

這份程式碼是一部份敲擊棒的OS，因為OS本身實在太大了，所以我只有上傳一部份。主要有更動的地方是 hda1 底下的 modules 以及模型轉換器。

Pineapple VM 開發環境建置

直接載入環境

我把平常用來開發的 VM 打包放在 3090 server 的 '/bowen/vm/bw_vm' 底下，順便放VMWare pro 的安裝檔，如果能直接使用的話就不用安裝環境

載入OVA檔

- OVA 檔包含整包韌體開發工具所需的環境以及套件，所有開發都會在上面執行
 - 去 /bowen 下載 ova 檔
logseq.order-list-type:: number
 - 下載 VMWare workstation 17
logseq.order-list-type:: number
 - 使用 VMWare 開啟 ova 檔
logseq.order-list-type:: number
 - 記得記憶體跟核心要切夠大

下載 NT9852x SDK & NvTAI converter

- NT9852x SDK 是 AI 晶片所使用的 OS 跟 module driver 的 package，NvTAI converter 則是將訓練好的 onnx 檔轉成 AI 晶片能用的模型檔。
 - 下載 FileZilia
logseq.order-list-type:: number
 - 輸入基本資料
logseq.order-list-type:: number
 - 主機: 61.219.106.83
logseq.order-list-type:: number
 - 帳號: Birckcom-ivot
logseq.order-list-type:: number
 - 密碼: nqEtRZ29
logseq.order-list-type:: number
 - 連接埠: 21
logseq.order-list-type:: number

- 下載需要的工具 (通常是整包載下來)

logseq.order-list-type:: number

- nt9852x_linux_ltd_release_v2.05.009

logseq.order-list-type:: number

- NvtAI_Tool_release

logseq.order-list-type:: number

基本 ubuntu 環境設定

- 開啟 VM 後的前置安裝

- 先登入，密碼: ubuntu

logseq.order-list-type:: number

- 先去 Settings → Displays 調整大小至1600 * 1200

logseq.order-list-type:: number

- VM 開啟前的網路設定從 Bridged 改為 Net

logseq.order-list-type:: number

- sudo apt upgrade

logseq.order-list-type:: number

- sudo apt *install* exfat-fuse exfat-util (安裝通用 usb driver)

logseq.order-list-type:: number

建置 NvTAI_Converter 環境

- NvTAI_Converter 是一個專門編譯模型的套件環境，將模型轉換成晶片能讀取的模型，其被包成一個 docker image，以利在任何地方都能轉換。
- 安裝與使用的詳細介紹請看 'Novaic_Converter_User_Guide_zh.pdf' 這份文件。
- Note: Docker image 只是開發環境，工具包要另外安裝進去

• 安裝 Docker

- 先卸除舊的 Docker 版本(舊版本的名稱為 docker, [docker.io](https://www.docker.io/) 或 docker-engine)

logseq.order-list-type:: number

```
$ sudo apt-get remove docker docker-engine docker.io containerd runc
```

- 先安裝會使用到的 package

logseq.order-list-type:: number

```
$ sudo apt-get install apt-transport-https ca-certificates curl gnupg lsb-rele
```

- 新增 Docker's official GPG key

logseq.order-list-type:: number

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor  
$ echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.g
```

- 更新 docker 內部套件

logseq.order-list-type:: number

```
$ sudo apt upgrade
```

- 安裝 Docker

logseq.order-list-type:: number

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

- 確認 Docker Engine 是否運作

logseq.order-list-type:: number

```
$ systemctl status docker
```

- 若 Docker Engine 沒有運作，執行下列指令開啟

logseq.order-list-type:: number

```
$ systemctl start docker.service
```

- 驗證 Docker 是否被正確的安裝，此指令會從 DockerHub 下載一個測試 image 並跑起一個 container，當 container 跑起來後，會打印出一些訊息後再離開

logseq.order-list-type:: number

```
$ sudo docker run hello-world
```

載入 docker image

- 解壓縮 Docker image (.tar.bz2 -> .tar)

logseq.order-list-type:: number

- 載入 Docker image

logseq.order-list-type:: number

- 確認目前已載入的 Docker image

logseq.order-list-type:: number

- 先創好 shared 資料夾，然後將該資料夾跟 container 的 /home 連接

logseq.order-list-type:: number

- 使用 Docker image ubuntu16p04:v00.07.2108260 創建一個新的 container，並把 < logseq.order-list-type:: number
host_shared_folder_path > 與 <docker_shared_folder_path> 視為同個資料夾，之後可將工具的安裝包透過 < host_shared_folder_path > 放到 docker 中，並執行 docker
- 確認目前的系統上的 container 狀態，在跑完上面那行指令後你應該在 docker 裡
logseq.order-list-type:: number
- 離開目前的 container
logseq.order-list-type:: number

