



FlashAir™ Dousoukai

FlashAirの同窓会

TAKE
FREE ¥0



無線 LAN搭載 SDメモ리카ード FlashAirの
同人誌が4年ぶりに今回限りの大復活！
今明かされる逸話と新ネタを見逃すな！

数年前に解散した元チームから今回の同人誌執筆の打診があった。さて、どうするか。

FlashAir チームは色々な立場、役割の人が同じ想いを共有している珍しくも素晴らしい集まりだった。SD メモリカードなのに人気若手女優を使った TV CM を作ってみたり、大手 TV 通販の社長に直接売り込みに行ったり、「挿すだけで簡単 IoT」と称してカメラ以外の機器でも使える事を PR するデベロッパーズサイトを開設したり。当時あまり聞きなれていなかったハッカソン&アイディアソンを開催して、FlashAir 芸人と名乗るしゃべくりのブロが大活躍（+ちょっと炎上）したり（笑）。

そんな FlashAir も、カメラ需要の減少とともにその役割を全うしようとしていた。

当時、そんな素敵なチームの解散を宣言したのは、この私。沢山の反発と、悲しみ、悔やみ、嘆きの声をもらったが心で泣きながら突き通した。その結果、チームメンバーは少しずつ私の元を去っていくことになった。私自身は残って保守品のアフターサポートを続けていた。手元には、メンバーに託された思い出の品が残っていたが、それを懐かしむ事もほとんど無くなっていた。

そんなある日、職場に配属された新卒新人が「FlashAir 知ってます」と言ってくれた。何かのキッカケの様な気がして、思い出の品を「少しずつ整理してみよう」と着手することにした。同人誌は手に取るとついつい読み返してしまうので、記念の一冊ずつキープして整理。その他、処分せずにそのままにしていたものが意外と有ったが断捨離した。それでも貴重なものは、かつてのメンバーに連絡して託した。

モノの整理が済むと、不思議なほど気持ちがスッキリして肩の荷が下りた様な感じがした。コロナ禍も有り、外出もままならない悶々とした雲に覆われた様な日々だったが、その雲が晴れた様な感じに。気持ちの区切りがつくと、新しいモノに取り組んでみたいという想いが湧いてきた。

FlashAir で走り回っていた日々、それを抱えて立ち止まっていた日々。いよいよ再び歩き出すタイミングなのだと気づいた。

そのような気持ちで今回筆を執った。かつて同じ想いを共有していたメンバーに会える嬉しさ。彼らが今どんな事を想って、何を生み出しているのか？この同人誌を手にとって読む日が楽しみでならない。



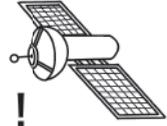
上岡 裕一

自称 FlashAir の義理の父。

リアルな方の子どもも巣立ちのタイミングとなり、感慨にふけるオヤジ。



フラッシュメモリが 使われている機器をさがそう！



！ 例えば、こんな機器で使われているよ！



ragnag (@ragnag1109)

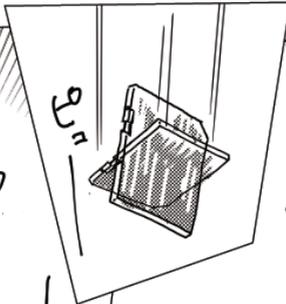
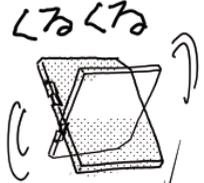
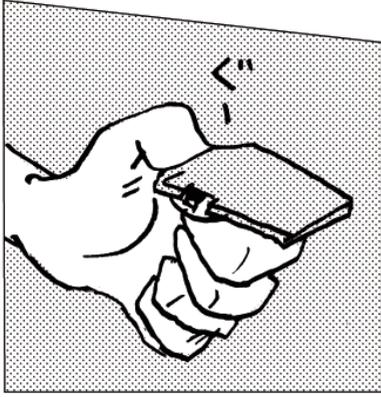
マンガを読むのと描くのをこよなく愛する広報担当。デジタル蔵書はまもなく500冊。今年はSSD同人誌はお休み。楽しみにしてくれたみなさん、ごめんなさい。今年は、FlashAir 同人誌にお邪魔します。世界はメモリであふれてる。いくつ見つけられたか教えてね。

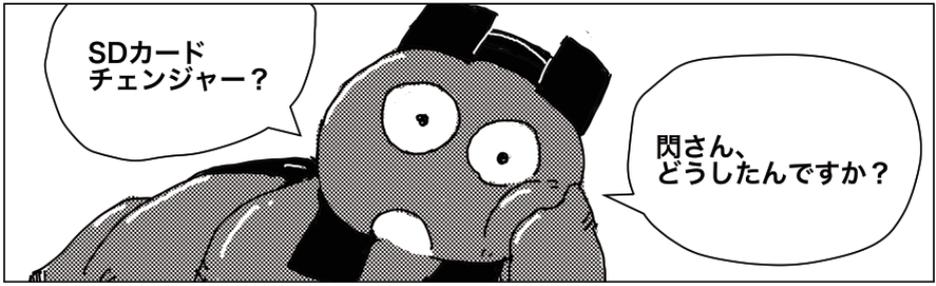


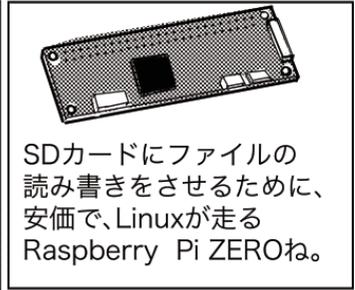
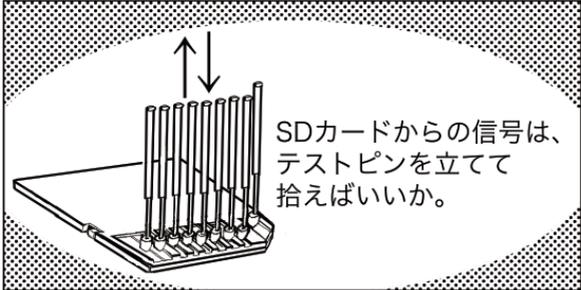
Contents

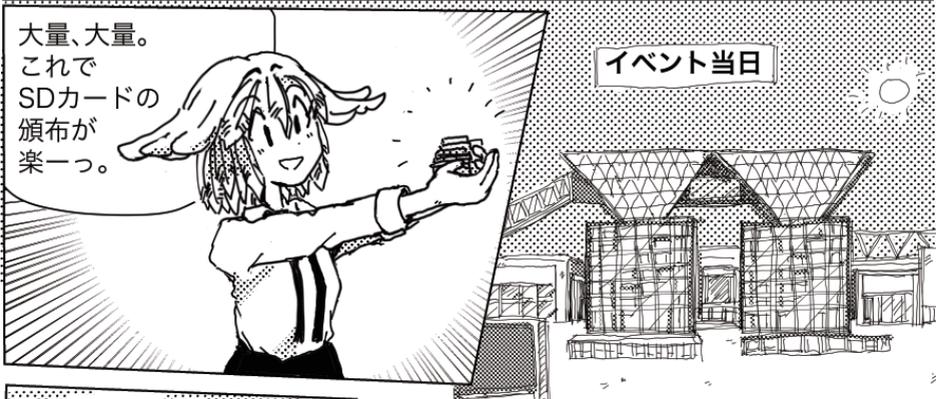
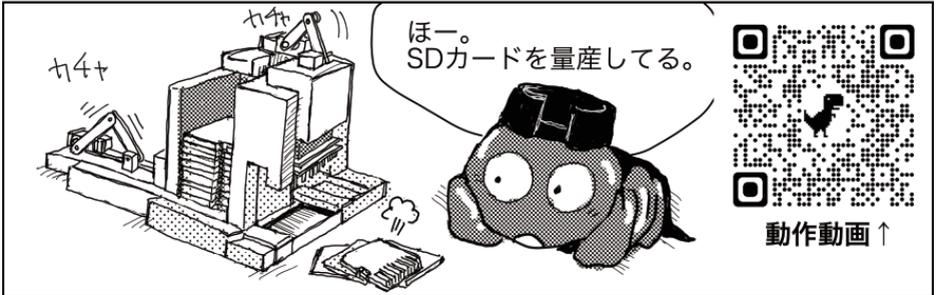
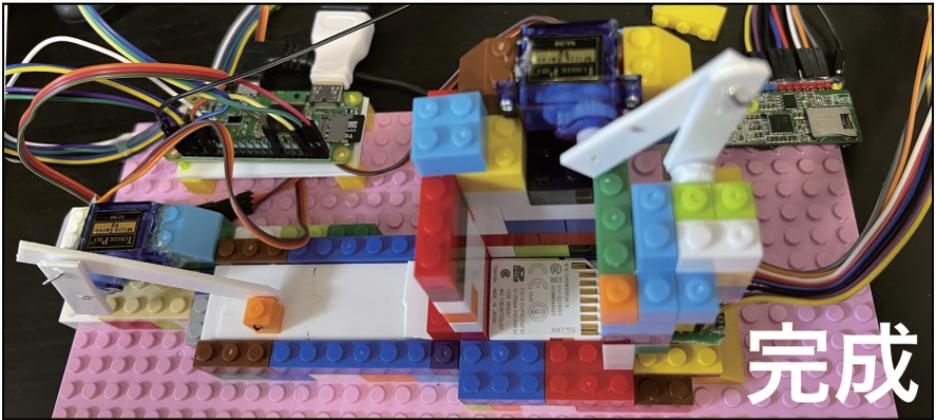
上岡 裕一	FlashAir「同想」会 … 3
ragnag	フラッシュメモリが使われている機器を探そう! … 4
じむ	SD カードチェンジャーを作るか。 … 6
伊藤 晋朗	やれる事、やりたい事 … 10
バブル世代の山女	山女日記～ FlashAir とフラッシュメモリ～ … 12
土居	いくつかの始まりと終わりの話 … 14
寺西	NFC 搭載 SD メモリカード「Mamolica」 … 18
余熱	FlashAir の思い出について語る … 20
田中 瞳	いつか見た夢 ある特許公開広報の解説 … 24
福屋 新吾	Airio-Base の半生 … 28
GPS_NMEA_JP	電波暗箱を借りて色々遊んでみた … 30
綾瀬ヒロ	FlashAir のたったひとつの冴えたやり方 … 34
せいみ まさみ	FlashAir お焚き上げ集 … 38
あおいさや	4 セグメント 1 桁時計の製作 … 42
にちか	FlashAir で作る自作 CPU サーバー … 44
にちか	世界初の NAND フラッシュは SSD の夢を見るか … 46
Pochio	室蘭紀行 2016 … 50
宮内	開幕! 非情の SD ファイト! … 58
とだ勝之	「フラッシュメモリのひみつ」への道 + 書評 (たまごまご) … 59

※本書に記載されている会社名、製品名、サービス名等は、各社の商標または登録商標です。
※本書に記載の内容は、各製品の製造元とは一切関係ありません。また、本書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な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。









やれる事、やりたい事

伊藤 晋朗

我が家には、15歳になる娘がいる。そんな彼女は受験生だ。しかし、彼女は塾に通うのは嫌いらしい。同じクラスの男子が勉強した事を自慢してきたりするが、代わりに、「あたしはお前より人生楽しんでいる。」と言い返す。確かに、彼女はテストの間でもゲームをしていたりして十分に楽しんでいる。おそらく、本人はそこそこできていれば良いようだ。

そんな娘が生まれたところに遡るが、FlashAirのプロジェクトが始まった。きっかけは、若手のprestで、2つのSDカード間で写真を交換するというアイデアを上の人が面白かった話から始まったらしい。なぜそんな話になったのか聞いてみると、彼のやりたい事は「デジカメで写真撮って後で送るよ。と言うけど、その場で交換したい!」という話からアイデアが生まれたそうだ。当時はみんなが持っているのはスマホではなくケータイの時代であり、写真を撮るのはデジカメというのが一般的だったからだと思う。

プロジェクトはその後に発展してSDカードから写真などをインターネットのサーバーに転送するというアップロードタイプのアプリケーションを実現していこうという話もしていくことになった。ただし、アップロード先のサーバーとかを我々が用意するのは難しいので、そこはサービスしたい会社に任せたいという前提である。そのため、我々はインターネットへ転送する機能だけSDカードでやりましょう、そして例えばカメラメーカーが、カメラからFlashAirを制御して、受け手のサーバーも用意してサービス提供するという仕様がいいのではないかという話ができた。さらに、その仕組み自体をSDカードの規格として作りましょうね、という話になった。この仕様に基づいて、気が付いたらものすごい勢いでハードウェアは作られていった。途中で東日本大震災などいろいろあったが、開発は進

み原理試作の基板と、SDカードタイプのサンプルなどが出来上がった。私はサンプルカードを目の前にして、「さて、どうするか」と思った。本来は仕様に基づいて、まずソフトウェアを実装すべきだったところ、そこは一旦置いて、ただ、勝手にやりたい事をやり始めた。それはスマホのブラウザからカードの中にある画像を見る仕組みだった。まあ、そこまでやったら面白いだろうなと思ってはいたが、あまり人には説明してこなかったのはなんとなく説明が面倒だったのかもしれない。最初の仕組みはHTTPでアクセスされたら簡単なHTMLをprint文で書き出す仕様だったと思う。今思うと90年代に掲示板を作った時にも同じ事をしていた気がする。しかし、最初に作ったのは、通信速度が遅かった。



たしか、100kbps ぐらいしか出ていなかった。そのため、「うわ、遅くてやべーな。」と思って、なぜかぞっとした記憶がある。その後にソフト的に通信速度を速くしたり、ブラウザ描画は JavaScript を使って通信負荷を減らしたりとか実装を変えながら、人に見せて遊んでいた。

そして、あと数か月で製品発売するというタイミングで問題が起きた。これまでアップロードタイプの製品を想定していたが、想定していた規格に対応する機器の話が遅れていた。そのため、このままでは発売しても使えないという話になった。「じゃあどうする？」という話になったときに、製品仕様を遊びで作っていたダウンロード型にしてしまおう、という事になった。その結果プロジェクトリーダーから「じゃあ、仕様を2日で作って。」と言われて、FlashAir の製品仕様が追加される事になる。

アイデアの時から、最初の製品が出る時も色々変わったが、その後も製品仕様としては追加機能という形で変わっていった。システムとしてはスマホアプリが追加されたり、IoT 向けとしてスクリプト言語が動くようにしたりと、色々で製品としてやれる事が追加されていった。色々な人が集まって、やりたい事を実現するために、あんな小さな SD カードの中に色々な機能を詰め込んでいったのは楽しい思い出だ。

最近、受験生の娘も、自分のやりたい事を考えていたようだ。その時の会話は、

娘：「あたし、漫画家になりたいって言っていたじゃない。」

父：「そうだね。」

娘：「でも、最近ゲームクリエイターにもなりたいたよね。」

父：「ふーん」

娘：「社長にもなりたいたいの。あと、エンジニアにもなりたいたいの。」

父：「・・・」

娘：「だから、人生が足りないの!」

父：「(爆笑)」

最初は、若いなあと思っていたけど、言われた後に、自分の人生に対して、漠然とした一本の道しか考えていない自分もいた。それから、おっさんでもやりたい事をたくさん抱えて同じように言ってもいいのかもしれないと考え方が変化していった。今の目標は、定年になった時に子供のように「人生が足りない」と叫べるようになってみたい。



伊藤 晋朗 (@ikainuk)

FlashAir の中の人だった。開発中の10年間ぐらいは、夢を見た記憶がない。常に FlashAir の事を考えていたようだ。

山女日記～ FlashAir とフラッシュメモリ～

バブル世代の山女

2023年7月某日

換気扇の掃除の見積もりをH姉さん(いとこ)が紹介してくれた掃除サービスに頼んだ。Y地区の店長と副店長の2人が来た。2人は社長の娘達(おばちゃん)。店長は着くなり「このマンションはいつ頃の建築ですか、だいぶ古いですね」と言う。狭さと古さに明らかに失望の様子。「いとこさんがお医者さんなのでお医者さんなのかと思いました」と言う。「すみませんね、しがないサラリーマンで」と返す。勝手に勘違いして失礼だなとムツとしたその時、副店長がハイトーンの声で「有名な登山家さんかと思いましたあ〜」。



図1: 額縁の中の山女

「は?」「その写真、講演会で話されたんだと思いましたあ〜」。目線の先には2019年CP+「私のカメラライフとFlashAir『バブル世代の山女』」の対談時に山登り姿で登壇した写真があった(図1)。カメラが趣味の同僚Mさんが撮影してフレームに入れてプレゼントしてくれたものだ。「あーこれは製品の販促イベントの時の写真なんですよ。山は趣味で登っているだけなんですけどね・・・」一気にたたみかけるように話したら、しばらく忘れていた記憶がよみがえった。

