AN14111

Target Wake Time (TWT) on RW61x

Rev. 2.0 — 3 September 2025

Application note

Document information

Information	Content
Keywords	target wake time, wake-up interval, agreement phase, negotiation phase, service period, low-power state, power-down state, active state
Abstract	Describes the target wake time feature and provides examples for RW61x.

1 Overview

IEEE 802.11ax standard supports the target wake time (TWT) feature for 2.4 GHz and 5 GHz. The feature is used to manage the activity of the Wi-Fi stations (STA) and access points (AP) in a basic service set (BSS). The TWT parameter values define how often and for how long a STA is in active state to send and/or receive data.

Compared to the standard power save modes, the power consumption of 802.11ax devices with TWT enabled is reduced. The spectral efficiency is also optimized as there is less contention and overlap between users. TWT is most beneficial for IoT devices such as sensors and actuators.

1.1 Prerequisites

By default, when RW61x is in low-power mode (PM3)[1], the on-chip SRAM is powered down and no data is retained in the memory. To disable the data retention loss in the memory[2]:

- Download the latest SDK from ref.[2].
- Edit the *host_sleep.h* file located in the *rdrw612bga/wifi_examples/wifi_cli/* directory:
  - Remove the macro `#define WLAN_MEM_PD_CFG (1UL << 8)`
  - Add the macro `#define WLAN_MEM_PD_CFG ((1UL << 8) | 0x3FUL)`

  Note: *`0x3FUL` is the value used to change the configuration of the register that controls the SRAM status in PM3 mode.*
- Build wifi_cli application from MCUXpresso IDE/IAR/ARMGCC/MDK. Refer to ref.[1].

[1] Figure 5 shows an example of RW61x current consumption measurement when the MCU is in low-power mode.

[2] To fully benefit from TWT, the change is required.

2 Sequence

The sequence of target wake time includes:

- The negotiation phase: the AP and STAs negotiate the TWT parameters to access the medium.
- The agreement phase: the AP and STAs agree on the TWT parameters and service period (SP).
- The service period (SP): the wake-up duration. The time during which the STA must be awake and active.

Figure 1 shows the negotiation phase, agreement phase, wake-up duration/service period in the TWT sequence.

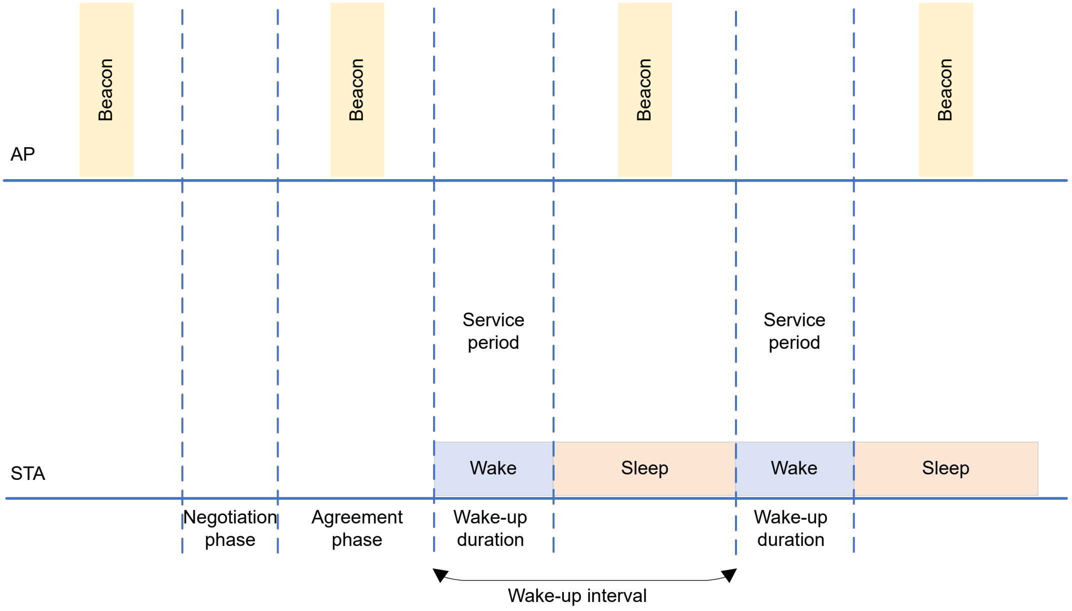


Figure 1. TWT sequence

3 TWT parameters

During the negotiation phase, the AP and STA use the following TWT parameter values:

- Explicit/implicit: the AP sends (does not send) frames in the current service period (SP). The frames notify a STA of the start of the next SP.
- Announced/unannounced: at the beginning of the SP, the STA sends (does not send) a QoS NULL frame with active mode (PM0). The QoS frame notifies the AP that the STA is awake.
- Trigger/non-trigger: the AP uses (does not use) the OFDMA trigger-based mechanism set in the SP. If the trigger-based mechanism is enabled in the SP, the STA does not send data to the AP until a trigger is received.
- TWT info disabled/enabled: indicates whether TWT information frames are acceptable or not during the TWT session.
- Individual/broadcast TWT:
  - Individual TWT (ITWT): initiated by a single STA to determine the wake interval (wake-up duration and start of sleep time)
    An 802.11ax STA can negotiate an individual TWT agreement.
  - Broadcast TWT (BTWT): scheduled by an AP in its beacon frames. The AP sets up a shared TWT session for a group of STAs, and restates the TWT parameters.
    An 802.11ax STA can establish a membership in broadcast TWT (BTWT) schedules.
  - ITWT and BTWT are used to set schedule parameters during the negotiation phase.
- Wake-up duration: time during which the STA must be awake and active.
- Flow identifier: up to eight TWT sessions can run in parallel based on a unique 3-bit flow identifier.
- Tweak/Not tweak: if the TWT parameters are not agreed upon, the FW can internally tweak/not tweak the parameters to renegotiate these.
- Exponent: a 1-byte value used to calculate the wake-up interval. See Section 4.
- Mantissa: a 1-byte value used to calculate the wake-up interval. See Section 4.
- Request/suggest:
  - Request: the STA requests the TWT parameters from the AP.
  - Suggest: the STA indicates the TWT parameters to the AP.

