

# Readme

## **BMP561 SDK**

**Rev: V1.0**

# 1 Introduction

The Geehy Semiconductor BMP561 software development kit includes a series driver library, a group of example applications that demonstrate key peripheral functionality, and other development files.

Software development kit have a hierarchy as follows:

- SDK directory
  - \* [Boards](#)
  - \* [Documents](#)
  - \* [Examples](#)
  - \* [Libraries](#)
  - \* [Package](#)

## Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
<b>2</b>	<b>About boards .....</b>	<b>3</b>
<b>3</b>	<b>About documents .....</b>	<b>4</b>
<b>4</b>	<b>About examples.....</b>	<b>5</b>
4.1	CADC_ContinuousConversion.....	7
4.2	EINT_WakeUp .....	7
4.3	FLASH_Write .....	7
4.4	GPIO_Toggle .....	7
4.5	HMAC_CalcMessage .....	8
4.6	HSC_TwoBoardsPolling .....	8
4.7	I2C_TwoBoards .....	8
4.8	IAP_Application .....	9
4.9	IAP_BootLoader .....	9
4.10	IWDT_FeedDog .....	10
4.11	Template .....	10
4.12	TMR_PWM .....	11
4.13	TMR_TimeBase .....	11
4.14	UART_Interrupt .....	11
4.15	UART_Polling .....	11
4.16	VADC_ContinuousConversion.....	12
4.17	WUPT_WakeUp .....	12
4.18	WWDT_OverTime.....	12
<b>5</b>	<b>About libraries .....</b>	<b>14</b>
<b>6</b>	<b>About Package .....</b>	<b>15</b>
<b>7</b>	<b>Revision History .....</b>	<b>16</b>

## 2 About boards

The boards folder includes a board support package for BMP561 board. It can help drive the peripheral circuit or components on the board quickly. The BSP can be found in the [~/Boards](#) directory.

The BSP provided are built for BMP561 board compatibility. For other user development board use, some minor modifications may be required.

Boards have a hierarchy as follows:

- board\_delay.h
- board\_delay.c

### 3 **About documents**

The documents folder includes a link file that can be redirected to the technical support center of Geehy semiconductor. The document can be found in the [~/Documents](#) directory.

## 4 About examples

The example applications can be found in the [~/Examples](#) directory.

The examples provided are built for BMP561 board compatibility. For other user development board use, some minor modifications may be required.

Example projects have a hierarchy as follows:

- Example folder

- \* Include
- \* Project
  - IAR
  - MDK
- \* Source

All example applications tested with: **BMP561\_StdPeriphDriver V1.0.0**, include the following examples:

- Examples

- \* CADC
  - [CADC\\_ContinuousConversion](#)
- \* EINT
  - [EINT\\_WakeUp](#)
- \* Flash
  - [FLASH\\_Write](#)
- \* GPIO
  - [GPIO\\_Toggle](#)
- \* HMAC
  - [HMAC\\_CalcMessage](#)
- \* HSC
  - [HSC\\_TwoBoardsPolling](#)
- \* I2C
  - [I2C\\_TwoBoards](#)

- \* IAP
  - [IAP\\_Application](#)
  - [IAP\\_BootLoader](#)
- \* IWDT
  - [IWDT\\_FeedDog](#)
- \* Template
  - [Template](#)
- \* TMR
  - [TMR\\_PWM](#)
  - [TMR\\_TimeBase](#)
- \* UART
  - [UART\\_Interrupt](#)
  - [UART\\_Polling](#)
- \* VADC
  - [VADC\\_ContinuousConversion](#)
- \* WUPT
  - [WUPT\\_WakeUp](#)
- \* WWDT
  - [WWDT\\_OverTime](#)

## **4.1 CADC\_ContinuousConversion**

### **4.1.1 Example Description**

This demo shows how to use the CADC. After the system starts, the CADC will continuously convert the input current values..

### **4.1.2 Directory contents**

This example can be found in the [~/CADC/CADC\\_ContinuousConversion](#) directory.

## **4.2 EINT\_WakeUp**

### **4.2.1 Example Description**

This demo shows how to use EINT to wake up system for low power mode.

When the system enter low power mode, give a falling edge to GPIO4, system will be awakened and GPIO3 will be set to a high level.

### **4.2.2 Directory contents**

This example can be found in the [~/EINT/EINT\\_WakeUp](#) directory.

## **4.3 FLASH\_Write**

### **4.3.1 Example Description**

This demo is based on the BMP561 board. This example implements data programming and erasing operations using flash memory.

### **4.3.2 Directory contents**

This example can be found in the [~/Flash/FLASH\\_Write](#) directory.

## **4.4 GPIO\_Toggle**

### **4.4.1 Example Description**

This example describes how to use DOUT for toggling IO.



#### **4.4.2 Directory contents**

This example can be found in the [~/GPIO/GPIO\\_Toggle](#) directory.

