

swissbit[®]

Application Note

AN2103de

Performance tests

© Swissbit AG 2024

  Creative-Commons-Lizenz¹

¹ Dieses Werk steht unter der Creative-Commons-Lizenz „Namensnennung 4.0 International“. Um eine Kopie dieser Lizenz zu sehen, besuchen Sie <http://creativecommons.org/licenses/by/4.0/>

Inhaltsverzeichnis

- 1 Einleitung
- 2 Grundlagen
- 3 Schreibgeschwindigkeit
- 4 Pseudo-SLC-Cache
- 5 Direct-Write
- 6 Lesegeschwindigkeit
- 7 Zusammenfassung

1 Einleitung

Dieses Dokument betrachtet die Geschwindigkeit beim Schreiben und Lesen von NAND-Flash-Speichermedien in Abhängigkeit des Zugriffsmusters sowie den Einfluss des zuvor geschriebenen Musters.

Dies soll den Anwender bei der Wahl von geeigneten Tests unterstützen, um seine eigenen Anforderungen an das Speichermedium zu überprüfen. Zum besseren Verständnis dieses Dokuments sollte die prinzipielle Arbeitsweise des Garbage-Collectors bekannt sein.

2 Grundlagen

Im Gegensatz zu Festplatten können NAND-Flash-Speichermedien beim Schreiben Echtzeitanforderungen im Millisekunden-Bereich häufig nicht erfüllen. Dies resultiert hauptsächlich aus dem mit höherer Priorität laufenden „Garbage-Collector“, dessen Last nur schwer vorhersagbar ist. Daher muss bei harten Echtzeitanforderungen immer ein entsprechend großer Cache im Betriebssystem eingesetzt werden.

Während bei Festplatten eine feste Zuordnung existiert zwischen den physikalischen Adressen des Speichers und logischen Adressen, die das Betriebssystem verwendet, so gibt es diese Verbindung bei NAND-Flash-Speichermedien nicht. Wird mehrfach auf die

gleiche logische Adresse geschrieben, so werden die Daten jeweils auf einer anderen physikalischen Adresse abgelegt. Der Zusammenhang zwischen logischen und physikalischen Adressen wird von der Firmware in Tabellen verwaltet, die ebenfalls im NAND-Flash gespeichert werden.

Die Art der Daten, die bei einem Test auf das Speichermedium geschrieben werden, beeinflussen die Geschwindigkeit nicht. Während in der Anfangszeit von NAND-Flash noch Controller mit eingebauter Datenkompression zum Einsatz kamen, sind diese heute nicht mehr verbreitet. Entsprechend kann auf die rechenintensive Erzeugung von Pseudo-Zufallsdaten verzichtet werden, die die Geschwindigkeitsmessung negativ beeinflussen könnte.

3 Schreibgeschwindigkeit

Abbildung 1 zeigt die Schreibgeschwindigkeit in Abhängigkeit der Zugriffsart und der bereits bestehenden Datenverteilung auf dem Speichermedium. Hier wird sieben Mal das Datenvolumen der Nennkapazität auf das Speichermedium geschrieben und jeweils das Zugriffsverfahren geändert. Die Darstellung zeigt ein Standard-Speichermedium mit „page-based Mapping“ ohne DRAM-Cache und ohne pSLC-Cache. Im Nachfolgenden werden die sieben Abschnitte erläutert.

Abschnitt ①: 4 KiB Random-Write

Zu Beginn von Abschnitt 1 ist das Speichermedium leer. Es wurde entweder noch nie beschrieben, ist vollständig getrimmt oder komplett gelöscht („Secure-Erase“). Entsprechend sind die physikalischen Adressen keinen logischen Adressen zugeordnet – die Zuordnungstabellen enthalten keine Einträge. Nun wird das Speichermedium mit zufälligen Schreibzugriffen gefüllt (4 KiB Random-Write). Dabei wird jede logische Adresse genau einmal beschrieben. Durch die kleine Datenmenge pro Schreibzugriff und den hohen Verwaltungsaufwand beim Nachführen der Zuordnungstabellen ist

