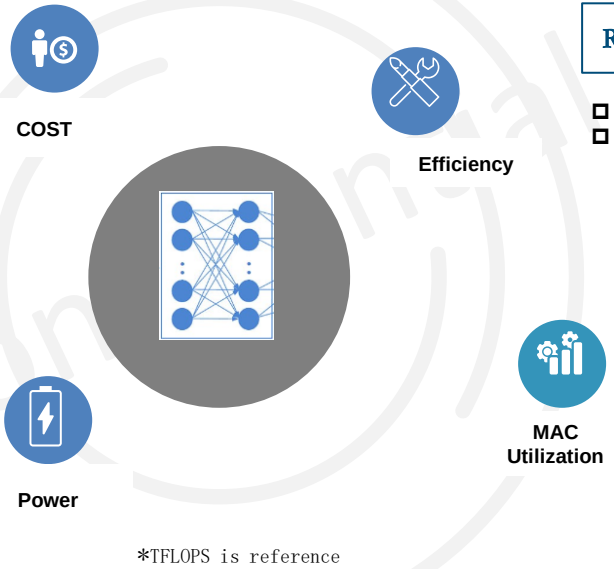
KL520 series AI SoC Training Materials

jeffrey-yc.chen@kneron.us

Kneron AI Chip Advantage

Kneron Advantage

- 28~40nm process
- SoC ready for use
- More less analog IP



*TFLOPS is reference

Reconfigurable Model update

- Model reconfigurable, support Block level
- OTA update

Power efficiency – Whole chip? NPU?

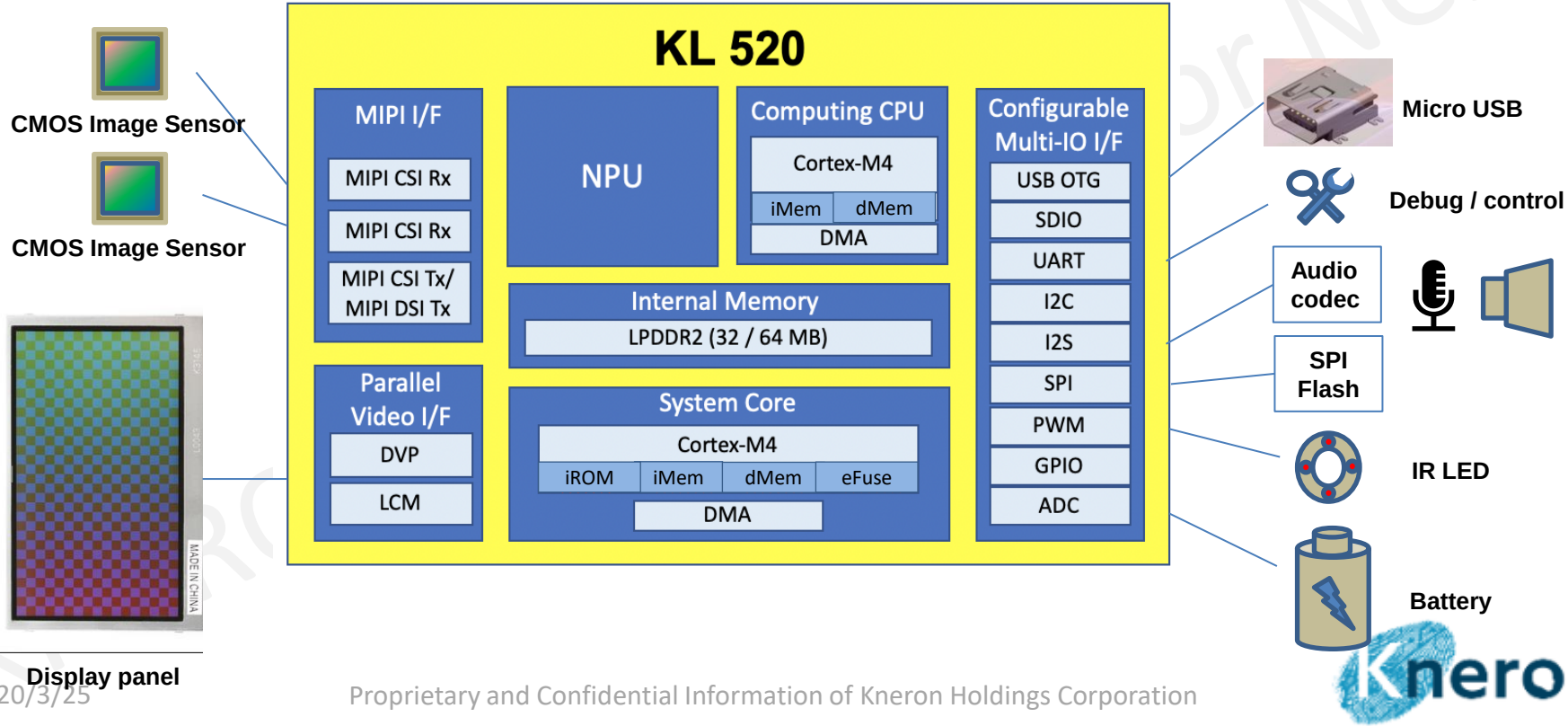
- Power efficiency is the key for edge device

MAC Utilization: cost and power (memory consumption)

- ResNet50 (~3x gain)
  - Kneron: 73%
  - Others: 23.16%
- GoogLeNet(~1.7x gain)
  - Kneron: 74%
  - Others: 43.19%

Development kit - HDK

Development board function diagram



Key Spec.

- NPU
  - Maximum Frequency @ 300 MHz
  - Peak Throughput of 8-bit mode: 345 GOPS, 576MAC/cycle
- CPU
  - ARM Cortex-M4@200MHz for system control
  - ARM Cortex-M4@250MHz as AI co-processor
- SDRAM
  - SIP, 32MB or 64MB, 16-bit LPDDR2-1066
- External flash
  - Up to 64 MB SPI NOR flash
- Power
  - Avg. power consumption 500mW
  - 1.1V core voltage
  - 3.3, 1.8V I/O voltage
- Process node
  - 40nm low power
- Video in interface
  - 2-lane MIPI-CSI-2 RX
  - DVP
- Video out interface
  - 2-lane MIPI-CSI-2 TX
  - MIPI-DSI TX
  - DVP
  - LCM
- Audio Interface
  - I2S interface for connecting to external audio codec
- Peripheral Interface
  - I2C
  - SPI
  - UART
  - USB 2.0 host/device interface
  - PWM
  - GPIO
  - SDIO
- Supporting OS
  - CMSIS RTX

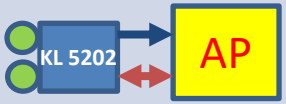
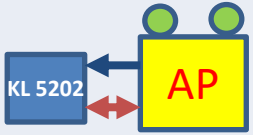
Packages

Variants	Package Type	Package Size	LPDDR2 SIP	Major difference	Product number
1	LQFP 128	14x14mm	512Mb	MIPI RX*2	KL52002-A0
2	TFBGA 159	8x8mm	512Mb	MIPI RX*2	KL52012-A0
3	TFBGA 161	8x8mm	512Mb	MIPI RX*1; MIPI TX*1	KL52022-A0
4	LQFP 128	14x14mm	256Mb	MIPI RX*2	KL52001-A0
5	TFBGA 159	8x8mm	256Mb	MIPI RX*2	KL52011-A0
6	TFBGA 161	8x8mm	256Mb	MIPI RX*1; MIPI TX*1	KL52021-A0

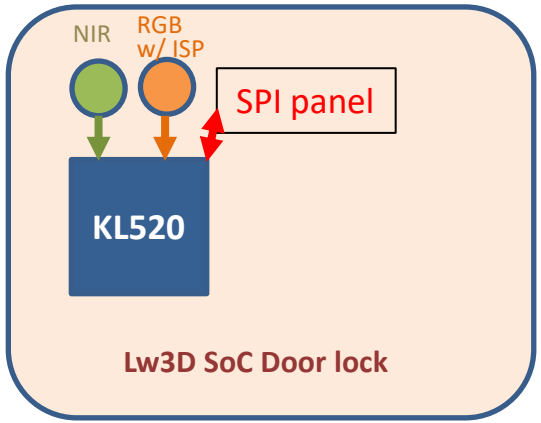
Peripheral count

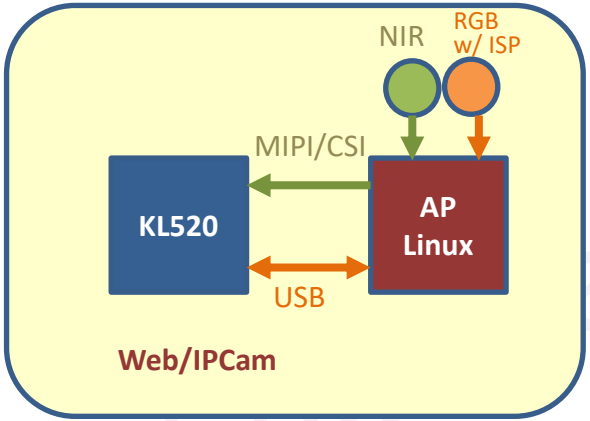
Package		LQFP 128		TFBGA 159		TFBGA 161	
Config. Mode (1)		Config 1	Config 2	Config 3	Config 4	Config 5	Config 6
MIPI inputs	MIPI CSI Rx	1	2	2	2	1	1
MIPI output	MIPI CSI Tx MIPI DSI Tx	N/A	N/A	N/A	N/A	CSI or DSI x1	
Camera Interface	DVP	N/A	N/A	0	0	0	1 (8 bit)
Display module interface	LCM (Intel 8080)	N/A	N/A	0	1 (18 bit)	0	0
USB OTG		1					
RTC		1					
ADC		N/A	N/A	4 channels			
Common I/O interface (2)	I2C	1	2	3	2	3	3
	I2S	0	0	1	1	1	1
	SPI	1	1	2	0	2	1
	UART	1	0	2	1	2	1
	SDIO	0	0	1	0	1	1
	GPIO	1	1	7	1	7	2
	PWM	1	1	2	1	2	1

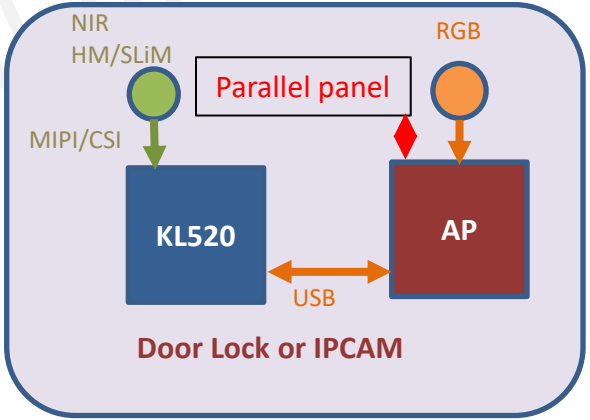
KL520 Applications

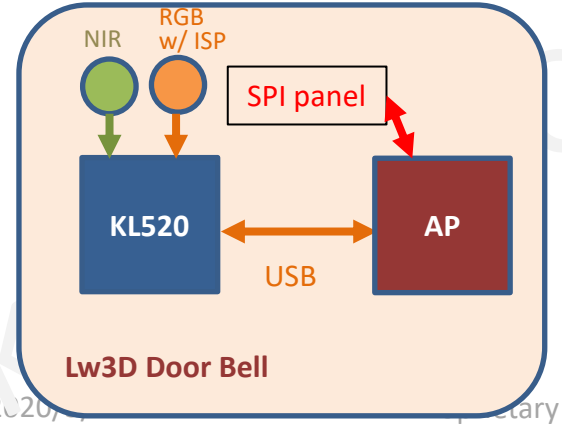
Type	示意圖	Application	S/W	Note
AI Camera		Lw3D 門鎖	FID + EID Image output + AP side lib. + OTA	
		Structure light 門鎖	FID + EID Image output + OTA	
		Structure light 門鎖	FID + EID OTA?	
AI Companion		Stereo web/IP cam	FID + EID Image input + AP side lib. + OTA	
		工業Addon 卡	USB command + AP side lib. + OTA	
AI SOC		Lw3D 門鎖 Structure light 門鎖	FID + EID Op flow + display + OTA	
		2D Platform for porting	FID + EID ; op flow; display Tool chain + compiler + OTA	