- ```
1
$ tar -xf archive.tar.bz2
2
$ sudo docker load -i ubuntu16p04_v00_07_2108260.tar
3
$ sudo docker ps -a
$ sudo docker image ls
5
$ sudo docker run -it -v /home/ubuntu/Desktop/shared:/home ubuntu16p04:v00.07.
7
[docker]# exit
```

## 模型編譯工具包安裝說明

- 將工具的安裝包透過 < host\_shared\_folder\_path > 放到 docker
- 解壓工具包

```
[docker] $ tar -zxvf release.tar.gz
```

請參考解壓縮後的說明書 'Novaic\_Converter\_User\_Guide\_zh.pdf'，解壓後主要目錄結構大致如圖所示，其中主要程式 compiler.nvtai、simulator.nvtai 會放在以下位置 (closeprefix/bin/)，內建的 example 會放在 test-tutorial 中。

## 更新 Docker python 版本

原本 python3.6 的版本不知為何不能動，須更新

```
Download python 3.7.4 package and put it to shared folder
Go into python3.7.4 folder
[docker] $ make
[docker] $ make install
[docker] $ pip3.7 install --upgrade pip
[docker] $ pip3.7 install --upgrade onnx
[docker] $ pip3.7 install --upgrade onnxsim
```

- Note: 你如果不想更新的話，Filezilla 那邊還有 ubuntu20 的版本，把他也載入就可以直接用 onnx2novaonnx 的 tool，轉換模型的 toolchain 還是要用 ubuntu16，不過開兩個視窗比較方便。另外如果要載入 ubuntu20，記得要把空出 40G 的空間(把一些VM的檔案清掉)。

## 部屬模型完整流程

---

### 產生 .onnx 檔

---

- 目前的模型產生的維度為  $1 * 1 * 18 * 513$  (B \* C \* H \* W)
- 在跑完 train\_3\_type.ipynb 這個 script 後，在 ../iPhone/pb/ 底下會產生模型檔
- 請使用 tf2onnx 這個開源工具將原本的模型轉換成 onnx 格式，在這邊請使用 python310 這個環境做轉換 (不確定 seedenv 能不能用)
- `python -m tf2onnx.convert --saved-model ../iPhone/pb/20240530_cup_v0.0.1/ --output ./cup.onnx --opset 12 --inputs 'input_2:0[1,18,513]'`
- 轉換完後可以去 netron.app 這個可視化工具看看效果如何

### 轉換 .onnx 檔至板子讀取的 .bin 檔

---

- 先將要轉換的原始 .onnx 檔放入以下路徑中  
`/home/0609_Novaic_v01.09.2306072_release_exe/Tool/ai_tool/novatek/novaic/toolchain/closeprefix/bin/compiler/frontend/onnx-onnx`  
`logseq.order-list-type:: number`
- 輸入以下指令轉換 onnx 檔至 novaonnx 檔  
`logseq.order-list-type:: number`
- 進入 `/home/0609_Novaic_v01.09.2306072_release_exe/Tool/ai_tool/novatek/novaic/test-tutorial/nvtai_tool/config/00007_cup/cnn25` 中修改 `gen_config.txt`  
`logseq.order-list-type:: number`
- 將轉好的 .novaonnx 檔放入 `[path/model_dir]` 中  
`logseq.order-list-type:: number`
- 進入  
`/home/0609_Novaic_v01.09.2306072_release_exe/Tool/ai_tool/novatek/novaic/toolchain`  
`logseq.order-list-type:: number`
- 輸入以下指令轉換 novaonnx 檔至 binary 檔  
`logseq.order-list-type:: number`

