RH850 Family

SENT Emulation Application Note

Introduction

This application note introduces a way of emulating a SENT transmitter (sensor side) to build a counterpart to the RSENT receiver modules which are equipped on most Renesas RH850 MCUs.

The SENT protocol (Single Edge Nibble Transmission) is intended to report sensor information to Engine Control Units (ECU) and should replace the lower resolution methods of 10-bit A/D's and PWM and as a simpler low-cost alternative to CAN or LIN. It is a unidirectional communications scheme from sensor device to a controller (This does not include an optional coordination signal from the controller) *.

The Renesas MCUs incorporating RSENT modules provide an easy way of receiving and preprocessing the sensor data, adds functions like timestamp for every received message and gives you numerous error detection functions. As the transmission of the sensor signals occurs independently, the RSENT module makes it easy to receive sensor data without CPU interaction.

This App Note should give the customer the possibility to easily start the development of RSENT based software, reducing the time to market without the need for a PCB design or a SENT sensor.

Table 1. Reference Documentation

Description	Link
SAE: SENT - Single Edge Nibble Single Edge Nibble Transmission for Automotive Applications J2716	https://www.sae.org/standards/content/j2716_201604/
RH850/F1KM-S1 Hardware User manual	https://www.renesas.com/us/en/document/mah/rh850f1km-rh850f1kh-users-manual-hardware?r=1170176
RH850/F1KM-S1 SENT Interface Application Note	https://www.renesas.com/us/en/document/apn/rh850f1km-s1-single-edge-nibble-transmission-sent-interface-application-note?r=1170176

Target Device

This application was written on a Renesas RH850/F1KM-S1 but could easily be ported to MCUs coming with a Timer Array Unit D (TAUD) like the RH850/U2x, RH850/P1x, RH850/C1x and other devices of the RH850/F1x family.

*Source: SENT SAE standard J2716

Contents

1. SENT Standard Basics

The sensor signal is transmitted as a series of pulses with data encoded as falling to falling edge periods. It is a one wire interface with minimal external components necessary.

There are two different communication channels in one SENT Interface:

Fast channel communication (primary)
— 24-bit data field
— Incorporates 2 bit (in COM nibble) for slow channel communication

Slow channel communication (secondary)
— Two different formats available:
- Short Serial Message (8-bit Data field)
- Enhanced Serial Message (with 12-bit or 16-bit Data field)

Note: This application note only gives a brief overview of the SAE J2716. For further details refer to the specific standards in their latest versions (Table 1. Reference Documentation).

1.1 General Requirements

The communication is initiated only by the sending device. The receiver performs no actions to start the SENT communication and does not send any synchronization signal to the transmitting device. The nominal communication clock is measured in "clock ticks". Both communication partners should be processing the SENT protocol with the same clock tick length.

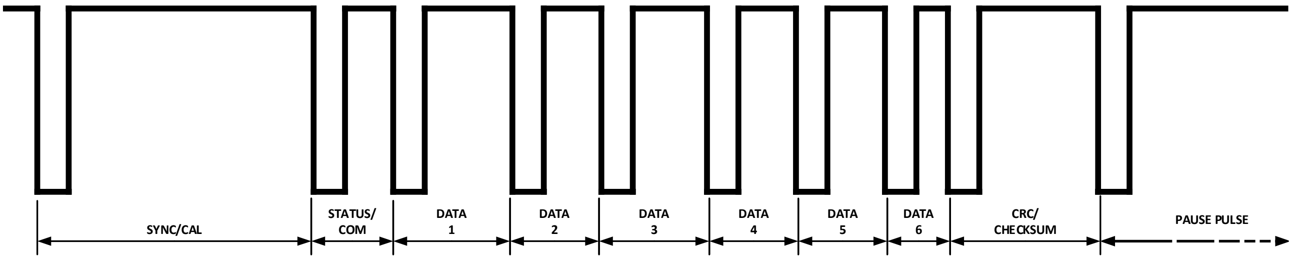
The SAE J2716 JAN2010 standard defines the following requirements:

- Clock tick time from 3µs to 90µs.
- Fixed message frame order for all transmitters
- ±20% maximum clock variation for the transmitter
- The transmission time depends on:
  — Actual data values which are send to the receiver.
  — The transmitter clock variations
- The maximum transmission time for the longest data messages and the maximum transmitter clock variation is less than 1ms at 3µs bit time and 6 data nibbles.

1.2 SENT Message Structure for Fast Channel

A valid SENT fast channel message contains the following elements:

- Synchronization/Calibration frame (56 clock ticks)
- Status & Communication frame (12-27 clock ticks)
- Data frames 1…6 (12-27 clock ticks per data frame)
- CRC/Checksum frame (12-27 clock ticks)
- Pause pulse (optional)



Note: The pause pulse is optional.

