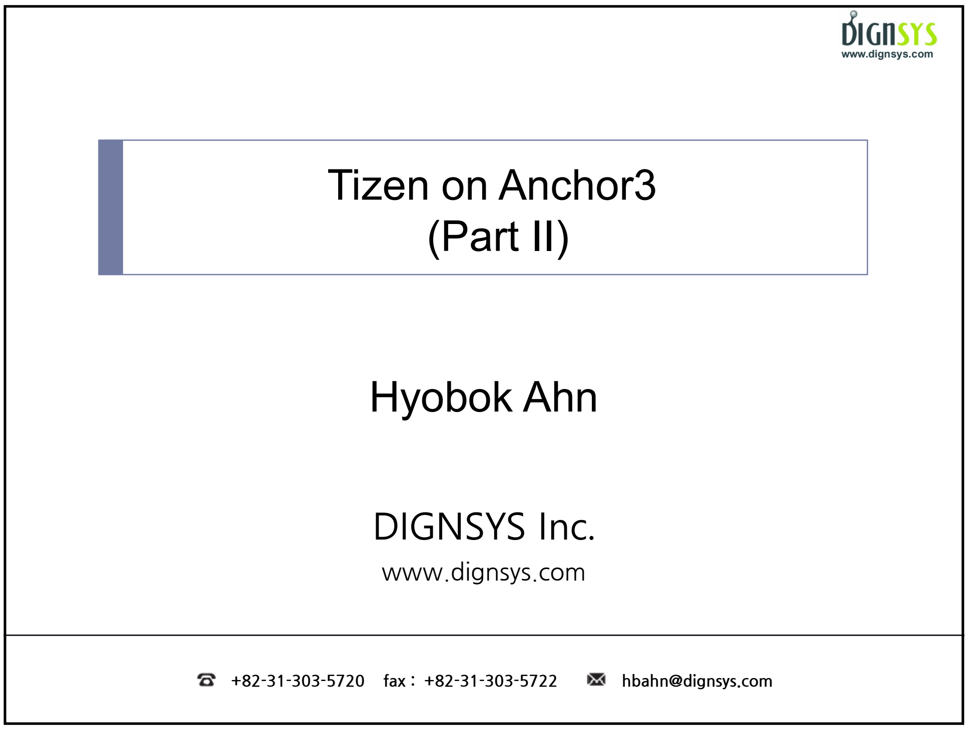


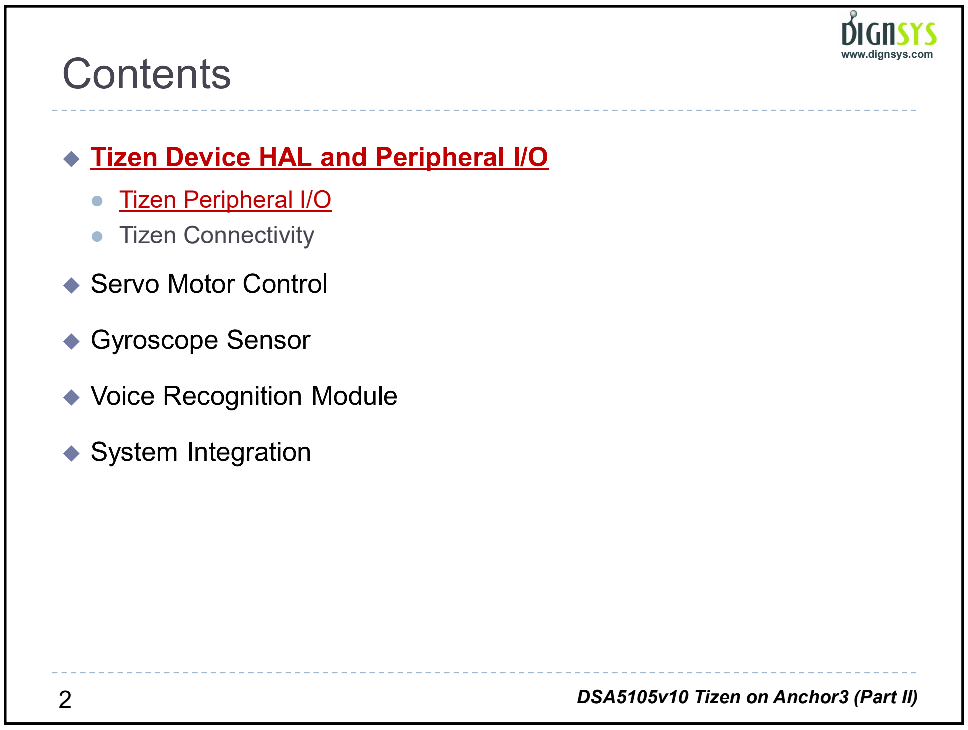
Tizen on Anchor3 (Part II)

Hyobok Ahn

DIGNSYS Inc.

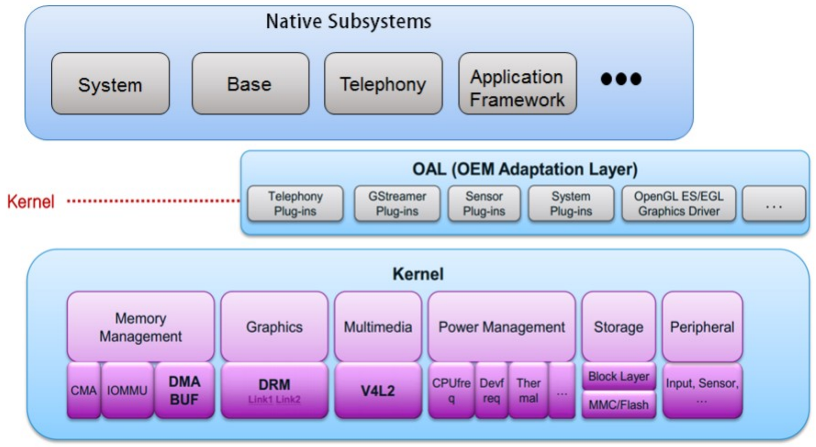
www.dignsys.com

☎ +82-31-303-5720 fax : +82-31-303-5722 ✉ hbahn@dignsys.com



Contents

Tizen OAL(OEM Adaptation Layer)

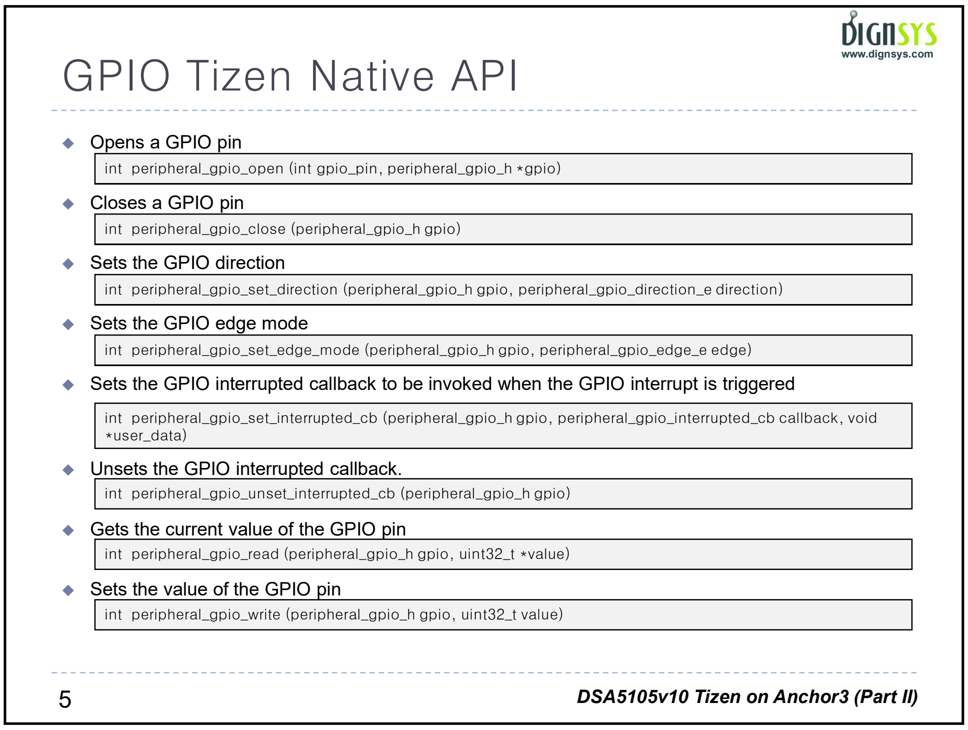


Peripheral I/O Native API

- Peripheral I/O
  - I/O interface APIs for access I/O device such as sensors
    - Communicate by D-Bus message bus

https://developer.tizen.org/dev-guide/tizen-iot-headless/latest/group__CAPI__SYSTEM__PERIPHERAL__IO__MODULE.html

- APIs for each Peripheral I/O
    - GPIO (General-Purpose Input/Output)
    - PWM (Pulse-Width Modulation)
    - SPI (Serial Peripheral Interface)
    - I2C (Inter-Integrated Circuit)
    - UART (Universal Asynchronous Receiver-Transmitter)
    - ADC (Analog Digital Converter)



GPIO Tizen Native API

- Opens a GPIO pin

```
int  peripheral_gpio_open (int gpio_pin, peripheral_gpio_h *gpio)
```

- Closes a GPIO pin

```
int  peripheral_gpio_close (peripheral_gpio_h gpio)
```

- Sets the GPIO direction

```
int  peripheral_gpio_set_direction (peripheral_gpio_h gpio, peripheral_gpio_direction_e direction)
```

- Sets the GPIO edge mode

```
int  peripheral_gpio_set_edge_mode (peripheral_gpio_h gpio, peripheral_gpio_edge_e edge)
```

- Sets the GPIO interrupted callback to be invoked when the GPIO interrupt is triggered

```
int  peripheral_gpio_set_interrupted_cb (peripheral_gpio_h gpio, peripheral_gpio_interrupted_cb callback, void
*user_data)
```

- Unsets the GPIO interrupted callback.

```
int  peripheral_gpio_unset_interrupted_cb (peripheral_gpio_h gpio)
```

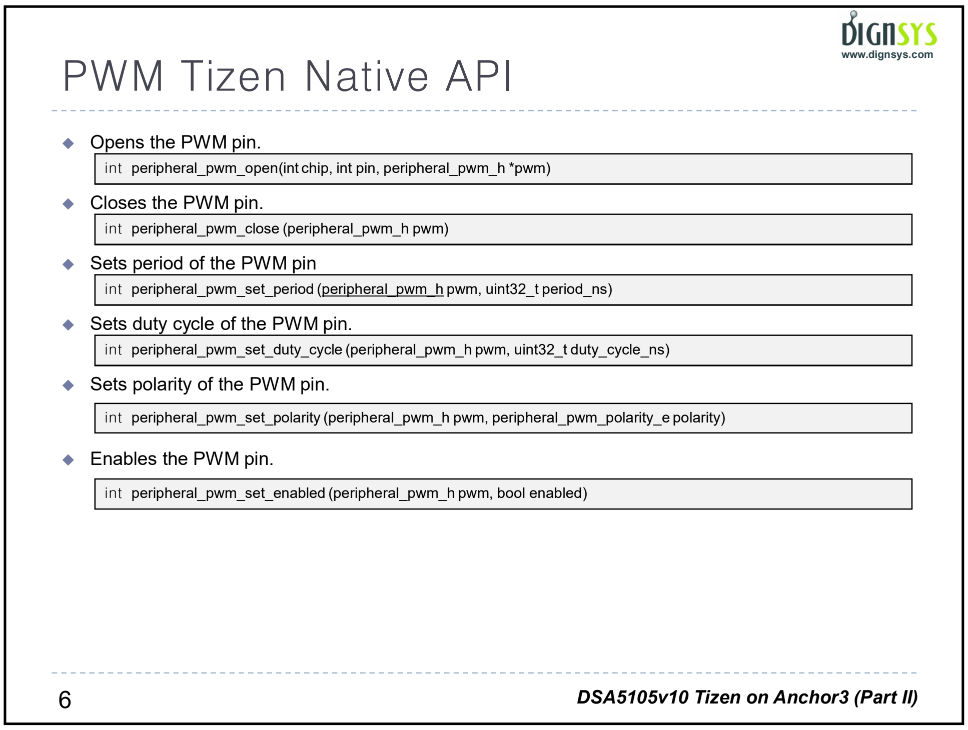
- Gets the current value of the GPIO pin

```
int  peripheral_gpio_read (peripheral_gpio_h gpio, uint32_t *value)
```

- Sets the value of the GPIO pin

```
int  peripheral_gpio_write (peripheral_gpio_h gpio, uint32_t value)
```

5 ***DSA5105v10 Tizen on Anchor3 (Part II)***



PWM Tizen Native API

- Opens the PWM pin.

```
int  peripheral_pwm_open(int chip, int pin, peripheral_pwm_h *pwm)
```

- Closes the PWM pin.

```
int  peripheral_pwm_close (peripheral_pwm_h pwm)
```

- Sets period of the PWM pin

```
int  peripheral_pwm_set_period (peripheral_pwm_h pwm, uint32_t period_ns)
```

- Sets duty cycle of the PWM pin.

```
int  peripheral_pwm_set_duty_cycle (peripheral_pwm_h pwm, uint32_t duty_cycle_ns)
```

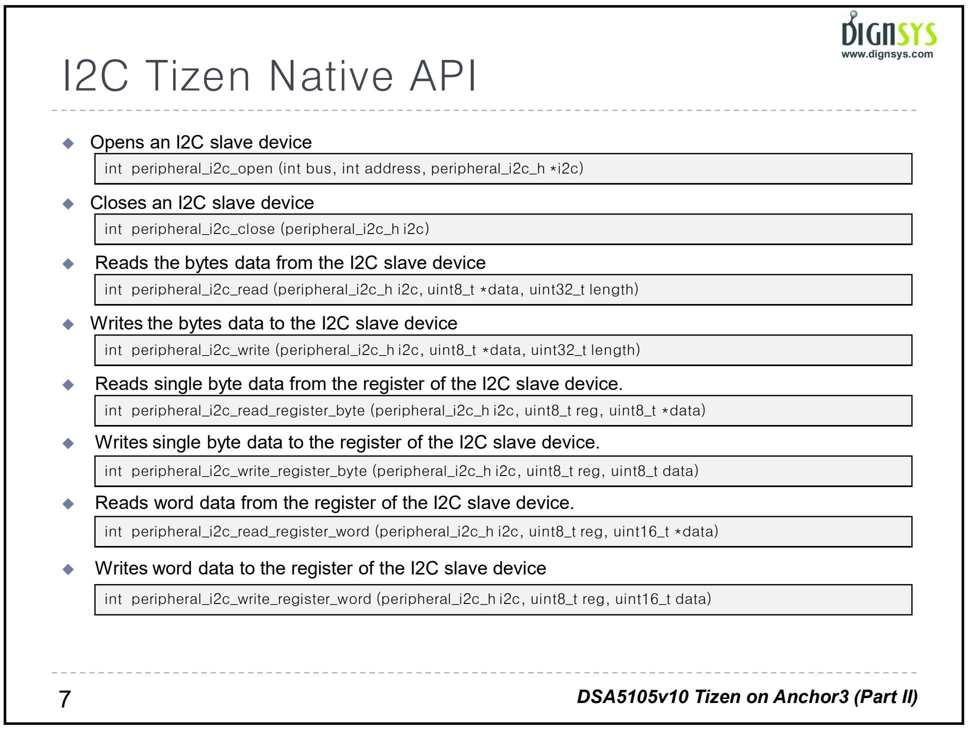
- Sets polarity of the PWM pin.

```
int  peripheral_pwm_set_polarity (peripheral_pwm_h pwm, peripheral_pwm_polarity_e polarity)
```

- Enables the PWM pin.

```
int  peripheral_pwm_set_enabled (peripheral_pwm_h pwm, bool enabled)
```

6 ***DSA5105v10 Tizen on Anchor3 (Part II)***



DIGNSYS
www.dignsys.com

I2C Tizen Native API

- Opens an I2C slave device

```
int  peripheral_i2c_open (int bus, int address, peripheral_i2c_h *i2c)
```

- Closes an I2C slave device

```
int  peripheral_i2c_close (peripheral_i2c_h i2c)
```

- Reads the bytes data from the I2C slave device

```
int  peripheral_i2c_read (peripheral_i2c_h i2c, uint8_t *data, uint32_t length)
```

- Writes the bytes data to the I2C slave device

```
int  peripheral_i2c_write (peripheral_i2c_h i2c, uint8_t *data, uint32_t length)
```

- Reads single byte data from the register of the I2C slave device.

```
int  peripheral_i2c_read_register_byte (peripheral_i2c_h i2c, uint8_t reg, uint8_t *data)
```

- Writes single byte data to the register of the I2C slave device.

```
int  peripheral_i2c_write_register_byte (peripheral_i2c_h i2c, uint8_t reg, uint8_t data)
```

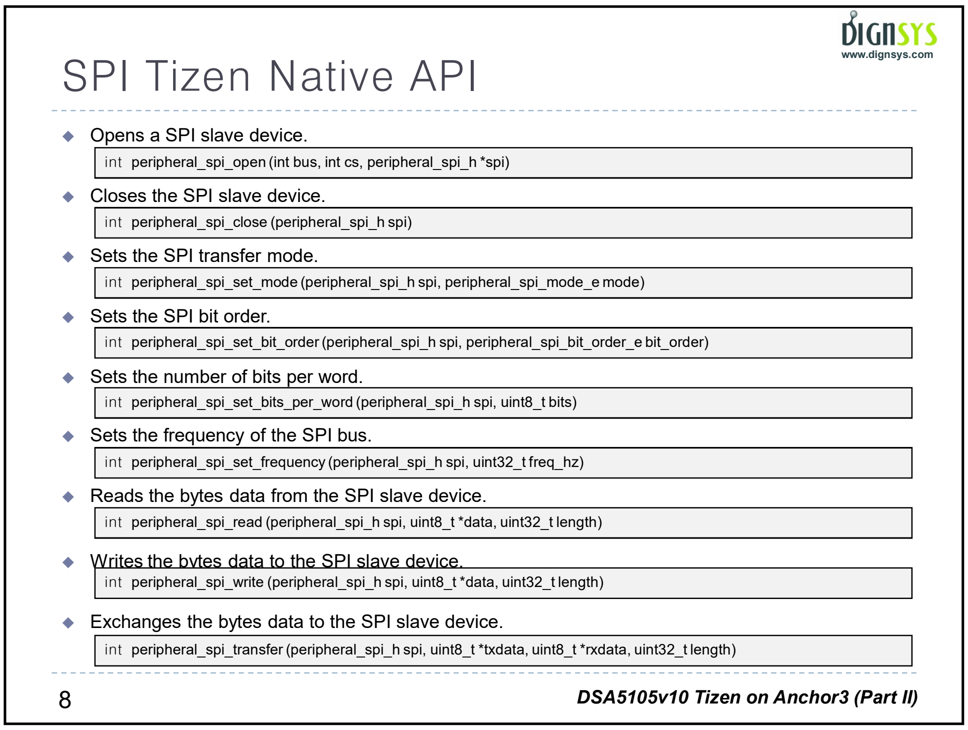
- Reads word data from the register of the I2C slave device.

```
int  peripheral_i2c_read_register_word (peripheral_i2c_h i2c, uint8_t reg, uint16_t *data)
```

- Writes word data to the register of the I2C slave device

```
int  peripheral_i2c_write_register_word (peripheral_i2c_h i2c, uint8_t reg, uint16_t data)
```

7 ***DSA5105v10 Tizen on Anchor3 (Part II)***



DIGNSYS
www.dignsys.com

SPI Tizen Native API

- Opens a SPI slave device.

```
int  peripheral_spi_open (int bus, int cs, peripheral_spi_h *spi)
```

- Closes the SPI slave device.

```
int  peripheral_spi_close (peripheral_spi_h spi)
```

- Sets the SPI transfer mode.

```
int  peripheral_spi_set_mode (peripheral_spi_h spi, peripheral_spi_mode_e mode)
```

- Sets the SPI bit order.

```
int  peripheral_spi_set_bit_order (peripheral_spi_h spi, peripheral_spi_bit_order_e bit_order)
```

- Sets the number of bits per word.

```
int  peripheral_spi_set_bits_per_word (peripheral_spi_h spi, uint8_t bits)
```

- Sets the frequency of the SPI bus.

```
int  peripheral_spi_set_frequency (peripheral_spi_h spi, uint32_t freq_hz)
```

- Reads the bytes data from the SPI slave device.

```
int  peripheral_spi_read (peripheral_spi_h spi, uint8_t *data, uint32_t length)
```