4 Wake-up interval calculations

The wake- up interval is determined by the target wake time exponent and mantissa (Section 3).

The formula is:

$$Wake\text{-}up\ interval = Mantissa\ \times 2^{Exponent} \tag{1}$$

Example:

- Exponent = 0x14 = 20 µs
- 2^20 µs = 1048576 us ~ 1 s
- Mantissa = 0x3c = 60 s
- 60 s/1 s = 60 ticks

For every second (exponent), 60 ticks occur.

5 Commands

The section details the commands used to configure ITWT and BTWT.

5.1 ITWT

Command to configure ITWT parameters using the Wi-Fi CLI demo application:

```
# wlan-11ax-twt-setup <option> <param_id> <param_data>
```

Table 1. Command parameters

Parameter	Definition
`option`	`dump` = show all TWT parameters `set` = set a single TWT parameter `done` = apply the TWT parameters
`param_id`	Only applicable when `option` = `set` Identifier of a parameter [0] = explicit/implicit • 0x0 = explicit: the TWT agreement is explicitly negotiated between the STA and AP. • 0x1 = implicit: the TWT agreement is implicitly understood without detailed negotiation. [1] = unannounced/announced • 0x0 = unannounced: the TWT schedule is not announced to other STAs. • 0x1 = announced: the TWT schedule is announced to other STAs, allowing them to synchronize. [2] = non-triggered/triggered • 0x0 = non-triggered: the STA wakes up at the scheduled TWT without needing a trigger frame. • 0x1 = triggered: the STA requires a trigger frame from the AP to wake up at the scheduled TWT. [3] = enable/disable TWT info • 0x0 = enable: TWT information is included in the management frames. • 0x1 = disable: TWT information is not included in the management frames. [4] = individual/broadcast • 0 = individual: the TWT agreement is for an individual STA. • 1 = broadcast: the TWT agreement is for multiple STAs (used in broadcast scenarios). [5] = wake-up duration (hexadecimal value) • Specifies the duration for which the STA remains awake during each TWT session. [6] = flow identifier • Identifies the TWT flow (useful when managing multiple TWT sessions). Range: [0-7] [7] = hard constraint (tweak/not tweak) • 0 = tweak: the TWT schedule can be adjusted slightly to accommodate other network activities. • 1 = no Tweak: the TWT schedule is fixed and cannot be adjusted. [8] = exponent • Specifies the exponent part of the wake interval. Used to calculate the actual wake interval. [9] = mantissa • Specifies the mantissa part of the wake interval. Used in conjunction with the exponent to calculate the wake interval. [10] = request/suggest TWT • 0x0 = request TWT • 0x1 = suggest TWT

Table 1. Command parameters*...continued*

Parameter	Definition
`param_data`	Only applicable when `option` = set 2-byte hexadecimal data corresponding to the value of the parameter identified by `param_id` (in bytewise little-endian format)

Example of command to set the exponent:

```
# wlan-11ax-twt-setup set 8 14
```

Example of command output:

```
twt_exponent #### Range: [0-63]
[8]: 0x14
```

Example of command to set the mantissa:

```
# wlan-11ax-twt-setup set 9 3c 0
```

Example of command output:

```
twt_mantissa #### Range: [0-65535]
[9]: 0x3c 0x00
```

Example of command to show all TWT parameters:

```
#wlan-11ax-twt-setup dump
```

Example of command output:

```
twt_exponent #### Range: [0-63]
cfg[twt_setup] len[12] param_num[11]:
implicit #### 0: TWT session is explicit, 1: Session is implicit
[0]: 0x01
announced #### 0: Unannounced, 1: Announced TWT
[1]: 0x00
trigger_enabled #### 0: Non-Trigger enabled, 1: Trigger enabled TWT
[2]: 0x00
twt_info_disabled #### 0: TWT info enabled, 1: TWT info disabled
[3]: 0x01
negotiation_type #### 0: Individual TWT, 3: Broadcast TWT
[4]: 0x00
twt_wakeup_duration #### time after which the TWT requesting STA can transition to doze
 state
[5]: 0x40
flow_identifier #### Range: [0-7]
[6]: 0x00
hard_constraint #### 0: FW can tweak the TWT setup parameters if it is rejected by AP,
  1: FW should not tweak any parameters
[7]: 0x01
twt_exponent #### Range: [0-63]
[8]: 0x14
twt_mantissa #### Range: [0-65535]
[9]: 0x39 0x00
twt_request #### Type, 0: REQUEST_TWT, 1: SUGGEST_TWT
[10]: 0x00
```

Command to set all TWT parameters:

```
#wlan-11ax-twt-setup done
```

5.2 BTWT

The wlan-11ax-bcast-twt command configures the BTWT parameters on the uAP side. The command uses the Wi-Fi CLI demo application.

