## Station Mode

### Description

This use case is to demonstrate station mode configuration and connection to an Access Point configured with personal security (WPA/WPA2 personal). And disconnecting to network from the connected network.

**Note**: Skip the passphrase if it an open AP.

### Prerequisites

Access Point configured with WPA/WPA2 personal security.

### AT Command Sequence

|  |  |
| --- | --- |
| **Command** | **Description** |
| *at* | Check communication state |
| *at+wscan* | Get list of APs available from the vicinity |
| *at+wcon* | Connect to a desired AP from the received scan results |
| *at+wstatus* | Get IP address of Talaria TWO to verify the connection is successful |
| *at+wdis* | To disconnect from the connected network |

Table 1: Station Mode - AT Commands

**Note**: at+wdis responds with ERROR response only when network state is disconnected.

### Procedure

Execute the following commands on Talaria TWO:

|  |
| --- |
| at  at+wscan  at+wcon=InnoPhase,Inno@9070  at+wstatus=0  at+wdis  at+wstatus=0 |

### Serial Log

Text

Description automatically generated

Figure 1: Station Mode – Serial log

## Power Management

Power save methods allow battery-operated devices to save power by shutting down their wireless radios. While devices are in a sleep state, the AP buffers frames destined for it.

IEEE 802.11 standard allows a wireless device to reduce its power consumption, to turn off its radios and to wake up at the correct time to retrieve its traffic. The purpose of power saving feature is to increase the battery life and to allow longer performance.

Basic Terminology in 802.11 Power Save:

1. AP will store data in buffer when associated Station is in sleep mode.
2. TBTT (Target Beacon Transmission Time): Time in which AP should send the Beacon.
3. Beacon Interval: Difference between two TBTTs is called Beacon Interval. Beacon interval is represented in Time Unit (TU). Each TU is 1024 microsecond. If Beacon interval is 100TU, it means 102.4ms.

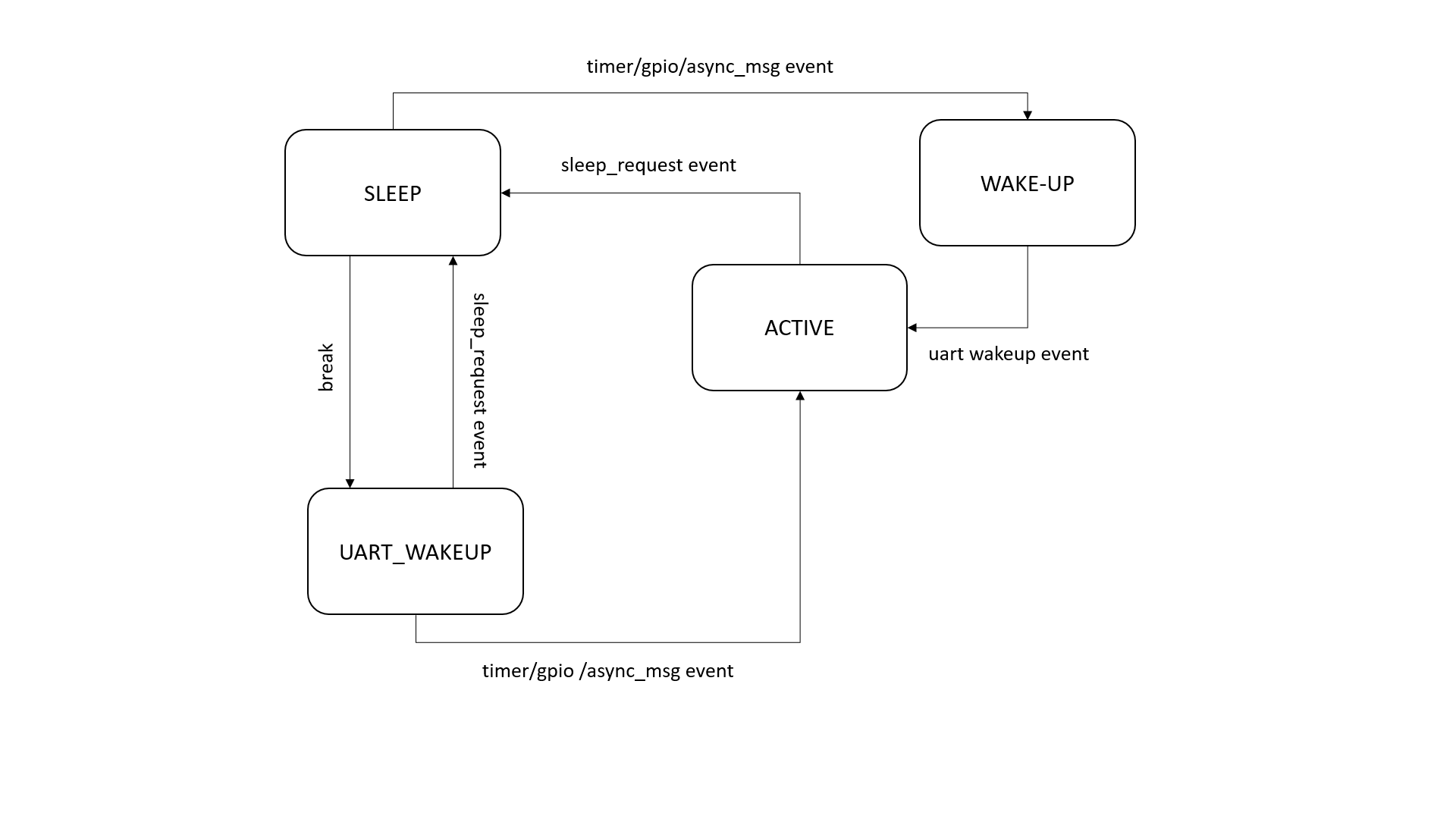


Figure 2: Sleep state machine

### Hardware Power Save Mode

#### Suspend/Deep Sleep Mode

**Description**

This use case is suitable for the system where power consumption is the major concern. Suspend mode demonstrates the system power save configuration. If this mode is enabled, the system will go into suspend mode or deep sleep mode when there is nothing to do for the processor. When in suspend state, the system will still wakeup if an interrupt occurs, but the latency will be much larger compared to running with suspend mode disabled.

**Prerequisites**

An Access Point configured with any Wi-Fi security types (WPA/WPA2/WPA3 Personal/Enterprise protocols).

