swissbit®

Application Note

AN2109en

Power Failure Testing

swissbit®

Contents

1 Abstract

NAND flash is a technology found in Solid State Drives (SSDs) that enables a significantly higher data rate compared to Hard Disk Drives (HDDs). In particular, random read and write accesses are faster with NAND flash. However, in the event of a sudden power failure, hard disks are able to cleanly finish the started write process by recovering the rotational energy of the disk stack, supplying the electronics with the required voltage for a short moment. With SSDs, large capacitors can be used to support the supply voltage for a few hundred milliseconds until the internal buffers and the DRAM cache have been written to the flash. The disadvantage here is the high cost of the capacitors, for which only tantalum electrolytic capacitors are viable based on the industrial temperature range and the available space. Therefore, such SSDs are only used for special applications. Storage media that can still write all data in the cache to the flash in the event of a power failure carry the label *PLP* (*Power Loss Protection*).

Because PLP media are protected against sudden power loss, the problems and test possibilities of media without PLP are considered in this document. However, the methods can also be applied to PLP media to verify the advertised properties.

2 Dangers in the event of a sudden power failure

A sudden power failure can lead to various errors and problems with flash media, resulting in a total failure, which is the worst case. The cause lies in the technology: All flash is divided into so-called *blocks*. These blocks, in turn, consist of *pages*. While only whole pages can be programmed in one operation, erasing can only occur on whole blocks. This results in SSDs, unlike HDDs, having no fixed mapping between logical addresses (seen by the host) and physical flash addresses. In addition, the number of programming and erase cycles is limited; so, *Wear Leveling* is employed to ensure even wear of the blocks, which, of course, is only possible if the address mapping is not fixed.

Because only whole pages, which have a size of up to 64 KiB (depending on the flash), can be programmed, this size must still be multiplied by the number of flashes connected in parallel. This means up to 1 MiB of flash cells are programmed with each write access – even if the operating system only updates one directory entry.

To reduce unnecessary flash wear, the SSD buffers the data in the controller or in an external DRAM for a few milliseconds. If more data arrives during this time, the SSD can write the data to the flash with the same write access. Because the flash is generally managed in segments of 4 KiB each, 16 individual write accesses fit into a page of 64 KiB if each write access does not exceed 4 KiB. Efficiency thus increases if the controller delays the write accesses and more data arrives before the programming process is started in the flash. The disadvantage, of course, is that this data is lost if there is a sudden power failure and the device does not have PLP.

Most applications can tolerate this behavior since the affected data remained in the

RAM of the host a few milliseconds earlier and would have been lost if the power failure had occurred while the data was in RAM.

In case of an unexpected power failure, however, the following problems, listed in ascending order of severity, may occur:

1. User data that the host has transmitted to the medium and whose receipt has already been acknowledged by the medium, although it has not yet been written to the flash, is lost. This is not a problem for many use cases, as described earlier. Sensitive systems, where no data may be lost, have an uninterruptible power supply, which then also covers the storage media.

2. Not only the data that was just transferred is lost, but also data that was written with the immediately preceding write accesses is lost. This happens even if these write operations were carried out a long time ago, but no power cycle has taken place since then. Due to the technology, this effect can occur with older Multi-Level Cell (MLC) and Tripe-Level Cell (TLC) memories in floating gate technology.

3. Data incompletely written during a power failure typically cannot be recovered by the error correction because the programming process was not completed. Failure of error correction incorrectly causes the firmware to judge that it is a bad flash block, which results in the block being replaced with one from the spare pool. If this happens too often, all reserve blocks are used up until the medium is no longer writable.

4. The so-called *reordering* occurs. There is subsequently no clean transition between the new and the old data when reading the logical addresses in the identical order as before when writing. This can lead to massive data loss in databases and file systems with a journal.

5. Static data (i.e., data that hasn't been written in a while or has remained since the last power cycle) is lost. A typical example is the corruption of operating system files, causing the system to suddenly fail to boot.

6. The media's firmware or its meta data has been corrupted. This can manifest itself through various symptoms: massive or total data loss; corrupted lifetime data (S.M.A.R.T.); or a permanent loss of performance that may not be noticed. In the worst case, the host no longer identifies the medium.

