

<b>GigaDevice MCU</b>  <b>Team</b>	<b>Version</b>	<b>8 Pages</b>
	<b>English V 1.0</b>	
	<b>Name: GigaDevice PMU Monitor User Manual</b>	

# **GigaDevice PMU Monitor**

## **User Manual**

## Contents

GigaDevice PMU Monitor.....	1
User Manual.....	1
<b>1. Introduction.....</b>	<b>3</b>
1.1 Function description.....	3
1.2 Purpose.....	3
1.3 Operating environment.....	3
1.4 Package composition.....	3
<b>2. Running.....</b>	<b>4</b>
<b>3. Using Details.....</b>	<b>4</b>
3.1 Layout introduction.....	4
3.2 Flowchart of Operation.....	5
3.3 Bit field value modification.....	6
3.4 Register operation.....	8
<b>4. Attentions.....</b>	<b>8</b>
<b>5. Update.....</b>	<b>8</b>
<b>6. Q&amp;A.....</b>	<b>8</b>

## 1. Introduction

The document introduces GD32 PMU Monitor software, which is convenient for users to familiarize themselves with the software usage and operation process more quickly.

### 1.1 Function description

GD32 PMU Monitor is a tool for battery management unit. It is connected to the development board through the USB Adapter to directly read, edit, and set the register value, as well as monitor the status and value of the register in real time. Users can directly operate the registers through this tool, without MCU programming control. GD32 PMU Monitor visualizes register configuration and status, reduces the difficulty of development, and simplifies the development process.

### 1.2 Purpose

In order to reduce the difficulty of development, simplify the development process. GD32 PMU Monitor was developed.

Users can directly operate the PMU through the pull-down selection, tick, and register value modification. Users can also monitor each register through software to view the operating status of the PMU.

The visualized operation makes the development process more convenient and friendly.

### 1.3 Operating environment

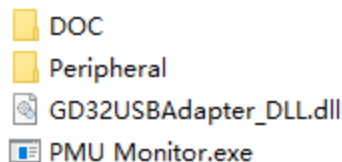
Operating system: win7/win10 64 bit

Processor: i3-9100 3.6GHz

Screen resolution: not less than 1218\*700

### 1.4 Package composition

The Package contains the following files and folders:



The Doc folder: Include the software user manual.

The Peripheral folder: Include the XML files of each series PMU.

The DLL file: USB Adapter dynamic link library.

Exe file: Software running file.