**AT Command Sequence**

|  |  |
| --- | --- |
| **AT Command** | **Description** |
| *at* | Check the communication state |
| *at+wcon* | Connect to a desired AP |
| *at+wstatus* | Get WLAN status |
| *at+syssleep=<suspend time>* | Puts the module into suspend/deep sleep mode for the configured “suspend time” value.  Unit is in milliseconds.  0 - Indicates infinite sleep mode  Range - [0- 32-bit value] |

Table 2: Suspend/Deep sleep Mode - AT Commands

**Procedure**

Issue the following commands to connect Talaria TWO module to an AP of SSID "InnoPhase" and passphrase "Inno@1020". Check the WLAN status with at+wstatus command. Put the module into deep sleep mode for about 10 sec (10000ms). Once the module timeouts (10 sec), it will respond with "WAKEUP" message and switch to active mode.

|  |
| --- |
| at  at+wcon=InnoPhase,Inno@1020  at+wstatus=0  at+syssleep=10000 |

**Serial Log**

Text

Description automatically generated

Figure 3: Suspend/Deep sleep Mode - Serial log

**OTTI Log**

A screenshot of a computer

Description automatically generated with medium confidence

Figure 4: Suspend/Deep sleep Mode - OTTI log

### IEEE Power Save Mode

#### Always Active Mode

**Description**

This use case demonstrates configuring Talaria TWO module to put the radio in an always active mode. This use case is **not** suitable for a low power solution. When this mode (traffic timeout set to 0) is configured, rest of the power save configurations (listen interval etc.) will have no effect.

**Prerequisites**

Access Point configured with any Wi-Fi security types (WPA/WPA2/WPA3 Personal/Enterprise protocols).

**AT Command Sequence**

|  |  |
| --- | --- |
| **AT Command** | **Description** |
| *at* | Check the communication state |
| *at+wpmcfg=<listen interval>, <traffic timeout>, <pspoll>,*  *<dyn\_listen\_int>* | 1. listen Interval: Beacon listen interval [0 - integer range] 2. traffic timeout: This specifies the amount of time (in milliseconds) that the device should stay awake with the radio powered-up after a transmission (to quickly receive any replies that may be the result of the transmission).   **Range**: 0/ integer range. When set to 0, device will stay awake forever.  **Default value** is set to 12ms.   1. pspoll: Send “ps poll”.   **Range**: 0-disable/1-enable. When set to 1, device will send “ps poll” if a beacon is missed. This will increase power consumption slightly but decrease latency.  **Default value** is set to 0.   1. dyn\_listen\_in - Dynamic listen interval.   **Range**: 0-disable/1-enable. When set to 1, device will listen to each beacon if there has been traffic recently.  **Default value** is set to 0. |
| *at+wcon* | Connect to a desired AP |
| *at+wstatus* | Get WLAN status |

Table 3: Always Active Mode - AT Commands

**Procedure**

Issue the following commands to configure the module to be in always active mode. Connect the Talaria TWO module to an AP of SSID "InnoPhase" and passphrase "Inno@1020". Check the WLAN status with at+wstatus command.

|  |
| --- |
| at  at+wpmcfg=3,0,0,0  at+wcon=InnoPhase,Inno@1020  at+wstatus=0 |

**Serial Log**

Text

Description automatically generated

Figure 5: Always Active Mode - Serial log

**OTTI Log**

A picture containing graphical user interface

Description automatically generated

Figure 6: Always Active Mode - OTTI log

#### Listen Interval Based Wakeup

Listen Interval specifies how often the device will wake-up and listen for beacons. On Talaria TWO, the listen interval is set to 0 by default and it will wake-up to listen to every DTIM beacon.

**Description**

This use case demonstrates wake up based on user defined Listen-Interval configuration (which is 10 beacons in this use case). In Listen Interval based Wakeup method, Station will miss the broad cast data.

The listen interval is always rounded up to the nearest even multiple of the DTIM interval.

For example: By setting the listen interval to 10, Talaria TWO will listen to each 12th beacon if the AP uses DTIM=3.

**Prerequisites**

An Access Point configured with any Wi-Fi security types (WPA/WPA2/WPA3 Personal/Enterprise protocols).

**AT Command Sequence**

|  |  |
| --- | --- |
| **AT Command** | **Description** |
| at | Check the communication state |
| at+wpmcfg=<listen interval>, <traffic timeout>, <pspoll>,  <dyn\_listen\_int> | 1. listen Interval: Beacon listen interval [0 - integer range] 2. traffic timeout: This specifies the amount of time (in milliseconds) that the device should stay awake with the radio powered-up after a transmission (to quickly receive any replies that may be the result of the transmission).   **Range**: 0/ integer range. When set to 0, device will stay awake forever.  **Default value** is set to 12ms.   1. pspoll: Send “ps poll”.   **Range**: 0-disable/1-enable. When set to 1, device will send “ps poll” if a beacon is missed. This will increase power consumption slightly but decrease latency.  **Default value** is set to 0.   1. dyn\_listen\_in - Dynamic listen interval.   **Range**: 0-disable/1-enable. When set to 1, device will listen to each beacon if there has been traffic recently.  **Default value** is set to 0. |
| at+wcon | Connect to a desired AP. |
| at+wstatus | Get WLAN status. |
| at+syssleep=<suspend time> | Puts the module into suspend/deep sleep mode for the configured “suspend time” value.  Unit is in milliseconds.  0 - Indicates infinite sleep mode.  Range - [0- 32-bit value]. |

Table 4: Listen Interval based Wakeup - AT Commands

**Procedure**

Issue following commands to configure the module to listen and wake up for every 10th Beacon. Connect the Talaria TWO module to an AP of SSID "InnoPhase" and passphrase "Inno@1020". Check the WLAN status with at+wstatus command. Put the module into deep sleep mode for about 60 sec (60000ms).

|  |
| --- |
| at  at+wpmcfg=10,12,0,0  at+wcon=InnoPhase,Inno@1020  at+wstatus=0  at+syssleep=60000 |

**Serial Log**

Text

Description automatically generated

Figure 7: Listen Interval based Wakeup - Serial log

**OTTI Log**

Chart

Description automatically generated

Figure 8: Listen Interval based Wakeup - OTTI log

#### Dynamic Listen Interval

Listen to all beacons if there has been traffic recently. This will decrease latency for incoming traffic but will increase power consumption slightly.

**Description**

This use case demonstrates the dynamic listen interval. Module wakes up based on user defined Listen-Interval configuration (which is 10 beacons in this use case). In Listen Interval based wake-up method, station will miss the broad cast data. Upon enabling dynamic listen interval, module wakes up and listens to each Beacon if there has been traffic recently.

**Prerequisites**

An Access Point configured with any Wi-Fi security types (WPA/WPA2/WPA3 Personal/Enterprise protocols).