山女のFlashAir対談イベントは、CP+のプログラムが1枠埋まらず打診を受けたものだ。山には30代のころから20年近く登ってきたが、仕事のネタになるとは思わなかった。若手の女性同僚が話すタイトルが「インスタ世代のカメラ術」と決まっていたので「バブル世代の山女」とつけた(図2左)。対談形式で面白く聞いてもらえるようにしようとしナリオを作ったのだが、ほぼ自慢話になった。司会のK氏がうまい具合に合いの手をいれてくれて、山で撮った一押しの写真を紹介できたので、自分が楽しんでしまった。対談の締めくくりは「なぜ山に登るんですか?」「おいしいビールを飲むため!」(図2右)だった。

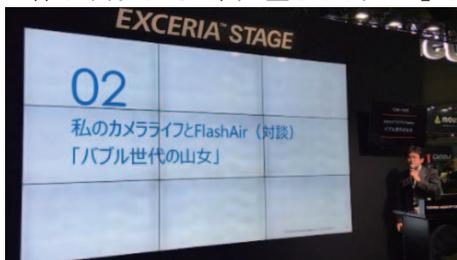


図2: 2019年CP+セミナーの様子

FlashAir の製品開発や事業展開は山登りと似たところがあったと思う。製品リリースにあわせて開発し、販促計画をたて、実行していく。社内外の仕事仲間、お客様と尾根道、沢道ともに歩んだ。FlashAir が見せてくれた景色は格別で、思い出深いものばかりだ。そしておいしいビールもたくさん飲んだ。

FlashAir という山を下りた現在、私はフラッシュメモリや SSD といった巨大な連邦を登っている人たちの見ている景色を記事として寄稿してもらい、Web サイトを通じて社内外に発信する仕事をしている。

10 月下旬（予定）にフラッシュメモリの動作の基本原理の説明動画を公開するので、ぜひ多くの人に見ていただきたい。フラッシュメモリが写真や動画、音楽と言うコンテンツを 0, 1 のデータとして記憶し、保存し、読みだす世界はミクロな魅力と不思議にあふれている。

「光る太陽と記憶するフラッシュメモリ～3次元フラッシュメモリの動作原理～」

キオクシアの研究技術開発



<https://www.kioxia.com/ja-jp/rd.html>

2023 年 8 月某日

副店長+1 人が来て、換気扇とダイニングを掃除してくれた。床も磨かれて気持ちよい。出費は痛いがああの油污れは手に負えなくなっていたから致し方なし。副店長はどこから掻き出してきたの? という埃の山を「こんなにたくさん出てきました～」と自慢げに見せてくれた。うーん、その山は見せてくれなくてよかったかも (笑)。



バブル世代の山女

FlashAir の企画に携わり、事業化では応用技術として展示会や顧客開拓に従事。2019 年より技術情報発信担当等に従事。

こんにちは！FlashAir Developers サイト（以下、Developers）の運営を担当していた土居と申します。株式会社フィックスターズという、マルチコアとかGPUとか量子コンピュータなど、難しいコンピュータをうまく使いこなすということをウリにしている会社で働いています。FlashAirのコミュニティの中では、FlashAirそのものを開発された東芝の皆様を除くと、実はもっとも古参の方だったりします。

今回、同窓会バージョンへの寄稿ということで、過去の同人誌を眺めておりましたら、最終6号でも私は思い出話を語っておりましたね。しかも面白い。せっかくなので、DevelopersやFlashAirコミュニティの思い出はダイジェストするにとどめ、今回はアプリの話を書いてみたいと思います。

FlashAir との出会い、そして Developers のはじまり

私がFlashAirに関わりはじめたのは、ちょうどこれからフィックスターズで働こうという2013年の4月いや3月でした。FlashAir事業チームの櫻井さんや高田さんにお会いして、FlashAirのマイコン的潜在性を開花させるためコミュニティを作るといふ会話をしたと思います。その後、コミュニティ構築経験を持つフィックスターズの蓮見女史とともにDevelopersサイトの原型を提案致しました(図1)。その後、FlashAir Developersの構想を具体的なコンテンツ案やデザイン案に落とし込み、開発を始めました。Developersのコンテンツの内、リファレンスやチュートリアルのような技術記事のかなりの割合は私が執筆あるいはドラフトしていたと思います。2013年の夏にサイトが公開された後は、様々な方に書いていただくようになり、コミュニティの広がりを肌で感じました。

Developers向けにはコンテンツの開発と並行して、FlashAirの活用事例の作業も始めました。当時はGPIOもLuaもなく、iSDIOはありましたが活用の研究が進んでいませんでした。ということで、お手軽なWebサーバー機能を使い、照度センサーのデータを記録して、ウェブブラウザでリアルタイムモニタリングするというデモを作りました(図2)。外箱も作ったので持ち運びやすく、手でセンサーを覆うと反応が得られてわかりやすいた



図 1: 2013 年 4 月の初期提案
すでに原型ができあがっている

め、結構重宝されたように記憶しています。鉄道模型を自在に動かしたり、音楽を奏でたりといった、コミュニティの皆さんの高度な作品からすると非常に簡素なものですが、個人的には思い出深い作品です。



図 2: リアルタイム照度モニタ

FlashAir アプリ開発への参画

さて、今回の主題はここからです。まずは大事なことですが、この本を手にとっている皆さんは FlashAir とはちょっと大きめのフラッシュメモリが載った超小型マイコンだと思っ
ていらっしやると思いますが、それは世の中の一般的な認識ではありません。FlashAir
ユーザーの 99.7% くらい（私調べ）の方は、デジタルカメラに差し込んで使う無線 LAN
付き SD カードだという認識でしょう。

つまり、FlashAir に搭載された無線 LAN と Web サーバーにより、撮影後にデジタル
カメラから SD カードを取り出すことなく、手元のスマホ、タブレットや PC から写真を確
認したりダウンロードしたりできる、というちょっとの手間だけど面倒くさいことを解消し
てくれる素晴らしい便利グッズなのです。

そして、スマホでのユーザーには専用のアプリが提供されていました¹。実は、フィッ
クスターズは Developers の運営と同時に、このアプリ開発にも参画することになったの
です。2013 年 8 月に私たちが関与した初め
のバージョンである 2.0 がリリースされま
した（図 3）。すでにリリースされていたバー
ジョン 1.0 に、ダウンロードやシェアの機能
を追加し、iPhone/Android で機能を揃えた
リリースであったと記憶しています。今となっ
ては見た目が古いですね。見返してみると
iPhone 5 だったようです。今は 13 でしたっ
け？

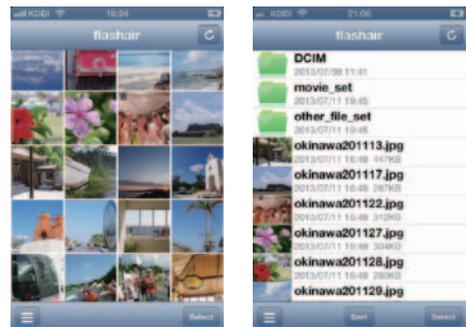


図 3: FlashAir アプリ 2.0

1 PC 向けのアプリというのもありました。

そして、なんといっても、一番思い出深いのはコンセプトの再整理から UI 刷新まで私たちが提案したバージョン 3.0 です。技術的に一番のチャレンジだったのはサクサクした動きを実現するために、一覧のサムネイル取得や拡大表示用のデータ取得を非同期化したことです。非同期化ということは、いつでも一覧のスクロールや拡大表示をやめることができるということであり、バックグラウンドで実行されているダウンロードをキャンセルできなければなりません。これを精密に動作させるというのが大変でした。また、コンセプト的には、FlashAir につながってなくても、写真アプリとして使えるものになりました。

V3 リリース、そして地獄の始まり

そうして 1 年弱の開発期間を経て、2015 年 2 月に iPhone 版のバージョン 3.0 が、少し遅れて Android 版もリリースされました (図 4)。大いにワクワクしましたが、同時に地獄も始まりました。手元のマシンではコンセプトを実現してよく動いていたのですが、リリース後にバグ報告が次々と舞い込んできたのです。スマホアプリ開発の難しさを知りました。iPhone/iPad も当時は進化が激しくユーザーの持っている機種がバラバラです。Android はそれに加えて、画面サイズも性能もストレージ構成も異なる機種が無数にあります。いずれも、手元のテストでは OK でもすべてで OK とはいかないのです。特に、FlashAir アプリはスマホの無線 LAN 接続を操作するのですが、Android は機種ごとに無線 LAN コントローラも違うので、命令の挙動やタイミングが違うのですね。Android は 80 機種くらいテストしていたと思いますが、それでも全然足りませんでした。そこから 1 年と少々、5 人ほどのチームで粘りましたが、テストのコスト増に耐えきれずとうとうグブアップ。結局無線 LAN のコントロールも完全にはいかなかったと思います。東芝メモリ様にて別チームに引き継いでいただく形となり、私たちフィックスターズチームの挑戦は終了しました。

このようにいろいろと苦労もあり、ご迷惑もおかけ致しましたが、世界中で 100 万人以上に使っていただくアプリ開発に携われたのは、今でも勇気をいただける思い出です。

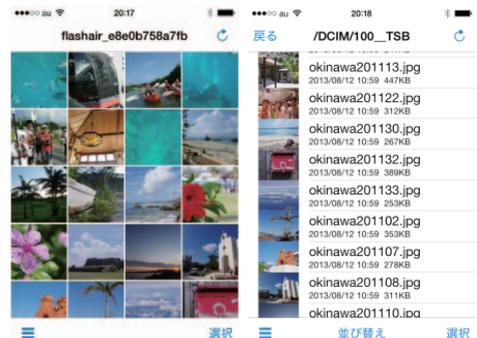


図 4: FlashAir アプリ 3.0

リーフレットを忘れちゃいけない！？

さて、前の章で過去の記録を漁っていたらだんだんと苦い記憶が呼び起されて辛くなってきたので、また Developers の話をしましょう。

本書を手にとっている皆さんの中には、Doujinshi を 6 冊コンプリートしているよ！という、FlashAir Developers フリークの方も数多くいらっしゃるかと思います。

しかし、実は Developers 関連の紙配布物には「リーフレット」もあるのをご存知でしょうか。リーフレットは前述の Developers の母こと蓮見女史のデザインによるもので、ブルーの表紙に細長いスタイルで始まりました(図 5 左)。その後、内容が増えていって A5 サイズに拡大(図 5 中央)。さらに、iSDIO の解説がサイトに掲載されたあたりでブルーとグリーンとの 2 冊体制になっていきました(図 5 右)。バージョンを見返してみると、ブルーは v6、グリーンは v10 までであるようです(v11 かも)。さらに日本語、英語、中国語版があります。こちらは一定の場所で配布したわけではなく、いろいろなイベントで配布しましたので、もしコンプリートしている方がいらっしゃると相当レアだと思います。



図 5: リーフレット 初代と v7



土居 (@munepi)

株式会社フィックスターズ在籍の FlashAir Developers の中の人だった人。FlashAir に携わってから 10 年、FlashAir はなくなりましたが(いや、中古市場で大人気のようですね)、コミュニティはこのように存続しています。こうしてまた Doujinshi もとい Dousoukai に呼んでいただけることがとても幸せです。

NFC 搭載 SD メモリカード「Mamolica」

寺西

無線三兄弟（「FlashAir」、「NFC 搭載 SD メモリカード」、「TransferJet 搭載 SD メモリカード」）と呼ばれたSDメモリカードのうち、あまり知られていないNFC搭載SDメモリカードについて書いていきます。

NFC 搭載 SD メモリカード

NFC 搭載 SD メモリカードは、交通系 IC カードと同じ無線機能を実装しており、電源が不要でパッシブタグとして動作します。そのため、技適（電波法）の規制を受けることなく使用できます。

最大の特徴として、他の無線カードとは異なり、無線転送速度（表 1）が著しく遅いというのがポイントです（泣）。

表 1: 無線系 SD メモリカードの無線転送速度

商品	無線規格	規格上の転送速度
FlashAir W-04	IEEE802.11b/g/n(SISO)	150Mbps
Transfer Jet 搭載 SD メモリカード	TransferJet	560Mbps
NFC 搭載 SD メモリカード	ISO/IEC 18092	212 / 424kbps

FlashAir のように写真データをスマホに直接転送できず、カード内の写真データからサムネイルを作成し、それを NFC タグ内のメモリに格納することで、カードの中に保存されている写真のサムネイルやカード残容量をスマホアプリ経由で確認できる B2C 商品として、2015 年に『Memory Card Preview』が発売されました（図 1）。その後、AR 機能をスマホアプリに実装し、NFC 経由ではなく、カードの表面にある二次元バーコードをスマホのカメラでかざすだけでサムネイルが確認できるようになりました。

当時では先進的な機能を実装したのですが、FlashAir という偉大な兄がいたため、カードの中に保存されている画像データを NFC 経由でスマホにダウンロードできるという誤解が生じてしまい、ヒット商品にはなりませんでした。



図 1: Memory Card Preview

Mamolica

海外出張の飛行機の中で、同僚と話しながら考案したのが、NFC 搭載 SD メモリカードの B2B 製品「Mamolica」です (図 2)。

当時、カメラやビデオカメラで撮影したデータが入ったカードを紛失したことで、情報漏洩問題が発生し、新聞などで良く取り上げられていました。Mamolica は NFC を使って SD I/F をロック (無効化) することで、壊れた SD メモリカードのようにふるまい、紛失時に情報漏洩を防げるというコンセプトの製品でした。スマホとカード内の NFC タグと通信しスマホアプリから手動でロック操作をした場合、もしくはカメラの電源が切れた瞬間にロックされます。アンロックする場合、スマホが鍵となり、スマホアプリから手動で操作します。たとえスマホと Mamolica を同時に落としたとしても、パスワードやアンロックできる時間設定、アンロックできる場所設定などを組み合わせることでセキュリティを強化することができる機能を盛り込みました。

Mamolica は、「守る」+「カード」から組み合わせた造語です (開発初期は、「Lock U」と呼んでいました)。様々な名称の案があったのですが、商標調査をすると英語を組み合わせた名称は取得が難しく、日本語を組み合わせた名称に落ち着きました。

その後、テレビ局のエンジニアの方々と共同で、撮影現場の中で試してみて、仕様を決めていきました。その中で、カメラの機種によっては動作時間を延ばすために定期的に電源を切っているものがあり、突然、SD I/F がロックされるという問題が見つかり、単純に SD I/F をロックするのではなく、Read だけを禁止するという機能を盛り込みました (Write だけは可能)。

最終的に、映像情報メディア学会の進歩開発賞 (研究開発部門)、映画テレビ技術協会の技術開発奨励賞および放送文化基金賞を受賞することができました。

しかし、Mamolica も現在 EOL になっています。



図 2: NFC 搭載 SD メモリカード「Mamolica」



寺西 (@soft128)

FlashAir、NFC 搭載 SD メモリカード、UHS-I / UHS-II SD メモリカードの商品企画の後、PCIe I/F 搭載リムーバブルストレージの商品企画担当。その前は、マルチコア CPU 向け RTOS 開発、エナジーハーベスト製品開発など、社会人になってから、半分以上は新規商品開発担当。

FlashAir の思い出について語る

余熱

4年ぶりにFlashAirの同人誌を書くことになったので、過去の思い出を振り返ったり、お蔵入りネタを蔵出ししていきます。

さしずめタイムカプセルを開けたようなイメージで楽しんで貰えると幸いです。

空と月

2015年6月6日のお酒のイベント「酒っと」で同人誌を売りたいとPochioさんから声がかかり、サークル「空と月」を結成しました。これがもの凄いイベントで、6/6の深夜24:30からスタートし、翌7日の4:30に終了するという同人イベントとしては異例の時間帯に開催するということに加え、出展サークルから来場者にお酒を振る舞うことが可能だったのです。そこで、我々のサークルも同人誌を購入して頂いた方にサイコロを振ってもらい、出た目に応じてお酒を振る舞うという企画を行いました(図1)。同人誌内で紹介していることもあり、Pochioさんが森伊蔵を持ってきてくれて、少し飲ませてもらったりしました。

出展サークルも来場者もお酒を飲みまくり、イベント終了後には開催場所であるCafe & Bar SIXTEENの前は死屍累々たるありさまで、酔いつぶれた人々が道路に横たわっていました。私も例に漏れず泥酔しており、臃げな記憶ではサークルメンバーに抱えてもらい階段を下りたような…。そのまま秋葉原のカプセルホテルに泊まったのでした。次の日、「酒っと」で売れ残った同人誌の新刊を委託予定だった秋葉原のショップ「三月兎」に行くと、なんとその新刊が既に販売されているではないか!これが一部で伝説の泥酔納品(図2)と呼ばれ、いまでもイジってもらってます笑。



図1: 酒っと出展の様子



図2: 伝説の泥酔納品

TISPY と TISPY2

空と月で作ったアルコールチェッカー「酔ったー」をもとに、アルコールガジェット TISPY を作り、2016 年 3 月にクラウドファンディングを行って製品化しました。TISPY の内部 (図 3) のマイコンには NXP の LPC11U35 を採用しており、USB 経由でファームウェアをアップデートする機能を持っています。クラウドファンディングのような製品では市場でファームウェア更新の必要があると考え、このマイコンを選定しました。また、Mbed™ オンラインコンパイラで開発できたのも良かったように思います。