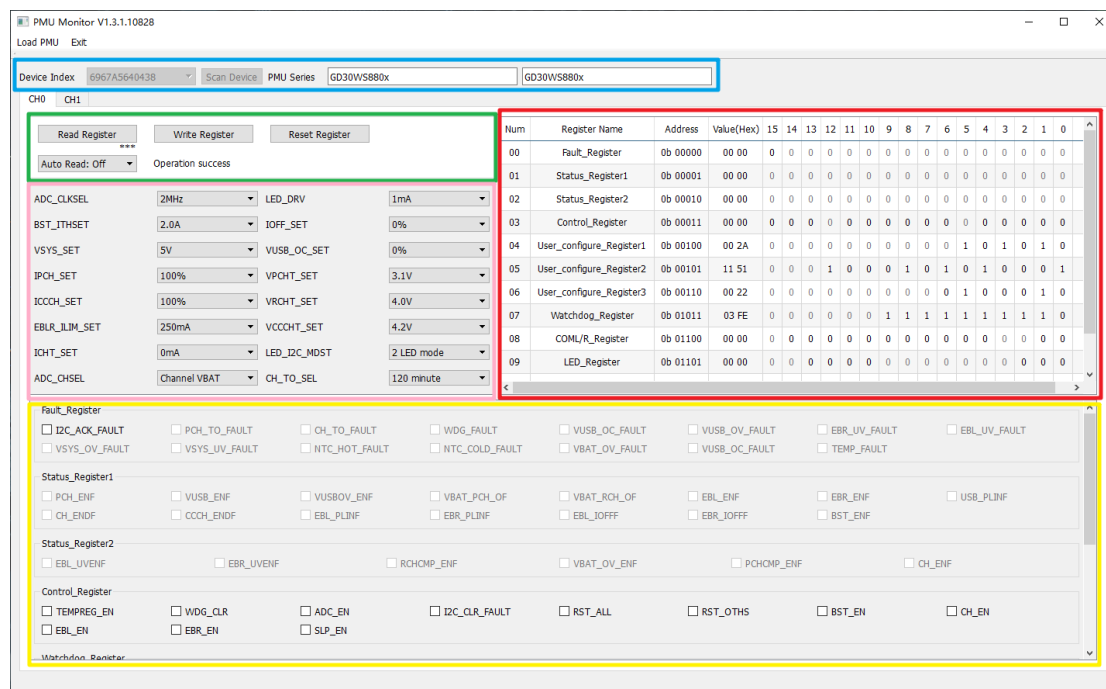
## 2. Running

This software is running on PC and compatible computers, and on platforms of WINDOWS.

There's no need to setup the software, the only thing you need to do is to click the icon to operate the software.

## 3. Using Details

### 3.1 Layout introduction



The software can support two series of PMU monitoring at the same time, and operate different PMU by switching different Tab.

Device and series display ( blue ) : Display USB Adapter and PMU series.

Register operation ( green ) : Read registers, write registers, reset all registers , Automatic reading time setting and Operation tips.

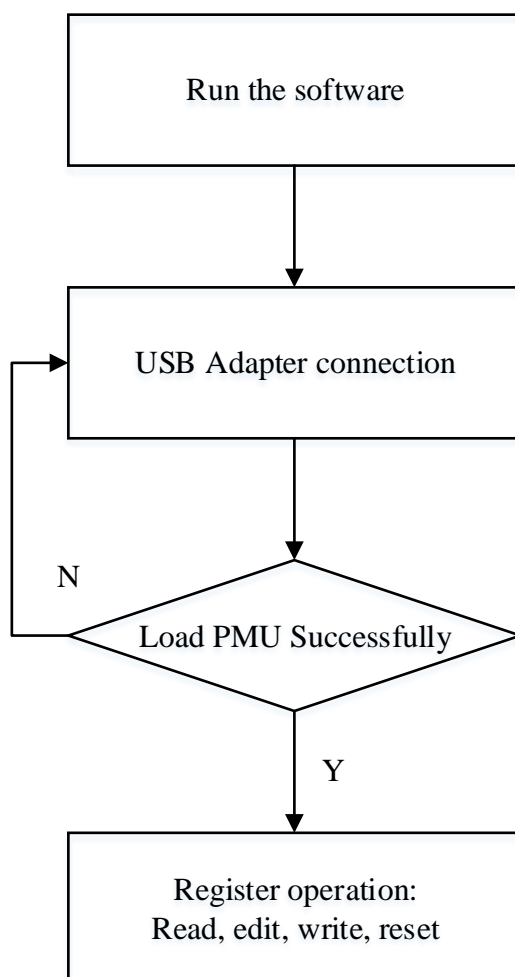
Bit field operation ( red ) : Click the mouse to change the value of the bit to realize the configuration of the register.

Function selection ( pink ) : Pull-down switch register function.

Register enable ( yellow ) : Check to enable the register.

Bit field operation is related to drop-down selection and check enable. If one of them is changed, the related one will be changed synchronously.

## 3.2 Flowchart of Operation



### 3.3 Bit field value modification

Click the bit in the bit field to switch the value of the bit to 0/1, thereby changing the value of the register. There will be a corresponding description when hovering over the position. The black bits are editable bits, and the gray bits are uneditable bits.

The value of the bit field is correlated with the check enable area. Changing the value of the bit field, the enable of the corresponding register will also be changed according to the value of the bit field. On the contrary, if the enable area is checked, the value of the bit field will also change accordingly.

