

SSD Doujinshi 3

SSDの同人誌 3

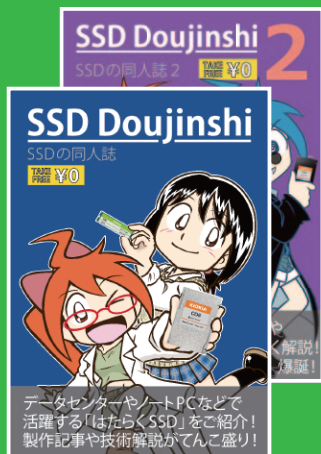
TAKE
FREE ¥0



目からウロコなSSDの面白記事が満載！
データセンターから宇宙まで、どんどん
つながるSSD！自作SSDもアクセル全開！

Contents

AI ぱぱ	SSD 同人誌 3 号の発刊によせて…3
ragnag	キオクシアの SSD が宇宙に旅立ったお話…4
とだ勝之	データセンターてんこ…6
ライゼ	OCP ってなに? データセンターを支える SSD に求められるものとは?…15
伊藤 普朗	ストレージは遠くにありて…18
じむ	SSD のガベージコレクション…20
松澤 太郎	大容量 SSD と OpenStreetMap その 3…24
こたまご a.k.a. ひなたん	CFexpress でもキオクシアが使いたい…28
YmY・Tomo	キオクシア コンシューマー向け SSD と SSD Utility のご紹介…32
Pochio	祝! Maker Faire Tokyo 出展 10 周年…36
余熱	こんなところにも SSD 同人誌が!…41
宮内	JISC-SSD で NAND 型フラッシュメモリを自由自在に動かそう!…43
GPS_NMEA_JP	JISC-SSD を使ってみる…48
@isariyokurari	電子ペーパーディスプレイ搭載 SSD…52
高田	SLC NAND を使用したセンサノード初号機の開発…54
にちか	NAND 型フラッシュメモリで創るコンピュータ…58
memukuge	JISC-SSD のケースをつくる…62
J	バッファ発 SSD 着 FS 鉄道のお仕事…64



SSD ネタ満載の「SSD 同人誌」の PDF 版は以下のリンクから無料でダウンロードできます。ぜひご覧ください！



<https://www.kioxia.com/ja-jp/business/ssd/solution/doujinshi.html>

SSD 同人誌 3 号の発刊によせて

AI ばば

みなさん、こんにちは。

今年も、楽しみにしていた Maker Faire Tokyo に出展させていただくことになりました。

私たちが生み出した誇るべき技術「NAND 型フラッシュメモリ」は、大容量データを不揮発で保持できるストレージを半導体チップ化することを実現しました。USB スティックやメモリカードで写真や音楽をどこにでも持ち歩くことを実現し、携帯電話やスマホ、そしてパソコンでは、手軽に自由にデータをやり取りする仕組みを作り上げました。そして今はデータセンターや企業向け IT システムの採用が加速的に進んでいて、超大容量ストレージの高速化、省電力化を実現しています。

直接的でも、間接的でも、みなさんが日々の生活の中で必ず触れられている、一番身近な半導体デバイスのひとつです。そしてこの技術が無ければ、いまの社会の姿も違ったものになっていたに違いありません。

そしていまウェブになっている生成 AI でも、巨大なデータを学習したり推論したりするなかで、超高速で動く NAND 型フラッシュメモリで構成された SSD が、その成長をささえているのです。GPU のことばかりが世の中では取りざたされていますが、実はこのフラッシュメモリが世の中を変える大きな原動力の一つになっています。

このことを誇りに思いながら私たちキオクシアは、技術開発への挑戦を日々行っています。今年のテーマは「はたらく SSD、つながる SSD」。みなさんの家庭から、宇宙ステーションまで、様々なシーンで使われるデータをつなげていきたいと考えています。

今回の展示で自作 SSD や世界中で使われているデバイスに、実際に触れていただく、私たちが夢中になっている理由がわかってもらえるのでは、と思うのです。単純に『おもしろい!』。だって、ちっちゃなチップに大きな画像、動画、データ、そしてそこに込められた皆さんの思い、夢をいつまでも残すことができるのだから。

キオクシアのミッションは『「記憶」で世界をおもしろくする』。そんな私たちのワクワク感を、今日ここで、この冊子を読みながら感じてもらえればうれしい限りです。

AI ばば

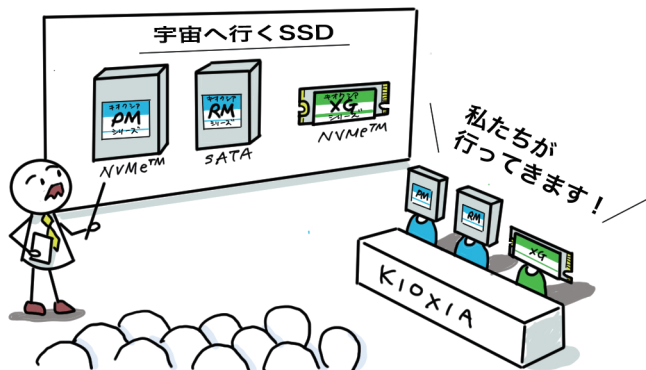


キオクシアの SSD 事業を牽引するリーダー。趣味の無線では、昔ながらの電波を飛ばすアナログ技術と、デジタル技術による交信の融合が起きていて、新たにチャレンジすることがたくさんあります。ちっちゃな部品やケーブルを探しにいく秋葉原の街の変わりようにとまどう日々。

キオクシアのSSDが宇宙に旅立った日

ragnag

キオクシアのSSDが2024年1月に
国際宇宙ステーション(ISS)に旅立つことが決定。



宇宙探掘



ヘルスケア



地球観測



5G



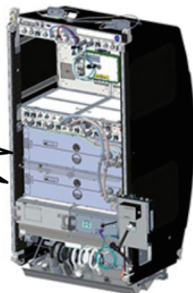
宇宙探索

などなど

いま、宇宙では民間主導で
さまざまな実験が行われ、
たくさんのデータが収集されている。

SSDはISSにあるエッジコンピュータとAIシステムで構成されたシステム『HPE Spaceborne Computer-2 (HPE SBC-2)システム』に設置され、宇宙での実験データや、解析したデータを保管する。

SSDは、ここには
はってるよ!

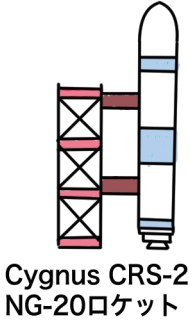


HPE SBC-2システム

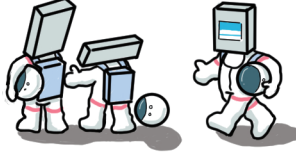
(C)Hewlett Packard Enterprise

キオクシアのSSDは
高いパフォーマンスと信頼性を評価されて
選ばれたんだ。

2024年1月、フロリダ州の宇宙基地から
130テラバイトを超えるSSDが
ロケットに乗って、ISSに向けて出発！



さあ、出発だ！



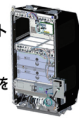
翌日、ロケットはISSに
ドッキングして、
SSDは無事に運び込まれた。

以前は、収集したデータを
地球に10時間以上かけて
送信して解析していた。

今回のプロジェクトでは、
HPE SBC-2システムで
高速に解析し、結果を
わずかな時間で
送信できるようになるんだ。



例えば
1.8ギガバイト
のデータは
約6分で計算、
圧縮したデータを
約2秒で送信



約408 km



もっと詳しい情報は
こちらから



ragnag (@ragnag1109)

まんがを読むのと描くのをこよなく愛する広報担当。
指先の関節がすり減り、指先を使うと痛みを感じる。
ヘパーデン結節と言らしい。
ゲームでスコアが出せなくなったら、、、恐怖！

イラストはイメージです。本文とは関係ありません。

キオクシアは、HPE SBC-2プロジェクトのストレージスポンサーです。

HPE : Hewlett Packard Enterprise

NVMeはNVM Express, Inc.の米国またはその他の国における登録商標または商標です。

その他記載されている社名・製品名・サービス名などは、それぞれ各社が商標として使用している場合があります。

TENCO

パパから
データセンター
見学の

許可が
出たのだー



SSDの申し子
ししご
穴戸ちゃん

というわけで
行くのだ
おと
てんこ姉



あづい〜

がっ



SSDのすごさを
教えてもらった
てんこちゃん

なんてカッコ
してるのだ！

そこまで
寒くないのだ！
脱ぐのだ！

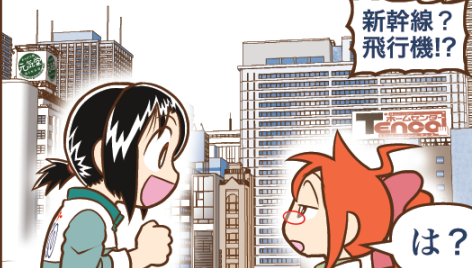
だってたっくさんの
サーバーを
冷やすために
寒くしてあるって



それで
それで!?

ここから
どうやって行くの
データセンター

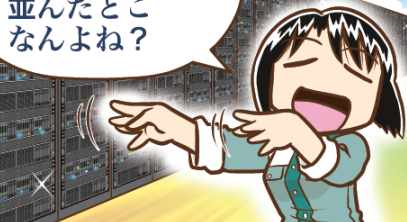
新幹線？
飛行機!?



は？

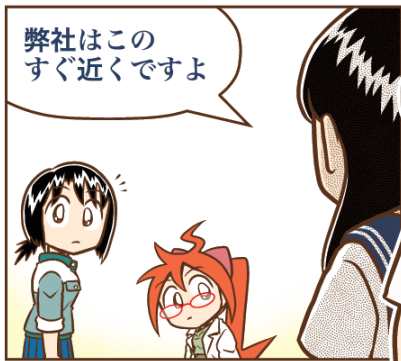
だって
サーバーラックが
ずら——つと
並んだとこ
なんよね？

都内にそんな
広い土地
ないやん！



いや そういう
データセンターも
あるかもだけど…



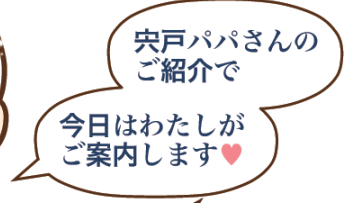
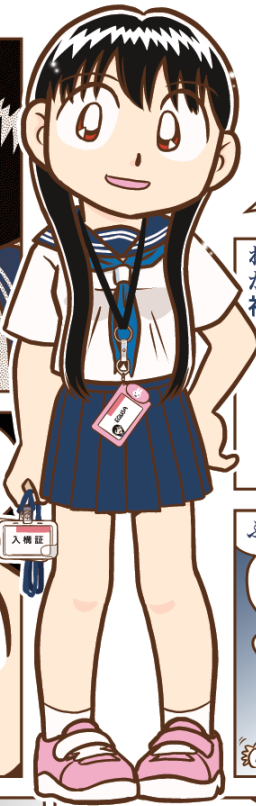


弊社はこのすぐ近くですよ



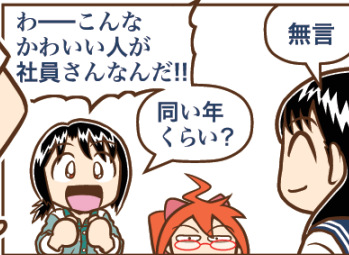
こんにちは

さくらインターネットの江草です



戸川パパさんのご紹介で

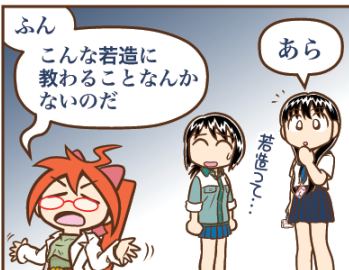
今日はわたしがご案内します♥



わーこんなかわいい人が社員さんなんだ!!

無言

同い年くらい?



ふん

こんな若造に教わることなんかないのだ

あら

おぼろ...



入構証も社員証も指紋認証もなくてどうやって見学するの?

う



データセンターは一見普通のビルの中にも多数あります

都心のビルで窓がないところはデータセンターの可能性が高いです

データセンターに窓は不要ですから

都市には利用客が多いので保守が駆けつけやすいというメリットもあります

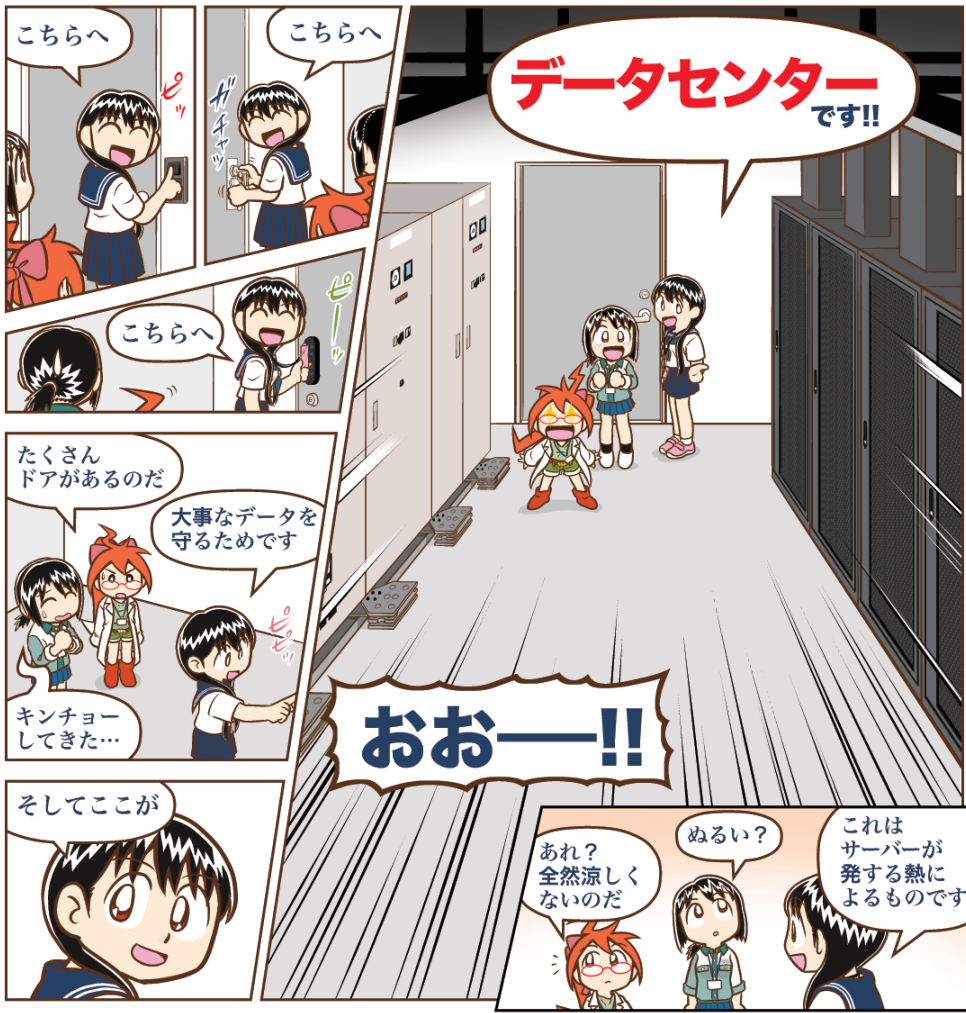
誓約書を書いたら身分証を提示してこちらへ

それくらい知ってるのだ



ぐぬぬ...仕方ないのだ...

ではこちらのビルになります



こちらへ

こちらへ

データセンター
です!!

こちらへ

たくさん
ドアがあるのだ

大事なデータを守るためです

キンチョー
してきた...

おおー!!

そしてここが

あれ?
全然涼しく
ないのだ

ぬるい?

これは
サーバーが
発する熱によるものです

データセンター
にとって
大事なものは
3つ!!

空調!!

回線!!

電力!!

空調はサーバーの入ってる

あの部屋に全フリしてあります

ここがコールドアイルです!!

わ!! ホントだ涼しい

でも思ったほど寒くなかったな

ち…違うのだ!! これは腹巻きじゃなくて…その…

サーバーラックのコスプレなのだ!!

おなかのそれなくても平気だったね

サーバーラックにはいくつものサーバーが入ってるけど

空いてるところがあるとそこに冷たい空気が集中してしまうのだ

そ…そのコスプレをしてきただけなのだ



役目はすんだからもう外すのだ

それを防ぐためにフタをしてるのだ!



SSDラック ちゃん 登場!

大量のSSDの
においが
するのだ!!

出ちゃった
かま...

おお! ここは
データセンター
ではないか!!

サーバーの
発する熱を抑え
最高の環境で
最大限の力を
発揮させる

SSDの理想郷!!

AIがガンガン
使われる今は
速度重視の時代

SSDこそが本流!

もちろんHDDを
載せることも
可能ではあるが

おそらくここは
ほとんど全て
SSDだろう

知っておるぞ
このラック1台で

10PBくらいまで
扱えるのであろう

ということは
ラックを数を
考えと

ここで扱う
データの
総量は...