Figure 1. SENT fast channel message structure.

1.2.1 Synchronization/Calibration Pulse Details

The Synchronization/Calibration pulse has a defined tick length of 56 SENT clock ticks. The receiver measures the actual period and adjusts his clock speed if it is necessary. This pulse type is also used to detect the end of the previously sent message and to perform message diagnostics.

The RSENT module supports automatic calibration pulse diagnostics in different modes.

1.2.2 Status and Communication Nibble Details

These nibbles contain miscellaneous information (e.g., error codes, part numbers or manufacture codes) implemented in the slow channel. One Status and Communication nibble contains 2-bits for the slow channel message. Therefore, a successful sequence of 16 (or 18 depending on the channel format) fast channel messages is needed for the reception of one complete slow channel message. The slow channel has a separate CRC error detection for the contained information.

Note: The content of the Status and Communication nibble may be called “Serial Data” in this application note and some other literary sources.

1.2.3 Data Nibble Details

The data information of the fast channel messages is decoded in the data frames from 1 to 6. The minimum tick length of a data nibble is 12 ticks, the maximum value is 27 ticks. The nibble data is decoded according to the following table and composed together.

Table 2. Data nibble encoding scheme.

Nibble Period (Number of Clock Ticks)	Nibble Value (Binary)
12	$(0000)_B$
13	$(0001)_B$
14	$(0010)_B$
15	$(0011)_B$
16	$(0100)_B$
17	$(0101)_B$
18	$(0110)_B$
19	$(0111)_B$
20	$(1000)_B$
21	$(1001)_B$
22	$(1010)_B$
23	$(1011)_B$
24	$(1100)_B$
25	$(1101)_B$
26	$(1110)_B$
27	$(1111)_B$

1.2.4 CRC/Checksum Nibble Details

This CRC-check detects errors in the fast channel data only. The 4-bit CRC value is measured in the same way as the data nibbles. The status and communication nibble are not included in this CRC-check.

1.3 SENT Message Structure for Slow Channel

Following, the two different frame formats are described.

1.3.1 Short Serial Message Format

This format consists of the following elements:

- Message ID
- 8-bit Data
- 4-bit CRC

The serial data bit #3 in a fast channel message indicates the first bit of the data signal with the only “1” in a serial message. The actual information (message Id, Data and CRC) is transmitted only in serial data bit #2.

Please find below the SENT slow channel pattern for a short serial message:

SENT SLOW CHANNEL PATTERN FOR SHORT SERIAL MESSAGE																
RECEIVED FAST CH. MESSAGE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
SERIAL DATA BIT #3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SERIAL DATA BIT #2	Message ID				8-bit Data								CRC			

Figure 2 Short Serial Message Format

1.3.2 Enhanced Serial Message Format

This format consists of the following elements:

- 8-bit Message ID field
- 12-bit Data field
- 6-bit CRC

Note: There is a variant of the ESMF with 4-bit message ID and 16-bit of data, but for this app note we only use the 12-bit data variant.

The beginning of an enhanced serial message is indicated by 6 ones followed by a zero in serial data bit #3.
This pattern cannot occur in the following message ID bits and therefore is unique.
Serial data bit #2 consists of the CRC followed by 12-bit of data.

Please find below the SENT slow channel pattern for an enhanced serial message:

<table>
<tr><th colspan="19">SENT SLOW CHANNEL PATTERN FOR ENHANCED SERIAL MESSAGE</th></tr>
<tr><td>RECEIVED SENT MESSAGE</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td></tr>
<tr><td>SERIAL DATA BIT (#3)</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>C*</td><td colspan="4">MESSAGE ID [7:4]</td><td>0</td><td colspan="4">MESSAGE ID [3:0]</td><td>0</td></tr>
<tr><td>SERIAL DATA BIT (#2)</td><td colspan="6">6-BIT CRC</td><td colspan="12">12-BIT DATA</td></tr>
</table>

Note: C = Configuration bit (12-bit or 16-bit of data)

Figure 3. Enhanced Serial Message Format

2. Used Software, Hardware and Debugging Tools

The following components were used to implement the SENT Emulation application.

- **Software:**
- Green Hills Software MULTI® ver. 7.1.6
- RH850/F1KM-S1 SENT Emulation Software Sample

- **Debugging Tools:**
- Renesas E2 Debugger

- **Hardware components:**
- Renesas RH850/F1KM-S1 Starter Kit V3 (**Y-BLDC-SK-RH850F1KM-S1-V2**).
- Used microcontroller:
  - RH850/F1KM-S1 100pin R7F7016844-AFP-C (ES 1.0)

The output of the TAUD0 Channel 1 is routed to the Port **P10_0** and externally connected to the input of the RSENT1 RX pin which is routed to the Port pin **P9_0**.