**AT Command Sequence**

|  |  |
| --- | --- |
| **AT Command** | **Description** |
| at | Check the communication state |
| at+wpmcfg=<listen interval>, <traffic timeout>, <pspoll>,  <dyn\_listen\_int> | 1. listen Interval: Beacon listen interval [0 - integer range] 2. traffic timeout: This specifies the amount of time (in milliseconds) that the device should stay awake with the radio powered-up after a transmission (to quickly receive any replies that may be the result of the transmission).   **Range**: 0/ integer range. When set to 0, device will stay awake forever.  **Default value** is set to 12ms.   1. pspoll: Send “ps poll”.   **Range**: 0-disable/1-enable. When set to 1, device will send “ps poll” if a beacon is missed. This will increase power consumption slightly but decrease latency.  **Default value** is set to 0.   1. dyn\_listen\_in - Dynamic listen interval.   **Range**: 0-disable/1-enable. When set to 1, device will listen to each beacon if there has been traffic recently.  **Default value** is set to 0. |
| at+wcon | Connect to a desired AP. |
| at+wstatus | Get WLAN status. |
| at+syssleep=<suspend time> | Puts the module into suspend/deep sleep mode for the configured “suspend time” value.  Unit is in milliseconds.  0 - Indicates infinite sleep mode.  Range - [0- 32-bit value]. |

Table 5: Dynamic Listen Interval - AT Commands

**Procedure**

Issue the following commands to configure the module to listen and wake up for every 10th Beacon and enable dynamic listen interval to listen to every Beacon if there has been any traffic. Connect the Talaria TWO module to an AP of SSID "InnoPhase" and passphrase "Inno@1020". Check the WLAN status with at+wstatus command. Put the module into deep sleep mode for about 60 sec (60000ms).

|  |
| --- |
| at  at+wpmcfg=10,12,0,1  at+wcon=InnoPhase,Inno@1020  at+wstatus=0  at+syssleep=60000 |

**Serial Log**

Text

Description automatically generated

Figure 9: Dynamic Listen Interval - Serial log

Connect the laptop to the same Access Point “InnoPhase” to which the Talaria TWO is connected.

Graphical user interface, text

Description automatically generated

Figure 10: Connect to InnoPhase AP

Ping from laptop to the Talaria TWO module (192.168.2.115 is the IP address of Talaria TWO) which sends ICMP packets. This generates traffic, and hence Dynamic listen interval is enabled on Talaria TWO.

To generate traffic, execute the ping command from the Windows Command Prompt.

Text

Description automatically generated

Figure 11: Connect to InnoPhase AP Command Prompt

**OTTI Log**

Graphical user interface, chart

Description automatically generated

Figure 12: Dynamic Listen Interval - OTTI log

#### Low Power Optimization

**Description**

This use case demonstrates the power optimization configurations.

**Prerequisites**

An Access Point configured with any Wi-Fi security types (WPA/WPA2/WPA3 Personal/Enterprise protocols.

**AT Command Sequence**

|  |  |
| --- | --- |
| **AT Command** | **Description** |
| *at* | Check communication state |
| *at+wpmcfg=<listen interval>, <traffic timeout>, <pspoll>, <dyn\_listen\_int>[<starx\_nap>, <sta\_only\_bc>, <txps>,*  *<mcast\_dont\_care>, <dtim>]* | 1. listen Interval: Beacon listen interval [0/ integer range] 2. traffic timeout: This specifies the amount of time (in milliseconds) that the device should stay awake with the radio powered-up after a transmission (to quickly receive any replies that may be the result of the transmission) [0/ integer range]. When set to 0, device will stay awake forever. Default value is set to 12ms. 3. pspoll: use PS-poll [0-disable/1-enable]. When set to 1, device will send “ps poll” if a beacon is missed. This will increase power consumption slightly but decrease latency. Default value is set to 0. 4. dyn\_listen\_in - Dynamic listen interval [0-disable/1-enable]. When set to 1, device will listen to each beacon if there has been traffic recently. Default value is set to 0. 5. starx\_nap: STA Receive nap [0-disable/1-enable]. When set to 1, turn off receiver for inappropriate frames for station. Default value is set to 0. 6. sta\_only\_bc: STA broadcast Only [0-disable/1-enable]. When set to 1, do not receive multicast frames that are not applicable. Default value is set to 0. 7. txps: Tx power save [0-disable/1-enable]. When set to 1, send outgoing frames without leaving Wi-Fi power save. Default value is set to 0. 8. mcast\_dont\_care: Multicast don’t care[0-disable/1-enable]. When set to 1, ignore the multicast flag in beacons. Default value is set to 0. 9. dtim: Delivery traffic indication message [0-disable/1-enable] Wakes up only at effective listen interval and does not switch to listen to every beacon in case of beacon miss |
| *at+wcon* | Connect to a desired AP. |
| *at+wstatus* | Get WALN status. |
| *at+syssleep=<suspend time>* | Puts the module into suspend/deep sleep mode for the configured “suspend time” value.  Unit is in milliseconds.  0 - Indicates infinite sleep mode.  Range - [0- 32-bit value]. |

Table 6: Low Power Optimization - AT Commands

**Note**: Arguments in [ ] are optional and those in <> are mandatory.

**Procedure**

Issue the following command to configure Talaria TWO to set the power save configuration as listen interval=10, traffic timeout=12ms, ps-poll=0, dynamic listen interval=0, station rx nap=1, station broadcast only=1, transmit PS=1, multicast don’t care =1, and dtim=1. Connect the Talaria TWO to an AP of SSID "InnoPhase" and passphrase "Inno@1020". Check the WLAN status with at+wstatus command. Put the module into deep sleep mode for about 60 sec (60000ms).

|  |
| --- |
| at  at+wpmcfg=10,12,0,0,1,1,1,1,1  at+wcon=InnoPhase,Inno@1020 at+wstatus=0  at+syssleep=60000 |

**Serial Log**

A screenshot of a computer

Description automatically generated with medium confidence

Figure 13: Low Power Optimization - Serial log

**OTTI Log**

Chart, histogram

Description automatically generated

Figure 14: Low Power Optimization - OTTI log

### Wakeup from Hardware Power Save Mode

#### Upon Suspend Mode Timeout

**Description**

The following usecase demonstrates wakeup from suspend mode upon timeout of the suspend timer.

**Note**: In case of infinite suspend time, it is mandatory to set at+wakeupcfg.

**Prerequisites**

An Access Point configured with any Wi-Fi security types (WPA/WPA2/WPA3 Personal/Enterprise protocols).

**AT Command Sequence**

|  |  |
| --- | --- |
| **AT Command** | **Description** |
| *at* | Check communication state |
| *at+wscan* | Get list of available APs from the vicinity |
| *at+wcon* | Connect to an Access Point |
| *at+wstatus=0* | Get IP address of Talaria TWO to verify if the connection is successful |
| *at+syssleep* | Puts system into suspend mode for the configured time (Time in milliseconds) |

Table 7: Upon suspend mode timeout - AT Commands

**Procedure**

**Step 1**: Issue the following commands to configure the module in sleep mode for 30secs and wakeup based on timeout (30secs in this example).

|  |
| --- |
| at  at+wscan  at+wcon=InnoPhase,Inno@9070  at+wstatus=0  at+syssleep=30000 |

**Serial Log**

Text

Description automatically generated

Figure 15: Upon suspend mode timeout - Serial log

**OTII Log**

Chart

Description automatically generated

Figure 16: Upon suspend mode timeout – OTII log

#### Upon GPIO Interrupt

**Description**

The following usecase demonstrates an interrupt based wakeup mechanism when the module is put in suspend mode.