得た人
たてて
すべす
をば退
許可を
受付外
強制し
いたた

でも あの子が
言ったことは
大体あってます

これだけのデータが行き来するには

回線の安定が大事です

SSDは読み書き速度の安定性にもすぐれてます

でも ネット経由で使うとき遅くて不安定だと困りますよね？

あー！うちのパソコン時々すごく遅くなる！

アパートやマンションの共同回線とはレベルの違う話なのだ

もし大地震が起きて東京のデータセンターが機能しなくなったら大変ですよね？

そうならないよう弊社では大阪・石狩のデータセンターとがっつき接続されています

それだけでなく他の会社ともつながって

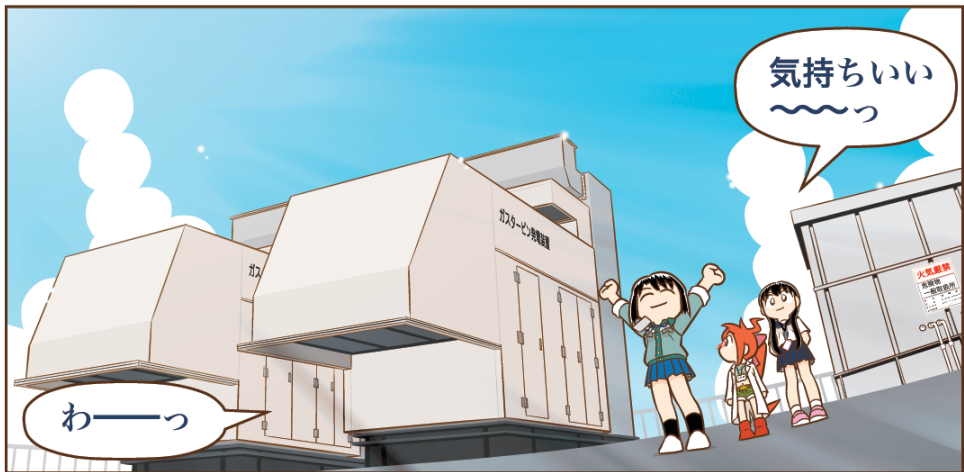
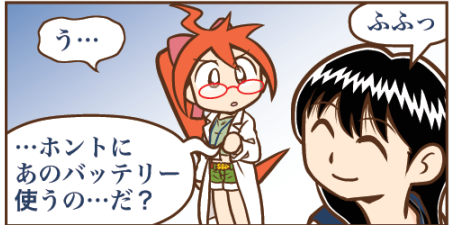
より強固でフレキシブルなネットワークを構築しています

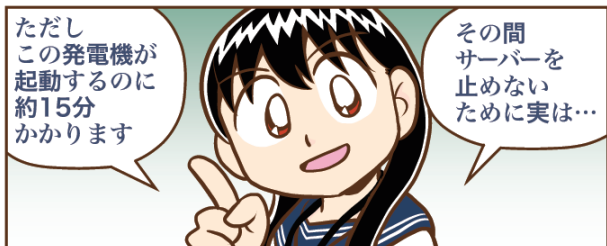
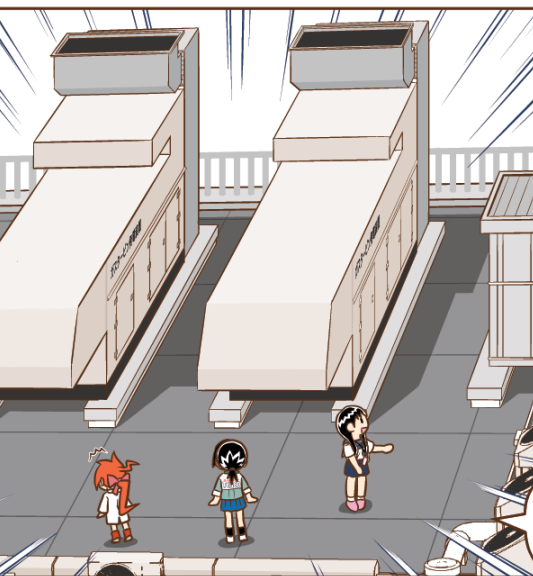
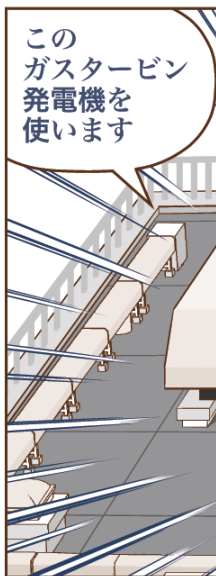
ホントに網の目なんだ！

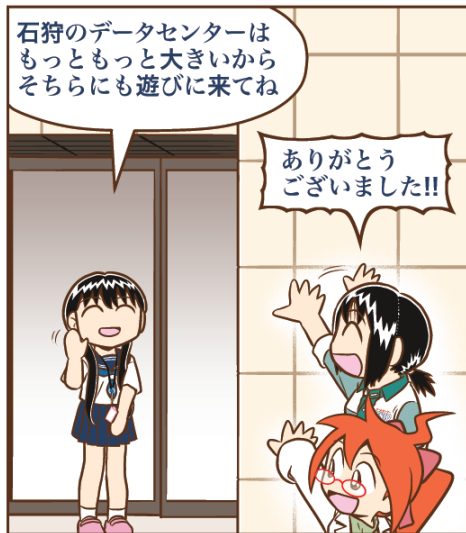
そして大事なもの3つめ!!

電力!!

電気がないとサーバーは止まってしまう







OCP ってなに？ データセンターを支える SSD に求められるものとは？

ライゼ

こんにちは。キオクシアのSSD 応用技術企画担当のライゼです。

突然ですが、みなさんは普段、動画サイトを観たり、ゲームをダウンロードしたり、SNS を使用する際に、視聴している動画や遊んでいるゲームのデータ、アップロードした写真や動画がどこに保存されているのかご存じでしょうか？

えーっと～、データが沢山蓄えられる倉庫のようなところでしょうか。

そう！それは、「データセンター」と呼ばれています！！

データセンターとは？

データセンターとは、その名の通りデータが入っている大きな建物です。みなさんのデータや、企業や学校、病院などが使う大切な情報もしっかりと守ってくれています。そして、そのデータがいつでも必要なときに素早く取り出せるようにしてくれます。

データセンターが無いと、私たちが楽しんだり、生活に役立てているインターネットのサービスがちゃんと動かなくなってしまうます。キオクシアのSSD は、データセンターの中でたくさんのデータを安全に保存しておくために使われているのです。

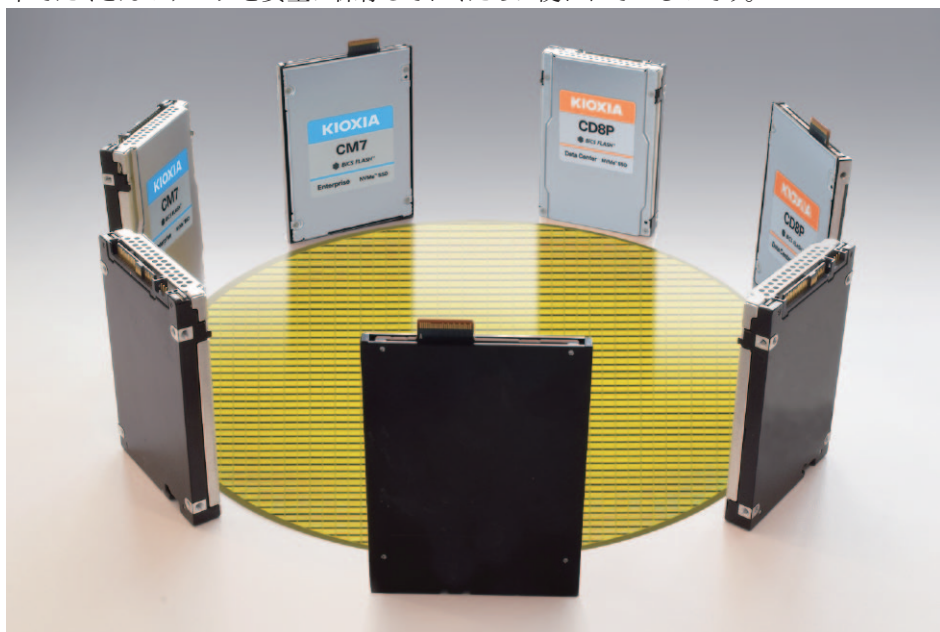


図 1：知恵を出し合う（イメージ）

Open Compute Project

大切な役割を果たすデータセンターですが、それをもっと安全にもっと効率よく設計するために、Microsoft、Meta、Dell、Hewlett Packard Enterprise (HPE) といった、みなさんも良く知る企業が Open Compute Project (OCP) を 2011 年に発足しました。

世界中のあらゆる分野の企業がこのプロジェクトに集まり、データセンターをもっと効率よくするため、みんなで知恵を出し合っていて考えています (図 1)。

OCP ではデータセンターに関わる色々な分野について議論されており、複数の Project があります。SSD は Storage Project に属していて、以下のようなことが求められています。

- **たくさんのデータを素早く取り出せること**

たとえば、ゲームがすぐに読み込まれたり、観たい動画がすぐに始まるようにします。

- **電気を節約すること**

余分な電気を使わないようにします。地球を守るために、エネルギーを節約します。

- **データが安全に保護されていること。**

悪い人が勝手にデータにアクセスできないように、しっかりと守ります。

これ以外にも、みなさんとは直接は関わらないかもしれませんが、保守点検に関わる仕組みについても効率を求められて議論がなされています。キオクシアはこういった要件に見合う SSD を開発しているのです。

OCP Solution Provider Program

Solution Provider Program は、OCP 機器の導入を検討しているユーザーが OCP コンポーネントまたはラックレベルソリューションを設計、購入、導入できるよう支援する Program です。Solution Provider は自社の製品が OCP Accepted™ および / または OCP Inspired™ として認められ、OCP マーケットプレイスウェブサイトに掲載される資格を得ることができます。

KIOXIA XD7P シリーズは、2024 年 3 月より OCP のウェブサイトに掲載されています。ぜひ、皆さんも一度ご覧になってください。

OCP Website 掲載 URL (右下の二次元バーコードからもご覧いただけます):

https://www.opencompute.org/products?cloud_products%5Bquery%5D=kioxia



キオクシアは、他にも OCP Datacenter NVMe™ SSD Specification に対応する製品（図 2：KIOXIA CM7 シリーズ、KIOXIA CD8P シリーズなど）を開発しており、順次 OCP Inspired™ の取得を申請し、製品掲載を拡充していく予定です。



図 2：認証済（KIOXIA XD7P シリーズ）と認証を取得計画中のキオクシアの SSD

OCP Global Summit への出展

毎年 10 月頃に米国カリフォルニア州・サンノゼコンベンションセンターで、OCP Global Summit という展示会が開催されます。ここでは、業界のリーダー、研究者、オープンコミュニティの先駆者が、プロジェクトの進捗状況だけでなく、課題や今後の道のりについての理解を深めることを目的とした対話、啓発的なワークショップ、インタラクティブなエンジニアリングセッションが行われます。2023 年は 4,401 名が参加し、これまでで最高の参加人数となりました（2022 年は 3,500 名参加）。

キオクシアはこの OCP Global Summit に毎年参加しており、今年も出展予定です。2023 年は次世代サーバーとストレージのための新しいフォームファクター EDSFF¹ の講演、及び展示会場ではイベントに先駆けてプレスリリースされた KIOXIA XD7P シリーズと、業界でも先行して PCIe² 5.0 Datacenter Class SSD として市場投入した KIOXIA CD8P シリーズの展示と、E1.L³を採用した KIOXIA LD2-L シリーズの参考展示などを行いました。

OCP Global Summit のウェブサイトにて、展示の内容をご覧になれます。

<https://www.opencompute.org/summit/global-summit>

- 1 EDSFF は、Enterprise and Datacenter Standard Form Factor の略です。
- 2 PCIe とは高速なデータ転送を行うことができる接続規格で、その第 5 世代目です。
- 3 EDSFF の中でも、長い定規のような形状で、1U サイズのサーバーに最適化され、最も高密度なフォームファクターです。



ライゼ

旅行や写真を撮ることが好きです。写真や動画のデータ量が
多く、フラッシュメモリや SSD にはいつも支えられています。

ストレージは遠くにおいて

伊藤 晋朗

我が家には K 君という中学生がいる。とりあえずゲームはやるほうも見るほうも大好きだ。だから、大量の YouTube の中から色々なゲームを見つけて楽しんでみている。まあ、いいのかそうでないのかは、わからないけど知識は大量にたまっているみたい。そんな K 君はしばらく前からゲーミング PC を自作したいらしい。最初は気持ちだけが先行しており何をしたいのか分かっていなかった。なので、何回か秋葉原に連れて行って実際の PC を見せてみた。そうするとなんとなくわかったのか、そのうち、もうネットで選んで買うから良いと断られるようになった。せっかく子供と出かけるというイベントを楽しんでいたのに、そんな時間というのはあっという間に終わってしまうようだ。一緒に秋葉原に行った時に、ただ連れていっただけだと、あまりに味気ないと思うので、ちょうど売っていた弊社製の M.2 SSD の 512 GB を買ってみた。家に帰って K 君にプレゼントすると、すごい頭を下げて喜んでいる。本人用のノート PC があるのだが、ゲームで一杯になりストレージが足りなかったようだ。まあ、本人はすぐ忘れるんだろうが、こっちにとっては思い出になるんだよね。

話は変わるが、かつて、SSD という名前がなかったころ、研究部門の期初方針説明会で、友人 T が「いやー、眠かったけどこれだけはメモした。」というのが、「NAND 型フラッシュメモリに OS 入れて起動すると早くなる。」というもの。当時は名前が違ったがそれが SSD のコンセプトだった。あの頃から 20 年位経って、ノート PC のストレージは HDD から SSD にほぼ置き換わっていった。振り返って考えると世の中変わったものである。まあ、なにせよコンセプトから製品になるってというのは見ていて面白い。

SSD が登場して HDD から何が変わったかというアクセスの反応が 1,000 倍速くなった事。それまで、HDD が 10m sec 以上とかかかっていたアクセスが、最近の SSD では 10 μ sec 前後で読み書きができるようになっている。そのため、ランダムアクセスが必要な PC の起動時間はかなり短縮された。昔は会社に出社したときに、PC の電源を入れてからコーヒーを入れに行こうとしたもんだが、最近ではそんな余裕がない。ある意味、人間には厳しい世の中になってきているのかもしれない。

2010 年代からは、スマホが一般的になりすっかりクラウドベースアプリが主流になってきた。スマホでいろんなことができるがデータそのものはネットの向こう側にあることが多い。そして、データセンターの中でもストレージというのはコンピューティングするサーバーとはだいたい離れているストレージ専用のサーバーや機



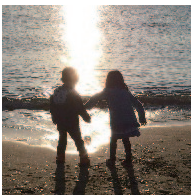
器で保存されている事が多い。そうするとコンピューティングとストレージが離れてしまうのだが、基本的にはファイルシステムレベルでしかストレージは利用しないのでダイレクトにはアクセスすることはないだろう。しかし、中にはネットワーク越しでもダイレクトにつなげようと考えている人たちがいる。そんな規格が NVMe™ over Fabrics (NVMe-oF™) である。この NVMe-oF は、ネットワークを通じて NVMe プロトコルを利用するための技術である。これにより高い性能の SSD をネットワーク越しにそのまま使えるものである。NVMe-oF には、RoCE、Fibre Channel、InfiniBand、TCP などのさまざまな物理層が存在し、それぞれ異なる特性と用途があったりする。

まず、RoCE (RDMA over Converged Ethernet) は、Ethernet ネットワーク上で RDMA (Remote Direct Memory Access) を実現する技術。これにより、高速で低遅延な通信が可能なる。RoCE は主にデータセンターで使用される規格で、データセンター向けの専用のハードウェアと設定が必要。プロトコル的には UDP を使っているので大丈夫かという人はいるけど、前提としてパケットロスがないネットワーク機器を使うので大丈夫。

次に、Fibre Channel は、ストレージエリアネットワーク (SAN) で使用される高速ネットワーク技術で、Ethernet とは全然違う規格でできている。高信頼性と低遅延が特徴だけど、ちょっと本当に NVMe-oF が実際に使われるかよくわからない。InfiniBand も、高速で低遅延なデータ転送を可能にする通信プロトコルで、主に高性能コンピューティング (HPC) で使用されていると言われているがこっちは正直見たことがない。

最後に、TCP (Transmission Control Protocol) は、インターネットなどで一般的に使用される通信プロトコル。TCP は広範囲での互換性を持ち、専用のハードウェアが不要で、既存のネットワークインフラを活用できるのが魅力的。RoCE や InfiniBand と比べて低遅延は提供しないけど、チューニングすれば 100 μ sec ぐらいのレイテンシで通信できたりするんで意外と使えるのかもね。このように、ストレージの世界にもネットワークの技術を使おうとしている人たちがいる。製品としても NVMe-oF をサポートしている機器もあるからどっかに需要があるんだろうし、データセンターの中も変わっていくのかもしれない。

