world.c

hello_world_2.o: hello_world_2.c
	$(CC)  -c hello_world_2.c

clean:
	rm -r *.o
	rm hello_world
```

# Rule Syntax




**make 在編譯時，若發現 target 比較新，**
**也就是 dependencies 都比 target 舊，**
**那麼將不會重新建立 target，如此可以避免不必要的編譯動作**

# Rule Syntax

```
CC=$(CROSS_COMPILE)gcc

CFLAGS=-Wl,-Map,out.map -lpthread -lm

all: hello_world

hello_world: hello_world.o hello_world_2.o
	$(CC) -o hello_world hello_world.o hello_world_2.o

hello_world.o: hello_world.c
	$(CC) $(CFLAGS) -c hello_world.c

hello_world_2.o: hello_world_2.c
	$(CC) $(CFLAGS) -c hello_world_2.c

clean:
	rm -r *.o
	rm hello_world
```

# Rule Syntax Sample 1

```
hello_world_2.o: hello_world_2.c
	$(CC) $(CFLAGS) -c hello_world_2.c
```

**hello_world_2.o depend hello_world_2.c**

If  hello_world_2.c alive and **be update**,

it will do command ($(CC) $(CFLAGS) -c hello_world_2.c),

then output hello_world_2.o object file

# Rule Syntax Sample 2

```
hello_world_2.o: hello_world_2.c
	$(CC) $(CFLAGS) -c hello_world_2.c
```

**hello_world.o depend hello_world.c**

If hello_world.c alive and **be update**,

it will do command ($(CC) $(CFLAGS) -c hello_world.c),

then output hello_world.o object file.

# Rule Syntax Sample 2

```
hello_world_2.o: hello_world_2.c
	$(CC) $(CFLAGS) -c hello_world_2.c
```

**hello_world depend hello_world.o and hello_world_2.o**

If hello_world.o and hello_world_2.o alive and **be update**,

it will do command ($(CC) -o hello_world hello_world.o hello_world_2.o),

then create hello_world execute file

# hello_world_ex1

```
CC=$(CROSS_COMPILE)gcc

AA ='1234' '5678'
AA := 'DDDD'

$(info AA=$(AA))

CFLAGS=-Wl,-Map,out.map -lpthread -lm

all: hello_world

hello_world: hello_world.o
	$(CC) -o hello_world hello_world.o

hello_world.o: hello_world.c
	$(CC) $(CFLAGS) -c hello_world.c

clean:
	rm -r *.o
	rm hello_world
```

: Tab

# Assignment Operators

| | |
|---|---|
| = | 定義一個 需做遞迴展開的 變數型態 |
| := | 定義一個 立即運作的 變數型態 |
| += | 將 指定值， 續加在 原變數中 |
| ?= | 如果之前 無任何設定該變數， 即現在設定，<br>否則 跳過設定 ( 就是不做任何事 ) |

# Assignment Operators Sample 1

```
AA ='1234' '5678'
BB = ${AA}
AA = '789'
AA += 'ABCDE'
```

**Output**

```
AA='789' 'ABCDE'
BB='789' 'ABCDE'
```

# Assignment Operators Sample 2

```
AA ='1234' '5678'
BB := ${AA}
AA = '789'
AA += 'ABCDE'
```

**Output**

```
AA='789' 'ABCDE'
BB='1234' '5678'
```

# Assignment Operators Sample 3

```
AA ='1234' '5678'
BB := ${AA}
AA = '789'
AA ?= 'ABCDE'
```

**Output**

```
AA='789'
BB='1234' '5678'
```

# The Automatic Variables

- $@ The target filename.
- $< The first prerequisite.
- $^ The list of prerequisites, excluding duplicate elements.

# The Automatic Variables

```
CC = gcc
CFLAGS = -Wall -g -std=c99
LDFLAGS = -lm

circle : circle.o circulararea.o
        $(CC) $(LDFLAGS) -o circle circle.o
circulararea.o

circle.o : circle.c
        $(CC) $(CFLAGS) -o circle.o -c circle.c

circulararea.o: circulararea.c
        $(CC) $(CFLAGS) -o circulararea.o -c
circulararea.c
```

# The Automatic Variables

```
CC = gcc
CFLAGS = -Wall -g -std=c99
LDFLAGS = -lm

circle : circle.o circulararea.o
        $(CC) $(LDFLAGS) -o $@ $^

circle.o : circle.c
        $(CC) $(CFLAGS) -o $@ -c $<

circulararea.o: circulararea.c
        $(CC) $(CFLAGS) -o $@ -c $<
```

# Phony Targets

- .PHONY
  - Any targets that are prerequisites of .PHONY are always treated as out of date.