Num	Register Name	Address	Value(Hex)	15	14	13	12	11	10	9	8	7	6	5	4	3	2
00	Fault_Register	0b 00000	80 00	1	0	0	0	0	0	0	0	0	0	0	0	0	0
01	Status_Register1	0b 00001	00 00	0	Bit15 Software can clear it by writing 1 00 : I2C normally 01 : I2C ACK fault/clear								0	0	0	0	0
02	Status_Register2	0b 00010	00 00	0									0	0	0	0	0
03	Control_Register	0b 00011	00 00	0	0	0	0	0	0	0	0	0	0	0	0	0	0
04	User_configure_Register1	0b 00100	00 2A	0	0	0	0	0	0	0	0	0	0	1	0	1	0
05	User_configure_Register2	0b 00101	11 51	0	0	0	1	0	0	0	1	0	1	0	1	0	0
06	User_configure_Register3	0b 00110	00 22	0	0	0	0	0	0	0	0	0	0	1	0	0	0
07	Watchdog_Register	0b 01011	03 FE	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Fault\_Register

☒ I2C\_ACK\_FAULT

☐ PCH\_TO\_FAULT
 ☐ CH\_TO\_FAULT
 ☐ WDG\_FAULT
 ☐ VUSB\_OC\_FAULT
 ☐ VUSB\_OV\_FAULT
 ☐ EBR\_UV\_FAULT
 ☐ EBL\_UV\_FAULT

☐ VSYS\_OV\_FAULT
 ☐ VSYS\_UV\_FAULT
 ☐ NTC\_HOT\_FAULT
 ☐ NTC\_COLD\_FAULT
 ☐ VBAT\_OV\_FAULT
 ☐ VUSB\_OC\_FAULT
 ☐ TEMP\_FAULT

Num	Register Name	Address	Value(Hex)	15	14	13	12	11	10	9	8	7	6	5	4	3	2
00	Fault_Register	0b 00000	00 00	0	0	0	0	0	0	0	0	0	0	0	0	0	0
01	Status_Register1	0b 00001	00 00	<div>Bit15 Software can clear it by writing 1 00 : I2C normally 01 : I2C ACK fault/clear</div>	0	0	0	0	0	0	0	0	0	0	0	0	0
02	Status_Register2	0b 00010	00 00		0	0	0	0	0	0	0	0	0	0	0	0	0
03	Control_Register	0b 00011	00 00		0	0	0	0	0	0	0	0	0	0	0	0	0
04	User_configure_Register1	0b 00100	00 2A	0	0	0	0	0	0	0	0	0	0	1	0	1	0
05	User_configure_Register2	0b 00101	11 51	0	0	0	1	0	0	0	1	0	1	0	1	0	0
06	User_configure_Register3	0b 00110	00 22	0	0	0	0	0	0	0	0	0	0	1	0	0	0
07	Watchdog_Register	0b 01011	03 FE	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Fault\_Register

☐ I2C\_ACK\_FAULT

☐ PCH\_TO\_FAULT
 ☐ CH\_TO\_FAULT
 ☐ WDG\_FAULT
 ☐ VUSB\_OC\_FAULT
 ☐ VUSB\_OV\_FAULT
 ☐ EBR\_UV\_FAULT
 ☐ EBL\_UV\_FAULT

☐ VSYS\_OV\_FAULT
 ☐ VSYS\_UV\_FAULT
 ☐ NTC\_HOT\_FAULT
 ☐ NTC\_COLD\_FAULT
 ☐ VBAT\_OV\_FAULT
 ☐ VUSB\_OC\_FAULT
 ☐ TEMP\_FAULT

The value of the bit field is related to the drop-down selection area. When the value of the bit field is changed, the corresponding function will be switched according to the value of the bit field.

Conversely, if you switch the pull-down selection, the value of the bit field will also change accordingly.

