



# FlashAir™ Doujinshi 2

FlashAirの同人誌2

TAKE  
FREE ¥0



待望の同人誌第2弾がついに登場！  
Luaが走るFlashAir最新版W-03で  
IoTワールドにテイクオフ！



# Maker Faire Tokyo 2015 出展にあたり

高田 真里

IoT デバイスとして使っていただくために、皆様に意見をいただいて進化してきた FlashAir ですが、新機能を Geek の方々にご紹介する場として Maker Faire を活用させてもらっています。

昨年、初出展した Maker Faire Tokyo 2014 では GPIO が使えるようになった FlashAir W-02 を使った作品を展示しました。「GPIO が使えるなんて知らなかったよ」という方は FlashAir Developers サイトで FlashAir Doujinshi(2014 年 11 月発行号)をダウンロードして読んでみてくださいね。

今年の Maker Faire Tokyo 2015 では、Lua スクリプトが使えるようになった FlashAir W-03 の作品を展示します。あわせて発行するこの FlashAir Doujinshi 第2号も Lua を使った Make 記事を中心に、豊富な話題で「薄い本」とは言えない厚さになりました。

私たちのこの活動は、FlashAir の IoT 活用事例を開発することが目的です。多くのアイデアが生まれている IoT で FlashAir の利用アイデアを様々な方に考えて実現してもらいたいのです。しかし、エンジニアの皆様が「FlashAir で何ができるのか」を知らなくてはアイデアの実現にも結びつきません。ですから、今は一人でも多くの方に FlashAir の開発経験をもってもらいたいと考えています。

この同人誌の中から面白いな、作ってみようかなと創作意欲をかきたてられるものがあれば、ぜひ、製作にチャレンジしてみてください。面白いものができたら FlashAir Developers のフォーラムに投稿して FlashAir Developers の仲間になってください。まだ工作したことがないという方には、今後はハッカソンへの参加も地方を含めて増やしていく予定なので、FlashAir が API 提供しているハッカソンを見かけたら是非参加してみてください。

さて、今年の Maker Faire Tokyo 2015 の出展は活動成果のご報告の場ともなります。ハッカソンの作品、サポータの方の作品、同人誌の作品が並びます。FlashAir の工作に挑戦してくれる仲間も増えてきたなど実感しています。今年はどんな出会いがあり、次のどんな FlashAir の進化につながっていくのか、今から楽しみです。



## 高田 真里 (@M\_Takada)

FlashAir の商品企画に携わり、現在は展示会や販促活動を担当。展示会でデモをして FlashAir を誉められると単純に嬉しいので、新しい使い方を紹介できるデモ機の開発を画策しています。

# 祝！ FlashAir W-03 発売！



## FlashAir でできること

### 撮ったらシェア！



カメラに入れて使うことで  
写真を撮ったらその場で  
友人とシェアできる！

### web サーバ機能搭載！



会議で資料を共有できる！  
マイコンと繋げて  
データロガーとしても使える！

### GPIO が使える！



FlashAir の信号端子を自由に使える！  
Lチカから液晶駆動まで  
用途は多彩！

### Lua スクリプトが実行可能！



FlashAir がマイコンのように使える！  
天気予報を調べたり、  
ツイッターへの投稿が簡単にできる！

# Contents

	Maker Faire Tokyo2015 出展にあたり	1
	FlashAir でゲーム？！	4
<b>FlashAir 開発秘話と ユーザーの声</b>	FlashAir と Lua	8
	FlashAir 年表	10
<b>FlashAir の新しい 使い道発掘活動</b>	Maker Faire Tokyo 初出展と FlashAir ハッカソン	15
	FlashAir の新たな可能性を ハッカソンから探る！	22
<b>FlashAir を使って 何か作りたい！</b>	FlashAir との出会い	26
	FlashAir Developers へようこそ	28
	FlashAir お悩み相談室	32
	FlashAir を使ったプロトタイプを作ろう！	34
	FlashAir の共有メモリを使い倒せ！	39
<b>製作事例を見たい！</b>	FlashAir のレトロ PC への応用	45
	Lua の PIO 機能で SPI Master	50
	FlashAir でホビーロボットを IoT 対応に！	52
	ATARI 仕様 Wi-Fi コントローラーの制作	56
	名刺サイズはもう古い！? IoT モジュールの大本命！ BlueNinja 誕生！	60

※ 本書に記載されている製品名は、各社の商標または登録商標です。

# FlashAir デゲーム



閃(ひらめき)さん、  
その格好、  
結構気に入ってる  
ようだね。

ちよつとねえ。  
コスプレまでさせて、  
ゲームショーに潜入捜査って  
よっぽど重要な  
ミッションなんでしょうね！

。。。まあまあ  
詳細はメールを  
送ったので  
見て欲しい。

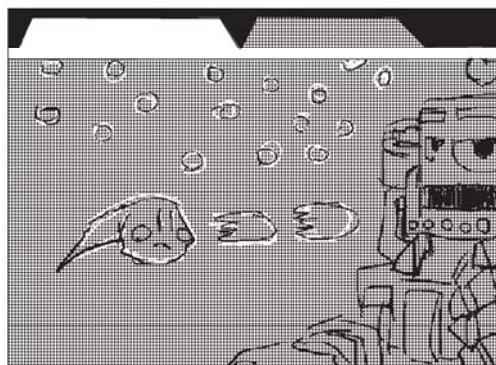
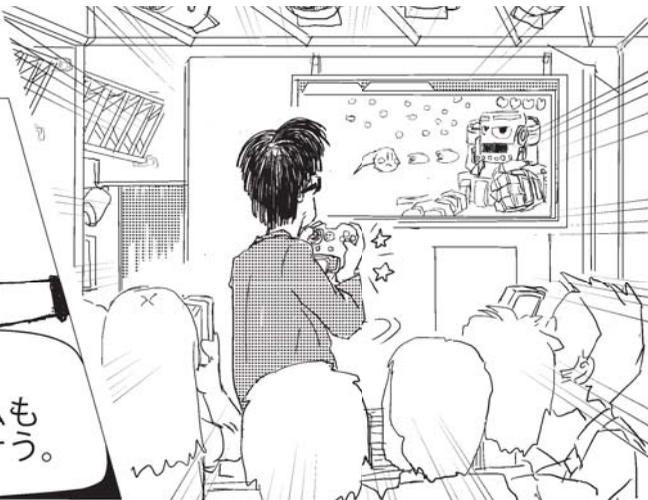
✉ ミッション

FlashAirを  
使った  
新作ゲームの  
秘密を探れ。



そ。  
電子・工員として  
興味が出たでしょ。  
じゃ、よろしく。





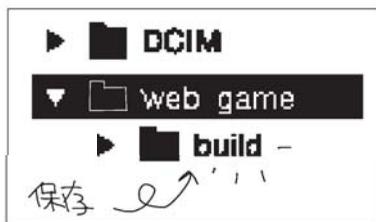
FlashAirはWebサーバです。写真だけでなく、Webで遊べるゲームをFlashAirに保存して、スマホやPCでアクセスすればゲームがでちゃいます。

【解説：FlashAirでゲームをしよう】

1) <http://enchantjs.com/ja> から `enchant.js` をダウンロードします。(この原稿時点ではV0.8.2)

2) 解凍したファイル群から「¥buildフォルダ」をFlashAirの適当なフォルダ(例えば「¥web\_gameフォルダ」)に保存します。

`enchant.js-builds-0.8.2-b.zip`



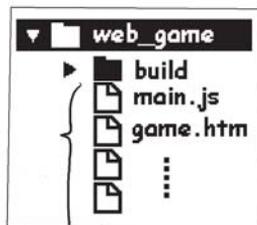
3)次は

"[http://rdstyle.cocolog-nifty.com/flashair/enchant\\_game.zip](http://rdstyle.cocolog-nifty.com/flashair/enchant_game.zip)"をダウンロードします。

4) そのファイルを解凍し、2)と同じフォルダーに保存します。

(例えば「web\_game」フォルダ)

5) 最後はいつものようにFlashAirとPC(またはスマホ)をWiFi接続します。



6) ブラウザを立ち上げてURLを入力します。

(例えば「[http://flashair/web\\_game/game.htm](http://flashair/web_game/game.htm)」

OSXだと「[http://flashair.local/web\\_game/game.htm](http://flashair.local/web_game/game.htm)」)

7) [Enter]でゲームがロードされて、画面が現れます。

(紙面の関係上、ゲームの操作ルールは省略)

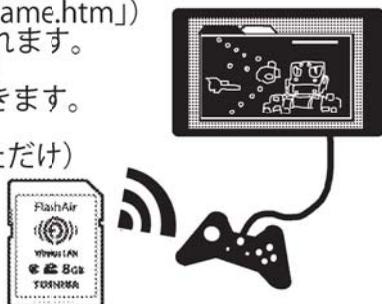
8) PCはカーソル(上下左右)、[z],[x]で操作できます。

また、ゲームパッドにも対応してみました。

(と言ってもXBOX360のパッドのみで試ただけ)

スマホは画面タップです。

9) もし動作しなかった場合、OSやブラウザを変えて試してみてください。

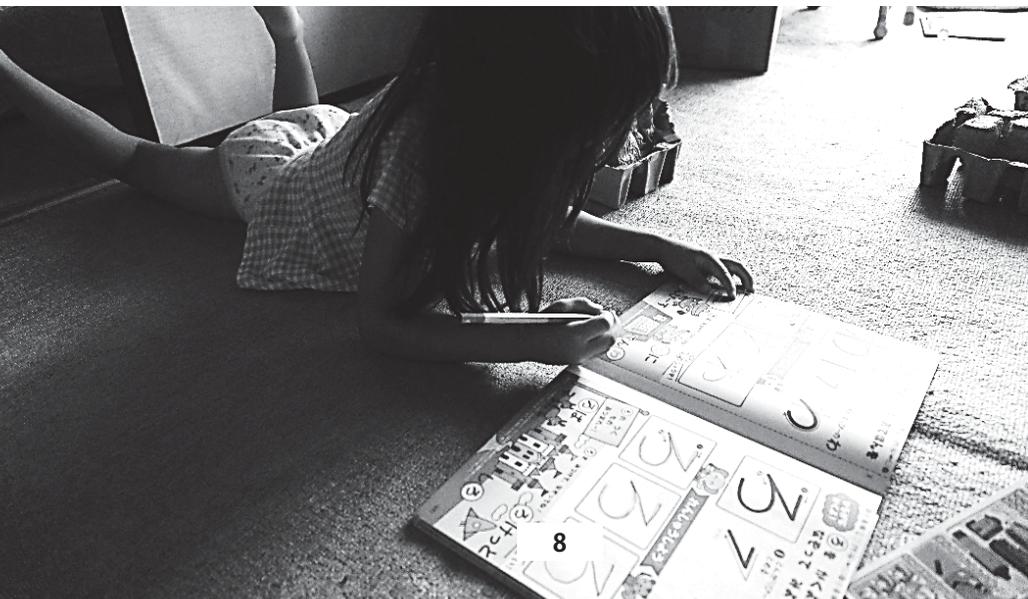


我が家には6歳の娘がいる。今年小学校に入ったばかりだ。毎日、小さな体で大きなランドセルを背負って学校に行っている姿はなんともうれしくなる光景だ。ちなみにランドセルの色が紫色だという事に父親としては時代を感じてしまうのは仕方のないことなのだろう。

娘は小学校に入る前は「勉強したい!」とキラキラした目で言っていた。しかし、実際に入った後は、現実を知ったのか残念ながら「疲れる」と言っている(笑)。ある時、「1年生の勉強はどんな感じなんだろう?」と思って、娘の時間割を見ると8割ぐらゐは国語と算数だった。小学校一年生が勉強するには文字や数字を読めないと始まらないのでそんなものなのかもしれない。それに、宿題もひらがなを書くことだったり、教科書を音読することだったりする。親としてはそんな宿題をしている姿を見られるのはとても楽しい気持ちになる。

ちなみに、そんな娘が通っている小学校では同級生が11人しかいない。最初に聞いた時は何かの間違いかと思ったが、入学式でもその人数だった。もしかしたら、自分が思っているよりも、日本の少子化というのは結構すすんでいると思う。

話を少し戻そう。第3世代のFlashAirには、スクリプト言語を少しだけ読めるようにしてみた。ただし、有名な言語を扱えるほど、内部のRAMが潤沢にあるわけではない。そのため、軽量なスクリプト言語であるLuaという言語を導入した。このLuaという言語はブラジルの大学で開発されたものらしく、Luaとは「月」という意味らしい。なんか名前の付け方がラテン系で面白い。



この言語は、「スクリプトは制御だけやって、演算なんかはC言語でやればいいじゃない。」という割り切った考え方をしている。そのため、スクリプトとC言語の連携が作りやすく、かつメモリ消費量が低いのが特徴である。だからというわけか知らないけれどメモリの小さなゲームシステムにはよく使われていてFlashAirのような小さな物にはちょうど良かった。

ちなみに周りの人に「FlashAirにLuaを入れようと良く思いましたね。」と言われる事がある。理由としては色々それぞれにあるので全部は書ききれぬものじゃない。それでも実際にLuaを入れると決めたまっかけを話すとする、ある日の居酒屋で、「FlashAirにスクリプト言語をいれるってのはどうですかねえ?」と、思いつきをポロツと言ったら、「いいんじゃない!」という返事が返ってきて盛り上がったからだ。

まあ、話が盛り上がったのはビールのせいかもしれないけれど、仕様なんてものはそんな感じで決まっていくのかもしれない。

その翌日から3日ぐらいかけてLuaをFlashAirに移植して、ちょっと動いたから、本気で製品仕様として入れる事にした。

FlashAirの製品としての仕様はWindows95が出たばかりの頃に、インターネットで実現していた事をそのまま順番に真似していると思う。つまり、20年くらい前の大学の授業でHTMLのファイルを書いてWebサーバーに置き、そのファイルをNetscapeのブラウザでアクセスして、「へー」と言っていた時代の話、SDカードという形に変えてやっているということだ。そのため、HTMLを書いて遊んでいた後に、次には何をしていたかなと考えて見ると、Perlで掲示板を作っていたなどというのを思い出した。個人的にはインターネットの初期の時代にコンパイルの必要ないスクリプト言語が果たした役割は大きかったと思う。

発売から3年経って、FlashAirはLuaという言語のスクリプトを読み、少し自分で考えて、インターネットにアクセスできるようになってきた。まるで、小学校1年生がひらがなを読んで、少しずつ世界が広がっていくみたいに。今後も、Luaが果たす役割が増えていくともうちょっと面白い事がおきるのかもしれないと思っている。

※この話はフィクションです。

### 伊藤 晋朗 (@ikainuk)



FlashAirの“中”の人。関係する領域は、無線の電波伝搬、信号処理、WLANスタック、TCP/IP、SDインターフェース、Webアプリケーション層、製品企画および技術営業。主な使用言語はVerilog-RTL、C、HTML、JavaScript、Java、Objective-C、ヒンディー語少々。休日は2児の父を演じている。

2012年3月10日に、初代FlashAirを発売してからはや3年。今年3月には、第3世代「FlashAir W-03」を発売することができました。FlashAirのアイデアが形になる前からの6年間で、イベントを中心に振り返ってまとめました。

## 原始時代：アイデアの発案から開発を始めるまで

2009年4月。社内の若手による、既存の技術を応用して新しい商品を考えるアイデア創出活動にて、無線で写真が共有できるメモ리카ードのアイデアが生まれました。「後で送るから!で、そのままの写真ってありませんか?せっかく撮った写真をより簡単に共有したい。」—これがFlashAirの原点です。

楽しい飲み会で写真を撮っても、あとからその写真をほかの人に送ったりせずに、ついそのままにしがちです。集まったその場ですぐに写真と楽しさが共有できないか?すでに東芝が持っているSDメモ리카ードと無線チップの技術を使って実現できるのではないか?ちょうど海外のパートナー企業から、同様のアイデアで作った試作品を提案されたこともあり、本格的に事業化の検討を始めることになりました。