```
#Naming our phony targets
.PHONY: clean install

#Removing the executable and the object files
clean:
    rm sample main.o example.o
    echo clean: make complete

#Installing the final product
install:
    cp sample /usr/local
    echo install: make complete
```

# Command-Line Options

- -C dir, --directory= dir
  - make changes the current working directory to dir before it does anything else. If the command line includes multiple -C options, each directory specified builds on the previous one
- -j [ number] , --jobs[= number]
  - Run multiple commands in parallel

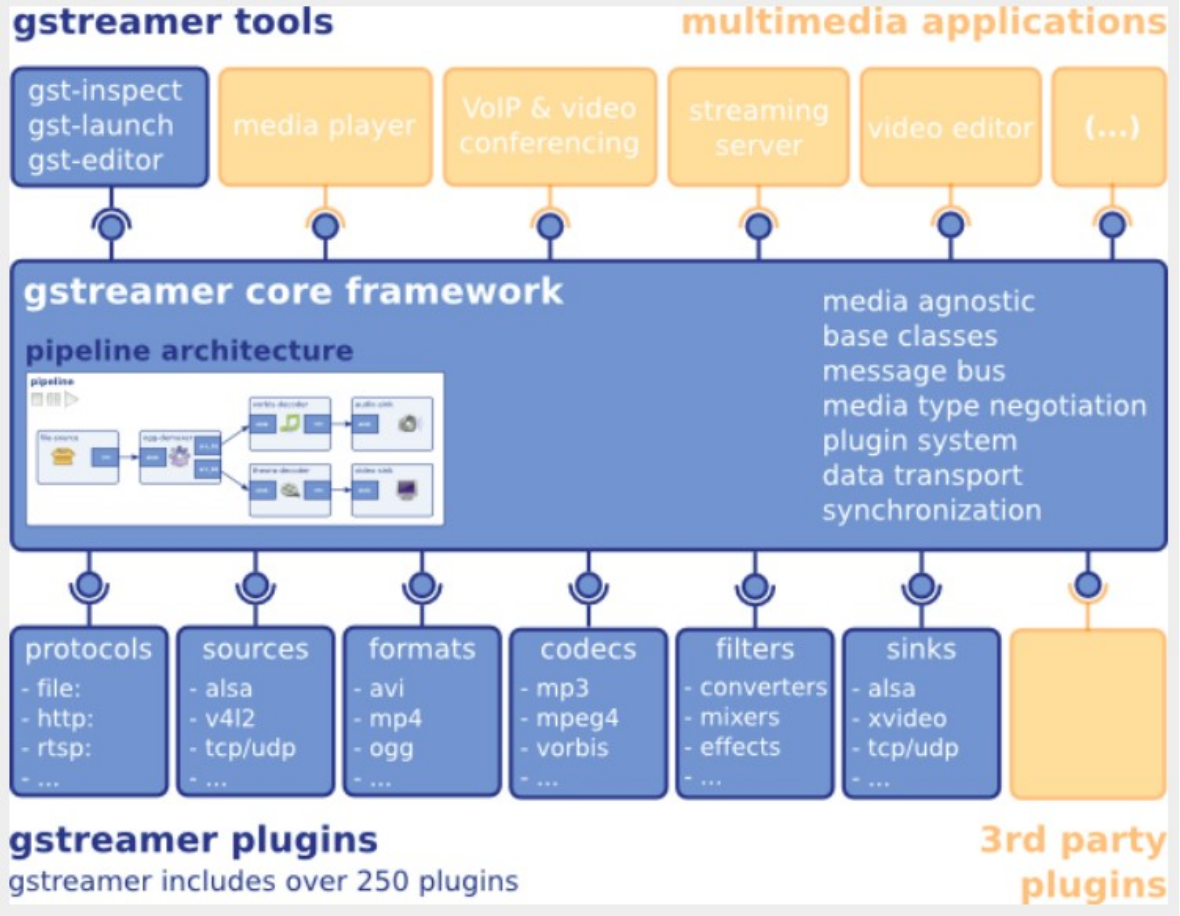
# Media Tool

# Gstreamer

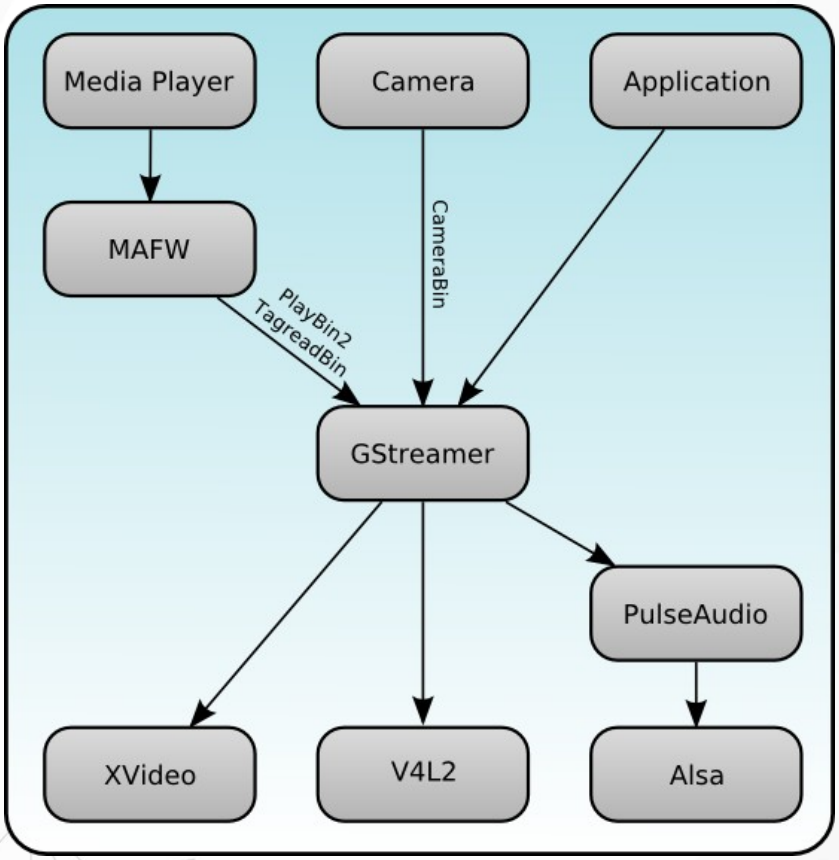
# Open Source Multimedia Framework

- https://gstreamer.freedesktop.org/
- Documentation
  - Tutorials

