



## Tizen on Anchor3 (Part I)

Hyobok Ahn

DIGNSYS Inc.

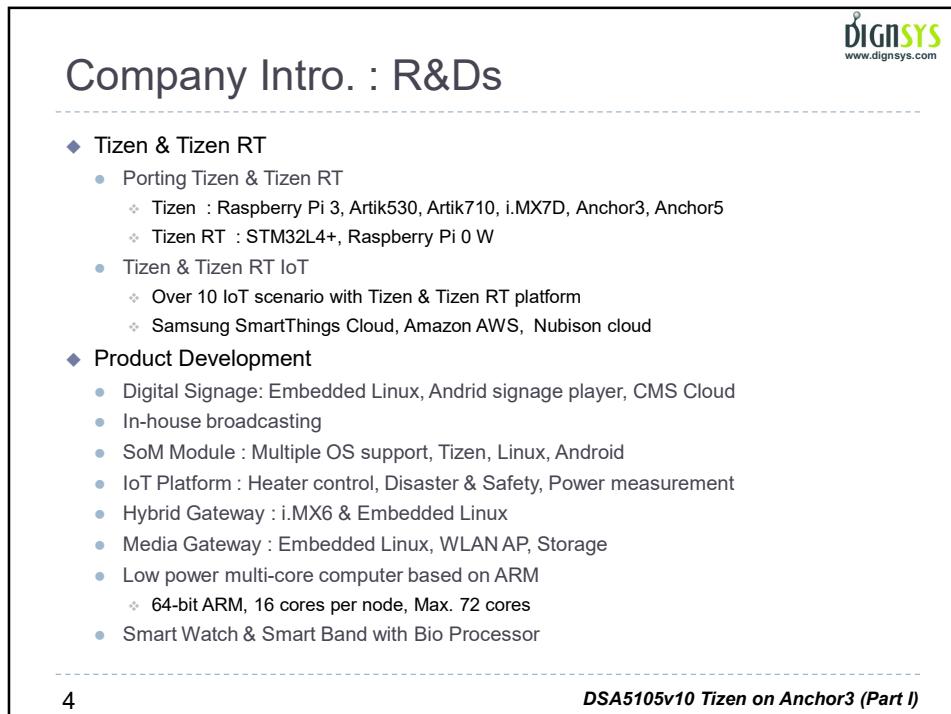
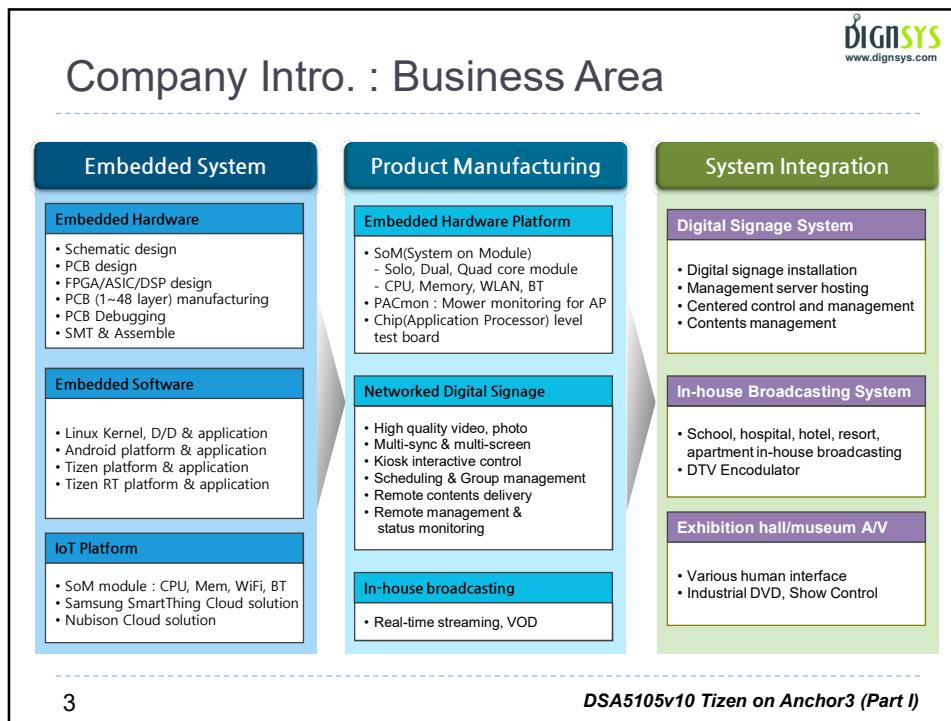
[www.dignsys.com](http://www.dignsys.com)

☎ +82-31-303-5720 fax : +82-31-303-5722 ✉ [hbahn@dignsys.com](mailto:hbahn@dignsys.com)

## Speaker

- ◆ Speaker : Hyobok Ahn
  - CEO, DIGNSYS Inc.
  - E-mail : [hbahn@dignsys.com](mailto:hbahn@dignsys.com)
- ◆ Technology
  - Linux kernel & Device Driver development
  - Tizen Specialist
    - ◊ Share experiences and skill in Tizen development
    - ◊ Specialized in Tizen & Tizen RT porting to custom hardware
    - ◊ Tizen Compatible Hardware & Software development
- ◆ Carrier
  - DIGNSYS Inc.
    - ◊ Embedded System, Linux Platform, Tizen IoT Platform, and Media Player(Signage Player)
  - Samsung Semiconductor, System LSI
    - ◊ Network device, ARM SoC
  - Daewoo Telecom
    - ◊ Mainframe Computer (UNIX Machine)
- ◆ Book
  - ARM and Embedded system, Hanvit media, 2006
  - ARM and Embedded Linux system, Hanvit, 2013





## Contents



### ◆ Course Introduction

- ◆ Tizen & Tizen enabled Hardware Platform
- ◆ Anchor3 Hardware Platform
- ◆ Tizen OS Development Environment
- ◆ Tizen Application Development Environment

## Training Schedule (1)



- ◆ Course Introduction
- ◆ Tizen & Tizen enabled Hardware Platform
  - Tizen OS
  - Tizen Enabled Hardware Platform
- ◆ Anchor3 Hardware Platform
  - Anchor3 Inside
  - How to Anchor3
- ◆ Tizen OS Development Environment
  - Setup Environment
  - Tizen Packages
- ◆ Tizen Application Development Environment
  - Install Tizen Studio
  - Practice Hello Application
- ◆ Tizen Device HAL and Peripheral I/O
  - Tizen Peripheral I/O
  - Tizen Connectivity

## Training Schedule (2)

- ◆ Servo Motor Control
  - Servo Motor and PWM
  - PWM Peripheral I/O
  - Practice PWM Motor Control
- ◆ Gyroscope Sensor
  - Gyroscope Introduction
  - SPI Peripheral I/O
  - Practice Gyroscope
- ◆ Voice Recognition Module
  - V3 Voice Recognition Module
  - UART Peripheral I/O
  - Practice Voice Recognition
- ◆ System Integration
  - Interwork between PWM and Voice Recognition

## Training Environment

- ◆ Host System
  - IBM Compatible, Pentium or more upgraded versions
  - OS : Windows 7 or higher or Ubuntu 16.04 or later
  - Adobe Acrobat Reader 5.0 or later versions
  - Tizen Studio 3.3 or later
- ◆ H/W Platform for Training
  - Anchor3 SoM Based IoT Platform
  - USB Type-C cable : Power & SDB interface (mandatory)
  - Micro USB Type-B cable : Console (optional)
  - USB to serial module : USB to TTL UART
  - Sensors :
    - ❖ Servo motor : PWM interface
    - ❖ MPU-9250 triple-axis MEMS gyroscope : SPI or I2C interface
    - ❖ ELECHOUSE V3.x Voice Recognition Module : UART interface

## Contents

- ◆ Course Introduction
- ◆ **Tizen & Tizen enabled Hardware Platform**
  - Tizen OS
  - Tizen Enabled Hardware Platform
- ◆ Anchor3 Hardware Platform
- ◆ Tizen OS Development Environment
- ◆ Tizen Application Development Environment

## Tizen Evolution



**Tizen Open Source Platform**

Hosted by Linux Foundation  
Easy to develop new features or customize for own device  
Flexible structure of covering Low-end to High-end device

**100 millions Tizen devices in the market**



11 *DSA5105v10 Tizen on Anchor3 (Part I)*

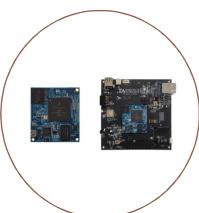
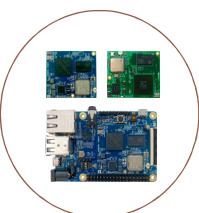
**Contents**

- ◆ Course Introduction
- ◆ **Tizen & Tizen enabled Hardware Platform**
  - Tizen OS
  - **Tizen Enabled Hardware Platform**
- ◆ Anchor3 Hardware Platform
- ◆ Tizen OS Development Environment
- ◆ Tizen Application Development Environment

12 *DSA5105v10 Tizen on Anchor3 (Part I)*

**Tizen Enabled Open Hardware platform**

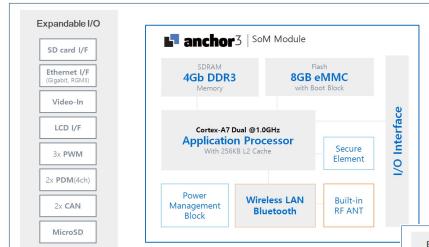
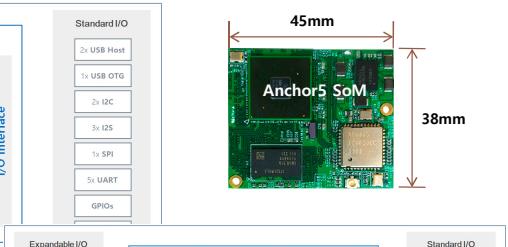
DIGNSYS  
www.dignsys.com

 <p>Raspberry Pi</p>	 <p>SDT</p>	 <p>Anchor</p>
<p><b>SoM Module</b></p> <ul style="list-style-type: none"> <li>- SDTA9X4 : Quad Core</li> <li>- SDTA53X8 : Octa Core</li> </ul>		
<p><b>SoM Module</b></p> <ul style="list-style-type: none"> <li>- Anchor3 : Dual Core</li> <li>- Anchor5 : Quad Core</li> <li>- Anchor7 : Octa Core</li> </ul>		