2019 年 8 月に TISPY2 がクラウドファンディングされました。TISPY2 では、ゲストモードや多言語対応に加え、センサ付近に活性炭シートが巻かれており、センサの劣化が起こりづらいという特徴があります。TISPY2 を分解すると、内部 (図 4)

には STM32L073CZT6 が使われていることが分かります。FW に機能を追加するために ROM/RAM 容量を大きくする必要があったためにマイコンの型番を変更したわけですが、USB 経由での FW アップデートができなくなっています。そのため、FlashAir (SD カード) を使ってアップデートするよう仕様変更されています。余談ですが、マクニカさんと一緒に作った TOY Board というマイコンボードでも同じ STM32L073 を使っています。

TISPY は Makuake Award 2016 にて Silver 賞を頂き、立派な盾を貰いました (図 5)。現在はスタッフ社さんにて飾っていただいています。

TISPY を作っていた時は FlashAir の IoT 用途の開拓を本気で信じており、特にクラウドファンディングのような少量多品種の製品には FlashAir はピッタリだと強く思っていました。製作にあたり色々若気の至りもあったのですが笑、FlashAir の機能を使うことを前提としたガジェットを自分で作ることができたのは、とても良い経験になりました。



図 3: TISPY 基板



図 4: TISPY2 基板



図 5: Makuake Award の盾

展示会で見かけた特殊な SD カード

microSD 版 Mamolica

寺西さんの記事で紹介されている Mamolica の microSD 版の試作品です(図 6)。NFC を使って SD I/F をロックするという Mamolica の機能はそのままに、サイズを microSD に収めた製品で、ただでさえ小さな microSD 内に、NFC チップやアンテナを詰め込むというもの凄い実装をした製品です。CP+2018 などでも展示されていました。

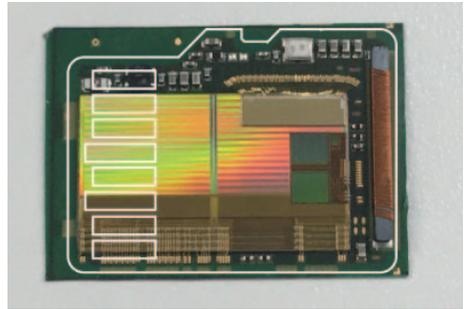


図 6: microSD 版 Mamolica 基板
(microSD 外形は余熱による追記)

NFC タッチ & コピー機能付き SD メモリカード

Mamolica 同様 NFC 搭載の SD メモリカードなのですが、NFC 経由で SD カードの中身にアクセスできるという特徴を持っており、カードリーダー無しでスマホとデータのやり取りをできるという便利な製品です。ただし、SD カードに電源を入れつつ NFC でタッチする必要があり、かつ転送速度も遅いため、なかなか活用場所が難しい製品でもありました。TISPY であればたまたますべての要求を満たすことができたため、2018 年の IoT/M2M 展 [春] の展示会向けにデモ機が作成され(図 7)、タッチ操作でデータのやり取りを行う様子が展示されました。



図 7: NFC 対応 TISPY

Bluetooth 版 FlashAir

同じく 2018 年の IoT/M2M 展 [春] に出展したネタなのですが、Bluetooth® low energy と無線 LAN の両技術を搭載した SD メモリカードの参考展示を行いました(図 8)。FlashAir は、かねてよりユーザーの皆さんから Bluetooth、とりわけ BLE を搭載して欲しいという声を頂いていたこともあり、内部でこのような製品も検討されていたようです。

今回は、田中さんに特許の解説記事を書いてもらいました。



図 8: BLE 搭載 FlashAir

FlashAir の殻割り

FlashAir の中身を開けると、基板の裏にいくつかテストパッドがあることが確認できます。ハードウェア解析の定番ネタなのですが、この中に UART の信号が含まれているのではないかとということでクレイン電子さんが Airio-Base の開発の際に解析をしていたのです。今回、その情報をもとに UART の信号を引き出した FlashAir を作成し、GPS_NMEA_JP (@Seg_faul) さんをお願いして解析してもらいました。

(おまけ) SD WIFI PRO

FlashAir のように無線 LAN を搭載した SD カードは市場に数種類あるのですが、電子工作ができるような製品が無いか探していたところ、FYSETC という 3D プリンタメーカーが SD WIFI PRO という製品を出していました (図 9)。UHS-II ソケットを使うことで SD I/F に加え 9 本の GPIO を使うことができる製品です。



図 9: SD WIFI PRO

こちらも GPS_NMEA_JP (@Seg_faul) さんに解析してもらいました。なお、電波暗箱はクレイン電子さんにお借りしました。

おわりに

今回の同人誌をはじめ、今年は過去の活動を振り返る機会が多く、色々考える機会を貰いました。それをこのように同人誌にまとめて、色々な人に読んでいただけるのは、本当にありがたいことです。FlashAir の活動は今回で本当に一区切りになると思いますが、SSD 同人誌などの活動は続けていきたく、引き続きよろしくお願ひします。



余熱 (@yone2_net)

去年、一児の父になりました。子育てしつつも執筆の時間をくれた妻に感謝。4年越しでこんな面白い本を書けるなんて、遊べるなあ FlashAir。息の長い活動になりましたよね？

いつか見た夢 ある特許公開広報の解説

田中 瞳

はじめに

FlashAir が採用する通信方式である無線 LAN は、接続操作を子機 (STA) から行うため、スマートフォン等で FlashAir に接続するには、FlashAir 側が親機 (AP)、スマートフォン側が子機 (STA) となります。

一般的に、無線 LAN の AP は、ネットワークルーターとして上位のネットワーク (多くはインターネット) に接続していて、STA にそのネットワークを提供することが期待されています。またスマホでは、セルラーの通信量を抑制する観点から、セルラー通信よりも無線 LAN 通信を優先する実装が多く、上位のネットワークに接続していない FlashAir にスマホから接続したとき、スマホのインターネット接続が失われてしまう課題があります。もしセルラーと無線 LAN の両方に接続しているときに、セルラーの通信を優先するような設定が可能なスマホであれば問題は起きませんが、製品のユーザー体験が、スマホのネットワーク部の特殊実装頼みになってしまいます。

これに対し FlashAir では、別にインターネットに接続している無線 LAN ネットワークがある環境で、FlashAir をそのネットワークの中継器として動作させ、接続したスマホにインターネット接続を提供する「インターネット同時接続モード」が搭載されています。

しかし、屋外など他の無線 LAN ネットワークがない状況では、FlashAir 接続時に一時的にスマホのインターネット接続が切れるため、ユーザーの体験を損ねる恐れがあります。

カメラが採った解決法「アドホックモード」

他方、デジタルカメラでは、近年すっかり無線通信機能の搭載が当たり前となりました。これらのカメラは、通信速度の問題から写真・動画転送に無線 LAN を採用したうえで、スマホのインターネット接続に影響を及ぼさない「アドホックモード」を使っていることが多いようです。アドホックモードは、FlashAir や無線 LAN ルーターで用いられるインフラストラクチャモードと異なり、ネットワークの上下接続を前提としない、一対一の接続です。標準的なスマホの実装では、アドホック接続があってもセルラーのインターネット接続は維持されます。

ただし、アドホックモードでは、接続する2機器がそれぞれ接続操作をする必要があります。カメラの画面、操作キーから、スマホを探して接続させる動作をしてもよいですが、操作が複雑になるため、カメラに無線 LAN に加えて Bluetooth® LE (Peripheral = 子機)

を搭載し、スマホ側と常時ペアリングし、スマホ側のカメラアプリの操作で、カメラからのアドホック接続操作を制御することが行われています。ちなみにこの辺の操作の隠ぺいは各カメラメーカでよく工夫されており、スマホの写真転送アプリを開いたとたん、スマホと、ペアリング済みのカメラの双方がアドホック接続動作を開始し、起動画面中に接続を完了していたりします。

無線 LAN におけるインフラストラクチャモードとアドホックモードに、ハードウェア的な違いはなく、FlashAir でもアドホックモードの通信は可能ですが、画面も操作キーも、Bluetooth も持たない FlashAir は、アドホックモードでの接続に際し、複雑な操作を伴いユーザ体験を損ねる恐れがあります。

ある特許公開広報（ここからが本題）

とある会社から出願された特許の公開広報「特開 2019-144734」には、SD カードのような記憶装置が、単一のアンテナを2種類の無線通信で利用するコンセプトが示されています(図1)。

SD カードに限らず、実装面積が限られる機器に無線 LAN と Bluetooth の両方を搭載する場合、2.4GHz 帯の無線 LAN を搭載し、周波数帯が同一の Bluetooth とアンテナを共用することはよく行われています。さてこのような時、それぞれの通信方式は、どうやって接続を維持するのでしょうか。

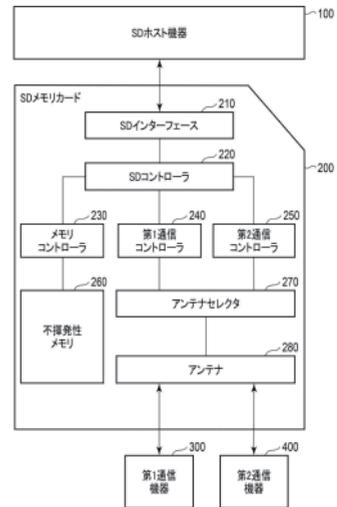


図 1: 特開 2019-144734 よりシステムブロック図

カギは衝突回避

無線 LAN と Bluetooth では、通信の衝突に対する実装コンセプトが異なります。無線 LAN の衝突制御は CSMA/CD といい、平たく言うと「送信を行いたい機器は、送信したいチャンネル（伝送路＝周波数帯のこと）で他の通信がなくなるまで待ってから送ろう。相手から届いたと返事がなかったら、相手端末近くでの衝突が考えられるので、ちょっと待ってから送りなおそう」です。まるで人間の会話のようですが、各デバイスが空気を読んで、他人（デバイス）がしゃべっているときに割り込まないようにしています。

Bluetoothでは、一定の時間間隔「スロットタイム」ごとに、端末は、送信するチャンネルで他の通信があるかを確認することなく、相手に送信を行うことができます。これではBluetoothで衝突が起きまくるようにも感じますが、Bluetoothは（小難しいですが）送信信号を周波数軸上で広く拡散させ、その中心周波数を頻繁に変化させており、また送信出力も小さいです。つまり、品質の悪い（部分的に衝突が起きている可能性のある）チャンネルでも通信できるようにし、さらに衝突の可能性そのものを低減し、通信を成立させています。

BLEにおける電源制御と無線LANとの共存

冒頭で触れたデジタルカメラのように、Bluetoothを別の機器の制御に用い、Bluetooth自体は高レートの転送（音声とか、映像とか）を行わない場合、Bluetoothの中でも消費電力を大幅に低減した規格であるBluetooth LE (BLE)がよく採用されます。BLEでは、特に送信電力の低減と、送信間隔を大幅に引き延ばし（数十msあるいはもっと長く）、互いに送受信するタイミング（多くの場合1ms以内）のみ無線機能が動作し、それ以外の時間では無線機能の電源を停止することで、大幅な消費電力の削減を可能としています。これには前述の衝突回避のコンセプトも大きくかかわっており、BLEがもし無線LANのような、送信前にチャンネルが空いているかを確認し、送信を待つような仕様であった場合、送受信時に無線機能の電源を入れておかなければならない（相手の送信を待たなければならない）時間が増加し、省電力効果が低減します。

この仕様は、逆に言えばBLEは数十msあたり1ms以下の、送受信タイミング以外だったら、アンテナ等の無線機器を使用せず、その時間はアンテナを無線LANに利用させても問題がありません。そこで、無線LANとBLEのアンテナ共用では、BLE側が送受信するタイミングに同期した信号によって、アンテナをBLE側に占有させ、それ以外の時間は無線LAN側にアンテナを占有させます（図2）。もし無線LANの通信中にアンテナがBLE側に切り替えられた場合、無線LANのそのフレーム（パケット）の転送は失敗し

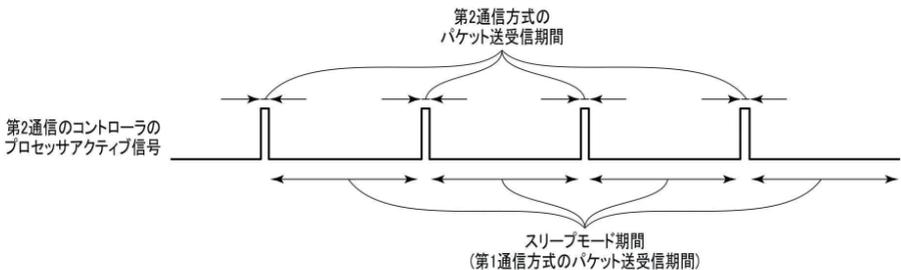


図 2: 特開 2019-144734 よりタイミング説明図

ますが、無線 LAN の再送制御によって、それらのパケットは再送され、通信が維持されます。

この特許公開広報(図3)では、具体的な実施例として、BLE のコントローラが送受信するタイミングとして、頻繁にスリープ・復帰動作を行うコントローラの電源状態を信号として取り出し、アンテナセレクタの入力としているほか、無線 LAN のコントローラ側にも、通信を抑止するべきタイミングとして通知するものを記載しています。無線 LAN コントローラには、外部から、送信すべきでないタイミングを入力できるものがあります。「Co-Ex (Co-Existence: ほかの通信方式との共存) 信号」などと呼ばれ、まさに Bluetooth との共存にも利用されます。

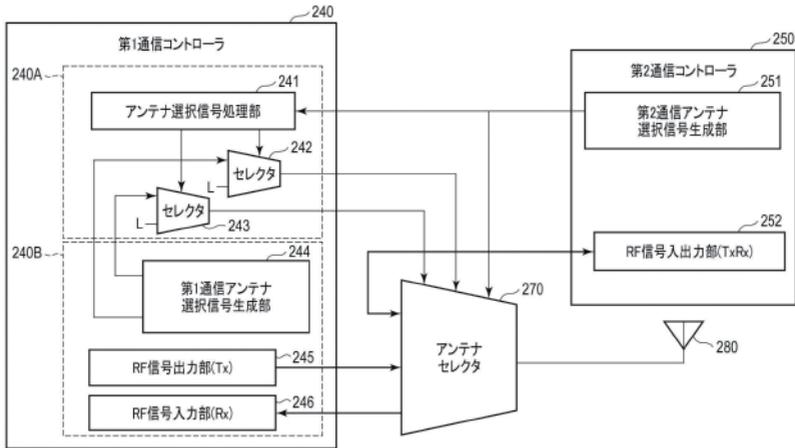


図 3: 特開 2019-144734 よりシステムブロック図
(第一通信コントローラが無線 LAN)

おことわり

本記事に記載されている特許の、将来の予定や実施状況について、発明者の所属ではお問い合わせにお答えすることができず、筆者も本記事以上のことはお答えできません。



田中 瞳

冒頭の特許公開広報の筆頭発明者。この話を何処かでしたかったけど、特許公報が公開されていて助かった。

Airio-Base の半生

(株)クレイン電子 福屋 新吾

Airio-Base は Arduino 互換形状の NXP LPC11U35 マイコンボードとして開発・販売されました。特徴をおさらいすると以下の通りです。

1. SPI 接続のメモ리카ードソケット搭載
2. ARM® Mbed™ 開発対応
3. Arduino UNO 互換形状
4. I²C Grove 互換コネクタ搭載

Airio-Base 自体は、"Mbed で開発する Arduino 形状の基板+各種 IF 付き" と言うものの **FlashAir 開発用 Mbed 互換機** と言っても過言ではありませんでした。しかし、Airio-Base は二つの方向から手足をもがれる致命傷を負わされることとなります。



図 1: Interop Tokyo 2019 東芝メモリーブースにて

致命傷その1 ～主君～

2019 年 1 月、大阪日本橋のシリコンハウス共立にて FlashAir Developers Summit @大阪 Vol.2 が開催されました(図 2、3)。Vol.1 の Arduino UNO R3 から Airio-Base に代わり、Airio-Base + FlashAir の構成で行われました。しかし Airio-Base の今後に期待するもつかの間、事態は良くない方向へと進みます。同年 9 月に Interop2019 の東芝メモリーブースのオープンプレゼンテーションにて、FlashAir Developers サイト閉鎖の案内が発表されます(図 1)。そこから、後を追うように FlashAir は生産終了の道を歩むこととなります。虚しくも Airio-Base は、セミナー開催から 1 年で主君 FlashAir を失う運命を歩むこととなりました。



図 2: イベント開催のポスター



図 3: FlashAir Developers Summit@大阪 Vol.2 の様子

致命傷その2 ～棲み処～

主君 FlashAir を失う運命を歩む中、Airio-Base はもう一つの致命傷を負うこととなります。2020 年に ARM Mbed が Mbed OS 2(classic) のサポート・メンテナンスを打ち切ることを発表しました。Airio-Base 採用のマイコンは Mbed OS 2 対応までとなっており、新しい Mbed には対応していません。その為、オンライン IDE で手軽に開発できる環境はいつ無くなってもおかしくない状況となり、Core ライブラリの提供も今後はどうなるのかわからない不安定な状況となります。つまり、もはや Airio-Base の開発環境に Mbed を使うこと自体が非推奨になったと言えます。こうして Airio-Base は、主君 FlashAir と棲み処たる Mbed の両方を失うことになりました。

残ったものは在庫と思い出

Airio-Base は、主君と棲み処、FlashAir と Mbed の両方を失いました。今は、単なるいくつかのコネクタがついた LPC11U35 基板です。率直に言って、今となっては Airio-Base は使いどころも、使う理由もないでしょう。しかし、**2023 年 9 月現在も受注生産品として生産終了をしています**。多くの経験や忘れられない思い出を作ってくれた Airio-Base は生産可能な間は終売せず、形だけだとしても現行品として残しています。とは言うものの、現実には非情なもので社内には存在感をしっかりと示す在庫が、山とまでは言いませんが、基板・部材・ラベルシールなどが残っています。一部の部材は既に他の製品に転用されていますが、一括で入手したメモリカードソケットは、FlashAir が挿されることなく積まれています(図 4)。また、生産の途中で止まった基板は哀愁の漂うものになっています(図 5)。

ただ、これだけは今ハッキリと言えます、最高の思い出をありがとう FlashAir !!



