GigaDevice Semiconductor Inc.

GD32C113C-START
Arm® Cortex®-M4 32-bit MCU

User Guide

Revision 1.0

(Aug. 2022)

Table of Contents

Table of Contents 1
List of Figures 3
List of Tables 4
1. Summary 5
2. Function Pin Assign 6
3. Getting started 7
4. Hardware layout overview 8
4.1. Power supply 8
4.2. Boot option 8
4.3. LED 8
4.4. KEY 9
4.5. USART 9
4.6. USB 10
4.7. GD-Link 11
4.8. MCU 12
4.9. Ardunio 12
5. Routine use guide 13
5.1. GPIO_Running_LED 13
5.1.1. DEMO purpose 13
5.1.2. DEMO running result 13
5.2. GPIO_Key_Polling_mode 13
5.2.1. DEMO purpose 13
5.2.2. DEMO running result 13
5.3. GPIO_Key_Interrupt_mode 14
5.3.1. DEMO purpose 14
5.3.2. DEMO running result 14
5.4. USART_Hyperterminal_Interrupt 14
5.4.1. DEMO purpose 14
5.4.2. DEMO running result 14
5.5. TIMER_Key_EXTI 15
5.5.1. DEMO purpose 15
5.5.2. DEMO running result 15
5.6. USB_MSC_Device 15
5.6.1. DEMO purpose 15
5.6.2. DEMO running result 16
5.7. USB_HID_Host 16
5.7.1. DEMO purpose 16
5.7.2. DEMO running result 16

6. Revision history 17

List of Figures

Figure 4-1. Schematic diagram of power supply 8
Figure 4-2. Schematic diagram of boot option 8
Figure 4-3. Schematic diagram of LED function 8
Figure 4-4. Schematic diagram of Key function 9
Figure 4-7. Schematic diagram of USART to USB 9
Figure 4-5. Schematic diagram of USB 10
Figure 4-6. Schematic diagram of GD-Link 11
Figure 4-8. Schematic diagram of MCU 12
Figure 4-9. Schematic diagram of Ardunio 12

List of Tables

Table 2-1. Function pin assignment 6
Table 6-1. Revision history 17

1. Summary

GD32C113C-START uses GD32C113CBT6 as the main controller. It uses Mini USB interface to supply 5V power. Reset, Boot, Wakeup key, LED, USB, GD-Link and Ardunio are also included. For more details, please refer to GD32C113C-START_Rev1.0 schematic.

2. Function Pin Assign

Table 2-1. Function pin assignment

Function	Pin	Description
LED	PA7	LED1
RESET		K1-Reset
KEY	PA0	K2-User key
USART	PA9	USART0_TX
	PA10	USART0_RX
USB	PA9	USB_VBUS
	PA11	USB_DM
	PA12	USB_DP
	PA8	USB pull up control

3. Getting started

The START board uses Mini USB connecter to get power DC +5V, which is the hardware system normal work voltage. A GD-Link on board is necessary in order to download and debug programs. Select the correct boot mode and then power on, the LEDPWR will turn on, which indicates that the power supply is OK.

There are Keil version and IAR version of all projects. Keil version of the projects are created based on Keil MDK-ARM 4.74 uVision4. IAR version of the projects are created based on IAR Embedded Workbench for ARM 7.40.2. In Firmware folder, Addon and Software Pack are used to add the devices, peripherals and others to IDE. During use, the following points should be noted:

1. If you use Keil uVision4 to open the project, install the GigaDevice_GD32C11x_AddOn.1.0.0.exe to load the associated files.

