DA145xx/DA146xx Diagnostic Registers for RF Debugging

This document describes how to put three digital diagnostic signals (RX enable, TX enable, and event_in_progress) to dedicated GPIOs to help debugging an application software.

Contents

Figures

Tables

1. Terms and Definitions

Bluetooth LE	Bluetooth® Low Energy
GPIO	General Purpose Input/Output
PID	Peripheral IO ID

2. Introduction

Renesas' Bluetooth® devices DA145xx/DA146xxcan put internal status signals to GPIOs. This can be done by programming Bluetooth diagnosis registers and setting the wanted port's PID. The procedures and limitations to achieve this are described in this document.

3. Diagnostic Registers

There are three relevant diagnostic registers in DA14xxx family of devices. BLE_DIAGCNTL_REG, BLE_DIAGCNTL2_REG, and BLE_DIAGCNTL3_REG. With these three registers internal status signals can be put to GPIOs of a DA14xxx.

3.1 BLE_DIAGCNTL_REG

In BLE_DIAGCNTL_REG (see Table 1), the output content of diagnostic port 0 to 3 are selected by programming a 6-bit identifier into the DIAGx bitfield.
Additionally, this register enables/disables DIAG ports by setting the DIAGx_EN bit.

Table 1. BLE_DIAGCNTL_REG

Bit	Mode	Symbol/Description	Reset
31	R/W	DIAG3_EN 0: Disable diagnostic port 3 output. All outputs are set to 0x0. 1: Enable diagnostic port 3 output.	0x0
29:24	R/W	DIAG3 Only relevant when DIAG3_EN = 1. Selection of the outputs that must be driven to the diagnostic port BLE_DIAG3.	0x0
23	R/W	DIAG2_EN 0: Disable diagnostic port 2 output. All outputs are set to 0x0. 1: Enable diagnostic port 2 output.	0x0
21:16	R/W	DIAG2 Only relevant when DIAG2_EN = 1. Selection of the outputs that must be driven to the diagnostic port BLE_DIAG2.	0x0
15	R/W	DIAG1_EN 0: Disable diagnostic port 1 output. All outputs are set to 0x0. 1: Enable diagnostic port 1 output.	0x0
13:8	R/W	DIAG1 Only relevant when DIAG1_EN = 1. Selection of the outputs that must be driven to the diagnostic port BLE_DIAG1.	0x0
7	R/W	DIAG0_EN 0: Disable diagnostic port 0 output. All outputs are set to 0x0. 1: Enable diagnostic port 0 output.	0x0
5:0	R/W	DIAG0 Only relevant when DIAG0_EN = 1. Selection of the outputs that must be driven to the diagnostic port BLE_DIAG0.	0x0

3.2 BLE_DIAGCNTL2_REG

This register (see Table 2) has the same function as BLE_DIAGCNTL_REG, but controls diagnostic port 4 to 7.

Table 2. BLE_DIAGCNTL2_REG

Bit	Mode	Symbol/Description	Reset
31	R/W	DIAG7_EN 0: Disable diagnostic port 7 output. All outputs are set to 0x0.	0x0

Bit	Mode	Symbol/Description	Reset
		1: Enable diagnostic port 7 output.	
30	R	-	0x0
		Reserved	
29:24	R/W	DIAG7	0x0
		Only relevant when DIAG7_EN = 1.	
		Selection of the outputs that must be driven to the diagnostic port BLE_DIAG7.	
23	R/W	DIAG6_EN	0x0
		0: Disable diagnostic port 6 output. All outputs are set to 0x0.	
		1: Enable diagnostic port 6 output.	
22	R	-	0x0
		Reserved	
21:16	R/W	DIAG6	0x0
		Only relevant when DIAG6_EN = 1.	
		Selection of the outputs that must be driven to the diagnostic port BLE_DIAG6.	
15	R/W	DIAG5_EN	0x0
		0: Disable diagnostic port 5 output. All outputs are set to 0x0.	
		1: Enable diagnostic port 5 output.	
14	R	-	0x0
		Reserved	
13:8	R/W	DIAG5	0x0
		Only relevant when DIAG5_EN = 1.	
		Selection of the outputs that must be driven to the diagnostic port BLE_DIAG5.	
7	R/W	DIAG4_EN	0x0
		0: Disable diagnostic port 4 output. All outputs are set to 0x0.	
		1: Enable diagnostic port 4 output.	
6	R	-	0x0
		Reserved	
5:0	R/W	DIAG4	0x0
		Only relevant when DIAG4_EN = 1.	
		Selection of the outputs that must be driven to the diagnostic port BLE_DIAG4.	