Swissbit has been a market leader in hardening flash media against sudden power failures for many years. Accordingly, only the first two cases have occurred with our products, but none occur with PLP media.

3 Power failure testing

If the robustness against sudden power failures is to be tested as part of the product qualification, the provoked failure can be carried out for the entire system or only for the storage medium. It is advisable to start with the storage medium alone, since much faster cycles are possible, in order to make a preselection. If the storage medium has successfully reached the target number of cycles (e.g., 10,000 cycles), at least another 1,000 cycles are completed in the overall system. These are used to find possible compatibility problems, such as errors in the BIOS. Particularly under extreme test conditions, a prolonged diagnostic phase of the medium can occur during the next power-on, in which data consistency is restored. However, some BIOS versions have greatly reduced the maximum waiting time with the replacement of HDDs with SSDs in the last few years. This, which is contrary to the recommendation in the SATA/ATAPI standard, can now lead to a timeout with newer SSDs with TLC or Quadruple Level Cell (QLC) flash and their complexity in internal management. As in an unfavorable case, the startup time can again be in the range of HDDs.

As previously described, the flash medium should not be damaged by a sudden shutdown. A good firmware must be able to prevent permanent errors or increased wear. However, this is only true as long as the number and speed at which the cycles occur is still within a realistic range. In qualification tests, many years of field operation are often simulated in just a few days. If this results in a very rapid sequence of extremely large numbers of cycles, in which a volume of data greater than zero but smaller than a few megabytes is written, then – depending on the size of the storage medium – the storage medium can fail after a few tens of thousands of cycles. The problem occurs mainly when the cycle time is only a few seconds. The background is that with each cycle, a new block is written, but none of these blocks is ever completely filled. If a block is programmed extremely often and quickly only partially, this can lead to insufficient erase operations, as a result of which the reprogramming has an ever increasing bit error rate until data loss occurs. This problem can be avoided by completely writing the medium once every 3,000 cycles. Alternatively, the amount of data written during one cycle should be at least 0.05 % of the total drive capacity.

3.1 USB

Testing USB devices is very simple. Only the +5 Volt supply line must be disconnected and provided with a switch. Although there are also USB hubs that have switchable outputs, these are difficult to obtain. Often it is only possible to disconnect the end device from the bus, but then the power supply remains.

If the +5 Volt line is interrupted, the end device is logged off from the host. As soon as the supply is restored, the device is recognized and initialized by the operating system.

3.2 CompactFlash

With all interfaces that do not work with differential signals, there is a risk of continuing to supply the storage medium with current via the data lines even if the actual supply has been interrupted. The current flows via the data lines to the storage medium and via the ESD protection diodes to the distribution of the power supply. Figure 1 shows the current flow after disconnection of the positive supply voltage in red dashed lines.

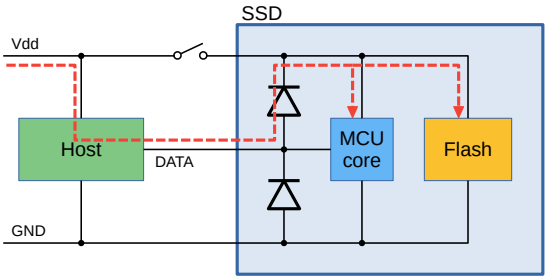

Figure 1: Drive powered via ESD diodes

The resulting supply voltage in the storage medium is then usually too low for normal operation. The controller detects the low supply voltage and pauses all further operations. When the supply voltage returns to nominal level, the controller continues with its operations. Since the controller's internal states are not lost in the process, however, this is not a true power failure.

It would be very complex to set up a test environment that separates all data and control lines in the case of CompactFlash. Therefore, it is a good idea to change the interface to the host. There are two possibilities: Either a CF card reader is used, which is connected via USB. Then the procedure is the same as it is with a USB storage medium. Or, an IDE-SATA bridge is employed, and the CF medium is used like a SATA medium. With a bridge, it is important to ensure the bridge is disconnected from the power supply together with the CF card because IDE devices are not hot-pluggable, unlike SATA devices.