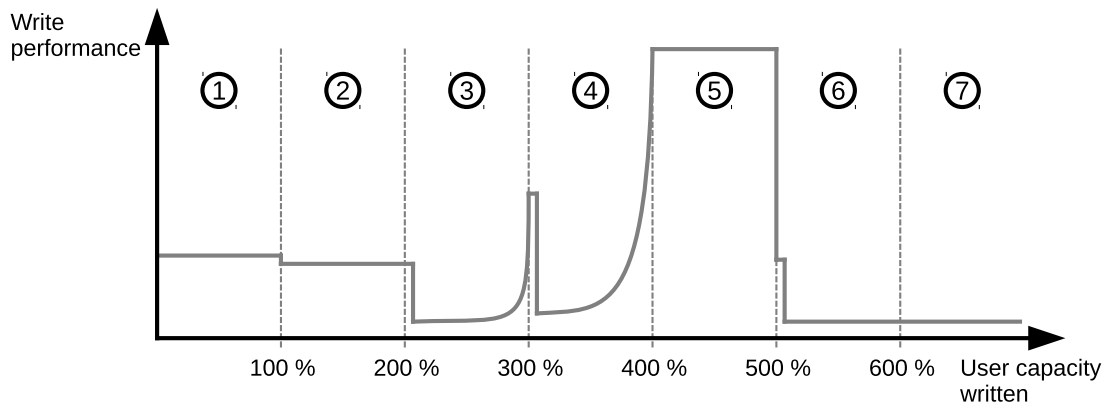


Abbildung 1: Schreibgeschwindigkeit

die Geschwindigkeit stark limitiert. Sie bleibt konstant über die gesamte Zeit.

Abschnitt ②):
4 KiB Random-Write,
gleiche Adressfolge

In Abschnitt 2 werden die Schreibzugriffe aus Abschnitt 1 exakt wiederholt: Die Abfolge der beschriebenen Adressen ist identisch.

Wie zu sehen ist, tritt eine leichte Reduktion der Schreibgeschwindigkeit auf. Dies resultiert daraus, dass die Zuordnungstabellen komplett gefüllt sind und nun für jede neu geschriebene Adresse der alte Eintrag anschließend invalidiert werden muss. Wegen der identischen Adressfolge entsteht hier aber kein Druck auf den Garbage-Collector, da mit jedem neu gefüllten Flash-Block automatisch auch wieder ein anderer Block frei wird. Ist ein Block komplett gefüllt worden, existiert ein zweiter Block mit den gleichen logischen Adressen, die nun aber alle veraltet sind, wodurch der zweite Block gelöscht werden kann.

Häufig wird die Geschwindigkeit von diesem Abschnitt als „Sustained Random Write“ im Datenblatt angegeben. Dies kann den Anwender aber fälschlicherweise annehmen lassen, dass es sich hierbei um die minimal erreichbare Schreibgeschwindigkeit handelt. Die minimale Schreibgeschwindigkeit wird aber erst bei maximaler Last auf dem Garbage-Collector er-

reicht. Dies wird in den folgenden Abschnitten behandelt.

Welche Minimal-Geschwindigkeit für den jeweiligen Anwendungszweck relevant ist, muss daher sorgfältig betrachtet werden.

Abschnitt ③):
4 KiB Random-Write,
neue Adressfolge

Nach dem zweimaligen Füllen des Speichermediums mit gleicher Abfolge von zufälligen Schreibzugriffen wird zu Beginn von Abschnitt 3 die Adressabfolge geändert. Bereits nach dem Schreiben einer kleinen Datenmenge bricht die Geschwindigkeit stark ein: Das „Overprovisioning“ wurde aufgebraucht und der Garbage-Collector, der bisher nicht tätig werden musste, arbeitet nun unter Volllast. Durch die Änderung der Adressabfolge werden nun keine Blöcke mehr automatisch frei, und der Garbage-Collector muss ständig Daten umkopieren, um freie Blöcke für die neuen Daten zu gewinnen.

Da jede Adresse weiterhin in jedem Abschnitt genau einmal beschrieben wird, steigt kurz vor Ende des Abschnitts die Schreibgeschwindigkeit exponentiell an, da – mit der schrumpfenden, noch fehlenden Adressmenge – mit jedem neu gefüllten Block jetzt wieder automatisch Blöcke frei werden. Mit dem letzten geschriebenen Block dieses Abschnitts werden

alle Blöcke des Overprovisionings automatisch wieder frei.

