Target Wake Time Manual

RA6W1/RA6W2 is a highly integrated, ultra-low-power Wi-Fi MCU that enables you to develop a complete Wi-Fi solution on a single chip. The purpose of this document is to provide detailed guidelines on configuring Target Wake Time (TWT) variables for RA6W1/RA6W2/RRQ61001/RRQ61051 project. It covers the configuration of TWT variables, the behavior of Wi-Fi software in various power-saving modes, and the decision-making process for selecting the appropriate Power-saving mode.

Contents

Figures

Tables

1. Terms and Definitions

AP	Access Point
DPM	Dynamic Power Management
PS	Power Save
SP	Service Period
STA	Station
TBTT	Target Beacon Transmission Time
TSF	Timing Synchronization Function
TWT	Target Wake Time

2. Introduction

Target Wake Time (TWT) is a new power saving feature introduced in the IEEE 802.11ax standard that significantly enhances battery efficiency for Wi-Fi-enabled devices. It allows devices to negotiate specific times with the Access Point (AP) to wake up and transmit or receive data, thereby reducing the need to constantly listen for traffic and minimizing power consumption.
Generally, TWT agreement is defined by, see Figure 1:

- TWT Service period (TWT-SP) – the Station (STA) wake duration.
- TWT wake interval – the period between two consecutive TWT SP.

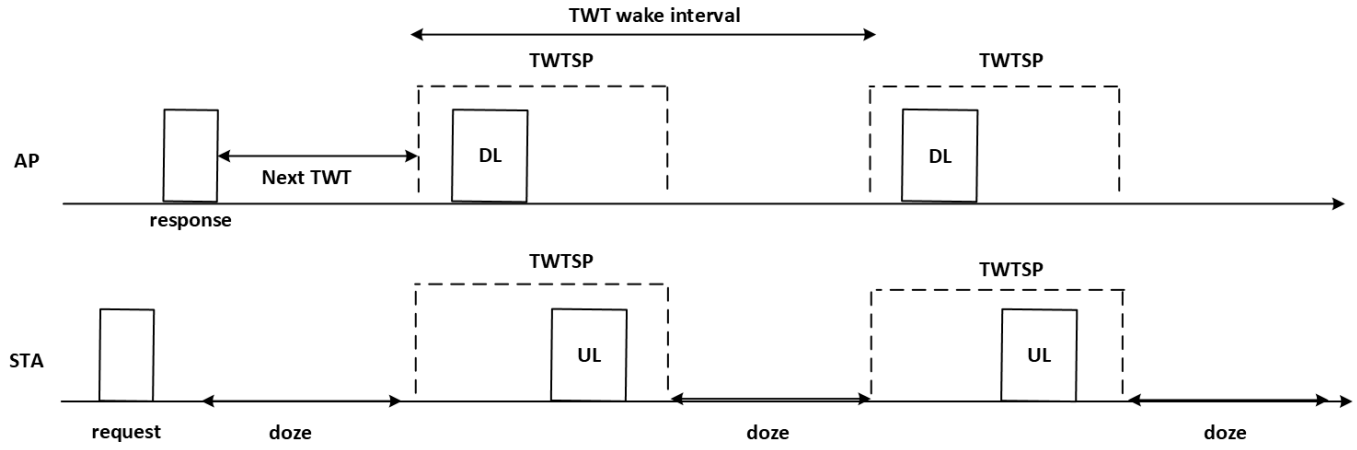

Figure 1. TWT basic flow

3. TWT Modes of Operation

RA6W1/RA6W2 supports two modes of operation for TWT:

- **Individual**: the client chooses when to wake up and when to sleep. The client is negotiating an agreement with the AP so all other clients should know when other clients wake up and send data.
- **Wake TBTT**: this type of TWT agreement specifies that the STA's wake time is aligned with the Target Beacon Transmission Time (TBTT) of the AP. The TWT interval starts at or relative to a beacon.

4. TWT Configuration

TWT configuration involves two key operations: TWT setup and TWT teardown.

