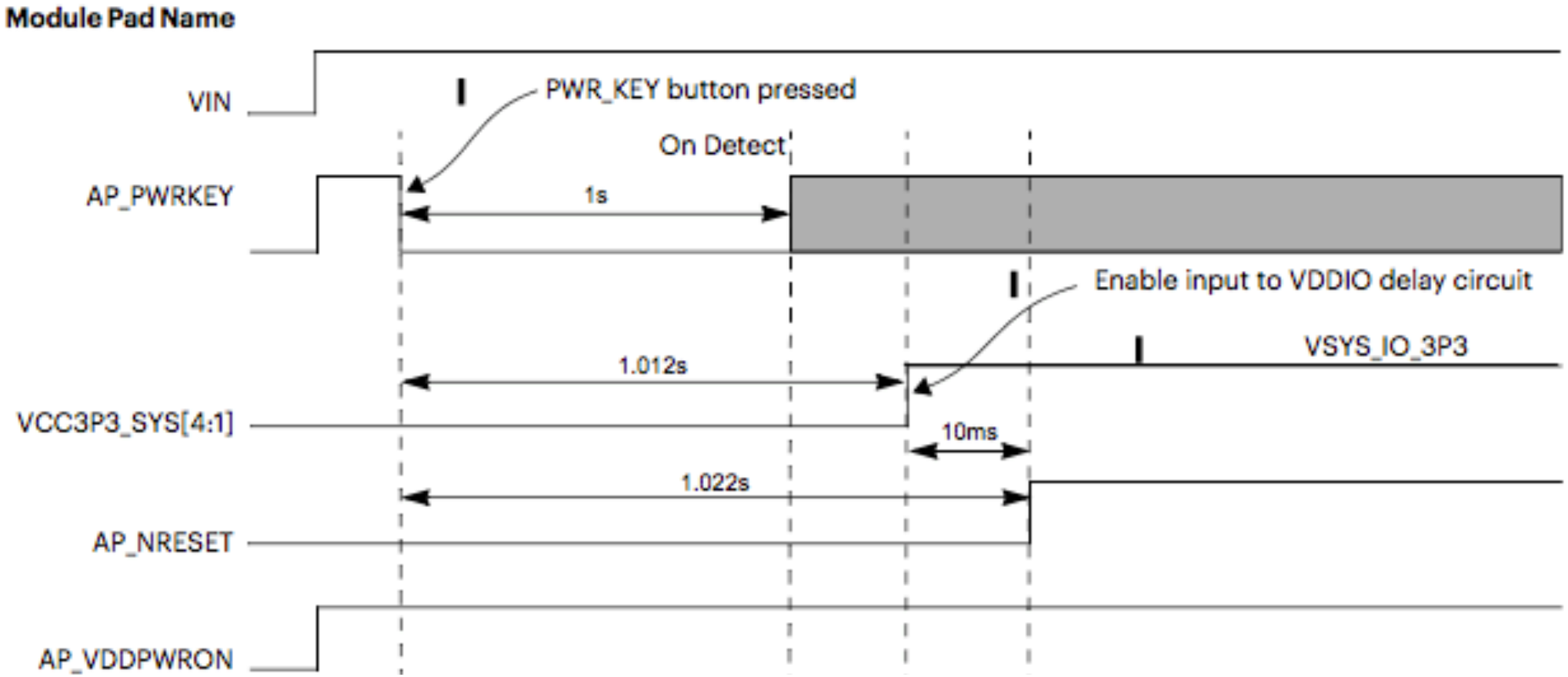
ARTIK System Design

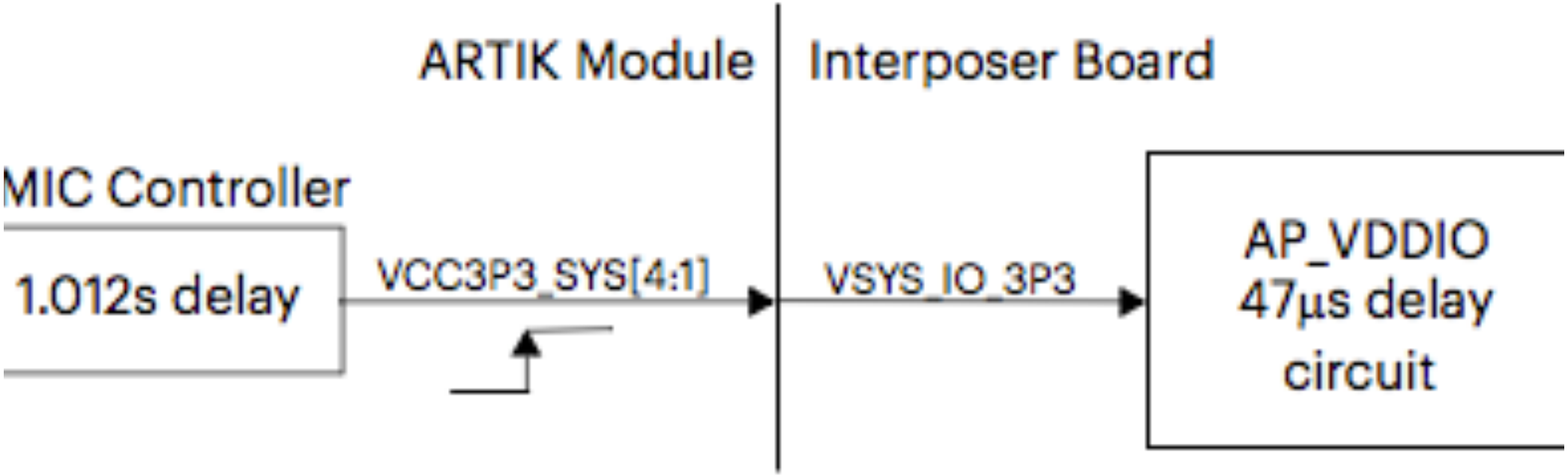
https://github.com/SamsungARTIK/Training