2.1 RH850 Series Build-In SENT Module (RSENT)

The built-in hardware module RSENT provides the full functionality according to SAE J2716 (APR2016) Standard and offers the user an easy to handle SENT receiver for quick implementations without the need to use other hardware and software components (e.g. capture and compare timer units).

The following features are provided:

- Triple speed expansion Tick Time with clock cycles from 1µs up to 90µs
- Variable data transmission rate:
  - 24.7kbps to 64.9 kbps: 3 clock rate 8 nibble data
  - 74.1 kbps to 194.8 kbps: 1 clock rate 8 nibble data
- Unidirectional communication between sensor and MCU
- Bidirectional communication possible in SPC mode
- Timestamp function with Master or Slave mode

2.1.1 Block Diagram of RSENT Module

The following figure shows a block diagram of the RSENT module.

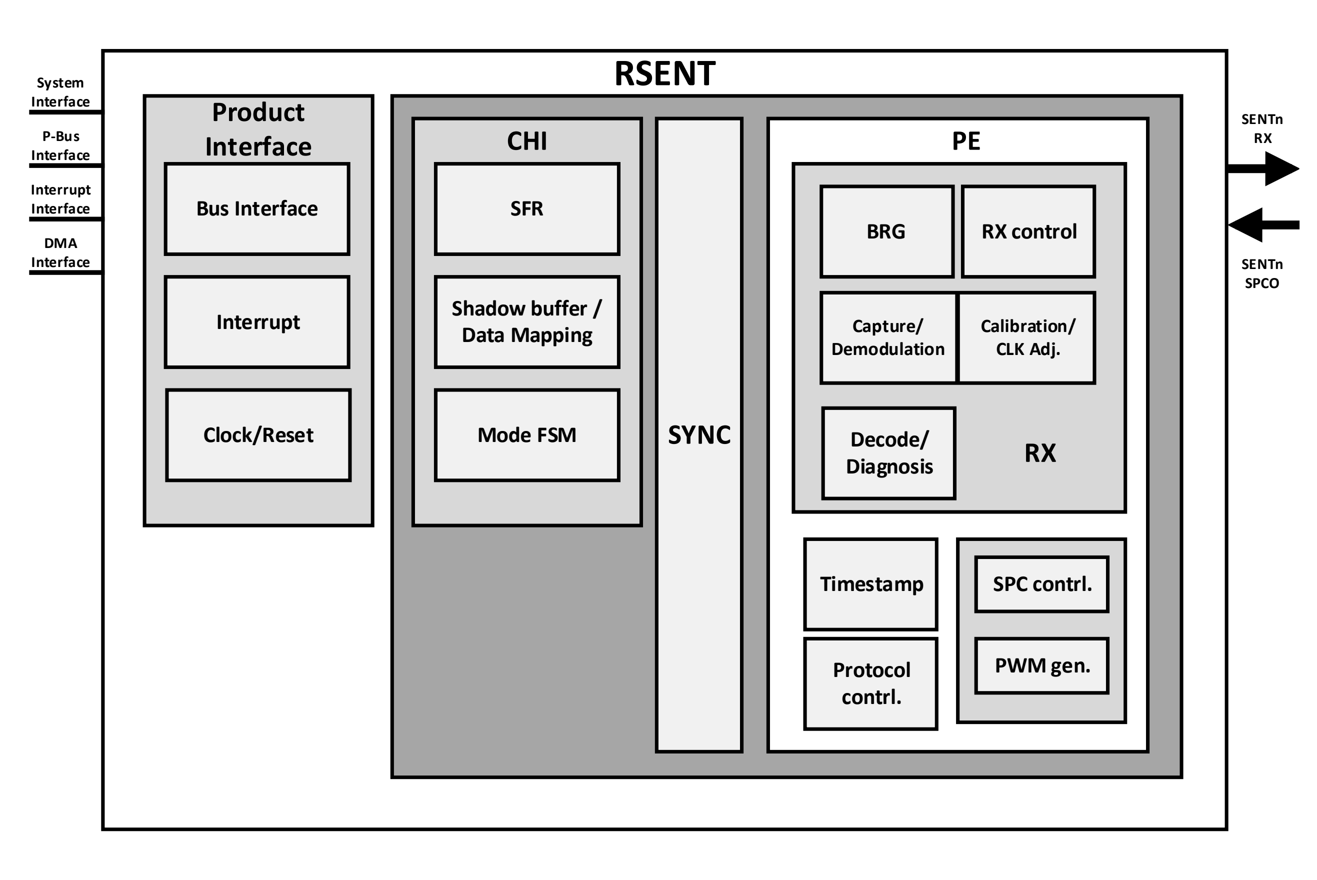


Figure 4. Block diagram RSENT module

As the focus of this application note is not the Renesas RSENT module, it is not described in any more detail here. If you want to have more details about it please refer to the “RH850/F1KM-S1 Harware User Manual” and the “RH850 SENT Interface Application Note” in Table 1. Reference Documentation.