**Note**:

1. All the GPIOs are pulled HIGH internally by default except GPIO18 (pulled LOW).
2. To generate a low-level interrupt, GPIO must be pulled LOW externally.
3. To generate a high level interrupt, pull down that particular GPIO externally through a pull down register. Only then the high interrupt will get detected.
4. Since there is always a pull down on high level interrupt GPIO, some amount of small current is always sinking through that pin, which will add-on to the power save suspend current.
5. Hence, it is recommended to use low level interrupt for low power use case.

**Prerequisites**

An Access Point configured with any Wi-Fi security types (WPA/WPA2/WPA3 Personal/Enterprise protocols).

**AT Command Sequence**

|  |  |
| --- | --- |
| **AT Command** | **Description** |
| *at* | Check communication state |
| *at+wscan* | Get list of available APs from the vicinity |
| *at+wcon* | Connect to an Access Point |
| *at+wstatus=0* | Get IP address of Talaria TWO to verify if the connection is successful |
| *at+wakeupcfg* | Configure interrupt-based system wakeup |
| *at+syssleep* | Puts system into suspend mode for the configured time (Time in milliseconds) |

Table 8: GPIO Interrupt - AT Commands

**Procedure**

Issue the following commands to configure the module in infinite sleep mode and wakeup based on GPIO14 interrupt. GPIO14 is configured for low wakeup level (which implies, generate interrupt when the GPIO14 is low level).

|  |
| --- |
| at  at+wscan  at+wcon=InnoPhase,Inno@9070  at+wstatus=0  at+wakeupcfg=14,0,0,0  at+syssleep=0 |

**Serial Log**

Text

Description automatically generated

Figure 17: GPIO Interrupt - Serial log

**OTTI Log**

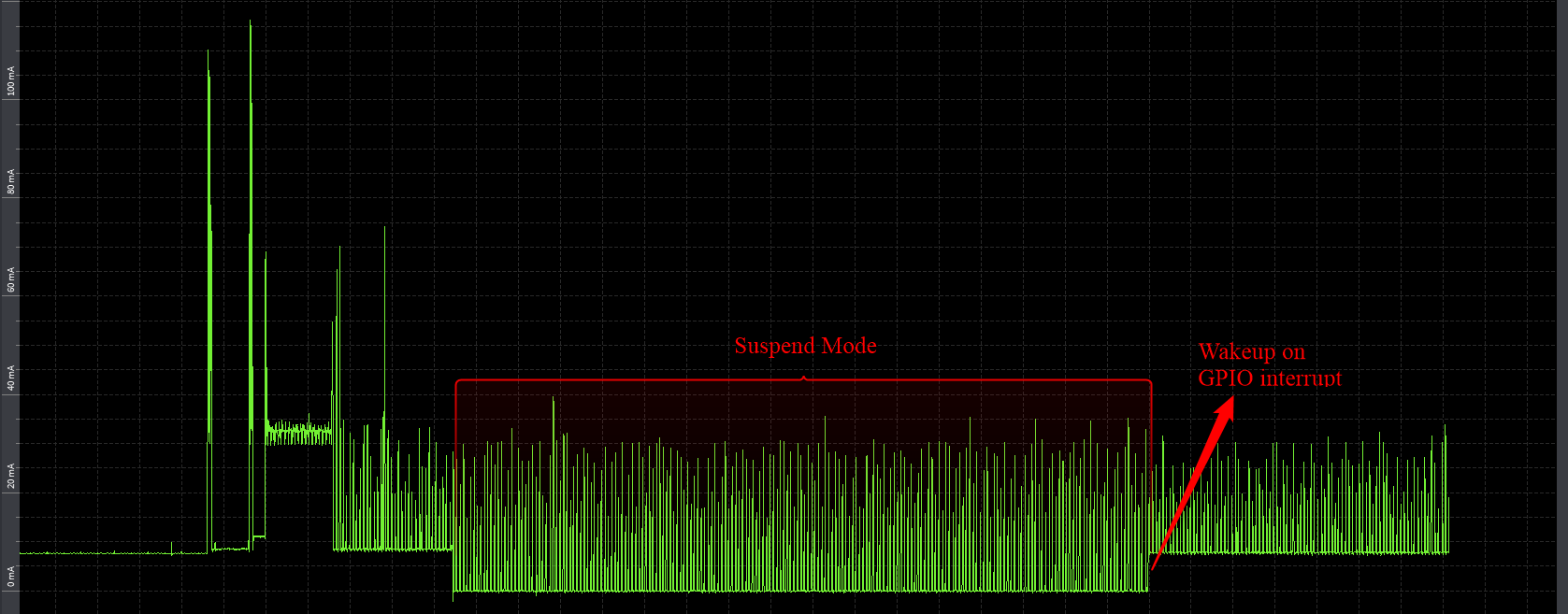


Figure 18: GPIO Interrupt - OTTI log

#### Upon Network Data Reception

This use case is to demonstrate Talaria TWO sleep management with an example of illustrating wakeup from sleep mode upon socket data receive.

**Description**

This use case describes connecting to a network of SSID InnoPhase and passphrase Inno@9070. Creates TCP server socket at port 9000. Waits for the TCP client connection and configures module deep sleep of 60 seconds.

During the 60 second timestamp, if there is data sent from the client, module wakes up upon network data and receives data and goes back to sleep till timeout.

**Prerequisites**

1. An Access Point configured with WPA/WPA2/WPA3 personal/enterprise security.
2. Hercules tool to create TCP client in Windows/Linux laptop.

**AT Command Sequence**

|  |  |
| --- | --- |
| **AT Command** | **Description** |
| *at* | Check communication state |
| *at+wscan* | Get list of available APs from the vicinity |
| *at+wcon* | Connect to a desired AP from the received scan results |
| *at+wstatus* | Get IP address of Talaria TWO to verify the connection(L2+L3) is successful |
| *at+socsrv* | Start TCP server |
| *at+syssleep* | Puts system into suspend mode for the configured time (Time in milliseconds) |

Table 9: Network data reception - AT Commands

**Procedure**

**Step 1**: Execute the following commands on Talaria TWO:

|  |
| --- |
| at  at+wscan  at+wcon=InnoPhase,Inno@9070  at+wstatus=0  at+socsrv=0,0,0,9000  at+syssleep=10000 |

**Step 2**: Connect your laptop to the same access point InnoPhase to which the Talaria TWO is connected.

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 19: Connect to InnoPhase

**Step 3**: Using Hercules tool create TCP client and connect to server of IP 192.1681.18 server socket and port 9000.

Graphical user interface, text, application

Description automatically generated

Figure 20: Hercules tool - create TCP client

**Serial Log**

Text

Description automatically generated

Figure 21: Network data reception - Serial log

## BLE Peripheral

### Description

This use case is to demonstrate configuring Talaria TWO as a BLE peripheral mode.