13      DSA5105v10 Tizen on Anchor3 (Part I)

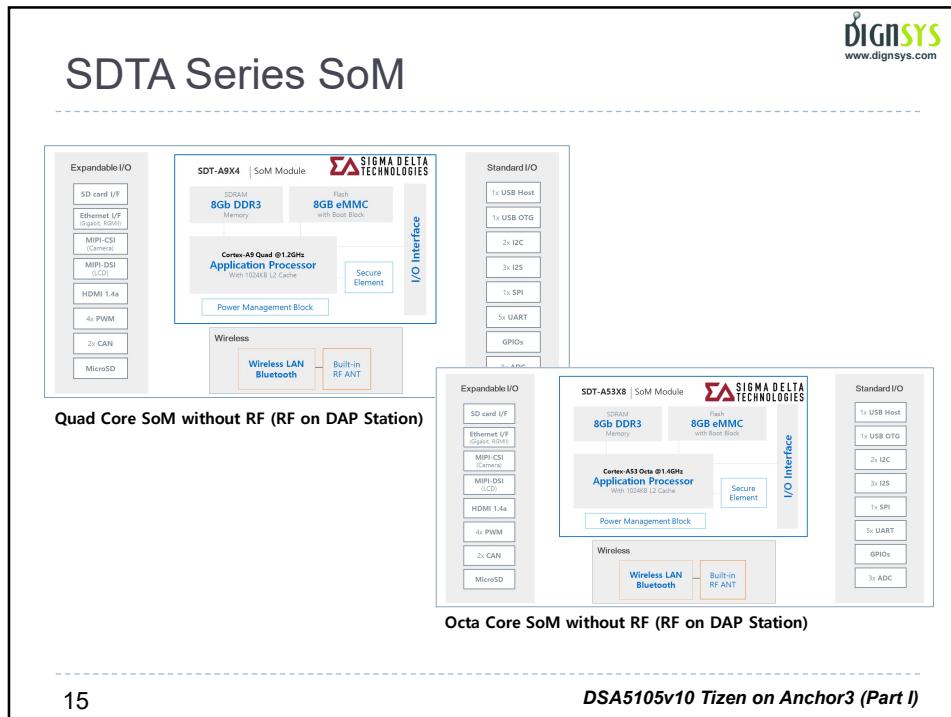
**Anchor Series SoM**

DIGNSYS  
www.dignsys.com

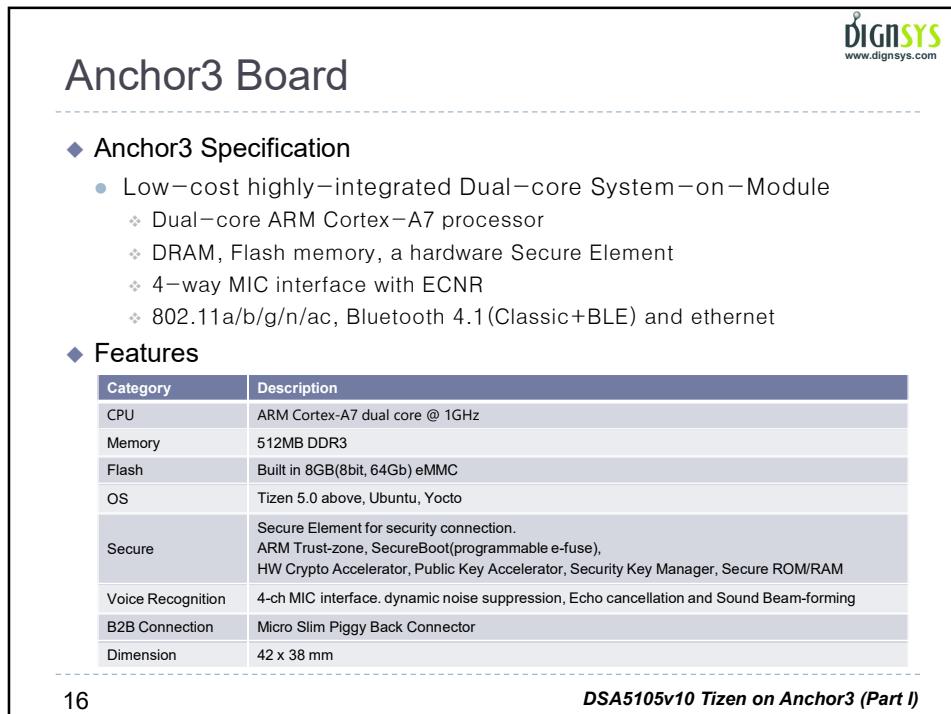
 <p><b>Anchor3 SoM</b></p> <p><b>Dual Core SoM with RF</b></p> <p>42mm      38mm</p>	 <p><b>Anchor5 SoM</b></p> <p>45mm      38mm</p>
<p><b>Standard I/O</b></p> <ul style="list-style-type: none"> <li>2x USB Host</li> <li>1x USB OTG</li> <li>2x I2C</li> <li>3x I2S</li> <li>1x SPI</li> <li>5x UART</li> <li>GPIOs</li> </ul>	
<p><b>Expandable I/O</b></p> <ul style="list-style-type: none"> <li>SD card I/F</li> <li>Ethernet I/F (original ROM)</li> <li>Video-In</li> <li>LCD I/F</li> <li>3x PWM</li> <li>2x PDM(lch)</li> <li>2x CAN</li> <li>MicroSD</li> </ul>	
<p><b>I/O Interface</b></p>	
<p><b>Power Management Block</b></p>	
<p><b>Wireless LAN</b></p>	
<p><b>Built-in RF ANT</b></p>	
<p><b>Cortex-A7 Dual @1.0GHz Application Processor</b></p> <p>With 256KB L2 Cache</p>	
<p><b>Secure Element</b></p>	
<p><b>SDRAM</b></p> <p><b>4Gb DDR3</b></p>	
<p><b>Flash</b></p> <p><b>8GB eMMC</b></p>	

14      DSA5105v10 Tizen on Anchor3 (Part I)

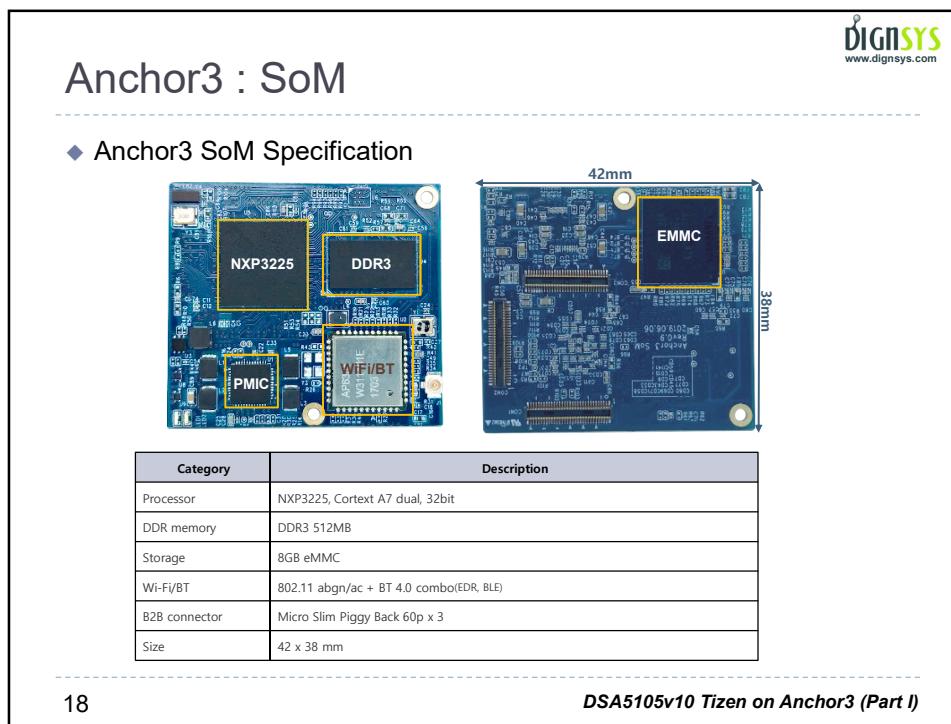
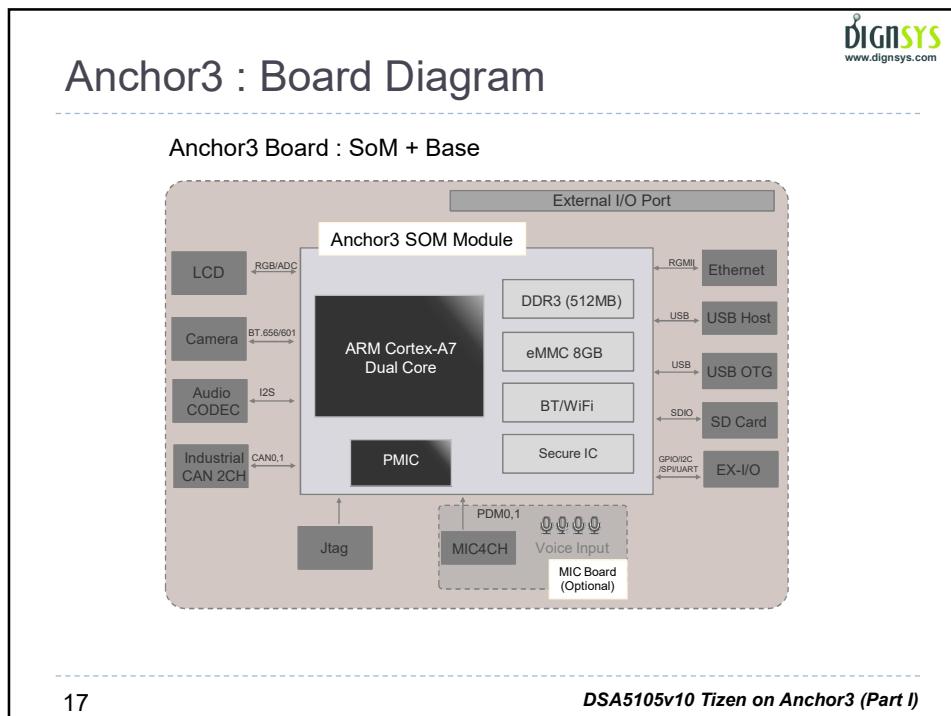
## DSA5105. Tizen with Anchor3 (Part1)



15

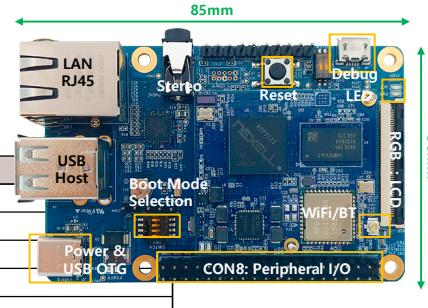


16



## Anchor3 : Base Board

### ◆ Anchor3 Base Specification



Category	Description
Ethernet	1000Mbps Ethernet
Display	RGB24, 800x480
Audio In/Out put	Stereo In/Out phone jack
MIC input	4 PDM MIC input
USB Host	USB2.0 Host x2
USB Device	USB2.0 OTG, Type-C connector, Debug & Power Input
SDCARD	SD card slot
Buttons	DIP Switches, LEDs, Buttons, Jumpers
External I/O Port	GPIO, SPI, UART, I2C, I2S, PWM, ADC
Power input	USB Type-C power
Size	85 x 56 mm

## Anchor3 : Connectivity & IOs

Category	Description
WLAN	IEEE 802.11a/b/g/n/ac
Bluetooth	V4.0 + EDR and BLE support
Ethernet	10/100 /1000 Base-T Ethernet
USB Host	2 x Host USB
USB Device	OTG, Debug
UART	Support Max 4xUART
SDIO	Support 1x SDIO for SD Memory Card
I2C	1x I2C
SPI	1x SPI
Display	800x480 @24bpp RGB LCD
Video In	Digital Video In for Camera
PDM	Max 4-ch(2-ch exclusive LCD)
Audio In	2x I2S Audio In
Audio Out	1x Audio Out
ADC	2xADC
PWM	2xPWM
GPIOs	3x GPIO
LEDs	Power Status, Act
Buttons	Reset

## Anchor3 : Connectivity & IOs

The diagram shows the Anchor3 Board with its pins numbered 1 through 40. A legend on the right maps each pin number to its function:

1	2	DC Power 3.3V
3	4	I2C-SDA1   GPB29
5	6	I2C-SCL1   GPB30
7	8	GPI17(EX_INT)
9	10	GND
11	12	GPD26   UART6_TX
13	14	GPD27   UART6_RX
15	16	GPA16(I2S2_BCLK)
17	18	DC Power 3.3V
19	20	SPI_MOSI   GPB31
21	22	SPI_MISO   GPC0
23	24	SPI_CLK   GPC3
25	26	GND
27	28	AIN0
29	30	GPD22(UART5_TX)
31	32	GPD23(UART5_RX)
33	34	GPD24(UART_DE)
35	36	GPA18(I2S2_LRCLK)
37	38	GPD25(UART5_RE)
39	40	GND
		5V DC Power
		5V DC Power
		GND
		GPD13   UART0_TX
		GPD11   UART0_RX
		GPC1   PWM0
		GND
		GPD12   UART1_TX
		GPD13   UART1_RX
		GND
		GPC8
		GPC4   SPI_SS
		GPC13
		AIN1
		GND
		GPA17   I2S2_MCLK
		GND
		GPC2   PWM1
		GPA15   I2S2_DI
		GPA14   I2S2_DO

21 DSA5105v10 Tizen on Anchor3 (Part I)

## Anchor3 : PDM MEMS Microphone

The diagram illustrates the connection between the Anchor3 Board and a PDM MEMS Microphone Board. The Anchor3 Board is connected to the Microphone Board via a ribbon cable. The Microphone Board has four microphones (Mic1, Mic2, Mic3, Mic4) connected to it. The connections are summarized in the following table:

Anchor3 Pin	Microphone Board Pin
3.3V	3.3V
PDMD0	-Strobe
-Strobe	PDMD0
PDMD1	PDMD1
GND	GND

4xPDM Mic

3.3V  
PDMD0  
-Strobe  
PDMD1  
GND

22 DSA5105v10 Tizen on Anchor3 (Part I)

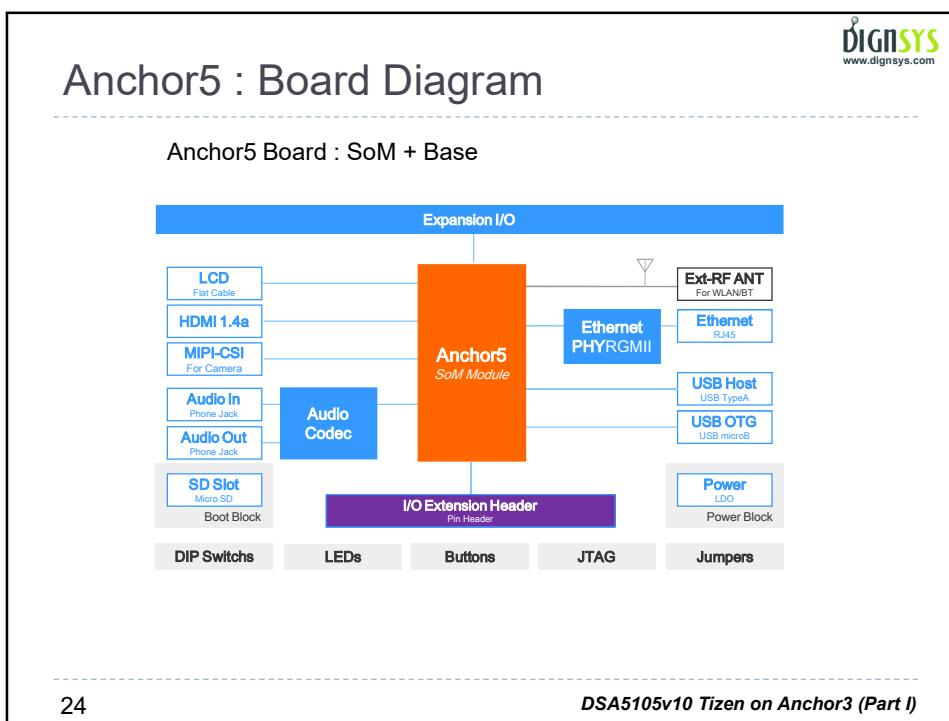
**Anchor5 Board**

**DIGNSYS**  
www.dignsys.com

- ◆ **Anchor5 Specification**
  - Low-cost highly-integrated Quad-core System-on-Module
    - ❖ Quad-core ARM Cortex-A9 processor
    - ❖ DRAM, Flash memory, a hardware Secure Element
    - ❖ 802.11a/b/g/n/ac, Bluetooth 4.1(Classic+BLE) and ethernet
- ◆ **Features**