Command syntax:

```
# wlan-11ax-bcast-twt <option> <sta_wait> <offset> <twtli> <session_num>
 <id0> <mantissa0> <exponent0> <nominal_wake0> <id1> <mantissa1> <exponent1>
 <nominal_wake1> ...
```

Table 2. Command parameters

Parameter	Definition

option	get = show all BTWT parameters
	set = set BTWT parameter

sta_wait	Set to 0
offset	Set to 0
twtli	Set to 0
session_num	Number of broadcast sessions Range: [2-5]
id(x)[1]	Session ID
mantissa(x) [1]	Mantissa of the session number (IDx). Specifies the mantissa part of the wake interval, used in conjunction with the exponent to calculate the wake interval.
exponent(x) [1]	Exponent of the session ID (IDx). Specifies the exponent part of the wake interval, used in calculating the actual wake interval.
nominal_wake(x) [1]	Nominal wake parameter of the session ID (IDx). Minimum time during which a STA must be awake in each TWT interval. Unit in 256 µs.

[1] x is the session number

Example of command to set two sessions on the uAP side:

```
# wlan-11ax-bcast-twt set 0 0 0 2 0 112 10 128 1 32 10 64
```

In the command example:

- Parameter values for session 0: fid=0, mantissa=112, exp=10, nominal wake=128
- Parameter values for session 1: fid=1, mantissa=32, exp=10, nominal wake=64

Example of command to get all BTWT parameters:

```
# wlan-11ax-bcast-twt get
```

Example of command output:

```
btwt_cfg, bet_sta_wait 0, offset 0, twtli 0, count 2
id 0, mantissa 112, exponent 10, nominal_wake 128
id 1, mantissa 32, exponent 10, nominal_wake 64
```

Example of commands to set BTWT parameters on the STA side:

```
# wlan-11ax-twt-setup set 3 0x01 //disable twt_info
# wlan-11ax-twt-setup set 4 0x03 //choose btwt
# wlan-11ax-twt-setup set 6 0x01 // set flow identifier 1
```

5.3 Teardown

The `teardown` command is used to end all TWT sessions. The command is used with the Wi-Fi CLI demo application.

Note: *The `teardown` command must be executed before a STA sets up another BTWT session with a different flow ID.*

```
# wlan-11ax-twt-teardown <option> <param_id> <param_data>
```

Table 3. Command parameters

Parameter	Definition
`option`	`dump` = show all teardown parameters `set` = set a teardown parameter `done` = apply the teardown parameters
`param_id`	Only applicable when `option` = `set` Identifier of a parameter 0 = flow identifier 1 = negotiation type 2 = teardown all TWT
`param_data`	Only applicable when `option` = `set` 2-byte hexadecimal data corresponding to the value of the parameter identified by `param_id` (in bytewise little-endian format)

Example of command to set the flow identifier:

```
# wlan-11ax-twt-teardown set 0 0x02
```

Example of command output:

```
FlowIdentifier #### Range: [0-7]
[0]: 0x02
```

Example of command to show all teardown parameters:

```
# wlan-11ax-twt-teardown dump
```

Example of command output:

```
cfg[twt_teardown] len[3] param_num[3]:
FlowIdentifier #### Range: [0-7]
[0]: 0x02
NegotiationType #### 0: Future Individual TWT SP start time, 1: Next Wake TBTT tim
[1]: 0x00
TearDownAllTWT #### 1: To teardown all TWT, 0 otherwise
[2]: 0x00
```

Example of command to apply all the teardown parameters:

```
# wlan-11ax-twt-teardown done
```

Example of command output:

```
send config [twt_teardown] ret 0
```

5.4 Report

The `report` command gets the TWT reports that include the configured TWT parameters of the AP. The command is used with the Wi-Fi CLI demo application.

Note: *The command applies to ITWT in AP mode and to BTWT in STA mode.*

```
# wlan-11ax-twt-report
```

Table 4. Command return parameters

Parameter	Definition
`id[N]`[1]	TWT session
`param_data`	Configured TWT session parameters

[1] N refers to the Nth TWT session configured.

Example of command output:

```
twt_report results:
Received B-TWT schedule from ex-AP's beacon. Total buff len = 18, count of schedules = 2,
 detail:
Schedule- [0]:
0x08 0x28 0x90 0xa6 0x80 0x70 0x00 0x00 0xff
## Explain: Broadcast TWT ID = 0; No trigger, Announced; Interval Exponent = 10, Mantissa
 = 112; Wake Duration = 128
## BTWT_ID [0] will be auto joined when STA join other BTWT schedule. Don't manually join
 it.
Schedule- [1]:
0x28 0x28 0xf0 0xa6 0x20 0x00 0x01 0x08 0xff
## Explain: Broadcast TWT ID = 1; No trigger, Announced; Interval Exponent = 10, Mantissa
 = 256; Wake Duration = 32
```

6 Examples

The section includes examples for ITWT and BTWT.

6.1 ITWT example

The example uses two RW61x boards. One board is brought up as AP and the other board is brought up as STA.

Figure 2 shows the sequence of ITWT example.