1. Start the GATT server.
2. Configure services, add characteristics, and advertise the services.
3. Connect BLE central
4. Perform READ/WRITE operations from the application endpoint (BLE Central, Android phone with BLE Scanner app in this example).

### Prerequisites

Android device installed with BLE app like BLE Scanner or nRF Connect.

### AT Command Sequence

|  |  |
| --- | --- |
| **AT Command** | **Description** |
| *at* | Checks the connection state |
| *at+btinit* | Initializes the BLE stack |
| *at+blecfg=02:03:04:1f:cc:9e,0, InnoPhase* | Configures BLE device with parameters,  BLE mac address: 02:03:04:1f:cc:9e  BLE address type: 0 (BLE public address)  BLE device name: InnoPhase |
| *at+blesrvstart* | Start the BLE GATT server |
| *at+bleservcfg=11111111111111111111111111111111* | Creates unique custom service ID (32-bit long value)  UUID: *11111111111111111111111111111111* |
| *at+blechradd=11111111111111111111111111111111,2A29,a,3* | Creates a unique custom characteristic ID for the above created custom service ID with parameters,  Services uuid: *11111111111111111111111111111111*  Char uuid: *2A29*  Properties: *"a" sets read and write property*  Permission: 3sets read and write permission |
| *at+bleservadd=11111111111111111111111111111111* | This includes the customized service into GATT server where,  Custom service UUID: *11111111111111111111111111111111* |
| *at+bleadvcfg=100,0,1600,0* | This configures the advertisement parameters where,  Fast adv interval: 100 which equals 62.5ms (100 \* 625 μs)  Fast adv duration: 0 (continues advertisement)  Slow adv interval: 1600, which equals 1000ms (1600 \* 625 μs)  Slow adv duration: 0 |
| *at+bleadvstart* | Start the BLE Advertisement |
| *at+blecharwrdata=2a29,5* | Acknowledges BLE Characteristic Write Request with parameters,  characteristic UUID: *2a29,*  data len (in hex format): 5 |
| *at+blecharrddata=2a29,a,6162636465* | Sends data for BLE Characteristic Read Request with parameters,  uuid: *2a29*  data len*: a*  data: *6162636465* |

Table 10: BLE Peripheral - AT Commands

### Procedure

**Step 1**: Execute the following commands on Talaria TWO:

|  |
| --- |
| at  at+btinit  at+blecfg=02:03:04:1f:cc:9e,0,InnoPhase  at+blesrvstart  at+bleservcfg=11111111111111111111111111111111  at+blechradd=11111111111111111111111111111111,2A29,a,3  at+bleservadd=11111111111111111111111111111111  at+bleadvcfg=100,0,1600,0  at+bleadvstart |

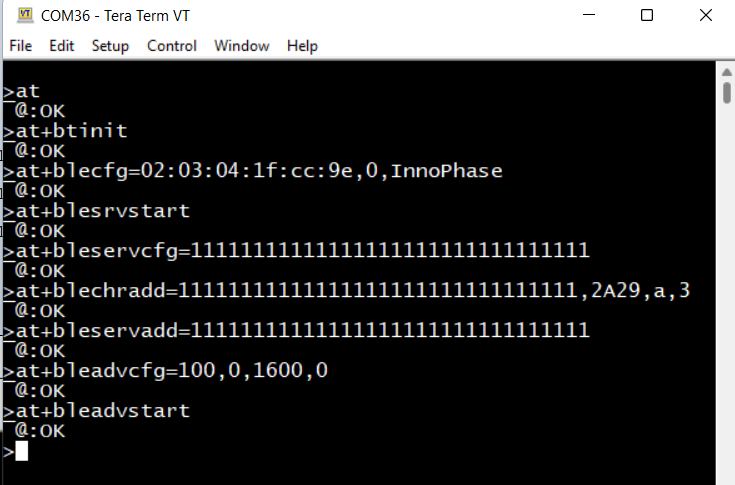


Figure 22: BLE Peripheral - serial log

**Step 2**: Launch the BLE Scanner app from the Android device and connect to Talaria TWO GATT Server with the BLE name InnoPhase.

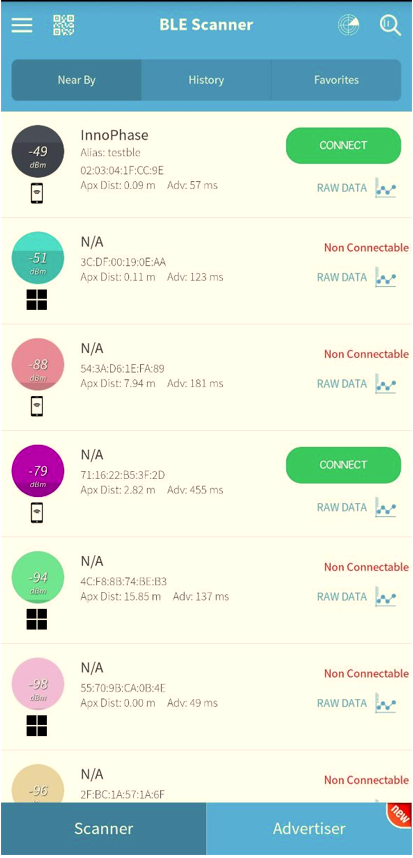


Figure 23: Talaria TWO GATT Server

**Step 3**: Once the BLE connection is established successfully send data/Write Request by pressing W from BLE central.

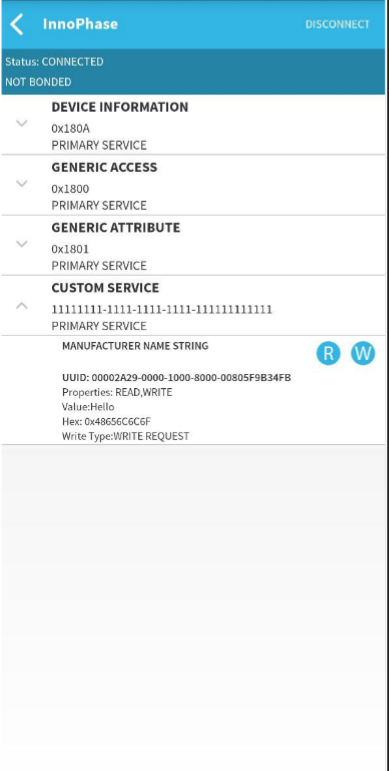


Figure 24: Write request

**Step 4**: An asynchronous message on Talaria TWO will be displayed for the Write Request sent from the device. Send an acknowledgment of the write request from Talaria TWO using following command:

|  |
| --- |
| at+blecharwrdata=2a29,5 |

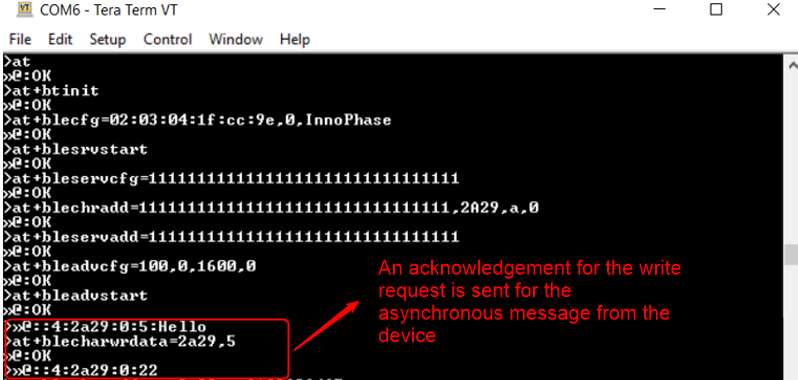


Figure 25: Write request from Talaria TWO

Write request acknowledgement displayed on the device:

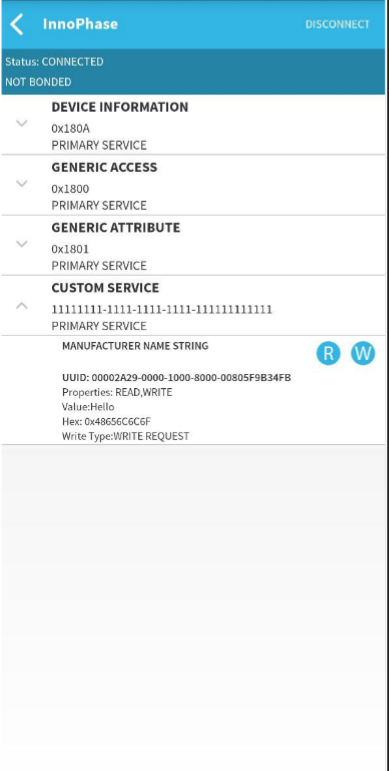


Figure 26: Write request acknowledgment on device

**Step 5**: Send Read request from device BLE central and send acknowledgement Read request data 6162636465 from Talaria TWO.

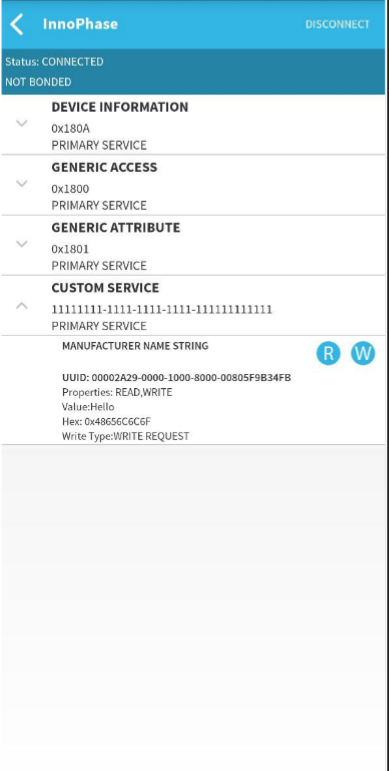


Figure 27: Read request on device

**Step 6**: An asynchronous message will be displayed on Talaria TWO for the received Read request. Send data 6162636465 as acknowledgement for the received read request from Talaria TWO:

|  |
| --- |
| at+blecharrddata=2a29,a,6162636465 |



Figure 28: Send data as an acknowledgement to Talaria TWO

Acknowledgment of Read request displayed on the device.

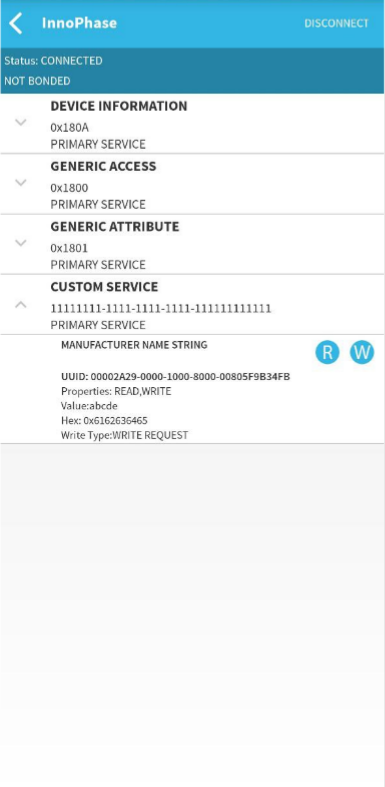


Figure 29: Read request acknowledgment on device

## BLE IO capabilities

### Description

This use case is to demonstrate the BLE IO capabilities for secured BLE connection.

### Prerequisites

Android device installed with native BLE app (BLE Scanner/nRF Connect).

### AT Command Sequence

|  |  |
| --- | --- |
| **AT Command** | **Description** |
| *at* | Checks the connection state |
| *at+btinit* | Initializes the BLE stack |
| *at+blecfg=02:03:04:1f:cc:9e,0,InnoPhase\_Secure* | Configures BLE device with parameters,  BLE mac address: 02:03:04:1f:cc:9e  BLE address type: 0 (BLE public address)  BLE device name: InnoPhase\_Secure |
| *at+blesrvstart* | Start the BLE GATT server |
| *at+bleservcfg=11111111111111111111111111111111* | Creates unique custom service ID (32-bit long value)  UUID: *11111111111111111111111111111111* |
| *at+blechradd=11111111111111111111111111111111, 2A29,a,3* | Creates a unique custom characteristic ID for the above created custom service ID with parameters,  Services uuid: *11111111111111111111111111111111*  Char uuid: *2A29*  Properties: *"a" sets read and write property*  Permission: *3* sets read and write permission |
| *at+bleservadd=11111111111111111111111111111111* | This includes the customized service into gatt server where,  Custom service UUID: *11111111111111111111111111111111* |
| *at+bleadvcfg=100,0,1600,0* | This configures the advertisement parameters where,  Fast adv interval: 100 which equals 62.5ms (100 \* 625 μs)  Fast adv duration: 0 (continues advertisement)  Slow adv interval: 1600, which equals 1000ms (1600 \* 625 μs)  Slow adv duration: 0 |
| *at+blesmpcfg=0,0,1,0,0,0,16,1* | Configures the SMP (security) with parameters,  io cap: 0 (display\_only)  oob : 0 (default)  bondable: 1  mitm: 0  sc: 0  keypress: 0  key size min: 16 bytes  encrypt: 1 |
| *at+bleadvstart* | Start the BLE Advertisement |
| *at+blecharwrdata=2a29,5* | Acknowledges BLE Characteristic Write Request with parameters,  characteristic UUID: *2a29,*  data len (in hex format): 5 |
| *at+blecharrddata=2a29,a,6162636465* | Sends data for BLE Characteristic Read Request with parameters,  uuid: *2a29*  data len*: a*  data: *6162636465* |