### **4.5 HMAC\_CalcMessage**

#### **4.5.1 Example Description**

The example is used to configure HMAC, and uses HMAC to calculate the pData data, storing the data in an array named 'result'. You can add the 'result' data to the watch window in debug mode to view its value.

#### **4.5.2 Directory contents**

This example can be found in the [~/HMAC/HMAC\\_CalcMessage](#) directory.

### **4.6 HSC\_TwoBoardsPolling**

#### **4.6.1 Example Description**

This routine is a simple HSC dual-board communication, using GPIO3(SDA) to connect two BMP561 development boards and additional pull-up resistors, allowing data transmission values to be viewed in debug mode.

#### **4.6.2 Directory contents**

This example can be found in the [~/HSC/HSC\\_TwoBoardsPolling](#) directory.

### **4.7 I2C\_TwoBoards**

#### **4.7.1 Example Description**

This routine is a template routine for I2C dual-board communication.

The Master development board sends a piece of data to the Slave development board at a fixed interval, and the device receives this data in real time.

The wiring is as follows:

GPIO2(SCL) --> GPIO2(SCL)

GPIO3(SDA) --> GPIO3(SDA)

## 4.7.2 Directory contents

This example can be found in the [~/I2C/I2C\\_TwoBoards](#) directory.

## 4.8 IAP\_Application

### 4.8.1 Example Description

This example shows how to generate a APP firmware to IAP.

GPIO4 are toggled with a timing defined by the Delay function.

note:

Before using IAP function, you must download the bootloader program to the target board.

Then compile the "Application1" project, and update the generated bin

firmware file to the target board with the function of IAP through the HyperTerminal tool.

### 4.8.2 Directory contents

This example can be found in the [~/IAP/IAP\\_Application](#) directory.

## 4.9 IAP\_BootLoader

### 4.9.1 Example Description

The example aim to show how to configure a bootloader firmware to IAP.

When device connet to HyperTerminal right, a usart menu will show to user.

The menu as follow:

\*\*\*\*\*

\* BMP561 In-Application Programming Application \*

\* (Version 1.0.0)\*

\*\*\*\*\*

\*\* Please select an operation item

- \* 1.Download Flash application 1 -----> 1 \*
- \* 2.Upload Flash application 1 -----> 2 \*
- \* 3.Jump to user application 1 -----> 3 \*

\*\*\*\*\*

note:

Before using IAP function, you must download the bootloader program to the target board.

Then compile the "Application1" or "Application2" project, and update the generated bin firmware file to the target board with the function of IAP through the HyperTerminal tool.

## 4.9.2 Directory contents

This example can be found in the [~/IAP/IAP\\_BootLoader](#) directory.

## 4.10 IWDT\_FeedDog

### 4.10.1 Example Description

This demo show how to use IWDT. The system will keep feeding the dog, and after feed it 5 times, the system will stop feeding dog, and then the system will occur a IWDT reset.

### 4.10.2 Directory contents

This example can be found in the [~/IWDT/IWDT\\_FeedDog](#) directory.

## 4.11 Template

### 4.11.1 Example Description

This demo is based on the BMP561 board. it provides a template project.

### 4.11.2 Directory contents

This example can be found in the [~/Template/Template](#) directory.

[www.geehy.com](http://www.geehy.com)

## 4.12 TMR\_PWM

### 4.12.1 Example Description

This demo shows how to use TMR. TMR will output 50% duty PWM through GPIO pin 0(INT).

### 4.12.2 Directory contents

This example can be found in the [~/TMR/TMR\\_PWM](#) directory.

## 4.13 TMR\_TimeBase

### 4.13.1 Example Description

This demo shows how to use TMR. TMR will toggle the GPIO pin 3(SDA) every microsecond.

### 4.13.2 Directory contents

This example can be found in the [~/TMR/TMR\\_TimeBase](#) directory.

## 4.14 UART\_Interrupt

### 4.14.1 Example Description

This demo shows how to use the UART. The UART will return the received data with interrupt mode.

### 4.14.2 Directory contents

This example can be found in the [~/UART/UART\\_Interrupt](#) directory.

## 4.15 UART\_Polling

### 4.15.1 Example Description

This demo shows how to use the UART. The UART will return the received data  
[www.geehy.com](http://www.geehy.com)

with polling mode..

#### **4.15.2 Directory contents**

This example can be found in the [~/UART/UART\\_Polling](#) directory.

### **4.16 VADC\_ContinuousConversion**

#### **4.16.1 Example Description**

This demo shows how to use the VADC. After the system starts, the VADC will continuously convert the input voltage values.

#### **4.16.2 Directory contents**

This example can be found in the [~/VADC/VADC\\_ContinuousConversion](#) directory.

### **4.17 WUPT\_WakeUp**

#### **4.17.1 Example Description**

This demo shows how to use the WUPT. When system enter low power mode, WUPT will wake up the system in 5 seconds.

#### **4.17.2 Directory contents**

This example can be found in the [~/WUPT/WUPT\\_WakeUp](#) directory.