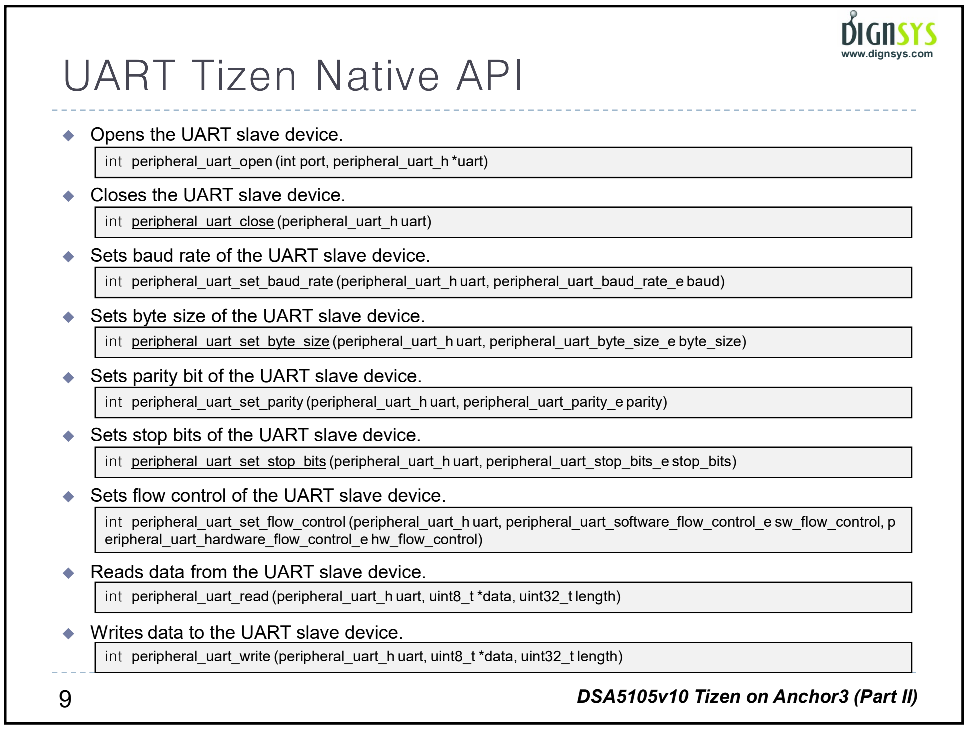
- Writes the bytes data to the SPI slave device.

```
int  peripheral_spi_write (peripheral_spi_h spi, uint8_t *data, uint32_t length)
```

- Exchanges the bytes data to the SPI slave device.

```
int  peripheral_spi_transfer (peripheral_spi_h spi, uint8_t *txdata, uint8_t *rxdata, uint32_t length)
```

8 ***DSA5105v10 Tizen on Anchor3 (Part II)***

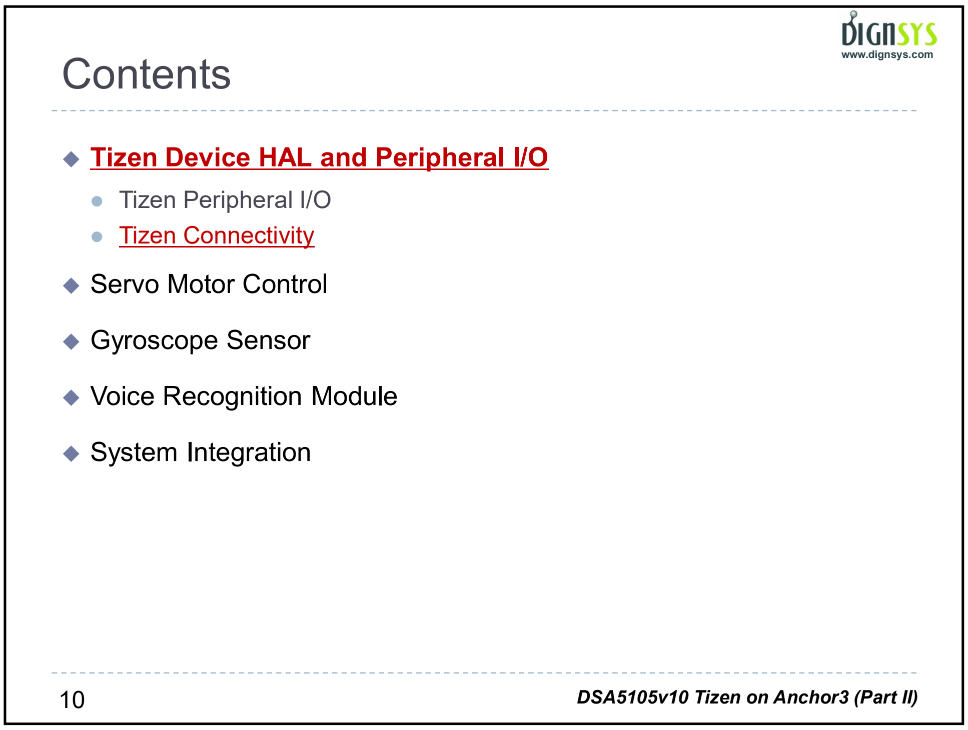


UART Tizen Native API

DIGNSYS www.dignsys.com

- Opens the UART slave device.
```
int peripheral_uart_open (int port, peripheral_uart_h *uart)
```
- Closes the UART slave device.
```
int peripheral_uart_close (peripheral_uart_h uart)
```
- Sets baud rate of the UART slave device.
```
int peripheral_uart_set_baud_rate (peripheral_uart_h uart, peripheral_uart_baud_rate_e baud)
```
- Sets byte size of the UART slave device.
```
int peripheral_uart_set_byte_size (peripheral_uart_h uart, peripheral_uart_byte_size_e byte_size)
```
- Sets parity bit of the UART slave device.
```
int peripheral_uart_set_parity (peripheral_uart_h uart, peripheral_uart_parity_e parity)
```
- Sets stop bits of the UART slave device.
```
int peripheral_uart_set_stop_bits (peripheral_uart_h uart, peripheral_uart_stop_bits_e stop_bits)
```
- Sets flow control of the UART slave device.
```
int peripheral_uart_set_flow_control (peripheral_uart_h uart, peripheral_uart_software_flow_control_e sw_flow_control, p
eripheral_uart_hardware_flow_control_e hw_flow_control)
```
- Reads data from the UART slave device.
```
int peripheral_uart_read (peripheral_uart_h uart, uint8_t *data, uint32_t length)
```
- Writes data to the UART slave device.
```
int peripheral_uart_write (peripheral_uart_h uart, uint8_t *data, uint32_t length)
```

9 ***DSA5105v10 Tizen on Anchor3 (Part II)***



Contents

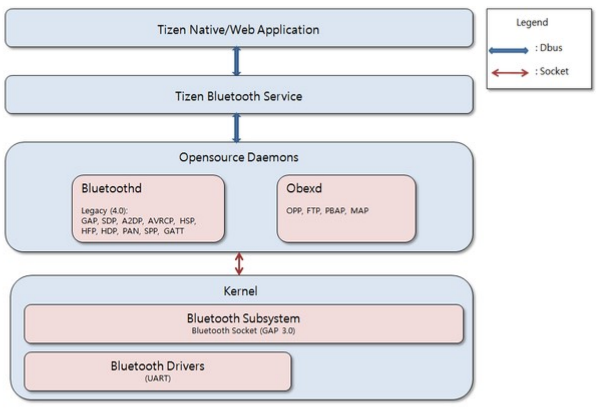
DIGNSYS www.dignsys.com

- **Tizen Device HAL and Peripheral I/O**
  - Tizen Peripheral I/O
  - Tizen Connectivity
- Servo Motor Control
- Gyroscope Sensor
- Voice Recognition Module
- System Integration

10 ***DSA5105v10 Tizen on Anchor3 (Part II)***

DIGNSYS
www.dignsys.com

Bluetooth



11 ***DSA5105v10 Tizen on Anchor3 (Part II)***

DIGNSYS
www.dignsys.com

Bluetooth Components

- Application
  - The user dialogue that controls the BlueZ, ObexD, and PulseAudio daemons
- ObexD
  - The open source component
  - Object exchange daemon
  - Supports OPP, FTP, PBAP, SYNC, and MAP profile stack
- BluetoothD
  - BluetoothD is the open souce component, Bluez 5.37 is supported
  - Bluetooth central daemon
  - Supports GAP, SDP, A2DP, AVRCP, HFP, HSP, and GATT profile stack
- Bluetooth Subsystem
  - Provides the BT unix socket. Each protocol can be accessed by its socket.
  - Supports the L2CAP, RFCOMM, SCO, and HCI protocols
- Bluetooth Driver
  - BT Chip Driver
  - In case of UART, Linux kernel provides the interface.
  - GPIO configuration, rfkill (Radio Frequency Management) and power management can be handled by both the vendor and the porting engineer.
- BT Firmware Loading Module
  - Depending on the environment, it loads the Bluetooth firmware to the Bluetooth chip.
  - Tizen and the chipset vendor need to implement this together.
  - Package: bluetooth-tools

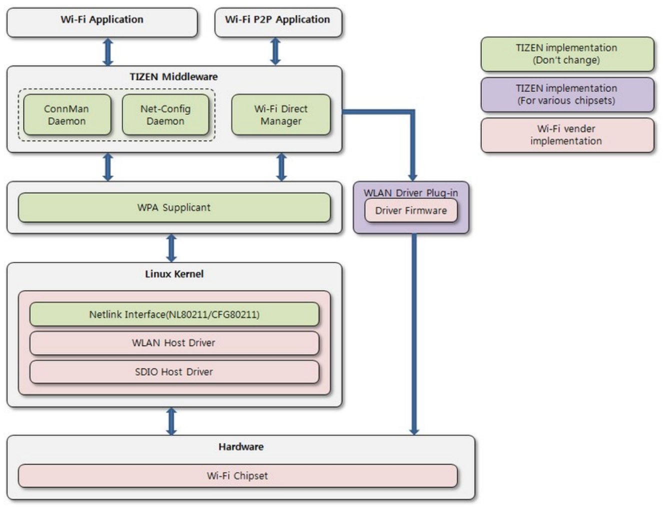
12 ***DSA5105v10 Tizen on Anchor3 (Part II)***

Bluetooth Porting OAL

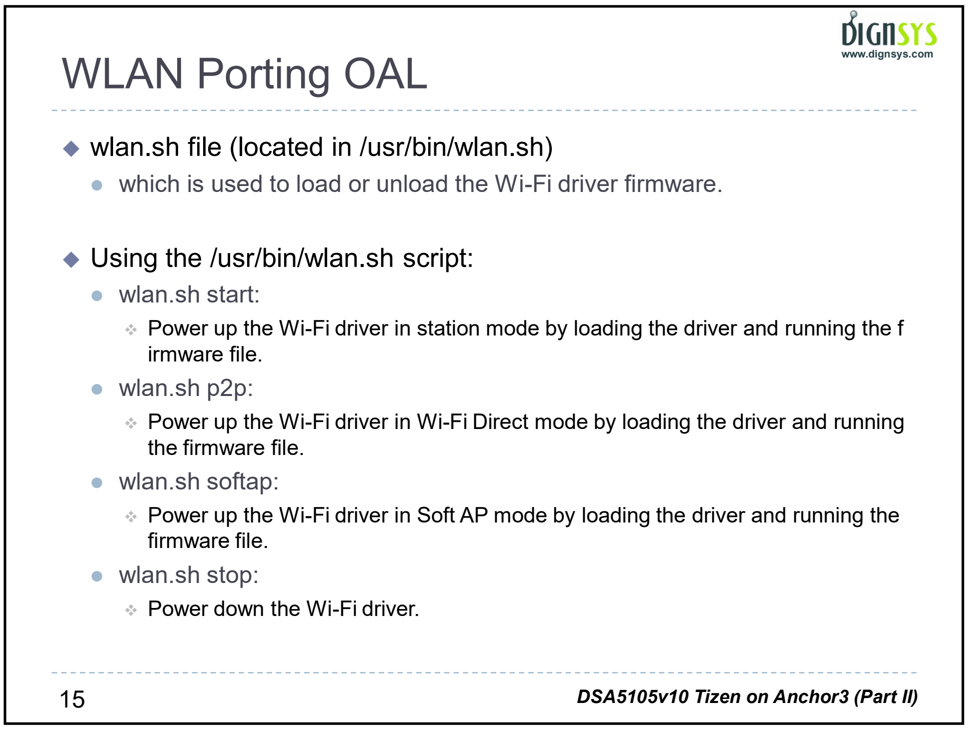
- OAL scripts are run during the Bluetooth stack start and end
  - bt-stack-up.sh
    - This script file is used to run the hardware specific script files to power up or star t the Bluetooth hardware along the background processes, such as bluez and o bexd.
  - bt-stack-down.sh
    - This script file is used to run the hardware specific script files to power down or stop the Bluetooth hardware along with the background processes, such as blue z and obexd.
  - bt-reset-env.sh
    - This script file is used to reset the Bluetooth chip by running the bt-stack-down.s h script along with the resource clean up.

13 *DSA5105v10 Tizen on Anchor3 (Part II)*

WLAN

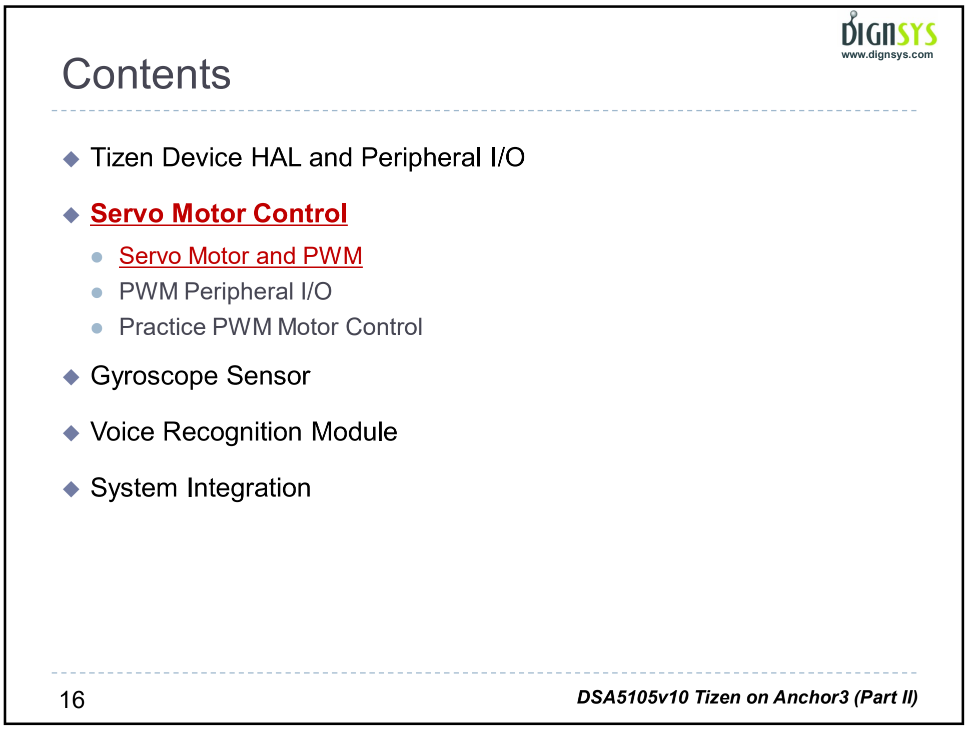


14 *DSA5105v10 Tizen on Anchor3 (Part II)*



WLAN Porting OAL

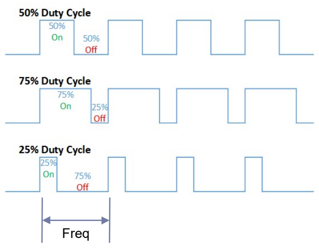
- wlan.sh file (located in /usr/bin/wlan.sh)
  - which is used to load or unload the Wi-Fi driver firmware.
- Using the /usr/bin/wlan.sh script:
  - wlan.sh start:
    - Power up the Wi-Fi driver in station mode by loading the driver and running the f irmware file.
  - wlan.sh p2p:
    - Power up the Wi-Fi driver in Wi-Fi Direct mode by loading the driver and running the firmware file.
  - wlan.sh softap:
    - Power up the Wi-Fi driver in Soft AP mode by loading the driver and running the firmware file.
  - wlan.sh stop:
    - Power down the Wi-Fi driver.



Contents

PWM

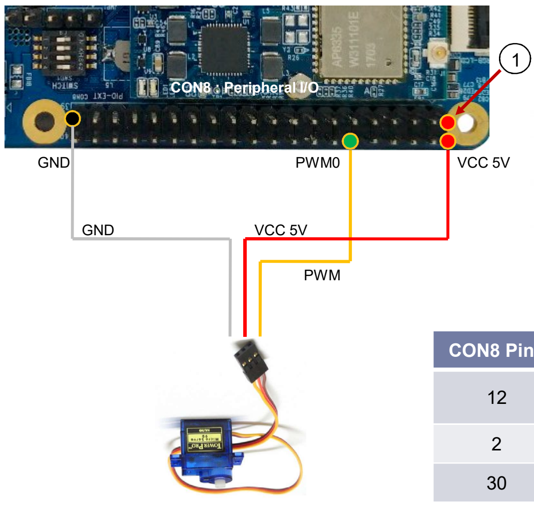
- PWM(Pulse Width Modulation)
  - Generate "Pulse" with digital signal
  - Generate with Rate(or Frequency) and Duty Cycle
- PWM Application
  - Motor Control
  - LED Brightness

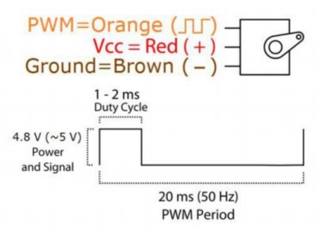


- Duty Cycle
  - Proportion of 'on(high)' time to the regular interval or 'period' of time
    - A low duty cycle corresponds to low power

17 *DSA5105v10 Tizen on Anchor3 (Part II)*

PWM with Servo Motor

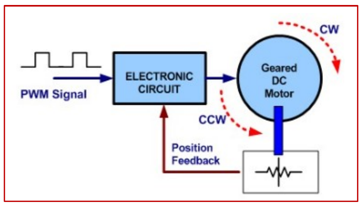




CON8 Pin#	Anchor3	Peri-I/O
12	PWM0	PWMCHIP0/ PWM0
2	VCC 5V	
30	GND	

18 *DSA5105v10 Tizen on Anchor3 (Part II)*

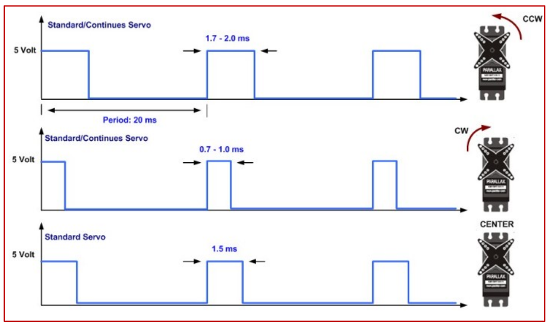
Servo Motor Control



Servo Motor Block Diagram

Servo Motor PWM Timing

```
period = 20 * MILLI_SECOND;
duty_cycle = 1 * MILLI_SECOND;// CLOCKWISE
duty_cycle = 2 * MILLI_SECOND;// COUNTER-CLOCKWISE
duty_cycle = 1.5 * MILLI_SECOND;// CENTER
```



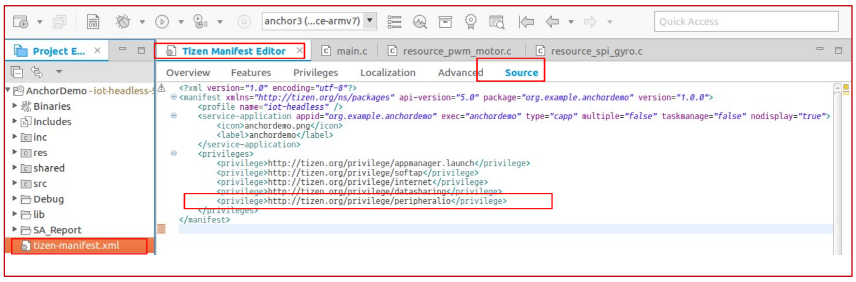
Contents

(주) 다인시스

Access Peripheral I/O

- Set access privileges on Manifest file (tizen-manifest.xml)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<manifest xmlns="http://tizen.org/ns/packages" api-version="5.0" package="org.example.hello" version="1.0.0">
    <profile name="iot-headless"/>
    <service-application appid="org.example.hello" exec="hello" multiple="false" nodisplay="true" taskmanage="false" type="capp">
        <label>anchordemo</label>
        <icon>anchordemo.png</icon>
    </service-application>
    <privileges>
        <privilege>http://tizen.org/privilege/peripheralio</privilege>
    </privileges>
</manifest>
```



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns="http://tizen.org/ns/packages" api-version="5.0" package="org.example.anchordemo" version="1.0.0">
    <profile name="iot-headless" />
    <service-application appid="org.example.anchordemo" exec="anchordemo" type="capp" multiple="false" taskmanage="false" nodisplay="true">
        <icon>anchordemo.png</icon>
        <label>anchordemo</label>
    </service-application>
    <privileges>
        <privilege>http://tizen.org/privilege/appmanager.launch</privilege>
        <privilege>http://tizen.org/privilege/softap</privilege>
        <privilege>http://tizen.org/privilege/internet</privilege>
        <privilege>http://tizen.org/privilege/datasharing</privilege>
        <privilege>http://tizen.org/privilege/peripheralio</privilege>
    </privileges>
</manifest>
```

21 *DSA5105v10 Tizen on Anchor3 (Part II)*

PWM Tizen Native API

- Opens the PWM pin.

```
int  peripheral_pwm_open(int chip, int pin, peripheral_pwm_h *pwm)
```

- Closes the PWM pin.

```
int  peripheral_pwm_close (peripheral_pwm_h pwm)
```

- Sets period of the PWM pin

```
int  peripheral_pwm_set_period (peripheral_pwm_h pwm, uint32_t period_ns)
```

- Sets duty cycle of the PWM pin.

```
int  peripheral_pwm_set_duty_cycle (peripheral_pwm_h pwm, uint32_t duty_cycle_ns)
```

- Sets polarity of the PWM pin.

```
int  peripheral_pwm_set_polarity (peripheral_pwm_h pwm, peripheral_pwm_polarity_e polarity)
```