4.1 TWT Setup

TWT setup is the process by which a STA and an AP agree on the parameters for scheduled wake-up times. The STA initiates the setup by sending a TWT setup frame, specifying preferred TWT parameters. Figure 2 shows the TWT setup frame, which is an action frame with category code "S1G" and the S1G action "TWT Setup". The AP can either accept, reject, or modify the STA's parameters, and then sends a response frame Figure 3 to the STA. A TWT agreement is established when the STA and AP agree on the TWT parameters.

```
▾ IEEE 802.11 Wireless Management
  ▾ Fixed parameters
      Category code: S1G (22)
      S1G Action: TWT Setup (6)
      Dialog token: 0xcb
    ▾ Tag: Target Wake Time
        Tag Number: Target Wake Time (216)
        Tag length: 15
      ▾ Control Field: 0x20, Negotiation type: Individual TWT, Reserved: 0x2
          .... ...0 = NDP Paging Indicator: Not Present
          .... ..0. = Responder PM Mode: AP is always awake
          .... 00.. = Negotiation type: Individual TWT (0x0)
          0010 .... = Reserved: 0x2
      ▾ Request Type: 0x2823
          .... .... .... ...1 = Requester: This STA is a TWT Requesting STA
          .... .... .... 001. = Setup Command: Suggest TWT (1)
          .... .... ...0 .... = Trigger: TWT SP does not include trigger frames
          .... .... ..1. .... = Implicit: TWT is implicit
          .... .... .0.. .... = Flow type: TWT is announced, the TWT Requesting STA will send trigger frames
          .... ..00 0... .... = Flow ID: 0
          .010 10.. .... .... = Wake Interval Exponent: 10
          0... .... .... .... = Protection: False
        Target Wake Time: 6397986648
        Nominal Minimum TWT Wake duration: 32
        TWT Wake Interval Mantissa: 14649
        TWT Channel: 0
```

Figure 2. TWT setup request frame

```
▾ IEEE 802.11 Wireless Management
  ▾ Fixed parameters
      Category code: S1G (22)
      S1G Action: TWT Setup (6)
      Dialog token: 0xcb
    ▾ Tag: Target Wake Time
        Tag Number: Target Wake Time (216)
        Tag length: 15
      ▾ Control Field: 0x20, Negotiation type: Individual TWT, Reserved: 0x2
          .... ...0 = NDP Paging Indicator: Not Present
          .... ..0. = Responder PM Mode: AP is always awake
          .... 00.. = Negotiation type: Individual TWT (0x0)
          0010 .... = Reserved: 0x2
      ▾ Request Type: 0x3428
          .... .... .... ...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. .... = Implicit: TWT is implicit
          .... .... .0.. .... = Flow type: TWT is announced, the TWT Requesting STA will send trigger frames
          .... ..00 0... .... = Flow ID: 0
          .011 01.. .... .... = Wake Interval Exponent: 13
          0... .... .... .... = Protection: False
        Target Wake Time: 6397984768
        Nominal Minimum TWT Wake duration: 32
        TWT Wake Interval Mantissa: 1832
        TWT Channel: 0
```