Abschnitt ④: 128 KiB Sequential-Write

Zu Beginn von Abschnitt 4 wird nun von einer zufälligen Adressabfolge auf eine sequentielle Adressabfolge gewechselt. Zudem wird die Datenmenge pro Zugriff von 4 KiB auf 128 KiB erhöht. Diese Erhöhung führt sofort zu einem höheren Durchsatz durch die größere Effizienz der Übertragung zwischen Host und Speichermedium. Nachdem das Overprovisioning aufgebraucht ist, bricht die Schreibgeschwindigkeit nahezu auf den gleichen Wert wie in Abschnitt 3 ein, da alle alten Daten mit zufälligen Adressen geschrieben wurden, und nun die Last auf dem Garbage-Collector genauso hoch ist, wie im vorangegangenen Abschnitt.

Da der Garbage-Collector beim Verschieben von Daten diese möglichst anhand aufsteigender Adressen sortiert, tritt hier schon früher der Effekt auf, dass bei immer kleiner werdender Adressmenge automatisch wieder Blöcke frei werden. Die Schreibgeschwindigkeit steigt dann schneller und höher als in Abschnitt 3 wegen der Zugriffsgröße von 128 KiB.

Abschnitt ⑤: 128 KiB Sequential-Write, zweiter Lauf

Im letzten Abschnitt wurde das Speichermedium sequentiell gefüllt. Die Adressen der Daten in den Flash-Blöcken sind nun in jedem Block streng monoton steigend. Alle Blöcke des Overprovisionings sind wieder frei, da jede Adresse wieder genau einmal beschrieben wurde. In Abschnitt 5 wird das Speichermedium nun genau wie in Abschnitt 4 wieder sequentiell gefüllt. Da die Adressabfolge die Gleiche ist wie im vorherigen Abschnitt, wird mit jedem geschriebenen Flash-Block automatisch wieder ein anderer frei. Durch die große Datenmenge pro Host-Schreibzugriff ist die Übertragung zwischen Host und Speichermedium sehr effizient. Der Garbage-Collector ist ohne Last,

und durch die sequentiellen Schreibzugriffe erzeugt die Nachführung der Zuordnungstabellen ebenfalls kaum zusätzliche Last. Hier wird nun die maximale Schreibgeschwindigkeit des Speichermediums erreicht. Lediglich ein leeres Speichermedium (nach Trim oder Secure-Erase) könnte noch eine geringfügig höhere Geschwindigkeit zeigen.

Abschnitte ⑥ und ⑦: 4 KiB Random-Write, volle Adresszufälligkeit

Zu Beginn von Abschnitt 6 wird wieder auf zufällige Adressfolgen gewechselt. Im Unterschied zu den Abschnitten 1–3, in denen jede Adresse genau einmal beschrieben wurde, steht nun für jede neue Adresse der gesamte Adressraum offen. Entsprechend werden in beiden Abschnitten zum Einen nicht alle logischen Adressen zwangsläufig neu beschrieben, und zum Anderen einige Adressen mehrfach beschrieben. Nach dem Füllen des zu Beginn noch freien Overprovisionings ist der Garbage-Collector unter Volllast. Da ein Flash-Block niemals automatisch frei wird, bleibt der Garbage-Collector dauerhaft unter Volllast. Dieser Betriebszustand ist nun das echte Sustained-Random-Write, bei dem die minimale Schreibgeschwindigkeit erreicht wird. Allerdings dürfte dies keinen praktischen Anwendungsfall widerspiegeln, zumal hier der „Write Amplification Factor“ so hoch ausfällt, dass das Speichermedium sehr schnell seine spezifizierte Anzahl an Schreib-/Löschzyklen erreichen würde.

4 Pseudo-SLC-Cache

Zur Erhöhung der Schreibgeschwindigkeit wurde auch für Industriespeicherlösungen der im Consumer-Bereich weit verbreitete „Pseudo-SLC-Cache“ eingeführt. Hierbei wird ein Teil der NAND-Kapazität als SLC-Speicher konfiguriert, in dem pro Zelle nur ein Bit gespeichert wird. Entsprechend schnell kann dieser Speicher geschrieben und gelesen werden. Da es

sich nicht um einen dedizierten, echten SLC-Speicher handelt, spricht man von Pseudo-SLC (pSLC). Ein solcher Cache kann bei allen Speicherarten, die mehrere Bits pro Flash-Zelle speichern (MLC, TLC, QLC) zum Einsatz kommen. Im Folgenden wird TLC (drei Bit pro Zelle) für die Beispiele angenommen.