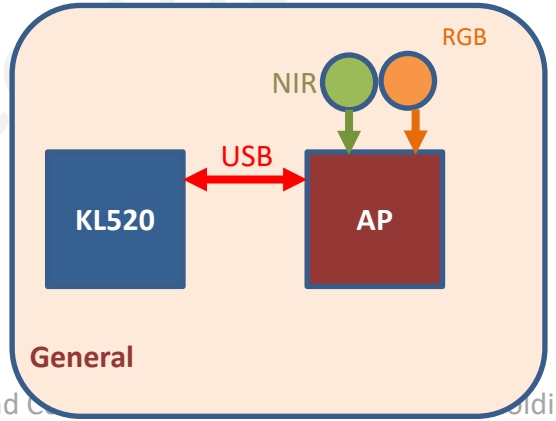
As an AI companion







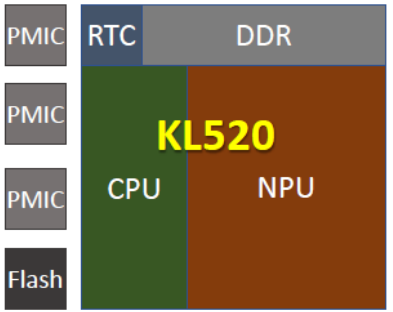




KL520 Power Consumption – Overview 2019.06.26

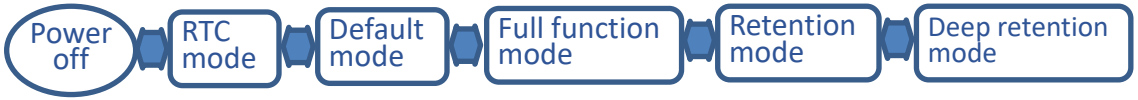
Power mode	RTC	Default	Full Function			Retention
Operation	Power On to Stand-by	OS Booting	Load Model	Load Image	FD/FR	1:N
Time	2 S	~55ms	~200ms	~20ms	~75ms	~20ms
Average Power consumption (25°C)	1uA/3uW	275mW	460mW	650mW (peak power on FD/FR: ~1.5W)		430mW
Note (companion mode)	Only 1st time	~100KB from Flash	~8MB from Flash	Load VGA from USB	VGG8/VGA	15 users (5faces/user)

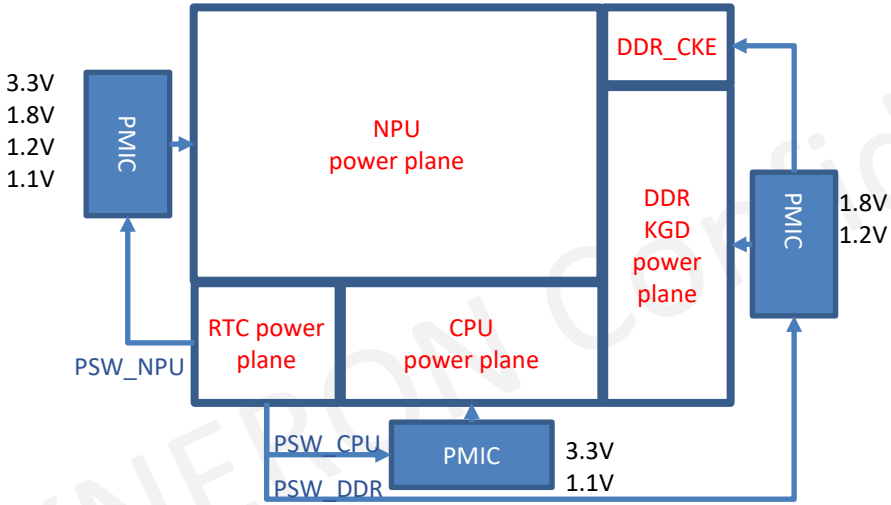
KL520 power mode

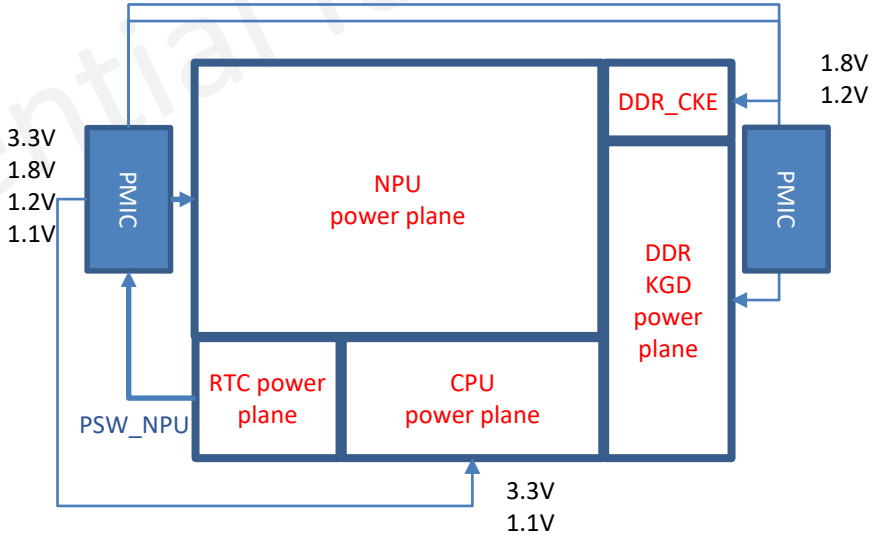


Kneron

Power Mode Operation







SDK function description

Items	Function description
Compiler tool	Compile Models into NPU instructions
model convertor	Convert models from different deep learning framework to ONNX
fixed-point analyzer	Analyze model data path to create best quatization models
Performance evaluator	Estimate model speed based on npu instructions
Simulator	Based on npu instruction, provide simulated result
Emulator	Run simulator with large batch of images in order to get better sense of accuracy
Keil MDK	CM4 CPU SDK from ARM
NPU library	Provide easy programming interface to control NPU
CMSIS RTX driver library	Provide easy programming interface to control Periphirals

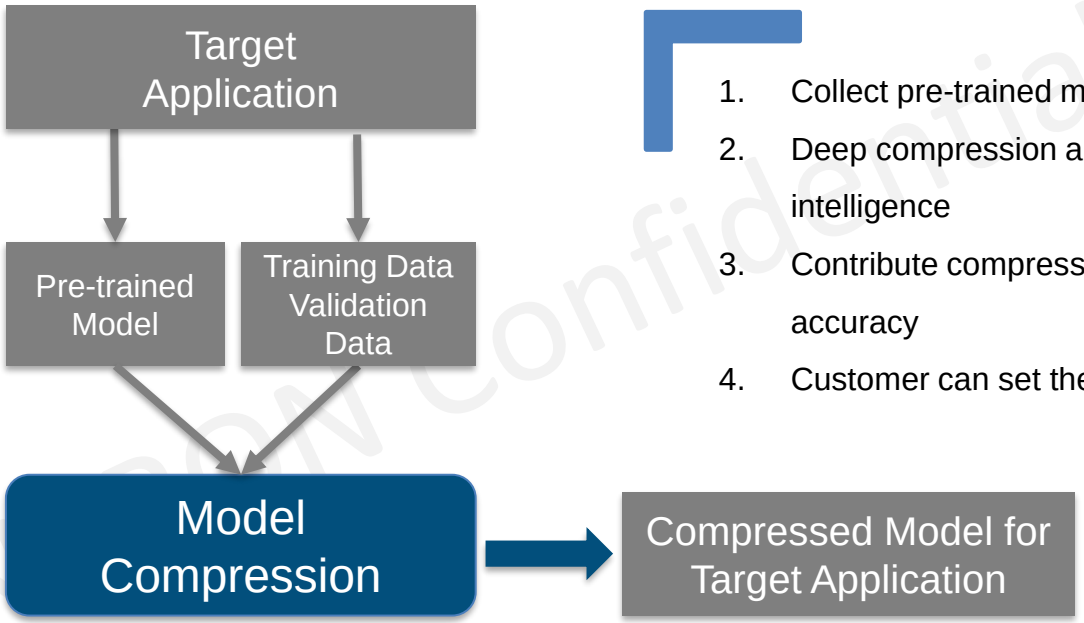
Development kit - SDK

Kneron Patented Quantization – No Retrain

Model Name	Data Set	32-bit floating (accuracy)	8-bit fixed point (accuracy)
Inception_v3	ImageNet	Top1: 70.62% (Top5: 89.38%)	Top1: 70.01% (Top5: 88.99%)
MobileNetV2	ImageNet	Top1: 71.43% (Top5: 90.46%)	Top1: 69.96% (Top5: 89.84%)
ResNet34	ImageNet	Top1: 73.08% (Top5: 91.14%)	Top1: 72.66% (Top5: 91.01%)
ResNet50	ImageNet	Top1: 75.87% (Top5: 92.91%)	Top1: 75.66% (Top5: 92.78%)
Tiny_Yolo_v2	VOC	MAP: 57.13%	MAP: 55.75%
Kneron_FR1	LFW	99.6%@FAR0.1%	99.4%@FAR0.1%
Kneron_FR2	LFW	97.92%@FAR0.1%	97.96%@FAR0.1%

Deep Compression Support Model

Deep level cooperation to achieve the ultimate goal

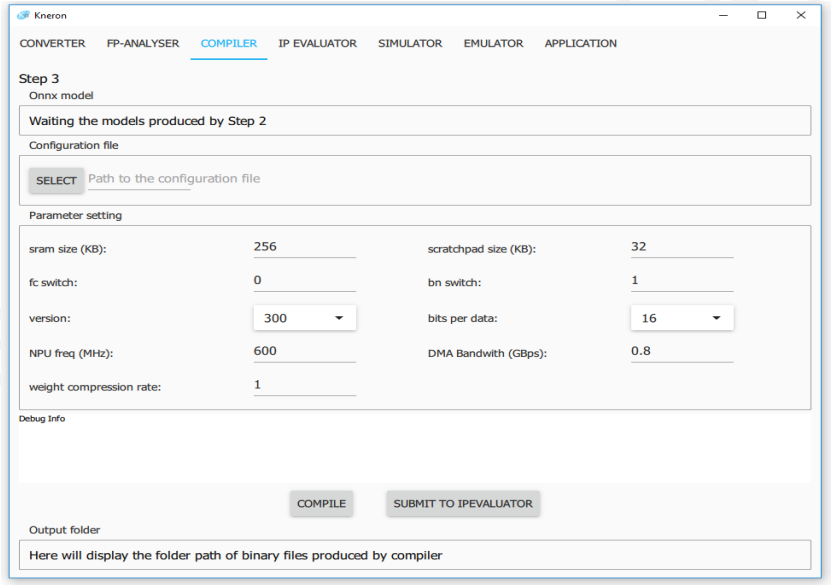


1. Collect pre-trained model and training/validation data
2. Deep compression and retraining by professional intelligence
3. Contribute compressed model and remain high accuracy
4. Customer can set the target accuracy loss