話は変わるが、K 君は中学生になって忙しそうだし教科書の量は増えているし、自分が子供のころと比べるとはるかに教科書の質が変わっている。まあ、まだまだテストは暗記で答えられる問題が大半のようではあるんだが、いつか自分の脳とストレージがつながる世界がきたら変わるのかもねえ。

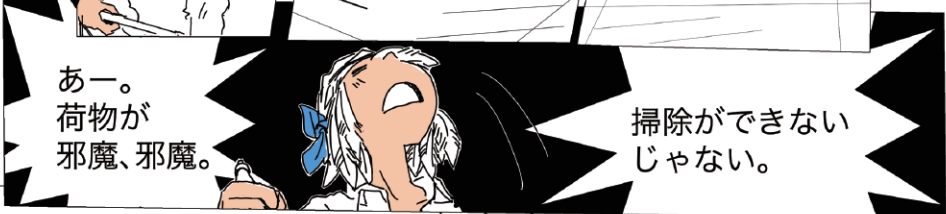
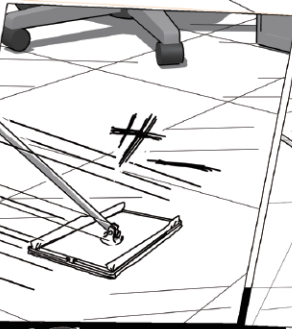


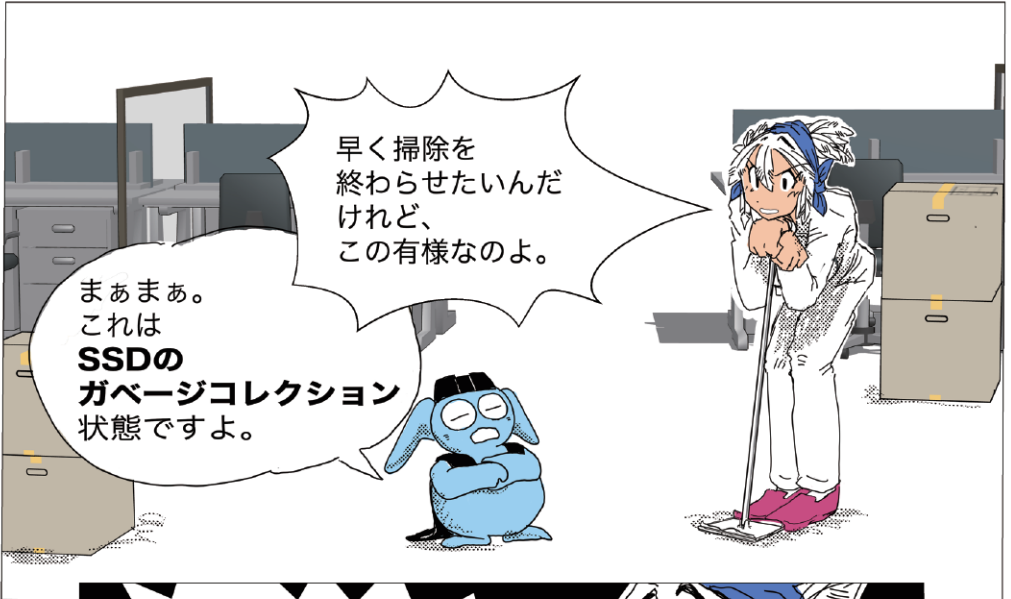
伊藤 晋朗 (@ikainuk)

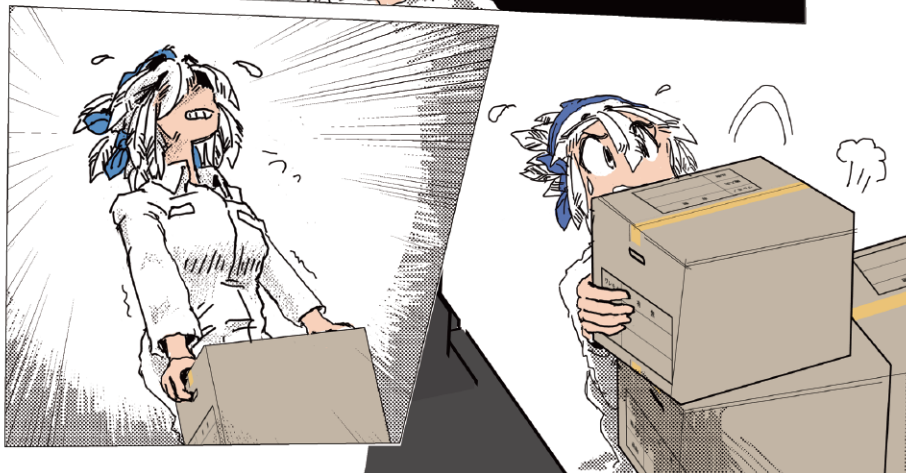
数年間、手を動かすことから離れていたけど、やっと作業ができる立場になった。新鮮な気分だ。

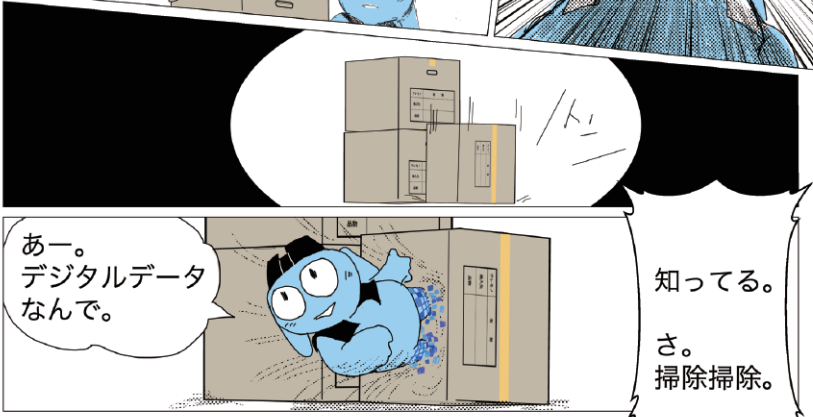
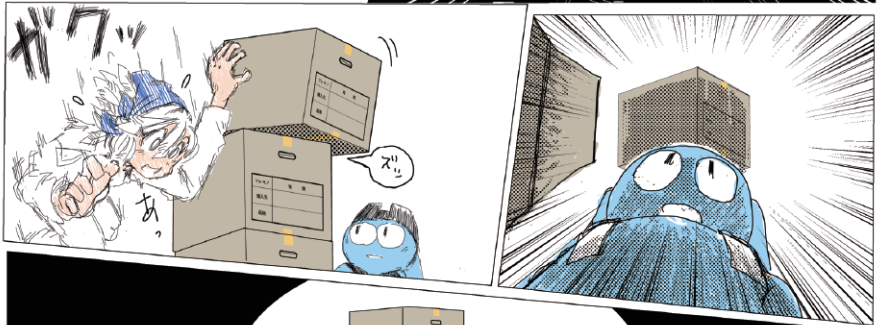
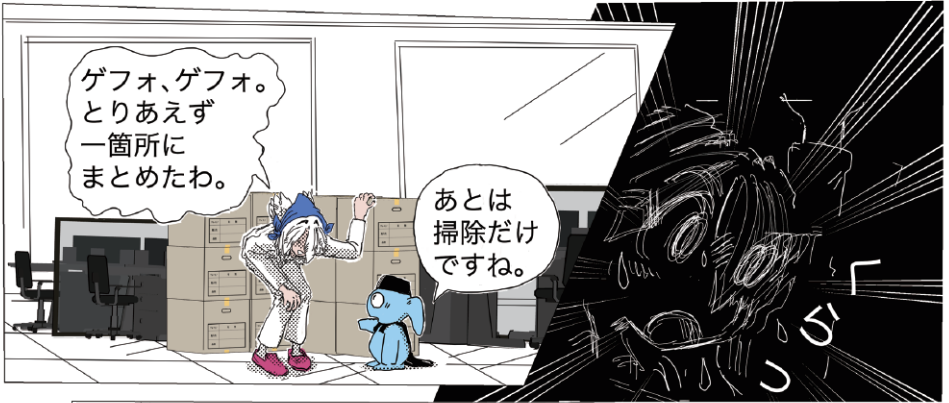


SSDの ガベージコレクション









じむ (@Hirameki_Sora)



キオクシアさんと会社が離れてしまいましたが、こうやって薄い本が出るたびに声をかけてもらって連載(?) できているのはありがたいことです。

個人ニュースとしては、会社を退職することになり以降は無職。さーて、身の振り方を考えなきゃ。

大容量 SSD と OpenStreetMap その 3

松澤 太郎

過去2回にわたり、「大容量 SSD と OpenStreetMap」というタイトルで寄稿をさせていただきました。今回はその第3回となります。

今回用意していただいたマシンは前回と同じく、EDSFF(Enterprise and Datacenter Standard Form Factor) に準拠した PCIe @4.0・NVMe™ 1.3c 対応の SSD、KIOXIA XD6 Series (図 1) をキオクシア様からお借りしています。



図 1 : KIOXIA XD6 Series

OpenStreetMap の基本的な説明については SSD 同人誌 1 号を参照していただきたいのですが、簡単に説明すると「世界単位の単一の地図データベース」となります。データベースの形式は XML ファイルで、世界全体 (<https://planet.openstreetmap.org/>) で 2024 年 7 月現在、bzip2 圧縮で 141GB、Protocol Buffer 形式で 76GB ほどになります。日々データは増え続けており、一昨年(2022 年 7 月) 同人誌と比較すると bzip2 圧縮で 23GB ほどデータが増えています。

検証マシンの紹介

検証マシンは前回と同じマシンを用いています。

- ASRock Z690 Pro RS
- Intel® Core™ i5-12400
- 16 GB Memory
- SSD KXD6CRJJ3T84 (3.84 TB PCIe®)

OS は Debian bookworm にアップグレードをしています。

Swap メモリは 500GB を割り当てています。

Planetiler による世界中のタイル作成

まずは、前回の同人誌の最後でも扱った Planetiler¹ による世界中のタイル作成を行います。前回からの変更点としては Planetiler 0.8.2 では Java 22(LTS ではないバージョン) に対応し、後述するベンチマークで利用する Java profile に対応したところと、全体的にパフォーマンスが向上したという点があります。

¹ <https://github.com/onthegomap/planetiler>

ただしインポートする Planet ファイルの大きさが肥大化したため、前回は 20GB のメモリで処理ができたのに対し、今回は 24GB のメモリを確保する必要があります。

では、Planetiler 0.8.2 を使って計測をします。前回と違う点としては docker を使って処理を実施しているところが異なります。また、メモリの確保量についてはスクリプト内に記述をしています。

```
git clone https://github.com/smellman/planetiler-scripts.git
cd planetiler-scripts
mkdir -p data/sources
cd data/sources
aria2c https://planet.openstreetmap.org/pbf/planet-240715.osm.pbf.torrent
mv planet-240715.osm.pbf planet.osm.pbf
cd ../../
time ./runworld-docker.sh
real    310m4.248s
user    0m1.021s
sys     0m3.125s
```

前回の同人誌のデータでは約 7 時間ぐらいの処理でしたが、今回の処理では約 5 時間 10 分と大幅に処理が改善されています。

では、SSD がどのぐらい利用されているかを見てみましょう。

まずは Read のグラフです (図 2)。Read の性能として 7.17GiB/s が最大値として出てきます。



図 2 : XD6 の読出速度の実測値

次に Write のグラフです (図 3)。Write の性能として 5GiB/s が最大値として出てきます。いずれにしても処理の一部を swap メモリに頼った状態でも高い性能が出ている事がわかります。



図 3 : XD6 の書込速度の実測値

Overture Maps のコンバートにチャレンジ

今回は OpenStreetMap 以外の地図のコンバートにもチャレンジしてみました。

Overture Maps(<https://overturemaps.org/>) は OpenStreetMap と同じく世界規模の地図のオープンデータです。大きく異なるのが OpenStreetMap はコミュニティがベースとなって地図の作成を行っているのに対して、Overture Maps は企業がベースとなって地図の作成やプロダクションレベルでの利用を想定したデータの正規化などを行っている所です。なお、企業がベースとありますが、中には OpenStreetMap のデータが使われていることもあります。

今回は Planetiler-example というレポジトリを用いて PMTiles 形式への変換を試みます。PMTiles 形式は Cloud Native に対応したタイル形式で、HTTP Range リクエストを用いてタイルにアクセス可能な所がポイントとなります。

Planetiler-example では Java 22 から採用された Java profile という機能を使います。そのため、まずは Java JDK 22 を Debian bookworm に導入します。

```
sudo su -
apt install -y wget apt-transport-https gpg
wget -q0 - https://packages.adoptium.net/artifactory/api/gpg/key/public | gpg
--dearmor | tee /etc/apt/trusted.gpg.d/adoptium.gpg > /dev/null
echo "deb https://packages.adoptium.net/artifactory/deb $(awk -F= '/^VERSION_CODENAME/{print$2}' /etc/os-release) main" | tee /etc/apt/sources.list.d/adoptium.list
apt update
apt install temurin-22-jdk
apt install awscli
```

最後の awscli は Overture Maps を AWS S3 からコピーするために利用します。では、Planetiler-example のセットアップとデータの整備を行います。

```
git clone https://github.com/onthegomap/planetiler-examples.git
cd planetiler-examples
wget https://github.com/onthegomap/planetiler/releases/latest/download/planetiler.jar
mkdir -p data/overture
cd data/overture
aws s3 cp --region us-west-2 --no-sign-request --recursive s3://overturemaps-us-west-2/release/2024-06-13-beta.1/ .
cd ../../
```

ここで注意をしていただきたいのは 2024-06-13.beta.1 であることです。beta.0 では文字化けが発生することが分かっています。また、423 GB ほどの GeoParquet 形式のデータをダウンロードする必要があるため、事前にストレージに余裕があることが必須です。なお、GeoParquet は Apache Parquet を地理情報で扱えるようにしたものです。それではコンバートを試してみましょう。なお、メモリのサイズは指定しないものとします。

```
time java -cp planetiler.jar Overture.java
real    181m27.227s
user    1984m4.285s
sys     17m45.534s
```

なんと、400GB を超えるデータが 3 時間弱で処理が完了しました。これは Overture Maps のスキーマが OpenStreetMap に比べて複雑ではなく、かつ GeoParquet という効率の良いファイル形式を使っていることが要因ではないかと筆者は考えています。また、出力されたファイル (data/overture.pmtiles) は 51GB とサイズが小さいのも特徴です。

グラフは割愛しますが、SSD の性能としては Read / Write とともに最大 2.6 GiB/s 程度のスピードで処理をしています。OpenStreetMap のデータ処理に比べると速度が半分程度ですが、Java のパラメータ次第では高速化の余地があるのではないかと考えています。

Overture Maps 自体は執筆時現在で 2024-07-22 が最新版となり、スキーマのバージョンが 1.0 になったばかりのまだ若いデータです。

OpenStreetMap 同様、簡単に処理ができるので皆さんも試してみたいはいかがでしょうか？



松澤 太郎 (@smellman)

一般社団法人オープンストリートマップファウンデーション
ジャパンで技術担当をしています。他にも日本 UNIX ユーザー
会副会長、一般社団法人 OSGeo 日本支部理事など、地図と
UNIX を愛するエンジニアとして活動中。

CFexpress でもキオクシアが使いたい

こたまご a.k.a. ひなたん

前回寄稿させていただいたFlashAir™ Doujinshi 6に引き続き、カメラ関連のお話です。

CFexpress の登場

皆さんは、普段カメラで撮影する際に EXCERIA ブランドの SD メモリカードをお使いですか？

キオクシアさんの EXCERIA SD メモリカードは高性能で信頼性も抜群です。しかし、最近のカメラでは SD メモリカードに代わり、CFexpress カードが採用されることが増えています。CFexpress カード(図1)は、内部的には NVMe™ インターフェースを使用しており、PCI Express® (PCIe®) 接続になっているため、非常に高速なデータ転送が可能です。

私も利用しているミラーレスカメラもそうですが、最近のカメラは連写性能も高く、毎秒30コマの RAW 撮影や、8K の RAW 動画が撮影できたりします。

8K RAW の動画は 60 秒で約 10 GB に達し、170 MB/sec 程度の書き込み速度が求められます。RAW 画像を毎秒 30 コマ撮影すると約 1,400 MB/sec の書き込み性能が必要となります。

通常の CF カードや SD メモリカードではこの速度は達成できず、最新規格である CFexpress も SD Express も中身は PCIe 接続の NVMe インターフェースとなっています。

連写や動画撮影を頻繁に行わない場合でも、転送速度は重要です。撮影後にパソコンにデータを取り込む際の時間に影響します。CFexpress の場合、USB 3.2 対応のリーダーを利用すると 1GB/sec 程度の速度でコピーができます。また、Thunderbolt™ 4 で PCIe Gen4 x2 の速度で接続できると、3 GB/sec 程度の速度(現時点ではインターフェースではなくカードの性能が律速段階)でコピーができます。UHS-II の SD メモリカードの読み出し時間に比べ、3分の1から10分の1の時間でコピーを完了させることができます。