Abbildung 2 zeigt den typischen Verlauf der Schreibgeschwindigkeit eines Speichermediums mit Pseudo-SLC-Cache.

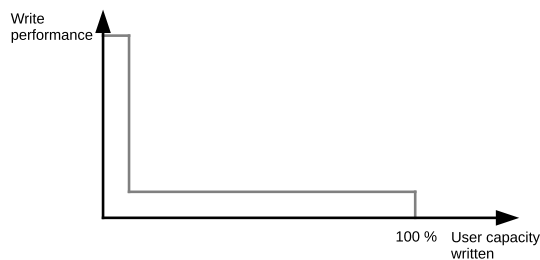


Abbildung 2: Sequential-Write mit pSLC-Cache

Ist der schnelle pSLC-Cache gefüllt, bricht die Geschwindigkeit stark ein, da für weitere Schreibzugriffe auf das Speichermedium erst wieder freier Cache geschaffen werden muss, indem ältere Daten aus dem Cache in den TLC-Speicher verschoben werden.

Die Verwendung von pSLC-Cache bringt besonders dann einen Geschwindigkeitsvorteil, wenn zwischen dem Schreiben größerer Datenmengen das Speichermedium nicht durch Lese- oder Schreibzugriffe ausgelastet ist. Diese Pausen nutzt das Speichermedium, um im Hintergrund die Daten aus dem Cache in den TLC-Bereich zu verschieben.

Ein pSLC-Cache bietet noch weitere Vorteile:

- Er kann das Power-Fail-Verhalten verbessern, da bei einem Stromausfall während eines Schreibvorgangs in den pSLC-Cache keine älteren Daten beschädigt werden können, da nur ein Bit pro Zelle gespeichert wird. Tritt der Power-Fail beim Verschieben der Daten in den TLC-Speicher auf, so sind immer noch die Daten im Cache erhalten.
- Er verbessert die Datensicherheit beim Auftreten von „Bad-Blocks“ besonders bei

3D-NAND bzw. reduziert die Komplexität und dem RAM-Bedarf des Controllers.

Bei der Qualifikation eines NAND-Flash-Speichermediums ist daher ein möglicher pSLC-Cache zu berücksichtigen, da es beim pausenlosen Schreiben großer Datenmengen zu einem massiven Einbruch der Schreibgeschwindigkeit kommen kann. Ein geeigneter Test ist das sequentielle Füllen des Speichermediums zu 100 % ohne Unterbrechung wie in Abbildung 2 gezeigt.

Dynamic-pSLC-Cache

Auch der dynamische pSLC-Cache hat Einzug in Industriespeicherlösungen gefunden. Im Gegensatz zum statischen pSLC-Cache in Abbildung 2 wird hierbei je nach Füllgrad des Speichermediums bis zu 100 % des NAND-Flashes dynamisch als pSLC-Cache verwendet. Hierbei muss folgendes beachtet werden:

- Die Schreibgeschwindigkeit des Speichermediums hängt nicht nur von der Menge der ohne Unterbrechung geschriebenen Daten ab sondern ebenfalls vom Füllgrad des Speichers. Damit ist die Schreibgeschwindigkeit schwer vorhersagbar.
- Vom dynamischen Wechsel der Konfiguration der Flash-Blöcke als pSLC- oder TLC-Speicher wird aus Gründen der Zuverlässigkeit – besonders im industriellen Temperaturbereich – von den NAND-Flash-Herstellern abgeraten. Maximal ist ein einziger Wechsel von pSLC zu MLC zulässig; dieser muss aber innerhalb von 1 % der spezifizierten pSLC-Zyklen erfolgen.
- Alle Hersteller von NAND-Speichermedien mit dynamischem Cache wechseln nach wenigen Programmier-/Löschzyklen auf statischen Cache. Davor erzielt das Speichermedium Bestwerte bei kurzen Geschwindigkeitstests, die sich nicht über die gesamte Kapazität erstrecken. Nach kurzer Nutzungszeit wird das Medium aber dauerhaft langsamer.

Abbildung 3 zeigt den Verlauf eines TLC-Speichermediums mit dynamischem Cache am Anfang der Lebenszeit (durchgezogene Linie) und nach spätestens 10 % der spezifizierten Zyklen (gestrichelte Linie).