# Block Diagram



# Block Diagram




http://maemo.org/development/sdks/maemo_5_beta_docs/using_multimedia_components/

# Overview

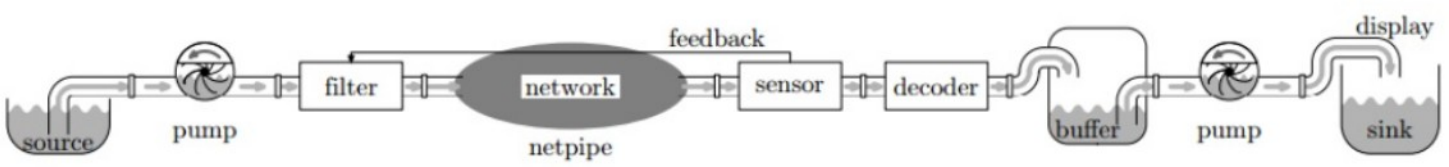
- GStreamer is a **framework** for creating streaming media applications
- The framework is based on **plugins** that will provide the various codec and other functionality

# Overview

- **Gst-plugins-base:**
  an essential exemplary set of elements
- **Gst-plugins-good:**
  a set of good-quality plug-ins under LGPL
- **Gst-plugins-ugly:**
  a set of good-quality plug-ins that might pose distribution problems
- **Gst-plugins-bad:**
  a set of plug-ins that need more quality

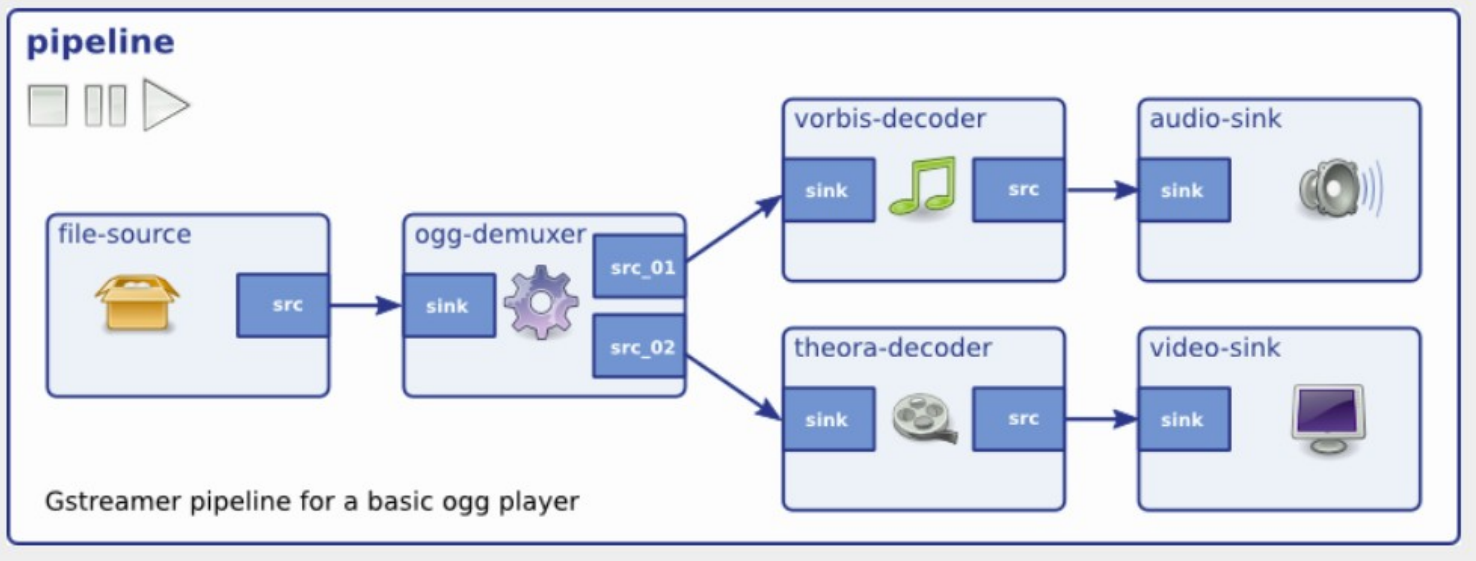
# Gstreamer Pipe

A streamer pipe




https://www.slideshare.net/rodrigo_costa/gstreamer-an-overview

# Gstreamer Pipe



# Gstreamer - Tool

- Gst-inspect-1.0