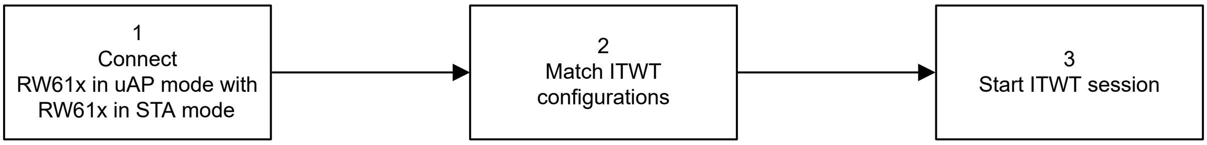


Figure 2. Sequence of ITWT example

Step 1 – Bring up one RW61x in uAP mode.

```
wlan-set-mac <MAC_address> // set MAC address
wlan-add <profile_name> ssid <ssid> ip:<ip_addr>,<gateway_ip>,<netmask> role uap channel
 <channel> // set the profile name, SSID, IP address, etc
wlan-start-network <profile_name> //start the AP
```

Step 2 – Bring up one RW61x in STA mode and connect the STA to the AP.

```
wlan-set-mac <MAC_address> // set MAC address
wlan-add <profile_name> ssid <ssid> <security> <secret_password> channel <channel> // set
 the profile name, SSID, etc of the uAP joining
wlan-connect test // connect to uAP
```

Step 3 – To further reduce the power consumption, enable IEEE power save mode on the STA.

```
wlan-ieee-ps 1
```

Step 4 – Configure ITWT parameters on RW61x in STA mode.

Commands for a wake-up interval of 15 minutes:

```
wlan-11ax-twt-setup set 8 14 // configure exponent to 0x14
wlan-11ax-twt-setup set 9 84 3 // configure mantissa to 0x0384
wlan-11ax-twt-setup done // apply configurations
```

Commands for a wake-up interval of 20 minutes:

```
wlan-11ax-twt-setup set 8 14 // configure exponent to 0x14
wlan-11ax-twt-setup set 9 b0 4 // configure mantissa to 0x04b0
wlan-11ax-twt-setup done // apply configurations
```

6.2 BTWT example

The example uses two RW61x boards. One board is brought up as AP and the other board is brought up as STA.

Figure 3 shows the sequence of BTWT example.

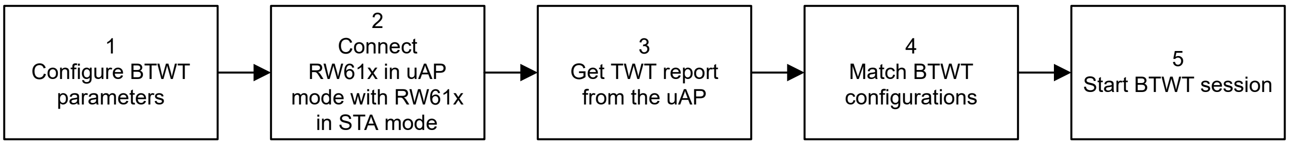

1
Configure BTWT parameters
2
Connect RW61x in uAP mode with RW61x in STA mode
3
Get TWT report from the uAP
4
Match BTWT configurations
5
Start BTWT session

Figure 3. Sequence of BTWT example

Step 1 – configure BTWT parameters.

```
wlan-11ax-bcast-twt set <sta_wait> <offset> <twtli> <session_num> <id0> <mantissa0>
 <exponent0> <nominal_wake0> <id1> <mantissa1> <exponent1> <nominal_wake1> ...
```

Example of command:

```
wlan-11ax-bcast-twt set 0 0 0 2 0 112 10 128 1 256 10 32
```

Step 2 – Bring up one RW61x in uAP mode.

```
wlan-add <profile_name> ssid <ssid> ip:<ip_addr>,<gateway_ip>,<netmask> role uap channel
 <channel> // set the profile name, SSID, IP address, etc
wlan-start-network <profile_name> //start the AP
```

Example of command:

```
wlan-add test_uap ssid btwtap ip:192.168.3.1,192.168.3.1,255.255.255.0 channel 6 role uap
 capa 11ax
wlan-start-network test_uap
```

Step 3 – Bring up one RW61x in STA mode and connect the STA to the AP.

```
wlan-add <profile_name> ssid <ssid> <security> <secret_password> channel <channel> // set
 the profile name, SSID, etc of the uAP joining
wlan-connect <profile_name> // connect to uAP
```

Example of command:

```
wlan-add test_sta ssid btwtap
wlan-connect test_sta
```

Step 4 – Get the TWT report.

```
# wlan-11ax-twt-report
```

Example of command output:

```
twt_report results:
Received B-TWT schedule from ex-AP's beacon. Total buff len = 18, count of schedules = 2,
 detail:
Schedule- [0]:
0x08 0x28 0x90 0xa6 0x80 0x70 0x00 0x00 0xff
## Explain: Broadcast TWT ID = 0; No trigger, Announced; Interval Exponent = 10, Mantissa
 = 112; Wake Duration = 128
## BTWT_ID [0] will be auto joined when STA join other BTWT schedule. Don't manually join
 it.
Schedule- [1]:
0x28 0x28 0xf0 0xa6 0x20 0x00 0x01 0x08 0xff
## Explain: Broadcast TWT ID = 1; No trigger, Announced; Interval Exponent = 10, Mantissa
 = 256; Wake Duration = 32
```

Step 5 – Set the TWT parameters to match the TWT report.