Category	Description
CPU	ARM Cortex-A9 Quad core @ 1.4 GHz
Memory	1GB DDR3 @ 800MHz
Flash	Built in 4GB(8bit, 32Gb) eMMC v4.5
OS	Tizen 5.0 above, Ubuntu Linux
Secure	Secure Element for point-to-point authentication and data transfer ARM Trust-zone, Secure Boot, Secure JTAG
PMIC	Provide all power of Anchor5 Module using onboard bucks and LDOs
B2B Connection	Micro Slim Piggy Back Connector
Dimension	42mm x 38mm

23 **DSA5105v10 Tizen on Anchor3 (Part I)**



**Anchor5 : SoM**

◆ Anchor5 SoM Specification

Category	Description
Processor	NXP i.MX6UL, Cortex A9 Quad, 32bit
DDR memory	DDR3 1GB
Storage	8GB eMMC
Wi-Fi/BT	802.11 abgn/ac + BT 4.0 combo(EDR, BLE)
B2B connector	Micro Slim Piggy Back
Size	42mm x 38mm

25 **DSA5105v10 Tizen on Anchor3 (Part I)**

**Anchor5 : Base Board**

◆ Anchor5 Base Specification

Category	Description
Ethernet	1000Mbps Ethernet
Display	HDMI1.4a, MIPI-DSI, LVDS
Audio In/Out put	Stereo In/Out phone jack
Camera	MIPI-CSI
USB Host	USB2.0 Host x2
USB Device	USB2.0 OTG, Type-C connector, Debug
SDCARD	SD card slot
Buttons	DIP Switches, LEDs, Buttons, Jumpers
External I/O Port	GPIO, SPI, UART, I2C, I2S, PWM, ADC
Power input	USB Type-C power
Size	85 x 56 mm

26 **DSA5105v10 Tizen on Anchor3 (Part I)**

**Anchor5 : Connectivity & IOs**

DIGNSYS  
www.dignsys.com

Category	Description
WLAN	IEEE 802.11a/b/g/n/ac
Bluetooth	V4.1 + EDR and BLE support
Ethernet	10/100 /1000 Base-T Ethernet
USB Host	2 x Host USB
USB Device	OTG, Debug
UART	Support Max 4xUART
SDIO	Support 1x SDIO for SD Memory Card
I2C	1x I2C
SPI	1x SPI
Display	4-lane MIPI-DSI up to 1920x1280, HDMI 1.4a up to 1920x1080@60fps LVDS up to 1280x720p@60fps
Camera	MIPI-CSI up to 5M (1920x1080@30fps)
Audio In	2x I2S Audio In
Audio Out	1x Audio Out
ADC	3xADC
PWM	4xPWM
GPIOs	Xx GPIO (TBD)
LEDs	Power Status, X X LED (TBD)
Buttons	Power / Reset
WLAN	IEEE 802.11a/b/g/n/ac

27 DSA5105v10 Tizen on Anchor3 (Part I)

**Anchor5 : Use Case**

DIGNSYS  
www.dignsys.com

SmartThings

RMS (Remote Maintenance System)

Internet

Home Appliances Control

Voice Control by Bixby

Mobile Plug-in App

Massage Chair Control Tablet with Tizen & SmartThings

TIZEN

Control Agent

UI Control Application

HW Control Interface Agent

Cloud Adapter

RMS Adapter

Massage Control Board

Cozyma Massage Chair

28 DSA5105v10 Tizen on Anchor3 (Part I)

### Anchor5 : How to write application ?

The diagram illustrates the MVVM (Model-View-ViewModel) architecture for Anchor5. It is divided into three main logical layers:

- UI Logic**: Contains the **View (UI)** component, which is implemented using XAML (Window, ContentPage).
- Presentation Logic**: Contains the **ViewModel** component, which is responsible for Binding, Command, and Notify.
- Business Logic**: Contains the **Model** component, which manages Entities.

The components interact via the following mechanisms:

- View (UI)** interacts with **ViewModel** through **Data Binding** and **Commands**.
- ViewModel** interacts with **Model** through **Update** and **Notification**.
- View (UI)** receives **Notification** from **Model**.

Below the architecture diagram, four screenshots of the Tizen application interface are shown:

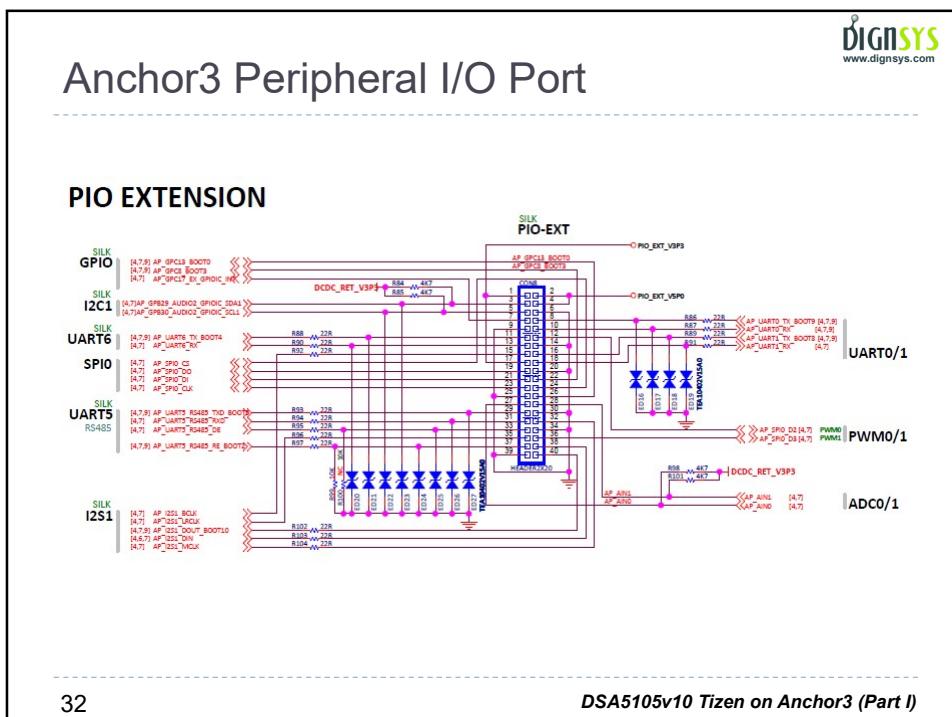
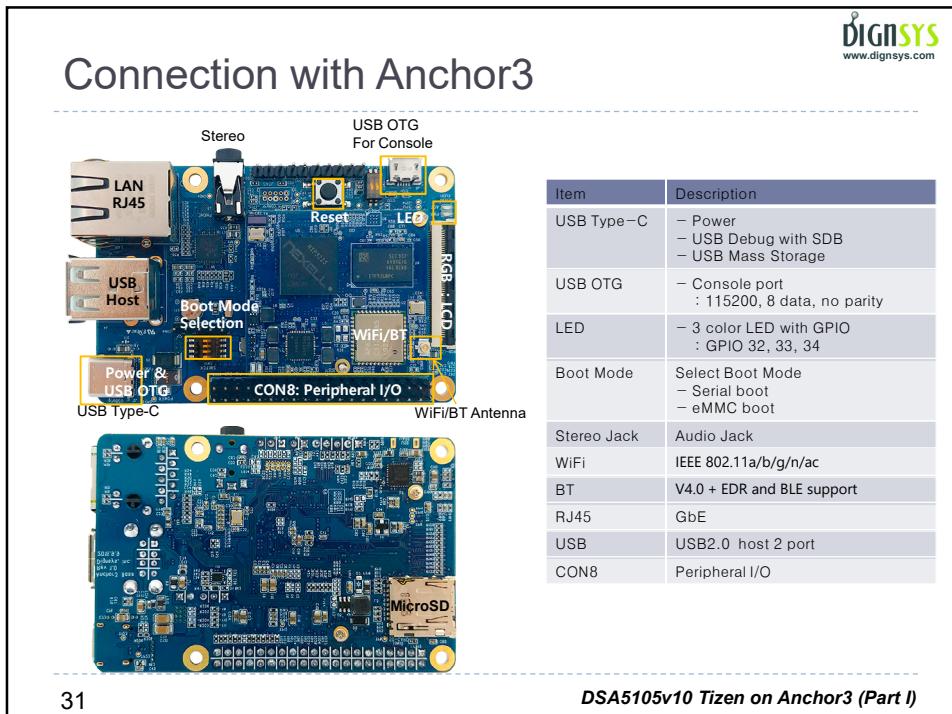
- Home UI**: Shows a grid of icons for Home, Games, and Music.
- Home with Status**: Shows the same grid with additional status indicators.
- Status UI**: Shows a detailed status screen with a 3D character and various status bars.
- Setting UI**: Shows a settings screen with options like Language, Brightness, Volume, and Screen Off.

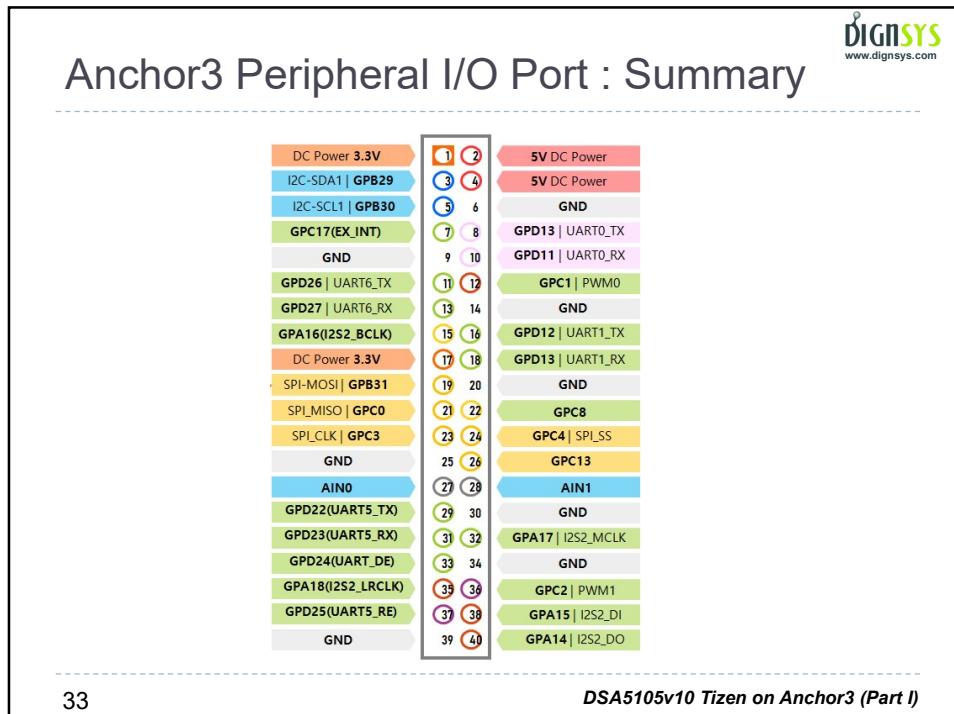
29 DSA5105v10 Tizen on Anchor3 (Part I)

### Contents

- ◆ Course Introduction
- ◆ Tizen & Tizen enabled Hardware Platform
- ◆ **Anchor3 Hardware Platform**
  - Anchor3 Inside
  - How to Anchor3
- ◆ Tizen OS Development Environment
- ◆ Tizen Application Development Environment

30 DSA5105v10 Tizen on Anchor3 (Part I)





33

DIGNSYS  
www.dignsys.com

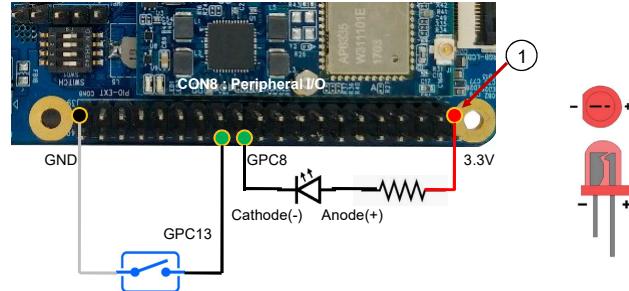
**Anchor3 Peripheral I/O Access**

Peripheral I/O	NXP3225		Anchor3 Board		remark
	Signal Name	Anchor3 Signal	Anchor3 CON8	Tizen Port Map	
SE	GPIO-I2C1	GPIO-I2C1	X	I2C-6	
	GPB24	AP_SECURE_FLOW	X	GPIO56	
BT	UART4	UART4	X	ttyS4	
CONSOLE	UART2	UART2	X	ttyS2	
GPIO	GPC8	GPC8	22	GPIO72	TCT
	GPC13	GPC13	26	GPIO77	
	GPC17	GPC17	7	GPIO81	
	GPA14	I2S2_DO	40	GPIO14	default : I2S
	GPA15	I2S2_DI	38	GPIO15	default : I2S
	GPA16	I2S2_BCLK	15	GPIO16	default : I2S
	GPA17	I2S2_MCLK	32	GPIO17	default : I2S
	GPA18	I2S2_LRCLK	35	GPIO18	default : I2S
	I2C	GPIO-I2C1	SDA(3), SCL(5)	I2C-6	TCT
PWM	PWM0	AP_SPI0_D2	12	pwmchip0/pwm0	TCT
	PWM1	AP_SPI0_D3	36	pwmchip0/pwm1	
SPI	SPI0	SPI0	CS(24), CLK(23), DI(21), DO(19)	spidev0.0	TCT
UART	UART0	UART0	TX(8), RX(10)	ttyS0	TCT
	UART1	UART1	TX(16), RX(18)	ttyS1	
	UART5_RS485	UART5	TX(29), RX(31), DE(33), RE(37)	ttyS5	
	UART6	UART6	TX(11), RX(13)	ttyS6	
ADC	AIN0	AIN0	27		
	AIN0	AIN1	28		
PWR	3.3V	3.3V	1,17		
	5V	5V	2,4		
	GND	GND	6,9,14,20,25,30,34,39		