## 創世記：開発スタートから発売まで

2010年9月。製品開発のキックオフが行われ、正式に開発がスタートしました。これまでに作ったことのない製品ということで、会議室に集められたのは様々な部門の、お互い知らないメンバー達。会議が終わっても、その後の居酒屋（通称：地下会議室）で議論を続ける日々で、飲み会で仕様が決まったと言っても過言ではありません。初めは、カメラとカメラの間で、無線通信で写真データの送受信ができるSDカードを検討していましたが、急速に普及しはじめたスマートフォンに写真を送るカードへ仕様変更したのもこの頃です。

2010年9月	製品開発スタート。 SDアソシエーション <sup>※</sup> にて無線カードの規格標準化に着手。 ※ 略称 SDA。SDカードの規格策定団体
2011年1月	約10カ月かかって商品名が「FlashAir」に決定。名付け親は社内のSDカード標準化担当の重鎮。
2011年3月	東日本大震災発生。工場が被災し、開発が遅延。計画停電によるサーバの停止で、ソフトウェア開発も苦しむ。
2011年6月	カードの色が「白（通称：白芝 <sup>※</sup> ）」に決定。 当初は「パールピンク」の予定だったが、東日本大震災の影響で樹脂の調達ができず、既存の色から選ぶことに。 ※ 過去に発売していた東芝の白いSDHCカードの愛称。性能が良いとマニアに評判の製品。

2011年9月	ドイツ・ベルリンで開催される「IFA 2011」へのデモ出展にあわせ、製品のプレスリリースを実施。「世界初」の記述が物議を醸す。
2011年12月	サンプル出荷を開始。
2012年1月	SDA から「iSDIO Wireless LAN SD 規格」が発表され、米国ラスベガスで開催される CES にデモ出展。
2012年2月	当初2月に予定していた発売を、3/10 に延期。

## 初代 FlashAir 発売

無事3月に発売を開始し、おしゃれな表参道のカフェでプレス発表会まで開催したものの、裏では、発売と同時に提供を始めたソフトウェアのアップデートのために、いきなりサポートセンターがパンクする事態に陥ります。メンバー一同、新製品を市場に出す大変さを痛感したできごとでした。今でも、ソフトウェアのアップデートは週末を避けて提供しています。

2012年3月	初代 FlashAir 発売 & プレス発表会開催。 表参道のカフェ6店舗にて製品体験プロモーション「あなたのスマホに飛んでいく。FlashAirways」を実施。 (プレス発表会でモデルさんが着用した CA 風の衣装がそのまま、応援キャラクター・閃ソラの制服になっています。)
2012年6月	カシオ計算機様から、初の FlashAir 対応カメラ (EXLIM EX-ZR300) が発売。
2012年8月	中国で発売を開始。国内でも中国の無線認証を取得した製品に徐々に切り替え。
2012年10月	キングジム様から、初の「サーバアップロード機能」対応機 (ポメラ DM100) 発表。
2012年12月	Android アプリ「FlashAir」の配信スタート。初めてスマホアプリに参入する。
2013年3月	全国紙3紙に初めて FlashAir 単独で全面広告を掲載。 タワーレコード店頭にて製品体験プロモーションを実施。
2013年4月	iOS アプリ「FlashAir」の配信スタート。
2013年4月	文化放送に開発メンバー2名が出演し、ラジオで製品紹介や開発秘話を語る。



図 1: 初代 FlashAir(SD-WL008G, SD-WB008G)

## 第 2 世代 FlashAir 発売

2013 年 7 月 6 日、“W”ireless の第 “2” 世代の意味で「FlashAir W-02」と名付けた、第 2 世代の FlashAir が発売になりました。ハードウェアとソフトウェアを改良し、無線の通信スピードと、カードへの読み書きスピードを大幅に改善したモデルです。翌月には Web サイト「FlashAir デベロッパーズ」を立ち上げ、技術情報の一般公開を始めました。カードから無線で写真を送信するだけでなく、無線でデータの送受信ができる点を広め、デジタルカメラ以外の機器やサービスでの活用を拡げるねらいです。

しかしなんといっても、第 2 世代で一番のイベントは初のテレビ CM です。広告キャラクターに女優の有村架純さんを起用し、FlashAir を持って、女友達と鎌倉を旅行する様子を CM にしました。

2013 年 7 月	FlashAir W-02 16GB 発売。 キャッチコピーは「撮る ⇒ シェアする」へ。
2013 年 8 月	技術情報サイト「FlashAir デベロッパーズ」オープン。
2013 年 8 月	横浜で開催の「世界ファブラボ会議 (Fab9)」にて FlashAir を使ったインストールーションを展示。
2013 年 9 月	「CEATEC JAPAN 2013」東芝ブースに初出展。
2013 年 10 月	W-02 8GB 発売。 「オープンソースカンファレンス (OSC) 2013 Tokyo/Fall」に初出展。
2013 年 11 月	W-02 32GB 発売。同時に公開されたファームウェアのアップデートで「インターネット同時接続機能」を提供。
2014 年 1 月	シンガポールで開催された国際会議「ASP-DAC 2014」にて、予稿集の配布に初めて FlashAir が採用される。
2014 年 2 月	社内連携・第 1 弾。東芝 PC「dynabook」の写真ソフト「思い出フォトビューア」が FlashAir 対応に。
2014 年 2 月	「OSC 2014 Tokyo/Spring」に出展。マスコットキャラクターのコスプレ大会が行われるということで、急遽、応援キャラ「閃ソラ」を作って参戦。
2014 年 2 月	「CP+ 2014」 SDA ブースに初出展。
2014 年 3 月	ドイツ・ドレスデンで開催された国際会議「Design, Automation & Test in Europe (DATE) 2014」に出展。出展した唯一の日本企業として、日経エレクトロニクスに取材される。
2014 年 3 月	有村架純さんをイメージキャラクターに起用したテレビ CM がオンエア。「有村架純、撮ったらシェアします」と「写真が送れる SD カード」の 2 つのキャッチコピーでプロモーションを展開。
2014 年 5 月	「Maker Faire Bay Area 2014」に FlashAir Developers として初出展。CD の代わりに音楽データを配信する FlashAir をデモ。
2014 年 6 月	FlashAir Make Idea コンテスト開催。最優秀賞はリアルタイムで入退室ログを管理する装置「今北ちゃん」が受賞。
2014 年 10 月	前年に続いて「CEATEC JAPAN 2014」出展
2014 年 11 月	「Maker Faire Tokyo 2014」に初出展。展示会で初めて即売会を実施し、開場 45 分で 50 枚が売り切れ。(注: FlashAir Developers 特製基板キットのプレゼント付きで販売。)
2014 年 12 月	社内連携・第 2 弾。「スマホでサイネージ」発売開始。
2015 年 2 月	東芝として初めて「CP+ 2015」に出展。FlashAir だけでなく、全ストレージ製品をアピール。



図 2: 第 2 世代 FlashAir (SD-WC008G)

### 第 3 世代 FlashAir 発売

今年 3 月 14 日からは、汎用スクリプト言語の Lua を用いてプログラムに従って動作させられるようになった、第 3 世代の「FlashAir W-03」を発売しています。ようやく、IoT 用途にも使っていただける製品になりました。

はじめはたった数人で始まった FlashAir ですが、もっともっとファンが増えるようなプロダクトに育てていきたいと思っています。今後の展開にもどうぞ期待ください。

2015 年 3 月	FlashAir W-03 発売開始。8GB/16GB/32GB を同時に発売。
2015 年 3 月	スペイン・バルセロナで開催された「Mobile World Congress (MWC) 2015」SDA/TransferJet コンソーシアム共同ブースに初出展。
2015 年 4 月	有村さんの新テレビ CM がオンエア。 「かすみドールのシェア旅」と題してプロモーションを展開。
2015 年 4 月	FlashAir アイデアソン・ハッカソンを初めて実施。3 日間にわたり、100 名近い参加者のエネルギーに圧倒される。
2015 年 5 月	朝日放送「ABC ハッカソン」に API を提供。
2015 年 6 月	インターネットラジオ「Julie Watai HARDWARE GIRLS RADIO」にメンバー 2 名が出演し、開発秘話と GPIO について語る。
2015 年 6 月	公式サイト*が全面的にリニューアル。 ※ 新 URL : <a href="http://www.toshiba-personalstorage.net/">http://www.toshiba-personalstorage.net/</a>
2015 年 8 月	前年に続いて「Maker Faire Tokyo 2015」に出展。



図 3: 第 3 世代 FlashAir (SD-WE008G)



### 笠原 (@sayachocolat)

プロジェクト発足当時からのメンバーの1人です。ビジネスの立ち上げを経て、FlashAirの渉外担当をしています。

# Maker Faire Tokyo 初出展と FlashAir ハッカソン

Pochio

その日の朝 10 時頃、ゆりかもめの車中でやや焦りながらツイートしていたのを今でも覚えている。思えばこの日が FlashAir チームの活動の転機だった。そこで今回も前号に引き続き形で、昨年 11 月の Maker Faire Tokyo 2014 から今日までの活動経緯をつらつらとつづってみようと思う。

## Maker Faire Tokyo 初出展

2014 年 11 月 23 日は 2 日間開催される Maker Faire Tokyo (MFT) 2014 の初日。前日は搬入日だったが、いきなり資本力の差を痛感した。長机 2 本を並べていつものテーブルクロスを広げた我々のブースの隣は、コーポレートカラーがブルーの、あの世界最大の半導体メーカーだ。広さはもちろん展示規模でも圧倒的な差を見せつけられていたが、「ウチはウチならではのやり方で」と自分に言い聞かせて準備を続けた。しかし一抹の不安は隠せない。なぜなら FlashAir のイメージカラーも、隣と同じブルーでかぶっているからだ。

気がつけば隣は展示ブースだけでなく、ワークショップのための小さなスペースまでも用意していた。ふと、ちょっと前に流れた「Edison の工作キットを先着 100 名に無料配布する」というニュースを思い出す。ここで製作体験をしてもらおう、ということなのだろう。

かたやこちらは販売用の FlashAir を 50 個用意していた。MFT での販売は、我々がこれまで訴求し続けてきた「FlashAir のデジタルカメラ以外での活用」というアプローチが、売り上げに結びつくかどうかを問う試金石なのだ。会場での販売価格は市価とほぼ同じだが、じむさんが作った FlashAir の GPIO 利用のための基板とその部品のセット(図 1)を、MFT 特典としてつけることにした。



図 1: MFT 特典のソラちゃん基板



図 2: FlashAir 同人誌

2日間で全 50 個を売り切りたいという思いから、ゆりかもめ乗車中にツイッターで FlashAir の会場販売を宣伝し続けた。関係者が同じようにツイッターで宣伝しているのを確認したころ、ゆりかもめは国際展示場正門駅についた。

ビッグサイトはまだ静かだった。しかし MFT の会場入口にいくと、Edison の無料配布を狙う人たちが大行列を作っており、あわよくば自分も並ぼうと考えていた浅知恵を恥じた。

ところで今回は FlashAir を使った様々な作例の展示に加えて、余熱さんのディレクションのもとで関係各位が執筆した「FlashAir 同人誌」(図 2) を配布することになっていた。これは 2014 年秋のオープンソースカンファレンス (OSC) 東京で出会った彼が、今回から我々の活動に協力してくれることになり、彼が自ら持ち込んだ企画であった。同人誌はフルカラー 32 ページで、これまでの活動経緯だけでなく、FlashAir を SPI Master にして液晶に画像を表示させてみた記事や、じむさん直筆の Raspberry Pi での活用方法を解説したマンガなど、読み物とマニアックな記事で充実を図った。一方でデジカメ用途の話は一切載せなかったのは、我々のポリシーだった。

そして迎えた MFT の開場時間、私は予想もしない光景を目にした。なんとツイッターを見た人たちがこぞって FlashAir を買いに来てくださり、販売ブースに行列ができたのだ(図 3)。50 個用意したセットは 45 分で完売し、その後も購入希望者が後を絶たなかった。さらに同人誌も、準備した 1000 部のうち半数以上を初日に配りきってしまった。その反響たるや我々の予想を超えており、ツイッターのみならず複数の web ニュースでも取り上げて頂いた。メーカーの中の人や同人誌という形で情報提供することは、多くの人の目に新鮮に映ったようだ。

もちろんブースにお越し頂いたお客さまには、FlashAir の GPIO 機能を使った事例をお楽しみ頂いた。特にじむさんが作ったワイヤレスクレーンゲームは好評で、たくさんの子供たちが挑戦しては涙を飲んだ。

またセミナー会場ではフィクスターズの土居さんが、OSC でおなじみの FlashAir セミナーを開催し、Make 的な観点で FlashAir の活用方法をご紹介くださった。こちらには 100 人以上にご参加頂き、立ち見ができるほど盛況だった(図 4)。



図 3: 大行列ができた FlashAir ブース



図 4: 立ち見が出たセミナーの様子

終わってみれば当初の予想を超える反響だった FlashAir のブース。ツイッターでは「Edison のライト層を完全に食った」とのコメントもあり、好評だったと記憶している。しかもこの Maker Faire では GUGEN さんとの出会いという、今後の活動にとって重要なきっかけも生まれていた。

### FlashAir ハッカソンの開催

MFT を終えて年が明けると、FlashAir は最新版である W-03 の発売を控え、我々は Maker への今後のアピールの仕方について模索していた。実は年明け (2015 年) 早々に、ラスベガスで開催される世界最大の家電見本市「Consumer Electronics Show (CES)」で W-03 が展示されると聞いていた。

W-03 の目玉機能は、何と言っても FlashAir で Lua (スクリプト言語) が動作することだ。これまでたくさんの人から、FlashAir でプログラムを走らせたいとか、写真を自動的に転送したいといったリクエストを聞いていたので、Lua が動作するようになった FlashAir は必ず注目してもらえると確信していた。ところが CES の初日に米国現地法人から出た W-03 のニュースリリースには、Lua の機能についての言及が全くなかった。

そのためしばらくこの件は黙秘をつらぬくしかなかったが、あるメディアの記者が CES の会場で Lua の機能について担当から聞いたらしく、ネット上でその情報が流れはじめた。敏感なツイッターユーザーが Lua の機能に気づいてツイートしはじめ、反響がじわじわ出はじめたのにも関わらず、我々は文字通り「指をくわえて眺めている」しかなかった。そこで何とかして Lua の機能をアピールし、新型 FlashAir の活用方法を見出したいという思いから、ハッカソンの開催にエンジンがかかった。

ハッカソンとは、ハック (hack) とマラソン (marathon) からの造語で、元々はプログラムが限られた時間内に共同でソフトウェアを開発するイベントだったが、最近はハードウェアのプロトタイプ開発にもこの手法が用いられているようだ。そこで FlashAir のハッカソンを開催して Lua の機能を知ってもらい、それを活用したプロダクトを考えてもらうことにした。



図 5: GUGEN ECO 開発システム

さて、我々には当然ハッカソンに関するノウハウはないので、そこに GUGEN さんの登場である。MFT では我々のブースの真裏が、プリント基板のネット通販で知られる P 板 .com さんのブースだった。GUGEN さんは P 板 .com の社内ベンチャーで、様々なハッカソンを手がけている。我々は GUGEN さんにハッカソンの開催についてご相談したところ、GUGEN の代表である崔さんが大変面白いシステムをご提案くださった。それは「GUGEN エコ開発システム」(図 5) と名付けられ、リーンスタートアップとよばれる開発手法を用いて、ハッカソンで生まれたプロトタイプを洗練させた後、最近流行のクラウドファンディングに挑戦し、ファンディングに成功したらネットで販売するというものだった。

この崔さんの提案したシステムは画期的だった。ハッカソンでは参加者がアイデアを出してプロトタイプを作るが、最終的には主催した企業がそのアイデアの一部を買取るか、もしくはアイデアのまま終わってしまい製品につながるケースはまれだった。しかし崔さんの提案したシステムは、参加者に自分たちのプロダクトを販売できる可能性を残し、これを企業の協力を得て出来る範囲でサポートするというものだった。

ハードウェア開発にとってハッカソンが有効かどうかを効果測定するには、そのプロトタイプが最終的に世に出たかどうかで判断するのが最もわかりやすいし、これがうまくいくと多くの企業がこの手法を採用しはじめるだろう。そういう意味で FlashAir ハッカソンは図らずも、メーカーにとってハッカソンが有効な開発手法かどうかを問う試金石となってしまった。実際、FlashAir ハッカソンの告知をした後に、他の家電メーカーの方が「FlashAir のハッカソンがうまくいかなかったら、今後どのメーカーもハッカソンをやらなくなるでしょうね」と言っていた。