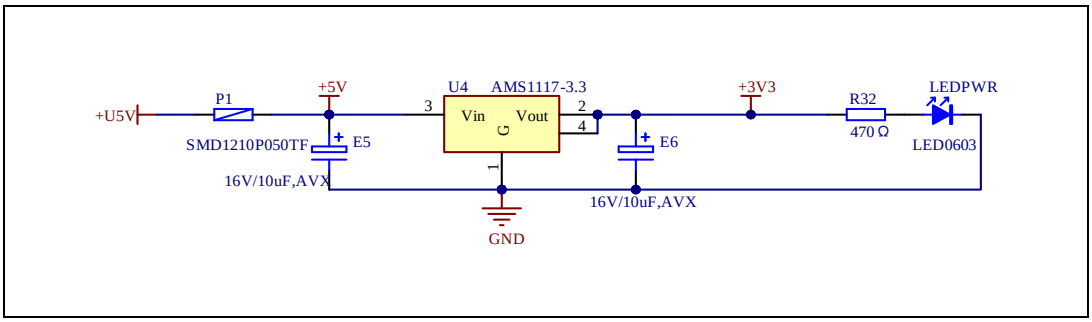
2. If you use Keil uVision5 to open the project, there are two ways to solve the "Device Missing (s)" problem. One is to install GigaDevice.GD32C11x_DFP.1.0.0.pack. In Project menu, select the Manage sub menu, click on the “Version Migrate 5 Format...” menu, the Keil uVision4 project will be converted to Keil uVision5 project. Then add “C:\Keil_v5\ARM\Pack\ARM\CMSIS\4.2.0\CMSIS\Include” to C/C++ in Option for Target. The other is to install Addon directly. Select the installation directory of Keil uVision5 software, such as C:\Keil_v5, in Destination Folder of Folder Selection. Select the corresponding device in Device of Option for Target and add “C:\Keil_v5\ARM\Pack\ARM\CMSIS\4.2.0\CMSIS\Include” to C/C++ in Option for Target.