3.3 SD cards

The same problem occurs with SD cards as with CF cards. The data and control lines are provided with pull-up resistors through which

sufficient current can flow to supply the controller when the positive voltage supply is disconnected. Therefore, it is important to disconnect positive voltage before it reaches the pull-up resistors. Alternatively, SD card readers can also be used and tested like USB devices.

3.4 eMMC

The interface of eMMC and SD is largely identical. Electrically, they are completely compatible. Accordingly, there is also the risk of power supply via the pull-up resistors. If the robustness of an eMMC is not to be checked in the finished system first, adapters from eMMC to SD are also available. With these adapters, the eMMC can either be soldered on or placed in a terminal socket. The adapter is then connected via USB using the SD card reader, and the positive voltage supply is disconnected in the USB cable.

3.5 SATA and CFast

CFast also uses the SATA interface. Like USB, SATA is hot-pluggable and uses differential signal transmission. The signals are DC-free because both the host and the storage medium have series capacitors in the lines. Thus, it is sufficient to switch only the positive supply voltage. The host usually offers two voltages to the different SATA form factors. Which one has to be switched can be found in the data sheet. It is, however, usually the lower voltage. There are also media that can be supplied with two voltages. In such a case, one line is permanently cut.

It is recommended to use an adapter to 2.5" for all small form factors. With the 12-pin connector, the red 5 Volt line must then be switched and the yellow 12 Volt line disconnected for safety.

If the medium is not recognized, the hot-plug option for SATA might have to be activated in the BIOS.

3.6 NVMe and CFexpress

Testing is somewhat more complex for storage media with a PCIe interface. This is due to the high transfer rates, which make it difficult to spatially remove the storage medium to disconnect the voltage supply. In addition, the hosts or the operating systems sometimes still have errors in the implementation of the hot swap function.

There are ready-made solutions for spatial separation of the storage medium, which allow the remote connection of m.2 or PCIe cards with a high-quality cable connection. The advantage is that there is no loss of performance. However, this solution is not recommended for power failure testing because the effort to set up a reliable test environment is very high. In principle, NVMe is hot-swappable. This means that a storage medium, which was connected at system start and for which the necessary system resources were thus allocated, can be removed and also connected again at a later time. What is supposed to work in theory, however, still shows a number of problems in practice. It depends on the operating system, the host chipset, the BIOS, and even the PCIe slot on the motherboard whether the storage medium is recognized again after a hot swap, and, if so, how often. Our experience is that the combination of Intel chipsets, Linux operating system, and server mainboards works best. But even with that, after a few thousand cycles, the storage medium may no longer be recognized, requiring a system reboot.

It, therefore, makes sense to use USB bridges for NVMe devices. However, even these do not work perfectly. In the Swissbit lab, for example, it happened several times that the storage medium was not recognized every few thousand cycles, even though it was recognized without any problems during the next cycle without restarting the host. This behavior would then only have to be taken into account in the test software.

Only the power supply via USB from the host can be a problem with this solution, because the storage medium can draw more (peak) current than the host provides. To avoid this prob-

lem, the +5 Volt line from the power supply can be directly connected (switched) to the USB bridge.

4 Test hardware

Ready-made hardware for testing power failures is hardly available. There are relays that can be controlled via USB, but most of them have complicated and outdated APIs, which makes it difficult to integrate them into current operating systems. Therefore, a simple build proposal is presented below as a solution.

The core element is an Atmel microcontroller with a USB connection. In this example, an Atmel-ATmega32U4 is installed on an Arduino-compatible board called "ProMicro". This board can be programmed with the Arduino IDE if "Leonardo" is selected as the board used. In principle, any Arduino board can be used that has a real USB interface, such as the Arduino Uno, Arduino Nano, or Arduino Micro. If necessary, the initialization of the USB interface must be modified. Variants that only emulate the USB interface in software often cause problems, such as boards with an ATtiny.

Figure 2 shows the schematic. The board is connected to the host via USB; powered and controlled by the host.