Table 11: BLE IO capabilities - AT Commands

### Procedure

**Step 1**: Execute the following commands on Talaria TWO:

|  |
| --- |
| at  at+btinit  at+blecfg=02:03:04:1f:cc:9e,0,InnoPhase\_Secure  at+blesrvstart  at+bleservcfg=11111111111111111111111111111111  at+blechradd=11111111111111111111111111111111,2A29,a,3  at+bleservadd=11111111111111111111111111111111  at+bleadvcfg=100,0,1600,0  at+blesmpcfg=0,0,1,0,0,0,16,1  at+bleadvstart |

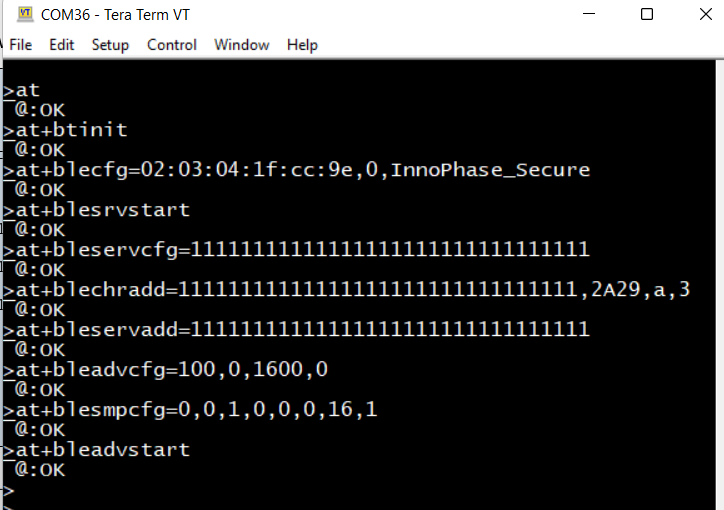


Figure 30: BLE IO capability - serial log

**Step 2**: Launch BLE Scanner app from the Android device and connect to Talaria TWO GATT server with the BLE name InnoPhase\_Secure

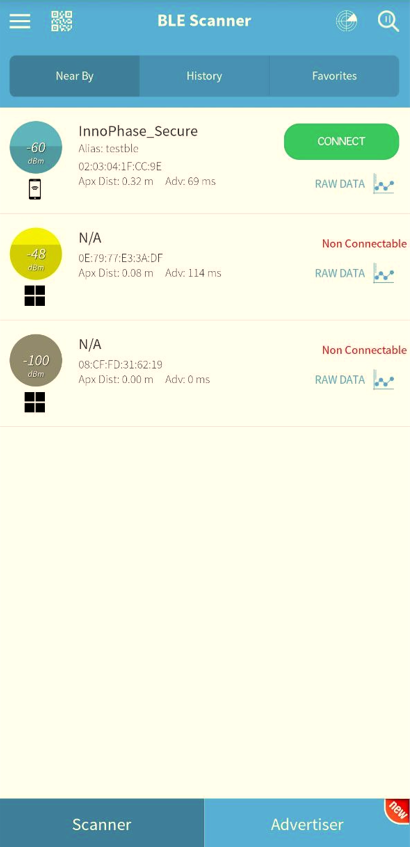


Figure 31: Talaria TWO GATT Server

**Step 3**: For the BLE GATT connection app requests for pairing as a pairing request, click on pair & connect.

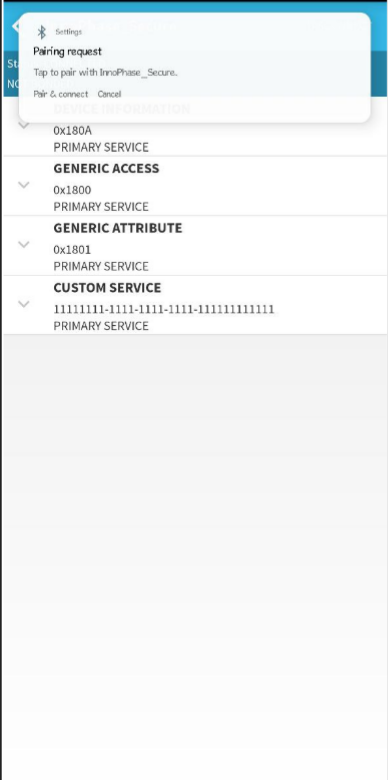


Figure 32: Pair request on the device

**Step 4**: App requests for a confirmation Pair with InnoPhase\_secure here click on Pair.

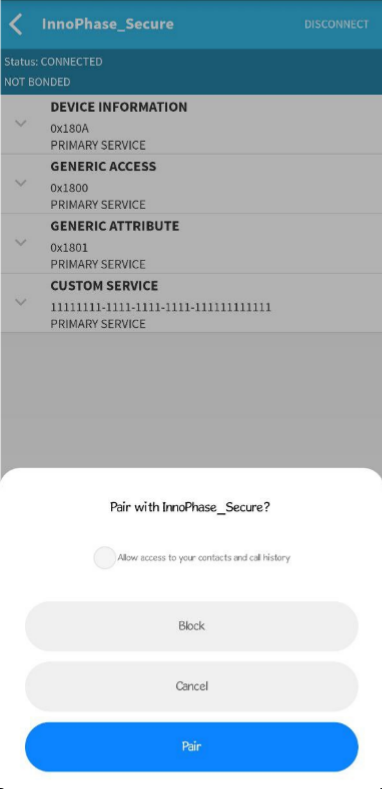


Figure 33: Confirmation to pair

Now, the passkey for pairing is shown on the AT command line. Add the passkey to the app and click on OK. Now the BLE GATT connection is created.