CFexpress カードは、Type A、Type B、Type C の 3 種類のサイズがあり、キヤノン、ニコン、パナソニックや富士フイルムは Type B を、ソニーは Type A を採用しています。本記事では Type B を対象としています。

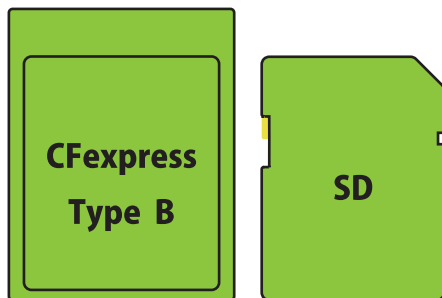


図1：CFexpress と SD カードの外形 (原寸大)

NVMe の重要性

現在の SSD の性能向上には、PCIe と NVMe (Non-Volatile Memory Express) プロトコルが非常に重要な役割を果たしています。これまで利用されてきた SATA や SD、CF などのインターフェースは物理層の帯域の伸びが芳しくありません。一方で、PCIe インターフェースは GPU の接続などで物理層の改善が進んでおり、PCI Express 6.0 では 1 レーンで 7.563 GB/sec の帯域が利用できます。NVMe SSD では、インターフェースに PCIe を利用することで、物理層の帯域を確保しています。

さらに、物理層の改善だけでなく、NVMe はコマンド体系も一新されました。新しいコマンド体系は、従来の SATA インターフェースに比べてシンプルに作り直されており、大幅に効率化されています。SATA インターフェースのコマンドは大昔の ATA (IDE) の時代から続いており、互換性維持のため複雑なコマンドになってしまっています。NVMe は ATA の時代から続く複雑なコマンドを捨て、現代的な設計のコマンド体系に置き換えられました。

特に I/O キューの設計は SSD に対する読み書きのスループット向上に重要な役割を果たしています。従来の SATA インターフェースは、最大でも 32 の I/O キューしか同時に処理できませんでした。NVMe は最大 64,000 の I/O キューをサポートし、各キューではそれぞれ 64,000 のコマンドを処理することができます。この違いは、連続して書き込み続けなければならない場合において効果的で、本来 NAND 型フラッシュメモリが持っている性能を最大限引き出すための重要な違いです。

CFexpress でもキオクシアが使いたい

さて、NVMe がなぜ素晴らしいのかを語りましたが、残念ながら、キオクシアさんからは CFexpress のカードは販売されていません。新しいカメラになり CFexpress が利用可能になった際、これまでの SD メモリカードと同様に、キオクシアの SSD を使いたいと考えていました。

ここで注目したいのが、キオクシア社の M.2 NVMe SSD¹ である KIOXIA BG シリーズです。特にこのシリーズは、サイズ 2230 (長さ 30mm) の NVMe SSD として、CFexpress Type B カードよりもさらに小型である点が特徴です。

この 2230 サイズの NVMe SSD を CFexpress Type B カードとして使用するための変換アダプタが市販されており、私はそれ以来、キオクシア BG4 シリーズを利用しています。

今回の寄稿にあたり、KIOXIA BG5 と BG6 シリーズ (図 2) もお借りしましたので、検証結果も踏まえてご紹介します。



図 2 : KIOXIA BG6 シリーズ (原寸大)

1 M.2 フォームファクターについては SSD Doujinshi 2 を参照

実際 CFexpress カードにして KIOXIA BG シリーズを使ってみる

私が使用している CFexpress 変換アダプタ (図 3) は、内部の基板に M.2 NVMe SSD を差し込んでから、蓋を閉めるだけのシンプルな構造です。SSD は発熱が大きいので、蓋を閉める前に熱伝導グリスを塗布し、蓋を通じて効率的に放熱できるようにするのが望ましいでしょう。CFexpress 変換アダプタには複数の構造が存在しますが、私はこのタイプのアダプタが接触不良が少なく、信頼性が高いと感じています。

キヤノンとニコンのミラーレス一眼でテストした結果 (図 4)、KIOXIA BG4 と BG5 シリーズは正常に動作しましたが²、KIOXIA BG6 シリーズではカードアクセスエラーが発生し、フォーマットなどの操作ができないことが判明しました。



図 3 : KIOXIA BG5 シリーズを入れる



図 4 : ミラーレスカメラに搭載

nvme-cli の紹介

M.2 NVMe SSD は、CFexpress カードとしての使用を目的に設計されていないため、動作しない場合があるのは自然なことです。しかしながら、何がその違いを生んでいるのかは気になるため、調べてみることにしました。

Linux カーネルでは NVMe が標準でサポートされており、nvme コマンド (nvme-cli パッケージ) を使用することで、NVMe SSD の詳細な情報を取得し、各種操作を行うことができます。詳細は省略しますが、**nvme id-ctrl /dev/nvme0** および **nvme id-ns /dev/nvme0n1** の結果を確認することで、NVMe SSD のパラメータを確認できます。

```
sudo apt install nvme-cli
sudo nvme id-ctrl /dev/nvme0
sudo nvme id-ns /dev/nvme0n1
```

KIOXIA BG5 と BG6 シリーズを比較した結果、いくつかのパラメータに違いが見られました³。

² 動作を保証するものではありません、自己責任でご利用ください。

³ なお、エンジニアリングサンプルと市場流通品には違いがある可能性があります。

例えば、気になった点としては、Host Memory Buffer Minimum Size (HMMIN) で、ホスト（この場合はカメラ）側に用意を求めるメモリサイズが KIOXIA BG6 シリーズで増加していました。

SSD の小型化を実現するために、内蔵 DRAM を省略することが求められました。この際、ホスト側の DRAM を利用して性能低下を抑えるために活用されるのが HMB (Host Memory Buffer) です。KIOXIA BG シリーズでも DRAM を省略するために HMB が採用されています。HMB の利用は必須ではありませんが、この最小サイズが増加したことが、カメラ側での対応の違いにつながったかもしれないと妄想することもできます。

何が原因かは想像することしかできませんが、KIOXIA BG6 シリーズのコントローラーが変更されたことが CFexpress としての動作に影響を与えていると思われます。将来的に発売されるカメラで対応されることを期待しています。

まとめ

今回の寄稿を通じて KIOXIA BG5 シリーズでの動作が確認できたため、早速 KIOXIA BG5 シリーズの 1TB を入手し、実際に使用しています。CFexpress カードの導入を検討されている方は、試してみるといいかもしれません。

最後に補足ですが、アダプタを利用しなくてもキオクシアの CFexpress を利用することはできます。キオクシアの NAND 型フラッシュメモリを採用していることを公開しているメモリカードメーカーがありますので、そのカードを購入すればキオクシアのチップを使用できます。CP+ などカードメーカーさんに聞くと教えてくれます。ただし、コントローラーがキオクシアの SSD と同等かどうかは保証されていないことに留意してください。



こたまご a.k.a. ひなたん (@chibiegg, @hinatan_net)

写真と電子工作が好きな男の娘

インフラ写真集やポートレート写真集を出しています

てんこちゃんをデータセンターに案内しました。

「調布技研」 <https://chofutech.booth.pm/>

「ひなたん写真館」 <https://hinatan.booth.pm/>

Instagram & X : @chibiegg, @hinatan_net

キオクシア コンシューマー向け SSD と SSD Utility のご紹介

YmY・Tomo

キオクシアは 2020 年よりキオクシアブランドでのコンシューマー向けの SSD の展開を始め、今年の 10 月で早 5 年となります。今までお読みいただいた方々ならご存じかもしれませんが、本同人誌は、法人向け SSD の話題を中心にしていました。今回ご縁を頂き、コンシューマー向け SSD とこれら SSD の管理・モニタリング等にお使い頂けるツールである SSD Utility の紹介記事を執筆させて頂けることになりました。

コンシューマー向け SSD ラインナップのご紹介

キオクシアのコンシューマー向け SSD 製品は USB メモリや SD メモリカード同様、「EXCERIA」という名前を冠しています。SSD のラインナップには図 1 の様に大きく分けて「NVMe™ SSD」、「SATA SSD」、そして「ポータブル SSD」の 3 つのカテゴリがあります。各カテゴリはハイエンドユーザー向けの「EXCERIA PRO」シリーズ、メインストリームユーザー向けの「EXCERIA PLUS」シリーズ、そしてエントリーユーザー向けの「EXCERIA」シリーズと区分けされています。また、コンソールゲーム機向け SSD として「EXCERIA with Heatsink」もあります。



図 1：キオクシアのコンシューマー向け SSD 製品

ヘビーゲーマーやワークステーション用には「EXCERIA PRO の NVMe™ SSD」、オフィスワークや HDD / SATA SSD からの換装には「EXCERIA の NVMe™ SSD」など、様々な用途に向けた SSD をラインナップしておりますので、是非一度弊社 Web サイトをご覧ください。

そして、次の章ではこれら SSD を管理する際にお使い頂ける Windows 10/11 向けツールである「SSD Utility」をご紹介します。

1 <https://www.kioxia.com/ja-jp/personal/ssd.html>

SSD を分かりやすく管理！ SSD Utility のご紹介

キオクシアのコンシューマー向け SSD 製品をお客様ご自身で簡単に管理できるソフトウェアアプリケーションツール「SSD Utility」を無償でご用意しています。SSD Utility は、SSD の容量状態の確認はもちろん、SSD の温度状況や、寿命をリアルタイムで知る事ができます(図2)。また、より詳しく SSD の状態を確認したいお客様に向けて S.M.A.R.T (Self-Monitoring, Analysis and Reporting Technology) 情報も確認可能です。

このほかに、パスワードロック機能、Secure Erase 機能、Firmware 更新機能、サポートパッケージ生成機能などがあります。



図2：SSD Utility の画面

<コラム>

秋葉原にキオクシア SSD の広告出現！

キオクシアがパソコン工房秋葉原パーツ館に広告を展開しています。秋葉原本店やグッドウィル四日市店でもご覧いただけますので、是非チェックしてみてください！

(2024年9月現在)



パスワードロック機能

キオクシアのポータブル SSD 製品においては、パスワードロック機能をご利用いただけます(図3)。持ち運びの際などに万が一紛失しても、ポータブル SSD をパスワード保護しておくことで、お客様のデータが漏洩してしまうのを防ぐことができます。



図 3 : SSD のパスワードロック画面

Secure Erase 機能

SSD の情報は単純にファイルを削除しただけでは、復元が可能であるという事をご存じでしょうか？ 中古のパソコンや、SSD を買い取った人がデータを復元して情報漏洩が知らないとこで起きている・・・これは身近な問題です。SSD Utility は、お客様がお使いの SSD を手放すとなった場合においても安全に SSD のデータを消去することができます。

Firmware 更新機能

製品の動作安定性や機能改善などを目的に SSD 製品の Firmware が更新されることがあります。SSD Utility はお客様がお使いの SSD の Firmware が最新であるか確認し、アップデートする Firmware があれば簡単に更新する事が可能です。SSD をシステムドライブとしてお使いの場合でも、データドライブとしてお使いの場合でも SSD Utility が Firmware 更新をナビゲートします。

SSD Utility を使ってみよう！ Firmware 更新

SSD Utility を用いた Firmware 更新をご紹介します！ 初めて Firmware 更新を行う場合でも、SSD Utility が簡単ステップでナビゲートしますので誰でも簡単に更新可能です(図4)。

0. まずは不測の事態に備えて、データをバックアップ！
1. SSD Utility のディスク情報画面から“ファームウェア更新”メニューをクリック！
2. ファームウェア更新欄の“更新”をクリック！（更新がない場合は押せません。）
3. データバックアップに関する確認をしたら“OK”をクリック！
4. ポップアップの指示に従って更新完了！

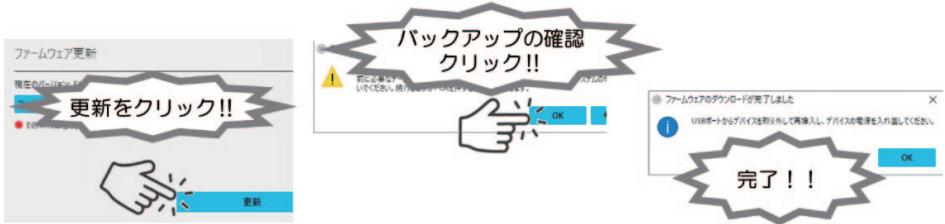


図4：SSD の Firmware 更新手順

データを削除する Secure Erase はもちろんの事、Firmware 更新も SSD の動作に関わる重要な操作になります。機能実行の際にお使いのパソコンで予期せぬトラブルが発生した場合、意図せずデータが消えてしまう、SSD が使用できなくなる、といった可能性がゼロではありませんので、Firmware 更新、Secure Erase などを実行する前に、お客様の大切なデータはしっかりバックアップを取っていただくよう、お願いいたします。

SSD Utility はこちらからダウンロードできますので是非お試しください。

<https://www.kioxia.com/ja-jp/personal/software/ssd-utility.html>



Tomo

ゲームと SUP が趣味の両極端おじさん



YmY

SSD 歴 3 年生。

最近ビオトープに沼り中。

祝！ Maker Faire Tokyo 出展 10 周年

Pochio

今年も SSD 同人誌の新刊を手にとっていただきまして、ありがとうございます。SSD 同人誌はこれまでに 2 冊発行しました。以前の FlashAir™ 同人誌は特別版を合わせて通算 7 冊発行しましたので、既刊は全 9 冊(図 1) となります。したがって、この SSD 同人誌 3 号が記念すべき 10 冊目です。10 冊全てお持ちの方がいらっしゃいましたら、ぜひ「# 同人誌 10 冊コンプリート」とハッシュタグをつけて、写真をツイートしてくださると嬉しいです！

振り返れば、初めて Maker Faire Tokyo (以降 MFT) にスポンサー出展したのが 2014 年でした。おかげさまで初出展から 10 周年を迎えることができました。ありがとうございます。ちなみに、この同人誌を長年お読みいただいている方ならご存知かと思いますが、初回出展当時は会社名が違いました

(汗)。2014 年から 2019 年までの 6 年間は、無線 LAN 搭載 SD メモリカード「FlashAir」を用いた様々な作品を展示し、FlashAir 同人誌を 6 冊発行しました。これらの同人誌は既に配布を終了しておりますが、全 6 冊の PDF 版を 2 次元バーコードからご覧いただくことができます。



FlashAir 同人誌の
サイト

ところで FlashAir 同人誌は 1 号から 6 号まで、通算 21,000 部を配布しました。MFT での同人誌の配布は私共にとりまして恒例行事となりました。残念ながら 2019 年で FlashAir 同人誌の製作は終了しました。翌 2020 年は FlashAir に代わる展示テーマが決まらず、出展そのものを断念しました。

そして 2021 年から制作を開始したのが、この SSD 同人誌になります。前振りが長くなりましたが、毎年この同人誌では前年の出展の様子を記録も兼ねて振り返っています。2023 年はスポンサー出展をお休みしましたので、2022 年の出展を振り返りたく思います。



図 1: これまでに発行した同人誌

2019 年以來 3 年ぶりの現地参加となった Maker Faire Tokyo 2022



図 2：データセンター向け等の SSD 製品



図 3：SSD が支える社会が覗ける AR 模型

MFT2022 で、3 年ぶりに東京ビッグサイトに帰ってくることができました。社名がキオクシアとなって初の現地参加でした。展示内容は当社の SSD 製品をメインとし、一般ユーザーがあまり目にするものない、データセンターなどで使用される業務用の SSD を盛大に並べさせていただきました(図 2)。また、スマートフォンを使って「フラッシュメモリと SSD が支える社会」を覗ける AR 模型(図 3)を展示し、来場者に体験いただきました。

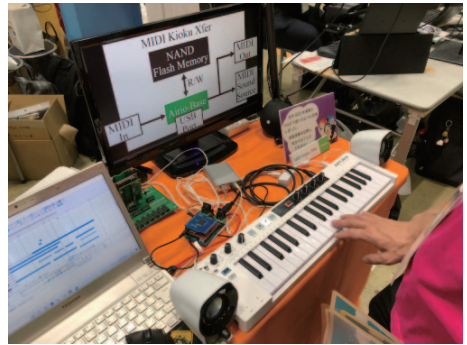


図 4：MIDI Kioku Xfer 展示の様子

このほかブースでは、SSD 同人誌 1 号に掲載した自作 SSD 基板を使った作品の展示も行いました。自作 SSD 基板とは、メモリコントローラのプログラムをユーザーが自作して、基板に搭載された NAND 型フラッシュメモリのチップの動作を試みるものです。2022 年は In-Storage Computing をテーマに、にちかさんに MIDI Kioku Xfer という作品(図 4)を作っていただきました。通常 MIDI 規格対応の楽器を使って演奏したり録音したりするにはパソコンを活用しますが、この作品では自作 SSD 基板が PC の代わりに音楽の再生、録音をすることができます。ブースでは MIDI Kioku Xfer の実物を来場者に実際に触っていただけるようにしました。

2022 年は久しぶりの出展でしたが、キオクシアのブースには 2 日間で約 2,000 人の方にお越しいただきまして、大変盛況でした。お越しいただいた皆様、ありがとうございました。