こうして FlashAir ハッカソンは 2015 年 4 月 11 日、18 日、19 日の合計 3 日間で開催することになった。初日は GUGEN さんの用意した方法でアイデアを出し合う「アイデアソン」を行い、参加者同士の投票で得票数の多かったアイデアを中心にチームを編成する。そして残りの 2 日間でチームごとにプロトタイプ開発、すなわち「ハッカソン」を行うという段取りだ。

参加費が 3000 円だったにも関わらず、アイデアソン当日は 90 人以上の方にご参加頂き、大変活気のあるイベントになった(図 6)。この日の午前中に私が FlashAir のワークショップを行ったのだが、会場の Wi-Fi 環境がよろしくなく、PC から FlashAir への接続に苦勞するトラブルに見舞われた。仕方なく場つなぎのしゃべりで取り繕ったが、この



図 6: FlashAir ハッカソン初日の様子

1 アメリカのシリコンバレーで生まれた起業のための手法で、仮説検証のための必要最小限のプロダクトを作り、それを使ったユーザーテストのフィードバックに基づいて製品の改善を繰り返すという手法。ユーザーニーズの訴求に有効であると注目されている。

ときの「しゃべり」が後に「FlashAir 芸人」になるきっかけになるとは思ってもみなかった。

初日のアイデアソンで12チームが編成され、ハッカソンを経て12個の興味深いプロトタイプが出来上がった。これらの作品は22ページから紹介しているのので、ぜひご覧きたい。優勝したのは「NEON」のみなさん(図7)で、作品はFlashAirで互いに同期を取りながら光る、ジャグリング用のデバイスだった(図8)。

ハッカソンによってFlashAirの組み込み活用事例がこれほど生まれたのは、ご参加頂いた方々の熱意と創作意欲によるところが大きい。アイデアや技術力のみならずプレゼンテーション能力も必要なハッカソンは、本当に根気の要る大変な作業で、まさに知力と技術力のマラソンといった印象だった。走りきってさぞくたびれたかと思いきや、懇親会で「とても楽しかった」とか、「次はもっとうまくやれると思うのでまた開催して欲しい」という声を聞き、ご参加頂いた方々(図9)の知的バイタリティに感服する次第であった。

なお、FlashAirハッカソンで生まれたいくつかのプロトタイプは、現在GUGENさんのサポートでリスタートアップのプロセスに入っている。この中からクラウドファンディングに成功する作品が出て欲しいと心から願っているし、そのために我々も各チームのお手伝いを続けているところである。



図7: 優勝した NEON の皆さん



図8: 光るジャグリングデバイス



図9: FlashAir ハッカソンにご参加頂いた皆様

## ABC ハッカソンの収録と FlashAir 芸人

FlashAir ハッカソンでお世話になった GUGEN の崔さんから、今度大阪の朝日放送が「ABC ハッカソン」という番組を制作するので、FlashAir を API として番組に協賛してはどうか、とお声がけ頂いた。FlashAir チームの隊長である高田さんから、「FlashAir ワークショップでの(私の)しゃべりがテレビ向きだったから、ABC ハッカソンに行って欲しい」と要請があり、5月9日から11日まで出張することになった。

この番組は初日にアイデアソンを行い、約20チームの中から7チームだけがハッカソンに進むことができるという厳しいものだった。ところが厳しいのは参加者だけの話ではなく、API 協賛各社にとっても同じであり、本戦に進むチームに FlashAir を採用してもらえなければ、残り2日間我々の仕事がなくなってしまうのだ。参加者の中には FlashAir ハッカソンにもご参加頂いた方がいたので、FlashAir を採用してもらえるよう各チームに宣伝してまわった。その甲斐あってかハッカソン進出7チームのうち、「チーム共犯者」の皆さんに FlashAir を採用して頂いた。

オレンジの T シャツを着たチーム共犯者のみなさんは、「FLEURIR (フルリール)」と名づけた花形のかわいいシュシュを作ろうとしていた(図10)。関西コレクションなどのイベントで来場者がこのシュシュを身につけていると、会場がカラフルな花で彩られるだけでなく、本人のおしゃれの好みに応じてシュシュの花びらの色が変わるガジェットだ。色の変化の制御は、ガジェットに組み込まれた FlashAir とクラウドとの通信で実現しようとしていた。

チーム共犯者の皆さんは、実際に関西コレクションでモデルをした方や一般の方から熱心にヒアリングしたり、FLEURIR に要求されるニーズについて真剣に悩みながら、アイデアの実現に向けて邁進していた。ところがハッカソン最終日の朝にスタジオを訪れると、プロトタイプ作成中に FlashAir が思うように動作しなかったため、FlashAir の採用を保留して代わりに Bluetooth デバイスを試しているところだった(汗)。この事態に同行した寺西さんと私はかなり焦った。ハッカソンの最終プレゼンを目前にして FlashAir を採用した作品が消えたなどと、会社に戻っ

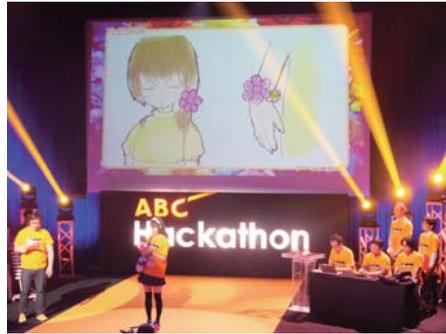


図 10: チーム共犯者の皆さん



図 11: シュシュのプロトタイプ

てから言えるはずがない。そこから我々の必死のサポートがはじまった。なんとかチーム共犯者の皆さんの目的をFlashAirで実現できる目処が立ち、制限時間ぎりぎりのところでプロトタイプ(図11)が完成した。作品はチームの皆さんの努力の甲斐あって、審査の結果サイボウズさんから賞(図12)を頂くことができた。



図12: 祝サイボウズ賞受賞

このABCハッカソンは現在でもYoutubeで見ることができるので、興味のある方はぜひご覧頂きたい。限られた時間の中で悩みながらハックし続ける参加者の様子が、ドキュメンタリー形式で描かれている。

ところでなぜFlashAir 芸人などと言いついたかということ、GUGENの崔さんと共にABCハッカソンの進行を務められたハッカソン芸人のハブチンさんに初めてお会いした際、「じゃあ僕はFlashAir 芸人です」と勢いで言ってしまったのがはじまりだったりする。

## おわりに

FlashAirにとって2回目のMaker Faire Tokyoを迎えるにあたり、同人誌2号を作るようになったので、前回に引き続きFlashAirチームの活動の顛末をつづってみた。この間、FlashAirハッカソンの開催やABCハッカソンへの参加だけでなく、Julie Wataiさんの「HARDWARE GIRLS RADIO」にお伺いしたり、fabcrossさんに取材頂いたり、活動の幅が広がってきた。関係者の皆様には本活動へのご協力に心から感謝申し上げる。

今後も引き続きFlashAir(芸人?)の活動を続けたいと考えているので、ぜひFlashAirに対するご意見および同人誌のご感想を、ハッシュタグ「#FlashAir」でツイッターにてつぶやいてくださると幸いです。



### Pochio (@l\_love\_nintendo)

FlashAirのデジカメ以外の用途を模索し続ける言いだしっぺ。本業は半導体の微細なパターン形成技術の開発だが、しゃべりが災いして最近(自称)FlashAir芸人に。ときどき論文書いたり学会発表したりしている。IEEEシニアメンバー、電子情報通信学会シニア会員、博士(工学)。

# FlashAir の新たな可能性をハッカソンから探る！

GUGEN 崔

FlashAir を使って新しい IoT 製品を生み出す「東芝 FlashAir ハッカソン」が、4 月 11 日、18 日、19 日の 3 日間、神奈川県川崎市の東芝スマートコミュニティセンターにて開催されました。

本イベントは GUGEN が企画・運営を担当し、主催の東芝をはじめ、クラウドサービス Microsoft Azure を提供する「日本マイクロソフト」、電子工作用の電子部品を販売する「スイッチサイエンス」、クラウドファンディングサービス「Makuake」、ネット通販の「Amazon」、「BASE」、FlashAir 技術サポートを行う「Fixstars」と、それぞれ得意分野を持つ企業が協賛。手を動かしながら新しいモノを発見する「アイデアソン・ハッカソン」を出発点とし、マーケットへのリリース、販売支援までフォローする「新しいハッカソンプロジェクト」として業界の注目を浴びました。

総勢 90 名のものづくりに燃える熱いクリエイター達が参加し、最終的に 12 チームから 12 のアイデアおよび試作品が生まれました。

## 参加者属性

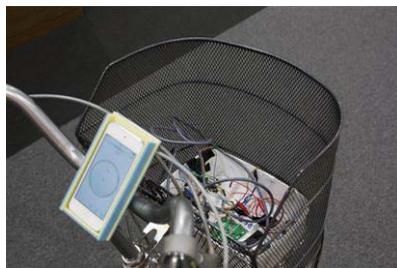
- ・企画・プランナー：21 名 (23%)
- ・ハードウェアエンジニア：20 名 (22%)
- ・ソフトウェアエンジニア：42 名 (47%)
- ・デザイナー：7 名 (8%)

2015 年 7 月現在、12 チームの内、5 チームが商品化に向けて開発活動を継続しています。BtoC 向けのプロジェクトは年内のクラウドファンディングリリースを目標に、BtoB 向けのプロジェクトはビジネスパートナーの検討など、活動を進めています。

それでは、「東芝 FlashAir ハッカソン」で生まれた 12 チーム [プロジェクト] をすべて紹介します！

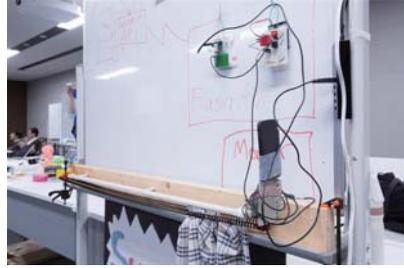
## Dyamon [Dynamon]

自転車を停めた場所を特定できるデバイスです。FlashAir を組み込んだデバイスを自転車のダイナモ部分に取り付け、FlashAir の ON/OFF と自転車の動作を関連付けます。FlashAir の起動・停止ログを取得することで、自転車の位置を特定することができます。



### チーム朝寝坊 [スマートカーテン]

コンセプトは「日の光を浴びて、気持ちの良い朝を迎える」。FlashAirを使ったスマートカーテンで、日の出や天候データをクラウド経由で取得し、適切なタイミングで自動で開きます。



### FlashCast [FlashCast]

FlashAir のデータ保存、通信といった機能を使用することで、結婚式などイベントで撮影した写真をそのイベント中に共有できるデバイスサービスです。撮影した写真を自動でクラウドに保存し、スライドなどにも映すことができます。



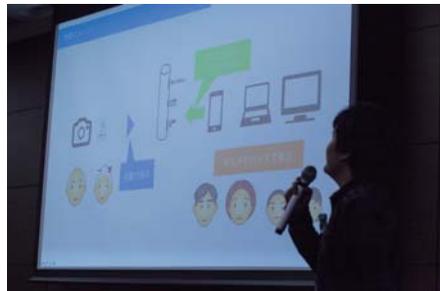
### NEON [スマートジャグリング]

FlashAir 間で通信を行うことで、互いに同期を取りながらパターン発光するデバイスです。用途はジャグリング機器を想定。複数個を演技や音楽と同期させながら、自動でパターン発光させることができます。



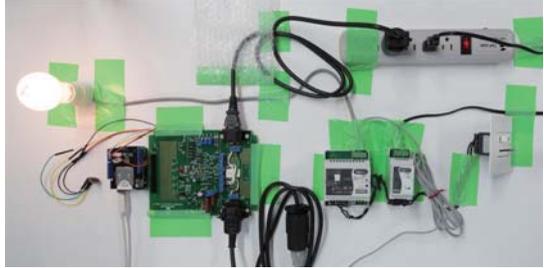
### ココヰコ [ココヰコ]

写真を撮影するだけで簡単に写真を共有できるデバイスサービスです。グループ登録しておけば、新しい写真が保存されると通知されます。FlashAir とクラウドをひもづける必要はありますが、専門知識は必要なく、誰でも簡単に使用することができます。



### 見守りスマートコンセント [スマートライト]

照明器具などの使用状況を知ること、離れた家族の安否を確認できるデバイスサービスです。既存の製品にSDカードを取り付け、FlashAirを使ってデータを取得します。



### nyaccessory [nyaccessory]

猫のプロフィールを交換・確認することができるアクセサリ型デバイスです。内蔵のFlashAirに飼い猫の情報を記録し、猫用アクセサリとして猫に付けます。Wi-Fiで、専用アプリからデバイスにアクセスするだけで猫情報が確認できます。



### Air Communication [Air Communication]

イベント会場のような限定された空間で話すきっかけを提供するデバイスサービス。FlashAirにプロフィールを入力しておき、互いに専用アプリを使ってアクセスすることで、コミュニケーションのきっかけを手に入れることができます。



### Flash\_P\_Air [Flash\_P\_Air]

SDカードに保存したデータを人に渡す際に発生する一手間を解消します。2枚のFlashAirをひもづけることで、片方のFlashAirに保存するデータを、もう一方のFlashAirにも自動で保存します。



### 長崎ちゃんぽん [五分前の世界]

「数分前の記憶が思い出せない」、そんな状況を解消するためのデバイスです。ヘッドセット型のデバイスを使って会話を録音し、FlashAirを経てクラウドへ転送します。転送後、録音データをテキスト化し、スマホなどへ配信することで過去の記憶を補完することができます。



### YAMAMORI [YAMAMORI]

遭難救助を手助けするデバイスです。FlashAirをONにしたときに発信する信号を端末で受信し、位置を特定します。山の中で通信ができなかったり、電源がすぐに切れてしまう可能性が高い携帯電話に比べ、より長時間、信号を発信することができます。



### 運ぶ根 [運ぶ根]

海外から日本に来た観光客向けのデバイスサービスです。FlashAirを使って地域特化型の情報配信を行います。特定の場所、例えばバス停やゲストハウスに地域の情報を登録した本デバイスを設置し、専用アプリでアクセスすることで、必要な情報を取得することができます。



### GUGEN 崔 (@GugenTheIdea)

GUGEN は、実用性や商品性の高いアイデアを表彰し、その具現化をサポートするものづくりプログラムです。GUGEN は、ものづくりを行う人たちに向けて、アイデアのブラッシュアップやプロトタイプづくりを応援する取り組みを行っています。

# FlashAir との出会い

Julie Watai

フォトグラファーをやってます Julie Watai と申します。

私は FlashAir を第一世代から愛用しているファンの一人でもあります。写真作品では赤外線リモコンを使ったセルフポートレート撮影をする機会が多いのですが、FlashAir を使い始めてからかなり効率的に撮影出来るようになりました。

例えばスタジオでセルフポートレート撮影する場合。有線でパソコン画面をモニターしながらのテザー撮影が出来るので特に問題は無いのですが、屋外や店舗内でのロケ撮影の場合はなかなかこういった余裕のある環境は確保出来ません。

人通りのある場所でパソコンを置きっぱなしにするのは転倒や管理の面で不安ですし、でも手元でモニターしながら確実に取りこぼしのないように撮影がしたい…

そんな時に色々ネットで調べてたどり着いたのが FlashAir (第一世代) です。

早速デジタル一眼レフ (EOS 5D-Mark2) にカード変換アダプターを取り付けて使いました。一眼レフで撮った画像を手元の iPad で確認しながら撮影が出来て、他の無線 LAN 付き SD カードに比べるとカメラ側の電池の減りも遅いため私の用途にまさにピッタリでした。

そして FlashAir を使って撮影をはじめてから気づいた事もあります。撮った写真を iPad のブラウザの一覧から選んで確認していくと、パソコンの画面で写真を選ぶ時とはまた違った感覚があります。おそらく紙に印刷した写真を手に取って選ぶ感覚に近いのだと思います。大量に写真を撮ったあとは選別に迷いが生じるため時間がかかるものですが、この方法でかなりの時間短縮に繋がりました。