ARTIK Based Product Development Phase Considerations

Board Bring-up – Power up Sequence



Board Bring-up – Power up Sequence

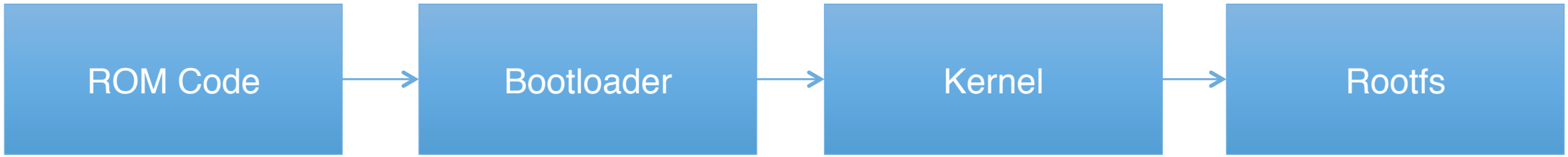


3 Boot Options

DIP Switch No	eMMC	microSD	USB OTG
1	OFF	OFF	ON
2	OFF	OFF	ON
4	OFF	ON	x

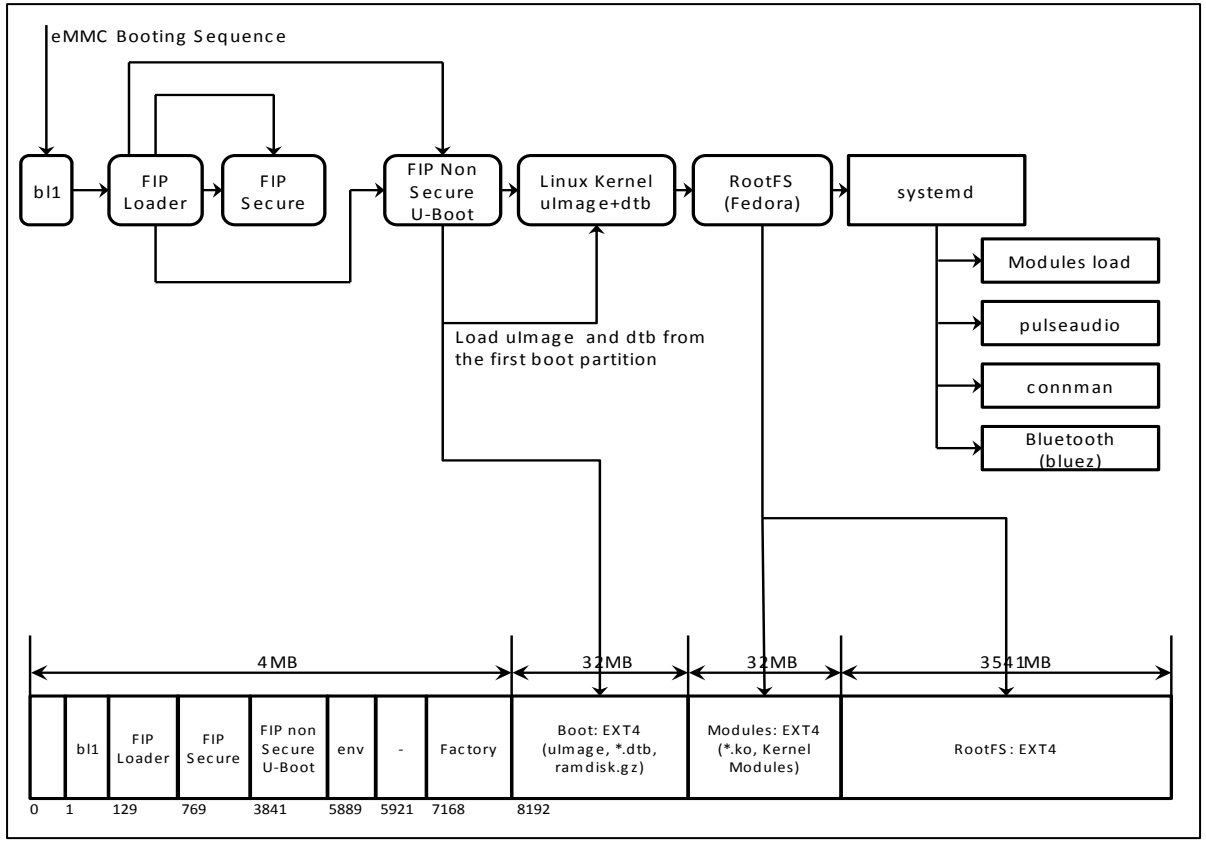
Dip Switch No. 3 is always OFF

Secure Boot



SAMSUNG ARTIK™

ARTIK Gateway eMMC Boot Sequence



Bootloader Partition

- ***bl1*** : The first Bootloader of the Hardware Modules. The content is loaded into SRAM.
- ***FIP-loader*** : This loader loads the FIP Binary.
- ***FIP-secure*** : Secure bootloader that will initialize the secure area.
- ***FIP-non secure***: U-Boot partition

User Partition