Figure 3. TWT setup response frame

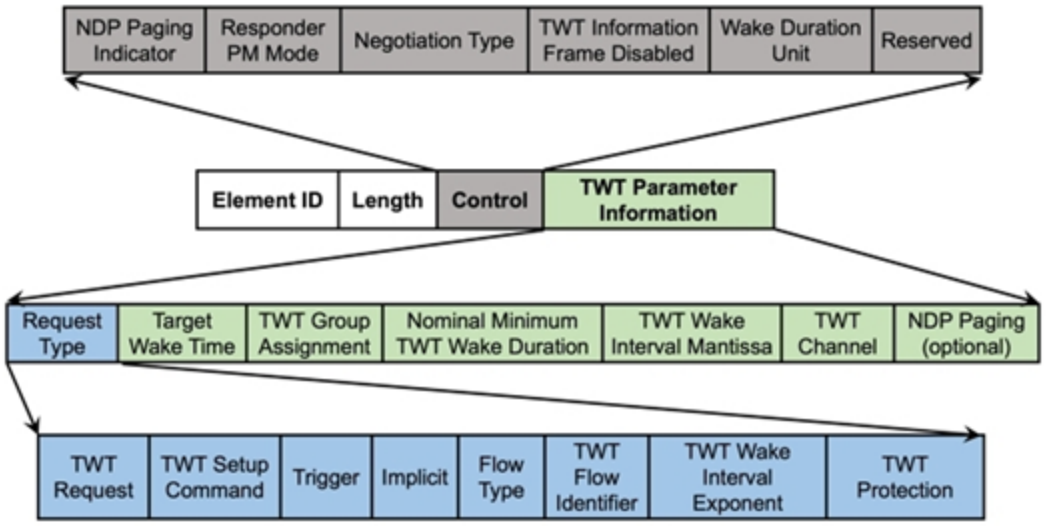

Figure 4. TWT element and subfields

4.1.1 TWT Parameters

This section describes the key configuration parameters involved in TWT setup negotiation, see Table 1.

Table 1. TWT parameters

Parameter	Description	Value range	Default value
Wake interval mantissa	Determine the mantissa part of the wake interval	0 – 65535	1850
Wake interval exponent	Determine the exponent part of the wake interval	0 – 31	13
Wake duration	Define the duration for which the device remains awake	0 – 255	32
Wake duration units	Specify the units for the wake duration	0 – Units of 256 µs 1 – Units of 1024 µs	1
Flow type	Indicate whether the transmission is announced or unannounced	0 – Announced mode 1 – Unannounced mode	1
TWT trigger	Specify whether the TWT is triggered	0 – Non-trigger 1 – Trigger based	0
Negotiation type	Define the negotiation type supported by device	0 – Individual 1 – Wake TBTT	0
Request Type	Define the type of TWT request	0 – Request TWT 1 – Suggest TWT 2 – Demand TWT	1 (Fixed value)
Implicit TWT	Indicate whether the TWT is implicit	0 – Explicit 1 – Implicit	1 (Fixed value)
Auto-setup	Enable auto-negotiation right after connection	0 - Disable 1 - Enable	0

4.2 TWT Teardown

TWT teardown terminates an active TWT agreement. The STA or AP can initiate the teardown by sending a TWT teardown frame, which is an action frame with category code SIG and SIG action TWT teardown, see Figure 5.

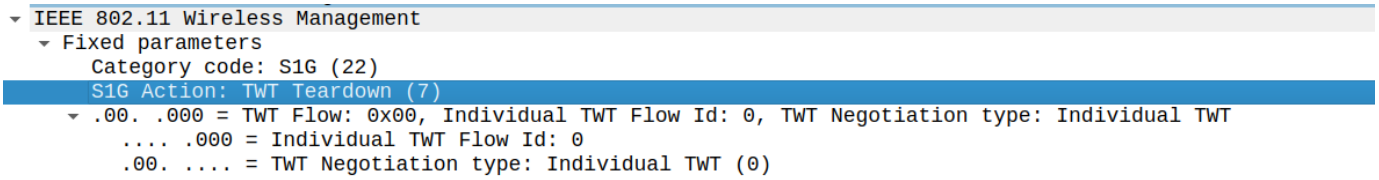

Figure 5. TWT teardown frame

4.3 TWT Configuration Using API

Table 2 explains the APIs for TWT configuration.