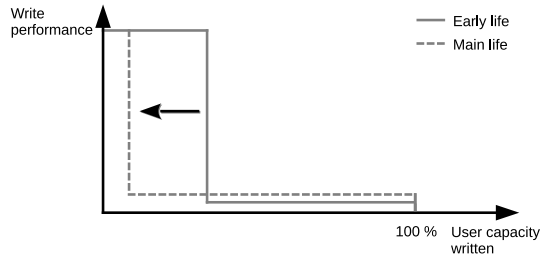


Abbildung 3: Plötzliche Abnahme der Schreibgeschwindigkeit mit dynamischem pSLC-Cache

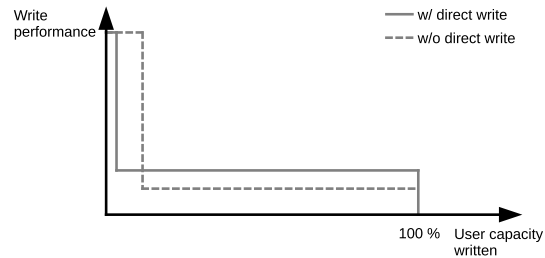


Abbildung 4: Wechsel auf Direct-Write bei Erkennung von sequentiell Schreiben

5 Direct-Write

Als „Direct-Write“ wird die Umgehung eines pSLC-Caches bezeichnet. Dies kann bei verschiedenen Fällen zum Einsatz kommen:

- Wird ununterbrochen auf das Speichermedium geschrieben, und geschieht dies zum größten Teil sequentiell, so kann angenommen werden, dass der pSLC-Cache ohnehin in kurzer Zeit vollläuft. Manche Speichermedien für den industriellen Einsatz umgehen dann den pSLC-Cache und schreiben direkt auf den TLC-Speicher. Dies reduziert die Abnutzung des pSLC-Caches und erhöht bei sehr großen Datenmengen die Schreibgeschwindigkeit, da die Daten nicht doppelt geschrieben werden müssen. Ein Nachteil ist jedoch die reduzierte Sicherheit gegen Stromausfälle. Da große Datenmengen typischerweise nur bei dem Einspielen des Betriebssystems während der Produktion des Endgerätes auftreten, ist dieser Nachteil meist hinnehmbar. Demgegenüber steht der Vorteil des schnelleren Einspielens. Abbildung 4 zeigt den Unterschied beim sequentiellen Schreiben mit und ohne Direct-Write.

- In Kombination mit einem dynamischen pSLC-Cache wird Direct-Write mit dem Ziel verwendet, die Auflösung und Reduzierung eines vergrößerten pSLC-Caches (während eines Geschwindigkeitstests) möglichst lange hinauszuzögern. Abbildung 5 zeigt einen solchen Fall: Das Speichermedium wird sequentiell gefüllt und besitzt einen dynamischen pSLC-Cache, dessen dynamischer Anteil zu Beginn des Tests etwa 3/4 des physikalischen NAND-Speichers belegt, was bei TLC-NAND 1/4 der Nennkapazität entspricht. Ist am Ende von Abschnitt 1 der pSLC-Cache vollständig gefüllt, wechselt das Speichermedium in Abschnitt 2 auf das langsame Direct-Write. Nach Abzug des pSLC-Caches verbleibt noch 1/4 der Nennkapazität als TLC-Speicher für Direct-Write. Ist auch dieser gefüllt, dann ist der physikalische Speicher erschöpft:

- 3/4 der physikalischen TLC-Kapazität wird als pSLC-Speicher verwendet, kann aber somit nur 1/4 der Nennkapazität speichern.
- 1/4 der physikalischen Kapazität wird noch als TLC-Speicher verwendet und kann somit ebenfalls 1/4 der Nennkapazität speichern.

Nun muss das Speichermedium in Abschnitt 3 den dynamischen Anteil des pSLC-Caches in TLC-Speicher umwandeln, um wieder freien physikalischen Speicher zu erhalten. Entsprechend gering fällt die Schreibgeschwindigkeit aus.