- Enables the PWM pin.

```
int  peripheral_pwm_set_enabled (peripheral_pwm_h pwm, bool enabled)
```

22 *DSA5105v10 Tizen on Anchor3 (Part II)*

Access PWM Device

◆ /src/resource_pwm_motor.c

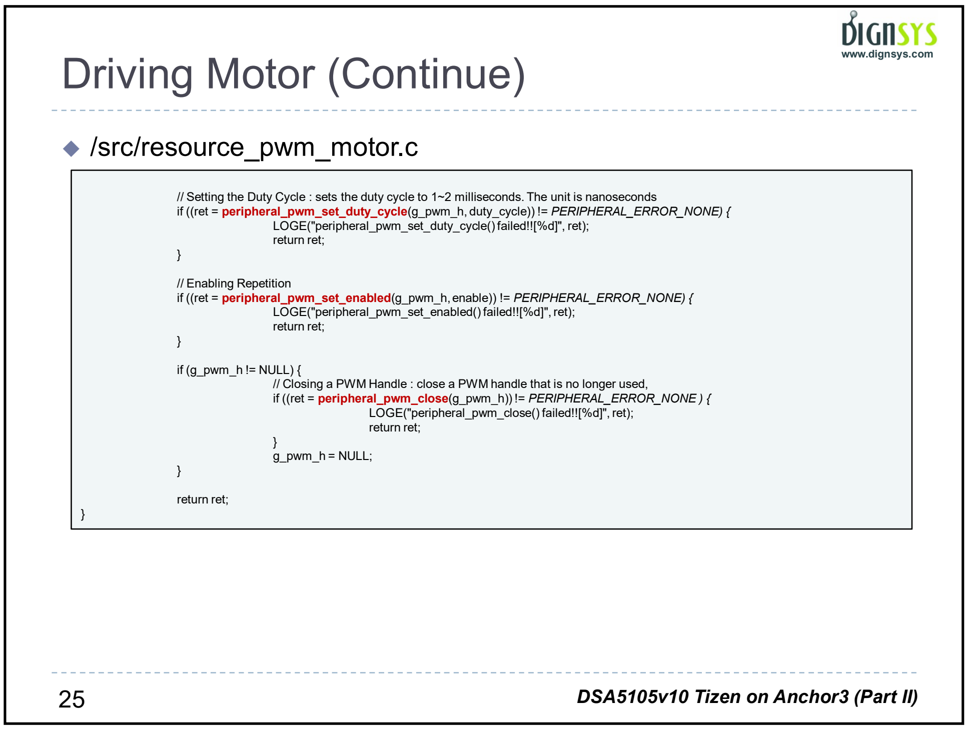
```
#include <tizen.h>
#include <peripheral_io.h>
#include <unistd.h>
#include "main.h"

#define ANCHOR3_PWM_CHIPID                    0
#define ANCHOR3_PWM_PIN                       0
#define MILLI_SECOND                          (1000000)
static peripheral_pwm_h g_pwm_h = NULL;
```

Define peripheral I/O header & PWM information

Driving Motor

◆ /src/resource_pwm_motor.c

```
peripheral_error_e resource_motor_driving(int mode)
{
        peripheral_error_e ret = PERIPHERAL_ERROR_NONE;

        int chip = ANCHOR3_PWM_CHIPID;            // Chip 0
        int pin  = ANCHOR3_PWM_PIN;               // Pin 0

        int period = 20 * MILLI_SECOND;
        int duty_cycle = 0;
        bool enable = true;

        ret = _get_duty_cycle(mode, &duty_cycle);
        if (ret != 0) {
                LOGE("_get_duty_cycle unknown mode=[%d]", mode);
                return PERIPHERAL_ERROR_INVALID_PARAMETER;
        }

        if (g_pwm_h == NULL){
                // Opening a PWM Handle : The chip and pin parameters required for this function must be set
                if ((ret = peripheral_pwm_open(chip, pin, &g_pwm_h)) != PERIPHERAL_ERROR_NONE ) {
                        LOGE("peripheral_pwm_open() failed!![%d]", ret);
                        return ret;
                }
        }

        // Setting the Period : sets the period to 20 milliseconds. The unit is nanoseconds
        if ((ret = peripheral_pwm_set_period(g_pwm_h, period)) != PERIPHERAL_ERROR_NONE) {
                LOGE("peripheral_pwm_set_period() failed!![%d]", ret);
                return ret;
        }
```

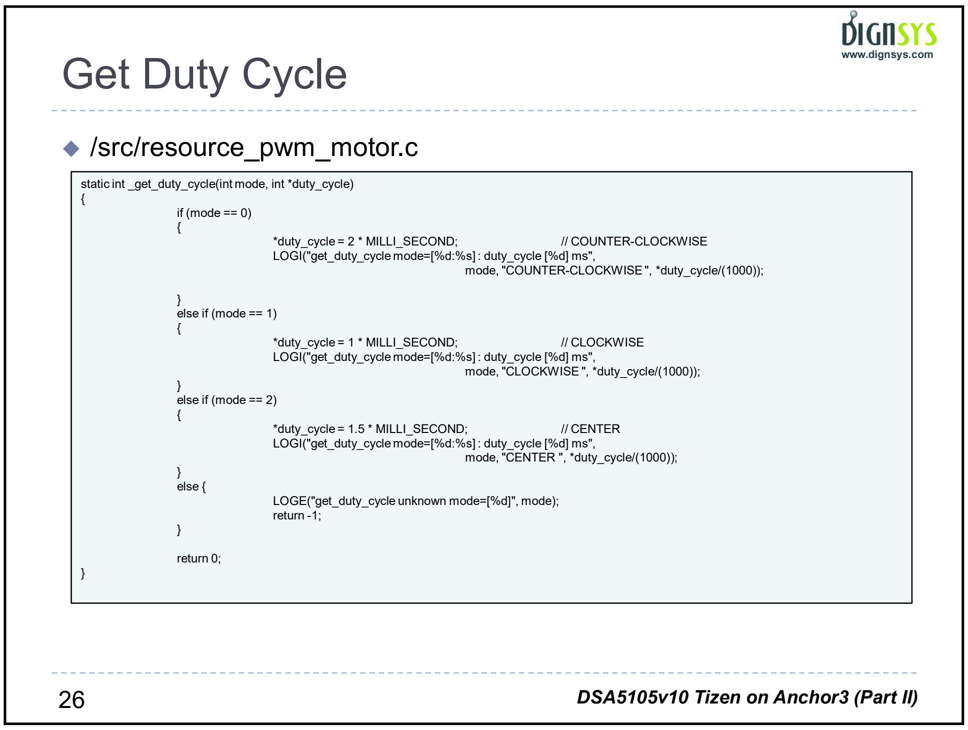


Driving Motor (Continue)

◆ /src/resource_pwm_motor.c

```
            // Setting the Duty Cycle : sets the duty cycle to 1~2 milliseconds. The unit is nanoseconds
            if ((ret = peripheral_pwm_set_duty_cycle(g_pwm_h, duty_cycle)) != PERIPHERAL_ERROR_NONE) {
                        LOGE("peripheral_pwm_set_duty_cycle() failed!![%d]", ret);
                        return ret;
            }

            // Enabling Repetition
            if ((ret = peripheral_pwm_set_enabled(g_pwm_h, enable)) != PERIPHERAL_ERROR_NONE) {
                        LOGE("peripheral_pwm_set_enabled() failed!![%d]", ret);
                        return ret;
            }

            if (g_pwm_h != NULL) {
                        // Closing a PWM Handle : close a PWM handle that is no longer used,
                        if ((ret = peripheral_pwm_close(g_pwm_h)) != PERIPHERAL_ERROR_NONE ) {
                                    LOGE("peripheral_pwm_close() failed!![%d]", ret);
                                    return ret;
                        }
                        g_pwm_h = NULL;
            }

            return ret;
}
```

25 DSA5105v10 Tizen on Anchor3 (Part II)



Get Duty Cycle

◆ /src/resource_pwm_motor.c

```
static int _get_duty_cycle(int mode, int *duty_cycle)
{
            if (mode == 0)
            {
                        *duty_cycle = 2 * MILLI_SECOND;                    // COUNTER-CLOCKWISE
                        LOGI("get_duty_cycle mode=[%d:%s] : duty_cycle [%d] ms",
                                                mode, "COUNTER-CLOCKWISE ", *duty_cycle/(1000));

            }
            else if (mode == 1)
            {
                        *duty_cycle = 1 * MILLI_SECOND;                    // CLOCKWISE
                        LOGI("get_duty_cycle mode=[%d:%s] : duty_cycle [%d] ms",
                                                mode, "CLOCKWISE ", *duty_cycle/(1000));
            }
            else if (mode == 2)
            {
                        *duty_cycle = 1.5 * MILLI_SECOND;                  // CENTER
                        LOGI("get_duty_cycle mode=[%d:%s] : duty_cycle [%d] ms",
                                                mode, "CENTER ", *duty_cycle/(1000));
            }
            else {
                        LOGE("get_duty_cycle unknown mode=[%d]", mode);
                        return -1;
            }

            return 0;
}
```

26 DSA5105v10 Tizen on Anchor3 (Part II)

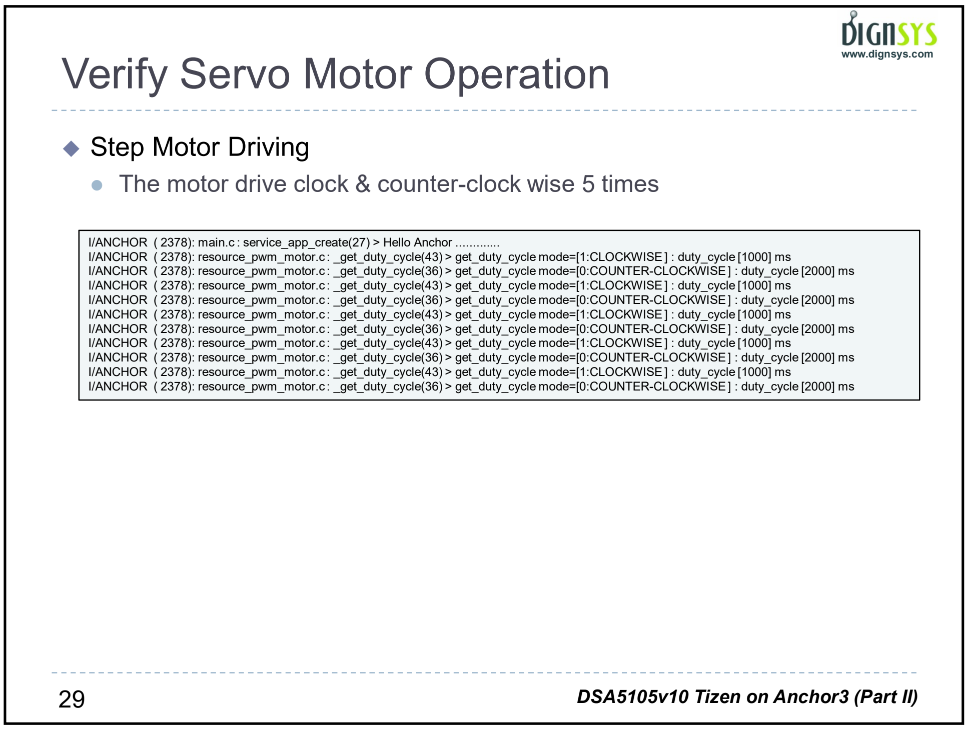
Contents

Practice PWM Motor

- /src/resource_pwm_motor.c

```
void pwm_motor_test_main(void)
{
                int loop;

                /* PWM Servo Motor Test */
                for (loop=0; loop<5;loop++) {
                                resource_motor_driving(1);
                                sleep(1);
                                resource_motor_driving(0);
                                sleep(1);
                }
}
```

Driving motor with clock/counter-clock wise

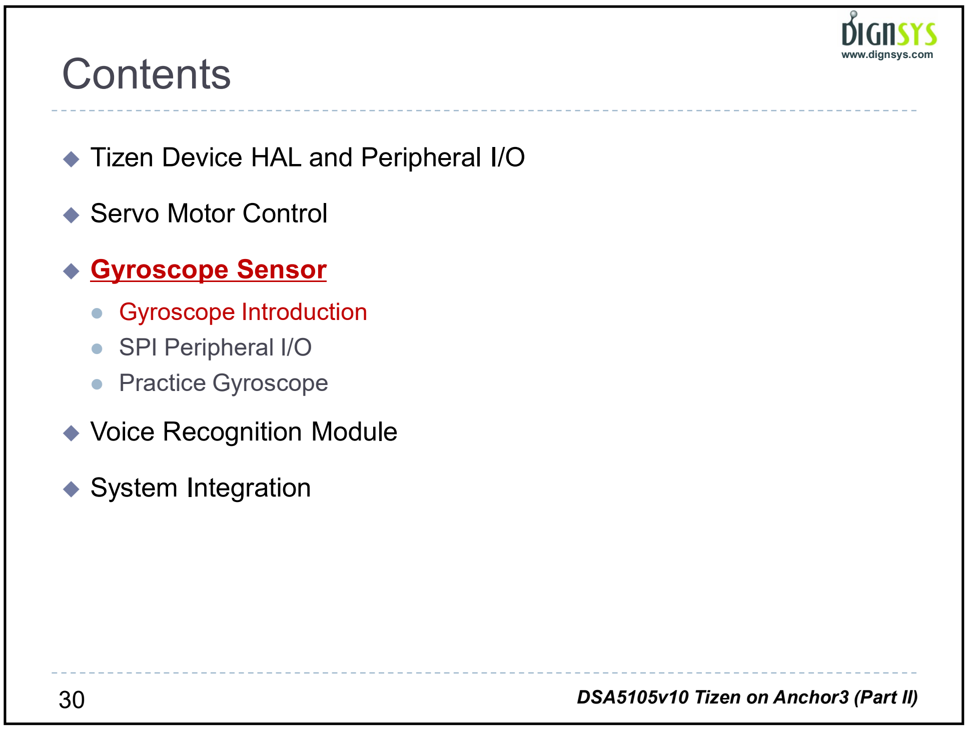


Verify Servo Motor Operation

- Step Motor Driving
  - The motor drive clock & counter-clock wise 5 times

```
I/ANCHOR  ( 2378): main.c : service_app_create(27) > Hello Anchor .............
I/ANCHOR  ( 2378): resource_pwm_motor.c : _get_duty_cycle(43) > get_duty_cycle mode=[1:CLOCKWISE ] : duty_cycle [1000] ms
I/ANCHOR  ( 2378): resource_pwm_motor.c : _get_duty_cycle(36) > get_duty_cycle mode=[0:COUNTER-CLOCKWISE ] : duty_cycle [2000] ms
I/ANCHOR  ( 2378): resource_pwm_motor.c : _get_duty_cycle(43) > get_duty_cycle mode=[1:CLOCKWISE ] : duty_cycle [1000] ms
I/ANCHOR  ( 2378): resource_pwm_motor.c : _get_duty_cycle(36) > get_duty_cycle mode=[0:COUNTER-CLOCKWISE ] : duty_cycle [2000] ms
I/ANCHOR  ( 2378): resource_pwm_motor.c : _get_duty_cycle(43) > get_duty_cycle mode=[1:CLOCKWISE ] : duty_cycle [1000] ms
I/ANCHOR  ( 2378): resource_pwm_motor.c : _get_duty_cycle(36) > get_duty_cycle mode=[0:COUNTER-CLOCKWISE ] : duty_cycle [2000] ms
I/ANCHOR  ( 2378): resource_pwm_motor.c : _get_duty_cycle(43) > get_duty_cycle mode=[1:CLOCKWISE ] : duty_cycle [1000] ms
I/ANCHOR  ( 2378): resource_pwm_motor.c : _get_duty_cycle(36) > get_duty_cycle mode=[0:COUNTER-CLOCKWISE ] : duty_cycle [2000] ms
I/ANCHOR  ( 2378): resource_pwm_motor.c : _get_duty_cycle(43) > get_duty_cycle mode=[1:CLOCKWISE ] : duty_cycle [1000] ms
I/ANCHOR  ( 2378): resource_pwm_motor.c : _get_duty_cycle(36) > get_duty_cycle mode=[0:COUNTER-CLOCKWISE ] : duty_cycle [2000] ms
```

29 ***DSA5105v10 Tizen on Anchor3 (Part II)***

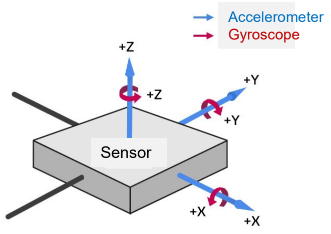


Contents

30 ***DSA5105v10 Tizen on Anchor3 (Part II)***

Gyro Scope

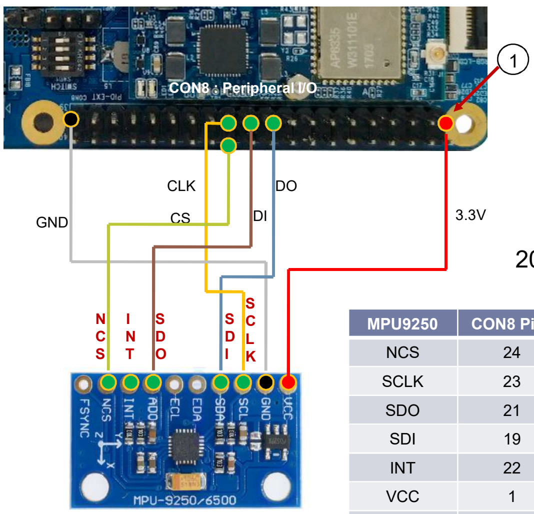
- Gyroscope
  - A device that uses Earth's gravity to help determine orientation
  - It can measure rotation from the balanced position
  - A gyroscope is intended to determine an angular position
- Accelerometer
  - A device designed to measure non-gravitational acceleration
  - Accelerometer gives users a direction of gravity



MPU-9250

- MPU-9250 is a 9-axis MotionTracking device
  - 3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer
  - Multi-chip module (MCM) consisting of two dies integrated into a single package
    - InvenSense 3-Axis gyroscope and the 3-Axis accelerometer
    - AK8963 3-Axis magnetometer
- Feature
  - Internal Digital Motion Processing™ (DMP™)
  - 3-axis MEMS gyroscope
  - 3-axis MEMS accelerometer
  - 3-axis MEMS magnetometer
  - Temperature sensor
- Host Interface
  - 1MHz SPI serial interface for communicating with all registers
  - 400kHz Fast Mode I2C for communicating with all registers

SPI with MPU-9250



20MHz SPI serial interface

MPU9250	CON8 Pin#	Anchor3	Peri-I/O
NCS	24	SPI0_CS	SPI0
SCLK	23	SPI0_CLK	SPI0
SDO	21	SPI0_DI(MISO)	SPI0
SDI	19	SPI0_DO(MOSI)	SPI0
INT	22	GPC8	GPIO72
VCC	1	3.3V	
GND	30, 39	GND	

MPU-9250 SPI Interface

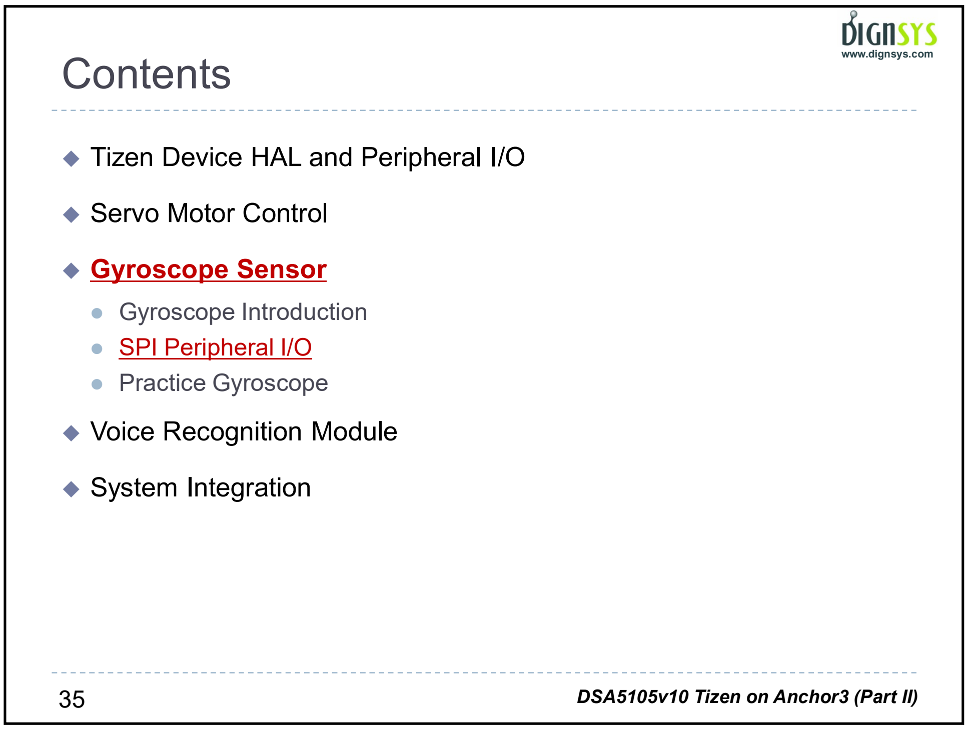
- Data is delivered MSB first and LSB last
  - Data is latched on the rising edge of SCLK
  - Data should be transitioned on the falling edge of SCLK
- The maximum frequency of SCLK is 1MHz
- SPI read and write operations are completed in 16 or more clock cycles (two or more bytes).
  - The first byte contains the SPI Address
    - The first bit of the first byte contains the Read/Write bit and indicates the Read (1) or Write (0) operation.
    - The following 7 bits contain the Register Address
  - The following byte(s) contain(s) the SPI data