SSD 同人誌 2 号の配布

2022 年は、前年に発行した 1 号に続いて SSD 同人誌 2 号 (図 5) を制作し、キオクシアブースにて約 2,000 部を配布しました。この 2 号では 1 号に掲載の自作 SSD 基板「Pico SSD」を改良し、M.2 の SSD に形状を似せた新たな基板、「JISC-SSD (Jisaku In-Storage Computation SSD)」が株式会社クレイン電子より登場しました。この JISC-SSD の関連記事や、メモリコントローラの大切なお仕事である、化けてしまったデータを特定して訂正する「誤り訂正符号 (ECC)」のまんがと解説記事を掲載しました。なお、先述の MIDI Kioku Xfer の解説記事も掲載しております。

また、1 号から連載を開始したまんが「データセンターてんこ」では、SSD の外形規格である「フォームファクター」についてご紹介しました。様々なフォームファクターの形状をわかりやすく整理して説明した記事はあまり見かけないので、SSD 同人誌ならではの企画としてお楽しみいただける内容になったと思います。

さらに 1 号に続いて「大容量 SSD と OpenStreetMap その 2」と題して、「世界単位の単一の地図データベース」である OpenStreetMap を用いた、当社製 SSD「KIOXIA XD6 Series」のベンチマーク実験記事を寄稿いただきました。一方、フラッシュメモリに関する学習記事として、太陽の核融合反応に必要な量子トンネル効果が、フラッシュメモリにおいてデータを記憶する際に利用されているという解説記事「太陽が輝くとき SSD も輝く」を掲載しました。こちらはその後「光る太陽と記憶するフラッシュメモリ」と題して動画が公開されまして、アニメーションでフラッシュメモリのデータの記憶の仕組みを学ぶことができます。動画は下記の 2 次元バーコードからご覧になれます。

SSD 同人誌 2 号の冊子版は 1 号と同様の 3,500 部を印刷し、MFT 会場だけでなく秋葉原の店舗でもお配りしました。わざわざ秋葉原に足を運んで冊子を入手してくださいました方が、ちらほらいらっしゃったようです。ありがとうございました。SSD 同人誌 2 号は 1 号と共に以下の 2 次元バーコードから PDF 版をご覧いただけますので、自作 SSD にご興味のある方はぜひご覧ください。



「光る太陽と記憶するフラッシュメモリ」の動画



SSD 同人誌の PDF 版



図 5 : SSD 同人誌 2 号

シリコンウェハーうちわの第二弾が登場！

この年のもう1つの目玉企画(?)は、2019年にお配りしました直径30cmの実物大シリコンウェハーうちわの改良版の配布です。2019年はMFT会場で2日間にわたって約2,000枚を配布し、SNSでちょっとだけ話題を呼びました。

しかし、2019年版のうちわ(図6)は個人的にやや気になる部分がありました。それはシリコンウェハーの虹色の輝きを印刷で表現したことでした。そもそもウェハーが虹色に見える理由は、シリコンウェハー表面に作られた非常に微細な配線パターンによって、光が干渉するからです。とはいえ、本物のウェハーの輝きを紙のうちわで再現できるはずがありません。それで虹色の印刷にしたのですが、もう少しだけリアルに近づけたいという欲が出てしまいました。

そこで、本物のウェハーに見られる虹色の輝きをホログラムフィルムで再現できないものか、と考え始めました。ところが、2019年に印刷をお願いした業者では対応できないとのことでした。そうなりますと、表面にホログラムフィルムを貼ることのできる印刷業者を探し出すしかありません。ところがこれがなかなか見つからず、インターネットで検索しても、30cmのうちわを印刷してくれる業者がそもそもまれでした。

どうにもこうにも行き詰ったころ、モノづくりが好きな人たちが集まる浅草橋の某所に以前お誘いいただいたことを思い出しました。もしかしたらヒントがあるかもしれないと思い、再びそちらにお伺いしたところ、ありがたくも対応してくれそうな業者を教えてくださいました。翌日早速問い合わせしてみましたら、様々なパターンのホログラムフィルムを貼ることができるというではありませんか！

こうして、うちわの色を元々のウェハー表面の色に寄せたデザインにして、その上に無地のホログラムフィルムを貼った「シリコンウェハーうちわ Ver.2」(図7)が出来上がりました。このうちわは大学の先生や半導体を研究している学生さんにとっても好評で、とある研究室では部屋がこのうちわだらけになっていました。

今年はVer.3となる当社の最新のフラッシュメモリ(第8世代 BiCS FLASH™)のウェハーうちわを、この記事執筆時点で製作中です。

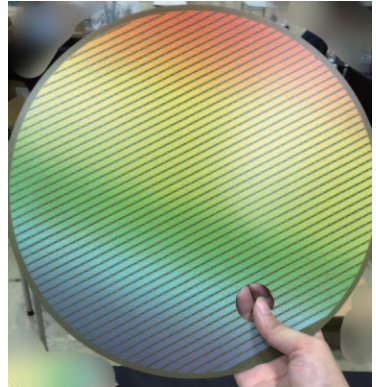


図6：2019年に配布したうちわ

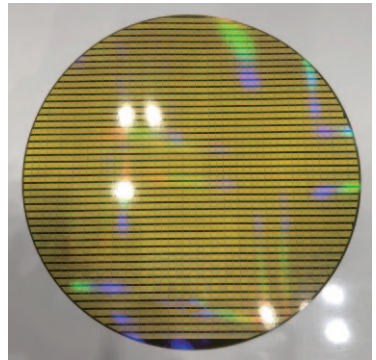


図7：シリコンウェハーうちわ Ver.2

そして 2023 年は初の一般参加へ・・・

2023 年は、諸般の事情でスポンサー出展ができませんでした。このとき脳裏をよぎったのが、2019 年の FlashAir 同人誌の最後の配布のとき、同人誌をもらいに来てくださった方から、「やめないでほしい」とお声がけいただいたことでした。その声は SSD 同人誌制作の原動力になりました。

そこで同人誌の発行を継続するために、有志で一般サークルとして参加し、自費で同人誌を印刷して配布することを考えました。特別編となる 2023 年の同人誌企画は「FlashAir が好きだったかつての執筆陣による、当時の思い出と裏話を語る番外編」とし、7 冊目の FlashAir 同人誌「FlashAir Dousoukai」を制作しました(図 8)。これをサークルの自費で 2,000 部印刷し、MFT 会場と秋葉原にて配布しました。なお、冊子は予算の関係で白黒印刷となりましたが、PDF 版はカラーにて FlashAir 同人誌のサイトで MFT2024 開催に合わせて公開予定です。

ところで会場のブースでは、FlashAir の思い出を FlashAir 型の付箋紙に書いて貼っていただきました(図 9)。FlashAir 時代にお世話になった方がたくさんお越しくださいました。ありがとうございました。

最後にネタですが、デイリーポータル Z さんのブースで「AI が作った架空の賞状で表彰する」という企画があり、今回の有志による出展に対し、「起死回生の星賞」をいただきました。「困難な状況下でも未来への希望を失わず、高い創作意欲で Maker Faire に出展し、自費で同人誌を出版したことに対し、起死回生の星賞を贈呈いたします。」とのことでした。ありがとうございました！



図 8 : FlashAir Dousoukai

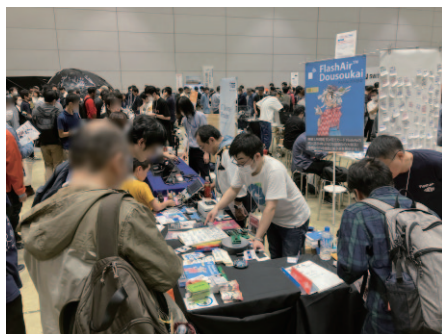
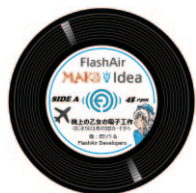


図 9 : MFT2023 のサークル出展の様子



Pochio (@i_love_nintendo)

元自称 FlashAir 芸人です。今年はうちの小学生の自由研究がてこずりまして、原稿の執筆が遅れてしまいました。小学生の自由研究はどこまで手を出していいのか、とても悩みますね。

こんなところにも SSD 同人誌が！

余熱

SSD 同人誌も早いもので 3 冊目となりました。ありがたいことに、多くの反響をいただき、さまざまな場所で取り上げていただく機会が増えました。今回は、SSD 同人誌が紹介された記事やイベントについてご紹介したいと思います。

サイボウズ・ラボのブログで自作 SSD・同人誌を紹介頂きました！

サイボウズ・ラボの内田さん (@uchanos) が開催した「第 1 回 自作 CPU を語る会」にて、にちかさん (@lxacas) が「自作 CPU で動かす自作 SSD (SSD 同人誌 1 号 p. 22)」を展示したところ、内田さんに興味を持っていただき、電子の森ラジオに呼んで頂きました。電子の森ラジオは電子工作とプログラミングに関するポッドキャストで、内田



図 1：インタビューの様子

さんがゲストの方と 30 分～1 時間程度みっちり技術的な話をしています。X で話題になった人などをゲストに呼んでいて、とても聞きごたえのあるポッドキャストです。

今回は、その電子の森ラジオにキオクシア社員としてインタビューを受けました (図 1)。インタビューには、自作 SSD の発案をした余熱、初期の FW 実装を担当した村口さん、そして応用デモを作成し展示を行ったにちかさんの 3 人で参加しました。インタビューでは、自作 SSD の構想からこれまでの経緯、そして今後の展望について話をしました。そのインタビューの様子をサイボウズ・ラボのブログにまとめて頂きました。

Vket にて SSD 同人誌を配布しました！

2023 年 7 月 15 ～ 30 日に開催されたバーチャルマーケット 2023 Summer (Vket) にキオクシアが出展し、その中で SSD 同人誌を配布して頂きました。Vket は VR 空間上で行われる展示会で、世界中から 100 万人を超す来場者が参加するイベントです。

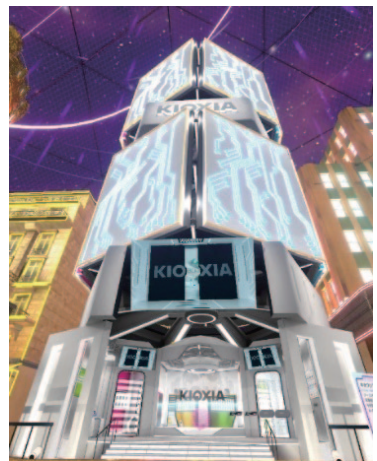


図 2：キオクシアブース外観

1 自作 SSD についてキオクシアの方に伺いました <https://blog.cybozu.io/entry/2024/01/11/160604>

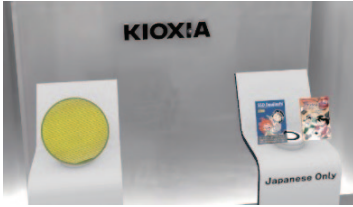


図 3: SSD 同人誌配布の様子



図 4: 偉い人の視察の様子

キオクシアブース(図2)内にてSSD同人誌を配布して頂きました(図3)。PDF版へのリンクになっていたようです。キオクシアの偉い人もVRヘッドセットを付けて視察していただきました(図4)。

なお、ブース内では「VR謎解き体験」というイベントを行い、クリアするとSSDの3DCGモデルをダウンロードすることができました。これは今後の漫画でも活かせるかも。



図 5: 配布された SSD モデル

(おまけ) CP+にてヨネセミナーを行いました!

SSD同人誌と直接関係はないのですが、いわゆるヨネセミナー(余熱のセミナー)のネタです。2024年2月に行われた「CP+2024」にて、キオクシアのSDメモリーカードを紹介するプレゼンを行いました。キオクシアのSDメモリーカードはいろいろな色がラインナップされているので、戦隊モノのテイストで紹介させていただきました。コロナになってセミナーもすっかりなくなっていたのですが、最近また復活してきたようで、MFTもですけどまた皆様にお会いできるのが楽しみです!



図 6: CP+ セミナー

2 今回の巻頭言を書いてくださったAIばばさんです。



余熱 (@yone2_net)

今回も執筆時間の捻出に協力してくれた妻に感謝。
今年も第2子も生まれていろいろとバタバタしてますが楽しく過ごしてます。

はじめに

SSD 同人誌でもこれまで取り上げられてきた株式会社クレイン電子の JISC-SSD (Jisaku In-Storage Computation SSD) は、NAND 型フラッシュメモリを手軽に扱うことができる優れたツールです。今回は、公開されている NAND 型フラッシュメモリ (当社 SLC NAND TC58NVG0S3HTA00) のデータシート (以下、データシート) を見ながら JISC-SSD を使って NAND 型フラッシュメモリを操作する基礎を解説していこうと思います。環境は Arduino IDE を使います。Arduino IDE の設定は GPS_NMEA_JP さんの ¹note に詳しく書かれています。

それでは、データシートに記載された基礎的な動作を Arduino 環境で実行してみましよう。

NAND 型フラッシュメモリのピン設定

前準備として、NAND 型フラッシュメモリと Raspberry Pi RP2040 は GPIO の 0 ~ 15 で接続されているので、NAND 型フラッシュメモリの各ピンを GPIO 番号に変換しておくとなかなか便利です (図 1)。

これら 16 本の接続を使って NAND 型フラッシュメモリを操作するのですが、基礎となる操作方法は 4 種類だけで、それらを組み合わせて様々な動作を実現します。

```
//NAND pin to GPIO No.
int IO1= 0;
int IO2= 1;
int IO3= 2;
int IO4= 3;
int IO5= 4;
int IO6= 5;
int IO7= 6;
int IO8= 7;
int CE0= 8;
int CE1= 9; //No NAND
int CLE=10;
int ALE=11;
int WP =12;
int WE =13;
int RE =14;
int RB =15;
```

図 1 : GPIO ピンの割当

1 <https://note.com/gpsnmeajp/n/n7a85a5118065>

4つの操作

1. Command Input

1つ目は“コマンド”です。データシート p. 7 にあります。CLE=High、ALE=Low、IO を1Byteの所望のコマンドにした状態でWEをLowからHighにすると、NAND型フラッシュメモリはコマンドとしてIOを取り込みます(図2)。

これを関数にするとこんな感じです(図3)。digitalWrite(WE, HIGH)の瞬間にNAND型フラッシュメモリにコマンドが入ります。

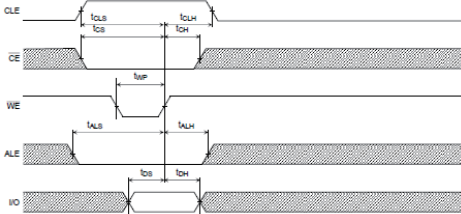


図2：タイミングチャート

```
void COM(int Command){
    digitalWrite(CLE, HIGH); digitalWrite(ALE, LOW);
    digitalWrite(WE, LOW);
    IO_OUTPUT();
    IO_SET(Command);
    digitalWrite(WE, HIGH);
    digitalWrite(CLE, LOW);
    IO_INPUT();
}
```

図3：プログラム例

2. Address Input

2つ目は“アドレス”です。これはコマンドにおけるCLEをALEと入れ替えただけで完了します(図4)。データシートでは4Byte連続で入力するフローで説明されていますが、1Byteで区切った方が後々便利です(図5)。

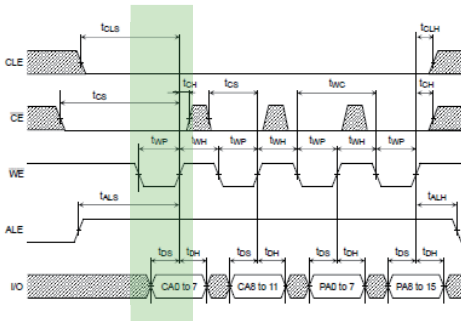


図4：タイミングチャート

```
void ADD(int Address){
    digitalWrite(CLE, LOW); digitalWrite(ALE, HIGH);
    digitalWrite(WE, LOW);
    IO_OUTPUT();
    IO_SET(Address);
    digitalWrite(WE, HIGH);
    digitalWrite(ALE, LOW);
    IO_INPUT();
}
```

図5：プログラム例

3. Data Input

3つ目は“データ入力”です。CLEもALEもLowのときにIOを1Byteの所望のデータにした状態でWEをLowからHighにすると、NAND型フラッシュメモリはデータインプットとしてIOを取り込みます(図6)。データシートでは連続して1ページサイズ分インプットするようなフローが書かれていますが、1Byteのみインプットする関数を作っておきます(図7)。

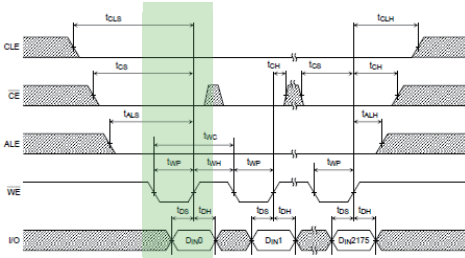


図6：タイミングチャート

```
void DATA_IN(int Din){
    digitalWrite(CLE, LOW); digitalWrite(ALE, LOW);
    digitalWrite(WE, LOW);
    IO_OUTPUT();
    IO_SET(Din);
    digitalWrite(WE, HIGH);
    IO_INPUT();
}
```

図7：プログラム例

4. Data Output

4つ目は“データアウトプット”です。CLEもALEもLowのときにREをLowからHighにすると、NAND型フラッシュメモリはデータをIOへ出力します(図8)。データインプットと同様に1ページサイズ分連続でアウトプットするフローですが、1Byteのみアウトプットし、戻り値として取得するような関数にします(図9)。

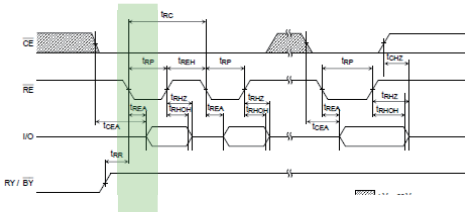


図8：タイミングチャート

```
int DATA_OUT(){
    digitalWrite(CLE, LOW); digitalWrite(ALE, LOW);
    digitalWrite(RE, HIGH);
    IO_INPUT();
    digitalWrite(RE, LOW);
    int DATA=IO_GET();
    return DATA;
}
```

図9：プログラム例

これら4つの関数でNAND型フラッシュメモリを操作する準備は完了です。

Ready/Busyピン

操作のためのシーケンスではありませんが、NAND型フラッシュメモリは自分の状態を“Ready/Busy”ピンを使って知らせてくれます。「NAND型フラッシュメモリが内部で動作しており、新たな操作は受け付けません」というときにはNAND型フラッシュメモリ側からReady/BusyピンをBusy状態=Lowにしてきます。(データシート p. 22 参照)

そこで、Ready/Busyピンを監視して、Busyの場合はReadyになるまで待つ関数も用意しておきましょう。

図 10 は、Ready/Busy ピンが Low である間はループを繰り返す関数です。一応、Busy スタックした場合に無限ループを防ぐため、タイムアウト設定を入れてあります。

```
void MATCH(){
  int Timeout=5000;
  for(int count=0; count<Timeout; count++){
    if(digitalRead(RB)){break;}
    else if(count==(Timeout-1)){Serial.println("Busy Timeout"); break;}
    delayMicroseconds(1);
  }
}
```

図 10 : プログラム例

実践 ~ Read 編~

それでは試しに NAND 型フラッシュメモリの特定のアドレスのデータを読んでみましょう。電源 (VCC) を投入してから最初に操作するまでに “Power-on Reset” するのをお忘れなく。データシート P. 40 より (図 11)。

(2) Power-on Reset

The following sequence is necessary because some input signals may not be stable at power-on.