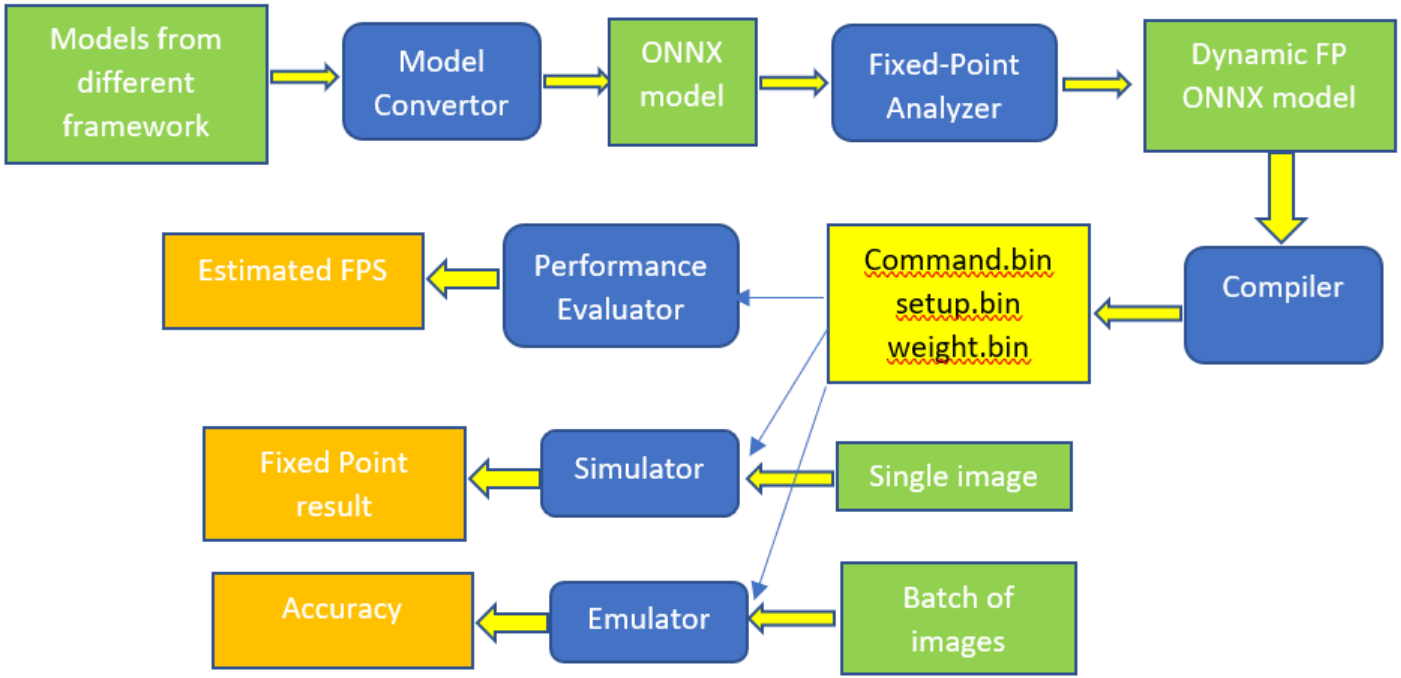
SDK Overview

Kneron NPU SDK provides:

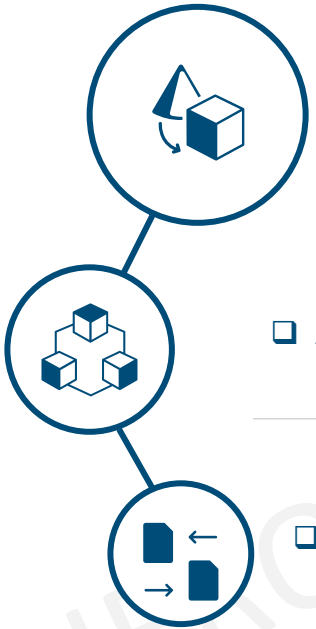
- Compiler for models compiling
- **Tool chains for simulation and evaluation**
- Runtime library for programming



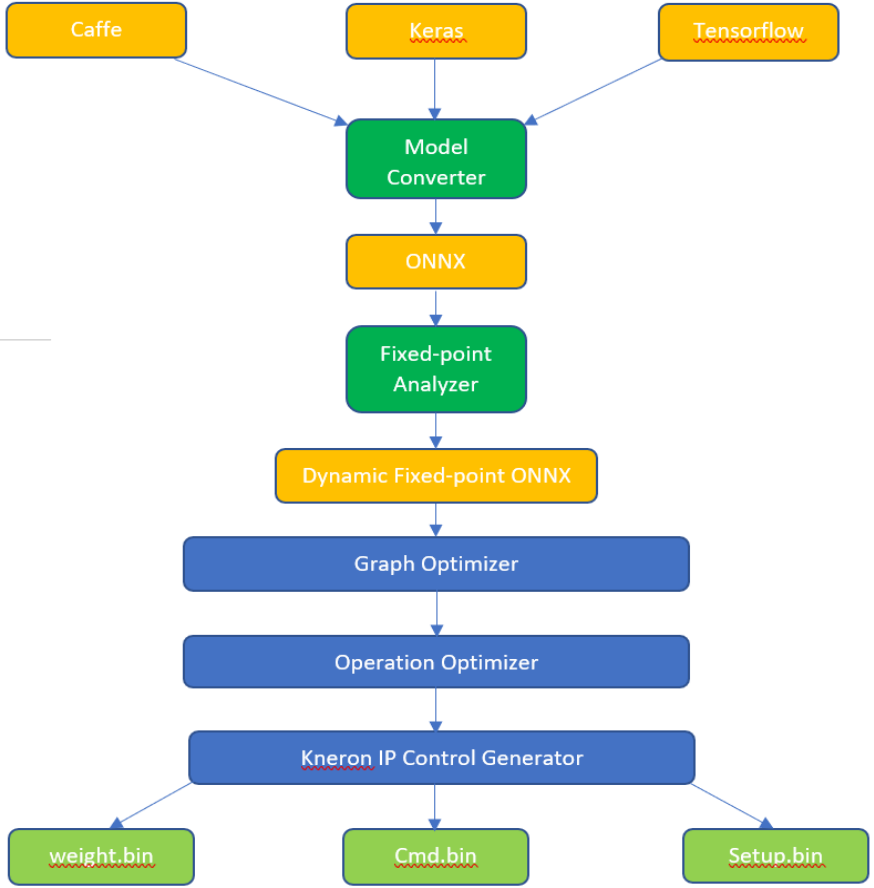
Tool Chain Flow



Compiler



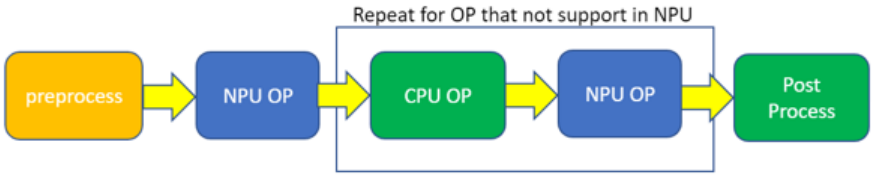
- Convert Models from Caffe, Keras, Tensorflow to ONNX
- Analyze ONNX model to create Dynamic FP ONNX model
- Through compiler to optimize model graph and generate binaries for NPU

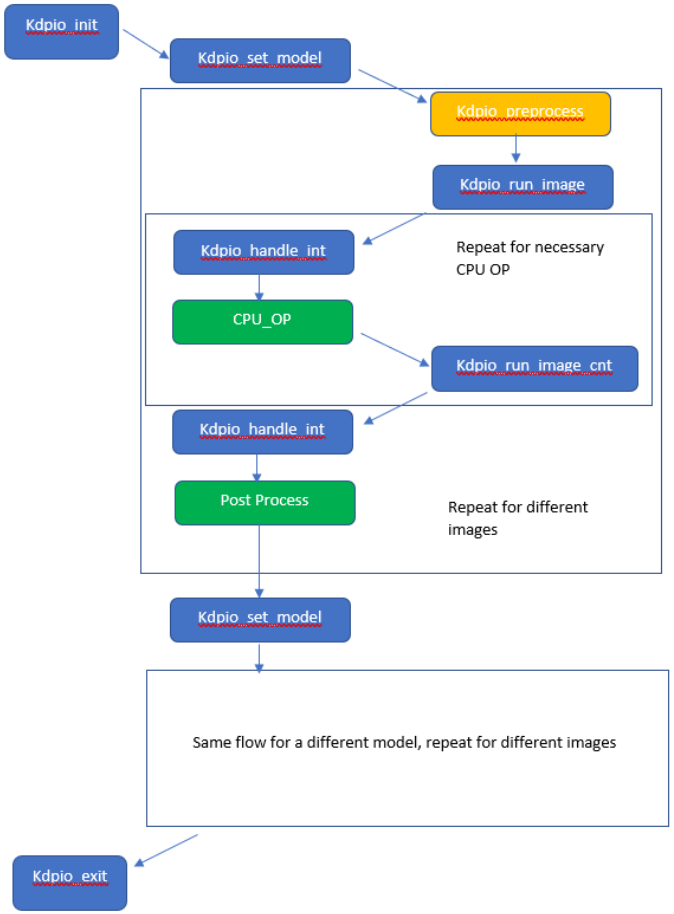


Runtime Library

- Kneron Runtime Library provides high level programming interface for users without knowing the NPU implementation details

- The reference flow as following:



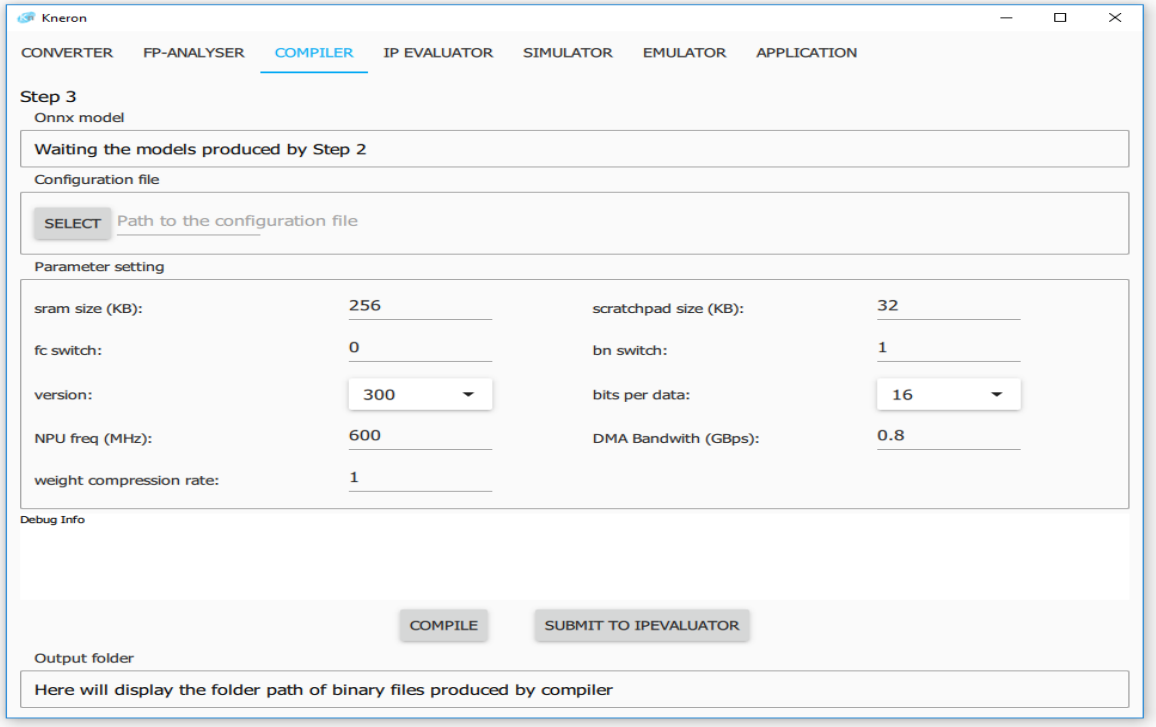


KL520 Support Layer

Layers/Modules	Functions/Parameters	Spec.
Convolution	Convolution kernal dimension:	1x1 up to 11x11
	Stride	1,2,4
	Padding:	0-15
	Depthwise Conv	Yes
	Deconvolution	Use Upsampling + Conv
Pooling	Max pooling 3x3	stride 1,2,3
	Max pooling 2x2	stride 1,2
	Ave Pooling 3x3	stride 1,2,3
	Ave Pooling 2x2	stride 1,2
	global ave pooling	Support
	global max pooling	Support
Activation	ReLu	Support
	Leaky ReLU	Support
	PReLU	Support
	ReLU6	Support
Other processing	Batch Normalization	Support
	Add	Support
	Concatenation	Support
	Dense/Fully Connected	Support
	Flatten	Support

SDK Tool and Evaluation Board Brief

Snap-shot of Tool Chain GUI

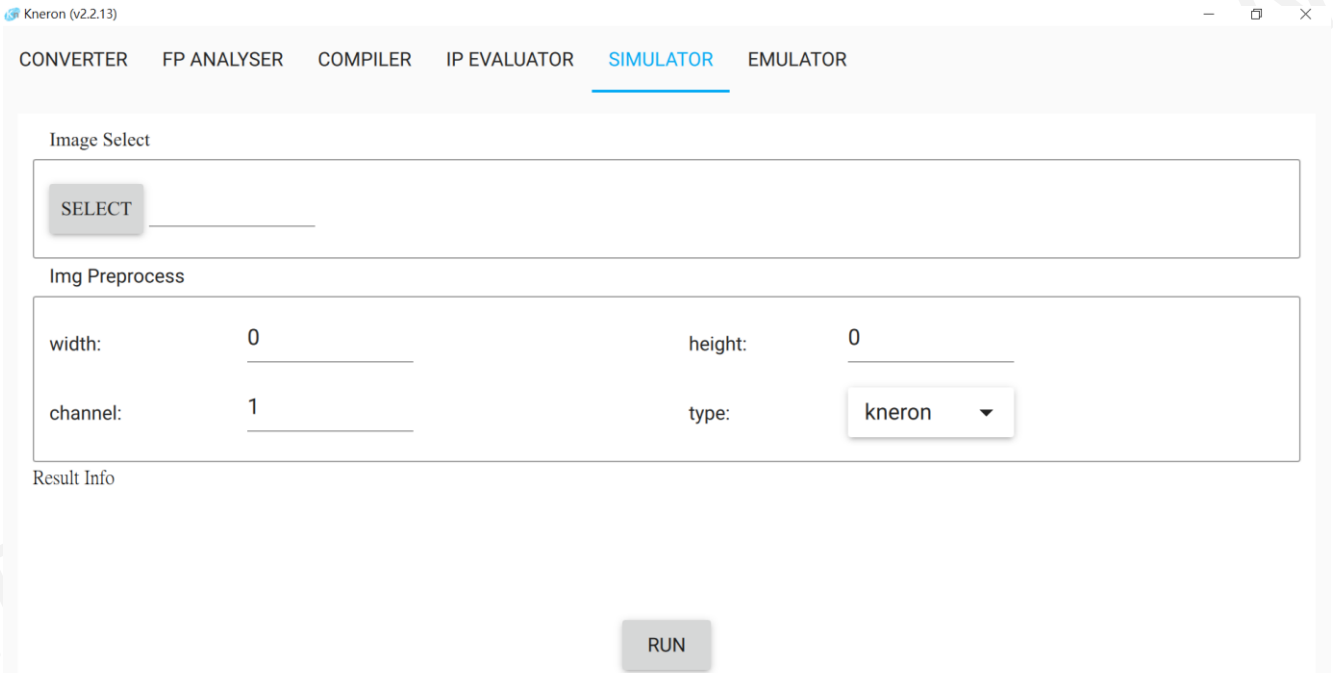


Kneronsdk_installer_2.2.13.exe

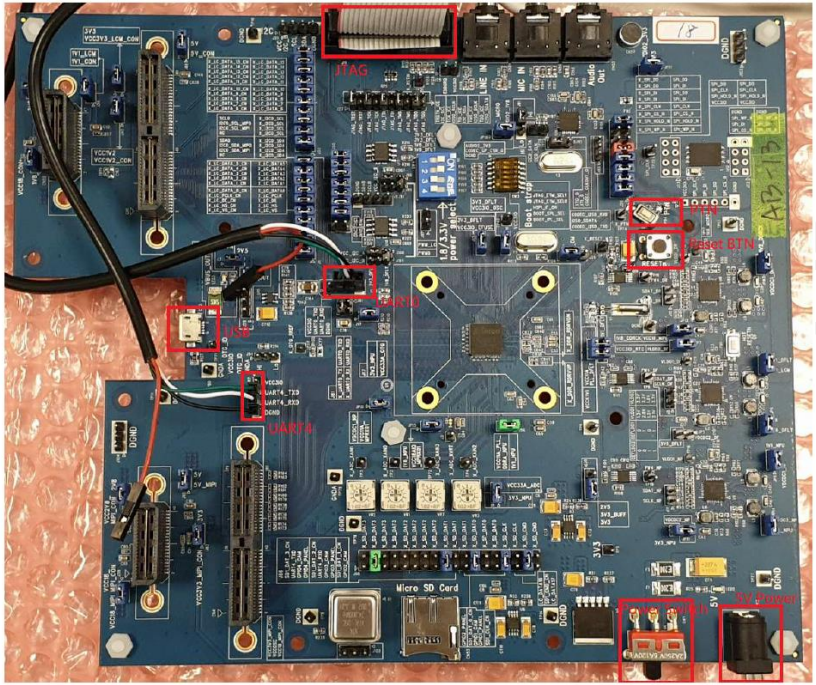
Kneron_NPU_TC_User_Manual

Kneron_sdk_v2.0.9_tutorial

Snap-shot of Tool Chain GUI



EVB Connector



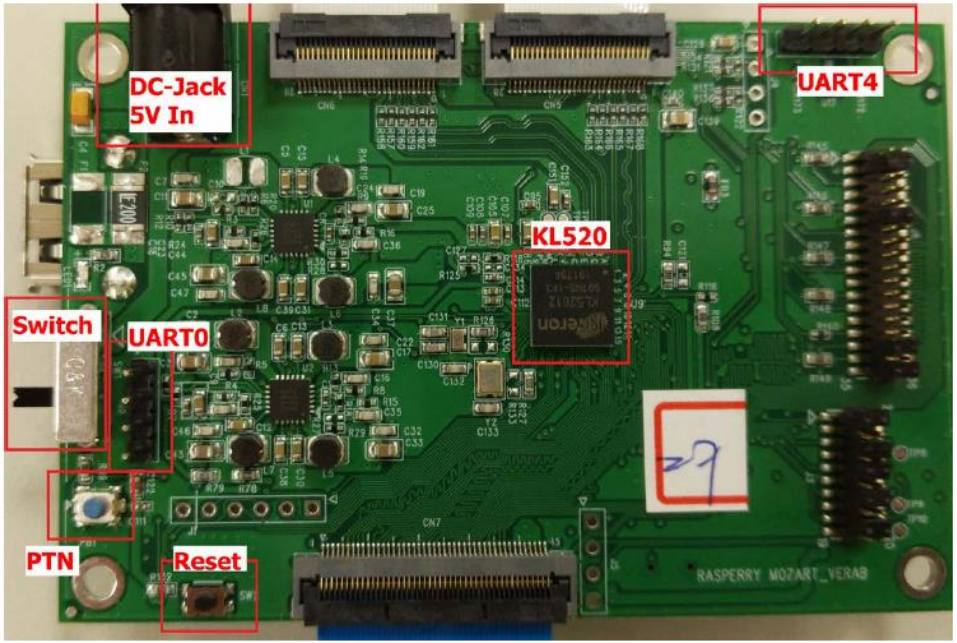
Major Components.