DSA5105v10 Tizen on Anchor3 (Part I)

34

## GPIOs



linux gpio number = (gpio\_bank - 1) \* 32 + gpio\_bit

Peri-I/O	NXP3225	GPIO #	CON8 Pin	USER LED
GPIO 72	GPC8	72	22	LED
GPIO 77	GPC13	77	26	Switch
GPIO 81	GPC17	81	7	
3.3V			1	
GND			39	

35

DSA5105v10 Tizen on Anchor3 (Part I)

## PWM

### ◆ PWM(Pulse Width Modulation)

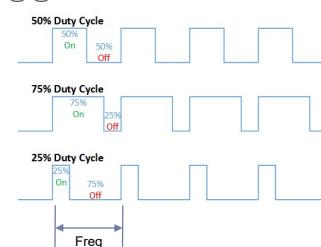
- Digital 신호를 이용하여 Analog와 유사한 Pulse 파형은 만들어 주는 장치
- 주파수(Frequency)와 Duty Cycle로 파형 생성

### ◆ PWM 응용

- Motor 회전 속도 제어
- LED 밝기 제어

### ◆ Duty Cycle

- 전체 주파수에서 high 인 구간이 차지하는 비율



36

DSA5105v10 Tizen on Anchor3 (Part I)

## Servo Motor TowerPro SG90

The diagram shows the connection between an Arduino Uno and a SG90 servo motor. The Arduino's CON8 Peripheral I/O pins are connected to the servo's power and ground pins. Pin 12 (PWM0) is connected to the servo's PWM pin. A detailed inset shows the PWM signal waveform: a square wave with a 1-2 ms duty cycle, operating at a 20 ms (50 Hz) period. The VCC 5V pin on the Arduino is connected to the servo's power source.

CON8 Pin#	Anchor3	Peri-I/O
12	PWM0	PWMCHIP0/ PWM0
2	VCC 5V	
30	GND	

37 DSA5105v10 Tizen on Anchor3 (Part I)

## I2C

- ◆ I2C(Inter-Integrated Circuit) 버스
  - 여러 개의 저속의 주변 기기를 연결하기 위해 사용
- ◆ 2 개의 라인을 이용한 I2C 통신
  - ◊ SDA (Serial DAta)
    - 데이터의 직렬 전송에 사용된다.
  - ◊ SCL (Serial CLock)
    - 디바이스간 신호 동기화에 사용되는 클럭이다.
- ◆ Address & Data 전송
  - 7비트 Address와 R/W 비트를 포함하여 8비트 전송 후 Data 송수신

The timing diagram illustrates the sequence of events in an I2C transaction. It shows the SDA (Serial DAta) and SCL (Serial CLock) lines. The process starts with a START condition (S), followed by an ADDRESS field (7 bits), an R/W bit (1 bit), and an ACK (Acknowledge) signal. This is followed by three DATA fields (8 bits each) and three more ACK signals. The transaction concludes with a STOP condition (P).

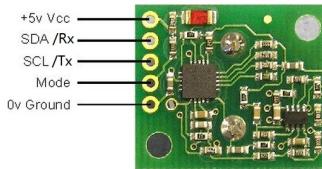
38 DSA5105v10 Tizen on Anchor3 (Part I)

## Ultrasonic Range Finder, SRF02

DIGNSYS  
www.dignsys.com

### ◆ Specification

- Range: 16cm to 6m.
- Power: 5v, 4mA Typ.
- Frequency: 40KHz.
- Analogue Gain
  - ❖ Automatic 64 step gain control
- Connection Modes:
  - ❖ 1 - Standard I2C Bus
  - ❖ 2 - Serial Bus(connects up to 16 devices to any uP or UART serial port)
- Full Automatic Tuning: No calibration, just power up and go
- Timing: Fully timed echo, freeing host controller of task.
- Units: Range reported in uS, mm or inches.

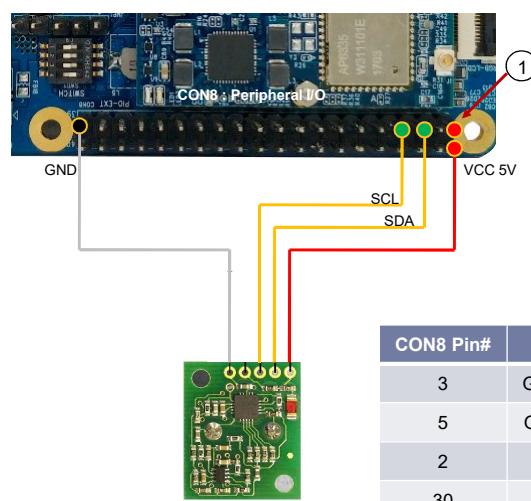


39

DSA5105v10 Tizen on Anchor3 (Part I)

## I2C bus with Range Finder

DIGNSYS  
www.dignsys.com



40

DSA5105v10 Tizen on Anchor3 (Part I)

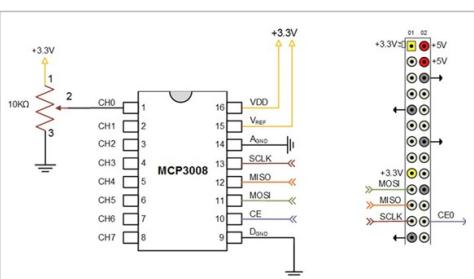
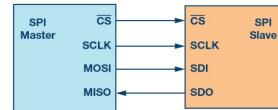
## SPI

- Serial peripheral interface (SPI)

- one of the most widely used interfaces between microcontroller and peripheral ICs such as sensors, ADCs, DACs, and others.

- Microchip MCP3008

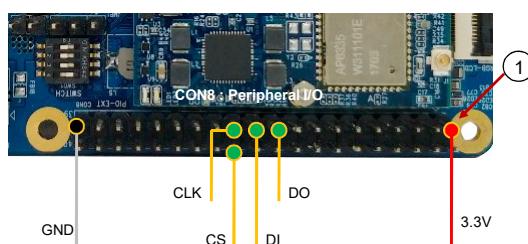
- 10-bit ADC with SPI



41

DSA5105v10 Tizen on Anchor3 (Part I)

## SPI with MCP3008



CH0	1	V <sub>DD</sub>
CH1	2	V <sub>REF</sub>
CH2	3	AGND
CH3	4	CLK
CH4	5	D <sub>OUT</sub>
CH5	6	D <sub>IN</sub>
CH6	7	CS/SHDN
CH7	8	DGND

CON8 Pin#	Anchor3	MCP3008/PIN	Peri-I/O
24	SPI0_CS	_CS/10	SPI0
23	SPI0_CLK	CLK/13	SPI0
21	SPI0_DI(MISO)	DO/12	SPI0
19	SPI0_DO(MOSI)	DI/11	SPI0
1	3.3V	15,16	
30	GND	9	

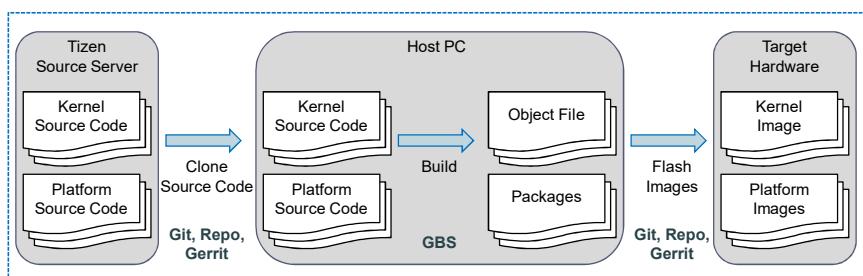
42

DSA5105v10 Tizen on Anchor3 (Part I)

## Contents

- ◆ Course Introduction
- ◆ Tizen & Tizen enabled Hardware Platform
- ◆ Anchor3 Hardware Platform
- ◆ **Tizen OS Development Environment**
  - [Setup Environment](#)
  - Tizen Packages
- ◆ Tizen Application Development Environment

## Tizen Build Tool



- ◆ Source Code Management Tools : before build
  - Git, Repo, Gerrit
- ◆ Build Tool
  - GBS(Git Build System)
- ◆ Packaging Tools : after build
  - MIC, RPM

## Contents

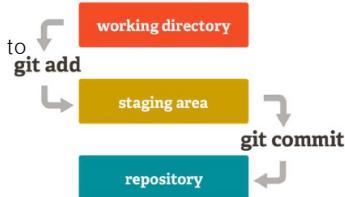
- ◆ Course Introduction
- ◆ Tizen & Tizen enabled Hardware Platform
- ◆ Anchor3 Hardware Platform
- ◆ **Tizen OS Development Environment**
  - Setup Environment
    - ◊ Source Code Management(SCM) : before build
    - ◊ Build Tool
    - ◊ Packaging Tool : after build
  - Tizen Packages
- ◆ Tizen Application Development Environment

45

DSA5105v10 Tizen on Anchor3 (Part I)

## Git : About

- ◆ A revision control and source management tool
  - Free & open source (GPL v2)
  - Branching & merging
  - Smaller & faster than other SCM tools
- ◆ Key Concept
  - Working directory : include working file
    - command “add” : add file & directory to staging area
  - Staging area(index) : file to commit
    - Command “commit” : apply staging area to repository
  - Repository(HEAD) : file that committed



46

DSA5105v10 Tizen on Anchor3 (Part I)

### Git : Command (1)

- ◆ Install

```
$ sudo apt-get install git
```

- ◆ New repository create (init)

- Create new repository on working directory

```
$ git init
```

- Create repository without Working directory

```
$ git init --bare
```

- ◆ Copy repository

- syntax

```
git clone protocol://user@host:path
```

- example

```
$ git clone ssh://user@review.tizen.org:external/systemd
```

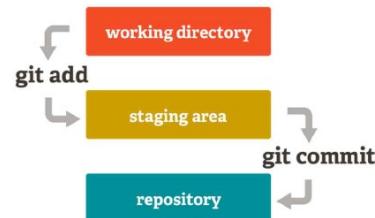
### Git : Command (2)

- ◆ Add source to staging area

```
git add <file name>
```

- example

```
$ git add *
```



- ◆ Commit staging area

```
$ git commit -m "Description about this version"  
$ git commit -a -m "Description about this version"
```

- '-a' option : add file to staging area



## Git : Command (3)

### ◆ Push local commit to remote repository

```
git push <remote repository name> <remote branch name>
```

- Default name of *remote repository* : origin
- Default name of *remote branch* : master
- example

```
$ git push origin master
```

### ◆ Add remote repository address(URI)

```
$ git remote add <remote repository name> <remote repository name>
```

- example

```
$ git remote add origin ssh://user@review.tizen.org:external/systemd
```

## Git : Command (4)

### ◆ Branch & Merge

- *Branch* create new working branch

- ❖ Create new branch

```
git branch <branch name>
```

- ❖ Default Branch : master

- *Merge* branch after finish development

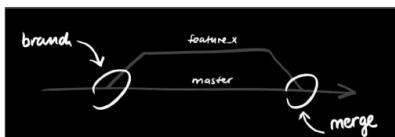
- ❖ Branch merge

```
git merge <branch name>
```

- ❖ Merge example : merge feature\_x to master branch

```
$ git checkout master
```

```
$ git merge feature_x
```



### ◆ Commit log

```
git log
```

- Branch list

```
git branch --list
```

- remote branch list

```
git branch --remotes
```

- Move to other branch (checkout)

```
git checkout <branch name>
```

### Repo

- ◆ Repository management tool that built on top of Git
  - Used for managing many git repositories
  - Automate parts of development workflow

- ◆ Installation

```
$ sudo apt-get install curl
$ mkdir -p ~/bin
$ curl http://commanddatastorage.googleapis.com/git-repodownloads/repo > ~/bin/repo
$ chmod +x ~/bin/repo
```

- Setup PATH environment (~/.bashrc)

```
PATH=~/bin:${PATH}
```

- ◆ Repo Command

- Initialize repo directory

```
repo init -u <URL> [OPTIONS]
```

- Synchronize repo directory

```
repo sync [OPTIONS]
```

### Contents

- ◆ Course Introduction
- ◆ Tizen & Tizen enabled Hardware Platform
- ◆ Anchor3 Hardware Platform
- ◆ **Tizen OS Development Environment**
  - Setup Environment
    - ❖ Source Code Management(SCM) : before build
    - ❖ Build Tool
    - ❖ Packaging Tool : after build
  - Tizen Packages
- ◆ Tizen Application Development Environment

**Gerrit & GBS(Git Build System)**

**◆ Gerrit**

- A Web-based code review system which checks the platform source code
- Tizen Gerrit : <http://review.tizen.org/gerrit>

**◆ GBS**

- A developer command line tool that supports Tizen development
- Features
  - ❖ Build a repository or repositories and make package files(\*.rpm)
  - ❖ Generate tarballs based on Git repositories
  - ❖ Do local test builds