The microcontroller receives the control commands from the host and switches pin 17 (internal "A0") to 5 Volt or 0 Volt accordingly. The series resistor R1 limits the base current of Q1. The resistor R3 blocks the P-channel MOSFET Q2 until Q1 conducts. R2 is only used to reliably block Q1 as long as the USB connection to the microcontroller is not yet established or port A0 is not yet initialized. A standard NPN transistor can be used for Q1. For Q2, IRLML6402 was chosen because this type with $V_{GS(th),max} = -1.2V$ reliably switches a supply voltage of 3.3 V.

The positive voltage supply of the SSD is disconnected and connected to J1 according to the circuit diagram. If the SSD and the microcontroller are connected to different hosts, a common ground potential must be established so that Q2 can switch.

Under Linux, the switching is done with the following commands:

```
echo '1' > /dev/ttyACM0
echo '0' > /dev/ttyACM0
```

A '1' will turn on the power supply, a '0' will turn it off. Before that, however, the program must be written to the microcontroller using the Arduino IDE. The program is kept very simple:

```
void setup() {
  pinMode(A0, OUTPUT);
  digitalWrite(A0, false);
  SerialUSB.begin();
}

void loop() {
  if (SerialUSB.available()) {
    char in = SerialUSB.read();
    if (in == '0') {
      digitalWrite(A0, false);
    } else if (in == '1') {
      digitalWrite(A0, true);
    }
  }
}
```

At the beginning, port A0 is defined as output and switched to 0 Volt. After that, the USB interface is initialized. Then, the microcontroller waits for the reception of a character via USB. If a '0' is received, the output is switched to 0 Volts, which blocks Q1 and Q2 and disconnects the SSD from the supply voltage. If a '1' is received, the output is switched to 5 Volts, Q1 and Q2 conduct, and the SSD starts. The finished setup is shown in figure 3.

A concrete construction proposal for an extension to switch the mains voltage of the host is omitted here. This is easy to do with a suitable relay, but should only be done by a qualified electrician.

5 Conclusion

The presented methods and the circuit provide a simple solution for implementing power supply failure tests. They can be used to well time

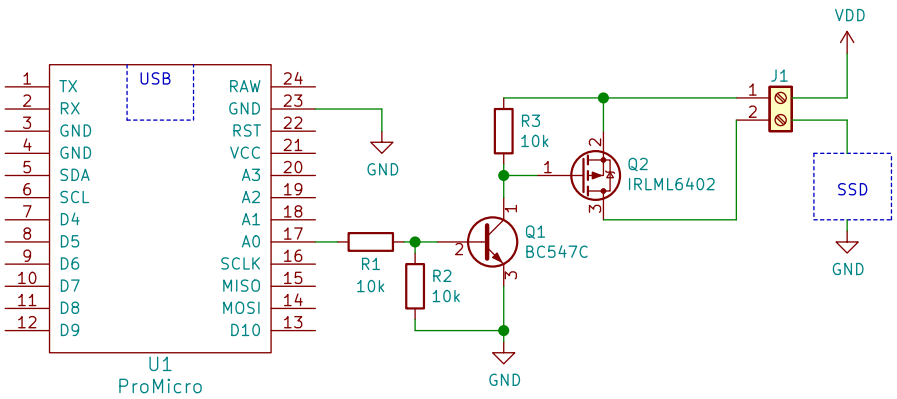

Figure 2: Schematic

failures during write and read accesses of the target application to achieve wide coverage during different operating states. The robustness and the suitability of a medium can be verified for the respective application in order to approve manufacturers for the final product that do not pose an incalculable risk of failure in the field.