Set BTWT.

```
# wlan-11ax-twt-setup set 4 0x3
```

Example of command output:

```
negotiation_type #### 0: Individual TWT, 3: Broadcast TWT
[4]: 0x03
```

Set the flow identifier 1.

```
# wlan-11ax-twt-setup set 6 0x01
```

Example of command output:

```
flow_identifier #### Required if setup BTWT. Range: [0-7]
[6]: 0x01
```

Step 6 – Apply the TWT parameters and start the BTWT session.

```
# wlan-11ax-twt-setup done
```

Example of command output:

```
TWT setup success. Flow id: 1. Use this in future TWT operation.
send config [twt_setup] ret 0
```

Step 7 – Further reduce the power consumption. Enable IEEE Power-save mode on the STA.

```
wlan-ieee-ps 1
```

Step 8 – Maximize the power save.

Enable the low-power mode (PM3) on the host side of RW61x (CortexM33). And enable TWT on the controller/ wireless radio side of RW61x.

```
wlan-host-sleep 1 manual
wlan-suspend 3
```

Example of command output:

```
Enter low power mode PM3
```

7 TWT current consumption measurements

The section includes the measurements for ITWT and BTWT.

7.1 ITWT current consumption values

Table 5 includes RW61x ITWT current consumption values measured on VBAT pins at different wake-up intervals.

Table 5. RW61x ITWT current consumption values measured on VBAT pin

Conditions	Wake-up interval	BGA package [1]	QFN package [1]	Unit
MCU in sleep mode (PM3) Wi-Fi subsystem in low power mode Narrowband subsystem in power down mode	1 min	280	285	µA
	5 min	240	250	µA
	10 min	230	240	µA
	20 min	225	235	µA
	30 min	220	230	µA

[1] 0 KB memory retention

7.2 BTWT current consumption values

Table 6 shows RW61x BTWT current consumption values measured on VBAT at different wake-up intervals.

Table 6. RW61x BTWT current consumption values

Conditions	Wake-up interval	BGA package[1]	Unit
MCU in sleep mode (PM3) Wi-Fi subsystem in low power mode narrowband subsystem in power down mode	10 seconds	410	µA
	30 seconds	290	µA
	60 seconds	290	µA
MCU in sleep mode (PM4) Wi-Fi subsystem in low power mode narrowband subsystem in power down mode	10 seconds	340	µA
	30 seconds	220	µA
	60 seconds	220	µA

[1] 0 KB memory retention for the host MCU

Table 7 shows the commands used to set the BTWT configurations of RW61x in uAP mode.

Table 7. Commands to set the BTWT configurations of RW61x in uAP modes

Wake up interval time	Commands

10 seconds	`wlan-11ax-bcast-twt set 0 0 0 2 0 9900 10 64 1 9900 10 64`
	`wlan-11ax-bcast-twt done`
	`wlan-11ax-bcast-twt`
	`wlan-add testAP13 ssid BTWT`
	`ip:192.168.100.1,192.168.100.1,255.255.255.0 role uap channel 36 capa 11ax`
	`wlan-start-network testAP13`

30 seconds	`wlan-11ax-bcast-twt set 0 0 0 2 0 29700 10 64 1 29700 10 64`
	`wlan-11ax-bcast-twt done`
	`wlan-11ax-bcast-twt`
	`wlan-add testAP13 ssid BTWT`
	`ip:192.168.100.1,192.168.100.1,255.255.255.0 role uap channel 36 capa 11ax`
	`wlan-start-network testAP13`

60 seconds	`wlan-11ax-bcast-twt set 0 0 0 2 0 59400 10 64 1 59400 10 64`
	`wlan-11ax-bcast-twt done`
	`wlan-11ax-bcast-twt`
	`wlan-add testAP13 ssid BTWT`
	`ip:192.168.100.1,192.168.100.1,255.255.255.0 role uap channel 36 capa 11ax`
	`wlan-start-network testAP13`

Table 8 shows the commands used to set the BTWT configurations of RW61x in STA mode.

Table 8. Commands to set the configuration of RW61x in STA mode

Wake up interval time	STA CMD

10 seconds	`wlan-ieee-ps 1`
	`wlan-add test ssid BTWT`
	`wlan-connect test`
	`wlan-11ax-twt-setup set 3 0x01`
	`wlan-11ax-twt-setup set 4 0x03`
	`wlan-11ax-twt-setup set 6 0x01`
	`wlan-11ax-twt-setup done`
	`wlan-11ax-twt-report`
	`wlan-auto-host-sleep 1 manual`
	`wlan-suspend 3`