- '***/boot***' (Read only): The kernel images such as uImage and tree blobs.
- '***/lib/modules***' (Read only) : This partition contains the kernel modules. In addition the root file system will mount '/lib/modules'.
- '***/***' (Read/Write) : Root file system.

SAMSUNG ARTIK™

ROM Code

- Need a way to store the public keys which will be used to decrypt the signature of the bootloader. The mechanism has to be tamper-proof.
- Basic flow:

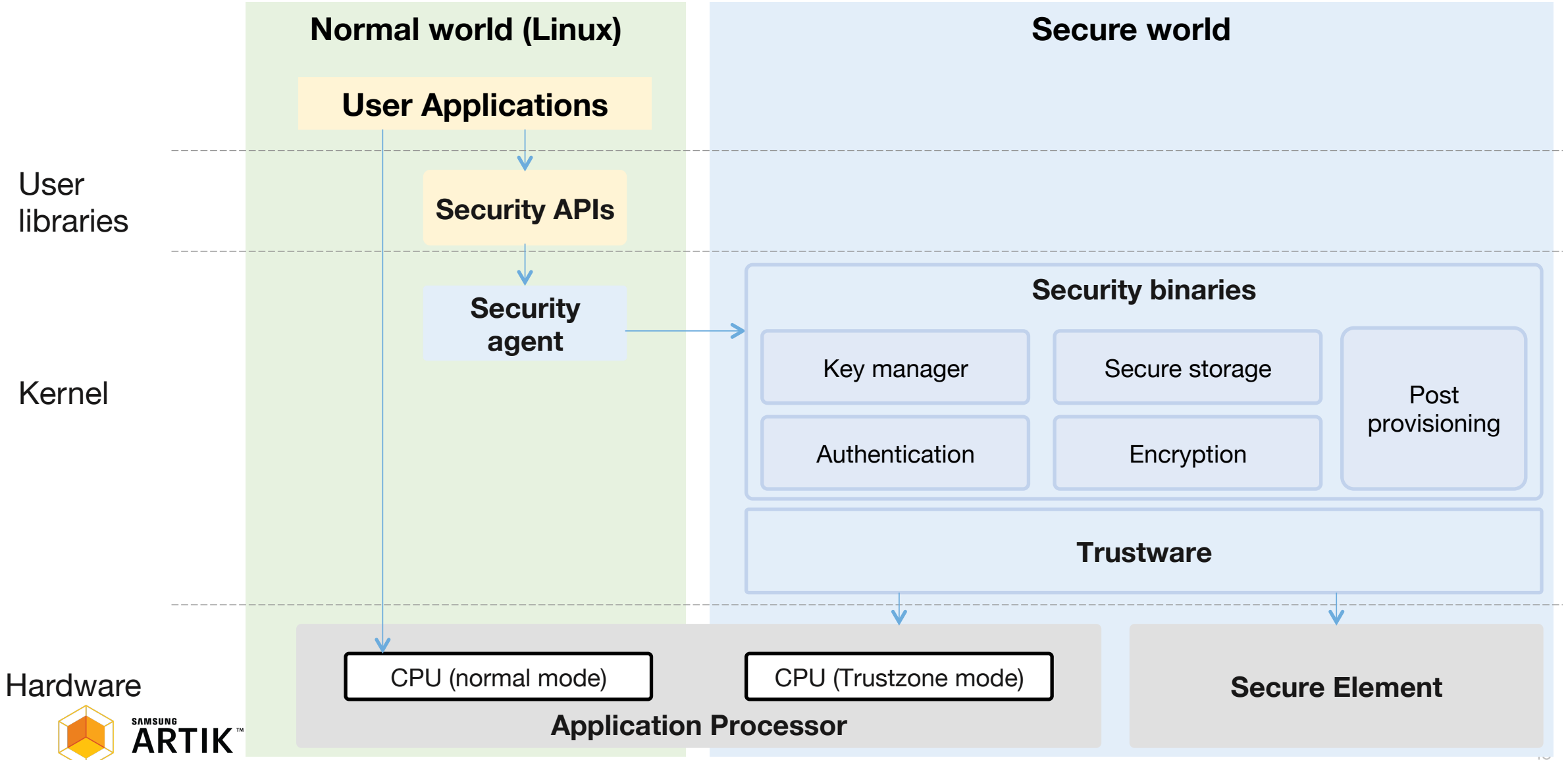
1. Loads the bootloader in a secure space to avoid physical attack
2. Loads the embedded public key
3. Checks the hash of the public key against the hash table in OTP
4. Uses this verified public key to check the signature of bootloader
5. Executes the bootloader binary