Table 2. TWT APIs

API name			
WIFIReturnCode_t WIFI_TwtSetup(struct twt_setup_req *req)			
Description	Send TWT setup request		
WIFIReturnCode_t WIFI_TwtSetup(struct twt_setup_req *req)			
Parameter	struct twt_setup_req *req	vif_idx	Virtual interface index. It should be 0 (Station).
		setup_type	Setup request type. It should be 1 (Suggest TWT).
		conf.flow_type	Flow type (0: Announced, 1: Unannounced).
		conf. wake_int_exp	Wake interval exponent.
		conf. wake_dur_unit	Unit of measurement of TWT minimum wake duration (0:256 µs, 1:1024 µs).
		conf. min_twt_wake_dur	Nominal minimum TWT wake duration (Unit is defined by wake_dur_unit).
		conf.trigger	Trigger disable/enable (0: Non-trigger, 1: Trigger enabled).
		conf.neg_type	Negotiation type (0: Individual, 1: Wake TBTT).
		conf. wake_int_mantissa	TWT wake interval mantissa.
		uint8_t auto_setup	To enable twt_auto_setup mode (0: parameters are stored, no negotiation until user triggers, 1: auto-negotiation right after connection)
Return	WIFIReturnCode_t	eWiFiSuccess or eWiFiFailure	
WIFIReturnCode_t WIFI_TwtTeardown(struct twt_teardown_req *req)			
Description	Send TWT Teardown request		

API name				
Parameter	struct twt_ teardown_req *req	neg_type	TWT negotiation type.	
		all_twt	All TWT.	
		id	TWT flow ID.	
		vif_idx	VIF index.	
void twt_event_callback(WIFIEvent_t * pxEvent)				
Description	Callback handler for TWT events. It is invoked whenever a TWT related event occurs.			
Parameter	WIFIEvent_t * pxEvent	xEventType	Wi-Fi event type.	
		xInfo.xTWT.xEvent	WIFI_TWT_EVENT_SESSION_SUCCESS – TWT session established successfully. WIFI_TWT_EVENT_AP_REJECTED – AP rejected the TWT request. WIFI_TWT_EVENT_AP_RESPONSE_PARAM_NOT_MATCHED – AP response parameters did not match the request. WIFI_TWT_EVENT_TEARDOWN_SUCCESS – TWT session teardown successful. WIFI_TWT_EVENT_AP_TEARDOWN_SUCCESS – AP initiated teardown of the TWT session. WIFI_TWT_EVENT_MAX_RETRIES_REACHED – Maximum retries reached for TWT setup. WIFI_TWT_EVENT_AP_NO_SUPPORT – AP does not support TWT. WIFI_TWT_EVENT_TWT_REQ_ABORT – TWT setup request aborted default – Unknown or unsupported TWT sub-event.	
WIFIReturnCode_t WIFI_RegisterEvent(WIFIEventType_t xEventType, WIFIEventHandler_t xHandler)				
Description	Registers a callback handler for a specific Wi-Fi event. The registered handler is invoked whenever the specified event occurs.			
Parameter	WIFIEventType_t xEventType	xEventType	Wi-Fi event type.	
	WIFIEventHandler_t xHandler		Function pointer to the callback that handles the event.	
Return	WIFIReturnCode_t	eWiFiSuccess or eWiFiFailure		

4.4 TWT Configuration Using AT Command

The AT command for TWT is the following:

```
AT+PMGRFORCE=<sleep_mode>, <twt_action>, <neg_type/wake_int_mantissa>, <wake_int_exp>, <min_twt_
wake_dur>, <flow_type>, <trigger>, <neg_type>, <auto_setup>
```

Table 3 shows arguments used in AT command.

Table 3. Arguments for TWT AT command

Argument	Description	
<sleep_mode>	It should be 5 for TWT.	
<twt_action>	TWT action (0 – TWT teardown, 1 – TWT setup)	
	0 (TWT teardown)	
	<neg_type>	0: Individual 1: Wake TBTT
	1 (TWT setup)	
	<wake_int_mantissa>	TWT wake interval mantissa
	<wake_int_exp>	Wake interval exponent
	<min_twt_wake_dur>	Nominal minimum TWT wake duration for example, 3
	<flow_type>	0 = Announced 1 = Unannounced
	<trigger>	0 = Disabled 1 = Enabled
	<neg_type>	0: Individual 1: Wake TBTT
	<auto_setup>	0: Disable 1: Enable

The device responds with OK or Error Code:

- 0x500002: Insufficient arguments
- 0x500004: Wrong arguments

The example for TWT setup:

```
AT+PMGRFORCE=5,1,8000,10,3,0,0,0,0
OK
```

Where:
Sleep_mode = 5 (TWT). Action = 1 (TWT setup). Mantissa = 8000, Exponent = 10. TWT interval = 8000 x 2^10 ~= 8 s. Minimum wake duration for each service period = 3 TU. flow_type = 0 [0 = announced. 1 = unannounced]. RA6W1 used to transmit PS poll when receiving beacon with TIM bit, in Wake TBTT negotiation type it should be announced. Trigger en = 0. Negotiation type = 0 (Individual), Auto-setup = 0 (disable).

The example for TWT teardown:

```
AT+PMGRFORCE=5,0,0
OK
```

Where:
Sleep_mode = 5 (TWT). Action = 1 (TWT teardown). Negotiation type = 0 (Individual).

5. Wake TBTT TWT

Wake TBTT TWT is a type of negotiated TWT in which the TWT Service Period (SP) is explicitly aligned with the Access Point's (AP's) Target Beacon Transmission Time (TBTT). That means the SP starts at the beacon timestamp (TSF) allowing the STA to receive both the beacon and data during the same wake-up window.
This alignment simplifies STA-side scheduling and reduces power consumption, as the STA can stay synchronized with the AP without requiring extra wake-ups solely to receive beacons.
In Wake TBTT TWT mode, the TWT parameters are configured with the following constraints:

- Wake interval exponent = 10
- Flow type = 0 (Announced)
- Trigger = 0
- Wake interval mantissa: Adjusted to align with the AP's DTIM period.

```
if ((mant % dtim_p) > 0)
{
    mant += dtim_p - (mant % dtim_p);
    req->conf.wake_int_mantissa = mant;
}
```

Where:
mant = req->conf.wake_int_mantissa (configured mantissa) and dtim_p = DTIM period of AP

6. Individual TWT

Individual TWT is a fully STA initiated and negotiated TWT agreement, where the SP timing is arbitrary and does not have to align with TBTT. This gives the STA full control over wake intervals, start times, and durations, enabling fine-tuned power and performance optimization.
However, the SP may occur asynchronously with beacon transmissions, the STA often needs to wake up separately to receive beacons to maintain synchronization and retrieve control information such as TIM. This results in additional power consumption compared to TBTT aligned TWT.

7. TWT Announced - PS Poll TX

When TWT announced mode is set, STA should transmit frame to AP indication that it is awake. RA6W1/RA6W2 sends a PS-POLL frame in Announced mode. Figure 6 shows a TWT individual non trigger announced more.