作品は sns にもアップするため、スマホ側からもアクセスして一覧が見れるのも大きいです。見るのは撮ってすぐの加工前のデータですが、小さい画面にも映える画像を加工前の段階で手軽に選別出来ると完成系のイメージがより掴みやすいです。

最新型の FlashAir では様々なアプリも出ていますし、自分にとって快適な環境を整えていく楽しみもあります。今後は高性能なガジェットとしても色々試してみたいと考えています。



### Julie Watai (フォトグラファー) (@JulieWatai)

24歳で単身イタリアに渡り、イタリアの美術書籍出版社 DRAGO&ARTS からフォトグラファーとして「SAMURAI GIRL」(2006年)を出版。ニューヨークの近代美術館(MoMA)に置かれるなど、世界的に販売された。帰国後もフォトグラファーとしてはもちろん、DJ、歌手、iPhoneアプリ開発など多岐に渡って活動をしている。多くのガジェットやロボットを作品に登場させており、昨年発表された、アンドロイド「ASUNA」とのコラボ作品が話題を呼ぶ。2015年6月に開催された Maker Faire Shenzhen 2015 に招待され、その功績は評価が高い。

# FlashAir Developers へようこそ

土居

こんにちは。この記事では、FlashAir 開発者情報サイト「FlashAir Developers」(図 1) の使い方を、ご紹介していきたいと思います。本記事を読み終わる頃には、ますます FlashAir Developers を活用できるようになっていることでしょう。ぜひ、FlashAir Developers のページをめくりながらご覧ください。



図 1: <https://flashair-developers.com>

## FlashAir Developers とは

そもそも FlashAir 開発者情報サイトとは何でしょう。

ご存じのとおり、FlashAir はフラッシュメモリに加えて、無線 LAN、そして Web サーバーまで SD カードのなかに搭載してしまった、大変ユニークな製品です。しかし、FlashAir のもっともユニークな点は何とんでも、これらの機能を利用する API を一般公開している部分にあります。API を使えば、無線 LAN の ON/OFF や、インターネットのデータ送受信などを、FlashAir に実行させることができます。さらに、Lua スクリプトインタプリタを搭載しており、FlashAir 上の CPU を使って簡単な制御や計算までできてしまいます。

FlashAir が備える機能は、ほとんどが標準規格やオープンテクノロジーのため、Web 上を探せば仕様書は集められます (表 1)。

しかし、情報が公開されていても、仕様書だけでは、なかなか開発のハードルが高いというのが事実です。そこで、FlashAir Developers の登場です。仕様書へのリンクの集約、独自仕様の公開、サンプルの公開、そしてこれらの機能を使って開発された世界中の情報を集約する、という役割を FlashAir Developers が担っているのです。

表 1: FlashAir の機能と規格

機能	規格	規格団体
フラッシュメモリ	SD	SD Association
web サーバ	HTTP, HTML, JavaScript	W3C
無線 LAN 制御	Intelligent SDIO (iSDIO) Wireless LAN Addendum	SD Association
Lua スクリプトインタプリタ	(オープンソース)	Lua.org

## FlashAir Developers のコンテンツ早わかり

FlashAir Developers にはどんなコンテンツがあるのか、具体的にご紹介しましょう。カッコ内の青字は、そのコンテンツがサイト中のどのページにあるかを示しています。

### 1. FlashAir の仕様やサンプルコードなどの情報公開

FlashAir の API 仕様や、基本的な使い方をチュートリアルなどで紹介しています。FlashAir Developers が作成して、開発者の方々向けに公開しているものです。

- FlashAir API 仕様 ([API ガイド](#))
  - CGI ([command.cgi](#), [config.cgi](#), [thumbnail.cgi](#), [upload.cgi](#))
  - HTML カスタマイズ ([ブラウザユーティリティ](#))
  - FlashAir の設定変更 ([CONFIG](#))
  - Lua スクリプト機能 ([Lua 機能](#))
  - ホスト機器からの FlashAir 制御 ([iSDIO](#))
- FlashAir の機能を利用した開発サンプル ([チュートリアル](#))
  - FlashAir に CGI でアクセスする iOS アプリの開発 ([iOS 向けチュートリアル](#))
  - FlashAir に CGI でアクセスする Android アプリの開発 ([Android 向けチュートリアル](#))
  - ブラウザユーティリティのカスタマイズ ([ウェブブラウザ向けチュートリアル](#))
  - 上級者向け情報 ([上級者向けチュートリアル](#))  
FlashAir の機能をカスタマイズする方法をチュートリアル形式で紹介しています。
  - iSDIO を使った組み込み機器からの無線 LAN 制御 ([Arduino 向けチュートリアル](#))
  - Lua スクリプトを使ったファイルのアップロード、ダウンロード ([Lua 機能](#))
- 開発上のよくある質問 ([FAQ](#))
- ニュースの発信 ([Twitter](#), [Facebook](#))
- そのほか、イベント情報など

### 2. コミュニティによるコンテンツの集約

FlashAir 開発者コミュニティ、つまり FlashAir Developers を利用する企業や一般の開発者の方々が作成された、アプリやツール、応用事例、研究ブログなどの情報を集約しています。開発者同士の Q&A で作るフォーラムも、コミュニティによるコンテンツのひとつです。FlashAir を使って何か作られたときには、FlashAir Developers までお知らせください。

- コミュニティの方による開発チュートリアル ([ユーザーズチュートリアル](#))
- FlashAir 対応アプリの紹介 ([アプリショーケース](#))
- FlashAir 組み込み事例の紹介 ([応用例](#))
- 開発に役立つライブラリの紹介 ([開発リソース](#))
- 開発者の相互協力による Q&A ([フォーラム](#))

### 目的別 FlashAir Developers おすすめコース

---

FlashAir Developers のどの記事を読めばいいか、目的別におすすめコースを紹介してみます。

#### FlashAir と連携するアプリを作りたい

##### 1. アプリショーケース

まずはショーケースを眺めて類似の機能のアプリがないか探してみるとよいでしょう。どんなことができるのか参考になります。

##### 2. iOS または Android 向けチュートリアル

このチュートリアルを一通りなれば、基本的な機能は理解できるでしょう。

##### 3. API ガイドの CGI リファレンス

チュートリアルに書かれていない機能はリファレンスで調べましょう。

##### 4. 開発リソース

コミュニティの方が作った、ライブラリ、スクリプト、研究事例などを紹介しています。

##### 5. FAQ、フォーラム

わからないことがあったらまずは FAQ を確認しましょう。解決しない場合は、フォーラムに投稿してみましょう。

#### 組み込み機器に使いたい

##### 1. 応用事例

まずは応用事例を見てどんなことができるのか確認しながら、構想を練りましょう。

##### 2. ユーザーズチュートリアル

コミュニティの方が作られた組み込み機器応用の具体例が紹介されています。

### 3. Lua 機能チュートリアル、Arduino 向けチュートリアル

FlashAir カード自体で完結したシステムを考えている場合は Lua スクリプトを使うとよいでしょう。処理の規模が大きくなってくるとホスト機器から無線 LAN や HTTP 通信を利用できる iSDIO 機能がおすすめです。Arduino 向けチュートリアルが参考になるでしょう。

### 4. API ガイドの Lua 機能、iSDIO リファレンス

チュートリアルに書かれていない機能はリファレンスで調べましょう。

### 5. 開発リソース

コミュニティの方が作った、ライブラリ、スクリプト、研究事例などを紹介しています。

### 6. FAQ、フォーラム

わからないことがあったらまずは FAQ を確認しましょう。

解決しない場合は、フォーラムに投稿してみましょう。

## ビジネスで応用できそうか調べたい

### 1. 応用事例、ユーザーズチュートリアル

応用事例を見て目的に近い事例がないか確認してみましょう。

### 2. お問い合わせフォーム

### 3. 開発を行う際に不明な点があれば、お問い合わせください。

## 開発はしないがデジカメに入れて便利に使いたい

東芝の FlashAir 公式サイトへ行きましょう。

## I speak English.

Scroll to the bottom and click “English” to go to English site.

## おわりに

私たちは FlashAir Developers という言葉に、2 つの意味を持たせています。ひとつは、この開発者情報サイトの名前。もうひとつは、FlashAir に関連した何かを作る開発者の方々とそのコミュニティです。2013 年 8 月 1 日にサイトがオープンしたときはまだ影も形もありませんでしたが、2 年を経てコミュニティとしての FlashAir Developers ができてきた実感があります。ここまで来られたのは、この同人誌 2 号に寄稿されている皆さんのご尽力によるものです。そしてここからは、いまこれを読んでいるあなたに FlashAir Developers の一員になっていただけることを願っています。



### 土居 (@munepi)

FlashAir Developers の発足当初から記事執筆と運営を担当しております、株式会社フィックスターズ 土居意弘です。いまはコミュニティの方々の素晴らしい作品を紹介するのが喜びです。サイト運営チーム、蓮見、生田にも感謝！

FlashAir ハッカソンの会場で、参加者のサポートをするメンターとして参加しました。

ハッカソンの参加者からの質問で、皆が開発する上で知っておいた方が良さと思う内容を、ホワイトボードで共有していました。それが、図1の「FAQ (メンターに聞こう!)」です。本章では、その内容を同人誌の読者と共有したいと思います。また、個人的に追加したら良いと思った情報もあわせて説明します。

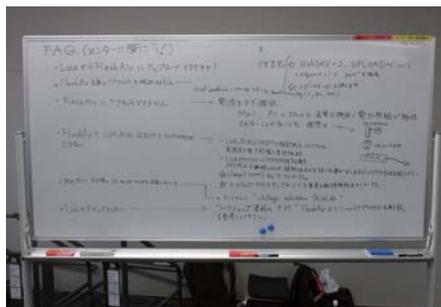


図 1: ハッカソン会場に置かれたホワイトボード「FAQ (メンターに聞こう!)」

読者の皆様に、巷で言われている変態 SD メモリカード「FlashAir W-03」で楽しく遊んで頂きたいと思います。それでは、レッツ・ハック!

## FlashAir にアクセスできないのですが?

PC や Mac の SD カードスロットに FlashAir を挿すと、PC 側の省電力機能により FlashAir に電源供給されなくなります。そのため、Airio や市販の USB 充電器に、SD カードリーダーライターを挿して開発することを推奨しています。

## Mac で FlashAir の /SD\_WLAN が参照できないのですが?

/SD\_WLAN は隠しフォルダとなっています。そのため、Mac の「ターミナル」を起動し、「chflags nohidden SD\_WLAN」と入力してください。

## LUA\_RUN\_SCRIPT で指定した Lua スクリプトで HTTP アクセスできないのですが?

FlashAir に電源投入してから 5 秒後に、LUA\_RUN\_SCRIPT に指定した Lua スクリプトが動きます。無線 LAN が接続する時間も必要なため、それまでスクリプト処理が行われないようにウェイトを入れてください。その場合、sleep(msec) を使用してください。

FlashAir Developers のサイトのサンプルプログラムに、無線 LAN の接続を待つ関数があります。そちらも参考にしてください。

1 FlashAir 評価ボード Airio については 34 ページを参照

### Lua を使って、FlashAir のメモリにアップロードできますか？

できます!まず、/SD\_WLAN 内にある CONFIG ファイルに、「WebDAV=2」、「UPLOAD=1」を追記してください。そして、Lua の fa.request() の引数 method に PUT を指定してください。または、fa.request() の引数 url に upload.cgi を指定してもアップロードできます。逆に削除する場合は、fa.remove(“パス+ファイル名”)を使ってください。

### Lua を使って、FlashAir の IP アドレスを確認する方法はありますか？

あります!以下のような関数を呼んでください。戻り値に IP アドレスが返ってきます。

```
local ipaddress = string.sub(fa.ReadStatusReg(), 160, 168)
```

### Lua スクリプトのデバッグ方法はありますか？

検索キーワード“FlashAir Lua デバッグ”で検索すると、ユーザの方がデバッグ方法を説明してくれています。参考にしてください。

### FlashAir を抜き差しせずに、CONFIG の設定を変えたいのですが？

ホスト側でリマウント処理をすることで可能です。または、PIO モードでホスト側のリセットピンに割り込みを入れることで、ホストをリセットしてください。また、config.cgi を使って、APPMODE などを切り替えることでリセットすることも可能です。

なお、ボードを含めた動作については、自己責任でお願いいたします。



#### 寺西 (@soft128)

2015 年 1 月、東芝入社。大手電機メーカーから外資系半導体メーカーに転籍し、流浪の末にたどり着いた組込み OS 屋。前職では、展示会向けデモ開発をしていた経緯もあり、その流れで FlashAir Developers に参加。時々、FlashAir 芸人の付き人を担当。

# FlashAir を使ったプロトタイプを作ろう！

余熱

FlashAir を使った製品を開発したい!と思った時に、必要となるのが評価ボードです。既存の評価ボードは SD I/F や SPI I/F での接続が一般的であり、FlashAir 特有の GPIO 機能を使うための評価ボードはほとんどありませんでした。しかし、最近その状況が少しずつ緩和されてきています。この章では、現在入手可能な FlashAir の評価ボード「Airio」<sup>1</sup>と「Airio RP」の紹介と製作事例について紹介します。

## Airio

Airio(えありお)は FlashAir の GPIO 機能を簡単に使用するための評価ボードです(図 1)。FlashAir にフルカラー LED、スライドスイッチが接続されています。USB コネクタからは給電のみを行います。また、ピンヘッダ・ピンソケットを取り付けることで Arduino 用の SD カードシールドとしても使用可能です。Airio は SD カードスロットの端子がそのまま外ピンに出ないため、基板上に実装された LED、スイッチ以外のデバイスを使ったプロトタイプを行う場合には、後述する「Airio RP」の方が適しています。



図 1: Airio

Airio はスイッチサイエンス、マルツエレクト、三月兔から購入することが可能です。

設計については「FlashAir 同人誌」に記載していますので合わせてお楽しみください。

## 【活用事例】FlashAir の GPIO 機能の使い方

FlashAir は SD カードの信号線を汎用 I/O(GPIO) として使用することが可能です。GPIO 機能を使うことで FlashAir から LED やセンサデバイスを駆動することができます。そのかわり、SD カードとして使用することはできなくなってしまいます。GPIO として使える信号は「CMD, DAT0, DAT1, DAT2, DAT3」の 5 本です。本節では Airio を使い FlashAir の GPIO 機能を使ってみます。

1 評価ボードの入手先は変更になる場合があります。

2 FlashAir W-03 では Lua スクリプトからファイルアクセスを行うこともできるので、使い方によっては Flash メモリ領域を無駄にせず済みます。

## CONFIG ファイル

FlashAir の動作を制御するためには `/SD_WLAN/CONFIG` ファイルに設定を書き込む必要があります。このフォルダは隠しフォルダになっていますので、隠しファイルの表示設定を行ってください。CONFIG ファイルをテキストエディタで変更し、FlashAir を抜き差しすることで設定が有効になります。

GPIO 機能を有効にするためには `IFMODE=1` と追記します。また、Lua スクリプトを起動時に動作させるには `LUA_RUN_SCRIPT=/bootscript.lua` と追記します。

Lua 下記にサンプルの CONFIG ファイルを示します。初期状態の CONFIG ファイルの下に 2 行追記しています。

```
[Vendor]
CIPATH=/DCIM/100__TSB/FA000001.JPG
APPMODE=4
APPNETWORKKEY=*****
VERSION=FA9CAW3AW3.00.00
CID=02544d535730384731c3d68e7900f101
PRODUCT=FlashAir
VENDOR=TOSHIBA
LUA_RUN_SCRIPT=/bootscript.lua
IFMODE=1
```

## Lua スクリプト

Lua から GPIO 機能を使う場合には、`fa.pio` 関数を使用します。スイッチ入力の検出には `bit32.btest` を使っています。これは論理積 (AND) の結果を `true/false` で返す関数です。

下記の内容を FlashAir のルートディレクトリに `bootscript.lua` という名前で保存します。Airio に FlashAir を挿入すると、赤 LED が 3 回点滅し、その後スイッチ入力の値によって青 LED と緑 LED が光ります。

```
for i=1,3 do
    s, indata = fa.pio(0x0E, 0x02)
    sleep(200)
    s, indata = fa.pio(0x0E, 0x00)
    sleep(200)
end

while(1) do
    if (bit32.btest(indata, 0x01)) then
        s, indata = fa.pio(0x0E, 0x04)
    else
        s, indata = fa.pio(0x0E, 0x08)
    end
    sleep(200)
end
```