53 **DSA5105v10 Tizen on Anchor3 (Part I)**

**Gerrit : Setting GBS Build**

**◆ Login to Tizen gerrit & go to Settings**

- Login site information : <http://review.tizen.org/gerrit>

Status	Algorithm	Key	Comment
<input type="checkbox"/>	ssh-rsa	AAAAB3NzaC1yc2EAAAQABAAQ...QicCTW2+3	hbahn@hbahn-linuxpc

54 **DSA5105v10 Tizen on Anchor3 (Part I)**

**Gerrit : Register SSH Key**

◆ Create & register SSH

```
$ ssh-keygen
```

- Key created on “~/.ssh” directory

Settings

Status	Algorithm	Key	Comment
<input type="checkbox"/>	ssh-rsa	AAAAB3NzaC1yc2EAAAQABAAQ...QiiCCTW2+3	hbahn@hbahn-linuxpc

Add SSH Public Key  
▶ How to Generate an SSH Key

Add new SSH public Key

Clear Add Close

55 **DSA5105v10 Tizen on Anchor3 (Part I)**

**Gerrit : SSH Setup & Connection Test**

◆ Setup SSH config file

- “~/.ssh/config” : specify ID with yours

```
Host tizen review.tizen.org
Hostname review.tizen.org
IdentityFile ~/.ssh/id_rsa
User hbahn
Port 29418
```

◆ SSH Connection test

```
hbahn@hbahn-linuxpc:~$ ssh tizen
**** Welcome to Gerrit Code Review ****
Hi Hyobok Ahn, you have successfully connected over SSH.

Unfortunately, interactive shells are disabled.
To clone a hosted Git repository, use:

git clone ssh://hbahn@review.tizen.org:29418/REPOSITORY_NAME.git

Connection to review.tizen.org closed.
```

56 **DSA5105v10 Tizen on Anchor3 (Part I)**

**GBS : Installation**

◆ Reference  
<https://source.tizen.org/documentation/developer-guide/getting-started-guide/installing-development-tools>

◆ Install Procedure(Ubuntu 16.04)

1. Open the package manager source list using a text editor.  
\$ sudo vim /etc/apt/sources.list
2. Add the Tizen tools repository to the source list.  
deb [trusted=yes] http://download.tizen.org/tools/latest-release/Ubuntu\_16.04/ /
3. Resynchronize the package index files from the sources specified in the source list:  
\$ sudo apt-get update
4. Install a development tool:  
\$ sudo apt-get install gbs mic
5. upgrade a development tool  
\$ sudo apt-get update && sudo apt-get upgrade

57 **DSA5105v10 Tizen on Anchor3 (Part I)**

**GBS : Configuration File**

◆ .gbs.conf : GBS configuration file

- Set Profile name, Working Directory path
- Set Local repository, remote repository path

◆ GBS configuration file for Anchor3 : ~/.gbsAnchor3.conf

```
[general]
#Current profile name which should match a profile section name
profile = profile.tizen_anchor3

[profile.tizen_anchor3]
#Common authentication info for whole profile
obs = obs.tizen

repos = repo.public_base_arm, repo.tizen_anchor3
buildroot = ~/GBS-ANCHOR3-ROOT/tizen_anchor3

[obs.tizen]
#OBS API URL pointing to a remote OBS.
url = https://api.tizen.org

[repo.public_base_arm]
## tizen 5.0
url = http://download.tizen.org/releases/milestone/tizen/base/tizen-base_20190503.1/repos/standard/packages/

[repo.tizen_anchor3]
## tizen 5.0
url = http://download.tizen.org/releases/milestone/tizen/unified/tizen-unified_20190523.1/repos/standard/packages/
```

58 **DSA5105v10 Tizen on Anchor3 (Part I)**

### GBS : Build

#### ◆ GBS syntax

```
gbs build -A <target architecture name>
```

- Build target
  - ❖ the all package on current directory with git repository
    - the build package must “git commit”
- example

```
$ gbs build -A armv7l
$ gbs build -A armv7l --threads=2 --clean-once --includeall --exclude=bash,bzip2-libs,rpm
```
- Target Architecture
  - ❖ x86, armv7l
- Options
  - ❖ --threads <number of thread>: The number of thread to build
  - ❖ --clean-once: initialize GBS build configuration at first build
  - ❖ --include-all: build working directory not git repository
  - ❖ --exclude=<package list>: exclude package list

### GBS : Build Anchor3 Package

#### ◆ Build

```
$ gbs -c ~/.gbsAnchor3.conf build -A armv7l --thread=4
```

- Build target: the all package on current directory with git repository
- Target Architecture : armv7l

#### ◆ RPM packages after build

- Build directory on GBS configuration file (.gbsAnchor3.conf)

```
 ${GBS-BUILD-ROOT}/local/repos/tizen_anchor3/armv7l/RPMS
```

## Contents

- ◆ Course Introduction
- ◆ Tizen & Tizen enabled Hardware Platform
- ◆ Anchor3 Hardware Platform
- ◆ **Tizen OS Development Environment**
  - Setup Environment
    - ❖ Source Code Management(SCM) : before build
    - ❖ Build Tool
    - ❖ Packaging Tool : after build
  - Tizen Packages
- ◆ Tizen Application Development Environment

## MIC(Moblin Image Creator)

- ◆ MIC(Moblin Image Creator)
  - Tool to create an operation system image for a Tizen device
- ◆ Installation
  - Install MIC package

```
$ sudo apt-get install gbs mic
```
- ◆ MIC on Tizen
  - Build boot image
    - ❖ tizen-unified\_iot-boot-armv7l-anchor3.tar.gz
  - Build Tizen Package
    - ❖ Minimal : tizen-unified\_iot-headless-2parts-armv7l-anchor3.tar.gz
    - ❖ Additional : tizen-unified\_iot-headless-3parts-armv7l-anchor3.tar.gz

### MIC : Image Build

#### ◆ Build Boot Image

```
$ sudo mic cr loop tizen-unified_iot-boot-armv7l-anchor3.ks \
    --pack-to=tizen-unified_iot-boot-armv7l-anchor3.tar.gz \
    --logfile=mic_build.log -A armv7l
```

#### ◆ Build Tizen Image

```
$ sudo mic cr loop tizen-unified_iot-headless-2parts-armv7l-anchor3.ks \
    --pack-to=tizen-unified_iot-headless-2parts-armv7l-anchor3.tar.gz \
    --logfile=mic_build.log -A armv7ld.log -A armv7l
```

### RPM

#### ◆ RPM

- Fedora & Red Hat Linux Package Management Tool
- RPM tool can build, install, query, verify, update, erase individual software packages

#### ◆ RPM Packages

- Components
  - ❖ An archive of files
  - ❖ Metadata
    - Used to install and erase the archive files
    - Helper scripts, file attributes, and descriptive information about the package
- Type
  - ❖ Binary package : used to encapsulate software to be installed
  - ❖ Source package : containing the source code and recipe necessary to product binary packages

#### ◆ Reference : <https://rpm.org/documentation.html>

## Contents

- ◆ Course Introduction
- ◆ Tizen & Tizen enabled Hardware Platform
- ◆ Anchor3 Hardware Platform
- ◆ **Tizen OS Development Environment**
  - Setup Environment
  - **Tizen Packages**
    - ❖ **Build Package**
    - ❖ Tizen Partition
    - ❖ Fusing Image
- ◆ Tizen Application Development Environment

## Package for Anchor3

- ◆ Anchor3 Boot Package
  - firmwares-anchor3 : firmwares for Anchor3 target BL1,BL2,BL32
  - u-boot-anchor3 : u-boot source for Anchor3
  - linux-anchor3 : linux kernel for Anchor3
  - tools-anchor3 : tools for Anchor3 build
  - boot-firmwares-anchor3 : binary files for booting Anchor3
- ◆ Anchor3 Tizen Package
  - building-blocks : building-blocks for Anchor Tizen porting
  - systemd : systemd porting for Anchor Tizen
  - device-manager-plugin-anchor : device manager plugin for Anchor Tizen porting
  - peripheral-bus : peripheral bus porting for Anchor Tizen
  - peripheral-io : peripheral IO porting for Anchor Tizen
  - bluetooth-firmware-bcm : Bluetooth firmware for Anchor with BCM4335(AP6335)
  - wlandrv-plugin-tizen-ap6335 : WLAN driver plugin for Anchor Tizen porting with AP6335
  - alsa-ucm-data-alc5623 : Anchor ALSA device data
  - audio-hal-alc5623 : Audio HAL for Anchor with ALC5623 Codec
  - mmfw-syconf : Multimedia firmware for Anchor Tizen porting
  - libtbtm-nexell-anchor3 : TBM library for NXP322x
  - smarthing-plugin : tizen smarthing-plugin project (only for smarthings with iotivity)

## Download Boot Package



### ◆ Download Anchor3 Boot packages from Github

```
$ mkdir ~/Tizen-Work-boot  
$ cd ~/Tizen-Work-boot  
$ git clone https://github.com/dignsys/firmwares-anchor3.git  
$ git clone https://github.com/dignsys/u-boot-anchor3.git  
$ git clone https://github.com/dignsys/linux-anchor3.git  
$ git clone https://github.com/dignsys/tools-anchor3.git  
$ git clone https://github.com/dignsys/boot-firmwares-anchor3.git
```

## Download Tizen Package



### ◆ Download Anchor3 Tizen packages from Github

```
$ mkdir ~/Tizen-Work  
$ cd ~/Tizen-Work  
$ git clone https://github.com/dignsys/building-blocks.git  
$ git clone https://github.com/dignsys/systemd.git  
$ git clone https://github.com/dignsys/device-manager-plugin-anchor.git  
$ git clone https://github.com/dignsys/peripheral-bus.git  
$ git clone https://github.com/dignsys/peripheral-io.git  
$ git clone https://github.com/dignsys/bluetooth-firmware-bcm.git  
$ git clone https://github.com/dignsys/wlandrv-plugin-tizen-ap6335.git  
$ git clone https://github.com/dignsys/alsa-ucm-data-alc5623.git  
$ git clone https://github.com/dignsys/audio-hal-alc5623.git  
$ git clone https://github.com/dignsys/libtbtm-nexell-anchor3.git  
$ git clone https://github.com/dignsys/mmfw-sysconf.git
```

## Tizen Package Build

### ◆ Prepare GBS build configuration

```
$ cp ~/Tizen-Work-boot/tools-anchor3/files/gbsAnchor3.conf ~/gbsAnchor3.conf
```

### ◆ Build TizenPackage with GBS

```
$ cd ~/Tizen-Work  
$ gbs -c ~/gbsAnchor3.conf build -A armv7l --include-all --thread=4
```

### ◆ Create Boot Images

- Copy Kick start from tools-anchor3 on boot package

```
$ cd ~/Tizen-Work  
$ cp ~/Tizen-Work-boot/tools-anchor3/files/tizen-unified_iot-headless-2parts-armv7l-anchor3.ks .  
$ cp ~/Tizen-Work-boot/tools-anchor3/files/tizen-unified_iot-headless-3parts-armv7l-anchor3.ks .
```

- 2 Parts Image : Only for IoT

```
$ sudo mic cr loop tizen-unified_iot-headless-2parts-armv7l-anchor3.ks --pack-to=tizen-unified_iot-headless-2parts-armv7l-anchor3.tar.gz --logfile=mic_build.log -A armv7l
```

- 3 Parts Image : IoT + something else

```
$ sudo mic cr loop tizen-unified_iot-headless-3parts-armv7l-anchor3.ks --pack-to=tizen-unified_iot-headless-3parts-armv7l-anchor3.tar.gz --logfile=mic_build.log -A armv7l
```

## Boot Package Build

### ◆ Build Boot Package with GBS

```
$ cd ~/Tizen-Work-boot  
$ gbs -c ~/gbsAnchor3.conf build -A armv7l --include-all --thread=4
```

### ◆ Create Boot Images

- Copy Kick start from tools-anchor3 on boot package

```
$ cd ~/Tizen-Work-boot  
$ cp ~/Tizen-Work-boot/tools-anchor3/files/tizen-unified_iot-boot-armv7l-anchor3.ks .
```

- Create Boot Image

```
$ cd ~/Tizen-Work-boot  
$ sudo mic cr loop tizen-unified_iot-boot-armv7l-anchor3.ks --pack-to=tizen-unified_iot-boot-armv7l-anchor3.tar.gz --logfile=mic_build.log -A armv7l
```

### ◆ Reference : boot-firmwares-anchor3/files

- BL1, BL2, BL32, U-Boot : do not build by GBS

- RAMDISKImage created from Tizen package ramdisk.img

```
$ mkimage -A arm -O linux -T ramdisk -C none -a 0 -e 0 -n ulnitr -d ramdisk.img ulnitr
```

## Prepare to Fusing

- ◆ Boot Package & Tizen Packages
  - Boot Package
    - ◊ tizen-unified\_iot-boot-armv7l-anchor3.tar.gz
  - 2 Part Tizen Package
    - ◊ tizen-unified\_iot-headless-2parts-armv7l-anchor3.tar.gz
  - 3 Part Tizen Package
    - ◊ tizen-unified\_iot-headless-3parts-armv7l-anchor3.tar.gz
- ◆ Copy binary & package to Tizen-Work-boot/result
  - Tizen Boot Packages
    - ◊ boot.img
    - ◊ modules.img
  - Boot Binaries & Logo
    - ◊ bl1-nxp3220.bin.raw, bl2.bin.raw, bl32.bin.raw, u-boot.bin.raw, params\_env.bin
    - ◊ logo.bmp
  - Tizen Packages
    - ◊ rootfs.img
    - ◊ system-date.img
    - ◊ user.img (only for 3 parts package)