SPI Address format

MSB							LSB
R/W	A6	A5	A4	A3	A2	A1	A0

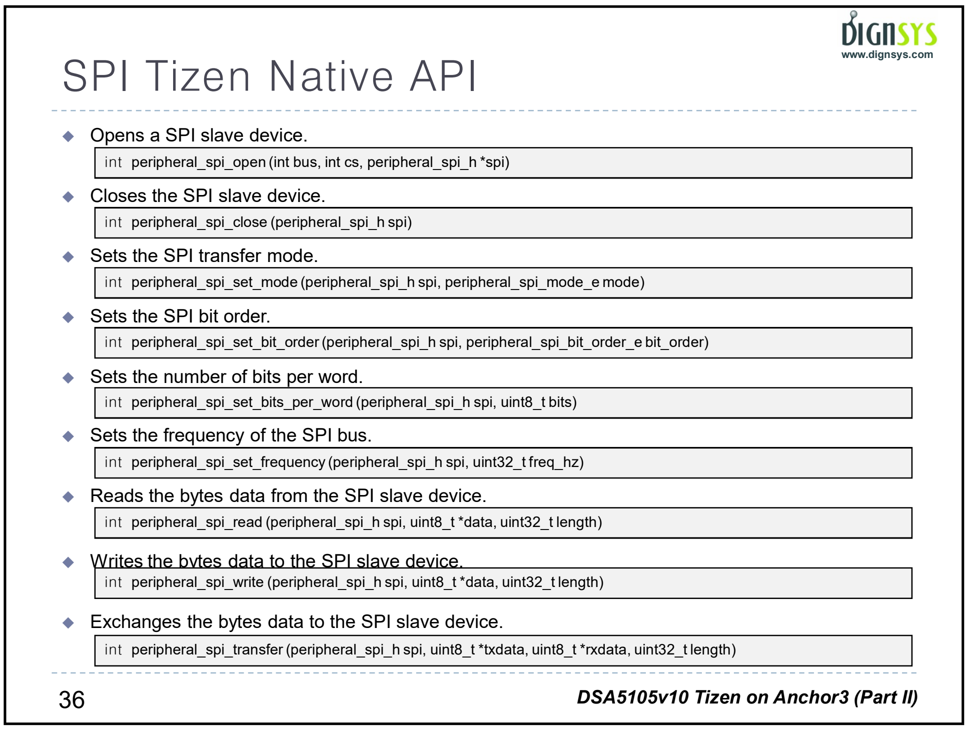
SPI Data format

MSB							LSB
D7	D6	D5	D4	D3	D2	D1	D0



Contents

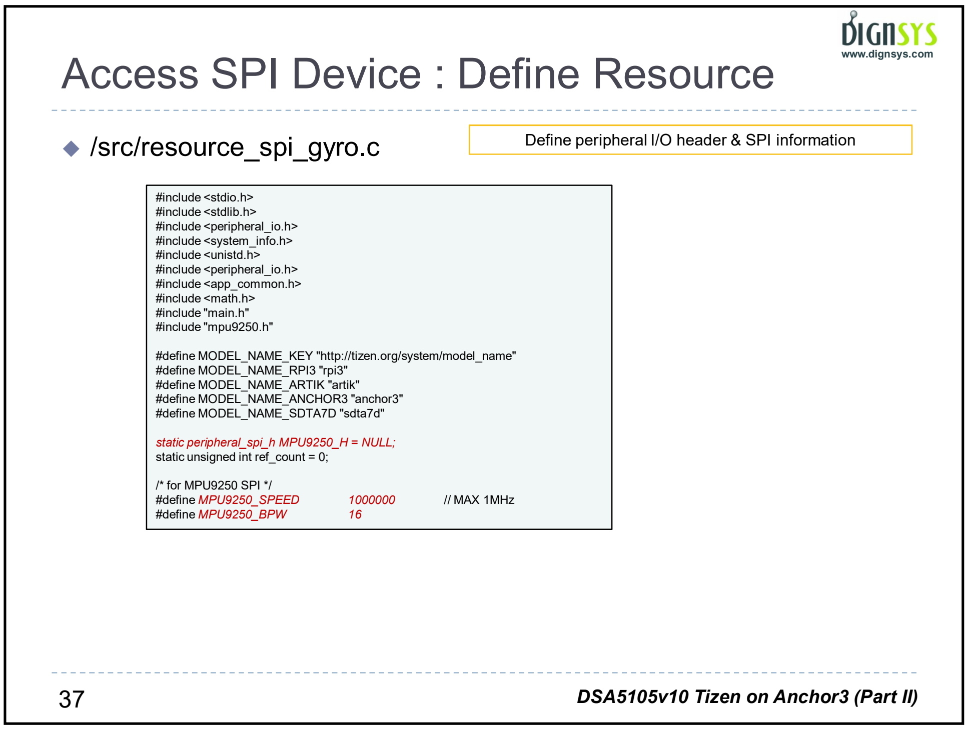
35 *DSA5105v10 Tizen on Anchor3 (Part II)*



SPI Tizen Native API

- Opens a SPI slave device.
  ```
  int peripheral_spi_open (int bus, int cs, peripheral_spi_h *spi)
  ```
- Closes the SPI slave device.
  ```
  int peripheral_spi_close (peripheral_spi_h spi)
  ```
- Sets the SPI transfer mode.
  ```
  int peripheral_spi_set_mode (peripheral_spi_h spi, peripheral_spi_mode_e mode)
  ```
- Sets the SPI bit order.
  ```
  int peripheral_spi_set_bit_order (peripheral_spi_h spi, peripheral_spi_bit_order_e bit_order)
  ```
- Sets the number of bits per word.
  ```
  int peripheral_spi_set_bits_per_word (peripheral_spi_h spi, uint8_t bits)
  ```
- Sets the frequency of the SPI bus.
  ```
  int peripheral_spi_set_frequency (peripheral_spi_h spi, uint32_t freq_hz)
  ```
- Reads the bytes data from the SPI slave device.
  ```
  int peripheral_spi_read (peripheral_spi_h spi, uint8_t *data, uint32_t length)
  ```
- Writes the bytes data to the SPI slave device.
  ```
  int peripheral_spi_write (peripheral_spi_h spi, uint8_t *data, uint32_t length)
  ```
- Exchanges the bytes data to the SPI slave device.
  ```
  int peripheral_spi_transfer (peripheral_spi_h spi, uint8_t *txdata, uint8_t *rxdata, uint32_t length)
  ```

36 *DSA5105v10 Tizen on Anchor3 (Part II)*



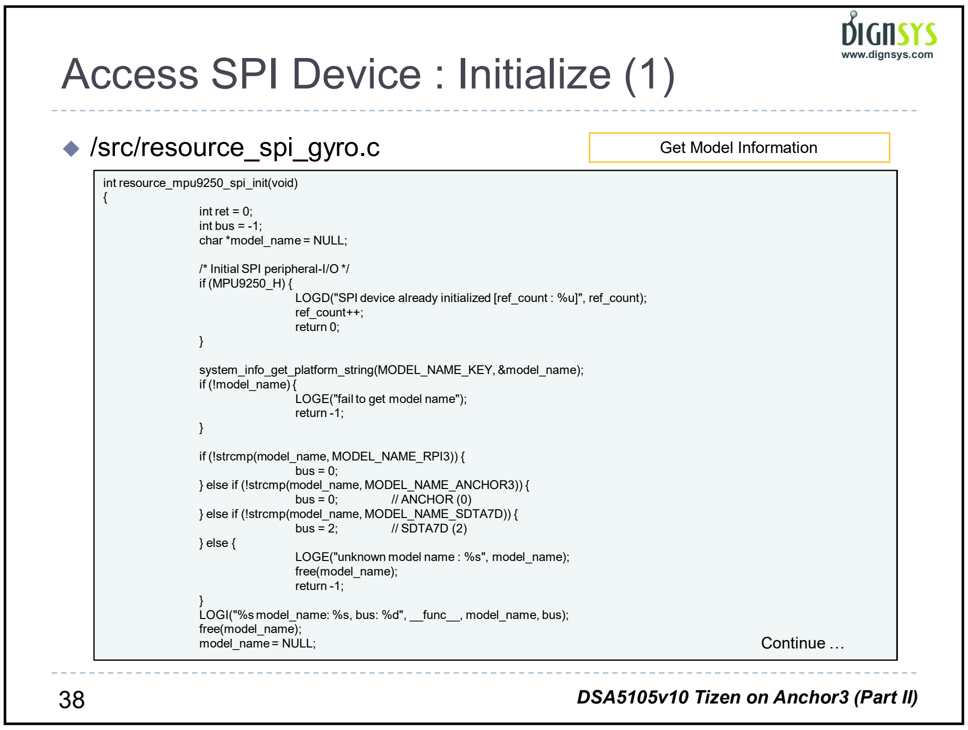
Access SPI Device : Define Resource

◆ /src/resource_spi_gyro.c

Define peripheral I/O header & SPI information

```
#include <stdio.h>
#include <stdlib.h>
#include <peripheral_io.h>
#include <system_info.h>
#include <unistd.h>
#include <peripheral_io.h>
#include <app_common.h>
#include <math.h>
#include "main.h"
#include "mpu9250.h"

#define MODEL_NAME_KEY "http://tizen.org/system/model_name"
#define MODEL_NAME_RPI3 "rpi3"
#define MODEL_NAME_ARTIK "artik"
#define MODEL_NAME_ANCHOR3 "anchor3"
#define MODEL_NAME_SDTA7D "sdta7d"

static peripheral_spi_h MPU9250_H = NULL;
static unsigned int ref_count = 0;

/* for MPU9250 SPI */
#define MPU9250_SPEED          1000000          // MAX 1MHz
#define MPU9250_BPW            16
```

37 DSA5105v10 Tizen on Anchor3 (Part II)



Access SPI Device : Initialize (1)

◆ /src/resource_spi_gyro.c

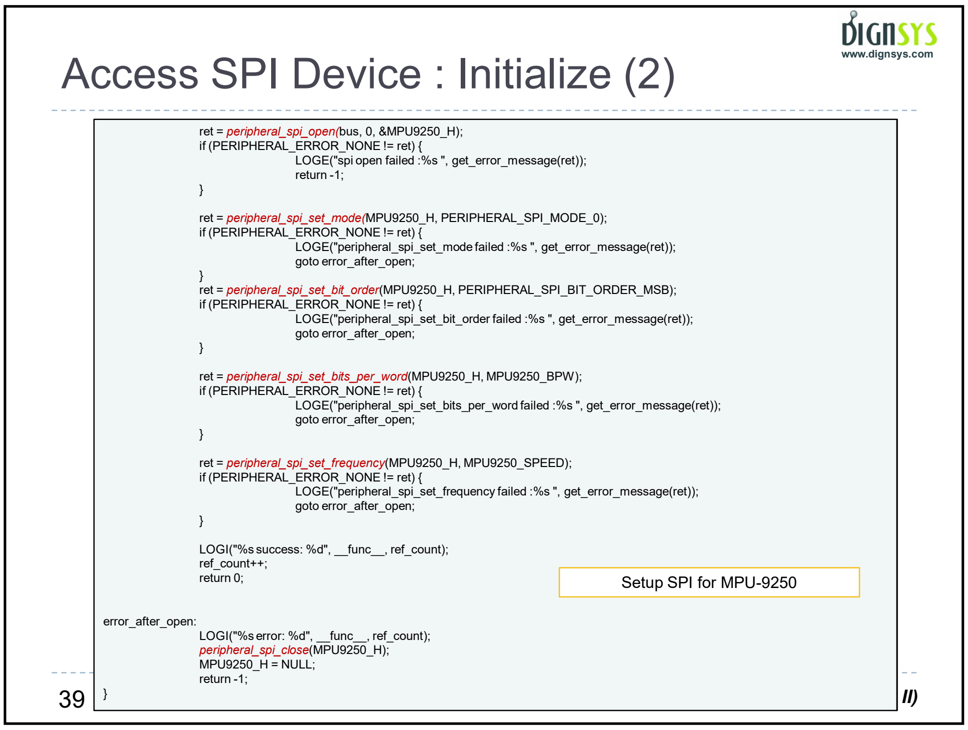
Get Model Information

```
int resource_mpu9250_spi_init(void)
{
        int ret = 0;
        int bus = -1;
        char *model_name = NULL;

        /* Initial SPI peripheral-I/O */
        if (MPU9250_H) {
                LOGD("SPI device already initialized [ref_count : %u]", ref_count);
                ref_count++;
                return 0;
        }

        system_info_get_platform_string(MODEL_NAME_KEY, &model_name);
        if (!model_name) {
                LOGE("fail to get model name");
                return -1;
        }

        if (!strcmp(model_name, MODEL_NAME_RPI3)) {
                bus = 0;
        } else if (!strcmp(model_name, MODEL_NAME_ANCHOR3)) {
                bus = 0;        // ANCHOR (0)
        } else if (!strcmp(model_name, MODEL_NAME_SDTA7D)) {
                bus = 2;        // SDTA7D (2)
        } else {
                LOGE("unknown model name : %s", model_name);
                free(model_name);
                return -1;
        }
        LOGI("%s model_name: %s, bus: %d", __func__, model_name, bus);
        free(model_name);
        model_name = NULL;
```

Continue …

38 DSA5105v10 Tizen on Anchor3 (Part II)

(주) 다인시스

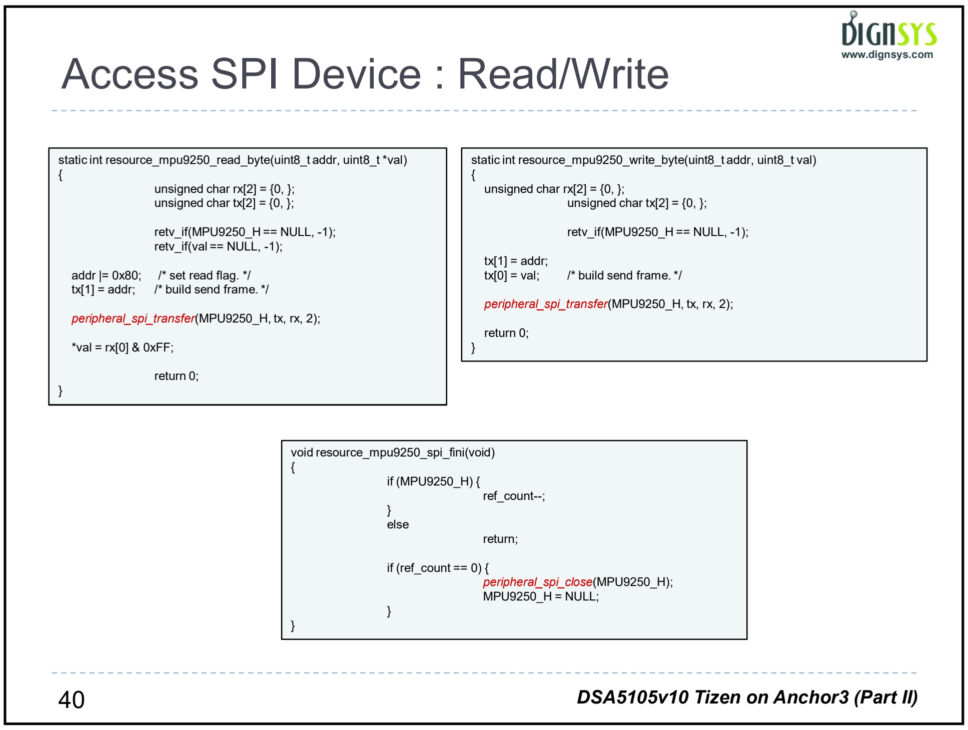

www.dignsys.com

Access SPI Device : Initialize (2)

```
            ret = peripheral_spi_open(bus, 0, &MPU9250_H);
            if (PERIPHERAL_ERROR_NONE != ret) {
                        LOGE("spi open failed :%s ", get_error_message(ret));
                        return -1;
            }

            ret = peripheral_spi_set_mode(MPU9250_H, PERIPHERAL_SPI_MODE_0);
            if (PERIPHERAL_ERROR_NONE != ret) {
                        LOGE("peripheral_spi_set_mode failed :%s ", get_error_message(ret));
                        goto error_after_open;
            }
            ret = peripheral_spi_set_bit_order(MPU9250_H, PERIPHERAL_SPI_BIT_ORDER_MSB);
            if (PERIPHERAL_ERROR_NONE != ret) {
                        LOGE("peripheral_spi_set_bit_order failed :%s ", get_error_message(ret));
                        goto error_after_open;
            }

            ret = peripheral_spi_set_bits_per_word(MPU9250_H, MPU9250_BPW);
            if (PERIPHERAL_ERROR_NONE != ret) {
                        LOGE("peripheral_spi_set_bits_per_word failed :%s ", get_error_message(ret));
                        goto error_after_open;
            }

            ret = peripheral_spi_set_frequency(MPU9250_H, MPU9250_SPEED);
            if (PERIPHERAL_ERROR_NONE != ret) {
                        LOGE("peripheral_spi_set_frequency failed :%s ", get_error_message(ret));
                        goto error_after_open;
            }

            LOGI("%s success: %d", __func__, ref_count);
            ref_count++;
            return 0;

error_after_open:
            LOGI("%s error: %d", __func__, ref_count);
            peripheral_spi_close(MPU9250_H);
            MPU9250_H = NULL;
            return -1;
}
```

Setup SPI for MPU-9250

39 II)


www.dignsys.com

Access SPI Device : Read/Write

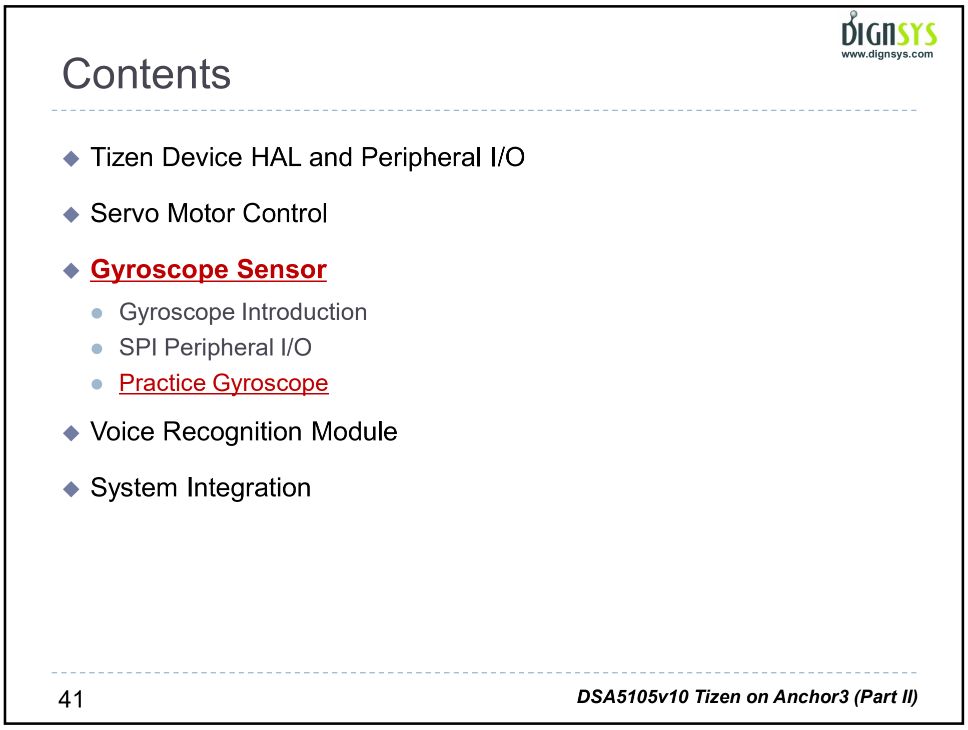
```
static int resource_mpu9250_read_byte(uint8_t addr, uint8_t *val)
{
            unsigned char rx[2] = {0, };
            unsigned char tx[2] = {0, };

            retv_if(MPU9250_H == NULL, -1);
            retv_if(val == NULL, -1);

    addr |= 0x80;    /* set read flag. */
    tx[1] = addr;    /* build send frame. */

    peripheral_spi_transfer(MPU9250_H, tx, rx, 2);

    *val = rx[0] & 0xFF;

            return 0;
}
```

```
static int resource_mpu9250_write_byte(uint8_t addr, uint8_t val)
{
    unsigned char rx[2] = {0, };
            unsigned char tx[2] = {0, };

            retv_if(MPU9250_H == NULL, -1);

    tx[1] = addr;
    tx[0] = val;      /* build send frame. */

    peripheral_spi_transfer(MPU9250_H, tx, rx, 2);

    return 0;
}
```

```
void resource_mpu9250_spi_fini(void)
{
            if (MPU9250_H) {
                        ref_count--;
            }
            else
                        return;

            if (ref_count == 0) {
                        peripheral_spi_close(MPU9250_H);
                        MPU9250_H = NULL;
            }
}
```