SAMSUNG ARTIK™

Kernel Authentication

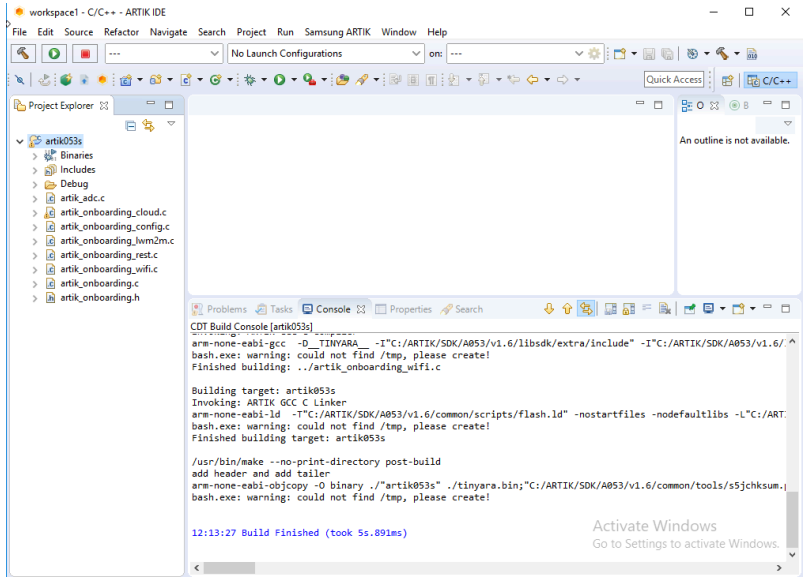
- U-Boot has Device Tree Blob(DTB) support
- DTB can also be used to store a public key, which is used to verify the signature of kernel image
- FIT Image that includes all the images needed to boot a system
- `mkImage` provides built-in support for signing of binaries hash

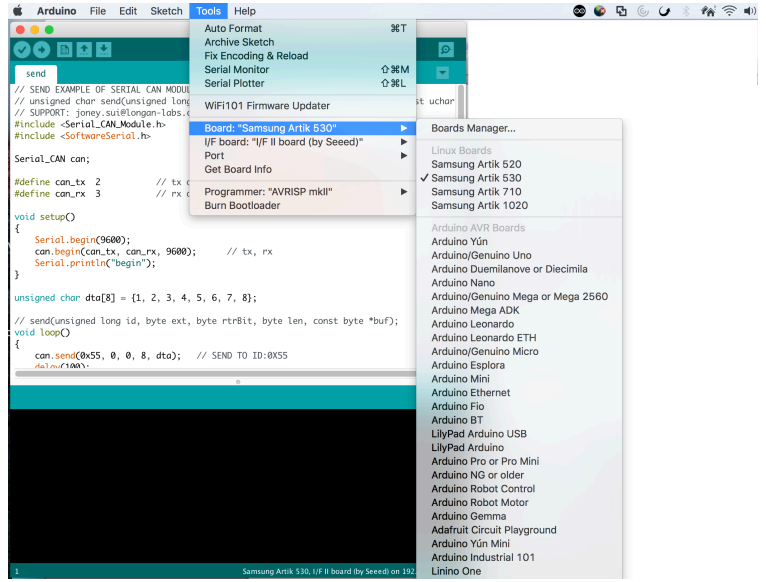
Trusted Execution Environment(TEE)