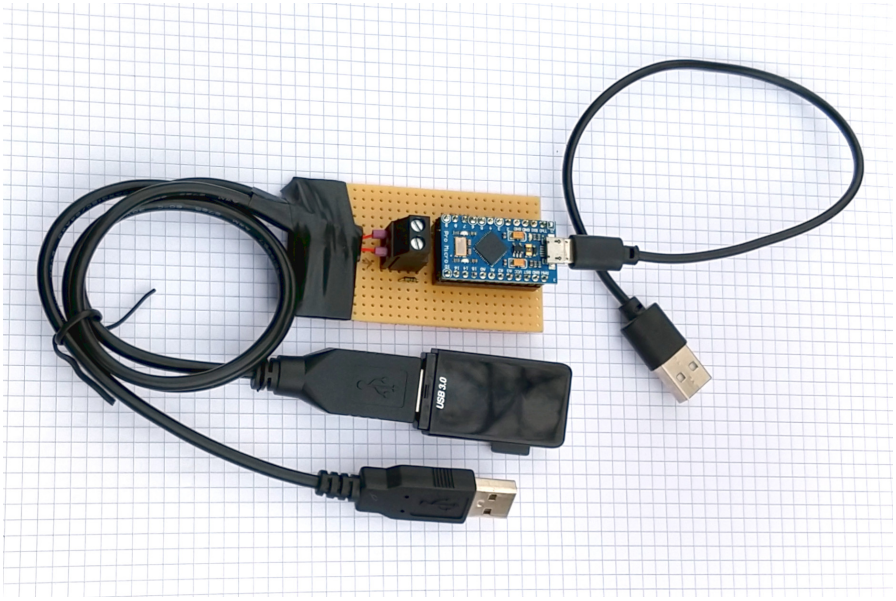

Figure 3: Setup with (micro)SD Card Reader

CONTACT US

Headquarters	**Swissbit AG** Industriestrasse 4 9552 Bronschhofen Switzerland	Tel. +41 71 913 03 03 sales@swissbit.com
Germany (Berlin)	**Swissbit Germany AG** Bitterfelder Strasse 22 12681 Berlin Germany	Tel. +49 30 936 954 0 sales@swissbit.com
Germany (Munich)	**Swissbit Germany AG** Leuchtenbergring 3 81677 Munich Germany	Tel. +49 30 936 954 400 sales@swissbit.com
North and South America	**Swissbit NA Inc.** 238 Littleton Road, Suite 202B Westford, MA 01886 USA	Tel. +1 978-490-3252 salesna@swissbit.com
Japan	**Swissbit Japan Co., Ltd.** CONCIERIA Tower West 2F 6-20-7 Nishishinjuku Shinjuku City, Tokyo 160-0023 Japan	Tel. +81 3 6258 0521 sales-japan@swissbit.com
Taiwan	**Swissbit Taiwan** 3F., No. 501, Sec.2, Tiding Blvd. Neihu District, Taipei City 114 Taiwan, R.O.C.	Tel. +886 912 059 197 salesasia@swissbit.com
China	**Swissbit China**	Tel. +886 958 922 333 salesasia@swissbit.com

Disclaimer:
The information in this document is subject to change without notice. Swissbit AG ("SWISSBIT") assumes no responsibility for any errors or omissions that may appear in this document, and disclaims responsibility for any consequences resulting from the use of the information set forth herein. SWISSBIT makes no commitments to update or to keep current information contained in this document. The products listed in this document are not suitable for use in applications such as, but not limited to, aircraft control systems, aerospace equipment, submarine cables, nuclear reactor control systems and life support systems. Moreover, SWISSBIT does not recommend or approve the use of any of its products in life support devices or systems or in any application where failure could result in injury or death. If a customer wishes to use SWISSBIT products in applications not intended by SWISSBIT, said customer must contact an authorized SWISSBIT representative to determine SWISSBIT willingness to support a given application. The information set forth in this document does not convey any license under the copyrights, patent rights, trademarks or other intellectual property rights claimed and owned by SWISSBIT.

ALL PRODUCTS SOLD BY SWISSBIT ARE COVERED BY THE PROVISIONS APPEARING IN SWISSBIT'S TERMS AND CONDITIONS OF SALE ONLY, INCLUDING THE LIMITATIONS OF LIABILITY, WARRANTY AND INFRINGEMENT PROVISIONS. SWISSBIT MAKES NO WARRANTIES OF ANY KIND, EXPRESS, STATUTORY, IMPLIED OR OTHERWISE, REGARDING INFORMATION SET FORTH HEREIN OR REGARDING THE FREEDOM OF THE DESCRIBED PRODUCTS FROM INTELLECTUAL PROPERTY INFRINGEMENT, AND EXPRESSLY DISCLAIMS ANY SUCH WARRANTIES INCLUDING WITHOUT LIMITATION ANY EXPRESS, STATUTORY OR IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.