```
2
python3 (or python3.7) onnx2novaonnx_converter.py --input path/to/input_model
6
closeprefix/bin/compiler.nvtai --config-dir /home/0609_Novaic_v01.09.2306072_r
```

- 

- # 可能需要修改的 config (gen\_config.txt, 以玻璃杯模型為範例)  
# 要轉換的模型路徑  
[path/model\_dir] = ..\nvtai\_tool\input\model\customer\cup  
# 輸入平均值檔案 (用不到, 但請放在你的模型資料夾裡, 並且值要一樣)  
[path/mean\_path] = ..\nvtai\_tool\input\model\customer\cup\mean\_data.txt  
  
# 轉換模型用校正資料  
# 參考資料資料夾路徑  
[path/ref\_img\_dir] = ..\nvtai\_tool\input\data\cupDataset\glass  
# 參考用資料圖片名稱  
[path/ref\_list\_path] = ..\nvtai\_tool\input\data\cupDataset\ref\_img\_list.txt  
# 參考資料數量  
[ref\_data/num] = 22  
  
# 模型輸入資料類別(應為 float)  
# 雖然為 float, 但實際運行時仍是輸入 16-bit, 只是 16-bit 都是小數點位  
[preproc/in/type] = 6  
# 輸入格式, FMT\_FEAT代表隨意  
[preproc/in/fmt] = 3  
# 輸出格式  
[preproc/out\_fmt] = 4  
  
# 轉換模型的四個維度, 在轉換時你會注意到這邊的維度大小不太一樣  
# 原因是 tensorflow 跟轉換工具的表現方式不太一樣  
[preproc/in/width] = 513  
[preproc/in/height] = 1  
[preproc/in/channel] = 18  
[preproc/in/batch] = 1  
[preproc/in/time] = 1  
[preproc/resize/width] = 513  
[preproc/resize/height] = 1

## 檢測 .bin 檔與原始的 .novaonnx 檔的誤差

- 這個工具是模擬工具, 用來測量轉換出來的 binary 檔跟原本 novaonnx 檔案的誤差
- 進入 /home/0609\_Novaic\_v01.09.2306072\_release\_exe/Tool/ai\_tool/novatek/novaic/test-tutorial/nvtai\_tool/config/00007\_cup/cnn25 中修改 sim\_config.txt  
logseq.order-list-type:: number
- 一樣是在  
logseq.order-list-type:: number

```
2
python3 (or python3.7) onnx2novaonnx_converter.py --input path/to/input_model
```

```
6
closeprefix/bin/simulator.nvtai --config-dir /home/0609_Novaic_v01.09.2306072_
```

- # 可能需要修改的 config (sim\_config.txt, 以玻璃杯模型為範例)  
# 還是要有的檔案, 裡面可以亂寫  
[path/label\_path] = ..\nvtai\_tool\data\cupDataset\label.txt  
# 要測試的模型路徑  
[path/model\_dir] = ..\nvtai\_tool\input\model\customer\cup  
# 要測試的檔案路徑  
[path/test\_img\_dir] = ..\nvtai\_tool\input\data\cupDataset\glass  
# 測試檔案的 Ground Truth (沒差, 隨便放就好)  
[path/test\_gt\_path] = ..\nvtai\_tool\input\data\fakeset\ground\_truth.txt  
# 測試的檔案名稱  
[path/test\_list\_path] = ..\nvtai\_tool\input\data\cupDataset\test\_img\_list.txt  
# 選取的資料集  
[dataset] = cup  
# 結果輸出路徑  
[path/out\_dir] = ..\nvtai\_tool\output\00007\_cup  
# 輸入大小  
[src/width] = 513  
[src/height] = 1  
[src/channel] = 18  
[src/batch] = 1  
[src/time] = 1

## 編譯與安裝 NT98528 的 FW (全新 SDK)

- 先至 /NT9852x\_SDK\_Package/na51055\_linux\_sdk 的路徑 (目前版本 v2.05.009)  
logseq.order-list-type:: number
- make clean
- lunch  
logseq.order-list-type:: number
  - Linux  
logseq.order-list-type:: number
  - cfg\_528\_IPCAM1\_RAMDISK\_EVB  
logseq.order-list-type:: number
  - toolchain: arm-ca9-linux-gnueabi-hf-6.5  
logseq.order-list-type:: number
- make linux config  
logseq.order-list-type:: number
  - Device Drivers -> USB support -> USB Gadget support -> USB Peripheral Controller  
-> FOTG210 = M

```
logseq.order-list-type:: number
```

- make all

```
logseq.order-list-type:: number
```

- 至 na51055\_linux\_sdk/output/packed/ 底下拿取 FW98528A.bin

```
logseq.order-list-type:: number
```

## 編譯與安裝 NT98528 的 FW (更改 module code)

---

- make hdal

```
logseq.order-list-type:: number
```

- make rootfs

```
logseq.order-list-type:: number
```

- make pack

```
logseq.order-list-type:: number
```

- 拿 FW98528.bin

```
logseq.order-list-type:: number
```

## 將 .bin 放入板子中

---

- 先用 mac 或其他工具格式化 SD 卡 (FAT32)

- 請依底下順序將檔案放入 SD 卡

- LD98528A.bin

- FW98528A.bin

- 將要讀取的模型依據 pine\_identify 設置的路徑放至 SD 卡上

- 修改 code/hdal/samples/pine\_identify 裡相關參數 (相關請看 module 底下的 README)

- 測試資料

## 連上板子並開始跑辨識模組

---

- 用 sudo screen /dev/ttyUSB0 115200 連至板子的 UART debugger port