40 *DSA5105v10 Tizen on Anchor3 (Part II)*

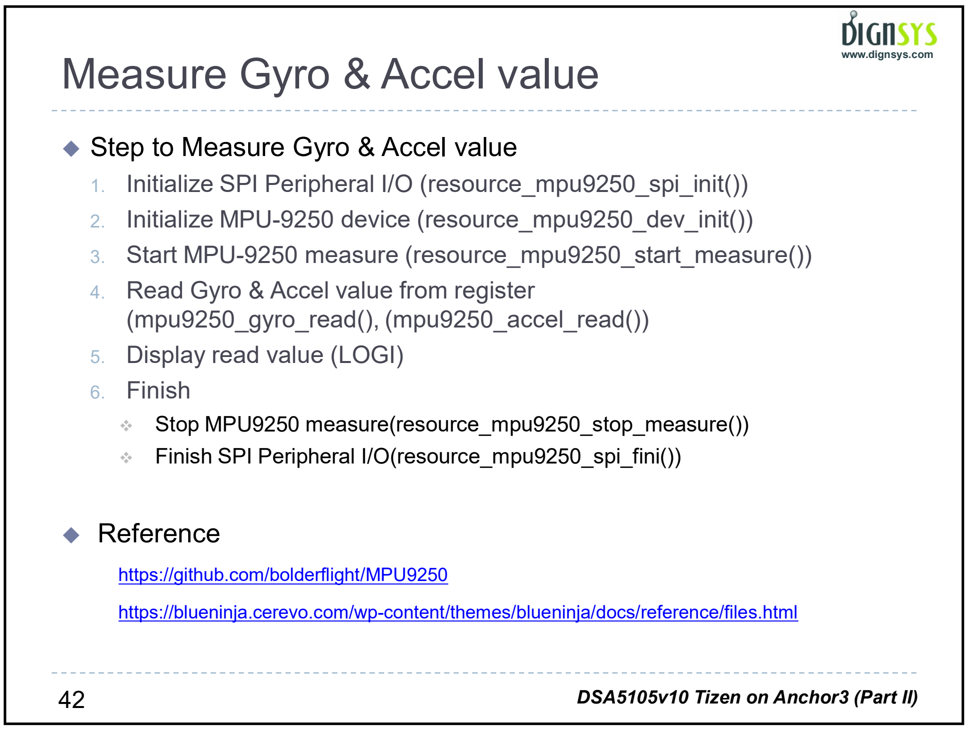
(주) 다인시스



Contents

DIGNSYS www.dignsys.com

41 ***DSA5105v10 Tizen on Anchor3 (Part II)***



Measure Gyro & Accel value

DIGNSYS www.dignsys.com

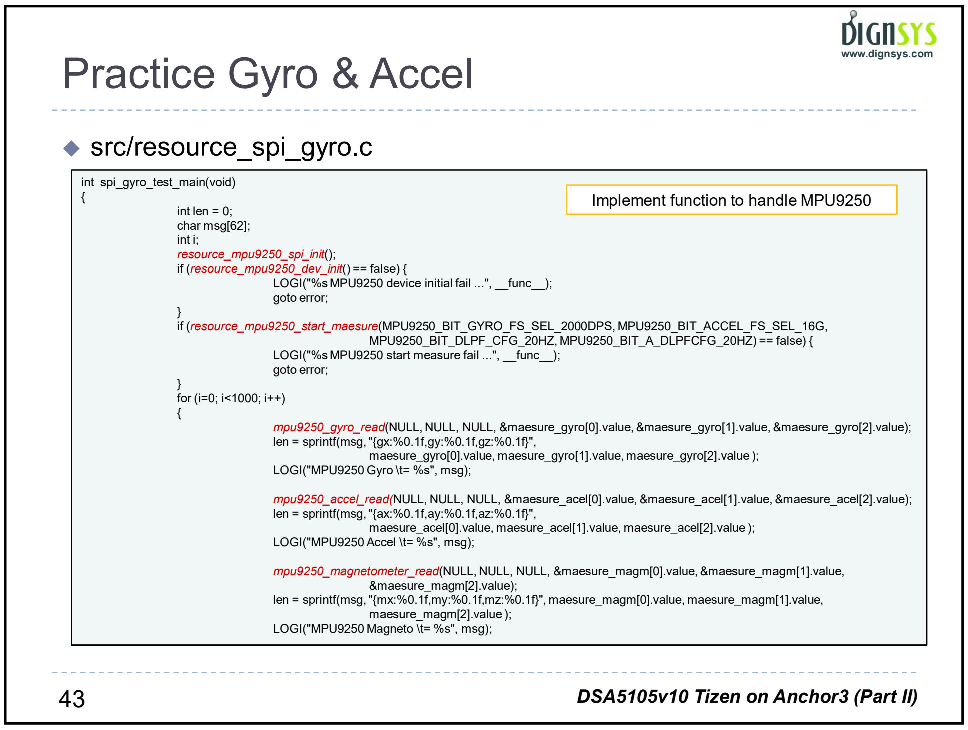
- Step to Measure Gyro & Accel value
  1. Initialize SPI Peripheral I/O (resource_mpu9250_spi_init())
  2. Initialize MPU-9250 device (resource_mpu9250_dev_init())
  3. Start MPU-9250 measure (resource_mpu9250_start_measure())
  4. Read Gyro & Accel value from register (mpu9250_gyro_read(), (mpu9250_accel_read())
  5. Display read value (LOGI)
  6. Finish
     - Stop MPU9250 measure(resource_mpu9250_stop_measure())
     - Finish SPI Peripheral I/O(resource_mpu9250_spi_fini())

- Reference

  https://github.com/bolderflight/MPU9250

  https://blueninja.cerevo.com/wp-content/themes/blueninja/docs/reference/files.html

42 ***DSA5105v10 Tizen on Anchor3 (Part II)***



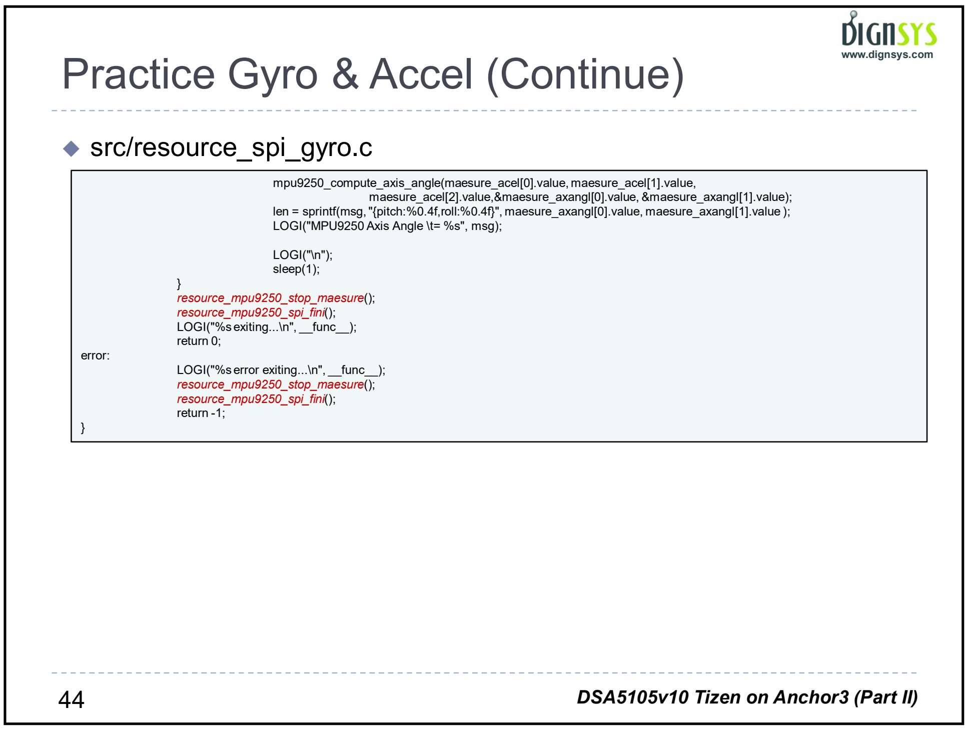
Practice Gyro & Accel

◆ src/resource_spi_gyro.c

Implement function to handle MPU9250

```
int  spi_gyro_test_main(void)
{
        int len = 0;
        char msg[62];
        int i;
        resource_mpu9250_spi_init();
        if (resource_mpu9250_dev_init() == false) {
                LOGI("%s MPU9250 device initial fail ...", __func__);
                goto error;
        }
        if (resource_mpu9250_start_maesure(MPU9250_BIT_GYRO_FS_SEL_2000DPS, MPU9250_BIT_ACCEL_FS_SEL_16G,
                        MPU9250_BIT_DLPF_CFG_20HZ, MPU9250_BIT_A_DLPFCFG_20HZ) == false) {
                LOGI("%s MPU9250 start measure fail ...", __func__);
                goto error;
        }
        for (i=0; i<1000; i++)
        {
                mpu9250_gyro_read(NULL, NULL, NULL, &maesure_gyro[0].value, &maesure_gyro[1].value, &maesure_gyro[2].value);
                len = sprintf(msg, "{gx:%0.1f,gy:%0.1f,gz:%0.1f}",
                        maesure_gyro[0].value, maesure_gyro[1].value, maesure_gyro[2].value );
                LOGI("MPU9250 Gyro \t= %s", msg);

                mpu9250_accel_read(NULL, NULL, NULL, &maesure_acel[0].value, &maesure_acel[1].value, &maesure_acel[2].value);
                len = sprintf(msg, "{ax:%0.1f,ay:%0.1f,az:%0.1f}",
                        maesure_acel[0].value, maesure_acel[1].value, maesure_acel[2].value );
                LOGI("MPU9250 Accel \t= %s", msg);

                mpu9250_magnetometer_read(NULL, NULL, NULL, &maesure_magm[0].value, &maesure_magm[1].value,
                        &maesure_magm[2].value);
                len = sprintf(msg, "{mx:%0.1f,my:%0.1f,mz:%0.1f}", maesure_magm[0].value, maesure_magm[1].value,
                        maesure_magm[2].value );
                LOGI("MPU9250 Magneto \t= %s", msg);
```

43 *DSA5105v10 Tizen on Anchor3 (Part II)*

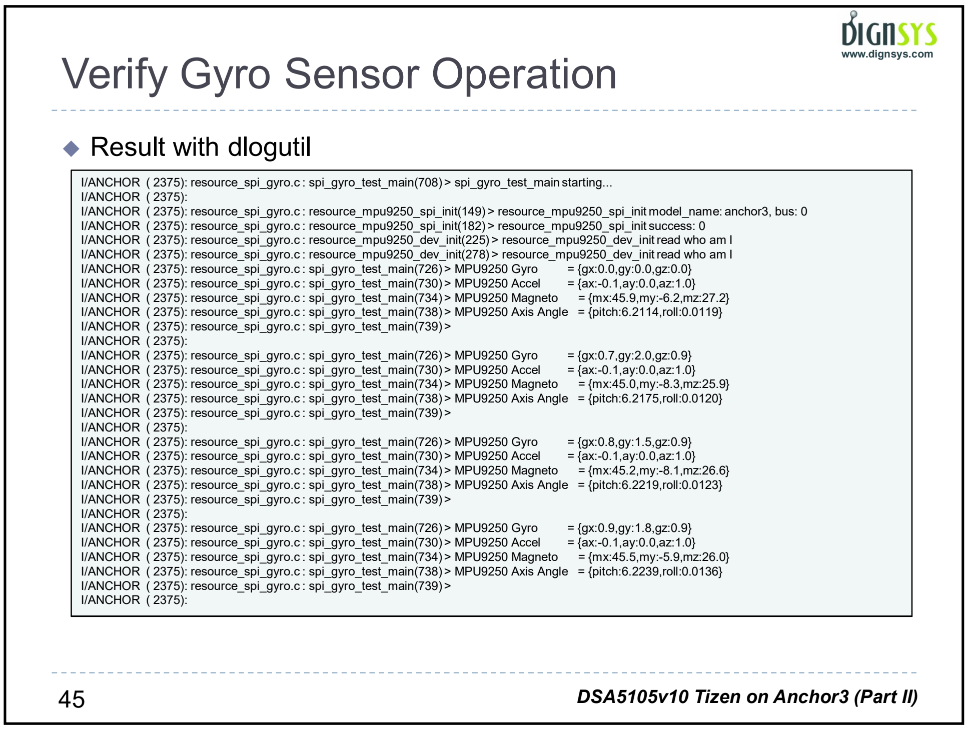


Practice Gyro & Accel (Continue)

◆ src/resource_spi_gyro.c

```
                mpu9250_compute_axis_angle(maesure_acel[0].value, maesure_acel[1].value,
                        maesure_acel[2].value,&maesure_axangl[0].value, &maesure_axangl[1].value);
                len = sprintf(msg, "{pitch:%0.4f,roll:%0.4f}", maesure_axangl[0].value, maesure_axangl[1].value );
                LOGI("MPU9250 Axis Angle \t= %s", msg);

                LOGI("\n");
                sleep(1);
        }
        resource_mpu9250_stop_maesure();
        resource_mpu9250_spi_fini();
        LOGI("%s exiting...\n", __func__);
        return 0;
error:
        LOGI("%s error exiting...\n", __func__);
        resource_mpu9250_stop_maesure();
        resource_mpu9250_spi_fini();
        return -1;
}
```

44 *DSA5105v10 Tizen on Anchor3 (Part II)*

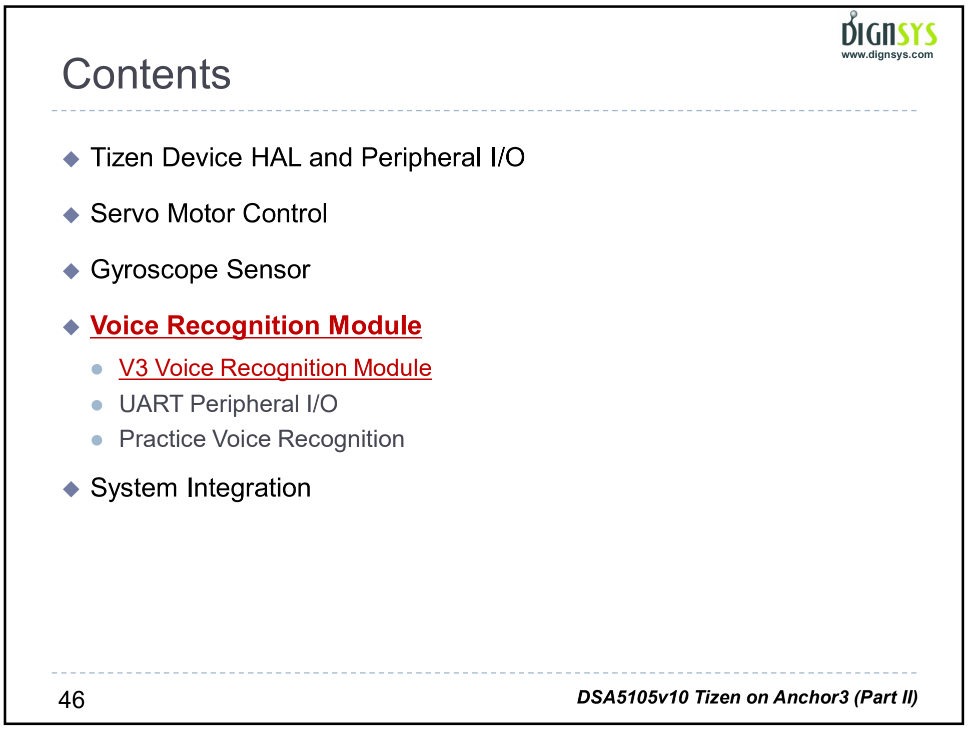


Verify Gyro Sensor Operation

- Result with dlogutil

```
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(708) > spi_gyro_test_main starting...
I/ANCHOR  ( 2375):
I/ANCHOR  ( 2375): resource_spi_gyro.c : resource_mpu9250_spi_init(149) > resource_mpu9250_spi_init model_name: anchor3, bus: 0
I/ANCHOR  ( 2375): resource_spi_gyro.c : resource_mpu9250_spi_init(182) > resource_mpu9250_spi_init success: 0
I/ANCHOR  ( 2375): resource_spi_gyro.c : resource_mpu9250_dev_init(225) > resource_mpu9250_dev_init read who am I
I/ANCHOR  ( 2375): resource_spi_gyro.c : resource_mpu9250_dev_init(278) > resource_mpu9250_dev_init read who am I
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(726) > MPU9250 Gyro       = {gx:0.0,gy:0.0,gz:0.0}
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(730) > MPU9250 Accel      = {ax:-0.1,ay:0.0,az:1.0}
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(734) > MPU9250 Magneto     = {mx:45.9,my:-6.2,mz:27.2}
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(738) > MPU9250 Axis Angle  = {pitch:6.2114,roll:0.0119}
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(739) >
I/ANCHOR  ( 2375):
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(726) > MPU9250 Gyro       = {gx:0.7,gy:2.0,gz:0.9}
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(730) > MPU9250 Accel      = {ax:-0.1,ay:0.0,az:1.0}
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(734) > MPU9250 Magneto     = {mx:45.0,my:-8.3,mz:25.9}
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(738) > MPU9250 Axis Angle  = {pitch:6.2175,roll:0.0120}
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(739) >
I/ANCHOR  ( 2375):
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(726) > MPU9250 Gyro       = {gx:0.8,gy:1.5,gz:0.9}
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(730) > MPU9250 Accel      = {ax:-0.1,ay:0.0,az:1.0}
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(734) > MPU9250 Magneto     = {mx:45.2,my:-8.1,mz:26.6}
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(738) > MPU9250 Axis Angle  = {pitch:6.2219,roll:0.0123}
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(739) >
I/ANCHOR  ( 2375):
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(726) > MPU9250 Gyro       = {gx:0.9,gy:1.8,gz:0.9}
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(730) > MPU9250 Accel      = {ax:-0.1,ay:0.0,az:1.0}
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(734) > MPU9250 Magneto     = {mx:45.5,my:-5.9,mz:26.0}
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(738) > MPU9250 Axis Angle  = {pitch:6.2239,roll:0.0136}
I/ANCHOR  ( 2375): resource_spi_gyro.c : spi_gyro_test_main(739) >
I/ANCHOR  ( 2375):
```

45 ***DSA5105v10 Tizen on Anchor3 (Part II)***

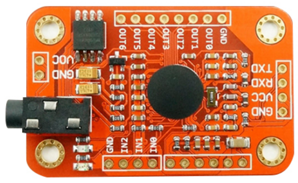


Contents

46 ***DSA5105v10 Tizen on Anchor3 (Part II)***

ELECHOUSE Voice Recognition Module

- Compact and easy-control speaking recognition board.
- This product is a speaker-dependent voice recognition module.
- It supports up to 80 voice commands in all.
- Max 7 voice commands could work at the same time.



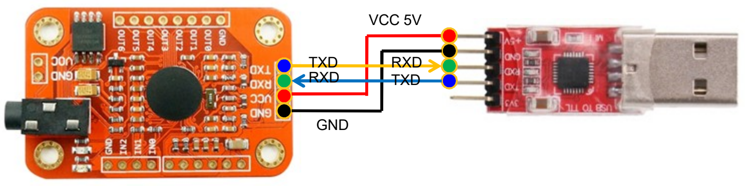
Terminology

- VR3 : Voice Recognition Module V3
- Recognizer
  - A container where acting voice commands (max 7) were loaded.
  - Core part of voice recognition module.
- Recognizer index
  - Max 7 voice commands could be supported in the recognizer.
  - One index corresponds to one region: 0~6
- Train
  - The process of recording your voice commands
- Load
  - Copy trained voice to recognizer
- Voice Command Record
  - The trained voice command store in flash, number from 0 to 79
- Signature
  - Text comment for record
- Group
  - Help to manage records, each group 7 records.
  - System group and user group are supported.

