

GigaDevice Semiconductor Inc.

GD32A508V-START

Arm[®] Cortex[®]-M33 32-bit MCU

User Guide

Revision 1.0

(Nov. 2023)

Tables of Contents

TABLES OF CONTENTS	1
LIST OF FIGURES	3
LIST OF TABLES	4
1. SUMMARY	5
2. FUNCTION PIN ASSIGN	5
3. GETTING STARTED	5
4. HARDWARE LAYOUT OVERVIEW	6
4.1. Power supply	6
4.2. Boot option	6
4.3. LED	6
4.4. KEY	7
4.5. USART	7
4.6. USB	7
4.7. Extension	8
4.8. GD-Link	8
4.9. MCU	9
5. ROUTINE USE GUIDE	10
5.1. GPIO_Running_LED	10
5.1.1. DEMO purpose	10
5.1.2. DEMO running result	10
5.2. GPIO_Key_Polling_mode	10
5.2.1. DEMO purpose	10
5.2.2. DEMO running result	10
5.3. EXTI_Key_Interrupt_mode	11
5.3.1. DEMO purpose	11
5.3.2. DEMO running result	11
5.4. USART_HyperTerminal_Interrupt	11
5.4.1. DEMO purpose	11
5.4.2. DEMO running result	11
5.5. TIMER_Key_EXTI	12
5.5.1. DEMO purpose	12
5.5.2. DEMO running result	12
5.6. USB MSC Device	12
5.6.1. DEMO purpose	12
5.6.2. DEMO Running Result	13
5.7. USB HID Host	13
5.7.1. DEMO Purpose	13

5.7.2.	DEMO Running Result	14
6.	REVISION HISTORY	15

List of Figures

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

List of Tables

Table 2-1. Function pin assignment.....	5
Table 6-1. Revision history	15

1. Summary

GD32A508V-START uses GD32A508VET7 as the main controller. It uses GD-Link Mini USB interface to supply 5V power. Reset, Boot, K2-User Key, LEDs, USB and USART to USB interface are also included. For more details please refer to GD32A508V-START-Rev1.1 schematic.

2. Function Pin Assign

Table 2-1. Function pin assignment

Function	Pin	Description
LED	PC6	LED1
	PC7	LED2
	PC8	LED3
	PC9	LED4
RESET		K1-Reset
KEY	PA0	K2
USART	PD8	USART2_TX
	PD9	USART2_RX
USB	PA9	USB_VBUS
	PA11	USB_DM
	PA12	USB_DP
	PD9	USB Pull-up Control

3. Getting started

The START board uses GD-Link Mini USB connector 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 the power supply is OK.

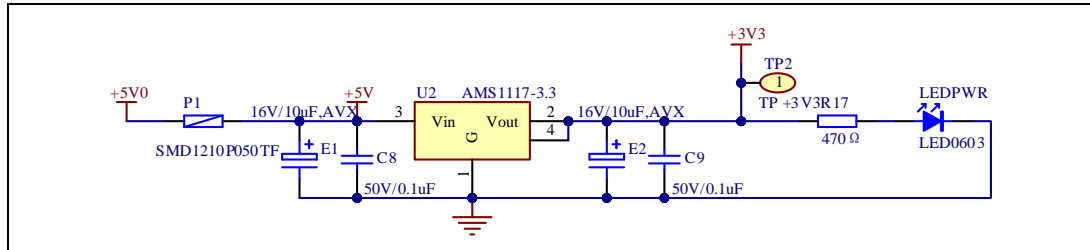
There are Keil version and IAR version of all projects. Keil version of the projects are created based on Keil MDK-ARM 5.28 uVision5. IAR version of the projects are created based on IAR Embedded Workbench for ARM 8.32.1. During use, the following points should be noted:

1. If you use Keil uVision5 to open the project. In order to solve the "Device Missing (s)" problem, you can install GigaDevice.GD32A508_DFP.1.0.0.pack.
2. If you use IAR to open the project, install IAR_GD32A508_ADDON_1.0.0.exe to load the associated files.