3. If you use IAR to open the project, install IAR_GD32C11x_ADDON_1.0.0.exe to load the associated files.

4. Hardware layout overview

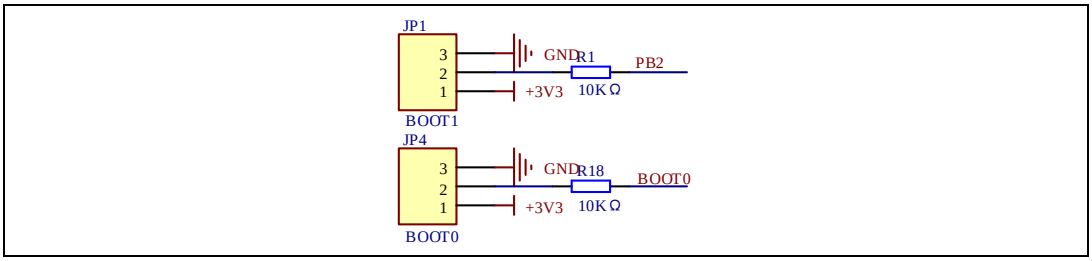
4.1. Power supply

Figure 4-1. Schematic diagram of power supply



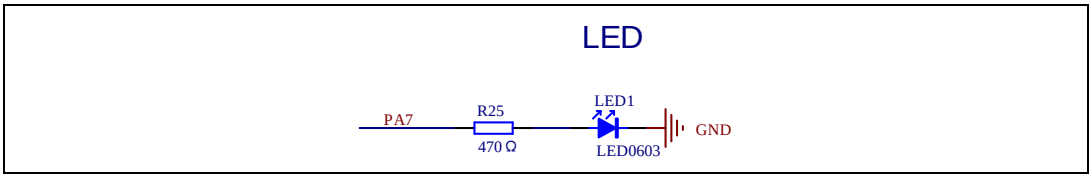
4.2. Boot option

Figure 4-2. Schematic diagram of boot option



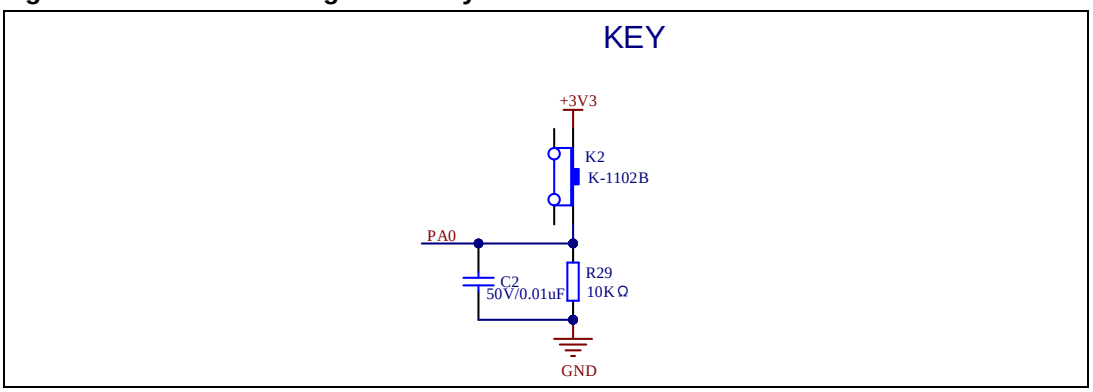
4.3. LED

Figure 4-3. Schematic diagram of LED function



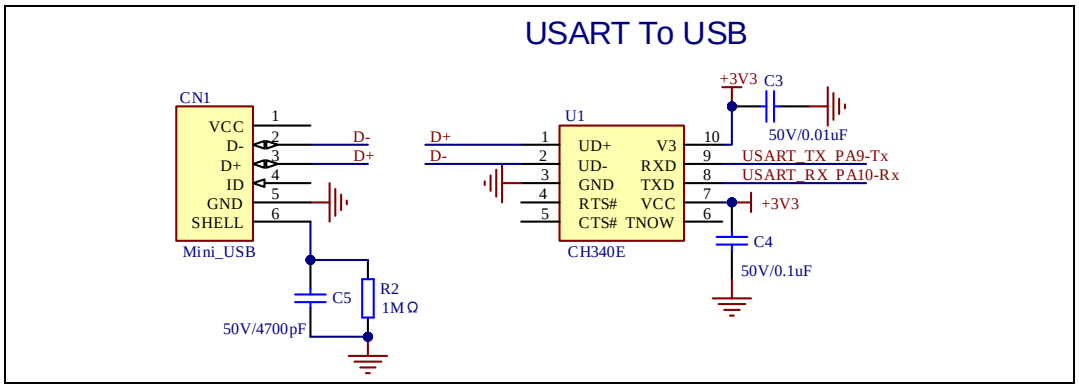
4.4. KEY

Figure 4-4. Schematic diagram of Key function



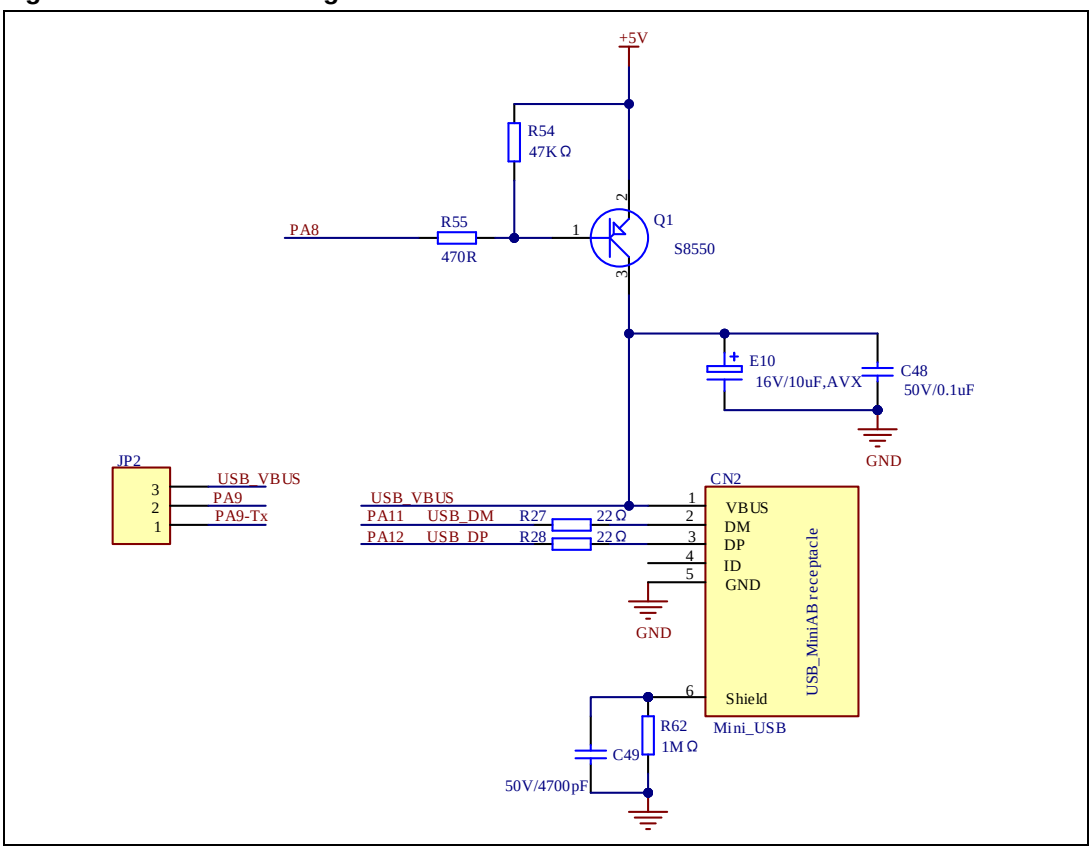
4.5. USART

Figure 4-5. Schematic diagram of USART to USB



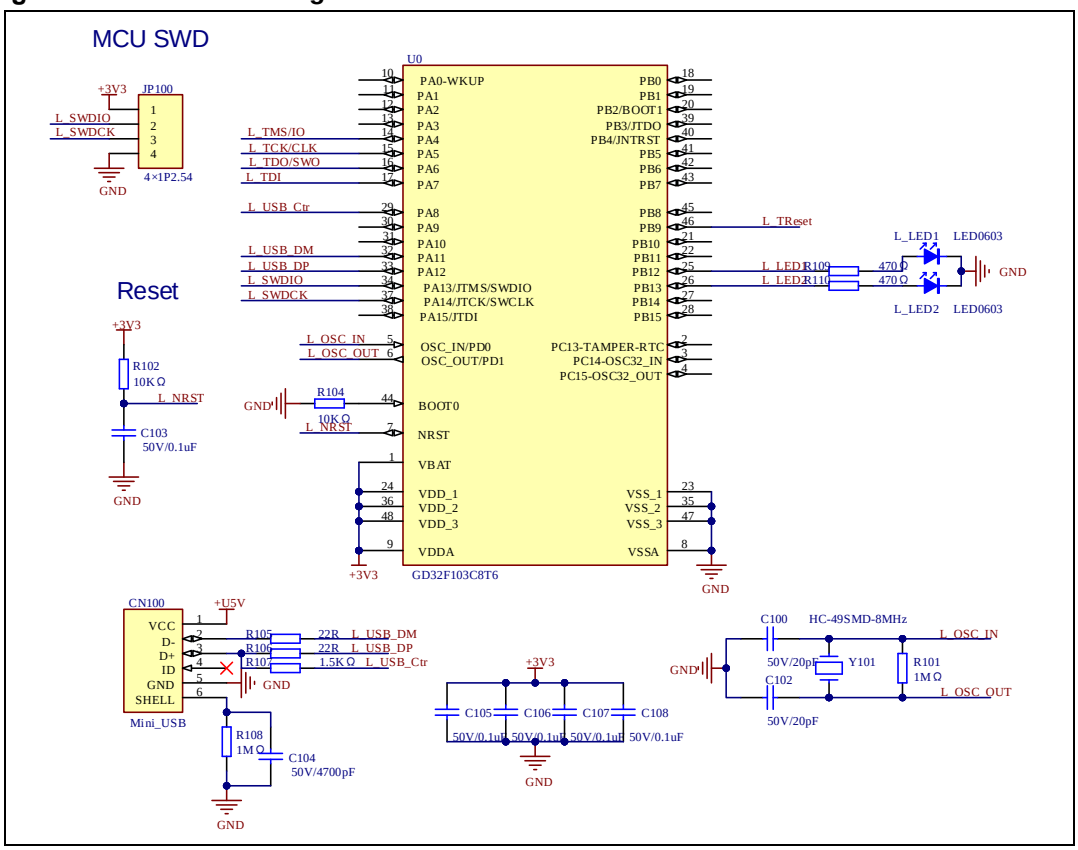
4.6. USB

Figure 4-6. Schematic diagram of USB



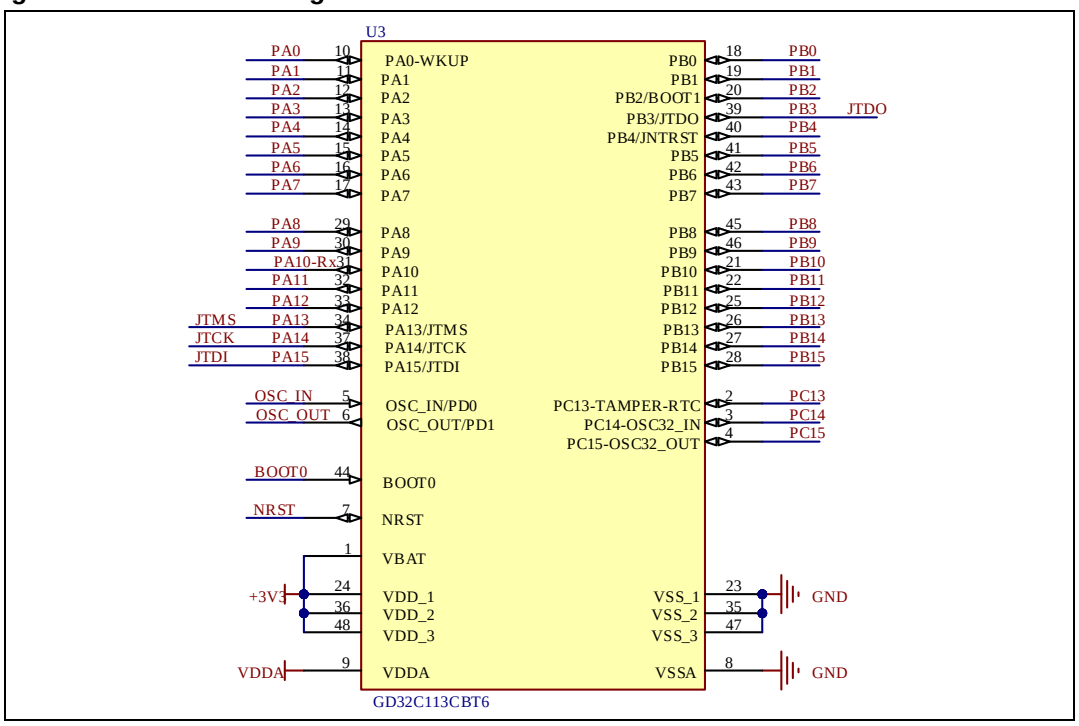
4.7. GD-Link

Figure 4-7. Schematic diagram of GD-Link



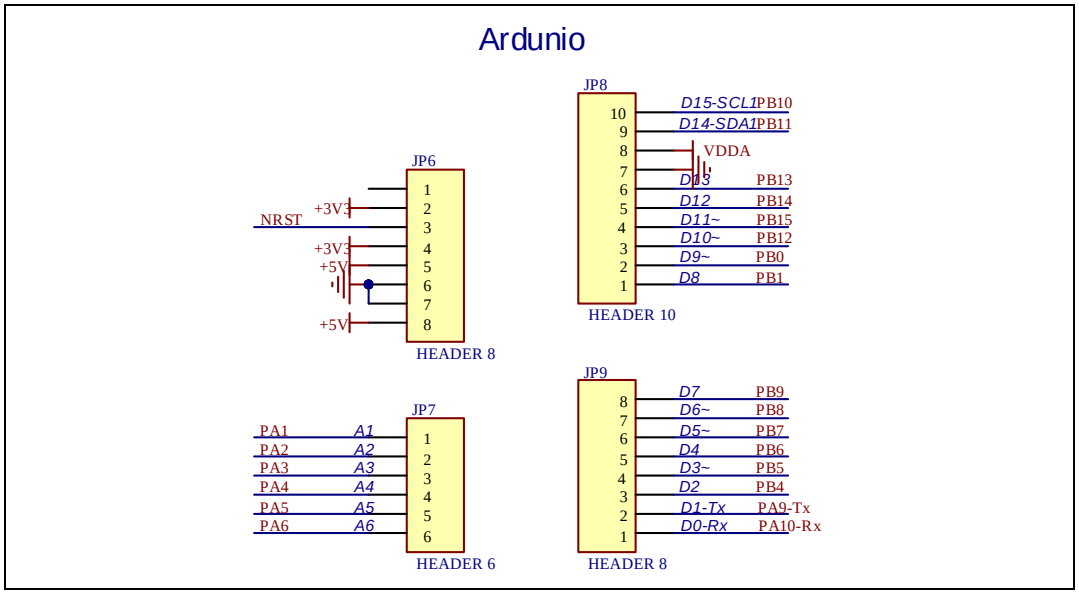
4.8. MCU

Figure 4-8. Schematic diagram of MCU



4.9. Ardunio

Figure 4-9. Schematic diagram of Ardunio



5. Routine use guide

5.1. GPIO_Running_LED

5.1.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use GPIO control the LED