Register Name	Address	Value(Hex)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Fault_Register	0b 00000	00 00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Status_Register1	0b 00001	00 00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Status_Register2	0b 00010	00 00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Control_Register	0b 00011	00 00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
User_configure_Register1	0b 00100	00 2A	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1
User_configure_Register2	0b 00101	11 51	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
User_configure_Register3	0b 00110	00 22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Watchdog_Register	0b 01011	03 FE	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1

Bit7  
ADC clock selection  
00 : ADC clock is 2MHz  
01 : ADC clock is 1MHz  
10 : ADC clock is 500kHz  
11 : ADC clock is 250kHz

ADC_CLKSEL	2MHz	LED_DRV	1mA
BST_ITHSET	2.0A	IOFF_SET	0%
VSYS_SET	5V	VUSB_OC_SET	0%
IPCH_SET	100%	VPCHT_SET	3.1V
ICCCH_SET	100%	VRCHT_SET	4.0V
EBLR_ILIM_SET	250mA	VCCCHT_SET	4.2V
ICHT SET	0mA	LED I2C MDST	2 LED mode

Register Name	Address	Value(Hex)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Fault_Register	0b 00000	00 00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Status_Register1	0b 00001	00 00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Status_Register2	0b 00010	00 00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Control_Register	0b 00011	00 40	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
User_configure_Register1	0b 00100	00 2A	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1
User_configure_Register2	0b 00101	11 51	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
User_configure_Register3	0b 00110	00 22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Watchdog_Register	0b 01011	03 FE	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1

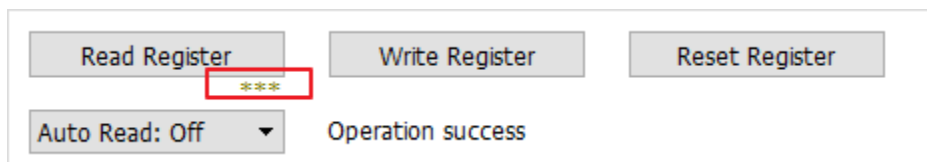
Bit6  
ADC clock selection  
00 : ADC clock is 2MHz  
01 : ADC clock is 1MHz  
10 : ADC clock is 500kHz  
11 : ADC clock is 250kHz

ADC_CLKSEL	1MHz	LED_DRV	1mA
BST_ITHSET	2.0A	IOFF_SET	0%
VSYS_SET	5V	VUSB_OC_SET	0%
IPCH_SET	100%	VPCHT_SET	3.1V
ICCCH_SET	100%	VRCHT_SET	4.0V
EBLR_ILIM_SET	250mA	VCCCHT_SET	4.2V
ICHT SET	0mA	LED I2C MDST	2 LED mode

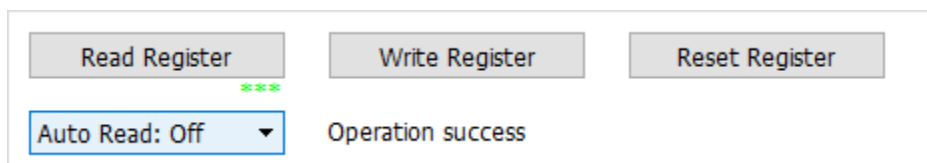
## 3.4 Register operation

Register operations mainly include reading, editing, writing, resetting and automatic reading. There are three \* under the read button to indicate whether the operation is successful.

If it displays dark yellow, it means that the currently displayed register value is modified but not written into the PMU.



If it displays green, it means that the current register value is the value read from the PMU. Each operation will have a corresponding text to display the operation result.



## 4. Attentions

- When the software is working, make sure that the serial port of the device is not occupied, or the software cannot initialize the device.
- When connecting, pay attention to the matching of the channels. The software's channel display Tab should correspond to the USB Adapter channel, that is, the PMU series selected by CHx(CH0/CH1) should be the same as the PMU connected to the USB Adapter CHx(CH0/CH1).

## 5. Update

Get updates on the official website. <http://gd32mcu.com/cn/download>

## 6. Q&A

Q1 : Run the software prompt "No device detected."

A1 : The USB Adapter is not connected or the USB Adapter is occupied, check the connection and scan the device. Displaying the device index indicates that the connection is successful.