## Copy binaries & package

- ◆ Copy binaries & package to ~/Tizen-Work-boot/result

```
$ cd ~/Tizen-Work-boot
$ mkdir -p result

$ cd ~/Tizen-Work/mic-output
$ tar zxvf tizen-unified_iot-headless-2parts-armv7l-anchor3.tar.gz
$ mv *.img ~/Tizen-Work-boot/result

$ cd ~/Tizen-Work-boot/mic-output
$ tar zxvf ..tizen-unified_iot-boot-armv7l-anchor3.tar.gz
$ mv *.img ~/Tizen-Work-boot/result
$ mv *.raw ~/Tizen-Work-boot/result
$ mv params_env.bin ~/Tizen-Work-boot/result
```
- ◆ Caution for “boot-firmware-anchor3/files”
  - RAMDISK
    - ◊ If you change RAMDISK, then you must generate ramdisk image for U-boot by mkimage and copy to directory
  - Boot logo : logo.bmp
    - ◊ If you want to change boot logo, you can change logo.bmp on directory

## Contents

- ◆ Course Introduction
- ◆ Tizen & Tizen enabled Hardware Platform
- ◆ Anchor3 Hardware Platform
- ◆ **Tizen OS Development Environment**
  - Setup Environment
  - Tizen Packages
    - ❖ Build Package
    - ❖ Tizen Partition
    - ❖ Fusing Image
- ◆ Tizen Application Development Environment

## Partition Information

- ◆ Tizen Partition Information

Partition	Mount Point label	Mount to	Option	Description
			ro	Boot loader
/mmcblk0/p1	boot		ro	Kernel & ramdisk
/mmcblk0/p2	modules	/usr/lib/modules	ro	Kernel modules
/mmcblk0/p3	rootfs	/	rw	Tizen Platform
/mmcblk0/p4	system-data	/opt	rw	System data
/mmcblk0/p5	user	/opt/usr	rw	User data

## Partition mapping table

- ◆ eMMC partition mapping table

eMMC Partition	Partition name	Start	Size	Size(MB)
boot area	bl1	0x0	0x10000	4MB
	bl2	0x18000	0x10000	
	bl32	0x30000	0x200000	
	U-boot	0x230000	0x100000	
	env	0x330000	0x4000	
/mmcblk0/p1	boot	0x400000	0x8000000	128MB
/mmcblk0/p2	modules	0x10400000	0x8000000	128MB
/mmcblk0/p3	rootfs	0x18400000	0x80000000	2048MB
/mmcblk0/p4	system-data	0x98400000	0x40000000	1024MB
/mmcblk0/p5	user	0xd8400000	-	-

## Setup Partition (1)

- ◆ Automatic Partition setup by using script (mkpart.sh)

```
#!/bin/bash
usage()
{
    echo "Usage: `basename $0` /dev/sdX"
    exit 1
}

[ $# -eq 1 ] || usage
[[ $1 =~ ^/dev/sd[b-z]$ ]] || usage

DISK=$1
SIZE=`fdisk -s $DISK`
SIZE_MB=$((SIZE >> 10))

SKIP_BOOT_SIZE=4
BOOT_SZ=128
MODULE_SZ=128
ROOTFS_SZ=2048
DATA_SZ=1024

let "USER_SZ = $SIZE_MB - $BOOT_SZ - $ROOTFS_SZ - $DATA_SZ - $MODULE_SZ - 4"

BOOT=boot
MODULE=modules
ROOTFS=rootfs
SYSTEMDATA=system-data
USER=user
```

## Setup Partition (2)

Continue.. (2)

```
if [[ $USER_SZ -le 100 ]]
then
    echo "We recommend to use more than 4GB disk"
    exit 0
fi

echo "=====
echo "Label      dev      size"
echo "====="
echo $BOOT"      "$DISK"1"   "$BOOT_SZ"MB"
echo $MODULE"    "$DISK"2"   "$MODULE_SZ"MB"
echo $ROOTFS"    "$DISK"3"   "$ROOTFS_SZ"MB"
echo $SYSTEMDATA" "$DISK"5"   "$DATA_SZ"MB"
echo $USER"       "$DISK"6"   "$USER_SZ"MB"

MOUNT_LIST=`mount | grep $DISK | awk '{print $1}'`
for mnt in $MOUNT_LIST
do
    umount $mnt
done

echo "Remove partition table..."
dd if=/dev/zero of=$DISK bs=512 count=1 conv=notrunc
```

Continue.. (3)

```
sfdisk $DISK <__EOF__
$SKIP_BOOT_SIZE)MiB,$(BOOT_SIZE)MiB,0xE,*"
4MiB,$(BOOT_SZ)MiB,-
$(MODULE_SZ)MiB,-
$(ROOTFS_SZ)MiB,-
$(DATA_SZ)MiB,-
$(USER_SZ)MiB,-
__EOF__

mkfs.ext4 -q ${DISK}1 -L $BOOT -F
mkfs.ext4 -q ${DISK}2 -L $MODULE -F
mkfs.ext4 -q ${DISK}3 -L $ROOTFS -F
mkfs.ext4 -q ${DISK}5 -L $SYSTEMDATA -F
mkfs.ext4 -q ${DISK}6 -L $USER -F
```

## Contents

- ◆ Course Introduction
- ◆ Tizen & Tizen enabled Hardware Platform
- ◆ Anchor3 Hardware Platform
- ◆ **Tizen OS Development Environment**
  - Setup Environment
  - **Tizen Packages**
    - ❖ Build Package
    - ❖ Tizen Partition
    - ❖ **Fusing Image**
- ◆ Tizen Application Development Environment

## Serial Boot (First boot)



### ◆ Fusing with linux-usbdowndloader utility for new board

- Verify Anchor3 OTG Connection

```
$ lsusb | grep 3225
Bus 001 Device 014: ID 2375:3225 Digit@lway, Inc.
```



Boot Mode	Switch
Normal (eMMC)	All off
USB Serial	On #1

- Download & Running Firmware

- ❖ BL1, BL2, BL32, U-Boot & Parameter
- ❖ Script : tools-anchor3/scripts/usb-down.sh

```
/tools-anchor3/scripts/usb-down.sh -f tools-anchor3/scripts/configs/udown.bootloader.sh -d result
```

### ◆ Hit any key on Target Immediately (on Target U-Boot Console)

```
Model: Dignsys Anchor3 board
DRAM: 496 MIB
BOOT: EMMC: 0x0
MMC: NXP3220 DWMMC: 0, NXP3220 DWMMC: 1, NXP3220 DWMMC: 2
Loading Environment from MMC... OK
In: serial
Out: serial
Err: serial
~~~~~
```

**U-boot is running on DRAM**

```
Hit any key to stop autoboot: 0
anchor3#
```

## Fusing Tizen Image



### ◆ Start fastboot (on Target U-Boot Console)

```
anchor3# fastboot
FASTBOOT [Google] Wait for download ...
```

### ◆ Fusing Image to eMMC with fastboot utility

- Script file : tools-anchor3/scripts/partmap\_fastboot.sh
- Partition map : tools-anchor3/files/partmap\_anchor3\_tizen\_emmc.txt
- Binary & Packages : ~/Tizen-Work-boot/result
- Run the fastboot utility by script (on Target U-Boot Console)

```
$ cd ~/Tizen-Work-boot
$ ./tools-anchor3/scripts/partmap_fastboot.sh -f tools-anchor3/files/partmap_anchor3_tizen_emmc.txt -d result
Partmap: tools-anchor3/files/partmap_anchor3_tizen_emmc.txt
[sudo] password for hbahn:
target reported max download size of 436207616 bytes
sending 'partmap' (0 KB)...
OKAY [ 0.003s]
writing 'partmap'...
OKAY [ 0.068s]
finished. total time: 0.071s
bl1: /home2/Anchor3/Tizen-Work-boot/result/bl1-nxp3220.bin.raw
target reported max download size of 436207616 bytes
~~~~~
```

## eMMC Partition Mapping Table

- ◆ eMMC Partition Mapping Table for fastboot
  - Partition map : tools-anchor3/files/partmap\_anchor3\_tizen\_emmc.txt

```
flash=mmc,0:bl1:bootsector:0x0,0x10000:bl1-nxp3220.bin.raw;
flash=mmc,0:bl2:bootsector:0x18000,0x10000:bl2.bin.raw;
flash=mmc,0:bl32:bootsector:0x30000,0x20000:bl32.bin.raw;
flash=mmc,0:bootloader:bootsector:0x230000,0x100000:u-boot.bin.raw;
flash=mmc,0:env:raw:0x330000,0x4000:params_env.bin;
flash=mmc,0:boot:partition:0x400000,0x8000000:boot.img;
flash=mmc,0:modules:partition:0x10400000,0x8000000:modules.img;
flash=mmc,0:rootfs:partition:0x18400000,0x8000000:rootfs.img;
flash=mmc,0:system-data:partition:0x98400000,0x4000000:system-data.img;
flash=mmc,0:user:partition:0xd8400000,0:user.img;
```

## Booting with Tizen

- ◆ Booting on Target

```
.....
anchor3 login: root
Password:
Welcome to Tizen
root:~>
root:~> uname -a
Linux anchor3 4.14.73-0-anchor3 #1 SMP PREEMPT Sun Sep 1 06:24:39 UTC 2019 armv7l GNU/Linux
root:~>
root:~> cat /etc/tizen-release
Tizen5/Unified 5.5.0 (arm)
VERSION = 5.5.0
CODENAME = Next
BUILD_ID=tizen-unified_iot-headless-3parts-armv7l-anchor3
root:~>
```

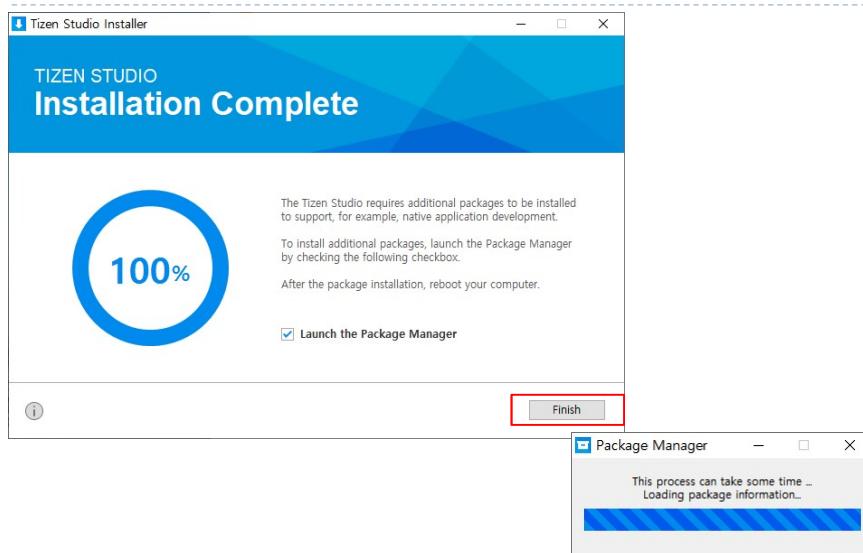
## Contents

- ◆ Course Introduction
- ◆ Tizen & Tizen enabled Hardware Platform
- ◆ Anchor3 Hardware Platform
- ◆ Tizen OS Development Environment
- ◆ **Tizen Application Development Environment**
  - [Install Tizen Studio](#)
  - Practice Hello Application

83

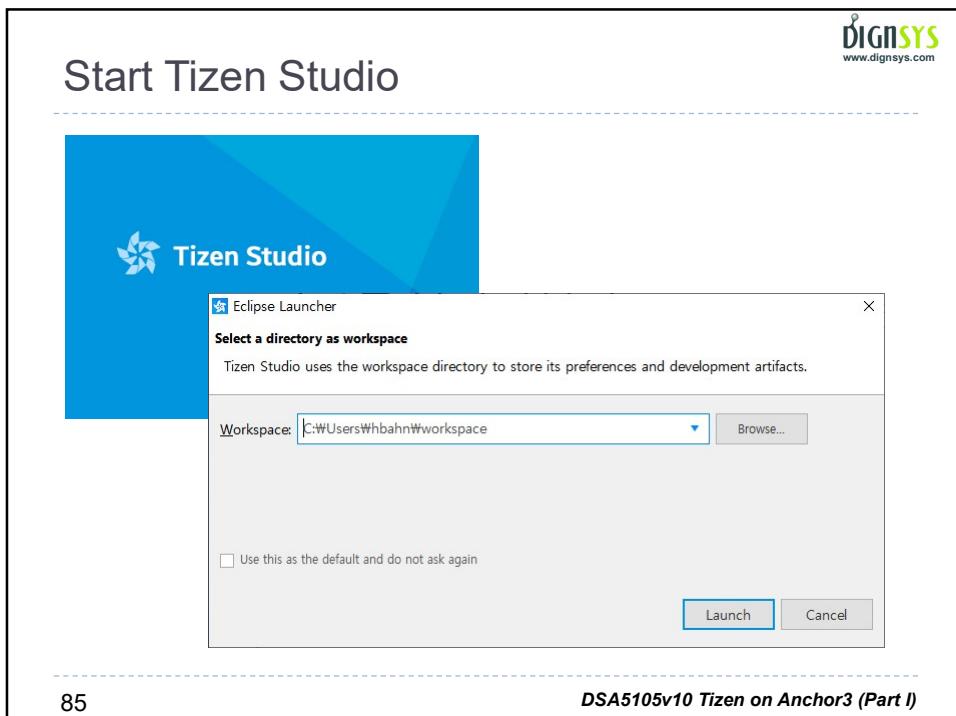
DSA5105v10 Tizen on Anchor3 (Part I)