4. Hardware layout overview

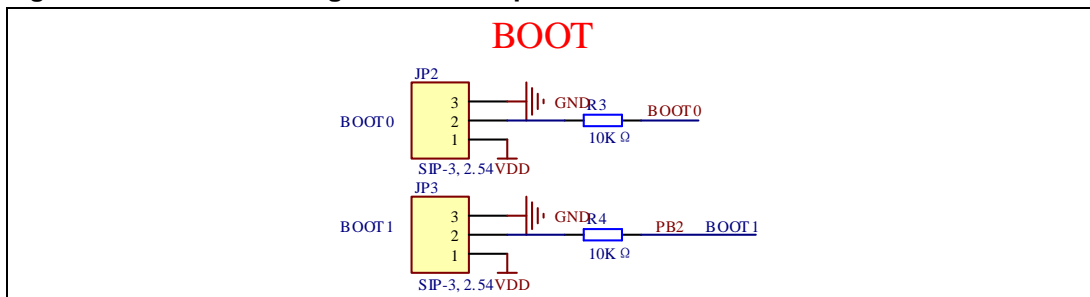
4.1. Power supply

Figure 4-1. Schematic diagram of power supply



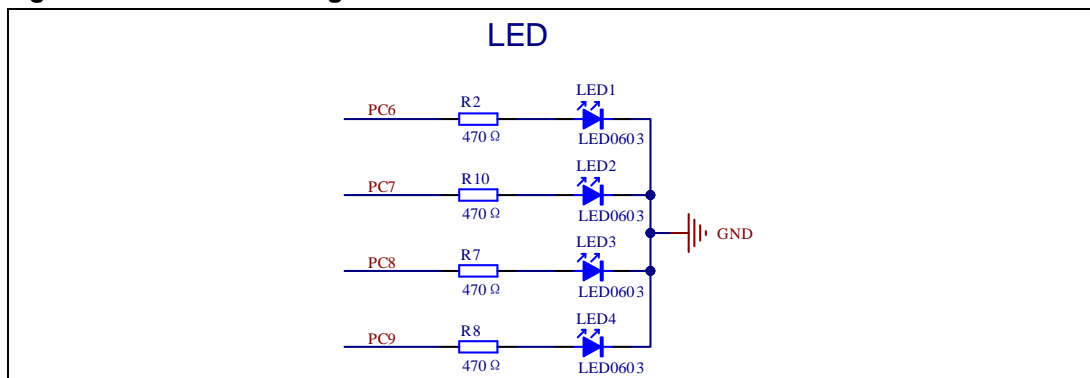
4.2. Boot option

Figure 4-2. Schematic diagram of boot option



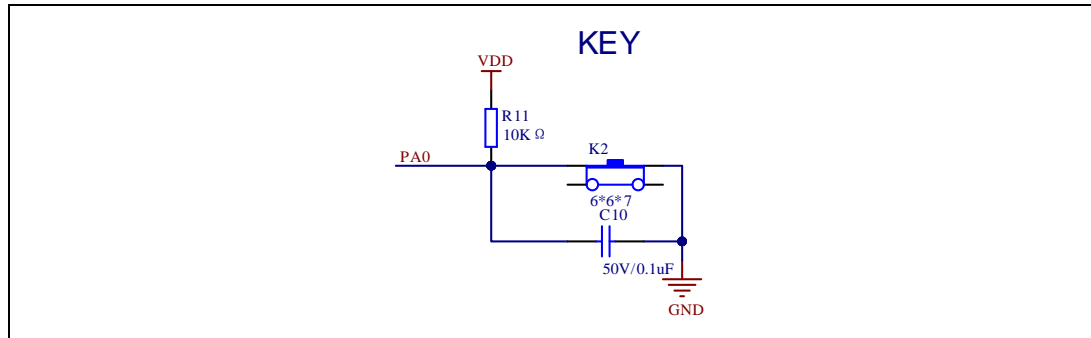
4.3. LED

Figure 4-3. Schematic diagram of LED function



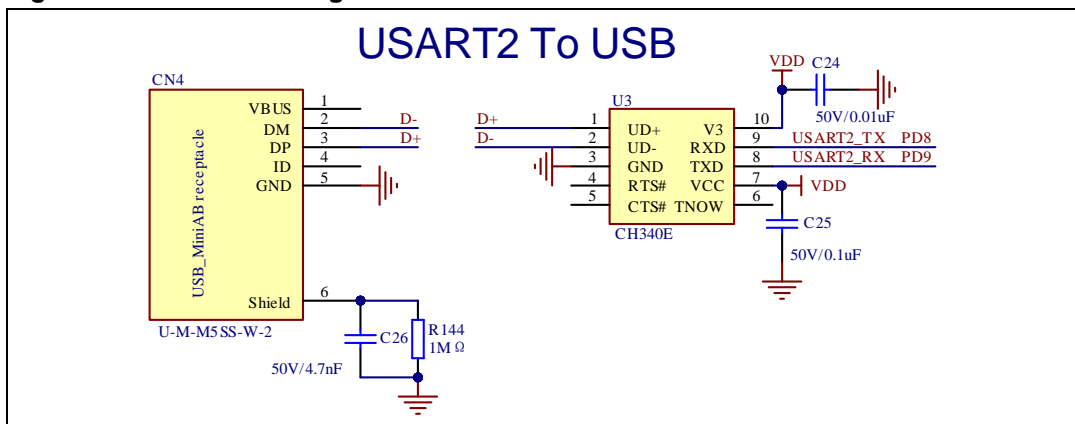
4.4. KEY

Figure 4-4. Schematic diagram of Key function



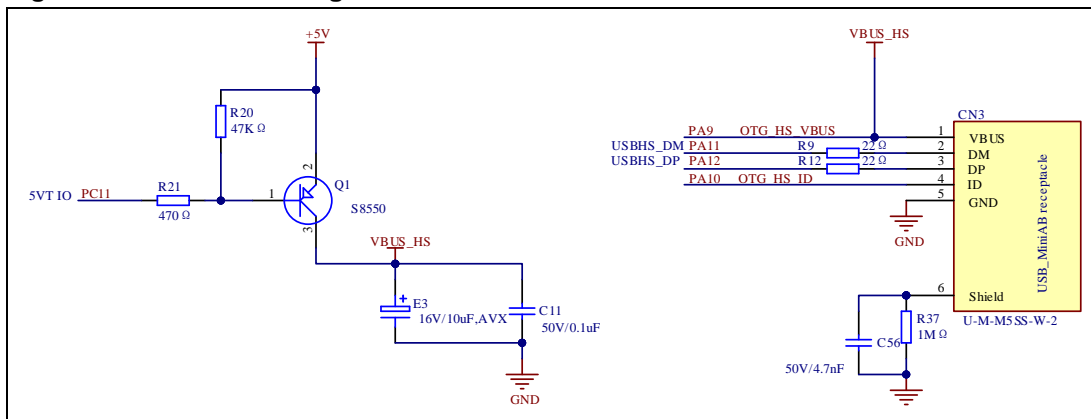
4.5. USART

Figure 4-5. Schematic diagram of USART



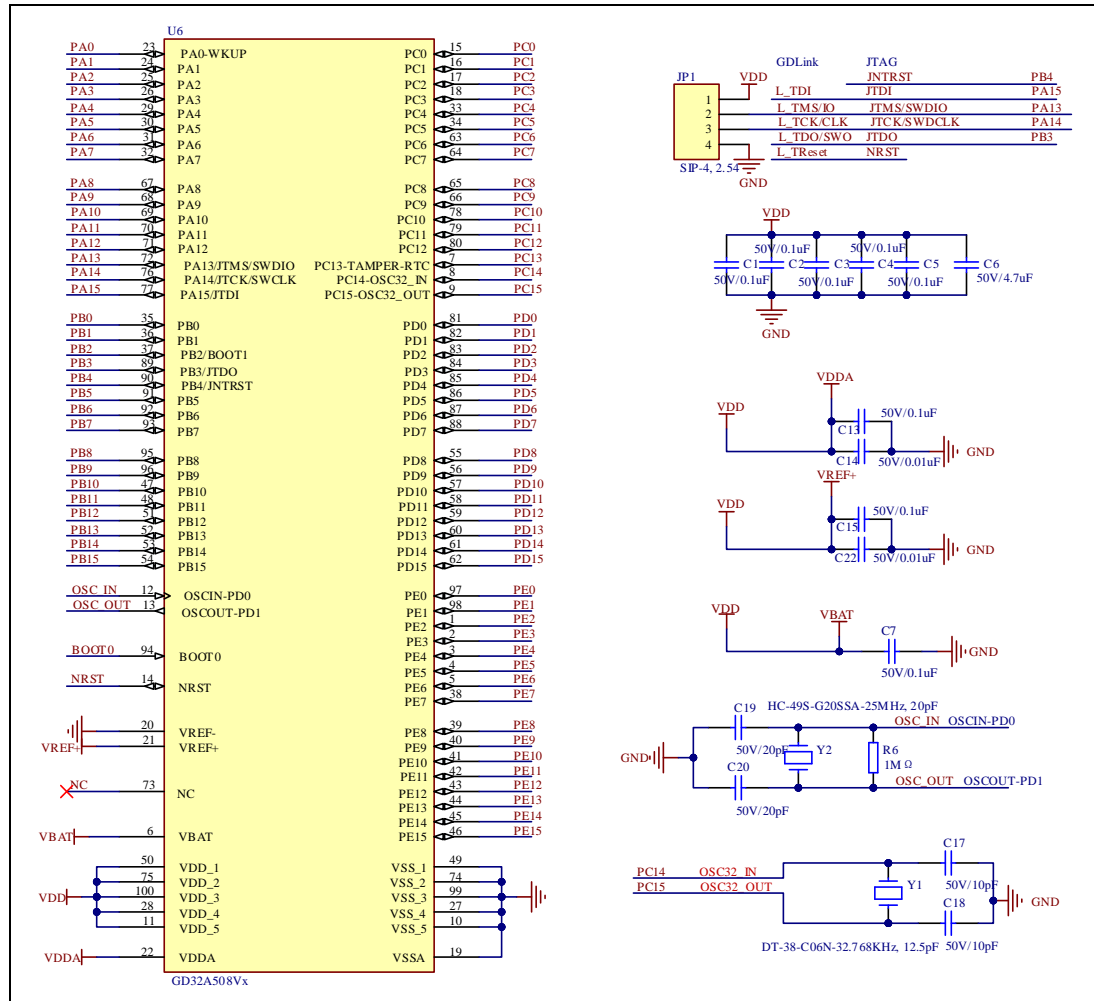
4.6. USB

Figure 4-6. Schematic diagram of USB



4.9. MCU

Figure 4-9. Schematic diagram of MCU



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

GD32A508V-START-V1.1 board has 2 keys and 4 LED. The keys are User Key and Reset Key. The LED 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 can light cycles.

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

GD32A508V-START-V1.1 board has 2 keys and 4 LED. The keys are User Key and Reset Key. The LED are controlled by GPIO.

This demo will show how to use the User Key to control the LED2. When press down the User Key, it will check the input value of the IO port. If the value is 0 and will wait for 100ms. Check the input value of the IO port again. If the value still is 0, it indicates that the button is pressed successfully and toggle LED2.

5.2.2. DEMO running result

Download the program < 02_GPIO_Key_Polling_mode > to the START board, press down the User Key, LED2 will be turned on. Press down the User Key again, LED2 will be turned off.