SAMSUNG ARTIK™

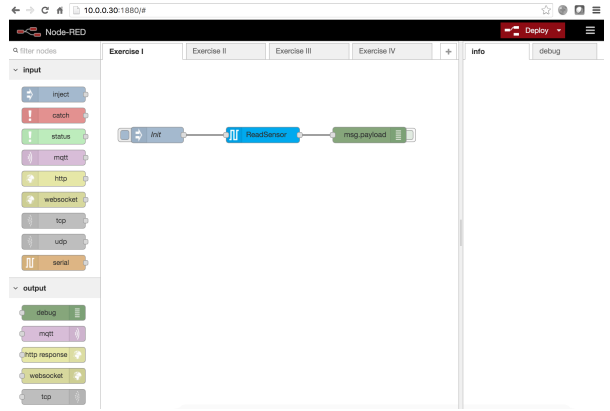
Application Development Environments





Arduino

SAMSUNG ARTIK™ IDE



Node red

SAMSUNG ARTIK™

gcc-arm-linux-gnueabihf
and
aarch64-linux-gnu

Application - Native Development

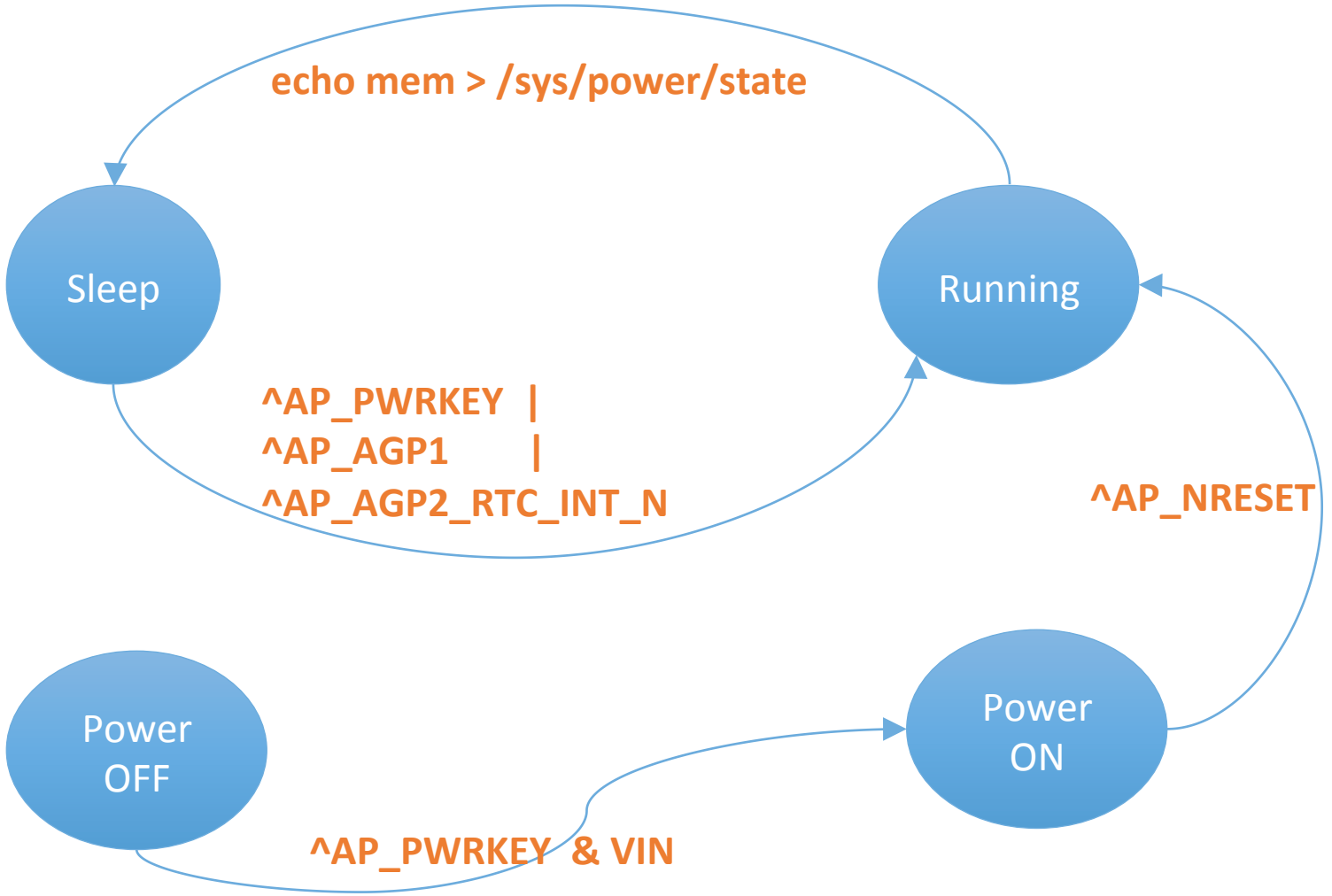
- C: Most popular programming languages for embedded devices. e.g, ARTIK SDK
- Python: 2.7 and 3.0
- JS: Node.js is the most popular JavaScript runtime for high-end IoT devices.
- Java: JDK is required.