2.2 TAUD

The TAUD has the following functions:

- 16 channels
- 16-bit counter and 16-bit data register per channel
- Independent channel operation
- Synchronous channel operation (master and slave operation)
- Generation of different types of output signal
- Real-time output
- Counter can be triggered by external signal
- Interrupt generation

The Timer Array Unit D is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler block for count clock generation and 16 channels, each equipped with a 16-bit counter TAUDnCNTm and a 16-bit data register TAUDnCDRm to hold the start or compare value of the counter.

It also contains several control and status registers.

Please find a block diagram below:

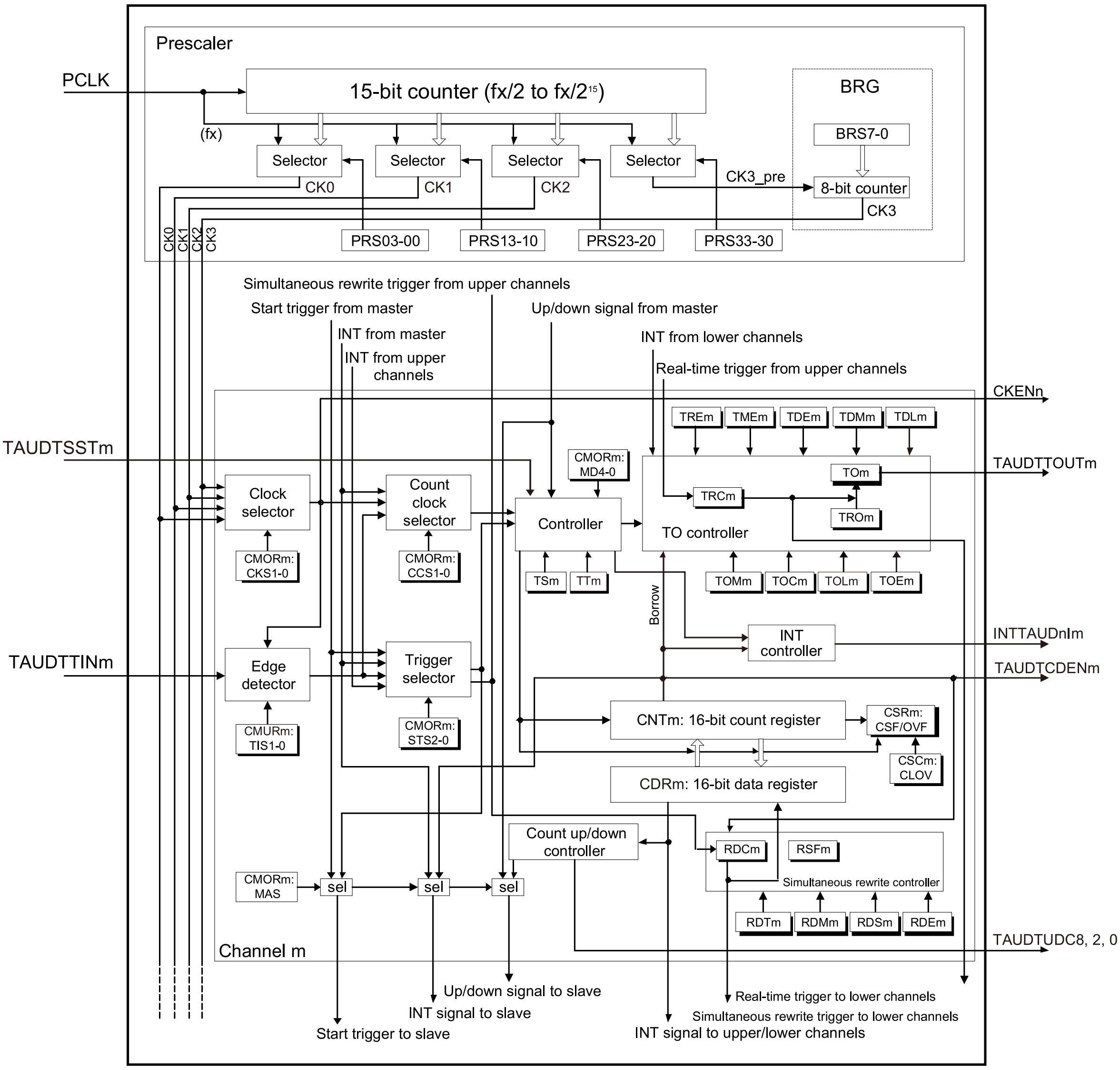


Figure 5 TAUD Block Diagram

2.2.1 Use Case

For this application note the “PWM Output Function” of the TAUD is used. This function generates a PWM output by using a master and a slave channel. It enables the pulse cycle (frequency) and the duty cycle of the TAUDTTOUTm to be set. The pulse cycle is set in the master channel. The duty cycle is set in the slave channel.