1. 5V Power (DCJack)
2. Power Switch
3. UART0/UART4
4. JTAG
5. PTN
6. Reset BTN
7. USB

TTL腳位定義說明

Dupont Line	Pin Define	Type	Direction:Host <<-->>Device	Desctiption
紅線	VO5	Output / Power	Host-->>Device	可以提供5V（100mA），以供外部的線路使用。另外可客製化成輸出5V（500mA） or 3.3V(100mA)。
白線	Tx	Output	Host-->>Device	Host: Transmitted Data，其準位是3.3V。如需要其它準位（1.8~3.3V），可以參考其它USB to TTL的板子。
綠線	Rx	Input	Host<<--Device	Host: Received Data，可以接受的準位最高可到5V。
黑線	GND	GND	Host<-->Device	地線

96 Board



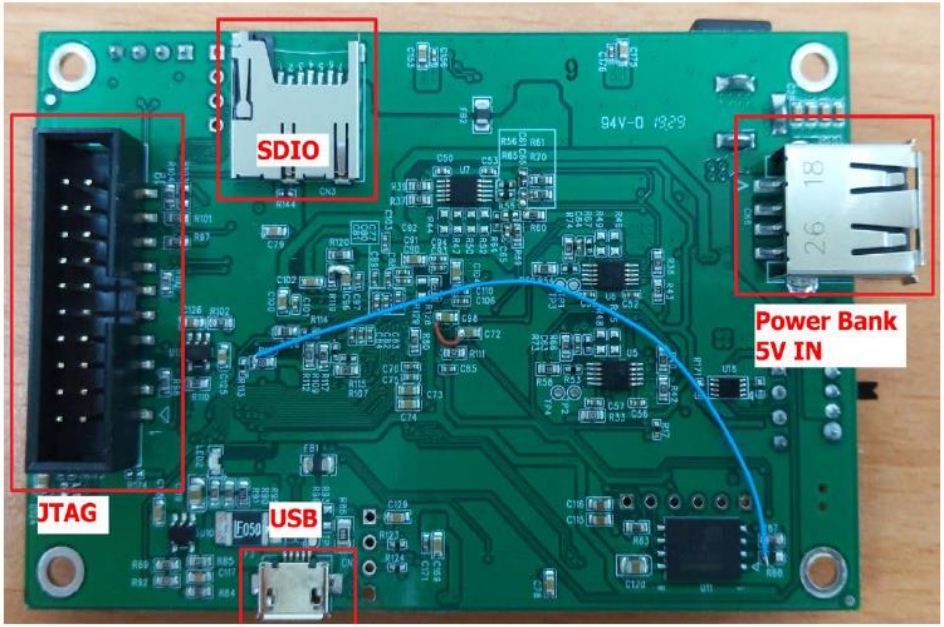


Figure 1, 96Board TOP overview

Board Debug HW Tool

UART cable

Figure 6, USB to TTL(3.3V) cable

JTAG cable

Figure 7, JTAG cable

Figure 5, 5V/4A adaptor

Board Debug HW Tool (II)

Connect UART cable and evaluation board as shown below:

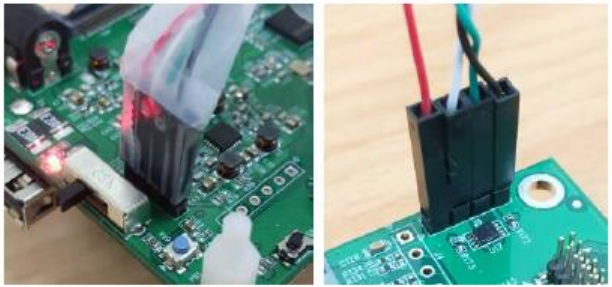


Figure 8, UART0 and UART4 connection

Connecting JTAG cable like picture shown below:

Figure 9, connecting JTAG cable

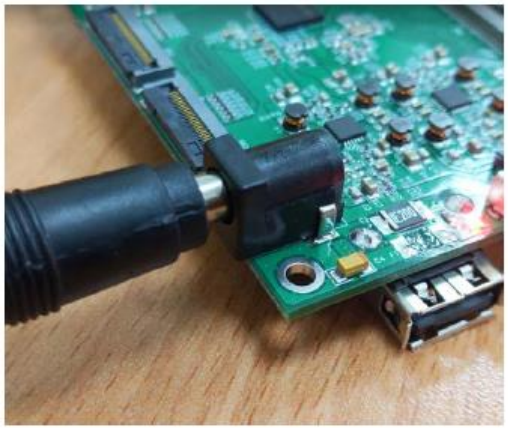
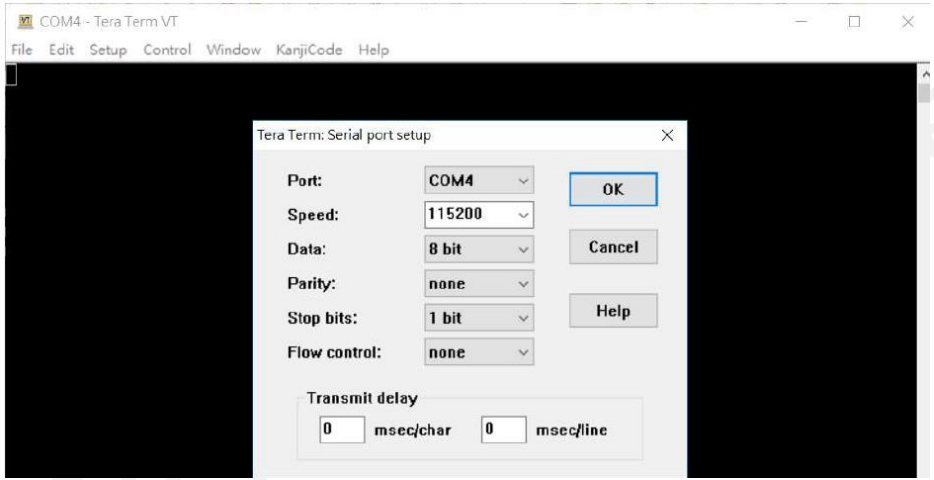


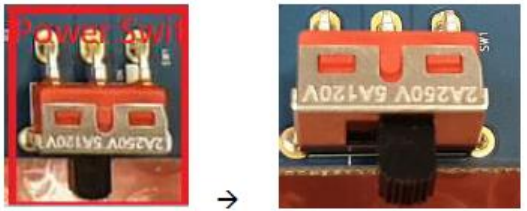
Figure 10, connecting 5V power

EVB Power On

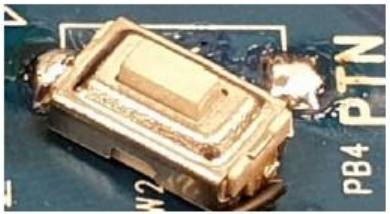
1. Open Uart COM port debug windows (Teraterm or Putty)
   UART0: Baudrate: 115200

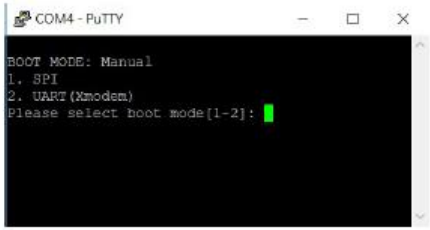


2. Trun ON power switch.



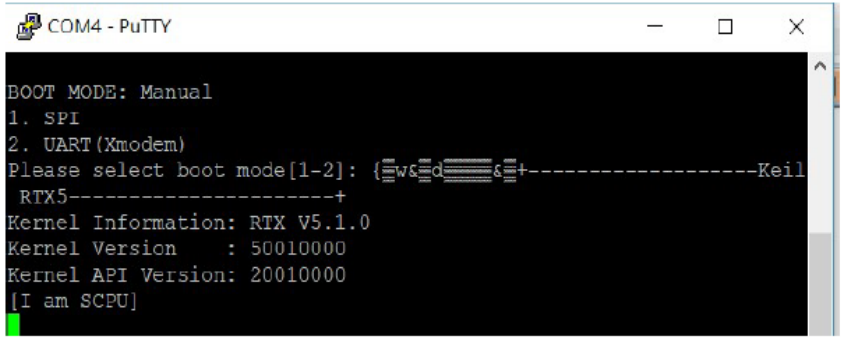
3. Wake up chip from RTC power domain.
   You will see boot message when you press PTN button



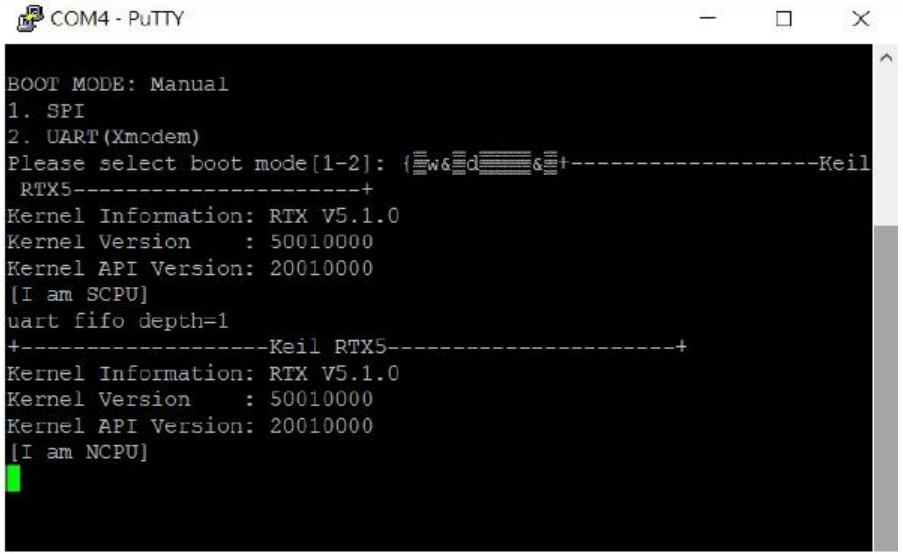


KL520 run on Keil SDK

Run Mozart SCPU project, it will show below message.

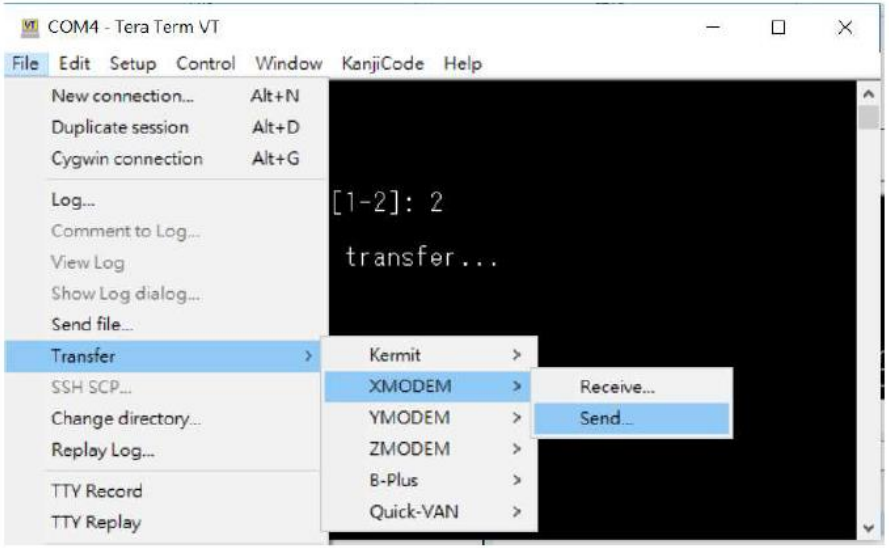


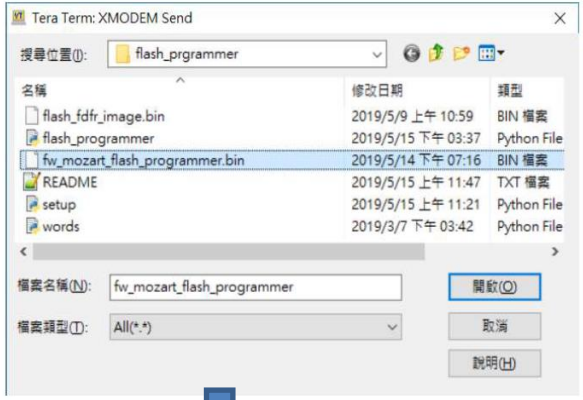
Run Mozart NCPU project, it will show below message.