図 4: ソケットと生基板



図 5: SMD 実装で止まった基板



図 6: 当時のポスター @本社事務所

1 Mbed OS 5 対応移植のチャレンジもしましたが、Lチカより先はできず



株式会社クレイン電子 (@crane_elec) 福屋 新吾
 基板の半田付けをする会社をしています。最近では NAND フラッシュメモリ搭載のマイコンボードを作っています。

電波暗箱を借りて色々遊んでみた

GPS_NMEA_JP (@Seg_faul)

今回執筆するにあたり、電波暗箱と、ちょっと面白そうな機材達をお借りすることができました。ので、その紹介をしていきたいと思います。

電波暗箱 MY1510

電波暗箱とは、無線機器の試験などで用いる電波暗室の小型版で、文字通りの箱です。電波を遮断する箱はシールドボックスと呼ばれますが、遮断に加えて、内部反射の防止のために電波吸収体を備えたものが「電波暗箱」と呼ばれます。

外部に電波を漏らさないため、技適の通っていない機器の試験をすることができますが、電波暗箱は名前の通り箱のため、中に入って試験するというよりは、有線インターフェースを経由して試験することになります。

今回は、クレイン電子さんより、マイクロニクス株式会社製の、MY1510をお借りしました(図1)。実売価格は19.8万円。レンタル会社に頼むと1ヶ月5万円程度で借りられるようです(但し法人のみ)。

この暗箱は、2.4GHzを効率よくシールド・吸収するように設計されているようなので、Bluetooth®や2.4GHz帯の無線LANをテストするのにぴったりですね。試しにiPhoneを入れたらきちんと通信が遮断されました。今回紹介するのはすべて、この箱の中で試しています。

裏面端子は、直流電源端子、USB 2.0-A 端子、Ethernet 端子、D-sub 9ピン、25ピンがそれぞれついているタイプ(IFM4)でした。IFが豊富で助かります。

シールドボックスを自作しようと試みたことがある方(?)ならわかると思いますが、電波をただ単純に遮断するだけなら、箱にアルミホイルを巻くだけの簡単なものでも実現できます(図2)が、ケーブルを通そうとすると一気に難易度が上がります。

今回の電波暗箱はさすがその辺も対策されており、ケーブルを繋いでもきちんと漏れることなく遮断されていたのでとても良かったです。



図1: 電波暗箱 MY1510



図 2: アルミホイルと缶で自作したシールドボックス

FlashAir W-04 の内部シリアル (と TELNET と内部シェル)

まずひとつ目として、FlashAir W-04 を殻割りして、内部テストポイントからシリアル端子を引き出したもの (図 3) をお借りしました。

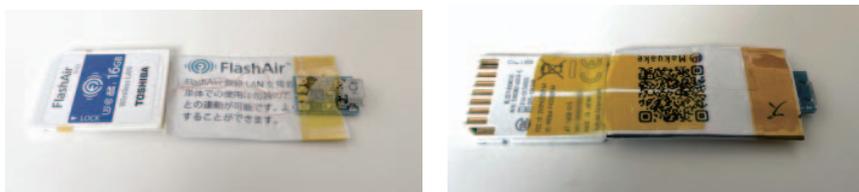


図 3: 内部シリアルに結線された FlashAir とシリアル変換 IC

お借りしたときには既にこの状態だったので、中身については詳細不明なのですが、電源投入時に流れるメッセージや、入力したときの挙動を見るに、fa.serial で使用できる外向きのシリアル端子とは違い、おそらく工場検査や開発のために使用するものだと推測しています。

前提知識として、まず、2018 年の Black Hat² によって、FlashAir 内部には、MeP プロセッサ上で、 μ ITRON を継承した RTOS(T-Kernel) が動作していると解析されています。そしてこの上では、対話的なシェルのようなものが動作しています。(なお、blackhat では、自前のバイナリをビルドして、シェル経由で起動するところまで行われています。)

一般ユーザや Lua 開発者は通常このシェルにアクセスすることはありませんが、いくつかの手段でアクセスすることができます。

大きく分けて 3 つの方法があり、今回の内部シリアルポートの他、eva.cgi を使った方法と、TELNET を使った方法があります。後者 2 つについては殻割りも改造も不要ですので、皆さんも気軽に試すことができます。

内部シェルには、面白いことはあまりありませんが、「level 1」「level 2」「level 1048」でモード切り替え、「help」でコマンド一覧を見ることができます。

1 FlashAir W-04 の Lua 関数。IO 端子をシリアルポートとして使用できる。

2 <https://threatpost.com/hacking-for-sport-a-journey-in-reverse-engineering-a-toshiba-wireless-sd-card/134883/>

eva.cgi

FlashAir に無線 LAN 接続した状態で、<http://192.168.0.1/eva.cgi> にアクセスすることで、シェル³の出力を観察することができます。起動時のログや Lua の実行エラーなどもここに出力されるため、FTLE ではここに自動でアクセスしてエラーなどを観察しています。eva.cgi は知る限り読み取り専用ですが、/SD_WLAN/CONFIG に COMMAND=help の形で記述することで、FlashAir 起動時に単発コマンドを実行し、その結果を eva.cgi で観察することもできます。(公式のファームウェアアップデートや初期化にも COMMAND が使用されているようです)

TELNET

/SD_WLAN/CONFIG に、TELNET=1 と記載し、再起動後 23 番ポートに telnet アクセスすることでシェルにアクセスできます。eva.cgi と違い、起動時のログなどは出ないようですが、代わりに対話的に操作できます。

SD WIFI PRO

SD WIFI PRO は、FYSETC の出しているオープンソースの SD カード型マイコンです (図 4)。ESP32 PICO V3 を搭載しており、無線 LAN と Bluetooth を搭載しています。電波暗箱内に USB 無線 LAN アダプタとともに入れて動かしてみたので、色々解説します (図 5)。

SD WIFI PRO の構造は簡単に言うと、SD カードの端子とホストの端子の間に、切り替え IC を挟み、ホストと ESP32 PICO V3 の間を切り替え可能にしています (図 6)。

従来の SD カード同様の端子は、マルチプレクサを通して SD-NAND に接続されているため、特に何もしなければ普通に 8GB の SD カードとして使用することができます。

ESP32 PICO V3 側の操作によってこのマルチプレクサを切り替えることができ、ホストを切り離して SD-NAND を ESP32 と接続することで、ESP32 側からの操作もとい、無線 LAN 経由でのファイル操作を実現しています。



図 4: SD WIFI PRO

図 5: SD WIFI PRO の試験風景

3 FlashTools Lua Editor: Web ブラウザで FlashAir 上の Lua を書いたりデバッグしたりできる。

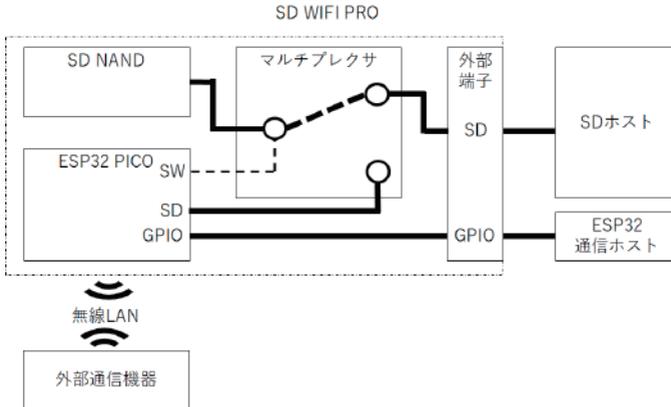


図 6: SD WIFI PRO の内部ブロック

このカードは、通常の SD カードより端子の数が多く、一見 UHS-II 対応にも見えますが、実際には増えている分の端子は ESP32 の GPIO に直接接続されています。これは、ESP32 への書き込み用の端子であり、専用リーダライタ基板を使用することでプログラムを書き込むことができます。

公式サンプルでは、無線 LAN AP (アクセスポイント) の立ち上げ、無線 LAN AP への接続、ファイル一覧の取得、ファイルの削除、ファイルのダウンロード・アップロード、接続状態の取得や変更、SD NAND の開放処理などの、基本的な動作が実装されています。

書き込みには、Arduino 開発環境で ESP32 のボードを追加し、「ESP32 Dev Module」を選択することで書き込むことができます。公式サンプルもスケッチとして提供されています。SD NAND は ESP32 側からは普通に SD カードに見えます。そのため、マルチプレクサの切り替え処理が必要なことを除けば、普通に Arduino で SD カードを扱うのと大して差はありません。

電源を接続すると、特に何もしていなければ電源投入とともにオープンネットワークとして立ち上がります。接続には少し時間はかかりますが、DHCP の割り振りもされるため、ブラウザで 192.168.4.1 にアクセスすることで各種機能を実行できます。

例えば「<http://192.168.4.1/list>」で SD カード領域内のファイル一覧が出ます。

無線 LAN 設定を SD 領域から読み込む機能なども実装されており、Lua インタプリタを導入すれば Lua 環境を作ったりもできるはずで。

興味のある方は電波暗箱などご用意の上、試してみると面白いと思います。



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。

<https://sabowl.sakura.ne.jp/gpsnmeajp/>

意外とまだ需要はあるようなので当分続ける予定です。

FlashAir のたったひとつの冴えたやり方

綾瀬ヒロ

少女コーティーは、誕生日プレゼントに両親から貰った宇宙船で宇宙に一人旅立ち、イーアという種族のシロベーンに出会った。コーティーとシロベーンは意気投合したが、そのうちシロベーンの様子がおかしくなってくる。このままではイーアの実験による犠牲者が増えることがわかり、メッセージを残し二人は黄金の太陽に向けて突入を決意する。

私が FlashAir に出会ったのは、2014 年 11 月に開催された Maker Faire Tokyo 2014 の会場でした。たまたま友人の手伝いで参加していた私は、軽い気持ちで「FlashAir 同人誌」を手にしてしまい、すぐに FlashAir を入手していじり始めると、その利便性を気に入って、鉄道模型への活用をはじめました。FlashAir は世代を重ねるごとに機能アップされ活用の幅が大きくなる一方で、あるときから不穏な空気が漂い始め、2019 年 9 月に「FlashAir Developers」サイトが閉鎖、そして FlashAir 自体の生産も終了してしまいました。今では、市場価格が信じられないくらいに高騰しており、新たに入手することは実質難しくなっています。この状態で、FlashAir の活用を紹介することで「犠牲者」を増やすことが正しいのか。すべての過去を背負って黄金の太陽に突入した方が「冴えたやり方」なのではないかという気もします。しかし、こんな製品が世の中に存在し、その活用に明け暮れた日々があったことを記録に残し、メッセージパイプに載せて流すことにも意義はあるのではないかと思います、この記事を書くことにしました。(前段が長い)

みなさま、ご無沙汰しております。自称 FlashAir エバンジェリストの綾瀬ヒロです。今回は「FlashAir 同窓会」企画ということで、私がこれまで鉄道模型という領域において FlashAir を活用した履歴を振り返ってきたいと思います。

2014/11/24

Maker Faire Tokyo 2014 にて FlashAir 同人誌を入手。

2014/11/29

FlashAir W-02 を入手。とりあえずお約束の GPIO で L チカ。

2014/12/5

FlashAir の GPIO 機能と Arduino を組み合わせて初めて鉄道模型の運転を実現(図 1)。このときに Twitter(X) にツイートしたところ、すぐに中の人(Pochio さん)に捕捉されたのでした。それをきっかけに中の人たちとの交流が始まります。

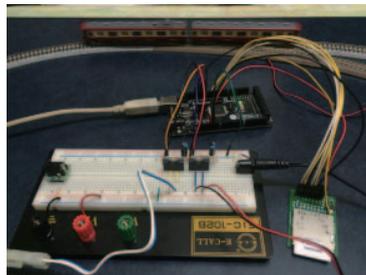


図 1: FlashAir+Arduino
鉄道模型デモ

2014/12/7

FlashAir の共有メモリ機能との運命の出会い!

2014/12/13

FlashAir の共有メモリ機能を使って、Arduino と FlashAir のデータ共有をして鉄道模型の運転をできるように実装ができました。その動作の様子は右の二次元バーコードから。



2014/12/23

後に数々のイベントで活躍してくれるアタッシュケースに仕込んだ鉄道模型ミニレイアウトの製作を開始(図2)。フォトカプラによる在線検知情報を共有メモリに書き込み Web ページ上に表示する仕組みをこの時に初めて作り、共有メモリを使った FlashAir と Arduino の双方向の情報共有を実装しています。



図 2: アタッシュケースデモ作成

2015/2/13

FlashAir W-03 を入手。ついに Lua スクリプトが使えるようになり、さっそく検証を始めますが、当初は Lua で共有メモリを直接読み書きできませんでした。Twitter で「できるようにしてくれー」という叫びをツイート。

2015/3/22

NT 京都に、アタッシュケース型鉄道模型ミニレイアウト(図3)を持ち込み、展示しました。スマホから FlashAir の無線 LAN に接続すれば誰でもすぐに鉄道模型の運転が可能になるということは画期的だったと思います。



図 3: アタッシュケースデモ完成

2015/4/5

Lua スクリプトによって独自スクリプトファイルを解釈し共有メモリを書き換え、鉄道模型を自動運転できるように実装しました。すぐ実装しているように見えますが、実は過去の実装を基にしています。少し前から鉄道模型の自動運転は Arduino とパソコンの連携によって実装してきており(この内容は、日経 Linux 2015 年 5 月号に特集記事として掲載)、その実装内容を Lua に移植したため早々に実装ができたのでした。



2015/4/11

FlashAir ハッカソンが開催され、私もエントリーして当日会場に行くと、技術メンターという役割をいただき、ハッカソン参加者の皆さまをサポートすることになり、ハッカソン中には共有メモリの活用を布教しました。

1 以下、同様に二次元バーコードは動画にリンクします。

2015/4/27

アタッシュケース型につづいて、エフェクターケースに鉄道模型ミニレイアウトの第2弾を作り始めました(図4)。こちらはあまり表に出してこなかったのですが、MFT 2015に出展を目指して製作をし始めました。第1弾との違いは、複数列車を同時に運転できるように自動列車停止装置(ATS)を組み込んだことでした。

2015/6/14

鉄道模型では、やはり長大編成を広いレイアウトで運転したいものです。自宅では部屋に組み立て式の線路を敷設して運転していましたので、それに使えるような運転装置を作成しました(図5)。2列車の運転と4つの転轍機の転換ができるようにしました。基本的な設計は鉄道模型ミニレイアウトと同じです。アルミケースに収めたので、市販品のようにも見えてこういうの市販されないかなと思っていました。(のちに、DSair2というFlashAirを使ったDCCコマンドステーションがDesktop Station様から発売となり、私も2つ購入してメルクリンHOで楽しんでいます)

2015/9/14

ここまでFlashAirで制御していた鉄道模型はNゲージという縮尺1/150線路幅9mmの小さな模型でした。我が家にはGゲージという縮尺約1/20線路幅45mmの大きめの模型もあり、車両が大きいためFlashAir + Arduinoで制御モジュールを作成して車両に搭載しました(図6)。鉄道模型ミニレイアウトと同じように、線路に15Vの直流電源を接続して、線路から電気を供給させ、車載制御モジュールで車両のモーターを制御しました。要するFlashAirでラジコン化したということでした。

2015/10/27

共有メモリの活用として転機となったFlashAir W-03のファームウェアV3.00.01がついに公開されます。しかし、リファレンスドキュメントは更新されず詳細がわからないままでしたが、内部的に共有メモリ用関数sharedmemoryが追加されていることがわかり、試行錯誤の末、関数の使い方を見出して、これまでcommand.cgiのop=130/131で読み書きしていた処理を、