  - Print supported plug-in
- Gst-launch-1.0
  - Gstreamer tester
- Gst-typefind-1.0
  - Check file for gstreamer plug-in type

# Gstreamer - Tool

- Gst-inspect-1.0
  - Check what kind of videosink in Rockpi4b
  - gst-inspect-1.0 | grep -i videosink

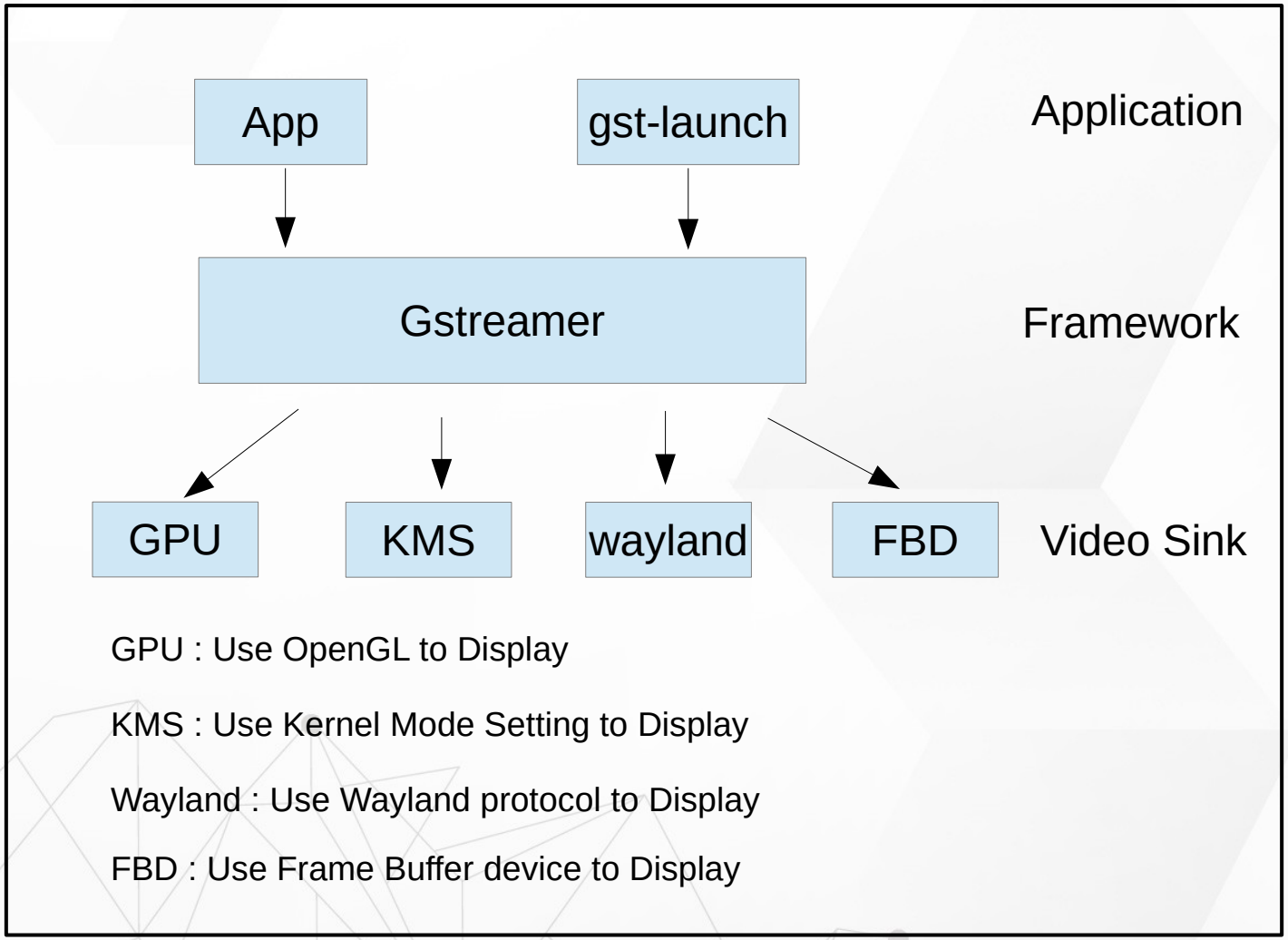
```
rock@rockpi4b:~$ gst-inspect-1.0 | grep -i videosink
debugutilsbad:  fakevideosink: Fake Video Sink
debugutilsbad:  fpsdisplaysink: Measure and show framerate on videosink
inter:  intervideosink: Internal video sink
decklink:  decklinkvideosink: Decklink Video Sink
autodetect:  autovideosink: Auto video sink
rock@rockpi4b:~$
```

# Gstreamer - Tool

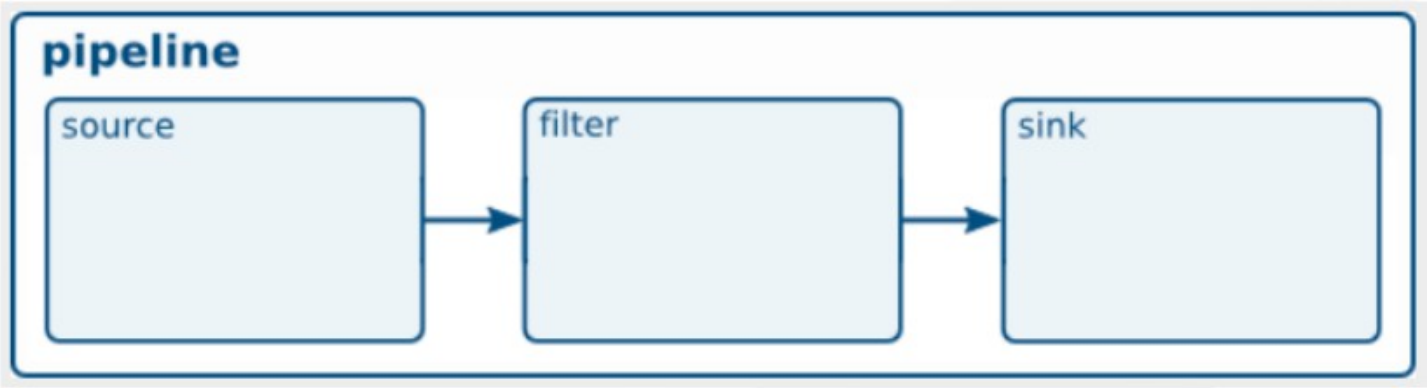
- Gst-typefind-1.0
  - Check Serenity-DVD-320x240.m4v what kind of file type in gstreamer
  - gst-typefind-1.0 ./Serenity-DVD-320x240.m4v