- Learn to use SysTick to generate 1ms delay

GD32C113C-START board has two keys and one LED. The two keys are Reset key and Wakeup key. The LED1 is controlled by GPIO. This demo will show how to light the LED.

5.1.2. DEMO running result

Download the program < 01_GPIO_Running_LED > to the START board, LED1 will turn on and off in sequence with interval of 1000ms, repeat the process.

5.2. GPIO_Key_Polling_mode

5.2.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use GPIO control the LED and the KEY
- Learn to use SysTick to generate 1ms delay

GD32C113C-START board has two keys and one LED. The two keys are Reset key and Wakeup key. The LED1 is controlled by GPIO.

This demo will show how to use the Wakeup key to control the LED1. When press down the Wakeup Key, it will check the input value of the IO port. If the value is 1 and will wait for 100ms. Check the input value of the IO port again. If the value still is 1, it indicates that the button is pressed successfully and toggle LED1.

5.2.2. DEMO running result

Download the program < 02_GPIO_Key_Polling_mode > to the START board, press down the Wakeup Key, LED1 will be turned on. Press down the Wakeup Key again, LED1 will be turned off.

5.3. GPIO_Key_Interrupt_mode

5.3.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use GPIO control the LED and the KEY
- Learn to use EXTI to generate external interrupt

GD32C113C-START board has two keys and one LED. The two keys are Reset key and Wakeup key. The LED1 is controlled by GPIO.

This demo will show how to use the EXTI interrupt line to control the LED1.When press down the Wakeup Key, it will produce an interrupt. In the interrupt service function, the demo will toggle LED1.

5.3.2. DEMO running result

Download the program < 03_EXTI_Key_Interrupt_mode > to the START board. Press down the Wakeup Key, LED1 will be turned on. Press down the Wakeup Key again, LED1 will be turned off.

5.4. USART_Hyperterminal_Interrupt

5.4.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the USART transmit and receive interrupts to communicate with the hyperterminal

5.4.2. DEMO running result

Download the program <04_USART_HyperTerminal_Interrupt> to the START board and run. Firstly, the COM sends the “USART interrupt test” to the hyperterminal and waits for receiving data from the hyperterminal that you must send. The string that you have sent is stored in the receiver_buffer array. The receive buffer have a receivesize=32 bytes as maximum. After that, compare rxcount with receivesize. If rxcount is same with receivesize, the COM sends the “USART receive successfully!” to the hyperterminal.

The information via a serial port output as following:

```
USART interrupt test


USART receive successfully!
```

5.5. TIMER_Key_EXTI

5.5.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use GPIO control the LED and the KEY
- Learn to use EXTI to generate external interrupt
- Learn to use TIMER to generate PWM

This demo will show how to use the TIMER PWM to trigger EXTI interrupt to toggle the state of LED2 and EXTI interrupt line to control the LED1. When press down the User Key, it will produce an interrupt. In the interrupt service function, the demo will toggle LED1.