Please find a block diagram of the PWM Function below:

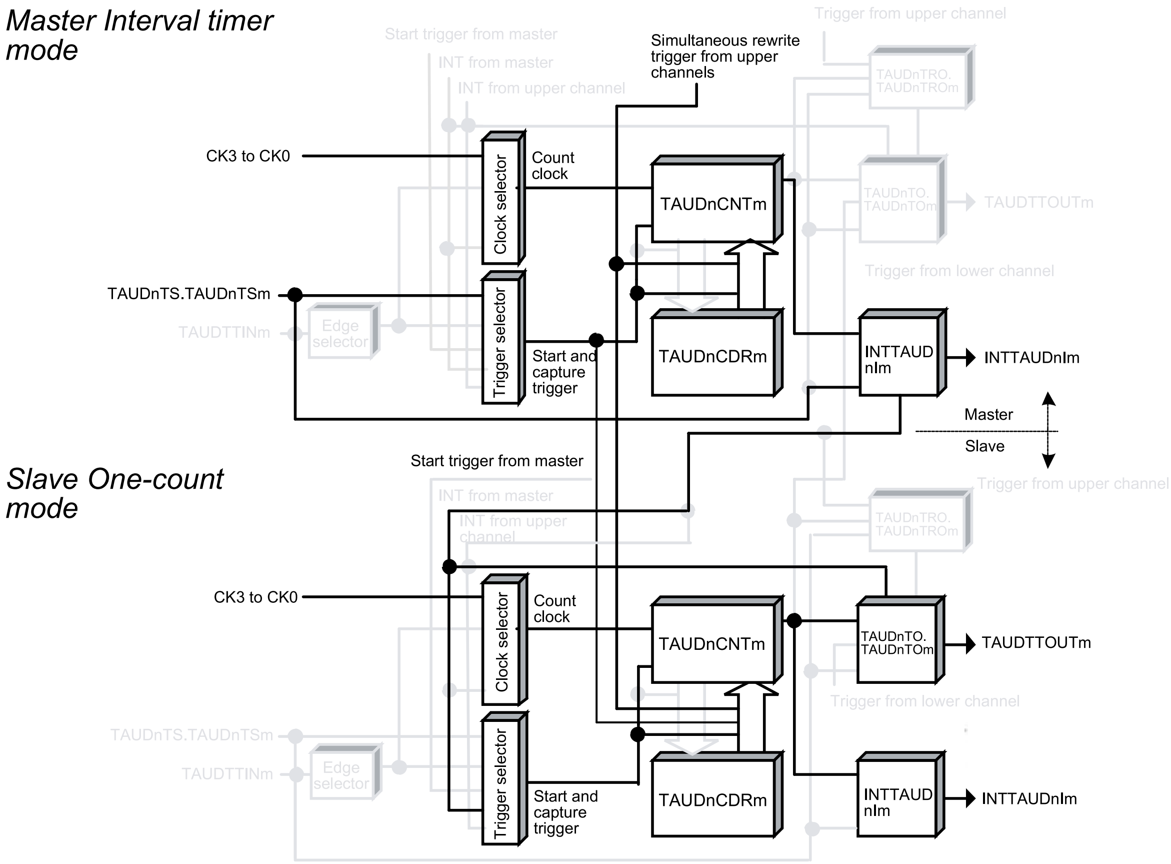


Figure 6 Block Diagram of TAUD PWM Output Function

Please find the General Timing Diagram of the PWM Output Function below:

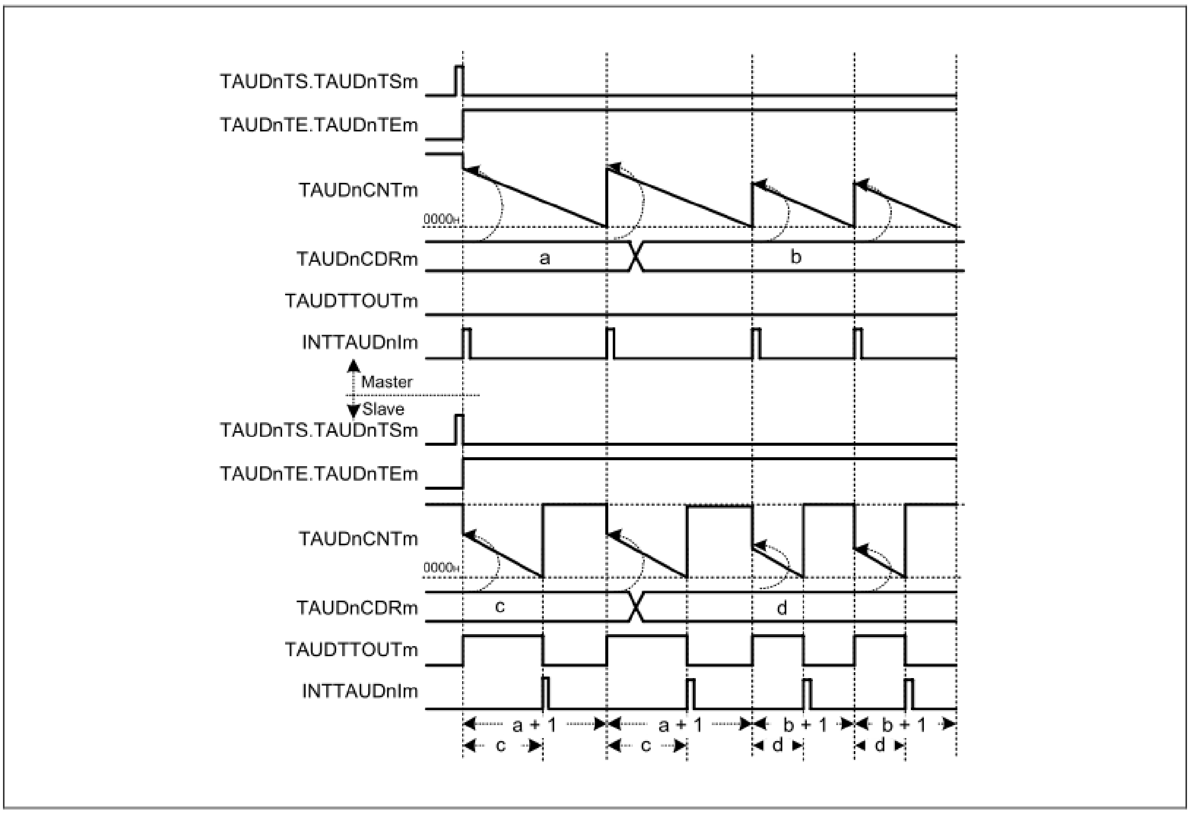


Figure 7 Timing Diagram of PWM Output Function

Note: As the SENT signal is active low, the output (TAUDTTOUTm) is inverted in this application.

3. Software Description

3.1 General Overview

In this application the output of 2 TAUD channels are used to emulate a SENT signal which is transmitted externally to the input of the RSENT Module. The period of a SENT nibble is set in an ISR of the TAUD.

With the used hardware, Port **P10_0** was connected to Port **P9_0** externally.

3.2 Software Flow

After the clock initialization, the TAUD is configured in the above mentioned “PWM output function”, which uses a master channel to generate the period of the SENT signal and consequently the nibble value and a slave channel for the down time of the SENT signal, which is always 3 ticks.

The RSENT module is configured to receive data with 3us tick time, but it can easily be changed in the `config.h`. The “short serial message format” is used per default but can also be changed in `config.h.`

The transmission is started after the fast channel array, the slow channel array and an array with the corresponding tick times are created.

At the beginning of each period an ISR of the TAUD is called, which sets the tick time for the following period depending on the values in “Tick Array” (Please also see 4 Further Development).

Please find below the software flow diagram:

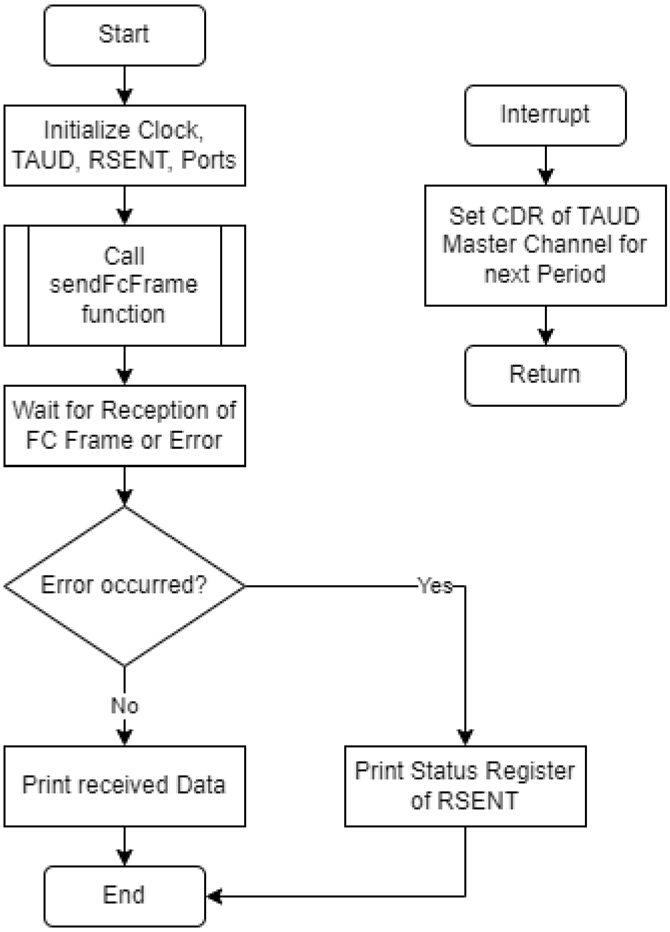


Figure 8 Software Flow

3.3 Files

The software consists of the following files, which are described in the table below:

Table 1. Files

Files in source folder	**Description**
main.c	Contains the main program which • calls initialization functions • initiates the communication • handles the received data and errors
SENT_Emu.h	Contains the following definitions: • Fast Channel data struct • Fast Channel tick time struct • Short serial message data struct • Enhanced serial message data struct
SENT_Emu.c	Contains the following functions: • SENT emulation initialization • SENT emulation frame generation and transmission start • Stop transmission • Interrupt Service Routine for tick time setting • CRC calculation
r_sent.h	Contains • Register definitions to easily use another RSENT channel
r_sent.c	Contains the following functions: • Initialization function for RSENT module
r_port.h and r_port.c	Port driver functions
config.h	Basic SENT related configurations.
Files in device folder	
device.h	Device specific header file with some definitions.
dr7f701684.dev.h	Device file header. Contains register names and addresses of the specific device.
dr7f701684_irq.h	Interrupt enabling and address allocation

`Config.h`

In this file, you can make some basic configurations related to the SENT Standard:

`SHORT_SERIAL_MESSAGE` (or `ENHANCED_SERIAL_MESSAGE`): You can decide what slow channel format you want to use. The short serial message format contains 8 bit of data and the enhanced serial message format contains 12 bit of data.

`TICK_TIME`: You can define the used tick time here. Please keep in the range of the SENT Standard:

3µs < tick time < 90µs

`FIXED_MESSAGE_LENGTH`: Change the fixed message length in the given thresholds:

282 <= FIXED_MESSAGE_LENGTH <= 1038

Some parts of the other files are described below.

3.4 Structs and Arrays

The following typedefs, structs and arrays are defined in SENT_Emu.h:

`scData_T` - Contains the slow channel data which is sent consecutively. Can directly be changed by CPU or DMA.

`fcFrame_T` - Contains the fast channel data nibbles. It is set and send by the sendFcFrame() function.

`ssmFrame_T` - Contains the short serial message frame data.

`esmFrame_T` - Contains the enhanced serial message frame data.

3.5 Functions and variables

The main function is described above.

r_sent.c:

`void R_SENT_Init(void);`	- RSENT initialization function
`void R_SENT_Deinit(void);`	- RSENT deinitialization function
`u32_T R_SENT_RxData(void);`	- Returns the received data from the RSENT module. You can uncomment a desired format.

r_port.c. (Only the used functions are described below):

```
void R_PORT_SetAltFunc(enum port_t Port, u32_T Pin, enum alt_t Alt, enum io_t
IO)
```

- Sets the alternative function of the port and takes as arguments the Port (e.g. Port0), the Pin (e.g. 6) and the alternative function (e.g. Alt6)

```
void R_PORT_SetDigitalFilter(enum dnfa_signal_t InputSignal, u08_T
FilterSetting)
```

- Sets the digital filter settings. Takes the filter register you want to set (e.g. R_DNFA_SENT1RX) and the filter setting you want to set (e.g. 0x67)

SENT_Emu.c:

```
void sentEmuInit(u08_T taud0SlaveChannel)
```

- Initializes the TAUD and takes the desired TAUD0 Slave Channel as an input which is also the output channel for the emulated SENT signal. Please only use ODD channels for that. The master channel number is automatically the channel (e.g. 0) before the slave channel number (e.g. 1).

```
void sendFcFrame( u16_T signal1, u16_T signal2, u08_T taud0SlaveChannel)
```

- Creates and sends a Fast Channel Frame which takes the two 12-bit signals that should be sent in the Fast Channel Frame as arguments and the desired TAUD0 slave channel.
    - `scBitCounter` holds the current bit position that is sent via the current fast channel message.
    - `scMessageCounter` holds the current message ID that is sent via the current slow channel message.
    - `serialDataBitsX` holds the data bits for the enhanced serial message format to be sent in the fast channel messages.