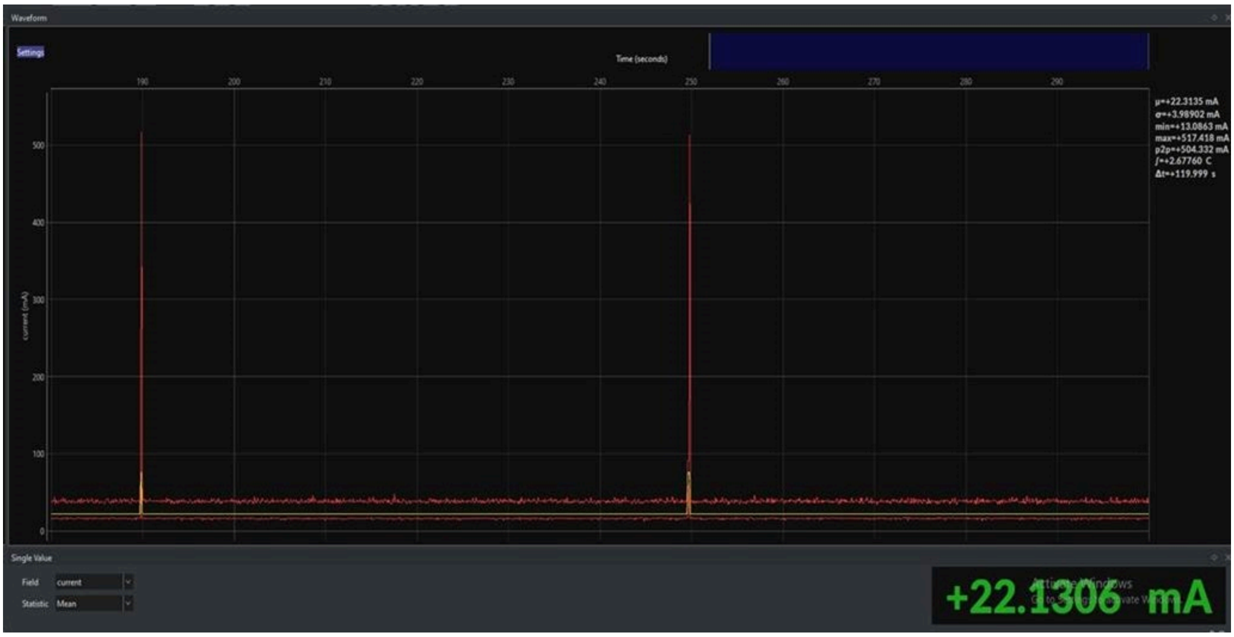
30 seconds	`wlan-ieee-ps 1`
	`wlan-add test ssid BTWT`
	`wlan-connect test`
	`wlan-11ax-twt-setup set 3 0x01`
	`wlan-11ax-twt-setup set 4 0x03`
	`wlan-11ax-twt-setup set 6 0x01`
	`wlan-11ax-twt-setup done`
	`wlan-11ax-twt-setup`
	`wlan-auto-host-sleep 1 manual`
	`wlan-suspend 3`

60 seconds	`wlan-ieee-ps 1`
	`wlan-add test ssid BTWT`
	`wlan-connect test`
	`wlan-11ax-twt-report`
	`wlan-11ax-twt-setup set 3 0x01`
	`wlan-11ax-twt-setup set 4 0x03`
	`wlan-11ax-twt-setup set 6 0x01`
	`wlan-11ax-twt-setup done`
	`wlan-11ax-twt-setup`
	`wlan-auto-host-sleep 1 manual`
	`wlan-suspend 3`

8 Appendix

8.1 Examples of current consumption measurements

Figure 4 shows the measurement of RW61x current consumption with the MCU (Cortex-M33) in active mode (PM0).



```
μ=+22.3135 mA
σ=+3.98902 mA
min=+13.0863 mA
max=+517.418 mA
p2p=+504.332 mA
∫=+2.67760  C
Δt=+119.999  s
```

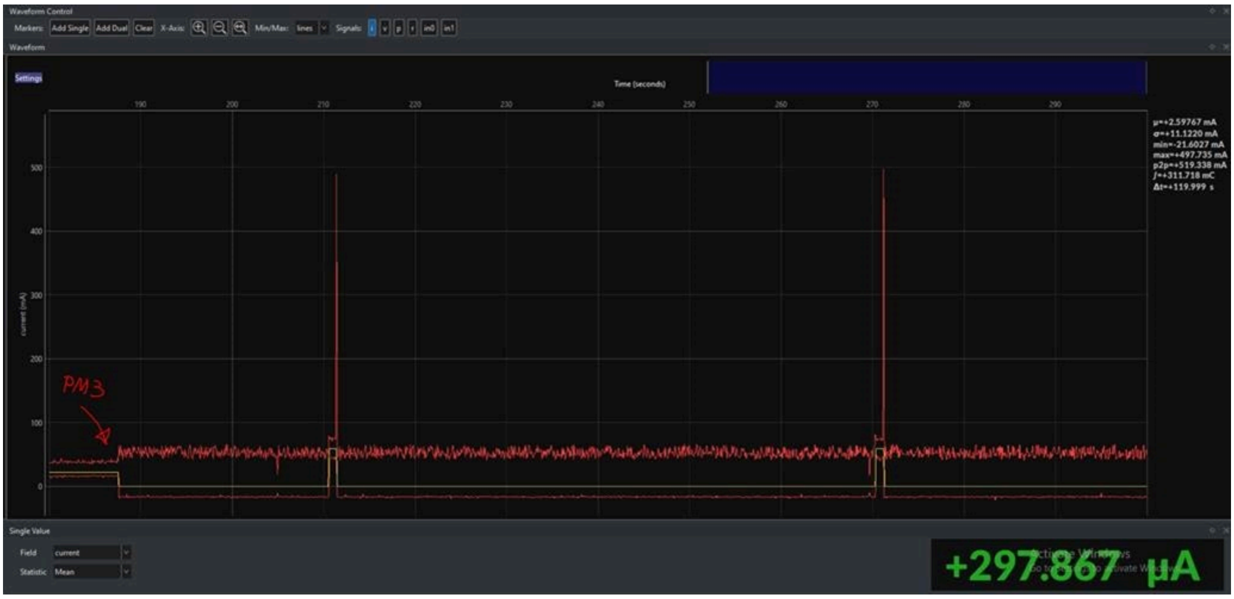
Field: current

Statistic: Mean

+22.1306 mA

Figure 4. Measurement of RW61x current consumption - MCU in active mode (PM0)

Figure 5 shows the measurement of RW61x current consumption with the MCU (Cortex-M33) in sleep mode (PM3).