```
rock@rockpi4b:~$ gst-typefind-1.0 ./Serenity-DVD-320x240.m4v
./Serenity-DVD-320x240.m4v - video/quicktime, variant=(string)iso
```

# Gstereamer - Video



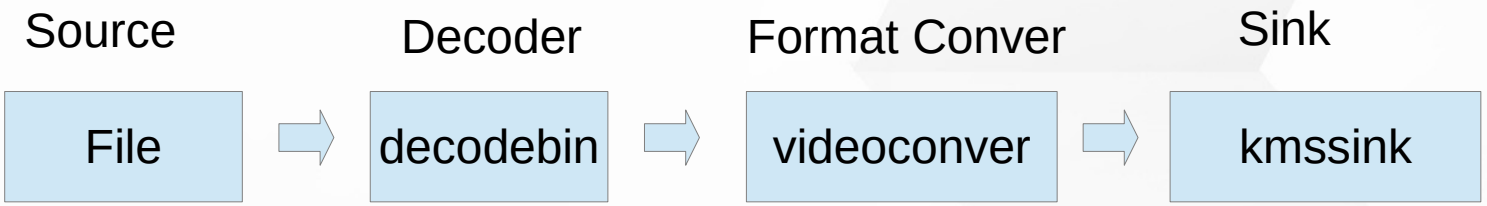
# Play Video Test Stream