### **4.18 WWDT\_OverTime**

#### **4.18.1 Example Description**

This demo show how to use WWDT. The system will feed the dog in interrupt, and after feed it 5 times, the system will stop feeding dog, and then the system will occur a WWDT reset.

#### **4.18.2 Directory contents**

This example can be found in the [~/WWDT/WWDT\\_OverTime](#) directory.



## 5 About libraries

The libraries folder includes a series library. It can provide supports for BMP561 MCU such as device support and standard peripheral etc. The libraries can be found in the [~/Libraries](#) directory.

BMP561 MCU include following library:

- Libraries folder
  - \* BMP561\_StdPeriphDriver
  - \* CMSIS
  - \* Device

## 6 About Package

The Package folder includes Geehy BMP561 DFP Package. The Package can be found in the [~/Package](#) directory.

The middlewares used by BMP561 include following:

- Package folder
  - \* Flash
  - \* SVD
  - \* Geehy.BMP561\_DFP.1.0.0.pack



## 7 Revision History

Table 1 File Revision History

Date	Rev	Description
2025.01.13	1.0	First Release version of BMP561 SDK

## Statement

This document is formulated and published by Geehy Semiconductor Co., Ltd. (hereinafter referred to as “Geehy”). The contents in this document are protected by laws and regulations of trademark, copyright and software copyright. Geehy reserves the right to make corrections and modifications to this document at any time. Please read this document carefully before using Geehy products. Once you use the Geehy product, it means that you (hereinafter referred to as the “users”) have known and accepted all the contents of this document. Users shall use the Geehy product in accordance with relevant laws and regulations and the requirements of this document.

### 1. Ownership

This document can only be used in connection with the corresponding chip products or software products provided by Geehy. Without the prior permission of Geehy, no unit or individual may copy, transcribe, modify, edit or disseminate all or part of the contents of this document for any reason or in any form.

The “极海” or “Geehy” words or graphics with “®” or “TM” in this document are trademarks of Geehy. Other product or service names displayed on Geehy products are the property of their respective owners.

### 2. No Intellectual Property License

Geehy owns all rights, ownership and intellectual property rights involved in this document.

Geehy shall not be deemed to grant the license or right of any intellectual property to users explicitly or implicitly due to the sale or distribution of Geehy products or this document.

If any third party's products, services or intellectual property are involved in this document, it shall not be deemed that Geehy authorizes users to use the aforesaid third party's products, services or intellectual property, unless otherwise agreed in sales order or sales contract.

### 3. Version Update

Users can obtain the latest document of the corresponding models when ordering Geehy products.

If the contents in this document are inconsistent with Geehy products, the agreement in the sales order or the sales contract shall prevail.

### 4. Information Reliability

The relevant data in this document are obtained from batch test by Geehy Laboratory or cooperative third-party testing organization. However, clerical errors in correction or errors caused by differences in testing environment may occur inevitably. Therefore, users should understand that Geehy does not bear any responsibility for such errors that may occur in this document. The relevant data in this document are only used to guide users as performance parameter reference and do not constitute Geehy's guarantee for any product performance.

Users shall select appropriate Geehy products according to their own needs, and effectively verify and test the applicability of Geehy products to confirm that Geehy products meet their own needs, corresponding standards, safety or other reliability requirements. If losses are caused to users due to the user's failure to fully verify and test Geehy products, Geehy will not bear any responsibility.

## 5. Legality

USERS SHALL ABIDE BY ALL APPLICABLE LOCAL LAWS AND REGULATIONS WHEN USING THIS DOCUMENT AND THE MATCHING GEEHY PRODUCTS. USERS SHALL UNDERSTAND THAT THE PRODUCTS MAY BE RESTRICTED BY THE EXPORT, RE-EXPORT OR OTHER LAWS OF THE COUNTRIES OF THE PRODUCTS SUPPLIERS, GEEHY, GEEHY DISTRIBUTORS AND USERS. USERS (ON BEHALF OR ITSELF, SUBSIDIARIES AND AFFILIATED ENTERPRISES) SHALL AGREE AND PROMISE TO ABIDE BY ALL APPLICABLE LAWS AND REGULATIONS ON THE EXPORT AND RE-EXPORT OF GEEHY PRODUCTS AND/OR TECHNOLOGIES AND DIRECT PRODUCTS.

## 6. Disclaimer of Warranty

THIS DOCUMENT IS PROVIDED BY GEEHY "AS IS" AND THERE IS NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, TO THE EXTENT PERMITTED BY APPLICABLE LAW.

GEEHY WILL BEAR NO RESPONSIBILITY FOR ANY DISPUTES ARISING FROM THE SUBSEQUENT DESIGN OR USE BY USERS.

## 7. Limitation of Liability

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL GEEHY OR ANY OTHER PARTY WHO PROVIDE THE DOCUMENT "AS IS", BE LIABLE FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, DIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE DOCUMENT (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY USERS OR THIRD PARTIES).

## 8. Scope of Application

The information in this document replaces the information provided in all previous versions of the document.

© 2024 Geehy Semiconductor Co., Ltd. - All Rights Reserved