μ=+2.59767 mA
σ=+11.1220 mA
min=-21.6027 mA
max=+497.735 mA
p2p=+519.338 mA
∫=+311.718 mC
Δt=+119.999 s

Field: current

Statistic: Mean

+297.867 μA

Figure 5. Measurement of RW61x current consumption - MCU in sleep mode (PM3)

8.2 How to interpret TWT report from sniffer capture

This section shows two examples of TWT reports with the sniffer captures.

First example of TWT report

```
Schedule- [0]:
0x08 0x28 0x90 0xa6 0x80 0x70 0x00 0x00 0xff
```

Figure 6 illustrates the first example of TW report.

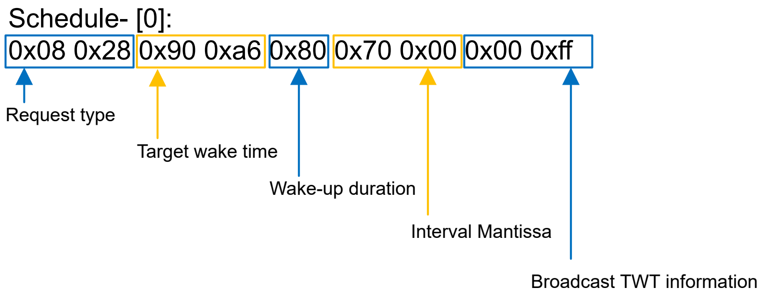
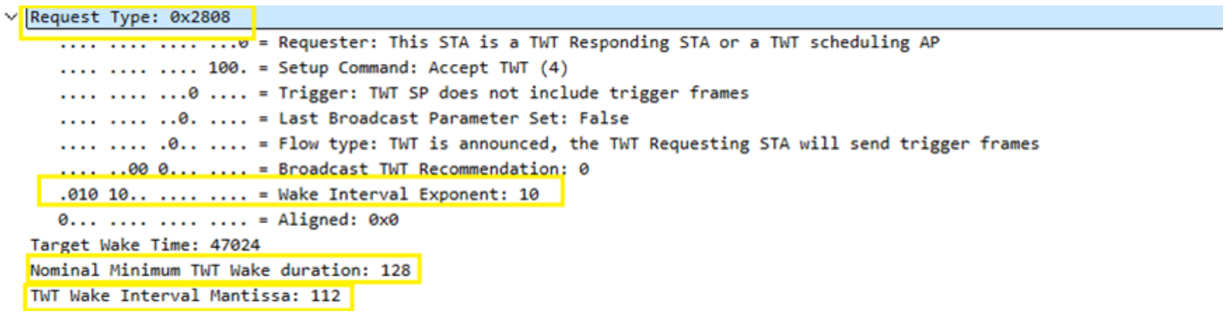


Figure 6. First example of TWT report – Schedule[0]

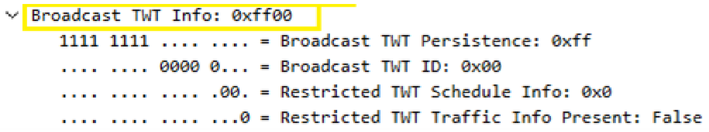
Figure 7 shows the sniffer capture for the request 0x2808.



```
Request Type: 0x2808
    .... .... .... ...0 = Requester: This STA is a TWT Responding STA or a TWT scheduling AP
    .... .... .... 100. = Setup Command: Accept TWT (4)
    .... .... ...0 .... = Trigger: TWT SP does not include trigger frames
    .... .... ..0. .... = Last Broadcast Parameter Set: False
    .... .... .0.. .... = Flow type: TWT is announced, the TWT Requesting STA will send trigger frames
    .... ..00 0... .... = Broadcast TWT Recommendation: 0
    .010 10.. .... .... = Wake Interval Exponent: 10
    0... .... .... .... = Aligned: 0x0
Target Wake Time: 47024
Nominal Minimum TWT Wake duration: 128
TWT Wake Interval Mantissa: 112
```

Figure 7. Sniffer capture for request type: 0x2808

Figure 8 shows the sniffer capture with Broadcast TWT information.



```
Broadcast TWT Info: 0xff00
    1111 1111 .... .... = Broadcast TWT Persistence: 0xff
    .... .... 0000 0... = Broadcast TWT ID: 0x00
    .... .... .... .00. = Restricted TWT Schedule Info: 0x0
    .... .... .... ...0 = Restricted TWT Traffic Info Present: False
```

Figure 8. Sniffer capture for the request type: 0x2808 – BTWT information

A second example of TWT report

```
Schedule- [1]:
0x28 0x28 0xf0 0xa6 0x20 0x00 0x01 0x08 0xff
```

Figure 9 illustrates the second example of TWT report.