```
gst-launch-1.0 videotestsrc ! video/x-raw, width=1280, height=720 ! kmssink
```

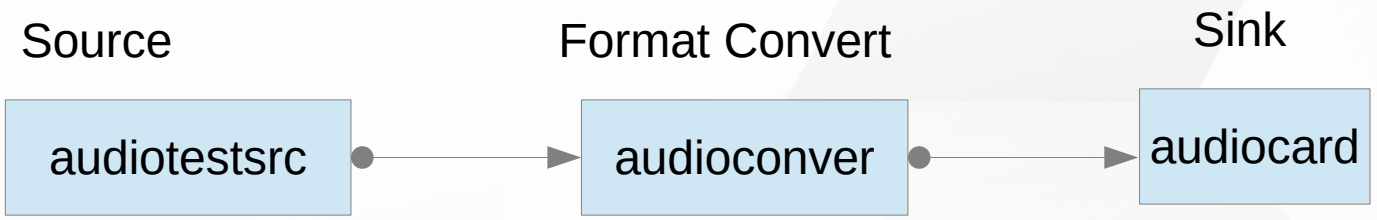
# Play a H.264 video

```
gst-launch-1.0 filesrc
location=/oem/200frames_count.h264 ! \
decodebin name=dec ! \
videoconvert ! \
kmssink
```



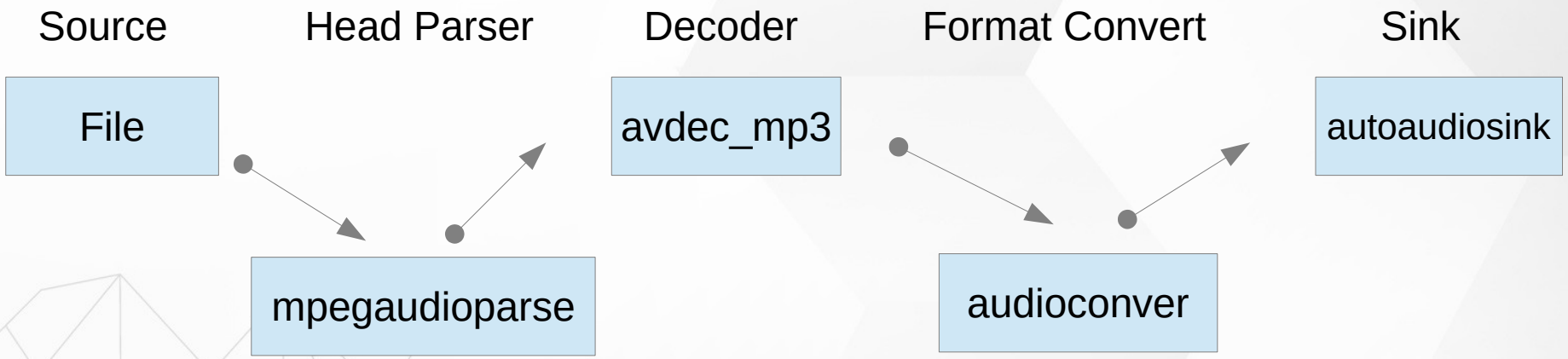
# Play a Audio Test

```
gst-launch-1.0 audiotestsrc ! audioconvert ! alsasink device-name=rockchipes8316c
```



# Play a MP3