## Install Tizen Studio



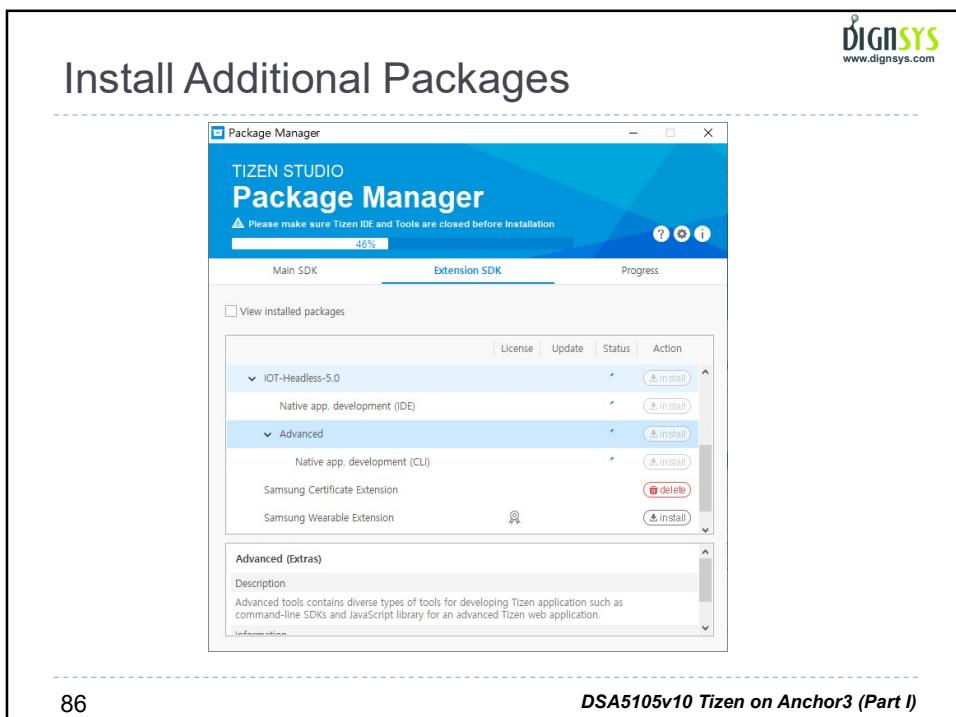
84

DSA5105v10 Tizen on Anchor3 (Part I)



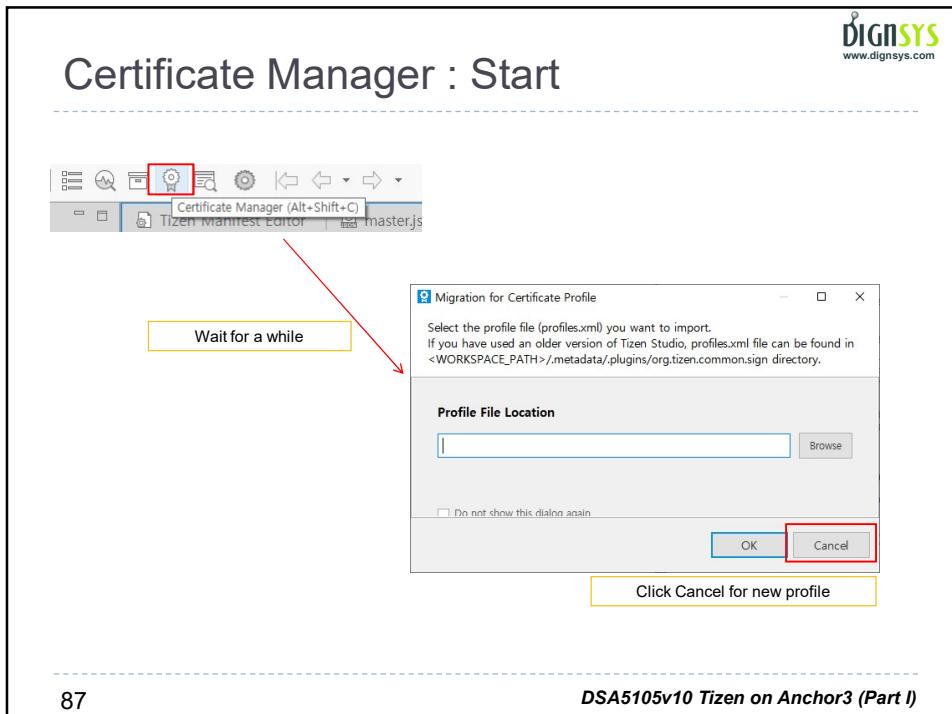
85

DSA5105v10 Tizen on Anchor3 (Part I)



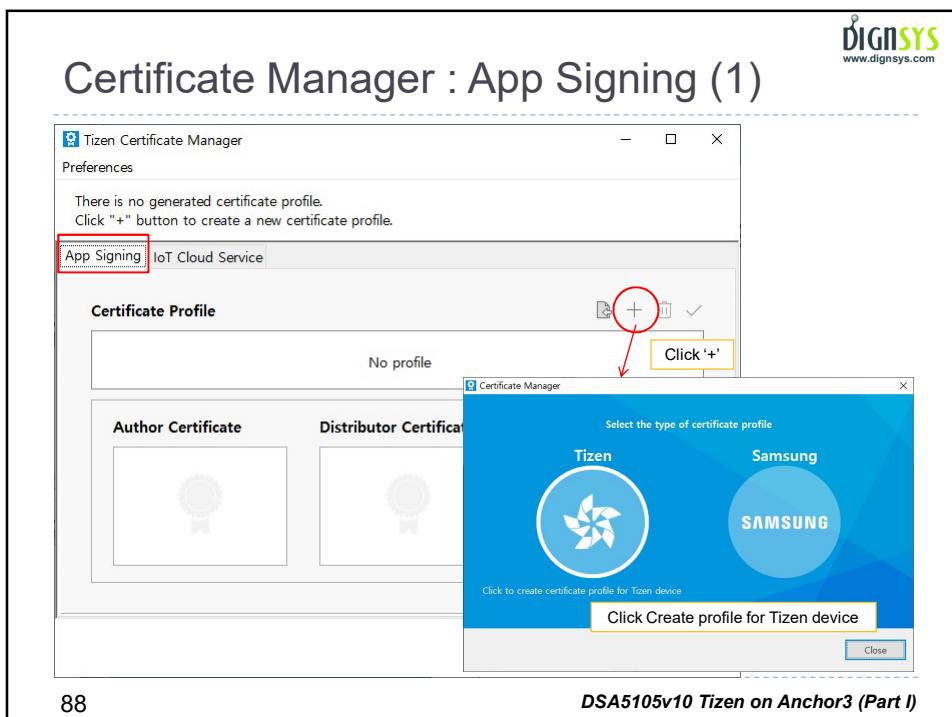
86

DSA5105v10 Tizen on Anchor3 (Part I)



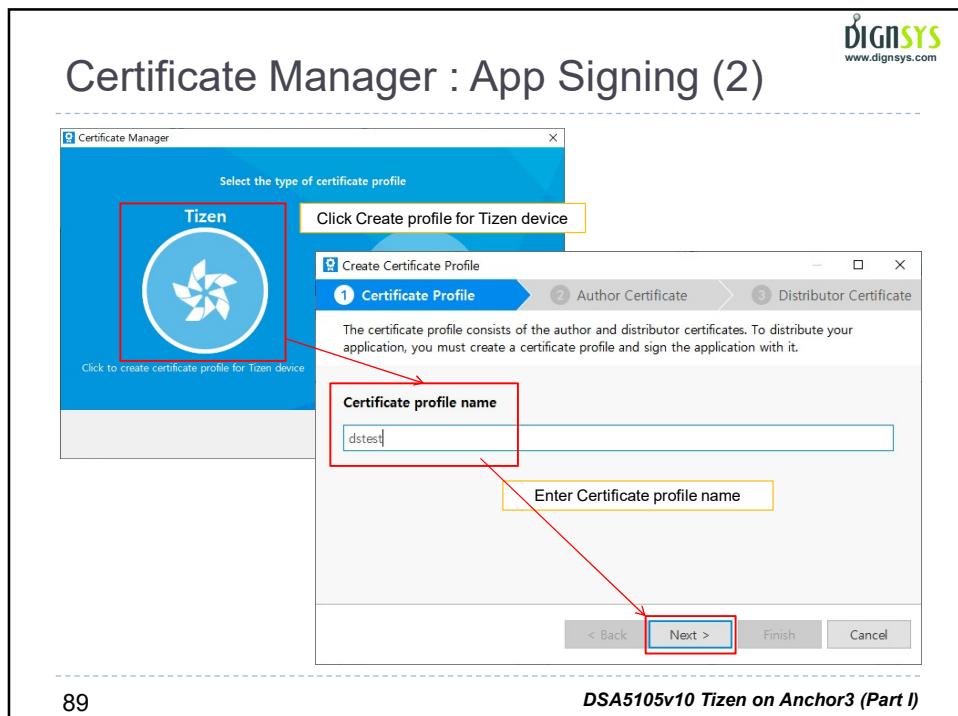
87

DSA5105v10 Tizen on Anchor3 (Part I)

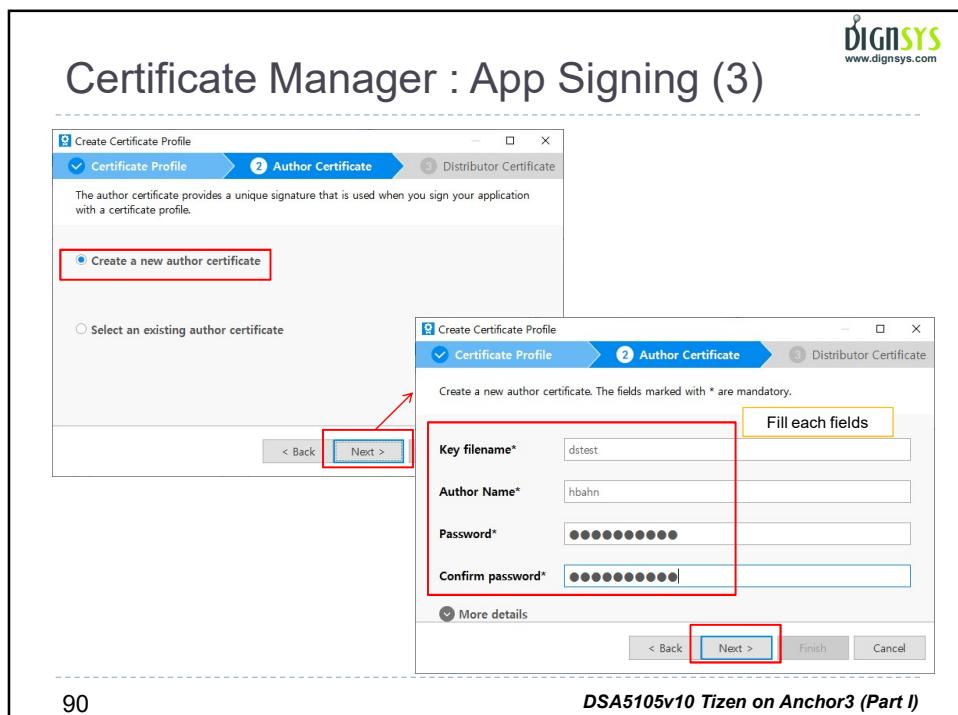


88

DSA5105v10 Tizen on Anchor3 (Part I)



89

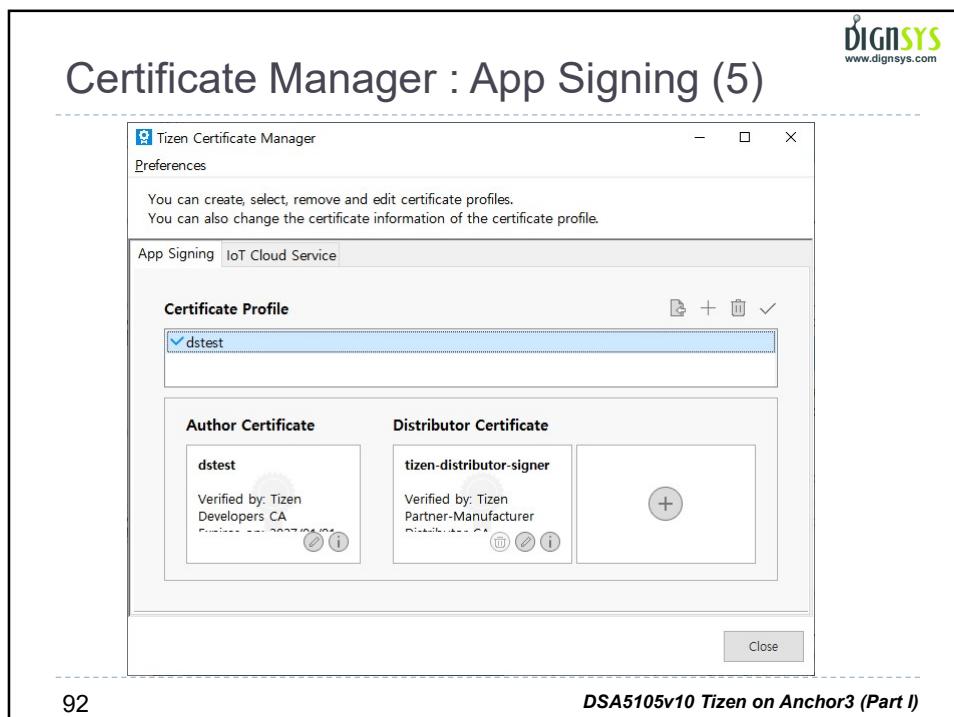


90



91

DSA5105v10 Tizen on Anchor3 (Part I)