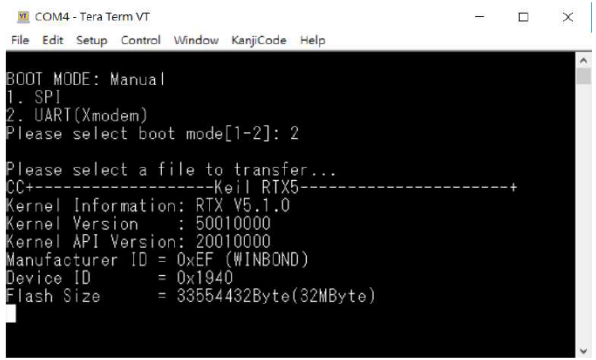
Load Flash Programmer Firmware

Upload “fw_mozart_flash_prgrammer.bin” firmware file by Teraterm XMODE send





After the firmware upload successful, the following message will be displayed.



Run Flash Programming on Python

Modify your **COM** port for **UART4** in "setup.py" (baudrate is 921600)

```
COM_ID = 5 # COM5
UART_BLOCK = 0x800
act_intf = INTF_UART
```

Memory Read/Write verification on Python

Please try run memory verification on python to verify your hardware platform.

```
>> python flash_programmer.py -t
```

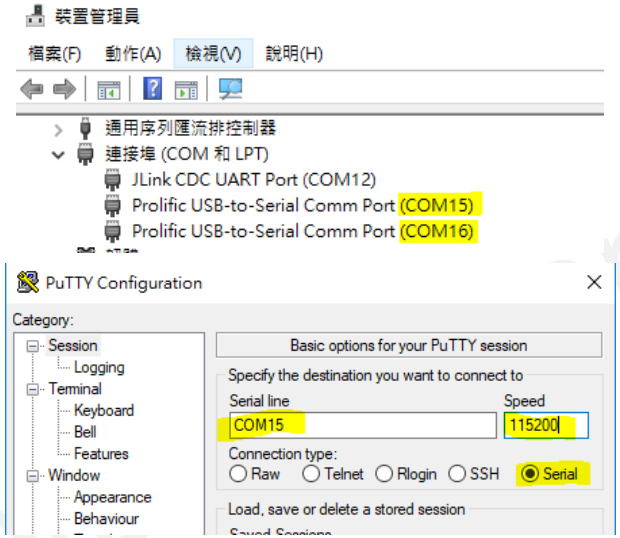
```
->uart write: 16
->uart read: 272
[DDR] Memory Read/Write verify PASS (100/100)
```

Necessary Tools/ Environment

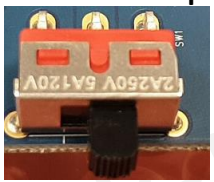
- Windows 10
- Putty
- Python / opencv_python

Board Bring up

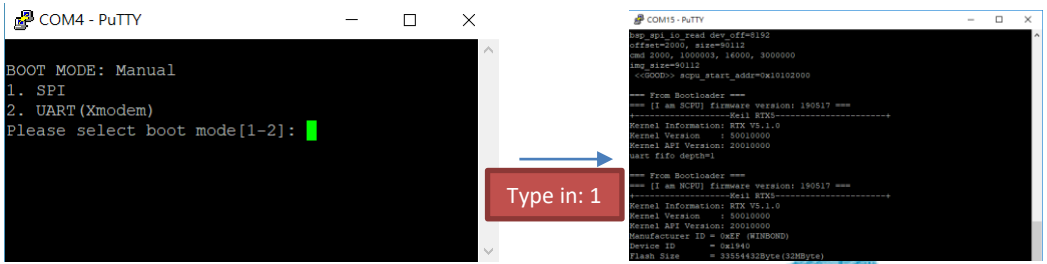
1. Open Uart COM port debug windows (Teraterm or Putty) UART0: Baudrate: 115200



2. Trun ON power switch.

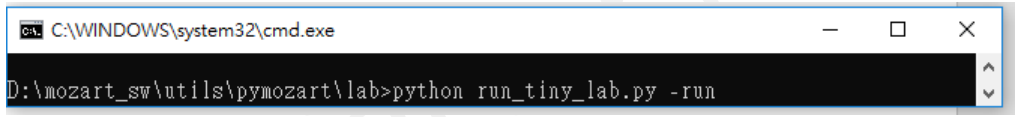


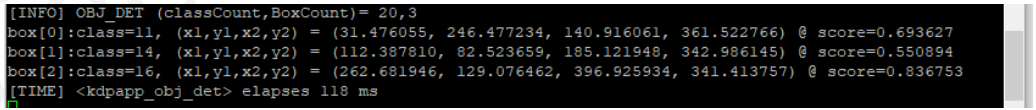
3. Wake up chip from RTC power domain.
You will see boot message when you press PTN button



Basic Operations through CMD console

- **[Basic flow]**
  - python run_tiny.py -info bin_files/yolo_info.bin
  - python run_tiny.py -set bin_files/yolo_setup.bin
  - python run_tiny.py -cmd bin_files/yolo_cmd.bin
  - python run_tiny.py -wt bin_files/yolo_wt.bin
  - python run_tiny.py -img images/img.bin
  - python run_tiny.py -run
  - python run_tiny.py -img images/img_2.bin
  - python run_tiny.py -run





- **[update model]**
  - python run_tiny.py -set bin_files/yolo_setup.bin
  - python run_tiny.py -cmd bin_files/yolo_cmd.bin
  - python run_tiny.py -wt bin_files/yolo_wt.bin

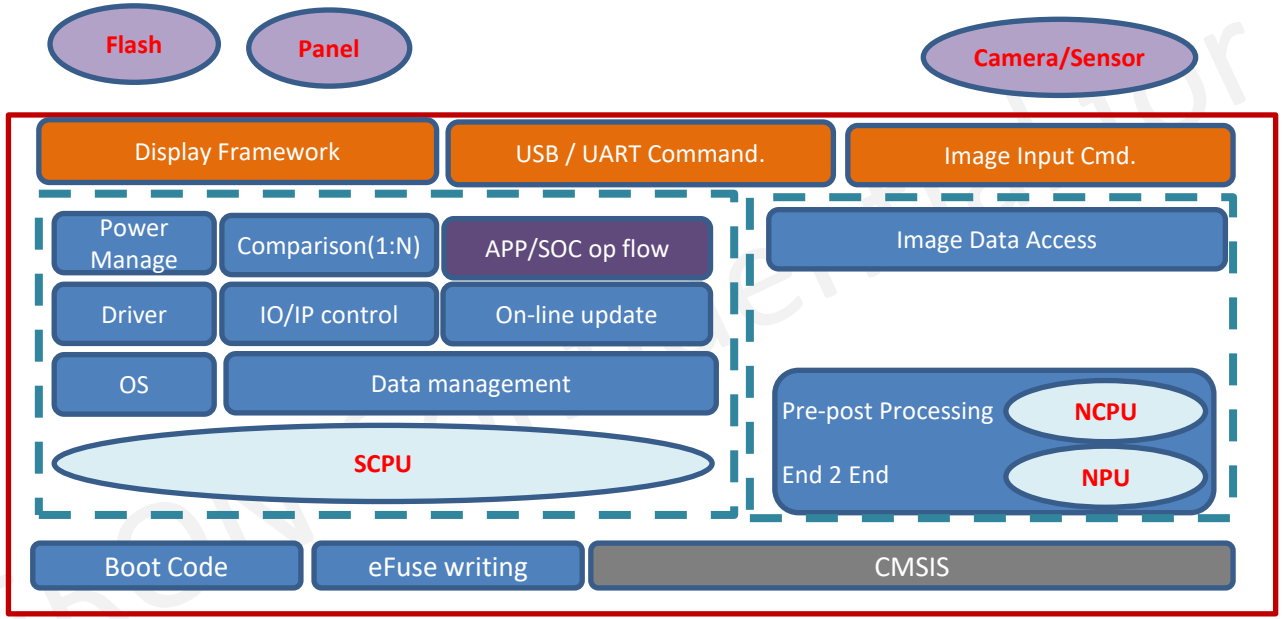
- **[prepare 416x416 image raw file]**
  - Prepare jpg image in directory **"orig_img"**
  - **python image_to_txtbin.py -t img2bin -a False -m kneron -i "orig_img/*.jpg" -o ./conv_bin -s_w 416 -s_h 416 -bw 8 -r 8 -c L**
  - The converted raw file will be generated as: **./conv_bin/orig_img/*.bin**

SDK

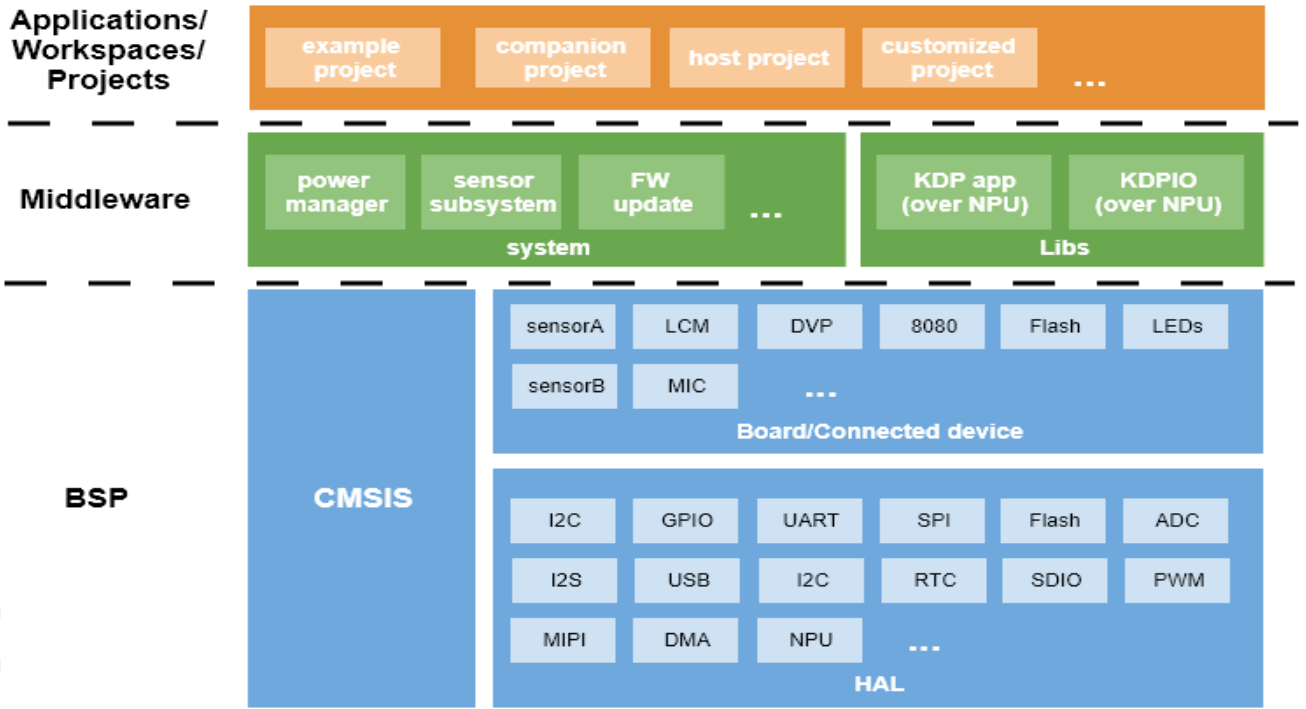
Basic Application Connections of KL520
Application Implementation Levels
- Host – Host Library
- SCPU implementation - KDP Application Library
- NCPU implementation - KDPIO library