Kernel development using a Virtual Machine

- Use virtual box with VM Virtual box extension on a Windows or Mac environment and create a Ubuntu 64bit virtual machine
- Reserve at least 8GB of RAM for virtual Machine
- Use VMDK memory for Virtual Memory Disk, allocated dynamically
- Share the USB and Network resources from host environment

ARTIK based design
Deployment Phase considerations

Power Management



SAMSUNG ARTIK™

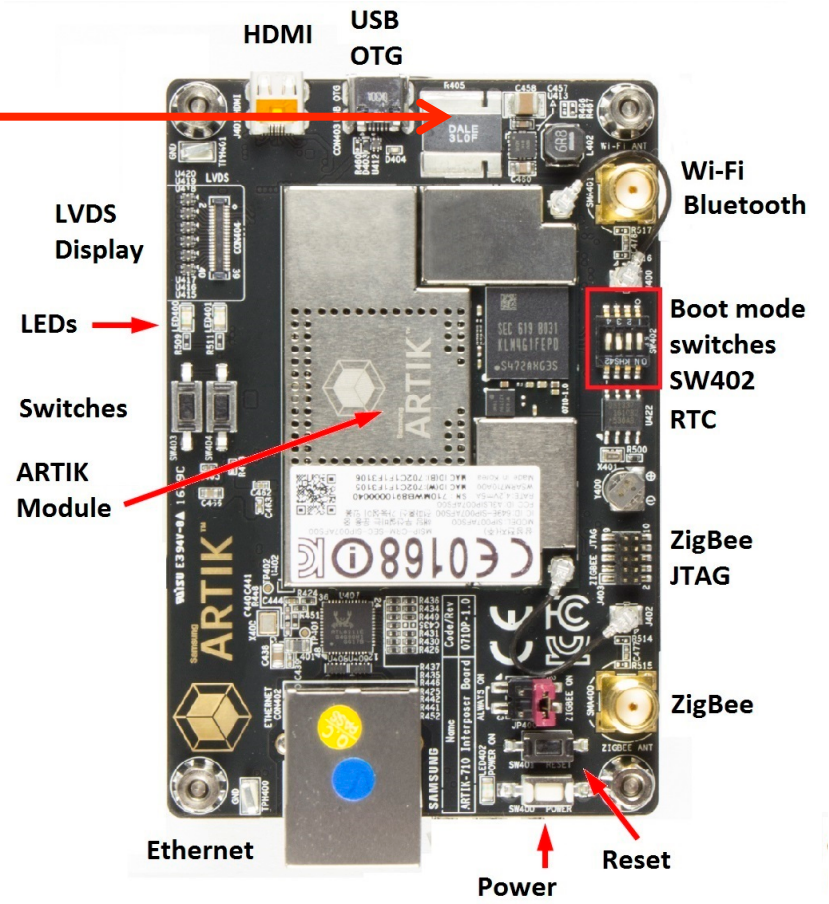
Performance Tuning – To Measure Power

Measure voltage drop against 3 mΩ sense resistor (R406) in interposer board

(Voltage drop (V) / resistance (Ω)) x Input voltage = Power (watts)

say, 1mV voltage drop across R406 indicates 1.667 watts by ARTIK module

(0.001/0.003) x 5 = 1.667 W



SAMSUNG ARTIK™

Performance Tuning - Power

- Disable unused GPIO from uboot
  - Uboot(530): u-boot-artik/arch/arm/dts/s5p4418-pinctrl.dtsi

```
pinctrl@C0010000 { +  unused_pins: unused-pins {
+  /*XGPIO0 - XGPIO9 on Artik platform 710/530 */
+  nexell,pins = "gpioe-0", "gpioe-1", "gpioe-2", "gpiob-14", "gpioa-14",
+  "gpiob-9", "gpioa-25", "gpioa-0", "gpioa-26", "gpioa-27";
+  nexell,pin-pull = <0>;
+  nexell,pin-strength = <0>;
+  };
/
```

  - Uboot(530): .../u-boot-artik/arch/arm/dts/s5p4418.dtsi

```
pinctrl-0 = <&gmac_txd>, <&gmac_rxd>, <&gmac_txen>,
-  <&gmac_mdc>, <&gmac_mdio>, <&gmac_rxclk>, <&gmac_txclk>;
+  <&gmac_mdc>, <&gmac_mdio>, <&gmac_rxclk>, <&gmac_txclk>, <&unused_pins>;

status = "disabled";
```