5.3. EXTI_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.

GD32A508V-START-V1.1 board has 2 keys and 1 LED. The keys are K2-User Key and Reset Key. The LED is controlled by GPIO.

This demo will show how to use the EXTI interrupt line to control the LED2. When press down the K2-User Key, it will produce an interrupt. In the interrupt service function, the demo will toggle LED2.

5.3.2. DEMO running result

Download the program < 03_EXTI_Key_Interrupt_mode > to the START board, LED2 is turned on and off for test. When press down the K2-User Key, LED2 will be turned on. Press down the K2-User Key again, LED2 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, connect serial cable to USART. Firstly, the LED1 flash 2 times for test. Then, the USART sends the tx_buffer array (from 0x00 to 0xFF) 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 rx_buffer array. The receive buffer have a BUFFER_SIZE bytes as maximum. After that, compare tx_buffer with rx_buffer. If tx_buffer is same with rx_buffer, LED1 flash. Otherwise, LED1 turn on.

The output information via the HyperTerminal is as following:

```

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A
1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35
36 37 38 39 3A 3B 3C 3D 3E 3F 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50
51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F 60 61 62 63 64 65 66 67 68 69 6A 6B
6C 6D 6E 6F 70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F 80 81 82 83 84 85 86
87 88 89 8A 8B 8C 8D 8E 8F 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F A0 A1
A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC
BD BE BF C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF D0 D1 D2 D3 D4 D5 D6 D7
D8 D9 DA DB DC DD DE DF E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF F0 F1 F2
F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF

```

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

GD32A508V-START-V1.1 board has 2 keys and 4 LED. The keys are User Key and Reset Key. The LED are controlled by GPIO.

This demo will show how to use the TIMER PWM to trigger EXTI interrupt to toggle the state of LED1 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, the LED1 is flashed once for test, press down the User Key, LED1 will be turned on. Press down the User 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 USBHS
- 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 SRAM 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 start board is connected to the PC, you will find a USB large capacity storage device is in the universal serial bus controller, and there is 1 more disk drives in the equipment 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 USBHS 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

GD32A508V-START board integrates the USBHS 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 USBHS 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 K2 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 K2 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	Nov.29, 2023

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 its 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 its 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.