```
gst-launch-1.0 filesrc location="/home/cadtc/audio/piano2-CoolEdit.mp3" ! \
mpegaudioparse ! \
mpg123audiodec ! \
audioconvert ! autoaudiosink
```



# ALSA Tool

- ALSA Utile
  - **aplay**
    - Play a WAV file
  - **arecord**
    - Record a sound
  - **alsamixer**
    - A graph tool for adjusting audio gain
  - **amixer**
    - A console tool for adjusting audio gain

# ALSA Tool

## ALSA Utile

- **aplay**
  - aplay -Dsysdefault:CARD=rockchipes8316c /usr/share/sounds/alsa/Front_Center.wav
  - aplay -Dsysdefault:CARD=HDMICODEC /usr/share/sounds/alsa/Front_Center.wav
- **arecord**
  - arecord -Dhw:0,0 -r 44100 -t wav -f CD -d 5 /tmp/test.wav
- **alsamixer**
  - alasmixer
- **amixer**
  - amixer scontrols | less
  - amixer sget 'HP' 0%
  - amixer sset 'HP' 0%

# WiFi and Network

# Basic Network Tool

- ifconfig → Network setting check
- ping → Network package check
- iperf3 → perform network throughput tests
- dhcpc → used for automatic retrieving of

# WPA/WPA2

- iw → Finding the WiFi device name
  - Scan SSID
- wpa_supplicant
  - For connecting to a WPA/WPA2 network

# WPA/WPA2 - Device

$ iw dev

```
[root@rk3399:/]# iw dev
phy#0
        Interface wlan0
                ifindex 3
                wdev 0x1
                addr cc:4b:73:92:50:6a
                type managed
                txpower 31.00 dBm
```

$ ls /sys/class/net

```
[root@rk3399:/]# ls /sys/class/net/
eth0  lo  wlan0
[root@rk3399:/]#
```

# WPA/WPA2 - iw

$ iw wlan0 scan

```
BSS 0c:9d:92:d9:e7:78(on wlan0)
        TSF: 7656316992 usec (0d, 02:07:36)
        freq: 2462
        beacon interval: 100 TUs
        capability: ESS Privacy ShortPreamble ShortSlotTime RadioMeasure (0x1431)
        signal: -73.00 dBm
        last seen: 2 ms ago
        SSID: kevin asus
        Supported rates: 1.0* 2.0* 5.5* 11.0*
        DS Parameter set: channel 11
        ERP: Use_Protection
        Extended supported rates: 6.0 9.0 12.0 18.0 24.0 36.0 48.0 54.0
        RSN:     * Version: 1
                 * Group cipher: CCMP
                 * Pairwise ciphers: CCMP
                 * Authentication suites: PSK
                 * Capabilities: 16-PTKSA-RC 1-GTKSA-RC (0x000c)
        HT capabilities:
                Capabilities: 0x12d
                        RX LDPC
                        HT20
                        SM Power Save disabled
                        RX HT20 SGI
                        RX STBC 1-stream
                        Max AMSDU length: 3839 bytes
                        No DSSS/CCK HT40
                Maximum RX AMPDU length 65535 bytes (exponent: 0x003)
                Minimum RX AMPDU time spacing: 2 usec (0x04)
                HT RX MCS rate indexes supported: 0-7
                HT TX MCS rate indexes are undefined
```

# WPA/WPA2 – SSID and PASSWD

$ wpa_passphrase “SSID” > /etc/wpa_supplicant.conf

```
ctrl_interface=/var/run/wpa_supplicant
update_config=1
ap_scan=1

network={
	ssid="ssid"
	#psk="testtest"
	psk="password"
}
```

# WPA/WPA2 - Connect

$ wpa_supplicant -B -D wext -i wlan0 -c /etc/wpa_supplicant.conf

```
[   29.752634] CFG80211-ERROR) wl_escan_handler : escan is not ready ndev ffffffc0782d1000
[   29.767372] wl_iw_set_essid: WLC_DISASSOC
[   29.772806] Setting the D11auth 1
[   29.788792] wl_iw_set_freq: chan=4
[   29.794068] wl_iw_set_wap: WLC_REASSOC failed (-22).
[   29.835315] Connecting with 62:07:b7:ed:02:4d channel (4) ssid "REASSO", len (6)
[   29.835315]
[   29.908754] wl_iw_event: Link UP with 62:07:b7:ed:02:4d
[   29.914341] wl_bss_connect_done succeeded with 62:07:b7:ed:02:4d
[   29.921748] wl_bss_connect_done succeeded with 62:07:b7:ed:02:4d
```

# WPA/WPA2 - DHCP

$ udhcpc -i wlan0

```
[root@rk3399:/]# udhcpc -i wlan0
udhcpc: started, v1.27.2
udhcpc: sending discover
udhcpc: sending select for 192.168.43.214
udhcpc: lease of 192.168.43.214 obtained, lease time 3599
deleting routers
adding dns 192.168.43.12
[root@rk3399:/]#
```

# WPA/WPA2 - IP

$ ifconfig wlan0

```
[root@rk3399:/]# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr CC:4B:73:92:50:6A
          inet addr:192.168.43.214  Bcast:192.168.43.255  Mask:255.255.255.0
          inet6 addr: fe80::7e7:9ca:dc48:71ab/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:40 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1312 (1.2 KiB)  TX bytes:4477 (4.3 KiB)

[root@rk3399:/]#
```

# WPA/WPA2 - Ping

$ Ping 8.8.8.8 (Google)