92

DSA5105v10 Tizen on Anchor3 (Part I)

## Contents

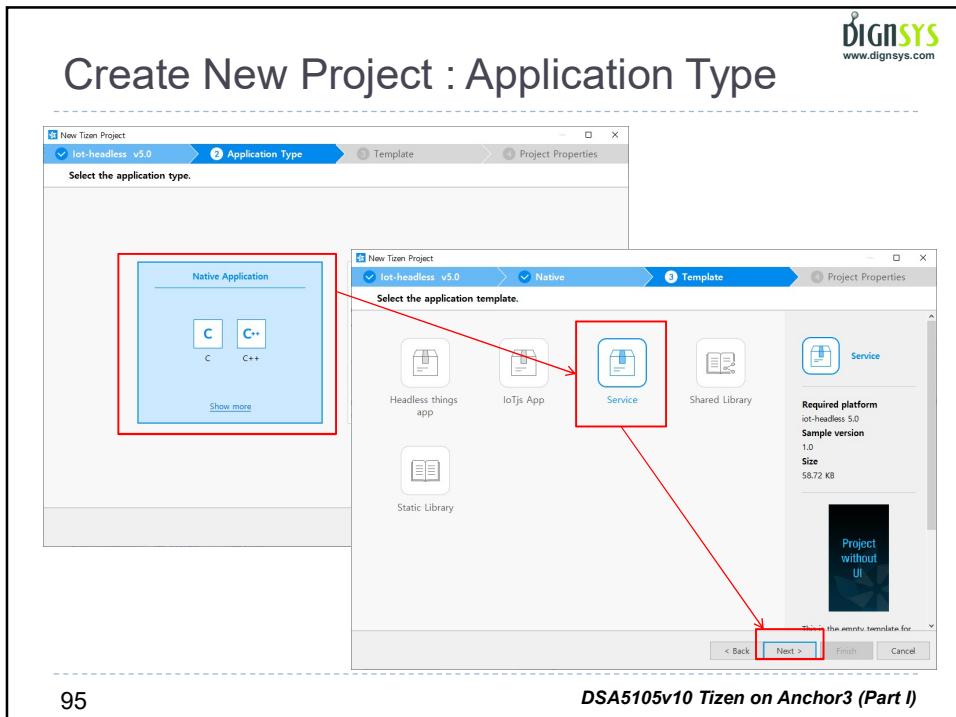
- ◆ Course Introduction
- ◆ Tizen & Tizen enabled Hardware Platform
- ◆ Anchor3 Hardware Platform
- ◆ Tizen OS Development Environment
- ◆ **Tizen Application Development Environment**
  - [Install Tizen Studio](#)
  - Practice Hello Application

93 **DSA5105v10 Tizen on Anchor3 (Part I)**

## Create New Project

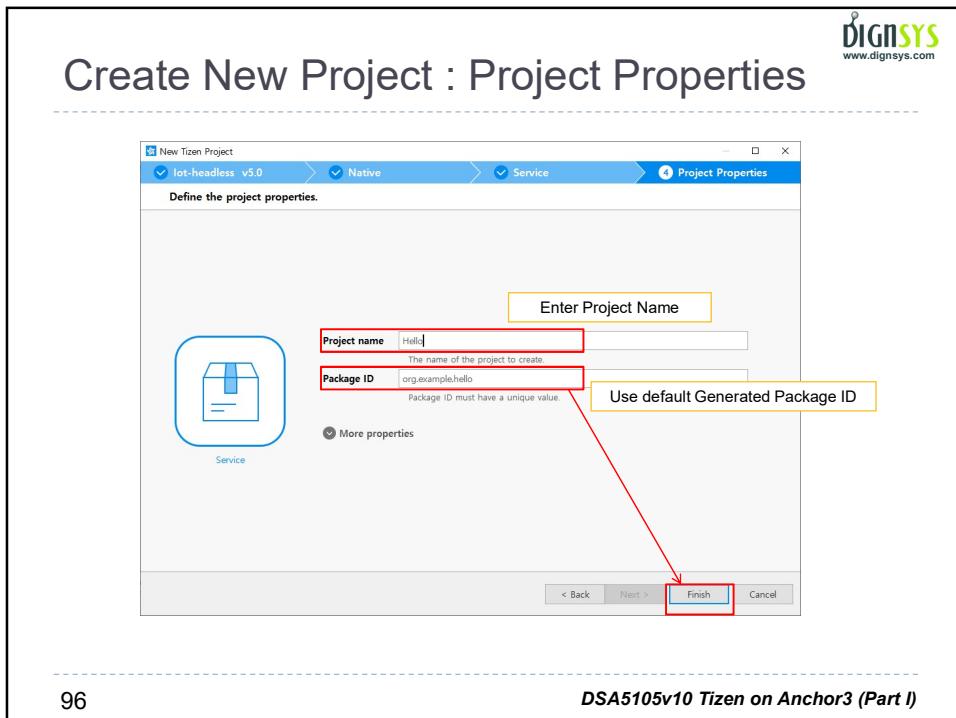
The screenshot shows the 'Create New Project' dialog in Tizen Studio. The 'File' menu is open, showing 'New' and 'Tizen Project' selected. The main window displays a 'Profile & Version' step with four options: 'Mobile' (v5.0), 'Wearable' (v5.0), 'TV' (v5.0), and 'Custom'. The 'Custom' option is highlighted with a red box. Red arrows point from the 'Template' section on the left to the 'Custom' icon, and from the 'Next >' button at the bottom to the 'Next >' button at the top right.

94 **DSA5105v10 Tizen on Anchor3 (Part I)**



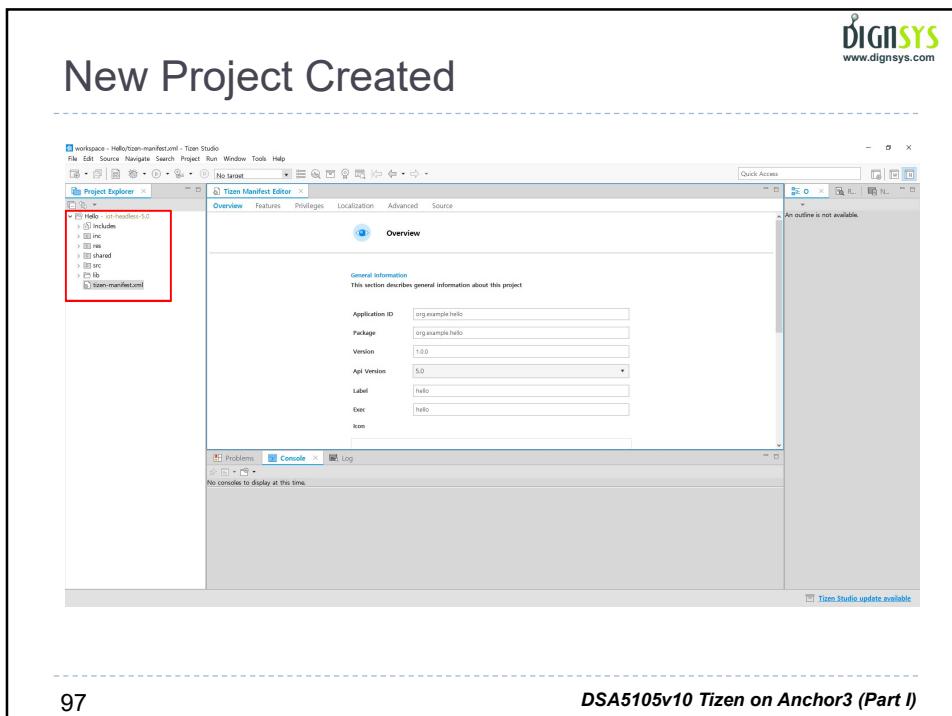
95

DSA5105v10 Tizen on Anchor3 (Part I)



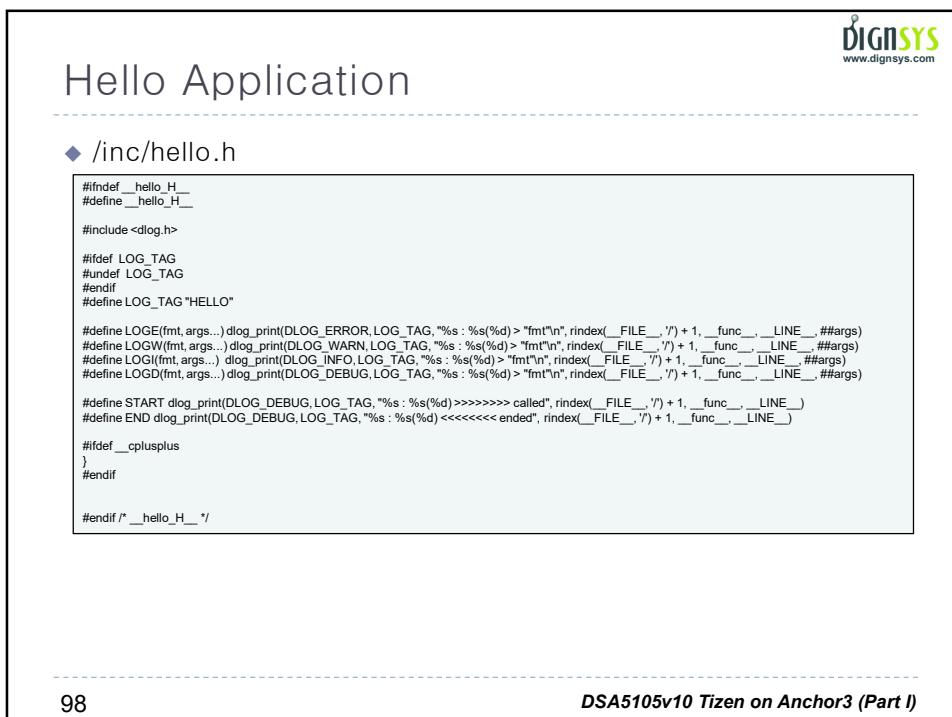
96

DSA5105v10 Tizen on Anchor3 (Part I)



97

DSA5105v10 Tizen on Anchor3 (Part I)



98

DSA5105v10 Tizen on Anchor3 (Part I)

## DSA5105. Tizen with Anchor3 (Part1)

Hello Application

◆ /src/hello.c

```
int main(int argc, char* argv[])
{
    char ad[50] = {0};
    service_app_lifecycle_callback_s event_callback;
    app_event_handler_h handlers[5] = {NULL,};

    event_callback.create = service_app_create;
    event_callback.terminate = service_app_terminate;
    event_callback.app_control = service_app_control;

    service_app_add_event_handler(&handlers[APP_EVENT_LOW_BATTERY], APP_EVENT_LOW_BATTERY, service_app_low_battery, &ad);
    service_app_add_event_handler(&handlers[APP_EVENT_LOW_MEMORY], APP_EVENT_LOW_MEMORY, service_app_low_memory, &ad);
    service_app_add_event_handler(&handlers[APP_EVENT_LANGUAGE_CHANGED], APP_EVENT_LANGUAGE_CHANGED,
                                  service_app_lang_changed, &ad);
    service_app_add_event_handler(&handlers[APP_EVENT_REGION_FORMAT_CHANGED], APP_EVENT_REGION_FORMAT_CHANGED,
                                  service_app_region_changed, &ad);

    LOGI("Hello Anchor .....");

    return service_app_main(argc, argv, &event_callback, ad);
}
```

99 DSA5105v10 Tizen on Anchor3 (Part I)

Build & Run Hello Application

100 DSA5105v10 Tizen on Anchor3 (Part I)

## Verify Working

- ◆ Connect to target by SDB shell & run dlogutil

Run command dlogutil prior to start application

```

C:\tizen-studio\tools>sdb devices
List of devices attached
0001-anchor3      device      anchor3

C:\tizen-studio\tools>sdb shell
sh-3.2$ uname -a
Linux anchor3 4.14.73-0-anchor3 #1 SMP PREEMPT Sun Sep 1 06:24:39 UTC 2019 armv7l GNU/Linux
sh-3.2$ dlogutil HELLO
1/HELLO  ( 290): hello.c : main(93) > [Hello Anchor .....]
1/HELLO  ( 290): hello.c : service_app_create(60) > _service_app_create
1/HELLO  ( 290): hello.c : service_app_control(44) > _service_app_control

```

101

DSA5105v10 Tizen on Anchor3 (Part I)

## Package List & Uninstall

- ◆ Connect to Device

```

C:\tizen-studio\tools>sdb devices
List of devices attached
0001-anchor3      device      anchor3

C:\tizen-studio\tools>sdb root on
Switched to 'root' account mode

C:\tizen-studio\tools>sdb shell
sh-3.2#

```

- ◆ List-up Packages

```

sh-3.2# pkgcmd -l
system apps  pkg_type [tpk] pkgid [com.samsung.tizen.smarthings-thing]  name [smarthings-thing]  version [1.0.11]
storage [internal]
system apps  pkg_type [tpk] pkgid [org.example.distancechecker]  name [distancechecker]  version [1.0.0] storage [internal]
system apps  pkg_type [tpk] pkgid [org.tizen.update-agent]  name [update-agent]  version [1.0.2] storage [internal]
spend time for pkgcmd is [24]ms
sh-3.2#

```

- ◆ Uninstall Package

```
sh-3.2# pkgcmd -u -n org.example.distancechecker
```

102

DSA5105v10 Tizen on Anchor3 (Part I)

## 질의 응답