図 11 : Power-on Reset の説明

図 12 のように、CE 選択もおきましょう。と言っても JISC-SSD はデフォルトでは CEO しか NAND 型フラッシュメモリが接続されていませんが。(君は自信があるなら KIOXIA 製 NAND 型フラッシュメモリを自分で購入して CE1 にはんだ付けしてもいい、しなくてもいい。)

```
void setup(){
  Serial.begin(115200);
  digitalWrite(CE0, LOW); //CE select
  COM(0xFF); delayMicroseconds(1000);
}
```

図 12 : setup 関数

NAND 型フラッシュメモリからのデータ読み、つまり Read 動作はデータシート p. 10 に記載されています。複雑っぽいフローですが、分解すると今まで作った基本 4 動作の組み合わせであることがわかります。

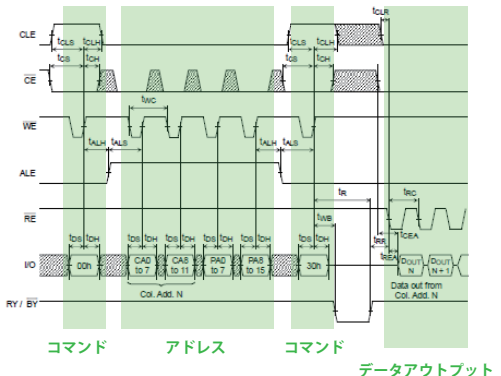


図 12 : Read 動作

次にアドレスの指定方法を説明します。データシートの p. 23 を見てみます (図 14)。

	I/O8	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	
First cycle	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0	CA0 to CA11: Column address
Second cycle	L	L	L	L	CA11	CA10	CA9	CA8	PA0 to PA5: Page address in block
Third cycle	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	PA6 to PA15: Block address
Fourth cycle	PA15	PA14	PA13	PA12	PA11	PA10	PA9	PA8	

図 14 : Addressing 表

Read フローでアドレスが 4Byte 繰り返されているのは、この表の 1st ~ 4th サイクルに対応しています。Column アドレスは一旦置いておき、目的の Page と Block を 2 進数に変換し、指定のサイクル/bit に当てはめていきます。

実際の Read 動作

それでは、試しに Block 51、Page 15 に対して Read しましょう。Block と Page を 2 進数にすると、51 → 110011、15 → 1111 なので、アドレス 4Byte は以下になります。

- 1st サイクル : 0x00、• 2nd サイクル : 0x00
- 3rd サイクル : 0xCF、• 4th サイクル : 0xC

図 15 のようなプログラムで Block 51、Page 15 に対する Read ができます。

```
COM(0x00); ADD(0x00); ADD(0x00); ADD(0xCF); ADD(0xC); COM(0x3
MATCH());
for(int col=0; col<2176; col++){
  int Dout=DATA_OUT();
  Serial.print(Dout, HEX); Serial.print(",");
}
```

図 15 : Block=51、Page=15 を Read するプログラム例

さて、何が読み出されたでしょうか。株式会社クレイン電子の JISC-SSD を購入した直後の状態なら、きっと全部 “FF” が読み出されたと思います。この製品は出荷状態では (不良でない) 全ての Block は消去状態 = 全 bit 1 になります。

今回はここまでです。

ここまでのやり方が理解できれば、データシートに記載されている機能を使うのは難しくないでしょう。今回の記事はページ数の都合上一部関数の説明を省略しています。コメント入りのプログラムをクレイン電子の HP にアップロードして頂いたので、よろしければ参考としてお使いいただけると嬉しいです。

<https://crane-elec.co.jp/products/vol-28/>



宮内

7 年ぶりに真面目な記事を書きました。

過去に FlashAir™ 同人誌に関わったりしていた関係もあり、ひょんなことから JISC-SSD (Jisaku In-Storage Computation SSD) を入手させていただきました。JISC-SSD は、キオクシア社の SSD の扱いを学習するための拡張ボードとしてクレイン電子さんから発売されているボードです。

<https://crane-elec.co.jp/products/vol-28/>

ボードには RP2040 が乗っており、ほぼ Raspberry Pi Pico 互換で動かすことができます。搭載されている NAND(TC58NVGOS3HTA00) のデータシートは以下から見るができます。

https://www.kioxia.com/ja-jp/search/parametric/slc-nand.param_104.html

本ページは note に記載したものを紙面に収まるように日本語で説明するように大幅に再編集したものです。元記事ではサンプルソースコードを中心に説明しています。

<https://note.com/gpsnmeajp/n/n7a85a5118065>

Lチカ

まずは何より Lチカでしょう。Raspberry Pi Pico の開発環境を使用しますので、環境構築は以下 adafruit の「Program RP2040 in Arduino」を参考に実施してください。

<https://learn.adafruit.com/rp2040-arduino-with-the-earlephilhower-core?view=all>

(Arduino 公式よりもこちらのほうが最新を使用できるようです。)

BOOL SEL ボタンを押しながら、RESET ボタンを押すと書き込み待ちになりますので、Arduino IDE から書き込みましょう。

なお、GPIO25 をトグルすることで、Lチカができるので動作確認に便利です。

通常の Arduino 同様、Serial ライブラリを使うことでシリアル出力ができます。

最初シリアルポートが出てこないかもしれませんが、一度シリアルポートを使うスケッチを書き込んでから起動すると、シリアルポートが選べるようになっているので、シリアルモニタを起動してください。

ID 読み出し

さて、フラッシュメモリと通信してみましょう。過去の SSD 同人誌では、リセットからの Busy 信号を観測していますが、オシロやロジアナがない場合、これはできません。

まあまあ安い価格で USB ロジアナが手に入る時代になったのでそれを利用してもよいのですが、ピンを引き出すのも面倒ですし、サクッとコードを書いて通信してしましましょう。案外難しくなく通信ができます。

GPIO 0-7 がアドレス・データ線、8-15 が制御線ですので、データシートを見ながらいい感じに制御していきます。

各ピンを初期化後、タイミング指定に沿ってリセット、その後コマンド 90、アドレス 0 を入れて 5 回 Read すると、(Maker code)98, (Device code)F1, (Chip Number, Cell Type)80, (Page Size, Block Size)15, (District Number)72 と読み取ることができます。

この値は、データシート (DST_TC58NVGOS3HTA00-TDE_EN_31435.pdf) の p. 35 に記載されているので確認してみてください。

フラッシュメモリの読み書き

正常に通信できることがわかったら、フラッシュメモリの読み書きをしてみましょう。フラッシュメモリとの通信のため、ID 読み出しで使用した

- I/O 初期化
- Flash メモリリセット
- コマンド転送
- 1 ワードアドレス転送
- 逐次データ取得

に加え、更に以下のプリミティブ操作を実装する必要があります。

- I/O ビジー待ち (Ready/Busy ピンの監視)
- 逐次データ転送
- 4 ワードアドレス転送
- 2 ワードアドレス転送
- ステータス取得

ID 読み出しと異なり時間のかかる処理になるため、リセット後や各種転送・取得 (一操作ごと) のあとは、必ずデータ・アドレス線を入力に戻し、I/O ビジー待ちを行う必要があります。

プリミティブ操作を実装できたら、下記の具合で Flash に対する読み書きができるようになります。(タイミング等はデータシートを参照してください。)

ページの読み込みは

- コマンド 0x00 (Read)
- 4ワードアドレスでページを指定
- コマンド 0x30 (読み出し開始)
- ページサイズ分のデータ読み込みの実行

ページの書き込みは

- コマンド 0x80 (Program)
- 4ワードアドレスでページを指定
- ページサイズ分のデータ書き込みの実行
- コマンド 0x10 (書き込み・ベリファイ実行・結果確認)
- (ビジー)
- コマンド 0x70 (ステータスの確認)
- ステータス取得

ブロックの消去は

- コマンド 0x60 (Erase)
- 2ワードアドレスでブロックを指定
- コマンド 0xD0 (消去の実行)
- (ビジー)
- コマンド 0x70 (ステータスの確認)
- ステータス取得

1ブロック 64 ページです。なお、消去しても読み込んで 0 が帰って来るブロックは不良ブロックですので、そこに対する操作はしないよう事前に走査しておく必要があります。

おわりに

冒頭に記載しました通り、この記事のサンプルソースコードは note に掲載しています。理解容易性のため、非常に低速なことを覚悟の上で Arduino 言語のみで記載するように心がけていますので、動かしてみたい方はぜひ御覧ください。

コピペで動くと思いますので、動作確認にもご利用いただけたと思います。

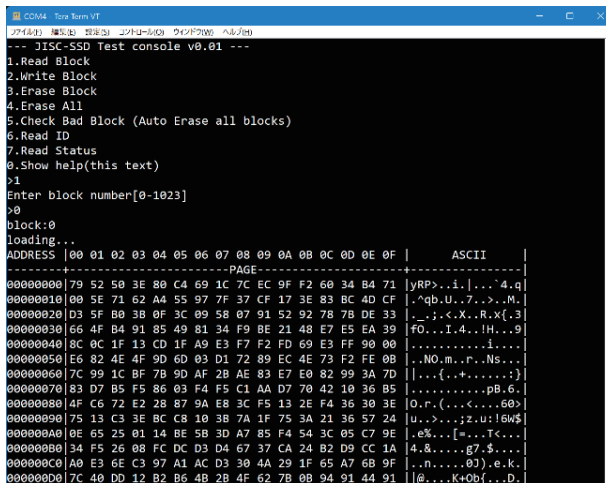
また、Github にて、

- ブロックの読み書き、イレース、不良ブロックの調査などを行うコンソールツール
- 誤り訂正あり / なしの USB-SSD として機能させるサンプル
- USB-SSD として動作しつつ、コントローラから読み書きの内容で LED を点滅させる in-storage computing(?) サンプル

も公開しておりますので参考にしていただけますと幸いです (図 1)。

こちらに関してはある程度速度が出るようにチューンしております。

https://github.com/gpsnmeajp/jisc_ssd



```

COM4: Prog Run V17
JISC-SSD Test console v0.01 ---
1.Read Block
2.Write Block
3.Erase Block
4.Erase All
5.Check Bad Block (Auto Erase all blocks)
6.Read ID
7.Read Status
8.Show help(this text)
>1
Enter block number[0-1023]
>0
block:0
Loading...
ADDRESS |00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F |          ASCII
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
00000000 |79 52 50 3E 80 C4 69 1C 7C EC 9F F2 60 34 B4 71 |yRP>.i.|...4.d
00000010 |00 5E 71 62 A4 55 97 7F 37 CF 17 3E 83 8C 4D CF |^qb.U.7.>.>.M.
00000020 |03 5F 80 38 0F 3C 09 58 07 91 52 92 78 78 DE 33 |.;>.x..R.x{.3
00000030 |66 4F B4 91 85 49 81 34 F9 BE 21 48 E7 E5 EA 39 |f0...I.4..!H..9
00000040 |8C 0C 1F 13 CD 1F A9 E3 F7 F2 FD 69 E3 FF 90 00 |.....i...
00000050 |E6 82 4E 4F 9D 60 03 D1 72 89 EC 4E 73 FE 0B |..No.m.r.Ns...
00000060 |7C 99 1C BF 7B 9D AF 2B AE 83 E7 E0 82 99 3A 7D ||...{...+...:}|
00000070 |83 D7 B5 F5 86 03 F4 F5 C1 AA D7 70 42 10 36 B5 |.....pB.6.
00000080 |4F C6 72 E2 28 87 9A E8 3C F5 13 2E F4 36 30 3E |0.r.(...<...60>
00000090 |75 13 C3 3E 8C C8 10 3B 7A 1F 75 3A 21 36 57 24 |u...>...z.u:|6w$
000000A0 |0E 65 25 01 14 BE 5B 3D A7 85 F4 54 3C 05 C7 9E |.e%...[...T<...
000000B0 |34 F5 26 08 FC DC D3 D4 67 37 CA 24 B2 D9 CC 1A |4.&...g7.$...
000000C0 |A0 E3 6E C3 97 A1 AC D3 30 4A 29 1F 65 A7 6B 9F |..n....0J).e.k.
000000D0 |7C 40 DD 12 B2 B6 4B 2B 4F 62 7B 0B 94 91 44 91 ||@...K+0B(...D.

```

図 1. Github に掲載のデバッグ用コンソールツール



GPS_NMEA_JP (@Seg_faul)

FlashTools Lua Editor など色々作ってました。
今でも FlashAir 開発者向け非公式 wiki を運営中です。
最近はバイクの免許を取って走り回ってます。

電子ペーパーディスプレイ搭載 SSD

@isariyokurari

microSD や USB メモリの中身が分からなくなった事はありませんか？私があります。例えば右の図 1 は、OS インストール用に用意した USB メモリなのですが、どちらがどの OS 用か分からなくなってしまいました。対策として、付箋やシールに手書きでメモして貼付することも考えましたが、手元にペン等の資材が必要になります。資材不要なソリューションを考えた結果、「ボリュームラベルを電子ペーパーディスプレイに表示する」に辿り着きました。ボリュームラベルは、Windows 10 のエクスプローラのナビゲーションウィンドウから変更可能なので、ペン等の資材は必要ありません。また、電子ペーパーディスプレイは、通電していない間も表示内容を保持し続けるので、PC から取り外した後も中身を識別できそうです。そこで今回、株式会社クレイン電子さんの JISC-SSD (Jisaku In-Storage Computation SSD 学習ボード) に、Crescent さんの e-Paper I2C モジュールを繋いで、電子ペーパーディスプレイ搭載 SSD を作成してみました。本稿では、その内容を紹介したいと思います。



図 1: どちらに何が入ってる？

JISC-SSD を USB-SSD として PC に認識させる

GPS_NMAE_JP さんのレポジトリ¹の `msc.ino` を書き込むことで JISC-SSD を USB-SSD として PC に認識させることができます。同レポジトリの `ssd.ino` でバッドブロックを確認、その情報を元に `msc.ino` の `bad_block_replace()` の内容も書き換えてから `msc.ino` を JISC-SSD に焼き込みました。これで無事、JISC-SSD が USB-SSD として PC に認識されるようになりました。

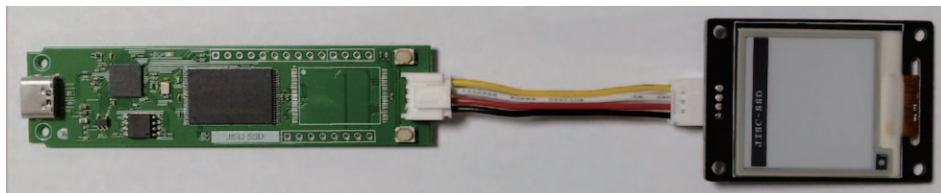


図 2: 電子ペーパーディスプレイ搭載 SSD

1 https://github.com/gpsnmeajp/jisc_ssd commit : 3c2d4da

ボリュームラベル更新タイミングとその内容の取得

PC上でUSB-SSDのボリュームラベルが変更された場合に、USB-SSD側でそのことを知り、変更後のボリュームラベルの内容を知る必要があります。雑誌 Interface の特集を参考に、`m_sc.ino` の `m_sc_read()` で `lba` が 0 の時にルートディレクトリブロックを算出、`m_sc_write()` では `lba` がルートディレクトリブロックの先頭を指しているかどうかとその中身を確認することで、ボリュームラベルの変更タイミングとその内容を知ることができるようになりました。

電子ペーパーディスプレイにボリュームラベルを表示する

JISC-SSD に Grove コネクタを実装し、専用ケーブルで電子ペーパーディスプレイと接続しました。概観を図 2 に示します。今回使用する電子ペーパーディスプレイは Crescent さんのリポジトリ³でサンプルコードが提供されています。I2C のポートを指定するコード `Wire.setSDA(20)` および `Wire.setSCL(21)` を追加して使用しました。前述のボリュームラベル更新タイミングで、電子ペーパーディスプレイにボリュームラベルを表示するよう調整して完成しました。

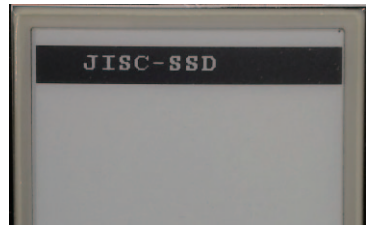


図 3: 更新前後の電子ペーパーディスプレイ

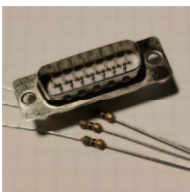
動作確認

Windows 10 のエクスプローラのナビゲーションウィンドウ上で USB-SSD のボリュームラベルを JISC-SSD から DEKITA に変更すると、電子ペーパーディスプレイの内容は、図 3 の通り更新されました。更新後は USB-SSD 本体を PC から切断しても更新後の内容 DEKITA が表示され続けています。欲しいものデキタ!