SDK Utilities
- Image preparation tool
- flash programmer
- Firmware update

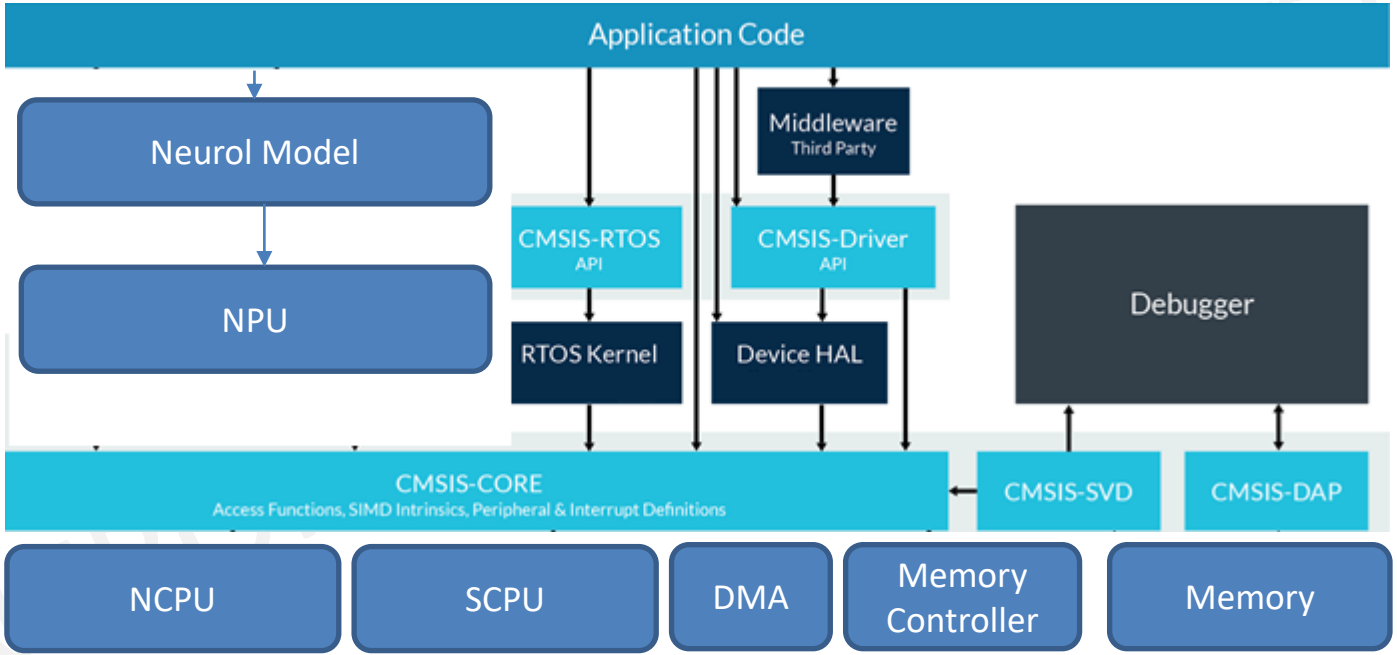
KL520 S/W Major Function Diagram



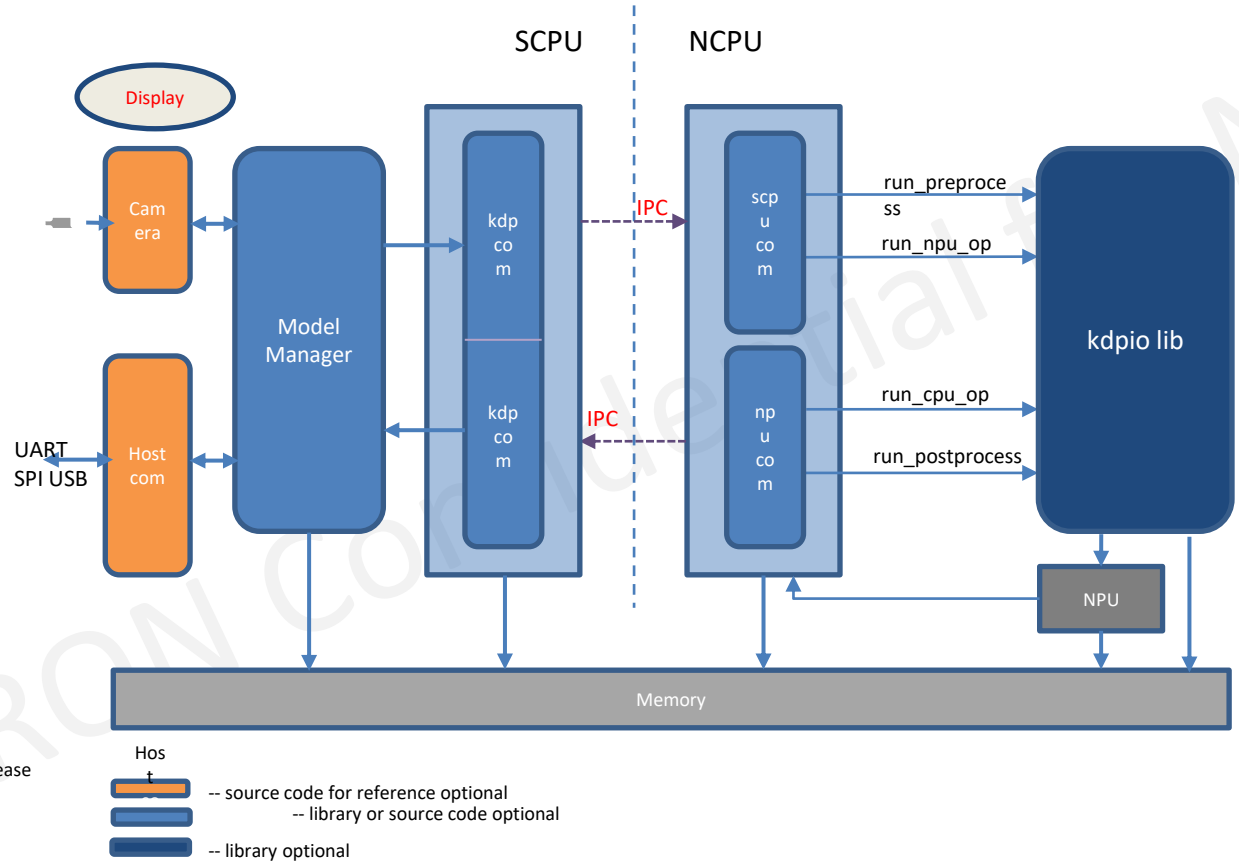
KL520 S/W Major Function Diagram (II)



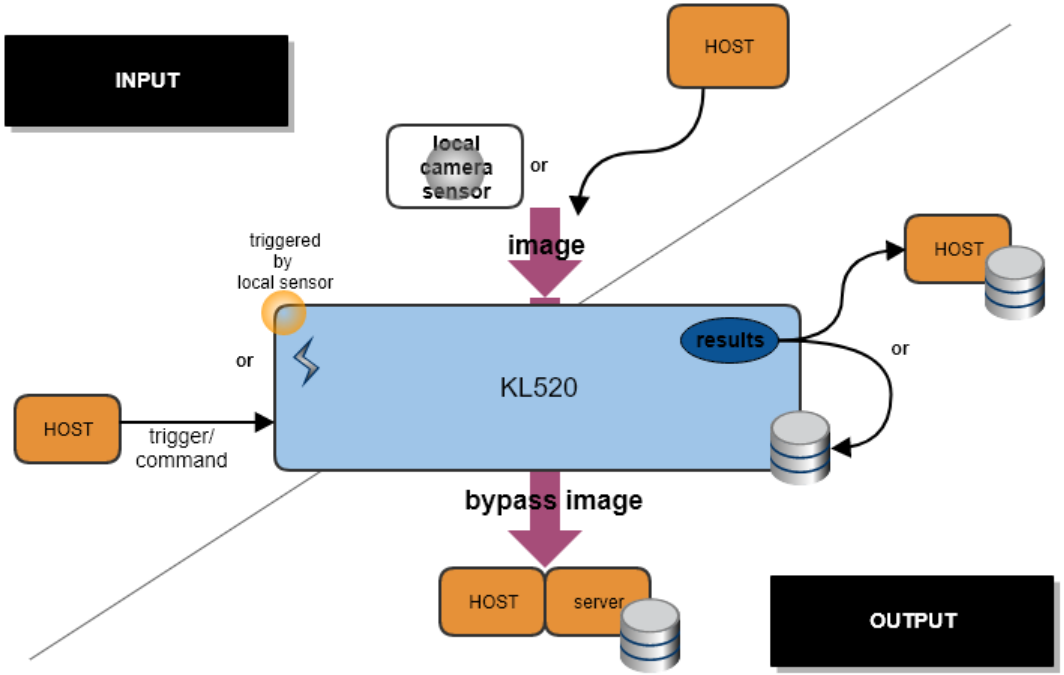
KL520 S/W Major Function Diagram (III)



SW System Blocks

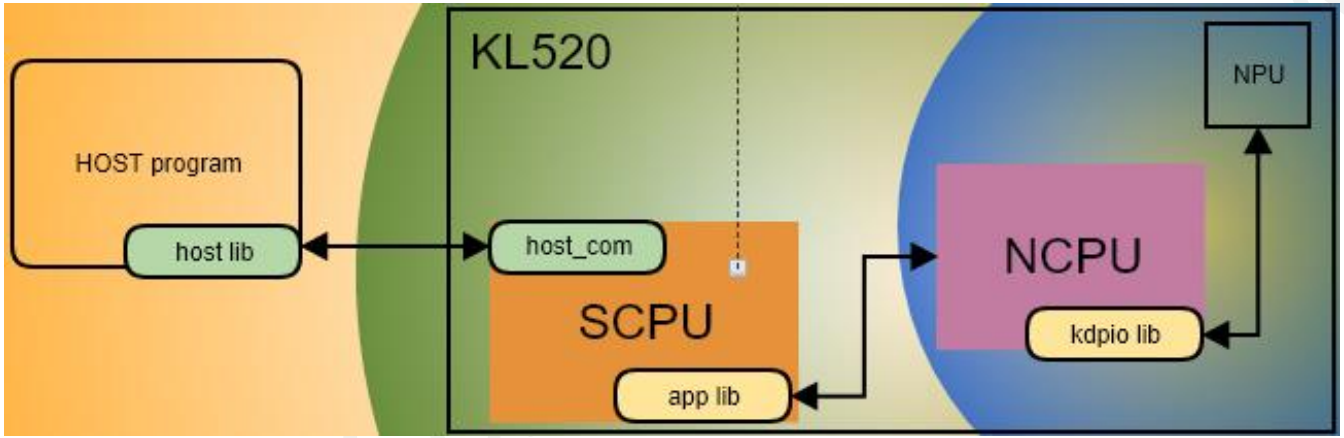


KL520 Application Architecture

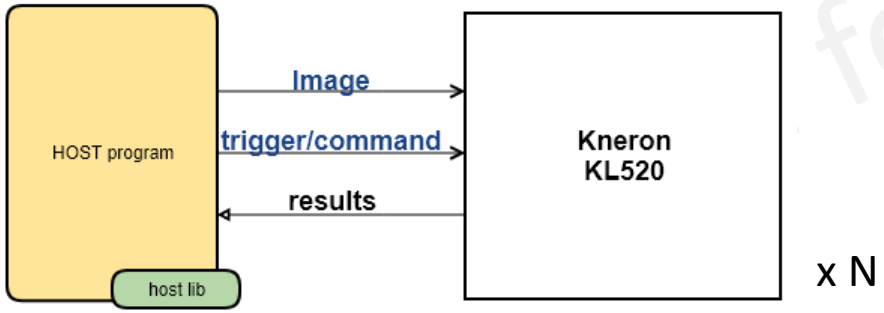


- Trigger
- Image source
- Result output
- Image output(opt.)