3 Mac での設定方法は 32 ページを参照してください。

## Airio RP

Airio RP は FlashAir でプロトタイピングを行うためのボードです (図 2、図 3)。下記の特徴があります。マルツエレクトロニクスから購入することが可能です。

### DIP 形状

900mil DIP 形状で FlashAir の全端子が外ピンに出ており、ブレッドボードを用いて簡単に外部回路と接続することができます。

### SPI-I2C ブリッジ IC

CP2120 を搭載しており、I2C デバイスと接続することが可能です。スライドスイッチを使い、CP2120 の有効 / 無効を切り替えることができます。

### リセットスイッチ・リセット端子

基板の上にリセットスイッチがあり、外ピンにリセット端子が出ています。FlashAir を抜き差しせずにリセット可能です。

### LED ピンの隠し機能

LED1 は電源 LED なのですが、NMOS を使って駆動させているため、外ピンの LED ピンを GND に落とすことで LED1 を消灯させることが可能です。ちょっとした動作確認を行うときに便利かも？

### ピンアサイン

表 1 に FlashAir の各 I/F におけるピンアサインを示します。Airio の LED、Airio RP 用のソフトウェア SPI のピンアサインも掲載します。

表 1: FlashAir のピンアサイン

ピン	SD I/F	SPI I/F	GPIO	Airio LED	Lua SPI
8	DAT1	-	0x04	Blue	CS
7	DAT0	DO	0x02	Red	CLK
6			Vss2		
5	CLK	SCLK	-		-
4			Vcc		
3			Vss1		
2	CMD	DI	0x01	SW1	DO
1	DAT3	CS	0x10	-	-
9	DAT2	-	0x08	Green	DI

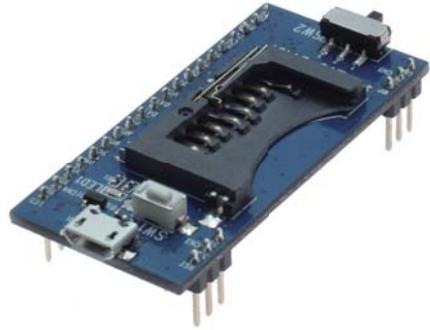


図 2: Airio RP

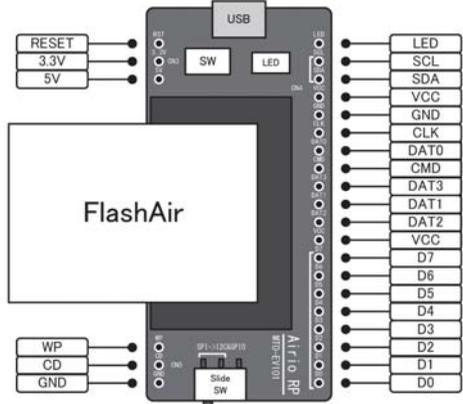


図 3: Airio RP のピンアサイン

## 【活用事例】IFTTT を使った iPhone への通知

IFTTT という web サービスがあります。簡単に言えばネット上のサービス同士をつなげるためのサービスで、例えば「明日の天気雨が雨ならメールを送る」というサービスをユーザが簡単にカスタマイズして使うことができます。このサービスを使って「タクトスイッチを押したときに iOS に通知が来る」サービスを作ってみましょう。

### IFTTT の設定

スイッチサイエンスマガジンに詳細な手順<sup>4</sup>が掲載されているので、詳細はここでは割愛します。下記の URL にアクセスしたときに iOS に通知が来るように設定してください。

```
https://maker.ifttt.com/trigger/test/with/key/[ シークレットキー ]
```

iOS のアプリをインストールして、IFTTT にログインします。レシピを有効にした後にアプリを再起動することで通知が有効になります。

### 基板の準備

Airio RP を使ってプロトタイプを行います。DATO 端子と LED ピンを接続します。FlashAir の端子は CP2120 によってプルアップされているため、CMD 端子と GND の間にタクトスイッチを接続します。SW2 は CP2120 を無効にするため、2-3 に合わせます。図 4 に配線例を示します。

### CONFIG ファイル

FlashAir を無線アクセスポイント (無線 AP) に接続するため、**APPMODE=5** に変更します。無線 AP の情報を **APPSSID** と **APPNETWORKKEY** を追記します。Lチカの例と同様、**LUA\_RUN\_SCRIPT** と **IFMODE** を追記します。

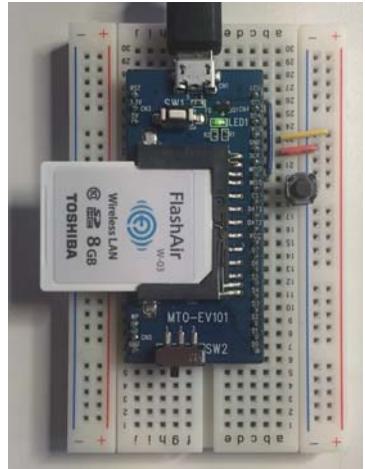


図 4: Airio RP を用いた配線例

[Vendor]

```
CIPATH=/DCIM/100_TSB/FA000001.JPG
APPMODE=5
APPSSID=[ 無線 AP の SSID ]
APPNETWORKKEY=[ 無線 AP のネットワークセキュリティキー ]
VERSION=FA9CAW3AW3.00.00
CID=02544d535730384731c3d68e7900f101
PRODUCT=FlashAir
VENDOR=TOSHIBA
LUA_RUN_SCRIPT=/bootscript.lua
IFMODE=1
```

4 IFTTT に Maker Channel ができました

<http://mag.switch-science.com/2015/06/25/ifttt-maker-channel/>

### Lua スクリプト

Lua スクリプトから無線 LAN 関係の関数を使う場合、起動時に sleep 関数を入れる必要があります。スイッチが押されたら GET リクエストを送り、iOS に通知が送られます。この作例は、Airio でも動作させることが可能です。

```
s, indata = fa.pio(0x0E, 0x00)      -- LED 消灯
sleep(10000)                        -- 10 秒スリープ
s, indata = fa.pio(0x0E, 0x02)      -- LED 点灯

flag = false                         -- 連続投稿禁止フラグ
while(1) do
  collectgarbage("collect")
  s, indata = fa.pio(0x0E, 0x02)      -- LED 点灯 & スイッチ入力読み取り
  if bit32.btest(indata, 0x01) then   -- スイッチ入力が 0x01 だった場合
    if flag then
      flag = false
      s, indata = fa.pio(0x0E, 0x00)  -- LED 消灯
      -- IFTTT に GET リクエストを送る
      b, c, h = fa.request[url = "http://maker.ifttt.com/trigger/test/with/key/[シー
      クレットキー]"]
    end
  else
    flag = true
  end
  sleep(200)                          -- 200 ミリ秒スリープ
end
```

この作例を応用して、例えば Airio RP に I2C で接続した温度センサの値を iOS に通知したり、スイッチが押された回数・温度センサの値をを FlashAir のフラッシュメモリに格納することも簡単にできます。

### おわりに

FlashAir は現状、試作用途で使うとその威力を発揮できると思います。評価ボードが揃ってきたことで、より使いやすくなりました。この本を読んで「こういうこともできるのか!」と思って頂けるととてもうれしいです。他の方の製作事例も是非読んでみてください。



#### 余熱 (@yone2\_net)

基板の裏面にパターンで絵を描いた「基板少女」の作者です。ひよんなことから FlashAir 同人誌作ってます。

8月からは正式に FlashAir チームに配属になりました。今後ともよろしく願います。

# FlashAir の共有メモリを使い倒せ！

綾瀬 ヒロ

FlashAir ファンの皆さん、こんにちは。私もファンの一人ですが、今回は、この紙面をお借りして、FlashAir の共有メモリについて紹介して、ファンの皆さんにも、そして、これから FlashAir を使ってみようという方にも、共有メモリを使い倒していただこうと思います。

## FlashAir の共有メモリ

FlashAir には、内蔵するマイコンにユーザーが自由に使える 512 バイトの共有メモリ空間が用意されています。この共有メモリは、揮発性メモリであり、SD ストレージ領域とは別で、不揮発ではないため FlashAir の電源が遮断されるとメモリ内容は消えてしまいます。しかし、この共有メモリには、いくつかの読み書きするアクセス手段が用意されており、これを活用することで、さまざまなデバイスから共有メモリにアクセスすることが可能です。図 1 に、FlashAir W-02 及び W-03 における共有メモリへのアクセス手段をまとめてみました。

一番容易に使えるのは、WebAPI として公開されている `command.cgi` から利用する方法でしょう。この WebAPI を使うことで、FlashAir と無線 LAN 接続された外部デバイスから HTTP リクエストでアクセスできますし、FlashAir に配置した HTML ファイル（例：

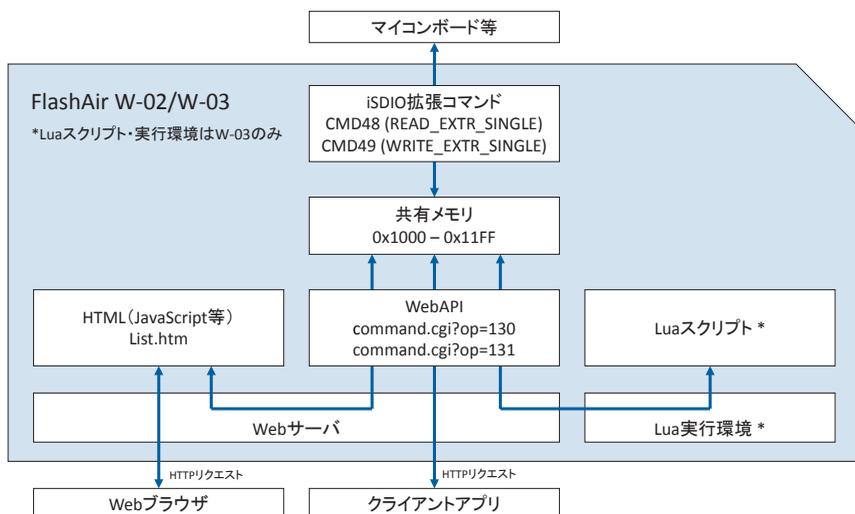


図 1: 自動実行 Lua スクリプトの動作

List.htm) の JavaScript や Lua スクリプトから HTTP リクエストを発行することでアクセスすることも可能です。また、FlashAir を SD カードとして接続したマイコンボード等からも iSDIO 規格でアクセスすることが可能です。

## 共有メモリを使うときの準備

前述の WebAPI でアクセスする場合は、何の準備も要りません。

マイコンボード等から iSDIO 拡張コマンドでアクセスする場合は、マイコンボードのプログラムに iSDIO のライブラリを追加する必要があるかもしれません。Arduino の場合は、FlashAir Developers サイトのチュートリアルを参考にするとよいでしょう。

## 共有メモリを使うときのポイント

ここでは、共有メモリを利用する上で、こうしたほうが良かったというを紹介합니다。

### 共有メモリで受け渡すデータはステータス情報とする

共有メモリを利用する場合、複数のデバイス間でデータを送受信したいという場合が多いと思います。一般的に、複数のデバイス間でデータを送受信する場合、図 2 に示すようなメッセージ方式とするかステータス方式とするでしょう。前者の場合、メモリ上には受信したデータをバッファリングすることで、受信側デバイスは受信したデータを読み込んで利用することになります。しかし、この場合、バッファリングしたデータを適切なタイミングで消去する必要があり、また書かれたデータをすべて読み込む前に、新たなメッセー

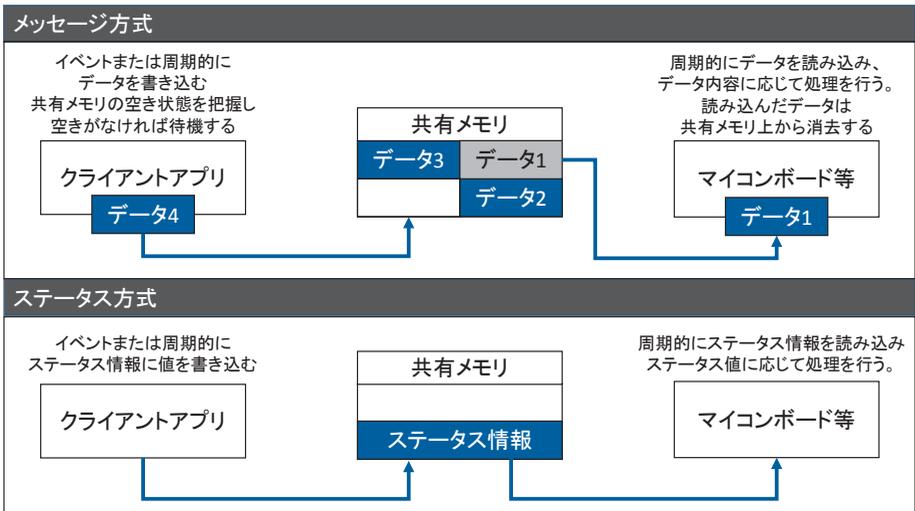


図 2: 共有メモリを使ったメッセージ送受信の方式比較

ジが到着することも考慮する必要があります。そのため、メッセージ方式の場合、プログラムが複雑になる可能性があります。一方、後者の場合、メモリ上には、決められた配列でステータス(状態)を表すデータを置いておくことになります。このデータは増減しないため、プログラム処理中にデータを消去しなければならないというような面倒はありません。常に、メモリ上の決められた範囲のデータを周期的に参照し、データ値に応じて処理を行うだけで良いので、プログラムが比較的簡単なものとなります。

### HTML ファイルから共有メモリにアクセスするには

HTML ファイルから共有メモリにアクセスするには、前述の通り WebAPI を使います。要するに HTTP リクエストを発行できれば良いので、JavaScript を用いて HTTP リクエストを発行するのが容易でしょう。このとき、URI はブラウザ等から FlashAir でアクセスする場合と同様になります。

```
//JavaScript で共有メモリにアクセスする例：
// 共有メモリの先頭から 8 バイト（文字）分のデータを書き込む
url = "http://flashair/command.cgi?op=131&ADDR=0&LEN=8&DATA=12345678" ;
var request = new XMLHttpRequest();
request.open("GET", url, false);
request.send(null);
result_value = request.responseText;
```

### Lua スクリプトから共有メモリにアクセスするには

Lua スクリプトから WebAPI に向けた HTTP リクエストを発行する場合、以下の点に注意する必要があります。

- Lua スクリプトを SD カード起動時に実行するようにしている場合、FlashAir の無線 LAN 機能が起動し終えてからでないと、リクエストに失敗します。Lua スクリプトの先頭部分で、無線 LAN 機能の状態を参照して起動し終えたことを確認してから本来の処理に入ることによって失敗を回避できます。面倒な場合は、経験上、Lua スクリプト先頭部分で sleep(30000) として 30 秒待機しても問題ありません。
- Lua スクリプトから WebAPI を呼び出すには、http://127.0.0.1/command.cgi のようにループバックアドレスを指定する必要があります。http://flashair/ などとすると NetBIOS の名前解決ができずリクエストが失敗してしまうようです。

```
--Lua スクリプトで共有メモリにアクセスする例：
-- 共有メモリの先頭から 8 バイト（文字）分のデータを読み込む
local get_uri = "http://127.0.0.1/command.cgi?op=130&ADDR=0&LEN=8"
local b, c, h = fa.request(get_uri)
if (c == 200) then
-- リクエスト成功時の処理をここに記述する
end
```

## 【活用事例】 鉄道模型制御装置の製作

### 概要

最後に、共有メモリを使った活用事例として、私が試作した鉄道模型ミニレイアウトに組み込んだFlashAirとArduinoによる列車運転制御と列車在線（位置）状態の取得、Lua スクリプトによる自動運転制御について紹介しましょう。この事例では、クライアントアプリはiOS アプリを自作し、iPad で動作させます。クライアントアプリではHTTP リクエストでFlashAirの