Connect to PC

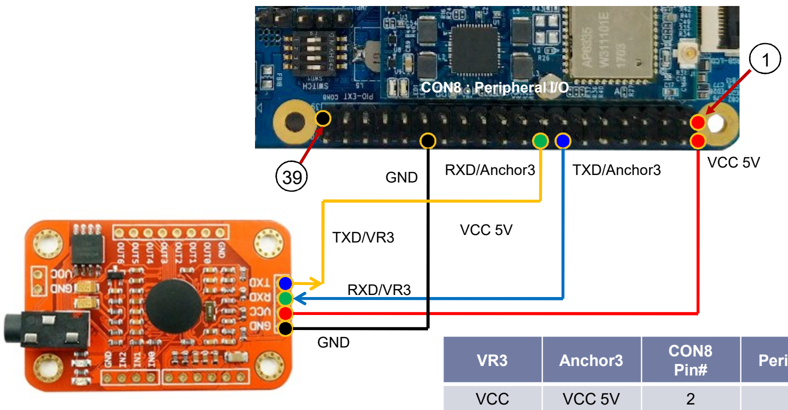
◆ Connect VR3 module to PC with USB to UART TTL Converter

VR3	Converter
VCC	VCC
GND	GND
TXD	RXD
RXD	TXD

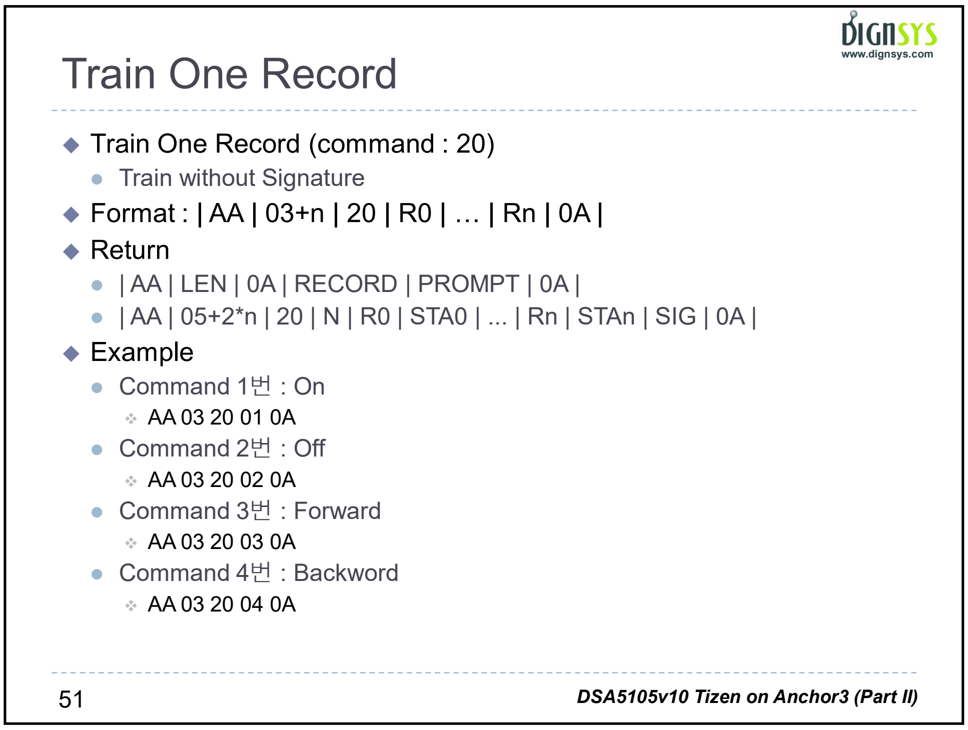


Connect to Anchor3

◆ Connect VR3 module to Anchor3



VR3	Anchor3	CON8 Pin#	Peri-I/O
VCC	VCC 5V	2	
GND	GND	30. 39	
TXD	RXD	18	UART1 (ttyS1)
RXD	TXD	16	

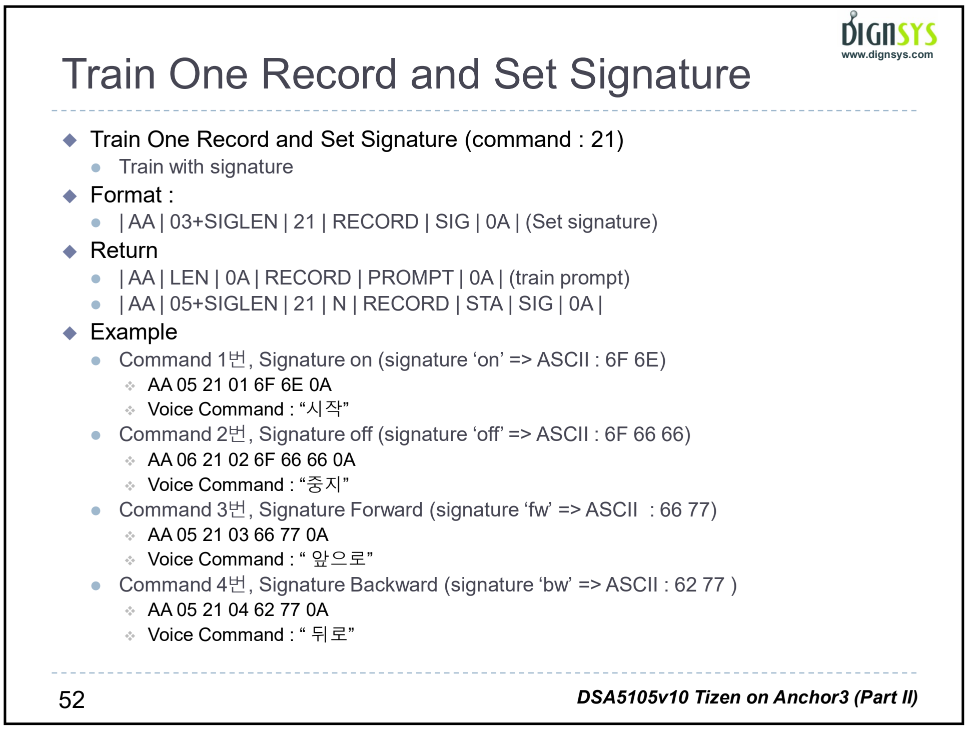


DIGNSYS
www.dignsys.com

Train One Record

- Train One Record (command : 20)
  - Train without Signature
- Format : | AA | 03+n | 20 | R0 | … | Rn | 0A |
- Return
  - | AA | LEN | 0A | RECORD | PROMPT | 0A |
  - | AA | 05+2*n | 20 | N | R0 | STA0 | ... | Rn | STAn | SIG | 0A |
- Example
  - Command 1번 : On
    - AA 03 20 01 0A
  - Command 2번 : Off
    - AA 03 20 02 0A
  - Command 3번 : Forward
    - AA 03 20 03 0A
  - Command 4번 : Backword
    - AA 03 20 04 0A

51 ***DSA5105v10 Tizen on Anchor3 (Part II)***



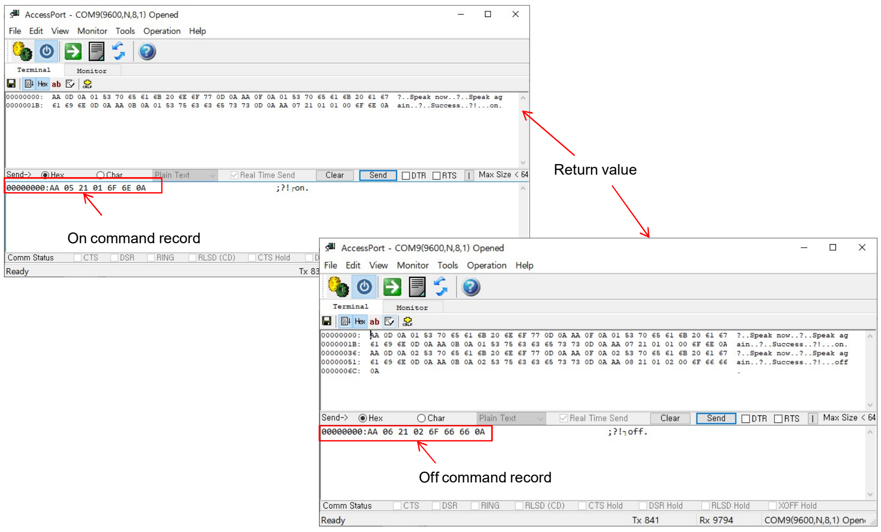
DIGNSYS
www.dignsys.com

Train One Record and Set Signature

- Train One Record and Set Signature (command : 21)
  - Train with signature
- Format :
  - | AA | 03+SIGLEN | 21 | RECORD | SIG | 0A | (Set signature)
- Return
  - | AA | LEN | 0A | RECORD | PROMPT | 0A | (train prompt)
  - | AA | 05+SIGLEN | 21 | N | RECORD | STA | SIG | 0A |
- Example
  - Command 1번, Signature on (signature ‘on’ => ASCII : 6F 6E)
    - AA 05 21 01 6F 6E 0A
    - Voice Command : “시작”
  - Command 2번, Signature off (signature ‘off’ => ASCII : 6F 66 66)
    - AA 06 21 02 6F 66 66 0A
    - Voice Command : “중지”
  - Command 3번, Signature Forward (signature ‘fw’ => ASCII : 66 77)
    - AA 05 21 03 66 77 0A
    - Voice Command : “ 앞으로”
  - Command 4번, Signature Backward (signature ‘bw’ => ASCII : 62 77)
    - AA 05 21 04 62 77 0A
    - Voice Command : “ 뒤로”

52 ***DSA5105v10 Tizen on Anchor3 (Part II)***

Train On/Off Record Example


AccessPort - COM9(9600,N,8,1) Opened
File Edit View Monitor Tools Operation Help
Terminal
Monitor
Return value
On command record
Off command record

```
00000000:  AA 0D 0A 01 53 70 65 61 6B 20 6E 6F 77 0D 0A AA 0F 0A 01 53 70 65 61 6B 20 61 67  ?..Speak now..?..Speak ag
0000001B:  61 69 6E 0D 0A AA 0B 0A 01 53 75 63 63 65 73 73 0D 0A AA 07 21 01 01 00 6F 6E 0A  ain..?..Success..?!...on.
```

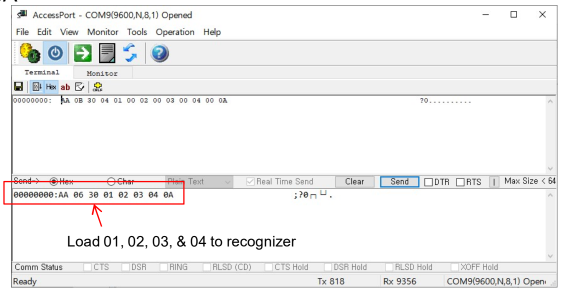
```
Send-> (●) Hex   ( ) Char   Plain Text   [✓] Real Time Send   Clear   Send   [ ] DTR  [ ] RTS   Max Size < 64
00000000:AA 05 21 01 6F 6E 0A                    ;?!┌on.
```

```
00000000:  AA 0D 0A 01 53 70 65 61 6B 20 6E 6F 77 0D 0A AA 0F 0A 01 53 70 65 61 6B 20 61 67  ?..Speak now..?..Speak ag
0000001B:  61 69 6E 0D 0A AA 0B 0A 01 53 75 63 63 65 73 73 0D 0A AA 07 21 01 01 00 6F 6E 0A  ain..?..Success..?!...on.
00000036:  AA 0D 0A 02 53 70 65 61 6B 20 6E 6F 77 0D 0A AA 0F 0A 02 53 70 65 61 6B 20 61 67  ?..Speak now..?..Speak ag
00000051:  61 69 6E 0D 0A AA 0B 0A 02 53 75 63 63 65 73 73 0D 0A AA 08 21 01 02 00 6F 66 66  ain..?..Success..?!...off
0000006C:  0A                                                                                .
```

```
Send-> (●) Hex   ( ) Char   Plain Text   [✓] Real Time Send   Clear   Send   [ ] DTR  [ ] RTS   Max Size < 64
00000000:AA 06 21 02 6F 66 66 0A                 ;?!┐off.

Comm Status  [ ] CTS  [ ] DSR  [ ] RING  [ ] RLSD (CD)  [ ] CTS Hold  [ ] DSR Hold  [ ] RLSD Hold  [ ] XOFF Hold
Ready                                   Tx 841        Rx 9794       COM9(9600,N,8,1) Open
```

Load Voice Record

- Load Voice Record (command : 30)
  - Load Voice Record to Recognizer
- Format : AA | 3+n | R0 | …. | Rn | 0A |
- Return : | AA | 3+2n | 30 | N | R0 | STA0 | ... | Rn | STAn | 0A |
- Example
  - Load 01, 02, 03, 04 to recognizer
    - AA 06 30 01 02 03 04 0A

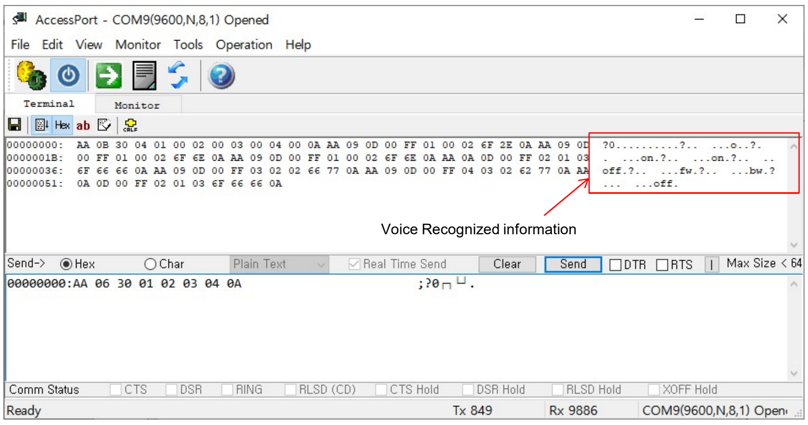

AccessPort - COM9(9600,N,8,1) Opened
File Edit View Monitor Tools Operation Help
Terminal
Monitor
Load 01, 02, 03, & 04 to recognizer

```
00000000:  AA 0B 30 04 01 00 02 00 03 00 04 00 0A          ?0..........
```

```
Send-> (●) Hex   ( ) Char   Plain Text   [✓] Real Time Send   Clear   Send   [ ] DTR  [ ] RTS   Max Size < 64
00000000:AA 06 30 01 02 03 04 0A                 ;?0┌└.

Comm Status  [ ] CTS  [ ] DSR  [ ] RING  [ ] RLSD (CD)  [ ] CTS Hold  [ ] DSR Hold  [ ] RLSD Hold  [ ] XOFF Hold
Ready                                   Tx 818        Rx 9356       COM9(9600,N,8,1) Open
```

Voice Recognition

- Speak Voice Command after Load Voice Record to Recognizer
- Voice Command : “시작”, “중지”, “앞으로”, “뒤로”


AccessPort - COM9(9600,N,8,1) Opened
File Edit View Monitor Tools Operation Help
Terminal
Monitor
Voice Recognized information
Send-> Hex Char Plain Text Real Time Send Clear Send DTR RTS Max Size < 64
Comm Status CTS DSR RING RLSD (CD) CTS Hold DSR Hold RLSD Hold XOFF Hold

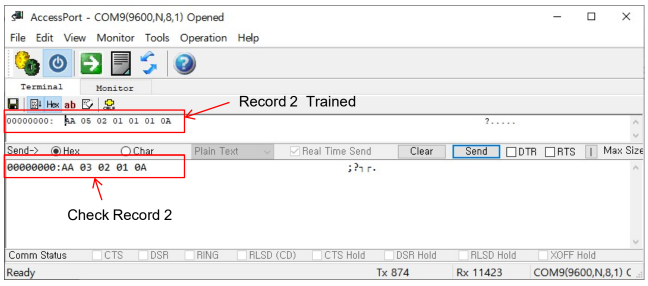
```
00000000:  AA 0B 30 04 01 00 02 00 03 00 04 00 0A AA 09 0D 00 FF 01 00 02 6F 2E 0A AA 09 0D  ?0..........?..  ...o..?.
0000001B:  00 FF 01 00 02 6F 6E 0A AA 09 0D 00 FF 01 00 02 6F 6E 0A AA 0A 0D 00 FF 02 01 03  .  ...on.?..  ...on.?..  ..
00000036:  6F 66 66 0A AA 09 0D 00 FF 03 02 02 66 77 0A AA 09 0D 00 FF 04 03 02 62 77 0A AA  off.?..  ...fw.?..  ...bw.?
00000051:  0A 0D 00 FF 02 01 03 6F 66 66 0A                                                  ...  ...off.
```

```
00000000:AA 06 30 01 02 03 04 0A                    ;?0ㅁ␣.
```

```
Ready                                   Tx 849        Rx 9886       COM9(9600,N,8,1) Open
```

Check Record Train Status

- Check Record Train Status (Command : 02)
  - Use "Check Record Train Status" command to check if the record is trained.
- Format:
  - *Check all records :* | AA | 03 | 02 | FF| 0A |
  - *Check specified records :* | AA | 03+n | 02 | R0 | ... | Rn | 0A |
- Return: | AA | 5+2*n | 02 | N | R0 | STA | ... | Rn | STA | 0A |


AccessPort - COM9(9600,N,8,1) Opened
File Edit View Monitor Tools Operation Help
Terminal
Monitor
Record 2 Trained
Check Record 2
Send-> Hex Char Plain Text Real Time Send Clear Send DTR RTS Max Size
Comm Status CTS DSR RING RLSD (CD) CTS Hold DSR Hold RLSD Hold XOFF Hold

```
00000000:  AA 05 02 01 01 01 0A                    ?.....
```

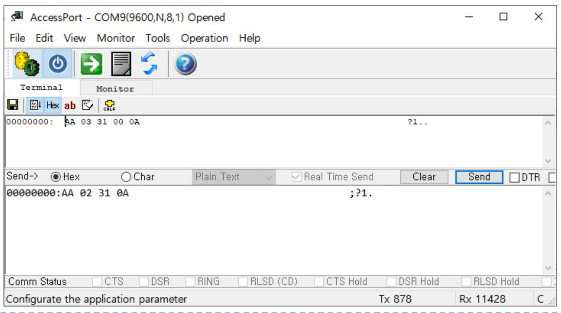
```
00000000:AA 03 02 01 0A                    ;?ㄱㄷ.
```

```
Ready                                   Tx 874        Rx 11423      COM9(9600,N,8,1) C
```

(주) 다인시스

Clear Recognizer

- Clear Recognizer (Command : 31)
  - Stop recognizing, and empty recognizer of Voice Recognition Module.
- Format:
  - | AA | 02 | 31 | 0A |
- Return:
  - | AA | 03 | 31 | 00 | 0A |

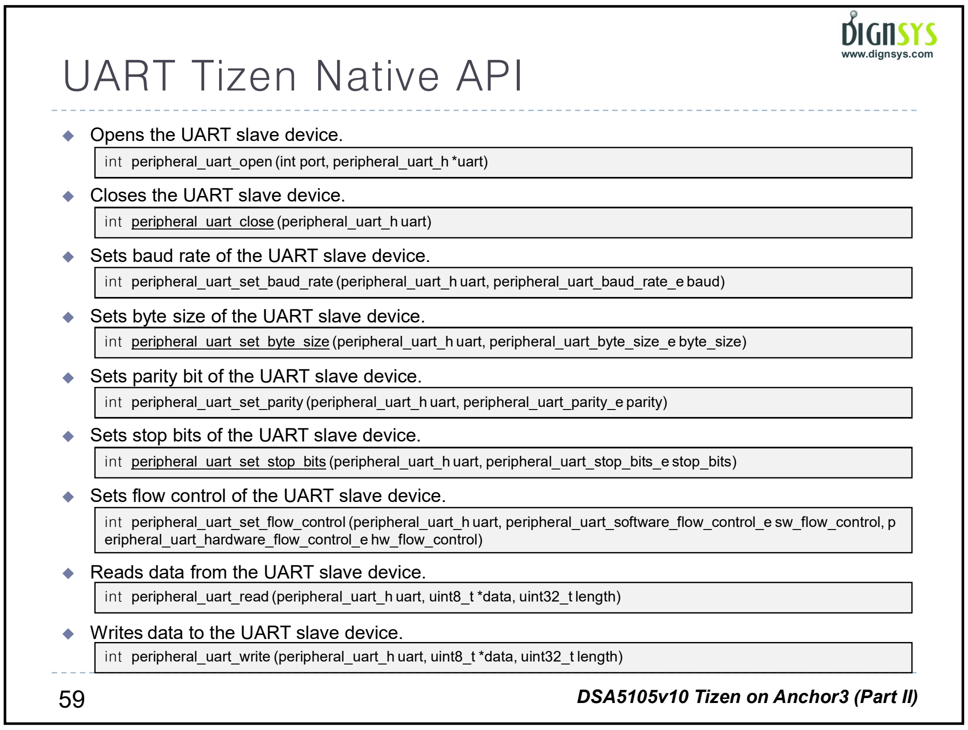

AccessPort - COM9(9600,N,8,1) Opened
File Edit View Monitor Tools Operation Help
Terminal
Monitor
Send-> Hex Char
Plain Text
Real Time Send
Clear
Send
DTR
Comm Status CTS DSR RING RLSD (CD) CTS Hold DSR Hold RLSD Hold

```
00000000:  AA 03 31 00 0A                                        ?1..
```

```
00000000:AA 02 31 0A                                ;?1.
```

Configurate the application parameter   Tx 878   Rx 11428

Contents



UART Tizen Native API

- Opens the UART slave device.

```
int peripheral_uart_open (int port, peripheral_uart_h *uart)
```

- Closes the UART slave device.

```
int peripheral_uart_close (peripheral_uart_h uart)
```

- Sets baud rate of the UART slave device.

```
int peripheral_uart_set_baud_rate (peripheral_uart_h uart, peripheral_uart_baud_rate_e baud)
```

- Sets byte size of the UART slave device.

```
int peripheral_uart_set_byte_size (peripheral_uart_h uart, peripheral_uart_byte_size_e byte_size)
```

- Sets parity bit of the UART slave device.

```
int peripheral_uart_set_parity (peripheral_uart_h uart, peripheral_uart_parity_e parity)
```

- Sets stop bits of the UART slave device.

```
int peripheral_uart_set_stop_bits (peripheral_uart_h uart, peripheral_uart_stop_bits_e stop_bits)
```

- Sets flow control of the UART slave device.

```
int peripheral_uart_set_flow_control (peripheral_uart_h uart, peripheral_uart_software_flow_control_e sw_flow_control, p
eripheral_uart_hardware_flow_control_e hw_flow_control)
```