```
logseq.order-list-type:: number
```

- insmod /lib/modules/4.19.91/hdal/kflow\_ai/kflow\_ai.ko

```
logseq.order-list-type:: number
```

- cd /mnt/sd

```
logseq.order-list-type:: number
```



- pine\_identify  
logseq.order-list-type:: number

## 測資產生

---

在板子上，一張頻譜會需要被轉成一個 .bin 檔以進行讀取

參考資料：

1. [https://blog.51cto.com/u\\_16175494/6793554](https://blog.51cto.com/u_16175494/6793554)
2. <https://blog.csdn.net/jizhidexiaoming/article/details/79943540>

## Linux OS 配置修改

---

### 新增開機指令至 os 上

---

- 進入 /na51055\_linux\_sdk/BSP/root-fs/rootfs/etc\_Model/etc\_IPCAM1\_EVB/init.d  
logseq.order-list-type:: number
- 至 S99\_Sysctl 新增想要安裝的模組或執行指令  
logseq.order-list-type:: number
- 重新編譯  
logseq.order-list-type:: number

### 開啟 USB 讀取功能

---

- make linux\_config  
logseq.order-list-type:: number
- 確認以下配置一樣  
logseq.order-list-type:: number
  - Device Drivers -> USB support -> Support for Host-side USB = m
  - Device Drivers -> USB support -> EHCI HCD (USB 2.0) support = m
  - Device Drivers -> USB support -> NVTIM EHCI support = y
  - Device Drivers -> USB support -> USB Mass Storage support = m
  - Device Drivers -> USB support -> USB Gadget Support = M
  - Device Drivers -> USB support -> USB Gadget Support -> USB Gadget Drivers = m
  - Device Drivers -> USB support -> USB Gadget Support -> Mass Storage Gadget = m
  - Device Drivers -> USB support -> USB Gadget Support -> USB Peripheral Controller -> FOTG210 = M

- 進入 NT98528 debugger 輸入以下指令(安裝以下模組)

logseq.order-list-type:: number

```
modprobe udc-core
modprobe fotg200-udc # (modprobe fotg210-udc for 528 case)
modprobe configfs
modprobe libcomposite
modprobe usb_f_mass_storage
modprobe g_mass_storage iSerialNumber=123456 file=/dev/sda1 stall=0 removable=
```

- 插入 USB2.0 or type-c 即可讀取

logseq.order-list-type:: number

## 更改記憶體配置

- 在 /pineapple-knocker/NT9852x\_SDK\_Package/nt9852x\_linux\_sdk\_release\_glibc\_v2.05.008/configs/Linux/cfg\_528\_IPCAM1\_RAMDISK\_EVB/ 底下
  - "nvt-na51055-mem-tbl.dtsi" 總 memory 的分區分配
  - "nvt-51055-storage-partition.dtsi" flash memory 的分區分配
  - cat /proc/meminfo Memtotal 指令可以看 linux memory
  - cat /proc/hdal/comm/info 指令可以看 hdal-memory
  - cat /proc/mtd 可以看到 nand 目前分區
  - 目前記憶體配置 (開機詳情可以看 NovaTek issue board)
    - nvt-na51055-mem-tbl.dtsi

```
hdal-memory{
 #address-cells = <1>;
 #size-cells = <1>;

 -- media {reg = <0x10000000 0x0D800000 0x40000000 0x20000000>}
 ++ media {reg = <0x10000000 0x0D800000>}
}

NVT_LINUX_SMP = "NVT_LINUX_SMP_OFF";
```

## 問題詢問

### 金鑽 email

<http://mail.brickcom.com/index.html>

帳號: [bowen.lee@brickcom.com](mailto:bowen.lee@brickcom.com)

密碼: brickcom@12345

## VPN 連線至麗台的伺服器

---

- 軟體下載請參考: <https://openvpn.net/client-connect-vpn-for-windows/>
- 安裝完畢後，將信件中的附件(\*.ovpn)拖拉到軟體內，再按連線就可以了。

## 遠端連線至麗台 VM

---

- VPN 登入後，請使用 Microsoft remote desktop，連線至10.40.41.226
- 帳號: bowen
- 密碼: bowen@brickcom

## NovaTek Jira 登入

---

- 當有跟 AI 開發板相關問題處理不了時，可以將問題發至 NovaTek 的 Jira 上，Jira 只能透過麗台的電腦登入。
- 帳號：blee\_brickcom
- 密碼：k[z9!6CFE
- Microsoft Authentication 兩階段登入

樹莓派ssh連線密碼 33564  
192.168.8.1