電子ペーパーディスプレイに表示する内容をボリュームラベルで管理するところが拘りポイントでした。楽しんでいただけましたら幸いです。

² CQ 出版 Interface 2010 年 9 月号 FAT ファイル・システムでファイルを読み書きしよう

³ https://github.com/meerstern/e-Paper_I2C_Module commit : 4586e5b



@isariyokurari

イサリです。ハードウェアとソフトウェアの両方にまたがる開発が好物です。今回の開発も楽しめました。

SLC NAND を使用したセンサノード初号機の開発

高田

JISC-SSD ボード（キオクシアの SLC NAND が搭載された株式会社クレイン電子の Jisaku In-Storage Computation SSD 学習ボード）を使用して、気温、湿度、気圧、時刻が測定可能な環境把握のためのセンサノード初号機（以降単にセンサノードと表記）を作成しました。当時、キオクシアとコラボを行っている大学の学生がパタゴニアでの実地実験を行うとのことで、作成したセンサノード初号機を託し、屋外の実験場に一晚放置して頂きました。時間が限られている中突貫で制作したため改善点は多くありますが、今回は作成したセンサノードの概要と、実験の顛末をお伝えできればと思います。

センサノードの構成

使用したモジュール

今回使用したモジュールは下記の 2 種類です。

- **SHILLEHTEK PRE-SOLDERED BME280 3.3V**

ボッシュ社の BME280 を搭載したセンサモジュール。気温、湿度、気圧の 3 つの環境情報を測定可能

- **ACEIRMC DS1302 Real Time Clock Module**

アナログデバイス社の DS1302 を搭載した RTC モジュール。秒、分、時間、日、週、月、年を取得可能。モジュール本体がボタン電池で駆動するため、マイコンからの電源供給が途絶えた場合にも、時間のカウントアップが途切れない。

電源

屋外環境下にて使用するため、電源は太陽光充電が可能な持ち運びタイプで且つ JISC-SSD ボードが Type-C® で接続されるため、USB Type-C® が使用できるものを選択しました。

- **Hiluckey Solar Charger 25000mAh**

選択した電源は主にスマホやタブレットを充電する目的のものであるため、100 mA 程度の比較的大きな電流が流れていないと電源供給を止めてスリープモードに入る仕様となっていました。一方で、JISC-SSD ボード並びに接続されているセンサノードは、測定時 / 無測定時両方において 100 mA を大きく下回る電流しか流れません。よって、一定時間稼働後、電源がスリープ状態に入ってしまう現象が発生しました。そこで、PNP トランジスタを用いて 30 秒間に一度 1 秒間の電流を流し、電源がスリープモードに入らないように対策しました。電流は、JISC-SSD ボードに搭載されている RP2040 マイコンの電源ピンからではなく、JISC-SSD ボードに使用されているレギュレータから直接引くことで、電源を起動させ続けるのに十分な量を確保することができました。

RP2040 マイコンに関する設定やプログラムは GPS_NMEA_JP 氏による下記のサイトの記事を参考にしました。シリアルモニターでデバッグを行った際の動作例を掲載します(図2)。センサノードの動作内容としては、既にセンサデータが書いてある領域を読み進め、未だデータが書かれていないページを見つけます。そして、センサからのデータを取得し未書き込み領域にデータを書き込みます。

```

Output  Serial Monitor x
[Message (Enter to send message to 'Raspberry Pi Pico' on '/dev/cu.usbmodem101')]

14b,0f,0,128,14b,0,1f4,fb,30,fb,18,49,4,5,46,41
199,86,0,153,128,0,172,76,0,27,18,29,2,5,24,36
18,87,0,157,128,0,172,76,18,27,18,29,2,5,24,37
247,86,0,171,128,0,177,76,34,27,18,29,2,5,24,41
258,86,0,171,128,0,177,76,44,27,18,29,2,5,24,41
235,87,0,165,128,0,171,76,27,29,18,29,2,5,24,43
239,87,0,164,128,0,169,76,37,29,18,29,2,5,24,41
133,87,0,155,128,0,168,76,47,29,18,29,2,5,24,40
61,86,0,155,128,0,168,76,57,29,18,29,2,5,24,40
26,86,0,155,128,0,169,76,7,30,18,29,2,5,24,38
119,87,0,153,128,0,168,76,17,30,18,29,2,5,24,39
6,90,0,164,128,0,168,76,19,30,18,29,2,5,24,34
255,255,255,255,255,255,255,255,255,255,255,120
Page: 78
Temp: 23.37C Pressure: 1025.08hPa Humidity: 15.47%
Time: 24/2/29 18:50:52
  
```

保存済データ

未書き込みページ

センサデータ

図2：センサノードが動作した様子

センサノードの完成！

最終的に完成したセンサノードは下記です(図3)。移動時に回路が破損しないように、簡易的にレゴブロックで固定できるような構造を作成しました。また、屋外環境下で雨の可能性があったため、回路をむき出しにせずプラスチック容器に入れることにしました。

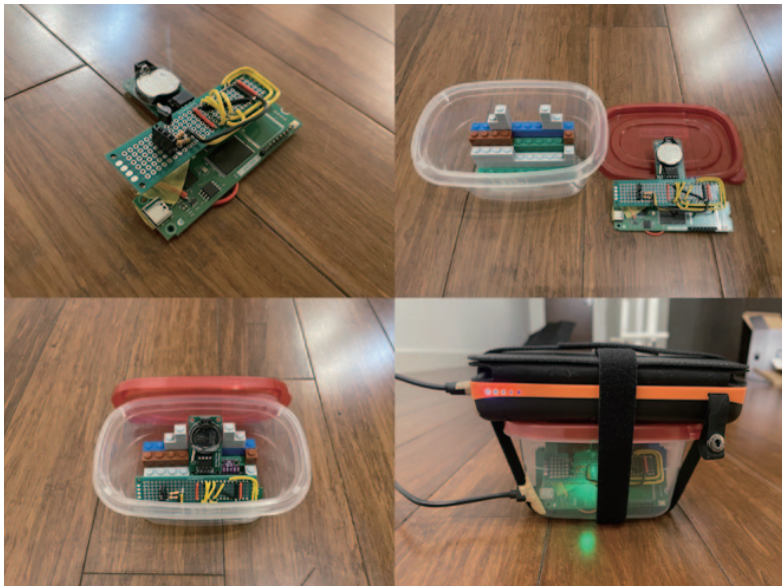


図3：完成したセンサノード初号機

1 <https://note.com/gpsnmeajp/n/n7a85a5118065>

センサノードがキツネに取られた！

実地実験では、パタゴニアの奥地にある草むらにセンサノードを一晩置いて頂きました。しかし、次の日の朝に研究員の方が確認したところ無くなっていたようで、キツネに取られた可能性が高いとのことでした。幸い、ケーブルで繋いでいた電源は置かれたままになっており、動物への影響は低度であるとのことでした。今回の反省点として、野生動物



が回路基板に興味を持つことを想定しておらず、GPS 機能を入れていなかったこと、電源を入れている間は発光する状態にしていたこと、また、基板の容器に透明のプラスチック容器を用いたことで野生動物が光に反応する可能性や、食べ物と勘違いする可能性を取り除けていなかったことが挙げられます。通常、野生動物がプラスチック容器を見て食べ物と認識する可能性は低いですが、センサノードを設置した地域では、人間の廃棄物を野生動物が漁ることがあるようで、原因の一部となったのかもしれませんが。また、デバイス自体を木に巻き付ける等の工夫も必要

であったと思います。一方で、今回のことは出張をされた研究員を始め、現地の生物系の研究者の方々も”通常キツネが人又は人が設置した見慣れないものに興味を示す可能性は低く、こういったことは初めてである”と仰っていたようで、想定することは難しかったものの、野生環境では起こり得ることとして反省すると共に大変勉強になりました。



今回の実地実験は想定外の結果に終わってしまいましたが、もしも再度機会を頂けましたら、2号機として基板はPCBから設計・制作を行うこと、更に多様なセンサを搭載し、より多くの環境情報を取得することでNAND型フラッシュメモリのユーザーデータ使用率を上げたいと考えています。また、NAND型フラッシュメモリーを用いたセンサデバイスが屋外環境下にて自動でデータを収集し続けることで、環境情報を扱う研究分野に貢献できれば尚良いと考えています。個人的には、JISC-SSDボードに搭載されているRP2040マイコンでの2コアを活かしたアプリ探索にも興味があります。皆様も是非、JISCボードでアプリを実装してみてください！



高田

普段はSSDやNAND型フラッシュメモリの中に関するお仕事をしているので、JISC-SSDボードから自社製品を使う経験は新鮮でした。

NAND 型フラッシュメモリで創るコンピュータ

にちか

今年は本当に暑いですね!にちかと申します。アツイといえば、最近はデジタルトランスフォーメーションや生成 AI などの分野がアツイですね。これらの分野では、サーバーの計算を担う CPU や GPU という部品が活躍しています。さて、フラッシュメモリはデータの記憶に用いられる部品ですが、もしこれでも計算ができてしまったらどうでしょうか?新しいブームが起きそうで、なんだかワクワクしませんか?そこで当記事は、NAND 型フラッシュメモリで 16bitCPU を実際に創り、コンピュータゲームを動かしてみました!

出来たもの

製作したハードウェアの外観を図1に示します。ハードウェアは、JISC-SSD (Jisaku In-Storage Computation SSD)¹ とタクトスイッチ (キーボード) で構成されます。株式会社クレイン電子製の JISC-SSD は Raspberry Pi Pico と NAND 型フラッシュメモリが一体となったデバイスで、難しいはんだ付けをすることなく電子工作が出来ます。JISC-SSD の詳細は、SSD 同人誌 2 号をご覧ください。今回は JISC-SSD の CPU とフラッシュメモリを用いて、新たな CPU を作りました。また、この CPU 上で 3D 迷路ゲームを動作させたときの画面を図2に示します。プレイヤーはキーボード操作で移動し、迷路からの脱出を目指します。

この CPU のアーキテクチャ設計は、「コンピュータシステムの理論と実装」³ という書籍で紹介されている Hack CPU を参考にしました。Hack CPU は、NAND ゲートというロジック素子 (図3) をいくつもつなぎ合わせることで組み立てられます。なお、NAND 型フラッ

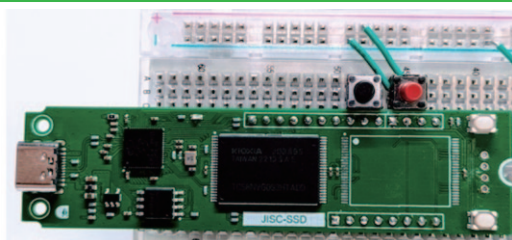


図1: 自作コンピュータの外観

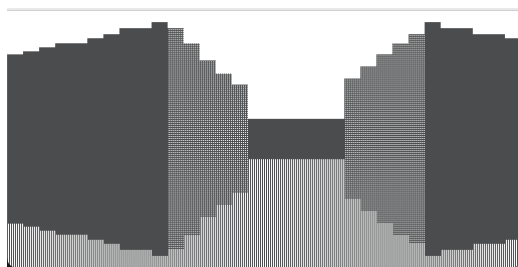


図2: 3D 迷路ゲームのスクリーン画像

1 <https://crane-elec.co.jp/products/vol-28/>

2 <https://github.com/Acedio/nand2tetris/tree/master/09/Bichromia> Copyright (c) 2016, Josh Simmons

3 Noam Nisan, Shimon Schocken 著、斎藤 康毅 訳

「コンピュータシステムの理論と実装」O'Reilly Japan (2015)

シュメモリと NAND ゲートは名前が似ていますが、用途が異なる別の回路です。具体的には、NAND 型フラッシュメモリは、メモリセルと呼ばれるデータ保存領域が沢山並んだ回路であるのに対し、NAND ゲート (図 3) は、表 1 のような計算を行う回路です。

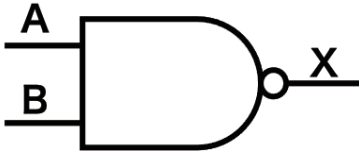


図 3: NAND ゲート

表 1: NAND ゲートの真理値表

入力 A	入力 B	出力 X
0	0	1
0	1	1
1	0	1
1	1	0

CPU を組み立てるには

なぜ NAND ゲートをつなぐことで CPU は作れるのでしょうか？ CPU は大きくて複雑な回路なので、想像しにくいかもしれません。ですが 1 箇所ずつ注目すると、実は AND ゲートや NOT ゲート、OR ゲートといった小さくて単純な回路の組合せで殆ど作られています。つまりこれら 3 つの論理回路の組み合わせで、皆さんが使う PC やスマートフォン並の CPU だって作ることができます。さらに図 4 のように NAND ゲートを並べると、1 種類だけで AND ゲートや NOT ゲート、OR ゲートと同じ計算ができます。以上から、もし NAND 型フラッシュメモリで NAND を計算できれば、それを繰り返すことで AND や NOT、OR も計算できるので、やがては CPU と同じ計算が可能だと分かります。

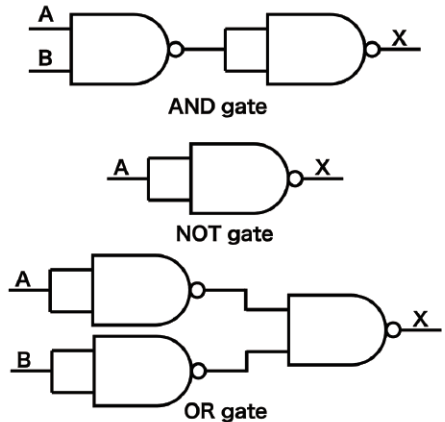


図 4: NAND ゲートの組み合わせ

メモリセルの動作

フラッシュメモリで NAND を計算する方法を説明するために、まずメモリセルの動作を説明します。メモリセルの構造を図 5 に示します。デジタルデータが "0" と "1" の 2 値で表現されることは多くの方がご存知だと思います。まずメモリセルはまっ更なとき、電荷蓄積

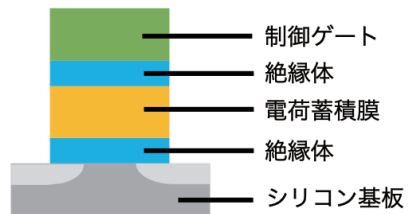


図 5: メモリセルの構造

膜に電荷がない状態で "1" を表します。もしこのセルに "1" が書き込まれた場合、メモリセルの制御回路は何もしないので、依然として "1" を表します。一方、もしこのセルに "0" が書き込まれた場合、制御回路は制御ゲートに電圧をかけ、電荷蓄積膜に電荷を貯めます。そうすると、このメモリセルは "0" を表すようになります。もしこのメモリセルを "1" に戻したい場合、"0" を書いた時と逆方向に電圧をかけて、電荷を抜きます。ちなみに NAND 型フラッシュメモリはメモリセルの密度を上げるため、常にブロックと呼ばれる範囲でまとめて電荷を抜きます。そのためアドレス N のデータだけを消去し、N+1 は残すというような細かい操作は難しく、ガベージコレクションなどの工夫が必要です。