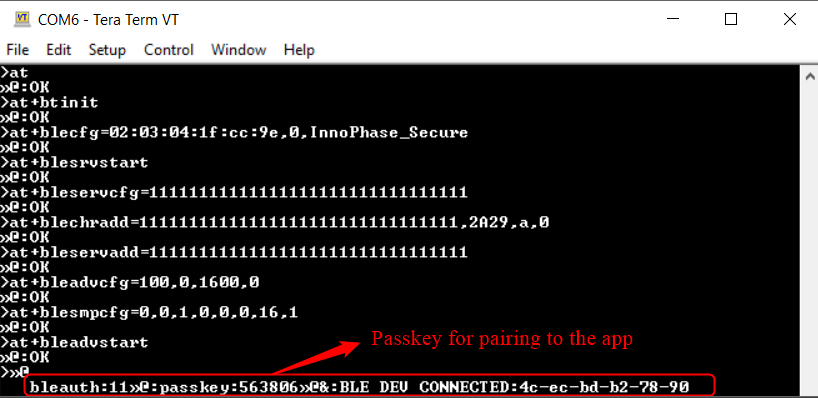


Figure 34: Pass key for pairing

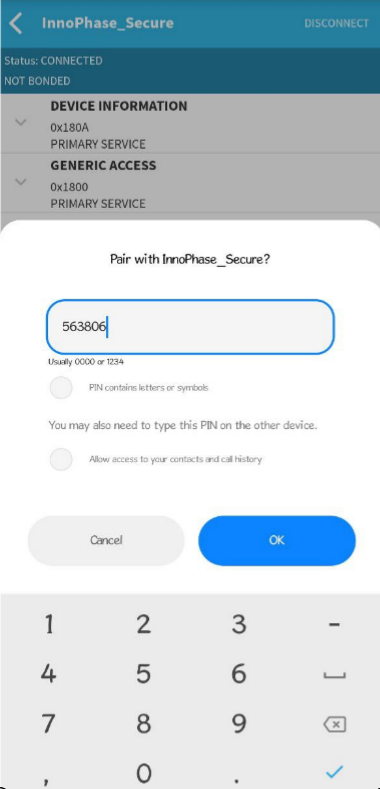


Figure 35: Add the passkey

**Step 5**: Once the BLE connection is established successfully send data/Write Request by pressing W from BLE central.

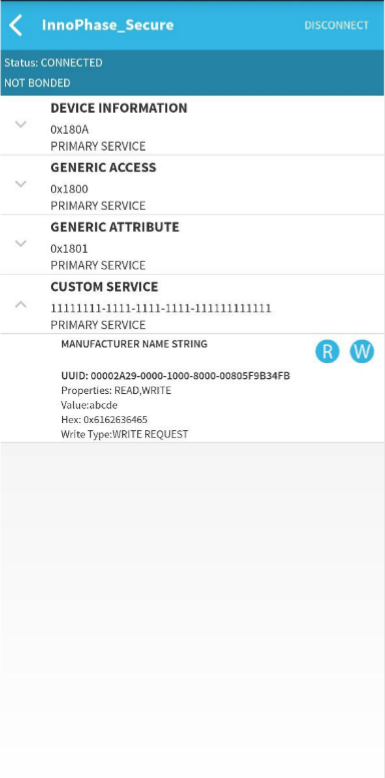


Figure 36: Write request- SMP

An asynchronous message on Talaria TWO will be displayed for the Write Request sent from the device. Send an acknowledgment of the write request from Talaria TWO using following command:

|  |
| --- |
| at+blecharwrdata=2a29,5 |

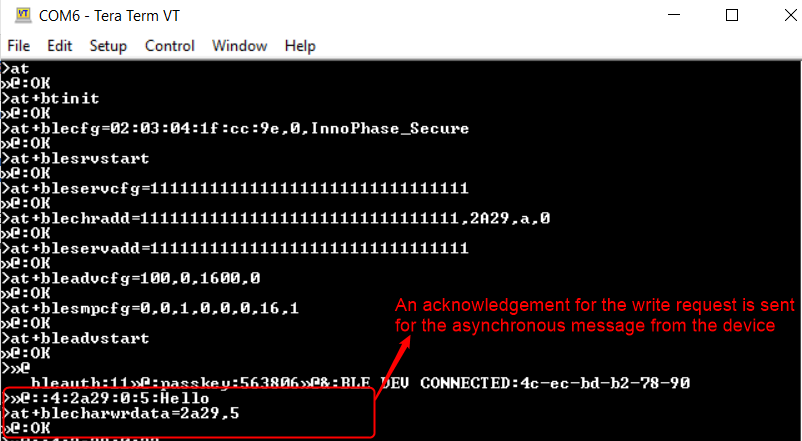


Figure 37: Acknowledgment of Write request-SMP

Write request acknowledgement displayed on the device:

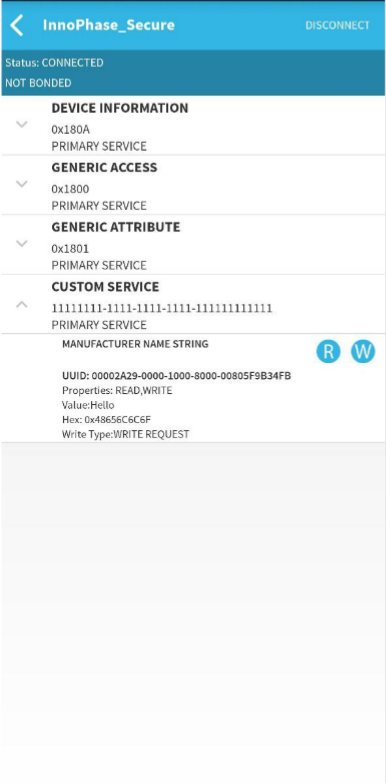


Figure 38: Write request acknowledgment on device-SMP

**Step 6**: Send Read request from device BLE central and send acknowledgement Read request data 6162636465 from Talaria TWO.

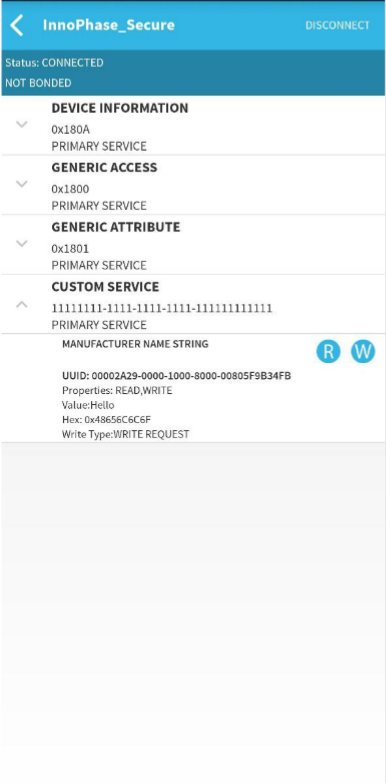


Figure 39: Read Request- SMP

An asynchronous message will be displayed on Talaria TWO for the received Read request. Send data 6162636465 as acknowledgement for the received read request from Talaria TWO:

|  |
| --- |
| at+blecharrddata=2a29,a,6162636465 |



Figure 40: Read request acknowledgment

Acknowledgment of Read request displayed on the device:

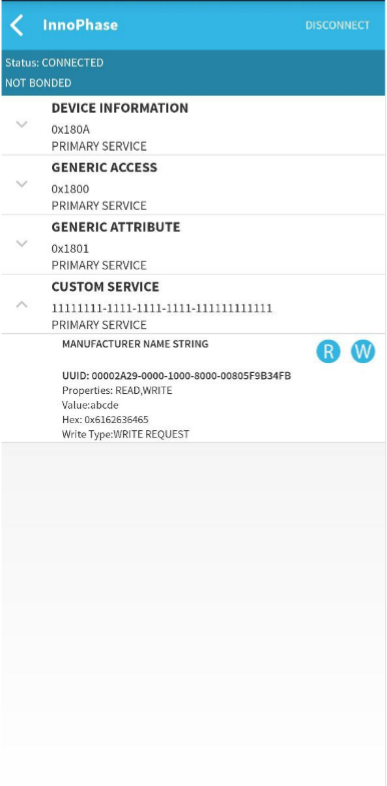


Figure 41: Read request acknowledgment on device