図4: 鉄道模型ミニレイアウト
第2弾



図5: 作成した運転装置

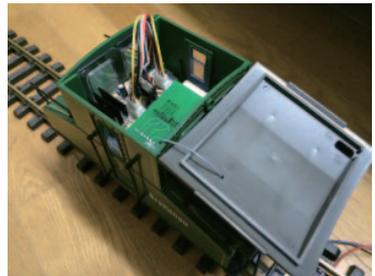


図6: GゲージにFlashAirを搭載



sharedmemory 関数に置き換えました。共有メモリの読み書きレスポンスが格段に良くなりました。

ちなみに、リファレンスドキュメントが更新されたのは1ヶ月ほど経った2015年12月4日ころでした。いま振り返ると、私はなぜ sharedmemory の使い方が分かったんですかね……。(内部情報は貰ってなかった)

2017/6/17

待望の FlashAir W-04 が発売されました。実は発売前に中の人から評価版をお借りして動作検証をしていました。そこで重大なことに気づきます。従来 Arduino などマイコンから iSDIO 機能 (CMD48/49) を使うことで FlashAir の共有メモリを読み書きできていたのですが、W-04 の初期版ではこれができなくなっていました。これは……と、中の人にすぐ通報したのでした。その後、2017年10月5日にリリースされたファームウェア V4.00.01 でこの不具合は解消されました。

その後

ほかにもたくさん作りました。

振り返ると、2014年12月から2015年10月の1年弱で鉄道模型における FlashAir の活用は完成していましたね。なかでも W-03 で追加された Lua 機能の存在は大きかったように思います。その後は細かな改良や Lua スクリプトの拡充によって SPI や I²C などが扱えるようになり、Arduino を介さず FlashAir とモータドライバ、IO エキスパンダなどを直接つなげて、より簡素な作りになできるようになりました。

FlashAir 以外にも安価な無線 LAN モジュールとして ESP-WROOM-02 などが出てきて、よくなぜ FlashAir なのかと聞かれました。私も FlashAir に並行して使ってみました。SD カード本来のストレージ機能と FlashAir が持つ Web サーバ機能は何よりも使いやすく、簡単に操作アプリを実装できる気軽さは他にはないものだったように思います。

FlashAir は生産中止となってしまいましたが、手元には買い溜めた FlashAir が何枚もありますので、これから末長く活用していきたいと思います。いつの日か同じような製品が出てくることを心待ちにしつつ、ここでこの記事を終えようと思います。

皆さんも、なんとか FlashAir を入手して、鉄道模型の制御に活用してみてください。DCC とは違うデジタル制御の世界を自分で作ることができますよ。



綾瀬ヒロ (@ayasehiro)

大手旅客鉄道会社の情報システム子会社からグローバル IT 企業日本法人を経て、現在交通系ベンチャーの役員。鉄道模型をこよなく愛しており、FlashAir を使い倒してきました。

FlashAir 関連の久々の同人誌にお呼ばれいただきました。ありがとうございます。久々にまとまった文書を書いているのでドキドキしている、せいみまさみです。FlashAir をいじっていた当時のアイデアを書き連ね、お焚き上げようと思います。

FlashAir でやりたかったこと

過去の FlashAir 同人誌の記事に音源 LSI を SPI 経由でアクセスし演奏させるという記事を掲載していただきましたが、音の再生について興味があったのもあり、音源 LSI ではない音声再生を FlashAir でできないかと、いろいろと考えていました。そこで I²C と似た規格である I2S はどうだろうか、思い至ったのです。

I2S はオーディオ・データをシリアルに転送するための規格で、同期式シリアル通信です。各種オーディオコーデックで作成された PCM データを D/A コンバータへ入力させたりする用途で使用されています。

FlashAir の大容量のデータを記録できる特性を利用して、無圧縮の PCM データをそのまま SD カードに記録しておき、それをそのままオーディオデバイスに I2S で垂れ流すことができれば、いいなと思っていました。

お試しで W-04 にて信号をエミュレートして動作させようかと考えていましたが、CLOCK 信号の周波数の問題で動作が厳しいと想定されたため Lua 上での実現を諦めていました。Sleep がマイクロ秒単位で指定できればワンチャンできたかもしれません。

Luaでの実現は難しそうなので、I2Sが組み込み関数としてあれば嬉しかったのですが、マニアック過ぎますよね。実現できていたらどんな音質になっていたのでしょうか。音声再生の簡単で正統派の方法としては、I2C や SPI に対応している MP3 デコーダ IC を利用することでしたね。

アマチュア模型の展示即売会のワンダーフェスティバルにも出展者側で参加していることもあって、模型にも興味があります。

2020 年前後から模型に LED を組み込む電飾を楽しむ流れがありました。PIC マイコンなどで模型内部に組み込む作例をよく見かけていたので、FlashAir を使用して、組み込んだあとでも色々変更できると便利だろうなと考えていて、この制御に FlashAir を使えればと、こちら FlashAir 同人誌に記事を掲載させていただきました。

電飾と先の PCM 音声再生の組み合わせもやりたかったことの一つにあります。具体的には、宇宙戦艦ヤマトの模型で波動砲発射のシーケンスを光と音で再現することを

FlashAir単体で実現できれば面白かったなあと思っています。このためにプラモデルも買っていたんですけどねえ(笑)

おまけに、同じ仕様の完成品も出ちゃって、ニーズの読みは当たっていたなあと思わず笑っちゃいました。

さて、Lチカの延長線には、WS2812BによるLEDテープライトやLEDドットマトリクスパネル表示があります。これもFlashAir単品で制御できると、状況に合わせてパターンを変更できたりして便利だろうなと思っていました。

LEDドットマトリクスパネル表示の動作規格にはHUB75というものがあります。こちらでも速いクロックに合わせた動作を行う必要があるため、I2S同様に実現を諦めていました。I2SやHUB75など、早いクロックに合わせてデータを出力するパターンを任意で設定できるような機構があると、よりシリアルデータ通信系は汎用性が上がったのかもしれないね。

光といえば、DMX512という規格もあります。こちらは舞台の照明をミキサーのような機械で向きや光量をコントロールするための規格です。人数が多くない学校演劇で照明コントロールに人員を割くことが難しいため、ここをワイヤレス制御できれば嬉しいなんて話を、高校の演劇部の顧問を長年している実弟から聞いていたので、スマホからFlashAirサーバで動作するミキサーGUIを動かし、ここから繋がっている照明を制御できると面白いかなど考えていました。

いずれもやりたかったことはFlashAir単品では荷が重く、もっと高性能なマイコンで行うべきなのでしょうが、自分はFlashAir単品で動作させることにロマンを感じていたのものでそれ以外は考えられませんでした。今だったら、M5Stackで行うのが良いのでしょうかね。

あと、アンテナのサイズなど物理的な問題が一番あるとは思いますが、やはりマイクロSDカードサイズのFlashAirは欲しかったですね。流石にフルサイズのSDカードは使用する機会が段々と減っていたと思います。ここが一番残念だったのかなあ。

FlashAir と MSX

ふと参加したMaker Faire Tokyo (MFT) 2014に東芝が参加していて、薄い本を配っていたというのが、自分とFlashAirの最初の出会いでした。

何かの可能性を感じ、手の遅い自分には珍しく年末にはW-02用のレトロPC用(と言っても自分はMSX^{もの}者なのでMSX用ですが)のワイヤレスコントローラを作成しました。

レトロPC系のイベントに珍しく参加して、ワイヤレスコントローラを参考出展とかしていたところに、FlashAir Developersの皆さんにサルベージされたのです。

MFT2015 に出展しませんかのご連絡いただき、勢いで 6 月 29 日の横浜某所で開催された出展会議に参加させていただきました（もちろん本職はお休みしました）。

この会議で、展示するのであれば

- ・ MSX でゲームを展示したい
- ・ MSX でゲームを展示するのであれば TV 必須
- ・ MSX 本体を東芝製、TV を REGZA にして展示

を東芝で固められれば、話題になるのでは

とか捲し立て、結果 MSX が一番目立つような展示をさせていただきました（図 1）。その節は大変ありがとうございました。

その甲斐もあり、多くの大小様々なお友達に MSX を触っていただき、また、ITmedia の MFT2015 の記事でタイトルはマイクロソフトが主になっていますが記事の最初は「東芝が MSX を展示」となっていて、一人小躍りして、話題になったことに胸をなでおろしておりました。（こちら記事の二次元バーコードです。）



図 1: MFT2015 での MSX の展示



さて、80 年代の古いコンピュータである MSX が今存在するための意義やら、最低限必要な機能は何か、とかたまに考えています。意義はともかく、2000 年代に欲しい機能としてはやはりインターネットによるネットワーク機能だろうと思うのです。極端な話、BASIC で作られたネットワークゲームとかあれば、新しい何かが開けそうな気がします。

そこで見つけた FlashAir は福音のように感じました。非力な MSX 本体側が主な制御をすることなく、小さなサイズで無線 LAN や各種サーバの機能を持てるのです。

MSX で WEB アクセスする際の問題は沢山あるのですが、地味に難しいのは、すでに主流になっている文字コード UTF-8 への対応でした。MSX で使用できる文字コードは SJIS であり、WEB から入手できる文字はほぼ UTF-8 なので、MSX では変換をする必要があったのです。しかし UTF-8 と SJIS は変換するためには巨大な変換テーブルが必要となるため MSX 本体だけで実現するには現実的にはほぼ不可能でした。

そんなこともあり、先に書いた MFT2015 の出展会議の打ち上げの席で、MSX などレトロ PC で FlashAir の WEB 機能をしゃぶり尽くすためには文字コードの問題があるので UTF-8 と SJIS の変換機能を外部に提供して欲しい、と直接訴えていたこともあって、W-03 でその機能がついたときは、これで MSX にもインターネットの世界が開くことができると嬉しくなりました。（要求した割に使い切れなくてすみませんでした）

1 <https://www.itmedia.co.jp/pcuser/articles/1508/12/news037.html>

そして、無線 LAN 機能については MFT2015 後の 15 年 12 月の MSX ユーザの集まりでは、MSX に RS-232C 機能を追加する仕様である MSX-SERIAL/ MODEM と同様に、図 2 のような機構で FlashAir を中心にして無線 LAN 機能を追加する MSX-NIC という仕様を作り出すのはどうだろうか、と、アイデア出しをしていました。しかし自分の手が遅く結局実現できませんでした。

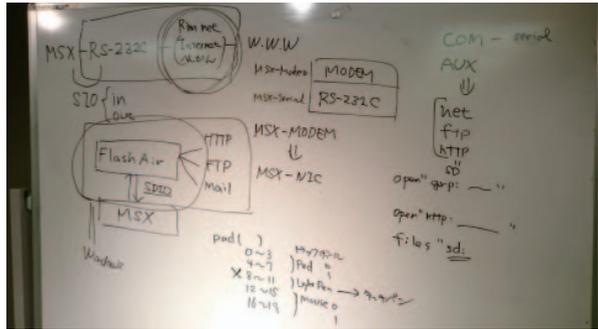


図 2: MSX と FlashAir でネットワーク接続する案

最近 MSX 関係で話題となっている IoT MSX といえる MSX0 について、展開されている内容が M5Stack 上の MSX エミュレータということであり食指が動いていないのですが、この機能を MSX 実機に落とし込むとしたらどうするのか、というお題を今、思考実験して楽しんでいます。

IoT デバイスとして MSX 実機は大きすぎますが、MSX 実機を MSX0 のように遊ぶための実機向けの拡張ハードウェア (カートリッジ) があっても良いと思っています。MSX-NIC のアイデアと MSX0 とを組み合わせた、実機向けの IoT カートリッジですね。これを実際のカートリッジとする場合の一番の問題は FlashAir を使用すればお手軽に実現できるだろうということ、他の手法で行わなくてはならないということです。

そういった点でも FlashAir が普通に入手できなくなったことは本当に残念に思っています。でも、そんなカートリッジが作れるなら、自分の IoT の機能の知識はほとんど FlashAir から得ているので、FlashAir ばい機能なり思想なりを載せていきたいなあと思考実験しながら、野望を持っていたりします。

なにせ、MSX の仕様には東芝が独自で拡張した機能を標準規格として取り込んだ過去があります。MSX 用 RS-232C の仕様もそうであったと聞いています。同じようなことを令和の時代にもできるとちょっと面白そうだなと思っているのです。

そして、FlashAir will never die. とか言いたいですね。あ、MSX も付け足さないといけないか。



せいみまさみ (@masa_seimi)

MSX と音楽と模型好き。FlashAir と付き合い始めたころが自身の結婚したころでもあるので、当時の写真を探している最中にいろんな写真を見てしまい悶絶しておりました。とさ。

4 セグメント 1 桁時計の製作

あいさや

みなさんは 7 セグメントディスプレイをご存じですか？電卓やデジタル時計などで見かける 7 つの画（セグメント）でアラビア数字を表現するパーツです。でも 0 ~ 9 を表現するだけなら 4 ビットで足りるはず。そこで 4 つのセグメントで数字を表現する方法を考え、見やすさの確認用に FlashAir で動く時計を作りました。

コンパイル不要の Lua で簡単にプログラムが書けて短時間で動作テストができる FlashAir はプロトタイピングにも最適です。

4 セグメントディスプレイの提案と利点

崩し字を意識して数字をデザインしました。0 を小さく書く人もいますよね。LED ディスプレイにする場合は、LED の数が減り使用する資源の量を削減できるだけでなく、平均点灯数が 7 セグメント LED の分以下 (46%) になり、省電力にも貢献できるのが特徴です (図 1)。

同時 点灯数	0	1	2	3	4	5	6	7	8	9	平均	比率
7セグ	6	2	5	5	4	5	6	4	7	6	5.0	100
4セグ	1	1	3	2	2	3	2	2	4	3	2.3	46



図 1: 平均点灯数の比較

時計モジュールの製作

アクリル板に穴を開けて透明なアクリル板を象嵌のようにはめ込み、横から LED で照らして数字を浮き上がらせることで LED 部を薄型化しました(図 2)。ユニバーサル基板に LED と電流制限抵抗、SD カードソケット、5V ⇒ 3.3V 変換器、電源端子を取り付けて配線すれば出来上がりです。LED の数は小数点を含め 5 つです。

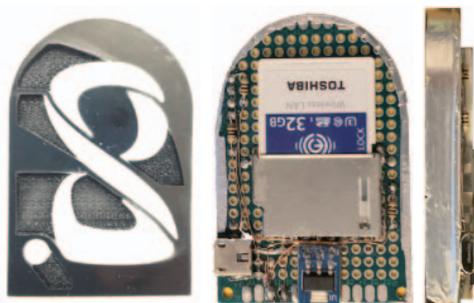


図 2: 時計モジュール

Lua プログラムの概要

USB 電源投入時に無線 LAN を起動して自宅の AP やスマホに自動接続し、現在時刻を取得します。GPS_NMEA_JP さんの libFlashTime.lua を使わせていただきました。NICT のサービスが終了しているので、worldtimeapi.org から取得するように一部修正しています。その後、一定時間ごとに時刻を 1 桁ずつ表示します。数字の切り替えには W-04 で追加された PWM 機能を使い、ふわっと点いたり消えたりさせてみました。

```
for i=0,10 do
  b,s,h = fa.request("http://worldtimeapi.org/api/timezone/Asia/Tokyo")
  if(s == 200) then break; end
  sleep(1000)
end
if(s ~= 200) then return nil; end
str = cJSON.decode(b)
itr = string.gmatch(str.datetime,"%d+")
Year,Month,Day,Hour,min,sec = itr(),itr(),itr(),itr(),itr(),itr()
```

デモンストレーション

MFT 2022 や YOXO FESTIVAL 2022 などに製作したガジェットを出展し、関心を持ってくださった方々に「ちゃんと読める」「デザインがきれい」など好評をいただきました(図 3)。見に来てくださった皆様、感想をくださった皆様がこの場をお借りして感謝申し上げます。



図 3: 4 セグメント 1 桁時計の点灯例



あおいさや (@La_zlo)

FlashAir で遊んでいたら研究所勤務になりました。
見たことないモノをつくるのが好きです。

FlashAir で作る自作 CPU サーバー

にちか

ここは同窓会ですが、はじめまして！ SSD Doujinshi という本で「自作 SSD で動かす自作 CPU」という記事を書きました、にちかと申します！こちらの記事では、2048bitのROMを手作業で書き込む代わりに、自作 SSD を使う方法を提案しました。しかし、この時の自作 SSD に対する書き込みは USB インターフェースであり、まだまだ改善の余地があります。例えば、もし無線通信でROMを随時書き換えられれば、自作 CPU がちょっとした電卓に進化しないだろうか？しかもその接続がネットワークなら、それはもう計算機サーバーと言っても過言では無いはず…！

