This application note provides details on using the following Radio and Module parameters on the Talaria TWO module.

1. RF antenna gain: The RF antenna gain is the default power at which an antenna can transmit or receive.
2. Regulatory domain: The domain regulatory body which regulates the usage of radio frequencies in a particular geographical region.
3. TX power: Tx power is a measure of transmitted signal strength.

# Radio and Module Parameters

## Regulatory Domain

A regulatory domain can be described as a set of rules and policies providing the end user with configurations of country code, calibration channel, and output power settings for a wireless device set up in a specific area.

The following boot argument allows the user to change the Regulatory domain:

**Boot Argument:**

|  |
| --- |
| reg\_domain=< reg domain needs to be changed> |

The following are the Regulatory domains supported in Talaria TWO with this application:

1. ETSI
2. FCC
3. KCC
4. SRRC
5. TELEC

## RF Antenna Gain

Antenna gain is the ability of the antenna to radiate in any direction compared to an isotropic antenna.

The following boot argument allows the user to set antenna gain in dBi:

**Boot Argument:**

|  |
| --- |
| rf.antenna\_gain |

The antenna gain value is used in calculation of output power to comply with regulatory domain settings.

**Note:** User can configure the antenna gain for INP1011, INP1012 and INP1015 modules.

The following antenna gain look-up table provides reference gain information for the E.I.R.P. power measurements.

|  |  |
| --- | --- |
| **Module Type** | **Antenna Gain (dBi)** |
| INP1010 | 1 |
| INP1011 | 2.15 |
| INP1012 | 2.15 |
| INP1013 | 1 |
| INP1014 | 1 |
| INP1015 | 2.15 |

Table : Antenna gain look-up table

## Tx Power

The Tx power setting specifies the strength of the signal which the station produces during transmission. Lowering the Tx power allows the user to reduce interferences when more Wi-Fi devices are in the vicinity.

The following boot argument allows the user to set maximum Tx power in dBm:

**Boot Argument:**

|  |
| --- |
| tx\_power=<MAX TX power in dBm> |

## Device Information

The device information available in the boot sector of the Talaria TWO module contains the module type which is stored in the flash memory in the factory.

# APIs used in the Application

## Radio and Module Parameter APIs

1. wcm\_set\_channel\_spec()

Sets channels and Tx power according to channel specification.

1. wcm\_get\_txpower()

Gets the Tx power value configured using wcm\_set\_txpower for the Wi-Fi interface.

1. wcm\_set\_txpower()

Sets maximum Tx power for the Wi-Fi interface.

1. os\_devinfo\_module\_type()

Gets module type for the Talaria TWO module.

# Source Code Walkthrough

## Source Code Walkthrough

create\_wcm\_hndl() function creates the WCM handle and applies the provided domain.

|  |
| --- |
| hndl = wcm\_create(NULL);  if(hndl == NULL) {  os\_printf(“wcm create failed.\n”);  return NULL;  }  if(domain != 0) {  /\* reg domain info given \*/  os\_printf(“Applying reg domain: %s\n”, domain);  if(wcm\_set\_channel\_spec(hndl, domain) != 0) {  os\_printf(“Applying reg domain failed.!\n”);  return NULL;  }  } |

get\_devicemodule\_type() function reads the device information from the boot sector of the Talaria TWO module.

|  |
| --- |
| os\_printf(“\r \n Reading module type\n”);  os\_devinfo\_module\_type(&type);  os\_printf(“\r \n Module type = INP%u \n”, type); |

wcmif\_txpowerset() API sets the maximum Tx power for the Wi-Fi interface.

|  |
| --- |
| wcmif\_txpowerset(txpower);  os\_printf(“txpower=%s\n”, txpower); |

wcm\_get\_txpower() API gets the maximum Tx power for the Wi-Fi interface.

|  |
| --- |
| tx\_pow = wcm\_get\_txpower(hndl);  os\_printf(“\r\n Tx power in dBm = %d\n”, (int)tx\_pow); |

wifi\_connect\_to\_network() creates a Wi-Fi network interface to connect to a network.

conn\_status checks if the Wi-Fi is in a connected or disconnected state. Returns 0 on success or a negative error code in case of an error.

|  |
| --- |
| rval = wifi\_connect\_to\_network(&hndl, WCM\_CONN\_WAIT\_INFINITE, &wcm\_connected);  if(rval < 0) {  os\_printf("\r\nError: Unable to connect to network\n");  return 0;  }  if(wcm\_connected != true) {  os\_printf("\r\nCouldn't Connect to network");  wcm\_disconnect(hndl);  return -1;  } |

# Building the Application

To build the sample application, execute the following commands from the FreeRTOS SDK directory:

|  |
| --- |
| cd examples/  make clean | make |

The make command should generate rf\_param.elf in the out directory.