3.3 BLE_DIAGCNTL3_REG

In BLE_DIAGCNTL3_REG (see Table 3), the register decides which bit of diagnostic port 0 to 7 is forwarded to the Bluetooth Diagnostic Port, which can be connected to a GPIO of the Bluetooth device.

NOTE
A certain DIAG port is always connected to a fixed bit of the Bluetooth diagnostic port. For example, if you want to see a signal on P0_0, it must be available on DIAG port 0.

Additionally, all diagnostic bits can be inverted individually by setting the DIAGx_INV bit.

Table 3. BLE_DIAGCNTL3_REG

Bit	Mode	Symbol/Description	Reset
31	R/W	DIAG7_INV	0x0
		If set, then the specific diagnostic bit is inverted.	
30:28	R/W	DIAG7_BIT	0x0
		Selects which bit from the DIAG7 word is forwarded to bit 7 of the Bluetooth LE Diagnostic Port.	
27	R/W	DIAG6_INV	0x0
		If set, then the specific diagnostic bit is inverted.	
26:24	R/W	DIAG6_BIT	0x0

Bit	Mode	Symbol/Description	Reset
		Select which bit from the DIAG6 word is forwarded to bit 6 of the Bluetooth LE Diagnostic Port.	
23	R/W	DIAG5_INV If set, then the specific diagnostic bit is inverted.	0x0
22:20	R/W	DIAG5_BIT Select which bit from the DIAG5 word is forwarded to bit 5 of the Bluetooth LE Diagnostic Port.	0x0
19	R/W	DIAG4_INV If set, then the specific diagnostic bit is inverted.	0x0
18:16	R/W	DIAG4_BIT Select which bit from the DIAG4 word is forwarded to bit 4 of the Bluetooth LE Diagnostic Port.	0x0
15	R/W	DIAG3_INV If set, then the specific diagnostic bit is inverted.	0x0
14:12	R/W	DIAG3_BIT Select which bit from the DIAG3 word is forwarded to bit 3 of the Bluetooth LE Diagnostic Port.	0x0
11	R/W	DIAG2_INV If set, then the specific diagnostic bit is inverted.	0x0
10:8	R/W	DIAG2_BIT Select which bit from the DIAG2 word is forwarded to bit 2 of the Bluetooth LE Diagnostic Port.	0x0
7	R/W	DIAG1_INV If set, then the specific diagnostic bit is inverted.	0x0
6:4	R/W	DIAG1_BIT Select which bit from the DIAG1 word is forwarded to bit 1 of the Bluetooth LE Diagnostic Port.	0x0
3	R/W	DIAG0_INV If set, then the specific diagnostic bit is inverted.	0x0
2:0	R/W	DIAG0_BIT Select which bit from the DIAG0 word is forwarded to bit 0 of the Bluetooth LE Diagnostic Port.	0x0

4. 6 Bit Identifier

The 6 Bit Identifier references to a set of eight internal diagnostic signals.

The focus of this document is on three signals (TX enable, RX enable, and event_in_progress) and all can be found under one identifier (see Figure 1), you only need to set the relevant DIAGx bit field in BLE_DIAGCNTL_REG or BLE_DIAGCNTL2_REG to 0x03.