Performance Tuning - Power

- Disable unused GPIO from kernel
  - linux-artik/arch/arm/dts/s5p4418-pinctrl.dtsi

```
serial2_pin:serial2 {
        nexell,pins = "gpiod-16", "gpiod-20";
        - nexell,pin-pull = <2>;
        + nexell,pin-pull = <0>;
        nexell,pin-strength = <0>;
};
```

Performance Tuning - Power

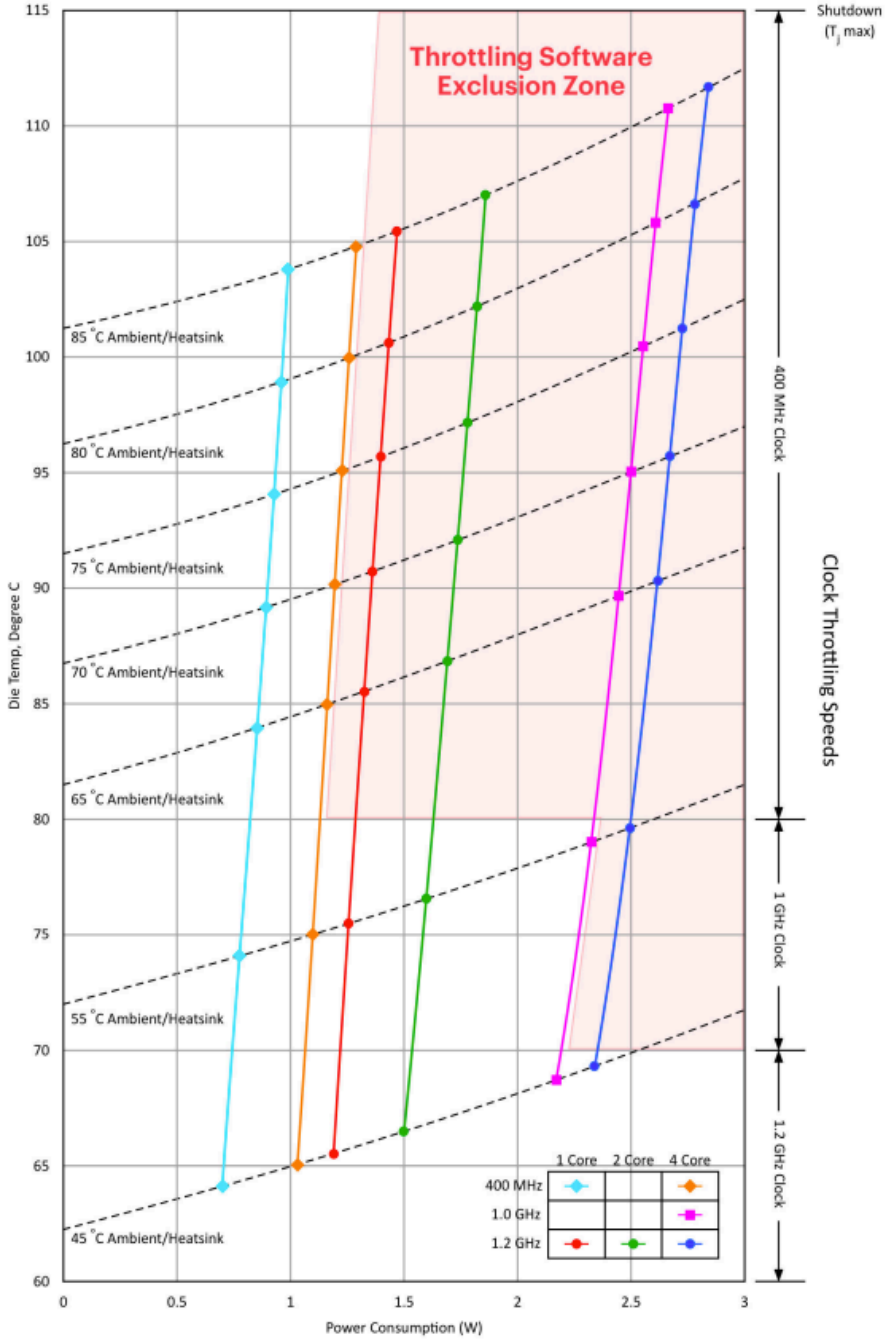
- Disable unused GPIO using user space from sysfs
  - echo 124 > / sys/class/gpio/export
  - echo out > /sys/class/gpio/gpio124/direction
  - echo 0 > /sys/class/gpio/gpio124/value

  - If the GPIO configuration is to be used or reused
    - echo 124 /sys/class/gpio/gpio124/unexport