```
[root@rk3399:/]# ping -I wlan0 8.8.8.8
PING 8.8.8.8 (8.8.8.8) from 192.168.43.214 wlan0: 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=49.3 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=44.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=37.5 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=117 time=35.3 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=117 time=63.8 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=117 time=29.5 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=117 time=40.3 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=117 time=45.8 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=117 time=53.0 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=117 time=35.6 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=117 time=33.8 ms
^C
--- 8.8.8.8 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10018ms
rtt min/avg/max/mdev = 29.590/42.629/63.804/9.521 ms
[root@rk3399:/]#
```

# SSH

# SSH

- Secure SHell protocol
- SSH Client
- SSH Server

# SSH

- SSH Client
  - # sudo apt-get install ssh
  - https://slashembeddedlinux.blogspot.com/p/tmp.html

# NFS

# NFS

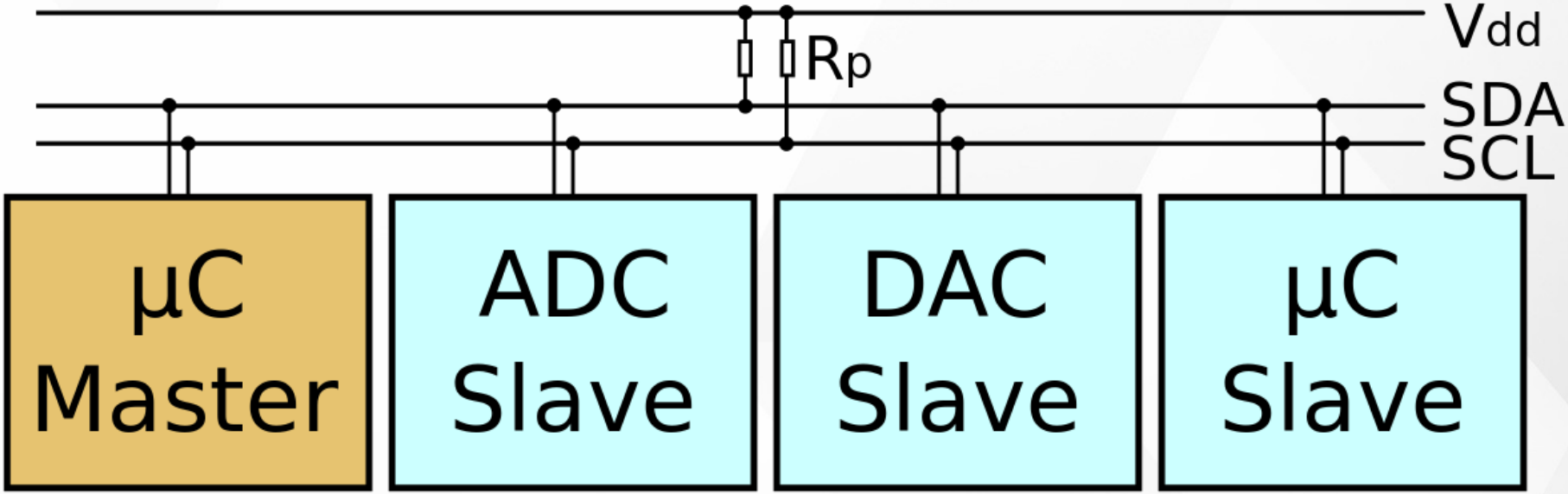
- Network File System
- NFS Client
- NFS Server

# NFS

- https://slashembeddedlinux.blogspot.com/p/tmp.html

# I2C Tool

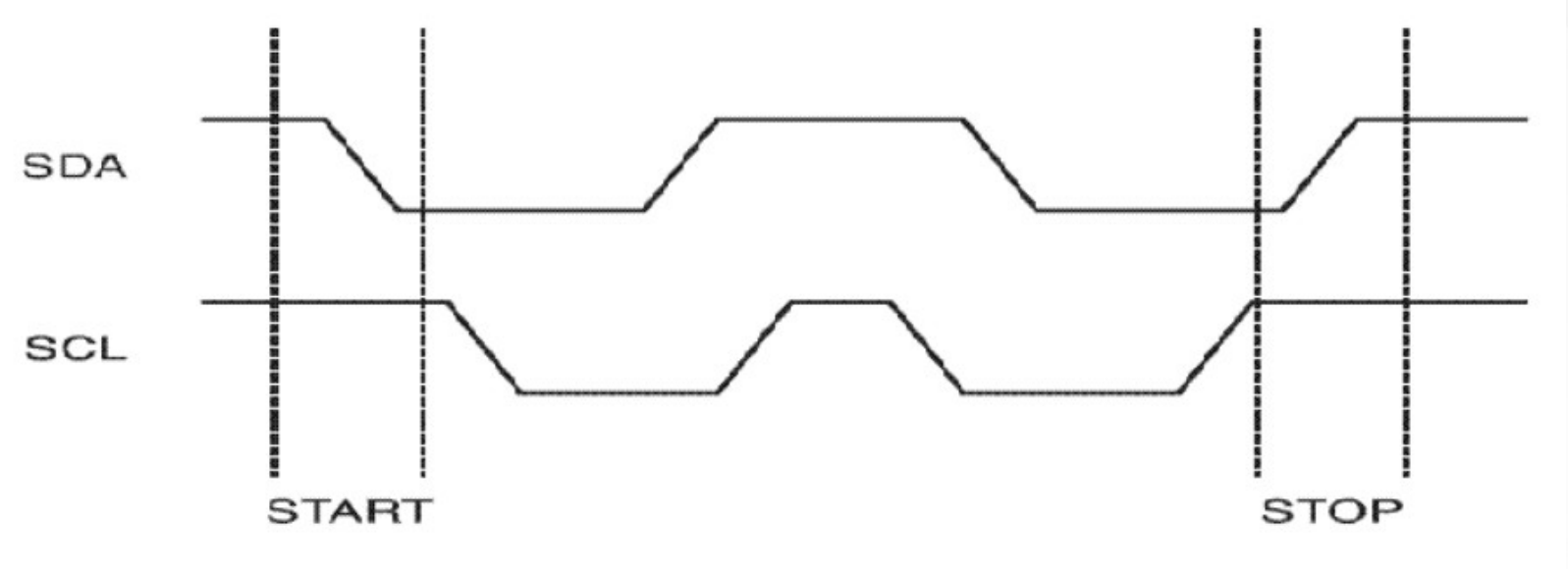
# I2C Driver




https://zh.wikipedia.org/wiki/I%C2%B2C

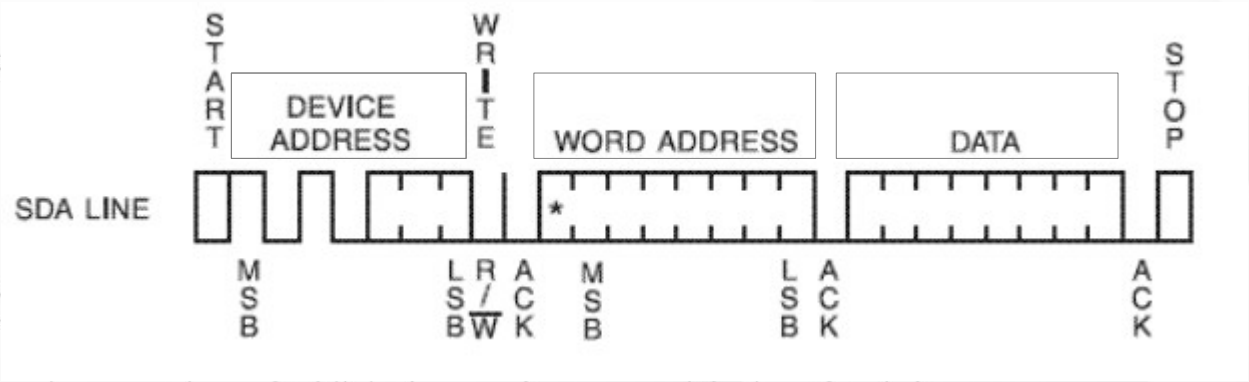
# I2C protocol

- Serial bus
  - SDA data line
  - SCL clock line



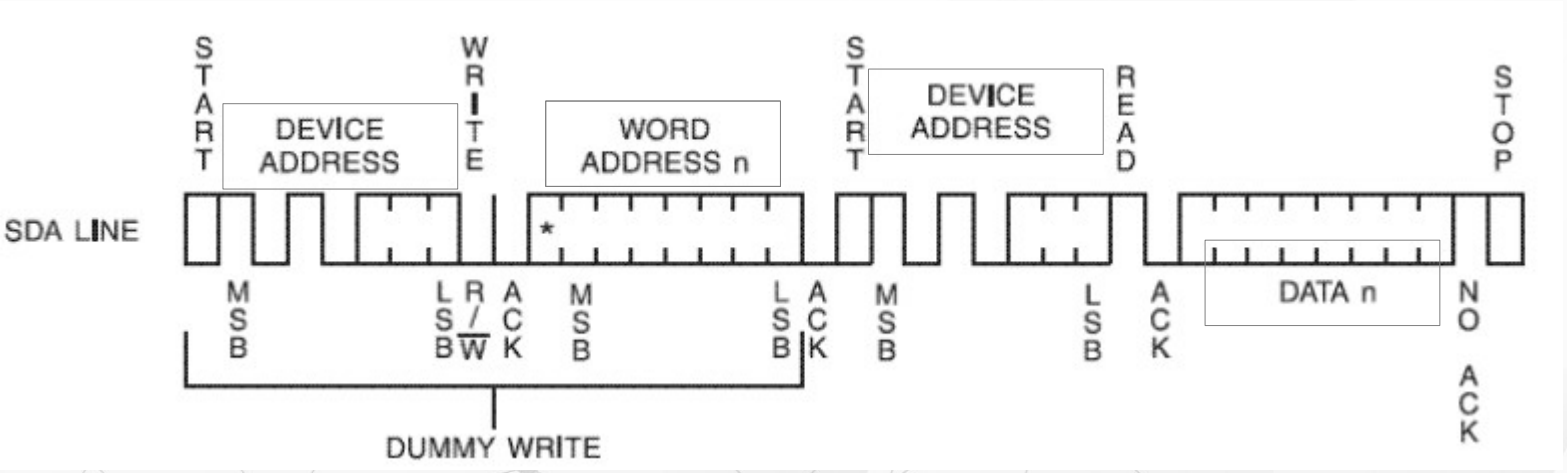
# I2C protocol

- Write
  - byte write
  - page write
- Device address
- Read/write bit : 0
- ACK



# I2C protocol

- Read
  - byte read
  - page read
- Device address
- **Read/write bit : 1**
- ACK



# I2C Dev Interface

- i2c tool
  - i2cset, i2cget, i2cdump
  - i2cdetect -l
- /dev/i2c-x
  - /dev/i2c-0, /dev/i2c-1, /dev/i2c-2 ...
- /sys/class/i2c-dev/
  - i2c-0 i2c-1 i2c-2 i2c-3 i2c-7 i2c-8 ...

# I2C Dev Interface

- Documentation/i2c/dev-interface
- i2c-tools
  - i2cdump
  - i2cdetect
  - i2cget
  - i2cset