共有メモリを読み書きします。Arduino では共有メモリの状態に応じて接続したモータドライバに PWM 出力して車両のモータを、線路を介して駆動させます。また分岐器線路に内蔵された転換用電磁石に電流を瞬間的に流して分岐器の方向を転換しています。一方、線路は4区間に区分しており、区間ごとにフォトカップラで電流検知を行っています。Arduino では周期的に在線状態を共有メモリに書き込みます。クライアントアプリでは共有メモリを参照し、列車在線表示を更新しています。

### 共有メモリのステータス情報

共有メモリのステータス情報は、図3に示すように、それぞれ速度値(3バイト)、走行方向(1バイト)、分岐器の向き(1バイト)、列車在線状態(1区間あたり1バイト)、運転モードを示すフラグ(1バイト)を配置しました。運転モードは、クライアントアプリからの手動運転モード(0)と、クライアントアプリから自動運転を指定された自動運転モード(1)、FlashAirの初回起動完了を示すモード(2)の3種類のモードを持たせています。

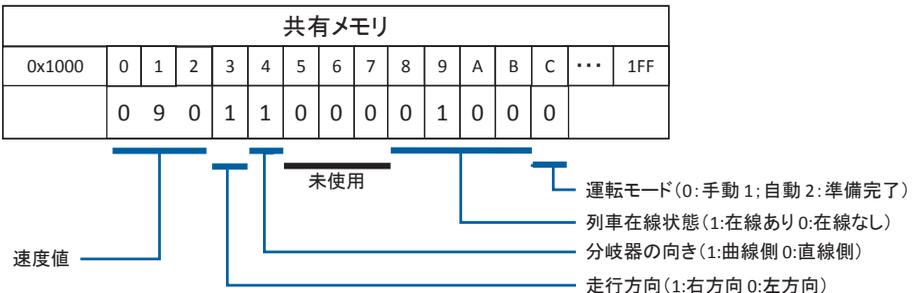


図3: 共有メモリにおけるステータス情報の定義

## 手動運転モードによる運転

手動運転モードでは、図 4 に示すように iPad の専用アプリケーションの各種ボタンを押下することで、共有メモリのステータス情報を HTTP リクエストで書き換えて、Arduino を制御します。

この専用アプリケーションは、個人的な勉強も兼ねてクロスプラットフォーム環境 Xamarin.Forms (C#) を使って開発しています。

## 自動運転モードにおける Lua スクリプトによる自動運転

この鉄道模型ミニレイアウトをほぼ製作し終えた 2015 年 3 月中頃、第三世代 FlashAir W-03 が発売となりました。W-03 の売りは、何と言っても Lua スクリプトによるプログラム実行環境が搭載されたことでした。これを活用しない手はない! ということで、鉄道模型の自動運転プログラムを作ることにしました。以前、パソコンと Arduino を使って鉄道模型の自動運転装置を製作したことがあり、このときにオリジナルの自動運転コマンドを定義したのですが、今回は、このコマンドを流用して、Lua スクリプトで自動運転できるようにしました。

図 5 のフローチャートに示すように、自動運転用の Lua スクリプトでは、FlashAir の SD ストレージ部分に自動運転するコマンドを記述したテキストファイルを置き、これを Lua スクリプト内で読み込み、コマンドに応じて、Lua スクリプトから HTTP リクエストで共有メモリのステータス情報に速度値、走行方向、分岐器の向きを書き込みます。

また、自動運転において列車在線状態(位置)を知る必要がある場合は Lua スクリプトから HTTP リクエストで共有メモリの列車在線状態(センサー値)を取得して、列車在線を検知したら次のコマンドの実行(例:列車を止めるなど)を行えるようにしました。

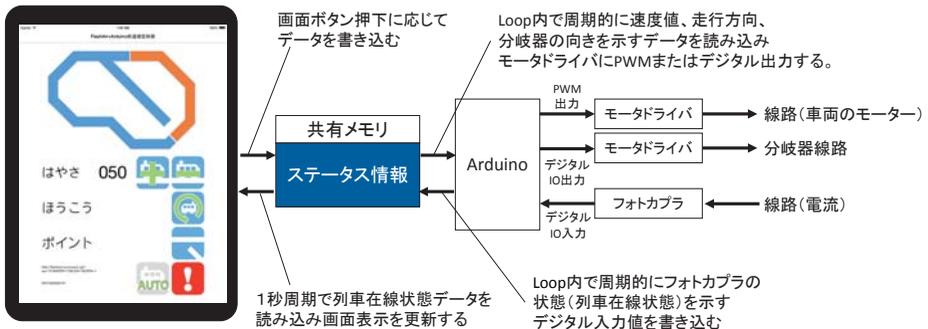


図 4: 専用アプリケーションと Arduino の連携

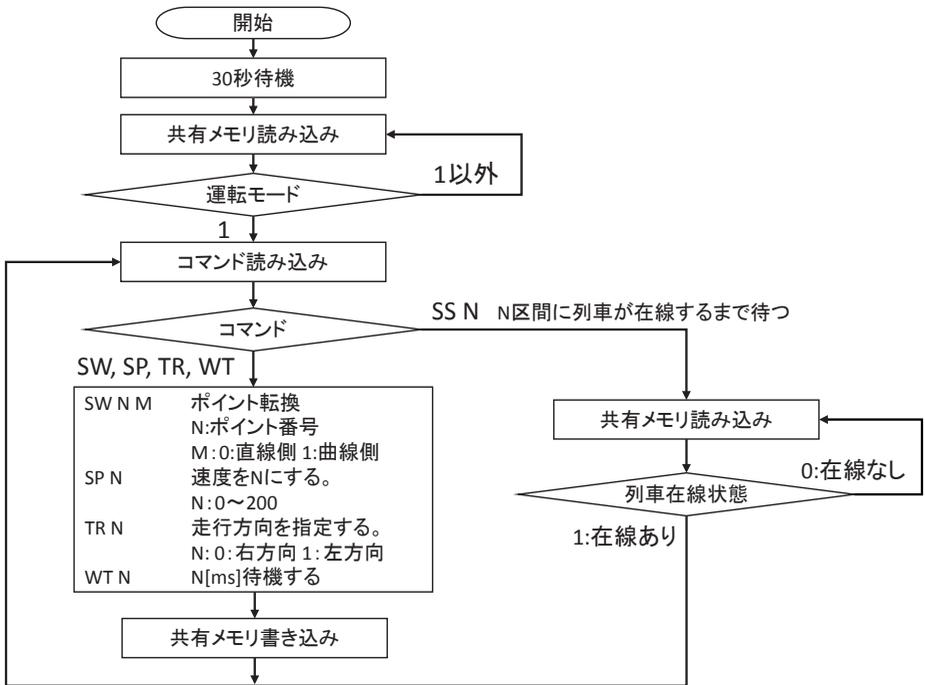


図 5: Lua スクリプトによる自動運転プログラムの処理フロー

### 綾瀬 ヒロ (@ayasehiro)

FlashAirを使った電子工作ファンの一人。

本業は、鉄道運行管理システムのSEです。

これまで Arduino と有線 LAN モジュールで鉄道模型制御装置を作りましたが、HTTP リクエスト処理部を自作する必要がありました。FlashAir ではこれを肩代わりしてくれる。FlashAir の優位性は、この部分にあると思います。

本稿のご質問は、Twitter @ayasehiro までお願いします。

秋ヶ瀬軽便鉄道 <http://www14.big.or.jp/~ayase/rail/>



## FlashAir との出会い

FlashAir を知ったのは、とある日本の Apple II コミュニティの方が、私の作ったフロッピー・エミュレータ UNISDISK ( 図 1 ) で使いそうな Wi-Fi 内蔵 SD カードを色々試して下さいました。SPI (シリアル・パライフェラル・インタフェース) に対応していないカードもある中で、FlashAir は SPI に対応しており、UNISDISK でも使えたという事で、私も一つ購入してみることにしました。

UNISDISK は、往年の名パソコン Apple II 用のフロッピー・エミュレータで、SD カード上のフロッピー・イメージに対し、本物のフロッピーと同じように読み書きが出来るものです。Apple II は 30 年以上も前のパソコンであり、現在では愛好家の活動により支えられています。今なお世界各国の愛好家が様々なソフトウェアや周辺機器を作っており、私もその一人としてディープな世界にはまり込んでいます。

UNISDISK のハードウェアはごくシンプル。Atmel のマイコン XMEGA AVR と 74HC125、LCD、押しボタンスイッチ、その他で構成されています。SD カードに記録されたイメージを用いてフロッピー・ドライブのプロトコルをエミュレートします。プロトコルのタイミングを作る部分には AVR のアセンブラを用い、クロック数を数えながらプログラミングしました。

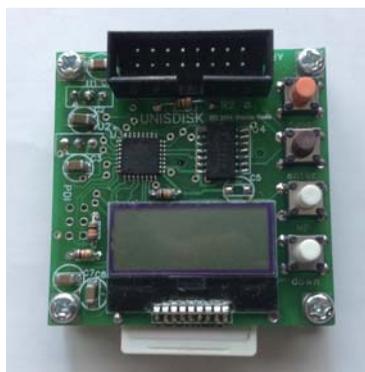


図 1: UNISDISK

## FlashAir への対応

FlashAir を購入して驚きました。これは単なる Wi-Fi 付の SD カードではない。共有メモリがあり、SPI から、ブラウザ上で動作する JavaScript からアクセスが出来る。もしかしたら、押しボタンスイッチと LCD でやっているユーザ・インターフェースを、全てブラウザ上で出来るのではないかな？

マイコンのプログラムには慣れていましたが、JavaScript なんてホームページを作る時に少しかじっただけです。FlashAir のアップロードの例題を参考に、FlashAir の API と共に勉強を始めました。「なるほど、jQuery を使えば JavaScript をシンプルに書けるのか。」「共有メモリへのアクセスは、FlashAir に対して SPI からコマンド 48、49 を発

行すれば出来る。」「ブラウザからは、command.cgi の op=130、131 で読み書きが出来る。」さらに FlashAir Developers のフォーラムで「CMD48/49 の、I/O スペース、Func#=1 へのメモリ読み書き。Field の各値は、MIO=1、FNO=1、MW=0 (Write の場合)、ADDR=1000h ~ 11FFh、LEN= 長さ -1」という情報を見つけました。これで何とかできます。

試行錯誤の末、UNISDISK+FlashAir は Wi-Fi を介して Web ブラウザからのコントロールが可能となり、ディスク・イメージのアップロード、ダウンロードだけでなく、ディスク・イメージの選択、モードの選択、UNISDISK のリセットもブラウザから出来るようになりました (図 2)。以下に、SPI から共有メモリへアクセスする AVR のコードの一部を示します。



図 2: ブラウザでのコントロール

```
SD_cmd (48, ((uint32_t)0b1001<<28) | (((adr+0x1000)&0x1ffff)<<9) | (((uint16_t) len-1)&0x1ff), err);
SD_cmd (49, ((uint32_t)0b100101<<26) | (((adr+0x1000)&0x1ffff)<<9) | (((uint16_t) len-1)&0x1ff), err);
```

## FlashAir 専用モデル

既に FlashAir によりブラウザからコントロールできるようにはなっている。しかしユーザの認知度は低い。画期的なことが出来るのに多くのユーザは興味を示さない。原因は、多くのユーザが FlashAir を単なる Wi-Fi 付 SD カードと見なしていることだと思いました。UNISDISK を FlashAir に対応させた時、「ラヂオ君、どのメーカーの Wi-Fi カードが一番良いの?」と聞かれることもありました。「いや、だから FlashAir しか使えないよ。」

そこで新しい UNISDISK では、LCD とスイッチを省くことにしました。LCD は故障が多い部品で悩みの種でもありました。しかし、このアイデア



図 3: UNISDISK Air c

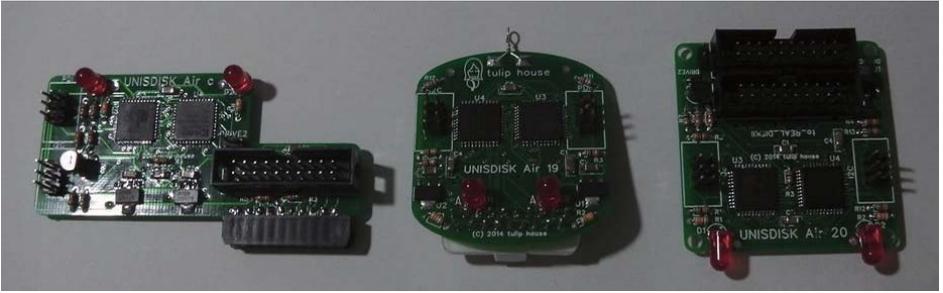


図 4: UNISDISK Air シリーズ

を海外の友人に話した時の反応は、良いものばかりではありませんでした。「お願いだからLCDを残してくれ。」ここは作って示すしかありません。

LCDとスイッチを省くメリットがもう一つありました。Apple IIcという機種は、フロッピー・ドライブを内蔵しているのですが、外付け用のフロッピー・ポートに接続したドライブからは起動することが出来ないのです。フロッピー・エミュレータ自体をApple IIcに内蔵してしまうことで全てを解決できるのではないかとそこでApple IIc用は専用のモデル(図3)とし、他機種用にも独立したモデルを作ることになりました。名前はUNISDISK Airシリーズとし、合計3モデルで販売を開始しました(図4)。

私はスマートフォンを持っていませんが、スマートフォンで操作できればユーザは嬉しいだろうと考え、最初の売上でiPod touchを購入、iPhoneやiPod touchでも使えるようにWebアプリを改良しました(図5)。ブラウザはiCab mobileを使います。Apple IIがUNISDISK Airにアクセスすると、ブラウザ上のドライブ・アイコンが光り、現在のトラック番号もリアルタイムで表示されます。これは結構なインパクトがあったようで、ようやくApple IIユーザに受け入れられました。合計約100台を手作りで製作し、世界中のApple IIユーザに発送しました(図6)。



図 5: スマートフォンで使う



図 6: UNISDISK Air の発送

## LOGIC BOY

UNISDISK Air シリーズでは、74HC125の代わりにCPLDを使いました。さらに、マイコンのファームウェアとCPLDのコンフィグレーションを、FlashAirにアップロードしたファイルからアップデートできるようにしてみました。機器のリリース後の改良を簡単にアップデートできることは、開発者にとってもユーザにとってもメリットです。

マイコンやCPLDにより電子工作の楽しさは飛躍的に向上しました。私はレトロPC用のビデオ・カードも作っていますが、ビデオ信号を扱うにはマイコンとCPLDだけでは力不足で、やはりFPGAを使いたいところです。

そこでXilinxの安価なFPGA XC3S50Aのベアチップを購入し、使ってみることにしました。FPGAを使うのは学生時代以来、19年ぶりです。

FPGAのコンフィグレーションは通常揮発性メモリに記録されるので、電源を切ると消えてしまいます。しかしFlashAirを使えば、これはむしろメリットになるのではないかと思います。そこでFlashAirの使用を前提とした手作りのFPGA実験ボードを作りました。名付けてLOGIC BOY(図7)。AVRマイコンも搭載し、AVRのフラッシュメモリにFPGAのコンフィグレーション・データも置けるようにしてみました。

FlashAirからFPGAに直接、コンフィグレーション・データをダウンロードしたり、まずAVRのフラッシュメモリにコンフィグレーション・データを書き込み、それをFPGAにダウンロードしたり、さらにAVRのファームウェアもFlashAirからプログラムできるようになっています。ちょっとした実験から、レトロPCの周辺機器のプロトタイプ開発まで、大変重宝しています。

LOGIC BOYを何に役立てるのか、他にも色々妄想しています。小さな容量のFPGAなのでコンフィグレーション・データは約53Kバイト、FlashAirからFPGAへのダウンロードもわずか一秒程度です。AVRのアプリケーションにより、離れた場所にあるLOGIC