KL520 Applications in 3 Levels



System Application by Customers (Host)



- HOST side program owned by user
- Interface: USB or UART (mixed mode is not supported)
- Reference:
  - ***Kneron KDP Host Lib***

KDP Host Library - Unittests

- **There are several demo programs: deluser, reguser, veruser, test, l3wd, dme, udt_fw, udt_md.**
  - Program deluser is used for remove a specific user from device database. If 0, remove all users.
  - Program reguser is used to register the uid (given by argument) to database.
  - Program veruser is used to extract the face feature and match it in the device database.
  - Program test is used for all the APIs contained in this library.
  - Program lw3d is used for the testing of light weight 3D functionalities.
  - Program dme is used for the testing of dynamic model execution functionalities.
  - Program udt_fw is used for the testing of firmware update functionalities.
  - Program udt_md is used for the testing of update model functionalities.

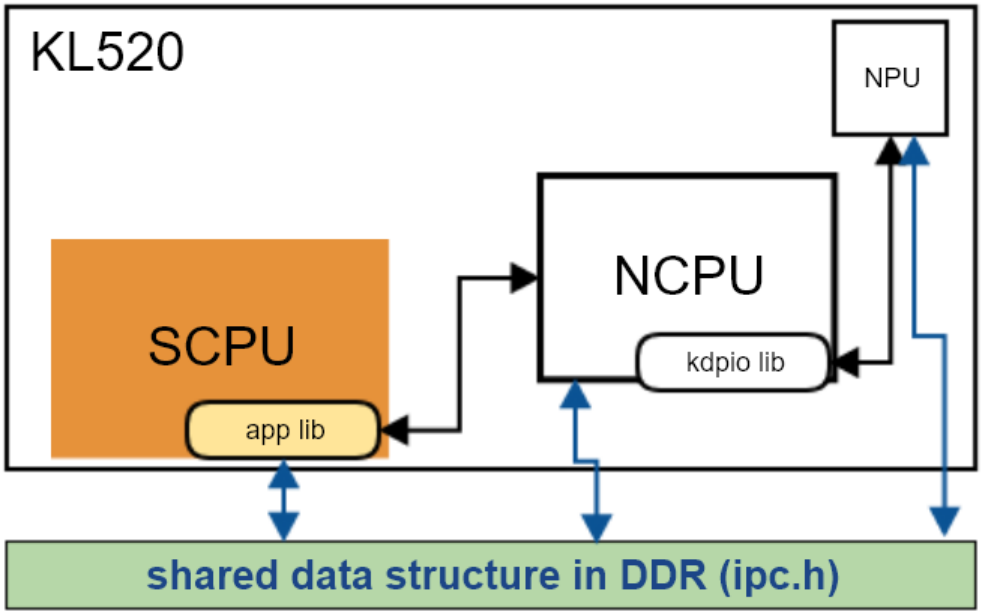
Board Application by Customers (SCPU)

- Init KDP Application (*kdp_app_init()*)
  - Ref: *main.c*
- Design protocol to communicate with Host
  - Ref: *host_com.c*
- Capture image and save image at DDR_addr: KDP_DDR_BASE_IMAGE_BUF
  - Ref: v2k_cam.h
- Get trigger and Call *kdp_app_xxxx()*
  - *Ref: host_com.c*
- output results to host
  - Ref: host_com.c

- Ref: *KDP Host Interface Message Protocol v1_0*

KDP Application Library – data structure (ipc.h / kdp_app_xxx.h)

- Image setting
  - *struct kdp_img_raw*
- Application result data structure (in/output)
  - *kdp_app_XXXX_t*



Memory Usage in SCPU - DDR

- *#include “kdp_memory.h”*
- *kdp_ddr_init(uint32_t start_addr, uint32_t end_addr)*
- *kdp_ddr_malloc(uint32_t size_in_byte)*
- ~~*kdp_ddr_free(uint32_t addr) //not implemented*~~

Memory Usage in SCPU - Flash

```
#include “kdp_memxfer.h”

extern const struct s_kdp_memxfer kdp_memxfer_module;

kdp_memxfer_module.ddr_to_flash(addr_dst_flash, addr_src_ddr,
    total_size);
```

For other usages, refer to kdp_memxfer.h

Print Functions

#include “dbg.h”

dbg_msg(…)

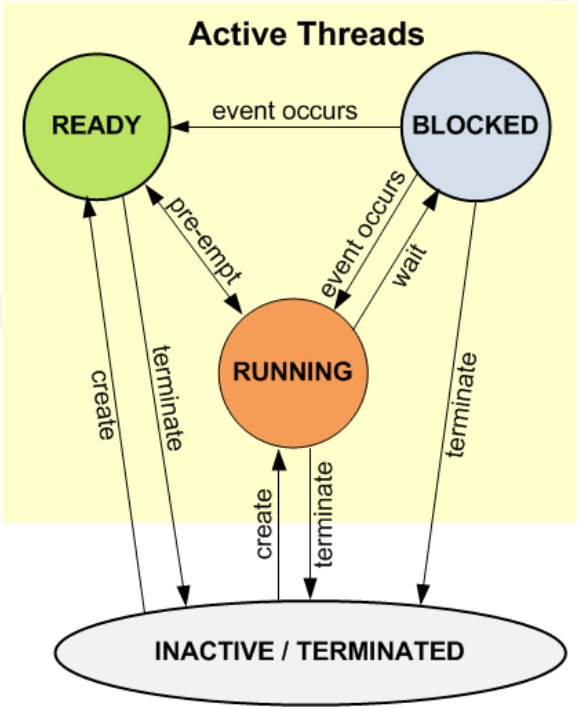
err_msg(…)

…

For other level of print functions, refer to dbg.h

Thread Controls

- https://www.keil.com/pack/doc/CMSIS/RTOS2/html/group__CMSIS__RTOS__ThreadMgmt.html



Model Design by Customer (NCPU)

What you need:

Kneron KL520 toolchain (separated package)

“KDP KDPIO Library”,

ref: KDP Host library software design v005-0801

NCPU reference design, ncpu.uvprojx

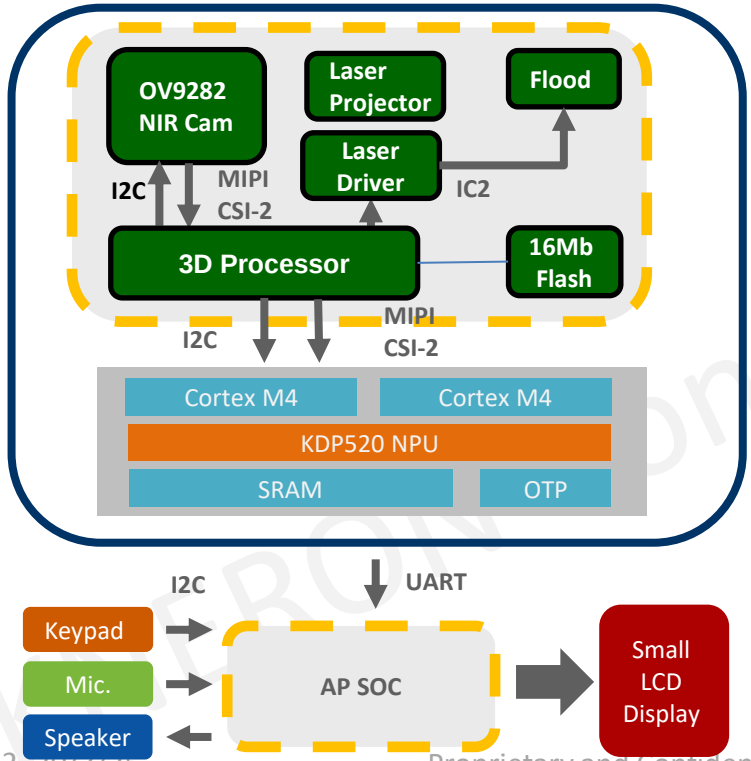
Apply the customized models

DME flow

flash_programmer utility

Extend Application

Structure Light Face ID Block Diagram



Item	Specification
Construction	3D projector + NIR sensor+ 3D processor + laser/flood combo driver IC + Flood (option)
Baseline (Tx to Rx)	25mm~ 32mm
Operation Distance	20~100cm
Wavelength	940nm
NIR Resolution	864*491 @ 30fps
3D Depth Resolution	216*124 @ 30fps
Field of View	DFOV:72.8° +/- 3°
3D projector	212*155 (~30K)
Reference Module size (baseline@25mm)	33.65mm * 10.82mm * 4.97mm (x*y*z)
Eye Safety Compliance	IEC60825-E3 2007 Class1
FR accuracy	99.2% @FAR 0.1%
Recognition speed	<0.2s

3D sensing comparison

	Structured Light	Stereo Vision	Time of Flight (ToF)
Distance	0.5m~1.0m	0.2m~3.5m	0.5m~4.0m
Suitable for	1. Indoor or low light environment 2. Short range and static objects with high depth precision	1. Both indoor and outdoor environment 2. Short to mid range detection. 3. Both static or moving objects	1. Indoor environment 2. Short to mid range detection 3. Quick moving objects

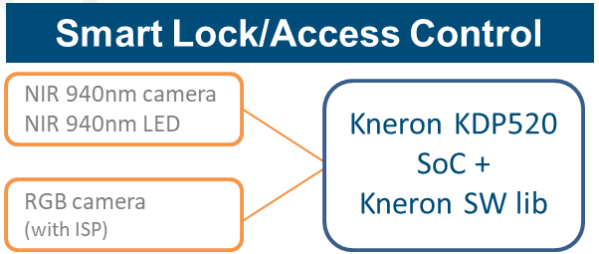
Lightweight 3D facial recognition

Definition

- 3D wide spectrum facial recognition solution, using **only regular RGB & NIR camera without baseline calibration needed**, and suitable for outdoor / indoor / low light / dark environments.

Applications

- 3D facial recognition
- 3D liveness detection
- Real-time face depth map
- 3D face modeling



Highlights

- High level security as structured light
- Smaller size and flexible camera position
- Dramatically low hardware cost

(less than 1/3 of structured light module, in-screen fingerprint sensor etc cost .)

Hardware requirements

Camera

1. NIR 940nm camera (1M pixel, rolling shutter)
2. NIR 940nm LED
3. RGB camera
    - **3-1. Door lock / Access control:** 1M pixel camera module with ISP integrated on CMOS
    - **3-2. Other devices:** 1M pixel camera module

#. No camera baseline calibration required

Processing unit

1.Door lock / Access control: KL520 SoC

2.Other devices: KL520 SoC + AP on main system

Thank You

Kneron

AI Everywhere

www.kneron.com