DIAGx[5:0]->	0x03 (3)
DiagOut[0]	radcntl_txen
DiagOut[1]	radcntl_rxen
DiagOut[2]	sync_window
DiagOut[3]	sync_found_pulse
DiagOut[4]	event_in_process
DiagOut[5]	ble_event_irq
DiagOut[6]	ble_rx_irq
DiagOut[7]	ble_error_irq

Figure 1. DIAG identifier 0x03

5. Example

If you want to put enable TX, enable RX, and event_in_progress to P01, P02, and P06 of a DA145xx/DA146xx device, complete the following steps.

All wanted signals can be found in DIAG identifier 0x03 (Figure 1). You want to see three different signals so 3 DIAGx bitfields out of BLE_DIAGCNTL_REG and BLE_DIAGCNTL2_REG must be set to this DIAG identifier (0x03). If you want to see a signal on P0_0, it is important to use DIAG0. If the signal must appear on P0_1, use DIAG1.

To see these signals on P0_1, P0_2 and P0_6, you must use DIAG1, DIAG2 and DIAG6.

```
SetBits32(BLE_DIAGCNTL_REG, DIAG1, 0x03);
SetBits32(BLE_DIAGCNTL_REG, DIAG2, 0x03);
SetBits32(BLE_DIAGCNTL2_REG, DIAG6, 0x03);
```

These three DIAG ports must be enabled as well.

```
SetBits32(BLE_DIAGCNTL_REG, DIAG1_EN, 1);
SetBits32(BLE_DIAGCNTL_REG, DIAG2_EN, 1);
SetBits32(BLE_DIAGCNTL2_REG, DIAG6_EN, 1);
```

With BLE_DIAGCNTL3_REG, you can now decide which signal should be available on which bit of the Bluetooth diagnostic port.

```
SetBits32(BLE_DIAGCNTL3_REG, DIAG1_BIT, 0);          // radcntl_txen
SetBits32(BLE_DIAGCNTL3_REG, DIAG2_BIT, 1);          // radcntl_rxen
SetBits32(BLE_DIAGCNTL3_REG, DIAG6_BIT, 4);          // event_in_progress
```

If you want to invert the event_in_progress signal, you must set the inverting bit of DIAG6.

```
SetBits32(BLE_DIAGCNTL3_REG, DIAG6_INV, 1);          // invert event_in_progress
```

In the last step, you must connect the GPIO to the diagnostic signal and set it to output.

```
SetBits16(P01_MODE_REG, PID , 18);
SetBits16(P01_MODE_REG, PUPD,  3);                    //TXEN = P0_1

SetBits16(P02_MODE_REG, PID , 18);
SetBits16(P02_MODE_REG, PUPD,  3);                    //RXEN = P0_2

SetBits16(P06_MODE_REG, PID , 18);
SetBits16(P06_MODE_REG, PUPD,  3);                    // event_in_progress = P0_6
```

The result can be seen in Figure 2. The light blue graph is showing the power consumption of a DA14531 in buck mode. When TX is enabled (grey graph), the expected TX current consumption can be measured. When RX is enabled (dark blue graph), the supply current is slightly lower. The event in progress waveform (red graph) is enveloping the RX and TX events.