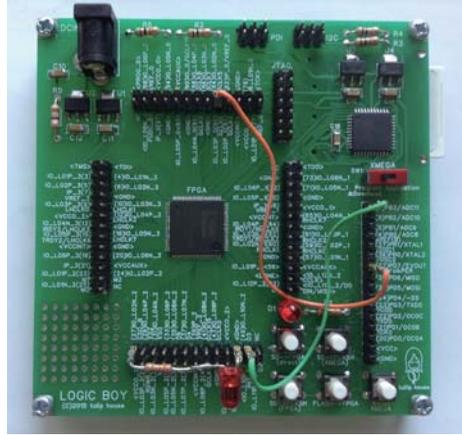


図 7: LOGIC BOY



図 8: LOGIC BOY を  
ブラウザでコントロール

BOYのFlashAirにコンフィグレーション・データを僅かな時間でアップロードしたり、コンフィグレーションをブラウザからリアルタイムで切り替えたり、ブラウザにLOGIC BOYからの情報を表示したりも可能になります(図8)。数えきれないほどのコンフィグレーション・データをFlashAirに入れておき、必要に応じてソフトウェアで瞬時に切り替えて使用することも考えられます。このような夢が、とても安く簡単に可能になったことがポイントだと思います。楽しさは無限に広がりそうです。

### FlashAirの可能性

FlashAirは、デジタル無線通信を僕達に解放してくれました。マイコンのメモリに余裕さえあれば、いつも作っているSDカード付の機器のファームウェアを改良するだけで、新たな魅力を追加できるでしょう。共有メモリを使えば、機器のLCDやスイッチを省き、PCやスマートフォンで操作が出来るようになるでしょう。HTML5とJavaScriptで、ブラウザ上にリッチなUIを手軽に作れます。もちろん大容量の記録メディアとして使用できます。機器をインターネットに接続することも出来るようになるでしょう。新しいFlashAirでは、さらに新しい機能が追加されているようです。やってみたいことが多すぎて頭が破裂しそうですが、FlashAirにはそれほどのインパクトがあるということ。いろんなモノと組み合わせることで可能性は無限に広がります。

また、作った機器をユーザにどうアピールしていくかも大事だと感じました。今後は、出来るだけ煩わしい設定などをしなくても使えるようにするなど、もっと手軽に楽しんでもらえるものを作りたいと思います。

私の活動は、以下のURLから辿れるようになっています。

<http://tulip-house.ddo.jp/>

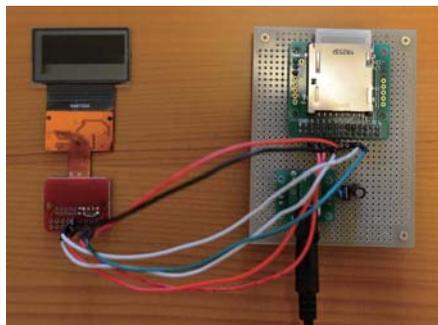


#### 西田 ラチオ (@bittercherryjam)

手作りハードウェアの作家。フロッピー・エミュレータ、ビデオアダプタ、ジョイスティック・アダプタ等、レトロPC用の多くの周辺機器を製作。インディー・ポップという音楽ジャンルのバンドもやっています。年に2日は、妻の手作り雑貨の販売を手伝います。好きなことは、半田付け、回路設計、プログラミング、曲作り、ライブ、田舎で過ごすこと。好きな場所は自分の部屋。好きな言語はOCaml。好きな食べ物は芋。悩みは体が弱いこと。

## はじめに

FlashAir のファームウェア 3.0.0 以降から FlashAir のカード内で Lua スクリプトが動作するようになりました! 前回の同人誌では、モノクログラフィック液晶を FlashAir の CGI 経由で制御しましたが、転送レートが約 1.8bps だったのですが、今回は Lua の PIO 機能で同じ液晶を制御して、Lua スクリプトの効果を検証します。



## Lua スクリプトの起動方法

ファームウェア 3.0.0 以上の FlashAir では、図 1 のように、電源投入後、FlashAir に保存した Lua スクリプトを FlashAir で実行する仕組みがあります。今回は、この自動実行機能で Lua スクリプトを動作させて SPI 液晶を制御します。

電源投入後に自動実行する Lua スクリプトは、FlashAir の SD\_WLAN/CONFIG ファイルの LUA\_RUN\_SCRIPT で指定します。例えば、FlashAir の lua/spilcd.lua を自動実行させる場合は、下記のように CONFIG ファイルを記述してください。

```
...  
LUA_RUN_SCRIPT=/lua/spilcd.lua  
...
```

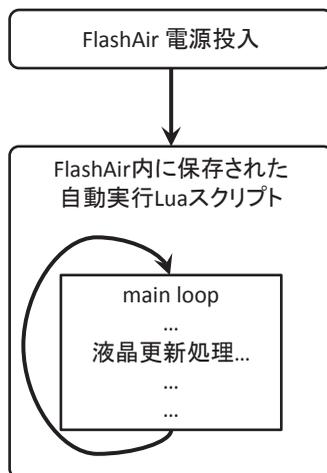


図 1: 自動実行 Lua スクリプトの動作

また、Lua スクリプトから SD インターフェイスを PIO として利用するために下記の設定も追加してください。

```
...  
IFMODE=1  
...
```

### Lua スクリプトでの PIO 制御と待ち時間制御

Lua から PIO 制御、待ち時間制御を行って液晶を制御します。Lua スクリプトからの PIO 制御は、fa.pio(ctrl,data) 関数が利用できます。待ち時間制御は、sleep(msec) 関数が利用可能です。詳細は、FlashAir™ Developers サイトをご確認ください。

### 結果

前は、100x32 ドットの液晶の全面の書き換え処理に、FlashAir の CGI 経由で 35 分かかっていましたが、今回は、Lua の PIO 機能を利用することで、7.5 秒に短縮されました。前回に比べて 280 倍高速になっており、静的な情報を表示する用途には、十分実用的な速度になっています。また外部から GPIO を制御する必要がなく、FlashAir 単体で動作するようになりました。

### 広がる FlashAir の応用例

FlashAir の Lua 環境には、外部のホストと情報のやりとりが可能な API(http/ftp) も用意されています。これらの API と FlashAir の 5 本の GPIO を組み合わせることで、様々な応用例が考えられます。

例えば、センサーから取得したデータを FlashAir 内で処理してからクラウドに送信することができます。また、クラウドから取得した情報を FlashAir の GPIO 経由で表示するといったことも可能になります。



#### 村口

自転車と電子工作が趣味で、座右の銘は、「困難は分割せよ」です。最近、結婚しました。初心を大切にしていきたいと思えます。

# FlashAir でホビーロボットを IoT 対応に！

ペンギンキャット

デザインも機能も優れた、親しみやすいホビーロボットが簡単に手に入る時代になりました。また、IoT というモノとインターネットの融合がブームです。各種センサーや家電、コミュニケーションロボットなどが連携してコンシェルジュのように振る舞うことで、生活をより便利で安全なものにしてくれる。こんな生活に近い将来実現するかもしれません。こんな生活の一部を、FlashAir で既存のホビーロボットを IoT 対応にすることで体験してみましょう。

## FlashAir との出会い

---

ガジェットが好きな私が、FlashAir に興味を持ち購入に至るのはあっという間でした。今や FlashAir は第三世代 (W-03) まで発売されていますが、2012 年当時はまだ第一世代 (W-01) が発売されたばかりの頃でした。まだそのころは今ほど API が充実していなかったこともあり、買ってはみたものの、面白い使い方が思いつきませんでした。オレンジ色のカードも持っていたので比較してみたり、デジタルカメラに入れてフォトシェア機能で無線 LAN 経由によりパソコンのブラウザから写真を見たりといった、FlashAir の普通の使い方（これだけでも画期的な機能だとは思いますが）しかしていなかったと思います。2013 年にはインターネット同時接続機能が追加された第二世代 (W-02) が発売されましたが、面白い使い方が思いつかず、購入を迷っていました。

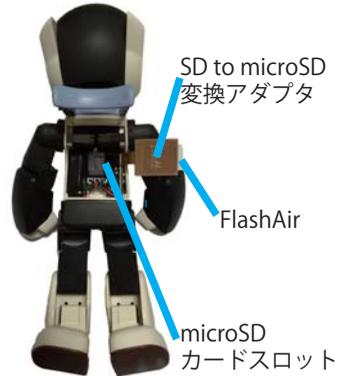
## ホビーロボットとの出会い

---

大学時代の研究テーマがロボットであったこともあり、自分でホビーロボットを組み立てられるパーツ付きマガジンが発売されると知った私は早速購読を申込み、少しずつ組み立てて 2014 年の 8 月頃に完成させました。このホビーロボットにインターネット接続機能は無く単体で動作します。また、Micro SD カードにプログラムが入っています。収録されているファイルをパソコンで見ると、プログラムはテキストベースの専用のインタプリタ型言語であることが分かりました。プログラミングキットは用意されていないものの、試行錯誤すれば独自のロボットの動作をプログラミングできるのではないかと思います。また、FlashAir のファイルアップロード機能を使い、無線 LAN 経由でロボットプログラムやデータを書き換えたら面白いのではないかとアイデアが膨らみました。早速、購入を迷っていた FlashAir 第二世代と SD から Micro SD への変換アダプタを購入し、ホビーロボットに FlashAir を挿入しました (図 1)。

## プログラミングで試行錯誤するはずが

しかし、ホビーロボット完成で結構満足してしまった私は、ホビーロボットのプログラムを試行錯誤しながら自分でプログラミングしようと思いつつ、しばらく放置していました。そしてしばらくして2015年1月初めのことです。インターネットで独自のプログラミングツールを公開されている方を見つけました。衝撃と共に、先を越されてしまったという感じでしたが、試行錯誤する必要がなくなり、やりたいことを実現するため環境が整いつつありました。



ホビーロボットの背面

図 1: ホビーロボットに FlashAir を挿入

## スマホでホビーロボットを遠隔操作

FlashAir 第二世代では単体で能動的にインターネットからデータを取得することが難しそうであったので、まずはホビーロボットを遠隔操作してみようと思いました。この時、パソコンとホビーロボットを USB ケーブルで直接接続してプログラムを書き換えて遠隔操作するツールを公開されている方はずでに見えたので、これを FlashAir によって無線化する方法を考えました。せっかくなのでスマホからか



図 2: スマホでホビーロボットを遠隔操作

こよく操作したいところです。HTML5 を使った Web アプリケーションならばブラウザから FlashAir の CGI 形式の API を使えるので、スマホ専用アプリを作ってインストールする必要も無く、簡単に実現できそうだと思います。そこで、まずはファイルのダウンロードとアップロードを試してみました。パソコンなどの SD カードホスト機器では SD カードの内容をキャッシュしている場合があり、注意しないとファイルのアップロードを行った際にファイルシステムの不整合が発生して不具合が起きる可能性があります。このホビーロボットではキャッシュしていないようで問題は発生しませんでした。しかし、ファイルの一部を編集して変数を書き換えるなどの処理もないとプログラムが作りにくいという別の問題に突き当たりました。スマホ用の専用アプリを作れば特に問題もなく出来ることなのですが、Web アプリケーションで簡単に済ませたいと思っていた私は、HTML5 の File API を使うことにしました。これを使えば、ブラウザで FlashAir からダウンロードしたファイルをメモリ上に読み込み、編集してアップロードといった一連の操作ができます。スマホからあらかじめ用意されているプログラムを選択して実行する簡単なものですが、2015年1月末に FlashAir で遠隔操作できるようになりました (図 2)。

### FlashAir が Lua スクリプトに対応

---

2015年1月に海外では Lua スクリプトに対応した FlashAir 第三世代が発表されました。これはついに待望の機能が来たという感じでした。Lua スクリプトは FlashAir 上で動作します。実行方法は3つあり、① FlashAir 起動時に実行する、② FlashAir へのファイル書き込み時に実行する、③ FlashAir への HTTP アクセス時に実行するになります。

①を利用すれば、定期的に単体で能動的にインターネットに対しデータを送受信することができそうです。また、②を利用すればホスト側機器がファイルを書き込んだタイミングに連動させて何か処理をさせることができそうです。発売が非常に待ち遠しい限りでした。2015年2月末に一部ショップで海外向けパッケージの販売が始まるとすぐに購入しました。Lua スクリプトは初めてだったので、解説書も購入してプログラミングを始めました。

### ホビーロボットに天気予報を教えてもらう

---

数日で Lua スクリプトには慣れることができました。あとは、どんなプログラムを作るのかです。インターネットから何かデータを取得してうれしいもの、ホビーロボットの音声認識語と相性が良いもの考えた結果、「おはよう」に反応してインターネットから天気予報情報を取得して教えてくれる機能になりました(図3)。天気予報情報は Open Weather Map API を使えば簡単に取得できます。また、Lua スクリプトならば JSON 形式のデータも簡単に扱え、さらに取得したデータを機器が理解できる形式に変換して保存することもできます。「おはよう」を認識した際に FlashAir へ Lua スクリプト実行のトリガーになるようにファイル書き込みするようにしました。どのファイルが書き込まれても指定した Lua スクリプトが実行されてしまうので、Lua スクリプト側でトリガーファイルかどうか確認し、さらにその中のコマンド値を確認するようにしました。コマンド値が天気情報取得のものであれば、Open Weather Map API にアクセスして天気情報を取得したあと、データ形式を変換してファイルに保存します。コマンド値は複数設定できるので、工夫次第で天気予報以外にも扱うことができます。ホビーロボット側はファイル書き込みの後、数秒間待ってデータ取得が完了した頃を見計らい、天気情報が保存されたファイルを読み込むようにしました。正常にデータが取得できていれば、対応した音声ファイルを再生することで「今日は晴れだよ」などと天気予報を教えてくれます。2015年3月中旬に FlashAir 第三世代の国内パッケージ販売が開始される頃には、FlashAir を使って天気予報情報を教えてくれるプログラムが完成しました。

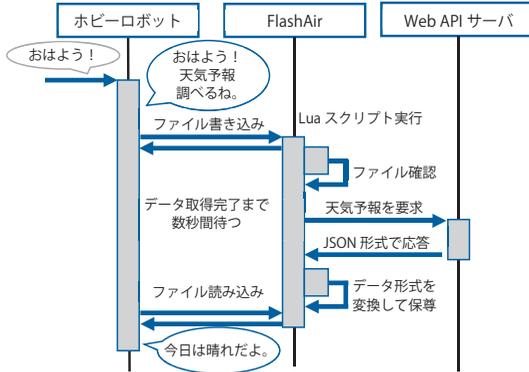


図 3: ホビーロボットに天気予報を教えてもらう処理の流れ

## おわりに

今回はインターネット接続機能を持たないホビーロボットを FlashAir により IoT 対応にし、遠隔操作や天気予報を教えることができましたようにしました。コミュニケーションロボットと暮らす近い将来の生活の一部を体験! ?とまでは行かなかったかもしれませんが、可能性は感じていただけたのではないのでしょうか。FlashAir を活用することで、既存の機器側にあまり手を加えることなく、無線 LAN を利用しスマホからの遠隔操作やインターネットから取得したデータを利用する機能を実現できました。Web API で利用できる機器やサービスはすでに多くあります。うまく機器と連携させることで、もっともっと面白いことができるでしょう。

本記事で紹介したプログラムや手順をまとめたものを下記サイトで公開しています。

<https://sites.google.com/site/gadgetszanmai/robi/remote>

### 免責事項 (おことわり)

本記事に記載の情報は自己責任において利用してください。ホビーロボットの改造に関して一切の責任を負いません。また、発売元への問い合わせは行わないでください。



### ペンギンキヤット (@penguin3neko3)

組み込み系ソフトウェアエンジニアで、デジタルガジェットが好きです。最近はロボットや家電を連携させて、何か新しく面白いことができないうかとネタを探している毎日です。

# ATARI 仕様 Wi-Fi コントローラーの制作

せいみまさみ

「最近、モノづくりらしいこと何もしてないなあ」

なんて思っていたさなか、ふと Maker Faire Tokyo 2014 に立ち寄った中、企業が同人誌作って参加している??と驚きで立ち寄った FlashAir のブース。そこでもらった同人誌を読むと、SD カードなのに I/O ポートがある? LED の点滅? クレーンゲーム? などなど良い意味で頭のおかしいネタがいっぱいあってとても刺激的でした。

こういう面白いハードを使ったお手軽モノづくりに参加してみようかと FlashAir を買って作ってみたのが、今回紹介する FlashAir による ATARI 仕様 Wi-Fi コントローラーです。LED 点滅 (L チカ) と同人誌 1 号で書かれていたワイヤレスクレーンゲームの元になった GPIO 制御の簡単な応用事例ですので、合わせて読んでみると良いと思います。

## MSX と ATARI 仕様コントローラー

皆さん、MSX ってご存知ですか?