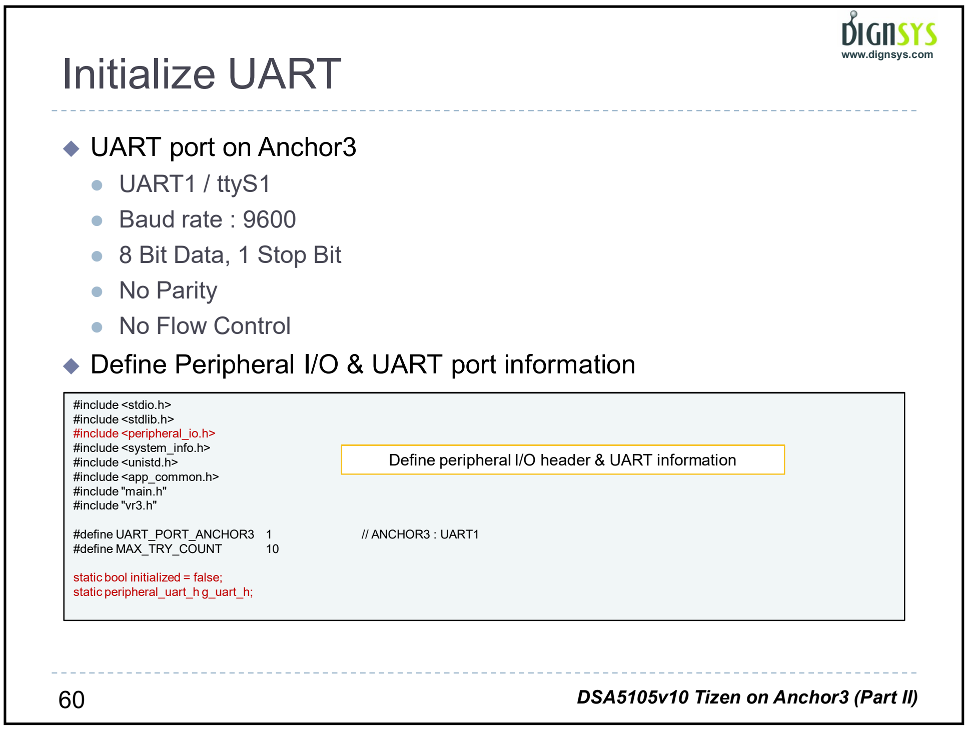
- Reads data from the UART slave device.

```
int peripheral_uart_read (peripheral_uart_h uart, uint8_t *data, uint32_t length)
```

- Writes data to the UART slave device.

```
int peripheral_uart_write (peripheral_uart_h uart, uint8_t *data, uint32_t length)
```

59 ***DSA5105v10 Tizen on Anchor3 (Part II)***



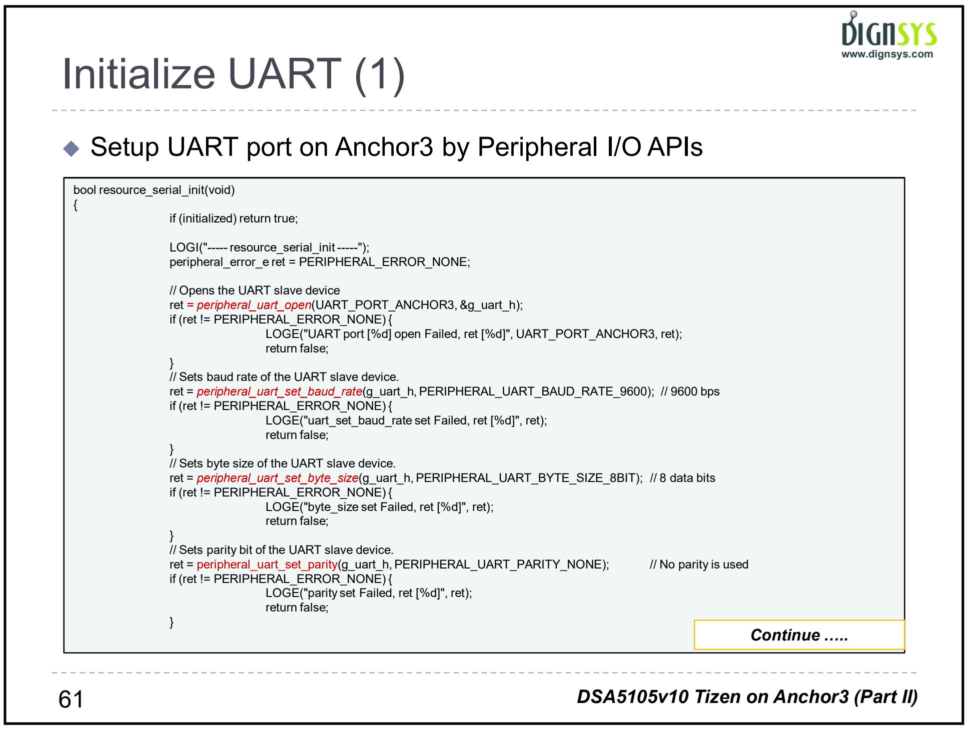
Initialize UART

- UART port on Anchor3
  - UART1 / ttyS1
  - Baud rate : 9600
  - 8 Bit Data, 1 Stop Bit
  - No Parity
  - No Flow Control
- Define Peripheral I/O & UART port information

```
#include <stdio.h>
#include <stdlib.h>
#include <peripheral_io.h>
#include <system_info.h>
#include <unistd.h>
#include <app_common.h>
#include "main.h"
#include "vr3.h"

#define UART_PORT_ANCHOR3   1                // ANCHOR3 : UART1
#define MAX_TRY_COUNT       10

static bool initialized = false;
static peripheral_uart_h g_uart_h;
```

Define peripheral I/O header & UART information

60 ***DSA5105v10 Tizen on Anchor3 (Part II)***



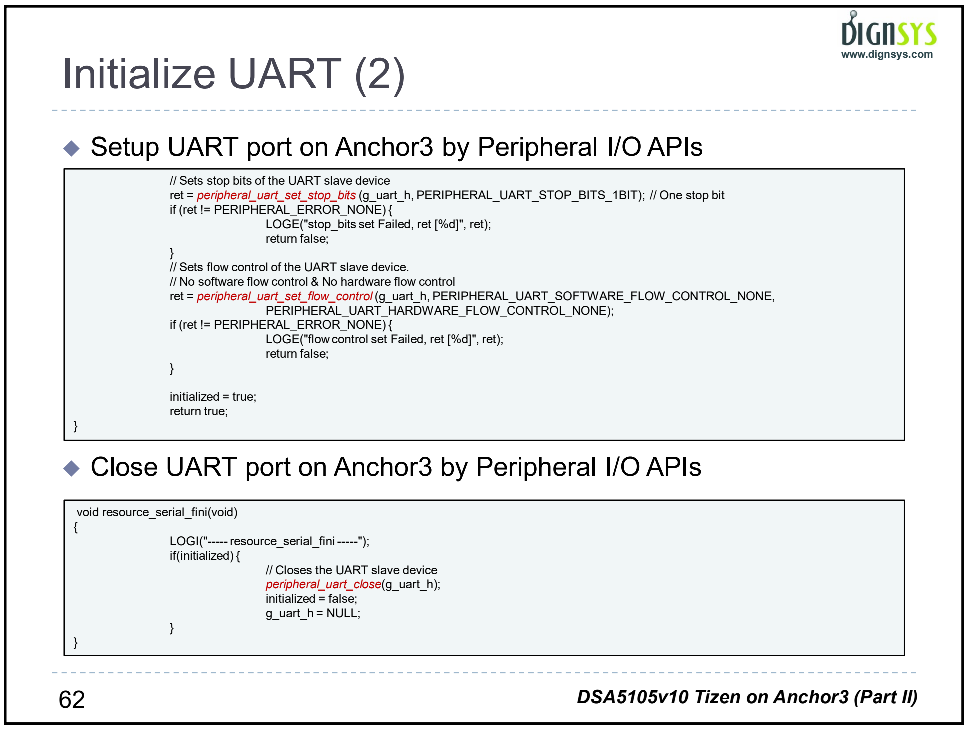
Initialize UART (1)

◆ Setup UART port on Anchor3 by Peripheral I/O APIs

```
bool resource_serial_init(void)
{
        if (initialized) return true;

        LOGI("----- resource_serial_init -----");
        peripheral_error_e ret = PERIPHERAL_ERROR_NONE;

        // Opens the UART slave device
        ret = peripheral_uart_open(UART_PORT_ANCHOR3, &g_uart_h);
        if (ret != PERIPHERAL_ERROR_NONE) {
                LOGE("UART port [%d] open Failed, ret [%d]", UART_PORT_ANCHOR3, ret);
                return false;
        }
        // Sets baud rate of the UART slave device.
        ret = peripheral_uart_set_baud_rate(g_uart_h, PERIPHERAL_UART_BAUD_RATE_9600);  // 9600 bps
        if (ret != PERIPHERAL_ERROR_NONE) {
                LOGE("uart_set_baud_rate set Failed, ret [%d]", ret);
                return false;
        }
        // Sets byte size of the UART slave device.
        ret = peripheral_uart_set_byte_size(g_uart_h, PERIPHERAL_UART_BYTE_SIZE_8BIT);  // 8 data bits
        if (ret != PERIPHERAL_ERROR_NONE) {
                LOGE("byte_size set Failed, ret [%d]", ret);
                return false;
        }
        // Sets parity bit of the UART slave device.
        ret = peripheral_uart_set_parity(g_uart_h, PERIPHERAL_UART_PARITY_NONE);          // No parity is used
        if (ret != PERIPHERAL_ERROR_NONE) {
                LOGE("parity set Failed, ret [%d]", ret);
                return false;
        }
```

Continue .....

61 ***DSA5105v10 Tizen on Anchor3 (Part II)***



Initialize UART (2)

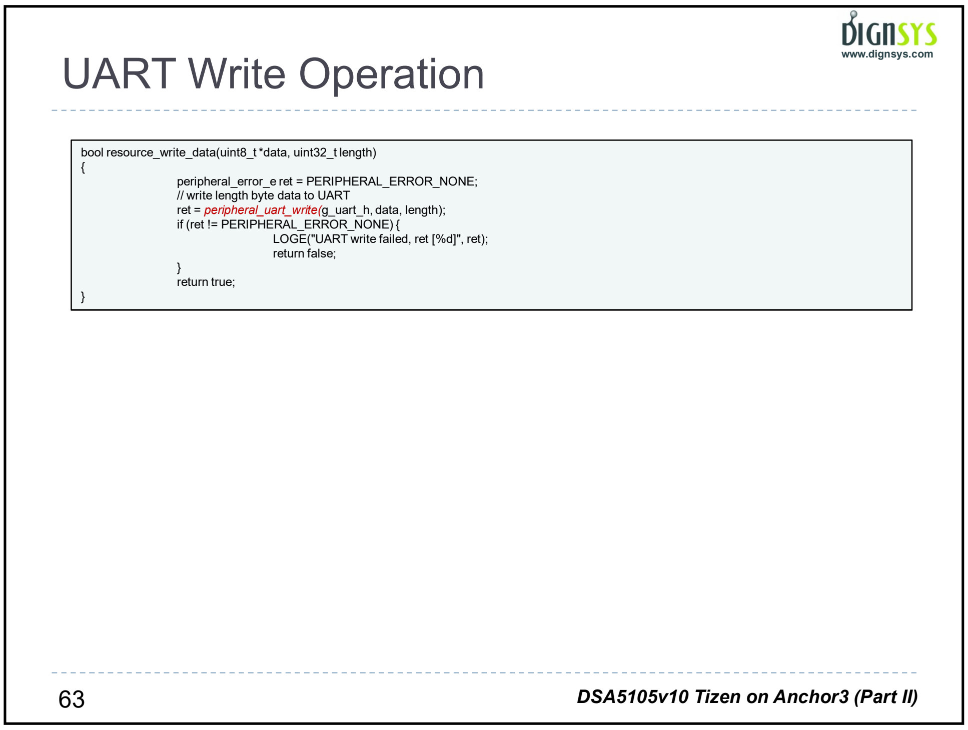
◆ Setup UART port on Anchor3 by Peripheral I/O APIs

```
        // Sets stop bits of the UART slave device
        ret = peripheral_uart_set_stop_bits (g_uart_h, PERIPHERAL_UART_STOP_BITS_1BIT); // One stop bit
        if (ret != PERIPHERAL_ERROR_NONE) {
                LOGE("stop_bits set Failed, ret [%d]", ret);
                return false;
        }
        // Sets flow control of the UART slave device.
        // No software flow control & No hardware flow control
        ret = peripheral_uart_set_flow_control (g_uart_h, PERIPHERAL_UART_SOFTWARE_FLOW_CONTROL_NONE,
                PERIPHERAL_UART_HARDWARE_FLOW_CONTROL_NONE);
        if (ret != PERIPHERAL_ERROR_NONE) {
                LOGE("flow control set Failed, ret [%d]", ret);
                return false;
        }

        initialized = true;
        return true;
}
```

◆ Close UART port on Anchor3 by Peripheral I/O APIs

```
void resource_serial_fini(void)
{
        LOGI("----- resource_serial_fini -----");
        if(initialized) {
                // Closes the UART slave device
                peripheral_uart_close(g_uart_h);
                initialized = false;
                g_uart_h = NULL;
        }
}
```

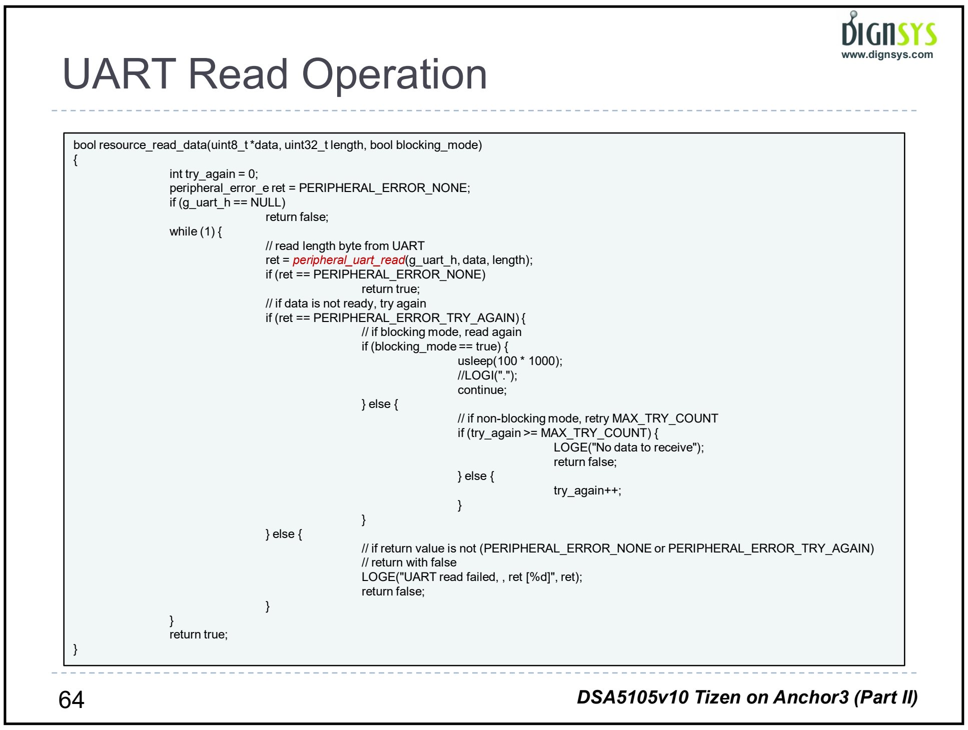
62 ***DSA5105v10 Tizen on Anchor3 (Part II)***



UART Write Operation

```
bool resource_write_data(uint8_t *data, uint32_t length)
{
        peripheral_error_e ret = PERIPHERAL_ERROR_NONE;
        // write length byte data to UART
        ret = peripheral_uart_write(g_uart_h, data, length);
        if (ret != PERIPHERAL_ERROR_NONE) {
                LOGE("UART write failed, ret [%d]", ret);
                return false;
        }
        return true;
}
```

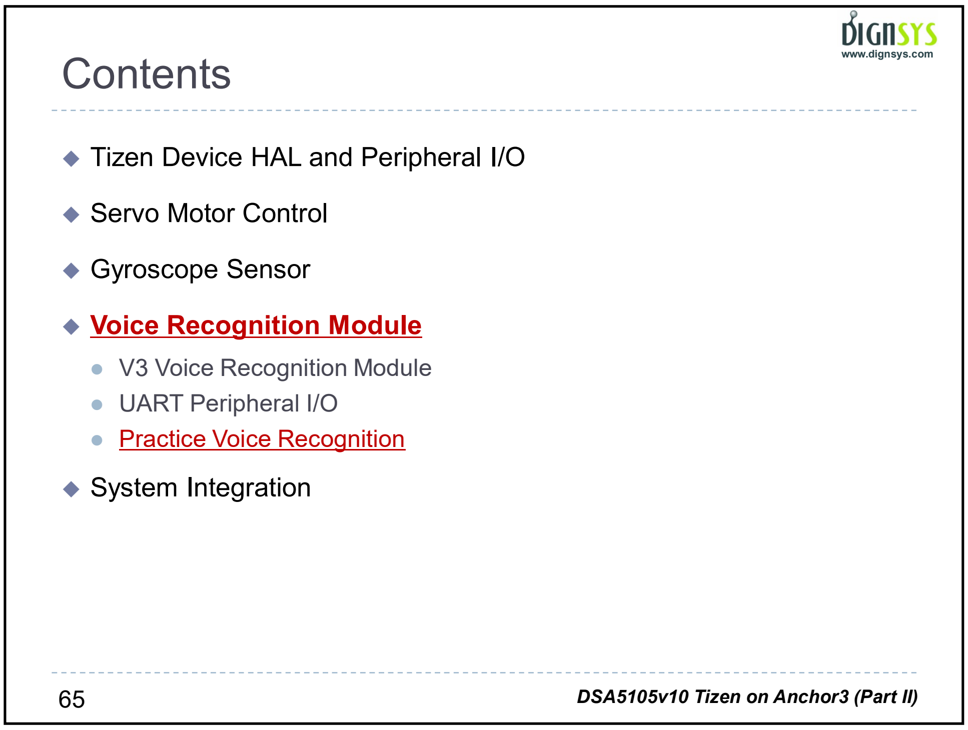
63 *DSA5105v10 Tizen on Anchor3 (Part II)*



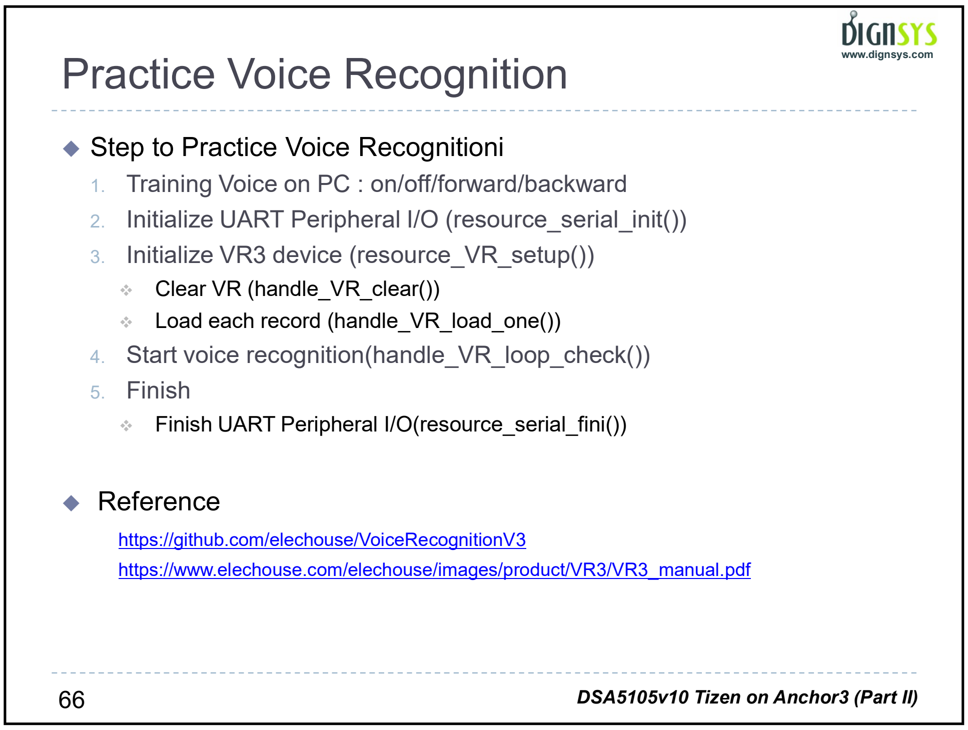
UART Read Operation

```
bool resource_read_data(uint8_t *data, uint32_t length, bool blocking_mode)
{
        int try_again = 0;
        peripheral_error_e ret = PERIPHERAL_ERROR_NONE;
        if (g_uart_h == NULL)
                return false;
        while (1) {
                // read length byte from UART
                ret = peripheral_uart_read(g_uart_h, data, length);
                if (ret == PERIPHERAL_ERROR_NONE)
                        return true;
                // if data is not ready, try again
                if (ret == PERIPHERAL_ERROR_TRY_AGAIN) {
                        // if blocking mode, read again
                        if (blocking_mode == true) {
                                usleep(100 * 1000);
                                //LOGI(".");
                                continue;
                        } else {
                                // if non-blocking mode, retry MAX_TRY_COUNT
                                if (try_again >= MAX_TRY_COUNT) {
                                        LOGE("No data to receive");
                                        return false;
                                } else {
                                        try_again++;
                                }
                        }
                } else {
                        // if return value is not (PERIPHERAL_ERROR_NONE or PERIPHERAL_ERROR_TRY_AGAIN)
                        // return with false
                        LOGE("UART read failed, , ret [%d]", ret);
                        return false;
                }
        }
        return true;
}
```