Performance Tuning – Thermal Threshold

Calculated Die Temperature	Maximum Frequency
<70 ℃	1.2 GHz
<70 ℃ to 80 ℃	1.0 GHz
>80 ℃	400 MHz
115 ℃ (T max)	Shutdown

Performance Tuning – Thermal Threshold



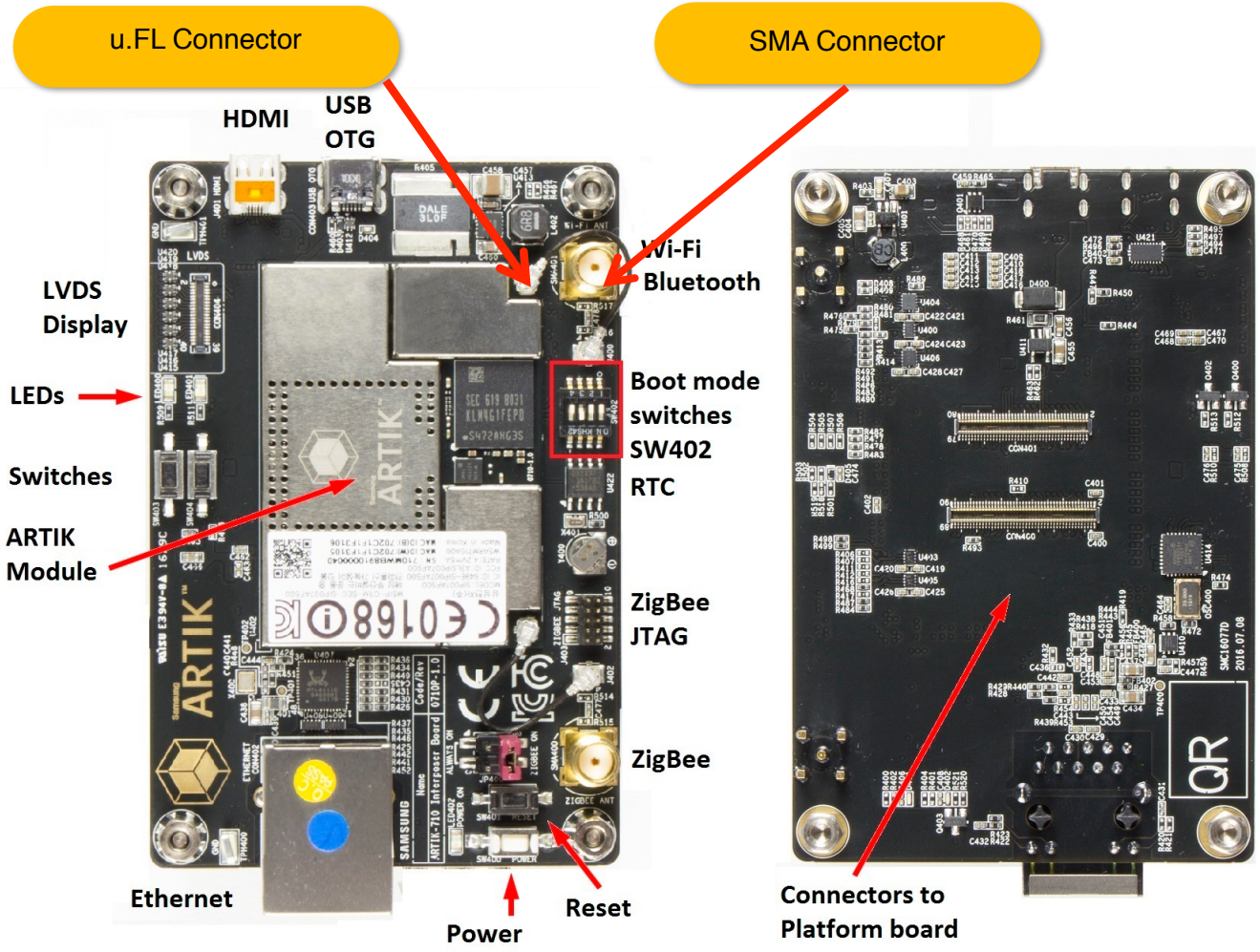
Managing Factory partition

- Save serial number and Manufacturer specific information in Factory Partition
- Factory partition can be updated using serial port or USB
- Update using fastboot in Assembly line
- artik530# run factory_load
  artik530# factory_info list
  artik530# factory_info write {variable} {value} artik710# run factory_save
  artik530# reset
- Pass value from factory partition to linux kernel on bootup

Factory Programming

- How can the ARTIK modules programmed in Assembly line ?
  - USB Fastboot
  - SD boot
  - ~~JTAG~~

FCC Certification and Antenna Tunning



Interactive exercise

APPENDIX