MSX とは今から 32 年前の 1983 年にマイクロソフトとアスキーが作った 8 ビットコンピュータの共通規格で、全世界に 500 万台弱普及した、ホームコンピュータです。まあ言うなれば Xbox One の遠いご先祖様に当たります。パソコンなので、ゲームをやるにしてもキーボードのカーソルキーなどを使うのが普通でしたが、ゲーム用コントローラーを接続するための、汎用入出力 (ジョイスティック) ポートも付いていたので、多くの MSX ゲーマーはいろんなメーカーから出ているコントローラーをそこに繋いでいました。



その MSX に接続できるコントローラーの規格が ATARI 仕様コントローラーです。ATARI 仕様コントローラーは最大で上下左右の 4 ボタンと A、B ボタンの計 6 ボタンの ON/OFF の状態がそのまま入力になる単純な仕様となっています。80 年代のレトロ PC の世界ではほぼ標準の仕様で機構が簡単なので自作ハードウェアの入門記事としてコントローラーを自作する記事も書かれたりもしていました。なので、それらに習いこの記事では FlashAir を使用して ATARI 仕様 Wi-Fi コントローラーを作ってみようと思います。

## Wi-Fi コントローラー Version1

さてここからは、W-02 を使用した Wi-Fi コントローラー Version1 の紹介です。MFT2014 などワイヤレス L チカや GPIO 機能の応用を見たときに、

- LED がチカチカするということは、LED のスイッチを ON/OFF 出来るということ
- LED のスイッチが入られるのなら、コントローラーのスイッチ (ボタン) を ON/OFF 出来るということ

という連想から、FlashAir でコントローラーのボタンの制御ができるじゃん!! と、ひらめいたのが大元です。すごく単純に書くと MSX の汎用入出力 (ジョイスティック) ポートに FlashAir の GPIO ポートを接続するだけということになります。

ベースとなるのはワイヤレススクリーンゲームの元となる、ソラちゃん作者のじむさんの GPIO 機能の回路図<sup>1</sup>になります。制作の前に簡単な回路図を書こうと、よくよく考えてみると大きな問題がありました。それは FlashAir の I/O が 5 ポートしかないことでした。MSX などの ATARI 仕様では上下左右の 4 ボタンに A、B ボタンが標準で 6 ポート必須なのです。ゲームを楽しむにはいろいろ問題はあるのですが、悩んでいても仕方ないので、とりあえず A ボタンのみのコントローラーと割り切った制作にしてみました。また、Wi-Fi なのでレスポンスもそんなに良くないだろうけど、とりあえず作ってみたいとわからんよなぁと行き当たりばったりでの制作に突入です。モノづくりにはイキオイも大事なのです!

ということで作った Wi-Fi コントローラーの回路図は、SD カードスロット (CN1) に 3.3V を 5V にするレベルコンバータ (U1) と、スイッチ IC (U2,U3)、最後の出力先が MSX を接続するための AMP9 ピンケーブル (CN2) というシンプル仕様となります。

最近では、ブレットボードでの制作が流行っているようなので、こういったお手軽制作派は気兼ねなく作業できてうれしいですね。また FlashAir の I/O の数は高々 5 つなので、規模が大きくなりようがないのが、特に良いです。基板制作のレイアウト等を、あまり考えずに済むのもありがたいです。

MSX のコントローラーインターフェースである汎用入出力 (ジョイスティック) ポートは、5V の電源が出ていますが、規格上容量が 50mA 程度しかないので、FlashAir は電力が足らずに起動できません。MSX 本体と AMP9 ピンケーブルだけで完結せずちょっと残念ですが、とりあえずの対策で外部からの電源供給を行い、この問題を解決しました。

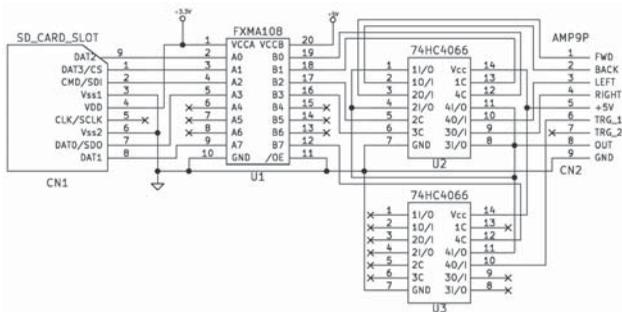


図 1: Wi-Fi コントローラ回路図

1 <http://rdstyle.cocolog-nifty.com/gm/2014/08/flashairgpio-be.html>

2 ATARI 仕様コントローラーの一般的なコネクタ。D-SUB9 ピンとは形状がほぼ同一であるが、D-SUB は耳が付いているが、AMP は耳が付いていないなどで厳密には異なる。

FlashAirを挿入するSDカードスロットは、いろいろなところで売ってはいますが今回は一番安くて3.3Vも用意されている秋月電子のSDカードスロットDIP化モジュールを流用しました。そのままではGPIOモードで使うことが出来ないので、チップ抵抗の取り外しとR1とR4、R6とR7のパターンを利用したジャンパー、そしてR4近辺のビアとSDカードスロットの端子の間の所で行うパターンカットを行います。カットの所は間隔が狭いので要注意!

ハードウェアだけではデジタルモノは動きません。もう一つ重要なのは、FlashAirのI/Oを制御するためにFlashAirに書いておくソフトウェア。今回はとにかくお手軽にモノづくり、がポイントだったので手間なく作業をするために、じむさんのJavaScriptのアプリケーションを拡張して、ゲームコントローラーをなんとなく再現したレイアウトにして表示するようにしました。こちらのアプリの反応速度もレスポンスには影響ですが、まずは動かしてみるのは。

そして完成したコントローラーをMSXに接続して、簡単なMSX-BASICのプログラムで動作確認をしてみました。FlashAirをブラウザからアクセスして、表示されたコントローラーのAボタンをタッチすると!ちゃんとMSX上でAボタンが押されている状態なのが確認できました。上下左右のボタンも押してみても、こちらも想定通りに状態が変わります。レスポンスはやっぱりちょっと悪いのですがこれはお試しアプリなのもあるので、アプリをちゃんと作ればある程度は改善するでしょう。

ネットワーク機能を持っていないMSXがお手軽にIoT(Internet of Things:モノのインターネット)が実現できたことは、大きな一歩であると言って過言ではありません。お

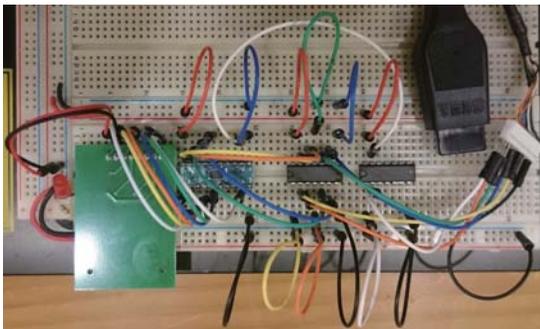


図3: Wi-Fi コントローラ

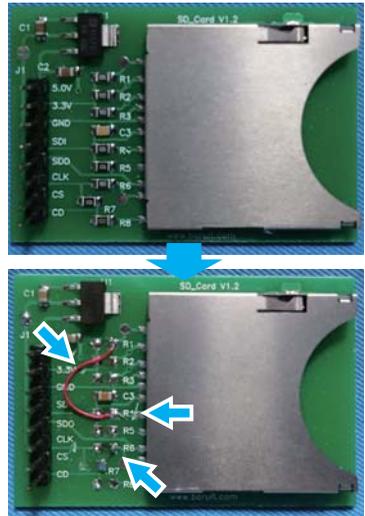


図2: 改造前(上)と改造後(下)のSDカードスロット

バカネタではありますが、Wi-FiでMSXを制御できてしまったのです。FlashAirのIPアドレスをグローバルにして公開すれば、宇宙空間からでも世界中からでも、スマートフォンやiPad、PS Vita、ニンテンドー3DSなど最新のハードウェアがWi-FiコントローラーとしてMSXの下僕に成り下がって「ここ」にあるMSXで遊べることになるのです!

## Wi-Fi コントローラー Version2( 仮 )

Version1 が出来るちょっと前に、新 FlashAir W-03 が発売されていました。W-03 ではスクリプト言語 Lua の対応がされ、これにより FlashAir が SPI マスタになれるのが大きなポイントです。この機能により、FlashAir ハッカソンで提供されたプロトタイプ基板 Airio RP を使用することによって、簡単に I/O が 8 ポートに増えるというのが個人的に一番うれしいポイントだったので調子に乗って Version2( 仮 ) を作ろうと思いました。

回路的な違いは、FlashAir の I/O ポートからレベルコンバータに接続するところが SPI と GPIO を変換する IC、CP2120 を介しての接続になったということです。コントローラー的には、2 ボタン対応が可能になったということが一番重要です。

ということで、原稿のページ数 ( と締切 ) の制限になってしまったので、実際の動作の結果がどうなっているかは MFT 当日のお楽しみです。間に合うと良いなあ。

## おわりに

FlashAir は MSX をはじめとする 80 年代レトロ PC にとって大きな福音です。今回の私の記事ではまったく関連がありませんが、レトロ PC のストレージ・ネットワークの問題を一気に解決するアイテムであると言えます。この記事を書きながらも、MSX 用に FlashAir の本来の機能 (?) のネットワークを使用したおバカなアイデアが 2 つ、3 つ浮かんでいます。次回にはそういったものを公開できればと思っています。もちろん実用性は皆無ですが、趣味なのでそういったものが無くても良いですよ！

ネットワーク用のモジュールを制御するための別 CPU がいない FlashAir は、本当にちょっとした工作で IoT を実現可能なので皆さんも手を出しやすいのではないのでしょうか。お手軽モノづくりとしての FlashAir の利用、なかなか楽しいと思います。皆さんも、ふと思いついた FlashAir でのお遊びをぜひ実現してみてくださいませ。ふと気づくとこの冊子に原稿書くようになっているかもしれませんよ。

### せいみ まさみ (@masa\_seimi)



MSX が好きすぎて、MSX のハードウェアを作ったメーカ ( の関連会社 ) に入社してしまった自称ソフト技術者。

得意な言語は MSX-BASIC だけのため、日本語も C 言語も、もちろん英語も不得意。面白そうなことには飛びつく性格からこちらに参加することに。MSX の半導体といえば東芝なので新旧東芝の技術で MSX を楽しんでいきたいと考えている。

1 週刊アスキー 実はいちばん儲けた!? MSX 陰の立役者はあのメーカーだった!  
MSX30 周年 (<http://weekly.ascii.jp/elem/000/000/152/152761/>)

# 名刺サイズはもう古い!?

## IoT モジュールの大本命! BlueNinja 誕生!

Cerevo 押切 真人

こんにちは! Cerevo で回路設計を担当している押切です。FlashAir の同人誌ではありますが、縁あって今回は Cerevo が東芝 セミコンダクター&ストレージ社の SoC を採用して開発した IoT モジュール「BlueNinja (読み:ブルー・ニンジャ)」(以下、本モジュール)の概要と使用例を解説したいと思います。

BlueNinja のコアとなる東芝の「TZ1001MBG」は Bluetooth® Low Energy のコントローラ機能、入力データを処理する Cortex-M4F プロセッサ、およびデータを保存するフラッシュメモリを 1 パッケージに内蔵した SoC です。さらに本モジュールには加速度センサ、気圧センサ、リチウムイオン電池の充電・放電回路を搭載。通信部は技適を取得済みです。

ブレイクアウトボードを取り付けることで、microUSB からの 5V 給電、mbed 互換のファイルの書き込み、最大 16 本の GPIO の引き出しに対応しています。

最大の特徴は本モジュール単体で 11x37.5mm という超小型サイズ。Arduino UNO に実装されている AVR マイコン (DIP28pin) とほぼ同じ大きさです (図 1)。

これまでのラビットプロトタイピングツールは試作向けが多く、サイズや技適などの規格取得の問題でそのまま実際の製品に組み込むことは難しく、再設計や規格取得対応などで量産には長い時間がかかりました。しかし、BlueNinja はそのまま量産製品に組み込めるので、量産 IoT 製品ををスピーディーに市場に投入可能となります

(図 2、ミニ四駆への実装例参照)。

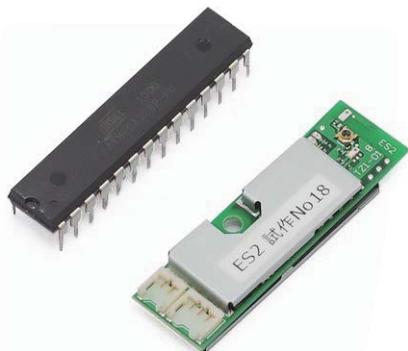


図 1: BlueNinja 外観



図 2: ミニ四駆への実装例

## 使用例

「BlueNinja」とタミヤのミニ四駆を組み合わせた、「ミニ四駆テレメトリシステム(ロガー)」です。本モジュールをミニ四駆のボディに内蔵して走らせ、以下のパラメータをBLE経由でスマートフォンにリアルタイムで表示します。

- 加速度 3 軸 (x, y, z)
- 角速度 3 軸 (x, y, z)
- 地磁気 3 軸 (x, y, z)
- 加速度から計算した姿勢角

小型リチウムポリマー電池もはめ込むだけの簡単接続できるため、本当に貼り付けるだけの実装です。図3は改造済車両とスマートフォンアプリの表示例です(写真は開発中のもの)。MFT2015では現物を展示予定ですので、完成品を是非ご覧になって下さい。

また、FlashAirとSPI接続することでWi-Fi通信とSDメモリアクセスが可能となります。このように組み合わせることで、超小型のWebストレージや、BLEでは届かない場所にある何かをネットワーク経由で遠隔操作する装置などが容易に実現できます。今後別の機会にこういった作例もご紹介できればと思っています。



図 3: ミニ四駆への実装例 2

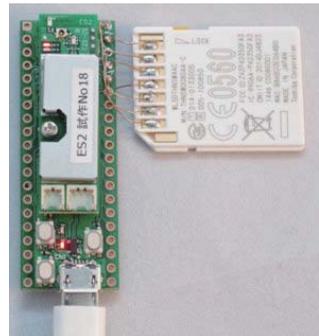


図 4: FlashAir との接続例

## 販売

8月中に直販サイト「Cerevo official store」で容易に入手頂ける体制を整えます。価格はこの記事が配布される頃には決定しているはずですが、製品はもちろん、アートや個人の一発ネタにも幅広く手軽に利用して頂ければと思います。



### 押切 真人

三児の父。本記事の為に社内チームλを結成。  
Special Thanks: ゆな、奥原、長谷川、あくやん  
Twitter: Cerevo Inc. Official @cerevo

## FlashAir 応援キャラクター「閃ソラ」

閃ソラ（ひらめきそら）は、FlashAir の非公式応援キャラクターです。  
とある航空会社の CA をしています。フライトでたびたび不在にしますが、オフにはミラーレス一眼で写真を撮ったり、電子工作したり、アプリ開発したり、忙しい毎日を送っています。

### ■ 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



### じむ

「会社でも技術報告書，特許提案 など事象を人に説明するときにはイラスト・漫画を描いて笑わせる人。（本人は真面目）それが高じて他部門から（ここ大事）も業務以外の「その手」話がたまに舞い降りる。（もち全部無償）

プライベートでもブログに日頃のモノ造り成果をアップしていて、ウケがいいのか現在では PV500 件 / 日をキープ中。

## ■ FlashAir Doujinshi 2 - FlashAir の同人誌 2

2015 年 8 月 1 日 第 1 版第 1 刷発行

2015 年 11 月 7 日 第 1 版第 2 刷発行

著者：高田 真里 / じむ / 伊藤 晋朗 / 笠原 / Pochio / GUGEN 崔 / Julie Watai

土居 / 寺西 / 余熱 / 村口 / 綾瀬 ヒロ / 西田 ラヂオ /

村口 / ペンギンキャット / せいみ まさみ / Cerevo 押切 真人

表紙イラスト：じむ

表紙デザイン：余熱 / じむ

編集：余熱

発行：FlashAir Developers

連絡先：support@flashair-developers.com

印刷：株式会社創英





<https://flashair-developers.com>