64 *DSA5105v10 Tizen on Anchor3 (Part II)*



Contents

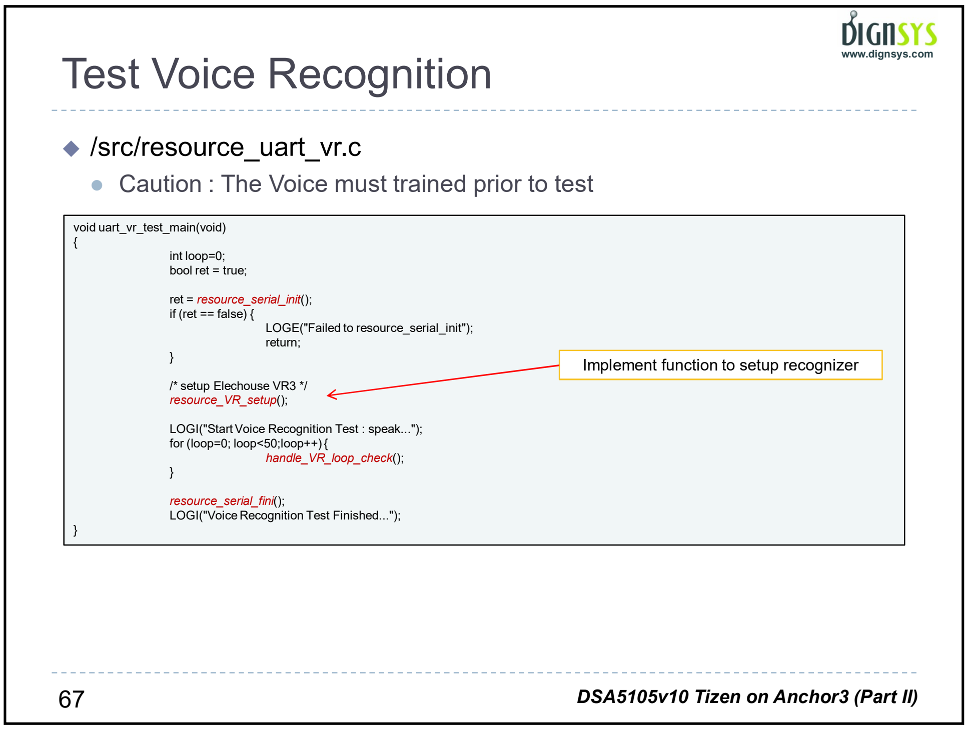


Practice Voice Recognition

- Step to Practice Voice Recognitioni
  1. Training Voice on PC : on/off/forward/backward
  2. Initialize UART Peripheral I/O (resource_serial_init())
  3. Initialize VR3 device (resource_VR_setup())
     - Clear VR (handle_VR_clear())
     - Load each record (handle_VR_load_one())
  4. Start voice recognition(handle_VR_loop_check())
  5. Finish
     - Finish UART Peripheral I/O(resource_serial_fini())

- Reference

  https://github.com/elechouse/VoiceRecognitionV3
  https://www.elechouse.com/elechouse/images/product/VR3/VR3_manual.pdf

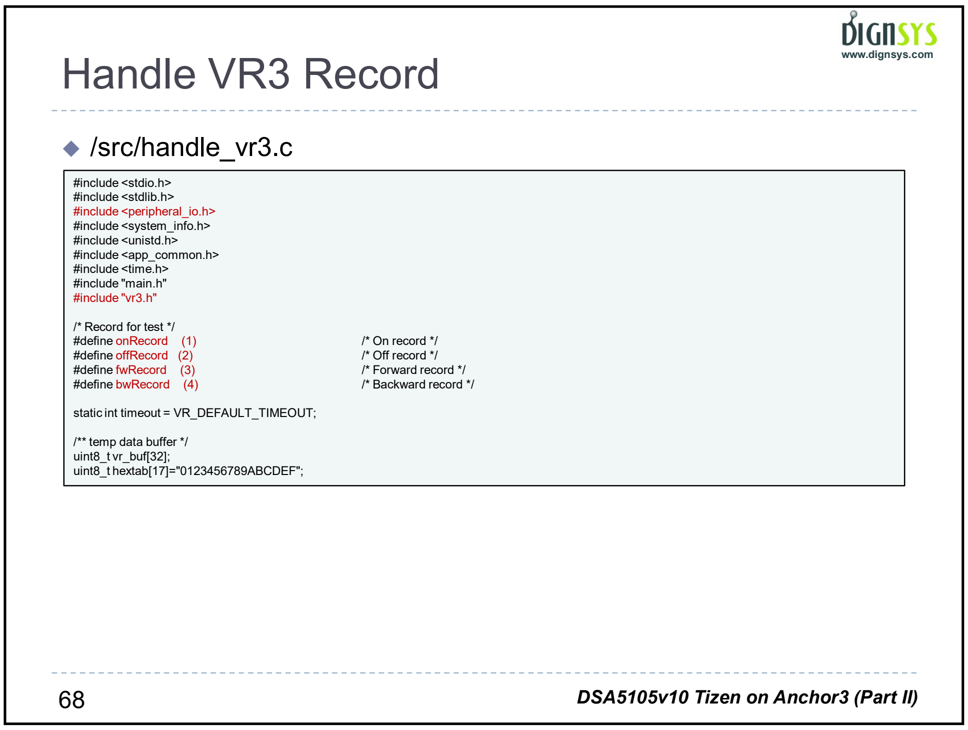


Test Voice Recognition

- /src/resource_uart_vr.c
  - Caution : The Voice must trained prior to test

```
void uart_vr_test_main(void)
{
            int loop=0;
            bool ret = true;

            ret = resource_serial_init();
            if (ret == false) {
                        LOGE("Failed to resource_serial_init");
                        return;
            }

            /* setup Elechouse VR3 */
            resource_VR_setup();

            LOGI("Start Voice Recognition Test : speak...");
            for (loop=0; loop<50;loop++) {
                        handle_VR_loop_check();
            }

            resource_serial_fini();
            LOGI("Voice Recognition Test Finished...");
}
```

67 DSA5105v10 Tizen on Anchor3 (Part II)

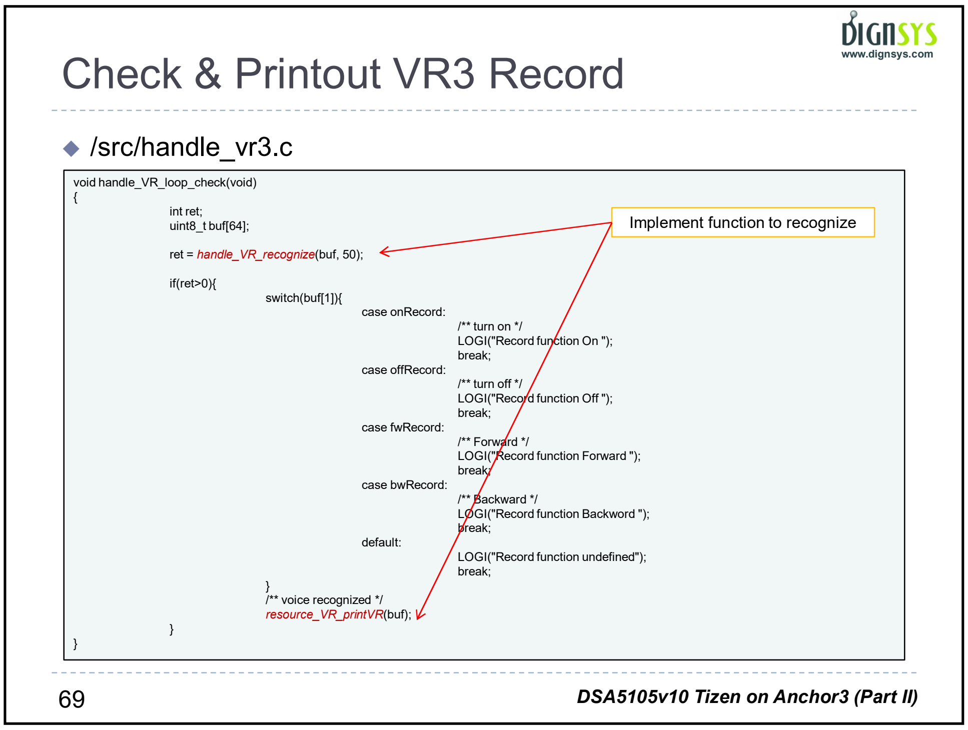


Handle VR3 Record

- /src/handle_vr3.c

```
#include <stdio.h>
#include <stdlib.h>
#include <peripheral_io.h>
#include <system_info.h>
#include <unistd.h>
#include <app_common.h>
#include <time.h>
#include "main.h"
#include "vr3.h"

/* Record for test */
#define onRecord    (1)                /* On record */
#define offRecord   (2)                /* Off record */
#define fwRecord    (3)                /* Forward record */
#define bwRecord    (4)                /* Backward record */

static int timeout = VR_DEFAULT_TIMEOUT;

/** temp data buffer */
uint8_t vr_buf[32];
uint8_t hextab[17]="0123456789ABCDEF";
```

68 DSA5105v10 Tizen on Anchor3 (Part II)



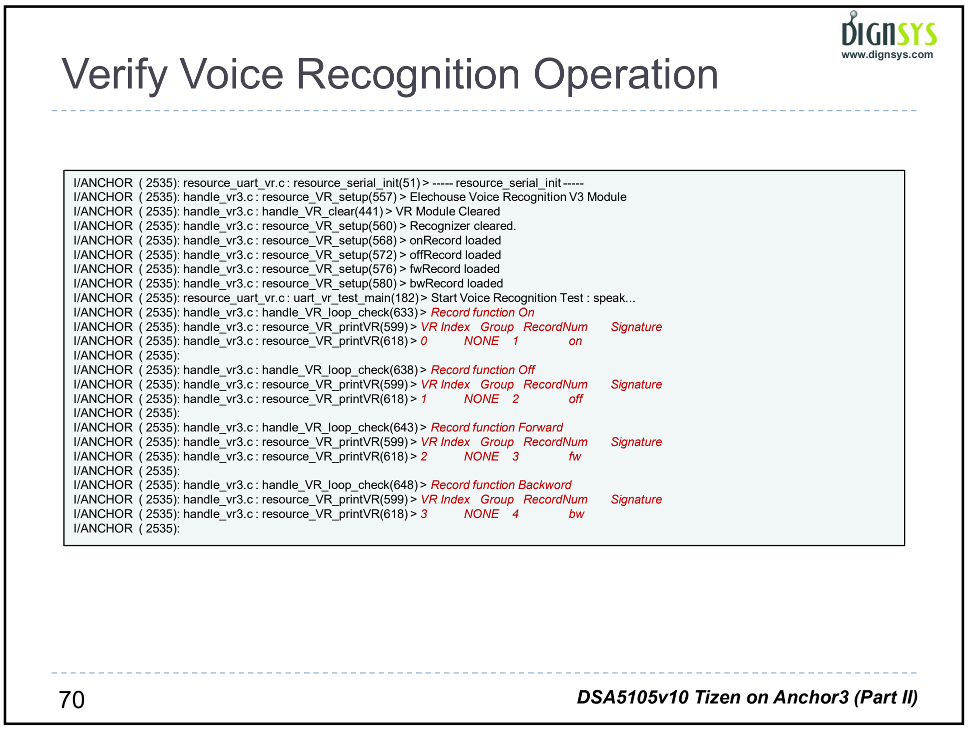
Check & Printout VR3 Record

◆ /src/handle_vr3.c

```
void handle_VR_loop_check(void)
{
        int ret;
        uint8_t buf[64];

        ret = handle_VR_recognize(buf, 50);

        if(ret>0){
                switch(buf[1]){
                        case onRecord:
                                /** turn on */
                                LOGI("Record function On ");
                                break;
                        case offRecord:
                                /** turn off */
                                LOGI("Record function Off ");
                                break;
                        case fwRecord:
                                /** Forward */
                                LOGI("Record function Forward ");
                                break;
                        case bwRecord:
                                /** Backward */
                                LOGI("Record function Backword ");
                                break;
                        default:
                                LOGI("Record function undefined");
                                break;
                }
                /** voice recognized */
                resource_VR_printVR(buf);
        }
}
```

Implement function to recognize

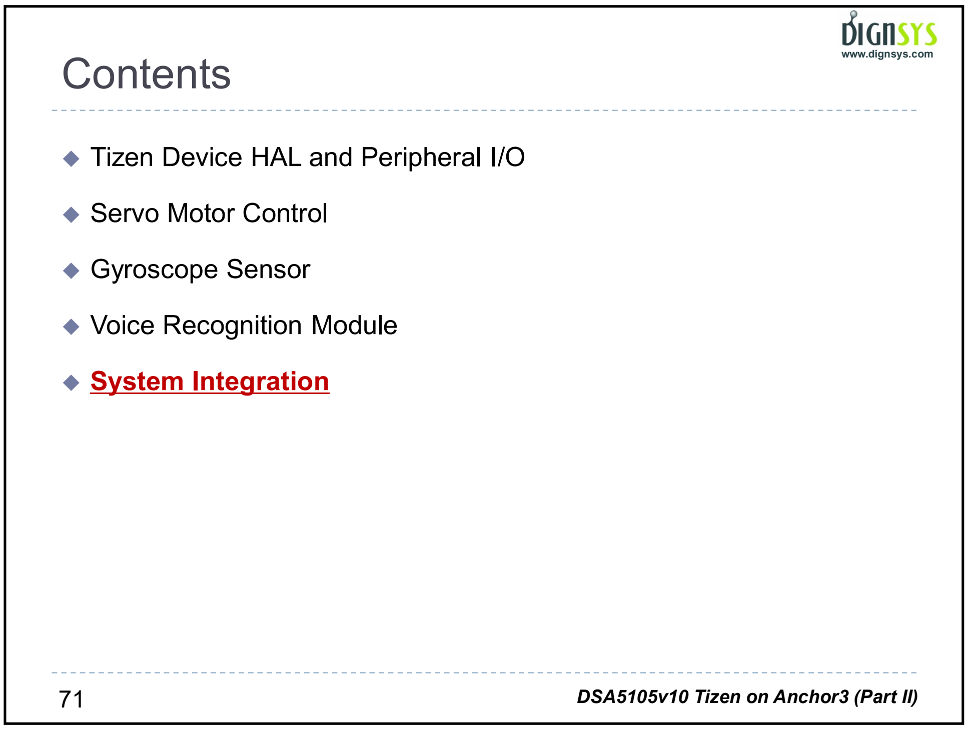
69 DSA5105v10 Tizen on Anchor3 (Part II)



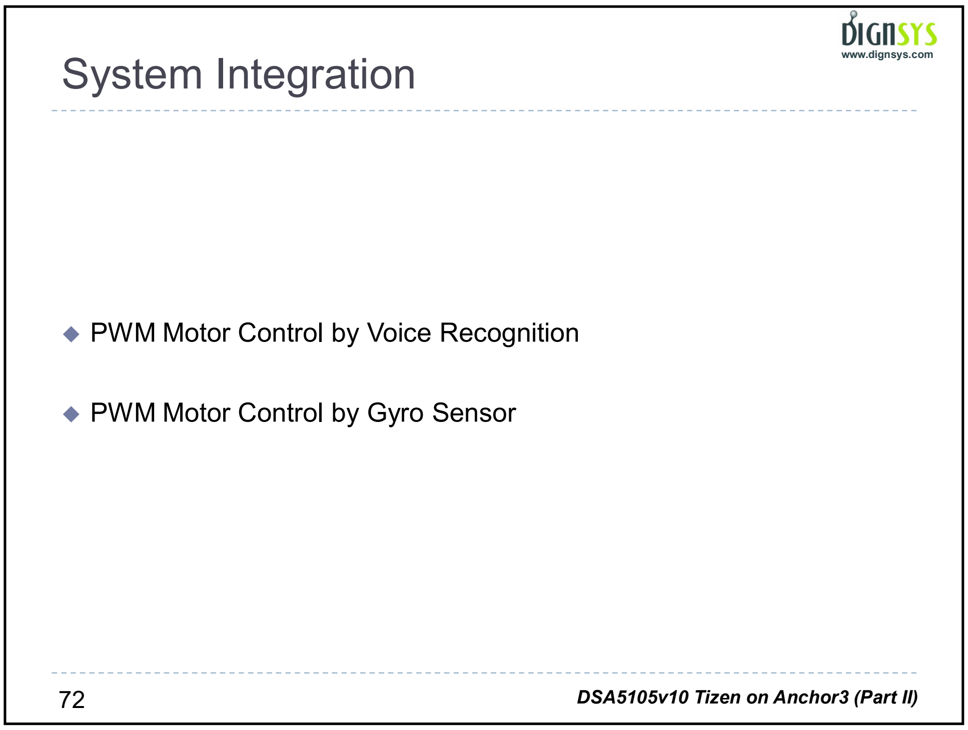
Verify Voice Recognition Operation

```
I/ANCHOR ( 2535): resource_uart_vr.c : resource_serial_init(51) > ----- resource_serial_init -----
I/ANCHOR ( 2535): handle_vr3.c : resource_VR_setup(557) > Elechouse Voice Recognition V3 Module
I/ANCHOR ( 2535): handle_vr3.c : handle_VR_clear(441) > VR Module Cleared
I/ANCHOR ( 2535): handle_vr3.c : resource_VR_setup(560) > Recognizer cleared.
I/ANCHOR ( 2535): handle_vr3.c : resource_VR_setup(568) > onRecord loaded
I/ANCHOR ( 2535): handle_vr3.c : resource_VR_setup(572) > offRecord loaded
I/ANCHOR ( 2535): handle_vr3.c : resource_VR_setup(576) > fwRecord loaded
I/ANCHOR ( 2535): handle_vr3.c : resource_VR_setup(580) > bwRecord loaded
I/ANCHOR ( 2535): resource_uart_vr.c : uart_vr_test_main(182) > Start Voice Recognition Test : speak...
I/ANCHOR ( 2535): handle_vr3.c : handle_VR_loop_check(633) > Record function On
I/ANCHOR ( 2535): handle_vr3.c : resource_VR_printVR(599) > VR Index  Group  RecordNum    Signature
I/ANCHOR ( 2535): handle_vr3.c : resource_VR_printVR(618) > 0         NONE   1            on
I/ANCHOR ( 2535):
I/ANCHOR ( 2535): handle_vr3.c : handle_VR_loop_check(638) > Record function Off
I/ANCHOR ( 2535): handle_vr3.c : resource_VR_printVR(599) > VR Index  Group  RecordNum    Signature
I/ANCHOR ( 2535): handle_vr3.c : resource_VR_printVR(618) > 1         NONE   2            off
I/ANCHOR ( 2535):
I/ANCHOR ( 2535): handle_vr3.c : handle_VR_loop_check(643) > Record function Forward
I/ANCHOR ( 2535): handle_vr3.c : resource_VR_printVR(599) > VR Index  Group  RecordNum    Signature
I/ANCHOR ( 2535): handle_vr3.c : resource_VR_printVR(618) > 2         NONE   3            fw
I/ANCHOR ( 2535):
I/ANCHOR ( 2535): handle_vr3.c : handle_VR_loop_check(648) > Record function Backword
I/ANCHOR ( 2535): handle_vr3.c : resource_VR_printVR(599) > VR Index  Group  RecordNum    Signature
I/ANCHOR ( 2535): handle_vr3.c : resource_VR_printVR(618) > 3         NONE   4            bw
I/ANCHOR ( 2535):
```

70 DSA5105v10 Tizen on Anchor3 (Part II)



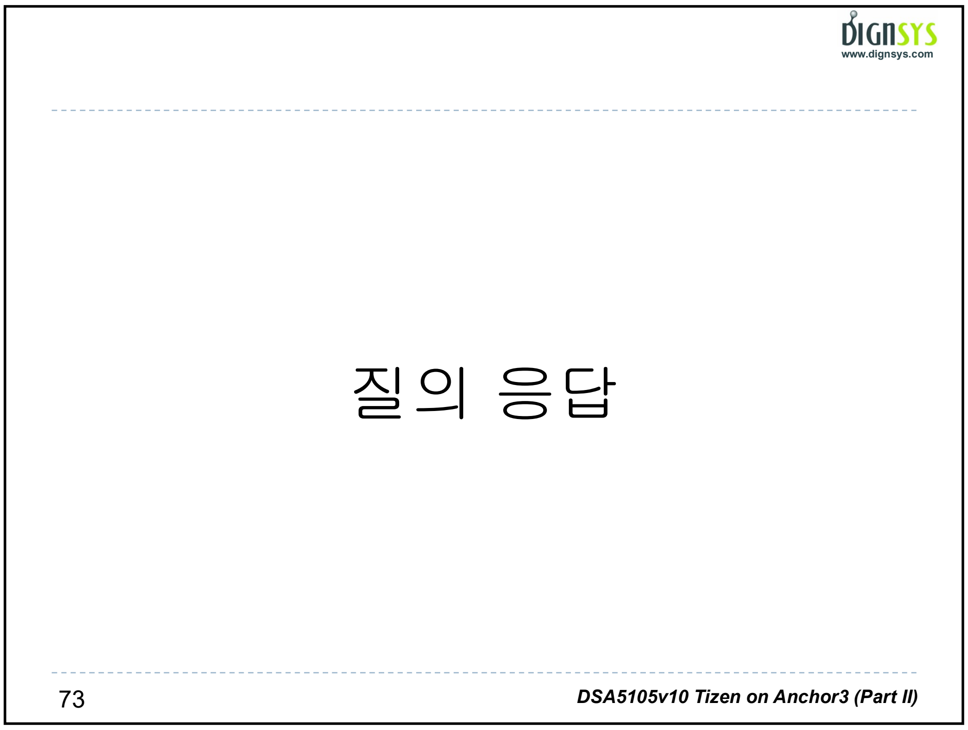
Contents



System Integration

◆ PWM Motor Control by Voice Recognition

◆ PWM Motor Control by Gyro Sensor



www.dignsys.com

질의 응답

73

DSA5105v10 Tizen on Anchor3 (Part II)