そこで当記事では、自作 CPU と FlashAir をつなげて、TWS (ただ動くだけの Web Service) というサービスを作ってみたので紹介します。

ハードウェア構成

ROM 回路

ROM 回路の外観を図 1 に示します。ROM 回路は第 4 世代 FlashAir とその I/O 拡張回路から構成されます。I/O 拡張回路は、プログラムカウンタ 8bit を入力し、命令 12bit を出力するため、MCP23017 を 2 個用いました。また FlashAir の制御信号が 3.3V であるのに対し、I/O 拡張回路の出力は 5V² のため、両者の間にレベル変換モジュールを挟んでいます。

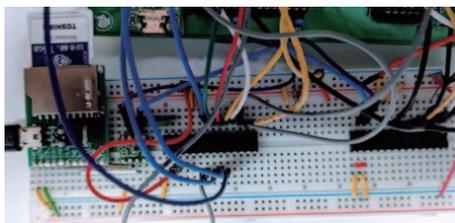


図 1: ROM 回路 (FlashAir)

自作 CPU

自作 CPU の外観を図 2 に示します。自作 CPU は、汎用的なディスクリット IC を 18 個用いています。「CPU の創りかた³」という本の設計を参考に、レジスタを 2 倍 (8bit) に拡張したもので、名前は TD8 といいます。

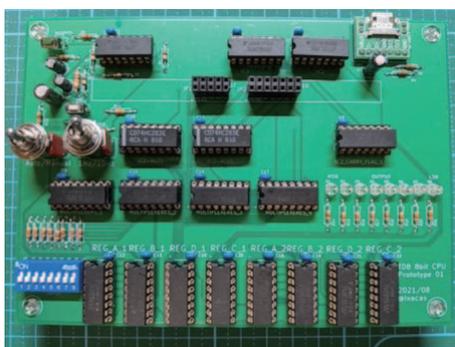


図 2: 8bit 自作 CPU TD8

1 https://akizukidenshi.com/download/ds/microchip/mcp23017_mcp23s17.pdf (23/09/30 閲覧)

2 <https://akizukidenshi.com/catalog/g/gM-05452/> (23/09/30 閲覧)

3 渡波郁「CPUの創り方」株式会社マイナビ出版 (2003)

ソフトウェア構成

本節では、製作した ROM と CPU で、掛け算を計算する方法を説明します。

FlashAir

FlashAir は、TD8 の ROM を生成し、さらにユーザーインターフェースを提供します。サービス画面を図3に示します。ユーザーが入力フォームに数値を入れて「掛け算」ボタンを押すことで、掛け算の ROM コードを生成します。

TD8

TD8 のプログラムは、4bit の命令コードと、8bit のイミディエイトデータの連なりで構成されます。FlashAir が生成する、掛け算を計算する機械語を以下に示します。

```
0x00000000 03070005 0E0F0409 0F000000 00000000
*
0x00000100 00xxyyFF 06020000 08000000 00000000
```

上記のコードに登場する xx は掛ける数を、yy は掛けられる数を代入します。TD8 は yy を xx 回繰り返して足すため、掛け算をした場合と同等の結果を求められます。ただし、TD 8 のレジスタサイズが 8bit のため、256 以上の値は計算できません。

計算させてみる

本節では、6 掛ける 7 を計算させてみました。TD8 を 1Hz 動作モードでリセットし、ブラウザから数字の 6 と 7 を入力後、「掛け算」ボタンを押して約 40 秒待つと、出力ポートに接続した LED に 2 進数で「00101010」と表示されました(図4)。これは 10 進数に直すと「42」で、正しく計算が出来ました。

掛け算 (数値は下に入力)

計算したい数式: ×

- ・ TWS はただ動くだけの Web Service です。
- ・ FlashAir の GPIO 機能を使用して、ブラウザから TD8 に指示を出すことができます。

図 3: TWS の画面

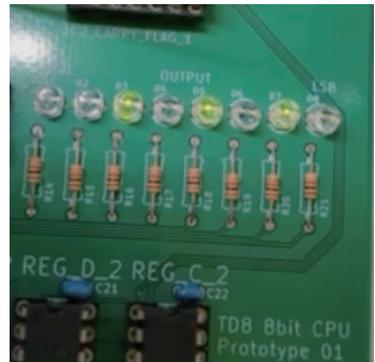
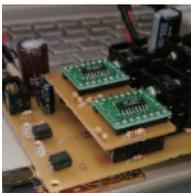


図 4: TWS の計算結果



にちか (@lxacas)

大変便利なデバッガを公開してくださった、GPS_NMEA_JP 氏に感謝いたします。また、I2C 制御に関する知見を共有してくださった綾瀬ヒロ氏にお礼申し上げます。さいごに、本誌にお誘いくださった Pochio 氏と余熱氏に感謝いたします。

世界初の NAND フラッシュは SSD の夢を見るか

にちか

とある余熱氏から「TC584000P」という NAND フラッシュメモリを頂きました(図1)。TC584000Pは、1991年にキオクシアが世界に先駆けて商品化したフラッシュメモリです。これはマイコンから読み書きして遊ぶしかないでしょう!そしてあわよくば、現代のパソコンにもつなぎたい!ということで、まずは Arduino Mega をつかって読み書きしてみました。

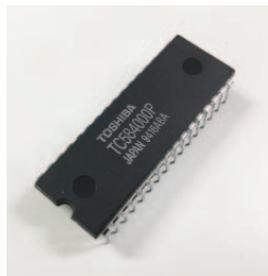


図 1: TC584000P

TC584000P の諸元

TC584000P の諸元¹を表1、2に示します。参考として、株式会社クレイン電子製の JISC-SSD に採用されている、同じくキオクシアから2012年に発売された NAND フラッシュメモリ TC58NVG0S3HTA00³と比較します。

表 1: NAND フラッシュメモリの諸元 (容量)

型番	Register/Page サイズ	Block サイズ	Block 数	容量
TC584000P	512 Byte	4096 Byte	128	4M bit
TC58NVG0S3HTA00	2112 Byte (2048 + 64)	132K Byte (128K + 4K)	1024	1G bit

Register サイズは1度に読み書きできるデータの単位サイズで、Page サイズと等しいです。Block サイズは、データを消去する操作 (= Block Erase) の最小単位です。Block はいくつもの Page から構成されるため、相対的にサイズが大きいです。もしある Page のデータを 1bit でも書き換えようとしたとき、Block 全体の読み出し→消去→新しいデータの書き込みという、3つの手順を踏む必要があります。また、NAND フラッシュメモリは Block の集まりと見なせるため、Block サイズ× Block 数=総容量となります。表1によるといずれの指標においても、NAND フラッシュメモリは 21 年前から大規模化・大容量化が進んでいることが分かります。余談ですが、TC58NVG0S3HTA00 はあくまで耐久性を重視した製品であり、容量は重視されていません。フェアな比較になるかはさておき、最新の多値化・多層化技術を採用したものと 1.67T bit の試作品が Intel 社から発表⁴されています。

1 東芝「NAND EEPROM FT-TC584000P /F/Ft/FR」データシート(1992)

2 キオクシア製 NAND Flash を搭載し、自作 SSD を作ることができる学習向け拡張ボードです。

3 https://www.kioxia.com/content/dam/kioxia/newidr/productinfo/datasheet/201910/DST_TC58NVG0S3HTA00-TDE_EN_31435.pdf (2023/9/30 閲覧)

4 <https://xtech.nikkei.com/atcl/nxt/mag/ne/18/00001/00299/>

表 2: NAND フラッシュメモリの諸元 (レイテンシ)

型番	Random Read	Page Read	Program	Block Erase
TC584000P	15us(29ns/B)	120ns	40us (78ns/B)	10ms
TC58NVG0 S3HTA00	40us(19ns/B)	25ns	330us (156ns/B)	2.5ms

Random Read は、任意の物理アドレスからデータを読み出す際の所要時間です。この操作はメモリセルからレジスタへデータ転送する時間が余分にかかるため低速です。Page Read は、同一ページから連続してデータを読み出す際の所要時間です。この操作は単にレジスタアクセスしか行わないため、Random Read より高速です。Program は、レジスタからメモリセルへデータを書き込む際の所要時間です。Block Erase は、1ブロックの消去にかかる時間です。表2によると、Page Read や Block Erase の処理時間は21年間で1/4程度に低減されていることが分かります。特に Block Erase は全操作のなかで最も処理時間が大きく、ボトルネックになりやすいため、高速化の恩恵が大きいと言えます。一方、Random Read や Program の処理時間は増大しています。しかしこれらの操作は Page 単位で行うため、1Byte あたりの転送時間(表2の括弧内の数値)で比較すると、TC58NVG0S3HTA00 も健闘していることが分かります。またいずれの操作においても、その所要時間のオーダー(ミリ、マイクロ、ナノ)は大きく変わっていません。

まとめると、NAND フラッシュメモリは21年間で高速・大容量化しているものの、その基本的な特性(Read 時間 < Write 時間 << 消去時間)は変わっていないと言えます。

マイコンの選定

最近の NAND フラッシュメモリは省電力化が進んでいるようで、1.8V や 3.3V 駆動のものが主流です。一方、TC584000P はなんと 5V 駆動でした。(時代を感じますね。)また、TC584000P はパラレル信号によるアドレス指定で読み書きするため、ピンが32本もあります(!)。つまりマイコンで制御するには、ほぼ同数の GPIO が必要です。さらに、将来 Read Modify Write することを考慮すると、データの退避先として SRAM も大きい方が良いです。以上の条件から、メモリも IO ピンも潤沢な Arduino Mega を選定しました(図2)。

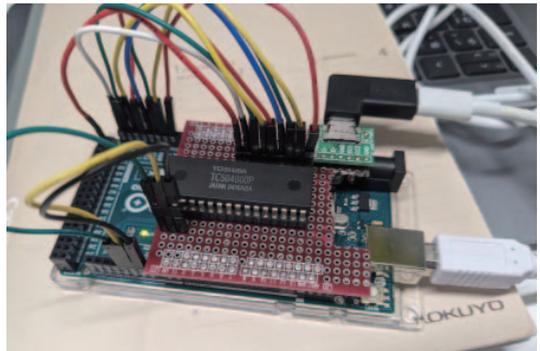


図 2: Arduino Mega + TC584000P

Arduino スケッチ

本節では、TC584000P を読み書きするために書いた Arduino スケッチ（プログラム）を解説します。まず、スケッチは NAND フラッシュメモリに対しパラレル信号を送受信する必要があるため、Mbed™ の BusInOut クラスに似たライブラリが欲しくなります。そこで図3のようなバス入出力クラスを作りました。

次に、一番簡単な Read コマンドを実装してみます。データシートに記載された Read コマンドのタイミング

チャートを図 4 に引用します。タイミングチャートは、各ピンで信号をやり取りするタイミングを図示したもので、基本的にはこの図の順番通りに制御信号をやり取りするだけで良いです。まず読み出したい Page アドレスを ADD0~18 ピンで指定します。次に Chip Enable ピンと Output Enable ピン（どちらも負論理）の電圧を下げ、IO レジスタから Read コマンド (0x00) を送ります。その状態でアクセス時間 (tAcc=15 μ s) 待ち、IO レジスタを読み出します。

```
uint16_t BusInOut::read(void)
{
    uint16_t val = 0;
    for(int i = m_busNum - 1; 0 <= i; i--)
        val = (val<<1) + digitalRead(m_bus[i]);
    return val;
}

void BusInOut::write(uint16_t val)
{
    for(int i = 0; i < m_busNum; i++)
        digitalWrite(m_bus[i], (val >> i) & 0x01);
}

void BusInOut::setMode(int mode)
{
    for(int i = 0; i < m_busNum; i++)
        pinMode(m_bus[i], mode);
}
```

図 3: BusInOut クラスの一部

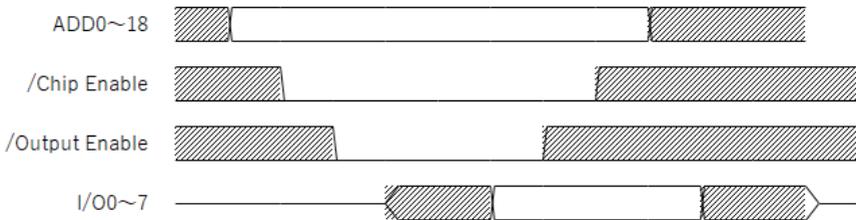


図 4: Read コマンドのタイミングチャート

```
BusInOut IO(11/*IO 0*/, 12/*IO 1*/, 13/*IO 2*/, A8/*IO 3*/, A9/*IO 4*/, A10/*IO 5*/, A11/*IO 6*/, A12/*IO 7*/);
BusInOut COLUM_ADDR(10/*A0*/, 9/*A1*/, 8/*A2*/, 7/*A3*/, 6/*A4*/, 5/*A5*/, 4/*A6*/, 3/*A7*/, A1/*A8*/);
BusInOut PAGE_ADDR(A0/*A9*/, A14/*A10*/, 52/*A11*/, 2/*A12*/, A2/*A13*/, A3/*A14*/, 15/*A15*/, 16/*A16*/, A4/*A17*/, 17/*A18*/);

uint8_t read_1st_byte(uint16_t page_addr, uint16_t colum_addr)
{
    PAGE_ADDR.write(page_addr);
    COLUM_ADDR.write(colum_addr);

    CE = false;
    OE = false;

    IO.output();
    IO.write(CMD_READ);

    delayMicroseconds(15);

    IO.input();

    return IO.read();
}
```

図 5: Read コマンドの実装例

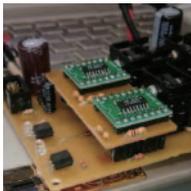
以上の要領で Read コマンドの実装ができました (図 5)。また、Program と Block Erase コマンドも同様に実装し、シリアル通信で動作確認を行いました。

今後の野望

Arduino Mega を USB デバイス化することで、世界初の NAND フラッシュメモリを自作 SSD にしたいと考えています。実は LUFA(Lightweight USB Framework for AVR) という Dean Camera 氏のライブラリに Mass Storage Class の実装があるため、それをを用いることで PC への接続がすんなり成功しました (図 6)。あとは前節の NAND フラッシュメモリを読み書きするスケッチと組み合わせるだけなのですが、ここである問題がおこり、製作が詰まっています。結局、同じ Logical Block Address (OS からみた時のデータアドレス) を 2 回 Read すると値が変わってしまう、大変愉快的なストレージが誕生しました。もしも今後この作品が完成したら、別の機会でお披露目したいと考えています。



図 6: Arduino Mega を Mass Storage として認識させた様子



にちか (@lxacas)

余熱さんと SSD Doujinshi を書いたご縁で、こちらの同窓会にお呼ばれました! 中高で技術工作部、大学でロボメカ工房 (NHK ロボコン+失禁研究会) の活動を経て社会人に。別のページで (ちゃんと同窓会らしく?) FlashAir の工作に挑戦したので、そちらもぜひご覧ください!