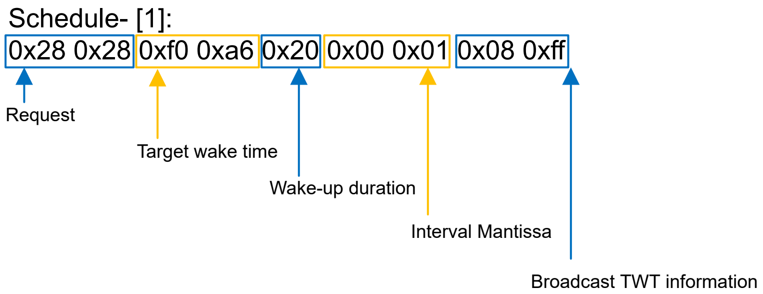
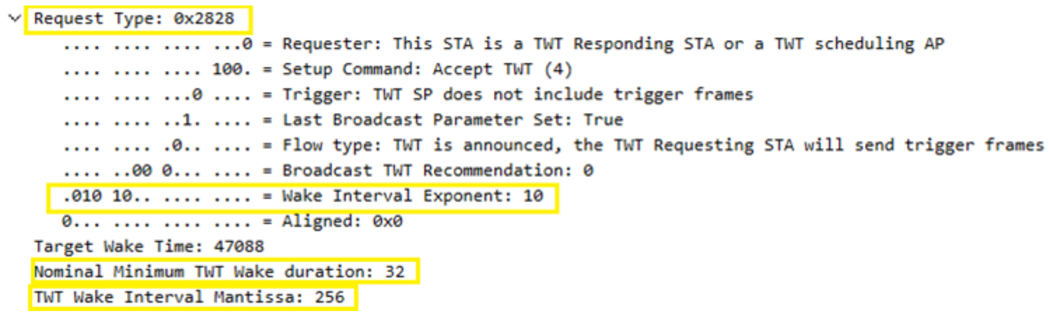


Figure 9. Second example of TWT report – Schedule[1]

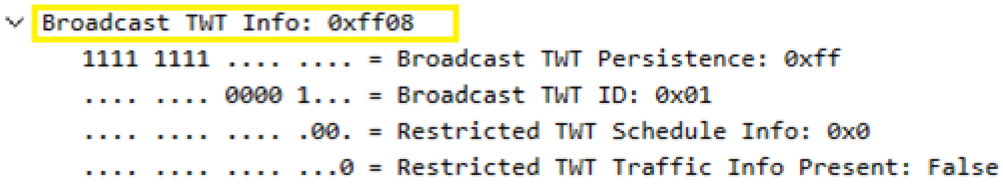
Figure 10 shows the sniffer capture for the request 0x2808.



```
Request Type: 0x2828
    .... .... .... ...0 = Requester: This STA is a TWT Responding STA or a TWT scheduling AP
    .... .... .... 100. = Setup Command: Accept TWT (4)
    .... .... ...0 .... = Trigger: TWT SP does not include trigger frames
    .... .... ..1. .... = Last Broadcast Parameter Set: True
    .... .... .0.. .... = Flow type: TWT is announced, the TWT Requesting STA will send trigger frames
    .... ..00 0... .... = Broadcast TWT Recommendation: 0
    .010 10.. .... .... = Wake Interval Exponent: 10
    0... .... .... .... = Aligned: 0x0
Target Wake Time: 47088
Nominal Minimum TWT Wake duration: 32
TWT Wake Interval Mantissa: 256
```

Figure 10. Sniffer capture for request type: 0x2828

Figure 11 shows the sniffer capture with broadcast TWT information for the request 0x2828.



```
Broadcast TWT Info: 0xff08
    1111 1111 .... .... = Broadcast TWT Persistence: 0xff
    .... .... 0000 1... = Broadcast TWT ID: 0x01
    .... .... .... .00. = Restricted TWT Schedule Info: 0x0
    .... .... .... ...0 = Restricted TWT Traffic Info Present: False
```

Figure 11. Sniffer capture for the request type: 0x2828 – BTWT information

9 Abbreviations

Table 9. Abbreviations

Acronym	Description
AP	Access point
BSS	Basic service set
BTWT	Broadcast TWT
ITWT	Individual TWT
QoS	Quality of service
SP	Service period
STA	Station
TWT	Target wake time

10 References

[1] User manual – UM11799: NXP Wi-Fi and Bluetooth Demo Applications for RW61x (link)

[2] Webpage – Select board | MCUXpresso SDK builder (link)

11 Note about the source code in the document

The example code shown in this document has the following copyright and BSD-3-Clause license:

12 Revision history

Table 10. Revision history

Document ID	Release date	Description
AN14111 v.2.0	3 September 2025	• Section 3 "TWT parameters": updated the definition of *Individual/broadcast TWT*. • Section 5.1 "ITWT": updated the command and the command parameter definitions. • Section 5.2 "BTWT": added. • Section 5.3 "Teardown": added. • Section 5.4 "Report ": added. • Section 6 "Examples": reorganized into Section 6.1 and Section 6.2. • Section 6.1 "ITWT example": added the commands for TWT intervals of 15 minutes and 20 minutes. • Section 6.2 "BTWT example": added. • Section 7 "TWT current consumption measurements": moved the content to Section 7.1. • Section 7.2 "BTWT current consumption values": added. • Section 8.2 "How to interpret TWT report from sniffer capture": added.
AN14111 v.1.0	21 June 2024	• Initial version

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Bluetooth — the Bluetooth wordmark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by NXP Semiconductors is under license.

Tables

Figures

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

For more information, please visit: https://www.nxp.com

Document feedback

Date of release: 3 September 2025
Document identifier: AN14111