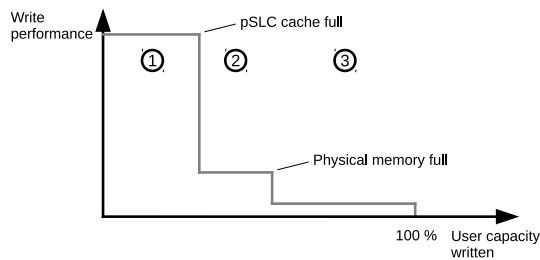


Abbildung 5: Direct-Write bei dynamischem pSLC-Cache

6 Lesegeschwindigkeit

Die Lesegeschwindigkeit ist nicht von so vielen Faktoren abhängig wie die Schreibgeschwindigkeit. Hier kommt es nur darauf an, mit welcher Zugriffsart geschrieben wurde und wie gelesen wird. Abbildung 6 illustriert dies:

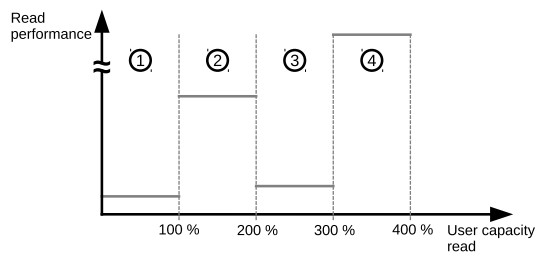


Abbildung 6: Lesegeschwindigkeit

Die Daten in Abschnitt 1 wurden zufällig geschrieben und wieder zufällig – aber in einer anderen Reihenfolge – gelesen. Hier wird die geringste Leseleistung erreicht, da der Suchaufwand in den Zuordnungstabellen sehr hoch ist und die Effizienz der Datenübertragung zwischen Speichermedium und Host wegen der Paketgröße von 4 KiB gering ist.

In Abschnitt 2 werden zufällig geschriebenen Daten sequentiell gelesen. Die Lesegeschwindigkeit ist viel höher, da pro Zugriff nun 128 KiB zum Host übertragen werden, was die Effizienz erhöht. Der Suchaufwand in den Zuordnungstabellen ist aber weiterhin sehr hoch.

Vor Abschnitt 3 wurde das Speichermedium nun sequentiell beschrieben. Die Lesezugriffe in diesem Abschnitt wurden zufällig aus-

geführt. Entsprechend niedrig fällt die Lesegeschwindigkeit wegen der Paketgröße von 4 KiB aus. Durch das sequentielle Schreiben ist die Suche in den Zuordnungstabellen aber einfacher und die Geschwindigkeit höher als in Abschnitt 1.

In Abschnitt 4 erreicht das Speichermedium die maximale Leseleistung. Die sequentiell geschriebenen Daten werden nun auch sequentiell in 128 KiB-Paketen gelesen. Die Suche in den Zuordnungstabellen ist sehr einfach, die Übertragung zum Host sehr effizient, und beim internen Zugriff auf den NAND-Flash kann „Read-Ahead“ zum Einsatz kommen.

Bei einem Speichermedium mit pSLC-Cache würden in den vier Abschnitten für einen Teil der Speicheradressen eine höhere Lesegeschwindigkeit auftreten – nämlich dann, wenn diese Adressen im pSLC-Cache liegen.

7 Zusammenfassung

Zum Vergleich von Geschwindigkeitsangaben in Datenblättern von NAND-Flash-Speichermedien müssen die Testbedingungen bekannt und identisch sein. Wurde die Messung nicht wiederholt gestartet, oder erstreckte sie sich nicht auf den gesamten logischen Adressraum, so kann die angegebene Geschwindigkeit deutlich über der real erreichbaren Geschwindigkeit liegen. Bei der Qualifikation eines NAND-Flash-Speichermediums für geschwindigkeits- oder zeitkritische Applikationen ist ratsam, die maximalen Lastspitzen und Zugriffsmuster der Applikation in einem Testprogramm möglichst gut nachzubilden, und dies über längere Zeit zu testen.

Kontaktieren Sie uns

Hauptsitz	Swissbit AG Industriestraße 4 9552 Bronschhofen Schweiz	Tel. +41 71 913 03 03 sales@swissbit.com
Deutschland (Berlin)	Swissbit Germany AG Bitterfelder Straße 22 12681 Berlin Deutschland	Tel. +49 30 936 954 0 sales@swissbit.com
Deutschland (München)	Swissbit Germany AG Leuchtenbergring 3 81677 München Deutschland	Tel. +49 30 936 954 400 sales@swissbit.com
Nord- und Südamerika	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.

© 2024 SWISSBIT AG