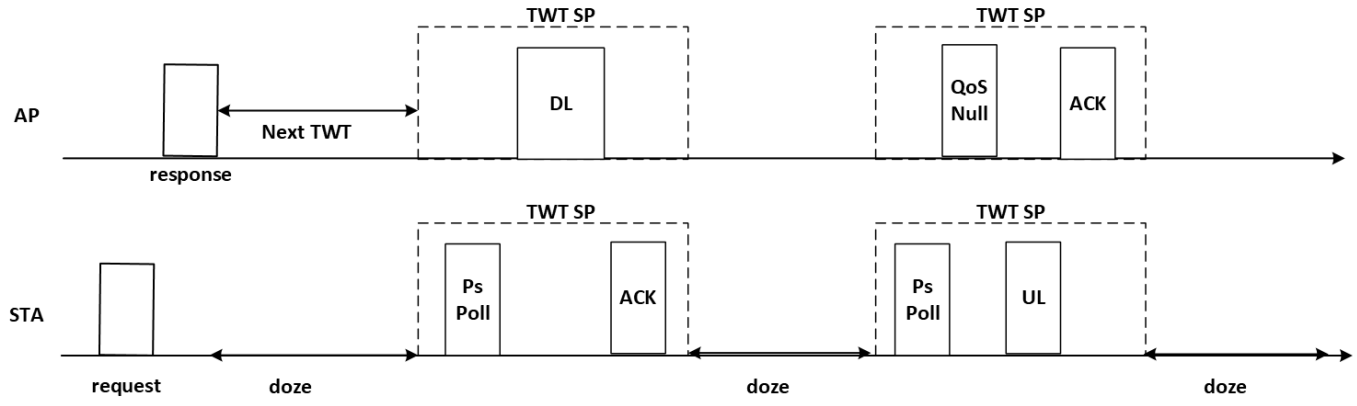

Figure 6. TWT individual announce non trigger

8. Wake-up Interval Calculations

The wake-up interval is determined by the TWT exponent and mantissa.
The formula is: TWT Wake-up Interval = Mantissa * 2Exponent
Example:

- Mantissa = 2000
- Exponent = 10

TWT Wake-up Interval = 2000 * 2^{10} µs = 2 s.

9. Transition Between Legacy PS and TWT Modes

The software should support both legacy PS (DTIM) and TWT features and should be able to switch between both modes dynamically.
By default, the device operates in Legacy DTIM Power Save (PS) mode. It listens to DTIM beacon based on DPM configuration and schedule other tasks (PS-Poll transmission) according to Beacon TIM element information.
When you issue a TWT setup command, the requested TWT parameters are stored in the device's internal memory. At this point:

- TWT session is initiated by sending a TWT setup suggest frame.
- If the AP responds with "Accept", the TWT session is successfully established.
- If the AP responds with "Reject" or other non-accept options, the session should not be opened.
- If the AP does not respond and the `auto_setup` flag is enabled, the RA6W1/RA6W2 tries again by sending another TWT suggest frame.
  - After two consecutive failures, the `auto_setup` should be cleared, and no further TWT setup attempts are made. It chooses legacy PS.
- If a reconnection occurs and the `auto_setup` flag is still set, the RA6W1/RA6W2 automatically attempts to reestablish a TWT session using the previously saved parameters.
- When you issue a TWT teardown command:
  - RA6W1/RA6W2 sends a TWT Teardown frame to the AP to terminate the active TWT session.
  - The `auto_setup` flag is cleared, preventing any further automatic attempts to establish TWT sessions.

Appendix A TWT Results

For SDK version V8.0.2.0.5:

- Operating temperature: 20-25 °C
- Operating platform: Standalone
- Xiaomi AX6000 2.4G_Ch6; 5G_ch36
- Wake Duration Unit: 256 µs
- TWT individual (minimum wake duration = 8192 µs), Unannounced mode

Table 4. TWT individual power measurement

Test item	Condition	Average current consumption for 5 minutes	
		2.4 G	5 G
Connection with TWT in IDLE	TWT interval 15 s without WLAN KA	48.93 µA	51.37 µA
	TWT interval 30 s without WLAN KA	31.031 µA	31.5 µA
	TWT interval 15 s with WLAN KA 30 s	55.01 µA	56.45 µA
	TWT interval 30 s with WLAN KA 60 s	37.03 µA	33.43 µA
Connection with TWT and TCP open session	TWT interval 15 s, WLAN KA 30 s, TCP KA 120 s	59.02 µA	57.09 µA
	TWT interval 30 s, WLAN KA 30 s, TCP KA 120 s	42.72 µA	36.64 µA

10. Revision History

Revision	Date	Description
1.04	May 7, 2026	Added new TWT test results, corrections to TWT configuration parameters. Made document generic to the RA6W1/RA6W2/RRQ61001/RRQ61051 projects.
1.03	Mar 6, 2026	Changed the document title.
1.02	Nov 28, 2025	Added WIFI_TWT_EVENT_TWT_REQ_ABORT to TWT event types.
1.01	Aug 31, 2025	Added twt_event_callback and WIFI_RegisterEvent APIs to the TWT APIs table.
1.00	May 23 2025	First release.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

CorporateHeadquarters

TOYOSU FORESIA, 3-2-24 Toyosu
Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/