NAND 型フラッシュメモリで計算するアイデア

やっとこの記事のアイデアを説明できます！先ほどメモリセルは、まっ更な時は "1" を表し、"1" を書き込めば "1" に、"0" を書き込めば "0" になると説明しました。では、"0" の状態で "1" を書き込むとどうなるのでしょうか？答えは、制御回路が何もしないため、"0" のままです。以上から、あるメモリセルに連続でデータを書き込んだ場合、表2のように動作します。

表 2: メモリセルに連続書き込み時の動作

1 回目の書込データ	2 回目の書込データ	読出データ
0	0	0
0	1	0
1	0	0
1	1	1

表2は、まるで書込データ同士の AND を計算したような結果ですね。つまり、メモリセルに2回書き込んで読み出すと、AND 演算が出来ます。そこで強引ですが図6のように、入力ピンの手前に NOT ゲートを挟んで読み出し値を反転すると、表3のように動作します。

表 3: 図 5 の回路で連続書き込み時の動作

1 回目の書込データ	2 回目の書込データ	読出データ
0	0	1
0	1	1
1	0	1
1	1	0

表3を見ると、表1と同じ結果になっているとわかります。以上から、図6の回路で NAND 型フラッシュメモリの読み書きを繰り返すと NAND 演算が出来ることが分かりました。これはつまり、フラッシュメモリの読み書きで AND も NOT も OR も計算できるということであり、やがては CPU と同じ計算が出来るということを意味します。

ソフトウェア構成と実行結果

自作した CPU のソフトウェア構成を説明します。開発フレームワークは、Pico SDK および Arduino フレームワークを利用して頂きました。また、NAND 型フラッシュメモリを読み書きするプログラムは、GPS_NMEA_JP 氏の JISC_ssd プロジェクト⁴を利用して頂きました。ゲーム画面は Processing で作成し、ノート PC とシリアル通信させて描画しています。

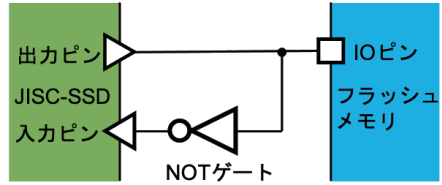


図 6: フラッシュメモリの読み書き回路

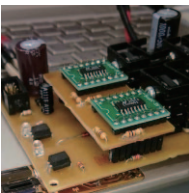
実行結果と考察

自作した CPU の動作周波数は 1.9 Hz で、非常に低速でした。これは足し算を 1 bit ずつ順番に繰り返して計算しているせいだと考えられます。実際、16 bit データ同士の足し算 1 回につき、200 回以上の書き込みと、100 回以上の読み出し操作が必要でした。この読み書き操作回数を減らすには、複数の bit を一括で計算する必要があります。その場合、下位の桁から発生するかもしれない繰り上がり（キャリー）を、予測しなければなりません。実際、本物の CPU にはキャリー予測器という回路が搭載されているようです。ちなみに、同じ設計でフラッシュメモリを用いずに作った CPU は、346.5k Hz で動作が確認できました。

おわりに

当記事は NAND 型フラッシュメモリの読み書きで動作する CPU を考案し、その原理を解説しました。また考案した CPU を製作し、その上で 3D ゲームが動作することを確認しました。ここまでお読みくださり誠にありがとうございます。また当記事の執筆にあたって、偉大な先人達の作品・知見をいくつもお借りしました。この場を借りて御礼申し上げます。

⁴ https://github.com/gpsnmeajp/jisc_ssd



にちか (@lxacas)

この同人誌活動のお陰もあって、素敵な出会いがありました。ありがとう、SSD 同人誌。次回は自作 NVMe-oF に挑戦したい。

JISC-SSD のケースをつくる

memukuge

クレイン電子さんから発売されている、フラッシュメモリ開発ボード「JISC-SSD (Jisaku In-Storage Computation SSD)」にピッタリなケースを作ったので紹介します。3D プリンタで印刷して利用するものですが、3D モデルの配布先や、3D プrintの案内もあるので、よろしければぜひ手に取ってみてください。

ケースについて

「JISC-SSD」は、M.2 SSD 風の 22 mm 幅基板、基板尾部のねじ止め用切り欠きがあるとはいえ、USB-C® 端子を搭載しているなどの差異もあり、M.2 SSD 向けのケースを使うことはできません。そこで、ケースに入れた状態で「USB Type-C® で接続できる」「2 つのスイッチを押下できる」ことを条件としたケースをモデリングし(図 1)、3D プリンタで印刷してみました。

3D モデルは以下の URL で公開されています。どなたでもライセンス (CC BY-SA) の範囲でご自由にご利用ください。

<https://www.thingiverse.com/thing:6732348>

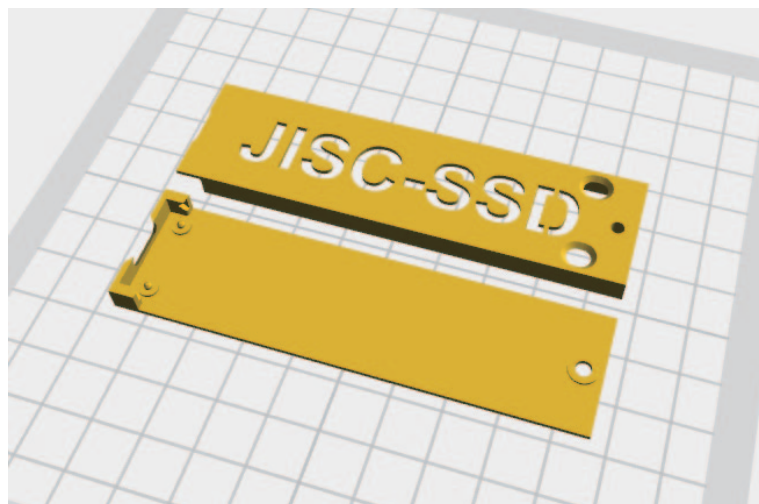


図 1 : JISC-SSD (Jisaku In-Storage Computation SSD) のケースの 3D データ

ケースは、上部、下部の2部品からなり、上下の部品の接続は、基板頭部側は嵌め込み、尾部側はねじ止めです(図2)。ねじはM3(φ3mmのJIS規格)、6mm~8mmの皿ねじが適合します。3Dプリント品へのねじ止めでは、受け側(ナット側)にインサートナットと呼ばれる部品を使うことが多いですが、このケースでは、ホームセンターで手軽に入手できるねじのみで固定します。ケース受け側の穴はφ2.7で開けてあり、φ3のネジをゆっくりねじ込み、ねじ山を切りながら固定します。写真の作例ではPLA樹脂で印刷していますが、ほかの樹脂でも同様に固定できると思います(もちろん、丁寧にタップでねじ山を切ればより強固に留められます)。



図2: 自作した、JISC-SSD (Jisaku In-Storage Computation SSD) のケース

3Dプリンタをもっていない時は

ファブスペース

まずはお近くのファブスペース(3Dプリンタ等の時間貸しを行う施設)を探してみるの
が一番です。このモデルは小さく背が低いので、日中にファブスペースに立ち寄り、その
日のうちに印刷を終えて持ち帰ることができるはずです。

3Dプリントサービス

3Dプリントサービスを使うことで、ケースを入手することも(一応)可能です。

ただし、上下部品セットで1点ずつ印刷すると、おそらく2,000円近くかかります。
一部海外の造形サービスで、送料が高いが単価が比較的安いサービスはあるので、欲し
い人みんなで共同購入、というケースなら、ありかもしれません。

1 2024年8月時点 執筆者調べ。

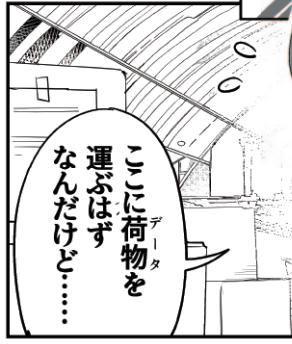


めむ (@memukuge)

いろいろ印刷してる人。それにしても、海外の3Dプリントサー
ビスの印刷そのもののコスパがすごいなあ、、、



ファイル保存中……



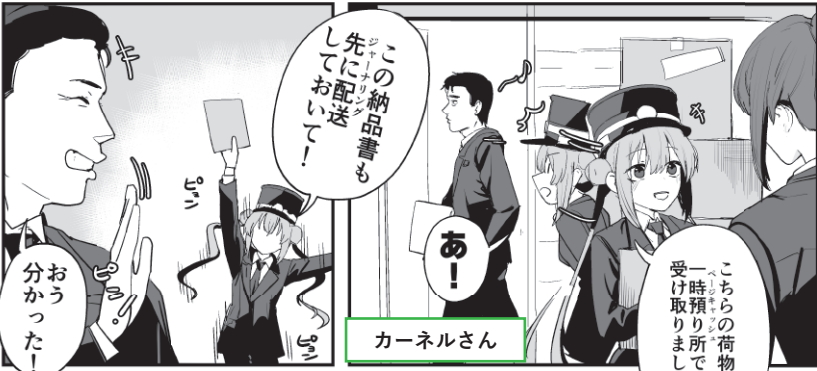
ファイルシステム内部

作 J.

すみません!

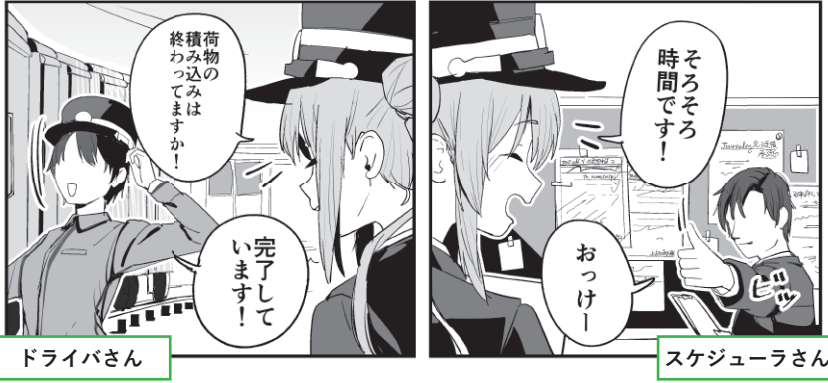
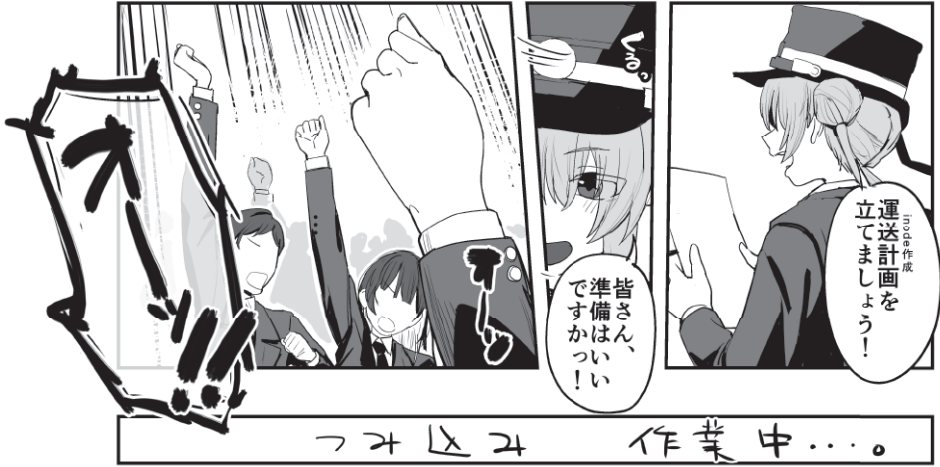


OS さん



職員へみなさんあつまってー

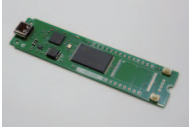






(ファイルの保存をしてからここまで大体0.01秒くらい)





<コラム> JISC-SSD

本同人誌に掲載される、自作 SSD (JISC-SSD) は、キオクシア製 NAND 型フラッシュメモリを搭載した、SSD の基本原理などを理解することを目的とした株式会社クレイン電子の学習向け拡張ボードです。その名も Jisaku In-Storage Computation SSD。

略して JISC-SSD です。



J

今回モチーフに選んだファイルシステムの機能は、ファイルシステムを構成するほんの一部でしかありません。調べれば調べるほど奥が深いので、ぜひ調べてみてください。

- 記憶容量の定義：1MB(1メガバイト) = 1,000,000(10の6乗)バイト、1GB = 1ギガバイトを1,000,000,000(10の9乗)バイト、1TBを1,000,000,000,000(10の12乗)バイトによる算出値です。但し、1GB=1,073,741,824(2の30乗)バイトによる算出値をドライブ容量として用いるコンピューターオペレーティングシステムでは、記載よりも少ない容量がドライブ容量として表示されます。ドライブ容量は、ファイルサイズ、フォーマット、セッティング、ソフトウェア、オペレーティングシステムおよびその他の要因で変わります。使用可能なストレージ容量(さまざまなメディアファイルの例を含む)は、ファイルサイズ、フォーマット、設定、ソフトウェア、Microsoft オペレーティングシステムやプリインストールされたソフトウェアアプリケーションなどのオペレーティングシステム、またはメディアコンテンツによって異なります。実際のフォーマット済み容量は異なる場合があります。
- 画像は説明用です。実際の商品・サービスとはデザイン・仕様が一部異なる場合があります。
- 「2.5 インチ」は SSD のフォームファクターを示し、SSD 自体の外形寸法を示すものではありません。
- NVMe、NVMe-oF は、NVMeExpress, Inc. の米国またはその他の国における登録商標 または商標です。
- PCIe、PCI Express は PCI-SIG の商標です。
- YouTube は、Google LLC の商標です。
- USB Type-C および USB-C™ は、USB Implementers Forum の商標です。
- microSD ロゴ、microSDXC ロゴは SD-3C LLC の商標です。
- Raspberry Pi、Raspberry Pi Pico は、Raspberry Pi Foundation の商標です。
- Linux は、米国およびその他の国における Linus Torvalds の商標です。
- Microsoft および Windows は、Microsoft group of companies の商標です。
- Dell は、Dell Inc. またはその関連会社の商標です。
- Debian は、Software in the Public Interest, Inc. の商標です。
- Intel、Core および Thunderbolt は、Intel Corporation またはその関連会社の商標です。
- Docker は、アメリカ合衆国およびその他の国における Docker, Inc. の商標または登録商標です。
- Java は、オラクルおよびその関連会社の商標または登録商標です。
- OCP、OPEN COMPUTE PROJECT は the Open Compute Project Foundation の商標です。
- その他記載されている社名・商品名・サービス名などは、それぞれ各社が商標として使用している場合があります。
- ©2024 KIOXIA Corporation. All right reserved.
- 製品の仕様、サービスの内容、お問い合わせ先などの情報は 2024 年 9 月時点の情報です。予告なしに変更される場合がありますので、あらかじめご了承ください。

本同人誌に掲載の情報や内容については十分に注意を払っておりますが、その利用によって利用者にかかる損害や被害が生じても、一切の責任を負いません。必ず利用者ご自身の責任においてご利用ください。

本同人誌記載の部品性能は部品単体での性能であり、自作 SSD の動作、性能等を保証するものではありません。



とだ勝之 (@katsudoren)

取材を快く応じてくださったさくらインターネットさん(特に江草さん)本当にありがとうございました! 現場見学も、原稿チェックで色々やり取りできたのも楽しかったです♪

…しかし今回はとだがメ切をきちんと把握してなくて雑誌連載した頃以来のハードスケジュールでした(^^)

てんこが主人公の漫画「ホームセンターてんこ」は下記の二次元バーコードから是非。



Kero

趣味でイラストを描く、まだ駆け出しの新人です。

これを読む皆さんと一緒に学んでいきたいと思います!

社会の役に立つ製品を生み出す技術者になるべく、日々の業務を頑張ります!

「データセンターてんこ」の作成にあたり、さくらインターネット株式会社様を訪問、取材させていただき、多大なるご協力をいただきましたことを改めて御礼申し上げます。

このSSD同人誌に対する皆さまのご意見・ご感想をお待ちしています。つぶやく際には、ハッシュタグ #SSD 同人誌 をつけてつぶやいてくださいね☆

■ SSD Doujinshi 3 - SSD の同人誌 3

2024 年 9 月 21 日 第 1 版第 1 刷発行

2025 年 1 月 16 日 第 2 版第 1 刷発行

著者 : AI ぱぱ / ragnag / とだ勝之 / ライゼ / 伊藤 普朗 / じむ /
松澤 太郎 / こたまご a.k.a. ひなたん / YmY・Tomo / Pochio /
余熱 / 宮内 / GPS_NMEA_JP / @isariyokurari / 高田 /
にちか / memukuge / J

本文イラスト（高田記事） : Kero

表紙イラスト : とだ勝之

表紙デザイン : 余熱 / ホーリー

編集 : Pochio / にちか / 余熱

取材協力 : さくらインターネット株式会社

発行 : キオクシア株式会社

連絡先 : kioxiahq-Exhibition-SSD@kioxia.com

印刷 : 株式会社 プリントパック

KIOXIA

キオクシア株式会社

〒108-0023 東京都港区芝浦 3-1-21 田町ステーションタワー S

www.kioxia.com