# Running the Application

## Programming Talaria TWO using the Download Tool

Program rf\_param.elf *(freertos\_sdk\_x.y\examples\radio\_module\_params\bin)* using the Download tool:

1. Launch the Download tool provided with InnoPhase Talaria TWO FreeRTOS SDK.
2. In the GUI window:
   1. Boot Target: Select the appropriate EVK from the drop-down.
   2. ELF Input: Load the rf\_param.elf by clicking on Select ELF File.
   3. Boot Arguments: Pass the following boot arguments to set the REG domain and TX power.

|  |
| --- |
| reg\_domain=<Reg domain>,tx\_power=<MAX TX power in dBm> |

* 1. Programming: Click on PROG Flash.

## Expected Output

On flashing the application using the Download Tool, the console output is as follows:

|  |
| --- |
| UART:SRWWWWAE4 DWT comparators, range 0x8000  Build $Id: git-ef87896f9 $  flash: Gordon ready!  Y-BOOT 208ef13 2019-07-22 12:26:54 -0500 790da1-b-7  ROM yoda-h0-rom-16-0-gd5a8e586  FLASH:PWWWWWWAE  Build $Id: git-bbd63ca $  Flash detected. flash.hw.uuid: 39483937-3207-0063-009c-ffffffffffff  Bootargs: reg\_domain=SRRC tx\_power=11 ssid=test passphrase=12345678  $App:git-c132c26  SDK Ver: FREERTOS\_SDK\_1.0  Radio and Module Parameters Demo App  reg\_domain = SRRC  addr e0:69:3a:00:16:1a  Applying reg domain  Reg Domain Applied  Maximum TX power set = 11  Maximum TX power get = 11  network profile created for ssid: test  Connecting to added network : test  [0.882,007] CONNECT:9a:96:21:2e:dc:32 Channel:11 rssi:-40 dBm  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_LINK\_UP  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_ADDRESS  [0.973,939] MYIP 192.168.70.179  [0.974,104] IPv6 [fe80::e269:3aff:fe00:161a]-link  wcm\_notify\_cb to App Layer - WCM\_NOTIFY\_MSG\_CONNECTED  Connected to added network : test  Reading module type  Module type = INP1010  Connected to < test > network |

**Console output**:

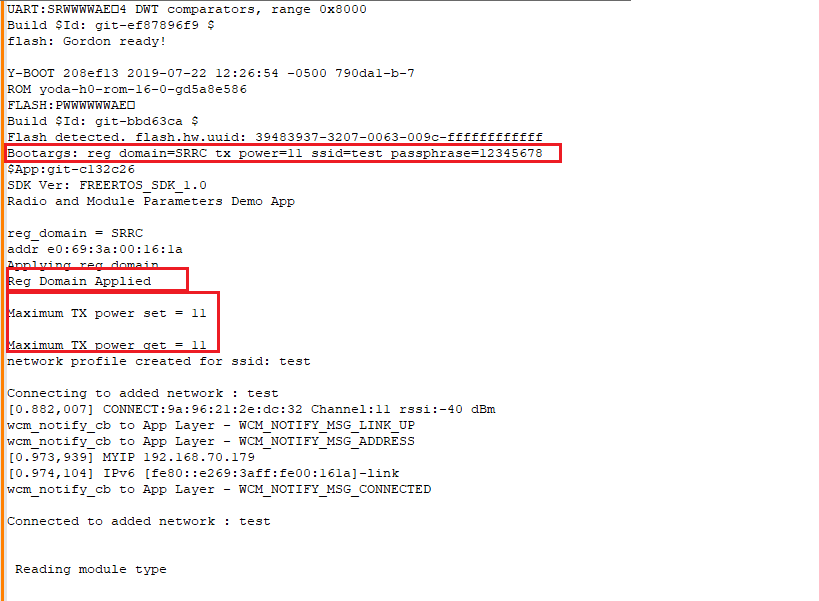


Figure : Console output