5.5.2. DEMO running result

Download the program < 05_TIMER_Key_EXTI > to the START board, LED1 is flashed once for test, press down the Wakeup Key, LED1 will be turned on. Press down the Wakeup Key again, LED1 will be turned off. Connect PA6 (TIMER2_CH0) and PA4 with DuPont line. The LED1 will be toggled every 500ms.

5.6. USB_MSC_Device

5.6.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn how to use the USBFS
- Learn how to implement USB MSC (mass storage) device

This demo mainly implements a U disk. U disk is currently very widely used removable MSC devices. MSC, the Mass Storage Device Class, is a transport protocol between a computer and mobile devices, which allow a universal serial bus (USB) equipment to access a host computing device, file transfer between them, mainly including mobile hard disk, mobile U disk drive, etc. The MSC device must have a storage medium, and this demo uses the MCU's internal flash as the storage medium. For more details of the MSC protocol please refer to the MSC protocol standard.

MSC device will use a variety of transport protocols and command formats for communication, so it need to choose the appropriate protocol and command format in the realization of the application. This demo selects the BOT (bulk only transport) protocol and the required SCSI (small computer interface) command, and is compatible with a wide variety of Window operating systems. Specific BOT protocol and SCSI command specification please refer to the standard of their agreement.

5.6.2. DEMO running result

Download the program < 06_USB_MSC_Device > to the START board and run. When the EV-board connect to the PC, user will find a USB large capacity storage device is in the universal serial bus controller, and there is one more disk drives in the device manager of PC.

Then, after opening the resource manager, you will see more of the 1 disk, as shown in the following diagram:

At this point, the write/read/formatting operation can be performed as the other mobile devices.

5.7. USB_HID_Host

5.7.1. DEMO purpose

This demo includes the following functions of GD32 MCU:

- Learn to use the USBFS as a HID host
- Learn the operation between the HID host and the mouse device
- Learn the operation between the HID host and the keyboard device

GD32C113C-START board integrates the USBFS module, and the module can be used as a USB device, a USB host or an OTG device. This demo mainly shows how to use the USBFS as a USB HID host to communicate with external USB HID device.

5.7.2. DEMO running result

Download the program < 07_USB_HID_Host > to the START board and run.

If a mouse has been attached, the user will see the information of mouse enumeration. First pressing the Wakeup key will see the inserted device is mouse, and then moving the mouse will show the position of mouse in the HyperTerminal.

If a keyboard has been attached, the user will see the information of keyboard enumeration. First pressing the Wakeup key will see the inserted device is keyboard, and then pressing the keyboard will show the state of the button in the HyperTerminal.

6. Revision history

Table 6-1. Revision history

Revision No.	Description	Date
1.0	Initial Release	Aug.19, 2022

Important Notice

This document is the property of GigaDevice Semiconductor Inc. and its subsidiaries (the "Company"). This document, including any product of the Company described in this document (the “Product”), is owned by the Company under the intellectual property laws and treaties of the People’s Republic of China and other jurisdictions worldwide. The Company reserves all rights under such laws and treaties and does not grant any license under its patents, copyrights, trademarks, or other intellectual property rights. The names and brands of third party referred thereto (if any) are the property of their respective owner and referred to for identification purposes only.

The Company makes no warranty of any kind, express or implied, with regard to this document or any Product, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Company does not assume any liability arising out of the application or use of any Product described in this document. Any information provided in this document is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Except for customized products which has been expressly identified in the applicable agreement, the Products are designed, developed, and/or manufactured for ordinary business, industrial, personal, and/or household applications only. The Products are not designed, intended, or authorized for use as components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, atomic energy control instruments, combustion control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or Product could cause personal injury, death, property or environmental damage ("Unintended Uses"). Customers shall take any and all actions to ensure using and selling the Products in accordance with the applicable laws and regulations. The Company is not liable, in whole or in part, and customers shall and hereby do release the Company as well as it’s suppliers and/or distributors from any claim, damage, or other liability arising from or related to all Unintended Uses of the Products. Customers shall indemnify and hold the Company as well as it’s suppliers and/or distributors harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of the Products.

Information in this document is provided solely in connection with the Products. The Company reserves the right to make changes, corrections, modifications or improvements to this document and Products and services described herein at any time, without notice.

© 2022 GigaDevice – All rights reserved