1年ぶりの同人誌ですが、ご機嫌いかがでしょうか。「今年も懲りずに配っているなあ」とか、「去年まで SSD ネタだったのに、なんか既視感のある表紙だな〜」、などと思ってくださった方も、そうでない方も、手に取っていただきありがとうございます。久々に FlashAir ネタの 1 冊ですが、おそらく今回限りの特別版でして、本企画はあくまでもファンによる同人誌であり、製造元とは何の関係もないことをここに申し添えておきます(汗)。

さて、FlashAir ネタで初参加となった 2014 年の Maker Faire Tokyo から記念すべき 10 年目となる今年は、奇しくもスポンサーではなく Maker として初参加することになりました。同人誌も 9 冊目にして初の自費製作です。今年は色々事情がありまして、Maker Faire Tokyo に参加するために残された選択肢は、「自費で参加する」の 1 択となり、それなりに悩みました。正直申しますと、出展準備はネタ作りから考える必要があり、毎年なかなか大変ですが、加えて今年は先立つものがないので出来ることが限られてしまいました。とはいえ参加を見送れば同人誌の無料配布も断念となるわけですが、2019 年に最後の FlashAir 同人誌を発行した際にいただいた、「有料でもよいので今後も継続してほしい」というありがたい言葉を思い出しますと、途絶えさせたくない思いが強くなり、本同人誌の企画が動き出しました。振り返ればこれまでの 10 年間、2020 年の MFT だけが不可抗力でどうにも参加できず、翌年テーマを変えて何とか復帰にこぎつけた経緯があります。ここでまた途切れてしまうのは惜しいと考え、有志にて久々に FlashAir などをネタに作ってみよう、となったのがこの 1 冊であります。

それで、いまさら FlashAir ネタで何を書くか、という話になるわけですが、過去の FlashAir 同人誌に書いていない話で、関係者の中でも特に印象に残っているエピソードを 1 つ取り上げてみたく、それがこの「室蘭紀行 2016」になります。とはいえ当時からすでに 7 年が経過していますので、正直思い出せないことが多く、Twitter (現 X) のダイレクトメッセージの記録や写真などからわかる範囲で書いてみました。

なぜ室蘭に行ったのか

そもそもどうして室蘭に行くことになったのか、その背景を説明しないとただの旅行記になってしまいます(汗)。今から 10 年前の話になりますが、当時 FlashAir のカメラ以外の用途を模索し始め、2013 年の秋と 2014 年の春にオープンソースカンファレンスというイベントに出展しました。これをきっかけに、様々な技術者とはつながりを持つようになります。そして 2014 年の秋に Maker Faire Tokyo に初出展し、FlashAir 同人誌の第 1 号を製作、配布しました。この同人誌の配布は、さらに多くの技術者に FlashAir の存

在をアピールすることになり、FlashAir を使って面白いことを考えてくださる方が徐々に増えていきました。このように外部の開発者とのつながりを持ち、彼らとの関係を重視しつつ自社製品のコミュニティを広げ、製品開発に生かしていくマーケティング手法を DevRel (Developer Relations) とよぶのだそうです。この言葉を知ったのは FlashAir の販売が終了した後だったと思います。

FlashAir 同人誌の 1 号を手にとった方の中に、本誌の執筆陣の一人である綾瀬ヒロさんがいらっしゃいました。FlashAir を使ってタブレットからワイヤレスで鉄道模型の運転を実現し、さらに自動運転も可能にされました。綾瀬さんにはこの鉄道模型の記事を FlashAir 同人誌の 2 号に寄稿いただき、最後の 6 号まで連続して毎年執筆いただいたのでした。

私個人の認識ですが、FlashAir のエバンジェリストはお二人いらっしゃると考えていて、その一人が綾瀬さんであり、もう一人が本誌に寄稿いただいている GPS_NMEA_JP さんです。突如現れた彼の存在は、次々と我々に衝撃的なネタを提供し続け、関係者の中では綾瀬さんと並び一目置かれる存在でありました。

8 年前の話なので時系列が怪しいのですが、まず 2015 年 3 月に第三世代の FlashAir W-03 が発売になりました。この世代から Lua スクリプトに対応し、ユーザーが作ったプログラムを FlashAir 上で実行可能になったのでした。すると発売して間もない 3 月 26 日に「Pomera で Lua スクリプトを動かしてみる」という動画をニコニコ動画に（その後 YouTube にも）アップする方が現れます。ちなみに Pomera とは、キングジム社が製造販売する小型のノート PC のような機械で、メモや文章を作成することに特化したものです。この Pomera 上でスクリプトを書いたり、スマホにプッシュ通知を試みたり、さらにはツイートを試みたりする動画が次々とアップされました。4 月になると「ニコニコ技術部に FlashAir を紹介したい」という動画がアップされました。どれもよくまとまっていたため、関係者の中で「これを作っているのは一体何者か？」と話題になったのでした。それが GPS_NMEA_JP さんだったのです。

当時のことをご本人に確認したのですが、FlashAir を最初に触ったのが 2015 年 3 月 25 日とのことで、その日から猛烈な勢いで使い倒している様子が Togetter の「FlashAir をいじった記録」にて確認できます。その約 2 週間後の 4 月 11 日から参加者約 100 名による「FlashAir ハッカソン」が開催されたのですが、その猛烈に使い倒す過程で発見した FlashAir の Lua スクリプト関数のバグを報告いただき、運営を手助けしてくださいました。

そしてこの年の 12 月に GPS_NMEA_JP さん作の、ブラウザ上で動く Lua スクリプトエディタ「FlashTools Lua Editor (FTLE)」が登場し、関係者の度肝を抜いたのでした。ブラウザ上で使えるということは、すなわち FlashAir 上で Lua スクリプトの編集、実行、デバッグが可能になるということです。それまではパソコン上で書いた自作の Lua スクリプトを FlashAir に書き込み、その FlashAir 上でスクリプトを実行するのですが、コーディ

ングが間違っているとうんともすんとも言わなくなり、デバッグのヒントになる情報が得られないので困ったものでした。この課題を解決する優れものが突如登場したのです。まさしく DevRel が成せる技といえましょうか。GPS_NMEA_JP さんの凄さはこれに留まらず、FlashAir のファームのアップデートをした際に、公開していなかった機能を次々と発見してしまうのでした。しまいには GPS_NMEA_JP さんの Twitter アカウントが @Seg_Faul だったので、社内で FlashAir についてわからないことがあれば「Seg_Faul に聞け」というのが合言葉になるほどでした。

ここまでお読みいただければ、おおよそ当時の状況についてご理解いただけたかと思います。短期間でこれほどの実績を残した GPS_NMEA_JP さんですが、FlashAir の関係者は誰も会ったことがありませんでした。会ったことがなければ、妄想だけがどんどん膨らんでしまうわけです。どんな天才凄腕技術者なのだろうかと、綾瀬さん含めて関係者の興味はつきません。GPS_NMEA_JP さんは、一体何者なのか？ どこで何をしている人なのか？ ぜひとも本人を一目見たい、一度でいいから会ってみたいとなったわけです。

どの時点で判明したのかは今となっては分かりませんが、FlashAir ハッカソンが開催されたころ、GPS_NMEA_JP さんが当時北海道の室蘭にお住まいの学生さんとわかりました。それから約 1 年後の 2016 年 6 月に、FlashAir 同人誌の発案者である余熱さんが札幌のイベントに行くことになり、その足で GPS_NMEA_JP さんに会いに行くことを提案したのではなかったかと思えます。これに綾瀬さんが呼応され、お仕事で関わられていた北海道新幹線に乗って行くという話になり、それならば私も飛行機で向かって 3 人が室蘭に集合し、GPS_NMEA_JP さんに会おうという話になったのでした。

そんなこんなで 2016 年 6 月 25 日に室蘭で GPS_NMEA_JP さんに会う作戦が実行されたのでした。

旅行計画の立案 (2016 年 6 月 21 日 (火))

事前の計画では、綾瀬さん、余熱さん、私の 3 名は 6 月 25 日 (土) に別々に北海道入りし、夕方に室蘭に集合。宿は「エスカル室蘭」の和室で 3 人が仲良く寝る、というものでした。綾瀬さんは新幹線で北海道入りし、特急で室蘭に向かって 15 時に到着する計画でした。余熱さんは札幌のイベントに参加したあと室蘭に向かうので、20 時ごろに到着の予定でした。私は午前中に秋葉原で開催される X68000 (1990 年前後に流行った有名なパソコン) 関連のイベントと、技術書典という技術系同人誌のイベントをチラ見した後、15 時羽田発の飛行機で新千歳空港に向かい、そこからバスに乗って 18 時半ごろに室蘭到着の見込みでした。この計画では綾瀬さんが最初に GPS_NMEA_JP さんに会うことができると考えられ、どんな人か報告してほしいとお願いした記憶があります。

ちなみに GPS_NMEA_JP さんによると、「北海道は高速バスのほうが鉄道より到着時間が安定している。鉄道はどこかで事故があると数百キロ離れたところまで影響するの

で、ひどいときは4時間遅れとかになる」のだそうです。鉄道で室蘭に向かう予定の綾瀬さんは元鉄道屋さんでもあるので、じわじわダメージを受けていましたが、まさかこの懸念が現実になるとは、誰も予想しなかったのです・・・。

2016年6月25日（土）

いよいよ室蘭旅行の日を迎えますが、函館付近は朝から悪天候とのこと。そこに11:30頃、綾瀬さんから「新函館北斗駅から札幌方面の特急が運休になってしまいました。とりえず13:08発の臨時が出るようですが・・・」との連絡があり、不穏な空気が流れます。一方で、この日に発売された雑誌「Interface」にFlashAirの特集記事が掲載されることになっていたので購入してみると、事前に提出したソラちゃんのカットが1枚しか採用されなかったことが発覚。特集のトビラ用の絵も出したそうですが、残念ながら採用されなかったようです（仕方ありません）。ちなみにこの号は、FlashAirの特集が掲載されているのに表紙に「500円Wi-Fi」と書いてあって（かのESP***のことですね）、FlashAirの厳しい戦いを暗にほめかしているかのようでした（汗）。

私は新千歳空港に向かう飛行機に乗り、定刻通り15時に出発。生まれて初めて北海道に向かいます。空にはきれいな雲海が広がって見えました。空港着陸間際には窓から防衛省東千歳通信所の巨大な円形のアンテナ、通称「像の檻」を見ることができました。そのころ綾瀬さんから、運休になった特急の代わりに「東室蘭駅19:04着の特急に変更した」と連絡がありました。しかしその後、座席がすべて自由席になってしまい「時刻通りにつくか分からない」とのこと。一方余熱さんは、イベントを終えて16:30に札幌駅に向かって出発とのことでした。

私は無事に新千歳空港に着陸し、16:42発のバスで室蘭に向かうことにしました。GPS_NMEA_JPさんによれば終点まで乗ればよいそうで、18:37に目的地の観光協会前（産業会館前）のバス停に到着予定でした。同じころ余熱さんも札幌に向かおうとしていましたが、絶妙な方向音痴を發揮し、電車かバスで迷った挙句GPS_NMEA_JPさんにバスのほうが良いと諭され、17:10発の高速むろらん号で向かうことに。室蘭到着は19:40の予定とのことでした。

17:40ごろ、綾瀬さんは無事特急に乗れたようですが、内浦湾をはさんで室蘭のちょうど反対側付近を走行中。予定より大幅に遅れている状況に思わず、「海を渡って行きたい・・・、ここに新幹線を通して（うわごと）」とつぶやきます(w)。一方私は順調にバ



図1: 室蘭やきとり

スで室蘭に向かっていました。17:55 ごろ、GPS_NMEA_JP さんから名物の室蘭やきとりを、地元の名店「鳥辰」さんで24本確保したと連絡がありました(図1)。余談ですが室蘭やきとりは鶏肉ではなく豚肉の串焼きです。鳥辰さんの室蘭やきとりはふるさと納税で手に入ります。とてもおいしかったのでおすすめです。ちなみに、この日は開店1時間ですすでに一部品切れになっていたそうで、さすが地元の名店ですね。

18:10 ごろ、綾瀬さんは3分遅れで長万部駅を出発。JR北海道のディーゼル特急が頑張って室蘭を目指します。18:23 ごろ、有名になった無人駅の小幌駅を通過と連絡が入りました。一方私はまもなく目的地であるバスの終点(観光協会前)に到着しようとするところ。当初の予定では最後に到着するはずだった私が、まさかの1着になるとは。まるでリアル桃鉄で知らず知らずのうちに順位が逆転してしまったかのようです。



図2: 旧室蘭駅駅舎

18:27 に GPS_NMEA_JP さんから「バス停の待合のところで立っています」と連絡がありました。いよいよ、SNS や動画などで抜群の技術と才能を知らしめて、関係者をざわつかせ続けた天才に会える。どれほどギラついた人物が待っているのだろうか、緊張しっぱなしでした。運命の時が近づきます。

18:40 ごろ、バスが終点に到着しました。周りは薄暗く、終点まで乗ってきたのは私ただ一人でした。運転手さんに会釈して前方の扉からバスを降りました。バスの進行方向左側はやや暗いためはっきりとはわかりませんが、土砂崩れ対策がされたような丘の壁面が見えました。右側には大きめの建物が見え、あとで旧室蘭駅駅舎(図2)と知りました。前方には屋根のある待合所があり、そこに人影を確認できました。その人影がこちらに近づいてきます。おもむろに私は「Seg_Faul くんですか?」と声をかけました。するとそこに現れたのは、気恥ずかしそうな表情と親しみやすい印象の超好青年ではありませんか。思わず「やっと会えた、ついに会えた・・・!」と口から出たのを覚えています。その感激度合はこの数年で一番だったのは間違いありません。

しかし一方の GPS_NMEA_JP さん、東京から見知らぬ3人の大人が学生である自分に会いに来るので、前の日から緊張でよく眠れなかったのだそうです。なんともご迷惑をおかけし申し訳ない次第でした(汗)。その後、綾瀬さんからは「東室蘭駅に19:11 ごろに到着し、19:21 発の室蘭行きに乗る」との連絡がありました。そして19:36 に綾瀬さんが無事室蘭駅(図3)に到着、余熱



図3: 室蘭駅

さんも 19:47 ごろに中央町のバス停に到着し、全員が無事合流したのです。

全員集まりましたので、まずは一次会として室蘭の海の幸をいただきましょうということで、居酒屋に向かうことにしました。その道中、室蘭の商店街(図4)を散歩し室蘭やきとりの「鳥辰」さんの前を通過しました。しかし通りを歩いていると、次第になんとも寂しげな雰囲気を感じるのです。といいますのも、ほとんどのお店が閉まっていたのです。この日たまたまお休みというわけではなく、完全に店じまいしてしまったようでした。街灯だけが明るくて、お店の明かりは全くないのです。それはもう尋常ではないレベルの寂しさです。

室蘭といえば、製鉄や石油関連で栄えた道内でも有数の工業地帯であり、国立の室蘭工業大学があることでも知られています。かつては駅から石炭を運び、また製鉄業が盛んだったので、室蘭の街はとても栄えていたそうです。しかし石炭や製鉄業の衰退で1970年代には18万人いた街の人口が、現在は8万人以下に激減してしまったとか。それで室蘭の市街地は、交通の便がよい東室蘭に移ったそうなのです。

その室蘭の旧市街地ですが、余熱さんが降りた中央町のバス停がアーケード街にありまして、おそらく当時はこのあたりが中心街だったのではないかと個人的に推測したのです。そのバス停の近くに閉店した大きなパチンコホールがあり、夜だったので中はよくわからなかったのですが、ガラスの自動ドアの向こう側に「テナント募集中」の大きな看板がよく見えました。その時は特に違和感はなく、その場を後にしたのでした・・・。

一次会の居酒屋に到着し、お刺身をいただきながら GPS_NMEA_JP さんと歓談しました。みんなで Interface 誌の FlashAir 特集の話や、自己紹介や近況で盛り上がりました。その後宿泊先にて二次会をすることになりました。GPS_NMEA_JP さんに予め確保いただいていた鳥辰さんの室蘭やきとりとご対面です。大変おいしくて、その年の年末にふるさと納税で注文したのでした。しかしここにきて、前日に勤務先(かの有名な)CEOと徹夜でオンライン会議をしていた綾瀬さんがダウン。最後は余熱さんがGPS_NMEA_JPさんにたしなめられるという、カオスな状況で二次会を終了したのでした(図5)。



図4: 室蘭の商店街



図5: エスカル室蘭で二次会

2016年6月26日（日）

一夜明けて、GPS_NMEA_JP さんに室蘭の街を案内していただきました。前日に私が降りた終点のバス停近くに「室蘭観光協会」の建物があるのですが、ここが先述のとおり旧室蘭駅の駅舎でして、中にはかつての駅名標や駅員の帽子、閉塞区間で使うタブレットなど室蘭駅にゆかりのあったものが置いてありました。室蘭の観光に関する情報も掲示されていたのですが、その一角にラーメンの小さなフィギュアがずらりと並べて飾ってあるのに気が付きました(図6)。そこには「カレーラーメン」の文字がありました。室蘭のもうひとつの名物です。つまり、このたくさんのフィギュアは室蘭市内の各店で食べられるカレーラーメンのミニチュアであり、いかに多くのお店がカレーラーメンを提供しているのかがよくわかるのでした。



図6: カレーラーメンの小さなフィギュア

というわけで、この日の昼食はカレーラーメン(図7)を食べに行くことになりました。お店はいくつもありますが、今回はGPS_NMEA_JPさんおススメの「味の大王室蘭本店」にお伺いしました。お店は日曜のお昼ということもあって、お客さんがたくさん入っていてほぼ満席でした。カウンターの上にはサイン色紙がきれいに並べてあり、かなり有名なお店のようでした。私はカレーうどんの類をほとんど食べないと、前の職場の食堂に「カレーチーズラーメン」というメニューがありまして、これの影響でなんとなく抵抗があったのですが、食べてみるとラーメンとしてかなりおいしく、しかも出汁が効いてうまみがしっかりしているながら、きちんとカレー味のラーメンで衝撃的でした。機会があればもう一度食べてみたいと未だに思っています。



図7: カレーラーメン

さてカレーラーメンを食べた後、再び昨晚歩いた室蘭の商店街に行ってみることにしました。暗くてよく見えなかった通りは、昼間に歩くと細部がよく見えてしまい、東京から来た一行は室蘭旧市街地の厳しい現状にさらなる衝撃を受けたのでした。ちなみに商店街にはまだ営業中のお店がいくつか存在し、その中におもちゃ屋さんがありまして、なんと昔のゲーム機の本体や周辺機器など、かなり懐かしい商品が新品で売られていました。しかしその向かいのお店は2階の窓ガラスが割れたままで、再びショックを受けました。