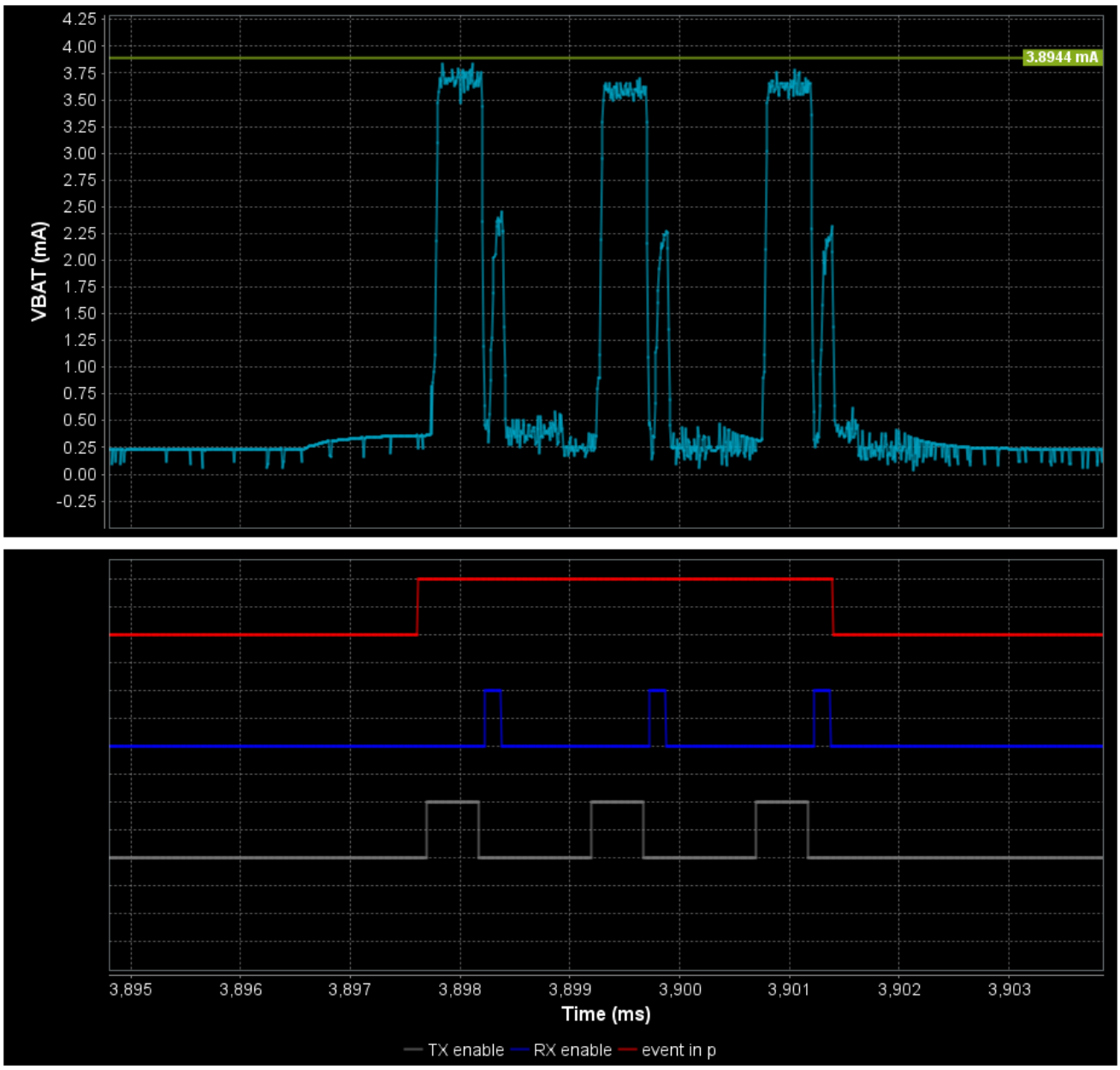

Figure 2. Waveform example

6. Important Notes

- P0_0 to P0_7 are connected to DIAG0 to DIAG7
- P0[3:0] are also mapped to P0[11:8])

7. Revision History

Revision	Date	Description
01.00	Nov 8, 2024	Initial release

Status Definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.

ROHS COMPLIANCE

Renesas Electronics' suppliers certify that its products are in compliance with the requirements of Directive 2011/65/EU of the European Parliament on the restriction of the use of certain hazardous substances in electrical and electronic equipment. RoHS certificates from our suppliers are available on request.

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.1.1 Jan 2024)

Corporate Headquarters

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/