Please find a simplified flow chart of the function below:

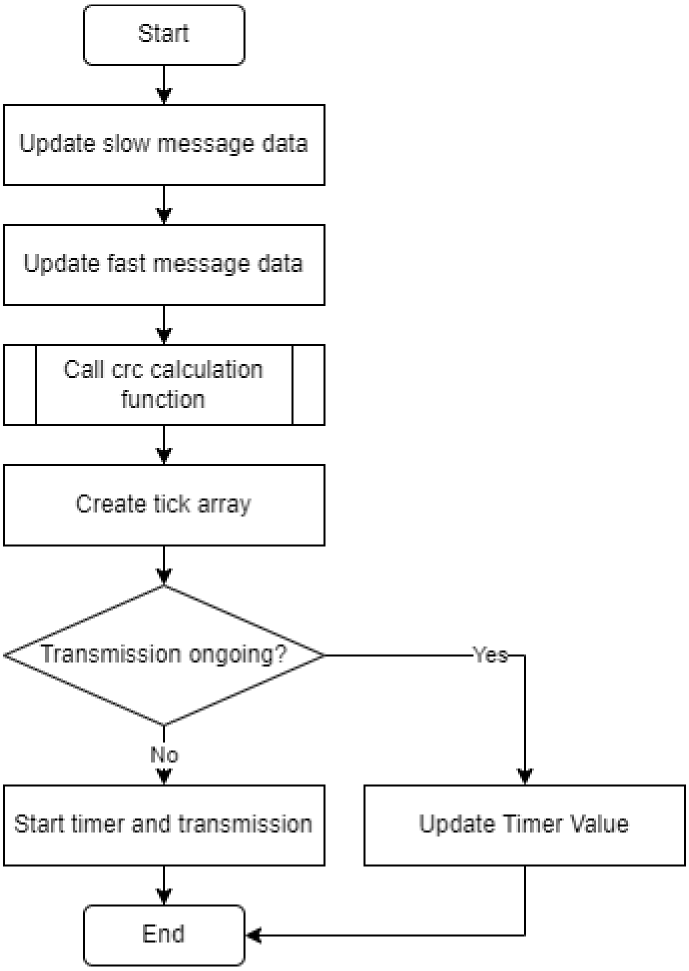


Figure 9 Flow chart of sendFcFrame

`void sendFcDummyFrame(u08_T taud0SlaveChannel)`

- Send a fast channel dummy frame at the beginning of the transmission.
  - The SENT Protocol is intended to be a stream of continuous messages. Therefore, it is necessary, because the beginning of an enhanced serial message starts with “1111110” but, according to the SENT Standard, to be recognized correctly the receiver needs a preceding “0” (start condition “01111110”), which is the normal case in a stream of messages, but not in case of the first message.
  - The last edge after the pause pulse is also the first edge of the next message and the fast channel data is not computed until the reception of this edge. Therefore a preceding message is necessary to correctly receive the fast channel messages.

`void stopTransmission(u08_T taud0SlaveChannel)`

- Stops the SENT transmission on channel “taud0SlaveChannel”

“Private” functions of SENT_Emu.c:

`void setDefaultData(void)`

- Sets some default values in the slow channel data struct `u08_T calc4BitCrc(u08_T* dataArray, u08_T startPtr, u08_T endPtr)`

Calculates the 4-bit CRC for the fast and the slow channel message. Takes the data array, a start and an end pointer as arguments. Example: The following function call would return the CRC of the short serial massage frame from the 1st to the 3rd data: `calc4BitCrc(ssmFrame.array, 1, 3);`

3.6 Interrupt Service Routines

There is one ISR in the SENT_Emu.c file called `EINTPRIO_15`. Every TAUD channel calls the Priority 15 Interrupt Service Routine `EINTPRIO_15` and the period time of the next SENT period is set. The EEIC register is used to determine which TAUD channel called the interrupt or if another interrupt called the Priority 15 interrupt service routine. In that case the ISR would be left immediately.

The `nibble_counter` variable holds the current nibble number that is sent. When the pause pulse is sent the next nibble is either the calibration pulse (if new data is available) or the transmission is stopped.

Note: Please keep in mind that this ISR can’t be delayed more than ~30us because otherwise the frame can’t be sent correctly.

4. Further Development

The software is designed for further development:

- By changing the taud0SlaveChannel variable, you could easily change the used TAUD Channels.
- The used ports can easily be changed by changing the R_PORT_SetAltFunc() arguments
- Another RSENT channel could be configured by changing the defines in r_sent.h
- You can change the used slow channel format and the tick time as well as the fixed message length in the config.h
- The fcTickArray[10] could be written to the TAUDnCDR register via DMA. In that case a DMA interrupt needs to perform the transition to the next frame.
- The program can be ported to most RH850 devices containing a TAUD.

Revision History

Rev.	Date	Description	
		Page	Summary
1.0	Dec.15.23	All	First Edition issued

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

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/.