アーケード街に出てくると昨晚余熱さんを迎えに行った中央町のバス停があり、近くにはタベ見たテナント募集中のパチンコホールがありました。しかし昨晚は暗くてよく分からなかったのですが、一行はなにやら様子がおかしいことに気が付きました。このパチンコホール、よく見ると店内に青空が広がっているのです(w)。状況を理解することができず、慌てて全員でパチンコホールの全景を見るために移動しました。すると、お店の前面は残っていたのですが、後ろがまったくくいのでした・・・(図8)。まるで映画のセットのように正面だけしかなくて、斜めから見たら天井も店の奥もまるっきりなくなっていたのです。この状況でテナント募集中って、テナントは雨が降ったらどうすりゃよいのでしょうか・・・(汗)。



図 8: 青空の広がるパチンコホール

その後地球岬という名所を観光し、東京からきた3人はGPS_NMEA_JPさんとの再会を約束して、室蘭駅から無事帰路についたのです。その再会の機会は案外早く来て、翌年の2017年2月に、GPS_NMEA_JPさんが就職活動で上京するタイミングに合わせてFlashAir Developers Summitというイベントを開催し、ここで室蘭に行けなかった関係者含めてみなさんでGPS_NMEA_JPさんを歓迎したのです。

室蘭旅行記はこれでおしまいです。GPS_NMEA_JPさんとの出会いと室蘭という街での貴重な体験は、その後のFlashAirのDevRel活動に大きな影響を与えたことは間違いなかったと思います。今回おそらく本当に最後となるこのFlashAirの同人誌に、室蘭での出来事を記すことができよかったです。

(おまけ) 動画

右の二次元バーコードから当日撮影した動画を閲覧できます。室蘭の雰囲気や、FlashAirエバンジェリストたちの当時の雰囲気をお楽しみください。



6月25日

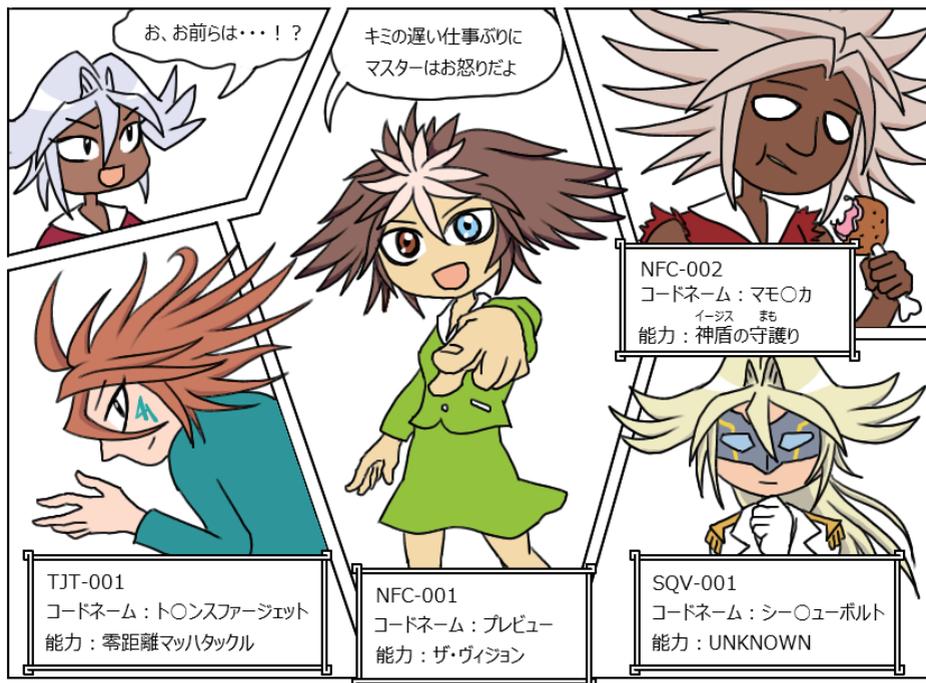


6月26日



Pochio (@i_love_nintendo)

元自称FlashAir芸人です。室蘭旅行のビデオが出てきたので記事を書くために確認したのですが、みなさん若くて7年という歳月の長さを感じました。ところで来年は無事に出展できるかどうか、一抹の不安を感じています・・・。



「フラッシュメモリのひみつ」への道

2017年12月
マイカーフェア開催
MFSZから帰国して
しばらく後――

1通のメールが
届いた

学研プラスです
お仕事をお願いか

ええ!!
学研!?

それは学研からの
「ひみつシリーズ」
執筆オファーの
メールだった

正式名は学研まんがでよくわかるシリーズ

学研の
ひみつシリーズ!!

それはとだが
目標としている
ものの1つだった!

きっかけは
「ホームセンターてんこ」
について書かれた
1本の書評だった

「てんこEx」にも
掲載させてもらった
その熱い書評を

筆者である
たまごまごさんの
了承を得たので
読んで欲しい



●少女はドリルを手に持って●

自分が始めて「ドキッ」と恋に似た感覚を感じたのは、紛れもなく**学研の学習と科学**でした。「主人公」「ヒロイン」「博士とかロボとかっぽいやつ」という組み合わせのマンガのヒロインの**かわいさ**にいつも恋をしました。ぼくの科学的な好奇心と一緒に満たす、元気で明るい女の子たちに。そんな感覚をぐいぐい思い起こさせるマンガが「ホームセンターてんこ」です。きちゃったよ。表紙を見て、はっ！と心惹かれる、学研的デザイン。いいですか、少女の手にドリルですよ。男の子のロマンアイテムのドリルを持つ女子高生ですよ。絵柄のやわらかさといい、学研世代の僕らの心をぎゅうぎゅう締め付けるこの感覚。これは初恋の人に会ったあの感覚だよ。

ヒロインの「てんこ」が、とにかく工作が好きで、工具に魅入られていく様がキュート、男の子心をくすぐります。あ、男心じゃないですよ、「男の子心」。もちろん工具に一番気合が入っているのは少年。しかし、僕らの見たいのはそんな主人公の姿だけじゃあないだろう。見たいのは、夢中になって何か没頭する少女の姿なわけです。それを見透かしたかのように、主人公「てんこ」がもうめりめりと、恋人のように工具に惚れ込むさまがかわいくてしかたない。



新アイテムが出るあたりも、学研やコロコロ・ポンポンのあのときめきがよくあがってきます。まさに少年マンガの王道。そんなわけでぐんぐんハート刺激されるのは、いまやもうパパだったり会社員だったりする大人の男性。工具を片手に木っ端にネジとめてわくわくした経験を思い出しましょうよ。そこに「わー！すごい!!!」って本気で喜ぶ女の子がいることを想像しましょうよ。ほら、ワクワクするじゃないか。

●ヒロインはつねにキュートで●

こういう作品の女の子は「かわいい」だけではないけません。常に前向きで元気。そして好奇心は人一倍パンパン。体もムチムチパンパン。これ男の子の夢ね。見てくださいよこの健康的なふともも。ダイエットなんてしません。運動も大好きなので健康美



そのもの。そんなピチピチボディ(死語)を包むのは**ホームセンターの制服**。うわ、たまんない。なんとも見事なまでに、少年から見た理想の少女像です。それに対し、少年も惹かれてはいても、工具や材料の方に夢中なあたりもまたいいのです。だから多少てんこが無防備でも、話は性欲に満ちません。読者側としては、もう後ろの子みたいにドキドキなんですけどもね。そのへんの「工具に夢中」「その姿がかわいい」の二点が描き分けられているテクニクがオヤジツボを押します。



●もちろん、親子で読んでほしい●

なんだかおっさんくさいヒロイン像の話で終始しましたが、基本は「ものづくり」の楽しさで満ちているので、パパさんと息子さんと読んでほしい。特に息子さんが読んだら、**まず間違いなく「てんこ」に恋をするはず**。そこがいいんだよ。ぼくらが学研ヒロインに恋をして、それが科学への興味に副次的につながったように、イキイキとしたヒロインを見て、何か作りたいな—という気持ちがわいたら素敵じゃないですか。大人は懐かしく、子供は少女に恋をしながら、世代を超えて楽しめる。マガジンらしいヤンキー臭も漂わせながら、ひたすらかわいい「てんこ」にメロメロです。こういう女の子がすきなのは、多分学研のせいです。ありがとう学研。ありがとうとだ先生。



思い返してみると
「Mr.釣りどれんも」
「ホームセンターてんこも」
自分が知らないことを
取材したり体験したり
したものを初心者にも
わかりやすく漫画に
落とし込んでいた：

それを
「学研風味」と
言われたことが
ストンと
腑に落ちたし

それがとても
自分に向いているし
楽しいと感じている
ことに気付かされたのだ



それから
「いつかは学研」
という思いを
頭のどこかに
持っていた

スカイツリーまんがは
かなり学研的学習まんがで
本当に楽しかった

RCマガジンの連載も
知らない世界を取材
できて楽しかった




RCマガジン2014.6~2015.12 マンガ東京スカイツリーのすべて2012.7

そんな学研さんからの
仕事のオファーに
大感激だった！

え…

しかも！
この仕事には特別な
運命を感じた

お願いしたいのは
「フラッシュメモリのひみつ」
です

東芝メモリが
クライアント
さんになります








動き出したのは
違う歯車でした(笑)

あ！
ああ〜！！



Maker Faireで
一度ご挨拶した
Y澤です

こないだの深圳にも
出展してたんですよ

全国の小学校の図書室や
公立図書館に配布されます

学研 まんがでよくわかるシリーズ144

フラッシュメモリ のひみつ

まんが ● とだ 勝之
原案 ● フラッシュメモリのひみつ
制作委員会
構成 ● 谷澤 皓

Gakken

そんなこんなで2018年10月
学研の
まんがでよくわかるシリーズ
「フラッシュメモリのひみつ」
リリースされました！

残念ながら
市販はありませんが
webで全編無料で
見ることができます！

電子書籍ストア学研BookBeyond
→ <https://goo.gl/ko8b7h>

コミック評論「フロガーたまご」による「フラッシュメモリのひみつ」の書評 とだ作品のキャラクターたちは時代とともに成長する



とだ勝之先生が「フラッシュメモリのひみつ」を学研で描いたと聞いたときは「ついにやった！」と驚きました。とだ作品の強みが、ついに

活かされた！と二読者として感じたからです。教育系マンガで一番重要なのは知識の正確さです。そして二番目に重要なのは、登場人物のキャラクター性です。メインターゲットの低年齢層読者にとつて、出てくるキャラクターは誰から見ても魅力的でなければいけない。読んでいて苛立つキャラクターの学習まんがなんて誰も読みません。

子供が感情移入しやすく惹かれる少年と少女は、事象に対して興味を深く持つっており、それを自分のものとして吸収していることが必須です。このあたりは学習まんがであれば必ず「へーなるほど！」という反応をするあたりによく現れています。

とだ先生の描く少年少女は、その反応が義務的ではありません。心の底からのワクワクが行動すべてに出ています。これは「M7」釣りどれん」や「ホームセンターてんこ」などの技術系作品で、存分に発揮されてきたキャラクターの描写です。

主人公のひとりであるサトルは、フラッシュメモリについて新しい事実を知る度に、自分で疑問をさらに深掘りし、夢中になってがんがん進んでいこうとするキャラクターです。このまっすぐな裏表

のないスタイルは非常に少年漫画的で、読んでいて大変気持ちがいいです。

一方でヒロインのアカリはフラッシュメモリに興味を持つてはいますが、思い出の写真の失ったことが一番心に引っかかっています。写真のデータにこだわ

る、いわば過去への思いをととても大切にしている彼女。その心理は、へアピンを失くした時に買い替えるのではなく、元のものにこだわっていた幼少期の様子とシンクロしています。キャラクターの性格付けのこだわりが感じられます（しかも「記録を大事にすることと前進することの両立」は、ラストでさらにテーマとして掘り下げられます。キャラクターの救い方が「貫している」）。

物語を進めるのはサトルですが、作品全体はアカリがサトルを見守るような視点で描かれています。5年生のとき、女の子のほうがちよっと大人びて見えていた感覚が呼び覚まされ、かなり感情移入しやすいです。

キャラクターの魅力を追求めた結果、ラストで結婚式のシーンが入るのに度肝を抜かれました。王道であるのに、テンプレートな



めもりん



アカリ

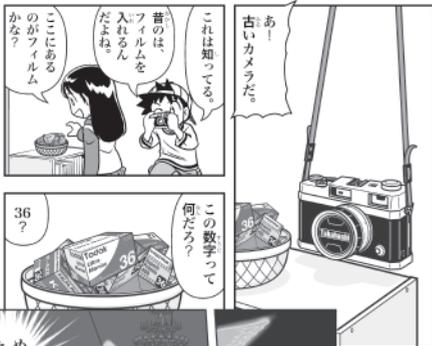


サトル

キャラクターにしない。血の通った、成長する人間ふたりを描こう、という思いが伝わってきます。もしかしたら、子どもたちが読んでこのシーンの良さは、理解はできてもピンとこないかもしれませんが、しかしある程度成長してからこの本を思い出した時、子供の時に感じたワクワクをちゃんと大人になっても持っていることや、幼い時の思い出を大切にしていることの意味が、ぐっと深く感じられると思います。大人の視点で読むと、例の写真と同じくらい、子供時代に冒険して新しいことを知って感動した日々そのものが宝物になっていることに気付かされます。

「フラッシュメモリの誕生と歴史」の章での描写は、昔の学研ひみつシリーズを読んでいた人にはなかなか衝撃的だと思います。小学5年生のふたりは、ダイヤル式電話は見たこと無いし、フィルムは「たった36枚」しか撮れないし、ビデオテープは「たった120分」しか録画できないとびつくりしている。確かに30年40年もたっています。子供たちが驚いてしまうほどに時代は変化しているんだと気付かされます。おそらく最先端を描いたこの漫画も、10年後には時代が変わって、また昔の話になります。でも決してそれは儂くはありません。この本が「記録」を大事にしているように、今の時代を描き留めることには大きな価値があります。

学習まんがは「永遠の子供時代」のような魅力があります。と



だ先生は学習まんが執筆に挑むにあたって、ネバーランドにはとどまりませんでした。時代は変化する、技術は進歩する、子供は成長する。それはとても素晴らしいことだとポジティブに描きます。子供ときには見えていた妖精キャラクターのめもりんは、存在しているはずなのにふたりには見えなくなりました。これは子供時代を中心に描く学習まんがジャンルに対しての、漫画的な挑戦です。子供のときにこのシーンを読んで、そんなの寂しいと納得できない子もいると思います。でも大人になってから読むと、その意味が読者ごとに解釈できて、胸にグツと来るものがあります。

見ていて楽しいだけでなく、心に刺さる人間的なエモもある。「フラッシュメモリのひみつ」を読み終えた子供たちが成長したとき、時代の変化と照らし合わせながら繰り返し読むことで、別の角度から楽しめる強度を持った作品です。



たまごまごさんのブログ「たまごまごほん」
<https://makaronisan.hatenablog.com/>



宮内

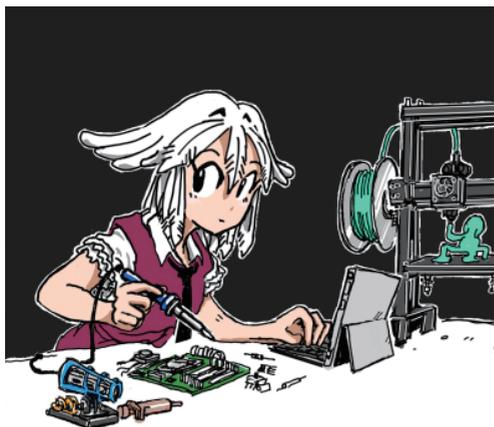
「昔は色々な製品があったな～」と懐かしく思いながら描きました。今回スペースの都合で出れなかった『ミラーリング』もいつか登場させたいですね!?



とだ勝之 (@katsudoren)

お仲間に入れて頂きありがとうございます!このご縁のきっかけとも言える「ホームセンターてんこ」の書評を書いてくださったたまごまごさんに、今回も書評を書いてもらったのもとても嬉しかったです!ありがとうございます!!

ホームセンターてんこを読みたい方は上の二次元バーコードから!



FlashAir 応援キャラクター 「閃ソラ」

■ プロフィール

名前	閃ソラ (ひらめきそら)
年齢	23 歳
職業	新人キャビンアテンダント
趣味	電子工作、アプリ開発
悩み	飛行機の中で電子機器が使えないこと
Twitter	@Hirameki_Sora

■ FlashAir Dousoukai - FlashAir の同窓会

2023 年 10 月 14 日 第 1 版第 1 刷発行

2024 年 9 月 21 日 第 1 版第 2 刷発行

著者: 上岡 裕一 / ragnag / じむ / 伊藤 晋朗 / バブル世代の山女 / 土居 /
寺西 / 余熱 / 田中 瞳 / 福屋 新吾 / GPS_NMEA_JP / 綾瀬ヒロ /
せいみ まさみ / あおいさや / にちか / Pochio / 宮内 / とだ勝之

表紙・本文イラスト: じむ

編集: 余熱 / Pochio / 笠原 / にちか

発行: ひらめき空と月

連絡先: sora_lua@googlegroups.com

URL: https://yone2.net/sora_lua

印刷: 株式会社 プリントパック



<https://flashair-developers.github.io/website/>

